ML-Powered Fraud Detection

Pranav Bakale
Student ID - 801422140
ITCS-5154 Fall 2024
November 24, 2024

Primary Research Paper Details

Title: Fraud Detection in Banking Transactions using Machine Learning
Authors: Rathnakar Achary and Chetan J Shelke
Publication and year of publication: IEEE Conference, 2023

Abstract

The increasing pace of digital banking has significantly increased the number of online financial transactions. This increase in the number of frauds was witnessed. Detecting fraud is no easy task at all because of the dynamism and complexity of fraud patterns with the volume and velocity of transaction data. Most of the time, these problems cannot be handled by traditional rule-based methods. Hence, the present work constitutes a fraud detection system using ML: fraud detection system which reports various advanced techniques for effective detection of fraudulent transactions. The project is concerned with developing and evaluating machine learning models, such as kNN, Random Forest, and XGBoost, against the simulated transaction input BankSim dataset. To address the class imbalance, SMOTE allows the learning of fraud patterns well. Of these models, XGBoost performed best, achieving an accuracy of 99%, with balanced precision and recall. The key findings keep on emphasizing the role of pre-processing, data balance, and algorithm choice in ensuring increased efficiency of fraud detection methods. These results show how ML-based methods outperform traditional techniques in managing fraud risk proactively and the institution's trustworthiness in digital banking systems. The project also entails future directions like real-time fraud detection using stream processing technologies to improve the system's effectiveness and scalability.

Keywords: Fraud, Machine Learning, kNN, Random Forest, XGBoost, SMOTE

1. Introduction

The growth of digital banking is unsurpassable today, with online financial transactions becoming the backbone of the global economy. Increasing digital payments open doors to more opportunities for fraudulent activities, resulting in damages worth billions of dollars in losses to poles and reputational damage. Today, fraud has become a trillion-dollar industry. Research conducted by financial institutions has found that dedicated domain expert teams with data scientists in fraud detection are working separately. Fraud detection is conventionally known not to work with rule-based methods in identifying the complex dynamic patterns of fraudulent transactions. To these deficiencies, there is added pressure for adaptive data-driven fraud detection improvement. Developing an ML-based fraud detection system able to diagnose such issues is the task of this project. [3] Taking this into consideration, the current system aims at utilizing advanced algorithms such as Random forest and XGBoost with technique SMOTE-AN for imbalance classes to high accuracy fraud detection and low rate of false positions. The proposed approach will be not just efficient in detection, but will alter with the patterns of fraud over time, hence proving itself robust for modern financial institutions. This project would provide a secure digital banking environment to win back the confidence of users and mitigate financial risks due to online fraud.

1.1 Problem Statement

As the age of banking grew digital, so did the online financial transactions, which began being increased, drawing the attention of fraudsters. When complex dynamic patterns regarding fraud are analyzed, present traditional rule-based methods are inadequate to face the scenarios caused; hence, banks and their customers experience huge losses due to fraud. These models try to construct a machine learning model that efficiently detects fake transactions by reducing the rate of false positive detection and adapting patterns to real-time due to the dynamic nature of fraud patterns.

1.2 Motivation

This is an important step in making computer banking systems reliable. Digital banking, where every day trillions of online transactions are conducted, minor fraudulent activities can achieve million-dollar effects. Adaptive machine learning models can improve the security of financial systems, reduce losses, and thus enhance user confidence.

1.3 Open Questions in the Domain

- How can efficient real-time management through machine-learning models be achieved, even if it counts as data?
- Best methods for handling class imbalance of fraud detection data?
- How can a model be built to fulfill the requirement of adapting to newly evolving fraud patterns?

1.4 Approach Overview

Intended to tackle the problem of class imbalance, fraud detection, and further fine-tuning performance, this solution tends to train models using Random Forest, XGBoost, and SMOTE. All these models shall be tested with BankSim data set.

2. Background/ Related Work

Related Research Summary:

1. Credit Card Fraud Detection using Machine Learning Techniques

- Authors: John O. Awoyemi, Adebayo O. Adetunmbi, Samuel A. Oluwadare
- Year: 2018 (IEEE)
- Pros: This manuscript addressed several classifiers and highlighted using SMOTE to handle imbalanced datasets.
- Cons: There is no investigation of the advanced algorithms such as XGBoost.
- Relevance: It demonstrates the productivity of SMOTE in the work at hand.

2. Critical Analysis of Machine Learning-Based Approaches for Fraud Detection in Financial Transactions

- Authors: Thushara Amarasinghe, Achala Aponso, Naomi Krishnarajah
- Year: 2018 (ACM)
- Pros: Identify why ML-based approaches are rather better than earlier ones; investigate neural networks.
- Cons: Less practical implementation.
- Relevance: Paves way for advanced investigation using techniques like XGBoost.

Relation to Current Work:

The results of both articles underline the same aspect regarding the use of machine learning in fraud detection and give useful insights into the benefits of using advanced techniques and data balancing methods.

3. Method

Methodology:

1. Data Collection: I collected credit card transaction data from the BankSim dataset. The synthetic data has transactions from different customers over different time periods and different amounts.

2. Data Preprocessing: Data cleaning, data scaling, and different manipulating techniques like exploring data, dropping irrelevant features, and handling categorical values turned out to be effective.

3. Handling Class Imbalance: I went through SMOTE analysis to make up for the class imbalance by synthesizing samples that are expected to be fraudulent.

4. Machine Learning Models: I have applied K-Nearest Neighbors, Random Forest, and XGBoost models on the pre processed dataset hence aiming at the project's purpose.

5. Model Evaluation: Evaluation of these models comes as accuracy, precision, recall, and ROC-AUC Curve.

6. Results: XGBoost achieves maximum accuracy with 99% and excellent identification of fraudulent transactions.

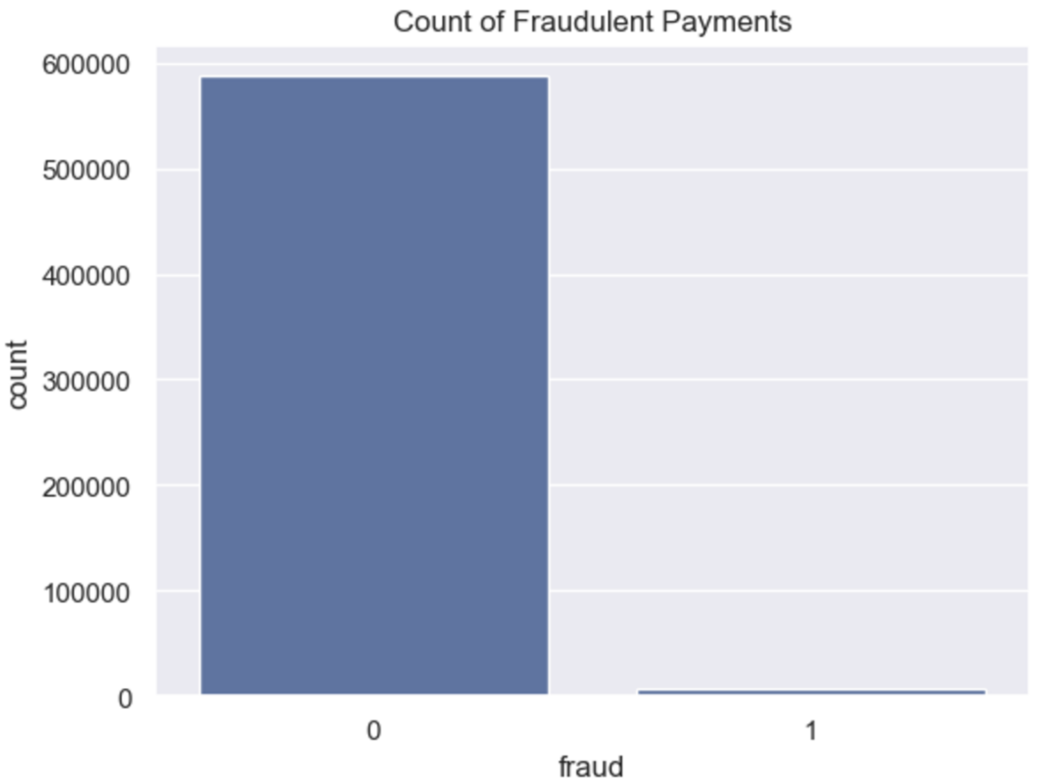


Fig 1. Countplot of Fraudulent Payments

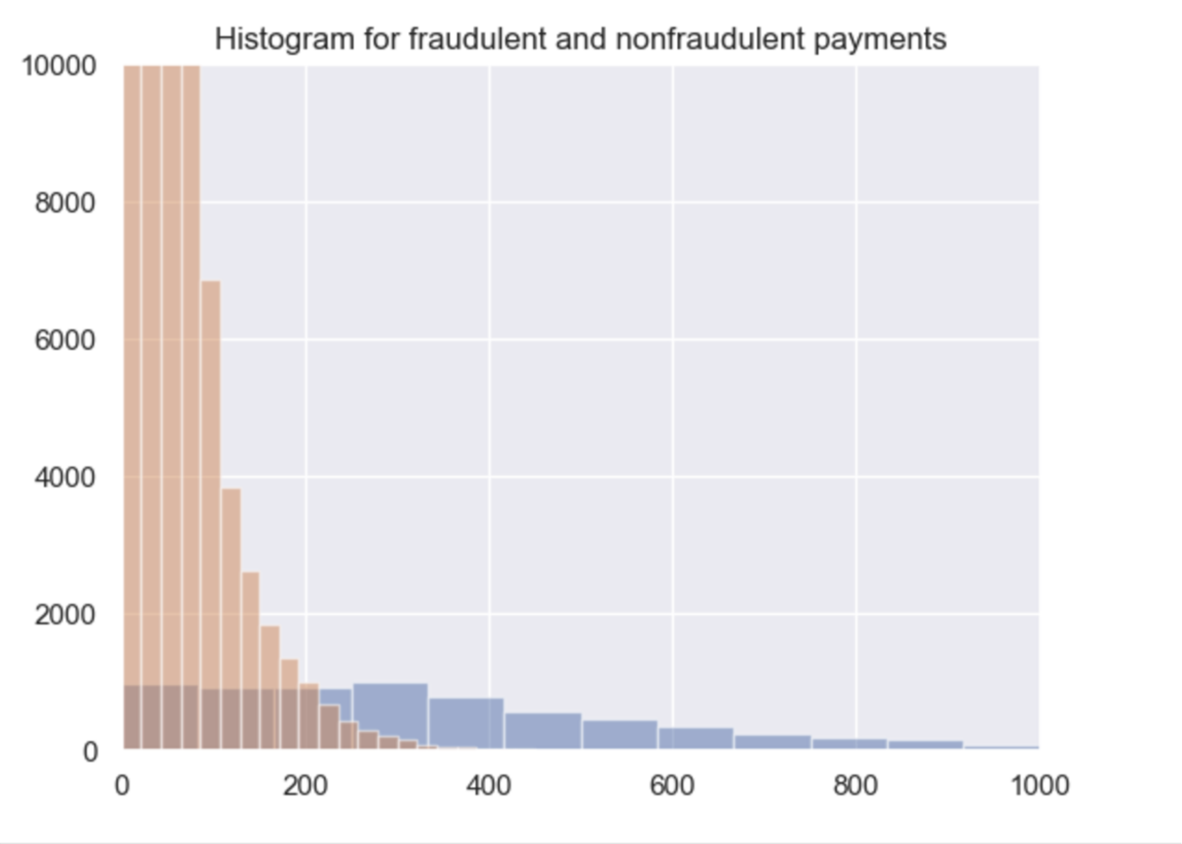
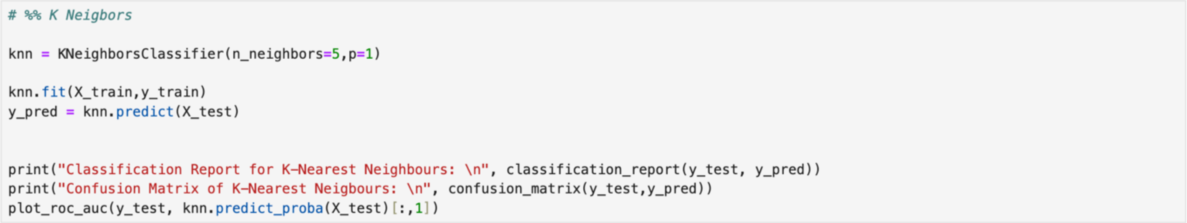


Fig 2. Histogram

Algorithms Implemented:

1) KNN:

K-nearest neighbor is an instance-based learning which performs its classification according to a similarity measure, such as using the functions of Euclidean, Mahanttan or Minkowski distance types. The first two distance measures work well with continuous variables while the third suits categorical variables. The Euclidean distance measure is used in this study for the kNN classifier. [1]

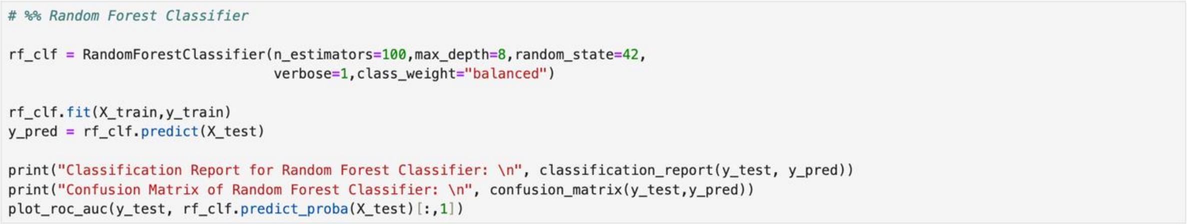


```
# %% K Neigbors

knn = KNeighborsClassifier(n_neighbors=5,p=1)

knn.fit(X_train,y_train)
y_pred = knn.predict(X_test)


print("Classification Report for K-Nearest Neighbours: \n", classification_report(y_test, y_pred))
print("Confusion Matrix of K-Nearest Neigbours: \n", confusion_matrix(y_test,y_pred))
plot_roc_auc(y_test, knn.predict_proba(X_test)[:,1])
```

Fig 3. KNN Classifier

2) Random Forest:

Random forest – RF is an ensemble classifier used in this both for classification and regression task. It involves the concept of bagging method, which is a collection of many weak learners. In RF they are considered as decision trees. [2]

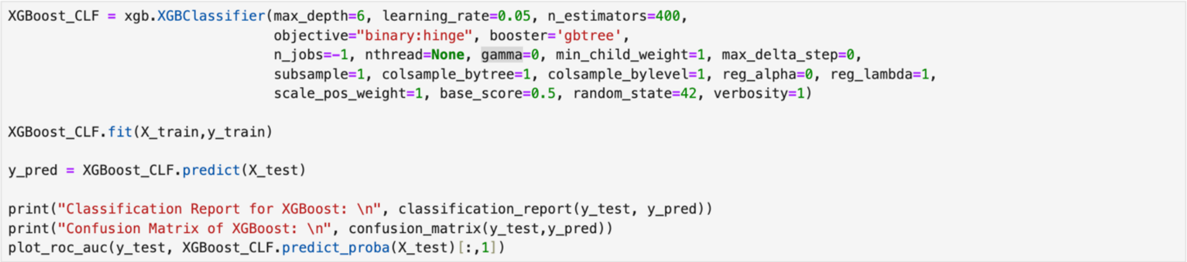


```
# %% Random Forest Classifier

rf_clf = RandomForestClassifier(n_estimators=100,max_depth=8,random_state=42,
                                verbose=1,class_weight="balanced")

rf_clf.fit(X_train,y_train)
y_pred = rf_clf.predict(X_test)

print("Classification Report for Random Forest Classifier: \n", classification_report(y_test, y_pred))
print("Confusion Matrix of Random Forest Classifier: \n", confusion_matrix(y_test,y_pred))
plot_roc_auc(y_test, rf_clf.predict_proba(X_test)[:,1])
```

Fig 4. Random Forest Classifier

3) XGBoost:

XGBoost is a gradient- boosting algorithm used in the research, which combines a weak learner with a strong learner. In which the multiple iterations involved to process the weak learner and predict the value of the class labels and, then calculate the loss. [2]



```
XGBoost_CLF = xgb.XGBClassifier(max_depth=6, learning_rate=0.05, n_estimators=400,
                                objective="binary:hinge", booster='gbtree',
                                n_jobs=-1, nthread=None, gamma=0, min_child_weight=1, max_delta_step=0,
                                subsample=1, colsample_bytree=1, colsample_bylevel=1, reg_alpha=0, reg_lambda=1,
                                scale_pos_weight=1, base_score=0.5, random_state=42, verbosity=1)

XGBoost_CLF.fit(X_train,y_train)

y_pred = XGBoost_CLF.predict(X_test)

print("Classification Report for XGBoost: \n", classification_report(y_test, y_pred))
print("Confusion Matrix of XGBoost: \n", confusion_matrix(y_test,y_pred))
plot_roc_auc(y_test, XGBoost_CLF.predict_proba(X_test)[:,1])
```

Fig 5. XGBoost Classifier

Architecture Diagram:

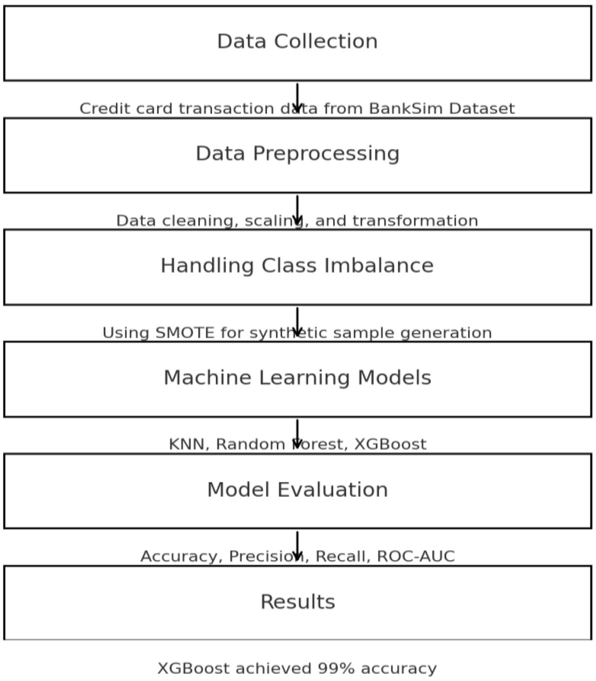

Data Collection
Credit card transaction data from BankSim Dataset
Data Preprocessing
Data cleaning, scaling, and transformation
Handling Class Imbalance
Using SMOTE for synthetic sample generation
Machine Learning Models
KNN, Random Forest, XGBoost
Model Evaluation
Accuracy, Precision, Recall, ROC-AUC
Results
XGBoost achieved 99% accuracy

Fig 6. Architecture Diagram

4. Experiments

Reproduction of Related Work:

Results align with existing literature; SMOTE effectively balances datasets and improves detection.

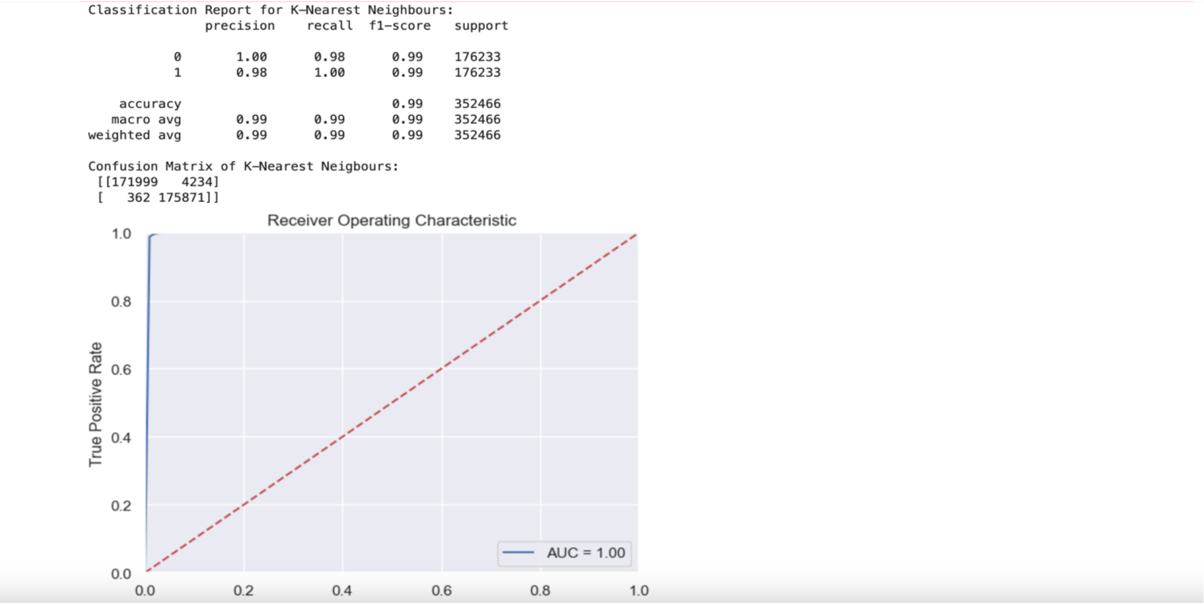
Experimental Setup:

- Dataset: BankSim (594,643 transactions; 7,200 frauds).
- Findings: The best is XGBoost with 99% accuracy over others.
- Evaluation Metrics: Accuracy, Precision, Recall, ROC-AUC

Outcomes:

1) kNN classifier:

KNN performs reasonably well in terms of accuracy but fails miserably in recall, especially in cases of fraud detection. It worked well in spotting normal transactions but often missed fraudulent ones, which caused some false negatives. This is likely because kNN doesn't always handle large or complex datasets well, particularly when there's an imbalance in the data. Even when balancing techniques like SMOTE were used, it couldn't fully overcome this limitation.

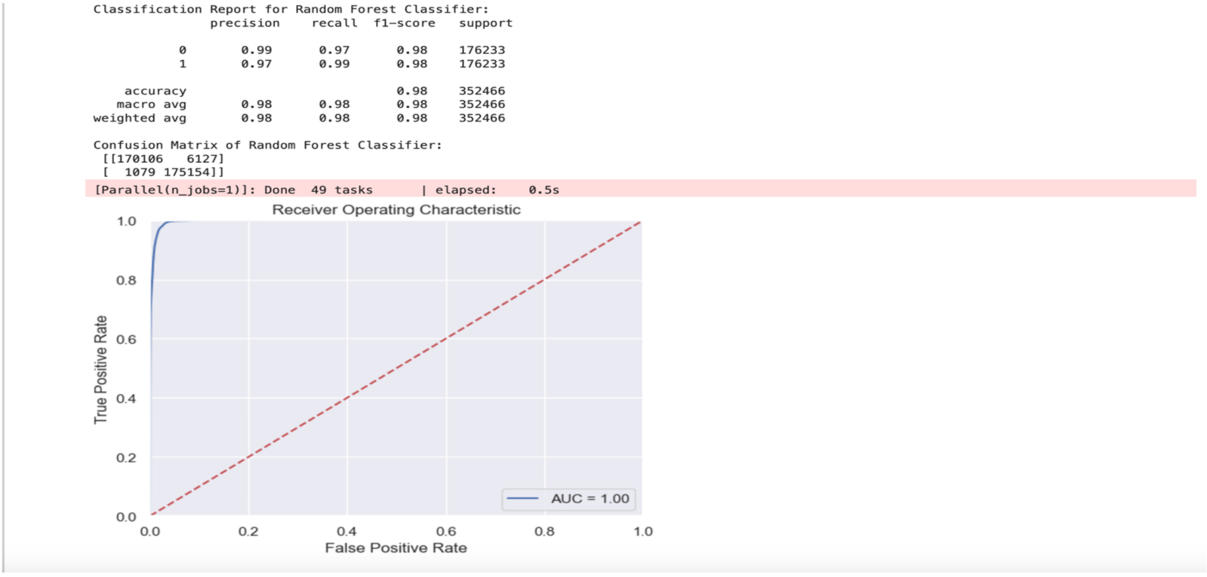


```
Classification Report for K-Nearest Neighbours:
              precision    recall  f1-score   support

           0       1.00      0.98      0.99    176233
           1       0.98      1.00      0.99    176233

    accuracy                           0.99    352466
   macro avg       0.99      0.99      0.99    352466
weighted avg       0.99      0.99      0.99    352466

Confusion Matrix of K-Nearest Neigbours:
 [[171999   4234]
 [   362 175871]]
```

Fig 7. KNN Classifier Result

2) Random Forests:

The Random Forest method yields approximately 98% accuracy and provides a better trade-off between precision and recall than kNN; however, there are still very few false positives and negatives. Random Forest works fine for most complex datasets but is sometimes unable to detect finer details of fraud once such fraud patterns change in real time. This makes it less effective in dynamic situations where quick adaptability is needed.

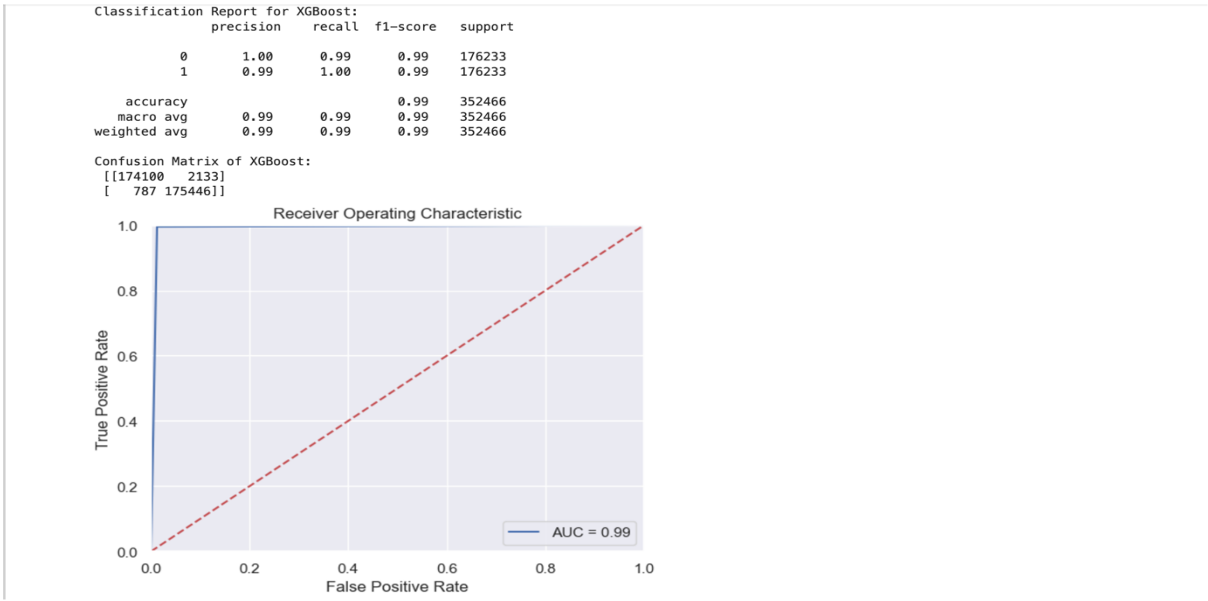


```
Classification Report for Random Forest Classifier:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98    176233
           1       0.97      0.99      0.98    176233

    accuracy                           0.98    352466
   macro avg       0.98      0.98      0.98    352466
weighted avg       0.98      0.98      0.98    352466

Confusion Matrix of Random Forest Classifier:
 [[170106   6127]
 [  1079 175154]]
[Parallel(n_jobs=1)]: Done  49 tasks      | elapsed:    0.5s
```

Fig 8. Random Forest Classifier Result

3) XGBoost Classifier:

On the other hand, XGBoost stood out with a remarkable 99% accuracy. It was especially good at identifying both fraudulent and legitimate transactions, striking a strong balance between precision and recall. This success is likely because XGBoost can process complex patterns in data with great efficiency. Compared to other models, it made fewer errors, which is why it's a solid option for real-world scenarios where accuracy is crucial.



```
Classification Report for XGBoost:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99    176233
           1       0.99      1.00      0.99    176233

    accuracy                           0.99    352466
   macro avg       0.99      0.99      0.99    352466
weighted avg       0.99      0.99      0.99    352466

Confusion Matrix of XGBoost:
 [[174100   2133]
 [   787 175446]]
```

Fig 9. XGBoost Classifier Result

Observations:

- XGBoost achieved superior accuracy of 99% due to its ability to handle complex patterns.
- Balanced metrics (Precision and Recall) demonstrate the model’s reliability.

Discussion:

- Results validate the effectiveness of ML-based fraud detection.
- The approach reduces false positives and adapts to emerging fraud trends.

5. Conclusions

Fraud detection relies heavily on balancing the dataset because fraud cases are rare in real-world data. An imbalanced dataset would lead the models toward taking biased decisions and failing to recognize fraudulent transactions. For better accuracy and clear results, data cleaning and feature generation would be key steps in the process. XGBoost is one of the machine learning models yielding good accuracy and class balance in fraud detection. Such a model works like a charm for banks, as the model will help them catch abnormal activities as soon as possible, keeping risks low and efficiently keeping track of the transactions.

Future Work:

- Real-time fraud detection systems and similar technologies could be modeled using tools like Apache Kafka or Spark Streaming.
- It will be possible to optimize the latency at which transactions can be monitored continuously.

6. References

[1] John O. Awoyemi, Adebayo O. Adetunmbi, Credit card fraud detection using Machine Learning Techniques.

[2] Rathnakar Achary, Chetan J Shelke. Fraud Detection in Banking Transactions Using Machine Learning. In 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE).

[3] Thushara Amarasinghe, Achala Aponso, Naomi Krishnarajah. Critical Analysis of Machine Learning Based Approaches for Fraud Detection in Financial Transactions.

[4] R. Rambola, P. Varshney and P. Vishwakarma, “Data Mining Techniques for Fraud Detection in Banking Sector,” 2018 4th International Conference on Computing Communication and Automation (ICCCA).

Sharing Agreement

Do you agree to share your work as an example for next semester? - Yes

Do you want to hide your name/team if you agree? - Yes

You are fine to say "no" if you don't want.