Finding Fraud Faster - Project 2

Executive Summary of Fraud Detection Methods Using Logistic Regression, Random Forests, and GBM

Author:
Kaitlyn Vickers

February 24, 2024

Table of Contents

1 Data Exploration and Preprocessing

This project focuses on a dataset designed for fraud detection in financial transactions. The data includes various features such as transaction amount, email domain, IP address, user agent, and 22 other variables for identifying potential fraudulent activities. For a detailed description of each feature, please refer to the data definition in the appendix.

The dataset contains 125,000 entries, each representing a unique transaction, and is spread across 27 columns. This extensive dataset allows us to explore and analyze the patterns and characteristics of transactions to differentiate between legitimate and fraudulent ones. The initial step in our analysis involves data exploration and preprocessing to prepare the data for the subsequent modeling phase.

1.1 Exploratory Data Analysis

Approximately 5.7% of instances in our data set are flagged as fraud; the dataset comprises 118,215 legitimate transactions and 6,785 fraudulent cases. This imbalance highlights the challenge of detecting fraudulent activities within a predominantly legitimate transaction environment, suggesting that using a synthetic data sampling technique in addition to our models may be beneficial. However, due to the size of our data and deadline, synthetic data sampling was not performed in this analysis.

The dataset spans a timeframe from October 25, 2020, to October 25, 2021, providing a comprehensive view of transactional activities over one year. Upon examining the numerical columns, we found them to be fairly evenly distributed, with no significant skewness, suggesting that the data does not require extensive transformation for modeling purposes. The following boxplots 1 and 2 illustrate this finding:

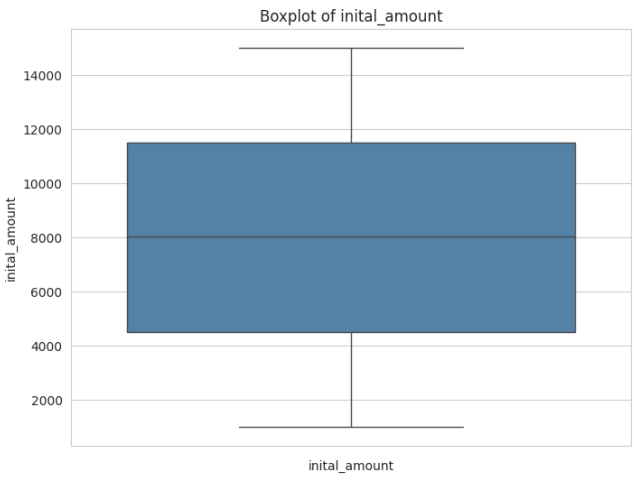

Figure 1: Box Plot of Initial Amount

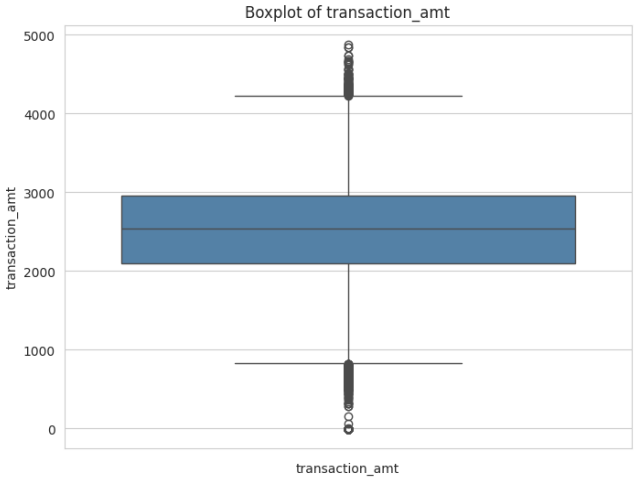

Figure 2: Box Plot of Transaction Amount

Our correlation analysis revealed that the variables in the dataset are generally uncorrelated, with the highest correlation being approximately 0.47 between account age and transaction amount. This low correlation suggests that multicollinearity is unlikely to be a concern in our modeling efforts. The following heatmap 3 of the correlations provides a clear overview of the relationships between variables, aiding in the selection of features for our models.

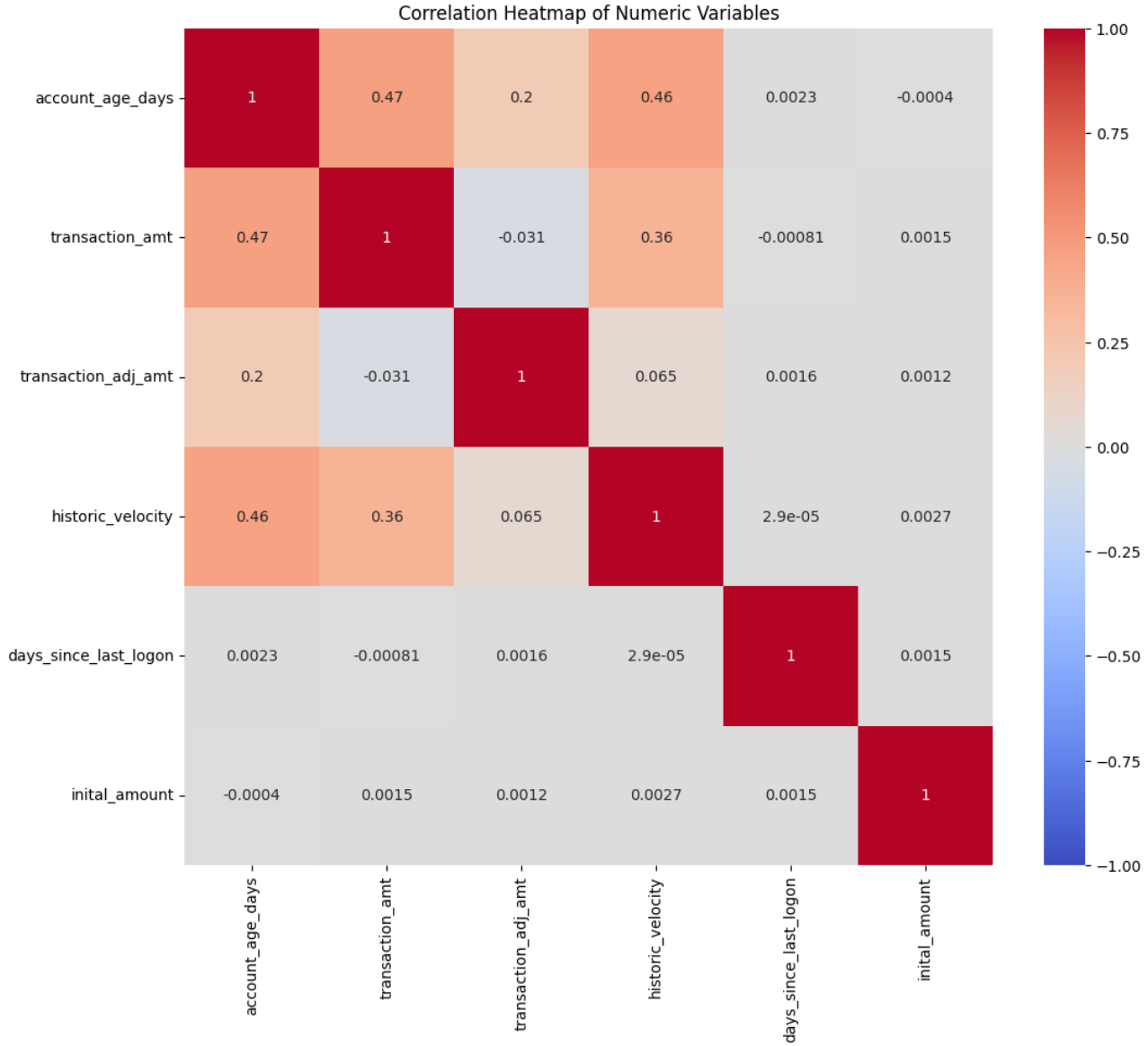

Figure 3: Correlations of Numeric Variables

When comparing the distribution of legitimate and fraudulent transactions across variables using the fraud flag, we found that the distributions mostly align, except for the account age and transaction amount, both slightly higher in fraudulent cases, and the transaction adjusted amount, which is significantly lower in fraudulent cases than in legitimate ones (see figure below 4. This observation could be crucial for distinguishing between the two types of transactions.

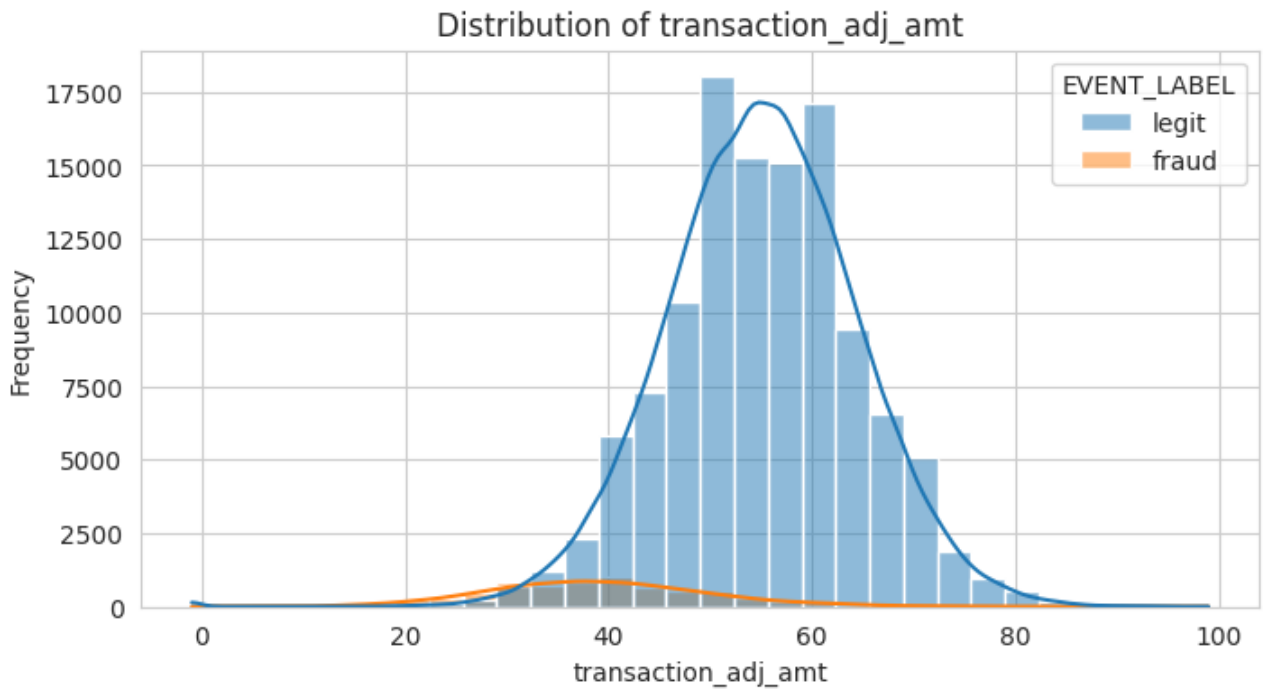

Figure 4: Adjusted Transaction Amount Fraud vs Legit

Additionally, in terms of currency usage, Canadian dollars are the most prevalent in the dataset, being used more than twice as often as all other currencies (USD and EUR) combined. Interestingly, while Indiana is the most popular billing state, Rhode Island has the highest number of fraud cases, suggesting that geographical location may play a role in fraud occurrence.

1.2 Feature Engineering and Screening

In our feature engineering process, we focused on extracting key attributes to enhance our model's predictive accuracy. We created a 'month' column to capture transaction timing and derived device-specific columns from the 'user_agent' field. We also introduced a column for phone number length and a binary column indicating signature presence.

For the postal code and email domain, we adopted a selective approach to reduce the number of variables. Given the high diversity of postal codes, we created a column representing the first digit, simplifying the geographical information while retaining its relevance. Similarly, for email domains, we categorized them into broader types such as '.com', '.net', '.biz', '.org', and '.info', to manage the variety without losing the essence of the data. This approach helped in decreasing the total number of variables in our model, making it more manageable and effective.

To further streamline our dataset, we dropped several columns at the end of the feature engineering process, including 'ip_address', 'email_domain', 'phone_number', 'billing_city', 'billing_postal', 'card_bin', 'cvv', 'applicant_name', 'billing_address', 'merchant_id', 'locale', 'transaction_initiate', and 'EVENT_TIMESTAMP'.

1.3 Dealing with Nulls

In addressing null values within our dataset, we adopted a strategic approach to ensure the integrity of our model. As previously mentioned, null values in the 'signature_image' column were interpreted as the absence of a signature image, which was reflected in the new binary column created for signature presence. For all other null values, we implemented a pipeline to fill missing data efficiently. Numerical values were imputed with the mean of their respective columns, while categorical values were filled with the most frequent category in each column.

2 Model Development

2.1 Logistic Regression

In the model development phase of our fraud detection project, we initially focused on Logistic Regression as a baseline model. This choice was motivated by the interpretability and simplicity of Logistic Regression, making it a suitable starting point for our analysis. We adopted a standard approach to partitioning our dataset into training and testing sets, allocating 70% of the data for training and reserved the remaining 30% for testing.

As part of our exploration, we also developed regularized versions of the Logistic Regression model to investigate their potential in improving performance. Specifically, we created models with L1 (Lasso), L2 (Ridge), and Elastic Net penalties, each with different regularization strengths. These regularized models were intended to address potential issues of over fitting and to enhance the generalization capabilities of our baseline model.

2.2 Random Forests

Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. This algorithm is particularly well-suited for our project due to its ability to handle large datasets with a high dimensionality of features and its robustness to overfitting, making it ideal for complex classification tasks such as fraud detection.

2.2.1 Model Training

For training our Random Forest models, we followed a similar data splitting strategy as used in our Logistic Regression model, allocating 70% of the data for training and 30% for testing. This split ensures that the model has a substantial amount of data to learn from, while also providing a significant portion for evaluating its performance on unseen data.

We developed two versions of the Random Forest model:

1. **Baseline Random Forest:** This model was trained with default parameters to establish a baseline for performance comparison. It served as a starting point for understanding the potential of the Random Forest algorithm in our fraud detection task.
2. **Random Forest with Optimal Parameters:** To improve upon the baseline model, we conducted parameter tuning to find the optimal settings for the Random Forest algorithm. This version of the model was trained with the best parameters identified through the tuning process, aiming to enhance its predictive accuracy and generalization ability.

By creating and comparing these two models, we aimed to assess the impact of parameter optimization on the Random Forest algorithm's effectiveness in detecting fraudulent transactions.

2.2.2 Parameter Tuning

In our fraud detection project, tuning the parameters of the Random Forest model was crucial to optimize its performance. We focused on adjusting several key parameters, including the number of trees in the forest, the maximum depth of each tree, the minimum number of samples required to split an internal node, and the minimum number of samples required for a leaf node.

To find the best combination of these parameters, we used GridSearchCV, a technique that systematically tests all possible parameter combinations. It evaluates the model's performance for each combination using cross-validation, which involves dividing the data into multiple parts, training the model on some parts, and testing it on others. The combination that yields the best cross-validation score is selected as the optimal set of parameters.

2.3 Gradient Boosting Model (GBM)

Gradient Boosting Machines (GBM) are a powerful ensemble technique that builds models sequentially, with each new model correcting errors made by previous models. This approach makes GBM particularly effective for complex datasets with intricate patterns, making it a suitable choice for our fraud detection project. The ability of GBM to handle imbalanced data, as often found in fraud detection scenarios, further underscores its appropriateness for this task.

2.3.1 Model Training

For this project, we developed three distinct Gradient Boosting models to explore different aspects of our dataset:

1. **Baseline Model:** This model was created with default parameters to serve as a point of comparison for other models.
2. **Model with Optimal Parameters:** This version was fine-tuned through parameter optimization to enhance its predictive accuracy.
3. **Model with Top 10 Features:** To investigate the impact of feature selection, this model was trained using only the ten most important features identified in the optimally parameterized model.

2.3.2 Parameter Tuning

In the parameter tuning phase for our Gradient Boosting Model, we focused on optimizing two key parameters using GridSearchCV: the model learning rate and number of estimators. The learning rate controls how quickly the model adapts to the data, with lower values typically leading to more robust models. The number of estimators refers to the number of sequential trees built, with more trees generally improving model accuracy up to a point.

GridSearchCV was employed to search for the best combination of these parameters. Similarly to before, it evaluated various combinations by training the model with each set and assessing its performance through cross-validation. The combination that yielded the highest cross-validation score was selected as the optimal set for our model.

This tuning process was applied to the Optimal Parameters model, refining its settings to enhance its predictive capability for our fraud detection.

2.3.3 Feature Selection

In the feature selection phase for our Gradient Boosting Model, we focused on simplifying the model by using only the most impactful features. After assessing feature importance scores from the Optimal Parameters model, we identified the top 10 features that had the greatest influence on the model's predictions. These features included both numerical columns and categorical columns that were transformed into one-hot encoded variables.

To create a more streamlined model, we constructed a new Gradient Boosting Model that only considered these top 10 features. By doing so, we aimed to reduce the complexity of the model and potentially improve its interpretability and computational efficiency.

3 Model Evaluation

3.1 Performance Metrics

Evaluating the performance of our models is crucial to ensure they are effectively identifying fraudulent transactions. We use several performance metrics to assess the strengths and weaknesses of our models:

- **Accuracy:** Measures the proportion of correct predictions (both true positives and true negatives) out of all predictions. While accuracy is a straightforward measure of overall performance, it can be misleading in imbalanced datasets, where the number of legitimate transactions significantly outweighs the number of fraudulent ones.
- **Precision:** Assesses the accuracy of positive predictions. It is the proportion of true positive predictions (correctly identified fraud) out of all positive predictions. In fraud detection, high precision is important to minimize false alarms, where legitimate transactions are incorrectly flagged as fraud.
- **Recall (Sensitivity):** Measures the ability of the model to detect all positive instances. It is the proportion of true positive predictions out of all actual positives. High recall is crucial in fraud detection to ensure that as many fraudulent transactions as possible are identified and investigated.
- **ROC-AUC (Area Under the Receiver Operating Characteristic Curve):** The ROC-AUC score evaluates the model's ability to distinguish between classes (legitimate and fraudulent transactions) across different thresholds. A higher AUC indicates better model performance in separating the two classes.
- **F1-Score:** The F1-Score is the harmonic mean of precision and recall. It provides a balance between these two metrics, making it a useful measure when there is a trade-off between precision and recall. In fraud detection, a high F1-Score indicates that the model is performing well in both identifying fraud and minimizing false positives.

Each of these metrics provides a different perspective on the model's performance. Together, they offer a comprehensive evaluation, helping us understand how well our models are performing.

3.2 Model Comparison

The following table 1 outlines our key performance metrics across models, with our chosen model bolded:

Table 1: Performance Metrics Across Models

Model	Accuracy	Precision	Recall	ROC-AUC	F1-Score
Log Regression	0.9701	0.8468	0.5553	0.9360	0.6708
Random Forest	0.9733	0.9329	0.5534	0.9419	0.6947
Random Forest - Opt	0.9497	0.9887	0.0850	0.9284	0.1565
GBM	0.9702	0.8709	0.5369	0.9329	0.6643
GBM - Opt	**0.9764**	**0.9433**	**0.8946**	**0.6471**	**0.7510**
GBM - Feature Select	0.9704	0.8739	0.5383	0.9332	0.6663

The model chosen is our Gradient Boosted model, due to it having the highest overall accuract (0.9764) as well as high levels of all other metrics.

3.3 Feature Importance Analysis

In our optimal Gradient Boosting model for fraud detection, feature importance helps identify the most influential attributes in predicting fraudulent transactions. The top 10 features by importance are:

1. **transaction_adj_amt (0.504494):** The adjusted transaction amount is the most critical factor, indicating that the size of a transaction is a key indicator of fraud.
2. **account_age_days (0.086593):** The age of the account suggests that newer accounts might be more prone to fraudulent activities.
3. **transaction_env_N (0.049415):** Certain transaction environments are more associated with fraud.
4. **transaction_amt (0.046570):** Similar to the adjusted amount, the raw transaction amount is also significant.
5. **transaction_env_V (0.037335), transaction_env_B (0.032550), transaction_env_Q (0.018835):** These features highlight the importance of the transaction environment in fraud detection.
6. **historic_velocity (0.025746):** The frequency of transactions over time can be an indicator of suspicious activity.
7. **currency_usd (0.013571):** Transactions in US dollars have a specific fraud risk profile.
8. **transaction_type_S (0.013494):** Certain types of transactions are more likely to be fraudulent.

These features underscore the multifaceted nature of fraud detection, where factors like transaction amounts, account age, transaction environment, and transaction types all play

critical roles. Refer to graphic 6 in Appendix for a visual representation of feature importances.

3.4 FPR/TPR/Threshold Table

In fraud detection, the False Positive Rate (FPR) represents the proportion of legitimate transactions incorrectly classified as fraudulent. Minimizing the FPR is crucial to reduce unnecessary alerts and maintain user trust. The True Positive Rate (TPR), also known as recall or sensitivity, measures the proportion of actual fraudulent transactions correctly identified by the model. A higher TPR indicates better detection of fraud.

The threshold is a critical value that determines the classification boundary between fraudulent and legitimate transactions. Adjusting the threshold allows us to balance the trade-off between the TPR and FPR. By setting different threshold values, we can achieve various levels of sensitivity and specificity, enabling us to tailor the model's performance to specific operational requirements.

The FPR/TPR/Threshold table provides insights into how the model's performance changes at different levels of stringency, helping us to select an optimal threshold that balances the need for fraud detection with the minimization of false positives.

Table 2: FPR/TPR/Threshold Table

Target FPR (%)	Expected TPR	Threshold
1.0	0.721359	0.317451
2.0	0.782524	0.173884
3.0	0.811650	0.121140
4.0	0.835437	0.091519
5.0	0.849515	0.072559
6.0	0.861650	0.060354
7.0	0.872816	0.050771
8.0	0.882524	0.044142
9.0	0.888835	0.039396
10.0	0.893689	0.035458

4 Insights and Recommendations

4.1 Chosen Model Performance

Our chosen Gradient Boosting Machine (GBM) model with optimal parameters demonstrates strong performance in fraud detection, highlighted by its accuracy of 0.9764. This

high accuracy, the primary reason for selecting this model, indicates that it correctly classifies approximately 97.64% of all transactions. The model's AUC-ROC score of 0.9433 reflects its excellent ability to distinguish between fraudulent and legitimate transactions. With a precision of 0.8946, the model ensures that a high proportion of transactions flagged as fraud are indeed fraudulent, minimizing false alarms. The recall of 0.6471 indicates that the model successfully identifies about 64.71% of all fraudulent transactions, while the F1-score of 0.7510 suggests a good balance between precision and recall. Overall, the model's performance metrics demonstrate its effectiveness in detecting fraud while maintaining a high level of accuracy.

4.2 Feature Evaluation

When approaching categorical variables like email domains and billing postal codes,we recognized the challenge of handling these high-dimensional variables. To address this, we transformed these variables into more manageable forms: extracting the domain type (e.g., .com, .org, .net) from email addresses and using only the first digit of the postal codes.

The email domain can be a significant predictor of fraud, as certain domains might be associated with higher instances of fraudulent activity. For example, free or less-regulated email services might be more commonly used for fraudulent purposes. By categorizing emails into broader domain types, we retain valuable information about the source of the email while reducing the complexity of our model.

Similarly, the first digit of the billing postal code can provide useful geographical information that correlates with fraud patterns. Different regions may have varying levels of fraud risk, and this transformation allows us to capture these regional differences without the need to manage a large number of unique postal codes.

However, neither categorical predictor was in any if our model's Top 10 Features by Importance. Overall, these transformations help simplify our model while preserving the predictive power of email domains and billing postal codes as indicators of potential fraud.

4.3 Ideal FPR

To achieve and maintain a 5% false positive rate (FPR) in our fraud detection system, we can adjust the decision threshold based on our model's predictions. By setting a specific threshold for the predicted probability of a transaction being fraudulent, we can control the trade-off between the false positive rate and the true positive rate (recall).

For example, if we set the threshold at 0.072559, as indicated by our model's performance metrics at 5% FPR, transactions with a predicted fraud probability equal to or greater than this threshold will be classified as fraud. This rule is expected to catch approximately 84.95% of all fraudulent transactions (as per the interpolated true positive rate at 5% FPR) while incorrectly classifying 5% of legitimate transactions as fraud.

At this threshold, the precision of our model is expected to be lower than at higher thresh-

olds, meaning that while we are effectively reducing false negatives (missed frauds), we are also increasing false positives (legitimate transactions flagged as fraud).

In practice, maintaining a 5% FPR would involve continuous monitoring and adjustment of the threshold based on evolving fraud patterns and business objectives. It may also be beneficial to implement additional verification processes for transactions flagged as fraud near the threshold to minimize the impact of false positives on legitimate users.

5 Model Predictions

On a new data set provided, our chosen GBM model predicted 24,011 legitimate and 989 fraud cases in the holdout dataset. The summary statistics for the predicted fraud probabilities show a mean of 5.42%, indicating that, on average, transactions are deemed low risk. The distribution is heavily skewed towards lower probabilities, with a maximum value near 100%, suggesting that while most transactions are considered low risk, a few are identified with high certainty as fraudulent. This skewness highlights the model's ability to differentiate between typical and suspicious transactions effectively.

6 Discussion

6.1 Random Forest vs. GBM/XGBoost

In our analysis, we explored both Random Forest and Gradient Boosting Machine (GBM)/XGBoost models for fraud detection. While we have previously discussed each model in detail, here we'll briefly compare their approaches and implications for fraud detection.

Random Forest is an ensemble method that builds multiple decision trees during training and aggregates their predictions for the final output. Each tree is trained on a random subset of the data, making the model robust against overfitting and providing a diverse perspective on the data. In fraud detection, Random Forest can quickly identify important features and provide a reliable baseline for identifying fraudulent transactions.

Gradient Boosting Machine (GBM)/XGBoost, on the other hand, builds trees sequentially, with each new tree correcting errors made by previous ones. This iterative process allows the model to focus on challenging cases, improving its accuracy over time. GBM/XGBoost is often more sensitive to detecting subtle patterns in the data, making it highly effective in uncovering complex fraud schemes.

In summary, while Random Forest offers a robust and straightforward approach to fraud detection, GBM/XGBoost provides a more nuanced and adaptive method, often resulting in higher accuracy at the cost of increased complexity and computational resources.

6.2 Understanding the FPR

Operating at a 5% false positive rate (FPR) means that approximately 5% of legitimate transactions may be flagged as fraudulent. This could lead to inconvenience for customers and impact the accuracy of fraud detection. Balancing precision (95% in this case) and recall is crucial to minimize false alarms while accurately identifying fraud.

Having a higher FPR would result in more legitimate transactions being incorrectly flagged as fraudulent. This could lead to customer dissatisfaction due to the inconvenience of dealing with false alarms. On the other hand, operating at a lower FPR would reduce the number of false alarms but might also result in some fraudulent transactions being missed, impacting the effectiveness of fraud detection. Achieving a balance is important to minimize both false alarms and missed detections.

A 5% FPR is often considered optimal in fraud detection because it strikes a balance between detecting fraudulent activity and minimizing false alarms. At this FPR, the model is efficient in identifying potentially fraudulent transactions while keeping false alarms relatively low. This balance is crucial to ensure that legitimate customers are not inconvenienced by unnecessary security checks or account freezes, while also maintaining a high level of fraud detection accuracy.

7 Appendix

7.1 Data Definitions

Variable	Description
EVENT_ID	Transaction Identifier
account_age_days	number of days since the account was created
transaction_amt	the USD $ value of the transaction
transaction_adj_amt	the adjustment USD $ value to the transaction
historic_velocity	measure of the historic USD $ amount used to purchase goods and services
ip_address	ip address of transactor
user_agent	user agent of the transactor
email_domain	email domain of the transactor
phone_number	phone number of the transactor
billing_city	billing city name
billing_postal	billing postal code
billing_state	billing state code
card_bin	first 6 digits of the credit card (determines the card type, issuing bank, debit/credit/prepaid)
currency	original currency code
cvv	Card Verification Value - the 3 digit number on back of your card
signature_image	boolean for whether or not signature image is recorded
transaction_type	code for the transaction type
transaction_env	code for the transaction environment
EVENT_TIMESTAMP	timestamp when the transaction occurred
applicant_name	name of the transactor - ignore
billing_address	billing address of the card holder
merchant_id	merchant identifier
locale	browser locale
tranaction_initiate	code for type of transaction initiation
days_since_last_logon	days since last transaction initiated
inital_amount	amount of first transaction USD $
EVENT_LABEL	TARGET fraud / legit
phone_number_len	number of digits in phone number
domain_type	denotes type of email domain (.com, .org, .biz, etc.)
month	month of transaction
postal_first_digit	first digit of billing postal code

Figure 5: Definition of Variables in Dataset

7.2 Feature Importances Top 10

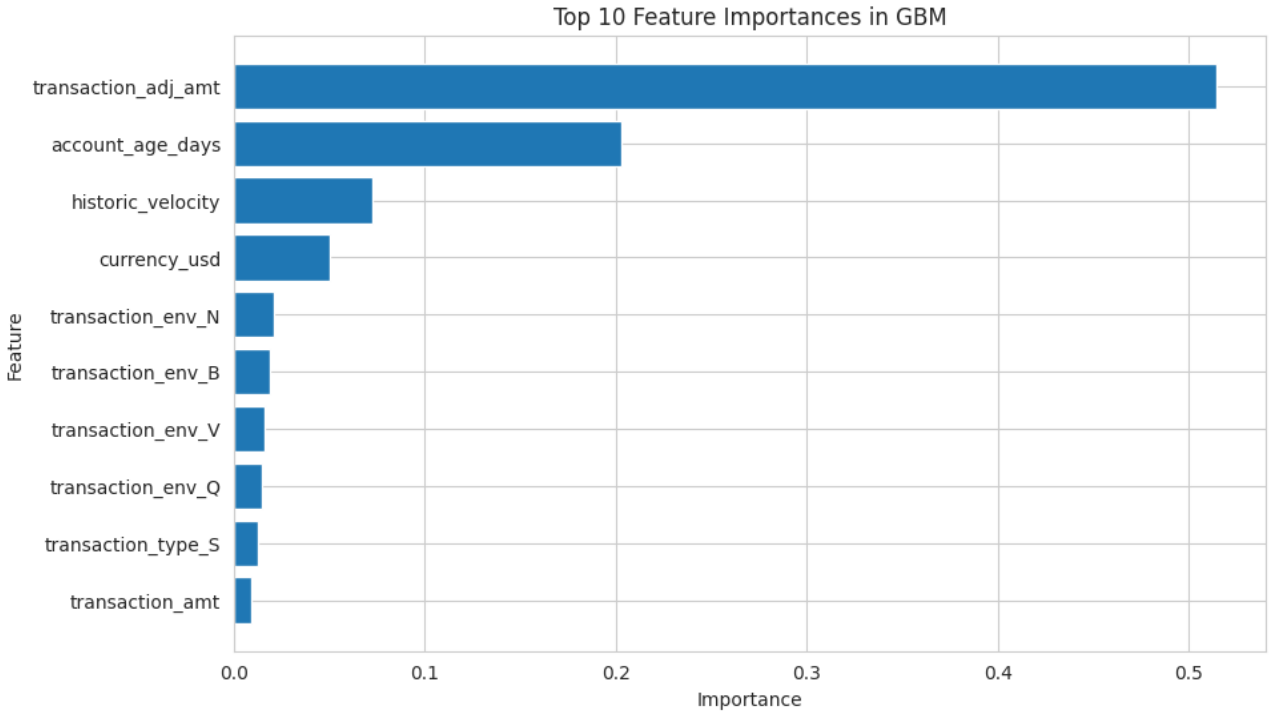

Figure 6: Adjusted Transaction Amount Fraud vs Legit