Northumbria University
NEWCASTLE

Department of Computer and Information Sciences

KV4004 – AI Fundamentals

Workshop 1- Exercise 2

October 2024

Explore Automated Machine Learning in Azure ML

The purpose of this workshop is to familiarize you with fundamental concepts of automated machine learning and how to train and test models using Azure Machine Learning.

Prerequisites:

- An Azure student subscription.
- Azure Machine Learning workspace and computing instance

A If you have not registered for an Azure student subscription and created an Azure Machine Learning workspace, please refer to the Week 1 workshop document for instructions.

Exercise - Part 1: Getting Started with Automated Machine Learning

In this part of the exercise, you will set up and run experiments using the Azure Machine Learning studio. Therefore:

1. Enable preview features:
   - Some features of Azure Machine Learning are in preview, and need to be explicitly enabled in your workspace. In Azure Machine Learning Studio, click on **manage preview features** (the loud speaker icon) and enable the following preview feature:
     – *Guided experience for submitting training jobs with serverless compute*

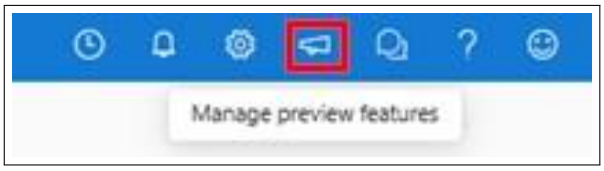


Fig. 2.1: Enabling preview features

2. In the left pane (under Authoring), select **Automated ML** and then click **+ New automated ML job**

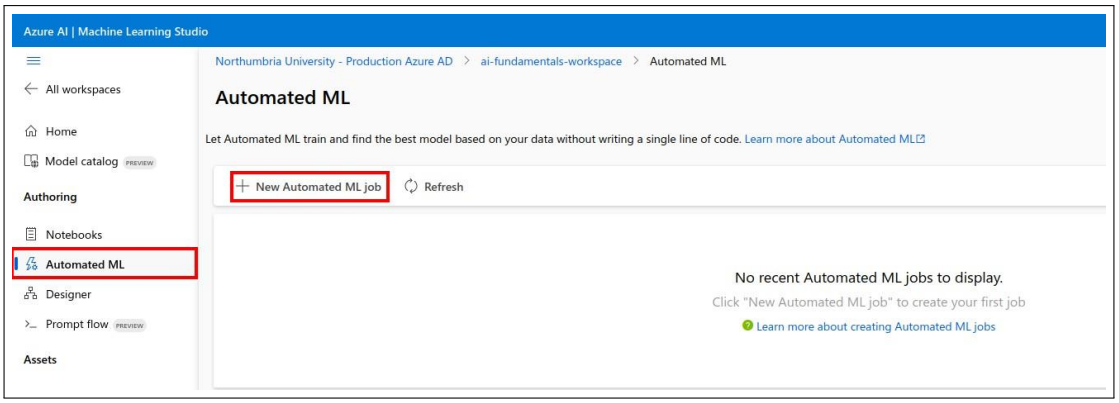
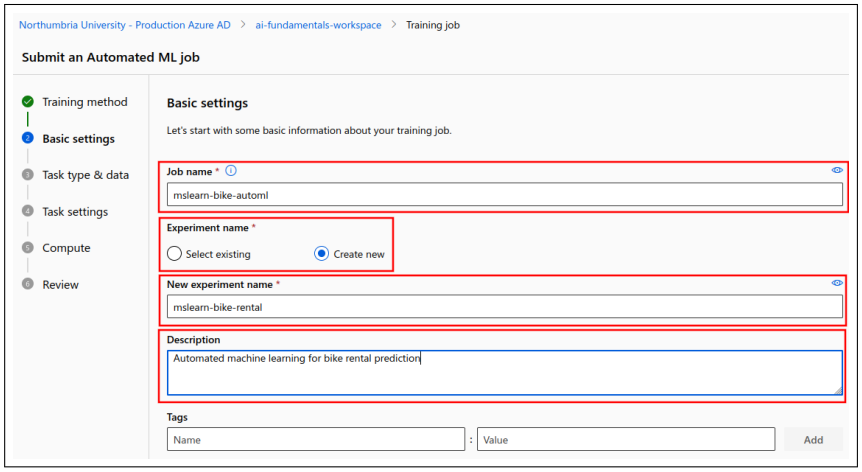


Fig. 2.2: Creating a new automated ML job.

[1] https://ml.azure.com

3. Create the automated ML job with the following settings (use **Next** as required to progress through the user interface):

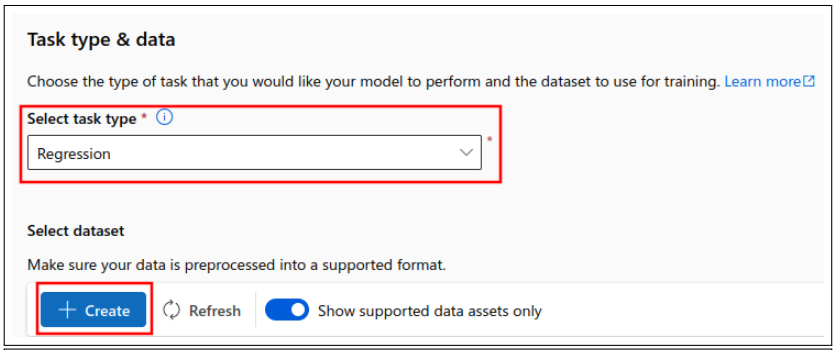


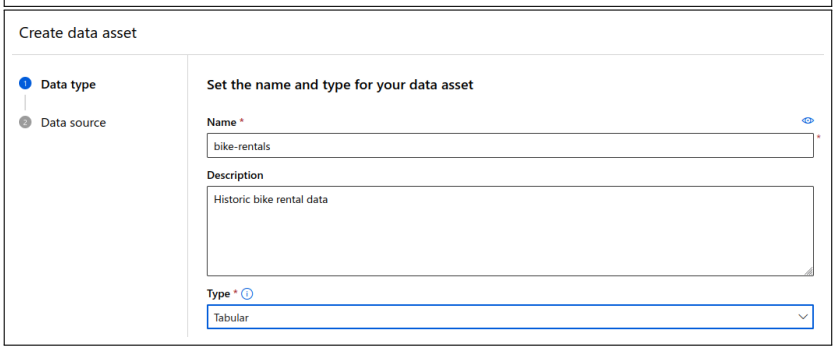
Basic settings:

- **Job name:** mslearn-bike-automl
- **New experiment name:** mslearn-bike-rental
- **Description:** Automated machine learning for bike rental prediction
- **Tags:** none

Fig. 2.3: Configuring a new automated ML job.

4. Create a new dataset with the following settings:

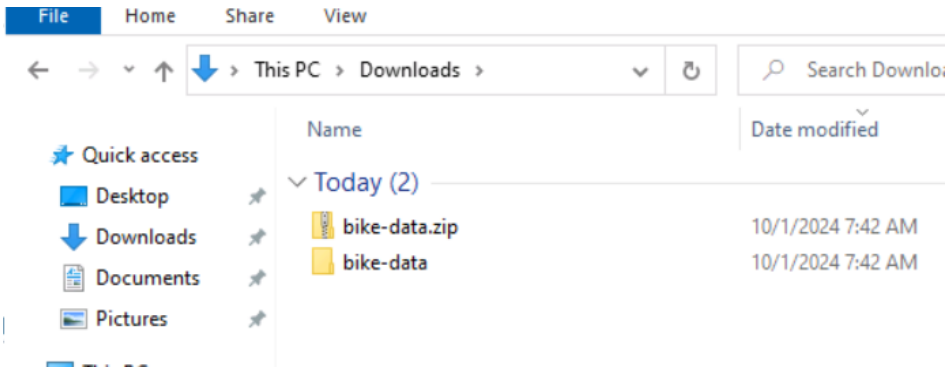




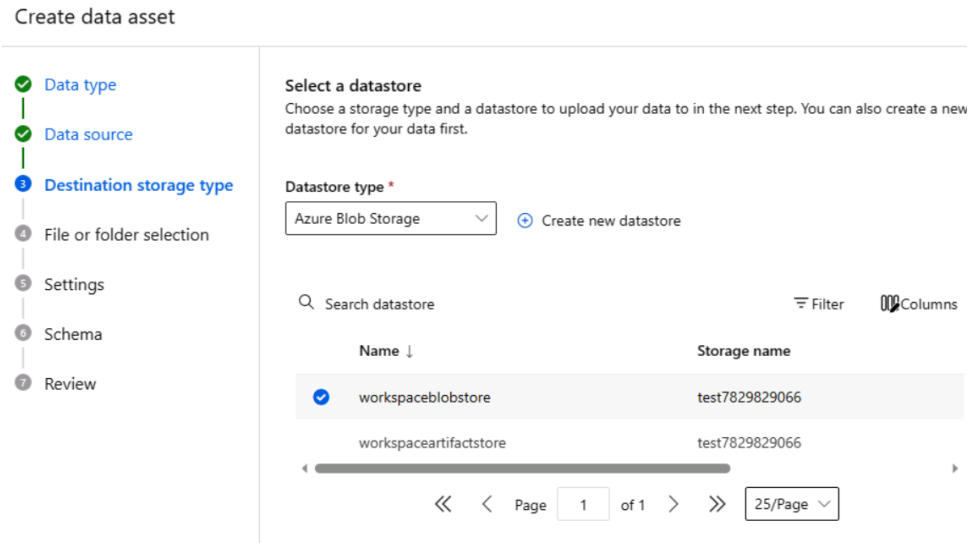
- **Select task type:** Regression
- **Select dataset:** Create:
  - **Description:** Historic bike rental data
  - **Data type:** Name: bike-rentals
  - **Type:** Tabular

Fig. 2.5: Configuring a new dataset.

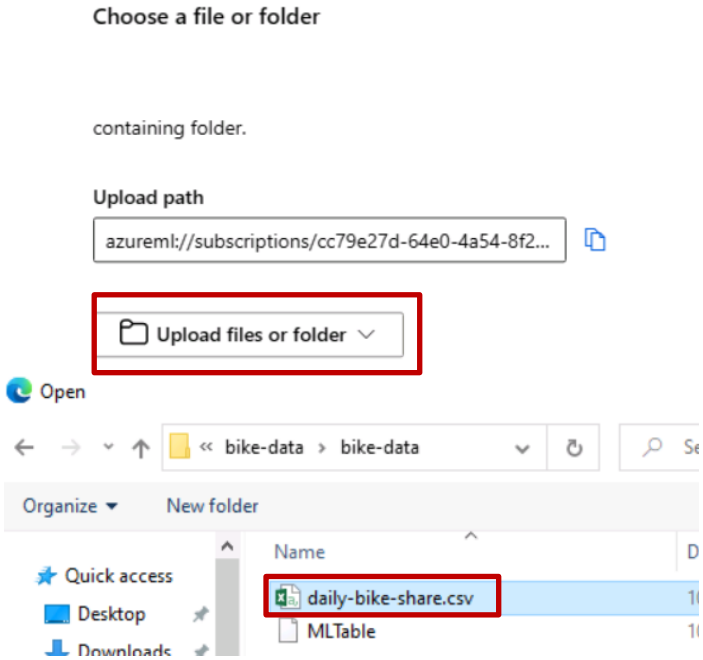
5. Download the data from: https://aka.ms/bike-rentals. Go to the folder, and extract the bike-data.zip file



6. On the Data source, choose **From local files**, and click 'next'

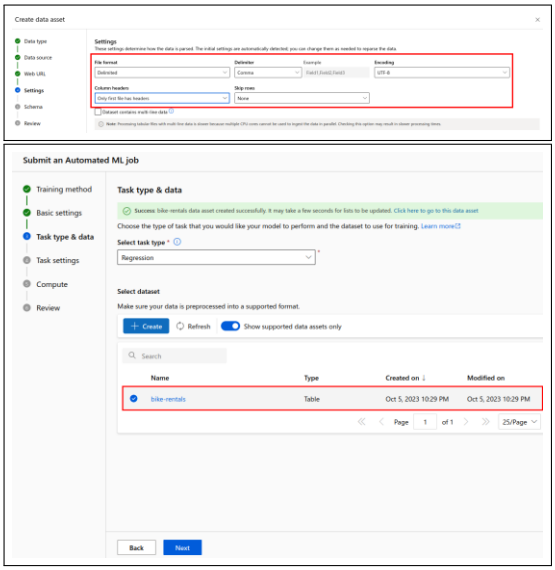


7. Click 'Upload files or folder', and choose the 'daily-bike-share.csv', and 'open'



A This **bike-rentals** dataset contains historical bicycle rental details used to train a model that predicts the number of bicycle rentals based on seasonal and meteorological features.

8. After that, complete the form with the following configuration settings (refer to Figure 2.6), and do not select **Skip data validation**. This is where you'll upload your data file to make it available to your workspace.



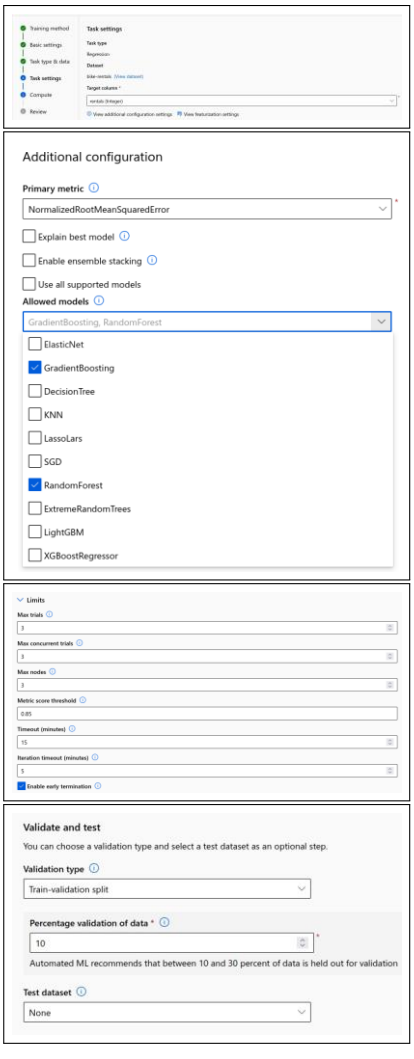
- **Settings:**
  - File format: Delimited
  - Delimiter: Comma
  - Encoding: UTF-8
  - Column headers: Only first file has headers
  - Skip rows: None
  - Dataset contains multi-line data: *do not select*
- **Schema:**
  - Include all columns other than **Path**

Review the automatically detected types Select the

bike-rentals dataset after you've created it.

Fig. 2.6: Data asset settings.

9. After you load and configure your data in the **Task type & data**, continue configuring your experiment with the following settings in **Task settings** (refer to Figure 2.7):



- **Task type:** Regression
- **Dataset:** bike-rentals
- **Target column:** Rentals (integer)
- **Additional configuration settings:**
  - **Primary metric:** Normalized root mean squared error
  - **Explain best model:** *Unselected*
  - **Use all supported models:** Unselected. *You'll restrict the job to try only a few specific algorithms.*
  - **Allowed models:** *Select only **RandomForest** and **LightGBM** — normally you'd want to try as many as possible, but each model added in- creases the time it takes to run the job.*
- **Limits:** Expand this section
  - **Max trials:** 3
  - **Max concurrent trials:** 3
  - Max nodes: 3
  - **Metric score threshold:** 0.85 *(so that if a model achieves a normalized root mean squared error metric score of 0.085 or less, the job ends.)*
  - **Timeout:** 15
  - **Iteration timeout: 15**
  - **Enable early termination:** Selected
- **Validation and test:**
  - **Validation type:** Train-validation split
  - **Percentage of validation data:** 10
  - **Test dataset:** None

Fig. 2.7: Task settings.

10. Select **Serverless** cluster as your compute type. A compute type is a local or cloud-based resource envi- ronment used to run your training script or host your service deployment. For this experiment, you use a cloud-based serverless compute.

- **Compute:**
  - **Select compute type:** Serverless
  - **Virtual machine type:** CPU
  - **Virtual machine tier:** Dedicated
  - **Virtual machine size:** Standard_DS3_V2
  - **Number of instances:** 1

Fig. 2.8: Compute settings.

11. Ensure everything is correctly set up as per the instructions and figures provided and **Submit** the training job. The system will now train multiple machine learning models based on your configurations and data. It starts automatically. Wait for the job to finish. It might take a while.

> **NOTE:**
>
> In production, you'd likely walk away for a bit. The Job Detail screen opens with the Job status at the top as the experiment preparation begins. This status updates as the experiment progresses. Notifications also appear in the top right corner of the studio to inform you of the status of your experiment.

Exercise - Part 2: Explore and review models

When the automated machine learning job has completed, you can monitor the progress of your experiment and access the results.

A You may see a message under the status "Warning: User specified exit score reached. . . ". This is an expected message. Please continue to the next step.

1. On the Overview tab of the automated machine learning job, note the best model summary.

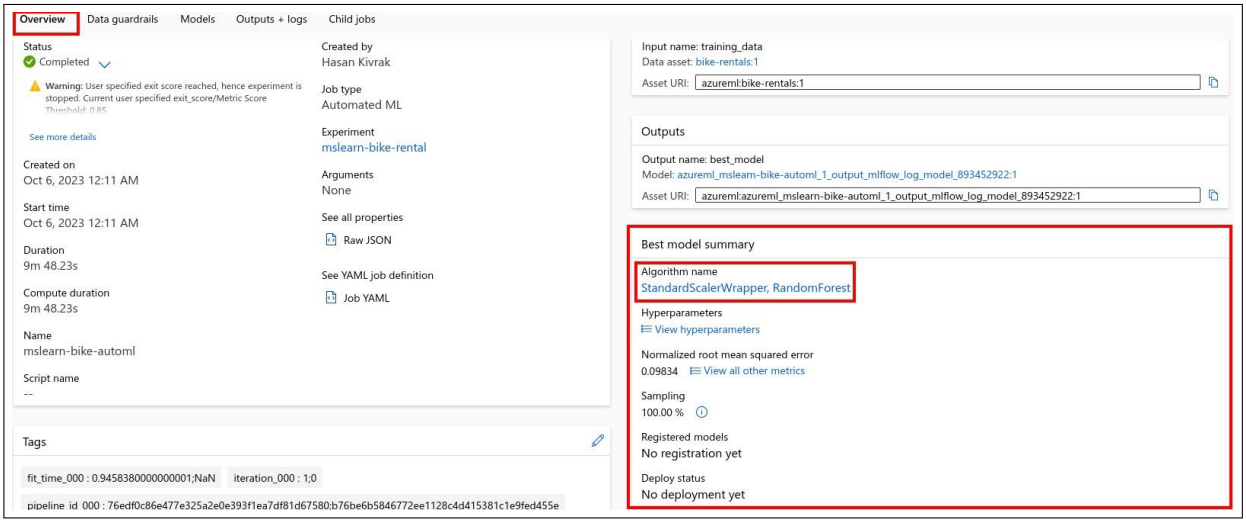


Fig. 2.9: Resource deployment done.

2. Select the text under **Algorithm name** for the best model to view its details.
3. Select the **Metrics** tab and select the **residuals** and **predicted true** charts if they are not already selected. Review the charts which show the performance of the model. The residuals chart shows the residuals (the differences between predicted and actual values) as a histogram. The predicted true chart compares the predicted values against the true values.

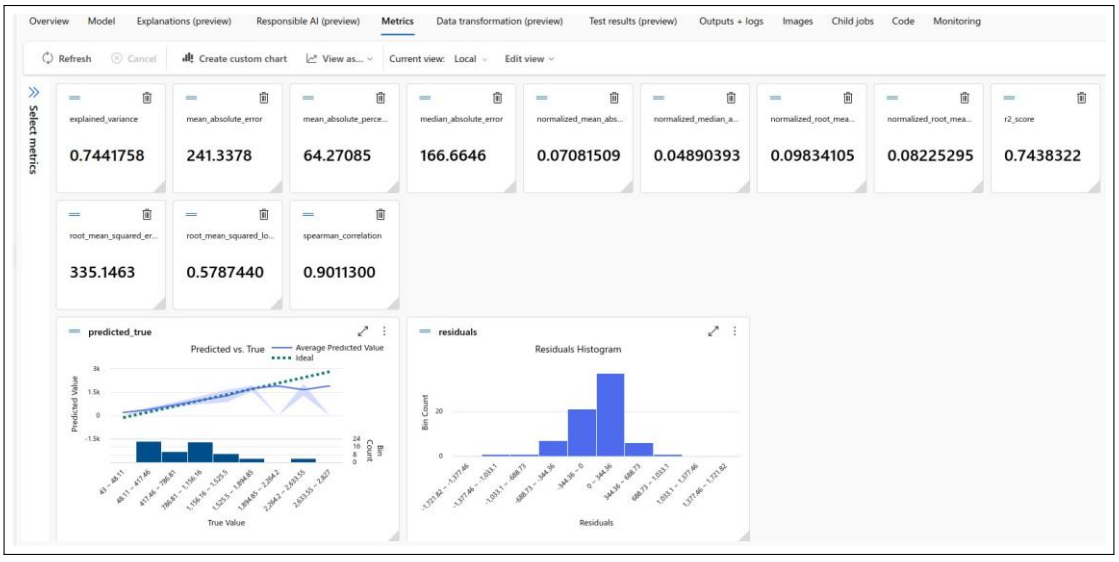


Fig. 2.10: Performance metrics and charts.

Model explanations:

You can also take a look at model explanations and see which data features influenced a particular model's predictions. These model explanations can be generated on demand, and are summarized in the model explanations dashboard that's part of the Explanations (preview) tab. To do so:

4. Select the **Explanations(preview)** button at the top and then click **Explain model**. Select the compute instance that you created previously. This compute instance initiates a child job to generate the model explanations. Select **Create** at the bottom. A green success message appears towards the top of your screen.

Fig. 2.11: Model explanations.

A If the computing instance is stopped you need to start it selecting **Compute** (under Manage) on the left pane and start it then.

NOTE:

The explainability job takes about 2-5 minutes to complete.

5. Next, click the **Explanations (preview)** button again to refresh. This tab populates once the explainability run completes.
6. On the left hand side, expand the pane and select the **row** that says raw under **Features**.
7. Select the **Aggregate feature importance** tab on the right. This chart shows which data features influ- enced the predictions of the selected model.
   In this example, the *working day* appears to have the most influence on the predictions of this model.

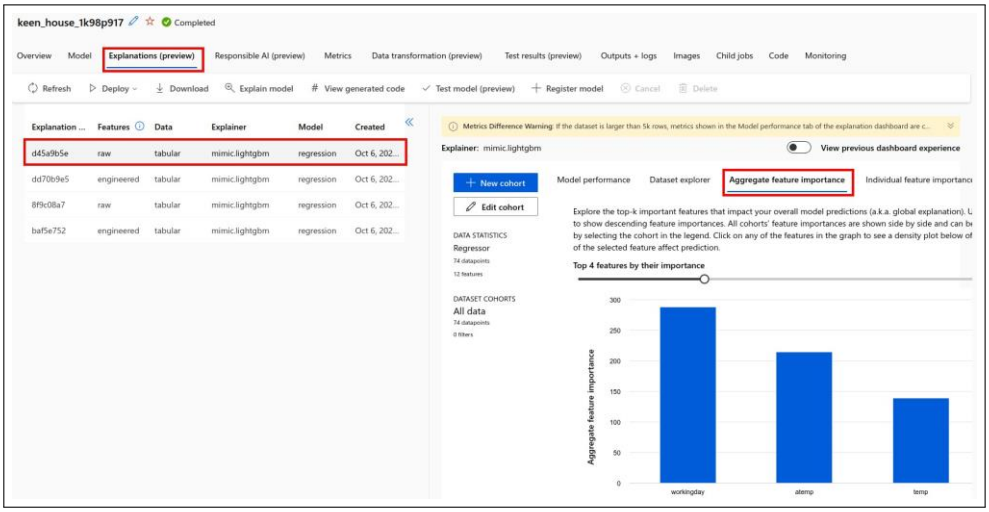


Fig. 2.12: Feature importance.

Exercise - Part 3: Extension Exercise

You are going to carry out same Automated ML pipeline for a regression task on **Ames Housing Dataset**. The data set contains information on 2,930 properties in Ames, Iowa, including columns related to:

- house characteristics (bedrooms, garage, fireplace, pool, porch, etc.)
- location (neighborhood)
- lot information (zoning, shape, size, etc.)
- ratings of condition and quality

- sale price

Our goal is to predict the sale price of a house based on other information we have, such as its characteristics and location. Start by creating a new worskpace, train and test models in the workspace, and report the test error you obtain. Follow the same following guideleness you previously followed for the **bike-rentals** dataset.

1. Create and load a dataset from web files
   - **Web URL:** https://raw.githubusercontent.com/wblakecannon/ames/master/data/housing.csv
2. Configure and run an automated ML experiment.
3. Explore the experiment results.
4. Deploy the best model.
5. Test the best model
6. Clean-up resources