CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION USING ML

Presented By:
1. Sahil Yadav - GLA University - CSE

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

Develop a machine learning model aimed at detecting and categorizing faults in a power distribution network.

By leveraging electrical measurements, such as voltage and current phasors, the model should be capable of identifying whether the system is operating normally or experiencing faults like line-to-ground, line-to-line, or three-phase faults. The goal is to enable swift and precise identification of faults, ensuring the stability and reliability of the power grid.

PROPOSED SOLUTION

- The proposed system aims to address the challenge of accurately detecting and classifying different types of faults in a power distribution network. This enhances grid stability, reduces downtime, and allows for proactive maintenance. The solution consists of the following components:
- Data Collection:
  - Gather labeled electrical measurement data (voltage and current phasors) under normal and various fault conditions.
  - Source the dataset from Kaggle and upload it to IBM Cloud Object Storage for secure access.
- Data Preprocessing:
  - Clean the dataset by handling missing values, noise, and irrelevant features.
  - Perform feature engineering to enhance key signal characteristics that help distinguish fault types.
- Machine Learning Algorithm:
  - Implement supervised learning algorithms (e.g., Random Forest, XGBoost, Neural Network) to classify fault types.
  - Train the model to distinguish between – No Fault, Line-to-Ground(LG), Line-toLine(LL), LLG, LLLG.
- Deployment:
  - Use Watsonx.ai Studio to develop, train, and deploy the model in a cloud environment.
  - Store model artifacts and datasets in IBM Cloud Object Storage for secure access and scalability.
- Evaluation:
  - Evaluate the model using metrics like Accuracy, Precision, Recall, and F1-score.
  - Perform cross-validation and fine-tuning to improve performance and reduce false classifications.
- Results :
  - Successfully deployed a fault classification model with high accuracy, enabling real-time fault detection.

SYSTEM APPROACH

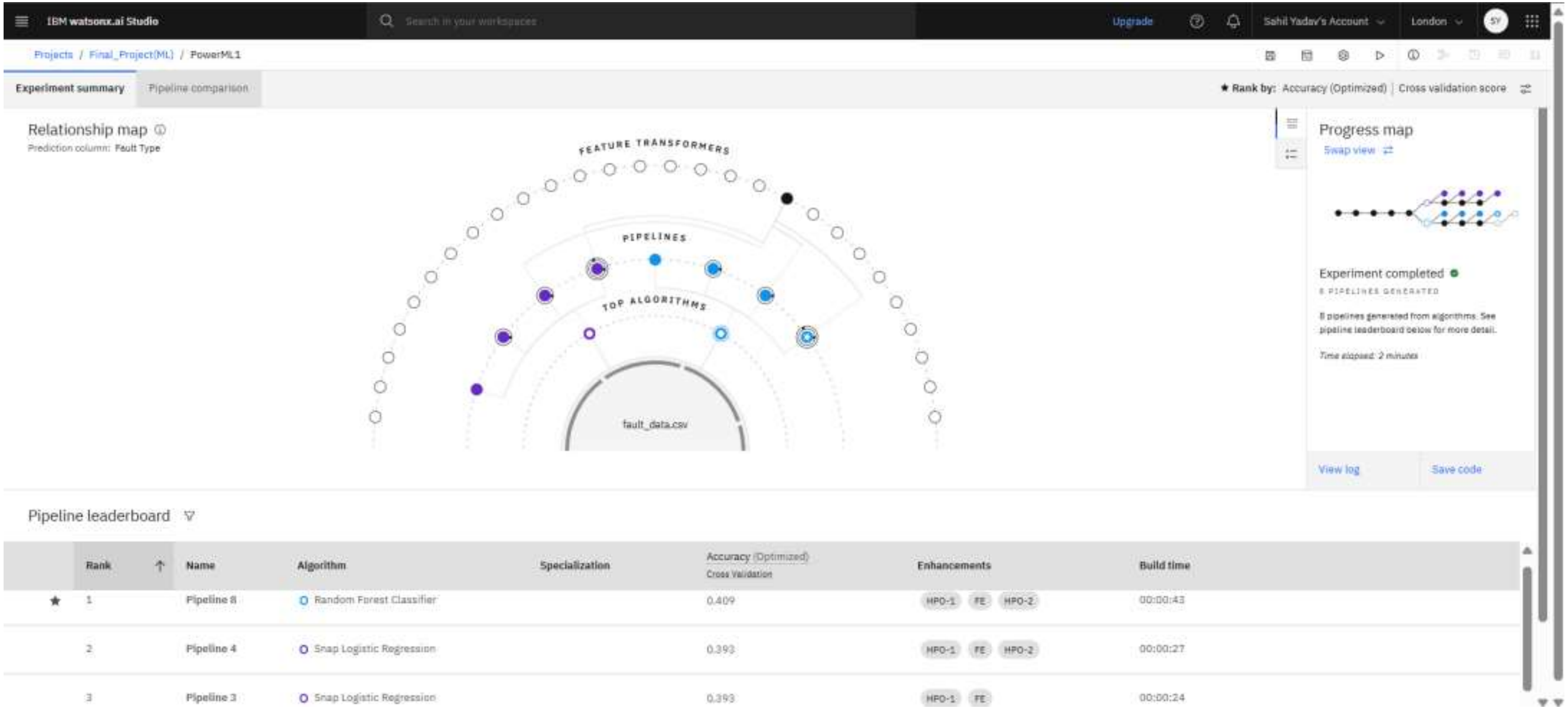
The system is designed to detect and classify faults in a power distribution network for the project: "Power System Fault Detection and Classification", using IBM Cloud services and machine learning

- Data Ingestion
  ⟶ Load power fault dataset from Kaggle into IBM Cloud Object Storage.
- Data Preprocessing
  ⟶ Clean and normalize voltage & current data, handle missing values.
- Feature Engineering
  ⟶ Select and extract relevant electrical features for better fault classification.
- Model Building
  ⟶ Use Watsonx.ai Studio to train ML models (Random Forest, XGBoost, etc.).
- Model Evaluation
  ⟶ Validate using accuracy, precision, recall, F1-score, and confusion matrix.
- Deployment
  ⟶ Deploy the trained model using Watsonx.ai for real-time predictions.
- Monitoring
  ⟶ Continuously monitor model performance and retrain when needed.

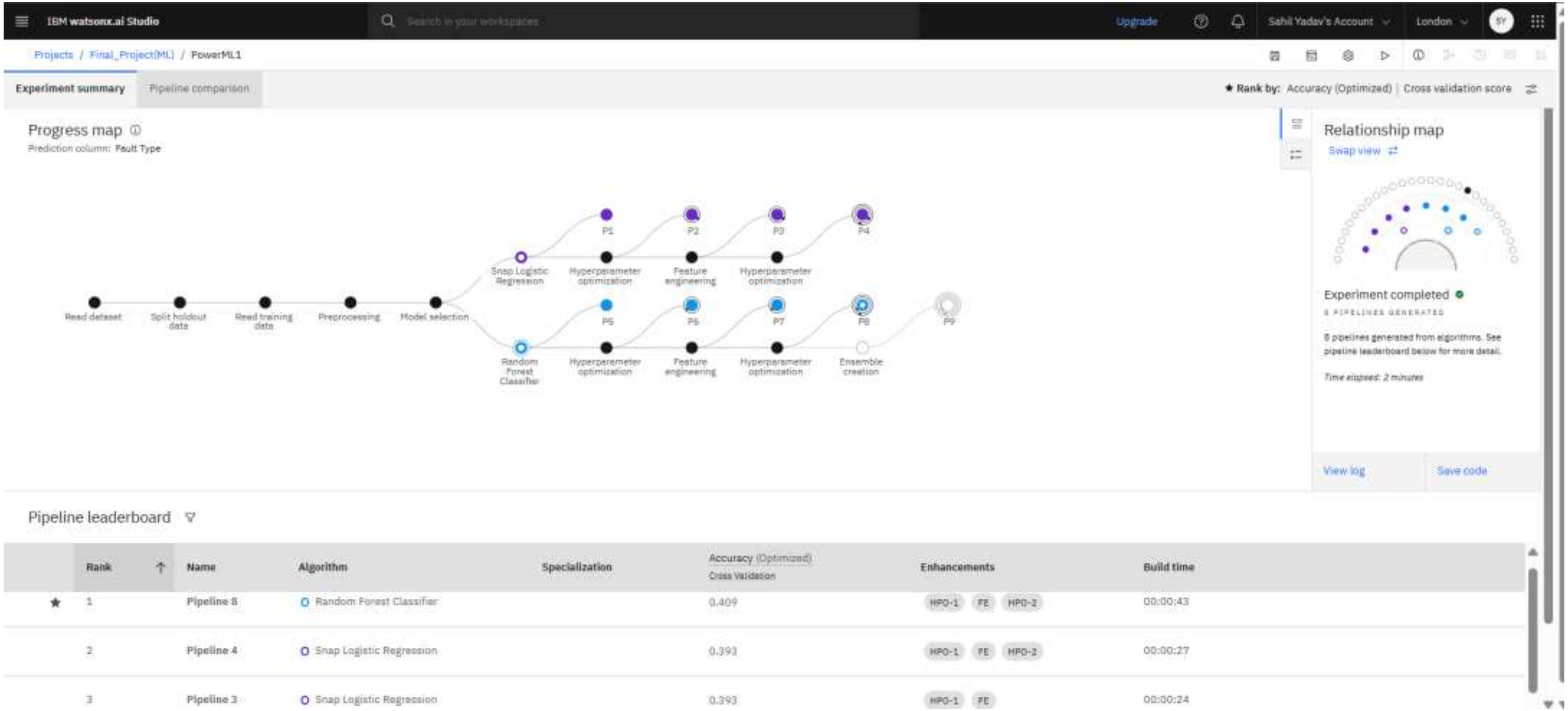
ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for predicting power system fault types. Here's an example structure for this section:
- **Algorithm Selection:**
  Used **Random Forest** and **XGBoost** for their high accuracy, robustness, and ability to handle multi-class fault classification effectively.
- **Data Input:**
  Voltage and current phasors from power system measurements under various fault and normal conditions.
- **Training Process:**
  Trained on a labeled Kaggle dataset using data split, **cross-validation**, and **hyperparameter tuning.** Preprocessing included scaling and encoding.
- **Prediction Process:**
  The model predicts fault type (LG, LL, LLG, LLLG, or No Fault) in real-time from new voltage/current inputs, enabling fast and accurate detection.

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Sahil Yadav's Account
London
Projects / Final_Project(ML) / PowerML1
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Relationship map
Prediction column: Fault Type
FEATURE TRANSFORMERS
PIPELINES
TOP ALGORITHMS
fault_data.csv
Progress map
Swap view
Experiment completed
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 2 minutes
View log
Save code
Pipeline leaderboard
Rank
Name
Algorithm
Specialization
Accuracy (Optimized)
Cross Validation
Enhancements
Build time
1 Pipeline 8 Random Forest Classifier 0.409 HPO-1 FE HPO-2 00:00:43
2 Pipeline 4 Snap Logistic Regression 0.393 HPO-1 FE HPO-2 00:00:27
3 Pipeline 3 Snap Logistic Regression 0.393 HPO-1 FE 00:00:24

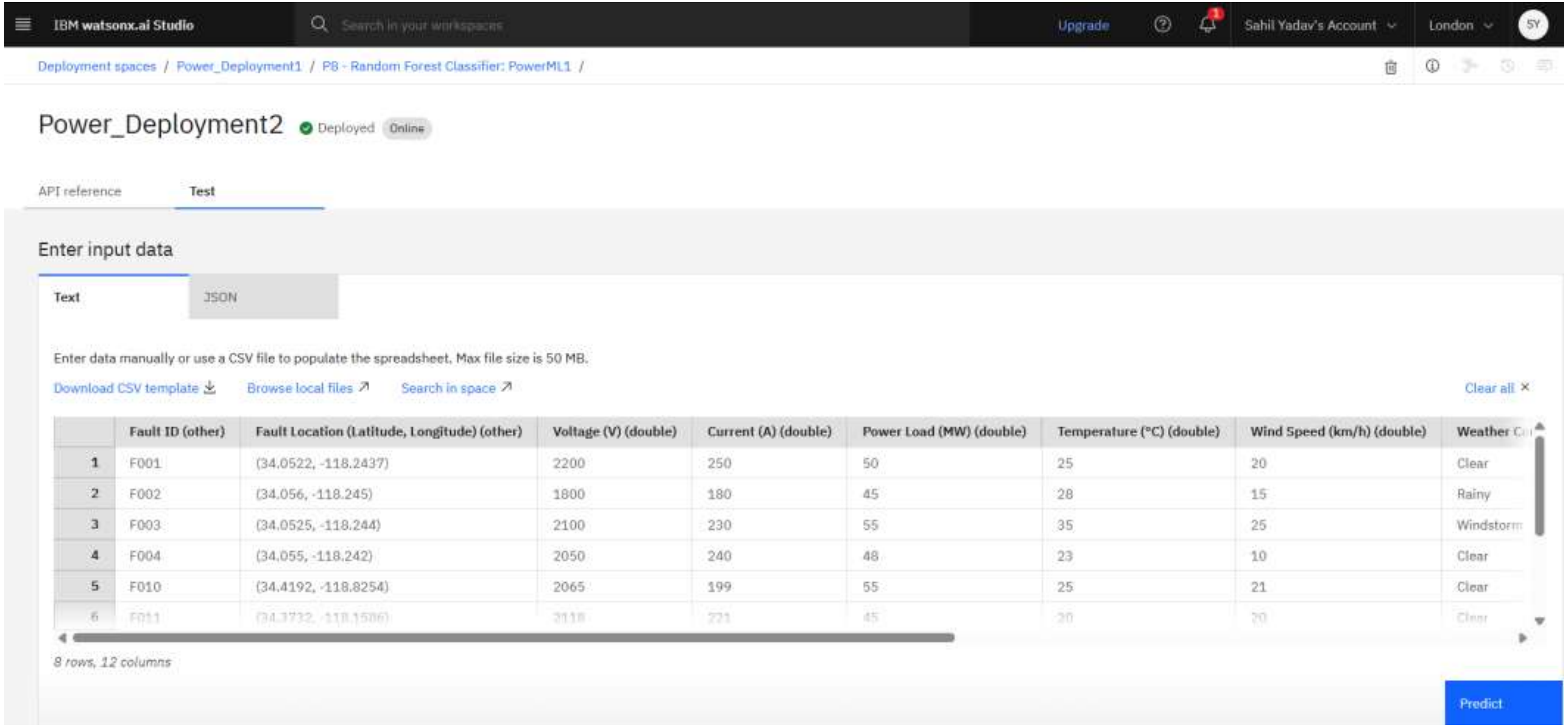
RESULT

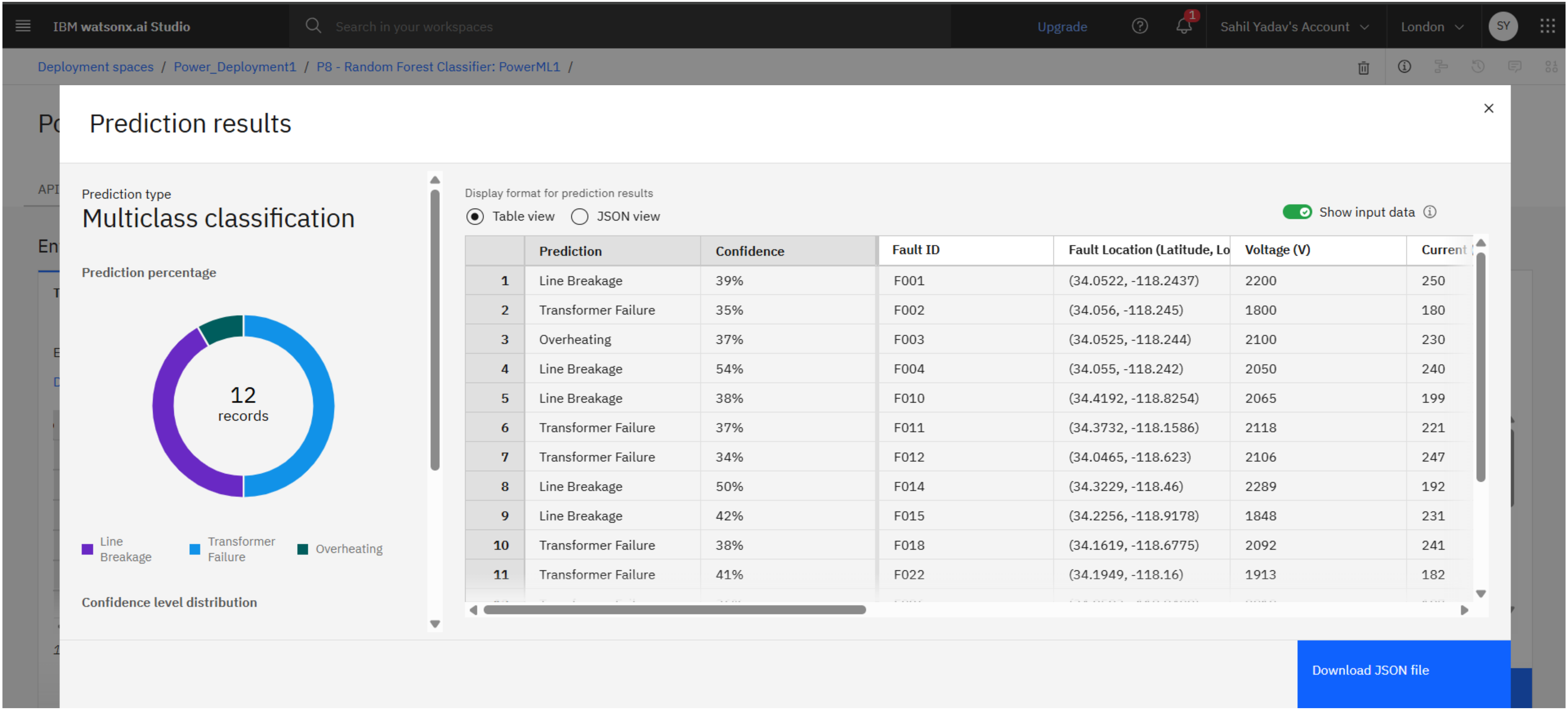


Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★ 1	Pipeline 8	Random Forest Classifier		0.409	HPO-1 FE HPO-2	00:00:43
2	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:00:27
3	Pipeline 3	Snap Logistic Regression		0.393	HPO-1 FE	00:00:24

edunet foundation

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Sahil Yadav's Account
London
SY
Deployment spaces / Power_Deployment1 / P8 - Random Forest Classifier: PowerML1 /
Power_Deployment2
Deployed
Online
API reference
Test
Enter input data
Text
JSON
Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.
Download CSV template
Browse local files
Search in space
Clear all
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 C
1 F001 (34.0522, -118.2437) 2200 250 50 25 20 Clear
2 F002 (34.056, -118.245) 1800 180 45 28 15 Rainy
3 F003 (34.0525, -118.244) 2100 230 55 35 25 Windstorm
4 F004 (34.055, -118.242) 2050 240 48 23 10 Clear
5 F010 (34.4192, -118.8254) 2065 199 55 25 21 Clear
8 rows, 12 columns
Predict



	Prediction	Confidence	Fault ID	Fault Location (Latitude, Lo	Voltage (V)	Current
1	Line Breakage	39%	F001	(34.0522, -118.2437)	2200	250
2	Transformer Failure	35%	F002	(34.056, -118.245)	1800	180
3	Overheating	37%	F003	(34.0525, -118.244)	2100	230
4	Line Breakage	54%	F004	(34.055, -118.242)	2050	240
5	Line Breakage	38%	F010	(34.4192, -118.8254)	2065	199
6	Transformer Failure	37%	F011	(34.3732, -118.1586)	2118	221
7	Transformer Failure	34%	F012	(34.0465, -118.623)	2106	247
8	Line Breakage	50%	F014	(34.3229, -118.46)	2289	192
9	Line Breakage	42%	F015	(34.2256, -118.9178)	1848	231
10	Transformer Failure	38%	F018	(34.1619, -118.6775)	2092	241
11	Transformer Failure	41%	F022	(34.1949, -118.16)	1913	182

CONCLUSION

- In this project, I successfully developed a **machine learning model** capable of accurately detecting and classifying various **power system faults** using voltage and current phasor data.
- We used a **realistic dataset** from Kaggle and integrated it with **IBM Cloud Object Storage** and **Watsonx.ai Studio** to create a scalable and cloud-based solution.
- The model was trained to classify fault types such as:

  - Line-to-Ground (LG)

  - Line-to-Line(LL)

  - Three-Phase Faults(LLLG)

FUTURE SCOPE

- The proposed system has significant potential for future development in real-time power system monitoring. Integrating IoT-enabled sensors can allow continuous, live data collection for immediate fault detection. Deploying the model on edge devices will ensure faster response times without depending on cloud connectivity. The system can evolve into a part of self-healing smart grids by enabling automatic fault isolation and restoration. Future enhancements may include the use of deep learning models like LSTM or CNN to detect complex fault behaviors. Additionally, training on diverse regional datasets can improve generalization, making the solution scalable and applicable to varied power grid infrastructures.

REFERENCES

- Kaggle Dataset – *Power System Faults Dataset*
  https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset
- IBM Cloud Documentation
  https://cloud.ibm.com/docs
- IBM Watsonx.ai Studio
  https://dataplatform.cloud.ibm.com

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

Sahil Yadav

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 24, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/57f9651b-10bf-4321-9df2-8e268e47eb5a

IBM

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM

Sahil Yadav

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 24, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/4f9caf26-2531-4d7c-a4a9-b9855f5ebb2c

IBM

edunet foundation

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

SAHIL YADAV

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 25 Jul 2025 (GMT) **Learning hours:** 20 mins

THANK YOU