CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

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

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

- 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:**

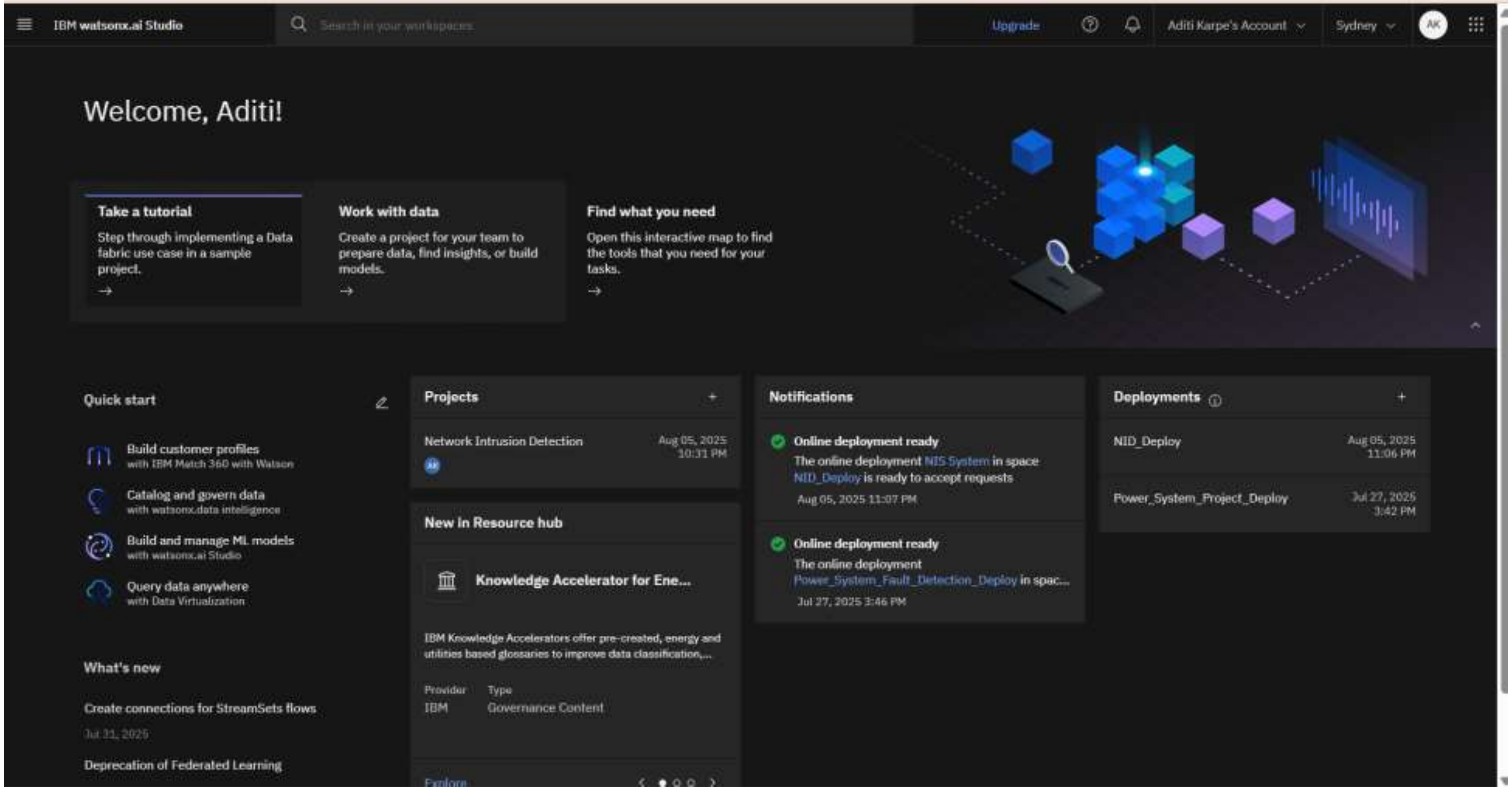
  Compared performance of Random Forest, SVM, and Snap Logistic Regression. Evaluated using confusion matrix and classification report.
- **Deployment:**

  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

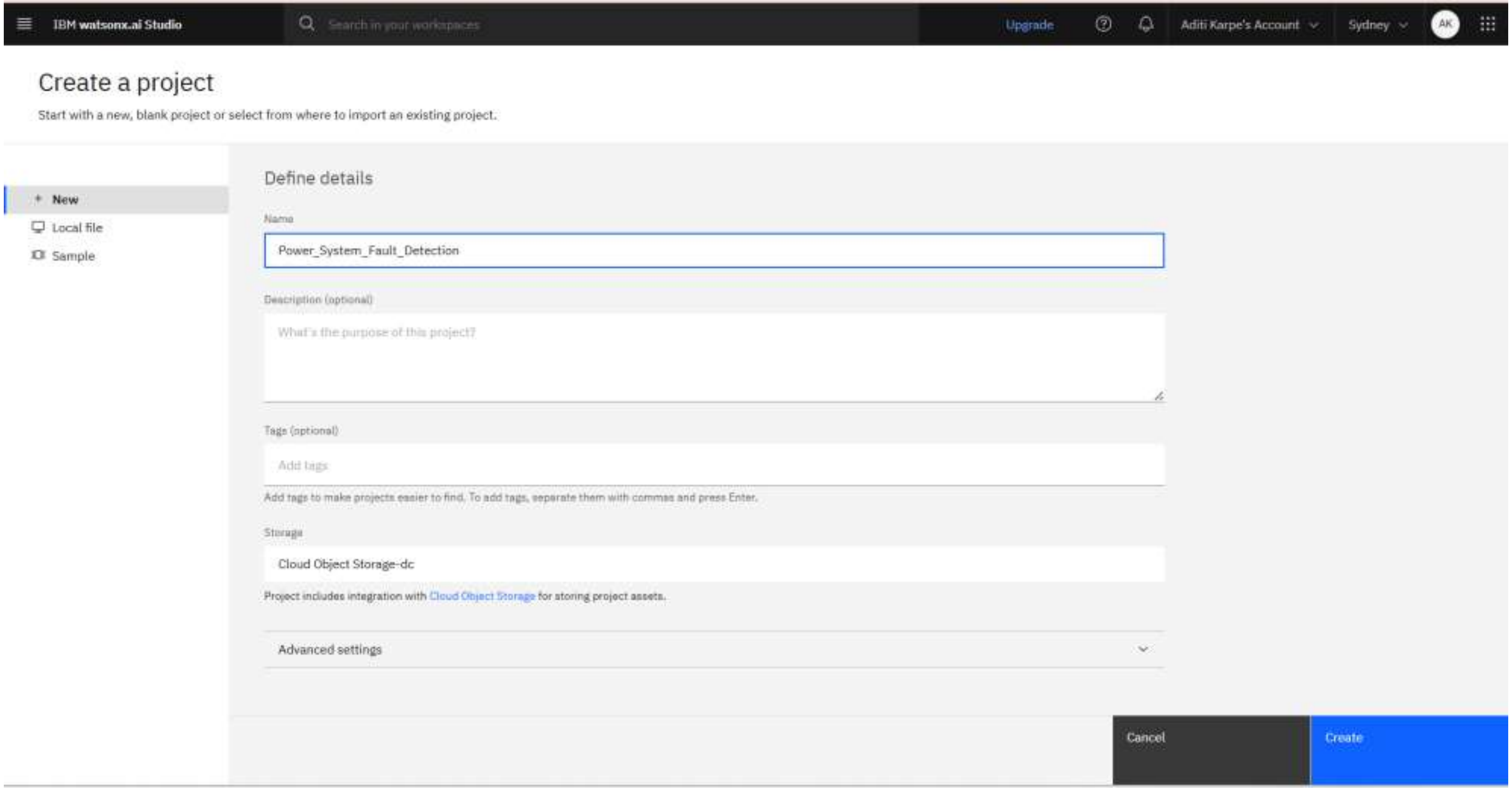
- **Algorithm Selection:**
  - Used Random Forest, SVM, and Snap Logistic Regression. Random Forest was chosen for its better performance and handling of multi-class fault types.
- **Data Input:**
  - Voltage and current phasor values from the Kaggle power system fault dataset.
- **Training Process:**
  - 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



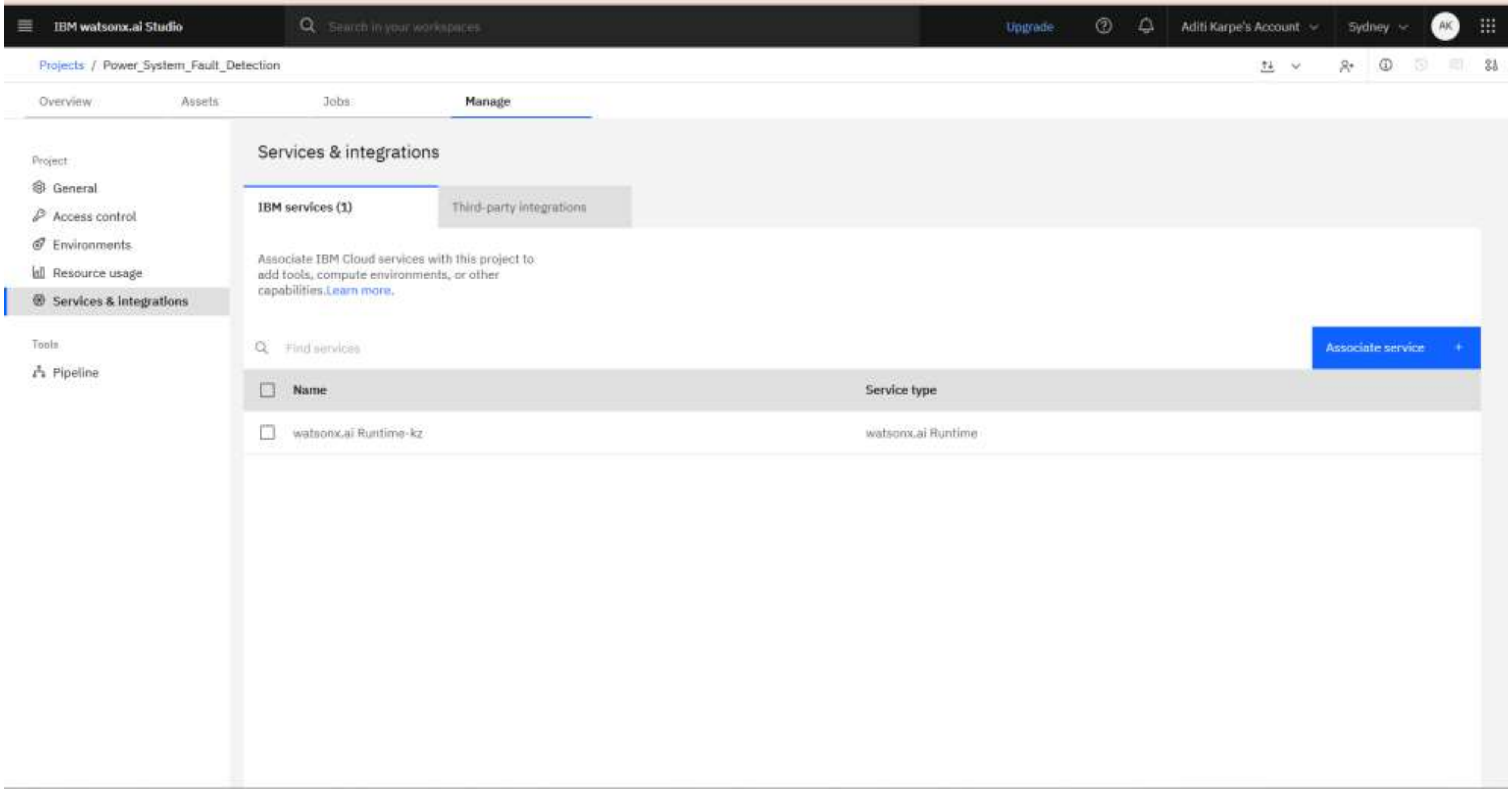
Step 1: Opened the IBM Watsonx.ai dashboard to begin the machine learning model development process.

RESULT



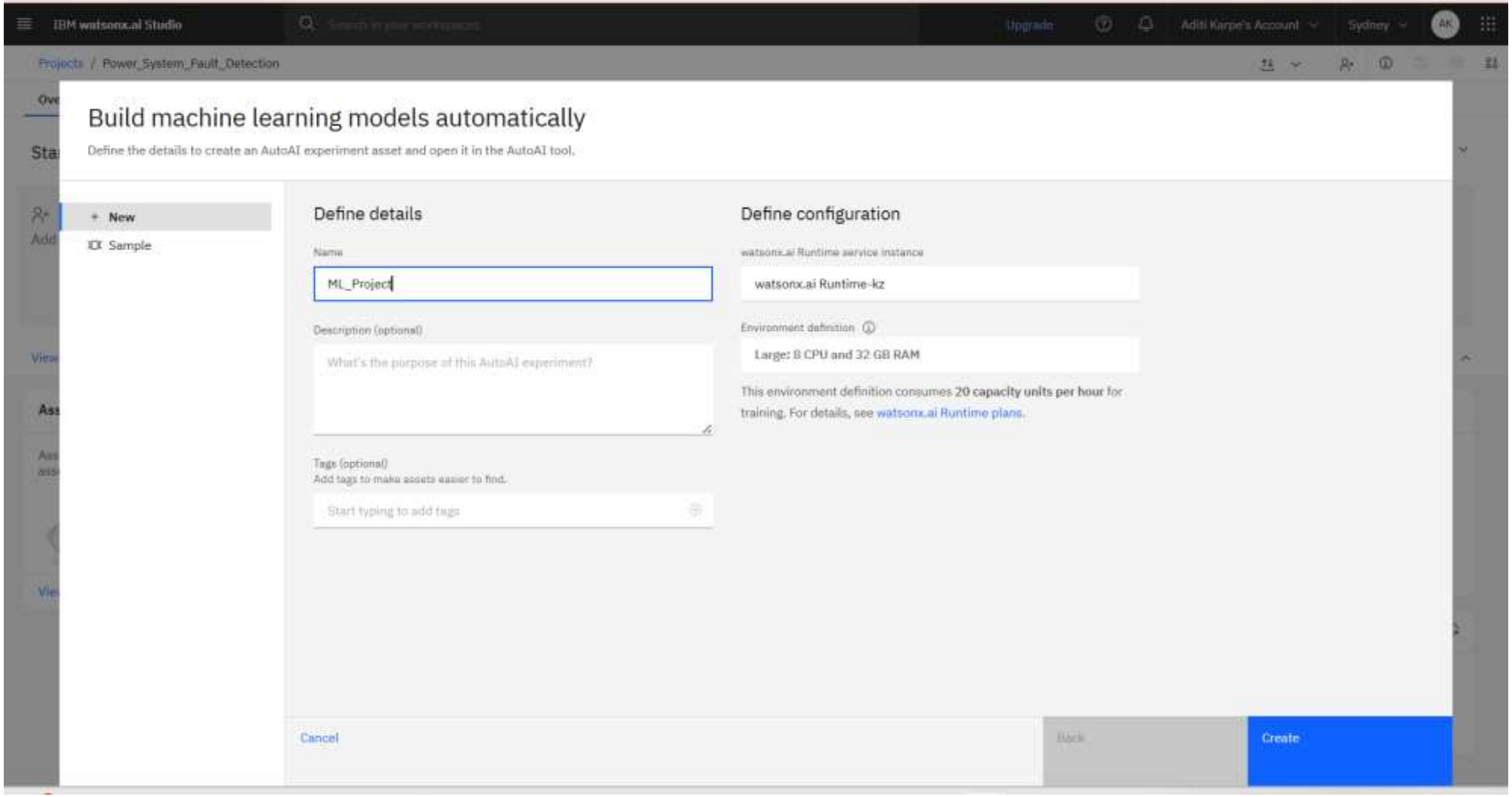
Step 2: Created a new project and named it **Power_System_Fault_Detection** to organize all related assets and experiments.

RESULT



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

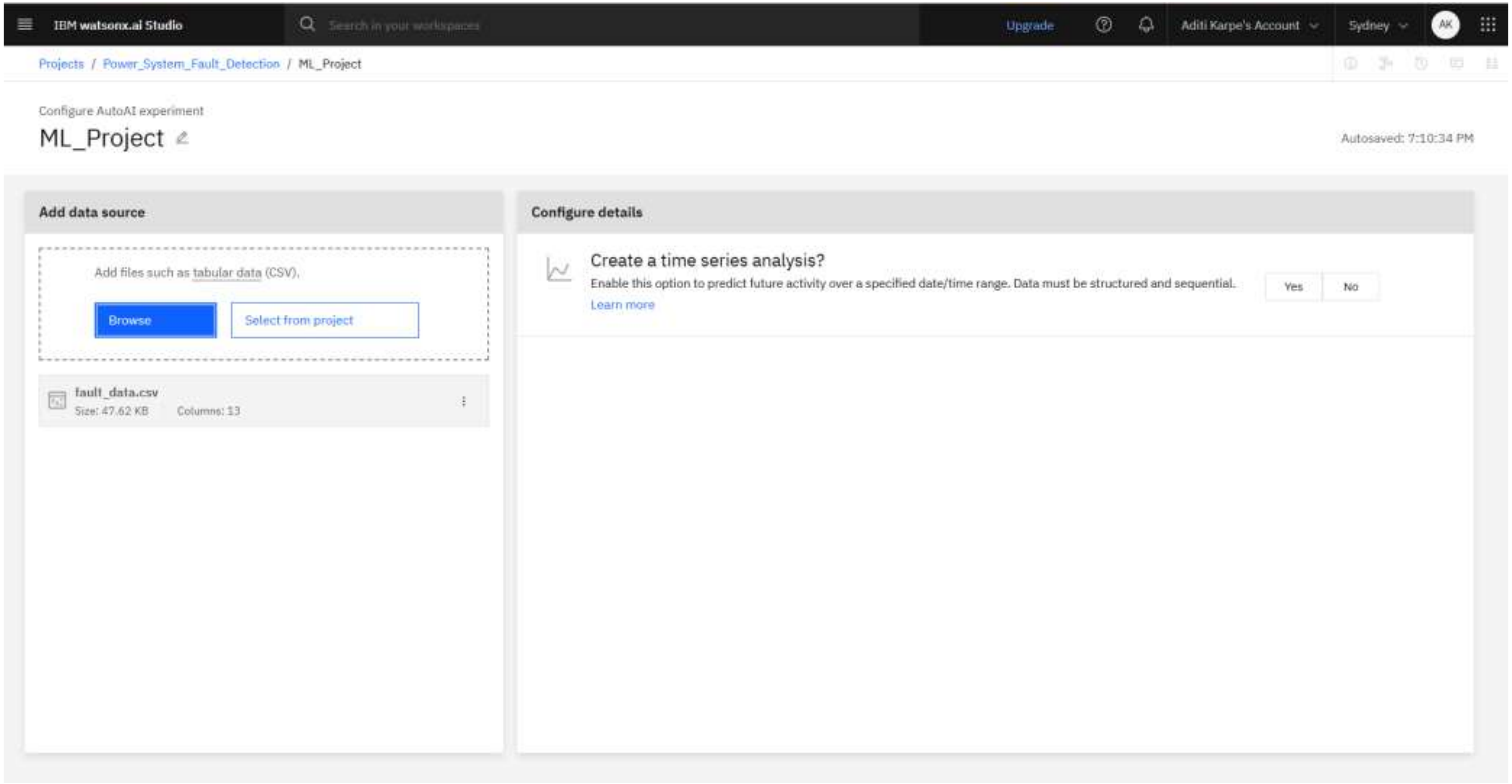
RESULT



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

edunet foundation

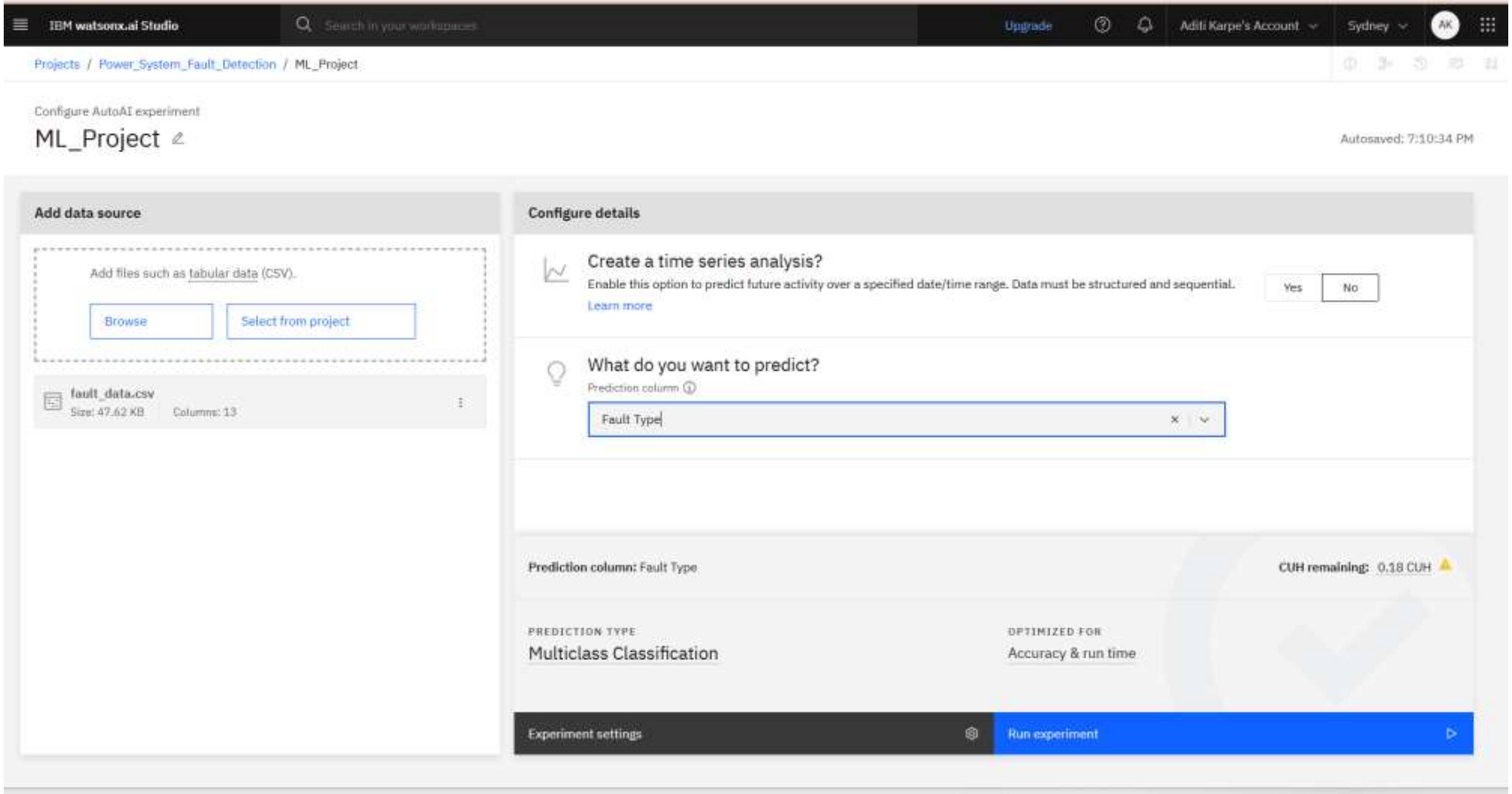
RESULT



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

edunet foundation

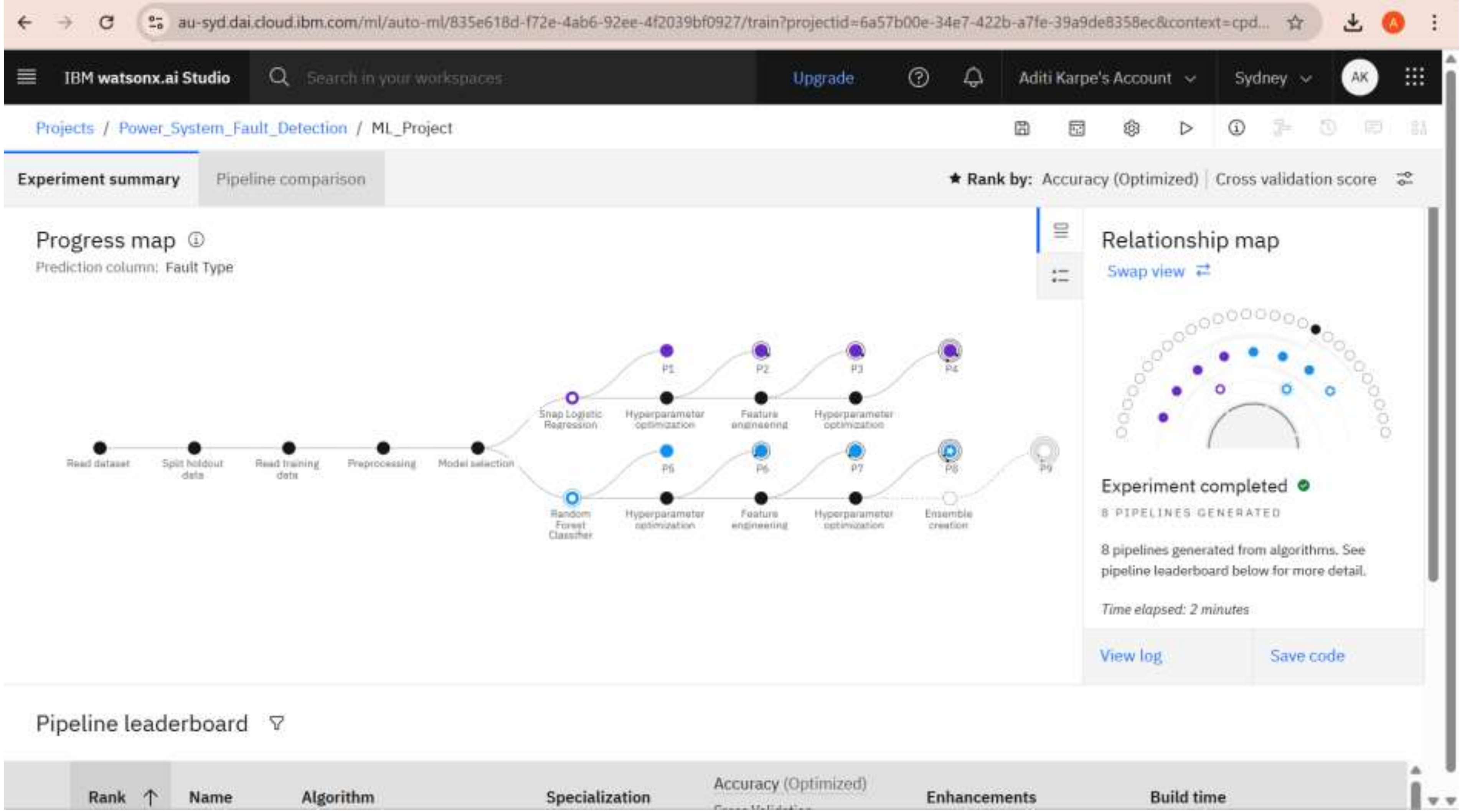
RESULT



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

edunet foundation

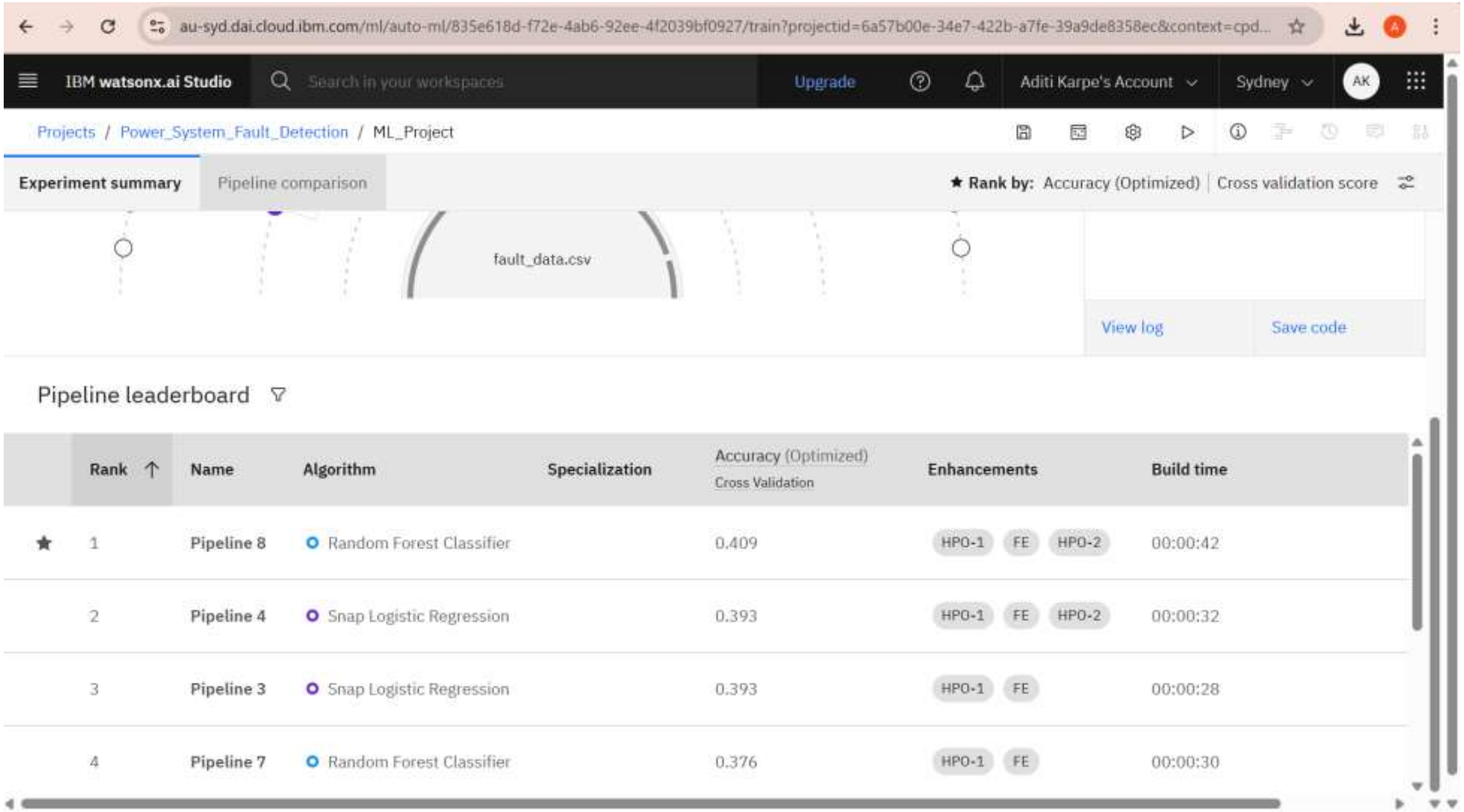
RESULT



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

edunet foundation

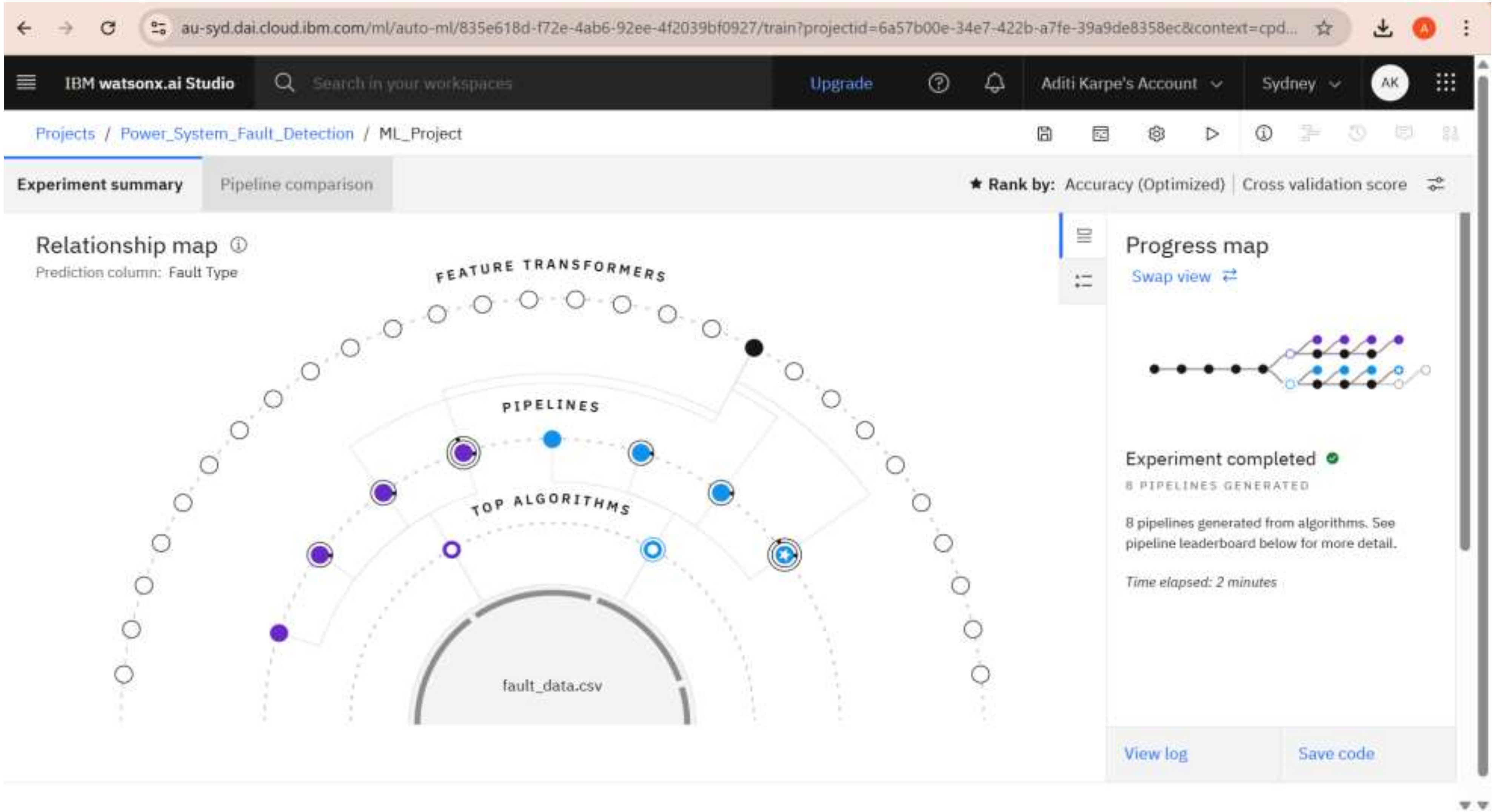
RESULT



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

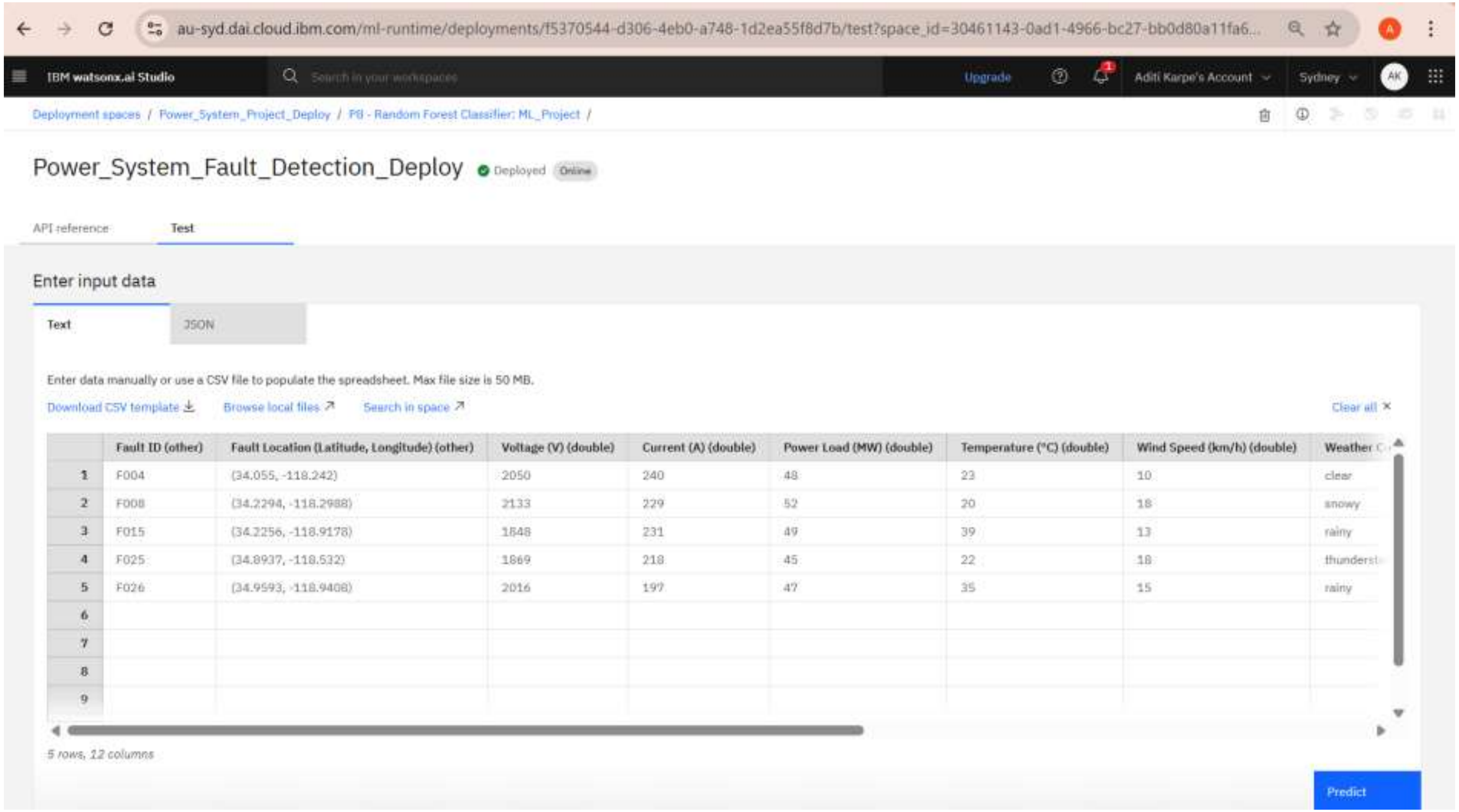
edunet foundation

RESULT



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

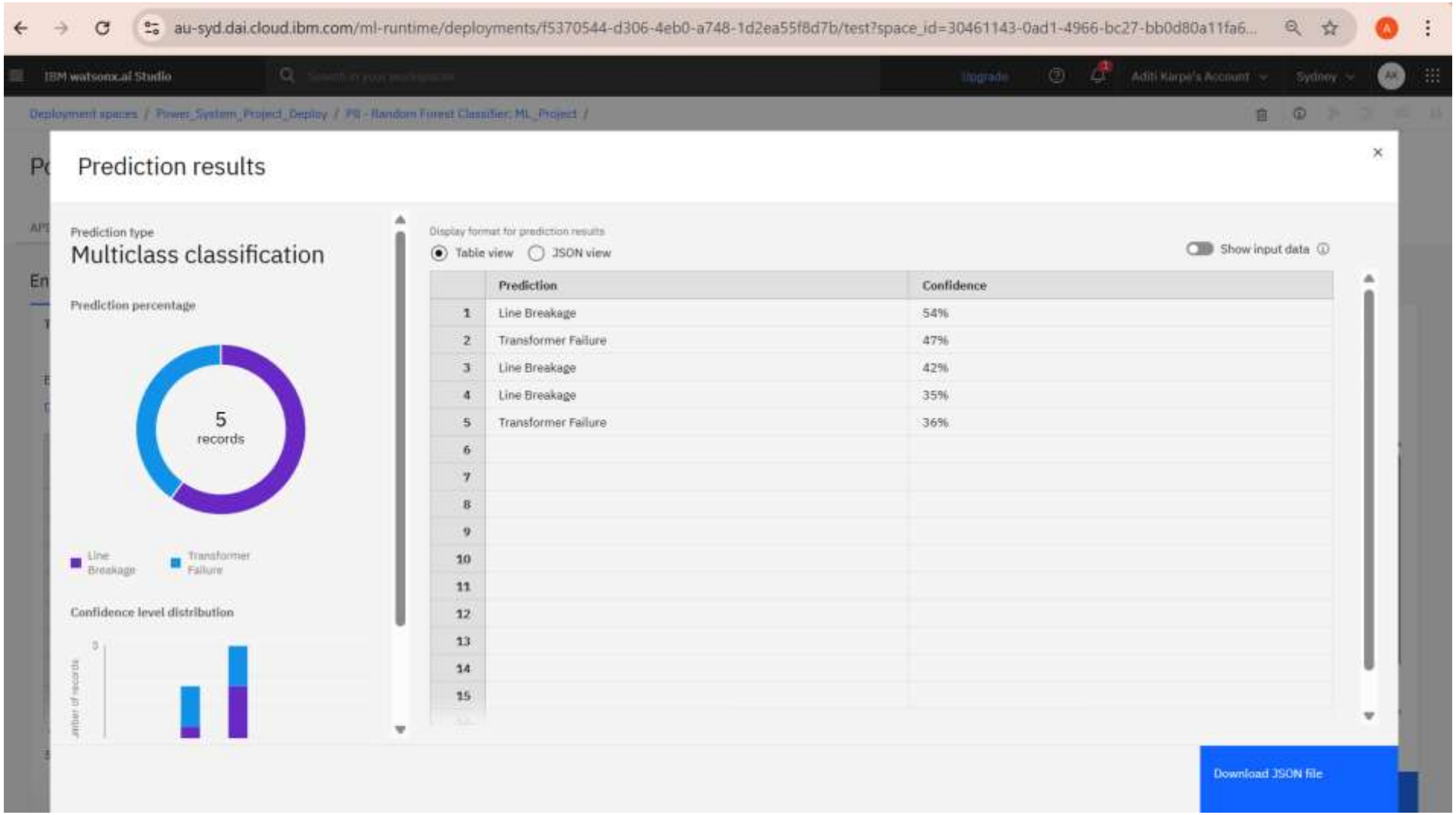
RESULT



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

edunet foundation

RESULT



Step 11: The system successfully predicted the **Fault Type** based on the input data using the trained model.

edunet foundation

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

Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

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

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Aditi Karpe

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

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

Verify: https://www.credly.com/badges/9603f713-d26f-42e5-88c2-9caa08ba357d

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