**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Online Payment Fraudulent Detection Using Machine Learning in Python**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Team Members**:

1. Chinthala Upagna Reddy (USN: 1RVU22CSE043)

2. Kotnada Muktha (USN: 1RVU22CSE085)

3. Shravya Hanasoge (USN: 1RVU22CSE150)

4. Sunkesula Himaja (USN: 1RVU22CSE172)

**Abstract**

The project aims to develop a fraud detection system for online payments using Python-based machine learning. Methodology includes exploratory data analysis, supervised learning algorithms, and evaluation metrics to gauge model efficacy. Once trained, the final model is deployed for real-time fraud detection, aiming to reduce financial losses and ensure secure transactions.

1. **Introduction**

As we are in the digital age where almost anything and everything is available and serviced online, payments are increasingly becoming digitalized. Online payments are extremely beneficial as they save the users a lot of time. Although there are many advancements made in payment processing, it is accompanied with payment fraud. As online transactions continue to grow, risk of fraudulent activities in online payments have become an area of increasing concern for many businesses. Thus, countermeasures have to be taken.

One such measure includes designing an ML model to identify fraudulent transactions.

1. **Related Work**

Previous implementations available on platforms like GitHub and Kaggle are referenced for addressing similar fraud detection challenges. Our project focuses on training a specific model and conducting extensive comparisons with alternative models. While we do not aim to discredit or surpass previous efforts, our emphasis lies in contributing to the existing body of knowledge by rigorously evaluating the performance of different models in the context of fraud detection, with a particular focus on training methodologies.

1. **Dataset**

The dataset has been taken from Kaggle from an account named jainilcoder. The dataset comprises transactional data collected from an online payment system.

* 1. **Data Preprocessing**

The dataset that we chose contains 63Lakh+ records. However, we took a subset of the dataset with 16000 samples where we could find a balance between number of Frauds detected (0 and 1 balanced), due to computational constraints.

1. **Methodology**

Our approach is grounded in machine learning techniques for fraud detection, leveraging supervised learning algorithms to classify transactions as fraudulent or legitimate based on historical data patterns. We adopt a data-driven approach, utilizing features extracted from transactional data to train predictive models capable of identifying fraudulent activities.

We employed supervised learning algorithms, including Decision Trees, random forest, Multi Lyer Perceptron and support vector machines (SVM), to build fraud detection models.

We even explored Unsupervised Learning algorithm- K Means for the project.

Extensive comparisons were made between different models to identify the most effective approach for fraud detection.

The best model- Random Forest was further tranied using hyper parameter tuning and feature selection.

Figures and visualizations were used to illustrate the methodology, including flowcharts depicting the data processing pipeline and model training process.

Results and findings were documented in detail, allowing for reproducibility and enabling others to replicate the study's methodology and outcomes.

1. **Experimental Details**

The dataset was split into training and testing sets using a stratified approach to preserve the class distribution. Training data was taken to be 70% while testing data was taken to be 30%.

On calculating training accuracies of each model, Random Forest came out with the best training accuracy of 99.9%. Decision Tree even surpassed the Random Forest model. However, Random Forest was selected due to its high precision in classifying tasks, crucial for fraud detection. While Decision Tree Classifier may achieve 100% training accuracy, it's prone to overfitting due to its complexity. Random Forest Classifier reduces overfitting by combining predictions from multiple trees, leading to better performance on new data.

In order to compare the models with the Random Forest Classifier, models were trained on the training set. Models were evaluated using various performance metrics such as accuracy, precision, recall, and F1-score to assess their effectiveness in detecting fraudulent transactions.

**5.1. Random Forest**

The trained Random Forest model yielded an accuracy of 99.16% and an AUC score of 1.

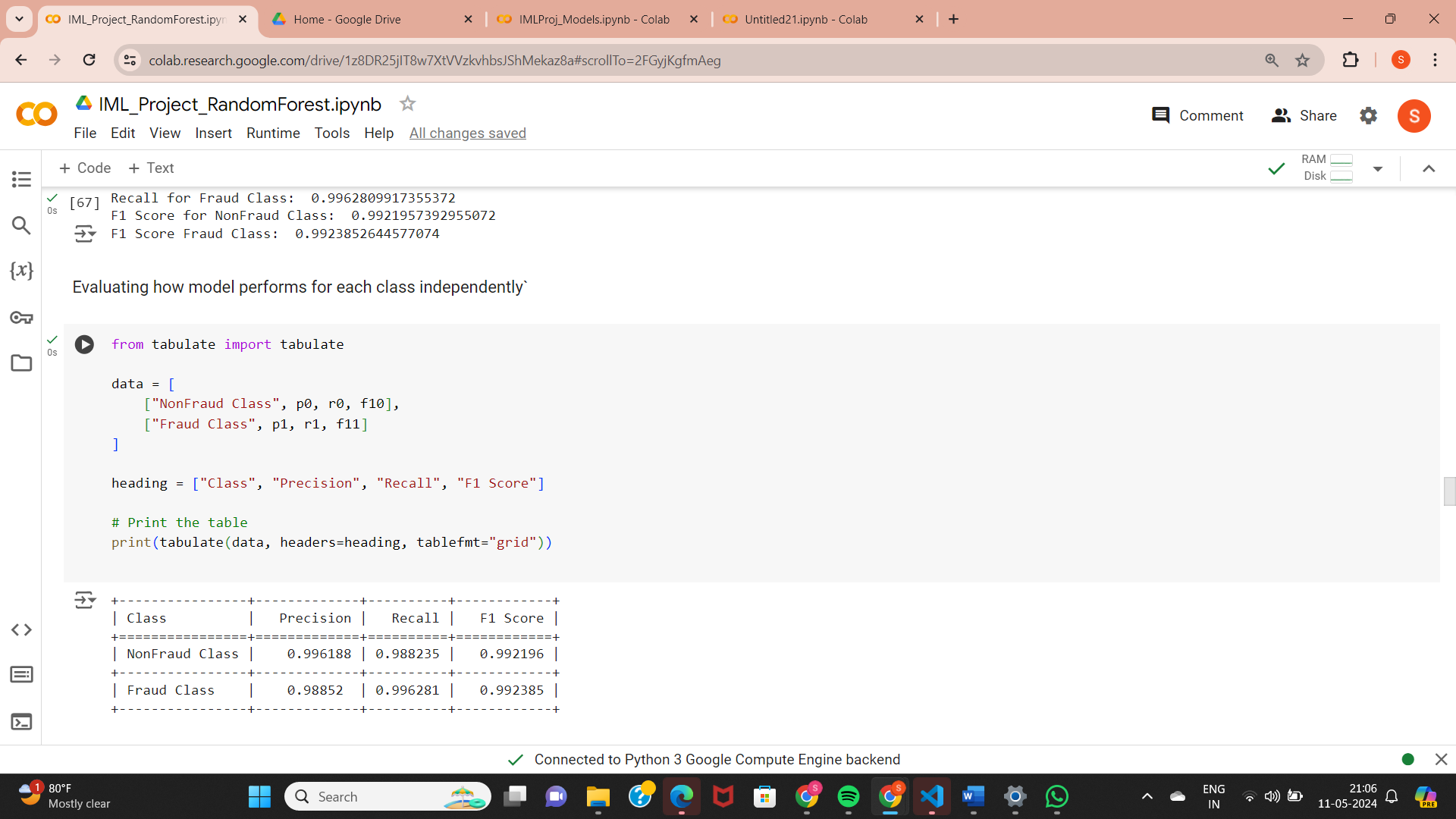


Fig1.: Precision, Recall and f1-score for Random Forest Model

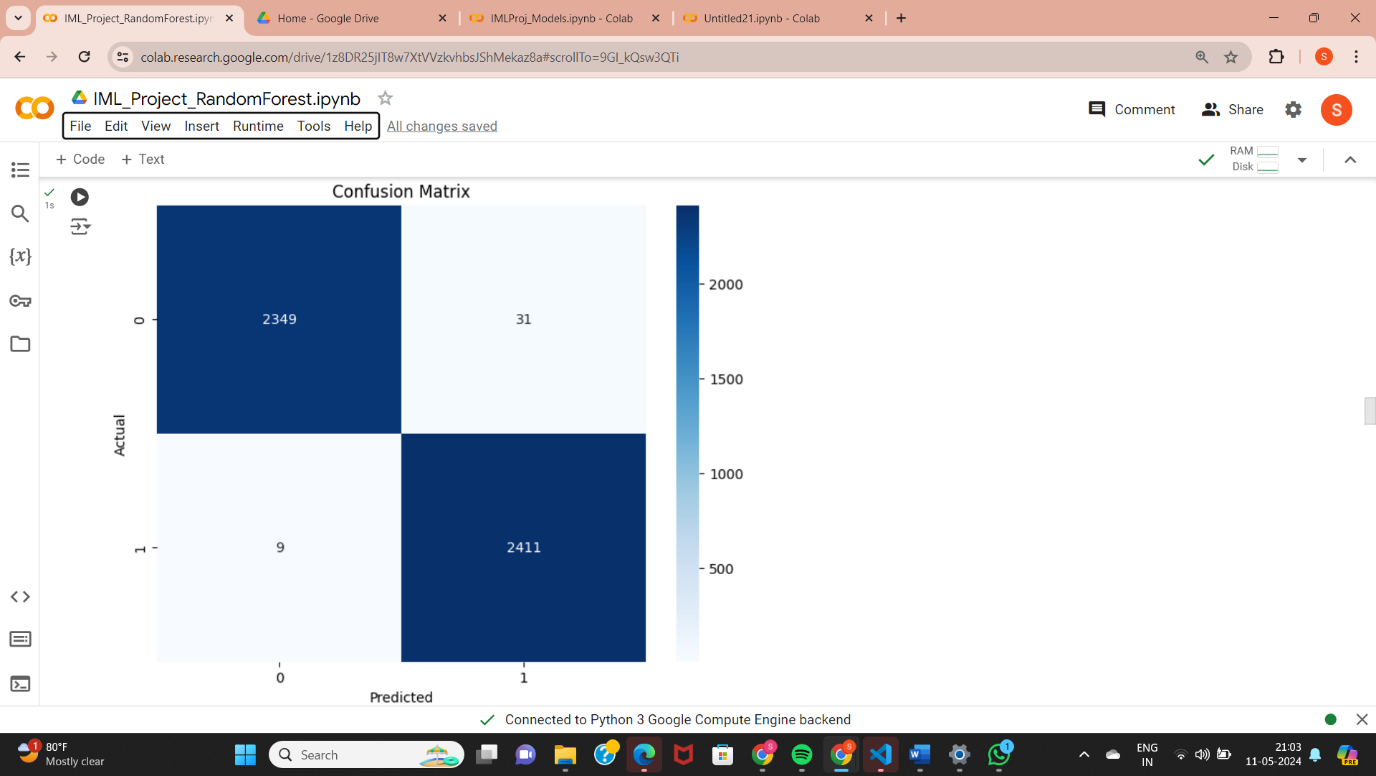


Fig2.: Confusion Matrix for Random Forest Model

In order to assess the model’s performance, an imbalanced dataset was taken next. Results suggested that the model was well trained and accuracy in detection remained unchanged.

**5.2. Decision Tree**

The trained Decision Tree model yielded an accuracy of 99.22% and an AUC score of 0.99.

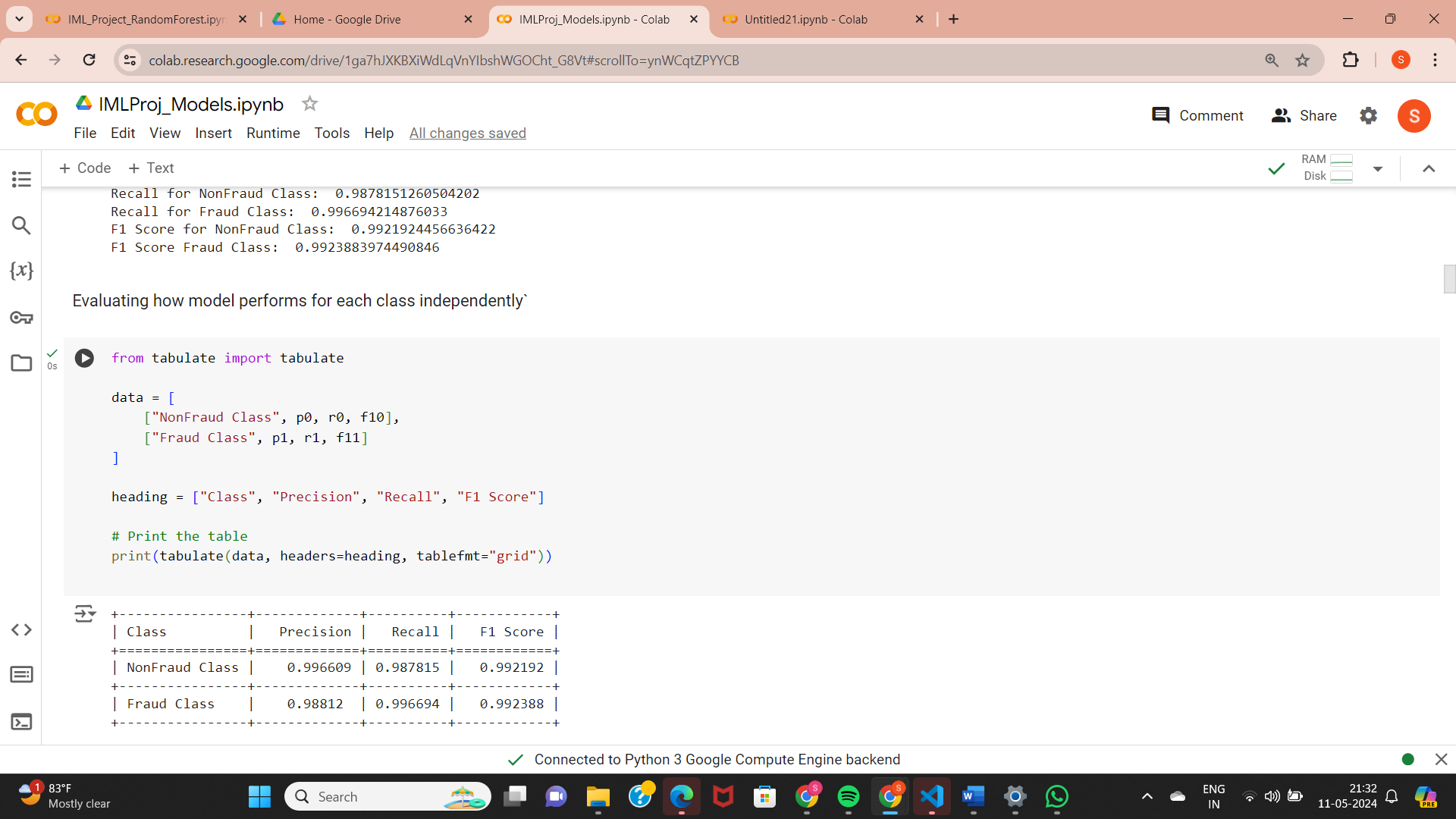


Fig3.: Precision, Recall and f1-score for Decision Tree Model

**5.3. Support Vector Machine**

The trained SVM model yielded an accuracy of 82.5% and an AUC score of 0.96.

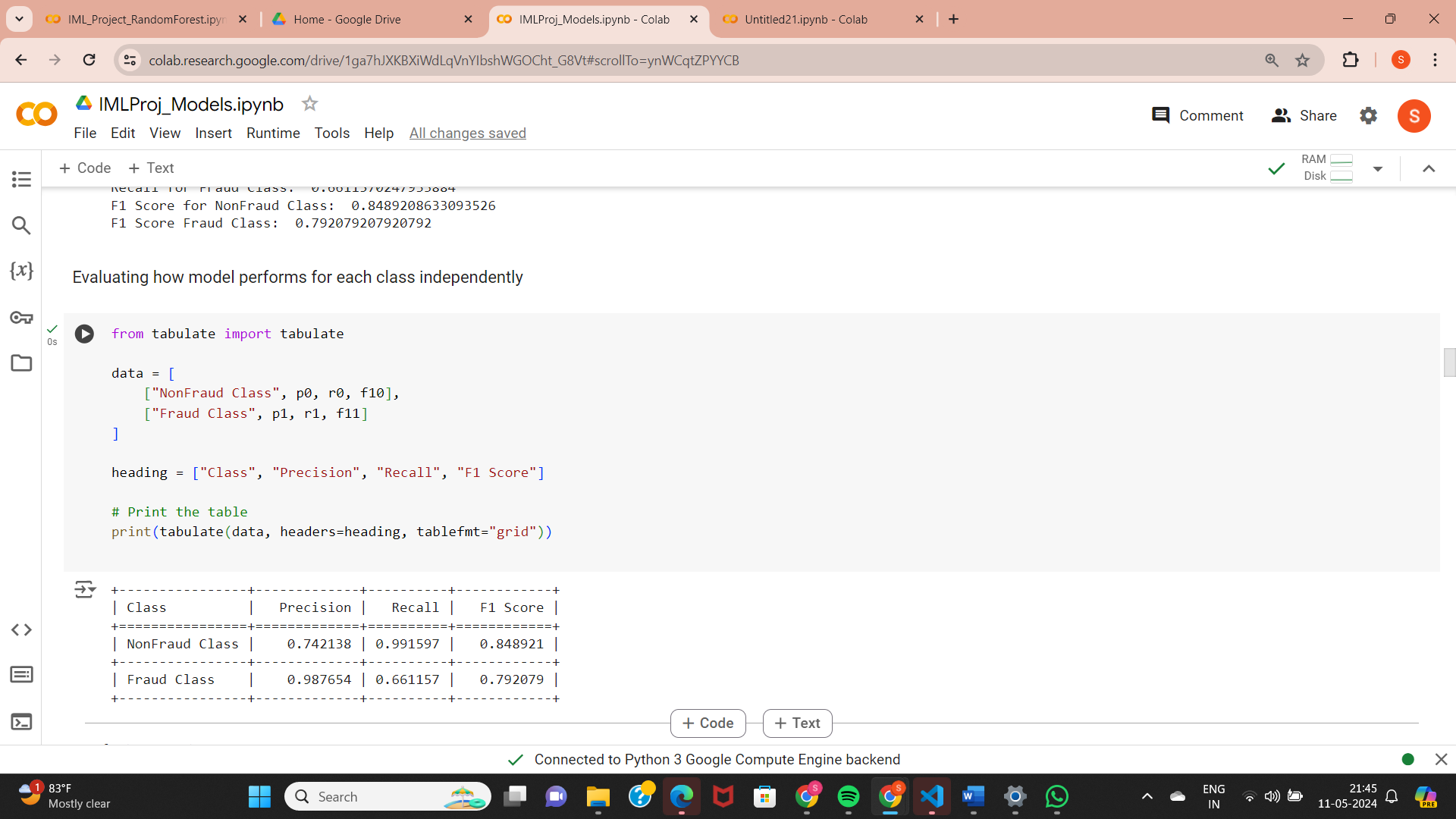


Fig4.: Precision, Recall and f1-score for SVM

**5.4. Multi-Layer Perceptron**

The trained MLP model yielded an accuracy of 94.29% and an AUC score of 0.98.

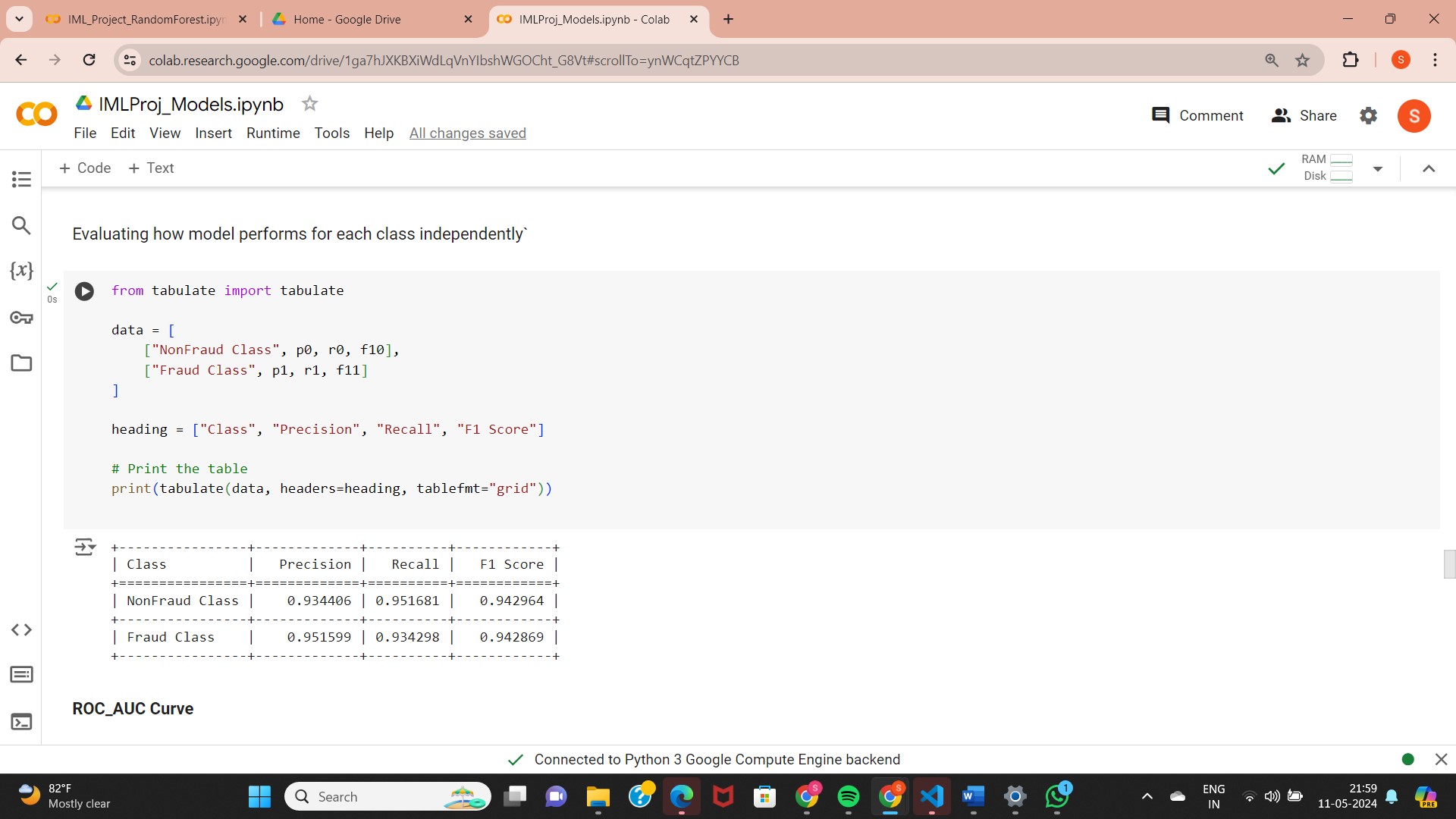


Fig5.: Precision, Recall and f1-score for MLP

**5.5. Comparison**

Extensive comparisons were made between different models to identify the most effective approach for fraud detection.

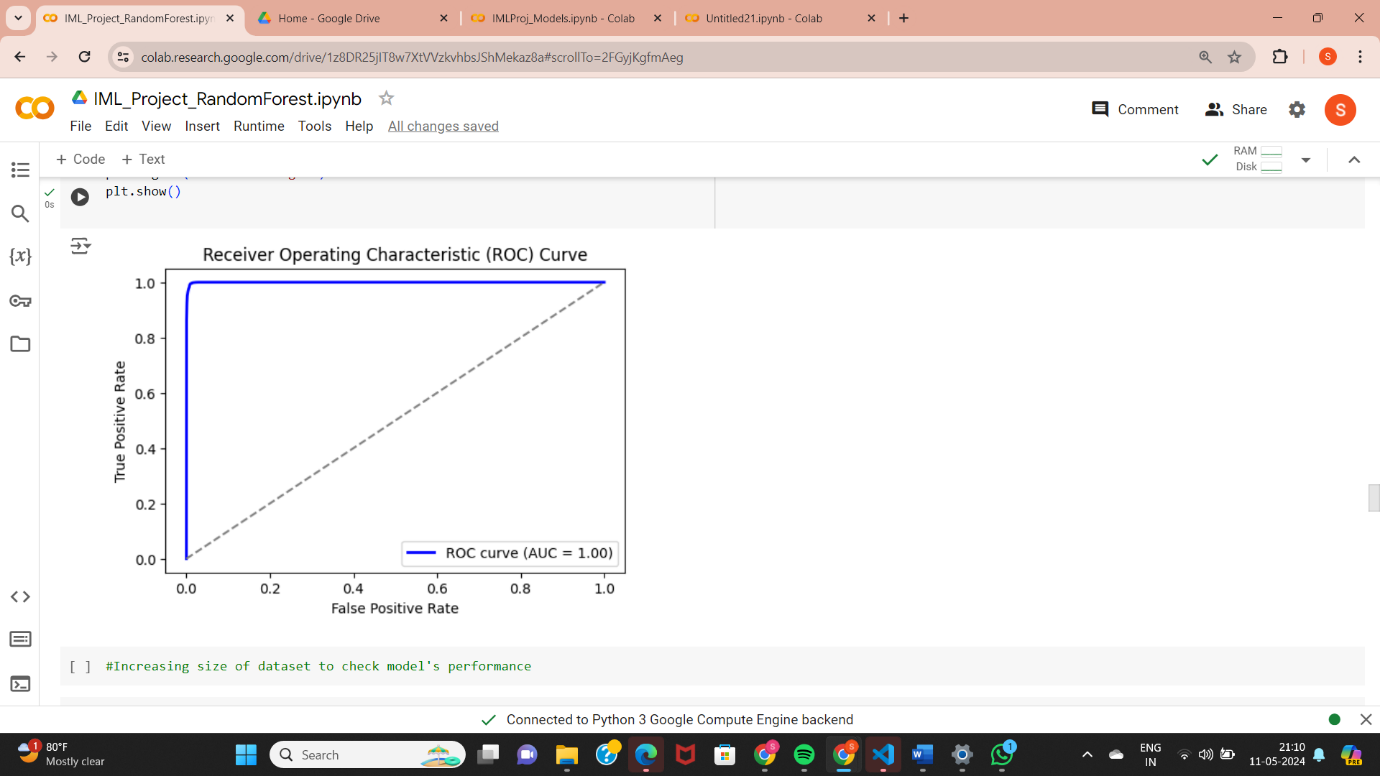
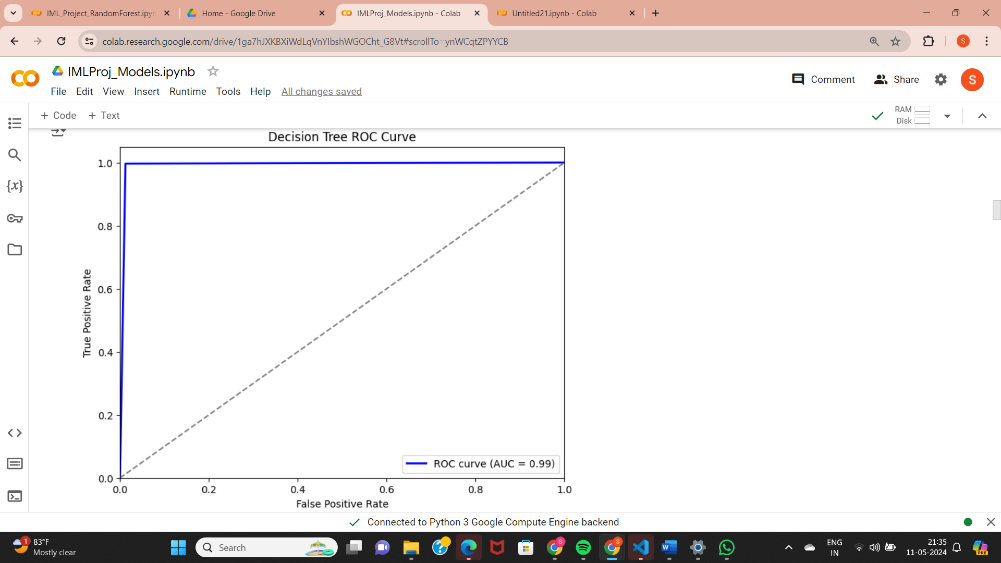
 ****

Fig6.: ROC\_AUC Curve for Random Forest Fig7.: ROC\_AUC Curve for Decision Tree

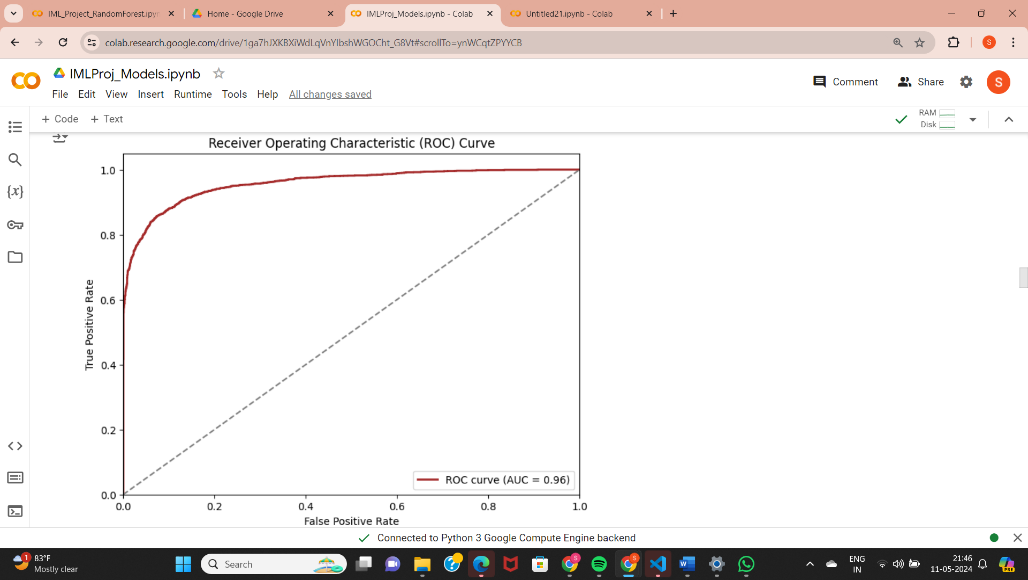
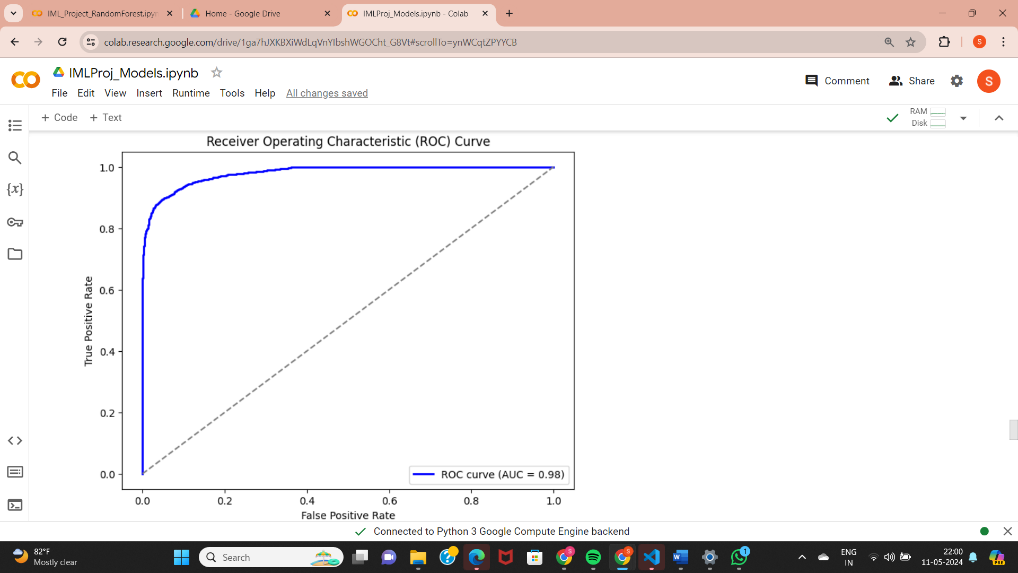
 

Fig8.: ROC\_AUC Curve for SVM Fig9.: ROC\_AUC Curve for MLP

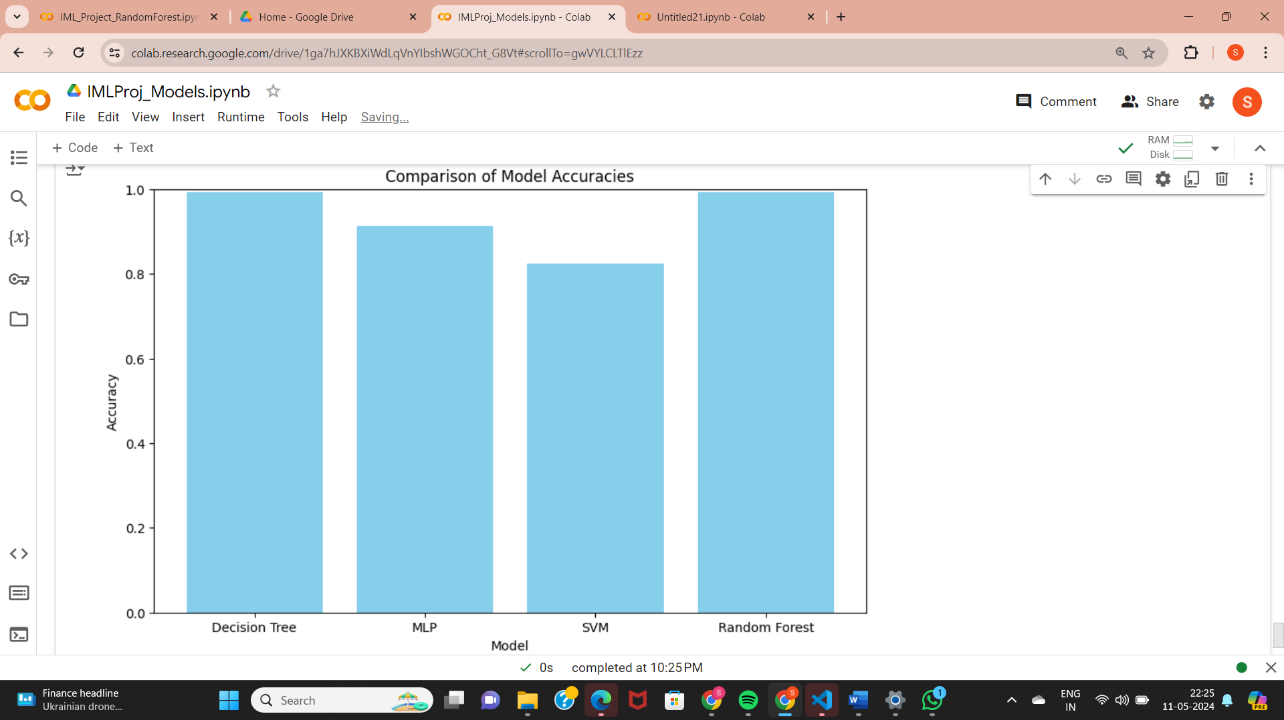


Fig10.: Comparison of Model Accuracies

**5.5. Exploring Unsupervised Learning using K-Means**

We employed K-Means to assess how an unsupervised learning model performs for the given dataset.

The trained K-Means model yielded an accuracy of 43.04%.

We applied the Elbow Method to determine optimal value for K (number of clusters).

Accuracy did improve on applying Elbow Method, Optimal value for K = 1. Initially the number of clusters chosen were 3. The accuracy of the model with K=1 is 49.58%.

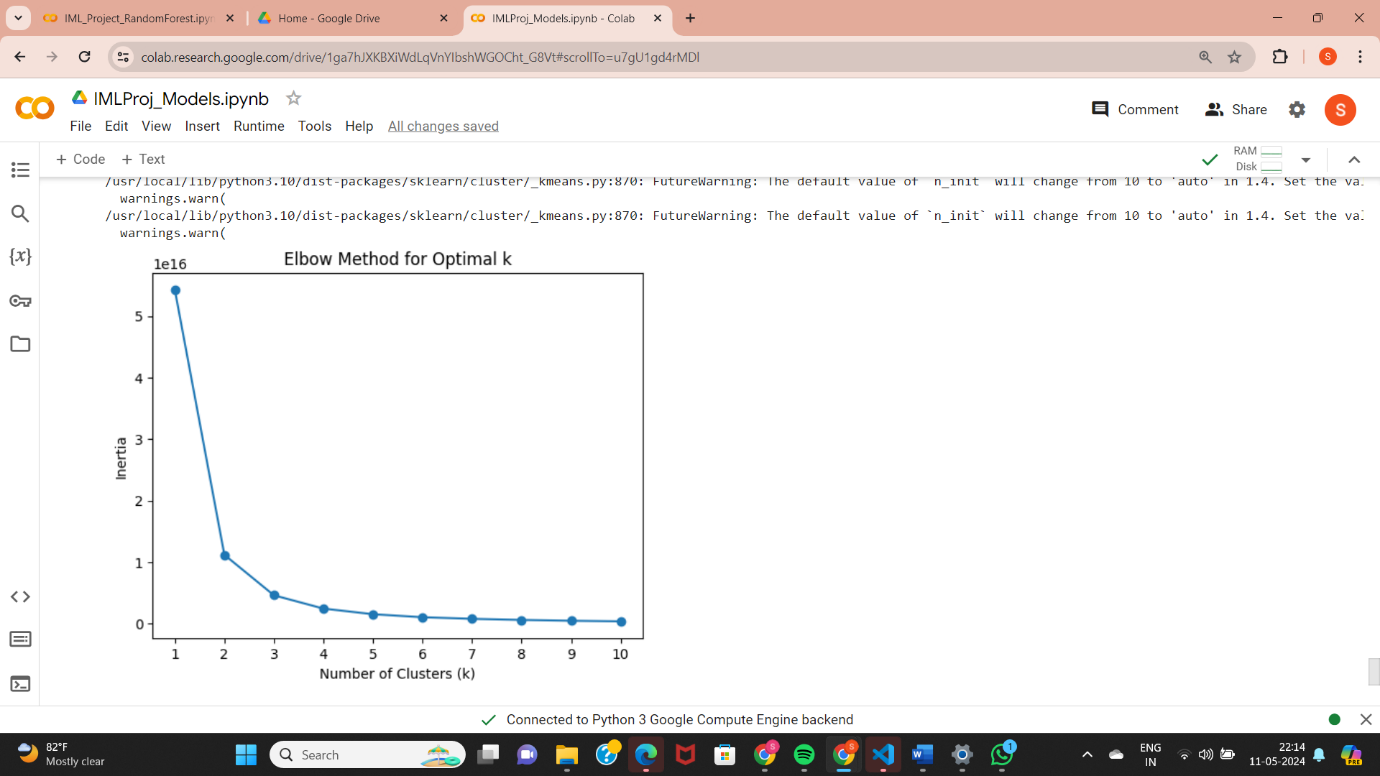


Fig11.: Elbow Method for optimal K

**5.6. Further Training of the Random Forest Model**

The best model- Random Forest was further trained using hyper parameter tuning and feature selection.

**5.6.1. Hyper parameter tuning using Randomized Search**

Hyperparameter tuning, using Randomized Search, was conducted to explore potential improvements or optimizations beyond the model's already excellent performance because it helps the model generalise better to unseen data.

The accuracy of the model indeed improved. 20 trees were used earlier, but suggested was 200 trees. Though accuracy does not change significantly, hyperparameter tuning makes the model more robust.

**5.6.2. Feature Selection**

By performing feature selection, we can potentially improve the performance and efficiency of the Random Forest model by focusing on the most informative features and reducing noise from less relevant ones. The selected features were ‘amount’, ‘oldbalanceOrg’, ‘newbalanceOrig’, for a threshold of 0.1.

**6. Results**

Random Forest proved superior in fraud detection, in comparison to other models, due to its precision and ability to handle complex data

Comprehensive evaluation metrics like accuracy, precision, recall, and F1-score ensured thorough model assessment.

Feature selection streamlined the model.

**7. Conclusion and Future Work**

In conclusion, this project addressed the crucial task of fraud detection in online payment systems using machine learning techniques. Through comprehensive exploration and evaluation of various models, including Decision Trees, Random Forest, and Support Vector Machines (SVM), we demonstrated the effectiveness of advanced analytical methods in accurately identifying fraudulent transactions. The random forest algorithm emerged as the most promising model, consistently outperforming others with high accuracy and precision. However, it's important to acknowledge the limitations of our study, including the reliance on a single dataset and the absence of real-time implementation.

Our future work includes exploring advanced feature engineering methods, including deep learning approaches, to extract richer insights from the data. To conduct fine-tuning of model hyperparameters to optimize fraud detection performance and enrich the dataset by integrating supplementary data sources like user behaviour analytics or transaction metadata is also our next agenda. Developing a real-time fraud detection system equipped with continuous monitoring capabilities for timely identification and prevention of fraudulent activities will be our ultimate future goal.

**Acknowledgements**

We would like to thank our professors Dr.Shabeer Basha and Prof.Ashwini Kumar Mathur for guiding us throughout the project with their valuable inputs.

**8.** **Code**

**https://github.com/sunkesulahimaja/IML\_Project**

# References

|  |  |
| --- | --- |
| [1] | S. G. a. Vaishnavi Nath Dornadula a, Credit Card Fraud Detection using Machine Learning Algorithms, vol. 165, Chennai: Procedia Computer Science, 2019, pp. pages 631-641. |
| [2] | A. A. a. A. P. A. S. Khatri, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection", 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 680-683. |
| [3] | C. B. S. P. S. M. a. N. K. A. Thennakoon, "Real-time Credit Card Fraud Detection Using Machine Learning", 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 488-493. |
| [4] | A. O. A. a. S. A. O. J. O. Awoyemi, Credit card fraud detection using machine learning techniques: A comparative analysis, IEEE,2017 International Conference on Computing Networking and Informatics (ICCNI), 2017. |