Data Science Challenge: Card Transactions

Guideline Report

Objective

As fraud can have significant financial and reputational implications for banking and financial institutions, detecting fraudulent transactions quickly and accurately is crucial. Therefore, I aim to develop a predictive model that can effectively identify potentially fraudulent transactions. This model will enhance business operations and enable institutions to offer improved services to their customers.

The purpose of this guideline report is to provide a guide to my initial ideas for this data challenge project, the operational process, summary of results, and aspects that I hope to continue analyzing in the future. I want to use this report as a guidance and explanation report to summarize the ideas in the .ipynb file. I explained the ideas, solutions, implementation, and conclusions in detail in the .ipynb file. Thank you very much for taking the time to review the results of my data challenge project.

Data Load and Preliminary Understanding of Data

1. Introduction

The dataset provided in line-delimited JSON format, contains transactional data with a field "isFraud" to identify fraudulent transactions. The objective of this part is to know the dataset, explore the dataset, clean it, and provide some initial analysis. In order to gain a better understanding of the data, I conducted statistical analyses on each column and provided some important visual interpretations. I also analyzed the data in combination with other columns, attempting to gain a preliminary understanding of the relationships between the data.

2. Data Structure

The original dataset contains 29 columns and 786363 records. From the perspective of whether it is a fraudulent transaction, this dataset is severely imbalanced, which also

corresponds to the situation in real life. After all, fraudulent transactions are a minority, and most of them are normal transactions.

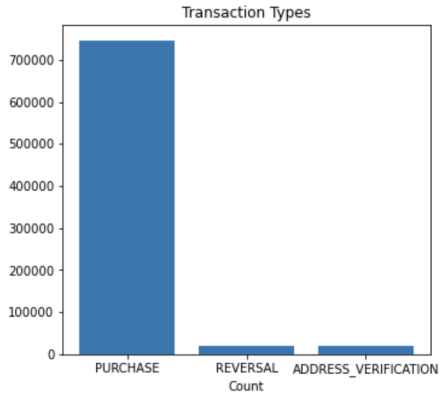
3. Data Cleaning

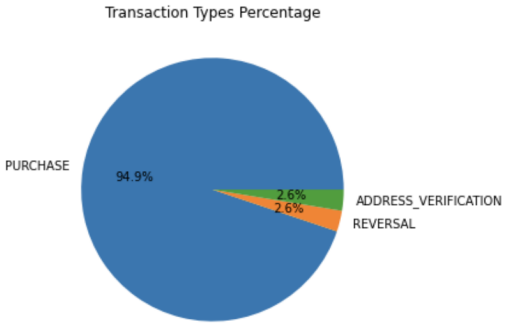
Columns that only contain null values were removed. Null values in other columns were filled where appropriate.

4. Exploratory Data Analysis

After conducting EDA analysis, I discovered many interesting phenomena. Please refer to the .ipynb file for a detailed analysis. Here are some of my impressive findings:

- The dataset contains transactional data for 5000 unique customers. These customers completed over 700,000 credit card transactions in one year, which indicates good purchasing power and a healthy credit card market.
- The majority of transactions (94.8%) are purchase, while reversals(2.6%) and address verifications(2.6%) are relatively infrequent.





- Credit limits only have ten levels, mainly focused on 0 - $20,000, with 428 customers having exceeded their limit in some transactions.
- 4986 customers never changed their address.

- The number of unique EnteredCVV values is not equal to the number of unique CardCVV values, which means that some customers may have mistakenly input the wrong CVV number during transactions. 7,015 transactions may have mistakenly input the wrong CVV number.
- Among all customers, the account number 380680241 had the highest number of fraudulent transactions. The total number of fraudulent transactions in one year (2016) was shocking, with 783 in total, averaging out to approximately 2 fraudulent transactions per day.

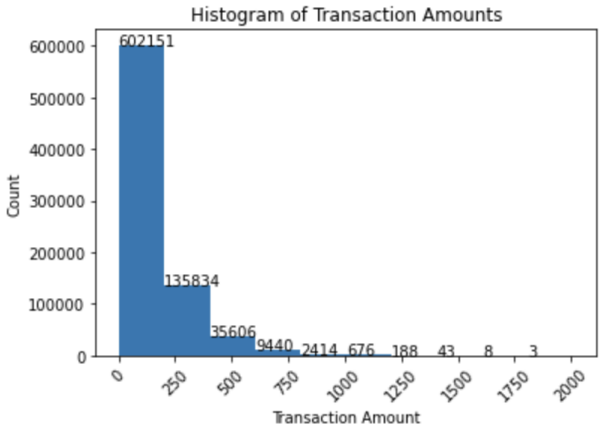
Conclusion:

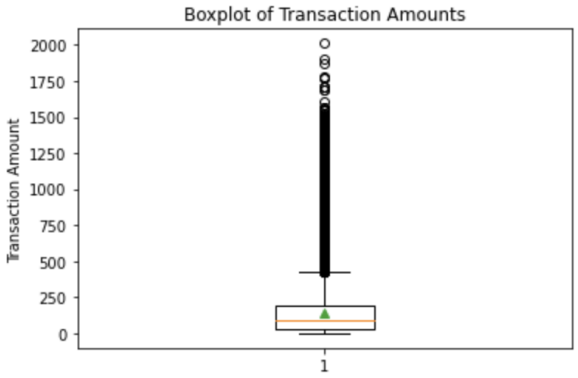
The dataset provided contains valuable information regarding customer transactions. Further analysis could be done using machine learning techniques to predict fraudulent transactions.

Transaction Analysis

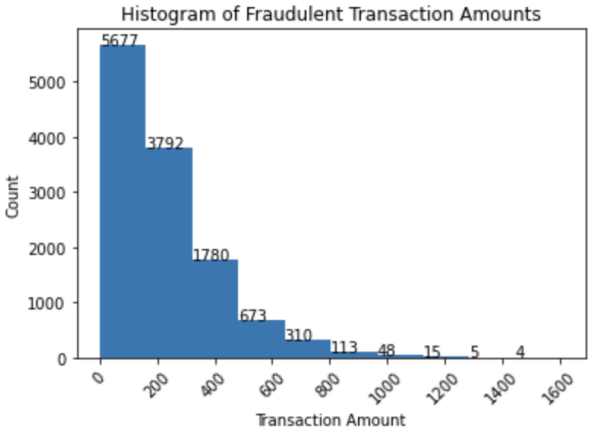
To analyze the transactions, descriptive statistics, box plots, and histograms were used, and the relationships between variables were examined using cross-tabulations and visualizations. These findings could be useful for fraud detection and prevention, as they provide insights into the types of transactions and merchant categories that are more susceptible to fraudulent activity.

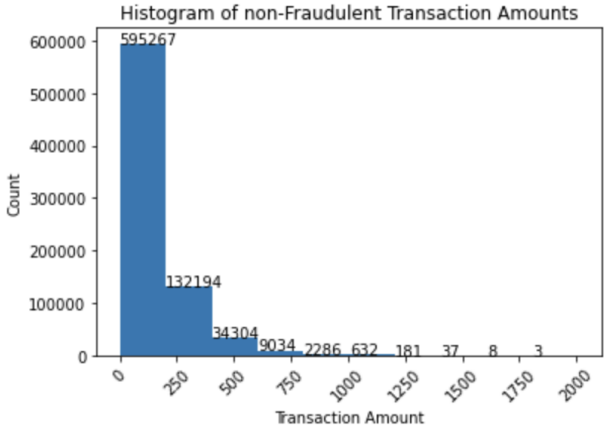
The analysis of transaction amounts revealed that the dataset includes a wide range of transaction amounts, with a mean of $136.99 and a standard deviation of $147.73. The distribution of transaction amounts is right-skewed, indicating that the majority of transactions fall within a certain price range, with a concentration of transactions between 0-$400. Additionally, there are many outliers in the data, indicating that there is a lot of variability in the data.





Further analysis of the relationship between fraudulent and non-fraudulent transactions and transaction amounts showed that both types of transactions have a similar right-skewed distribution, but fraudulent transactions are more likely to occur for smaller amounts.

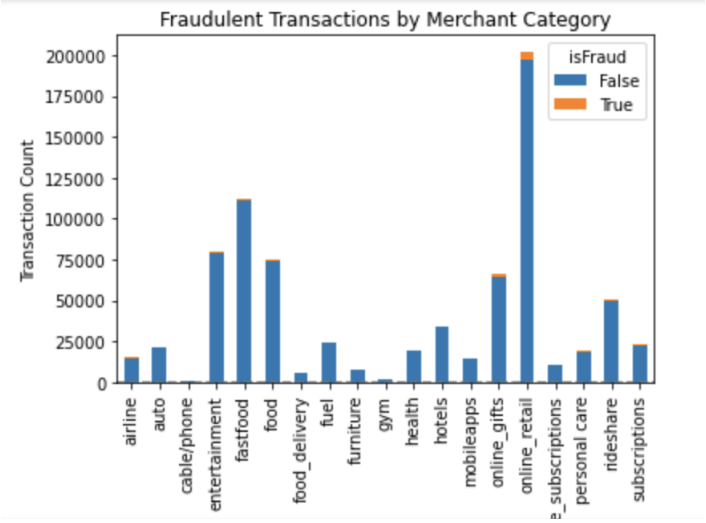




The analysis of merchant categories and transaction amounts revealed that subscriptions had the highest median transaction amounts, followed by entertainment and furniture, while mobile apps, fuel, and gym had the lowest median transaction amounts. Personal care has the highest mean transaction amount, followed by subscriptions and rideshare. The lowest mean transaction amounts are seen in mobile apps, online subscriptions, and gym.The median and mean for each category had

similar trends, indicating that the distribution of transaction amounts for each category is skewed.

Finally, the analysis of the relationship between merchant categories and transaction amounts, transaction counts, and fraud rates showed that some categories had a zero fraud rate, indicating that transactions in these categories are less likely to be fraudulent. On the other hand, some categories, such as "airline", "rideshare", and "online_retail", had higher fraud rates, suggesting that transactions in these categories may be more susceptible to fraudulent activity. Interestingly, the median transaction amount for merchant categories with relatively high fraud rates was higher compared to those with low fraud rates, indicating a correlation between the median transaction amount and the likelihood of fraud in a particular merchant category.



Data Processing

There are duplicated transactions in the dataset. One type of duplicated transaction is a reversed transaction, where a purchase is followed by a reversal. Another example is a multi-swipe, where a vendor accidentally charges a customer's card multiple times within a short time span.

1. Reversal Transactions

In this question, I first raised some assumptions about duplicate transactions. The first assumption related to duplicate transactions, which is the reversal transaction. We will explain what a reversal transaction is, its causes, and its impact on credit card transactions.

Assumption 1:

The first assumption related to reversal transactions is that purchase transactions and reversal transactions are successive, having the same account number, same merchantName, and the same transaction amount. However, after conducting the analysis, it was found that purchase transactions and reversal transactions are not always consecutive. This means that there can be gaps between purchase and reversal transactions.

Assumption 2:

The second assumption related to reversal transactions is that there are purchase transactions and reversal transactions that are not successive, but they have the same account number, same merchantName, and the same transaction amount. Additionally, the purchase was earlier than the reversal. During the analysis, it was also observed that there were multiple matching transactions for reversal requests that could not be explained.

Assumption 3:

The third assumption related to reversal transactions is that there were 2526 reversal transactions that could not be matched. This implies that there may be several transactions that are being reversed, but the original purchase transaction could not be found.

Additional Situations:

Apart from the above assumptions, three additional situations involving reversal transactions were also observed during the analysis:

1. The same purchase transaction being reversed multiple times: In some cases, it was observed that the same purchase transaction was being reversed multiple times. This could happen due to errors in the transaction processing or fraudulent activities.
2. The reversal amount not matching the purchase amount: It was also observed that the reversal amount was not matching the purchase amount. This could happen due to various reasons, such as the merchant refunding only a portion of the purchase amount or the merchant deducting additional charges from the reversal amount.
3. Data Quality: Finally, it was found that some of the issues related to reversal transactions could be attributed to data quality. It is essential to ensure that the data is accurate and complete to conduct a thorough analysis of transactions.

Conclusion

In conclusion, the first assumption related to duplicate transactions, i.e., the reversal transaction, can lead to several issues that need to be considered while analyzing credit card transactions. It is important to be aware of the various situations involving reversal transactions to identify duplicate transactions accurately. Additionally, it is crucial to ensure that the data quality is maintained to conduct an effective analysis of credit card transactions.

Q3 part 1: What total number of transactions and total dollar amount do you estimate for the reversed transactions?

Result 1: As a result of my analysis, I found 17678 reversal transactions and total dollar value of these are $2,655,349.36

Result 2: I was not able to find a purchase associated with reversal transactions (2625) that corresponds to $166,443.14.

2. Multi-Swipe Transactions

In this part, I want to discuss the second assumption related to duplicate transactions, which is the multi-swipe transaction. I will explain what a multi-swipe transaction is, the method used to identify them, and the findings from the analysis.

Assumption of Multi-Swipe Transactions:

The assumption related to multi-swipe transactions is that a customer makes multiple credit card transactions with the same amount within 5 minutes at the same merchant location. This could result in duplicate transactions if not identified correctly.

Method

To identify multi-swipe transactions, I sorted data by 'accountNumber', 'merchantName', 'Timestamp', and 'transactionAmount' in ascending order to ensure accurate matching of customer and merchant transaction time difference columns. Then, all transactions with time differences within 5 minutes were selected and grouped by 'accountNumber', 'merchantName', and 'transactionAmount'. Finally, all transactions where a customer makes more than one transaction with the same amount at the same merchant within 5 minutes were filtered out. These transactions were defined as multiple credit card transactions.

The analysis of multi-swipe transactions yielded the following findings:

1. Multi-swipe transactions are mainly concentrated within 0-3.5 minutes of card-swiping time. This indicates that customers tend to make multiple transactions quickly in a short period.
2. Multi-swipe transactions are mostly concentrated within 2 card-swipes. This indicates that customers tend to make two transactions with the same amount at the same merchant within 5 minutes.

Conclusion

In conclusion, the identification of multi-swipe transactions is crucial to avoid duplicate transactions in credit card transactions. The method used in this analysis helped in

identifying multi-swipe transactions accurately. The findings indicate that multi-swipe transactions are concentrated within a short time period and multiple card-swipes. It is essential to identify and remove such transactions to ensure accurate analysis of credit card transactions.

Q3 part b: What total number of transactions and total dollar amount do you estimate for the multi-swipe transactions?

Result 1: As a result of my analysis, the total number of multi-swipe transactions is 4056 based on the 5 minutes time range.

Result 2: Total dollar value of multi-swipe transactions is $ 594,370.

Q3 part c: Did you find anything interesting about either kind of transaction?

Yeah! Sure!

For reversal transactions, I was not able to match 2526 reversal transactions. I think there are three additional situations involving reversal transactions:

1. The same purchase transaction being reversed multiple times
2. The reversal amount not matching the purchase amount
3. Data quality.

For Multi-swipe transactions, I assume that multiple credit card transactions occur when a customer makes multiple credit card transactions with the same amount within 5 minutes at the same merchant location. And I found that multi-swipe times(frequency) was not too many, mostly concentrated within 2 card-swipes.

1. Multi-swipe transactions are mainly concentrated within 0-3.5 minutes of card-swiping time.
2. Multi-swipe transactions are mostly concentrated within 2 card-swipes.
3. I found 21734 observations as duplicates (combination of both reversal and multi-swipes) that corresponds to 2.76% of original data.

Modeling

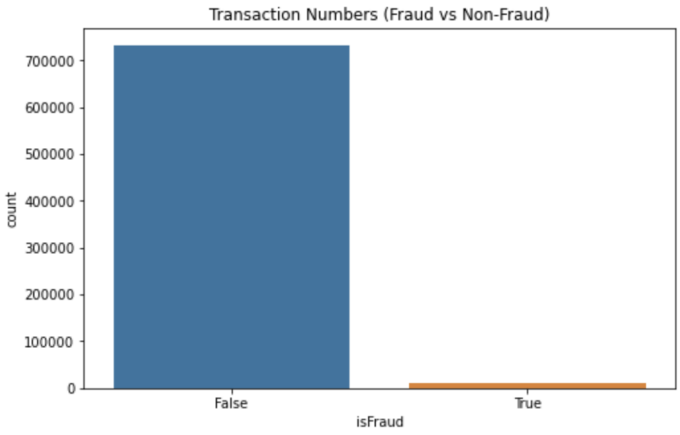
1. Methodology

To build a predictive model for fraud detection, I used a combination of feature engineering, exploratory data analysis, and machine learning. The dataset contains a variety of features, including transaction amount,merchant information, and customer information. I started by exploring the data to identify patterns and potential variables that could be useful for prediction.

Next, I used two different machine learning algorithms - random forest classifier and XGBoost classifier - to build the predictive model. Both algorithms are capable of handling complex data and making accurate predictions, but I wanted to compare their performance to determine which one was better for this particular dataset. I also used grid search to optimize the hyperparameters of both models to achieve the best possible accuracy.

2. Sampling

To address the imbalanced nature of the dataset, I used an undersampling technique where I kept all fraud cases in the dataset and drew a sample (with a ratio of 0.1) from the non-fraudulent transactions of the same customer. This approach helped the model learn the regular patterns of each customer and improved its accuracy.



To further improve the model's generalization ability and its ability to handle noise, I also took samples from customers who have not encountered any fraud cases. This approach ensured that the model was able to identify new types of fraud as they emerged, and also helped to minimize false positives.

3. Features Engineering

As the objective is to predict fraud, I decided to encode the isFraud column using label encoding. For the other categorical features, such as posEntryMode, posConditionCode, and merchantCategoryCode, I chose to use frequency encoding based on the frequency of fraud within each category. This approach aims to capture the relationships between these features and the target variable.

Additionally, I used one-hot encoding for the boolean features to represent them as binary variables. Overall, these encoding strategies were chosen to provide the model with the most relevant and meaningful information, which can improve its predictive power and accuracy.

4. Model choosing

Why did I choose Random Forest Classifier and XGBoost Classifier?

Ensemble learning algorithms that combine multiple decision trees to create a more accurate model.

Random Forest Classifier:

I think it is effective in dealing with high-dimensional datasets with complex relationships between features. It is a good choice for fraud detection because it can handle imbalanced data, where the number of fraudulent transactions is typically much smaller than the number of non-fraudulent transactions. It can also provide an estimate of feature importance, which can be helpful in identifying the most relevant features for fraud detection.

XGBoost Classifier:

XGBoost is particularly good at handling imbalanced data, and it can also handle missing data effectively. XGBoost also provides an estimate of feature importance, which can be used to identify the most relevant features for fraud detection.

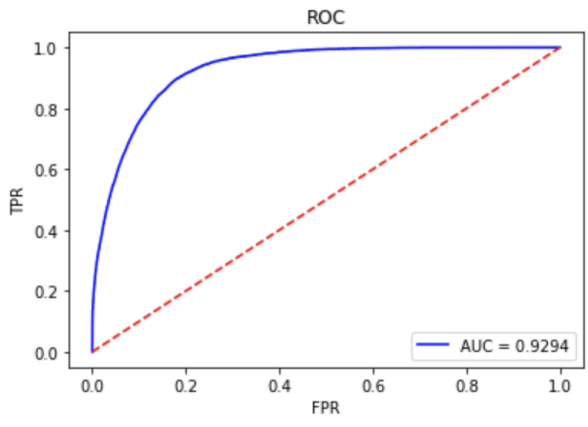
Performance Evaluation

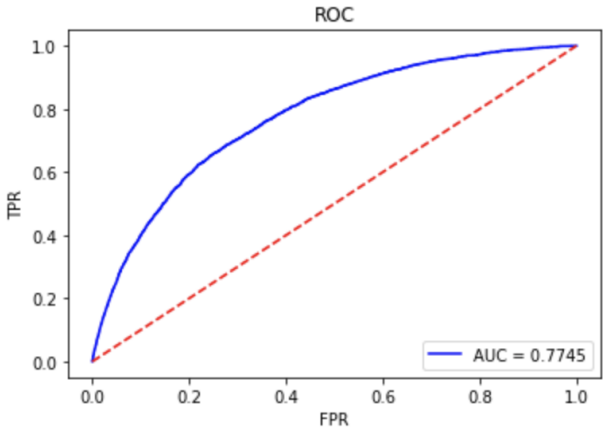
To estimate the performance of the model, I used a sample of the data as a test set to evaluate the accuracy of the model. The test set was randomly selected from the full dataset and included approximately 30% of the data. The model was trained on the remaining 70% of the data.

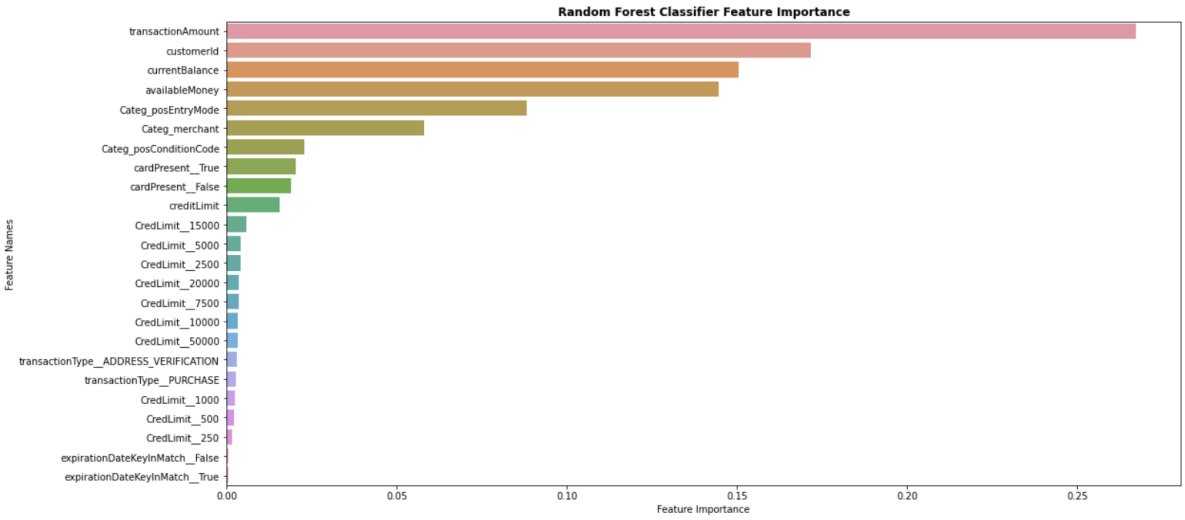
The results of the model were evaluated using a number of different metrics, including accuracy, precision, auc score and recall. Because in fraudulent transactions, we are more concerned about identifying the fraudulent transactions, but in real life, fraudulent transactions are a minority class. Therefore, I chose the AUC score as an evaluation metric.

For Random Forest Classifier model

The accuracy of the model was 0.95, which indicates that the model is able to correctly classify 95% of transactions as either fraudulent or not fraudulent. The precision of the model was 0.90, which means that out of all the transactions predicted to be fraudulent, 90% of them were actually fraudulent. I also plot a feature importance graph to understand what features are important to detect fraudulent transactions.

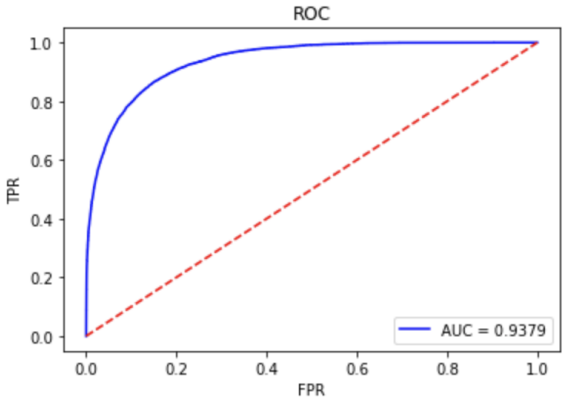


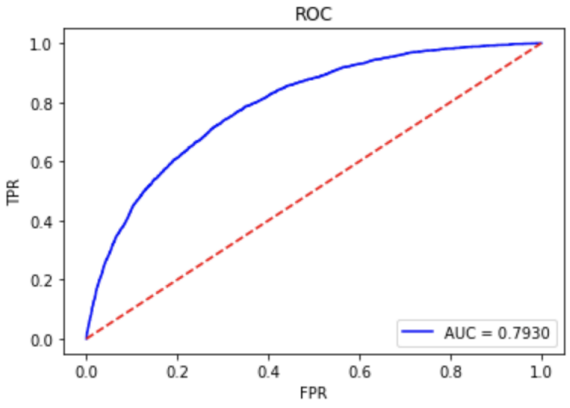


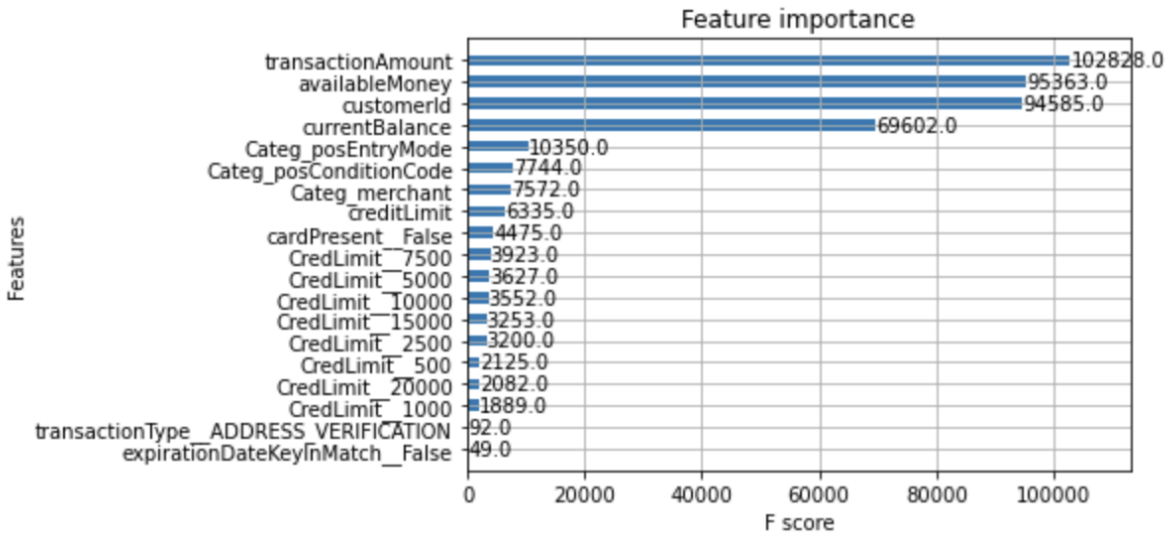


For XGBoost Classifier model

The performance of the XGBoost model was evaluated using the area under the curve (AUC) score. I chose the AUC score since it is not affected by imbalanced data. For this XGBoost model, in the training data, the AUC score is 0.9379, in the test data, it's 0.7930. This suggests that there may be overfitting. I tried GridSearch, also implemented regularization, so it was better than the Random Forest Model, but you know tree models are more prone to overfitting by nature. To better understand which features are important in detecting fraudulent transactions, I created a feature importance chart. Compared to the Random Forest Classifier, the XGBoost model showed that transactionAmount, availableMoney, customerId and currentBalance were more important in detecting fraudulent transactions.







Therefore, it appears that the XGBoost model performs slightly better than the Random Forest model, due to its ability (like regularization term) to prevent overfitting.

Next Steps for Future

With more time, I would explore additional machine learning algorithms, such as neural networks, to see if they could improve the performance of the model.

Tree models are more like overfitting from native. Overly complex models, such as tree models with too many levels, are more prone to overfitting. Simplifying the model by reducing the number of features or levels can help improve its generalization ability, so this is also a direction I will try.

I would also investigate the impact of different feature sets on the accuracy of the model and try to identify which features are most important for fraud detection. Additionally, I would experiment with different sampling techniques, such as oversampling or undersampling, to address any imbalances in the dataset. Finally, I would explore the possibility of using additional data sources, such as external fraud databases or transaction logs, to improve the accuracy of the model.