Oracle® Cloud

Using Oracle Machine Learning on Autonomous Database

E78525-47
April 2022

Oracle Cloud Using Oracle Machine Learning on Autonomous Database,

E78525-47

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

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 on Autonomous Database

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

Table 1-1 New Feature

Feature	Description
Support for cross-region Oracle Data Guard	Oracle Machine Learning Notebooks provides cross-region Oracle Data Guard support in newly provisioned and migrated databases.

Table 1-1 (Cont.) New Feature

Feature	Description
Oracle Machine Learning repository migrated from shared database to each respective Autonomous Database instance.	The Oracle Machine Learning (OML) repository has been migrated from shared database to each respective Autonomous Database instance. The migration of the Oracle Machine Learning repository ensures: • That all OML objects such as tables, jobs, store procedures, and metadata are moved to the appropriate 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. In case you are not able to view the expected behavior after you clone your Oracle Autonomous Database, check again in a few days. Otherwise, you may contact Oracle Support for more information The OML repository version is mentioned in **About** in the `<user>` drop-down list on the top right corner of your Oracle Machine Learning Notebooks page. If the version is `1.0.0.0.0`, it indicates that the OML metadata is still in the shared database. If the version is `22.x`, it indicates that the OML repository has been migrated to your Autonomous Database instance.

Table 1-1 (Cont.) New Feature

Feature	Description
Oracle Machine Learning Notebooks supported on all Oracle Autonomous Database clones	Oracle Machine Learning Notebooks is supported on all types of Oracle Autonomous Database 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 Notebooks

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

- About Oracle Machine Learning Notebooks
  Oracle Machine Learning Notebooks is an Apache Zeppelin-based collaborative web-based interface that supports notebook creation, scheduled run, and versioning. You can document work using Markdown and run SQL and Python code for data exploration, visualization, and preparation, and machine learning model building, evaluation, and deployment.
- Oracle Machine Learning Notebooks Home Page
  The Oracle Machine Learning Notebooks home page is the default landing page when you log in to Oracle Machine Learning Notebooks. The home page provides you quick links to important interfaces, and the log of your high-level recent activities.
- Get Started with Oracle Machine Learning Notebooks
  Here is how you can get started with Oracle Machine Learning Notebooks.
- Access Oracle Machine Learning Notebooks
  You can access Oracle Machine Learning **Notebooks** from Autonomous Database.
- Typical Workflow For Using Notebooks
  To begin with Oracle Machine Learning Notebooks, refer to the tasks listed in the table as a guide.

Related Topics

- Algorithms

About Oracle Machine Learning Notebooks

Oracle Machine Learning Notebooks is an Apache Zeppelin-based collaborative web-based interface that supports notebook creation, scheduled run, and versioning. You can document work using Markdown and run SQL and Python code for data exploration, visualization, and preparation, and machine learning model building, evaluation, and deployment.

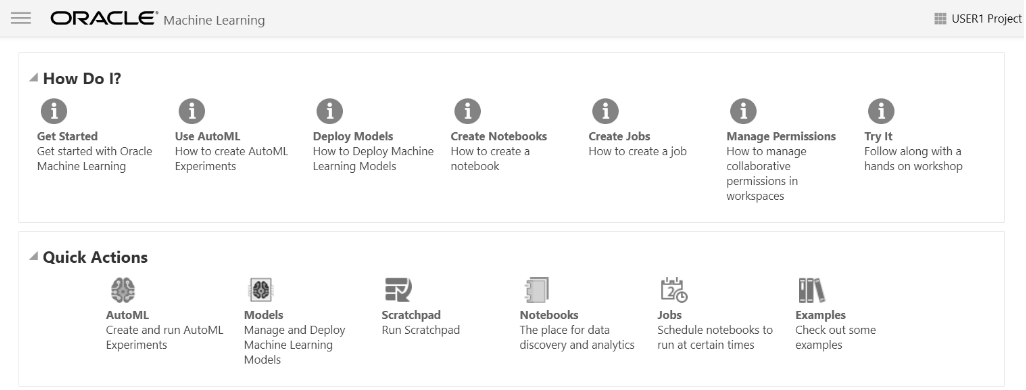
Key features of Oracle Machine Learning Notebooks:

- 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, and Markdown

Oracle Machine Learning Notebooks Home Page

The Oracle Machine Learning Notebooks home page is the default landing page when you log in to Oracle Machine Learning Notebooks. The home page provides you quick links to important interfaces, and the log of your high-level recent activities.

Figure 2-1 Oracle Machine Learning Notebooks Home Page



ORACLE Machine Learning USER1 Project

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

In the home page, you can access:

- The **How Do I** help links to:
  - **Get Started**
  - **Use AutoML**
  - **Deploy Models**
  - **Create Notebooks**
  - **Create Jobs**
  - **Manage Permissions**
  - **Try It**
- The **Quick Actions** links to:
  - **AutoML**
  - **Models**
  - **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
  The Notebooks page lists all the notebooks associated with the selected project. You can create, edit, and run your notebooks here.
- Jobs
  Jobs allow you to schedule the running of notebooks. In 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.

Get Started with Oracle Machine Learning Notebooks

Here is how you can get started with Oracle Machine Learning Notebooks.

1. Request access to Oracle Machine Learning Notebooks. Contact your Service Administrator to provide access to your Oracle Machine Learning Notebooks account.
2. Access the Oracle Machine Learning Notebooks 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 and assign it to the default project and workspace. You can also create your own project and workspace.

   > **Note:**
   >
   > If you are using Oracle Cloud Free Tier, then 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 secured password.

Access Oracle Machine Learning Notebooks

You can access Oracle Machine Learning **Notebooks** from Autonomous Database.

To access Oracle Machine Learning Notebooks from the Autonomous Database Service Console:

1. Select an Autonomous Database instance and on the Autonomous Database details page click **Service Console**.
2. Click **Development**.
3. On the Development page click **Oracle Machine Learning User Interface**.
4. Enter your username and password.
5. Click **Sign In**.

This shows 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 secured password.

About Oracle Machine Learning Password Policy

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

Typical Workflow For Using 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 Notebooks	Access Oracle Machine Learning
Create workspaces	Create Projects and Workspaces
Create projects	Create Projects and Workspaces
Create notebooks	Create a Notebook
Run a Notebook with Python Interpreter	Run a Notebook with Python Interpreter
Use the Scratchpad	Use the Scratchpad

Tasks	More Information
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 by granting permissions to access your workspace. Your workspace contains your projects and notebooks.

3

Get Started with Project and Workspaces

A project is a container for storing your notebooks and other objects such as dashboards and so on. A workspace is a virtual space where your projects reside, and multiple users with the appropriate permission type can work on different projects.

- Create Projects and Workspace
  A project is a container for your notebooks, and a workspace is a container for your projects. While you may own many projects, other workspaces and projects may be shared with you.
- Grant Workspace Permissions
  You can collaborate with other users in Oracle Machine Learning by granting permissions to access your workspace. Your workspace contains your projects and notebooks.

Create Projects and Workspace

A project is a container for your notebooks, and a workspace is a container for your projects. While you may own many projects, other workspaces and projects may be shared with you.

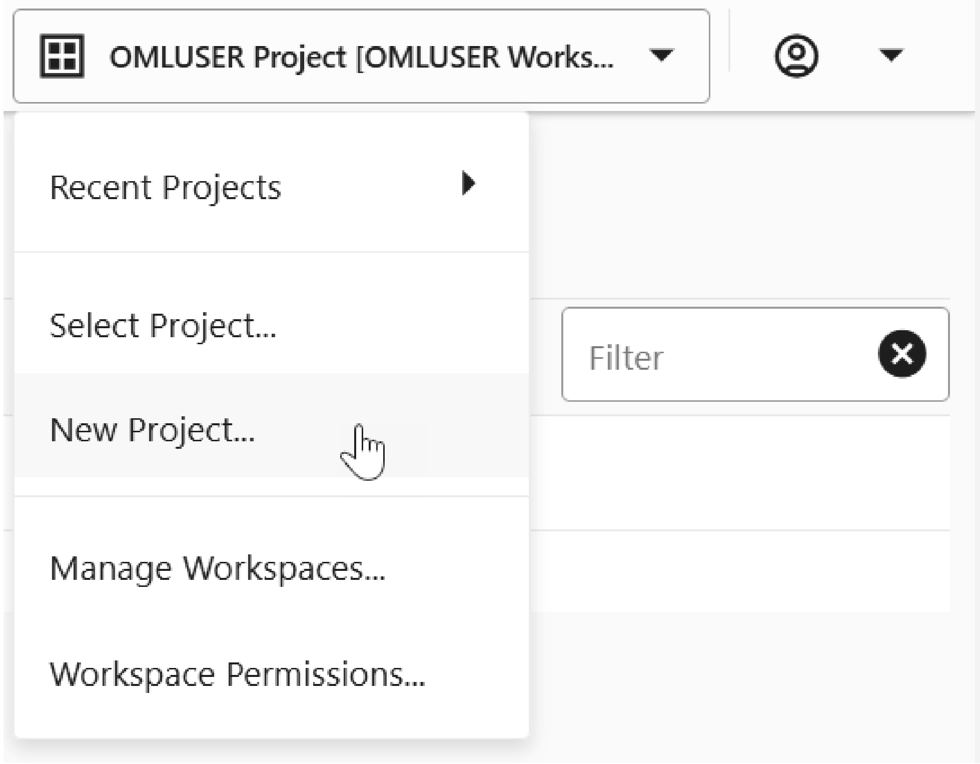
The initial workspace and the default project is created by the Oracle Machine Learning service automatically when you log in to Oracle Machine Learning for the first time. To create a new project and a workspace:

1. On the top right corner of Oracle Machine Learning home page, click the project workspace drop-down list. The project name and the workspace, in which the project resides, are displayed here. In this screenshot, the project name is USER1 Project, and the workspace name is USER1 Workspace. 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.

2. To create a new project, click **New Project.**

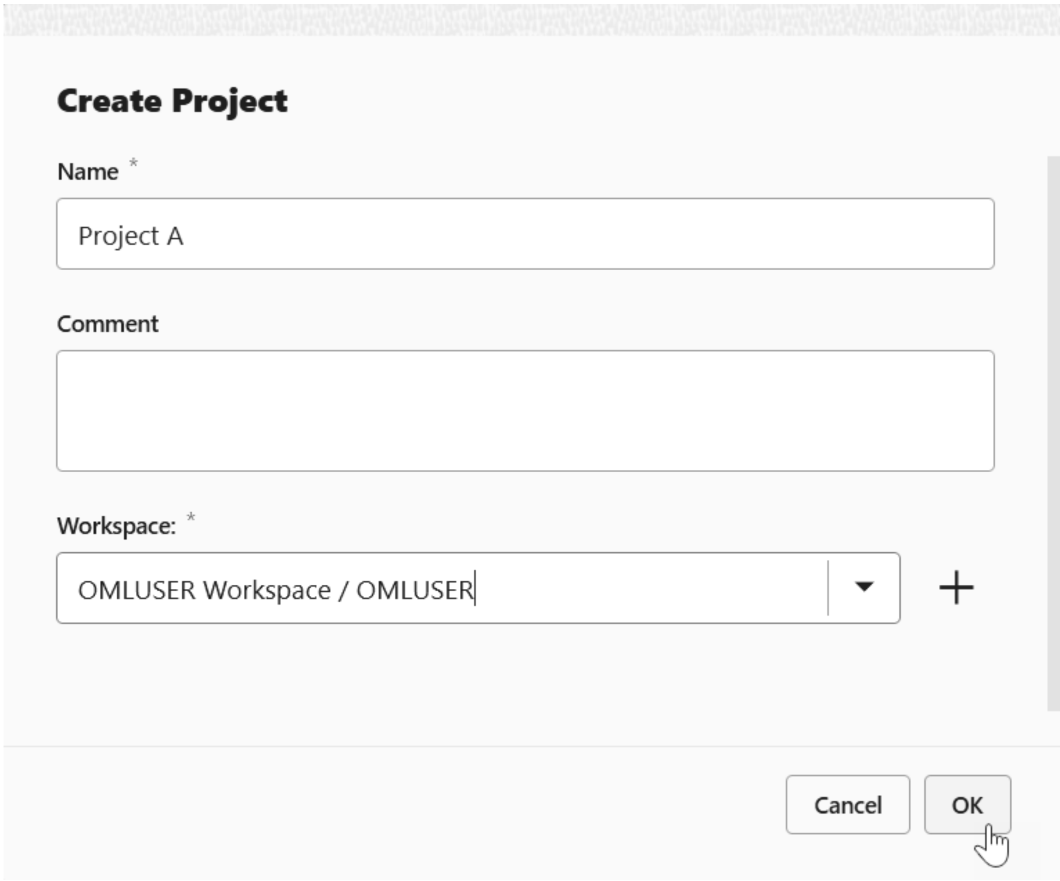
Figure 3-1   Project Workspace Drop-down



The Create Project dialog box opens.

3. In the Create Project dialog, enter the following:

Figure 3-2 Create Project Dialog



Create Project

Name *

Project A

Comment

Workspace: *

OMLUSER Workspace / OMLUSER

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

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

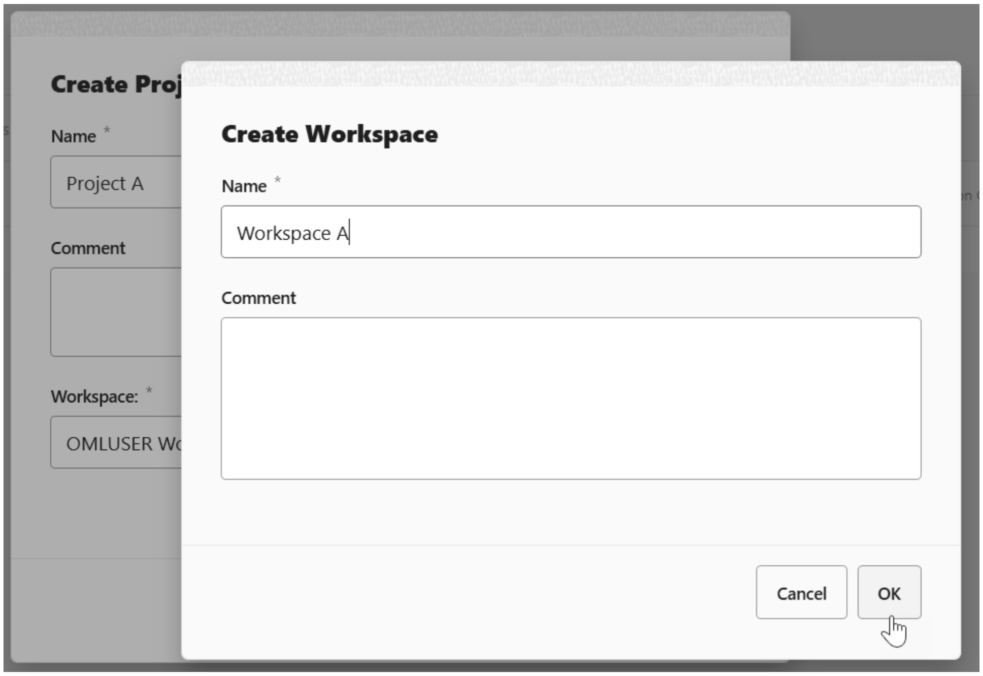
c. **Workspace:** By default, the default workspace is selected. To choose another workspace or create a new workspace, click the down arrow and select a workspace.

4. To create a new workspace, click + next to the **Workspace** field in the Create Project dialog. The Create Workspace dialog opens.

> **Note:**
>
> You can also create a workspace by clicking **Create** in the Select Project dialog.

5. In the Create Workspace dialog box, define the new workspace:

Figure 3-3 Create Workspace Dialog



Create Workspace

Name *

Workspace A

Comment

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

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

c. Click **OK.**

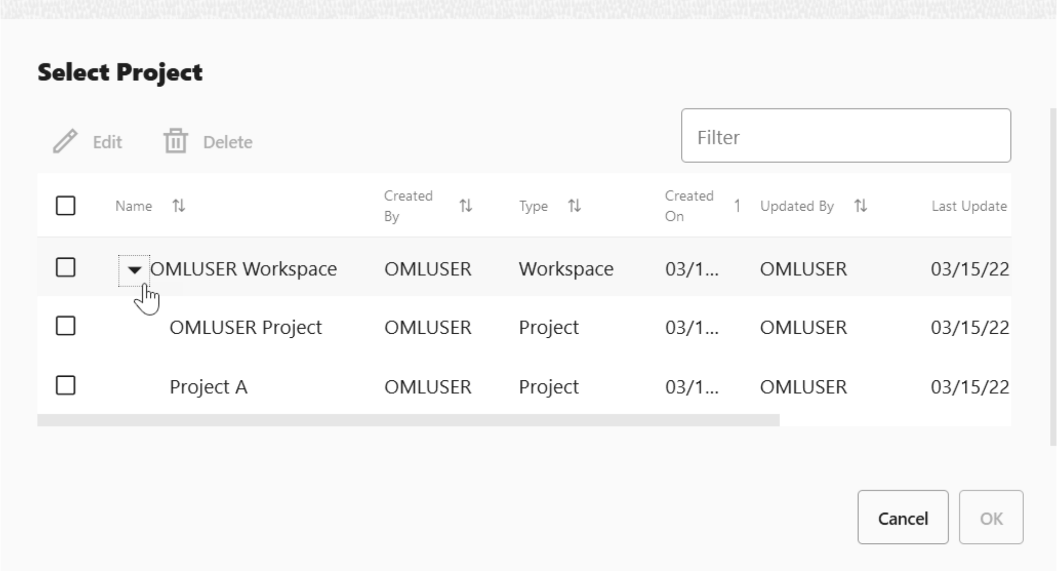
This creates your workspace, and navigates back to the Create Project dialog box. The project that you are creating is now assigned to the newly created workspace.

6. Click **OK.**

7. To select a different project inside another workspace, click **Select Project** in the project workspace drop-down list.

8. In the Select Project dialog, expand Workspace to view the projects listed in it and then click the project you want to access.

Figure 3-4 Select Project Dialog



□	Name	Created By	Type	Created On	Updated By	Last Update
□	OMLUSER Workspace	OMLUSER	Workspace	03/1...	OMLUSER	03/15/22
□	OMLUSER Project	OMLUSER	Project	03/1...	OMLUSER	03/15/22
□	Project A	OMLUSER	Project	03/1...	OMLUSER	03/15/22

The Select Project dialog lists the workspace and all the projects in it, along with the following information:

- Name
- Created By
- Type
- Created on
- Updated By
- Last Update
- Comments

You can sort, edit, and delete projects here.

> **Note:**
>
> To delete workspaces, click **Manage Workspaces** in the project workspace drop-down list.

The project `Project A` is created and assigned to the workspace `Workspace A`. The screenshot in this example displays two workspaces - `USER1 Workspace` and `Workspace A` with one project in each workspace.

Related Topics

- Manage Workspaces
  You can provide access to your workspace, manage permissions for users, and edit and delete workspace.

Grant Workspace Permissions

You can collaborate with other users in Oracle Machine Learning by granting permissions to access your workspace. Your workspace contains your projects and notebooks.

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.

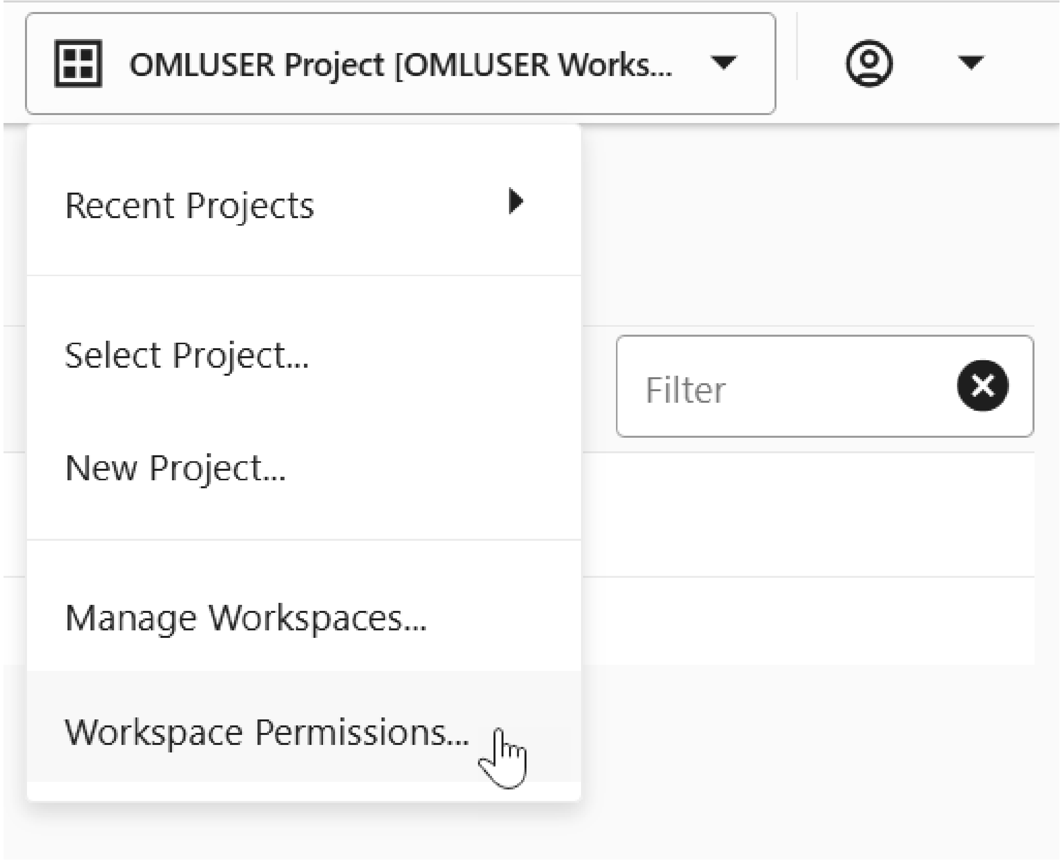
> **Caution:**
>
> If you grant the permission type **Manager** or **Developer,** then the user can also drop tables, create tables, and run any scripts at any time on your account. The user with **Viewer** permission type can only view your notebooks, and is not authorized to run or make any changes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

To grant permission to another user:

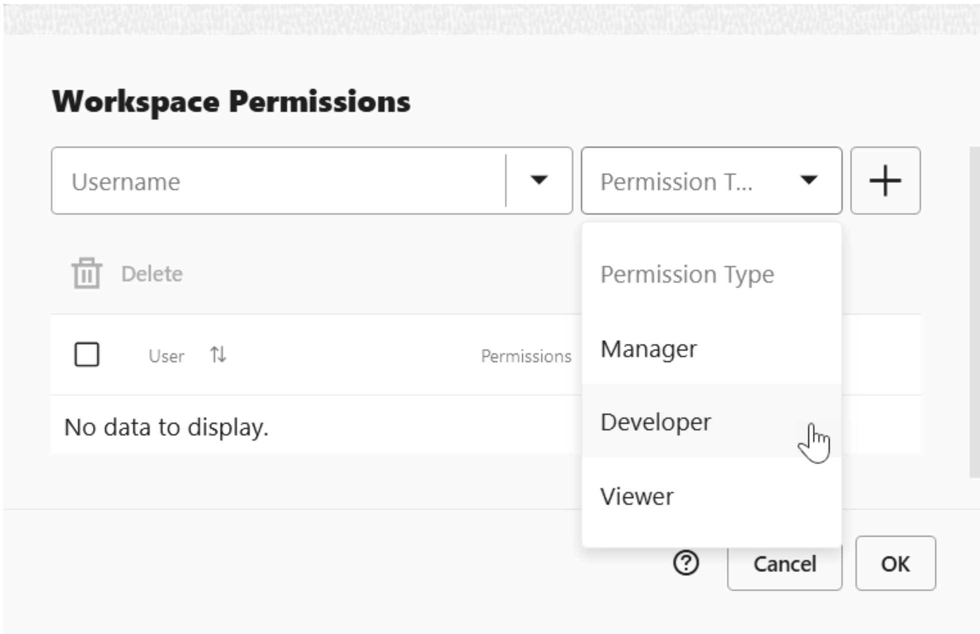
1. On the top right corner of Oracle Machine Learning home page, click the project and workspace drop-down list and select **Workspace Permission.**

   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 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.
- Manage Workspaces
  You can provide access to your workspace, manage permissions for users, and edit and delete workspace.

About Workspace Permission Types

Oracle Machine Learning 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 through 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.

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

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

Manage Workspaces

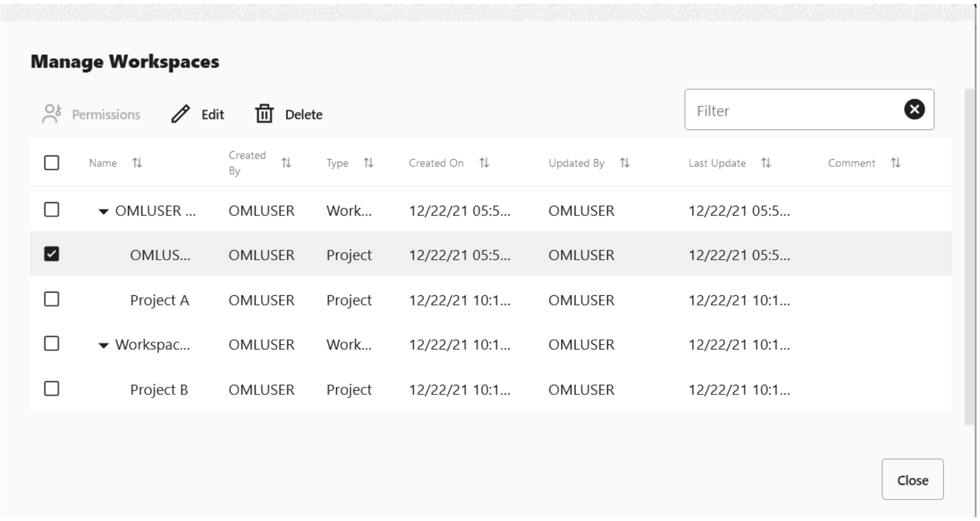
You can provide access to your workspace, manage permissions for users, and edit and delete workspace.

To manage workspaces:

1. On the top right corner of your home page, click the Project Workspace drop-down list.
2. Click **Manage Workspaces.**

   The Manage Workspaces dialog box opens.
3. In the Manage Workspace dialog, you can perform the following tasks:

Figure 3-5 Manage Workspace



□	Name	Created By	Type	Created On	Updated By	Last Update	Comment
□	▾ OMLUSER ...	OMLUSER	Work...	12/22/21 05:5...	OMLUSER	12/22/21 05:5...	
☑	OMLUS...	OMLUSER	Project	12/22/21 05:5...	OMLUSER	12/22/21 05:5...	
□	Project A	OMLUSER	Project	12/22/21 10:1...	OMLUSER	12/22/21 10:1...	
□	▾ Workspac...	OMLUSER	Work...	12/22/21 10:1...	OMLUSER	12/22/21 10:1...	
□	Project B	OMLUSER	Project	12/22/21 10:1...	OMLUSER	12/22/21 10:1...	

- Click **Permissions** to add users and grant permissions to access the workspace. You can also delete users and modify their permissions.
- Click **Edit** to edit the selected workspace.
- Click **Delete** to delete the selected workspace.

4. Once you have finished your tasks, click **Close.**

4

Get Started with Notebooks for Data Analysis and Data Visualization

Oracle Machine Learning Notebooks is based on Apache Zeppelin technology. 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, where you can create, edit, delete, and even save notebooks as templates.

- About OML4Py
  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 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 Notebooks is available through Autonomous Data Warehouse (ADW) , Autonomous Transaction Processing (ATP) and Autonomous JSON Database (AJD) services.
- About Notebooks
  The Notebooks page lists all the notebooks associated with the selected project. You can create, edit, and run your notebooks here.
- Use the Scratchpad
  The Scratchpad provides you convenient one-click access to a notebook for running SQL statements, PL/SQL scripts, and Python scripts that can be renamed. The Scratchpad is available on the Oracle Machine Learning Notebooks home page.
- About Interpreter Bindings and Notebooks
  An interpreter is a plug-in that allows you to use a specific data processing language backend.
- Collaborate in Oracle Machine Learning
  Two or more users can collaborate and share Oracle Machine Learning notebooks with other users.

About OML4Py

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 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 Notebooks 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 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

About Notebooks

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:

- To create a new notebook, select a project and click **Create.**
- To edit a notebook, select the notebook and click **Edit.** You can edit the notebook name, and add comments in the Edit Notebook dialog box.
- To create a copy of a notebook, select the notebook and click **Duplicate.** The duplicate copy of the selected notebook is created, and listed in the Notebooks page with the suffix _1 in the notebook name.

- To save a notebook as a template, select the notebook and click **Save as Template.** You can save the template in **Personal** or **Shared** under Templates.
- To delete a notebook, select it and click **Delete.**
- To import a notebook as `.json` files, click **Import.** Select the project and workspace in which to import the notebook.
- 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.**
- To open a notebook and run it, click the notebook. The 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` or `%md`
- 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.
- Set Output Format in Notebooks
  Oracle Machine Learning Notebooks allow you to preformat query output in notebooks.

Related Topics

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

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. In the Oracle Machine Learning home page, click **Notebooks.** The Notebooks page opens.
2. In 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. In the **Connections** field, select a connection in the drop-down list. By default, the Global Connection Group is assigned.
6. 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, and run Python 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 run Python scripts

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` or `%md`

Set the context with a project with which your notebook is associated.

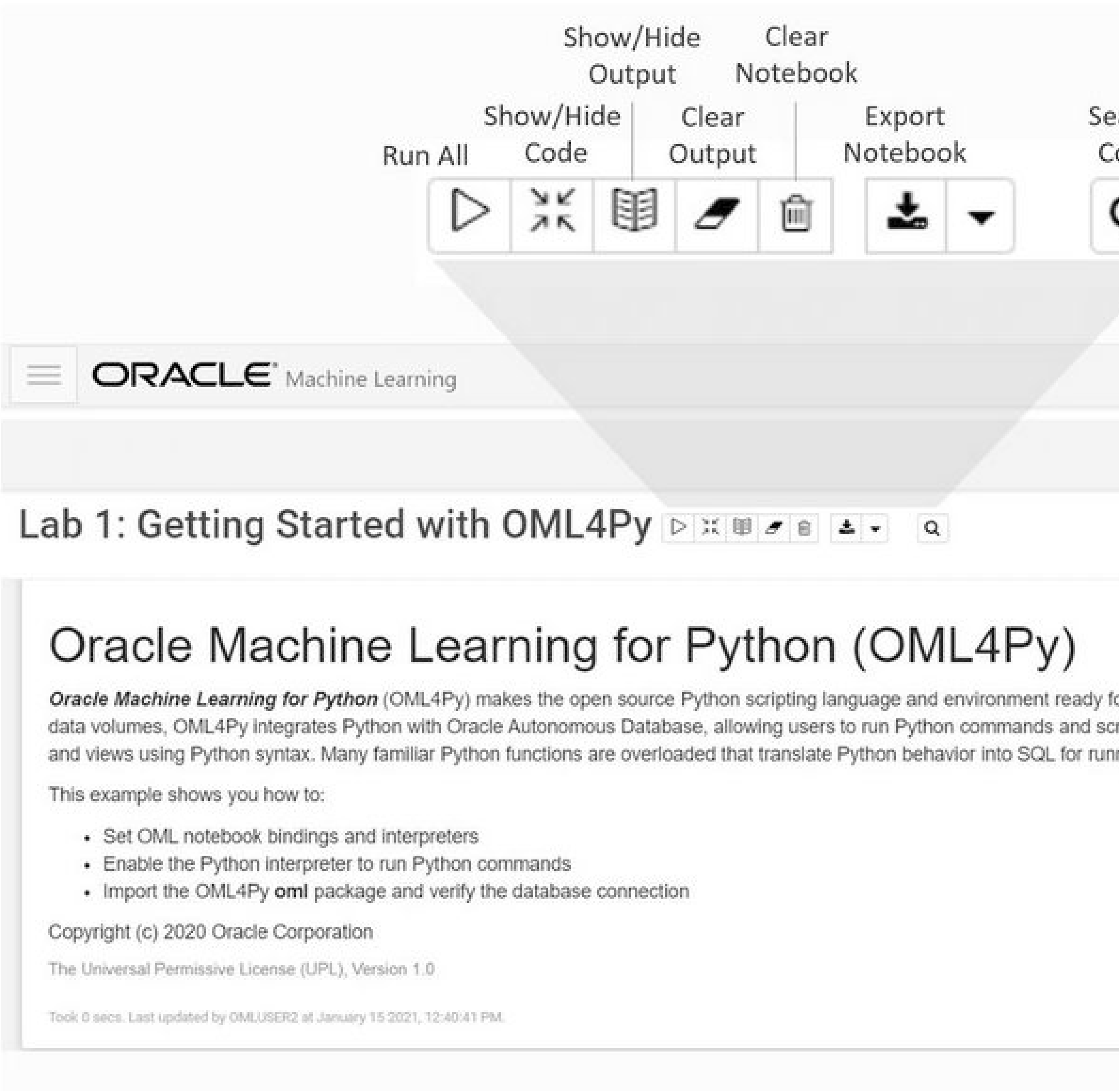
You can edit an existing notebook in your project. To edit an existing notebook:

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

   In 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 4-1 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.

- Run a Notebook with Python 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.
- Run a Notebook with SQL Interpreter
  A notebook can contain many paragraphs which is a notebook component where you can write SQL statements, run PL/SQL scripts, and run Python commands. You can run one or all the paragraphs in a notebook.
- Call the Markdown Interpreter and Generate Static html from Markdown Plain Text
  Use the Markdown interpreter and generate static html from Markdown plain text.
- 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.
- 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.

Run a Notebook with Python Interpreter

To run Python commands in a notebook, you must first connect to the Python interpreter. To use OML4Py, you must import the `oml` module.

In an Oracle Machine Learning notebook, you can add multiple paragraphs, and each paragraph can be connected to different interpreters such as SQL or Python. 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 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`

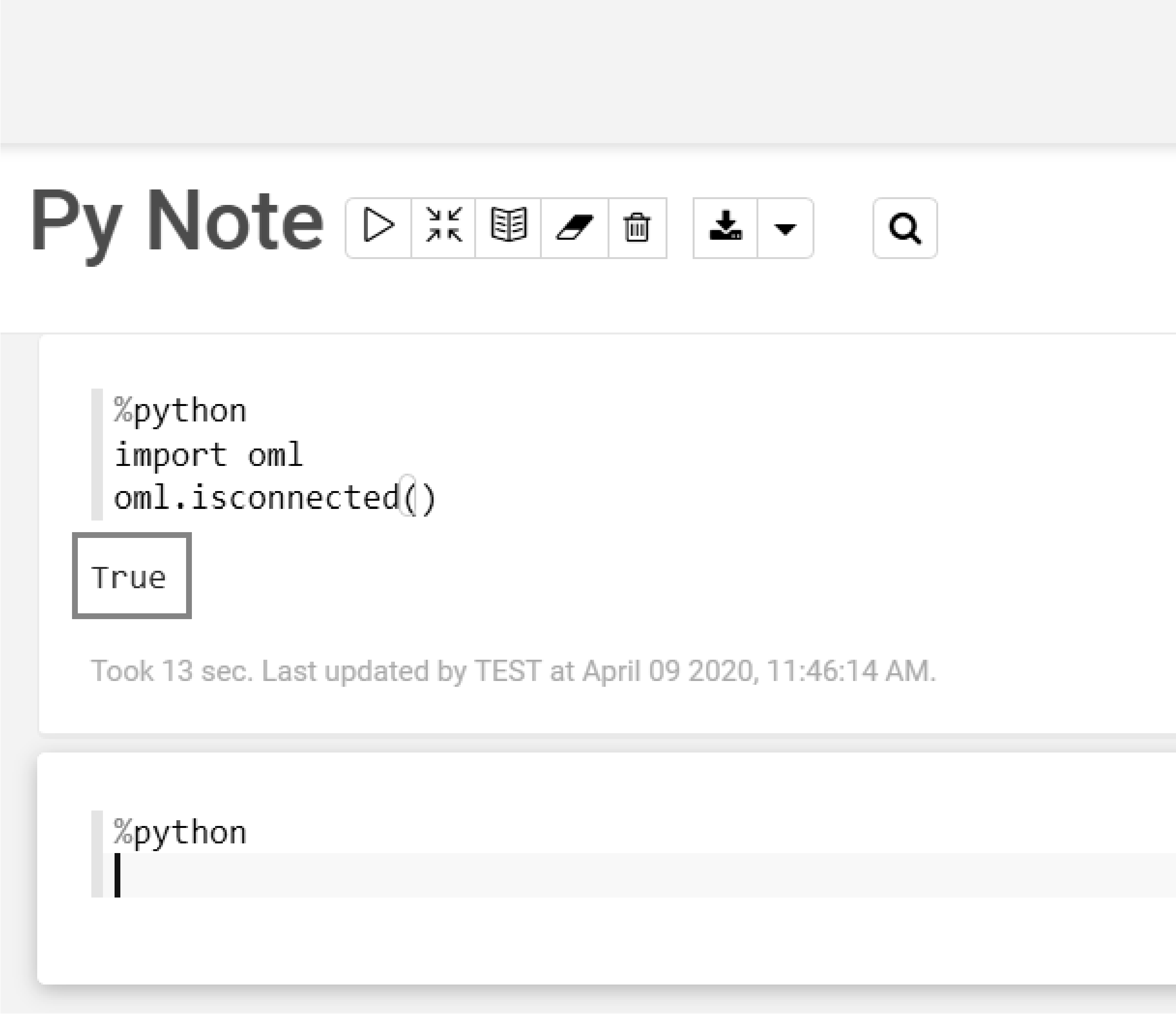
You are now ready to run Python scripts in your notebook.

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
```

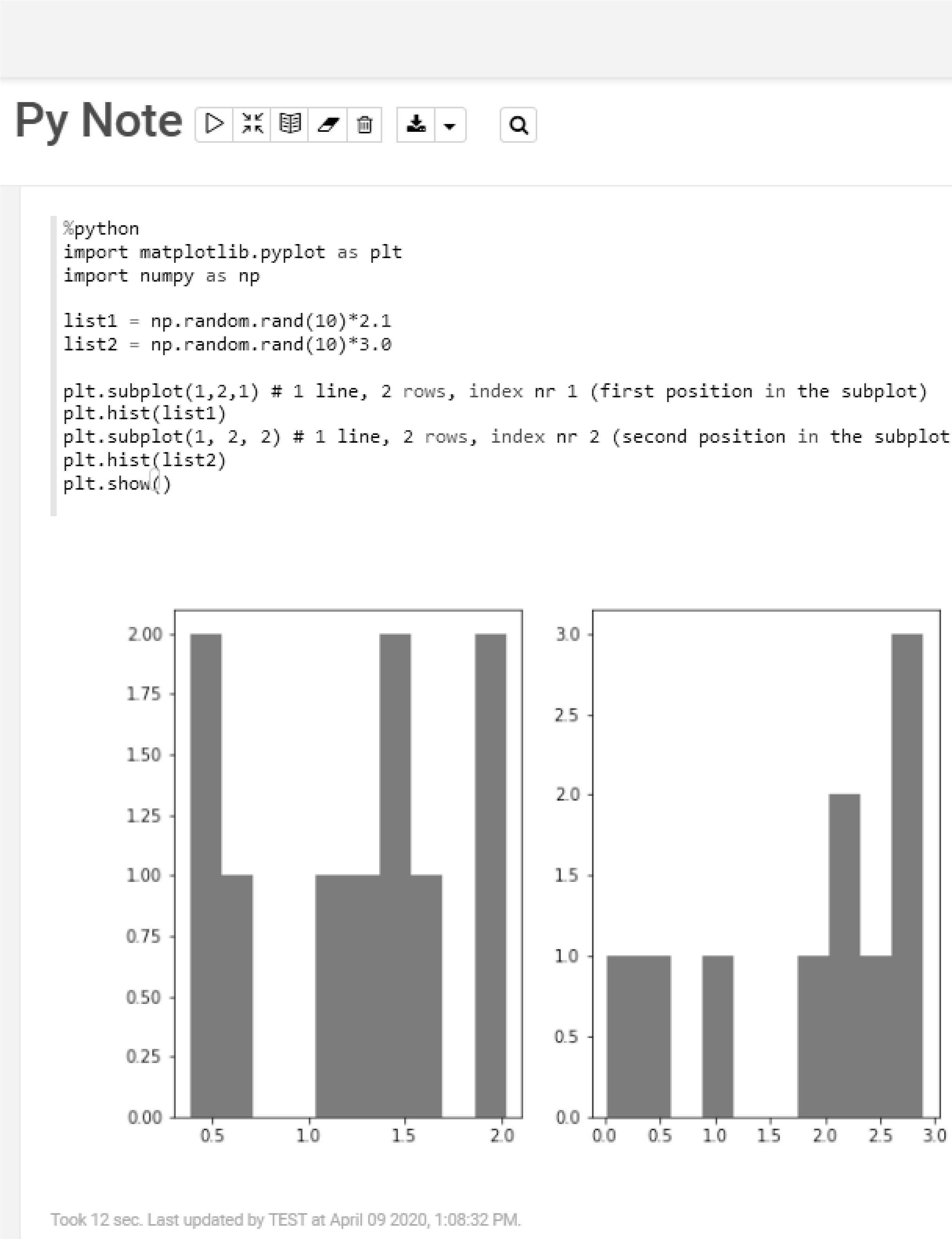
- `Matplotlib` - Python module to render graphs
- `Numpy` - Python module for computations

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.

Run a Notebook with SQL Interpreter

A notebook can contain many paragraphs which is a notebook component where you can write SQL statements, run PL/SQL scripts, and run Python commands. You can run one or all the paragraphs in a notebook.

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:

- `%python` - Supports Python scripts. Type `%python` at the beginning of the paragraph to connect to the Python interpreter.
- `%sql` - Supports standard SQL statements. In `%sql` the results of a `SELECT` statement are directly displayed in a Zeppelin 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.
- `%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** in the Connections Group page.

To fetch and visualize data in a notebook:

1. In 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.

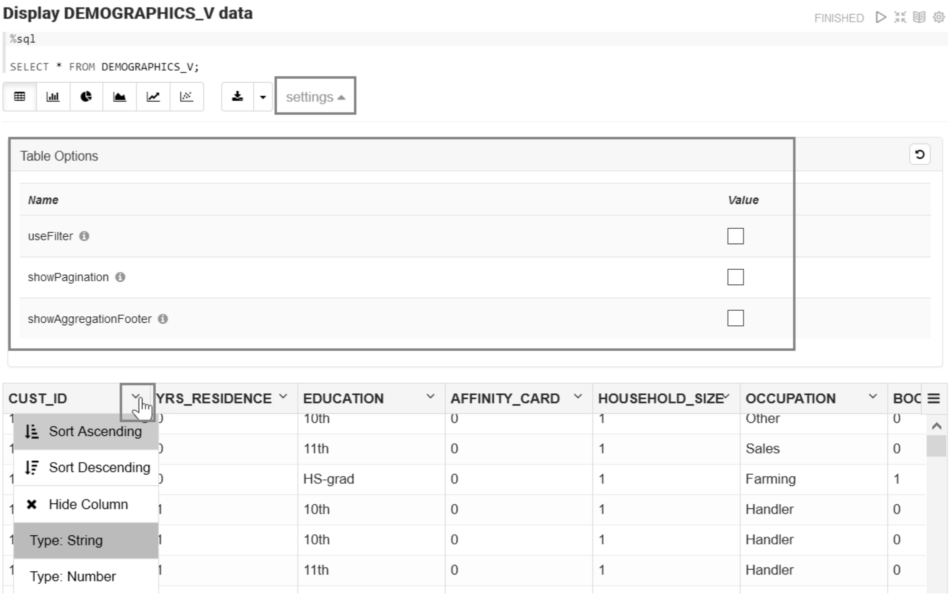
4. The data is displayed in the output of the paragraph.

   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;
```

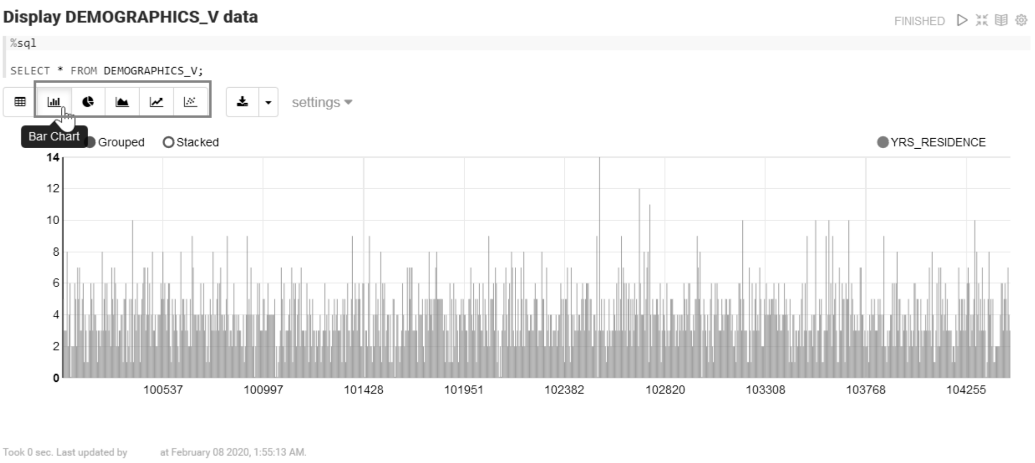
Table Options

Name	Value
useFilter	☐
showPagination	☐
showAggregationFooter	☐

CUST_ID	YRS_RESIDENCE	EDUCATION	AFFINITY_CARD	HOUSEHOLD_SIZE	OCCUPATION	BOC
1	0	10th	0	1	Other	0
1	0	11th	0	1	Sales	0
1	0	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.

Related Topics

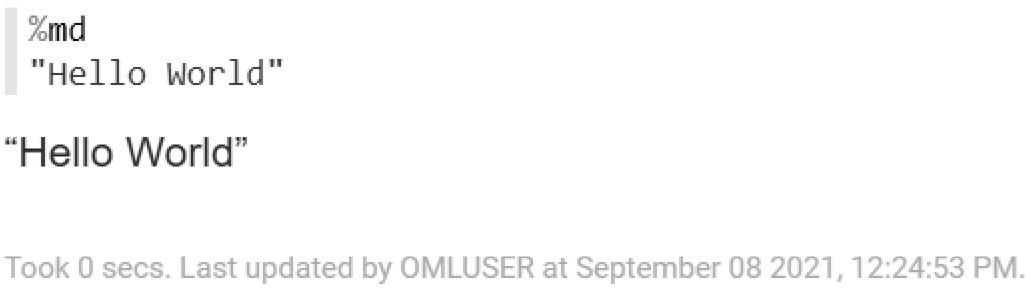
- Edit Oracle Database Interpreter Connection
  When defining an Oracle Database interpreter connection, a reference to a compute resource is created. This reference contains all connection-related information about the interpreter.
- Edit Oracle Database Interpreter Connection
  When defining an Oracle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

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.

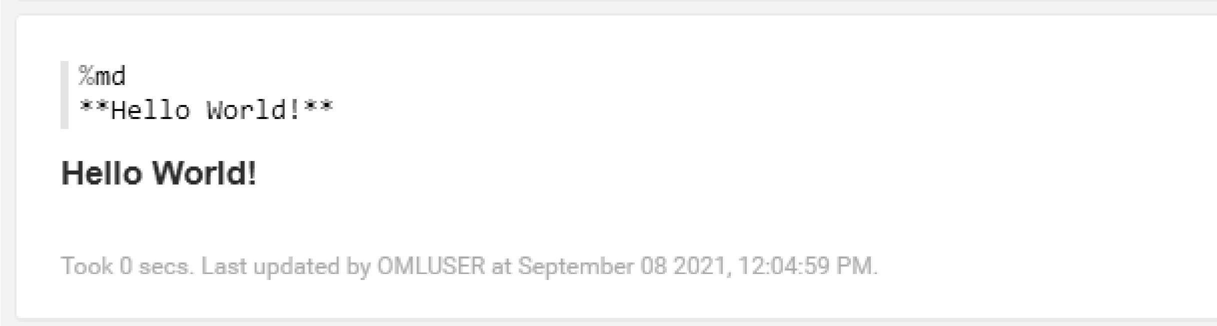


```
%md
"Hello World"
```

“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.

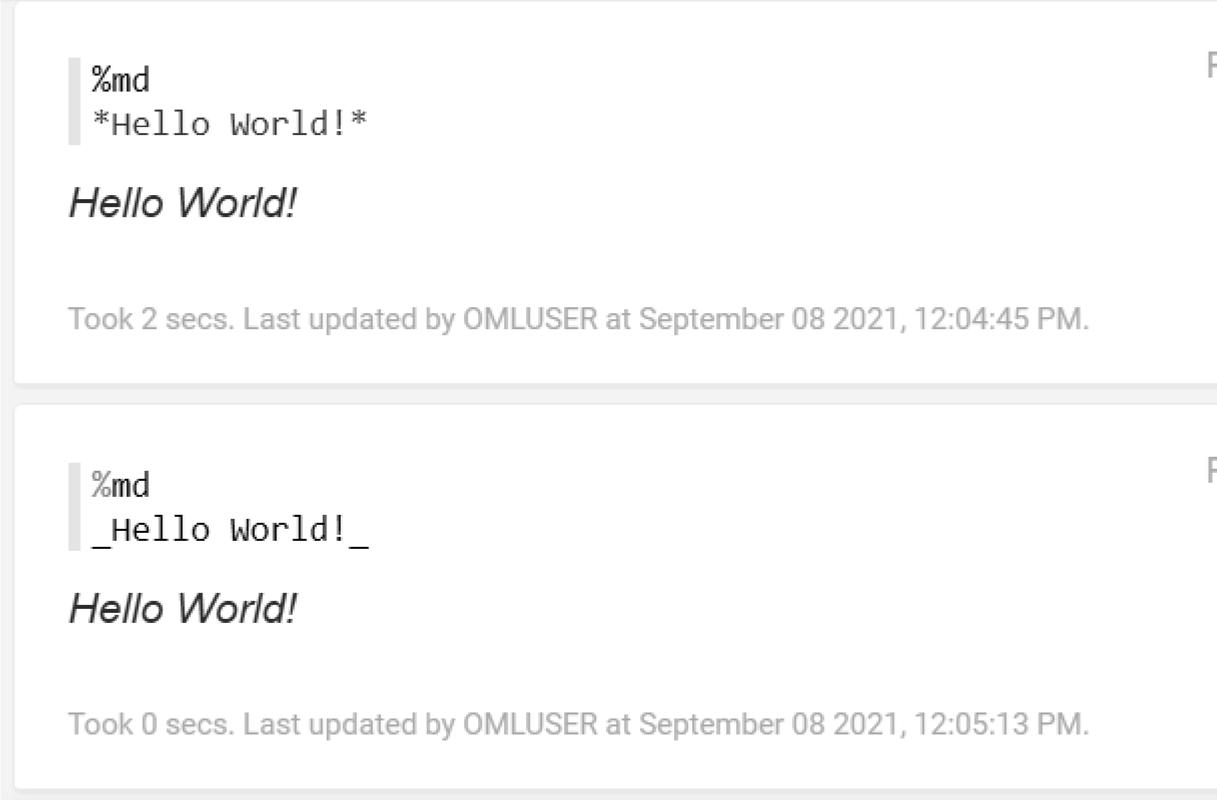


```
%md
**Hello World!**
```

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!

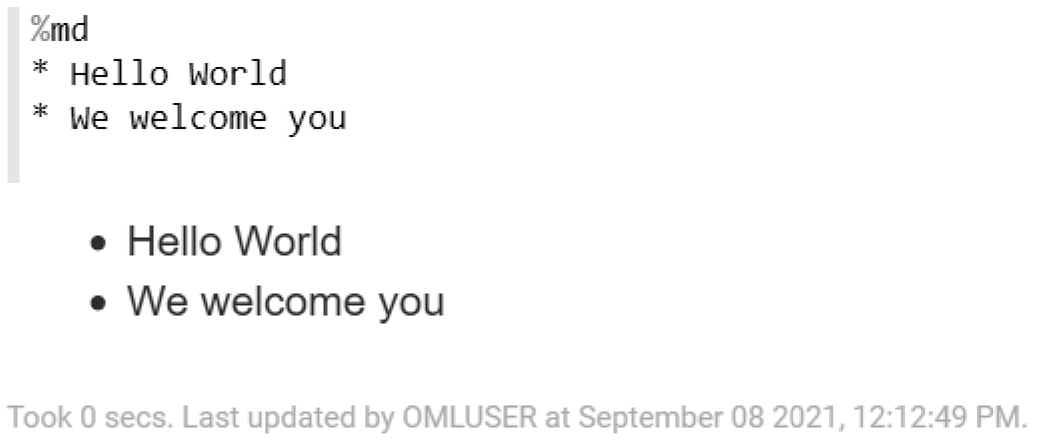
Took 2 secs. Last updated by OMLUSER at September 08 2021, 12:04:45 PM.

```
%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:

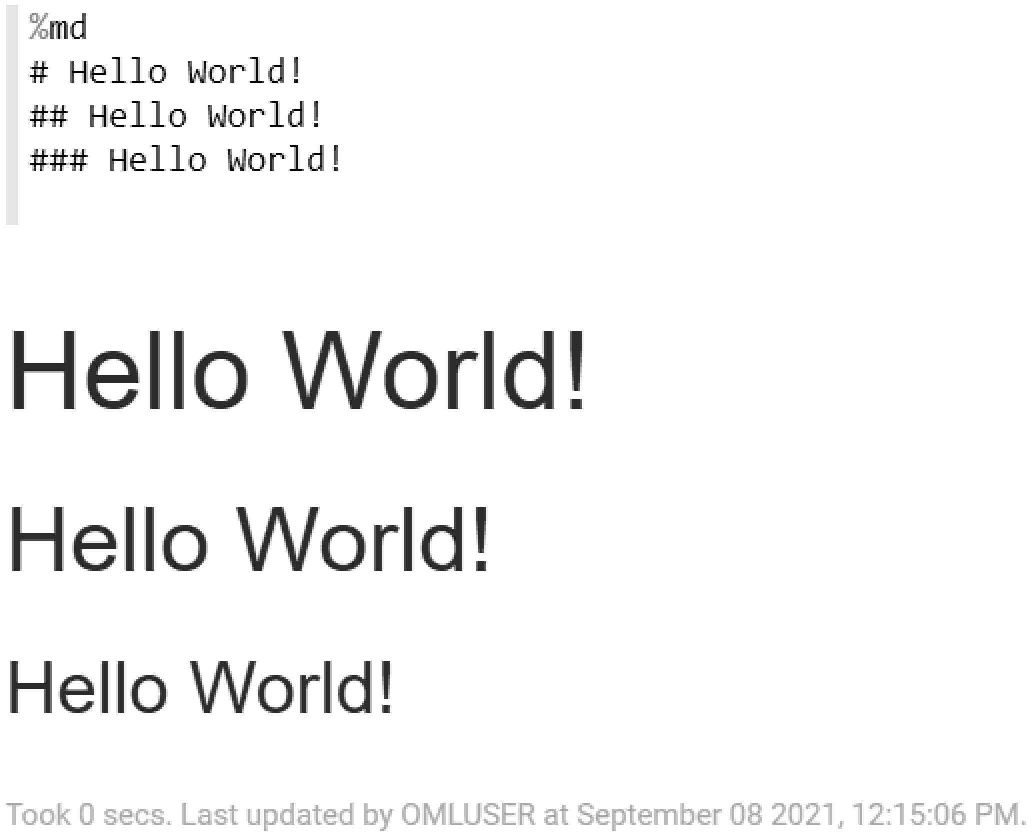


```
%md
* Hello World
* We welcome you
```

- 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!

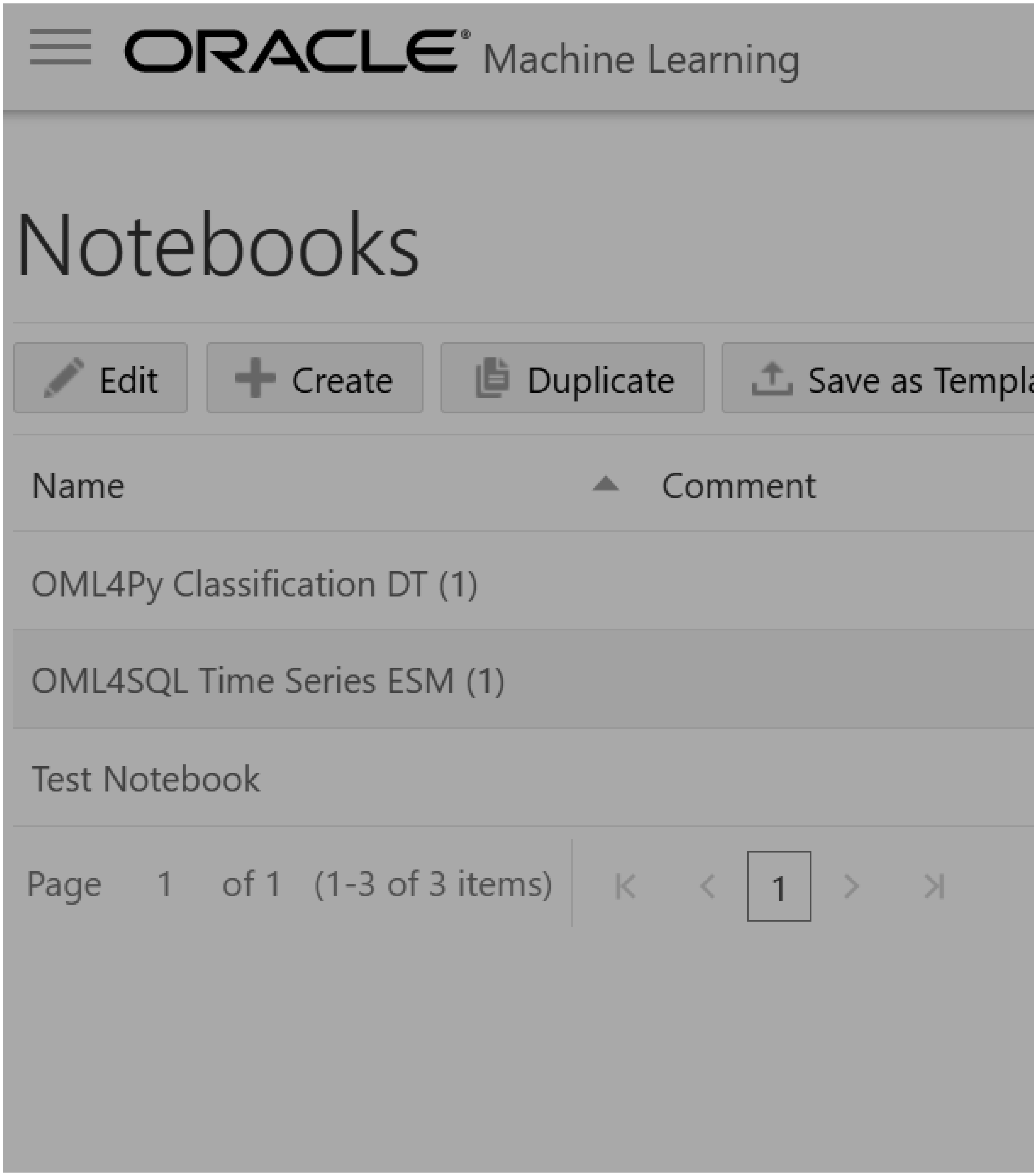
Took 0 secs. Last updated by OMLUSER at September 08 2021, 12:15:06 PM.

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. In 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.

Import a Notebook

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

Oracle Machine Learning 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. In Oracle Machine Learning home page, click **Notebooks.**
2. In 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. In the notebook 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

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:

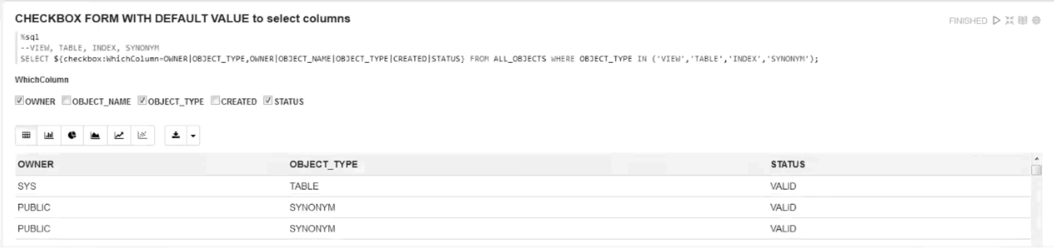
   ```
   ${checkbox:formName=defaultValue1|defaultValue2...,option1|option2...}
   ```

   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

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.

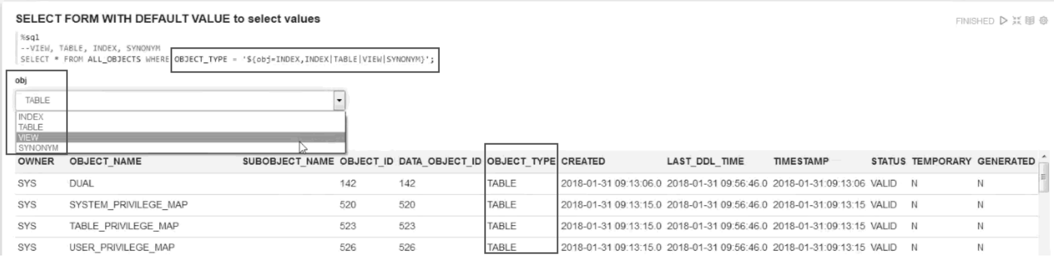
To create a Select form:

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

INDEX
TABLE
VIEW
SYNONYM

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.

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:

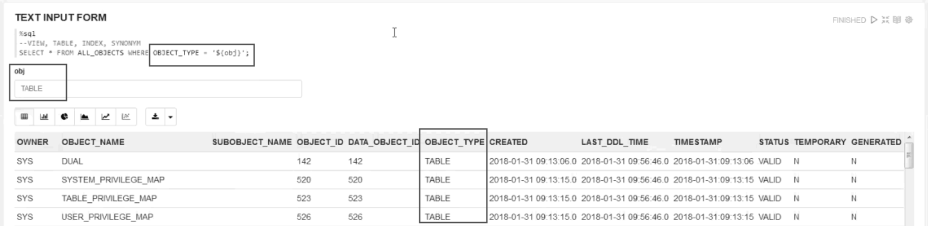
   ```
   SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = '${OBJ}';
   ```

   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.

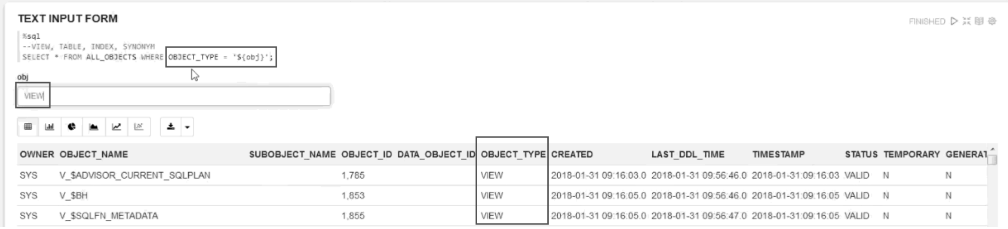
     TEXT INPUT FORM

     ```
     %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.



```
%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

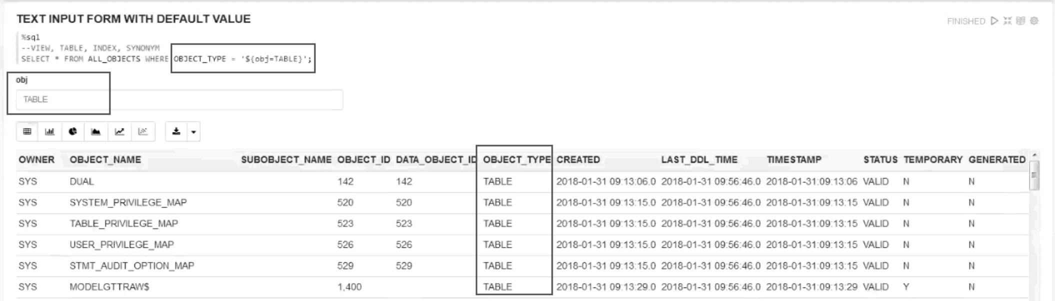
4. You can also assign default values in the form by using the syntax:

   `${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.



```
%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

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

   The Versions page opens.

2. In 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 in 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.

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. In 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.

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.

- Output Formats Supported by SET SQLFORMAT Command
  By using the `SET SQLFORMAT` command, you can generate the query output in a variety for formats.

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.

Use the Scratchpad

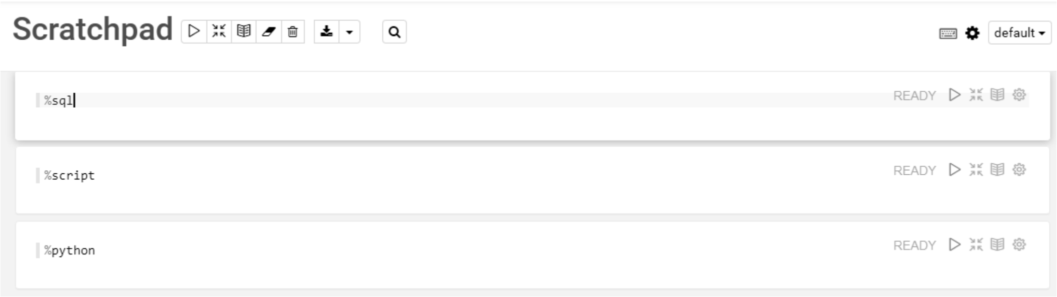
The Scratchpad provides you convenient one-click access to a notebook for running SQL statements, PL/SQL scripts, and Python scripts that can be renamed. The Scratchpad is available on the Oracle Machine Learning Notebooks home page.

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

After you run your scripts, the Scratchpad is automatically saved as a notebook by the default name **Scratchpad** in the Notebooks page. You can access it later in 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 Notebooks 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.

Figure 4-2 Scratchpad



```
%sql
```

```
%script
```

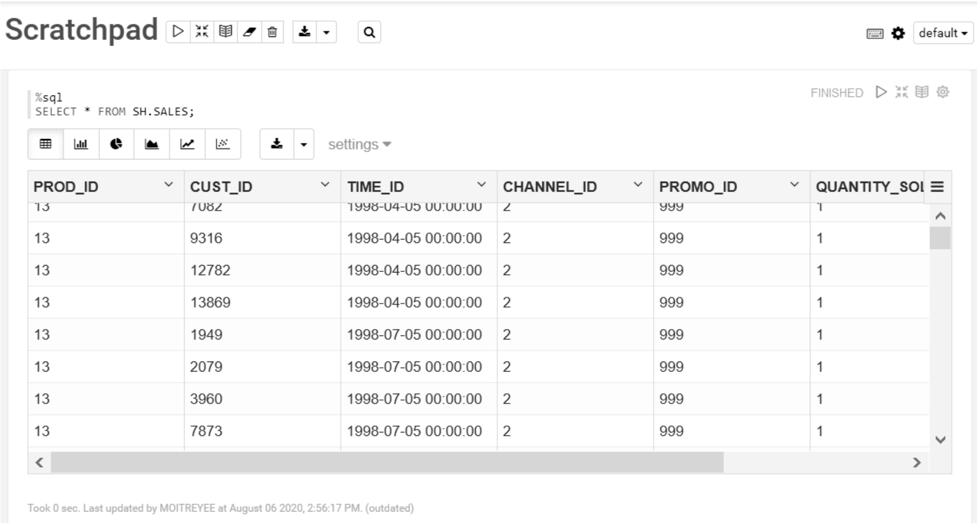
```
%python
```

2. To run SQL script:
   a. Go to the paragraph with the `%sql` invocation.
   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 fetches the relevant data from the database and displays it in a tabular format.

Figure 4-3 SQL Statement in Scratchpad



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

PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOl
13	7082	1998-04-05 00:00:00	2	999	1
13	9316	1998-04-05 00:00:00	2	999	1
13	12782	1998-04-05 00:00:00	2	999	1
13	13869	1998-04-05 00:00:00	2	999	1
13	1949	1998-07-05 00:00:00	2	999	1
13	2079	1998-07-05 00:00:00	2	999	1
13	3960	1998-07-05 00:00:00	2	999	1
13	7873	1998-07-05 00:00:00	2	999	1

Took 0 sec. Last updated by MOITREYEE at August 06 2020, 2:56:17 PM. (outdated)

3. To run PL/SQL script:

   a. Go to the paragraph with the `%script` invocation.

   b. Enter the following PL/SQL script and click the Run icon. Alternatively, you can press Shift+Enter keys to run the paragraph.

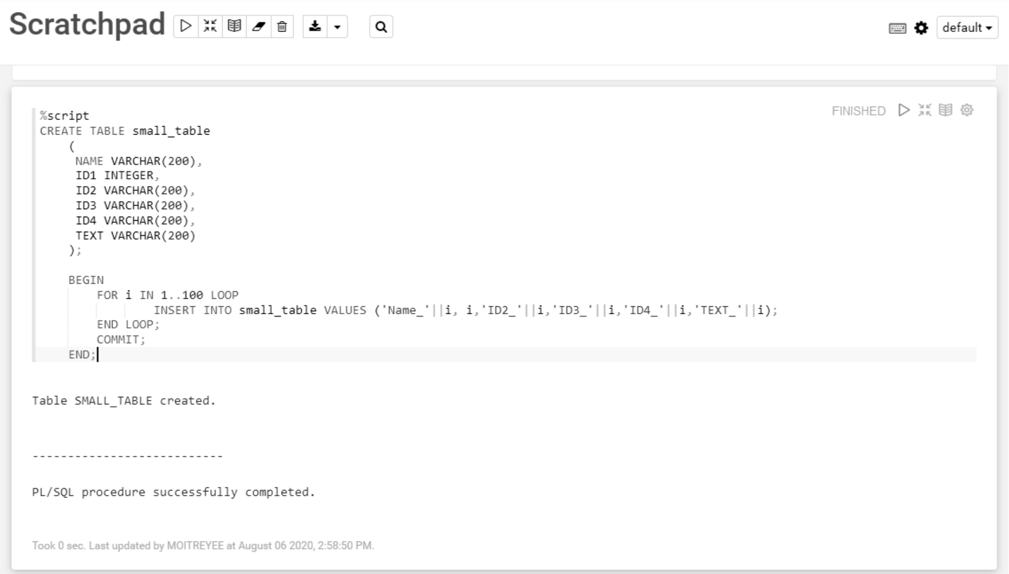
```
CREATE TABLE small_table
    (
     NAME VARCHAR(200),
     ID1 INTEGER,
     ID2 VARCHAR(200),
     ID3 VARCHAR(200),
     ID4 VARCHAR(200),
     TEXT VARCHAR(200)
    );

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

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

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

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

Figure 4-4   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.
```

Took 0 sec. Last updated by MOITREYEE at August 06 2020, 2:58:50 PM.

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.

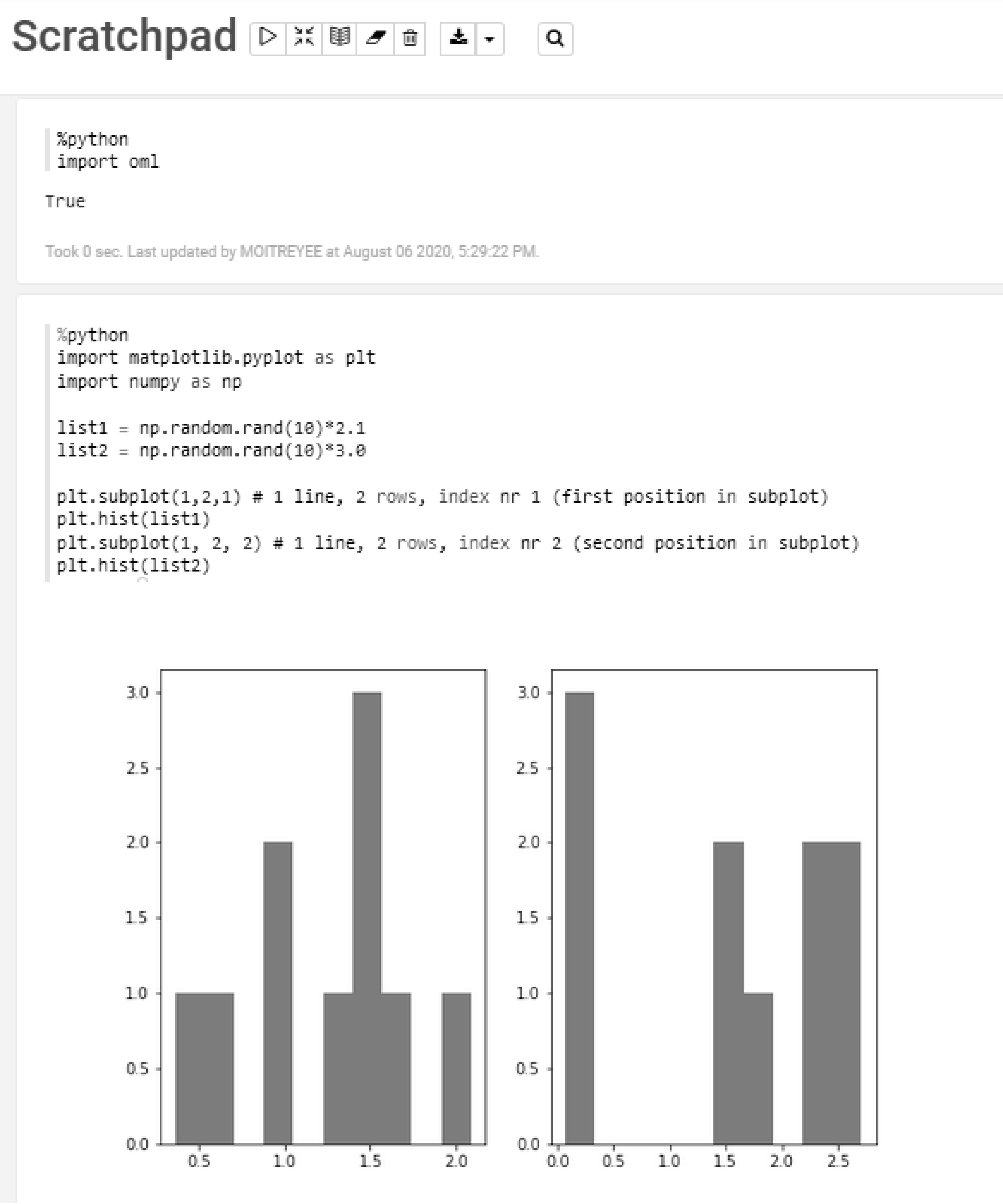
      ```
      import matplotlib.pyplot as plt
      import numpy as np

      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 4-5 Python Script in Scratchpad



```
%python
import oml
```

```
True
```

Took 0 sec. Last updated by MOITREYEE at August 06 2020, 5:29:22 PM.

```
%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)
```

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** in the Notebooks page. You can edit the name of the notebook and save it with the new name by clicking **Edit.**

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, you use the `sql` and `pl/sql` interpreters within an Oracle Database interpreter group and the `md` (MarkDown) interpreter for plain text formatting syntax so that it can be converted to HTML.

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.

- 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.

- 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 and PL/SQL interpreters for any specific paragraph in the notebook.
- Verify Interpreter Bindings
  After setting and changing the order of interpreter bindings, you can verify the interpreter bindings whether you are using SQL or Python interpreter or both in a given notebook. You use a SQL statement to view and verify the interpreter binding information about your notebook.

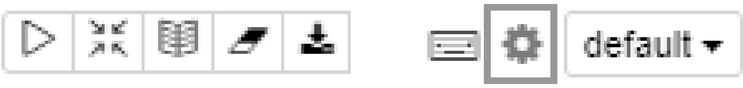
Set Interpreter Bindings for Notebooks

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.

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

1. In 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 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.

   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

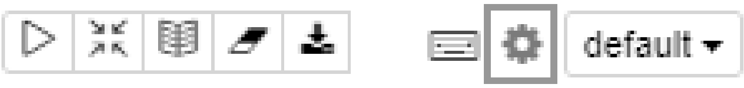
Change Interpreter Bindings for Specific Paragraphs in a Notebook

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 and PL/SQL interpreters for any specific paragraph in the notebook.

> **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.

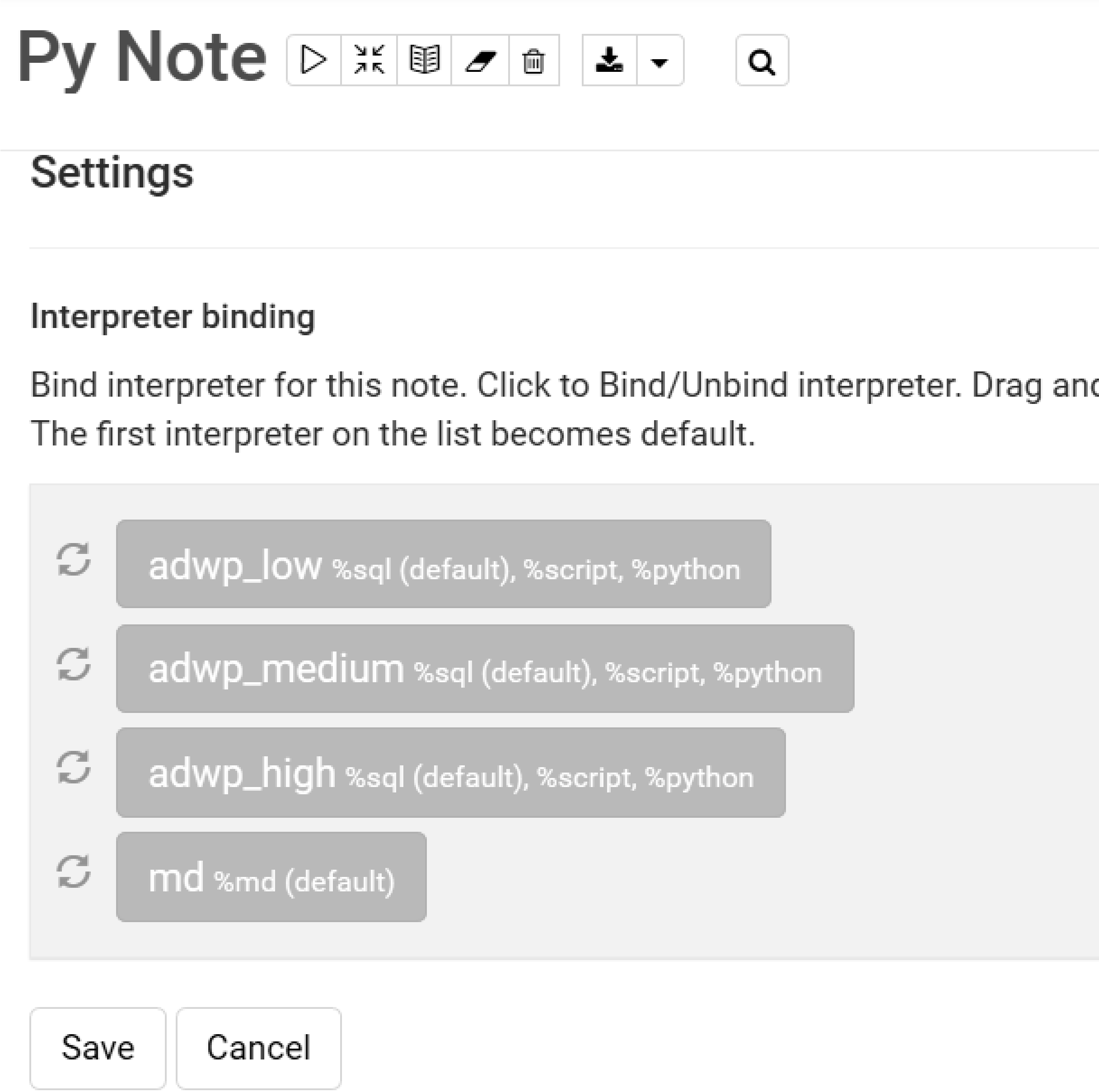


In this example, all the three SQL interpreters are bound to the notebook, and the interpreter with `low` resource allocation `adwpcwdp_low %sql` is the default, as it is the first interpreter on the list. The MarkDown interpreter is not bound to the notebook

> **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`
- adwp_medium `% sql(default),%script, %python`
- adwp_high `% sql(default), %script,%python.`
- md `%md(default)`



Settings

Interpreter binding

Bind interpreter for this note. Click to Bind/Unbind interpreter. Drag and
The first interpreter on the list becomes default.

adwp_low %sql (default), %script, %python

adwp_medium %sql (default), %script, %python

adwp_high %sql (default), %script, %python

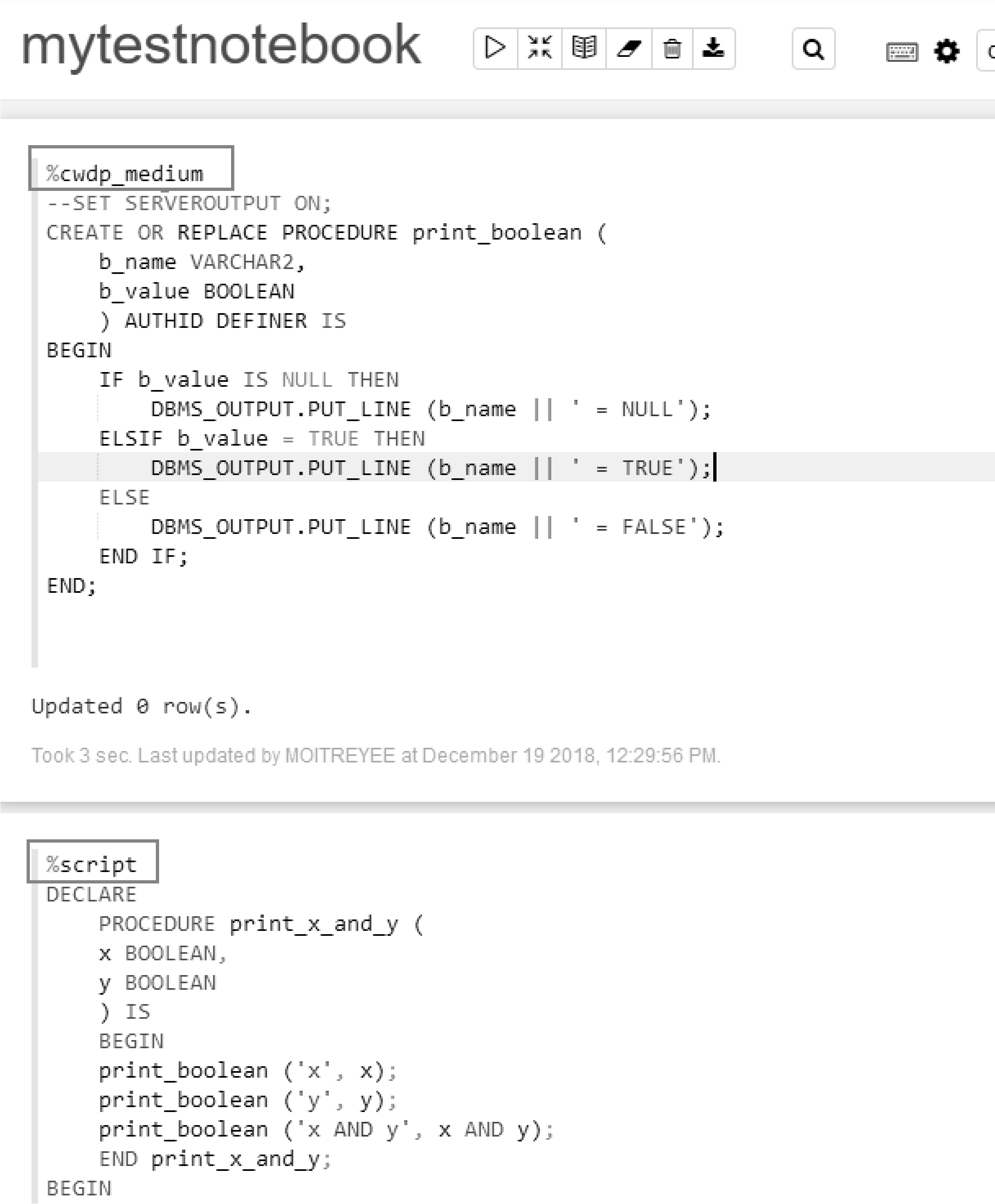
md %md (default)

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.

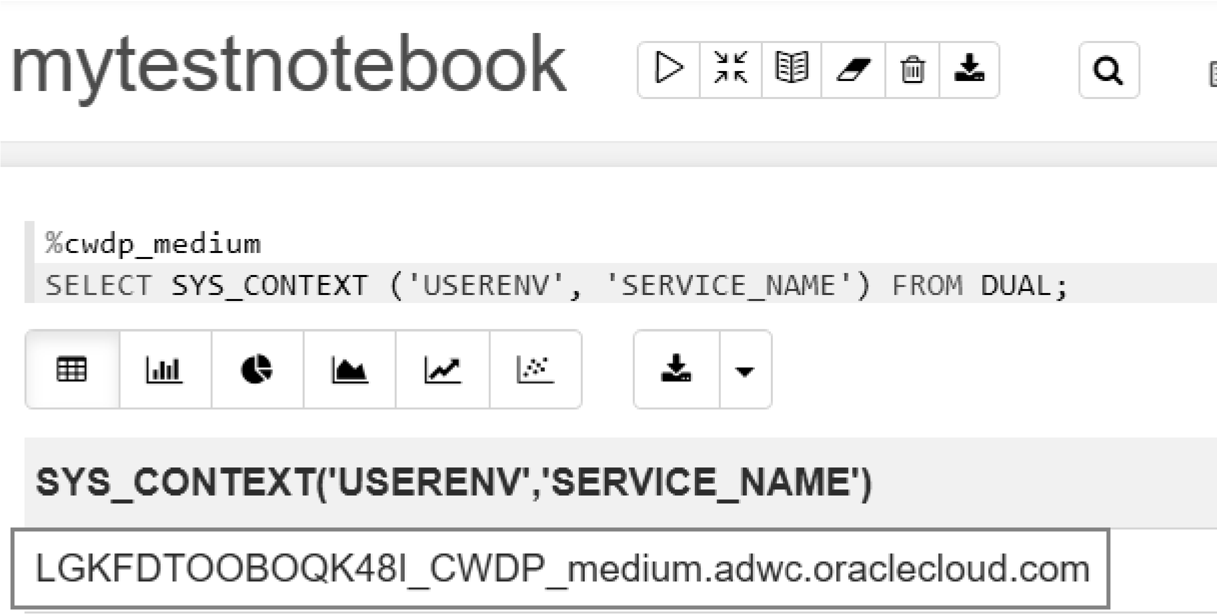
```
%script
DECLARE
    PROCEDURE print_x_and_y (
    x BOOLEAN,
    y BOOLEAN
    ) IS
    BEGIN
    print_boolean ('x', x);
    print_boolean ('y', y);
    print_boolean ('x AND y', x AND y);
    END print_x_and_y;
BEGIN
```

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
```



mytestnotebook

```
%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.

Verify Interpreter Bindings

After setting and changing the order of interpreter bindings, you can verify the interpreter bindings whether you are using SQL or Python interpreter or both 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:

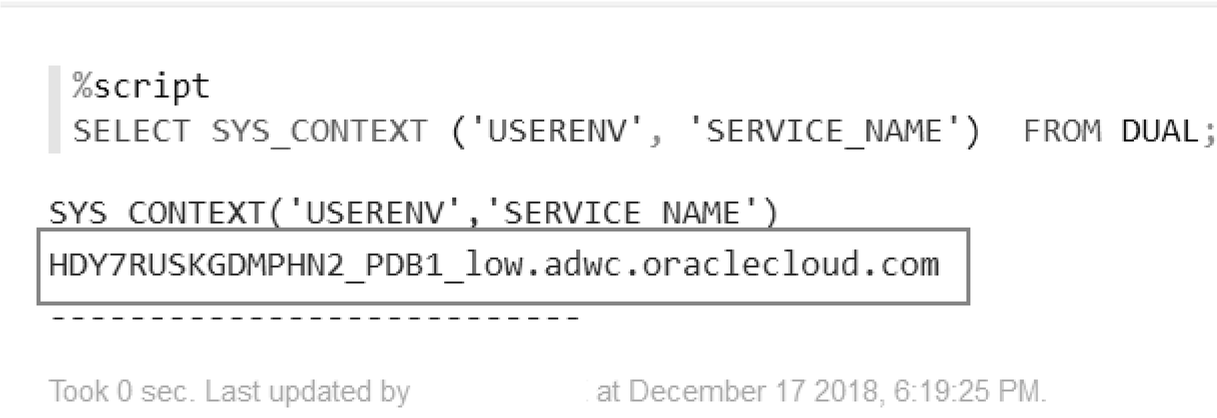
1. Open the notebook for which you want to check the interpreter binding.
2. Run the following SQL statement:

   ```
   %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.

Collaborate in Oracle Machine Learning

Two or more users can collaborate and share Oracle Machine Learning notebooks with other users.

You can collaborate by:

- Granting Access to Workspace of Another User
- Using the Export Option

- Using Oracle Machine Learning Notebook Templates

Collaborating in Oracle Machine Learning

5

Use Templates to Collaborate with Users

In the Oracle Machine Learning Templates UI, 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 Notebook 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 #unique_54

- 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.

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.

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. In 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 in the Notebooks page.

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.

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.**

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.

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. In 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 in the Notebooks page.

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.**

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
  In Oracle Machine Learning Example Templates, you can create a notebook from the available templates.

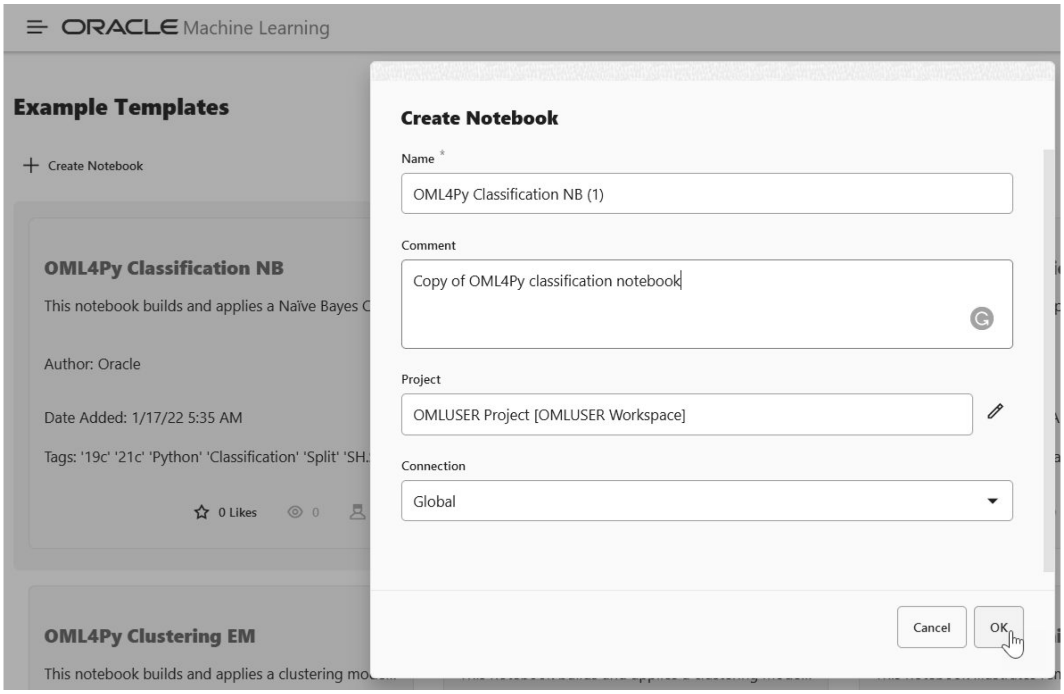
Create a Notebook from the Example Templates

In Oracle Machine Learning Example Templates, you can create a notebook from the available templates.

To create a notebook:

1. In 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** window, the name of the selected template appears. In the **Name** field, you can change the notebook name.


ORACLE Machine Learning
Example Templates
Create Notebook
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.
0 Likes
0
OML4Py Clustering EM
This notebook builds and applies a clustering mo...
Cancel
OK

Create Notebook

Name *
OML4Py Classification NB (1)

Comment
Copy of OML4Py classification notebook

Project
OMLUSER Project [OMLUSER Workspace]

Connection
Global

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 in the Notebooks page.

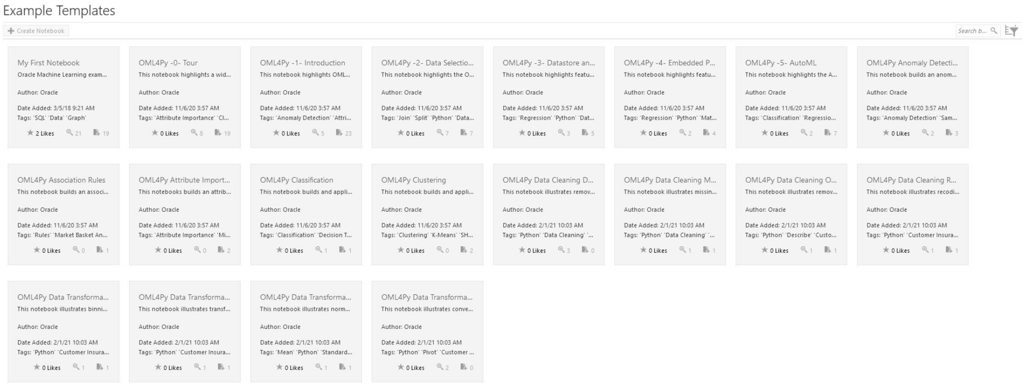
- Example Templates
  Oracle Machine Learning Notebooks provide you the following notebook Example templates that are based on different machine learning algorithms. The Example templates are processed in Oracle Autonomous Database.

Example Templates

Oracle Machine Learning Notebooks provide you the following notebook Example templates that are based on different machine learning algorithms. The Example templates are processed in Oracle Autonomous Database.

You can create your notebook based on any of these templates:

Figure 5-1 Example Templates



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 Example Template 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 template 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 template 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 template notebook to understand the AutoML workflow in OML4Py. In this notebook, the `WINE` dataset from scikit-learn is used. Here, AutoML is used for classification on the `target` column, and for regression on the `alcohol` column.
- **OML4Py Data Cleaning Duplicates Removal:** Use this template 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.

> **Note:**
>
> The data set CUSTOMER_INSURANCE_LTV_PY has duplicated values artificially generated by **OML4SQL Noise** notebook, which must be run before running this notebook.

- **OML4Py Data Cleaning Missing Data:** Use this template 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.

> **Note:**
>
> The data set CUSTOMER_INSURANCE_LTV_PY has missing values artificially generated by **OML4SQL Noise** notebook, which must be run before running this notebook.

- **OML4Py Data Cleaning Outlier Removal:** Use this template 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 Cleaning Recode Synonymous Values:** Use this template 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.

> **Note:**
>
> The data set CUSTOMER_INSURANCE_LTV_PY has recoded values artificially generated by **OML4SQL Noise** notebook, which must be run before this notebook. Specifically, MARITAL_STATUS has divorced status coded as DIV, married coded as M.

- **OML4Py Data Transformation Binning:** Use this template 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 template 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 template 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 template 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 template 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 template 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 template 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 template 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 template 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 Dataset Creation:** Use this template notebook to create dataset 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 template 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 template 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 template notebook to invoke 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 template notebook to predict numerical values using multiple regression.
- **OML4Py Statistical Functions:** Use this template 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 template 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.
- **OML4SQL Anomaly Detection:** Use this template 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 template 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 (ADB).

- **OML4SQL Association Rules:** Use this template 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 template 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 template notebook to identify key factors, also known as attributes, predictors, variables that have the most influence on a target attribute. This template notebook builds an Attribute Importance model, which uses the Minimum Description Length algorithm, using the `SH` schema data. All functionality executes inside Oracle Autonomous Database.
- **OML4SQL Classification - Predicting Target Customers:** Use this template 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 template notebook to identify natural clusters of customers. Oracle Machine Learning supports clustering using several algorithms, including k-Means, O-Cluster, and Expectation Maximization. This template 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 template 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 template 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 template 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 template 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 template 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 template 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 template 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 template notebook to perform dimensionality reduction using the in-database singular value decomposition (SVD) algorithm.
- **OML4SQL: Exporting Serialized Models:** Use this template 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 (ADB). These endpoints enable the storing of 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 template 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 template 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 template 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 template 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 template 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 theOML4SQL Noise notebook before the Data Preparation notebooks.

- **OML4SQL Partitioned Model:** Use this template 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 template 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 template notebook to predict numerical values. This template uses multiple regression algorithms such as Generalized Linear Models (GLM).
- **OML4SQL Statistical Function:** Use this template notebook for descriptive and comparative statistical functions. The notebook template uses `SH` schema data.
- **OML4SQL Time Series:** Use this template 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.

6

Get Started with Jobs

Jobs allow you to schedule the running of notebooks. In 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.

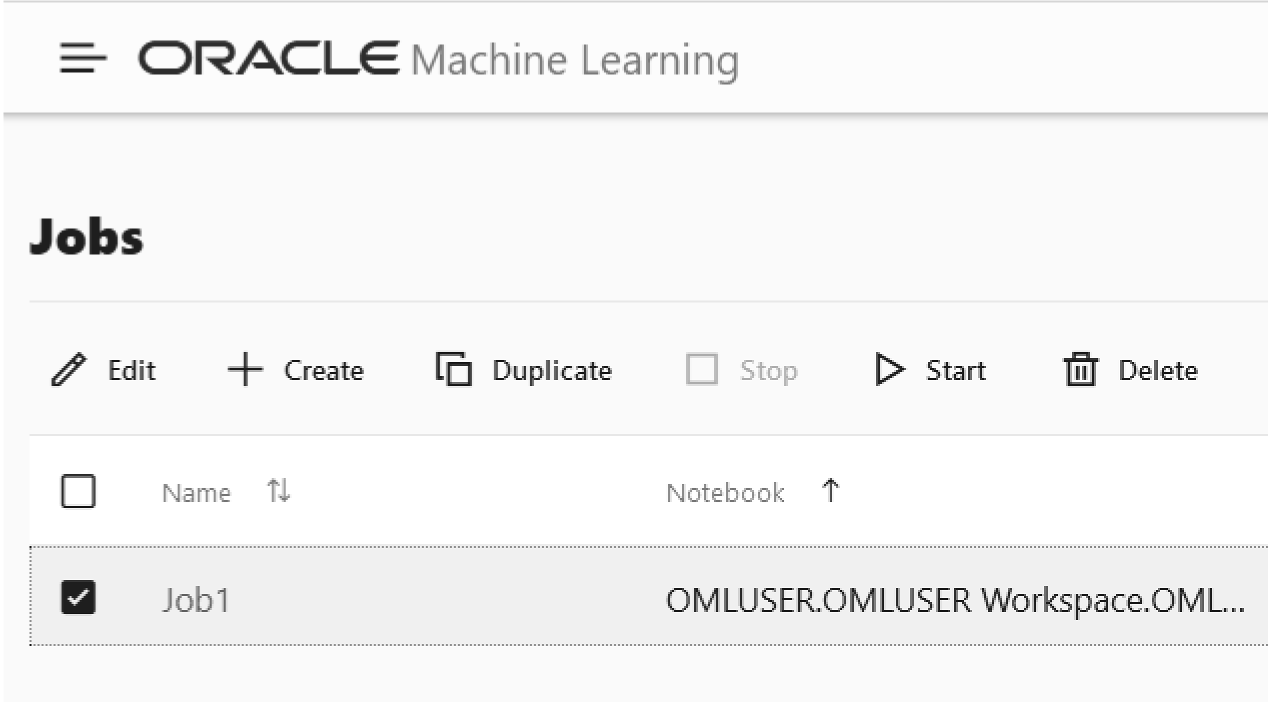
- About Jobs
  The Jobs page lists all the jobs created, along with the job name, notebook, owner of the job, last start date, next run date, status, and schedule.
- Create Jobs to Schedule Notebooks
  You can create jobs to schedule your notebook with preferred scheduling settings.
- View Job Logs
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

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 6-1 Jobs



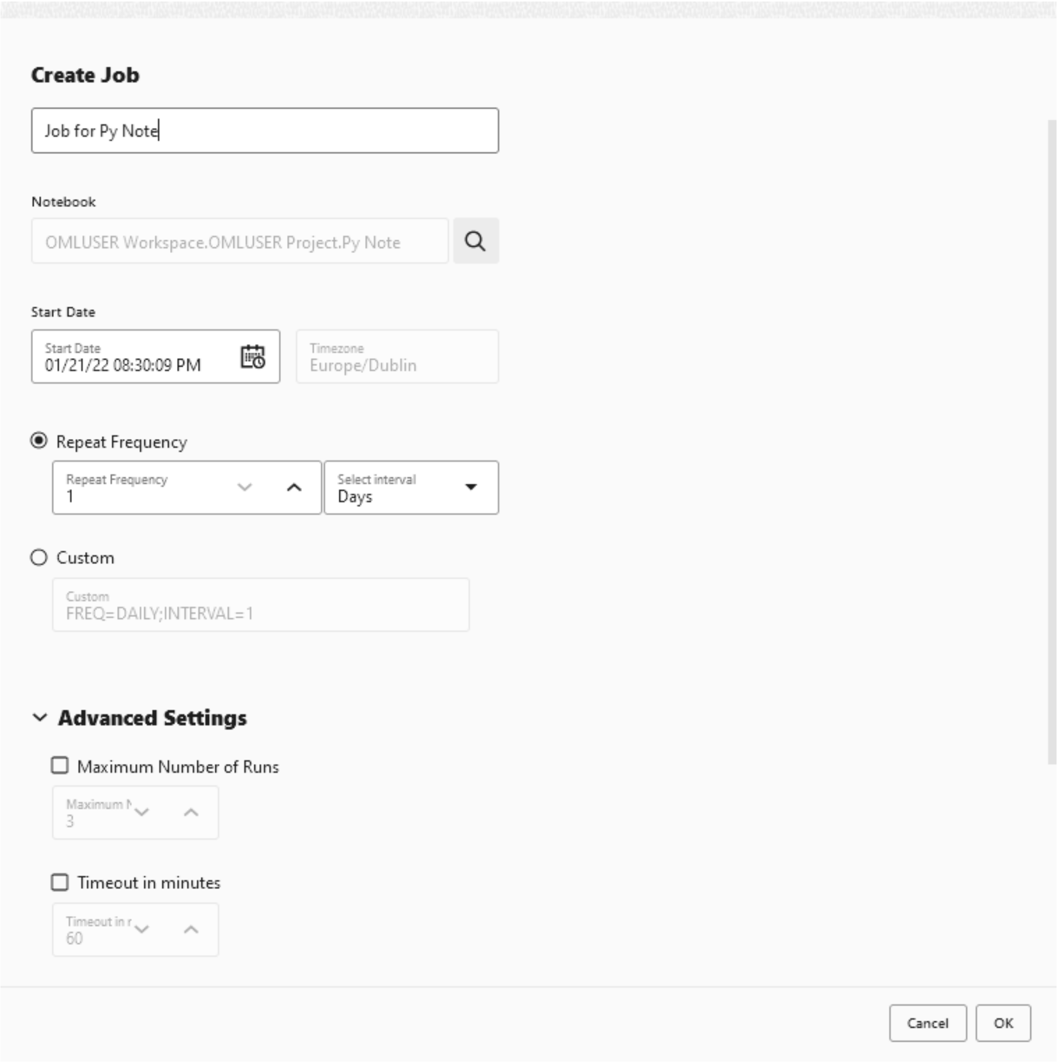
- Edit jobs: You can edit the metadata of any job listed in the Jobs page. Click **Edit** to edit the selected job.
- Create jobs. Click **Create** to create a new job to schedule your Notebook.
- Duplicate jobs: You can create a copy of an existing job listed in the Jobs page. Click **Duplicate** to 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 in the Jobs page.

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. In 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.**

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.

7

Get Started with Connection Groups

A connection group, also known as a Zeppelin interpreter set, is a collection of database connections.

- About Connection Groups
  In 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.

About Connection Groups

In 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 in 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.

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 in 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.

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.**

8

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.

In 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.

9

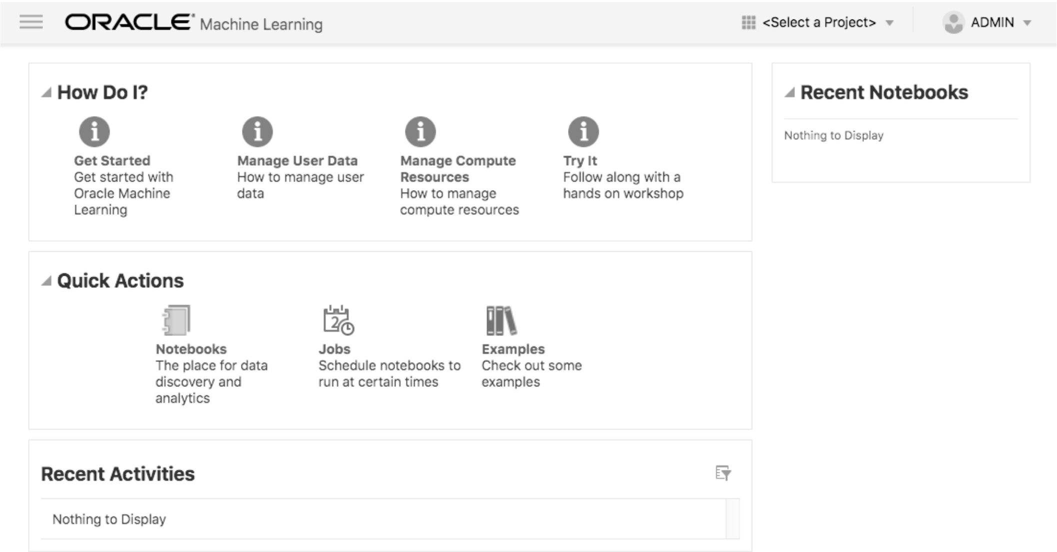
Administer Oracle Machine Learning

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

- Administrator — Creates, edits, and deletes Oracle Machine Learning user accounts. The Administrator reassigns user workspace.

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

  Figure 9-1 Admin Home page

  How Do I?

  Get Started
  Get started with Oracle Machine Learning

  Manage User Data
  How to manage user data

  Manage Compute Resources
  How to manage compute resources

  Try It
  Follow along with a hands on workshop

  Recent Notebooks

  Nothing to Display

  Quick Actions

  Notebooks
  The place for data discovery and analytics

  Jobs
  Schedule notebooks to run at certain times

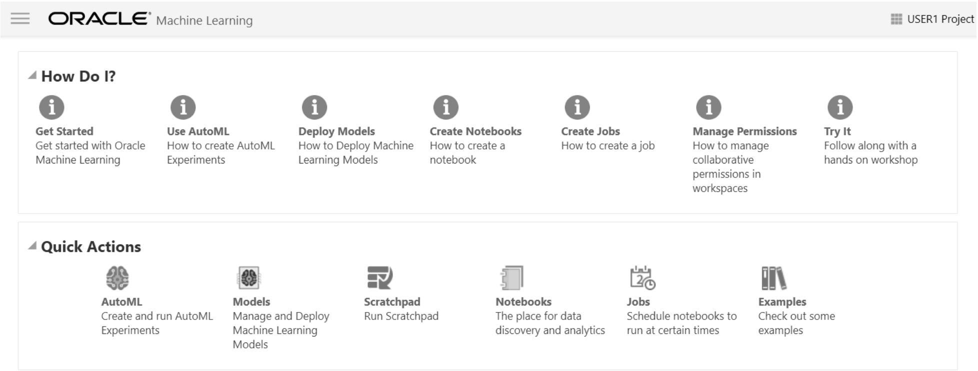
  Examples
  Check out some examples

  Recent Activities

  Nothing to Display

- 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, create and run AutoML experiments, and deploy models.

Figure 9-2 Developer Home page



ORACLE Machine Learning

USER1 Project

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

- Typical Workflow for Managing Oracle Machine Learning
  To manage Oracle Machine Learning and other administrative tasks, refer to the tasks listed in the table as a guide.
- 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
  In 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.

Typical Workflow for Managing Oracle Machine Learning

To manage Oracle Machine Learning and other administrative tasks, refer to the tasks listed in the table as a guide.

Tasks	Oracle Machine Learning 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	Work with Connection Groups
Compute Resource — View	Oracle Machine Learning	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	About User Data

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

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 a new user account and user credentials for Oracle Machine Learning in the User Management interface.
- Add Existing Database User Account to Oracle Machine Learning
  An administrator adds an existing database user account for Oracle Machine Learning in the User Management interface.

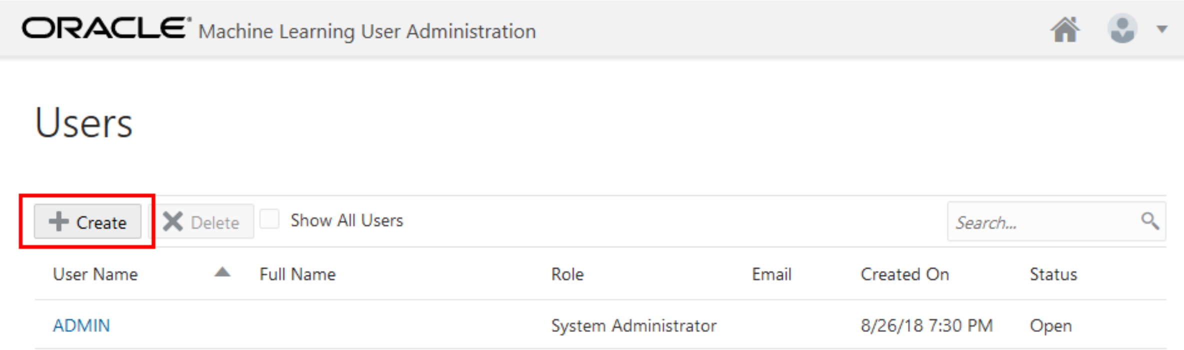
Create User

An administrator creates a new user account 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. Select an Autonomous Data Warehouse instance and on the details page click **Service Console**.
2. On the Service Console click **Administration**.
3. Click **Manage OML Users** to open the Oracle Machine Learning User Administration page.
4. Click **Create** on the Oracle Machine Learning User Administration page.



User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		8/26/18 7:30 PM	Open

5. In the **Username** field, enter a username for the account. Using the username, the user will log in to an Oracle Machine Learning instance.
6. Enter a name in the **First Name** field.
7. Enter a name in the **Last Name** field.

8. In the **Email Address** field, enter the email ID of the user.
9. Select the option **Generate password and email account details to user. User will be required to reset the password on first sign in.** to auto generate a temporary password and send an email with the account credentials to the user.

   If you select this option, you need not enter values in the **Password** and **Confirm Password** fields; the fields are grayed out.
10. In the **Password** field, enter a password for the user, if you choose to create a password for the user.

    This option is disabled if you select the **Generate password...** option to auto generate a temporary password for the user.
11. In the **Confirm Password** field, enter a password to confirm the value that you entered in the **Password** field.

    By doing so, you create the password for the user. The user can change the password when first logging in.
12. Click **Create**.

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.

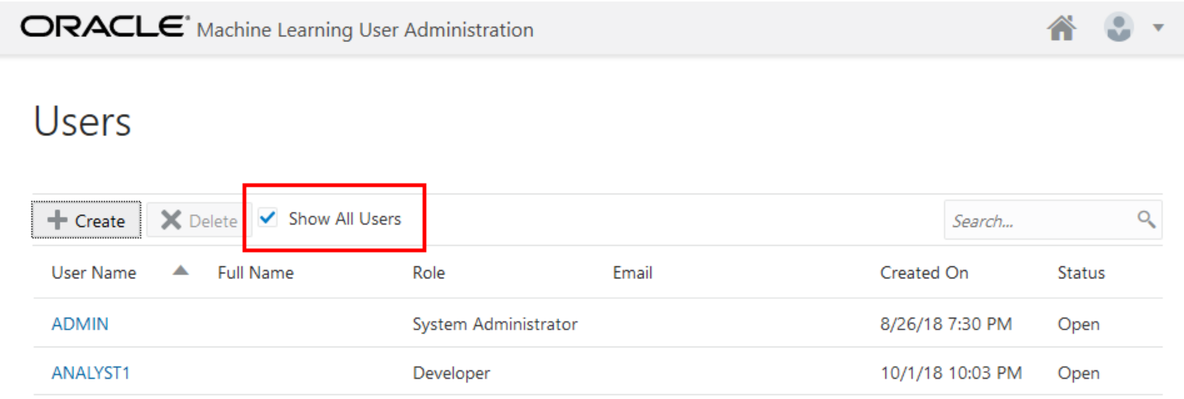
Add Existing Database User Account to Oracle Machine Learning

An administrator adds an existing database user account 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 add an existing database user account:

1. Select an Autonomous Database instance, and on the details page click **Service Console**.
2. On the Service Console, click **Administration**.
3. Click **Manage OML Users** to add Oracle Machine Learning users.
4. 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.

5. 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.
6. Enter a name in the **First Name** field. (Optional)
7. Enter the last name of the user in the **Last Name** field. (Optional)
8. 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 a Oracle Machine Learning user.
9. 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.

About User Data

In 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.

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.

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 in the **Oracle Resources** page.

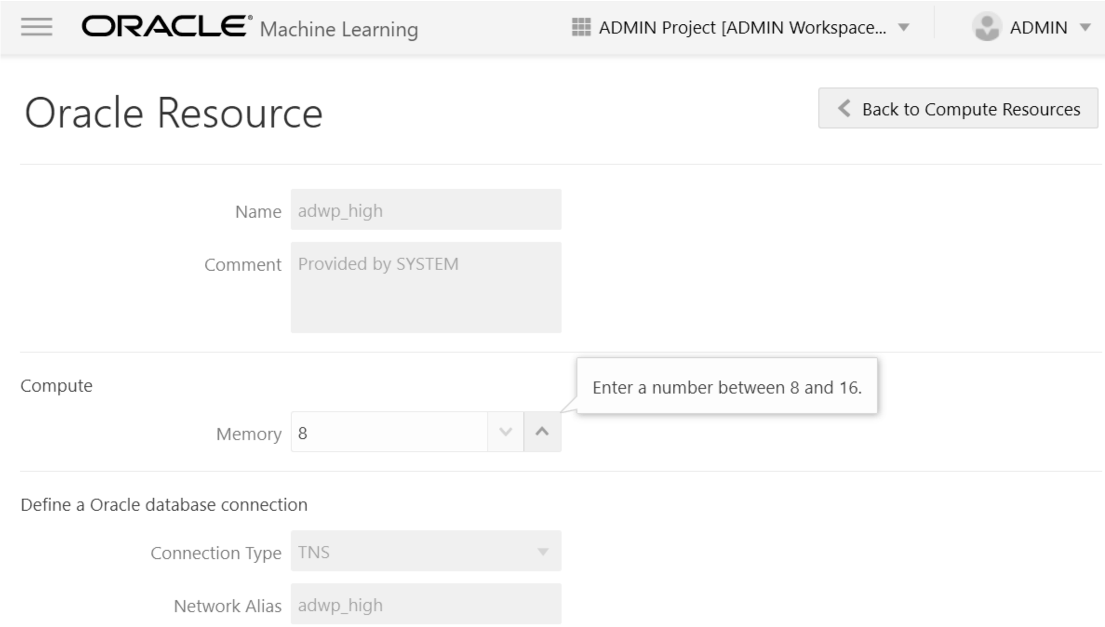
- Oracle Resource
  The Oracle Resource page displays the details of the selected compute resource on the Compute Resources page. You can configure the memory settings (in Gigabytes) for the Python interpreter for the selected compute resource.

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 9-3 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.

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 9-1 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 9-1 (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 9-1 (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 9-1 (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 9-1 (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 9-1 (Cont.) Resource Services and Number of Notebooks

Resource Service	OCPUs (Oracle CPUs)	Memory	Number of Concurrent Notebooks
			er of notebooks run would be 5 (1.25x4) in Medium level.
Low	1	2 GB (up to 4 GB)	Up to 100