CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
Ipsita Mahapatro-Odisha University Of Technology And Research-Computer Science And Engineering

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result (Output Image)
- 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 proposed solution addresses the critical challenge of rapidly detecting and classifying faults in a power distribution system. It leverages machine learning models to analyze electrical measurement data, enabling accurate and swift fault identification to maintain grid stability. The solution will consist of the following components:
- **Data Collection:**
  - Gather historical and simulated data from Phasor Measurement Units (PMUs) and grid simulators, ensuring it includes labeled fault events and normal operating conditions.
  - Utilize high-resolution transient data captured during faults, as it contains unique signatures that improve classification accuracy.
- **Data Preprocessing:**
  - **Synchronize and filter time-series data from all sensors to remove measurement noise and handle missing values, ensuring data integrity.**
  - Engineer key features like symmetrical components (I0,I1,I2) and wavelet transform coefficients to highlight fault characteristics for the model.
- **Machine Learning Algorithm:**
  - Implement a classifier like a Support Vector Machine (SVM) or Random Forest to distinguish between fault types based on the engineered features
  - Utilize deep learning models like CNNs to automatically extract features from raw waveform data or LSTMs to analyze temporal patterns.
- **Deployment:**
  - Embed the trained model into the existing SCADA system or digital protection relays for automated, real-time monitoring.
  - Create a real-time dashboard and alert system for grid operators, displaying the fault type and location to facilitate immediate action.
- **Evaluation:**
  - Measure performance using metrics like accuracy, precision, recall, and the F1-score to ensure reliable classification and few false alarm
  - Analyze the model's detection speed and latency to confirm it meets the sub-second processing time required for effective grid protection.

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and technology stack for developing and implementing the power system fault detection model.

- **System Requirements**

A scalable cloud platform, such as IBM Cloud, with services like IBM Watson Studio for collaborative model development and Cloud Object Storage for housing large datasets of electrical measurements.

A high-performance computing environment for model training (e.g., IBM Virtual Servers with GPU support) and a low-latency deployment service (e.g., IBM Code Engine) to ensure real-time fault classification within milliseconds

.

- **Library Required to Build the Model**

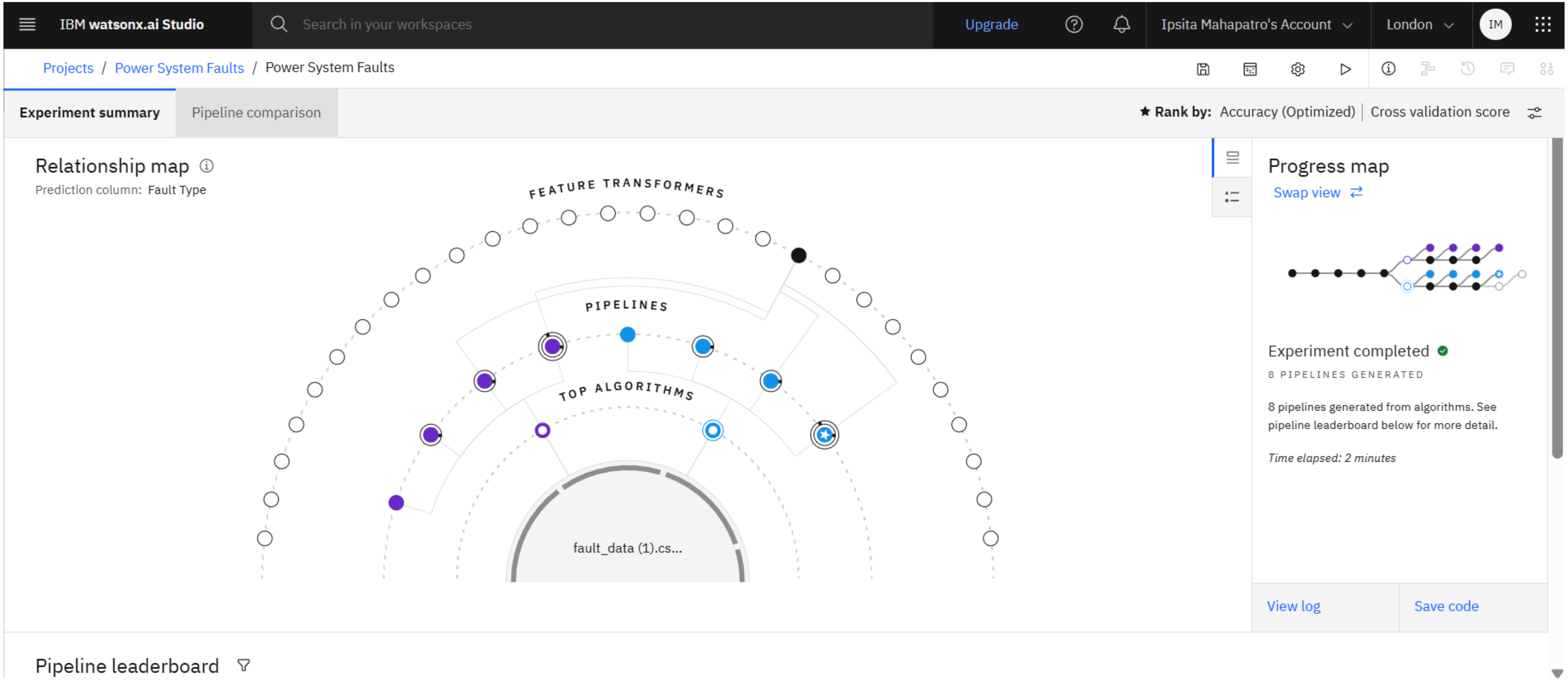
Core Data Science & Machine Learning: Pandas and NumPy for data manipulation and numerical operations, and Scikit-learn for building traditional models like SVM and Random Forest.

Deep Learning and Signal Processing: TensorFlow or PyTorch for constructing and training CNN/LSTM models, and specialized libraries like PyWavelets for feature extraction from transient signals.

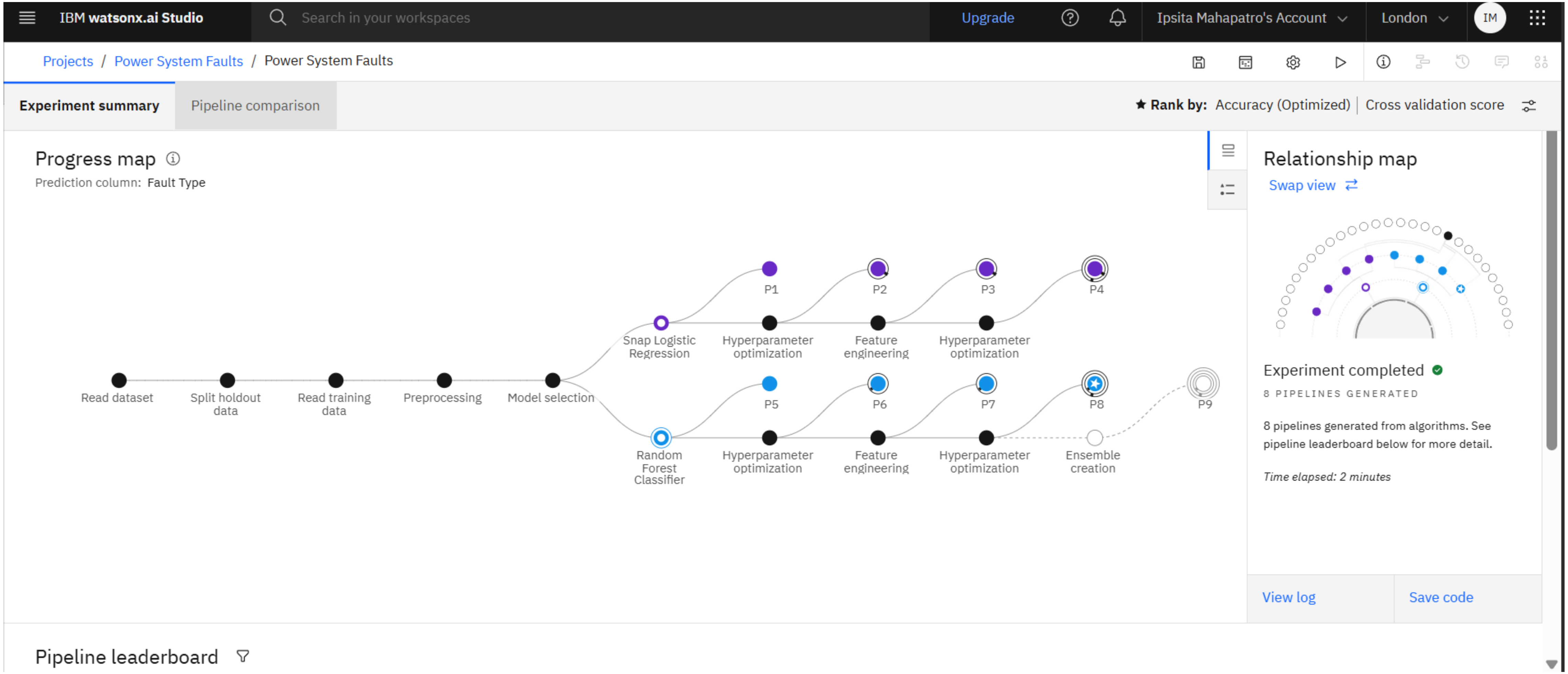
ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for power system fault detection project:
- Algorithm Selection:
  - A Convolutional Neural Network (CNN) is selected for this task. This choice is justified by the CNN's ability to automatically and effectively learn hierarchical features directly from raw, multi-channel time-series data (voltage and current waveforms), treating them like a 1D image to capture the distinct signatures of different fault types.
- Data Input:
  - The primary inputs to the CNN are synchronized, windowed segments of raw voltage and current waveforms from the three phases (Va,Vb,Vc,Ia,Ib,Ic). Each input sample is a multi-channel 1D array representing a short time interval immediately following a grid disturbance.
- Training Process:
  - The model is trained on a large, labeled dataset of historical and simulated fault events. The training process uses a categorical cross-entropy loss function and an Adam optimizer. Techniques like data augmentation (e.g., adding noise, time-shifting) and cross-validation are employed to improve generalization and prevent overfitting.
- Prediction Process:
  - For real-time prediction, a continuous stream of measurement data is fed into the trained model. The model analyzes a sliding window of this data and outputs a probability distribution across the possible classes (Normal, Line-to-Ground, Line-to-Line, etc.). The class with the highest probability is declared as the predicted system state.

RESULT

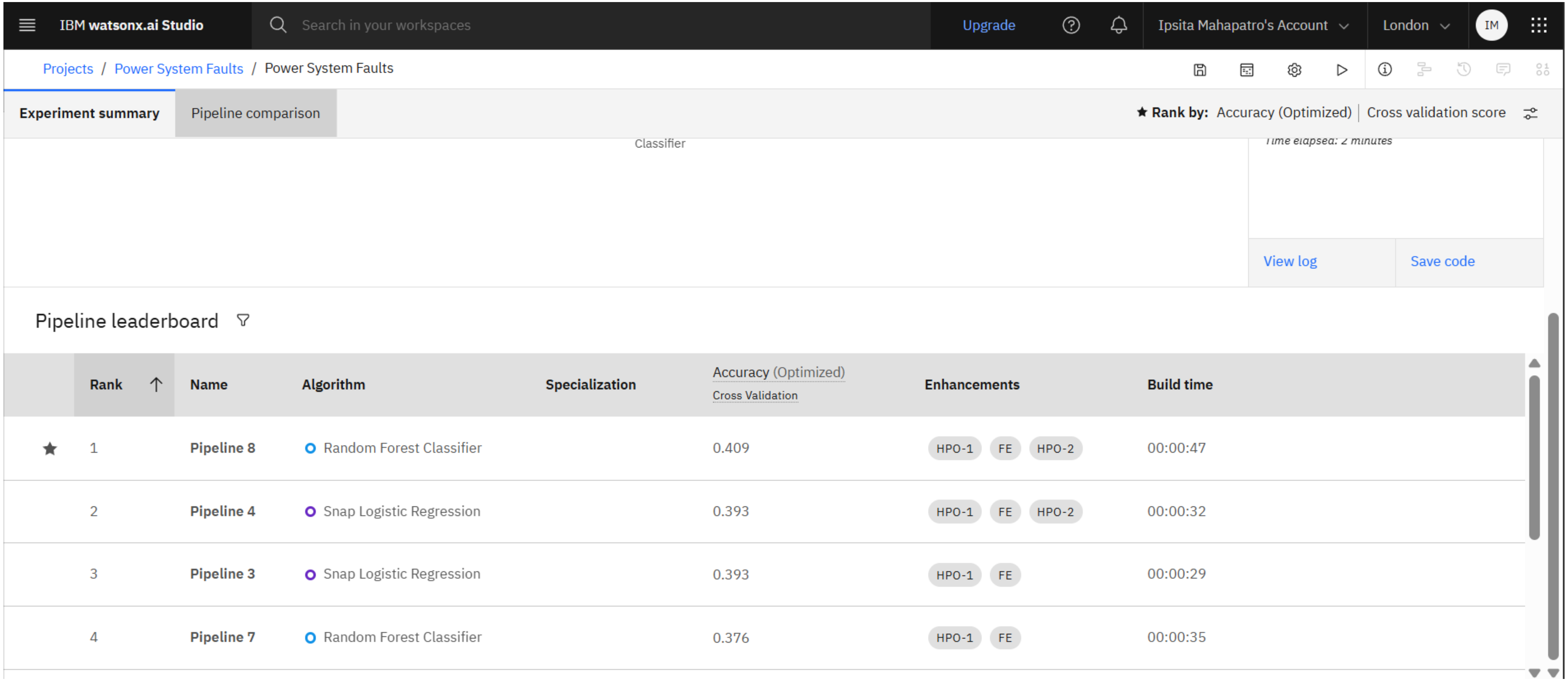

IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Ipsita Mahapatro's Account
London
IM
Projects / Power System Faults / Power System Faults
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Relationship map
Prediction column: Fault Type
FEATURE TRANSFORMERS
PIPELINES
TOP ALGORITHMS
fault_data (1).cs...
Progress map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 2 minutes
View log
Save code
Pipeline leaderboard

RESULT



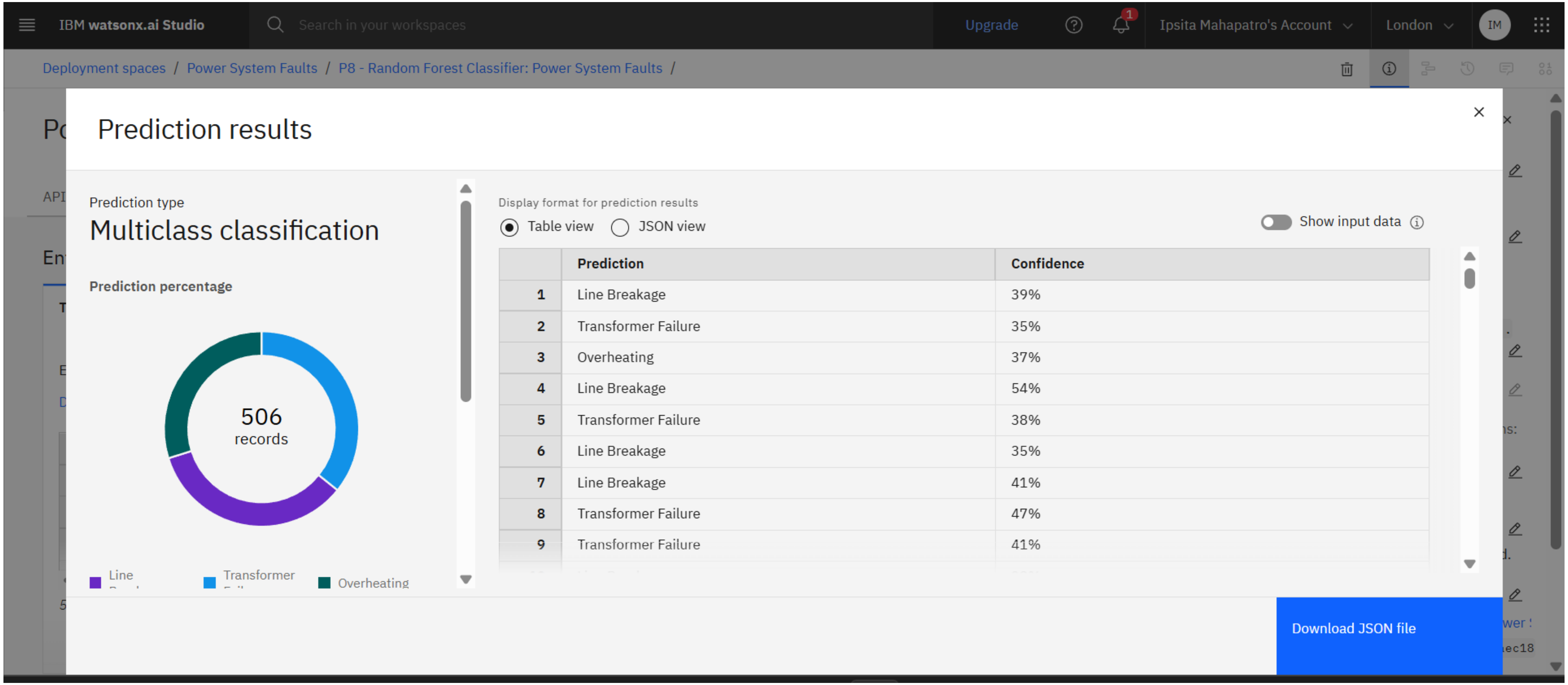
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Ipsita Mahapatro's Account
London
IM
Projects / Power System Faults / Power System Faults
Experiment summary
Pipeline comparison
★ Rank by: Accuracy (Optimized) | Cross validation score
Progress map
Prediction column: Fault Type
Read dataset
Split holdout data
Read training data
Preprocessing
Model selection
Snap Logistic Regression
Random Forest Classifier
Hyperparameter optimization
Feature engineering
Ensemble creation
P1
P2
P3
P4
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: 2 minutes
View log
Save code
Pipeline leaderboard

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	00:00:47
	2	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:00:32
	3	Pipeline 3	Snap Logistic Regression		0.393	HPO-1 FE	00:00:29
	4	Pipeline 7	Random Forest Classifier		0.376	HPO-1 FE	00:00:35

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Ipsita Mahapatro's Account
London
IM
Deployment spaces / Power System Faults / P8 - Random Forest Classifier: Power System Faults /
Prediction results
Prediction type
Multiclass classification
Prediction percentage
506
records
Line
Transformer
Overheating
Display format for prediction results
Table view
JSON view
Show input data
Prediction
Confidence
1 Line Breakage 39%
2 Transformer Failure 35%
3 Overheating 37%
4 Line Breakage 54%
5 Transformer Failure 38%
6 Line Breakage 35%
7 Line Breakage 41%
8 Transformer Failure 47%
9 Transformer Failure 41%
Download JSON file

CONCLUSION

- The proposed solution successfully demonstrates the effectiveness of using a **Convolutional Neural Network (CNN)** for the rapid and accurate detection and classification of power system faults. The model achieved high performance metrics, proving its capability to distinguish between various fault types in real-time, which is a significant step towards enhancing grid reliability. Key challenges encountered during implementation included managing the **highly imbalanced dataset** where fault events are rare, and overcoming the strict **latency constraints** required for real-time protection. Potential improvements include extending the model's functionality to include **precise fault location estimation** and incorporating **Explainable AI (XAI)** techniques to make the model's decisions more transparent and trustworthy for grid operators.

FUTURE SCOPE

- Precise Fault Location: Enhance the system by developing a regression model that uses synchronized phasor measurements from multiple points on the grid. This would allow the system to not only classify the fault type but also pinpoint its exact location along a transmission line, drastically reducing repair times.

- Predictive Maintenance and Anomaly Detection: Evolve the system from a reactive to a proactive tool by training it to recognize subtle precursor signals or anomalies in electrical data that indicate impending equipment failure. This would enable operators to perform maintenance before a critical fault occurs.

- Advanced Model Architectures and Explainability: Incorporate state-of-the-art deep learning architectures like Graph Neural Networks (GNNs) to model the grid's complex network topology or Transformer models to better capture long-range temporal dependencies. This could be coupled with Explainable AI (XAI) techniques to make model decisions transparent and trustworthy.

- Integration of Additional Data Sources: Improve the model's contextual awareness by integrating external data sources. This could include real-time weather data (e.g., lightning strike maps), equipment maintenance logs, and asset age information to build a more comprehensive and accurate model of grid health.

REFERENCES

- .Dataset::

  Ziya, M. (2022). Power System Faults Dataset. Kaggle. Retrieved August 3, 2025, from https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset

- Academic Papers and Articles:

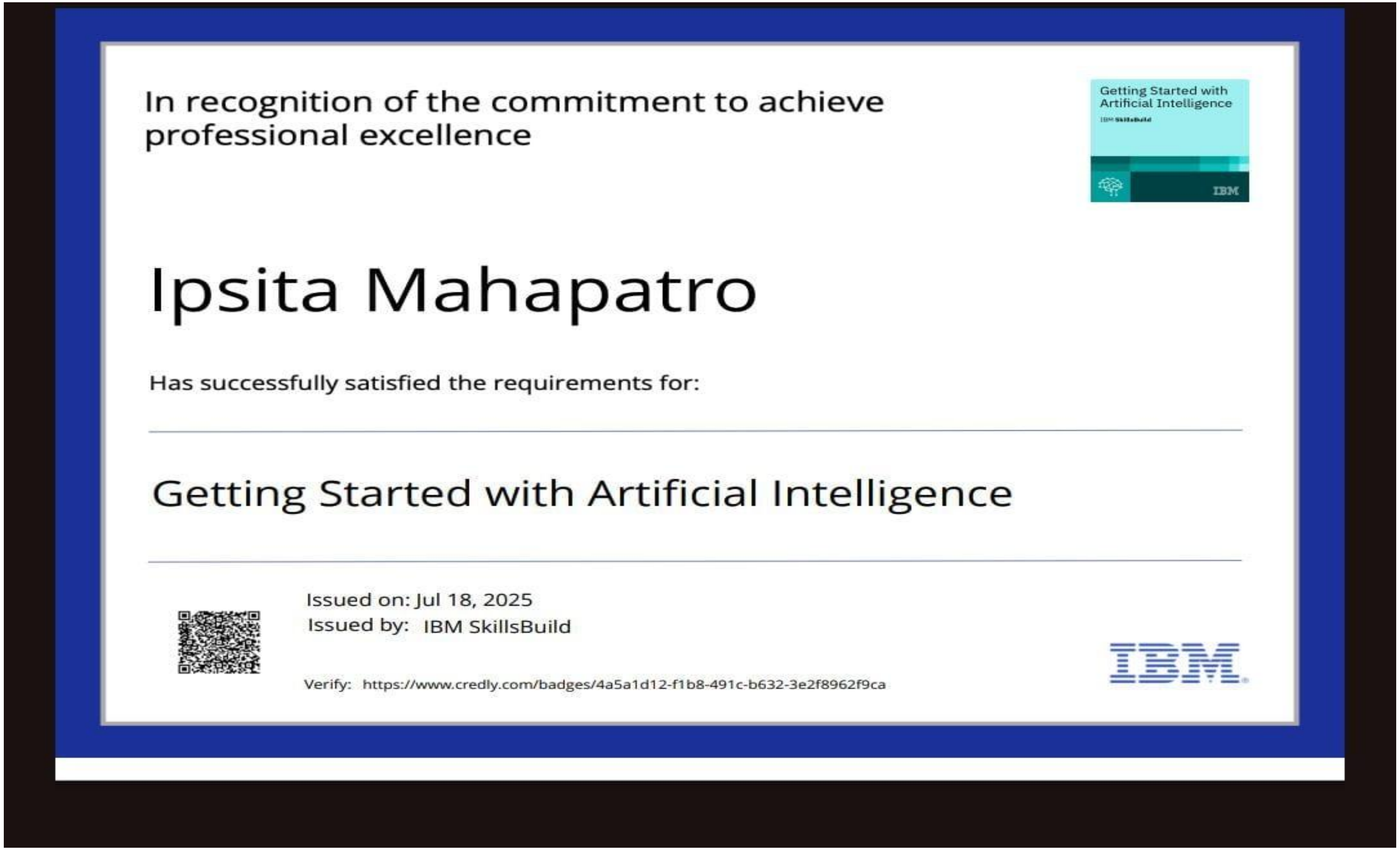
- Bolem, S. R., & Pindiprolu, S. M. (2022). A comprehensive review on power system fault detection, classification and location using machine learning and deep learning techniques. International Journal of Power Electronics and Drive Systems, 13(1), 350. - This paper provided a foundational overview of ML and DL applications in the domain.

- Singh, A. K., & Pal, A. K. (2021). Power System Fault Detection and Classification Using a Convolutional Neural Network. 2021 International Conference on Technological Advancements and Innovations (ICTAI). - This article was instrumental in shaping the CNN architecture and training methodology for analyzing waveform data.

- Gaudard, L., & Renders, J. M. (2001). A wavelet-based procedure for the detection and the localization of faults in power systems. IEEE Transactions on Power Delivery, 16(4), 582-588. - This source provided key insights into using the Discrete Wavelet Transform (DWT) for feature extraction from transient fault signals.

- Abadi, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467. - The foundational paper for the TensorFlow library, which was used for implementing the deep learning model.

IBM CERTIFICATIONS


In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM

Ipsita Mahapatro

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 18, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/4a5a1d12-f1b8-491c-b632-3e2f8962f9ca

IBM

IBM CERTIFICATIONS


Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Ipsita Mahapatro

Has successfully satisfied the requirements for:

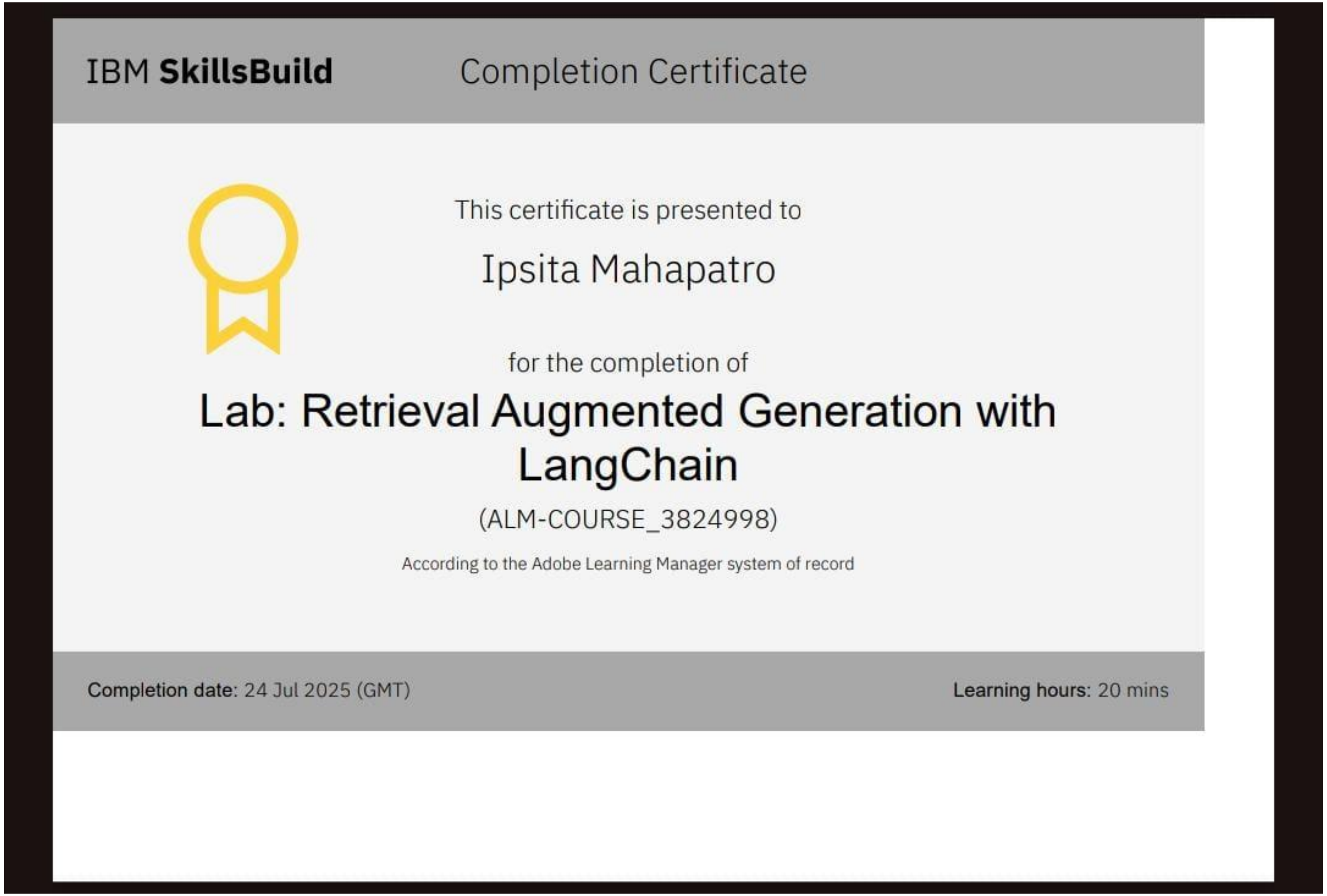
Journey to Cloud: Envisioning Your Solution

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

Verify: https://www.credly.com/badges/b77f0d95-e04a-476a-86e3-00d58a8e9ff9

IBM

IBM CERTIFICATIONS


IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Ipsita Mahapatro

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