DP-100: Microsoft Certified: Azure Data Scientist Associate

Predictive Maintenance using Azure ML and IoT Data

By

Miriam El Hage Sleiman

A Portfolio Project Created to Showcase Competencies and Skills Acquired from Microsoft's AI Engineer Associate and Data Scientist Associate Certifications.

SUMMARY DESCRIPTION OF THE PROJECT

Project Developer:
Miriam El Hage Sleiman
Senior Mechatronics Engineering Student at AUST Lebanon

Project Title:
Predictive Maintenance using Azure ML and IoT Data

Azure Subscription:
Free trial

Date:
October 2025

Keywords:
Predictive Maintenance, IoT, Azure ML, Machine Learning, Industrial AI

Abstract:
This project demonstrates the development and deployment of a predictive maintenance system using Azure Machine Learning and simulated IoT sensor data. The project leverages a synthetic dataset representing machine telemetry such as temperature, vibration, pressure, voltage, and hours since last maintenance for the purpose of training a Random Forest classifier capable of predicting maintenance requirements. The model was trained and tested within an Azure ML notebook, then registered and deployed as a real-time endpoint using Azure Container Instances. Endpoint testing was conducted using sample JSON payloads in Postman, confirming the system's ability to provide accurate real-time predictions. This project showcases the practical application of AI and machine learning concepts in industrial monitoring scenarios and serves as a practical demonstration of skills aligned with Microsoft's AI Engineer Associate and Data Scientist Associate certifications.

TABLE OF CONTENTS

LIST OF FIGURES

I. Introduction

This project demonstrates how predictive maintenance can be achieved by integrating Azure Machine Learning [1] and IoT telemetry data. It expands the scope of my previous Smart CCTV project (also present in this portfolio) where I used Azure IoT Hub for edge-to-cloud communication and Computer Vision to analyze scenes. For this continuation, the system concept was modified to simulate sensor-based monitoring of industrial machinery, with the goal of predicting maintenance needs using machine learning.

II. Objective

The objective of this project was to design, train, and deploy a machine learning model capable of predicting whether a machine requires maintenance based on telemetry data such as temperature, vibration, pressure, voltage, and hours since last maintenance. Thus, practicing concepts learned from the DP-100 (Azure Data Scientist Associate) certification and applying them in an IoT-based industrial scenario [2].

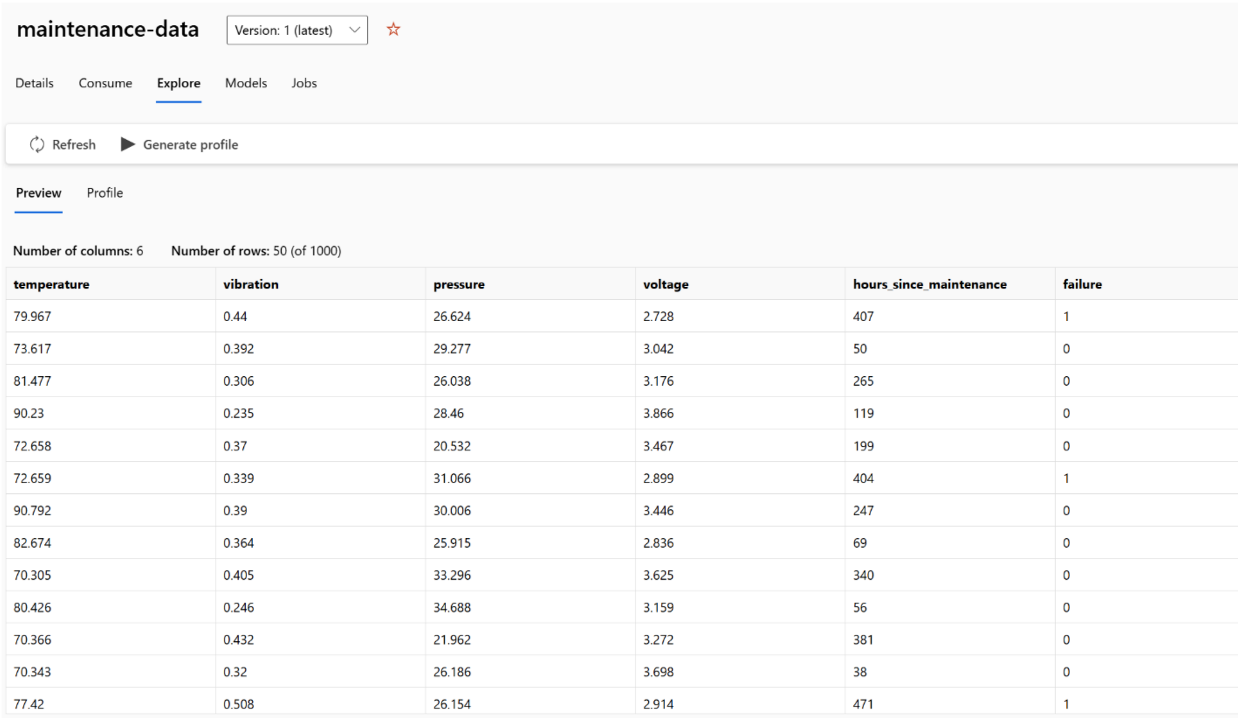
III. System Overview

The proposed setup is an adaptation of my previous Smart CCTV IoT project. Instead of relying on visual data, it uses virtual sensors to monitor various machine parameters. These simulated sensors send data to Azure IoT Hub, where it is compiled into a CSV dataset representing the factory's operational telemetry. This dataset forms the basis for training the predictive maintenance model.

IV. Dataset

A synthetic dataset containing 1,000 records for the following features of temperature (°C), vibration (m/s^2), pressure (bar), voltage (V), hours_since_maintenance (h), and failure (Boolean label representing the need for maintenance) was created to simulate machine telemetry.

This dataset was saved locally as a CSV file and uploaded to Azure Machine Learning as a data asset named 'maintenance-data'. The data type was set to tabular, with UTF-8 encoding and comma delimiters.



temperature	vibration	pressure	voltage	hours_since_maintenance	failure
79.967	0.44	26.624	2.728	407	1
73.617	0.392	29.277	3.042	50	0
81.477	0.306	26.038	3.176	265	0
90.23	0.235	28.46	3.866	119	0
72.658	0.37	20.532	3.467	199	0
72.659	0.339	31.066	2.899	404	1
90.792	0.39	30.006	3.446	247	0
82.674	0.364	25.915	2.836	69	0
70.305	0.405	33.296	3.625	340	0
80.426	0.246	34.688	3.159	56	0
70.366	0.432	21.962	3.272	381	0
70.343	0.32	26.186	3.698	38	0
77.42	0.508	26.154	2.914	471	1

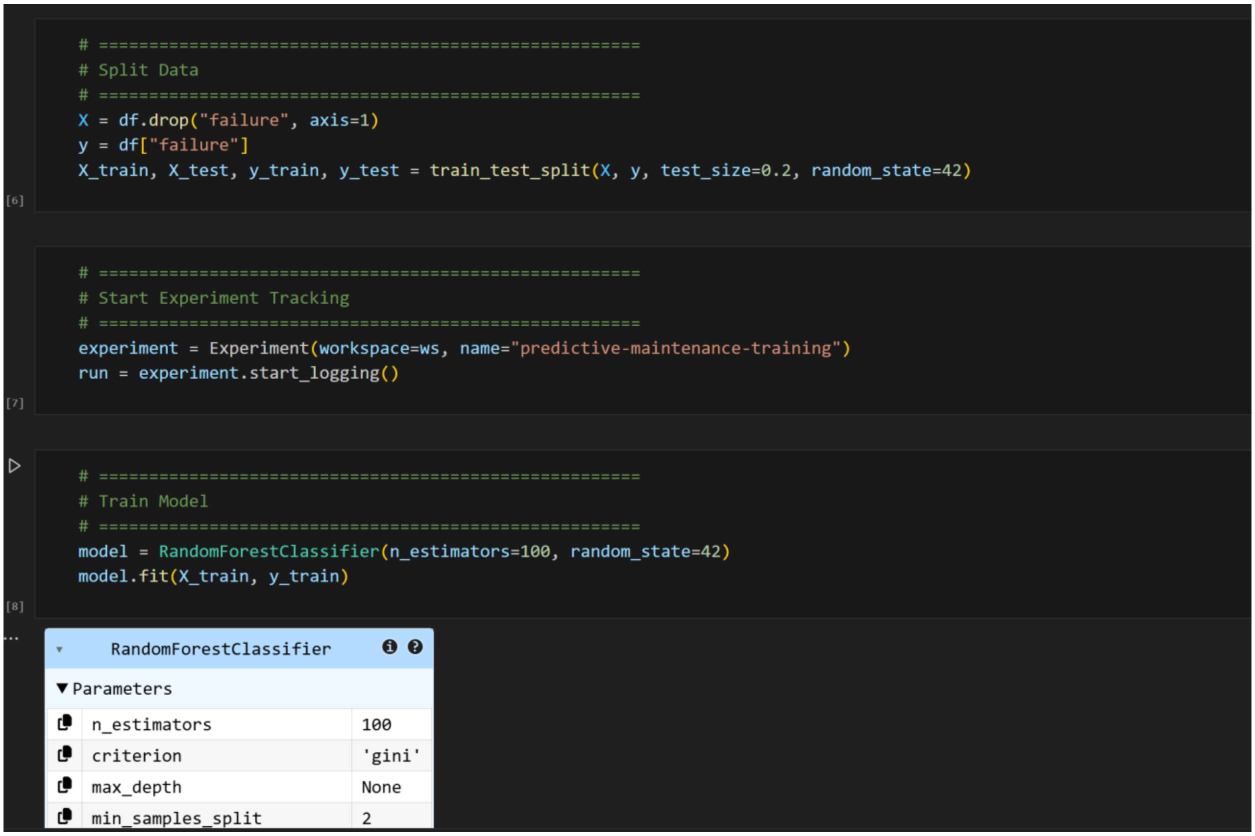
Figure 1. First rows of the dataset.

V. Model Development

Model training and evaluation were conducted within an Azure Machine Learning notebook. The following Python libraries were used: pandas for data handling and preprocessing, scikit-learn for building, training and evaluating the model, and azureml.core to connect to the Azure Machine Learning workspace in the cloud. A Random Forest Classifier was selected for this binary classification problem due to its robustness and ability to handle mixed numeric data efficiently.

The following steps were performed by the notebook:

1. Load the dataset from the registered data asset.
2. Split the data into training and testing sets (80/20).
3. Train the Random Forest model.
4. Evaluate performance using accuracy and F1-score.
5. Save the trained model as a .pkl file for deployment.



```
# ======================================================
# Split Data
# ======================================================
X = df.drop("failure", axis=1)
y = df["failure"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
[6]

# ======================================================
# Start Experiment Tracking
# ======================================================
experiment = Experiment(workspace=ws, name="predictive-maintenance-training")
run = experiment.start_logging()
[7]

# ======================================================
# Train Model
# ======================================================
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
[8]
```

RandomForestClassifier

▼ Parameters

n_estimators	100
criterion	'gini'
max_depth	None
min_samples_split	2

Figure 2. Development excerpt.

VI. Model Registration and Deployment

After successful training, the model was registered in Azure ML, and a real-time endpoint was created using Azure Container Instance (ACI) for inference. Deployment steps included defining the inference_config referencing the score.py script, creating a deployment configuration, and deploying the model to an ACI endpoint. Once RBAC permissions were updated to fix an issue pertaining to endpoint creation failure, the endpoint became accessible for real-time predictions [3].

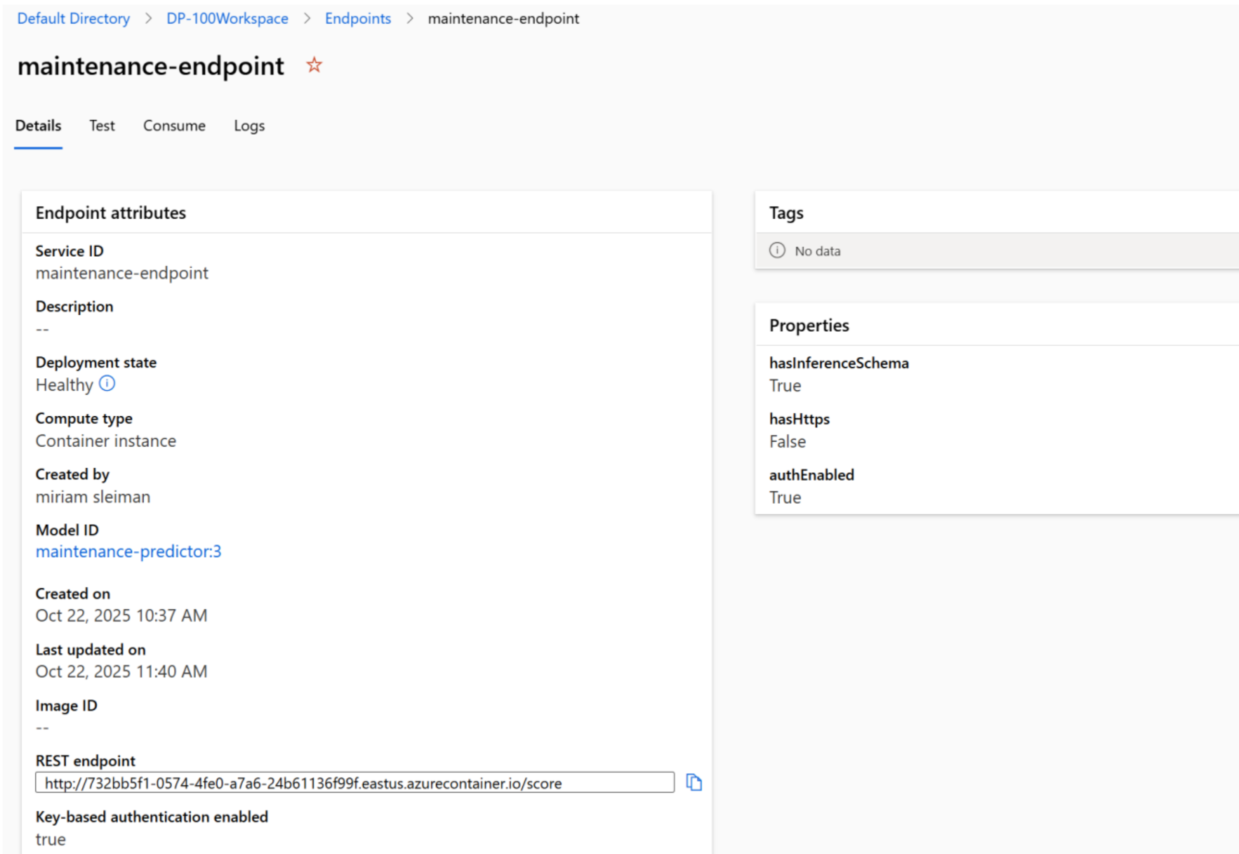


Figure 3. Endpoint in AML Studio.

VII. Endpoint Testing

The endpoint was tested using the notebook as well as Postman to simulate machine telemetry input. The two example JSON payloads below were used for Postman.

Example 1 –No maintenance:

```
{
    "data": [
      [72.4, 0.18, 29.1, 220, 310]
    ]
  }
```

Example 2 – Maintenance:

```
{
```

```
  "data": [
    [85.9, 0.62, 36.7, 205, 980]
  ]
}
```

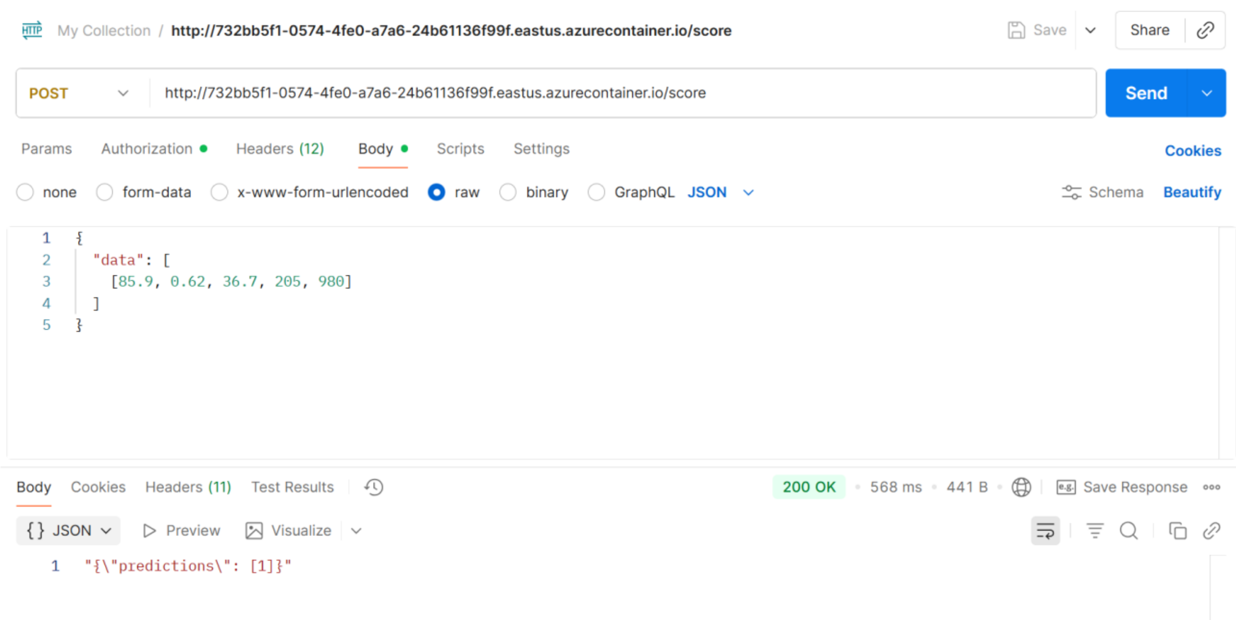


Figure 4. Maintenance is needed.

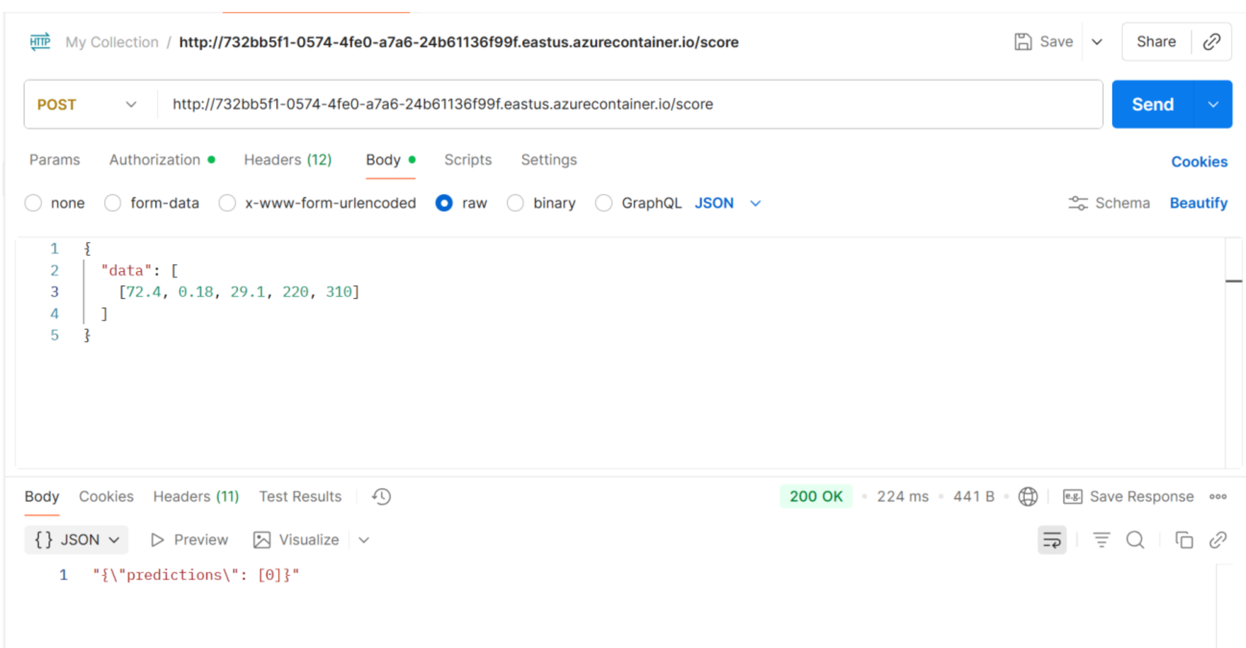


Figure 5. No maintenance is needed.

```python
# ============================================================
# Mimic IoT payload
# ===========================================================
import requests
import json

#Replace with your scoring URI
url = service.scoring_uri

#Replace with your actual API key
headers = {'Content-Type': 'application/json', 'Authorization': f'Bearer {key}'}

#Sample data (one record)
data = {
    "data": [[75.2, 0.22, 30.5, 220, 500]]
}

response = requests.post(url, headers=headers, data=json.dumps(data))

print("Response status:", response.status_code)
print("Prediction:", response.json())
```

```
Response status: 200
Prediction: {"predictions": [0]}
```

```python
data = {
    "data": [
        [80.1, 0.45, 32.1, 210, 900],
        [72.5, 0.18, 28.3, 225, 300]
    ]
}

response = requests.post(url, headers=headers, data=json.dumps(data))
print(response.json())
```

[8]

```
{"predictions": [0, 0]}
```

Figures 6,7. Notebook results.

It’s worth noting that all responses correctly predicted whether maintenance was required.

VIII. Results and Discussion

The Random Forest model [4] achieved high prediction accuracy with a score of 0.965 for precision as well as high recall and F1-score, demonstrating its capability to correctly identify sensor patterns that correlate with maintenance needs. The deployed endpoint also successfully responded to HTTP requests, confirming the feasibility of integrating Azure ML models into IoT-based maintenance systems.

```python
# =======================================================
# Evaluate Model
# =======================================================
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)

# Log metrics
run.log("accuracy", acc)
run.log("true_failures", int(y_test.sum()))
run.log("predicted_failures", int(y_pred.sum()))

print(f"✅ Accuracy: {acc:.3f}")
print("\nConfusion Matrix:\n", cm)
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
✅ Accuracy: 0.965

Confusion Matrix:
 [[182   0]
 [  7  11]]

Classification Report:
               precision    recall  f1-score   support

           0       0.96      1.00      0.98       182
           1       1.00      0.61      0.76        18

    accuracy                           0.96       200
   macro avg       0.98      0.81      0.87       200
weighted avg       0.97      0.96      0.96       200
```

Figure 8. Logged metrics with code.

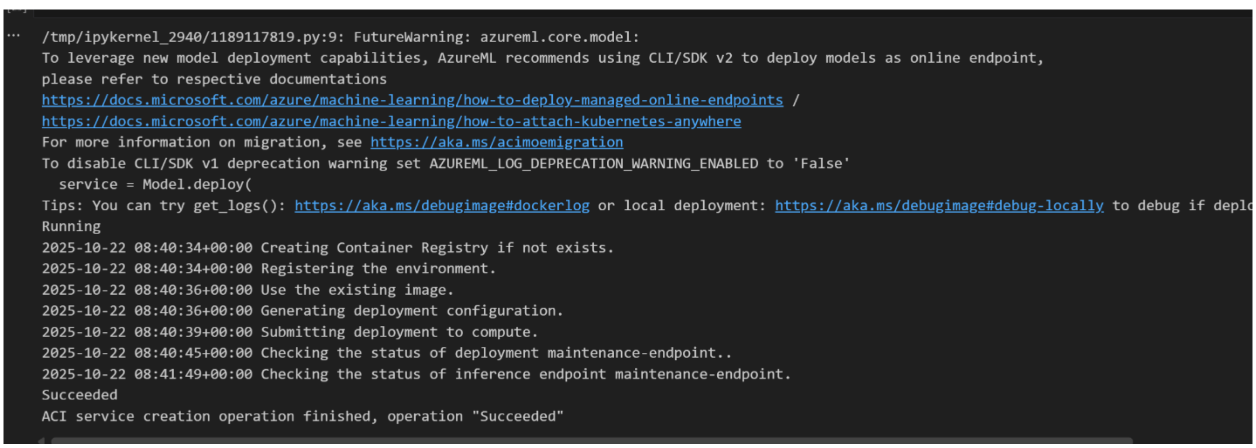
IX. Conclusion

This project successfully combined IoT concepts with Azure Machine Learning for predictive maintenance. It shows that IoT data can be used to train ML models and deploy them for real-time inference. The solution can be further extended to real industrial settings using live IoT data streams [5].

X. Evidence and Files

Supporting evidence includes:

- Jupyter notebook for data preparation, training, and deployment.
- Registered dataset (maintenance-data/iot_sensor_data.csv).
- Model file (predictive_maintenace_model.pkl).
- Deployed real-time endpoint (screenshots).
- Postman requests showing model inference.



```
/tmp/ipykernel_2940/1189117819.py:9: FutureWarning: azureml.core.model:
To leverage new model deployment capabilities, AzureML recommends using CLI/SDK v2 to deploy models as online endpoint,
please refer to respective documentations
https://docs.microsoft.com/azure/machine-learning/how-to-deploy-managed-online-endpoints /
https://docs.microsoft.com/azure/machine-learning/how-to-attach-kubernetes-anywhere
For more information on migration, see https://aka.ms/acimoemigration
To disable CLI/SDK v1 deprecation warning set AZUREML_LOG_DEPRECATION_WARNING_ENABLED to 'False'
  service = Model.deploy(
Tips: You can try get_logs(): https://aka.ms/debugimage#dockerlog or local deployment: https://aka.ms/debugimage#debug-locally to debug if deplo
Running
2025-10-22 08:40:34+00:00 Creating Container Registry if not exists.
2025-10-22 08:40:34+00:00 Registering the environment.
2025-10-22 08:40:36+00:00 Use the existing image.
2025-10-22 08:40:36+00:00 Generating deployment configuration.
2025-10-22 08:40:39+00:00 Submitting deployment to compute.
2025-10-22 08:40:45+00:00 Checking the status of deployment maintenance-endpoint..
2025-10-22 08:41:49+00:00 Checking the status of inference endpoint maintenance-endpoint.
Succeeded
ACI service creation operation finished, operation "Succeeded"
```

Figure 9. Endpoint creation confirmed in notebook.

References

[1] Microsoft, “Azure Machine Learning service overview,” Microsoft Learn, 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/machine-learning/. [Accessed: Oct. 14, 2025].

[2] Microsoft, “Exam DP-100: Design and Implement a Data Science Solution on Azure,” Microsoft Learn, 2025. [Online]. Available: https://learn.microsoft.com/en-us/certifications/exams/dp-100/. [Accessed: Sept. 13, 2025].

[3] Microsoft, “Create and deploy a machine learning model in Azure Machine Learning,” Microsoft Learn, 2025. [Online]. Available: https://learn.microsoft.com/en-us/training/modules/create-deploy-ml-model-azure-ml/. [Accessed: Oct. 12, 2025].

[4] Scikit-learn Developers, “Random Forest Classifier,” Scikit-learn Documentation, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: Oct. 20, 2025].

[5] Microsoft, “Connect IoT data to Azure Machine Learning for predictive maintenance,” Microsoft Learn, 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/iot-fundamentals/iot-predictive-maintenance. [Accessed: Oct. 19, 2025].