
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.

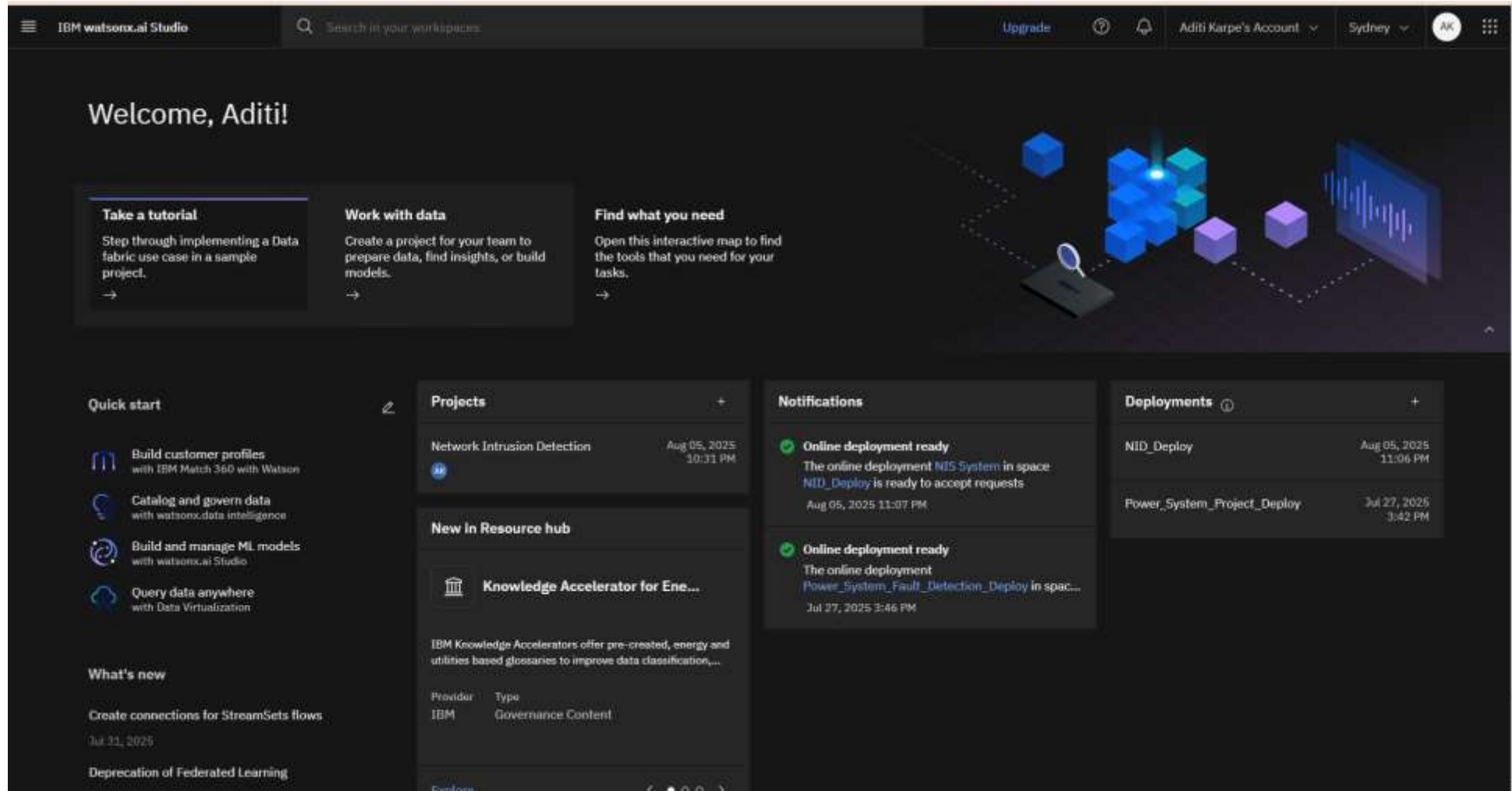
- **Deployment:**

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: Opened the IBM Watsonx.ai dashboard to begin the machine learning model development process.

RESULT

The screenshot shows the 'Create a project' dialog in IBM Watsonx.ai Studio. The interface includes a top navigation bar with the IBM Watsonx.ai Studio logo, a search bar, and user account information. The main content area is titled 'Create a project' and includes a subtitle: 'Start with a new, blank project or select from where to import an existing project.' On the left, there is a sidebar with a '+ New' button and two options: 'Local file' and 'Sample'. The main form is titled 'Define details' and contains several fields: 'Name' (filled with 'Power_System_Fault_Detection'), 'Description (optional)' (with a placeholder 'What's the purpose of this project?'), 'Tags (optional)' (with a placeholder 'Add tags'), and 'Storage' (set to 'Cloud Object Storage-dc'). Below the 'Storage' field, there is a note: 'Project includes integration with [Cloud Object Storage](#) for storing project assets.' At the bottom right of the form, there are two buttons: 'Cancel' and 'Create' (highlighted in blue).

Step 2: Created a new project and named it **Power_System_Fault_Detection** to organize all related assets and experiments.

RESULT

The screenshot displays the IBM watsonx.ai Studio interface. The top navigation bar includes the 'IBM watsonx.ai Studio' logo, a search bar, and user account information. The main content area is titled 'Projects / Power_System_Fault_Detection'. The left sidebar shows a navigation menu with options like 'General', 'Access control', 'Environments', 'Resource usage', 'Services & Integrations' (selected), and 'Tools'. The 'Services & Integrations' section is active, showing a table of associated services. The table has columns for 'Name' and 'Service type'. One service is listed: 'watsonx.ai Runtime-kz' with the service type 'watsonx.ai Runtime'. A blue button labeled 'Associate service' is visible in the top right corner of the table area.

Name	Service type
watsonx.ai Runtime-kz	watsonx.ai Runtime

Step 3: Associated the project with the **Watson Machine Learning Runtime service** to enable model training and deployment capabilities.

RESULT

The screenshot displays the IBM watsonx.ai Studio interface. At the top, the header includes the IBM watsonx.ai Studio logo, a search bar, and user account information (Aditi Karpe's Account, Sydney, AK). The main content area is titled 'Build machine learning models automatically' and includes a subtitle 'Define the details to create an AutoAI experiment asset and open it in the AutoAI tool.' The interface is divided into two main sections: 'Define details' and 'Define configuration'. In the 'Define details' section, the 'Name' field is filled with 'ML_Project', and the 'Description (optional)' field contains the text 'What's the purpose of this AutoAI experiment?'. The 'Define configuration' section shows the 'Environment definition' as 'Large: 8 CPU and 32 GB RAM'. At the bottom of the form, there are three buttons: 'Cancel', 'Back', and 'Create'.

IBM watsonx.ai Studio

Search in your workspace

Upgrade

Aditi Karpe's Account

Sydney

AK

Projects / Power_System_Fault_Detection

Build machine learning models automatically

Define the details to create an AutoAI experiment asset and open it in the AutoAI tool.

Define details

Name

ML_Project

Description (optional)

What's the purpose of this AutoAI experiment?

Tags (optional)

Add tags to make assets easier to find.

Start typing to add tags

Define configuration

watsonx.ai Runtime service instance

watsonx.ai Runtime-kz

Environment definition

Large: 8 CPU and 32 GB RAM

This environment definition consumes 20 capacity units per hour for training. For details, see [watsonx.ai Runtime plans](#).

Cancel Back Create

Step 4: Created and named the machine learning model as **ML_Project** to perform fault detection and classification tasks.

RESULT

The screenshot displays the IBM Watsonx.ai Studio interface for configuring an AutoAI experiment. The top navigation bar includes the IBM Watsonx.ai Studio logo, a search bar, and user account information (Aditi Karpe's Account, Sydney, AK). The breadcrumb trail indicates the current location: Projects / Power_System_Fault_Detection / ML_Project. The main heading is "Configure AutoAI experiment ML_Project", with an "Autosaved: 7:10:34 PM" timestamp.

The interface is divided into two main panels:

- Add data source:** This panel shows the process of adding a data source. It includes a dashed box with the text "Add files such as tabular data (CSV)." and two buttons: "Browse" and "Select from project". Below this, a file named "fault_data.csv" is listed with a size of 47.62 KB and 13 columns.
- Configure details:** This panel contains the "Create a time series analysis?" section. It includes a line graph icon, the text "Enable this option to predict future activity over a specified date/time range. Data must be structured and sequential.", a "Learn more" link, and two buttons: "Yes" and "No".

Step 5: Uploaded fault_data.csv as the dataset in IBM Watsonx.ai Studio.

RESULT

The screenshot displays the IBM Watson AI Studio interface. The top navigation bar includes the IBM Watson AI Studio logo, a search bar, and user account information (Aditi Karpe's Account, Sydney). The breadcrumb trail shows the path: Projects / Power_System_Fault_Detection / ML_Project. The main heading is "Configure AutoAI experiment ML_Project", with an "Autosaved: 7:10:34 PM" timestamp.

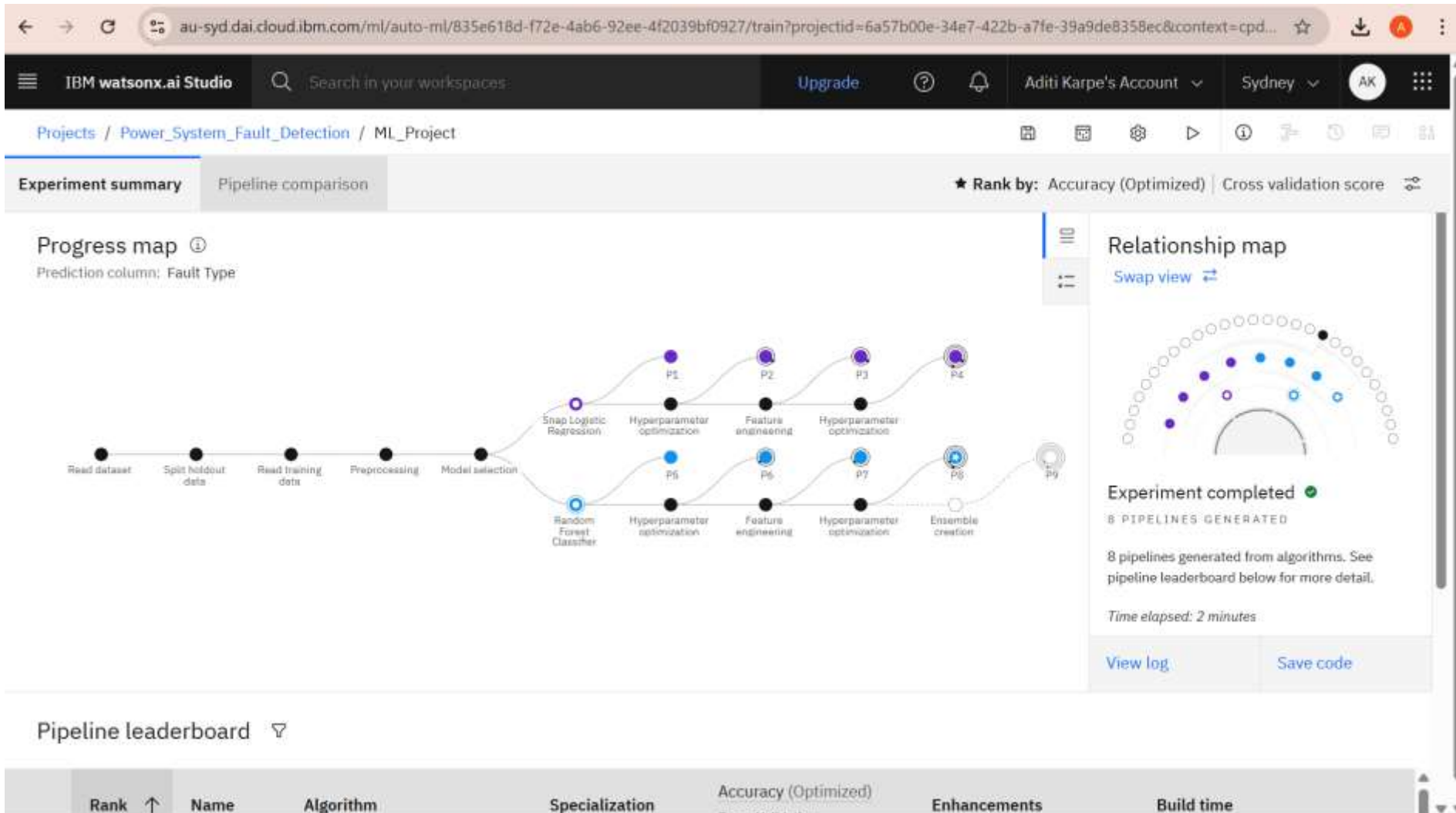
The interface is divided into two main panels:

- Add data source:** This panel on the left shows options to "Add files such as tabular data (CSV)". It includes "Browse" and "Select from project" buttons. Below these, a file named "fault_data.csv" is listed with a size of 47.62 KB and 13 columns.
- Configure details:** This panel on the right contains several configuration steps:
 - Create a time series analysis?** A toggle switch is set to "No".
 - What do you want to predict?** A dropdown menu shows "Fault Type" selected as the prediction column.
 - Prediction column:** Confirms "Fault Type" as the prediction column.
 - Prediction type:** Set to "Multiclass Classification".
 - Optimized for:** Set to "Accuracy & run time".

At the bottom, there is a "Run experiment" button and a "CUH remaining: 0.18 CUH" indicator.

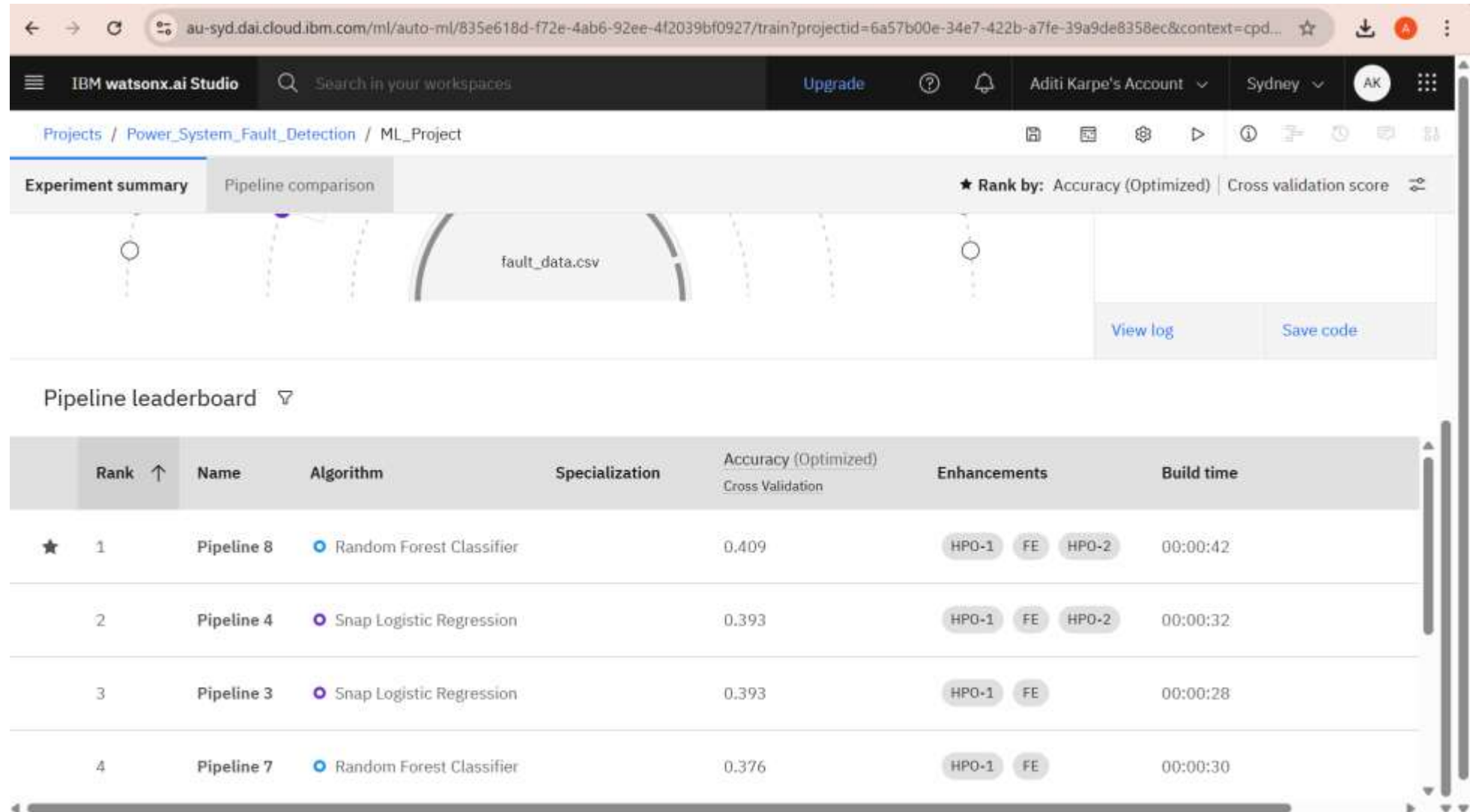
Step 6: Selected Fault Type as the prediction column and initialized the AutoAI experiment for multiclass classification.

RESULT



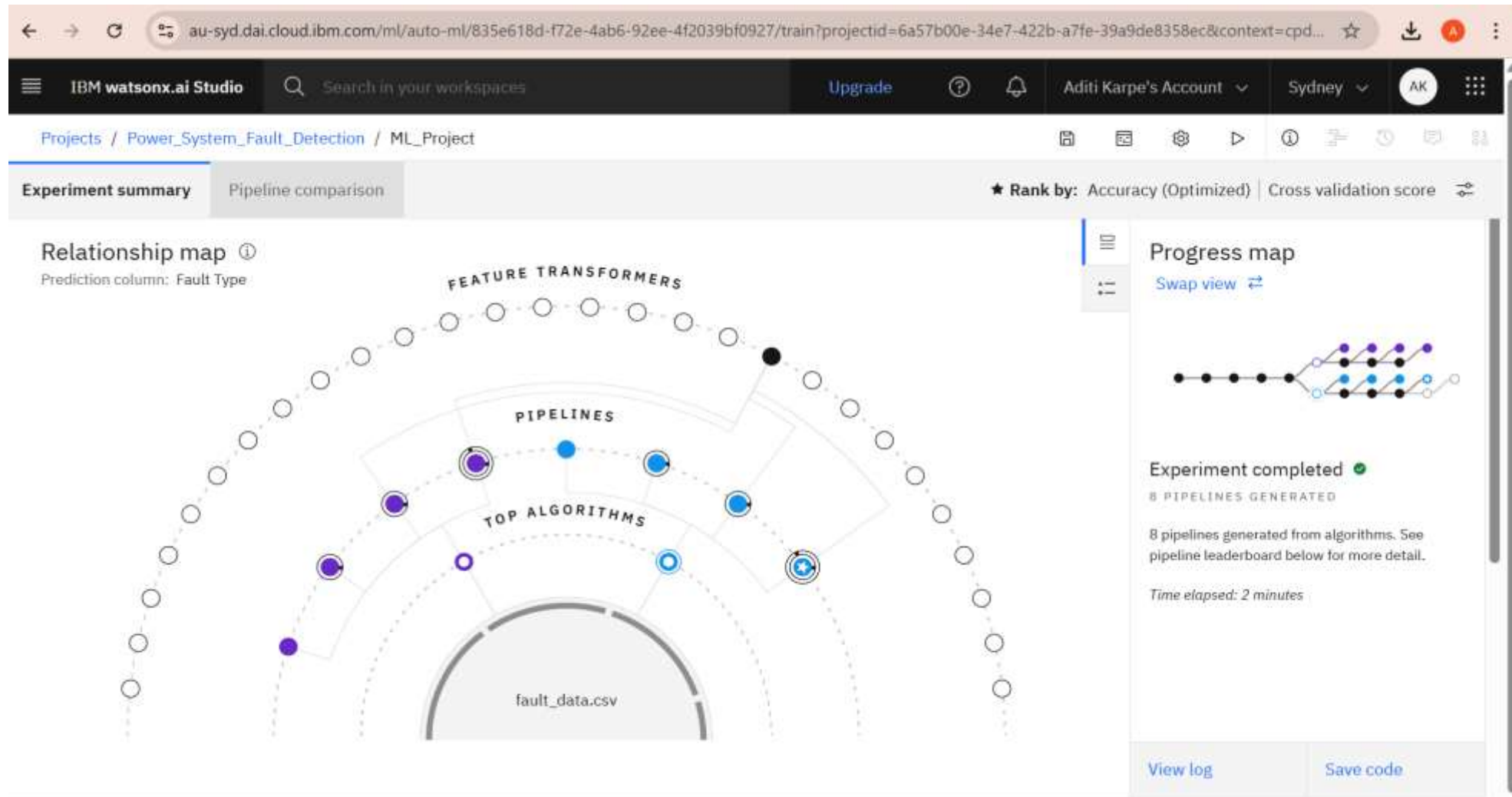
Step 7: AutoAI generated 8 machine learning pipelines using different algorithms and feature transformers.

RESULT



Step 8: Pipelines were ranked based on optimized accuracy using cross-validation.

RESULT



Step 9: Visualized the relationship map showing connections between the dataset, algorithms, and transformers.

RESULT

au-syd.dai.cloud.ibm.com/ml-runtime/deployments/f5370544-d306-4eb0-a748-1d2ea55f8d7b/test?space_id=30461143-0ad1-4966-bc27-bb0d80a11fa6...

IBM watsonx.ai Studio Search in your workspace Upgrade Aditi Karpe's Account Sydney AK

Deployment spaces / Power_System_Project_Deploy / PG - 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](#)

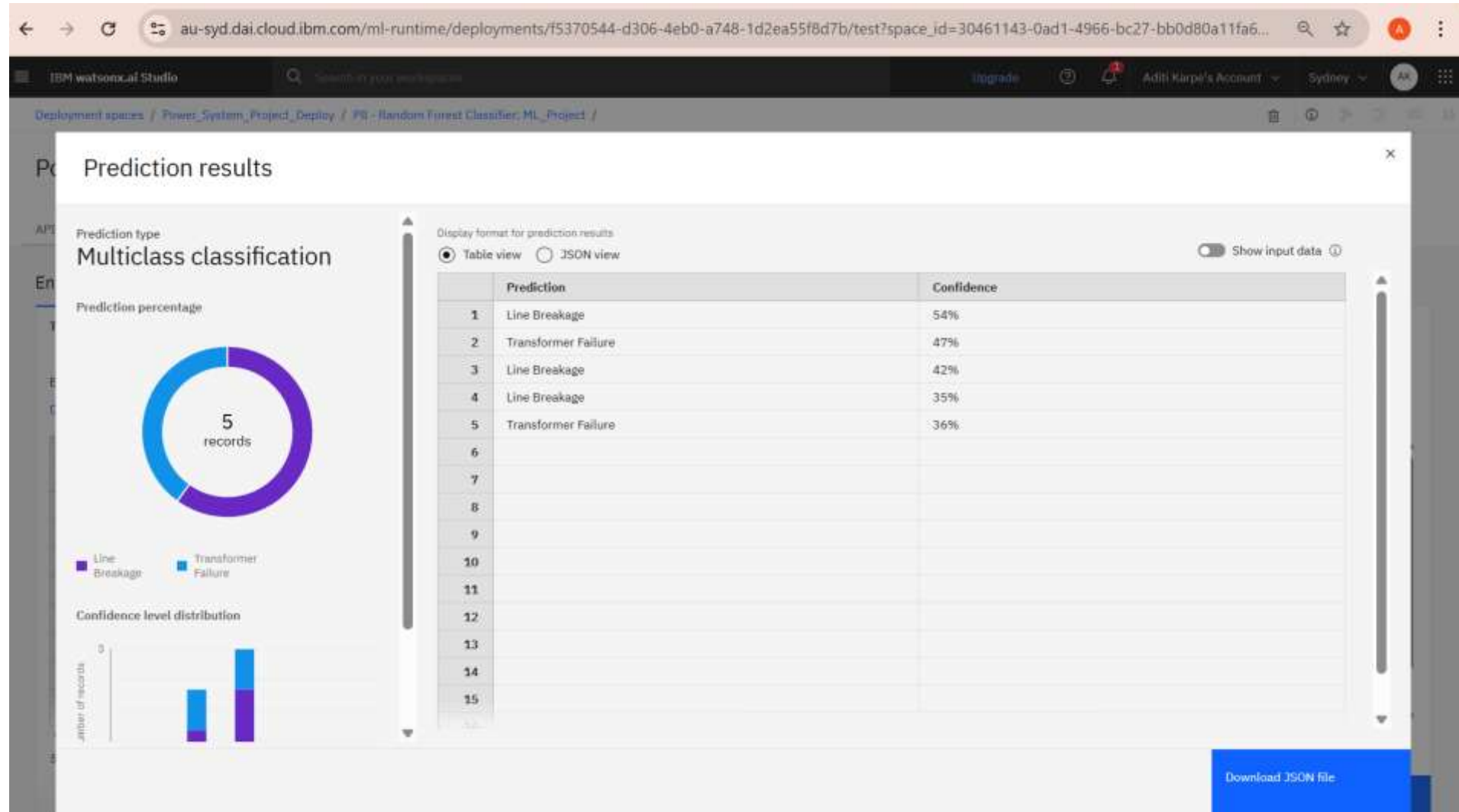
	Fault ID (other)	Fault Location (Latitude, Longitude) (other)	Voltage (V) (double)	Current (A) (double)	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h) (double)	Weather C
1	F004	(34.055, -118.242)	2050	240	48	23	10	clear
2	F008	(34.2294, -118.2988)	2133	229	52	20	18	snowy
3	F015	(34.2256, -118.9178)	1848	231	49	39	13	rainy
4	F025	(34.8937, -118.532)	1869	218	45	22	18	thunderst
5	F026	(34.9593, -118.9408)	2016	197	47	35	15	rainy
6								
7								
8								
9								

5 rows, 12 columns

Predict

Step 10: After completing the experiment, input data was provided to the best-performing pipeline for prediction.

RESULT



Step 11: 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

In recognition of the commitment to achieve
professional excellence



Aditi Karpe

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

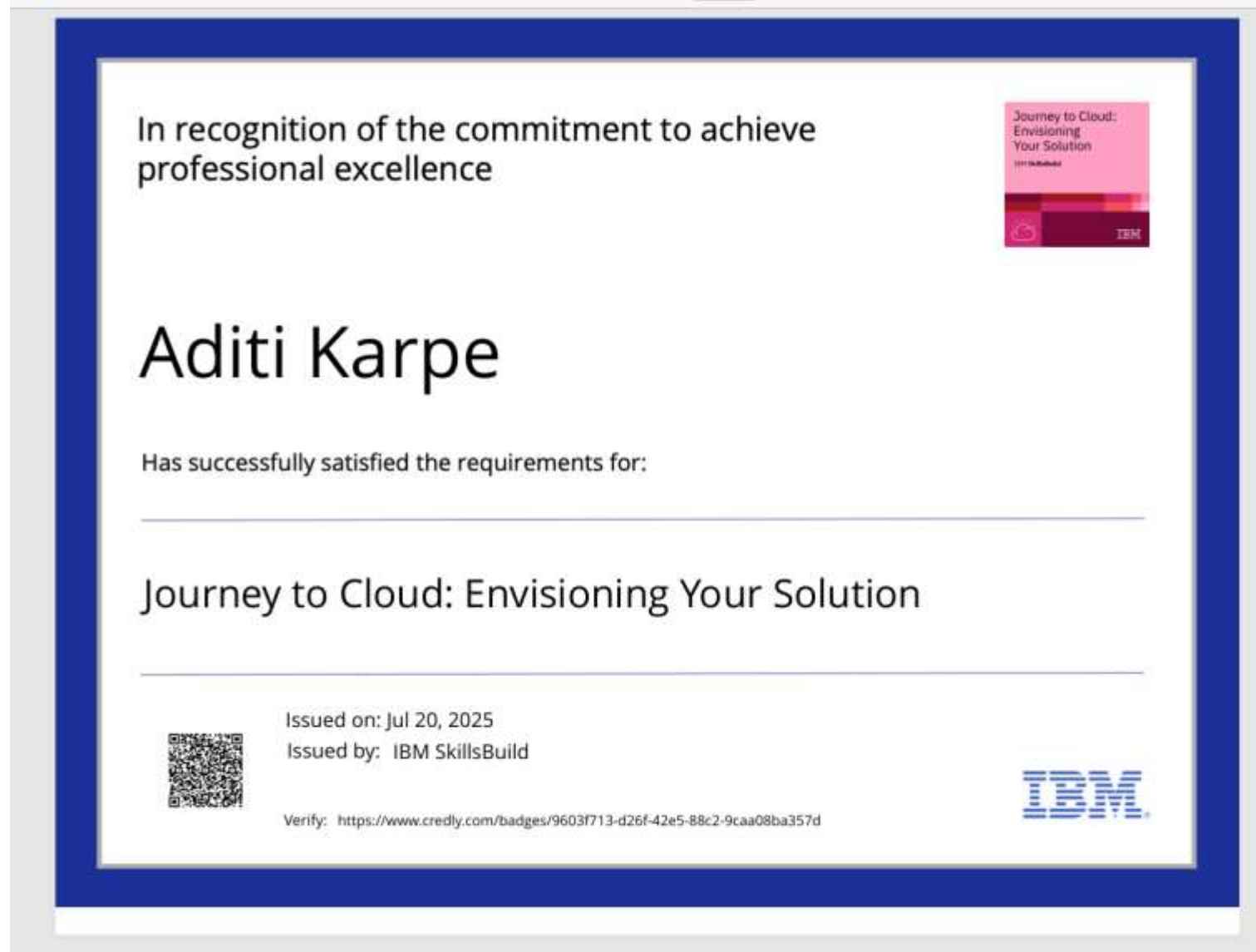


Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/c6b21c7c-b437-478c-a00d-e608267a3666>



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