CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION USING MACHINE LEARNING

Presented By:
1. JAYASRI K– COLLEGE OF ENGINEERING GUINDY,ANNA UNIVERSITY- B>TECH IT

OUTLINE

- **Problem Statement** (Should not include solution)
- **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

- Develop a machine learning model that classifies power system faults using the dataset provided . The model will process electrical measurements to identify the type of fault rapidly and accurately. This classification will help to automate fault detection and assist in rapid recovery actions, ensuring system reliability.
- KEY COMPONENTS:
- Data Collection:
  - Use the dataset for power system faults provided on Kaggle
- Data Preprocessing:
  - Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.
- Machine Learning Algorithm:
  - Implement a machine learning algorithm, such as a Decision Tree , Snap Logistic Regression, Random Forest , or SVM
- Deployment:
  - Deploy the solution on a scalable and reliable platform, considering factors like server infrastructure, response time and accessibility.
- Evaluation:
  - Assess the model's performance using appropriate metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or other relevant metrics.
  - Fine-tune the model based on feedback and continuous monitoring of prediction accuracy.
  - Validate the model using accuracy , precision, recall and F1-Score

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the rental bike prediction system. Here's a suggested structure for this section:

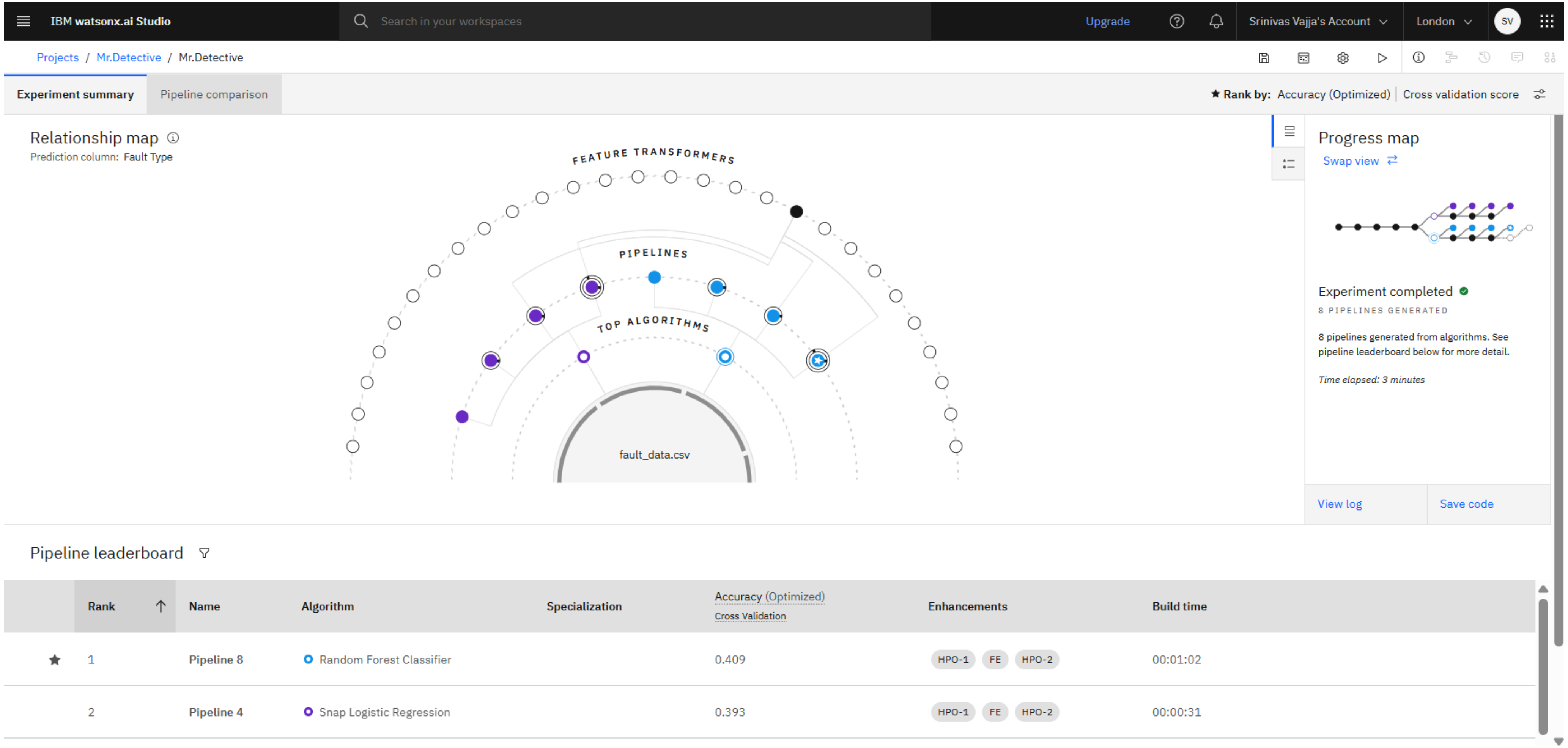
- System requirements:
  - IBM Cloud(Mandatory)
  - IBM Watsonx.ai Studio for development and deployment
  - IBM Cloud Object Storage for data feeding and dataset handling

ALGORITHM & DEPLOYMENT

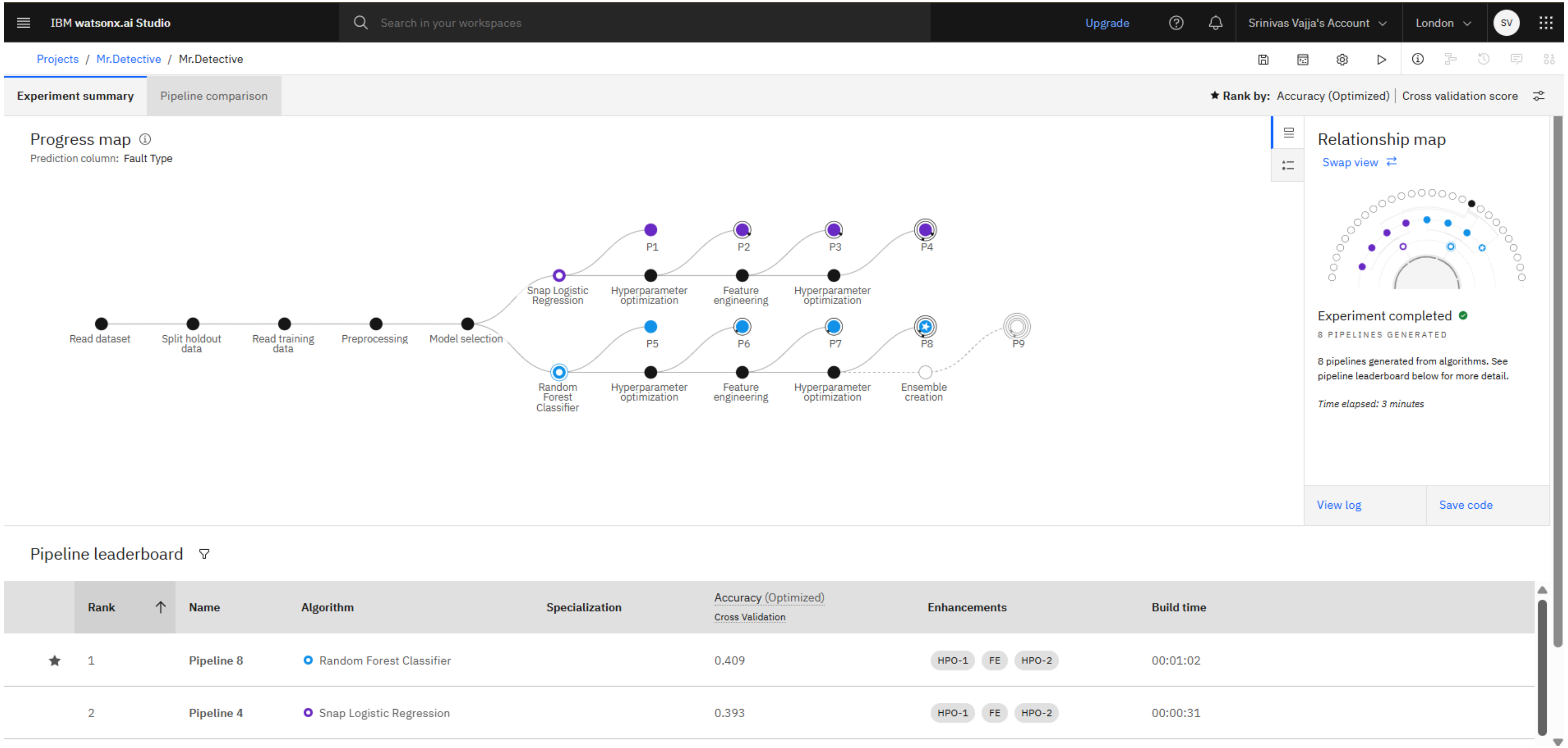
- In the Algorithm section, describe the machine learning algorithm chosen for predicting the types of Faults. Here's an example structure for this section:
- **Algorithm Selection:**
  - Random Forest Classifier (or SVM based on Performance)
- **Data Input:**
  - Voltage ,Current , Power Load ,Temperature ,Wind Speed ,Weather Condition ,Maintenance Status ,Component Health, Duration of Fault
- **Training Process:**
  - Supervised Learning (Using Labelled Fault-Types)
- **Prediction Process:**
  - Model was deployed on IBM Watsonx.ai Studio and fed data using Cloud Storage Object with API endpoints for real-time predictions on the Fault-Types
  - We provide the model with required data for the prediction , once the data is sufficient we can then wait for the model to predict the Fault-Types

edunet
foundation

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Srinivas Vajja's Account
London
SV
Projects / Mr.Detective / Mr.Detective
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Relationship map
Prediction column: Fault Type
FEATURE TRANSFORMERS
PIPELINES
TOP ALGORITHMS
fault_data.csv
Progress map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 3 minutes
View log
Save code
Pipeline leaderboard
Rank
Name
Algorithm
Specialization
Accuracy (Optimized)
Cross Validation
Enhancements
Build time
1 Pipeline 8 Random Forest Classifier 0.409 HPO-1 FE HPO-2 00:01:02
2 Pipeline 4 Snap Logistic Regression 0.393 HPO-1 FE HPO-2 00:00:31

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Srinivas Vajja's Account
London
SV
Projects / Mr.Detective / Mr.Detective
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
Hyperparameter optimization
Ensemble creation
P1
P2
P3
P4
P5
P6
P7
P8
P9
Relationship map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 3 minutes
View log
Save code
Pipeline leaderboard
Rank
Name
Algorithm
Specialization
Accuracy (Optimized)
Cross Validation
Enhancements
Build time
1
Pipeline 8
Random Forest Classifier
0.409
HPO-1
FE
HPO-2
00:01:02
2
Pipeline 4
Snap Logistic Regression
0.393
HPO-1
FE
HPO-2
00:00:31

RESULT

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Srinivas Vajja's Account

London

SV

Deployment spaces / Fault_Detection / Mr.Detective /

Fault_Detector

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

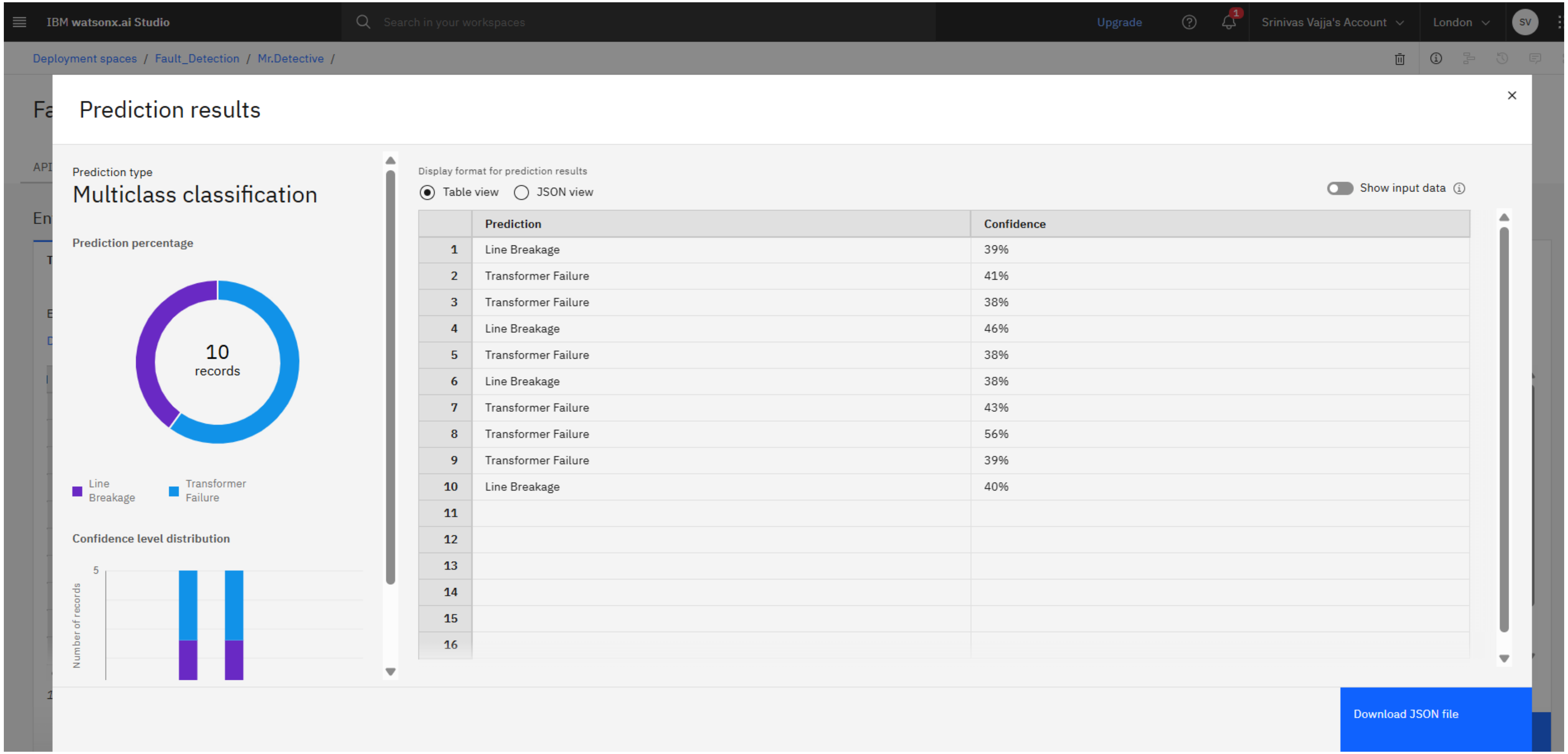
	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h) (double)	Weather Condition (other)	Maintenance Status (other)	Component Health (other)	Duration of Fault (hrs) (double)	Down time (hrs) (double)
1	50	25	20	clear	scheduled	normal	2	1
2	45	21	29	rainy	pending	overheated	3.2	4.7
3	50	30	18	snowy	scheduled	faulty	3.5	4
4	52	24	29	clear	completed	faulty	3.9	6.4
5	52	31	16	thunderstrom	completed	faulty	6	3.7
6	54	33	16	snowy	scheduled	faulty	3.2	6.4
7	49	36	13	thunderstorm	completed	normal	4.6	2.8
8	51	21	30	clear	completed	overheated	4.9	6.4
9	48	38	15	clear	pending	faulty	2.4	6.5
10	54	37	17	snowy	completed	normal	5.7	1.5

6.4

10 rows, 12 columns

Predict

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Srinivas Vajja's Account
London
SV
Deployment spaces / Fault_Detection / Mr.Detective /
Prediction results
Prediction type
Multiclass classification
Prediction percentage
10
records
Line Breakage
Transformer Failure
Confidence level distribution
5
Number of records
Display format for prediction results
Table view
JSON view
Show input data
Prediction
Confidence
1 Line Breakage 39%
2 Transformer Failure 41%
3 Transformer Failure 38%
4 Line Breakage 46%
5 Transformer Failure 38%
6 Line Breakage 38%
7 Transformer Failure 43%
8 Transformer Failure 56%
9 Transformer Failure 39%
10 Line Breakage 40%
11
12
13
14
15
16
Download JSON file

CONCLUSION

- The development of a **machine learning-based power system fault classification model** has demonstrated that data-driven approaches can significantly improve the speed and accuracy of fault detection in electrical grids. By leveraging a **Random Forest Classifier**, the system efficiently processed diverse input features—such as voltage, current, power load, environmental conditions, and component health—to identify various fault types. Deployment on **IBM Watsonx.ai Studio** provided a scalable and reliable infrastructure for real-time predictions, ensuring quick and informed decision-making. This model not only reduces downtime but also supports utilities in maintaining system reliability, safety, and operational efficiency.

FUTURE SCOPE

- In the future, this model can be expanded to incorporate real-time data from IoT-enabled devices, such as smart meters and phasor measurement units, enabling more responsive and accurate fault detection across wider grid networks. Beyond classification, the system could evolve into a predictive maintenance tool, forecasting potential component failures before they occur and allowing utilities to take proactive measures. Advanced deep learning techniques, such as LSTM networks or Graph Neural Networks, may further enhance the model’s ability to capture temporal patterns and grid topology for even greater accuracy. Additionally, integrating continuous learning pipelines will allow the model to adapt automatically as new fault data becomes available, ensuring it stays relevant in changing grid conditions

REFERENCES

- Dataset:- https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset
- IBM Cloud:- https://www.ibm.com/cloud
- Fault-Detection using ML:- https://ieeexplore.ieee.org/document/10306740
- Fault-Detection using ML:- https://www.sciencedirect.com/science/article/pii/S2352484724007807

edunet foundation

GITHUB LINK

https://github.com/jayasrikamaraj/AICTE-ML-Project.git

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Jayasri K

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/82824ba5-9600-4b0a-b4f0-7106dd243dd6

IBM

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM

Jayasri K

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 19, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/a20fff39-0cdd-4854-8a65-ea83066330ea

IBM

edunet
foundation

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Jayasri K

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 25 Jul 2025 (GMT)

Learning hours: 20 mins

edunet foundation

THANK YOU