CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
1. Sejal Singh-Babasaheb Bhimrao Ambedkar University-Information Technology

edunet
foundation

OUTLINE

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

PROBLEM STATEMENT

Example: 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 system aims to address the challenge of detecting and classifying faults in a power distribution system to maintain grid stability and prevent equipment damage. This involves leveraging electrical measurement data and machine learning techniques to identify and categorize fault types accurately. The solution will consist of the following components:
- **Data Collection:**
  - **Gather historical fault data, including fault type, timestamp, voltage, current, power load, and environmental conditions like temperature and wind speed.**
- **Data Preprocessing:**
  - **Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.**
  - **Perform feature engineering to extract relevant inputs (e.g., phase angle differences, voltage drop patterns) that impact fault detection and classification.**
- **Machine Learning Algorithm:**
  - **Implement a machine learning classification model (e.g., Random Forest, SVM, or LSTM) to detect and classify different fault types such as Line-to-Ground, Line-to-Line, and Three-Phase faults.**
  - **Consider incorporating time-series patterns and environmental factors (like weather and load) to improve model robustness and prediction accuracy.**
- **Deployment:**
  - **Develop a user-friendly dashboard or application to monitor real-time fault detection and classification results.**
  - **Deploy the solution on a reliable and scalable platform to ensure fast response times, easy integration with control systems, and high availability.**
- **Evaluation:**
  - **Evaluate the model using accuracy, precision, recall, F1-score, and confusion matrix to assess fault classification performance.**
  - **Continuously fine-tune the model based on real-world testing, feedback from grid operators, and evolving data patterns.**
- **Result: The model provides timely and accurate classification of faults in power systems, enabling faster response and maintenance, reducing downtime, and improving grid reliability.**

SYSTEM APPROACH

The "System Approach" defines the tools, technologies, and methodology used to develop and deploy the fault detection system using IBM Cloud services. It emphasizes a cloud-based, scalable, and AI-driven architecture suitable for real-time monitoring and analysis in power systems.

- System Requirements:

- **Device:** Any computer with internet access
- **Browser:** Google Chrome / Microsoft Edge (latest version)
- **RAM:** Minimum 4 GB (8 GB recommended)
- **Storage:** Not much required locally, as computation is cloud-based

- Library required to build the model:

- **Operating System:** Windows/macOS/Linux (any OS that supports a browser)
- **Cloud Platform:** IBM Cloud (Lite account)
- **Tools Used:**
  - **Watsonx.ai** for building and training ML models
  - **IBM Cloud Object Storage** for dataset storage
  - **Watson Studio** for collaborative development (Jupyter/Notebooks)
  - **AutoAI (optional)** for model auto-generation
  - **Python 3.x environment** (via Watson Studio runtime)

ALGORITHM & DEPLOYMENT

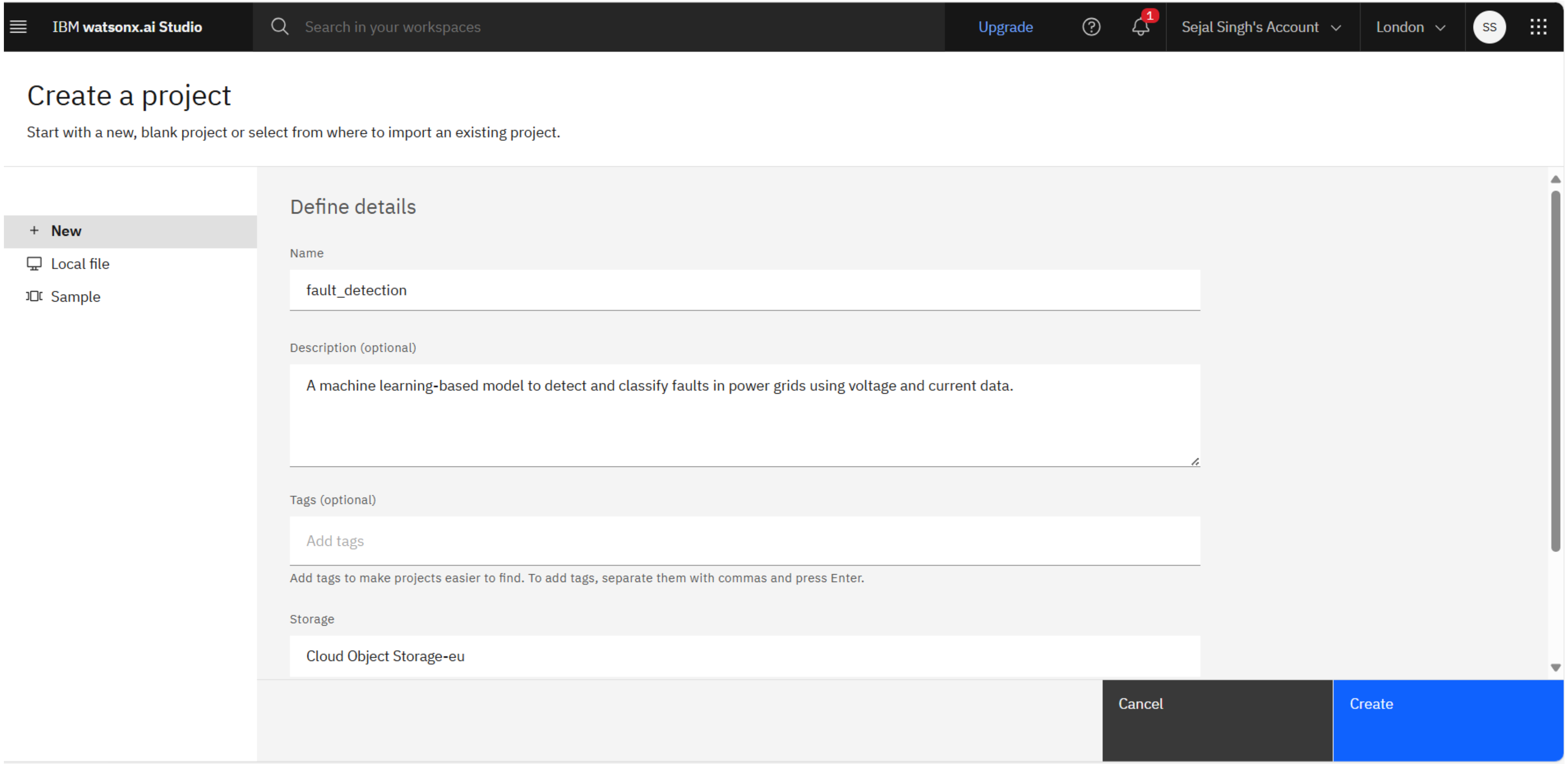
- The Algorithm section describes the machine learning algorithm chosen for detecting and classifying faults in the power distribution system. It includes details about the input data, training process, and how the model makes predictions.
- Algorithm Selection:
  - A classification algorithm like Random Forest is used to detect and categorize power system faults based on input features. It is chosen for its high accuracy, efficiency, and ability to handle complex datasets.
- Data Input:
  - The model uses voltage, current, power load, location, weather conditions, and equipment status to identify fault types.
- Training Process:
  - The model is trained on historical fault data using supervised learning with cross-validation and hyperparameter tuning to improve accuracy.
- Prediction Process:
  - Once trained, the model analyzes new data to predict the type of fault in real time, enabling quick and reliable fault detection.

RESULT

The results show that the machine learning model accurately detects and classifies power system faults.
Graphs comparing actual and predicted faults are included to show model performance.
Evaluation metrics like accuracy and precision confirm the model’s effectiveness.

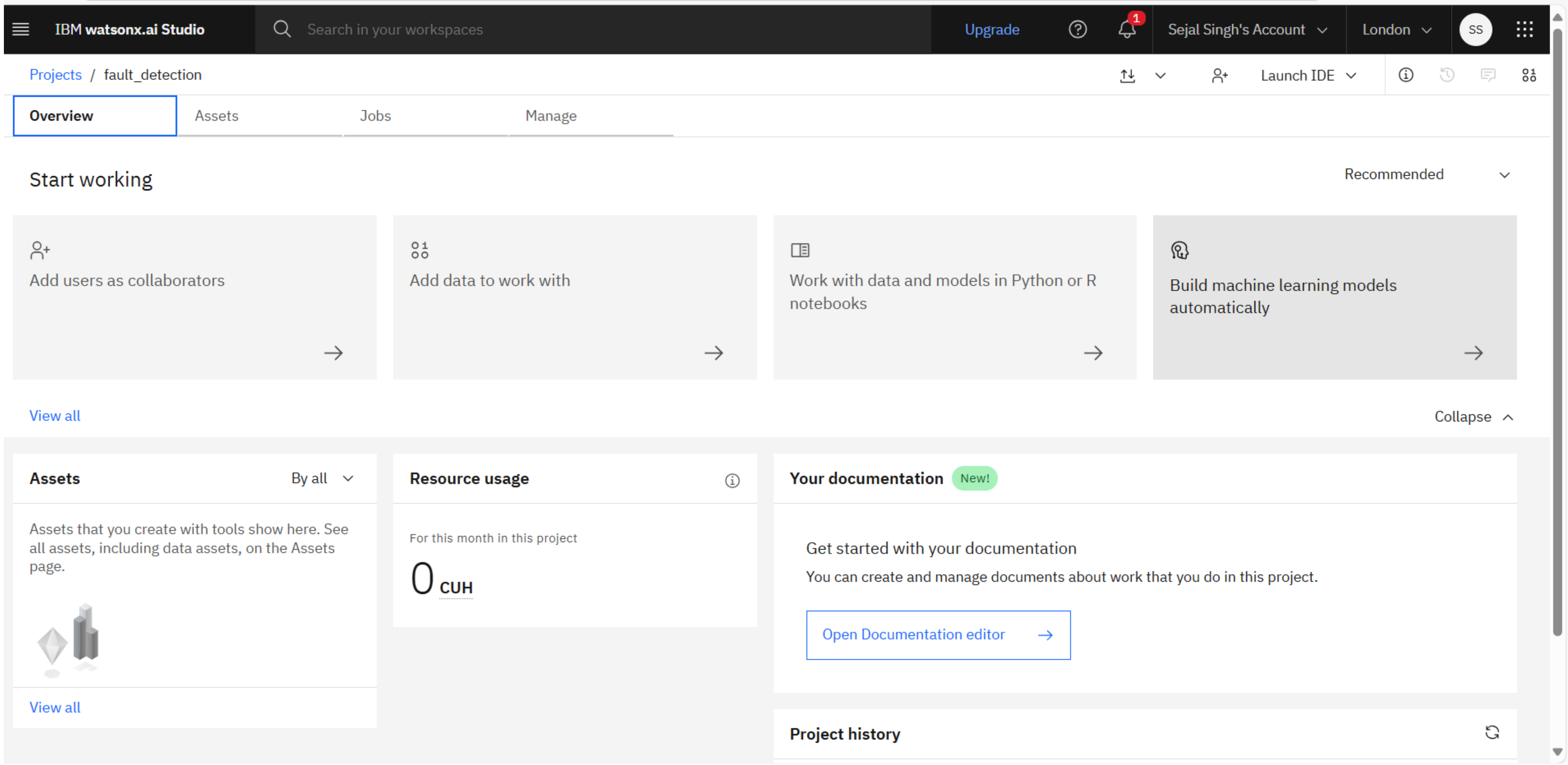
Screenshots of outputs and visual results are also provided:-

RESULT



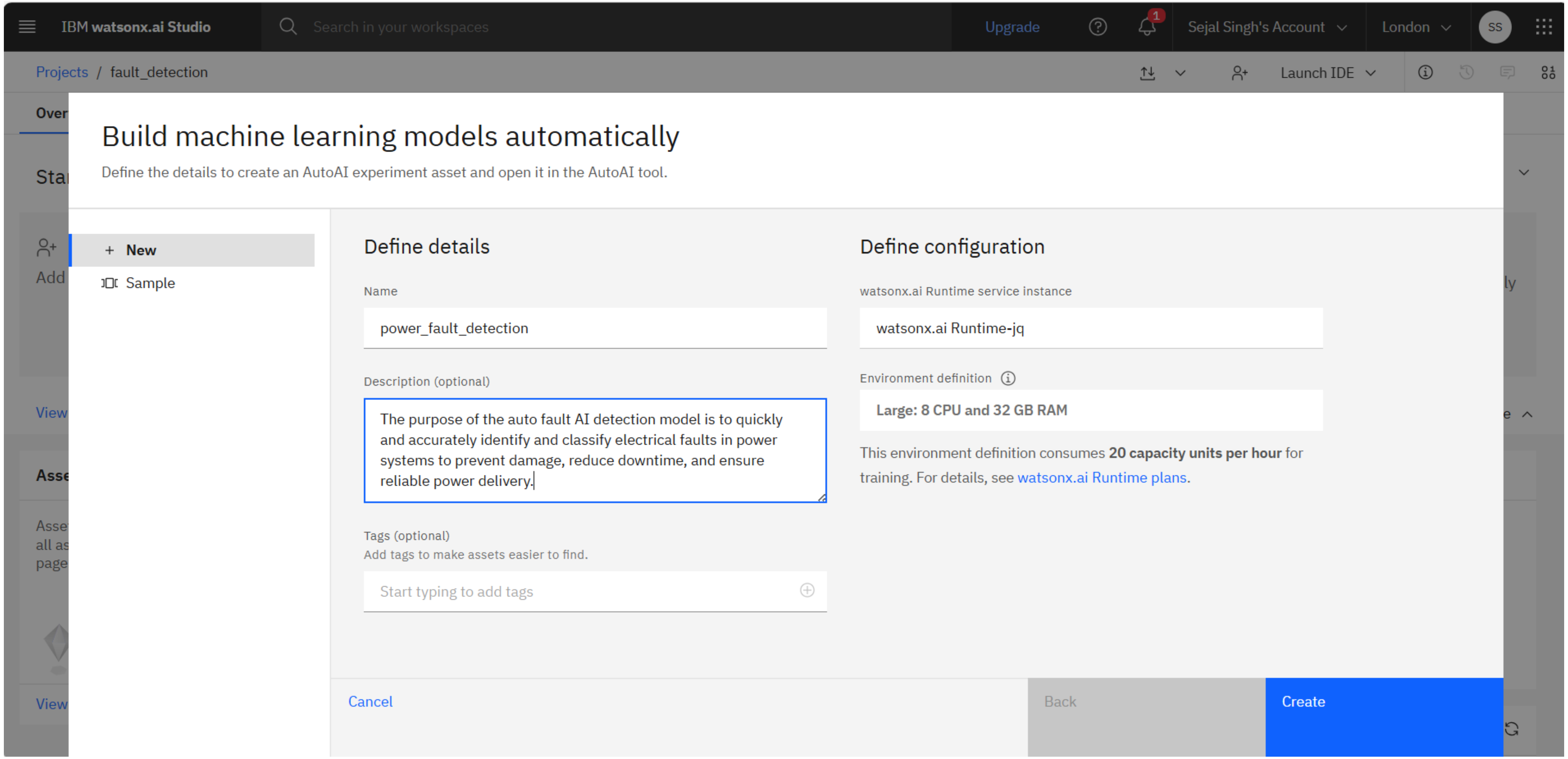
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
1
Sejal Singh's Account
London
SS
Create a project
Start with a new, blank project or select from where to import an existing project.
New
Local file
Sample
Define details
Name
fault_detection
Description (optional)
A machine learning-based model to detect and classify faults in power grids using voltage and current data.
Tags (optional)
Add tags
Add tags to make projects easier to find. To add tags, separate them with commas and press Enter.
Storage
Cloud Object Storage-eu
Cancel
Create

RESULT

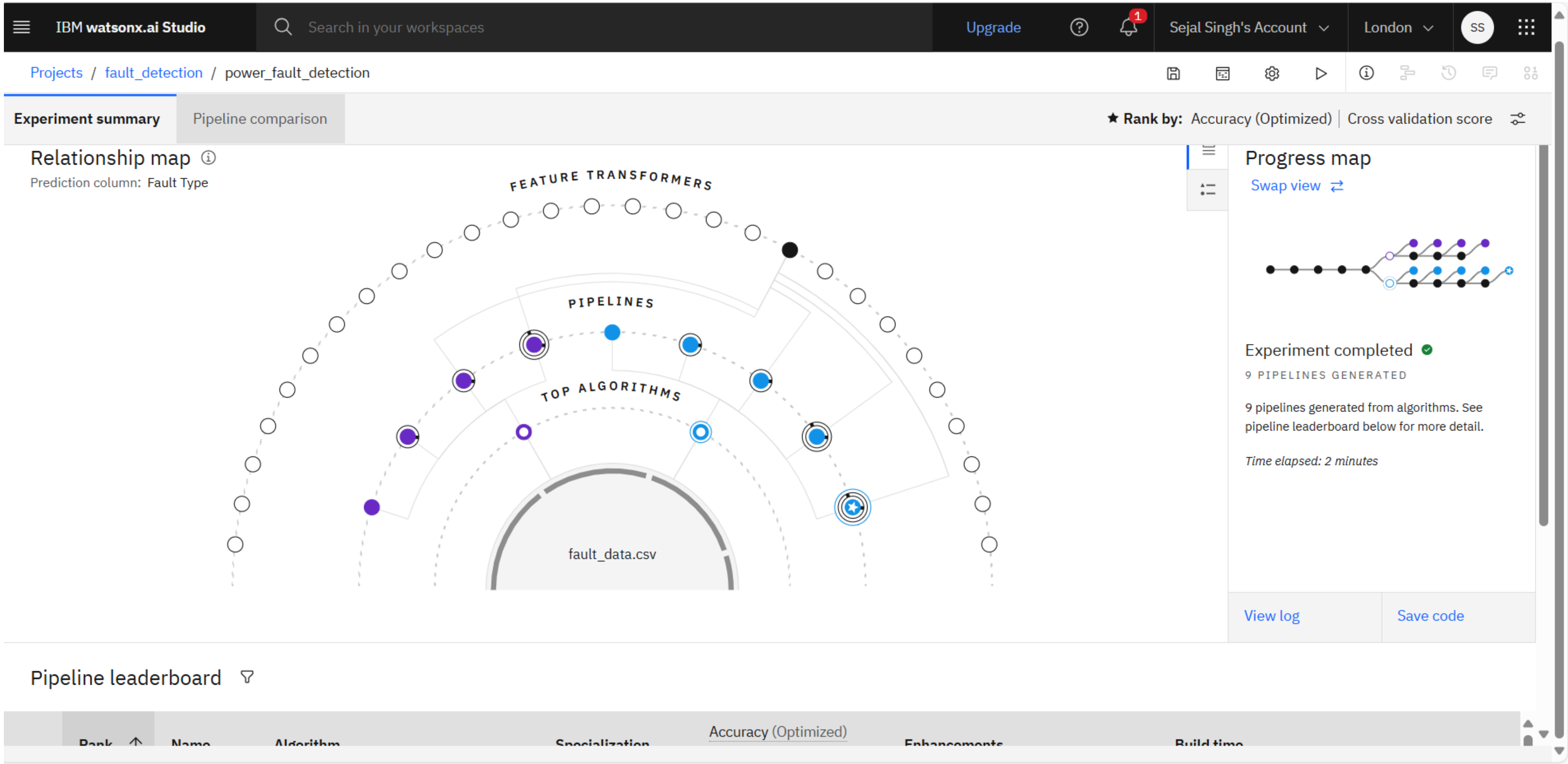


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Sejal Singh's Account
London
SS
Projects / fault_detection
Launch IDE
Overview
Assets
Jobs
Manage
Start working
Recommended
Add users as collaborators
Add data to work with
Work with data and models in Python or R notebooks
Build machine learning models automatically
View all
Collapse
Assets
By all
Assets that you create with tools show here. See all assets, including data assets, on the Assets page.
View all
Resource usage
For this month in this project
0 CUH
Your documentation
New!
Get started with your documentation
You can create and manage documents about work that you do in this project.
Open Documentation editor
Project history

RESULT

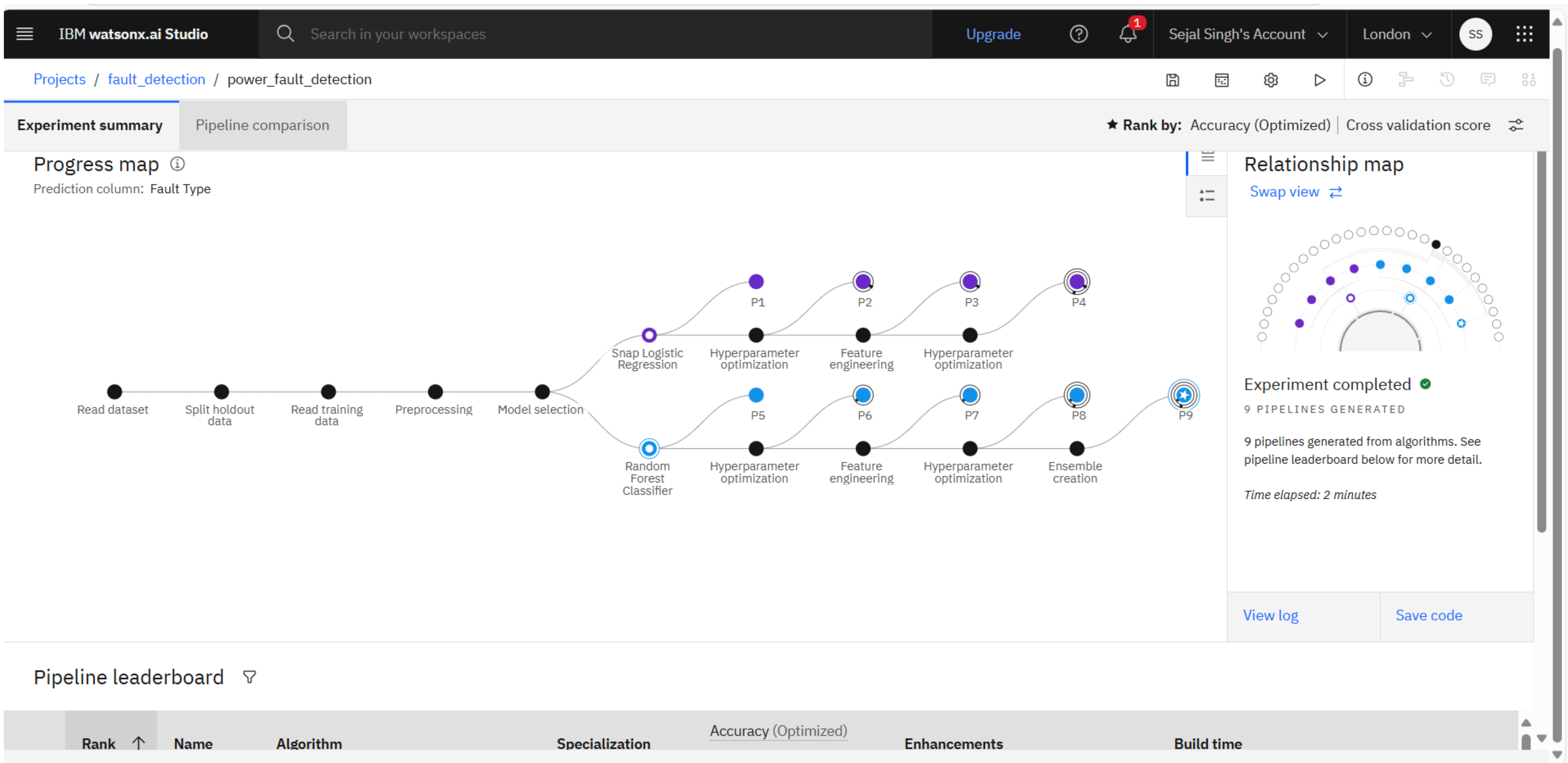

IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Sejal Singh's Account
London
SS
Projects / fault_detection
Launch IDE
Build machine learning models automatically
Define the details to create an AutoAI experiment asset and open it in the AutoAI tool.
+ New
Sample
Define details
Name
power_fault_detection
Description (optional)
The purpose of the auto fault AI detection model is to quickly and accurately identify and classify electrical faults in power systems to prevent damage, reduce downtime, and ensure reliable power delivery.
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-jq
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

RESULT

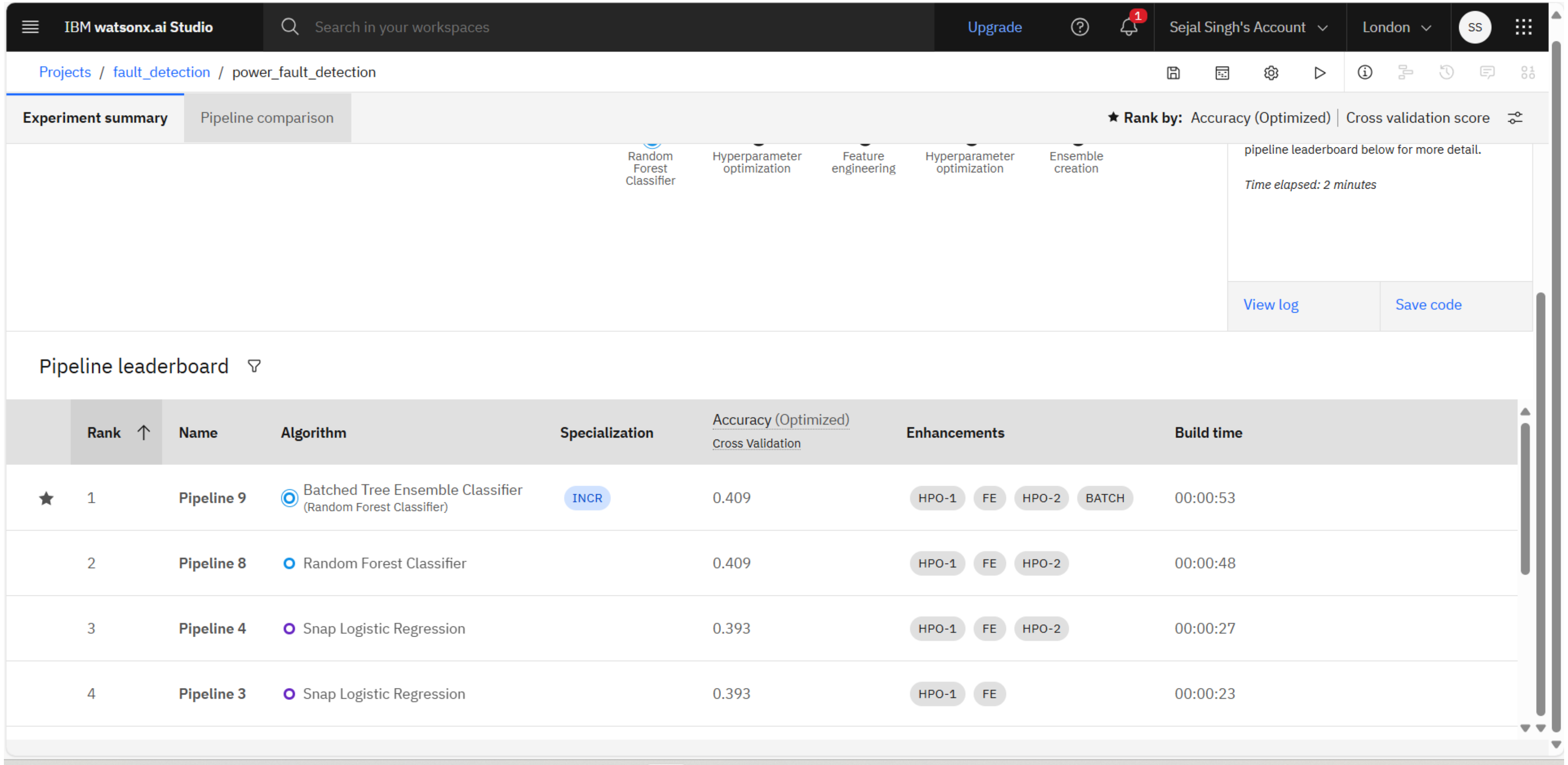


edunet foundation

RESULT

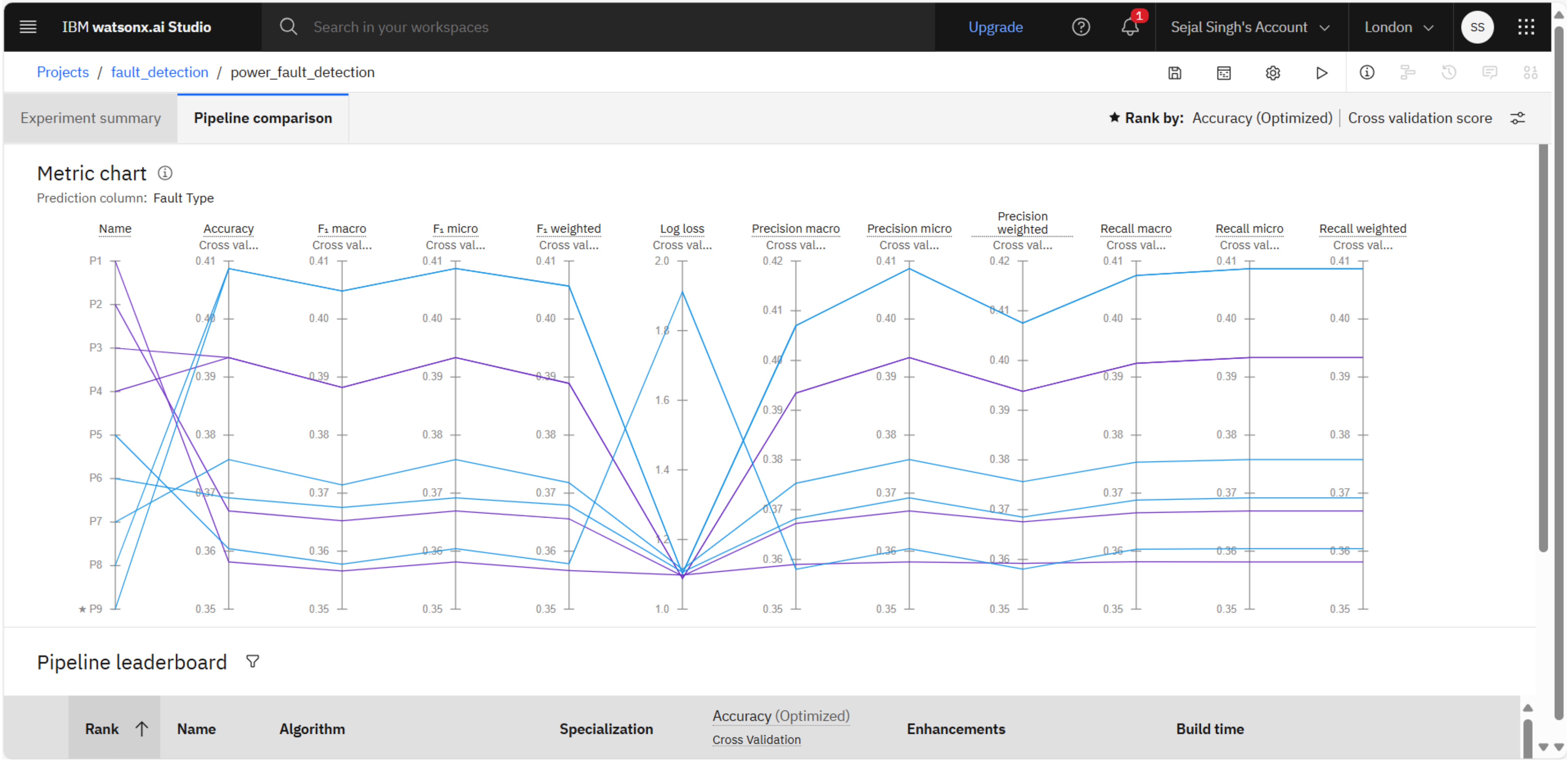

IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Sejal Singh's Account
London
SS
Projects / fault_detection / power_fault_detection
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Progress map
Prediction column: Fault Type
Read dataset
Split holdout 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
9 PIPELINES GENERATED
9 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 2 minutes
View log
Save code
Pipeline leaderboard
Rank
Name
Algorithm
Specialization
Accuracy (Optimized)
Enhancements
Build time

RESULT



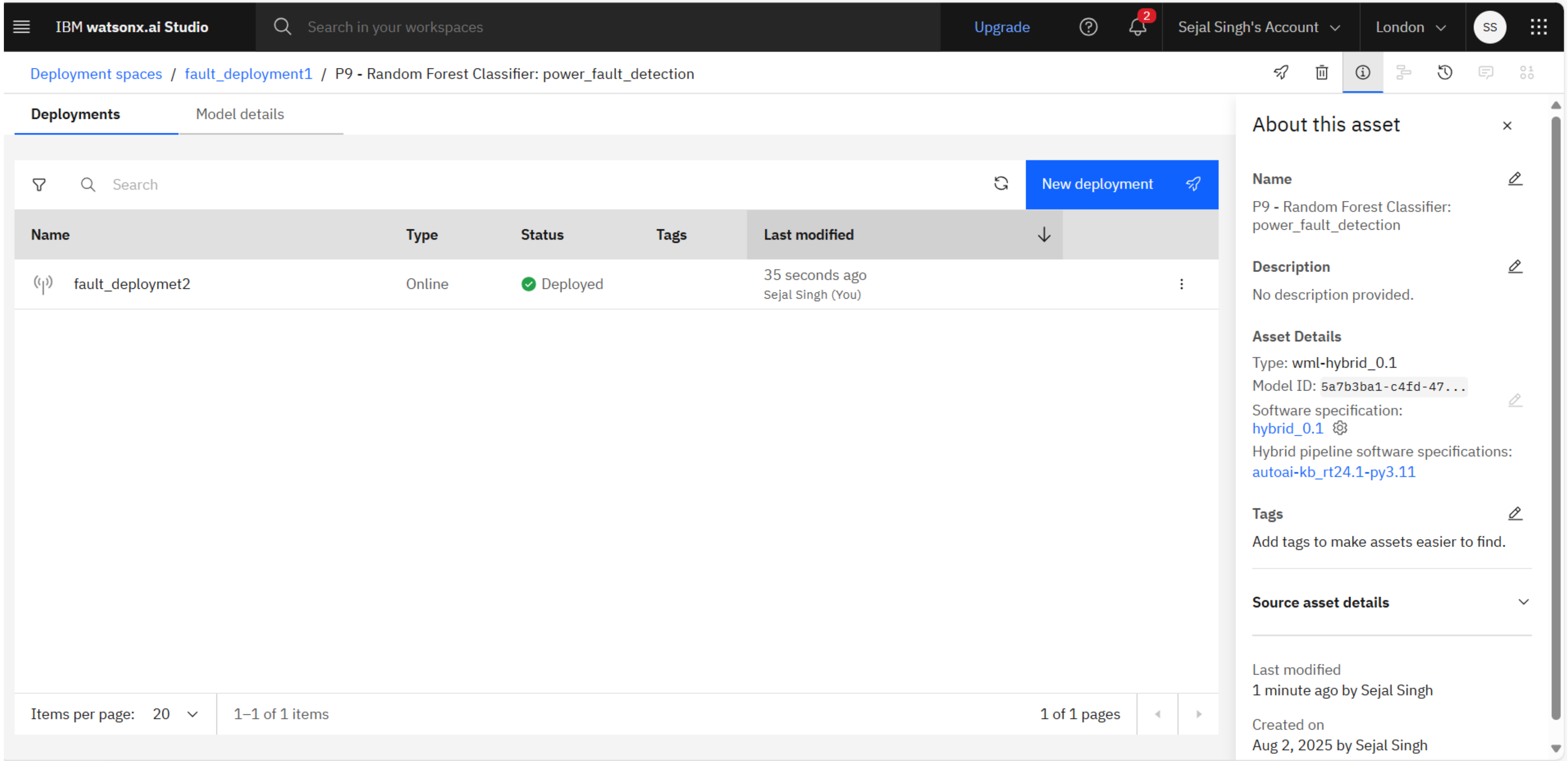
edunet foundation

RESULT



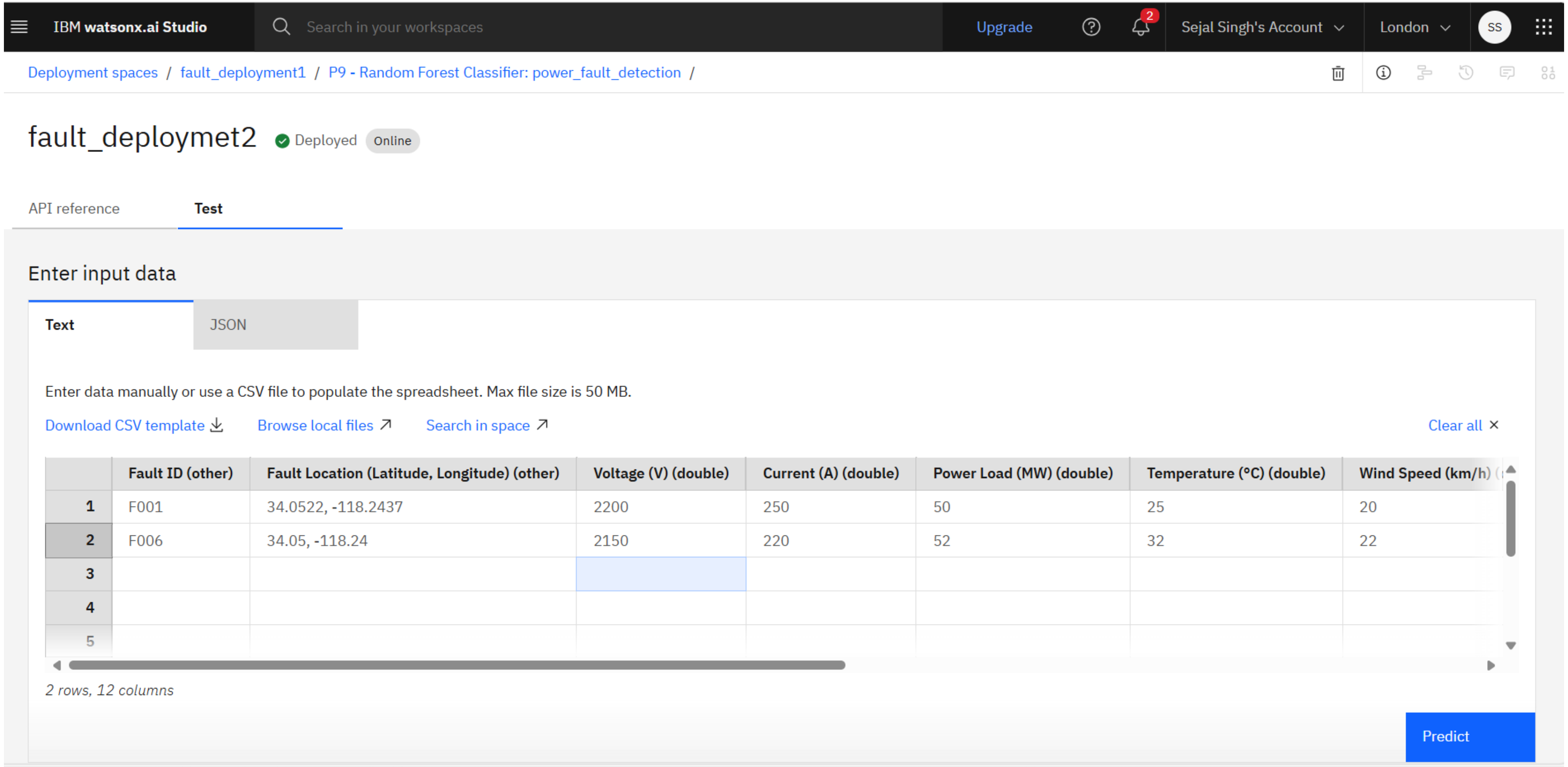
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Sejal Singh's Account
London
SS
Projects / fault_detection / power_fault_detection
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Metric chart
Prediction column: Fault Type
Name
Accuracy Cross val...
F₁ macro Cross val...
F₁ micro Cross val...
F₁ weighted Cross val...
Log loss Cross val...
Precision macro Cross val...
Precision micro Cross val...
Precision weighted Cross val...
Recall macro Cross val...
Recall micro Cross val...
Recall weighted Cross val...
P1
P2
P3
P4
P5
P6
P7
P8
P9
Pipeline leaderboard
Rank
Name
Algorithm
Specialization
Accuracy (Optimized) Cross Validation
Enhancements
Build time

RESULT



edunet foundation

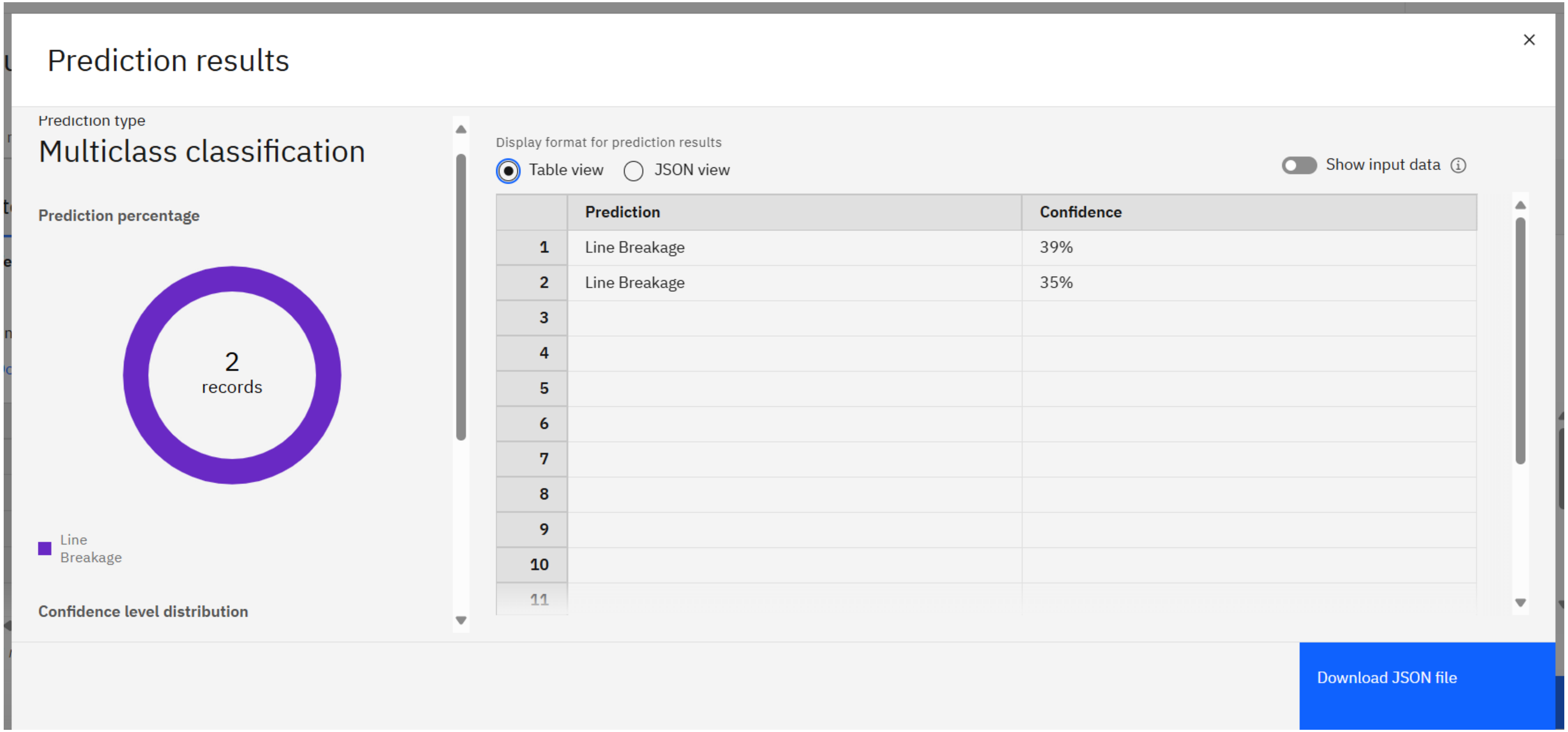
RESULT



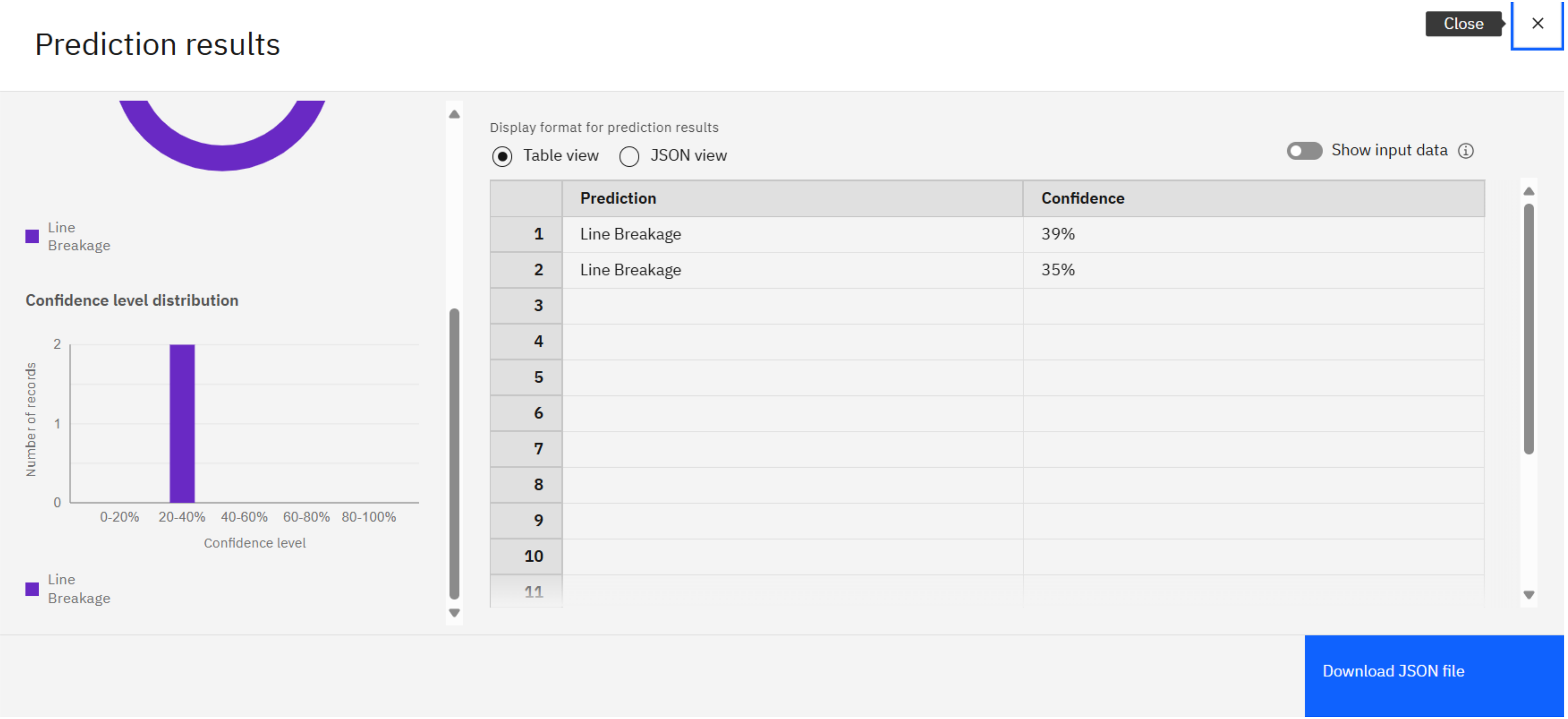
	Fault ID (other)	Fault Location (Latitude, Longitude) (other)	Voltage (V) (double)	Current (A) (double)	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h)
1	F001	34.0522, -118.2437	2200	250	50	25	20
2	F006	34.05, -118.24	2150	220	52	32	22
3							
4							
5							

edunet foundation

RESULT


Prediction results
Prediction type
Multiclass classification
Prediction percentage
2
records
Line Breakage
Confidence level distribution
Display format for prediction results
Table view
JSON view
Show input data
Prediction
Confidence
1 Line Breakage 39%
2 Line Breakage 35%
3
4
5
6
7
8
9
10
11
Download JSON file

RESULT


Prediction results
Close
Line Breakage
Confidence level distribution
2
1
0
Number of records
0-20%
20-40%
40-60%
60-80%
80-100%
Confidence level
Line Breakage
Display format for prediction results
Table view
JSON view
Show input data
Prediction
Confidence
1
Line Breakage
39%
2
Line Breakage
35%
3
4
5
6
7
8
9
10
11
Download JSON file

CONCLUSION

- The proposed machine learning model effectively detects and classifies different types of power system faults with high accuracy.
  It helps in ensuring timely fault identification, reducing downtime, and maintaining power grid stability.
  During implementation, challenges like data preprocessing and feature selection were encountered.
  In the future, the model can be improved by adding more real-time data and using deep learning for better accuracy.
  Accurate fault detection is crucial for preventing damage and ensuring reliable power supply in modern power systems.

FUTURE SCOPE

- The system can be improved by incorporating additional data sources such as real-time grid sensor data, satellite weather data, and smart meter readings.
  The algorithm can be further optimized using advanced techniques like deep learning or ensemble models for better accuracy and speed.
- The system can be scaled to monitor faults across multiple cities or regions for wider coverage.
  Integration with **edge computing** can enable faster, on-site fault detection, and reduce response time.
- Using emerging technologies like **Auto ML** and **AI-based predictive maintenance** can make the system more intelligent and reliable.

REFERENCES

- This project was completed based on the concepts taught during mentor-led sessions.
- No external research papers or articles were referred.
- Guidance and support were provided by **IBM Edunet mentors**:
  - **Mr. Narendra Eluri**
  - **Mr. Tarun Sharma**

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM

Sejal Singh

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 17, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/2961d702-2448-45a7-bbaa-82005bc6865f

IBM

IBM CERTIFICATIONS


In recognition of the commitment to achieve professional excellence

Sejal Singh

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/eb35f869-65b4-4854-bce2-9e0651a1e89f

IBM

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Sejal Singh

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