Oracle®

Using Oracle Machine Learning User Interface on Autonomous AI Database

E78525-87
September 2026

Oracle Using Oracle Machine Learning User Interface on Autonomous AI Database,

E78525-87

Primary Author: Moitreyee Hazarika

Contributors: Mark Hornick, Sherry LaMonica, Denny Wong, Eros Espinola, Francisco Noriega, Gloria Castillo, Jim Dadashev, Marat Spivak, Pedro Reta, Zabdiel Jaramillo Roman

Contents

4 OML Notebooks

Preface

This document describes how to use Oracle Machine Learning and provides references to related documentation.

- Audience
  This document is intended for data scientists, developers, and business users.
- Conventions
- Documentation Accessibility
- Related Resources
  For more information, see these related resources.

Audience

This document is intended for data scientists, developers, and business users.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Resources

For more information, see these related resources.

- *Getting Started with Oracle Cloud* Getting Started with Oracle Cloud
- *Accessibility Guide for Oracle Cloud Services* Accessibility Guide for Oracle Cloud Services

Preface

This document describes how to use Oracle Machine Learning and provides references to related documentation.

- Audience
  This document is intended for data scientists, developers, and business users.
- Conventions
- Documentation Accessibility
- Related Resources
  For more information, see these related resources.

Audience

This document is intended for data scientists, developers, and business users.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Resources

For more information, see these related resources.

- *Getting Started with Oracle Cloud* Getting Started with Oracle Cloud

- *Accessibility Guide for Oracle Cloud Services* Accessibility Guide for Oracle Cloud Services

1

What's New

Provides a summary of the latest enhancements and features for Oracle Machine Learning User Interface on Oracle Autonomous AI Database.

Table 1-1 New Features

Features	Description
Data Science Agent	Oracle Machine Learning UI offers an intelligent built-in conversational chatbot—Data Science Agent. The agent is included in Oracle Autonomous AI Database and is available in all regions. You provide the LLM, whether from a third-party AI provider, OCI GenAI Service, or one you privately host. Data Science Agent enables you to interact in natural language via the chat interface for a wide range of data science related tasks such as data profiling, data wrangling and transformation, statistical analysis of variables relationships, model training and evaluation, inference on new data, and so on. The response you get is both in natural language format and in structured format such as codeblocks, tables, charts, logs and so on. Data Science Agent is seamlessly integrated with the Select AI Agent framework. To use Data Science Agent, you must have your `DBMS_CLOUD_AI` profile (AI profile) and `DBMS_CLOUD` credentials (AI Credential) defined. You must have access to the relevant schemas and objects based on your role and privileges. For more information, see Manage AI Profiles. Key highlights of Data Science Agent include: • **Data discovery:** Data Science Agent can access and discover data locally as well as from remote sources including non-Oracle databases in multi-cloud environments. • **Data analysis and visualization:** Data Science Agent simplifies and automates data analysis while also providing insights into your data with built-in visualization. • **Data preparation and feature engineering:** The agent can profile datasets, identify data quality issues or outliers, clean and organize the data, compute correlations, and perform both feature selection and feature engineering. • **Model training and evaluation:** Data Science Agent handles model training, model evaluation, comparison, and inference, thereby providing clear explanations of metrics and results to support learning and decision-making. It supports the in-database Classification and Regression algorithms, including XGBoost, Support Vector Machine, Generalized Linear Model, Clustering, and Anomaly Detection. • **Security:** Data Science Agent runs entirely inside the database, and hence it provides you database-supported performance, security, and ease of operation. • **Interactive and autonomous mode:** You can use the agent in both interactive and autonomous mode. You can adopt an interactive, that is, conversational mode for clarifications, validation of assumptions, and step-by-step guidance. You can also choose to adopt the autonomous mode for repetitive tasks for faster iteration. In both modes, the agent maintains the contexts and ensures efficiency and transparency. Overall, it provides a friendly and conversational way to work with data, much like using a modern AI assistant.

Table 1-1 (Cont.) New Features

Features	Description
Oracle Machine Learning Workflow	Oracle Machine Learning UI now includes OML Workflow, a no-code interface for building and running end-to-end machine learning pipelines. Using a visual canvas, you can drag and drop nodes, connect them into a workflow, and run each step of the machine learning process—without writing code or requiring deep machine learning expertise. OML Workflow is fully integrated into Oracle Machine Learning UI and includes the following nodes: • Data Source Node • Feature Selection Node • Model Build Node • Model Evaluation Node • Model Apply Node See Workflows for more information.
GitHub notebooks enhancement	The GitHub notebook clone operation has been enhanced to support cloning of multiple notebooks simultaneously. You can now browse the folder structure of a remote GitHub repository and select multiple files for cloning in a single operation. For more information, see Clone and Access your GitHub Notebooks.
OML Jobs performance	The jobs functionality in Oracle Machine Learning UI has been enhanced for performance. If you have many notebooks in your project, the performance of OML UI is not affected when opening, cloning, and moving the notebooks across projects and workspaces. Earlier, the process of running an OML job involved cloning the associated notebook and storing it in the database. After the enhancement, the notebooks associated with any job are run directly instead of cloning the original notebook. The enhancement has the following user interfaces and behavior impacts: • All logs of the notebook job, including errors, are listed in a table on the job log page. Clicking on any job opens the job log on a separate page. The **Detail** column displays any error, as a clickable link. The details of older job logs are also available on the same page—the errors are listed as clickable links in the **Date** column on the same page. For details, see View Job Logs and About Jobs. • A new icon has been introduced on the Notebooks page. The icon is displayed against any notebook with a job associated with it. • A new column **Status** has been added on the Notebooks page. The statuses are —`Created`, `Running`, `Succeeded`, `Stopped`, and `Failed`. For details, see Access your Oracle Machine Learning Notebooks • If you open a notebook when a job associated with it is running, you will see a running horizontal bar indicating that the notebook is currently being run by a job. • If you have a notebook opened and the job associated with it begins to run, there will be no change in the UI, except a horizontal running bar and a message. The following conditions apply: – If you run the notebook, the notebook will be placed in a queue. It will wait until all the previous runs are complete, and the notebook will run only after all the queued runs are complete. – If you run a paragraph in a notebook, that paragraph will get placed in a queue until all the previous notebook runs are complete. • If you modify a paragraph in a notebook while it is being run by a job, the job result will not be affected. However, the changes to the notebook will be reflected only in the next job run. • You are not allowed to delete a notebook that is running. • If you try to delete multiple notebooks, and if one or more notebooks are running, then only the notebooks that are not running will be deleted.

Table 1-1 (Cont.) New Features

Features	Description
OML notebook and GitHub integration	Oracle Machine Learning UI has been enhanced to support direct interaction of OML Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories. With GitHub integration, you can interact with your GitHub repositories in the following ways: • Notebook import: You can import notebooks directly from your GitHub repository. When you import notebooks from your GitHub repository, it clones the notebooks and creates a local copy in OML UI by using the branching mechanism in GitHub. For more information, see Import a Notebook. • Notebook updates: You can update remote changes in your notebooks and also upload local changes in the notebooks back to your GitHub repository using the version control options **Pull changes** and **Push & commit** in OML notebook editor. For more information, see Edit and Sync your GitHub Notebook. • Credential management: You must have separate credentials to connect to your GitHub repositories. The credentials are created and securely stored in the Autonomous Database. For more information, see Create GitHub Credentials.
Support for email notifications	Oracle Machine Learning User Interface has been enhanced with the functionality to send email notifications for jobs. Jobs allow you to schedule the running of notebooks. You can now send email notifications to the specified users' email addresses about the selected events for the job. See Create Jobs to Schedule Notebooks for more information on how to send email notifications.
Support for NVIDIA GPU compute capabilities in OML Notebooks	Oracle Machine Learning Notebooks offers support for NVIDIA GPUs (Graphics Processing Unit) compute capabilities. With the new GPU capabilities in Oracle Machine Learning Notebooks, you can run advanced machine learning algorithms such as deep learning models, transformers (embedding models) for generating vectors, and small LLMs. The GPU feature is enabled for Oracle Autonomous Data Warehouse Serverless or Oracle Autonomous Transaction Processing Serverless instances with 16 or more ECPUs specified for the OML application. Both the Licensed and the Bring Your Own Licence (BYOL) versions are available. For cost details, refer to the *Oracle PaaS and IaaS Universal Credits Service Descriptions* document available on the Oracle Cloud Services contracts page. **Note** GPU resources are available only on paid Oracle Autonomous AI Database Serverless instances. GPU resources are not available on Always Free Oracle Autonomous AI Database Serverless or Oracle Autonomous AI Database Serverless instances with fewer than 16 ECPUs allocated.

Table 1-1 (Cont.) New Features

Features	Description
Oracle Machine Learning Notebooks Classic deprecated	Oracle Machine Learning Notebooks Classic has been deprecated since June 11, 2024. On October 29, 2024, you will no longer be able to create Classic notebooks, save them as templates, or select them for job scheduling. Existing Classic notebooks can be opened in read-only mode. You can continue to convert Classic notebooks to the new format using the **Copy to OML Notebooks** button on the Notebook Classic listing page. On December 31, 2024, Classic notebooks will no longer be available. The ADMIN user can access Classic notebooks in read-only mode and convert them to the new format. Jobs that still use Classic notebooks will show a status of `Disabled`. Associated job logs will not be accessible. On June 4, 2025, the ADMIN user will no longer have access to Classic notebooks, and any remaining notebooks will be deleted. If you have Classic notebooks or Classic template notebooks (personal or shared) that you wish to keep, you must convert these to the new format. If you have jobs that rely on Classic notebooks, these jobs must either be updated with a new notebook or recreated with a new notebook.
Oracle Machine Learning Notebooks update	Oracle Machine Learning User Interface offers an enhanced notebook environment. Initially released as **Notebooks EA (Early Adopter)** in Oracle Autonomous AI Database Serverless, it is now accessed using **Notebooks** under the left navigation menu and home page. The enhanced notebook interface supports SQL, SQL Script, R, Python, Conda, and Markdown interpreters. You can write code, text, create rich visualizations, and perform data analytics including machine learning in the enhanced notebooks. **Note** The original Zeppelin-based notebook interface is still available for a limited time under the left navigation menu item **Notebooks Classic**.
Support for model monitoring in Oracle Machine Learning User Interface	Oracle Machine Learning User Interface offers support for model monitoring. It allows you to create model monitors. The model monitors enable you to monitor the quality of model predictions over time, and provides you with insights on the underlying causes.
Support for data monitoring in Oracle Machine Learning User Interface	Oracle Machine Learning User Interface offers support for data monitoring. It allows you to monitor your data and evaluate how your data evolves over time. It helps you with insights on trends and multivariate dependencies in the data. It also provides you an early warning about data drift.

Table 1-1 (Cont.) New Features

Features	Description
Support for enhanced notebooks in Autonomous Database - Serverless	Oracle Machine Learning User Interface offers a new enhanced notebook environment *Notebooks EA (Early Adopter)* in Autonomous Database - Serverless. The enhanced notebook supports SQL, SQL Script, R, Python, Conda, and Markdown interpreters. You can write code, text, create rich visualizations, and perform data analytics including machine learning in the enhanced notebooks. **Note** The enhanced notebook is available in the Oracle Machine Learning Notebook Early Adopter release. During the Early Adopter release period, both Zeppelin and the enhanced notebooks will be available, after which all notebooks will be converted to the new notebook environment. During the Early Adopter phase, you can use both the original Zeppelin and new Early Adopter notebook interfaces. Notebooks in the original interface can be copied to the Early Adopter release. The enhanced notebook interface in Oracle Autonomous AI Database Serverless provides the following enhanced features and user experiences: • **Rich and enhanced user experience:** The enhanced notebook offers modern look and feel, and richer visualization with many charting options. This will benefit users to better visualize and understand their data. In addition, it offers some useful features like side-by-side versions comparison, option to add comments to paragraphs, full screen size mode for paragraphs, option to define paragraph dependency, and so on. • **High availability:** The enhanced notebook, a multi-tenant application is deployed to the same middle-tier as Oracle Machine Learning server, and this requires no additional resources. Therefore, it is always running and readily available to render the new enhanced notebooks. • **High scalability:** The enhanced notebook assures high scalability in production. To scale up due to increased user demands, additional notebook instances can be easily added. There are tools to monitor system loads, and if a system is consistently overloaded, additional instance can be easily added to mitigate risks related to scalability.
Support for Python and R third-party libraries	Third-party libraries for Python and R are available on Oracle Machine Learning Notebooks. Oracle Machine Learning UI provides the Conda interpreter to install third-party Python and R libraries inside a notebook session. Conda is an open-source package and environment management system that enables the use of environments containing third-party Python and R libraries. • Users with `OML_SYS_ADMIN` role can install Python and R third-party libraries and upload them to object storage for persistence. The user with `OML_SYS_ADMIN` role is the administrator, also known as the admin. • Users with `OML_DEVELOPER` role can use the Conda interpreter to download and activate the third-party libraries using the Conda environment that are provisioned by the administrator. The user with `OML_DEVELOPER` role is the regular Oracle Machine Learning user.

Table 1-1 (Cont.) New Features

Features	Description
Support for R	Oracle Machine Learning for R is supported within Oracle Machine Learning Notebooks. By using Oracle Machine Learning for R, you can perform data exploration and machine learning modeling. OML4R is available through Oracle Machine Learning Notebooks on Oracle Autonomous AI Database Serverless, including Autonomous AI Lakehouse , Autonomous AI Transaction Processing and Oracle Autonomous AI JSON Database services.
Support for cross-region Autonomous Data Guard	Oracle Machine Learning Notebooks provide cross-region Autonomous Data Guard support in newly provisioned and migrated databases.
Oracle Machine Learning repository migrated from Serverless database to each respective Oracle Autonomous AI Database instance	The Oracle Machine Learning (OML) repository has been migrated from Serverless database to each respective Oracle Autonomous AI Database instance. The migration of the Oracle Machine Learning repository ensures: • That all OML objects such as tables, jobs, stored procedures, and metadata are moved to the appropriate Oracle Autonomous AI Database instance. • Provides support for Refreshable Clones, which enables cloning of the Oracle Machine Learning metadata as well. **Note** The migration of the Oracle Machine Learning (OML) repository is expected to be completed over a period of 30 days. The OML repository version is mentioned in **About** in the `<user>` drop-down list on the top right corner of your Oracle Machine Learning User Interface page. If the version is `1.0.0.0.0`, it indicates that the OML metadata is still in the Serverless database. If the version is `22.x`, it indicates that the OML repository has been migrated to your Oracle Autonomous AI Database instance.
Oracle Machine Learning Notebook supported on all Oracle Autonomous AI Database clones	Oracle Machine Learning Notebook is supported on all types of Oracle Autonomous AI Database Serverless clones, including: • Full Clone: a new database is created with the data in the source database and metadata. • Refreshable Clone: a read-only full clone is created that can be easily refreshed with the data from the source database • Metadata Clone: a new database is created that includes all of the source database schema metadata, but not the source database data. **Note** For a metadata clone, the Example Template notebooks are not supported.

2

Get Started

This section discusses how to get started with Oracle Machine Learning User Interface, and use Apache Zeppelin-based machine learning notebooks along with the OML Notebooks, where you can perform data exploration and data visualization, data preparation and machine learning.

- Get Started with Oracle Machine Learning User Interface
  Here is how you can get started with Oracle Machine Learning User Interface (UI).
- Access Oracle Machine Learning User Interface
  You can access Oracle Machine Learning **User Interface** from Autonomous AI Database.
- Oracle Machine Learning User Interface Home Page
  The Oracle Machine Learning User Interface home page provides you quick links to important interfaces, help links, and the log of your high-level recent activities.
- Oracle Machine Learning Notebooks
  Oracle Machine Learning Notebooks is a collaborative web-based interface that supports notebook creation, scheduled run, and versioning. You can document work using Markdown and run SQL, R, and Python code for data exploration, visualization, and preparation, and machine learning model building, evaluation, and deployment. Oracle Machine Learning Notebooks also provide the Conda interpreter to create a custom conda environment that includes user-specified third-party Python and R libraries.
- GitHub Notebooks
  Oracle Machine Learning UI supports direct interaction of Oracle Machine Learning Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories.
- AutoML UI
  AutoML User Interface (AutoML UI) is an Oracle Machine Learning interface that provides you no-code automated machine learning modeling. When you create and run an experiment in AutoML UI, it performs automated algorithm selection, feature selection, and model tuning, thereby enhancing productivity as well as potentially increasing model accuracy and performance.
- Data Science Agent
  Data Science Agent is an intelligent built-in conversational chatbot integrated with Oracle Machine Learning UI included in your Oracle Autonomous AI Database subscription. You must provide the LLM, whether from a third-party AI provider, OCI GenAI Service, or one you privately host. You can run many common data science workflows, including discovery, exploration, preparation, model training, evaluation, and supported scoring workflows, using natural language.
- Workflows
  Workflows enable you to design machine learning processes using a drag-and-drop canvas composed of nodes and connectors. Each node represents a task or step in the machine learning workflow. The connectors define the flow between these steps.
- Data Monitoring
  Data Monitoring evaluates how your data evolves over time. It helps you with insights on trends and multivariate dependencies in the data. It also gives you an early warning about data drift.

- Model Monitoring
  Model monitoring allows you to monitor the quality of model predictions over time and helps you with insights on the causes of model quality issues.
- REST API for Oracle Machine Learning Services
  The REST API for Oracle Machine Learning Services provides REST API endpoints hosted on Oracle Autonomous AI Database. These endpoints allow you to store Machine Learning models along with its metadata, and create scoring endpoints for the model.

Related Topics

- About AutoML UI
- Jobs
  Jobs allow you to schedule the running of notebooks. On the Jobs page, you can create jobs, duplicate jobs, start and stop jobs, delete jobs, and monitor job status by viewing job logs, which are read-only notebooks.
- Templates
  By using the Oracle Machine Learning Notebooks templates, you can collaborate with other users by sharing your work, publishing your work as reports, and by creating notebooks from templates. You can store your notebooks as templates, share notebooks, and provide sample templates to other users.

2.1 Get Started with Oracle Machine Learning User Interface

Here is how you can get started with Oracle Machine Learning User Interface (UI).

1. Request access to Oracle Machine Learning UI. Contact your Service Administrator to create your OML user credentials and provide access to your Oracle Machine Learning UI account.
2. Access the Oracle Machine Learning UI account by using your credentials. In case you forget your password, then request the Administrator to reset it.

> **Note**
>
> Once you receive your new password, you must change it immediately. Refer to the Oracle Machine Learning password policy for more information.

3. Once you log in for the first time, a workspace and project will be created for you. You can start creating your notebook. You can also create your additional projects in the default workspace or create new workspace with corresponding projects.

> **Note**
>
> If you are using Oracle Cloud Free Tier, you have to provision an Oracle Autonomous AI Database. For more information, see Provision an Autonomous AI Database

Related Topics

- OML Notebooks Interactive Tour
- About Oracle Machine Learning Password Policy
  All Oracle Machine Learning users must follow the password policy to create a strong and secure password.

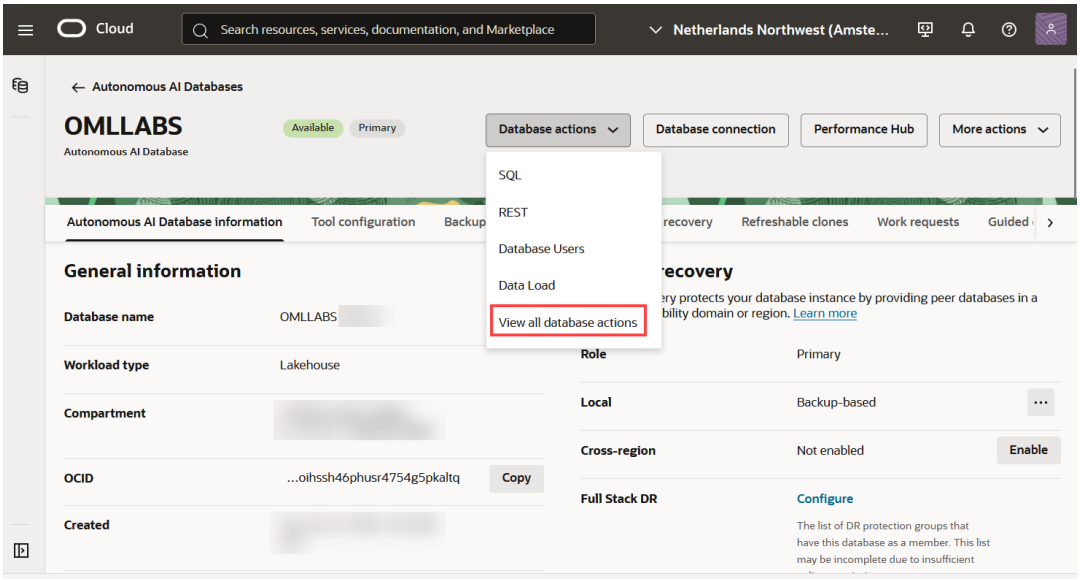
2.2 Access Oracle Machine Learning User Interface

You can access Oracle Machine Learning **User Interface** from Autonomous AI Database.

To access Oracle Machine Learning User Interface (UI) from Autonomous AI Database:

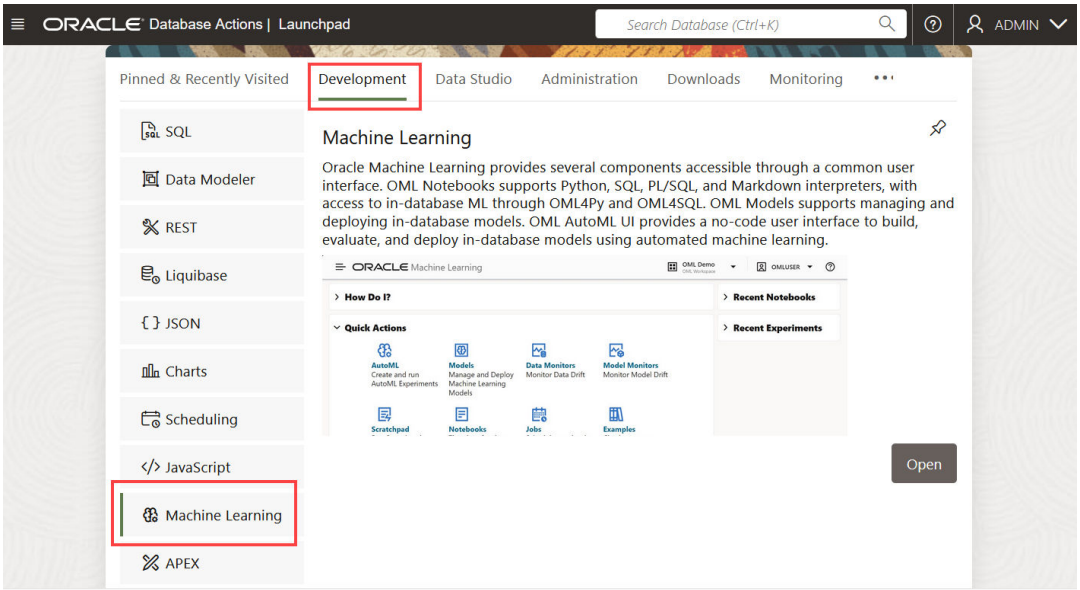
1. Select your Oracle Autonomous AI Database instance and on the Autonomous AI Database information page ,click **Database Actions** and then click **View all database actions**.

Figure 2-1 Database Actions



2. On the Database Actions page, go to the **Development** section and click **Machine Learning**. The Oracle Machine Learning sign in page opens.

Figure 2-2 Oracle Machine Learning option on Database Actions - Development tab



3. On the Oracle Machine Learning sign in page, enter your username and password.
4. Click **Sign In**.

This opens the Oracle Machine Learning user application.

- About Oracle Machine Learning Password Policy
  All Oracle Machine Learning users must follow the password policy to create a strong and secure password.

2.2.1 About Oracle Machine Learning Password Policy

All Oracle Machine Learning users must follow the password policy to create a strong and secure password.

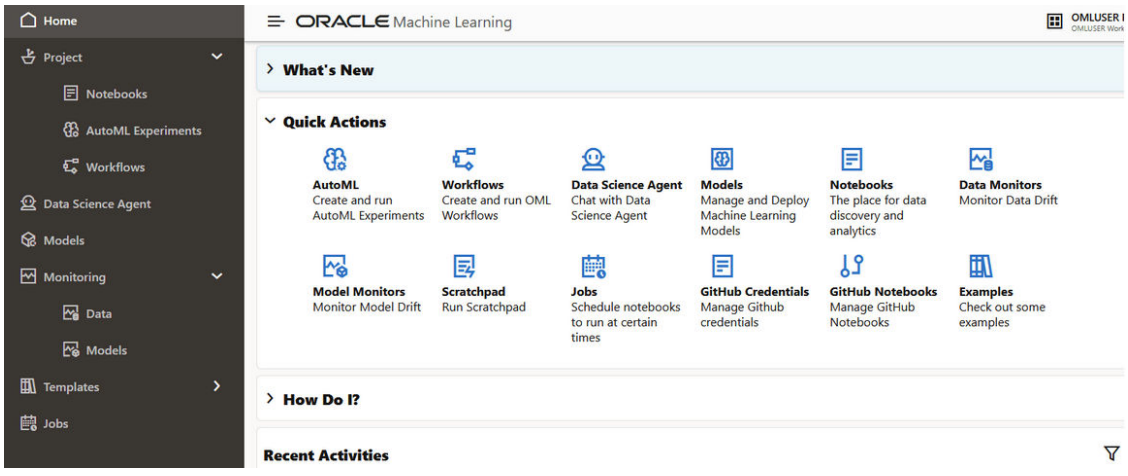
When changing or modifying your Oracle Machine Learning password, ensure that you follow these conditions:

- The password must be between 12 and 30 characters long. It must include at least one uppercase letter, one lowercase letter, and one numeric character.
- The password cannot contain the username.
- The password cannot be one of the last 4 passwords used for the same username.
- The password cannot contain the double quote (") character.
- The password must not be the same password that is set less than 24 hours ago.

2.3 Oracle Machine Learning User Interface Home Page

The Oracle Machine Learning User Interface home page provides you quick links to important interfaces, help links, and the log of your high-level recent activities.

Figure 2-3 Oracle Machine Learning UI Home Page



On the home page, you can access:

- The **How Do I** help links to:
  - **Get Started**

  - **Use AutoML**
  - **Deploy Models**
  - **Monitor Data**
  - **Create Notebooks**
  - **Create Jobs**
  - **Manage Permissions**
  - **Try It**
- The **Quick Actions** links to:
  - **AutoML**
  - **Workflows**
  - **Data Science Agent**
  - **Models**
  - **Notebooks**
  - **Data Monitors**
  - **Model Monitors**
  - **Scratchpad**
  - **Jobs**
  - **GitHub Credentials**
  - **GitHub Notebooks**
  - **Examples**
- The log of your **Recent Activities.**
- The application navigation by clicking  on the top left corner of the home page.
- The options to select and create new projects, access recent projects, manage workspace, and set workspace permissions by clicking  on the top right corner of the home page.
- The **Recent Notebooks** on the right pane of the home page.
- Preferences
  You have the option to change your user preferences such as timezone settings in Oracle Machine Learning User Interface.
- GitHub Notebooks
  Oracle Machine Learning UI supports direct interaction of Oracle Machine Learning Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories.
- Change Password
  You have the option to change the password from Oracle Machine Learning User Interface.

Related Topics

- Notebooks

- Jobs
  Jobs allow you to schedule the running of notebooks. On the Jobs page, you can create jobs, duplicate jobs, start and stop jobs, delete jobs, and monitor job status by viewing job logs, which are read-only notebooks.
- Examples
  The Example Templates page lists the pre-populated Oracle Machine Learning notebook templates. You can view and use these templates to create your notebooks.

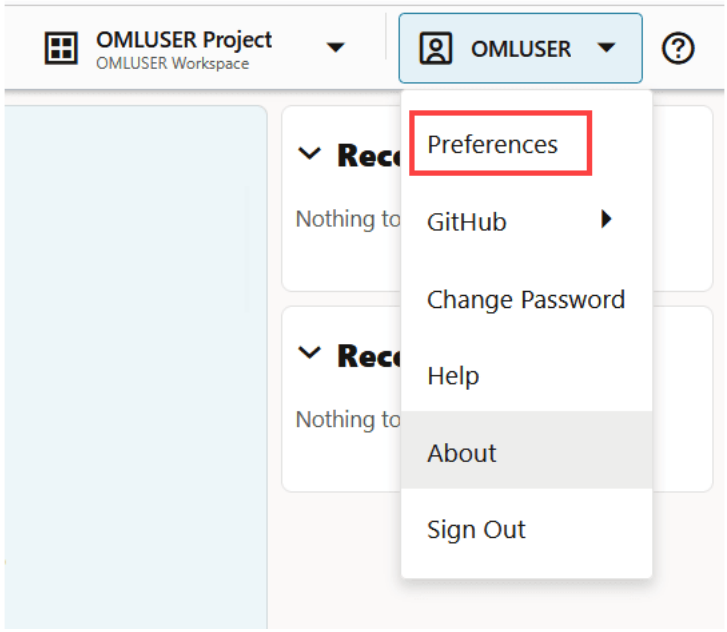
2.3.1 Preferences

You have the option to change your user preferences such as timezone settings in Oracle Machine Learning User Interface.

To edit your user profile preferences:

1. Sign into Oracle Machine Learning User Interface.
2. Click on the user profile menu at the top right corner of the page and click **Preferences**.

Figure 2-4 Preferences



3. On the Preferences page, you have the following options:
   - **Profile:** Click profile to view your Username, Role and other details such as First Name, Last Name and Email Address.
   - **Language:** Click Language to edit your Timezone settings. The Locale field is non-editable.

2.3.2 GitHub Notebooks

Oracle Machine Learning UI supports direct interaction of Oracle Machine Learning Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories.

With GitHub integration, you can interact with your GitHub repositories in the following ways:

- Notebook clone and import—You can import notebooks directly from your GitHub repository. When you import notebooks from your GitHub repository, it clones the notebooks and creates a local copy in OML UI by using the branching mechanism in GitHub. See Clone and Access your GitHub Notebooks and Import a Notebook.
- Notebook updates and sync—You can update remote changes in your notebooks and also upload local changes in the notebooks back to your GitHub repository using the version control options Pull changes and Push & commit in OML notebook editor. See Edit and Sync your GitHub Notebook.
- Credential management—You must have separate credentials to connect to your GitHub repositories. The credentials are created and securely stored in the Autonomous AI Database. See Create GitHub Credentials.

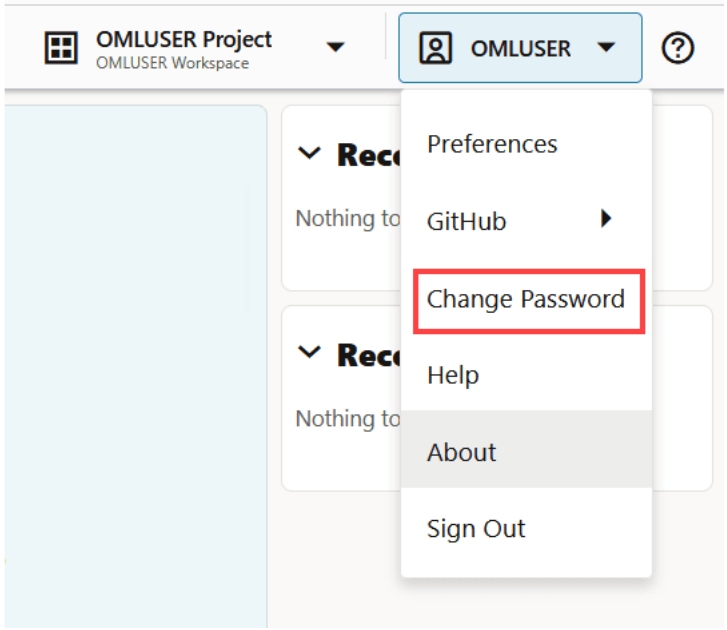
2.3.3 Change Password

You have the option to change the password from Oracle Machine Learning User Interface.

To change your password:

1. Sign into Oracle Machine Learning User Interface.
2. Click on the user profile menu at the top right corner of the page and click **Change Password**.

Figure 2-5 Change Password



3. On the Change Password page, enter the following:
   a. **Old Password**
   b. **Password**
   c. **Confirm Password**
   d. Click **Apply**.
4. After your password is changed, a message is displayed indicating the same.

This completes the task of changing your password.

- About Oracle Machine Learning Password Policy
  All Oracle Machine Learning users must follow the password policy to create a strong and secure password.

2.3.3.1 About Oracle Machine Learning Password Policy

All Oracle Machine Learning users must follow the password policy to create a strong and secure password.

When changing or modifying your Oracle Machine Learning password, ensure that you follow these conditions:

- The password must be between 12 and 30 characters long. It must include at least one uppercase letter, one lowercase letter, and one numeric character.
- The password cannot contain the username.
- The password cannot be one of the last 4 passwords used for the same username.
- The password cannot contain the double quote (") character.
- The password must not be the same password that is set less than 24 hours ago.

2.4 Oracle Machine Learning Notebooks

Oracle Machine Learning Notebooks is a collaborative web-based interface that supports notebook creation, scheduled run, and versioning. You can document work using Markdown and run SQL, R, and Python code for data exploration, visualization, and preparation, and machine learning model building, evaluation, and deployment. Oracle Machine Learning Notebooks also provide the Conda interpreter to create a custom conda environment that includes user-specified third-party Python and R libraries.

Key features of Oracle Machine Learning Notebooks:

- Allows collaboration among data scientists, business and data analysts, application and dashboard developers, Database Administrators, and IT professionals
- Leverages the scalability and performance of Oracle Autonomous AI Database Platform
- Supports scheduling notebooks as jobs for one-off and recurring running of notebooks
- Enables versioning of notebooks
- Supports interpreters for SQL, PL/SQL, Python, R, Markdown, and Conda
- Typical Actions when Using Oracle Machine Learning Notebooks
  To begin with Oracle Machine Learning Notebooks, refer to the tasks listed in the table as a guide.

2.4.1 Typical Actions when Using Oracle Machine Learning Notebooks

To begin with Oracle Machine Learning Notebooks, refer to the tasks listed in the table as a guide.

Tasks	More Information
Access Oracle Machine Learning User Interface	Access Oracle Machine Learning User Interface (UI)
Create workspaces	Create Projects and Workspaces
Create projects	Create Projects and Workspaces

Tasks	More Information
Access and create a notebook	Access and create a Notebook
Manage your GitHub notebook	• Manage GitHub Notebooks • Edit and Sync your GitHub Notebook • Create GitHub Credentials
Enable GPU Compute Capabilities in a Notebook	Enable GPU Compute Capabilities in a Notebook through the Python Interpreter
Create and Run an AutoML Experiment	Create AutoML UI Experiment
Run a Notebook using SQL Interpreter	Use the SQL Interpreter in a Notebook Paragraph
Run a Notebook using Python Interpreter	Use the Python Interpreter in a Notebook Paragraph
Run a Notebook using R Interpreter	Use the R Interpreter in a Notebook Paragraph
Run a Notebook using Conda Interpreter	Use the Conda Interpreter in a Notebook Paragraph
Use the Markdown Interpreter	Use the Markdown Interpreter and Generate Static html from Markdown Plain Text
Create a Data Monitor	Create a Data Monitor
Create a Model Monitor	Create a Model Monitor
Use the Scratchpad	Use the Scratchpad
Create jobs to schedule notebooks	Create Jobs to Schedule Notebook

Related Topics

- Grant Workspace Permissions
  You can collaborate with other users in Oracle Machine Learning User Interface (UI) by granting permissions to access your workspace. A workspace consists of projects. Projects consist of notebooks and experiments.

2.5 GitHub Notebooks

Oracle Machine Learning UI supports direct interaction of Oracle Machine Learning Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories.

With GitHub integration, you can interact with your GitHub repositories in the following ways:

- Notebook clone and import—You can import notebooks directly from your GitHub repository. When you import notebooks from your GitHub repository, it clones the notebooks and creates a local copy in OML UI by using the branching mechanism in GitHub. See Clone and Access your GitHub Notebooks and Import a Notebook.
- Notebook updates and sync—You can update remote changes in your notebooks and also upload local changes in the notebooks back to your GitHub repository using the version control options Pull changes and Push & commit in OML notebook editor. See Edit and Sync your GitHub Notebook.
- Credential management—You must have separate credentials to connect to your GitHub repositories. The credentials are created and securely stored in the Autonomous AI Database. See Create GitHub Credentials.

2.6 AutoML UI

AutoML User Interface (AutoML UI) is an Oracle Machine Learning interface that provides you no-code automated machine learning modeling. When you create and run an experiment in

AutoML UI, it performs automated algorithm selection, feature selection, and model tuning, thereby enhancing productivity as well as potentially increasing model accuracy and performance.

> **Note**
>
> Oracle Machine Learning AutoML UI is desupported and will be unavailable starting January 15, 2027. Customers using OML AutoML UI should migrate to Oracle Data Science Agent, which uses new, advanced technology to automate data science and machine learning tasks and produce results superior to those provided by OML AutoML UI. For details, see Oracle Machine Learning Data Science Agent.

The following steps comprise a machine learning modeling workflow and are automated by the AutoML user interface:

1. Algorithm Selection: Ranks algorithms likely to produce a more accurate model based on the dataset and its characteristics, and some predictive features of the dataset for each algorithm.
2. Adaptive Sampling: Finds an appropriate data sample. The goal of this stage is to speed up Feature Selection and Model Tuning stages without degrading the model quality.
3. Feature Selection: Selects a subset of features that are most predictive of the target. The goal of this stage is to reduce the number of features used in the later pipeline stages, especially during the model tuning stage to speed up the pipeline without degrading predictive accuracy.
4. Model Tuning: Aims at increasing individual algorithm model quality based on the selected metric for each of the shortlisted algorithms.
5. Feature Prediction Impact: This is the final stage in the AutoML UI pipeline. Here, the impact of each input column on the predictions of the final tuned model is computed. The computed prediction impact provides insights into the behavior of the tuned AutoML model.

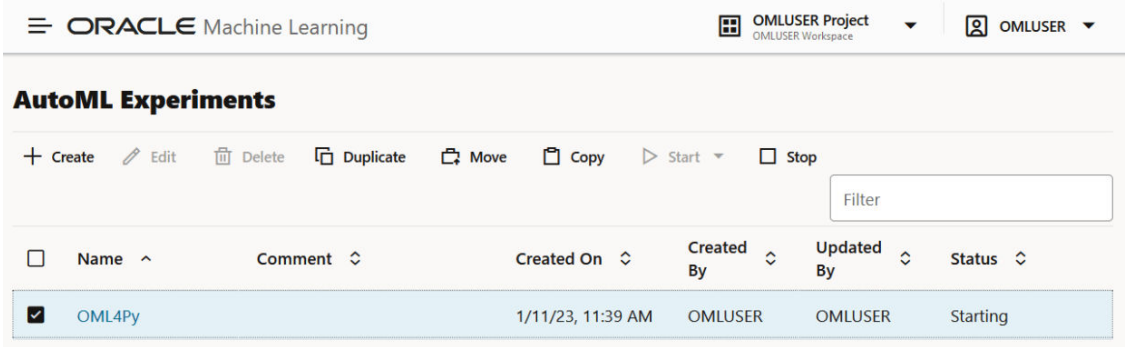
Business users without extensive data science background can use AutoML UI to create and deploy machine learning models. Oracle Machine Learning AutoML UI provides two functional features:

- Create machine learning models
- Deploy machine learning models

AutoML UI Experiments

When you create an experiment in AutoML UI, it automatically runs all the steps involved in the machine learning workflow. In the Experiments page, all the experiments that you have created are listed. To view any experiment details, click an experiment. Additionally, you can perform the following tasks:

Figure 2-6 AutoML Experiments page



- **Create:** Click **Create** to create a new AutoML UI experiment. The AutoML UI experiment that you create resides inside the project that you selected in the Project under the Workspace.
- **Edit:** Select any experiment that is listed here, and click **Edit** to edit the experiment definition.
- **Delete:** Select any experiment listed here, and click **Delete** to delete it. You cannot delete an experiment which is running. You must first stop the experiment to delete it.
- **Duplicate:** Select an experiment and click **Duplicate** to create a copy of it. The experiment is duplicated instantly and is in Ready status.
- **Move:** Select an experiment and click **Move** to move the experiment to a different project in the same or a different workspace. You must have either the `Administrator` or `Developer` privilege to move experiments across projects and workspaces.

> **Note**
>
> An experiment cannot be moved if it is in RUNNING, STOPPING or STARTING states, or if an experiment already exists in the target project by the same name.

- **Copy:** Select an experiment and click **Copy** to copy the experiment to another project in the same or different workspace.
- **Start:** If you have created an experiment but have not run it, then click **Start** to run the experiment.
- **Stop:** Select an experiment that is running, and click **Stop** to stop the running of the experiment.

Related Topics

- Automatic Machine Learning

2.7 Data Science Agent

Data Science Agent is an intelligent built-in conversational chatbot integrated with Oracle Machine Learning UI included in your Oracle Autonomous AI Database subscription. You must provide the LLM, whether from a third-party AI provider, OCI GenAI Service, or one you privately host. You can run many common data science workflows, including discovery,

exploration, preparation, model training, evaluation, and supported scoring workflows, using natural language.

As a human-in-the-loop chatbot, Data Science Agent requires your intervention and approval in the following areas:

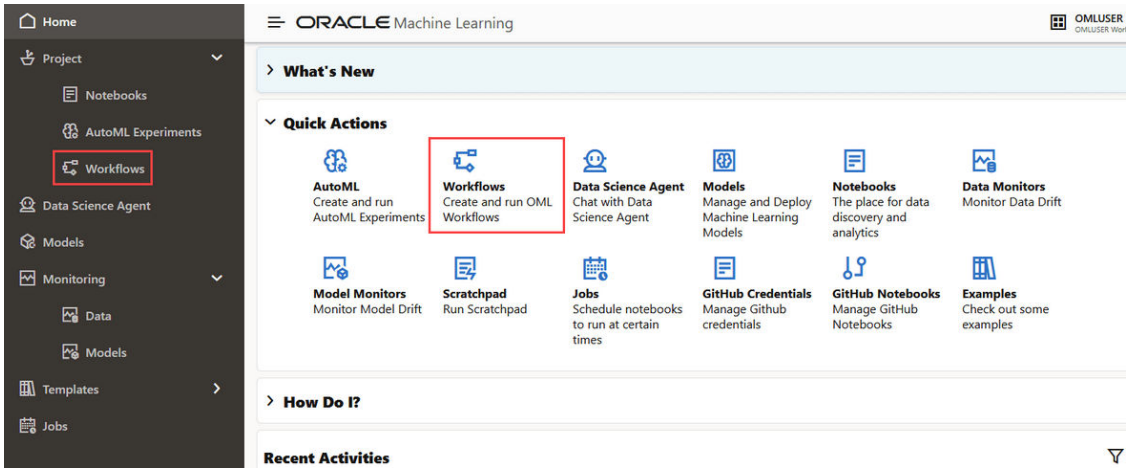
- **Object association:** Associating database objects with a conversation improves precision and efficiency. Objects must be associated or approved before the agent creates views that use them; other supported operations may still use accessible objects based on the user's database privileges. See Add Objects to Data Science Agent Conversation.
- **Conversation management:** The scope of Data Science Agent is limited to the active conversation context only. You must create, manage, and delete your conversations. See Manage Data Science Conversations.
- **Profile selection:** You must select an AI profile, and therefore, the LLM that powers the agent. You can also switch profiles mid-conversation. See Manage AI Profile and Service Level of Data Science Conversation.
- **Approval for object view:** During any conversation, Data Science Agent compares the `object_list` with the objects already associated with the active conversation. If required, the agent proposes any table or view outside the user-approved set. It stops the conversation and explicitly seeks your approval to associate those missing objects before continuing. If you approve associating the missing objects, the association is recorded and the agent continues the conversation. Else, the operation is canceled and no view is created.

2.8 Workflows

Workflows enable you to design machine learning processes using a drag-and-drop canvas composed of nodes and connectors. Each node represents a task or step in the machine learning workflow. The connectors define the flow between these steps.

To access Workflows, click **Workflows** on your OML UI home page. Alternatively, you can select **Workflows** under Project on the left navigation menu to open the Workflows page.

Figure 2-7 Home page with Workflows icon highlighted



The Workflows page lists all the workflows that are created.

Figure 2-8 Workflows Listing page

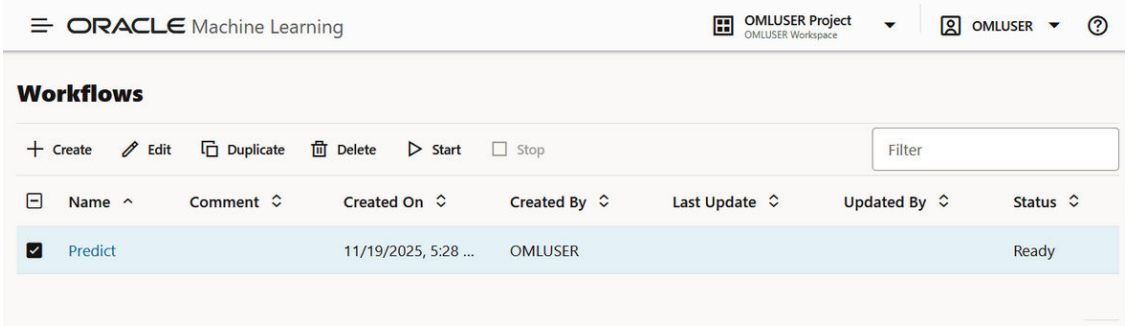
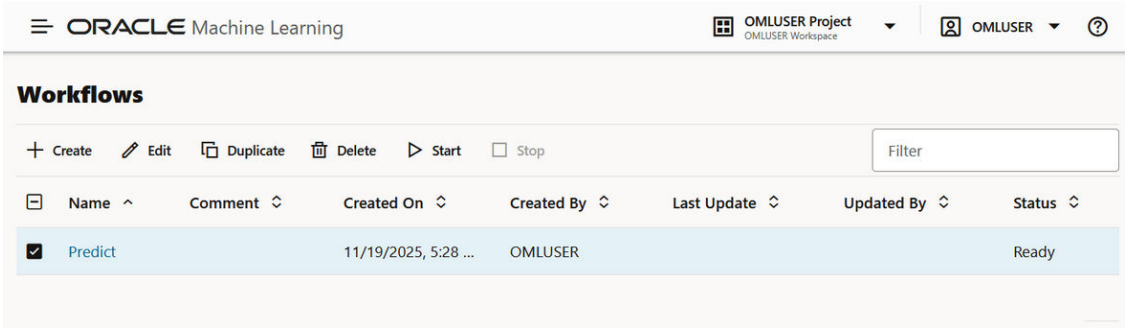


Figure 2-9 Workflows Listing page



You can perform the following tasks:

- **Create:** Click **Create** to create a new workflow.
- **Edit:** Select a workflow from the list and click **Edit** to edit it.
- **Duplicate:** Select any workflow from the list and click **Duplicate** to create a copy. The duplicated workflow will immediately appear with a Ready status.
- **Delete:** Select any workflow from the list and click **Delete** to remove it. You cannot delete a workflow that is currently running. You must stop it before attempting to delete.
- **Start:** If you have created a workflow but have not run it, click **Start** to run the workflow.
- **Stop:** If a workflow is running, select it and click **Stop** to halt the running of the workflow.

2.9 Data Monitoring

Data Monitoring evaluates how your data evolves over time. It helps you with insights on trends and multivariate dependencies in the data. It also gives you an early warning about data drift.

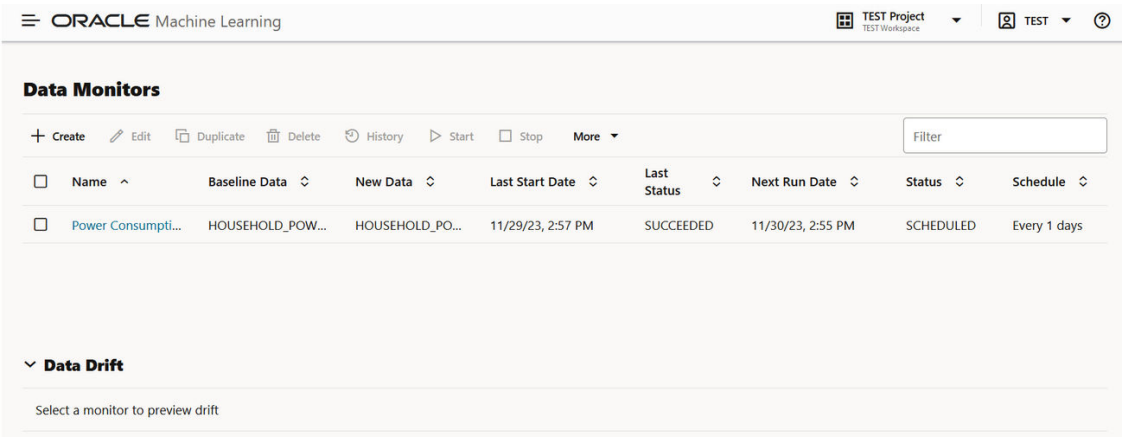
Data drift occurs when data diverges from the original baseline data over time. Data drift can happen for a variety of reasons, such as a changing business environment, evolving user behavior and interest, data modifications from third-party sources, data quality issues, or issues with upstream data processing pipelines.

The key to accurately interpret your models and to ensure that the models are able to solve business problems is to understand how data evolves over time. Data monitoring is complementary to successful model monitoring, as understanding the changes in data is

critical in understanding the changes in the efficacy of the models. The ability to quickly and reliably detect changes in the statistical properties of your data ensures that your machine learning models are able to meet business objectives.

You can monitor your data using the data monitoring functionality of Oracle Machine Learning User Interface. To monitor your data, click on the Cloud menu on the Oracle Machine Learning UI home page, click **Monitoring** and then click **Data** to open the Data Monitors page. On the Data Monitors page, you can perform the following tasks:

Figure 2-10 Data Monitors page

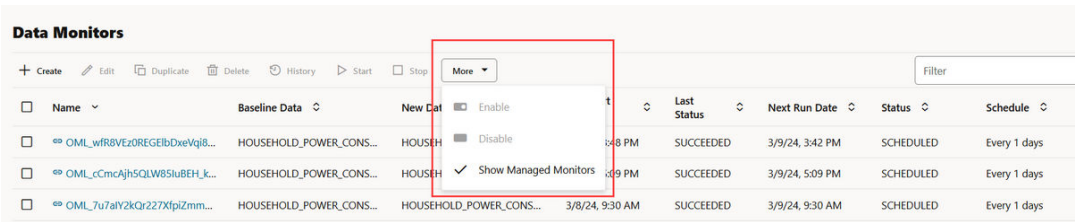


- **Create:** Create a data monitor.

  > **Note**
  >
  > The supported data types for data monitoring are NUMERIC and CATEGORICAL.

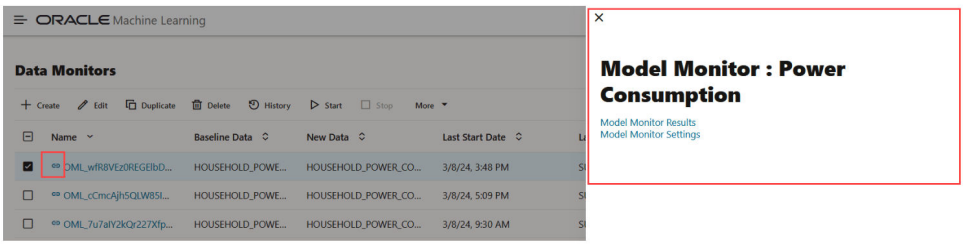
- **Edit:** Select a data monitor and click **Edit** to edit a data monitor.
- **Duplicate:** Select a data monitor and click **Duplicate** to create a copy of the monitor.
- **Delete:** Select a data monitor and click **Delete** to delete a data monitor.
- **History:** Select a data monitor and click **History** to view the runtime details. Click **Back to Monitors** to go back to the Data Monitoring page.
- **Start:** Start a data monitor.
- **Stop:** Stop a data monitor that is running.
- **More:** Click **More** for additional options to:

Figure 2-11 More option under Data Monitors



- **Enable:** Select a data monitor and click **Enable** to enable a monitor that is disabled. By default, a data monitor is enabled. The status is displayed as `SCHEDULED`.
- **Disable:** Select a data monitor and click **Disable** to disable a data monitor. The status is displayed as `DISABLED`.
- **Show Managed Monitors:** Click this option to view the data monitors that are created and managed by OML Services REST API and Model Monitors in Oracle Machine Learning UI. The data monitors that are managed by these two components have a system generated name, and are indicated by specific icons against its name.
  * Click on the link icon against a managed data monitor name to view the details of the associated model monitor. The associated model monitor details are displayed on a separate pane that slides in. The slide-in pane displays the model monitor name with links to view the model monitor results and settings. Clicking on the link icon also displays the data drift details on the lower pane of the Data Monitors page. Click on the X on the top left corner to close the pane.

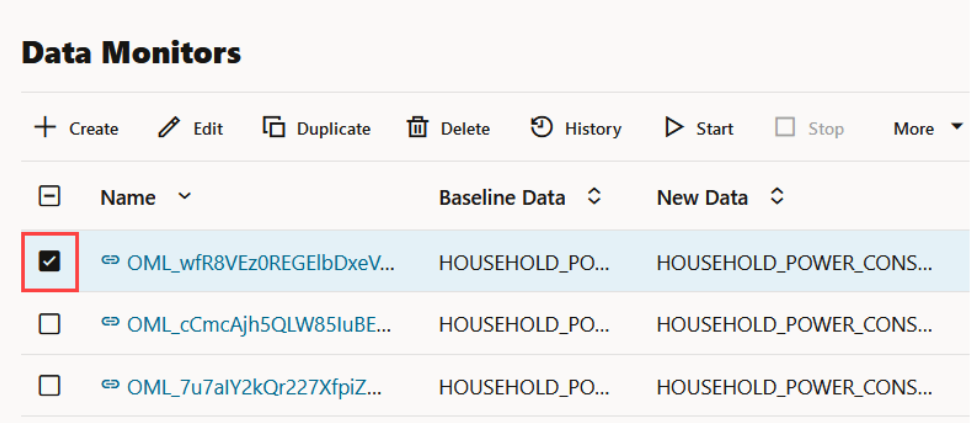
Figure 2-12 Data Monitors page displaying the associated model monitor results and settings



In this example, the slide-in pane displays the details of the model monitor Power Consumption. On the slide-in pane:

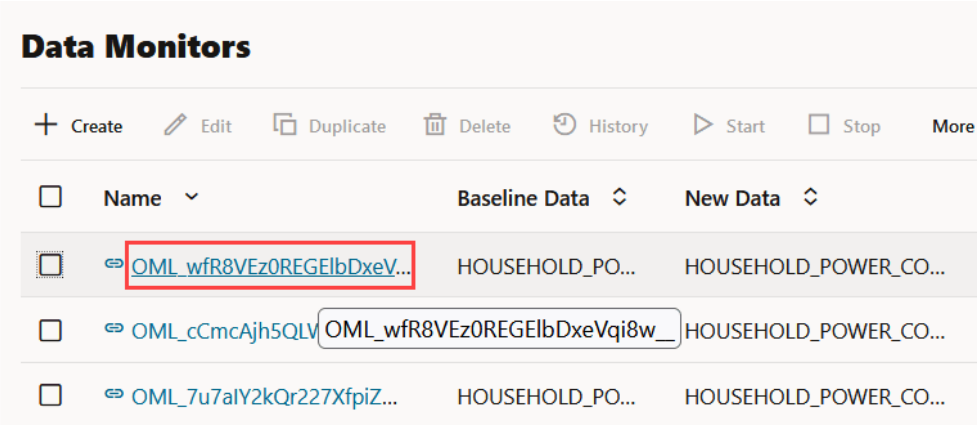
  * Click **Model Monitor Results** to view the results computed by the model monitor — settings, models, model drift, metric, and prediction statistics. Click **Monitors** to return to the Data Monitors page. See View Model Monitor Results.
  * Click **Model Monitor Settings** to view and edit the settings, details, and models monitored by the model monitor on the Edit Model Monitor page. Click **Cancel** to return to the Data Monitors page. Click **Save** to save any changes.
* Click on the check box against the data monitor name to view the data drift values on the lower pane.

Figure 2-13 Select a managed data monitor



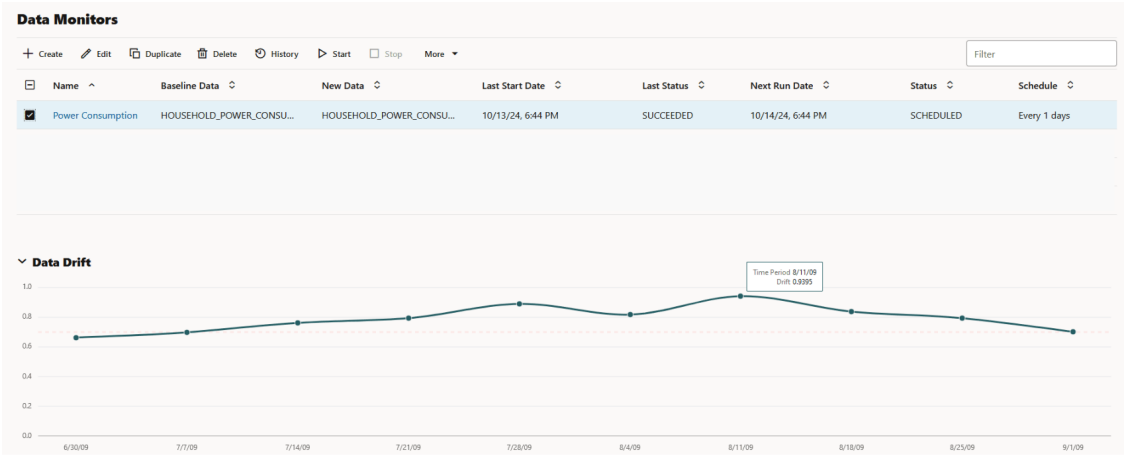
* Click on the data monitor name to view the details of the data monitor — settings, data drift values and monitored features.

Figure 2-14 Data monitor click



The Data Monitors page displays the information about the selected monitor: Monitor name, Baseline Data, New Data, Last Start Date, Last Status, Next Run Data, Status, and Schedule. The page also displays the data drift, if the data monitor has run successfully. To view data drift:

Figure 2-15 Data Drift preview on Data Monitors page



Select a data monitor that has run successfully, as shown in the screenshot. On the lower pane, the data drift of the selected monitor is displayed. The X axis depicts the analysis period, and the Y axis depicts the data drift values. The horizontal dotted line is the threshold value, and the line depicts the drift value for each point in time for the analysis period. Hover your mouse over the line to view the drift values. For more information on this example, see View Data Monitor Results.

Related Topics

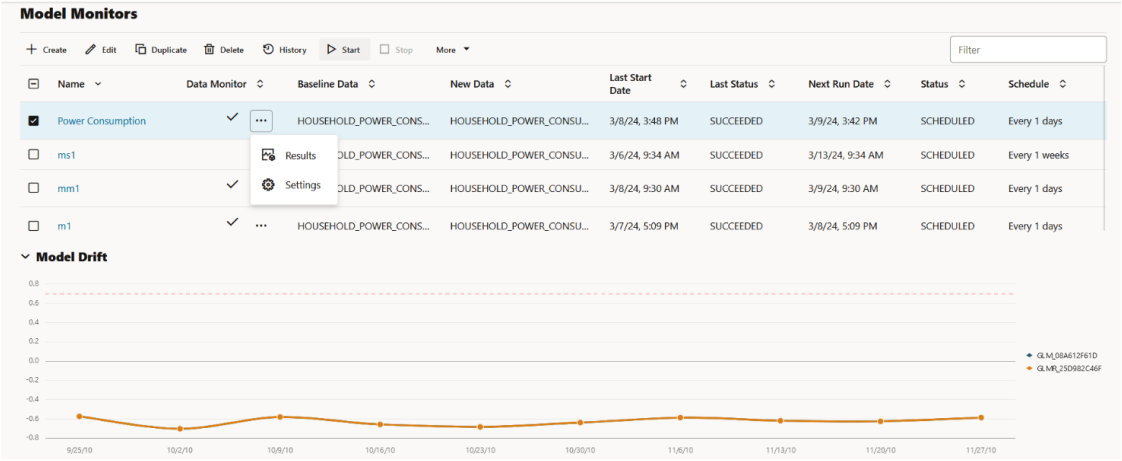
- View History
  The History page displays the runtime details of data monitors.

2.10 Model Monitoring

Model monitoring allows you to monitor the quality of model predictions over time and helps you with insights on the causes of model quality issues.

The Model Monitor page allows you to create, run, and track model monitors and results. This page lists the model monitors. You can preview the model drift by selecting a single model monitor of interest, as shown in the screenshot below. In this screenshot, the monitor **Power Consumption** is selected. On the lower pane of the Model Monitors page, the **Model Drift** for the selected monitor is displayed. The X axis depicts the analysis period, and the Y axis depicts the model drift values. The horizontal red dotted line is the threshold value. The line depicts the drift value for each point in time for the analysis period. Hover your mouse over the line to view the drift values.

Figure 2-16 Model Monitors page



You can perform the following tasks on the Model Monitors page:

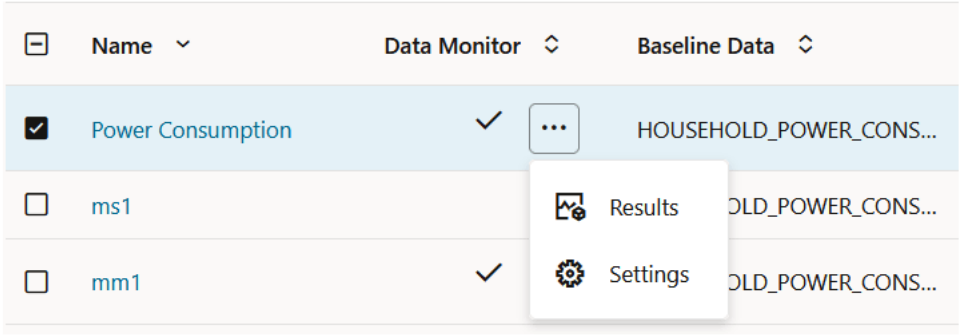
- **Create:** Create a model monitor.
- **Edit:** Select a model monitor and click **Edit** to edit the monitor.
- **Duplicate:** Select a model monitor and click **Duplicate** to create a copy of the monitor.
- **Delete:** Select a model monitor and click **Delete** to delete the model monitor.
- **History:** Select a model monitor and click **History** to view the runtime details of the model monitor. If there is a data monitor job associated with the model monitor, the data monitor runtime details are displayed on the lower pane. Click **Back to Monitors** to go back to the **Model Monitoring** page.

- **Start:** Start a model monitor.
- **Stop:** Stop a model monitor that is running.
- **More:** Select a model monitor, click **More** and then click:
  - **Enable:** Enable a model monitor schedule. By default, the model monitor is enabled. The status is displayed as `SCHEDULED`.
  - **Disable:** Disable a model monitor schedule. The status is displayed as `DISABLED`.
  - **Show Managed Monitors:** Click this option to view the models created and managed by OML Services REST API.

The Model Monitors page displays the following information about the model monitors:

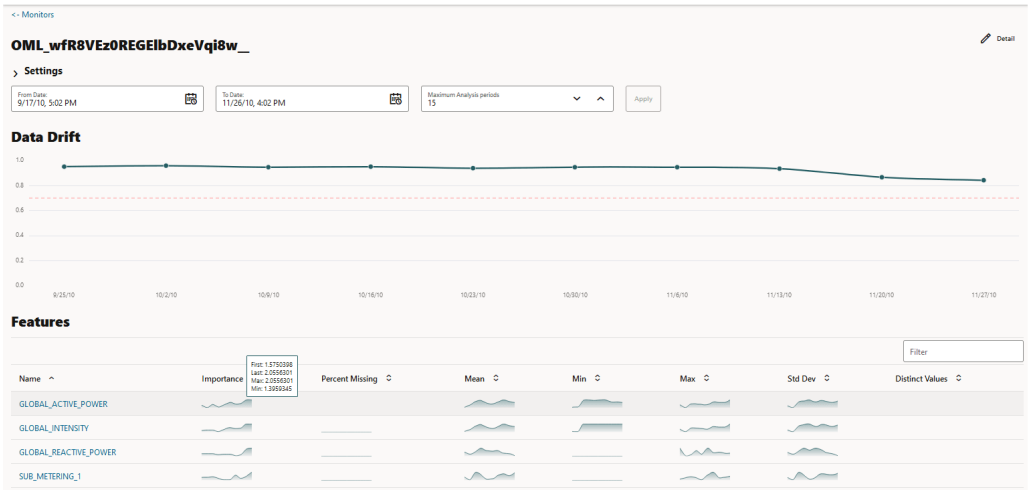
- **Name:** This is the name of the model monitor.
- **Data Monitor:** The check mark in this column indicates that there is a data monitor job associated with the model monitor. Click on the three dots to view:

Figure 2-17 View Results and Settings of Data Monitor



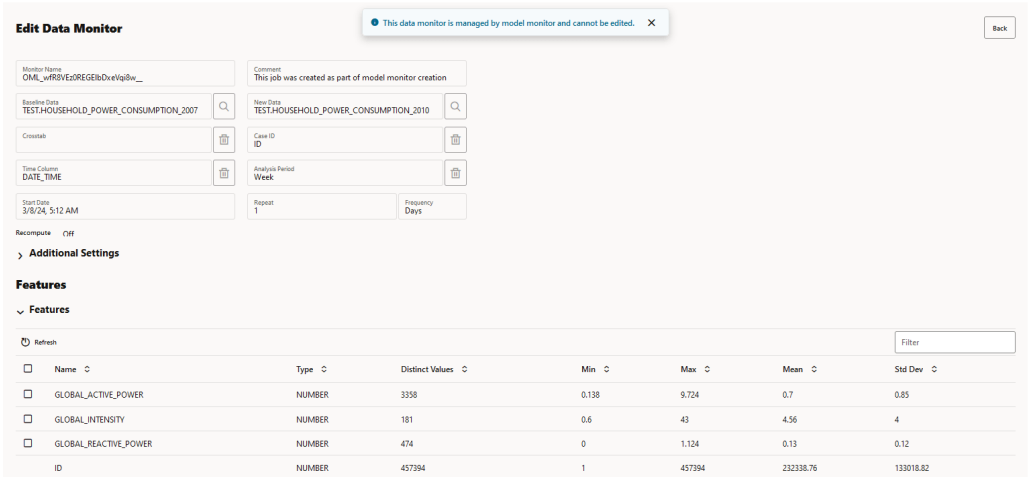
  - **Results:** Displays the data monitoring results computed by the data monitor job. The data monitor name is auto-generated along with the prefix OML. The data drift and monitored features along with the computed statistics are displayed in view-only mode on a separate page. Click **Monitors** on the top left corner to return to the Model Monitors page. Click **Details** on the top right corner to view the settings of the data monitor.

Figure 2-18 Data Monitor Results



– **Settings:** Displays the data monitor settings, additional settings, and the monitored features in view-only mode. Click **Back** on the top right corner of the page to return to the Model Monitors page.

Figure 2-19 Data Monitor Settings



- **Baseline Data:** This is a table or view that contains baseline data to monitor.
- **New Data:** This is a table or view with new data to be compared against the baseline data.
- **Last Start Date:** This is the date on which the model monitor was last started.
- **Last Status:** This is the latest status of the model monitor. The statuses are SUCCEEDED, FAILED, RUNNING, SCHEDULED.
- **Next Run Date:** This is the date on which the next run is scheduled.
- **Status:** This is the current status of the model monitor.
- **Schedule:** This is the frequency of the monitoring job, that is, how frequently the model monitor is set to run on the New Data.

2.11 REST API for Oracle Machine Learning Services

The REST API for Oracle Machine Learning Services provides REST API endpoints hosted on Oracle Autonomous AI Database. These endpoints allow you to store Machine Learning models along with its metadata, and create scoring endpoints for the model.

See REST API for Oracle Machine Learning Services

3

Project and Workspaces

A project is a logical grouping of notebooks and experiments within a workspace. A workspace is a virtual space where your projects reside, and multiple users can work on different projects. Permissions are granted on workspaces, and not on projects.

- Create Projects
  A project is a logical grouping of notebooks and experiments within a workspace. While you may own many projects, other workspaces and projects may be shared with you.
- Create and Manage Workspaces and Projects
  You can create and manage new projects and workspaces, provide access to your workspaces, manage permissions for users, and edit and delete workspaces.
- Grant Workspace Permissions
  You can collaborate with other users in Oracle Machine Learning User Interface (UI) by granting permissions to access your workspace. A workspace consists of projects. Projects consist of notebooks and experiments.

3.1 Create Projects

A project is a logical grouping of notebooks and experiments within a workspace. While you may own many projects, other workspaces and projects may be shared with you.

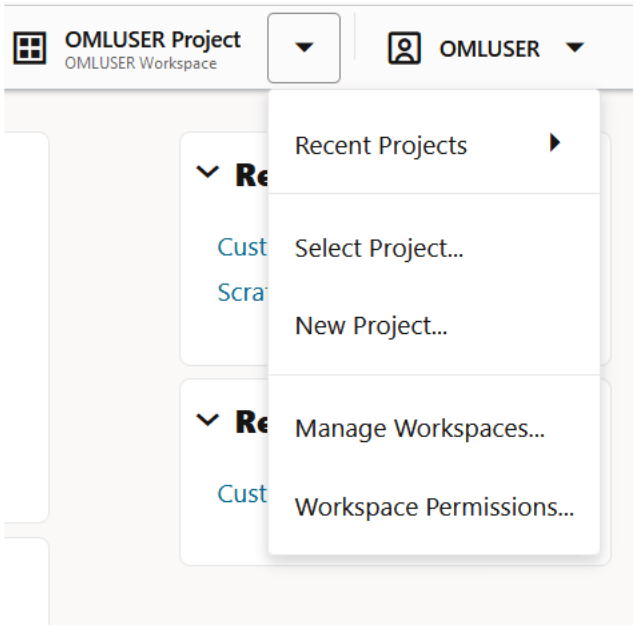
The initial workspace and the default project is created for you when you log in to Oracle Machine Learning User Interface (UI) for the first time. To create a new project:

1. On the top right corner of Oracle Machine Learning UI home page, click the project drop-down list. The project name and the workspace, in which the project resides, are displayed here. In this screenshot, the project name is `OMLUSER`, and the workspace name is also `OMLUSER`. If a default project exists, then the default project name is displayed here.

   > **Note**
   >
   > The last project that you have worked on is stored in the browser cache and is the default project. If you clear the cache, then no default exists and you must select a project.

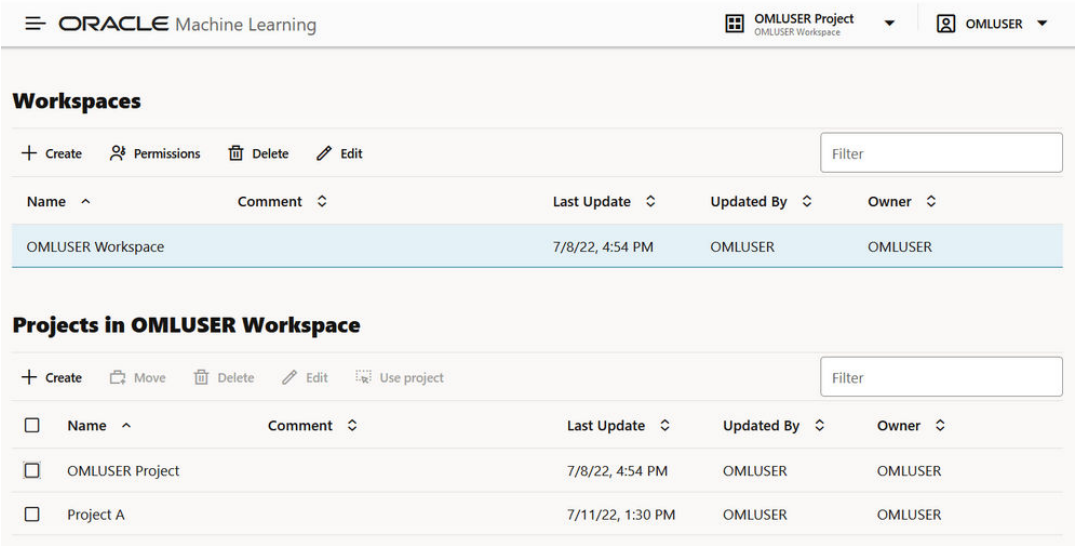
   You can create projects by using the **New Project** option: Click the down arrow next to the **Project** field and then click **New Project.** The Create Project dialog opens.

Figure 3-1 <OMLUSER> Project drop-down menu



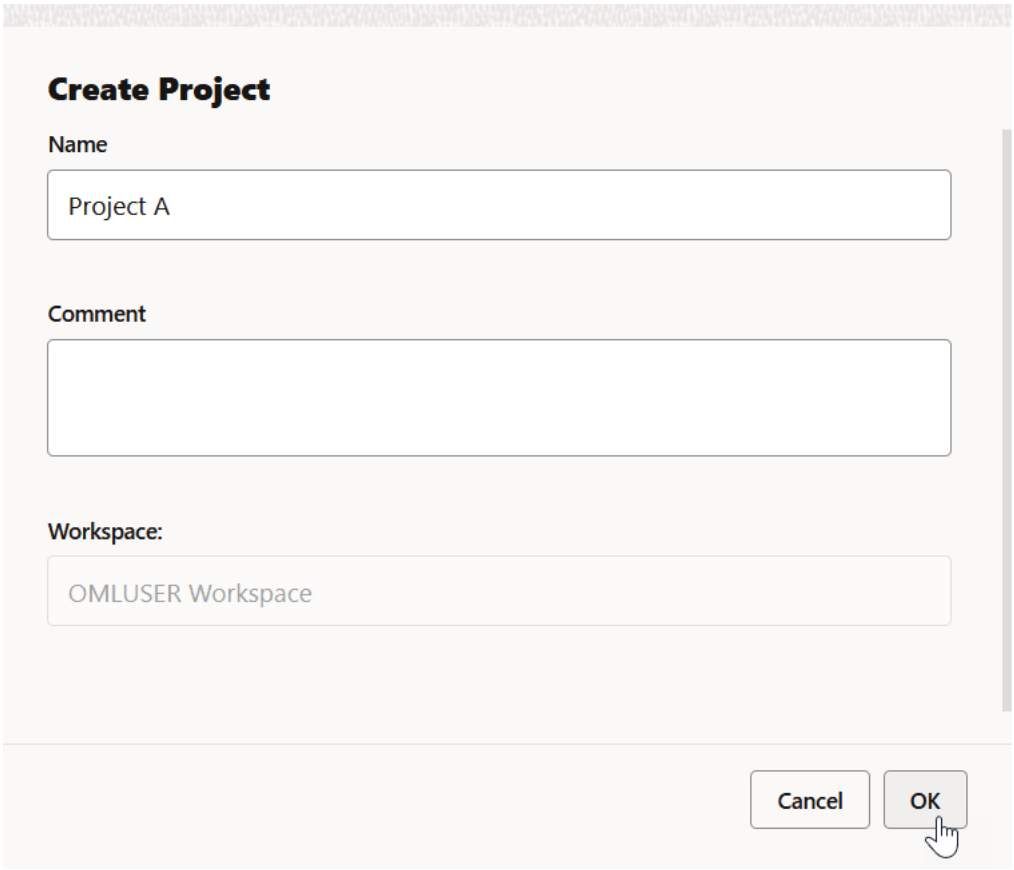
Use the **Create** option on the Workspaces page: Click the down arrow next to the **Project** field and click **Manage Workspace.** On the Manage Workspace page, under the *Projects in Workspace* section, click **Create.** The Create Project dialog opens.

Figure 3-2 Create Project in Manage Workspace page



2. In the **Create Project** dialog, enter the following:

Figure 3-3 Create Project dialog



a. **Name:** Enter a name for your project.

b. **Comments:** Enter comments, if any.

c. **Workspace:** The default workspace is selected. This is a non-editable field. To select a different workspace or to create a new workspace, go to **Manage Workspace**.

3. Click **OK.**

The project `Project A` is created and is assigned to the default workspace. In this example, the default workspace is `OMLUSER`.

Related Topics

- Create and Manage Workspaces and Projects
  You can create and manage new projects and workspaces, provide access to your workspaces, manage permissions for users, and edit and delete workspaces.

3.2 Create and Manage Workspaces and Projects

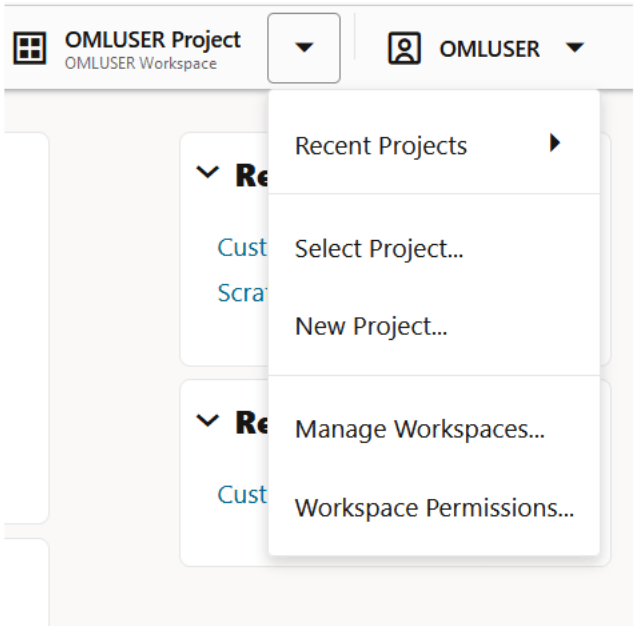
You can create and manage new projects and workspaces, provide access to your workspaces, manage permissions for users, and edit and delete workspaces.

The Workspaces page comprises two sections, one for workspaces and the other for projects.

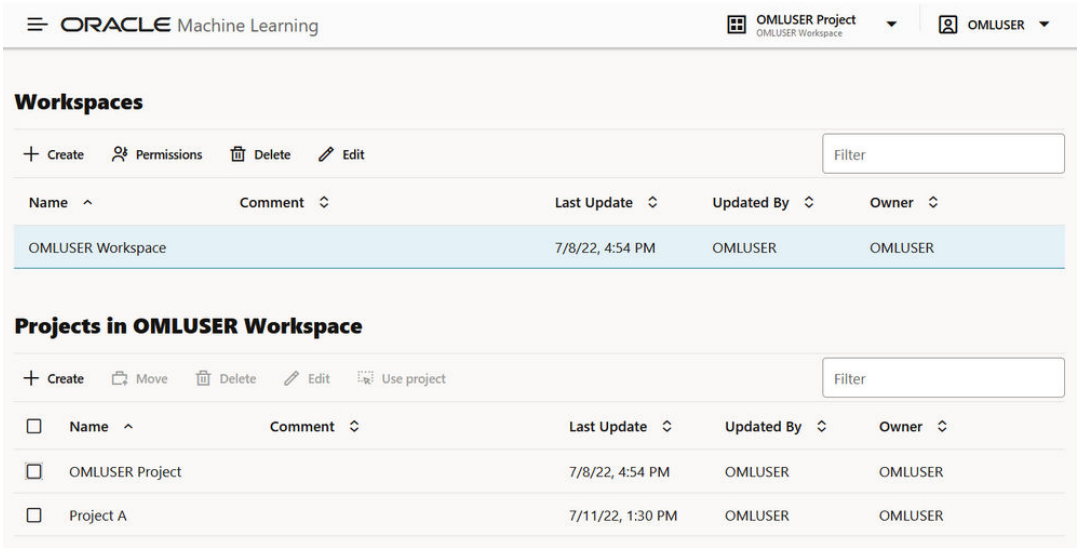
Create and Manage Workspace

To create and manage workspaces:

1. On the top right corner of your home page, click the down arrow next to the **Project** field, and click **Manage Workspaces.** The Workspaces page opens.



2. On the Workspaces page, you can create and manage workspaces and projects. To create and manage workspaces:

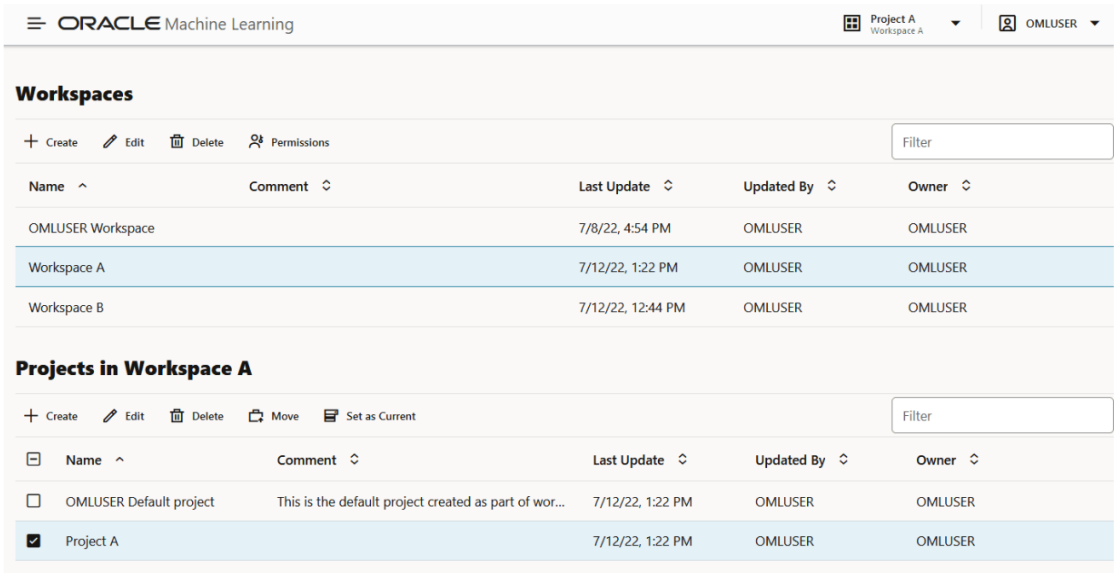


- On the upper section for workspace, click **Create** to create new workspace. In the Create Workspace dialog, enter a name for the workspace and click **OK.**
- Click **Edit** to edit the selected workspace. In the Edit Workspace dialog, you can edit the workspace name, and click **OK.**

- Click **Delete** to delete the selected workspace. Click **OK** to confirm the deletion of the selected workspace.
- Click **Permissions** to grant permissions to the selected users to access the workspace. In the Workspace Permission dialog, you can also delete users and modify their permissions.

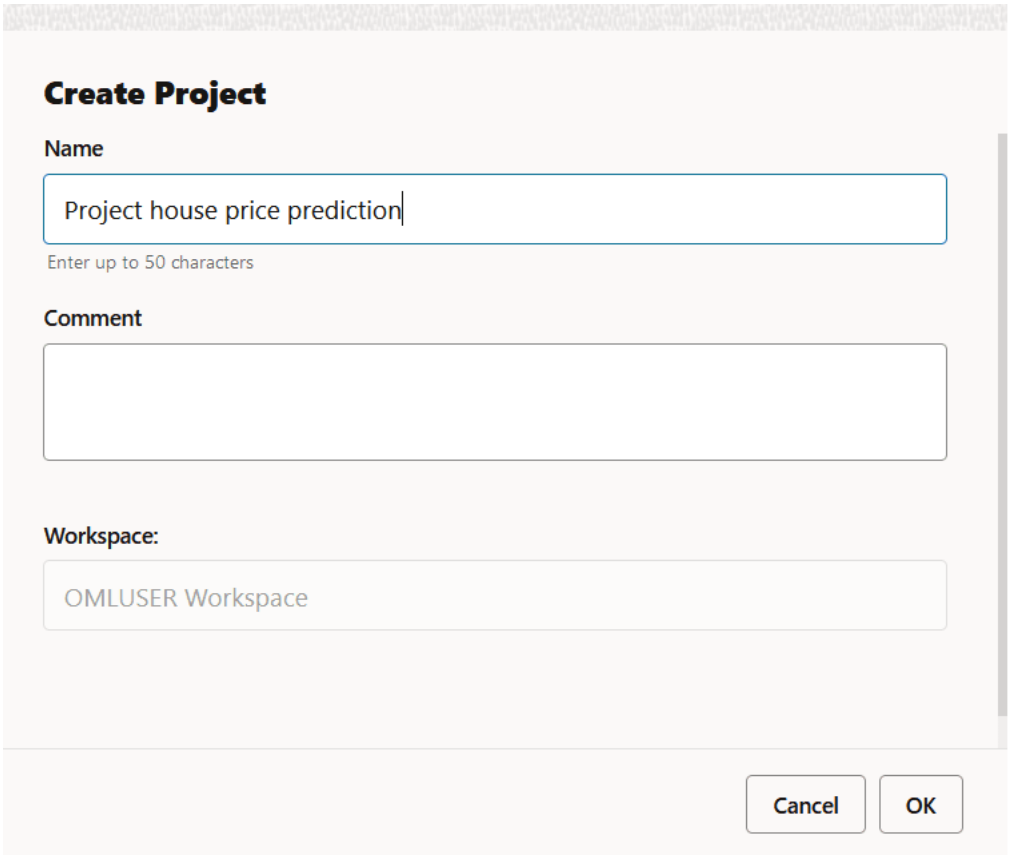
Create and Manage Projects

You can create and manage projects in the Projects in Workspaces section on the Workspaces page.



To create and manage projects:

1. On the upper section for workspace, click on the workspace in which you want to create the project.
2. On the lower section for projects, click **Create** to create a new project. The Create Project dialog opens. Enter a project name, enter comments if any, and click **OK.** Note that the Workspace field displays the workspace that you selected on the upper section.



3. Click on the project that you want to modify. The following options are enabled.
   - Click **Edit** to change the project name.
   - Click **Delete** to delete the selected project from the workspace.
   - Click **Move** to move the selected project to another workspace.
   - Click **Set as Current** to set the selected project as the default or current project.

3.3 Grant Workspace Permissions

You can collaborate with other users in Oracle Machine Learning User Interface (UI) by granting permissions to access your workspace. A workspace consists of projects. Projects consist of notebooks and experiments.

By granting different types of permissions such as **Manager, Developer,** and **Viewer,** you can allow another user to view your workspace and perform different tasks in your projects and notebooks such as edit, create, update, delete, run, view notebooks and so on. For more information about the permission types, see About Workspace Permission Types.

> ⚠ **Caution**
>
> If you grant the permission type **Manager** or **Developer,** the user can also drop tables, create tables, and run any scripts at any time on your account. The user with **Viewer** permission type can only view your notebooks and is not authorized to run or make any changes to your notebooks.

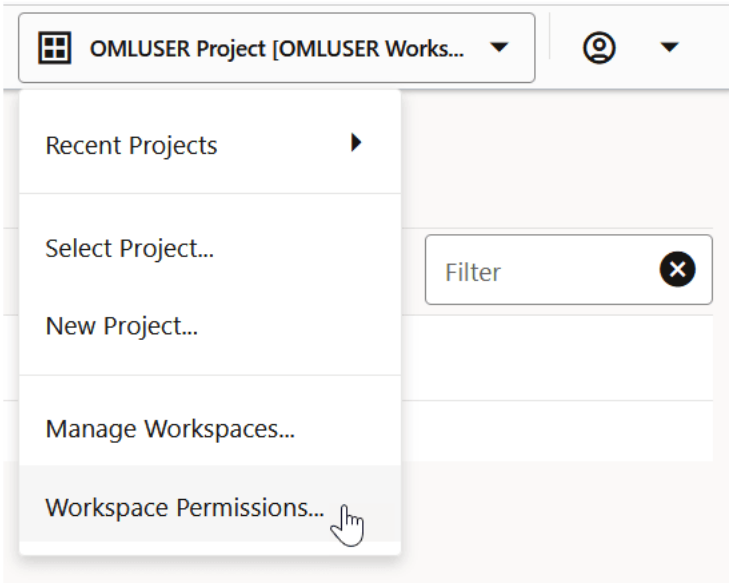
To grant permission to another user:

1. On the top right corner of Oracle Machine Learning UI home page, click the project and workspace drop-down list and select **Workspace Permission.**

   The Permissions dialog box opens.

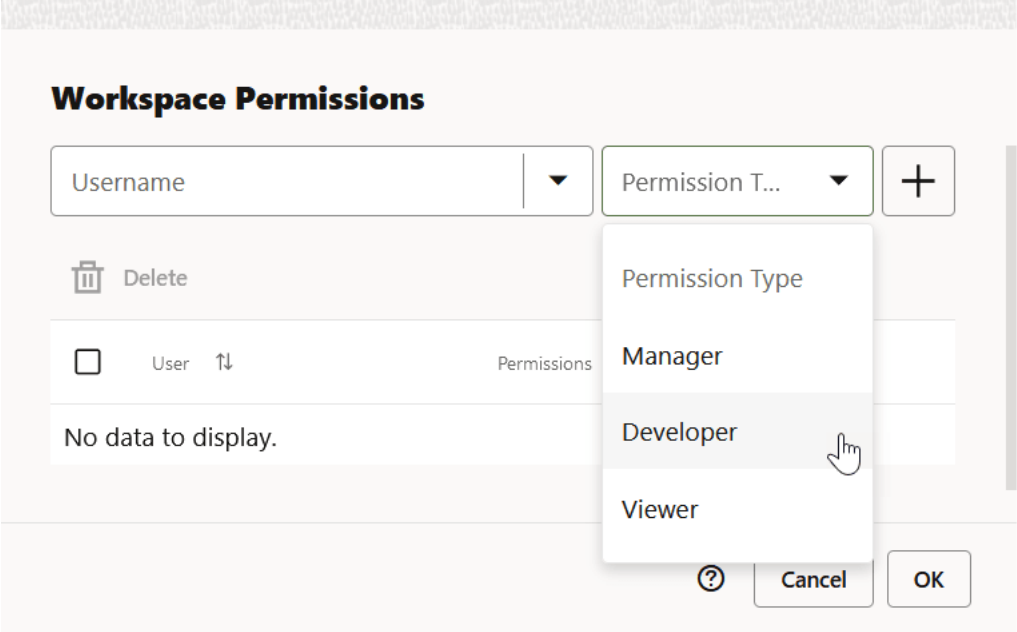
2. In the Permissions dialog box, select a user from the **Username** drop-down list.

Figure 3-4 OMLUSER Project drop-down menu



3. In the **Permissions** drop-down list, select the permission type you want to grant to the selected user.
   - **Manager**
   - **Developer**
   - **Viewer**

Figure 3-5 Workspace Permissions dialog



4. Click **Add.**

   The selected user is granted the assigned permission. The user name is displayed in the Permissions dialog box, along with the permission type.

5. Click **OK.** This completes the task of granting permission to a user. To delete a user and the associated permission, select the user and click **Delete.**

- About Workspace Permission Types
  Oracle Machine Learning UI allows three types of permissions. Depending on the permission type, you can allow the user to view or perform different tasks in your workspace, projects, and notebooks.

3.3.1 About Workspace Permission Types

Oracle Machine Learning UI allows three types of permissions. Depending on the permission type, you can allow the user to view or perform different tasks in your workspace, projects, and notebooks.

The permissions are privileges granted to shared objects in a given workspace. Here, operations that are allowed on shared objects, for example notebooks, are discussed. For instance, the user to whom the `Developer` permission is granted can only view scheduled jobs on shared notebooks; he cannot run these jobs. However, the user with `Developer` permission can always view and run scheduled jobs on notebooks that are created by him. The three types of permissions are listed in the following table along with the actions that are allowed.

> **Note**
>
> Permissions are granted on workspaces, and not on projects.

Permission Types	Actions based on permission
Manager	• **Project:** Create, update, view and delete. • **Workspace:** View only. • **Notebooks:** Create, update, duplicate, run, export, import, and delete. **Jobs:** Schedule and run jobs.
Developer	• **Project:** View only. • **Workspace:** View only. • **Notebooks:** Create, update, run, duplicate, import, export, and delete notebooks. **Note** A user with `Developer` permission can update, run, duplicate, import, export, and delete only those notebooks that were created by the user. • **Jobs:** View and run jobs of shared notebooks only. **Note** A user with `Developer` permission cannot create jobs for notebooks that are shared.
Viewer	• **Project:** View only. • **Workspace:** View only. • **Notebooks:** View only. • **Jobs:** View jobs and job runs of shared notebooks only.

For tasks that can be performed by an Administrator, see: Oracle Machine Learning Administration Tasks

4

OML Notebooks

Oracle Machine Learning Notebooks is an enhanced web-based notebook platform for data analyst and data scientists. You can write code, text, create visualizations, and perform data analytics including machine learning. Notebooks work with interpreters in the back-end. In Oracle Machine Learning user interface, notebooks are available in a project, where you can create, edit, delete, copy, move, and even save notebooks as templates.

- Oracle Machine Learning Notebooks
  Oracle Machine Learning Notebooks is an enhanced web-based notebook platform for data engineers, data analyst, R and Python users, and data scientists. You can write code, text, create visualizations, and perform data analytics including machine learning. Notebooks work with interpreters in the back-end.
- GitHub Notebooks
  Oracle Machine Learning UI supports direct interaction of Oracle Machine Learning Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories.
- Enable GPU Compute Capabilities in a Notebook through the Python Interpreter
  This topic demonstrates how to enable GPU compute capabilities in a notebook through the Python interpreter. It also shows how to get information about the current GPU on which the notebook is running, and other details.
- Visualize your Data in Oracle Machine Learning Notebooks
  Oracle Machine Learning Notebooks offer rich visualization capabilities of your data. The visualizations depend on the type of your dataset.
- Export a Notebook
  You can export a Notebook in Native format (`.dsnb`) file, Zeppelin format (`.json`) file, in Jupyter format (`.ipynb`), and later import them in to the same or a different environment.
- Import a Notebook
  You can import notebooks into your Oracle Machine Learning UI projects. Oracle Machine Learning UI supports the import of notebooks in the native format (`.dsnb`), Zeppelin (`.json`) and Jupyter (`.ipynb`) notebooks. Oracle Machine Learning UI also supports importing notebooks from your GitHub repository.
- About Interpreters and Notebook Service Levels
  An interpreter is a plug-in that allows you to use a specific data processing language backend.
- Run Notebooks using different interpreters
  An interpreter is a plug-in that allows you to use a specific data processing language backend. You select an interpreter by inserting its directive at the beginning of a notebook paragraph.
- Use the Scratchpad
  The Scratchpad provides you convenient one-click access to a notebook for running SQL statements, PL/SQL, R, and Python scripts that can be renamed. The Scratchpad is available on the Oracle Machine Learning User Interface (UI) home page.

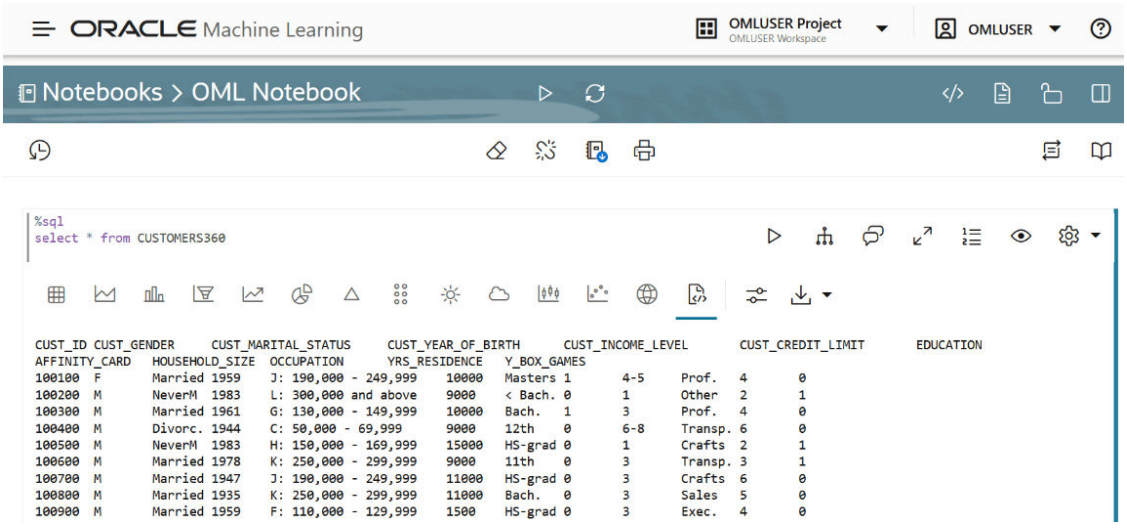
4.1 Oracle Machine Learning Notebooks

Oracle Machine Learning Notebooks is an enhanced web-based notebook platform for data engineers, data analyst, R and Python users, and data scientists. You can write code, text, create visualizations, and perform data analytics including machine learning. Notebooks work with interpreters in the back-end.

In Oracle Machine Learning, notebooks are available in a project within a workspace, where you can create, edit, delete, copy, move, and even save notebooks as templates. A notebook can contain many paragraphs. A paragraph is a notebook component where you can write and run SQL statements, PL/SQL scripts, R and Python code, and conda instructions. You can run paragraphs individually or, run all the paragraphs in a notebook using a single button. A paragraph has an input section and an output section. In the input section, specify the interpreter to run along with the code or text. This information is sent to the interpreter to be run. In the output section, the results of the interpreter are provided.

> **Note**
>
> There is a single namespace for both the original notebooks and the new notebooks. You cannot have a notebook with the same name in both notebook lists. A notebook copied from the original interface to the new will have `_new` appended to it.

Figure 4-1 OML Notebook



The Oracle Machine Learning Notebooks provides:

- Faster notebook loading time.
- The Oracle look and feel as it based on the Oracle Redwood theme.
- Enriched visualization in its Line chart, Area chart, Bar chart, Pyramid chart, Pie chart, Donut chart, Funnel chart, Tag Cloud, Treemap Diagram, Sunburst Diagram, Scatter Plot, Box Plot.
- Option to enter comments in notebook paragraphs.

- Option to create Paragraph Dependencies. The Paragraph Dependencies feature allows you to add dependencies between paragraphs. The dependents of a paragraph automatically run after the original paragraph is run.
- Simplified service level selection of High, Medium, Low, and GPU through drop-down menu.
- Layout of Zeppelin and Jupyter notebook.
- On-page versioning, viewing of version history, and version comparison.
- Access your Oracle Machine Learning Notebooks
  You can access the **OML Notebooks** page from the Oracle Machine Learning UI home page and the left navigation pane. You can access the **GitHub Notebooks** page from the Oracle Machine Learning UI home page and from the user profile drop-down menu.
- Manage Oracle Machine Learning Notebooks
  The Notebooks page lists all the notebooks—both Oracle Machine Learning notebooks and GitHub notebooks.
- Create a Notebook
  An OML Notebook is a web-based interface for data analysis, data discovery, data visualization and collaboration.
- Edit your Oracle Machine Learning Notebook
  Upon creating a notebook, it opens automatically, presenting you with a single paragraph using the default %sql interpreter. You can change the interpreter by explicitly specifying one of `%script`, `%python`, `%sql` , `%r` , `%md`, or `%conda`.

4.1.1 Access your Oracle Machine Learning Notebooks

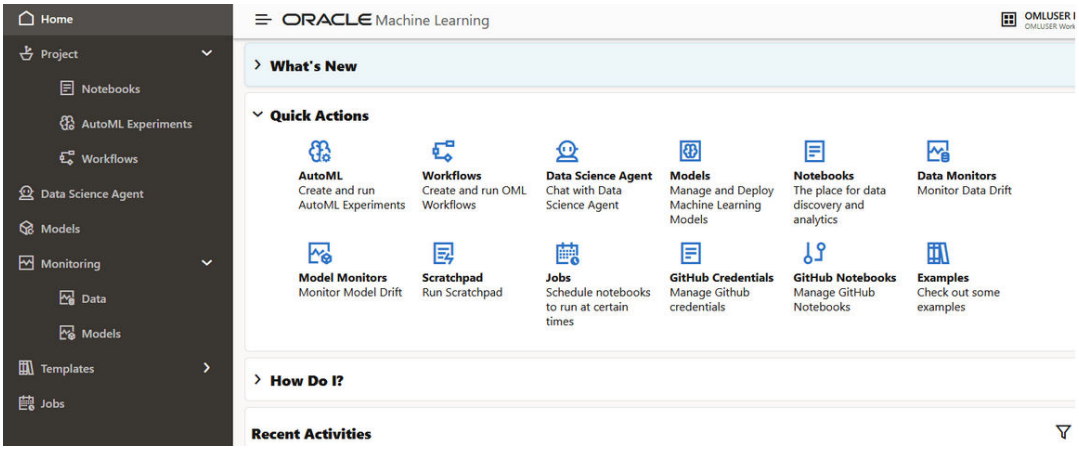
You can access the **OML Notebooks** page from the Oracle Machine Learning UI home page and the left navigation pane. You can access the **GitHub Notebooks** page from the Oracle Machine Learning UI home page and from the user profile drop-down menu.

Accessing the Oracle Machine Learning Notebooks page

To access the OML Notebooks page:

- Go to the Oracle Machine Learning left navigation pane, expand **Projects** and then click **Notebooks**. Alternatively, you can click **Notebooks** under Quick Link on the home page to open the Notebooks page.

Figure 4-2 Notebooks option on the left navigation pane and OML UI home page



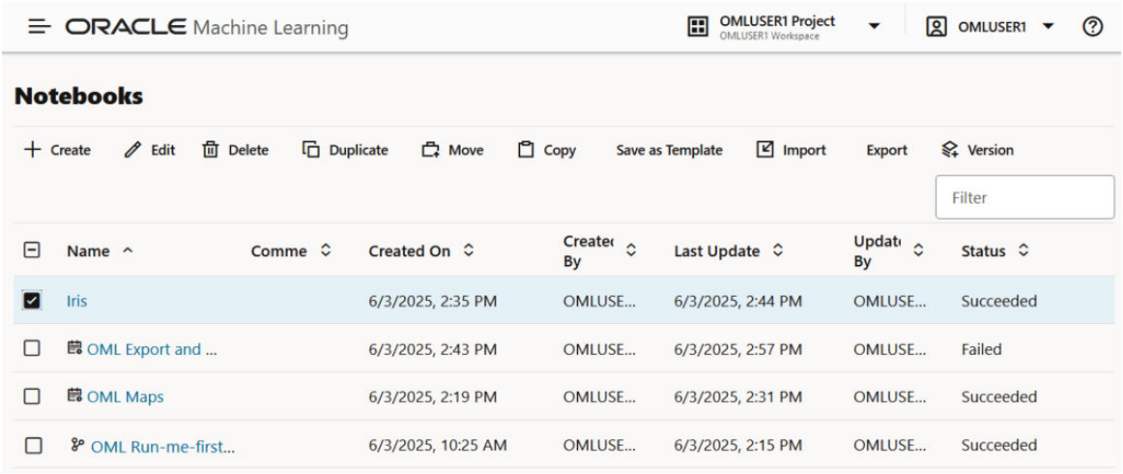
4.1.2 Manage Oracle Machine Learning Notebooks

The Notebooks page lists all the notebooks—both Oracle Machine Learning notebooks and GitHub notebooks.

This is the Oracle Machine Learning Notebooks page. Selecting a notebook enables all the options on this page.

> **Note**
>
> The **Version** and **Edit** options are not enabled for GitHub notebooks. For GitHub notebooks, open and edit the notebook in the editor, and use the Version Control option to commit the changes to the GitHub repository.

Figure 4-3 Notebooks page



Note that some notebooks displayed in the screenshot here have different icons displayed.

- The icon indicates that the notebook is imported from a GitHub repository.
- The icon indicates that the notebook has a job associated with it.

> **Note**
>
> A GitHub notebook can also have a job associated with it.

You can perform the following tasks:

- **Create:** Click **Create** to create a new notebook.
- **Edit:** Click on a notebook row to select it and click **Edit.** You can edit the notebook name, and add comments in the Edit Notebook dialog box.

> **Note**
>
> The **Edit** option is not enabled for GitHub notebooks.

- **Delete:** Click on a notebook row to select it and click **Delete.**
- **Duplicate:** Click on a notebook row to select it , and click **Duplicate.** This creates a copy of a notebook, and the duplicate copy of the is listed on the Notebooks page with the suffix `_1` in the notebook name.
- **Move:** To move notebooks from one project to another project in the same or different workspace.
- **Copy:** To copy notebooks to another project in the same or different workspace. When you copy notebooks, the a copy of the notebook remains in the original project.
- **Save as Template:** To save a notebook as a template, select the notebook and click **Save as Template.** In the Save as Template dialog, you can define the location of the template to save it in Personal or Shared under Templates.
- **Import:** To import a notebook, click **Import**. one of the following options:
  - **File:** To import a notebook as a `.json` file
  - **Git:** To import a notebook from a remote GitHub repository. This clones the notebook.
- **Export:** To export a notebook, click **Export.** You can export Notebooks in the `.dsnb` format, Zeppelin format (`.json`) file and in Jupyter format (`.ipynb`), and later import them in to the same or a different environment.
- **Version:** To create versions of a notebook, select it and click **Version**. The Versions page for that particular notebook opens. Here, you can create a new version of the notebook by clicking **+Version**. The Create Version dialog opens. Enter a name of the notebook version, a description, and click **OK.** The new version of the notebook gets created by the same name with a suffix _2 for the second version. For subsequent versions, suffix (number) increments by one. To revert to an older version by clicking **Revert Version**. You also have the option to delete any version of the notebook. Click **Back to Notebooks** to go to the OML Notebooks page.

> **Note**
>
> The **Version** option is not enabled for GitHub notebooks.

> **Note**
>
> You can also version a notebook by opening it, and then clicking on the  option. By using this option, you can create new versions, view version history, restore older versions, and delete any older versions of the notebook that you have opened.

For each notebook listed on this page, the following information is displayed:

- Name: This is the name of the notebook.
- Comment: This is the comment you entered when creating the notebook.
- Created on: This is the date on which the notebook was first created.

- Created by: This is the name of the user who created the notebook.
- Last Update: This is the date on which the notebook was last updated.
- Last Updated By: This is the name of the user who last updated the notebook.
- Status: The following statuses are available for any notebook:
  - Created—Indicates that the notebook has been created but not yet run
  - Running—Indicates that the notebook is currently running
  - Succeeded—Indicates that the notebook has run successfully without any errors
  - Stopped—Indicates that the notebook run has been canceled while it was running
  - Failed—Indicates that the notebook run has failed or has being rejected

4.1.3 Create a Notebook

An OML Notebook is a web-based interface for data analysis, data discovery, data visualization and collaboration.

A Notebook comprises paragraphs which is a notebook component where you can write and run SQL statements, PL/SQL scripts, Python, R, Conda and markdown commands. A paragraph has an input section and an output section. In the input section, specify the interpreter to run along with the text. This information is sent to the interpreter to be run. In the output section, the results are provided. A Notebook has the interpreter settings specification. It contains the internal list of bindings that determines the order of the interpreter bindings.

To create a Notebook:

1. On the Oracle Machine Learning UI home page, click **Notebooks.** The Notebooks page opens.
2. On the Notebooks page, click **Create.**
3. In the **Name** field, provide a name for the notebook.
4. In the **Comments** field, enter comments, if any.
5. Click **OK.**

Your OML notebook is created and it opens in the notebook editor. You can now use it to run SQL statements, PL/SQL scripts, Python, R, Conda and markdown commands. To do so, specify any one of the following directives in the input section of the paragraph:

- `%sql`: To connect to the SQL interpreter and run SQL statements
- `%script`: To connect to the PL/SQL interpreter and run PL/SQL scripts
- `%md`: To connect to the Markdown interpreter and generate static html from Markdown plain text
- `%python`: To connect to the Python interpreter and run Python scripts
- `%r`: To connect to the R interpreter and run R scripts.
- `%conda`: To connect to the Conda interpreter, and install third-party Python and R libraries inside a notebook session.

4.1.4 Edit your Oracle Machine Learning Notebook

Upon creating a notebook, it opens automatically, presenting you with a single paragraph using the default %sql interpreter. You can change the interpreter by explicitly specifying one of `%script`, `%python`, `%sql` , `%r` , `%md`, or `%conda`.

Set the context with a project with which your notebook is associated. You can edit an existing notebook in your project. To edit an existing notebook:

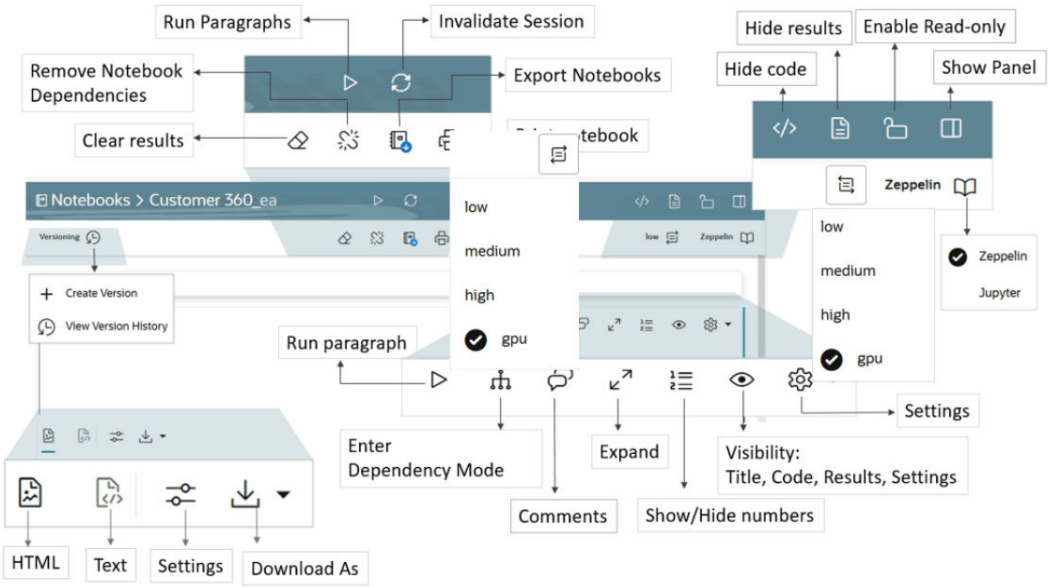
1. On the Oracle Machine Learning home page, select the project in which your notebook is available.

> **Note**
>
> A project is a logical grouping of notebooks and experiments within a workspace. While you may own many projects, other workspaces and projects may be shared with you.

2. Click the notebook that you want to open and edit.

   The selected notebook opens in edit mode.

3. In the edit mode, you can use the following Oracle Machine Learning Notebook toolbar options:



Notebook level edit options:

- Click  to run all paragraphs in the notebook
- Click  to invalidate and reset the notebook session.
- Click  to create a new version this notebook, or to view the earlier versions of the notebook.

  - **Create Version:** Click this option to create a new version of this notebook. You have the option to provide a new name for the version, and a description about it. When you create a new notebook version, the paragraph results of each run session are stored in the versioned notebook. When you restore a notebook, the paragraph results of each run session are also restored. You can also create notebook versions from the editor. For more information, see Work with Notebook Versions in the Notebook Editor
  - **View Version History:** Click this option to view the earlier versions of the notebook. You have the option to restore any earlier version, compare versions, and delete any earlier version that you created.

> **Note**
>
> You can also create notebook versions, view version history, and delete older versions from the 
>
> option on the Notebooks page.

- Click  to clear paragraph results.
- Click  to clear paragraph dependencies.
- Click  to export the notebook. You can export the notebook as a `.dsnb` file, `.zpln` file (Zeppelin notebook) and `.ipynb` file (Jupyter notebook). You have these settings while exporting a notebook:
  - Export All
  - Exclude code
  - Exclude results
  - Exclude Timestamp
- Click  to print the notebook
- Click  to hide the code of all paragraphs in the notebook
- Click  to hide the results of all the paragraph in the notebook
- Click  to enable read-only mode for this notebook.

> **Note**
>
> The read-only mode is available only for the Oracle Machine Learning Notebook.

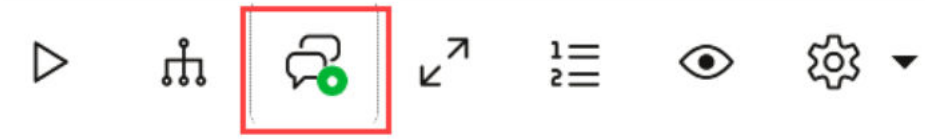
- Click  to show the edit panel. The edit options in the panel are the same edit options available for the paragraph. Clicking the panel icon opens the edit pane on the right, and the edit tool bar in the paragraph is hidden.
- Click  to change the notebook service level to either low, medium, high, or GPU.
- Click  to switch the OML notebook to either Zeppelin or Jupyter notebook

Paragraph level edit options:

- Click  to run the selected paragraph
- Click  to enter into Dependency Mode. In Dependency Mode, you must select and deselect paragraphs in order to add or remove them as dependents.

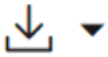
> **Note**
>
> The Paragraph Dependencies feature allows you to add dependencies between paragraphs. The dependents of a paragraph automatically run after the original paragraph is run.

- Click  to open the Comments dialog. Type in your comments here, and press **Enter** to add the comment. You can also delete any comments by clicking the corresponding **Delete** icon. Click the comments icon to close the dialog. You can provide comments for each paragraph in a notebook. Paragraphs with comments are indicated by a green dot on the comments icon.



- Click  to view the notebook paragraph in full screen mode. To view the paragraph in the normal mode, click the collapse icon.
- Click  to show line numbers in the notebook paragraph.
- Click  to view the paragraph title, code, results, and the paragraph settings.
- Click  to :
  - **Move up:** Click to move the paragraph up in the notebook.
  - **Move down:** Click to move the paragraph down in the notebook.

  - **Clear results:** Click to clear the results of the commands that you ran in the paragraph.
  - **Open as Embedded Window:** Click to view the current paragraph separately in your browser.
  - **Clone Paragraph:** Click to clone the paragraph. The paragraph is cloned in the same notebook.
  - **Disable Run:** Click to disable running of the paragraph. To enable run, go to **Settings** and click **Enable Run**.
  - **Delete Paragraph:** Click to delete the paragraph.

- Click  to view the paragraph in HTML format
- Click  to view the paragraph in text format
- Click  to adjust settings of the notebook paragraph output. This setting is specifically applicable to visualizations in graphs, charts etc.
- Click  to download the paragraph as a text file, or as .png or .svg files, as applicable, for paragraphs that contains graphs or charts as output.

- Work with Notebook Versions in the Notebook Editor
  By creating versions of your notebook, you can archive your work in a notebook.
- Create Paragraph Dependencies
  Paragraph Dependencies allow you to add dependencies between paragraphs. The dependent paragraphs automatically run after the original paragraph is run, according to the order of dependency.
- Work with Notebook Versions on the Notebooks Page
  By creating versions of your notebook, you can archive your work in a notebook.

4.1.4.1 Work with Notebook Versions in the Notebook Editor

By creating versions of your notebook, you can archive your work in a notebook.

You can create versions of notebooks in the Notebooks editor, as well as on the Notebooks page. In this example:

- The original notebook *Notebook Versioning Demo*, is edited to add a script to build a machine learning model.
- The *Notebook Versioning Demo* notebook is then versioned as **Version 2** to archive the code to build the machine learning model.
- The **Version 2** and **Version 1** of the **Notebook Versioning Demo** notebook are compared using the **Compare Versions** feature.

> **Note**
>
> A versioned notebook is non-editable. If you want to make any changes to a particular version of a notebook, you must restore that version to edit it.

To create a new notebook version and view version history:

1. On the Notebooks EA page, click on the **Notebook Versioning Demo** notebook to open it in the notebook editor.

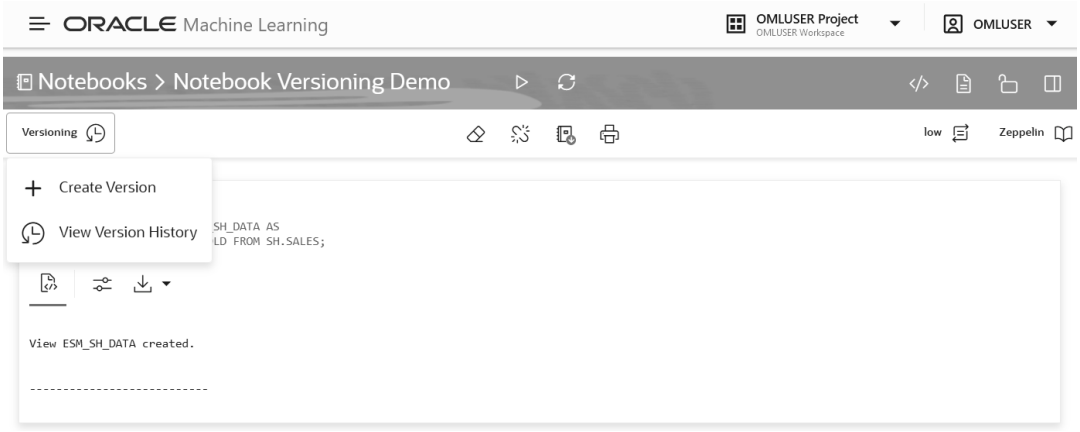
> **Note**
>
> Version 1 of this notebook is already created as part of the example in Work with Notebook Versions on the Notebooks Page. It contains the archived code to create the view `ESM_SH_DATA`, count the record, and view the data. Clicking on the notebook opens the original version which is editable.

2. Now, edit the notebook to add a script to build a machine learning model. On the notebook, hover your cursor over the lower border of the third paragraph, and click the + icon to add a new paragraph.

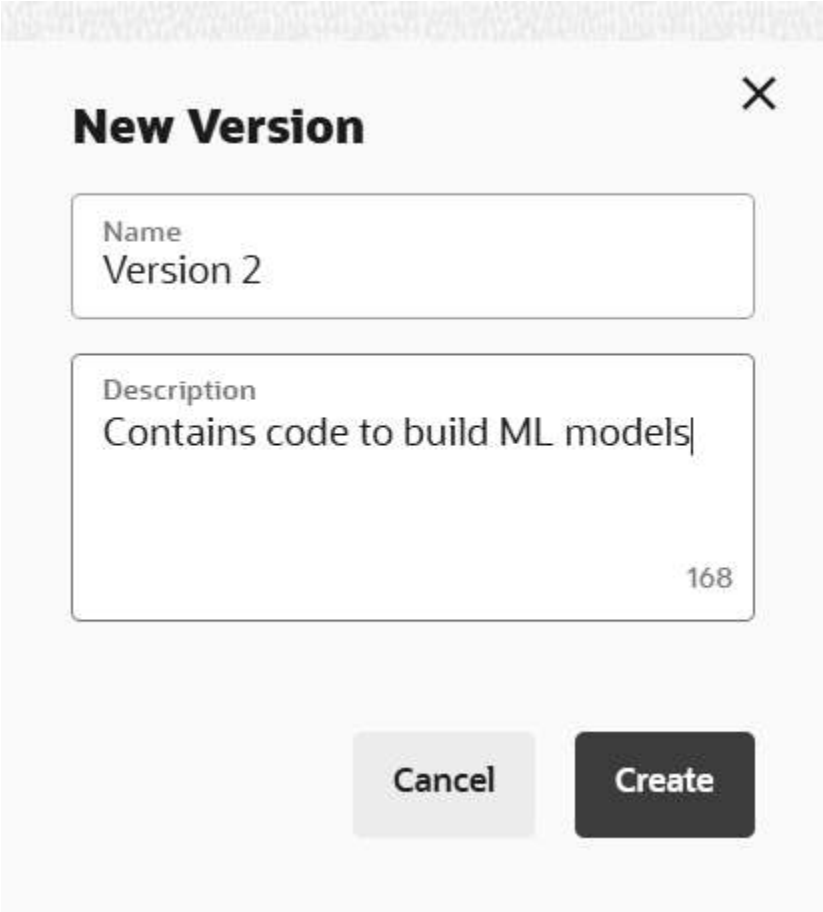
3. Copy and paste the following script to the new paragraph. This script builds a machine learning model using the ESM algorithm.

```
%script

BEGIN DBMS_DATA_MINING.DROP_MODEL('ESM_SALES_FORECAST_1');
EXCEPTION WHEN OTHERS THEN NULL; END;
/
DECLARE
    v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN

    v_setlst('ALGO_NAME')            := 'ALGO_EXPONENTIAL_SMOOTHING';
    v_setlst('EXSM_INTERVAL')        := 'EXSM_INTERVAL_QTR'; --
accumulation int'l = quarter
    v_setlst('EXSM_PREDICTION_STEP') := '4';                 -- prediction
step = 4 quarters
    v_setlst('EXSM_MODEL')           := 'EXSM_WINTERS';      -- ESM model
= Holt-Winters
    v_setlst('EXSM_SEASONALITY')     := '4';                 -- seasonal
cycle = 4 quarters

    DBMS_DATA_MINING.CREATE_MODEL2(
        MODEL_NAME          => 'ESM_SALES_FORECAST_1',
        MINING_FUNCTION     => 'TIME_SERIES',
        DATA_QUERY          => 'select * from ESM_SH_DATA',
        SET_LIST            => v_setlst,
        CASE_ID_COLUMN_NAME => 'TIME_ID',
        TARGET_COLUMN_NAME  =>'AMOUNT_SOLD');
END;
```

4. Now, archive this notebook along with the code to build the machine learning model by versioning it. On the top left corner of the notebook editor, click **Versioning** 
5. The options to **Create Version** and **View Version History** opens.

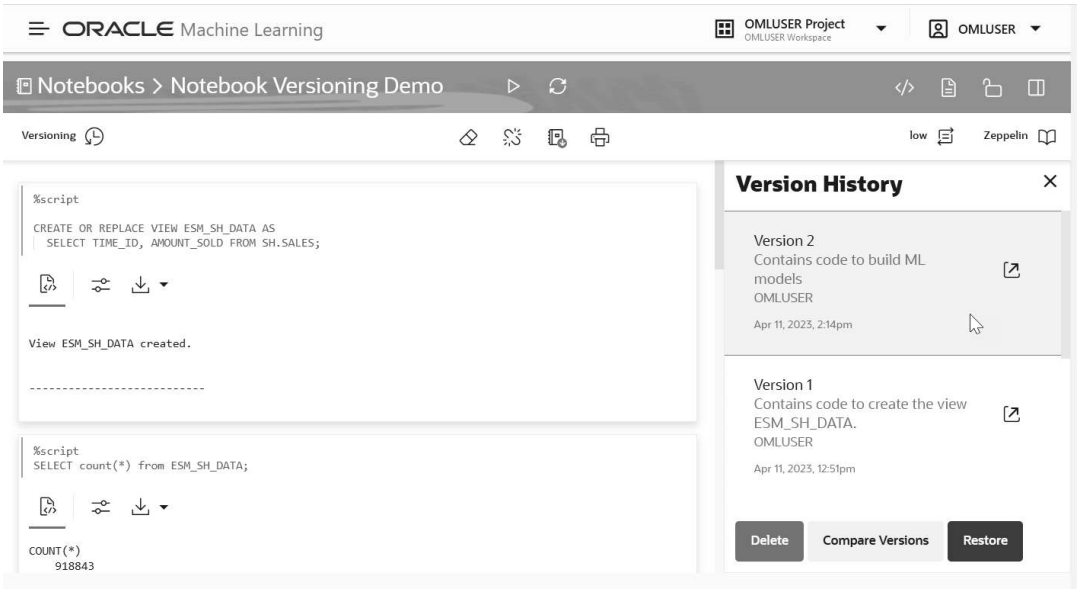


6. Click **Create Version**. The New Version dialog opens.
7. In the **New Version** dialog:
   a. **Name:** Here, the name `Version 2` is taken by default. Let's retain this name.
   b. **Comments:** Enter comments, if any.
   c. Click **Create.** A message is displayed confirming the creation of the new version.

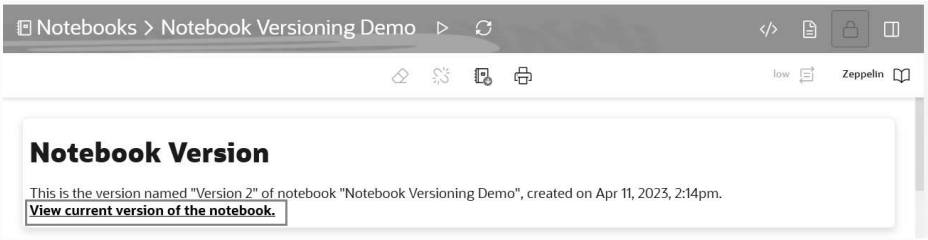


Version 2 of this notebook now contains the archived code to create the machine learning model.

8. To view the version that you just created, click **Versioning** , and then click **View Version History**.
9. On the right pane of the notebook editor, the **Version History** pane opens. Hover your cursor over any notebook version and click on it to enable the available options.
10. You can perform the following tasks in the Version History pane:



- Click  to open the selected version.

> **Note**
>
> Clicking on any versioned notebook opens the notebook in read-only mode, as versioned notebooks are non-editable. To view the current editable version, click **View current version of the notebook**.
>
> 
>

- Click **Delete** to delete the selected version.
- Click **Compare Versions** to compare the selected and current version of the notebook. You can select other available versions from the drop-down list. In this example, Version 2 of the notebook (under Current State) is compared with Version 1. New additions are highlighted in green, as shown in the screenshot here, and deletions are highlighted in red.

- Click **Restore** to restore the selected version.

> **Note**
>
> Restoring a selected version of the notebook will discard all the unversioned changes, if any.

4.1.4.2 Create Paragraph Dependencies

Paragraph Dependencies allow you to add dependencies between paragraphs. The dependent paragraphs automatically run after the original paragraph is run, according to the order of dependency.

To create paragraph dependencies:

1. On the Notebooks page, click **Create Notebooks**.
2. In the Create Notebooks dialog, enter the name `Paragraph Dependencies Demo` in the **Name** field and click **OK**.

   The notebook is created and it opens in the notebook editor.
3. On the notebook, hover your cursor over the lower border of the paragraph and click the + icon to add a paragraph. Add two more paragraphs to this notebook, and paste the following PL/SQL script in the paragraphs:

   a. In the first paragraph, copy and paste the following PL/SQL script. This script creates the view `ESM_SH_DATA` from the `SALES` table present in the `SH` schema.

```
%script
```

```
CREATE OR REPLACE VIEW ESM_SH_DATA AS
  SELECT TIME_ID, AMOUNT_SOLD FROM SH.SALES;
```

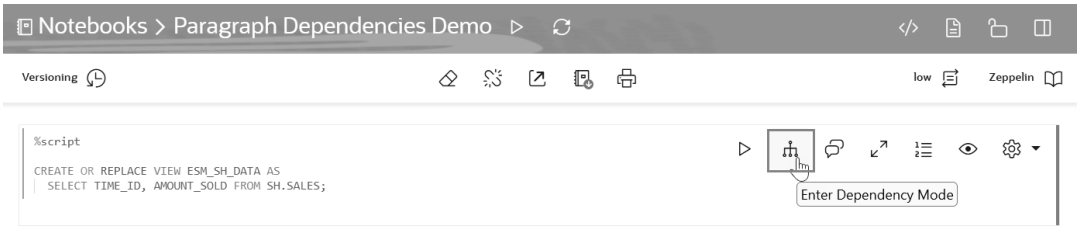
b. In the second paragraph, copy and paste the following SQL script. This script gives a count of the record present in the view `ESM_SH_DATA`.

```
%script
SELECT COUNT(*) FROM ESM_SH_DATA;
```

c. In the third paragraph, copy and paste the following SQL script to review the data in a tabular format.

```
%sql
SELECT * FROM ESM_SH_DATA
FETCH FIRST 10 ROWS ONLY;
```

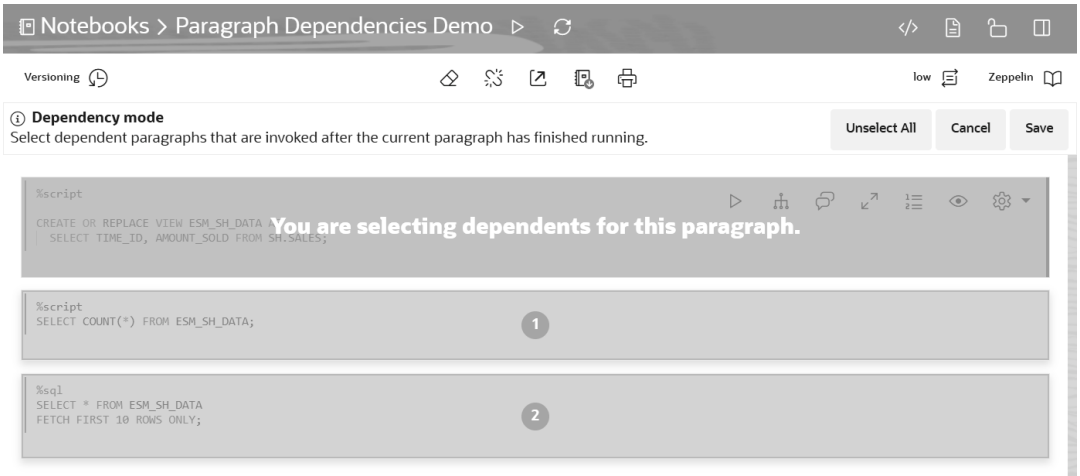
4. Go to the first paragraph and click on the **Enter Dependency Mode** icon.



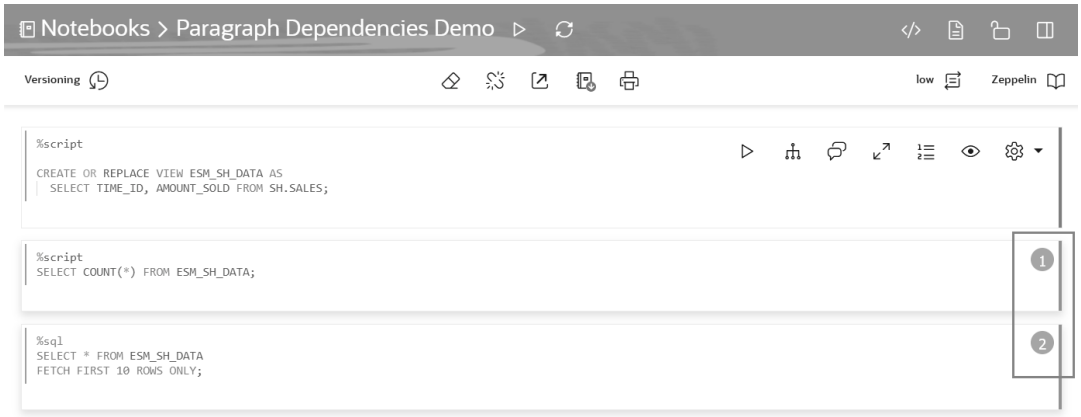
The message appears: *You are selecting dependents for this paragraph*

5. Click on the second and third paragraph to add them as dependents of paragraph one.

> **Note**
>
> The order of paragraph dependency is based on the order of your click.



6. Click **Save.** Once the dependent paragraphs are defined and saved, it is indicated by the numbers as shown in the screenshot here:



7. Now, go to the first paragraph and click **Run**. After the first paragraph starts successfully, the subsequent dependent paragraphs start to run according to the order of dependency.

4.1.4.3 Work with Notebook Versions on the Notebooks Page

By creating versions of your notebook, you can archive your work in a notebook.

You can create versions of notebooks on the Notebooks page, as well as in the Notebook editor. In this example, the **Notebook Versioning Demo** notebook is created and versioned as **Version 1**.

> **Note**
>
> A versioned notebook is non-editable. If you want to make any changes to a particular version of a notebook, you must restore that version to edit it.

Prerequisites: The *Notebook Versioning Demo* notebook. This notebook is created as part of the example here.

To create a new notebook version and view version history:

1. On the Notebooks page, click **Create Notebooks**.
2. In the Create Notebooks dialog, enter the name `Notebook Versioning Demo` in the **Name** field and click **OK**. The notebook is created and it opens in the notebook editor.
3. On the notebook, hover your cursor over the lower border of the paragraph and click the + icon to add a paragraph. Add two more paragraphs to this notebook, and paste the following PL/SQL script in the paragraphs:

a. In the first paragraph, copy and paste the following PL/SQL script. This script creates the view `ESM_SH_DATA` from the `SALES` table present in the `SH` schema.

```
%script

CREATE OR REPLACE VIEW ESM_SH_DATA AS
  SELECT TIME_ID, AMOUNT_SOLD FROM SH.SALES;
```

b. In the second paragraph, copy and paste the following SQL script. This script gives a count of the record present in the view `ESM_SH_DATA`.

```
%script
SELECT count(*) from ESM_SH_DATA;
```

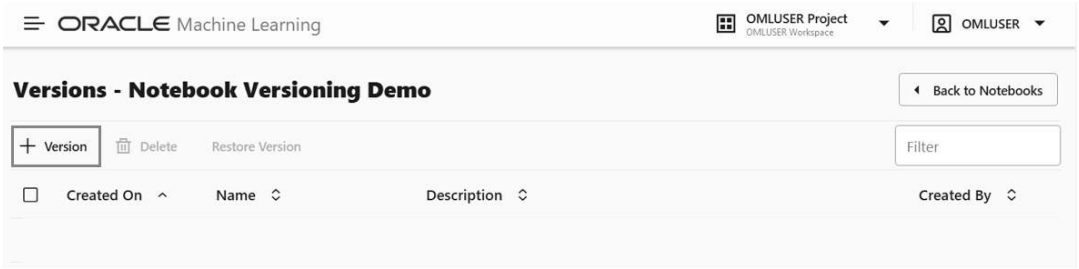
c. In the third paragraph, copy and paste the following SQL script to review the data in a tabular format.

```
SELECT * FROM ESM_SH_DATA
FETCH FIRST 10 ROWS ONLY;
```

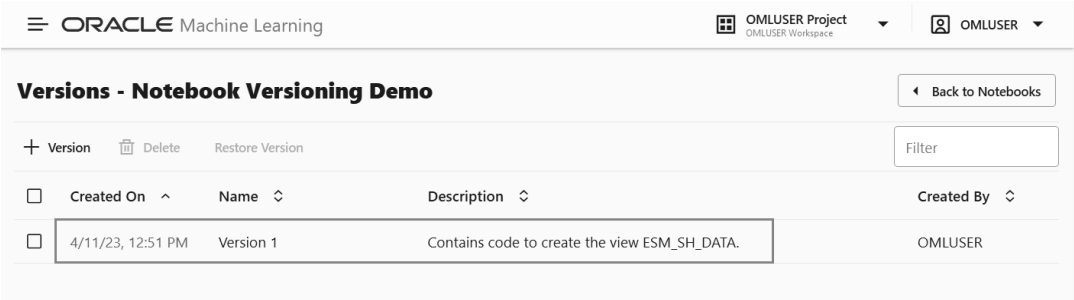
4. Run all the paragraphs, and go back to the Notebooks page after all the paragraphs run successfully.
5. On the Notebooks page, select the **Notebook Versioning Demo** notebook to enable all the edit options, and click **Versions** to go to the versions page for this notebook.

   The *Versions - Notebook Versioning Demo* page opens.
6. On the *Versions - Notebook Versioning Demo* page, click **Version** to create a new version of the notebook. This opens the Create Version dialog.

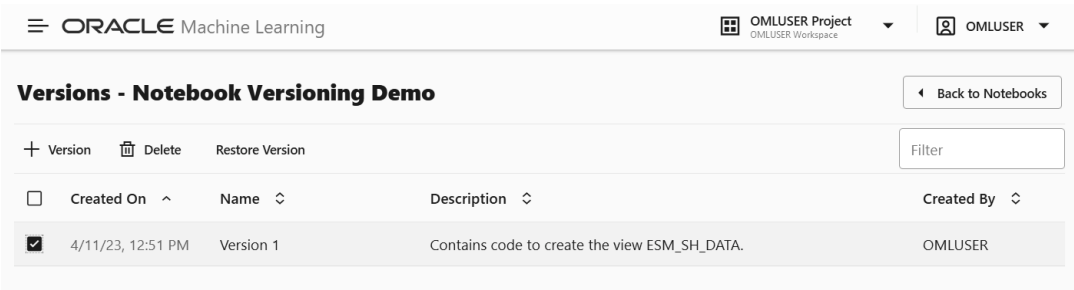
Figure 4-4 Versions page



7. In the Create Versions dialog:
   a. **Name:** Enter `Version 1` for the new version of this notebook
   b. **Comments:** Enter comments, if any.
   c. Click **OK.** Once the notebook version is created, it is listed on the *Versions - Notebook Versioning Demo* page.



8. On the *Versions - Notebook Versioning Demo* page, select **Version 1** of the notebook version that you just created to enable all the available options.



- Click **Delete** to delete the selected version of the notebook.
- Click **Restore** to restore the selected version of the notebook.

> **Note**
>
> Restoring a selected version of the notebook will discard all the unversioned changes, if any.

- Click **Back to Notebooks** to go back to the Notebooks page.

4.2 GitHub Notebooks

Oracle Machine Learning UI supports direct interaction of Oracle Machine Learning Notebooks with your external GitHub repositories. You can now directly import notebooks from your GitHub repositories.

> **Note**
>
> You may encounter an error if you retrieve information too frequently. This is because both public and private GitHub repositories are subjected to specific request limits. While it is unlikely that you will exceed the private repository limit of 5,000 requests per hour, it is common to hit the public limit of 60 requests per hour. GitHub restricts the number of API requests to prevent abuse and denial-of-service attacks, and also to ensure that the API remains available to all users.

With GitHub integration, you can interact with your GitHub repositories in the following ways:

- Create GitHub Credentials
  You must have separate credentials to connect to your GitHub repositories. The credential object with your GitHub credentials are stored in the Autonomous AI Database. To connect to your GitHub repositories that are marked private, you require credentials containing the necessary information such as name, email, personal access token and so on.
- Clone and Access your GitHub Notebooks
  To clone and access your GitHub notebook in Oracle Machine Learning UI, you must import the notebook from your GitHub repository. When you import the notebook, it clones the notebook and makes it available both on the Notebooks listing page and GitHub Notebooks listing page in Oracle Machine Learning UI.
- Edit and Sync your GitHub Notebook
  You can update remote changes in your notebooks and also upload local changes in the notebooks back to your GitHub repository using the Version Control options—**Pull changes** and **Push & commit** in OML notebook editor to synchronize the cloned notebook with your GitHub repository copy of the notebook.
- Manage GitHub Notebooks
  This page lists all the notebooks that you have cloned from a remote GitHub repository.

4.2.1 Create GitHub Credentials

You must have separate credentials to connect to your GitHub repositories. The credential object with your GitHub credentials are stored in the Autonomous AI Database. To connect to your GitHub repositories that are marked private, you require credentials containing the necessary information such as name, email, personal access token and so on.

The GitHub Credential interface in Oracle Machine Learning UI is the interface where you create the credentials to connect to your GitHub repositories.

> **Note**
>
> GitHub repositories that are public can be accessed without credentials.

A repository, also known as repo, is the fundamental element of GitHub. It is a storage space where you can keep all your project files, including code, images, and other resources, along with their revision history. Repositories are essential for organizing, managing, and collaborating on projects.

You can perform the following tasks on the GitHub Credentials page:

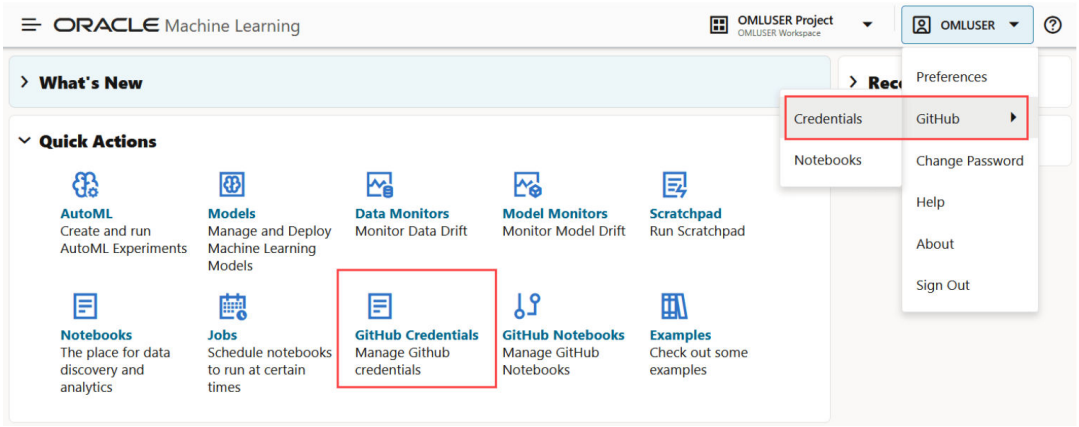
- **Create:** Click **Create** to create GitHub credentials
- **Delete:** Select a credential and click **Delete.** Deleting a credential also deletes the repository definition associated with the credential.

> **Note**
>
> When you delete a credential, all notebooks associated with the credential are converted into regular Oracle Machine Learning notebooks. As the credential is deleted, you can longer access and edit the associated notebooks in the GitHub repository.

To create GitHub credentials:

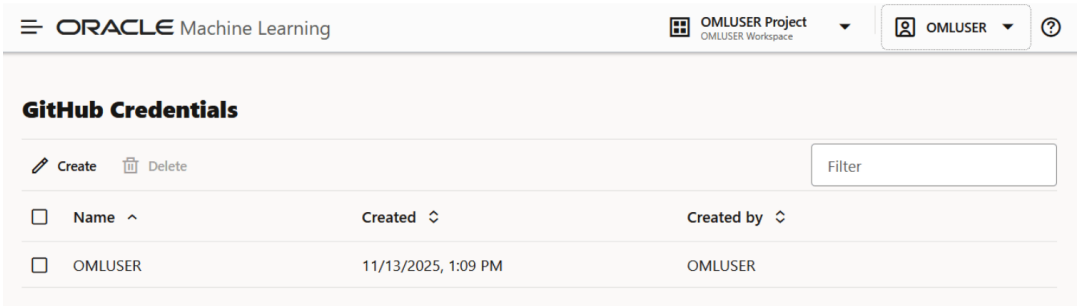
1. On the Oracle Machine Learning UI home page, click **GitHub Credentials**. Alternatively, you may click on the user profile drop-down menu, click **GitHub** and then click **Credentials**. This opens the **GitHub Credentials** page.

Figure 4-5 Home page



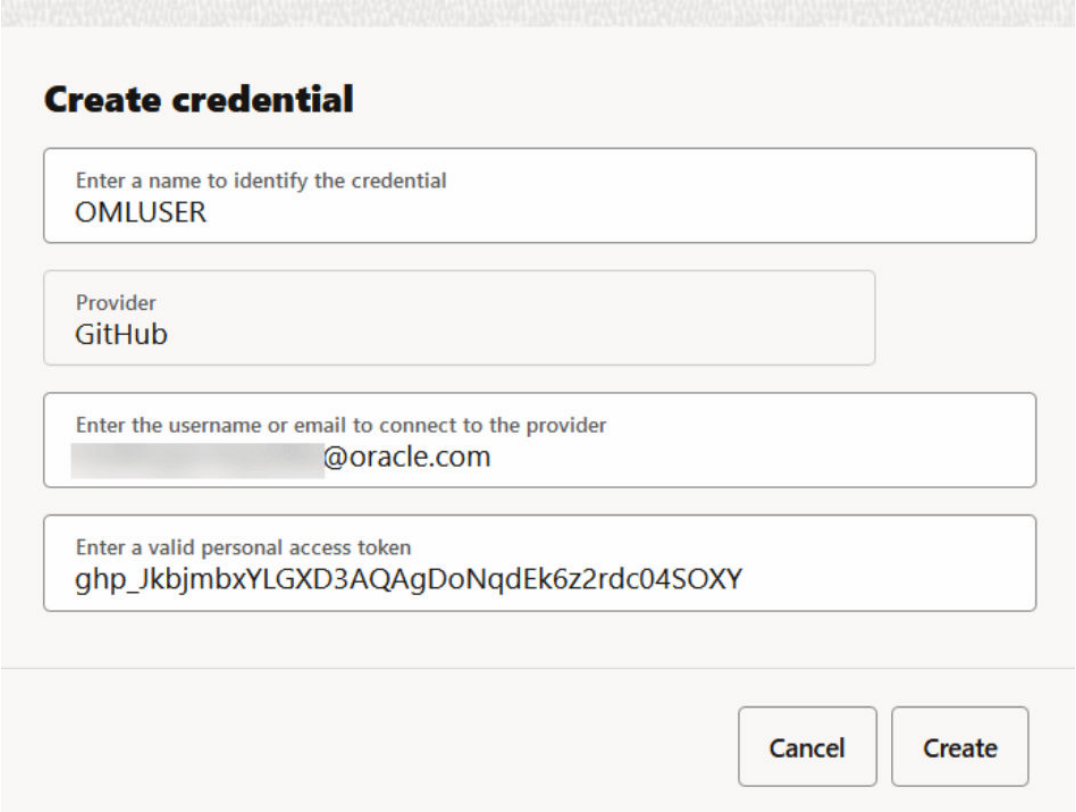
2. On the GitHub Credentials page, click **Create**. The **Create Credentials** dialog opens.

Figure 4-6 GitHub Credentials page



3. On the **Create Credentials** dialog, provide the following details:

Figure 4-7 Create Credential dialog



a. **Name:** This is a name to identify this credential.

> **Note**
>
> The length of the name is limited to 64 characters only. The valid characters are alphanumeric, underscore and the dollar sign.

b. **Provider:** This is the name of the external repository. Currently, only the GitHub repository is supported.

c. **Email:** This is the email ID of the Git user that created the repository. The maximum limit is 128 characters.

d. **Token:** This is the Personal Access Token, a secure method to authenticate with GitHub repository. You can generate it from your GitHub profile. See Generate Personal Access Token for details.

e. Click **Create.**

This completes the task of creating the credentials and takes you back to the GitHub Credentials page.

- Generate Personal Access Token
  A Personal Access Token (PAT) is an authentication token. It is a secure method used for user authentication.

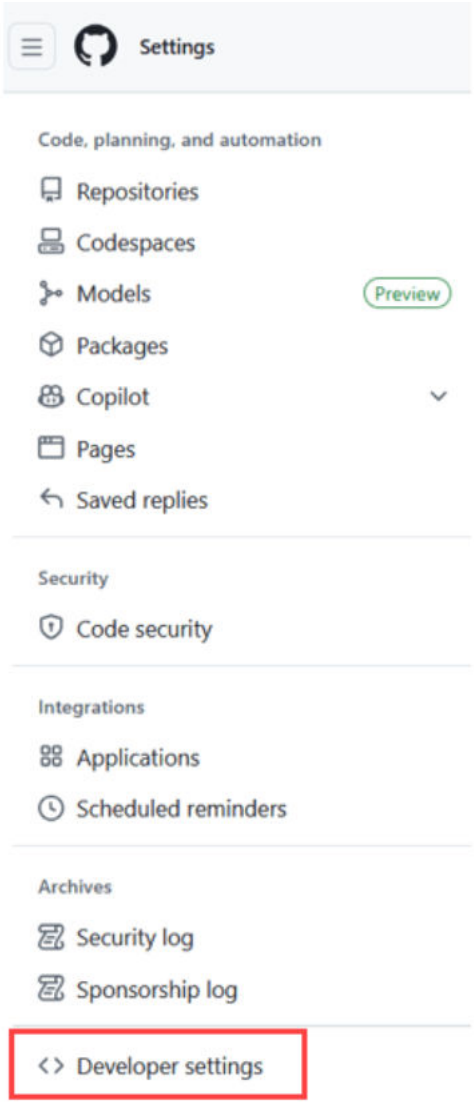
4.2.1.1 Generate Personal Access Token

A Personal Access Token (PAT) is an authentication token. It is a secure method used for user authentication.

To generate the Personal Access Token (PAT):

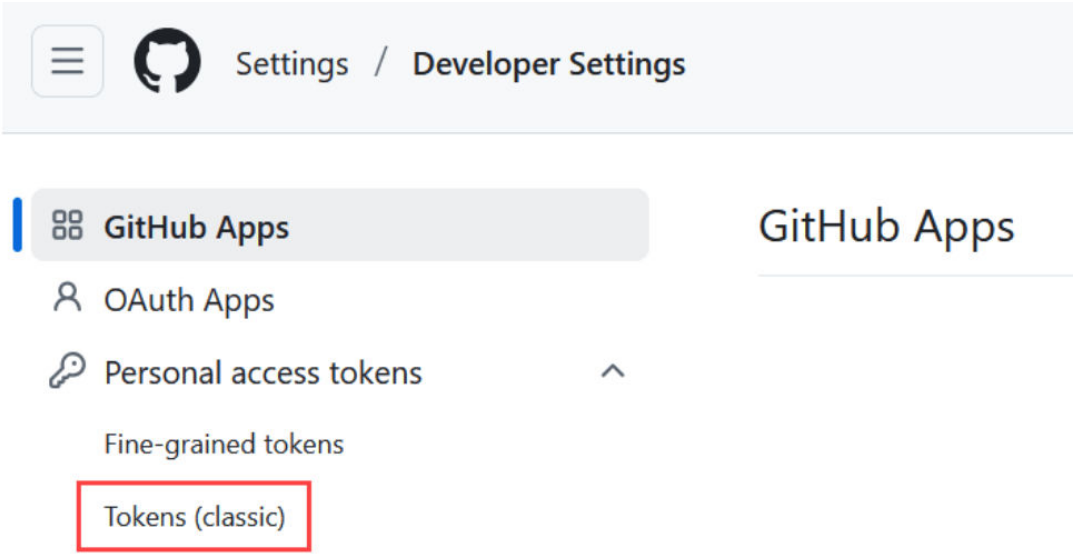
1. Open your GitHub profile, scroll down and click **Settings.**
2. On the left navigation of your GitHub Settings page, scroll down and click **Developer Settings.**

Figure 4-8 Developer Settings on left navigation pane



3. On the left navigation of the Developer Settings page, expand **Personal access tokens** and click **Tokens (classic).**

Figure 4-9 Github Developer Settings



4. Click **Generate new tokens** and then click **Generate new token (classic).**
5. On the **New personal access token (classic)** page, enter the following:
   - **Note:** Enter any note about the token.
   - **Expiration Date:** Click the down arrow and select an option about the validity of this token.
   - **Scope:** In this entire section, select **repo**.
6. Click **Generate token.**

   The token is generated. Click the copy icon to copy the token and use it when creating the GitHub credentials in Oracle Machine Learning UI.

4.2.2 Clone and Access your GitHub Notebooks

To clone and access your GitHub notebook in Oracle Machine Learning UI, you must import the notebook from your GitHub repository. When you import the notebook, it clones the notebook and makes it available both on the Notebooks listing page and GitHub Notebooks listing page in Oracle Machine Learning UI.

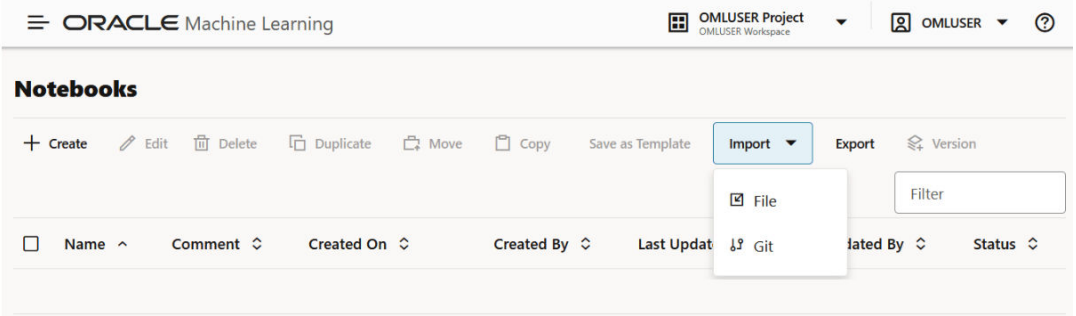
Prerequisites:

- You must have your GitHub credentials created.
- The notebooks to be cloned must reside in your GitHub repository.

To clone your GitHub notebook:

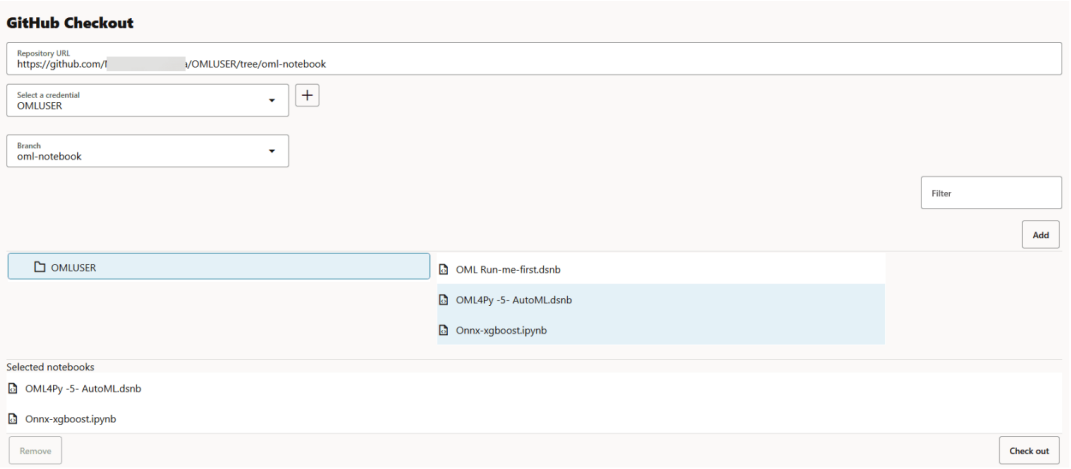
1. Go to your Oracle Machine Learning Notebooks listing page and click **Import.** Here, you have two options—**File** and **Git.**

Figure 4-10 Import options



2. Click **Git**. This opens the **GitHub Checkout** page.
3. On the **GitHub Checkout** page, enter these details:

Figure 4-11 GitHub Checkout page



a. **Repository URL:** Enter the URL of the GitHub repository you want to access. You have the following options to provide the GitHub repository URL:

- **Minimum valid URL:** You can provide the GitHub URL containing only the repository name and the owner. For example, `https://github.com/RepoOwner/RepoName`. This loads the branches found in the remote repository.
- **Base URL and branch:** Enter the GitHub repository URL along with the branch you want to clone. For example, `https://github.com/RepoOwner/RepoName/tree/BranchName/` or `https://github.com/RepoOwner/RepoName/blob/BranchName/`. This automatically loads the **Branches** field. Select the Branch Name defined in the URL. and this will trigger the load of the directory structure found in the repository.
- **Base URL, branch and directory:** You can also provide the GitHub URL containing one or several sub-directories in the remote repository. For example, `https://github.com/RepoOwner/RepoName/tree/Branch/Dir/Di2r` This loads the directory structure and pre-select the directory specified in the URL.
- **Complete file path:** You can also enter the complete file path in the remote GitHub repository. For example, `https://github.com/RepoOwner/RepoName/blob/`

`Branch/Dir/file.dsnb`. This loads the branches, directories and the file in the **Selected notebooks** field at the bottom of the dialog.

b. **Select a credential:** Click the down arrow and select a credential. If you do not have a credential created, click the + icon to create one. See Create GitHub Credentials for more information.

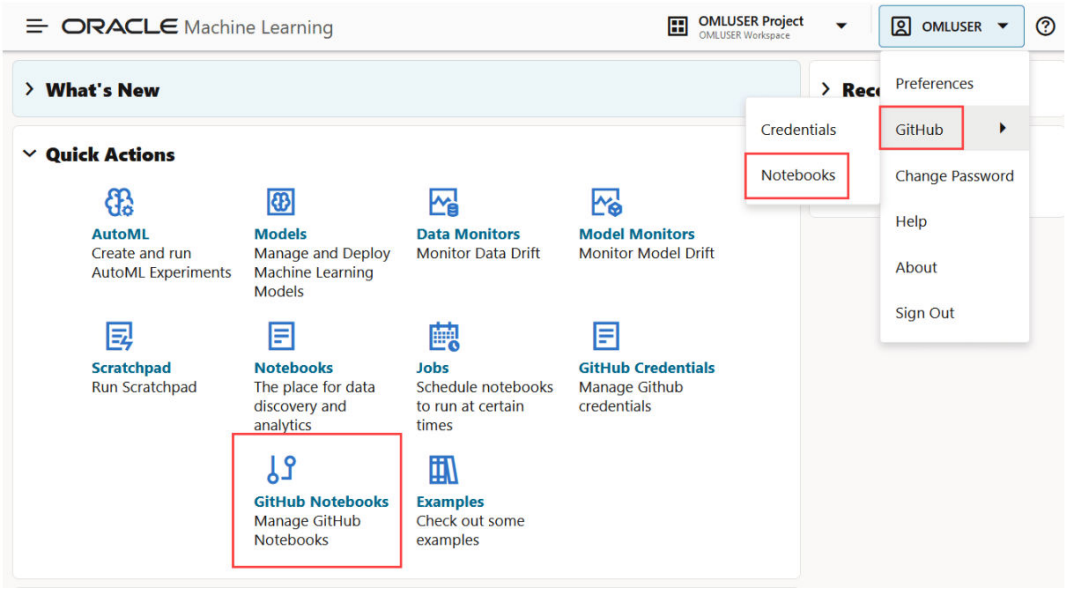
c. **Branch:** The drop-down menu displays the branches available in the remote GitHub repository based on the specified repository owner, repository name, and credential combination. Select a branch. The notebooks available in the branch are listed. You can also filter the notebooks you are looking for by typing in the notebook name in the **Filter** field.

d. Select the notebooks you want to clone and click **Add.** The notebooks you selected are now listed in the **Selected notebooks** section.

e. Click **Checkout.** This starts cloning all the GitHub notebooks you selected. Once completed, it displays the message "Notebooks successfully cloned". Click **Open Notebook listing** on the message box to go to the Notebooks listing page.

4. You can access all your cloned GitHub notebooks from the Oracle Machine Learning Notebooks listing page or from the GitHub Notebooks listing page.

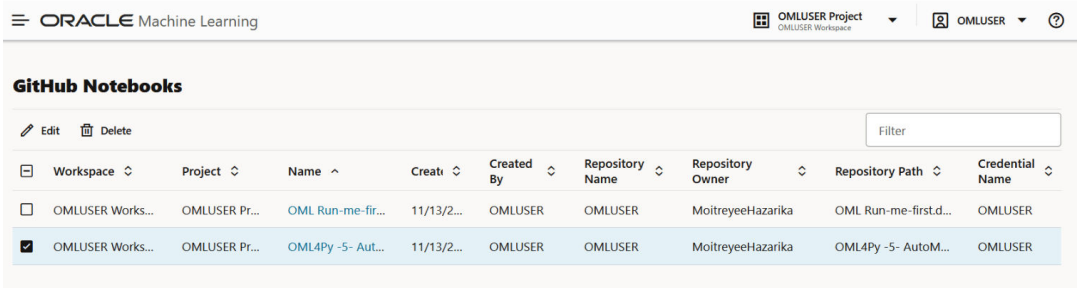
Figure 4-12 GitHub Notebooks option on the User Profile menu and OML UI home page



- To access your GitHub Notebooks listing page, click on the user profile drop-down menu on the top-right corner of the Oracle Machine Learning UI home page. Click **GitHub**, and then click **Notebooks**.
- Alternatively, you can access the GitHub Notebooks listing page by clicking **GitHub Notebooks** on the Oracle Machine Learning UI home page.

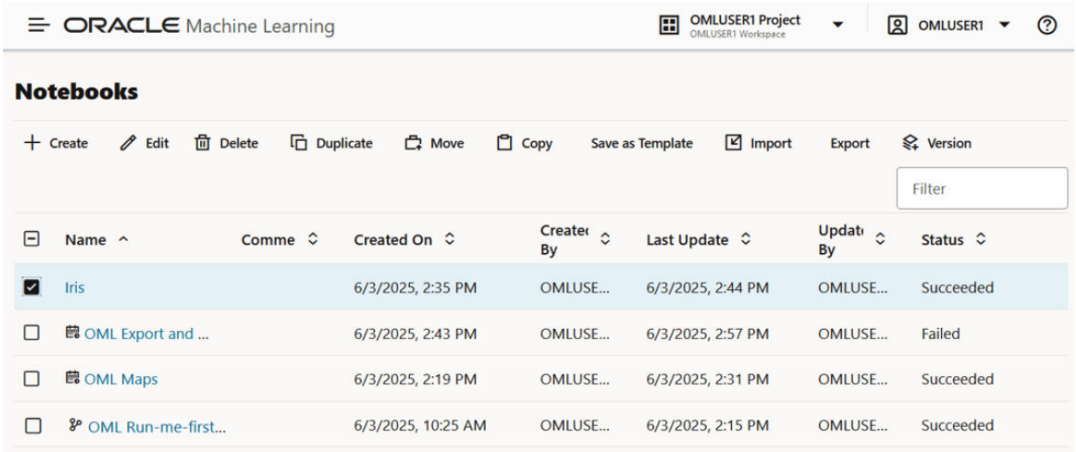
On the **GitHub Notebooks** listing page, click on the cloned notebook name to open it in the editor.

Figure 4-13 GitHub Notebooks listing page



5. To access your cloned GitHub notebooks from your Oracle Machine Learning Notebooks listing page, click on the notebook with the  icon.

Figure 4-14 OML Notebooks listing page



This completes the task of cloning and accessing your GitHub notebooks.

4.2.3 Edit and Sync your GitHub Notebook

You can update remote changes in your notebooks and also upload local changes in the notebooks back to your GitHub repository using the Version Control options—**Pull changes** and **Push & commit** in OML notebook editor to synchronize the cloned notebook with your GitHub repository copy of the notebook.

Prerequisites:

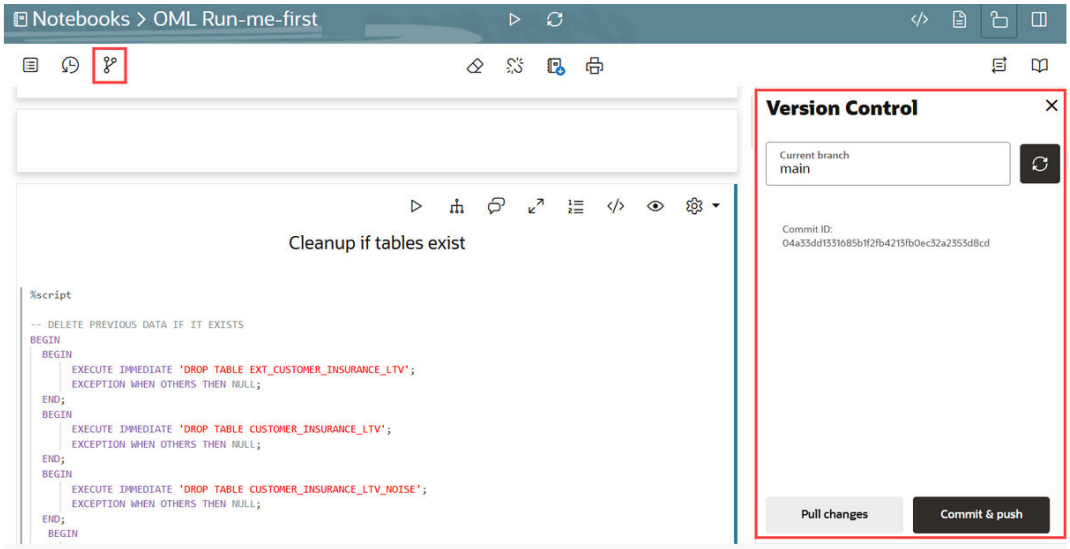
- Import the notebook from the GitHub repository. For more information, see Clone and Access your GitHub Notebooks
- Open the notebook:
  - Either from the GitHub Notebooks listing page. See Manage GitHub Notebooks
  - Or, from the Oracle Machine Learning Notebooks listing page. See Access your Oracle Machine Learning Notebooks .

To edit and sync the changes in your GitHub notebook:

1. After editing your notebook, click **Version Control** to commit the changes to the GitHub repository. The Version Control pane opens on the right.

2. On the Version Control pane:

Figure 4-15 Version Control in OML notebook editor



> **Note**
>
> To use these features, you must have your credentials linked to the GitHub notebook. Otherwise, the operations will fail.

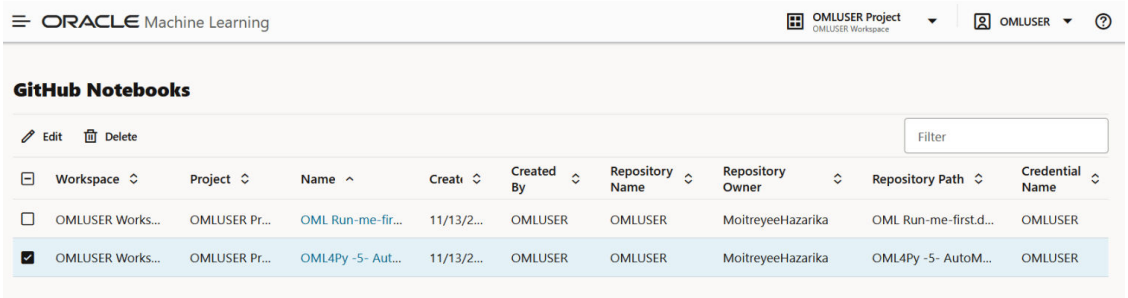
- Click **Commit & Push** to save and commit the edits done to the local (cloned) copy of your notebook to the `main` branch of your Git repository. This opens the Commit and Push dialog. Evaluate the changes (highlighted in a different color), provide a comment and click **Continue.** Once the changes are committed, it displays the confirmation message: `New commit <commit ID> created and pushed.`
- Click **Pull changes** to pull all changes in the notebook from your Git repository to your cloned copy in Oracle Machine Learning UI. Note the Git repository/branch name in the **Current branch** field. A confirmation dialog opens with the repository/branch name. Click **Pull** to confirm. Once complete, it displays the confirmation message: `Notebook pulled from the <branch name>.`
- Click  to change the branch and pull all changes in the notebook from the updated branch to your cloned copy in Oracle Machine Learning UI. In the **Current branch** field, copy paste the name of the branch if you want to change the branch from the default `main` to another branch. Once complete, it displays the confirmation message: `Current branch is set to <branch name>.`

4.2.4 Manage GitHub Notebooks

This page lists all the notebooks that you have cloned from a remote GitHub repository.

When you clone a notebook from a GitHub repository, it creates a repository definition. It contains the URL, directory information, the notebook name and creation date. All these information are listed here along with the notebook. You can directly open any notebook listed here, and edit or delete them.

Figure 4-16 GitHub Notebooks page



- **Workspace:** This is the name of the workspace where the project (in which the notebook is saved) is created.
- **Project:** This is the name of the project in which the notebook is saved in Oracle Machine Learning UI.
- **Name:** This is the name of the notebook. It is a clickable link. You can click on the name to open and edit it directly in the notebook editor.
- **Created:** This is the date and time of creation and last edit of the notebook.
- **Created By:** This is the name of the OML user.
- **Repository Name:** This is the name of the GitHub repository.
- **Repository Owner:** This is the name of the GitHub repository owner.
- **Repository Path:** This is the file name of the notebook.
- **Credential Name:** This is the name of the user credential

You can perform the following tasks:

1. **Edit:** To update repository path or credentials, select a notebook and click **Edit.** The **Update Repository** dialog opens. You can perform the following information: , you can update the GitHub repository URI and choose a different credential.
   a. Update the URI of the GitHub notebook.
   b. Click the down arrow to choose a different user.
   c. Click the + icon to create a new GitHub credential.
2. **Open:** To open a notebook, click on the notebook name. It opens the notebook directly on the Oracle Machine Learning Notebook editor.
3. **Delete:** To delete any notebook, select the notebook and click **Delete**. This deletes the local (cloned) copy of your GitHub notebook.

Related Topics

- Edit and Sync your GitHub Notebook
  You can update remote changes in your notebooks and also upload local changes in the notebooks back to your GitHub repository using the Version Control options—**Pull changes** and **Push & commit** in OML notebook editor to synchronize the cloned notebook with your GitHub repository copy of the notebook.

4.3 Enable GPU Compute Capabilities in a Notebook through the Python Interpreter

This topic demonstrates how to enable GPU compute capabilities in a notebook through the Python interpreter. It also shows how to get information about the current GPU on which the notebook is running, and other details.

Prerequisites:

- Paid Oracle Autonomous AI Database Serverless database instances.
- The GPU feature is enabled for Oracle Autonomous AI Lakehouse Serverless or Oracle Autonomous AI Transaction Processing Serverless instances with 16 or more ECPUs specified for the OML application. For cost details, refer to the *Oracle PaaS and IaaS Universal Credits Service Descriptions* document available on the Oracle Cloud Services contracts page.
- While basic NVIDIA libraries are included with the base environment, you are expected to create a custom Conda environment with the GPU-enabled 3rd party libraries required for your project. Only GPU-enabled packages will benefit from GPUs in Python paragraphs.

> **Note**
>
> By default, pre-installed and pre-configured NVIDIA libraries are provided to the GPU interpreter container in the host VM. However, third-party Python packages that use GPUs typically require specific versions of NVIDIA CUDA libraries as dependencies, which may override the included libraries.

- Third-party GPU-enabled Python packages. In this example, we use `pytorch`.

> **Note**
>
> There is an expected delay in starting a notebook with GPU compute capabilities due to reserving and starting the GPU resources, which may take a few minutes.

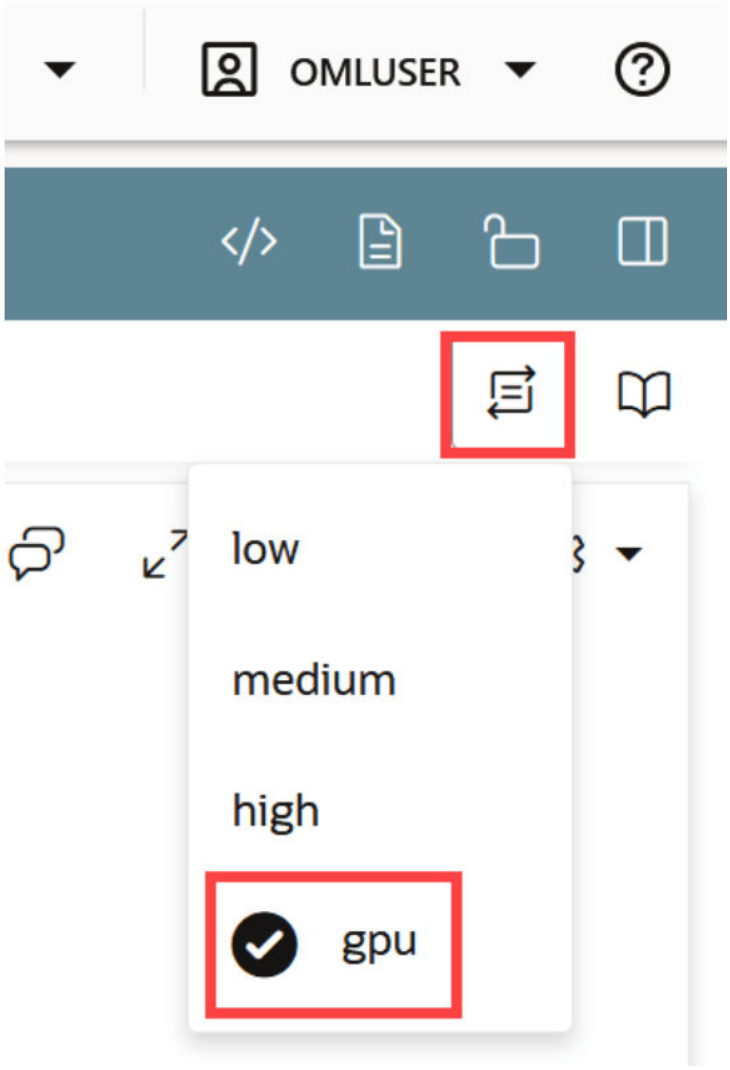
Generating embeddings using transformer models can be done in Python memory using OML Notebooks' Python interpreter. Using GPUs, transformer models, e.g., for generating sentence and image embeddings, can process larger volume data more quickly and efficiently.

To use the GPU compute capability in OML Notebooks:

- Create a Conda environment with the desired third-party GPU-enabled Python packages (ADMIN role required).
- Download and activate the Conda environment in OML Notebooks to use the GPU compute capabilities (OML_DEVELOPER role required).
- In Oracle Machine Learning Notebooks, select the notebook type **gpu** from the **Update Notebook Type** drop-down menu in the notebook editor (OML_DEVELOPER role required). This setting is persisted in the notebook until you change it to another type.

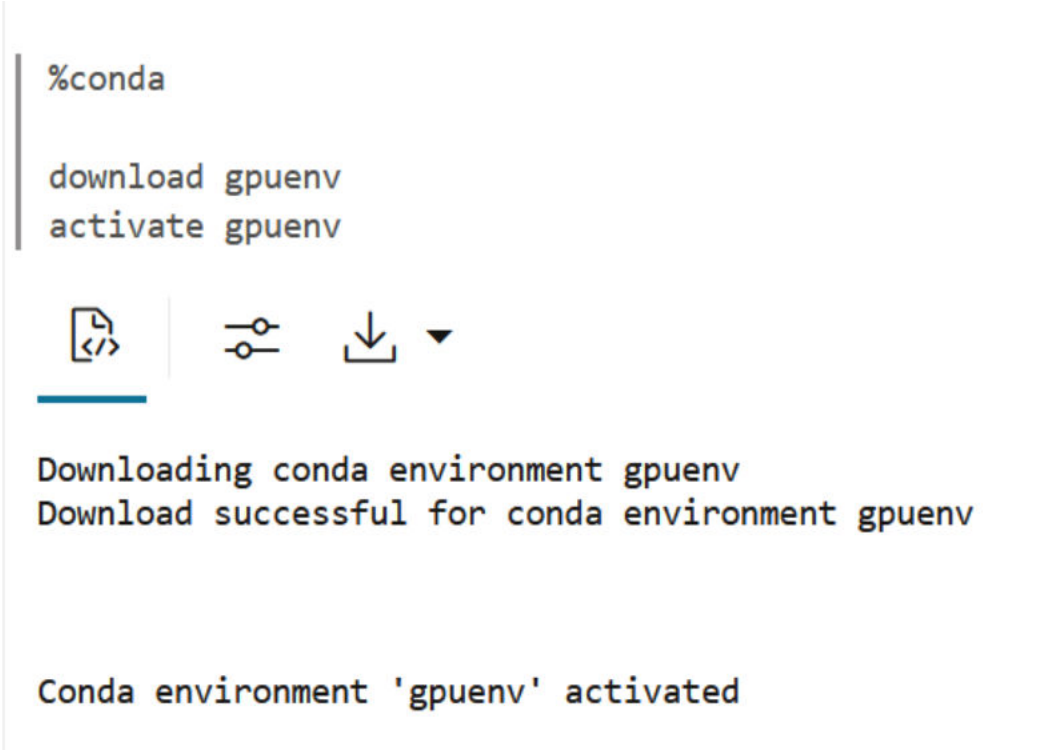
To enable GPU compute capabilities in a notebook, and to view information on GPU:

1. Create a notebook and open it in the notebook editor. Click on the **Update Notebook Type** icon and select **gpu**. By selecting **gpu**, you enable the notebook with GPU compute capabilities.



2. Run the following command to download and activate a Conda environment. To know more about how to create Conda environments for Python and R, see the Related Links section below.

```
%conda

download gpuenv
activate gpuenv
```



3. In this Conda environment, you will import the third-party GPU-enabled Python library Torch. Add a Python paragraph in the notebook to import the Python library `torch`. Type the directive `%python` and run the following command:

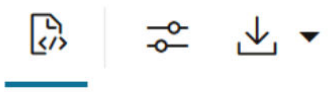
```
%python
import torch
import torch.nn as nn
import torch.optim as optim
import time
```

4. In another Python paragraph in the notebook and run the following command to check if GPU is available and use it:

```
%python
        device = torch.device('cuda' if torch.cuda.is_available()else
'cpu')
        print(f"Device in use is: {device}")
```

The command returns the following information:

```
%python

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(f"Device in use is: {device}")
```

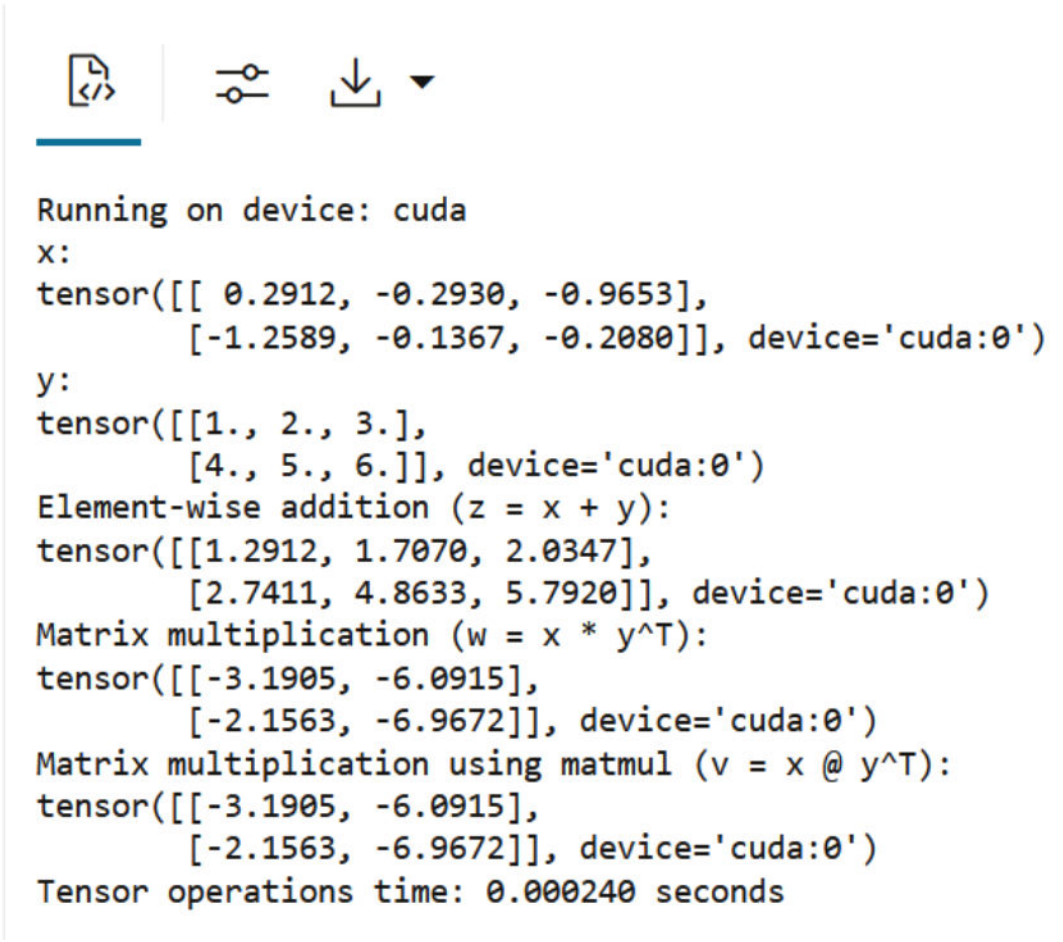


```
Device in use is: cuda
```

5. In this step, we will create another Python paragraph and run the following to:
   - Set the device to CPU or GPU if available
   - Create tensors on the specified device
   - Measure time for basic operations, and
   - Print the operations and results

```
%python

import torch
import time
import matplotlib.pyplot as plt

# Set the device to CPU or GPU if available
device = 'cuda' if torch.cuda.is_available() else 'cpu'
print(f"Running on device: {device}")

# Create tensors on the specified device
x = torch.randn(2, 3, device=device)
y = torch.tensor([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]], device=device)

# Measure time for basic operations
start_time = time.time()
z = x + y  # Element-wise addition
w = torch.mm(x, y.T)  # Matrix multiplication
v = torch.matmul(x, y.T)  # Another way to perform matrix multiplication
end_time = time.time()

# Print the operations and results
print(f"x:\n{x}")
print(f"y:\n{y}")
print(f"Element-wise addition (z = x + y):\n{z}")
print(f"Matrix multiplication (w = x * y^T):\n{w}")
print(f"Matrix multiplication using matmul (v = x @ y^T):\n{v}")
print(f"Tensor operations time: {end_time - start_time:.6f} seconds\n")
```

The command returns the following information:



```
Running on device: cuda
x:
tensor([[ 0.2912, -0.2930, -0.9653],
        [-1.2589, -0.1367, -0.2080]], device='cuda:0')
y:
tensor([[1., 2., 3.],
        [4., 5., 6.]], device='cuda:0')
Element-wise addition (z = x + y):
tensor([[1.2912, 1.7070, 2.0347],
        [2.7411, 4.8633, 5.7920]], device='cuda:0')
Matrix multiplication (w = x * y^T):
tensor([[-3.1905, -6.0915],
        [-2.1563, -6.9672]], device='cuda:0')
Matrix multiplication using matmul (v = x @ y^T):
tensor([[-3.1905, -6.0915],
        [-2.1563, -6.9672]], device='cuda:0')
Tensor operations time: 0.000240 seconds
```

Related Topics

- Resource Services and Notebooks
  This topic lists the number of notebooks that you can run concurrently per Autonomous AI Database instance for each resource service.
- Use the Conda Interpreter in a Notebook Paragraph
  Oracle Machine Learning Notebooks provides a Conda interpreter to enable administrators to create conda environments with custom third-party Python and R libraries. Once created, you can download and activate Conda environments inside a notebook session also using the Conda interpreter.
- Create a Conda Environment for Python, Install a Python Package and Upload it to Object Storage
  This topic demonstrates how to create a Conda environment `mypyenv` for Python packages, with Python 3.13.5 that is compatible with OML4Py. Here, you will also install the Python package seaborn from the `conda-forge` channel, and upload the Conda environment `mypyenv` to object storage.
- Create a Conda Environment for R and Install R Package and Upload it to Object Storage
  This topic shows how to create a Conda environment `myrenv` for R packages, with R-4.0.5 for OML4R compatibility, install the `forecast` package, and upload the Conda environment to object storage.

4.4 Visualize your Data in Oracle Machine Learning Notebooks

Oracle Machine Learning Notebooks offer rich visualization capabilities of your data. The visualizations depend on the type of your dataset.

The following visualization options are available:

- Visualize Data in a Table
  A table is an arrangement of information or data in rows and columns. Using the Oracle Machine Learning Notebooks, you can create database tables, and also view the information in a tabular format.
- Visualize Data in a Bar Chart
  A bar chart is a graphical representation of data in rectangular bars. The length or height of the bars, depending on the horizontal or vertical orientation, depict the data set distribution. One axis represents a category, while the other represents values or counts.
- Visualize Data in a Funnel Chart
  A funnel chart is a graphical representation that resembles the shape of a funnel where each segment gets progressively narrower. The segments are arranged vertically and depict a hierarchy. Within the funnel chart, each segment corresponds to a step or stage in a sequential process.
- Visualize Data in a Pyramid Chart
  Pyramid charts present your data in a distinctive triangular configuration, horizontally segmented into partitions. Each segment in the pyramid charts represents points or steps in ascending or descending order.
- Visualize Data in a Scatter Plot
  Scatter plots represent the relationship between two numeric variables in a data set. It represents data points on a two-dimensional plane and show how much one variable is affected by another. The independent variable is plotted on the X-axis, while the dependent variable is plotted on the Y-axis. You can display points by one or more grouping variables such that each group has a distinct color and shape.
- Visualize Data in a Line Chart
  A line chart is a graphical representation used to display data points connected by straight lines.
- Visualize Data in an Area Chart
  An area chart uses lines to connect the data points and fills the area below these lines to the x-axis. Each data series contributes to the formation of a distinct shaded region. This emphasizes its contribution to the overall trend. As the data points fluctuate, the shaded areas expand or contract.
- Visualize Data in a Pie Chart
  A pie chart is a graphical representation of data in a circular form, with each slice of the circle representing a fraction that is a proportionate part of the whole.
- Visualize Data in a Box Plot
  A box plot provides an overview of data distributions in numeric data. It provides general information about the symmetry, skewness, variance, and outliers in a dataset. The box plot uses boxes and lines to depict the data distribution.
- Visualize your Data in a Sunburst Chart
  The sunburst chart is typically used to visualize hierarchical data structures - with part-to-whole relationships in data depicted additionally.

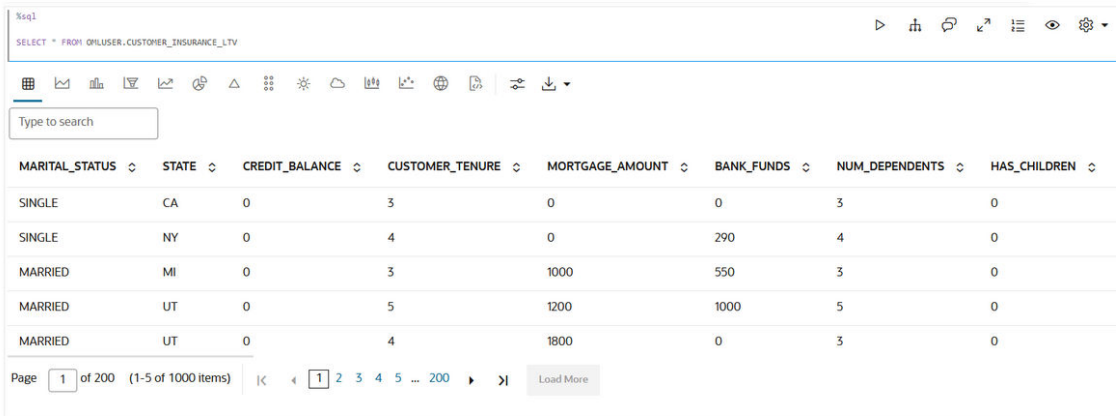
- Visualize your Data in a Tag Cloud
  A tag cloud is a visual representation of the most popular words or tags found in free-form text. The font size or the color of the text represents the frequency of occurrence.
- Visualize your Data in a Treemap
  A treemap is a visualization composed of nested rectangles, that represent certain categories within a selected dimension and are ordered in a hierarchy, or “tree.” Quantities and patterns can be compared and displayed in a limited chart space.
- Visualize Data in a Map
  The map visualization of OML Notebooks plots data points on a geographical map. For visualizing your data in a map in OML Notebooks, your data must contain explicit latitude and longitude values of locations to correctly position data on the map.

4.4.1 Visualize Data in a Table

A table is an arrangement of information or data in rows and columns. Using the Oracle Machine Learning Notebooks, you can create database tables, and also view the information in a tabular format.

Here is the data from the `CUSTOMER_INSURANCE_LTV` table presented in a tabular format. The table has 8 columns and 200 rows.

Figure 4-17 Table

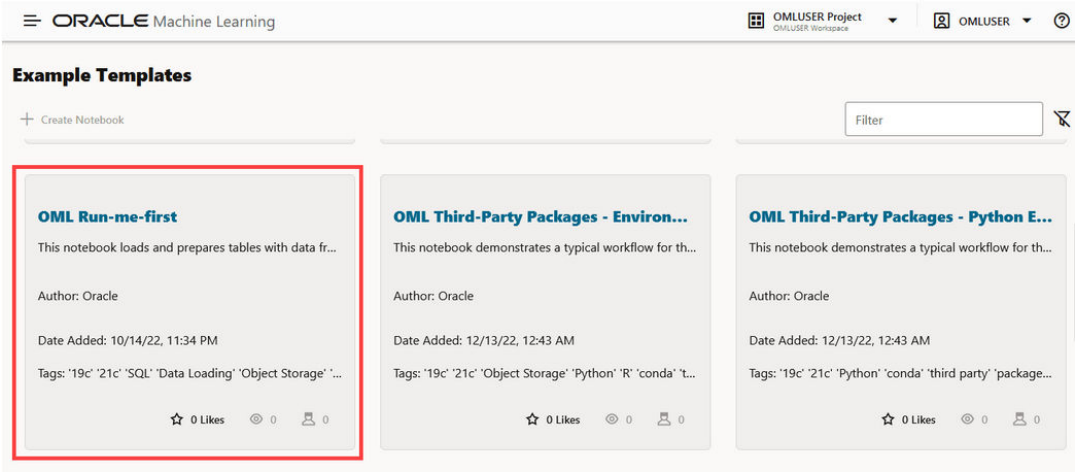


Data set: `CUSTOMER_INSURANCE_LTV`. In this example, we will use the example template notebook *OML-Run-me-first*.

To visualize data in a table:

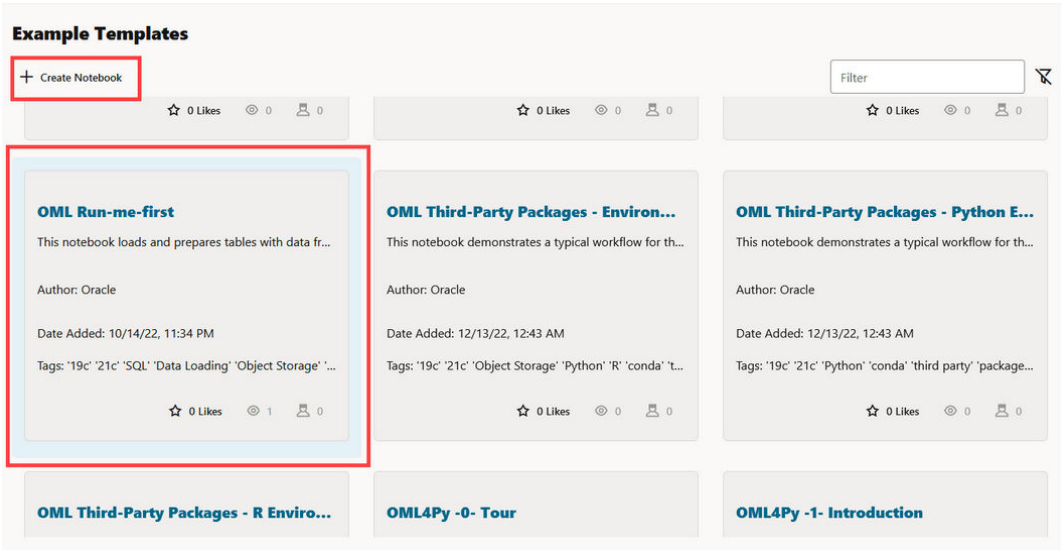
1. On the Oracle Machine Learning UI home page, click **Examples**. Or, open the left navigation menu by clicking the Cloud menu icon click Cloud menu icon on the top left corner of the page. Click **Templates** and then click **Examples.**
2. The *OML-Run-Me-First* example template is listed. If you are unable to view it, type the name in the Filter field.

Figure 4-18 OML Run-me-first Example Template



3. Click on the *OML Run-me-first* tile (and not on the name) to highlight it in blue. Then click the **Create Notebook** icon.

Figure 4-19 Create option on the Example Template page

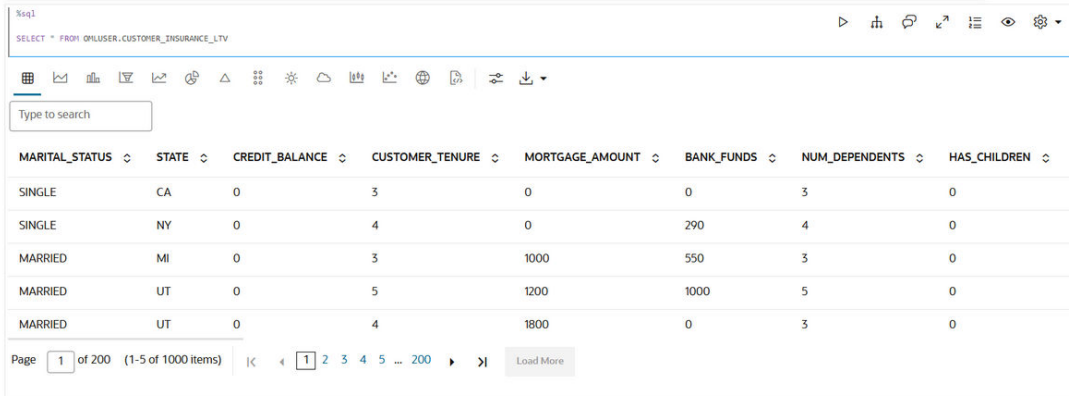


4. In the Create Notebook Dialog, click **OK.**
5. Click **Open Notebook** in the confirmation dialog to open the notebook.
6. Click the **Run Paragraphs** icon in the notebook to run all the paragraphs. This will also create the `CUSTOMER_INSURANCE_LTV` table. Click **Confirm** in the Confirmation dialog.
7. To view the data in a table format, run the following script in a SQL paragraph:

```
%sql

SELECT * FROM OMLUSER.CUSTOMER_INSURANCE_LTV
```

The script presents the data in a tabular format as shown in the screenshot:

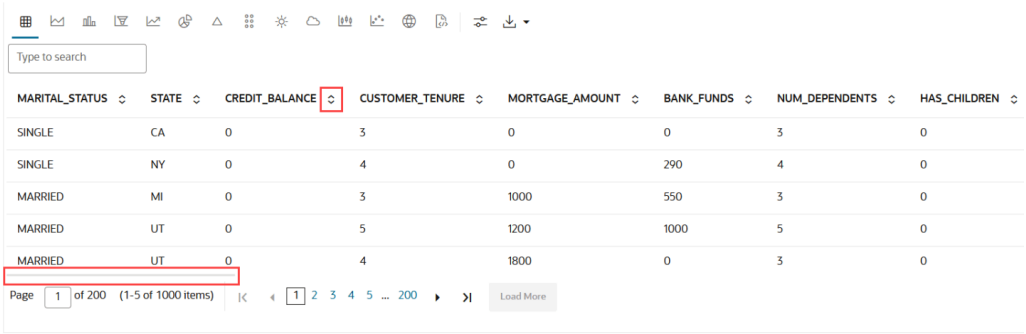
Figure 4-20 View the CUSTOMER_INSURANCE_LTV table



8. In this table, you can customize your views and settings.

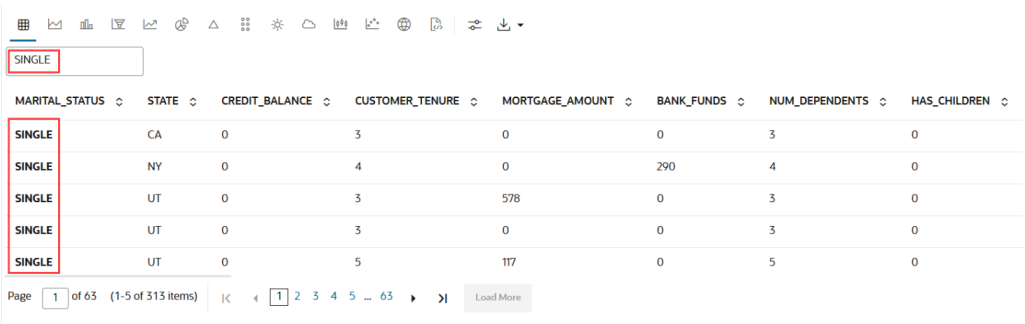
- Sort the columns in ascending or descending order: Click on the down arrow or up arrow against the columns to sort the data in ascending or descending order.

Figure 4-21 Sort Table Columns



- Use the horizontal scroll bar to scroll horizontally to view the columns on the right.
- Filter specific search terms. In the **Search** field, type the entry or term that you are looking for. In this example, the term `Single` is entered. All the rows that contain the term SINGLE in the column `MARITAL_STATUS` are filtered for display.

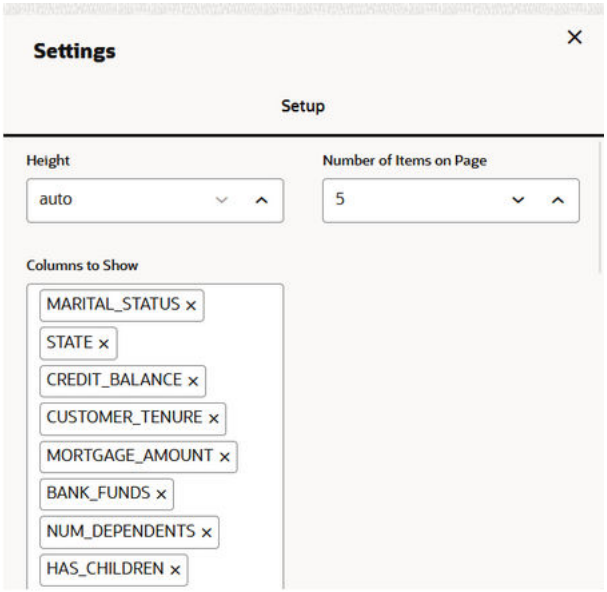
Figure 4-22 Filtering table columns by keywords



> **Note**
>
> Rows that do not contain this term are hidden from the view and the remaining rows highlight the location of the search term within the row.

9. By default, 5 rows are displayed. If you want to view more rows or customize the table settings, click on the **Settings** icon to open the Settings dialog. Here, you can edit the following:

Figure 4-23 Settings dialog



- **Height:** This parameter changes the height of the visualization. Click on the up or down arrow to increase or decrease the visualization height.
- **Number of Items on Page:** Click on the up or down arrow, as applicable, to set the number of rows to be displayed on the page. By default, 5 rows are displayed.

Columns to Display: By default, all the columns are listed. If you want to remove any column from displaying, click on the X in the column name. To view the column again, click inside the **Columns to Show** field. The hidden columns are displayed. Click on the column that you want to view again. In this example the column `MARITAL_STATUS` was removed. Clicking on the **Columns to Show** field displays it. Click on it to include in the display.

This completes the task of creating a table, and visualizing the data in it.

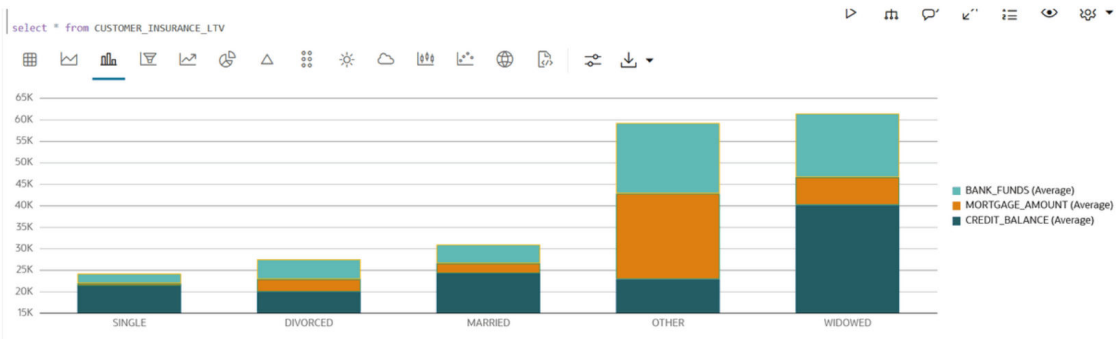
4.4.2 Visualize Data in a Bar Chart

A bar chart is a graphical representation of data in rectangular bars. The length or height of the bars, depending on the horizontal or vertical orientation, depict the data set distribution. One axis represents a category, while the other represents values or counts.

This is a stacked bar chart generated for the `CUSTOMER_INSURANCE_LTV` table. The average of `CREDIT_BALANCE`, `MORTGAGE_AMOUNT`, and `BANK_FUNDS` are plotted along the Y-axis, and `MARITAL_STATUS` is depicted along the X-axis. Here, `CREDIT_BALANCE`, `MORTGAGE_AMOUNT`, and

BANK_FUNDS are stacked in ascending order in each bar for the groups—SINGLE, MARRIED, DIVORCED, WIDOWED, and OTHERS.

Figure 4-24 Bar Chart



When to use this chart: Use bar charts to show a distribution of data points, and perform a comparison of metric values across different subgroups of your data.

Data set: CUSTOMER_INSURANCE_LTV. In this example, we will use the example template notebook OML-Run-me-first.

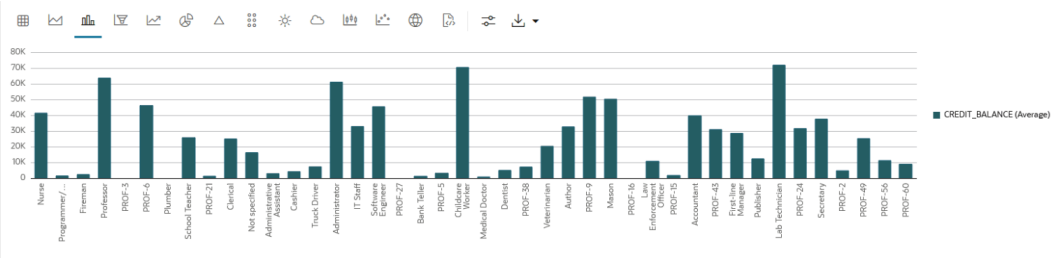
To visualize data in a bar chart:

1. In the OML-Run-Me-First notebook, go to the paragraph where you viewed the CUSTOMER_INSURANCE_LTV. Click on the bar chart icon.

Figure 4-25 Tool bar showing the bar chart icon

2. By default, it shows CREDIT_BALANCE along the Y-axis and the data is grouped by MARITAL_STATUS along the X-axis.

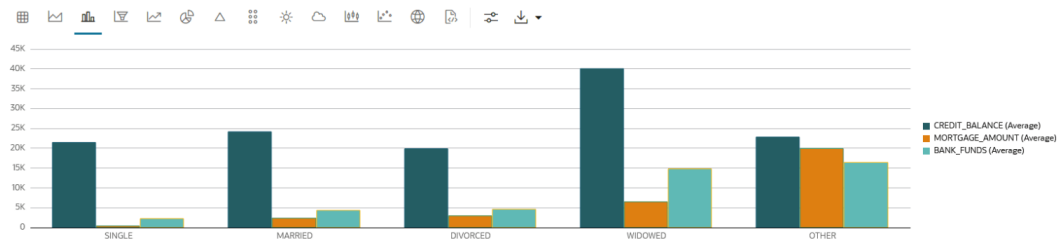
Figure 4-26 Default Bar Chart



3. Click on the **Settings** icon to get a different view of this data. Under **Setup:**
   - **Series to Show:** Select CREDIT_BALANCE, MORTGAGE_AMOUNT, and BANK_FUNDS.
   - In **Group By:** Select MARITAL_STATUS.
4. The average of CREDIT_BALANCE, MORTGAGE_AMOUNT, and BANK_FUNDS are each represented by adjacent bar charts, and the bar charts are grouped by MARITAL_STATUS—single,

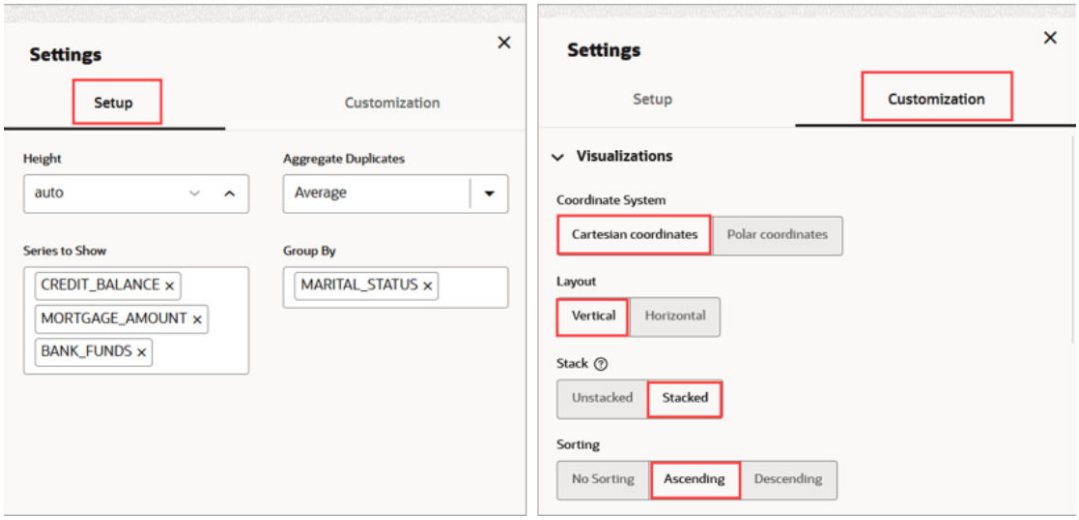
married, divorced, widowed, and others. The bar chart now looks like this, as shown in the screenshot below:

Figure 4-27 Bar Chart



5. Once again, click on the Settings icon and click **Customization**:

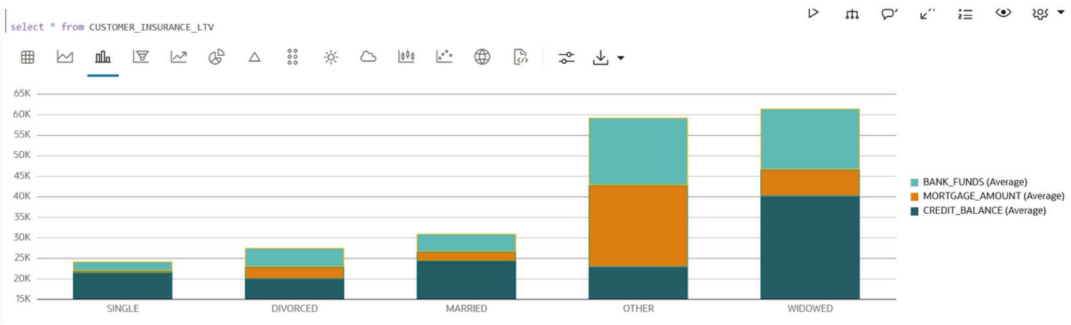
Figure 4-28 Settings - Bar Chart



- **Coordinate System:** The coordinate system of the chart. Supported Values—Polar Coordinates, Cartesian Coordinates. Click **Cartesian Coordinates.**
- **Layout:** The chart orientation. Either horizontal or vertical.
- **Stack:** Defines whether the data items are stacked or not. Click **Stacked.**
- **Sorting:** Specifies the sorting of the data. It should only be used for pie charts, bar/line/area charts with one series, or stacked bar/area charts. Sorting will not apply when using a hierarchical group axis. Click **Ascending.**
- **Zoom:** Specifies the zoom and scroll behavior of the chart. Live behavior means that the chart will be updated continuously as it is being manipulated, while "delayed" means that the update will wait until the zoom/scroll action is done. While "live" zoom and scroll provides the best end user experience, no guarantees are made about the rendering performance or usability for large data sets or slow client environments. If performance is an issue, "delayed" zoom and scroll should be used instead.

The bar chart now presents the data in a stacked manner, and in ascending order, as shown in the screenshot below:

Figure 4-29 Stacked Bar Chart



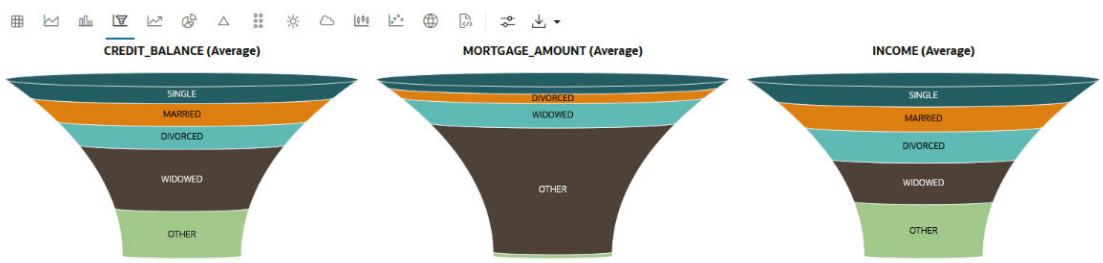
This completes the task of visualizing your data in a bar chart, and customizing its output.

4.4.3 Visualize Data in a Funnel Chart

A funnel chart is a graphical representation that resembles the shape of a funnel where each segment gets progressively narrower. The segments are arranged vertically and depict a hierarchy. Within the funnel chart, each segment corresponds to a step or stage in a sequential process.

In this data visualization, the attributes `CREDIT_BALANCE`, `MORTGAGE_AMOUNT` and `INCOME` are depicted for each of these groups—SINGLE, MARRIED, DIVORCED, WIDOWED, and OTHERS in three funnel charts. This is the data visualization after customizing the funnel chart.

Figure 4-30 Funnel Chart



When to use this chart: Use this chart to visualize a linear sequential process, mostly in business and sales contexts. For example, you can use a funnel chart to track the sales process, order fulfillment, website visitor trends, and so on.

Data set: `CUSTOMER_INSURANCE_LTV`. In this example, we will use the example template notebook *OML-Run-me-first*.

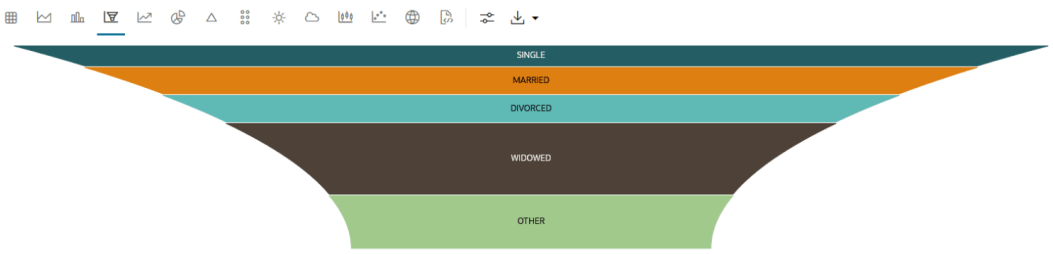
To visualize the data in a funnel chart:

1. In the OML-Run-Me-First notebook, go to the paragraph where you viewed the `CUSTOMER_INSURANCE_LTV`. Click on the Funnel chart icon.

Figure 4-31 Tool bar showing the funnel chart icon

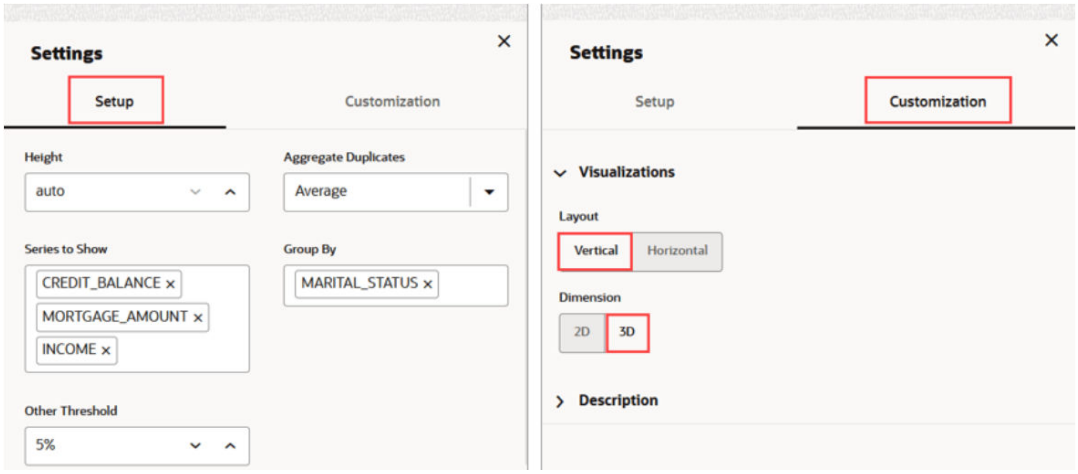
2. The data is displayed as below. Hover your cursor to view the series that is plotted in the funnel chart for each of the 5 groups.

Figure 4-32 Funnel Chart



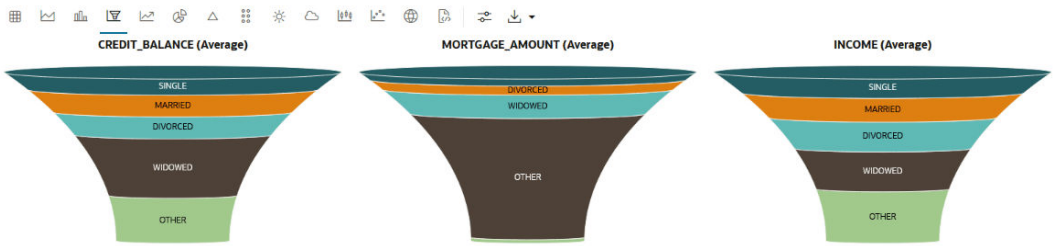
3. Let's compare a few attributes `CREDIT_BALANCE`, `MORTGAGE_AMOUNT` and `INCOME` of the same groups. Click **Settings** and edit the following:

Figure 4-33 Settings - Funnel Chart



4. These attributes `CREDIT_BALANCE`, `MORTGAGE_AMOUNT` and `INCOME` are now displayed in three funnel charts, as shown in the screenshot below:

Figure 4-34 Funnel Chart after Customization



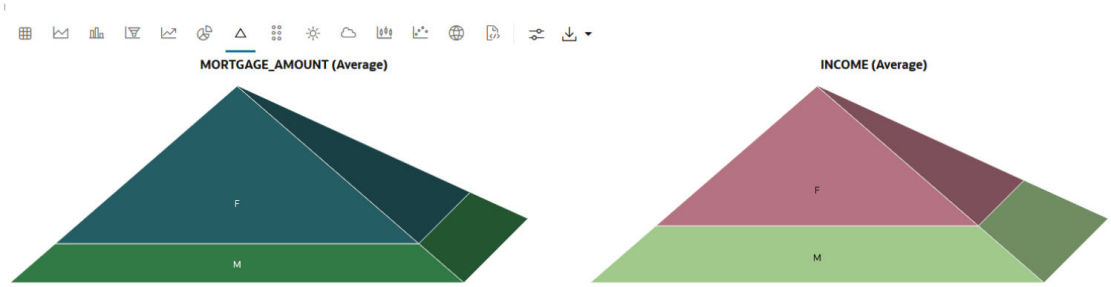
This completes the task of visualizing your data in a funnel chart.

4.4.4 Visualize Data in a Pyramid Chart

Pyramid charts present your data in a distinctive triangular configuration, horizontally segmented into partitions. Each segment in the pyramid charts represents points or steps in ascending or descending order.

Here are two pyramid charts depicting the attributes `MORTGAGE_AMOUNT` and `INCOME` for the two groups M (Male) and F (Female). Each pyramid is divided into two segments to represent the two groups in ascending order.

Figure 4-35 Pyramid chart



When to use this chart: Use this chart to depict hierarchical structures and the relative proportions of different values. They are typically used for displaying demographic data, market segmentation, or organizational structures. In any case, the data must have a progressive order.

Data set: `CUSTOMER_INSURANCE_LTV`. In this example, we will use the example template notebook OML-Run-me-first.

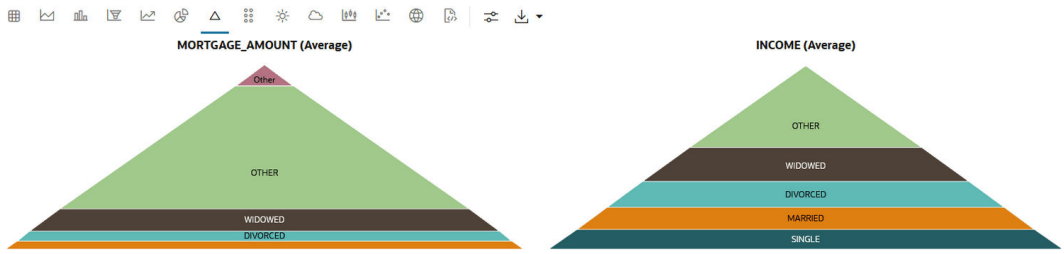
To visualize data in a pyramid chart:

1. In the OML-Run-Me-First notebook, go to the paragraph where you viewed the `CUSTOMER_INSURANCE_LTV`.
2. Click on the pyramid chart icon.

Figure 4-36 Pyramid icon on the toolbar

3. Click the **Settings** icon . Under **Setup:**
   - **Aggregate Duplicates:** Select **Average.**
   - **Series to Show:** Select `INCOME` and `MORTGAGE_AMOUNT`.
   - **Group By:** Select `MARITAL_STATUS`.

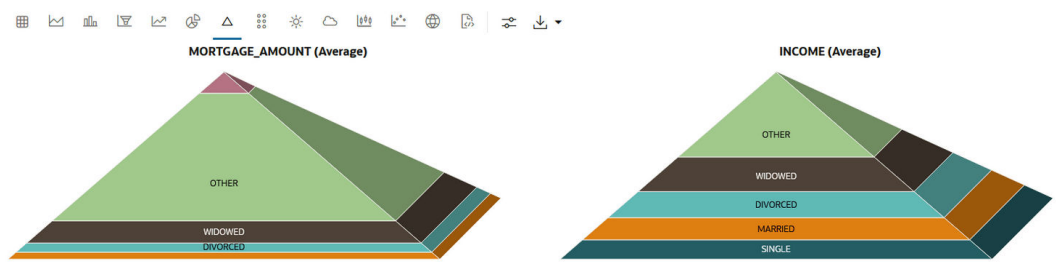
Figure 4-37 Pyramid Chart showing two series



The Pyramid chart is now shows the average for the two series `INCOME` and `MORTGAGE_AMOUNT`, grouped by `MARITAL_STATUS`.

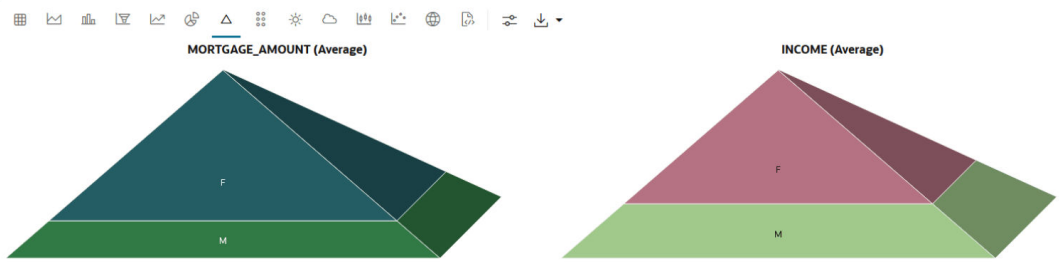
4. On the Settings dialog, click **Customization:**
   - Click **Customization** expand **Visualization.**
   - Click **3D** under **Dimension**

Figure 4-38 Pyramid Chart rendered in three dimension



5. Click the **Setup** tab again, and change **Group By** to `GENDER`.

Figure 4-39 Pyramid Chart - Depicting the two series grouped by Gender



This completes the task of visualizing your data in a pyramid chart. The pyramid chart shows a clear correlation between the two genders, and their income level and mortgage amount. For both the categories, the average income and mortgage amount taken is higher for Females.

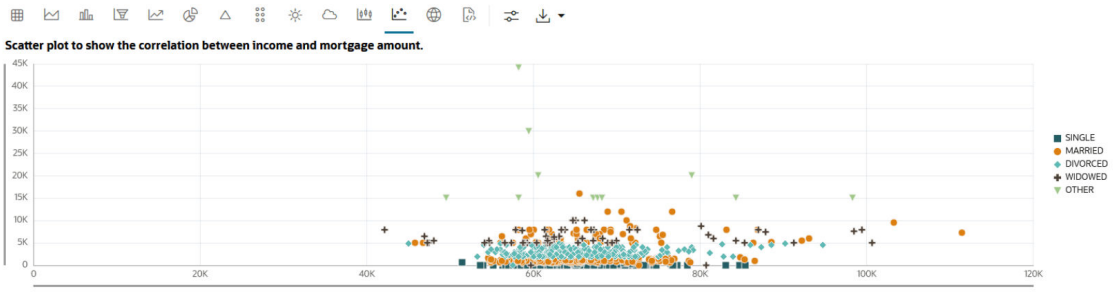
4.4.5 Visualize Data in a Scatter Plot

Scatter plots represent the relationship between two numeric variables in a data set. It represents data points on a two-dimensional plane and show how much one variable is affected by another. The independent variable is plotted on the X-axis, while the dependent

variable is plotted on the Y-axis. You can display points by one or more grouping variables such that each group has a distinct color and shape.

This is a scatter plot rendered for the `CUSTOMER_INSURANCE_LTV` table. The scatter plot shows a strong correlation between `INCOME` and `MORTGAGE_AMOUNT` in the income range 50k to 80k. Along the X-axis, `INCOME` is plotted and along the Y-axis, `MORTGAGE_AMOUNT` is plotted. The data is grouped by `MARITAL_STATUS`.

Figure 4-40 Scatter plot



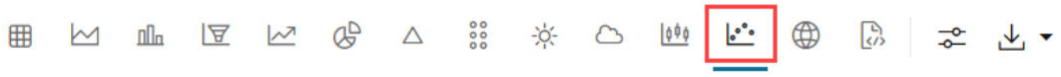
When to use this chart: Use the scatter plot when you have paired numerical data, and you want to determine the relationship between the related variables in certain scenarios, identifying correlations and trends (linear and non-linear relationships), detecting outliers, understanding data distribution, identifying groupings or clusters of data. Scatter plots can also be useful when comparing multiple datasets where each datasets values are represented as a different group. Scatter plots are also useful for evaluating regression models by plotting, e.g., actual versus predicted values.

Data set: `CUSTOMER_INSURANCE_LTV`. In this example, we will use the example template notebook OML-Run-me-first.

To visualize data in a scatter plot:

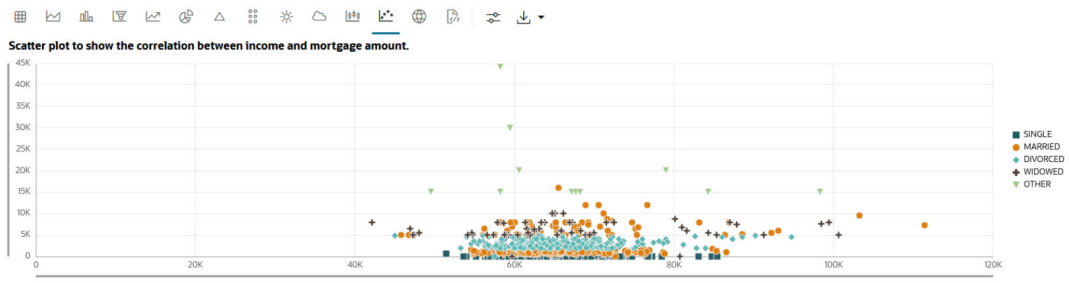
1. In the OML-Run-Me-First notebook, go to the paragraph where you viewed the `CUSTOMER_INSURANCE_LTV`. Click on the Scatter plot icon. A default scatter plot is shown that you will customize in the next step.

   Figure 4-41 Toolbar highlighting the Scatter Plot icon

2. Click the **Settings** icon. In the Settings dialog, under **Setup**:
   - **Series to show on X-axis:** Click and select `INCOME`.
   - **Series to show on Y-axis:** Click and select `MORTGAGE_AMOUNT`.
   - **Group By:** Select `MARITAL_STATUS`.
3. Click **Customization:**
   - **Visualization:** Retain the default settings.
   - **Description:** Under **Title**, enter `Scatter plot to show the correlation between income and mortgage amount.`

Figure 4-42 Scatter Plot



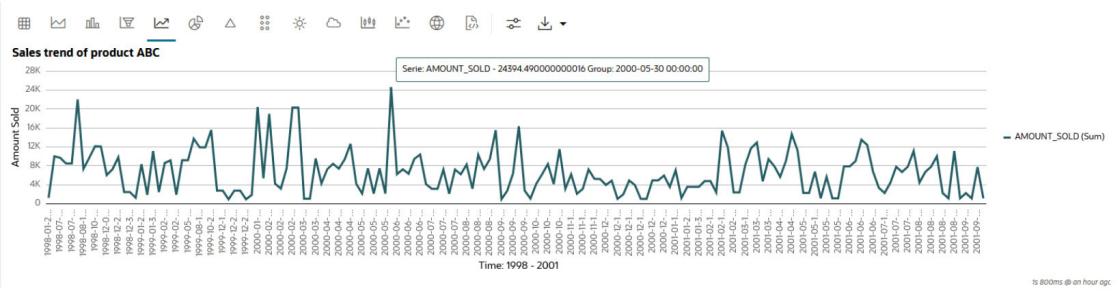
This completes the task of visualizing your data in a scatter plot. The scatter plot shows a strong correlation between Income and Mortgage amount in the income range 50k to 80k.

4.4.6 Visualize Data in a Line Chart

A line chart is a graphical representation used to display data points connected by straight lines.

Here is a line chart displaying the sum of the amount sold from the year 1998 to 2001. It also shows the cursor hovering over the highest point in the line chart to view the values. The image shows that on 5/30/2000, the product recorded the highest sale in terms of the sum of the amount sold.

Figure 4-43 Line chart



When to use this chart: Use this chart to visualize trends, changes, and relationships in data over a continuous period.

Data set: `SH.SALES` table in the `SH` schema.

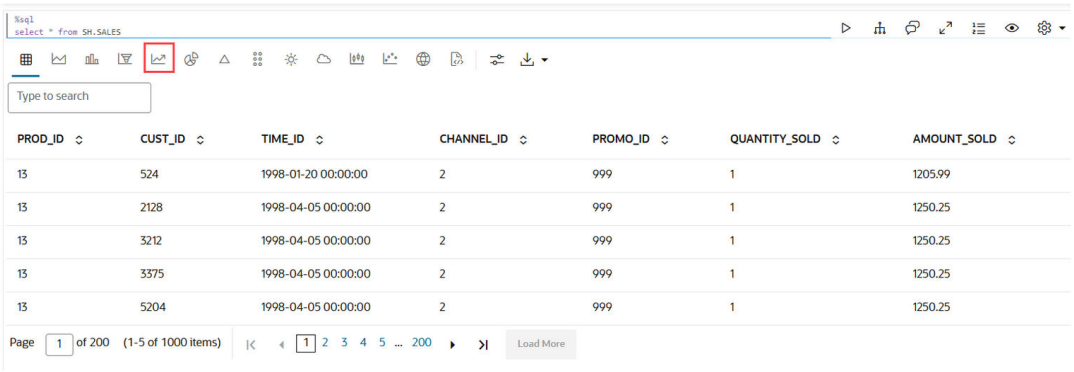
To visualize data in a line chart:

1. Enter the text of the first step here.

```
%sql
select * from SH.SALES
```

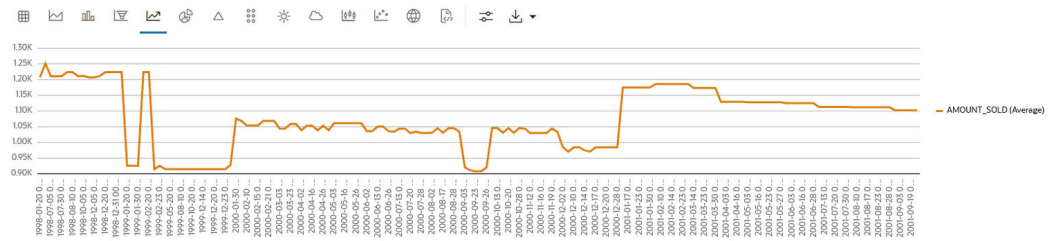
2. By default, the data set is displayed in a table. Click on the line chart icon.

Figure 4-44 Sales Table with the Line Chart icon highlighted



3. By default, the line chart shows the average amount sold from the year 1998 till 2001, as shown in the screenshot below.

Figure 4-45 Line Chart - Depicting the average of amount sold from 1998 - 2001

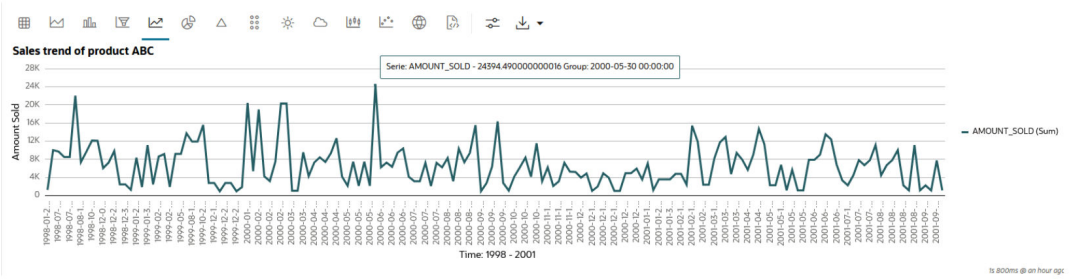


Click on the Settings icon to view the attributes that are plotted along the X and Y axis. The dates on which the product was sold from the year 1998 till 2001 are plotted along the X-axis. Corresponding to each sale date, the average of the amount sold is plotted along the Y-axis.

4. Click on the Settings icon. Under **Setup,** edit the following:
   - **Aggregate Duplicate:** Decides what should happen with values that are within the same group. Select **Sum.** This will show the sum of the amount sold for the product with `PROD_ID` 13, from 1998 to 2001.
   - **Series to Show:** All fields in the result-set that are of type number can be selected. Selecting multiple fields will add additional diagrams to the visualization. Select `AMOUNT SOLD`.
   - **Group By:** All fields in the result-set can be selected. The more groups exist, the more the data set shrinks since it collects all fields and concatenates same values. Select `TIME_ID`.
5. Click **Customization** and edit the following:
   - **X-axis:** Enter Time: `1998 - 2001`. The dates on which the product was sold from the year 1998 till 2001 are plotted along the X-axis.
   - **Y-axis:** Enter `Amount Sold`. Corresponding to each sale date, the sum of the amount sold is plotted along the Y-axis.
   - **Description:** Enter `Sales trend of product ABC`

The line chart now displays the sum of the amount sold from the year 1998 to 2001, as shown below. Hover your cursor over the highest point in the line chart to view the values. You can see that on 5/30/2000, the product recorded the highest sale in terms of the sum of the amount sold.

Figure 4-46 Line chart Line chart - Depicting the sum of amount sold from 1998 - 2001



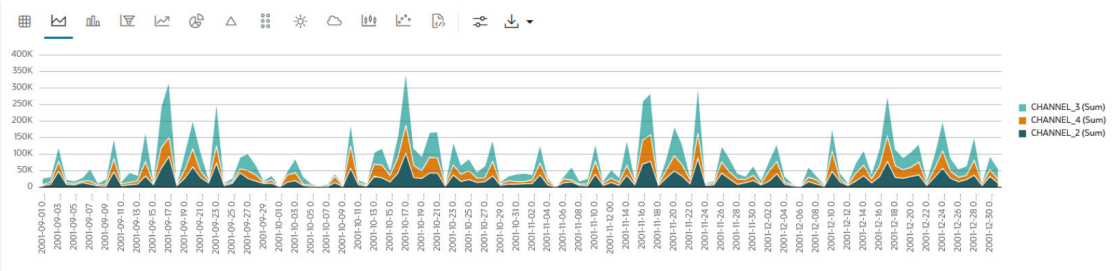
This completes the task of visualizing your data in a line chart.

4.4.7 Visualize Data in an Area Chart

An area chart uses lines to connect the data points and fills the area below these lines to the x-axis. Each data series contributes to the formation of a distinct shaded region. This emphasizes its contribution to the overall trend. As the data points fluctuate, the shaded areas expand or contract.

Here is an area chart rendered for the SALES table. It is customized to depict the sum of the three attributes—`CHANNEL_3`, `CHANNEL_4` and `CHANNEL_9` in stacks. Along the X-axis, time is plotted, and along the Y-axis, sum is plotted.

Figure 4-47 Area chart



When to use this chart: Use this chart to visualize trends, changes, and relationships in data over a continuous period.

Data set: `SALES` table in the `SH` schema.

To visualize data in an area chart:

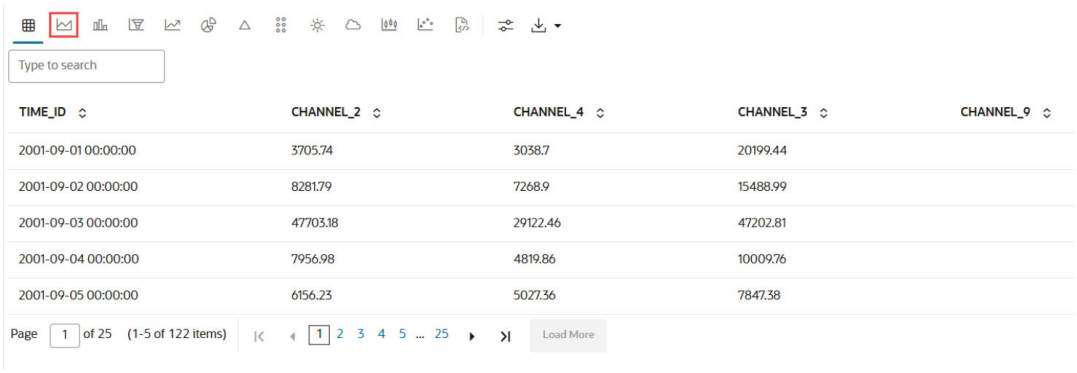
1. In another `%sql` paragraph, run the following script:

```
SELECT
      TIME_ID,
```

```
      -- Use MAX(TOTAL_SOLD) to handle cases with duplicate TIME_ID and
CHANNEL_ID
      MAX(CASE WHEN CHANNEL_ID = 2 THEN TOTAL_SOLD ELSE NULL END) AS
Channel_2,
      MAX(CASE WHEN CHANNEL_ID = 4 THEN TOTAL_SOLD ELSE NULL END) AS
Channel_4,
      MAX(CASE WHEN CHANNEL_ID = 3 THEN TOTAL_SOLD ELSE NULL END) AS
Channel_3,
      MAX(CASE WHEN CHANNEL_ID = 9 THEN TOTAL_SOLD ELSE NULL END) AS
Channel_9
FROM (SELECT TIME_ID, CHANNEL_ID, sum(AMOUNT_SOLD) TOTAL_SOLD
  FROM SH.SALES
      WHERE TIME_ID >= TO_DATE('2001-09-01', 'YYYY-MM-DD')
      GROUP BY TIME_ID, CHANNEL_ID
      ORDER BY TIME_ID)
GROUP BY TIME_ID
ORDER BY TIME_ID
```

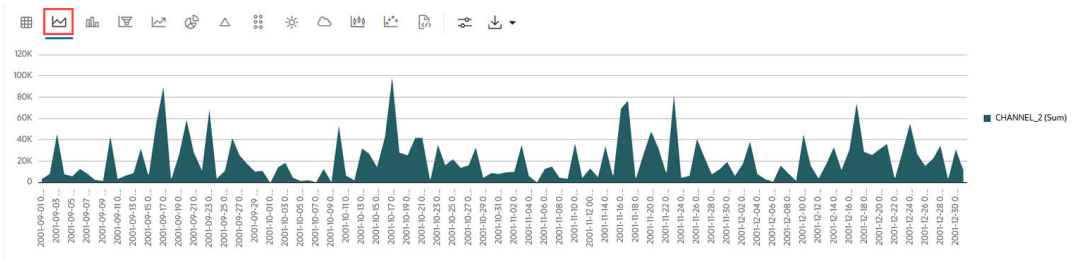
This script groups the data by `TIME_ID` and `CHANNEL_ID`. It presents the data from 2001-09-01 and later. It shows the value for `TOTAL_SOLD` for each of the four channels grouped by `CHANNEL_2`, `CHANNEL_3`, `CHANNEL_4` and `CHANNEL_9`.

2. The data from the `SALES` table is now presented for the following columns—`TIME_ID`, `CHANNEL_2`, `CHANNEL_3`, `CHANNEL_4` and `CHANNEL_9`.



3. Now, click on the the area chart icon Area chart icon in the tool bar to visualize the data in an area chart.

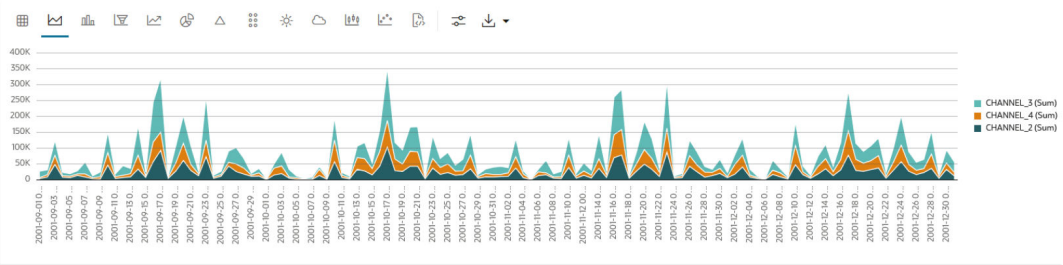
Figure 4-48 Area Chart showing the sum for CHANNEL_2



4. Click on the **Settings** icon to open the Settings dialog. Under **Setup:**

- **Series to Show:** Click to add `CHANNEL_3, CHANNEL_4` and `CHANNEL_9`.
- **Group By:** Retain the default, that is, `TIME_ID`.
- **Aggregate Duplicates:** Retain the default, that is, `SUM`.

5. Click **Customization** and under **Visualization**, click **Stacked.** The area chart is now presented as shown in the screenshot.



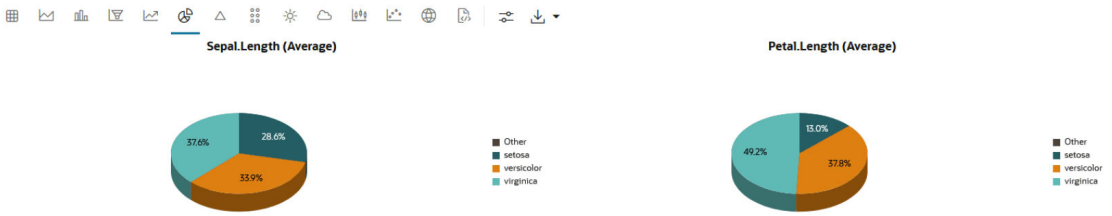
This completes the task of visualizaing your data in an area chart.

4.4.8 Visualize Data in a Pie Chart

A pie chart is a graphical representation of data in a circular form, with each slice of the circle representing a fraction that is a proportionate part of the whole.

Here are two pie charts generated for the IRIS dataset. The pie charts have been customized to render the visualization in 3D. One pie chart shows the average sepal length, and the other shows the average petal length for each of the three species of the iris flower.

Figure 4-49 Pie chart



When to use this chart: Use this chart to visualize the numerical proportion of the parts to the whole.

Data set: The `IRIS` data set. The `IRIS` data set contains 3 classes. These classes are the three different Iris species—Setosa, Versicolor, and Virginica. The data set has 50 samples each, and four numeric properties about those classes—Sepal Length, Sepal Width, Petal Length, and Petal Width.

To visualize data in a pie chart:

1. Run the following script in an R paragraph to create the IRIS data set:

```
%r
library(ORE)
```

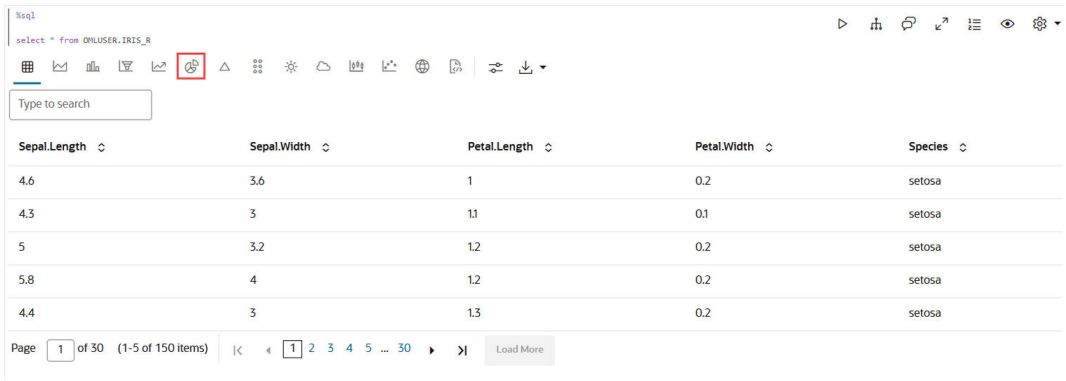
```
if (ore.exists("IRIS_R")) ore.drop(table="IRIS_R")

ore.create(iris, table = "IRIS_R", overwrite=TRUE)

ore.exec("ANALYZE TABLE IRIS_R COMPUTE STATISTICS")

z.show(cat("Shape:", dim(IRIS_R)))
```

This creates the table `IRIS_R`.

2. Now, run the following SQL command to view the data set.

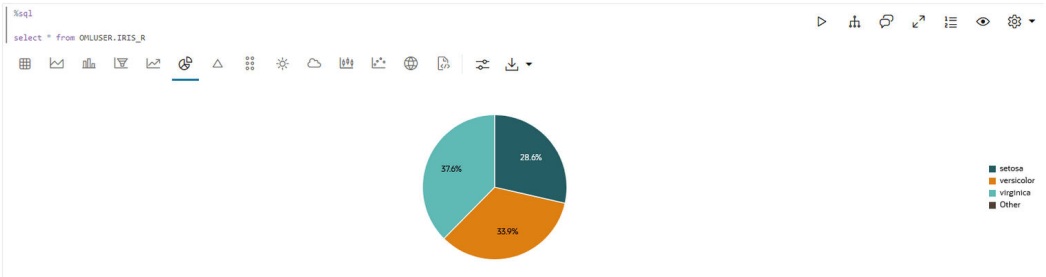
```
%sql
select * from OMLUSER.IRIS_R
```

3. By default, the data set is displayed in a table. Click on the pie chart icon.



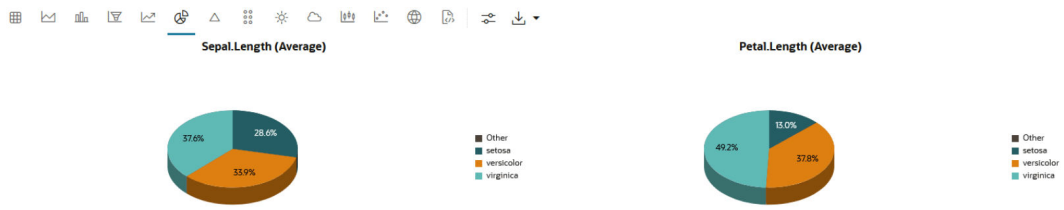
4. The data is now displayed in a pie chart. By default, the pie chart shows the average sepal length for each of the three species of iris. It also shows a 5% threshold for others.

Figure 4-50 Pie Chart—Depicts the average sepal length for the three classes



5. Click on the **Settings** icon to open the Settings dialog. Click **Customization.**
   - **Variant:** Click **Donut.**
   - **Inner Radius:** Click the up arrow and set it to `40`.
   - **Label:** Type `Sepal Length of the 3 Iris species.`
   - Close the dialog. The data is now rendered in a donut chart, as shown below:

6. Once again, click on the Settings icon. Under **Setup**, click **Series to Show** and then select **Petal Length** while retaining **Sepal Length.**
7. Click Customization:
   - **Variant:** Click **Pie.**
   - **Dimension:** Click **3D.**
   - **Sorting:** Click **Ascending.**



The data is now displayed in two pie charts in 3D, one showing the average sepal length, and the other showing the average petal length for each of the three species of the iris flower.

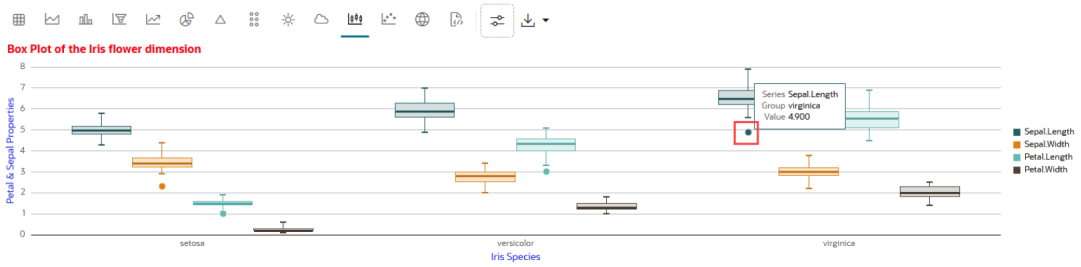
This completes the task of visualizing your data in a pie chart.

4.4.9 Visualize Data in a Box Plot

A box plot provides an overview of data distributions in numeric data. It provides general information about the symmetry, skewness, variance, and outliers in a dataset. The box plot uses boxes and lines to depict the data distribution.

Here is a box plot generated for the IRIS dataset. The box plot depicts the numeric distribution of the dimensions (`Sepal Length`, `Sepal Width`, `Petal Length`, and `Petal Width`) of the three species of the iris flower— Setosa, Versicolor, and Virginica. The basic box plot (Box plot 1) shows the numeric distribution of the 3 species Setosa, Versicolor, and Virginica. This box plot has been customized to display the outlier value for each of the three species. In this image, the cursor is over the outlier dot for the class `Virginica`. It highlights the outlier value at 4.9 for the sepal length of Virginica. This implies that in the species Virginica, there are sepals whose length is significantly below the lower count 5.6.

Figure 4-51 Box plot



The box plot has the following components:

- **Central Box:** Inter-quartile range and quartiles:
  - **Q1 (First Quartile):** This is the value below which 25% of the data falls. It represents the boundary between the lowest 25% and highest 75% of values.
  - **Q3 (Third Quartile):** This represents the value below which 75% of the data falls, serving as a border between the lowest 75% and highest 25% of values.
  - **Interquartile Range (IQR):** The IQR is the range in which the central 50% of the values fall. IQR = Q3 - Q1
- **Whiskers:** The whiskers of the box plot extend from the central box to the minimum and maximum data values that are not considered outliers. They provide a graphical representation of the majority of the data distribution.
- **Outliers:**Outliers are data points that deviate significantly from other data points, typically due to data variability or errors. An outlier is plotted as a dot beyond the ends of the whiskers of a box plot.
- **Median:** The median is the value that divides the dataset into two halves, with 50% of the values falling below it and 50% falling above it. In the box plot, a line or a mark inside the central frame represents the median.

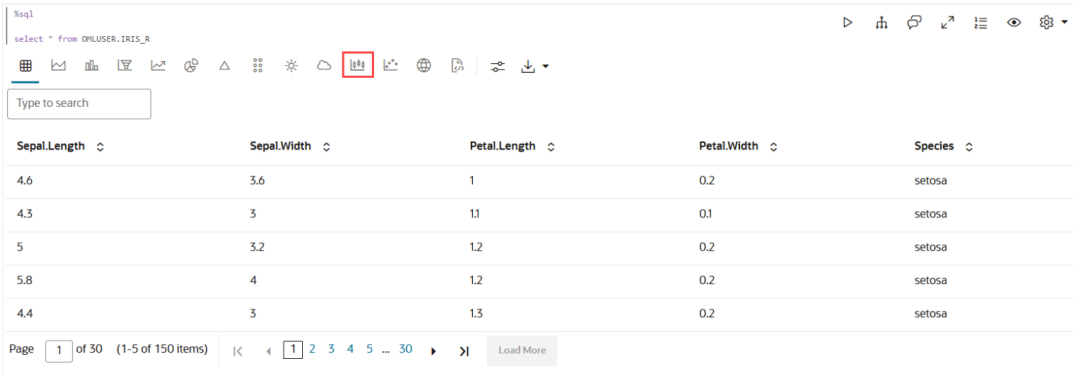
When to use this chart: Use this chart to show distributions of numeric data, especially if you want to compare them between multiple groups.

Data set: `IRIS` dataset. The `IRIS` dataset contains 3 classes, the three different Iris species—Setosa, Versicolor, and Virginica, along with 50 samples each, and four numeric properties about those classes: Sepal Length, Sepal Width, Petal Length, and Petal Width.

To visualize data in a box plot:

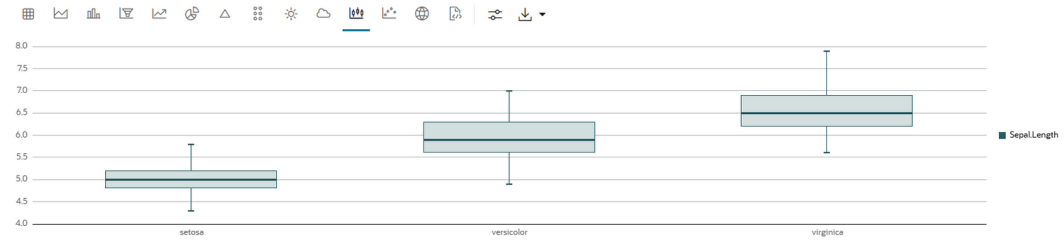
1. We have already created the `IRIS` dataset in the topic *Visualize your data in a pie chart*. We will use the same table `IRIS_R` to visualize the data in a box plot. Open the notebook and go to the paragraph where the `IRIS_R` table is populated.

Figure 4-52 IRIS dataset in a table with boxplot icon highlighted



2. Click the box plot icon. The dataset is now displayed in a box plot.

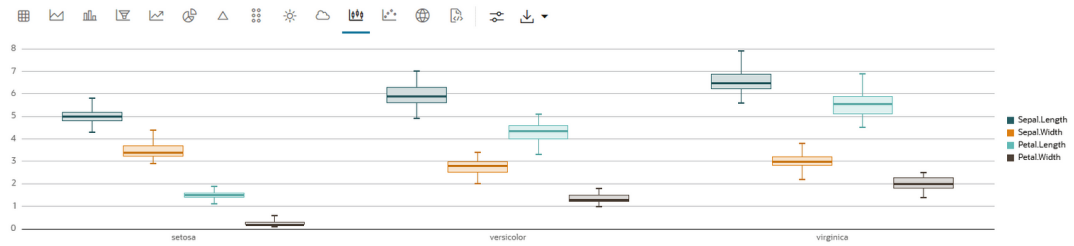
Figure 4-53 Box plot 1— Depicts the data grouped by the 3 species Setosa, Versicolor, and Virginca



As you can see, by default the data is grouped by the 3 species (classes) - Setosa, Versicolor, and Virginca along the X-axis, and the sepal length along the Y axis. Hover your cursor over each box plot to view the count.

3. Click on Settings to view how the data is plotted. Under **Setup**, go to **Series to Show**, and click to add the other three numeric properties—`Sepal Width`, `Petal Length`, and `Petal Width`.

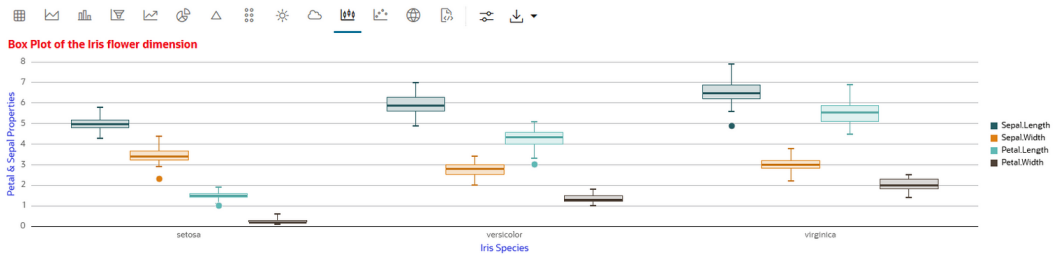
Figure 4-54 Box Plot 2 - Depicts the Iris 3 species and their properties Sepal Width, Sepal Length, Petal Width, and Petal Length



4. Under Settings, click **Customizations**, edit the following settings:
   - **Visualization:** Click **Show Outliers.**

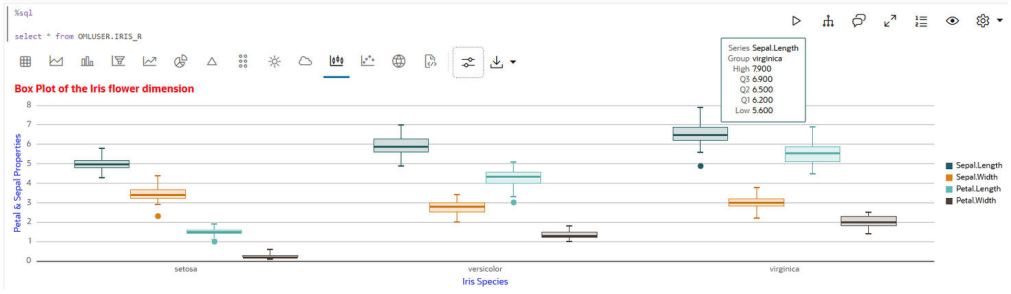
- In the **Text** field, enter `Iris Species`. **Color:** Enter `rgb(7, 17, 215, 0.88)`
- **Y-Axis:** In the **Text** field, enter `Petal & Sepal Properties`. **Color:** Enter `rgb(7, 17, 215, 0.88)`
- **Description:** Enter the following - `Box Plot of the Iris flower dimension.`
- **Color:** Enter `rgb(241, 8, 24)`
- Once done, close the dialog.

Figure 4-55 Box Plot 3 - Shows the Outlier, Box Plot description, and descriptions for X-Axis and Y-Axis



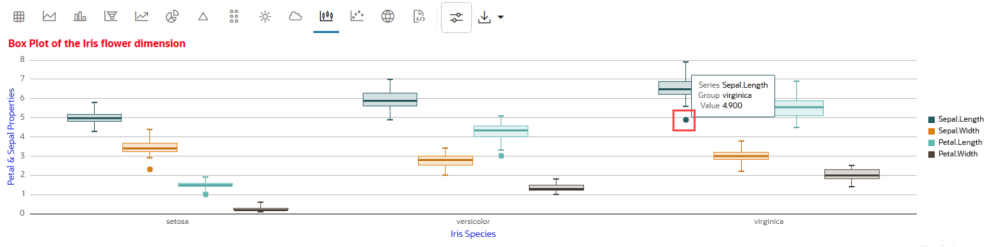
5. The box plot now displays the dataset as below:
   - Hover your cursor over each box plot to view the values. In the screenshot here, the cursor is over the Sepal Length series for the species Virginica. The length ranges from 5.6 to 7.9. There is also an outlier for this, and it is indicated by the dot below the box plot whisker.

Figure 4-56 Box Plot 4 - Shows the value for the class Virginica and property Sepal Length



   - Hover your cursor over the dot that indicates the outlier for the group virginica. It shows the outlier value at 4.9 for Virginica sepal length. This means that in the species Virginica, there are sepals whose length is significantly below the lower count (5.6).

Figure 4-57 Box Plot 5 - Shows the Outlier value for Virginica (Class) Sepal Length (Property)



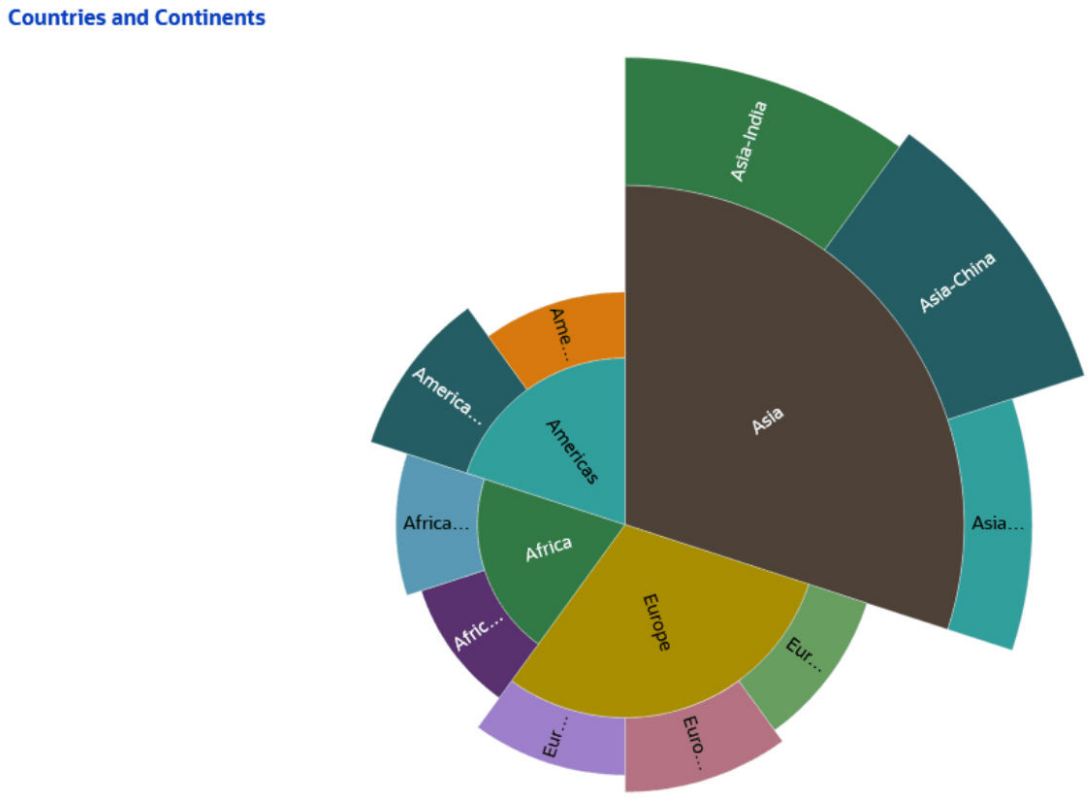
This completes the task of visualizing your data in a box plot.

4.4.10 Visualize your Data in a Sunburst Chart

The sunburst chart is typically used to visualize hierarchical data structures - with part-to-whole relationships in data depicted additionally.

This is a Sunburst chart that depicts the four continents in four slices. The slices are of different sizes and color. After customization, the slices are segmented to depict the countries of that continent. The size of the slices is determined by the count in the `Population` column.

Figure 4-58 Sunburst chart



When to use this chart: Sunburst charts handle multi-level pie chart data more effectively than regular pie charts. Sunburst charts enhance efficiency and clarity by integrating the functionality of multiple pie charts into a single visual.

Data set: Custom dataset on continents, country and population.

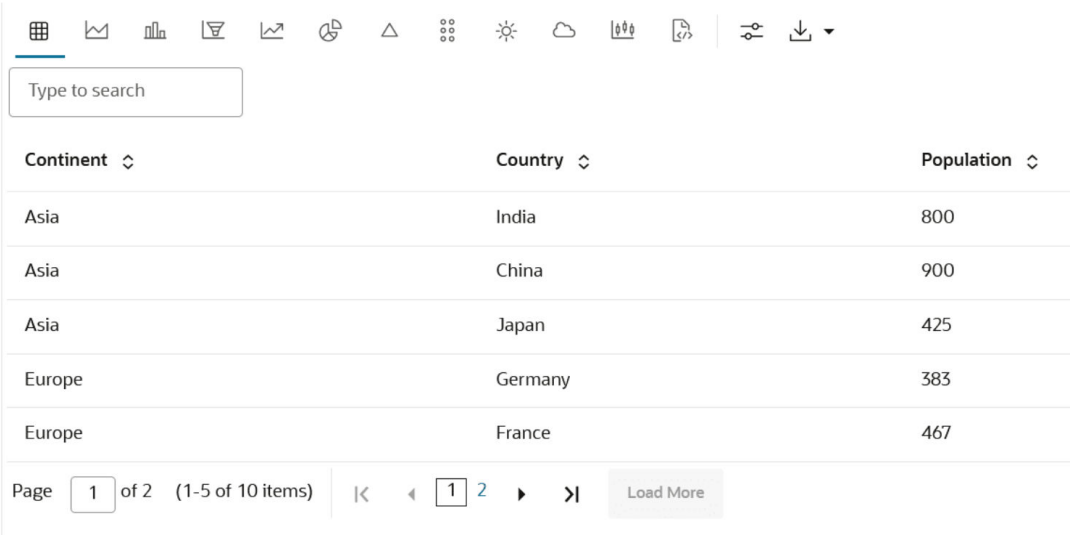
To visualize your data in a sunburst chart:

1. Open a notebook and in a `%python` paragraph, import the following python libraries—`oml` and `pandas`.
2. Run the following command to create a dataset on continent, country and population:

```
%python

data = [
    ["Asia", "India", 800],
    ["Asia", "China", 900],
    ["Asia", "Japan", 425],
    ["Europe", "Germany", 383],
    ["Europe", "France", 467],
    ["Europe", "Italy", 360],
    ["Africa", "Nigeria", 416],
    ["Africa", "Egypt", 510],
    ["Americas", "USA", 631],
    ["Americas", "Brazil", 413]
]

print("Continent\tCountry\tPopulation")
for row in data:
    print(f"{row[0]}\t{row[1]}\t{row[2]}")
```

3. Once the code is run, the data is displayed in a table with three columns—Continent, Country and Population.

Figure 4-59 Table

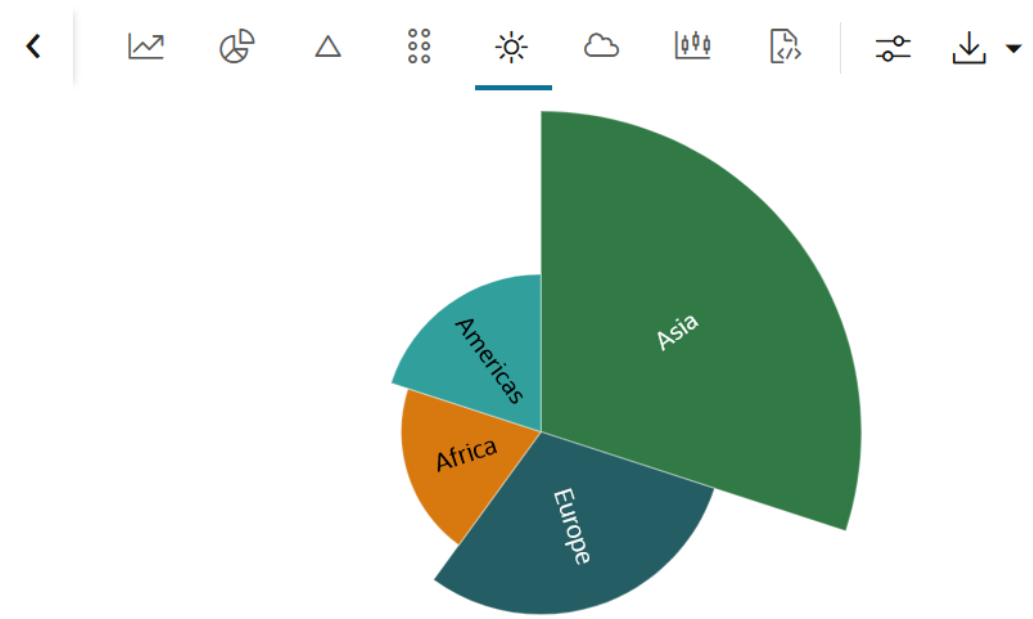


4. Click on the sunburst icon.

Figure 4-60 Sunburst icon

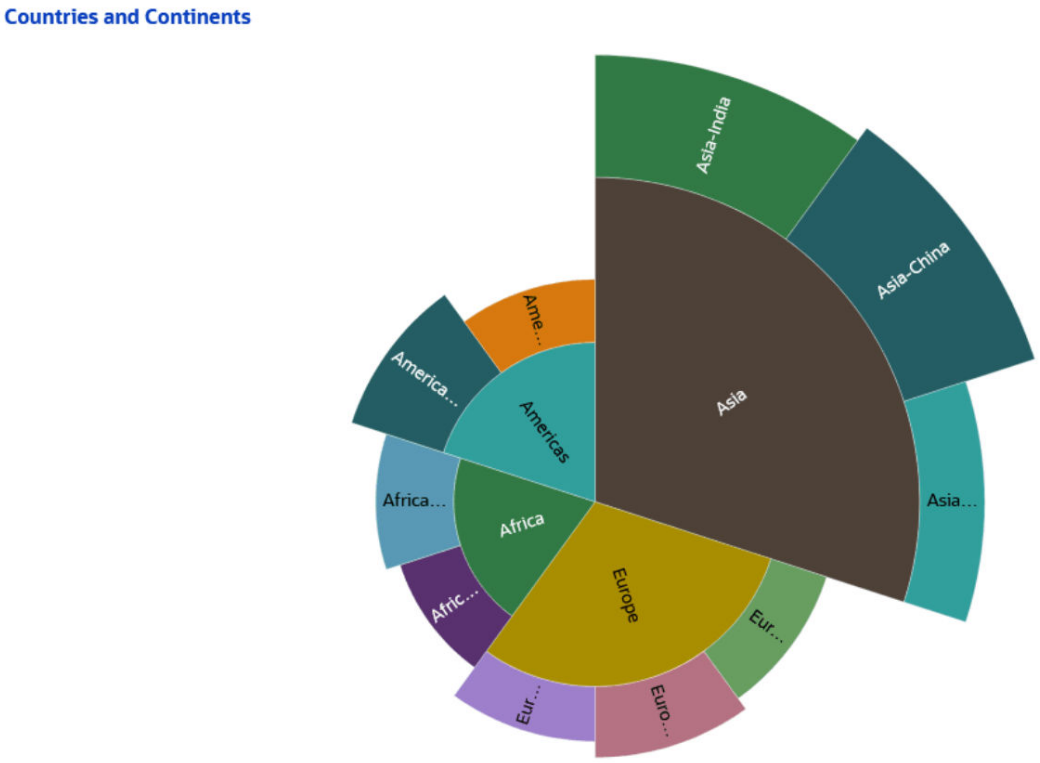
5. The data is now displayed in a sunburst chart as shown in the screenshot here. The four continents are depicted in four slices of the sunburst chart. The slices are of different sizes and color. The size of the slices is determined by the count in the Population column.

Figure 4-61 Sunburst chart



6. Click  and on the **Setup** tab, enter the following:
   a. **Aggregate Duplicates:** Click and select `Count`.
   b. **Radius Column:** The value provided in this field defines the radius by its value. Click to add `Population`. If Population is not available for selection, type `Population`.
   c. **Group By:** Click to add `Country` to it. The countries are now depicted in smaller segments of the slices representing the Continents. By default, `Continents` is already added to the **Group By** field.
7. Click **Customization** edit the following under **Description**:
   - **Title Setup:** Enter the text `Countries and Continents`.
   - **Color:** Click on the color palette. Move your cursor and click on a color of your choice. Click **Select**.
8. The sunburst chart now displays the data in the Country and Continent columns in segmented slices of the sunburst chart. Inside each slice representing the four continents, the countries are depicted by separate segments of different color and sizes. This shows the relation between countries and continents. Hover your cursor over each segment to view more details.

Figure 4-62 Sunburst chart after customization

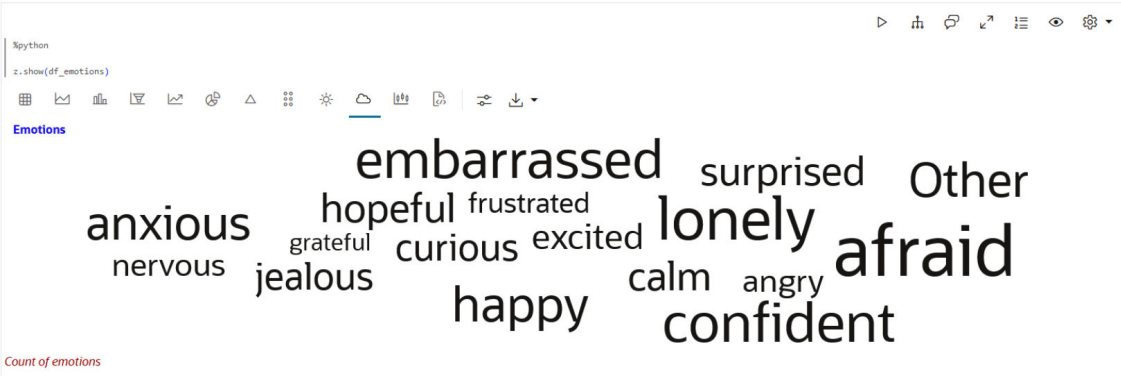


4.4.11 Visualize your Data in a Tag Cloud

A tag cloud is a visual representation of the most popular words or tags found in free-form text. The font size or the color of the text represents the frequency of occurrence.

Here is a tag cloud representing some emotions picked up from user comments. The font size implies the average count of the emotions.

Figure 4-63 Tag cloud



When to use this chart: Use Tag Cloud to visualize text data, and to conduct word-frequency analysis across tags, keywords etc.

Data set: In this example, we create a data set that contains user comments.

To visualize your data in a tag cloud:

1. Open a notebook and in a `%python` paragraph, enter the following code and run the paragraph:

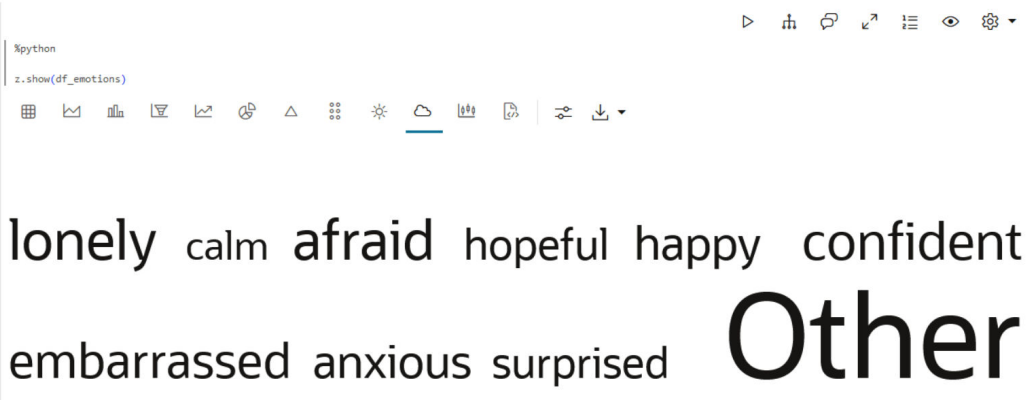
```
import pandas as pd
import random

emotions = ["happy", "sad", "angry", "excited", "nervous", "calm",
"anxious", "confident",
            "bored", "curious", "frustrated", "hopeful", "jealous",
"lonely", "grateful",
            "afraid", "embarrassed", "relieved", "surprised", "proud"]

weights = [random.randint(1, 10) for _ in emotions]
total_weight = sum(weights)
probabilities = [w / total_weight for w in weights]

random.seed(21)
emotion_samples = random.choices(emotions, weights=probabilities, k=500)

df_emotions = pd.DataFrame(emotion_samples, columns=["emotion"])
df_emotions.insert(0, "id", range(1, len(df_emotions) + 1))
```

   This code generates a large collection of strings (emotions) and then prints each one with a unique ID, formatted in a two-column table.

2. Click on the Tag cloud icon to view the data in a tag cloud.

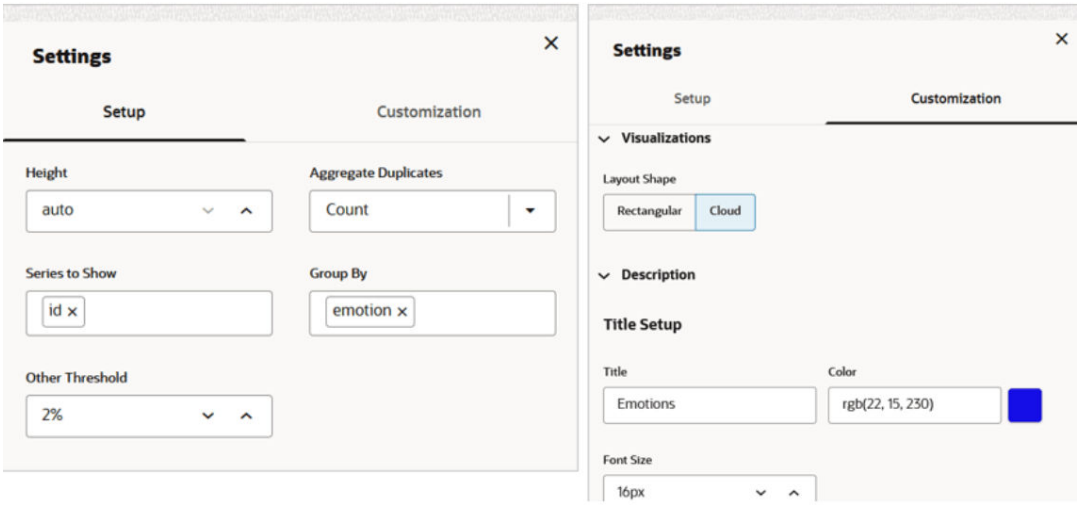
   Here is the image of a default tag cloud output. The data is generated in a rectangular layout.

Figure 4-64 Tag Cloud in a Rectangular layout



3. Click on the Settings icon. This opens the Settings dialog. It has the Setup and Customization tabs.

Figure 4-65 Settings - Tag Cloud



On the Setup tab, you can configure the columns or series to display, and other settings:

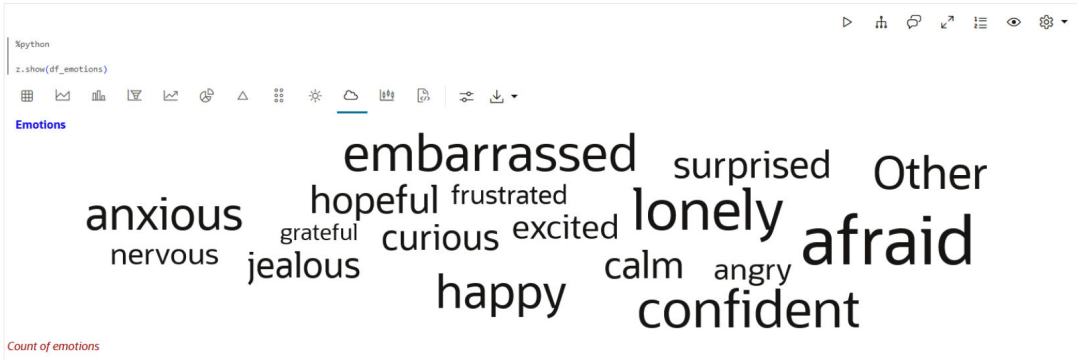
- **Height:** This parameter determines the height of the visualization.
- **Aggregate Duplicates:** Determines the computation of the values for display. The available values are —`Average`, `Sum`, `Maximum`, `Minimum`, `Count`, and `Last`.
- **Series to Show:** All applicable fields in the result-set are available for selection. Selecting multiple fields will add additional diagrams to the visualization.
- **Group By:** All applicable fields in the result-set are available for selection. The more groups exist, the more the data set shrinks since it collects all fields and concatenates the same values.
- **Other Threshold:** All applicable fields in the result-set are available for selection.

On the **Customization** tab, you have the option to customize the Title setup, Subtitle setup, and Footnote setup.

- **Visualization:** The options are Rectangle and Cloud. Click **Cloud**.
- **Title Setup:** Enter values in each of these fields to customize the appearance. The fields are **Title**, **Color**, **Font Size**.
- **Subtitle Setup:** Enter values in each of these fields to customize the appearance. The fields are **Subtitle**, **Color**, **Font Size**.
- **Footnote Setup:** Enter values in each of these fields to customize the appearance. The fields are **Footnote**, **Color**, **Font Size**.

Here is a tag cloud after customizing the settings. It shows the average count of the emotions. You may hover your cursor to view the count. Note the change in layout format and the title in blue.

Figure 4-66 Tag Cloud in a Cloud layout

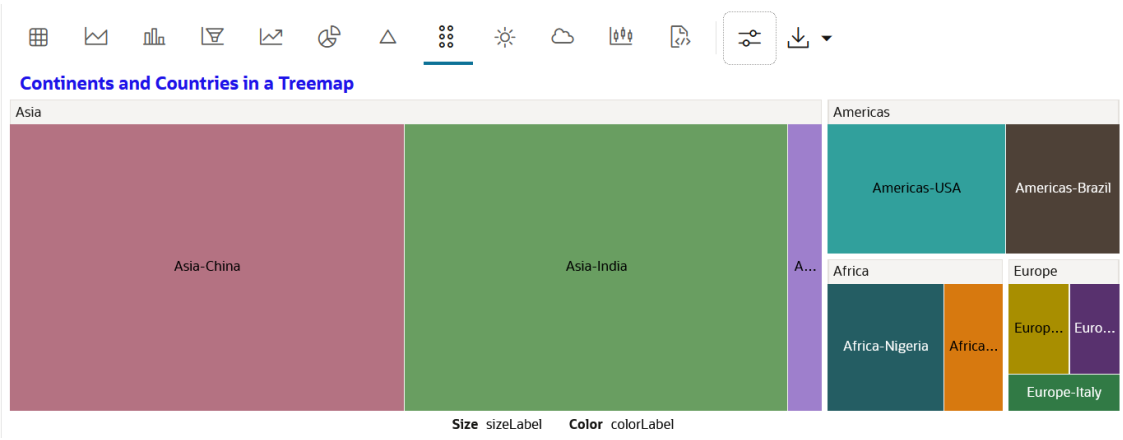


4.4.12 Visualize your Data in a Treemap

A treemap is a visualization composed of nested rectangles, that represent certain categories within a selected dimension and are ordered in a hierarchy, or “tree.” Quantities and patterns can be compared and displayed in a limited chart space.

Treemaps also show the relationship of each part (or nested rectangles) to the whole (tree). Here is a treemap depicting the relationship between the parts (countries) and the whole (continents).

Figure 4-67 Treemap



When to use this chart: Use this chart to visualize a large number of related categories, and also to analyze the part-to-whole relationship in your data set.

Data set: Custom dataset on continents, country and population.

To visualize your data in a tree map:

1. Open a notebook and in a `%python` paragraph, import the following python libraries—`oml` and `pandas`.
2. Run the following command to create a dataset on continent, country and population:

```
%python

data = [
```

```
    ["Asia", "India", 1400],
    ["Asia", "China", 1440],
    ["Asia", "Japan", 125],
    ["Europe", "Germany", 83],
    ["Europe", "France", 67],
    ["Europe", "Italy", 60],
    ["Africa", "Nigeria", 216],
    ["Africa", "Egypt", 110],
    ["Americas", "USA", 331],
    ["Americas", "Brazil", 213]
]

print("Continent\tCountry\tPopulation")
for row in data:
    print(f"{row[0]}\t{row[1]}\t{row[2]}")
```

3. Once the code is run, the data is displayed in a table with three columns—Continent, Country and Population.

Figure 4-68 Table

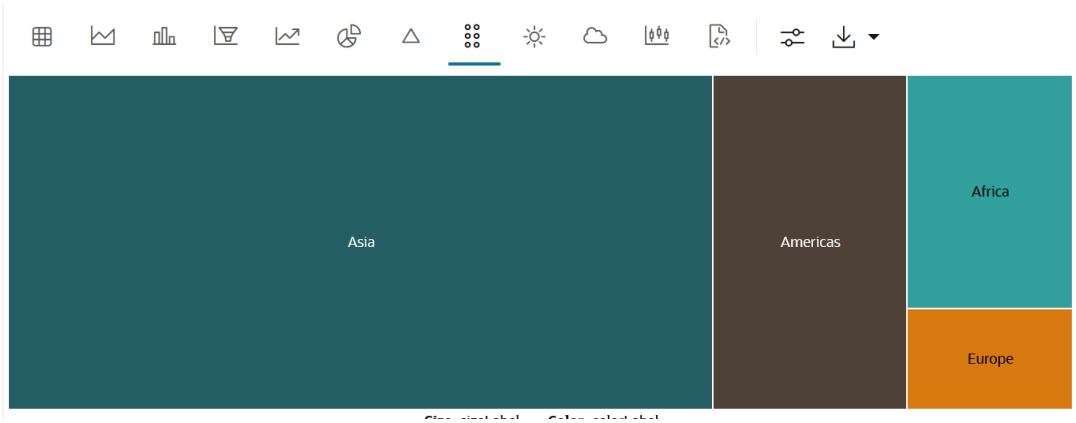
Continent	Country	Population
Asia	India	1400
Asia	China	1440
Asia	Japan	125
Europe	Germany	83
Europe	France	67
Europe	Italy	60
Africa	Nigeria	216
Africa	Egypt	110
Americas	USA	331
Americas	Brazil	213

4. Click on the tree map icon.

Figure 4-69 Treemap icon on the toolbar

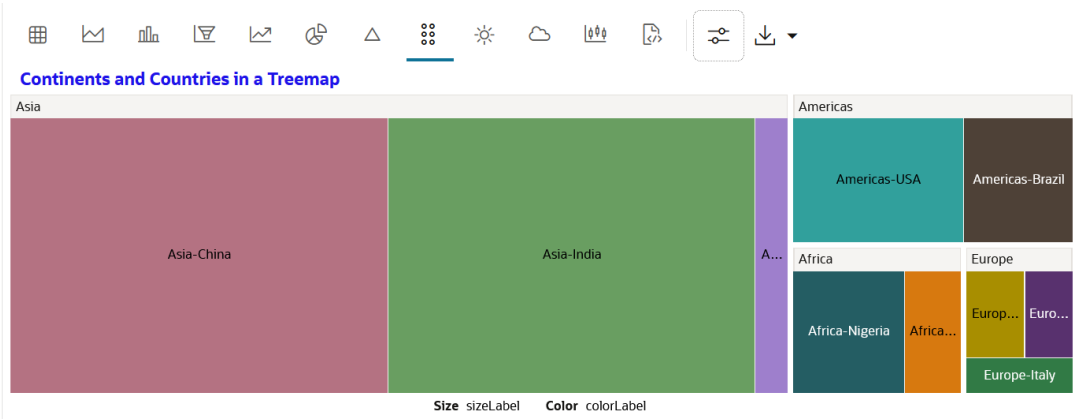
5. The data is now displayed in a treemap as shown in the screenshot here. The treemap represents the four continents in four rectangles of different sizes and color. The size is defined by the count in the Population column.

Figure 4-70 Treemap



6. Click  and on the Setup tab, click the **Group By** field. Click on `Country` to add it. Now, the countries are depicted in nested rectangles inside the super set rectangles representing the Continents. By default, `Continents` is already added to the Group By field.
7. Click on the Customization tab, edit the following under **Description**:
   - **Title Setup:** Enter the text `Continents and Countries in a Treemap`.
   - **Color:** Click on the color palette. Move your cursor and click on a color of your choice. Click **Select**.
8. The Treemap now displays the relation between countries and continents in nested rectangles. The continents are represented as supersets. And the countries of the continent are depicted as nested rectangles inside the rectangles representing the continents. Hover your cursor over each rectangle to view more details.

Figure 4-71 Treemap after customization



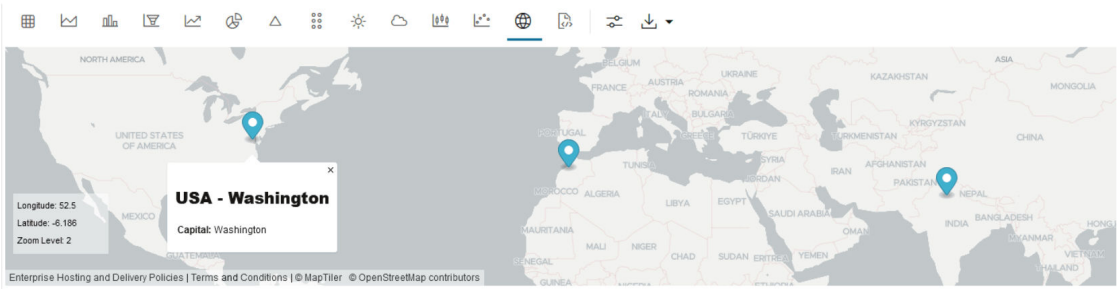
4.4.13 Visualize Data in a Map

The map visualization of OML Notebooks plots data points on a geographical map. For visualizing your data in a map in OML Notebooks, your data must contain explicit latitude and longitude values of locations to correctly position data on the map.

Here is a visualization of a dataset containing information about countries, capital, geographical coordinates and other related information. The displayed map type is OSM

Positron style. It has the pushpin on USA clicked. The marker dialog displays the country name and its capital.

Figure 4-72 Map



When to use this chart: Use map charts to visualize your data with geographical implications

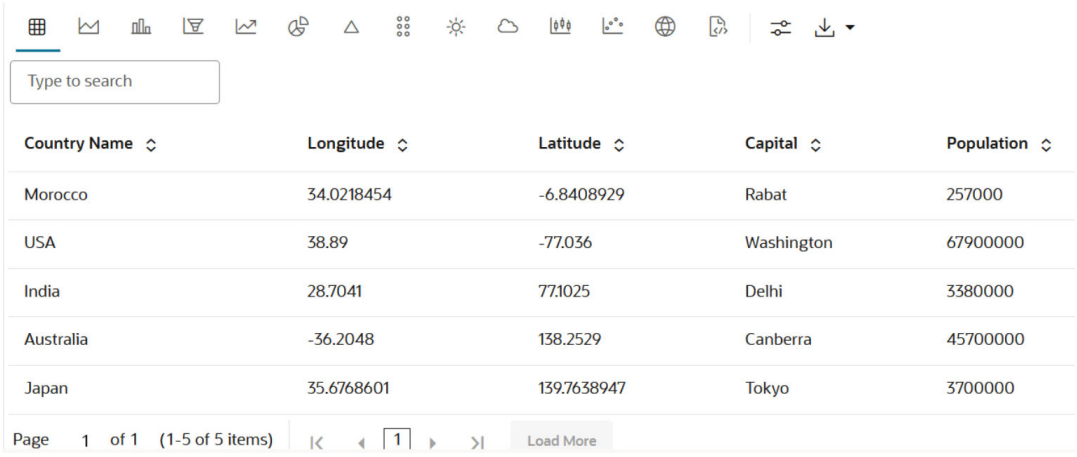
Data set: In this example, we create a table containing data about countries and related information with geographical coordinates.

To visualize data in a map:

1. In a Python paragraph in your notebook, run the following statements to create a table with five columns—`Country Name`, `Longitude`, `Latitude`, `Capital`, and `Population` with five entries for each:

   ```
   %python
   print('Country Name\tLongitude\tLatitude\tCapital\tPopulation')
   print('Morocco\t34.0218454\t-6.8408929\tRabat\t257000')
   print('USA\t38.89\t-77.036\tWashington\t67900000')
   print('India\t28.7041\t77.1025\tDelhi\t3380000')
   print('Australia\t-36.2048\t138.2529\tCanberra\t45700000')
   print('Japan\t35.6768601\t139.7638947\tTokyo\t3700000')
   ```

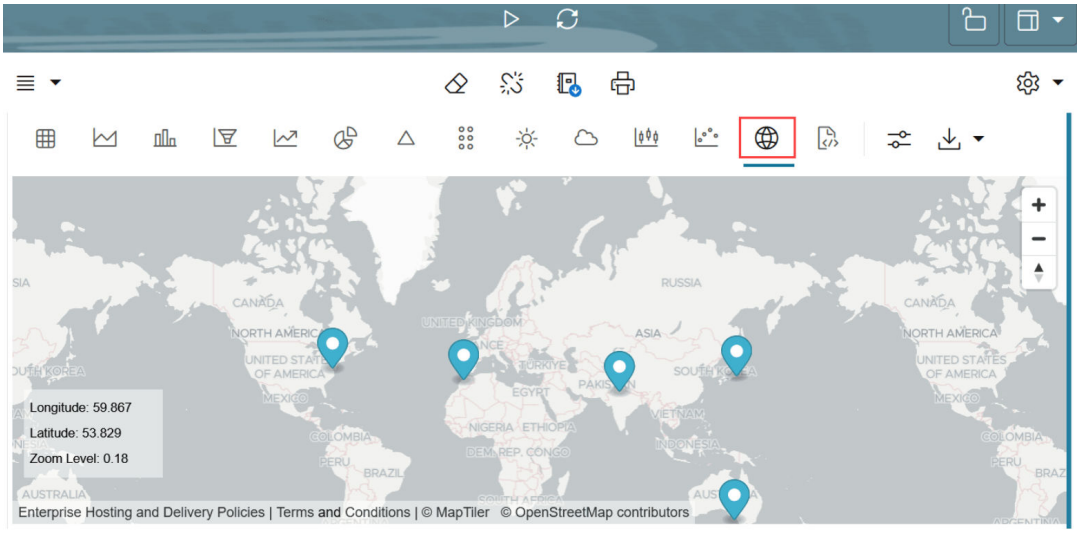
   A table is created with these columns—`Country Name`, `Longitude`, `Latitude`, `Capital`, and `Population`.

2. Click on the map icon.

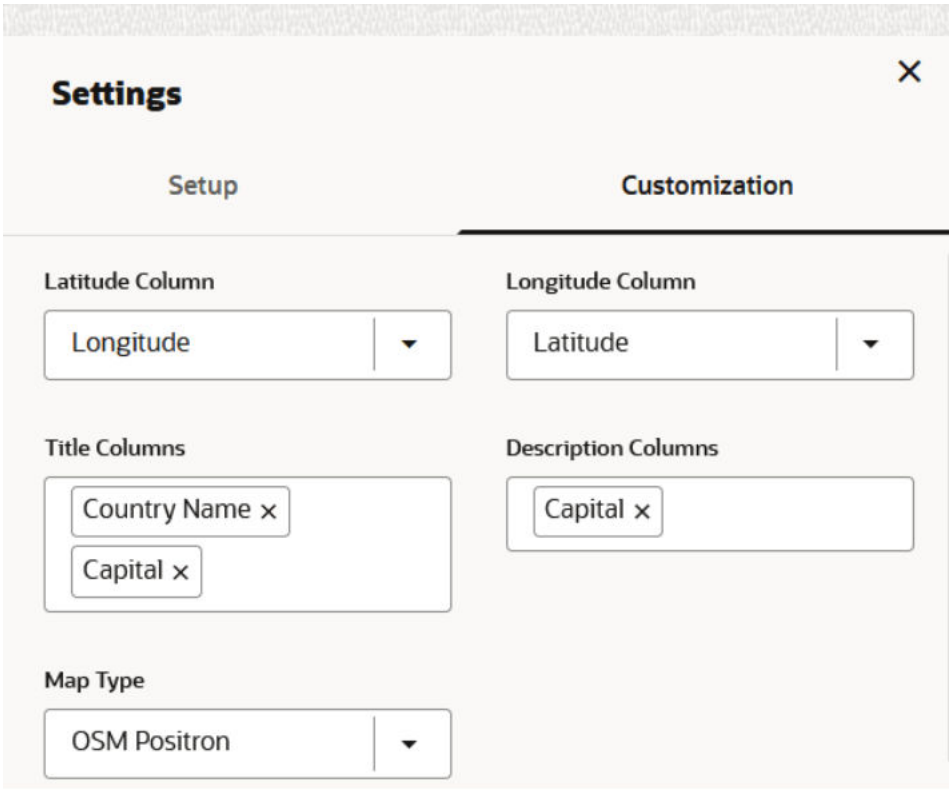
The data is displayed on a world map in OSM Positron style. This is a non-obtrusive light vector tile basemap based on OpenStreetMap (OSM) data. This is the default style.

Figure 4-73 Map



3. Under Settings, on the **Setup** tab, you can adjust the height of the map. You can also show and hide the zoom controls by clicking **Show Zoom Control**.
4. Under Settings, in the **Customizations** tab, you can edit the following settings. Click on the arrow to change any entries for these columns.

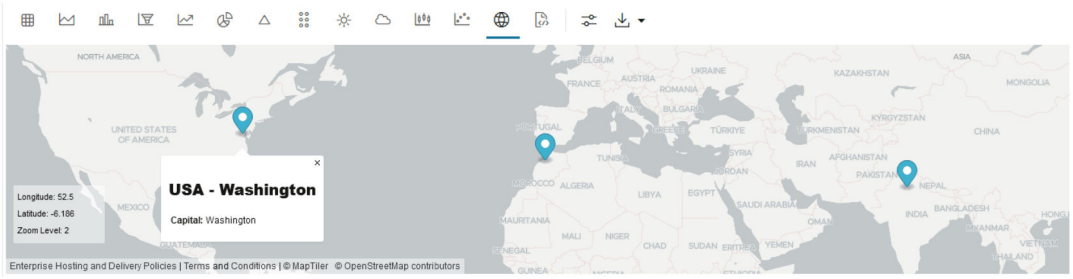
Figure 4-74 Map settings



- **Latitude column:** Select the column from the dataset which can be considered as latitude value. The column must contain numeric values (float or integer) with valid geographic coordinates. Latitude values must be between -90 and 90.
- **Longitude column:** Select the column from the dataset which can be considered as longitude value. The column must contain numeric values (float or integer) with valid geographic coordinates. Longitude values must be between -180 and 180.
- **Title Columns:** Select one or more columns from the dataset to be displayed as a tooltip label and marker. If you select two columns, it will be concatenated with a dash and displayed in the marker dialog that opens when you click on any pushpin on the map.
- **Description Columns:** Select one or more columns from the dataset to be displayed in the marker dialog that opens when you click on any pushpin on the map.

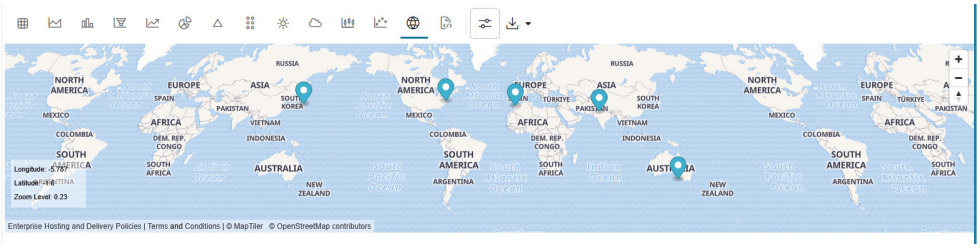
As shown in the Map Settings screenshot above, the **Title Columns** have Country Name and Capital selected, and the **Description Columns** have Capital selected. On clicking the pushpin for USA on the map, the marker dialog displays **USA - Washington** and **Capital: Washington** displayed based on the selections in Settings—Customization.

Figure 4-75 Map with the marker dialog



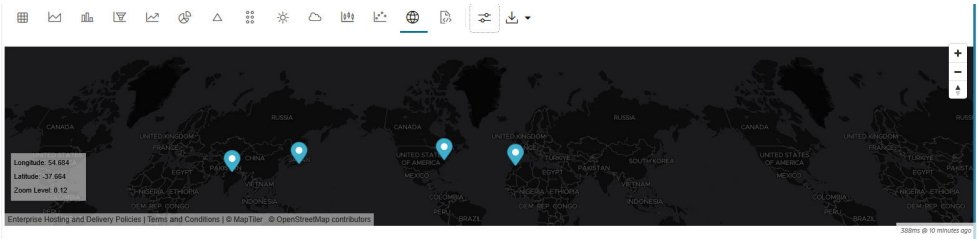
- **Map Type:** By default, the `OSM Positron` type is selected. This is a map style built using the OpenStreetMap (OSM) data. The other map types you can choose are:
  - **OSM Bright:** This is a general purpose vector tile basemap based on OpenStreetMap (OSM) data. Use this basemap style to view detailed location context for your data. Here is an example of an OSM Bright map type:

Figure 4-76 Map type - OSM Bright



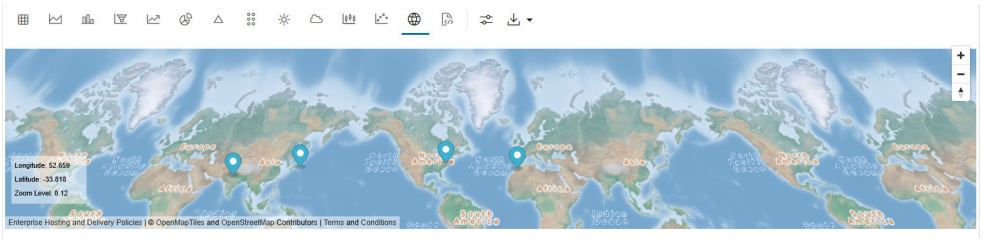
  - **OSM Dark Matter:** This is a non-obtrusive dark vector tile basemap based on OpenStreetMap (OSM) data. Use this basemap style to accentuate visualizations of your data. Here is an example of an OSM Dark Matter map type:

Figure 4-77 Map type - OSM Dark Matter



– **World Map:** This is a 2D physical map, a type of map where the geographical information is displayed in color. Use this map style to visualize the geography of a region in terms of elevation, landforms and other geospatial information. Here is an example of a two-dimensional physical world map:

Figure 4-78 Map Type - 2D Physical World Map



– **Customize Type:** Here, you have the option to change the map source and map style.

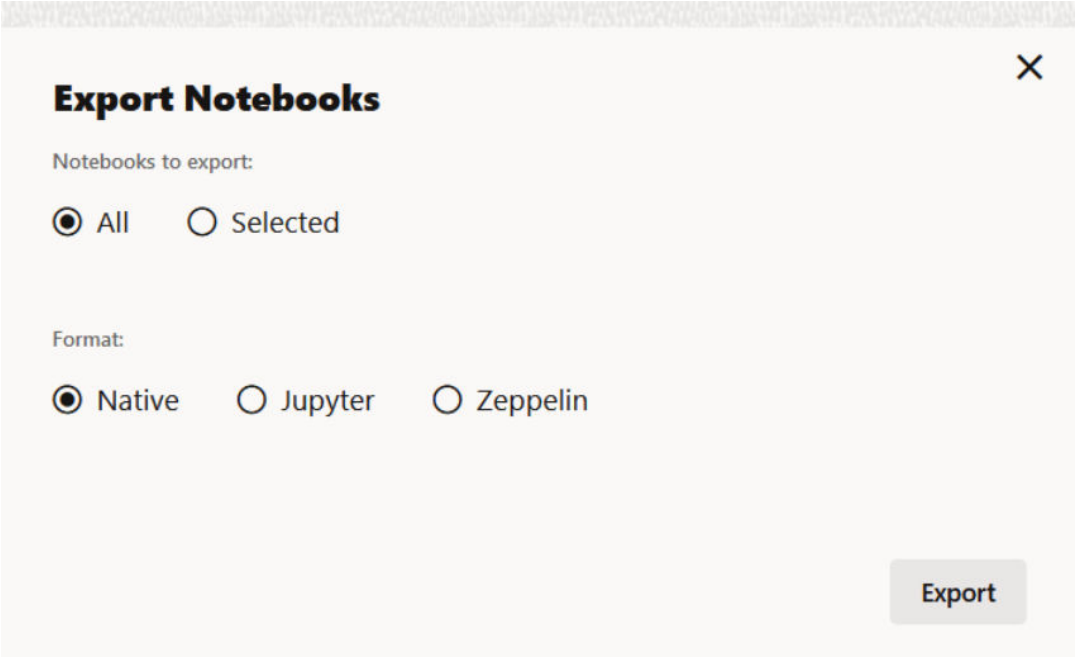
This completes the task of visualizing data in a map.

4.5 Export a Notebook

You can export a Notebook in Native format (`.dsnb`) file, Zeppelin format (`.json`) file, in Jupyter format (`.ipynb`), and later import them in to the same or a different environment.

To export a Notebook:

1. On the Notebooks page, select the notebooks that you want to export. You have the option to export one or more or all the notebooks.
2. On the top panel of the notebook editor, click **Export** and then click any one of the following options:



- **Notebooks to Export:** To export notebooks, click:
  - **All:** To export all the notebooks listed on the Notebooks page.
  - **Selected:** To export the selected notebooks.
- **Format:** Click the format in which you want to export your notebook:
  - **Native:** Exports the notebook as a `.dsnb` (Data Studio Notebook) file.
  - **Jupyter:** Exports the notebook as a `.ipynb` file.
  - **Zeppelin:** Exports the notebook as a `.json` (JavaScript Object Notation) file.

The exported notebooks are saved either as `.dsnb` files, `.json` files or `.ipynb` files in a zipped folder.

Related Topics

- About Interpreters and Notebook Service Levels
  An interpreter is a plug-in that allows you to use a specific data processing language backend.

4.6 Import a Notebook

You can import notebooks into your Oracle Machine Learning UI projects. Oracle Machine Learning UI supports the import of notebooks in the native format (`.dsnb`), Zeppelin (`.json`) and Jupyter (`.ipynb`) notebooks. Oracle Machine Learning UI also supports importing notebooks from your GitHub repository.

Oracle Machine Learning UI supports the import of both Zeppelin (`.json`) and Jupyter (`.ipynb`) notebooks.

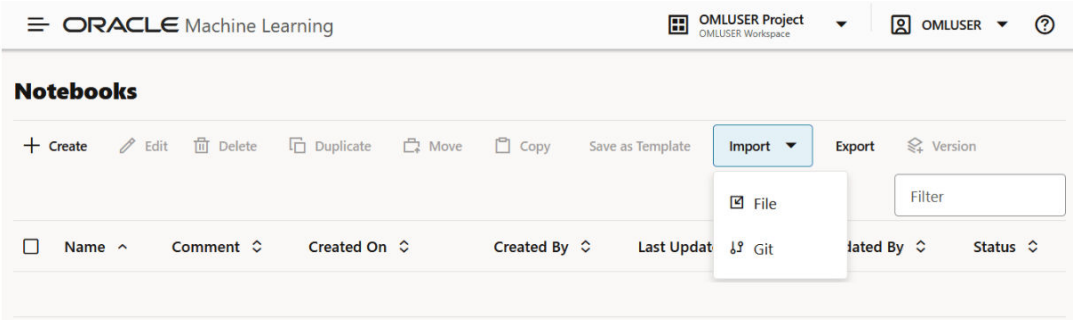
You can import a notebook on the Notebooks page. To import a notebook, go to the OML Notebooks page.

Import Notebook from File

To import a notebook from file:

1. On the Notebooks page, click **Import.** Here, you have two options—**File** and **Git**.

Figure 4-79 Notebook Import Options



2. Click **File**. This opens the **File Upload** dialog.
3. In the **File Upload** dialog, browse and select the notebook to import.

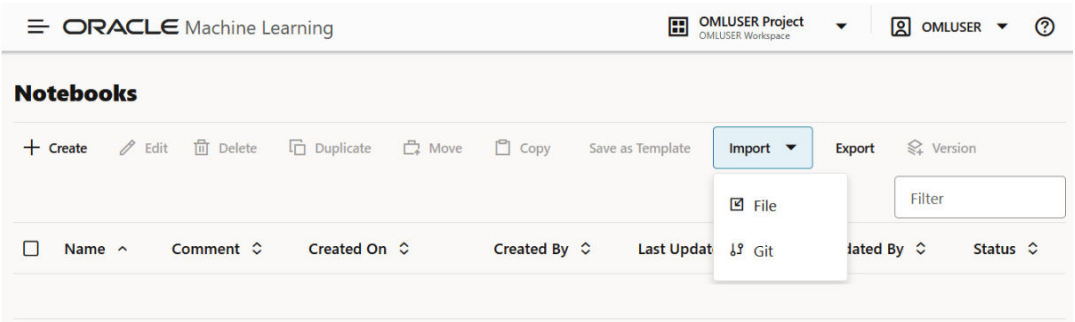
> **Note**
>
> You must have the notebook saved as a `.json` file or `.dsnb` file to import it. You can import notebooks exported from non-Oracle Apache Zeppelin environments, but only paragraphs types that are supported may be run.

4. Click **Open.**
   This completes the task of importing a notebook file into your project.

Import/Clone Notebooks from GitHub

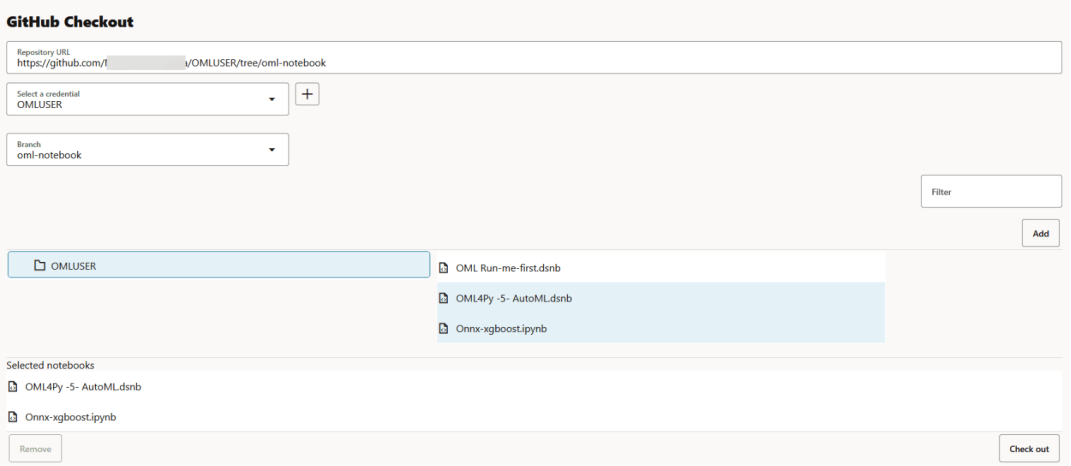
1. On the Notebooks page, click **Import.** Here, you have two options—**File** and **Git**.
2. Click **Git.** This opens the GitHub Checkout page.

Figure 4-80 Notebook Import Options



3. On the **GitHub Checkout** page, enter these details:

Figure 4-81 GitHub Checkout page



a. **Repository URL:** Enter the URL of the GitHub repository you want to access. You have the following options to provide the GitHub repository URL:

   - **Minimum valid URL:** You can provide the GitHub URL containing only the repository name and the owner. For example, `https://github.com/RepoOwner/RepoName`. This loads the branches found in the remote repository.
   - **Base URL and branch:** Enter the GitHub repository URL along with the branch you want to clone. For example, `https://github.com/RepoOwner/RepoName/tree/BranchName/` or `https://github.com/RepoOwner/RepoName/blob/BranchName/`. This automatically loads the **Branches** field. Select the Branch Name defined in the URL. and this will trigger the load of the directory structure found in the repository.
   - **Base URL, branch and directory:** You can also provide the GitHub URL containing one or several sub-directories in the remote repository. For example, `https://github.com/RepoOwner/RepoName/tree/Branch/Dir/Di2r` This loads the directory structure and pre-select the directory specified in the URL.
   - **Complete file path:** You can also enter the complete file path in the remote GitHub repository. For example, `https://github.com/RepoOwner/RepoName/blob/Branch/Dir/file.dsnb`. This loads the branches, directories and the file in the **Selected notebooks** field at the bottom of the dialog.

b. **Select a credential:** Click the down arrow and select a credential. If you do not have a credential created, click the + icon to create one. See Create GitHub Credentials for more information.

c. **Branch:** The drop-down menu displays the branches available in the remote GitHub repository based on the specified repository owner, repository name, and credential combination. Select a branch. The notebooks available in the branch are listed. You can also filter the notebooks you are looking for by typing in the notebook name in the **Filter** field.

d. Select the notebooks you want to clone and click **Add.** The notebooks you selected are now listed in the **Selected notebooks** section.

e. Click **Checkout.** This starts cloning all the GitHub notebooks you selected. Once completed, it displays the message "Notebooks successfully cloned". Click **Open Notebook listing** on the message box to go to the Notebooks listing page.

This completes the task of cloning and importing a notebook from your GitHub repository.

Related Topics

- About Interpreters and Notebook Service Levels
  An interpreter is a plug-in that allows you to use a specific data processing language backend.
- Database and Instance

4.7 About Interpreters and Notebook Service Levels

An interpreter is a plug-in that allows you to use a specific data processing language backend.

For the Zeppelin Notebooks in Oracle Machine Learning UI, you use the SQL, PL/SQL, Python and R interpreters within an Oracle AI Database interpreter group, and the Markdown interpreter for plain text formatting syntax so that it can be converted to HTML. You use the Conda interpreter to connect to the Conda environment and work with Python third-party library packages.

To use the interpreters, you must use these directives at the beginning of the paragraph in a notebook

- SQL—`%sql`
- PL/SQL—`%script`
- Python—`%python`
- R—`%r`
- Markdown—`%md`
- Conda—`%conda`

Notebooks contain an internal list of service levels that define the order of the interpreters in an interpreter group. The default order of interpreter bindings in the Oracle AI Database interpreter group is:

- **low:** Provides the least level of resources for in-database operations, typically serial (non-parallel) execution. It supports the maximum number of concurrent in-database operations by multiple users. The interpreter with low priority is listed at the top of the interpreter list, and hence, is the default.
- **medium:** Provides a fixed number of CPUs to run in-database operations in parallel, where possible. It supports a limited number of concurrent users, typically 1.25 times the number of CPUs allocated to the pluggable database.
- **high:** Provides the highest level of CPUs to run in-database operations in parallel, up to the number of CPUs allocated to the pluggable database. It offers the highest performance, but supports the minimum number of concurrent in-database operations, typically 3.
- **gpu:** Provides GPU compute capabilities in a notebook through the Python interpreter with the database service level set to **high**. The memory settings is 8 GB (DDR4), by default. It is extensible up to 200 GB.

With respect to interpreter bindings, you can perform the following tasks:

- Bind and unbind interpreters: If you do not bind any specific interpreter to your notebook, then you get the error message:

  ```
  Not supported interpreter <name of interpreter>
  ```

- Set and re-order interpreter bindings. You may want to set and re-order interpreter bindings if you want to use a specific interpreter for a specific paragraph in a notebook. In that case, you have to select the specific interpreter for that paragraph.
- Change the interpreter binding for any specific paragraph in a notebook

You must note the interpreter binding order in the following scenarios:

- Notebook creation: When you create a notebook, the notebook inherits the initial interpreter binding order, which is low (default), medium, high.
- Notebook import: When importing a notebook, the notebook inherits the defined interpreter bindings. However, after you import a notebook, ensure to check the order of the interpreter bindings and that the required interpreters are selected.
- Notebook export: When exporting a notebook, the notebook inherits the defined interpreter bindings.
- Notebook creation from templates: When you create a notebook from templates, the notebook inherits the default order of interpreter bindings.
- Change Notebook Service Level
  Notebook type corresponds to the ADB service levels — low, medium, high and gpu. These service levels affect parallelism in the database. The notebook type that is set for a notebook applies to all the paragraphs in that notebook.
- Verify Notebook Service Level
  After changing the Notebook service level, you can use a SQL statement to view and verify the information. Notebook type, referred to as interpreter bindings in Notebooks Classic, corresponds to the ADB service levels — low, medium, high and gpu. These service levels affect parallelism in the database.

4.7.1 Change Notebook Service Level

Notebook type corresponds to the ADB service levels — low, medium, high and gpu. These service levels affect parallelism in the database. The notebook type that is set for a notebook applies to all the paragraphs in that notebook.

> **Note**
>
> In Notebooks Classic, the notebook type is referred to as the interpreter bindings. The notebook type **gpu** is not available in Notebooks Classic.

To use the interpreters, you must use these directives at the beginning of a paragraph in a notebook

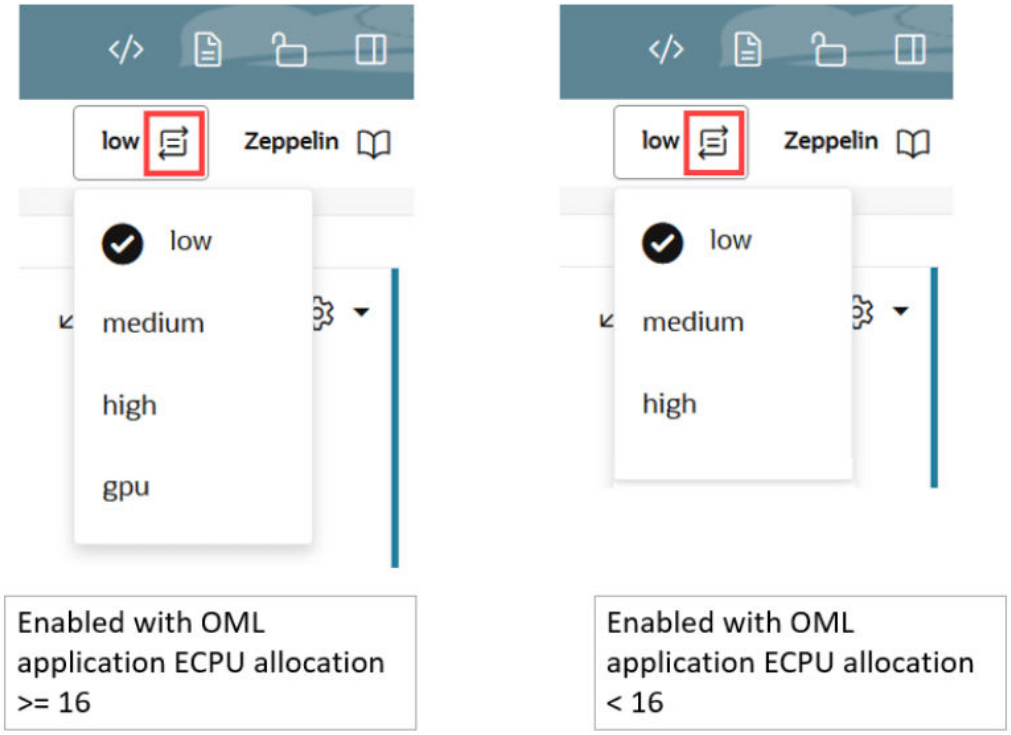
- SQL — `%sql`
- PL/SQL — `%script`
- Python — `%python`
- R — `%r`
- Markdown — `%md`
- Conda — `%conda`

You can change notebook service levels in both the new Notebook and Notebooks Classic.

Change Notebook Service Levels in the new Notebook

1. Open your notebook in the notebook editor.
2. Click on the **Update Notebook Type** icon on the top right corner. The available notebook types are displayed. The current notebook type is indicated by a tick mark, and is also displayed next to the **Update Notebook Type** icon.

Figure 4-82 ADB service levels as displayed in a notebook



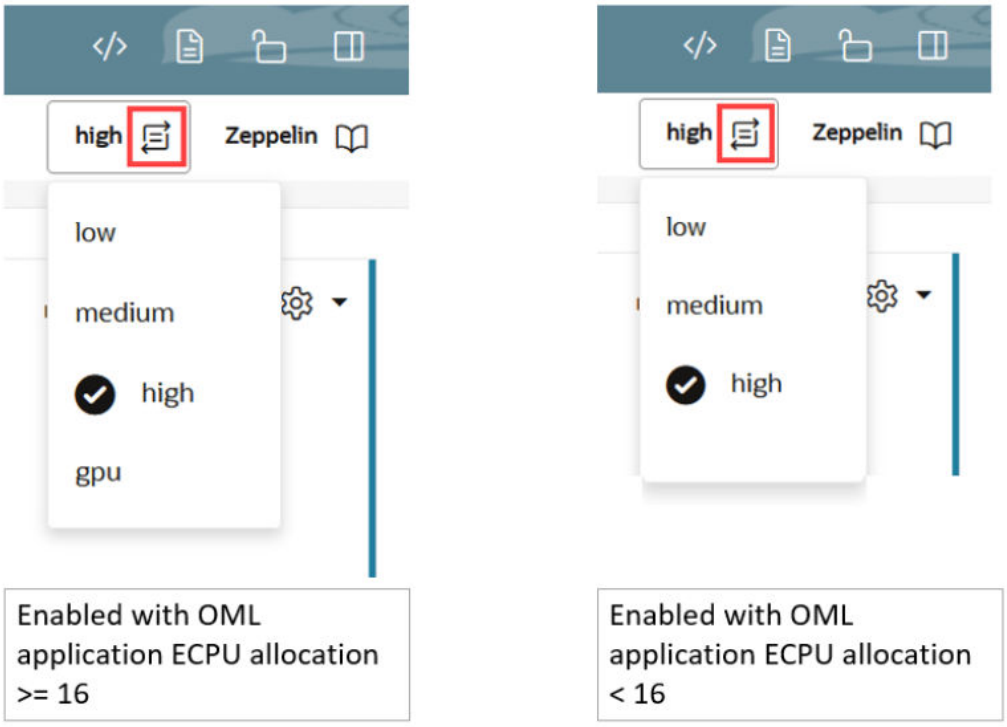
The Notebook Types (ADB service levels) are:

- **low** —Provides the least level of resources for in-database operations, typically serial (non-parallel) execution. It supports the maximum number of concurrent in-database operations by multiple users. The interpreter with low priority is listed at the top of the interpreter list, and hence, is the default.
- **medium**—Provides a fixed number of CPUs to run in-database operations in parallel, where possible. It supports a limited number of concurrent users, typically 1.25 times the number of CPUs allocated to the pluggable database.
- **high**—Provides the highest level of CPUs to run in-database operations in parallel, up to the number of CPUs allocated to the pluggable database. It offers the highest performance, but supports the minimum number of concurrent in-database operations, typically 3.
- **gpu**—Provides GPU compute capabilities in a notebook through the Python interpreter with the database service level set to **high**. The notebook memory setting is 32 GB (DDR4), by default. It is extensible up to 200 GB.

3. To change the notebook type, click on the type that you want to select. In this example, let's click **high.** A confirmation message is displayed stating: `Notebook Type is updated to "high".`

> **Note**
>
> The updated notebook type is applicable to all the paragraphs in the notebook. You cannot change the notebook type at the paragraph level.

Figure 4-83 ADB service level - high



4.7.2 Verify Notebook Service Level

After changing the Notebook service level, you can use a SQL statement to view and verify the information. Notebook type, referred to as interpreter bindings in Notebooks Classic, corresponds to the ADB service levels — low, medium, high and gpu. These service levels affect parallelism in the database.

For Python notebooks, the interpreter binding is used for all python paragraphs.

> **Note**
>
> For Python notebooks, do not override the interpreter binding at the paragraph level.

Verify Notebook Service Level in a Notebook

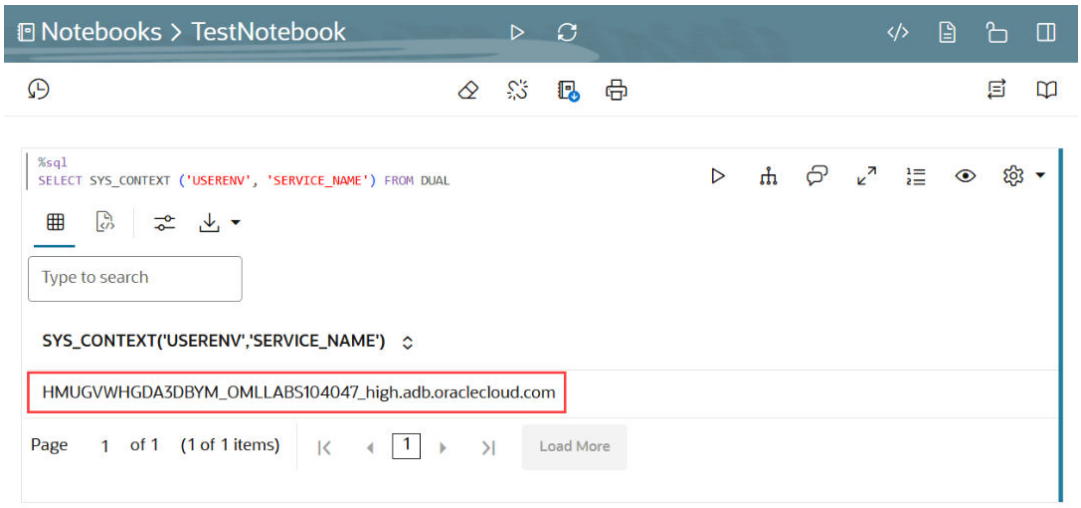
1. Open the notebook for which you want to verify the updated notebook type.

2. Run the following SQL statement:

```
%sql
SELECT SYS_CONTEXT ('USERENV', 'SERVICE_NAME') FROM DUAL;
```

3. Click **Run.** The SQL statement returns the following information about the updated notebook type, as shown in the screenshot below:

Figure 4-84 ADB service level information as displayed in a notebook



You can see that the service level is now displayed as **high**. In this example:

- `HMUGVWHGDA3DBYM` is the tenant name
- `OMLLABS104047` is the database name
- `high` is the ADB service name
- `adb.oraclecloud.com` is the domain

4.8 Run Notebooks using different interpreters

An interpreter is a plug-in that allows you to use a specific data processing language backend. You select an interpreter by inserting its directive at the beginning of a notebook paragraph.

Topics:

- Use the SQL Interpreter in a Notebook Paragraph
  An Oracle Machine Learning notebook supports multiple languages. Each paragraph is associated with a specific interpreter. For example, to run SQL statements use the SQL interpreter. To run PL/SQL statements, use the `script` interpreter.
- Use the Python Interpreter in a Notebook Paragraph
  An Oracle Machine Learning notebook supports multiple languages. Each paragraph is associated with a specific interpreter. To run Python commands in a notebook, you must first connect to the Python interpreter. To use OML4Py, you must import the `oml` module.
- Use the R Interpreter in a Notebook Paragraph
  An Oracle Machine Learning notebook supports multiple languages. Each paragraph is associated with a specific interpreter. To run R functions in an Oracle Machine Learning notebook, you must first connect to the R interpreter.

- Use the Conda Interpreter in a Notebook Paragraph
  Oracle Machine Learning Notebooks provides a Conda interpreter to enable administrators to create conda environments with custom third-party Python and R libraries. Once created, you can download and activate Conda environments inside a notebook session also using the Conda interpreter.
- Use the Markdown Interpreter and Generate Static html from Markdown Plain Text
  Use the Markdown interpreter and generate static html from Markdown plain text.

4.8.1 Use the SQL Interpreter in a Notebook Paragraph

An Oracle Machine Learning notebook supports multiple languages. Each paragraph is associated with a specific interpreter. For example, to run SQL statements use the SQL interpreter. To run PL/SQL statements, use the `script` interpreter.

In an Oracle Machine Learning notebook, you can add multiple paragraphs, and each paragraph can be connected to different interpreters such as SQL or Python. You identify which interpreter to use by specifying `%` followed by the interpreter to use: `sql`, `script`, `r`, `python`, `conda`, `markdown`.

A paragraph has an input section and an output section. In the input section, specify the interpreter to run along with the text. This information is sent to the interpreter to be run. In the output section, the results of the interpreter are provided.
You can use the following directives in a notebook paragraph:

- `%sql`: Supports standard SQL statements. In `%sql` the results of a `SELECT` statement are directly displayed in a table viewer, with access to other visualization options. Use the options in the chart settings to perform groupings, summation, and other operations.
- `%script`: Supports both SQL statements and PL/SQL. In `%script`, the results of a `SELECT` statement are provided as text string output.
- `%conda`: Supports the Conda environment. Type `%conda` at the beginning of the paragraph to connect to the Conda environment and work with third-party libraries for Python.
- `%r`: Supports R scripts. Type `%r` at the beginning of the paragraph to connect to the R interpreter.
- `%python`: Supports Python scripts. Type `%python` at the beginning of the paragraph to connect to the Python interpreter.
- `%md`: Supports Markdown markup language.

> **Note**
>
> To run a **Group By** on all your data, then it is recommended to use SQL scripts to do the grouping in the database, and return the summary information for charting in the notebook. Grouping at the notebook level works well for small sets of data. If you pull too much data to the notebook, you may encounter issues due to insufficient memory. You can set the row limit for your notebook by using the option **Render Row Limit** on the Connections Group page.

To fetch and visualize data in a notebook:

1. On the Notebook page, click the notebook that you want to run.

   The notebook opens in edit mode.

2. Type `%SQL` to call the SQL interpreter, and press Enter. Your notebook is now ready to run SQL statements.
3. Type the SQL statement to fetch data from an Oracle AI Database. For example, type `SELECT * FROM TABLENAME` and click ▷. Alternatively, press Shift+Enter keys to run the notebook.

> **Note**
>
> Notebooks must be opened as a regular user, that is, a non-administrator user. The **Run** notebook option is not available to the Administrator.

This fetches the data in the notebook.

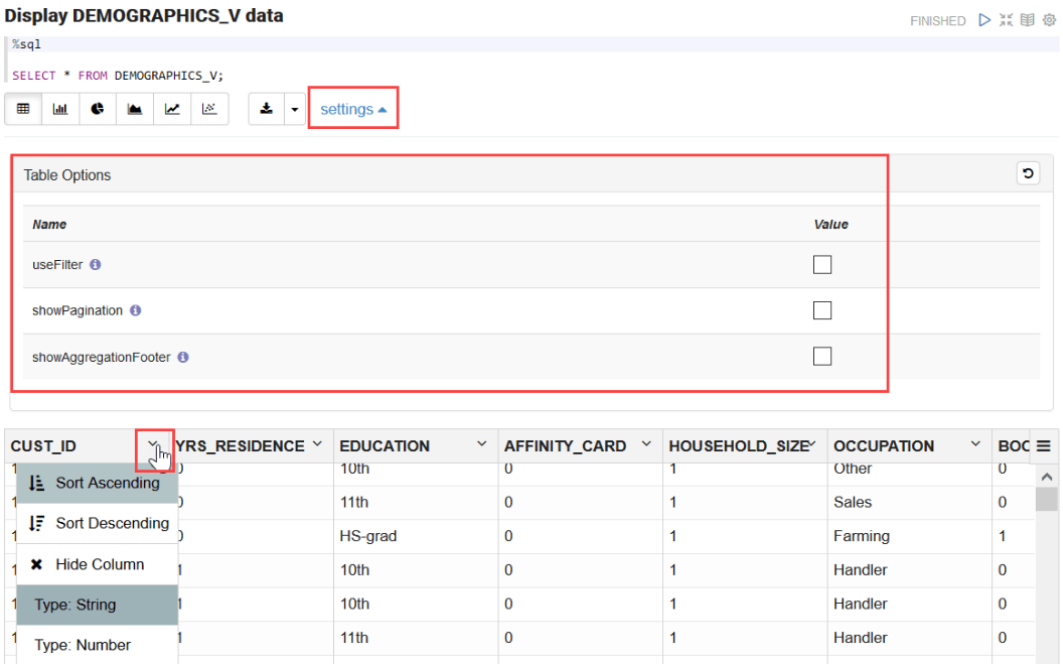
4. The data is displayed in the output of the paragraph.

The results of the interpreter appear in the output section. The output section of the paragraph comprises a charting component that displays the results in graphical output. The chart interface allows you to interact with the output in the notebook paragraph. You have the option to run and edit single a paragraph or all paragraphs in a notebook.

For Table Options, click **settings** and select:

- **useFilter:** To enable filter for columns.
- **showPagination:** To enable pagination for enhanced navigation.
- **showAggregationFooter:** To enable a footer to display aggregated values.

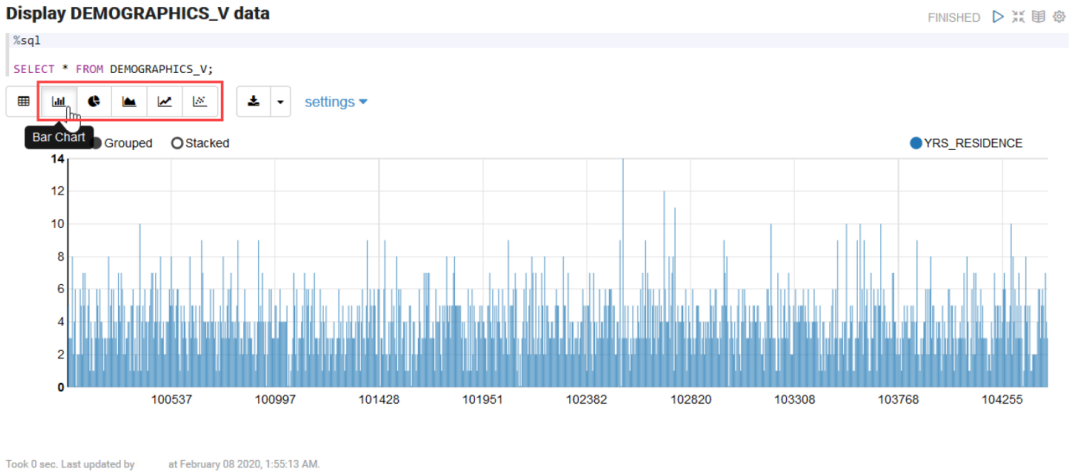
You can also sort the columns by clicking the down arrow next to the column name.



To visualize the tabular data, click the respective icons for each of the each graphical representation, as shown here:

- Click to represent the data in a Bar Chart.

- Click to represent the data in a Pie Chart.
- Click to represent the data in an Area Chart.
- Click to represent the data in a Line Chart.
- Click to represent the data in a Scatter Chart.



- About Oracle Machine Learning for SQL
  Oracle Machine Learning for SQL (OML4SQL) provides a powerful, state-of-the-art machine learning capability within Oracle AI Database. You can use Oracle Machine Learning for SQL to build and deploy predictive and descriptive machine learning models, add intelligent capabilities to existing and new applications.
- Set Output Format in Notebooks
  Oracle Machine Learning Notebooks allow you to preformat query output in notebooks.
- Output Formats Supported by SET SQLFORMAT Command
  By using the `SET SQLFORMAT` command, you can generate the query output in a variety for formats.

Related Topics

- Edit Oracle AI Database Interpreter Connection
  When defining an Oracle AI Database interpreter connection, a reference to a compute resource is created. This reference contains all connection-related information about the interpreter.

4.8.1.1 About Oracle Machine Learning for SQL

Oracle Machine Learning for SQL (OML4SQL) provides a powerful, state-of-the-art machine learning capability within Oracle AI Database. You can use Oracle Machine Learning for SQL to build and deploy predictive and descriptive machine learning models, add intelligent capabilities to existing and new applications.

Oracle Machine Learning for SQL offers an extensive set of in-database algorithms for performing a variety of machine learning tasks, such as classification, regression, anomaly detection, feature extraction, clustering, and market basket analysis, among others. The programmatic interfaces to OML4SQL are PL/SQL for building and maintaining models and a family of SQL functions for scoring.

Use Oracle Machine Learning Notebooks with the SQL and PL/SQL interpreters to run SQL statements (`%sql`) and PL/SQL scripts (`%script`) respectively. Use Oracle Machine Learning for SQL to:

- Perform data exploration and data analysis
- Build, evaluate and deploy machine learning models, and
- Score data using those models

4.8.1.2 Set Output Format in Notebooks

Oracle Machine Learning Notebooks allow you to preformat query output in notebooks.

To preformat query output, you must use the command `SET SQLFORMAT` as follows:

1. Open a notebook in Oracle Machine Learning.
2. Type the command:

   ```
   %script

   SET SQLFORMAT format_option
   ```

   For example, if you want the output in ansiconsole format, then type the command followed by the SQL query as:

   ```
   SET SQLFORMAT ansiconsole;

   SELECT * FROM HR.EMPLOYEES;
   ```

   Here, the output format is `ansiconsole`, and the table name is `HR.EMPLOYEES`.

> **Note**
>
> This formatting is available for the Script interpreter. Therefore, you must add the prefix `%script` as shown in this example.

4.8.1.3 Output Formats Supported by SET SQLFORMAT Command

By using the `SET SQLFORMAT` command, you can generate the query output in a variety for formats.

> **Note**
>
> These output formats are available for the Script interpreter. Therefore, you must include the prefix `%script`.

The available output formats are:

- CSV: The CSV format produces standard comma-separated variable output, with string values enclosed in double quotes. The syntax is:

  ```
  %script

  SET SQLFORMAT CSV
  ```

- HTML: The HTML format produces the HTML for a responsive table. The content of the table changes dynamically to match the search string entered in the text field. The syntax is:

```
%script

SET SQLFORMAT HTML
```

- XML: The XML format produces a tag based XML document. All data is presented as `CDATA` tags. The syntax is:

  ```
  %script

  SET SQLFORMAT XML
  ```

- JSON: The JSON format produces a JSON document containing the definitions of the columns along with the data that it contains. The syntax is:

  ```
  %script

  SET SQLFORMAT JSON
  ```

- ANSICONSOLE: The ANSICONSOLE format resizes the columns to the width of the data to save space. It also underlines the columns, instead of separate line of output. The syntax is:

  ```
  %script

  SET SQLFORMAT ANSICONSOLE
  ```

- INSERT: The INSERT format produces the INSERT statements that could be used to recreate the rows in a table. The syntax is:

  ```
  %script

  SET SQLFORMAT INSERT
  ```

- LOADER: The LOADER format produces pipe delimited output with string values enclosed in double quotes. The column names are not included in the output. The syntax is:

  ```
  %script

  SET SQLFORMAT LOADER
  ```

- FIXED: The FIXED format produces fixed width columns with all data enclosed in double-quotes. The syntax is:

  ```
  %script

  SET SQLFORMAT FIXED
  ```

- DEFAULT: The DEFAULT option clears all previous `SQLFORMAT` settings, and returns to the default output. The syntax is:

  ```
  %script

  SET SQLFORMAT DEFAULT
  ```

  > **Note**
  >
  > You can also run this command without the format name `DEFAULT` by simply typing `SET SQLFORMAT`.

- DELIMITED: The DELIMITED format allows you to manually define the delimiter string, and the characters that are enclosed in the string values. The syntax is:

  ```
  %script

  SQLFORMAT DELIMITED delimiter left_enclosure right_enclosure
  ```

  For example,

```
%script

SET SQLFORMAT DELIMITED ~del~ " "

SELECT * FROM emp WHERE deptno = 20;
```

Output:

```
"EMPNO"~del~"ENAME"~del~"JOB"~del~"MGR"~del~"HIREDATE"~del~"SAL"~del~"COMM"~del~"DEPTNO"
```

In this example, the delimiter string is `~del~` and string values such as `EMPNO, ENAME, JOB` and so on, are enclosed in double quotes.

4.8.2 Use the Python Interpreter in a Notebook Paragraph

An Oracle Machine Learning notebook supports multiple languages. Each paragraph is associated with a specific interpreter. To run Python commands in a notebook, you must first connect to the Python interpreter. To use OML4Py, you must import the `oml` module.

In an Oracle Machine Learning notebook, you can add multiple paragraphs, and each paragraph can be connected to different interpreters such as SQL or Python. You identify which interpreter to use by specifying `%` followed by the interpreter to use: `sql`, `script`, `r`, `python`, `conda`, `markdown`.

This example shows you how to:

- Connect to a Python interpreter to run Python commands in a notebook
- Import the Python modules - `oml`, `matplotlib`, and `numpy`
- Check if the `oml` module is connected to the Oracle AI Database

> **Note**
>
> `z` is a reserved keyword and must not be used as a variable in `%python` paragraphs in Oracle Machine Learning notebooks.

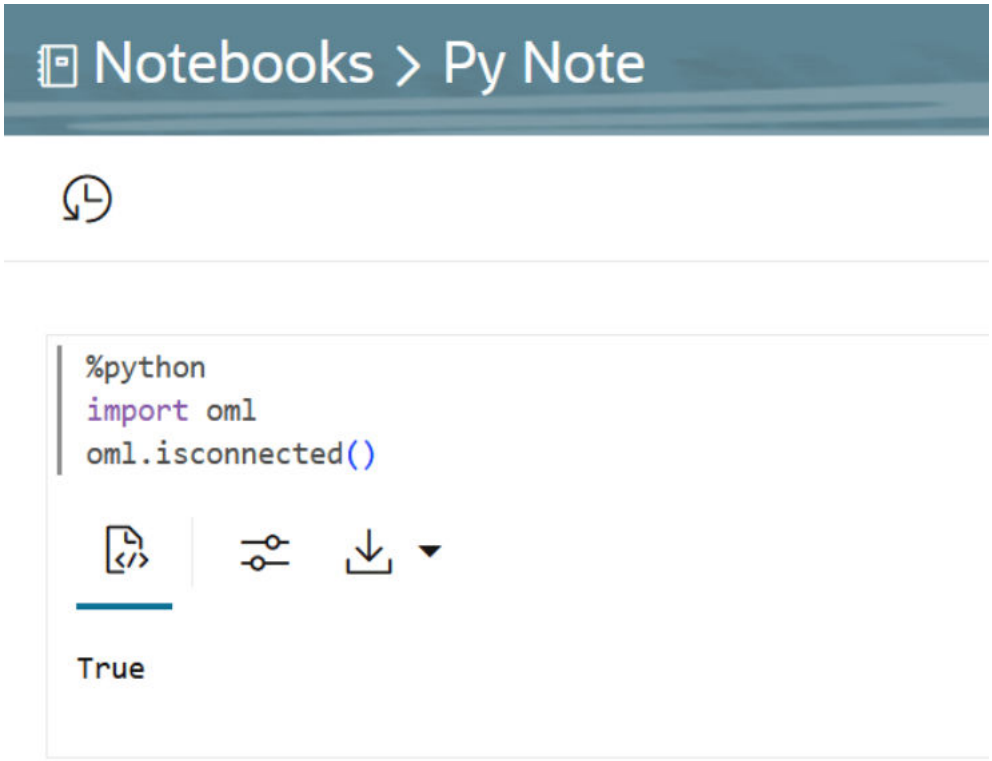
Assumption: The example assumes that you have created a new notebook named **Py Note.**

1. Open the Py Note notebook and click the interpreter bindings icon. View the available interpreter bindings.
2. To connect to the Python interpreter, type `%python`

   You are now ready to run Python scripts in your notebook.
3. To use OML4Py module, you must import the `oml` module. Type the following Python command to import the `oml` module, and the click run icon. Alternatively, you can press Shift+Enter keys to run the notebook.

   ```
   import oml
   ```
4. To verify if the `oml` module is connected to the Database, type:

   ```
   oml.isconnected()
   ```



Once your notebook is connected, the command returns `TRUE`. The notebook is now connected to the Python interpreter, and you are ready to run python commands in your notebook.

Example to demonstrate the use of the Python modules - `matplotlib` and `numpy` , and use random data to plot two histograms.

5. Type the following commands to import the modules:

```
%python
import matplotlib.pyplot as plt
import numpy as np
```

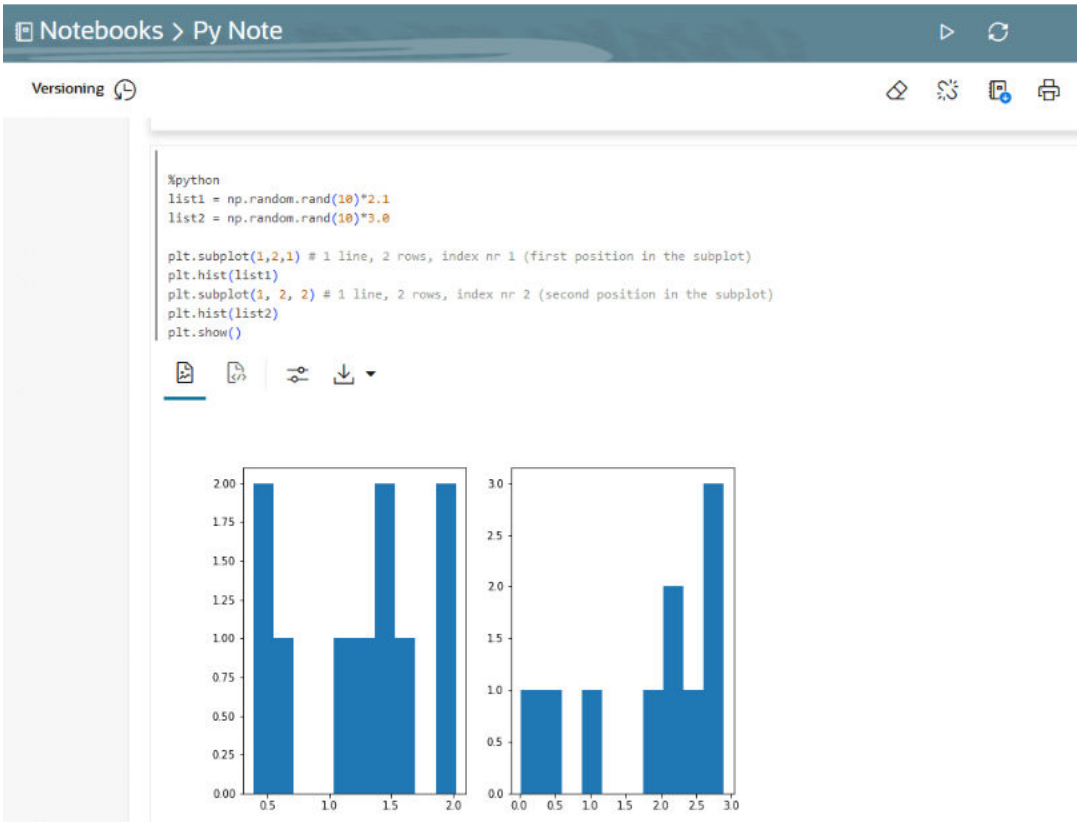
   - `Matplotlib` - Python module to render graphs
   - `Numpy` - Python module for computations

6. Type the following commands to compute and render the data in two histograms.

```
list1 = np.random.rand(10)*2.1
list2 = np.random.rand(10)*3.0

plt.subplot(1,2,1) # 1 line, 2 rows, index nr 1 (first position in the
subplot)
plt.hist(list1)
plt.subplot(1, 2, 2) # 1 line, 2 rows, index nr 2 (second position in the
subplot)
plt.hist(list2)
plt.show()
```

In this example, the commands import two Python module to compute and render the data in two histograms list1 and list2.

7. Click **Run.**



The output section of the paragraph which contains a charting component displays the results in two histograms - list1 and list2, as shown in the screenshot.

- About Oracle Machine Learning for Python
  Oracle Machine Learning for Python (OML4Py) is a component of Oracle Autonomous AI Database, which includes Oracle Autonomous AI Lakehouse (ADW), Oracle Autonomous AI Transaction Processing (ATP), and Oracle Autonomous AI JSON Database (AJD). By using Oracle Machine Learning UI notebooks, you can run Python functions on data for data exploration and preparation while leveraging Oracle AI Database as a high-performance computing environment. Oracle Machine Learning User Interface (UI) is available through Autonomous AI Lakehouse (ADW) , Autonomous AI Transaction Processing (ATP) and Autonomous AI JSON Database (AJD) services.

4.8.2.1 About Oracle Machine Learning for Python

Oracle Machine Learning for Python (OML4Py) is a component of Oracle Autonomous AI Database, which includes Oracle Autonomous AI Lakehouse (ADW), Oracle Autonomous AI Transaction Processing (ATP), and Oracle Autonomous AI JSON Database (AJD). By using Oracle Machine Learning UI notebooks, you can run Python functions on data for data exploration and preparation while leveraging Oracle AI Database as a high-performance computing environment. Oracle Machine Learning User Interface (UI) is available through Autonomous AI Lakehouse (ADW) , Autonomous AI Transaction Processing (ATP) and Autonomous AI JSON Database (AJD) services.

Oracle Machine Learning for Python (OML4Py) makes the open source Python scripting language and environment ready for the enterprise and big data. Designed for problems involving both large and small volumes of data, Oracle Machine Learning for Python integrates Python with Oracle Autonomous AI Database, including its powerful in-database machine learning algorithms, and enables deployment of Python code.

Use Oracle Machine Learning for Python to:

- Perform data exploration, data analysis, and machine learning using Python leveraging Oracle AI Database as a high performance compute engine
- Build and evaluate machine learning models and score data using those models from an integrated Python API using in-database algorithms
- Deploy user-defined Python functions through a REST interface with data-parallel and task-parallel processing

The Python interpreter uses Python 3.13.5 to process Python scripts in Oracle Machine Learning notebooks. To use the interpreter, specify the `%python` directive at the beginning of the paragraph. The Python interpreter supports the following Python modules:

- `cffi-1.15.1`
- `contourpy-1.1.0`
- `cryptography-43.0.1`
- `cycler-0.10.0`
- `fonttools-4.49.0`
- `joblib-1.4.2`
- `kiwisolver-1.4.5`
- `matplotlib-3.8.4`
- `numpy-2.0.1`
- `oracledb-2.4.1`
- `packaging-23.1`
- `pandas-2.2.2`
- `pillow-10.4.0`
- `pycparser-2.21`
- `pyparsing-2.4.0`
- `python_dateutil-2.8.2`
- `pytz-2022.1`
- `scikit_learn-1.5.1`
- `scipy-1.14.0`
- `setuptools-70.0.0`
- `six-1.16.0`
- `threadpoolctl-3.5.0`

Related Topics

- About Oracle Machine Learning for Python

4.8.3 Use the R Interpreter in a Notebook Paragraph

An Oracle Machine Learning notebook supports multiple languages. Each paragraph is associated with a specific interpreter. To run R functions in an Oracle Machine Learning notebook, you must first connect to the R interpreter.

In an Oracle Machine Learning notebook, you can add multiple paragraphs, and each paragraph can be connected to different interpreters such as R or SQL or Python. You identify which interpreter to use by specifying `%` followed by the interpreter to use: `sql`, `script`, `r`, `python`, `conda`, `markdown`.

This example shows how to:

- Connect to the R interpreter to run R commands in a notebook.
- Verify the connection to Oracle Autonomous AI Database, and
- Load the ORE libraries

1. To connect to the R interpreter, type the following directive at the beginning of the notebook paragraph, and press Enter:

   ```
   %r
   ```

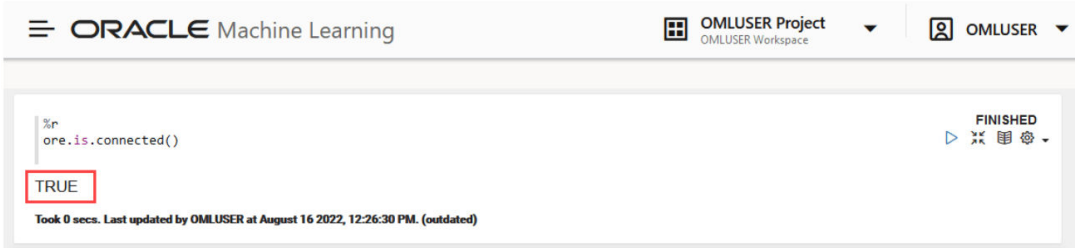
   To know more about interpreters, see About Interpreters and Notebook Service Levels and Change Notebook Service Level.

2. To verify the database connection, type the following command and press Enter:

   ```
   ore.is.connected()
   ```

   Once your notebook is connected, the command returns `TRUE`, as shown in the screenshot here. The notebook is now connected to the R interpreter, and you are ready to run R commands in your notebook.

Figure 4-85 Command to Test Database Connection

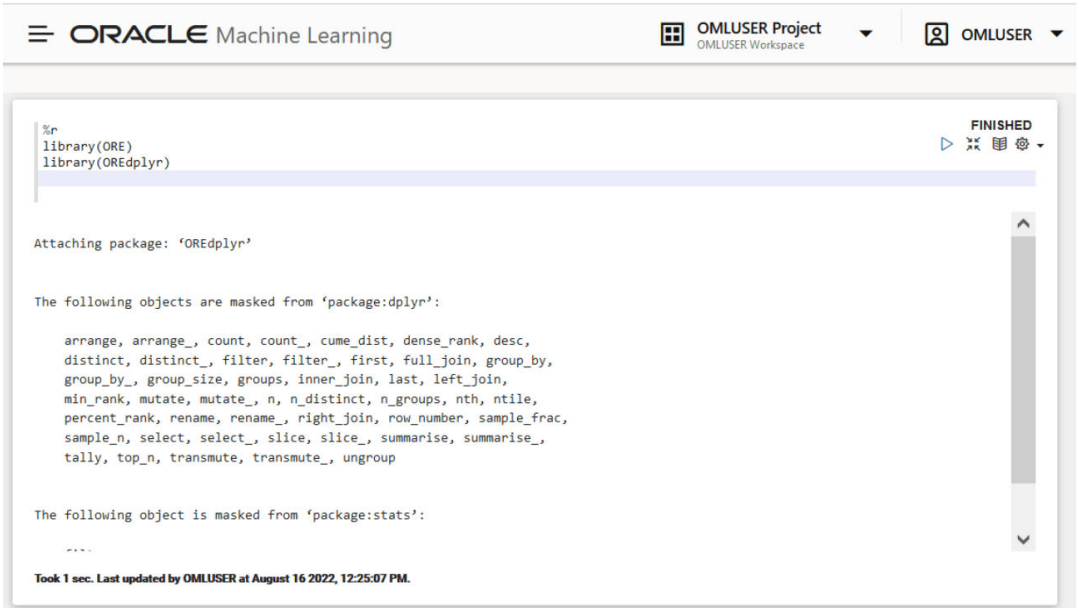


3. To import R Libraries, run the following commands:

   ```
   library(ORE)
   library(OREdplyr)
   ```

   Once the packages are loaded successfully, the list of ORE packages are displayed as shown in the screenshot here. Scroll down to view the entire list.

Figure 4-86 Command to Import R packages



- About Oracle Machine Learning for R
  Oracle Machine Learning for R (OML4R) is a component of the Oracle Machine Learning family of products, which integrates R with Oracle Autonomous AI Database.

4.8.3.1 About Oracle Machine Learning for R

Oracle Machine Learning for R (OML4R) is a component of the Oracle Machine Learning family of products, which integrates R with Oracle Autonomous AI Database.

Oracle Machine Learning for R makes the open source R scripting language and environment ready for enterprise and big data. It is designed for problems involving for both large and small data volumes. OML4R allows users to run R commands and scripts for statistical, machine learning, and perform visualization analyses on database tables and views using R syntax.

Oracle Machine Learning for R is available in Oracle Machine Learning UI, currently available through Oracle Autonomous AI Database, including Autonomous AI Lakehouse, Autonomous AI Transaction Processing, and Autonomous AI JSON Database. Oracle Machine Learning for R Embedded R Execution functionality can be deployed through SQL and REST APIs on Autonomous AI Database.

Use Oracle Machine Learning for R to:

- Perform data exploration and data preparation while seamlessly leveraging as a high-performance computing environment.
- Run user-defined R functions on database spawned and controlled R engines, with system-supported data-parallel and task-parallel capabilities.
- Access and use powerful in-database machine learning algorithms from R language.

To use the R interpreter, specify the `%r` directive at the beginning of the paragraph. The following R packages are installed to support Oracle Machine Learning for R.

Supported Oracle Machine Learning for R Proprietary R Packages

The supported Oracle Machine Learning for R proprietary R packages are:

- ORE_1.5.1
- OREbase_1.5.1
- OREcommon_1.5.1
- OREdm_1.5.1
- OREdplyr_1.5.1
- OREeda_1.5.1
- OREembed_1.5.1
- OREgraphics_1.5.1
- OREmodels_1.5.1
- OREpredict_1.5.1
- OREstats_1.5.1
- ORExml_1.5.1

Supported Open Source R Modules

The following open source R packages are supported by Oracle Machine Learning for R:

- R-4.0.5
- Cairo_1.5-15
- ROracle_1.4-1: DBI_1.1-2
- arules_1.7-3
- png_0.1-7
- randomForest_4.6-14
- statmod_1.4-36
- dplyr_1.0-9:
- R6_2.5.1
- assertthat_0.2.1
- cli_3.3.0
- crayon_1.5.1
- ellipsis_0.3.2
- fansi_1.0.3
- generics_0.1.2
- glue_1.6.2
- lazyeval_0.2.2
- lifecycle_1.0.1
- magrittr_2.0.3
- pillar_1.7.0
- pkgconfig_2.0.3
- purrr_0.3.4

- rlang_1.0.2
- tibble_3.1.7
- tidyselect_1.1.2
- utf8_1.2.2
- vctrs_0.4.1

Oracle Machine Learning for R Interpreter Requirements

The R interpreter requires the following open source R packages:

- Rkernel 1.3:
  - base64enc 0.1-3
  - cli 3.3.0
  - crayon 1.5.1
  - digest 0.6.29
  - ellipsis 0.3.2
  - evaluate 0.15
  - fansi 1.0.3
  - fastmap 1.1.0
  - glue 1.6.2
  - htmltools 0.5.2
  - IRdisplay 1.1
  - jsonlite 1.8.0
  - lifecycle 1.0.1
  - pbdZMQ 0.3-7
  - pillar 1.7.0
  - repr 1.1.4
  - rlang 1.0.2
  - utf8 1.2.2
  - uuid 1.1-0
  - vctrs 0.4.1
- knitr 1.39:
  - evaluate_0.15
  - glue_1.6.2
  - highr_0.9
  - magrittr_2.0.3
  - stringi_1.7.6
  - stringr_1.4.0
  - xfun_0.31

- `yaml_2.3.5`

4.8.4 Use the Conda Interpreter in a Notebook Paragraph

Oracle Machine Learning Notebooks provides a Conda interpreter to enable administrators to create conda environments with custom third-party Python and R libraries. Once created, you can download and activate Conda environments inside a notebook session also using the Conda interpreter.

An Oracle Machine Learning notebook supports multiple languages. For this, you must create a notebook with some paragraphs to run SQL queries, and other paragraphs to run PL/SQL scripts. To run a notebook in different scripting languages, you must first connect the notebook paragraphs with the respective interpreters such as SQL, PL/SQL, R, Python, or Conda.

This topic shows how to start working in the Conda environment:

- Connect to the Conda interpreter
- Download and activate the Conda environment
- View the list of packages in the Conda environment
- Run a Python function to import the Iris dataset, and use the seaborn package for visualization

1. Type `%conda` at the beginning of the paragraph to connect to the Conda interpreter, and press Enter.

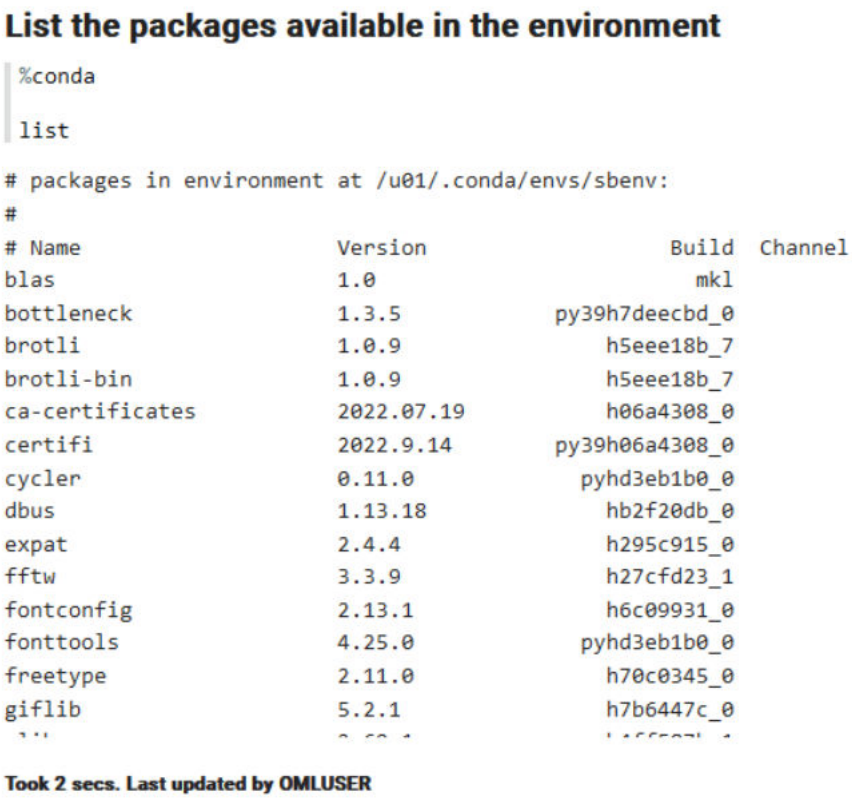
   ```
   %conda
   ```

2. Next, download and activate the Conda environment. Type:

   ```
   download sbenv
   activate sbenv
   ```

In this example, the Conda environment is downloaded and activated. The name of the Conda environment in this example is `sbenv`.

3. You can view all the packages that are present in the Conda environment. To view the list of packages, type `list`.



4. Here's an example that demonstrates how to use the seaborn library package for visualization. Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. This example

   - Imports Pandas and seaborn
   - Loads the Iris dataset
   - Plots the datapoints, that is, the three different species of the Iris flower - Setosa, Virginica, and Versicolor based on its dimensions. It creates a scatter plot

   Type:

```
%python

def sb_plot():
    import pandas as pd
    import seaborn as sb
    from matplotlib import pyplot as plt
    df = sb.load_dataset('iris')
    sb.set_style("ticks")
    sb.pairplot(df,hue = 'species',diag_kind = "kde",kind =
"scatter",palette = "husl")
    plt.show()
```

Create a visualization using seaborn

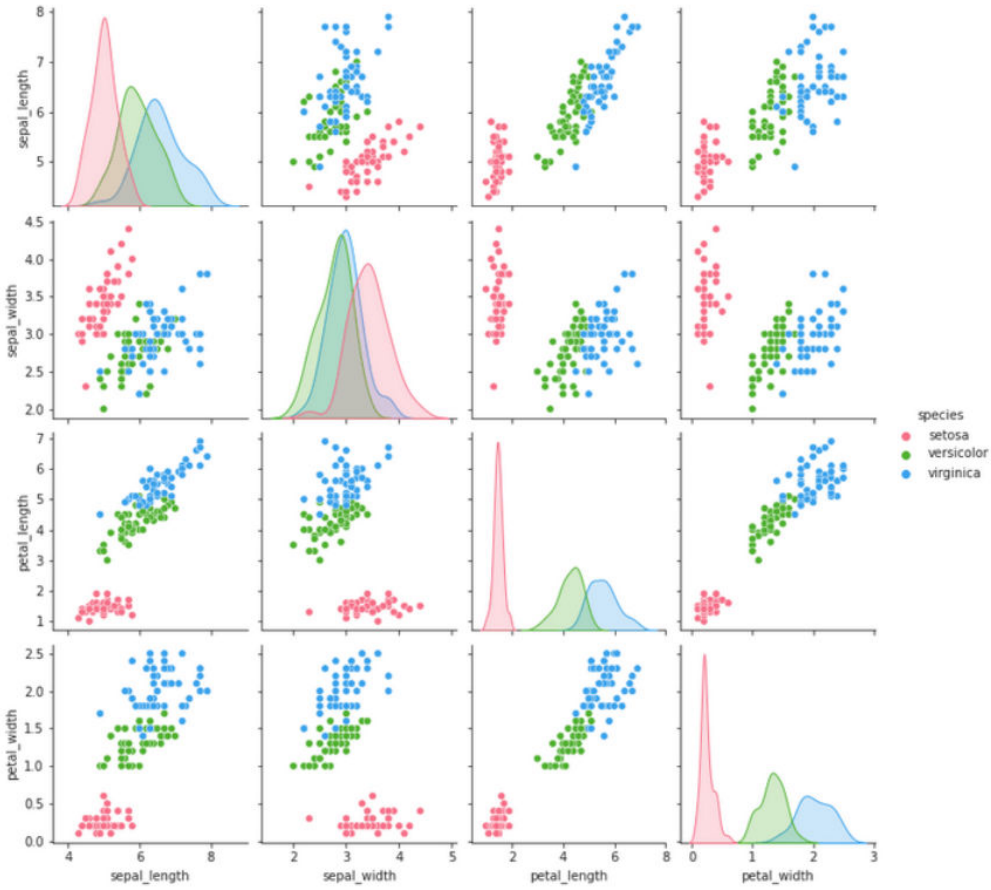
```
%python

def sb_plot():
    import pandas as pd
    import seaborn as sb
    from matplotlib import pyplot as plt
    df = sb.load_dataset('iris')
    sb.set_style("ticks")
    sb.pairplot(df,hue = 'species',diag_kind = "kde",kind = "scatter",palette = "husl")
    plt.show()
```

Took 0 secs. Last updated by OMLUSER

5. Run the function in a Python paragraph.

   Type:

   ```
   %python
   sb_plot()
   ```



Took 6 secs. Last updated by OMLUSER

- About the Conda Environment and Conda Interpreter
  Conda is an open-source package and environment management system that allows the use of environments containing third-party Python and R libraries. Oracle Machine Learning User Interface (UI) provide the conda interpreter to install third-party Python and R libraries inside a notebook session.

- Conda Interpreter Commands
  This table lists the commands for the Conda interpreter.
- Create a Conda Environment for Python, Install a Python Package and Upload it to Object Storage
  This topic demonstrates how to create a Conda environment `mypyenv` for Python packages, with Python 3.13.5 that is compatible with OML4Py. Here, you will also install the Python package seaborn from the `conda-forge` channel, and upload the Conda environment `mypyenv` to object storage.
- Create a Conda Environment for R and Install R Package and Upload it to Object Storage
  This topic shows how to create a Conda environment `myrenv` for R packages, with R-4.0.5 for OML4R compatibility, install the `forecast` package, and upload the Conda environment to object storage.
- Upload the Conda environments to an Object Storage bucket associated with the Autonomous AI Database
  There is one Object Storage bucket for each data center region. The Conda environment is saved to a folder in Object Storage corresponding to the tenancy and database. The folder is managed by Autonomous AI Database and is only available to users through Oracle Machine Learning Notebooks. There is an 8G maximum size for a single Conda environment, and no size limit on Object Storage.

4.8.4.1 About the Conda Environment and Conda Interpreter

Conda is an open-source package and environment management system that allows the use of environments containing third-party Python and R libraries. Oracle Machine Learning User Interface (UI) provide the conda interpreter to install third-party Python and R libraries inside a notebook session.

Third-party libraries installed in Oracle Machine Learning Notebooks can be used in:

- Standard Python
- Standard R
- Oracle Machine Learning for Python embedded Python execution from the Python, SQL and REST APIs
- Oracle Machine Learning for R embedded R execution from the R, SQL, and REST APIs

To start working in the Conda environment:

1. Ensure that a Conda environment is saved to Object Storage, or update an existing package by installing a new version.

> **Note**
>
> You must be signed in as ADMIN user to create the Conda environment. The administrator manages the lifecycle of an environment, including adding or deleting packages from it, and removing environments. The Conda environments are stored in an Object Storage bucket associated with the Autonomous AI Database.

2. Sign into Oracle Machine Learning UI and download the Conda environment. To download the Conda environment, type:

```
%conda


download myenv
```

3. Activate the Conda environment. To activate the Conda environment, type:

```
activate myenv
```

> **Note**
>
> There is only one active Conda environment at a given point in time.

4. Create a notebook, use the Conda interpreter to use third-party libraries in the Object Storage. To use the Conda interpreter, type `%conda` at the beginning of the paragraph to connect to the Conda environment, and work with third-party libraries for Python. You can switch between the preinstalled Conda environments. For example, you can have an environment for working with Graph analysis, and another environment for Oracle Machine Learning analysis.
5. Deactivate the Conda environment. As a best practice, deactivate the Conda environment after you have finished working on your machine learning analysis. To deactivate the environment, type:

```
deactivate
```

4.8.4.2 Conda Interpreter Commands

This table lists the commands for the Conda interpreter.

Conda Interpreter Commands

Table 4-1 Conda Interpreter Commands

Tasks	Commands	Role
Create a Conda environment.	`create -n <env_name> <python_version>`	• ADMIN
Remove a list of packages from a specified conda environment. It is also an alias for conda `uninstall`.	`remove -n <env_name> --all` **Note** The Conda environment is deleted from the user session.	• ADMIN • OML user

Table 4-1 (Cont.) Conda Interpreter Commands

Tasks	Commands	Role
List user-created local environment.	`env list`	• ADMIN • OML user
Remove user-created local environment.	`env remove -n <env_name>`	• ADMIN • OML user
List all packages and versions installed in active environment.	`list`	• ADMIN • OML user
Activate a user-created local environment.	`activate -n <env_name>`	• ADMIN • OML user
Deactivate the current environment.	`deactivate`	• ADMIN • OML user
Install an external package from a public Conda channel.	`install -n <env_name> <package_name>`	• ADMIN
Uninstall a specific package from a Conda environment. It is also an alias for `remove`.	`uninstall -n <env_name> <package_name>`	• ADMIN
Display information about current conda install.	`info`	• ADMIN • OML user
View the command line help.	`COMMANDNAME --help`	• ADMIN • OML user
Upload a Conda environment to object storage. **Note** This is an Oracle Autonomous AI Database specific command.	`upload --overwrite <env_name> --description 'some description' -t <name> <value>` **Note** You can provide many tags. For example: `-t <name1> <value1> -t <name2> <value2> ..`	• ADMIN

Table 4-1 (Cont.) Conda Interpreter Commands

Tasks	Commands	Role
Download and unpack a specific Conda environment from the object storage. **Note** This is an Oracle Autonomous AI Database specific command.	`download --overwrite <env_name>`	• ADMIN • OML user
List local environments available to the user.	`list-local-envs`	• ADMIN • OML user
List all Conda environments in the object storage. **Note** This is an Oracle Autonomous AI Database specific command.	`list-saved-envs --installed-packages -e <env_name>`	• ADMIN • OML user
Delete a Conda environment. **Note** This is an Oracle Autonomous AI Database specific command. **Note** The Conda environment is deleted from the Object Storage.	`delete <env_name>`	• ADMIN
Update conda packages to the latest compatible version.	`update`	• ADMIN
Upgrade the current conda package. It is also an alias for conda `update`.	`upgrade`	• ADMIN
Search for packages and display associated information. The input is a `MatchSpec`, a query language for conda packages.	`search`	• ADMIN • OML user

4.8.4.3 Create a Conda Environment for Python, Install a Python Package and Upload it to Object Storage

This topic demonstrates how to create a Conda environment `mypyenv` for Python packages, with Python 3.13.5 that is compatible with OML4Py. Here, you will also install the Python package seaborn from the `conda-forge` channel, and upload the Conda environment `mypyenv` to object storage.

Conda channels are the locations where packages are stored. They serve as the base for hosting and managing packages. Conda packages are downloaded from remote channels, which are URLs to directories containing conda packages. The conda command searches a set of channels. By default, packages are automatically downloaded and updated from the default channel. The `conda-forge` channel is a community channel made up of thousands of contributors, and is free for all to use. You can modify what remote channels are automatically searched. You might want to do this to maintain a private or internal channel.

> **Note**
>
> You must be signed in as ADMIN user to create the Conda environment.

> **Note**
>
> To avoid version conflicts, please do not install packages that are already included in OML4Py. For a complete list of pre-installed packages and their versions, see Python Libraries in OML4Py.

To create a Conda environment:

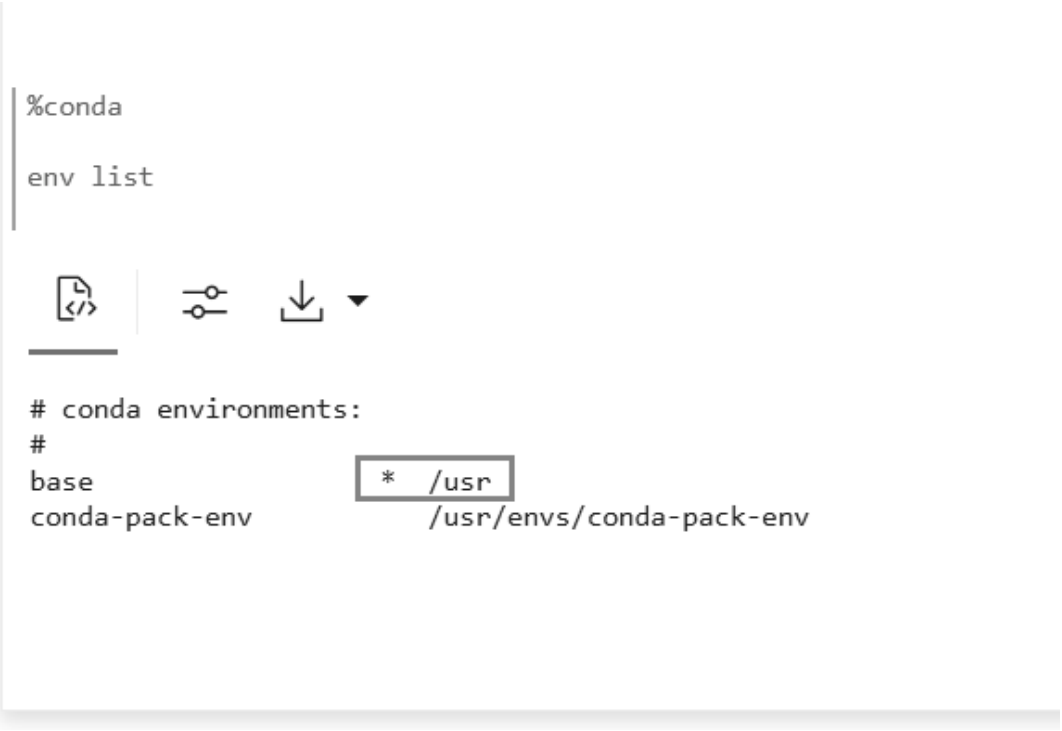
1. Run the following command to list the environments that are available by default. Conda contains default environments with some core system libraries and conda dependencies. Run the following command to list the available environments.

   ```
   %conda

   env list
   ```

   > **Note**
   >
   > The active environment is marked with an asterisk (*).

   The command returns the list of following environments:

Figure 4-87 Command to list Conda environments



2. Now, run the following command to create an environment by the name `mypyenv` from the `conda-forge` channel, install the Python package seaborn, and upload the conda environment `mypyenv` to Object Storage:

```
%conda

create -n mypyenv -c conda-forge --override-channels --strict-channel-
priority python=3.13.5 seaborn

upload mypyenv --overwrite -t application "OML4PY"
```

> **Note**
>
> This example uses Python version 3.13.5. You must ensure compatibility between the third-party package version and the Python version that OML4Py uses.

To determine the Python version being used, run the following command:

```
import sys
sys.version
```

In this command:

- `-n`: This is the name of the environment. In this example, it is `mypyenv`.
- `-c`: This is the channel name. In this example, the `conda-forge`channel is used to install the Python package.

- `--override-channels`: This argument ensures that the system does not search default, and requires a channel to be mentioned.
- `--strict-channel-priority`: This argument ensures that packages in lower priority channels are not considered if a package with the same name appears in a higher priority channel. In this example, the priority is given to Python 3.13.5 and seaborn.
- `upload`: This argument uploads the Conda environment named `mypyenv` to a target location.
- `--overwrite`: This argument ensures that if a file or directory with the same name already exists at the target location, it should be overwritten.
- `-t application`: This argument specifies the type of the target as an application.
- `"OML4PY"`: This is the name or path of the target application where the environment will be uploaded.

The command returns the following message once it creates the environment and installs the listed package. Scroll down the paragraph to view the complete details.

Figure 4-88 Command to create a Conda environment

```
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: /u01/.conda/envs/mypyenv

  added / updated specs:
    - python=3.13.5
    - seaborn


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    brotli-1.2.0               |       h41a2e66_0          19 KB  conda-forge
    brotli-bin-1.2.0           |       hf2c8021_0          21 KB  conda-forge
    contourpy-1.3.3            |  py313h7037e92_2         290 KB  conda-forge
    cycler-0.12.1              |     pyhd8ed1ab_1          13 KB  conda-forge
    fonttools-4.60.1           |  py313h3dea7bd_0         2.8 MB  conda-forge
    freetype-2.14.1            |       ha770c72_0         169 KB  conda-forge
    kiwisolver-1.4.9           |  py313hc8edb43_1          75 KB  conda-forge
```

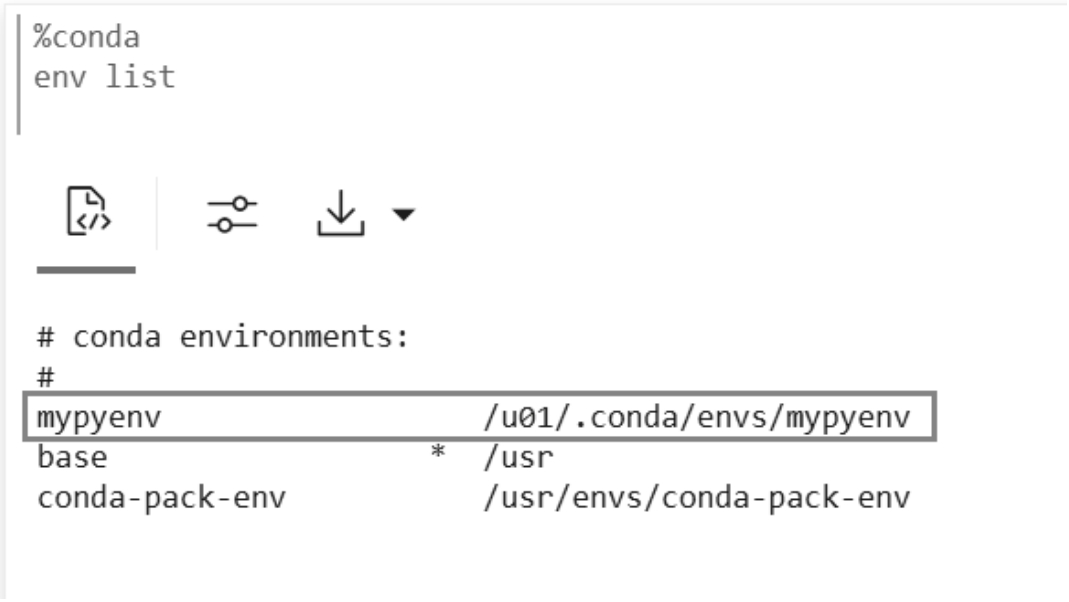
3. Now, verify the environment that you created. Run the following command once again to view the list of environments.

```
%conda

env list
```

> **Note**
>
> The `mypyenv` environment is not yet active. The active environment is marked with an asterisk (*), and in this list, it is present against the base environment usr.

Note that the environment `mypyenv` is listed in the output.

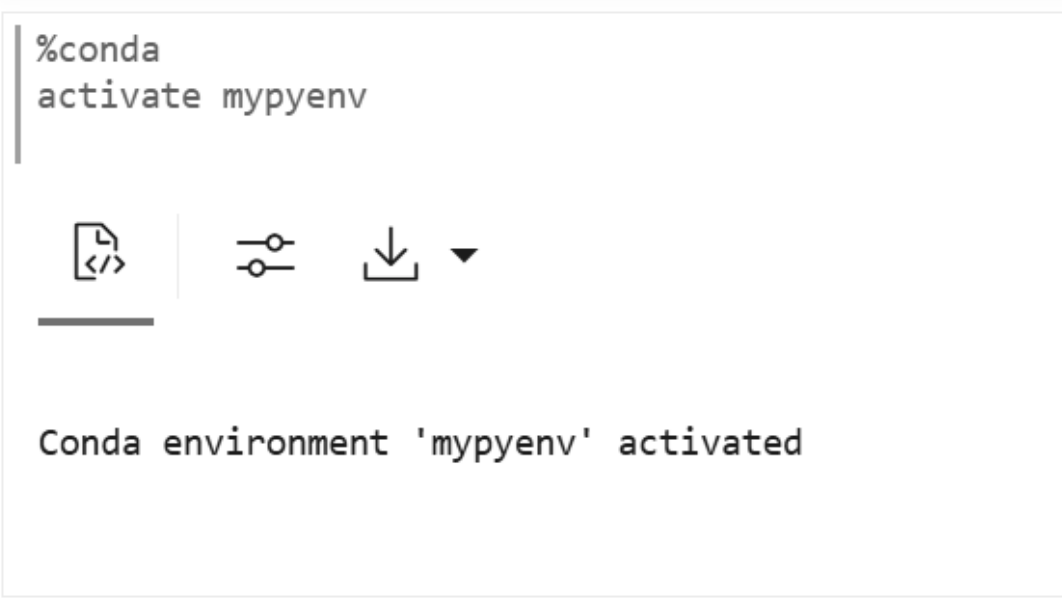
Figure 4-89 mypyenv environment created



4. To activate the environment `mypyenv`, run the following command:

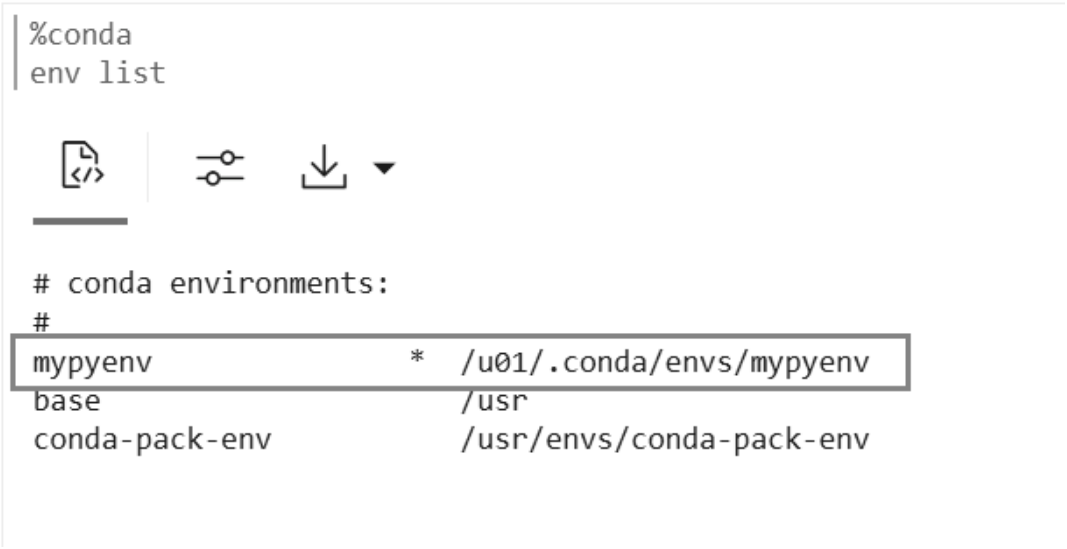
```
%conda

activate mypyenv
```

Figure 4-90 Command to activate Conda environment



5. Now, run the command to list the environment once again to view the list of active environments. Note that the asterisk (*) is now present against the `mypyenv` environment.

Figure 4-91 Activated environment mypyenv listed



The asterisk (*) next to the environment name confirms the activation. This completes the task of creating, activating and verifying a Conda environment.

4.8.4.4 Create a Conda Environment for R and Install R Package and Upload it to Object Storage

This topic shows how to create a Conda environment `myrenv` for R packages, with R-4.0.5 for OML4R compatibility, install the `forecast` package, and upload the Conda environment to object storage.

> **Note**
>
> You must be signed in as ADMIN user to create the Conda environment.

> **Note**
>
> To avoid version conflicts, please do not install packages that are already included in OML4R. For a complete list of pre-installed packages and their versions, see About the OML4R Supporting Packages

To create a Conda environment named `myrenv` with R-4.0.5 for OML4R compatibility, install the package `forecast`, and upload the Conda environment `myrenv` to object storage:

1. Run the following command to create the environment for R by the name `myrenv`, install the package `forecast` and upload the Conda environment `myrenv` to object storage:

```
%conda

create -n myrenv -c conda-forge --override-channels --strict-channel-
priority r-base=4.0.5 r-forecast

upload myrenv --overwrite -t application "OML4R"
```

> **Note**
>
> In this example, we are using the `conda-forge` channel to install the R packages.

In this command:

- `-n`: This is the name of the environment. In this example, it is `myrenv`.
- `-c`: This is the channel name. In this example, it is `conda-forge`.
- `--override-channels`: This argument ensures that the system does not search default, and requires a channel to be mentioned.
- `--strict-channel-priority`: This argument ensures that packages in lower priority channels are not considered if a package with the same name appears in a higher priority channel. In this example, the priority is given to `R-4.0.5` and `forecast`.
- `upload`: This argument uploads the Conda environment named `myrenv` to a target location.
- `--overwrite`: This argument ensures that if a file or directory with the same name already exists at the target location, it should be overwritten.

- `-t application`: This argument specifies the type of the target as an application.
- `"OML4R"`: This is the name or path of the target application where the environment will be uploaded.

The command returns the following:

Figure 4-92 Command to create Conda environment for R and install R Packages

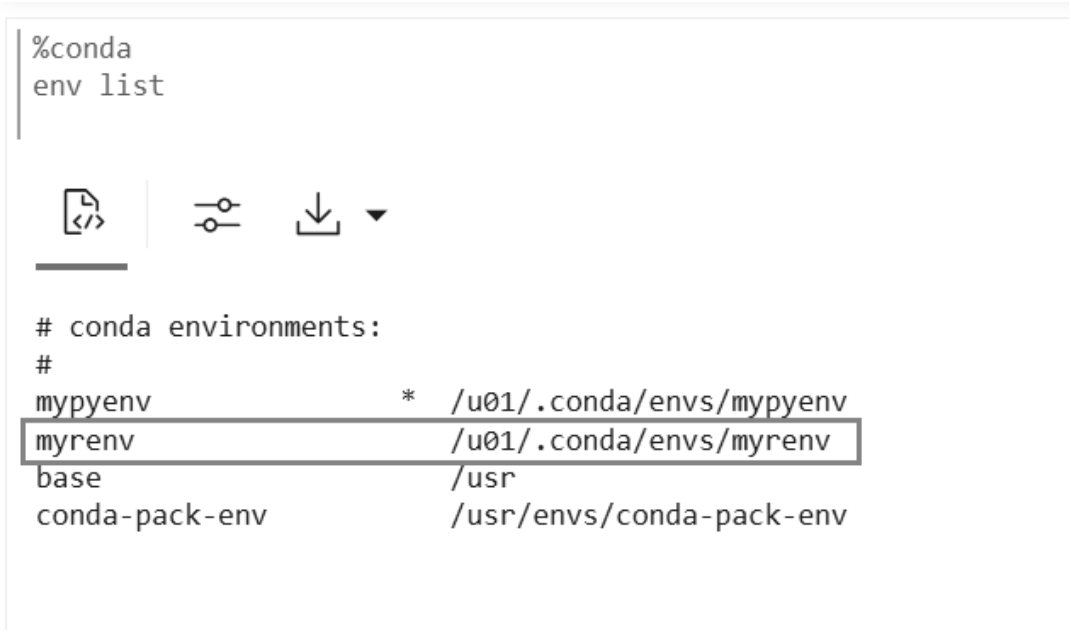
```
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: /u01/.conda/envs/myrenv

  added / updated specs:
    - r-base=4.0.5
    - r-forecast


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    _r-mutex-1.0.1             |      anacondar_1           3 KB  conda-forge
    binutils_impl_linux-64-2.44|       hdf8817f_2         3.6 MB  conda-forge
    bwidget-1.10.1             |       ha770c72_1         127 KB  conda-forge
    c-ares-1.34.5              |       hb9d3cd8_0         202 KB  conda-forge
    cairo-1.16.0               |    ha61ee94_1014         1.5 MB  conda-forge
    curl-7.86.0                |       h2283fc2_1          91 KB  conda-forge
    expat-2.7.1                |       hecca717_0         138 KB  conda-forge
    font-ttf-dejavu-sans-mono-2.37|       hab24e00_0         388 KB  conda-forge
```

2. You may now verify the Conda environment for R that you just created. Run the following command in a `%conda` paragraph to view the list of environments.

```
env list
```

Note that the `myrenv` environment is now listed along with the `mypyenv` in the output.

Figure 4-93 myrenv environment created and listed



Note that the asterisk (*) is against the `mypyenv`, indicating that this environment is active currently.

4.8.4.5 Upload the Conda environments to an Object Storage bucket associated with the Autonomous AI Database

There is one Object Storage bucket for each data center region. The Conda environment is saved to a folder in Object Storage corresponding to the tenancy and database. The folder is managed by Autonomous AI Database and is only available to users through Oracle Machine Learning Notebooks. There is an 8G maximum size for a single Conda environment, and no size limit on Object Storage.

This topic shows how to upload a Conda environment to the Object Storage bucket associated with the Autonomous AI Database instance using the `upload` command. You will provide environment descriptions and tags, one for the user name and one for the application name. You will also overwrite any environment with the same name if it exists. The application tag is required for use with embedded execution. For example, OML4Py embedded Python execution works with Conda environments containing the OML4PY tag.

To upload a Conda environment to the Object Storage bucket:

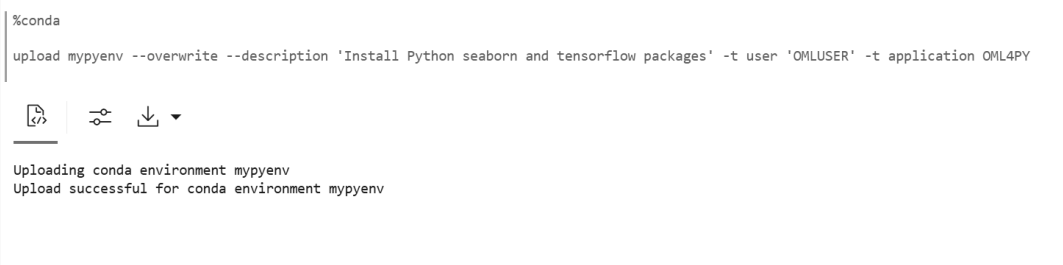
1. In a Conda paragraph in your notebook, run the following command to upload the `mypyenv` to the Object Storage bucket associated with the Autonomous AI Database.

   ```
   upload mypyenv --overwrite --description 'Install Python seaborn and
   tensorflow packages' -t user 'OMLUSER' -t application OML4PY
   ```

   In this command, you use the upload command and provide the following:

   - `--description`: a description of your task.
   - `-t`: A tag for the user `OMLUSER` and for the application `OML4PY`.

Figure 4-94 Command to load the Conda environment for Py

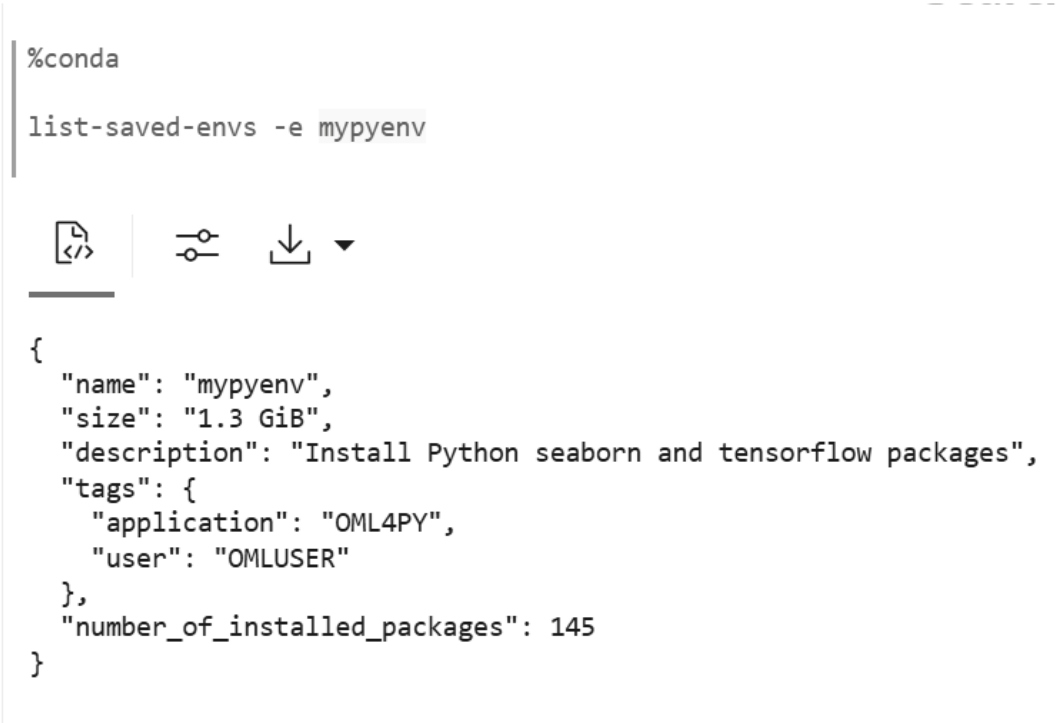


2. Now, run the following command to view the list of saved environments in `mypyenv`:

```
list-saved-envs -e mypyenv
```

The command returns the following:

Figure 4-95 View saved environment mypyenv

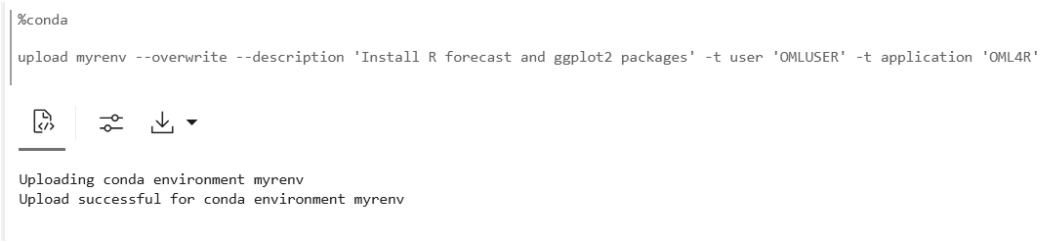


3. Now upload the `myrenv` to the object storage by running the following command in a Conda paragraph:

```
%conda

upload myrenv --overwrite --description 'Install R forecast and ggplot2
packages' -t user 'OMLUSER' -t application 'OML4R
```

The command returns the following:

Figure 4-96 Command to upload Conda environment for R

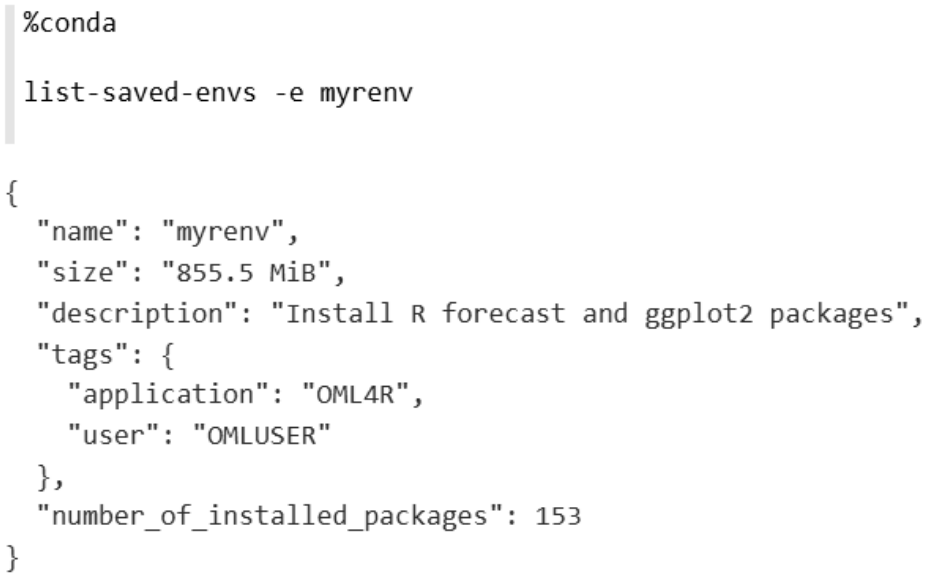


4. Again, run the following command to view the list of saved environments in `myrenv`:

```
%conda

list-saved-envs -e myrenv
```

The command returns the following:

Figure 4-97 View saved environment myrenv



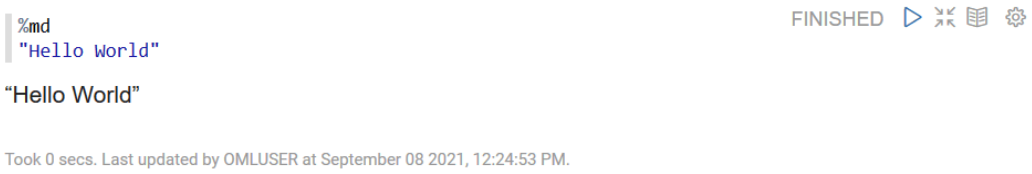
This completes the task of uploading the Python and R environments `mypyenv` and `myrenv` to the Object Storage associated with the Autonomous AI Database. The environments are now available for you to download and use. The environments will remain in the Object Storage until it is deleted.

4.8.5 Use the Markdown Interpreter and Generate Static html from Markdown Plain Text

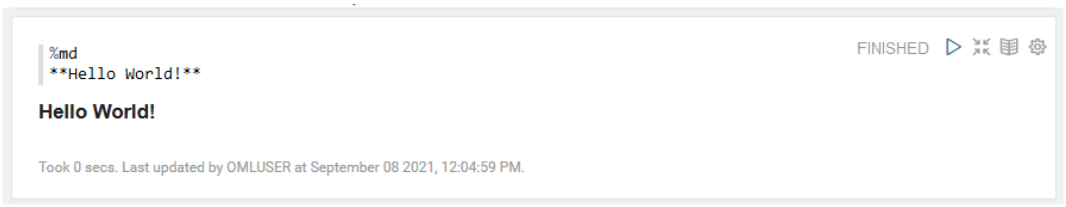
Use the Markdown interpreter and generate static html from Markdown plain text.

To call the Markdown interpreter and generate static html from Markdown plain text:

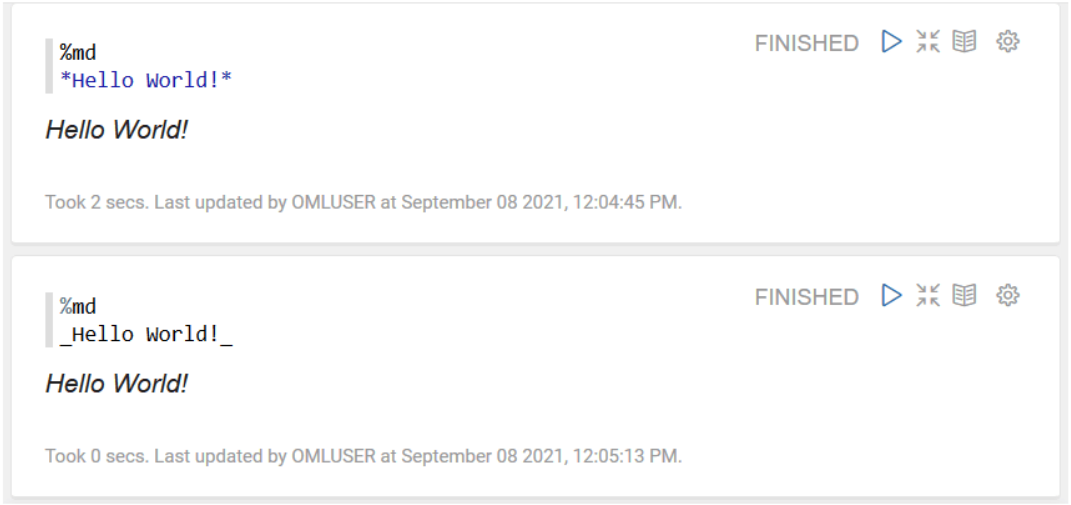
1. In your notebook, type `%md` and press Enter.
2. Type `"Hello World!"` and click Run. The static html text is generated, as seen in the screenshot below.



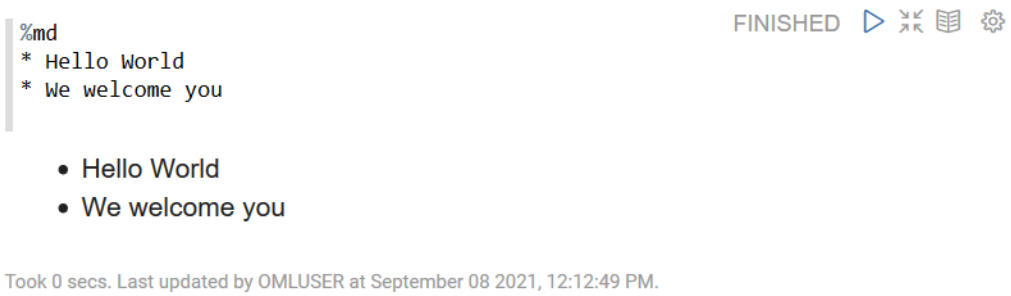
3. You can format the text in bold. To display the text in bold, write the same text inside two asterisks pair and click Run.



4. To display the text in italics, write the same text inside an asterisk pair or underscore pair as shown in the screenshot, and click Run.



5. To display the text in a bulleted list, prefix *(asterisk) to the text, as shown in the screenshot below:



6. To display the text in heading1, heading 2 and heading 2, prefix # (hash) to the text and click Run. For H1, H2, and H3, you must prefix one, two, and three hashes respectively.

4.9 Use the Scratchpad

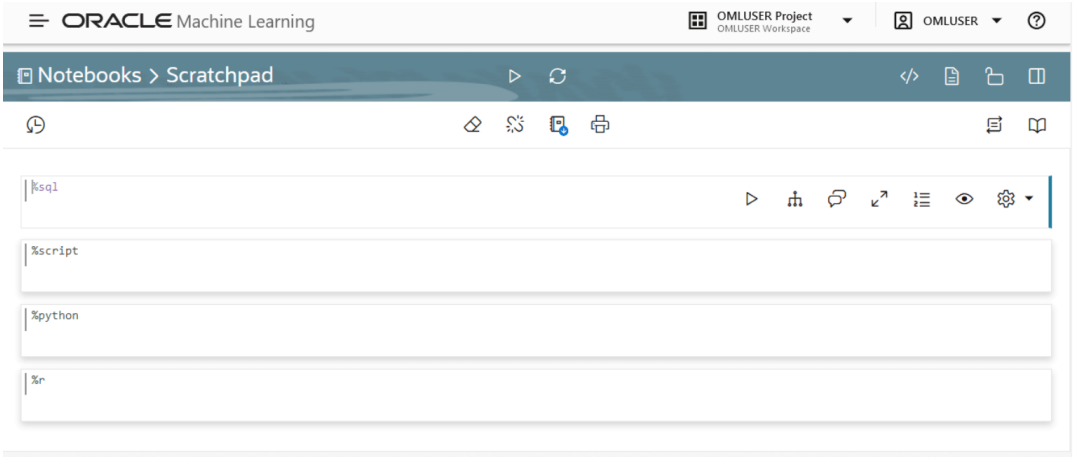
The Scratchpad provides you convenient one-click access to a notebook for running SQL statements, PL/SQL, R, and Python scripts that can be renamed. The Scratchpad is available on the Oracle Machine Learning User Interface (UI) home page.

> **Note**
>
> The Scratchpad is a regular notebook that is prepopulated with four paragraphs - `%sql, %script, , %python` and `%r.`

After you run your scripts, the Scratchpad is automatically saved as a notebook by the default name **Scratchpad** on the Notebooks page. You can access it later on the Notebooks page. You can run all the paragraphs together or one paragraph at a time.

1. To open and use the scratchpad, click **Scratchpad** on the Oracle Machine Learning UI home page under **Quick Actions.** The Scratchpad opens. The Scratchpad has four paragraphs each with the following directives:
   - `%sql`: Allows you to run SQL statements.
   - `%script`: Allows you to run PL/SQL scripts.
   - `%python`: Allows you to run Python scripts.
   - `%r`: Allows you to run R scripts.

Figure 4-98 Scratchpad

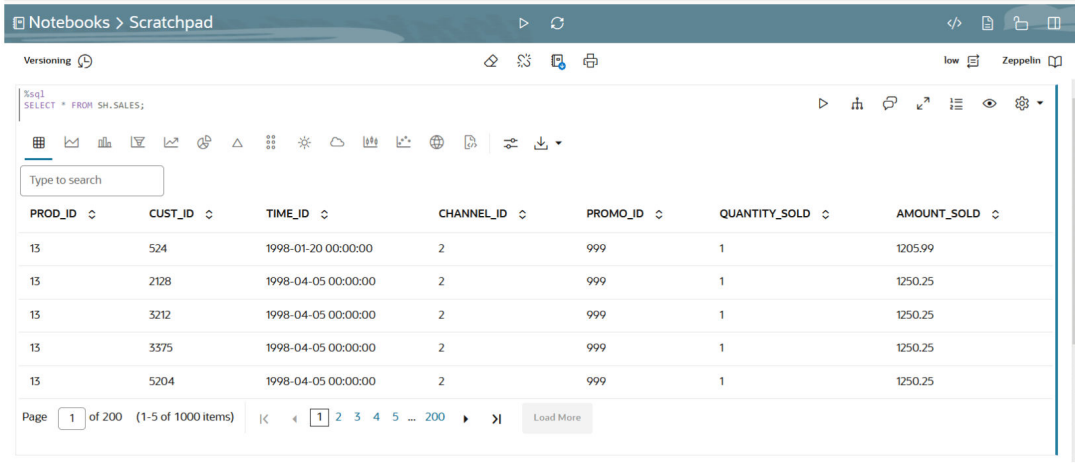


2. To run SQL script:
   a. Go to the paragraph with the `%sql` directive.
   b. Type the following command and click the Run icon. Alternatively, you can press Shift+Enter keys to run the paragraph.

```
SELECT * FROM SH.SALES;
```

In this example, the SQL statement fetches all of the data about product sales from the table SALES. Here, SH is the schema name, and SALES is the table name. Oracle Machine Learning UI fetches the relevant data from the database and displays it in a tabular format.

Figure 4-99 SQL Statement in Scratchpad



3. To run PL/SQL script:
   a. Go to the paragraph with the `%script` directive.
   b. Enter the following PL/SQL script and click the Run icon. Alternatively, you can press Shift+Enter keys to run the paragraph.

```
CREATE TABLE small_table
    (
```

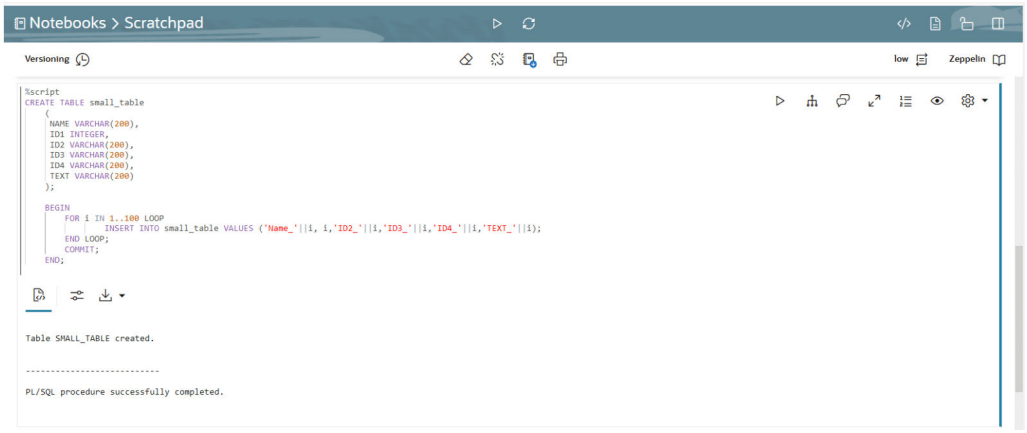
```
    NAME VARCHAR(200),
    ID1 INTEGER,
    ID2 VARCHAR(200),
    ID3 VARCHAR(200),
    ID4 VARCHAR(200),
    TEXT VARCHAR(200)
   );

   BEGIN
       FOR i IN 1..100 LOOP
                INSERT INTO small_table VALUES ('Name_'||i, i,'ID2_'||
i,'ID3_'||i,'ID4_'||i,'TEXT_'||i);
       END LOOP;
       COMMIT;
   END;
```

The PL/SQL script successfully creates the table SMALL_TABLE. The PL/SQL script in this example contains two parts:

- The first part of the script contains the SQL statement `CREATE TABLE` to create a table named `small_table`. It defines the table name, table column, data types, and size. In this example, the column names are `NAME`, `ID1`, `ID2`, `ID3`, `ID4`, and `TEXT`.
- The second part of the script begins with the keyword `BEGIN`. It inserts 100 rows in to the table `small_table`.

> **Note**
>
> When using the `CREATE` statement with a primary key, it fails and displays the error message Insufficient privileges. This error occurs due to lockdown profiles in the database. If you encounter this error, contact your database administrator or the designated security administrator to grant the required privileges.

Figure 4-100 PL/SQL Script in Scratchpad



4. To run python script:
   a. To use OML4Py, you must first import the `oml` module. `oml` is the OML4Py module that allows you to manipulate Oracle AI Database objects such as tables and views, call

user-defined Python functions using embedded execution, and use the database machine learning algorithms. Go to the paragraph with `%python` directive. To import the `oml` module, type the following command and click the Run icon. Alternatively, you can press Shift+Enter keys to run the paragraph.

```
import oml
```

b. To check if the `oml` module is connected to Oracle AI Database, type `oml.isconnected()` and click the Run icon. Alternatively, you can press Shift+Enter keys to run the paragraph.

```
oml.isconnected()
```

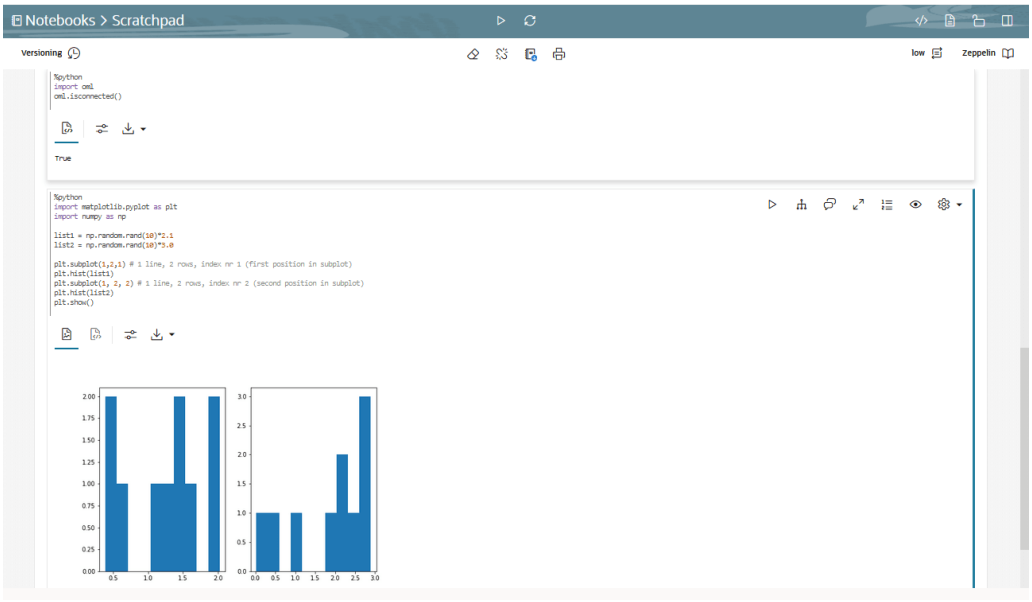
c. You are now ready to run your Python script. Type the following Python code and click the run icon. Alternatively, you can press Shift+Enter keys to run the paragraph.

```
import matplotlib.pyplot as plt
import numpy as np

list1 = np.random.rand(10)*2.1
list2 = np.random.rand(10)*3.0

plt.subplot(1,2,1) # 1 line, 2 rows, index nr 1 (first position in
subplot)
plt.hist(list1)
plt.subplot(1, 2, 2) # 1 line, 2 rows, index nr 2 (second position in
subplot)
plt.hist(list2)
plt.show()
```

In this example, the commands import two python packages to compute and render the data in two histograms for `list1` and `list2`. The Python packages are:

- Matplotlib: Python package to render graphs.
- Numpy: Python package for computations.

Figure 4-101 Python script in Scratchpad



The two graphs for `list1` and `list 2` are generated by the python engine, as shown in the screenshot here.

5. After you have created and run your scripts in the Scratchpad, the Scratchpad is automatically saved as a notebook by the name default name **Scratchpad** on the Notebooks page. You can edit the name of the notebook and save it with the new name by clicking **Edit.**

5
AutoML UI

AutoML User Interface (AutoML UI) is an Oracle Machine Learning interface that provides you no-code automated machine learning modeling. When you create and run an experiment in AutoML UI, it performs automated algorithm selection, feature selection, and model tuning, thereby enhancing productivity as well as potentially increasing model accuracy and performance.

> **Note**
>
> Oracle Machine Learning AutoML UI is desupported and will be unavailable starting January 15, 2027. Customers using OML AutoML UI should migrate to Oracle Data Science Agent, which uses new, advanced technology to automate data science and machine learning tasks and produce results superior to those provided by OML AutoML UI. For details, see Oracle Machine Learning Data Science Agent.

The following steps comprise a machine learning modeling workflow and are automated by the AutoML user interface:

1. Algorithm Selection: Ranks algorithms likely to produce a more accurate model based on the dataset and its characteristics, and some predictive features of the dataset for each algorithm.
2. Adaptive Sampling: Finds an appropriate data sample. The goal of this stage is to speed up Feature Selection and Model Tuning stages without degrading the model quality.
3. Feature Selection: Selects a subset of features that are most predictive of the target. The goal of this stage is to reduce the number of features used in the later pipeline stages, especially during the model tuning stage to speed up the pipeline without degrading predictive accuracy.
4. Model Tuning: Aims at increasing individual algorithm model quality based on the selected metric for each of the shortlisted algorithms.
5. Feature Prediction Impact: This is the final stage in the AutoML UI pipeline. Here, the impact of each input column on the predictions of the final tuned model is computed. The computed prediction impact provides insights into the behavior of the tuned AutoML model.

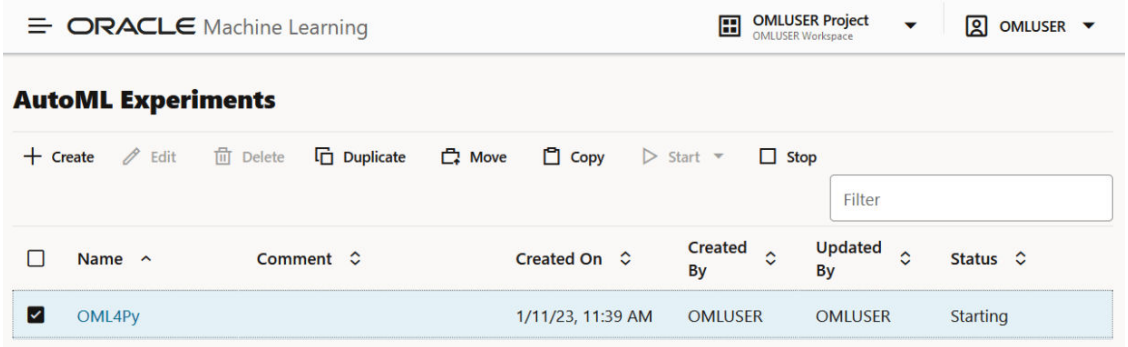
Business users without extensive data science background can use AutoML UI to create and deploy machine learning models. Oracle Machine Learning AutoML UI provides two functional features:

- Create machine learning models
- Deploy machine learning models

AutoML UI Experiments

When you create an experiment in AutoML UI, it automatically runs all the steps involved in the machine learning workflow. In the Experiments page, all the experiments that you have created are listed. To view any experiment details, click an experiment. Additionally, you can perform the following tasks:

Figure 5-1 AutoML Experiments page



- **Create:** Click **Create** to create a new AutoML UI experiment. The AutoML UI experiment that you create resides inside the project that you selected in the Project under the Workspace.
- **Edit:** Select any experiment that is listed here, and click **Edit** to edit the experiment definition.
- **Delete:** Select any experiment listed here, and click **Delete** to delete it. You cannot delete an experiment which is running. You must first stop the experiment to delete it.
- **Duplicate:** Select an experiment and click **Duplicate** to create a copy of it. The experiment is duplicated instantly and is in Ready status.
- **Move:** Select an experiment and click **Move** to move the experiment to a different project in the same or a different workspace. You must have either the `Administrator` or `Developer` privilege to move experiments across projects and workspaces.

  > **Note**
  >
  > An experiment cannot be moved if it is in RUNNING, STOPPING or STARTING states, or if an experiment already exists in the target project by the same name.

- **Copy:** Select an experiment and click **Copy** to copy the experiment to another project in the same or different workspace.
- **Start:** If you have created an experiment but have not run it, then click **Start** to run the experiment.
- **Stop:** Select an experiment that is running, and click **Stop** to stop the running of the experiment.
- Access AutoML UI
  You can access AutoML UI from the Oracle Machine Learning UI home page.
- Create AutoML UI Experiment
  To use the Oracle Machine Learning AutoML UI, you start by creating an experiment. An experiment is a unit of work that minimally specifies the data source, prediction target, and prediction type. After an experiment runs successfully, it presents you a list of machine learning models in order of model quality according to the metric selected. You can select any of these models for deployment or to generate a notebook. The generated notebook contains Python code using OML4Py and the specific settings AutoML used to produce the model.

- View an Experiment
  In the AutoML UI Experiments page, all the experiments that you have created are listed. Each experiment will be in one of the following stages: Completed, Running, and Ready.

Related Topics

- Automatic Machine Learning

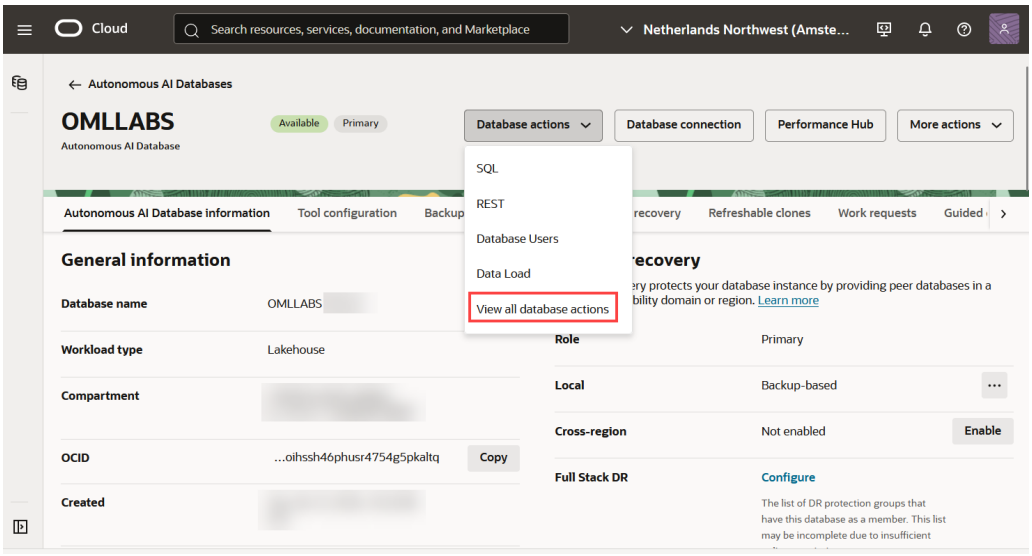
5.1 Access AutoML UI

You can access AutoML UI from the Oracle Machine Learning UI home page.

To access AutoML UI, you must first sign in to Oracle Machine Learning from Autonomous AI Database:

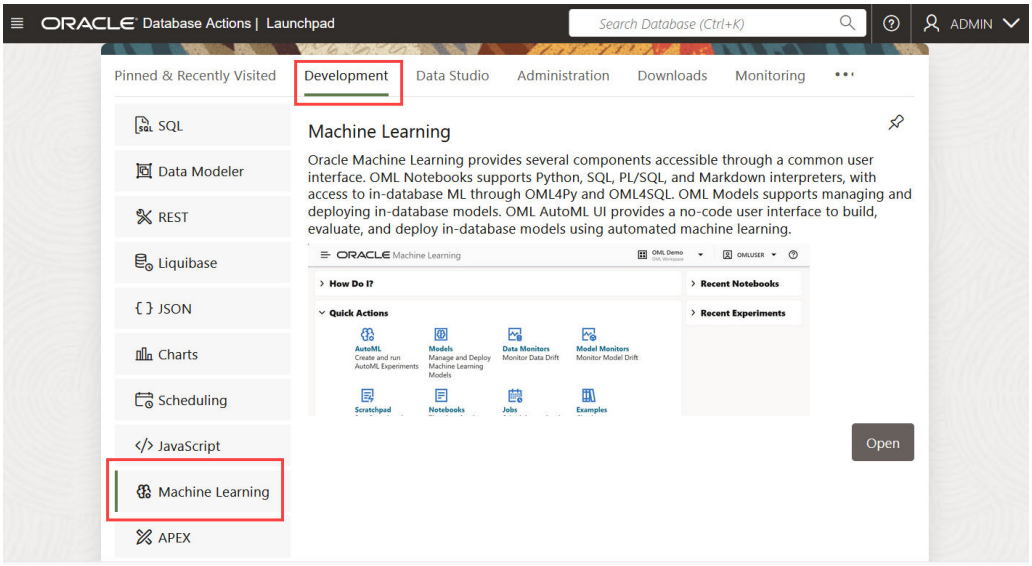
1. To sign in to Oracle Machine Learning from an Autonomous AI Database instance:
   a. Select an Autonomous AI Database instance and on the Autonomous AI Database information page, click **Database actions** and then click **View all database actions.**

Figure 5-2 Database Actions



   b. On the Database Actions page, go to the **Development** section and click **Machine Learning**.

Figure 5-3 Machine Learning option in Database Actions Launchpad



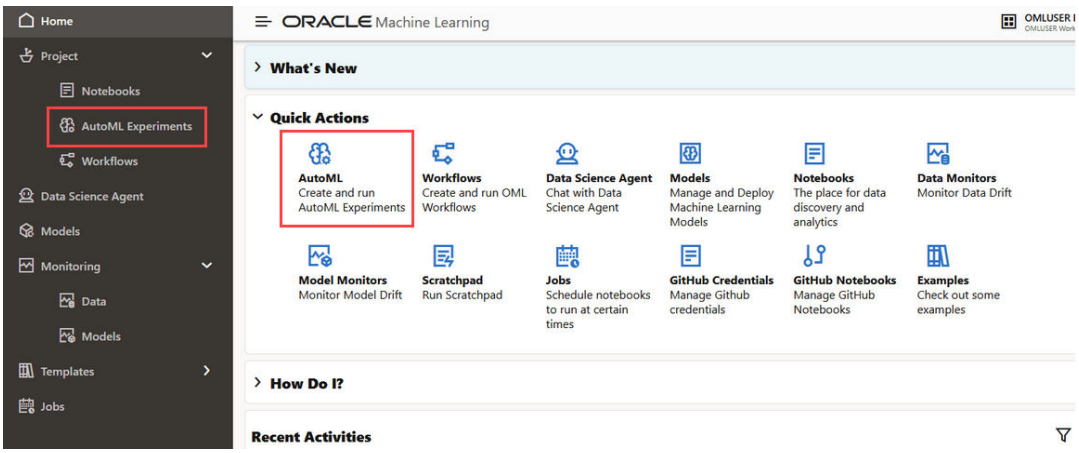
The Oracle Machine Learning UI sign in page opens.

c. Enter your username and password, and click **Sign in.**

This opens the Oracle Machine Learning UI home page.

2. On your Oracle Machine Learning UI home page, click **AutoML.**.

Figure 5-4 OML UI home page



Alternatively, you can click the Cloud menu icon and click **AutoML Experiments** under Projects. This opens the AutoML listing page.

5.2 Create AutoML UI Experiment

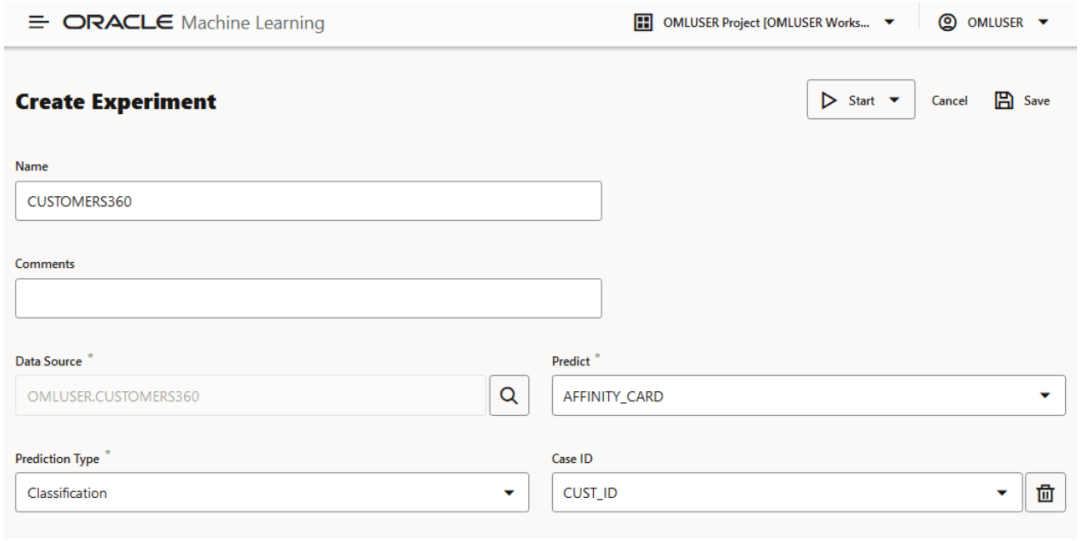
To use the Oracle Machine Learning AutoML UI, you start by creating an experiment. An experiment is a unit of work that minimally specifies the data source, prediction target, and prediction type. After an experiment runs successfully, it presents you a list of machine learning models in order of model quality according to the metric selected. You can select any of these

models for deployment or to generate a notebook. The generated notebook contains Python code using OML4Py and the specific settings AutoML used to produce the model.

To create an experiment, specify the following:

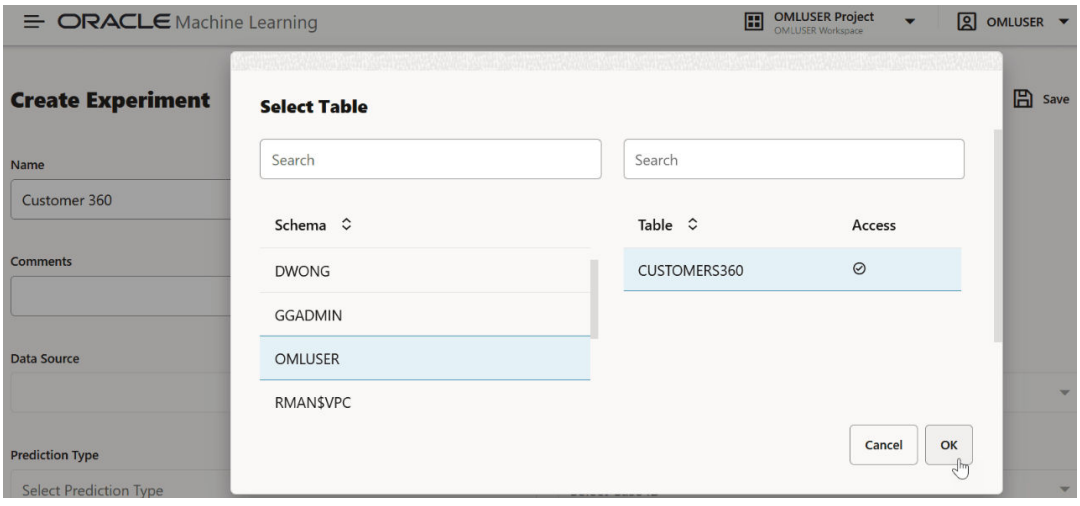
1. In the **Name** field, enter a name for the experiment.

Figure 5-5 Create an AutoML Experiment



2. In the **Comments** field, enter comments, if any.
3. In the **Data Source** field, select the schema and a table or view in that schema. Click the search icon to open the Select Table dialog box. Browse and select a schema, and then select a table from the schema list, which is the data source of your AutoML UI experiment.

Figure 5-6 Select Table dialog



a. In the Schema column, select a schema.

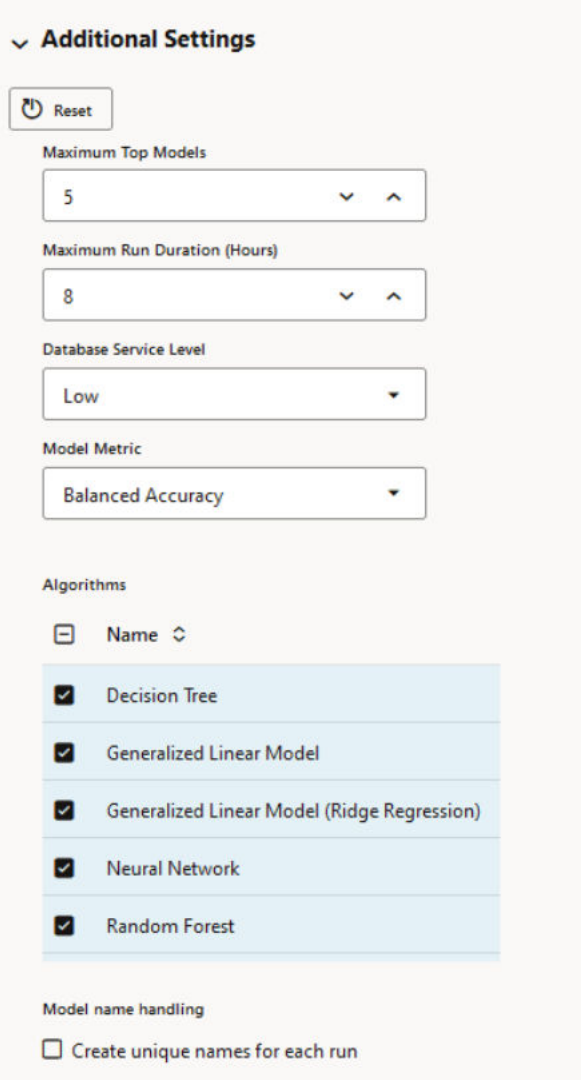
> **Note**
>
> While you select the data source, statistics are displayed in the Features grid at the bottom of the experiment page. Busy status is indicated until the computation is complete. The target column that you select in Predict is highlighted in the Features grid.

b. Depending on the selected schema, the available tables are listed in the Table column. Select the table and click **OK.**

> **Note**
>
> To create an AutoML experiment for a table or view present in the schema of another user, ensure that you have explicit privileges to access that table or view in the schema. Request the Database Administrator or the owner of the schema to provide you with the privileges to access the table or view. For example:
>
> ```
> grant select on <table> to <user>
> ```

4. In the **Predict** drop-down list, select the column from the selected table. This is the target for your prediction.
5. In the **Prediction Type** field, the prediction type is automatically selected based on your data definition. However, you can override the prediction type from the drop-down list, if data type permits. Supported Prediction Types are:
   - **Classification:** For non-numeric data type, Classification is selected by default.
   - **Regression:** For numeric data type, Regression is selected by default.
6. The **Case ID** helps in data sampling and dataset split to make the results reproducible between experiments. It also aids in reducing randomness in the results. This is an optional field.
7. In the **Additional Settings** section, you can define the following:

Figure 5-7 Additional Settings for AutoML Experiments



a. **Reset:** Click **Reset** to reset the settings to the default values.

b. **Maximum Top Models:** Select the maximum number of top models to create. The default is `5` models. You can reduce the number of top models to 2 or 3 since tuning models to get the top one for each algorithm requires additional time. If you want to get the initial results even faster, consider the top recommended algorithm. For this, set the **Maximum Top Models** to `1`. This will tune the model for that algorithm.

c. **Maximum Run Duration:** This is the maximum time for which the experiment will be allowed to run. If you do not enter a time, then the experiment will be allowed to run for up to the default, which is `8` hours.

d. **Database Service Level:** This is database connection service level and query parallelism level. Default is `Low`. This results in no parallelism and sets a high runtime limit. You can create many connections with `Low` database service level. You can also change your database service level to `Medium` or `High`.

   - `High` level gives the greatest parallelism but significantly limits the number of concurrent jobs.

- `Medium` level enables some parallelism but allows greater concurrency for job processing.

> **Note**
>
> Changing the database service level setting on the *Always Free Tier* will have no effect since there is a 1 OCPU limit. However, if you increase the OCPUs allocated to your Autonomous AI Database instance, you can increase the **Database Service Level** to `Medium` or `High`.

> **Note**
>
> The **Database Service Level** setting has no effect on AutoML container level resources.

e. **Model Metric:** Select a metric to choose the winning models. The following metrics are supported by AutoML UI:

- For Classification, the supported metrics are:
  - Balanced Accuracy
  - ROC AUC
  - F1 (with weighted options). The weighted options are weighted, binary, micro and macro.
    - Micro-averaged: Here, all samples equally contribute to the final averaged metric
    - Macro-averaged: Here, all classes equally contribute to the final averaged metric
    - Weighted-averaged: Here, each class' contribution to the average is weighted by its size
  - Precision (with weighted options)
  - Recall (with weighted options)
- For Regression, the supported metrics are:
  - R2 (default)
  - Negative mean squared error
  - Negative mean absolute error
  - Negative median absolute error

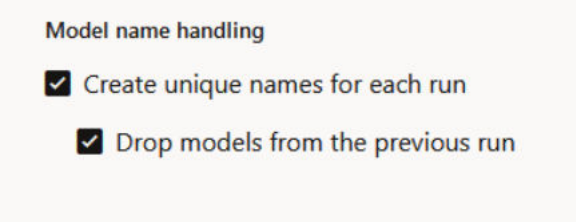
f. **Algorithm:** The supported algorithms depend on **Prediction Type** that you have selected. Click the corresponding checkbox against the algorithms to select it. By default, all the candidate algorithms are selected for consideration as the experiment runs. The supported algorithms for the two Prediction Types:

- For Classification, the supported algorithms are:
  - Decision Tree
  - Generalized Linear Model
  - Generalized Linear Model (Ridge Regression)

  - Neural Network
  - Random Forest
  - Support Vector Machine (Gaussian)
  - Support Vector Machine (Linear)
- For Regression, the supported algorithms are:
  - Generalized Linear Model
  - Generalized Linear Model (Ridge Regression)
  - Neural Network
  - Support Vector Machine (Gaussian)
  - Support Vector Machine (Linear)

> **Note**
>
> You can remove algorithms from being considered if you have preferences for particular algorithms, or have specific requirements. For example, if model transparency is essential, then excluding models such as Neural Network would make sense. Note that some algorithms are more compute intensive than others. For example, Naïve Bayes and Decision Tree are normally faster than Support Vector Machine or Neural Network.

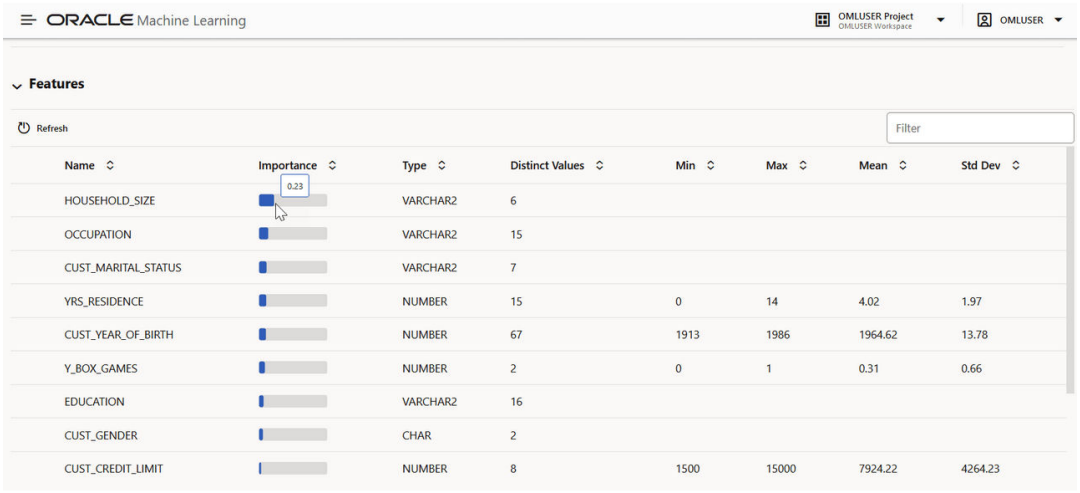
g. **Model Name Handling:** Here, you have the option to retain the original model name or generate unique model names everytime you run an experiment. By default, this option is deselected.



- **Create unique names for each run:** Select this option to generate model names that are unique. Selecting this option also gives you the choice to select or deselect the option **Drop models from the previous run.**
  - **Drop models from the previous run:** Select this option to drop the models that were generated in the prior experiment runs. Deselect this option to retain the models that were generated in the prior runs. These models are available in the user schema.

8. Expand the **Features** grid to view the statistics of the selected table. The supported statistics are Distinct Values, Minimum, Maximum, Mean, and Standard Deviation. The supported data sources for Features are tables, views and analytic views. The target column that you selected in Predict is highlighted here. After an experiment run is completed, the Features grid displays an additional column **Importance**. Feature Importance indicates the overall level of sensitivity of prediction to a particular feature.

Figure 5-8 Features

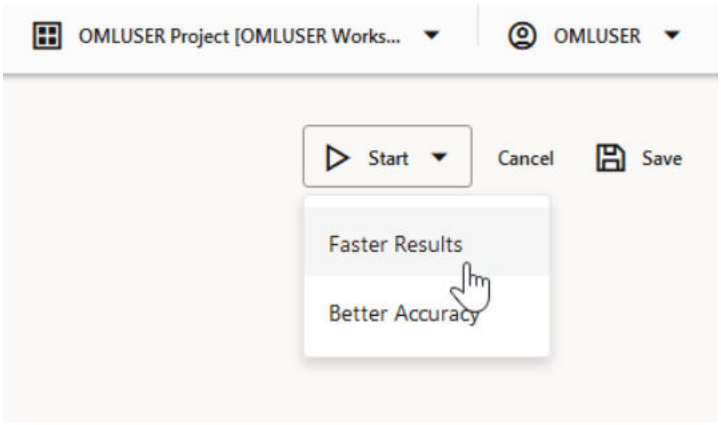


You can perform the following tasks:

- **Refresh:** Click Refresh to fetch all columns and statistics for selected data source.
- **View Importance:** Hover your cursor over the horizontal bar under Importance to view the value of Feature Importance for the variables. The value is always depicted in the range `0` to `1`, with values closer to `1` being more important.

9. When you complete defining the experiment, the **Start** and **Save** buttons are enabled.

Figure 5-9 Start Experiment options



- Click **Start** to run the experiment and start the AutoML UI workflow, which is displayed in the progress bar. Here, you have the option to select:
  a. **Faster Results:** Select this option if you want to get candidate models sooner, possibly at the expense of accuracy. This option works with a smaller set of the hyperparamter combinations, and hence yields faster result.
  b. **Better Accuracy:** Select this option if you want more pipeline combinations to be tried for possibly more accurate models. A pipeline is defined as an algorithm, selected data feature set, and set of algorithm hyperparameters.

> **Note**
>
> This option works with the broader set of hyperparameter options recommended by the internal meta-learning model. Selecting **Better Accuracy** will take longer to run your experiment, but may provide models with more accuracy.

Once you start an experiment, the progress bar appears displaying different icons to indicate the status of each stage of the machine learning workflow in the AutoML experiment. The progress bar also displays the time taken to complete the experiment run. To view the message details, click on the respective message icons.

- Click **Save** to save the experiment, and run it later.
- Click **Cancel** to cancel the experiment creation.

- Supported Data Types for AutoML UI Experiments
  When creating an AutoML experiment, you must specify the data source and the target of the experiment. This topic lists the data types for Python and SQL that are supported by AutoML experiments.

5.2.1 Supported Data Types for AutoML UI Experiments

When creating an AutoML experiment, you must specify the data source and the target of the experiment. This topic lists the data types for Python and SQL that are supported by AutoML experiments.

Table 5-1 Supported Data Types by AutoML Experiments

Data Types	SQL Data Types	Python Data Types
Numerical	• NUMBER • INTEGER • FLOAT • NUMERICALS • BINARY_DOUBLE • BINARY_FLOAT	• INTEGER • FLOAT(NUMBER) • FLOAT(BINARY_DOUBLE) • FLOAT(BINARY_FLOAT)
Categorical	• CHAR • VARCHAR2 • CATEGORICALS	• STRING(VARCHAR2) • STRING(CHAR) • STRING(CLOB)
Unstructured Text	• CHAR • VARCHAR2 • CLOB • BLOB • BFILE	• BYTES(RAW, BLOB)

5.3 View an Experiment

In the AutoML UI Experiments page, all the experiments that you have created are listed. Each experiment will be in one of the following stages: Completed, Running, and Ready.

To view an experiment, click the experiment name. The Experiment page displays the details of the selected experiment. It contains the following sections:

Edit Experiment

In this section, you can edit the selected experiment. Click **Edit** to make edits to your experiment.

> **Note**
>
> You cannot edit an experiment that is running.

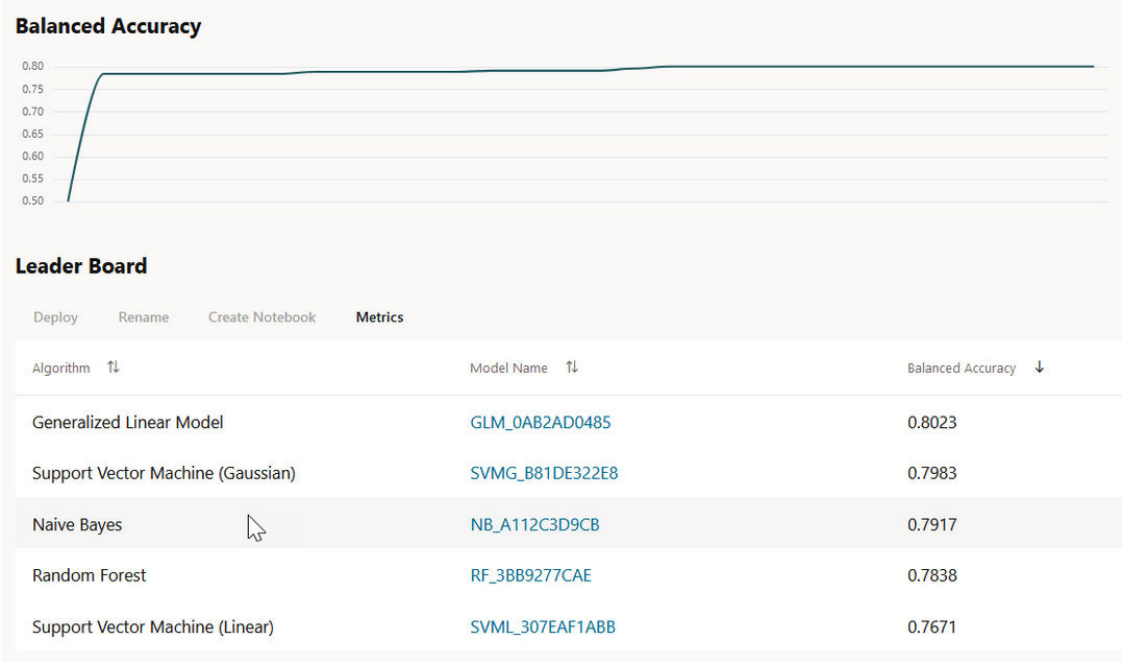
Metric Chart

The Model Metric Chart depicts the best metric value over time as the experiment runs. It shows improvement in accuracy as the running of the experiment progresses. The display name depends on the selected model metric when you create the experiment.

Leader Board

When an experiment runs, it starts to show the results in the Leader Board. The Leader Board displays the top performing models relative to the model metric selected along with the algorithm and accuracy. You can view the model details and perform the following tasks:

Figure 5-10 Leader Board



- View Model Details: Click on the **Model Name** to view the details. The model details are displayed in the Model Details dialog box. You can click multiple models on the Leader Board, and view the model details simultaneously. The Model Details window depicts the following:
  - Prediction Impact: Displays the importance of the attributes in terms of the target prediction of the models.

  - Confusion Matrix: Displays the different combination of actual and predicted values by the algorithm in a table. Confusion Matrix serves as a performance measurement of the machine learning algorithm.

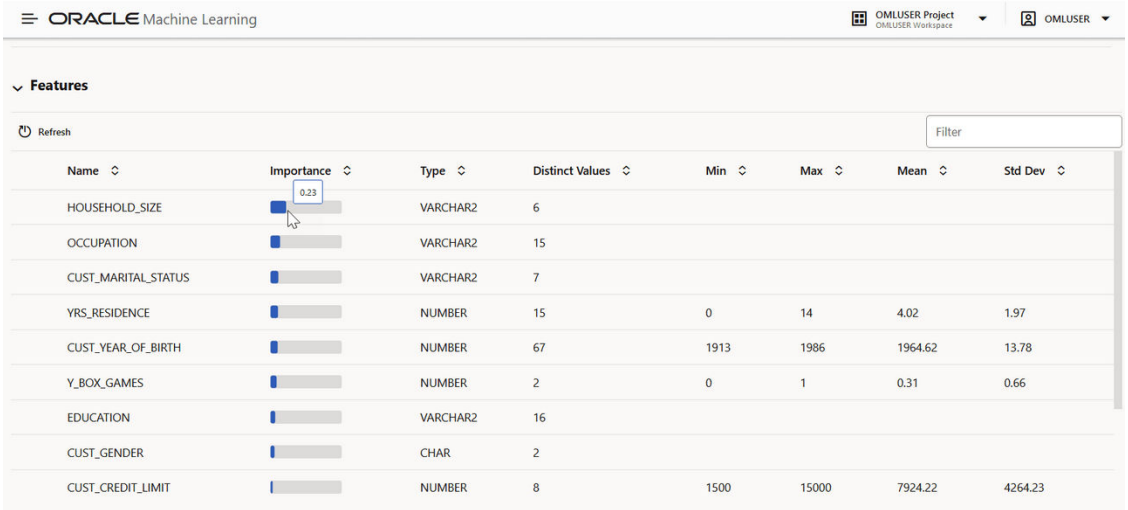
- **Deploy:** Select any model on the Leader Board and click **Deploy** to deploy the selected model. Deploy Model.
- **Rename:** Click **Rename** to change the name of the system generated model name. The name must be alphanumeric (not exceeding 123 characters) and must not contain any blank spaces.
- **Create Notebook:** Select any model on the Leader Board and click Create Notebooks from AutoML UI Models to recreate the selected model from code.
- **Metrics:** Click **Metrics** to select additional metrics to display in the Leader Board. The additional metrics are:
  - For Classification
    * Accuracy: Calculates the proportion of correctly classifies cases - both Positive and Negative. For example, if there are a total of TP (True Positives)+TN (True Negatives) correctly classified cases out of TP+TN+FP+FN (True Positives+True Negatives+False Positives+False Negatives) cases, then the formula is: `Accuracy = (TP+TN)/(TP+TN+FP+FN)`
    * Balanced Accuracy: Evaluates how good a binary classifier is. It is especially useful when the classes are imbalanced, that is, when one of the two classes appears a lot more often than the other. This often happens in many settings such as Anomaly Detection etc.
    * Recall: Calculates the proportion of actual Positives that is correctly classified.
    * Precision: Calculates the proportion of predicted Positives that is True Positive.
    * F1 Score: Combines precision and recall into a single number. F1-score is computed using harmonic mean which is calculated by the formula: `F1-score = 2 × (precision × recall)/(precision + recall)`
  - For Regression:
    * R2 (Default): A statistical measure that calculates how close the data are to the fitted regression line. In general, the higher the value of R-squared, the better the model fits your data. The value of R2 is always between 0 to 1, where:
      * `0` indicates that the model explains none of the variability of the response data around its mean.
      * `1` indicates that the model explains all the variability of the response data around its mean.
    * Negative Mean Squared Error: This is the mean of the squared difference of predicted and true targets.
    * Negative Mean Absolute Error: This is the mean of the absolute difference of predicted and true targets.
    * Negative Median Absolute Error: This is the median of the absolute difference between predicted and true targets.

Features

The **Features** grid displays the statistics of the selected table for the experiment.The supported statistics are Distinct Values, Minimum, Maximum, Mean, and Standard Deviation. The supported data sources for Features are tables, views and analytic views. The target column that you selected in Predict is highlighted here. After an experiment run is completed,

the Features grid displays an additional column **Importance**. Feature Importance indicates the overall level of sensitivity of prediction to a particular feature. Hover your cursor over the graph to view the value of **Importance.** The value is always depicted in the range `0` to `1`, with values closer to `1` being more important.

Figure 5-11 Features



- Create Notebooks from AutoML UI Models
  You can create notebooks using OML4Py code that will recreate the selected model using the same settings. It also illustrates how to score data using the model. This option is helpful if you want to use the code to re-create a similar machine learning model.

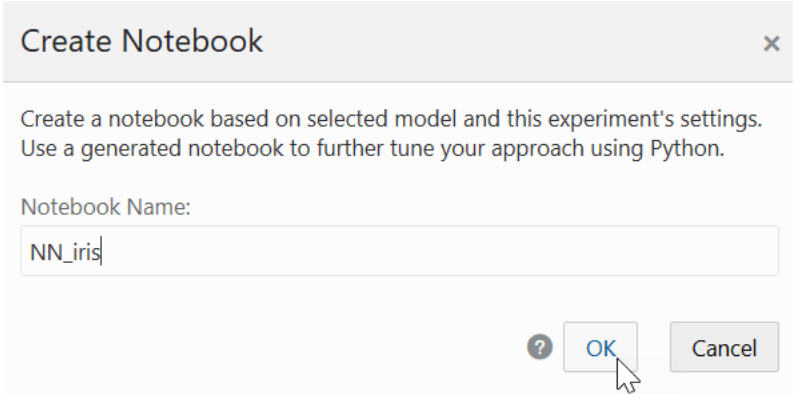
5.3.1 Create Notebooks from AutoML UI Models

You can create notebooks using OML4Py code that will recreate the selected model using the same settings. It also illustrates how to score data using the model. This option is helpful if you want to use the code to re-create a similar machine learning model.

To create a notebook from an AutoML UI model:

1. Select the model on the Leader Board based on which you want to create your notebook, and click **Create Notebook.** The Create Notebook dialog opens.

Figure 5-12 Create Notebook dialog



2. In the **Notebook Name** field, enter a name for your notebook.

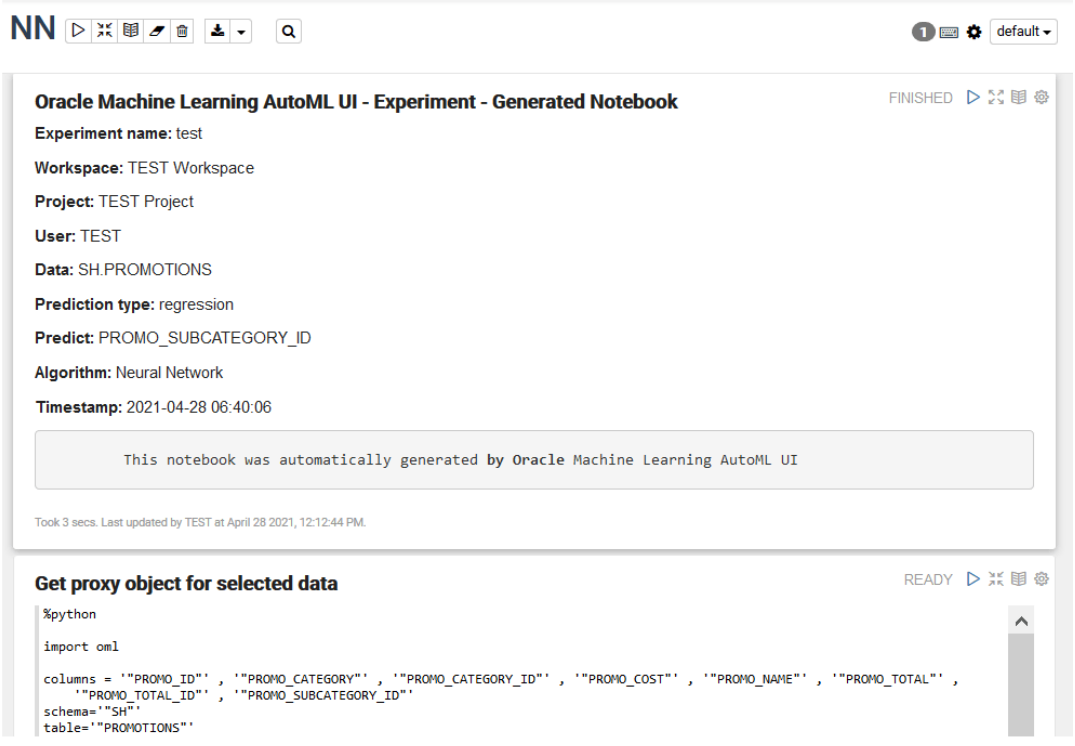
   The REST API endpoint derives the experiment metadata, and determines the following settings as applicable:

   - Data Source of the experiment (schema.table)
   - Case ID. If the Case ID for the experiment is not available, then the appropriate message is displayed.
   - A unique model name based on the current model name is generated
   - Information related to scoring paragraph:
     - Case ID: If available, then it merges the Case ID column into the scoring output table
     - Generate unique predict output table name based on build data source and unique suffix
     - Prediction column name: PREDICTION
     - Prediction probability column name: PROBABILITY (applicable only for Classification)

3. Click **OK.** The generated notebook is listed in the Notebook page. Click to open the notebook

   The generated notebook displays paragraph titles for each paragraph along with the python codes. Once you run the notebook, it displays information related to the notebook as well as the AutoML experiment such as the experiment name, workspace and project in which the notebook is present, the user, data, prediction type and prediction target, algorithm, and the time stamp when the notebook is generated.

Figure 5-13 AutoML UI Generated notebook



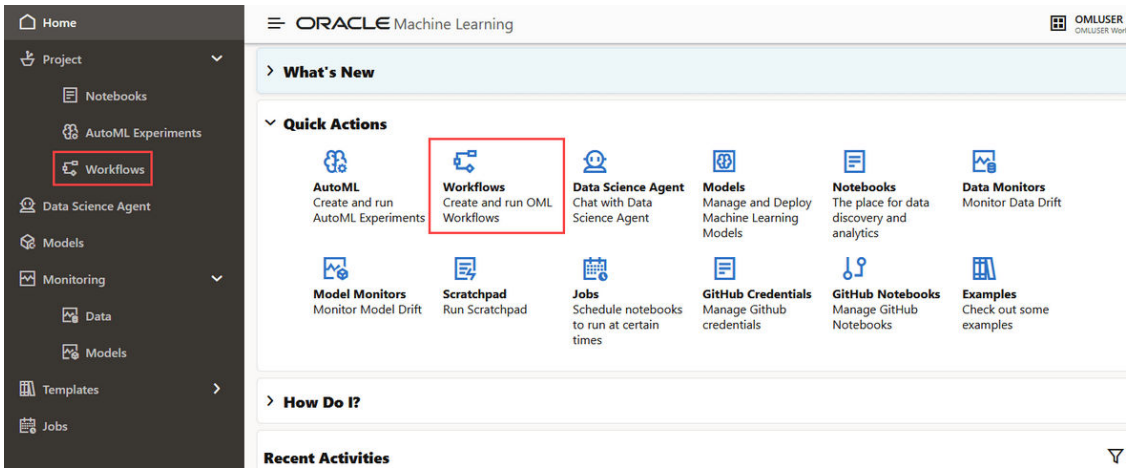
6

Workflows

Workflows enable you to design machine learning processes using a drag-and-drop canvas composed of nodes and connectors. Each node represents a task or step in the machine learning workflow. The connectors define the flow between these steps.

To access Workflows, click **Workflows** on your OML UI home page. Alternatively, you can select **Workflows** under Project on the left navigation menu to open the Workflows page.

Figure 6-1 Home page with Workflows icon highlighted



The Workflows page lists all the workflows that are created.

Figure 6-2 Workflows Listing page

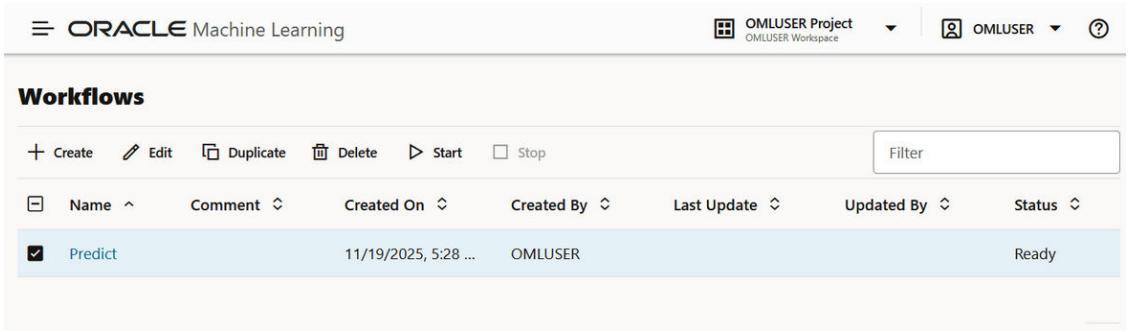
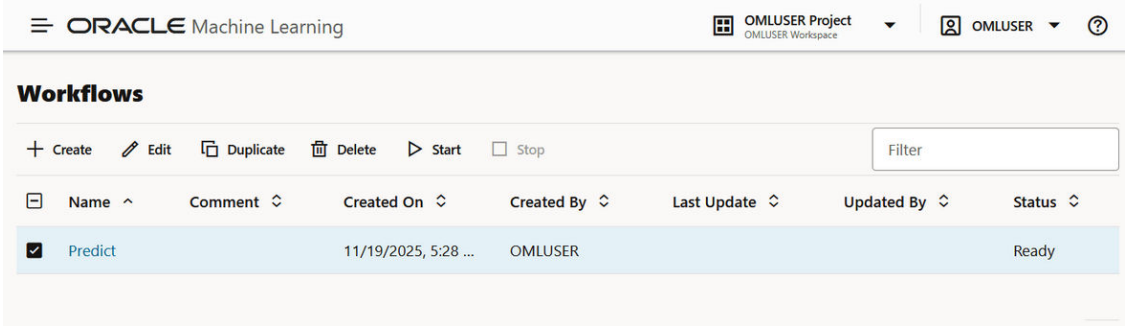


Figure 6-3 Workflows Listing page



You can perform the following tasks:

- **Create:** Click **Create** to create a new workflow.
- **Edit:** Select a workflow from the list and click **Edit** to edit it.
- **Duplicate:** Select any workflow from the list and click **Duplicate** to create a copy. The duplicated workflow will immediately appear with a `Ready` status.
- **Delete:** Select any workflow from the list and click **Delete** to remove it. You cannot delete a workflow that is currently running. You must stop it before attempting to delete.
- **Start:** If you have created a workflow but have not run it, click **Start** to run the workflow.
- **Stop:** If a workflow is running, select it and click **Stop** to halt the running of the workflow.
- High-Level Steps to Create a Workflow
  This topic lists the high-level steps to create a workflow.
- Create a Workflow
  A workflow is a collection of interconnected machine learning tasks or operations, represented by workflow nodes. Each node represents a distinct computational task—such as data import, data transformation, feature selection, model training, model evaluation, or model scoring—within the workflow. Connectors define the sequence and dependencies between nodes.
- Workflow Nodes
  A workflow node is an individual computational component in a workflow. Each node represents a specific task or operation, such as importing and preparing data, training a model, or evaluating results.

6.1 High-Level Steps to Create a Workflow

This topic lists the high-level steps to create a workflow.

To create a workflow:

1. Open an existing workflow, or click **Create** on the Workflow listing page.
2. Add workflow nodes to the canvas by dragging them from the palette or by clicking their Add buttons.

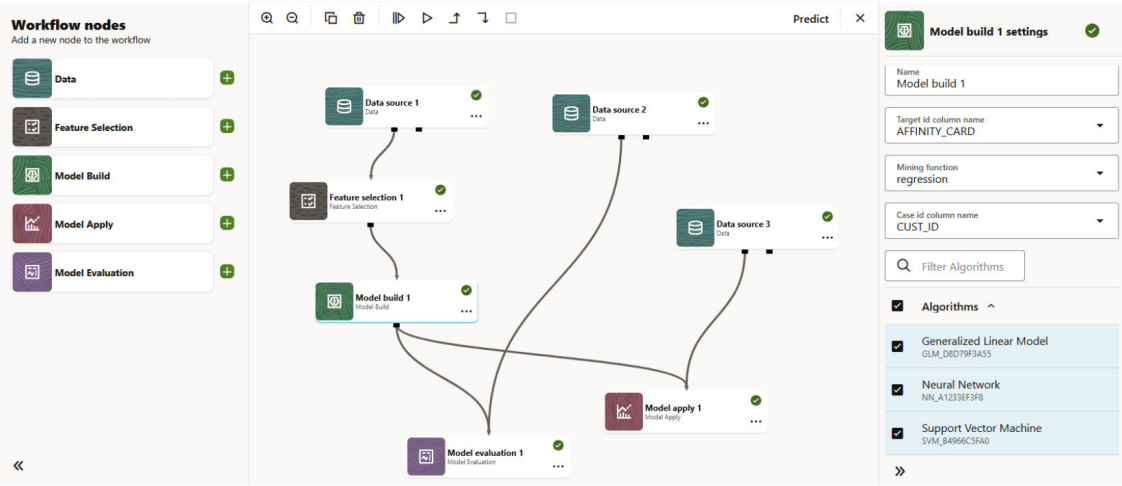
> **Note**
>
> This is a typical flow of nodes. The nodes can be used in multiple ways and you need not use all the nodes in a workflow.

a. Data Source Node
b. Feature Selection Node
c. Model Build Node
d. Model Apply Node
e. Model Evaluation Node
f. Deploy a model

3. Configure the settings for each node.
4. Run the nodes individually, or click **Run All** to run the entire workflow.

Here is a screenshot depicting a workflow with all node types.

Figure 6-4 A workflow with all node types



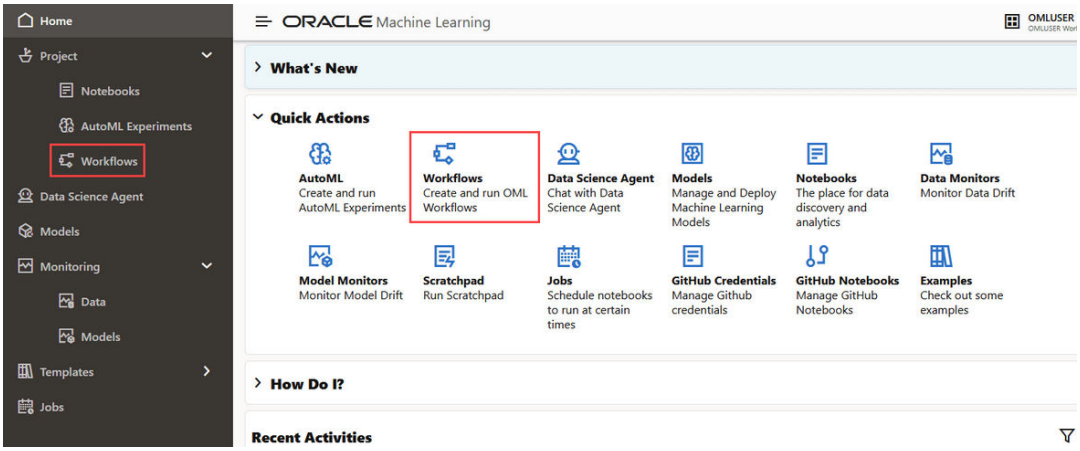
6.2 Create a Workflow

A workflow is a collection of interconnected machine learning tasks or operations, represented by workflow nodes. Each node represents a distinct computational task—such as data import, data transformation, feature selection, model training, model evaluation, or model scoring—within the workflow. Connectors define the sequence and dependencies between nodes.

To create a workflow:

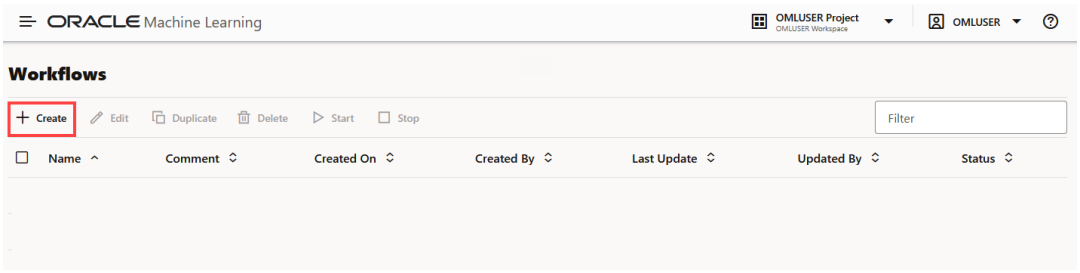
1. On the Oracle Machine Learning UI home page, click **Workflows**. Alternatively, open the left navigation menu, expand **Project** and then click **Workflows**. This opens the Workflows listing page.

Figure 6-5 Home page with Workflows options highlighted



2. On the Workflows listing page, click **Create**.

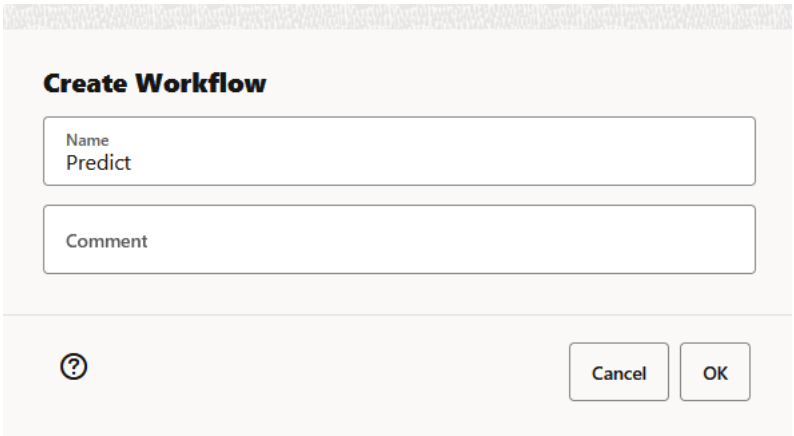
Figure 6-6 Workflows listing page



This opens the Create Workflow dialog.

3. In the Create Workflow dialog, enter these details:

Figure 6-7 Create Workflow dialog

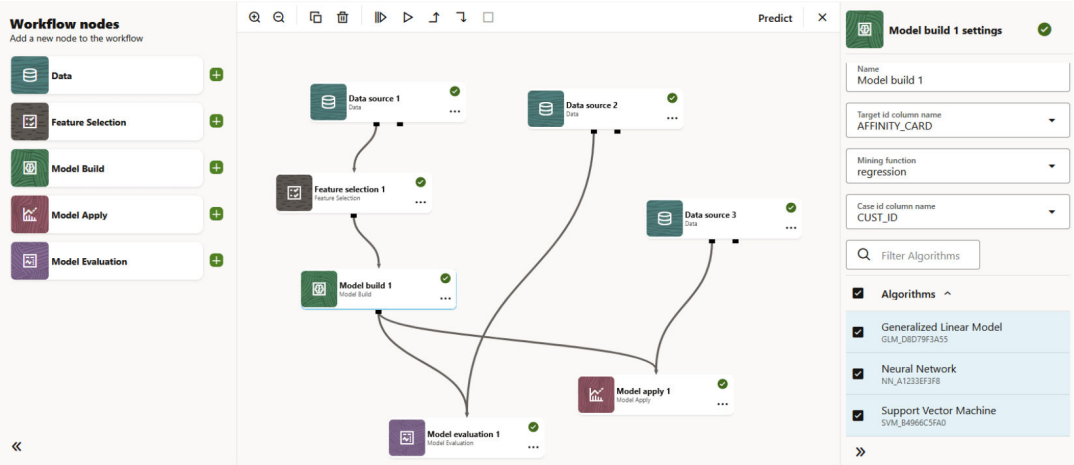


a. **Name:** Enter a name for the workflow. In this example, the name of the workflow is `Predict`.

b. **Comment:** Enter comments, if any.

c. Click **OK.** A blank workflow is created and it opens in the workflow editor. In the workflow editor, you can drag and drop the nodes from the **Workflow nodes** palette to the canvas, or simply click to add the nodes to the canvas and run it.

Figure 6-8 A complete workflow



On the Workflows editor:

- To the left is the **Workflow Nodes** palette. This is where all the workflow nodes are available.
- At the center is the canvas. This is where you create a workflow.
- To the right is the **Settings** pane. This pane opens specific settings for the nodes that you select. By default, this pane is collapsed. Click « to expand the pane. In the screenshot above, the Model apply node is selected, and the settings pane of this node is visible to the right.
- At the top is the canvas toolbar.

  - Click for a broader view of the canvas.
  - Click for a closer view of the canvas.
  - Click to run all the nodes in the workflow.

> **Note**
>
> To use this option, ensure that all the nodes in the workflow are defined correctly.

  - Click to run a selected node.

- Click to run the nodes with dependencies—run all parents nodes (upstream nodes in the workflow) along with selected nodes, if any.
- Click to run the node with dependents—run children nodes (downstream nodes) along with selected nodes, if any.
- Click to stop a workflow or a node that is running.
- Click to exit the canvas. This takes you back to the Workflows listing page.

6.3 Workflow Nodes

A workflow node is an individual computational component in a workflow. Each node represents a specific task or operation, such as importing and preparing data, training a model, or evaluating results.

Add nodes to the canvas by dragging them from the palette and dropping them onto the canvas, or by selecting a node from the palette to place it.

The available workflow nodes include:

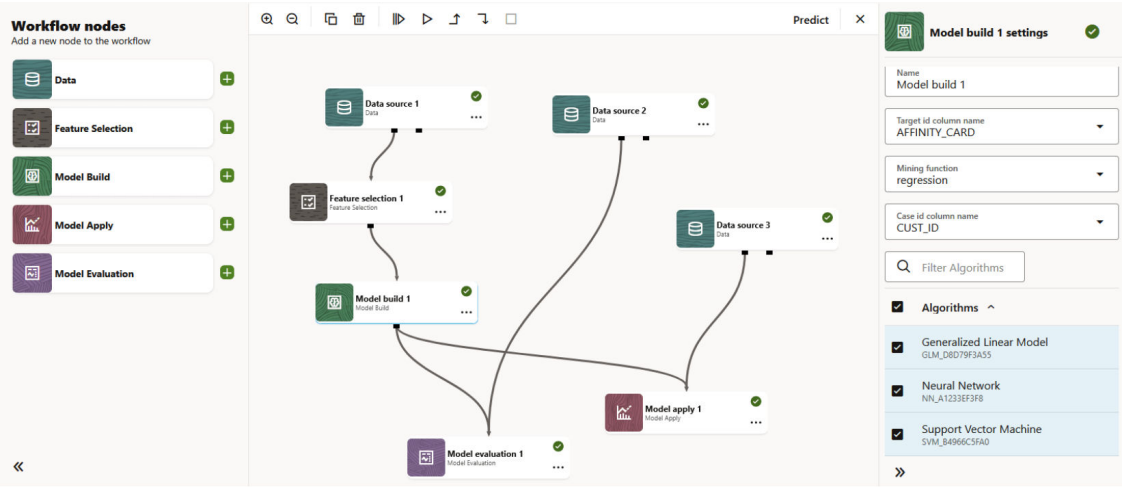
- **Data Source Node:** Specifies the workflow's data source (schema and table). This node supports data-related tasks such as data import, data splitting, and computing basic statistics. It typically serves as the starting point of the workflow.
- **Feature Selection Node:** Identifies and selects a subset of the most important features (columns) to help improve model performance. This node evaluates attribute importance during model training.
- **Model Build Node:** Trains machine learning or statistical models using supported algorithms. This node manages model training and model creation for downstream evaluation and scoring.
- **Model Evaluation Node:** Measures model performance by running evaluation tasks.
- **Model Apply Node:** Applies a trained model to input data to generate predictions or scores.

Here are the high-level steps to create a workflow:

1. Open an existing workflow, or click **Create** on the Workflow listing page.
2. Drag and drop the required workflow nodes from the palette onto the canvas.
3. Configure settings for each node as needed.
4. Run individual nodes or click **Run all** to run the entire workflow.

Below is a screenshot of a workflow containing all the available workflow nodes.

Figure 6-9 A complete workflow



- Data Source Node
  The Data source node defines the workflow's data source by specifying the schema and table. It supports data-related tasks such as data import, data splitting, and basic statistical computations. This node typically serves as the starting point of a workflow.
- Feature Selection Node
  The Feature Selection node selects a subset of relevant features (columns) to help improve model performance. It uses attribute-importance results generated during model training to guide feature selection.
- Model Build Node
  The Model Build node builds and trains machine learning or statistical models using supported algorithms.
- Model Evaluation Node
  The Model Evaluation node evaluates model performance using available evaluation metrics and reports.
- Model Apply Node
  The Model Apply node applies a trained model to input data to generate predictions or scores.
- Deploy a model
  Deploying a model creates an Oracle Machine Learning Services endpoint that can be used for scoring.

6.3.1 Data Source Node

The Data source node defines the workflow's data source by specifying the schema and table. It supports data-related tasks such as data import, data splitting, and basic statistical computations. This node typically serves as the starting point of a workflow.

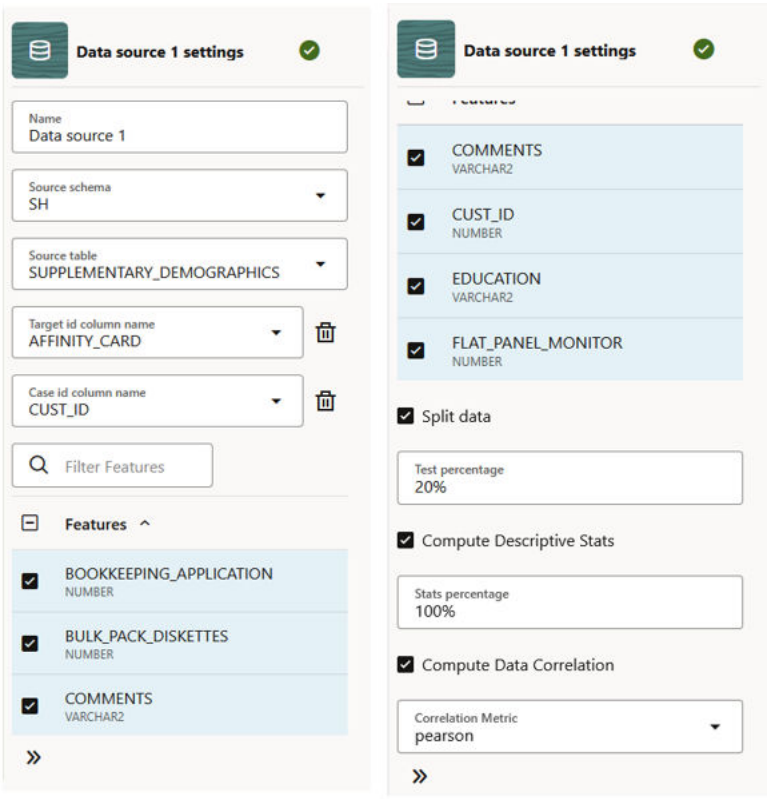
Allowed Connections:

- **Upstream node:** None
- **Downstream node:** Feature Selection node, Model Build node, Model Evaluation node

To create and run a Data Source node:

1. On the Workflows editor, drag and drop the Data Source node from the palette to the canvas. Alternatively, you may also click icon next to the Data Source node to add it to the canvas.
2. Adding a node automatically opens the settings pane on the right.

Figure 6-10 Settings - Data Source node



Here, define the data source:

a. **Name:** Enter a name. In this example, it is `Data source 1`.
b. **Source Schema:** Click on the drop-down menu to select the schema. Here, `SH` schema is selected.
c. **Source table:** Click on the drop-down menu to select the table. Here, `SUPPLEMENTARY_DEMOGRAPHICS` table is selected. Once you select the table, the **Features** field below displays the columns of the table.
d. **Target ID column name:** From the drop-down menu, select a target. This is the target for your prediction.
e. **Features:** This field displays all the columns of the selected table, and has all of them selected. You may deselect any column.
f. **Split Data:** By default, this option is selected and `20%` is allotted for test data. You can edit this field.
g. **Compute Descriptive Stats:** Select this option to allot a dataset percentage for descriptive statistics computation. The default is `100%`.
h. **Compute Data Correlation:** Select this option if you want to compute correlation between two variables in the dataset. Correlation is a bivariate analysis that measures

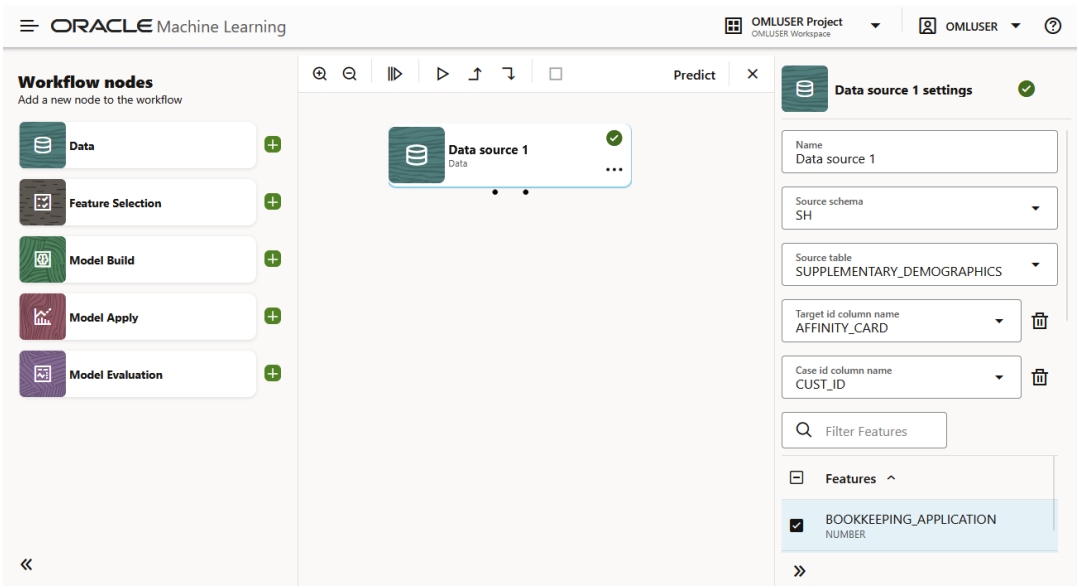
the strength of association between two variables and the direction of the relationship. The correlation metrics are:

- **Pearson correlation:** Measures the degree of the relationship (strength and direction) between two linearly related variables.
  - **Strength:** A value close to `1` or `-1` indicates a strong linear relationship, while a value near 0 suggests a weak or nonexistent one.
  - Direction can be positive or negative. **Positive Correlation**—Occurs when one variable increases, the other also tends to increase. **Negative Correlation**—Occurs when one variable increases, the other tends to decrease.
- **Kendall correlation:** A non-parametric test that measures the strength of dependence between two variables.
- **Spearman correlation:** A non-parametric test that measures the degree of association between two variables.

3. After defining the node, click on the ellipses on the node or right-click to open the menu. Click **Run**.

   Once the node runs successfully, it is indicated by the icon.

Figure 6-11 Data Source node

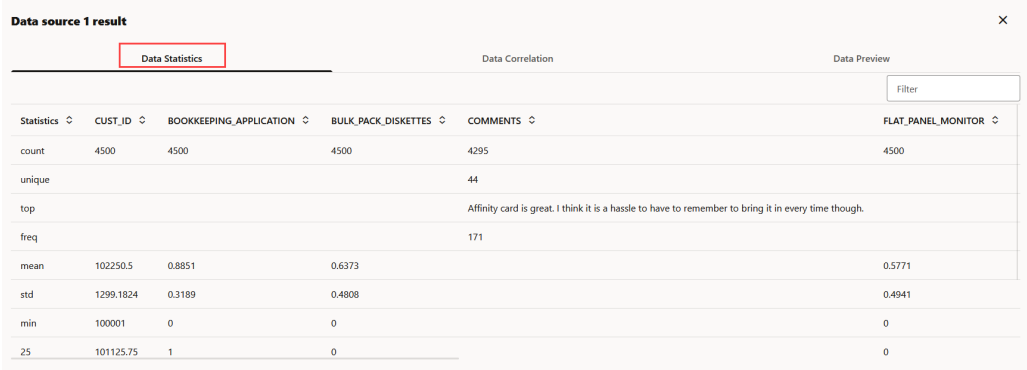


> **Note**
>
> You also have the option to add all the nodes to the workflow first, define the settings and connections for each node, and then run all the nodes by clicking **Run all.**

4. Click on the ellipses on the node or right click on the node and click **Results** to to view the results.
   - The **Data Statistics** tab displays the basic statistics computed for the selected features (columns). The computed statistics are Distinct Count, Number of Unique

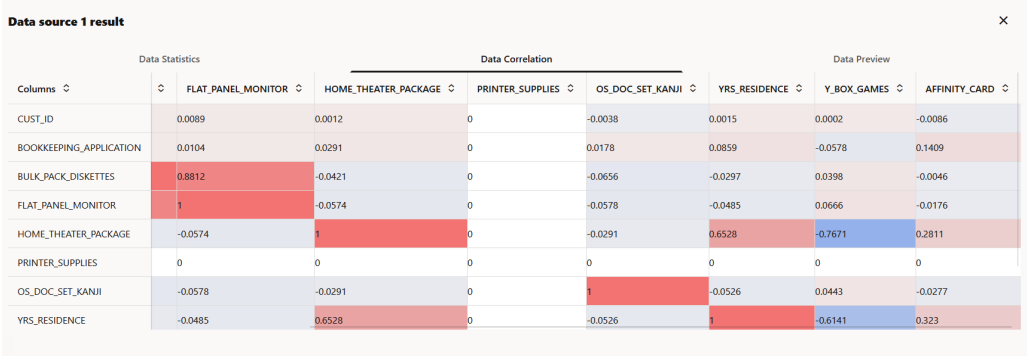
values, Top N, Frequent values, Mean, Std deviation, Min, 25, 50, 75 quantiles, and Max.

Figure 6-12 Data Statistics tab



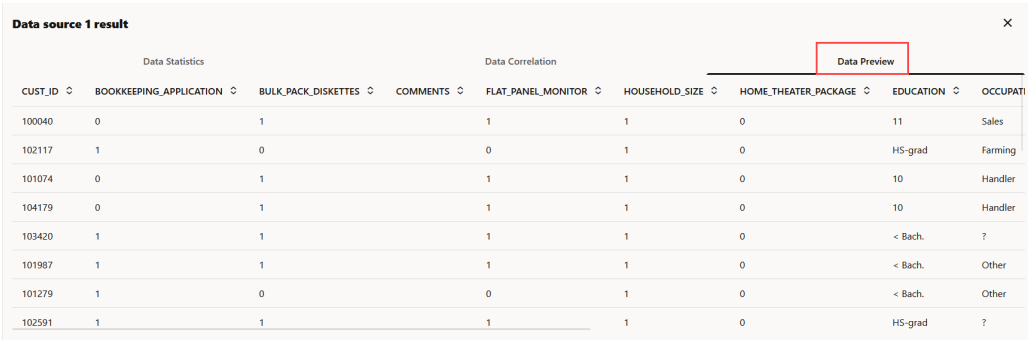
- The **Data Correlation** tab displays the correlation level between the columns. Positive values indicate positive correlation, and the different shades of red indicate the different levels of positive correlation. Negative values indicate negative correlation, and the different shades of blue indicate the levels of negative correlation. A value of 1 indicates perfect positive correlation and a value of -1 indicates perfect negative correlation. See screenshot below:

Figure 6-13 Data Correlation tab



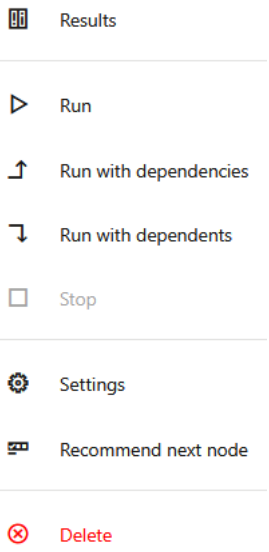
- The **Data Preview** tab shows the preview of the data set used for the workflow.

Figure 6-14 Data Preview tab



5. You may now proceed to add the next node in the flow. If you need suggestions, click the ellipses on the Data Source node or right-click to open the available options, and click **Recommend next node**. The applicable nodes are displayed. Click on the one that applies to add it to the canvas. Here you have two options:

Figure 6-15 Nodes - right-click options



- Add node manually: Drag and drop the node from the palette to the canvas. In this example, the **Feature Selection node** is added. You must now connect the Data Source node to the Feature Selection node. For this, click on the connection port indicated by the dot on the Upstream node. A connector appears. Drag the connector to the downstream node to establish the connection.
- Click **Recommend next node:** Click this option to view the list of recommended nodes.

6. Clicking **Recommend next node** gives you the following options under two categories.
   a. **Train Split**
      - Feature Selection
      - Model Build
   b. **Test Split**
      - Model Evaluation

   Click on the node that you want to add to the workflow. The connector is automatically added from the Data Source node to the selected node.

6.3.2 Feature Selection Node

The Feature Selection node selects a subset of relevant features (columns) to help improve model performance. It uses attribute-importance results generated during model training to guide feature selection.

Allowed Connections:

- **Upstream node:** Data Source node
- **Downstream nodes:** Feature Selection Node, Model Evaluation node, Model Apply node

To add and run a Feature Selection node:

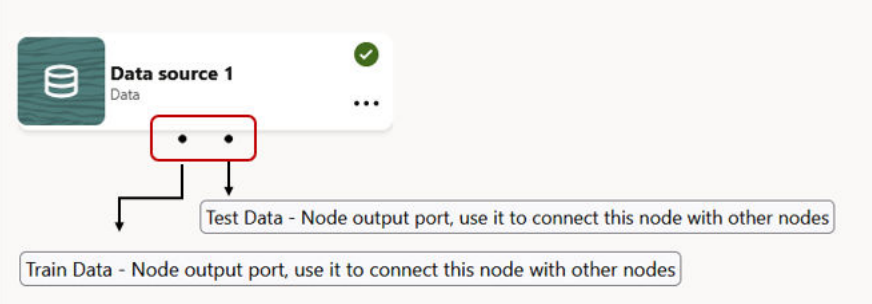
1. You can add the Feature Selection node to the workflow in two ways:
   - From the workflow palette, drag and drop the **Feature Selection node** to the canvas. Alternatively, you may also click on the node.
   - You may also click **Recommend next node** on the upstream node (Data Source node) and then click **Feature Selection**.

   > **Note**
   >
   > Using the **Recommend next node** option automatically connects the nodes.

2. If you manually add the node, you must establish a connection from the Data Source node (upstream node) to the Feature Selection node (downstream node). For this, click on the connection port indicated by the dot on the Upstream node. A connector appears. Drag the connector to the downstream node to establish the connection.

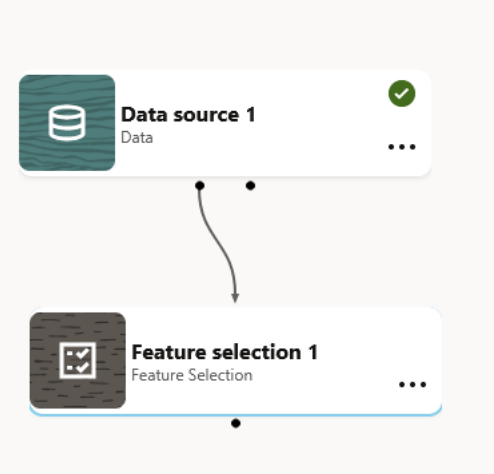
   The Data Source node has two connection ports. The one on the left is for Train dataset, and the one on the right is for Test dataset.

Figure 6-16 Connection ports on a Data Source node



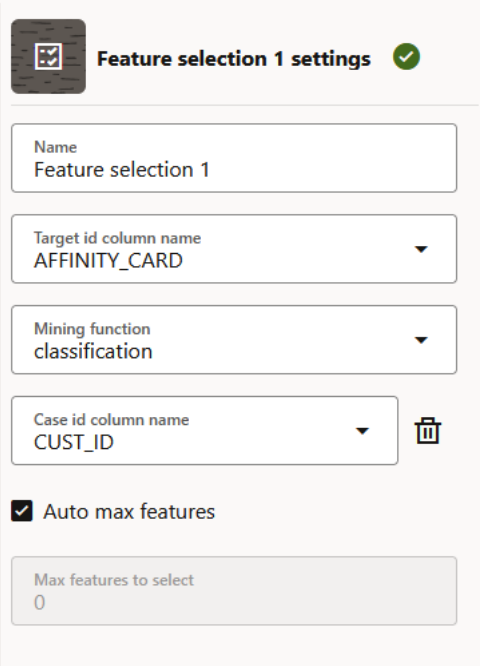
3. Once you connect the nodes, click on the **Feature Selection node** to open the settings. If the Settings pane is not visible, click « at the bottom to expand the pane.

Figure 6-17 Data Source and Feature Selection node connection



4. On the Settings pane, define the settings for the node.

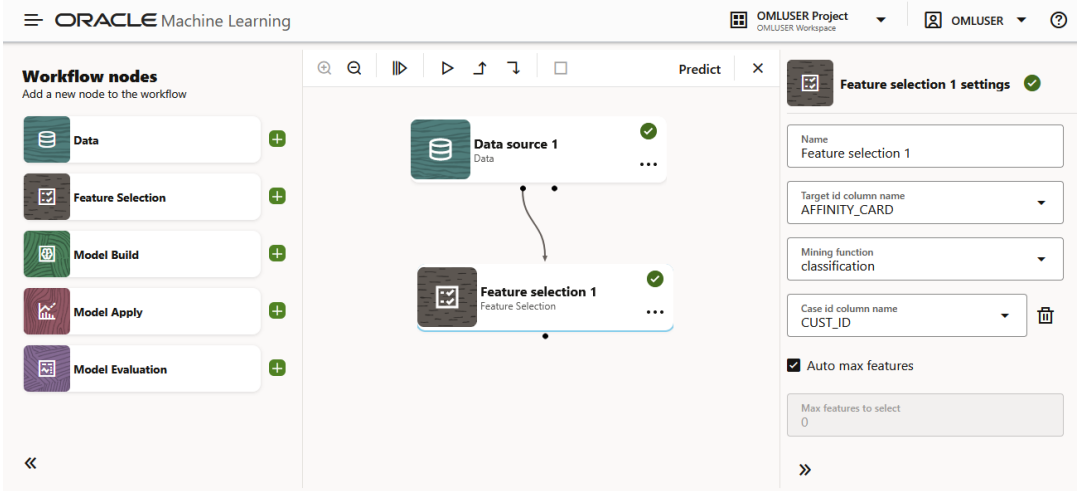
Figure 6-18 Settings - Feature Selection node



In this example, `AFFINITY_CARD` is selected as the **target id column name** and `CUST_ID` is the **Case id column**. The **Mining function** selected is `Classification`. The selection of mining function is automatic. It depends on the target you selected. However, you can override it.

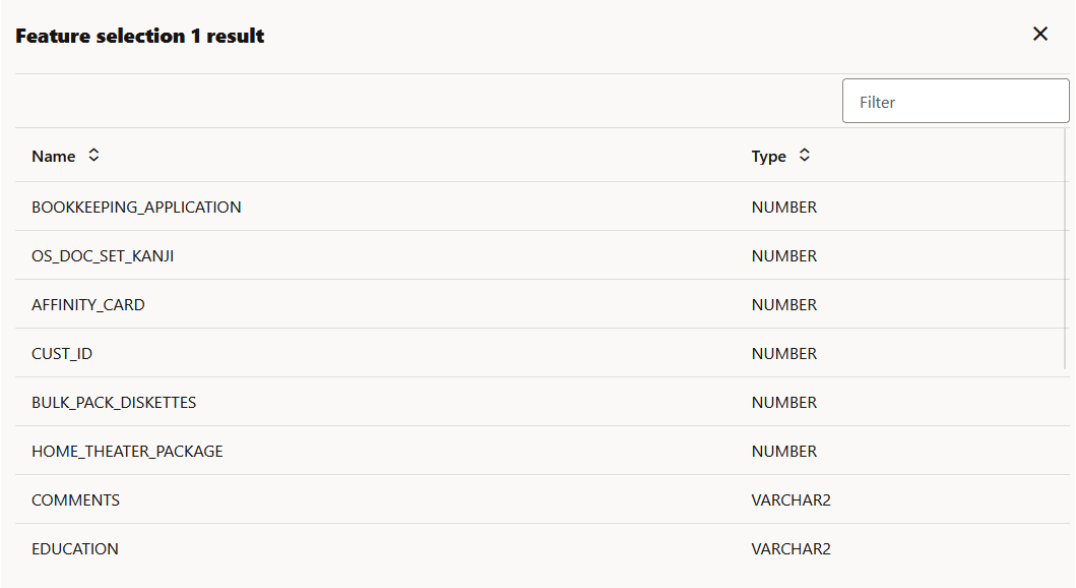
5. Right click on the node or click on the ellipses and then click **Run.** Once the node runs successfully, it is indicated by the icon, as shown in the screenshot below.

Figure 6-19 Feature Selection node



6. Once the node is run successfully, right-click on the node and then click **Results** to view the results.

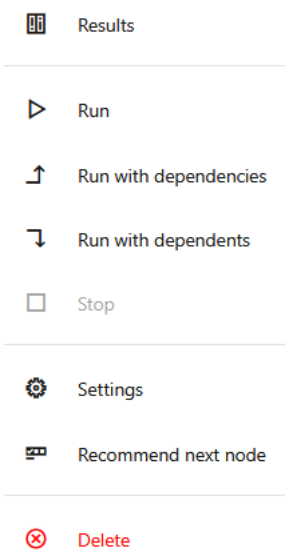
Figure 6-20 Feature Selection Results



7. You may now add the next node in the workflow. If you need suggestions, right-click on the node, and click **Recommend next node** to proceed with the workflow.

   The recommended next node is the **Model Build** node.

Figure 6-21 Right-click options



6.3.3 Model Build Node

The Model Build node builds and trains machine learning or statistical models using supported algorithms.

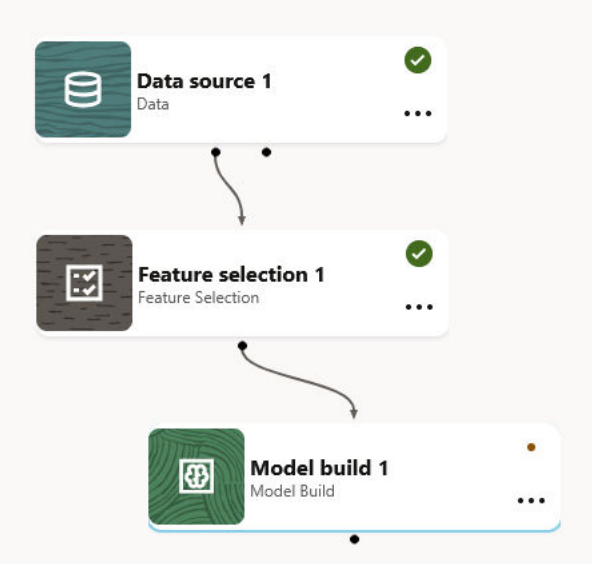
To add the Model Build node to your workflow, you must first have the Data Source node and Feature Selection node defined:
Allowed Connections:

- **Upstream nodes:** Data Source node, Feature Selection node
- **Downstream nodes:** Model Evaluation node, Model Apply node

To add and run a Model Build node:

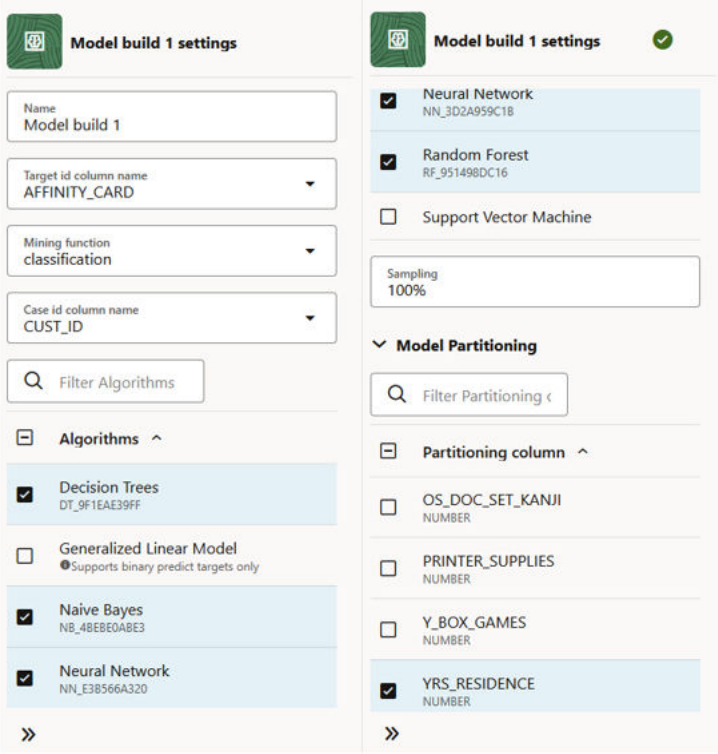
1. On the Workflow editor, drag and drop a Model Build node from the palette to the canvas.
2. From the Upstream node (Feature Selection node), establish a connection to the **Model Build node**. For this, click on the connection port indicated by the dot on the Upstream node. A connector appears. Drag the connector to the downstream node to establish the connection.

Figure 6-22 Model Build node in the workflow



3. Once you connect the nodes, define the settings. Click on the **Model Build node** to open the settings pane. If the Settings pane is not visible, click « on the bottom right to expand the pane.

Figure 6-23 Settings - Model Build node

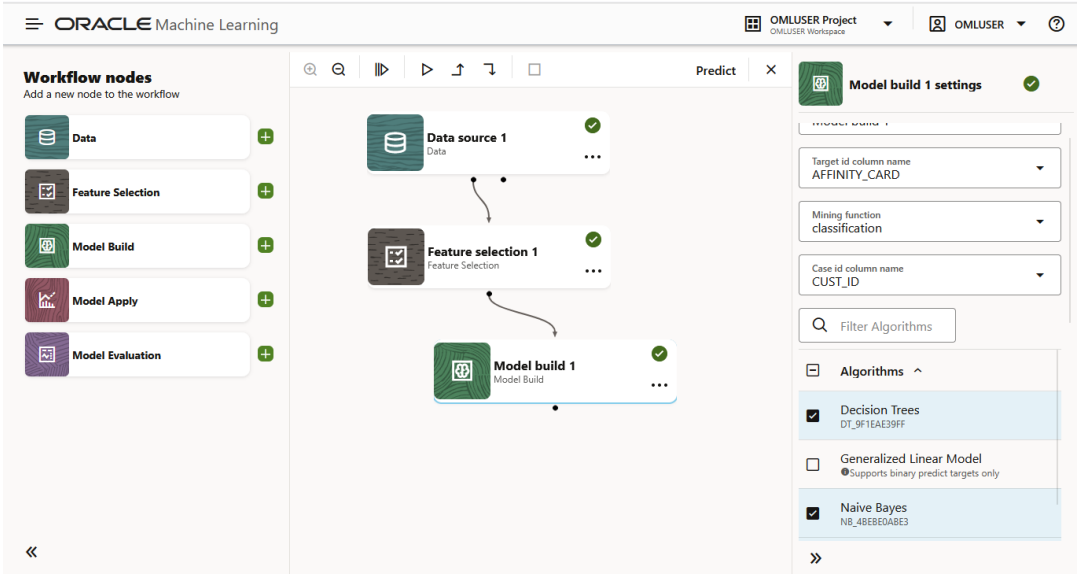


For the Model Build settings, the value for **Target ID column name** is set to `AFFINITY_CARD`, **Mining Function** is set to `Classification`, **Case ID** is `CUST_ID`,

Algorithms selected are Decision Tree, Naive Bayes, Neural Network, and Random Forest, and `YRS_RESIDENCE` is selected as the **Partitioning column**. The selection of mining function is automatic. It depends on the target you selected. However, you can override it.

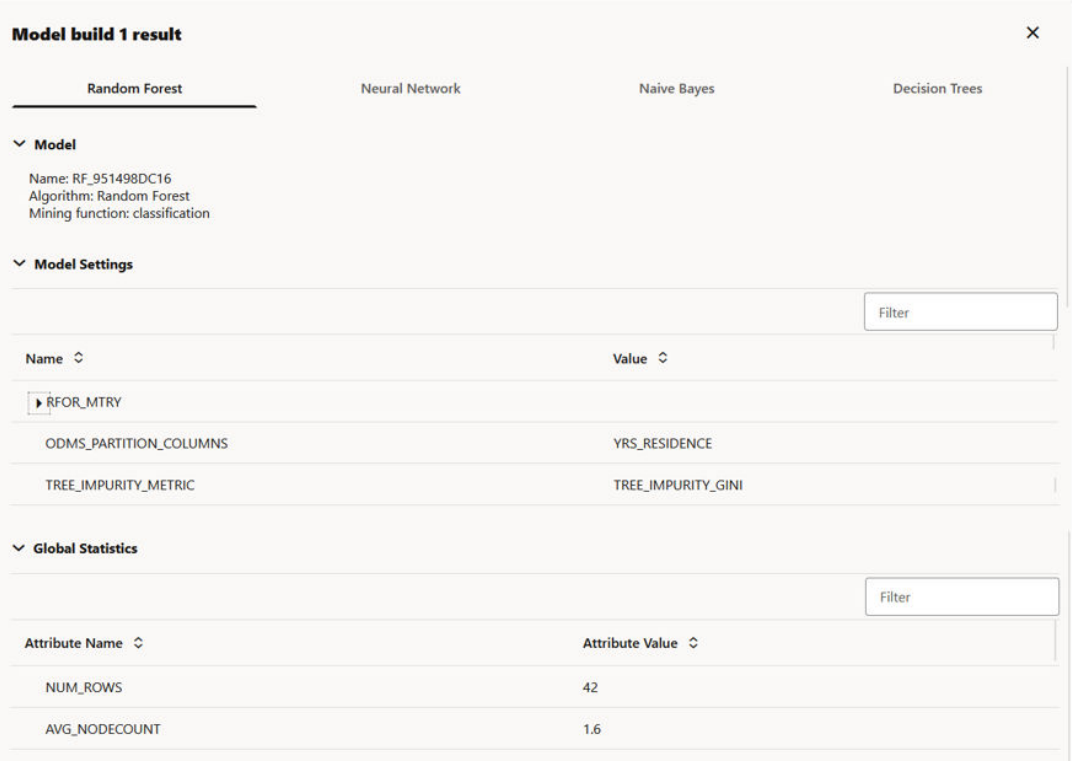
4. Right click on the node or click on the ellipses, and then click **Run.** Once the node runs successfully, it is indicated by the icon, as shown in the screenshot below.

Figure 6-24 Model Build node in the workflow



5. Once the node is run successfully, right-click on the node and then click **Results** to view the results. The results computed by the selected algorithms are displayed in respective tabs, as shown in the screenshot here.

Figure 6-25 Results - Model Build node



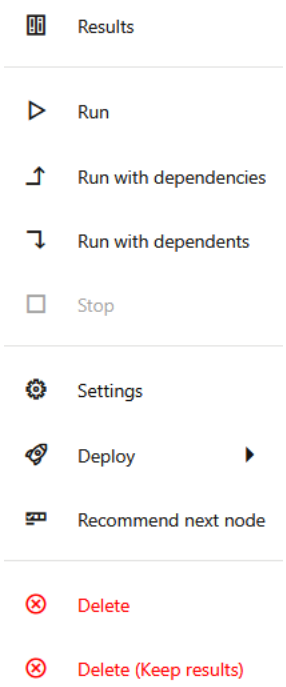
The results for each algorithm are displayed in separate tabs. In this example, the results are displayed in separate tabs by the name—**Random Forest**, **Neural Network**, **Naive Bayes**, and **Decision Tree**. These are the algorithm selected in the settings for this node. Each tab displays the **Model Details** (model name, algorithm, mining function), **Model Settings** for the specific algorithm, and **Global Statistics** containing the attributes and attribute values, as shown in the screenshot above.

> **Note**
>
> You cannot rename the underlying model. However, you can assign a name to a model deployed to OML Services.

6. You may now add the next node in the workflow. If you need suggestions, right-click on the node and click **Recommend next node** to proceed with the workflow.

   The recommended next nodes are **Model Evaluation node** and **Model Apply node**.

Figure 6-26 Model Build node right-click options



On the Model Build node, you also have the option to **Deploy** the model.

See Deploy a model for more information.

6.3.4 Model Evaluation Node

The Model Evaluation node evaluates model performance using available evaluation metrics and reports.

Allowed Connections:

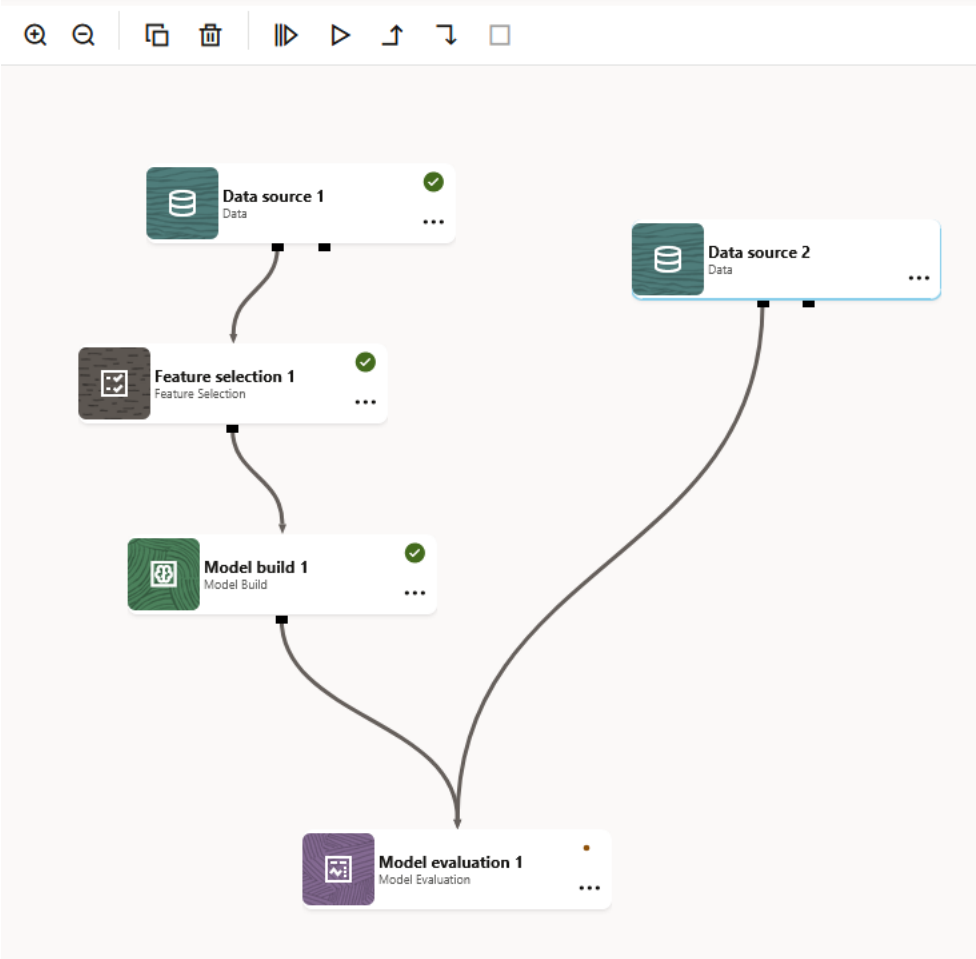
- **Upstream nodes:** Data Source node, Model Build Node
- **Downstream nodes:** None

To add and run a Model Evaluation node:

1. In the Workflow editor, drag and drop a **Model Evaluation node** from the palette to the canvas. Alternatively, you may also click on the node.
2. From the Model Build node, establish a connection to the Model Evaluation node. If you are adding it manually, establish a connection from the Upstream node (Model Build node) to the Downstream node (Model Evaluation node). For this, click on the connection port indicated by the dot on the Upstream node. A connector appears. Drag the connector to the downstream node to establish the connection.
3. Drag and drop another Data Source node and establish a connection to the Model Evaluation node. Define the second Data Source node with the same settings.

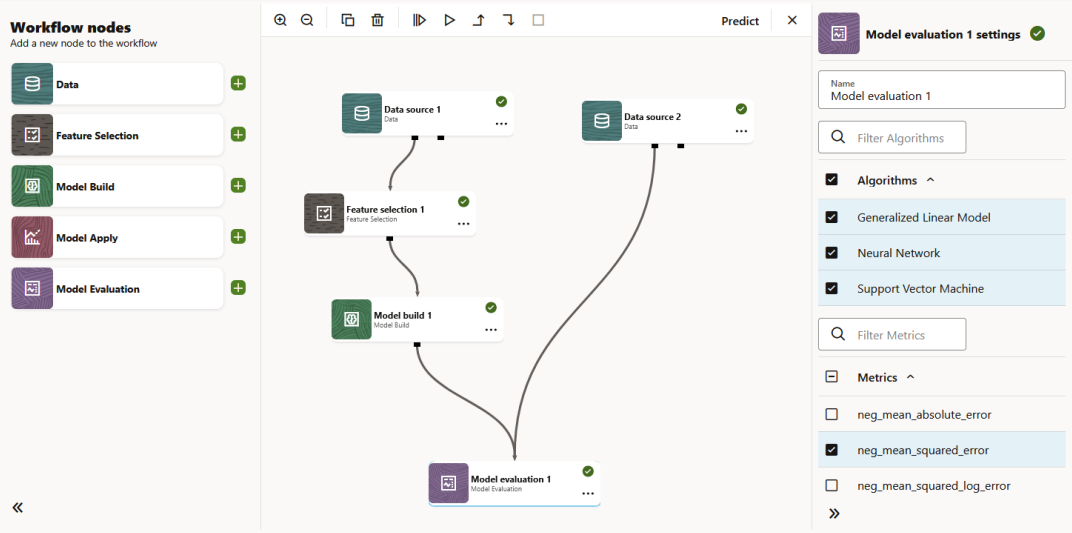
> **Note**
>
> The second nub on the `Data source 1` node can be a test data set for model evaluation.

Figure 6-27 Model Evaluation node



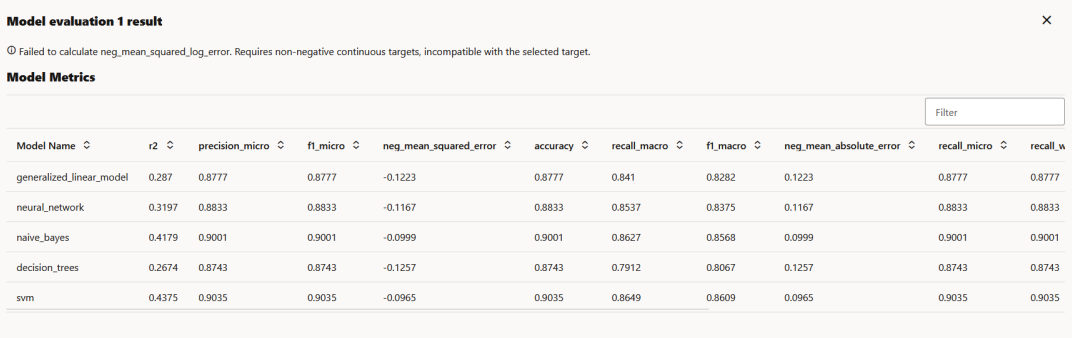
4. Once you connect the nodes, define the settings. Click on the node to open the settings pane. If the Settings pane is not visible, click « at the bottom to expand the pane.
5. Right click on the node or click on the ellipses and then click **Run.** Once the node runs successfully, it is indicated by the ✓ icon, as shown in the screenshot below.

Figure 6-28 Model Evaluation node in the workflow



6. Once the node is run successfully, right-click on the node and then click **Results** to view the results.

Figure 6-29 Results - Model Evaluation node



> **Note**
>
> The Model Evaluation node is a terminal node. You cannot add any downstream nodes to it.

6.3.5 Model Apply Node

The Model Apply node applies a trained model to input data to generate predictions or scores.

Allowed Connections:

- **Upstream node:** Data Source node, Model Build node.
- **Downstream node:** None.

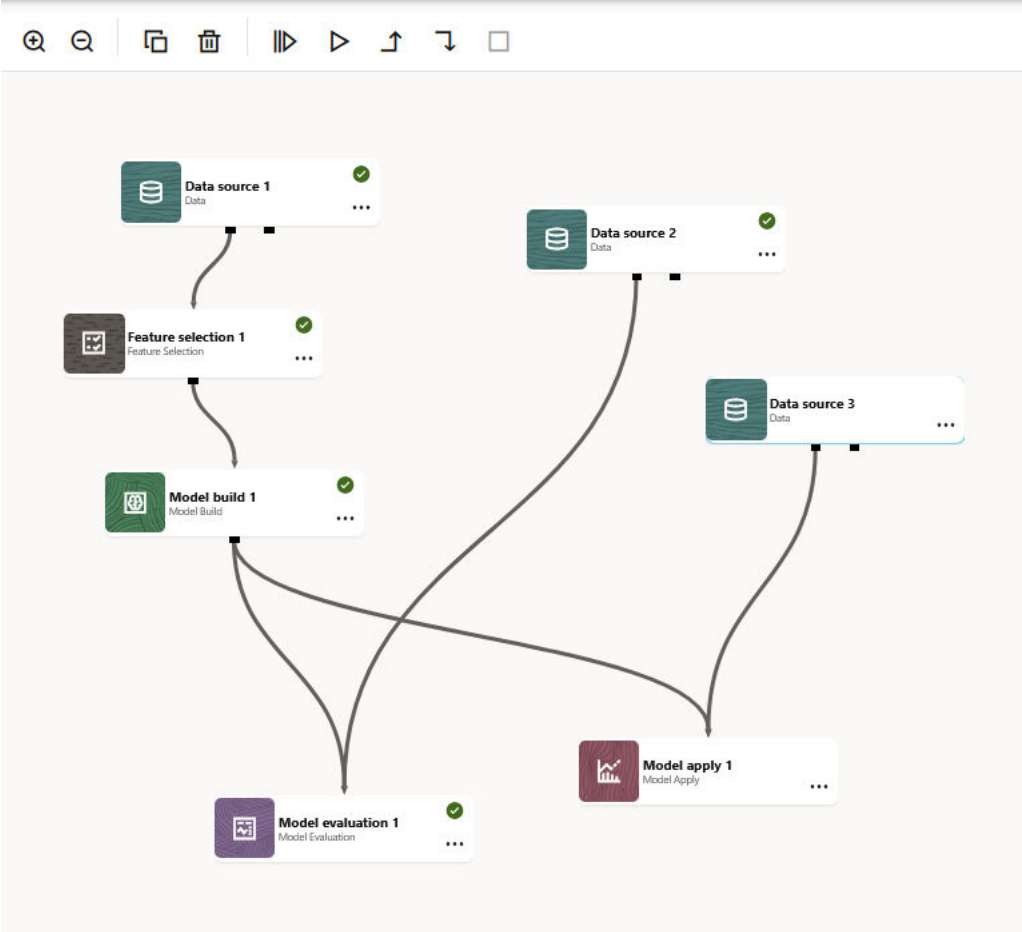
To add and run a model apply node:

1. From the workflow palette, drag and drop the Model Apply node to the canvas. Alternatively, you may also click on the node on the palette.

2. From the Model Build node, establish a connection to the Model Apply node. If you are adding it manually, establish a connection from the Upstream node (Model Build node) to the Downstream node (Model Apply node). For this, click on the connection port indicated by the dot on the source node to the child node.
3. Drag and drop another Data Source node to the palette. Establish a connection from the Upstream node (Data Source node) to the Downstream node (Model Apply node).

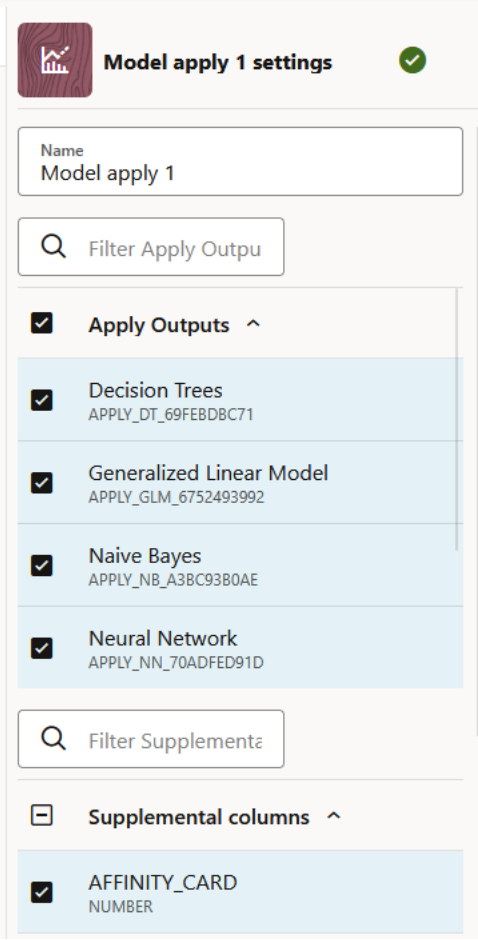
   You may add additional Data Source nodes for test and train datasets, as shown in the screenshot here.

Figure 6-30 Model Apply node



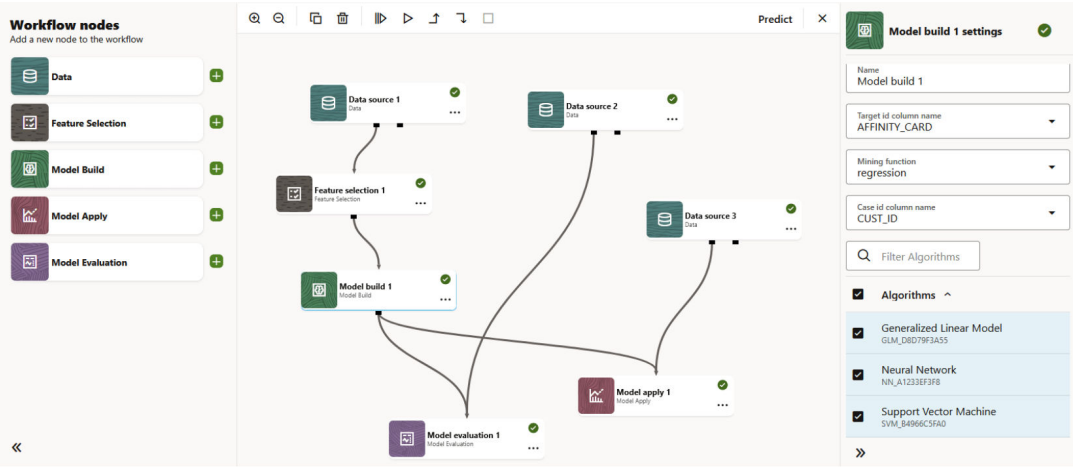
4. Once you connect the nodes, define the settings. Click on the node to open the settings pane. If the Settings pane is not visible, click « on the bottom right to expand the pane.

Figure 6-31 Settings - Model Apply node



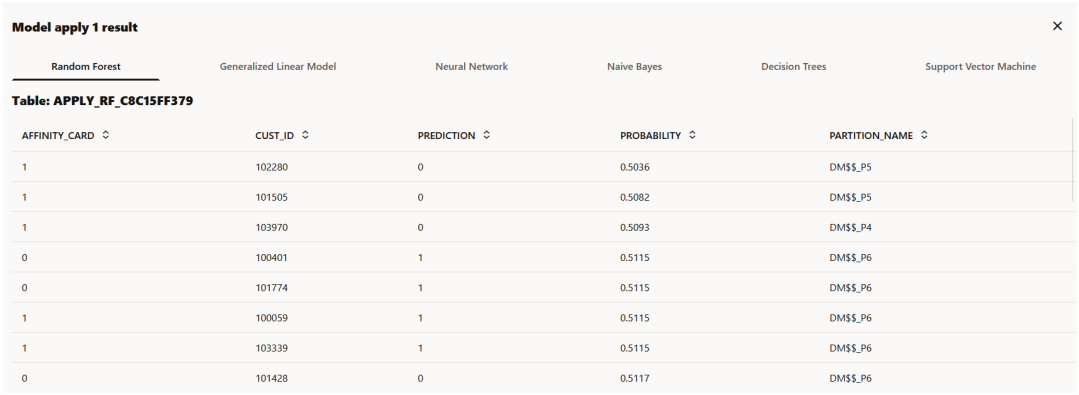
5. Right click on the node or click on the ellipses and then click **Run.** Once the node runs successfully, it is indicated by the icon, as shown in the screenshot below.

Figure 6-32 Model Apply node in the workflow



6. Once the node is run successfully, right-click on the node and then click **Results** to view the results.

Figure 6-33 Results - Model Apply node



> **Note**
>
> The Model Apply node is a terminal node. You cannot add any downstream nodes to it.

6.3.6 Deploy a model

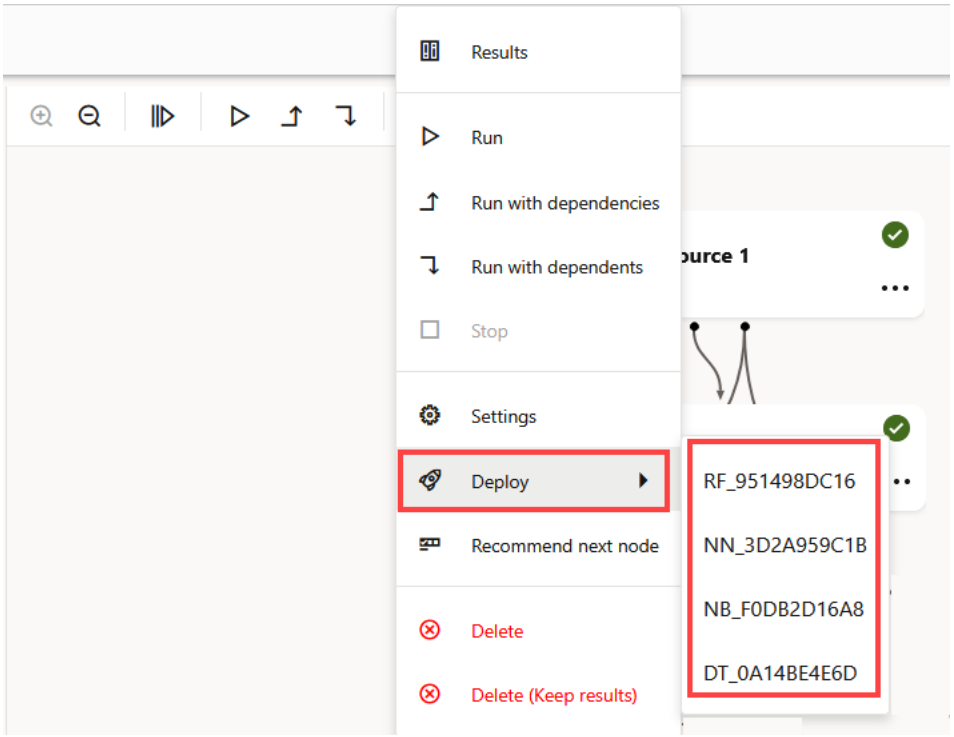
Deploying a model creates an Oracle Machine Learning Services endpoint that can be used for scoring.

You can deploy a model directly from the **Model Build node** in the Oracle Machine Learning workflow.

To deploy a model:

1. Open the workflow and right-click on the **Model Build node**. You may also click on the ellipses on the node. This opens the right-click menu.
2. Click **Deploy.** The models are listed.

Figure 6-34 Deploy option in Model Build node



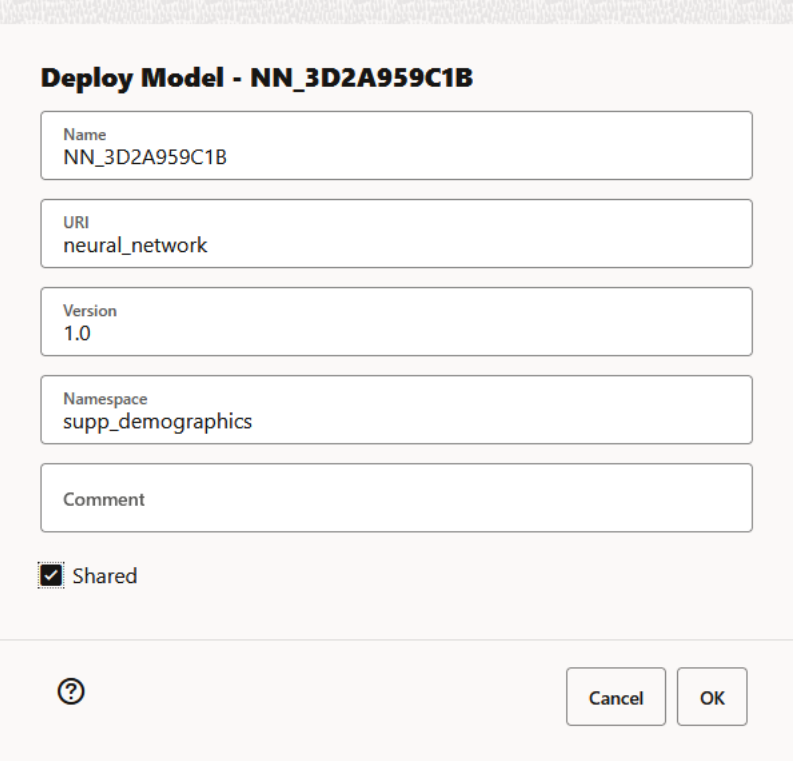
3. Click on the model you want to deploy. The Deploy Model dialog opens.

> **Note**
>
> The model name begins with a prefix made up of the first two letters of the algorithm name. For example, the model name RF_951498DC16 indicates that the model was built using the Random Forest algorithm. Similarly, NN indicates Neural Network, NB indicates Naive Bayes, DT indicates Decision Tree and so on.

Once you've built an in-database model using Oracle Machine Learning Workflow, it is ready to be used in SQL queries. As such it is already "deployed." You can grant access to this model to other schemas, export this model and import it into another database for deployment, and deploy this model to OML Services, which is enabled directly through the Oracle Machine Learning Workflow UI. OML Services deployment supports lightweight, real-time scoring using REST, among other features.

4. In the Deploy Model dialog, define the following:

Figure 6-35 Deploy Model dialog



a. In the **Name** field, the system-generated model name is displayed here by default. This is the name the model for use in OML Services. The in-database model name remains unchanged. You can edit this name, which must be a unique alphanumeric name with maximum 50 characters.

b. In the **URI** field, enter a name for the model URI. The URI must be alphanumeric, and the length must be max 200 characters. In the example here, the `neural_network` is the name provided for URI.

c. In the **Version** field, enter a version of the model. The version must be in the format `x.x` where `x` is a number.

d. In the `Namespace` field, enter a name for the model namespace. In the example here, `supp_demographics` is the name provided for Namespace. This is an optional field.

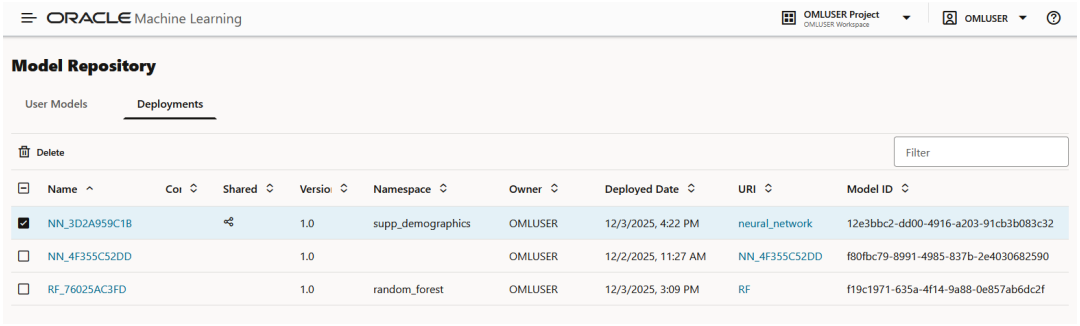
e. In the **Comments** field, you may enter any comments.

f. Click **Shared** to allow users with access to the database schema to view and deploy the model.

g. Click **OK.**

5. After a model is successfully deployed, it is listed in the **Deployments** tab on the **Models** page.

Figure 6-36 Deployed models in the Deployments tab



In the screenshot here, the model `NN_3D2A959C1B` deployed in steps 4 is listed on the Deployments tab.

Related Topics

- Models

7

Data Science Agent

This section discusses how to get started with Data Science Agent in Oracle Machine Learning UI.

Topics:

- Example of a Conversation with Data Science Agent
  This example demonstrates how Data Science Agent supports a data analyst in exploring a dataset, and in building and evaluating a machine learning model. It also shows the effectiveness of Natural Language to SQL (NL2SQL) in Data Science Agent.
- Known Issues in Data Science Agent
  Learn about the issues you may encounter when using Data Science Agent in Oracle Machine Learning UI on Oracle Autonomous AI Database and their possible workarounds.

7.1 Data Science Agent

Data Science Agent is an intelligent built-in conversational chatbot integrated with Oracle Machine Learning UI included in your Oracle Autonomous AI Database subscription. You must provide the LLM, whether from a third-party AI provider, OCI GenAI Service, or one you privately host. You can run many common data science workflows, including discovery, exploration, preparation, model training, evaluation, and supported scoring workflows, using natural language.

As a human-in-the-loop chatbot, Data Science Agent requires your intervention and approval in the following areas:

- **Object association:** Associating database objects with a conversation improves precision and efficiency. Objects must be associated or approved before the agent creates views that use them; other supported operations may still use accessible objects based on the user's database privileges. See Add Objects to Data Science Agent Conversation.
- **Conversation management:** The scope of Data Science Agent is limited to the active conversation context only. You must create, manage, and delete your conversations. See Manage Data Science Conversations.
- **Profile selection:** You must select an AI profile, and therefore, the LLM that powers the agent. You can also switch profiles mid-conversation. See Manage AI Profile and Service Level of Data Science Conversation.
- **Approval for object view:** During any conversation, Data Science Agent compares the `object_list` with the objects already associated with the active conversation. If required, the agent proposes any table or view outside the user-approved set. It stops the conversation and explicitly seeks your approval to associate those missing objects before continuing. If you approve associating the missing objects, the association is recorded and the agent continues the conversation. Else, the operation is canceled and no view is created.

7.2 Data Science Agent Concepts

Here is a list of key concepts and terms commonly used in Data Science Agent.

AI Credential

An AI credential stores authentication details that the database uses to access the selected AI provider or related cloud resources. Depending on the provider, it may contain an API key, OCI signing key details, or other provider-specific authentication fields. The credential comprises the following information:

- `user_ocid`: This is the unique identifier of the OCI user.
- `tenancy_ocid`: This is the unique identifier of the OCI tenancy in your cloud account.
- `private_key`: The private key associated with the OCI user. It is required for secure authentication.

- `fingerprint`: The fingerprint of the public key linked to the OCI user.

AI Profile

An AI Profile is a named configuration object that specifies how the database connects to an LLM — including the provider, (for example, openai, oci), credential, model, and optional parameters such as temperature and max_tokens and so on.

For more information, see Manage AI Profiles.

Conversation

The interaction with Data Science Agent takes the form of a conversation, each made up of alternating `turns`. Each `turn` begins with a user prompt, followed by the agent's response. The conversation retains the context throughout, allowing you to refer back to previous answers in later questions. For example, you might ask, "filter the dataset you just profiled," or "train a model using the training dataset we prepared".

Conversation history

Conversation history is a persistent record of past conversations with the agent. It allows you to browse through conversation history, review previous results, and continue past sessions without losing the context. This ensures continuity over time, allows multiple workloads in separate chats, supports reproducibility of analyses, and provides an auditable trail on how insights were derived.

Conversation Objects Catalog

Data Science Agent operates on and produces three types of database objects while handling requests. If you associate these objects to your conversation, the agent can inspect, analyze, transform, and model from those objects directly. This will thereby enhance the quality of the agent's response. If you do not associate any object, the agent will automatically scan the database for relevant objects based on your prompt.

The objects are available in the **Conversation Object Catalog**:

- Tables: Source data and persisted modeling results (created by the agent)
- Views: Views may be pre-existing data sources or derived datasets created by the agent. They are used for analysis, modeling, or general data transformation. Views created by the agent use the prefix `DSAGENT$` and may include a unique suffix.
- Mining models: The Oracle Machine Learning (OML) models trained by the agent.

For more information on how to associate these objects to your conversation, see Add Objects to Data Science Agent Conversation.

Prompt

A prompt is your input or message that initiates an interaction. It can be a question, command, statement, or request that Data Science Agent processes in order to generate an appropriate response. Essentially, the prompt guides the agent on what information or action you are seeking.

Prompt library

A prompt library is a curated set of system, task, and tool-specific prompts that defines how the agent interprets your prompts, interprets results, and calls various tools. The prompts are designed to encode domain knowledge and ensure consistent, reliable behavior.

Service Levels

In Oracle Machine Learning (OML) on Autonomous AI Database, service levels refer to the predefined configurations for resource allocation and workload management. Essentially, it determines how many ECPUs, and memory are allocated to a session. There are three types of service levels - Low, Medium, and High.

These service levels help manage and prioritize workloads running on the database, ensuring appropriate performance based on the use case.

For more information how to change the service levels of your conversation, see Use Data Science Agent Chat Interface.

Tools

Tools are modular components that enable the agent to perform specific tasks such as profiling a data object, computing feature correlations, or training a model. Each tool has clearly defined inputs, outputs, and constraints. In short, tools serve as the building blocks of Data Science Agent's functionality. Although end users do not interact with these tools directly, it determines the user experience of the agent.

7.3 Key Highlights of Data Science Agent

Data Science Agent offers a range of powerful features designed to streamline data science workflows. The key features include:

- **Data Discovery and Inspection:** Accesses and discovers data locally as well as from supported accessible database objects.
- **Exploratory Statistical Analysis:** Conducts single-variable analysis as well as relationship analysis. Relationship analysis is performed pairwise, that is, between two variables such as predictors and outcomes. This means each predictor is examined individually against one outcome. Data Science Agent can scan many predictors against a single outcome; however, this process does not replace multivariate modeling.

  > **Note**
  >
  > Relationship analyses are most reliable when performed on row-level (fine-grained) datasets, rather than on heavily aggregated data.

- **View-based Data Preparation:** Transforms and prepares data for modeling by creating new views. This is how it joins tables, filters populations, and derives new features from existing attributes.
- **Data Analysis and Visualization:** Simplifies and automates data analysis with built-in visualization for actionable insights.
- **Feature Selection and Feature engineering:** Profiles datasets, and performs feature selection and feature engineering.
- **Model Training (supervised and unsupervised) including Automated Model Search:** Handles training for both supervised and unsupervised models, thereby providing clear explanations of metrics and results to support learning and decision-making. It supports Classification, Regression, Clustering, and Anomaly Detection. Supported algorithms include XGBoost, Random Forest, Decision Tree, Neural Network, Naive Bayes, SVM, GLM, K-Means, Expectation Maximization, and O-Cluster. Converse with the agent to:

  - Train models to predict a categorical outcome (Classification) or a numeric value (Regression)
  - Evaluate multiple supervised algorithms and pick the best model based on a metric (automated model search), and
  - Build models without a labeled target (Clustering and Anomaly Detection)
- **Model Comparison and Evaluation:** Handles model comparison and evaluation. If you have multiple models, either created by the agent or otherwise, you can request the agent for a comparison based on a common validation dataset.
- **Inference (scoring) on new data:** Performs inference (scoring) on new data. Inference requires a trained model, a dataset containing the full feature set expected by the model, and a dataset containing the IDs to score.

> **Note**
>
> Inference is supported only in ID-based scoring mode, that is IDs to score along with full feature dataset. Broader scoring options will be available soon.

- **Reuse of existing objects:** Data Science Agent may reuse existing objects—views or models, although starting from scratch is also an option. If you prefer that the agent doesn't reuse previous objects, you can state so in your response. Otherwise, the agent may refer to or reuse relevant objects created earlier—including those from other conversations—when they are manually associated or automatically discovered. This is done to save time by avoiding repeated creation of the same objects.

7.4 Best Practices for using Data Science Agent

Follow these best practices to maximize the benefits of Data Science Agent.

Topics:

7.4.1 Recommended Models

Data Science Agent works with large language models accessed through Oracle `DBMS_CLOUD_AI` and `DBMS_CLOUD_AI_AGENT` packages. The `DBMS_CLOUD_AI` package, with Select AI, supports the translation of natural language prompts to generate, run, explain SQL statements, and also enables RAG and natural language-based interactions, including chats with LLMs. For more information, see DBMS_CLOUD_AI Package and DBMS_CLOUD_AI_AGENT Package.

This table lists the recommended large language models and the scenarios in which each should be used.

> **Note**
>
> Recommended models may change as providers update model availability, latency, pricing, and quality. Check the current list of supported models for your AI provider before creating a profile.

Table 7-1 Recommended Models

Provider	Tier	Large Language Model
OpenAI or OCI GenAI	Top	`gpt-5.5`
OpenAI or OCI GenAI	Cost-effective	`gpt-5.4-mini`
OCI GenAI	Top	`xai.grok-4.3`

> **Note**
>
> When using OCI Generative AI, use the provider and model identifiers exactly as documented for OCI GenAI. Some model identifiers may include the original model family or vendor name.

Table 7-1 (Cont.) Recommended Models

Provider	Tier	Large Language Model
OCI GenAI	Cost-effective	`xai.grok-4-1-fast-reasoning`
Google	Top	`gemini-3.5-flash`
Google	Cost-effective	`gemini-3-flash-preview`
Anthropic	Top	`claude-opus-4-8`
Anthropic	Cost-effective	`claude-sonnet-4-6`

> **Note**
>
> When using OCI Generative AI, use the provider and model identifiers exactly as documented for OCI GenAI. Some model identifiers may include the original model family or vendor name.

Explanation of tiers:

- **Top:** Represents the best state-of-the-art model from a specific provider. This tier is the strongest option in terms of quality, reliability, and precision.
- **Cost-effective:** Represents a good compromise between quality, cost, and speed. These models are typically faster and less expensive. But the trade-off is lower quality and reliability compared to the Top tier.

Profile Creation for GPT-5.5

To create an AI profile for GPT-5.5 through OpenAI, run the following script in a notebook:

```
DECLARE
    profile_name VARCHAR2(128) := 'OPENAI_GPT_5_5';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "credential_name": "OPENAI_CRED",
            "model": "gpt-5.5",
```

```
            "provider": "openai",
            "temperature": 1,
            "max_tokens": 8192
        }'
    );
END;
/
```

To create an AI profile for GPT 5.5 through Oracle Cloud Infrastructure (OCI), run the following script in a notebook:

```
DECLARE
    profile_name VARCHAR2(128) := 'OCI_GPT_5_5';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "credential_name": "OCI_CRED",
            "model": "openai.gpt-5.5",
            "provider": "oci",
            "temperature": 1,
            "max_tokens": 8192,
            "oci_compartment_id": "<your-dep-id>"
          }'
        );
    END;
/
```

Profile Creation for Grok 4.3

To create an AI profile for Grok 4.3 through Oracle Cloud Infrastructure (OCI), run the following script in a notebook:

```
DECLARE
    profile_name VARCHAR2(128) := 'OCI_GROK_4_3';
BEGIN
        dbms_cloud_ai.drop_profile(profile_name, TRUE);
        dbms_cloud_ai.create_profile(
            profile_name => profile_name,
            attributes => '{
                "credential_name": "OCI_CRED",
                "model": "xai.grok-4.3",
                "provider": "oci",
                "temperature": 1,
                "max_tokens": 8192,
                "oci_compartment_id": "<your-dep-id>"
               }'
            );
    END;
/
```

Parameters:

- `profile_name`: Name of the AI profile. It must follow Oracle SQL identifier naming rules.

- `credential_name`: Name of the credential used to authenticate with the selected AI provider.
- `model`: Model identifier used by the selected provider.
- `provider`: AI provider for the profile, for example `openai`, `oci`, `google`, or `anthropic`.
- `temperature`: Recommended value for Data Science Agent examples: `1`.
- `max_tokens`: Recommended value for these examples: `8192`.

> **Note**
>
> Data Science Agent profile inspection treats values below `4096` as not recommended.

- `oci_compartment_id`: Required for the OCI GenAI examples. Use the target compartment OCID or documented deployment/compartment identifier.

For more information, see Manage AI Profiles.

7.4.2 Associate Database objects to your conversation

Consider associating database objects such as tables, views, and mining models to a Data Science Agent conversation. Once you associate these objects, the agent can inspect, analyze, transform, and model from those objects directly. This enhances the quality of the agent's response. If you do not associate any object, the agent will automatically scan the database for relevant objects based on your query.

> **Note**
>
> Some operations such as feature ranking, model search, training can be compute-intensive and may take time.

7.4.3 Ask for clarification

During the course of your conversation, you can ask for clarification at any time. Some examples:

- What was done in the previous step
- Why was a particular step necessary
- What is the next recommended step
- Explain the <concept>. For example, what is unstructured data in machine learning?

7.4.4 Ask in multiple iterations

If you are using Data Science Agent for extended workflows, consider asking the agent in multiple iterations. Longer workflows are generally more effective when handled iteratively. For instance, you can start by creating a dataset view, then move to validating assumptions, and finally focus on model training and evaluation.

7.4.5 Clarify terminology

If you use specific terms in your conversation, it is a good practice to clarify those terms to the agent.

7.4.6 Follow suggestions provided by Data Science Agent

Follow the suggestion of the agent when appropriate. The agent frequently proposes the next steps of a workflow. For example, data preparation, analysis, model training. Accept or refine these suggestions for a smooth progress.

7.4.7 Limit your conversation length and scope

Although Data Science Agent can handle extended interactions, very long conversations may gather context that negatively affects clarity or performance. For extended work, consider starting a new conversation, especially if you encounter these situations:

- If your conversation has a lot of messages (around 50 messages), or
- If your objective changes

7.4.8 Provide context to your conversation

The interaction with Data Science Agent is structured as a conversation, consisting of alternating `turns`. A `turn` begins with your prompt, followed by the agent's response. A Data Science Agent conversation maintains the context across `turns`.

Therefore, providing context to your conversation is a good practice, especially if you resume a conversation at a later time.

7.4.9 Specify a clear objective

Clearly state your objective at the beginning of the conversation. For example, "I want to predict customer churn" or "I want to identify the main causes". Sharing a high-level intent early in the conversation helps guide the rest of the conversation. When the agent understands your objective, it can suggest the most appropriate workflow.

7.5 Data Science Agent: Sample Prompts and Outputs

Here are some sample prompts and outputs related to various machine learning domains on which you may have conversations with Data Science Agent.

1. Data Discovery

Discovery is semantic and goal-driven. It works best and more efficiently when the goal and domain are stated explicitly.

You can ask the agent to find data objects relevant to a business topic or analysis goal. For example, marketing response, churn, fraud, product demand. It can also help you obtain a general overview of all available objects.

Example 7-1 Discover available tables, views, and models

Sample prompts

- Find tables and views related to bank marketing subscriptions and campaign contacts.
- What data exists related to bank marketing?
- Find tables related to customer churn and retention?
- What objects are available?

Outputs

Here are some expected outputs for the above prompts:

- A curated set of relevant objects—tables, views, models.
- Business-oriented summaries and hints about how objects relate. For example, likely join keys.
- Additional extended report with detailed information about all relevant objects.

> **Note**
>
> - Best results depend on meaningful metadata. Semantically clear tables, views, column names and well-maintained annotations improve quality and relevance of results.
> - You can manually associate database objects—tables, views, or models to the conversation so the agent can use them immediately. This is useful when the relevant objects are already known and discovery is unnecessary. Once associated, the agent can inspect, analyze, transform, and model from those objects directly. Discovery can remain optional unless additional data needs to be found.

2. Inspect Specific Object

You can ask for details about a specific table, view, or mining model.

Example 7-2 Ask questions related to specific tables, views and mining models

Sample prompts

- Describe the `CUSTOMERS` table?
- Show the columns and types in `SCHEMA.SALES_TRANSACTIONS`?
- What attributes are used in the model `CHURN_MODEL`?

Outputs

Here are some expected outputs for the above prompts:

- For tables and views, the agent will typically retrieve information related to row and column counts, column list and data types, a small data sample.
- For models, the agent will typically retrieve information related to features, target and algorithm data.

3. Exploratory Statistical Analysis

For exploratory statistical analysis, you can ask the agent for both single-variable analysis and relationship analysis (pair).

Example 7-3 Single-variable analysis

You can request distribution and qualitative summaries for one or more individual columns.

Sample prompts

- Describe the `SALES.CUSTOMERS` table and provide an overview of all its attributes.
- Provide an overview of all variables in `SCHEMA.CUSTOMERS_VIEW`
- Analyze the distribution of `AGE`, `INCOME`, and `JOB_CATEGORY`.
- Analyze which factors are most associated with subscription behavior.

Outputs

Here are some expected outputs for the above prompts:

- Global interpretation of analysis results.
- Distribution summaries for each variable, using statistics and plots appropriate to the variable type, that is, numeric versus categorical.
- Percentage of missing values and number of categories, as applicable.

Example 7-4 Relationship analysis

You can request statistical analysis of the relationship between two variables, such as predictors and outcomes. Relationship analysis is performed pairwise. This means each predictor is examined individually against one outcome. Data Science Agent can scan many predictors against a single outcome; however, this process does not replace multivariate modeling.

> **Note**
>
> Relationship analyses are most reliable when performed on row-level (fine-grained) datasets, rather than on heavily aggregated data.

Sample prompts

- What factors are most associated with subscriptions?
- How does `CONTACT_CHANNEL` relate to `AGE`?
- Analyze relationships of all features versus `CHURN_FLAG`.

Outputs

Here are some expected outputs for the above prompts:

- Global interpretation of pairwise analysis results.
- Pairwise relationship summaries for each attribute (against target variable), using statistics and plots appropriate to the variable types (numeric vs. categorical).

4. View-Based Data Transformation and Preparation

The agent can transform and prepare data for modeling by creating new views. This is how it joins tables, filters populations, and derives new features from existing attributes. Here are some common view-building tasks:

- Join customer, transaction, and interaction tables into a unified dataset.
- Filter to a time window or segment. For example, last 12 months, specific product line and so on.
- Create derived fields. For example, date components such as year/month/day or day of week.
- Exclude unsupported or non relevant fields from training datasets when needed.
- Create a new view joining clients, contacts, and past campaigns; extract day and month from timestamps

The agent does not run arbitrary adhoc SQL queries and return full result sets for interactive browsing. Views are the primary mechanism for shaping data.

The agent does not directly modify base tables.

Example 7-5 View-Based Data Transformation and Preparation

Sample prompts

- Join `CLIENTS`, `CONTACTS`, and `PAST_CAMPAIGNS` into a modeling dataset.
- Make the dataset ready for modeling by extracting features from timestamps.

Outputs

Here are some expected outputs for the above prompts:

- A new view in the user schema starting with the prefix `DSAGENT$`
- A plain-language summary of what the view contains and how it was created
- SQL code used to create the view.
- Visual diagram to track dependencies and operations at a glance.

5. Feature Importance and Feature Selection

Ask which variables matter most for predicting a specific target and optionally reduce the dataset to most important features.

Example 7-6 Feature Importance and Feature Selection

Sample Prompts

- Rank feature importance for predicting `SUBSCRIBED`.
- Create a reduced dataset with only important features

Outputs

Here are some expected outputs for the above prompts:

- A ranked list of attributes with importance scores
- Optionally, a new top-features view created from the original dataset

> **Note**
>
> Feature importance can be computed using different supported algorithms. The agent can guide you on algorithm choice in business terms.

6. Dataset Splitting for Training and Evaluation

Use the agent to split dataset as database views.

Example 7-7 Dataset Splitting for Training and Evaluation

Sample Prompts

- Split into train/validation/test using standard percentages.
- Create an 80/20 train/test split.
- Split data into train, validation and test sets, then find best model optimizing Accuracy.

Outputs

Here are some expected outputs for the above prompts:

- New views with suffixes such as `_TRAIN`, `_VAL` (if requested), `_TEST`
- Optional `_UNLABELED` view if a target column is provided and some rows have `NULL` targets. You can use this view later for inference.
- SQL code used to perform the split.

7. Model Training

Use Data Science Agent for model training, automated model selection (supervised model search), and model building (unsupervised learning).

Example 7-8 Supervised learning (classification and regression)

Here are some prompts to use Data Science Agent to train models to predict a categorical outcome (classification) or a numeric value (regression).

Sample Prompts

- Train a classifier to predict `SUBSCRIBED`.
- Train a regression model to predict `CALL_DURATION`.

Outputs

Here are some expected outputs for the above prompts:

- A trained OML mining model stored in the database
- A summary of the training run and configuration choices
- SQL code to replicate the training

Example 7-9 Automated model selection (supervised model search)

You can ask the agent to evaluate multiple supervised algorithms and pick the best model based on a metric.

> **Note**
>
> Automated model selection requires a validation set for comparing models against the selected metric.

After automated model selection, the winning model is retrained on combined train and validation dataset. Therefore, the final model is not the same object as the one that scored best during comparison.

Sample Prompts

- Find the best model for predicting SUBSCRIBED using F1 metric.
- Run an automated model search for churn prediction.

Outputs

Here are some expected outputs for the above prompts:

- A best-performing model selected using the chosen metric (on validation data).
- A report of validation performance across tested algorithms.
- Optionally, a results table containing the benchmark metrics.

Example 7-10 Unsupervised learning (Clustering and Anomaly Detection)

You can use Data Science Agent to build models without a labeled target.

> **Note**
>
> For unsupervised models, only model build is supported currently. Additional scoring capabilities and interpretations for clustering will be available soon.

Sample Prompts

- Segment customers into clusters.
- Detect anomalies in transaction behavior.

Outputs

Here are some expected outputs for the above prompts:

- A trained clustering or anomaly model stored in the database
- A summary describing how to use the model for downstream scoring

8. Compare Models and Select a Winner

When multiple models exist, either created by the agent or otherwise, you can request the agent for a comparison based on a common validation dataset.

Example 7-11 Model comparison

Sample Prompts

- Compare these three models using AUC and select the best.
- Rank candidate models and store results in a table.

Outputs

Here are some expected outputs for the above prompts:

- A ranked comparison (best-to-worst) on the specified metric.
- Optional persistence of the full ranking into a results table for auditability.

9. Evaluate Models

Use Data Science Agent for an unbiased evaluation on held-out test data.

> **Note**
>
> Evaluation on held-out test set is intended as the most reliable estimate of generalization performance of a trained model.

Example 7-12 Model Evaluation

Sample Prompts

- Evaluate the selected model on the test set.
- Provide Precision, Recall, F1 and a confusion matrix on test.
- Evaluate regression error on test.
- Evaluate the best model on the test set, then score the prospects dataset and return the highest-probability cases

Outputs

Here are some expected outputs for the above prompts:

- For Classification: accuracy-family metrics and confusion-matrix reporting (binary and multiclass supported)
- For Regression: fit and error metrics. For example, R^2, MAE, RMSE.
- A test-results table stored in the database.
- SQL code to use the models in inference on arbitrary data.

10. Inference and Scoring

Once a model exists, you can request scoring on new records. Inference requires:

- A trained model
- A dataset containing the full feature set expected by the model
- A dataset containing the IDs to score

> **Note**
>
> Inference is supported only in ID-based scoring mode, that is IDs to score along with full feature dataset. Broader scoring options will be available soon.

Example 7-13 Inference and Scoring

Sample Prompts

- Score the prospects table and return the top 500 most likely to subscribe.

- Run inference for these customer IDs.

Outputs

Here are some expected outputs for the above prompts:

- Predictions returned in the UI, linked back to case IDs
- For Classification: predicted class and probability (based on the designated positive class)
- For Regression: predicted numeric value

You can also use the agent in interactive mode for suggestions and interpretations as well.

Examples:

- Suggestion request: "I want to predict clients most likely to subscribe, assist me in designing a suitable workflow"
- Interpretation request: "Can you help me interpret the model metrics so that I can better assess its performance?"

7.6 Prerequisites to use Data Science Agent

To use Data Science Agent, you must have the following:

Table 7-2 Prerequisites to use Data Science Agent

Requirement	Needed for	Who performs	Where to configure
AI credential	LLM access	User/Admin	`DBMS_CLOUD` For more information, see Use DBMS_CLOUD_AI to Configure AI Profiles.
AI Profile	Select AI/Data Science Agent Select AI must be configured to use supported AI providers. For more information, see Perform Prerequisites for Select AI	User/Admin	`DBMS_CLOUD_AI` For more information, see Use DBMS_CLOUD_AI to Configure AI Profiles.

Table 7-2 (Cont.) Prerequisites to use Data Science Agent

Requirement	Needed for	Who performs	Where to configure
OML_DEVELOPER	OML UI Access	Admin	SQL grants

Note

If the user (OML_USER) is created through Database Acti

Table 7-2 (Cont.) Prerequisites to use Data Science Agent

Requirement	Needed for	Who performs	Where to configure
			ons, the OML_DEVELOPER role is automatically granted.

Table 7-2 (Cont.) Prerequisites to use Data Science Agent

Requirement	Needed for	Who performs	Where to configure
Host ACL User must be added to the host ACL (Access Control List).	Third-party providers only	Admin	`DBMS_NETWORK_ACL_ADMIN`
Note This is not required for OCI GenerativeAI.			
Object privileges	Data access	Admin/Object owner	SQL grants

For additional information, see Manage AI Profiles

- Use DBMS_CLOUD_AI to Configure AI Profiles
  Autonomous Database uses AI profiles to facilitate and configure access to an LLM and to enable generating, running, and explaining SQL based on natural language prompts. It also facilitates retrieval augmented generation using embedding models and vector indexes and allows for chatting with the LLM.
- Grant OML_DEVELOPER Role to OML User
  To use Data Science Agent, the administrator must grant the `OML_DEVELOPER` role to the OML user.
- Add User to the Host ACL
  For model providers like OpenAI, add users to the host ACL (Access Control List).
- Perform Prerequisites for Select AI
  Before you use Select AI, here are the steps to enable `DBMS_CLOUD_AI`.

7.6.1 Use DBMS_CLOUD_AI to Configure AI Profiles

Autonomous Database uses AI profiles to facilitate and configure access to an LLM and to enable generating, running, and explaining SQL based on natural language prompts. It also facilitates retrieval augmented generation using embedding models and vector indexes and allows for chatting with the LLM.

AI profiles include database objects that are the target for natural language queries. Metadata used from these targets can include database table names, column names, column data types, and comments. You create and configure AI profiles using the following procedures:

- DBMS_CLOUD_AI.CREATE_PROFILE
- DBMS_CLOUD_AI.SET_PROFILE

7.6.2 Grant `OML_DEVELOPER` Role to OML User

To use Data Science Agent, the administrator must grant the `OML_DEVELOPER` role to the OML user.

If the OML user (`OMLUSER`) is created through Database Actions, the `OML_DEVELOPER` role is automatically granted.

To grant the `OML_DEVELOPER` role, run the following:

```
GRANT OML_DEVELOPER to OMLUSER
```

7.6.3 Add User to the Host ACL

For model providers like OpenAI, add users to the host ACL (Access Control List).

> **Note**
>
> Host ACL entry is not required for OCI GenAI.

The following procedure grants the privilege to use the *api.openai.com* endpoint.

> **Note**
>
> This procedure is not applicable to OCI Generative AI.

```
BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
         host => 'api.openai.com',
         ace  => xs$ace_type(privilege_list => xs$name_list('http'),
                             principal_name => 'OMLUSER',
                             principal_type => xs_acl.ptype_db)
   );
END;
```

The parameters are:

- `host`: The host, which can be the name or the IP address of the host. You can use a wildcard to specify a domain or an IP subnet. The host or domain name is not case sensitive.

AI Provider	Host
OpenAI	*api.openai.com*
OpenAI-compatible providers	For example, for Fireworks AI, use *api.fireworks.ai*
Cohere	*api.cohere.ai*
Azure OpenAI Service	*<azure_resource_name>.openai.azure.com* See Profile Attributes to know more about `azure_resource_name`.
Google	*generativelanguage.googleapis.com*
Anthropic	*api.anthropic.com*
Hugging Face	*api-inference.huggingface.co*
AWS	*bedrock-runtime.us-east-1.amazonaws.com*

- `ace`: The access control entries (ACE). The `XS$ACE_TYPE` type is provided to construct each ACE entry for the ACL. For more details, see Creating ACLs and ACEs .

7.6.4 Perform Prerequisites for Select AI

Before you use Select AI, here are the steps to enable `DBMS_CLOUD_AI`.

The following are required to use `DBMS_CLOUD_AI`:

- Access to an Oracle Cloud Infrastructure cloud account and to an Autonomous Database instance.
- A paid API account of a supported AI provider, one of:

AI Provider	API Keys
OpenAI	See Use OpenAI to get your API keys.
OpenAI-compatible providers	See Use OpenAI-Compatible Providers to get your API keys and `provider_endpoint`.

AI Provider	API Keys
Cohere	See Use Cohere to get your secret API keys.
Azure OpenAI Service	See Use Azure OpenAI Service for more information on how to configure Azure OpenAI Service.
OCI Generative AI	See Use OCI Generative AI .
Google	See Use Google to get your API keys.
Anthropic	See Use Anthropic to get your API keys.
Hugging Face	See Use Hugging Face to get your API keys.
AWS	See Use AWS to get your API keys and model ID.

- Network ACL privileges to access your external AI provider.

  > **Note**
  >
  > Network ACL privileges are not required for OCI Generative AI.

- A credential that provides access to the AI provider.
- Grant Privileges for Select AI
  To use Select AI, the administrator must grant the `EXECUTE` privilege on the `DBMS_CLOUD_AI` package. Learn about additional privileges required for Select AI and its features.
- Examples of Privileges to Run Select AI
  Review examples of privileges required to use Select AI and its features.

7.6.4.1 Grant Privileges for Select AI

To use Select AI, the administrator must grant the `EXECUTE` privilege on the `DBMS_CLOUD_AI` package. Learn about additional privileges required for Select AI and its features.

To configure `DBMS_CLOUD_AI`:

1. Grant the `EXECUTE` privilege on the `DBMS_CLOUD_AI` package to the user who wants to use Select AI.

   By default, only the system administrator has `EXECUTE` privilege. The administrator can grant `EXECUTE` privilege to other users.

2. Grant `EXECUTE` privilege on `DBMS_CLOUD_PIPELINE` to the user who wants to use Select AI with RAG.

   > **Note**
   >
   > If the user already has the `DWROLE` role, this privilege is included and additional grant is not required.

3. Grant network ACL access to the user who wants to use Select AI and for the AI provider endpoint.

   The system administrator can grant network ACL access. See APPEND_HOST_ACE Procedure for more information.

4. Create a credential to enable access to your AI provider.

See CREATE_CREDENTIAL Procedure for more information.

5. Grant quotas in tablespace to manage the amount of space in a specific tablespace to the user who wants to use Select AI with RAG.

7.6.4.2 Examples of Privileges to Run Select AI

Review examples of privileges required to use Select AI and its features.

The following example grants the `EXECUTE` privilege to `ADB_USER`:

```
GRANT execute on DBMS_CLOUD_AI to ADB_USER;
```

The following example grants `EXECUTE` privilege for the `DBMS_CLOUD_PIPELINE` package required for RAG:

```
GRANT EXECUTE on DBMS_CLOUD_PIPELINE to ADB_USER;
```

To check the privileges granted to a user for the `DBMS_CLOUD_AI` and `DBMS_CLOUD_PIPELINE` packages, an administrator can run the following:

```
SELECT table_name AS package_name, privilege
 FROM DBA_TAB_PRIVS
 WHERE grantee = '<username>'
 AND   (table_name = 'DBMS_CLOUD_PIPELINE'
        OR table_name = 'DBMS_CLOUD_AI');
```

The following example grants `ADB_USER` the privilege to use the *api.openai.com* endpoint.

> **Note**
>
> This procedure is not applicable to OCI Generative AI.

```
BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
         host => 'api.openai.com',
         ace  => xs$ace_type(privilege_list => xs$name_list('http'),
                             principal_name => 'ADB_USER',
                             principal_type => xs_acl.ptype_db)
   );
END;
/
```

The parameters are:

- `host`: The host, which can be the name or the IP address of the host. You can use a wildcard to specify a domain or an IP subnet. The host or domain name is not case sensitive.

AI Provider	Host
OpenAI	*api.openai.com*

AI Provider	Host
OpenAI-compatible providers	For example, for Fireworks AI, use *api.fireworks.ai*
Cohere	*api.cohere.ai*
Azure OpenAI Service	*<azure_resource_name>.openai.azure.com* See Profile Attributes to know more about `azure_resource_name`.
Google	*generativelanguage.googleapis.com*
Anthropic	*api.anthropic.com*
Hugging Face	*api-inference.huggingface.co*
AWS	*bedrock-runtime.us-east-1.amazonaws.com*

- `ace`: The access control entries (ACE). The `XS$ACE_TYPE` type is provided to construct each ACE entry for the ACL. For more details, see Creating ACLs and ACEs.

The following example creates a credential to enable access to OpenAI.

```
EXEC
DBMS_CLOUD.CREATE_CREDENTIAL(
credential_name   => 'OPENAI_CRED',
username          =>  'OPENAI',
password          =>  '<your_api_token>');
```

The parameters are:

- `credential_name`: The name of the credential to be stored. The `credential_name` parameter must conform to Oracle object naming conventions.
- `username`: The `username` and `password` arguments together specify your AI provider credentials.

  The `username` is a user-specified user name.
- `password`: The `username` and `password` arguments together specify your AI provider credentials.

  The password is your AI provider secret API key, and depends on the provider, that is, OpenAI, Cohere, or Azure OpenAI Service.

AI Provider	API Keys
OpenAI	See Use OpenAI to get your API keys.
OpenAI-compatible providers	See Use OpenAI Compatible Providers to get your API keys and `provider_endpoint`.
Cohere	See Use Cohere to get your API keys.

AI Provider	API Keys
Azure OpenAI Service	See Use Azure OpenAI Service to get your API keys and to configure the service. **Note** If you are using the Azure OpenAI Service principal to authenticate, you can skip the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure. See Example: Select AI Actions for an example of authenticating using Azure OpenAI Service principal.
OCI Generative AI	See Use OCI Generative AI to generate API signing keys.
Google	See Use Google to generate your API keys.
Anthropic	See Use Anthropic to generate your API keys.
Hugging Face	See Use Hugging Face to generate your API keys.
AWS	See Use AWS to get your API keys and model ID.

The following example grants quotas on tablespace to the `ADB_USER` to use Select AI with RAG:

```
ALTER USER ADB_USER QUOTA 1T ON <tablespace_name>;
```

To the check the tablespace quota granted to a user, run the following:

```
SELECT TABLESPACE_NAME, BYTES, MAX_BYTES
FROM DBA_TS_QUOTAS
WHERE USERNAME = '<username>' AND
      TABLESPACE_NAME LIKE 'DATA%';
```

The parameters are:

- `TABLESPACE_NAME`: The tablespace for which the quota is assigned. In Autonomous Database, tablespaces are managed automatically and have `DATA` as a prefix.
- `BYTES`: The amount of space currently used by the user in the tablespace.
- `MAX_BYTES`: The maximum quota assigned (in bytes). If `MAX_BYTES` is -1, it means the user has unlimited quota on the tablespace. The database user creating the vector index must have `MAX_BYTES` sufficiently larger than bytes to accommodate the vector index, or `MAX_BYTES` should be -1 for unlimited quota.

7.7 Limitations of Data Science Agent

While Data Science Agent offers numerous benefits, there are certain limitations that may impact its use in specific scenarios.

The current limitations of Data Science Agent include:

- Ad hoc SQL queries cannot be run directly
- Algorithms supported by Oracle permitted for models
- Conversation length and scope
- Error handling
- Performance and latency related limitations
- Reuse of existing objects

7.7.1 Ad hoc SQL queries cannot be run directly

Data Science Agent is capable of generating SQL internally to create views. However, it does not support running of ad hoc SQL queries or direct visualization of raw result sets currently.

Statistical analysis of Data Science Agent is highly effective when working with row-level datasets, and not aggregated outputs. However, some analyses on grouped data can be performed if the row count per group is large. Therefore, for more reliable analysis and modeling, use views that have ungrouped or only minimally aggregated data.

> **Note**
>
> You can define arbitrary views to structure and transform data for downstream analysis and modeling.

7.7.2 Algorithms supported by Oracle permitted for models

Data Science Agent permits algorithms that are only supported by Oracle. Currently, the agent supports the following machine learning functions—Classification, Regression, Clustering, and Anomaly Detection.

> **Note**
>
> Inference or scoring is not supported for Clustering and Anomaly Detection.

7.7.3 Conversation length and scope

While Data Science Agent can handle extended interactions, very long conversations may gather context that negatively affects clarity or performance. For extended work, consider starting a new conversation after a substantial number of interactions (around 50 messages), particularly when your objectives change.

7.7.4 Error handling

Data Science Agent may occasionally encounter constraints. For example, unsupported column types for modeling. These are typically resolved by adjusting the data or approach. If you encounter such constraint, ask the agent for suggestions on how to solve minor issues.

> **Note**
>
> Oracle recommends refining prompts, adjusting goals, or re-running steps.

7.7.5 Performance and latency related limitations

Certain operations such as data discovery, feature analysis, and model training may require a few minutes to process. Model training on very large datasets can take even longer. During these operations, the conversation may not progress until the operation is completed. If you encounter such performance or latency related issues, you can start other conversations.

7.7.6 Reuse of existing objects

Data Science Agent may reuse existing objects—views or models, although *starting from scratch* is also an option. If you prefer that the agent doesn't reuse previous objects, you can state so in your response. Otherwise, the agent may refer to or reuse relevant objects created earlier—including those from other conversations—when they are manually associated or automatically discovered. This is done to save time by avoiding repeated creation of the same objects.

> **Note**
>
> If several similar objects are available, make sure that you specify whether to reuse or recreate the objects.

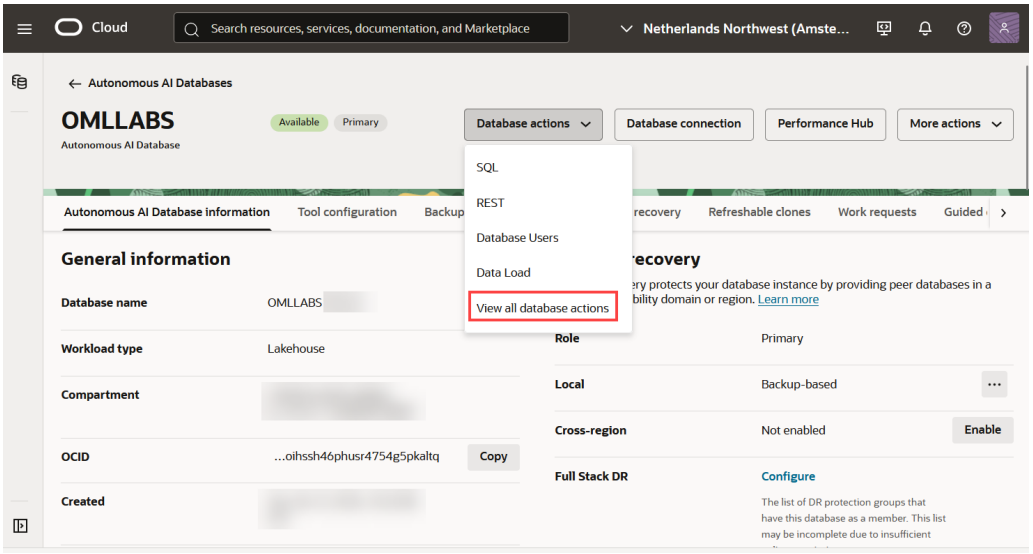
7.8 Access Data Science Agent

You can access Data Science Agent directly from Oracle Machine Learning UI home page.

To access Data Science Agent, you must first sign into Oracle Machine Learning from Autonomous AI Database:

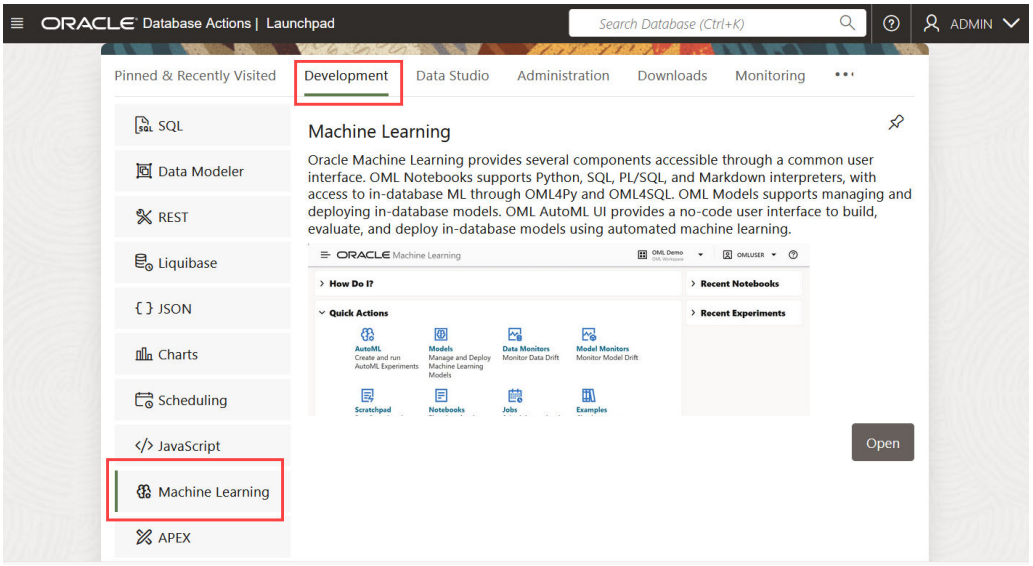
1. To sign into Oracle Machine Learning from an Autonomous AI Database instance:
   a. On the Autonomous AI Database information page click **Database actions** and then click **View all database actions.**

Figure 7-1 Database Actions



b. On the Database Actions page, go to the **Development** tab and click **Machine Learning**.

Figure 7-2 Oracle Machine Learning option in Database Actions Launchpad



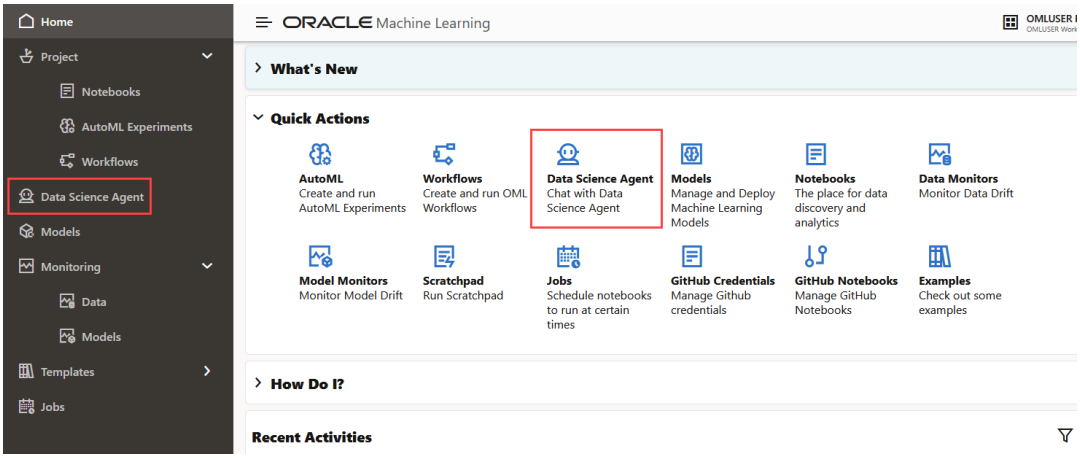
The Oracle Machine Learning UI sign in page opens.

c. Enter your username and password, and click **Sign in.**

This opens the Oracle Machine Learning UI home page.

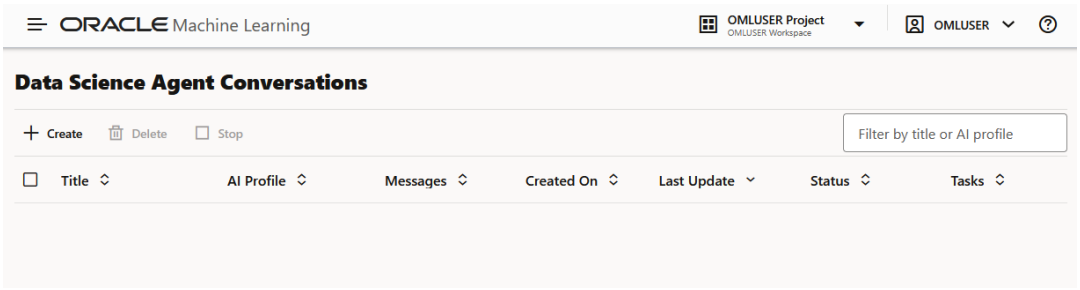
2. On your Oracle Machine Learning UI home page, click **Data Science Agent**.

Figure 7-3 Oracle Machine Learning UI homepage



This opens the Data Science Conversations listing page.

Figure 7-4 Data Science Agent Conversations Listing page



7.9 Create an OCI Generative AI Credential and AI Profile

An AI credential stores authentication details used by the database to access the selected AI provider or related cloud resources. The AI credential comprises information such as `user_ocid`, `tenancy_ocid`, `private_key` and `fingerprint`. An AI Profile is information about a user and their attributes, such as `provider`, `credential_name`, and `object_list`.

Prerequisites

- `DBMS_CLOUD_AI` package
- `user_ocid`
- `tenancy_ocid`
- `private_key`
- `fingerprint`

You can create and manage your AI profiles using the `DBMS_CLOUD_AI` package.

To create an OCI Generative AI credential and AI Profile:

1. Create a notebook and in a `%script` paragraph, run the following command to create an AI credential:

```
%script
DECLARE
    credential_name VARCHAR2(128) := 'OCI_CRED';
BEGIN
    BEGIN
        dbms_cloud.drop_credential(credential_name => credential_name);
    EXCEPTION
        WHEN OTHERS THEN
            NULL;
    END;
    dbms_cloud.create_credential(
        credential_name => credential_name,
        user_ocid       => '<ocid1.user.oc1..>',
        tenancy_ocid    => '<ocid1.tenancy.oc1..>',
        private_key     => '<private_key>',
        fingerprint     => '<fingerprint>'
    );
END;
/
```

   This PL/SQL script calls the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure to create a new credential with the given parameters:

   - `credential_name`: Name of the credential. In this example, the credential name is `OCI_CRED`.
   - `user_ocid`: This is the Oracle Cloud Identifier, a unique ID for the user. See Where to Get the Tenancy's OCID and User's OCID for details.
   - `tenancy_ocid`: This is the Oracle Cloud Identifier for your tenancy (your OCI account). See Where to Get the Tenancy's OCID and User's OCID for details.
   - `private_key`: Specify the generated private key. Private keys generated with a passphrase are not supported. You must generate the private key without a passphrase. See How to Generate an API Signing Key for details.
   - `fingerprint`: Specify the fingerprint. After a generated public key is uploaded to the user's account the fingerprint is displayed in the console. Use the displayed fingerprint for this argument. See How to Get the Key's Fingerprint and How to Generate an API Signing Key for more information.

2. In another `%script` paragraph in the same notebook, run the following command to create an AI profile by the name `GROK_4_3_PROFILE`.

```
%script
DECLARE
    profile_name VARCHAR2(128) := 'GROK_4_3_PROFILE';
BEGIN
    dbms_cloud_ai.drop_profile(
        profile_name,
        TRUE
    );
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes   => '{
```

```
            "credential_name": "OCI_CRED",
            "model": "xai.grok-4.3",
            "provider": "oci",
            "temperature": 1,
            "max_tokens": 8192,
            "oci_compartment_id": "<ocid1.compartment.oc1..>"
        }'
    );
END;
/
```

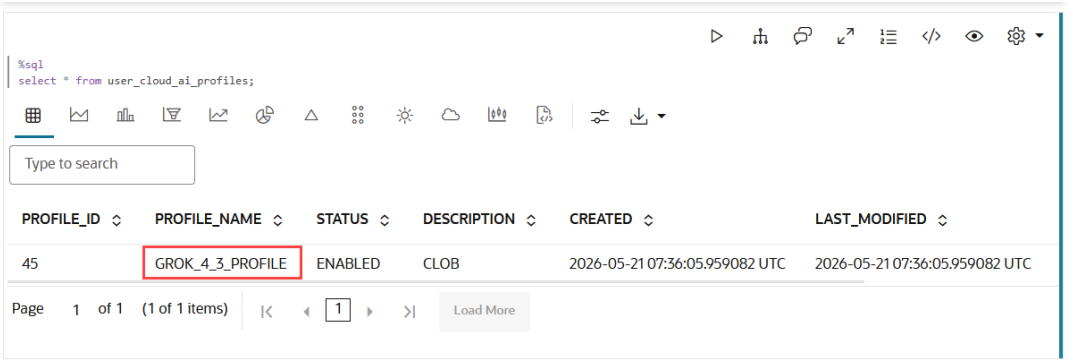
Define the following attributes for this profile:

- `profile_name`: A name for the AI profile. The profile name must follow the naming rules of Oracle SQL identifier. The maximum profile name length is 125 characters.
- `credential_name`: This is the name of the credential used to authenticate requests to the selected AI provider.
- `model`: The name of the AI model being used to generate responses in the conversation. In this example, it is `xai.grok-4.3`. For more information, see Recommended Models.
- `provider`: This is the provider of the model. It is a mandatory field. Supported providers are:
  - openai
  - cohere
  - azure
  - database
  - oci
  - google
  - anthropic
  - huggingface
  - aws
- `max_tokens`: Specify the maximum number of tokens (words and pieces of words) in the response. Prevents overly long outputs and manages cost.
- `oci_compartment_id`: This is the OCID of the compartment you are permitted to access when calling the OCI Generative AI service. The compartment ID can contain alphanumeric characters, hyphens and dots.

3. Check the status of the profile creation by running the following:

```
%sql select * from
    user_cloud_ai_profiles;
```

Figure 7-5 AI Profile



This completes the task of creating an AI credential and AI Profile.

- Recommended Models

7.9.1 Recommended Models

Data Science Agent works with large language models accessed through Oracle `DBMS_CLOUD_AI` and `DBMS_CLOUD_AI_AGENT` packages. The `DBMS_CLOUD_AI` package, with Select AI, supports the translation of natural language prompts to generate, run, explain SQL statements, and also enables RAG and natural language-based interactions, including chats with LLMs. For more information, see DBMS_CLOUD_AI Package and DBMS_CLOUD_AI_AGENT Package.

This table lists the recommended large language models and the scenarios in which each should be used.

> **Note**
>
> Recommended models may change as providers update model availability, latency, pricing, and quality. Check the current list of supported models for your AI provider before creating a profile.

Table 7-3 Recommended Models

Provider	Tier	Large Language Model
OpenAI or OCI GenAI	Top	`gpt-5.5`
OpenAI or OCI GenAI	Cost-effective	`gpt-5.4-mini`

Table 7-3 (Cont.) Recommended Models

Provider	Tier	Large Language Model
OCI GenAI	Top	`xai.grok-4.3`
		Note When using OCI Generative AI, use the provider and model identifiers exactly as documented for OCI GenAI. Some model identifiers may include the original model family or vendor name.
OCI GenAI	Cost-effective	`xai.grok-4-1-fast-reasoning`
		Note When using OCI Generative AI, use the provider and model identifiers exactly as documented for OCI GenAI. Some model identifiers may include the original model family or vendor name.
Google	Top	`gemini-3.5-flash`

Table 7-3 (Cont.) Recommended Models

Provider	Tier	Large Language Model
Google	Cost-effective	`gemini-3-flash-preview`
Anthropic	Top	`claude-opus-4-8`
Anthropic	Cost-effective	`claude-sonnet-4-6`

Explanation of tiers:

- **Top:** Represents the best state-of-the-art model from a specific provider. This tier is the strongest option in terms of quality, reliability, and precision.
- **Cost-effective:** Represents a good compromise between quality, cost, and speed. These models are typically faster and less expensive. But the trade-off is lower quality and reliability compared to the Top tier.

Profile Creation for GPT-5.5

To create an AI profile for GPT-5.5 through OpenAI, run the following script in a notebook:

```
DECLARE
    profile_name VARCHAR2(128) := 'OPENAI_GPT_5_5';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "credential_name": "OPENAI_CRED",
            "model": "gpt-5.5",
            "provider": "openai",
            "temperature": 1,
            "max_tokens": 8192
        }'
    );
END;
/
```

To create an AI profile for GPT 5.5 through Oracle Cloud Infrastructure (OCI), run the following script in a notebook:

```
DECLARE
    profile_name VARCHAR2(128) := 'OCI_GPT_5_5';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "credential_name": "OCI_CRED",
            "model": "openai.gpt-5.5",
            "provider": "oci",
            "temperature": 1,
            "max_tokens": 8192,
            "oci_compartment_id": "<your-dep-id>"
          }'
```

```
        );
    END;
/
```

Profile Creation for Grok 4.3

To create an AI profile for Grok 4.3 through Oracle Cloud Infrastructure (OCI), run the following script in a notebook:

```
DECLARE
    profile_name VARCHAR2(128) := 'OCI_GROK_4_3';
BEGIN
        dbms_cloud_ai.drop_profile(profile_name, TRUE);
        dbms_cloud_ai.create_profile(
            profile_name => profile_name,
            attributes => '{
                "credential_name": "OCI_CRED",
                "model": "xai.grok-4.3",
                "provider": "oci",
                "temperature": 1,
                "max_tokens": 8192,
                "oci_compartment_id": "<your-dep-id>"
               }'
            );
    END;
/
```

Parameters:

- `profile_name`: Name of the AI profile. It must follow Oracle SQL identifier naming rules.
- `credential_name`: Name of the credential used to authenticate with the selected AI provider.
- `model`: Model identifier used by the selected provider.
- `provider`: AI provider for the profile, for example `openai`, `oci`, `google`, or `anthropic`.
- `temperature`: Recommended value for Data Science Agent examples: `1`.
- `max_tokens`: Recommended value for these examples: `8192`.

> **Note**
>
> Data Science Agent profile inspection treats values below `4096` as not recommended.

- `oci_compartment_id`: Required for the OCI GenAI examples. Use the target compartment OCID or documented deployment/compartment identifier.

For more information, see Manage AI Profiles.

7.10 Manage Data Science Conversations

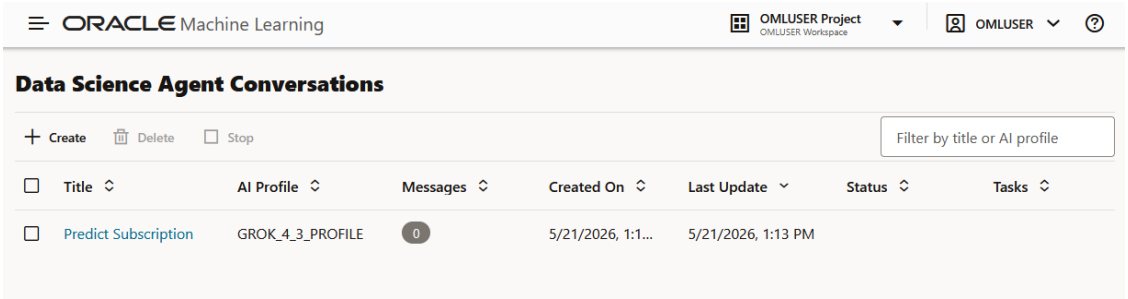
The Data Science Agent Conversations page lists all the conversations you created. Here, you create and manage conversations.

Create your Data Science Agent conversation to interact with the agent on these areas of data science and machine learning:

- Data profiling
- Data wrangling and transformation
- Statistical analysis of variable relationships
- Feature Importance, Classification, Regression, XGBoost, Clustering, and Anomaly Detection.
- Model training and evaluation
- Inference on new data

Here is the Data Science Conversations page. It lists all the conversations that you created. Click a conversation title to open it and resume it.

Figure 7-6 Data Science Conversation listing page



This page lists the following details about the conversations:

- **Title:** This is the name of the conversation you provided while creating the conversation.
- **Profile Name:** This is the AI Profile you selected while creating the conversation. An AI Profile contains information about the user and their attributes, such as the `provider`, `credential_name`, and `object_list`. You can create and manage your AI profiles through `DBMS_CLOUD_AI` package.
- **Messages:** This indicates the number of interactions in the conversation with the agent. As shown in the screenshot, the conversation titled `SQL` has 2 messages or interactions as of the last updated date.
- **Created on:** This is the date on which the conversation was first created.
- **Last Updated:** This is the date on which the conversation was last used or updated.
- **Status:** There are two statuses ACTIVE and IDLE.

You can perform the following tasks here:

- Create a Data Science Agent Conversation
  A conversation is a set of interactions with Data Science Agent in the chat interface. Before you start a conversation with the Data Science Agent, you must create a conversation.
- Delete a Data Science Agent Conversation
  You can delete a conversation from the Data Science Agent Conversations listing page.

7.10.1 Create a Data Science Agent Conversation

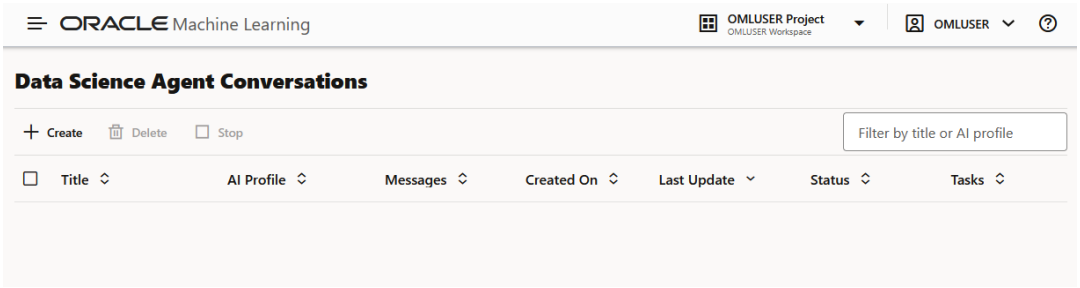
A conversation is a set of interactions with Data Science Agent in the chat interface. Before you start a conversation with the Data Science Agent, you must create a conversation.

> **Note**
>
> You can use the same Database user credentials to access the same conversation in multiple browsers. However, Oracle does not recommend this as it may lead to unexpected behavior. If you attempt this, Data Science Agent will display a warning, but you will have the option to override it.

To create a Data Science Agent conversation:

1. On the Data Science Agent Conversation page, click **Create**.

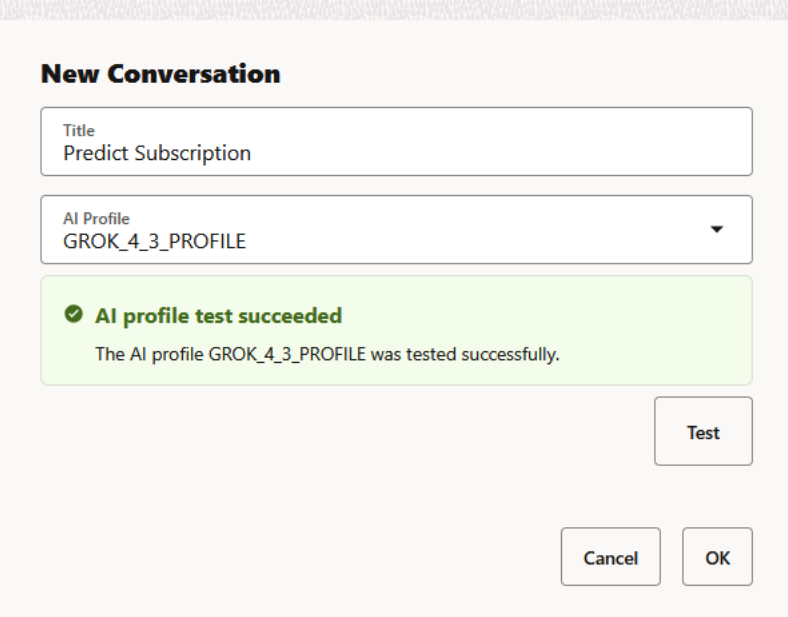
Figure 7-7 Data Science Agent listing page



This opens the New Conversation dialog.

2. In the New Conversation dialog, enter the following details:

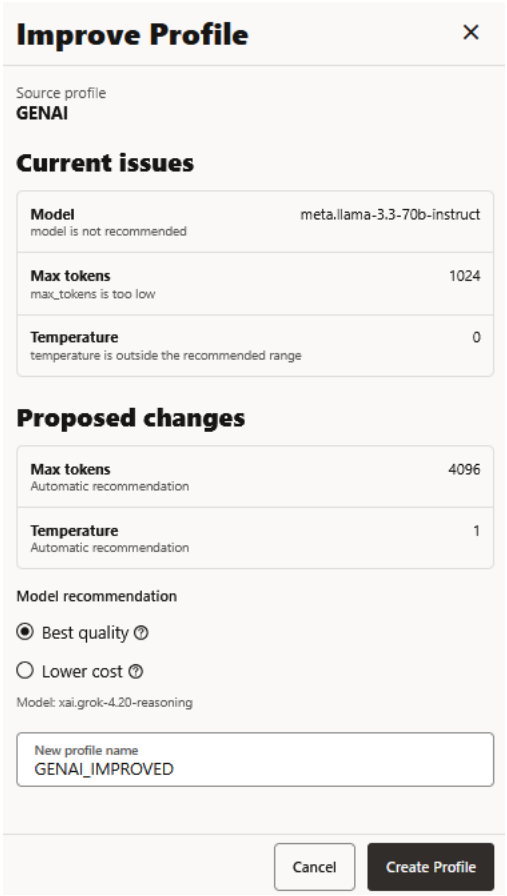
Figure 7-8 Create Conversation dialog



3. In the **Title** field, provide a name for your conversation. In this example, create a conversation named `Predict Subscription`.
4. In the **AI Profile** drop-down menu, click on the down arrow and select a profile. Select the profile **GROK_4_3_PROFILE**.
5. In case the selected profile has any issues, Data Science Agent displays a message showing the current issues with the profile. In the example below, Data Science Agent has diagnosed issues related to model compatibility, tokens, and temperature settings with the AI profile GENAI.

   Scroll down to view the proposed changes. Data Science Agent proposes the required changes, and recommends models based on best quality and lower costs. Select any model recommendation options here. You may also change the improved profile name suggested by the agent. Click **Create Profile**.

Figure 7-9 Improve Profile



> **Note**
>
> You will encounter this dialog only if there are issues with the selected AI Profile.

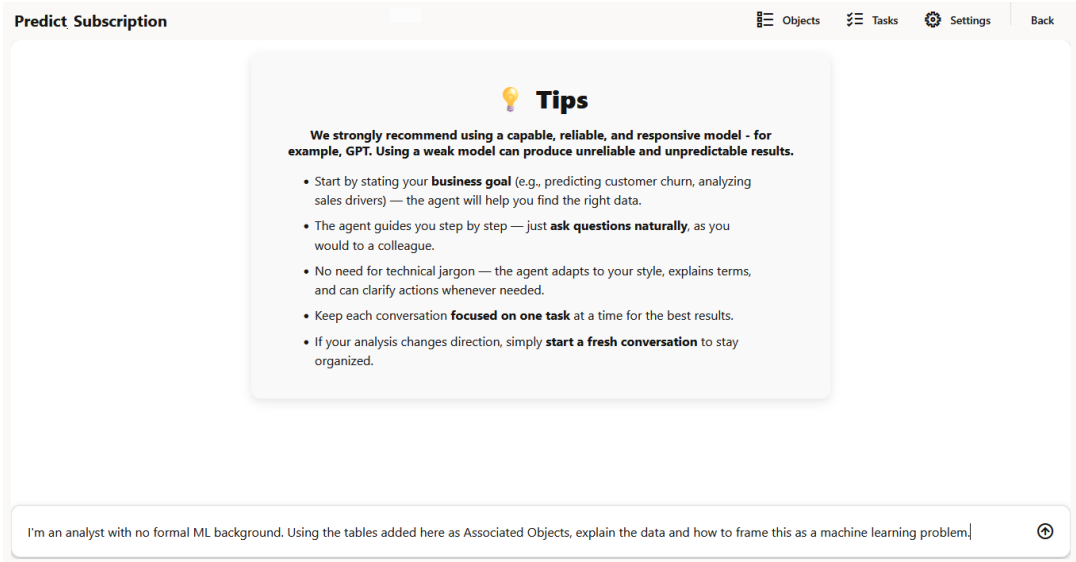
6. Click **Test** to test the selected AI profile. AI profiles may show warnings if the parameters `model`, `temperature`, or `max_tokens` are outside the recommended DSAgent ranges.

> **Note**
>
> A warning does not necessarily mean the profile cannot be selected. Review the warning before continuing.

> **Note**
>
> Profile test failures can be caused by Access Control List (ACL), missing or deleted credentials, invalid credentials, invalid model, timeout, or unexpected `DBMS_CLOUD_AI.GENERATE` errors. To resolve any errors, check the ACL access, credential validity, model availability, and the request ID shown in the error.

7. Click **OK**. The conversation is created only if the selected profile test succeeds. Once the conversation is created, it opens the Data Science Agent chat interface. Here, you can start chatting with Data Science Agent.
8. In the chat interface, Data Science presents tips to begin your conversation. In the **Send a message** field, type in your prompt in natural language and press Enter.

Figure 7-10 Chat with Data Science Agent



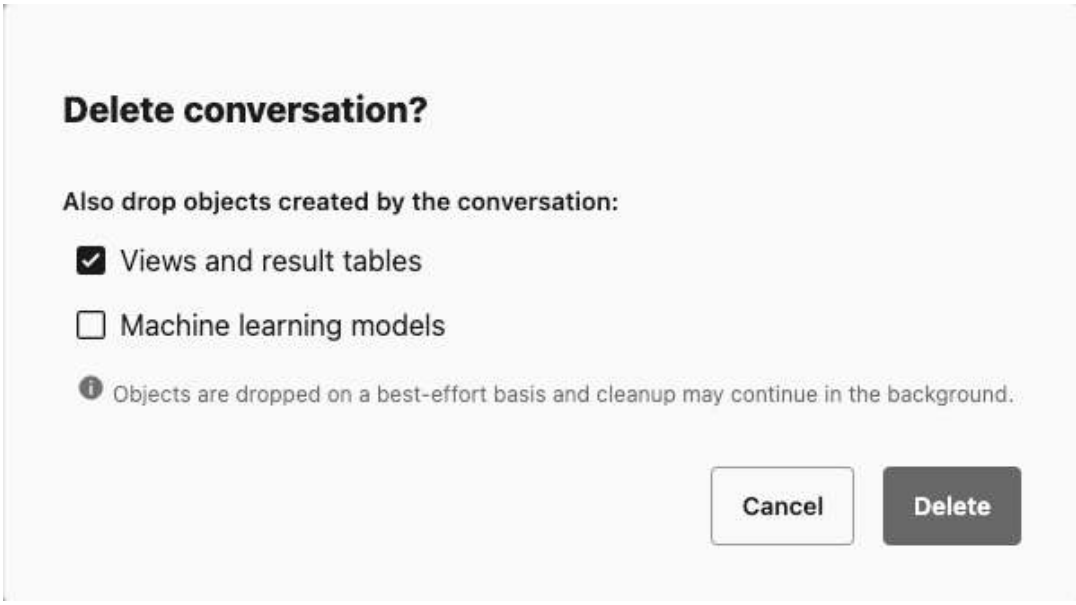
7.10.2 Delete a Data Science Agent Conversation

You can delete a conversation from the Data Science Agent Conversations listing page.

To delete a conversation:

1. On the Data Science Agent Conversations listing page, select the conversation you want to delete and click **Delete**. The Delete conversation dialog opens.
2. In the Delete Conversation dialog:
   - Select **Views and results table** to drop views and tables associated with the conversation.
   - Select **Machine learning models** to drop machine learning models associated with the conversation.

Figure 7-11 Delete Conversation dialog



3. Click **Delete**. This completes the task of deleting the conversation.

7.11 Use Data Science Agent Chat Interface

On the Data Science Agent chat interface, you interact with the agent for all machine learning and data science related questions.

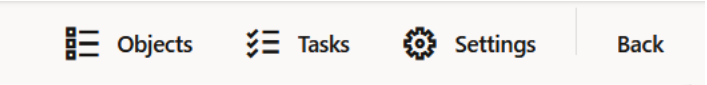
To begin a chat with Data Science Agent:

1. On the Data Science Agent chat interface, type your question in the **Ask a question** field, and press enter.

   For more information, see Data Science Agent: Sample Prompts and Outputs and Associate Database objects to your conversation.

2. On the top right corner of the chat interface, you can configure the settings of your conversation:

Figure 7-12 Toolbar



   a. Click **Objects** to manually associate database objects to the conversation. The objects are tables, views, and machine learning models.

   b. Click **Tasks** to view tasks running in the background.

   c. Click **Settings** to view and edit AI Profile, and define database service levels.

   d. Click **Back** to go back to the Data Science Conversation listing page.

- Add Objects to Data Science Agent Conversation
  To improve the precision and efficiency of Data Science Agent's responses, you can manually add database objects to a conversation.

- Manage AI Profile and Service Level of Data Science Conversation
  You can manage AI Profile and Database Service Levels of a conversation from the Settings option in the Data Science Conversation chat interface. These settings are conversation-specific.

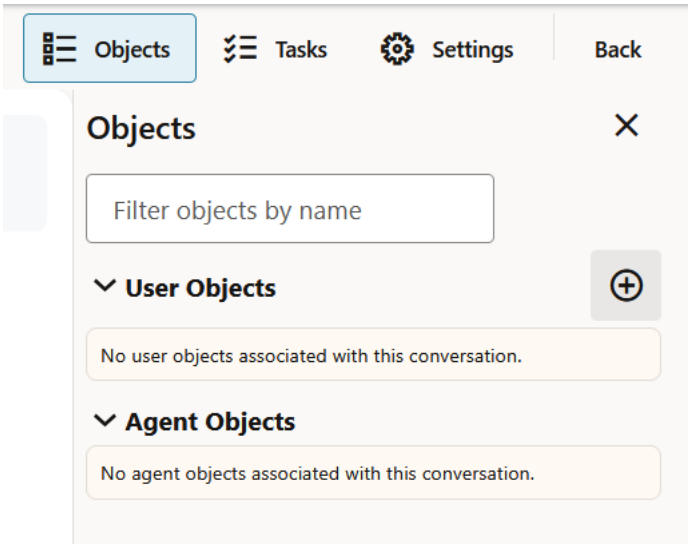
7.11.1 Add Objects to Data Science Agent Conversation

To improve the precision and efficiency of Data Science Agent's responses, you can manually add database objects to a conversation.

To add objects to your conversation:

1. On the Data Science Agent chat interface, click **Objects** on the top right corner of the page. This opens the Objects pane on the right.

Figure 7-13 Objects pane

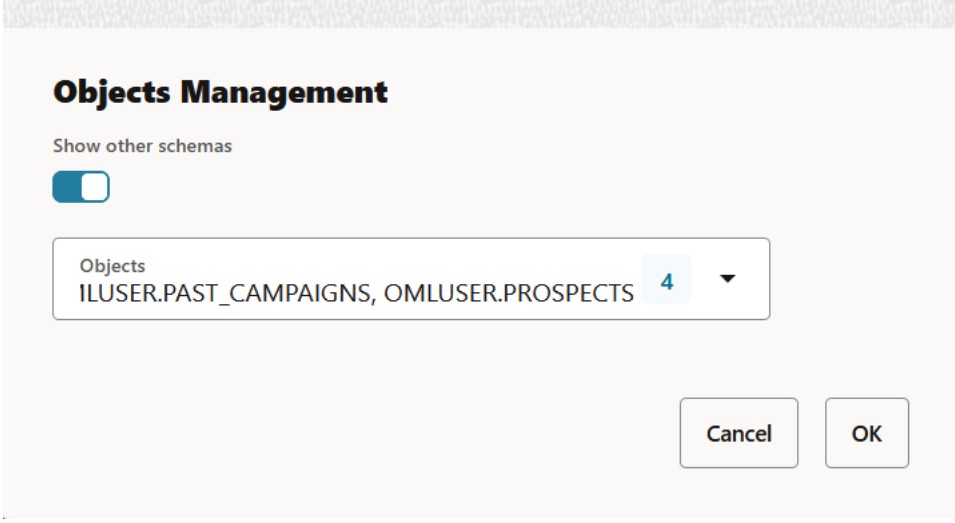


2. Click the + icon to add objects to the conversation. The Object Management dialog opens. You can also search for objects by typing the name in the search box.

> **Note**
>
> The objects are tables, views, and machine learning models. You can associate multiple objects with the conversation.

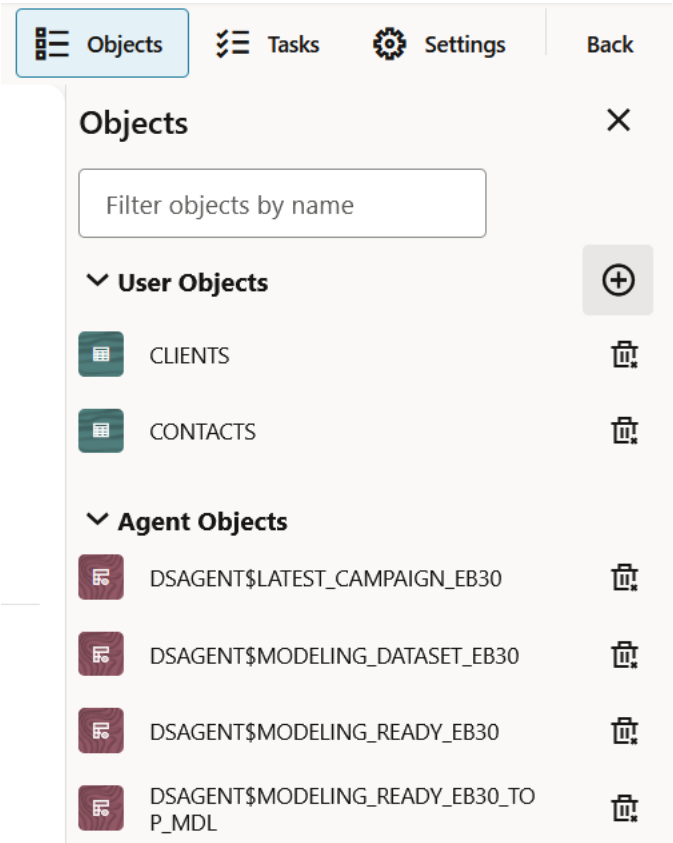
Turn on **Show other schemas** to view objects present in schemas besides your own. Select the objects from the drop-down list. You can associate multiple objects with the conversation.

Figure 7-14 Object Management dialog



3. Click **OK** to associate objects with the conversation. Once the objects are successfully associated, they are listed under **User Objects**. To disassociate an object, click the delete icon.

Figure 7-15 Objects pane



The **Agent Objects** section displays the agent-generated objects for the conversation.

> **Note**
>
> Objects created by Data Science Agent are prefixed with `DSAGENT$`. This distinguishes them as agent-generated, prevents naming conflicts, and enables safe routine cleanup.

4. Click **X** to exit the Objects pane.
5. Click **Back** to go back to the Data Science Agent Conversation listing page.

This completes the task of associating objects with your conversation.

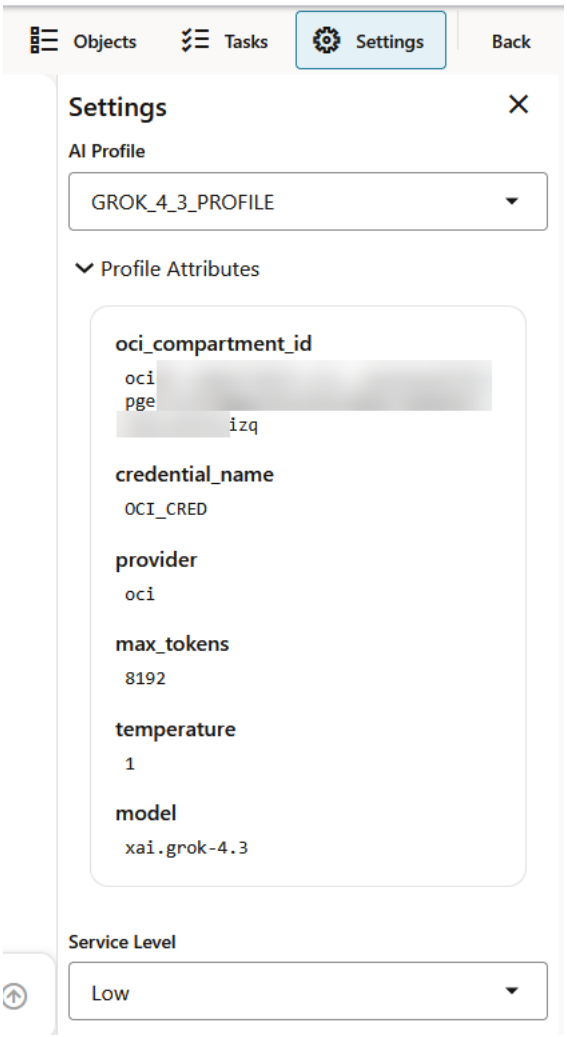
7.11.2 Manage AI Profile and Service Level of Data Science Conversation

You can manage AI Profile and Database Service Levels of a conversation from the Settings option in the Data Science Conversation chat interface. These settings are conversation-specific.

To view and edit AI Profile and Database Service level :

1. On the Data Science Agent chat interface, click **Settings** on the top right corner of the page. This opens the Settings pane on the right.
2. On the Settings pane, you can manage the following:

Figure 7-16 Settings pane



- Click **AI Profile** down arrow to select an AI profile for your conversation. You can also change the profile in the middle of a conversation.
- Click **Profile Attributes** down arrow to view the details of the selected AI profile.
- Click **Service Level** down arrow to select a database service level. By default, the service level is set to **Low**.

> **Note**
>
> If you are working with a large dataset, consider changing the service level to **Medium** or **High** to achieve parallelism in the database backend.

3. Click **X** to exit the Settings pane.
4. Click **Back** to go to the Data Science Agent Conversation listing page.

7.12 Example of a Conversation with Data Science Agent

This example demonstrates how Data Science Agent supports a data analyst in exploring a dataset, and in building and evaluating a machine learning model. It also shows the effectiveness of Natural Language to SQL (NL2SQL) in Data Science Agent.

The user runs a complete machine learning workflow using natural language. The conversation between the user and Data Science Agent progresses from generic questions on data exploration in the `OMLUSER` schema to specific ones on model training, model building and scoring. Broadly, the conversation flow can be categorized into these sections:

- Goal setting for the conversation
- Explanation of the dataset in the tables `CLIENTS`, `CONTACTS`, `PAST_CAMPAIGNS`, and `PROSPECTS`.
- Data preparation for model building
- Training and building a model
- Computation of model accuracy
- Interpretation of the model's prediction
- Making prediction by using the trained model

Highlights:

This example highlights the following capabilities of Data Science Agent:

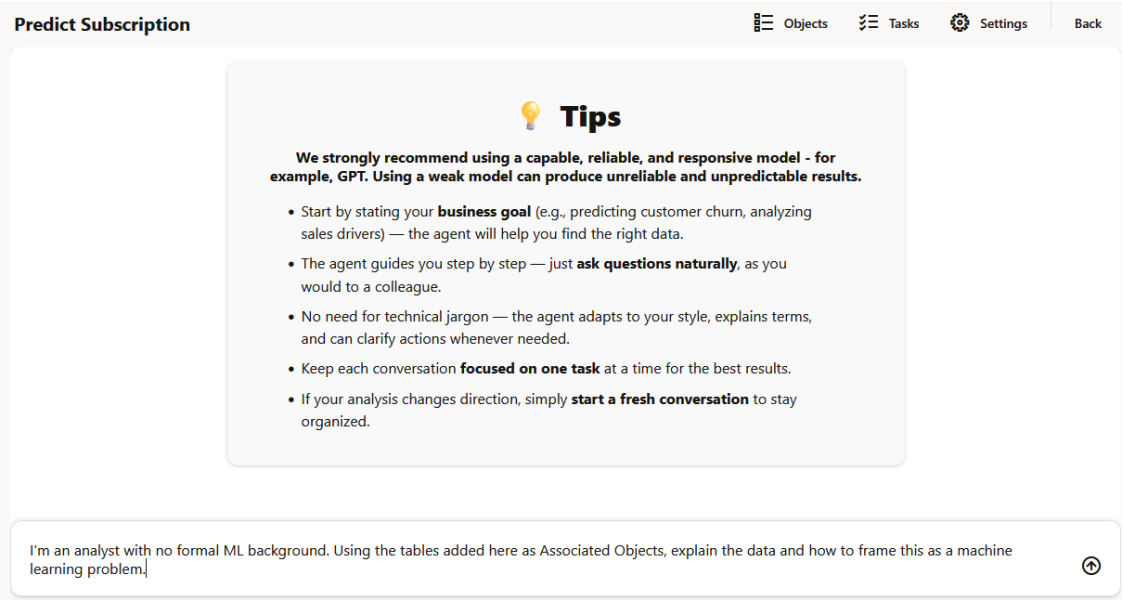
- Efficiency in performing machine learning tasks—data preparation, model building, model training, model evaluation, data analysis and so on
- Natural language to SQL capabilities
- Visualization capabilities
- Intelligence to suggest the next steps and options to the user

1. Goal and context setting

For every new conversation, Data Science Agent opens the chat interface with some important tips for you. Get started to interact with the agent with a prompt that sets your goal and context for the conversation. Start the conversation with an introduction with specific information to set the goal and context.

Prompt 1: `I'm an analyst with no formal ML background. Using the tables added here as Associated Objects, explain the data and how to frame this as a machine learning problem.`

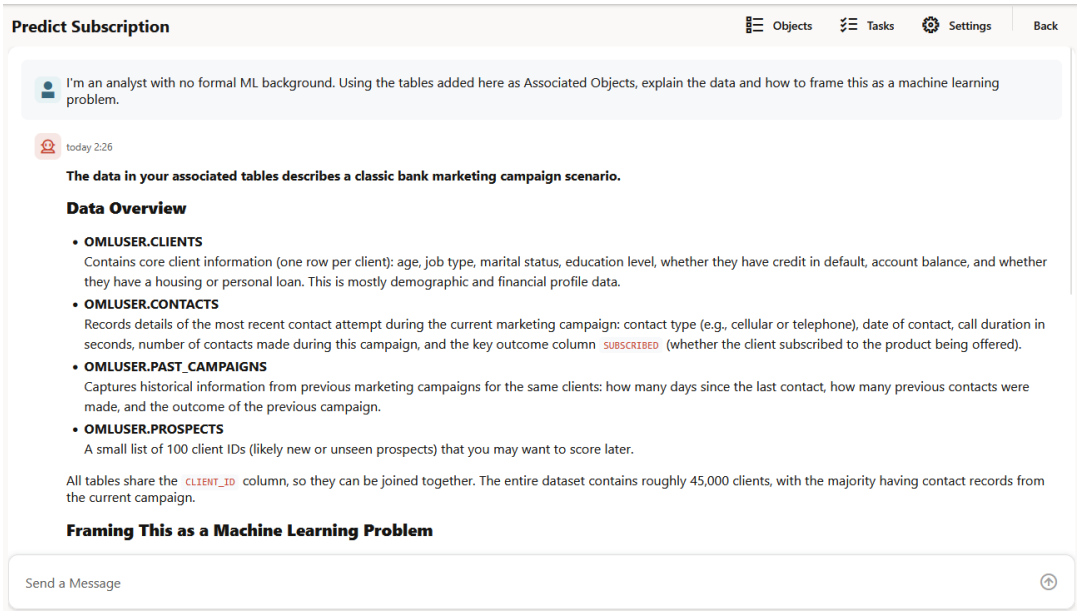
Figure 7-17 Goal and context setting



Agent's response:

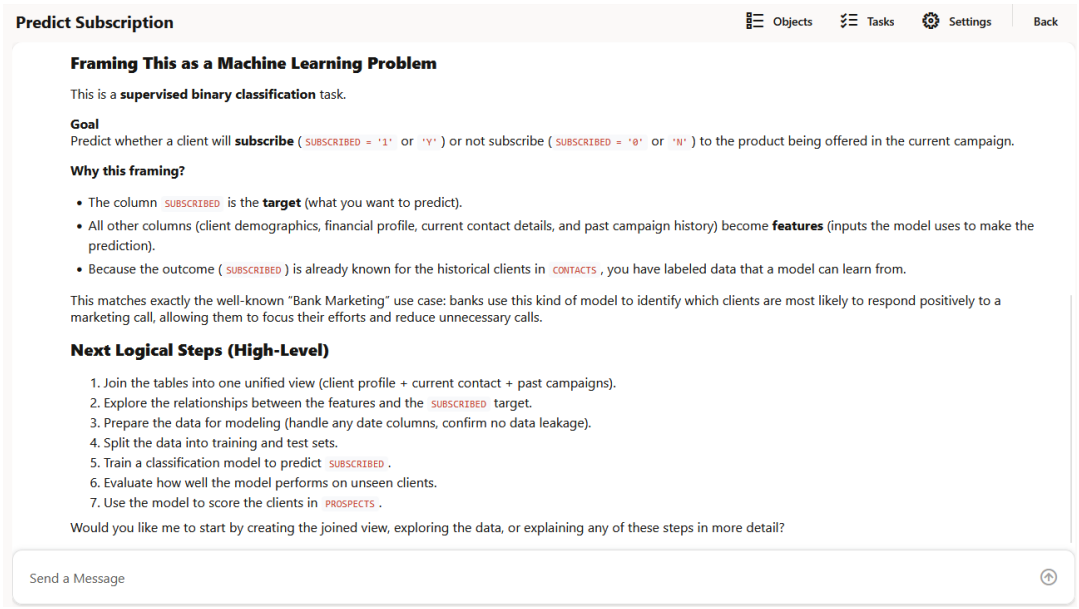
- Summary of the data in each table and the key columns

Figure 7-18 Prompt 1 and response



- How to frame a machine learning problem
- Steps to frame ML problem
- A summary of the dataset

Figure 7-19 Response 1 (concluded)



2. Data Science Agent explores the dataset

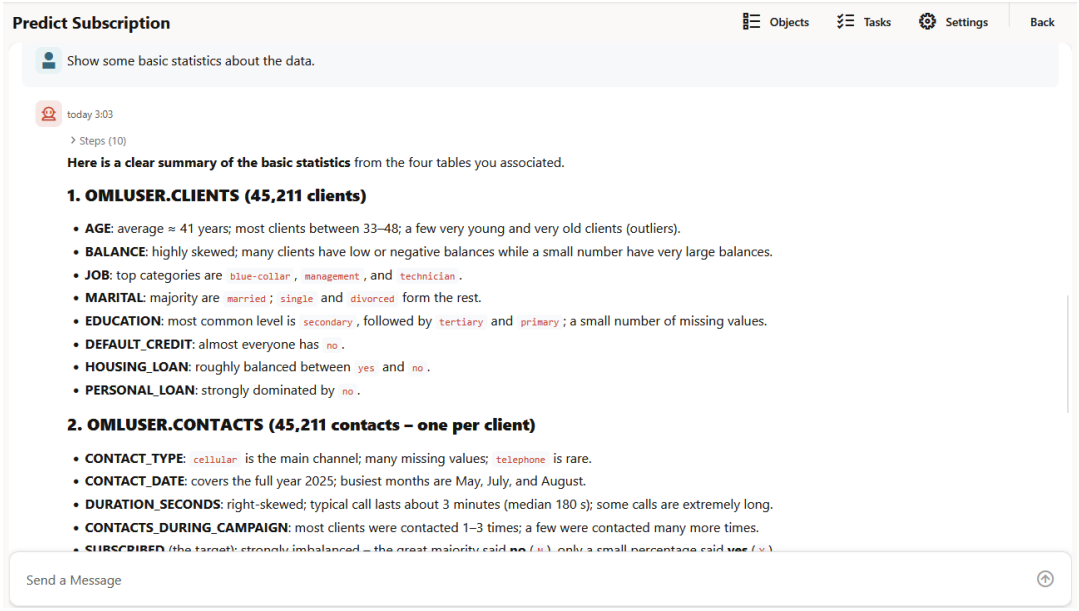
Here, let's explore the dataset.

Prompt 2: `Show some basic statistics about the data.`

Agent's response: Data Science Agent responds to this prompt in a structured format. It provides the following information:

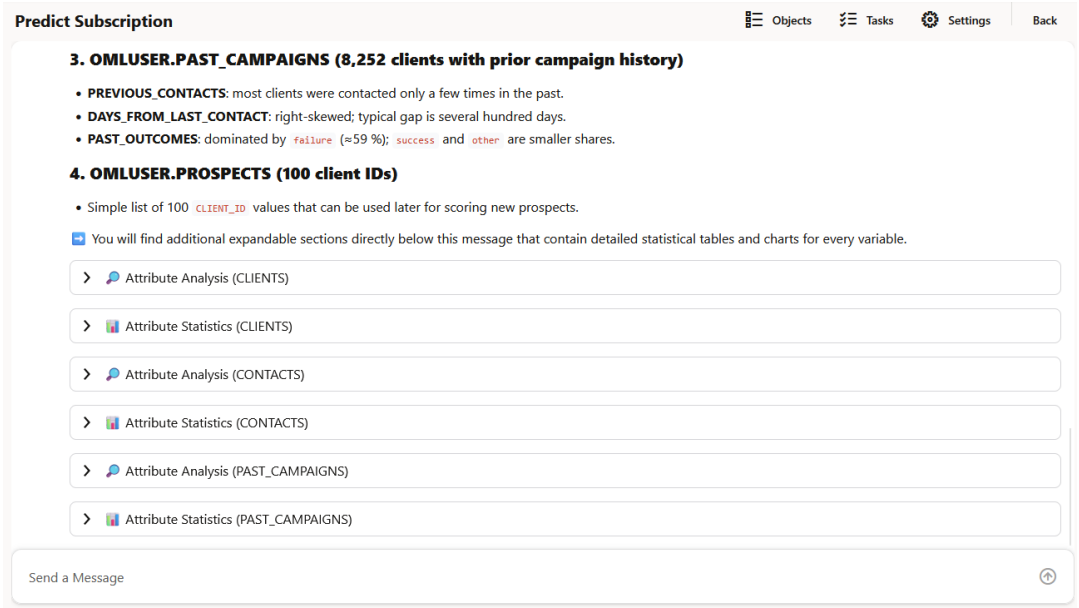
- An insight on the tables `CLIENTS`, `CONTACTS`, `PAST_CAMPAIGNS` and `PROSPECTS`.

Figure 7-20 Prompt 2 and response



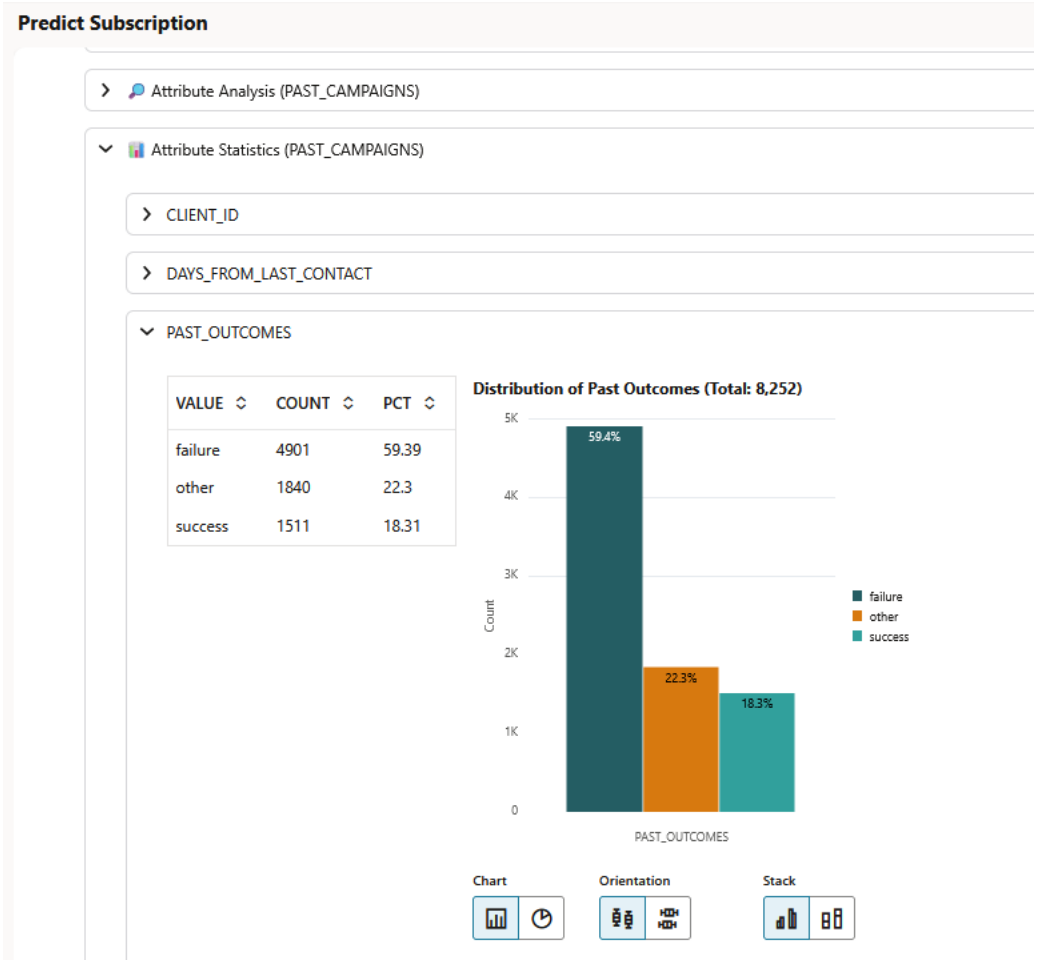
- Provides statistical analysis of each table. Click **Attribute Statistic (table name)** to expand the section. It presents the information in a tabular format and also in a graph, as applicable.

Figure 7-21 Response 2 (contd.)



- Provides an attribute analysis of each table. Click **Attribute Analysis (table name)**. It presents the information in a tabular format and also in a graph, as applicable.

Figure 7-22 Response 2



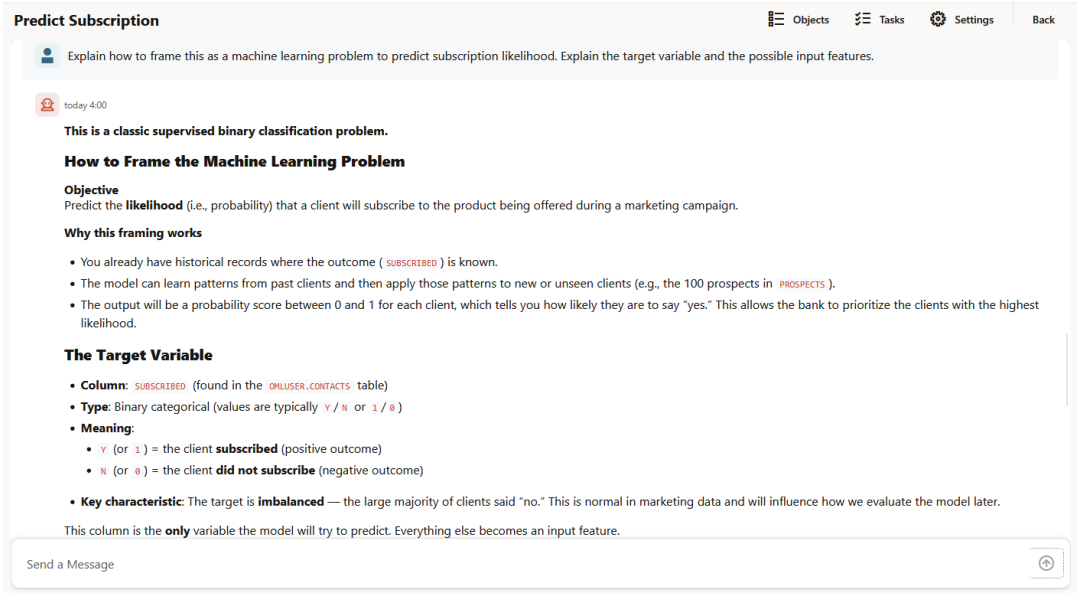
3. Data Science Agent explains how to use the data to frame a machine learning problem

Prompt 3: `Explain how to frame this as a machine learning problem to predict subscription likelihood. Explain the target variable and the possible input features.`

Agent's response: Data Science Agent provides a point-wise explanation of how to frame the question as a machine learning (ML) problem.

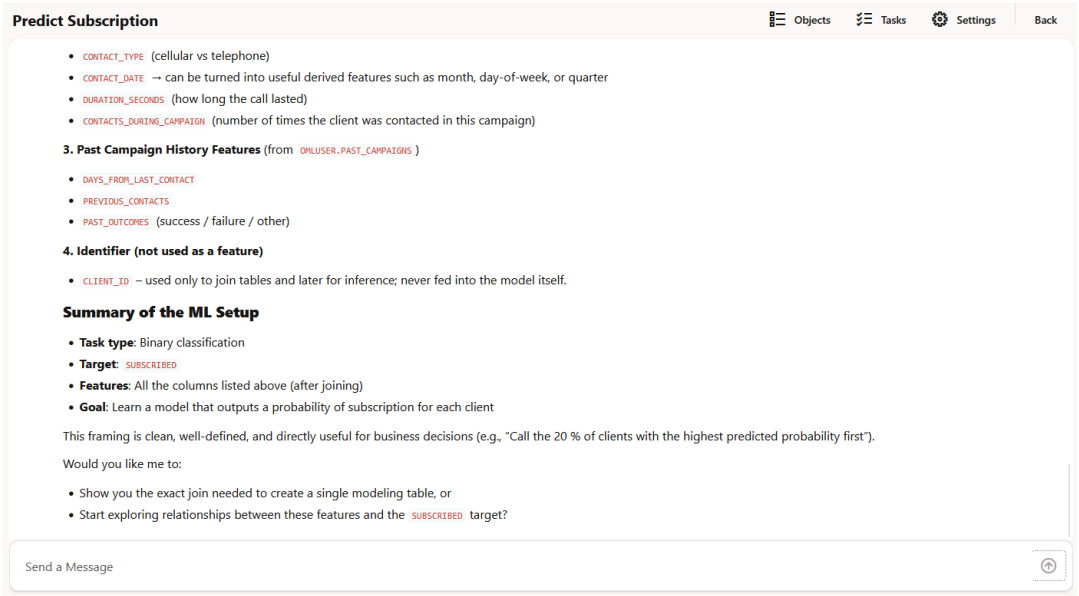
- Explains how to frame the machine learning problem
- Describes the target variable in the machine learning problem

Figure 7-23 Prompt 3 and response



- Describes the input features in the context of making predictions in an ML problem and how to use all these features to solve an ML problem
- A crisp summary of the machine learning setup
- The next steps

Figure 7-24 Response to prompt 3 (concluded)



4. Data Science Agent creates a single modeling table

Prompt 4: `Yes. Show the exact joins to create a single modeling table.`

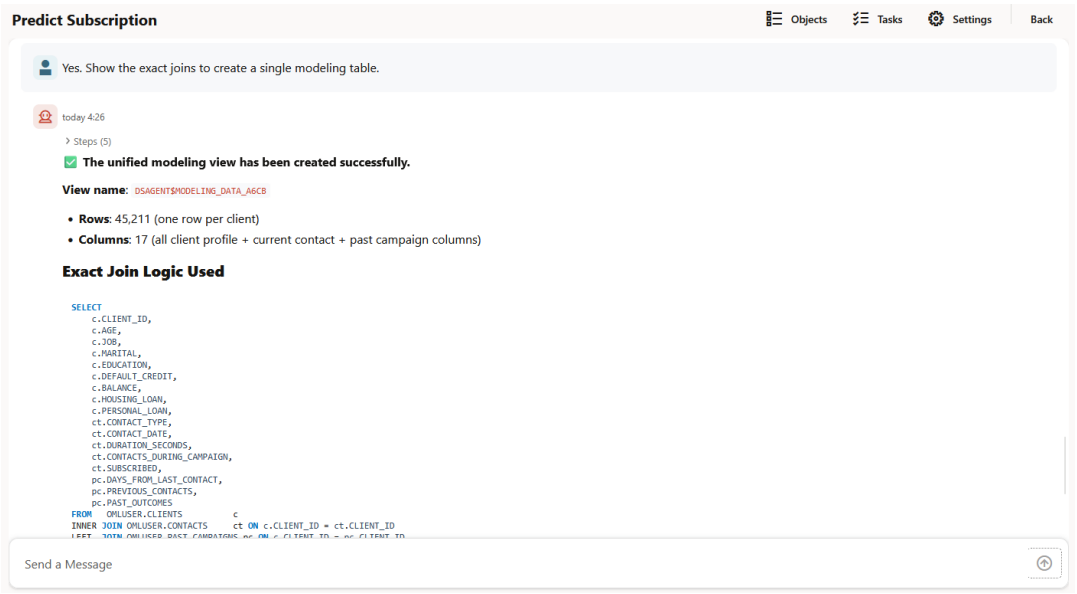
Agent's response:

- Data Science Agent creates the unified modeling view `DSAGENT$MODELING_DATA_A6CB`

> **Note**
>
> Views and objects created by Data Science Agent have the prefix `DSAGENT$`.

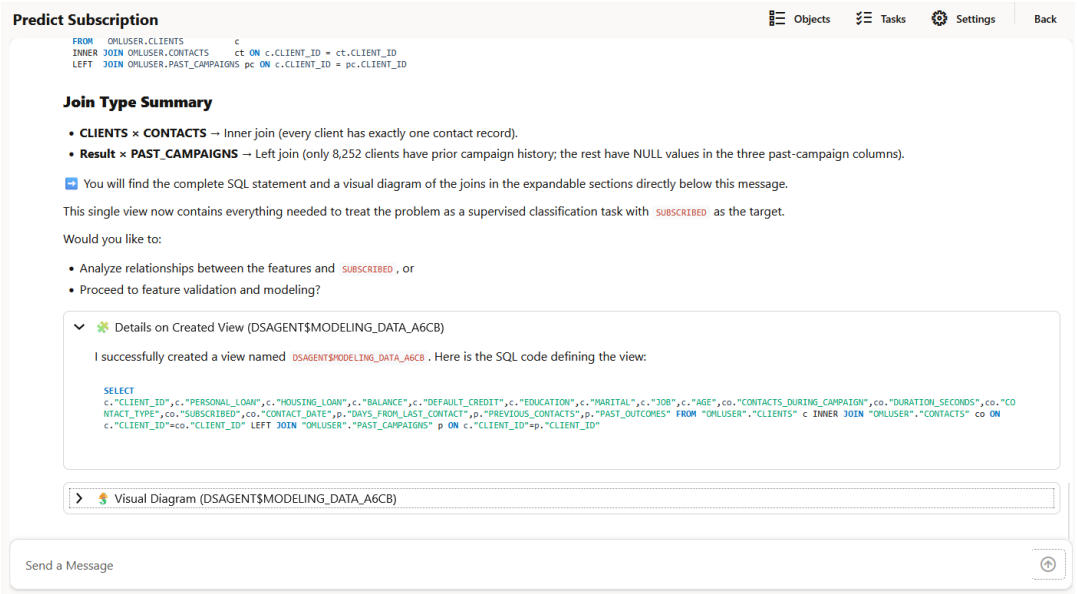
- Provides a summary of the view
- Explains the exact logic behind the join used

Figure 7-25 Prompt 4 and response



- Provides a summary of the join type
- Provides the SQL code and a visual diagram for the view.

Figure 7-26 Response 4 (concluded)



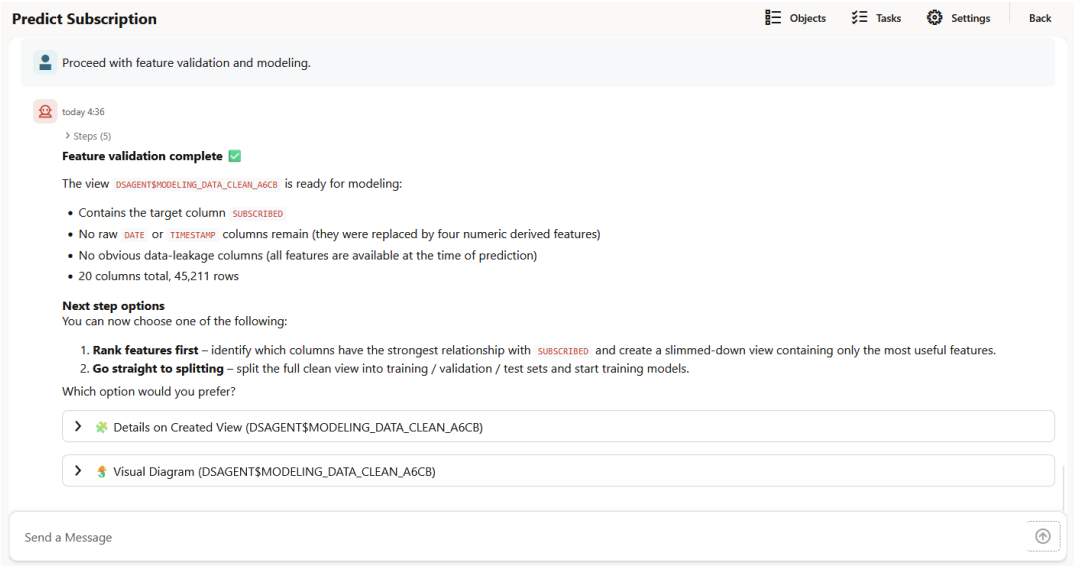
5. Data Science Agent performs feature validation and modeling

Prompt 5: `Proceed with feature validation and modeling.`

Agent's response:

- Data Science Agent creates a new database view `DSAGENT$MODELING_DATA_CLEAN_A6CB`
- Makes this view `DSAGENT$MODELING_DATA_CLEAN_A6CB` ready for modeling.
- Provides the SQL code and a visual diagram for the view.
- Provides the next step options

Figure 7-27 Prompt 5 and response



6. Data Science Agent performs data split, model training and model evaluation

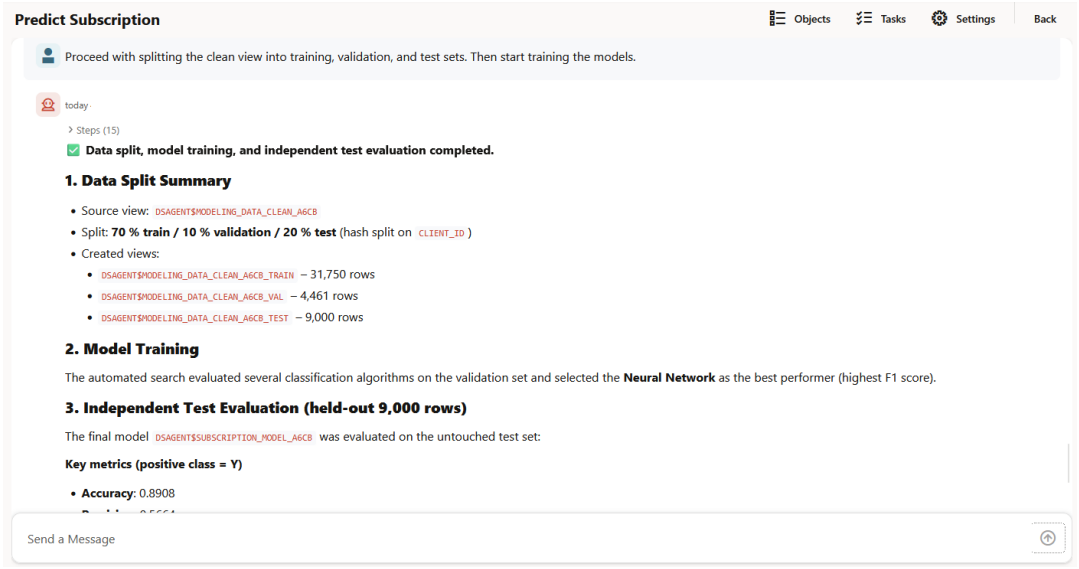
Data Science Agent uses the view `DSAGENT$MODELING_DATA_CLEAN_A6CB` for data split. It splits the data into 70% training, 10% validation and 20% test.

Prompt 6: `Proceed with splitting the clean view into training, validation, and test sets. Then start training the models.`

Agent's response:

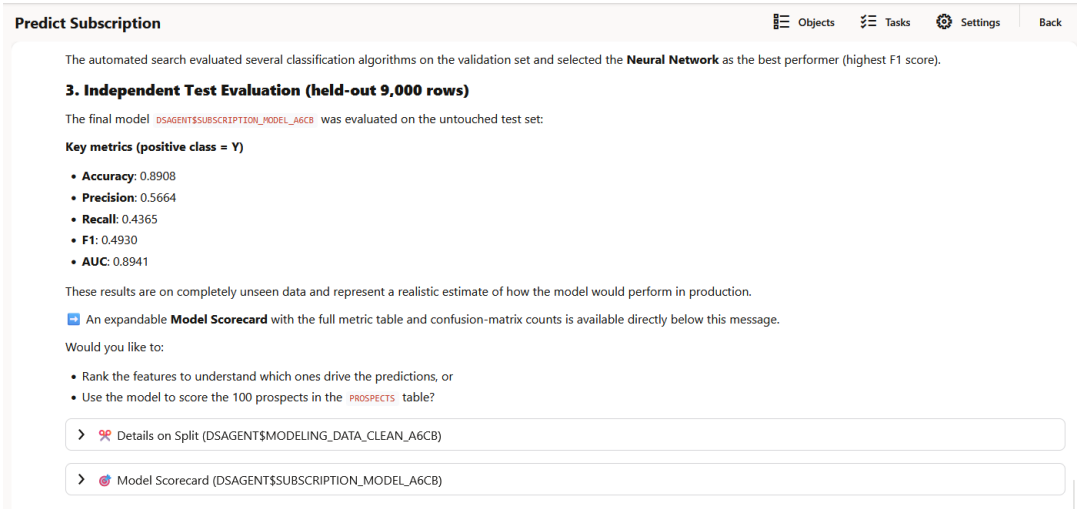
- Splits the data and creates the following views `DSAGENT$MODELING_DATA_CLEAN_A6CB_TRAIN` with 31,750 rows, `DSAGENT$MODELING_DATA_CLEAN_A6CB_VAL` with 4,461 rows, and `DSAGENT$MODELING_DATA_CLEAN_A6CB_TEST` with 9,000 rows.
- Provides a summary of the data split. The agent splits the data into 70% training, 10% validation and 20% test.
- Conducts model training and uses Neural Network as the best algorithm for this machine learning problem

Figure 7-28 Prompt 6 and response



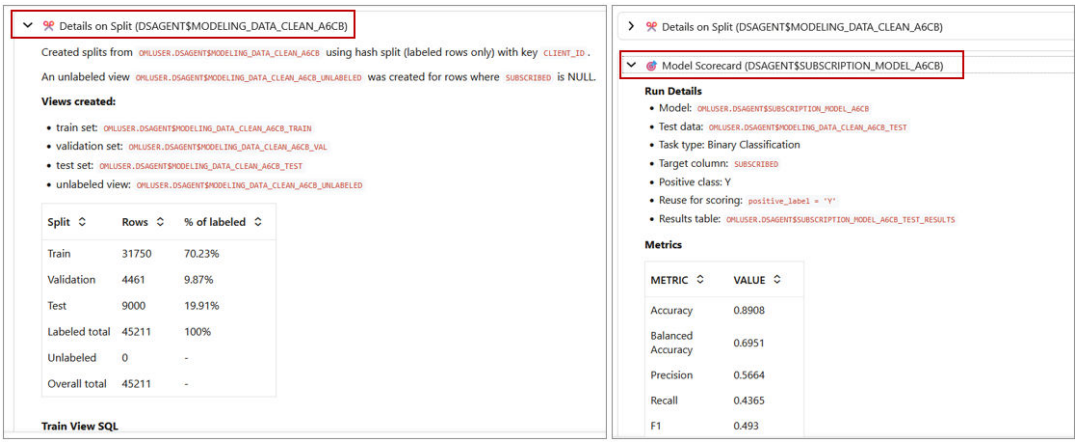
- Builds and evaluates the final model `DSAGENT$SUBSCRIPTION_MODEL_A6CB`

Figure 7-29 Response 6 (contd.)



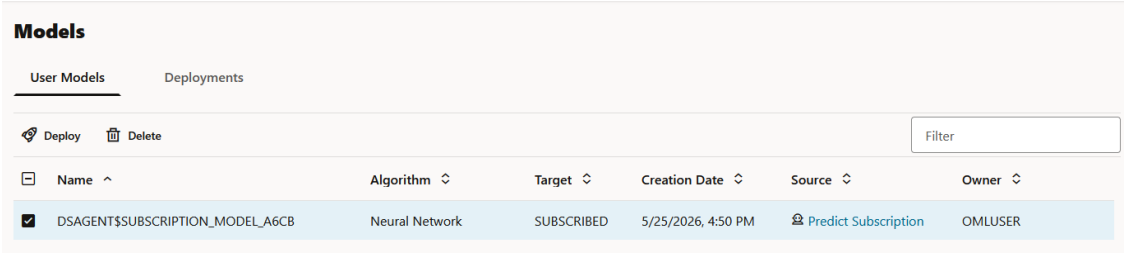
- Provides details on data split done on `OMLUSER.DSAGENT$MODELING_DATA_CLEAN_A6CB`
- Shows the scorecard for the model `OMLUSER.DSAGENT$SUBSCRIPTION_MODEL_A6CB`. It contains the metrics and binary confusion matrix.

Figure 7-30 Response 6 (concluded)



The final model `OMLUSER.DSAGENT$SUBSCRIPTION_MODEL_A6CB` is also listed on the Models page, as shown in the screenshot here:

Figure 7-31 DSAGENT$SUBSCRIPTION_MODEL_A6CB model listed on the Models page



7. Data Science Agent performs scoring to predict likelihood of subscription

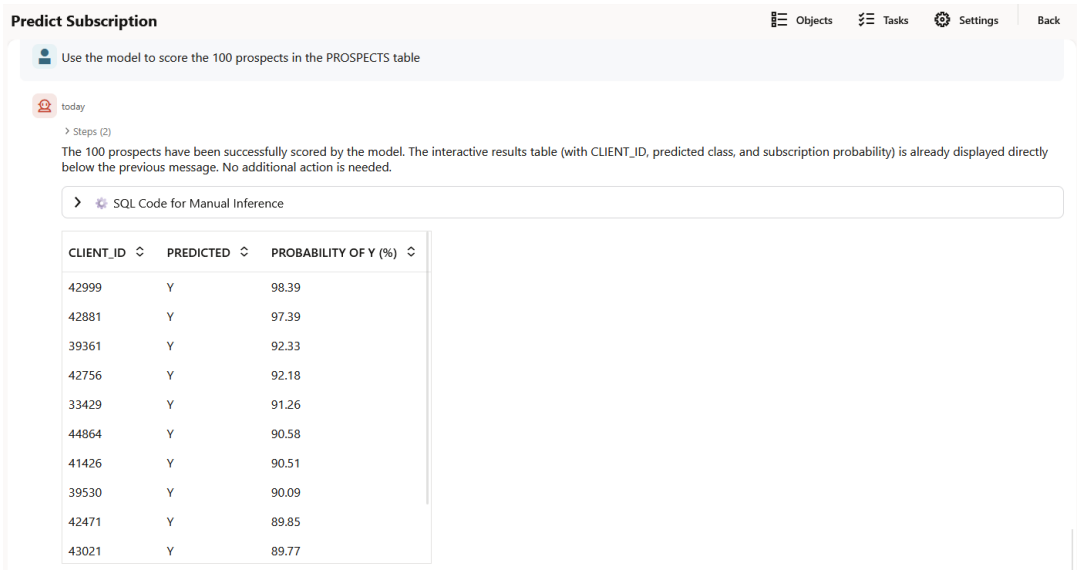
Data Science Agent uses the model `DSAGENT$SUBSCRIPTION_MODEL_A6CB` to predict the likelihood of subscription.

Prompt 7: `Use the model to score the 100 prospects in the PROSPECTS table`

Agent's response: Data Science Agent performs scoring on 100 prospective clients in the PROSPECTS table.

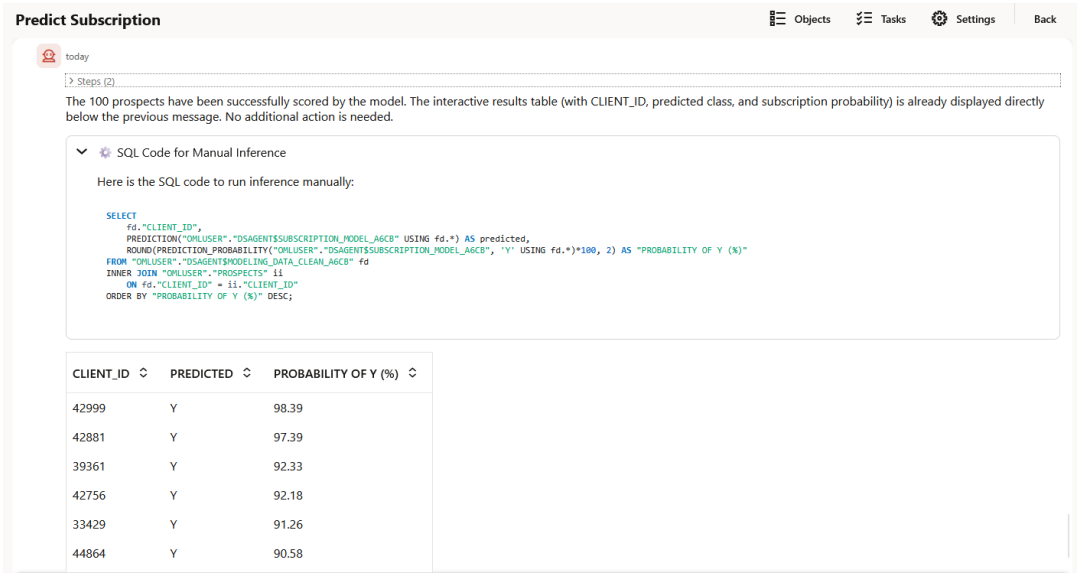
- Predicts the probability of subscription and presents the prediction data in a table.

Figure 7-32 Prompt 7 and response



- Provides the SQL query to run the inference manually

Figure 7-33 Prompt 7 response (concluded)



7.13 Known Issues in Data Science Agent

Learn about the issues you may encounter when using Data Science Agent in Oracle Machine Learning UI on Oracle Autonomous AI Database and their possible workarounds.

Topics:

- Objects present in user schema are not listed

7.13.1 Objects present in user schema are not listed

Issue: Due to a known UI issue, objects in your own schema may not appear in the **Objects** field in the Object Management dialog when **Show other schemas** is turned off. The affected objects types include tables, views, and machine learning models.

Expected behavior: In the Object Management dialog, the user is expected to see the objects —tables, views, and machine learning models present in their own schema when **Show other schemas** is turned off.

Workaround: To display these objects, turn on **Show other schemas**. Follow these steps to view and add objects present in your own schema and in other schemas:

1. In a Data Science Agent Conversation, open the **Settings** pane and click + on the **Associated Objects** field.
2. In the Object Management dialog, turn on **Show other schemas** to view all the objects present in your schema as well as in other schemas.
3. Select the objects and click **OK.** The selected objects appear in the Associated Objects field in the **Settings** pane and are available to the conversation.

8
Data Monitoring

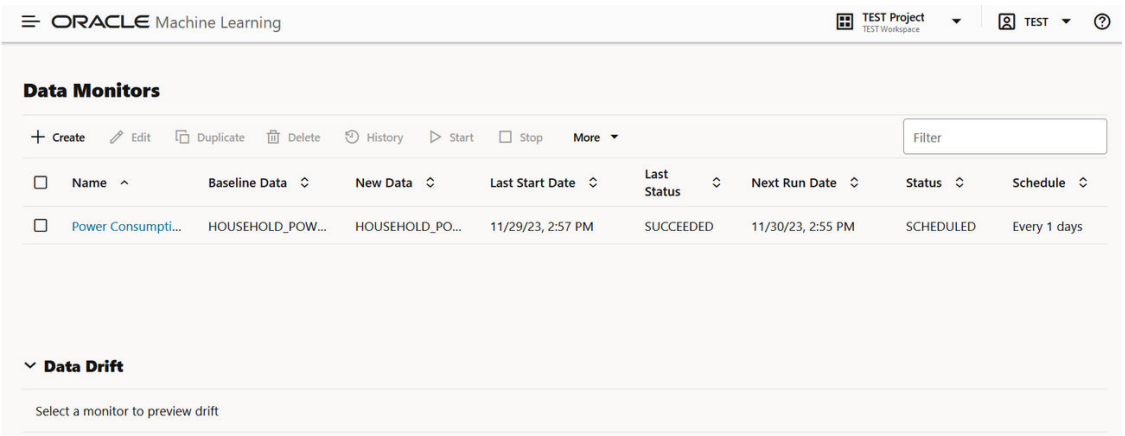
Data Monitoring evaluates how your data evolves over time. It helps you with insights on trends and multivariate dependencies in the data. It also gives you an early warning about data drift.

Data drift occurs when data diverges from the original baseline data over time. Data drift can happen for a variety of reasons, such as a changing business environment, evolving user behavior and interest, data modifications from third-party sources, data quality issues, or issues with upstream data processing pipelines.

The key to accurately interpret your models and to ensure that the models are able to solve business problems is to understand how data evolves over time. Data monitoring is complementary to successful model monitoring, as understanding the changes in data is critical in understanding the changes in the efficacy of the models. The ability to quickly and reliably detect changes in the statistical properties of your data ensures that your machine learning models are able to meet business objectives.

You can monitor your data using the data monitoring functionality of Oracle Machine Learning User Interface. To monitor your data, click on the Cloud menu on the Oracle Machine Learning UI home page, click **Monitoring** and then click **Data** to open the Data Monitors page. On the Data Monitors page, you can perform the following tasks:

Figure 8-1 Data Monitors page



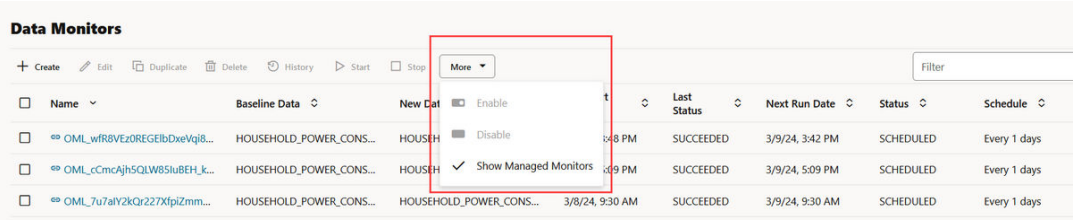
- **Create:** Create a data monitor.

  > **Note**
  >
  > The supported data types for data monitoring are NUMERIC and CATEGORICAL.

- **Edit:** Select a data monitor and click **Edit** to edit a data monitor.
- **Duplicate:** Select a data monitor and click **Duplicate** to create a copy of the monitor.

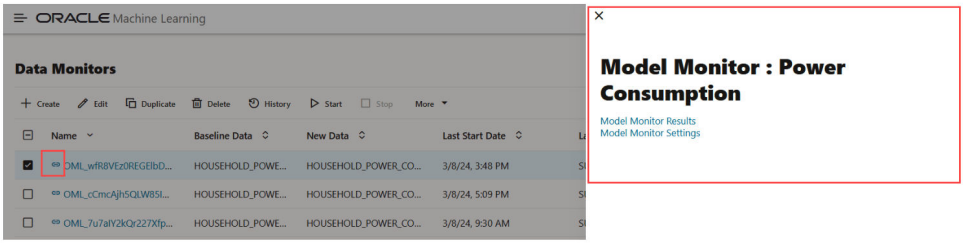
- **Delete:** Select a data monitor and click **Delete** to delete a data monitor.
- **History:** Select a data monitor and click **History** to view the runtime details. Click **Back to Monitors** to go back to the Data Monitoring page.
- **Start:** Start a data monitor.
- **Stop:** Stop a data monitor that is running.
- **More:** Click **More** for additional options to:

Figure 8-2 More option under Data Monitors



- **Enable:** Select a data monitor and click **Enable** to enable a monitor that is disabled. By default, a data monitor is enabled. The status is displayed as `SCHEDULED`.
- **Disable:** Select a data monitor and click **Disable** to disable a data monitor. The status is displayed as `DISABLED`.
- **Show Managed Monitors:** Click this option to view the data monitors that are created and managed by OML Services REST API and Model Monitors in Oracle Machine Learning UI. The data monitors that are managed by these two components have a system generated name, and are indicated by specific icons against its name.
  - Click on the link icon against a managed data monitor name to view the details of the associated model monitor. The associated model monitor details are displayed on a separate pane that slides in. The slide-in pane displays the model monitor name with links to view the model monitor results and settings. Clicking on the link icon also displays the data drift details on the lower pane of the Data Monitors page. Click on the X on the top left corner to close the pane.

Figure 8-3 Data Monitors page displaying the associated model monitor results and settings



In this example, the slide-in pane displays the details of the model monitor Power Consumption. On the slide-in pane:

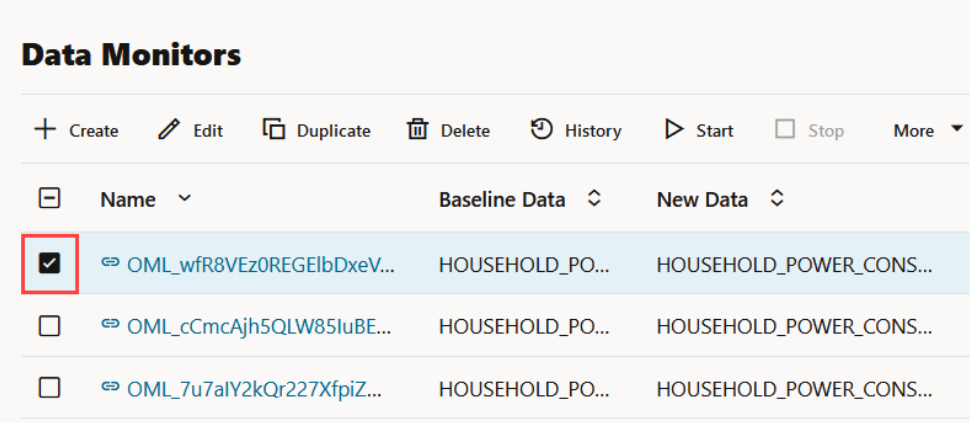
- Click **Model Monitor Results** to view the results computed by the model monitor — settings, models, model drift, metric, and prediction statistics. Click

Monitors to return to the Data Monitors page. See View Model Monitor Results.

* Click **Model Monitor Settings** to view and edit the settings, details, and models monitored by the model monitor on the Edit Model Monitor page. Click **Cancel** to return to the Data Monitors page. Click **Save** to save any changes.

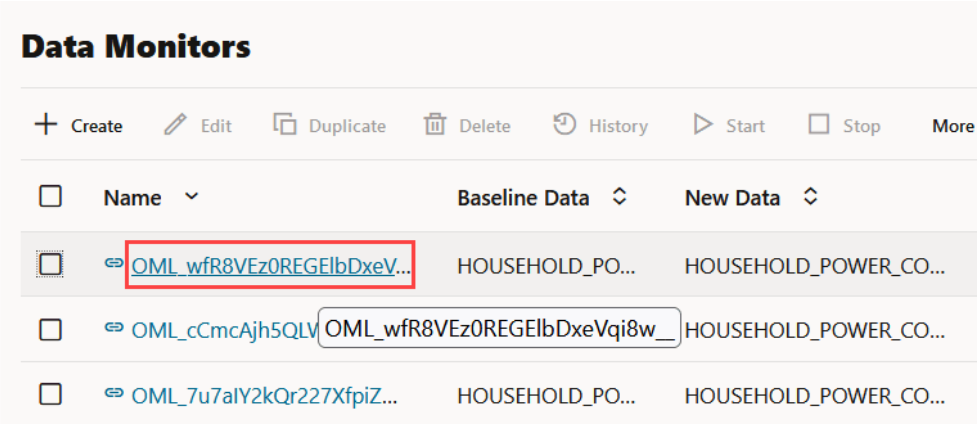
* Click on the check box against the data monitor name to view the data drift values on the lower pane.

Figure 8-4 Select a managed data monitor



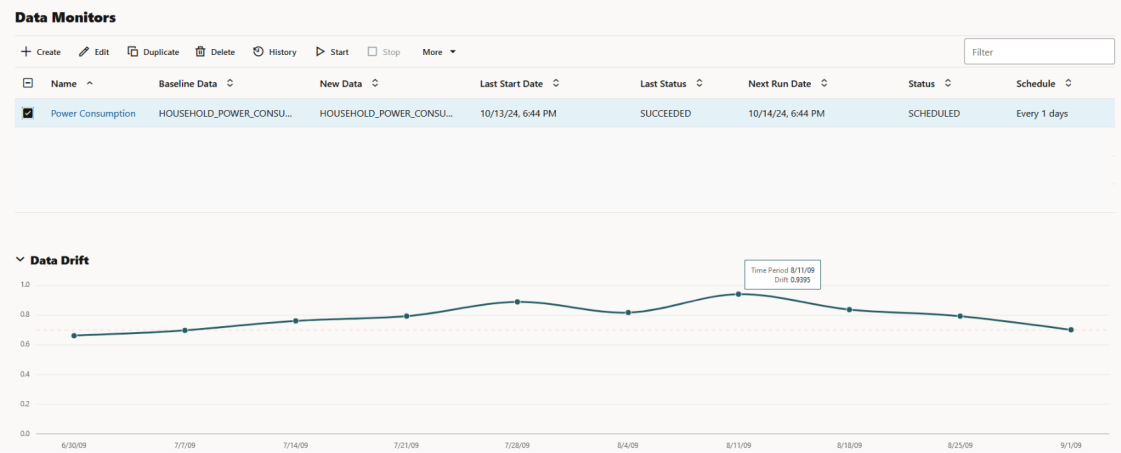
* Click on the data monitor name to view the details of the data monitor — settings, data drift values and monitored features.

Figure 8-5 Data monitor click



The Data Monitors page displays the information about the selected monitor: Monitor name, Baseline Data, New Data, Last Start Date, Last Status, Next Run Data, Status, and Schedule. The page also displays the data drift, if the data monitor has run successfully. To view data drift:

Figure 8-6 Data Drift preview on Data Monitors page



Select a data monitor that has run successfully, as shown in the screenshot. On the lower pane, the data drift of the selected monitor is displayed. The X axis depicts the analysis period, and the Y axis depicts the data drift values. The horizontal dotted line is the threshold value, and the line depicts the drift value for each point in time for the analysis period. Hover your mouse over the line to view the drift values. For more information on this example, see View Data Monitor Results.

- Create a Data Monitor
  Data Monitoring allows you to detect data drift over time and the potentially negative impact on the performance of your machine learning models. On the Data Monitor page, you can create, run, and track data monitors and the results.
- View Data Monitor Results
  The Data Monitor Results page displays the information on the selected data monitor that have run successfully, along with data drift details for each monitored feature.
- View History
  The History page displays the runtime details of data monitors.

Related Topics

- View History
  The History page displays the runtime details of data monitors.

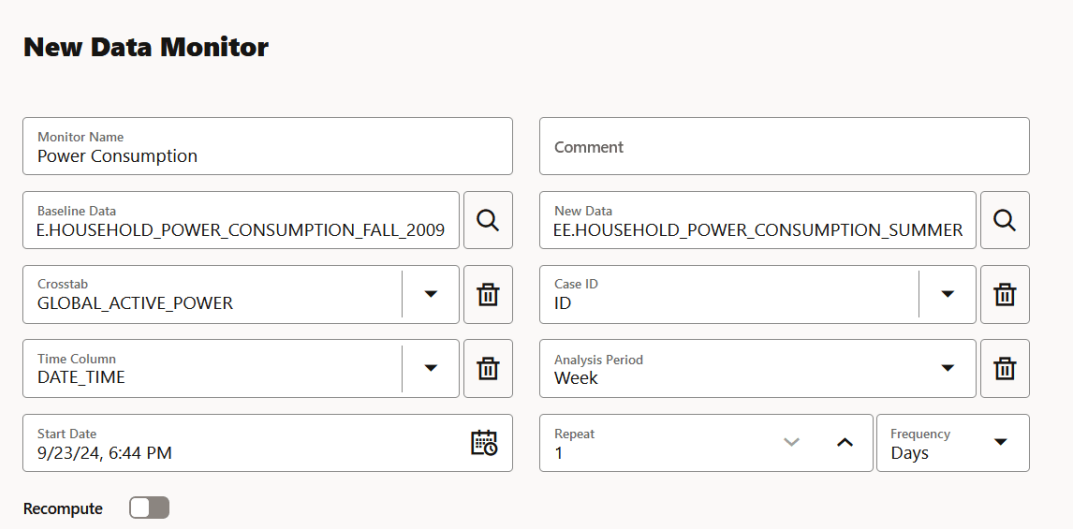
8.1 Create a Data Monitor

Data Monitoring allows you to detect data drift over time and the potentially negative impact on the performance of your machine learning models. On the Data Monitor page, you can create, run, and track data monitors and the results.

To create a data monitor:

1. On the Oracle Machine Learning UI left navigation menu, expand **Monitoring** and then click **Data** to open the Data Monitoring page.
2. On the Data Monitoring page, click **Create** to open the New Data Monitor page.
3. On the New Data Monitor page, enter the following details:

Figure 8-7 New Data Monitor



a. **Monitor Name:** Enter a name for the data monitor.

b. **Comments:** Enter comments. This is an optional field.

c. **Baseline Data:** This is a table or view that contains baseline data to monitor. Click the search icon to open the **Select Table** dialog. Here, select a schema, and then a table.

> **Note**
>
> The supported data types for data monitoring are `NUMBER`, `BINARY_DOUBLE`, `FLOAT`, `BINARY_FLOAT`, `VARCHAR2`, `CHAR`, `NCHAR`, and `NVARCHAR2` with length `<=4000`.

d. **New Data:** This is a table or view with new data to be compared against the baseline data. Click the search icon to open the **Select Table** dialog. Select a schema, and then a table.

> **Note**
>
> The supported data types for data monitoring are `NUMBER`, `BINARY_DOUBLE`, `FLOAT`, `BINARY_FLOAT`, `VARCHAR2`, `CHAR`, `NCHAR`, and `NVARCHAR2` with length `<=4000`.

e. **Crosstab:** Select an attribute from the drop-down list. This attribute in the baseline and new data acts as an anchor or target for bi-variate analysis of your data.

> **Note**
>
> The target column in supervised problems can be passed as an anchor column in this field. For unsupervised problems, it can be any column of interest. However, it will be application specific.

f. **Case ID:** This is an optional field. Enter a case identifier for the baseline and new data to improve the repeatability of the results.

g. **Time Column:** This is the name of a column storing time information in the New Data table or view. Select the time column from the drop-down list.

> **Note**
>
> If the Time Column is blank, the entire New Data is treated as one period.

h. **Analysis Period:** This is the length of time for which data monitoring is performed on the New Data. Select the analysis period for data monitoring. The options are `Day`, `Week`, `Month`, `Year`.

i. **Start Date:** This is the start date of your data monitor schedule. If you do not provide a start date, the current date will be used as the start date.

j. **Repeat:** This value defines the number of times the data monitor run will be repeated for the frequency defined. Enter a number between 1 and 99. For example, if you enter `2` in the **Repeat** field here, and `Minutes` in the **Frequency** field, then the data monitor will run every 2 minutes.

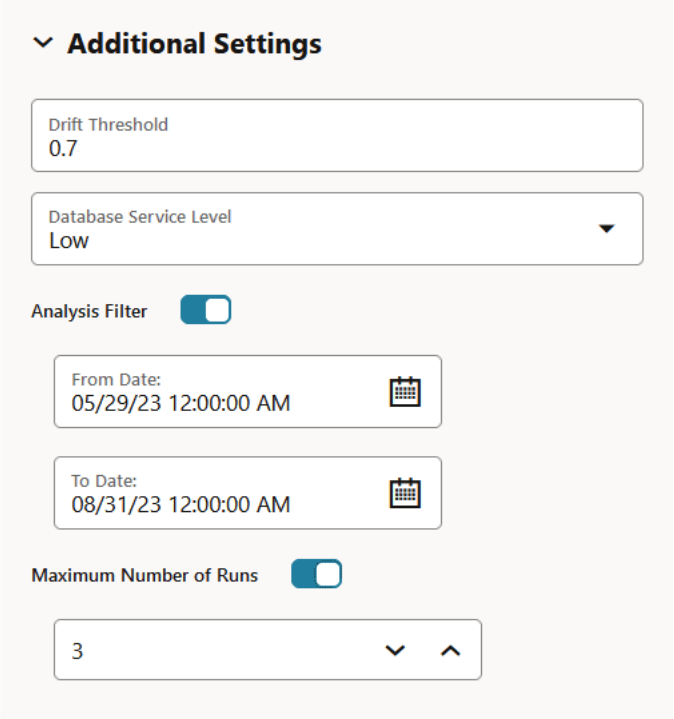
k. **Frequency:** This value determines how frequently the data monitor run will be performed on the New Data. Select a frequency for data monitoring. The options are Minutes, Hours, Days, Weeks, Months. For example, if you select `Minutes` in the **Frequency** field, `2` in the **Repeat** field, and `5/30/23` in the **Start Date** field, then as per the schedule, the data monitor will run from 5/30/23 every 2 minutes.

4. Click **Recompute:** Select this option to recompute the analysis for the already computed time period. By default, Recompute is disabled.

   - When enabled, the data drift analysis is performed for the time period specified in the Start Date field and the end time. The analysis will overwrite the already existing results for the specified time period. This means that the analysis will be computed for the time period with new data other than the current data. New analysis results may overlap with the existing results depending on the selected frequency.
   - When disabled, the data for the time period that is present in the results table will be retained as is. Only the new data for the most recent time period will be considered for analysis, and the results will be added to the results table.

5. Click **Additional Settings** to expand this section and provide advanced settings for your data monitor:

Figure 8-8 Data Monitoring Additional Settings



a. **Drift Threshold:** Drift captures the relative change in performance between the baseline data and the new data period. Based on your specific machine learning problem, set the threshold value for your data drift detection. The default is `0.7`.

> **Note**
>
> You may adjust the threshold value depending on your use case. Increasing the value will generate fewer alerts, while decreasing the value will generate more alerts.

- A drift above this threshold indicates a significant change in your data. Exceeding the threshold indicates that rebuilding and redeploying your model may be necessary.
- A drift below this threshold indicates that there are insufficient changes in the data to warrant further investigation or action.

b. **Database Service Level:** This is the Autonomous AI Database service levels - `Low`, `Medium`, `High` and `GPU`. The default is `Low`.

- Service level `Medium` provides more resources to the data monitor run compared to `Low`.
- Service level `High` provides more resources to the data monitor run compared to `Medium`.

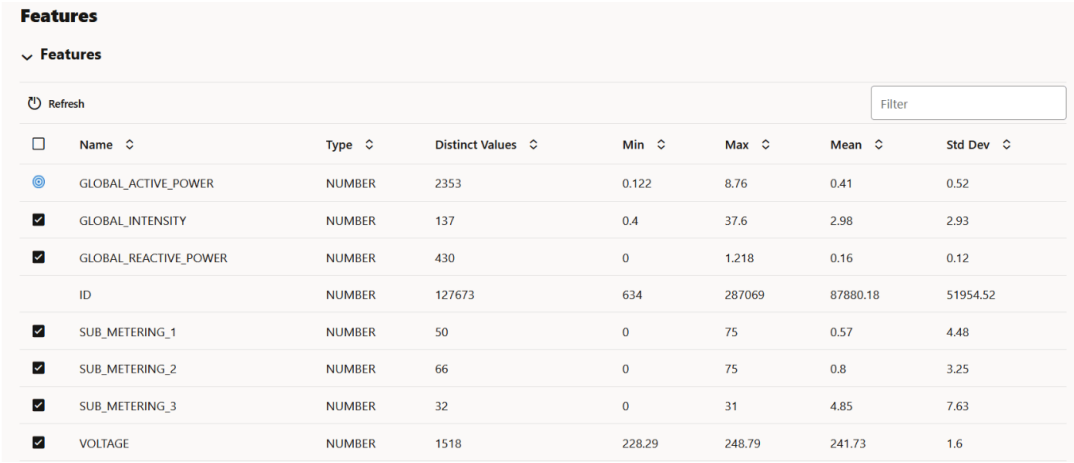
c. **Analysis Filter:** Enable this option if you want the data monitoring analysis for a specific time period. Move the slider to the right to enable it, and then select a date in **From Date** and **To Date** fields respectively. By default, this field is disabled.

- **From Date:** This is the start date or timestamp of monitoring in New Data. It assumes the existence of a time column in the table. This is a mandatory field if you use the Analysis Filter option.
- **To Date:** This is the end date or timestamp of monitoring in the New Data. It assumes the existence of a time column in the table. This is a mandatory field if you use the Analysis Filter option.

d. **Maximum Number of Runs:** This is the maximum number of times the data monitor can be run according to this schedule. The default is 3.

6. The Features grid displays the list of features to monitor. Here, you can select or deselect features to include or exclude from monitoring. By default, all features are selected. Feature statistics are provided if the selected data is a table and has RDBMS statistics automatically gathered by Autonomous AI Database. Oracle Machine Learning Services calculates the statistics on the first run for both, tables and views, and the computations are displayed here after the first run. The statistics are updated by subsequent runs.

Figure 8-9 Features grid in Data Monitor



Name	Type	Distinct Values	Min	Max	Mean	Std Dev
GLOBAL_ACTIVE_POWER	NUMBER	2353	0.122	8.76	0.41	0.52
GLOBAL_INTENSITY	NUMBER	137	0.4	37.6	2.98	2.93
GLOBAL_REACTIVE_POWER	NUMBER	430	0	1.218	0.16	0.12
ID	NUMBER	127673	634	287069	87880.18	51954.52
SUB_METERING_1	NUMBER	50	0	75	0.57	4.48
SUB_METERING_2	NUMBER	66	0	75	0.8	3.25
SUB_METERING_3	NUMBER	32	0	31	4.85	7.63
VOLTAGE	NUMBER	1518	228.29	248.79	241.73	1.6

> **Note**
>
> The **Case ID** and **Cross-Tab** columns cannot be selected.

7. Click **Save.** This completes the task of creating your data monitor.

> **Note**
>
> You must now go to the Data Monitoring page, select the data monitor and click **Start** to begin data monitoring.

After the data monitor runs successfully, select the monitor on the **Data Monitoring** page to view the data drift and other details of the data monitor. See Data Monitoring for more information.

8.2 View Data Monitor Results

The Data Monitor Results page displays the information on the selected data monitor that have run successfully, along with data drift details for each monitored feature.

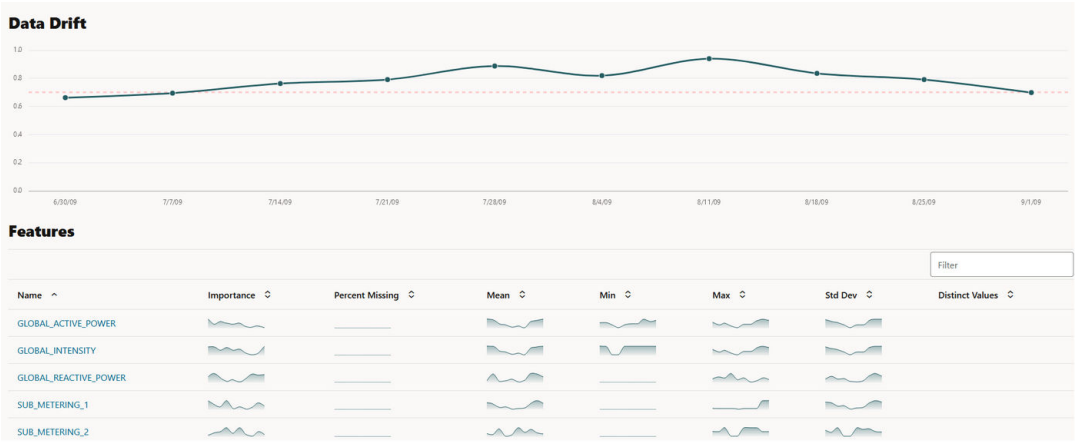
On the Data Monitors page, click on a data monitor that has run successfully. In this example, the data monitor **Power Consumption** is selected. The results of the data monitor is displayed on the Data Monitor Results page which comprises these sections:

- **Settings:** The Settings section displays the data monitor settings. Click on the arrow against **Settings** to expand this section. You have the option to edit the data monitor settings by clicking **Edit** on the top right corner of the page. In this screenshot, the settings for the data monitor **Power Consumption** is seen.

Figure 8-10 Settings section on the Data Monitor Results page

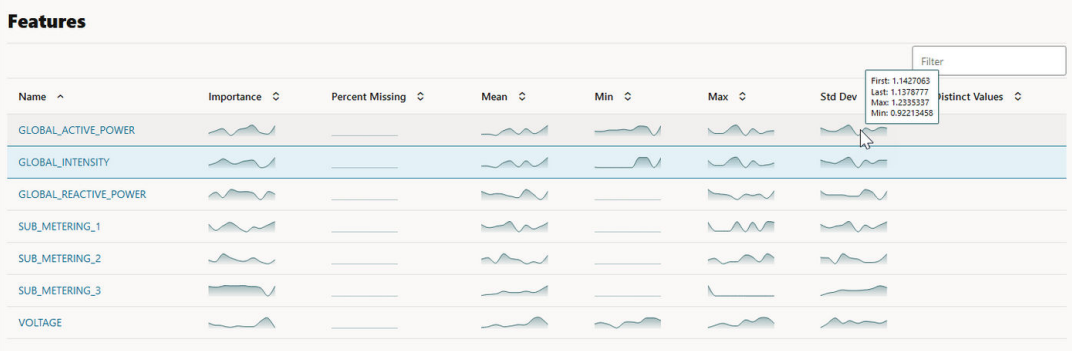
- **Drift:** The Drift section displays the details of data drift for each monitored feature. In this example, the data monitor **Power consumption** data monitor is selected. The X axis depicts the analysis period, and the Y axis depicts the data drift values. The horizontal dotted line is the threshold value, and the line depicts the drift value for each point in time for the analysis period. Hover your mouse over the line to view the drift values.

Figure 8-11 Data Drift section on the Data Monitor Results page



- **Features:** The Features section displays the monitored features along with the computed statistics.

Figure 8-12 Features section on the Data Monitor Results page



The value in the **Importance** column indicates how impactful the feature has been on data drift over a specified time period.

For numerical data, the following statistics are computed:

- Mean
- Standard Deviation
- Range (Minimum, Maximum)
- Number of nulls

For categorical data, the following statistics are computed:

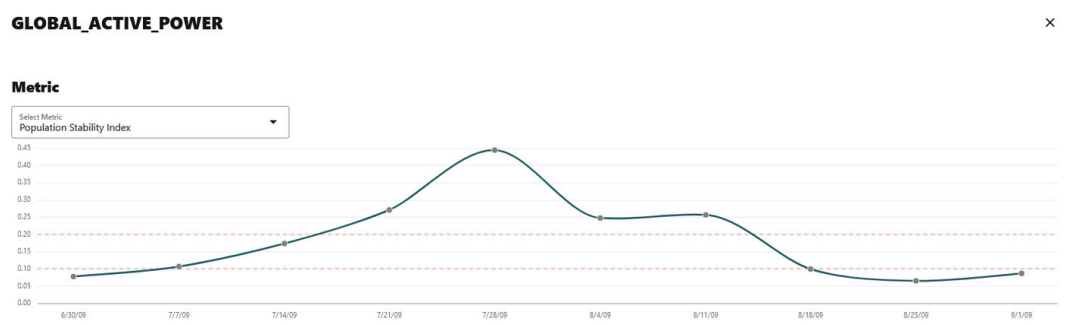
- Number of unique values
- Number of nulls

For each monitored feature, hover your mouse to view the following additional details, as shown in the screenshot here.

- First: This is the first value of the computed statistics for the analysis period.
- Last: This is the last value of the computed statistics for the analysis period.
- Max: This is the highest value of the computed statistics for the analysis period.
- Min: This is the lowest value of the computed statistics for the analysis period.

- Click on any monitored feature in the **Features** section to view the Metric, Statistics, Distribution, and Distribution with Crosstab Column, as shown in the screenshot here. In the screenshot here, the Population Stability Index is shown for the feature GLOBAL_REACTIVE_POWER.

Figure 8-13 Population Stability Index



The computations include:

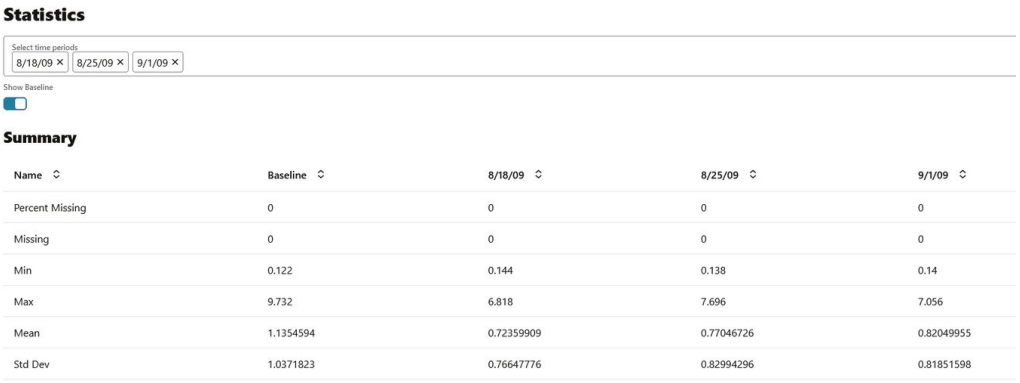
- **Metric:** The following metrics are computed:
  - Population Stability Index (PSI): This is a measure of how much a population has shifted over time or between two different samples of a population in a single number. The two distributions are binned into buckets, and PSI compares the percents of items in each of the buckets. PSI is computed as
    `PSI = sum((Actual_% - Expected_%) x ln (Actual_% / Expected_%))`

    The interpretation of PSI value is:
    - `PSI < 0.1` implies no significant population change
    - `0.1 <= PSI < 0.2` implies moderate population change
    - `PSI >= 0.2` implies significant population change
  - Jenson Shannon Distance (JSD): This is a measure of the similarity between two probability distributions. JSD is the square root of the Jensen-Shannon Divergence which is related to the Kullbach-Leibler Divergence (KLD). JSD is computed as:
    `SD(P || Q)= sqrt(0.5 x KLD(P || M) + 0.5 x KLD(Q || M))`

    Where, P and Q are the 2 distributions, `M = 0.5 x (P + Q), KLD(P || M) = sum(Pi x ln(Pi / Mi)), and KLD(Q || M) = sum(Qi x ln(Qi / Mi))`

    The value of JSD ranges between 0 and 1.
  - Crosstab Population Stability Index: This is the PSI for two variables.
  - Crosstab Jenson Shannon Distance: This is the JSD for two variables.
- **Statistics:** You can view statistics for up to 3 selected periods. Data drift is quantified using these statistical computations.

Figure 8-14 Statistics



Name	Baseline	8/18/09	8/25/09	9/1/09
Percent Missing	0	0	0	0
Missing	0	0	0	0
Min	0.122	0.144	0.138	0.14
Max	9.732	6.818	7.696	7.056
Mean	1.1354594	0.72359909	0.77046726	0.82049955
Std Dev	1.0371823	0.76647776	0.82994296	0.81851598

For numerical data, the following statistics are computed:

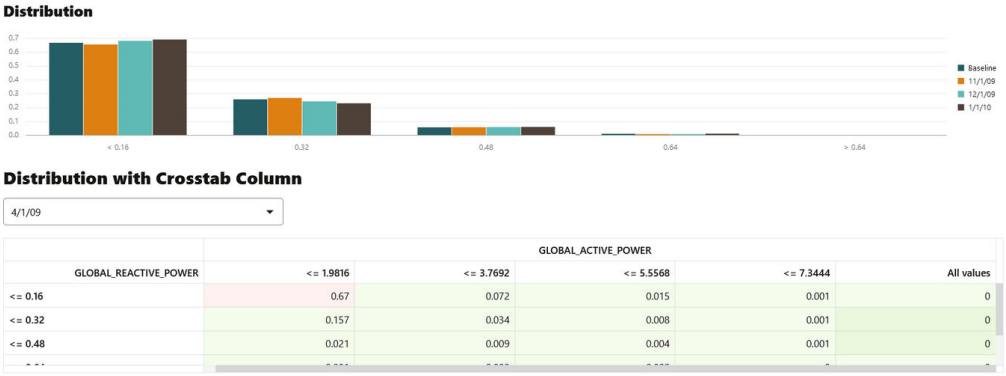
- Mean
- Standard Deviation
- Range (Minimum, Maximum)
- Number of nulls

For categorical data, the following statistics are computed:

* Number of unique values
* Number of nulls

– **Distribution:** The feature distribution chart with legend displays bins of feature for selected periods and the baseline (optional).

Figure 8-15 Distribution Chart and Distribution with Crosstab column



– **Distribution with Crosstab Column:** The heat map indicates the density of distribution for the selected crosstab and the feature column. Red denotes highest density.

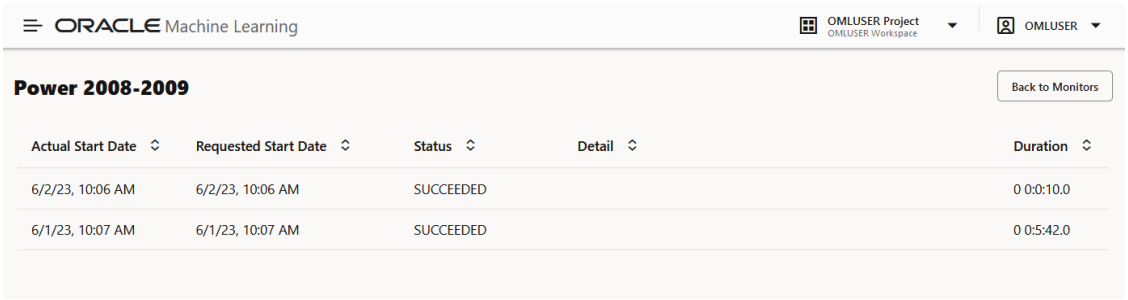
> **Note**
>
> In data drift monitoring, `nulls` are are tracked separately as `number_of_missing_values`.

8.3 View History

The History page displays the runtime details of data monitors.

Select a data monitor and click **History** to view the runtime details. The history page displays the following information about the data monitor runtime:

Figure 8-16 Data Monitor History page



- **Actual Start Date:** This is the date when the data monitor actually started.

- **Requested Start Date:** This is the date entered in the `Start Date` field while creating the data monitor.
- **Status:** The statuses are `SUCCEEDED` and `FAILED`.
- **Detail:** If a data monitor fails, the details are listed here.
- **Duration:** This is the time taken to run the data monitor.

Click **Back to Monitors** to go back to the Data Monitoring page.

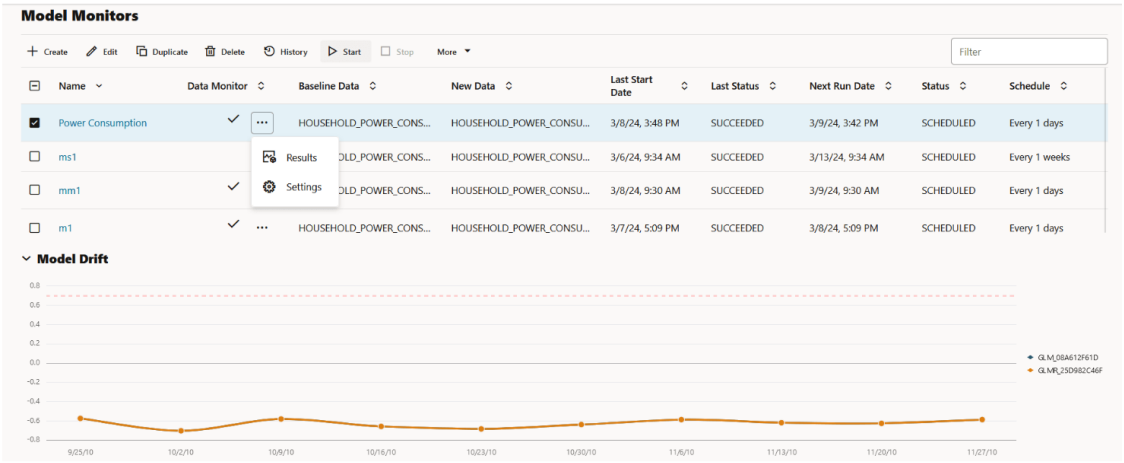
9

Model Monitoring

Model monitoring allows you to monitor the quality of model predictions over time and helps you with insights on the causes of model quality issues.

The Model Monitor page allows you to create, run, and track model monitors and results. This page lists the model monitors. You can preview the model drift by selecting a single model monitor of interest, as shown in the screenshot below. In this screenshot, the monitor **Power Consumption** is selected. On the lower pane of the Model Monitors page, the **Model Drift** for the selected monitor is displayed. The X axis depicts the analysis period, and the Y axis depicts the model drift values. The horizontal red dotted line is the threshold value. The line depicts the drift value for each point in time for the analysis period. Hover your mouse over the line to view the drift values.

Figure 9-1 Model Monitors page



You can perform the following tasks on the Model Monitors page:

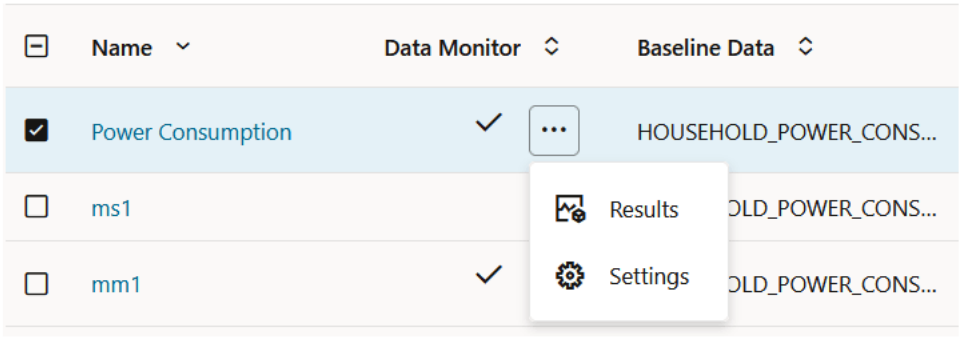
- **Create:** Create a model monitor.
- **Edit:** Select a model monitor and click **Edit** to edit the monitor.
- **Duplicate:** Select a model monitor and click **Duplicate** to create a copy of the monitor.
- **Delete:** Select a model monitor and click **Delete** to delete the model monitor.
- **History:** Select a model monitor and click **History** to view the runtime details of the model monitor. If there is a data monitor job associated with the model monitor, the data monitor runtime details are displayed on the lower pane. Click **Back to Monitors** to go back to the **Model Monitoring** page.
- **Start:** Start a model monitor.
- **Stop:** Stop a model monitor that is running.
- **More:** Select a model monitor, click **More** and then click:

- **Enable:** Enable a model monitor schedule. By default, the model monitor is enabled. The status is displayed as `SCHEDULED`.
- **Disable:** Disable a model monitor schedule. The status is displayed as `DISABLED`.
- **Show Managed Monitors:** Click this option to view the models created and managed by OML Services REST API.

The Model Monitors page displays the following information about the model monitors:

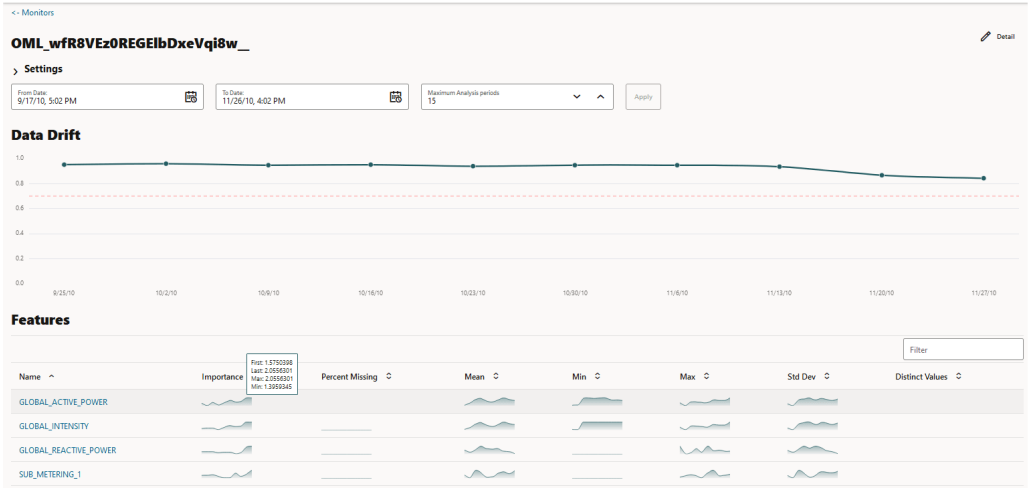
- **Name:** This is the name of the model monitor.
- **Data Monitor:** The check mark in this column indicates that there is a data monitor job associated with the model monitor. Click on the three dots to view:

Figure 9-2 View Results and Settings of Data Monitor



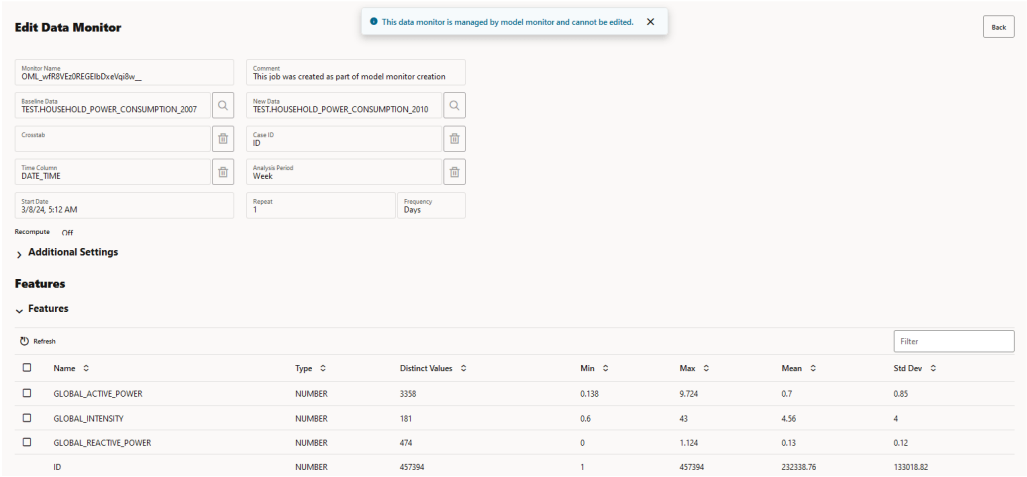
- **Results:** Displays the data monitoring results computed by the data monitor job. The data monitor name is auto-generated along with the prefix OML. The data drift and monitored features along with the computed statistics are displayed in view-only mode on a separate page. Click **Monitors** on the top left corner to return to the Model Monitors page. Click **Details** on the top right corner to view the settings of the data monitor.

Figure 9-3 Data Monitor Results



– **Settings:** Displays the data monitor settings, additional settings, and the monitored features in view-only mode. Click **Back** on the top right corner of the page to return to the Model Monitors page.

Figure 9-4 Data Monitor Settings



- **Baseline Data:** This is a table or view that contains baseline data to monitor.
- **New Data:** This is a table or view with new data to be compared against the baseline data.
- **Last Start Date:** This is the date on which the model monitor was last started.
- **Last Status:** This is the latest status of the model monitor. The statuses are SUCCEEDED, FAILED, RUNNING, SCHEDULED.
- **Next Run Date:** This is the date on which the next run is scheduled.
- **Status:** This is the current status of the model monitor.
- **Schedule:** This is the frequency of the monitoring job, that is, how frequently the model monitor is set to run on the New Data.
- Create a Model Monitor
  A model monitor helps you monitor several compatible models, and compute the model drift chart. Compatible models refer to those models that are trained on the same target and mining function. The model drift chart consists of multiple series of data drift points, one for each monitored model.
- View Model Monitor Results
  The Model Monitor Results page displays the monitoring results of each model that are being monitored. By clicking on any model that has run successfully, you can view the detailed analysis of each model such as model drift, model metrics, prediction statistics, feature impact, prediction distribution and predictive versus drift importance for each feature. The predictive impact versus drift importance for each monitored feature is computed only if data monitoring is enabled.
- View Model History
  The History page displays the runtime details of the model monitors.

9.1 Create a Model Monitor

A model monitor helps you monitor several compatible models, and compute the model drift chart. Compatible models refer to those models that are trained on the same target and mining

function. The model drift chart consists of multiple series of data drift points, one for each monitored model.

A model monitor can optionally monitor data to provide additional insight. This additional insight is the Drift Feature Importance versus Predictive Feature Impact chart which is generated when you select the Monitor Data option while creating the model monitor.

This topic discusses hows how to create a model monitor. The example uses the *Individual household electricity consumption* dataset which includes various consumption metrics of a household for the year 2009. The data is split into seasons - SPRING, SUMMER, FALL, and WINTER. The goal is to understand if and how household consumption has changed over the seasons. The example shows how to track the effects of data drifts on model predictive accuracy.

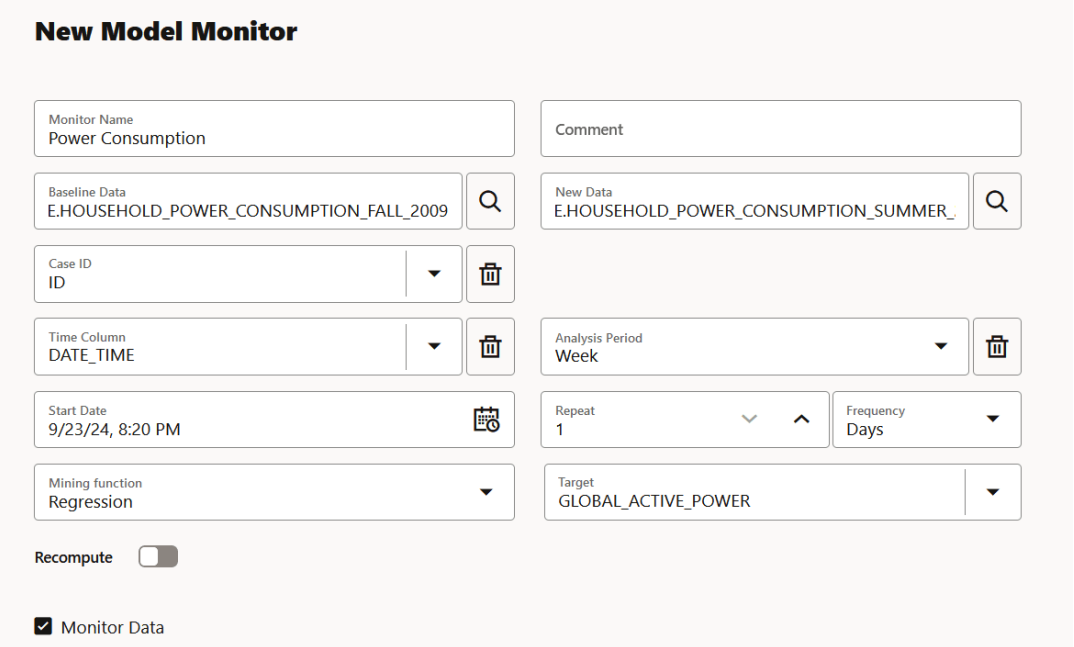
The dataset comprises the following columns:

- `DATE_TIME`: Contains the date and time related information in `dd:mm:yyyy:hh:mm:ss` format.
- `GLOBAL_ACTIVE_POWER`: This is the household global minute-averaged active power (in kilowatt).
- `GLOBAL_REACTIVE_POWER`: This is the household global minute-averaged reactive power (in kilowatt).
- `VOLTAGE`: This is the Minute-averaged voltage (in volt).
- `GLOBAL_INTENSITY`: This is the household global minute-averaged current intensity (in ampere).
- `SUB_METERING_1`: This is the energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen.
- `SUB_METERING_2`: This is the energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room.
- `SUB_METERING_3`: This is the energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to an electric water heater and air conditioner.

To create a model monitor:

1. On the Oracle Machine Learning UI left navigation menu, expand **Monitoring** and then click **Models** to open the Model Monitoring page. Alternatively, you can click on the Model Monitoring icon to open the Model Monitoring page.
2. On the Model Monitoring page, click **Create** to open the New Model Monitor page.
3. On the New Model Monitor page, enter the following details:

Figure 9-5 New Model Monitor page



a. **Monitor Name:** Enter a name for the model monitor. Here, the name Power Consumption is used.

b. **Comment:** Enter comments. This is an optional field.

c. **Baseline Data:** This is a table or view that contains baseline data to monitor. Click the search icon to open the **Select Table** dialog. Select a schema, and then a table. Here, the table containing the data for the year 2007 is selected.

d. **New Data:** This is a table or view with new data to be compared against the baseline data. Click the search icon to open the **Select Table** dialog. Select a schema, and then a table. Here, the table containing the data for the year 2009 is selected.

e. **Case ID:** This is an optional field. Enter a case identifier for the baseline and new data to improve the repeatability of the results.

f. **Time Column:** This is the name of a column storing time information in the New Data table or view. The `DATE_TIME` column is selected from the drop-down list.

> **Note**
>
> If the Time Column is blank, the entire New Data is treated as one period.

g. **Analysis Period:** This is the length of time for which model monitoring is performed on the New Data. Select the analysis period for model monitoring. The options are `Day`, `Week`, `Month`, `Year`.

h. **Start Date:** This is the start date of your model monitor schedule. If you do not provide a start date, the current date will be used as the start date.

i. **Repeat:** This value defines the number of times the model monitor run will be repeated for the frequency defined. Enter a number between 1 and 99. For example, if you enter 2 in the **Repeat** field here, and Minutes in the **Frequency** field, then the model monitor will run every 2 minutes.

j. **Frequency:** This value determines how frequently the model monitor run will be performed on the New Data. Select a frequency for model monitoring. The options are Minutes, Hours, Days, Weeks, Months. For example, if you select `Minutes` in the **Frequency** field, `2` in the **Repeat** field, and `5/30/23` in the **Start Date** field, then as per the schedule, the model monitor will run from `5/30/23` every `2` minutes.

k. **Mining Function:** The available mining functions are `Regression` and `Classification`. Select a function as applicable. In this example, `Regression` is selected.

l. **Target:** Select an attribute from the drop-down list. In this example, `GLOBAL_ACTIVE_POWER` is used as the target for regression models.

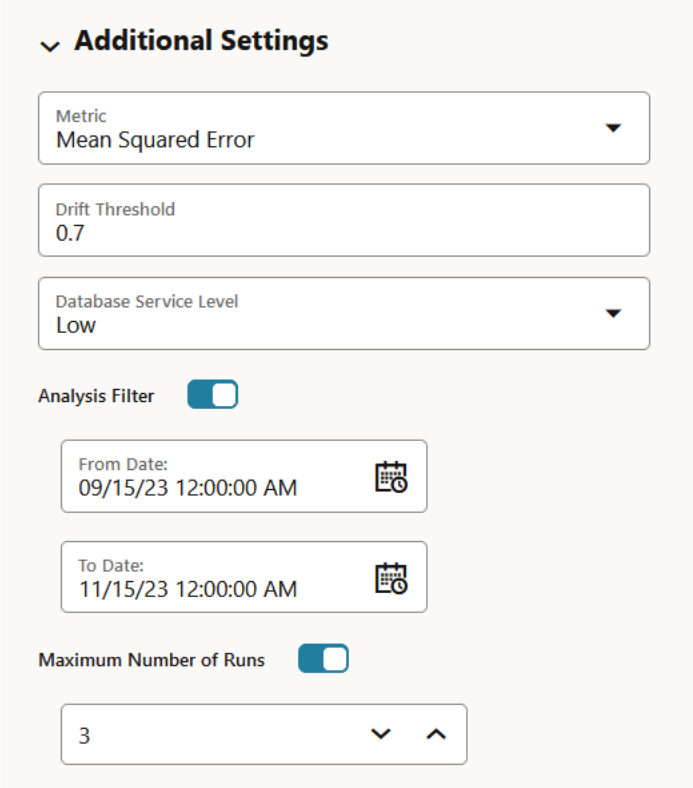
m. **Recompute:** Select this option to update the already computed periods. This means that only time periods not present in the output result table will be computed. By default, Recompute is disabled.

   - When enabled, the drift analysis is performed for the time period specified in the Start Date field and the end time. The analysis will overwrite the already existing results for the specified time period. This means that the analysis will be computed for the time period with new data other than the current data.
   - When disabled, the data for the time period that is present in the results table will be retained as is. Only the new data for the most recent time period will be considered for analysis, and the results will be added to the results table.

n. **Monitor Data:** Select this option to enable data monitoring for the specified data. When enabled, a data monitor is also created along with the model monitor to compute the Predictive Feature Impact versus Drift Feature Impact in the model specific results.

4. Click **Additional Settings** to expand this section and provide advanced settings for your model monitor:

Figure 9-6 Additional Settings section on the New Model Monitor page



a. **Metric:** Depending on the mining function selected in the **Mining Function** field in the Create Model Monitor page, the applicable metrics are listed. Click on the drop-down list to select a metric.

   For the mining function Classification, the metrics are:

   - Accuracy: Calculates the proportion of correctly classifies cases - both Positive and Negative. For example, if there are a total of TP (True Positives)+TN (True Negatives) correctly classified cases out of TP+TN+FP+FN (True Positives+True Negatives+False Positives+False Negatives) cases, then the formula is: `Accuracy = (TP+TN)/(TP+TN+FP+FN)`
   - Balanced Accuracy: Evaluates how good a binary classifier is. It is especially useful when the classes are imbalanced, that is, when one of the two classes appears a lot more often than the other. This often happens in many settings such as Anomaly Detection etc.
   - ROC AUC (Area under the ROC Curve): Provides an aggregate measure of discrimination regardless of the decision threshold. AUC - ROC curve is a performance measurement for the classification problems at various threshold settings.
   - Recall: Calculates the proportion of actual Positives that is correctly classified.
   - Precision: Calculates the proportion of predicted Positives that is True Positive.
   - F1 Score: Combines precision and recall into a single number. F1-score is computed using harmonic mean which is calculated by the formula: `F1-score = 2 × (precision × recall)/(precision + recall)`

   For multi-class classification, the metrics are:

- Accuracy
- Balanced Accuracy
- Macro_F1
- Macro_Precision
- Macro_Recall
- Weighted_F1
- Weighted_Precision
- Weighted_Recall

For Regression, the metrics are:

- R2: A statistical measure that calculates how close the data are to the fitted regression line. In general, the higher the value of R-squared, the better the model fits your data. The value of R2 is always between 0 to 1, where:
  - `0` indicates that the model explains none of the variability of the response data around its mean.
  - `1` indicates that the model explains all the variability of the response data around its mean.
- Mean Squared Error: This is the mean of the squared difference of predicted and true targets.
- Mean Absolute Error: This is the mean of the absolute difference of predicted and true targets.
- Median Absolute Error: This is the median of the absolute difference between predicted and true targets.

b. **Drift Threshold:** Drift captures the relative change in performance between the baseline data and the new data period. Based on your specific machine learning problem, set the threshold value for your model drift detection. The default is `0.7`.
   - A drift above this threshold indicates a significant change in model predictions. Exceeding the threshold indicates that rebuilding and redeploying your model may be necessary.
   - A drift below this threshold indicates that there are insufficient changes in the data to warrant further investigation or action.

c. **Database Service Level:** This is the service level for the job, which can be LOW, MEDIUM, HIGH, or GPU.

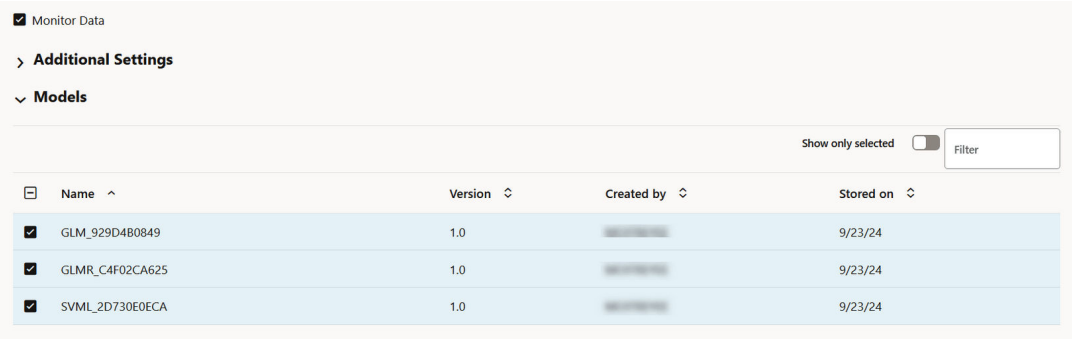
d. **Analysis Filter:** Enable this option if you want the model monitoring analysis for a specific time period. Move the slider to the right to enable it, and then select a date in **From Date** and **To Date** fields respectively. By default, this field is disabled.
   - **From Date:** This is the start date or timestamp of monitoring in New Data. It assumes the existence of a time column in the table. This is a mandatory field if you use the Analysis Filter option.
   - **To Date:** This is the end date or timestamp of monitoring in the New Data. It assumes the existence of a time column in the table. This is a mandatory field if you use the Analysis Filter option.

e. **Maximum Number of Runs:** This is the maximum number of times the model monitor can be run according to this schedule. The default is `3`.

5. In the **Models** section, select the model that you want to monitor and then click **Save** on the top right corner of the page. Once you provide a value in the **Mining Function** and **Target** fields, the list of models that have been deployed are obtained and is displayed here in the Models section. Models are deployed from the Models page, or from the AutoML Leaderboard. You can view the complete list of deployed models in the Deployments tab on the Models page. The deployed models are managed by OML Services.

> **Note**
>
> If you drop any models, you have to redeploy the models. Models are not schema based models, but models deployed to OML Services.

Figure 9-7 Models section on the New Model Monitor page



Once the model monitor is successfully created, it displays the message: `Model monitor has been created successfully.`

> **Note**
>
> You must now go to the Model Monitoring page, select the model monitor and click **Start** to begin model monitoring.

9.2 View Model Monitor Results

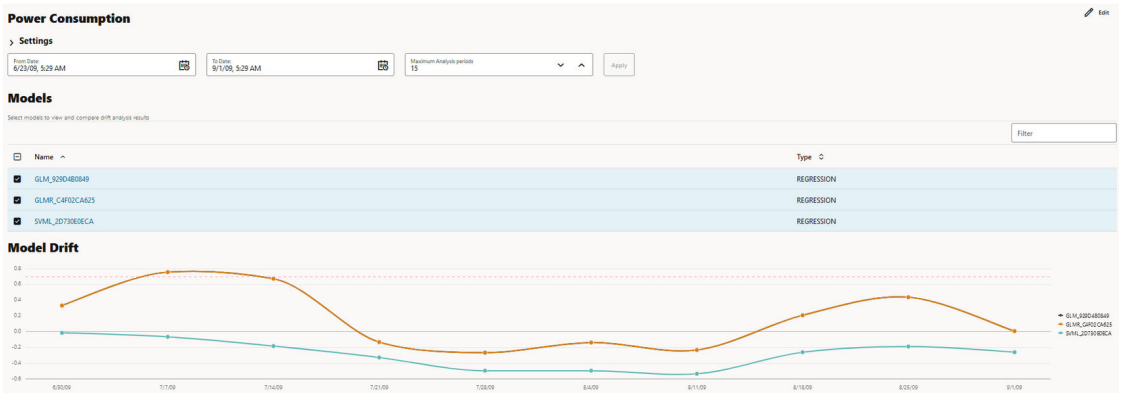
The Model Monitor Results page displays the monitoring results of each model that are being monitored. By clicking on any model that has run successfully, you can view the detailed analysis of each model such as model drift, model metrics, prediction statistics, feature impact, prediction distribution and predictive versus drift importance for each feature. The predictive impact versus drift importance for each monitored feature is computed only if data monitoring is enabled.

1. Model Monitor Results page

The Model Monitor Results page lists all the models that are monitored by the monitor. The name of the monitor is displayed at the top of the page. As seen in this screenshot, the monitor name **Power Consumption** is displayed at the top. The models GLM_929D4B0849, GLMR_C4F02CA625 and SVML_2D730E0ECA monitored by the **Power Consumption** monitor are listed in the Models section. By default, the details of all the monitored models are

displayed. You can choose to view the details of one monitor at a time by deselecting the other monitors.

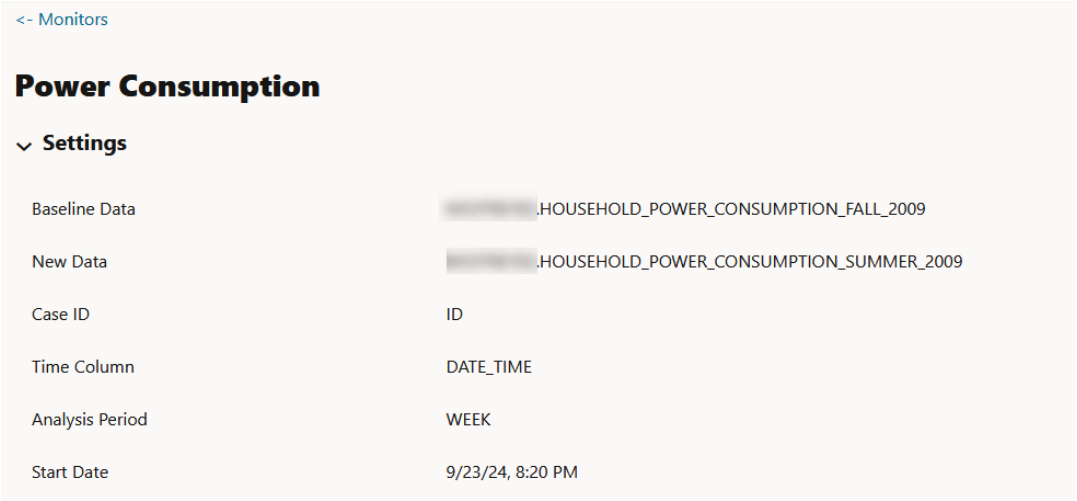
Figure 9-8 Model Monitor Results page



The Model Monitor Results page comprises these sections:

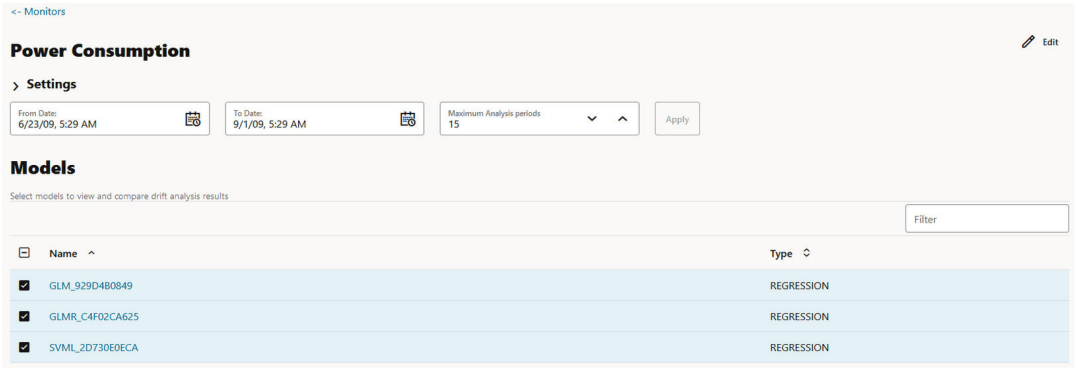
- **Settings:** The Settings section displays the model monitor settings. Click on the arrow against **Settings** to expand this section. You have the option to edit the model monitor settings by clicking **Edit** on the top right corner of the page.

Figure 9-9 Model Monitor Settings



- **Models:** The Models section lists all the models that are monitored by the monitor. In this example, the models GLM_929D4B0849, GLMR_C4F02CA625 and SVML_2D730E0ECA monitored by the **Power Consumption** monitor are listed.

Figure 9-10 Models on the Model Monitor Results page

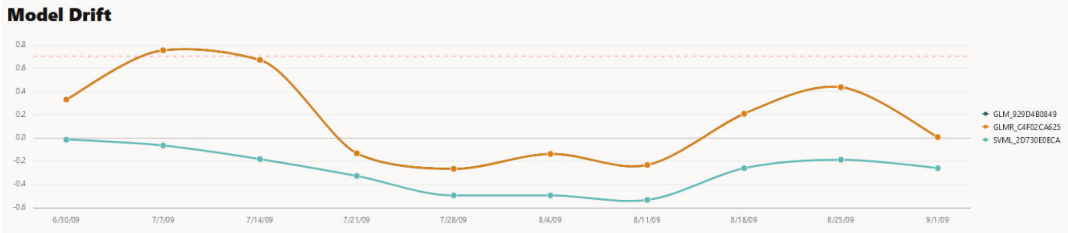


You can choose to view and compare the results of one or more monitored models by deselecting the ones that you want to exclude. You can also view the results of each feature of the model by clicking on the model. These results — **Feature Impact chart**, **Prediction Distribution**, and **Predictive Impact versus Drift Importance chart** are displayed on a separate pane that slides in.

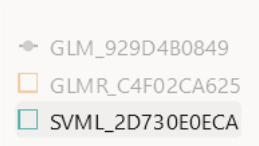
> **Note**
>
> The Predictive Impact versus Drift Importance chart is computed only if the **Monitor Data** option is selected while creating the model monitor.

- **Model Drift:** The Model Drift section is displayed just below the Models section. Model drift is the percentage change in the performance metric between the baseline period and the new period. A negative value indicates that the new period has a better performance metric which could happen due to noise.
  The X axis depicts the analysis period, and the Y axis depicts the drift values. The horizontal dotted line represents the drift threshold settings that each monitor gets by default. The default covers the typical use case. However, you can choose to customize it based on specific use case. The line depicts the drift value for each point in time for the analysis period. Hover your mouse over the line to view the drift values. A drift above the threshold indicates significant change in model predictions. Exceeding the threshold suggests rebuilding and redeploying your model may be necessary. If the drift is below the threshold, it indicates that there are insufficient changes in the data to warrant further investigation or action. That is, possibly rebuilding a machine learning model using this data.

Figure 9-11 Model Drift on the Models Monitor Results page

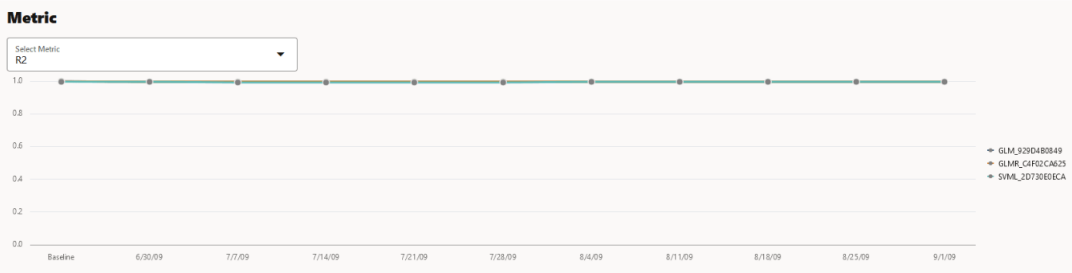


If you want to view the drift details of one model at a time, click on the model name on the right to select or deselect it, as shown here.



- **Metric:** This Metric section displays the computed metrics for the selected models. The computed metric is plotted along the y axis, and the time period is plotted along the x axis. In this example, the metric R2 or R-squared is displayed for all the three models. Hover your cursor on other points on the line to view the details of the computed metric. The value of R2 for all the models is equal to 1. Here, the value of R2 for all the three monitored models is 1. This indicates that all the three models are good fit for the data.

Figure 9-12 Metric



The computed metrics for **Regression** are:

- **R2:** A statistical measure that calculates how close the data are to the fitted regression line. In general, the higher the value of R-squared, the better the model fits your data. The value of R2 is always between 0 to 1, where:
  - `0` indicates that the model explains none of the variability of the response data around its mean.
  - `1` indicates that the model explains all the variability of the response data around its mean.
- **Mean Squared Error:** This is the mean of the squared difference of predicted and true targets.
- **Mean Absolute Error:** This is the mean of the absolute difference of predicted and true targets.
- **Median Absolute Error:** This is the median of the absolute difference between predicted and true targets.

The computed metrics for **Binary Classification** are:

- **Accuracy:** Calculates the proportion of correctly classifies cases - both Positive and Negative. For example, if there are a total of TP (True Positives)+TN (True Negatives) correctly classified cases out of TP+TN+FP+FN (True Positives+True Negatives+False Positives+False Negatives) cases, then the formula is:
  `Accuracy = (TP+TN)/(TP+TN+FP+FN)`
- **Balanced Accuracy:** Evaluates how good a binary classifier is. It is especially useful when the classes are imbalanced, that is, when one of the two classes appears a lot more often than the other. This often happens in many settings such as Anomaly Detection etc.

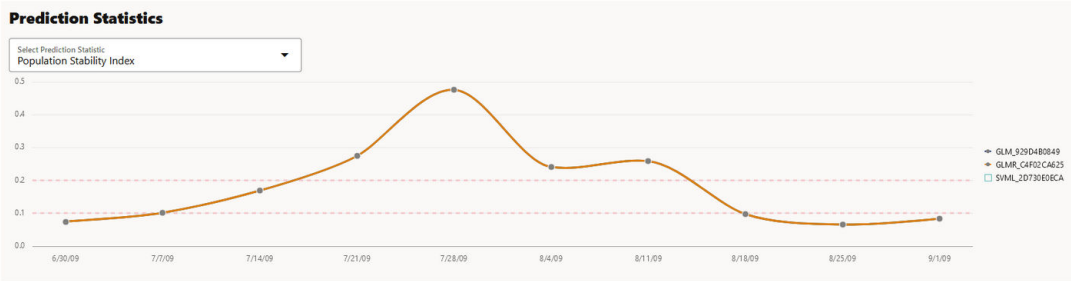
- **ROC AUC (Area under the ROC Curve):** Provides an aggregate measure of discrimination regardless of the decision threshold. AUC - ROC curve is a performance measurement for the classification problems at various threshold settings.
- **Recall:** Calculates the proportion of actual Positives that is correctly classified.
- **Precision:** Calculates the proportion of predicted Positives that is True Positive.
- **F1 Score:** Combines precision and recall into a single number. F1-score is computed using harmonic mean which is calculated by the formula:
  `F1-score = 2 × (precision × recall)/(precision + recall)`

The computed metrics for **Multi-class Classification** are:

- Accuracy
- Balanced Accuracy
- Macro_F1
- Macro_Precision
- Macro_Recall
- Weighted_F1
- Weighted_Precision
- Weighted_Recall

- **Prediction Statistics:** Scroll further down to view the **Prediction Statistics** section. The computed prediction statistic is plotted along the y axis, and the time period is plotted along the x axis. In this screenshot, the Population Stability Index for the Generalized Linear Model — Regression model GLMR_C4F02CA625 for 10/30/10 is displayed. Hover your cursor on other points on the line to view the computed metric.

Figure 9-13 Prediction Statistics



Click on the drop-down list to view all the prediction statistics. The statistics of the predictions of the model vary according to the type of model.

For **Regression**, the computed prediction statistics are:

- **Population Stability Index:** This is a measure of how much a population has shifted over time or between two different samples of a population in a single number. The two distributions are binned into buckets, and PSI compares the percents of items in each of the buckets. PSI is computed as
  `PSI = sum((Actual_% - Expected_%) x ln (Actual_% / Expected_%))`

  The interpretation of PSI value is:

  * `PSI < 0.1` implies no significant population change

    * `0.1 <= PSI < 0.2` implies moderate population change
    * `PSI >= 0.2` implies significant population change
  - **Min:** This is the lowest value of the computed statistics for the analysis period.
  - **Mean:** This is the average value of the computed statistics for the analysis period.
  - **Max:** This is the highest value of the computed statistics for the analysis period.
  - **Standard Deviation:** This is the value that shows how much variation from the mean exists.

  For **Binary Classification**, the computed prediction statistics are:

  - Population Stability Index
  - Mean
  - Min
  - Max
  - Standard Deviation
  - Bin Distribution of prediction probabilities
  - Class distribution

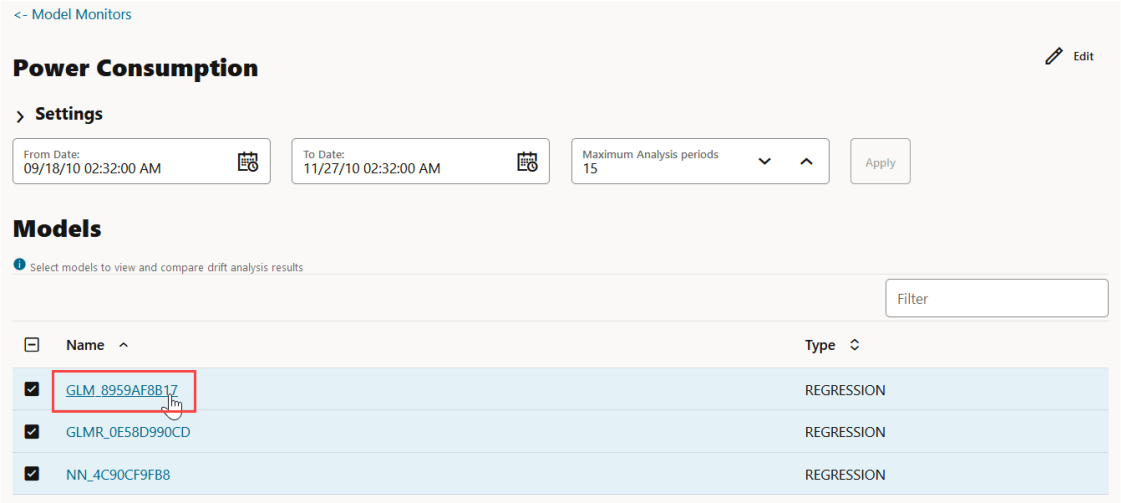
  For **Multi-class Classification**, the computed prediction statistics are:

  - Population Stability Index
  - Class Distribution

2. Model Monitor Details

You can view the details of each feature of the model by clicking on the model name. These details include the Feature Impact chart, Prediction Distribution and the Predictive Impact versus Drift Importance chart. In this example, the model GLM_8959AF817 is selected.

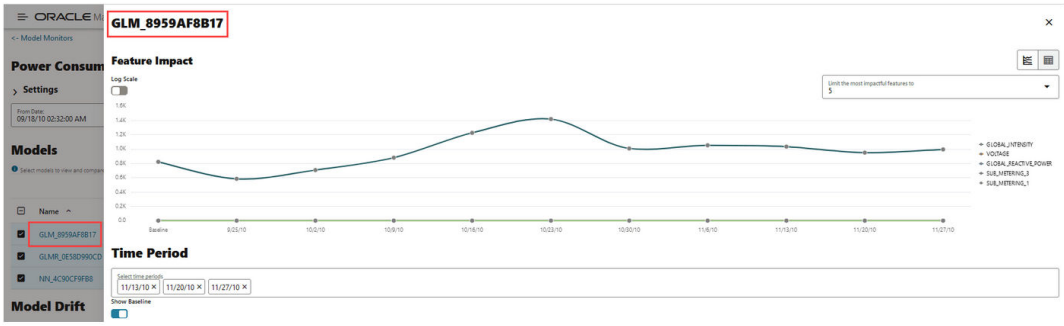
Figure 9-14 Model selection for details



The computed results are displayed on a separate pane that slides in. You can select up to 3 analysis periods comparison. You also have the option to hide or show the baseline details. The computed details of the model features are:

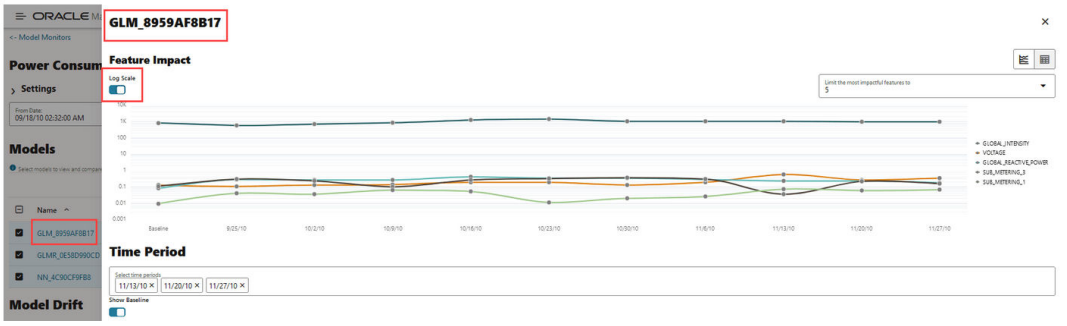
- **Feature Impact:** The Feature Impact chart computes the impact of each feature in the model for the specified time. The chart also the gives you the option to view the feature impact on a linear scale as well as on a logarithmic scale. Hover your mouse over the chart to view the details - Feature Name, Date, and Feature Impact.
  - Click **Log Scale** to view the feature impact computation on a logarithmic scale.
  - Click to view the feature impact computation in a line graph.
  - Click to view the feature impact computation in a table.
  - Click **Limit the most impactful features to** drop down list to select a value.

Figure 9-15 Viewing Feature Impact on a liner scale



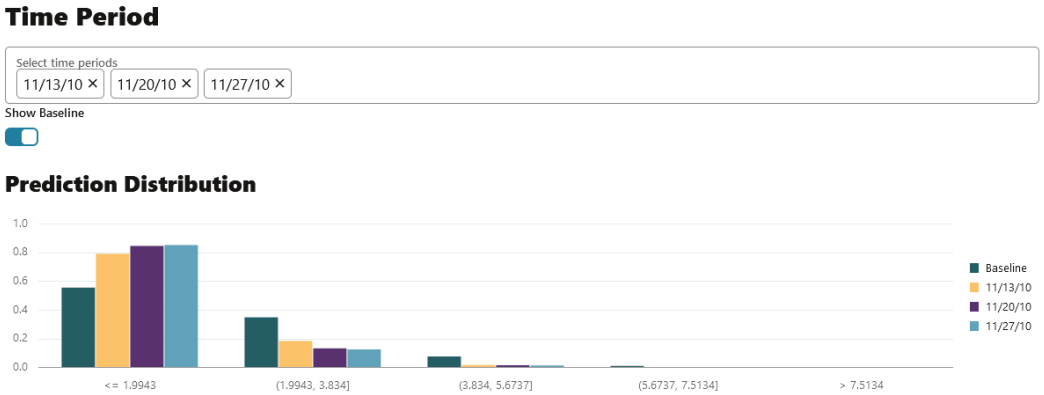
In this screenshot, the feature GLOBAL_INTENSITY, that is, the global minute-averaged current intensity of the household electric consumption is seen to have the maximum impact on the model GLM_8959AF817 as compared to the other features. Click on **Log Scale** to view feature impact computation on a logarithmic scale, as shown in the screenshot below. Click X on the top right corner of the pane to exit.

Figure 9-16 Viewing Feature Impact on a Logarithmic Scale



- **Prediction Distribution:** Scroll down to view the Prediction Distribution. Prediction Distribution is plotted for each analysis period. The Baseline data is displayed, if selected. The bins are plotted along X-axis, and the values are plotted along the Y-axis. Hover your mouse over each histogram to view the computed details. Click X on the top right corner of the pane to exit.

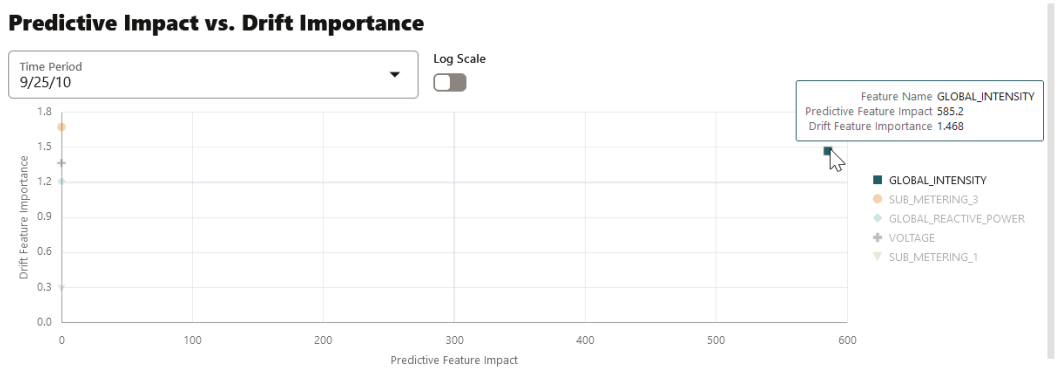
Figure 9-17 Prediction Distribution



- **Predictive Impact vs Drift Importance:** Scroll further down the pane to view the Prediction Impact versus Drift Importance chart. This chart helps in understanding how the most impactful features drift over time. Drift Feature Importance is plotted along the Y-axis and Prediction Feature Impact is plotted along the X-axis. Click X on the top right corner of the pane to exit.

> **Note**
>
> The Prediction Impact vs Drift Importance chart is computed only if you select the **Monitor Data** option while creating the model monitor.

Figure 9-18 Predictive Feature Impact versus Drift Importance



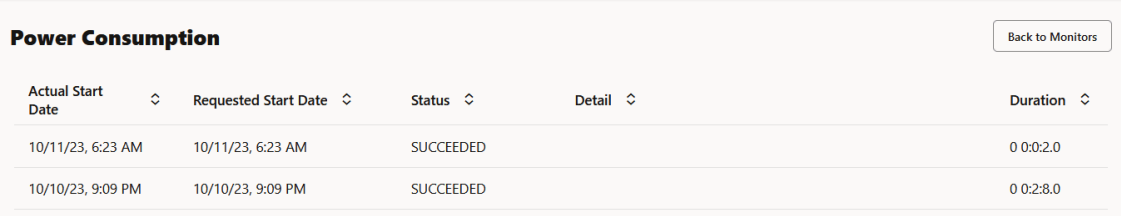
In this screenshot, you can see that the feature GLOBAL_INTENSITY has the maximum impact on the selected predictive model GLM_8959AF817 as compared to the other features - SUB_METERING_3, GLOBAL_REACTIVE_POWER, VOLTAGE, and SUB-METERING_1.

9.3 View Model History

The History page displays the runtime details of the model monitors.

Select a model monitor and click **History** to view the runtime details. The history page displays the following information about the model monitor runtime:

Figure 9-19 View Model History



- **Actual Start Date:** This is the date when the model monitor actually started.
- **Requested Start Date:** This is the date entered in the **Start Date** field while creating the model monitor.
- **Status:** The statuses are SUCCEEDED and FAILED.
- **Detail:** If a model monitor fails, the details are listed here.
- **Duration:** This is the time taken to run the model monitor.

Click **Back to Monitors** to go back to the Model Monitoring page.

10

Models

The Models page displays the user models and the list of deployed models. User Model lists the models in a user's schema, and Deployments lists the models deployed to Oracle Machine Learning Services.

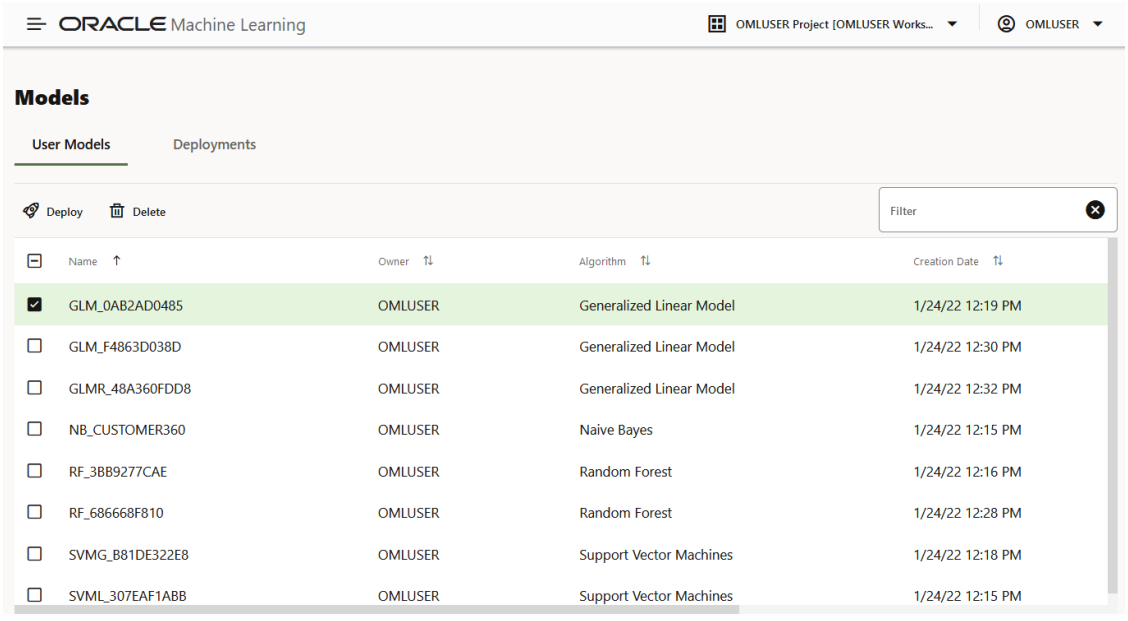
Under Models, the model information and model deployment are available under:

- **User Models:** Lists all the models that are created in a database schema. In the Models view, you can browse, view, deploy and delete models.
- **Deployments:** Lists all the deployed models. In the Deployments view, you can view the model metadata and the REST API URI of the deployed models.

User Models

In the User Models view, you can browse, view, and deploy models. The User Models view lists the models that are available in the database schema:

Figure 10-1 User Models



- Name: Displays the model name. Model names can be any valid database object name.
- Owner: Displays the user who built the model.
- Algorithm: Displays the name of the algorithm used.
- Creation Date: Displays the date on which the model is built.
- Target: Displays the prediction target selected when the experiment is created.

You can perform the following tasks:

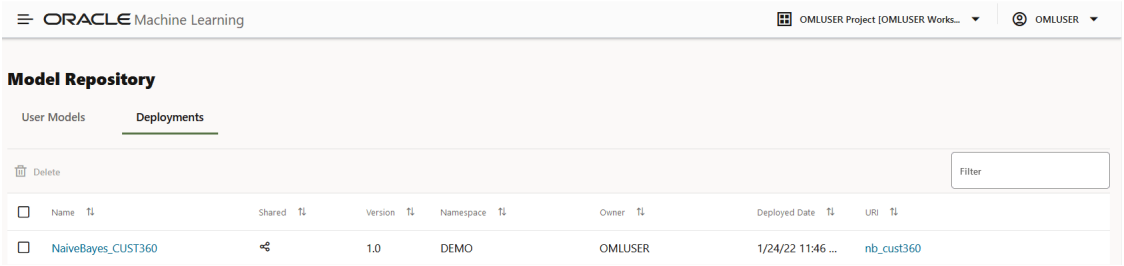
- **Deploy:** To deploy a model, select the model and click **Deploy.**
- **Delete:** To delete a model, select the model and click **Delete.**

Deployments

In the Deployments view, you can view the list of all the deployed models. Here, you can view the model metadata, view the REST API URI of the deployed models, and also delete any deployed model.

To delete a deployed model, select the model and click **Delete.**

Figure 10-2 Deployed Models



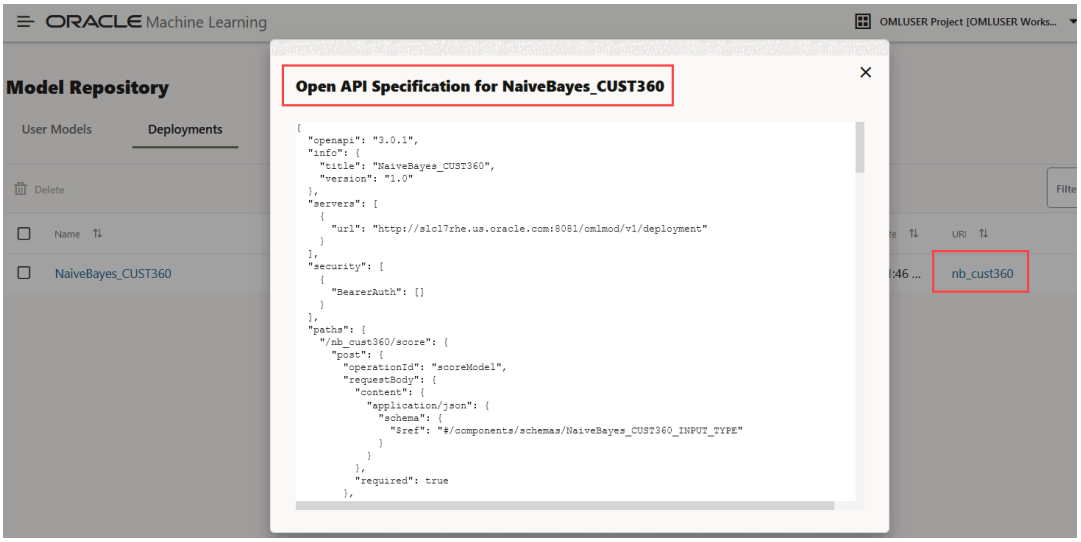
The following information are displayed for each deployed model:

- **Name:** The name of the deployed model.
- **Shared:** Allows users in the same PDB to use the model.
- **Version:** Displays the model version.
- **Namespace:** Displays the model namespace.
- **Owner:** The name of the user who deployed the model.
- **Deployed Date:** Displays the date of model deployment.

> **Note**
>
> You cannot re-deploy the same model. However, you can create a new version of the model and deploy it. You can then track the model based on the version.

- **URI:** Displays the URI name. Click on the URI link to view the REST API URI of the model.

Figure 10-3 REST API Specification of a Deployed Model



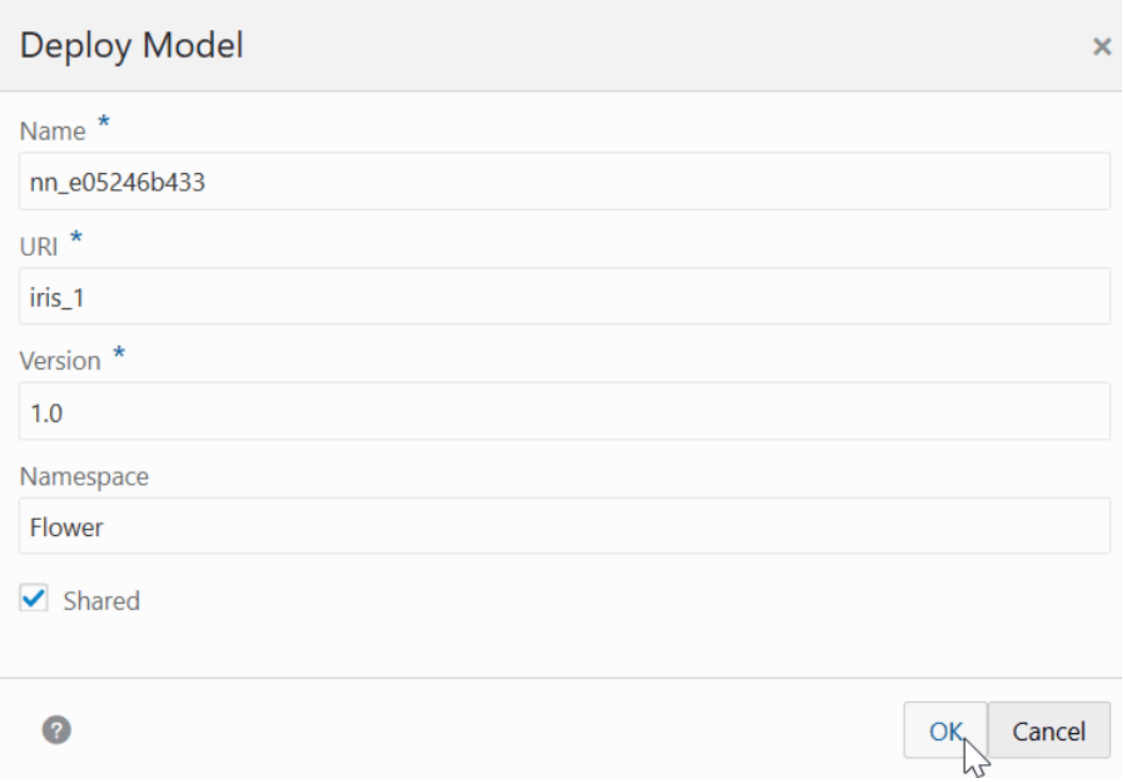
- Deploy Model
  When you deploy a model, you create an Oracle Machine Learning Services endpoint for scoring.

10.1 Deploy Model

When you deploy a model, you create an Oracle Machine Learning Services endpoint for scoring.

In the Deploy Model dialog, you can define the model deployment in the context of your AutoML UI experiment. To deploy a model, define the following:

Figure 10-4 Deploy Model dialog



1. In the **Name** field, the system generated model name is displayed here by default. You can edit this name. The model name must be a unique alphanumeric name with maximum 50 characters.
2. In the **URI** field, enter a name for the model URI. The URI must be alphanumeric, and the length must be max 200 characters.
3. In the **Version** field, enter a version of the model. The version must be in the format `x.x` where `x` is a number.
4. In the **Namespace** field, enter a name for the model namespace. This is an optional field.
5. Click **Shared** to allow users with access to the database schema to view and deploy the model.
6. Click **OK.** After a model is successfully deployed, it is listed in the Deployments page.
7. You can view the following details:
   - Model Metadata - Select a deployed model and click the model name to view model metadata such as the model name, mining function, algorithm, attributes and so on.
   - REST API - Select a deployed model and click the link under URI to view the REST API URI of the model.

11

Templates

By using the Oracle Machine Learning Notebooks templates, you can collaborate with other users by sharing your work, publishing your work as reports, and by creating notebooks from templates. You can store your notebooks as templates, share notebooks, and provide sample templates to other users.

> **Note**
>
> You can also collaborate with other Oracle Machine Learning Notebooks users by providing access to your workspace. The authenticated user can then access the projects in your workspace, and access your notebooks. The access level depends on the permission type granted - Manager, Developer, or Viewer. For more information about collaboration among users, see How to collaborate in Oracle Machine Learning Notebooks

- Use the Personal Templates
  Personal Templates lists the notebook templates that you have created.
- Use the Shared Templates
  In the Shared Templates, you can share notebook templates with all authenticated users the notebook templates you create from existing notebooks available in Templates.
- Use the Example Templates
  The Example Templates page lists the pre-populated Oracle Machine Learning notebook templates. You can view and use these templates to create your notebooks.

11.1 Use the Personal Templates

Personal Templates lists the notebook templates that you have created.

You can perform the following tasks:

- View selected templates in read-only mode.
- Create new notebooks from selected templates.
- Edit selected templates.
- Share selected notebook templates in Shared Templates.
- Delete selected notebook templates.
- Create Notebooks from Templates
  You can create new notebooks from an existing template, and store them in Personal Templates for later use.
- Share Notebook Templates
  You can share templates from Personal Templates. You can also share templates for editing.
- Edit Notebook Templates Settings
  You can modify the settings of an existing notebook template in Personal Templates.

11.1.1 Create Notebooks from Templates

You can create new notebooks from an existing template, and store them in Personal Templates for later use.

You must select a notebook template.

To create a new notebook from a template:

1. On the Personal Templates page, select the template based on which you want to create the notebook, and click **New Notebook.**

   The Create Notebook dialog box opens.
2. In the **Name** field, provide a name for the notebook.
3. In the **Comments** field, enter comments, if any.
4. In the **Project** field, select the project in which you want to save your notebook.
5. In the **Connection** field, the default connection is selected.
6. Click **OK.**

The notebook is created, and is available on the Notebooks page.

11.1.2 Share Notebook Templates

You can share templates from Personal Templates. You can also share templates for editing.

To share a template:

1. Select the notebook template in Personal Templates and click **Share.**

   The Save to Shared Templates dialog box opens.
2. In the **Name** field, enter a new name for the template.
3. In the **Comments** field, provide comments, if any.
4. In the **Tags** field, enter tags separated by commas. To enable easy searching, use descriptive tags.
5. Click **OK.**

Once the template is successfully created and shared, a message appears stating that the template is created in Shared.

11.1.3 Edit Notebook Templates Settings

You can modify the settings of an existing notebook template in Personal Templates.

To edit notebook template settings:

1. Select the notebook template in Personal Templates and click **Edit Settings.**

   The Edit Template dialog box opens.
2. In the **Name** field, edit the name, as applicable.
3. In the **Comments** field, edit the comments, if any.
4. In the **Tags** field, edit the tags, as applicable.
5. Click **OK.**

11.2 Use the Shared Templates

In the Shared Templates, you can share notebook templates with all authenticated users the notebook templates you create from existing notebooks available in Templates.

The Shared Templates page tracks notebook templates when you perform the following:

- Like templates
- Create notebooks from templates
- View templates

The Shared Templates page displays the following information about the templates:

- Template name
- Description
- Number of likes
- Number of creations
- Number of static views

You can perform the following tasks:

- Create templates by clicking **New Notebook**
- Edit template settings by clicking **Edit Settings**
- Delete any selected template by clicking **Delete**
- Search templates by Name, Tag, Author
- Sort templates by Name, Date, Author, Liked, Viewed, Used
- View templates by clicking **Show Liked Only** or **Show My Items Only**
- Create Notebooks from Templates
  You can create new notebooks from an existing template, and store them in Personal Templates for later use.
- Edit Notebook Templates Settings
  You can modify the settings of an existing notebook template in Personal Templates.

11.2.1 Create Notebooks from Templates

You can create new notebooks from an existing template, and store them in Personal Templates for later use.

You must select a notebook template.

To create a new notebook from a template:

1. On the Personal Templates page, select the template based on which you want to create the notebook, and click **New Notebook.**

   The Create Notebook dialog box opens.
2. In the **Name** field, provide a name for the notebook.
3. In the **Comments** field, enter comments, if any.
4. In the **Project** field, select the project in which you want to save your notebook.
5. In the **Connection** field, the default connection is selected.

6. Click **OK.**

The notebook is created, and is available on the Notebooks page.

11.2.2 Edit Notebook Templates Settings

You can modify the settings of an existing notebook template in Personal Templates.

To edit notebook template settings:

1. Select the notebook template in Personal Templates and click **Edit Settings.**

   The Edit Template dialog box opens.
2. In the **Name** field, edit the name, as applicable.
3. In the **Comments** field, edit the comments, if any.
4. In the **Tags** field, edit the tags, as applicable.
5. Click **OK.**

11.3 Use the Example Templates

The Example Templates page lists the pre-populated Oracle Machine Learning notebook templates. You can view and use these templates to create your notebooks.

The Example Templates page displays the following information about the templates:

- Template name
- Description
- Number of likes. Click **Likes** to mark it as liked.
- Number of static views
- Number of uses

You cannot alter any templates in the Example Templates page. The search options are:

- Search templates by Name, Tag, Author
- Sort templates by Name, Date, Author, Liked, Viewed, Used
- View templates that are liked by clicking **Show Liked only**
- Create a Notebook from the Example Templates
  Using the Oracle Machine Learning Example Templates, you can create a notebook from the available templates.
- Example Templates
  Oracle Machine Learning Notebooks provide you notebook example templates that are based on different machine learning algorithms and languages such as Python, R, and SQL. The example templates are processed in Oracle Autonomous AI Database.

11.3.1 Create a Notebook from the Example Templates

Using the Oracle Machine Learning Example Templates, you can create a notebook from the available templates.

To create a notebook:

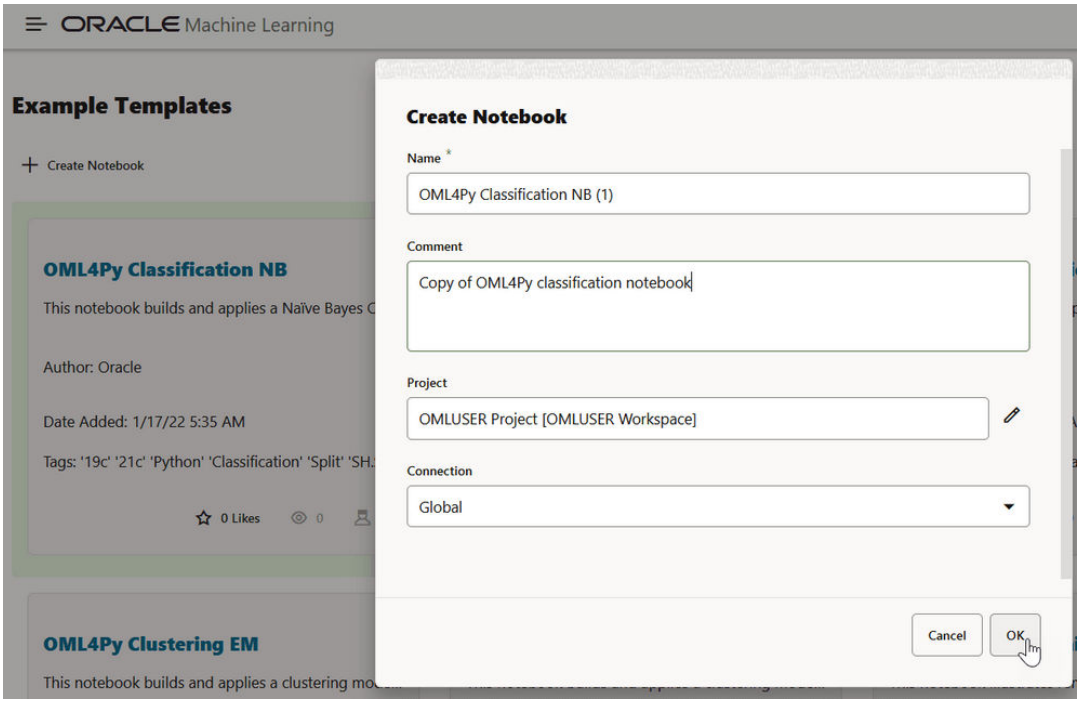
1. On the Example Templates page, select the template based on which you want to create a notebook.

2. Click **New Notebook.**

   The Create Notebook dialog box opens.

3. In the **Create Notebook** dialog, the name of the selected template appears. In the **Name** field, you can change the notebook name.

Figure 11-1 Create Notebook dialog



4. In the **Comment** field, if any comment is available for the template, then it is displayed. You can edit the comment.

5. In the **Project** field, click the edit icon .

6. Select the project in which you want to save the notebook.

7. In the **Connection** field, the default connection is selected.

8. Click **OK.**

The notebook is created and is available on the Notebooks page.

11.3.2 Example Templates

Oracle Machine Learning Notebooks provide you notebook example templates that are based on different machine learning algorithms and languages such as Python, R, and SQL. The example templates are processed in Oracle Autonomous AI Database.

You can create your own editable and runnable Oracle Machine Learning notebooks based on any of these Oracle Machine Learning for R example template notebooks:

- **OML Export and Import Serialized Models:** Use this notebook to export and import native serialized models using the `DBMS_DATA_MINING.EXPORT_SERMODEL` and `DBMS_DATA_MINING.IMPORT_SERMODEL` procedures. Oracle Machine Learning provides APIs to streamline the process of migrating models across databases and platforms.

- **OML Wiki ESA Model:** Use this notebook for text document categorization by calculating semantic relatedness (how similar in meaning two words or pieces of text are to each other) between the documents and a set of topics that are explicitly defined and described by humans. The Oracle Machine Learning for SQL function `ESA`, Oracle Machine Learning for Python function `oml.esa` and Oracle Machine Learning for R function `ore.odmESA` extract text-based features from a corpus of documents and performs document similarity comparisons. In this notebook, the wiki ESA model is imported to Autonomous AI Database for use with the following OML template notebook examples:
  - OML4SQL Feature Extraction ESA Wiki Model
  - OML4Py Feature Extraction ESA Wiki Model
  - OML4R Feature Extraction ESA Wiki Model
- **OML Services Batch Scoring:** Use this notebook to run batch scoring jobs through a REST interface via OML Services. OML Services supports batch scoring for regression, classification, clustering, and feature extraction.
  - Authenticate the database user and obtain a token
  - Create a batch scoring job
  - View the details and output of the batch scoring job
  - Update, disable, and delete a batch scoring job
- **OML Services Data Monitoring:** Use this notebook to perform data monitoring. This notebook runs provides you with the data monitoring workflow steps through the REST interface that includes:
  - Authenticate the database user and obtain a token
  - Create a data monitoring job
  - View the details and output of the data monitoring job
  - Update, disable, and delete a data monitoring job
- **OML Services Model Monitoring:** Use this notebook to understand and perform model monitoring. This notebook runs provides you with the model monitoring workflow steps through the REST interface that includes:
  - Authenticate the database user and obtain a token
  - Create a model monitoring job
  - View the details and output of the model monitoring job
  - Update, disable, and delete a model monitoring job
- **OML Third-Party Packages - Environment Creation:** Use this notebook to download and activate the Conda environment, and to use the libraries in your notebook sessions. Oracle Machine Learning Notebooks provide a Conda interpreter to install third-party Python and R libraries in a Conda environment for use within Oracle Machine Learning Notebooks sessions, as well as within Oracle Machine Learning for Python and Oracle Machine Learning for R embedded execution calls.
  The third-party libraries installed in Oracle Machine Learning Notebooks can be used in:
  - Standard Python
  - Standard R
  - Oracle Machine Learning for Python embedded Python execution from the Python, SQL and REST APIs

– Oracle Machine Learning for R embedded R execution from the R, SQL, and REST APIs

> **Note**
>
> The Conda environment is installed and managed by the ADMIN user with `OML_SYS_ADMIN` role. The administrator can create a shared environment, and add or delete packages from it. The Conda environments are stored in the Object Storage bucket associated with the Autonomous AI Database.

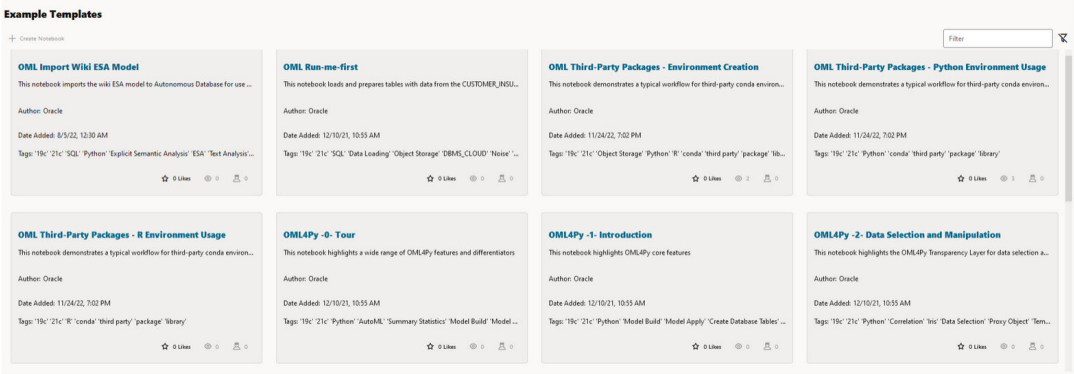
Conda is an open source package and environment management system that enables the use of virtual environments containing third-party R and Python packages. With conda environments, you can install and update packages and their dependencies, and switch between environments to use project-specific packages.

This template notebook *OML Third-Party Packages - Environment Creation* contains a typical workflow for third-party environment creation and package installation in Oracle Machine Learning Notebooks.

– Section 1 contains commands to create and test Conda environments.

– Section 2 contains commands to create a Conda environment, install packages, and commands to upload the conda environment to an Object Storage bucket associated with the Oracle Autonomous AI Database.

Figure 11-2 Conda Example Templates



- **OML Third-Party Packages - Python Environment Usage:** Use this template notebook to understand the typical workflow for third-party environment usage in Oracle Machine Learning Notebooks using Python and Oracle Machine Learning for Python. You download and use the libraries in Conda environments that were previously created and saved to an Object Storage bucket folder associated with the Autonomous AI Database.
  This notebook contains commands to:
  – List all environments stored in Object Storage
  – List a named environment stored in Object Storage
  – Download and activate the `mypyenv` environment
  – List the packages available in the Conda environment
  – Import Python libraries
  – Load datasets

  - Build models
  - Score models
  - Create Python user defined functions (UDFs)
  - Run the user defined functions (UDFs) in Python
  - Create and run user defined functions in the SQL and REST APIs for embedded Python execution
  - Create and run Python user defined functions using the SQL API for embedded Python execution - asynchronous mode
- **OML Third-Party Packages - R Environment Usage:** Use this template notebook to understand the typical workflow for third-party environment usage in Oracle Machine Learning for R.
  This notebook contains commands to:
  - List all environments stored in Object Storage
  - List a named environment stored in Object Storage
  - Download and activate the `myrenv` environment
  - Show the list of available OML4R Conda environments
  - Import R libraries
  - Load and prepare data
  - Build models
  - Score models
  - Create R user defined functions (UDFs)
  - Run user defined functions (UDFs) in R
  - Save the user defined functions (UDFs) to the script repository
  - Run the R user defined function in SQL and REST APIs for embedded R execution
  - Add the OML user to the Access Control List
  - Run the R user defined functions using the REST API for embedded R execution in synchronous mode
- **OML4R-1: Introduction:** Use this notebook to understand how to:
  - Load the ORE Library
  - Create database tables
  - Use the transparency layer
  - Rank attributes for predictive value using the in-database attribute importance algorithm
  - Build predictive models, and
  - Score data using these models
- **OML4R-2: Data Selection and Manipulation:** Use this notebook to understand the features of the transparency layer involving data selection and manipulation.
- **OML4R-3: Datastore and Script Repository:** Use this notebook to understand the datastore and script repository features of OML4R.
- **OML4R-4: Embedded R Execution:** Use this notebook to understand OML4R embedded R execution. First, a linear model is built in R directly, then a user-define R function is

created to build the linear model, the function is then saved to the script repository, and the data is scored in parallel using R engines spawned by the Oracle Autonomous AI Database environment. The notebook also demonstrates how to call these scripts using the SQL interface and REST API for R with embedded R execution.

> **Note**
>
> To use the SQL API for embedded R execution, a user-defined R function must reside in the OML4R script repository, and an Oracle Machine Learning (OML) cloud account USERNAME, PASSWORD, and URL must be provided to obtain an authentication token.

- **OML4R Anomaly Detection Support Vector Machine (SVM):** Use this notebook to build a One-Class SVM model and then use it to flag unusual or suspicious records.
- **OML4R Association Rules Apriori:** Use this notebook to build association rules models using the A Priori algorithm with data from the SH schema (SH.SALES). All computation occurs inside Oracle Autonomous AI Database.
- **OML4R Attribute Importance Minimum Description Length (MDL):** Use this notebook to compute Attribute Importance, which uses the Minimum Description Length algorithm, on the SH schema data. All functionality runs inside Oracle Autonomous AI Database. Oracle Machine Learning supports Attribute Importance to identify key factors such as attributes, predictors, variables that have the most influence on a target attribute.
- **OML4R Classification Generalized Linear Model (GLM):** Use this notebook to predict customers most likely to be positive responders to an Affinity Card loyalty program. This notebook builds and applies a classification generalized linear model using the Sales History (SH) schema data. All processing occurs inside Oracle Autonomous AI Database.
- **OML4R Classification Naive Bayes (NB):** Use this notebook to predict customers most likely to be positive responders to an Affinity Card loyalty program. This notebook builds and applies a classification decision tree model using the Sales History (SH) schema data. All processing occurs inside Oracle Autonomous AI Database.
- **OML4R Classification Random Forest (RF):** Use this notebook to use the Random Forest algorithm for classification in OML4R, and predict customers most likely to be positive responders to an Affinity Card loyalty program.
- **OML4R Classification Modeling to Predict Target Customers using Support Vector Machine (SVM):** Use this notebook to use the Classification modeling to predict Target Customers using Support Vector Machine Model.
- **OML4R Clustering - Identifying Customer Segments using Expectation Maximization Clustering:** Use this notebook to understand how to identify natural clusters of customers using the CUSTOMERS data set from the SH schema using the unsupervised learning Expectation Maximization (EM) algorithm. The data exploration, preparation, and machine learning run inside Oracle Autonomous AI Database.
- **OML4R Clustering - Identifying Customer Segments using K-Means Clustering:** Use this notebook to understand how to identify natural clusters of customers using the `CUSTOMERS` data set from the SH schema using the unsupervised learning K-Means (KM) algorithm. The data exploration, preparation, and machine learning run inside Oracle Autonomous AI Database.
- **OML4R Clustering - Orthogonal Partitioning Clustering (OC):** Use this notebook to understand how to identify natural clusters of customers using the CUSTOMERS data set from the SH schema using the unsupervised learning k-Means algorithm. The data exploration, preparation, and machine learning runs inside Oracle Autonomous AI Database.

- **OML4R Data Cleaning Outlier:** Use this notebook understand and to exclude records with outliers using OML4R.
- **OML4R Data Cleaning - Recode Synonymous Values:** Use this notebook to recode synonymous value using OML4R.
- **OML4R Data set Creation:** Use this notebook to load the sample data sets `MTCARS` and `IRIS` and to import them into your Oracle Autonomous AI Database instance using the `ore.create()` function.

> **Note**
>
> The following example template notebooks prefixed with an asterisk (*), use the `CUSTOMER_INSURANCE_LTV` data set. This data set is generated by the `OML Run-me-first` notebook. Hence, you must run the `OML Run-me-first` notebook that is available under the Examples Templates.

- ***** **OML4R Data Cleaning Missing Data:** Use this notebook to perform missing value replacement using OML4R.
- ***** **OML4R Data Cleaning Duplicate Removal:** Use this notebook to remove duplicate records using OML4R.
- ***** **OML4R Data Transformation Binning:** Use this notebook to bin numerical columns using OML4R.
- ***** **OML4R Data Transformation Categorical Record:** Use this notebook to recode a categorical string variable to a numeric variable and string-to-string recoding using OML4R.
- ***** **OML4R Data Transformation: Normalization and Scaling:** Use this notebook to normalize and scale data using OML4R.
- ***** **OML4R Data Transformation: One-Hot Encoding:** Use this notebook to perform one-hot encoding using OML4R.
- ***** **OML4R Feature Selection - Supervised Algorithm:** Use this notebook to perform feature selection using in-database supervised algorithms using OML4R. This notebook demonstrates how to build a Random Forest model to predict whether the customer would buy insurance or not, and then use Feature Importance to perform feature selection.
- ***** **OML4R Feature Selection Using Summary Statistics:** Use this notebook to perform feature selection using summary statistics using OML4R. This notebook demonstrates how to use OML4R to select features based on number of distinct values, null values, proportion of constant values.
- **OML4R Feature Engineering Aggregation:** Use this notebook to perform aggregation for min, max, mean, and count using OML4R. This template uses the `SALES` table present in the `SH` schema, and shows how to create features by aggregating the amount sold for each customer and product pair.
- **OML4R Feature Extraction Explicit Semantic Analysis (ESA) Wiki Model:** This notebook uses the use the Wikipedia Model as an example. Use this notebook to use the Oracle Machine Learning for R function `ore.odmESA` to extract text-based features from a corpus of documents and performs document similarity comparisons. All processing occurs inside Oracle Autonomous AI Database.

> **Note**
>
> The pre-built Wikipedia model must be installed in your Autonomous AI Database instance to run this notebook.

- **OML4R Data Transformation Date Data types:** Use this notebook to perform various operations on date and time data using database table proxy objects using Oracle Machine Learning for R.
- **OML4R Feature Extraction Singular Value Decomposition (SVD):** Use this notebook to use the in-database SVD for feature extraction. This notebook uses the Oracle Machine Learning for R function `ore.odmSVD` to create a model that uses the Singular Value Decomposition (SVD) algorithm for feature extraction.
- **OML4R Partitioned Model Support Vector Machine (SVM):** Use this notebook to build an SVM model to predict the number of years a customer resides at their residence but partitioned on customer gender. The model is then used to predict the target, then predict the target with prediction details.
- **OML4R Regression Generalized Linear Model (GLM):** Use this notebook to understand how to predict numerical values using multiple regression. This notebook uses the Generalized Linear Model algorithm.
- **OML4R Regression Neural Network (NN):** Use this notebook to understand how to predict numerical values using multiple regression. This notebook uses the Neural Network algorithm.
- **OML4R Regression Support Vector Machine (SVM):** Use this notebook to understand how to predict numerical values using multiple regression. This notebook uses the Support Vector Machine algorithm.
- **OML4R REST API:** Use this notebook to understand how to use the OML4R REST API to call user-defined R functions, and to list those available in the R script repository.

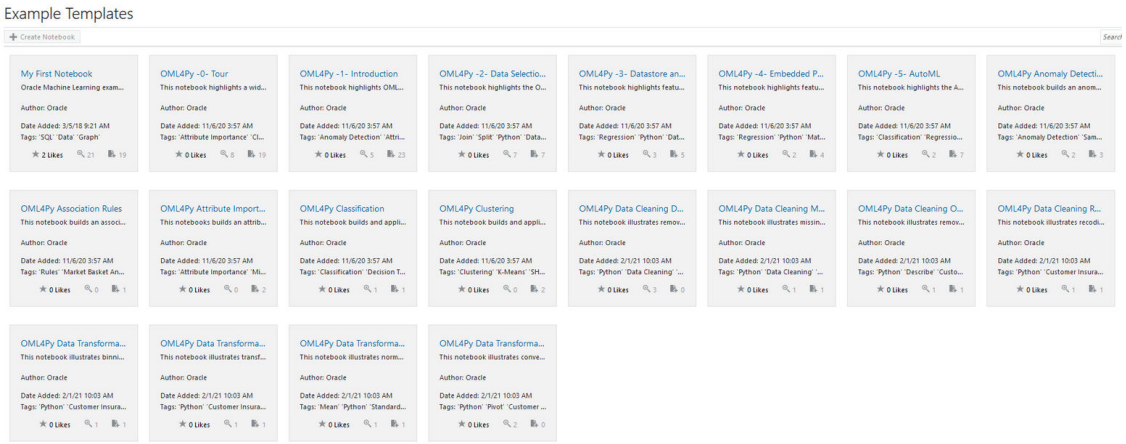
> **Note**
>
> To run a script, it must reside in the R script repository. An Oracle Machine Learning cloud service account user name and password must be provided for authentication.

- **OML4R Statistical Function:** Use this notebook to understand and use various statistical functions. The notebook uses data from the SH schema through the OML4R transparency layer.
- **OML4R Text Mining Support Vector Machine (SVM):** Use this notebook to understand how to use unstructured text data to build machine learning models, leverage Oracle Text, use Oracle Machine Learning in-database algorithms predictive features, and extract features from text columns.
  This notebook builds a Support Vector Machine (SVM) model to predict customers that are most likely to be positive responders to an Affinity Card loyalty program. The data comes from a text column that contains user generated comments.

Oracle Machine Learning for Python Example Templates

You can create your Oracle Machine Learning notebooks based on any of these Oracle Machine Learning for Python example templates:

Figure 11-3 Oracle Machine Learning for Python Example Templates



- **My First Notebook:** Use the My First Notebook notebook for basic machine learning functions, data selection and data viewing. This template uses the `SH` schema data.
- **OML4Py -0- Tour:** This notebook is the first of a series 0 through 5 that is intended to introduce you to the range of OML4Py functionality through short examples.
- **OML4Py -1- Introduction:** This notebook provides an overview on how to load OML library, create database tables, use the transparency layer, rank attributes for predictive value using the in-database attribute importance algorithm, build predictive models, and score data using these models.
- **OML4Py -2- Data Selection and Manipulation:** Use this notebook to learn how to work with the transparency layer which involves data selection and manipulation.
- **OML4Py -3- Datastores:** Use this notebook to learn how to work with datastores, move objects between datastore and a Python sessions, manage datastore privileges, save model objects and Python objects in a datastore, delete datastores and so on.
- **OML4Py -4- Embedded Python Execution:** Use this notebook to understand embedded Python Execution. In this notebook, a linear model is build in Python directly, and then a function is created that uses Python engines spawned by the Autonomous AI Database environment.
- **OML4Py -5- AutoML:** Use this notebook to understand the AutoML workflow in OML4Py. In this notebook, the `WINE` data set from scikit-learn is used. Here, AutoML is used for classification on the `target` column, and for regression on the `alcohol` column.

> **Note**
>
> The following example template notebooks prefixed with an asterisk (*), use the `CUSTOMER_INSURANCE_LTV_PY` data set. This data set has duplicated values artificially generated by **OML4SQL Noise** notebook. Hence, you must run the **OML4SQL Noise** first before running the notebook.

- *** OML4Py Data Cleaning Duplicates Removal:** Use this notebook to understand how to remove duplicate records using OML4Py. This notebook uses the customer insurance lifetime value data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance.

- *** OML4Py Data Cleaning Missing Data:** Use this notebook to understand how to fill in missing values using OML4Py. This notebook uses the customer insurance lifetime value data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance.
- *** OML4Py Data Cleaning Recode Synonymous Values:** Use this notebook to understand how to recode synonymous value using OML4Py. This notebook uses the customer insurance lifetime value data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance.
- **OML4Py Data Cleaning Outlier Removal:** Use this notebook to understand how to clean data to remove outliers. This notebook uses the `CUSTOMER_INSURANCE_LTV` data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance. In the data set `CUSTOMER_INSURANCE_LTV`, the focus is on numerical values and removal of records with values in the top and bottom 5%.
- **OML4Py Data Transformation Binning:** Use this notebook to understand how to bin a numerical column and visualize the distribution.
- **OML4Py Data Transformation Categorical - Convert Categorical Variables to Numeric Variables:** Use this notebook to understand how to convert categorical variables to numeric variables using OML4Py. The notebooks demonstrates how to convert a categorical variable with each distinct level/value coded to an integer data type.
- **OML4Py Data Transformation Normalization and Scaling:** Use this notebook to understand how to normalize and scale data using z-score (mean and standard deviation), min max scaling, and log scaling.

  > **Note**
  >
  > When building or applying a model using in-database Oracle Machine Learning algorithms, automatic data preparation will normalize data automatically as needed, by specific algorithms.

- **OML4Py Data Transformation One Hot Encoding:** Use this notebook to understand how to perform one hot encoding using OML4Py. Machine learning algorithms cannot work with categorical data directly. Categorical data must be converted to numbers. This notebook uses the customer insurance lifetime value data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance.

  > **Note**
  >
  > If you plan to use the in-database algorithms, one hot encoding is automatically applied for those algorithms requiring it. The in-databae algorithms automatically explode the categorical columns and fit the model on the prepared data internally.

- **OML4Py Anomaly Detection:** Use this notebook to detect anomalous records, customers or transactions in your data. This template uses the unsupervised learning algorithm 1-Class Support Vector Machine. The notebook template builds a 1-Class Support Vector Machine (SVM) model.
- **OML4Py Association Rules:** Use this notebook for market basket analysis of your data, or to detect co-occurring items, failures or events in your data. This template uses the apriori Association Rules model using the `SH` schema data (`SH.SALES`).
- **OML4Py Attribute Importance:** Use this notebook to identify key attributes that have maximum influence over the target attribute. The target attribute in the build data of a

supervised model is the attribute that you want to predict. The template builds an Attribute Importance model using the SH schema data.

- **OML4Py Classification:** Use this notebook for predicting customer behavior and similar predictions. The template builds and applies the classification algorithm Decision Tree to build a Classification model based on the relationships between the predictor values and the target values. The template uses the SH schema data.
- **OML4Py Clustering:** Use this notebook to identify natural clusters in your data. The notebook template uses the unsupervised learning *k*-Means algorithm on the SH schema data.
- **OML4Py Data Transformation :** Use this notebook to convert categorical variables to numeric variables using OML4Py. This template shows how to convert a categorical variable with each distinct level/value coded to an integer data type.
- **OML4Py Data Set Creation:** Use this notebook to create data set from sklearn package to OML data frame using OML4Py.
- **OML4Py Feature Engineering Aggregation:** Use this notebook template to fill in missing values using OML4Py. This notebook uses the SH schema SALES table, which contains transaction records for each customer and products purchased. Features are created by aggregating the amount sold for each customer and product pair.
- **OML4Py Feature Selection Supervised Algorithm Based:** Use this notebook to perform feature selection using in-database supervised algorithms using OML4Py.
- **OML4Py Feature Selection Summary Statistics:** Use this notebook template to perform feature selection using summary statistics using OML4Py. The notebook shows how to use OML4Py to select features based on number of distinct values, null values, proportion of constant values. The data set used here CUSTOMER_INSURANCE_LTV_PY has null values generated by the OML4SQL Noise notebook artificially. You must first run the OML4SQL Noise notebook before running the OML4Py Feature Selection Summary Statistics notebook.
- **OML4Py Partitioned Model:** Use this notebook to build partitioned models. This notebook builds an SVM model to predict the number of years a customer resides at their residence but partitioned on customer gender. It uses the model to predict the target, and then predict the target with prediction details.
  Oracle Machine Learning enables automatically building of an ensemble model comprised of multiple sub-models, one for each data partition. Sub-models exist and are used as one model, which results in simplified scoring using the top-level model only. The proper sub-model is chosen by the system based on partition values in the row of data to be scored. Partitioned models achieve potentially better accuracy through multiple targeted models.
- **OMP4Py REST API:** Use this notebook to call embedded Python execution. OML4Py contains a REST API to run user-defined Python functions saved in the script repository. The REST API is used when separation between the client and the Database server is beneficial. Use the OML4Py REST API to build, train, deploy, and manage scripts.

  > **Note**
  >
  > To run a script, it must reside in the OML4Py script repository. An Oracle Machine Learning cloud service account user name and password must be provided for authentication.

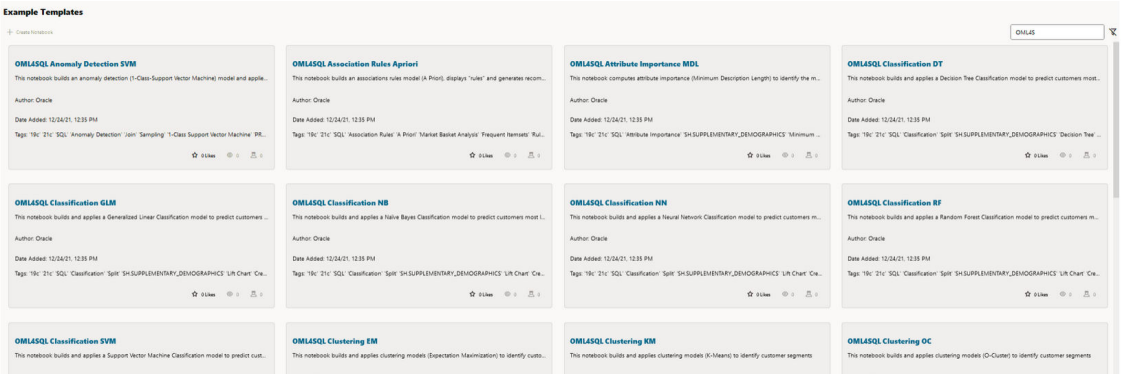
- **OML4Py Regression Modeling to Predict Numerical Values:** Use this notebook to predict numerical values using multiple regression.

- **OML4Py Statistical Functions:** Use this notebook to use various statistical functions. The statistical functions use data from the SH schema through the OML4Py transparency layer.
- **OML4Py Text Mining:** Use this notebook to build models using Text Mining capability in Oracle Machine Learning.
  In this notebook, an SVM model is built to predict customers most likely to be positive responders to an Affinity Card loyalty program. The data comes with a text column that contains user generated comments. With a few additional specifications, the algorithm automatically uses the text column and builds the model on both the structured data and unstructured text.

Oracle Machine Learning for SQL Example Templates

You can create your Oracle Machine Learning notebooks based on any of these Oracle Machine Learning for SQL example templates:

Figure 11-4 Oracle Machine Learning for SQL Example Templates



- **OML4SQL Anomaly Detection:** Use this notebook to detect unusual or rare occurrences. Oracle Machine Learning supports anomaly detection to identify rare or unusual records (customers, transactions, etc.) in the data using the semi-supervised learning algorithm One-Class Support Vector Machine. This notebook builds a 1Class-SVM model and then uses it to flag unusual or suspicious records. The entire machine learning methodology runs inside Oracle Autonomous AI Database.
- **OML4SQL Association Rules:** Use this notebook to apply Association Rules machine learning technique, also known as Market Basket Analysis to discover co-occurring items, states that lead to failures, or non-obvious events. This notebook builds associations rules models using the A Priori algorithm with the SH.SALES data from the SH schema. All computation occurs inside Oracle Autonomous AI Database.
- **OML4SQL Attribute Importance - Identify Key Factors:** Use this notebook to identify key factors, also known as attributes, predictors, variables that have the most influence on a target attribute. This notebook builds an Attribute Importance model, which uses the Minimum Description Length algorithm, using the SH schema data. All functionality runs inside Oracle Autonomous AI Database.
- **OML4SQL Classification - Predicting Target Customers:** Use this notebook to predict customers that are most likely to be positive responders to an Affinity Card loyalty program. This notebook builds and applies classification decision tree models using the SH schema data. All processing occurs inside Oracle Autonomous AI Database.
- **OML4SQL Clustering - Identifying Customer Segments:** Use this notebook to identify natural clusters of customers. Oracle Machine Learning supports clustering using several algorithms, including k-Means, O-Cluster, and Expectation Maximization. This notebook

uses the CUSTOMERS data set from the SH schema using the unsupervised learning k-Means algorithm. The data exploration, preparation, and machine learning runs inside Oracle Autonomous AI Database.

- **OML4SQL Data Cleaning - Removing Duplicates:** Use this notebook to remove duplicate records using Oracle SQL. The notebook uses the customer insurance lifetime value data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance. The data set CUSTOMER_INSURANCE_LTV_SQL has duplicate values generated by the OML4SQL Noise notebook.

  > **Note**
  >
  > You must first run the OML4SQL Noise notebook before running the OML4SQL Data Cleaning notebook.

- **OML4SQL Data Cleaning - Missing Data:** Use this template to replace missing values using Oracle SQL and the DBMS_DATA_MINING_TRANSFORM package. The data set CUSTOMER_INSURANCE_LTV_SQL has missing values artificially generated by the OML4SQL Noise notebook. You must first run the OML4SQL Noise notebook before running the OML4SQL Data Cleaning notebook.

  > **Note**
  >
  > When building or applying a model using in-database Oracle Machine Learning algorithms, this operation may not be needed separately if automatic data preparation is enabled. Automatic data preparation automatically replaces missing values of numerical attributes with the mean and missing values of categorical attributes with the mode.

- **OML4SQL Data Cleaning Outlier Removal:** Use this notebook to remove outliers using Oracle SQL and the DBMS_DATA_MINING_TRANSFORM package. The notebook uses the customer insurance lifetime value data set, which contains customer financial information, lifetime value, and whether or not the customer bought insurance. In the data set CUSTOMER_INSURANCE_LTV, it focuses on numeric values and removes records with values in the top and bottom 5%.
- **OML4SQL Data Cleaning Recode Synonymous Values:** Use this notebook to recode synonymous value of a column using Oracle SQL. The notebook uses the customer insurance lifetime value data set, which contains customer financial information, lifetime value, and whether or not the customer bought insurance. The data set CUSTOMER_INSURANCE_LTV_SQL has recoded values generated by the OML4SQL Noise notebook. You must first run the OML4SQL Noise notebook before running the OML4SQL Data Cleaning - Recode Synonymous Values notebook.
- **OML4SQL Data Transformation Binning:** Use this notebook to bin numeric columns using Oracle SQL and the DBMS_DATA_MINING_TRANSFORM package. This notebook shows how to bin a numerical column and visualize the distribution.
- **OML4SQL Data Transformation Categorical:** Use this notebook to convert a categorical variable to a numeric variable using Oracle SQL. The notebook shows how to convert a categorical variable with each distinct level/value coded to an integer, and how to create an indicator variable based on a simple predicate.
- **OML4SQL Data Transformation Normalization and Scale:** Use this notebook to normalize and scale data using Oracle SQL and the DBMS_DATA_MINING_TRANSFORM package. The notebook shows how to normalize data using using z-score (mean and

standard deviation), min max scaling, and log scaling. When building or applying a model using in-database Oracle Machine Learning algorithms, automatic data preparation normalizes data automatically, as needed, by specific algorithms.

- **OML4SQL Dimensionality Reduction - Non-negative Matrix Factorization:** Use this notebook to perform dimensionality reduction using the in-database non-negative matrix factorization algorithm. This notebook shows how to convert a table with many columns to a reduced feature set. Non-negative Matrix Factorization produces non-negative coefficients.
- **OML4SQL Dimensionality Reduction - Singular Value Decomposition:** Use this notebook to perform dimensionality reduction using the in-database singular value decomposition (SVD) algorithm.
- **OML4SQL: Exporting Serialized Models:** Use this notebook to export serialized models to Oracle Cloud Object Storage. This notebook creates Oracle Machine Learning regression and classification models and exports the models in a serialized format so that they can be scored using the Oracle Machine Learning (OML) Services REST API. OML Services provides REST API endpoints hosted on Oracle Autonomous AI Database. These endpoints enable the storing of Oracle Machine Learning models along with its metadata and create scoring endpoints for the model. The REST API for OML Services supports both Oracle Machine Learning models and ONNX format models, and enables cognitive text functionality.
- **OML4SQL Feature Engineering Aggregation and Time:** Use this notebook to generate aggregated features and also extract date and time features using Oracle SQL. The notebook also shows how to extract date and time features from the field `TIME_ID`.
- **OML4SQL Feature Selection Algorithm Based:** Use this notebook to perform feature selection using in-database supervised algorithms. The notebook first builds a random forest model to predict if the customer will buy insurance, then it uses Feature Importance values for feature selection. It then build a decision tree model for the same classification task, and obtains split nodes. For the top splitting nodes with highest support, features associated with those nodes are selected.
- **OML4SQL Feature Selection Unsupervised Attribute Importance:** Use this notebook to perform feature selection using the in-database unsupervised algorithm Expectation Maximization (EM). This notebook illustrates using the `CREATE_MODEL` function, which leverages the settings table in contrast to the `CREATE_MODEL2` function used in other notebooks.
- **OML4SQL Feature Selection Using Summary Statistics:** Use this notebook to perform feature selection using summary statistics using Oracle SQL. The data set `CUSTOMER_INSURANCE_LTV_SQL` has null values generated by OML4SQL Noise notebook artificially. You must first run the OML4SQL Noise notebook before running the OML4SQL Feature Selection Using Summary Statistics.
- **OML4SQL Noise:** Use this notebook to replace normal values by null values, and to add duplicated rows. In this notebook, the data set used by the Data Preparation notebooks is prepared, in particular those for data cleaning and feature selection. It uses the customer insurance lifetime value data set which contains customer financial information, lifetime value, and whether or not the customer bought insurance.

  > **Note**
  >
  > Run the OML4SQL Noise notebook before the Data Preparation notebooks.

- **OML4SQL Partitioned Model:** Use this notebook to build partitioned models. Partitioned models achieve potentially better accuracy through multiple targeted models. The

notebook builds an SVM model to predict the number of years a customer resides at their residence but partitioned on customer gender. The model is then used to predict the target first, and then to predict the target with prediction details.

- **OML4SQL Text Mining:** Use this notebook to build models using text mining capability. Oracle Machine Learning handles both structured data and unstructured text data. By leveraging Oracle Text, Oracle Machine Learning in-database algorithms automatically extracts predictive features from the text column.
  This notebook builds an SVM model to predict customers most likely to be positive responders to an Affinity Card loyalty program. The data comes with a text column that contains user generated comments. With a few additional specifications, the algorithm automatically uses the text column and builds the model on both the structured data and unstructured text.
- **OML4SQL Regression:** Use this notebook to predict numerical values. This template uses multiple regression algorithms such as Generalized Linear Models (GLM).
- **OML4SQL Statistical Function:** Use this notebook for descriptive and comparative statistical functions. The notebook template uses `SH` schema data.
- **OML4SQL Time Series:** Use this notebook to build time series models on your time series data for forecasting. This notebook is based on the Exponential Smoothing Algorithm. The sales forecasting example in this notebook is based on the `SH.SALES` data. All computations are done inside Oracle Autonomous AI Database.

12 Jobs

Jobs allow you to schedule the running of notebooks. On the Jobs page, you can create jobs, duplicate jobs, start and stop jobs, delete jobs, and monitor job status by viewing job logs, which are read-only notebooks.

- About Jobs
  The Jobs page lists all the jobs created, along with the job name, notebook, owner of the job, last start date, next run date, status, and schedule.
- Create Jobs to Schedule Notebooks
  You can create jobs to schedule your notebook with the notebook scheduling settings.
- View Job Logs
  You can view the historical logs of any particular job in the Job Log interface.

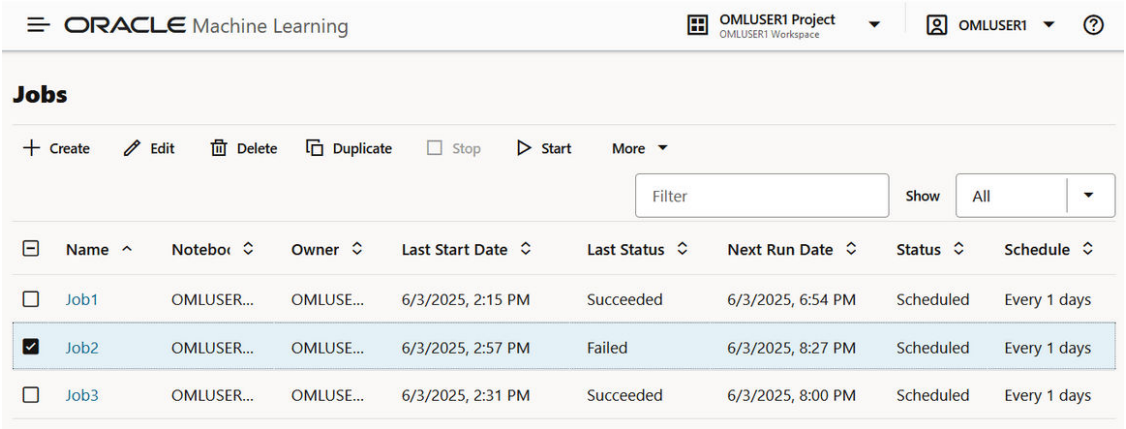
12.1 About Jobs

The Jobs page lists all the jobs created, along with the job name, notebook, owner of the job, last start date, next run date, status, and schedule.

You can create multiple jobs for a notebook. You can run a job only when the notebook associated with it is not running.

> **Note**
>
> You can run multiple jobs at the same time, but you must start them individually, and not in bulk.

Figure 12-1 Jobs page



This is the Jobs page. You can perform the following tasks:

> **Note**
>
> You can perform these actions only on jobs created by you. For jobs created by another user, these options are disabled. You must have the `OML_SYS_ADMIN` role to perform these actions—`Delete, Enable, Disable` on jobs created by another user.

- **Create:** Click **Create** to create a new job to schedule your notebook.
- **Edit:** You can edit the metadata of any job listed on the Jobs page. Select the job you want to edit and then click **Edit**.
- **Delete:** Select the job you want to delete and click **Delete**. You must have the `OML_SYS_ADMIN` to delete a job created by another user.
- **Duplicate:** You can create a copy of an existing job listed on the Jobs page. Click **Duplicate** to make a copy of the selected job.
- **Stop:** Click **Stop** to end a job that is currently running.
- **Start:** The **Start** button is enabled only for jobs that are in *Scheduled* status. Click **Start** to start a scheduled job. You can start multiple jobs at the same time, but you must start them individually, and not in bulk. The **Start** option is not applicable for the following conditions:
  - Jobs that have already completed its scheduled run cannot be re-started.
  - Jobs that have failed more than the allowed number of times, and are currently in *Broken* status, cannot be re-started.
- **More:** Under **More**, you have the options to enable or disable a job.
  - **Enable:** Select a job that you want to enable, click **More** and then click **Enable**. You must have the `OML_SYS_ADMIN` to enable a job created by another user.
  - **Disable:** Select a job that you want to disable, click **More** and then click **Disable**. You must have the `OML_SYS_ADMIN` to disable a job created by another user.

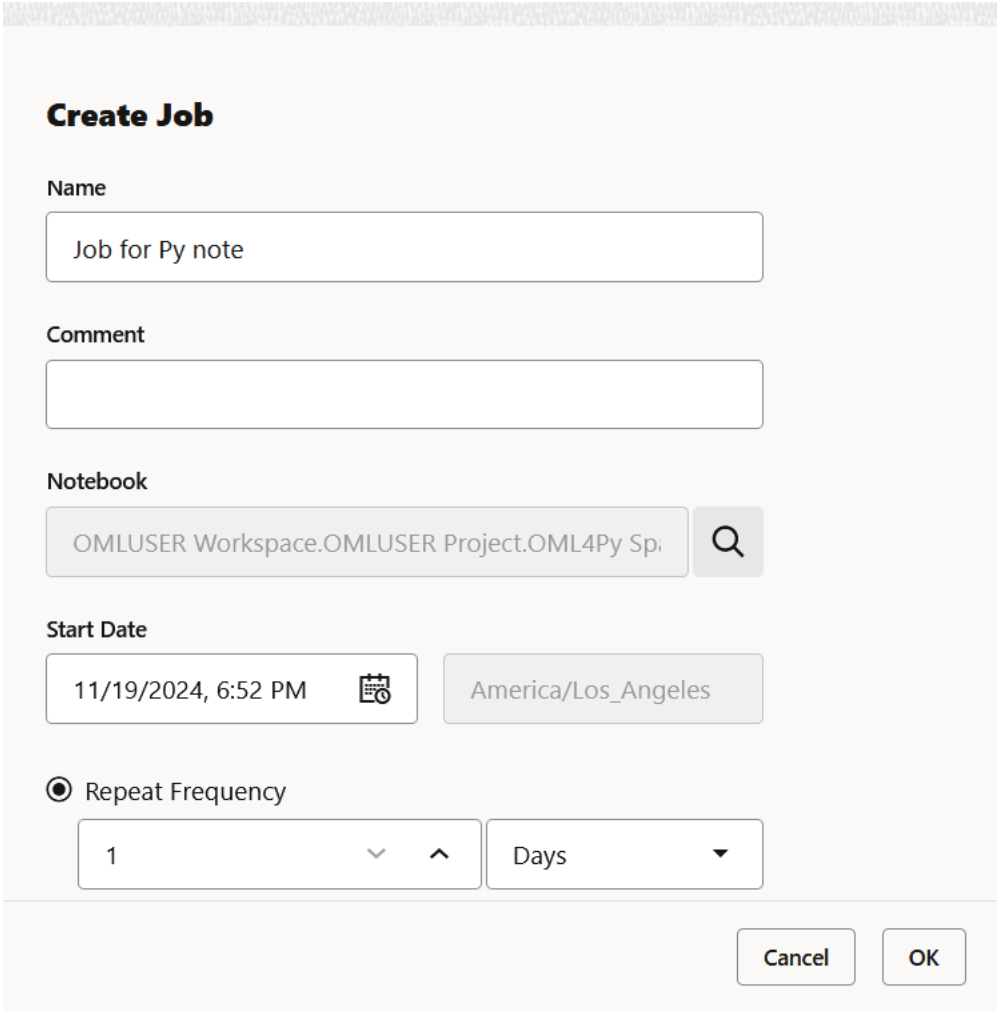
12.2 Create Jobs to Schedule Notebooks

You can create jobs to schedule your notebook with the notebook scheduling settings.

To create jobs, enter the following details in the Create Jobs dialog box:

1. On the Jobs page, click **Create.** The Create Jobs dialog opens.

Figure 12-2 Create Jobs dialog

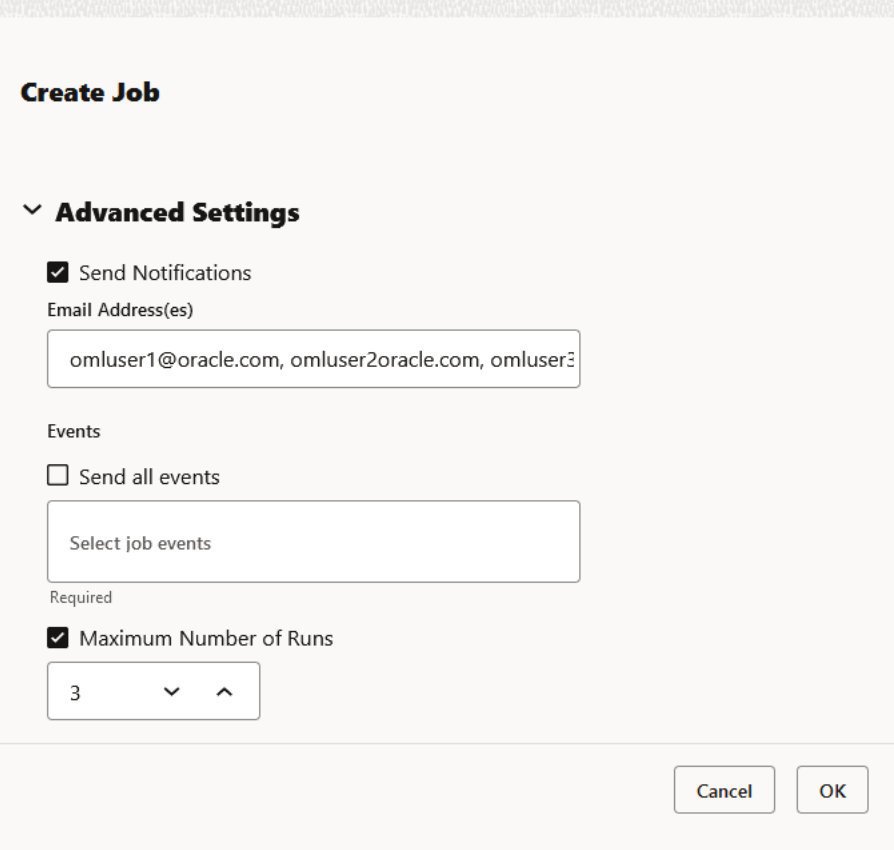


2. In the **Name** field, enter a name for the job. The number of characters in the job name must not exceed 128 bytes.
3. In the **Notebook** field, click the search icon to select a notebook to create a job.

> **Note**
>
> Only notebooks that are owned by the user or shared are available for selection.

4. In the **Start Date** field, click the date-time editor to set the date and time for your job to commence. Based on the selected date and time, the next run date is computed.
5. Optionally, in the **Repeat** section, select:
   - **Frequency:** To set the repeat settings and frequency. You can set the frequency in minutes, hours, days, week, and month.
   - **Custom:** To customize the job settings.
6. Expand the **Advanced Settings** section, select one or more of the following options:

Figure 12-3 Advanced Settings in Create Jobs dialog



- **Send Notifications:** Select this option to send email notifications to the specified users' email addresses about the selected events for the job.

  > **Note**
  >
  > Email server must be set up for notifications to work.

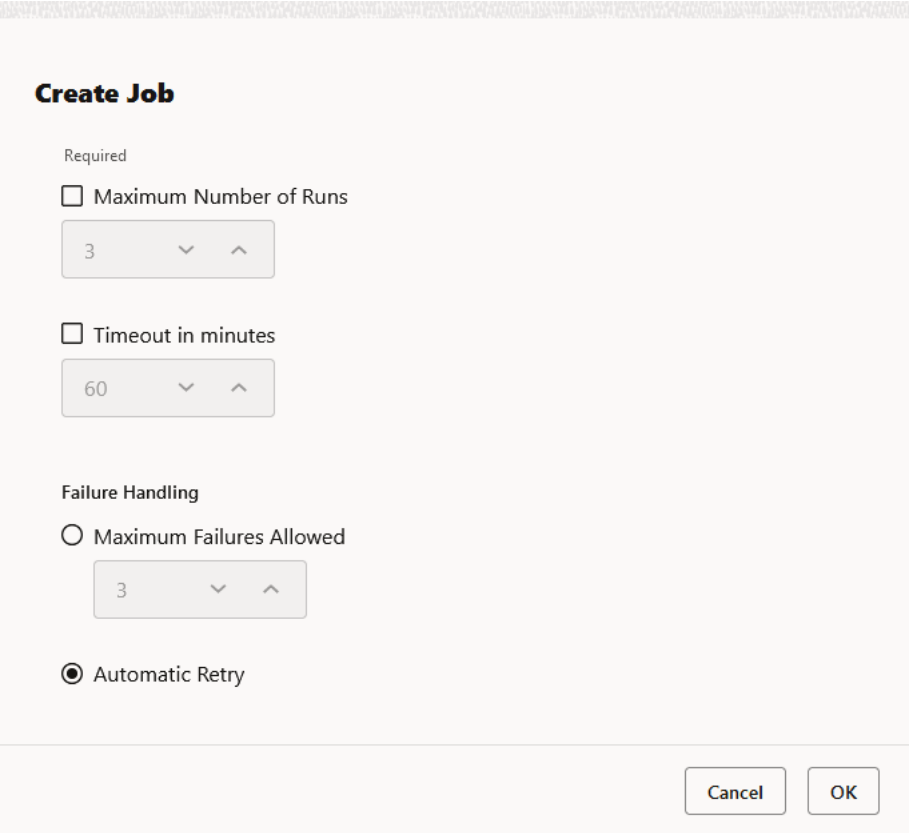
- **Email Address:** Enter the email ID of the users. By default, you can enter three email IDs, separated by comma. To send emails to more users, contact your Administrator.

  > **Note**
  >
  > The email notification feature is only available to the paid customer. This feature is not available in the free trial account.

- **Events:** Select the event for which you want to send the notification. The supported job events are:
  - `JOB_START`: Indicates that the job has started for the first time, or a job was started on a retry attempt.
  - `JOB_SUCCEEDED`: Indicates that the job has completed successfully.

  - `JOB_FAILED`: Indicates that the job resulted in an error, or was not able to run due to process death or database shutdown.
  - `JOB_BROKEN`: Indicates that the job is marked broken after retrying unsuccessfully. It also indicates failed job run after 16 attempts.
  - `JOB_COMPLETED`: Indicates that the job has a status of `COMPLETED` after it has reached its maximum number of runs or its end date.
  - `JOB_STOPPED`: Indicates that the job has terminated normally after a soft stop or hard stop was issued.
- **Maximum Number of Runs:** To specify the maximum number of times the job must run before it is stopped. When the job reaches the maximum run limit, it will stop.

Figure 12-4 Advanced Settings-2 in Create Jobs dialog



- **Timeout in Minutes:** To specify the maximum amount of time a job should be allowed to run.
- **Maximum Failures Allowed:** To specify the maximum number of times a job can fail on consecutive scheduled runs. When the maximum number of failures is reached, the next run date column in the Jobs UI will show an empty value to indicate the job is no longer scheduled to run. The Status column may show the status as `Failed`.
- **Automatic Retry:** Select this option if you want the job to restart in case the job fails.

7. Click **OK.**

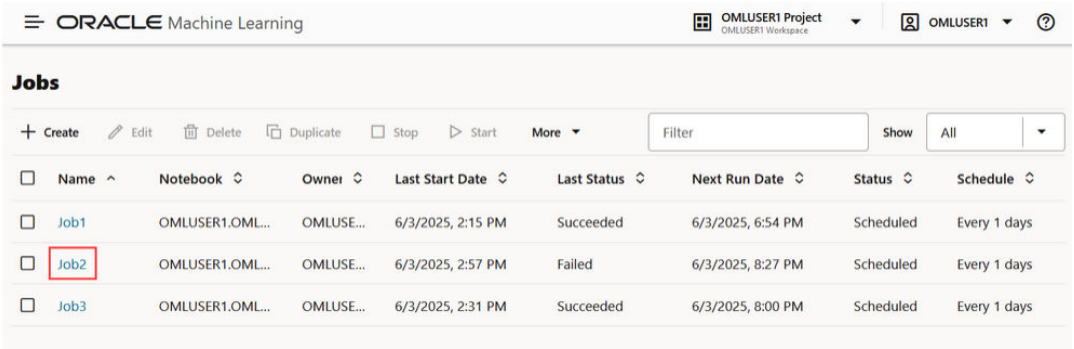
12.3 View Job Logs

You can view the historical logs of any particular job in the Job Log interface.

You can view a log in the read-only Notebook. To view job logs:

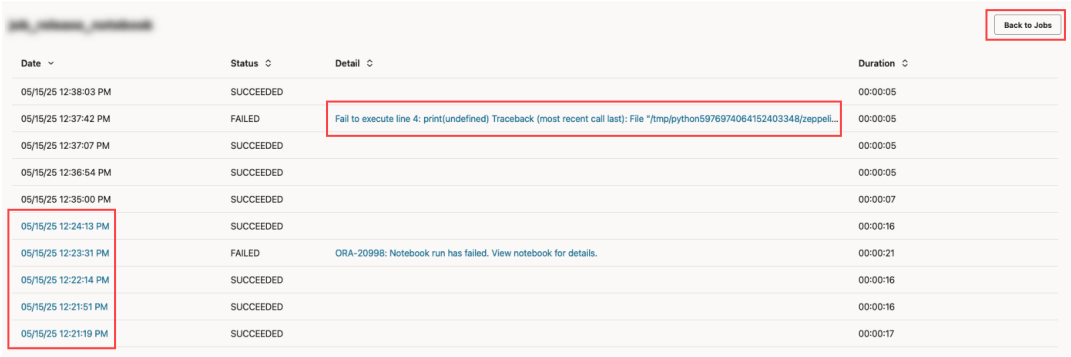
1. On the Jobs page, click on the job name for which you want to view the job logs. The logs for the specific job are displayed on the job log page.

Figure 12-5 Jobs page



2. The job logs page lists all the logs for the specific job you clicked. It displays the date, status, details of the error (if any), and the duration of the job run. For every job that fails, the **Detail** column lists the error message with a link. Click on the error link to view the complete error message. The old job logs are preserved and are accessible through the date column. Note the total number of logs are limited to 50. The metadata are available as clickable links under the **Date** column on this page, as shown in the screenshot here.

Figure 12-6 Job logs page



3. The complete error is displayed on a separate pane that slides in. Click X to close the pane.

Figure 12-7 Job log details and metadata

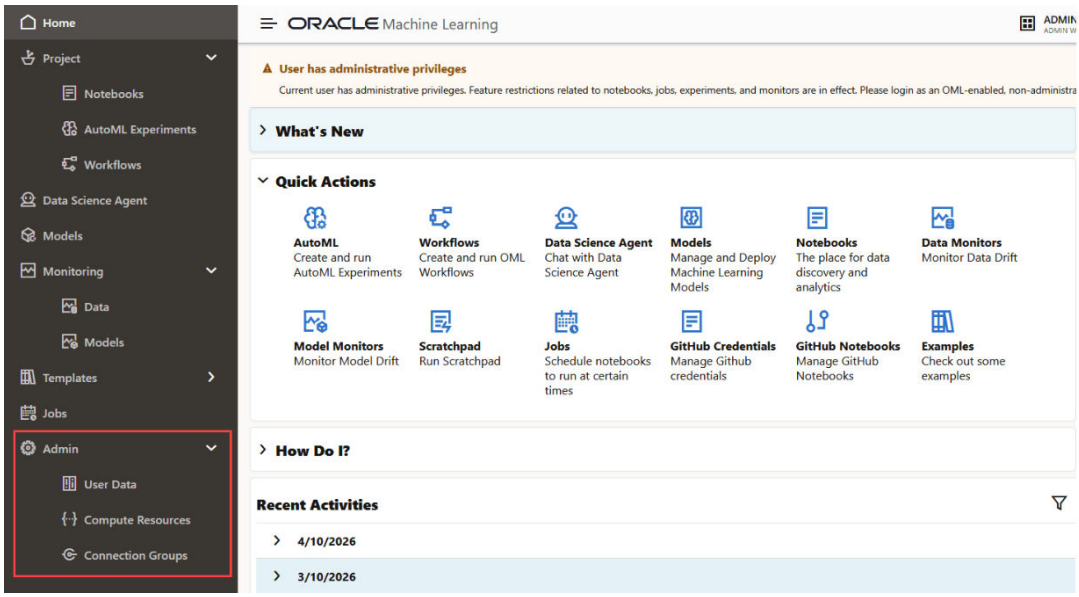
4. Click **Back to Jobs** on the job log page to return to the Jobs page.

13
Admin

Oracle Machine Learning is managed at the system level and at the application level by an administrator.

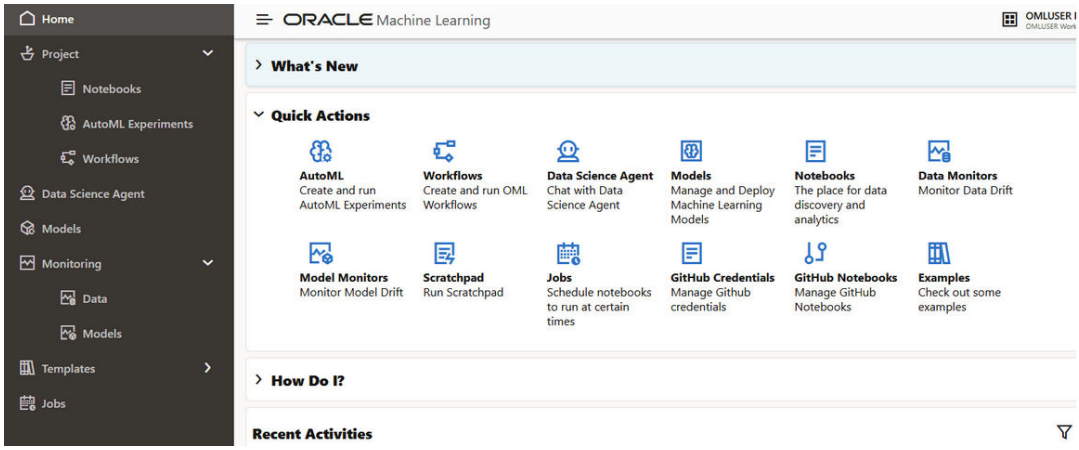
- Administrator — Creates and manages Oracle Machine Learning user accounts, manages compute resourses, connection groups, and notebook sessions. The Administrator also reassigns user workspace.

Figure 13-1 OML UI Admin home page and left navigation menu



- Developer — This is the default user role that allows you to perform these tasks:
  - Create and run AutoML experiments
  - Create conversations with Data Science Agent
  - Create and run machine learning workflows
  - Create and run notebooks. Here, you can run SQL Statements, SQL scripts, Python scripts, R scripts, Conda commands and Markdown
  - Deploy models
  - Create Data Monitor to monitor your data
  - Create Model Monitors to monitor your machine learning models
  - Create and run jobs to schedule and run notebooks
  - Use example template notebooks

Figure 13-2 OML UI Developer homepage



- Typical Workflow for Managing Oracle Machine Learning
  To manage Oracle Machine Learning User Interface and other administrative tasks, refer to the tasks listed in the table as a guide.
- Access OML User Management from Command Line
  You can obtain the Oracle Machine Learning User Management URL for a specific tenancy from the Oracle Cloud Infrastructure (OCI) command line.
- Manage OML Users and User Accounts
  An administrator manages new user account and user credentials creation for Oracle Machine Learning.
- About User Data
  On the User Data page in Oracle Machine Learning, you can view existing user data, reassign, and delete it.
- About Compute Resource
  The term Compute Resource refers to services such as a database, or any other backend service to which an interpreter connects.
- Get Started with Connection Groups
  A connection group, also known as a Zeppelin interpreter set, is a collection of database connections.
- Get Started with Notebook Sessions
  The Notebook Sessions page provides you an overview of your notebooks, and allows you to manage notebook sessions from your workspace or in workspaces where you have collaboration rights.

13.1 Typical Workflow for Managing Oracle Machine Learning

To manage Oracle Machine Learning User Interface and other administrative tasks, refer to the tasks listed in the table as a guide.

Tasks	Oracle Machine Learning Interface/OCI CLI Interface	More Information
Obtain Oracle Machine Learning User Management URL from OCI command line	Oracle Cloud Infrastructure (OCI) Command Line Interface (CLI)	Access OML User Management from Command Line
User account and password creation	Oracle Machine Learning User Management interface	Create Users for Oracle Machine Learning

Tasks	Oracle Machine Learning Interface/OCI CLI Interface	More Information
Connection Groups — View and Reset	Oracle Machine Learning User Interface	Work with Connection Groups
Compute Resource — View	Oracle Machine Learning User Interface	About Compute Resource
User Data administration — Delete all users, all user related objects such as workspace, projects, and notebooks, and workspace reassignment	Oracle Machine Learning User Interface	About User Data
Notebook session — Loading and stopping of notebook sessions	Oracle Machine Learning User Interface	Get Started with Notebook Sessions
Conda environment — Installation and management of the Conda environment, add and delete of packages from the environment.	Oracle Autonomous AI Database	About the Conda Environment and Conda Interpreter
Change the default number of email recipients for job notifications.	Create Jobs in Oracle Machine Learning User Interface	Change the variable `oml.emails.maxRecipients` in the `oml.conf` file. By default, a user can send email notification to 3 email addresses.

> **Note**
>
> The tasks listed here can be performed by an administrator only.

13.2 Access OML User Management from Command Line

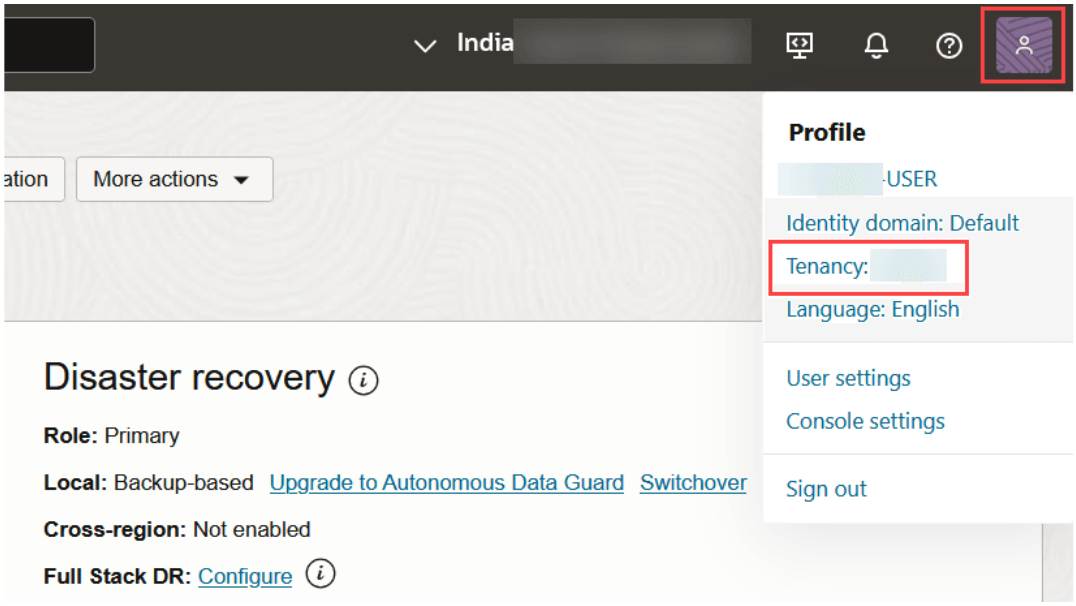
You can obtain the Oracle Machine Learning User Management URL for a specific tenancy from the Oracle Cloud Infrastructure (OCI) command line.

Prerequisite: Tenancy ID

To obtain the Oracle Machine Learning User Management URL for a specific tenancy from the OCI command line, you must first obtain the tenancy ID.

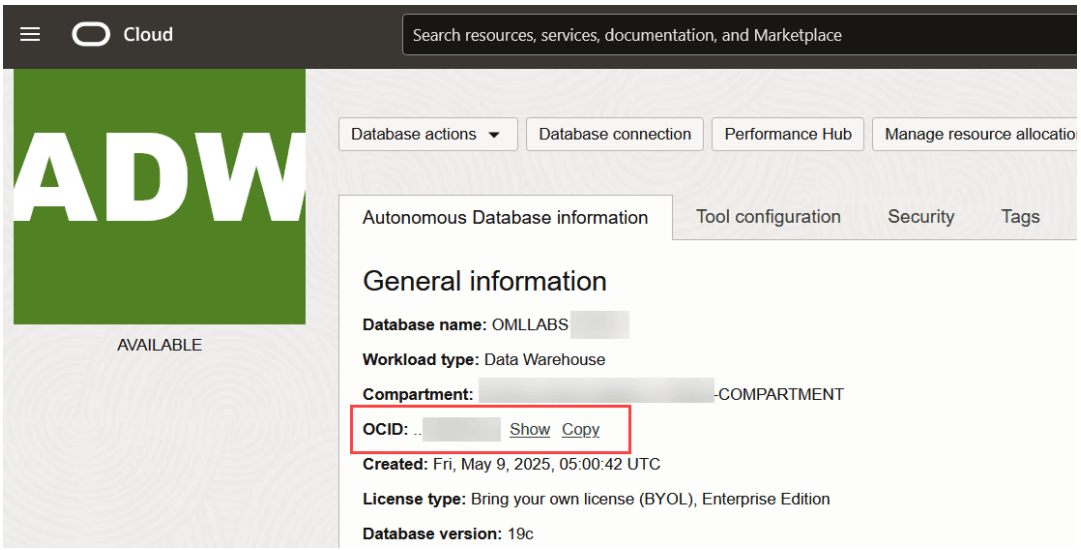
1. To obtain the tenancy ID, go to your OCI **Profile** on the top right corner of the Oracle Cloud page and click **Tenancy**.

Figure 13-3 Tenancy information in PCI Profile



2. On the Tenancy details page, click **Copy** to obtain the tenancy URL.

Figure 13-4 OCID



3. Type the following command in your OCI command line interface:

```
oci db database list  --compartment-id  <tenancy OCID>
```

Here,

- `compartment-id`: This is the unique ID assigned to your compartment.
- `OCID`: This is the Oracle Cloud Identifier (OCID) for your tenancy.

This command returns the following:

```
"connection-urls": {
       "apex-url": https://<tenancy ID>-<database
```

```
name>.<region>.oraclecloudapps.com/ords/apex,
        "graph-studio-url": https://<tenancy ID>-<database
name>.<region>.oraclecloudapps.com/graphstudio/,
        "machine-learning-user-management-url": https://<tenancy ID>-
<database name>.<region>-1.oraclecloudapps.com/omlusers/,
        "sql-dev-web-url": https://<tenancy ID>-<database
name>.<region>-1.oraclecloudapps.com/ords/sql-developer
      },
```

This completes the task of obtaining the Oracle Machine Learning User Management URL from OCI command line interface.

- Obtain URLs for OML User Management, REST API for OML Services, REST API for OML4Py and OML4R embedded execution from Command Line or the Autonomous AI Database
  You can obtain the URLs for OML User Management, REST API for OML Services, REST API for OML4Py and OML4R embedded execution using a query. This approach is useful for automation or when you need programmatic access to the URLs.

13.2.1 Obtain URLs for OML User Management, REST API for OML Services, REST API for OML4Py and OML4R embedded execution from Command Line or the Autonomous AI Database

You can obtain the URLs for OML User Management, REST API for OML Services, REST API for OML4Py and OML4R embedded execution using a query. This approach is useful for automation or when you need programmatic access to the URLs.

To obtain the URLs:

1. Use the `v$pdbs` view to run the query.

   > **Note**
   >
   > Non-admin users must have the `SELECT` privilege on `v$pdbs` to run this query. A DBA can grant this privilege using:

   ```
   GRANT SELECT ON v$pdbs TO <username>;
   ```

2. Run this query URLs from the database using the `v$pdbs`:

   ```
   SELECT 'https://' ||
          LOWER(REPLACE(p.name, '_', '-')) ||
          '.' ||
          REGEXP_REPLACE(j.PUBLIC_DOMAIN_NAME, '[^.]+', 'oraclecloudapps',
   1, 3) ||
          '/omlusers/' AS auth_token_url,
         'https://' ||
          LOWER(REPLACE(p.name, '_', '-')) ||
          '.' ||
          REGEXP_REPLACE(j.PUBLIC_DOMAIN_NAME, '[^.]+', 'oraclecloudapps',
   1, 3) ||
          '/oml/' AS embed_python_r_url,
         'https://' ||
   ```

```
        LOWER(REPLACE(p.name, '_', '-')) ||
        '.' ||
        REGEXP_REPLACE(j.PUBLIC_DOMAIN_NAME, '[^.]+', 'oraclecloudapps',
1, 3) ||
        '/omlmod/' AS rest_services_url
FROM v$pdbs p,
     JSON_TABLE(p.cloud_identity, '$' COLUMNS (
        PUBLIC_DOMAIN_NAME PATH
'$.PUBLIC_DOMAIN_NAME'
     )) j;
```

This query returns three URLs:

- OML User Management URL (same as returned by OCI CLI)
- URL for OML4Py and OML4R embedded execution
- URL for OML Services REST APIs

13.3 Manage OML Users and User Accounts

An administrator manages new user account and user credentials creation for Oracle Machine Learning.

Table 13-1 Administrative Tasks for OML Users and User Accounts

Tasks	Links
Create User Accounts for Oracle Machine Learning Components	Creating User Accounts for Oracle Machine Components
Create Users on Autonomous AI Database with Database Actions	Creating Users on Autonomous AI Database with Database Actions
Create Users on Autonomous AI Database - Connecting with a Client Tool	Creating Users with Autonomous AI Database with Client-Side Tools
Create Users on Autonomous AI Database	Create Users on Autonomous AI Database
Add Existing Database User Account to Oracle Machine Learning Components	Add Existing Database User Account to Oracle Machine Learning Components
Unlock User Accounts on Autonomous AI Database	Unlock User Accounts with Autonomous AI Database
About User Passwords on Autonomous AI Database	About User Passwords on Autonomous AI Database

- Add Existing Database User Account to Oracle Machine Learning Components
  As the ADMIN user you can add an existing database user account to provide access to Oracle Machine Learning components.

Related Topics

- Manage Users

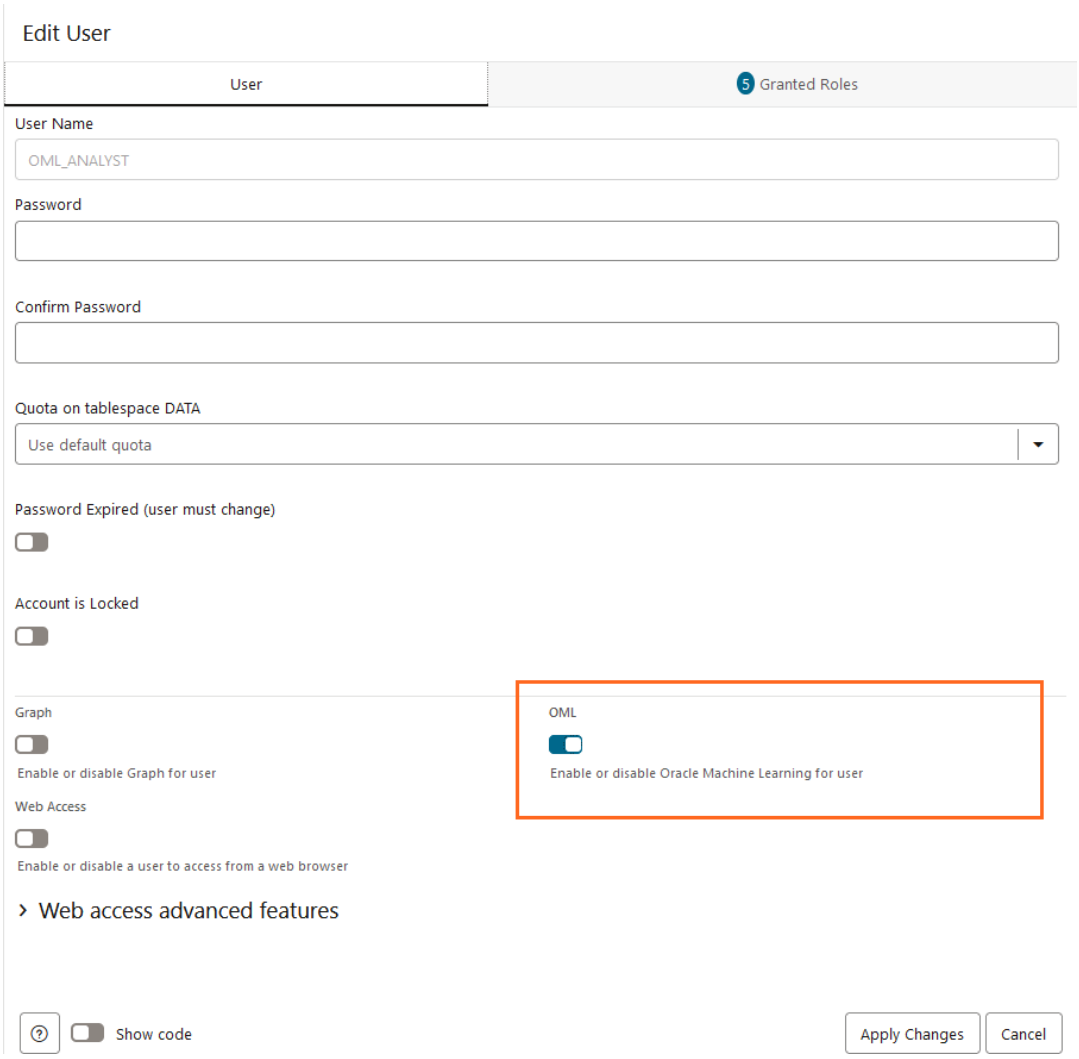
13.3.1 Add Existing Database User Account to Oracle Machine Learning Components

As the ADMIN user you can add an existing database user account to provide access to Oracle Machine Learning components.

To add an existing database user account:

1. On the Autonomous AI Database page, under the **Display Name** column, select an Autonomous AI Database.
2. On the Autonomous AI Database Details page, select **Database Actions** and click **Database Users**.
3. In the All Users, search for the user of interest or select the user. For example, search the user **OML_ANALYST**.
4. In the user's card, click ··· and select **Edit**
5. In the Edit User panel, select **OML**.

   For example:



6. Click **Apply Changes**.

This grants the required privileges to use the Oracle Machine Learning application. In Oracle Machine Learning this user can then access any tables the user has privileges to access in the database.

13.4 About User Data

On the User Data page in Oracle Machine Learning, you can view existing user data, reassign, and delete it.

The User Data page lists details of the Oracle Machine Learning user such as the name, role, comments, last updated date. You can perform the following tasks:

- **Delete User Data:** To delete a user, select the user to delete and click **Delete User Data.**
- **Reassign:** To reassign workspace and templates from one user to another.
- Reassign
  The **Reassign** option allows you to reassign workspaces, along with templates, from one user to another.

13.4.1 Reassign

The **Reassign** option allows you to reassign workspaces, along with templates, from one user to another.

To reassign workspaces:

1. On the User Data page, select the user from whom you want to reassign workspace and click **Reassign.**

   The Reassign page opens.
2. In the **Target User** field, select the user to whom you want to reassign workspace.
3. Select **All Templates** if you want to reassign all the templates associated with the user selected in the User Data page.
4. Select:
   - **Reassign all workspaces:** To reassign all the workspaces associated with the selected user.
   - **Select workspaces to reassign:** To reassign particular workspaces associated with the selected user.
5. Click **Reassign.**

After the templates and workspaces are reassigned successfully, a notification message is displayed on the User Data page with the number of templates and workspaces reassigned.

13.5 About Compute Resource

The term Compute Resource refers to services such as a database, or any other backend service to which an interpreter connects.

> **Note**
>
> You must have the Administrator role to access the Compute Resources page.

The Compute Resources page displays the list of compute resources along with the name of each resource, its type, comments, and last updated details. To view details of each Compute Resource, click the Compute Resource name. The connection details are displayed on the **Oracle Resources** page.

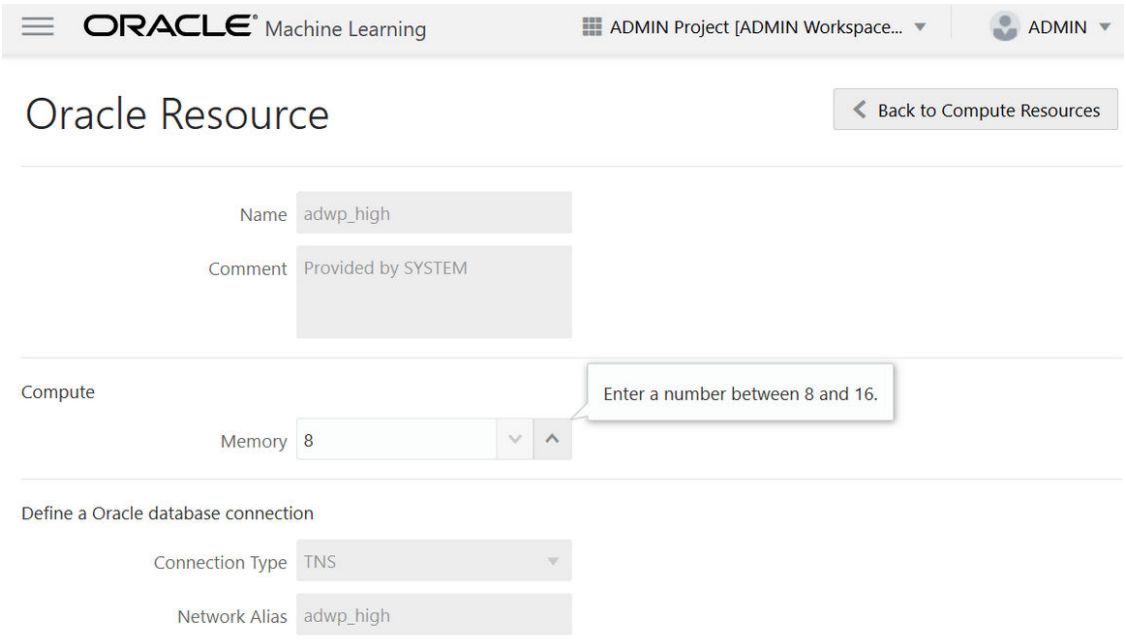
- Oracle Resource
  The Oracle Resource page displays the details of the selected compute resource on the Compute Resources page. You can configure the memory settings (in Gigabytes) for the Python interpreter for the selected compute resource.

13.5.1 Oracle Resource

The Oracle Resource page displays the details of the selected compute resource on the Compute Resources page. You can configure the memory settings (in Gigabytes) for the Python interpreter for the selected compute resource.

> **Note**
>
> You must have Administrator privilege to configure the memory settings.

Figure 13-5 Oracle Resource



To manage memory settings for the interpreter:

1. **Name:** Displays the name of the selected resource.

2. **Comment:** Displays comment, if any.
3. **Memory:** You can configure memory settings (in Gigabytes) for the interpreter in this field. The interpreter supports Markdown, Python, SQL, Script, and R languages.
   - For the resource `databasename_gpu`, the memory settings (in Gigabytes) must be between 8 and 200. The memory setting for `gpu` configures the amount of host RAM that the interpreter container can use. The GPU VRAM is not configurable and the container has access to all GPU memory available. For NVIDIA A10 Tensor Core GPUs, it is 24GB.
   - For the resource `databasename_high`, the memory settings (in Gigabytes) must be between 8 and 96.
   - For the resource `databasename_medium`, the memory settings (in Gigabytes) must be between 4 and 8.
   - For the resource `databasename_low`, the memory settings (in Gigabytes) must be between 2 and 4.

   > **Note**
   >
   > The Memory setting is applicable only for the Python interpreter.

4. **Connection Type:** Displays the database connection of the resource.
5. **Network Alias:** Displays the alias of the network connection.

- Resource Services and Notebooks
  This topic lists the number of notebooks that you can run concurrently per Autonomous AI Database instance for each resource service.

13.5.1.1 Resource Services and Notebooks

This topic lists the number of notebooks that you can run concurrently per Autonomous AI Database instance for each resource service.

The Resource Services and Number of Notebooks table lists the Compute Resources assigned for running at different Resource Service levels - GPU, High, Medium and Low. The GPU compute capability applies only to the Python interpreter.

Table 13-2 Resource Services and Number of Notebooks

Resource Service	ECPUs and GPUs	Memory	Number of Concurrent Notebooks, UDFs
GPU	1 NVIDIA A10 Tensor Core **Note** The GPU setting includes a HIGH setting on the database	8 GB (DDR4), by default. Extensible up to 200 GB	The number of concurrent notebooks you can run is determined by: • The GPU resources of the region where your ADB instance is deployed, and • The number of GPU resources available at the time you run the notebooks If GPU resources are not available when requested, you will receive an error message. You should try again later. **Note** GPU resources are available only on paid Oracle Autonomous AI Database Serverless. GPU resources are not available if less than 16 ECPUs are allocated for OML.

Table 13-2 (Cont.) Resource Services and Number of Notebooks

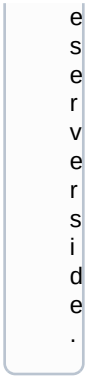
Resource Service	ECPUs and GPUs	Memory	Number of Concurrent Notebooks, UDFs
			
High	Up to 8	8 GB (up to 16 GB)	Up to 3

Table 13-2 (Cont.) Resource Services and Number of Notebooks

Resource Service	ECPUs and GPUs	Memory	Number of Concurrent Notebooks, UDFs
Medium	Up to 4	4 GB (up to 8 GB)	Up to max (1.25 × number of ECPUs)
Low	1	2 GB (up to 4 GB)	Up to 100
TP This service is available for Oracle Autonomous Transaction Processing (ATP) database.	User specified	2 GB	Up to 60
TPURGENT This service is available for Oracle Autonomous Transaction Processing (ATP) database.	User specified	2 GB.	Up to 60

Note

The number of current notebooks run is calculated by the formula `1.25 x (number of ECPUs)` provisioned for the correspondi ng Autonomous AI Database instance. For example, if a database is provisioned with 4 ECPUs, then the maximum number of notebooks run would be 5 (1.25 x 4) in Medium level.

Table 13-2 (Cont.) Resource Services and Number of Notebooks

Resource Service	ECPUs and GPUs	Memory	Number of Concurrent Notebooks, UDFs
ECPU setting. OML apps on ADB-Serverless have ECPU specifications separate from the database.	User specified	• Low - 2GB • Medium - 4GB • High - 8GB • GPU - 8GB (default). Can be extended upto 200GB by the Admin. This allocation is based on the assumption that one VM is allocated for the PDB.	All processes share the CPU resources. Running of UDFs is situation-specific. • If you are performing data processing at the compute level, you may require more memory depending on your data size. **Note** The Admin can allocate more memory in Oracle Resource. • If Low resource level is sufficient, you may be able to run approximately 60 UDFs concurrently. • If High resource level is required, you may be able to run approximately 16 UDFs concurrently.

For more information on Database service and concurrency, see Database Service Names for Autonomous AI Database

13.6 Get Started with Connection Groups

A connection group, also known as a Zeppelin interpreter set, is a collection of database connections.

- About Connection Groups
  On the Connection Group page, a user with Administrator role can manage your connections that constitute the connection group.
- About Global Connection Group
  The Global Connection Group is created automatically when a new database is provisioned.
- Edit Oracle AI Database Interpreter Connection
  When defining an Oracle AI Database interpreter connection, a reference to a compute resource is created. This reference contains all connection-related information about the interpreter.

13.6.1 About Connection Groups

On the Connection Group page, a user with Administrator role can manage your connections that constitute the connection group.

You can **Edit,** and **Stop** one or more connections that are listed under a connection group on this page.

> **Note**
>
> Only an Administrator user can manage connection groups.

The following information about the connections are available:

- **Name:** This is the name of the interpreter.
- **Default:** A check mark indicates whether the connection is the default connection or not.
- **Scope:** Indicates the scope of the connection.
- **Comment:** Displays any comment related to the interpreter.
- **Owner:** Displays the name of the user who created the connection.
- **Last Updated:** Indicates the date and time when the connection was last updated.

You can perform the following tasks:

- **Edit:** To edit the interpreter connection, select the connection and click **Edit.**
- **Stop:** To stop the interpreter connection, select the connection and click **Stop.**
- **Refresh:** Click the **Refresh** button in the following conditions:
  - If you rename the Pluggable Database (PDB).
  - If you do a Wallet rotation. Wallet rotation invalidates the current wallet. Hence, a new Wallet is needed for the database connection.

13.6.2 About Global Connection Group

The Global Connection Group is created automatically when a new database is provisioned.

The Global Connection Group comprises the following:

- **Compute Resource definition:** A Compute Resource is associated with the Pluggable Database (PDB). After a new PDB is provisioned, a Compute Resource is added for the PDB. A tenant may provision more than one PDB, and for each PDB a Compute Resource is added. The settings in the Compute Resource are relevant to its own PDB. The Compute Resource is associated to an Oracle Wallet. The Oracle wallet contains the credentials to connect to the user PDB.

  > **Note**
  >
  > The Compute Resource definition can be edited by the Administrator only.

- **Connection Group definition:** The Global Connection Group comprises a single connection of type `Global`. Only one Global Connection Group for each Compute Resource is allowed per PDB. No password is required for this connection as it uses the Wallet containing the credentials for the PDB. The Wallet is associated to the Compute Resource.

  > **Note**
  >
  > A Global Connection Group can be edited by the Administrator only.

Reset: To reset the interpreter connection, click the connection group name. The connection group opens on a separate page, listing all the interpreter connections in the group. Select the connection you want to reset and click **Reset.** When you click **Reset,** then all connections supported by the interpreter are closed, and all notebooks using that connection are canceled.

> **Note**
>
> The **Reset** option is available only to the Administrator.

13.6.3 Edit Oracle AI Database Interpreter Connection

When defining an Oracle AI Database interpreter connection, a reference to a compute resource is created. This reference contains all connection-related information about the interpreter.

Compute Resources for an Oracle AI Database interpreter is defined by your service. You can edit the following:

> **Note**
>
> You must have the Administrator role to edit these fields.

1. **Name:** You can edit the name of the interpreter editor here. This is useful if you have several definitions of the same interpreter type in the same interpreter set. By specifying a name, you can turn on or turn off the specific binding to a notebook.
2. **Type:** This is a non-editable field. It indicates the connection type
3. **Binding Mode:** This is a non-editable field. It defines the behavior of the interpreter instance in memory, and how the resources are shared. By default, the Binding Mode of the Global Connection Group is set to Scoped. It ensures that each notebook creates a new interpreter instance in the same interpreter process.
4. **Row Render Limit:** This determines the number of rows to be displayed in the paragraph results when fetching a data structure that can be presented as a table or graph using the Zeppelin built-in plotting service. You must consider the browser capabilities when modifying this setting. The default limit is `1000`.

   > **Note**
   >
   > Zeppelin plotting service works with data that is fetched previously to the client-side for a snapper UI.

5. **Comments:** Enter any information related to the interpreter not exceeding `1000` characters.

   > **Note**
   >
   > You must have Administrator role to edit this field.

6. In the Compute Resource section, the **Resources** field indicates the priority of the compute resource. This is a non-editable field.
7. In the Database section, you can specify additional settings related to PL/SQL DBMS output. Select **Enabled** to allow the PL/SQL interpreter to display the messages sent to the DBMS_OUTPUT in the paragraph results.
8. Click **Save.**

13.7 Get Started with Notebook Sessions

The Notebook Sessions page provides you an overview of your notebooks, and allows you to manage notebook sessions from your workspace or in workspaces where you have collaboration rights.

On the Notebook Sessions page, you unload and cancel notebook sessions. You can perform the following tasks:

- **Stop:** Select the notebook that is running, and click **Stop.** This stops the selected notebook in the server.
- **Unload:** Select the notebook that is loaded, and click **Unload.** This removes the selected notebook from memory on the server.

The Notebook Sessions page displays the following information about your notebooks:

- Notebook: The name of the notebook.
- Project: The project in which the notebook resides.
- Workspace: The workspace in which the project is available.
- Connection: The connection name.
- Owner: The owner of the notebook.
- Status: The statuses of a notebook are:
  - **Loaded:** Indicates that the notebook is loaded but not tied to the websocket or running.
  - **Active:** Indicates that the notebook is tied to the websocket but is not running.
  - **Running:** Indicates that the notebook paragraph is queued to run or is running.

A.1 FAQs

Topics:

A.1.1 How to grant access to an OML model by giving global or selective access to other users?

Use the grant `SELECT` access on in-database models for inferencing. For a specific database user, use:

```
GRANT SELECT ON MINING MODEL <model name> TO
    username;
```

A.1.2 How to obtain the URL for REST API for OML Services, REST API for OML4Py and OML4R embedded execution from the Autonomous AI Database ?

To obtain the OML URL, SQL Developer URL, Graph Studio URL, APEX URL

You can obtain the Oracle Machine Learning User Management URL for a specific tenancy from the Oracle Cloud Infrastructure (OCI) command line. You must first obtain the tenancy ID.

1. To obtain the tenancy ID, go to your OCI Profile on the top right corner of the Oracle Cloud page and click **Tenancy**.

2. On the Tenancy details page, click **Copy** to obtain the tenancy URL.
3. Type the following command in your OCI command line interface:

   ```
   oci db database list  --compartment-id  <tenancy OCID>
   ```

   Here,

   - `compartment-id`: This is the unique ID assigned to your compartment.
   - `OCID`: This is the Oracle Cloud Identifier (OCID) for your tenancy.
4. This command returns the following:

   ```
   "connection-urls": {
           "apex-url": https://<tenancy ID>-<database
   name>.<region>.oraclecloudapps.com/ords/apex,
           "graph-studio-url": https://<tenancy ID>-<database
   name>.<region>.oraclecloudapps.com/graphstudio/,
           "machine-learning-user-management-url": https://<tenancy ID>-
   <database name>.<region>-1.oraclecloudapps.com/omlusers/,
           "sql-dev-web-url": https://<tenancy ID>-<database
   name>.<region>-1.oraclecloudapps.com/ords/sql-developer
         },
   ```

For more information, see Access OML User Management from Command Line

To obtain the URLs for OML User Management, REST API OML Services, REST API for OML4Py and OML4R embedded execution from Command Line or Autonomous AI Database

Use the view `v$pdbs` and run the following:

```
select 'https://' ||
        lower(REPLACE(p.name, '_', '-')) ||
        '.' ||
        REGEXP_REPLACE(j.PUBLIC_DOMAIN_NAME, '[^.]+', 'oraclecloudapps', 1,
3) ||
        '/omlusers/' auth_token_url,
        'https://' ||
        lower(REPLACE(p.name, '_', '-')) ||
        '.' ||
        REGEXP_REPLACE(j.PUBLIC_DOMAIN_NAME, '[^.]+', 'oraclecloudapps', 1,
3) ||
        '/oml/' embed_python_r_url,
        'https://' ||
        lower(REPLACE(p.name, '_', '-')) ||
        '.' ||
        REGEXP_REPLACE(j.PUBLIC_DOMAIN_NAME, '[^.]+', 'oraclecloudapps', 1,
3) ||
        '/omlmod/' rest_services_url
from v$pdbs p,
     json_table(p.cloud_identity, '$' columns (
        PUBLIC_DOMAIN_NAME path
'$.PUBLIC_DOMAIN_NAME'
     )) j;
```

This command works for non-private endpoint databases. Users that create their own URL behind a private endpoint could modify it.

This query returns three URLs:

- OML User Management URL (same as returned by OCI CLI)
- URL for OML4Py and OML4R embedded execution
- URL for OML Services REST APIs

To obtain the URL for OML REST Services from the Autonomous AI Database

You can see the URLs for your database in Database Actions

1. On your Oracle Autonomous AI Database console, click **Database actions**, and then select the option **View all database actions**.
2. Click **RESTFul Services** and then click **Oracle Machine Learning RESTful services**. The Oracle Machine Learning RESTful Services dialog opens.
3. On the Oracle Machine Learning RESTful Services dialog, copy the URL for your ADB instance.

For more information, see Task 2: Get the OML Services URL to Obtain Your REST Authentication Token

A.1.3 How to push content represented by an OML proxy object into a table in the Oracle Autonomous AI Database?

From a table in Autonomous Data Warehouse , in Oracle Machine Learning notebook we have converted the table data as `'oml.core.frame.DataFrame'` and using my model have done the predictions. Now my predicted column is in `'oml.core.frame.DataFrame'`. How can I insert the predicted column back into the database table?

For this, use the `materialize()` function. This function pushes the contents represented by an OML proxy object into a table in the database.

```
proxy_object.materialize(table="MYTABLENAME")
```

A.1.4 What is the expected behavior if the embedded REST execution requests surpass the capacity of the VM?

What is the expected behavior if the embedded REST execution requests surpass the capacity of the VM? Are they enqueued or does the request fail?

The following error is displayed: `Unable to reserve a STANDARD resource at this time: no STANDARD type VM is available. Please try again later`

Embedded REST execution calls would fail if no container resources are available to serve.

A.1.5 Who owns and manages the VMs on which embedded Python execution is downloaded?

Is the VM to which the embedded Python execution workload is offloaded, owned and managed by OML (service-owned)? Or is it under the ownership of the database itself (customer-owned)?

For Oracle Machine Learning, each tenant has its own VM managed by Autonomous AI Database. Embedded execution does not run on customer-owned VM. Multiple VMs might belong to a tenant but no VM can be shared across different tenants. Through the service levels - `Low`, `Medium`, `High` and `GPU` settings, the different sized containers can be provisioned respectively. There is no dynamic resource adjustment in place at runtime.

A tenant may provision one or more databases. Oracle Autonomous AI Database Serverless maintains a pool of VMs. Each embedded execution is processed in a container, which runs on a VM. When we need a VM to run a container, we request a VM from Oracle Autonomous AI Database Serverless. The VM is released when no more containers are running. Currently, a VM only serves one database at a time. All database users share this VM). If a tenant has 2 databases, each database will have its own VM.

> **Note**
>
> Oracle Autonomous AI Database Serverless does not support dynamic resource adjustment for the containers.

A.1.6 Where can I find the list of all the functions available in the oml python package?

For a list of overloaded transparency layer functions, run:

```
%python

import pandas as pd
import oml

# Create a simple Pandas DataFrame
df = pd.DataFrame({
    'col1': [1, 2, 3],
    'col2': ['a', 'b', 'c']})

# Create an OML proxy object
oml_df = oml.create(df, "df")

# List all public methods and attributes
methods = [m for m in oml_df.__dir__() if not m.startswith('_')]
methods.sort()
print(methods)
```

For more information, see OML4Py API Reference Guide

A.1.7 What is the difference between the OML ECPUs and the Database ECPUs?

The OML ECPUs are the app-level ECPUs. For example, OML Notebooks use resources allocated with the OML App ECPUs. The in-database ML algorithms use the ADB (server side) ECPU resources.

A.1.8 What is an alternative solution to oml.connect() for Oracle Machine Learning solutions deployment?

For Oracle Machine Learning solutions deployment, every UDF has its own `oml.connect()`. Instead of hard-coding credentials into the `oml.connect` statement, is there an alternative solution that can work on multiple database environments?

- For the Python API for embedded execution, use `oml_connect=True`
- For SQL API, use `"oml_connect":1`
- For more information, see pyqTableEval Function (On-Premises Database)

A.1.9 What other models can I host on OML besides the in-database models?

Besides OML's in-database models, you can also host your own models on Oracle Machine Learning.

- Native Python and R models can be stored using the OML4Py and OML4R datastores. You can call these models through Python, R, SQL, and REST endpoints, optionally in parallel.
- ONNX format classification, regression, feature extraction, and clustering models can be imported and use the same SQL prediction operators as native models. You can also deploy these models to OML Services for real-time scoring through REST endpoints.
- Transformer models from Hugging Face can be converted to ONNX format, augmented through a pipeline, and loaded to the database for use with AI Vector Search.
- LLMs can be accessed through Select AI for SQL generation, RAG, and chat.

For more information, see:

- Enhance your AI/ML applications with flexible Bring Your Own Model options
- Bring Your Own Model to Your Database for AI and ML

A.1.10 Does Oracle Machine Learning support reading of data from OCI Streaming services or Kafka?

Oracle Machine Learning does not directly consume data from OCI Streaming or Kafka since it works with database tables. But there are several ways to get streaming data into tables for OML to use. You could use:

- OCI GoldenGate for real-time sync to Oracle Autonomous AI Database
- OCI Connector Hub along with OCI Functions to process and load stream data
- OCI Functions triggered by streaming events

This is the workflow:

1. Use data from OCI Streaming
2. Load data into Oracle Autonomous AI Database tables
3. Use Oracle Machine Learning to build models on those tables using Oracle Machine Learning for SQL, Oracle Machine Learning for R, or Oracle Machine Learning for Python

Once the streaming data is loaded in your database, Oracle Machine Learning can work with it just like any other database tables.

A.1.11 How can I create a Conda environment for Python and R, install packages and upload the environment in a single step

Create a Conda Environment for Python, Install Python Packages and upload the environment to object storage

To create a Conda environment, install a Python package, and upload the environment to object storage in a single step, run the following command in a Conda paragraph:

```
%conda

create -n mypyenv -c conda-forge --override-channels --strict-channel-
priority python=3.13.5 seaborn

upload mypyenv --overwrite -t application "OML4PY"
```

> **Note**
>
> This example uses Python version 3.13.5. You must ensure compatibility between the third-party package version and the Python version that OML4Py uses.

To determine the python version being used, run the following command:

```
import sys
sys.version
```

In this command:

- `-n`: This is the name of the environment. In this example, it is `mypyenv`.
- `-c`: This is the channel name. In this example, the `conda-forge`channel is used to install the Python package.
- `--override-channels`: This argument ensures that the system does not search default, and requires a channel to be mentioned.
- `--strict-channel-priority`: This argument ensures that packages in lower priority channels are not considered if a package with the same name appears in a higher priority channel. In this example, the priority is given to Python 3.13.5 and seaborn.
- `upload`: This argument uploads the Conda environment named `mypyenv` to a target location.
- `--overwrite`: This argument ensures that if a file or directory with the same name already exists at the target location, it should be overwritten.
- `-t application`: This argument specifies the type of the target as an application.
- `"OML4PY"`: This is the name or path of the target application where the environment will be uploaded.

For more information, see: Create a Conda Environment for Python, Install a Python Package and Upload it to Object Storage

Create a Conda Environment for R, Install R Packages and upload the environment to object storage

To create a Conda environment, install a R package and upload the environment to object storage in a single step, run the following command in a Conda paragraph:

```
%conda

create -n myrenv -c conda-forge --override-channels --strict-channel-priority
r-base=4.0.5 r-forecast

upload myrenv --overwrite -t application "OML4R"
```

> **Note**
>
> In this example, we are using the `conda-forge` channel to install the R packages.

In this command:

- `-n`: This is the name of the environment. In this example, it is `myrenv`.
- `-c`: This is the channel name. In this example, it is `conda-forge`.
- `--override-channels`: This argument ensures that the system does not search default, and requires a channel to be mentioned.
- `--strict-channel-priority`: This argument ensures that packages in lower priority channels are not considered if a package with the same name appears in a higher priority channel. In this example, the priority is given to `R-4.0.5` and `forecast`.
- `upload`: This argument uploads the Conda environment named `myrenv` to a target location.

- `--overwrite`: This argument ensures that if a file or directory with the same name already exists at the target location, it should be overwritten.
- `-t application`: This argument specifies the type of the target as an application.
- `"OML4R"`: This is the name or path of the target application where the environment will be uploaded.

For more information, see Create a Conda Environment for R and Install R Package and Upload it to Object Storage

A.2 Troubleshooting

A.2.1 OML4Py

A.2.1.1 Error when importing matplotlib

Issue: Error occurs when running this command to import `matplotlib` in a notebook:

```
%python

import matplotlib.pyplot as plt
plt.style.use('seaborn')
plt.figure(figsize=[10,5])

oml.graphics.boxplot(IRIS[:, :4], notch=True,
                     showmeans = True,
                     labels=IRIS.columns[:4])

plt.title('Distribution of IRIS Attributes')

plt.ylabel('cm')
```

Error message:

```
Fail to execute line 2: plt.style.use('seaborn')

Traceback (most recent call last):

  File "/usr/local/lib/python3.12/site-packages/matplotlib/style/core.py", line 137, in
use

    style = _rc_params_in_file(style)

            ^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/site-packages/matplotlib/__init__.py", line 866, in
_rc_params_in_file

    with _open_file_or_url(fname) as fd:
```

```
  File "/usr/local/lib/python3.12/contextlib.py", line 137, in __enter__

    return next(self.gen)

           ^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/site-packages/matplotlib/__init__.py", line 843, in
_open_file_or_url

    with open(fname, encoding='utf-8') as f:

         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

FileNotFoundError: [Errno 2] No such file or directory: 'seaborn'....

....

OSError: 'seaborn' is not a valid package style, path of style file, URL of style file,
or library style name (library styles are listed in `style.available`)
```

Resolution: This error occurred due to a change that was introduced in `Matplotlib 3.7.0`. It affects all Python `3.12.x` versions. Specifically, using `plt.style.use('seaborn')` was deprecated in `Matplotlib 3.6` and removed in `3.7`. You will encounter this error if you're using Python 3.12.x (any point version).

> **Note**
>
> Ensure that you import `Matplotlib ≥3.7.0`.

A.2.1.2 Error when importing Python interpreters

Issue: The error “Fail to open Python Interpreter” encountered in OML notebooks.

Resolution: This issue is related to version conflict. To avoid version conflicts, do not install packages that are already included in OML4Py. This issue is related to version conflict. For a complete list of pre-installed packages and their versions, to view the list of Python interpreters, run the following:

```
%python
import os os.listdir('/usr/local/lib/python3.12/site-packages')
```

A.2.1.3 How to add HostAce using cursor in Python?

To add HostAce using cursor in Python, run the following command in a Python paragraph:

```
%python
cursor.execute("BEGIN pyqAppendHostAce('USER_OML',
'<adb_instance>.oraclecloudapps.com');
```

A.2.1.4 Error when installing OML4Py Server for On-Premises Oracle AI Database

Issue: Encountered the following error when installing OML4Py Server for On-Premises Oracle AI Database

```
SQL> SELECT * FROM sys.pyq_config;  NAME          VALUE
      ____________ ____________________________________________________
PYTHONHOME
      /opt/oracle/product/26ai/dbhomeFree/python             PYTHONPATH
      /opt/oracle/product/26ai/dbhomeFree/oml4py/modules     VERSION       2.0
      PLATFORM      ODB
DSWLIST
      oml.*;pandas.*;numpy.*;matplotlib.*;sklearn.*           SQL> begin   2
      sys.pyqScriptCreate('pyqFun1', 'func = lambda: "Hello World from a
lambda!"', FALSE, TRUE);
      3  end;   4  /  PL/SQL procedure successfully completed.  SQL> SELECT
name, value FROM
      table(pyqEval(NULL,'XML','pyqFun1')); SELECT name, value FROM
      table(pyqEval(NULL,'XML','pyqFun1')) * ERROR at line 1: ORA-20000:
PyQuery error ImportError:
      Error importing numpy: you should not try to import numpy from its
source directory; please
      exit the numpy source tree, and relaunch your python interpreter from
there. ORA-06512: at
      "PYQSYS.PYQ$EVALIMPL_IN", line 77 ORA-06512: at
"PYQSYS.PYQ$EVALIMPL_IN", line 74
      ORA-06512: at "SYS.DBMS_SQL", line 1838 ORA-06512: at
"PYQSYS.PYQ$ETSTART", line 176
      ORA-06512: at "PYQSYS.PYQEVALIMPL", line 54 ORA-06512: at line 1
```

Resolution: This error usually occurs if you are running Python from inside the Numpy directory. To resolve this issue, ensure that the working directory in your SQL API session is not pointing to a local `numpy/` folder.

1. Do `cd ..` to exit the `numpy` directory.
2. Now, follow the steps to install OML4Py Server for On-Premises Oracle AI Database again.

A.2.2 OML4R

A.2.2.1 How to add rows to a table in the database through an ore.frame

To modify the database table:

1. Use the Oracle R package.

2. Then run an `INSERT` into command to add the rows to the table.

A.2.2.2 Unable able to push an object to Autonomous AI Lakehouse from an Oracle Machine Learning Notebook session using OML4R equivalent of OML4Py

Issue: Used this command to push an object to Autonomous AI Lakehouse.

The Oracle Machine Learning for Python command that works:

```
oml_iris = oml.create(iris_df, table = 'IRIS_OML4PY')
```

Error

```
Error in .oci.GetQuery(conn, statement, data = data, prefetch = prefetch, : ORA-06598:
insufficient INHERIT PRIVILEGES privilege ORA-06512: at
"RQSYS.RQ$ADDRSESSIONREFDBOBJECT", line 1 ORA-06512: at line 1 Traceback: 1.
ore.push(iris_df, table = "IRIS_OML4R") 2. ore.push(iris_df, table = "IRIS_OML4R")
3. .ore.addObjects(env, tabName, "table", "purge")
4. .ore.QueryEnv$addRefObjects(objName, objType, objParm) 5. .ore.dbGetQuery(stmt, data
= refdbobjs) 6. .ore.QueryEnv$dbGetQuery(qry, noframe = noframe, return = return, . ...)
7. ROracle::dbGetQuery(.ore.con(), qry, ...) 8. ROracle::dbGetQuery(.ore.con(),
qry, ...) 9. .local(conn, statement, ...) 10. .oci.GetQuery(conn, statement, data =
data, prefetch = prefetch, . bulk_read = bulk_read, bulk_write = bulk_write)
```

Resolution: Run this command instead:

```
oml.create != ore.push
```

> **Note**
>
> Oracle Machine Learning for Python has its own `push` function. Moreover, the `push` function does not take a table argument.

A.2.3 OML Notebooks

- How to install third-party packages for both Python and R in Oracle Machine Learning Notebooks
- How to upload data from a .csv file into a Pandas DataFrame
- "Allow Run" button to warn users about potential security threat
- Resource Management in Oracle Machine Learning

A.2.3.1 How to install third-party packages for both Python and R in Oracle Machine Learning Notebooks

How to install third-party packages for both Python and R in Oracle Machine Learning Notebooks

Oracle Machine Learning Notebooks in Autonomous AI Lakehouse leverages Conda environment to install third-party packages in Oracle Machine Learning on Oracle Autonomous AI Database Serverless. The ADMIN user creates (installing any needed packages) and uploads the Conda environment to Object Storage. Once uploaded to Object Storage, these environments are available to all non-ADMIN users, who can load them directly into their Oracle Machine Learning Notebooks sessions. These conda environments work with Python and R, whether you are using OML4R (R, SQL, REST APIs) or OML4Py (Python, SQL, REST APIs).

For more information, see Install third-party packages

A.2.3.2 How to upload data from a .csv file into a Pandas DataFrame

How to upload data from a .csv file into a Pandas DataFrame

For example, how can I upload data from a `.csv` file into a Pandas DataFrame?

Resolution: To import the data from this `IRIS.csv` file present in Object Storage, and read into a Pandas DataFrame, run the following command:

```
%python

import pandas as pd
import ssl

url="https://objectstorage.us-ashburn-1.oraclecloud.com/n/adwc4pm/b/
OML_Data/o/IRIS.csv"

ssl._create_default_https_context = ssl._create_unverified_context
test = pd.read_csv(url)
test.head()
```

To fetch the same information for R, use `read.csv()` instead. Run the following command:

```
%r

url="https://objectstorage.us-ashburn-1.oraclecloud.com/n/adwc4pm/b/
OML_Data/o/IRIS.csv"

test = read.csv(url)
head(test)
```

A.2.3.3 "Allow Run" button to warn users about potential security threat

Issue: In Example template notebooks, if there is any javascript in the notebook codes or results, an **Allow Run** button will be displayed to warn users about potential threat for security reasons. If you don't want the warning to be displayed, then avoid using javascript in the (example) notebook.

Resolution: For example:

Replace `<a href="https://github.com/oracle/oracle-db-examples/tree/master/machine-learning" onclick="return ! window.open('https://github.com/oracle/`

```
oracle-db-examples/tree/master/machine-learning');">Oracle Machine Learning
GitHub folder</a>
```

with

```
<a href="https://github.com/oracle/oracle-db-examples/tree/master/machine-
learning" target="_blank">Oracle Machine Learning GitHub folder</a>
```

This also has the same behavior without javascript.

A.2.3.4 Resource Management in Oracle Machine Learning

Issue: Does Oracle Machine Learning impose any limitations on traffic or resource usage originating from a single source when triggering embedded Python execution, particularly for heavy workloads or resource-intensive tasks?

Resolution: For each embedded REST execution call, the Autonomous AI Database broker provisions an individual container from the VM based on the service levels - `Low`, `Medium`, `High` and releases it after completion. The resource allocated for the container is limited and the Autonomous AI Database load balancer controls the network traffic.

A.2.4 Conda

A.2.4.1 Error when loading libraries in Conda environments in Oracle Machine Learning Notebooks

Issue: When loading Conda libraries with statsmodels library and its dependencies, the following error occurs.

Error message:

```
Traceback (most recent call last): File "/tmp/python4753125085416874695/
zeppelin_python.py", line 145, in <module> import oml, numpy, pandas, scipy, matplotlib,
oracledb, sklearn File "/usr/local/lib/python3.12/site-packages/oml/__init__.py", line
43, in <module> from oml.core import * File "/usr/local/lib/python3.12/site-packages/oml/
core/__init__.py", line 33, in <module> from .methods import connect File "oml/core/
methods.py", line 79, in init oml.core.methods.....
```

Resolution:

1. Adjust the version of statsmodels that is being installed.
2. Use the conda `search` command to view the library version and the Python build. This includes the Python version.
3. Use `statsmodels==0.14.0` to address this issue.