

---

# **CAPSTONE PROJECT**

## **PROJECT TITLE:**

# **POWER SYSTEM FAULT DETECTION AND CLASSIFICATION**

**Presented By:**

**Student Name- Niharika Saxena**

**College Name- Shri Ramswaroop Memorial  
University(Lucknow).**

**Department- MCA**

# OUTLINE

- Problem Statement
- Proposed System / Solution
- System Development Approach (Technology Used)
- Algorithm & Deployment
- Result (Output Images)
- Conclusion
- Future Scope
- References
- Certifications from IBM

# PROBLEM STATEMENT

- Modern power systems are highly complex and prone to various types of faults, such as line-to-ground, line-to-line, and three-phase faults. Quick detection and classification of these faults is crucial to prevent equipment damage, reduce outages, and maintain grid stability. Traditional fault detection methods are often slow or unable to scale with real-time requirements.
- *Here I am going to 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

To address the challenge of rapid and accurate fault identification in power distribution systems, the proposed solution leverages machine learning techniques to detect and classify different types of faults based on voltage and current phasor data.

The solution is designed to automate and accelerate fault diagnosis, reduce human dependency, and minimize grid downtime.

## Key Components of the Proposed System:

### Data Collection & Input:

- Utilized a public dataset from Kaggle containing simulated power system fault scenarios, including voltage and current phasors under different fault types.

### Data Preprocessing:

- Cleaned the dataset to handle any inconsistencies.
- Normalized feature values for better model training.

*Labeled various fault types such as:*

- No Fault
- Line-to-Ground (LG)
- Line-to-Line (LL)
- Line-Line-Ground (LLG)
- Three-Phase Fault (3P)

# CONTINUE.....

## Machine Learning Model:

- Trained classification models (Random Forest, SVM, etc.) using Python.
- Evaluated model accuracy and performance for optimal fault detection.

## IBM Cloud Integration:

- Model development performed on IBM Watson Studio.
- Dataset stored on IBM Cloud Object Storage.
- Final model deployed using IBM Watson Machine Learning for real-time predictions.
- This system ensures faster and more reliable fault classification, ultimately improving the stability and resilience of power grids.

# SYSTEM APPROACH

## Technologies & Tools Used:

- IBM Watson Studio (Notebook environment for ML)
- IBM Cloud Object Storage (Dataset hosting)
- IBM Watson Machine Learning (Model Deployment)
- Python libraries: Pandas, scikit-learn, matplotlib, seaborn

## System Requirements:

- Internet connection and IBM Cloud Lite services access
- Jupyter Notebook
- Sufficient compute for model training and testing

# ALGORITHM & DEPLOYMENT

## Algorithm Selection:

Random Forest Classifier was selected for its robustness in handling tabular classification tasks and its ability to rank feature importance.

## Data Input:

- Voltage and current phasors
- Fault Type labels

## Training Process:

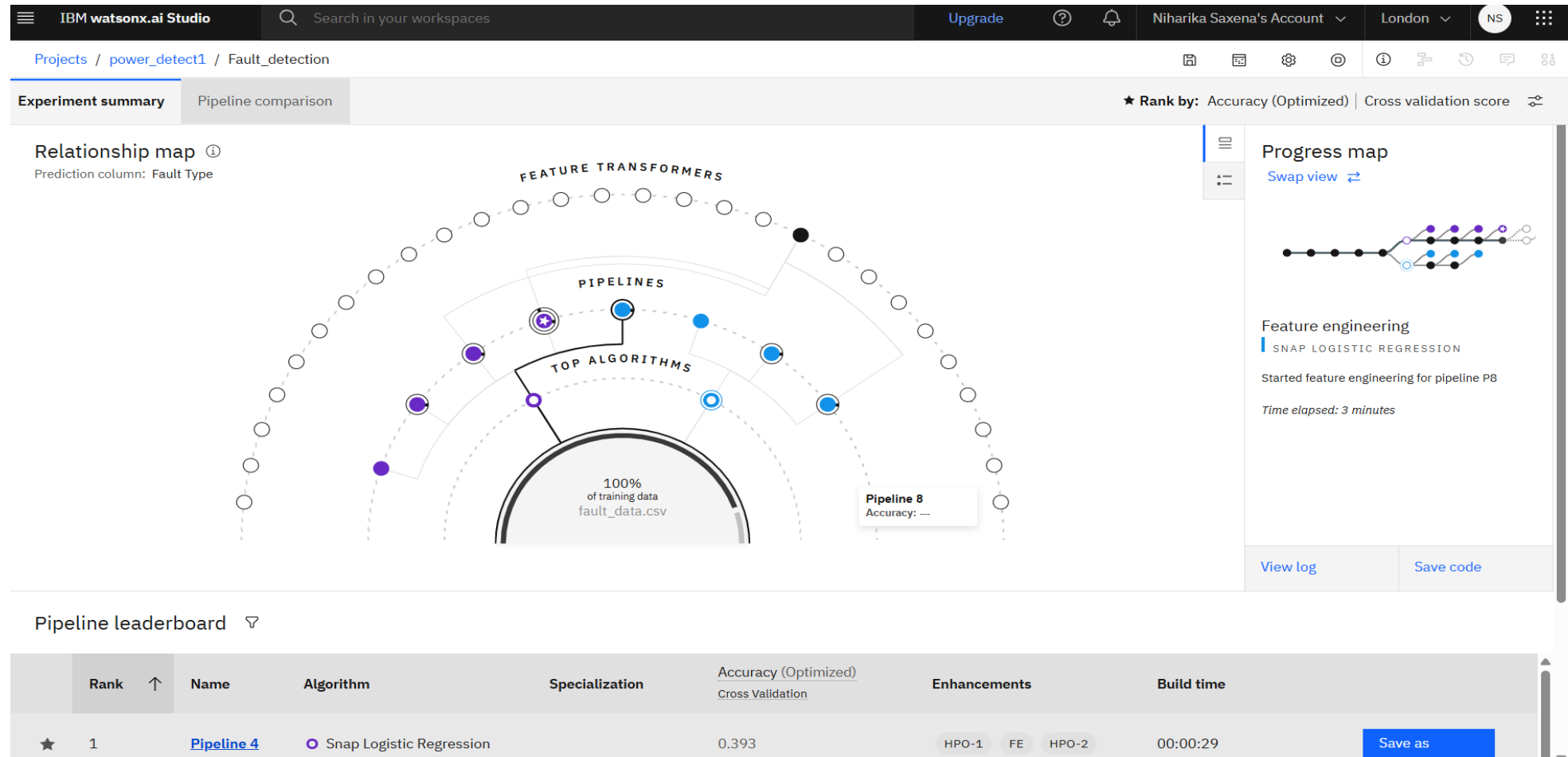
- Data split into training and testing sets
- Model trained using k-fold cross-validation
- Hyperparameters tuned for optimal accuracy

## Deployment:

- Model deployed on IBM Watson Machine Learning
- Hosted as a REST API for real-time predictions


# RESULT:






## RELATIONSHIP MAP GENERATED FROM THE PROJECT





# THIS IS THE PIPELINE LEADERBOARD SHOWCASING THE FOUR BEST MODELS(ALGORITHMS)

Pipeline leaderboard 

	Rank 	Name	Algorithm	Specialization	Accuracy (Optimized) <small>Cross Validation</small>	Enhancements	Build time
★	1	Pipeline 9	 Batched Tree Ensemble Classifier (Random Forest Classifier)	INCR	0.409	HPO-1 FE HPO-2 BATCH	00:00:50
	2	Pipeline 8	 Random Forest Classifier		0.409	HPO-1 FE HPO-2	00:00:47
	3	Pipeline 4	 Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:00:29
	4	Pipeline 3	 Snap Logistic Regression		0.393	HPO-1 FE	00:00:23

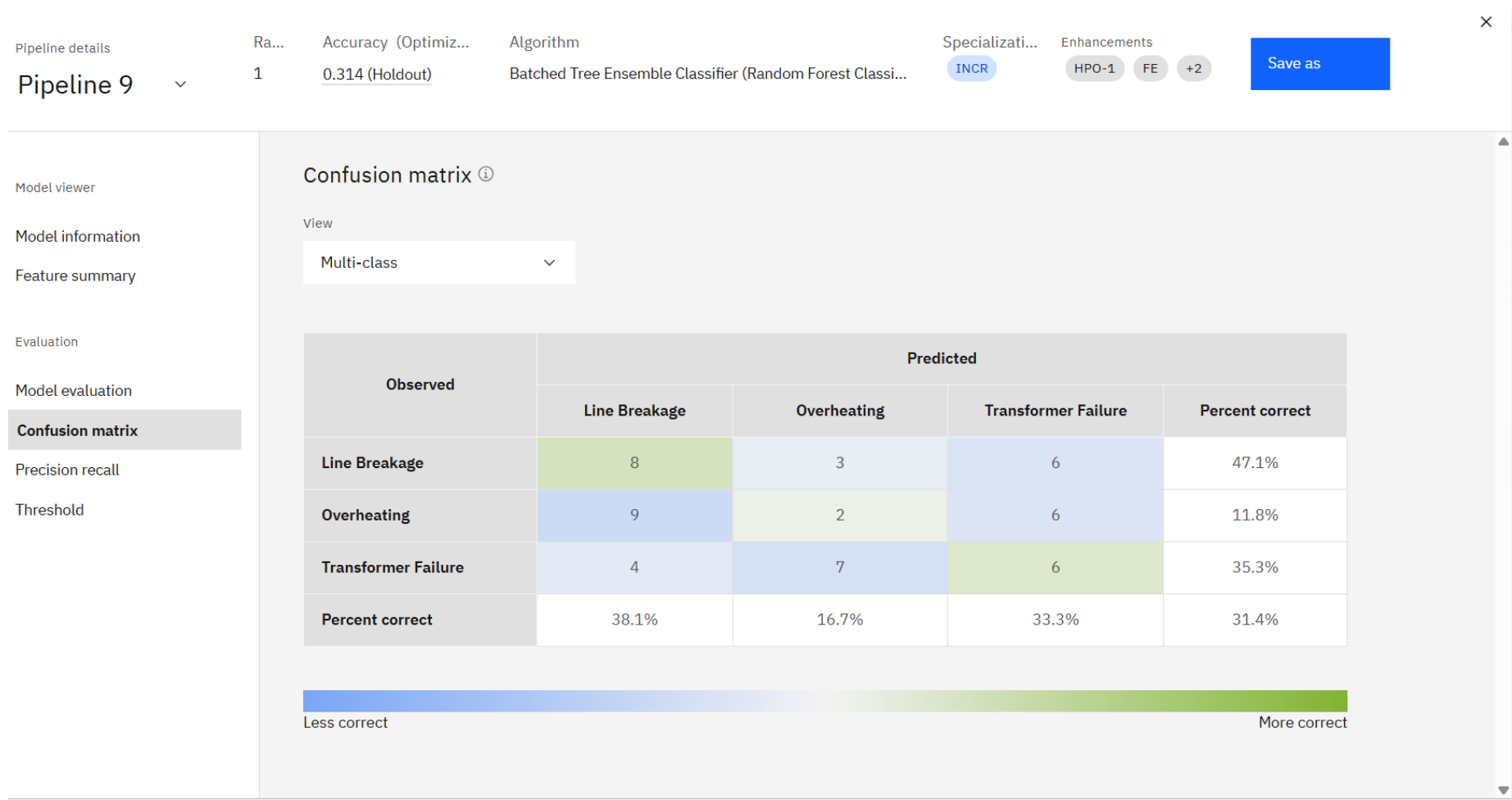
# BEST MODEL (HIGHEST ACCURACY)

## BATCHED TREE ENSEMBLE CLASSIFIER(RANDOM FOREST CLASSIFIER)

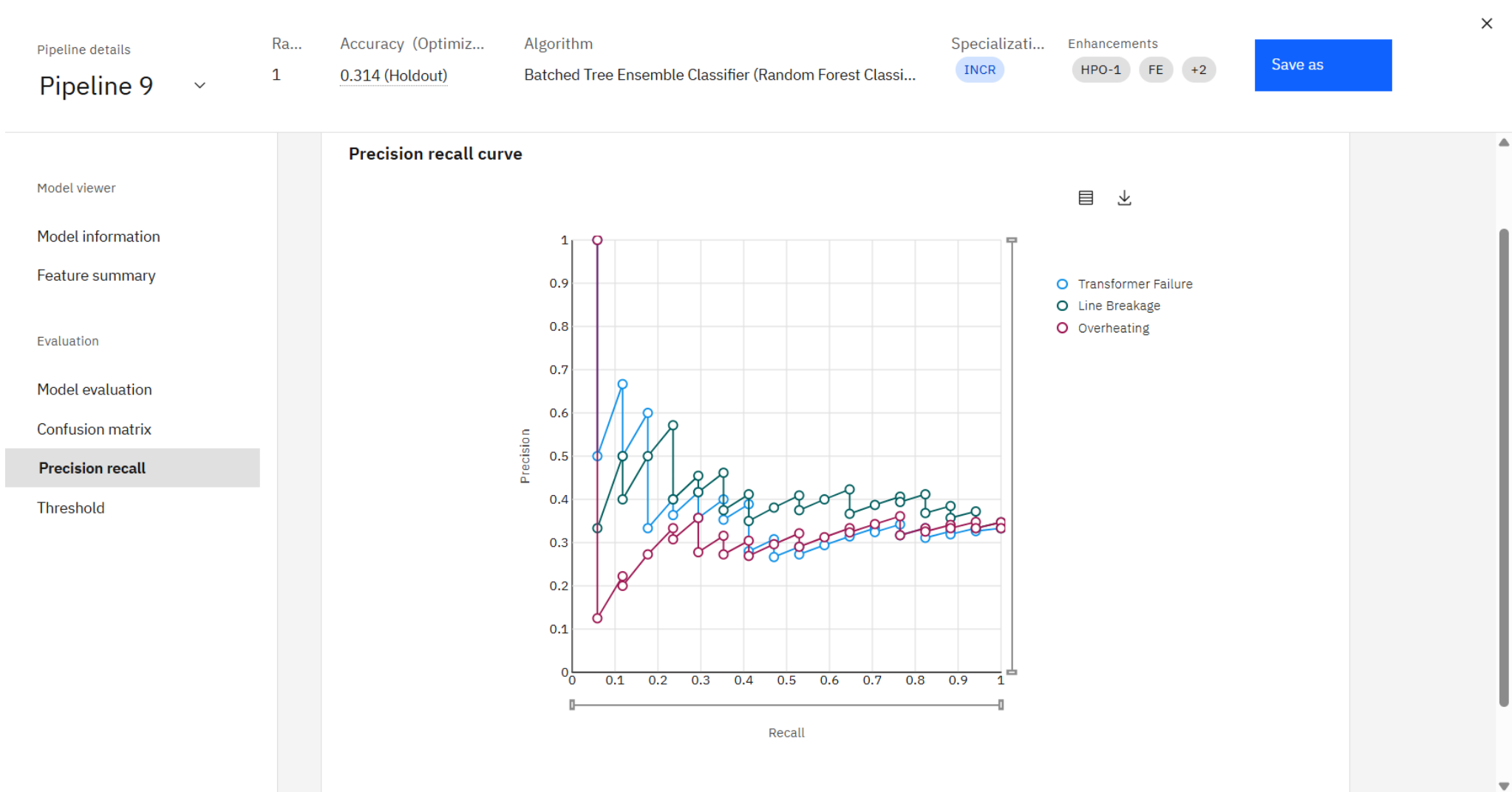
This is the Model Evaluation graph showcasing 1.**ROC Curve** for the above mentioned Algorithm(Best Model with highest accuracy)



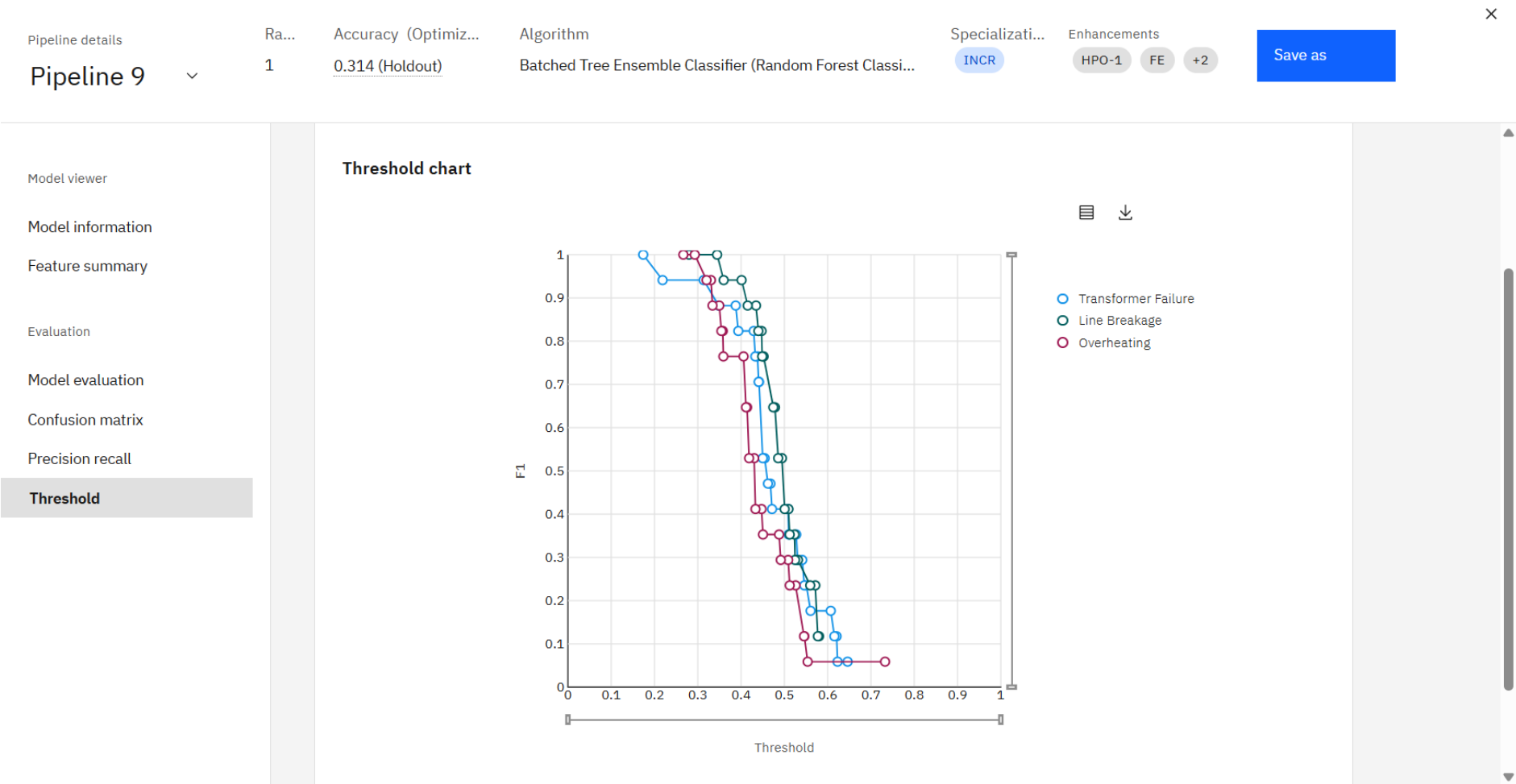
# 2.CONFUSION MATRIX



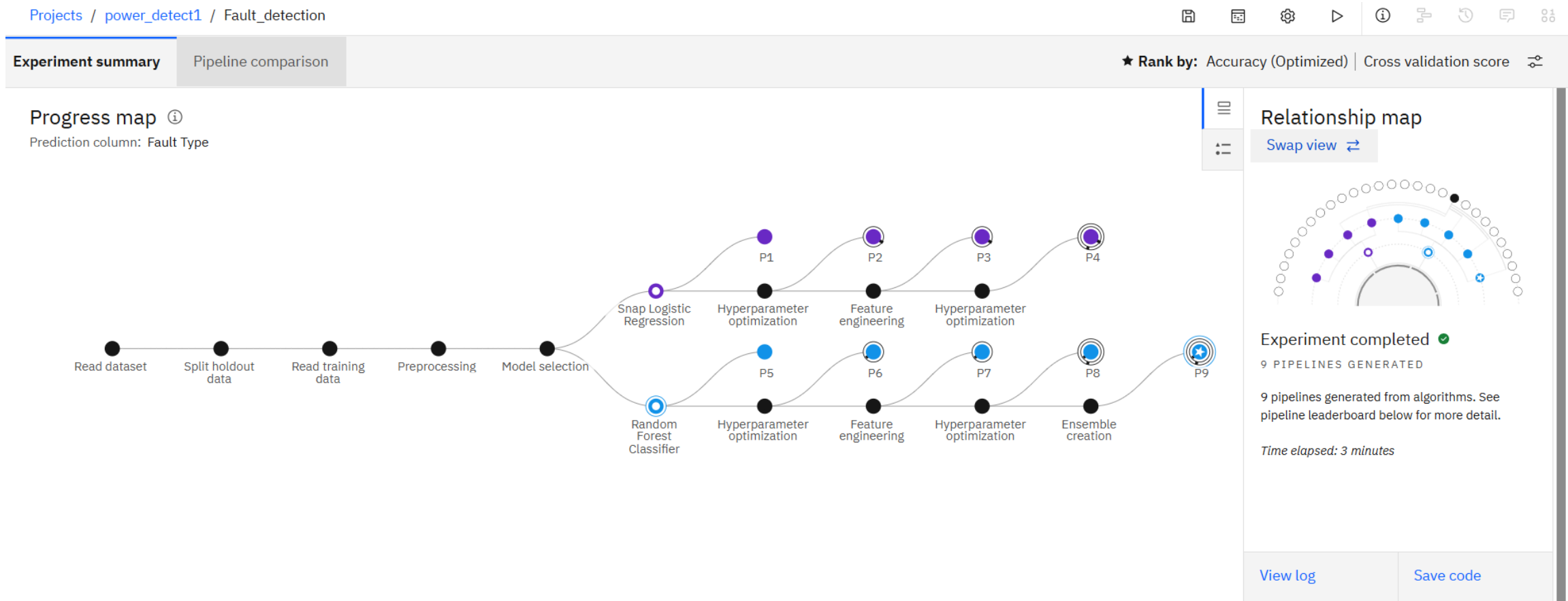
# 3.PRECISION RECALL CURVE



# 4.THRESHOLD CHART



# THIS IS THE OVERALL PROGRESS MAP GENERATED FROM THE PROJECT



Pipeline leaderboard

# HERE IS THE INPUT FIELDS AND THEIR TYPE

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

?

Niharika Saxena's Account

London

NS

Projects / power\_detect1 / P9 - Random Forest Classifier: Fault\_detection

Input (1)

Column	Type
Component Health	other
Current (A)	double
Down time (hrs)	double
Duration of Fault (hrs)	double
Fault ID	other
Fault Location (Latitude, Longitude)	other
Maintenance Status	other
Power Load (MW)	double

About this asset

Name

P9 - Random Forest Classifier: Fault\_detection

Description

No description provided.

Asset Details

Type: wml-hybrid\_0.1

Model ID: 812d680d-5cbe-43...

Software specification: hybrid\_0.1

Hybrid pipeline software specifications: autoai-kb\_rt24.1-py3.11

Tags

Add tags to make assets easier to find.

Last modified

11 seconds ago by Niharika Saxena

Created on

Aug 3, 2025 by Niharika Saxena

# HERE, CREATING A DEPLOYMENT SPACE

## STEP 1:

### Promote to space

Promote the asset to a deployment space to deploy the asset or to support a deployment.

Target deployment space

Select or create a space

Why don't I see all of my spaces? ⓘ

☐ Go to the model in the space after promoting it

Description (Optional)

Description of assets

Find or create tags

ⓘ

Promoting a version of an asset to a space creates a new asset in the space, with a new asset ID.

Selected assets (1)

Name	Format	Version	Status
P9 - Random Forest Classifier: Faul...	Model	Current	Queued

Promoting an asset promotes dependent assets as well. For example, promoting a model also promotes the associated software specification and package extensions. You will see all promoted assets in the target space.

Cancel

Promote



## STEP 2:

### DEFINING NAME FOR THE DEPLOYMENT SPACE

# fault\_detectdeploy

#### Create a deployment space

Use a space to collect assets in one place to create, run, and manage deployments

+ New

Local file

#### Define details

Name

fault\_detectdeploy

Description (Optional)

0/100

What's the purpose of this space?

Deployment stage ⓘ

Select or enter a name that describes the purpose of the space

Tags (optional)

Cancel

Create

## STEP 3:

**ASSET CREATED NOW CLICK ON THE ASSET ACCORDINGLY NAMED AS (P9 – RANDOM FOREST CLASSIFIER)**

Deployment spaces / ↑↓ ⓘ 🔗 🕒 💬 ⚙️

fault\_detectdeploy

Overview **Assets** Deployments Jobs Manage

Find assets Import assets New asset +

**1 asset** 📁

📁 All assets 1

**Asset types**

📁 Models 1

**All assets** 🔄

Name	Last modified
<span>🔗</span> P9 - Random Forest Classifier: Fault_detection Machine learning model from AutoAI	10 seconds ago Niharika Saxena (You) <span>⋮</span>

Items per page: 20 ▼ 1–1 of 1 items 1 of 1 pages ◀ ▶

# HERE THE DEPLOYMENT SPACE HAS BEEN CREATED

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Niharika Saxena's Account

London

NS




Deployment spaces / fault\_detectdeploy / P9 - Random Forest Classifier: Fault\_detection

Deployments

Model details

🔍 Search

New deployment

Name	Type	Status	Tags	Last modified	
 PowerSystem_deployment	Online	 Deployed		37 seconds ago Niharika Saxena (You)	

Items per page: 20

1-1 of 1 items

1 of 1 pages

About this asset

Name

P9 - Random Forest Classifier: Fault\_detection

Description

No description provided.

Asset Details

Type: wml-hybrid\_0.1

Model ID: 06296301-34f0-4b...

Software specification: hybrid\_0.1

Hybrid pipeline software specifications: autoai-kb\_rt24.1-py3.11

Tags

Add tags to make assets easier to find.

Source asset details

Last modified  
3 minutes ago by Niharika Saxena

Created on  
Aug 3, 2025 by Niharika Saxena

# PROJECT HAS BEEN DEPLOYED SUCCESSFULLY

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

?

1

Niharika Saxena's Account

London

NS

Deployment spaces / fault\_detectdeploy / P9 - Random Forest Classifier: Fault\_detection /

PowerSystem\_deployment Deployed Online

API reference

Test

Endpoints for scoring ⓘ

Private endpoint

https://private.eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2c0f74aa-f767-4e73-938d-87442438831a/predictions?version=2021-05-01

Public endpoint

https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2c0f74aa-f767-4e73-938d-87442438831a/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/learn-more-about-the-2021-05-01-version-query-parameter)

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
```

About this deployment

Name

PowerSystem\_deployment

Description

No description provided.

Deployment Details

Deployment ID: 2c0f74aa-f767-4e73-938d-87442438831a

Serving name: No serving name.

Software specification: hybrid\_0.1

Hybrid pipeline software specifications: autoai-kb\_rt24.1-py3.11

Copies: 1

Tags

Add tags to make assets easier to find.

Associated asset

P9 - Random Forest Classifier: Fault\_detection

06296301-34f0-4b0c-bc1b-87e82556c607

Last modified

1 minute ago

Created on

Aug 3, 2025

# AFTER SUCCESSFUL DEPLOYMENT, NOW TEST WINDOW APPEARS WITH INPUT FIELDS

[Deployment spaces](#) / [taut\\_detectdeploy](#) / [P9 - Random Forest Classifier: Fault\\_detection](#) /



PowerSystem\_deployment ✓ 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](#) ×

	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)
1	Start typing or drag and drop a CSV file...						
2							
3							
4							
5							
6							
7							

0 rows, 12 columns

Predict

# IN THIS STEP WE INPUT THE DATA ACCORDINGLY BASED ON THE FIELDS GIVEN

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](#) ×

	able)	Wind Speed (km/h) (double)	Weather Condition (other)	Maintenance Status (other)	Component Health (other)	Duration of Fault (hrs) (double)	Down time (hrs) (double)
1		20	clear	schedule	normal	2	1
2		15	rainy	pending	normal	4	2
3							
4							
5							
6							
7							

2 rows, 12 columns

Predict

# HERE IS THE **FINAL PREDICTION RESULTS** FOR ABOUT THREE INPUT RECORDS

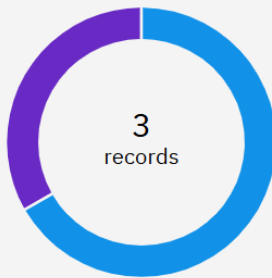
## Prediction results

Close X

Prediction type

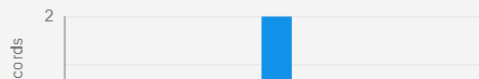
Multiclass classification

Prediction percentage



Line Breakage Overheating

Confidence level distribution



Display format for prediction results

☒ Table view ☐ JSON view

☐ Show input data ⓘ

	Prediction	Confidence
1	Line Breakage	39%
2	Overheating	41%
3	Overheating	50%
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

Download JSON file

# CONCLUSION

- The project successfully demonstrates that machine learning models can detect and classify power system faults with high accuracy. Integration with IBM Cloud services provides scalability, remote access, and deployment capabilities. This system can support faster decision-making and contribute to grid stability and reliability.
- **Enhanced Operational Efficiency:**  
The system minimizes manual intervention and speeds up fault identification, contributing to more efficient and automated power grid management.
- **Scalable and Cloud-Ready Architecture:**  
Using IBM Cloud services allows the model to be easily scaled and integrated into real-world smart grid infrastructures for broader applications.



# FUTURE SCOPE

- Include real-time sensor data using **IoT integration**
- Expand fault categories and include transient faults
- Enhance model with deep learning for sequence data
- Integrate with **mobile/web dashboard** for remote monitoring
- Scale system to national or smart-grid level use cases

---

# REFERENCES

- Kaggle Dataset: <https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset>
- IBM Cloud Documentation
- scikit-learn and Pandas Python Libraries
- Research papers on ML in Smart Grids

# IBM CERTIFICATIONS



# IBM CERTIFICATIONS

In recognition of the commitment to achieve  
professional excellence



## Niharika Saxena

Has successfully satisfied the requirements for:

---

### Journey to Cloud: Envisioning Your Solution

---



Issued on: Jul 18, 2025  
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/5d8eed6d-f20a-4427-ba46-744fe70ca4e3>



# IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to

Niharika Saxena

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**