
CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:

Aditi Karpe - K. K. Wagh Institute of Engineering Education & Research- M.C.A.

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

- Develop a machine learning model that classifies different power distribution system faults using voltage and current phasor data. The model will distinguish between normal conditions and fault types. This classification enables rapid and accurate fault detection, improving response time and ensuring grid stability.
- **Key Components:**
 - **Data Source:** Kaggle power system faults dataset
 - **Preprocessing:** Cleaning, normalization, and feature engineering
 - **Modeling:** Used Random Forest, Snap Logistic Regression, and SVM
 - **Platform:** IBM Cloud (Watsonx.ai Studio + Cloud Object Storage)
 - **Evaluation Metrics:** Accuracy, precision, recall, F1-score

SYSTEM APPROACH

- **System requirements Data Collection:**

Dataset from Kaggle with various fault scenarios and phasor measurements.

- **Preprocessing:**

Null value removal, normalization. Splitting dataset into train/test sets.

- **Model Building:**

Implemented multiple ML algorithms using IBM Watsonx.ai Studio. Random Forest gave highest accuracy : 0.409

- **Model Evaluation:**

Compared performance of Random Forest, SVM, and Snap Logistic Regression. Evaluated using confusion matrix and classification report.

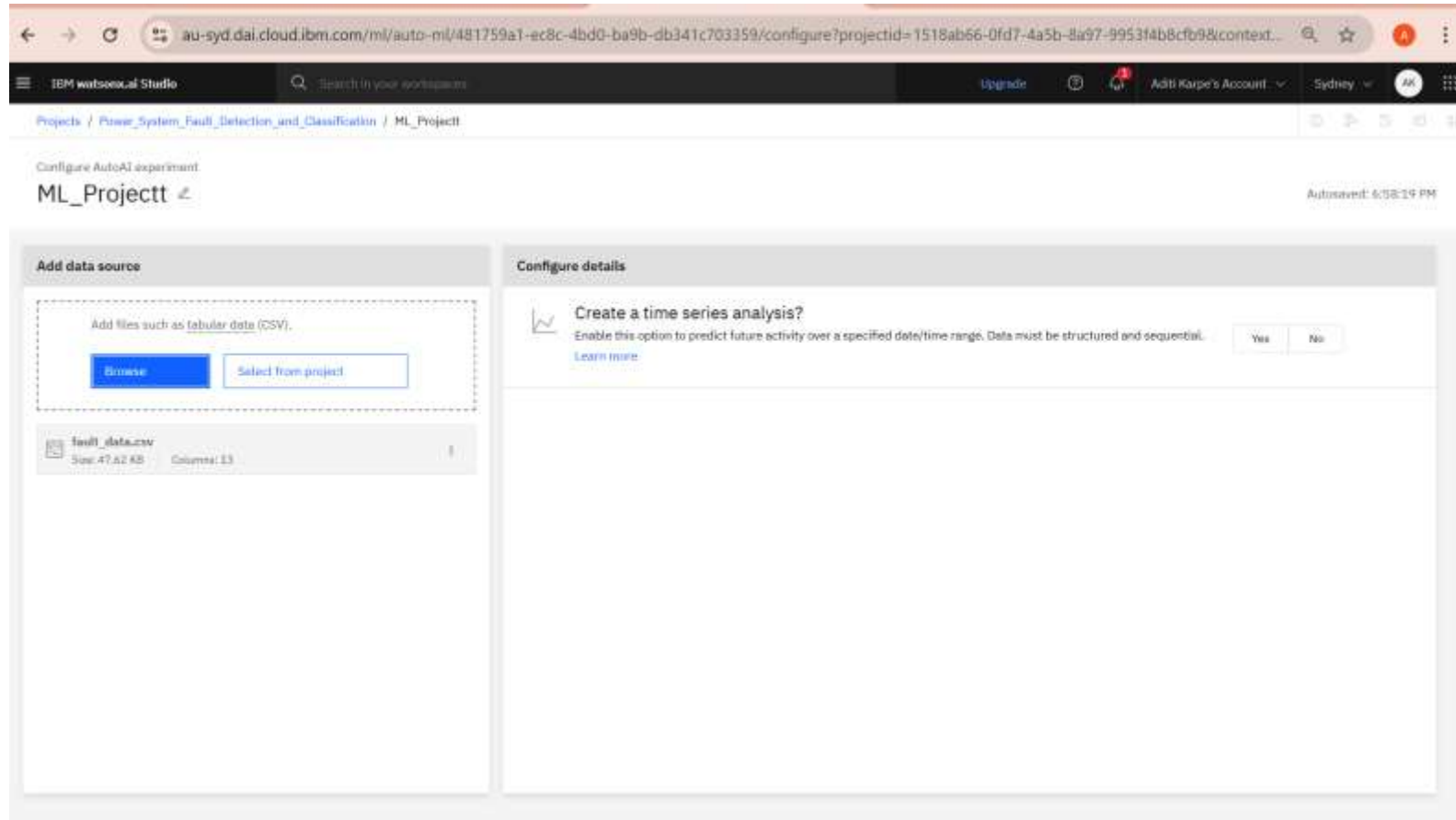
- **Deployment:**

Trained model deployed using IBM Watsonx.ai Studio. Storage handled via IBM Cloud Object Storage. Model ready to classify faults on new input data.

ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**
 - Used Random Forest, SVM, and Snap Logistic Regression. Random Forest was chosen for its better performance and handling of multi-class fault types.
- **Data Input:**
 - Voltage and current phasor values from the Kaggle power system fault dataset.
- **Training Process:**
 - Supervised Learning using labelled fault types. Models were trained in IBM Watson Studio.
- **Prediction Process:**
 - The model predicts the type of fault based on new phasor inputs.

RESULT



Step 1: Uploaded fault_data.csv as the dataset in IBM Watsonx.ai Studio.

RESULT

Configure AutoAI experiment
ML_Projectt [↗](#) Autosaved: 6:58:19 PM

Add data source

Add files such as tabular data (CSV).


[Browse](#) [Select from project](#)

fault_data.csv


Size: 47.62 KB Columns: 13

1

Configure details

 **Create a time series analysis?**
Enable this option to predict future activity over a specified date/time range. Data must be structured and sequential.
[Learn more](#)

☐ Yes ☒ No

 **What do you want to predict?**
Prediction column [?](#)

Fault Type

Prediction column: Fault Type CUH remaining: 10.01 CUH

PREDICTION TYPE

Multiclass Classification

OPTIMIZED FOR

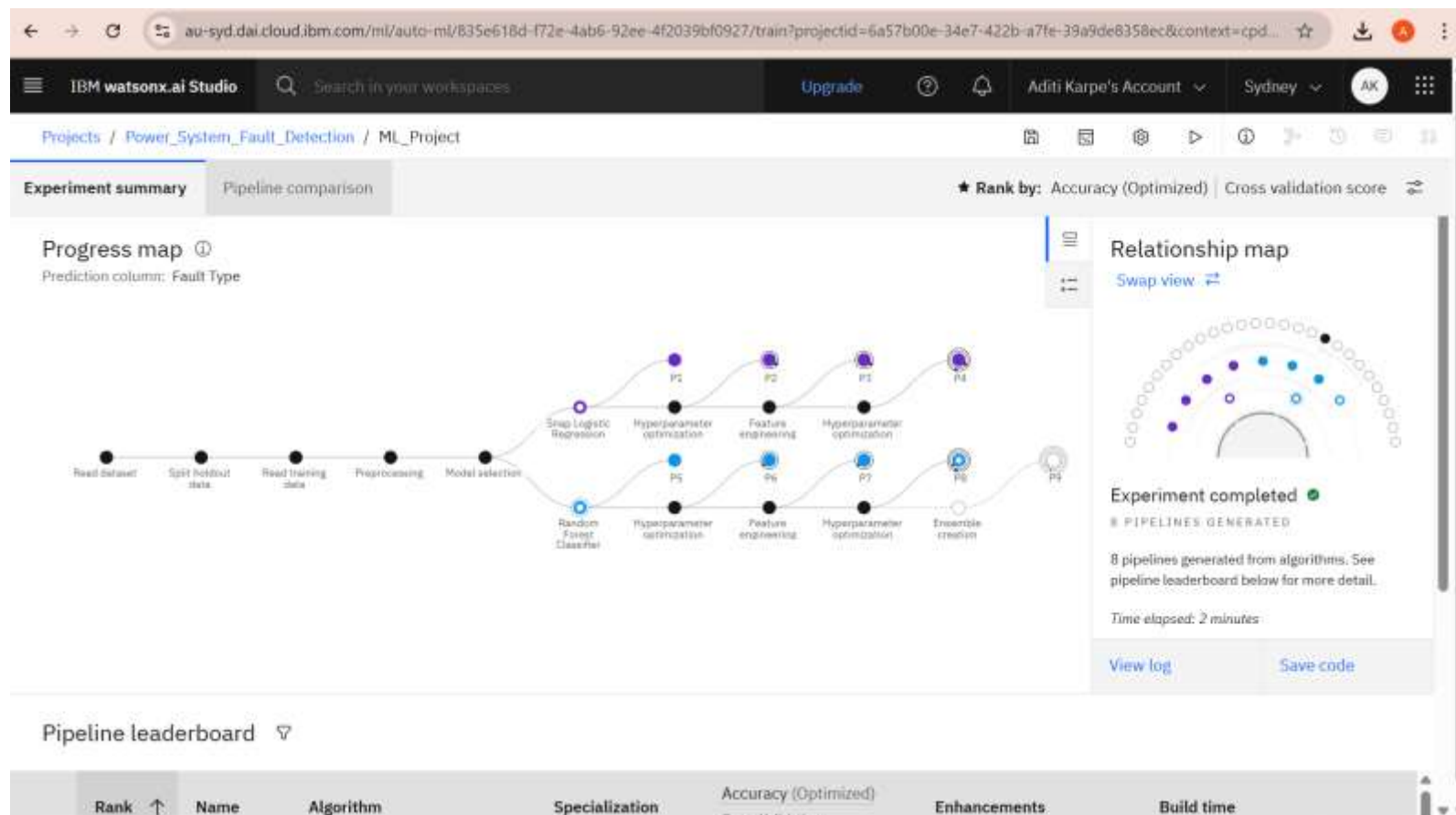
Accuracy & run time

Experiment settings

[Run experiment](#)

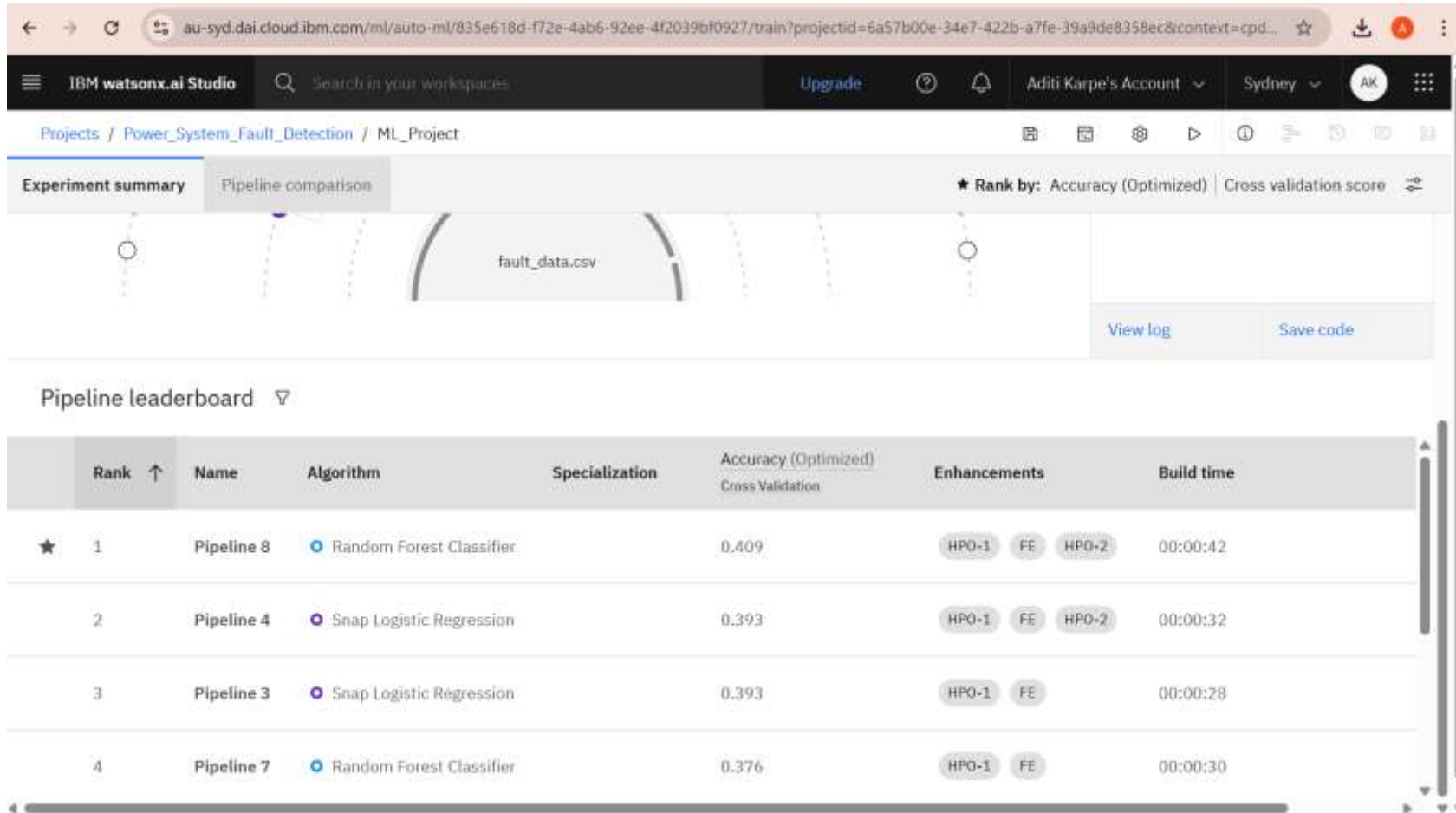
Step 2: Selected Fault Type as the prediction column and initialized the AutoAI experiment for multiclass classification.

RESULT



Step 3: AutoAI generated 8 machine learning pipelines using different algorithms and feature transformers.

RESULT

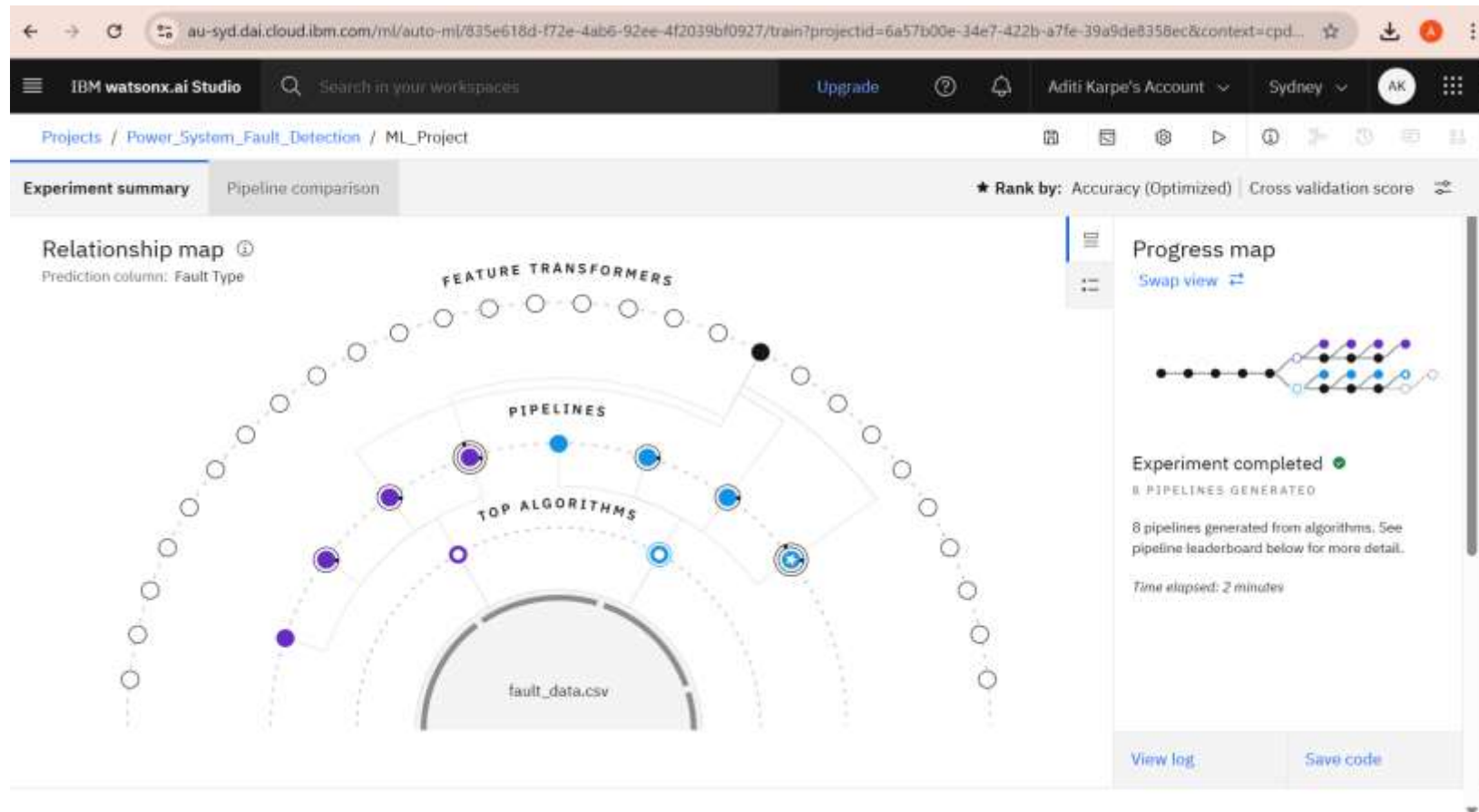


The screenshot displays the IBM Watsonx.ai Studio interface. At the top, the browser address bar shows a URL for a specific project. The interface includes a navigation bar with 'IBM watsonx.ai Studio', a search bar, and user account information. Below this, the breadcrumb path is 'Projects / Power_System_Fault_Detection / ML_Project'. The main content area has two tabs: 'Experiment summary' and 'Pipeline comparison', with the latter being active. A visualization of a pipeline is shown, including a data source 'fault_data.csv'. To the right of the pipeline, there are buttons for 'View log' and 'Save code'. Below the pipeline visualization, a 'Pipeline leaderboard' is displayed as a table. The table is sorted by 'Rank' and 'Accuracy (Optimized)'. The top-ranked pipeline is 'Pipeline 8', which uses a 'Random Forest Classifier' and has an accuracy of 0.409. Other pipelines follow in descending order of accuracy.

	Rank ↑	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 8	Random Forest Classifier		0.409	HPO-1 FE HPO-2	00:00:42
	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	00:00:28
	4	Pipeline 7	Random Forest Classifier		0.376	HPO-1 FE	00:00:30

Step 4: Pipelines were ranked based on optimized accuracy using cross-validation.

RESULT



Step 5: Visualized the relationship map showing connections between the dataset, algorithms, and transformers.

RESULT

IBM watsonx.ai Studio

Deployment spaces / Power_System_Project_Deploy / PII - Random Forest Classifier: ML_Project /

Power_System_Fault_Detection_Deploy 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](#)

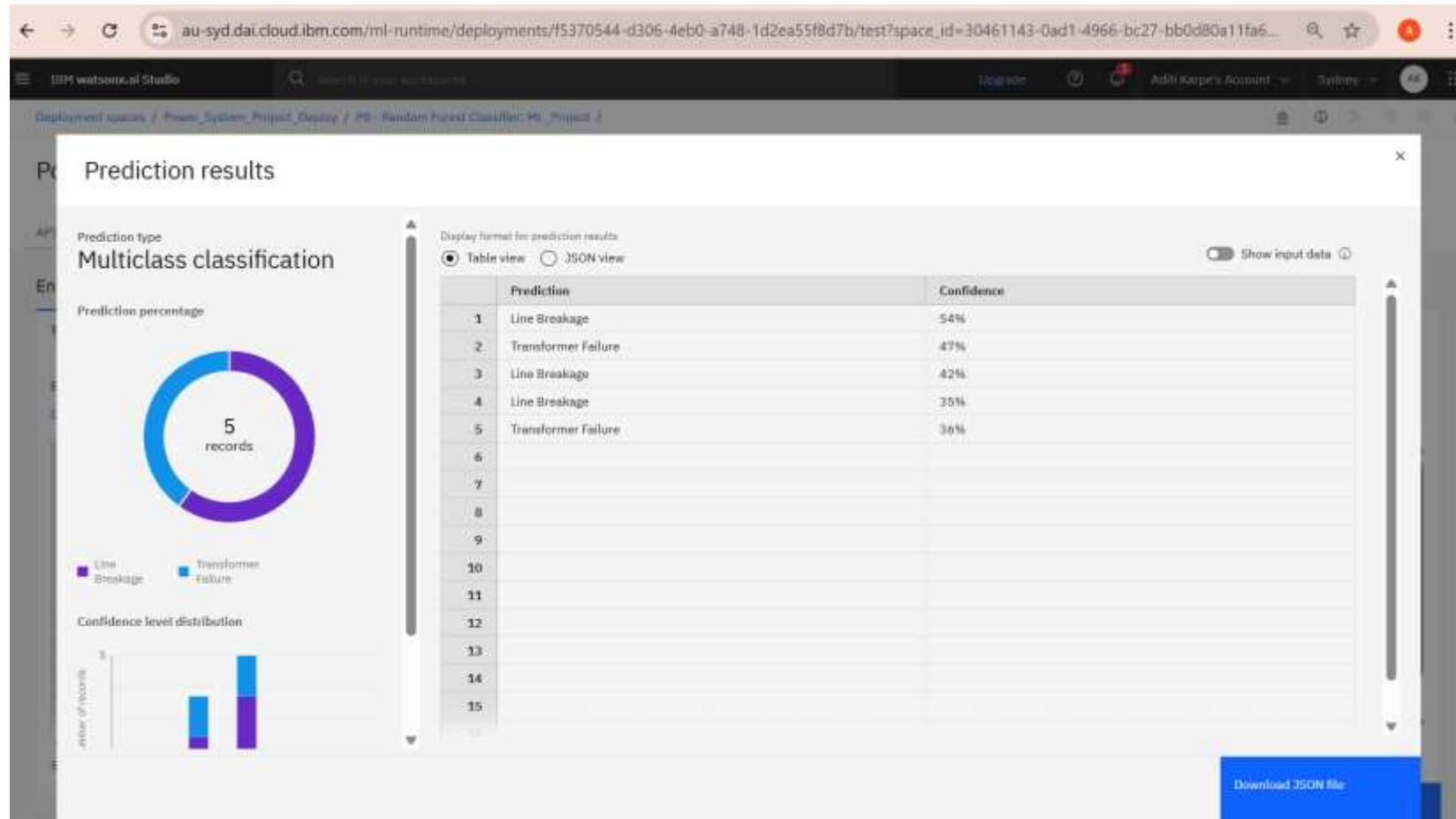
	Fault ID (other)	Fault Location (Latitude, Longitude) (other)	Voltage (V) (double)	Current (A) (double)	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h) (double)	Weather (other)
1	F004	(34.055, -118.242)	2050	240	48	23	10	clear
2	F009	(34.2294, -118.2988)	2133	229	52	20	18	snowy
3	F015	(34.2256, -118.9178)	1848	231	49	39	13	rainy
4	F025	(34.8937, -118.532)	1869	218	45	22	18	thunderstorm
5	F026	(34.9593, -118.9488)	2016	197	47	35	15	rainy
6								
7								
8								
9								

5 rows, 22 columns

Predict

Step 6: After completing the experiment, input data was provided to the best-performing pipeline for prediction.

RESULT



Step 7: The system successfully predicted the **Fault Type** based on the input data using the trained model.

CONCLUSION

- Successfully implemented a fault detection system using ML on IBM Cloud.
- Implemented multiple ML algorithms using IBM Watsonx.ai Studio.
- The models identifies fault types with moderate accuracy and serves as proof-of-concept.
- Demonstrates the potential of ML in real-time power system monitoring and fault classification.

FUTURE SCOPE

- Incorporate data from IoT sensors, smart meters, and SCADA systems to enhance fault detection accuracy.
- Extend to include fault location and severity prediction.
- Use deep learning models (e.g., LSTM, CNN) to improve performance and accurately analyze complex fault patterns.
- Improve model accuracy using Deep Learning (e.g., LSTM for time-series data).
- Develop a web-based dashboard to display real-time fault detection, sensor data, and fault insights in a user-friendly format.

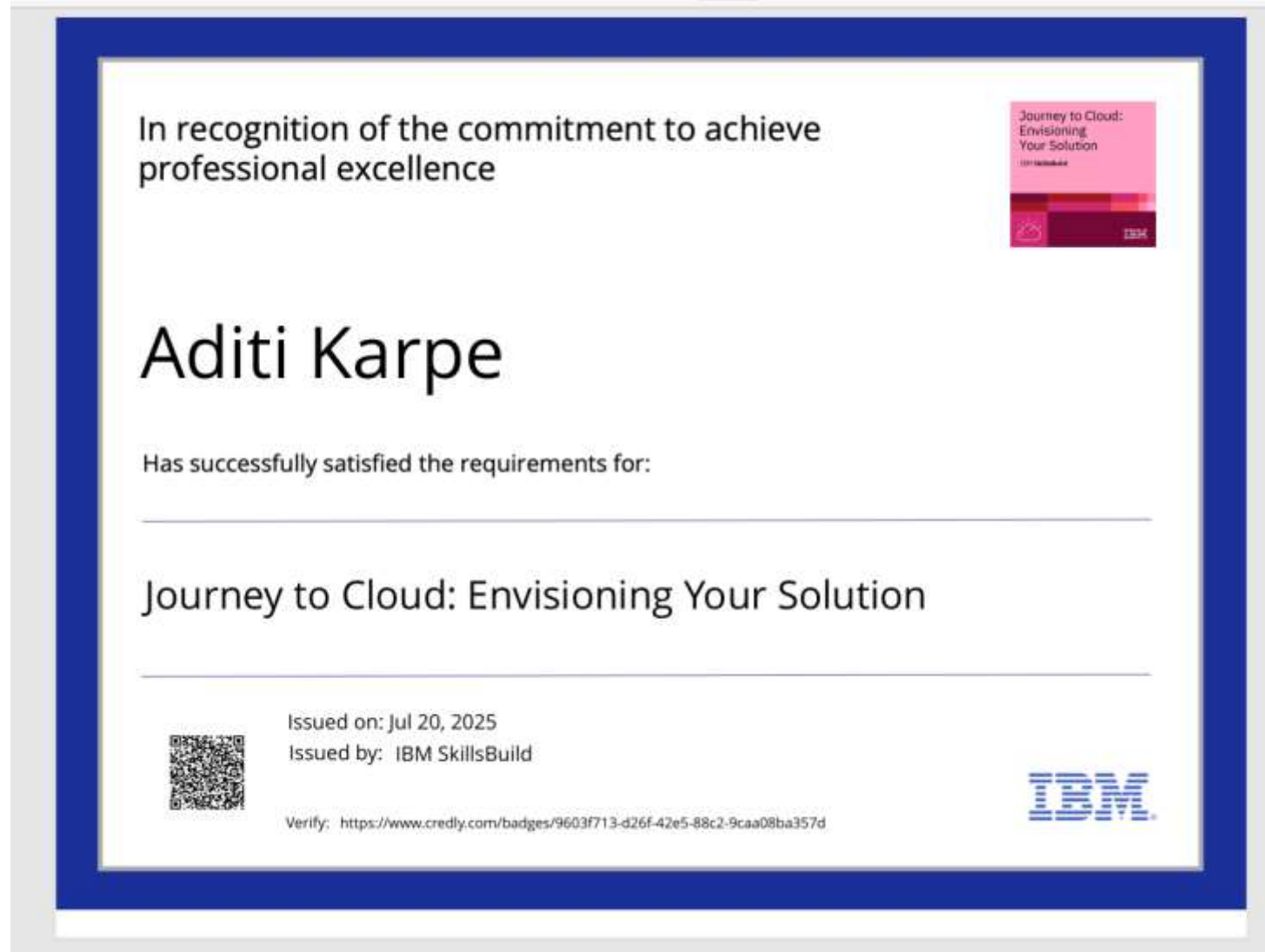
REFERENCES

- Kaggle Dataset : [Power System Faults Dataset](#)
- IBM Cloud Documentation : <https://cloud.ibm.com/docs>

IBM CERTIFICATIONS



IBM CERTIFICATIONS



IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to

Aditi Karpe

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