# CAPSTONE PROJECT

# NETWORK INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

## PROBLEM STATEMENT NO. 40

**Presented By:**
**Yojith Reddy Makireddy – Vellore Institute Of Technology, Andhra Pradesh – Computer Science And Engineering**

edunet foundation

# OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

# PROBLEM STATEMENT

The Challenge: Create a robust network intrusion detection system (NIDS) using machine learning. The system should be capable of analyzing network traffic data to identify and classify various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) and distinguish them from normal network activity. The goal is to build a model that can effectively secure communication networks by providing an early warning of malicious activities.

# PROPOSED SOLUTION

- The system detects and classifies network intrusions using machine learning to distinguish cyber-attacks (DoS, Probe, R2L, U2R) from normal traffic. Key components:
- **Data Collection:**
  - Use Kaggle's labeled network traffic dataset.
  - Extract features like protocol type, service, source/destination bytes, etc.

**Data Preprocessing:**

- Handle missing values and encode categorical data.
- Normalize features and balance class distribution.

**Machine Learning Algorithm:**

- Train classifiers Decision Tree*.
- Use cross-validation and hyperparameter tuning.

**Deployment:**

- Train and deploy models on **IBM Watson Studio**.
- Store datasets on **IBM Cloud Object Storage**.

**Evaluation:**

- **Assess the model's performance using appropriate metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or other relevant metrics.**
- Analyze class-wise performance and ROC curves.
- Result: Achieved ~99% accuracy.

edunet foundation

# SYSTEM APPROACH

## System Requirements:

- Python, Jupyter Notebook
- IBM Watson Studio (Lite)
- IBM Cloud Object Storage
- scikit-learn, pandas, matplotlib

## Libraries Used:

- Pandas , Numpy , matplotlib, seaborn, scikit-learn, Decision Tree classifier

**Architecture Flow:**

1. Data Collection
2. Preprocessing
3. Model Training
4. Evaluation
5. Deployment on IBM Cloud

# ALGORITHM & DEPLOYMENT

**Algorithm Selection:**

The best-performing model was the **Snap Decision Tree Classifier**, selected after evaluating multiple pipelines in IBM Watson Studio AutoAI. This model was chosen due to its high classification accuracy (0.995) and ability to handle categorical and numerical features efficiently. Decision Trees also offer interpretability, which is crucial in intrusion detection tasks.

**Data Input:**

Protocol type, service, flag

Source and destination bytes

Class labels: Normal, DoS, Probe, R2L, U2R

**Training Process:**

Data was uploaded and preprocessed within **IBM Watson Studio AutoAI.Prediction Process:**

The dataset was split using cross-validation and Hyperparameter Optimization (HPO-1) was applied in some pipelines.

Out of all pipelines, **Pipeline 2** using Snap Decision Tree achieved the highest accuracy of **99.5%** in **6 seconds**.

## Prediction Process:

Once trained, the Snap Decision Tree Classifier predicts whether a given network activity is normal or belongs to one of the four attack categories. The model can be used for batch processing or real-time input using deployed APIs in Watson Studio.

## Deployment:

The best model was deployed on **IBM Cloud** using **Watson Studio's AutoAI** feature.

Dataset storage and retrieval was managed through **IBM Cloud Object Storage**.

The final model is ready for integration into a security dashboard or alerting system for real-time threat detection.

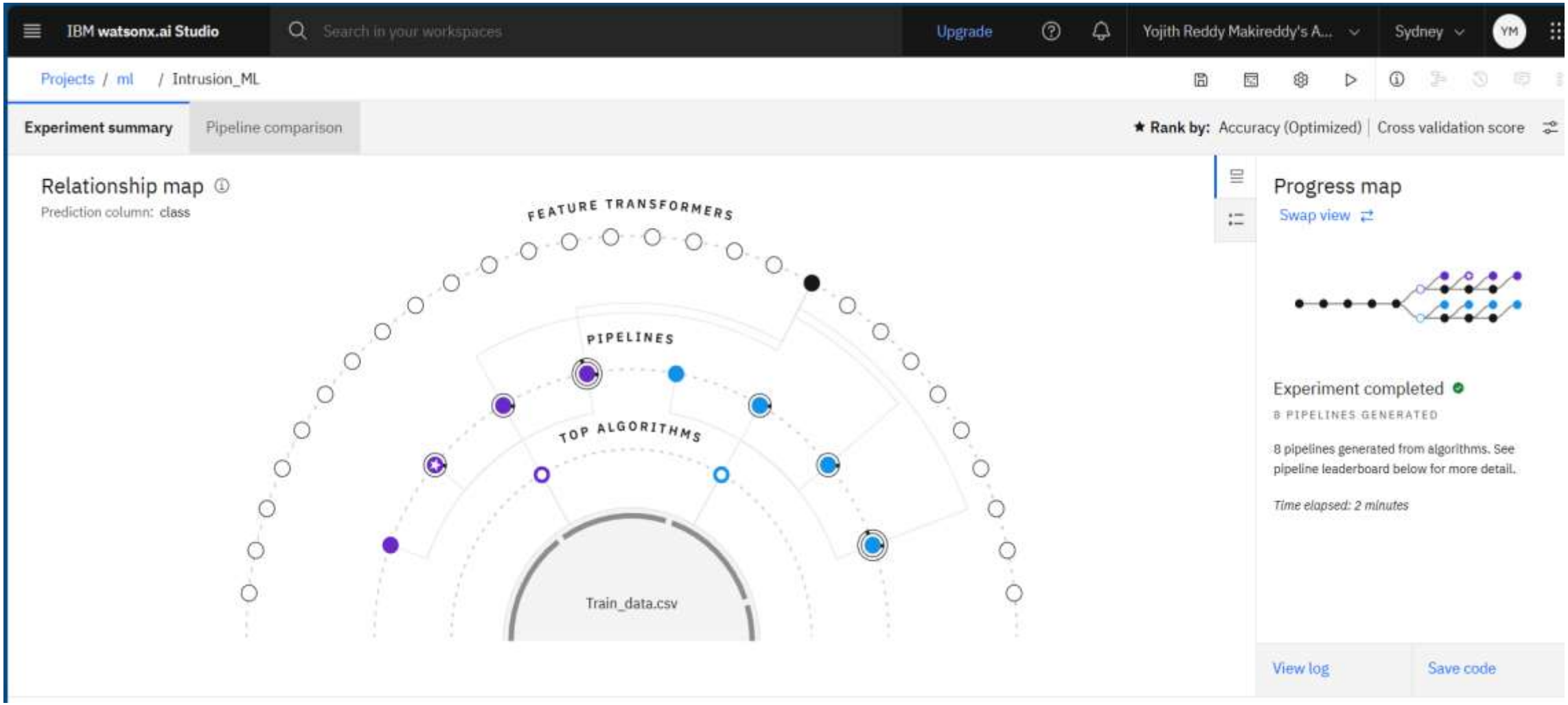
edunet foundation

# RESULT

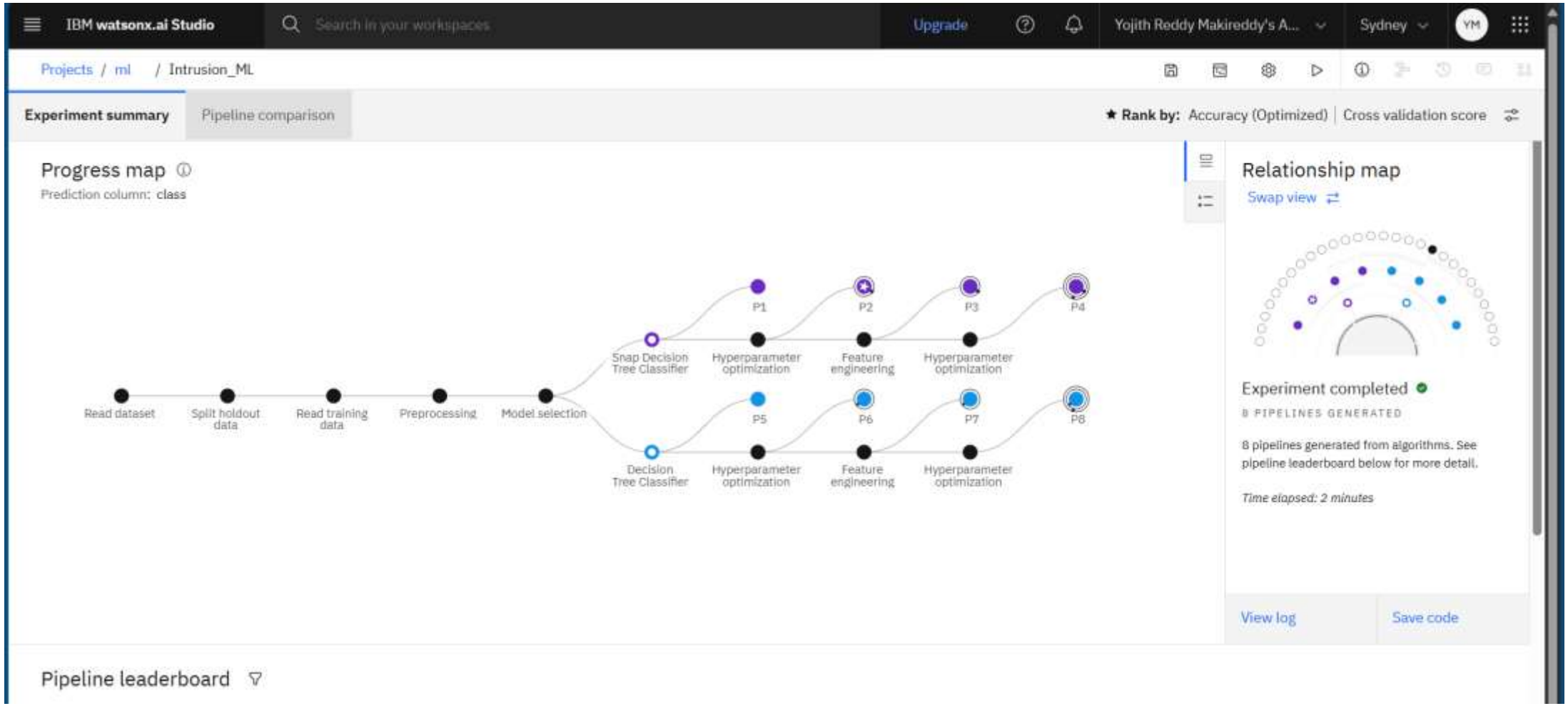
- **Result:**

- ● Multiple models were trained and evaluated using IBM Watson Studio AutoAI.
  ● The best-performing model was **Pipeline 2** using the **Snap Decision Tree Classifier**.
  ● Achieved an optimized **accuracy of 99.5%** using **cross-validation**.
  ● Model enhancements included **Hyperparameter Optimization (HPO-1)**.
  ● Build time for the top model was **6 seconds**, showcasing rapid training performance.
  ● Model is suitable for real-time classification of network traffic and detecting cyber threats.

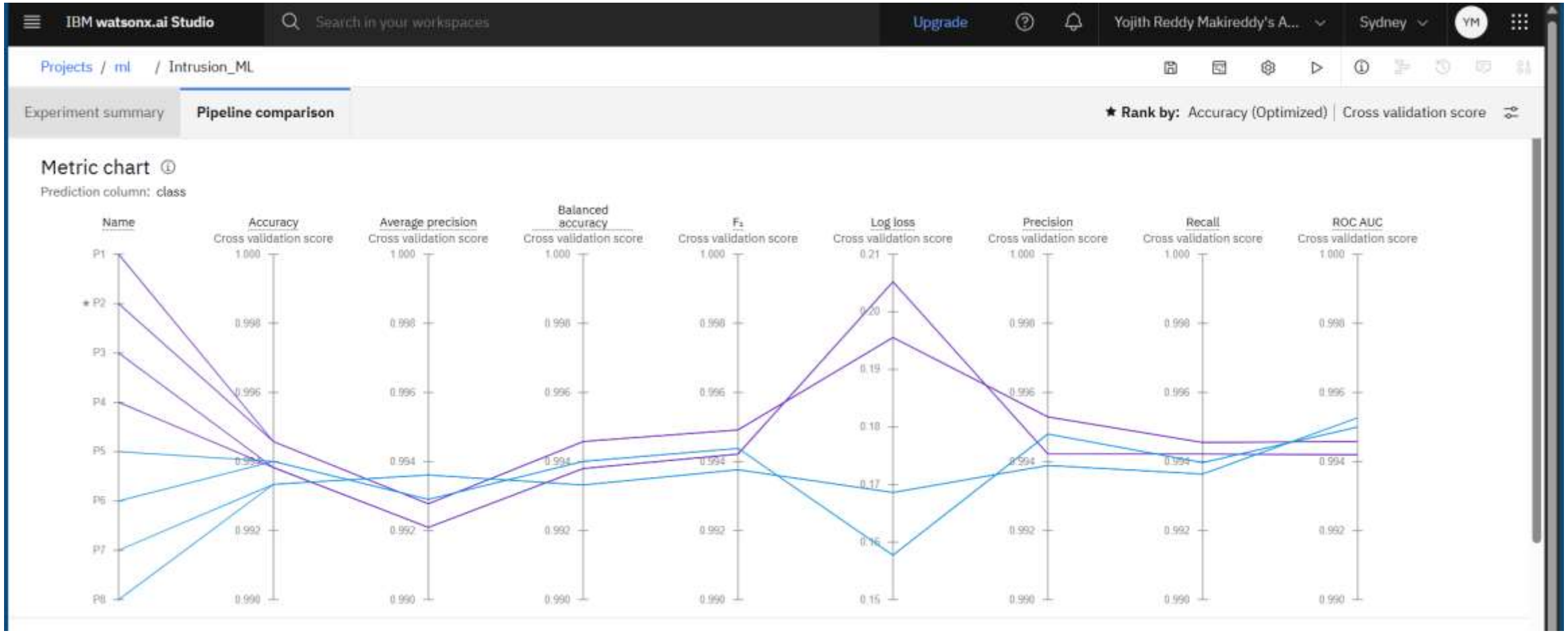
# RESULT:



# RESULT:



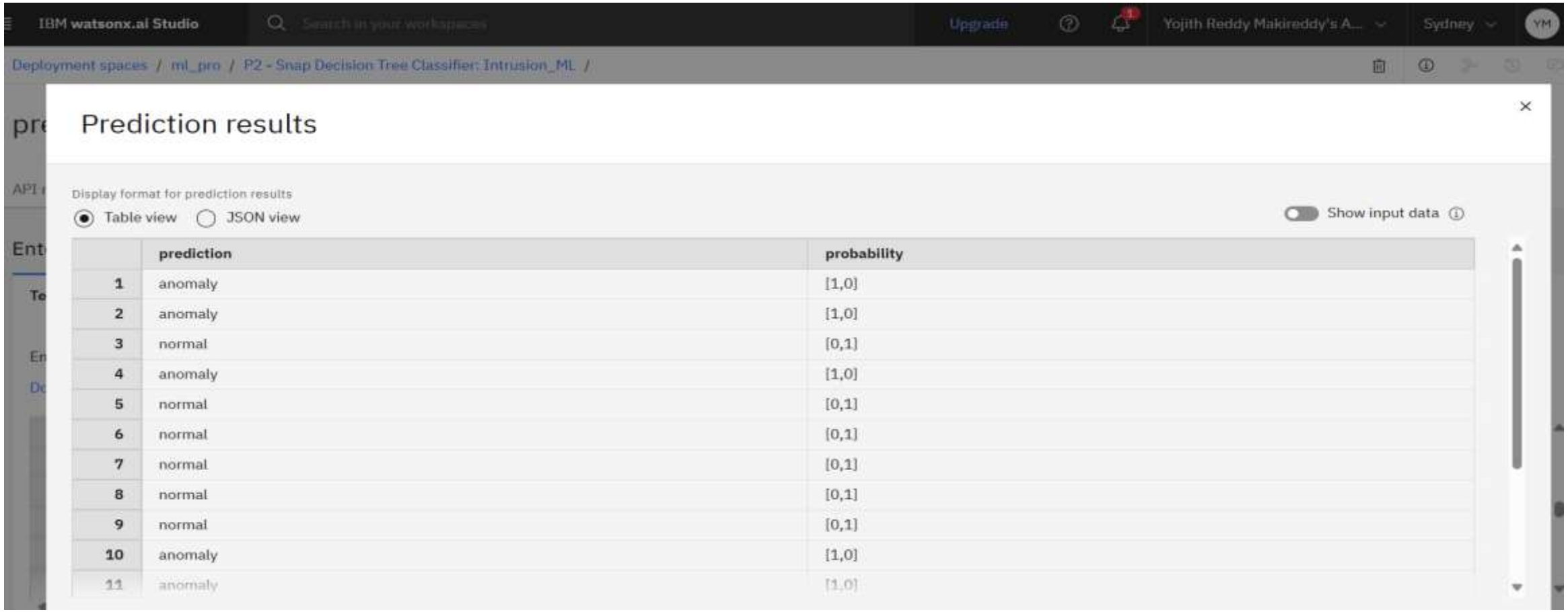
# RESULT:



# RESULT:

| | | | | | | |
|---|---|---|---|---|---|---|
| ★ | 1 | **Pipeline 2** | Snap Decision Tree Classifier | 0.995 | HPO-1 | 00:00:06 |
| | 2 | **Pipeline 1** | Snap Decision Tree Classifier | 0.995 | *None* | 00:00:02 |
| | 3 | **Pipeline 6** | Decision Tree Classifier | 0.994 | HPO-1 | 00:00:07 |
| | 4 | **Pipeline 5** | Decision Tree Classifier | 0.994 | *None* | 00:00:03 |
| | 5 | **Pipeline 4** | Snap Decision Tree Classifier | 0.994 | HPO-1 FE HPO-2 | 00:00:37 |
| | 6 | **Pipeline 3** | Snap Decision Tree Classifier | 0.994 | HPO-1 FE | 00:00:32 |
| | 7 | **Pipeline 8** | Decision Tree Classifier | 0.993 | HPO-1 FE HPO-2 | 00:00:47 |
| | 8 | **Pipeline 7** | Decision Tree Classifier | 0.993 | HPO-1 FE | 00:00:42 |

# RESULT - PREDICTION ON TEST CASES:




| | prediction | probability |
|---|---|---|
| 1 | anomaly | [1,0] |
| 2 | anomaly | [1,0] |
| 3 | normal | [0,1] |
| 4 | anomaly | [1,0] |
| 5 | normal | [0,1] |
| 6 | normal | [0,1] |
| 7 | normal | [0,1] |
| 8 | normal | [0,1] |
| 9 | normal | [0,1] |
| 10 | anomaly | [1,0] |
| 11 | anomaly | [1,0] |

# CONCLUSION

- The proposed system successfully detects and classifies various types of network intrusions using machine learning.
- The Snap Decision Tree Classifier achieved **99.5% accuracy**, demonstrating high effectiveness in identifying threats.
- IBM Watson Studio and Cloud Object Storage ensured smooth model training, deployment, and scalability.
- The system provides a reliable early warning mechanism to enhance network security and prevent potential cyber-attacks.
- This approach proves that ML-based intrusion detection can be both accurate and efficient when deployed on a cloud platform.

# FUTURE SCOPE

- Integrate the system with **real-time monitoring tools** for live intrusion alerts and log analysis.
- Extend support for **deep learning models** (e.g., LSTM, CNN) to improve detection of complex attack patterns.
- Implement **automated threat response** actions using IBM Cloud Functions or edge computing.
- Expand the dataset with **recent and diverse traffic sources** to improve generalization across networks.
- Build a comprehensive **dashboard interface** with visual analytics for security teams and administrators.

# REFERENCES


- Kaggle Dataset – Network Intrusion Detection
https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection
- IBM Watson Studio Documentation
https://dataplatform.cloud.ibm.com/docs
- IBM Cloud Object Storage
https://www.ibm.com/cloud/object-storage
- Scikit-learn Documentation
https://scikit-learn.org/stable/

# IBM CERTIFICATIONS - GETTING STARTED WITH ARTIFICIAL INTELLIGENCE

In recognition of the commitment to achieve professional excellence


## Yojith Reddy Makireddy

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/3f684384-f2f9-445a-82b0-a46cae2db23e

IBM

edunet foundation

# IBM CERTIFICATIONS - JOURNEY TO CLOUD: ENVISIONING YOUR SOLUTION

In recognition of the commitment to achieve professional excellence


## Yojith Reddy Makireddy

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/6004533f-4341-4d99-8b41-92659386c0b3

IBM

edunet foundation

# IBM CERTIFICATIONS

# RAG WITH LANGCHAIN

IBM **SkillsBuild**

Completion Certificate

This certificate is presented to

Yojith Reddy Makireddy

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

edunet foundation