Final Project: Optimizing Fraud Detection with Machine Learning Models

Submitted to:

Dr. Erol Ozkan
Applied Statistics and Data Science
College of Science
University of Texas at Arlington

Team: 6
Eraam Khan
Jani Shariff Shaik
Lakshman Kumar Reddy Peddireddy
Venkata Nithin Reddy Yerraguntla

Executive Summary

This project delves into fraud detection using a highly imbalanced dataset from Capital One, focusing on identifying and analyzing fraud patterns. We used data cleaning techniques and advanced machine learning algorithms including XG - Boost, Gradient Boost, and Random Forest to pinpoint fraudulent activities. To tackle the dataset's imbalance, we utilized resampling techniques such as SMOTE for oversampling and RandomUnderSampler for under sampling, achieving the best performance with XG - Boost in the under sampled dataset. Our analysis revealed that the majority of fraudulent transactions occur in the merchant categories of online retail, online gifts, and rideshare, with specific merchants like Lyft, ebay.com, and Fresh Flowers having the highest incidence of fraud. This information suggests a higher risk of fraud associated with these categories and merchants, prompting us to recommend that customers exercise increased caution when transacting with these merchants.

Problem Statement

This analysis focuses on utilizing advanced machine learning models to detect fraud in Capital One's transaction data, a crucial task for ensuring financial security. Given the highly imbalanced nature of the dataset, where fraud cases are rare, accurately identifying these instances while minimizing false positives is challenging. This analysis specifically describes a classification problem, aiming to differentiate between fraudulent and non-fraudulent transactions. By improving our ability to detect fraud, we aim to protect both the institution and its customers from potential financial losses, enhancing the overall safety of transactions and strengthening Capital One's risk assessment processes.

Data Preprocessing

Introducing the Capital One’s Fraud Transaction Dataset

The dataset used for our analysis comprises of transactional data from Capital One, spanning the entire year of 2016. This data, which was provided during a real-time internship, consists of 786,363 instances across 29 distinct features. This comprehensive dataset forms the basis for our in-depth analysis, aimed at identifying and understanding fraud transactions.

Looking into Data Types

We investigated the data types of our 29 features. We found our features to have datatypes ranging from integer, float, object, and Boolean.

Integer: accountNumber, customerID, creditLimit, cardCVV, enteredCVV, and cardLast4Digits.

Float: availableMoney, transactionAmount, and currentBalance.

Object: transactionDateTime, merchantName, acqCountry, merchantCountryCode, posEntryMode, posConditionCode, merchantCategoryCode, currentExpDate, accountOpenDate, dateofLastAddressChange, transactionType, echoBuffer, merchantCity, merchantState, merchantZip, posOnPremises, and recurringAuthInd.

Boolean: cardPresent, expirationDateKeyInMatch, and isFraud.

Missing Values

We found no missing values for any of our attributes.

Empty Values

Several attributes in our dataset were found to contain a significant number of empty or blank values. Specifically, the variables echoBuffer, merchantCity, merchantState, merchantZip, posOnPremises, and recurringAuthInd each had 786,363 empty values, indicating that no data was recorded for these fields throughout the dataset. Consequently, these variables were excluded from further analysis as they provided no substantive information. Additionally, the variables acqCountry, merchantCountryCode, posEntryMode, posConditionCode, and transactionType also exhibited a considerable number of missing entries and were similarly removed from the dataset.

```
accountNumber                  0
customerId                     0
creditLimit                    0
availableMoney                 0
transactionDateTime            0
transactionAmount              0
merchantName                   0
acqCountry                  4562
merchantCountryCode          724
posEntryMode                4054
posConditionCode             409
merchantCategoryCode           0
currentExpDate                 0
accountOpenDate                0
dateOfLastAddressChange        0
cardCVV                        0
enteredCVV                     0
cardLast4Digits                0
transactionType              698
echoBuffer                786363
currentBalance                 0
merchantCity              786363
merchantState             786363
merchantZip               786363
cardPresent                    0
posOnPremises             786363
recurringAuthInd          786363
expirationDateKeyInMatch       0
isFraud                        0
dtype: int64
```

Dropping Columns

We decided to remove the accountNumber column from our dataset as it duplicated the function of the customerId column, with both serving as unique identifiers for individual records. Since each column contains the same number of values, retaining both would be redundant and unnecessarily increase the dimensionality of our dataset.

Understanding our Dataset

In our dataset, the acqCountry column contains four unique values, representing the countries included in our analysis: the United States, Mexico, Canada, and Puerto Rico. Notably, most of the data is concentrated in the United States, with only a minimal spread across the other three countries.

Additionally, the merchantName column within our dataset encompasses a variety of unique entries, indicating a wide range of merchants involved in the transactions. Due to the extensive number of unique merchant names, a detailed list is not presented here. For a comprehensive overview of the merchant names, please refer to the relevant

section of the accompanying code. This will provide a clear understanding of the diverse merchant entities captured in our data.

We had four unique values for the column, posEntryMode. They were as follows:

- 09: Contact Less transaction
- 02: Chip Read with Pin Entry
- 05: Chip Read without Pin Entry
- 90: Manual Entry without card present
- 80: Fallback transaction

We had several merchant categories for our dataset, highlighting the type of merchant's groups we would be working with.

The merchant categories are as follows:

- Rideshare
- Entertainment
- Mobile Apps
- Fast Food
- Food Delivery
- Auto
- Online Retail
- Gym
- Health
- Personal Care
- Food
- Fuel
- Online Subscriptions
- Online Gifts
- Hotels
- Airline
- Furniture
- Subscriptions
- Cable/Phone

The comparison between the cardCVV and enteredCVV columns reveals a discrepancy, as indicated by the result `False`. This mismatch suggests that there are instances where the CVV numbers entered during transactions do not align with the card’s recorded security CVV. This finding could point to fraudulent activities where incorrect CVV details are being used.

Upon analyzing the isFraud column, we observed that 764702 transactions were marked as 'False', indicating non-fraudulent activities, while 11966 transactions were flagged as 'True', denoting fraudulent transactions. This significant disparity highlights the highly imbalanced nature of our dataset, with a predominant share of legitimate transactions over fraudulent ones. Addressing this imbalance is crucial and forms a key part of our subsequent analytical efforts to ensure the accuracy and reliability of our fraud detection models.

Feature Engineering

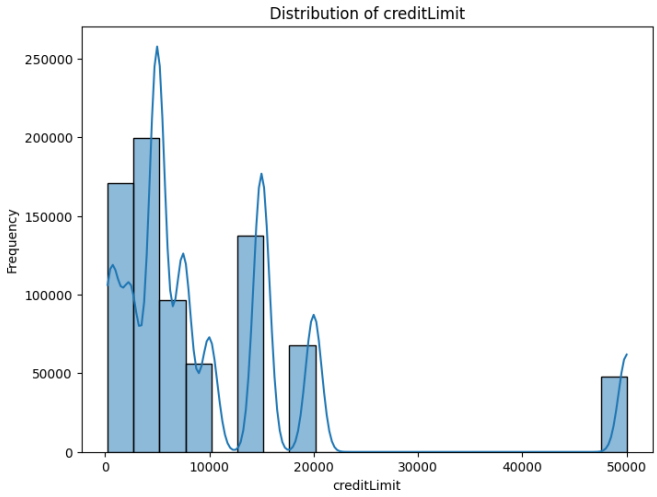
We made several modifications to the dataset to enhance the predictive capabilities of our models. Initially, we converted the transactionDateTime column to a datetime data type. We further extracted the day of the transaction and stored it in a new column named day.

Subsequently, we streamlined the dataset by removing several columns: `transactionDateTime`, `currentExpDate`, `accountOpenDate`, and `dateOfLastAddressChange`. These columns were deemed less relevant for our modeling needs.

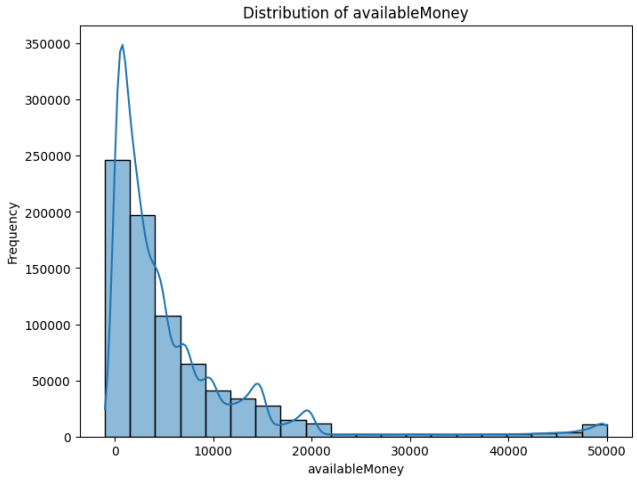
Additionally, we introduced a new feature named transactionAmount_availableMoney_ratio, calculated as the ratio of transactionAmount to availableMoney. This ratio provides insight into the extent of funds utilized for each transaction relative to the available credit, which can be a critical indicator of potential fraudulent behavior.

Explanatory Data Analysis (EDA)

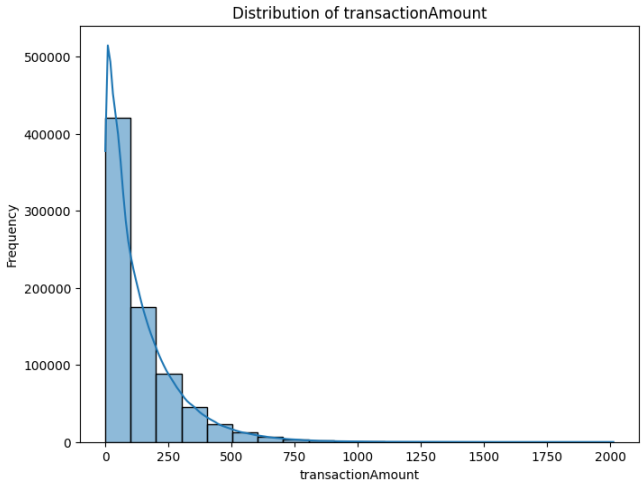
Visualizing the Skewness of our Dataset



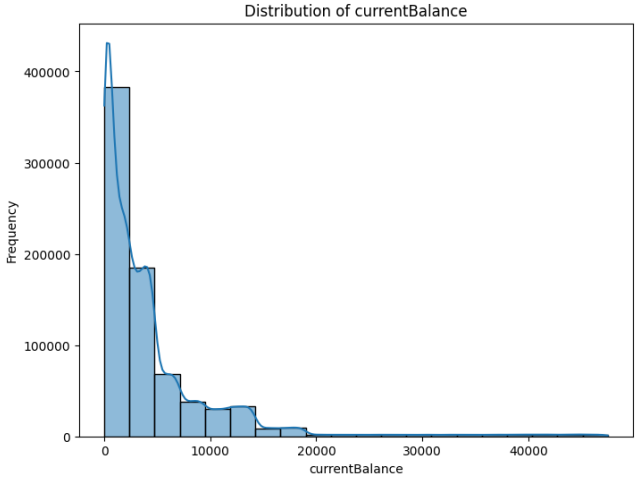
The histogram of the creditLimit distribution is right skewed, indicating that most cardholders possess lower credit limits, with fewer individuals holding higher limits. The multi-modal nature of the histogram, characterized by several peaks, suggests that credit limits are typically grouped around common thresholds, reflecting different customer segments based on financial criteria. The long tail towards higher limits signifies their rarity among the cardholders.



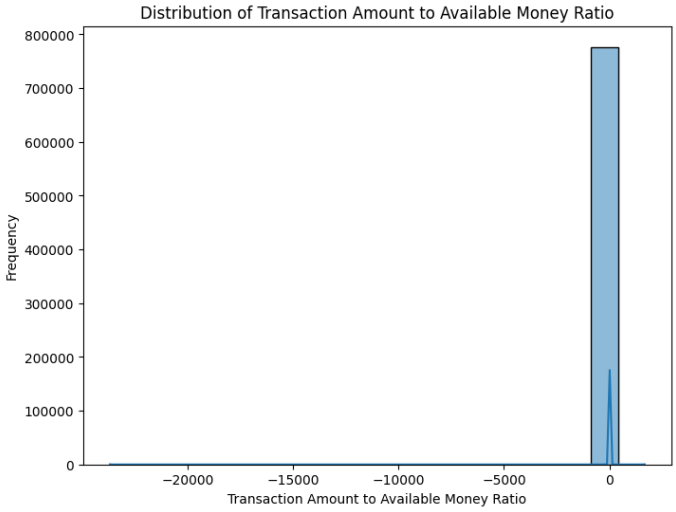
The histogram depicting the distribution of availableMoney demonstrates a right-skewed pattern, where most data points are concentrated at lower values, with frequencies diminishing as the amount of available money increases. This indicates that most cardholders have lower balances in their accounts, with fewer individuals maintaining higher balances. The distribution also shows a gradual tail extending towards the higher end of the available money spectrum, which underscores the lesser prevalence of high balances among the cardholders.



The histogram of transactionAmount shows a pronounced right-skewed distribution, indicating that most transactions involve tiny amounts, with a rapid decrease in frequency as the transaction amounts increase. Most transactions are clustered near the lower end of the spectrum, with significantly fewer transactions approaching the upper limit of $2000. This skewness suggests that higher transaction amounts are much less common, reflecting typical consumer spending behavior where smaller purchases predominate.

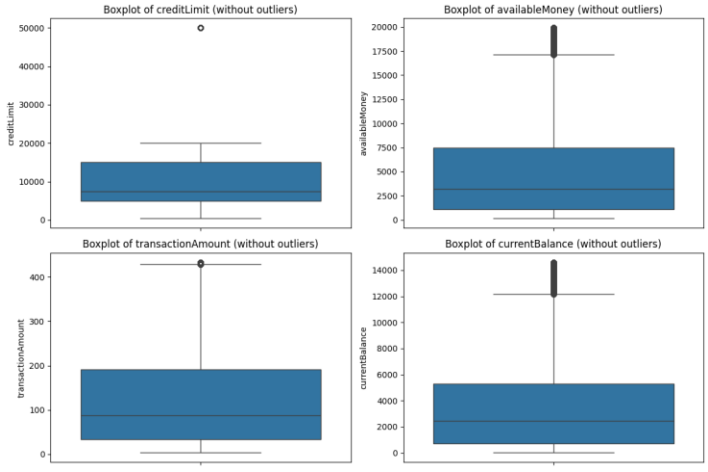


The histogram displaying the distribution of currentBalance demonstrates a pronounced right-skewed pattern. The bulk of the data clusters near the lower end of the balance range, with a steep decline in frequency as the balance amount increases. This skewness indicates that most account holders maintain low current balances, with fewer individuals possessing higher balances in their accounts. The distribution also shows a long tail extending towards the higher end, suggesting that while high balances are less common, they do exist among a small segment of the population.



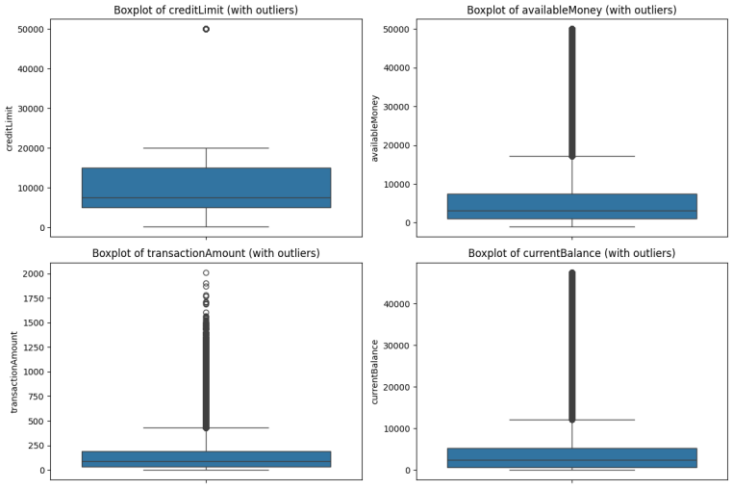
The histogram shows the distribution of the ratio between transaction amount and available money, primarily clustered near zero. This indicates that most transactions use a small fraction of the available funds. The presence of negative values suggests data errors or scenarios where transactions exceed the available money, potentially leading to overdrafts. Further analysis is needed to clarify these negative ratios.

Handling Outliers in our Dataset



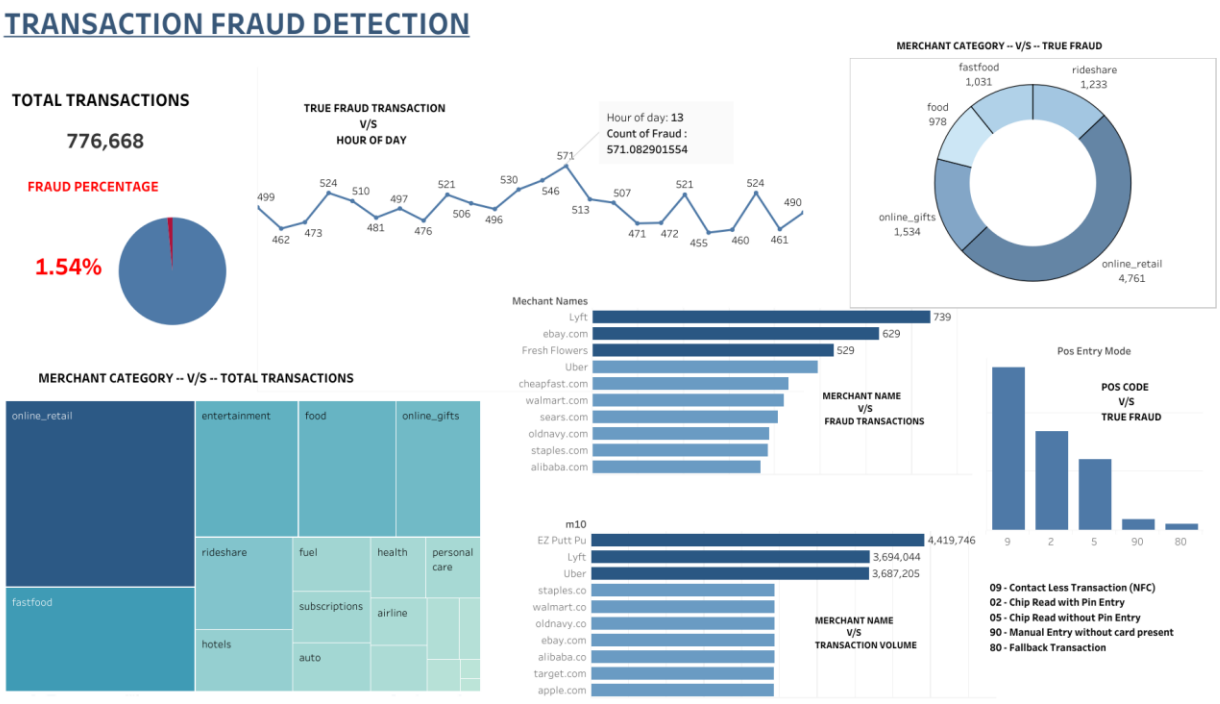
We focused on handling outliers by utilizing boxplots to visualize the distribution of variables: creditLimit, availableMoney, transactionAmount, and currentBalance. These boxplots revealed the presence of significant outliers in each variable, as depicted in the with outliers' plots, where the outliers are evident beyond the whiskers of the boxplots.

To address this issue, we applied a winsorizing technique, which involves adjusting the extreme values to specific percentiles to reduce the impact of outliers. We winsorized the variables by setting the lower and upper 5% of the data to the 5th and 95th percentiles, respectively.



Post-winsorization, new box plots were generated to confirm the reduction of outliers, visible in the without outliers' plots. These revised plots show a cleaner distribution with fewer extreme values, allowing for more accurate and reliable data analysis.

Data Visualization – Tableau



For More Details Click Here

Observations

Referencing the Tableau dashboard for our analysis, we can highlight some inferences based on the visualizations.

- The dataset exhibited significant imbalance with fraudulent transactions constituting only 1.54% of the total 776,668 transactions.
- A thorough temporal analysis revealed no consistent trends in fraudulent transactions on a yearly, monthly, weekly, or daily basis. However, a slight increase in fraud was observed during the 13th hour of the day (1 PM).
- The distribution of transaction amounts varied significantly across different merchant categories. Online retail, fast food, and entertainment sectors accounted for most of the transaction volumes, suggesting these areas are most used by cardholders.
- Fraudulent activities were most prevalent in the merchant categories of online retail, online gifts, and rideshare services. This pattern highlights higher risks associated with these sectors.
- Specific merchants, namely Uber, ebay.com, and Fresh Flowers, recorded the highest number of fraudulent transactions. This indicates that rideshare services and online marketplaces are particularly susceptible to fraud.
- Merchants like Ez Putt Pu, Uber, and Lyft featured prominently in terms of transaction volumes, especially in the rideshare category. This suggests a high customer usage.
- There are various POS entry codes which help determine how each transaction was processed. Among these, the codes for Contactless transactions (09), Chip Read with Pin Entry (02), and Chip Read without Pin Entry (05) are identified as the most susceptible to fraud.
    - These codes reflect higher risk; Contactless transactions and Chip Read without Pin Entry lack the security of PIN verification, and transactions that fall back to magnetic stripe usage (Fallback Transaction) pose increased security risks due to the ease of cloning or compromising magnetic strip data.
- Although online retail and online gifts sectors report the highest fraud transactions, their overall fraud rates are low due to their substantial total transaction volumes.
- Other POS entry codes, being more frequently used, exhibit a higher likelihood of fraud due to their broader application across various transaction scenarios.
- Online retail sectors implement stringent fraud detection measures such as two-factor authentication, which effectively reduce the incidence of successful fraud attempts compared to other POS entry codes.

Preparing our Dataset for Modelling

Converting Categorical Variables to Numerical Variables

We have utilized label encoding to transform categorical data into numerical formats suitable for machine learning models. Specifically, we have applied label encoding to several key categorical variables in our dataset: acqCountry, merchantCountryCode, posEntryMode, posConditionCode, merchantCategoryCode, and transactionType. This transformation assigns a unique integer to each category within these columns, enabling our machine learning algorithms to process and learn from these variables effectively. T

Defining the X and Y Variables

We structured the dataset by separating features and the target variable. The variable `X` is formed by excluding the columns isFraud, merchantName, and customerId. This is essential because isFraud is our target for predictions and including it in the feature set would lead to data leakage, while merchantName and customerId are excluded to prevent model bias as they are identifiers that do not contribute to predictive performance. The target variable `Y` is defined as the isFraud column, which consists of binary values indicating fraudulent (1) and non-fraudulent (0) transactions.

Splitting our Dataset

We split the dataset into training and testing sets, allocating 80% of the data for training and 20% for testing.

Addressing Class Imbalance

We used both oversampling and under sampling techniques to achieve a balanced class distribution.

Oversampling with SMOTE

SMOTE was utilized to equalize the class distribution in our training dataset. Initially, the class distribution was heavily skewed with 611,806 non-fraudulent transactions and only 9,528 fraudulent transactions. By applying SMOTE, which creates synthetic samples for the minority class, we made the fraudulent class to match the non-fraudulent class count, resulting in a balanced dataset of 611,806 for both classes.

Under Sampling with RandomUnderSampler

Also, we did under sampling using the RandomUnderSampler to decrease the size of the majority class to equal the minority class. Before resampling, the distribution mirrored the initial imbalance. After applying RandomUnderSampler, both classes were equal to 9,528 instances each, drastically reducing the size of the non-fraudulent class to match the fraudulent class.

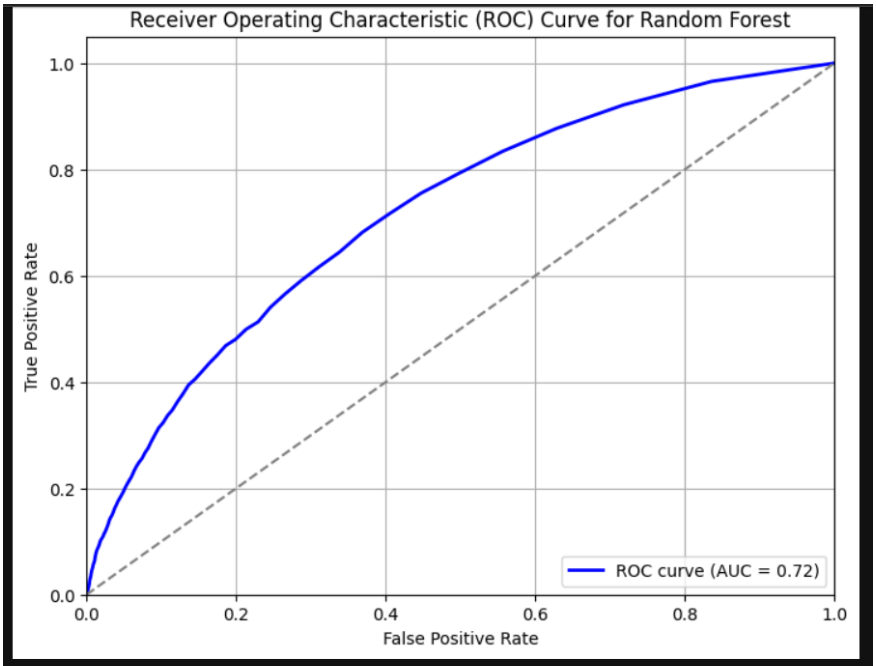
Model Selection

We chose Random Forest, Gradient Boost, and XG – Boost as machine learning models to achieve our goal of accurately identifying fraudulent transactions. These models are known to handle classification problems, especially in imbalanced datasets like ours.

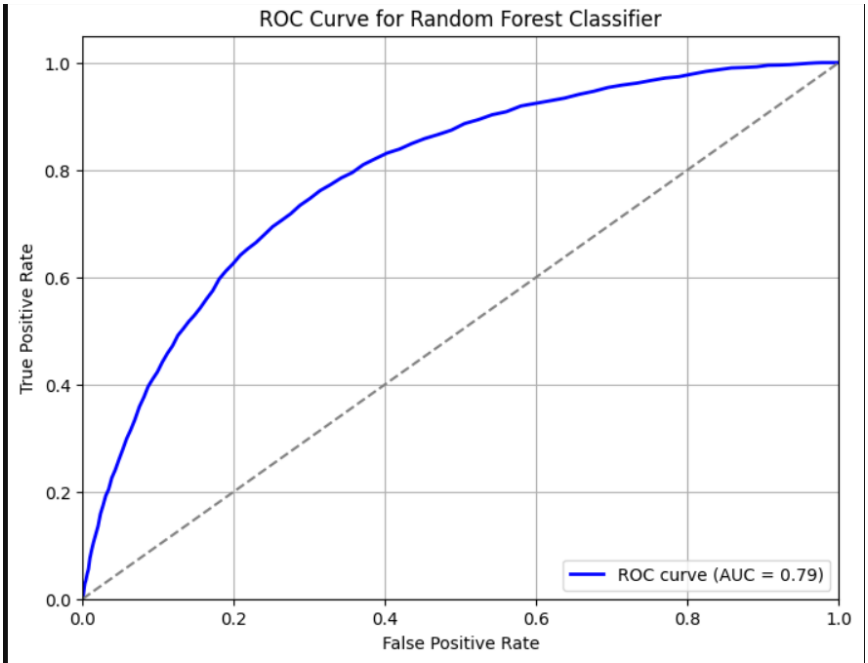
Random Forest

Random Forest utilizes an ensemble of deep decision trees, each trained on different data samples, to enhance prediction accuracy and control overfitting. This approach effectively reduces variance and bias, essential in fraud detection. Additionally, its ability to handle imbalanced data through diverse tree construction leads to more robust and reliable predictions across varied data scenarios.

Random-Forest results with Over-sampling (Using SMOTE):



Random-Forest results after under-sampling (Using Random Selection – Random Sampling):

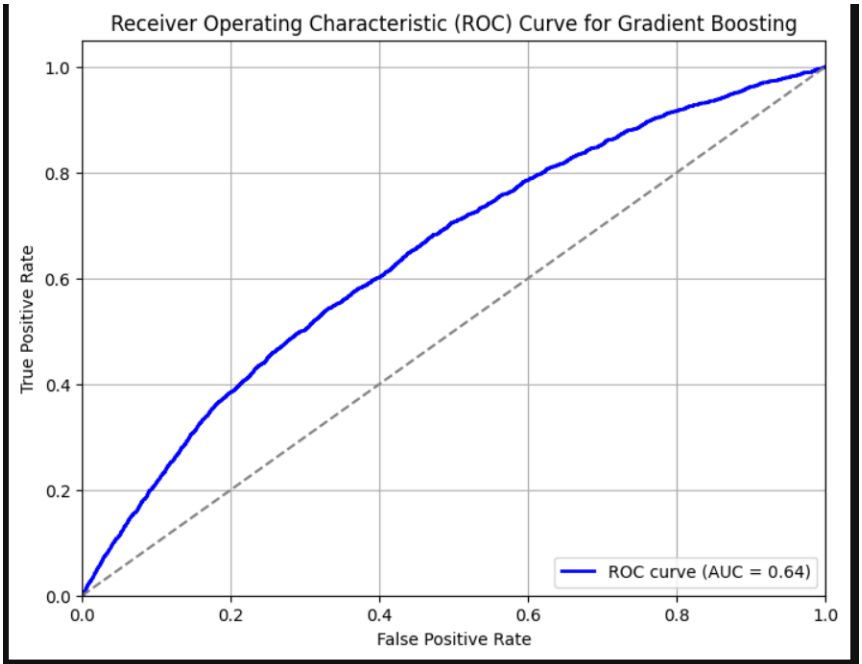



```
Accuracy: 0.7363745219977597
Precision: 0.041021118367736094
Recall: 0.7059064807219032
F1-score: 0.07753649306181294
ROC-AUC Score: 0.721383415120265
```

Gradient Boost

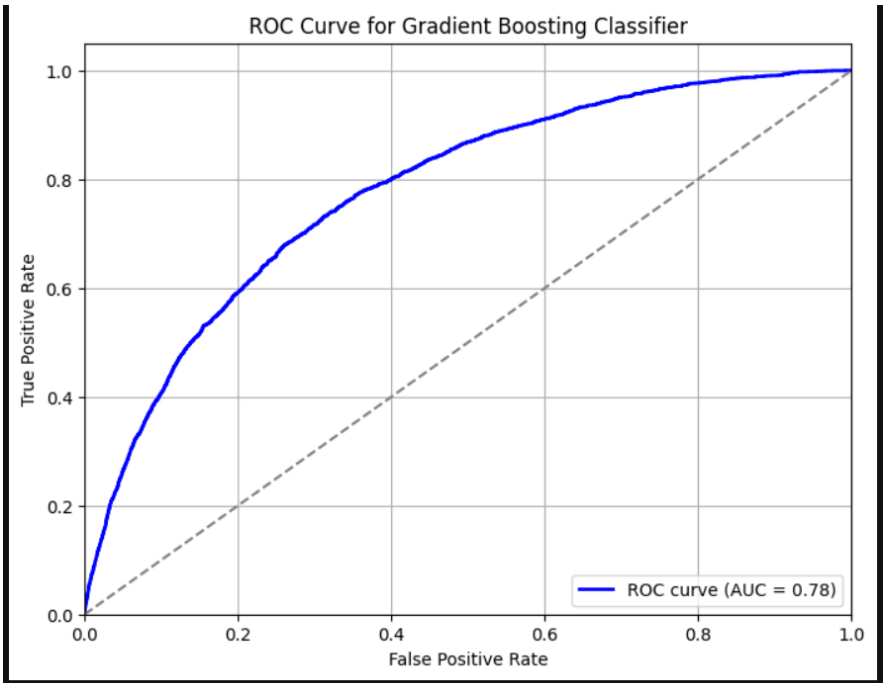
Gradient Boosting uses sequential tree building to correct previous errors, effectively reducing bias and variance, crucial for precise fraud detection. Known for its high performance on structured data, it offers flexibility in optimizing various loss functions and extensive hyperparameter tuning, allowing for adjustments to enhance model accuracy and adaptability.

Gradient-Boost results with Over-sampling (Using SMOTE):




```
Accuracy: 0.7141256904476805
Precision: 0.026790546635395995
Recall: 0.48728465955701394
F1-score: 0.050788764909580605
ROC-AUC Score: 0.6025137194813115
```

Gradient-Boost results after under-sampling (Using Random Selection – Random Sampling):



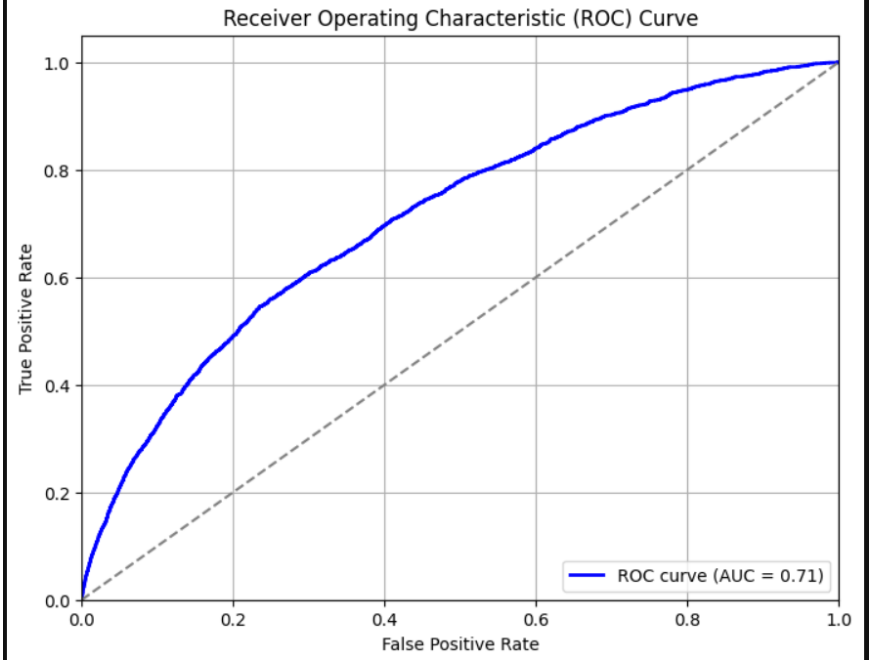

```
Accuracy: 0.7244067622027374
Precision: 0.03854330978670873
Recall: 0.6915504511894995
F1-score: 0.0730170416405015
ROC-AUC Score: 0.7082405615093583
```

XG – Boost

XG - Boost, an advanced form of gradient boosted decision trees, is known for speed and efficiency. It uses regularization to prevent overfitting, crucial in fraud detection scenarios, and is scalable, making it applicable for large datasets. Also, XG - Boost is good in handling imbalanced datasets by integrating with techniques like SMOTE to effectively manage and detect fraud within minority classes.

XG-Boost results with Over-sampling (Using SMOTE):

```
Accuracy: 0.8307646748297217
Precision: 0.041310870066928226
Recall: 0.44052502050861364
F1-score: 0.07553805035870025
ROC-AUC Score: 0.7118750659404723
```

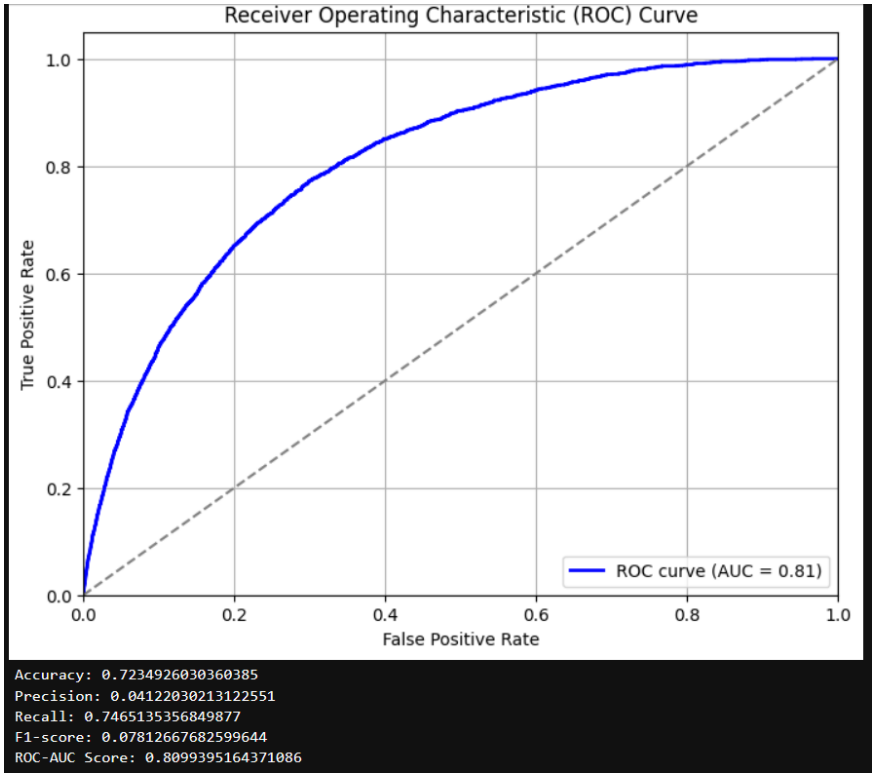



```
Training Accuracy: 0.8332619814785607
Test Accuracy: 0.8307646748297217

Confusion Matrix for Training Data:
[[512829  98977]
 [  4623   4905]]

Confusion Matrix for Testing Data:
[[127972  24924]
 [  1364   1074]]
```

XG-Boost results after under-sampling (Using Random Selection – Random Sampling):



Model Evaluation and Performance Metrics

Over-Sampling Metrics

Metric	XG-BOOST	RANDOM FOREST	GRADIENT BOOST
Accuracy	0.8381	0.9524	0.7081
Precision	0.0475	0.0732	0.0287
Recall	0.4893	0.1743	0.5361
F1-score	0.0866	0.1031	0.0545
ROC-AUC Score	0.7447	0.5695	0.6234

Over-Sampling Metrics

- XG-Boost has the highest accuracy among the three models, followed by Random Forest and Gradient Boost.
- For Precision, Random Forest performs slightly better than XG-Boost and Gradient Boost.

- Recall is highest for XG-Boost, indicating it identifies more relevant instances correctly.
- F1-score, which balances precision and recall, is highest for Random Forest.
- ROC-AUC score, which measures the model's ability to discriminate between positive and negative classes, is highest for XG-Boost.

Under-Sampling Metrics

Metric	XG-BOOST	RANDOM FOREST	GRADIENT BOOST
Accuracy	0.7287	0.7434	0.7337
Precision	0.0420	0.0424	0.0399
Recall	0.7477	0.7116	0.6927
F1-score	0.0796	0.0801	0.0755
ROC-AUC Score	0.8109	0.7278	0.7136

Under-Sampling Metrics

- XG-Boost still maintains the highest accuracy among the models, followed closely by Random Forest and Gradient Boost.
- Precision, Recall, and F1-score are generally low across all models due to the imbalanced nature of the dataset after under-sampling.
- ROC-AUC score is highest for Random Forest, indicating its better ability to distinguish between classes.

From the above metrics we can see XG-Boost with under-sampling is the best model so let us do hyperparameter tuning to see the best parameter results.

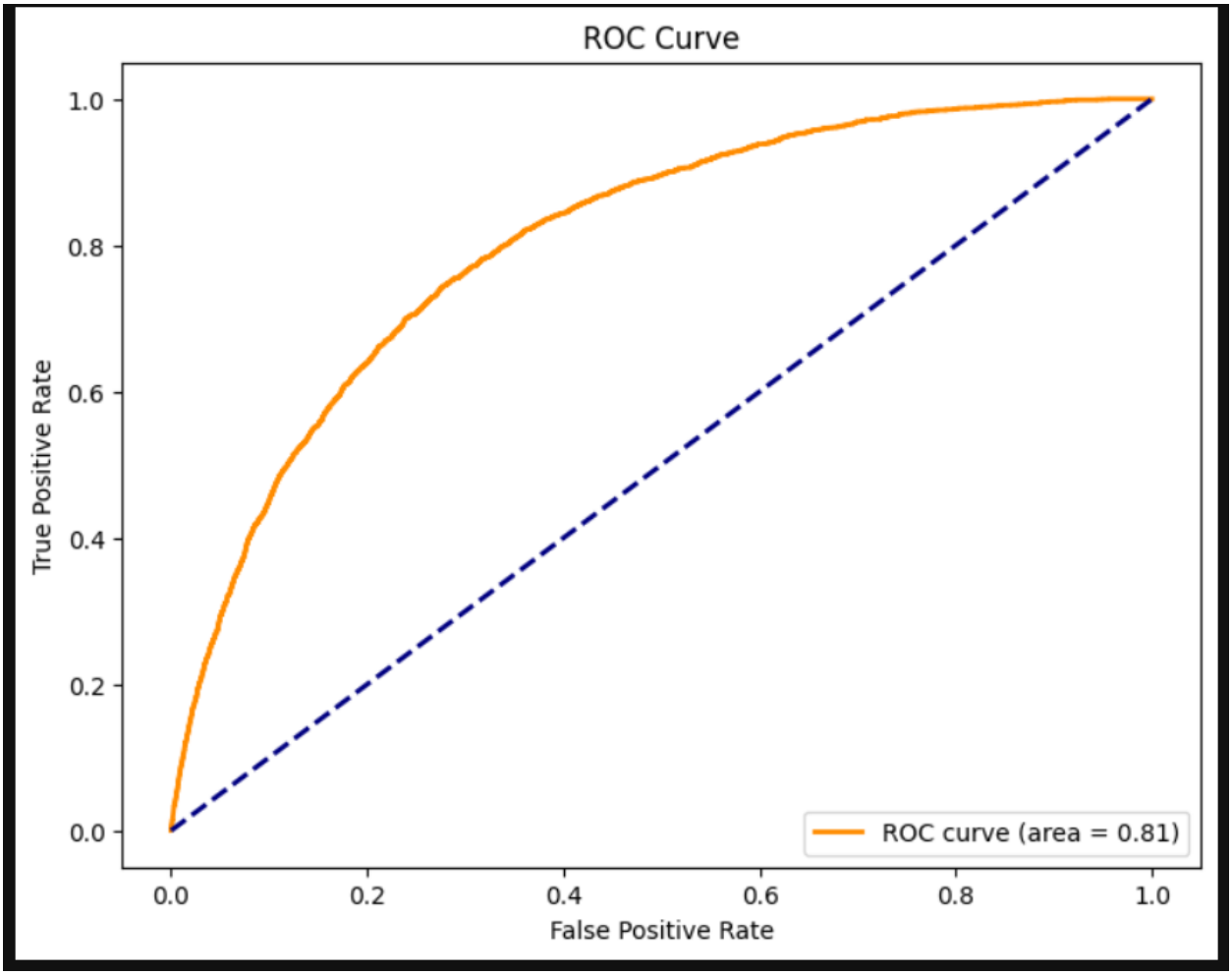
Hyperparameter Tuning

```
Best Hyperparameters: {'learning_rate': 0.1, 'max_depth': 5, 'subsample': 0.8}
Test Accuracy: 0.7355633666808297
Test Precision: 0.04198672356567094
Test Recall: 0.7264150943396226
Test F1-score: 0.07938500156887354
Test ROC-AUC Score: 0.8056997479517999
Test PR AUC Score: 0.06856605366854192

Confusion Matrix for Testing Data:
[[112487  40409]
 [   667   1771]]

Training Accuracy: 0.764850965575147

Confusion Matrix for Training Data:
[[7315 2213]
 [2268 7260]]
```



- After hyperparameter tuning, the XG-Boost model with under-sampling achieved slightly improved performance compared to its initial results.
- Test Accuracy, Precision, Recall, and F1-score have improved marginally.

- ROC-AUC score remains high, indicating good discrimination ability between classes.
- PR AUC Score is relatively low, which might suggest that the precision-recall curve doesn't achieve high precision for all recall levels.
- The confusion matrix for testing data shows the model's performance in terms of true positives, true negatives, false positives, and false negatives. It indicates that the model correctly identifies a significant portion of true positive cases but also misclassifies some negative cases.

Recommendations

1. Encourage NFC Usage Over Physical Cards: Given the lower frequency of transactions associated with physical cards compared to other methods, prioritize promoting NFC (Near Field Communication) payments for offline transactions. This could enhance convenience and security, reducing reliance on physical card transactions.
2. Enhance Security Measures During Midday Peaks: Notable spikes in fraud transactions around 1 PM suggest a potential vulnerability during midday hours. Strengthen security protocols and surveillance systems during this period to mitigate the risk of fraudulent activities. Although overall fraud transaction rates remain relatively stable throughout the day, targeted measures during peak hours could bolster security.
3. Merchant-Led Fraud Awareness Initiatives: Collaborate with merchants to organize educational sessions aimed at raising awareness among customers about common fraud tactics and precautionary measures. Empowering users with knowledge on how to recognize and respond to fraudulent transactions can help prevent financial losses and protect customer trust.
4. Practice Selective Card Information Sharing: Emphasize caution when saving card details with online merchants, advising users to limit such transactions to trusted and reputable vendors. Prioritize security and privacy by encouraging users to exercise discretion in sharing sensitive payment information online, thereby minimizing the risk of unauthorized transactions and data breaches.
5. Emphasize Bank Loyalty and Customer Trust: Highlight the low incidence rate of fraud transactions, constituting only 1.5% of total transactions, as evidence of the bank's commitment to customer security and trustworthiness. Reinforce the bank's dedication to maintaining a secure financial environment and fostering loyalty among its clientele through proactive fraud prevention measures and responsive customer support.
6. Implement Real-Time Transaction Monitoring: Introduce real-time transaction monitoring systems equipped with advanced fraud detection algorithms to

promptly identify and flag suspicious activities. Leveraging data analytics and machine learning capabilities, these systems can enhance fraud detection accuracy and enable swift intervention to mitigate potential risks.

7. Provide Personalized Fraud Alerts: Offer customizable fraud alert settings that enable users to receive real-time notifications for any suspicious transactions detected on their accounts. Empowering customers with the ability to monitor their financial activities closely and take immediate action in case of fraudulent incidents enhances their sense of control and security.
8. Facilitate Secure Authentication Methods: Introduce multi-factor authentication mechanisms, such as biometric authentication or one-time passcodes, to add an extra layer of security during online transactions. By implementing robust authentication protocols, banks can reduce the likelihood of unauthorized access and fraudulent activities, enhancing overall transaction security.

Future Scope

- **Data Transformation and Scaling**
    - Addressing right-skewed continuous data through transformations like logarithmic or square root transformations can help normalize the data distribution and improve model performance.
    - Scaling techniques like Min-Max scaling or Standardization can further preprocess the data to ensure features are on a similar scale, which can benefit models like logistic regression or neural networks.

- **Class Imbalance Handling**

    - Exploring alternative techniques beyond SMOTE for handling class imbalance is crucial. Techniques such as ensemble methods, cost-sensitive learning, or custom loss functions tailored to the specific imbalance pattern in the data could be explored.

- **Model Stacking**

    - Stacking models, such as combining XG-Boost with logistic regression or other base models, can harness the strengths of each model and potentially improve overall predictive performance. Techniques like blending or model averaging could also be considered.

- **Feature Engineering and Insights**

- Implementing filters in the pipeline for feature engineering can help extract more insights from the data, potentially uncovering hidden patterns or relationships that can benefit model performance.
- Providing actionable recommendations to the business team based on the insights gained from the data analysis can add significant value. These recommendations could include targeted marketing strategies, risk management approaches, or customer segmentation strategies, among others.

Dataset Link

As the dataset originates directly from our trusted source, we are unable to provide an online link for public access.

Note: The data set is included in the zip file submitted (transactions.txt).

Number of Meetings Conducted before Final Presentation

We had 15 team meetings either in-person or online.