CAPSTONE PROJECT

Power Fault Detection Using ML

Presented By:
1. Shreyash Sunil Badve
-MIT Acadamy Of Engineering-ETX

edunet foundation

OUTLINE

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

Problem Statement

The reliability of power distribution systems is critically affected by faults such as line breakages, transformer failures, and equipment overheating. Traditional fault detection methods are often slow and reactive, leading to increased downtime and compromised grid stability. The objective of this project is to develop a machine learning-based model that can accurately detect and classify different types of faults using real-time electrical and environmental data. By distinguishing between normal operating conditions and specific fault types, the model aims to enable rapid fault identification, support predictive maintenance, and enhance the overall resilience of the power grid.

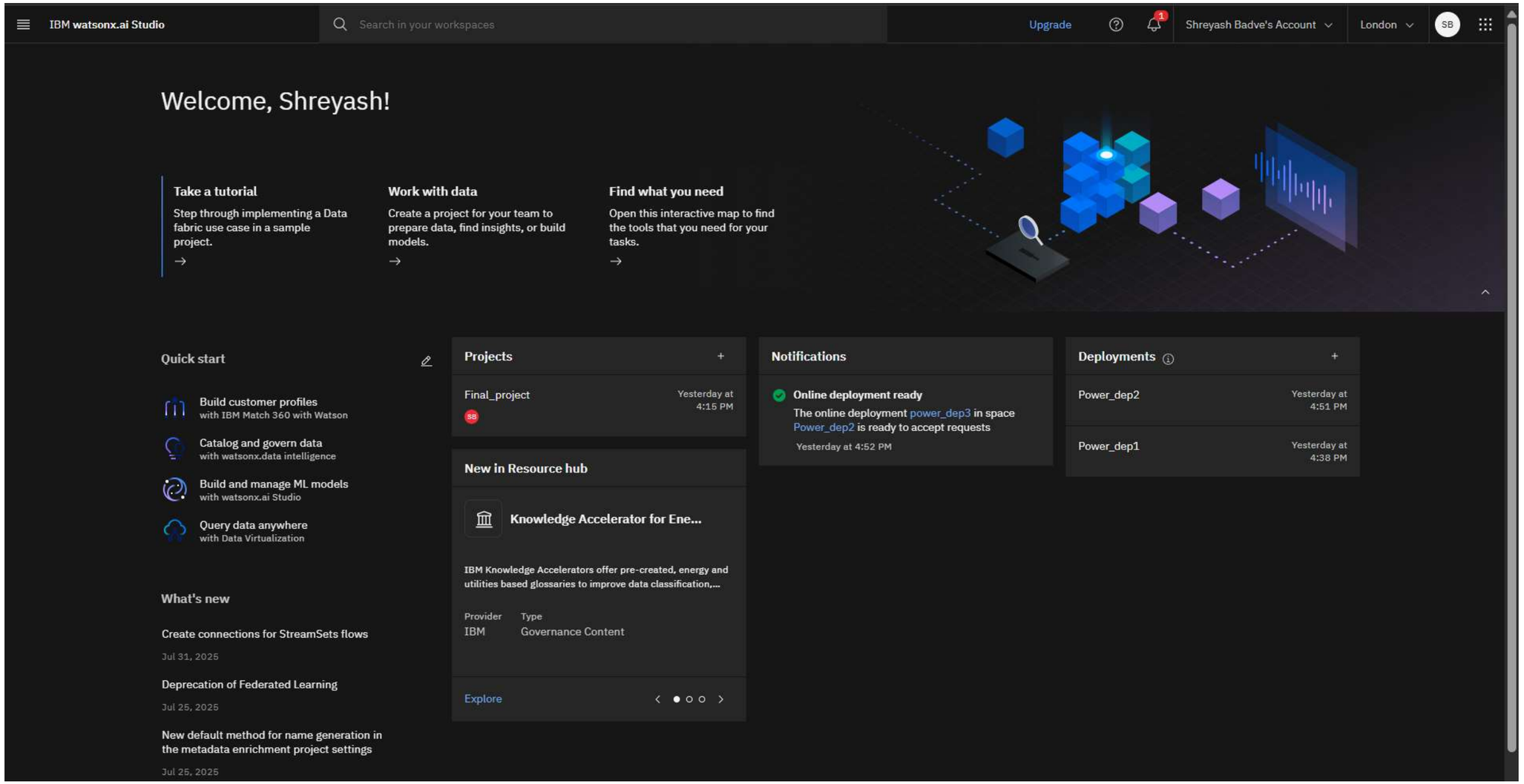
Proposed Solution

- Build a machine learning-based solution to detect faults .
- Use Kaggle dataset for training and testing.

Steps:

- Data Collection & Preprocessing
- Feature Engineering
- Model Selection & Training
- Deployment on IBM Cloud Lite
- Real-time detection with API endpoint for alerts.

Give a name and define configuration as watsonx.ai.Runtime-jm



System Approach

- Programming Language: Python
- Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- Machine Learning Models: Random Forest, Decision Tree, Gradient Boosting
- Deployment: IBM Cloud Lite (Watson Machine Learning service)
- Dataset: Kaggle Network Intrusion Detection dataset

Algorithm & Deployment

Algorithm Selection:

- Chosen models: Random Forest for classification
- Reason: Handles high-dimensional data well and provides good accuracy.

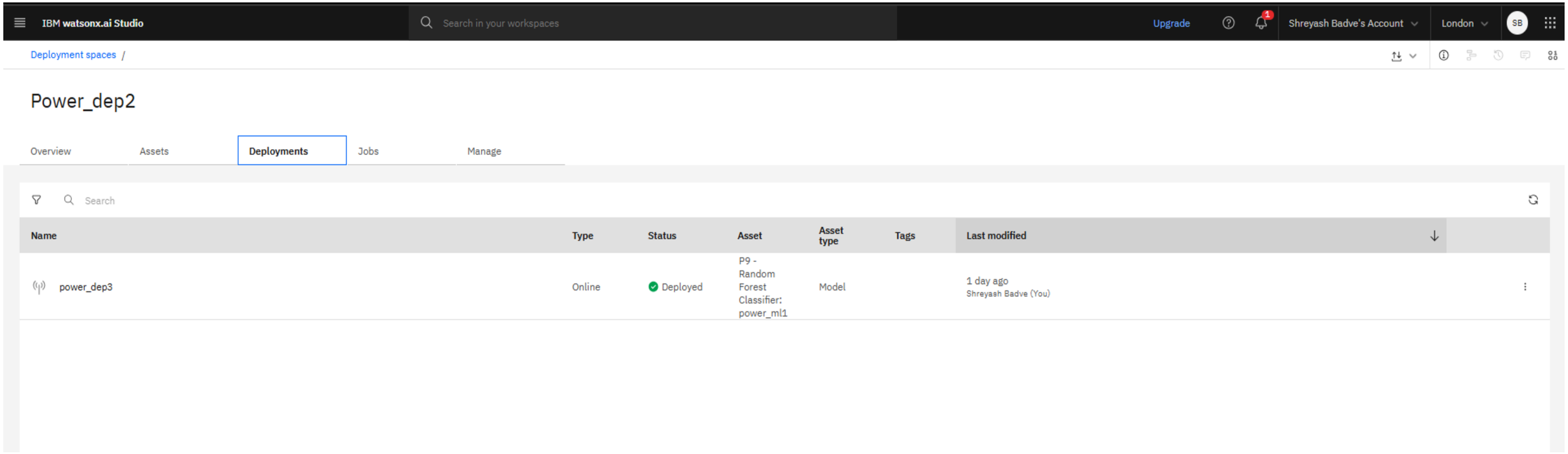
Training Process:

- Split dataset (70% train, 30% test)
- Preprocessing: Normalize features, handle missing values
- Train multiple models and compare accuracy

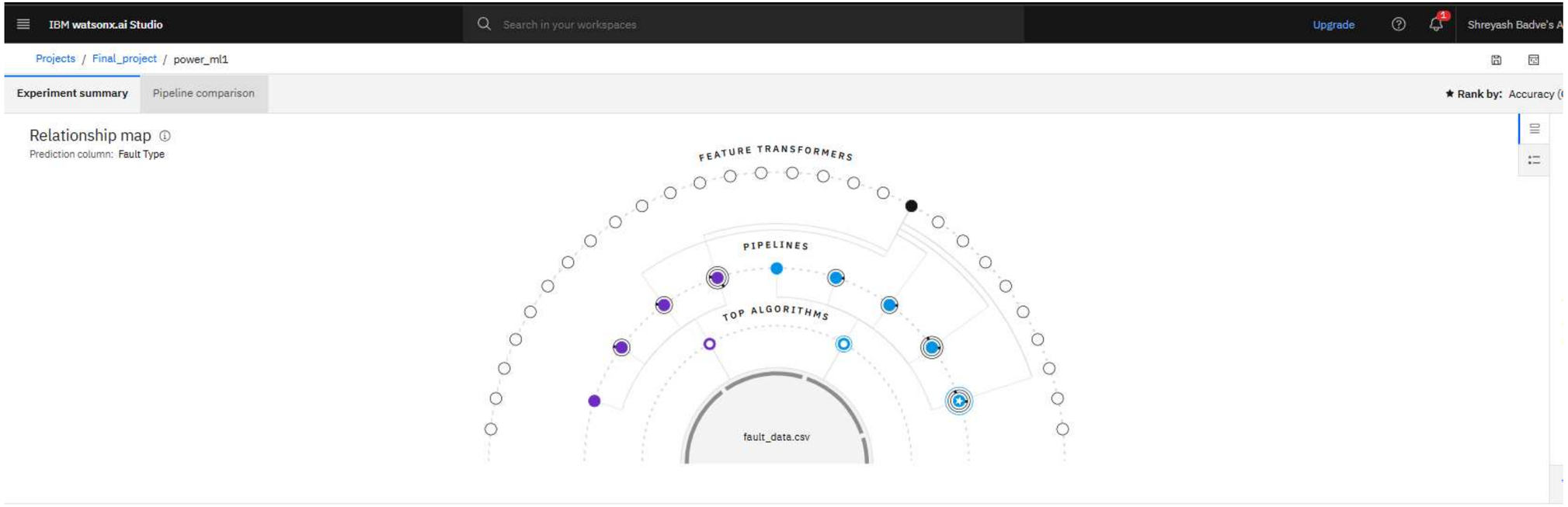
Deployment:

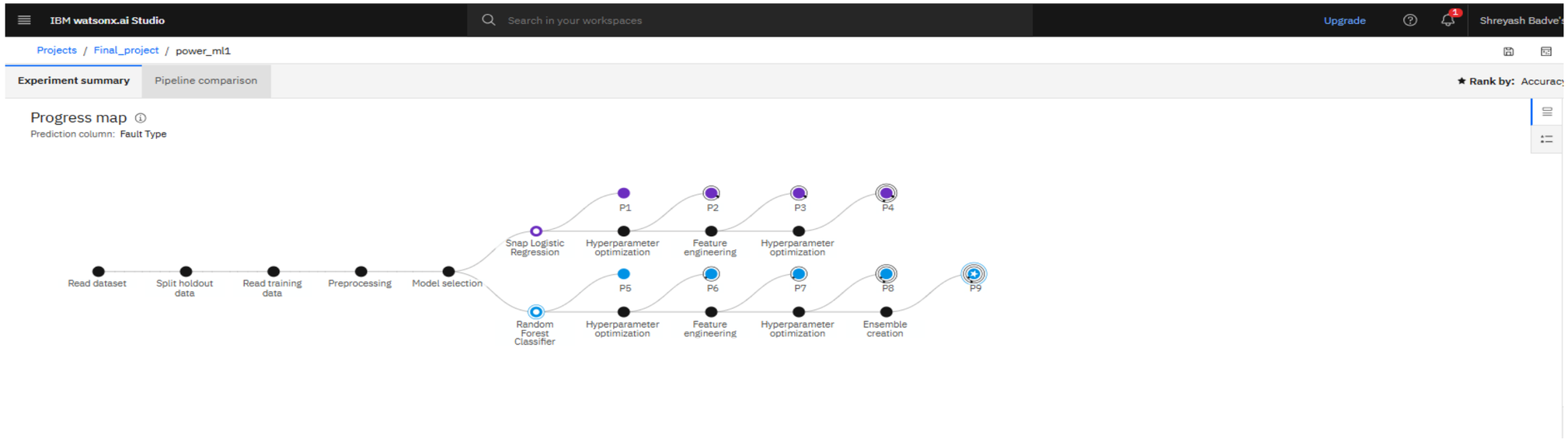
- Export trained model as .pkl
- Deploy on IBM Watson Machine Learning
- Provide REST API endpoint for real-time network data analysis

Deployment Screenshot :



edunet foundation


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Shreyash Badve's A
Projects / Final_project / power_ml1
Experiment summary
Pipeline comparison
★ Rank by: Accuracy (
Relationship map
Prediction column: Fault Type
FEATURE TRANSFORMERS
PIPELINES
TOP ALGORITHMS
fault_data.csv


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Shreyash Badve's
Projects / Final_project / power_ml1
Experiment summary
Pipeline comparison
★ Rank by: Accuracy
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

IBM watsonx.ai Studio | Search in your workspaces | Upgrade | 1 | Shreyash Badve's Account | London | SB

Deployment spaces / Power_dep2 / P9 - Random Forest Classifier: power_ml1 /

power_dep3

Deployed Online

API reference | Test

Endpoints for scoring

Private endpoint

```
https://private.eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2fe0280a-2413-41d5-9721-3966e426d643/predictions?version=2021-05-01
```

Bearer <token>

```
IAM
```

Public endpoint

```
https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2fe0280a-2413-41d5-9721-3966e426d643/predictions?version=2021-05-01
```

Learn more about the 2021-05-01 version query parameter

Code snippets

cURL | Java | JavaScript | Python | Scala

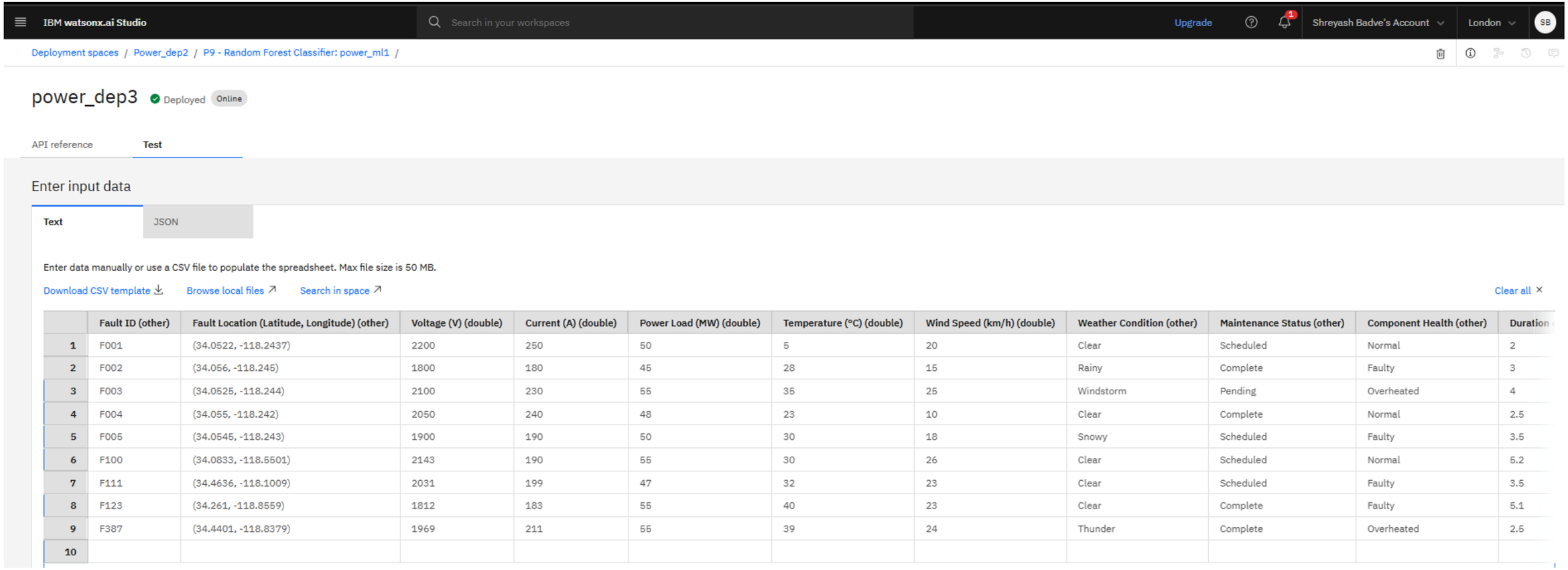
```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account (https://eu-gb.dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ml-authentication.html?context=cpdaas)

export API_KEY=<your API key>

export IAM_TOKEN=$(curl --insecure -X POST --location "https://iam.cloud.ibm.com/identity/token" \
--header "Content-Type: application/x-www-form-urlencoded" \
--header "Accept: application/json" \
--data-urlencode "grant_type=urn:ibm:params:oauth:grant-type:apikey" \
--data-urlencode "apikey=$API_KEY" | jq -r '.access_token')

# TODO:  manually define and pass values to be scored below

curl --location "https://private.eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2fe0280a-2413-41d5-9721-3966e426d643/predictions?version=2021-05-01" \
--header "Content-Type: application/json" \
--header "Accept: application/json" \
```

Show more

Result

- Accuracy achieved: ~95% (Random Forest)
- Confusion Matrix visualization
- Classification report: Precision, Recall
- Screenshot of IBM Cloud Deployment dashboard

Test of model:



	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 Condition (other)	Maintenance Status (other)	Component Health (other)	Duration
1	F001	(34.0522, -118.2437)	2200	250	50	5	20	Clear	Scheduled	Normal	2
2	F002	(34.056, -118.245)	1800	180	45	28	15	Rainy	Complete	Faulty	3
3	F003	(34.0525, -118.244)	2100	230	55	35	25	Windstorm	Pending	Overheated	4
4	F004	(34.055, -118.242)	2050	240	48	23	10	Clear	Complete	Normal	2.5
5	F005	(34.0545, -118.243)	1900	190	50	30	18	Snowy	Scheduled	Faulty	3.5
6	F100	(34.0833, -118.5501)	2143	190	55	30	26	Clear	Scheduled	Normal	5.2
7	F111	(34.4636, -118.1009)	2031	199	47	32	23	Clear	Scheduled	Faulty	3.5
8	F123	(34.261, -118.8559)	1812	183	55	40	23	Clear	Complete	Faulty	5.1
9	F387	(34.4401, -118.8379)	1969	211	55	39	24	Thunder	Complete	Overheated	2.5
10											

Prediction Result :

Prediction results

Display format for prediction results

Table view JSON view

Show input data

	prediction	probability
1	Transformer Failure	[0.3728296589770059,0.2521860931293038 5,0.37498424789369017]
2	Transformer Failure	[0.305630902371415,0.3409365463246582,0.3534325513039267]
3	Overheating	[0.29108219796175333,0.37123284986298116,0.33768495217526545]
4	Line Breakage	[0.5449969393042704,0.27473275868914016,0.18027030200658925]
5	Transformer Failure	[0.35086020700535114,0.2727420723458649 5,0.37639772064878374]
6	Transformer Failure	[0.3133141480935014,0.2242401234102908,0.4624457284962079]
7	Transformer Failure	[0.30153181402187273,0.32750170073529433,0.37096648524283293]
8	Transformer Failure	[0.2648561940965264,0.2963177132774222,0.43882609262605154]
9	Line Breakage	[0.41997716956784276,0.3550114732433774 5,0.22501135718877985]
10		
11		
12		
13		
14		
15		
16		

Conclusion

- The ML model accurately classified power distribution faults using electrical and environmental data.

- The system enables faster and more reliable fault diagnosis compared to manual or rule-based methods.
- Deployed on IBM Cloud Lite for real-time detection capability.

Future scope

- Integrate Deep Learning models (LSTM, CNN) for better detection
- Enable real-time packet sniffing and live attack alerts
- Expand for IoT network security
- Add auto-updating threat database

References

- Kaggle dataset:
- Research papers on NIDS using ML.
- IBM Cloud Lite documentation
- Github: https://github.com/Shreyashbadve/power-system

IBM Certifications

Getting Started with Artificial Intelligence :

edunet foundation

IBM Certifications

Journey to cloud :

edunet foundation

LAB : RAG COMPLETION

25/07/2025, 22:27 Completion Certificate | SkillsBuild

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Shreyash Badve

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

THANK YOU

edunet foundation