# CAPSTONE PROJECT

# POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

**Presented By:**
**Saurya Sircar-Jadavpur University-Electrical Engineering**

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

Stable power systems are a vital part of a modern society, and swift maintenance of faults is paramount for ensuring reliability.

Traditional fault detection relies on analog instruments and manual monitoring, which leads to increased response times, and using an automated system to identify faults based on electrical parameters can lead to faster resolutions.

# PROPOSED SOLUTION

- An Artificial Neural Network can be leveraged to identify and classify faults based on the available electrical parameters (Voltage and Current)
- Artificial neural networks offer excellent generalization capabilities and immunity to noise, which reduces the impact of variations in power system parameters
- Accuracy and fast execution times are prioritized, to ensure reliable and timely detection of faults

# DATA GATHERING AND PREPROCESSING

- For the purposes of this model, a generated through a MATLAB simulation of transmission lines under normal and fault conditions
- https://www.kaggle.com/datasets/esathyaprakash/electrical-fault-detection-and-classification
- The dataset contains LG fault only with phase A, LL faults only between phase B and C, and LLG faults only with phases A and B. This is a major roadblock for real-world viability, but serves as an appropriate training set for a proof of concept.
- The first four columns [G, C, B, A] contain binary values that denote whether each phase is part of a fault
- These columns are replaced with a “Fault Type” column programmatically using a python script, which shall act as labels for classification.
- Github link for data preprocessing: https://github.com/Kei74/ibm-cloud-project

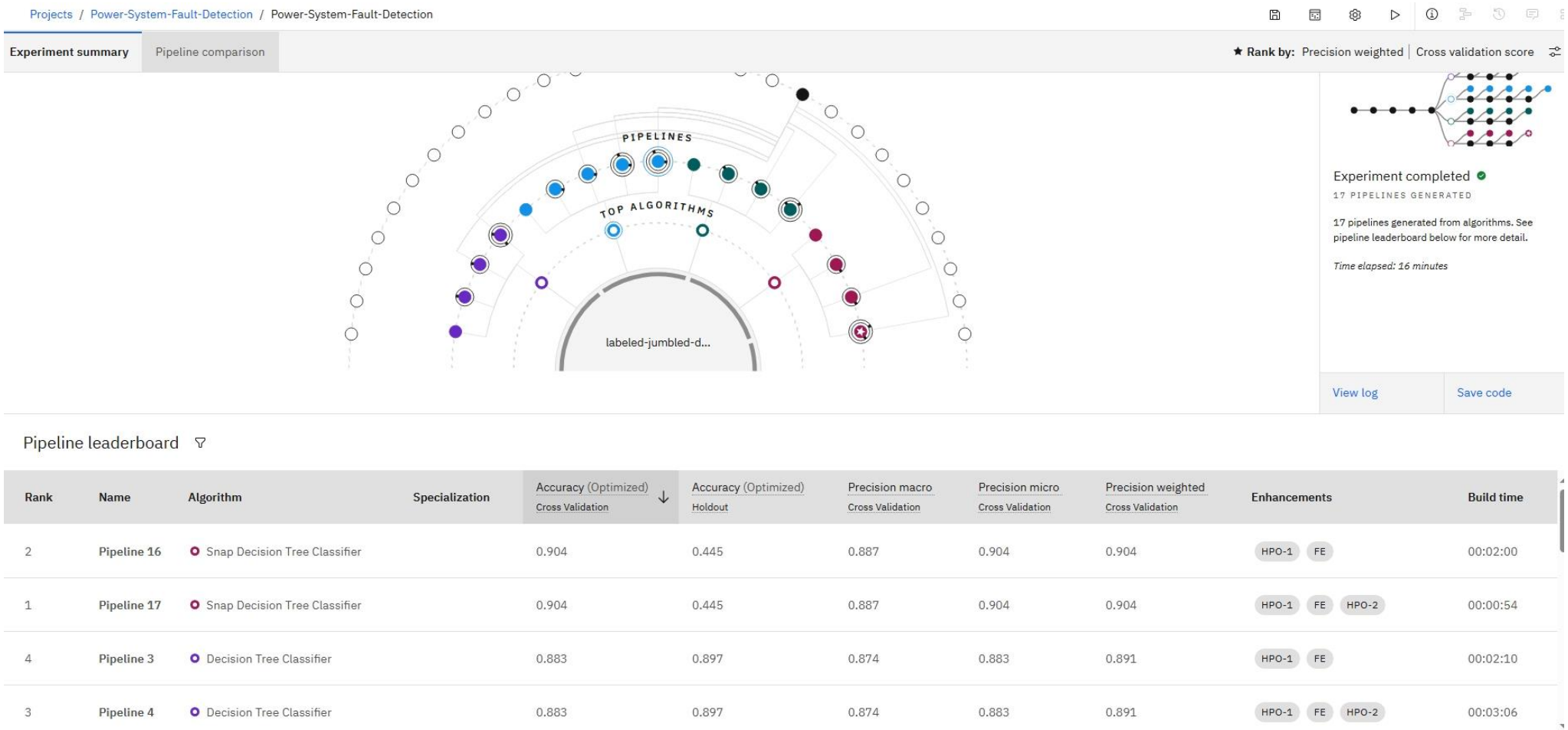
# SYSTEM APPROACH

IBM Cloud's Watsonx.ai Studio is utilized to test algorithms and train the model

- Watsonx.ai Runtime is used as hardware for training and testing,
- IBM Cloud Pak for Data is used to conduct the AutoAI experiment for selecting, training and testing the algorithm through a no-code approach

# ALGORITHM & DEPLOYMENT

- The AutoAI experiment's results are used to select the algorithm;




Pipeline leaderboard

| Rank | Name | Algorithm | Specialization | Accuracy (Optimized) Cross Validation | Accuracy (Optimized) Holdout | Precision macro Cross Validation | Precision micro Cross Validation | Precision weighted Cross Validation | Enhancements | Build time |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Pipeline 16 | Snap Decision Tree Classifier | | 0.904 | 0.445 | 0.887 | 0.904 | 0.904 | HPO-1 FE | 00:02:00 |
| 1 | Pipeline 17 | Snap Decision Tree Classifier | | 0.904 | 0.445 | 0.887 | 0.904 | 0.904 | HPO-1 FE HPO-2 | 00:00:54 |
| 4 | Pipeline 3 | Decision Tree Classifier | | 0.883 | 0.897 | 0.874 | 0.883 | 0.891 | HPO-1 FE | 00:02:10 |
| 3 | Pipeline 4 | Decision Tree Classifier | | 0.883 | 0.897 | 0.874 | 0.883 | 0.891 | HPO-1 FE HPO-2 | 00:03:06 |

# ALGORITHM & DEPLOYMENT

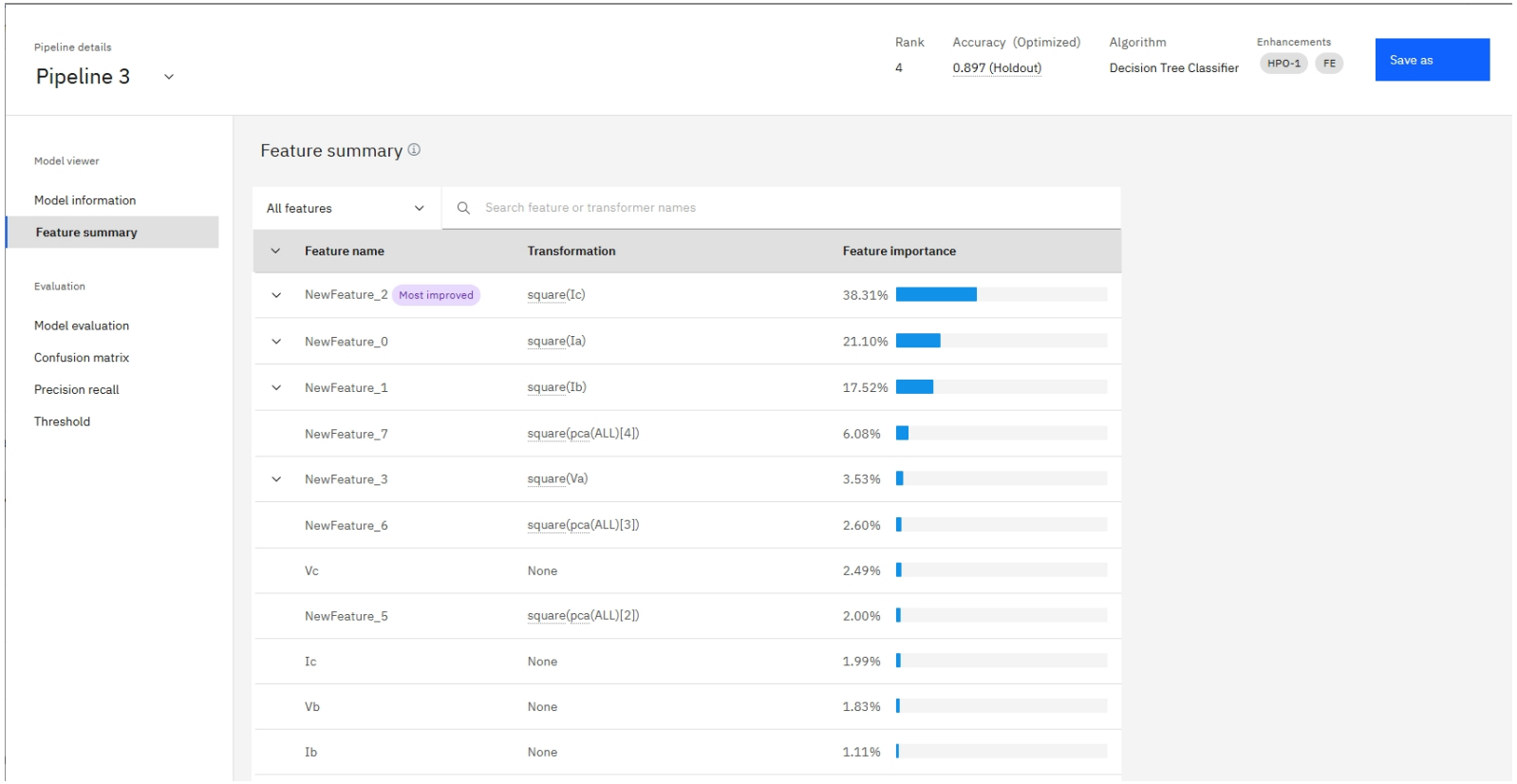
- Algorithm Selection:
  - Accuracy is the primary priority, and Snap Decision Tree Classifier appears to have the highest accuracy score
  - However the poor accuracy vs holdout data indicates overfitting
  - Hence **Decision Tree Classifier** is chosen as the algorithm
- Training method
  - Since Pipeline 3 offers identical accuracy to pipeline 4 at a lower build time, it is selected as the training pipeline, with the enhancements of Feature Engineering and 1st hyperparameter optimization

Pipeline leaderboard

| Rank | Name | Algorithm | Specialization | Accuracy (Optimized) Cross Validation | Accuracy (Optimized) Holdout | Precision macro Cross Validation | Precision micro Cross Validation | Precision weighted Cross Validation | Enhancements | Build time |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Pipeline 16 | Snap Decision Tree Classifier | | 0.904 | 0.445 | 0.887 | 0.904 | 0.904 | HPO-1 FE | 00:02:00 |
| 1 | Pipeline 17 | Snap Decision Tree Classifier | | 0.904 | 0.445 | 0.887 | 0.904 | 0.904 | HPO-1 FE HPO-2 | 00:00:54 |
| 4 | Pipeline 3 | Decision Tree Classifier | | 0.883 | 0.897 | 0.874 | 0.883 | 0.891 | HPO-1 FE | 00:02:10 |
| 3 | Pipeline 4 | Decision Tree Classifier | | 0.883 | 0.897 | 0.874 | 0.883 | 0.891 | HPO-1 FE HPO-2 | 00:03:06 |

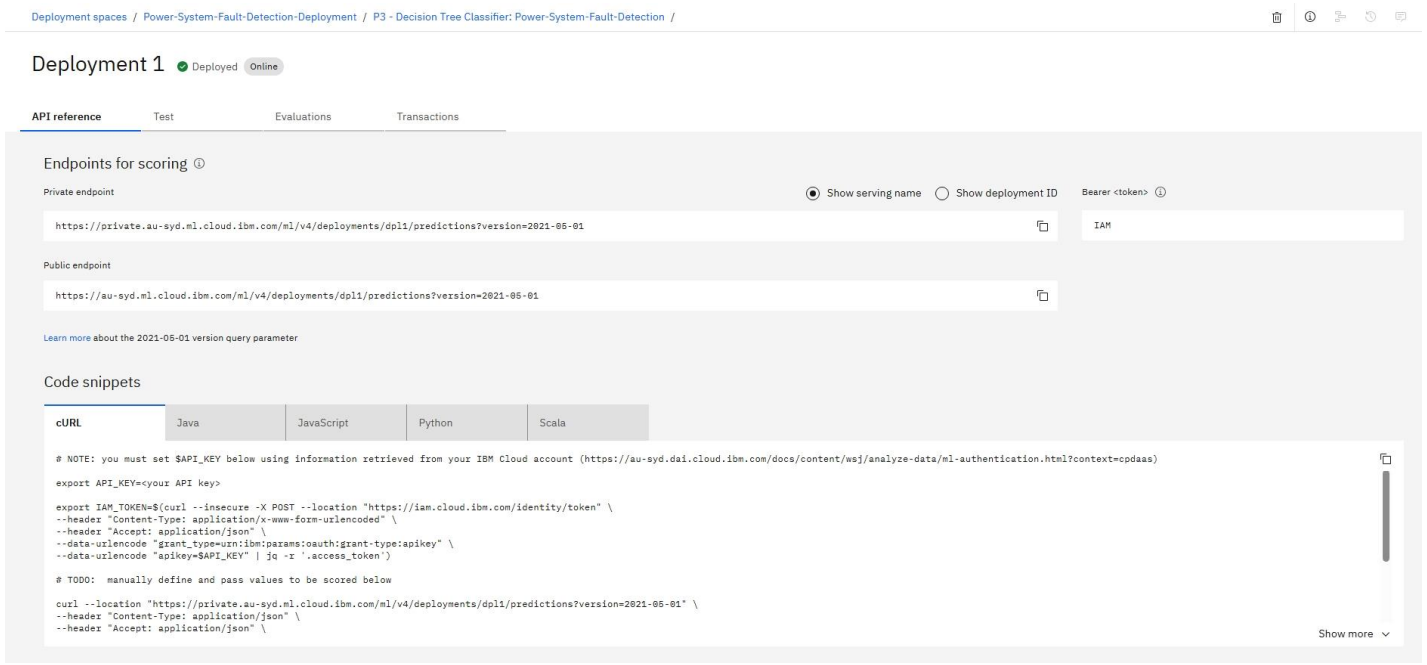
# ALGORITHM & DEPLOYMENT

- **Feature Summary:**
  - As seen below, the more important features are identified as the square of Currents, which are directly correlated with the absolute values of the currents which are the primary parameter for detecting line faults through traditional means
  - Thus, the model allows for detection and classification of faults based on measured electrical parameters



# ALGORITHM & DEPLOYMENT

- Deployment
  - The model is saved as a watsonx.ai Runtime model asset, and promoted to a newly created deployment space
  - A new deployment is then created for the model
  - Public Endpoint: https://au-syd.ml.cloud.ibm.com/ml/v4/deployments/dpl1/predictions?version=2021-05-01
  - API Key: pfcv1TZzZcD3E066ohHuRwHhu_DZwamQV6FcJnL3okBV



# RESULT

As seen below from the confusion matrix, the model has >98% accuracy for labels other than three-phase faults, and only has a low accuracy for distinguishing between LLL and LLLG faults.

| Observed | Predicted | | | | | | |
|---|---|---|---|---|---|---|---|
| | Line-to-ground fault at phase: A | Line-to-line fault at phases: B, C | Line-to-line-to-ground fault at phases: A, B | Line-to-line-to-line fault at phases: A, B, C | Line-to-line-to-line-to-ground fault at phases: A, B, C | No Fault | Percent correct |
| Line-to-ground fault at phase: A | 111 | 0 | 0 | 0 | 0 | 2 | 98.2% |
| Line-to-line fault at phases: B, C | 0 | 100 | 0 | 0 | 0 | 0 | 100.0% |
| Line-to-line-to-ground fault at phases: A, B | 1 | 0 | 113 | 0 | 0 | 0 | 99.1% |
| Line-to-line-to-line fault at phases: A, B, C | 0 | 0 | 0 | 65 | 45 | 0 | 59.1% |
| Line-to-line-to-line-to-ground fault at phases: A, B, C | 1 | 0 | 3 | 29 | 80 | 0 | 70.8% |
| No Fault | 0 | 0 | 0 | 0 | 0 | 237 | 100.0% |
| Percent correct | 98.2% | 100.0% | 97.4% | 69.1% | 64.0% | 99.2% | 89.7% |

# CONCLUSION

- The model is able to classify between single line, line-to-line and three-phase faults with >98% accuracy

**Limitations:**

- Classification between LLL and LLLG faults is not accurate
  - **Workaround:** This can be compensated for using specialized analog instruments such as GFRs (Ground Fault Relays)
- The model is trained with a dataset not containing faults between phases A & B or between phases A & C, or single line faults with either B or C
  - **Workaround:** Given the model was observed to have very high precision, it might be possible to train and utilize three separate models together in parallel, specialized for each phase, in order to ensure coverage of all forms of line faults.

# FUTURE SCOPE

- **Three Phase extension:**
  - The current model is specialized for detecting LG faults for Phase A and LL faults between phases B and C
  - Given availability of training data including faults between other phases, the model can be extended to allow prediction of LG and LL faults between all 3 phases
- **Line Breakage and Transformer Fault:**
  - The current model is specialized for predicting line-line or line-ground faults
  - Line breakage and Transformer failures can also be incorporated given sufficient training data
- **Incorporation of weather and maintenance data:**
  - Rough weather conditions and poor maintenance frequency can contribute to faults, which are real-time datapoints that can be incorporated into the prediction model

# IBM CERTIFICATIONS

# IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild

IBM

Saurya Sircar

Has successfully satisfied the requirements for:

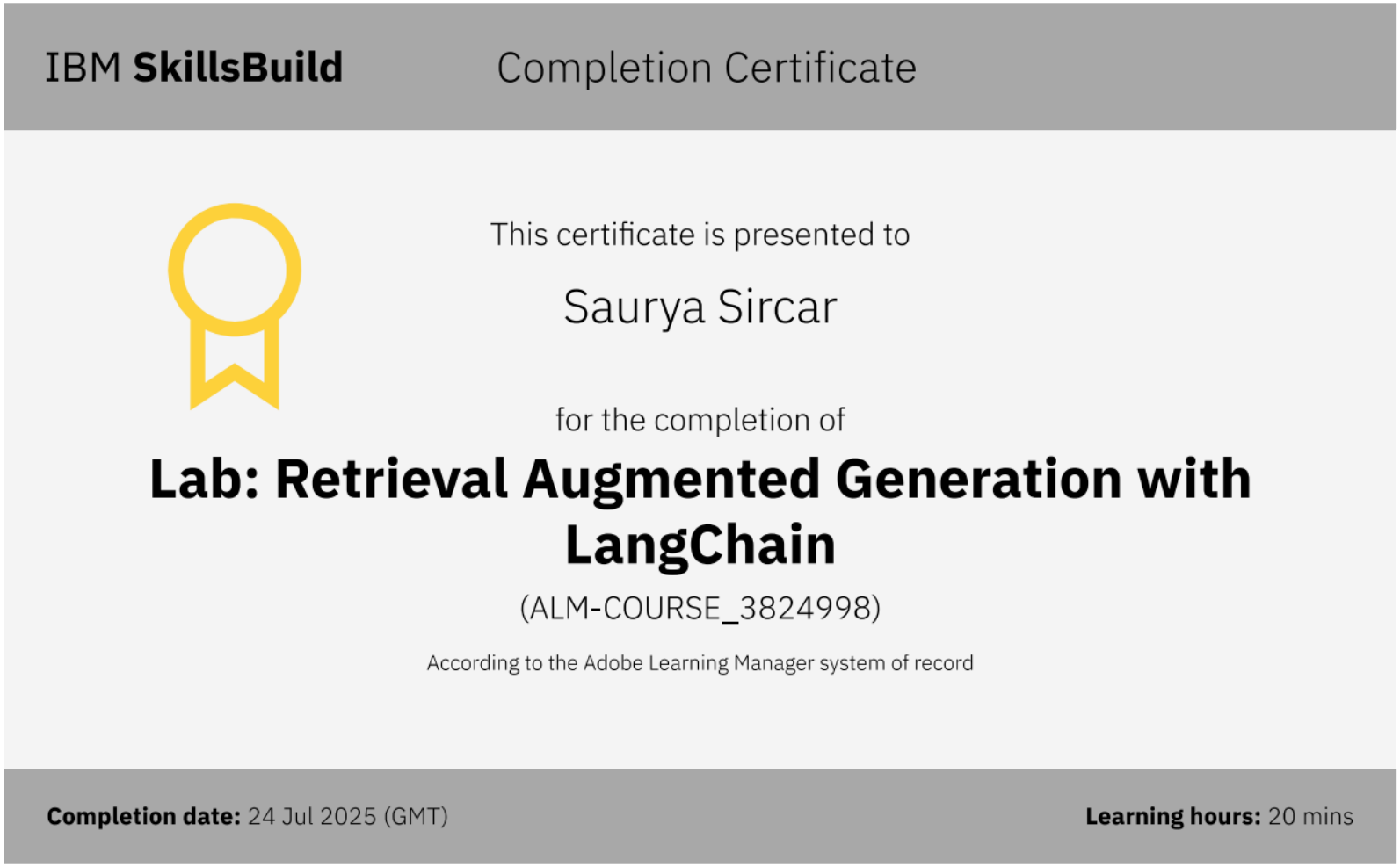
Journey to Cloud: Envisioning Your Solution

Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/d999906b-6f49-4e85-9417-9cd6ce71e7ee

IBM

# IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Saurya Sircar

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

edunet foundation