

Oracle Machine Learning and Auto ML

Demo/Workshop

Move the Algorithms – Not the Data



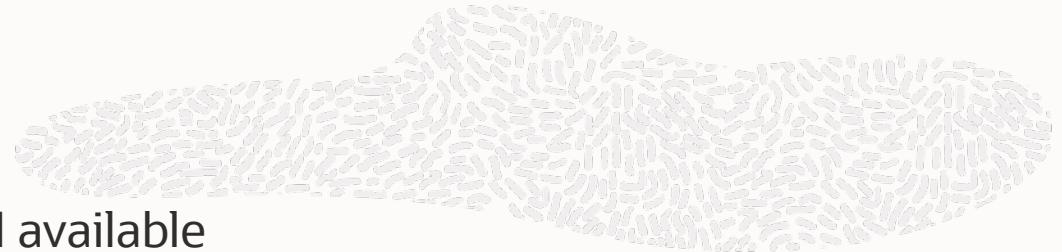
Renée Wikestad
Cloud Specialist Engineer,
Oracle Norge



May 27th 2021

O

Goals



- Test drive the new Oracle Machine Learning AutoML UI available with Oracle Autonomous Database
- Build models with OML AutoML UI and deploy them directly to OML Services
- Auto-generate OML4Py notebooks that can build the top models, with all Hyperparameters and options already set
- Use OML Services REST APIs to score

Create Your Free Trial Account & Sign in to Your Account. Provision an Autonomous Database



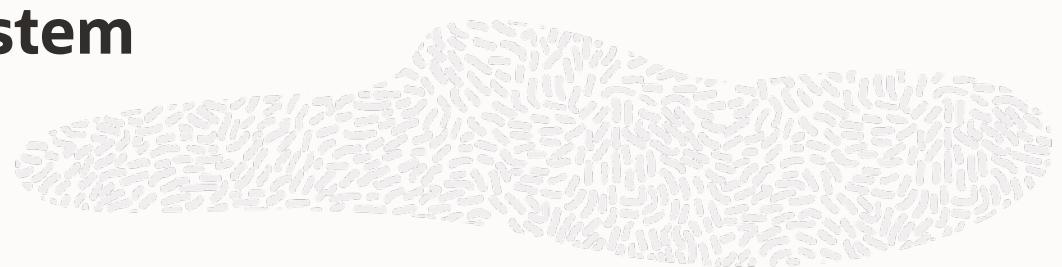
- **STEP 1:** [Create free Trial Account](#)
- **STEP 2:** [Sign in to Your Account](#)
- **STEP 3:** [Provision an Autonomous Database](#) (Here you can use ATP instead of ADW)

(Or you can reserve a workshop environment on **LiveLabs**:

OML4Py LiveLab: <https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/view-workshop?wid=786>)

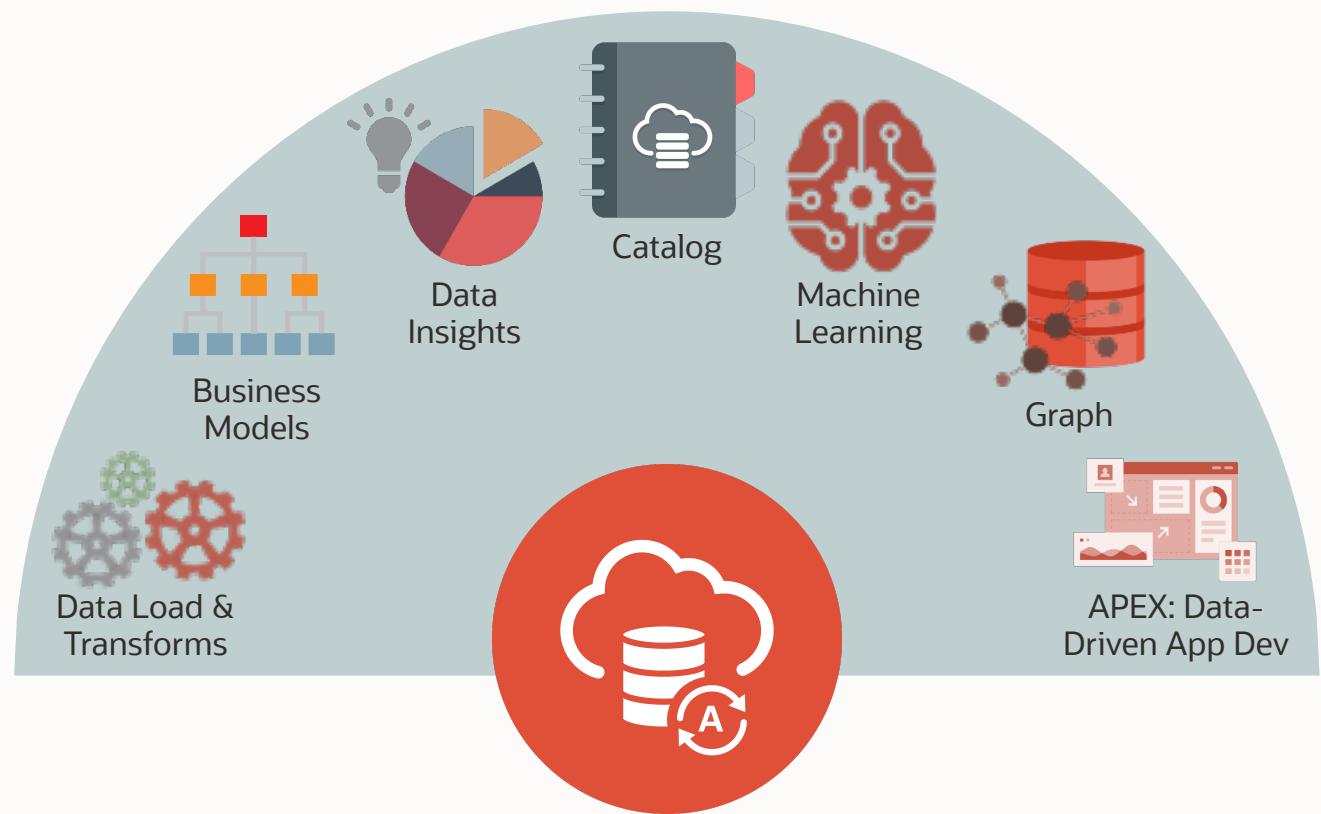
Oracle Autonomous Database is an ecosystem

Autonomous Database **with built-in tool suite**



Data Sources

Applications
ORACLE Fusion Applications ORACLE Siebel ORACLE PeopleSoft
Oracle EBS Oracle NetSuite SAP salesforce
Databases
File
Files
Streaming Services
Data Lake
OCI Object Storage Amazon S3 Microsoft Azure Blob Storage



Visualization & Development

Application Express	
Oracle ML Notebook	
Oracle Analytics Cloud	
IDEs	
Spreadsheets	
3rd Party BI Tools	

Oracle Machine Learning Algorithms and Analytics

CLASSIFICATION

- Naïve Bayes
- Logistic Regression (GLM)
- Decision Tree
- Random Forest
- Neural Network
- Support Vector Machine (SVM)
- Explicit Semantic Analysis
- **XGBoost***

ANOMALY DETECTION

- One-Class SVM
- **MSET-SPRT***

CLUSTERING

- Hierarchical K-Means
- Hierarchical O-Cluster
- Expectation Maximization (EM)

TIME SERIES

- Forecasting - Exponential Smoothing
- Includes popular models
e.g. Holt-Winters with trends,
seasonality, irregularity, missing data

REGRESSION

- Linear Model
- Generalized Linear Model (GLM)
- Support Vector Machine (SVM)
- Stepwise Linear regression
- Neural Network
- **XGBoost***

ATTRIBUTE IMPORTANCE

- Minimum Description Length
- Principal Component Analysis (PCA)
- Unsupervised Pairwise KL Divergence
- CUR decomposition for row & AI

ASSOCIATION RULES

- A priori/ market basket

SQL ANALYTICS

- SQL Windows
- SQL Patterns
- SQL Aggregates

FEATURE EXTRACTION

- Principal Comp Analysis (PCA)
- Non-negative Matrix Factorization
- Singular Value Decomposition (SVD)
- Explicit Semantic Analysis (ESA)

ROW IMPORTANCE

- CUR Decomposition

RANKING

- **XGBoost***

TEXT MINING SUPPORT

- Algorithms support text columns
- Tokenization and theme extraction
- Explicit Semantic Analysis (ESA)

STATISTICAL FUNCTIONS

- min, max, median, stdev, t-test, F-test, Pearson's, Chi-Sq, ANOVA, etc.

R AND PYTHON PACKAGES

- Third-party R and Python Packages through Embedded Execution

* New in 21c



Oracle Machine Learning

OML Notebooks

OML4SQL

OML4R

OML4Py

Oracle Data Miner

OML4Spark

OML AutoML UI

OML Services

Interfaces for 3 popular
data science languages:
SQL, R, and Python

Code-free AutoML interface
on Autonomous Database

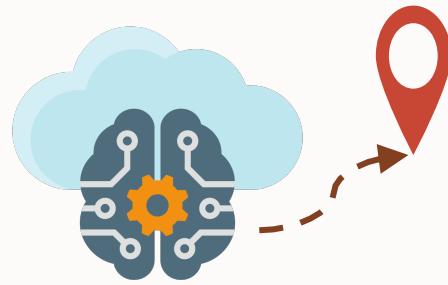
Model Deployment and
Management, Cognitive Text

Collaborative notebook
environment based on
Apache Zeppelin with
Autonomous Database

SQL Developer extension to
create, schedule, and deploy
ML solutions through a
drag-and-drop interface

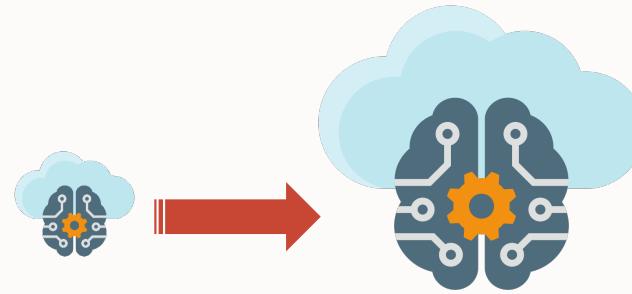
ML for the big data environment
from R with scalable algorithms

Oracle Machine Learning Key Attributes



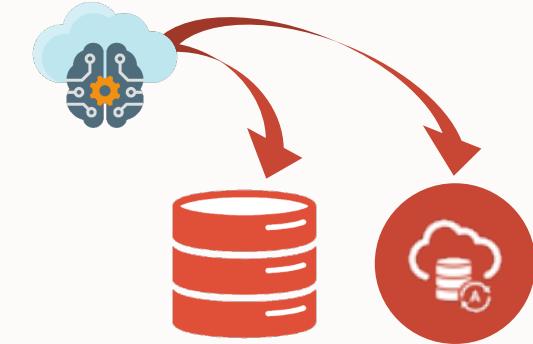
Automated

Get better results faster
with less effort –
even non-expert users



Scalable

Handle big data volumes using
parallel, distributed algorithms –
no data movement

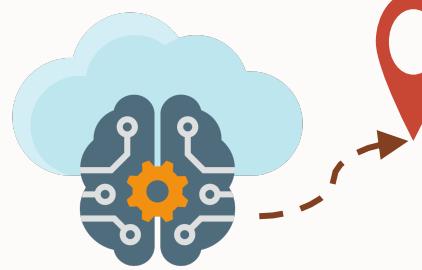


Production-ready

Deploy and update data
science solutions faster with
integrated ML platform

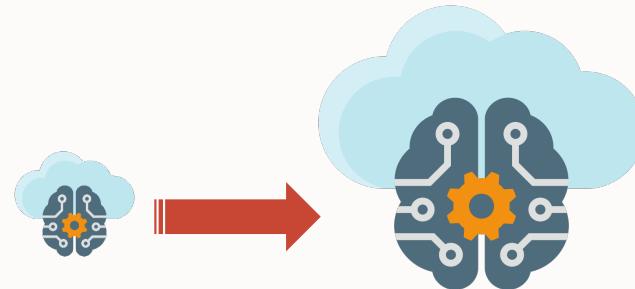
Increase productivity, Achieve enterprise goals, Innovate more

Oracle Machine Learning **Key Attributes**



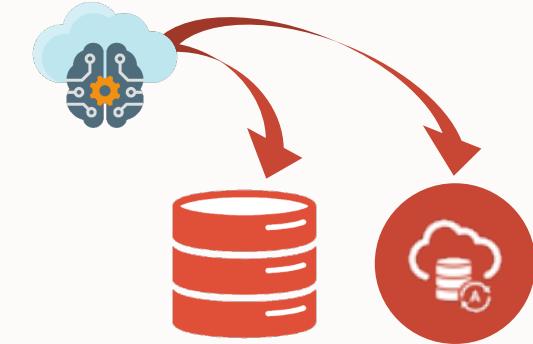
Automated

Get better results faster
with less effort –
even non-expert users



Scalable

Handle big data volumes using
parallel, distributed algorithms –
no data movement



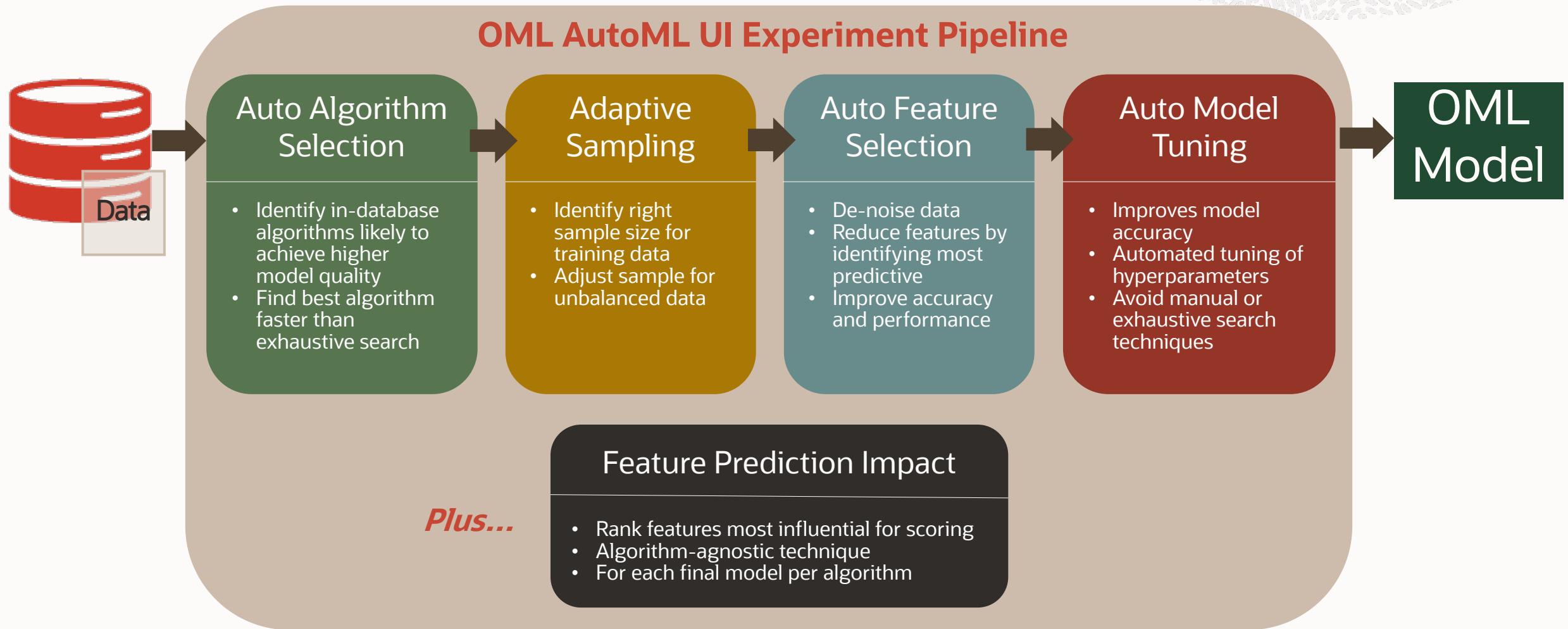
Production-ready

Deploy and update data
science solutions faster with
integrated ML platform

Increase productivity, Achieve enterprise goals, Innovate more

OML AutoML UI

Simplify the machine learning modeling and deployment process



OML AutoML UI

No-code AutoML-based user interface supporting automatic machine learning

Powerful, easy to use UI

Automates model building,
tuning, and deployment

- Enhance data scientist productivity
- Support model management
- Empower non-expert users

Featuring

- Minimal user input: data, target
- Model leaderboard
- Model deployment via REST endpoints
- Generate OML4Py notebooks from models

The image displays three screenshots of the Oracle Machine Learning AutoML UI. The top-left screenshot shows the main dashboard with sections for 'How Do I?' (links to AutoML, Get Started, Create Notebooks, Create Jobs, Manage Permissions, Try It), 'Recent Notebooks' (AffinityCard Notebook 1, SVM Gaussian Insurance Whatever RF Affinity Card test), and 'Recent Projects' (OMLUSER Project [OMLUSER Works...], MSPIVAK Project [MSPIVAK Works...]). The middle-left screenshot shows the 'Experiment: AutoML Experiment Demo' page with a 'Metric Chart' showing performance over time, a 'Leader Board' listing various models, and a 'Features' section. The bottom-right screenshot shows the 'Experiment: AutoML Experiment Demo' page with an 'Insight Options' dialog open, displaying 'SVM Linear 1' metrics like Prediction Impact, Lift, ROC, and Confusion Matrix. The bottom-left screenshot shows the 'Experiment: AutoML Experiment Demo' page with a 'Metric Chart' and a 'Leader Board' listing models.

Using OML Notebooks

The screenshot displays the Oracle Machine Learning Notebook interface. At the top, there's a toolbar with several icons: Run All, Show/Hide Code, Clear Output, Export Notebook, and Search Code. Below the toolbar is a main workspace containing a notebook titled "Lab 1: Getting Started with OML4Py". The notebook content is titled "Oracle Machine Learning for Python (OML4Py)" and describes the OML4Py package. It mentions that OML4Py integrates Python with Oracle Autonomous Database, allowing users to run Python commands and scripts for statistical, machine learning, and visualization analyses on database tables and views using Python syntax. The workspace also shows a "Connected Users" section with one user connected, and a "List Shortcuts" section with various keyboard shortcuts. A large green callout highlights the "Run All" button in the toolbar. Another green callout highlights the "Connected Users" and "List Shortcuts" sections. A red arrow points from the "Run All" button in the toolbar down to the "Run All" button in the notebook toolbar.

Oracle Machine Learning

Lab 1: Getting Started with OML4Py

Oracle Machine Learning for Python (OML4Py)

OML Project [OML Workspace]

OMLUSER03

Connected

FINISHED

Width 12

Font size 9

Move down Ctrl+Alt+J

Insert new Ctrl+Alt+B

Run all below Ctrl+Shift+Enter

Clone paragraph Ctrl+Shift+C

Show title Ctrl+Alt+T

Show line numbers Ctrl+Alt+M

Disable run Ctrl+Alt+R

Clear output Ctrl+Alt+L

Remove Ctrl+Alt+D

Show Hide Output

Clear Notebook

Run All

Show/Hide Code

Clear Output

Export Notebook

Search Code

List Shortcuts

Connected Users

Interpreter Bindings

Paragraph Status

Run

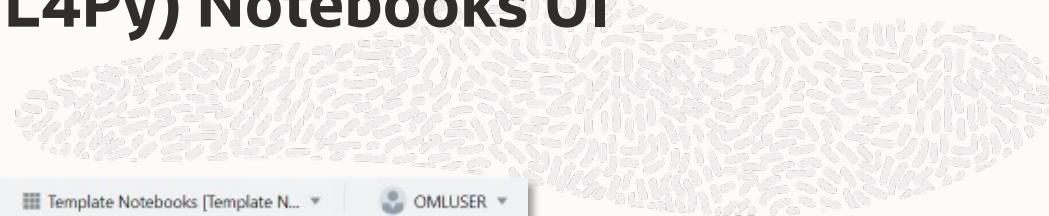
Hide

More Features

Paragraph Output

Oracle Machine Learning for Python (OML4Py) Notebooks UI

Notebook with model created in Python



ORACLE® Machine Learning

Template Notebooks [Template N... OMLUSER Connected

Oracle Machine Learning for Python

Overloaded data visualization functions

OML4Py overloads select graphics functions.

boxplot

Here, we use the overloaded `boxplot` function to show the distribution of the `IRIS` table 'length' and 'width' columns. The statistical computations take place in-database - avoiding data movement. The function returns only the summary statistics needed to produce the plot, which enables scalability. No overhead is incurred for moving the data to the client for processing, whether 150 rows (as in IRIS) or 150 million rows.

histogram

Similarly, we use the overloaded `hist` function, where the statistics are computed in-database. Only the summary statistics needed for the plot are returned to the client.

Took 0 sec. Last updated by MARK at August 13 2020, 12:19:30 PM. (outdated)

FINISHED % X #

Python

```
import matplotlib.pyplot as plt
plt.style.use('seaborn')
plt.figure(figsize=[18,5])

oml.graphics.boxplot(IRIS[:, :4], notch=True, showmeans = True,
                     labels=IRIS.columns[:4])
plt.title('Distribution of IRIS Attributes')
plt.xlabel('cm')

Text(72.625, 0.5, 'cm')
```

Distribution of IRIS Attributes

Took 7 sec. Last updated by MARK at August 13 2020, 12:19:38 PM. (outdated)

FINISHED % X #

python

```
oml.graphics.hist(IRIS['SEPAL_LENGTH'], bins=10, color='red',
                  linestyle='solid', edgecolor='black')

plt.title('Sepal Length variation in IRIS data set')
plt.xlabel('Sepal Length')
plt.ylabel('# of iris instances')

plt.show()
```

Sepal Length variation in IRIS data set

Took 2 sec. Last updated by MARK at August 13 2020, 12:19:41 PM. (outdated)

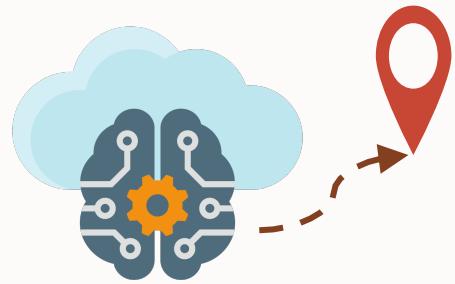
FINISHED % X #

Create derived variables

Python

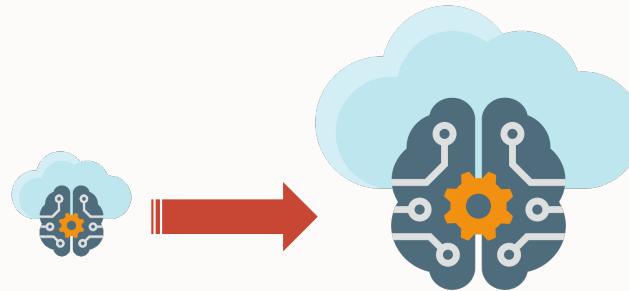
```
is_large_petal = (IRIS['PETAL_LENGTH'] > 5.0) & (IRIS['PETAL_WIDTH'] > 2.0)
```

Oracle Machine Learning Key Attributes



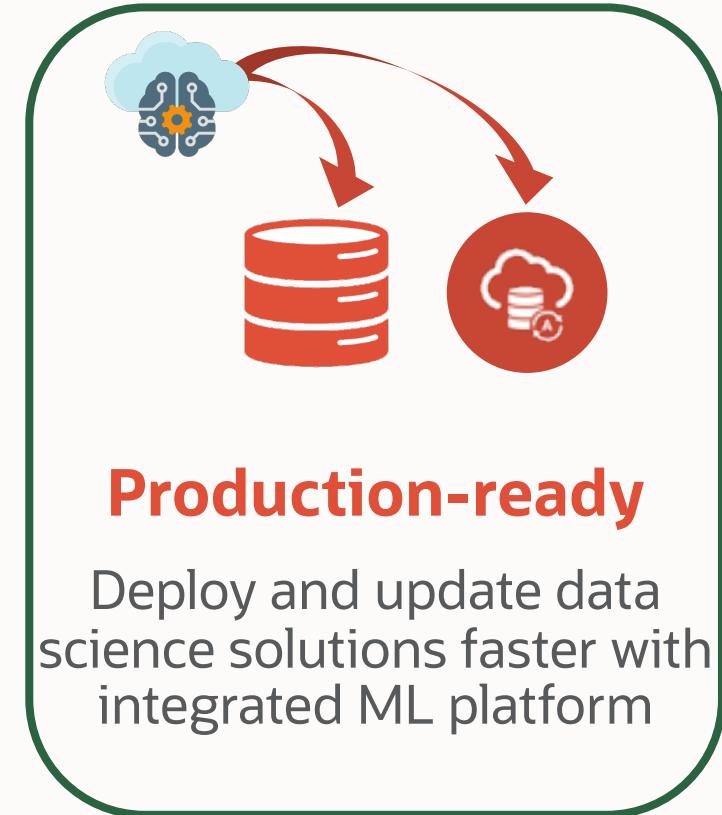
Automated

Get better results faster
with less effort –
even non-expert users



Scalable

Handle big data volumes using
parallel, distributed algorithms –
no data movement



Production-ready

Deploy and update data
science solutions faster with
integrated ML platform

Increase productivity, Achieve enterprise goals, Innovate more

OML Services

REST API – deploy models outside the database

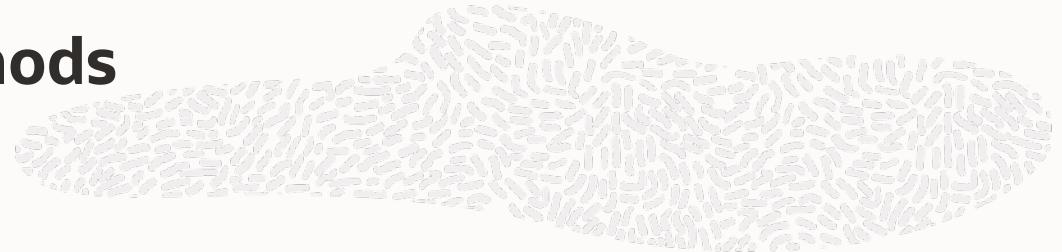


The REST API for Oracle Machine Learning Services on Oracle Autonomous Database provides:

- Endpoints that enable the **storing machine learning models** along with its metadata
- Creation of **scoring endpoints** for the registered models.
- Support for **classification and regression of third-party ONNX models**, including from packages like Scikit-learn and TensorFlow, among several others.
- **Proprietary cognitive text** capabilities in English, French and Spanish for topic discovery, keywords, summary, sentiment, and feature extraction, based on a Wikipedia knowledge base using Embeddings.
- **Cognitive image functionality**, supported through the ONNX format third-party model deployment feature, with the ability to score using images or tensors.

Oracle Machine Learning Services - Methods

Components with built-in Oracle Machine Learning



Admin

POST

- Token using ADB user and password

Generic

GET

- Metadata for all Versions: Version 1 Metadata
- Open API Specification

Repository

POST

- Store Model
- Update Model Namespace

GET

- Models list
- Model Info
- Model Metadata
- Model Content

DELETE

- Model

Deployment

POST

- Create Model Endpoint
- Score Model using Endpoint

GET

- Endpoints
- Endpoint Details
- Open API Specification for Endpoint

DELETE

- Endpoint

Cognitive Text

POST

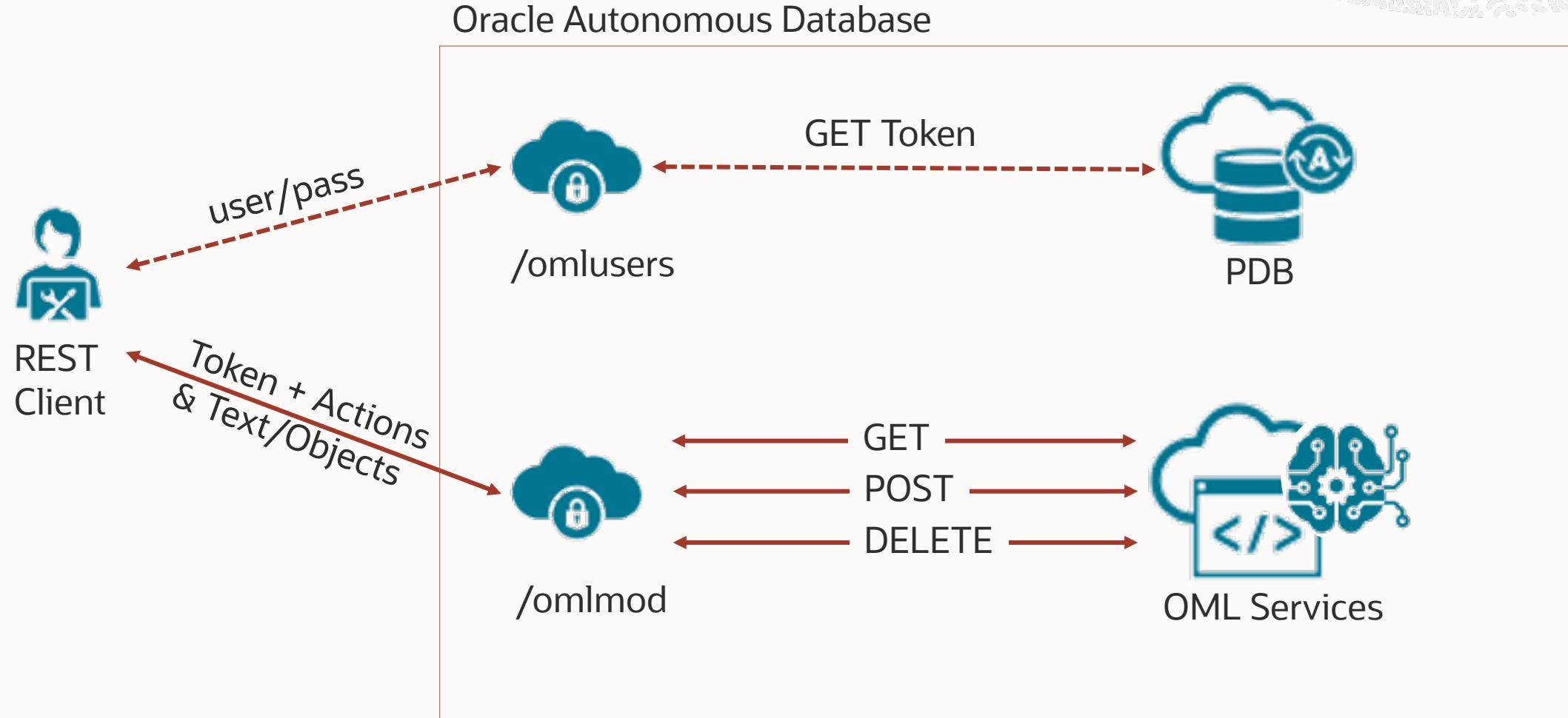
- Get Most Relevant Topics
- Get Most Relevant Keywords
- Get Summaries
- Get Sentiments
- Get Semantic Similarities
- Numeric Features

GET

- Get Endpoints

Oracle Machine Learning Services architecture

Connectivity and use from Client



Score OML Model (Single Class w Pred Details)

Examples 0 | BUILD |  

POST {{omlserver}}/omlmod/v1/deployment/{{omlModelURI}}/score

Send Save

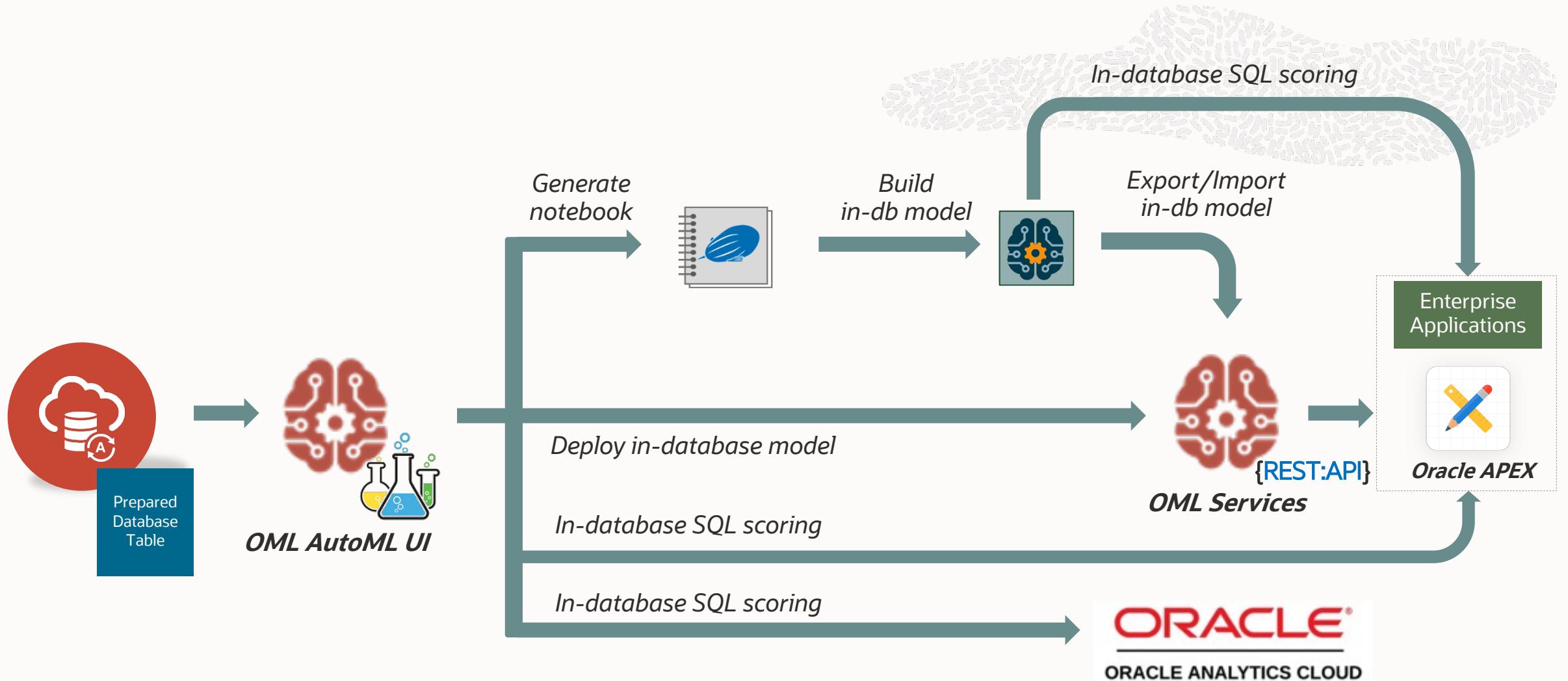
Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2     "topNdetails":5,
3     "inputRecords": [
4         {
5             "AGE":41,
6             "BOOKKEEPING_APPLICATION": 1,
7             "CUST_GENDER":"M",
8             "CUST_MARITAL_STATUS":"NeverM",
9             "EDUCATION":"HS-grad",
10            "HOME_THEATER_PACKAGE":1,
11            "HOUSEHOLD_SIZE":"4",
12            "OCCUPATION":"Crafts",
13            "YRS_RESIDENCE":6,
14            "Y_BOX_GAMES":1
15        }
16    ]
17 }
```

Response

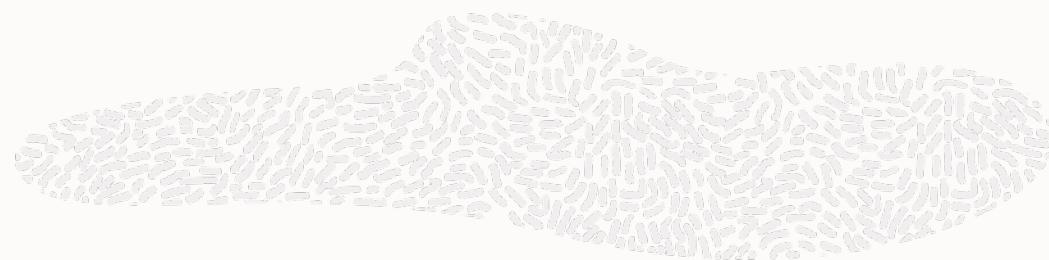
Hit Send to get a response



Model and Evaluate

Deploy

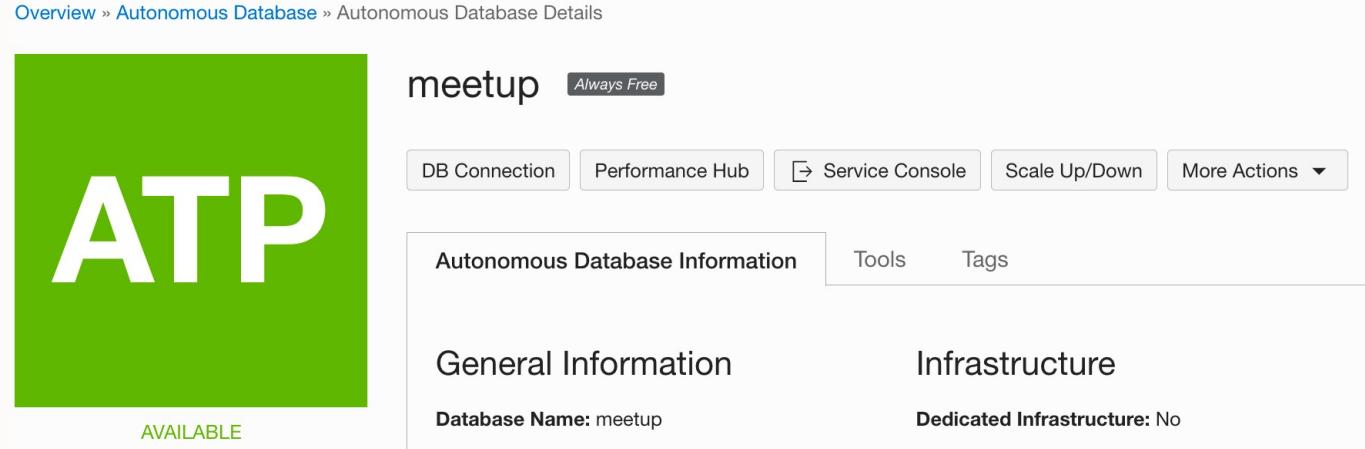
Workshop high-level outline



- **Environment**
 - Create Your Free Trial Account & Sign in to Your Account
 - Provision an Autonomous Database
 - (Use **LiveLabs**)
- Create «OMLUSER» (Already in **LiveLabs**) and Access Oracle Machine Learning Notebooks
- Create and run an **AutoML Experiment**
- Deploy an AutoML UI model to REST API on **OML Services**
- Create an **auto-generated OML Notebook** from the model
- Use Postman to access **OML Services REST APIs to score** the OML AutoML UI model deployments

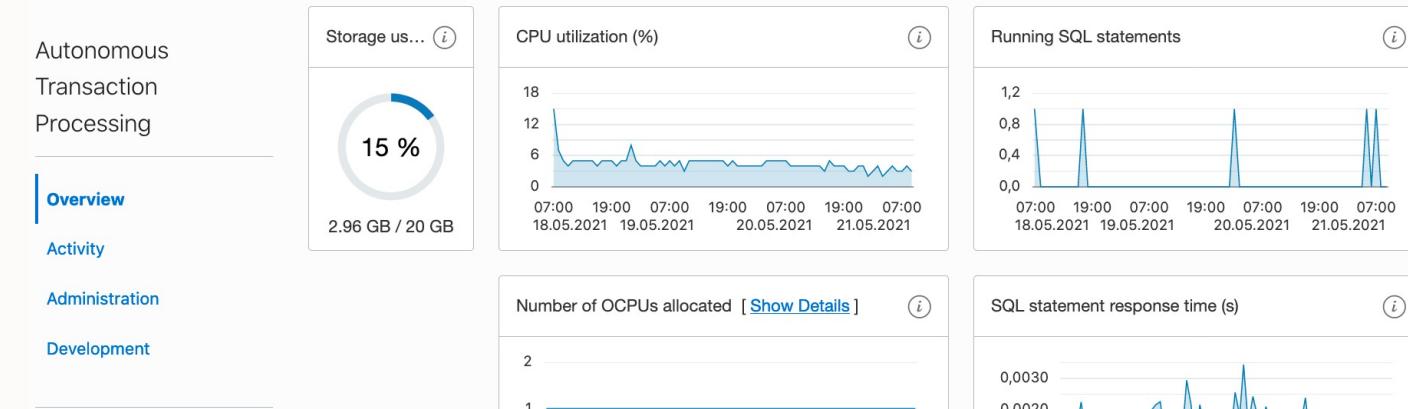
Create an OML User

On the Autonomous Database Details page, click **Service Console**.



The screenshot shows the Oracle Autonomous Database Details page. At the top left is a green square icon with 'ATP' in white. To its right is the database name 'meetup' and a 'Always Free' badge. Below the name are several buttons: 'DB Connection', 'Performance Hub', 'Service Console' (which is highlighted in blue), 'Scale Up/Down', and 'More Actions'. Underneath these buttons is a tab bar with 'Autonomous Database Information' selected, followed by 'Tools' and 'Tags'. The main content area is divided into two sections: 'General Information' and 'Infrastructure'. In 'General Information', the 'Database Name' is listed as 'meetup'. In 'Infrastructure', the 'Dedicated Infrastructure' status is shown as 'No'.

On the Service Console click **Administration**.



Create an OML User

Click **Manage Oracle ML Users** to open the Oracle Machine Learning User Administration page.

Autonomous
Transaction Processing

Overview

Activity

Administration

Development

DATABASE
MEETUP

Download Client Credentials (Wallet)

Connections to Autonomous Transaction Processing use a secure connection. Your existing tools and applications will need to use this wallet file to connect to your Autonomous Transaction Processing instance. If you are familiar with using an Oracle Database within your own data center, you may not have previously used these secure connections.

Set Resource Management Rules

Set resource management rules to allocate CPU/IO shares to consumer groups and to cancel SQL statements based on their runtime and amount of IO.

Set Administrator Password

Set or reset your database administrator user's (ADMIN) password and when locked unlock your administrator user account on Autonomous Transaction Processing.

Manage Oracle ML Users

Create new Oracle Machine Learning user accounts and manage the credentials for existing Oracle Machine Learning users.

Database name:
MEETUP

Sign in with your Oracle Machine Learning Database Administrator credentials

USERNAME

PASSWORD

Sign In

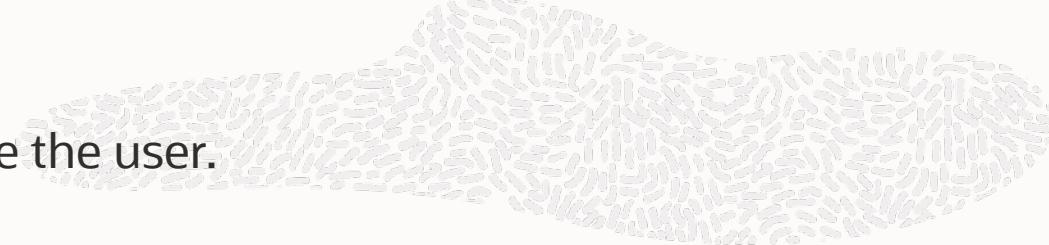
Click **Create** on the Oracle Machine Learning User Administration page.

Users

+ Create		X Delete	<input type="checkbox"/> Show All Users	Search...	
User Name	▲	Full Name	Role	Email	Created On
ADMIN			System Administrator		1/28/21 5:30 PM

Create an OML User

In the Create User page, enter the following details to create the user.



Create User

Username: OMLUSER

First Name:

Last Name:

Email Address:

Generate password and email account details to user. User will be required to reset the password on first sign in.

Password:

Confirm Password:

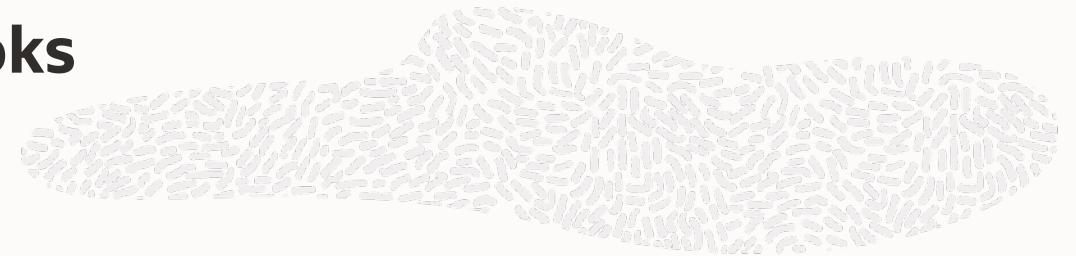
Create **Cancel**

The form shows fields for Username, First Name, Last Name, Email Address, Password, and Confirm Password. A checkbox option is available to generate password and email account details. Buttons for Create and Cancel are at the top right.

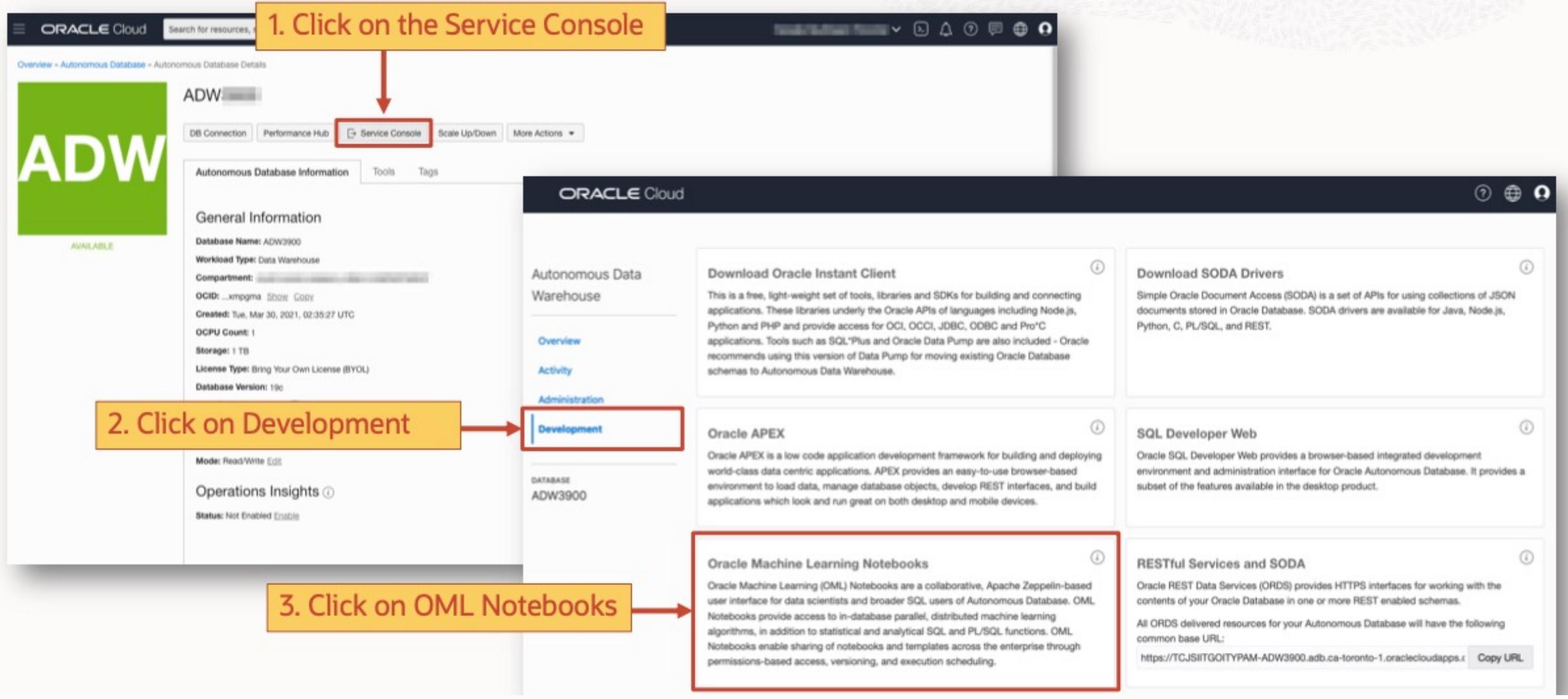
- **Username:** Enter «OMLUSER» for username.
- **First Name:** Enter the first name of the user.
- **Last Name:** Enter the last name of the user.
- **Email Address:** Enter email ID.
- **Password:** Enter a password for the user.
- **Confirm Password:** Enter a password to confirm the value that you entered in the Password field.

Click **Create**.

Access Oracle Machine Learning Notebooks



**Inside the Autonomous Database page, click on "Service Console".
In there, select "Development" and then click on "Oracle Machine Learning Notebooks"**



In the Sign-In screen of OML Notebooks, use «OMLUSER» and your «Password». Once you are in, click on the OML Notebooks link

Sign in with "OMLUSER" and the "Password"

Sign in with your Oracle Machine Learning Database User credentials

USERNAME
OMLUSER
PASSWORD
Sign In

ORACLE Machine Learning

OMLUSER Project [OMLUSER Works...]

Recent Notebooks

Nothing to Display

How Do I?

- Use AutoML
- Get Started
- Create Notebooks
- Create Jobs
- Manage Permissions
- Try It

Quick Actions

- AutoML
- Scratchpad
- Notebooks
- Jobs
- Examples

Recent Activities

No items to display.

Click on AutoML

In the AutoML Experiments screen, click "Create" to create a new one

The screenshot shows the Oracle Machine Learning interface. At the top, there's a navigation bar with the Oracle logo, the text 'OML Project [OML Workspace]', and a user dropdown for 'OMLUSER'. Below the header, the title 'AutoML Experiments' is displayed. Underneath the title is a toolbar with several buttons: '+ Create' (highlighted with a red box and arrow), 'Edit', 'Delete', 'Duplicate', 'Start', 'Stop', and search functions ('Search...' and a magnifying glass icon). A message 'No data to display.' is shown below the toolbar. At the bottom left, there's a page navigation bar indicating 'Page 1 (0 of 0 items)'.

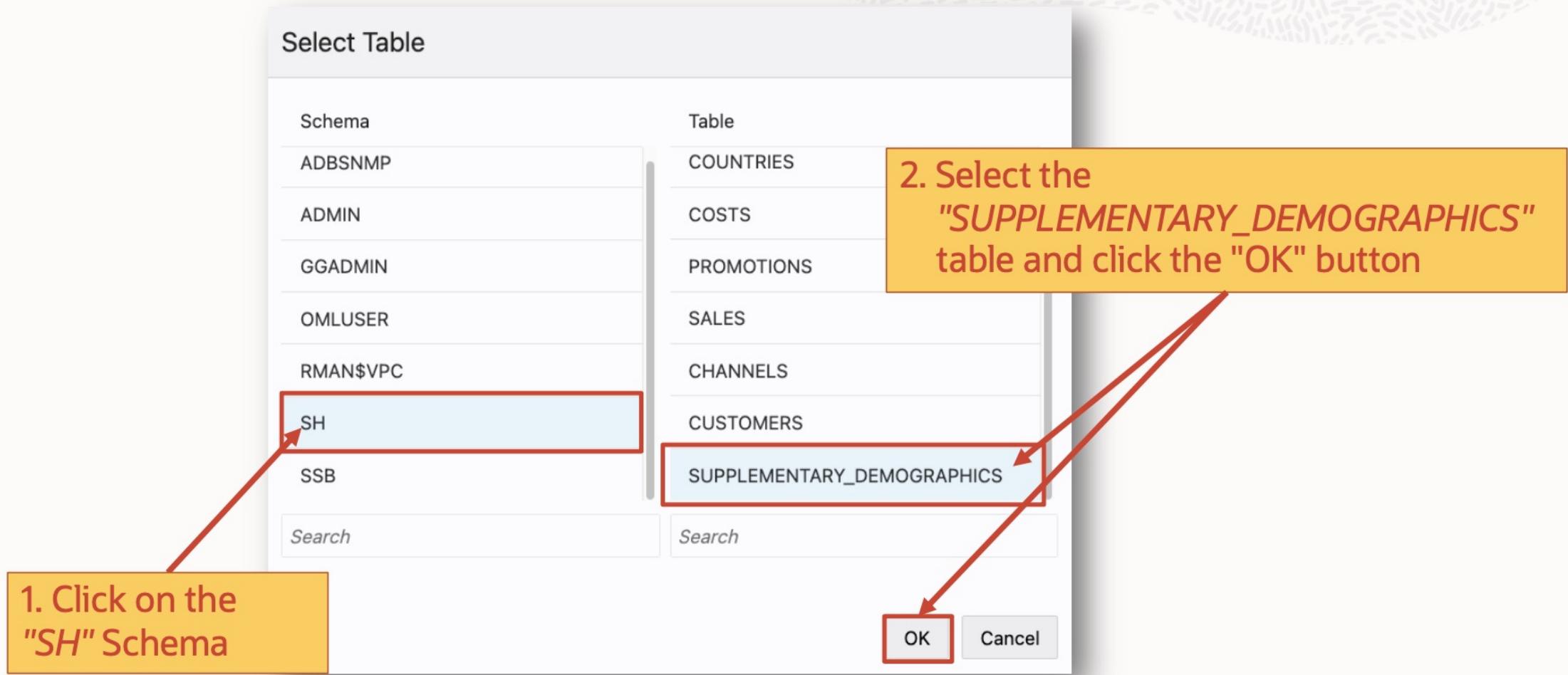
1. Click on +Create

2. Give the Experiment a name
3. Click on the Loupe to search for a Data Source

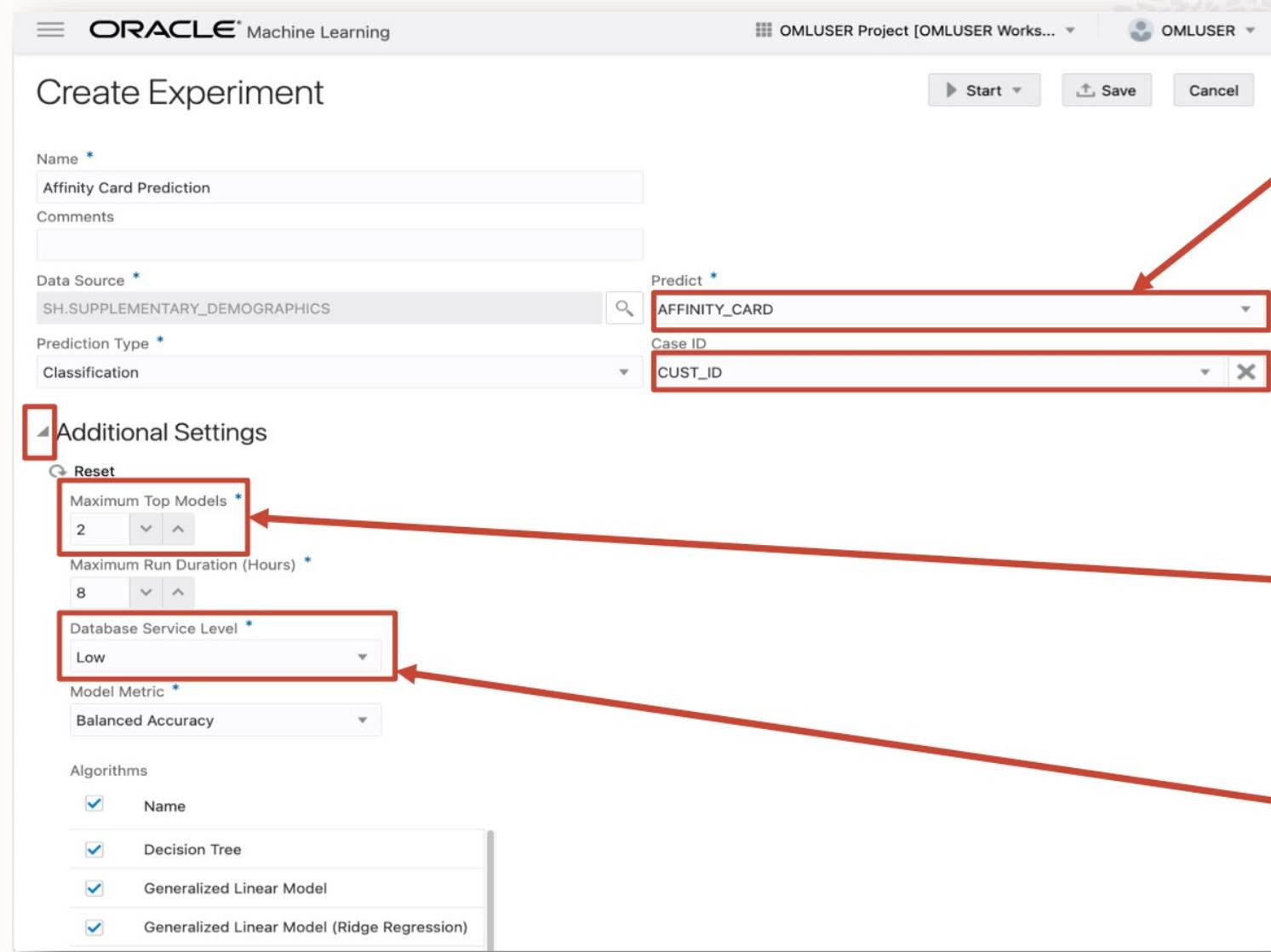
A large callout box contains steps 1, 2, and 3. Step 1 points to the '+ Create' button. Step 2 points to the 'Name' field where 'Affinity Card Prediction' is entered. Step 3 points to the search icon (a magnifying glass) in the 'Data Source' field.

The 'Create Experiment' dialog box is open on the right side of the interface. It contains fields for 'Name *' (set to 'Affinity Card Prediction'), 'Comments' (empty), 'Data Source *' (empty with a search icon highlighted with a red box and arrow), 'Prediction Type *' (set to 'Classification'), and a dropdown menu.

In the menu that appears, select the SH.SUPPLEMENTARY_DEMOGRAPHICS



In the Create Experiments screen, select Predict, Case ID, and adjust performance settings



1. For the "Predict" attribute, select AFFINITY_CARD from the pull-down menu

2. For the "Case ID" attribute, select CUST_ID from the pull-down menu

3. Expand the "Additional Settings" and reduce the Maximum Top Models to "2" to save time in this exercise.

4. Change the "Database Service Level" to Medium to get increased parallelism

We are ready to start the Experiment

Click on the Start -> Faster Results

The screenshot shows the Oracle Machine Learning interface for creating an experiment. The top navigation bar includes the Oracle logo, the project name "OMLUSER Project [OMLUSER Works...]", and the user "OMLUSER". The main title is "Create Experiment".

Form fields include:

- Name *: Affinity Card Prediction
- Comments: (empty)
- Data Source *: SH.SUPPLEMENTARY_DEMOGRAPHICS
- Predict *: AFFINITY_CARD
- Prediction Type *: Classification
- Case ID: CUST_ID

Additional Settings section:

- Reset button
- Maximum Top Models *: 2
- Maximum Run Duration (Hours) *: 8

Two callout boxes provide instructions:

1. Click on Start → Start ▾
2. Click on Faster Results to make sure we have a quick result for this Session → Faster Results

Click on the three dots (...) to see the Progress

The screenshot shows the Oracle Machine Learning interface. On the left, under 'Experiments', there's a card for 'Affinity Card Prediction' with a 'Balanced Accuracy' graph that rises from ~0.50 to ~0.85. Below it is a 'Leader Board' table. A yellow callout box with red text says: '1. Click on the three dots (...) to open the Progress Report'. An arrow points from this box to a red-bordered 'Progress' modal window on the right. The 'Progress' window lists several steps: 'Algorithm Selection' (Completed), 'Adaptive Sampling' (Completed), 'Feature Selection' (Completed), 'Model Tuning' (Running), 'Naive Bayes' (Running), 'Generalized Linear Model' (Queued), and 'Feature Prediction Impact' (Queued). Each step has a circular progress indicator.

ORACLE® Machine Learning

OMLUSER Project [OMLUSER Works...]

OMLUSER

<- Experiments

Affinity Card Prediction

Experiment Settings Edit

Balanced Accuracy

Leader Board

Deploy Create Notebook Metrics

Algorithm	Model Name	Balanced Accuracy
Naive Bayes	nb_47d21ed6fc	0.8868
Generalized Linear Model	glm_c48b041b24	

Features

1. Click on the three dots (...) to open the Progress Report

Running Stop

Progress

- Algorithm Selection Completed
- Adaptive Sampling Completed
- Feature Selection Completed
- Model Tuning Running
- Naive Bayes Running
- Generalized Linear Model Queued
- Feature Prediction Impact Queued

A completed Experiment looks like this

The screenshot shows the Oracle Machine Learning interface with the following elements:

- Header:** ORACLE® Machine Learning, OMLUSER Project [OMLUSER Works...], OMLUSER
- Left Sidebar:** Experiments, Affinity Card Prediction, Experiment Settings, Edit, Balanced Accuracy, Leader Board, Deploy, Create Notebook, Metrics.
- Middle Area:** A yellow callout box contains the text: "The Experiment finishes when you see the \"Completed\" message at the top". An arrow points from this text to the "Completed" status indicator in the top right corner of the main area.
- Run Summary Overlay:** A modal window titled "Run Summary" lists completed steps:
 - Algorithm Selection (Completed)
 - Adaptive Sampling (Completed)
 - Feature Selection (Completed)
 - Model Tuning (Completed)
 - Naive Bayes (Completed)
 - Generalized Linear Model (Completed)
 - Feature Prediction Impact (Completed)A red box highlights the list of completed steps, and another red arrow points from the "Completed" text in the sidebar to this red box.
- Right Sidebar:** A yellow callout box contains the text: "Checkboxes are shown for all completed steps".

Let's check the global Feature Importance

Features						
Name	Importance	Type	Min	Max	Mean	Std Dev
COMMENTS	0.05	VARCHAR2	4.7730	43	1.0	3.770
HOUSEHOLD_SIZE	0.05	VARCHAR2	0	6	0.5	2.236
YRS_RESIDENCE	0.05	NUMBER	0	15	0	14.0
OCCUPATION	0.05	VARCHAR2	0	15	0.5	3.770
EDUCATION	0.05	VARCHAR2	0	16	0.5	3.770
Y_BOX_GAMES	0.05	NUMBER	0	2	0	1
FLAT_PANEL_MONITOR	0.05	NUMBER	0	2	0	1
BULK_PACK_DISKETTES	0.05	NUMBER	0	2	0	1
HOME_THEATER_PACKAGE	0.05	NUMBER	0	2	0	1
BOOKKEEPING_APPLICATION	0.05	NUMBER	0	2	0	1
OS_DOC_SET_KANJI	0.05	NUMBER	0	2	0	1
PRINTER_SUPPLIES	0.05	NUMBER	0	1	1	1
AFFINITY_CARD	0.05	NUMBER	0	2	0.24	0.6
CUST_ID	0.05	NUMBER	4500	100001	104500	102250.8

1. Scroll down to reveal the Features list

2. After an experiment run is completed, the Features grid displays an additional column **Importance**.

Feature Importance indicates the overall level of sensitivity of prediction to a particular feature.

The value is always depicted in the range 0 to 1, with values closer to 1 being more important.

When the Experiment finishes, the models can be inspected

Let's see how to deploy the models to OML Services from here.

The screenshot shows the Oracle Machine Learning interface. On the left, there's a sidebar with 'Experiments' and a 'Leader Board'. The 'Leader Board' section displays two rows of data:

Algorithm	Model Name	Metrics
Naive Bayes	nb_47d21ed6fc	Balanced Accuracy 0.8868
Generalized Linear Model	glm_c48b041b24	0.8697

A yellow callout box with red text and a red arrow points to the 'Deploy' button in the 'Leader Board' header. Another yellow callout box with red text and a red arrow points to the 'Run Summary' section on the right, which lists completed steps like 'Algorithm Selection' and 'Naive Bayes'.

1. Click anywhere in the area around the Model to highlight the it (the whole row will become a shade of blue)

2. The Deploy button will be available. Click on it to get to the Deploy Model form.

When the Experiment finishes, the models can be inspected

Let's see how to deploy the models to OML Services from here.

The screenshot shows the Oracle Machine Learning interface. On the left, under 'Affinity Card Prediction', there's a 'Leader Board' section with two rows:

Algorithm	Model Name	Balanced Accuracy
Naïve Bayes	nb_47d21ed6fc	0.8868
Generalized Linear Model	glm_c48b041b24	0.8697

A red arrow points from the 'Deploy' button in the 'Leader Board' to a callout box containing step 1. Another red arrow points from the blue model name 'nb_47d21ed6fc' to a callout box containing step 2. A third red arrow points from the blue model name to a callout box containing a note about feature impacts.

1. Click anywhere in the area around the Model to highlight the it (the whole row will become a shade of blue)

2. The Deploy button will be available. Click on it to get to the Deploy Model form.

Click on the Naïve Bayes model name in blue

A new window with the Model Details show the Prediction Impacts of the Attributes.

It uses OML's Machine Learning Explainability module to provide model-agnostic functionality to identify the important features that impact a trained model's predictions.

The Confusion Matrix shows an evaluation of the Model on the Validation Data selected by AutoML at the end of the Process

Run Summary details:

- Algorithm Selection: Completed
- Adaptive Sampling: Completed
- Feature Selection: Completed

Prediction Impacts table:

Actual:	0	1
Predicted: 0	598	63
Predicted: 1	32	212

Confusion Matrix table:

Name	Prediction Impact
COMMENTS	High Impact
HOUSEHOLD_SIZE	Medium Impact
OCCUPATION	Medium Impact
YRS_RESIDENCE	Low Impact
Y_BOX_GAMES	Medium Impact
HOME_THEATER_PACKAGE	Low Impact
EDUCATION	Low Impact
BOOKKEEPING_APPLICATION	Low Impact
BULK_PACK_DISKETTES	Low Impact

In the Deploy Model form, fill in the form with the fields like the one below.
The model will be saved and then immediately deployed.

Deploy Model

Name *
AutoML_Affinity_Pred

URI *
automl_affinity_pred

Version *
1.0

Namespace
OML_MODELS

Shared

OK Cancel

The selected model has been deployed successfully.

<- Experiments

Affinity Card Prediction

▶ Experiment Settings 

Balanced Accuracy



Leader Board

Algorithm	Model Name	Balanced Accuracy
Naive Bayes	nb_47d21ed6fc	0.8868
Generalized Linear Model	glm_c48b041b24	0.8697

Auto-generation of an OML Notebook with OML4Py

Let's see how to access a Notebook with the entire model building process

The screenshot shows the Oracle Machine Learning interface. On the left, there's a sidebar with 'Experiments' and a 'Leader Board'. Below the Leader Board, there are tabs for 'Deploy', 'Create Notebook' (which is highlighted with a red box), and 'Metrics'. A table lists 'Algorithm', 'Model Name', and 'Metrics'. The first row shows 'Naive Bayes' with 'nb_47d21ed6fc' and 'Balanced Accuracy' of 0.8868. The second row shows 'Generalized Linear Model' with 'glm_c48b041b24' and 'Balanced Accuracy' of 0.8697.

1. Click anywhere in the area around the Model to highlight the it (the whole row will become a shade of blue)

2. The Create Notebook button will become available. Click on it to get to the Create Notebook form.

Algorithm	Model Name	Balanced Accuracy
Naive Bayes	nb_47d21ed6fc	0.8868
Generalized Linear Model	glm_c48b041b24	0.8697



Auto-generation of an OML Notebook with OML4Py

Give the new Notebook a name

The screenshot shows the Oracle Machine Learning interface. In the center, a modal window titled "Create Notebook" is displayed. It contains instructions: "Create a notebook based on selected model and this experiment's settings. Use a generated notebook to further tune your approach using Python." Below this is a "Notebook Name:" field containing "AutoML NaiveB Affinity Pred". At the bottom right of the modal are "OK" and "Cancel" buttons, with "OK" being highlighted with a red border. In the background, the main interface shows a "Balanced Accuracy" graph and a "Leader Board" section. The "Leader Board" has tabs for "Deploy", "Create Notebook", and "Metrics". The "Create Notebook" tab is active. The main workspace shows an algorithm named "Model Name" and a metric named "Balanced Accuracy". At the bottom left, there is a message bar: "Notebook AutoML NaiveB Affinity Pred sucessfully created". Above this message bar, the status is "Completed".

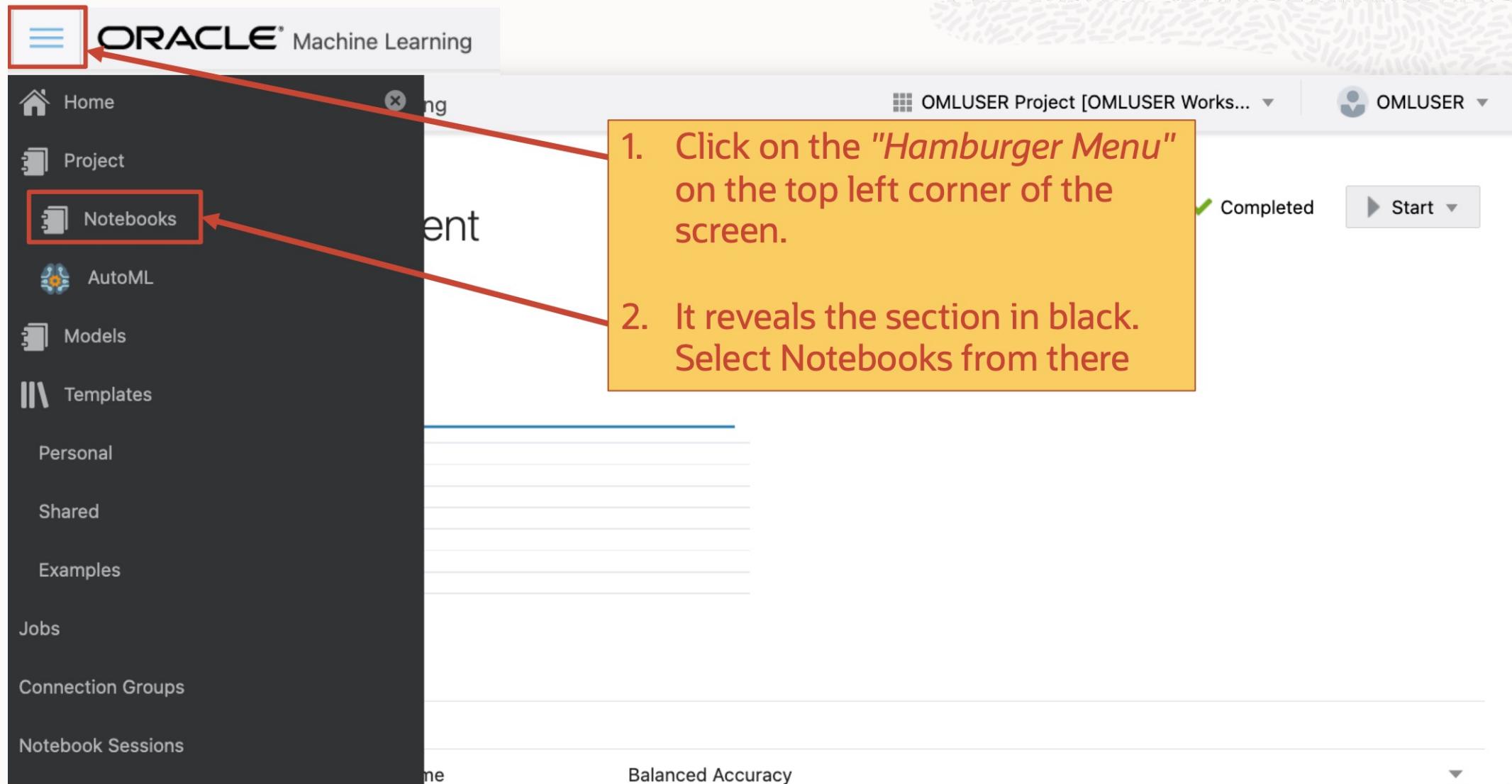
In the new "Create Notebook" window, give it a Short name

Click on OK to Create the Notebook

The message at the Top of the screen confirms that the Notebook was created successfully

Let's check the Notebook code

We need to get to the Notebooks list



The auto-generated Notebook is available

Let's access the Notebook listed

ORACLE® Machine Learning

OMLUSER Project [OMLUSER Works...]

OMLUSER

Notebooks

Name	Comment	Last Update	Updated By	Connection Group
AutoML NaiveB Affinity Pred		5/23/21 9:10 PM	OMLUSER	Global

Page 1 of 1 (1 of 1 items) | K < 1 > K

Click on the Notebook name to open it

The Notebook is opened, but it has not been run yet

Let's run it so we can see all statistics and the resulting output

Click on the "Play" button to Run all paragraphs

AutoML NaiveB Affinity Pred

Oracle Machine Learning AutoML UI - Experiment - Generated Notebook

Get proxy object for selected data

```
%python  
import oml  
  
columns = '"CUST_ID"', '"BOOKKEEPING_APPLICATION"', '"BULK_PACK_DISKETTES"', '"COMMENTS"', '"EDUCATION"', '"FLAT_PANEL_MONITOR"',  
'"HOME_THEATER_PACKAGE"', '"HOUSEHOLD_SIZE"', '"OCCUPATION"', '"OS_DOC_SET_KANJI"', '"PRINTER_SUPPLIES"', '"YRS_RESIDENCE"', '"Y_BOX_GAMES"',  
'"AFFINITY_CARD"'  
schema='SH'  
table='SUPPLEMENTARY_DEMOGRAPHICS'  
  
column = ','.join(columns)  
query = 'SELECT ' + column + ' FROM ' + schema + '.' + table  
  
build_data = oml.sync(query=query)  
z.show(build_data)
```

Check the Notebook output

We can now verify each paragraph

The screenshot shows a Jupyter notebook cell titled "Get proxy object for selected data" containing Python code to build a dataset from a database table. A yellow callout box highlights the purpose of the code.

```
%python  
  
import oml  
  
columns = '"CUST_ID"' , '"BOOKKEEPING_APPLICATION"' , '"BULK_PACK_DISKETTES"' , '"COMMENTS"' , '"EDUCATION"' , '"FLAT_PANEL_MONITOR"' ,  
        '"HOME_THEATER_PACKAGE"' , '"HOUSEHOLD_SIZE"' , '"OCCUPATION"' , '"OS_DOC_SET_KANJI"' , '"PRINTER_SUPPLIES"' , '"YRS_RESIDENCE"' ,  
        '"Y_BOX_GAMES"' , '"AFFINITY_CARD"'  
schema='SH'  
table='SUPPLEMENTARY_DEMOGRAPHICS'  
  
column = ','.join(columns)  
query = 'SELECT ' + column + ' FROM ' + schema + '.' + table  
  
build_data = oml.sync(query=query)  
z.show(build_data)
```

This section builds the Dataset the way the Model needs, only with the required columns

CUST_ID	BOOKKEEPING_APPLICATION	BULK_PACK_DISKETTES	COMMENTS	EDUCATION	FLAT_PANEL_MONITOR
102547	0	1	None	10th	1
101050	0	1	None	10th	1
100040	0	1	None	11th	1
102117	1	0	None	HS-grad	0
101074	0	1	None	10th	1
104179	0	1	None	10th	1
100417	0	0	None	11th	0
101146	1	1	None	< Bach.	1

Took 8 secs. Last updated by OMLUSER at May 23 2021, 8:58:31 PM.

Check the Notebook output

We can now verify each paragraph

The screenshot shows the Oracle Machine Learning interface with two code cells displayed.

Code Cell 1: Prepare training data

```
%python  
import oml  
  
X_train = build_data[:,['CUST_ID', 'BOOKKEEPING_APPLICATION', 'BULK_PACK_DISKETTES', 'COMMENTS', 'EDUCATION', 'FLAT_PANEL_MONITOR',  
    'HOME_THEATER_PACKAGE', 'HOUSEHOLD_SIZE', 'OCCUPATION', 'OS_DOC_SET_KANJI', 'PRINTER_SUPPLIES', 'YRS_RESIDENCE', 'Y_BOX_GAMES']]  
y_train = build_data[:, 'AFFINITY_CARD']
```

Took 0 secs. Last updated by OMLUSER at May 23 2021, 8:58:32 PM.

Text Overlay for Code Cell 1: This section splits the Target column from the Input Attributes, to prepare the Training data for the Model

Code Cell 2: Build 'NAIVE_BAYES' model

```
%python  
import oml  
  
nb_settings = {  
    'ALGO_NAME' : 'ALGO_NAIVE_BAYES' , 'ODMS_SAMPLE_SIZE' : '1292' , 'NABS_SINGLETON_THRESHOLD' : '0.0333333333333333' , 'CLAS_WEIGHTS_BALANCED' :  
    'OFF' , 'NABS_PAIRWISE_THRESHOLD' : '0' , 'ODMS_DETAILS' : 'ODMS_DISABLE' , 'ODMS_SAMPLING' : 'ODMS_SAMPLING_ENABLE'  
}  
  
nb_mod = oml.nb(**nb_settings)  
  
nb_mod = nb_mod.fit(X_train, y_train)
```

Took 1 sec. Last updated by OMLUSER at May 23 2021, 8:58:34 PM.

Text Overlay for Code Cell 2: Using the Algorithm settings identified by AutoML to be the best ones for this model, we proceed to building the model (nb_mod.fit)

Check the Notebook output

We can now verify each paragraph

The screenshot shows the Oracle Machine Learning interface with two main sections: "Show model details" and "Data for scoring".

Show model details:

- %python
- nb_mod
- Algorithm Name: Naive Bayes
- Mining Function: CLASSIFICATION
- Target: AFFINITY_CARD
- Settings:

	setting name	setting value
0	ALGO_NAME	ALGO_NAIVE_BAYES
1	CLAS_WEIGHTS_BALANCED	OFF
2	NABS_PAIRWISE_THRESHOLD	0
3	NABS_SINGLETON_THRESHOLD	0.03333333333333333
4	ODMS_DETAILS	ODMS_DISABLE
5	ODMS_MISSING_VALUE_TREATMENT	ODMS_MISSING_VALUE_AUTO
6	ODMS_SAMPLE_SIZE	1292
7	ODMS_SAMPLING	ODMS_SAMPLING_ENABLE
8	ODPS_AUTO	ON

Took 0 secs. Last updated by OMLUSER at May 23 2021, 8:58:35 PM.

Data for scoring:

- %python
- # using build data for prediction

Took 0 secs. Last updated by OMLUSER at May 23 2021, 8:58:36 PM.

Annotations:

- A yellow callout box with a red arrow points to the "Settings" table in the "Show model details" section. It contains the text: "This section shows the model details. You can scroll down inside the paragraph to see more information about the model Metadata".
- A yellow callout box with a red arrow points to the "# using build data for prediction" line in the "Data for scoring" section. It contains the text: "This is just a notice to mention that the Prediction section will use the same input data for Scoring. You can provide a different Table for it, by using: my_scoring_data = oml.sync(table='MY_SCORING')."

Check the Notebook output

We can now verify each paragraph

This section runs a prediction on the original Build Data. It also selects only the *PREDICTION* column, and brings it into Python's memory for using with Sci-Kit Learn in the next paragraph

Score data

```
%python  
y_pred = nb_mod.predict(build_data)[‘PREDICTION’].pull()
```

Took 0 secs. Last updated by OMLUSER at May 23 2021, 8:58:37 PM.

FINISHED

This paragraph imports the Sci-Kit Learn package and uses the *metrics.balanced_accuracy_score* function to check the predictions quality if one were to use the same input data as Test Data.

Show model quality metric

```
%python  
import sklearn as skl  
metric_score = skl.metrics.balanced_accuracy_score(y_train.pull(), y_pred)  
print(metric_score.round(4))
```

0.8787

Took 0 secs. Last updated by OMLUSER at May 23 2021, 8:58:37 PM.

We expect a value similar to the one we see in the OML AutoML UI Leader Board, but it can be slightly different since that one is not computed over the entire dataset, but on a validation subset (or cross-validation) generated by AutoML

FINISHED

Check the Models menu to view User Models and Deployments

The screenshot displays the Oracle Machine Learning interface with a brain icon watermark. The left sidebar shows navigation options: Home, Project, Notebooks, AutoML, **Models** (highlighted with a red box), Templates, Personal, Shared, Examples, Jobs, Connection Groups, and Notebook Sessions. The main content area has tabs for **User Models** (highlighted with a red box) and **Deployments**. The **User Models** tab shows a table with columns: Name, Owner, Algorithm, Creation Date, and Target. It lists two entries: GLM_C48B041B24 (Owner: OMLUSER, Algorithm: Generalized Linear Model, Creation Date: 5/17/21 2:16 PM, Target: AFFINITY_CARD) and NB_47D21ED6FC (Owner: OMLUSER, Algorithm: Naive Bayes, Creation Date: 5/17/21 2:16 PM, Target: AFFINITY_CARD). The **Deployments** tab shows a table with columns: Name, Shared, Version, Namespace, Owner, Deployed Date, and URI. It lists one entry: AutoML_Affinity_Pred (Shared: checked, Version: 1.0, Namespace: OML_MODELS, Owner: OMLUSER, Deployed Date: 5/17/21 7:17 PM, URI: automl_affinity_pred).

Name	Owner	Algorithm	Creation Date	Target
GLM_C48B041B24	OMLUSER	Generalized Linear Model	5/17/21 2:16 PM	AFFINITY_CARD
NB_47D21ED6FC	OMLUSER	Naive Bayes	5/17/21 2:16 PM	AFFINITY_CARD

Name	Shared	Version	Namespace	Owner	Deployed Date	URI
AutoML_Affinity_Pred	<input checked="" type="checkbox"/>	1.0	OML_MODELS	OMLUSER	5/17/21 7:17 PM	automl_affinity_pred

Add your own code to the Notebook (Optional)

Save and deploy the model from the generated Notebook

```
Build 'NAIVE_BAYES' model
%python

import oml

nb_settings = {
    'ODMS_SAMPLE_SIZE': '1292', 'NABS_SINGLETON_THRESHOLD': '0.0333333333333333', 'CLAS_WEIGHTS_BALANCED': 'OFF', 'NABS_PAIRWISE_THRESHOLD': '0', 'ODMS_DETAILS': 'ODMS_DISABLE', 'ODMS_SAMPLING': 'ODMS_SAMPLING_ENABLE', 'ALGO_NAME': 'ALGO_NAIVE_BAYES'
}

nb_mod = oml.nb(**nb_settings)

nb_mod = nb_mod.fit(X_train, y_train, model_name='AC_NB_Model')

Took 2 secs. Last updated by OMLIOT at May 26 2021, 5:59:28 PM. (outdated)
```

The screenshot shows the Oracle Machine Learning interface. On the left, a sidebar menu includes 'Home', 'Project', 'Notebooks', 'AutoML', 'Models' (which is highlighted with a red arrow), 'Templates', 'Personal', 'Shared', 'Examples', and 'Jobs'. The main area displays a 'Recent Notebooks' section and a 'Models' section. The 'Models' section is divided into 'User Models' and 'Deployments'. Under 'User Models', there is a table:

Name	Owner	Algorithm	Creation Date	Target
AC_NB_MODEL	OMLIOT	Naive Bayes	5/26/21 5:59 PM	AFFINITY_CARD
GLM_2B989915E6	OMLIOT	Generalized Li...	5/23/21 3:26 PM	AFFINITY_CARD
NB_E58D944C13	OMLIOT	Naive Bayes	5/23/21 3:26 PM	AFFINITY_CARD

Under 'Deployments', there is another table:

Name	Shared	Version	Namespace	Owner	Deployed Date	URI
AC_NB_MODEL		1.0		OMLIOT	5/26/21 5:03 PM	AC_NB_Model

Add your own code to the Notebook (Optional)

Create DataFrame with testdata and save DataFrame to database table

Create a DataFrem with testdata

```
%python  
import pandas as pd  
test_DF = pd.DataFrame({"AGE": [39,41], "BOOKKEEPING_APPLICATION": [0,1], "CUST_GENDER": ["F","M"], "CUST_MARITAL_STATUS": ["Married", "NeverM"], "EDUCATION": ["Masters", "HS-grad"], "HOME_THEATER_PACKAGE": [1,1], "HOUSEHOLD_SIZE": ["3", "4"],  
    "OCCUPATION": ["Exec.", "Crafts"], "YRS_RESIDENCE": [6,6], "Y_BOX_GAMES": [1,1]})  
z.show(test_DF.head())
```



AGE	BOOKKEEPING_APPLICATION	CUST_GENDER	CUST_MARITAL_STATUS	EDUCATION	HOME_THEATER_PACKAGE	HOUSEHOLD_SIZE	OCCUPATION	YRS_RESIDENCE	Y_BOX_GAMES
39	0	F	Married	Masters	1	3	Exec.	6	1
41	1	M	NeverM	HS-grad	1	4	Crafts	6	1

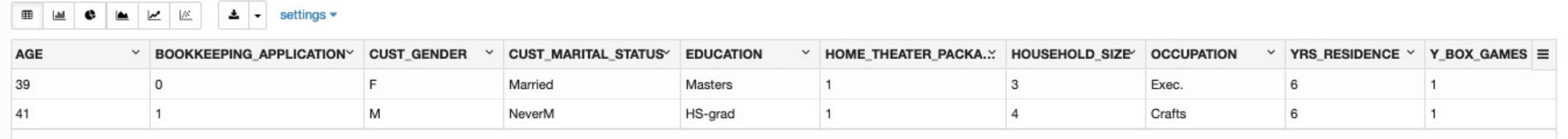
Save DataFrame as database table

```
%python  
oml.create(test_DF, table = 'TESTDATA')  
#oml.drop('TESTDATA')
```

Took 8 secs. Last updated by OMLIOT at May 26 2021, 12:27:47 AM. (outdated)

Use SQL to select from table

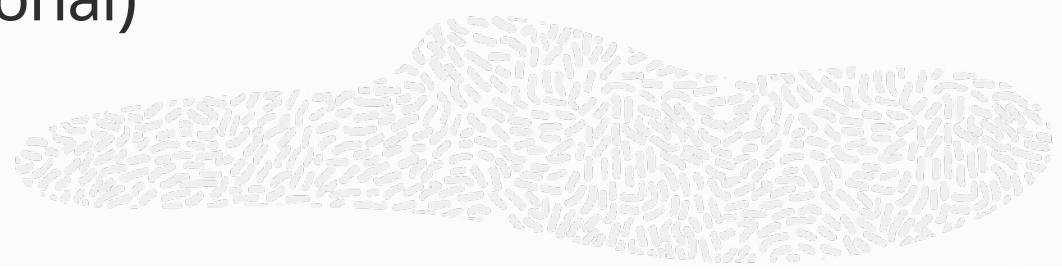
```
%sql  
select * from testdata;
```



AGE	BOOKKEEPING_APPLICATION	CUST_GENDER	CUST_MARITAL_STATUS	EDUCATION	HOME_THEATER_PACKAGE	HOUSEHOLD_SIZE	OCCUPATION	YRS_RESIDENCE	Y_BOX_GAMES
39	0	F	Married	Masters	1	3	Exec.	6	1
41	1	M	NeverM	HS-grad	1	4	Crafts	6	1

Add your own code to the Notebook (Optional)

Use saved model in SQL to predict on testdata.



Use SQL and the model to find probability for AFFINITY_CARD=0

FINISHED ▶ ✎

```
%sql  
select AGE,BOOKKEEPING_APPLICATION,CUST_GENDER,EDUCATION,HOUSEHOLD_SIZE,  
PREDICTION_PROBABILITY(AC_NB_Model, '0' USING * ) as PROBABILITY FROM testdata;
```

grid chart scatter line bar settings ▾

AGE	BOOKKEEPING_APPLICATION	CUST_GENDER	EDUCATION	HOUSEHOLD_SIZE	PROBABILITY
39	0	F	Masters	3	0.755191254972626
41	1	M	HS-grad	4	0.9461652664745478

Use SQL and the model to find probability for AFFINITY_CARD=0 for some attributes

FINISHED ▶ ✎

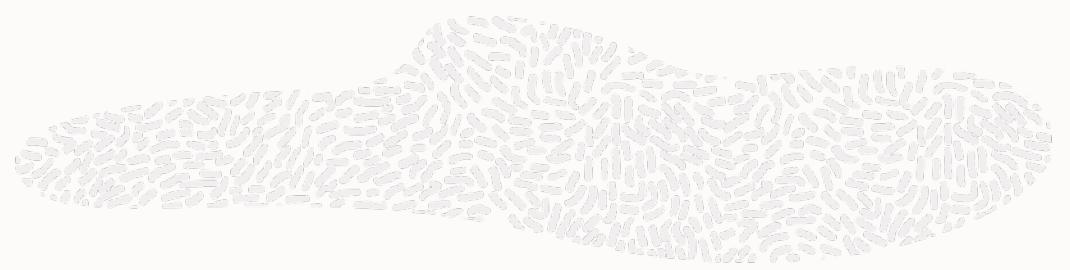
```
%sql  
SELECT PREDICTION_PROBABILITY(AC_NB_Model,'0' USING 25 AS AGE,'F' AS GENDER,'Bach.' AS EDUCATION, 2 AS HOUSEHOLD_SIZE) PROBABILITY_0 FROM DUAL;
```

grid chart scatter line bar settings ▾

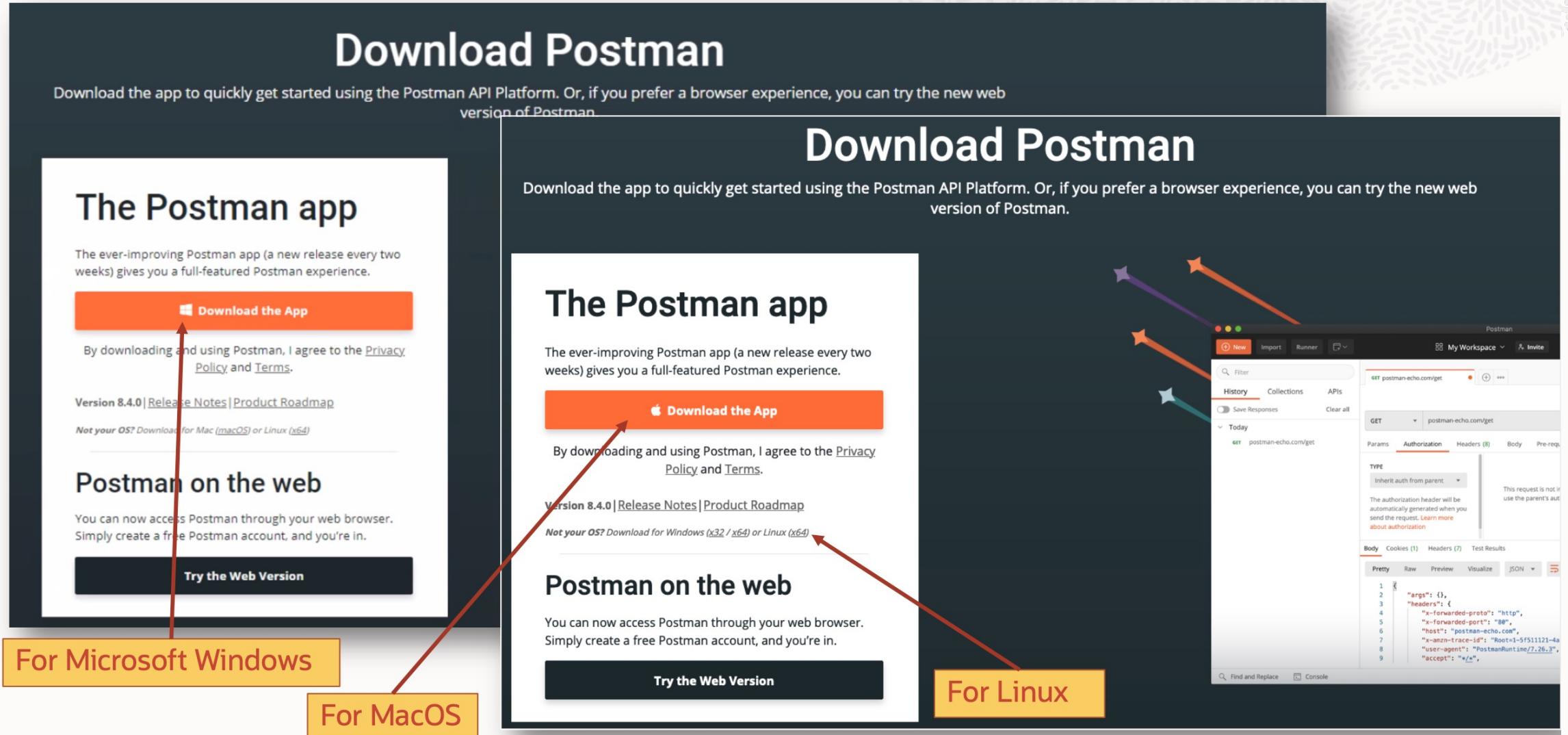
PROBABILITY_0

0.8400373292539522

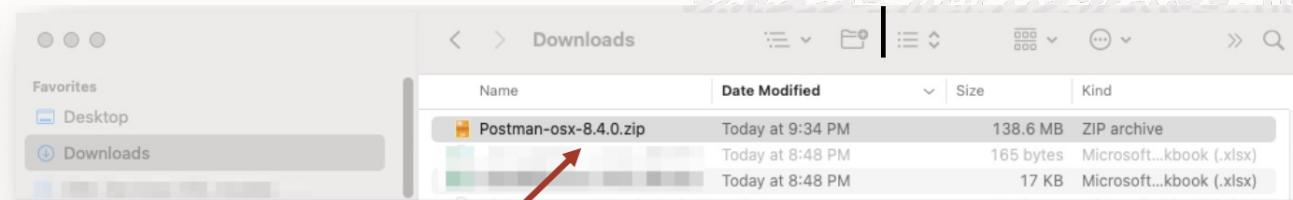
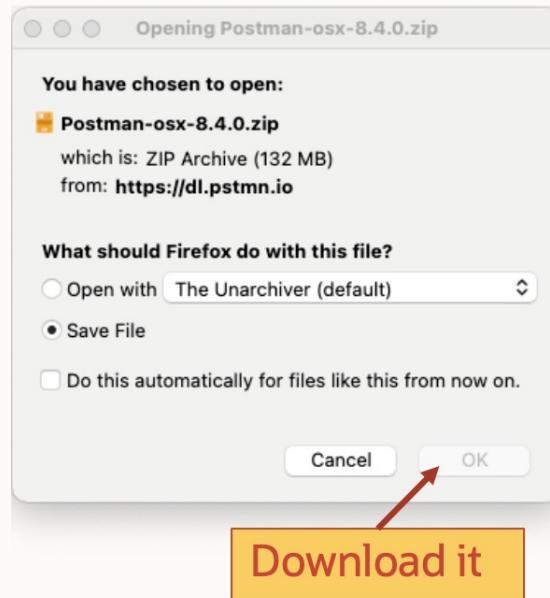
Postman



On your browser, go to <https://www.postman.com/downloads/>, and click on the Orange Button to «Download the App»

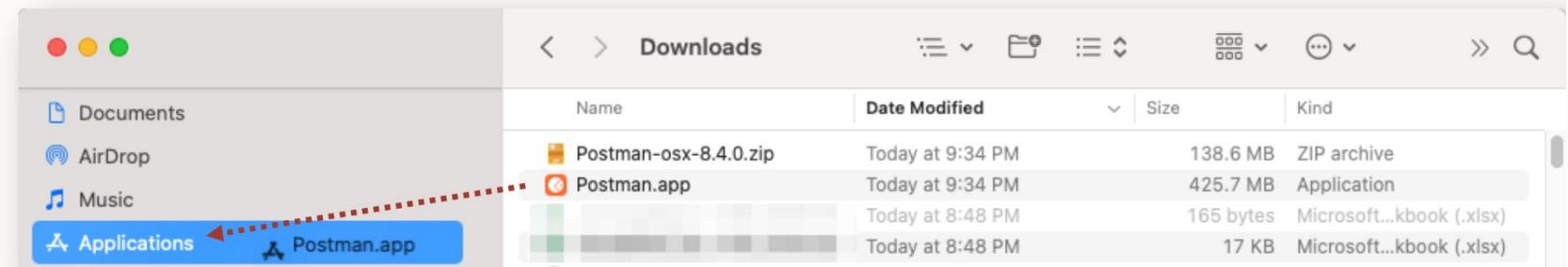


As an example, in the MacOS it will download a ZIP file. Extract the zip file with a double-click, and the Postman.app will be uncompressed



On Mac OS, double click to Extract the App.
On Windows you can run the Installer ".exe"

On Mac OS, you can Drag the "Postman.app" to the Applications folder



Launch Postman on your Desktop

The screenshot shows the Postman desktop application window. At the top left is the Postman logo. Below it, there are two buttons: "Create Free Account" (orange) and "Sign in" (grey). A red arrow points from the "Skip and go to the app" link at the bottom of the window to the "Create Free Account" button. To the right of the window, a large yellow callout box contains the text: "For this Hands-on-Lab, we will NOT need to create an Account with Postman, so you can click on *'Skip and go to the app'* on either the MacOS or Windows version". Below the main window, there is a smaller, semi-transparent view of the same window.

For this Hands-on-Lab, we will NOT need to create an Account with Postman, so you can click on "*Skip and go to the app*" on either the MacOS or Windows version

Create an account or sign in

Create Free Account

Sign in

Create your account or sign in later? [Skip and go to the app](#)

A free Postman account lets you

- Organize all your API development in workspaces
- Create public workspaces to collaborate with over 10 million developers
- Back up your work on Postman's cloud
- Experience the best API development platform for free!

Postman

File Edit View Help

POSTMAN

Create an account or sign in

Create Free Account

Sign in

Create your account or sign in later? [Skip and go to the app](#)

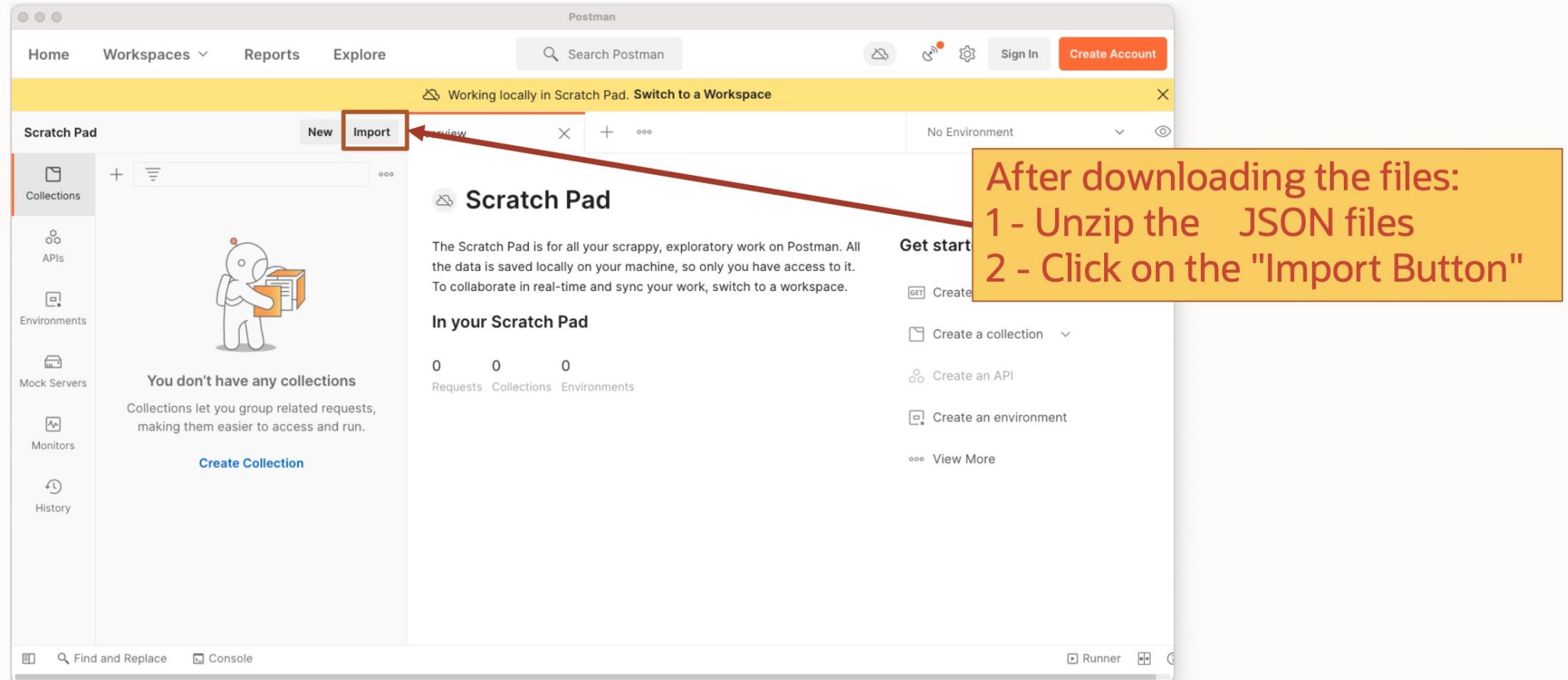
A free Postman account lets you

- Organize all your API development in workspaces
- Create public workspaces to collaborate with over 10 million developers
- Back up your work on Postman's cloud
- Experience the best API development platform for free!

Launch Postman on your Desktop

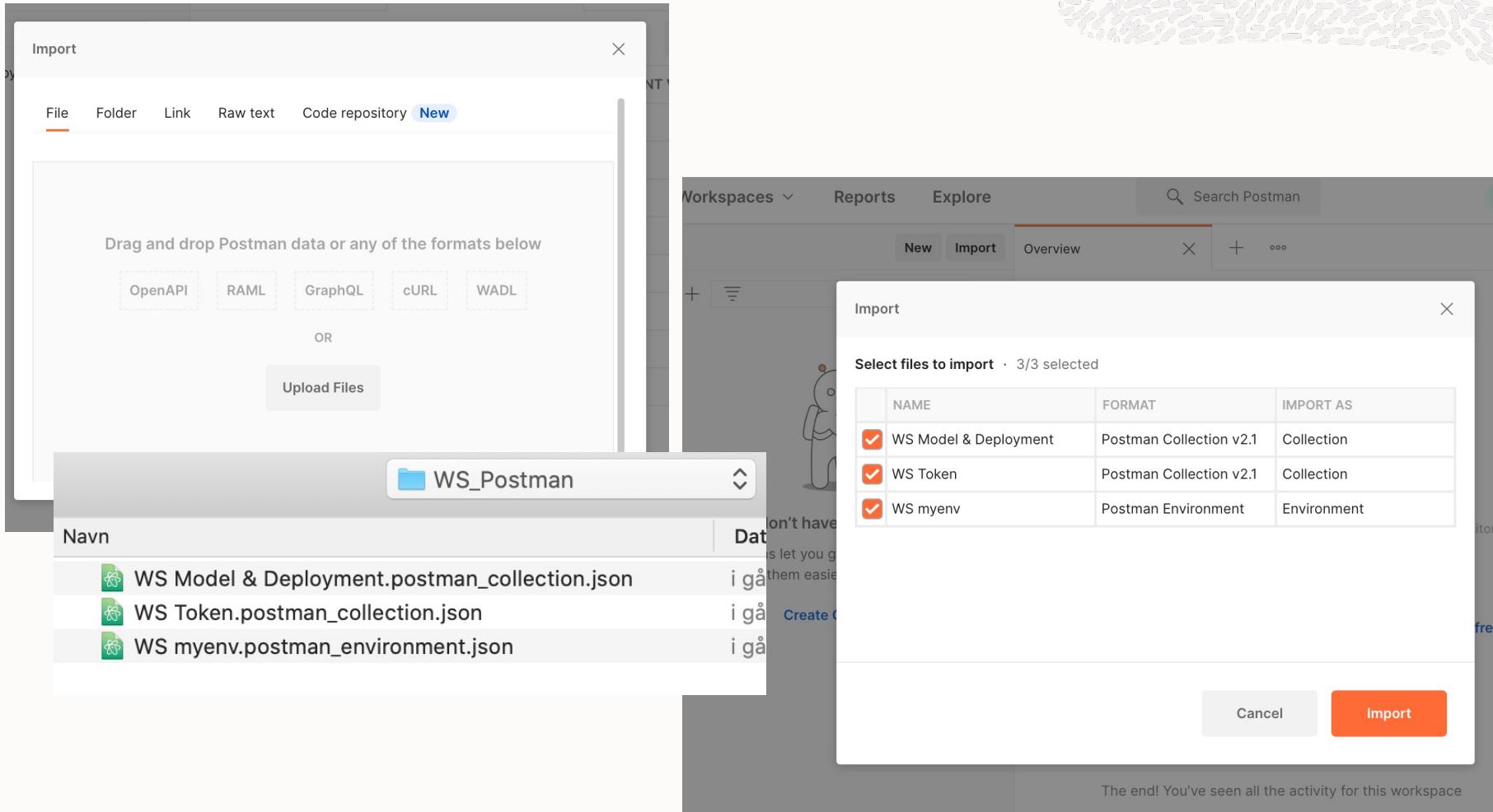
First step is to Import the Collections and Environment that will in file WS_Postman.zip on:

<https://github.com/bios62/meetups/tree/bios62-11-05-2021-arduino-lab>



Import the WS_Postman collections

You can just drag and drop the Postman collection JSON files you unzipped into the Window



Setup the Postman Environment

When back to the main Collection Tab, select your Environment

VARIABLE	INITIAL VALUE	CURRENT VALUE
omlserver		
tenant		
database		
username		
password		
Token		
omlModelURI		

Fill in everything with your environment except *Token*

<input checked="" type="checkbox"/>	omlserver	https://adb.eu-frankfu...	https://adb.eu-frankfurt-1.oraclecloud.com
<input checked="" type="checkbox"/>	tenant	ocid1.tenancy.oc1..aaa...	ocid1.tenancy.oc1..aaaaaaaaafif2uasr2shm5aq
<input checked="" type="checkbox"/>	database	MEETUP	MEETUP
<input checked="" type="checkbox"/>	username	! OMLUSER	OMLUSER

Name	Shared	Version	Namespace	Owner	Deployed Date	URI
AutoML_Affinity_Pred	<input checked="" type="checkbox"/>	1.0	OML_MODELS	OMLUSER	5/17/21 7:17 PM	automl_affinity_pred

where:

• **omlserver** = OML Services server for your Autonomous Database, for example, for Frankfurt: <https://adb.eu-frankfurt-1.oraclecloud.com>

• **tenant** = Oracle Cloud Tenancy OCID (not to be confused with the ADB OCID), in the form of: OCID1.TENANCY.OC1.....

• **database** = Oracle Autonomous Database name, for example: MEETUP

OML Services – Request a Token

Initial call to get a Token and be able to access all other OML Services endpoints

Request a Token

To request a Token for accessing all other OML Services endpoints, you need a valid user and password for your Oracle Autonomous Database that has the proper grants as an OML Developer from the OML Administrator, **usually created by the OMLADMIN**

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** {{omlserver}}/omlusers/tenants/{{tenant}}/databases/{{database}}/api/oauth2/v1/token
- Body (JSON):**

```
1 {
2   "grant_type": "password",
3   "username": "{{username}}",
4   "password": "{{password}}"
5 }
```
- Headers:** (9)
- Params:** none
- Authorization:** (dropdown menu)
- Body:** (selected tab)
- Pre-request Script:** (button)
- Tests:** (button)
- Settings:** (button)
- Cookies:** (button)
- Code:** (button)
- Send:** (button)
- Save:** (button)

Below the interface, there is a placeholder icon of a rocket and an astronaut, with the text "Hit Send to get a response".

GET**POST**

OML Services – List Models and Score model

Call to get the basic Metadata Description for the current OML Services

The screenshot shows the Oracle Model Server interface. On the left, there's a sidebar with icons for Collections, APIs, Environments, Lock Servers, Monitors, and a Help section. The 'WS Model & Deployment' section is expanded, showing various endpoints:

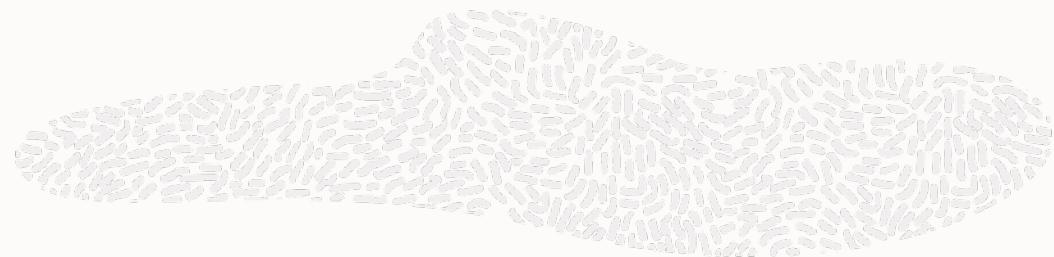
- GET List All Models**
- GET Get Model deployment details**
- GET List All Model deployments** (highlighted in grey)
- POST Score model(Single Classificati...**
- POST Score model (Two Classificatio...**
- POST Score OML Model (Single Class...**
- POST Score OML Model (Two Class w...**

Below this, the 'WS Token' section contains:
POST Get user token

The main panel displays the 'WS Model & Deployment / List All Model deployments' endpoint. It includes a 'Send' button and tabs for Params, Auth, Headers (5), Body, Pre-req., Tests, and Settings. The 'Params' tab is selected, showing a table for Query Params:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Helpful Links



ORACLE MACHINE LEARNING ON O.COM

<https://www.oracle.com/machine-learning>

OML TUTORIALS

OML LiveLab: https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/view-workshop?p180_id=560

OML4Py LiveLab: <https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/view-workshop?wid=786>

Interactive tour: <https://docs.oracle.com/en/cloud/paas/autonomous-database/oml-tour>

Picking a Good Wine Using ADW, OML and OAC Workshop: <https://go.oracle.com/LP=109257?elqCampaignId=293121>

OML OFFICE HOURS

<https://asktom.oracle.com/pls/apex/asktom.search?office=6801#sessions>

ORACLE ANALYTICS CLOUD

<https://www.oracle.com/solutions/business-analytics/data-visualization/examples.html>

OML4PY

[OML4Py](#) (2m video)

[OML4Py Introduction](#) (17m video)

[OML4Py Technical Brief](#)

[OML4Py User's Guide](#)

[Blog: Introducing OML4Py](#)

[GitHub Repository with Python notebooks](#)

ORACLE AUTOML UI

[Oracle Machine Learning AutoML UI](#) (2m video)

[Oracle Machine Learning Demonstration](#) (6m video)

[OML AutoML UI Technical Brief](#)

[Blog: Introducing Oracle Machine Learning AutoML UI](#)

OML SERVICES

[Oracle Machine Learning Services](#) (2m video)

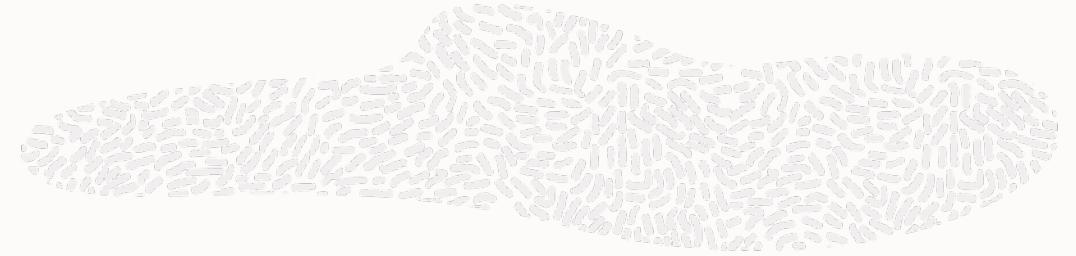
[OML Services Technical Brief](#)

[Oracle Machine Learning Services Documentation](#)

[Blog: Introducing Oracle Machine Learning Services](#)

[GitHub Repository with OML Services examples](#)





Live Webinars

Oracle Operations and Machine Learning (27.5 at 5.00PM CET): <https://bit.ly/3bTc7dY>