Project 2

Finding Fraud Faster

Niklas Baldis

# Executive Summary

## Analysis

Three total models were created to evaluate the fraud dataset. A logistic regression with 300 iterations was created as a baseline against a random forest and gradient boost. The random forest started with 50 estimators, resulting in an accuracy of about 0.9599 and AUC of 0.8875. After optimizing the random forest using grid search for estimators and sample splits, the accuracy decreased to 0.9569, but the AUC increased to 0.9088. The same baseline and optimize approach was done for the gradient boost method. The baseline resuted in an accuracy of 0.9609 and AUC of 0.9152. The optimized model increased slightly to 0.961 as well as the AUC to 0.9159.

Although the accuracies are very similar, the most telling statistic for fraud detection is F1 score, which combines precision and recall to provide a better picture of the performance of the model. For the logistic regression, the F1 was 0.4476, for the optimized random forest it was 0.4, and for the optimized GBM the F1 was 0.5476. This indicates that GBM may be a better solution for an issue as sensitive as fraud analysis.

## Recommendations

Due to the fact that the optimized GBM had an F1 score much higher than both the logistic regression and the optimized random forest, I would recommend that the firm uses GBM to predict fraud as it provided the best balance between precison and recall, as evidence by the F1 score of 0.5476. This model will have the most performance of the 5 total models analyzed, including the baseline models.

## METHODOLOGY

**Data Exploration and Preprocessing**

1. **Exploratory Data Analysis (EDA) & FEATURE SCREENING**: Conduct an initial analysis to understand the data's characteristics, including distribution of the target variable, missing values, and potential outliers.

The dataset includes both categorical and numerical columns. The "EVENT\_LABEL" column is of particular interest, as it indicates whether a transaction is fraudulent or not. This column is used to train the models to detect fraud. During the exploratory analysis, it was found that account age (in days) and transaction amounts follow a normal distribution for legitimate transactions but exhibit a slightly left skew for fraudulent ones, suggesting a somewhat higher likelihood of fraud with increasing account age. Expectedly, the transaction amounts of an account also increase with account age. The analysis also revealed no significant skew in fraudulent transactions concerning historical velocity.

A diagram of a diagram of a diagram

Description automatically generated with medium confidence A screenshot of a graph

Description automatically generated

A graph of a distribution of account age days with Ryugyong Hotel in the background

Description automatically generatedA graph of a box plot

Description automatically generated with medium confidence

1. **Data Preprocessing**: Address missing values, encode categorical variables, and standardize numerical features to prepare the data for modeling.

Numerous potential outliers were observed on both ends of the variable relationships, as shown in the boxplot. Missing values in the categorical columns were imputed using the most frequent value, while in the numeric columns, missing values were imputed using the mean.

**Model Development**

1. **Model Training**: Develop models using Logistic Regression, Random Forest, and GBM/XGBoost on the training data.

The initial analysis started with a Logistic Regression model to establish a baseline for overall model accuracy. GBM (Gradient Boosting Machine) and Random Forest models were then utilized for their ability to adapt and identify important variables through variable importance metrics. The iteration of these models played a crucial role in improving desired metrics, as they are based on deeper learning techniques. The Logistic Regression model underwent 300 iterations without optimization. The Random Forest model initially had parameters set to n\_estimators=50, n\_jobs=-1, and random\_state=0. Through grid search optimization, the best combination of parameters was determined to be n\_estimators=50 and min\_samples\_split=10. The Gradient Boost model's initial parameters were set to learning\_rate=0.1 and n\_estimators=100. Following optimization using grid search, the optimal parameters were determined to be classifier\_learning\_rate=0.1 and classifier\_n\_estimators=200.

1. **Parameter Tuning**: Optimize model parameters to enhance performance.

To optimize Random Forest, I performed a grid search of the estimators and minimum samples split to find the best combination of parameters, which were n\_estimators=50 and min\_samples\_split=10. I also leveraged grid search for our optimized Gradient Boost model to find the best parameters: classifier\_learning\_rate: 0.1 classifier\_\_n\_estimators: 200.

1. **Feature Selection**: Identify and retain the most informative features for the models.

I utilized the `get\_feature\_names\_out` package from the OneHotEncoder library to extract the features of each model. The analysis revealed that, for the Random Forest model, the most important variables were `transaction\_adj\_amt` (0.668754), `account\_age\_days` (0.147781), and `historic\_velocity` (0.092815). Similarly, for the Gradient Boost model, the most influential variables were `transaction\_adj\_amt` (0.646816), `account\_age\_days` (0.226095), and `historic\_velocity` (0.095214).

**Model Evaluation**

1. **Performance Metrics**: Evaluate models using accuracy, AUC-ROC, precision, recall, and F1-score.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Baseline Random  Forest | Optimized Random  Forest | Baseline GBM | Optimized GBM |
| Accuracy | 0.9599 | 0.9569 | 0.9609 | 0.9610 |
| Precision | 0.7610 | 0.8488 | 0.7585 | 0.7534 |
| Recall | 0.3942 | 0.2617 | 0.4238 | 0.4301 |
| AUC | 0.8875 | 0.9088 | 0.9152 | 0.9159 |
| F1 | 0.5193 | 0.400 | 0.5438 | 0.5476 |

1. **Feature Importance Analysis**: Determine the most influential features in predicting fraudulent transactions.

|  |  |
| --- | --- |
| Feature | Importance |
| transaction\_adj\_amt | 0.668754 |
| account\_age\_days | 0.147781 |
| historic\_velocity | 0.092815 |
| transaction\_amt | 0.073502 |
| tranaction\_initiate\_N | 0.001147 |
| tranaction\_initiate\_I | 0.001013 |
| tranaction\_initiate\_P | 0.000912 |
| tranaction\_initiate\_L | 0.000811 |
| tranaction\_initiate\_O | 0.000785 |
| tranaction\_initiate\_Q | 0.000755 |

**Random Forest**

**GBM**

|  |  |
| --- | --- |
| Feature | Importance |
| transaction\_adj\_amt | 0.646816 |
| account\_age\_days | 0.226095 |
| historic\_velocity | 0.095214 |
| transaction\_amt | 0.028524 |
| tranaction\_initiate\_R | 0.000602 |
| tranaction\_initiate\_J | 0.000402 |
| tranaction\_initiate\_W | 0.000331 |
| tranaction\_initiate\_N | 0.000323 |
| tranaction\_initiate\_I | 0.000296 |
| tranaction\_initiate\_Z | 0.000258 |

1. **Model FPR/TPR/Threshold Table**

**Random Forest**

|  |  |  |
| --- | --- | --- |
| Target FPR (%) | Expected TPR | Threshold |
| 1 | 0.410680 | 0.371722 |
| 2 | 0.517961 | 0.275163 |
| 3 | 0.588350 | 0.197714 |
| 4 | 0.644660 | 0.150783 |
| 5 | 0.679126 | 0.126603 |
| 6 | 0.708252 | 0.110193 |
| 7 | 0.732039 | 0.096359 |
| 8 | 0.752913 | 0.085186 |
| 9 | 0.764078 | 0.077240 |
| 10 | 0.778155 | 0.070187 |

**GBM**

|  |  |  |
| --- | --- | --- |
| Target FPR (%) | Expected TPR | Threshold |
| 1 | 0.454854 | 0.453908 |
| 2 | 0.554854 | 0.307959 |
| 3 | 0.628155 | 0.231008 |
| 4 | 0.666990 | 0.183183 |
| 5 | 0.699515 | 0.145654 |
| 6 | 0.725728 | 0.115210 |
| 7 | 0.750000 | 0.095749 |
| 8 | 0.772816 | 0.079560 |
| 9 | 0.791748 | 0.068227 |
| 10 | 0.803398 | 0.059882 |

**Insights and Recommendations**

1. **Model Comparison**: Compare the models based on their performance and feature importance scores to identify the most effective model.
   1. **ROC Charts for each model on Test Set.**

A graph of a graph of a graph

Description automatically generated with medium confidence

Based on the above outputs, the optimized GBM model is the most effective. It is tied for the highest accuracy, has the highest AUC, the highest recall, and the highest F1 score. All of these defend the consistency and effectiveness of the Optimized Gradient Booster model compared to others.

1. **Feature Evaluation**: Discuss the importance of email domain and billing postal code as predictors of fraud.

Email domain and billing postal code were not in the top 10 most important variables for predicting fraud and, therefore should not be taken into account.

1. **Operational Strategy at 5% FPR**: Propose a strategy to achieve and maintain a 5% false positive rate, detailing its implications on recall and precision.

To achieve and maintain a 5% false positive rate, the predicted probability threshold should be greater than or equal to 0.126603. This rule will catch 67% of all frauds while incorrectly classifying 5% of legitimate transactions as fraud. Additionally, the precision at 0.126603 and above is 95%.

**Plain Language Explanations**

1. **Random Forest vs. GBM/XGBoost**:   
   In Random Forest, the model aggregates predictions from multiple decision trees. Each tree is trained independently on a random subset of the data and features. The final prediction is made by averaging the predictions of all the trees (regression) or taking a majority vote (classification). On the other hand, Gradient Boosting builds trees sequentially, where each new tree corrects errors made by the previous ones. It focuses more on difficult-to-predict instances, gradually improving the model's performance.
2. **Understanding 5% False Positive Rate**:

Operating at a 5% false positive rate means that out of every 100 transactions flagged as legitimate, 5 are falsely labeled as legitimate. Although there is a risk in allowing fraudulent transactions to go through, the model correctly identifies both fraudulent and legitimate transactions at a 95% accuracy rate. However, the system still makes occasional mistakes, highlighting the need for fine-tuning to improve efficiency and customer experience. In this scenario, using a threshold of 0.126603 for predicted probability would capture 67% of all frauds with a precision of 95% while incorrectly flagging 5% of legitimate transactions as fraudulent.