CAPSTONE PROJECT

Network Intrusion Detection

❖Presented By:

Yadnyesh Pramodkumar kumbhar-
MIT AOE pune –
Electronics and Telecommunications Engineering (ENTC)

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

- Cyber threats like DoS, R2L, and U2R attack network security daily.

This project aims to build a smart NIDS that detects suspicious activity in real-time using ML techniques—balancing accuracy and minimal false alerts.

Platform: IBM Cloud Lite.

Dataset: Kaggle – 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. Here's how the project structure is designed**

❖ **System Requirements:**

- Python 3.x environment with high computational capability
- A machine or cloud platform capable of handling real-time traffic analysis
- 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
- socket, scapy or similar for live packet sniffing (if deployed in real-time)
- Let me know if you want me to design matching points for your next slide—perhaps on “Model Architecture” or “System

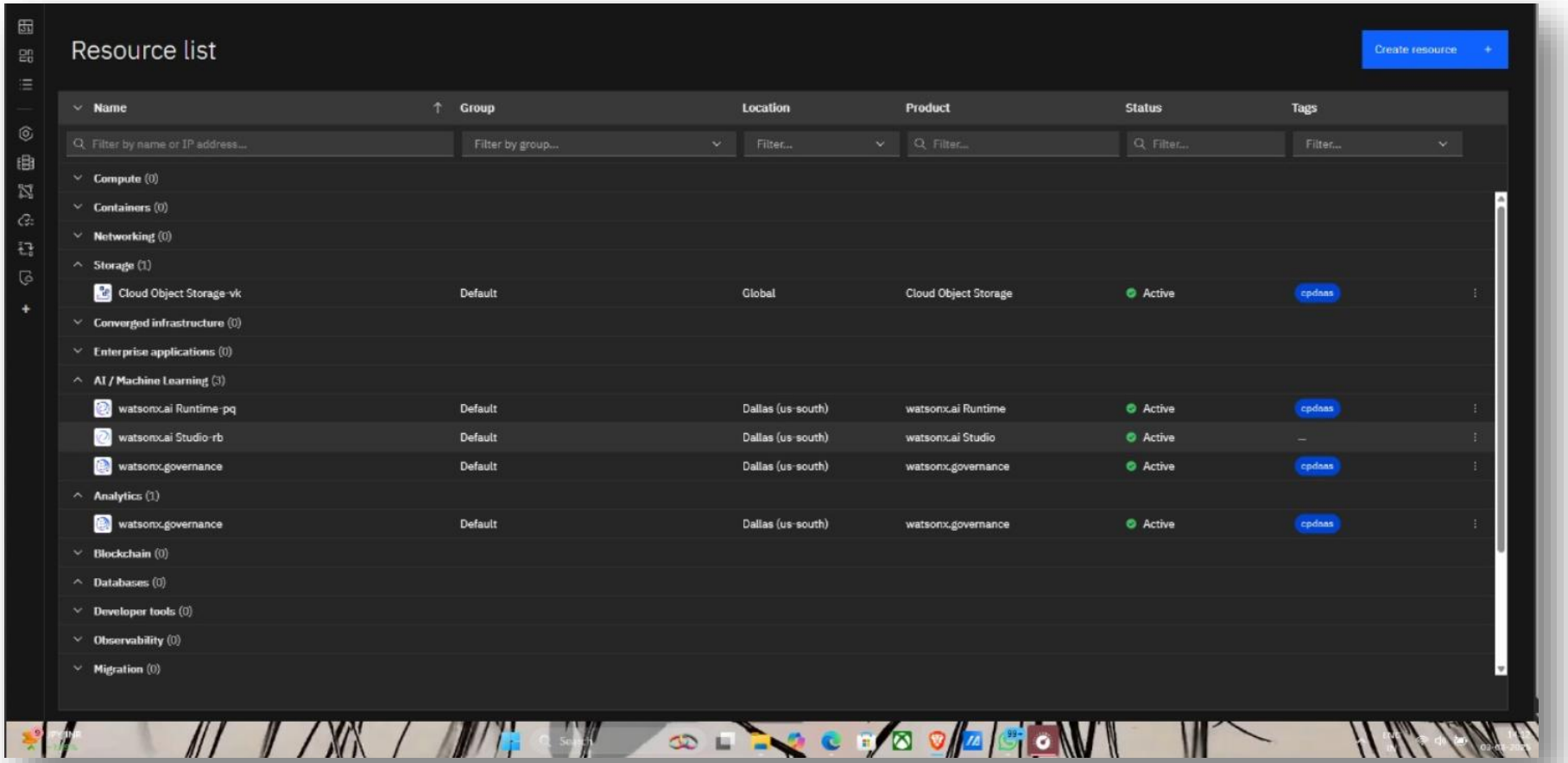
ALGORITHM & DEPLOYMENT

❖ 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

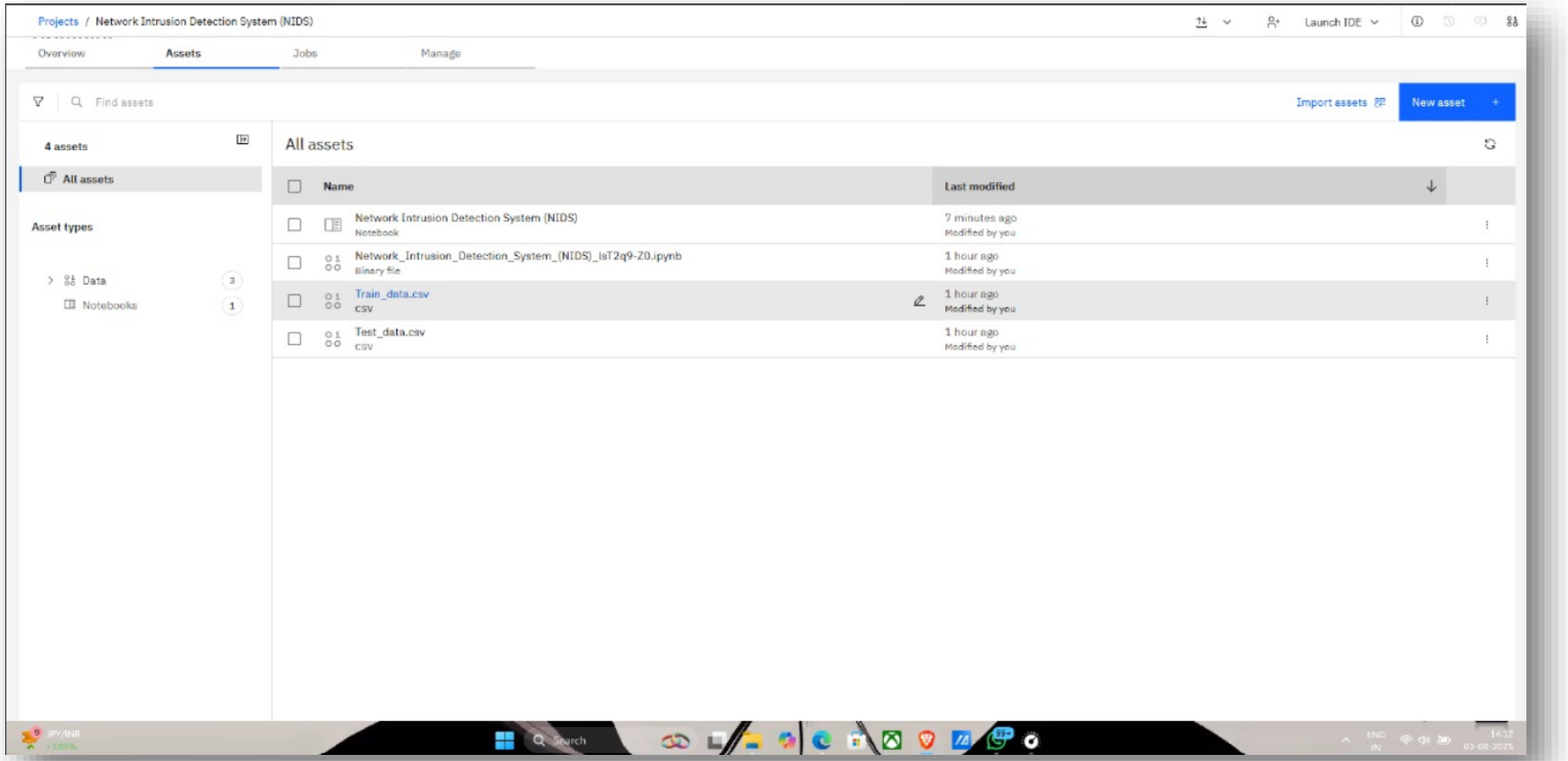
RESULT

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

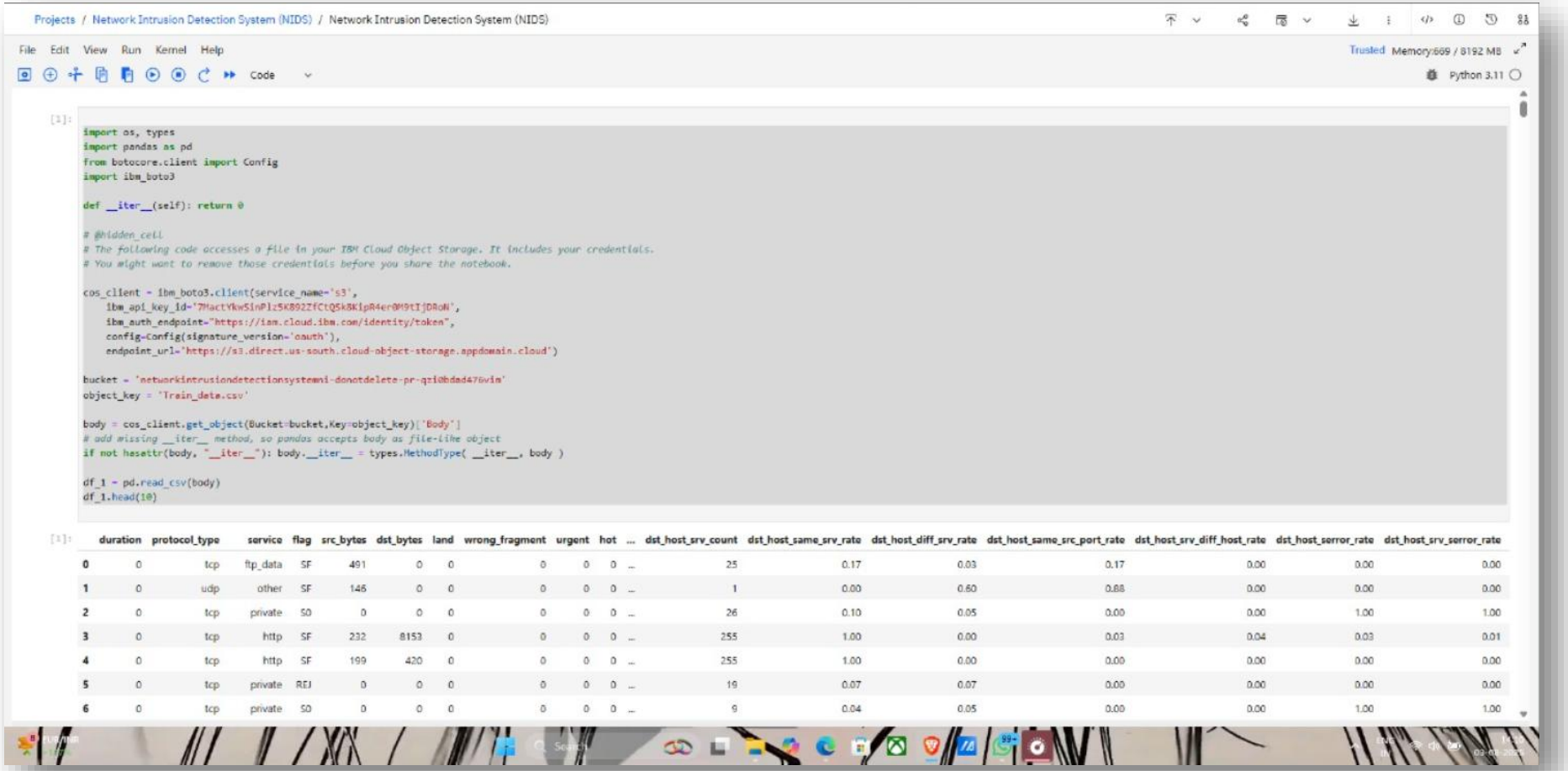


edunet foundation

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

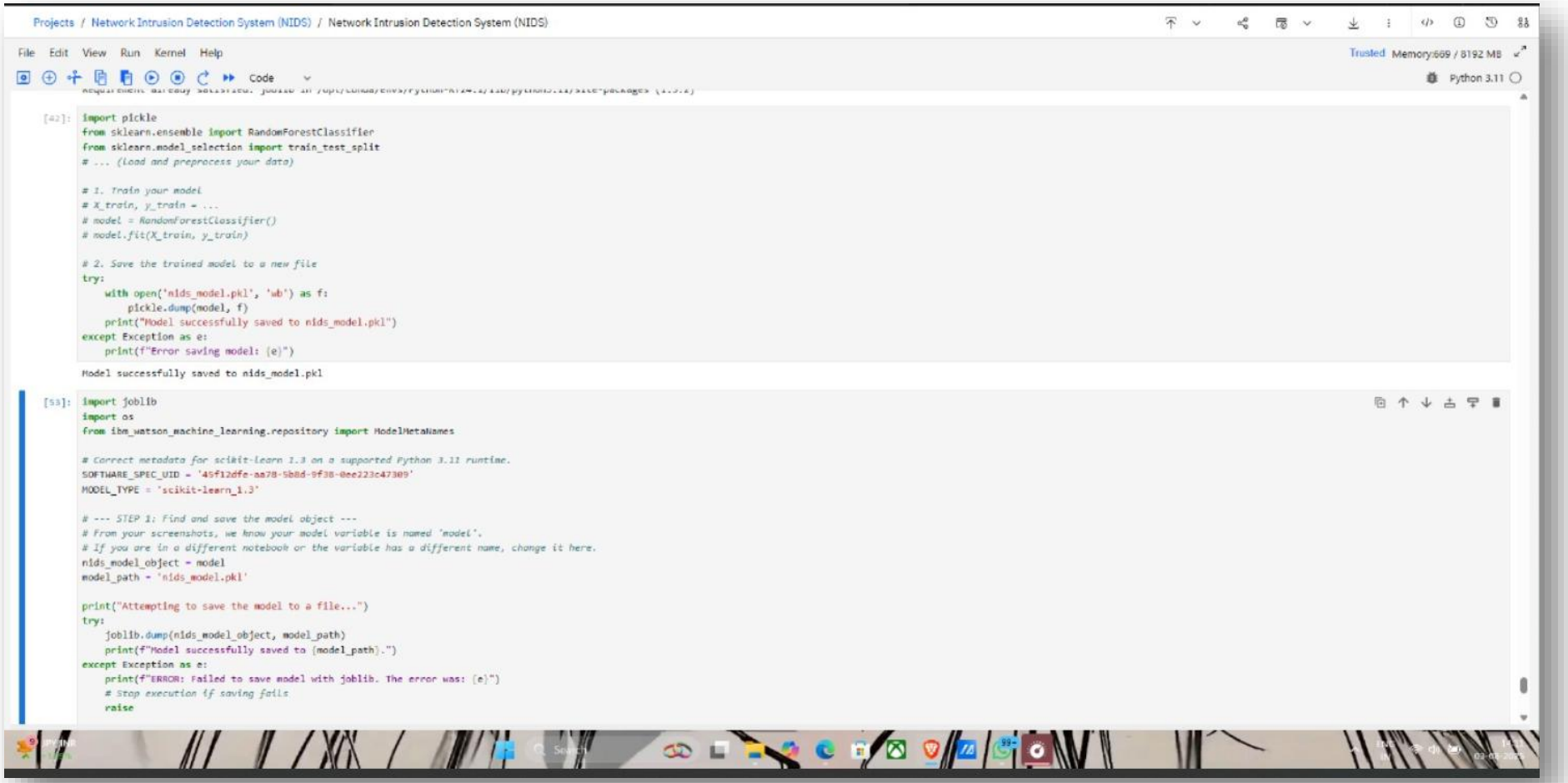


- **In the notebook: Load the CSV file from the data asset.**



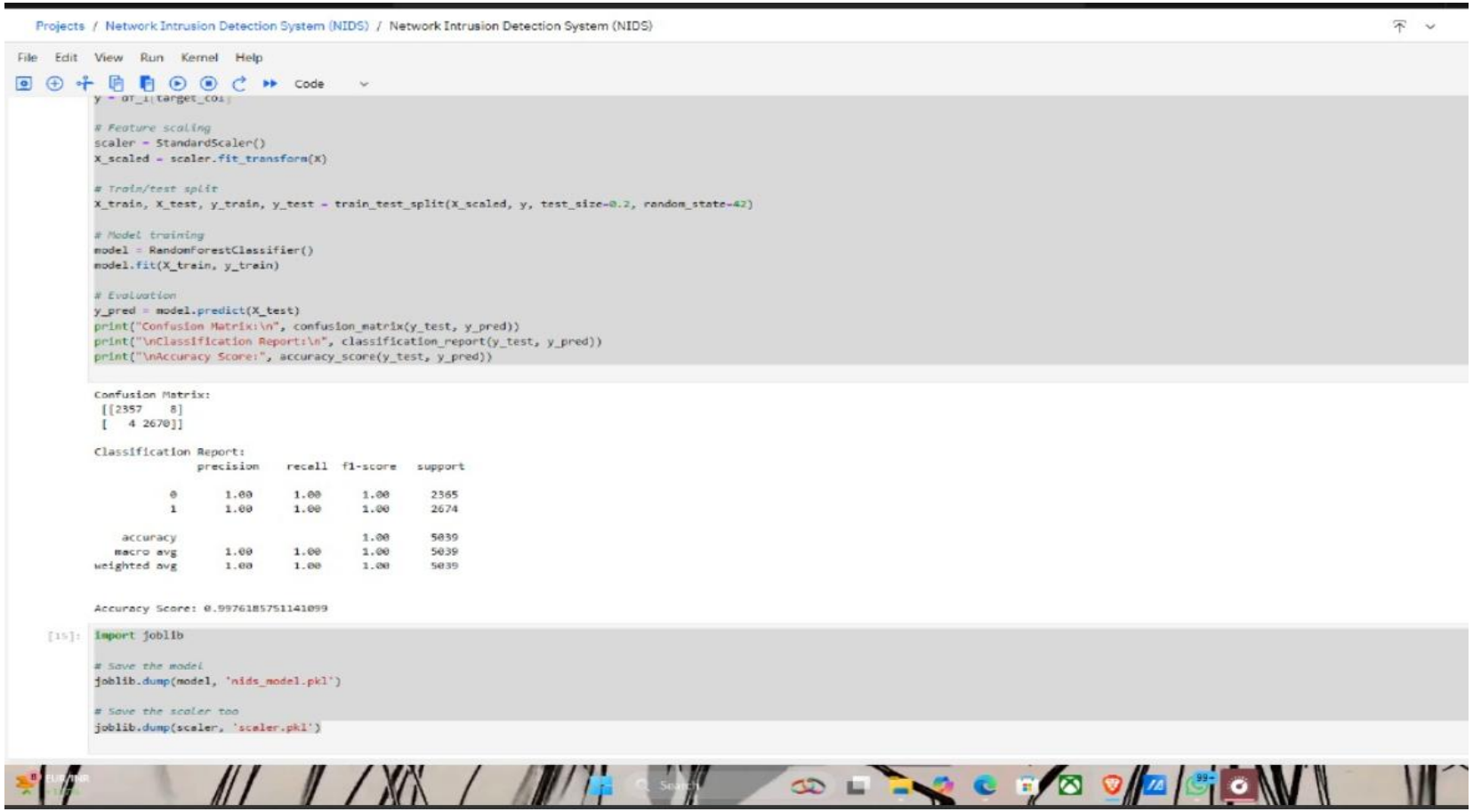
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	0.03	0.17	0.00	0.00	0.00
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	0.60	0.88	0.00	0.00	0.00
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	0.05	0.00	0.00	1.00	1.00
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	0.00	0.03	0.04	0.03	0.01
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	0.00	0.00	0.00	0.00	0.00
5	0	tcp	private	REJ	0	0	0	0	0	0	...	19	0.07	0.07	0.00	0.00	0.00	0.00
6	0	tcp	private	S0	0	0	0	0	0	0	...	9	0.04	0.05	0.00	0.00	1.00	1.00

- **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 traffic no any Intrusion Detected. This is your actual test result — the model detects if the input traffic is suspicious.



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

- This system offers high scalability by incorporating varied datasets, optimizing detection algorithms, and extending deployment to smart urban networks. The integration of technologies like edge computing and next-gen ML models will strengthen its accuracy, responsiveness, and adaptability to evolving cyber threats.

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
- Academic papers focused on data preprocessing, feature selection, and performance evaluation techniques in NIDS
- Official documentation for tools/libraries used (e.g., Scikit-learn, NumPy, pandas)

IBM CERTIFICATIONS

(Getting Started with Artificial Intelligence)

IBM CERTIFICATIONS

(Journey to Cloud)

edunet foundation

IBM CERTIFICATIONS

(RAG Lab)

This certificate is presented to

YADNYESH KUMBHAR

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

edunet foundation

THANK YOU

edunet foundation