**DEPLOYING AZURE ML MODEL USING IRIS DATASET OR DIABETES**

**Using Iris Dataset:**

**1. INTRODUCTION AND OVERVIEW**

**Purpose:** *This document aims to guide stakeholders through the process of deploying a machine learning model trained on the Iris dataset using Azure Machine Learning.*

**Audience:** *Data scientists, developers, and operations teams involved in model deployment and maintenance.*

I used Iris Flower Dataset from Kuggle website, which has 150 samples of iris flower which has three species; Virginica, Versicolor, and Setosa. I have to train ML Model to be able to classify whether an Iris flower is Virginica, Versicolor, or Setosa by looking at the following features: Sepal Length , Sepal Width, Petal Length and Petal Width.

**2. SYSTEM ARCHITECTURE**

**Diagrams:** *Include a flowchart or architectural diagram showing how different components like Azure ML Workspace, Compute Instance and Model Deployment interact*.

**3. DEPLOYMENT ENVIRONMENT**

**Hardware specifications:**

*System Manufacturer – HP*

*Processor – 12th Gen Intel(R) Core(TM) i7-1255U, 1700 Mhz, 10 Core(s), 12 Logical Processor(s)*

*Hardware Abstraction Layer – Version= “10.0.22621.2506”*

*BIOS Version/Date – AMI F.19, 2023/07/03*

*RAM – 16.0 GB*

*Total Physical Memory – 15.7 GB*

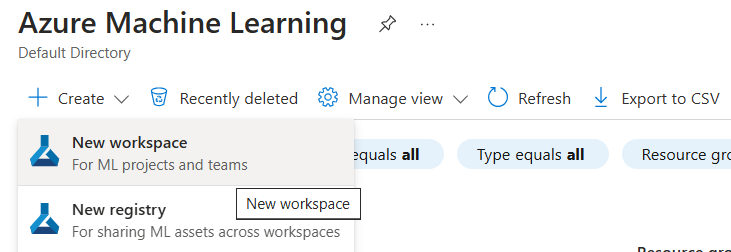
*Total Virtual Memory – 32.6 GB*

**Software dependencies:** *Azure ML SDK, Python 3.8*

**Operating System**:*I am using Microsoft Windows 11 home Single Language*

**4. DEPLOYMENT STEPS**

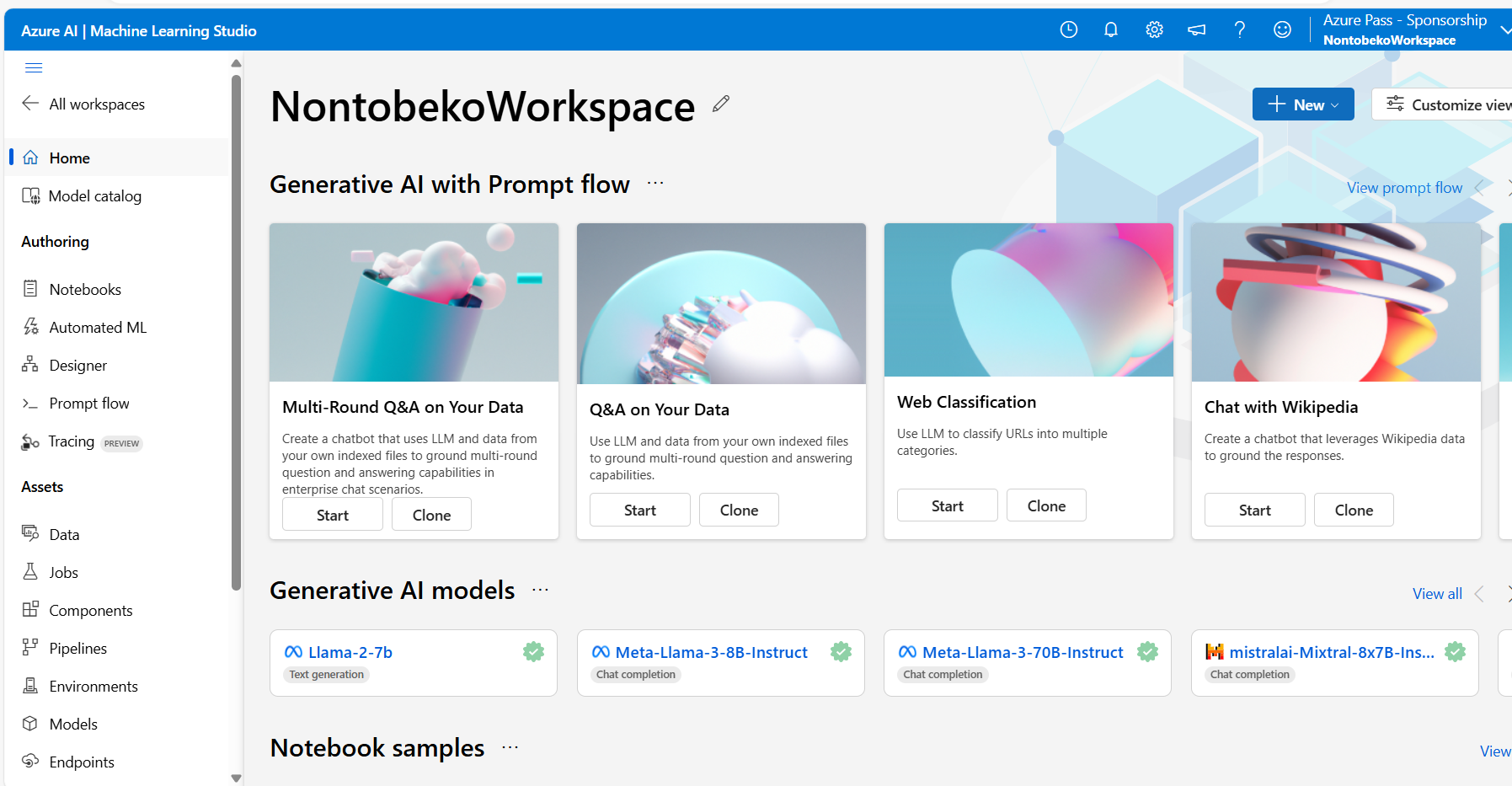
**Step 1 : Create AIML Workspace**

****

**A screenshot of a computer

Description automatically generated**

**Step 2 : Open the workspace studio**

****

**Step 3 : Upload the Iris dataset under Data component within the workspace.**

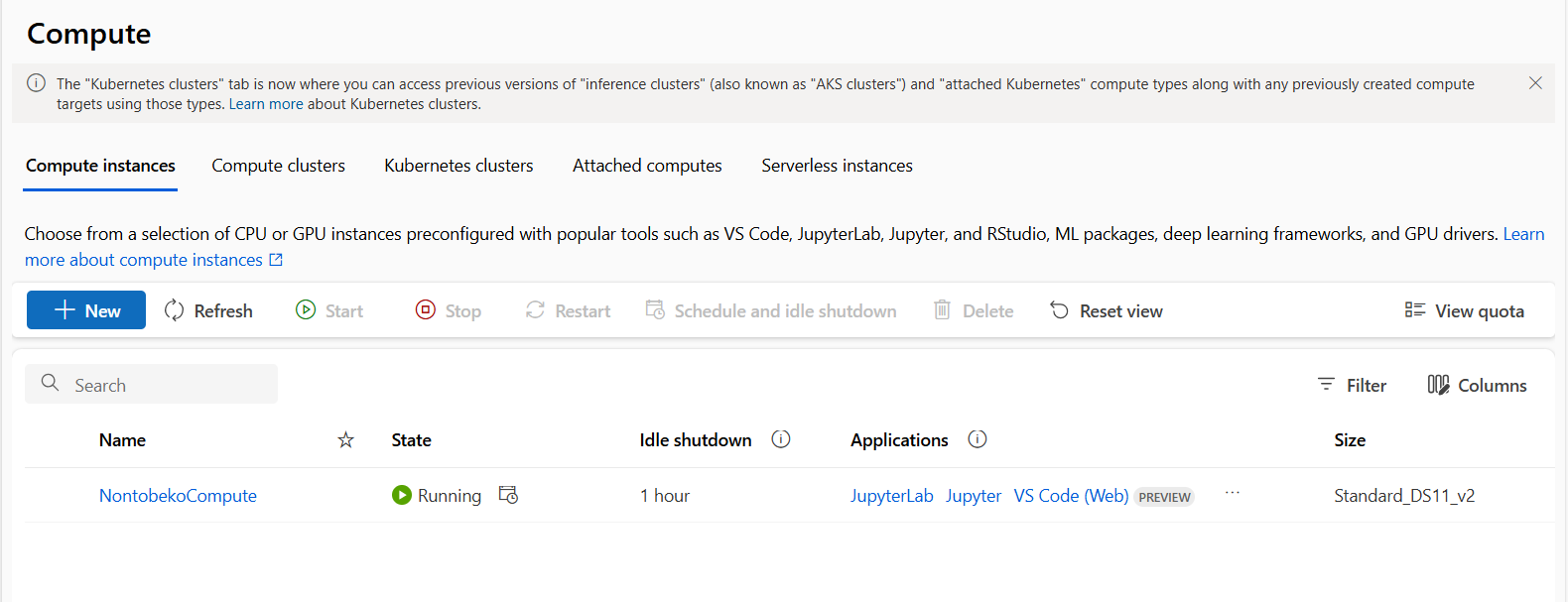
**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Step 4: Create the compute instance and open the Jupyter Lab**

****

**Step 5: Use the python 3.8 Azure ML**

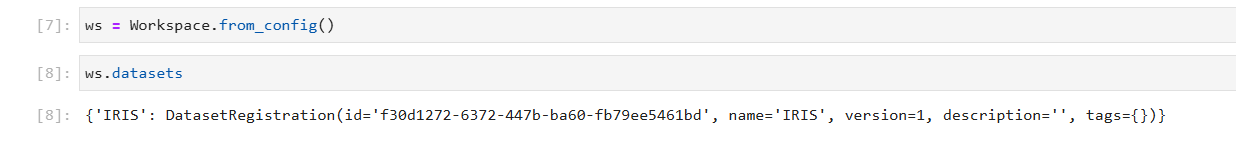
**Import the necessary packages and libraries**

****

**A screenshot of a computer

Description automatically generated**

**Step 6: Connect to your workspaces**

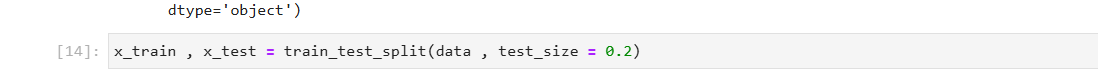
****

**Step 7: Work with the datasets and read your data. (I used Pandas)**

**A screenshot of a computer

Description automatically generated**

**Step 8 : Train and split your data**

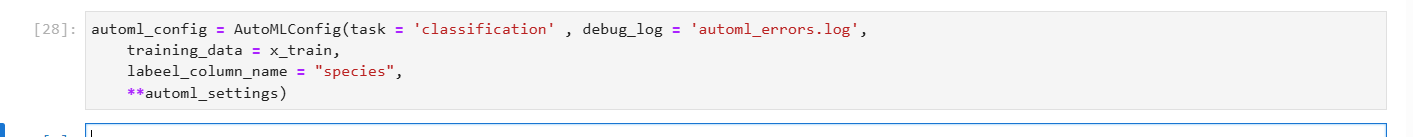
****

**Step 9: Set up your automl and your experiments settings**

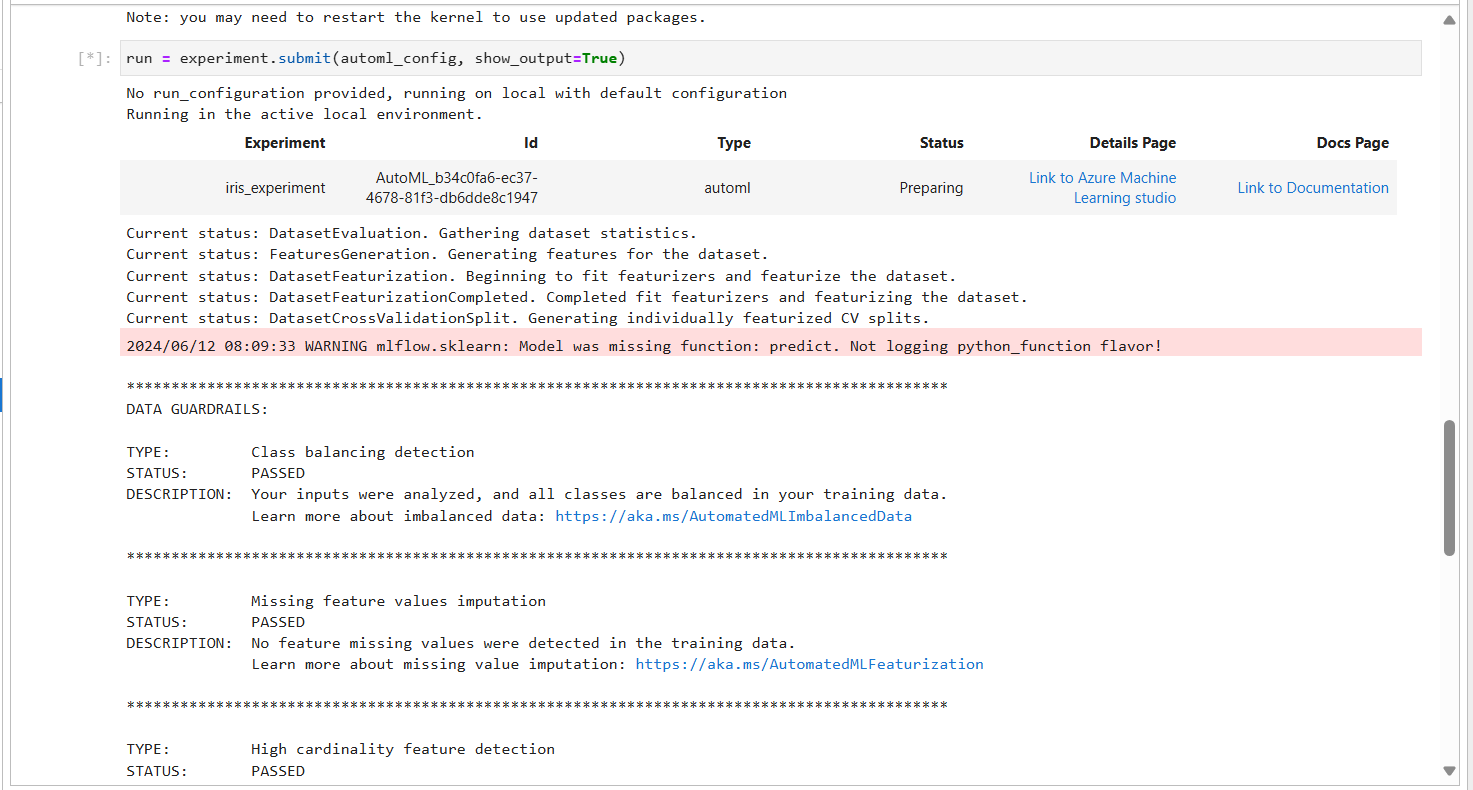
**A screenshot of a computer

Description automatically generated**

**Step 10: Specify the task and algorithm to use and the specie column as your label (dependent variable)**

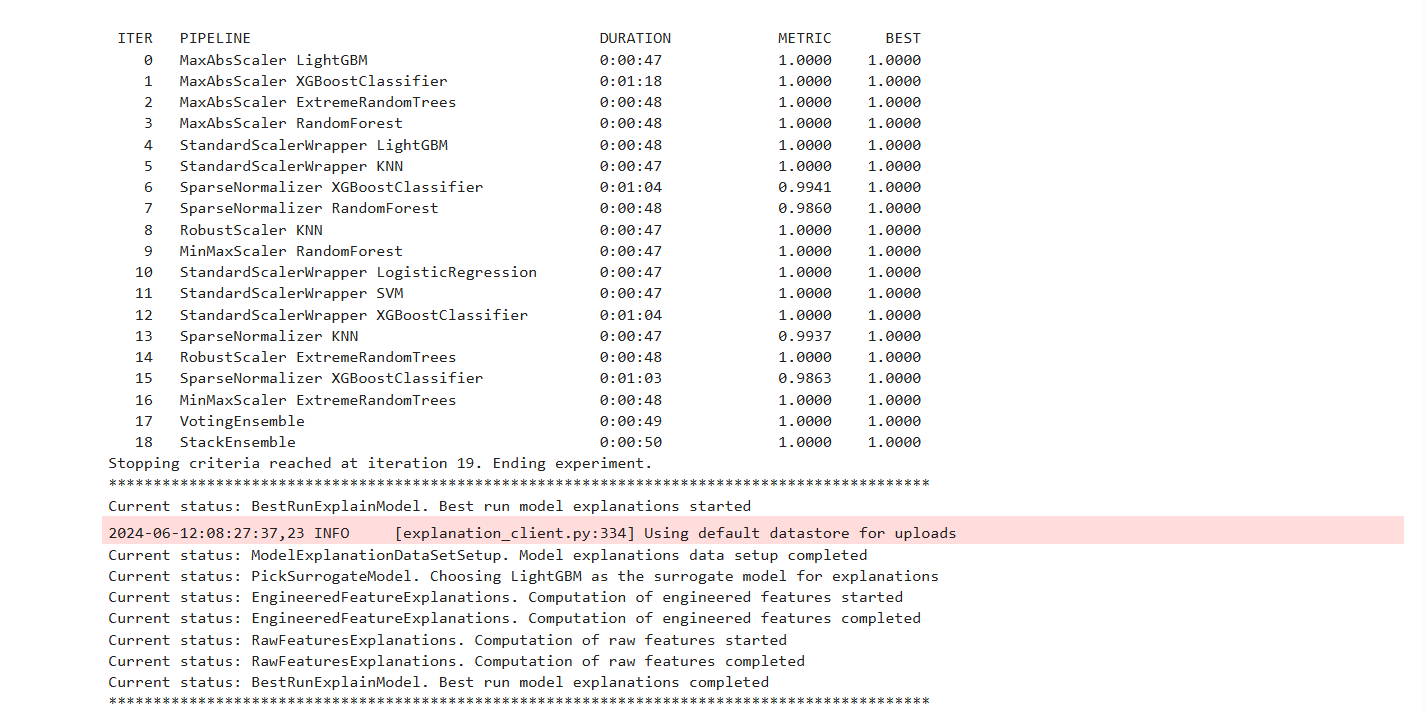
****

**Step 11: Create your experiment to use for deployment**

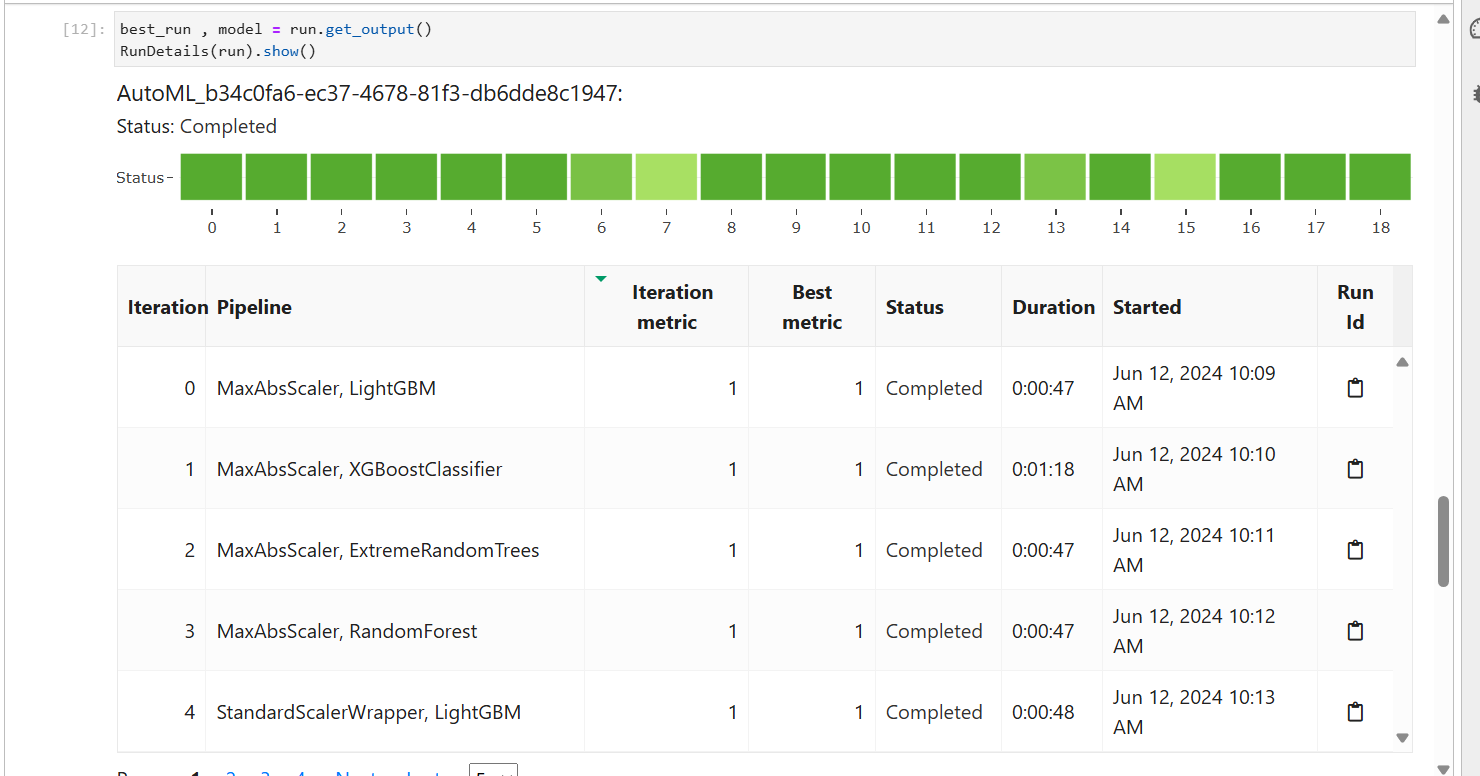
****

**A screenshot of a computer

Description automatically generated**

****

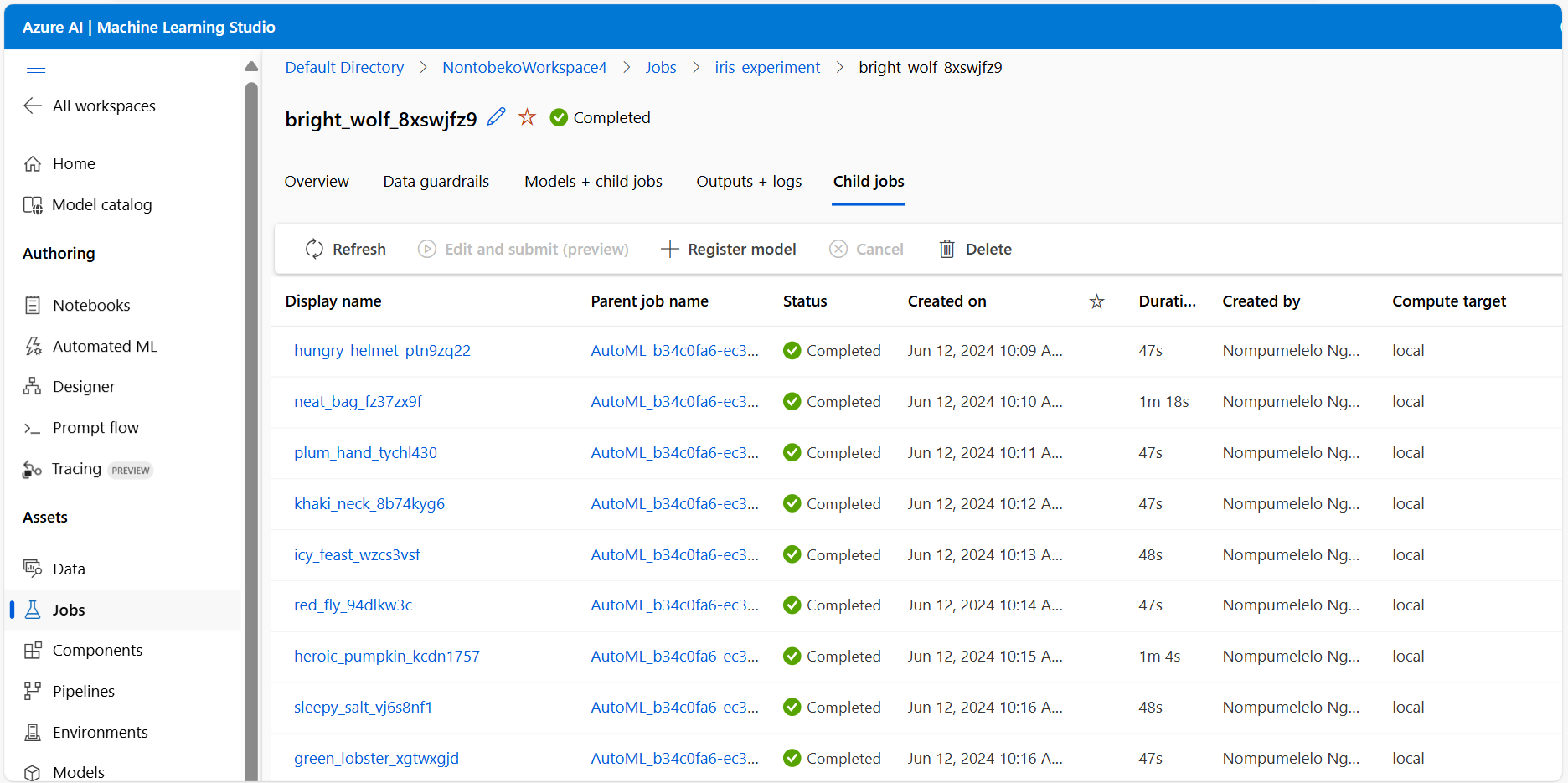
**Step 12: Get the run output**

****

**A screen shot of a graph

Description automatically generated**

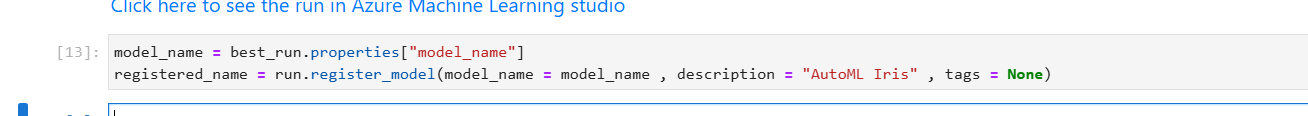
**Step 13: See the experiment and the scoring file created**

****

**A screenshot of a computer

Description automatically generated**

**Step 14: Create and register your model**

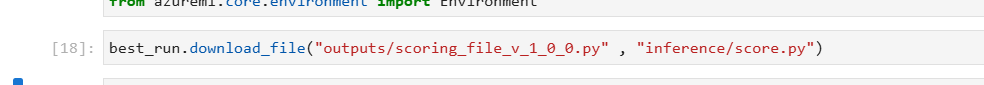
****

**Step 15: Import the packages for deployments**

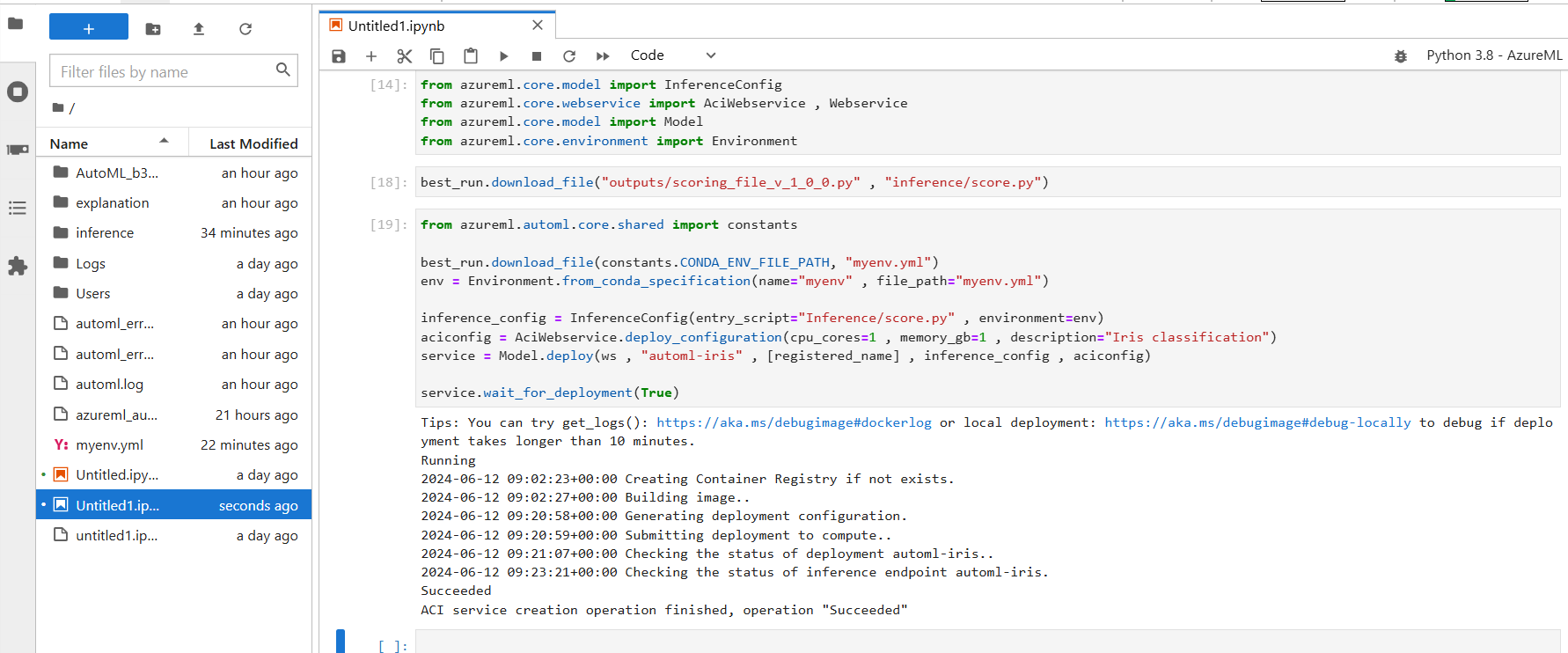
**A screenshot of a computer

Description automatically generated**

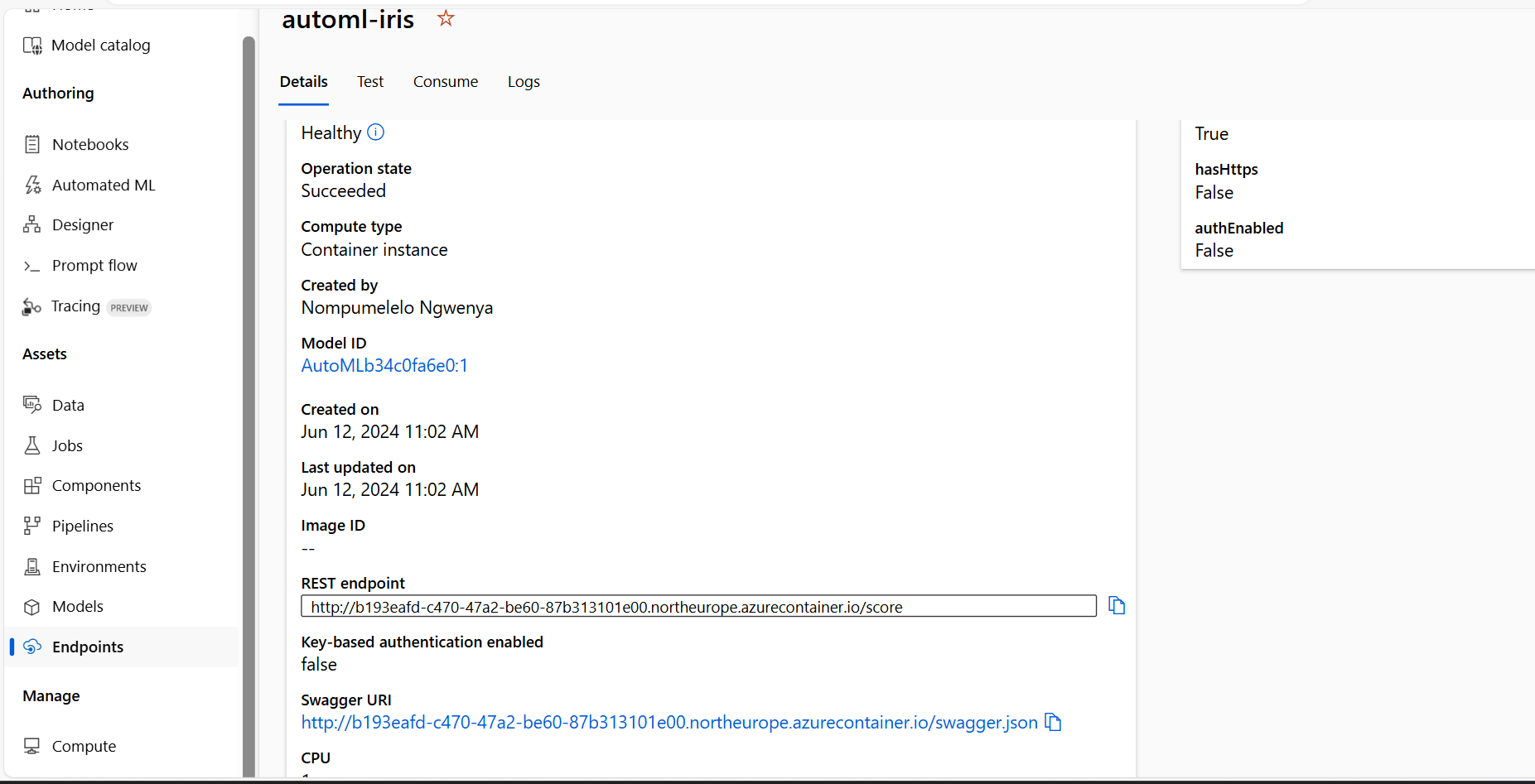
**Step 16: Download and bring in the scored .py file**

****

**Step 17: Wait for the deployment to complete (10 – 20mins)**

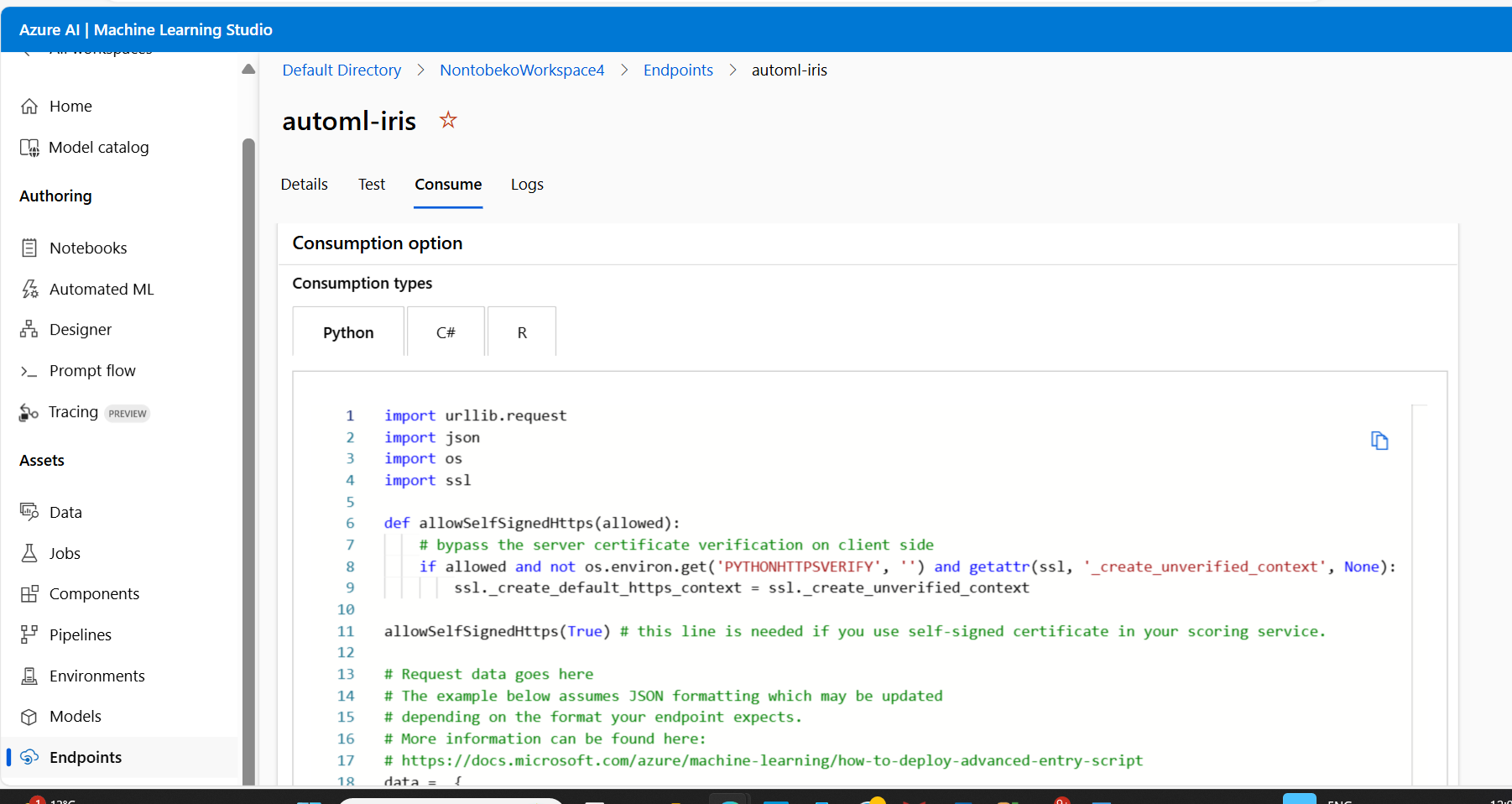
****

**Step 18: Look at the completed deployment and copy the url link and test it**

****

**Test the predicted result: REST Endpoint link**

<http://b193eafd-c470-47a2-be60-87b313101e00.northeurope.azurecontainer.io/score>

****

**5. CONFIGURATION SETTINGS**

*automl\_settings = {*

*“iteration\_timeout\_minutes”:2,*

*“experiment\_timeout\_minutes”:15,*

*“enable\_early\_stopping”:True,*

*“primary\_metric” : ‘AUC\_weighted’ ,*

*“featurization”: ‘auto’,*

*“n\_cross\_validation”:5,*

*}*

*automl\_config = AutoMLConfig(task = ‘classification’ , debug\_log = ‘automl\_errors.log’,*

*training\_data = x\_train,*

*label\_column\_name = ‘Species’,*

*\*\*automl\_settings*

**6. TESTING AND VALIDATION**

*Dataset : Iris Dataset (*[*https://www.kaggle.com/datasets/arshid/iris-flower-dataset*](https://www.kaggle.com/datasets/arshid/iris-flower-dataset)*)*

*Test the predicted result: REST EndPoint*

[*http://b193eafd-c470-47a2-be6087b313101e00.northeurope.azurecontainer.io/score*](http://b193eafd-c470-47a2-be6087b313101e00.northeurope.azurecontainer.io/score)

*Procedure for testing the deployed model to ensure it performs as expected:*

1. Environment Configuration
2. *Data Preparation*
3. *Input Data Validation*
4. *Testing (Use typical examples of data that the model is expected to encounter in production)*
5. *Prediction Output & Accuracy Assessment*
6. *Performance Testing (Latency & Throughput*
7. *Integration Testing*
8. *Validation Against Baselines*
9. *Bias and Fairness Testing (if applicable)*
10. *Documentation of Testing Results*
11. *Based on testing results, iteratively refine the model if necessary, addressing any identified issues or performance gaps.*

**7. MONITORING AND LOGGING**

*Monitoring the performance and health of a deployed model is crucial for ensuring it continues to operate effectively and meets service level expectations. Azure provides several tools and services that can be leveraged for performance monitoring.*

***Azure Monitor***

*Metrics - Collects performance metrics such as CPU usage, memory usage, and response times of the deployed model endpoint.*

*Alerts - Set up alerts based on predefined thresholds for metrics, for example if response time exceeds a certain limit.*

*Logs: Azure Monitor can also collect logs from various Azure services, including Application Insights and Azure Machine Learning, to provide deeper insights into model performance.*

*Logging mechanisms are essential for troubleshooting, debugging, and auditing purposes.*

***Azure Monitor Logs***

*Querying Logs - Use Azure Monitor Logs to query and analyse logs collected from various Azure services, including Application Insights and Azure Machine Learning.*

*Log Analytics - Leverage Log Analytics to perform advanced queries, create dashboards, and gain insights into the operational health of the deployed model.*

**8. SCALABILITY AND PERFORMANCE**

***Scalability Considerations***

*When scaling a machine learning model to handle increased traffic or larger datasets on Azure, several considerations come into play to ensure optimal performance and efficiency:*

* *Compute Resources, Auto-scaling, Data Storage, Load Balancing, and Caching.*

***Performance Optimization***

*Optimizing the performance of a machine learning model on Azure involves enhancing its speed, efficiency, and resource utilization. Here are techniques and benchmarks for achieving optimal performance:*

* *Model Optimization, Hardware Acceleration, Batch Processing, Model Compression, Pipeline Optimization, Benchmarking and Monitoring.*

**9. SECURITY CONSIDERATIONS**

***Security Measures***

*Ensuring robust security measures are implemented during the deployment of machine learning models on Azure is crucial to protect data, maintain privacy, and comply with regulatory requirements. Security practices and compliance considerations:*

* *Authentication and Authorization,* *Network Security, and Data Encryption*

***Compliance***

* *Regulatory Compliance, Data Privacy, Audit and Compliance Reporting, Legal and Ethical Considerations.*

**10. MAINTENANCE AND SUPPORT**

***Maintenance Guidelines***

*Maintaining a deployed machine learning model on Azure involves regular updates, monitoring, and ensuring the model continues to perform effectively. Here are guidelines for maintaining and updating the deployed model over time:*

* *Version Control, Monitoring and Performance Evaluation, Regular Updates and Retraining, Security Updates*

***Troubleshooting***

*Documenting common issues and troubleshooting steps is essential for efficiently resolving issues that may arise during the deployment and maintenance of machine learning models on Azure. Here are common issues­ and corresponding troubleshooting steps:*

* *Problem: Model predictions are inaccurate due to changes in input data quality or distribution.*

*Troubleshooting: Implement data drift monitoring and retrain the model periodically using updated datasets.*

* *Problem: Unauthorized access or data breach related to Azure resources hosting the model.*

*Troubleshooting: Review Azure Security Center alerts and audit logs for suspicious activities. Implement Azure AD authentication and RBAC to restrict access.*