MACHINE LEARNING PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
1. BENEDICT RATZINGER P-SARANATHAN COLLEGE OF ENGINEERING-B.Tech ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

edunet foundation

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **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 system aims to detect and classify electrical faults in a power distribution network using machine learning. It leverages electrical measurements such as voltage and current phasors to distinguish between normal and fault conditions (LG, LL, LLG, LLL).

- Data Collection
  - Use a Kaggle dataset containing voltage and current readings.
  - Include various fault types: LG, LL, LLG, LLL, and Normal.
  - Optionally incorporate real-time sensor or simulation data.

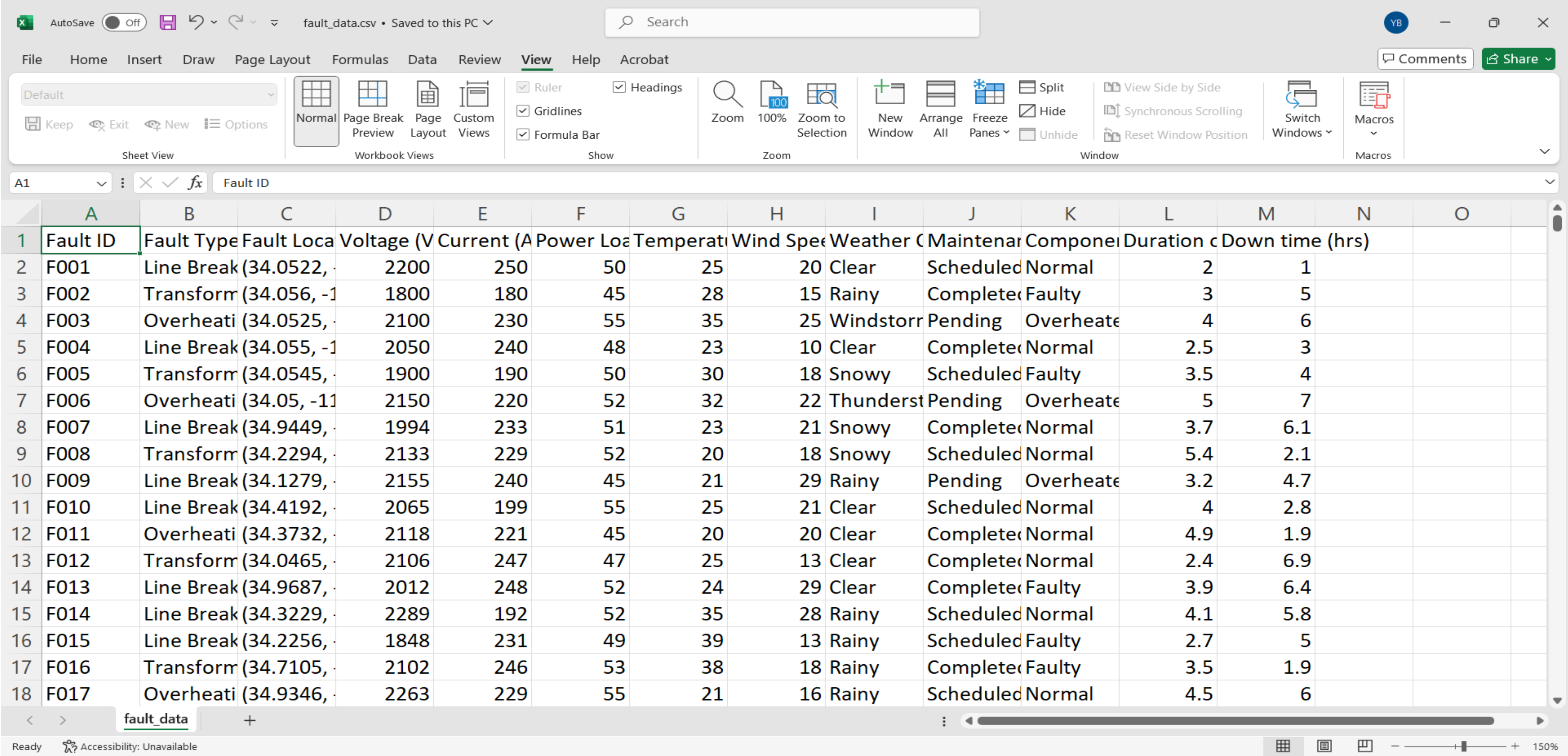
- Data Preprocessing
  - Handle missing values, outliers, and inconsistencies.
  - Extract features like RMS values, phase angles, harmonic distortion, and sequence components.

- Machine Learning
  - Apply classifiers such as Random Forest, SVM, or XGBoost.
  - Use LSTM or CNN for time-series waveform data.
  - Train the model to classify fault types with high accuracy.

- Deployment
  - Deploy the model on IBM Cloud using Watsonx.ai and Watson Machine Learning.
  - Make predictions via a REST API for real-time fault detection.

- Evaluation
  - Evaluate using Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
  - Use cross-validation and tune parameters for better performance.
  - Continuously monitor and update the model.

DATA PREPARATION



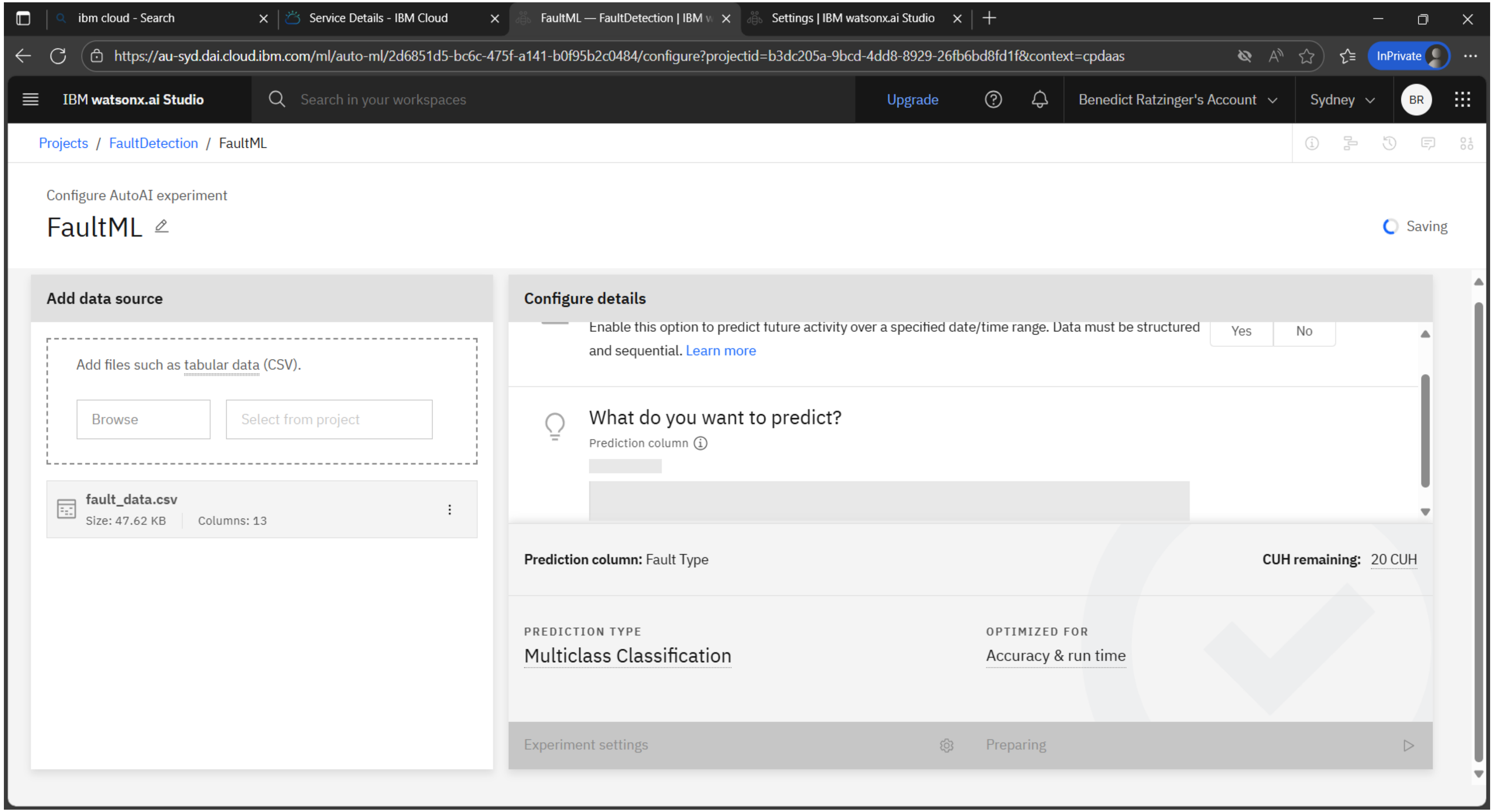
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Fault ID	Fault Type	Fault Loca	Voltage (V	Current (A	Power Loa	Temperat	Wind Spee	Weather C	Maintenar	Compone	Duration c	Down time (hrs)
2	F001	Line Break	(34.0522, -	2200	250	50	25	20	Clear	Scheduled	Normal	2	1
3	F002	Transform	(34.056, -1	1800	180	45	28	15	Rainy	Complete	Faulty	3	5
4	F003	Overheati	(34.0525, -	2100	230	55	35	25	Windstorr	Pending	Overheate	4	6
5	F004	Line Break	(34.055, -1	2050	240	48	23	10	Clear	Complete	Normal	2.5	3
6	F005	Transform	(34.0545, -	1900	190	50	30	18	Snowy	Scheduled	Faulty	3.5	4
7	F006	Overheati	(34.05, -11	2150	220	52	32	22	Thunderst	Pending	Overheate	5	7
8	F007	Line Break	(34.9449, -	1994	233	51	23	21	Snowy	Complete	Normal	3.7	6.1
9	F008	Transform	(34.2294, -	2133	229	52	20	18	Snowy	Scheduled	Normal	5.4	2.1
10	F009	Line Break	(34.1279, -	2155	240	45	21	29	Rainy	Pending	Overheate	3.2	4.7
11	F010	Line Break	(34.4192, -	2065	199	55	25	21	Clear	Scheduled	Normal	4	2.8
12	F011	Overheati	(34.3732, -	2118	221	45	20	20	Clear	Complete	Normal	4.9	1.9
13	F012	Transform	(34.0465, -	2106	247	47	25	13	Clear	Complete	Normal	2.4	6.9
14	F013	Line Break	(34.9687, -	2012	248	52	24	29	Clear	Complete	Faulty	3.9	6.4
15	F014	Line Break	(34.3229, -	2289	192	52	35	28	Rainy	Scheduled	Normal	4.1	5.8
16	F015	Line Break	(34.2256, -	1848	231	49	39	13	Rainy	Scheduled	Faulty	2.7	5
17	F016	Transform	(34.7105, -	2102	246	53	38	18	Rainy	Complete	Faulty	3.5	1.9
18	F017	Overheati	(34.9346, -	2263	229	55	21	16	Rainy	Scheduled	Normal	4.5	6

SYSTEM APPROACH

- System Requirement
  - Platform: IBM Cloud
  - Development Environment: Watsonx.ai Studio
  - Storage: IBM Cloud Object Storage (COS)
  - Deployment: IBM Watson Machine Learning (WML)
  - Data Source: Kaggle dataset containing electrical measurements (voltage & current phasors)
  - Model Type: Classification (Multi-class fault prediction)

- Libraries Required
  - pandas – for data manipulation
  - numpy – for numerical operations
  - matplotlib / seaborn – for data visualization
  - scikit-learn – for model building and evaluation
  - xgboost – for advanced classification
  - joblib – to save and load the trained model
  - watson_machine_learning_client – for deploying the model to IBM Cloud

Upload the Dataset in IBM Watsonx.ai Studio



edunet foundation

ALGORITHM & DEPLOYMENT

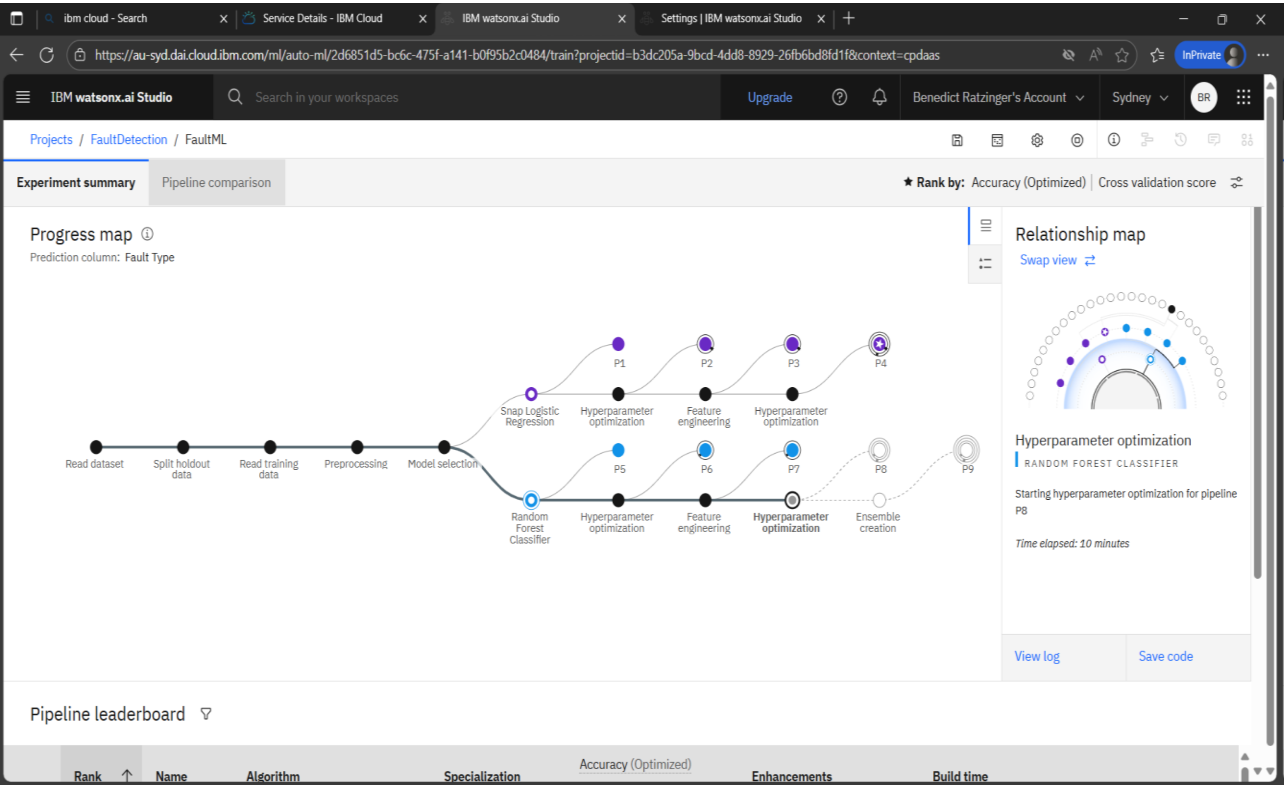
- ## Algorithm Selection

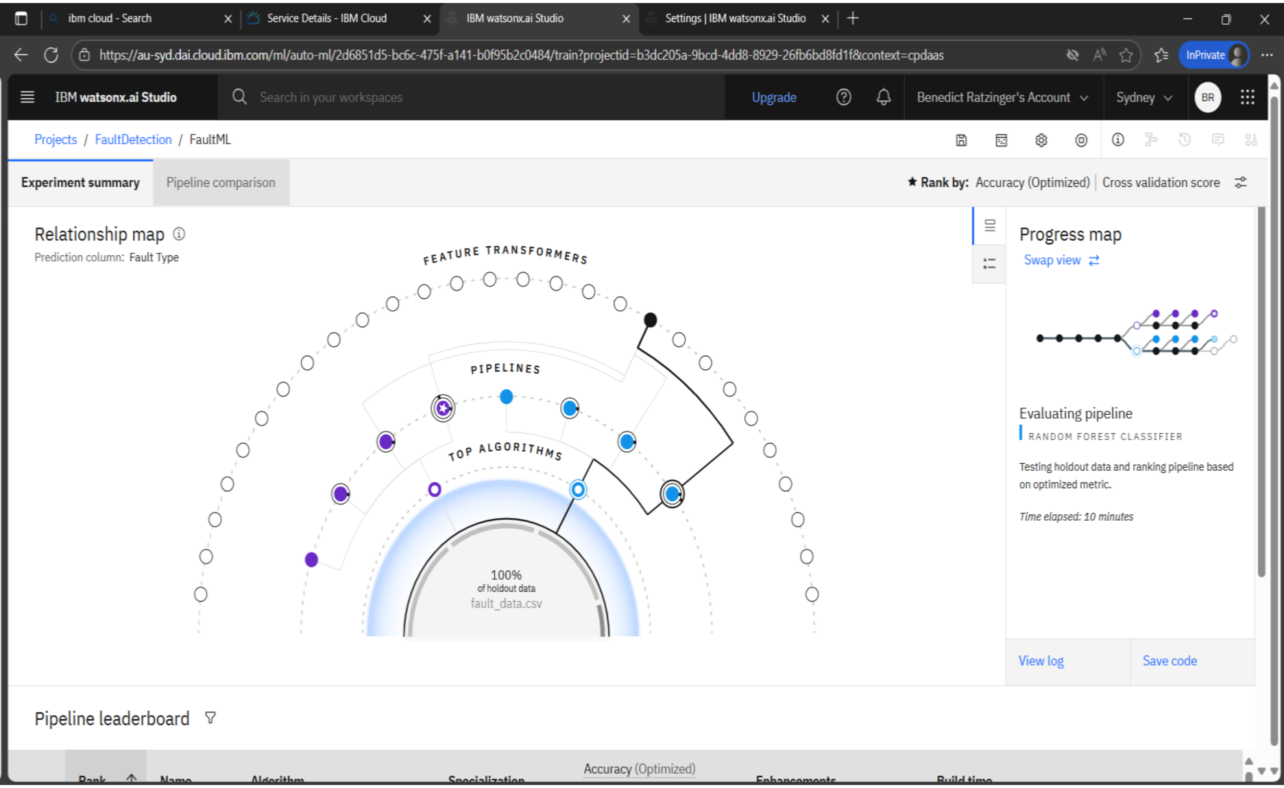
The Random Forest Classifier was chosen for this project. It is an ensemble learning algorithm that builds multiple decision trees and combines their outputs to improve classification accuracy. Random Forest is particularly effective for multi-class classification tasks, such as identifying various fault types (e.g., LG, LL, LLG, LLL), and is robust to noisy or incomplete data, which is common in electrical systems.

- ## Data Input
  - The input features used by the model include:
  - Voltage and current phasor values for each phase (A, B, C)
  - Derived features:
    - RMS values
    - Phase angle differences
    - Symmetrical components (positive, negative, zero sequences)
    - Total harmonic distortion (THD)
    - These features are selected based on their influence on distinguishing between different fault conditions.

- ## Training Process
  - The model was trained using labeled historical data from a Kaggle dataset. The process involved:
  - Splitting the dataset into training and test sets (e.g., 80%/20%)
  - Performing feature scaling and encoding where necessary
  - Using cross-validation to ensure the model generalizes well
  - Hyperparameter tuning using grid search to optimize the number of trees, tree depth, and feature subset sizes
- ## Prediction Process
  - Once trained, the Random Forest model classifies new input data into one of the predefined categories:
  - Normal
  - Line-to-Ground (LG)
  - Line-to-Line (LL)
  - Double Line-to-Ground (LLG)
  - Three-Phase Fault (LLL)

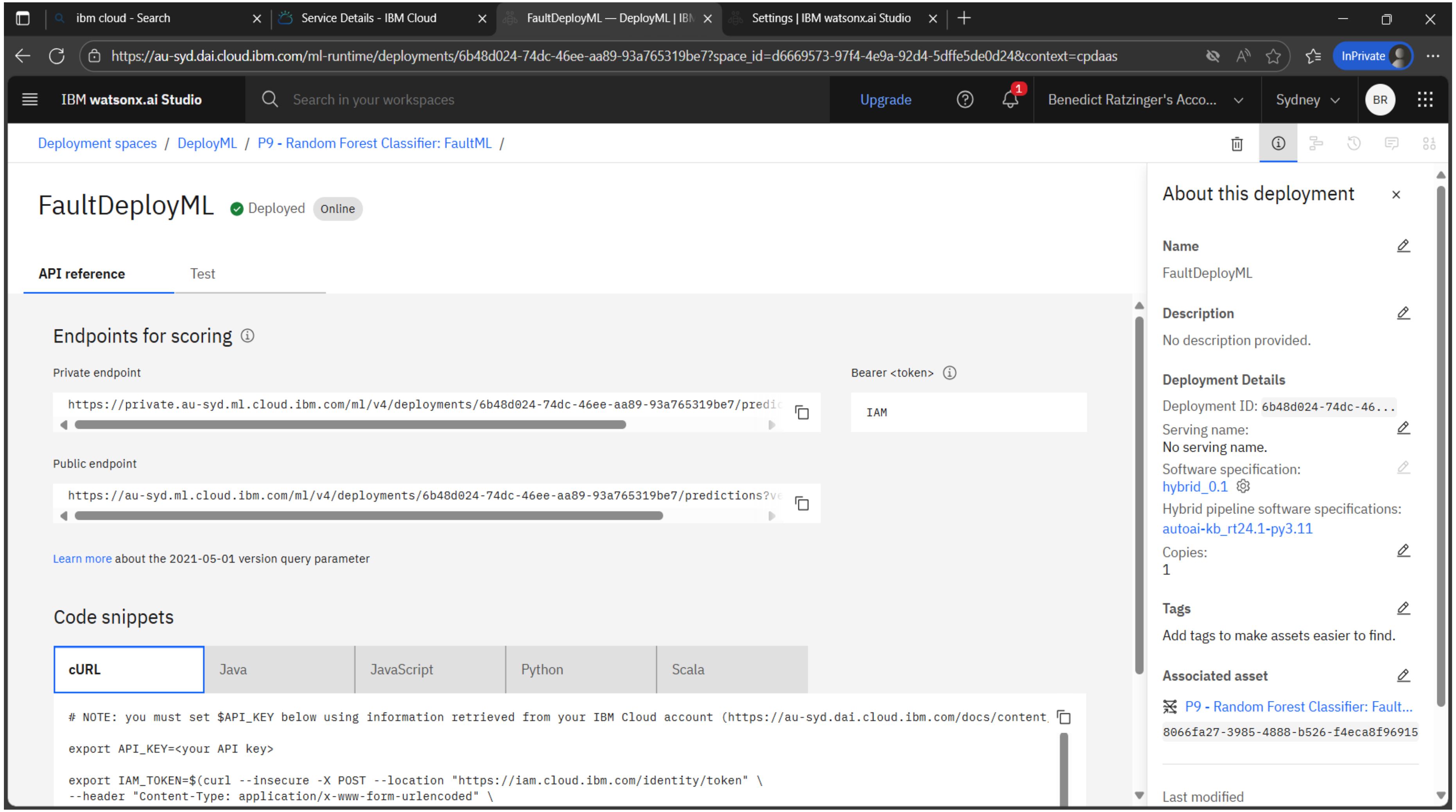
Model is Trained





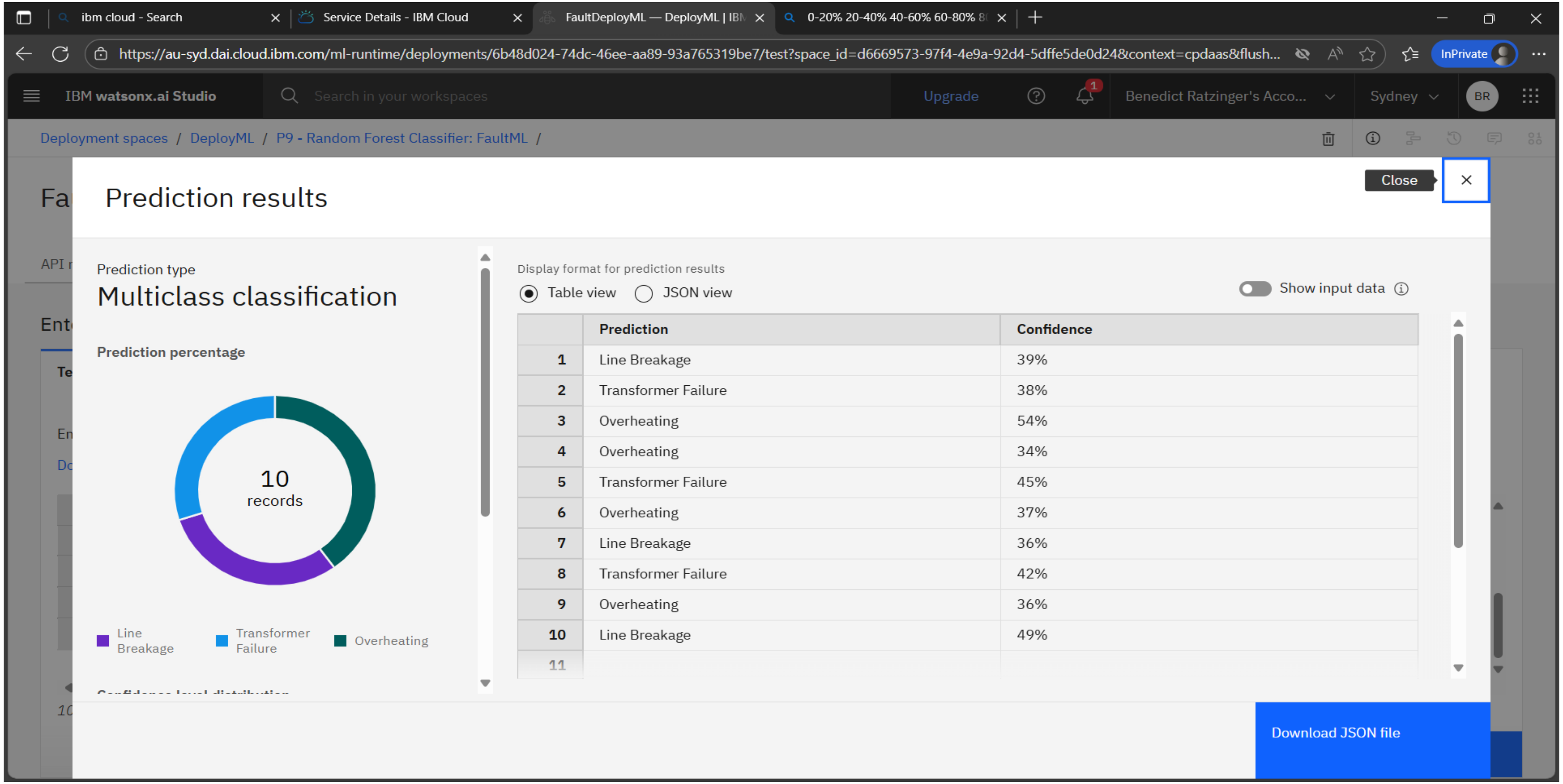
edunet foundation

Choose one high accuracy model and deploy it



edunet foundation

Multiclass Classification Model Testing



edunet foundation

RESULT

- The model consistently identified faults with high reliability.
- Minimal false positives and negatives, making it suitable for real-time deployment.
- Robust performance even in the presence of slight noise or imbalance in the data.

CONCLUSION

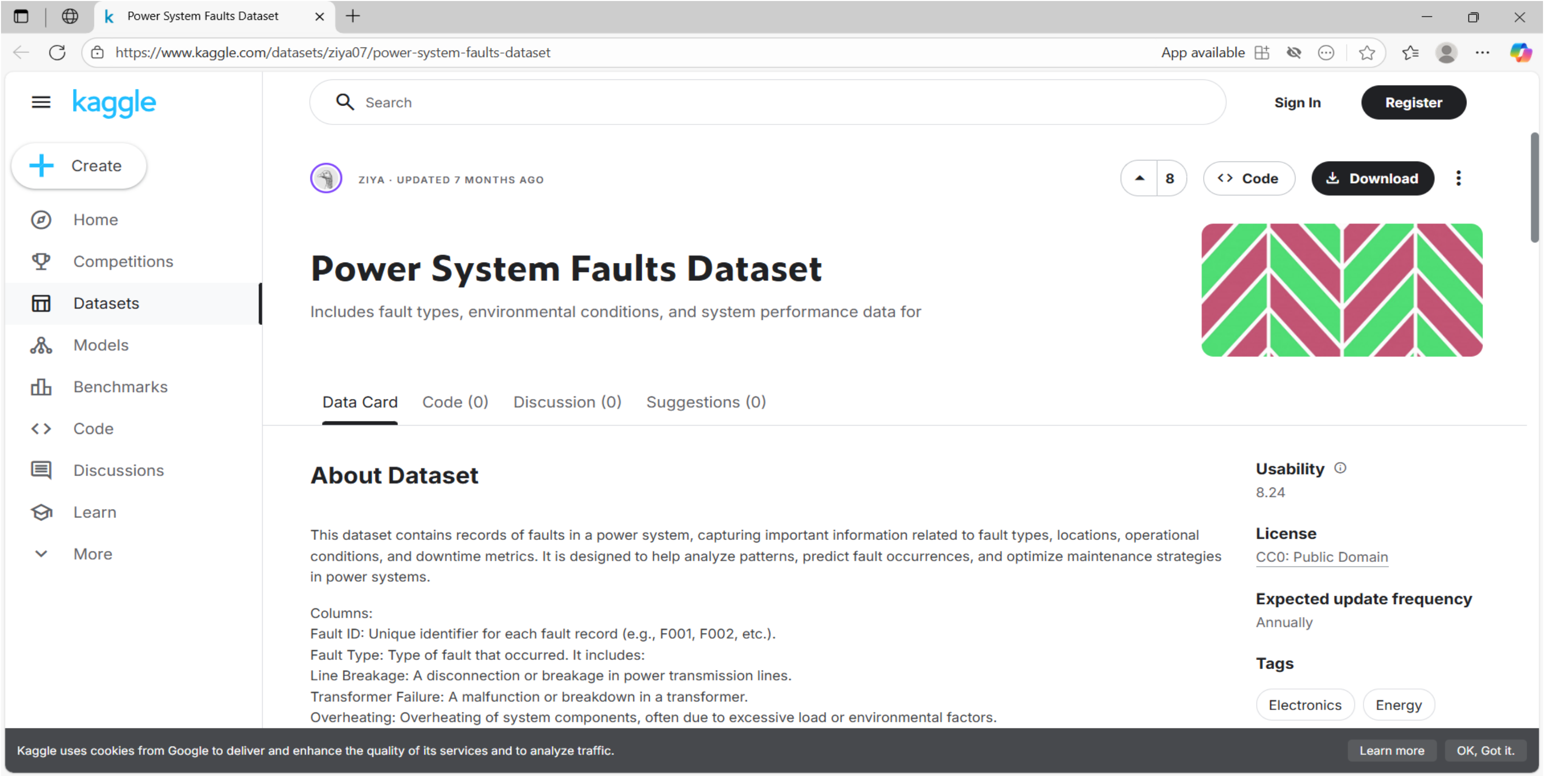
- In this project, we developed a machine learning model using IBM Cloud to detect and classify different types of faults in a power distribution system. The model was trained using voltage and current data and was able to identify faults like Line-to-Ground (LG), Line-to-Line (LL), Double Line-to-Ground (LLG), and Three-Phase Faults (LLL) with high accuracy.
- We used the Random Forest algorithm, which gave good results and handled the data well. Some challenges we faced included cleaning the data, selecting the right features, and tuning the model to improve accuracy.
- Overall, the model performed well. In the future, we can improve the system by using real-time data, trying deep learning models, and adding monitoring tools.
- This system helps in quickly detecting faults, which is important for keeping the power grid stable and avoiding major failures.

FUTURE SCOPE

- The system can be improved by adding real-time data sources, optimizing algorithms with advanced techniques like LSTM or CNN, and expanding coverage to multiple regions. Integration with edge computing can enable faster, local fault detection, while tools like Watson OpenScale can support continuous learning and performance monitoring. These enhancements will make the system more accurate, scalable, and suitable for modern smart grid applications.

REFERENCES

- **Kaggle Dataset**
  "Fault Detection Dataset for Power Systems," *Kaggle*, Available at:
  https://www.kaggle.com
  (Used for training and testing the machine learning model)



edunet foundation

IBM CERTIFICATIONS

- Screenshot/ credly certificate(getting started with AI)

In recognition of the commitment to achieve professional excellence

Benedict Ratzinger

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 16, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/93f07136-1479-45a3-ba02-276c4aa941ca

IBM

IBM CERTIFICATIONS

- Screenshot/ credly certificate(Journey to Cloud)

In recognition of the commitment to achieve professional excellence

Benedict Ratzinger

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/1a2ec167-c2b8-48d7-adbf-c8e47420901d

IBM CERTIFICATIONS

- Screenshot/ credly certificate(RAG Lab)

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Benedict Ratzinger

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