

---

# **CAPSTONE PROJECT**

## **POWER SYSTEM FAULT DETECTION AND CLASSIFICATION USING MACHINE LEARNING**

**Presented By:**

**1. AYUSH SHARMA GALGOTIAS UNIVERSITY – B.TECH CSE(AIML)**

# OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

---

# PROBLEM STATEMENT

Power distribution systems are prone to various types of faults such as line-to-ground, line-to-line, and three-phase faults. These faults can disrupt power supply and reduce system reliability. The challenge lies in accurately detecting and classifying these faults using electrical measurement data (voltage, current, phasors) to differentiate them from normal operating conditions, thereby ensuring the stability of the power grid.

# PROPOSED SOLUTION

- Develop a machine learning model that classifies power system faults using the dataset provided. The model will process electrical measurements to identify the type of fault rapidly and accurately. This classification will help automate fault detection and assist in quicker recovery actions, ensuring system reliability.
- **Key components:**
- **Data Collection:** Use the Kaggle dataset on power system faults.
- **Preprocessing:** Clean and normalize the dataset.
- **Model Training:** Train a classification model (e.g., Decision Tree, Random Forest, or SVM).
- **Evaluation:** Validate the model using accuracy, precision, recall, and F1-score.

# SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the Power System Fault Detection and Classification. Here's a suggested structure for this section:

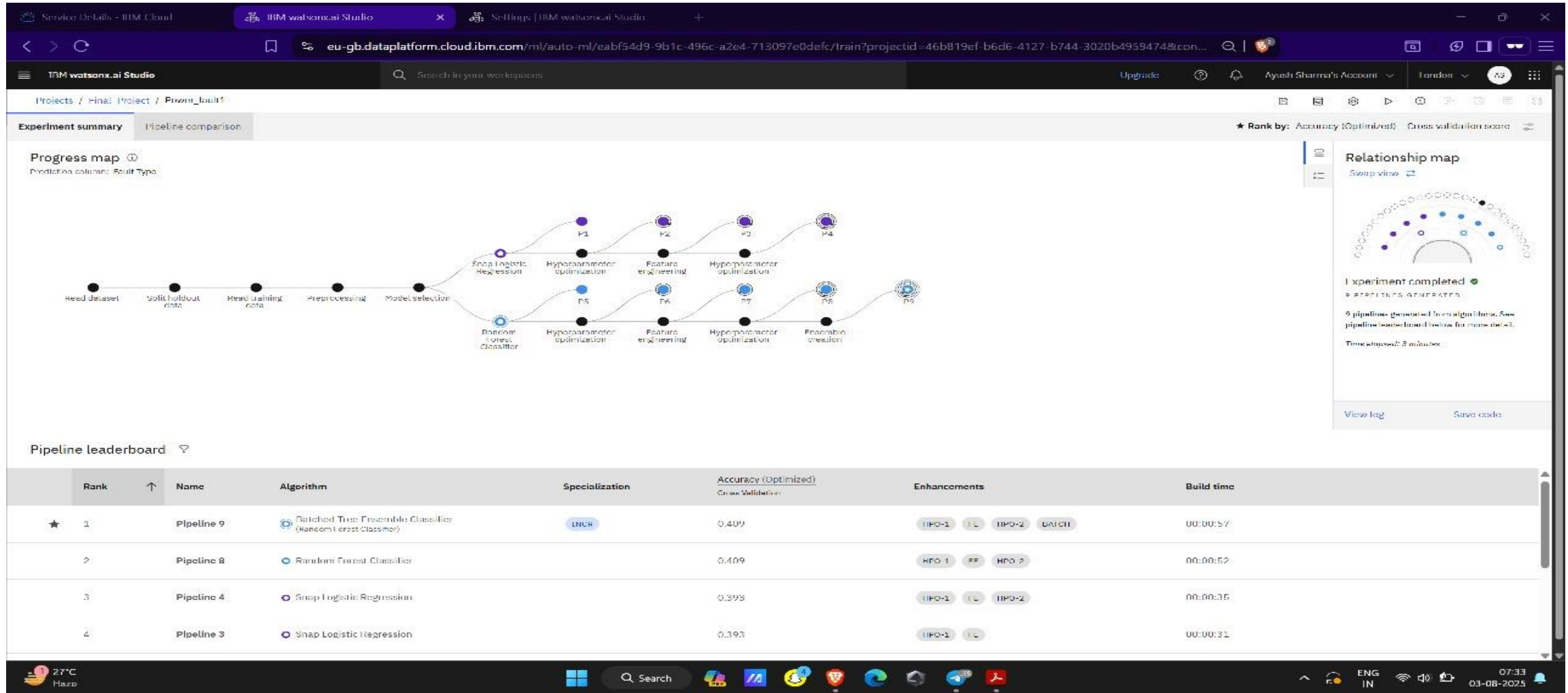
- System requirements
  - IBM Cloud (Mandatory)
  - IBM Watson studio for model development and deployment
  - IBM cloud storage for dataset handling
- Library required to build the model
  - Pandas – for data manipulation
  - NumPy – for numerical operations
  - Scikit-learn – for building and evaluating ML models
  - Matplotlib – for visualization

---

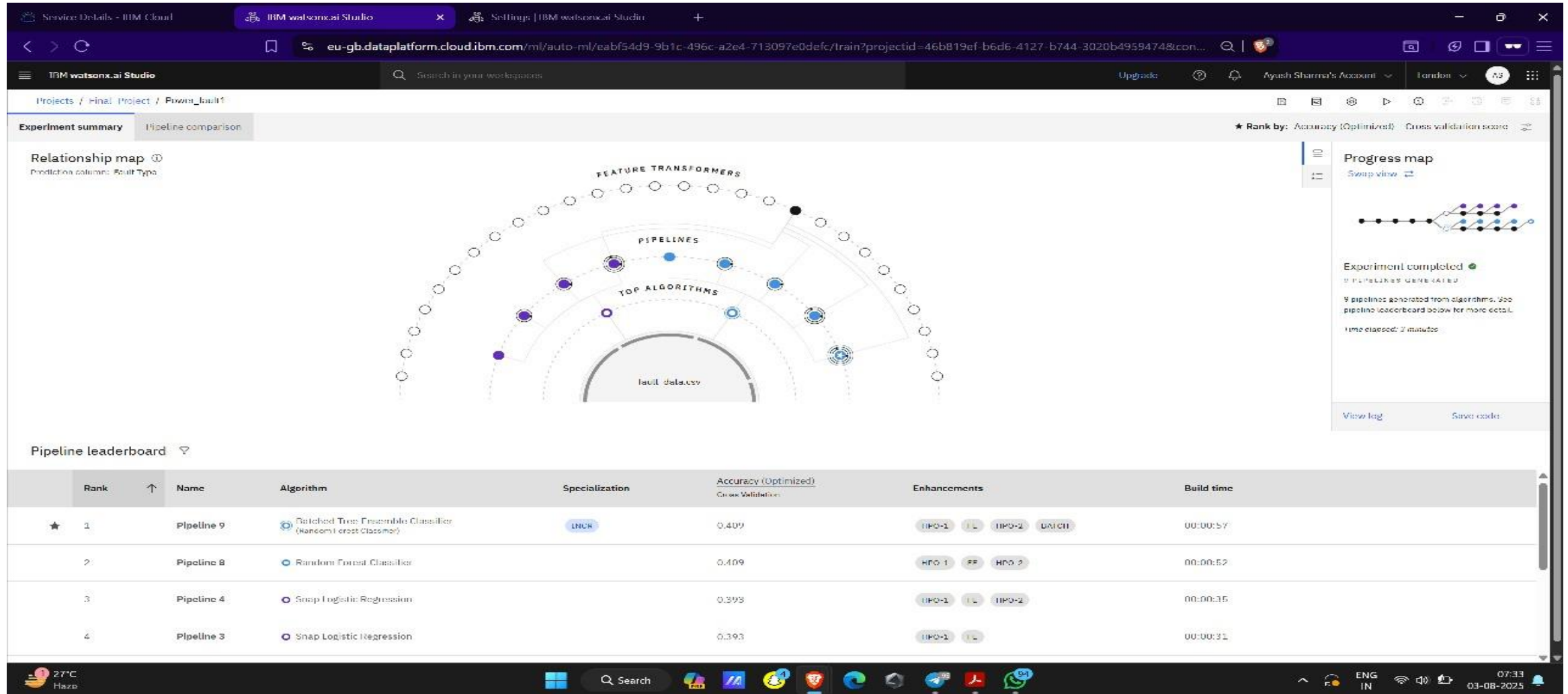
# ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**  
Random Forest Classifier (or SVM based on performance)
- **Data Input:**  
Voltage, current, and phasor measurements from the dataset
- **Training Process:**  
Supervised learning using labeled fault types
- **Prediction Process:**  
Model deployed on IBM Watson Studio with API endpoint for real-time predictions

# RESULT

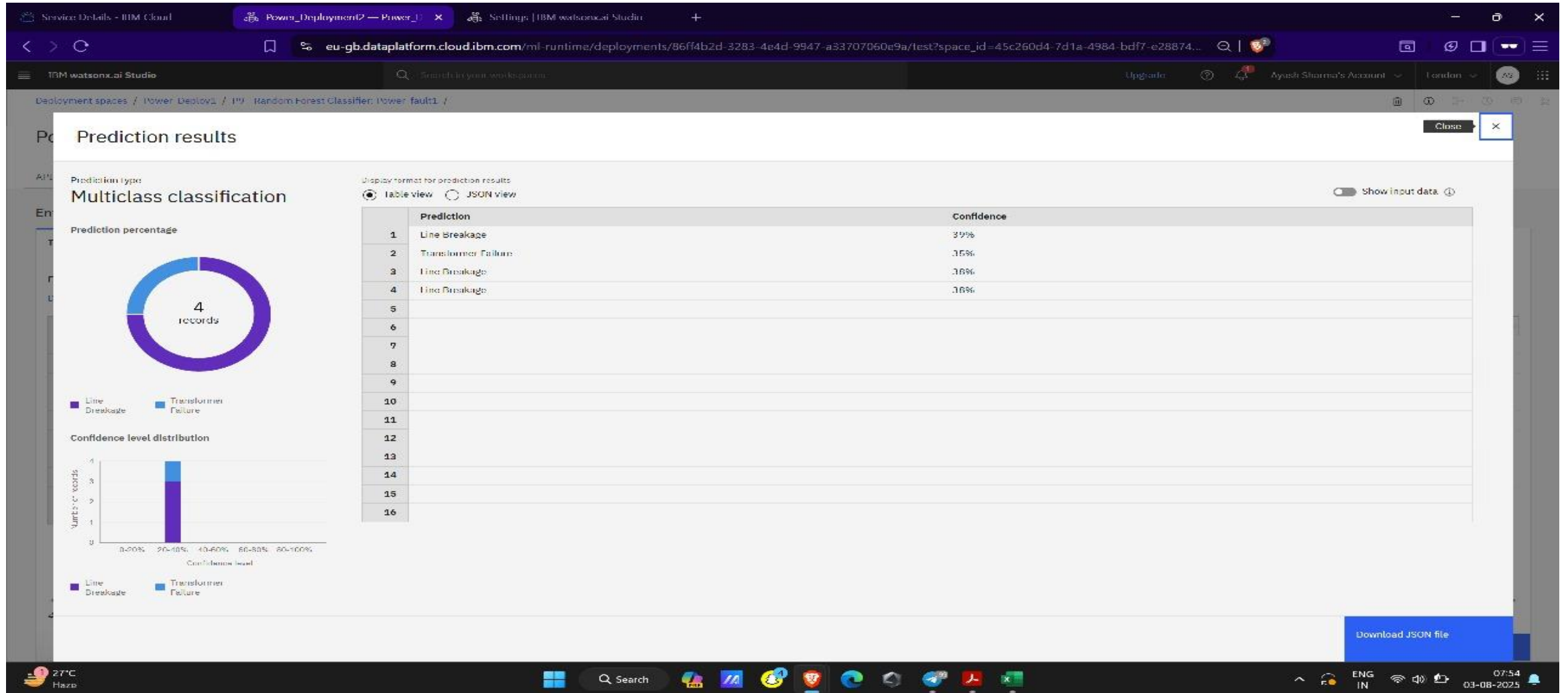


# RESULT





# RESULT



# RESULT

The screenshot displays the IBM Watson AI Studio interface. At the top, the browser address bar shows the URL: `eu-gb.dataplatform.cloud.ibm.com/ml-runtime/deployments/86ff4b2d-3283-4e4d-9947-a33707060e9a/test?space_id=45c260d4-7d1a-4984-bdf7-e28874...`. The page title is "Power\_Deployment2" with a "Deployed" status and a "Test" button.

Below the title, there is a section "Enter input data" with a "Text" tab selected. A "JSON" button is also visible. Below this, there is a prompt: "Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB." and links for "Download CSV template", "Browse local files", and "Search in space".

A table with 11 columns and 10 rows is displayed. The columns are: Fault ID (other), Fault Location (Latitude, Longitude) (other), Voltage (V) (double), Current (A) (double), Power Load (MW) (double), Temperature (°C) (double), Wind Speed (km/h) (double), Weather Condition (other), Maintenance Status (other), and Component Health (other). The first four rows contain data, and the remaining six rows are empty.

At the bottom right of the table area, there is a "Predict" button. Below the table, a status bar indicates "4 rows, 12 columns".

The bottom of the screenshot shows a Windows taskbar with various icons, including the Start button, Search, and several application icons. The system clock shows the date "03-08-2025" and time "07:55".

	Fault ID (other)	Fault Location (Latitude, Longitude) (other)	Voltage (V) (double)	Current (A) (double)	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h) (double)	Weather Condition (other)	Maintenance Status (other)	Component Health (other)
1	F001	(34.0522, 118.2437)	2200	250	50	25	20	Clear	Scheduled	Normal
2	F002	(34.056, -118.245)	1800	100	45	28	15	Rainy	Completed	Faulty
3	F016	(34.7105, -118.5379)	2100	53246	53	38	18	Rainy	Completed	Faulty
4	F010	(34.4192, 118.8254)	2065	199	55	25	21	Clear	Scheduled	Normal
5										
6										
7										
8										
9										
10										

# CONCLUSION

- The machine learning model effectively classifies various power system faults, providing accurate and timely fault detection. This improves the reliability and stability of power grids and assists in automating recovery actions. Future improvements can focus on real-time data integration and advanced ensemble methods.

---

# FUTURE SCOPE

- Integrate real-time sensor data using IoT
- Explore deep learning models for better accuracy
- Expand to classify faults in regional and national grids
- Deploy the model on edge devices in substations

# REFERENCES

- Kaggle Dataset on Power System Faults
- IBM Watson Studio Documentation
- Research papers on fault classification
- Python Libraries: Scikit-learn, Pandas, Matplotlib

# IBM CERTIFICATIONS

In recognition of the commitment to achieve  
professional excellence



## AYUSH SHARMA

Has successfully satisfied the requirements for:

---

### Getting Started with Artificial Intelligence

---



Issued on: Jul 16, 2025  
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/afaa7bf7-e9e9-4532-8373-1772bf508ec0>



# IBM CERTIFICATIONS

In recognition of the commitment to achieve  
professional excellence



## AYUSH SHARMA

Has successfully satisfied the requirements for:

---

### Journey to Cloud: Envisioning Your Solution

---



Issued on: Jul 17, 2025

Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/5571a1a6-8c2f-4c63-8ee8-24b9f2958351>



# IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to

AYUSH SHARMA

for the completion of

**Lab: Retrieval Augmented Generation with  
LangChain**

(ALM-COURSE\_3824998)

According to the Adobe Learning Manager system of record

**Completion date:** 24 Jul 2025 (GMT)

**Learning hours:** 20 mins





**THANK YOU**