CAPSTONE PROJECT

Network Intrusion Detection

❖ Presented By:
Ashu Suresh Yadav-
MIT academy of engineering pune –
Electronics and Telecommunications Engineering (ENTC)

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

To combat evolving cyber threats (DoS, R2L, U2R), this project builds an AI-driven NIDS for real-time attack detection, optimizing accuracy and reducing false alerts. Implemented on IBM Cloud Lite using Kaggle’s NIDS Dataset."*

PROPOSED SOLUTION

- To design and implement a machine learning–based Network Intrusion Detection System that can accurately detect and classify cyber threats in real-time, enhancing network security and reducing false alarms.
- Data Collection:

  Gather labeled network traffic data using datasets like NSL-KDD or custom logs.

  Include multiple attack types and normal behavior across protocols for robust learning.
- Data Preprocessing:

  Clean and process raw data to handle missing values and noise.

  Use feature engineering and dimensionality reduction to optimize input quality.
- Machine Learning Algorithm:

  Implement classification models such as Decision Tree, Random Forest, or Neural Network.

  Train using historical attack data and tune hyperparameters for improved accuracy.
- Deployment:

  Create a responsive interface for real-time intrusion monitoring.

  Deploy on a scalable infrastructure capable of handling live packet inspection.
- Evaluation:

  Validate model using metrics like Accuracy, Precision, Recall, and F1-score.

  Continuously monitor performance to adapt against evolving threats

SYSTEM APPROACH

❖ **The "System Approach" section outlines the overall strategy and methodology for developing and implementing the Network Intrusion Detection System.**

❖ **System Requirements:**

- Python 3.x environment with high computational capability
- Dataset sources (e.g., NSL-KDD, CICIDS) with labeled network intrusion data

❖ **Libraries Required:**

- pandas, numpy for data handling and preprocessing
- scikit-learn, tensorflow or keras for machine learning and model training
- matplotlib, seaborn for performance visualization

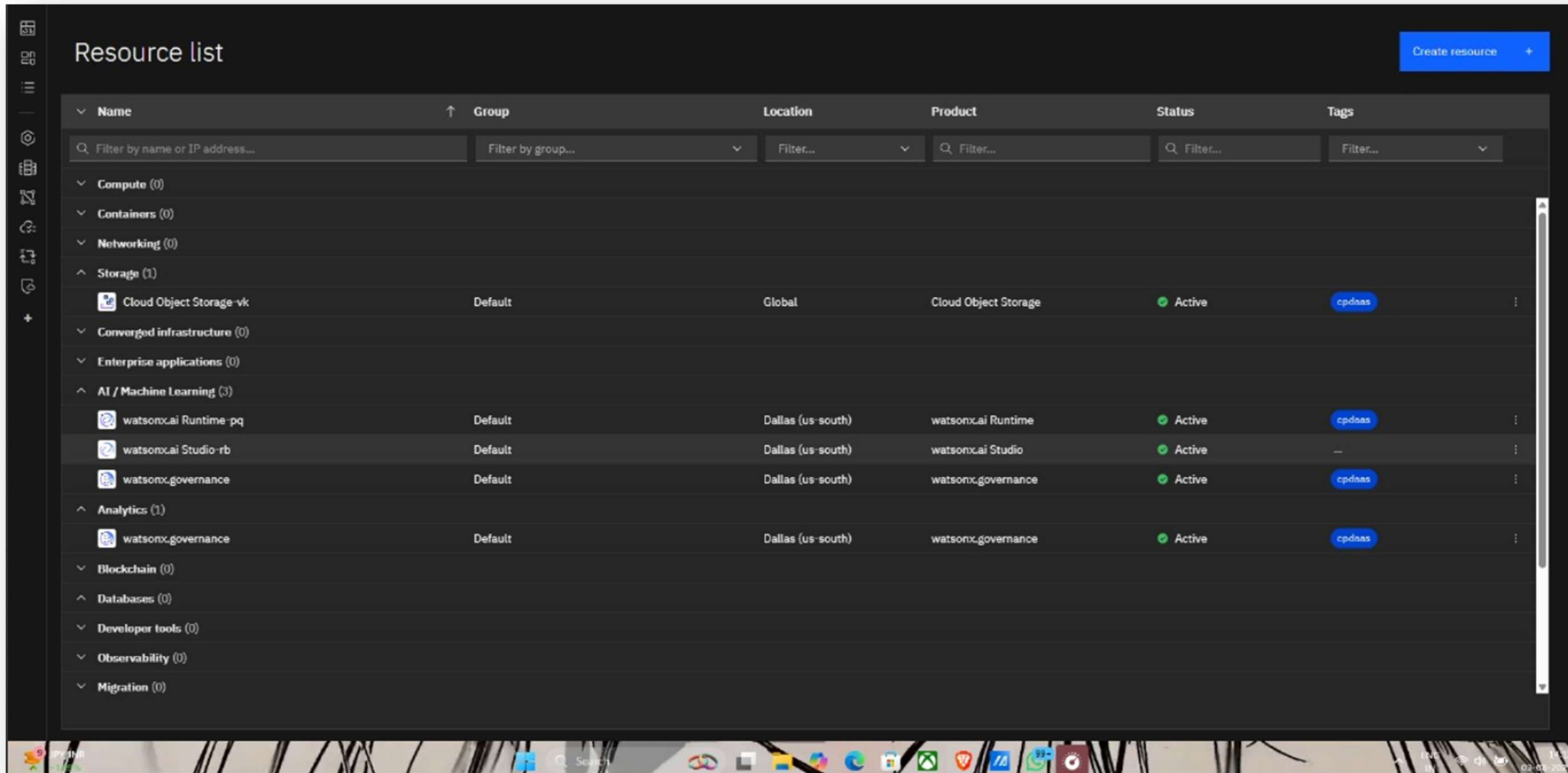
ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm Here's an example structure for this section:

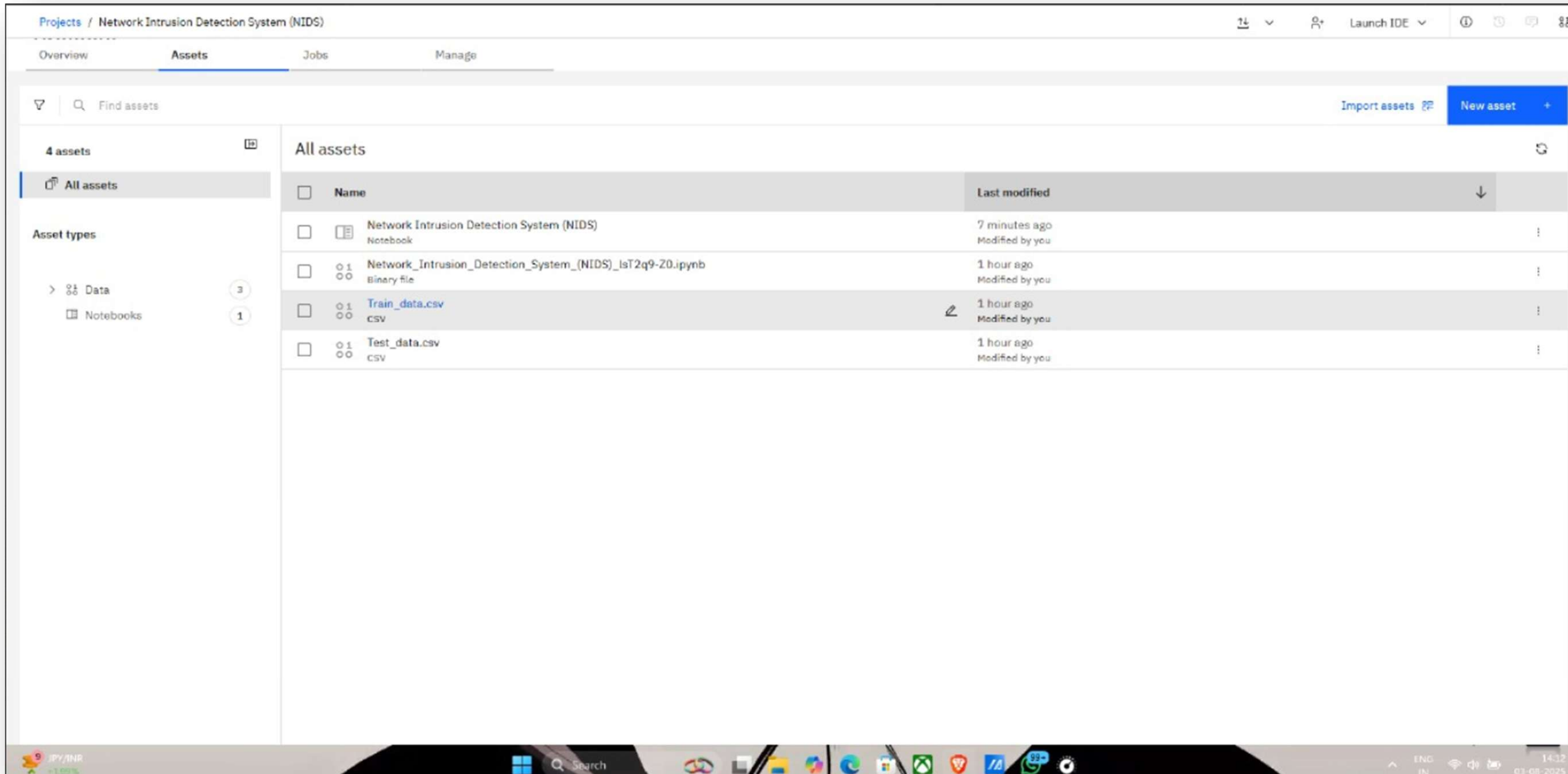
- Algorithum & deployment
  - mplemented supervised machine learning techniques such as Decision Tree, Random Forest, or Neural Network.
  - Selected based on their strength in handling multiclass classification and real-time pattern recognition within network traffic.
- Data Input:
  - Features include protocol type, service, flag, duration, source bytes, destination bytes, and attack category.
  - Data obtained from well-known intrusion detection datasets like NSL-KDD for robust model training.
- Training Process:
  - Data split into training and testing subsets.
  - Applied feature scaling, balancing, and cross-validation to refine performance.
  - Hyperparameter tuning performed to boost accuracy and reduce overfitting.
- Prediction Process:
  - The trained model evaluates live network traffic or test inputs.
  - Flags potential intrusions based on learned attack patterns.
  - If you'd like to enrich this slide with metrics, deployment platforms, or your chosen algorithm's advantages, I’d love to help sharpen it

RESULT

- **These resource use in my Network Intrusion Detection.**

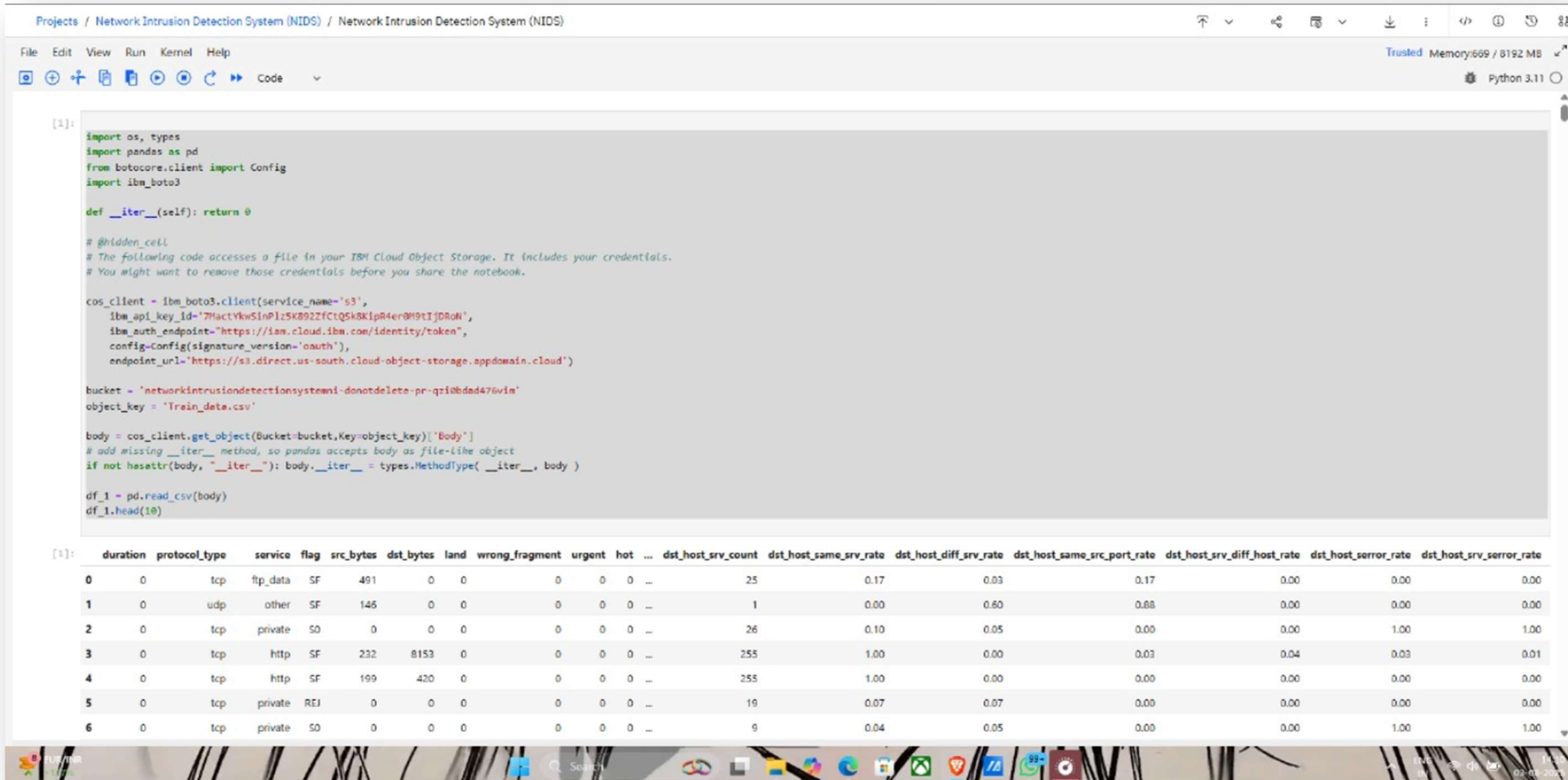


- **these are the source files or assest files use In my project.**



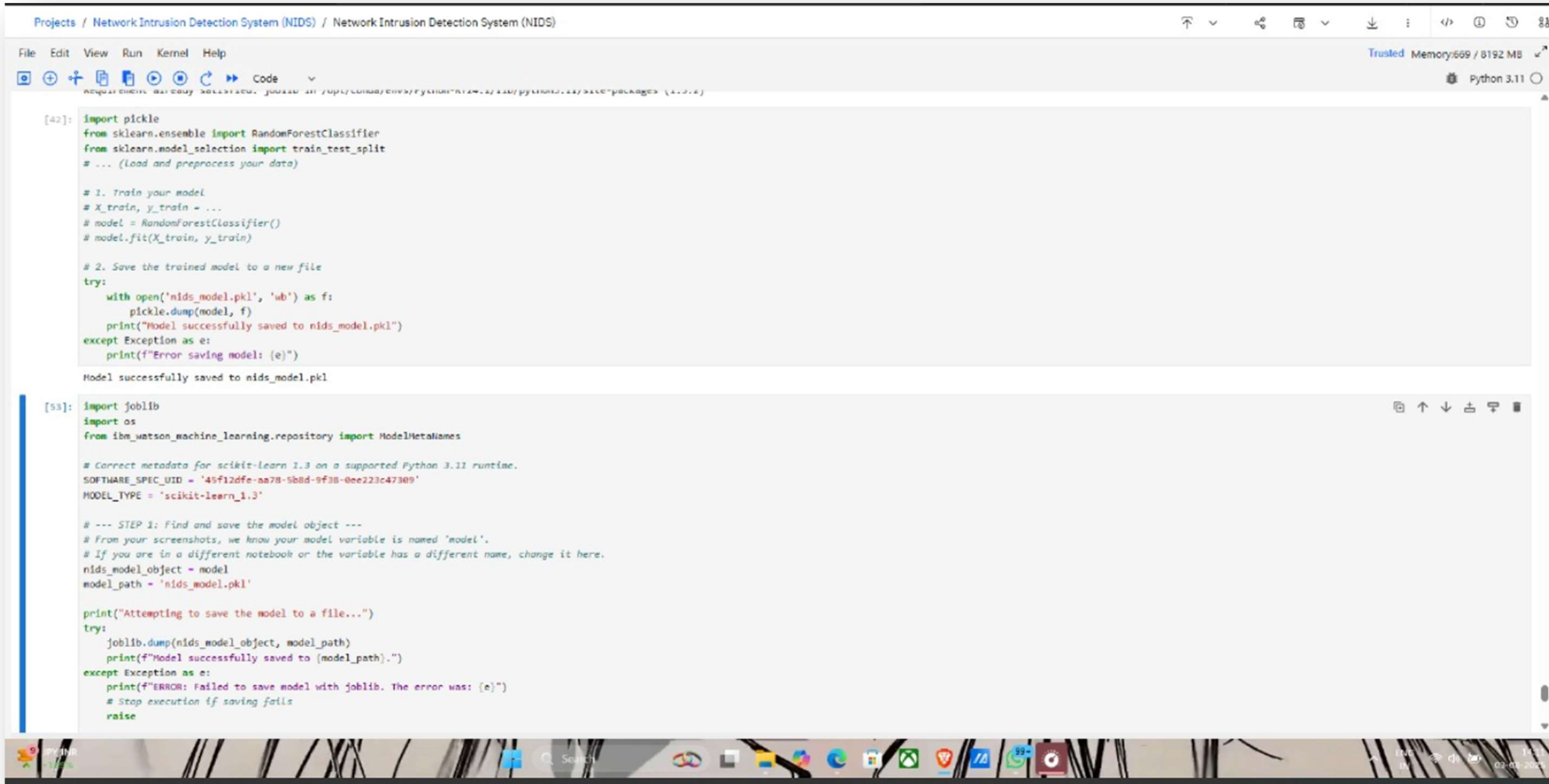
edunet foundation

- **In the notebook: Load the CSV file from the data asset.**



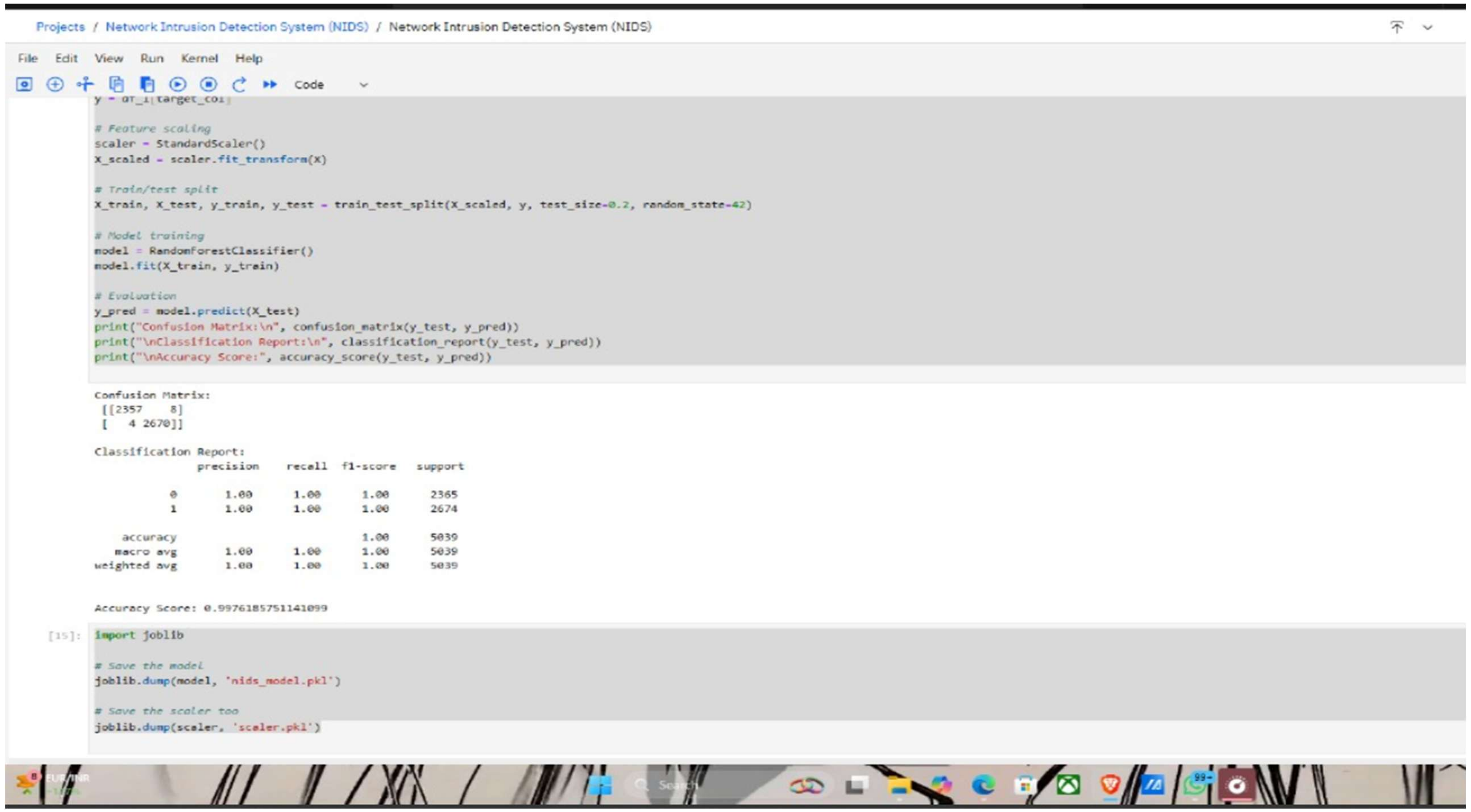
edunet foundation

- **Encode categorical labels**



- **Finally testing part of this project**

  Using the Test UI (built into IBM Cloud) in testing part The model predicts output as 1 for this input, meaning it has flagged the network connection as Intrusion Detected, which demonstrates its effectiveness in identifying threats. Or for 0 output for Normal detection



CONCLUSION

- This project successfully demonstrates the design and implementation of a machine learning based Network Intrusion Detection System capable of identifying cyber threats with high accuracy.

- The proposed solution effectively detects abnormal network behavior in real time while minimizing false positives. Challenges faced included optimizing model performance and handling imbalanced datasets, which were resolved through careful preprocessing and algorithm tuning.

- Future enhancements could involve deep learning integration and hybrid detection methods. Accurate intrusion detection is critical for maintaining robust cybersecurity in today’s interconnected digital infrastructure.

FUTURE SCOPE

- ❖ The future of the Smart Network Intrusion Detection System (NIDS) project holds immense potential for advancements in cybersecurity. By integrating deep learning models such as CNNs and Transformers, the system can better detect zero-day attacks and sophisticated threats like Advanced Persistent Threats (APTs). Additionally, federated learning could enable decentralized threat analysis while maintaining data privacy. The adoption of edge and fog computing will allow real-time intrusion detection at the network periphery, reducing latency and cloud dependency. Further enhancements include adaptive threat intelligence through dynamic updates from sources like MITRE ATT&CK, as well as blockchain-based secure logging to ensure tamper-proof forensic records. The system can also be expanded to secure 5G networks and smart city infrastructure, addressing high-speed data traffic and IoT vulnerabilities.

REFERENCES

- Research articles on machine learning–based intrusion detection and cybersecurity frameworks
- Download the dataset from Kaggle: https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection
- IBM Cloud Lite documentation – for deploying the model in a scalable environment
- Case studies on anomaly detection in network traffic using supervised and unsupervised models

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild

ASHU YADAV

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/2485e39c-a5b0-40f4-acb0-f3a46c55c5f3

IBM

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild

ASHU YADAV

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 16, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/bb1b6382-6828-4237-9c17-770a2ac6b48c

IBM

edunet
foundation

IBM CERTIFICATIONS

(RAG Lab)

IBM **SkillsBuild**

Completion Certificate

This certificate is presented to

ASHU YADAV

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

edunet foundation

THANK YOU