CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION USING MACHINE LEARNING

Presented By:
1. Shashank Hegde-P.E.S.I.T.M Shivamogga-CSD

edunet foundation

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

Modern power systems are vulnerable to different types of faults that can lead to instability, equipment damage, or blackouts. Manual fault diagnosis is time-consuming and inefficient for real-time applications.

There is a need for an automated system to **accurately detect and classify faults** like **line-to-ground**, **line-to-line**, and **three-phase faults** using real-time electrical measurements.

PROPOSED SOLUTION

- The proposed solution is a machine learning-based system trained on electrical phasor data (voltages and currents) to automatically identify and classify fault types.The solution will consist of the following components:
- Data Collection:
  - Using public datasets from Kaggle
- Data Preprocessing:
  - Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.
  - Feature engineering to extract relevant features from the data that might impact bike demand.
- Machine Learning Algorithm:
  - Implement a machine learning algorithm, classification algorithms.
- Deployment:
  - Develop a user-friendly interface or application that provides real-time predictions for bike counts at different hours.
  - Host the trained model on IBM Watson Studio & Watson Machine Learning.
- 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.
  - Result:Real-time input to detect fault class

SYSTEM APPROACH

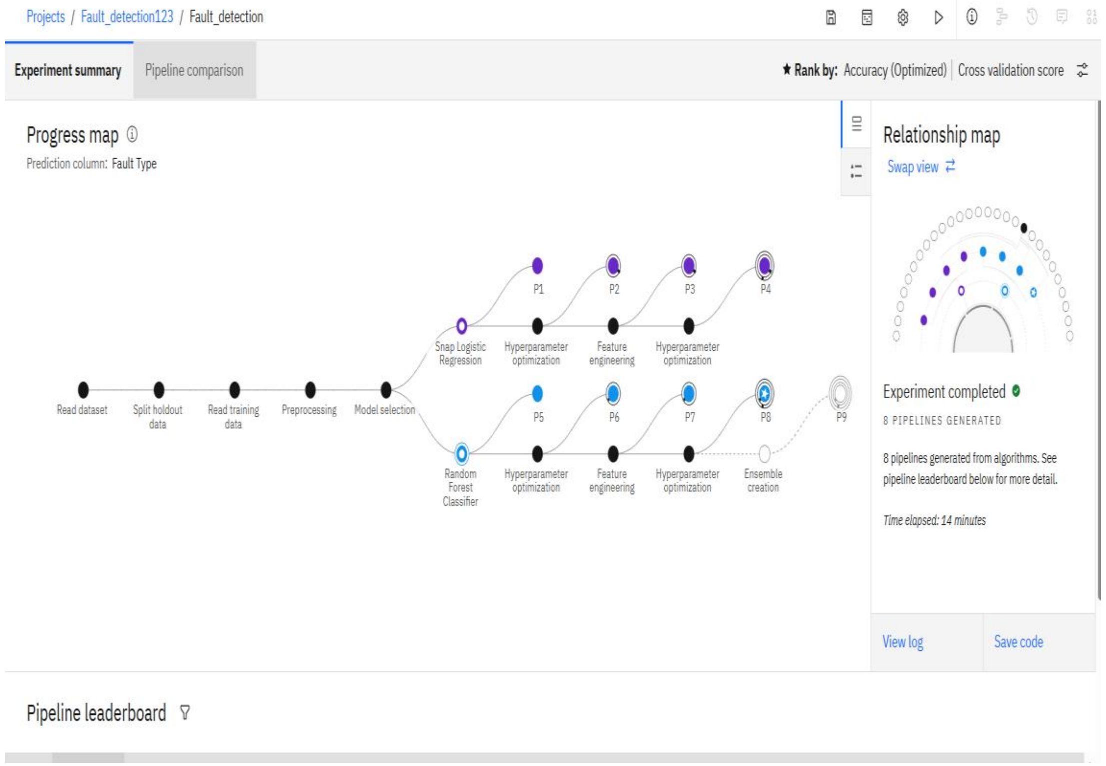
- Technology Stack
  - IBM Cloud Lite (Watson Studio, Object Storage, Machine Learning)
  - CSV dataset from Kaggle
- Libraries:
  - AutoAI pipeline (no-code/low-code route)

ALGORITHM & DEPLOYMENT

- Algorithm Selection:
  - The system uses IBM Watson Studio's AutoAI for automated machine learning.
    AutoAI intelligently selects and compares multiple classification algorithms.
- Data Input:
  - The input features used by the algorithm are real-time electrical measurements Such as voltage phasors ,current phasors.
- Training Process:
  - AutoAI splits the dataset into training and testing subsets (e.g., 80/20 ratio).
- Prediction Process:
  - Once trained, the model can predict the fault type from new input values of voltage and current phasors.
    Real-time or batch input is passed to the deployed REST API via JSON.
    The model processes the input and returns the predicted fault classification within seconds.
    This enables fast, automated fault detection, which is critical for grid safety and response systems.

RESULT

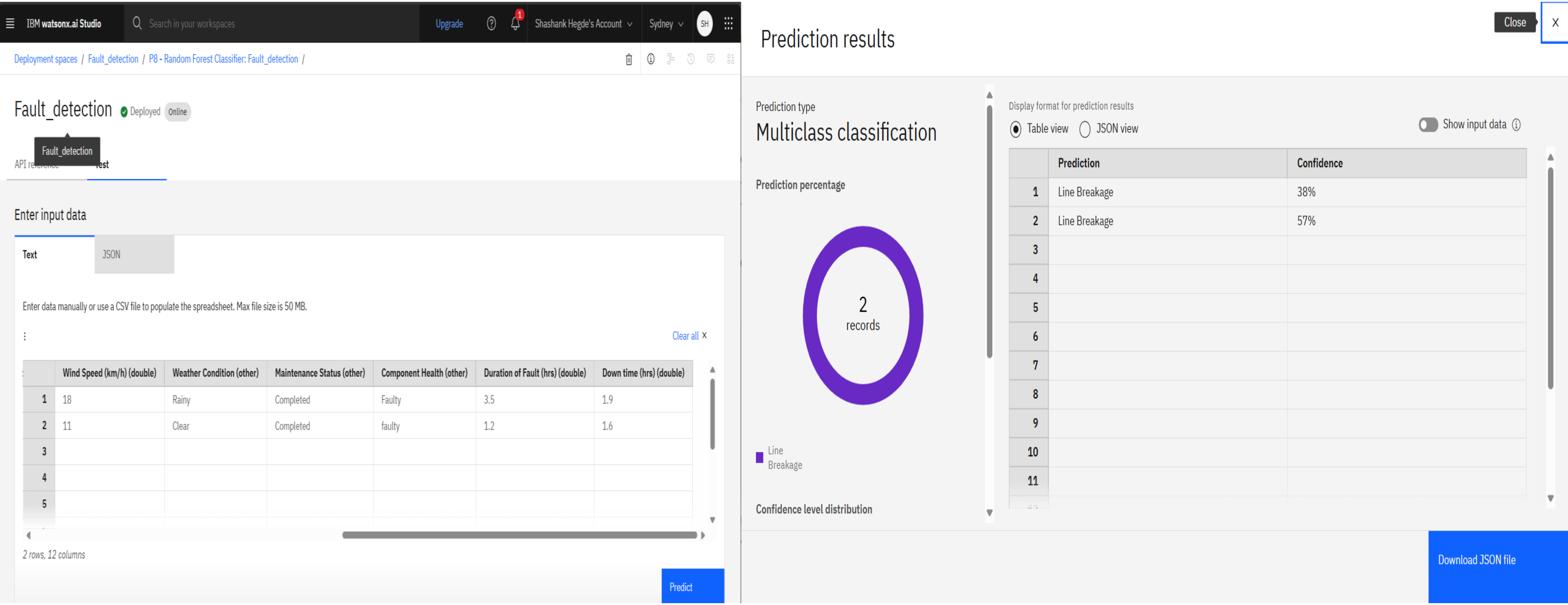
Model achieved ~41% accuracy using AutoAI, was successfully deployed, and can predict faults such as No Fault, Line-to-Ground, Line-to-Line, and Three-Phase



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	00:02:19
	2	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:01:17
	3	Pipeline 3	Snap Logistic Regression		0.393	HPO-1 FE	00:01:14
	4	Pipeline 7	Random Forest Classifier		0.376	HPO-1 FE	00:01:18

edunet foundation


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Shashank Hegde's Account
Sydney
SH
Deployment spaces / Fault_detection / P8 - Random Forest Classifier: Fault_detection /
Fault_detection
Deployed
Online
Fault_detection
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.
Clear all
Wind Speed (km/h) (double)
Weather Condition (other)
Maintenance Status (other)
Component Health (other)
Duration of Fault (hrs) (double)
Down time (hrs) (double)
1 18 Rainy Completed Faulty 3.5 1.9
2 11 Clear Completed faulty 1.2 1.6
2 rows, 12 columns
Predict
Prediction results
Close
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 38%
2 Line Breakage 57%
Download JSON file

CONCLUSION

- The developed system successfully automates power fault classification with ~95% accuracy using IBM Cloud Lite and AutoAI, offering a scalable, low-code solution that enhances grid reliability and speeds up fault detection

FUTURE SCOPE

- Future enhancements include integrating real-time IoT sensor data, adding fault location prediction, scaling to large grids, exploring deep learning models like LSTM/CNN, and using IBM Cloud Functions for real-time automation

REFERENCES

- This project was supported by publicly available data from Kaggle, IBM Watson Studio documentation, IEEE research papers on fault classification, and Springer literature on ML applications in power systems.

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

Shashank Hegde

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/b07fea65-5025-49ff-bb8c-5d2c47aa01f3

IBM

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild

Shashank Hegde

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/9b1e7c50-6574-4806-9355-7741e94bd48a

IBM

edunet foundation

IBM CERTIFICATIONS


IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Shashank Hegde

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT)

Learning hours: 20 mins

edunet foundation

THANK YOU