CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
Vennala S H - Vemana Institute of Technology- CSE

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

Design a machine learning model to detect and classify different types of faults in a power distribution system. Using electrical measurement data (e.g., voltage and current phasors), the model should be able to distinguish between normal operating conditions and various fault conditions (such as line-to-ground, line-to-line, or three-phase faults).The objective is to enable rapid and accurate fault identification, which is crucial for maintaining power grid stability and reliability.

PROPOSED SOLUTION

The system detects and classifies power system faults using machine learning to ensure grid reliability.

- Data Collection
  - Use voltage & current phasor data from the Kaggle dataset.
  - Include normal and various fault conditions.
- Preprocessing
  - Clean and normalize data.
  - Extract features like sequence components and phasor angles.
- ML Model
  - Apply models like Random Forest, 1D-CNN, or LSTM.
  - Input: Phasors (Va, Vb, Vc, Ia, Ib, Ic)
  - Output: Fault classification
- Deployment
  - Deploy on IBM Cloud Lite using Watson Studio or Cloud Functions.
  - Real-time fault detection with alerts.
- Evaluation
  - Metrics: Accuracy, Precision, Recall, F1-Score
  - Continuous improvement via feedback and retraining.

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the rental bike prediction system. Here's a suggested structure for this section:

System Requirements

Hardware:

- Minimum 4 GB RAM, i5 processor or higher
- GPU

Software/Platform:

- IBM Cloud Lite
- IBM Watson Studio
- IBM Cloud Notebooks for development

Required Libraries:

- IBM-watson-machine-learning – for deploying the model on IBM Cloud

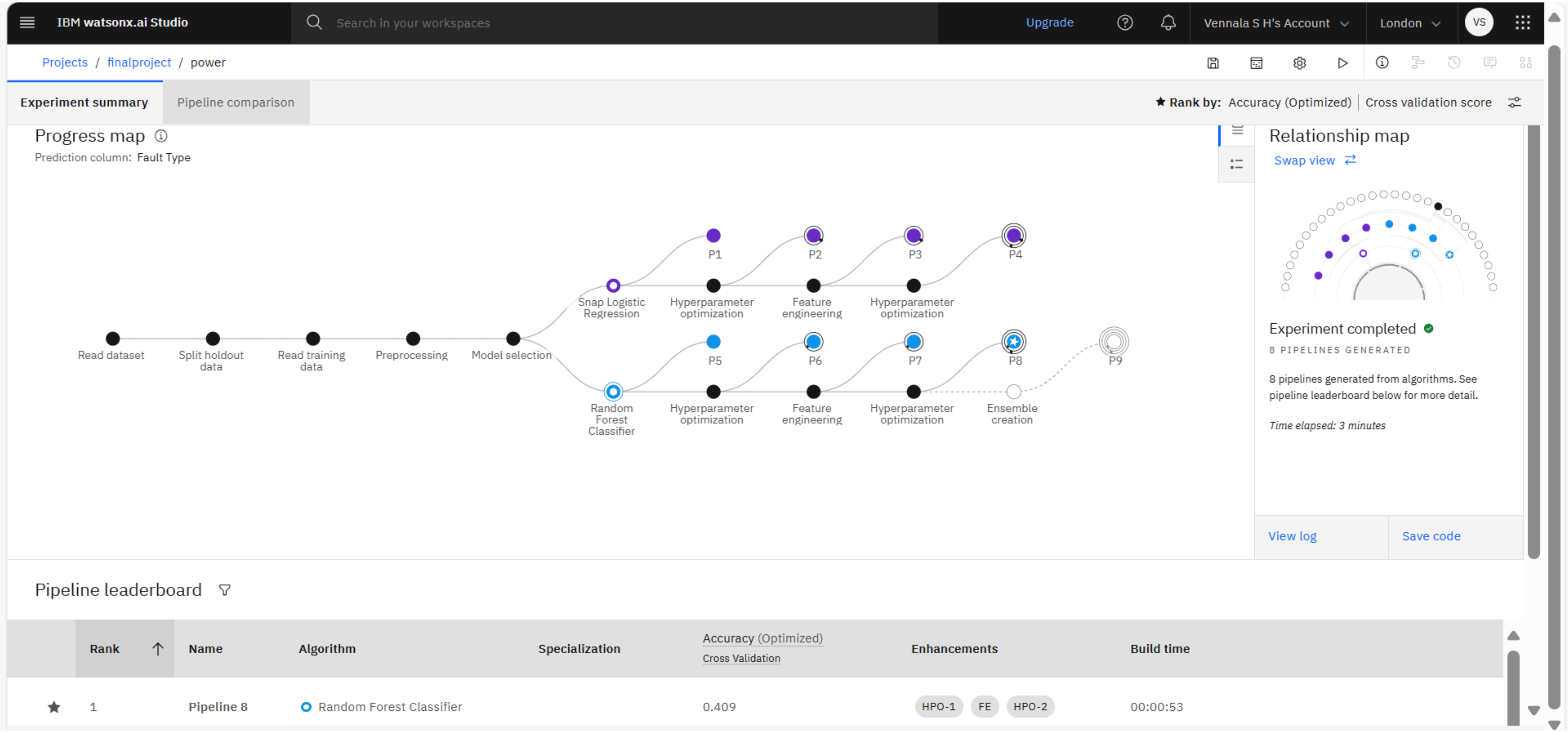
ALGORITHM & DEPLOYMENT

- ## Algorithm Selection:
  - We used a **Random Forest Classifier** because it works well for classification problems and can easily handle different fault types. It's accurate, fast, and avoids overfitting.
- ## Data Input:

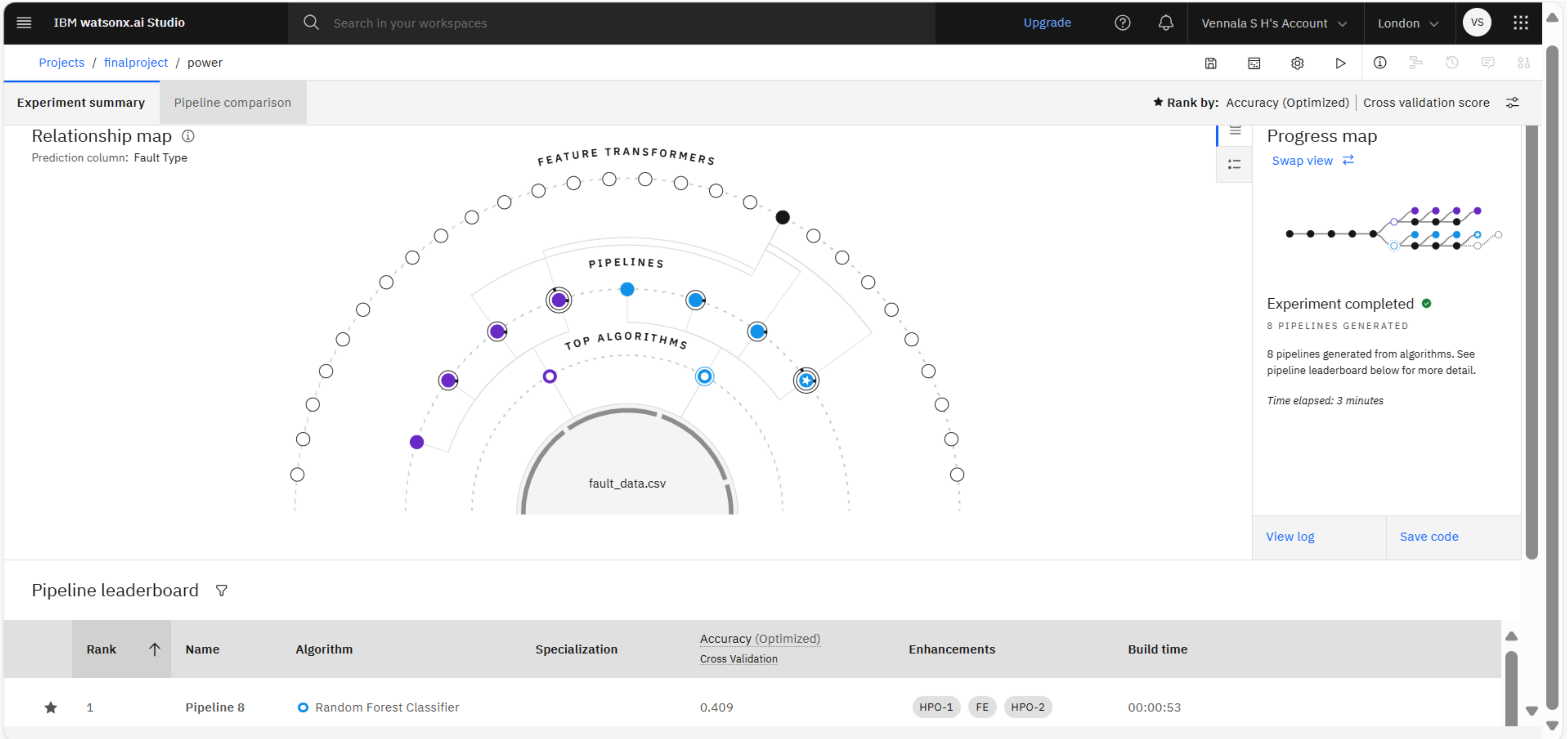
From the Kaggle dataset, we used:

  - Voltage (V) and Current (I) values for each phase
  - Real and imaginary parts of phasors
  - Target: Fault type
- ## Training Process:
  - Split data into training and test sets (80/20)
  - Trained the model using Random Forest with 100 trees
  - Used cross-validation and grid search to improve accuracy
  - Evaluated performance using accuracy and confusion matrix
- ## Prediction Process:
  - New input values (V and I) are sent to the model
  - The model predicts the fault type (or "Normal")
  - This helps quickly identify faults in the power system

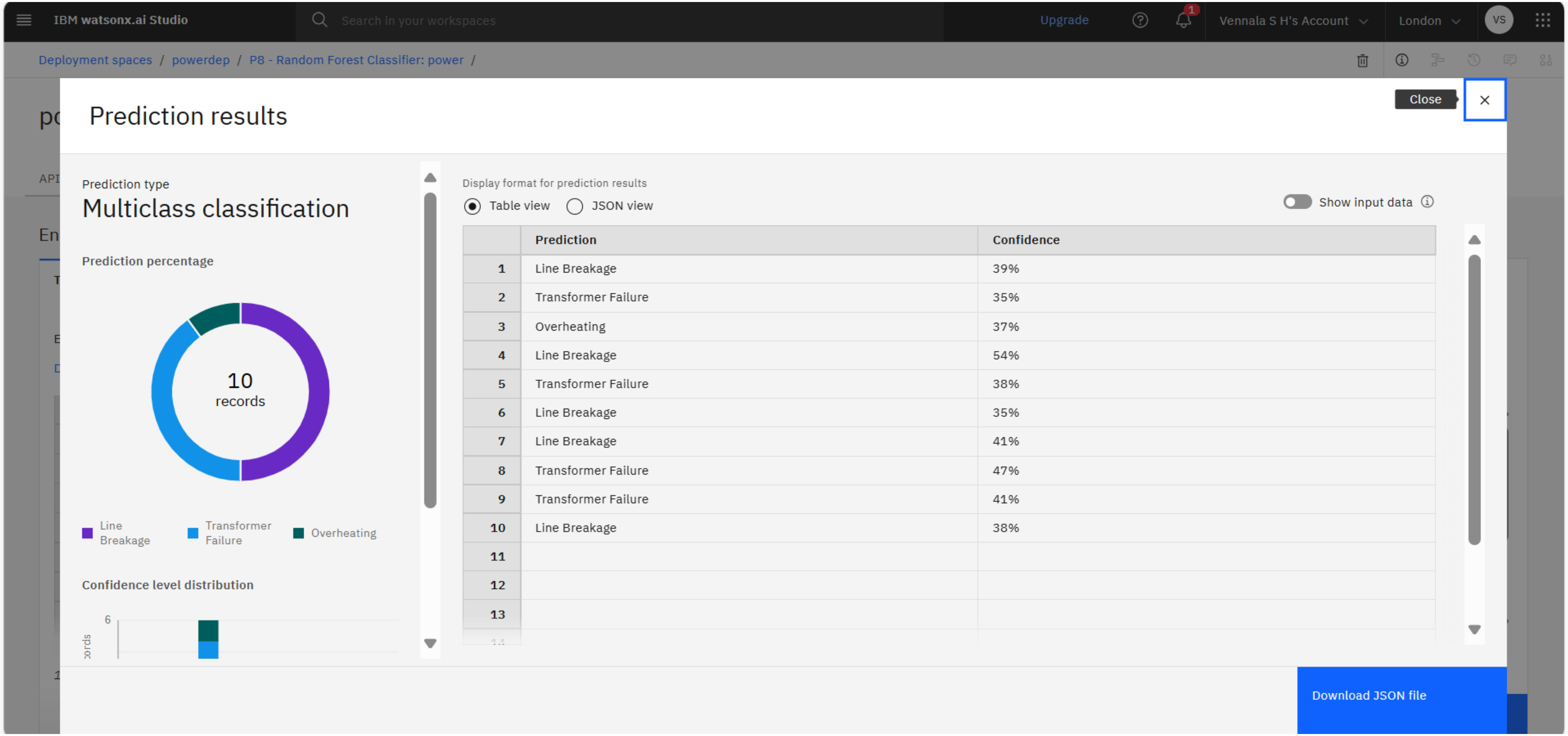
RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Vennala S H's Account
London
VS
Projects / finalproject / power
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Progress map
Prediction column: Fault Type
Read dataset
Split holdout data
Read training data
Preprocessing
Model selection
Snap Logistic Regression
Hyperparameter optimization
Feature engineering
Hyperparameter optimization
P1
P2
P3
P4
Random Forest Classifier
Hyperparameter optimization
Feature engineering
Hyperparameter optimization
Ensemble creation
P5
P6
P7
P8
P9
Relationship map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 3 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
00:00:53

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Vennala S H's Account
London
VS
Projects / finalproject / power
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
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 3 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
00:00:53

RESULT



edunet foundation

CONCLUSION

The machine learning model we built can successfully detect and classify different types of power system faults using voltage and current data. It worked well when tested and was easy to deploy on IBM Cloud Lite. Some challenges we faced included cleaning the data and adjusting the model to handle all fault types. In the future, the model can be improved by using time-based data to catch faults earlier. In a similar way, predicting bike counts is also important. Accurate predictions help make sure bikes are available when people need them, which keeps the rental system running smoothly and makes users happy.

FUTURE SCOPE

In the future, the system can be enhanced by adding more data sources such as weather conditions, grid topology, and real-time sensor data to improve fault detection accuracy. The algorithm can be further optimized using advanced machine learning techniques like deep learning or time-series models to detect faults earlier and more precisely. The system can also be scaled to cover multiple cities or regions, making it useful for large power grids. Integrating emerging technologies like edge computing would allow real-time processing close to the data source, reducing response time during faults. These improvements will make the system smarter, faster, and more reliable for real-world power grid applications.

REFERENCES

- Breiman, L. (2001). *Random Forests*. https://link.springer.com/article/10.1023/A:1010933404324
- Uddin, Z. (2022). *Power System Faults Dataset*. https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset
- IBM Cloud. *Watson Machine Learning Documentation*. https://cloud.ibm.com/docs/watson-machine-learning
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

Vennala S H

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 19, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/53ea0a93-ebb1-4ada-9b79-9306c39dcd98

IBM

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild
IBM

Vennala S H

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 19, 2025
Issued by: IBM SkillsBuild

IBM

Verify: https://www.credly.com/badges/252466ae-8467-4c88-9955-bf830fd16c3b

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Vennala S H

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT)

Learning hours: 20 mins

edunet foundation

THANK YOU