IST 718

Final Project – Predicting Fraudulent Transactions

Toby Anderson

03/25/2023

# Introduction

# Banks deal with large volumes of funds on a daily basis, while profits are derived from marginal percentages. Because of this, losses from fraudulent transaction can be difficult to cover for banks, and in the worst scenarios, the risk is offset to the customer. Even a small percentage of fraudulent transactions can eat into a banks bottom line, and cause significant mistrust in the customer base. To protect bank and customer funds, fraudulent transactions should be caught quickly and accurately.

In this report, we use banking transaction sample data to demonstrate the prediction of fraudulent transactions. Additionally, modeling allows us to present significant features which are the most useful data for determining a fraudulent transaction which can be used in future models.

# Data Exploration

The dataset is sourced from Kaggle, <https://www.kaggle.com/datasets/sanskar457/fraud-transaction-detection>, and consists of nearly 2 million samples of ATM transactions. A description of the relevant columns is given in Table 1 below.

|  |  |
| --- | --- |
| Column | Description |
| TX\_DATETIME | Timestamp of the transaction occurrence. |
| CUSTOMER\_ID | An anonymous unique identifier for the customer who performed the transaction. |
| TERMINAL\_ID | A unique identifier for the terminal where the transaction occurred. |
| TX\_AMOUNT | The dollar value of the withdrawal. |
| TX\_FRAUD | The target variable, indicating whether the transaction was fraudulent. |

Table 1 – Description of relevant features

The time occurrence of the transactions was normally distributed, with the majority of transactions occurring between the peak hours of 6am – 4pm, as shown in figure 1 below.

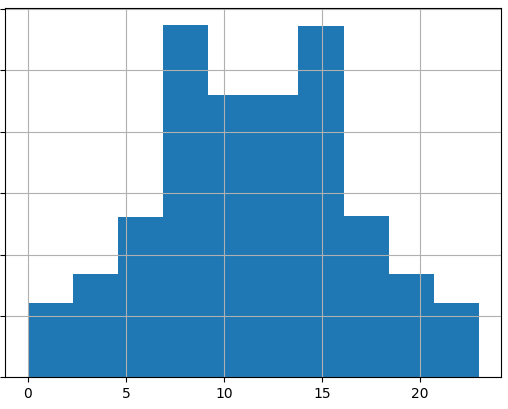


Figure 1 – histogram of transaction time

The transaction amount appears to be logarithmically distributed, with lower dollar amounts occurring far more frequently than larger amounts.

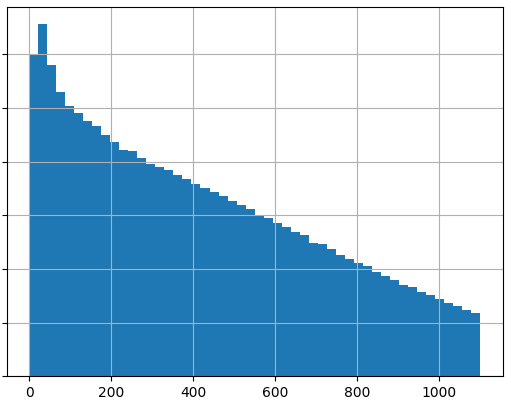


Figure 2 – histogram of transaction dollar amounts

Nearly 1,000 unique terminals were tracked in the dataset, as well as over 5,000 unique customers.

The data is highly imbalanced, with less than 5% of transactions being marked as fraudulent.

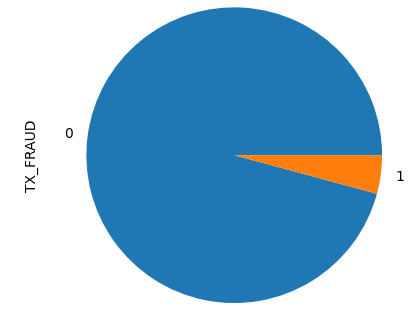


Figure 3 – distribution of fraudulent and normal transactions

# Feature Engineering

In order to accurately model fraudulent transactions, features must be engineered and extracted from the basic level columns. The primary goal will be to develop customer and terminal level features. To accomplish this, the customers and terminals are split into separate data sources, and related to the transactions so that deep feature synthesis can be automated as shown in figure 4.

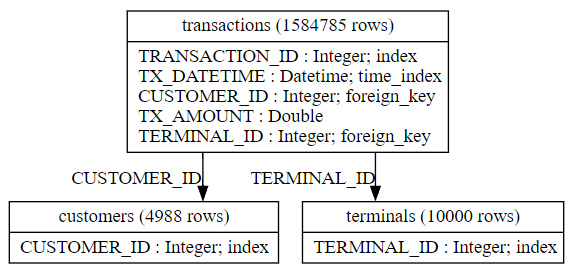


Figure 4 – Data relationships

Using the featuretools python package, several features were created. The engineered features are described in table 2.

|  |  |
| --- | --- |
| Feature | Description |
| Weekend | Denotes whether a transaction occurs on Fri, Sat, or Sun |
| Working\_hours | Denotes whether a transaction occurs during normal working hours. |
| Cust\_avg\_time\_between | The average time between each customer’s transactions. |
| Cust\_max\_tx | The maximum withdrawal by a customer in the dataset. |
| Cust\_mean\_tx | The average withdrawal by each customer in the dataset. |
| Cust\_time\_since\_last | The time since the previous withdrawal by each customer. |
| Term\_avg\_time\_between | The average time between transactions for each terminal. |
| Term\_max\_tx | The maximum withdrawal in the dataset for each terminal. |
| Term\_mean\_tx | The average withdrawal in the dataset for each terminal. |
| Term\_time\_since\_last | The time since the last withdrawal for each terminal transaction. |
| Cust\_lag | The ratio of the time since the last customer transaction and the average time between customer transactions. |
| Terminal\_lag | The ratio of the time since the last terminal transaction to the average time between terminal transactions. |
| Tx\_ratio | The ratio of the customer’s transaction amount to that customer’s average transaction amount. |

Table 2 – Engineered Feature description

After creating the features, the correlation to the target fraudulent variable was calculated and shown in figure 5 below. Based on the correlation measure, it appears that features built out of the transaction amount are the most significant.

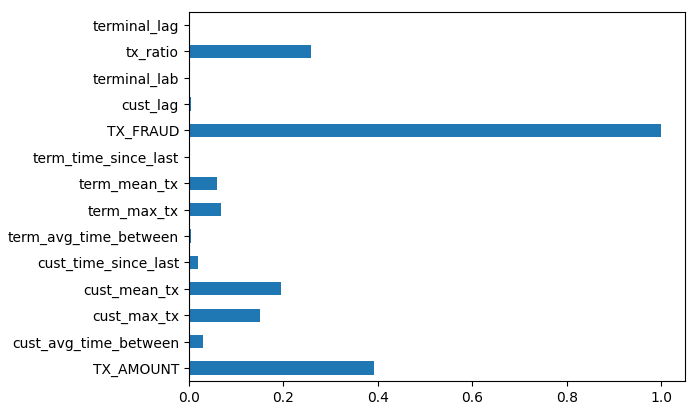


Figure 5 – feature correlation to target

# Prediction

Since the data is so highly imbalanced, the models will be tuned and evaluated based on the F1 score and AUC metric, instead of accuracy. Using accuracy would not give a reasonable measure of the value of the predictions, because it is far more important to flag true fraudulent transactions than to reduce false positive classifications. However, a model can reach 95% accuracy by simply marking all transactions as non-fraudulent, which would defeat the purpose of this prediction exercise.

Three classification models were used for comparison, Decision Tree, Naïve Bayes, and XGBoost. The model parameters and performance summary are given in the results section below.

## Decision Tree

To prevent overfitting of the data onto a highly complex decision tree, cost complexity pruning was used to reduce the number of leaves based on the increase in model accuracy. After tuning multiple parameters, it appears that the decision tree classifier regularly settles on a single criteria using the transaction amount. While this allowed the model to greatly reduce the number of false positives, it caused the model to incur a significant number of false negatives, which is an undesirable result.

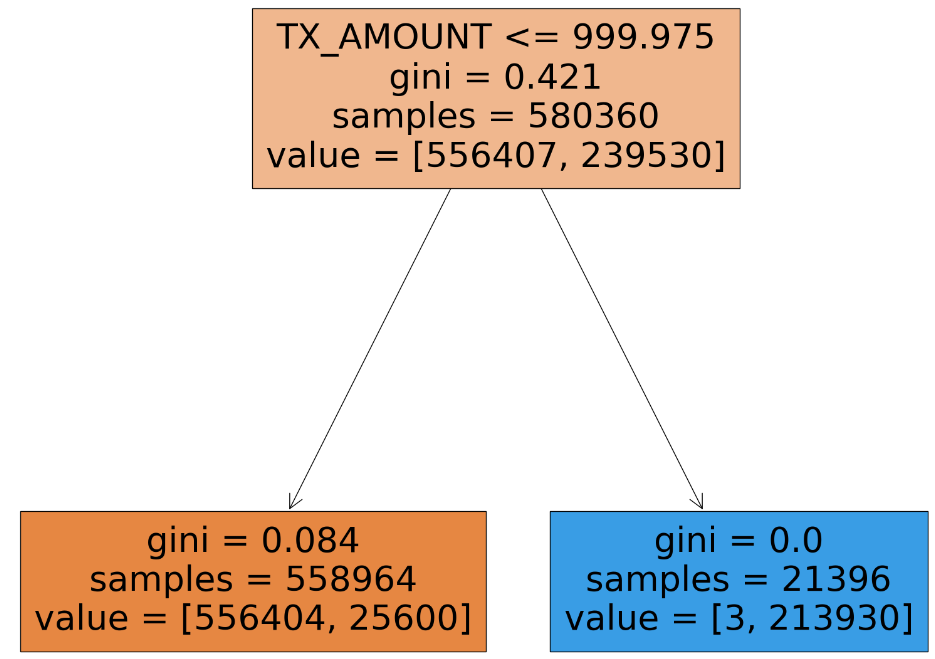


Figure 6 – Decision Tree representation

In summary, the decision tree shows that the transaction amount is the largest indicator of a fraudulent transaction. This is a very simple rule that banks can use to create additional safeguards and checks around transactions exceeding $1000.

## Naïve Bayes

While multiple kernels were attempted while modeling with Naïve Bayes, the Gaussian kernel achieved the best performance. However, overall, the Naïve Bayes performance was very poor. This may be because the model assumes a normal distribution of underlying predictors, however, the most important predictor, transaction amount, was logarithmically distributed, which could introduce significant bias into the model.

Because of the poor performance of the Naïve Bayes models, we chose not to include any significant features from these models in the report, as the model performance does not lend validation to the importance of these features overall. However, the model performance is still included in the results section below.

## XGBoost

XGBoost is a variation on gradient boosted trees, which is a type of ensemble model that achieves a high level of performance at the expense of model complexity. However, through simulation, the importance of each feature can still be extracted from the high performance model.

The most important tuning parameter for the XGBoost model was the evaluation metric. By setting the AUC as the evaluation metric, the model was able to balance the optimization process and avoid the large number of false negatives that the Decision Tree model missed. As expected, the model achieved a high level of performance as shown in the results section below.

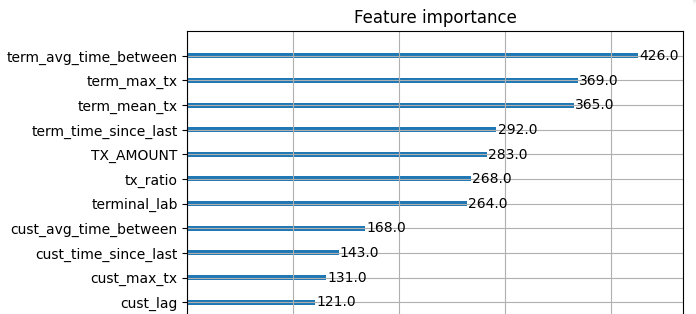


Figure 7 – XGBoost significant features

While the XGBoost model still relied on features derived from the transaction amount, interestingly, it derived the most value from terminal level timing features. This might indicate that fraudulent actors prefer terminals with low levels of activity to avoid suspicion from surrounding customers, and wait for long stretches of inactivity to strike.

Banks can use this information to monitor terminals in more remote locations, and use terminal dormancy to help flag suspicious transaction behavior.

# Results

The summary of the results is given in Table 3 below. Overall, simpler models had great difficulty with the imbalance of datasets, and the results show that more sophisticated models may be required in these cases. However, this report demonstrates that it is a very approachable classification problem which can be solved with the appropriate feature engineering and model selection and tuning.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Parameters | F1 Score | AUC |
| Decision Tree | ccp\_alpha = 0.01  class\_weight={0:1,1:10} | 0.939 | 0.949 |
| Naïve Bayes | Gaussian Kernel | 0.416 | 0.933 |
| XGBoost | subsample=0.9  early\_stopping = 20 | 0.945 | 0.999 |

Table 3 – Summary of model performance