

Oracle® Cloud

Data Science Agent User's Guide



G50084-02
March 2026



Oracle Cloud Data Science Agent User's Guide,

G50084-02

Copyright © 2026, 2026, Oracle and/or its affiliates.

Primary Author: Moitreyee Hazarika

Contributors: Mark Hornick, Boriana Milenova, Denny Wong, David Gurriere, Sherry LaMonica, Marat Spivak

Contents

1 About Data Science Agent

1.1	Prerequisites to use Data Science Agent	1
1.1.1	Use DBMS_CLOUD_AI to Configure AI Profiles	2
1.1.2	Grant OML_DEVELOPER Role to OML User	2
1.1.3	Add User to the Host ACL	2
1.1.4	Perform Prerequisites for Select AI	3
1.1.4.1	Grant Privileges for Select AI	4
1.1.4.2	Examples of Privileges to Run Select AI	5
1.2	Data Science Agent Concepts	8
1.3	Key Highlights of Data Science Agent	9
1.4	Limitations of Data Science Agent	10
1.4.1	Ad hoc SQL queries cannot be run directly	11
1.4.2	Algorithms supported by Oracle permitted for models	11
1.4.3	Conversation length and scope	11
1.4.4	Error handling	11
1.4.5	Limitations in result visualization	12
1.4.6	Performance and latency related limitations	12
1.4.7	Reuse of existing objects	12
1.5	Data Science Agent: Sample Prompts and Outputs	12

2 Get Started with Data Science Agent

2.1	Access Data Science Agent	1
2.2	Create AI Credential and AI Profile	3
2.3	Manage Data Science Conversations	6
2.3.1	Create a Data Science Agent Conversation	7
2.3.2	Delete a Data Science Agent Conversation	9
2.4	Use Data Science Agent Chat Interface	10
2.5	Example of a Conversation with Data Science Agent	12

3 Best Practices in using Data Science Agent

3.1	Recommended Models	1
3.2	Associate Database objects to your conversation	4

3.3	Ask for clarification	4
3.4	Ask in multiple iterations	4
3.5	Clarify terminology	5
3.6	Follow suggestions provided by Data Science Agent	5
3.7	Limit your conversation length and scope	5
3.8	Provide context to your conversation	5
3.9	Specify a clear objective	5

4 Known Issues in Data Science Agent

4.1	Objects present in user schema are not listed	1
-----	---	---

1

About Data Science Agent

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

Topics:

- [Prerequisites to use Data Science Agent](#)
To use Data Science Agent, you must have the following:
- [Data Science Agent Concepts](#)
Here is a list of key concepts and terms commonly used in Data Science Agent.
- [Key Highlights of Data Science Agent](#)
Data Science Agent offers a range of powerful features designed to streamline data science workflows. The key features include:
- [Limitations of Data Science Agent](#)
While Data Science Agent offers numerous benefits, there are certain limitations that may impact its use in specific scenarios.
- [Data Science Agent: Sample Prompts and Outputs](#)
Here are some sample prompts and outputs related to various machine learning domains on which you may have conversations with Data Science Agent.

1.1 Prerequisites to use Data Science Agent

To use Data Science Agent, you must have the following:

- `DBMS_CLOUD_AI` profile (AI profile) and `DBMS_CLOUD` credentials (AI Credential) created. For more information, see [Use DBMS_CLOUD_AI to Configure AI Profiles](#).
- The `OML_DEVELOPER` role must be granted to the `OMLUSER`.

Note

If the user (`OMLUSER`) is created through Database Actions, the `OML_DEVELOPER` role is automatically granted.

- User must be added to the host ACL (Access Control List).

Note

This is not required for OCI Generative AI.

- Select AI must be configured to use supported AI providers. For more information, see [Perform Prerequisites for Select AI](#)

- Access to the relevant schemas and objects based on your role and privileges.

For additional information, see [Manage AI Profiles](#)

- [Use DBMS_CLOUD_AI to Configure AI Profiles](#)
Autonomous AI Database uses AI profiles to facilitate and configure access to an LLM and to setup for generating, running, and explaining SQL based on natural language prompts. It also facilitates retrieval augmented generation using embedding models and vector indexes and allows for chatting with the LLM.
- [Grant OML_DEVELOPER Role to OML User](#)
To use Data Science Agent, the administrator must grant the OML_DEVELOPER role to the OML user.
- [Add User to the Host ACL](#)
For model providers like OpenAI, add users to the host ACL (Access Control List).
- [Perform Prerequisites for Select AI](#)
Before you use Select AI, here are the steps to enable DBMS_CLOUD_AI.

1.1.1 Use DBMS_CLOUD_AI to Configure AI Profiles

Autonomous AI Database uses AI profiles to facilitate and configure access to an LLM and to setup for generating, running, and explaining SQL based on natural language prompts. It also facilitates retrieval augmented generation using embedding models and vector indexes and allows for chatting with the LLM.

AI profiles include database objects that are the target for natural language queries. Metadata used from these targets can include database table names, column names, column data types, and comments. You create and configure AI profiles using the following procedures:

- [DBMS_CLOUD_AI.CREATE_PROFILE](#)
- [DBMS_CLOUD_AI.SET_PROFILE](#)

In addition to specifying tables and views in the AI profile, you can also specify tables mapped with external tables, including those described in [Query External Data with Data Catalog](#) Query External Data with Data Catalog. This enables you to query data not just inside the database, but also data stored in a data lake's object store.

1.1.2 Grant `OML_DEVELOPER` Role to OML User

To use Data Science Agent, the administrator must grant the OML_DEVELOPER role to the OML user.

If the OML user (OMLUSER) is created through Database Actions, the OML_DEVELOPER role is automatically granted.

To grant the OML_DEVELOPER role, run the following:

```
GRANT OML_DEVELOPER to OMLUSER
```

1.1.3 Add User to the Host ACL

For model providers like OpenAI, add users to the host ACL (Access Control List).

Note

Host ACL entry is not required for OCI GenAI.

The following procedure grants the privilege to use the *api.openai.com* endpoint.

Note

This procedure is not applicable to OCI Generative AI.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host => 'api.openai.com',
    ace => xs$ace_type(privilege_list => xs$name_list('http'),
                      principal_name => 'OMLUSER',
                      principal_type => xs_acl.ptype_db)
  );
END;
```

The parameters are:

- host:** The host, which can be the name or the IP address of the host. You can use a wildcard to specify a domain or an IP subnet. The host or domain name is not case sensitive.

AI Provider	Host
OpenAI	<i>api.openai.com</i>
OpenAI-compatible providers	For example, for Fireworks AI, use <i>api.fireworks.ai</i>
Cohere	<i>api.cohere.ai</i>
Azure OpenAI Service	<i><azure_resource_name>.openai.azure.com</i> See Profile Attributes to know more about <i>azure_resource_name</i> .
Google	<i>generativelanguage.googleapis.com</i>
Anthropic	<i>api.anthropic.com</i>
Hugging Face	<i>api-inference.huggingface.co</i>
AWS	<i>bedrock-runtime.us-east-1.amazonaws.com</i>

- ace:** The access control entries (ACE). The `XS$ACE_TYPE` type is provided to construct each ACE entry for the ACL. For more details, see [Creating ACLs and ACEs](#).

1.1.4 Perform Prerequisites for Select AI

Before you use Select AI, here are the steps to enable `DBMS_CLOUD_AI`.

The following are required to use `DBMS_CLOUD_AI`:

- Access to an Oracle Cloud Infrastructure cloud account and to an Autonomous AI Database instance.

- A paid API account of a supported AI provider, one of:

AI Provider	API Keys
OpenAI	See Use OpenAI to get your API keys.
OpenAI-compatible providers	See Use OpenAI-Compatible Providers to get your API keys and <code>provider_endpoint</code> .
Cohere	See Use Cohere to get your secret API keys.
Azure OpenAI Service	See Use Azure OpenAI Service for more information on how to configure Azure OpenAI Service.
OCI Generative AI	See Use OCI Generative AI .
Google	See Use Google to get your API keys.
Anthropic	See Use Anthropic to get your API keys.
Hugging Face	See Use Hugging Face to get your API keys.
AWS	See Use AWS to get your API keys and model ID.

- Network ACL privileges to access your external AI provider.

Note

Network ACL privileges are not required for OCI Generative AI.

- A credential that provides access to the AI provider.
- [Grant Privileges for Select AI](#)
To use Select AI, the administrator must grant the `EXECUTE` privilege on the `DBMS_CLOUD_AI` package. Learn about additional privileges required for Select AI and its features.
- [Examples of Privileges to Run Select AI](#)
Review examples of privileges required to use Select AI and its features.

1.1.4.1 Grant Privileges for Select AI

To use Select AI, the administrator must grant the `EXECUTE` privilege on the `DBMS_CLOUD_AI` package. Learn about additional privileges required for Select AI and its features.

To configure `DBMS_CLOUD_AI`:

1. Grant the `EXECUTE` privilege on the `DBMS_CLOUD_AI` package to the user who wants to use Select AI.

By default, only the system administrator has `EXECUTE` privilege. The administrator can grant `EXECUTE` privilege to other users.

2. Grant `EXECUTE` privilege on `DBMS_CLOUD_PIPELINE` to the user who wants to use Select AI with RAG.

Note

If the user already has the `DWROLE` role, this privilege is included and additional grant is not required.

3. Grant network ACL access to the user who wants to use Select AI and for the AI provider endpoint.
The system administrator can grant network ACL access. See [APPEND_HOST_ACE Procedure](#) for more information.
4. Create a credential to enable access to your AI provider.
See [CREATE_CREDENTIAL Procedure](#) for more information.
5. Grant quotas in tablespace to manage the amount of space in a specific tablespace to the user who wants to use Select AI with RAG.

1.1.4.2 Examples of Privileges to Run Select AI

Review examples of privileges required to use Select AI and its features.

The following example grants the `EXECUTE` privilege to `ADB_USER`:

```
GRANT execute on DBMS_CLOUD_AI to ADB_USER;
```

The following example grants `EXECUTE` privilege for the `DBMS_CLOUD_PIPELINE` package required for RAG:

```
GRANT EXECUTE on DBMS_CLOUD_PIPELINE to ADB_USER;
```

To check the privileges granted to a user for the `DBMS_CLOUD_AI` and `DBMS_CLOUD_PIPELINE` packages, an administrator can run the following:

```
SELECT table_name AS package_name, privilege
FROM DBA_TAB_PRIVS
WHERE grantee = '<username>'
AND (table_name = 'DBMS_CLOUD_PIPELINE'
OR table_name = 'DBMS_CLOUD_AI');
```

The following example grants `ADB_USER` the privilege to use the `api.openai.com` endpoint.

Note

This procedure is not applicable to OCI Generative AI.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host => 'api.openai.com',
    ace => xs$ace_type(privilege_list => xs$name_list('http'),
      principal_name => 'ADB_USER',
      principal_type => xs_acl.p_type_db)
  );
END;
/
```

The parameters are:

- `host`: The host, which can be the name or the IP address of the host. You can use a wildcard to specify a domain or an IP subnet. The host or domain name is not case sensitive.

AI Provider	Host
OpenAI	<i>api.openai.com</i>
OpenAI-compatible providers	For example, for Fireworks AI, use <i>api.fireworks.ai</i>
Cohere	<i>api.cohere.ai</i>
Azure OpenAI Service	<i><azure_resource_name>.openai.azure.com</i> See Profile Attributes to know more about <i>azure_resource_name</i> .
Google	<i>generativelanguage.googleapis.com</i>
Anthropic	<i>api.anthropic.com</i>
Hugging Face	<i>api-inference.huggingface.co</i>
AWS	<i>bedrock-runtime.us-east-1.amazonaws.com</i>

- `ace`: The access control entries (ACE). The `XS$ACE_TYPE` type is provided to construct each ACE entry for the ACL. For more details, see [Creating ACLs and ACEs](#).

The following example creates a credential to enable access to OpenAI.

```
EXEC
DBMS_CLOUD.CREATE_CREDENTIAL(
credential_name => 'OPENAI_CRED',
username       => 'OPENAI',
password      => '<your_api_token>');
```

The parameters are:

- `credential_name`: The name of the credential to be stored. The `credential_name` parameter must conform to Oracle object naming conventions.
- `username`: The `username` and `password` arguments together specify your AI provider credentials.

The `username` is a user-specified user name.

- `password`: The `username` and `password` arguments together specify your AI provider credentials.

The `password` is your AI provider secret API key, and depends on the provider, that is, OpenAI, Cohere, or Azure OpenAI Service.

AI Provider	API Keys
OpenAI	See #unique_13 to get your API keys.
OpenAI-compatible providers	See #unique_14 to get your API keys and <code>provider_endpoint</code> .
Cohere	See #unique_15 to get your API keys.

AI Provider	API Keys
Azure OpenAI Service	See #unique_16 to get your API keys and to configure the service.
<div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>If you are using the Azure OpenAI Service principal to authenticate, you can skip the <code>DBMS_CLOUD.CREATE_CREDENTIAL</code> procedure. See Examples of Using Select AI for an example of authenticating using Azure OpenAI Service principal.</p> </div>	
OCI Generative AI	See #unique_17 to generate API signing keys.
Google	See #unique_18 to generate your API keys.
Anthropic	See #unique_19 to generate your API keys.
Hugging Face	See #unique_20 to generate your API keys.
AWS	See #unique_21 to get your API keys and model ID.

The following example grants quotas on tablespace to the `ADB_USER` to use Select AI with RAG:

```
ALTER USER ADB_USER QUOTA 1T ON <tablespace_name>;
```

To check the tablespace quota granted to a user, run the following:

```
SELECT TABLESPACE_NAME, BYTES, MAX_BYTES
FROM DBA_TS_QUOTAS
WHERE USERNAME = '<username>' AND
      TABLESPACE_NAME LIKE 'DATA%';
```

The parameters are:

- `TABLESPACE_NAME`: The tablespace for which the quota is assigned. In Autonomous AI Database, tablespaces are managed automatically and have `DATA` as a prefix.
- `BYTES`: The amount of space currently used by the user in the tablespace.
- `MAX_BYTES`: The maximum quota assigned (in bytes). If `MAX_BYTES` is `-1`, it means the user has unlimited quota on the tablespace. The database user creating the vector index must have `MAX_BYTES` sufficiently larger than bytes to accommodate the vector index, or `MAX_BYTES` should be `-1` for unlimited quota.

1.2 Data Science Agent Concepts

Here is a list of key concepts and terms commonly used in Data Science Agent.

AI Credential

An AI credential is information that contains authentication details required for the database to access Oracle Cloud Infrastructure (OCI) resources on behalf of a user or application. You create an AI credential using the `dbms_cloud.create_credential` procedure. The credential comprises the following information:

- `user_ocid`: This is the unique identifier of the OCI user.
- `tenancy_ocid`: This is the unique identifier of the OCI tenancy in your cloud account.
- `private_key`: The private key associated with the OCI user. It is required for secure authentication.
- `fingerprint`: The fingerprint of the public key linked to the OCI user.

AI Profile

An AI Profile is a named configuration that specifies how the database connects to an LLM — including the provider, (for example, `openai`, `oci`), credential, model, and optional parameters such as `temperature` and `max_tokens` and so on. You create and manage AI profiles through the `DBMS_CLOUD_AI` package.

For more information, see [Manage AI Profiles](#).

Conversation

The interaction with Data Science Agent takes the form of a conversation, each made up of alternating `turns`. Each `turn` begins with a user prompt, followed by the agent's response. The conversation retains the context throughout, allowing you to refer back to previous answers in later questions. For example, you might ask, "filter the dataset you just profiled," or "train a model using the training dataset we prepared".

Conversation history

Conversation history is a persistent record of past conversations with the agent. It allows you to browse through conversation history, review previous results, and continue past sessions without losing the context. This ensures continuity over time, allows multiple workloads in separate chats, supports reproducibility of analyses, and provides an auditable trail on how insights were derived.

Conversation Objects Catalog

Data Science Agent operates on and produces three types of database object while handling requests. If you associate these objects to your conversation, the agent can inspect, analyze, transform, and model from those objects directly. This will thereby enhance the quality of the agent's response. If you do not associate any object, the agent will automatically scan the database for relevant objects based on your prompt.

The objects are available in the **Conversation Object Catalog**:

- **Tables**: Source data and persisted modeling results (created by the agent)
- **Views**: Views may be pre-existing data sources or derived datasets created by the agent. They are used for analysis, modeling, or general data transformation. Views created by the agent use the prefix `DSAGENT$` and may include a unique suffix.

- Mining models: The Oracle Machine Learning (OML) models trained by the agent.

For more information on how to associate these objects to your conversation, see .

Prompt

A prompt is your input or message that initiates an interaction. It can be a question, command, statement, or request that Data Science Agent processes in order to generate an appropriate response. Essentially, the prompt guides the agent on what information or action you are seeking.

Prompt library

A prompt library is a curated set of system, task, and tool-specific prompts that defines how the agent interprets your prompts, interprets results, and calls various tools. The prompts are designed to encode domain knowledge and ensure consistent, reliable behavior.

Service Levels

In Oracle Machine Learning (OML) on Autonomous AI Database, Service Levels refer to the predefined configurations for resource allocation and workload management. Essentially, it determines how much OCPUs (Oracle CPUs), ECPUs, and memory are allocated to a session. There are four types of service levels—Low, Medium, and High.

These service levels help manage and prioritize workloads running on the database, ensuring appropriate performance based on the use case.

For more information how to change the Service Levels of your conversation, see [Use Data Science Agent Chat Interface](#).

Tools

Tools are modular components that enable the agent to perform specific tasks such as profiling a data object, computing feature correlations, or training a model. Each tool has clearly defined inputs, outputs, and constraints. In short, tools serve as the building blocks of Data Science Agent's functionality. Although, the end users do not interact with these tools directly, it determines the user experience of the agent.

1.3 Key Highlights of Data Science Agent

Data Science Agent offers a range of powerful features designed to streamline data science workflows. The key features include:

- **Data Discovery and Inspection:** Accesses and discovers data locally as well as from remote sources including non-Oracle databases in multi-cloud environments.
- **Exploratory Statistical Analysis:** Conducts single-variable analysis as well as relationship analysis. Relationship analysis is performed pairwise, that is, between two variables such as predictors and outcomes. This means each predictor is examined individually against one outcome. Data Science Agent can scan many predictors against a single outcome; however, this process does not replace multivariate modeling.

Note

Relationship analyses are most reliable when performed on row-level (fine-grained) datasets, rather than on heavily aggregated data.

- **View-based Data Preparation:** Transforms and prepares data for modeling by creating new views. This is how it joins tables, filters populations, and derives new features from existing attributes.
- **Data Analysis and Visualization:** Simplifies and automates data analysis with built-in visualization for actionable insights.
- **Feature Selection and Feature engineering:** Profiles datasets, and performs feature selection and feature engineering.
- **Model Training (supervised and unsupervised) including Automated Model Search:** Handles training for both supervised and unsupervised models, thereby providing clear explanations of metrics and results to support learning and decision-making. It supports Classification, Regression, Clustering, and Anomaly Detection. Supported algorithms include XGBoost, Random Forest, Decision Tree, Neural Network, Naive Bayes, SVM, GLM, K-Means, Expectation Maximization, and O-Cluster. Converse with the agent to:
 - Train models to predict a categorical outcome (Classification) or a numeric value (Regression)
 - Evaluate multiple supervised algorithms and pick the best model based on a metric (automated model search), and
 - Build models without a labeled target (Clustering and Anomaly Detection)
- **Model Comparison and Evaluation:** Handles model comparison and evaluation. If you have multiple models, either created by the agent or otherwise, you can request the agent for a comparison based on a common validation dataset.
- **Inference (scoring) on new data:** Performs inference (scoring) on new data. Inference requires a trained model, a dataset containing the full feature set expected by the model, and dataset containing the IDs to score.

Note

Inference is supported only in ID-based scoring mode, that is IDs to score along with full feature dataset. Broader scoring options will be available soon.

1.4 Limitations of Data Science Agent

While Data Science Agent offers numerous benefits, there are certain limitations that may impact its use in specific scenarios.

The current limitations of Data Science Agent include:

- [Ad hoc SQL queries cannot be run directly](#)
- [Algorithms supported by Oracle permitted for models](#)
- [Conversation length and scope](#)
- [Error handling](#)
- [Limitations in result visualization](#)
- [Performance and latency related limitations](#)
- [Reuse of existing objects](#)

1.4.1 Ad hoc SQL queries cannot be run directly

Data Science Agent is capable of generating SQL internally to create views. However, it does not support running of ad hoc SQL queries or direct visualization of raw result sets currently.

Statistical analysis of Data Science Agent is highly effective when working with row-level datasets, and not aggregated outputs. However, some analyses on grouped data can be performed if the row count per group is large. Therefore, for more reliable analysis and modeling, use views that has ungrouped or only minimally aggregated data.

Note

You can define arbitrary views to structure and transform data for downstream analysis and modeling.

1.4.2 Algorithms supported by Oracle permitted for models

Data Science Agent permits algorithms that are only supported by Oracle. Currently, the agent supports the following machine learning functions—Classification, Regression, Clustering, and Anomaly Detection.

Note

Inference or scoring is not supported for Clustering and Anomaly Detection.

1.4.3 Conversation length and scope

While Data Science Agent can handle extended interactions, very long conversations may gather context that negatively affects clarity or performance. For extended work, consider starting a new conversation after substantial number of interactions (around 50 messages), particularly when your objectives change.

1.4.4 Error handling

Data Science Agent may occasionally encounter constraints. For example, unsupported column types for modeling. These are typically resolved by adjusting the data or approach. If you encounter such constraint, ask the agent for suggestions on how to solve minor issues.

Note

Oracle recommends refining prompts, adjusting goals, or re-running steps.

1.4.5 Limitations in result visualization

Data Science Agent only provides the summaries of its analysis or limited data samples. Interactive viewing of raw query results is not supported currently.

1.4.6 Performance and latency related limitations

Certain operations such as data discovery, feature analysis, and model training may require a few minutes to process. Model training on very large datasets can take even longer. During these operations, the conversation may not progress until the operation is completed. If you encounter such performance or latency related issues, you can start other conversations.

1.4.7 Reuse of existing objects

Data Science Agent may reuse existing objects—views or models, although *starting from scratch* is also an option. If you prefer that the agent doesn't reuse previous objects, you can state so in your response. Otherwise, the agent may refer to or reuse relevant objects created earlier—including those from other conversations—when they are manually associated or automatically discovered. This is done to save time by avoiding repeated creation of the same objects.

Note

If several similar objects are available, make sure that you specify whether to reuse or recreate the objects.

1.5 Data Science Agent: Sample Prompts and Outputs

Here are some sample prompts and outputs related to various machine learning domains on which you may have conversations with Data Science Agent.

1. Data Discovery

Discovery is semantic and goal-driven. It works best and more efficiently when the goal and domain are stated explicitly.

You can ask the agent to find data objects relevant to a business topic or analysis goal. For example, marketing response, churn, fraud, product demand. It can also you obtain a general overview of all available objects.

Example 1-1 Discover available tables, views, and models

Sample prompts

- Find tables and views related to bank marketing subscriptions and campaign contacts.
- What data exists related to bank marketing?
- Find tables related to customer churn and retention?
- What objects are available?

Outputs

Here are some expected outputs for the above prompts:

- A curated set of relevant objects—tables, views, models.
- Business-oriented summaries and hints about how objects relate. For example, likely join keys.
- Additional extended report with detailed information about all relevant objects.

① Note

- Best results depend on meaningful metadata. Semantically clear tables, views, column names and well-maintained annotations improve quality and relevance of results.
- You can manually associate database objects—tables, views, or models to the conversation so the agent can use them immediately. This is useful when the relevant objects are already known and discovery is unnecessary. Once associated, the agent can inspect, analyze, transform, and model from those objects directly. Discovery can remain optional unless additional data needs to be found.

2. Inspect Specific Object

You can ask for details about a specific table, view, or mining model.

Example 1-2 Ask questions related to specific tables, views and mining models

Sample prompts

- Describe the `CUSTOMERS` table?
- Show the columns and types in `SCHEMA.SALES_TRANSACTIONS`?
- What attributes are used in the model `CHURN_MODEL`?

Outputs

Here are some expected outputs for the above prompts:

- For tables and views, the agent will typically retrieve information related to row and column counts, column list and data types, a small data sample.
- For models, the agent will typically retrieve information related to features, target and algorithm data.

3. Exploratory Statistical Analysis

For exploratory statistical analysis, you can ask the agent for both single-variable analysis and relationship analysis (pair).

Example 1-3 Single-variable analysis

You can request distribution and qualitative summaries for one or more individual columns.

Sample prompts

- Describe the `SALES.CUSTOMERS` table and provide an overview of all its attributes.
- Provide an overview of all variables in `SCHEMA.CUSTOMERS_VIEW`
- Analyze the distribution of `AGE`, `INCOME`, and `JOB_CATEGORY`.

- Analyze which factors are most associated with subscription behavior.

Outputs

Here are some expected outputs for the above prompts:

- Global interpretation of analysis results.
- Distribution summaries for each variable, using statistics and plots appropriate to the variable type, that is, numeric versus categorical.
- Percentage of missing values and number of categories, as applicable.

Example 1-4 Relationship analysis

You can request statistical analysis of the relationship between two variables, such as predictors and outcomes. Relationship analysis is performed pairwise. This means each predictor is examined individually against one outcome. Data Science Agent can scan many predictors against a single outcome; however, this process does not replace multivariate modeling.

Note

Relationship analyses are most reliable when performed on row-level (fine-grained) datasets, rather than on heavily aggregated data.

Sample prompts

- What factors are most associated with subscriptions?
- How does `CONTACT_CHANNEL` relate to `AGE`?
- Analyze relationships of all features versus `CHURN_FLAG`.

Outputs

Here are some expected outputs for the above prompts:

- Global interpretation of pairwise analysis results.
- Pairwise relationship summaries for each attribute (against target variable), using statistics and plots appropriate to the variable types (numeric vs. categorical).

4. View-Based Data Transformation and Preparation

The agent can transform and prepare data for modeling by creating new views. This is how it joins tables, filters populations, and derives new features from existing attributes. Here are some common view-building tasks:

- Join customer, transaction, and interaction tables into a unified dataset.
- Filter to a time window or segment. For example, last 12 months, specific product line and so on.
- Create derived fields. For example, date components such as year/month/day or day of week.
- Exclude unsupported or non relevant fields from training datasets when needed.
- Create a new view joining clients, contacts, and past campaigns; extract day and month from timestamps

The agent does not run arbitrary adhoc SQL queries and return full result sets for interactive browsing. Views are the primary mechanism for shaping data.

The agent does not directly modify base tables.

Example 1-5 View-Based Data Transformation and Preparation

Sample prompts

- Join CLIENTS, CONTACTS, and PAST_CAMPAGNS into a modeling dataset.
- Make the dataset ready for modeling by extracting features from timestamps.

Outputs

Here are some expected outputs for the above prompts:

- A new view in the user schema starting with the prefix DSAGENT\$
- A plain-language summary of what the view contains and how it was created
- SQL code used to create the view.
- Visual diagram to track dependencies and operations at a glance.

5. Feature Importance and Feature Selection

Ask which variables matter most for predicting a specific target and optionally reduce the dataset to most important features.

Example 1-6 Feature Importance and Feature Selection

Sample Prompts

- Rank feature importance for predicting SUBSCRIBED.
- Create a reduced dataset with only important features

Outputs

Here are some expected outputs for the above prompts:

- A ranked list of attributes with importance scores
- Optionally, a new top-features view created from the original dataset

Note

Feature importance can be computed using different supported algorithms. The agent can guide you on algorithm choice in business terms.

6. Dataset Splitting for Training and Evaluation

Use the agent to split dataset as database views.

Example 1-7 Dataset Splitting for Training and Evaluation

Sample Prompts

- Split into train/validation/test using standard percentages.
- Create an 80/20 train/test split.
- Split data into train, validation and test sets, then find best model optimizing Accuracy.

Outputs

Here are some expected outputs for the above prompts:

- New views with suffixes such as `_TRAIN`, `_VAL` (if requested), `_TEST`
- Optional `_UNLABELED` view if a target column is provided and some rows have `NULL` targets. You can use this view later for inference.
- SQL code used to perform the split.

7. Model Training

Use Data Science Agent for model training, automated model selection (supervised model search), and model building (unsupervised learning).

Example 1-8 Supervised learning (classification and regression)

Here are some prompts to use Data Science Agent to train models to predict a categorical outcome (classification) or a numeric value (regression).

Sample Prompts

- Train a classifier to predict `SUBSCRIBED`.
- Train a regression model to predict `CALL_DURATION`.

Outputs

Here are some expected outputs for the above prompts:

- A trained OML mining model stored in the database
- A summary of the training run and configuration choices
- SQL code to replicate the training

Example 1-9 Automated model selection (supervised model search)

You can ask the agent to evaluate multiple supervised algorithms and pick the best model based on a metric.

Note

Automated model selection requires a validation set for comparing models against the selected metric.

After automated model selection, the winning model is retrained on combined train and validation dataset. Therefore, the final model is not the same object as the one that scored best during comparison.

Sample Prompts

- Find the best model for predicting `SUBSCRIBED` using F1 metric.
- Run an automated model search for churn prediction.

Outputs

Here are some expected outputs for the above prompts:

- A best-performing model selected using the chosen metric (on validation data).
- A report of validation performance across tested algorithms.
- Optionally, a results table containing the benchmark metrics.

Example 1-10 Unsupervised learning (Clustering and Anomaly Detection)

You can use Data Science Agent to build models without a labeled target.

Note

For unsupervised models, only model build is supported currently. Additional scoring capabilities and interpretations for clustering will be available soon.

Sample Prompts

- Segment customers into clusters.
- Detect anomalies in transaction behavior.

Outputs

Here are some expected outputs for the above prompts:

- A trained clustering or anomaly model stored in the database
- A summary describing how to use the model for downstream scoring

8. Compare Models and Select a Winner

When multiple models exist, either created by the agent or otherwise, you can request the agent for a comparison based on a common validation dataset.

Example 1-11 Model comparison**Sample Prompts**

- Compare these three models using AUC and select the best.
- Rank candidate models and store results in a table.

Outputs

Here are some expected outputs for the above prompts:

- A ranked comparison (best-to-worst) on the specified metric.
- Optional persistence of the full ranking into a results table for auditability.

9. Evaluate Models

Use Data Science Agent for an unbiased evaluation on held-out test data.

Note

Evaluation on held-out test set is intended as the most reliable estimate of generalization performance of a trained model.

Example 1-12 Model Evaluation**Sample Prompts**

- Evaluate the selected model on the test set.
- Provide Precision, Recall, F1 and a confusion matrix on test.
- Evaluate regression error on test.

- Evaluate the best model on the test set, then score the prospects dataset and return the highest-probability cases

Outputs

Here are some expected outputs for the above prompts:

- For Classification: accuracy-family metrics and confusion-matrix reporting (binary and multiclass supported)
- For Regression: fit and error metrics. For example, R^2 , MAE, RMSE.
- A test-results table stored in the database.
- SQL code to use the models in inference on arbitrary data.

10. Inference and Scoring

Once a model exists, you can request scoring on new records. Inference requires:

- A trained model
- A dataset containing the full feature set expected by the model
- A dataset containing the IDs to score

Note

Inference is supported only in ID-based scoring mode, that is IDs to score along with full feature dataset. Broader scoring options will be available soon.

Example 1-13 Inference and Scoring

Sample Prompts

- Score the prospects table and return the top 500 most likely to subscribe.
- Run inference for these customer IDs.

Outputs

Here are some expected outputs for the above prompts:

- Predictions returned in the UI, linked back to case IDs
- For Classification: predicted class and probability (based on the designated positive class)
- For Regression: predicted numeric value

You can also use the agent in interactive mode for suggestions and interpretations as well.

Examples:

- Suggestion request: "I want to predict clients most likely to subscribe, assist me in designing a suitable workflow"
- Interpretation request: "Can you help me interpret the model metrics so that I can better assess its performance?"

2

Get Started with Data Science Agent

This section discusses how to get started with Data Science Agent in Oracle Machine Learning UI.

Topics:

- [Access Data Science Agent](#)
You can access Data Science Agent directly from Oracle Machine Learning UI home page.
- [Create AI Credential and AI Profile](#)
An AI Credential is information that contains authentication details required for the database to access Oracle Cloud Infrastructure (OCI) resources on behalf of a user or application. The AI credential comprises information such as `user_ocid`, `tenancy_ocid`, `private_key` and `fingerprint`. An AI Profile is information about a user and their attributes, such as `provider`, `credential_name`, and `object_list`.
- [Manage Data Science Conversations](#)
The Data Science Agent Conversations page lists all the conversations you created. Here, you create and manage conversations.
- [Use Data Science Agent Chat Interface](#)
On the Data Science Agent chat interface, you interact with the agent for all machine learning and data science related questions.
- [Example of a Conversation with Data Science Agent](#)
This example demonstrates how Data Science Agent supports a novice user in exploring the dataset present in the SH schema, and in building and evaluating a machine learning model. It also shows the effectiveness of Natural Language to SQL (NL2SQL) in Data Science Agent.

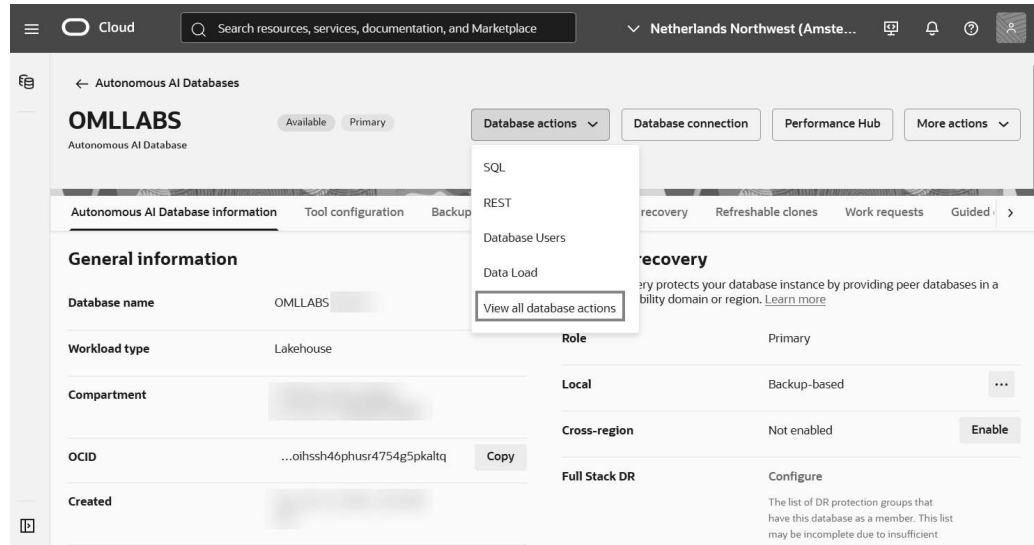
2.1 Access Data Science Agent

You can access Data Science Agent directly from Oracle Machine Learning UI home page.

To access Data Science Agent, you must first sign into Oracle Machine Learning from Autonomous AI Database:

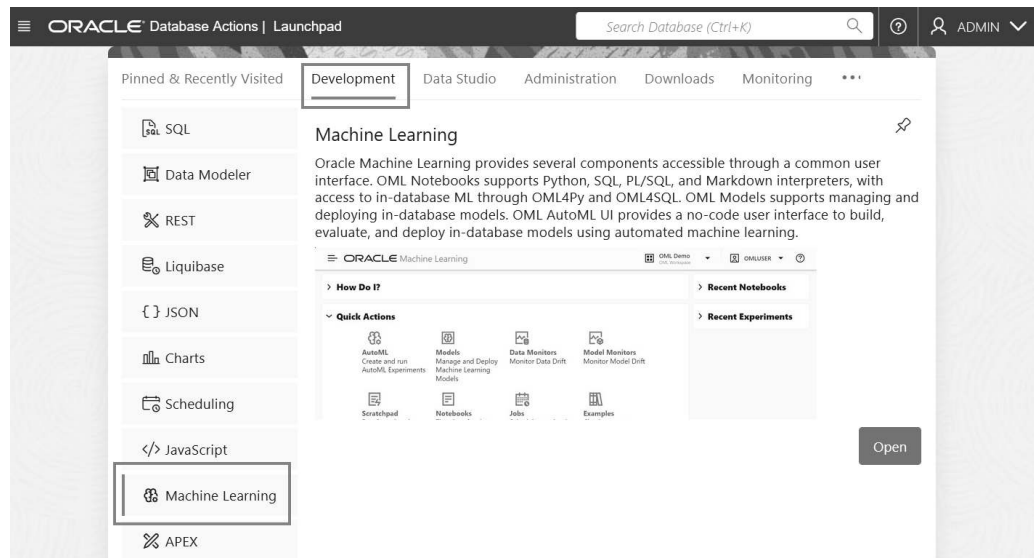
1. To sign into Oracle Machine Learning from an Autonomous AI Database instance:
 - a. On the Autonomous AI Database information page click **Database actions** and then click **View all database actions**.

Figure 2-1 Database Actions



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

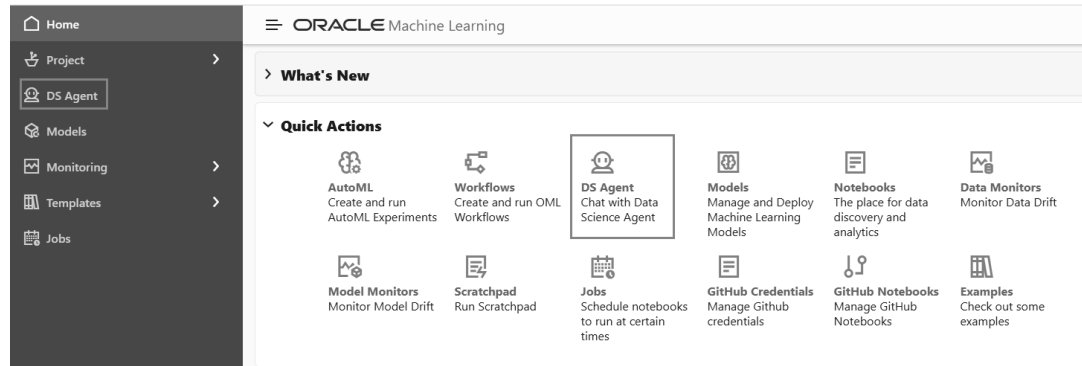
Figure 2-2 Oracle Machine Learning option in Database Actions Launchpad




The Oracle Machine Learning UI sign in page opens.

- c. Enter your username and password, and click **Sign in**.
This opens the Oracle Machine Learning UI home page.
2. On your Oracle Machine Learning UI home page, click **DS Agent**..

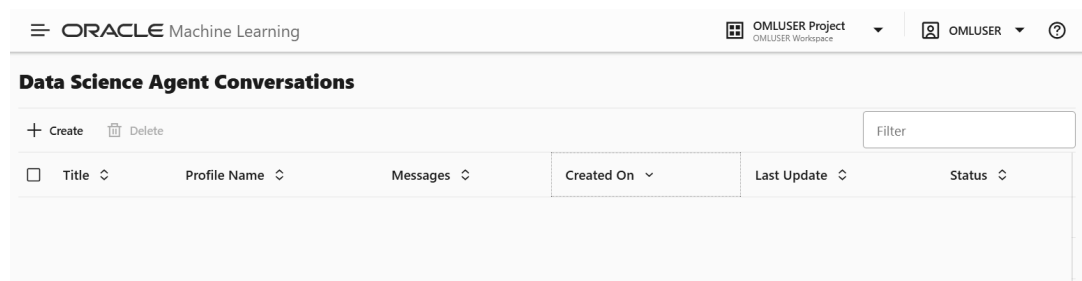
Figure 2-3 Oracle Machine Learning UI homepage



Alternatively, you can click the Cloud menu icon  to open on the left navigation menu. Here, click **DS Agent**.

This opens the Data Science Conversations listing page.

Figure 2-4 Data Science Agent Conversations listing page



2.2 Create AI Credential and AI Profile

An AI Credential is information that contains authentication details required for the database to access Oracle Cloud Infrastructure (OCI) resources on behalf of a user or application. The AI credential comprises information such as `user_ocid`, `tenancy_ocid`, `private_key` and `fingerprint`. An AI Profile is information about a user and their attributes, such as `provider`, `credential_name`, and `object_list`.

Prerequisites

- `DBMS_CLOUD_AI` package
- `user_ocid`
- `tenancy_ocid`
- `private_key`
- `fingerprint`

You can create and manage your AI profiles using the `DBMS_CLOUD_AI` package.

To create an AI credential and AI Profile:

1. Create a notebook and in a `%script` paragraph, run the following command to create an AI credential:

```
%script
DECLARE
    credential_name VARCHAR2(128) := 'OCI_CRED';
BEGIN
    BEGIN
        dbms_cloud.drop_credential(credential_name => credential_name);
    EXCEPTION
        WHEN OTHERS THEN
            NULL;
    END;
    dbms_cloud.create_credential(
        credential_name => credential_name,
        user_ocid       => '<ocid1.user.oc1..>',
        tenancy_ocid    => '<ocid1.tenancy.oc1..>',
        private_key     => '<private_key>',
        fingerprint     => '<fingerprint>'
    );
END;
/
```

This PL/SQL script calls the `create_credential` procedure of `DBMS_CLOUD_AI` package to create a new credential with the given parameters:

- `credential_name`: Name of the credential. In this example, the credential name is `OCI_CRED`.
 - `user_ocid`: This is the Oracle Cloud Identifier, a unique ID for the user. See [Where to Get the Tenancy's OCID and User's OCID](#) for details.
 - `tenancy_ocid`: This is the Oracle Cloud Identifier for your tenancy (your OCI account). See [Where to Get the Tenancy's OCID and User's OCID](#) for details.
 - `private_key`: Specify the generated private key. Private keys generated with a passphrase are not supported. You must generate the private key without a passphrase. See [How to Generate an API Signing Key](#) for details.
 - `fingerprint`: Specify the fingerprint. After a generated public key is uploaded to the user's account the fingerprint is displayed in the console. Use the displayed fingerprint for this argument. See [How to Get the Key's Fingerprint](#) and [How to Generate an API Signing Key](#) for more information.
2. In another `%script` paragraph in the same notebook, run the following command to create an AI profile by the name `NL2SQL`.

```
%script
DECLARE
    profile_name VARCHAR2(128) := 'NL2SQL';
BEGIN
    dbms_cloud_ai.drop_profile(
        profile_name,
        TRUE
    );
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes   => '{
```

```

        "comments": false,
        "conversation": true,
        "credential_name": "OCI_CRED",
        "model": "openai.gpt-4.1",
        "provider": "oci",
        "temperature": 1,
        "max_tokens": 4096,
        "oci_compartment_id": "<ocidl.compartment.ocl.>",
        "oci_apiformat": "GENERIC"
    },
);
END;
/

```

Define the following attributes for this profile:

- `profile_name`: A name for the AI profile. The profile name must follow the naming rules of Oracle SQL identifier. Maximum length of profile name is 125 characters.
- `comments`: Set this to `false`. Setting it to `false` <placeholder>.
- `conversation`: A `VARCHAR2` attribute that indicates if conversation history is enabled for a profile. Allowed values are `true` or `false`. The default value is `false`. The values are not case sensitive.
- `credential_name`: This is the AI credential name. name of the Oracle Cloud Infrastructure (OCI) credential object for authenticating requests.
- `model`: The name of the AI model being used to generate responses in the conversation. In this example, it is OpenAI's GPT-4.1. For more information, see [Recommended Models](#).
- `provider`: This is the provider of the model. It is a mandatory field. Supported providers are:
 - openai
 - cohere
 - azure
 - database
 - oci
 - google
 - anthropic
 - huggingface
 - aws
- `max_tokens`: Specify the maximum number of tokens (words and pieces of words) in the response. Prevents overly long outputs and manages cost.
- `oci_compartment_id`: This is the OCID of the compartment you are permitted to access when calling the OCI Generative AI service. The compartment ID can contain alphanumeric characters, hyphens and dots.
- `oci_apiformat`: This is the format in which the API expects data to be sent and received. Use this attribute to generate text responses. This attribute applies to OCI Generative AI Chat Models in a dedicated AI cluster. Specify this attribute when you

specify a model OCID in the model attribute or provide an endpoint in the `oci_endpoint_id` attribute. Supported values are:

- COHERE
- GENERIC

3. Check the status of the profile creation by running the following:

```
%sql select * from
      user_cloud_ai_profiles;
```

Figure 2-5 AI Profile

PROFILE_ID	PROFILE_NAME	STATUS	DESCRIPTION	CREATED	LAST_MODIFIED
1	OCI_GPT_4_1_OMLUSER	ENABLED	CLOB	2026-01-13 15:03:42.851397 UTC	2026-01-13 15:03:42.851397 UTC
7	NL2SQL	ENABLED	CLOB	2026-02-05 12:50:44.560307 UTC	2026-02-05 12:50:44.560307 UTC

This completes the task of creating an AI credential and AI Profile.

2.3 Manage Data Science Conversations

The Data Science Agent Conversations page lists all the conversations you created. Here, you create and manage conversations.

Create your Data Science Agent conversation to interact with the agent on these areas of data science and machine learning:

- Data profiling
- Data wrangling and transformation
- Statistical analysis of variable relationships
- Feature Importance, Classification, Regression, XGBoost, Clustering, and Anomaly Detection.
- Model training and evaluation
- Inference on new data

Here is the Data Science Conversations page. It lists all the conversations that you created. Click on a conversation title to open it and resume it.

Figure 2-6 Data Science Conversation listing page

<input type="checkbox"/>	Title	Profile Name	Messages	Created On	Last Update	Status
<input type="checkbox"/>	SQL	NL2SQL	2	2/5/2026, 8:03 PM	2/5/2026, 8:06 PM	Idle

This page lists the following details about the conversations:

- **Title:** This is the name of the conversation you provided while creating the conversation.
- **Profile Name:** This is the AI Profile you selected while creating the conversation. An AI Profile contains information about the user and their attributes, such as the `provider`, `credential_name`, and `object_list`. You can create and manage your AI profiles through `DBMS_CLOUD_AI` package.
- **Messages:** This indicates the number of interactions in the conversation with the agent. As shown in the screenshot, the conversation titled `SQL` has 2 messages or interactions as on the last updated date.
- **Created on:** This is the date on which the conversation was first created.
- **Last Updated:** This is the date on which the conversation was last used or updated.
- **Status:** There are two statuses ACTIVE and IDLE.

You can perform the following tasks here:

- [Create a Data Science Agent Conversation](#)
A conversation is a set of interactions with Data Science Agent in the chat interface. Before you start a conversation with the Data Science Agent, you must create a conversation.
- [Delete a Data Science Agent Conversation](#)
You can delete a conversation from the Data Science Agent Conversations listing page.

2.3.1 Create a Data Science Agent Conversation

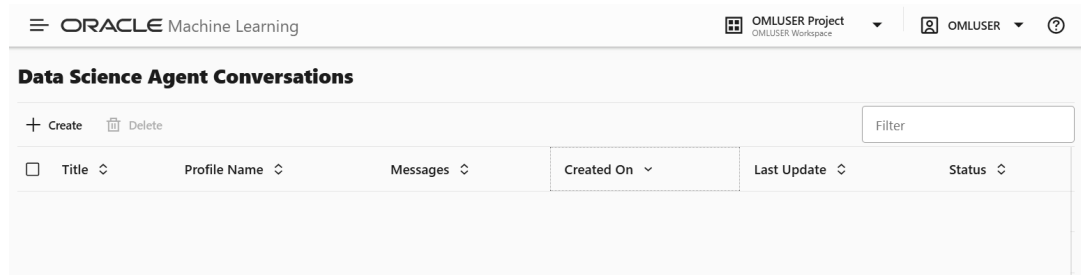
A conversation is a set of interactions with Data Science Agent in the chat interface. Before you start a conversation with the Data Science Agent, you must create a conversation.

Note

You can use the same Database user credentials to access the same conversation in multiple browsers. However, Oracle does not recommend this as it may lead to unexpected behavior. If you attempt this, Data Science Agent will display a warning, but you will have the option to override it.

To create a Data Science Agent conversation:

1. On the Data Science Agent Conversation page, click **Create**.

Figure 2-7 Data Science Agent Conversations listing page

This opens the Create DS Agent Conversation dialog.

2. In the Create DS Agent Conversation dialog, enter the following details:

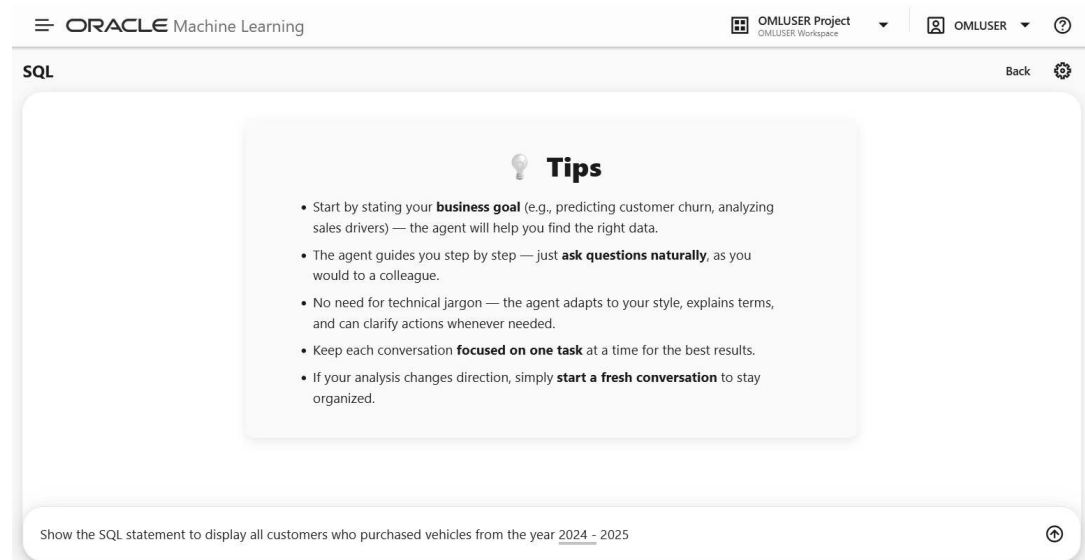
Figure 2-8 Create Conversation dialog

3. In the **Title** field, provide a name for your conversation. In this example, create a conversation by the name SQL.
4. In the **AI Profile** drop-down menu, click on the down arrow and select a profile. Select the profile **NL2SQL**.
5. Click **OK**.

The conversation is created and it opens in the Data Science Agent chat interface. Here, you can start chatting with Data Science Agent.

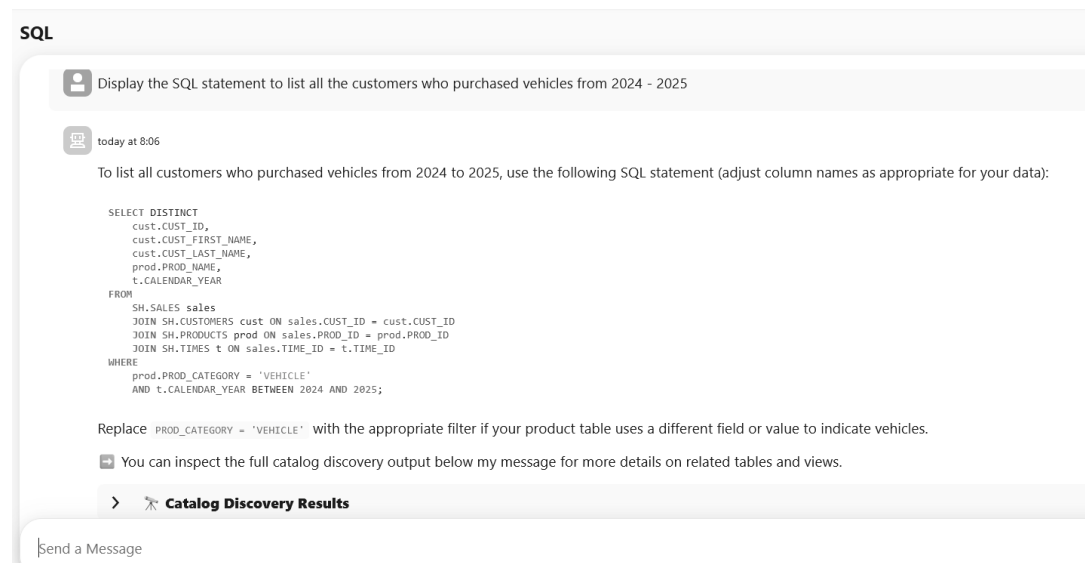
6. In the chat interface, Data Science presents you with tips to begin your conversation. In the **Send a message** field, type in your prompt in natural language and press enter. In this example, type `Display the SQL statement to list all customers who purchased vehicles from 2024 - 2025` and press enter.

Figure 2-9 Chat with Data Science Agent



- The agent may take some time to respond. Here is a response from Data Science Agent to the prompt you entered in the previous step. Continue your conversation with the agent. You may also choose to accept the suggestions provided by the agent.

Figure 2-10 Response of Data Science Agent



2.3.2 Delete a Data Science Agent Conversation

You can delete a conversation from the Data Science Agent Conversations listing page.

To delete a conversation:

- On the Data Science Agent Conversations listing page, select the conversation you want to delete.
- Click **Delete**.

2.4 Use Data Science Agent Chat Interface

On the Data Science Agent chat interface, you interact with the agent for all machine learning and data science related questions.

To begin a chat with Data Science Agent:

1. On the Data Science Agent chat interface, type your question in the **Ask a question** field, and press enter.

For more information, see [Data Science Agent: Sample Prompts and Outputs](#) and [Associate Database objects to your conversation](#).

2. On the Settings pane, you can define database service levels and manually associate


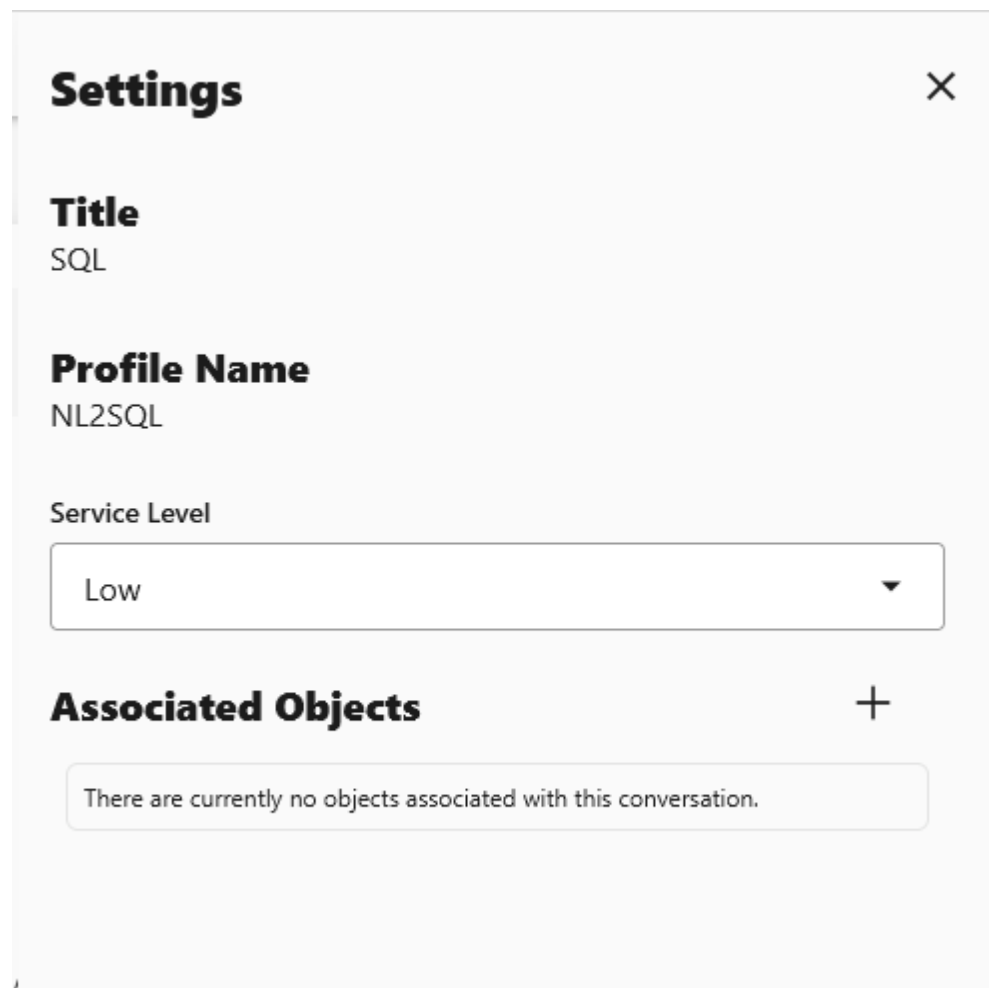
database objects to the conversation. Click  on the top right corner to open the Settings pane.

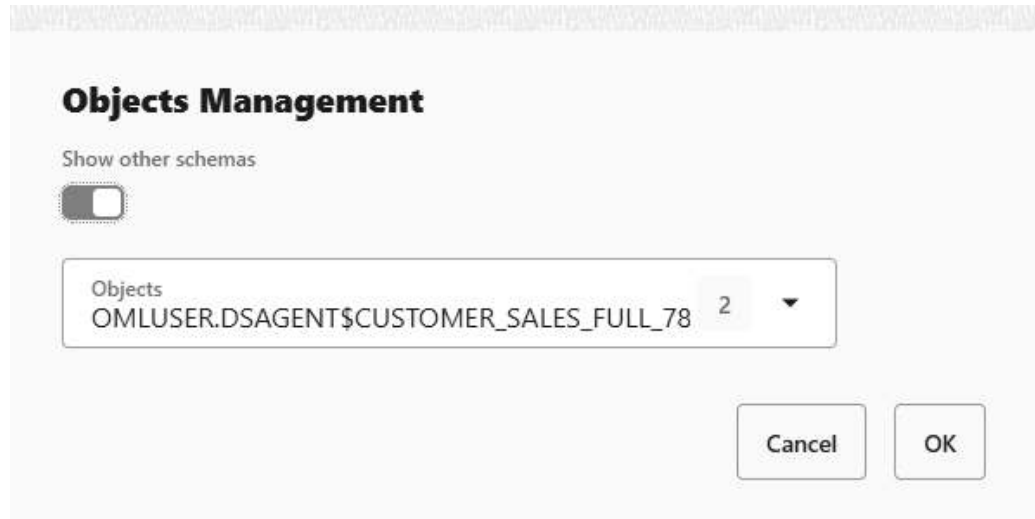
Figure 2-11 Data Science Agent settings



3. In the **Service Level** field, click the drop-down menu to select a service level. By default, the service level is set to **Low**. If you are working with large dataset, consider changing the Service Level to **Medium** or **High** to achieve parallelism in the database backend.

4. To improve the precision and efficiency of the responses of the agent, associate database objects to the conversation. In the **Associated Objects** field, click + to open the **Object Management** dialog. In the Object Management dialog:
 - a. Turn on **Show other schemas** to view objects in schemas besides your own. The object are tables, views, and machine learning models.

Figure 2-12 Object Management dialog



- b. Select the objects from the drop-down list. You can select multiple objects to the conversation.
 - c. Click **OK** to associate objects to the conversation.
5. The associated objects are listed below the **Associated Objects** field on the Settings pane. To remove any associated objects, click the delete icon against the object.

Figure 2-13 Data Science Agent settings

Settings ✕









Title
SQL

Profile Name
NL2SQL

Service Level

Low ▼

Associated Objects +

	OMLUSER.DSAGENT\$CUSTOMER_SALES_FULL_78 21 View	
	OMLUSER.DSAGENT\$CUST_VALUE_EXTENDED_7 19B View	
	OMLUSER.DSAGENT\$SALES_BY_YEAR_98C2 View	
	OMLUSER.DSAGENT\$SALES_BY_YEAR_98C2_UNL ABELED View	

6. Data Science Agent provides the responses in the text area. You can continue your conversation by asking further questions.
7. To exit the conversation page, click **Back** on the top right corner of the page. This takes you back to the Data Science Agent Conversations listing page.

2.5 Example of a Conversation with Data Science Agent

This example demonstrates how Data Science Agent supports a novice user in exploring the dataset present in the SH schema, and in building and evaluating a machine learning model. It also shows the effectiveness of Natural Language to SQL (NL2SQL) in Data Science Agent.

The user runs a complete machine learning workflow using natural language. The conversation between the user and Data Science Agent progresses from generic questions on data

exploration in the `SH` schema to specific ones on model training, model building and scoring. Broadly, the conversation flow can be categorized into these sections:

- User sets a goal for the conversation
- Data Science Agent explains the dataset in the `SH` schema
- Data Science Agent prepares data for model building
- Data Science Agent trains and builds a model to predict customer income level
- Data Science Agent computes model accuracy
- Data Science Agent interprets the confusion matrix
- Data Science Agent uses the trained model to predict customer income levels on new data

Highlights:

This example highlights the following capabilities of Data Science Agent:

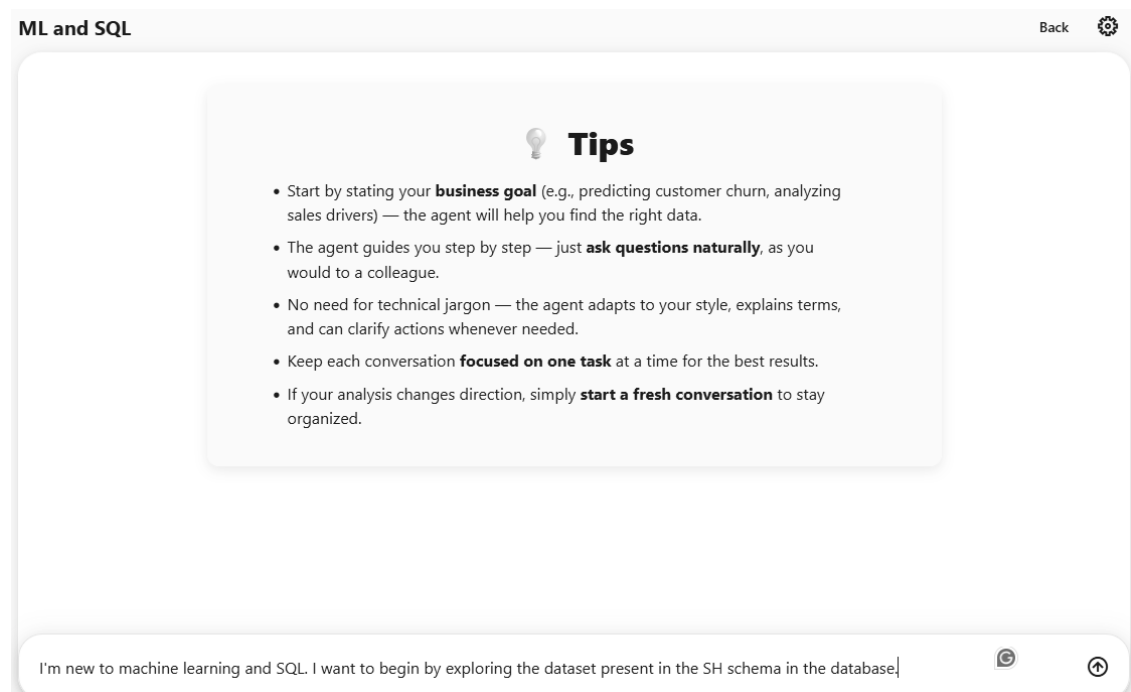
- Efficiency in performing machine learning tasks—data preparation, model building, model training, model evaluation, data analysis and so on
- Natural language to SQL capabilities
- Visualization capabilities
- Intelligence to suggest the next steps and options to the user

1. User sets the context

For every new conversation, Data Science Agent opens the chat interface with some important tips for the user. The user begins to interact with the agent with a prompt that sets the goal and context for the conversation.

Prompt 1: I'm new to machine learning and SQL. I want to begin by exploring the dataset present in the `SH` schema in the database.

Figure 2-14 Introduction

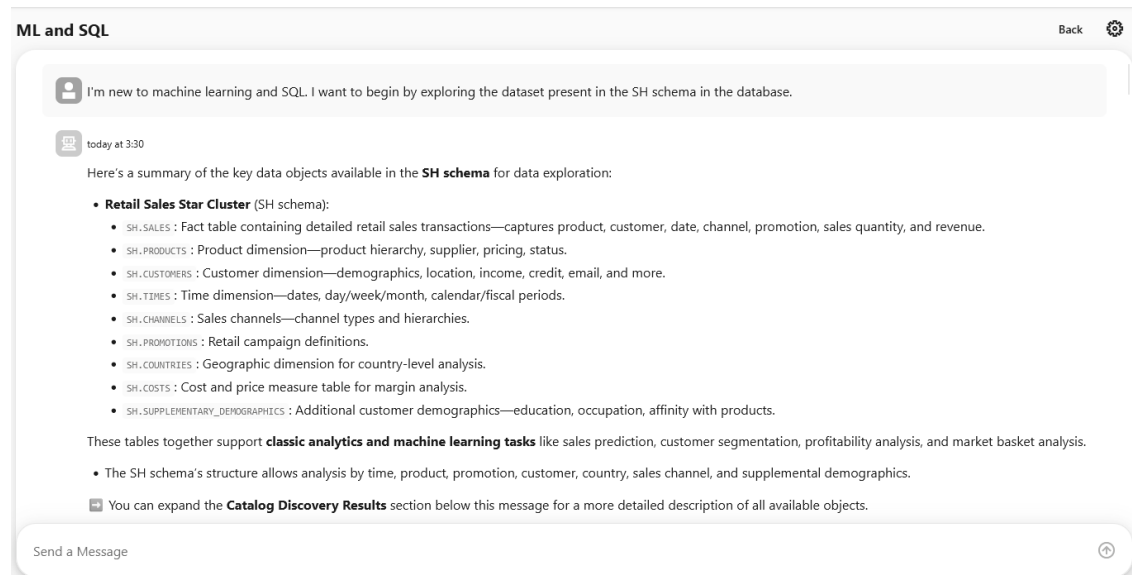


Agent's response:

Data Science Agent responds with the following information:

- A summary of each table present in the SH schema.
- A detailed description of all available objects in the **Catalog Discovery Results** section.
- Provides options to explore specific business area (like sales, customers, or products), or an overview with basic statistics on a few key tables. It asks in natural language about the user's preference and mentions that it'll provide interactive guidance.

Figure 2-15 User sets the goal of the conversation

**Catalog Discovery Results**

The agent also provides a detailed description of all available objects in the SH schema in the **Catalog Discovery Results**, a separate section in the same response. The user clicks to expand it and view the details, as shown in the screenshot below:

Figure 2-16 Catalog Discovery Results

The screenshot shows a chat interface titled "ML and SQL". At the top right, there is a "Back" button and a settings icon. Below the title, a message states: "You can expand the **Catalog Discovery Results** section below this message for a more detailed description of all available objects." A "Next step" box asks: "Would you like to start by exploring a specific business area (like sales, customers, or products), or would you prefer an overview with basic statistics on a few key tables? Let me know your preference and I'll guide you interactively through the next steps!" Below this is a section titled "Catalog Discovery Results" with a dropdown arrow. Underneath, a cluster is identified as "Cluster: Retail Sales Star". A description follows: "Analytics of sales transactions with detailed context from dimensions; supports analysis by product, customer, time, channel, and promotion." Two tables are listed:

- SH_SALES**
 - Type: Table
 - Purpose: Fact table of retail sales transactions.
 - Main attributes: Product, customer, date, channel, promotion (identifiers); sales quantity and revenue (measures).
 - Notes: No PK; row uniqueness is composite on all FKs; intentionally missing FK for PROMO_ID.
 - Joins: Product (PROD_ID) → PRODUCTS, Customer (CUST_ID) → CUSTOMERS, Date (TIME_ID) → TIMES, Channel (CHANNEL_ID) → CHANNELS, Promotion (PROMO_ID) → PROMOTIONS (outer join).
- SH_PRODUCTS**
 - Type: Table
 - Purpose: Product dimension table with hierarchies and characteristics.
 - Main attributes: Product, subcategory, category (identifiers/hierarchies); supplier; pricing; status; validity dates.

At the bottom left, there is a "Send a Message" button, and at the bottom right, there is a circular icon with a right-pointing arrow.

2. Data Science Agent explores dataset in the SH schema

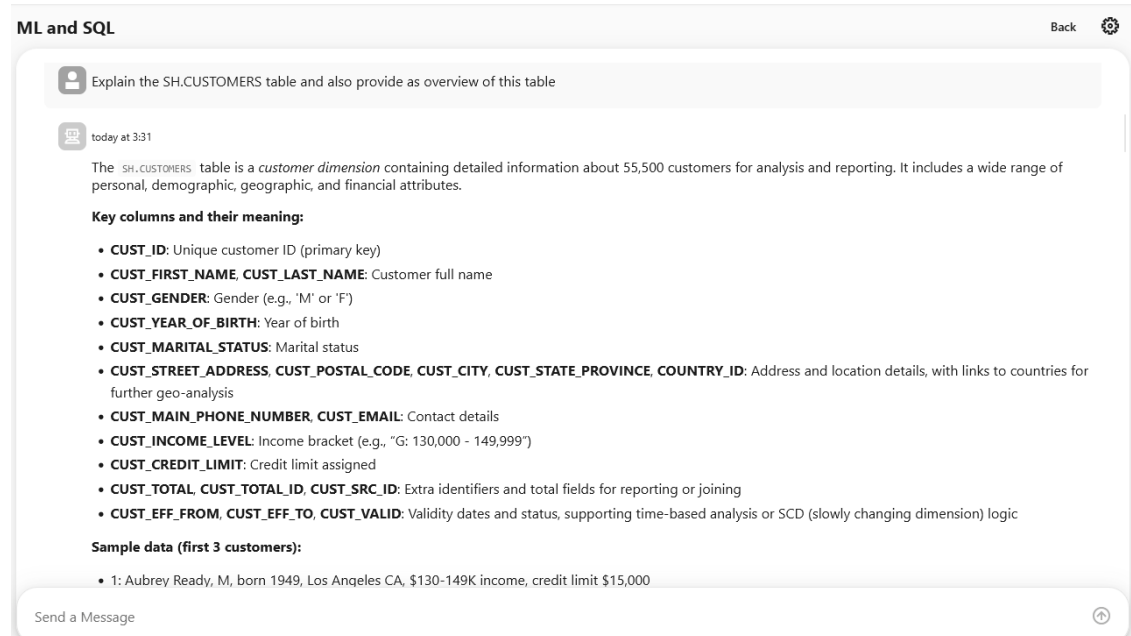
Conversation progression: The user opts to explore a particular table in the SH schema.

Prompt 2: Explain the SH.CUSTOMERS table and also provide an overview of this table.

Agent's response: Data Science Agent responds to this prompt in a structured format. It provides the following information:

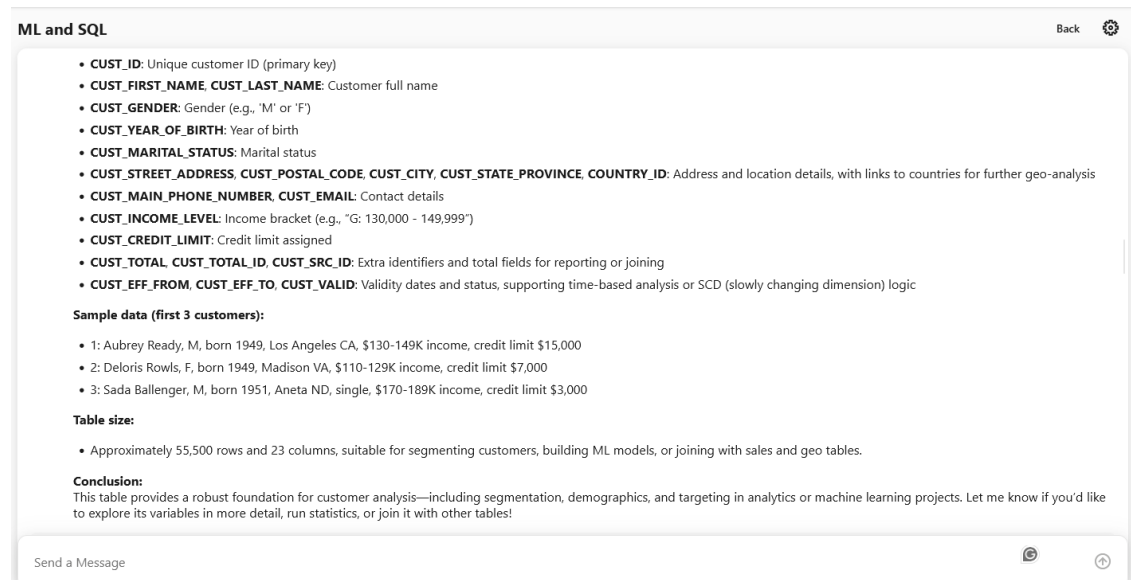
- A brief summary of the table
- Lists all the columns in the SH.CUSTOMERS table alongwith their meaning
- Sample data
- Table size
- Conclusion and options to continue the conversation.

Figure 2-17 Data Science Agent explains the CUSTOMERS table



This image shows the sample data, table size and a conclusion. This is in continuation to the same response to prompt 2. The agent also asks how the user would like to proceed from here—explore variables in more details, run statistics, or join the CUSTOMERS table with other tables.

Figure 2-18 Data Science Agent explains the CUSTOMERS table (Contd.)



3. Data Science Agent provides Extended Analysis and Statistics

Conversation progression: The user now chooses to dive deeper into some important statistics. The user provides the third prompt.

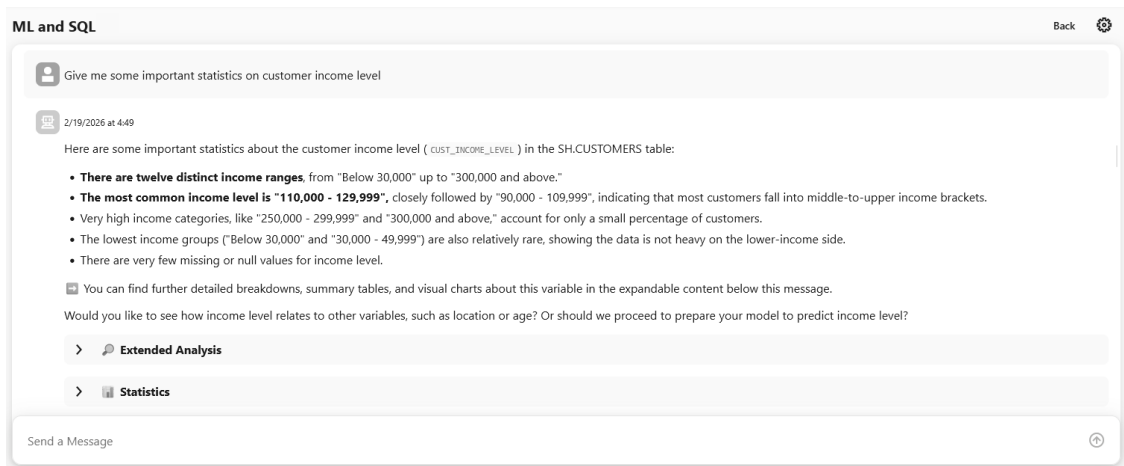
Prompt 3: Give me some important statistics on customer income level.

Agent's response:

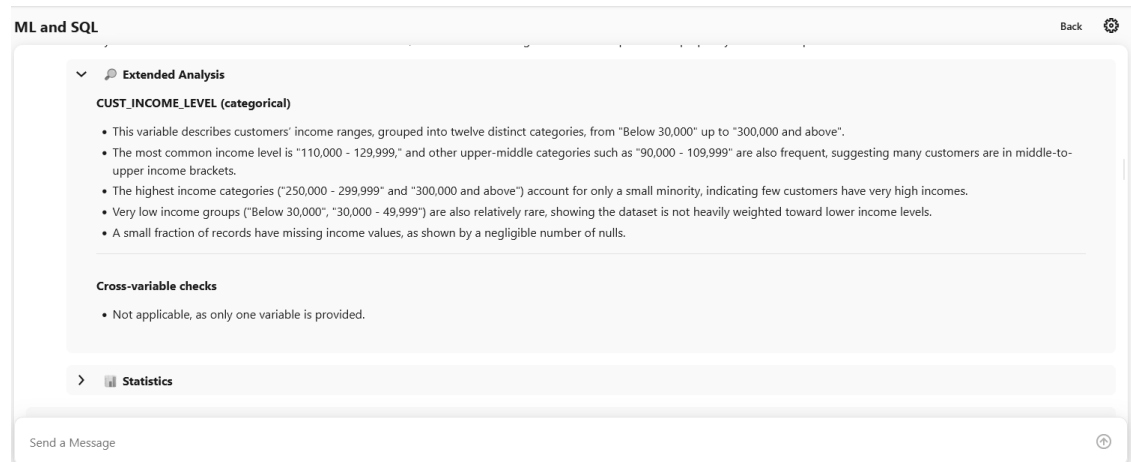
Data Science Agent analyzes data related to customer income level and provides detailed statistical information on customer income level based on the `SH.CUSTOMERS` table.

- Related to customer income level, it identifies the following categories:
 - Twelve distinct income ranges
 - The most common income level
 - Very high income categories
 - Lowest income groups
 - Missing values for income level
- Presents an extended analysis customer income level
- Presents a graphical representation of customer income level

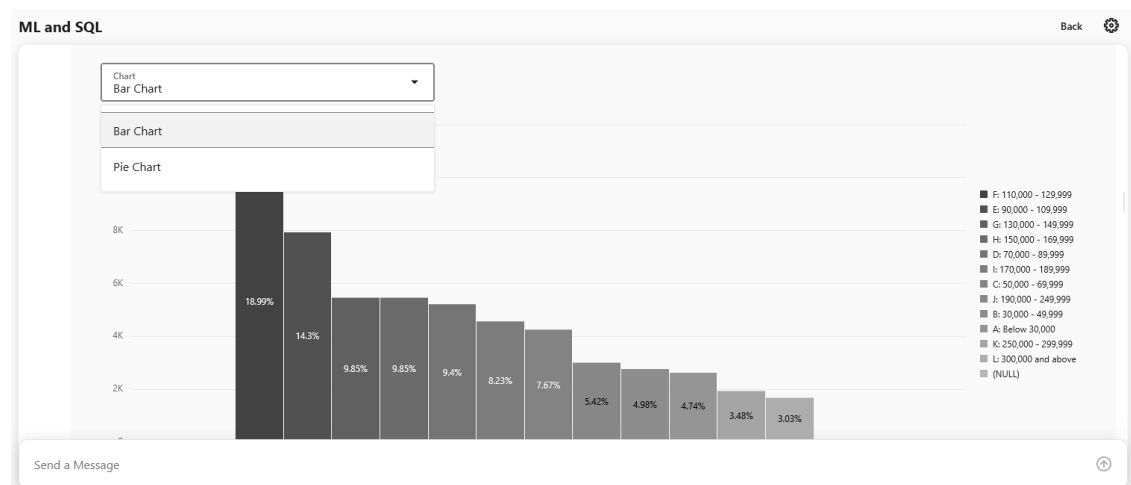
Figure 2-19 Data Science Agent provides statistical information on customer income level in the `SH.CUSTOMERS` table



Here is a screenshot of the extended analysis of customer income level in the `SH.CUSTOMERS` table.

Figure 2-20 Extended Analysis of customer income level

Here is a screenshot of the graphical representation of customer income level in the SH.CUSTOMERS table.

Figure 2-21 Graphical representation of customer income level

4. Data Science Agent prepares data

Conversation progression: After reviewing the dataset and exploring the customer income level, the user now proceeds to train a machine learning model with the help of Data Science Agent. The user wants to predict the customer income level.

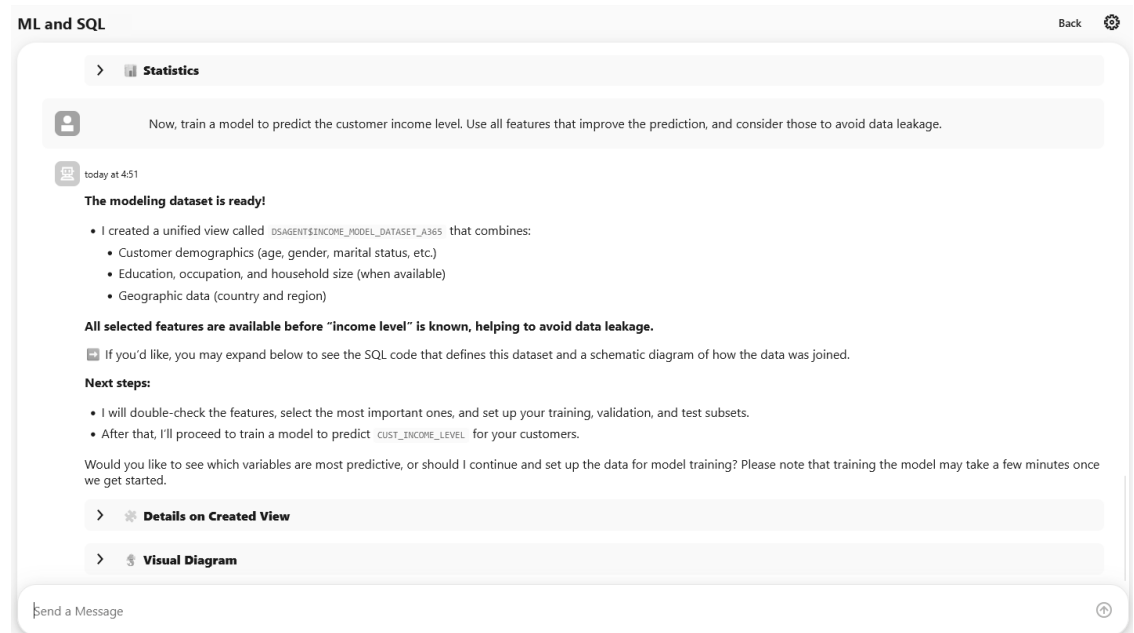
Prompt 4: Now, train a model to predict the customer income level. Use all features that improve the prediction, and consider those to avoid data leakage.

Agent's response:

- Data Science Agent creates a view `DSAGENT$INCOME_MODEL_DATASET_A365` combining the following tables in the SH schema— `SH.CUSTOMERS`, `SH.SUPPLEMENTARY_DEMOGRAPHICS`, and `SH.COUNTRIES`.

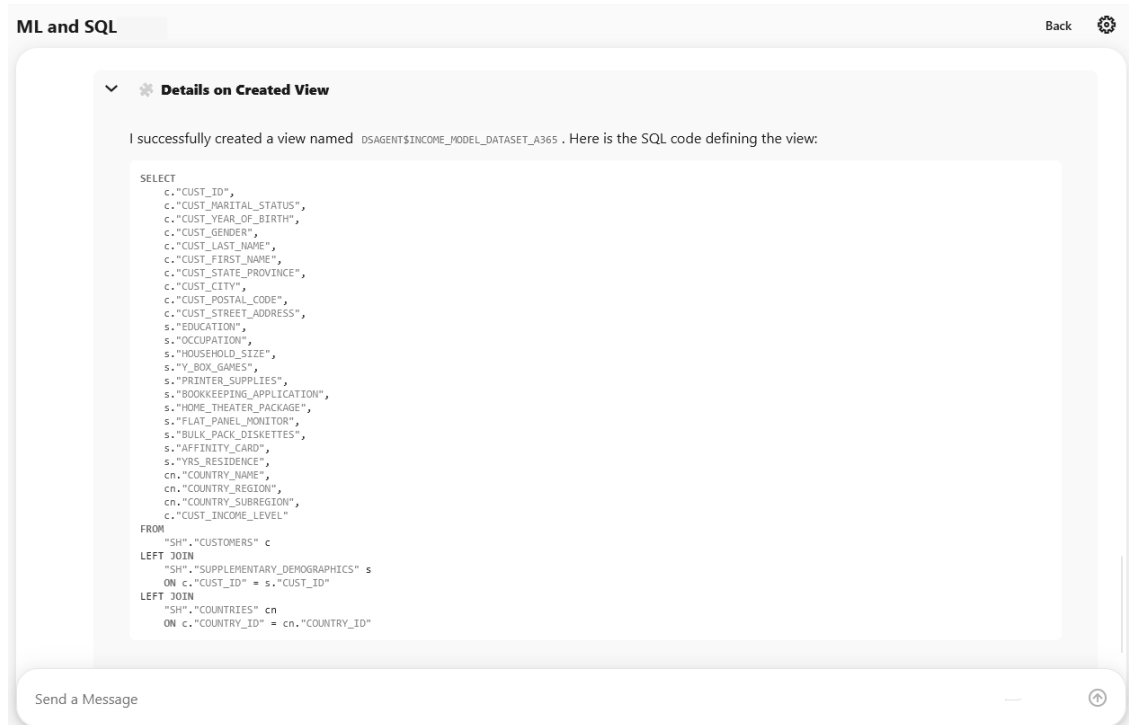
- Provides the next steps
- Provides the SQL script that it used to create the view `DSAGENT$INCOME_MODEL_DATASET_A365`
- Provides a visual diagram to depict the workflow to create the view.

Figure 2-22 Data Science Agent prepares data for model building



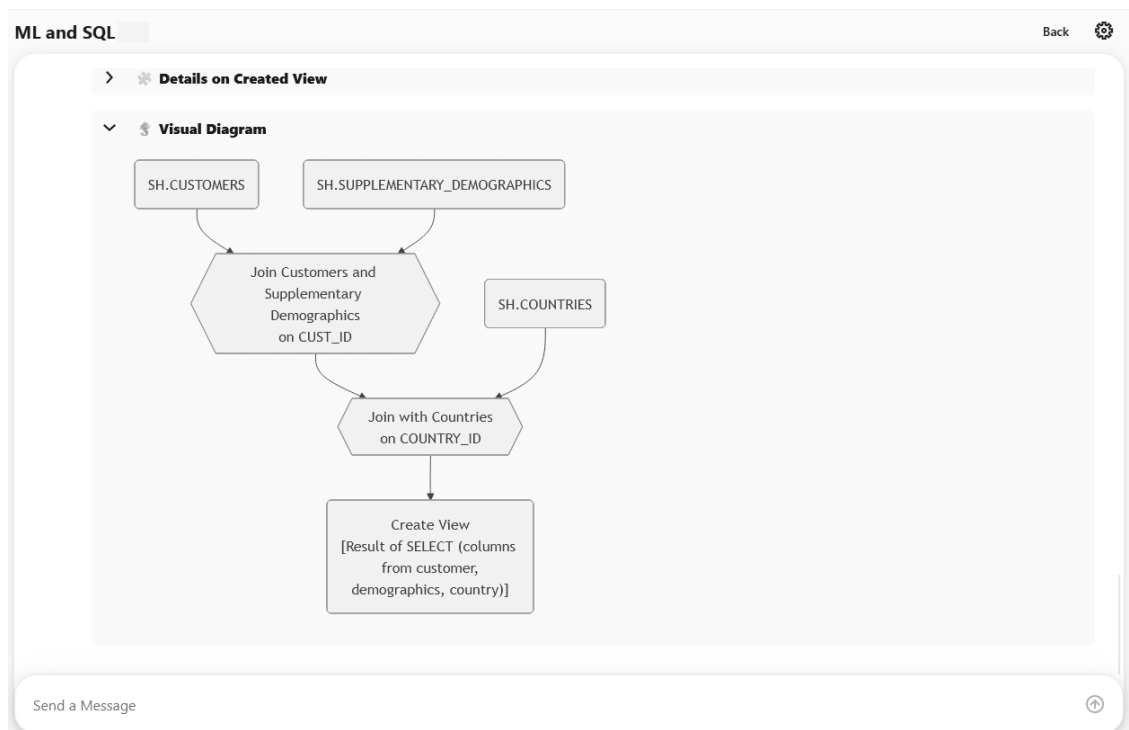
Here's the SQL code developed by Data Science Agent to define the view `DSAGENT$INCOME_MODEL_DATASET_A365`.

Figure 2-23 Data Science Agent provides the SQL code for defining the view



Data Science Agent also presents a visual representation of the sources (tables) from where it pulled the data to create the view DSAGENT\$INCOME_MODEL_DATASET_A365.

Figure 2-24 Visual representation of how Data Science Agent created the view



5. Data Science Agent splits data

Conversation progression: The user now proceeds to prepare the data to predict customer income level. The target column is `CUST_INCOME_LEVEL` in the view `DSAGENT$INCOME_MODEL_DATASET_A365`.

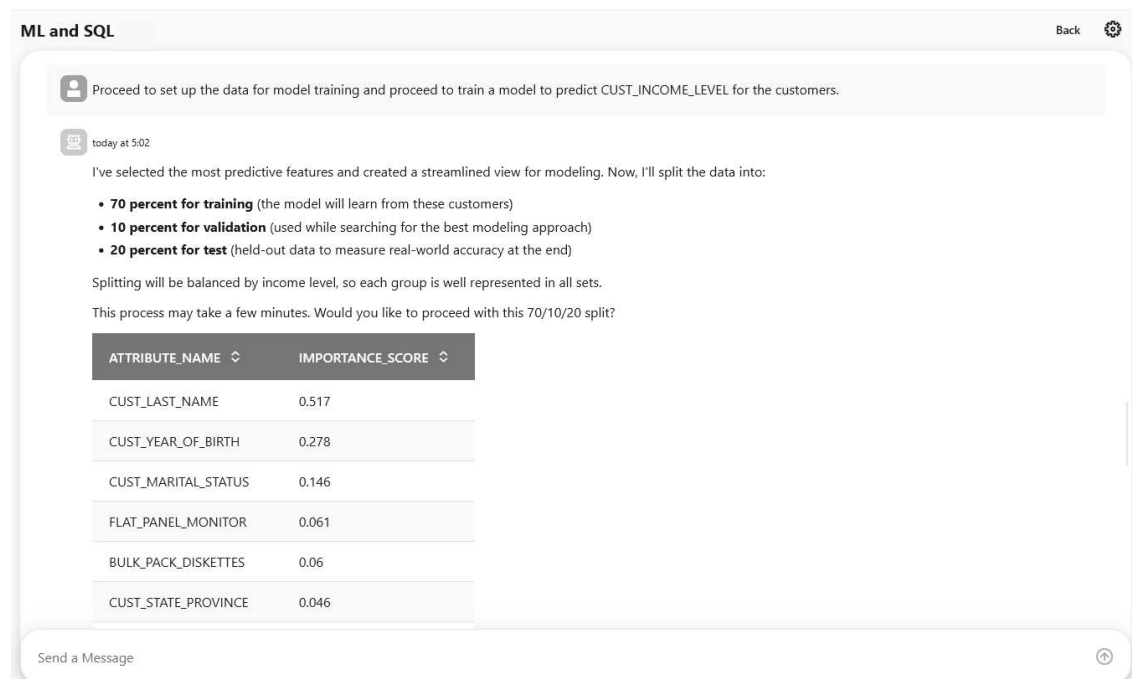
Prompt 5: Proceed to set up the data for model training and to train a model to predict `CUST_INCOME_LEVEL` for the customers.

Agent's response: Data Science Agent now splits the data for training, testing and validation:

- Presents information about the data split along with a brief explanation about how this data would be used.
 - Training dataset 70%
 - Validation dataset 10%
 - Testing dataset 20%
- How the data split is balanced by income level
- Asks if the user would like to proceed with the 70-10-20 data split?

Here is a screenshot of the data split done by Data Science Agent.

Figure 2-25 Data Split



Prompt 6: Yes. Go ahead with this 70/10/20 split.

Agent's response: Data Science Agent performs data split and presents the following information to the user:

- Overview of the data split.
- Presented detailed information about how the split was split along with the SQL code in an expandable section.

Figure 2-26 Data Split - Overview

ML and SQL Back

Yes. Go ahead with this 70/10/20 split

today at 5:03

The data is now split for machine learning:

- **Training set:** 70% of customers
- **Validation set:** 10% of customers
- **Test set:** 20% of customers (for fair assessment after training)
- **Unlabeled set:** A very small group with missing income levels (these won't be used for training)

All splits are balanced by income group for best reliability during model evaluation.

You can view details about how the data was split and the SQL code in the expandable section below.

Next, I'll train a model to predict the customer income level using the training and validation sets. This process may take a few minutes, as several modeling approaches will be tested to find the best one. Do you want me to proceed with automated model selection and training?

[> Details on Split](#)

Agent's response:

Data Science Agent created splits from OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL, created the following views, and also presented the corresponding SQL code for each view.

- train set: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_TRAIN
- validation set: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_VAL
- test set: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_TEST
- unlabeled: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_UNLABELED

Here's a screenshot of the data split performed by Data Science Agent.

Figure 2-27 Data Split - Details

ML and SQL Back

Created splits from OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL using stratified (classification) with key CUST_ID (seed = 42).

Views created:

- train set: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_TRAIN
- validation set: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_VAL
- test set: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_TEST
- unlabeled: OMLUSER.DSAGENT\$INCOME_MODEL_DATASET_A365_TOP_MDL_UNLABELED

Split	Rows	% of labeled
Train	38,815	69.99%
Validation	5,548	10.00%
Test	11,096	20.01%
Labeled total	55,459	100%
Unlabeled total (excluded)	41	-
Overall total	55,500	-

```
CREATE OR REPLACE VIEW "DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL_TRAIN"
AS WITH ranked AS (
  SELECT t."CUST_ID" AS id_key,
         COUNT(*) OVER (PARTITION BY t."CUST_INCOME_LEVEL") AS class_n,
         ROW_NUMBER() OVER (
           PARTITION BY t."CUST_INCOME_LEVEL"
           ORDER BY ORA_HASH(t."CUST_ID", 4294967295, 42), t."CUST_ID"
         ) AS rn
FROM "OMLUSER"."DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL" t
```

Send a Message

Here's a screenshot of the SQL code for each of the views created by Data Science Agent.

Figure 2-28 Data Split - SQL code for each view

The screenshot shows a chat window titled "ML and SQL" with a "Back" button and a settings icon. The chat contains three SQL queries for creating views. The first query is for the training set, the second for the validation set, and the third for the test set. Each query uses Oracle SQL syntax with window functions like RANK and COUNT to partition data by income level and rank within each partition. The training set query filters for ranks less than or equal to 7, the validation set for ranks greater than 7 and less than or equal to 8, and the test set for ranks greater than 8.

```

CREATE OR REPLACE VIEW "DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL_TRAIN"
AS WITH ranked AS (
  SELECT t."CUST_ID" AS id_key,
         COUNT(*) OVER (PARTITION BY t."CUST_INCOME_LEVEL") AS class_n,
         RANK_NUMBER() OVER (
           PARTITION BY t."CUST_INCOME_LEVEL"
           ORDER BY ORA_HASH(t."CUST_ID", 4294967295, 42), t."CUST_ID"
         ) AS rn
  FROM "OHLUSER"."DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL" t
  WHERE t."CUST_INCOME_LEVEL" IS NOT NULL
)
SELECT t.*
FROM "OHLUSER"."DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL" t
JOIN ranked r ON r.id_key = t."CUST_ID"
WHERE rn <= GREATEST(1, FLOOR(class_n * .7))

validation set:

CREATE OR REPLACE VIEW "DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL_VAL"
AS WITH ranked AS (
  SELECT t."CUST_ID" AS id_key,
         COUNT(*) OVER (PARTITION BY t."CUST_INCOME_LEVEL") AS class_n,
         RANK_NUMBER() OVER (
           PARTITION BY t."CUST_INCOME_LEVEL"
           ORDER BY ORA_HASH(t."CUST_ID", 4294967295, 42), t."CUST_ID"
         ) AS rn
  FROM "OHLUSER"."DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL" t
  WHERE t."CUST_INCOME_LEVEL" IS NOT NULL
)
SELECT t.*
FROM "OHLUSER"."DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL" t
JOIN ranked r ON r.id_key = t."CUST_ID"
WHERE rn > GREATEST(1, FLOOR(class_n * .7))
AND rn <= GREATEST(1, FLOOR(class_n * .8))

test set:

CREATE OR REPLACE VIEW "DSAGENT$INCOME_MODEL_DATASET_A365_TOP_MDL_TEST"
AS WITH ranked AS (
  SELECT t."CUST_ID" AS id_key,

```

Send a Message

6. Data Science Agent trains model

Conversation progression: After the data split is complete, the user proceeds to model training using the training and validation datasets.

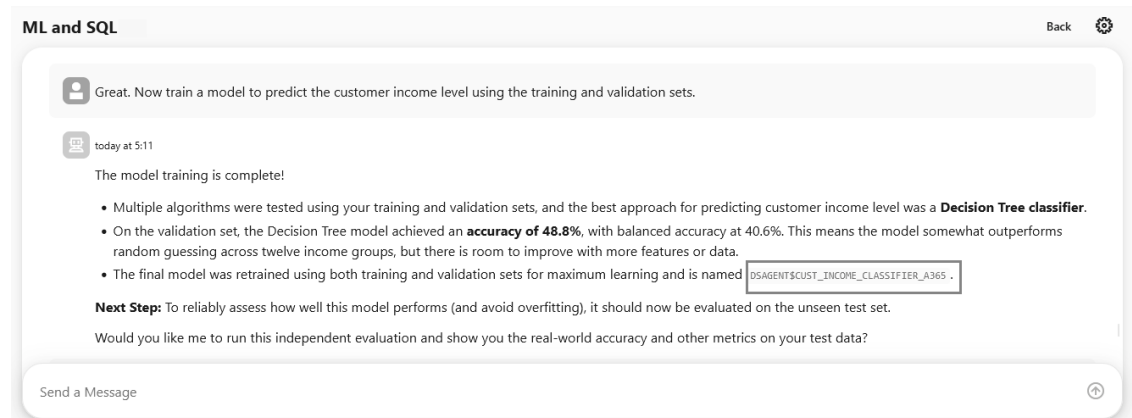
Prompt 7: Great. Now train a model to predict the customer income level using the training and validation sets.

Agent's response:

Data Science Agent confirms completing of model training. It provides the following additional information about the same:

- Algorithm used
- Accuracy achieved
- Model details. The model built is DSAGENT\$CUST_INCOME_CLASSIFIER_A365.
- Next steps for the user to choose from

See screenshot for more details.

Figure 2-29 Data Science Agent trains model

You can also check the model built by the agent on the Models page in Oracle Machine Learning UI, as shown in the screenshot here.

Figure 2-30 Model listed on the Models page in OML UI

Models						
User Models						Deployments
Name	Algorithm	Target	Creation Date	Source	Owner	
DSAGENTSCUST_INCOME_CLASSIFIER_A365	Decision Tree	CUST_INCOME_LEVEL	2/19/2026, 5:11 PM	ML and SQL	OMLUSER	

7. Data Science Agent evaluates the model and computes model accuracy

Conversation progression: The user now focuses on model evaluation.

Prompt 8: Yes. Evaluate the model and show its prediction accuracy.

Agent's response:

Data Science Agent evaluates the model's accuracy. It computes the overall accuracy, balanced accuracy, macro precision, macro precision, and F1 score. Here's a screenshot that shows prediction accuracy done by Data Science Agent.

Figure 2-31 Model evaluation by Data Science Agent

The screenshot shows a chat window titled "ML and SQL". The user asks: "Yes. Evaluate the model and show its prediction accuracy". The agent responds: "The model has been evaluated on the independent test set!"

Prediction Accuracy:

- Overall accuracy: 47.2% (micro metrics align)
- Balanced accuracy: 40.1%
- Macro Precision: 50.7%
- Macro Recall: 40.1%
- Macro F1 Score: 37.3%

This means that the model can correctly predict the exact customer income level about 47% of the time across 12 possible income categories—a challenging multiclass task. Some mid-tier and upper-tier income groups are predicted better than very low or very high ones.

Per-class performance is included for full transparency.

Expand below to see the SQL for running model inference on new customers and more technical details if needed.

Would you like an interpretation of the confusion matrix or tips for improving performance? Or would you like to try using this model for predicting income levels on fresh data?

A button labeled "SQL Code for Inference" is visible at the bottom of the response.

Data Science Agent also provides the SQL code to run inference or scoring using model "OMLUSER"."DSAGENT\$CUST_INCOME_CLASSIFIER_A365" on new raw or unlabelled data. Here's a screenshot depicting the same:

Figure 2-32 Data Science Agent provides SQL query for inference

The screenshot shows the same chat window as Figure 2-31. The agent's response is expanded to show the SQL code for inference. The text reads: "Here is the SQL code to run inference using model 'OMLUSER'.'DSAGENT\$CUST_INCOME_CLASSIFIER_A365' on new raw (unlabelled) data:"

```
SELECT
  "CUST_ID" AS case_id,
  PREDICTION("OMLUSER"."DSAGENT$CUST_INCOME_CLASSIFIER_A365" USING *) AS predicted
, (SELECT pred || ':' || TO_CHAR(prob, '0.9999')
  FROM (
    SELECT ps.PREDICTION AS pred, ps.PROBABILITY AS prob,
           ROW_NUMBER() OVER (ORDER BY ps.PROBABILITY DESC) rn
    FROM TABLE(PREDICTION_SET("OMLUSER"."DSAGENT$CUST_INCOME_CLASSIFIER_A365", NULL USING t.*)) ps
  ) WHERE rn = 1
  ) AS top1
FROM YOUR_NEW_DATA t
```

8. Data Science Agent interprets confusion matrix

Conversation progression: The user now dives into analysis

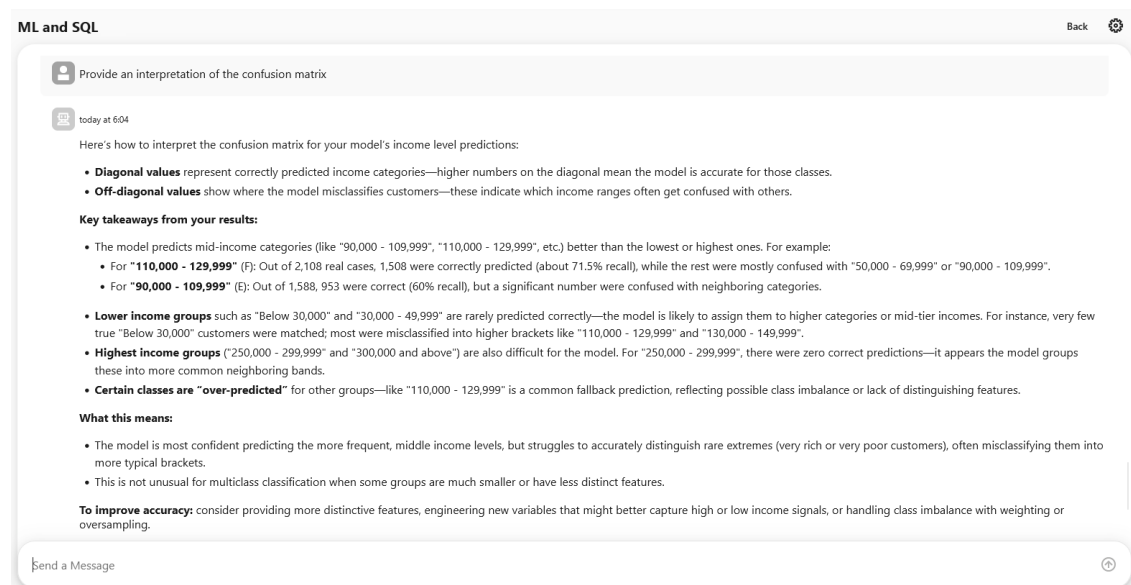
Prompt 9: Provide an interpretation of the confusion matrix.

Agent's response:

Data Science Agent provides a comprehensive interpretation of the confusion matrix for customer income level prediction done by the model. Data Science Agent provides the following information:

- Provides an overview of how to interpret the confusion matrix
- Lists the key takeaways of the results
- Provides an interpretation of the meaning
- Provides suggestions on how the user can improve accuracy of the model's prediction.

Figure 2-33 Data Science Agent interprets Confusion Matrix



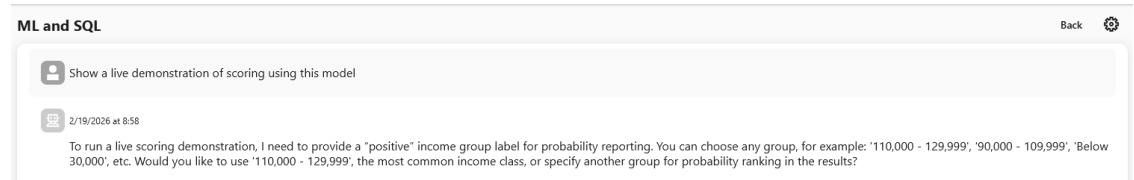
9. Data Science Agent runs live scoring demonstration

Conversation progression: The user now decides to use the model to predict customer income levels on new data. And so, the user prompts Data Science Agent to demonstrate how to score using the model.

Prompt 10: Show a live demonstration of scoring using this model

Agent's response: Data Science Agent responds by explaining what information it requires for scoring with the model. It provides some options to the user. Here is the agent's response:

To run a live scoring demonstration, I need to provide a "positive" income group label for probability reporting. You can choose any group, for example: '110,000 - 129,999', '90,000 - 109,999', 'Below 30,000', etc. Would you like to use '110,000 - 129,999', the most common income class, or specify another group for probability ranking in the results?

Figure 2-34 Response of Data Science Agent for running live scoring demonstration

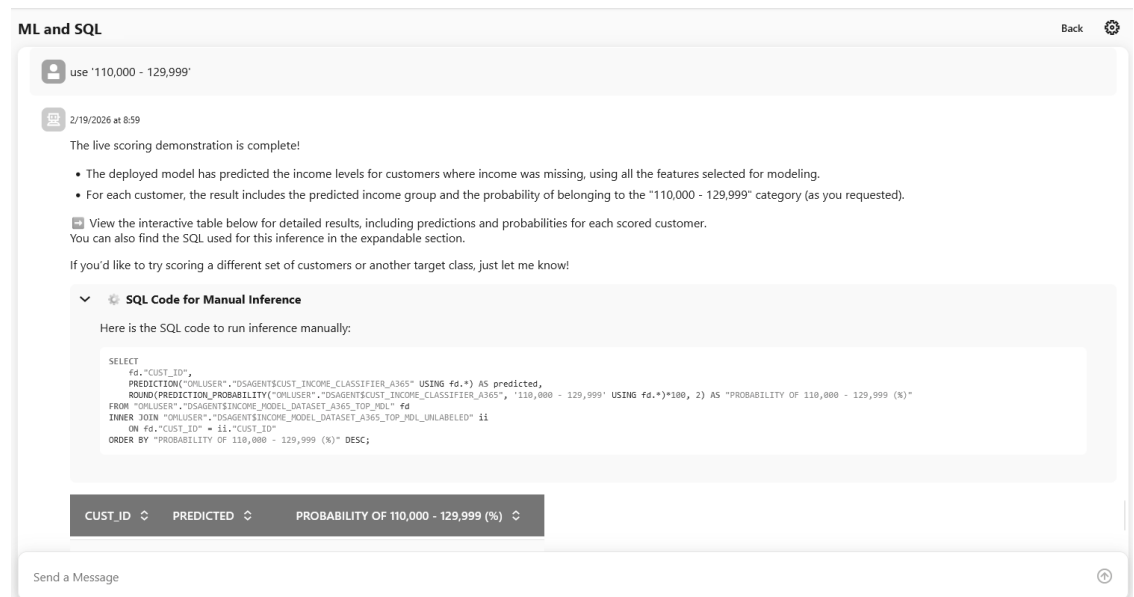
The user responds to the agent, and provides an input in prompt 11.

Prompt 11: use '110,000 - 129,999'

Agent's response: Data Science Agent demonstrates live scoring with the model. It provides the following information:

- Confirms the completion of the task.
- Provides details of the task—scoring.
- Provides the SQL code for running manual inference
- Provides a table containing detailed results, including predictions and probabilities for each scored customer

Here's a screenshot depicting the SQL code for manual inference. This information is presented in an expandable section.

Figure 2-35 SQL code for manual inference

Here's a screenshot depicting a table containing detailed results, including predictions and probabilities for each scored customer.

Figure 2-36 Table containing predictions and probabilities of each scored customer

ML and SQL Back

```

SELECT
  fd."CUST_ID",
  PREDICTION("OHLUSER", "DSAGENTS$CUST_INCOME_CLASSIFIER_A365" USING fd.*) AS predicted,
  ROUND(PREDICTION_PROBABILITY("OHLUSER", "DSAGENTS$CUST_INCOME_CLASSIFIER_A365", '110,000 - 129,999' USING fd.*)*100, 2) AS "PROBABILITY OF 110,000 - 129,999 (%)"
FROM "OHLUSER"."DSAGENTS$INCOME_MODEL_DATASET_A365_TOP_MDL" fd
INNER JOIN "OHLUSER"."DSAGENTS$INCOME_MODEL_DATASET_A365_TOP_MDL_UNLABELED" ll
ON fd."CUST_ID" = ll."CUST_ID"
ORDER BY "PROBABILITY OF 110,000 - 129,999 (%)" DESC;
        
```

CUST_ID	PREDICTED	PROBABILITY OF 110,000 - 129,999 (%)
590	F: 110,000 - 129,999	
323	C: 50,000 - 69,999	
31	C: 50,000 - 69,999	
403	F: 110,000 - 129,999	
856	F: 110,000 - 129,999	
617	C: 50,000 - 69,999	
643	C: 50,000 - 69,999	
697	C: 50,000 - 69,999	

Send a Message

3

Best Practices in using Data Science Agent

Follow these best practices to maximize the benefits of Data Science Agent.

Topics:

- [Recommended Models](#)
- [Associate Database objects to your conversation](#)
- [Ask for clarification](#)
- [Ask in multiple iterations](#)
- [Clarify terminology](#)
- [Follow suggestions provided by Data Science Agent](#)
- [Limit your conversation length and scope](#)
- [Provide context to your conversation](#)
- [Specify a clear objective](#)

3.1 Recommended Models

Data Science Agent works with large language models accessed through Oracle `DBMS_CLOUD_AI` and `DBMS_CLOUD_AI_AGENT` packages. The `DBMS_CLOUD_AI` package, with `Select AI`, supports the translation of natural language prompts to generate, run, explain SQL statements, and also enables RAG and natural language-based interactions, including chats with LLMs. For more information, see [DBMS_CLOUD_AI Package](#) and [DBMS_CLOUD_AI_AGENT Package](#).

This table lists the recommended large language models and the scenarios in which each should be used.

Note

Oracle recommends using GPT-4.1 as the preferred model for all Data Science Agent users for reliability, responsiveness, and cost-effectiveness.

Table 3-1 Recommended models

Large Language Models	Provider	Scenario	Strengths
GPT-4.1	OpenAI or OCI GenAI	Use this model for all general-purpose data science tasks.	<ul style="list-style-type: none">• Reliable• Cost-effective

Table 3-1 (Cont.) Recommended models

Large Language Models	Provider	Scenario	Strengths
Grok-4	OCI GenAI	Use this model for complex, multi-step tasks requiring advanced reasoning. The response time of Data Science Agent is expected to be slow when using Grok-4.	Powerful reasoning capabilities

Profile Creation for GPT-4.1

To create an AI profile for GPT-4.1 through Open AI, run the following script in a notebook:

```

DECLARE
    profile_name VARCHAR2(128) := 'OPENAI_GPT_4_1';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "comments": false,
            "conversation": true,
            "credential_name": "OPENAI_CRED",
            "model": "gpt-4.1",
            "provider": "openai",
            "temperature": 1,
            "max_tokens": 8192
        }'
    );
END;
/

```

To create an AI profile for GPT 4.1 through Oracle Cloud Infrastructure (OCI), run the following script in a notebook:

```

DECLARE
    profile_name VARCHAR2(128) := 'OCI_GPT_4_1';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "comments": false,
            "conversation": true,
            "credential_name": "OCI_CRED",
            "model": "openai.gpt-4.1",
            "provider": "oci",
            "temperature": 1,
            "max_tokens": 8192,
            "oci_compartment_id": "<your-dep-id>",

```

```

        "oci_apiformat": "GENERIC"
    }'
);
END;
/

```

Profile Creation for Grok 4

To create an AI profile for Grok 4 through Oracle Cloud Infrastructure (OCI), run the following script in a notebook:

```

DECLARE
    profile_name VARCHAR2(128) := 'OCI_GROK_4';
BEGIN
    dbms_cloud_ai.drop_profile(profile_name, TRUE);
    dbms_cloud_ai.create_profile(
        profile_name => profile_name,
        attributes => '{
            "comments": false,
            "conversation": true,
            "credential_name": "OCI_CRED",
            "model": "xai.grok-4",
            "provider": "xAI",
            "temperature": 1,
            "max_tokens": 8192,
            "oci_compartment_id": "<your-dep-id>",
            "oci_apiformat": "GENERIC"
        }'
    );
END;
/

```

Parameters:

- `profile_name`: A name for the AI profile. The profile name must follow the naming rules of Oracle SQL identifier. Maximum length of profile name is 125 characters.
- `comments`: Includes table and column comments in the metadata used for translating natural language prompts using AI. `BOOLEAN` data type is supported. The valid values are `TRUE` or `FALSE` for a string with `VARCHAR2` data type. The values are not case sensitive.
- `conversation`: A `VARCHAR2` attribute that indicates if conversation history is enabled for a profile. Valid values are `true` or `false`. The default value is `false`. The values are not case sensitive.
- `credential_name`: The name of the credential to access the AI provider APIs.
- `model`: The name of the AI model being used to generate responses.
- `provider`: AI provider for the AI profile. This is a mandatory attribute.
- `temperature`: Controls the randomness of the model's output. Lower values, for example 0, make the responses more deterministic and focused. Higher values, for example, 1, make them more creative and varied. You may want to tune it depending on your use case.
 - Lower values are generally preferred for structured or factual tasks
 - Higher values can be useful for more open-ended generation. `temperature = 1` gives the best results.

- `max_tokens`: Denotes the number of tokens to predict per generation. Default is 1024. Sets the maximum length of the agent's response. The default value of 1024 tokens may be too low for complex or verbose answers. Setting it to 4096 should be sufficient for most use cases, while 8192 provides extra headroom for longer responses.

Note

This can be an arbitrary number, but not strictly a power of 2.

- `oci_compartment_id`: Specifies the OCID of the compartment you are permitted to access when calling the OCI Generative AI service. The compartment ID can contain alphanumeric characters, hyphens and dots. The default is the compartment ID of the PDB.
- `oci_apiformat`:

For more information, see [Manage AI Profiles](#).

3.2 Associate Database objects to your conversation

Consider associating database objects such as tables, views and mining models to a Data Science Agent conversation. Once you associate these objects, the agent can inspect, analyze, transform, and model from those objects directly. This will thereby enhance the quality of the agent's response. If you do not associate any object, the agent will automatically scan the database for relevant objects based on your query.

Note

Some operations such as feature ranking, model search, training can be compute-intensive and may take time.

3.3 Ask for clarification

During the course of your conversation, you can ask for clarification at anytime. Some examples:

- What was done in the previous step
- Why was a particular step necessary
- What is the next recommended step
- Explain the <concept>. For example, what is unstructured data in machine learning?

3.4 Ask in multiple iterations

If you are using Data Science Agent for extended workflows, consider asking the agent in multiple iterations. Longer workflows are generally more effective when handled iteratively. For instance, you can start by creating a dataset view, then move to validating assumptions, and finally focus on model training and evaluation.

3.5 Clarify terminology

If you use specific terms in your conversation, it is a good practice to clarify those terms to the agent.

3.6 Follow suggestions provided by Data Science Agent

Follow the suggestion of the agent when appropriate. The agent frequently proposes the next steps of a workflow. For example, data preparation, analysis, model training. Accept or refine these suggestions for a smooth progress.

3.7 Limit your conversation length and scope

Although Data Science Agent can handle extended interactions, very long conversations may gather context that negatively affects clarity or performance. For extended work, consider starting a new conversation, especially if you encounter these situations:

- If your conversation has a lot of messages (around 50 messages), or
- If your objective changes

3.8 Provide context to your conversation

The interaction with Data Science Agent is structured as a conversation, consisting of alternating `turns`. A `turn` begins with your prompt, followed by the agent's response. A Data Science Agent conversation maintains the context across `turns`.

Therefore, providing context to your conversation is a good practice, especially if you resume a conversation at a later time.

3.9 Specify a clear objective

Clearly state your objective at the beginning of the conversation. For example, "I want to predict customer churn" or "I want to identify the main causes". Sharing a high-level intent early in the conversation helps guide the rest of the conversation. When the agent understands your objective, it can suggest the most appropriate workflow.

4

Known Issues in Data Science Agent

Learn about the issues you may encounter when using Data Science Agent in Oracle Machine Learning UI on Oracle Autonomous AI Database and the workaround.

Topics:

- [Objects present in user schema are not listed](#)

4.1 Objects present in user schema are not listed

Issue: Objects present in the user schema are not listed in the **Object** field in the Object Management dialog when **Show other schema** is turned off. The objects are tables, views, and mining models.

Expected behavior: In the Object Management dialog, the user is expected to see the objects—tables, views, and mining models present in their own schema when **Show other schema** is turned off.

Workaround: Turn on **Show other schemas** to view the objects present in your schema. Follow these steps to view and add objects present in your own schema and in other schemas:

1. In a Data Science Agent Conversation, open the **Settings** pane and click + on the **Associated Objects** field.
2. In the Object Management dialog, turn on **Show other schemas** to view all the objects present in your schema as well as in other schemas.
3. Select the objects and click **OK**.