Oracle® Cloud

Using Oracle Machine Learning User Interface on Autonomous Database

E78525-58
February 2024

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

E78525-58

Primary Author: Moitreyee Hazarika

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

Contributors: Charlie Berger

Contents

7 Get Started with Data Monitoring

8 Get Started with Model Monitoring

9 Get Started with Models

10 Use Templates to Collaborate with Users

11 Get Started with Jobs

12 Administer Oracle Machine Learning

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 in Oracle Machine Learning User Interface on Autonomous Database

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

Table 1-1 New Features

Features	Description
Support for model monitoring in Oracle Machine Learning User Interface	Oracle Machine Learning User Interface supports for model monitoring. It allows you to create model monitors using which you can 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 Autonomous 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

Table 1-1 (Cont.) New Features

Features	Description
	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.
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 Database - Serverless, including Autonomous Data Warehouse , Autonomous Transaction Processing and Oracle Autonomous 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.

Table 1-1 (Cont.) New Features

Features	Description
Oracle Machine Learning repository migrated from Serverless database to each respective Oracle Autonomous Database instance.	The Oracle Machine Learning (OML) repository has been migrated from Serverless database to each respective Oracle Autonomous 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 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 Database instance.
Oracle Machine Learning Notebook supported on all Oracle Autonomous Database clones	Oracle Machine Learning Notebook is supported on all types of Oracle Autonomous 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 with Oracle Machine Learning User Interface

This section discusses how to get started with Oracle Machine Learning User Interface, and use Apache Zeppelin-based machine learning notebooks along with the Notebooks EA (Early Adopter), 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 Database.
- About 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.
- 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.

Related Topics

- About AutoML UI
- Get Started with 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.
- Use Templates to Collaborate with Users
  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 Database. For more information, see Provision Autonomous Database

Related Topics

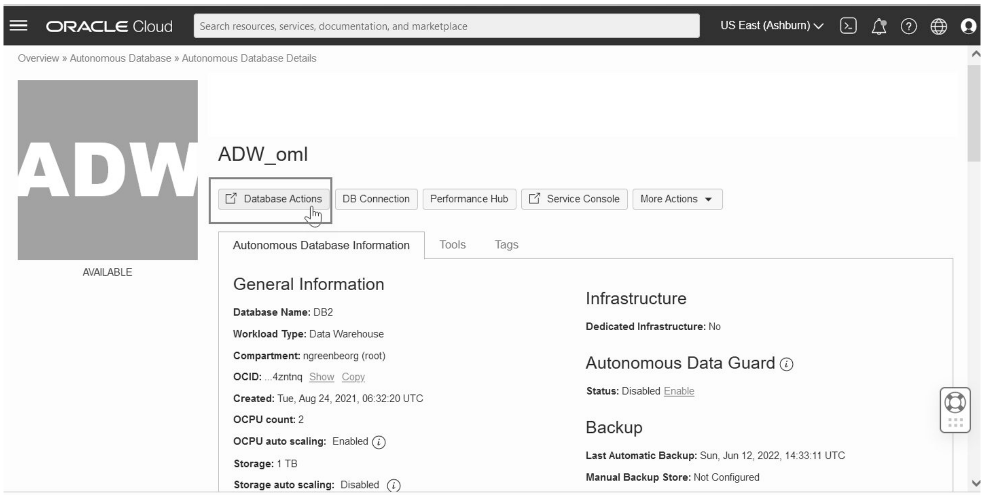
- OML Notebooks Interactive Tour
- About Oracle Machine Learning Password Policy
  All Oracle Machine Learning users must follow the password policy to create a strong and secure password.

2.2 Access Oracle Machine Learning User Interface

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

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

1. Select your Autonomous Database instance and on the Autonomous Database details page click **Database Actions**.


ORACLE Cloud
Search resources, services, documentation, and marketplace
US East (Ashburn)
Overview » Autonomous Database » Autonomous Database Details
ADW
AVAILABLE
ADW_oml
Database Actions
DB Connection
Performance Hub
Service Console
More Actions
Autonomous Database Information
Tools
Tags

General Information

Database Name: DB2
Workload Type: Data Warehouse
Compartment: ngreenbeorg (root)
OCID: ...4zntnq Show Copy
Created: Tue, Aug 24, 2021, 06:32:20 UTC
OCPU count: 2
OCPU auto scaling: Enabled
Storage: 1 TB
Storage auto scaling: Disabled

Infrastructure

Dedicated Infrastructure: No

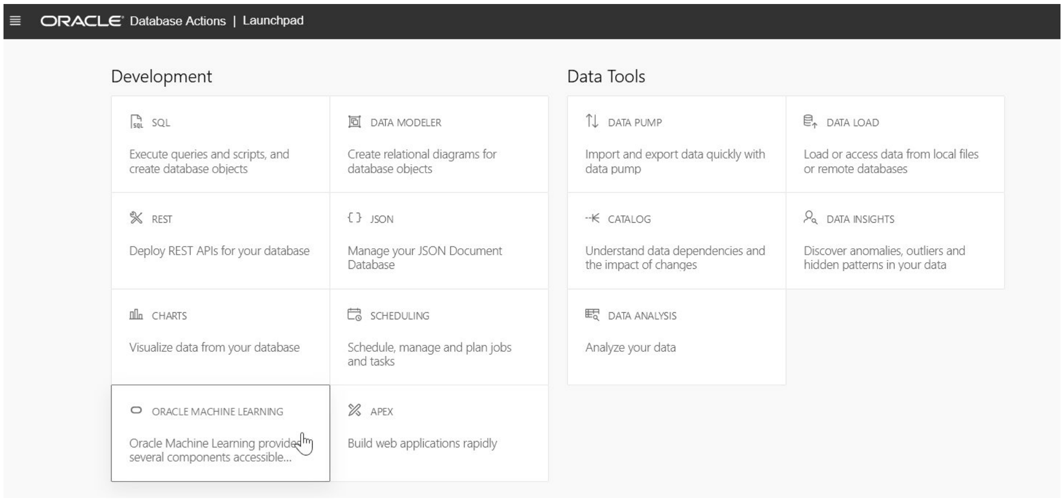
Autonomous Data Guard

Status: Disabled Enable

Backup

Last Automatic Backup: Sun, Jun 12, 2022, 14:33:11 UTC
Manual Backup Store: Not Configured

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



Development

SQL
Execute queries and scripts, and create database objects

DATA MODELER
Create relational diagrams for database objects

REST
Deploy REST APIs for your database

JSON
Manage your JSON Document Database

CHARTS
Visualize data from your database

SCHEDULING
Schedule, manage and plan jobs and tasks

ORACLE MACHINE LEARNING
Oracle Machine Learning provides several components accessible...

APEX
Build web applications rapidly

Data Tools

DATA PUMP
Import and export data quickly with data pump

DATA LOAD
Load or access data from local files or remote databases

CATALOG
Understand data dependencies and the impact of changes

DATA INSIGHTS
Discover anomalies, outliers and hidden patterns in your data

DATA ANALYSIS
Analyze your data

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 About 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 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.

2.3.1 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
Create notebooks	Create a Notebook
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	Call 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

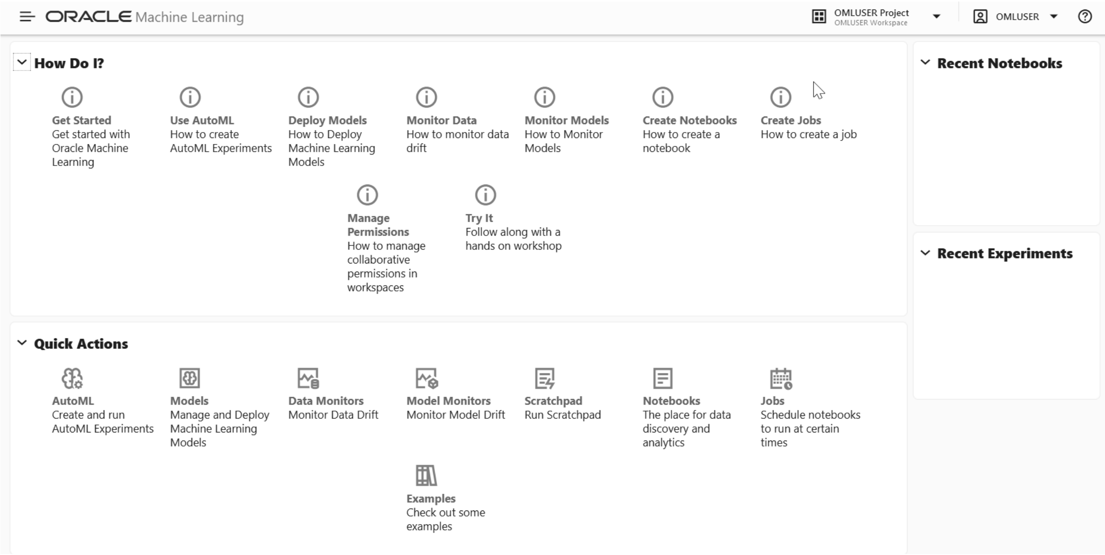
Related Topics

- 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.

2.4 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 2-1 Oracle Machine Learning UI Home Page



ORACLE Machine Learning

OMLUSER Project
OMLUSER Workspace

OMLUSER

How Do I?

Get Started
Get started with Oracle Machine Learning

Use AutoML
How to create AutoML Experiments

Deploy Models
How to Deploy Machine Learning Models

Monitor Data
How to monitor data drift

Monitor Models
How to Monitor Models

Create Notebooks
How to create a notebook

Create Jobs
How to create a job

Manage Permissions
How to manage collaborative permissions in workspaces

Try It
Follow along with a hands on workshop

Quick Actions

AutoML
Create and run AutoML Experiments

Models
Manage and Deploy Machine Learning Models

Data Monitors
Monitor Data Drift

Model Monitors
Monitor Model Drift

Scratchpad
Run Scratchpad

Notebooks
The place for data discovery and analytics

Jobs
Schedule notebooks to run at certain times

Examples
Check out some examples

Recent Notebooks

Recent Experiments

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**
  - **Models**
  - **Data Monitors**
  - **Model Monitors**
  - **Scratchpad**
  - **Notebooks**
  - **Jobs**
  - **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.

Related Topics

- Notebooks
  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.
- 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.

3

Get Started with 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.

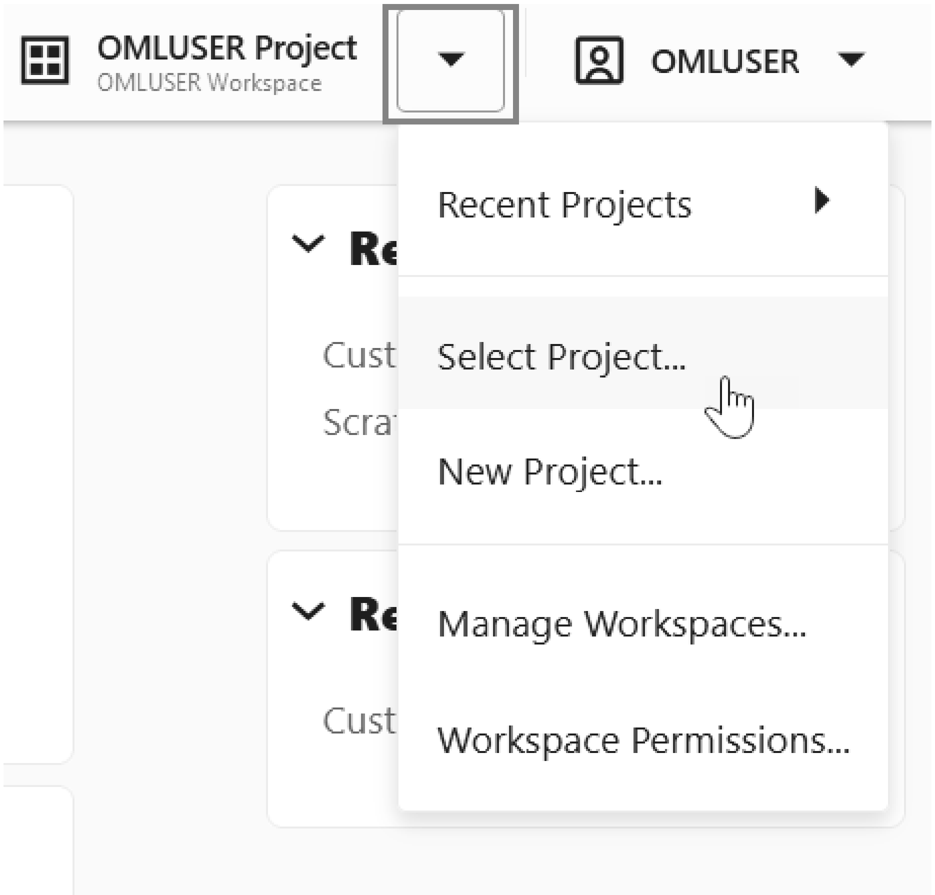
The initial workspace and the default project is created for you when you log in to Oracle Machine Learning User Interface (UI) for the first time. To create a new project:

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.

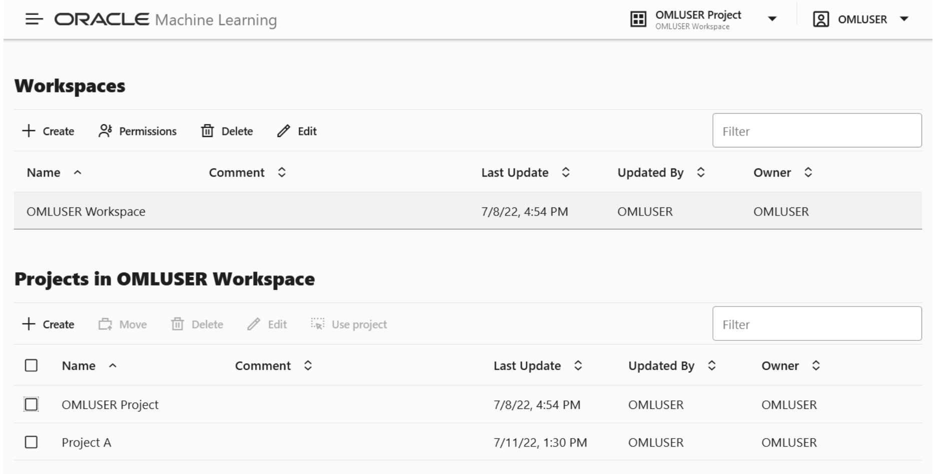
   > **Note:**
   >
   > The last project that you have worked on is stored in the browser cache and is the default project. If you clear the cache, then no default exists and you must select a project.

   You can create projects in two ways:

- Use the **New Project** option: Click the down arrow next to the **Project** field and then click **New Project.** The Create Project dialog opens.



- 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.



Workspaces

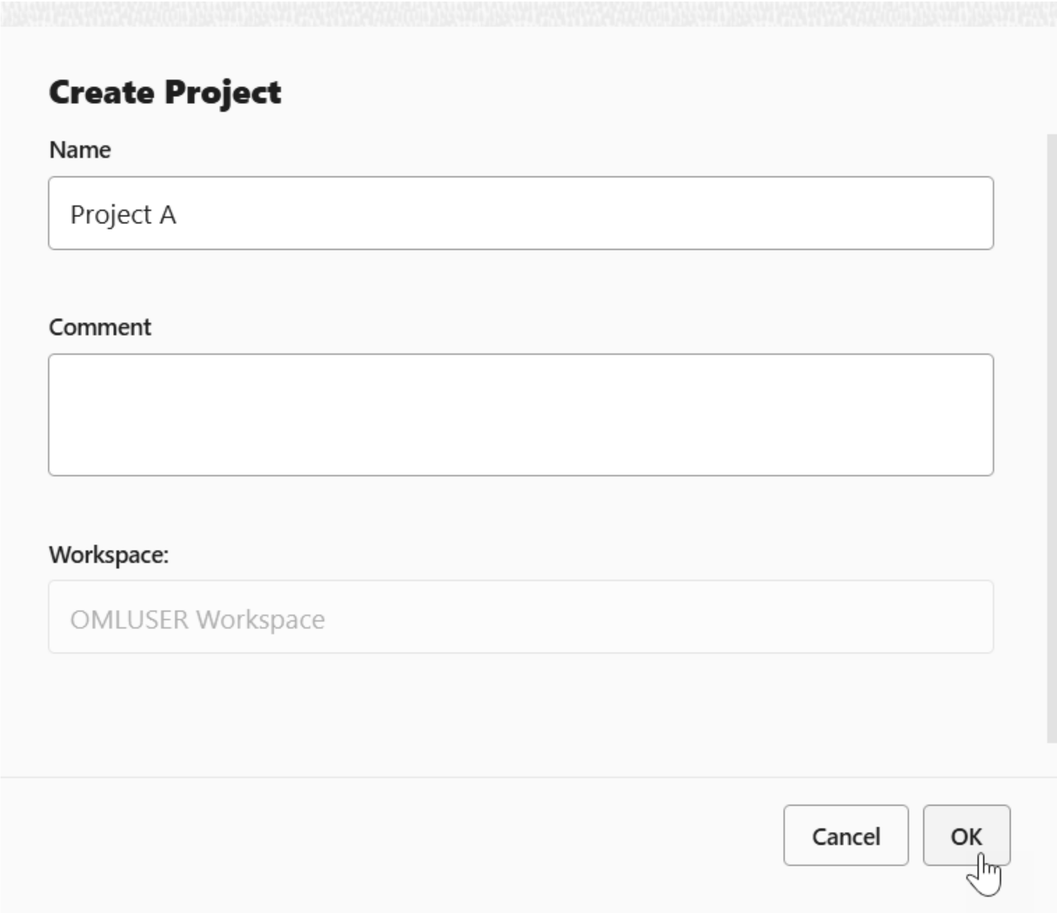
Name	Comment	Last Update	Updated By	Owner
OMLUSER Workspace		7/8/22, 4:54 PM	OMLUSER	OMLUSER

Projects in OMLUSER Workspace

☐	Name	Comment	Last Update	Updated By	Owner
☐	OMLUSER Project		7/8/22, 4:54 PM	OMLUSER	OMLUSER
☐	Project A		7/11/22, 1:30 PM	OMLUSER	OMLUSER

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

Figure 3-1 Create Project dialog



Create Project

Name: Project A

Comment:

Workspace: OMLUSER Workspace

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

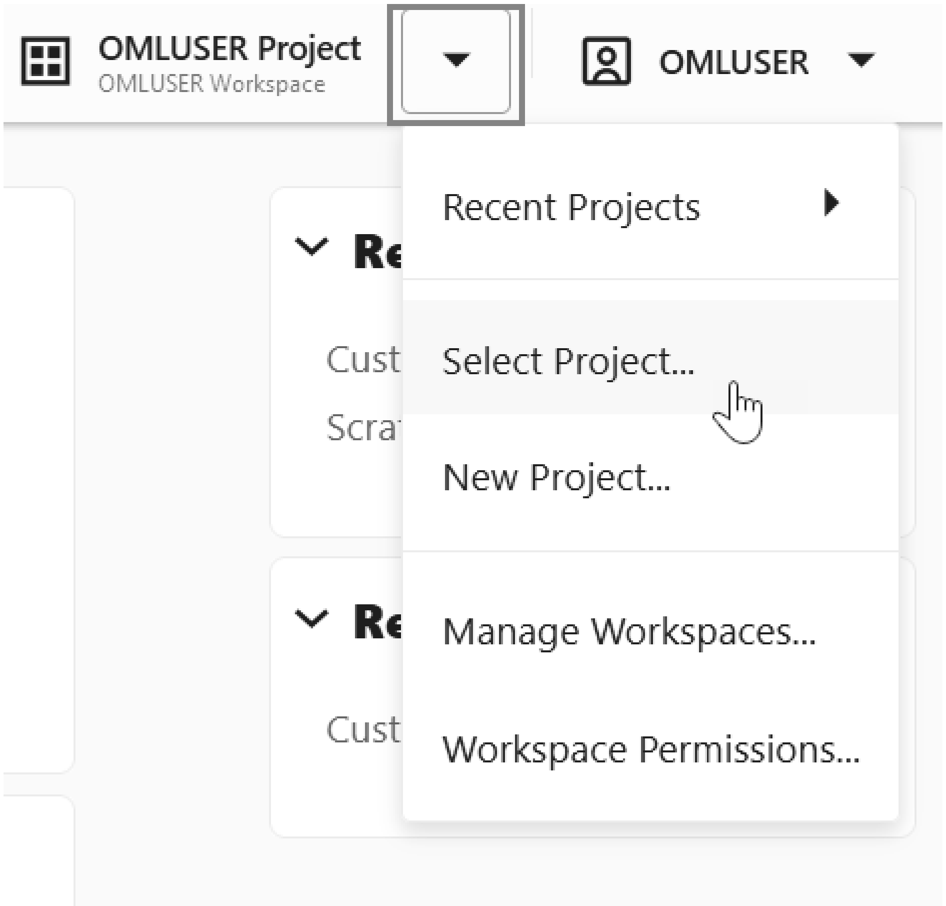
You can create and manage new projects and workspaces, provide access to your workspaces, manage permissions for users, and edit and delete workspaces.

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

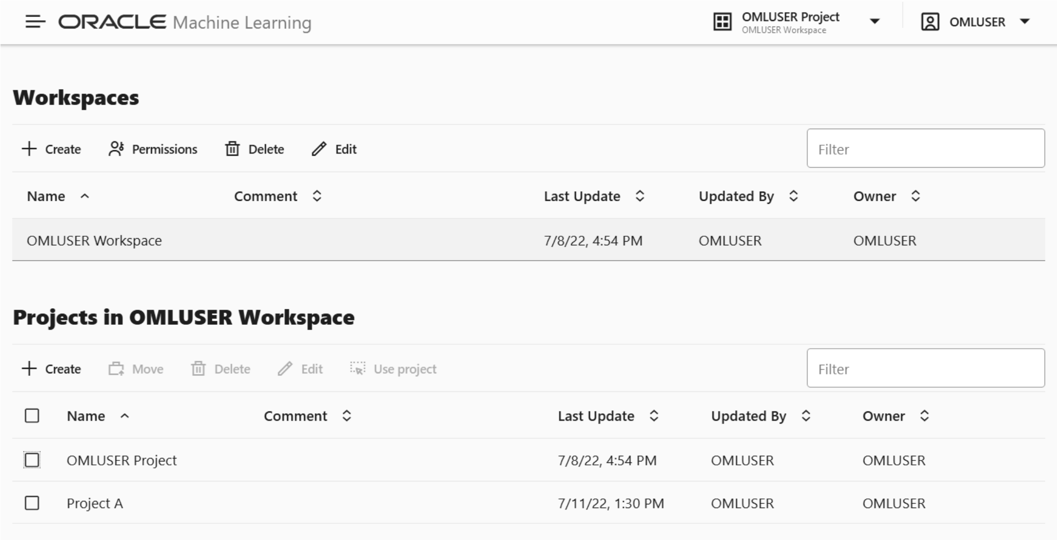
Create and Manage Workspace

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:



Workspaces

Name	Comment	Last Update	Updated By	Owner
OMLUSER Workspace		7/8/22, 4:54 PM	OMLUSER	OMLUSER

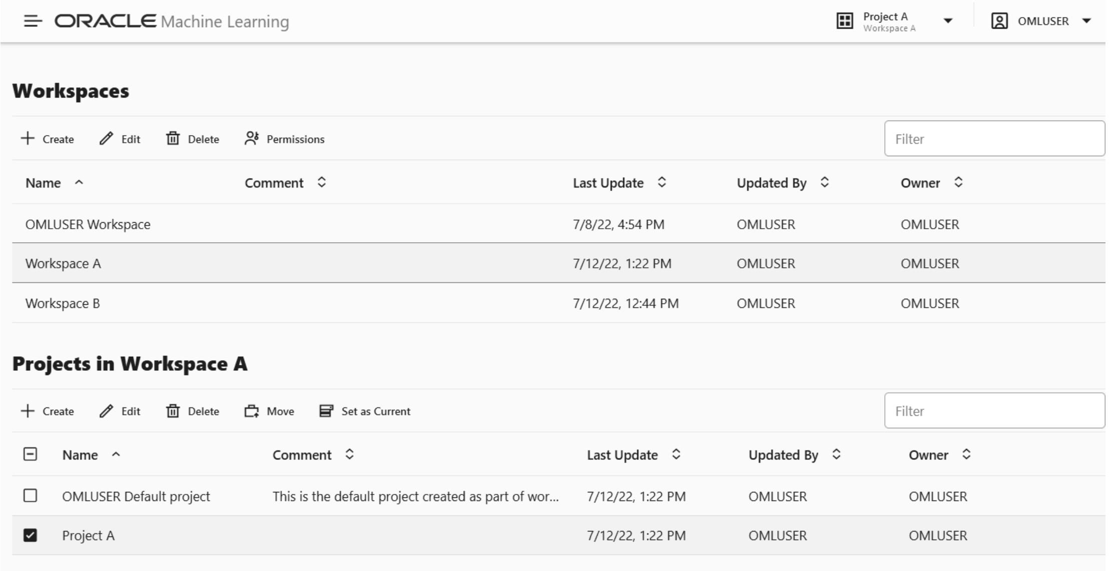
Projects in OMLUSER Workspace

☐	Name	Comment	Last Update	Updated By	Owner
☐	OMLUSER Project		7/8/22, 4:54 PM	OMLUSER	OMLUSER
☐	Project A		7/11/22, 1:30 PM	OMLUSER	OMLUSER

- 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.



Workspaces

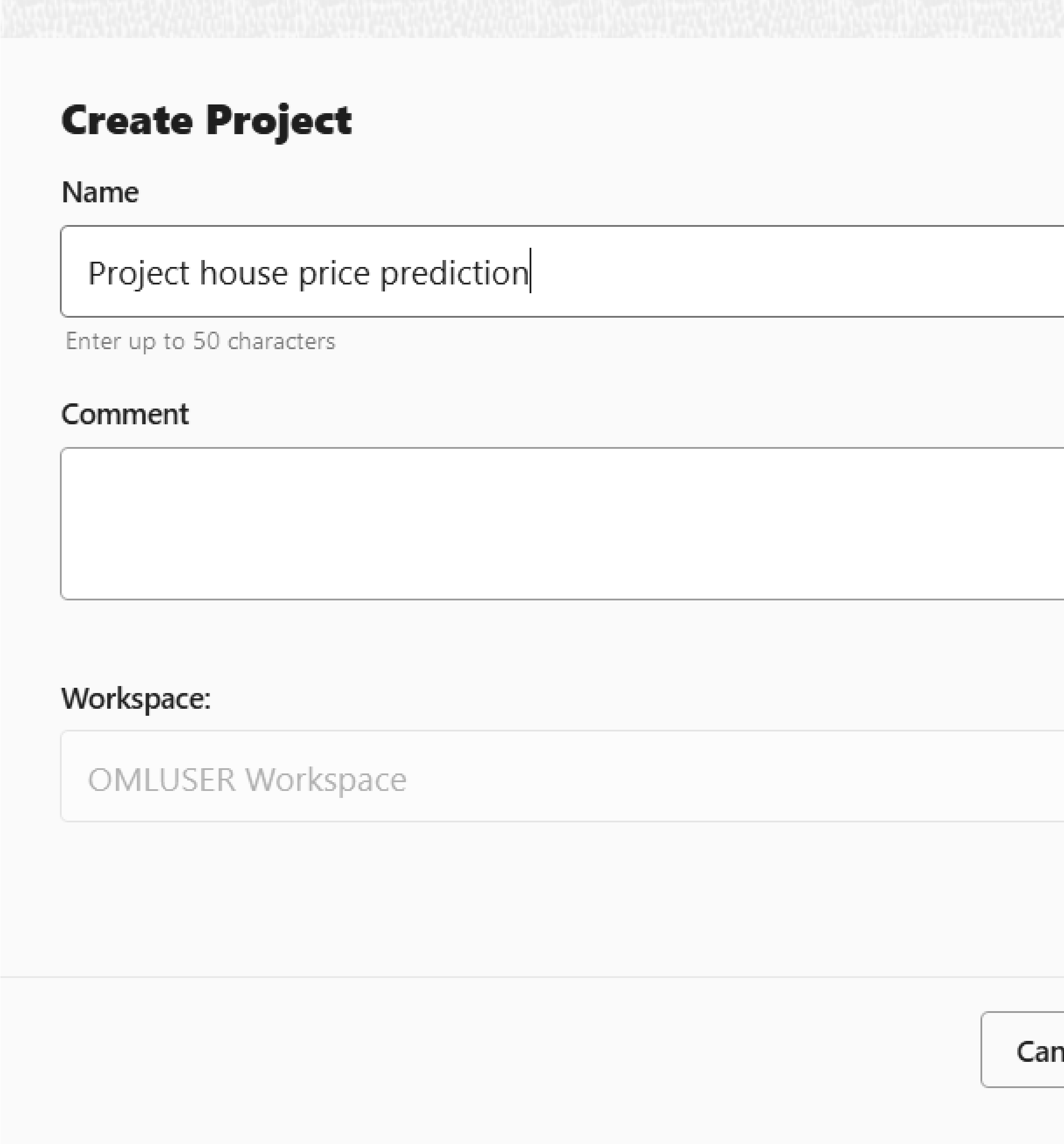
Name	Comment	Last Update	Updated By	Owner
OMLUSER Workspace		7/8/22, 4:54 PM	OMLUSER	OMLUSER
Workspace A		7/12/22, 1:22 PM	OMLUSER	OMLUSER
Workspace B		7/12/22, 12:44 PM	OMLUSER	OMLUSER

Projects in Workspace A

⊟	Name	Comment	Last Update	Updated By	Owner
☐	OMLUSER Default project	This is the default project created as part of wor...	7/12/22, 1:22 PM	OMLUSER	OMLUSER
☑	Project A		7/12/22, 1:22 PM	OMLUSER	OMLUSER

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.



Create Project

Name

Project house price prediction

Enter up to 50 characters

Comment

Workspace:

OMLUSER Workspace

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.

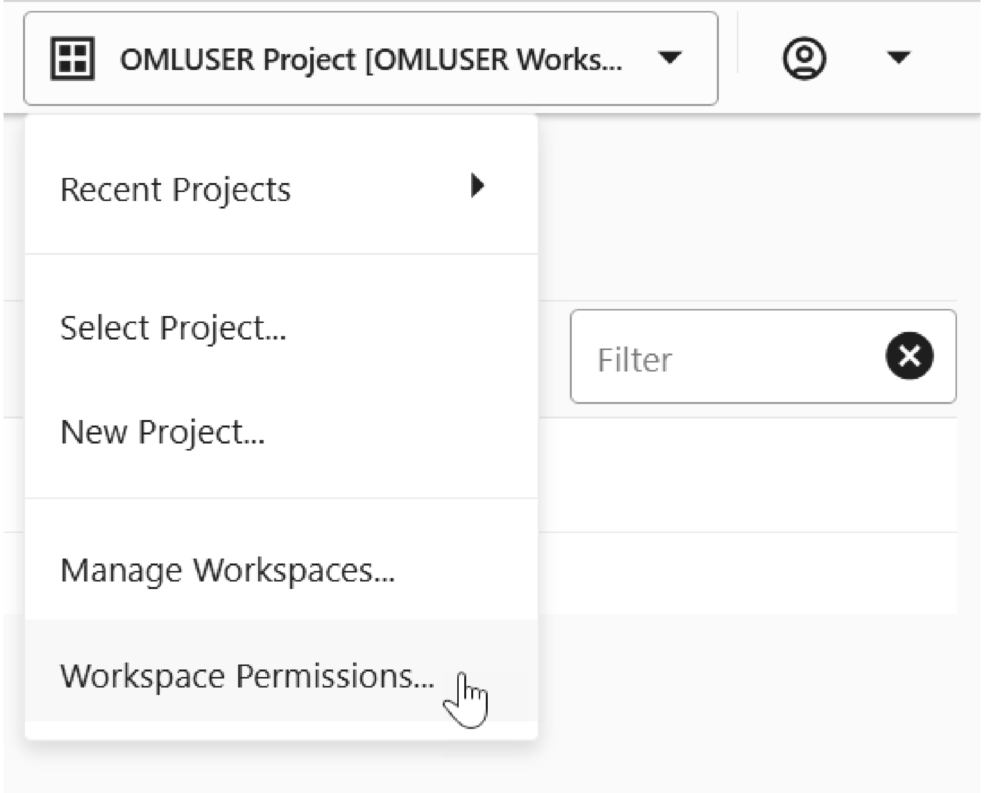
> **Caution:**
>
> If you grant the permission type **Manager** or **Developer,** the user can also drop tables, create tables, and run any scripts at any time on your account. The user with **Viewer** permission type can only view your notebooks and is not authorized to run or make any changes to your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

To grant permission to another user:

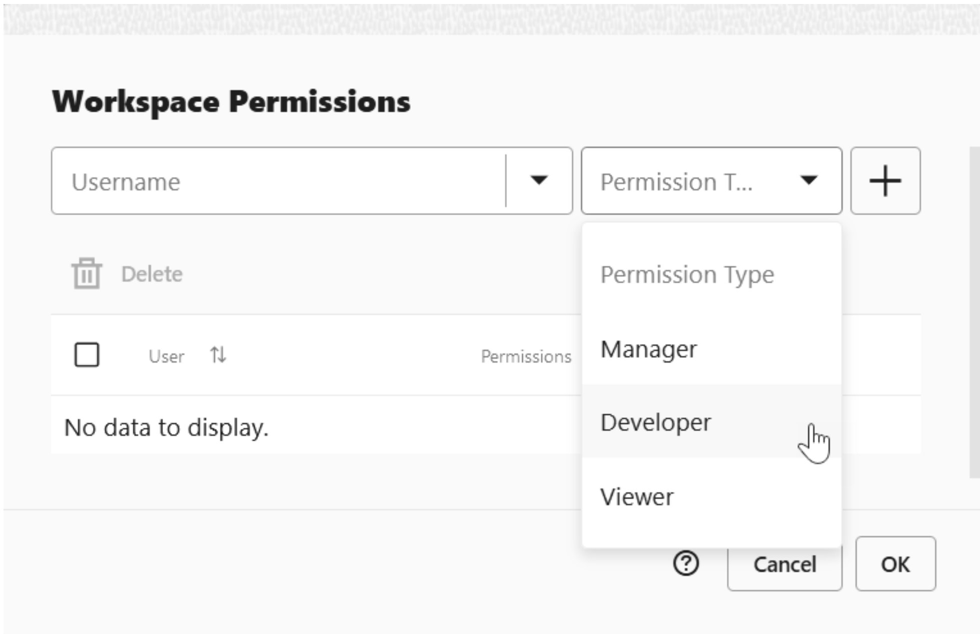
1. On the top right corner of Oracle Machine Learning UI home page, click the project and workspace drop-down list and select **Workspace Permission.**

   The Permissions dialog box opens.

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



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



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.

Permission Types	Actions based on permission
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 Get Started with Notebooks Early Adopter for Data Analysis and Data Visualization

Oracle Machine Learning Notebooks Early Adopter 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.

- About Notebooks Early Adopter
  Oracle Machine Learning Notebooks Early Adopter 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.
- Access your Notebooks Early Adopter Page
  You can access the Notebooks Early Adopter page from the left navigation pane of Oracle Machine Learning Notebooks, or from the Notebooks page.
- Edit your Oracle Machine Learning Notebook Early Adopter
  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`.

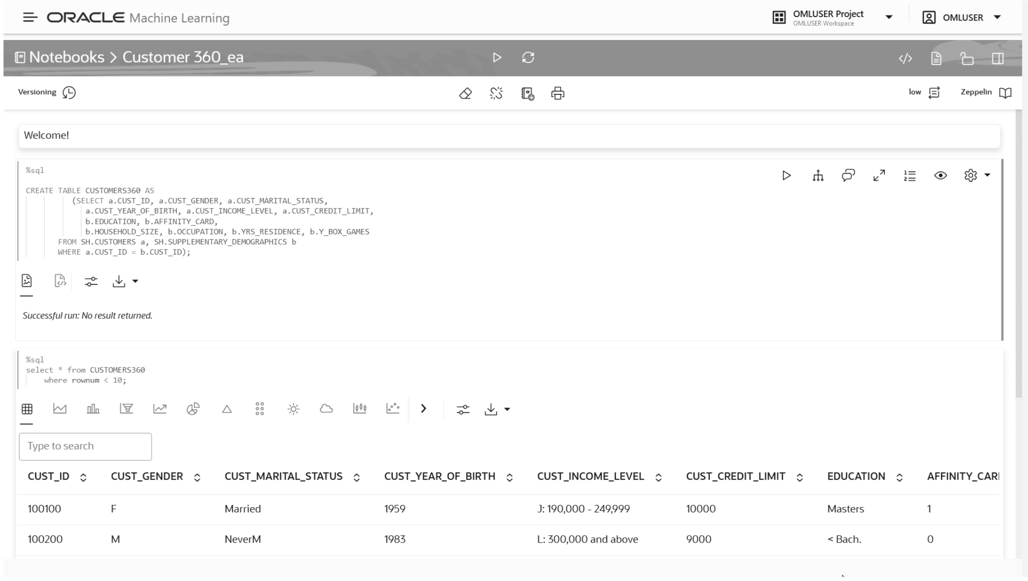
4.1 About Notebooks Early Adopter

Oracle Machine Learning Notebooks Early Adopter 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 early adopter 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 `_ea` appended to it.



Welcome!

```
%sql

CREATE TABLE CUSTOMERS360 AS
          (SELECT a.CUST_ID, a.CUST_GENDER, a.CUST_MARITAL_STATUS,
             a.CUST_YEAR_OF_BIRTH, a.CUST_INCOME_LEVEL, a.CUST_CREDIT_LIMIT,
             b.EDUCATION, b.AFFINITY_CARD,
             b.HOUSEHOLD_SIZE, b.OCCUPATION, b.YRS_RESIDENCE, b.Y_BOX_GAMES
       FROM SH.CUSTOMERS a, SH.SUPPLEMENTARY_DEMOGRAPHICS b
       WHERE a.CUST_ID = b.CUST_ID);
```

Successful run: No result returned.

```
%sql
select * from CUSTOMERS360
    where rownum < 10;
```

Type to search

CUST_ID	CUST_GENDER	CUST_MARITAL_STATUS	CUST_YEAR_OF_BIRTH	CUST_INCOME_LEVEL	CUST_CREDIT_LIMIT	EDUCATION	AFFINITY_CARI
100100	F	Married	1959	J: 190,000 - 249,999	10000	Masters	1
100200	M	NeverM	1983	L: 300,000 and above	9000	< Bach.	0

The Oracle Machine Learning Notebook Early Adopter provides:

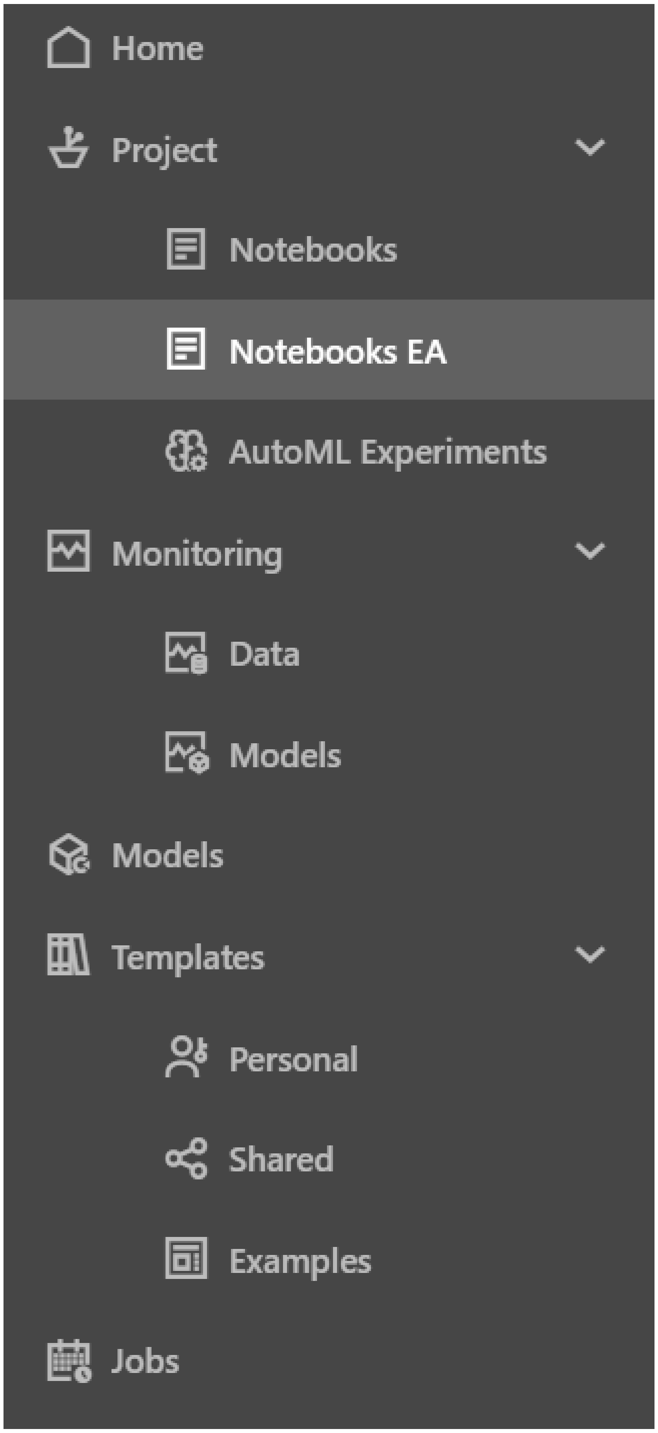
- Faster notebook loading time.
- The Oracle look and feel as it based on the Oracle Redwood theme.
- Enriched visualization in its Line chart, Area chart, Bar chart, Pyramid chart, Pie chart, Donut chart, Funnel chart, Tag Cloud, Treemap Diagram, Sunburst Diagram, Scatter Plot, Box Plot.
- Option to 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 through drop-down menu.
- Layout of Zeppelin and Jupyter notebook.
- On-page versioning, viewing of version history, and version comparison.

4.2 Access your Notebooks Early Adopter Page

You can access the Notebooks Early Adopter page from the left navigation pane of Oracle Machine Learning Notebooks, or from the Notebooks page.

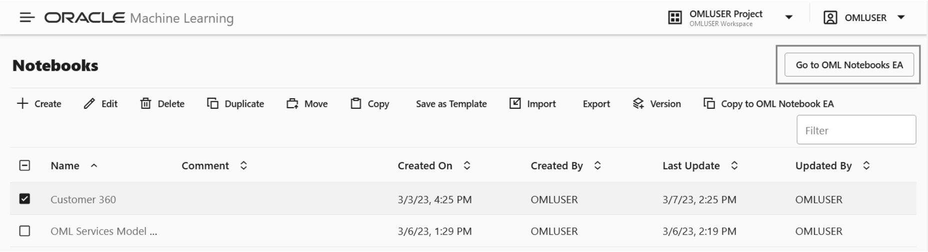
1. To access the Notebooks Early Adopter page:
   - Go to the Oracle Machine Learning left navigation pane, expand Projects and then click **Notebooks EA**.

Figure 4-1 Left navigation pane



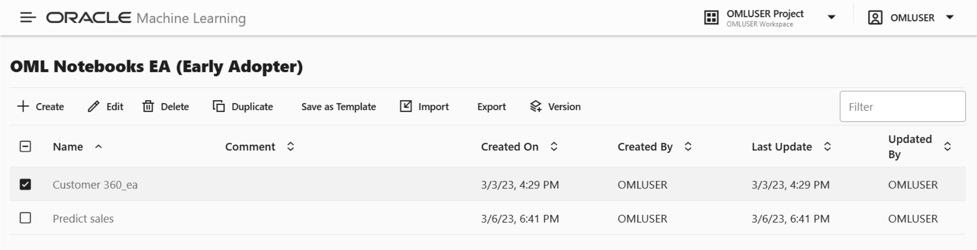
- Alternatively, you can click **Notebooks** under quick link on the home page to open the Notebooks page. Here, click **Go to OML Notebooks EA** to open the Notebooks

Early Adopter page.



	Name	Comment	Created On	Created By	Last Update	Updated By
☑	Customer 360		3/3/23, 4:25 PM	OMLUSER	3/7/23, 2:25 PM	OMLUSER
☐	OML Services Model ...		3/6/23, 1:29 PM	OMLUSER	3/6/23, 2:19 PM	OMLUSER

2. This opens the OML Notebooks EA (Early Adopter) page.



	Name	Comment	Created On	Created By	Last Update	Updated By
☑	Customer 360_ea		3/3/23, 4:29 PM	OMLUSER	3/3/23, 4:29 PM	OMLUSER
☐	Predict sales		3/6/23, 6:41 PM	OMLUSER	3/6/23, 6:41 PM	OMLUSER

Here, you have the option to:

- **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.
- **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.
- **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 as .json files, click **Import**. Select the project and workspace in which to import the notebook.
- **Export:** To export a notebook, click **Export.** You can export Notebooks Early Adopter 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 EA (Early Adopter) page.

> **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.

- Work with Notebook Versions on the Notebooks Page
  By creating versions of your notebook, you can archive your work in a notebook.

4.2.1 Work with Notebook Versions on the Notebooks Page

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

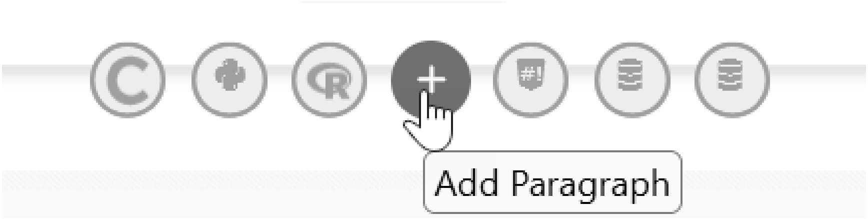
You can create versions of notebooks on the Notebooks page, as well as in the Notebook editor. In this example, the **Notebook Versioning Demo** notebook is created and versioned as **Version 1**.

> **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 EA 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`.

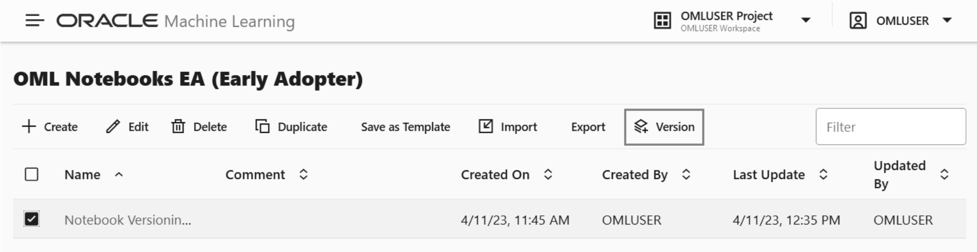
```
%script
SELECT count(*) from ESM_SH_DATA;
```

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 EA page after all the paragraphs run successfully.

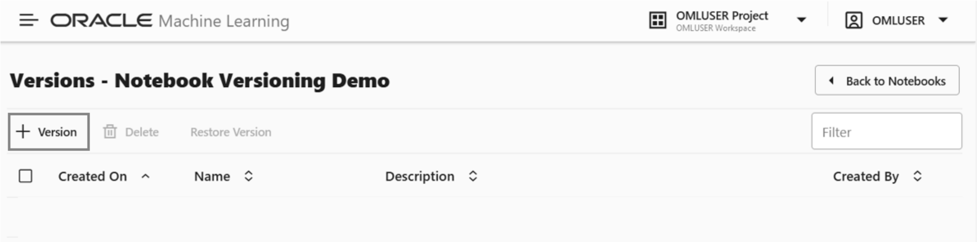
5. On the Notebooks EA page, select the **Notebook Versioning Demo** notebook to enable all the edit options, and click **Versions** to go to the versions page for this notebook.



☐	Name	Comment	Created On	Created By	Last Update	Updated By
☑	Notebook Versionin...		4/11/23, 11:45 AM	OMLUSER	4/11/23, 12:35 PM	OMLUSER

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.



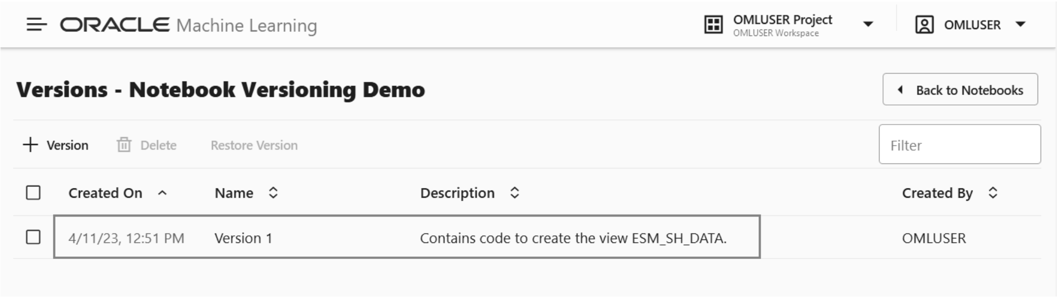
☐	Created On	Name	Description	Created By

7. In the Create Versions dialog:

a. **Name:** Enter `Version 1` for the new version of this notebook

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

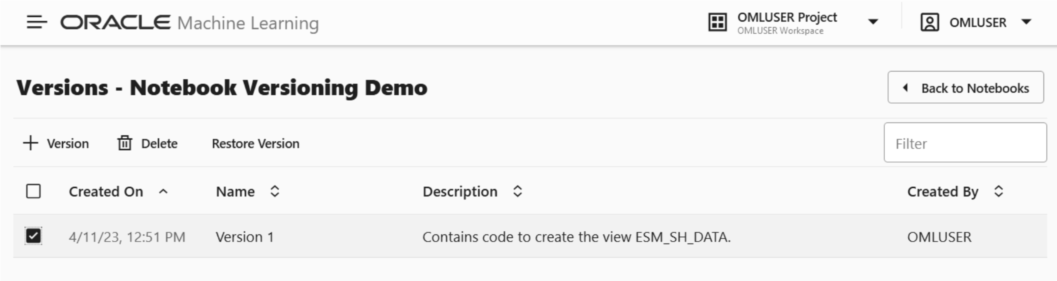
c. Click **OK.** Once the notebook version is created, it is listed on the *Versions - Notebook Versioning Demo* page.



Versions - Notebook Versioning Demo

☐	Created On	Name	Description	Created By
☐	4/11/23, 12:51 PM	Version 1	Contains code to create the view ESM_SH_DATA.	OMLUSER

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.



Versions - Notebook Versioning Demo

☐	Created On	Name	Description	Created By
☑	4/11/23, 12:51 PM	Version 1	Contains code to create the view ESM_SH_DATA.	OMLUSER

- 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.

4.3 Edit your Oracle Machine Learning Notebook Early Adopter

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:

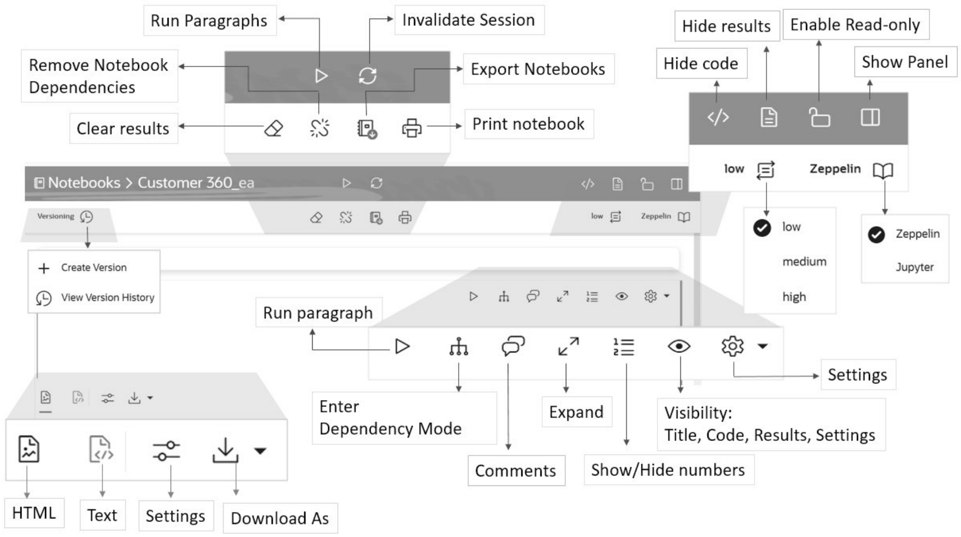
1. On the Oracle Machine Learning home page, select the project in which your notebook is available.

> **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 Early Adopter 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 Version 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 (Early Adopter notebook), `.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 Early Adopter.

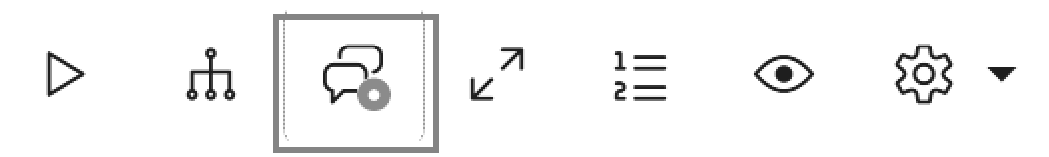
- Click ◫ to show the edit panel. The edit options in the panel are the same edit options available for the paragraph. Clicking the panel icon opens the edit pane on the right, and the edit tool bar in the paragraph is hidden.
- Click ⇄ to change the interpreter binding to either low, medium, or high
- Click 📖 to switch the Early Adopter 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.

4.3.1 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.

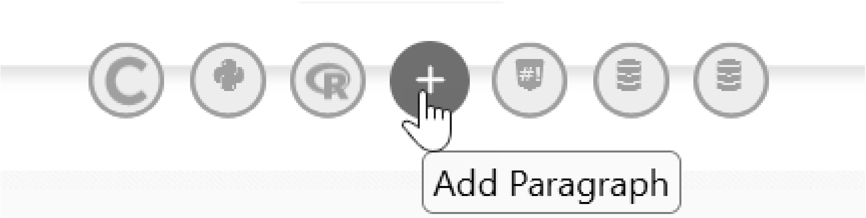
> **Note:**
>
> A versioned notebook is non-editable. If you want to make any changes to a particular version of a notebook, you must restore that version to edit it.

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.

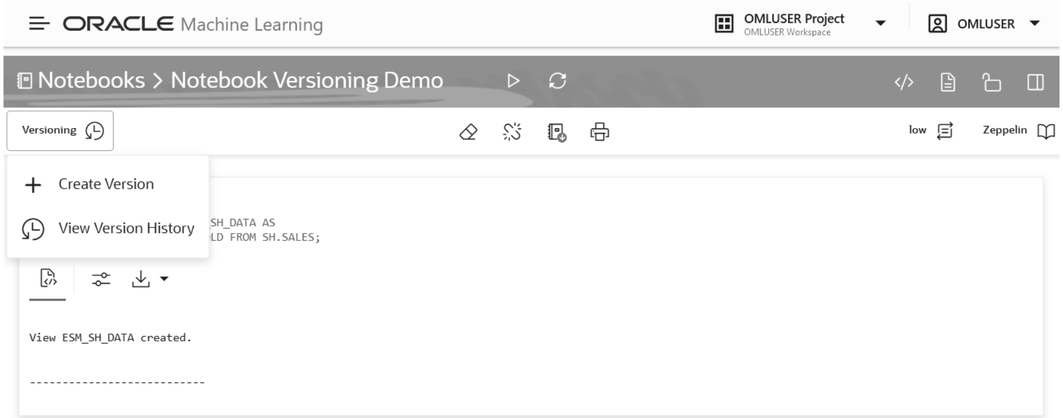
   

3. Copy and paste the following script to the new paragraph. This script builds a machine learning model using the ESM algorithm.

```
%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;
```

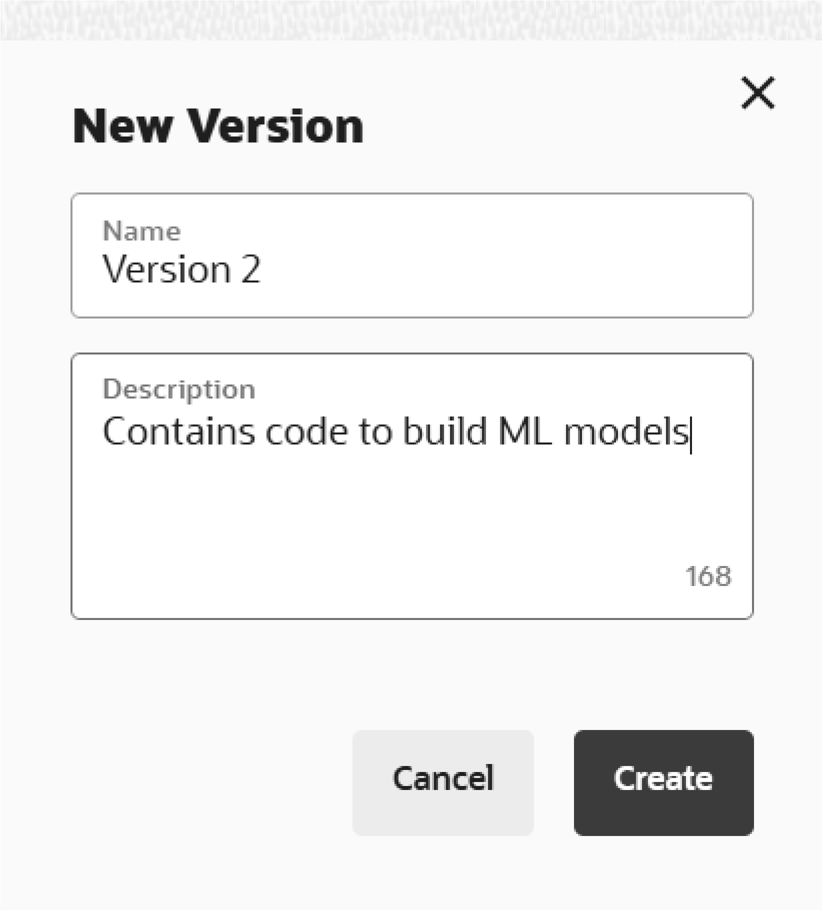
4. Now, archive this notebook along with the code to build the machine learning model by versioning it. On the top left corner of the notebook editor, click **Versioning**
5. The options to **Create Version** and **View Version History** opens.



```
SH_DATA AS
LD FROM SH.SALES;
```

```
View ESM_SH_DATA created.

---------------------------
```

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.

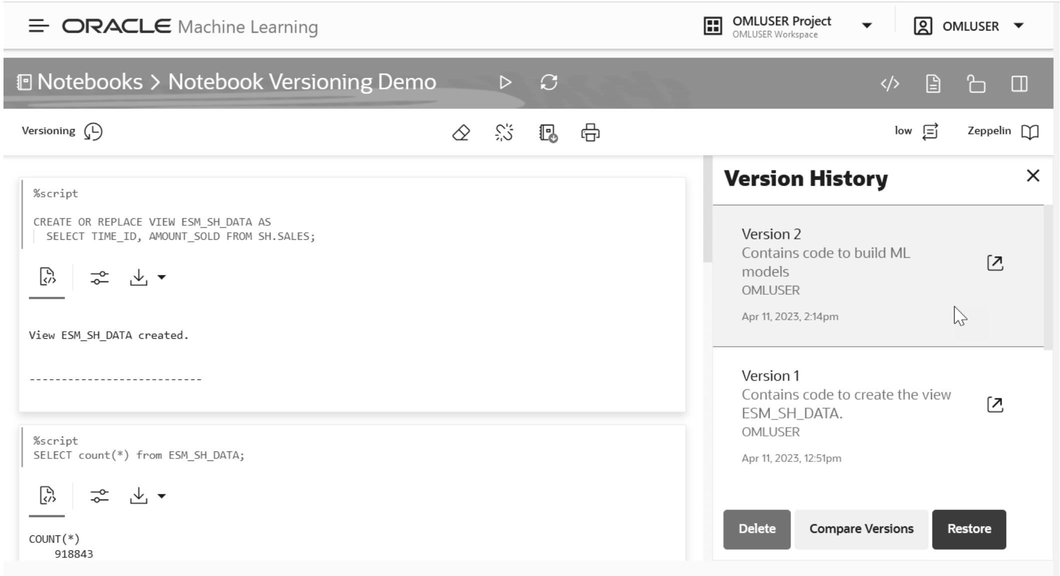


Name
Version 2

Description
Contains code to build ML models
168

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** panel 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 panel:



```
%script

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

```
View ESM_SH_DATA created.

---------------------------
```

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

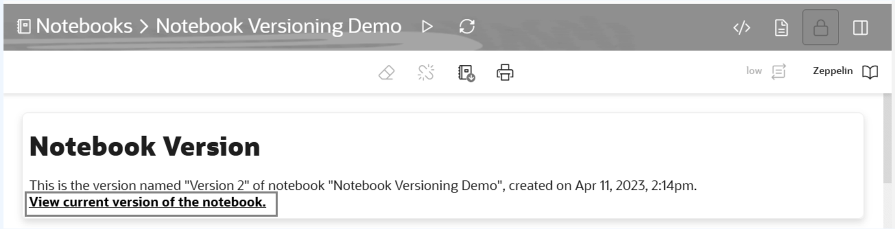
```
COUNT(*)
    918843
```

Version History

Version 2
Contains code to build ML models
OMLUSER
Apr 11, 2023, 2:14pm

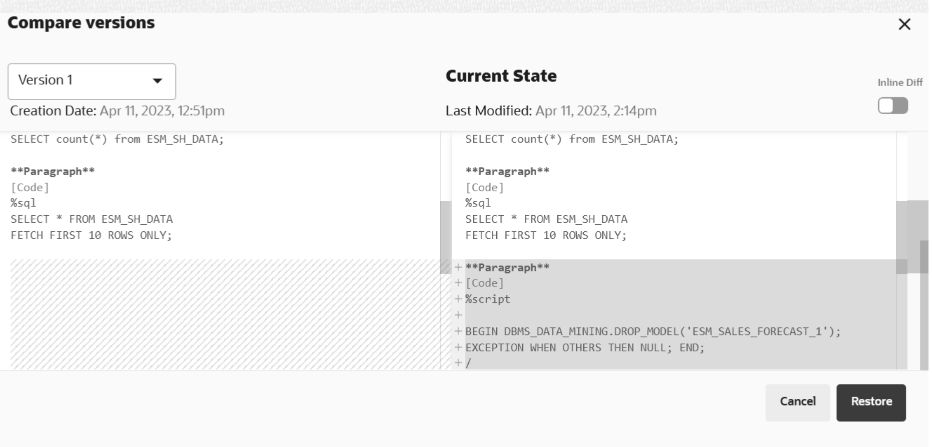
Version 1
Contains code to create the view ESM_SH_DATA.
OMLUSER
Apr 11, 2023, 12:51pm

- 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**.
>
> 
>
>
> **Notebook Version**
>
> This is the version named "Version 2" of notebook "Notebook Versioning Demo", created on Apr 11, 2023, 2:14pm.
> 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.



Current State

Creation Date: Apr 11, 2023, 12:51pm

Last Modified: Apr 11, 2023, 2:14pm

```
SELECT count(*) from ESM_SH_DATA;

**Paragraph**
[Code]
%sql
SELECT * FROM ESM_SH_DATA
FETCH FIRST 10 ROWS ONLY;
```

```
SELECT count(*) from ESM_SH_DATA;

**Paragraph**
[Code]
%sql
SELECT * FROM ESM_SH_DATA
FETCH FIRST 10 ROWS ONLY;

+**Paragraph**
+[Code]
+%script
+
+BEGIN DBMS_DATA_MINING.DROP_MODEL('ESM_SALES_FORECAST_1');
+EXCEPTION WHEN OTHERS THEN NULL; END;
+/
```

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

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

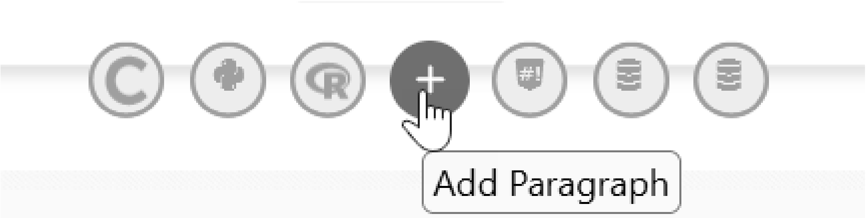
4.3.2 Create Paragraph Dependencies

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 EA 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;
```

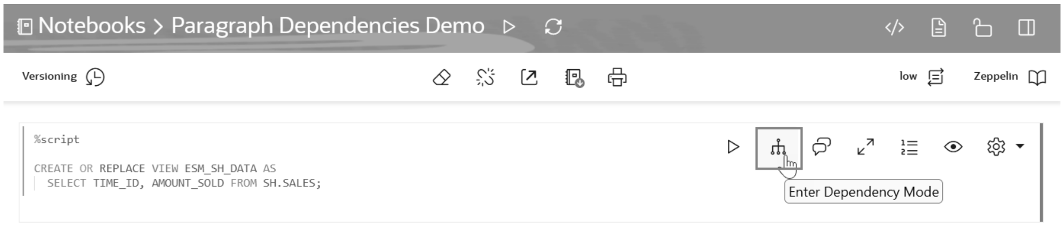
b. In the second paragraph, copy and paste the following SQL script. This script gives a count of the record present in the view `ESM_SH_DATA`.

```
%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.

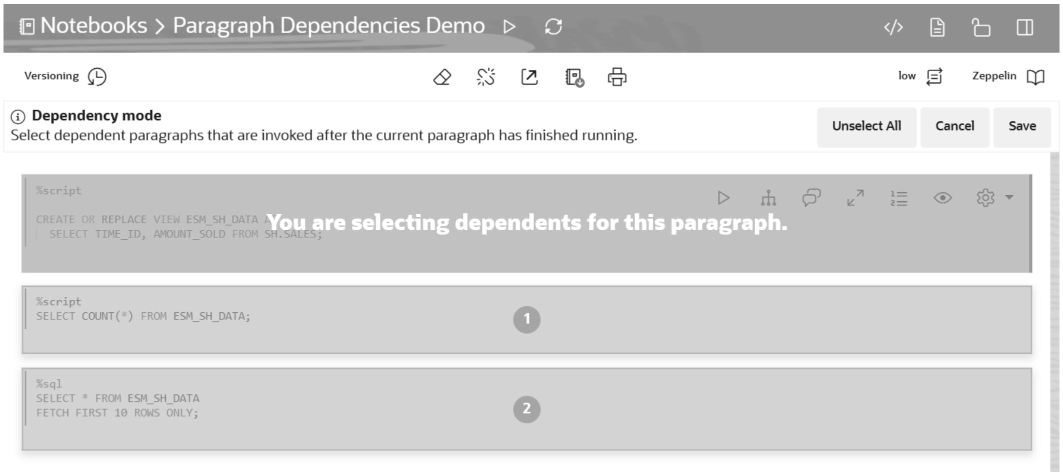


```
%script

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

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.



Dependency mode
Select dependent paragraphs that are invoked after the current paragraph has finished running.

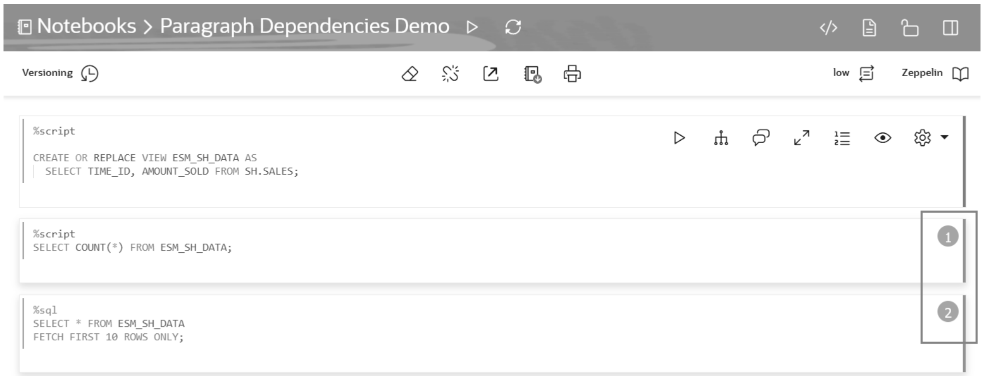
```
%script

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

You are selecting dependents for this paragraph.

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

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

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

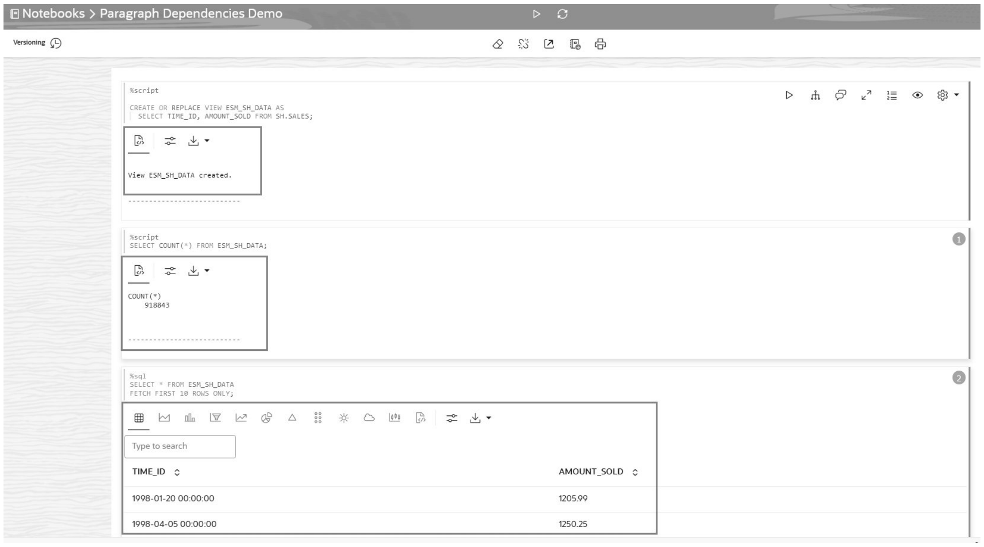


```
%script

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

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

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

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.



```
%script

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

```
View ESM_SH_DATA created.

---------------------------
```

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

```
COUNT(*)
    918843

---------------------------
```

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

TIME_ID	AMOUNT_SOLD
1998-01-20 00:00:00	1205.99
1998-04-05 00:00:00	1250.25

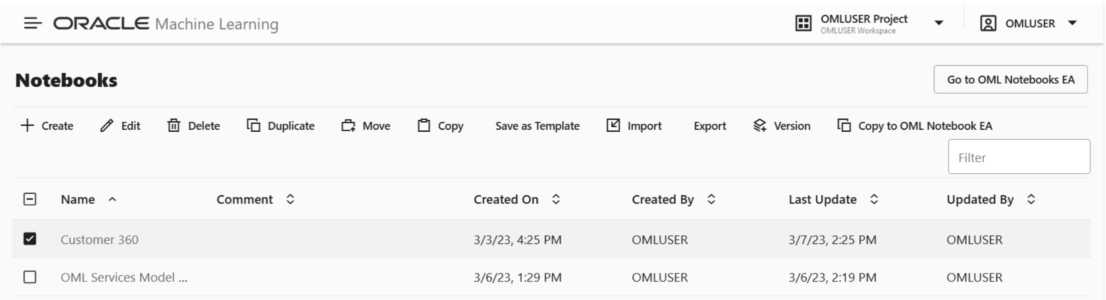
5
About Oracle Machine Learning Notebooks

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.

The Notebooks page lists all the notebooks associated with the selected project. You can create, edit, and run your notebooks here.

You can perform the following tasks in the Notebooks page:



Notebooks

☐	Name	Comment	Created On	Created By	Last Update	Updated By
☑	Customer 360		3/3/23, 4:25 PM	OMLUSER	3/7/23, 2:25 PM	OMLUSER
☐	OML Services Model ...		3/6/23, 1:29 PM	OMLUSER	3/6/23, 2:19 PM	OMLUSER

- **Go to OML Notebook EA:** Click **Go to OML Notebook EA** to go to the Oracle Machine Learning Notebooks Early Adopter page.

> **Note:**
>
> EA stands for Early Adopter.

- **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.
- **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:** Select a notebook and click **Move** to move the notebook to a different project, either in the same or in a different workspace. You must have either the `Administrator` or `Developer` privilege to move notebooks across projects and workspaces. You can move notebooks that are running from jobs because when you run notebooks from jobs, it runs a copy of the notebook.

> **Note:**
>
> A notebook cannot be moved if it is in RUNNING state, or if a notebook already exists in the target project by the same name.

- **Copy:** Select a notebook and click **Copy** to copy the notebook to another project, either in the same or different workspace. You must have either the `Administrator` or the `Developer` privilege to copy notebooks in different projects and workspaces.
- **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 as `.json` files, click **Import.** Select the project and workspace in which to import the notebook.
- **Export:** To export a notebook, click **Export.** You can export notebooks in 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.** You can experiment with your notebook by creating versions of it, and revert to an older version by clicking **Revert Version.**

> **Note:**
>
> You can also version a notebook by opening it, and then clicking on the Versioning option 🕓. 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.

- **Copy to OML Notebook EA:** To copy the selected notebook(s) to the early adopter interface using the original name with the suffix _ea.`_ea`. You can now access this notebook by clicking on **Notebooks EA** in the left navigation menu, or by clicking on **Go to OML Notebooks EA** on the top right corner of this page.
- To open a notebook and run it, click the notebook. Notebook opens in the edit mode.
- Create a Notebook
  A notebook is a web-based interface for data analysis, data discovery, data visualization and collaboration.
- Edit Your 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`.
- Version a Notebook
  You can version or create a backup a notebook, experiment on it, and revert to the original notebook, if required.

- Save Notebooks as Templates
  You can save an existing notebook as a template in Personal or in Shared.
- 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.
- Call the Markdown Interpreter and Generate Static html from Markdown Plain Text
  Use the Markdown interpreter and generate static html from Markdown plain text.
- About Interpreter Bindings and Notebooks
  An interpreter is a plug-in that allows you to use a specific data processing language backend.
- 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.
- Collaborate in Oracle Machine Learning
  Two or more users can collaborate and share Oracle Machine Learning User Interface (UI) notebooks with other users.

Related Topics

- About Interpreter Bindings and Notebooks
  An interpreter is a plug-in that allows you to use a specific data processing language backend.

5.1 Create a Notebook

A notebook is a web-based interface for data analysis, data discovery, data visualization and collaboration.

Whenever you create a notebook, it has an interpreter settings specification. The notebook contains an internal list of bindings that determines the order of the interpreter bindings. A notebook comprises paragraphs which is a notebook component where you can write SQL statements, run PL/SQL scripts, and run Python 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 executed. In the output section, the results of the interpreter are provided.

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.**

   The Create Notebook window appears.
3. In the **Name** field, provide a name for the notebook.
4. In the **Comments** field, enter comments, if any.
5. Click **OK.**

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

- `%sql` — To call the SQL interpreter and run SQL statements
- `%script` — To call PL/SQL interpreter and run PL/SQL scripts
- `%md` — To call the Markdown interpreter and generate static html from Markdown plain text
- `%python` — To call the Python interpreter and run Python scripts
- `%r` — To call 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.

5.2 Edit Your 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.

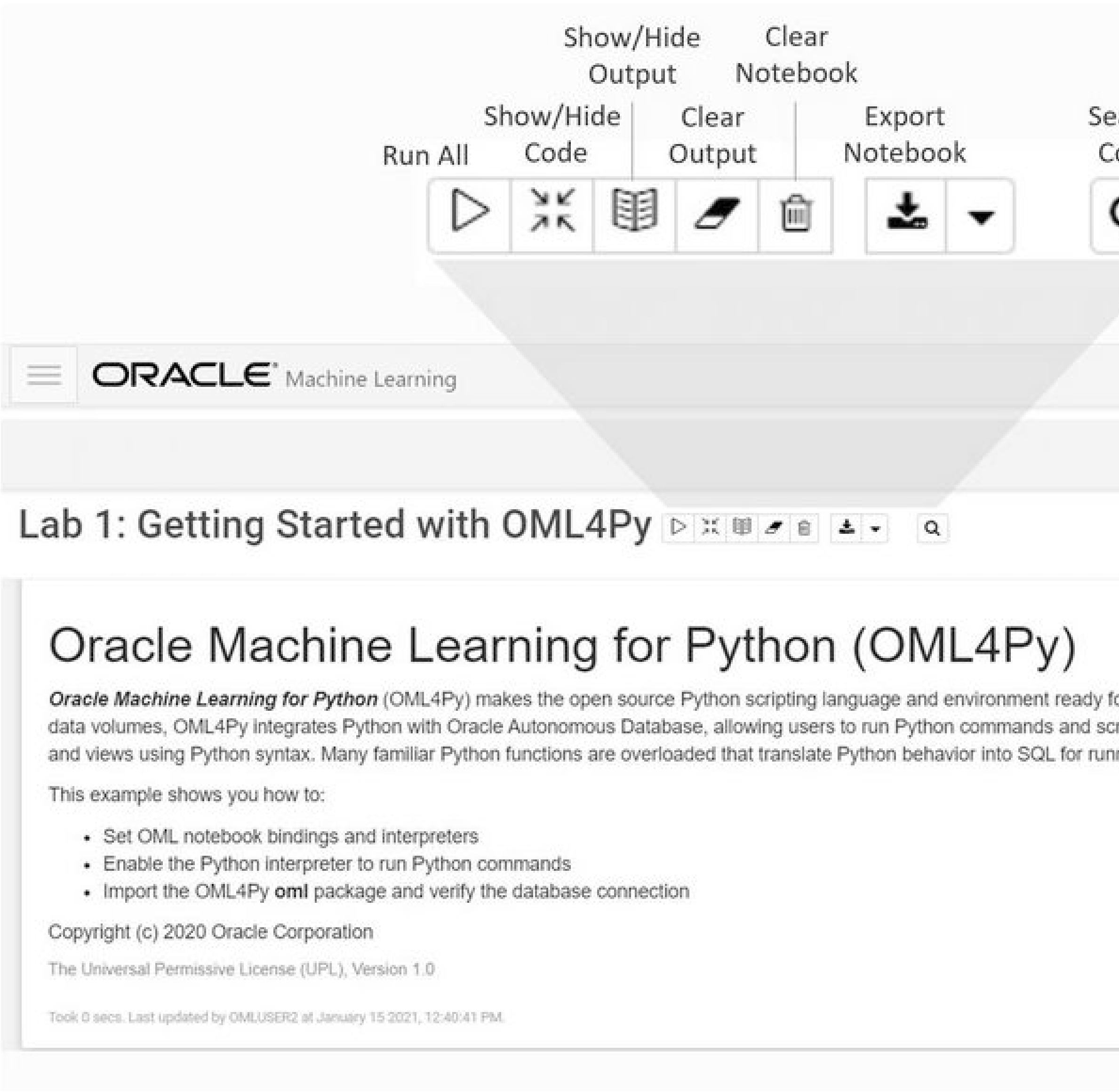
You can edit an existing notebook in your project. To edit an existing notebook:

1. On Oracle Machine Learning UI home page, select the project in which your notebook is available.
2. Go to the Oracle Machine Learning UI navigator, and select **Notebooks.** Alternatively, you can click the **Notebooks** quick link on the home page.

   On the right pane, all notebooks that are available in the project are listed.
3. Click the notebook that you want to open and edit.

   The selected notebook opens in edit mode.
4. In the edit mode, you can use the Oracle Machine Learning notebooks toolbar options to run code in paragraphs, for configuration settings, and display options.

Figure 5-1 Notebook toolbar



Lab 1: Getting Started with OML4Py

Oracle Machine Learning for Python (OML4Py)

Oracle Machine Learning for Python (OML4Py) makes the open source Python scripting language and environment ready fo
data volumes, OML4Py integrates Python with Oracle Autonomous Database, allowing users to run Python commands and scr
and views using Python syntax. Many familiar Python functions are overloaded that translate Python behavior into SQL for runn

This example shows you how to:

- Set OML notebook bindings and interpreters
- Enable the Python interpreter to run Python commands
- Import the OML4Py **oml** package and verify the database connection

Copyright (c) 2020 Oracle Corporation

The Universal Permissive License (UPL), Version 1.0

Took 0 secs. Last updated by OMLUSER2 at January 15 2021, 12:40:41 PM.

You can perform the following tasks:

- Write code to fetch data
- Click ▷ to run one or all paragraphs in the notebook.
- Click ↘↙↗↖ to hide all codes from all the paragraphs in the notebook. Click it again to display the codes.
- Click 📖 to hide all outputs from all the paragraphs in the notebook. Click it again to view the outputs.

- Click ⌫ to remove all outputs from all the paragraphs in the notebook. To view the output, click the run icon again.
- Click 🗑 to delete all the paragraphs in the notebook.
- Click 📥 to export the notebook.
- Click 🔍 to search any information in the codes present in the notebook.
- Click ⌨ to view the list of keyboard shortcuts.
- Click ⚙ to set the order for interpreter bindings for the notebook.
- Click default ▾ to select one of the three notebook display options.
  - Click **default** to view the codes, output, and metadata in all paragraphs in the notebook.
  - Click **Simple** to view only the code and output in all paragraphs in the notebook. In this view, the notebook toolbar and all edit options are hidden. You must hover your mouse to view the edit options.
  - Click **Report** to view only the output in all paragraphs in the notebook.
- Click ⚙ to access paragraph specific edit options such as clear output, remove paragraph, adjust width, font size, run all paragraphs above or below the selected paragraph and so on.
- Add dynamic forms such as the Text Input form, Select form, Check box form for easy selection of inputs and easy filtering of data in your notebook. Oracle Machine Learning supports the following Apache Zeppelin dynamic forms:
  - Text Input form — Allows you to create a simple form for text input.
  - Select form — Allows you to create a form containing a range of values that the user can select.
  - Check Box form — Allows you to insert check boxes for multiple selection of inputs.

> **Note:**
>
> The Apache Zeppelin dynamic forms are supported only on SQL interpreter notebooks.

5. Once you have finished editing the notebook, click **Back.**

This takes you back to the Notebook page.

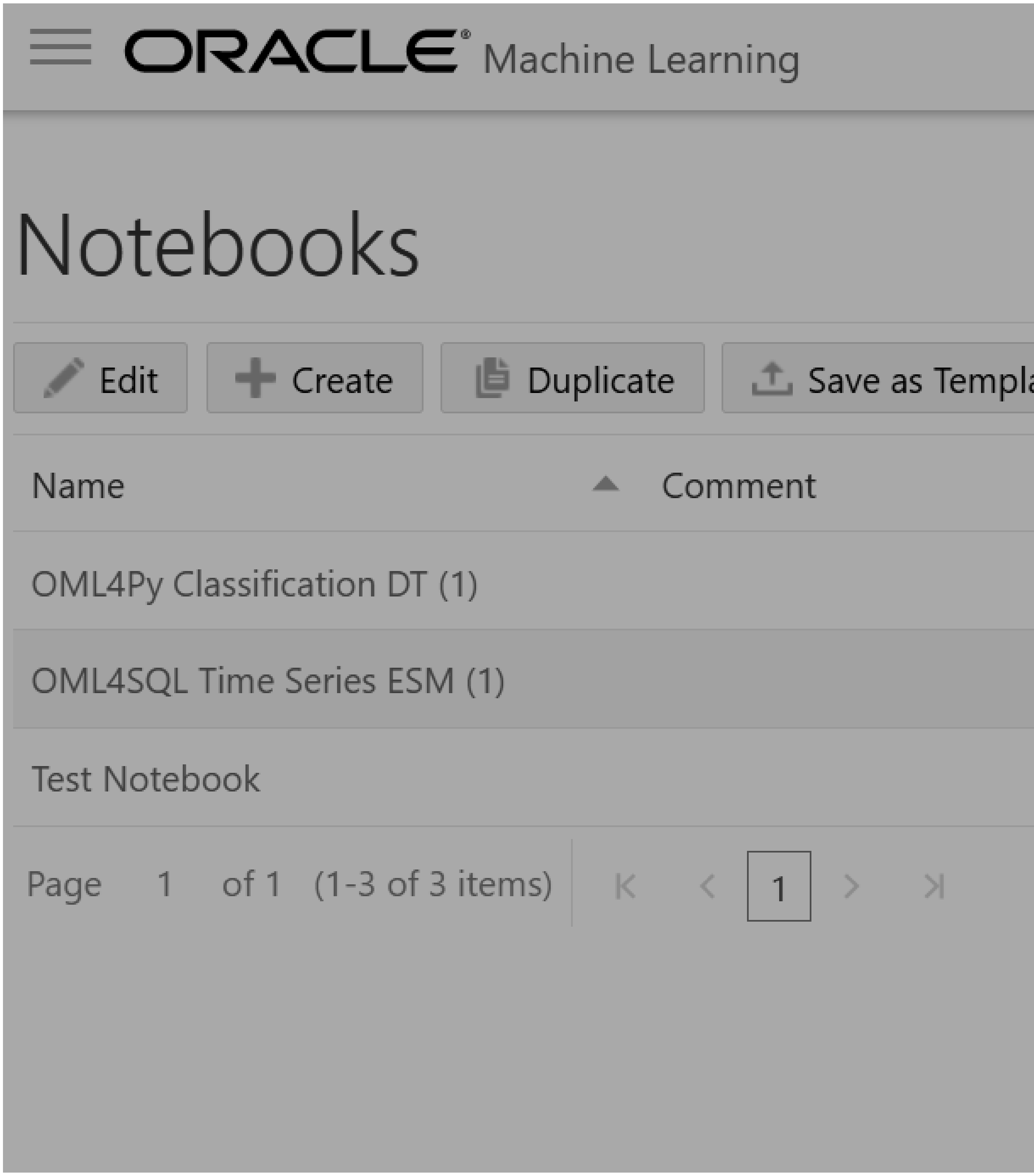
- Export a Notebook
  You can export a notebook in Zeppelin format (`.json`) file and in Jupyter format (`.ipynb`), and later import them in to the same or a different environment.
- Import a Notebook
  You can import notebooks across Pluggable Databases (PDBs) into your workspace. You can also import Jupyter notebooks into Oracle Machine Learning UI.
- Create Check Box Forms in Notebooks
  The Check Box Form supports multiple selection of inputs in a paragraph. The inputs are available as check box options in the notebook.
- Create Select Forms in Notebooks
  The Select Form allows you to select input values from a list of values, and dynamically retrieve the selected values as defined in the paragraph.
- Create Text Input Forms in Notebooks
  The Text Input form allows you to dynamically retrieve values as defined in the notebook.

5.2.1 Export a Notebook

You can export a notebook in Zeppelin format (`.json`) file and 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.
  - **Selected** — To export the selected notebooks.
- **Format** — Select the format in which you want to export your notebook:
  - **Zeppelin** — Exports the notebook as a `.json` (JavaScript Object Notation) file.
  - **Jupyter** — Exports the notebook as a `.ipynb` file.

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

Related Topics

- About Interpreter Bindings and Notebooks
  An interpreter is a plug-in that allows you to use a specific data processing language backend.

5.2.2 Import a Notebook

You can import notebooks across Pluggable Databases (PDBs) into your workspace. You can also import Jupyter notebooks into Oracle Machine Learning UI.

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

> **Note:**
>
> Starting in Oracle Database 20c,"database" refers specifically to the data files of a multitenant container database (CDB), pluggable database (PDB), or application container.

To import a notebook:

1. On Oracle Machine Learning UI home page, click **Notebooks.**
2. On the Notebooks page, click **Import.**

   This opens the **File Upload** dialog. Browse and select the notebook that you want to import.

   > **Note:**
   >
   > You must have the notebook saved as a `.json` 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.

3. In the File Upload dialog, browse and select the `.json` file and click **Open.**

   This imports the notebook file into your workspace.

4. Click the imported notebook to open it. On the Notebooks page, click the gear icon to view the interpreter bindings.

Related Topics

- About Interpreter Bindings and Notebooks
  An interpreter is a plug-in that allows you to use a specific data processing language backend.
- Database and Instance

5.2.3 Create Check Box Forms in Notebooks

The Check Box Form supports multiple selection of inputs in a paragraph. The inputs are available as check box options in the notebook.

To create a Check Box Form:

1. Open the notebook in which you want to add the Check Box Form.
2. In a SQL statement, define the Check Box form by using the syntax:

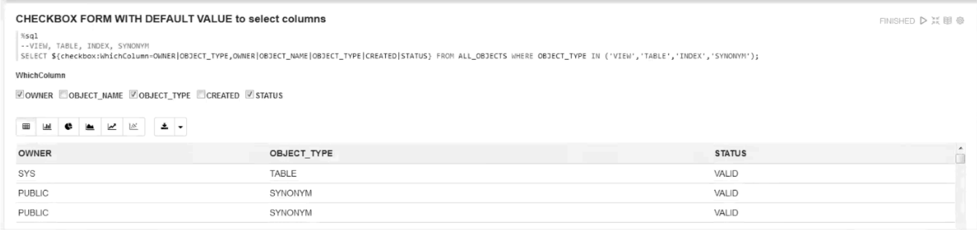
   ```
   ${checkbox:formName=defaultValue1|defaultValue2...,option1|option2...}
   ```

   For example, run the SQL statement:

   ```
   SELECT ${checkbox:whichcolumn=OWNER|OBJECT_TYPE, OWNER|OBJECT_NAME|
   OBJECT_TYPE|CREATED|STATUS} FROM ALL_OBJECTS WHERE OBJECT_TYPE IN
   ('VIEW', 'TABLE', 'INDEX', 'SYNONYM');
   ```

   In this example,

   - The Check Box form is **WhichColumn**
   - The multiple selection options available in the check boxes are **OWNER, OBJECT_NAME, OBJECT_TYPE, CREATED,** and **STATUS**
   - The fields **OWNER** and **OBJECT_TYPE** are defined as default
   - The table name is ALL_OBJECTS
   - The columns that are configured for display are OWNER, OBJECT_NAME, OBJECT_TYPE, CREATED, and STATUS
3. Run the notebook. The Check Box form called WhichForm is available in the notebook, as shown in the screenshot.



CHECKBOX FORM WITH DEFAULT VALUE to select columns

```
%sql
--VIEW, TABLE, INDEX, SYNONYM
SELECT ${checkbox:WhichColumn=OWNER|OBJECT_TYPE,OWNER|OBJECT_NAME|OBJECT_TYPE|CREATED|STATUS} FROM ALL_OBJECTS WHERE OBJECT_TYPE IN ('VIEW','TABLE','INDEX','SYNONYM');
```

WhichColumn

☑ OWNER ☐ OBJECT_NAME ☑ OBJECT_TYPE ☐ CREATED ☑ STATUS

OWNER	OBJECT_TYPE	STATUS
SYS	TABLE	VALID
PUBLIC	SYNONYM	VALID
PUBLIC	SYNONYM	VALID

5.2.4 Create Select Forms in Notebooks

The Select Form allows you to select input values from a list of values, and dynamically retrieve the selected values as defined in the paragraph.

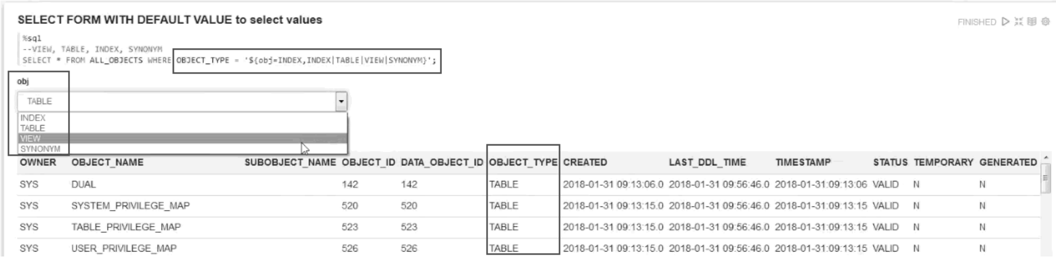
To create a Select form:

1. Open the notebook in which you want to add the text input form.
2. In a SQL statement, define the Select form by using the syntax:

   ```
   ${formName=defaultValue,option1|option2...}.
   ```

   For example, run the SQL statement:

   ```
   SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${OBJ=INDEX,INDEX|
   TABLE|VIEW|SYNONYM}';
   ```



SELECT FORM WITH DEFAULT VALUE to select values

```
%sql
--VIEW, TABLE, INDEX, SYNONYM
SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${obj=INDEX,INDEX|TABLE|VIEW|SYNONYM}';
```

obj: TABLE

OWNER	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE	CREATED	LAST_DDL_TIME	TIMESTAMP	STATUS	TEMPORARY	GENERATED
SYS	DUAL		142	142	TABLE	2018-01-31 09:13:06.0	2018-01-31 09:56:46.0	2018-01-31:09:13:06	VALID	N	N
SYS	SYSTEM_PRIVILEGE_MAP		520	520	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	TABLE_PRIVILEGE_MAP		523	523	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	USER_PRIVILEGE_MAP		526	526	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N

In this example,

- The form name is `obj`
- The list of available values are `INDEX, TABLE, VIEW, SYNONYM.`
- The table name is ALL_OBJECTS
- The column name is OBJECT_TYPE

Select any values from the drop-down list in the **obj** form. The selected value will be retrieved in the OBJECT_TYPE column in the ALL_OBJECTS table.

5.2.5 Create Text Input Forms in Notebooks

The Text Input form allows you to dynamically retrieve values as defined in the notebook.

To create a Text Input form:

1. Open the notebook in which you want to add the Text Input form.
2. In a SQL statement, define the Text Input form by using the syntax:

   ```
   ${formName}
   ```

   For example, run the SQL statement:

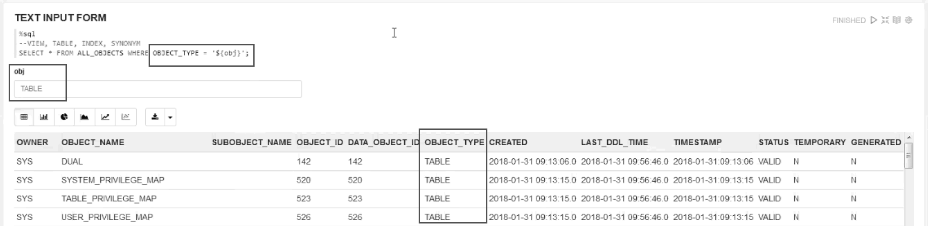
   ```
   SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${OBJ}';
   ```

   In this example,

   - The form name is `obj`
   - The table name is ALL_OBJECTS
   - The column name is OBJECT_TYPE

   Here, the text form **obj** is created for the column OBJECT_TYPE in the table ALL_OBJECTS. You can enter different values in the form field **obj** and run the notebook to retrieve the corresponding values in the column OBJECT_TYPE.
3. Run the paragraph. The notebook now displays the text input form field **obj,** as shown in the screenshot. You can enter values in the **obj** field, and run the notebook to retrieve the corresponding values for the column OBJECT_TYPE in the table ALL_OBJECTS.

- If you enter TABLE in the **obj** field, and run the notebook, then the notebook retrieves TABLE in the column OBJECT_TYPE, as shown in the screenshot.



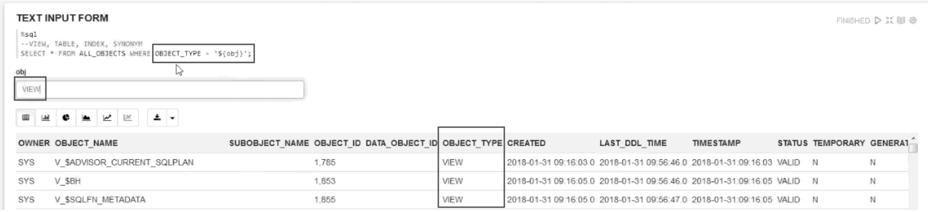
TEXT INPUT FORM

```
%sql
--VIEW, TABLE, INDEX, SYNONYM
SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${obj}';
```

obj: TABLE

OWNER	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE	CREATED	LAST_DDL_TIME	TIMESTAMP	STATUS	TEMPORARY	GENERATED
SYS	DUAL		142	142	TABLE	2018-01-31 09:13:06.0	2018-01-31 09:56:46.0	2018-01-31:09:13:06	VALID	N	N
SYS	SYSTEM_PRIVILEGE_MAP		520	520	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	TABLE_PRIVILEGE_MAP		523	523	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	USER_PRIVILEGE_MAP		526	526	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N

- If you enter VIEW in the **obj** form field and run the notebook, then the notebook retrieves the value VIEW in the column OBJECT_TYPE, as shown in the screenshot.



TEXT INPUT FORM

```
%sql
--VIEW, TABLE, INDEX, SYNONYM
SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${obj}';
```

obj: VIEW

OWNER	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE	CREATED	LAST_DDL_TIME	TIMESTAMP	STATUS	TEMPORARY	GENERAT
SYS	V_$ADVISOR_CURRENT_SQLPLAN		1,785		VIEW	2018-01-31 09:16:03.0	2018-01-31 09:56:46.0	2018-01-31:09:16:03	VALID	N	N
SYS	V_$BH		1,853		VIEW	2018-01-31 09:16:05.0	2018-01-31 09:56:46.0	2018-01-31:09:16:05	VALID	N	N
SYS	V_$SQLFN_METADATA		1,855		VIEW	2018-01-31 09:16:05.0	2018-01-31 09:56:47.0	2018-01-31:09:16:05	VALID	N	N

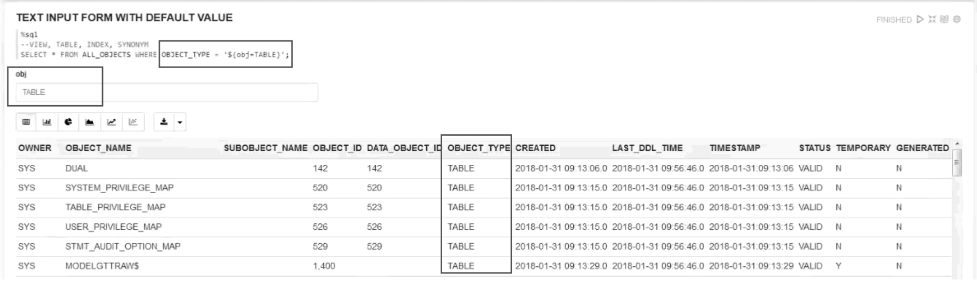
4. You can also assign default values in the form by using the syntax:

   `${formName=defaultValue}`

   To assign a default value to the Text Input form, modify the SQL statement to:

   `SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${obj=TABLE}'`

   Here, the default value assigned to the form is TABLE. Once you run the paragraph, the default value TABLE will be retrieved in the column OBJECT_TYPE, as shown in the screenshot.



TEXT INPUT FORM WITH DEFAULT VALUE

```
%sql
--VIEW, TABLE, INDEX, SYNONYM
SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${obj=TABLE}';
```

obj: TABLE

OWNER	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE	CREATED	LAST_DDL_TIME	TIMESTAMP	STATUS	TEMPORARY	GENERATED
SYS	DUAL		142	142	TABLE	2018-01-31 09:13:06.0	2018-01-31 09:56:46.0	2018-01-31:09:13:06	VALID	N	N
SYS	SYSTEM_PRIVILEGE_MAP		520	520	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	TABLE_PRIVILEGE_MAP		523	523	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	USER_PRIVILEGE_MAP		526	526	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	STMT_AUDIT_OPTION_MAP		529	529	TABLE	2018-01-31 09:13:15.0	2018-01-31 09:56:46.0	2018-01-31:09:13:15	VALID	N	N
SYS	MODELGTTRAW$		1,400		TABLE	2018-01-31 09:13:29.0	2018-01-31 09:56:46.0	2018-01-31:09:13:29	VALID	Y	N

5.3 Version a Notebook

You can version or create a backup a notebook, experiment on it, and revert to the original notebook, if required.

To version a notebook:

1. On the Notebooks page, select the notebook that you want to version and click **Version.**

   The Versions page opens.

2. On the Versions page for the selected notebook, click **+ Version.**

   The Create Versions dialog box opens.

3. In the Create Versions dialog box, enter comments for the specific version of your notebook, and click **OK.**
4. The versioned notebook is now listed on the Versions page. You can perform the following tasks:
   - Click **Revert Version** to restore the older version of your notebook.
   - Click **Delete** to delete the selected version of your notebook.
   - Click **New Notebook** to create a new notebook from the selected notebook version.

5.4 Save Notebooks as Templates

You can save an existing notebook as a template in Personal or in Shared.

To save a notebook as a template:

1. On the Notebooks page, select the notebook that you want to save as template and click **Save as Template.**

   The Save as Template dialog box opens.
2. In the **Name** field, enter a name for the notebook template.
3. In the **Comments** field, enter comments, if any.
4. In the **Tags** field, enter tags for the template.
5. In **Save To**, select:
   - **Personal:** If you want to save this notebook template to Personal such that only your account can view or use this notebook.
   - **Shared:** If you want to save and share this notebook template such that other users can view and create notebooks from this template that they can run and edit.

5.5 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 UI 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 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.

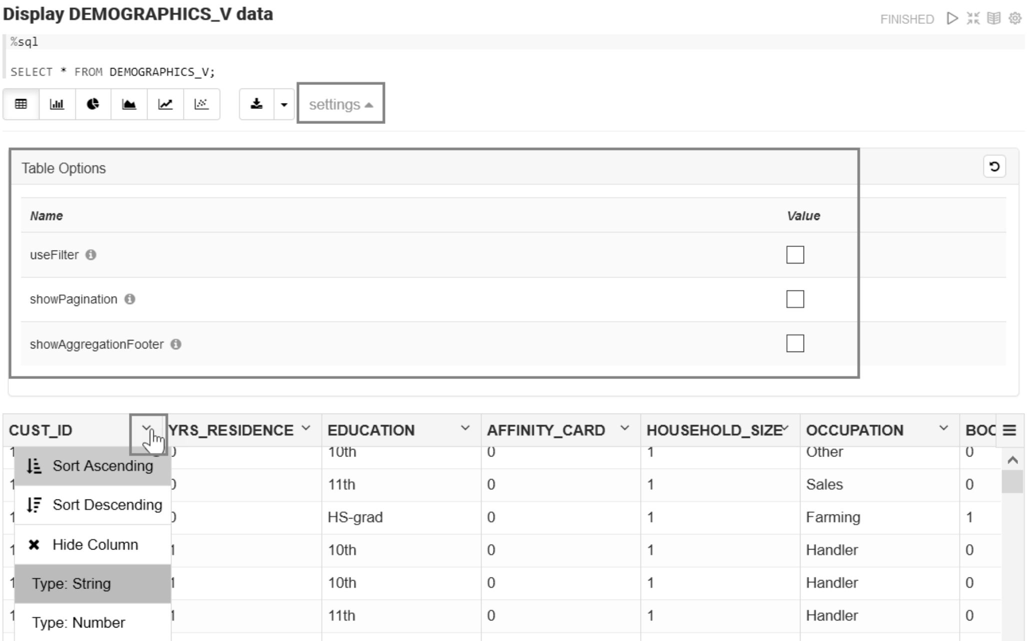
   The results of the interpreter appear in the output section. The output section of the paragraph comprises a charting component that displays the results in graphical output. The chart interface allows you to interact with the output in the notebook paragraph. You have the option to run and edit single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



Display DEMOGRAPHICS_V data

```
%sql

SELECT * FROM DEMOGRAPHICS_V;
```

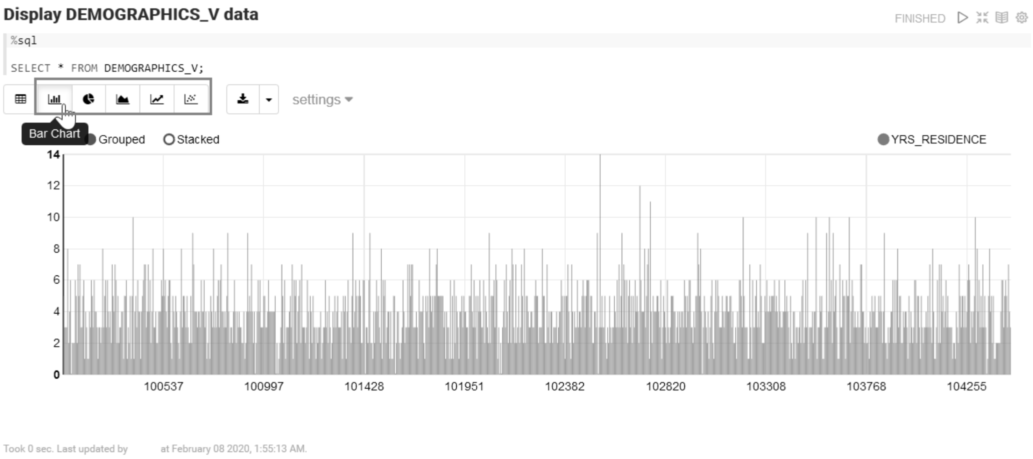
Table Options

Name	Value
useFilter	□
showPagination	□
showAggregationFooter	□

CUST_ID	YRS_RESIDENCE	EDUCATION	AFFINITY_CARD	HOUSEHOLD_SIZE	OCCUPATION	BOC
1		10th	0	1	Other	0
1		11th	0	1	Sales	0
1		HS-grad	0	1	Farming	1
1	1	10th	0	1	Handler	0
1	1	10th	0	1	Handler	0
1	1	11th	0	1	Handler	0

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.



Display DEMOGRAPHICS_V data

```
%sql

SELECT * FROM DEMOGRAPHICS_V;
```

Took 0 sec. Last updated by at February 08 2020, 1:55:13 AM.

- About Oracle Machine Learning for SQL
Oracle Machine Learning for SQL (OML4SQL) provides a powerful, state-of-the-art machine learning capability within Oracle 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 Database Interpreter Connection
  When defining an Oracle Database interpreter connection, a reference to a compute resource is created. This reference contains all connection-related information about the interpreter.
- Edit Oracle Database Interpreter Connection
  When defining an Oracle Database interpreter connection, a reference to a compute resource is created. This reference contains all connection-related information about the interpreter.

5.5.1 About Oracle Machine Learning for SQL

Oracle Machine Learning for SQL (OML4SQL) provides a powerful, state-of-the-art machine learning capability within Oracle 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 Notebook 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

5.5.2 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.

5.5.3 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.

5.6 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 UI 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 Database

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

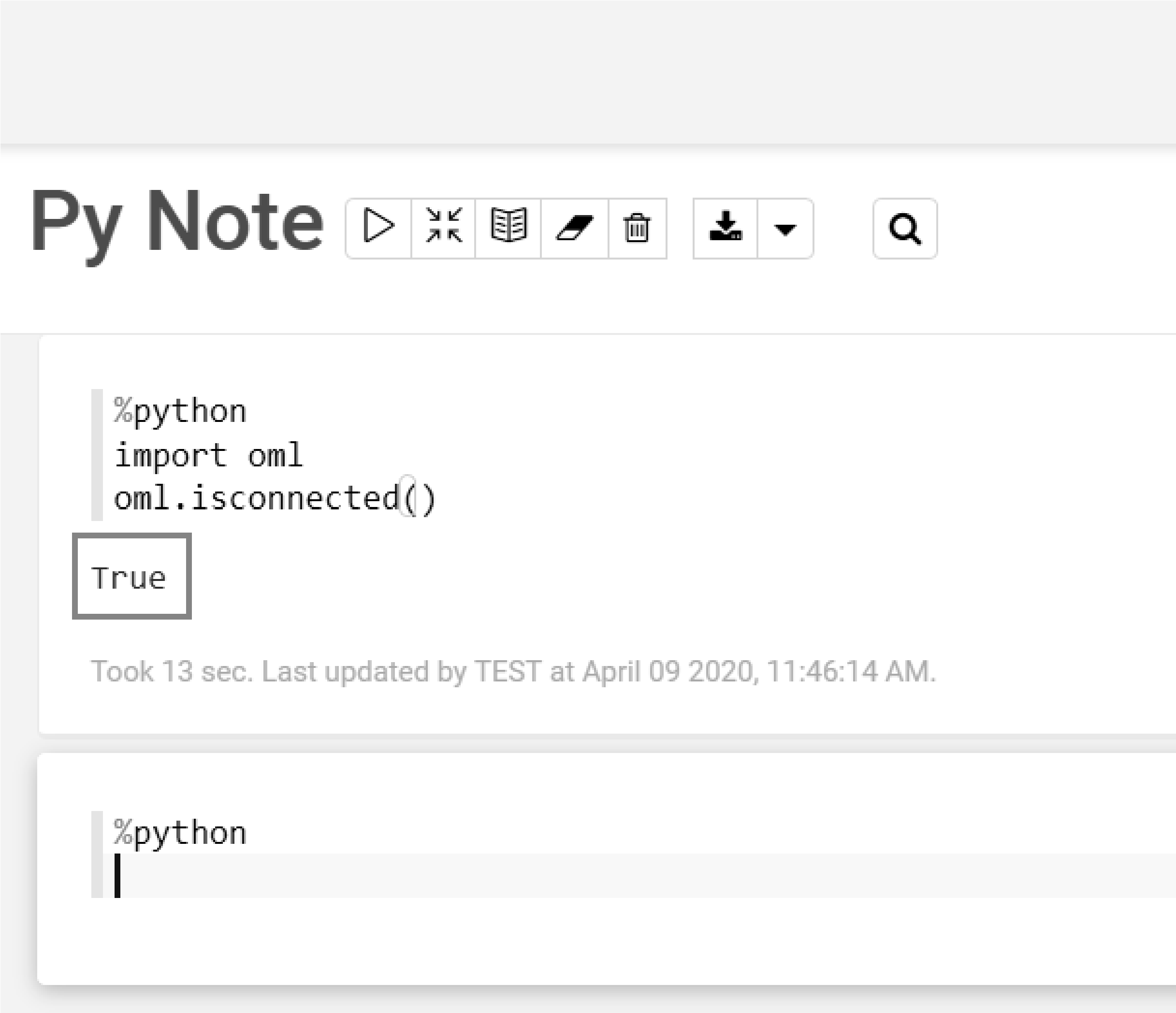
Assumption: The example assumes that you have created a new notebook named **Py Note.**

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()
   ```



```
%python
import oml
oml.isconnected()
```

```
True
```

Took 13 sec. Last updated by TEST at April 09 2020, 11:46:14 AM.

```
%python
```

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

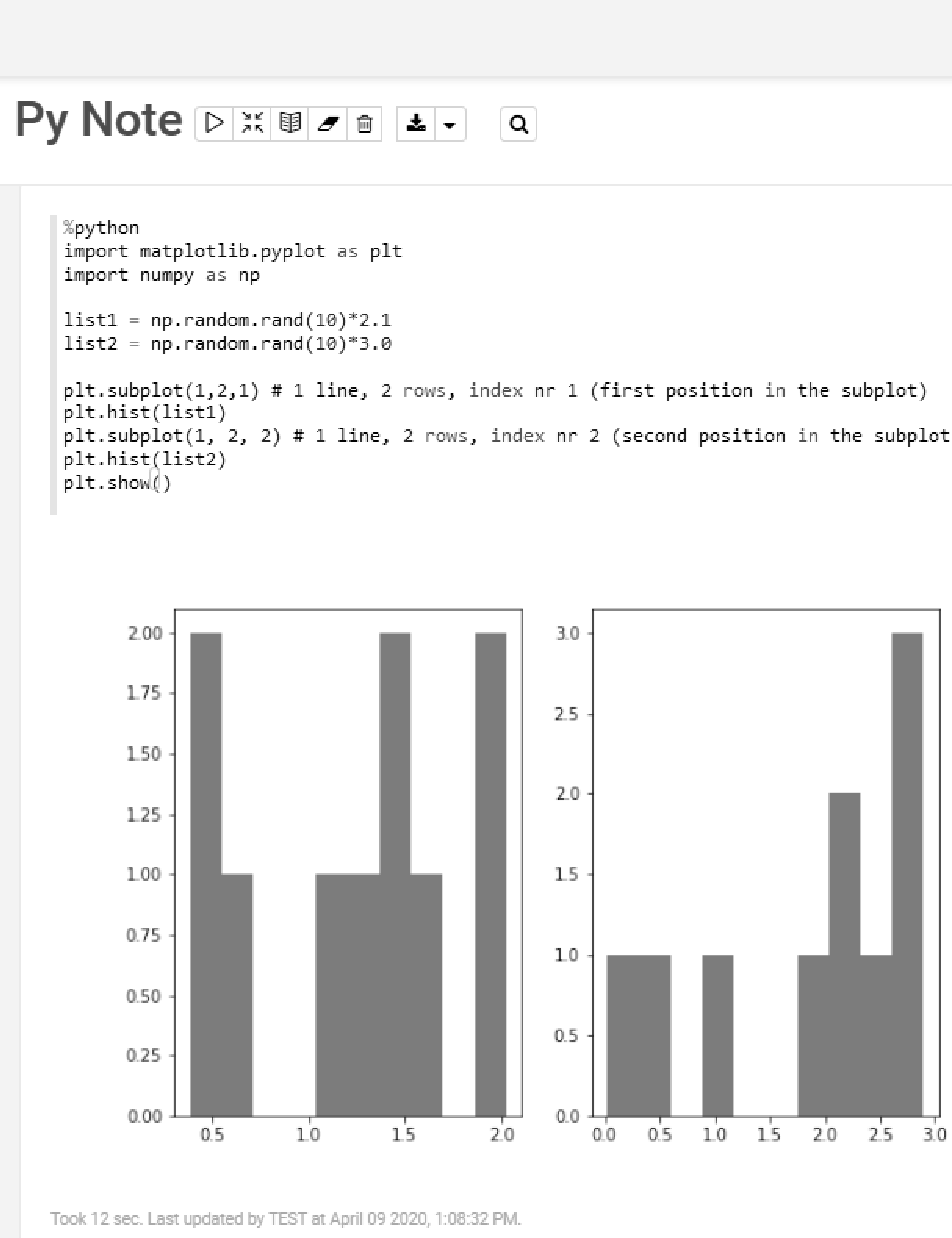
6. Type the following commands to compute and render the data in two histograms.

   ```
   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.**



```
%python
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 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()
```

Took 12 sec. Last updated by TEST at April 09 2020, 1:08:32 PM.

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 Database, which includes Oracle Autonomous Data Warehouse (ADW), Oracle Autonomous Transaction Processing (ATP), and Oracle Autonomous 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 Database as a high-performance computing environment. Oracle Machine Learning User Interface (UI) is available through Autonomous Data Warehouse (ADW) , Autonomous Transaction Processing (ATP) and Autonomous JSON Database (AJD) services.

5.6.1 About Oracle Machine Learning for Python

Oracle Machine Learning for Python (OML4Py) is a component of Oracle Autonomous Database, which includes Oracle Autonomous Data Warehouse (ADW), Oracle Autonomous Transaction Processing (ATP), and Oracle Autonomous 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 Database as a high-performance computing environment. Oracle Machine Learning User Interface (UI) is available through Autonomous Data Warehouse (ADW) , Autonomous Transaction Processing (ATP) and Autonomous 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 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 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 3.8.5 to process Python scripts in Oracle Machine Learning UI notebooks. To use the interpreter, specify the `%python` directive at the beginning of the paragraph. The Python interpreter supports the following Python modules:

- `cx_Oracle 7.3.0`
- `cycler 0.10.0`
- `joblib 0.14.0`
- `kiwisolver 1.1.0`
- `matplotlib 3.1.2`
- `numpy 1.18.1`
- `pandas 0.25.3`
- `pyparsing 2.4.0`
- `python-dateutil 2.8.1`
- `pytz 2019.3`

- `scikit_learn 0.22.1`
- `scipy 1.4.1`
- `six 1.13.0`

Related Topics

- About Oracle Machine Learning for Python

5.7 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 UI 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 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
   ```

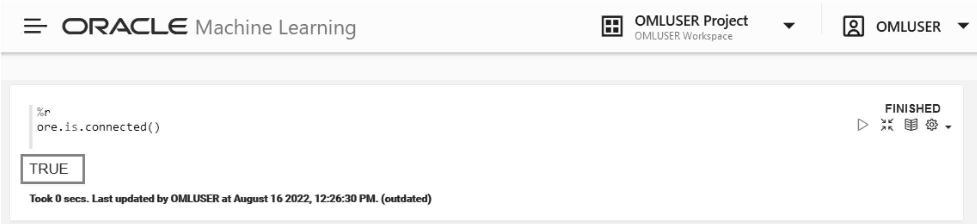
   To know more about interpreters, see About Interpreter Bindings and Notebooks and Change Interpreter Bindings for Specific Paragraphs in a Notebook.

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 5-2 Test Database Connection



```
%r
ore.is.connected()
```

```
TRUE
```

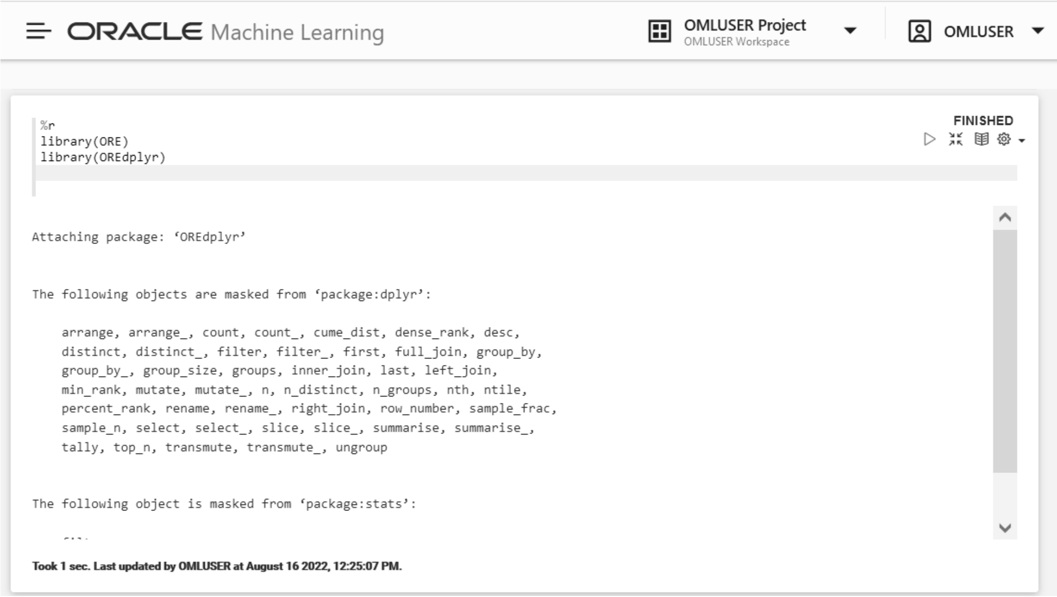
Took 0 secs. Last updated by OMLUSER at August 16 2022, 12:26:30 PM. (outdated)

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 5-3 Commands to Load R Packages



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

```
Attaching package: ‘OREdplyr’


The following objects are masked from ‘package:dplyr’:

    arrange, arrange_, count, count_, cume_dist, dense_rank, desc,
    distinct, distinct_, filter, filter_, first, full_join, group_by,
    group_by_, group_size, groups, inner_join, last, left_join,
    min_rank, mutate, mutate_, n, n_distinct, n_groups, nth, ntile,
    percent_rank, rename, rename_, right_join, row_number, sample_frac,
    sample_n, select, select_, slice, slice_, summarise, summarise_,
    tally, top_n, transmute, transmute_, ungroup


The following object is masked from ‘package:stats’:
```

Took 1 sec. Last updated by OMLUSER at August 16 2022, 12:25:07 PM.

- 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 Database.

5.7.1 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 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 Database, including Autonomous Data Warehouse, Autonomous Transaction Processing, and Autonomous JSON Database. Oracle Machine Learning for R Embedded R Execution functionality can be deployed through SQL and REST APIs on Autonomous Database.

Use Oracle Machine Learning for R to:

- Perform data exploration and data preparation while seamlessly leveraging Oracle Database 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`
- `OERxml_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`

5.8 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

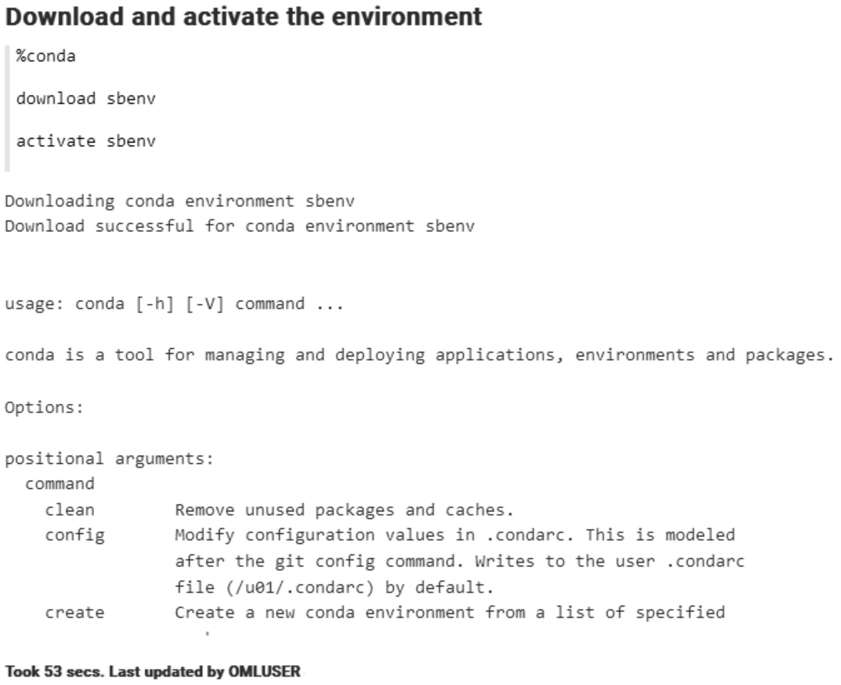
1. Type `%conda` at the beginning of the paragraph to connect to the Conda interpreter, and press Enter.

   ```
   %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`.



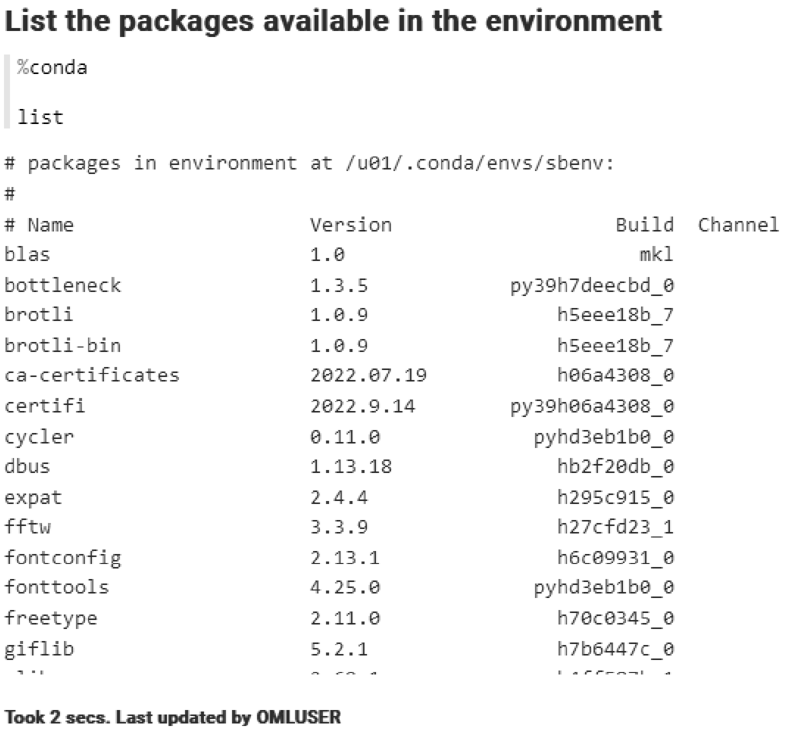
Download and activate the environment

```
%conda

download sbenv

activate sbenv
```

```
Downloading conda environment sbenv
Download successful for conda environment sbenv


usage: conda [-h] [-V] command ...

conda is a tool for managing and deploying applications, environments and packages.

Options:

positional arguments:
  command
    clean        Remove unused packages and caches.
    config       Modify configuration values in .condarc. This is modeled
                 after the git config command. Writes to the user .condarc
                 file (/u01/.condarc) by default.
    create       Create a new conda environment from a list of specified
```

Took 53 secs. Last updated by OMLUSER

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



List the packages available in the environment

```
%conda

list
```

```
# packages in environment at /u01/.conda/envs/sbenv:
#
# Name                    Version                   Build  Channel
blas                      1.0                         mkl
bottleneck                1.3.5            py39h7deecbd_0
brotli                    1.0.9                h5eee18b_7
brotli-bin                1.0.9                h5eee18b_7
ca-certificates           2022.07.19           h06a4308_0
certifi                   2022.9.14        py39h06a4308_0
cycler                    0.11.0             pyhd3eb1b0_0
dbus                      1.13.18              hb2f20db_0
expat                     2.4.4                h295c915_0
fftw                      3.3.9                h27cfd23_1
fontconfig                2.13.1               h6c09931_0
fonttools                 4.25.0             pyhd3eb1b0_0
freetype                  2.11.0               h70c0345_0
giflib                    5.2.1                h7b6447c_0
```

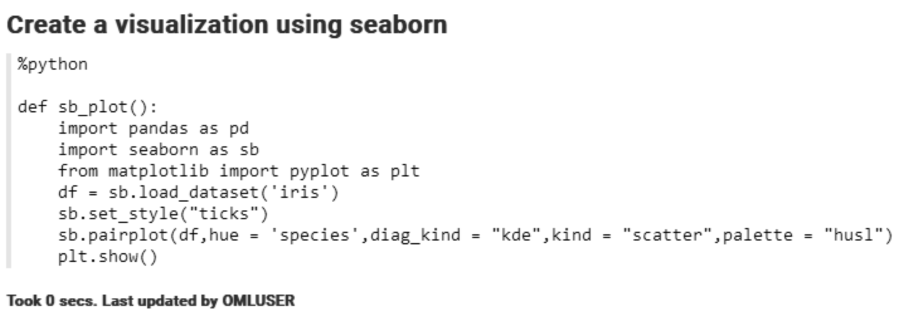
Took 2 secs. Last updated by OMLUSER

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

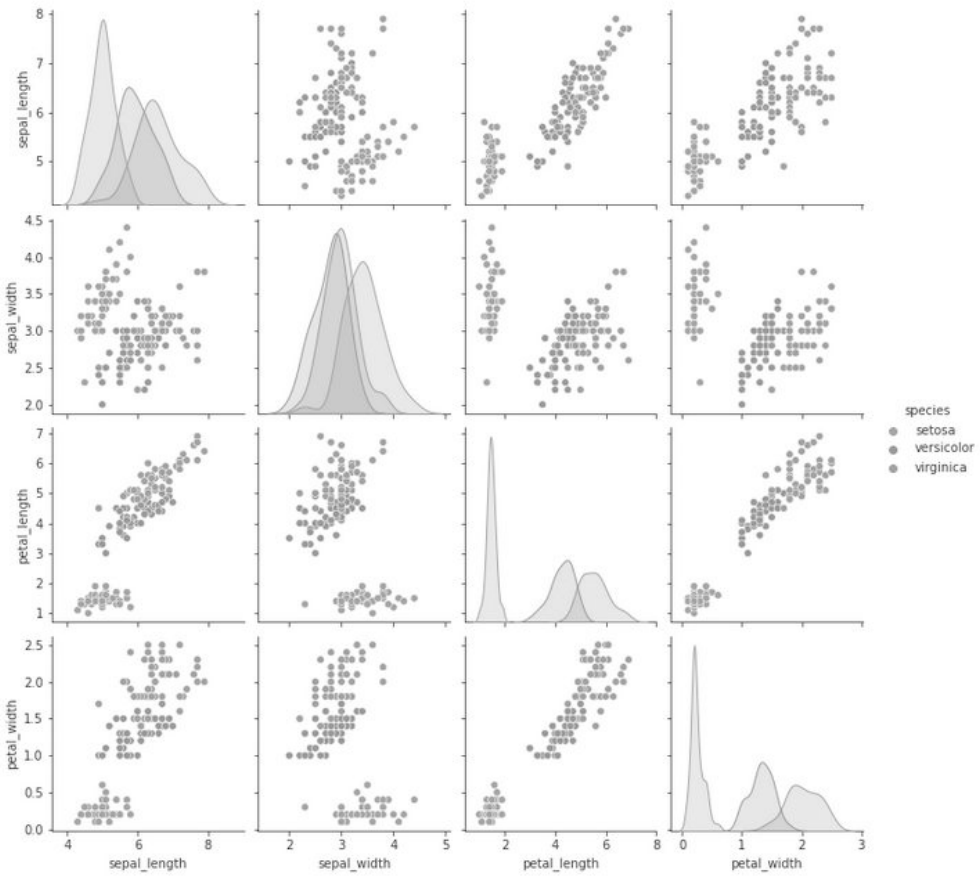
```
%python

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.

5.8.1 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:**
   >
   > The Conda environment is created by the ADMIN user with `OML_SYS_ADMIN` role. 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 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
   ```

5.8.2 Conda Interpreter Commands

This table lists the commands for the Conda interpreter.

Conda Interpreter Commands

Table 5-1   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
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

Table 5-1 (Cont.) Conda Interpreter Commands

Tasks	Commands	Role
Upload a Conda environment to object storage. **Note:** This is an Autonomous 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
Download and unpack a specific Conda environment from the object storage. **Note:** This is an Autonomous Database specific command.	`download --overwrite <env_name>`	- ADMIN - OML user
List local environments available to the user.	`list-local-envs`	- ADMIN - OML user

Table 5-1 (Cont.) Conda Interpreter Commands

Tasks	Commands	Role
List all Conda environments in the object storage. **Note:** This is an Autonomous Database specific command.	`list-saved-envs --installed-packages -e <env_name>`	- ADMIN - OML user
Delete a Conda environment. **Note:** This is an Autonomous Database specific command.	`delete <env_name>` **Note:** The Conda environment is deleted from the Object Storage.	- 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

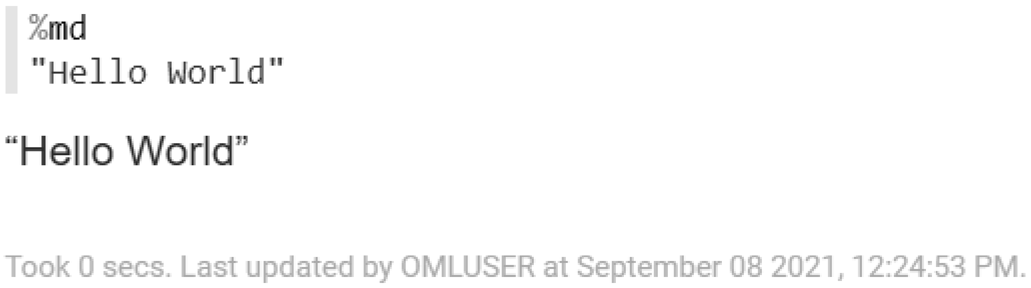
5.9 Call the Markdown Interpreter and Generate Static html from Markdown Plain Text

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.

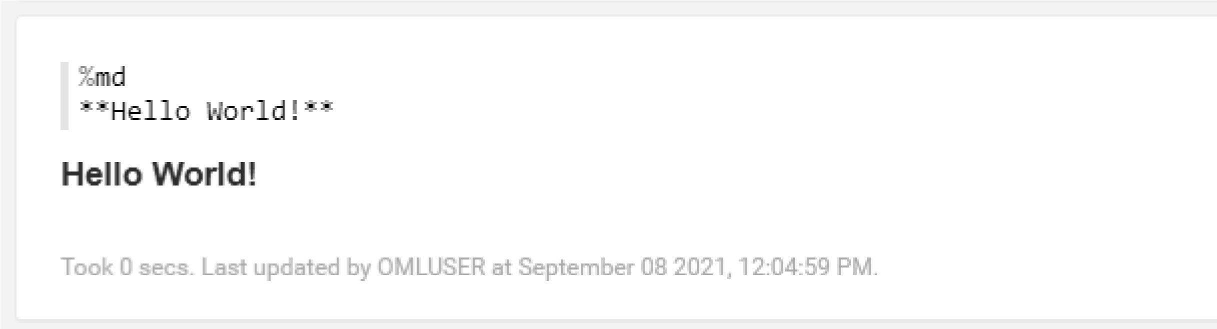


```
%md
"Hello World"
```

“Hello World”

Took 0 secs. Last updated by OMLUSER at September 08 2021, 12:24:53 PM.

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.

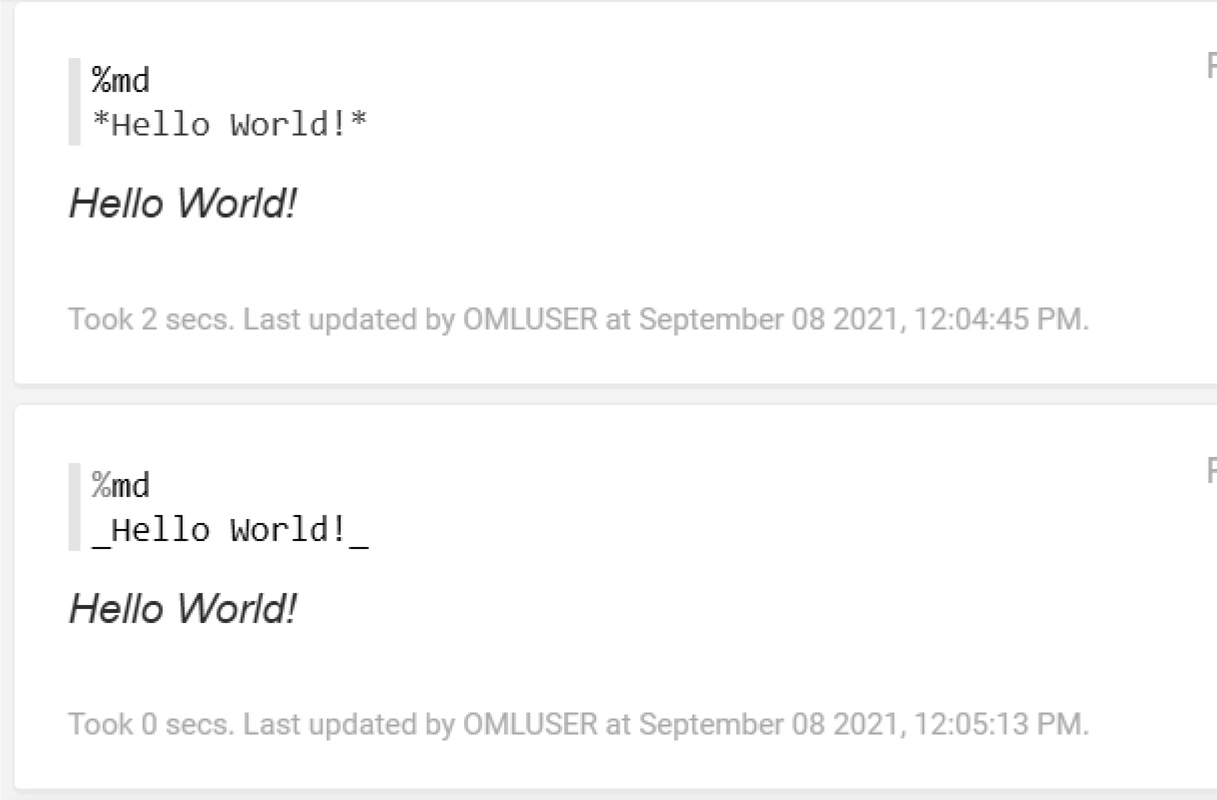


```
%md
**Hello World!**
```

Hello World!

Took 0 secs. Last updated by OMLUSER at September 08 2021, 12:04:59 PM.

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.



```
%md
*Hello World!*
```

Hello World!

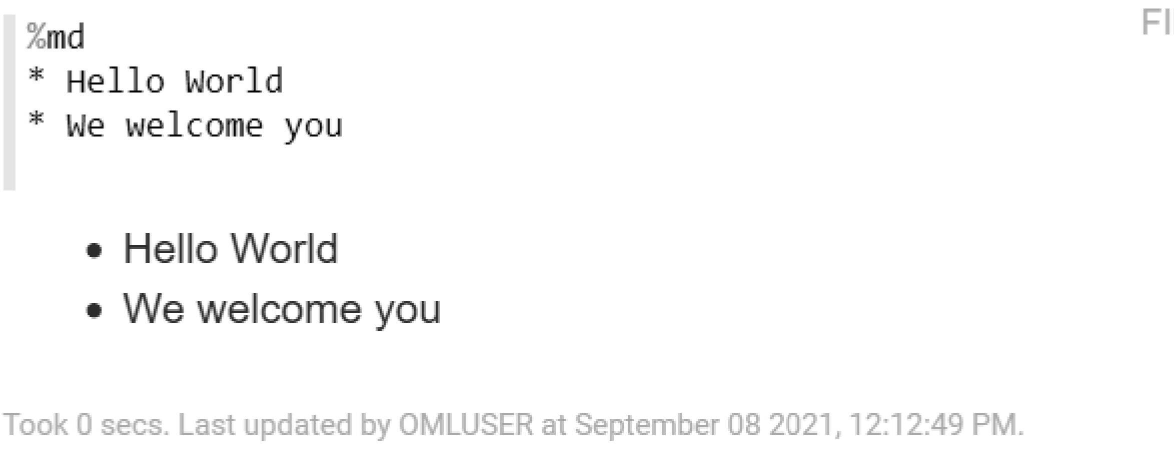
Took 2 secs. Last updated by OMLUSER at September 08 2021, 12:04:45 PM.

```
%md
_Hello World!_
```

Hello World!

Took 0 secs. Last updated by OMLUSER at September 08 2021, 12:05:13 PM.

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

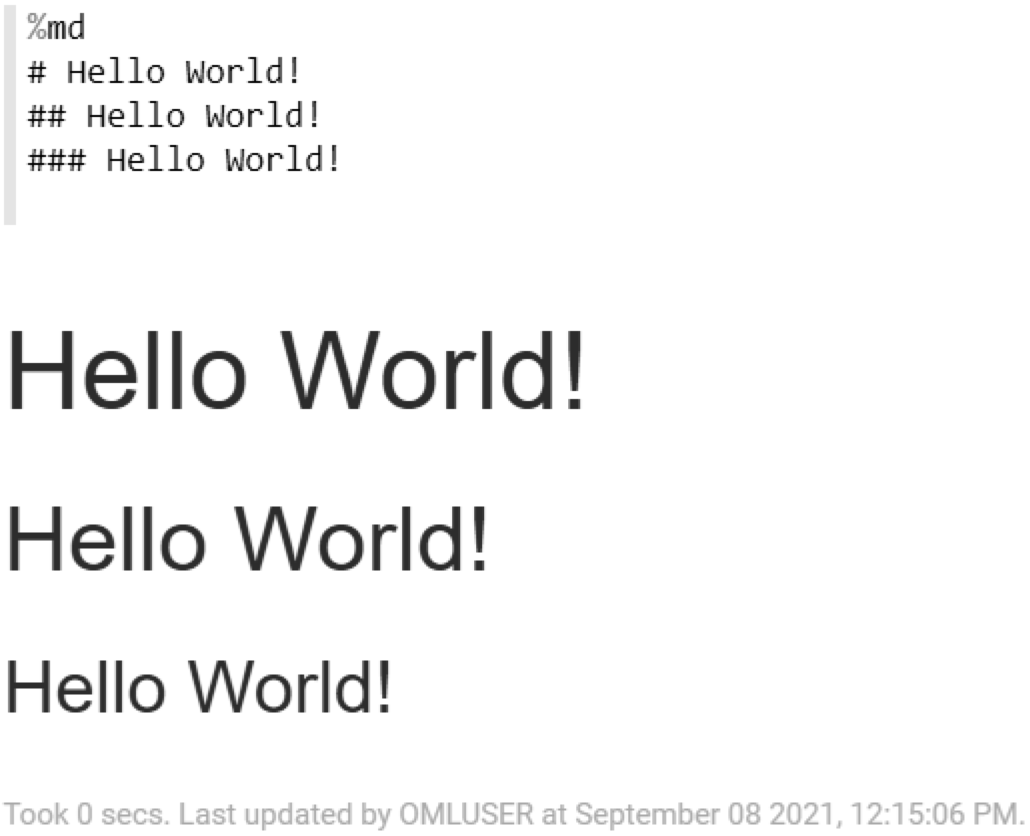


```
%md
* Hello World
* We welcome you
```

- Hello World
- We welcome you

Took 0 secs. Last updated by OMLUSER at September 08 2021, 12:12:49 PM.

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.



```
%md
# Hello World!
## Hello World!
### Hello World!
```

Hello World!

Hello World!

Hello World!

Took 0 secs. Last updated by OMLUSER at September 08 2021, 12:15:06 PM.

5.11 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.

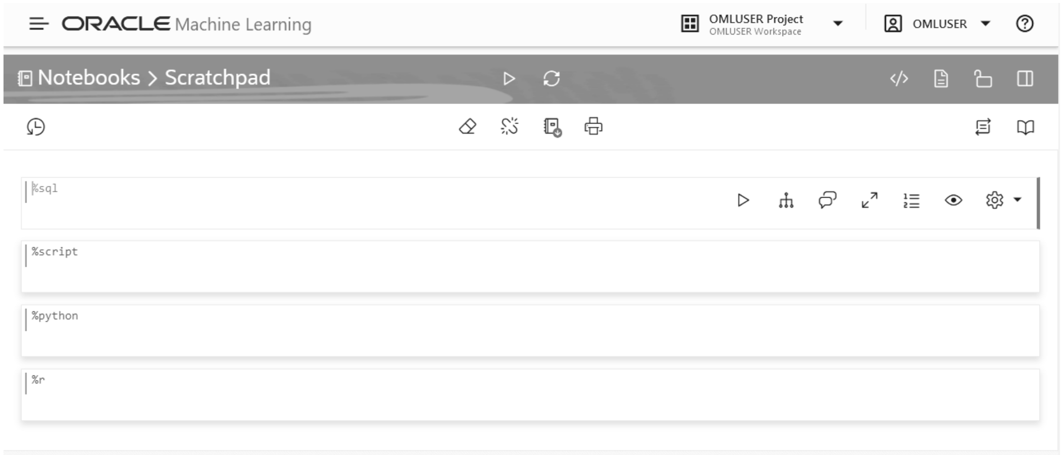
> **Note:**
>
> The Scratchpad is a regular notebook that is prepopulated with four paragraphs - `%sql, %script, ,` `%python` and `%r.`

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 three 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 5-4 Scratchpad



```
%sql
```

```
%script
```

```
%python
```

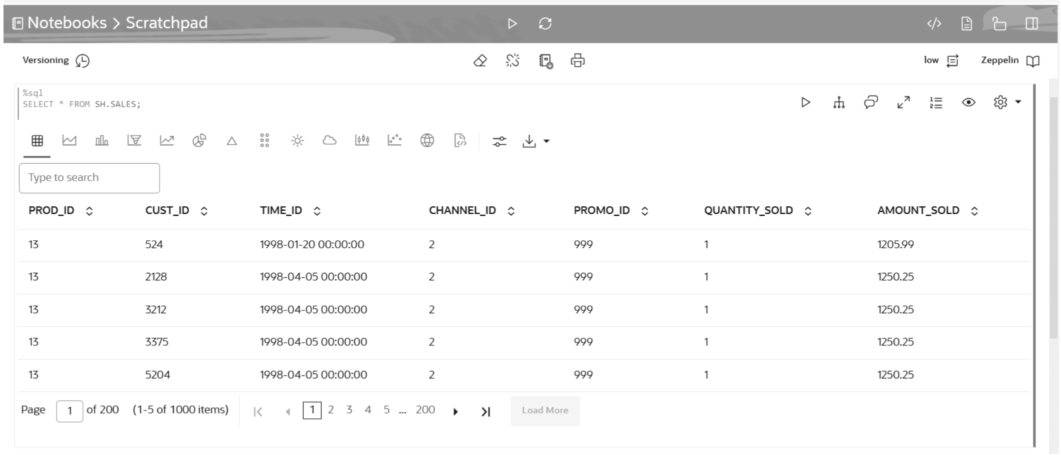
```
%r
```

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 5-5 SQL Statement in Scratchpad



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

PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_SOLD
13	524	1998-01-20 00:00:00	2	999	1	1205.99
13	2128	1998-04-05 00:00:00	2	999	1	1250.25
13	3212	1998-04-05 00:00:00	2	999	1	1250.25
13	3375	1998-04-05 00:00:00	2	999	1	1250.25
13	5204	1998-04-05 00:00:00	2	999	1	1250.25

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;
```

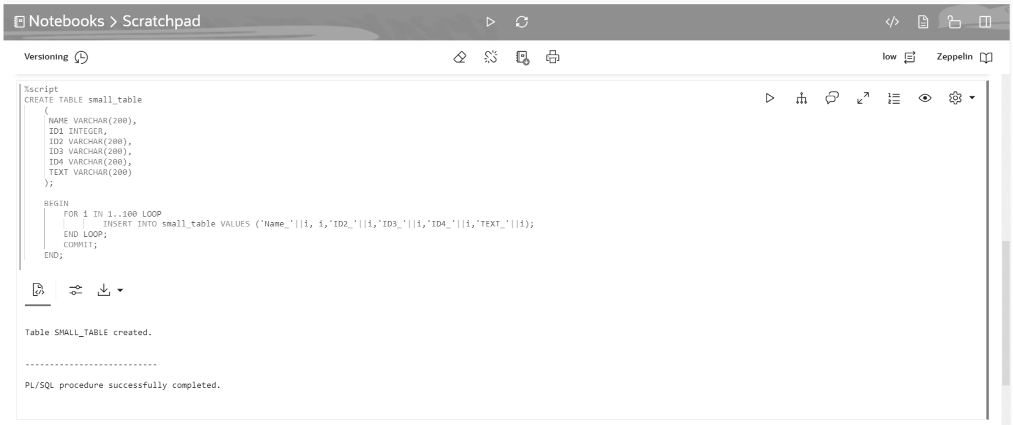
The PL/SQL script successfully creates the table SMALL_TABLE. The PL/SQL script in this example contains two parts:

- 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 5-6 PL/SQL Script in Scratchpad



```
%script
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;
```

```
Table SMALL_TABLE created.


---------------------------

PL/SQL procedure successfully completed.
```

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 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 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.

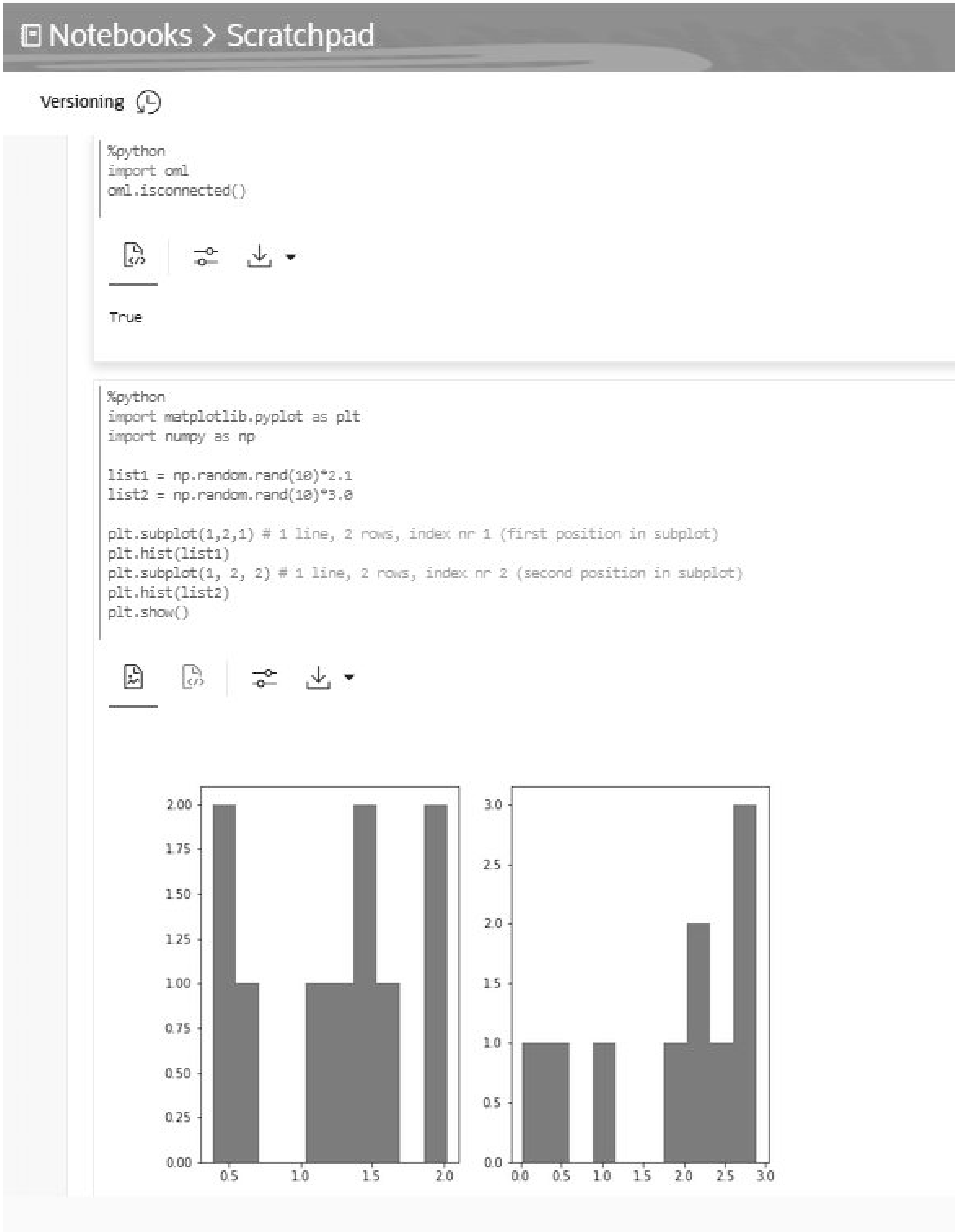
   ```
   import matplotlib.pyplot as plt
   import numpy as np

   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 5-7 Python Script in Scratchpad



```
%python
import oml
oml.isconnected()
```

```
True
```

```
%python
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()
```

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.**

5.12 Collaborate in Oracle Machine Learning

Two or more users can collaborate and share Oracle Machine Learning User Interface (UI) notebooks with other users.

You can collaborate by:

- Granting Access to Workspace of Another User
- Using the Export Option
- Using Oracle Machine Learning UI notebooks templates

Collaborating in Oracle Machine Learning

5.10 About Interpreter Bindings and Notebooks

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 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 these 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 bindings that define the order of the interpreter bindings in an interpreter group. The default order of interpreter bindings in the Oracle 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 execute 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 execute 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.

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 Interpreter Bindings for Specific Paragraphs in a Notebook
  The interpreter binding order that is set for a notebook applies to all the paragraphs in that notebook. However, you can override the interpreter binding for SQL, PL/SQL, R, Python and Conda interpreters for any specific paragraph in the notebook.
- Set Interpreter Bindings for Notebooks
  You must bind a notebook to an interpreter to fetch data from the database or any data source. A default set of interpreter bindings is available.
- Verify Interpreter Bindings
  After setting and changing the order of interpreter bindings, you can verify the interpreter bindings whether you are using SQL, PL/SQL, R or Python interpreter in a given notebook. You use a SQL statement to view and verify the interpreter binding information about your notebook.

5.10.1 Change Interpreter Bindings for Specific Paragraphs in a Notebook

The interpreter binding order that is set for a notebook applies to all the paragraphs in that notebook. However, you can override the interpreter binding for SQL, PL/SQL, R, Python and Conda interpreters for any specific paragraph in the notebook.

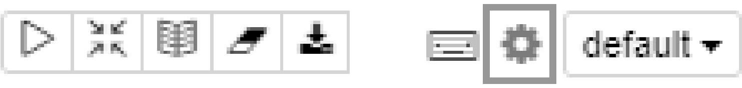
To use these interpreters, you must use these directives at the beginning of the paragraph in a notebook

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

> **Note:**
>
> Do not override Python paragraph interpreter bindings as they will not share the same Python engine backend.

To change the interpreter binding for a specific paragraph in a notebook:

1. Open the notebook and click the gear icon to view the interpreter bindings and its order.

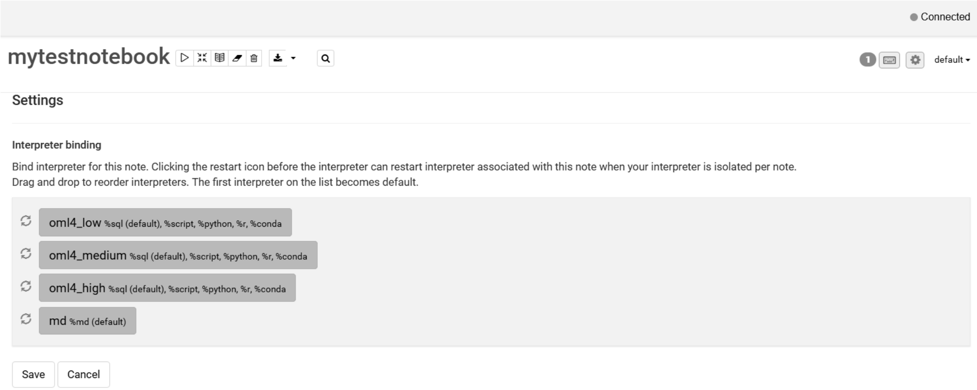
   In this example, all the three SQL interpreters are bound to the notebook, and the interpreter with `low` resource allocation `adwpcwdp_low %sql` is the default, as it is the first interpreter on the list. The Markdown interpreter is not bound to the notebook

   > **Note:**
   >
   > The names of the interpreters are in the format `databasename_low`, `databasename_medium` and `databasename_high` which is the same as the interpreter binding order name.

   In this example, the interpreter names are:

   - adwp_low `% sql(default),%script, %python, %r, %conda`
   - adwp_medium `% sql(default),%script, %python, %r, %conda`
   - adwp_high `% sql(default), %script,%python, %r, %conda`
   - md `%md(default)`

Figure 5-8 Interpreter Bindings



Settings

Interpreter binding

Bind interpreter for this note. Clicking the restart icon before the interpreter can restart interpreter associated with this note when your interpreter is isolated per note.
Drag and drop to reorder interpreters. The first interpreter on the list becomes default.

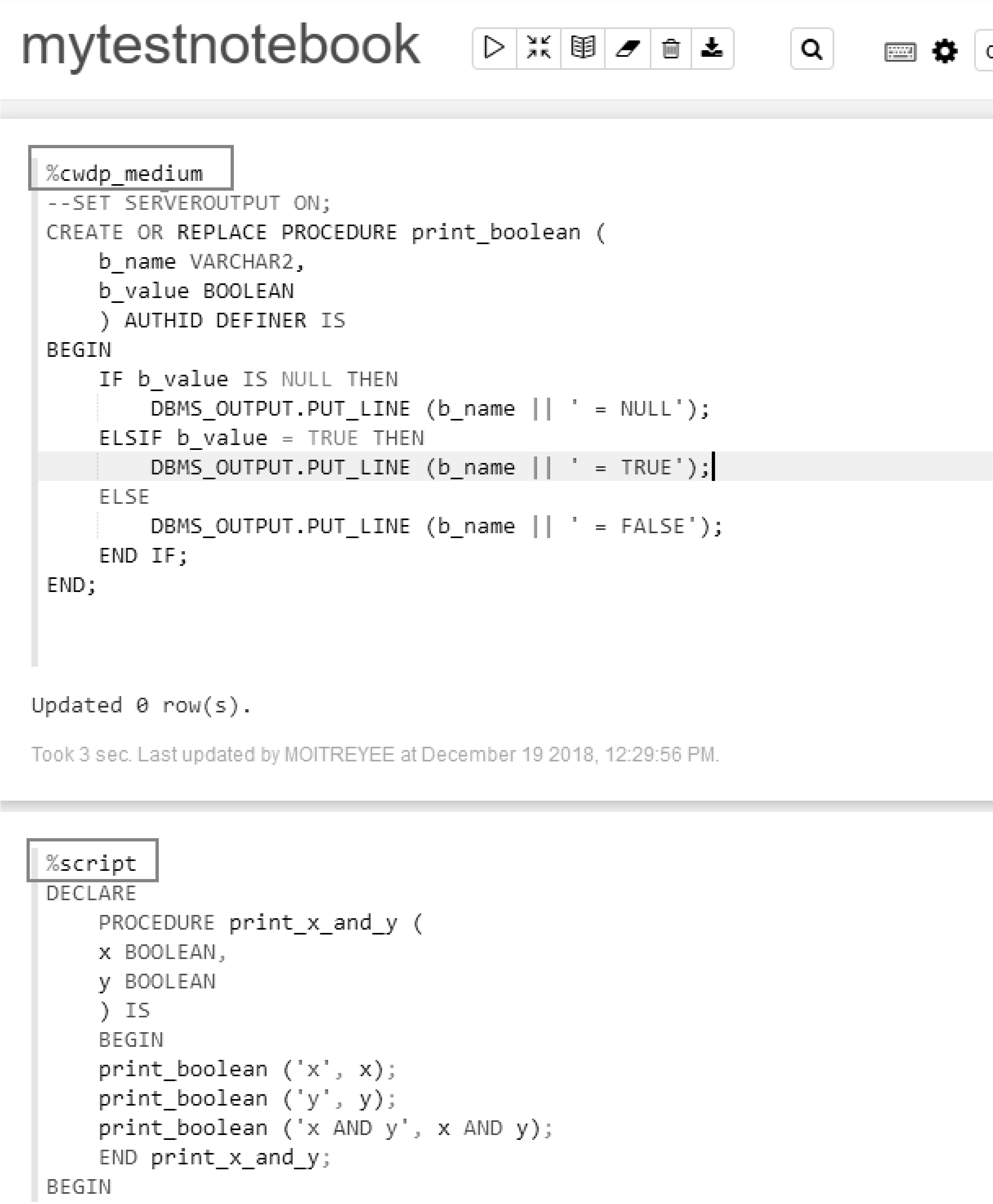
- oml4_low %sql (default), %script, %python, %r, %conda
- oml4_medium %sql (default), %script, %python, %r, %conda
- oml4_high %sql (default), %script, %python, %r, %conda
- md %md (default)

The first Python interpreter in the list is used to run all Python paragraphs in the notebook. For example, if the `low` binding is selected, then all Python paragraphs are run using the `low` binding Python interpreter. All the other paragraphs with SQL and Script interpreter bindings are run using the `low` database name service, that is, `adwp_low`. If any Python scripts run queries against the database, then those queries are run using the `low` database name service, that is, `adwp_low` in this example.

2. To change the interpreter bindings order for a particular paragraph in the notebook:

   - Scroll down to the paragraph for which you want to change the interpreter
   - Call the interpreter with the specific binding
   - Run the paragraph

   For example, call the interpreter with medium resource allocation by typing `%adwp_medium` for the first paragraph in the notebook, and run the paragraph. In this example, `adwp` is the database name.



```
%cwdp_medium
--SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE print_boolean (
    b_name VARCHAR2,
    b_value BOOLEAN
    ) AUTHID DEFINER IS
BEGIN
    IF b_value IS NULL THEN
        DBMS_OUTPUT.PUT_LINE (b_name || ' = NULL');
    ELSIF b_value = TRUE THEN
        DBMS_OUTPUT.PUT_LINE (b_name || ' = TRUE');
    ELSE
        DBMS_OUTPUT.PUT_LINE (b_name || ' = FALSE');
    END IF;
END;
```

```
Updated 0 row(s).
```

Took 3 sec. Last updated by MOITREYEE at December 19 2018, 12:29:56 PM.

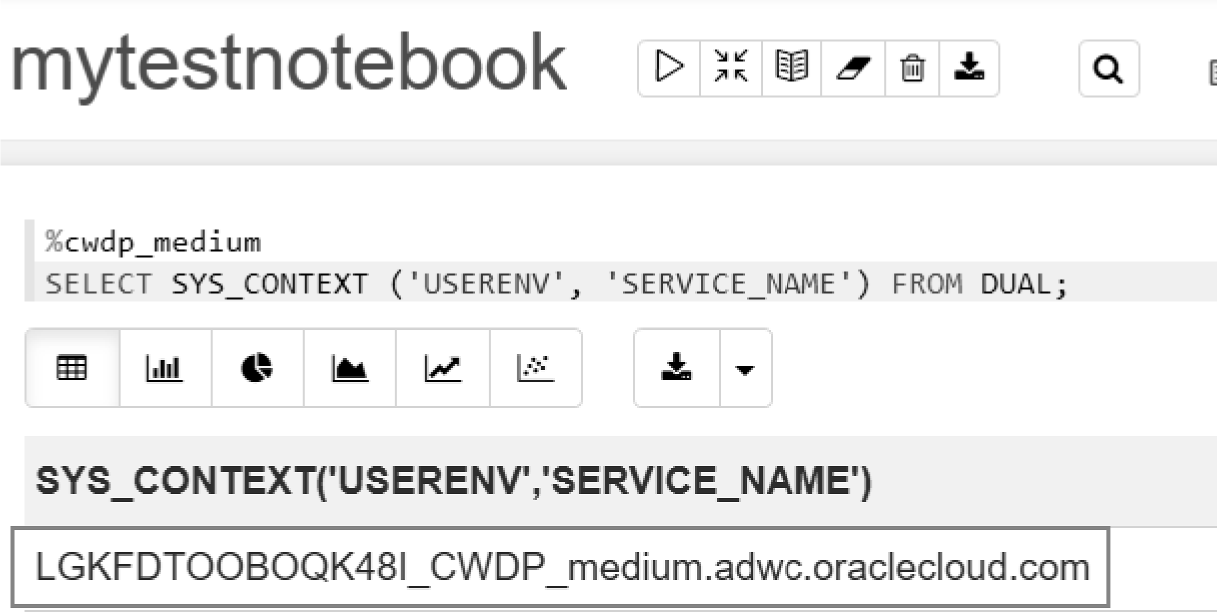
```
%script
DECLARE
    PROCEDURE print_x_and_y (
    x BOOLEAN,
    y BOOLEAN
    ) IS
    BEGIN
    print_boolean ('x', x);
    print_boolean ('y', y);
    print_boolean ('x AND y', x AND y);
    END print_x_and_y;
BEGIN
```

Notice that the first paragraph runs without any error after changing the interpreter binding. The second paragraph in this notebook has the default binding.

3. Validate the interpreter binding for first paragraph of this notebook by typing the SQL statement `SELECT SYS_CONTEXT ('USERENV', 'SERVICE_NAME') FROM DUAL;`

   The SQL statement returns the following information about the interpreter with medium binding:

   ```
   LGKFDTOOBOQK48I_CWDP_medium.adwc.oraclecloud.com
   ```



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

SYS_CONTEXT('USERENV','SERVICE_NAME')
LGKFDTOOBOQK48I_CWDP_medium.adwc.oraclecloud.com

In this example, with reference to the screenshot:

- `LGKFDTOOBOQK48I` is the tenant name
- `CWDP` is the database name
- `medium` is the service name
- `adwc.oraclecloud.com` is the domain

> **Note:**
>
> For the rest of the paragraphs in this notebook, the interpreter binding is the default. You may validate the bindings for each paragraph by running step 3.

This completes the task of changing the interpreter binding for a particular paragraph in the notebook. The rest of the paragraphs in the notebook have the default binding for the interpreter.

5.10.2 Set Interpreter Bindings for Notebooks

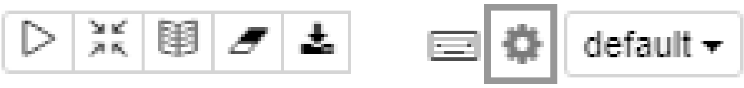
You must bind a notebook to an interpreter to fetch data from the database or any data source. A default set of interpreter bindings is available.

You can set the order of interpreter bindings if you have more than one set available. To set the order of interpreter bindings:

1. On the Notebook page, click the notebook for which you want to set the interpreter bindings.

   The notebook opens in edit mode.

2. Click the gear icon at the top panel.



The Settings pane opens listing the interpreter bindings for the notebook.

3. Drag and drop the interpreters to reorder the interpreter bindings. The first interpreter on the list is the default. The order of interpreter bindings is:
   - Low (Default): 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 execute 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.

   This is the initial binding order of the interpreters. You can change the order of the interpreter bindings.
4. Click **Save.**

Related Topics

- Managing Concurrency and Priorites
- Predefined Database Service Names for Autonomous Data Warehouse

5.10.3 Verify Interpreter Bindings

After setting and changing the order of interpreter bindings, you can verify the interpreter bindings whether you are using SQL, PL/SQL, R or Python interpreter in a given notebook. You use a SQL statement to view and verify the interpreter binding information about your notebook.

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.

To verify the interpreter binding of notebooks:

1. Open the notebook for which you want to check the interpreter binding.
2. Run the following SQL statement:

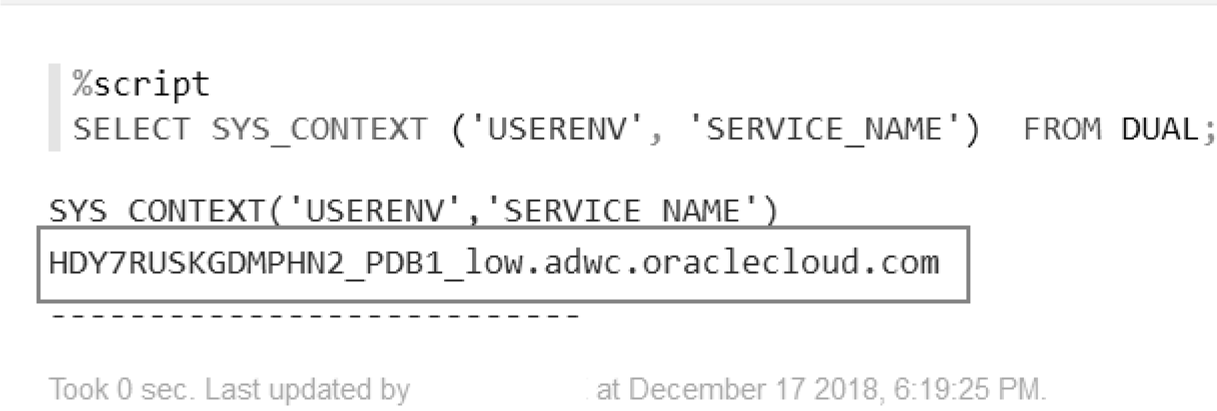
```
%sql
SELECT SYS_CONTEXT ('USERENV', 'SERVICE_NAME') FROM DUAL;
```

The SQL statement provides the name of the service to which a given session is connected.

3. Click **Run.**

   The SQL statement returns the information about the interpreter, the order of the binding, and the service name. The result is displayed in the following format *tenantname__databasename_ servicename.domain*. Here:

   - `HDY7RUSKGDMPHN2` is the tenant name
   - `PDB1` is the database name
   - `low` is the interpreter binding order
   - `adwc.oraclecloud.com` is the domain name



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

```
SYS_CONTEXT('USERENV','SERVICE_NAME')
HDY7RUSKGDMPHN2_PDB1_low.adwc.oraclecloud.com
---------------------------
```

Took 0 sec. Last updated by at December 17 2018, 6:19:25 PM.

6
Get Started with 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.

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.

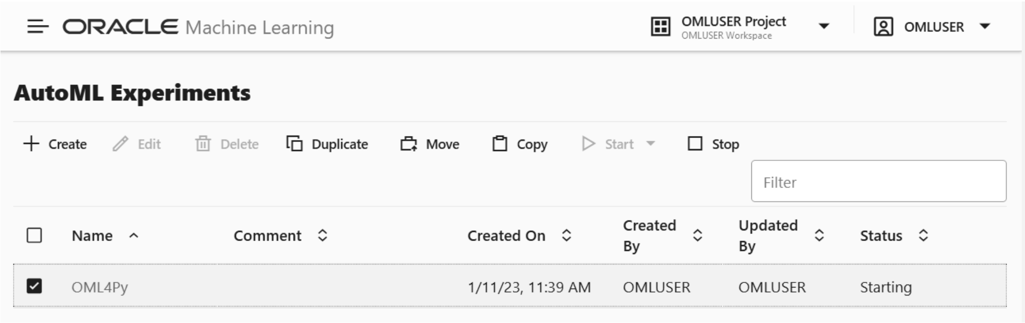
Business users without extensive data science background can use AutoML UI to create and deploy machine learning models. Oracle Machine Learning AutoML UI provides two functional features:

- 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 6-1 Experiments page



AutoML Experiments

□	Name	Comment	Created On	Created By	Updated By	Status
☑	OML4Py		1/11/23, 11:39 AM	OMLUSER	OMLUSER	Starting

- **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 Oracle Machine Learning Notebooks.
- 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

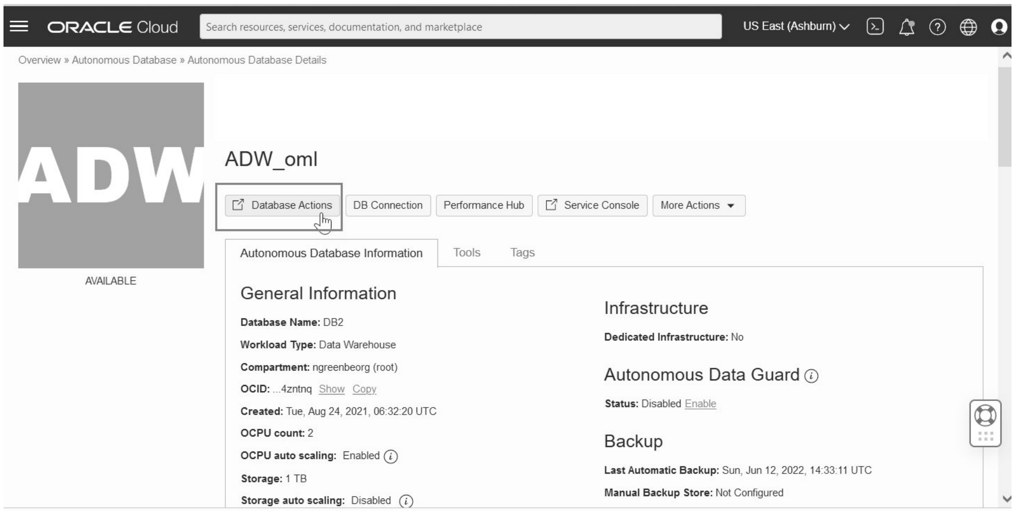
6.1 Access AutoML UI

You can access AutoML UI from Oracle Machine Learning Notebooks.

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

1. To sign in to Oracle Machine Learning Notebooks from the Autonomous Database:
   a. Select an Autonomous Database instance and on the Autonomous Database details page click **Database Actions.**

Figure 6-2 Database Actions



ORACLE Cloud
Search resources, services, documentation, and marketplace
US East (Ashburn)
Overview » Autonomous Database » Autonomous Database Details
ADW
AVAILABLE
ADW_oml
Database Actions
DB Connection
Performance Hub
Service Console
More Actions
Autonomous Database Information
Tools
Tags

General Information

Database Name: DB2

Workload Type: Data Warehouse

Compartment: ngreenbeorg (root)

OCID: ...4zntnq Show Copy

Created: Tue, Aug 24, 2021, 06:32:20 UTC

OCPU count: 2

OCPU auto scaling: Enabled

Storage: 1 TB

Storage auto scaling: Disabled

Infrastructure

Dedicated Infrastructure: No

Autonomous Data Guard

Status: Disabled Enable

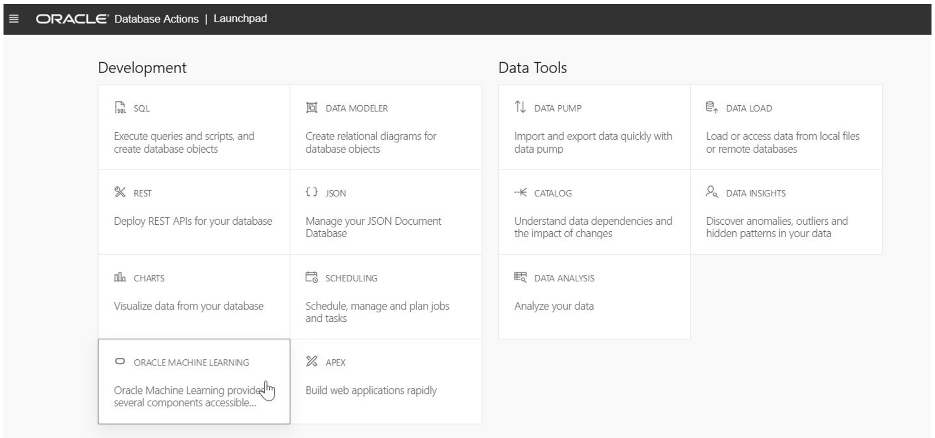
Backup

Last Automatic Backup: Sun, Jun 12, 2022, 14:33:11 UTC

Manual Backup Store: Not Configured

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

Figure 6-3 Oracle Machine Learning


ORACLE Database Actions | Launchpad
Development
Data Tools
SQL
Execute queries and scripts, and create database objects
DATA MODELER
Create relational diagrams for database objects
DATA PUMP
Import and export data quickly with data pump
DATA LOAD
Load or access data from local files or remote databases
REST
Deploy REST APIs for your database
JSON
Manage your JSON Document Database
CATALOG
Understand data dependencies and the impact of changes
DATA INSIGHTS
Discover anomalies, outliers and hidden patterns in your data
CHARTS
Visualize data from your database
SCHEDULING
Schedule, manage and plan jobs and tasks
DATA ANALYSIS
Analyze your data
ORACLE MACHINE LEARNING
Oracle Machine Learning provides several components accessible...
APEX
Build web applications rapidly

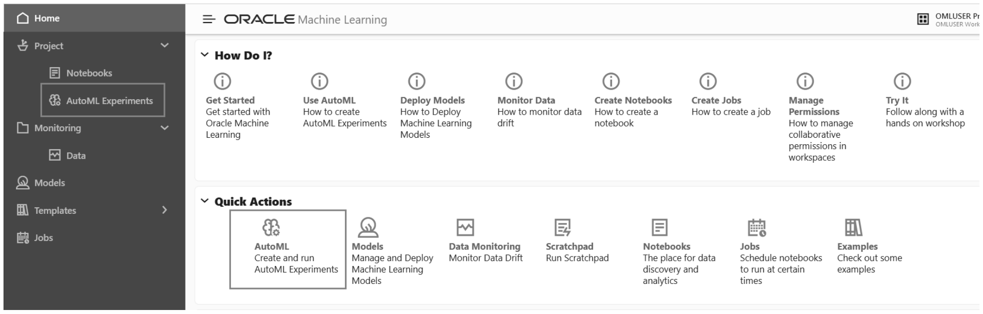
The Oracle Machine Learning sign in page opens.

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

This opens the Oracle Machine Learning Notebooks homepage.

2. On your Oracle Machine Learning Notebooks homepage, click **AutoML.**.

Figure 6-4 AutoML options



Alternatively, you can click the hamburger menu and click **AutoML** under Projects.

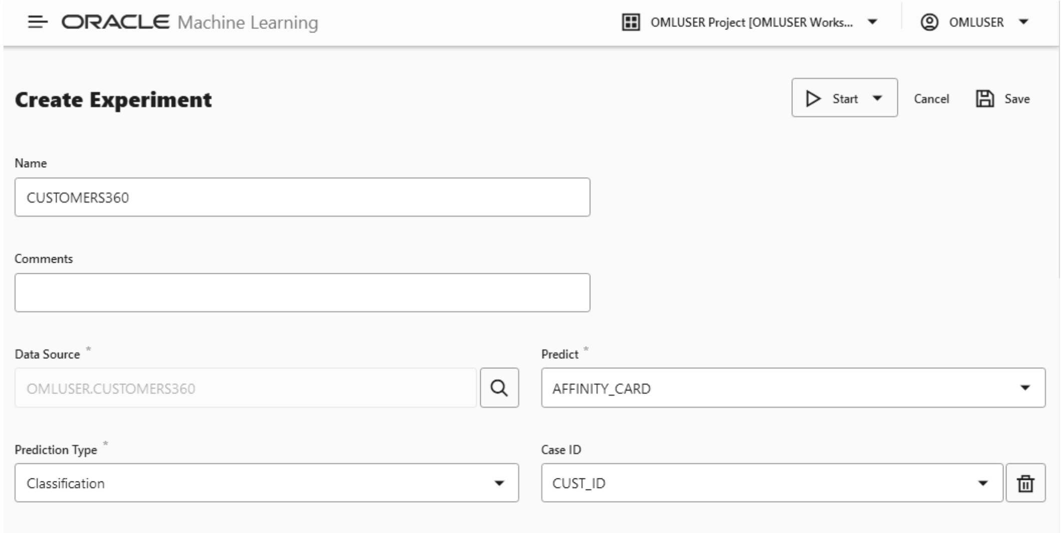
6.2 Create AutoML UI Experiment

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 6-5 Create an AutoML Experiment



Create Experiment

Name: CUSTOMERS360

Comments:

Data Source *: OMLUSER.CUSTOMERS360

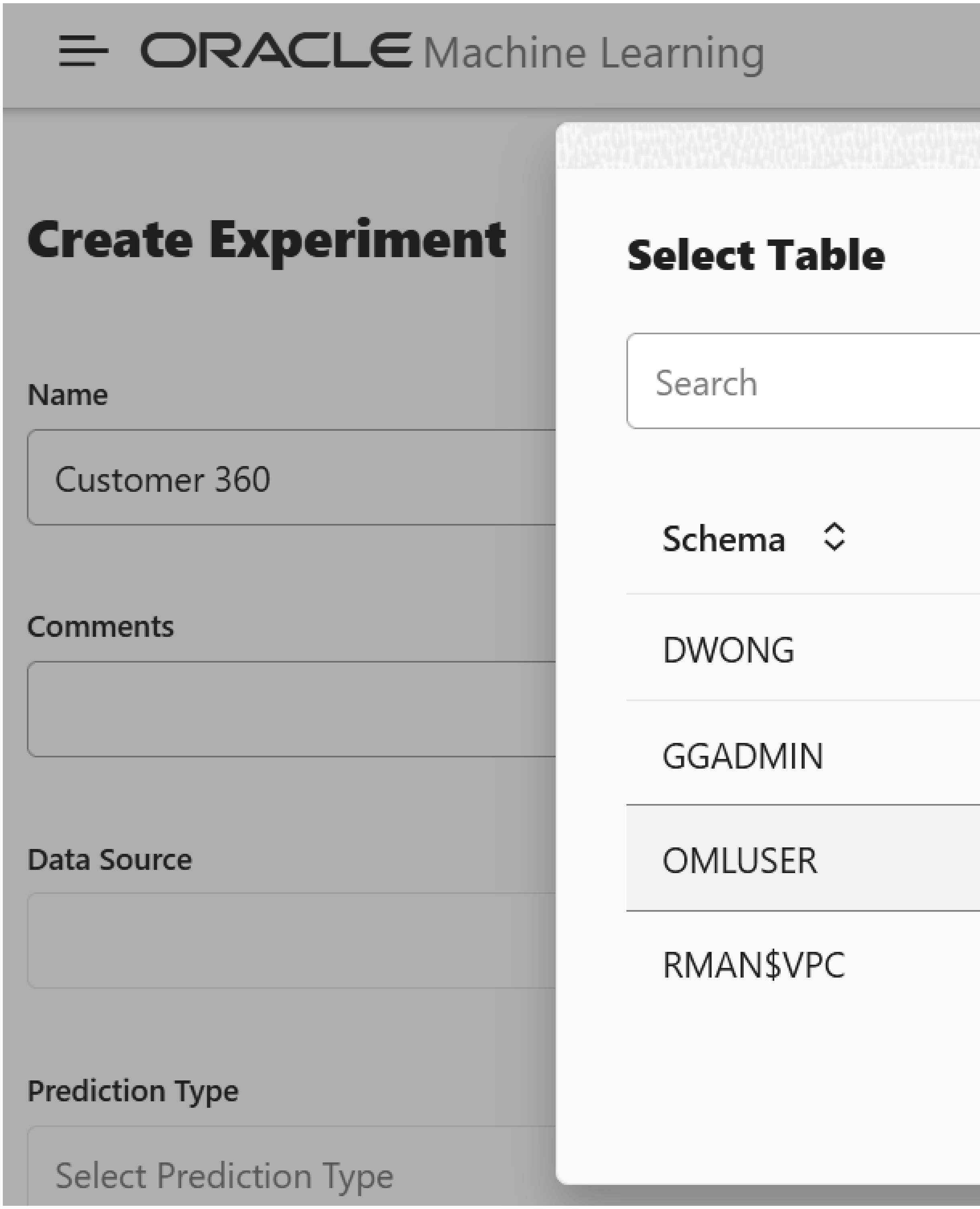
Predict *: AFFINITY_CARD

Prediction Type *: Classification

Case ID: CUST_ID

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 6-6 Select Table dialog



Name

Customer 360

a. In the Schema column, select a schema.

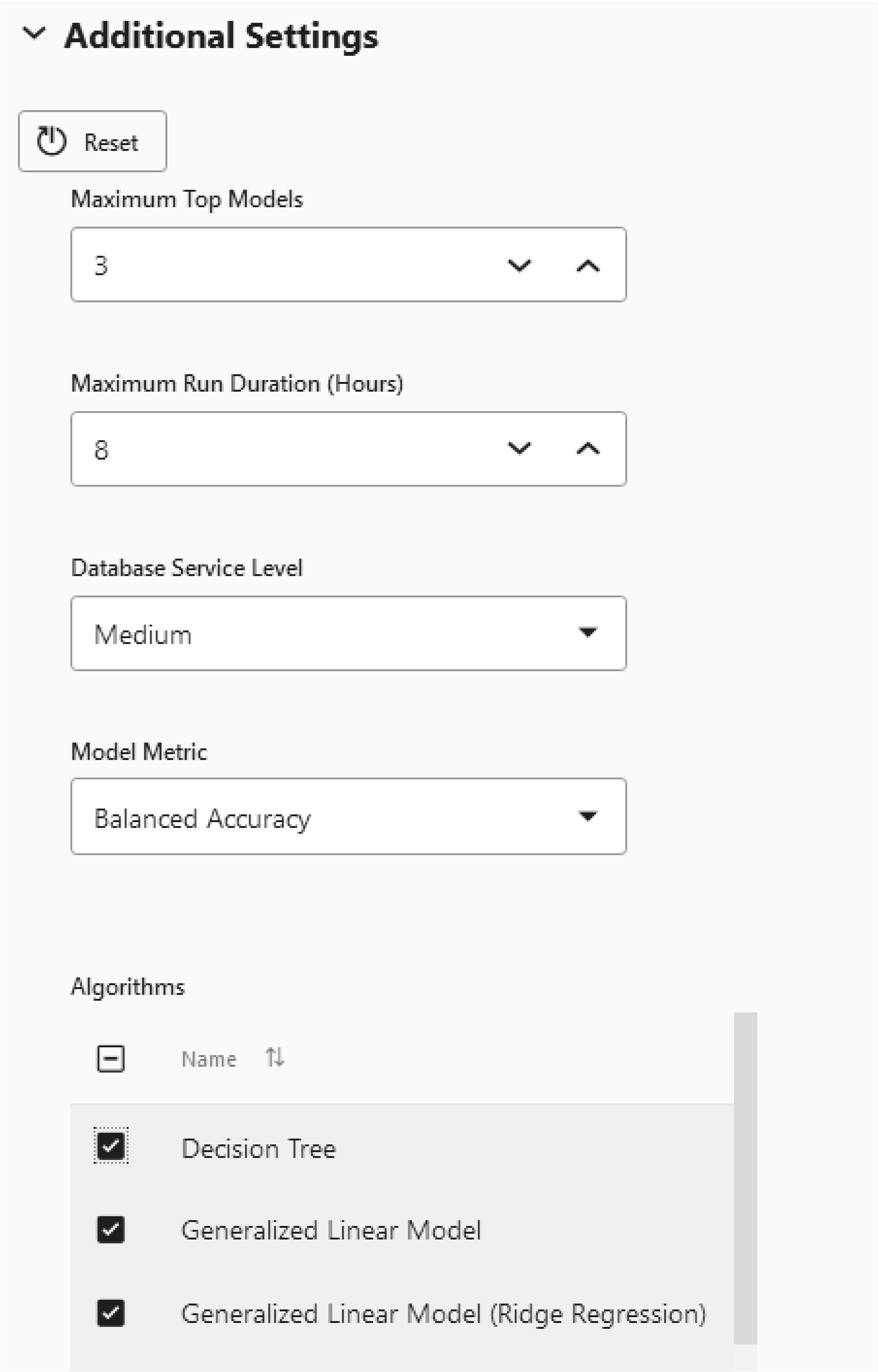
> **Note:**
>
> While you select the data source, statistics are displayed in the Features grid at the bottom of the experiment page. Busy status is indicated until the computation is complete. The target column that you select in Predict is highlighted in the Features grid.

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 6-7 Additional Settings of an AutoML Experiment



Additional Settings

Maximum Top Models: 3

Maximum Run Duration (Hours): 8

Database Service Level: Medium

Model Metric: Balanced Accuracy

Algorithms

⊟	Name
☑	Decision Tree
☑	Generalized Linear Model
☑	Generalized Linear Model (Ridge Regression)

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 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

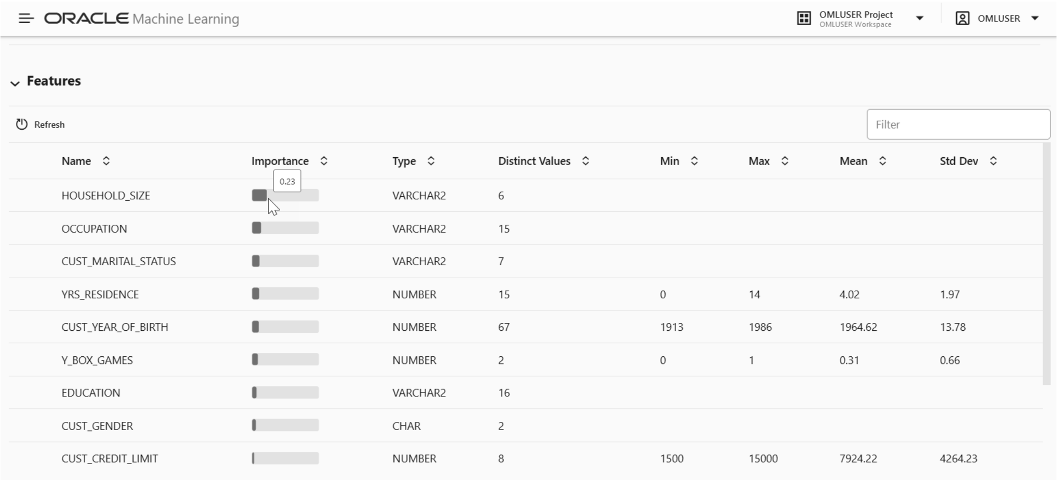
f. **Algorithm:** The supported algorithms depend on **Prediction Type** that you have selected. Click the corresponding checkbox against the algorithms to select it. By default, all the candidate algorithms are selected for consideration as the experiment runs. The supported algorithms for the two Prediction 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.

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 6-8 Features



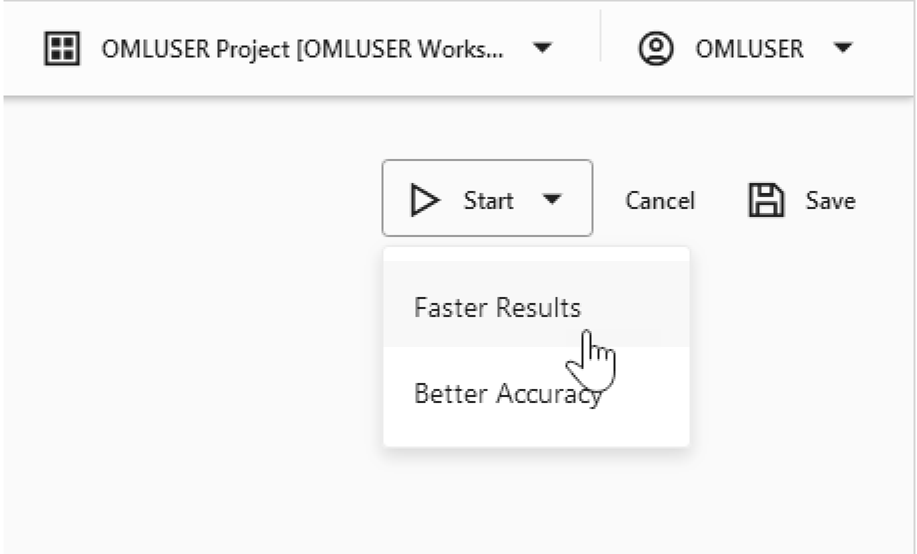
Name	Importance	Type	Distinct Values	Min	Max	Mean	Std Dev
HOUSEHOLD_SIZE		VARCHAR2	6				
OCCUPATION		VARCHAR2	15				
CUST_MARITAL_STATUS		VARCHAR2	7				
YRS_RESIDENCE		NUMBER	15	0	14	4.02	1.97
CUST_YEAR_OF_BIRTH		NUMBER	67	1913	1986	1964.62	13.78
Y_BOX_GAMES		NUMBER	2	0	1	0.31	0.66
EDUCATION		VARCHAR2	16				
CUST_GENDER		CHAR	2				
CUST_CREDIT_LIMIT		NUMBER	8	1500	15000	7924.22	4264.23

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 6-9 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.

6.2.1 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 6-1 Supported Data Types by AutoML Experiments

Data Types	SQL Data Types	Python Data Types
Numerical	NUMBER, INTEGER, FLOAT, BINARY_DOUBLE, NUMBER, BINARY_FLOAT, DM_NESTED_NUMERICALS, DM_NESTED_BINARY_DOUBLES, DM_NESTED_BINARY_FLOATS	INTEGER, FLOAT(NUMBER, BINARY_DOUBLE, BINARY_FLOAT)
Categorical	CHAR, VARCHAR2, DM_NESTED_CATEGORICALS	STRING(VARCHAR2, CHAR, CLOB)
Unstructured Text	CHAR, VARCHAR2, CLOB, BLOB, BFILE	BYTES (RAW, BLOB)

6.3 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.

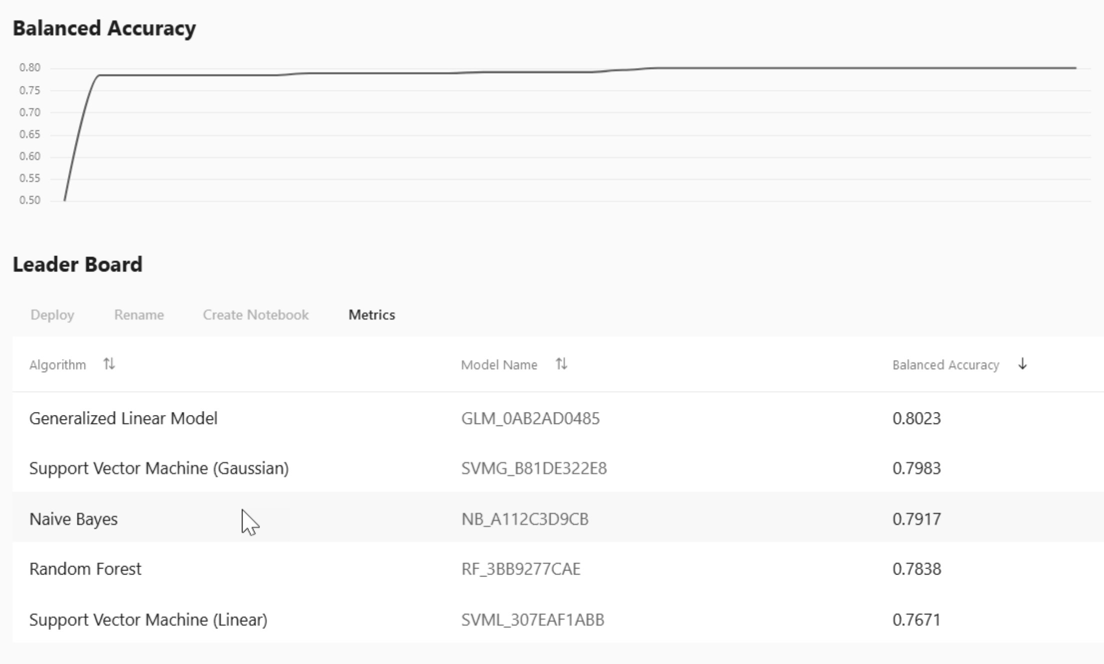
Metric Chart

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 6-10 Leader Board



Algorithm ⇅	Model Name ⇅	Balanced Accuracy ↓
Generalized Linear Model	GLM_0AB2AD0485	0.8023
Support Vector Machine (Gaussian)	SVMG_B81DE322E8	0.7983
Naive Bayes	NB_A112C3D9CB	0.7917
Random Forest	RF_3BB9277CAE	0.7838
Support Vector Machine (Linear)	SVML_307EAF1ABB	0.7671

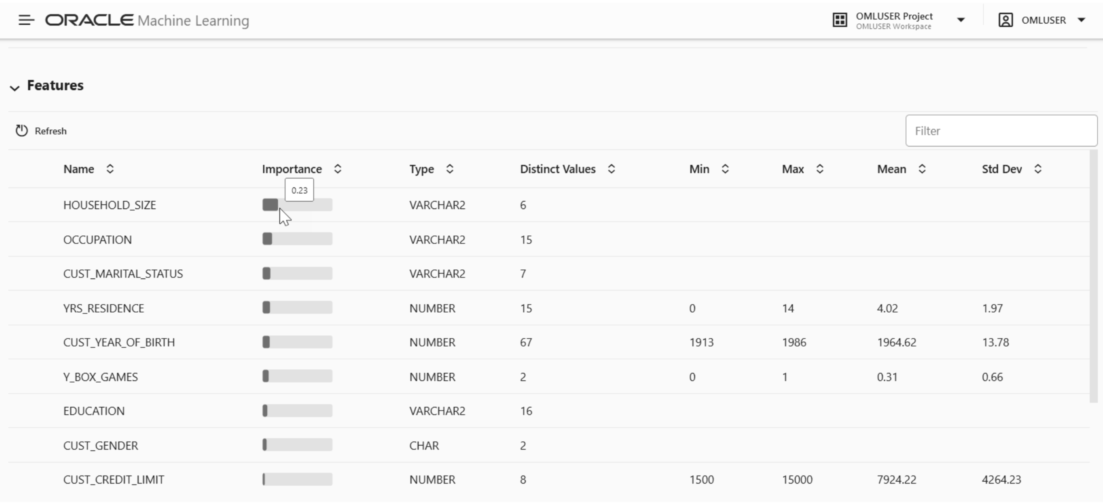
- View Model Details: Click on the **Model Name** to view the details. The model details are displayed in the Model Details dialog box. You can click multiple models on the Leader Board, and view the model details simultaneously. The Model Details window depicts the following:
  - Prediction Impact: 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 6-11 Features



Name	Importance	Type	Distinct Values	Min	Max	Mean	Std Dev
HOUSEHOLD_SIZE	0.23	VARCHAR2	6				
OCCUPATION		VARCHAR2	15				
CUST_MARITAL_STATUS		VARCHAR2	7				
YRS_RESIDENCE		NUMBER	15	0	14	4.02	1.97
CUST_YEAR_OF_BIRTH		NUMBER	67	1913	1986	1964.62	13.78
Y_BOX_GAMES		NUMBER	2	0	1	0.31	0.66
EDUCATION		VARCHAR2	16				
CUST_GENDER		CHAR	2				
CUST_CREDIT_LIMIT		NUMBER	8	1500	15000	7924.22	4264.23

- 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.

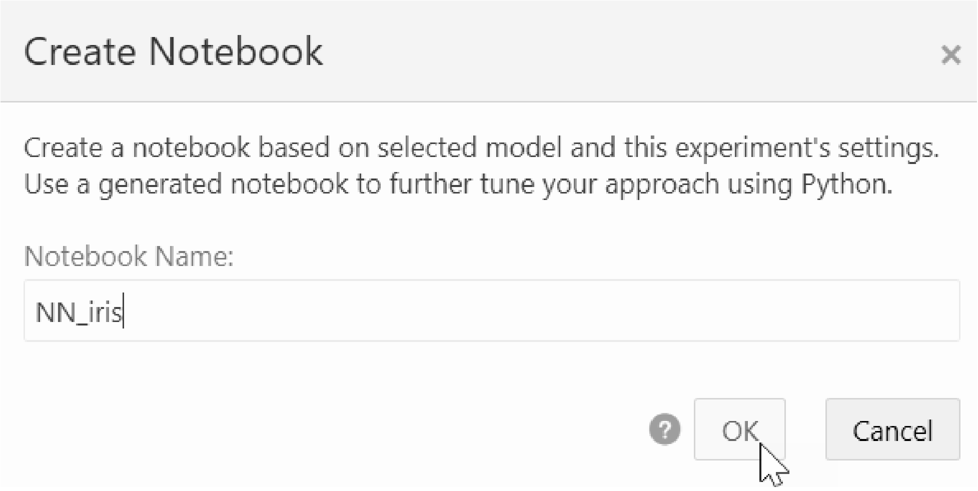
6.3.1 Create Notebooks from AutoML UI Models

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 6-12 Create Notebook



Create Notebook

Create a notebook based on selected model and this experiment's settings. Use a generated notebook to further tune your approach using Python.

Notebook Name:

NN_iris

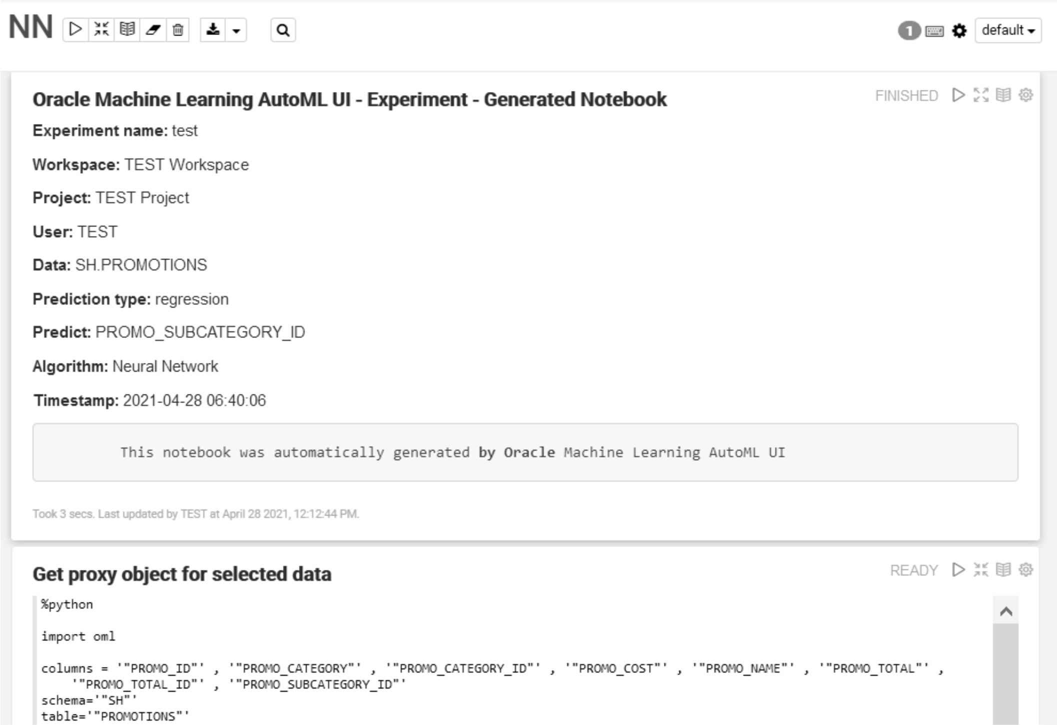
2. In the **Notebook Name** field, enter a name for your notebook.

   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.



Oracle Machine Learning AutoML UI - Experiment - Generated Notebook

Experiment name: test

Workspace: TEST Workspace

Project: TEST Project

User: TEST

Data: SH.PROMOTIONS

Prediction type: regression

Predict: PROMO_SUBCATEGORY_ID

Algorithm: Neural Network

Timestamp: 2021-04-28 06:40:06

```
This notebook was automatically generated by Oracle Machine Learning AutoML UI
```

Took 3 secs. Last updated by TEST at April 28 2021, 12:12:44 PM.

Get proxy object for selected data

```
%python

import oml

columns = '"PROMO_ID"' , '"PROMO_CATEGORY"' , '"PROMO_CATEGORY_ID"' , '"PROMO_COST"' , '"PROMO_NAME"' , '"PROMO_TOTAL"' ,
    '"PROMO_TOTAL_ID"' , '"PROMO_SUBCATEGORY_ID"'
schema='"SH"'
table='"PROMOTIONS"'
```

7

Get Started with Data Monitoring

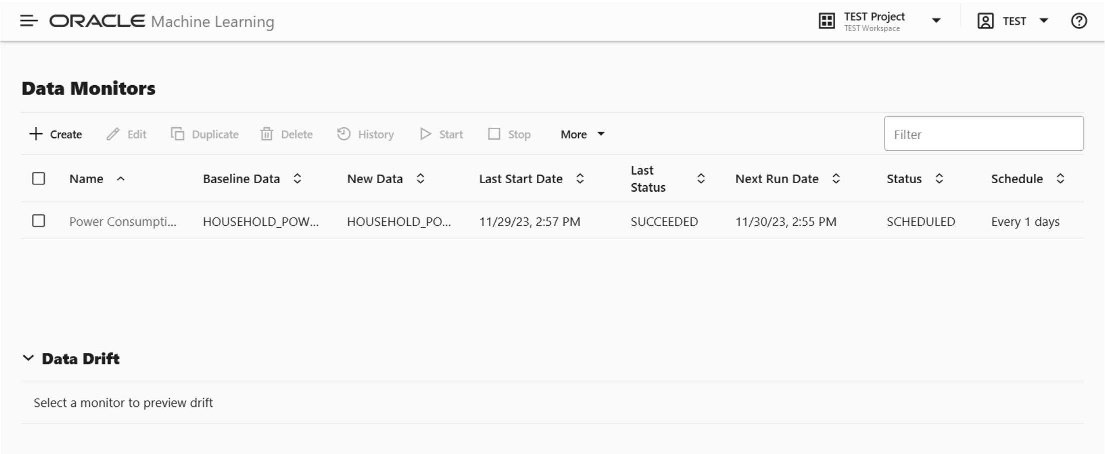
Data Monitoring evaluates how your data evolves over time. It helps you with insights on trends and multivariate dependencies in the data. It also gives you an early warning about data drift.

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 7-1 Data Monitors page



Data Monitors

☐	Name	Baseline Data	New Data	Last Start Date	Last Status	Next Run Date	Status	Schedule
☐	Power Consumpti...	HOUSEHOLD_POW...	HOUSEHOLD_PO...	11/29/23, 2:57 PM	SUCCEEDED	11/30/23, 2:55 PM	SCHEDULED	Every 1 days

Data Drift

Select a monitor to preview drift

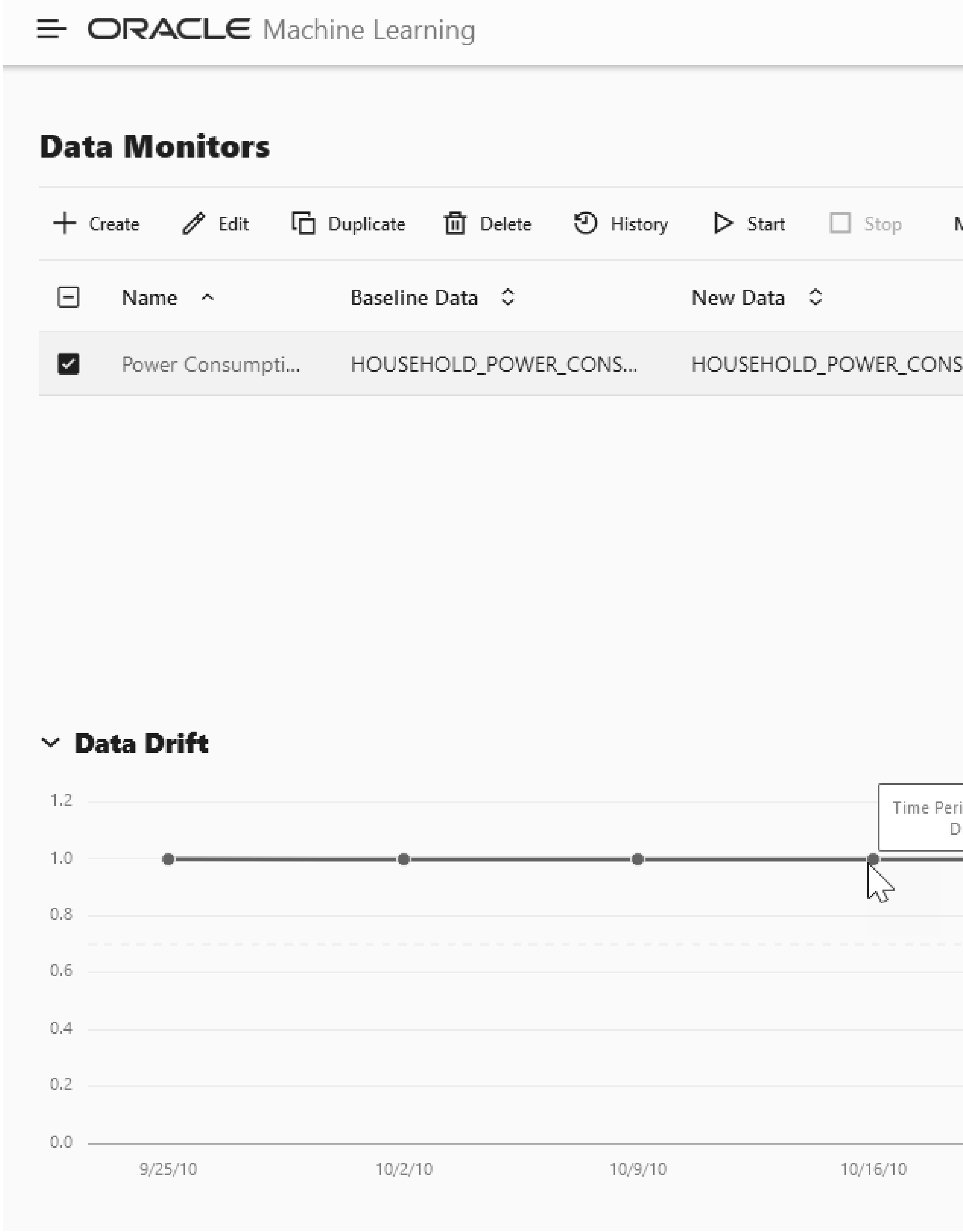
- **Create:** Create a data monitor.

> **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:** Select a data monitor, click **More** and then click:
  - **Enable:** Enable the data monitor schedule. By default, the data monitor is enabled. The status is displayed as `SCHEDULED`.
  - **Disable:** Disable a data monitor schedule. The status is displayed as `DISABLED`.

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 7-2 Data Drift preview on Data Monitors page



☐	Name ^	Baseline Data	New Data
☑	Power Consumpti...	HOUSEHOLD_POWER_CONS...	HOUSEHOLD_POWER_CONS

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.

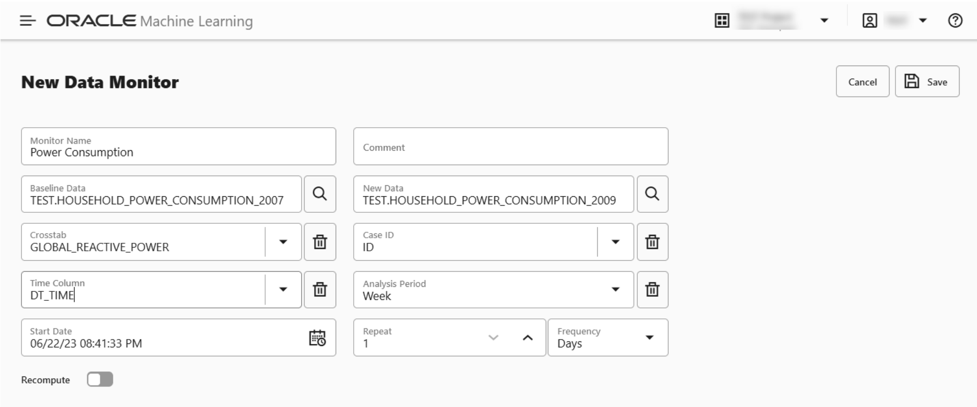
7.1 Create a Data Monitor

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 7-3 New Data Monitor



New Data Monitor

Monitor Name: Power Consumption

Comment

Baseline Data: TEST.HOUSEHOLD_POWER_CONSUMPTION_2007

New Data: TEST.HOUSEHOLD_POWER_CONSUMPTION_2009

Crosstab: GLOBAL_REACTIVE_POWER

Case ID: ID

Time Column: DT_TIME

Analysis Period: Week

Start Date: 06/22/23 08:41:33 PM

Repeat: 1

Frequency: Days

Recompute

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.

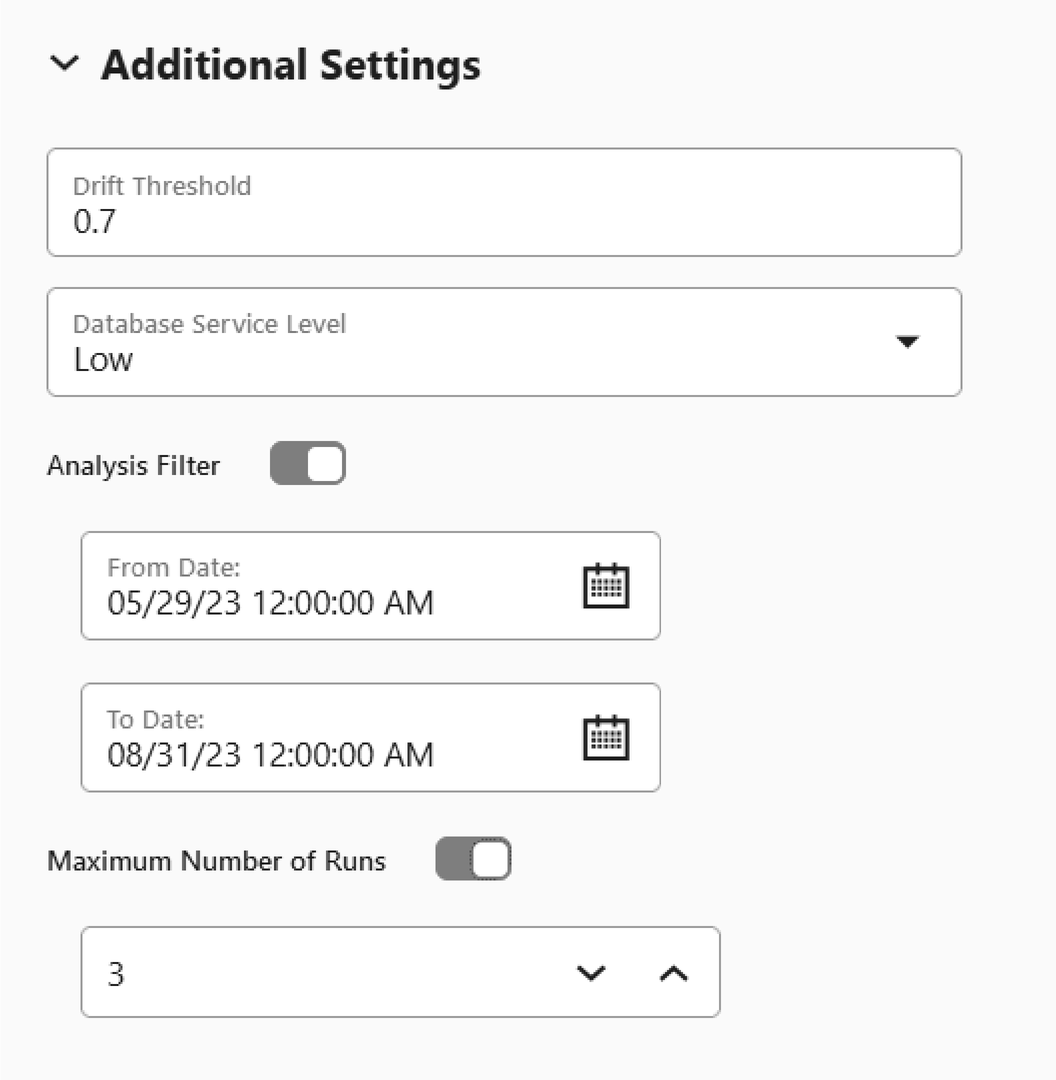
k. **Frequency:** This value determines how frequently the data monitor run will be performed on the New Data. Select a frequency for data monitoring. The options are Minutes, Hours, Days, Weeks, Months. For example, if you select `Minutes` in the **Frequency** field, `2` in the **Repeat** field, and `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 7-4 Data Monitoring Additional Settings



Additional Settings

Drift Threshold
0.7

Database Service Level
Low

Analysis Filter

From Date:
05/29/23 12:00:00 AM

To Date:
08/31/23 12:00:00 AM

Maximum Number of Runs

3

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 Database service levels - `Low`, `Medium`, `High`. 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`.

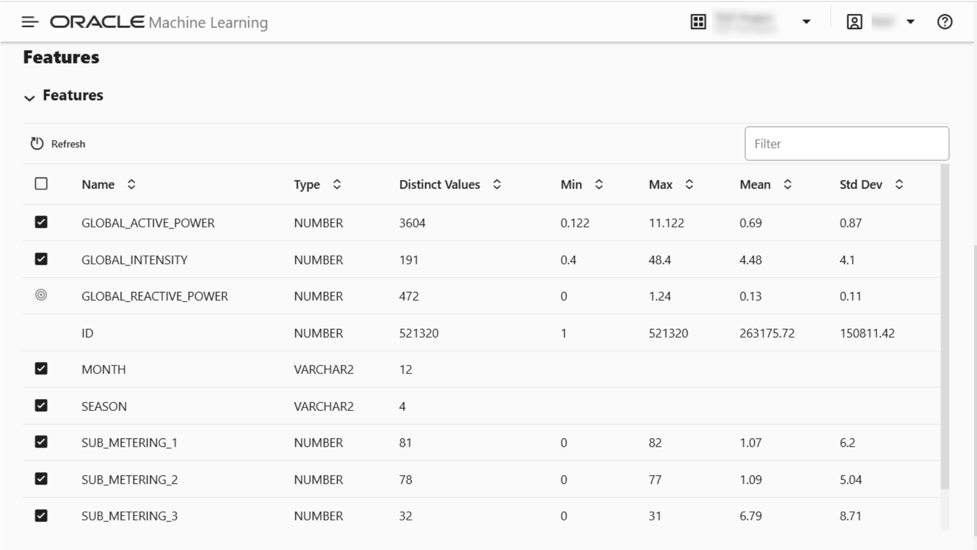
   c. **Analysis Filter:** Enable this option if you want the data monitoring analysis for a specific time period. Move the slider to the right to enable it, and then select a date in **From Date** and **To Date** fields respectively. By default, this field is disabled.

      - **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 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 7-5 Features grid in Data Monitor



□	Name	Type	Distinct Values	Min	Max	Mean	Std Dev
☑	GLOBAL_ACTIVE_POWER	NUMBER	3604	0.122	11.122	0.69	0.87
☑	GLOBAL_INTENSITY	NUMBER	191	0.4	48.4	4.48	4.1
◎	GLOBAL_REACTIVE_POWER	NUMBER	472	0	1.24	0.13	0.11
	ID	NUMBER	521320	1	521320	263175.72	150811.42
☑	MONTH	VARCHAR2	12				
☑	SEASON	VARCHAR2	4				
☑	SUB_METERING_1	NUMBER	81	0	82	1.07	6.2
☑	SUB_METERING_2	NUMBER	78	0	77	1.09	5.04
☑	SUB_METERING_3	NUMBER	32	0	31	6.79	8.71

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 Get Started with Data Monitoring for more information.

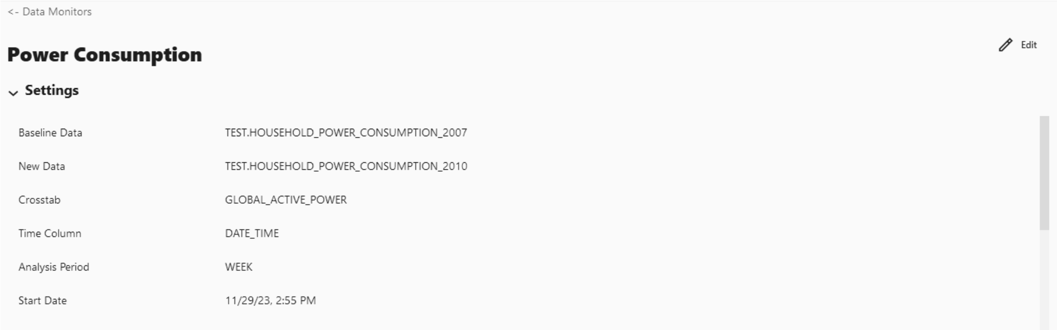
7.2 View Data Monitor Results

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 7-6 Settings section on the Data Monitor Results page



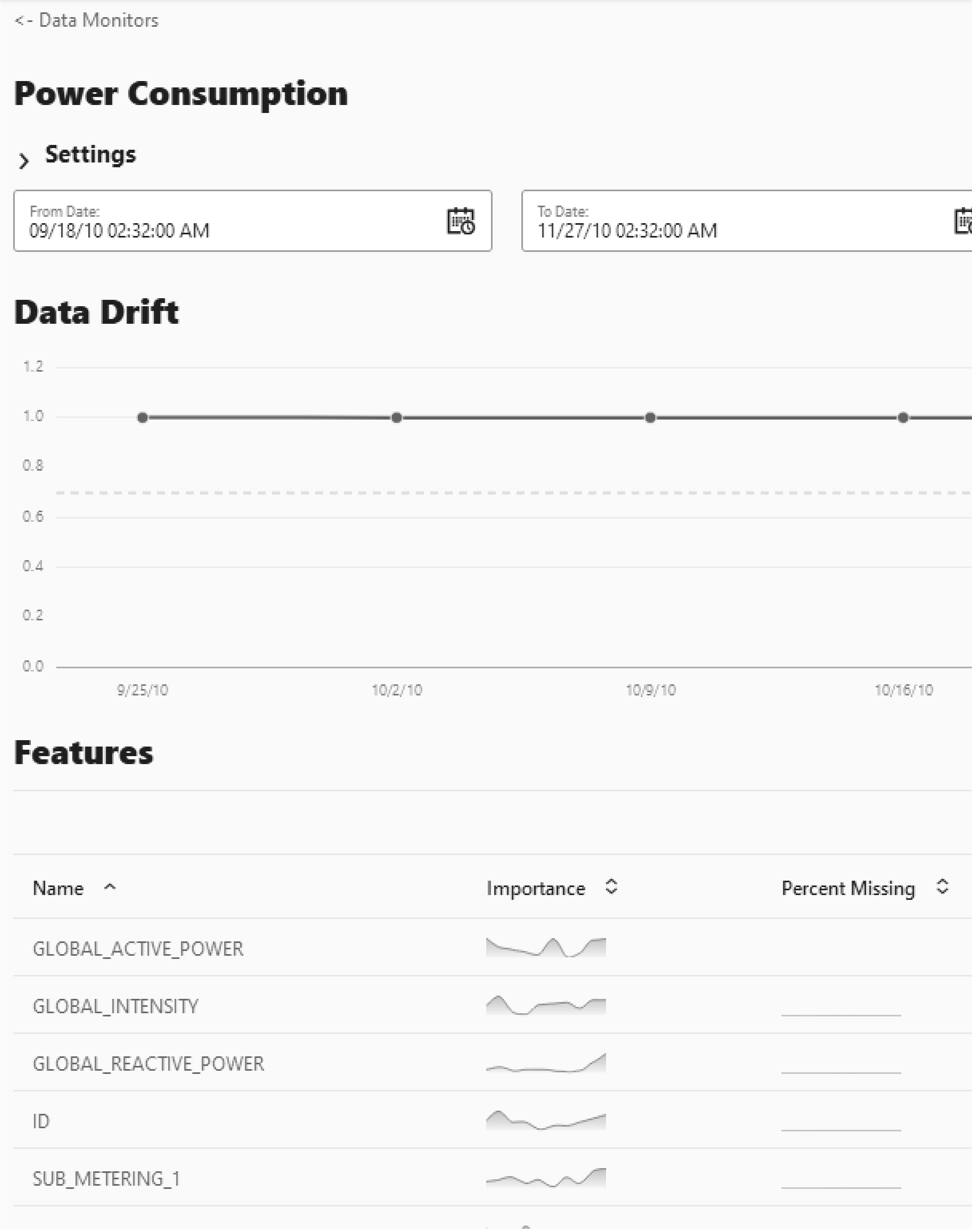
Power Consumption

Settings

Baseline Data	TEST.HOUSEHOLD_POWER_CONSUMPTION_2007
New Data	TEST.HOUSEHOLD_POWER_CONSUMPTION_2010
Crosstab	GLOBAL_ACTIVE_POWER
Time Column	DATE_TIME
Analysis Period	WEEK
Start Date	11/29/23, 2:55 PM

- **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 7-7   Data Drift section on the Data Monitor Results page



Power Consumption

› Settings

From Date: 09/18/10 02:32:00 AM

To Date: 11/27/10 02:32:00 AM

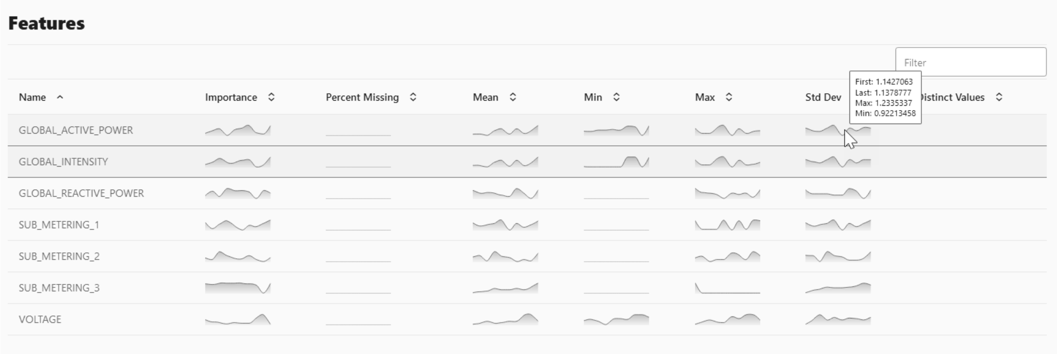
Data Drift

Features

Name ^	Importance	Percent Missing
GLOBAL_ACTIVE_POWER		
GLOBAL_INTENSITY		
GLOBAL_REACTIVE_POWER		
ID		
SUB_METERING_1		

- **Features** — The Features section displays the monitored features along with the computed statistics.

Figure 7-8   Features section on the Data Monitor Results page



First: 1.1427063
Last: 1.1378777
Max: 1.2335337
Min: 0.92213458

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:

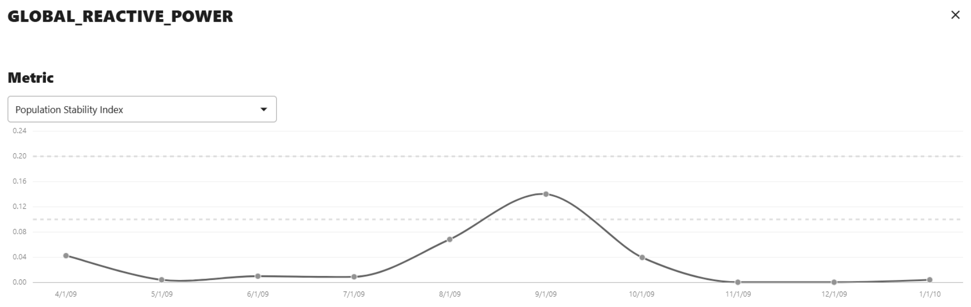
- Number of unique values
- Number of nulls

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 7-9   Population Stability Index



GLOBAL_REACTIVE_POWER

Metric: Population Stability Index

The computations include:

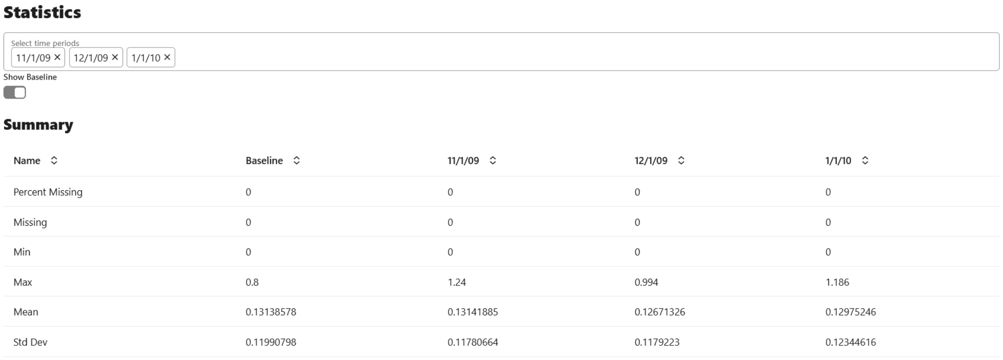
- **Metric:** The following metrics are computed:
  * Population Stability Index (PSI): This is a measure of how much a population has shifted over time or between two different samples of a population in a single number. The two distributions are binned into buckets, and PSI compares the percents of 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 7-10 Statistics



Statistics

Select time periods: 11/1/09 × 12/1/09 × 1/1/10 ×

Show Baseline

Summary

Name	Baseline	11/1/09	12/1/09	1/1/10
Percent Missing	0	0	0	0
Missing	0	0	0	0
Min	0	0	0	0
Max	0.8	1.24	0.994	1.186
Mean	0.13138578	0.13141885	0.12671326	0.12975246
Std Dev	0.11990798	0.11780664	0.1179223	0.12344616

For numerical data, the following statistics are computed:

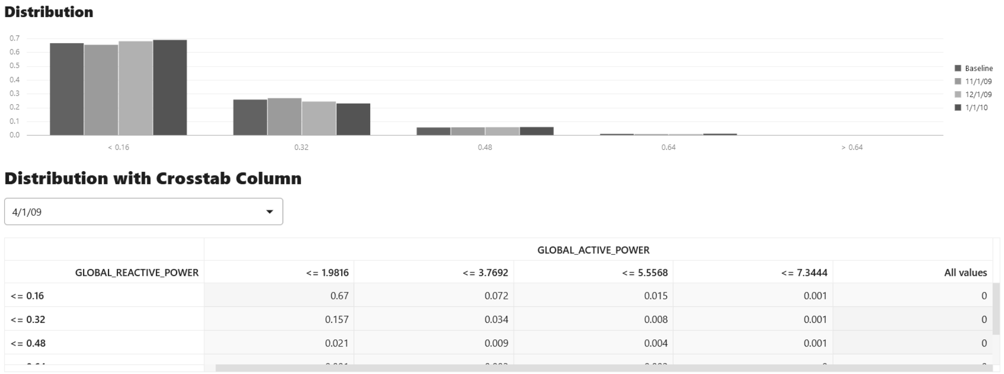
- Mean
- Standard Deviation
- Range (Minimum, Maximum)
- Number of nulls

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 7-11 Distribution Chart and Distribution with Crosstab column



	GLOBAL_ACTIVE_POWER				
GLOBAL_REACTIVE_POWER	<= 1.9816	<= 3.7692	<= 5.5568	<= 7.3444	All values
<= 0.16	0.67	0.072	0.015	0.001	0
<= 0.32	0.157	0.034	0.008	0.001	0
<= 0.48	0.021	0.009	0.004	0.001	0

- **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:

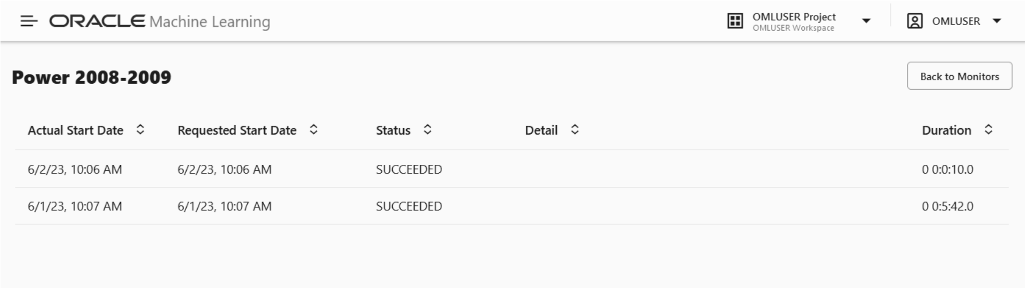
In data drift monitoring, `nulls` are are tracked separately as `number_of_missing_values`.

7.3 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 7-12 Data Monitor History page



Power 2008-2009

Actual Start Date	Requested Start Date	Status	Detail	Duration
6/2/23, 10:06 AM	6/2/23, 10:06 AM	SUCCEEDED		0 0:0:10.0
6/1/23, 10:07 AM	6/1/23, 10:07 AM	SUCCEEDED		0 0:5:42.0

- **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.

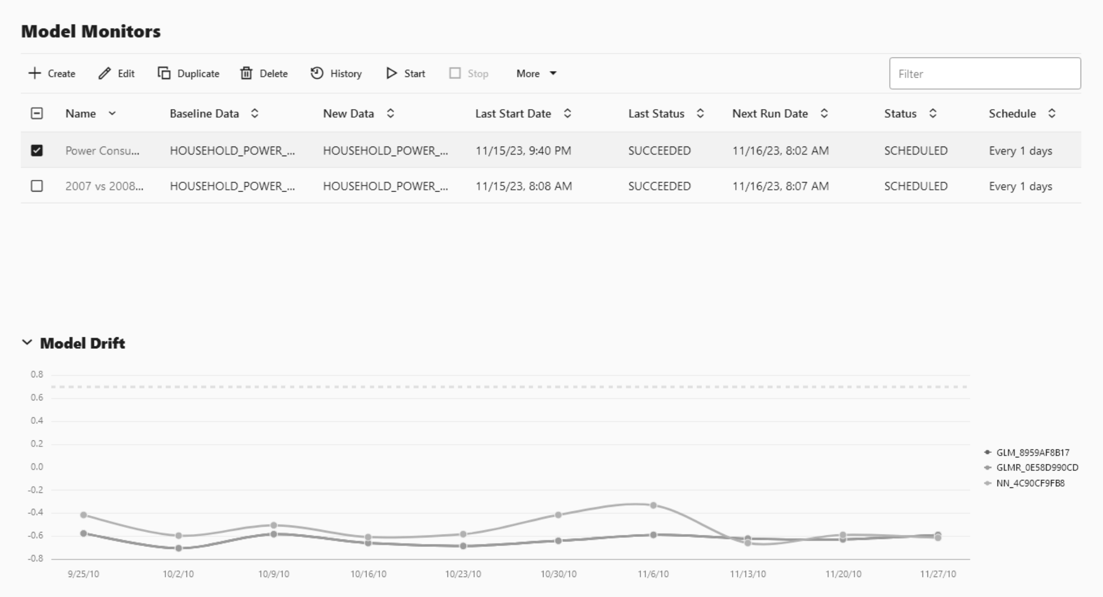
8

Get Started with 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 8-1 Model Monitors page



⊟	Name	Baseline Data	New Data	Last Start Date	Last Status	Next Run Date	Status	Schedule
☑	Power Consu...	HOUSEHOLD_POWER_...	HOUSEHOLD_POWER_...	11/15/23, 9:40 PM	SUCCEEDED	11/16/23, 8:02 AM	SCHEDULED	Every 1 days
□	2007 vs 2008...	HOUSEHOLD_POWER_...	HOUSEHOLD_POWER_...	11/15/23, 8:08 AM	SUCCEEDED	11/16/23, 8:07 AM	SCHEDULED	Every 1 days

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. 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 External Monitors:** Click this option to display the models created and managed by OML Services REST API.
- 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.

8.1 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 from 2007 to 2010. The goal is to understand if and how household consumption has changed over four years. The example shows how to track the effects of data drifts on model predictive accuracy.

The dataset comprises the following columns:

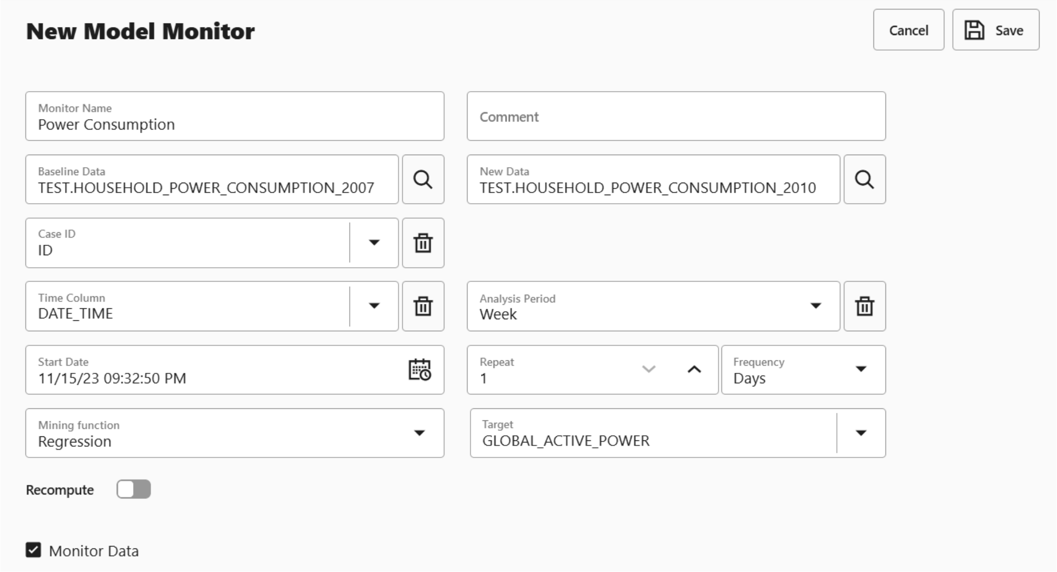
- `DATE_TIME` — Contains the date and time related information in `dd:mm:yyyy:hh:mm:ss` format.
- `GLOBAL_ACTIVE_POWER` — This is the household global minute-averaged active power (in kilowatt).
- `GLOBAL_REACTIVE_POWER` — This is the household global minute-averaged reactive power (in kilowatt).
- `VOLTAGE` — 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 8-2 New Model Monitor page



Monitor Name: Power Consumption
Comment
Baseline Data: TEST.HOUSEHOLD_POWER_CONSUMPTION_2007
New Data: TEST.HOUSEHOLD_POWER_CONSUMPTION_2010
Case ID: ID
Time Column: DATE_TIME
Analysis Period: Week
Start Date: 11/15/23 09:32:50 PM
Repeat: 1
Frequency: Days
Mining function: Regression
Target: GLOBAL_ACTIVE_POWER
Recompute
☑ Monitor Data

   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 2010 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.

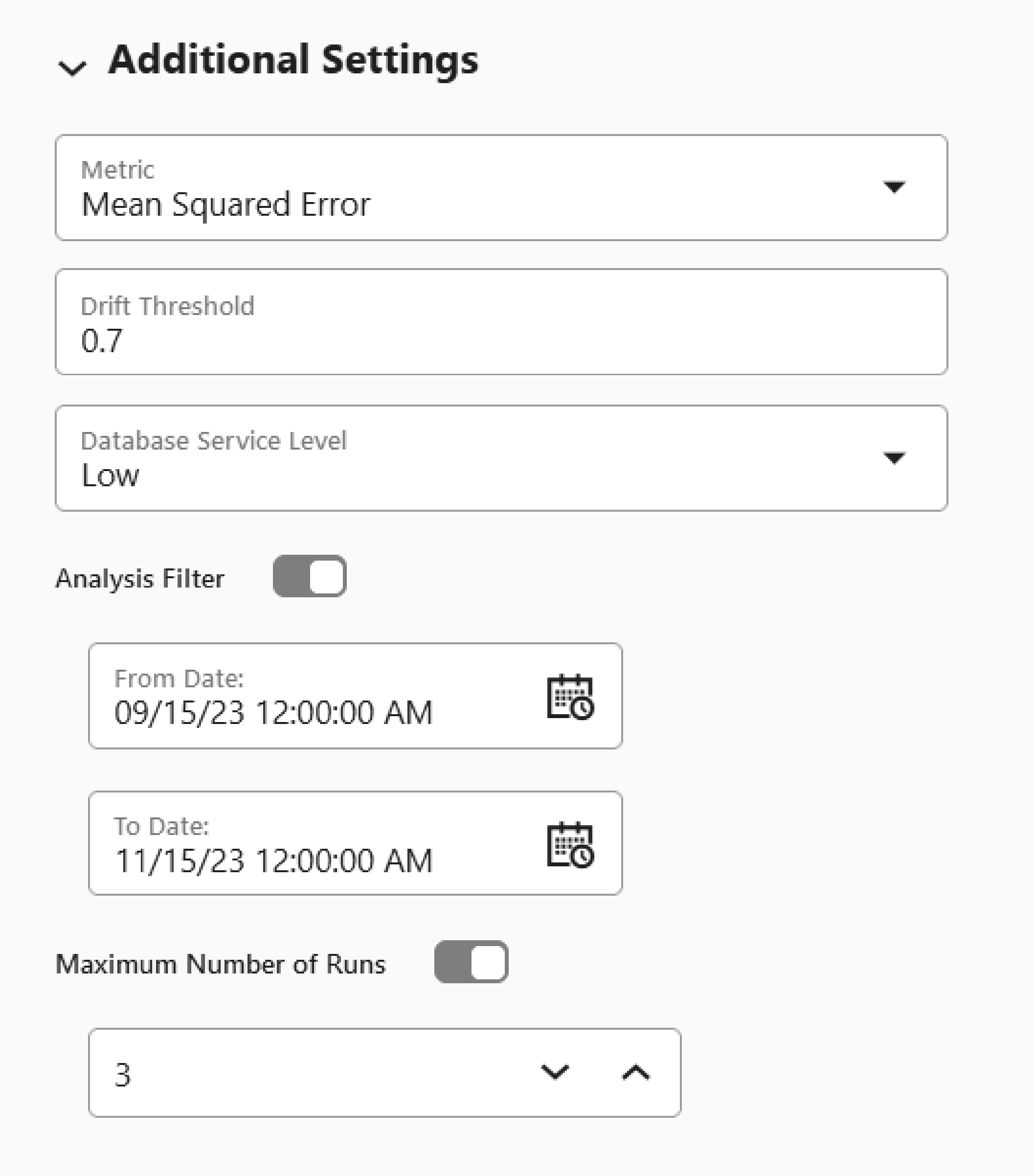
   - When enabled, the drift analysis is performed for the time period specified in the Start Date field and the end time. The analysis will overwrite the already existing results for the specified time period. This means that the analysis will be computed for the time period with new data other than the 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 8-3 Additional Settings section on the New Model Monitor page



Additional Settings

Metric: Mean Squared Error

Drift Threshold: 0.7

Database Service Level: Low

Analysis Filter

From Date: 09/15/23 12:00:00 AM

To Date: 11/15/23 12:00:00 AM

Maximum Number of Runs

3

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, or HIGH.

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.

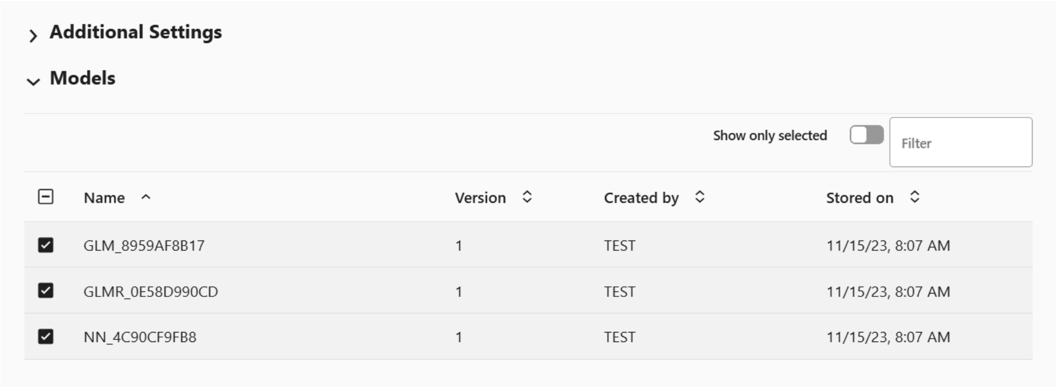
   - **From Date:** This is the start date or timestamp of monitoring in New Data. It assumes the existence of a time column in the table. This is a mandatory field if you use the Analysis Filter option.
   - **To Date:** This is the end date or timestamp of monitoring in the New Data. It assumes the existence 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 8-4 Models section on the New Model Monitor page



⊟	Name ^	Version	Created by	Stored on
☑	GLM_8959AF8B17	1	TEST	11/15/23, 8:07 AM
☑	GLMR_0E58D990CD	1	TEST	11/15/23, 8:07 AM
☑	NN_4C90CF9FB8	1	TEST	11/15/23, 8:07 AM

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.

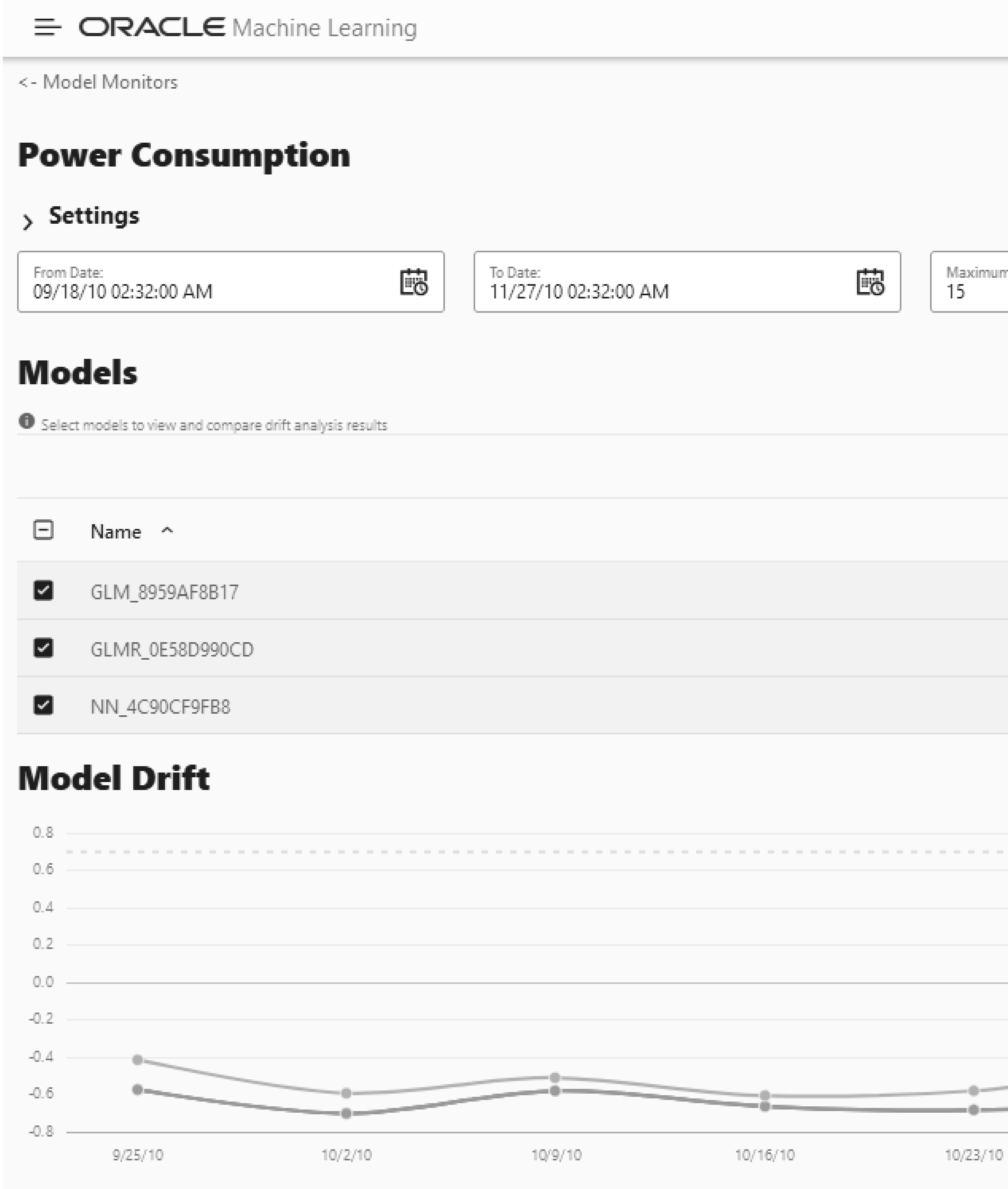
8.2 View Model Monitor Results

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_8959AF817, GLMR_0E58D990CD and NN_4C90CF9FB8 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 8-5   Model Monitor Results page



Power Consumption

› **Settings**

From Date: 09/18/10 02:32:00 AM

To Date: 11/27/10 02:32:00 AM

Maximum 15

Models

Select models to view and compare drift analysis results

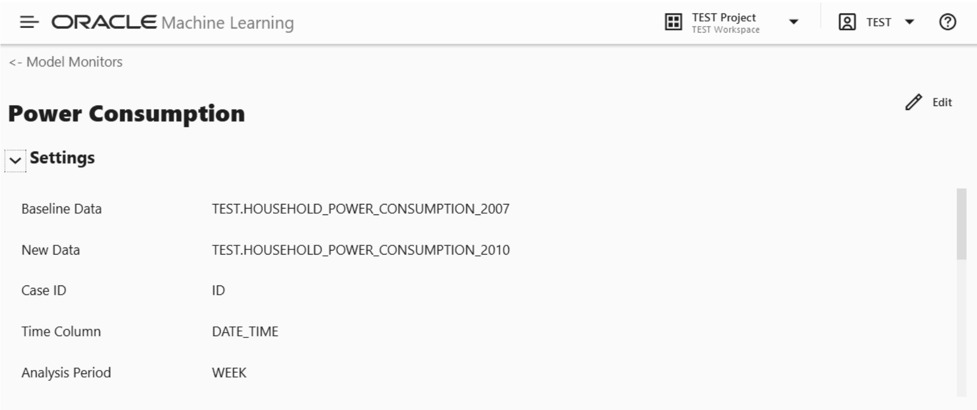
⊟	Name ^
☑	GLM_8959AF8B17
☑	GLMR_0E58D990CD
☑	NN_4C90CF9FB8

Model Drift

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 8-6   Model Monitor Settings



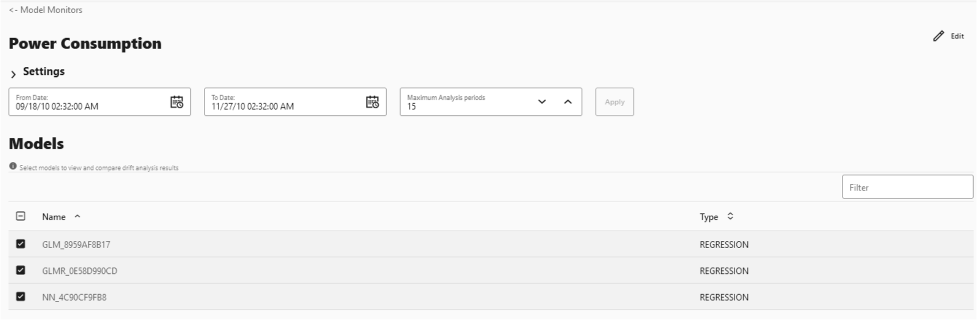
Power Consumption

Settings

Baseline Data	TEST.HOUSEHOLD_POWER_CONSUMPTION_2007
New Data	TEST.HOUSEHOLD_POWER_CONSUMPTION_2010
Case ID	ID
Time Column	DATE_TIME
Analysis Period	WEEK

- **Models** — The Models section lists all the models that are monitored by the monitor. In this example, the models GLM_8959AF817, GLMR_0E58D990CD and NN_4C90CF9FB8 monitored by the **Power Consumption** monitor are listed.

Figure 8-7   Models on the Models Monitor Results page



Power Consumption

From Date: 09/18/10 02:32:00 AM

To Date: 11/27/10 02:32:00 AM

Maximum Analysis periods: 15

Models

Select models to view and compare drift analysis results

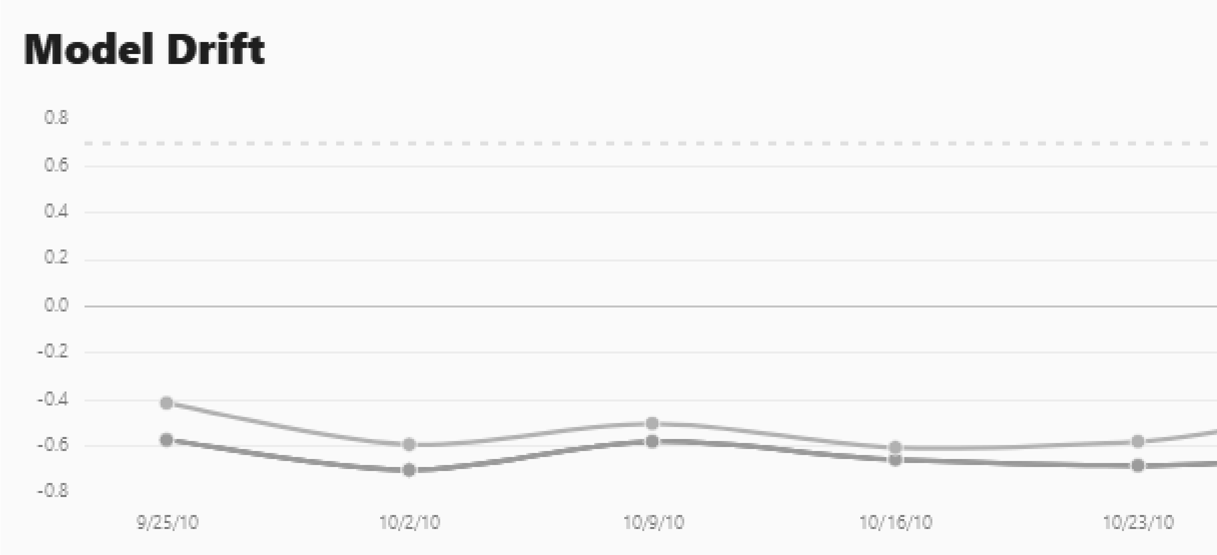
⊟	Name	Type
☑	GLM_8959AF8B17	REGRESSION
☑	GLMR_0E58D990CD	REGRESSION
☑	NN_4C90CF9FB8	REGRESSION

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. The Predictive Impact versus Drift Importance chart is computed only if the **Monitor Data** option is selected while creating the model monitor.

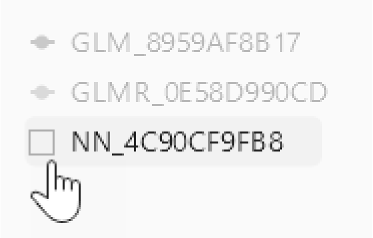
- **Model Drift** — The Model Drift section is displayed just below the Models section. Model drift is the percentage change in the performance metric between the baseline period and the new period. A negative value indicates that the new period has a better performance metric which could happen due 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 8-8 Model Drift on the Models Monitor Results page

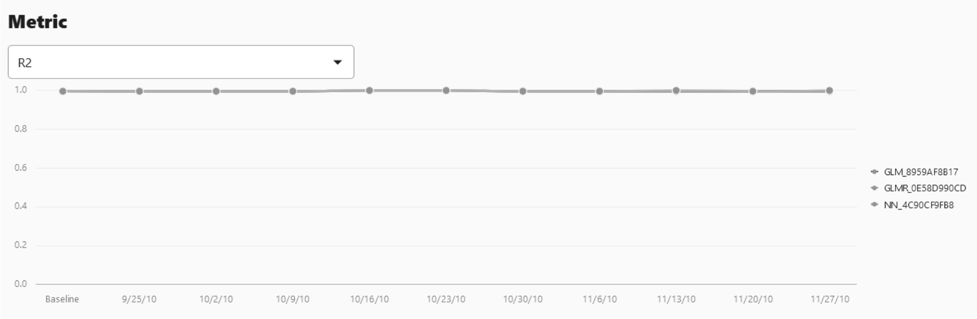


If you want to view the drift details of one model at a time, click on the model name on the right to select or deselect it, as shown here.



- **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 8-9   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.

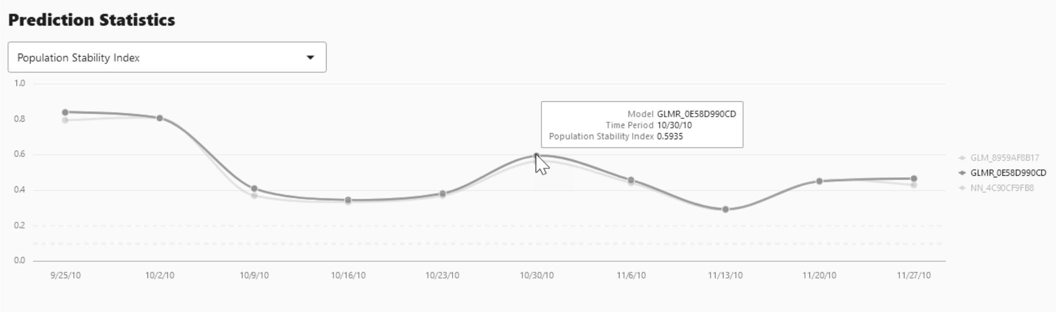
  - F1 Score — Combines precision and recall into a single number. F1-score is computed using harmonic mean which is calculated by the formula:
    `F1-score = 2 × (precision × recall)/(precision + recall)`

  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_0E58D990CD for 10/30/10 is displayed. Hover your cursor on other points on the line to view the computed metric.

  Figure 8-10   Prediction Statistics

  Prediction Statistics

  Population Stability Index ▾

  Model GLMR_0E58D990CD
  Time Period 10/30/10
  Population Stability Index 0.5935

  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

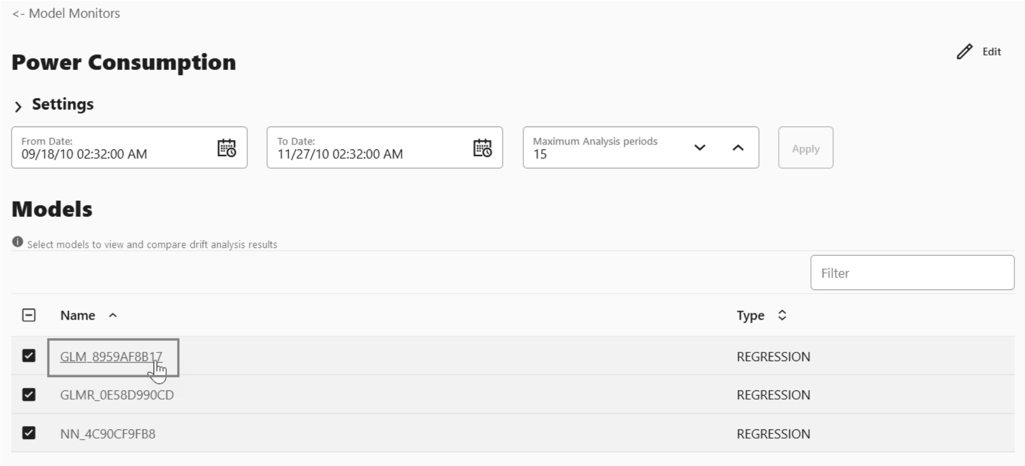
For **Multi-class Classification**, the computed prediction statistics are:

- 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 8-11 Model selection for details



Power Consumption

From Date: 09/18/10 02:32:00 AM

To Date: 11/27/10 02:32:00 AM

Maximum Analysis periods 15

Models

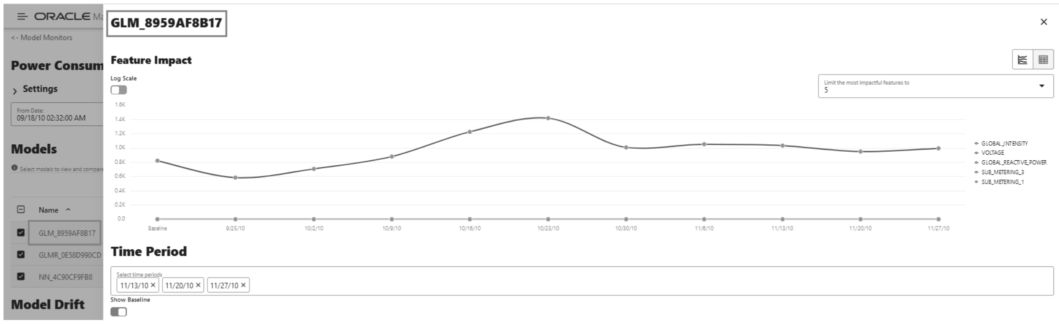
Select models to view and compare drift analysis results

⊟	Name	Type
☑	GLM_8959AF8B17	REGRESSION
☑	GLMR_0E58D990CD	REGRESSION
☑	NN_4C90CF9FB8	REGRESSION

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 8-12 Viewing Feature Impact on a liner scale



From Date: 09/18/10 02:32:00 AM

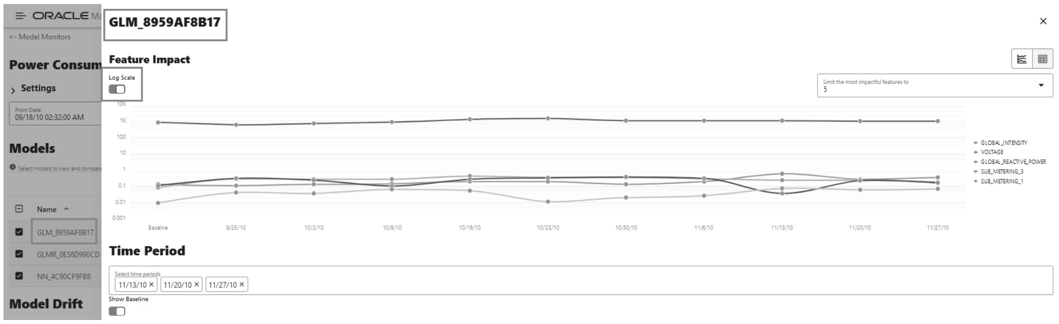
☑ GLM_8959AF8B17
☑ GLMR_0E58D990CD
☑ NN_4C90CF9FB8

Limit the most impactful features to: 5

Select time periods: 11/13/10 × 11/20/10 × 11/27/10 ×

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 8-13 Viewing Feature Impact on a Logarithmic Scale



From Date: 09/18/10 02:32:00 AM

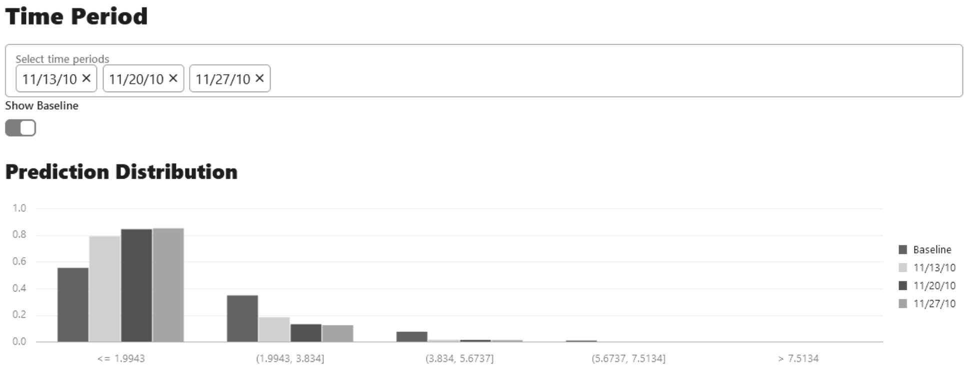
☑ GLM_8959AF8B17
☑ GLMR_0E58D990CD
☑ NN_4C90CF9FB8

Limit the most impactful features to: 5

Select time periods: 11/13/10 × 11/20/10 × 11/27/10 ×

- **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 8-14 Prediction Distribution



Time Period

Select time periods: 11/13/10 × 11/20/10 × 11/27/10 ×

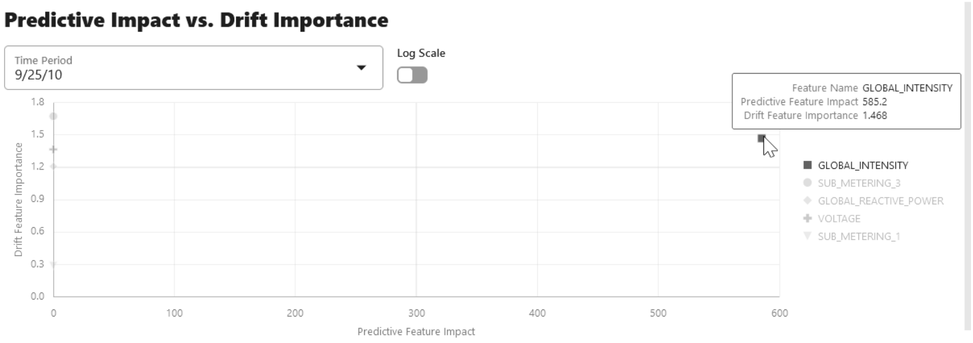
Show Baseline

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 8-15 Predictive Feature Impact versus Drift Importance



Predictive Impact vs. Drift Importance

Time Period: 9/25/10

Log Scale

Feature Name GLOBAL_INTENSITY
Predictive Feature Impact 585.2
Drift Feature Importance 1.468

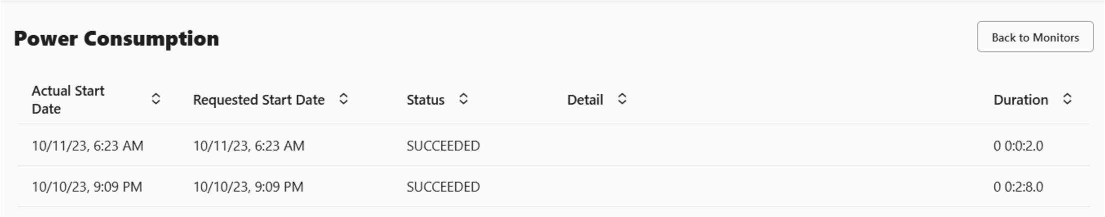
In this screenshot, you can see that the feature GLOBAL_INTENSITY has the maximum impact on the selected predictive model GLM_8959AF817 as compared to the other features - SUB_METERING_3, GLOBAL_REACTIVE_POWER, VOLTAGE, and SUB-METERING_1.

8.3 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 8-16   View Model History



Power Consumption

Actual Start Date	Requested Start Date	Status	Detail	Duration
10/11/23, 6:23 AM	10/11/23, 6:23 AM	SUCCEEDED		0 0:0:2.0
10/10/23, 9:09 PM	10/10/23, 9:09 PM	SUCCEEDED		0 0:2:8.0

- **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.

9

Get Started with 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.

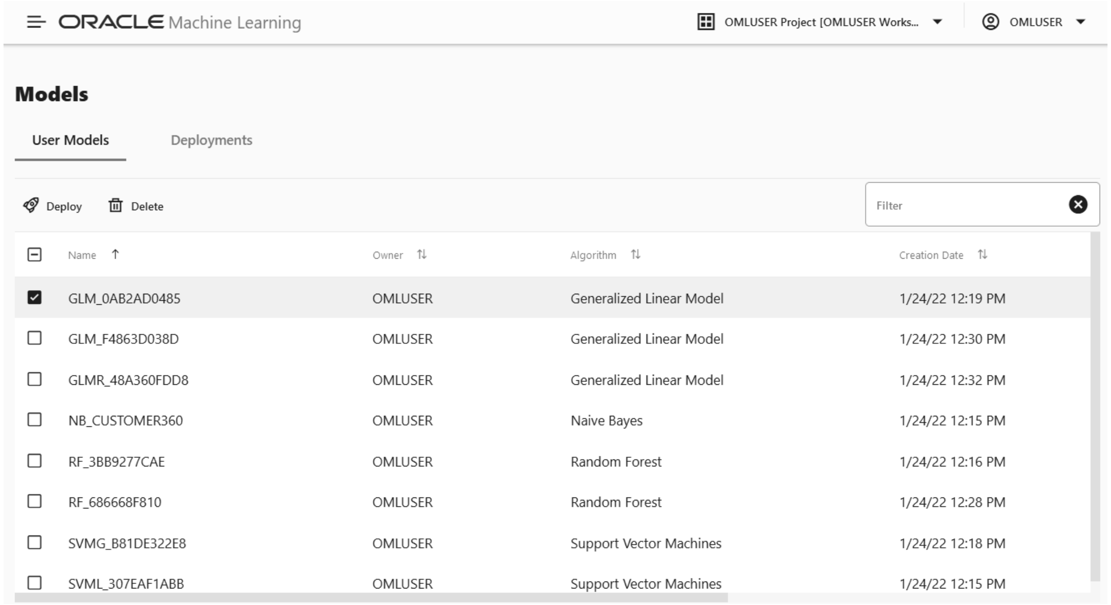
Under Models, the model information and model deployment are available under:

- **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 9-1 User Models



⊟	Name	Owner	Algorithm	Creation Date
☑	GLM_0AB2AD0485	OMLUSER	Generalized Linear Model	1/24/22 12:19 PM
□	GLM_F4863D038D	OMLUSER	Generalized Linear Model	1/24/22 12:30 PM
□	GLMR_48A360FDD8	OMLUSER	Generalized Linear Model	1/24/22 12:32 PM
□	NB_CUSTOMER360	OMLUSER	Naive Bayes	1/24/22 12:15 PM
□	RF_3BB9277CAE	OMLUSER	Random Forest	1/24/22 12:16 PM
□	RF_686668F810	OMLUSER	Random Forest	1/24/22 12:28 PM
□	SVMG_B81DE322E8	OMLUSER	Support Vector Machines	1/24/22 12:18 PM
□	SVML_307EAF1ABB	OMLUSER	Support Vector Machines	1/24/22 12:15 PM

- 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:

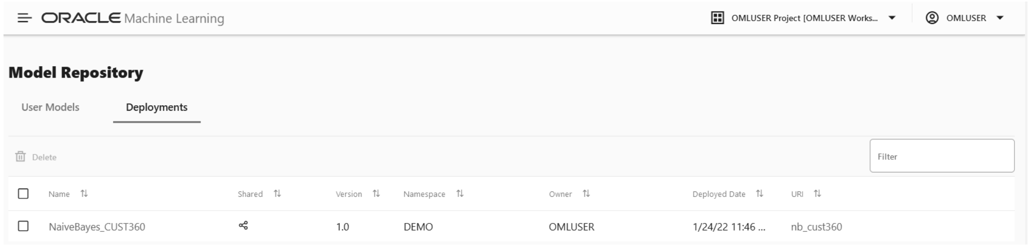
- **Deploy:** To deploy a model, select the model and click **Deploy.**
- **Delete:** To delete a model, select the model and click **Delete.**

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 9-2 Deployed Models



Model Repository

□	Name	Shared	Version	Namespace	Owner	Deployed Date	URI
□	NaiveBayes_CUST360		1.0	DEMO	OMLUSER	1/24/22 11:46 ...	nb_cust360

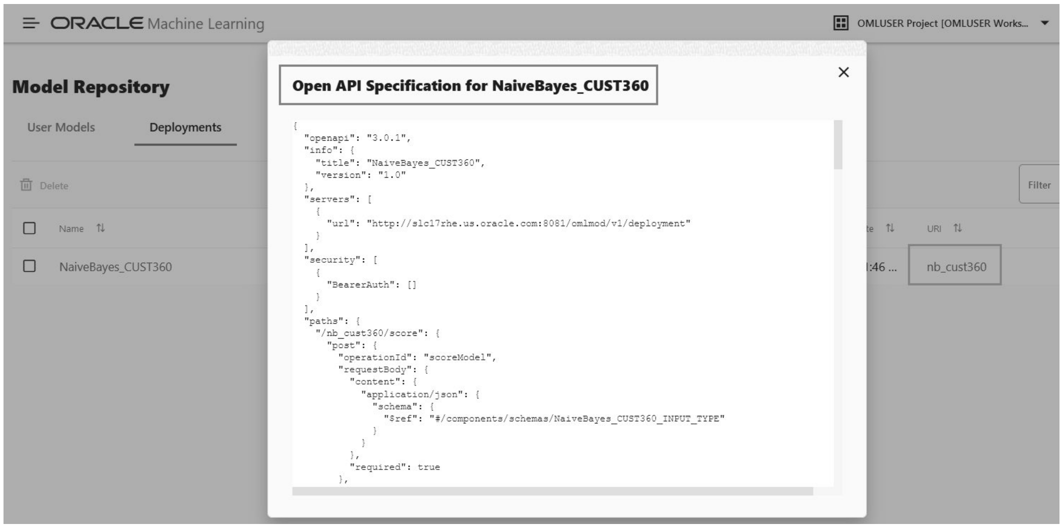
The following information are displayed for each deployed model:

- **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 9-3 REST API Specifications of a Deployed Model



Open API Specification for NaiveBayes_CUST360

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "NaiveBayes_CUST360",
    "version": "1.0"
  },
  "servers": [
    {
      "url": "http://slc17rhe.us.oracle.com:8081/omlmod/v1/deployment"
    }
  ],
  "security": [
    {
      "BearerAuth": []
    }
  ],
  "paths": {
    "/nb_cust360/score": {
      "post": {
        "operationId": "scoreModel",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/NaiveBayes_CUST360_INPUT_TYPE"
              }
            }
          },
          "required": true
        },
```

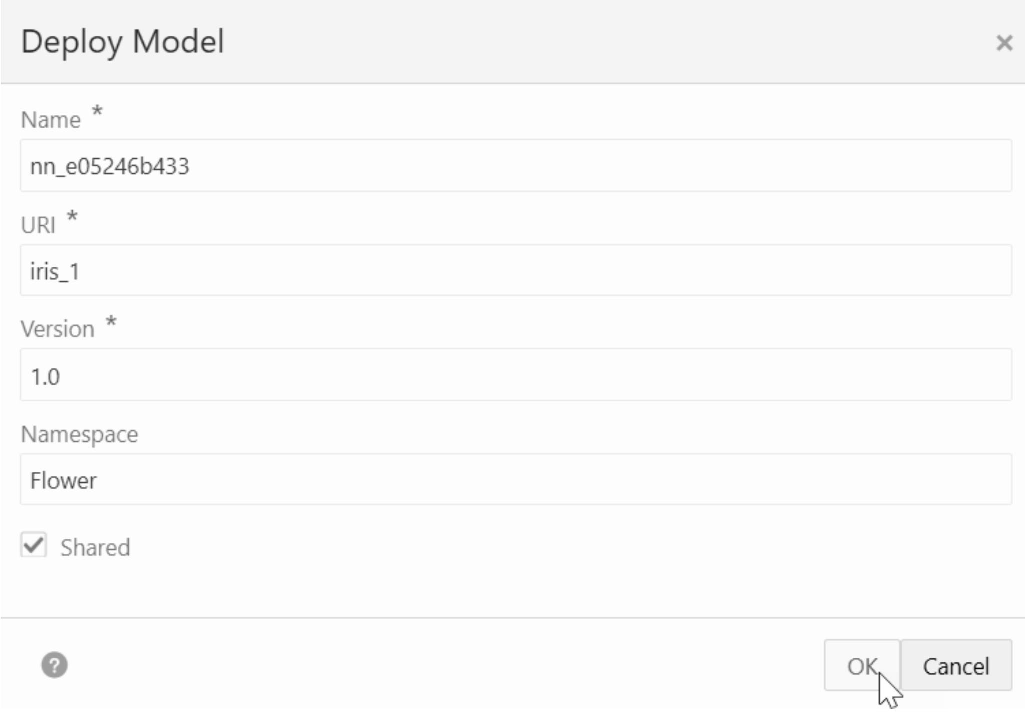
- Deploy Model
  When you deploy a model, you create an Oracle Machine Learning Services endpoint for scoring.

9.1 Deploy Model

When you deploy a model, you create an Oracle Machine Learning Services endpoint for scoring.

In the Deploy Model dialog box, you can define the model deployment in the context of your AutoML UI experiment. To deploy a model, define the following:

Figure 9-4 Deploy Model



Deploy Model

Name *

nn_e05246b433

URI *

iris_1

Version *

1.0

Namespace

Flower

☑ Shared

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 `xx.xx` where `x` is a number.
4. In the **Namespace** field, enter a name for the model namespace.
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.

10
Use Templates to Collaborate with Users

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.

10.1 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.

10.1.1 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.

10.1.2 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.

10.1.3 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.**

10.2 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.

10.2.1 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.

10.2.2 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.**

10.3 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 Database.

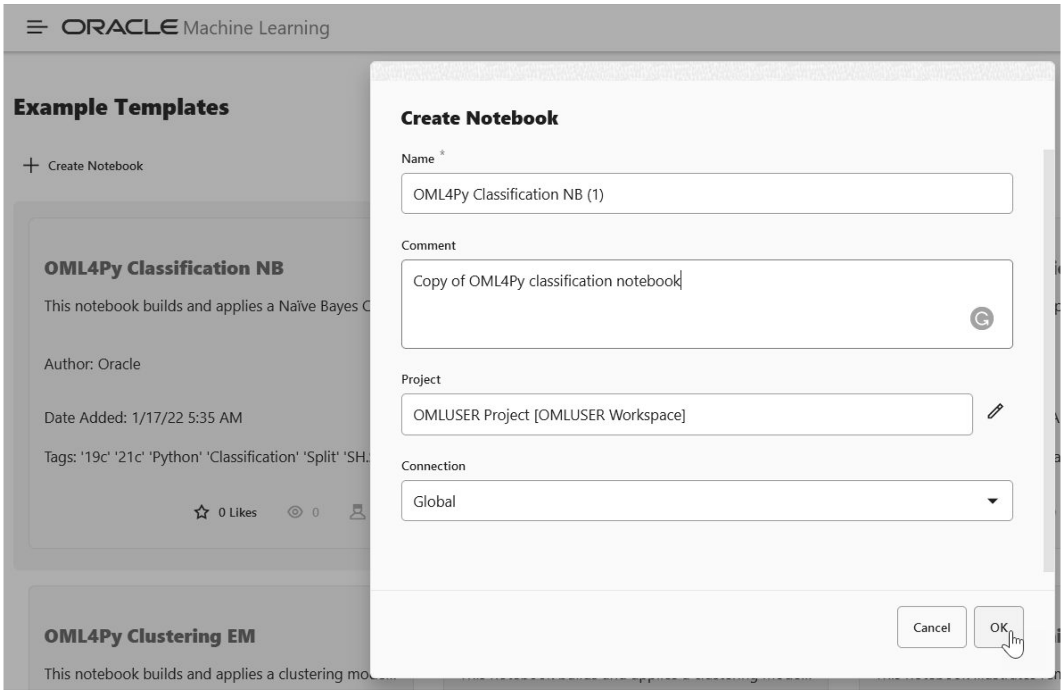
10.3.1 Create a Notebook from the Example Templates

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.



Create Notebook

Name *: OML4Py Classification NB (1)

Comment: Copy of OML4Py classification notebook

Project: OMLUSER Project [OMLUSER Workspace]

Connection: Global

Example Templates

OML4Py Classification NB

This notebook builds and applies a Naïve Bayes C

Author: Oracle

Date Added: 1/17/22 5:35 AM

Tags: '19c' '21c' 'Python' 'Classification' 'Split' 'SH.

OML4Py Clustering EM

This notebook builds and applies a clustering mo...

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.

10.3.2 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 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 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

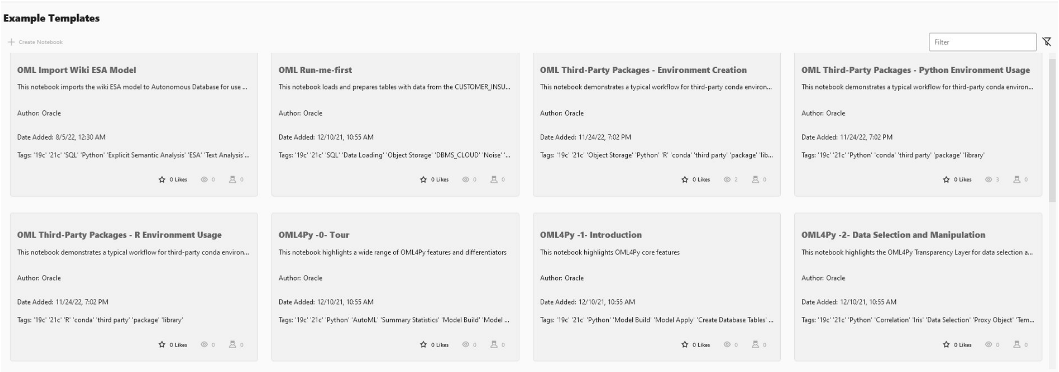
> **Note:**
>
> The Conda environment is installed and managed by the ADMIN user with `OML_SYS_ADMIN` role. The administrator can create a shared environment, and add or delete packages from it. The Conda environments are stored in the Object Storage bucket associated with the Autonomous Database.

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 Database.

Figure 10-1   Conda Example Templates



Example Templates

OML Import Wiki ESA Model
This notebook imports the wiki ESA model to Autonomous Database for use ...
Author: Oracle
Date Added: 8/5/22, 12:30 AM
Tags: '19c' '21c' 'SQL' 'Python' 'Explicit Semantic Analysis' 'ESA' 'Text Analysis'...

OML Run-me-first
This notebook loads and prepares tables with data from the CUSTOMER_INSU...
Author: Oracle
Date Added: 12/10/21, 10:55 AM
Tags: '19c' '21c' 'SQL' 'Data Loading' 'Object Storage' 'DBMS_CLOUD' 'Noise' '...

OML Third-Party Packages - Environment Creation
This notebook demonstrates a typical workflow for third-party conda environ...
Author: Oracle
Date Added: 11/24/22, 7:02 PM
Tags: '19c' '21c' 'Object Storage' 'Python' 'R' 'conda' 'third party' 'package' 'lib...

OML Third-Party Packages - Python Environment Usage
This notebook demonstrates a typical workflow for third-party conda environ...
Author: Oracle
Date Added: 11/24/22, 7:02 PM
Tags: '19c' '21c' 'Python' 'conda' 'third party' 'package' 'library'

OML Third-Party Packages - R Environment Usage
This notebook demonstrates a typical workflow for third-party conda environ...
Author: Oracle
Date Added: 11/24/22, 7:02 PM
Tags: '19c' '21c' 'R' 'conda' 'third party' 'package' 'library'

OML4Py -0- Tour
This notebook highlights a wide range of OML4Py features and differentiators
Author: Oracle
Date Added: 12/10/21, 10:55 AM
Tags: '19c' '21c' 'Python' 'AutoML' 'Summary Statistics' 'Model Build' 'Model ...

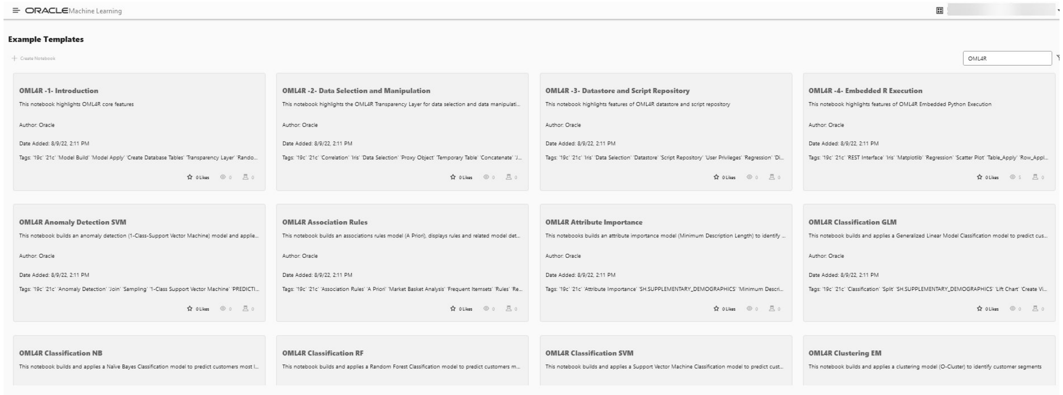
OML4Py -1- Introduction
This notebook highlights OML4Py core features
Author: Oracle
Date Added: 12/10/21, 10:55 AM
Tags: '19c' '21c' 'Python' 'Model Build' 'Model Apply' 'Create Database Tables' ...

OML4Py -2- Data Selection and Manipulation
This notebook highlights the OML4Py Transparency Layer for data selection a...
Author: Oracle
Date Added: 12/10/21, 10:55 AM
Tags: '19c' '21c' 'Python' 'Correlation' 'Iris' 'Data Selection' 'Proxy Object' 'Tem...

- **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 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

Figure 10-2 Oracle Machine Learning for R Example Templates



ORACLE Machine Learning

Example Templates

+ Create Notebook

OML4R

OML4R -1- Introduction
This notebook highlights OML4R core features
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Model Build' 'Model Apply' 'Create Database Tables' 'Transparency Layer' 'Rando...
☆ 0 Likes 0 0

OML4R -2- Data Selection and Manipulation
This notebook highlights the OML4R Transparency Layer for data selection and data manipulati...
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Correlation' 'Iris' 'Data Selection' 'Proxy Object' 'Temporary Table' 'Concatenate' 'J...
☆ 0 Likes 0 0

OML4R -3- Datastore and Script Repository
This notebook highlights features of OML4R datastore and script repository
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Iris' 'Data Selection' 'Datastore' 'Script Repository' 'User Privileges' 'Regression' 'Di...
☆ 0 Likes 0 0

OML4R -4- Embedded R Execution
This notebook highlights features of OML4R Embedded Python Execution
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'REST Interface' 'Iris' 'Matplotlib' 'Regression' 'Scatter Plot' 'Table_Apply' 'Row_Appl...
☆ 0 Likes 5 0

OML4R Anomaly Detection SVM
This notebook builds an anomaly detection (1-Class-Support Vector Machine) model and applie...
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Anomaly Detection' 'Join' 'Sampling' '1-Class Support Vector Machine' 'PREDICTI...
☆ 0 Likes 0 0

OML4R Association Rules
This notebook builds an associations rules model (A Priori), displays rules and related model det...
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Association Rules' 'A Priori' 'Market Basket Analysis' 'Frequent Itemsets' 'Rules' 'Re...
☆ 0 Likes 0 0

OML4R Attribute Importance
This notebooks builds an attribute importance model (Minimum Description Length) to identify ...
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Attribute Importance' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Minimum Descri...
☆ 0 Likes 0 0

OML4R Classification GLM
This notebook builds and applies a Generalized Linear Model Classification model to predict cus...
Author: Oracle
Date Added: 8/9/22, 2:11 PM
Tags: '19c' '21c' 'Classification' 'Split' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Lift Chart' 'Create Vi...
☆ 0 Likes 0 0

OML4R Classification NB
This notebook builds and applies a Naïve Bayes Classification model to predict customers most l...

OML4R Classification RF
This notebook builds and applies a Random Forest Classification model to predict customers m...

OML4R Classification SVM
This notebook builds and applies a Support Vector Machine Classification model to predict cust...

OML4R Clustering EM
This notebook builds and applies a clustering model (O-Cluster) to identify customer segments

- **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 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 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 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 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 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 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 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 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 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 Database.

  > **Note:**
  >
  > The pre-built Wikipedia model must be installed in your Autonomous 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.

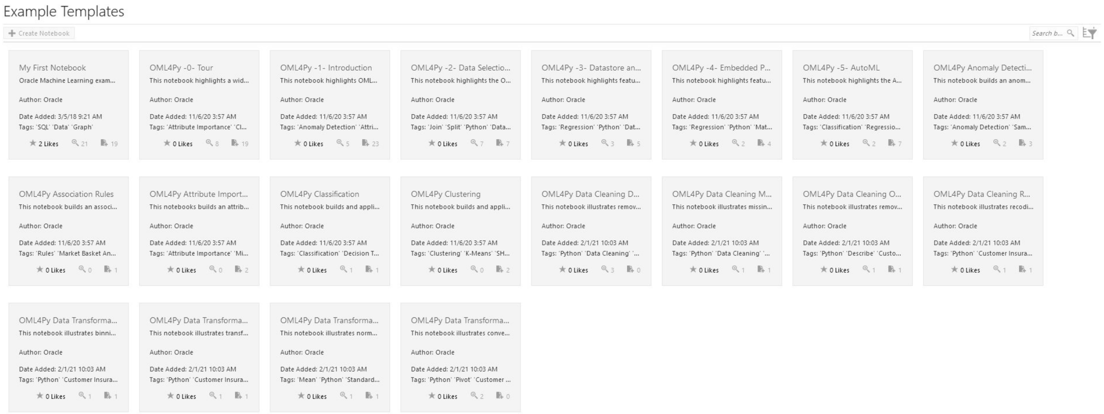
> **Note:**
>
> To run a script, it must reside in the R script repository. An Oracle Machine Learning cloud service account user name and password must be provided for authentication.

- **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 10-3 Oracle Machine Learning for Python Example Templates



My First Notebook
Oracle Machine Learning exam...
Author: Oracle
Date Added: 3/5/18 9:21 AM
Tags: 'SQL' 'Data' 'Graph'
2 Likes 21 19

OML4Py -0- Tour
This notebook highlights a wid...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Attribute Importance' 'Cl...
0 Likes 8 19

OML4Py -1- Introduction
This notebook highlights OML...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Anomaly Detection' 'Attri...
0 Likes 5 23

OML4Py -2- Data Selectio...
This notebook highlights the O...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Join' 'Split' 'Python' 'Data...
0 Likes 7 7

OML4Py -3- Datastore an...
This notebook highlights featu...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Regression' 'Python' 'Dat...
0 Likes 3 5

OML4Py -4- Embedded P...
This notebook highlights featu...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Regression' 'Python' 'Mat...
0 Likes 2 4

OML4Py -5- AutoML
This notebook highlights the A...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Classification' 'Regressio...
0 Likes 2 7

OML4Py Anomaly Detecti...
This notebook builds an anom...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Anomaly Detection' 'Sam...
0 Likes 2 3

OML4Py Association Rules
This notebook builds an associ...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Rules' 'Market Basket An...
0 Likes 0 1

OML4Py Attribute Import...
This notebooks builds an attrib...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Attribute Importance' 'Mi...
0 Likes 0 2

OML4Py Classification
This notebook builds and appli...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Classification' 'Decision T...
0 Likes 1 1

OML4Py Clustering
This notebook builds and appli...
Author: Oracle
Date Added: 11/6/20 3:57 AM
Tags: 'Clustering' 'K-Means' 'SH...
0 Likes 0 2

OML4Py Data Cleaning D...
This notebook illustrates remov...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Data Cleaning' '...
0 Likes 3 0

OML4Py Data Cleaning M...
This notebook illustrates missin...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Data Cleaning' '...
0 Likes 1 1

OML4Py Data Cleaning O...
This notebook illustrates remov...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Describe' 'Custo...
0 Likes 1 1

OML4Py Data Cleaning R...
This notebook illustrates recodi...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Customer Insura...
0 Likes 1 1

OML4Py Data Transforma...
This notebook illustrates binni...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Customer Insura...
0 Likes 1 1

OML4Py Data Transforma...
This notebook illustrates transf...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Customer Insura...
0 Likes 1 1

OML4Py Data Transforma...
This notebook illustrates norm...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Mean' 'Python' 'Standard...
0 Likes 1 1

OML4Py Data Transforma...
This notebook illustrates conve...
Author: Oracle
Date Added: 2/1/21 10:03 AM
Tags: 'Python' 'Pivot' 'Customer ...
0 Likes 2 0

- **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 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.

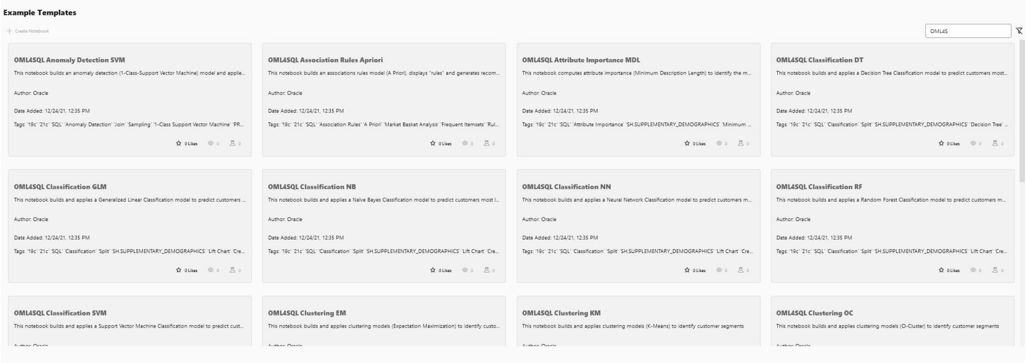
> **Note:**
>
> To run a script, it must reside in the OML4Py script repository. An Oracle Machine Learning cloud service account user name and password must be provided for authentication.

- **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 10-4 Oracle Machine Learning for SQL Example Templates



OML4SQL Anomaly Detection SVM
This notebook builds an anomaly detection (1-Class-Support Vector Machine) model and applie...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Anomaly Detection' 'Join' 'Sampling' '1-Class Support Vector Machine' 'PR...
0 Likes 0 0

OML4SQL Association Rules Apriori
This notebook builds an associations rules model (A Priori), displays "rules" and generates recom...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Association Rules' 'A Priori' 'Market Basket Analysis' 'Frequent Itemsets' 'Rul...
0 Likes 0 0

OML4SQL Attribute Importance MDL
This notebook computes attribute importance (Minimum Description Length) to identify the m...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Attribute Importance' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Minimum ...
0 Likes 0 0

OML4SQL Classification DT
This notebook builds and applies a Decision Tree Classification model to predict customers most...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Classification' 'Split' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Decision Tree' ...
0 Likes 0 0

OML4SQL Classification GLM
This notebook builds and applies a Generalized Linear Classification model to predict customers ...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Classification' 'Split' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Lift Chart' 'Cre...
0 Likes 0 0

OML4SQL Classification NB
This notebook builds and applies a Naïve Bayes Classification model to predict customers most l...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Classification' 'Split' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Lift Chart' 'Cre...
0 Likes 0 0

OML4SQL Classification NN
This notebook builds and applies a Neural Network Classification model to predict customers m...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Classification' 'Split' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Lift Chart' 'Cre...
0 Likes 0 0

OML4SQL Classification RF
This notebook builds and applies a Random Forest Classification model to predict customers m...
Author: Oracle
Date Added: 12/24/21, 12:35 PM
Tags: '19c' '21c' 'SQL' 'Classification' 'Split' 'SH.SUPPLEMENTARY_DEMOGRAPHICS' 'Lift Chart' 'Cre...
0 Likes 0 0

OML4SQL Classification SVM
This notebook builds and applies a Support Vector Machine Classification model to predict cust...

OML4SQL Clustering EM
This notebook builds and applies clustering models (Expectation Maximization) to identify custo...

OML4SQL Clustering KM
This notebook builds and applies clustering models (K-Means) to identify customer segments

OML4SQL Clustering OC
This notebook builds and applies clustering models (O-Cluster) to identify customer segments

- **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 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 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 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 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 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 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 Database.

11 Get Started with 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.

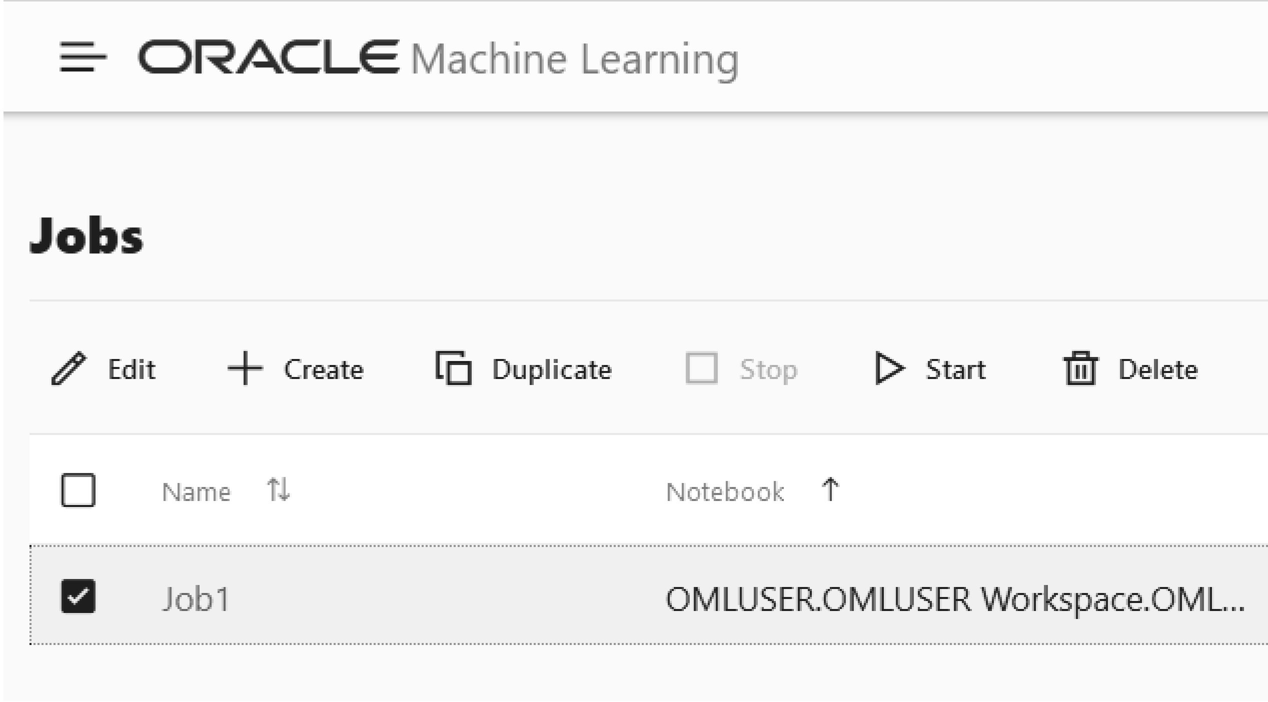
- About Jobs
  The Jobs page lists all the jobs created, along with the job name, notebook, owner of the job, last start date, next run date, status, and schedule.
- Create Jobs to Schedule Notebooks
  You can create jobs to schedule your notebook with preferred scheduling settings.
- View Job Logs
  You can view the historical logs of any particular job in the Job Log interface.

11.1 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 perform the following tasks:

Figure 11-1 Jobs



☐	Name	Notebook
☑	Job1	OMLUSER.OMLUSER Workspace.OML...

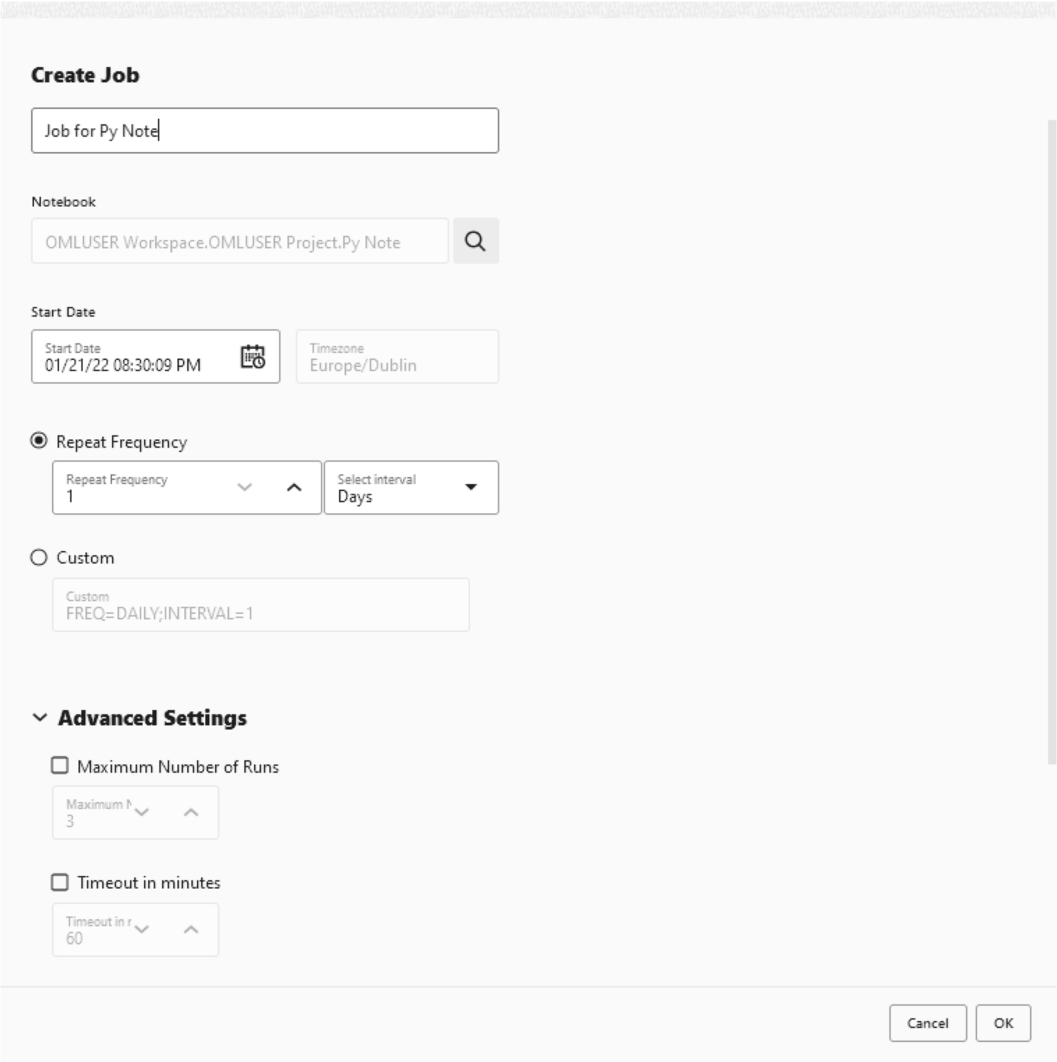
- Edit jobs: You can edit the metadata of any job listed on the Jobs page. Click **Edit** to edit the selected job.
- Create jobs. Click **Create** to create a new job to schedule your notebook.
- Duplicate jobs: You can create a copy of an existing job listed on the Jobs page. Click **Duplicate** to make a copy of the selected job.
- Stop jobs: Click **Stop** to terminate a job that is currently running.
- Start jobs: The **Start** button is enabled only for jobs that are in *Scheduled* status. Click **Start** to start a scheduled job. 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.
- Delete jobs: Click **Delete** to delete any job listed on the Jobs page.

11.2 Create Jobs to Schedule Notebooks

You can create jobs to schedule your notebook with preferred 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 box opens.



Create Job

Job for Py Note

Notebook
OMLUSER Workspace.OMLUSER Project.Py Note

Start Date
Start Date: 01/21/22 08:30:09 PM
Timezone: Europe/Dublin

● Repeat Frequency
Repeat Frequency: 1
Select interval: Days

○ Custom
Custom: FREQ=DAILY;INTERVAL=1

Advanced Settings

☐ Maximum Number of Runs
Maximum N: 3

☐ Timeout in minutes
Timeout in r: 60

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. Optionally, in **Advanced Settings,** select one or more of the following options:

- **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.
- **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`.
- **Timeout in Minutes:** To specify the maximum amount of time a job should be allowed to run.

7. Click **OK.**

11.3 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:

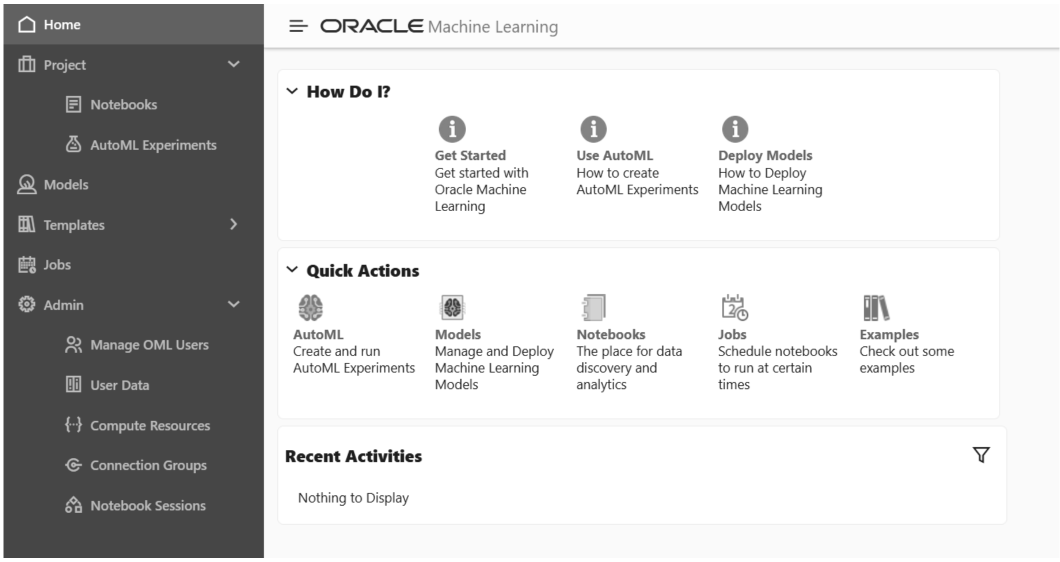
1. To view the history of a job, select the job and click View.
2. To delete a particular job log, select it and click Delete.

12 Administer Oracle Machine Learning

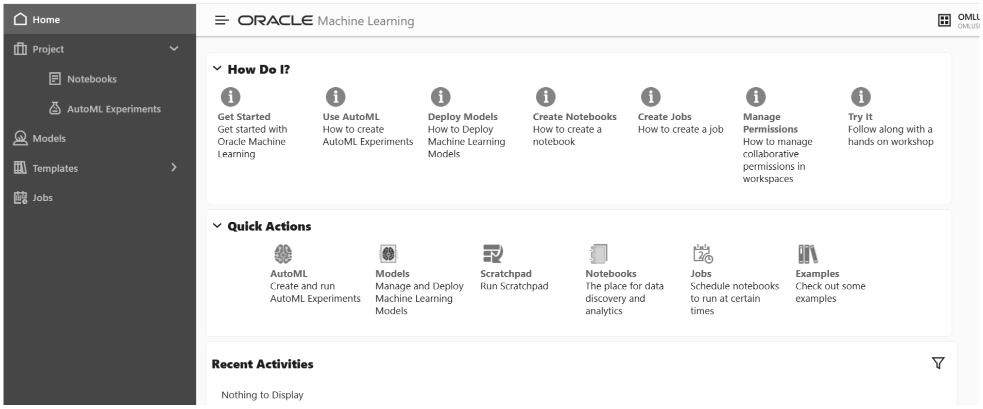
Oracle Machine Learning is managed at the system level and at the application level by an administrator.

- Administrator — Creates and manages Oracle Machine Learning user accounts, manages compute resourses, connection groups, and notebook sessions. The Administrator also reassigns user workspace.

  > **Note:**
  >
  > The Administrator is not authorized to run notebooks. The Administrator can only read notebooks.



- Developer — This is the default user role that allows you to create and run notebooks, run SQL Statements, create SQL scripts, run Python scripts, create jobs to schedule and run notebooks, use example template notebooks, create and run AutoML experiments, and deploy models.


Home
Project
Notebooks
AutoML Experiments
Models
Templates
Jobs
ORACLE Machine Learning
OMLU
OMLUS
How Do I?
Get Started
Get started with Oracle Machine Learning
Use AutoML
How to create AutoML Experiments
Deploy Models
How to Deploy Machine Learning Models
Create Notebooks
How to create a notebook
Create Jobs
How to create a job
Manage Permissions
How to manage collaborative permissions in workspaces
Try It
Follow along with a hands on workshop
Quick Actions
AutoML
Create and run AutoML Experiments
Models
Manage and Deploy Machine Learning Models
Scratchpad
Run Scratchpad
Notebooks
The place for data discovery and analytics
Jobs
Schedule notebooks to run at certain times
Examples
Check out some examples
Recent Activities
Nothing to Display

- 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
  An administrator manages new user account and user credentials creation for Oracle Machine Learning in the User Management interface.
- 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.

12.1 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

Tasks	Oracle Machine Learning Interface/OCI CLI Interface	More Information
User account and password creation	Oracle Machine Learning User Management interface	Create Users for Oracle Machine Learning
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 Database	About the Conda Environment and Conda Interpreter

> **Note:**
>
> The tasks listed here can be performed by an administrator only.

12.2 Access OML User Management from Command Line

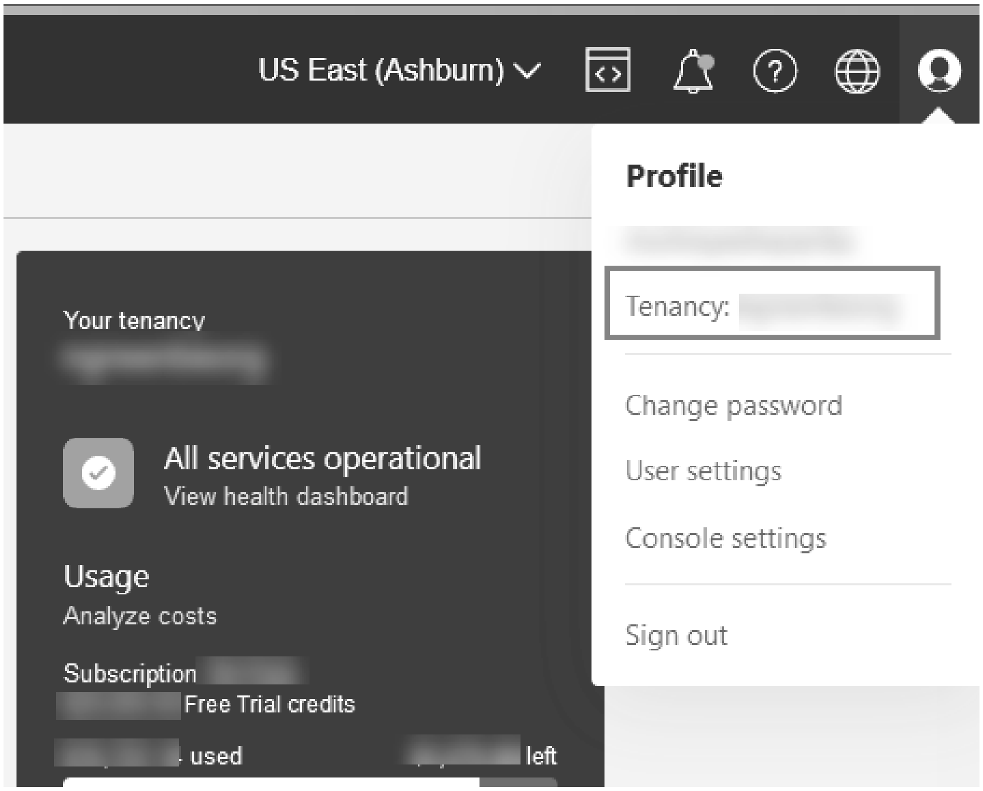
You can obtain the Oracle Machine Learning User Management URL for a specific tenancy from the Oracle Cloud Infrastructure (OCI) command line.

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.

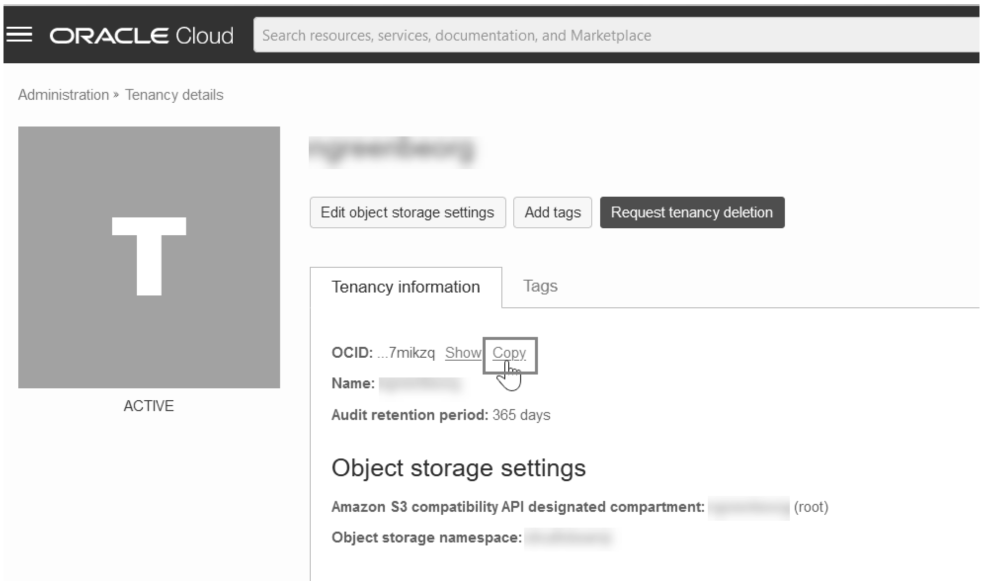
1. To obtain the tenancy ID, go to your OCI **Profile** on the top right corner of the Oracle Cloud page and click **Tenancy**.

Figure 12-1   OCI Profile



2. On the Tenancy details page, click **Copy** to obtain the tenancy URL.

Figure 12-2   Tenancy Information


ORACLE Cloud
Search resources, services, documentation, and Marketplace
Administration » Tenancy details
T
ACTIVE
Edit object storage settings
Add tags
Request tenancy deletion
Tenancy information
Tags

OCID: ...7mikzq Show Copy
Name: [redacted]
Audit retention period: 365 days

Object storage settings

Amazon S3 compatibility API designated compartment: [redacted] (root)
Object storage namespace: [redacted]

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.
- `tenancy 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.

12.3 Manage OML Users

An administrator manages new user account and user credentials creation for Oracle Machine Learning in the User Management interface.

- Create User
  An administrator creates new user accounts and user credentials for Oracle Machine Learning in the User Management interface.
- Add Existing Database User Account to Oracle Machine Learning Components
  As the ADMIN user you can add an existing database user account for Oracle Machine Learning components.

12.3.1 Create User

An administrator creates new user accounts and user credentials for Oracle Machine Learning in the User Management interface.

> **Note:**
>
> You must have the administrator role to access the Oracle Machine Learning User Management interface.

To create a user account:

1. On the Autonomous Databases page, under the **Display Name**, select an Autonomous Database.
2. On the Autonomous Database Details page, click **Database Actions**.
3. On the Database Actions launchpad, under **Administration**, click **Database Users**.
4. Click **+ Create User**.

5. In the **User Name** field, enter a username for the account. Using the username, the user will log in to an Oracle Machine Learning instance.
6. (Optional) Select the option **Password Expired (user must change).** to required the user to change their password when they login for the first time.
7. In the **Password** field, enter a password for the user.
8. In the **Confirm Password** field, enter a password to confirm the value that you entered in the **Password** field.
9. Select **OML** to enable Oracle Machine Learning for the user.
10. Click **Create User**.

This creates a new database user and grants the required privileges to use Oracle Machine Learning.

> **Note:**
>
> With a new database user, an administrator needs to issue grant commands on the database to grant table access to the new user for the tables associated with the user's Oracle Machine Learning notebooks.

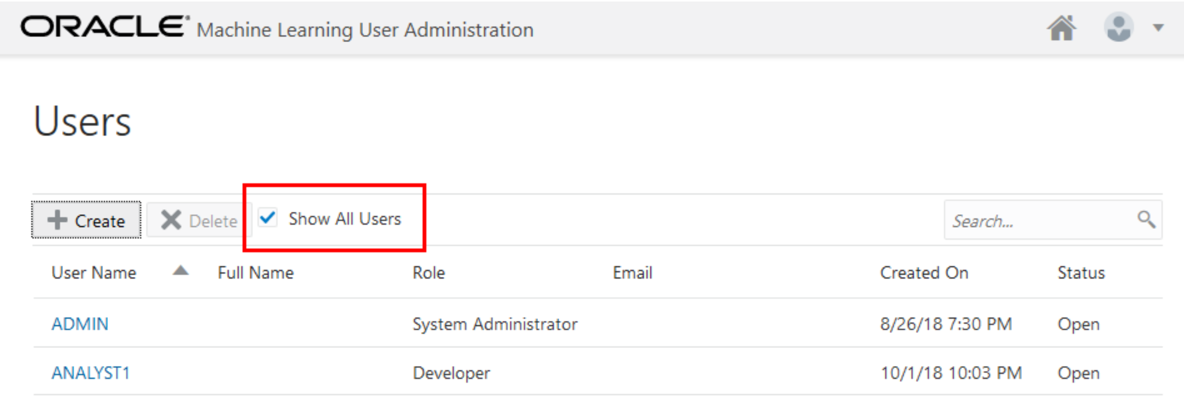
12.3.2 Add Existing Database User Account to Oracle Machine Learning Components

As the ADMIN user you can add an existing database user account for Oracle Machine Learning components.

> **Note:**
>
> You must have the ADMIN role to access the Oracle Machine Learning User Management interface.

To add an existing database user account:

1. On the Autonomous Databases page, under the **Display Name** column, select an Autonomous Database.
2. On the Autonomous Database Details page, click **Database Actions**.
3. On the Database Actions Launchpad, under **Development**, click **Oracle Machine Learning**.
4. Expand the navigator by clicking the ≡ next to Oracle Machine Learning.
5. Under Admin, select **Manage OML Users** to add Oracle Machine Learning Notebooks users.
6. Click **Show All Users** to display the existing database users.



User Name ▲	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		8/26/18 7:30 PM	Open
ANALYST1		Developer		10/1/18 10:03 PM	Open

> **Note:**
>
> Initially, the **Role** field shows the role **None** for existing database users. After adding a user the role **Developer** is assigned to the user.

7. Select a user. To select a user select a name in the **User Name** column. For example, select **ANALYST1**.

   Selecting the user shows the Oracle Machine Learning **Edit User** page.
8. Enter a name in the **First Name** field. (Optional)
9. Enter the last name of the user in the **Last Name** field. (Optional)
10. In the **Email Address** field, enter the email ID of the user.

    Making any change on this page adds the existing database user with the required privileges as an Oracle Machine Learning component user.
11. Click **Save**.

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.

12.4 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.

12.4.1 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.

12.5 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.

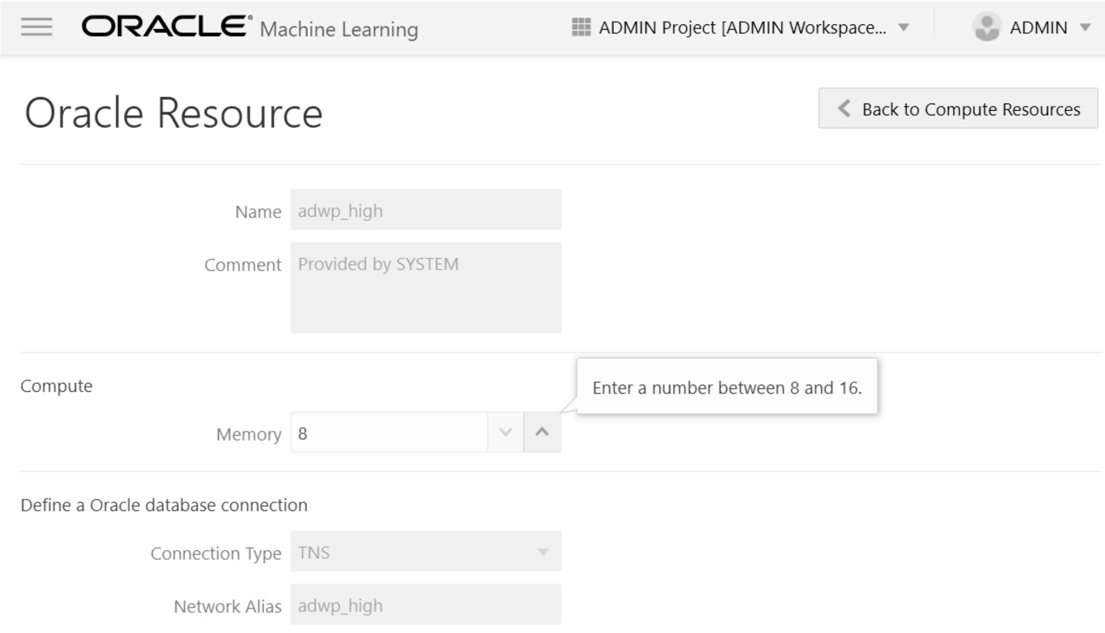
- Oracle Resource
  The Oracle Resource page displays the details of the selected compute resource on the Compute Resources page. You can configure the memory settings (in Gigabytes) for the Python interpreter for the selected compute resource.

12.5.1 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 12-3 Oracle Resource



Oracle Resource

Name: adwp_high

Comment: Provided by SYSTEM

Compute

Memory: 8

Enter a number between 8 and 16.

Define a Oracle database connection

Connection Type: TNS

Network Alias: adwp_high

To manage memory settings for the Python 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 Python interpreters in this field.
   - For the resource `databasename_high`, the memory settings (in Gigabytes) must be between 8 and 16
   - 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 PDB for each Resource service.

12.5.1.1 Resource Services and Notebooks

This topic lists the number of notebooks that you can run concurrently per PDB for each Resource service.

The Resource Services and Number of Notebooks table lists the Compute Resources assigned for running of Python scripts at different Resource Service levels - High, Medium and Low. The High level is assigned the maximum number of Compute Resources to run the Python script, which could result in faster running of the scripts. The Low level is assigned the least number of Resource Services, which results in slower running of the scripts.

Table 12-1 Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
High	Up to 8 OCPUs	8 GB (up to 16 GB)	Up to 3

Table 12-1 (Cont.) Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
Medium	Up to 4 (OCPUs)	4 GB (up to 8 GB)	Up to max (1.25 × number of OCPUs)

Note: The number of current notebook run is calcula

Table 12-1 (Cont.) Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
			ted by the formula `1.25 x (number of OCPUs)` provisioned for

Table 12-1 (Cont.) Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
			the corresponding PDB. OCPU stands for Oracle CPU. For examp

Table 12-1 (Cont.) Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
			le, if a PDB is provisioned with 4 OCPUs, then the maximum numb

Table 12-1 (Cont.) Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
			er of notebooks run would be 5 (1.25 x 4) in Medium level.
Low	1	2 GB (up to 4 GB)	Up to 100

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

12.6.1 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.

12.6.2 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 cancelled.

  > **Note:**
  >
  > The **Reset** option is available only to the Administrator.

12.6.3 Edit Oracle Database Interpreter Connection

When defining an Oracle 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 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.**

12.7 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.