Project 2 Model – Finding Fraud

Finding Fraud Faster

Anthony Ayala

Executive Summary

Analysis

Recommendations

Executive Summary

Business Problem

The machine learning team at Bepo, a large financial institution, has initiated a project focused on identifying fraud within their payment stream. Missed fraudulent transactions undermine the institution's credibility and result in financial losses if detected after the fact. With the rise of data science and machine learning, the demand for fraud detection has increased significantly. However, alongside this growth, there has been an exponential increase in knowledge and computing power. My role is to leverage technical expertise to generate insights for accurately predicting fraud. This project involves analyzing a sample dataset of transactions, where each transaction is labeled as fraudulent or legitimate, and a holdout set for which predictions must be made based on unlabeled data. Additionally, we are tasked with implementing three classification models of increasing sophistication, with the objective of identifying the best-performing model and assessing its suitability for integration into Bepo's fraud detection system.

Methodology:

Predicting fraud with classification models follows a systematic process: feature selection, data splitting, preprocessing, model training, hyperparameter tuning, performance evaluation, and feature importance analysis. Three experiments were conducted in this project: one focusing solely on numeric features, another considering both numeric and categorical features, and a third incorporating Synthetic Minority Over-sampling Technique (SMOTE) to address imbalanced data. These experiments aim to highlight progress and differences, ultimately determining the most effective method for accurate predictions.

Recommendations:

Prior to making predictions, it is crucial to address data imbalance, as overlooking this issue can lead to biased results. Employing sampling techniques such as SMOTE improves accuracy and ensures that fraudulent transactions are adequately represented in the training data. After evaluating various models, the Random Forest algorithm emerged as the optimal choice, exhibiting impressive accuracy and area under the curve. Notably, it identified key features contributing to fraud, such as transaction initiation code, transaction environment code, card verification value, account age, and transaction amount. Understanding feature importance enables informed decision-making and enhances fraud detection capabilities. Moving forward, continual monitoring of model performance and periodic adjustments are recommended to maximize effectiveness and adapt to evolving trends in fraudulent activity.

METHODOLOGY

Data Exploration and Preprocessing

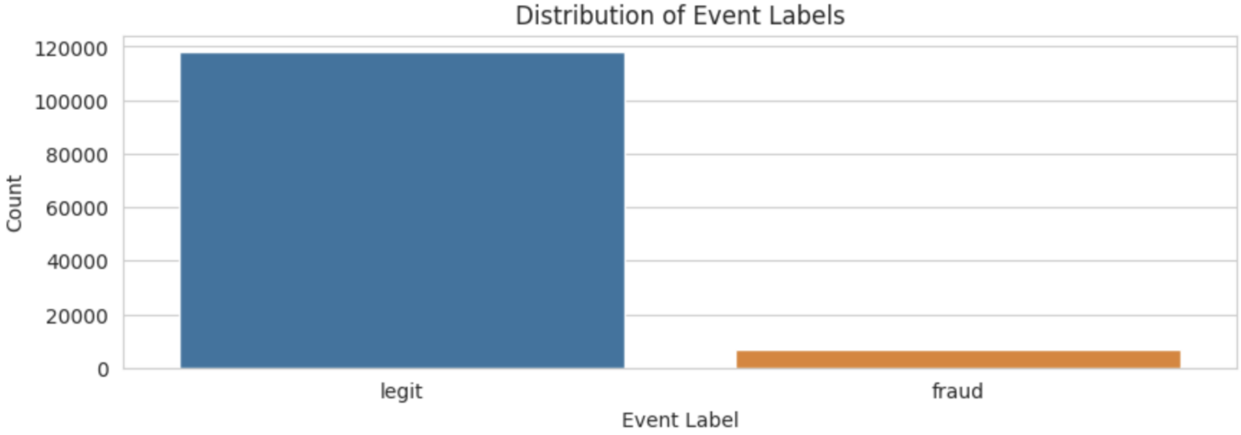
Predicting fraud within the financial institution involves analyzing customer activity, demographics, and transaction details. Typically, predictions are not based on demographics to avoid bias and focus on actionable variables. However, this assignment requires assessment of demographic data, specifically billing postal and email domains.

In the dataset, there are 118,215 rows labeled as legitimate transactions and 6,785 labeled as fraudulent. Analysis of these demographic variables reveals interesting insights. The top ten most common legitimate email domains occur 50 or more times, whereas the most fraudulent domains occur once or only a few times, indicating potential spam or scam activity contributing to fraudulent transactions.

Similarly, in billing postal codes, some areas have numerous legitimate transactions, while fraudulent postal codes have lower occurrence rates. These findings suggest certain postal codes may correlate with socioeconomic status, potentially introducing bias into the analysis.

Considering these insights, email domain alone may not be a reliable predictor of fraud, especially given the imbalance between legitimate and fraudulent observations. Exploring additional variables could yield more robust fraud detection methods.

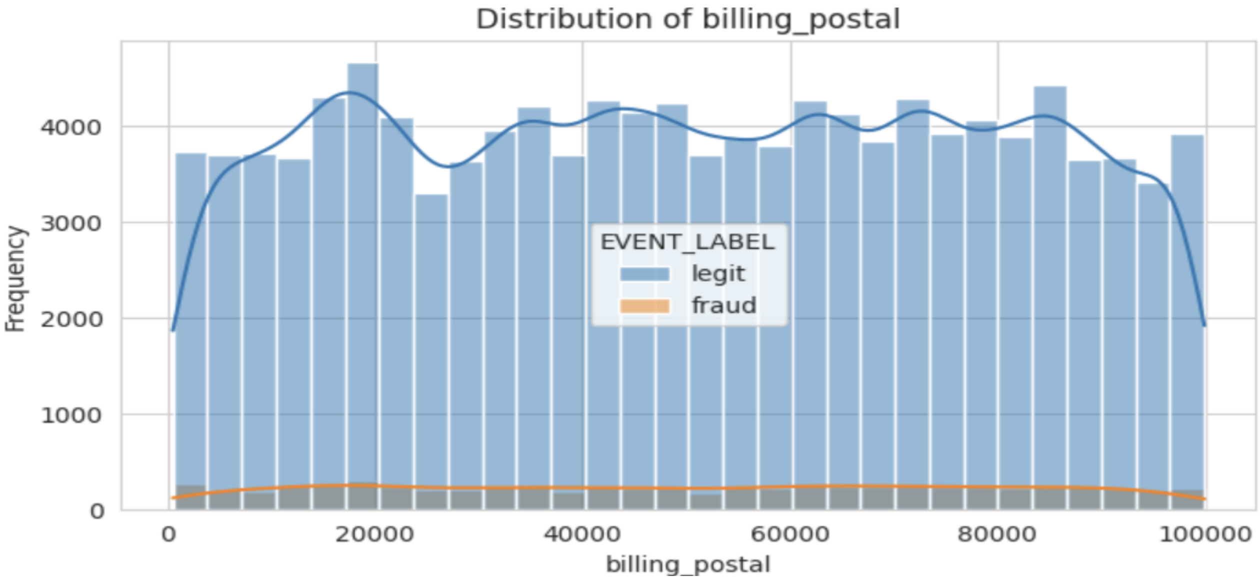
Imbalanced Classification Data



In exploratory data analysis, numerical data and transaction-related information emerge as crucial indicators for predicting fraud. For instance, transaction amount often serves as a key discriminator between legitimate and fraudulent transactions. Unusually high transaction amounts, particularly those exceeding the 75th percentile, are typically flagged for further scrutiny by managers or the fraud team.

Additionally, variables such as the duration since the account was opened, adjusted transaction amount, and historical transaction velocity prove to be actionable predictors. These factors offer objective insights, facilitating informed decision-making when establishing decision boundaries. Ultimately, these predictive variables contribute to delineating the typical characteristics associated with legitimate and fraudulent transactions.

Biased / Irrelevant Predictors



As stated before, billing postal is not a good predictor of fraud as the visual shows the sheer difference between fraud and legit transactions, and there is no movement for the fraud curve nor is there is a specific billing postal code that is alarming.

Top 5 Fraudulent Email Domains

Event Label	Email Domain	Count
Fraud	**Cruz.net**	**11**
Fraud	**Pham-stone.com**	**10**
Fraud	**Arias.biz**	**9**
Fraud	**Beck.biz**	**9**
Fraud	**Murphy-Sander.org**	**9**

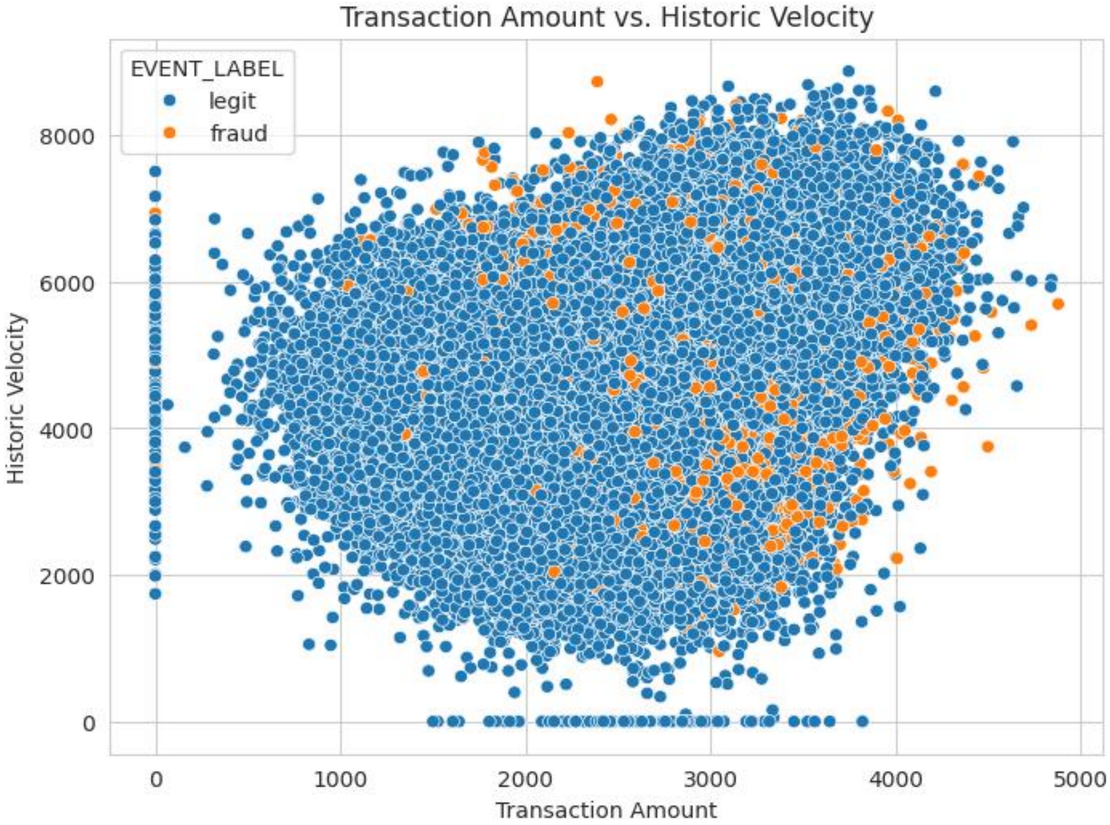
Top 5 Legitimate Email Domains

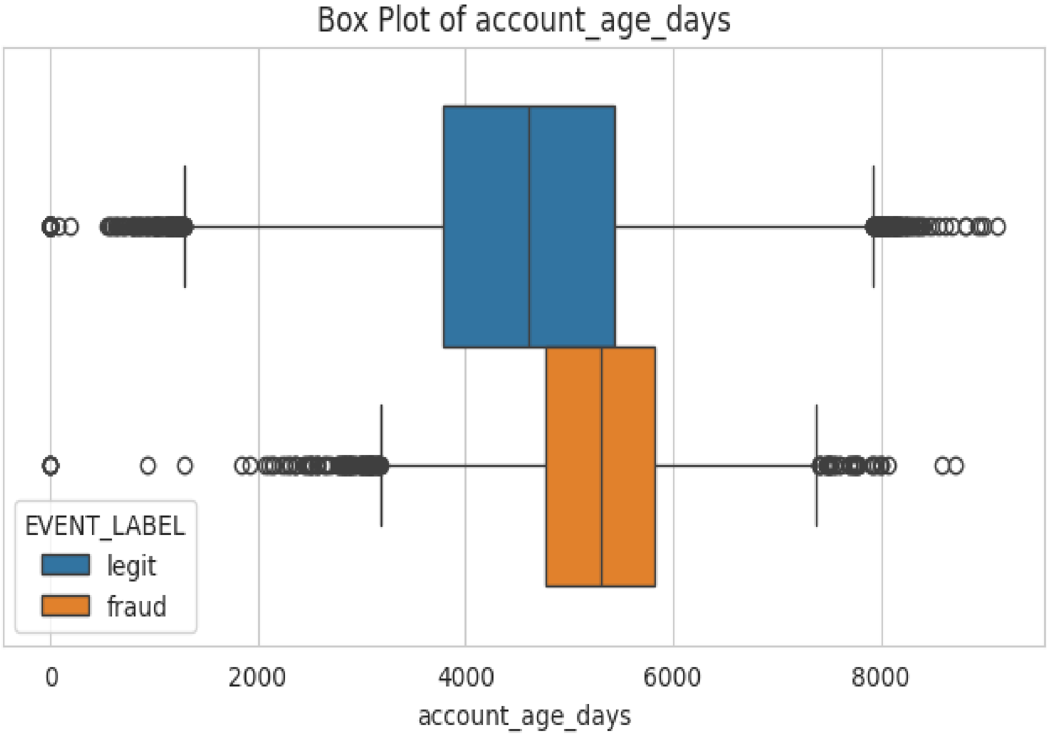
Event Label	Email Domain	Count
Legit	Freeman-adams.com	59
Legit	Cochran.biz	56
Legit	Lane.info	55
Legit	Boyle-murray.com	55
Legit	Solis.com	55

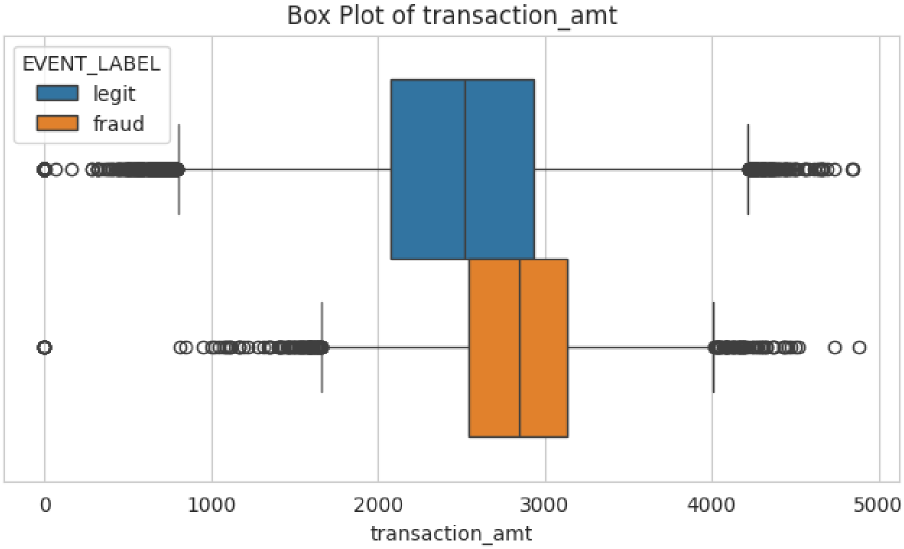
Feature Selection and Preprocessing

Feature Name	Data Type	Transformation
Account Age Days	Numeric	Standard Scaler: Mean is 0 and Standard Deviation is 1
Transaction Amount	Numeric	Standard Scaler: Mean is 0 and Standard Deviation is 1
Historic Velocity	Numeric	Standard Scaler: Mean is 0 and Standard Deviation is 1
Days Since Last Log on	Numeric	Standard Scaler: Mean is 0 and Standard Deviation is 1
Initial Amount	Numeric	Standard Scaler: Mean is 0 and Standard Deviation is 1

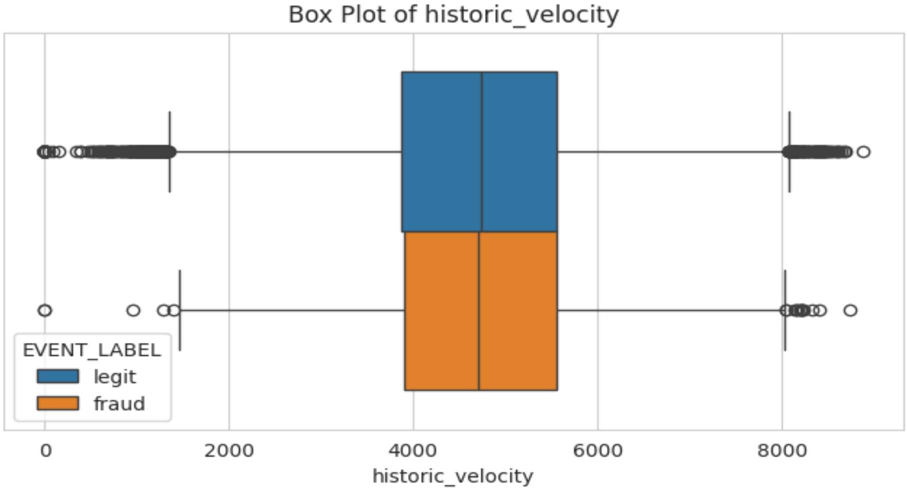
Feature Name	Data Type	Transformation
Billing State	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
Currency	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
CVV	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
Signature Image	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
Transaction Type	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
Transaction Environment	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
Locale	Categorical	OneHotEncoder: Numeric Array / Dummizying Features
Transaction Initiate	Categorical	OneHotEncoder: Numeric Array / Dummizying Features


Transaction Amount vs. Historic Velocity
EVENT_LABEL
legit
fraud
Historic Velocity
8000
6000
4000
2000
0
0
1000
2000
3000
4000
5000
Transaction Amount

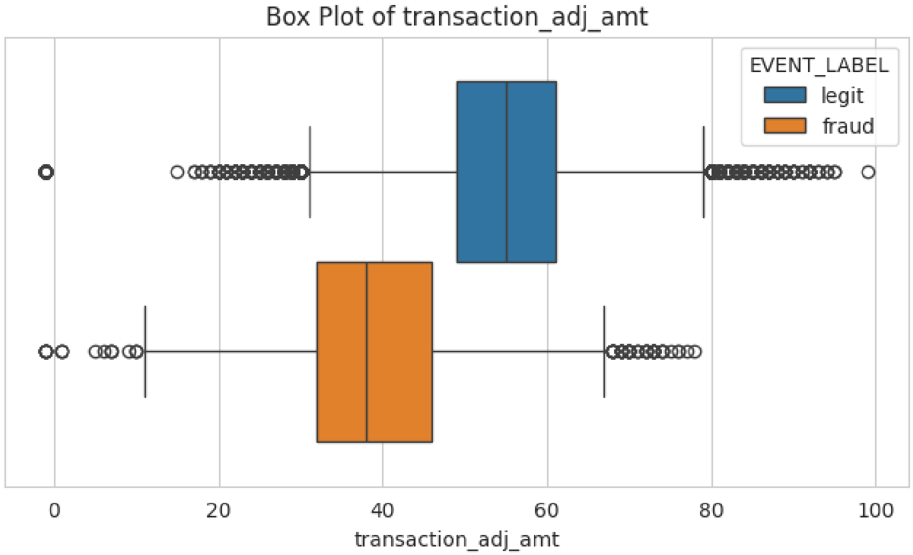

Box Plot of account_age_days
EVENT_LABEL
legit
fraud
0
2000
4000
6000
8000
account_age_days



Box Plot of transaction_amt



Box Plot of historic_velocity

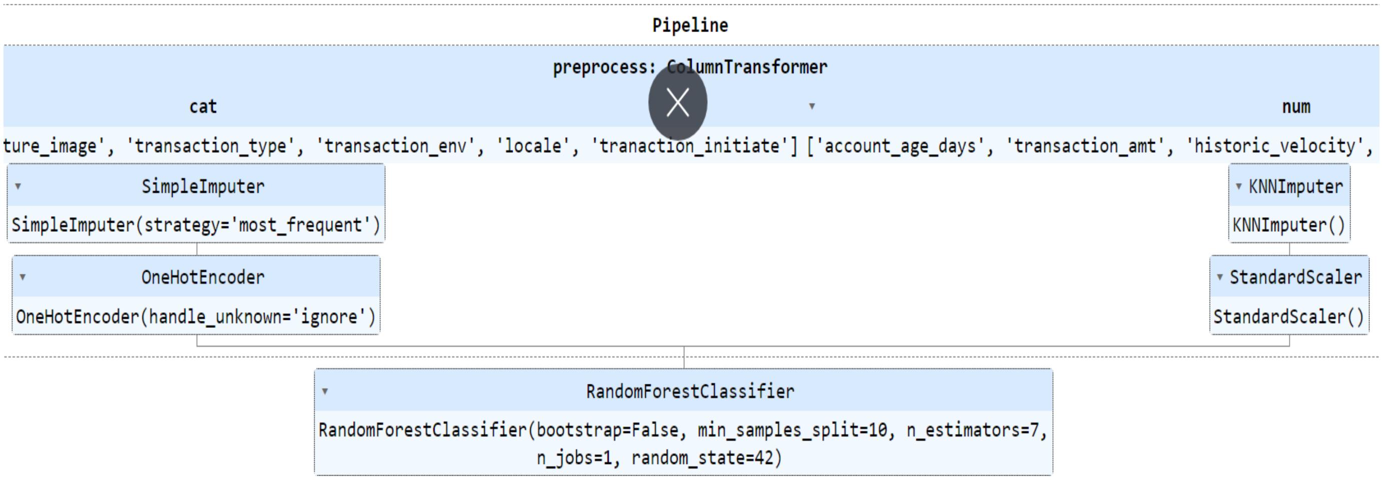


Box Plot of transaction_adj_amt

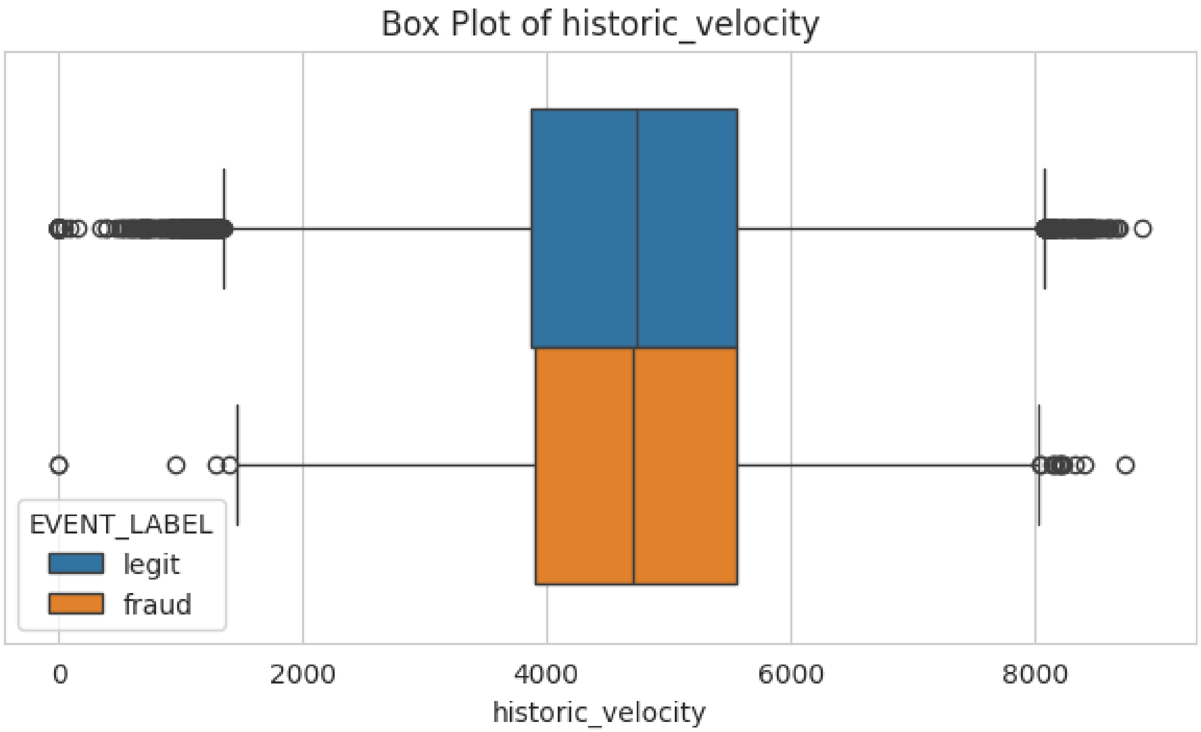
I selected the following features from the graphs as they seemed to show some slight difference between fraudulent and legitimate transactions, referring to the numeric variables. As for the categorical variables, these features hold information on the transaction which are actionable variables and can be useful to see if there is a combination or pattern with fraudulent transactions.

After mitigating bias and conducting additional analysis, I opted to build my models using a reduced set of variables. Our dataset is manageable, with a manageable number of unique values that do not hinder prediction, nor is it excessively large to handle. Like any proficient machine learning specialist, I adhere to the standard procedure outlined in the machine learning recipe. This involves preprocessing the data to address null values through imputation and scaling, followed by model fitting on training data. Ultimately, predictions are made on the test data, and the model's performance is assessed. To streamline this process, pipelines are employed to ensure consistency and efficiency.

Pipeline: Preprocess Data and Fit Train and Test Data with a Random Forest Classifier



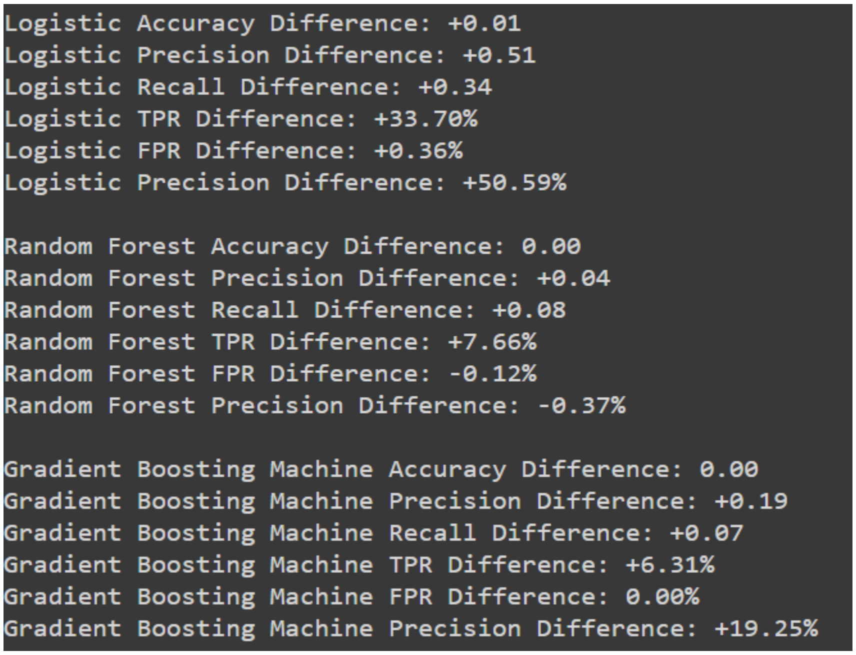
```
Pipeline
preprocess: ColumnTransformer
cat                                                          num
ture_image', 'transaction_type', 'transaction_env', 'locale', 'tranaction_initiate'] ['account_age_days', 'transaction_amt', 'historic_velocity',
SimpleImputer                                                KNNImputer
SimpleImputer(strategy='most_frequent')                      KNNImputer()
OneHotEncoder                                                StandardScaler
OneHotEncoder(handle_unknown='ignore')                       StandardScaler()
RandomForestClassifier
RandomForestClassifier(bootstrap=False, min_samples_split=10, n_estimators=7,
n_jobs=1, random_state=42)
```



Model Development

For classification handling, we chose to implement three types of models, ranging from less advanced to highly advanced. Logistic regression serves as a baseline for performance comparison, while random forest acts as an ensemble method known for its robustness in handling complex data structures. Gradient boosting machine represents an advanced ensemble technique recognized for its predictive power. The ensemble approach aims to enhance accuracy and resilience in forecasting by combining predictions from multiple models.

Growth from predicting with just numeric to numeric and categorical:

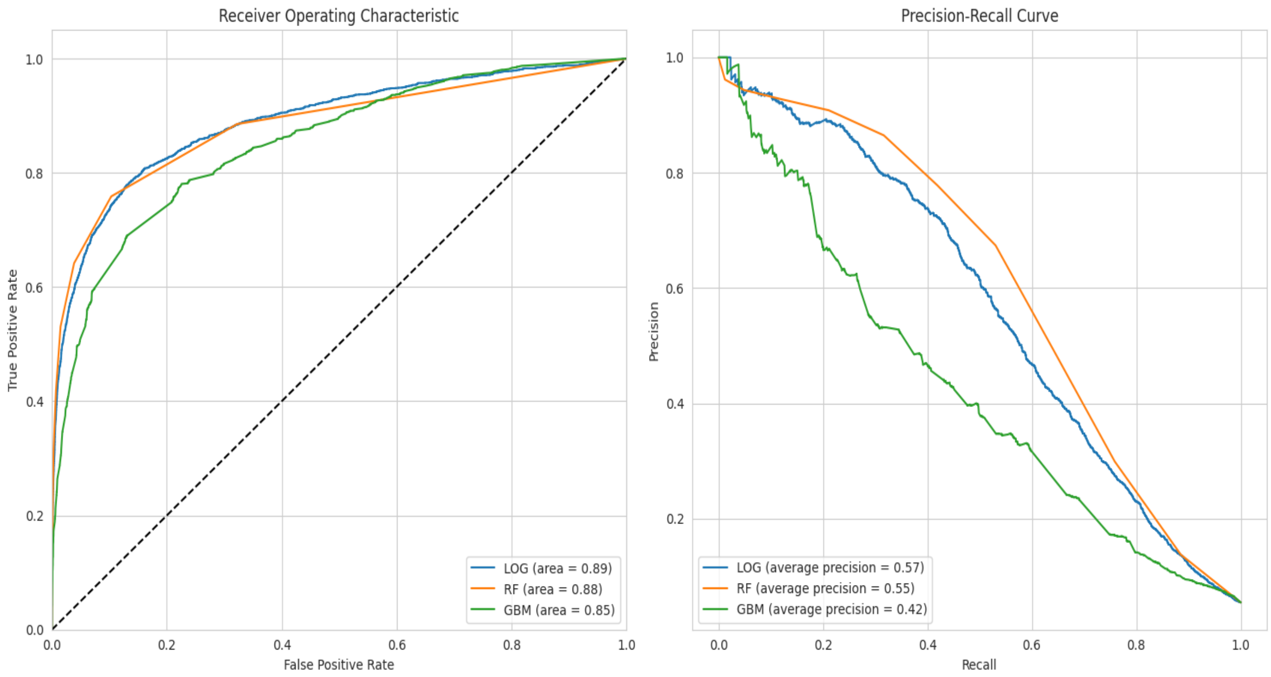


```
Logistic Accuracy Difference: +0.01
Logistic Precision Difference: +0.51
Logistic Recall Difference: +0.34
Logistic TPR Difference: +33.70%
Logistic FPR Difference: +0.36%
Logistic Precision Difference: +50.59%

Random Forest Accuracy Difference: 0.00
Random Forest Precision Difference: +0.04
Random Forest Recall Difference: +0.08
Random Forest TPR Difference: +7.66%
Random Forest FPR Difference: -0.12%
Random Forest Precision Difference: -0.37%

Gradient Boosting Machine Accuracy Difference: 0.00
Gradient Boosting Machine Precision Difference: +0.19
Gradient Boosting Machine Recall Difference: +0.07
Gradient Boosting Machine TPR Difference: +6.31%
Gradient Boosting Machine FPR Difference: 0.00%
Gradient Boosting Machine Precision Difference: +19.25%
```

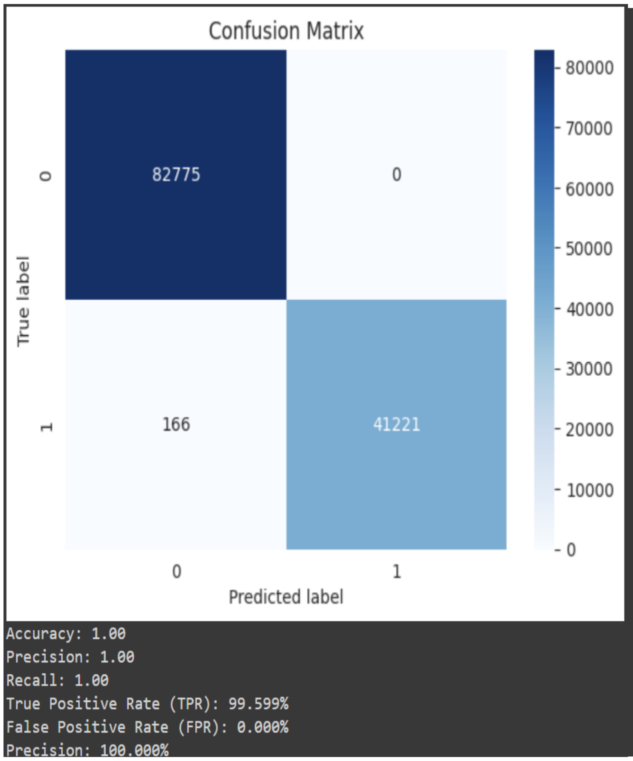
Tuning these models for improved performance is crucial, and employing a guess-and-check approach is impractical. Hyper parameterization is the preferred method, involving the creation of parameter combinations and utilizing functions such as random search or grid search to identify the optimal parameter values that yield the highest accuracy or area under the ROC curve. In this case, a random grid search was employed for efficiency, particularly effective when dealing with a large parameter space.

In this scenario, attention is directed towards the AUC (Area Under the Curve) metric, considered superior to accuracy in imbalanced datasets, such as ours where legitimate transactions significantly outnumber fraudulent ones. The focus lies on the True Positive Rate (TPR) and False Positive Rate (FPR), where a higher AUC indicates better discrimination between fraudulent and legitimate customers. TPR signifies the proportion of correctly predicted fraudulent customers among all actual fraudulent customers, while FPR indicates the proportion of incorrectly predicted legitimate customers among all actual legitimate customers.

<u>SMOTE Performance:</u>

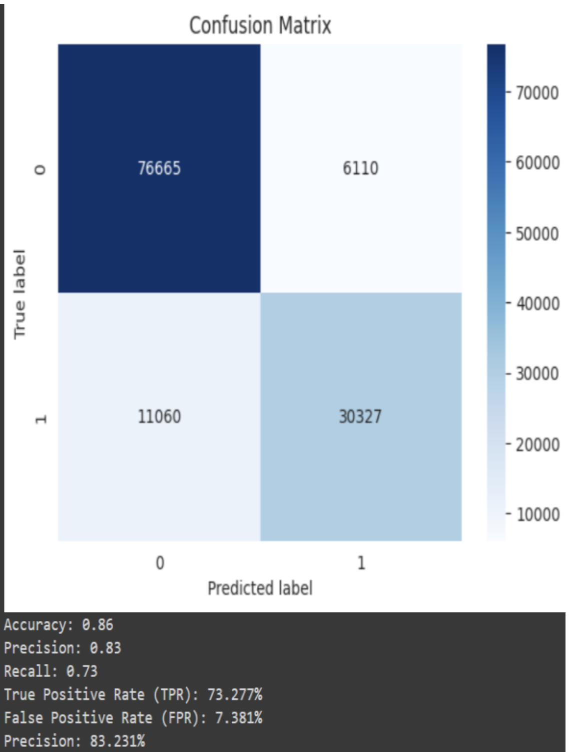


Random Forest (Left) & Logistic Regression (Right)



	0	1
0	82775	0
1	166	41221

```
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
True Positive Rate (TPR): 99.599%
False Positive Rate (FPR): 0.000%
Precision: 100.000%
```



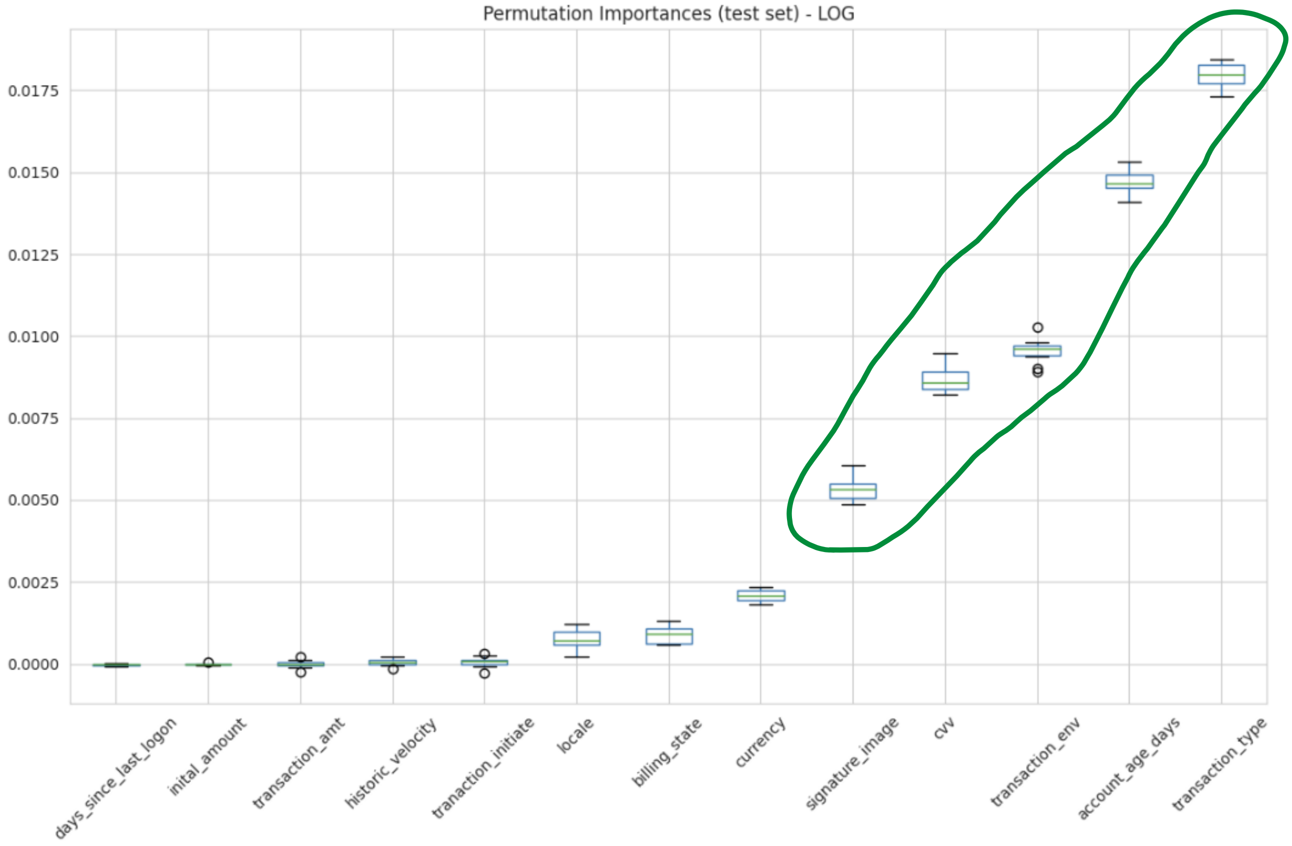
	0	1
0	76665	6110
1	11060	30327

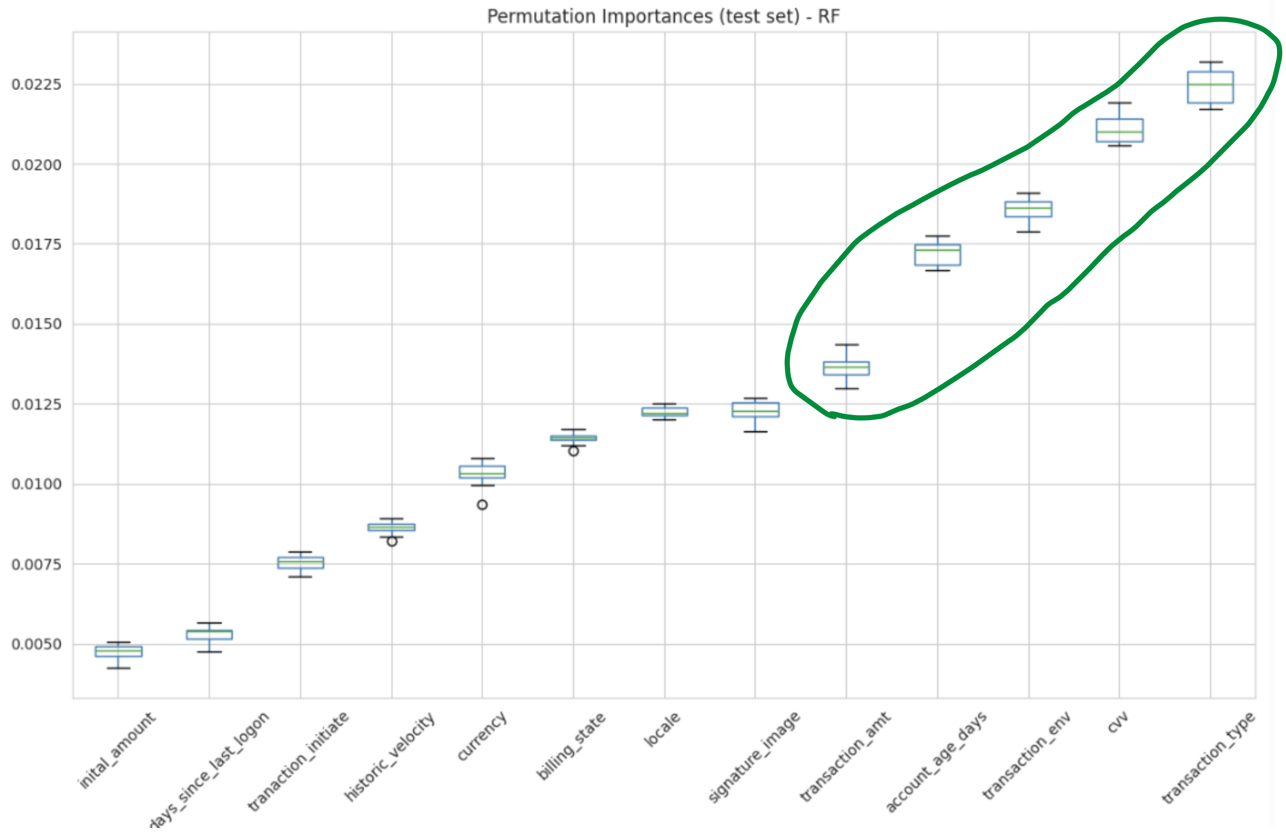
```
Accuracy: 0.86
Precision: 0.83
Recall: 0.73
True Positive Rate (TPR): 73.277%
False Positive Rate (FPR): 7.381%
Precision: 83.231%
```

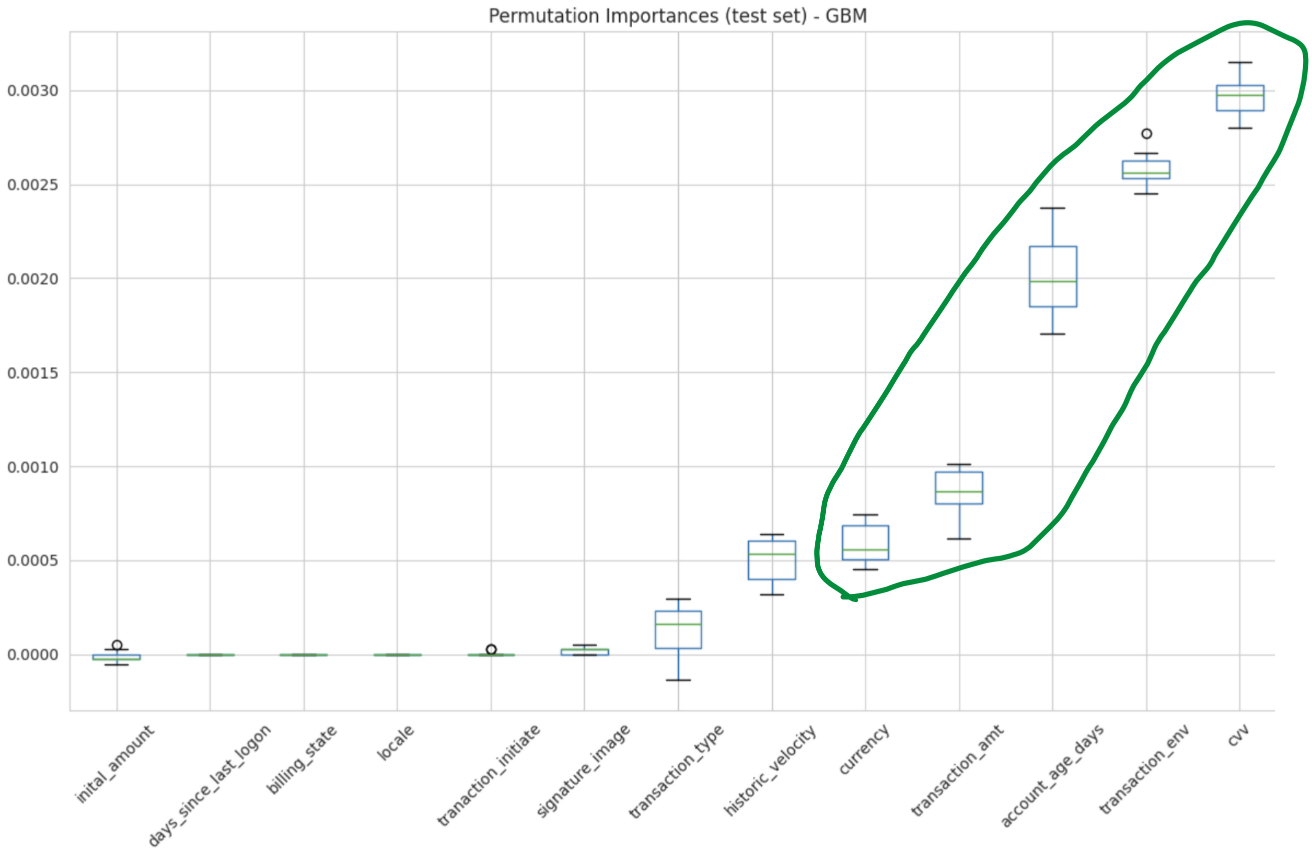
Following three iterations of the cookbook recipes, two types of analyses were utilized for feature selection: permutation importance and feature coefficients. Permutation importance measures the impact of a variable by systematically shifting its values and observing the resulting change in model scoring. Feature coefficients, on the other hand, are akin to linear relationships with the target variable.

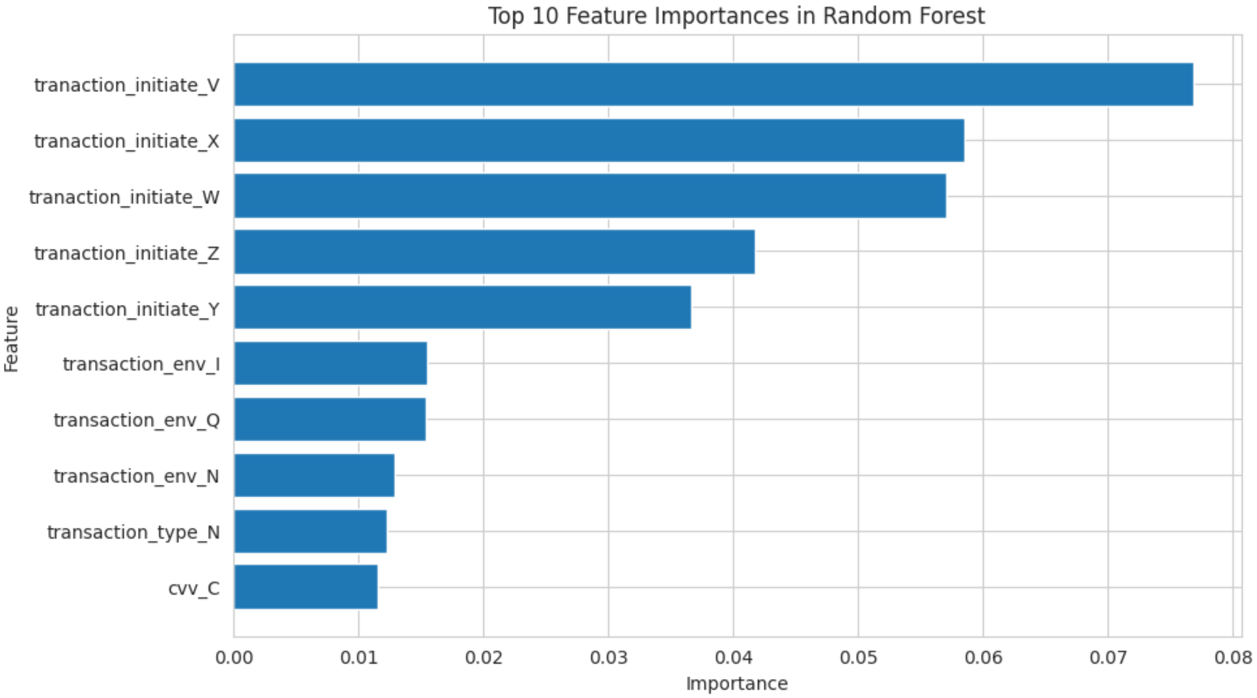
Across the board, permutation importance identified the top five variables as transaction type, number of days since the account creation, transaction environment, transaction account, and card verification value. Meanwhile, feature coefficients highlighted transaction initiation, transaction environment, card verification value, and billing state as the most influential variables.

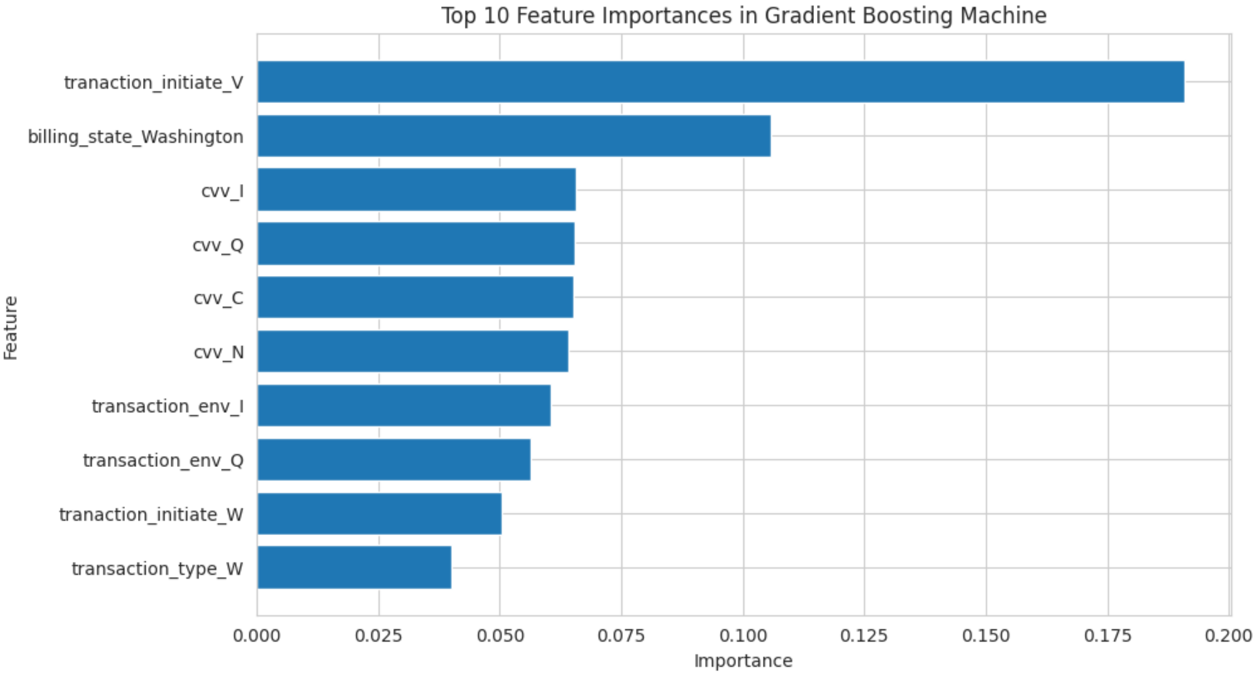
Feature Importance:




Permutation Importances (test set) - RF
0.0225
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
inital_amount
days_since_last_logon
tranaction_initiate
historic_velocity
currency
billing_state
locale
signature_image
transaction_amt
account_age_days
transaction_env
cvv
transaction_type


Permutation Importances (test set) - GBM
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
inital_amount
days_since_last_logon
billing_state
locale
tranaction_initiate
signature_image
transaction_type
historic_velocity
currency
transaction_amt
account_age_days
transaction_env
cvv


Top 10 Feature Importances in Random Forest
tranaction_initiate_V
tranaction_initiate_X
tranaction_initiate_W
tranaction_initiate_Z
tranaction_initiate_Y
transaction_env_I
transaction_env_Q
transaction_env_N
transaction_type_N
cvv_C
Feature
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Importance


Top 10 Feature Importances in Gradient Boosting Machine
tranaction_initiate_V
billing_state_Washington
cvv_I
cvv_Q
cvv_C
cvv_N
transaction_env_I
transaction_env_Q
tranaction_initiate_W
transaction_type_W
Feature
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Importance

Model Evaluation

My approach to model evaluation involved conducting several iterations of the cookbook recipe, each comprising predictions and training on both training and test datasets. Through these iterations, I gained valuable insights and refined my methodology for model performance.

Initially, my focus was primarily on accuracy as a metric for evaluating model performance. However, as I delved deeper into the business problem, I realized that accuracy alone might not adequately address the nuances of fraud detection. Therefore, I shifted my focus to the Area Under the Curve (AUC) metric.

By prioritizing AUC, I aimed to optimize the model's performance in correctly classifying legitimate transactions as 0s and fraudulent transactions as 1s. This shift in focus allowed me to better align the model's predictions with the objectives of the business problem, ultimately enhancing the accuracy and effectiveness of fraud detection efforts.

Performance Metrics

Model	ROC AUC	Precision	Recall	F1 Score
1st Logistic	**0.70**	**0**	**0**	**0**
1st Random Forest	**0.91**	**0.96**	**0.59**	**0.73**
1st Gradient Boost Machine	**0.79**	**1.00**	**0**	**0.01**
2nd Logistic	**0.87**	**0.84**	**0.34**	**0.484**
2nd Random Forest	**0.83**	**1.00**	**0.66**	**0.796**
2nd Gradient Boosting Machine	**0.82**	**0.99**	**0.07**	**0.131**
3rd Logistic (Smote)	**0.89**	**0.83**	**0.73**	**0.776**

3rd Random Forest (Smote)	**0.88**	**1.00**	**1.00**	**1.00**
3rd Gradient Boosting Machine (Smote)	**0.85**	**0.97**	**0.50**	**0.659**

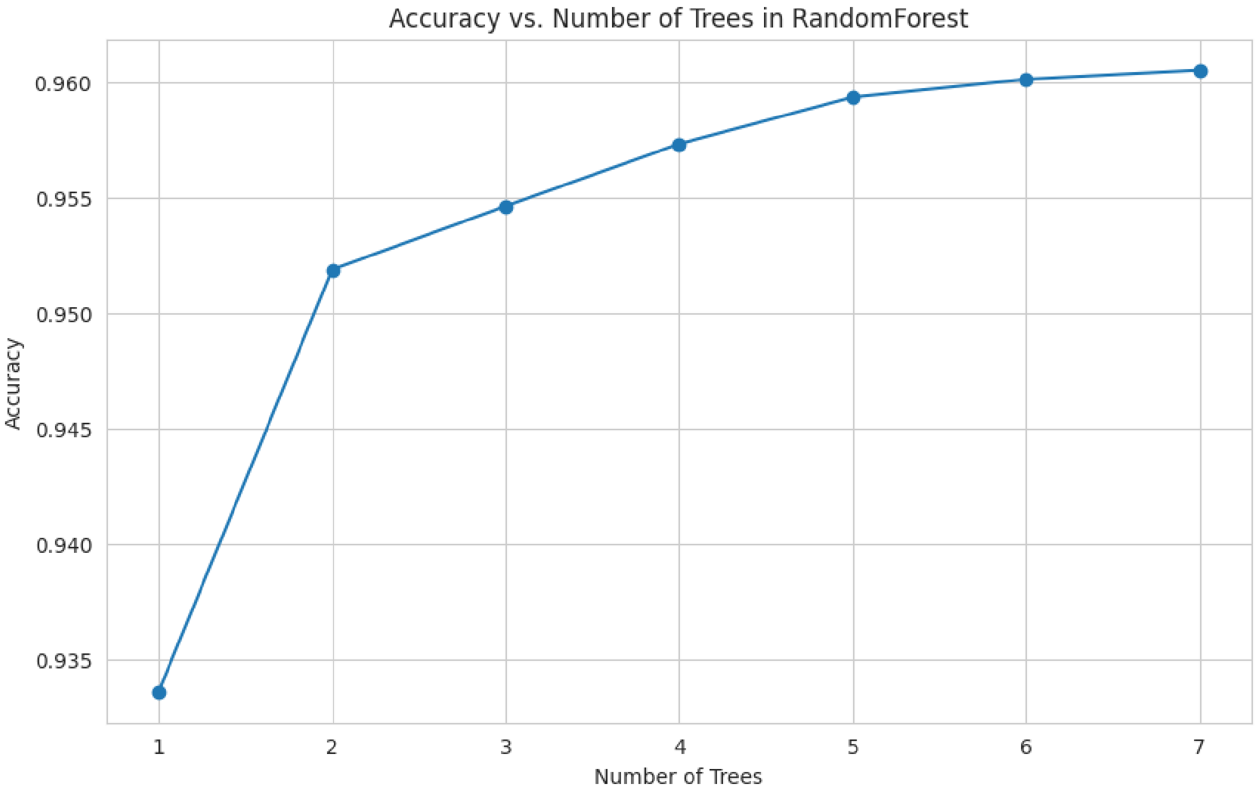
Model improvements with Hyper-parameterization

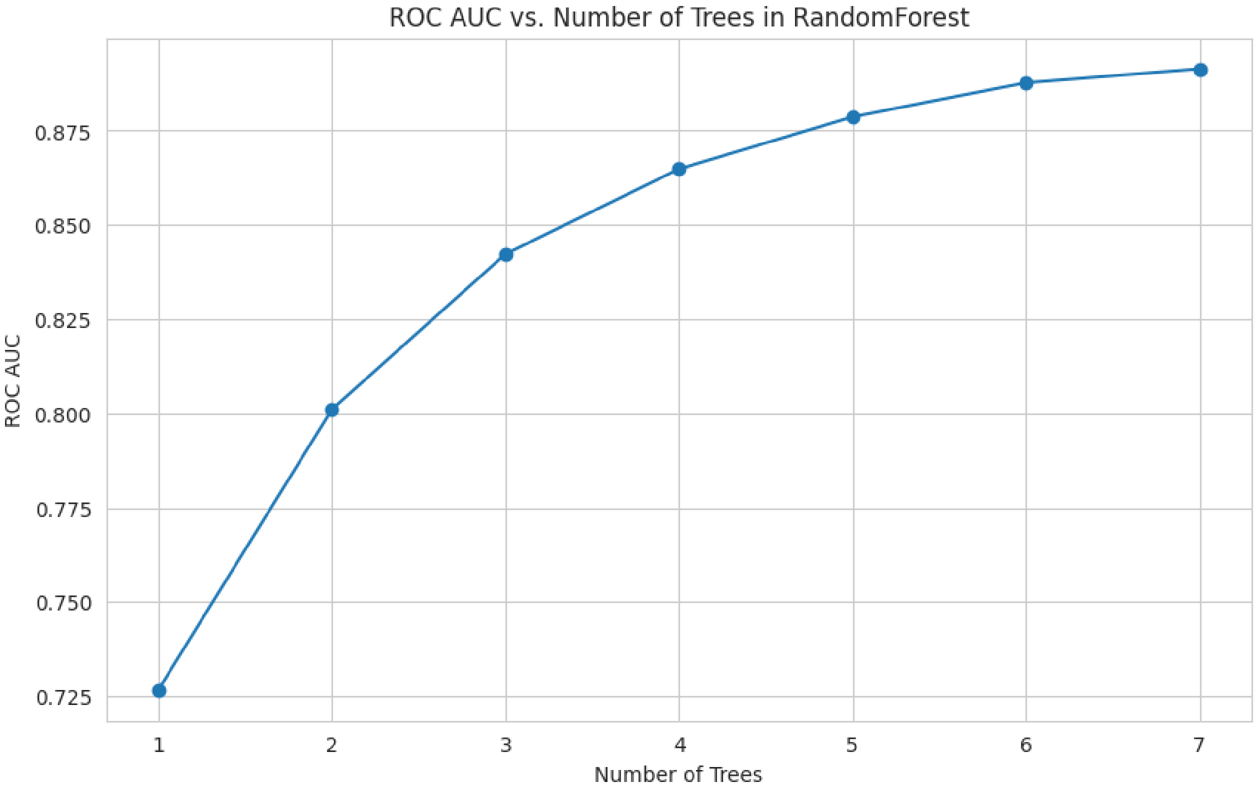
After obtaining the optimal parameters through hyper-parameterization, I was curious to explore whether further improvements could be achieved or to understand why those specific parameter combinations were deemed optimal. To investigate this, I conducted experiments by slightly adjusting one parameter at a time and observing the resulting changes in model performance.

For instance, what you will see down below is I sought to determine if the accuracy and AUC of the best random forest model would plateau at a certain point, indicating diminishing returns in accuracy gains. This approach allowed me to gauge the sensitivity of the model's performance to variations in specific parameters and provided insights into the limits of parameter optimization.

By systematically sampling parameter variations and assessing their impact on model performance, I gained a deeper understanding of the nuances involved in optimizing the model. This iterative process not only helped refine the model's parameters but also provided valuable insights into the underlying dynamics of the data and the model's behavior. Ultimately, it enabled me to make more informed decisions regarding parameter selection and model optimization, contributing to the overall effectiveness of the fraud detection system.

Hyper-parameterizing Random Forest and reaching limits





Final Fit on Holdout Set

Random Forest ROC Curve	**0.92**

Model FPR/TPR/Threshold Table

Target False Positive Rate	True Positive Rate (TPR)	Prob Threshold
1%	99.90%	0.2342
2%	99.96%	0.1935
3%	99.97%	0.1825
4%	99.97%	0.1715
5%	99.97%	0.1605
6%	99.98%	0.1496
7%	99.98%	0.1386
8%	99.99%	0.1276
9%	99.99%	0.1166
10%	100%	0.1056

Operational Strategy at 5% FPR

With a strategy like this, our threshold for predicting fraud is set to be greater than 0.1615, ensuring that we capture 99.97% of all actual frauds while allowing for a 5% false positive rate (FPR), where legitimate transactions are incorrectly classified as fraud. This approach prioritizes the detection of fraudulent activity while striving to minimize errors in identifying legitimate transactions.

Maintaining a low false positive rate is crucial for preserving customer experience and trust. A 5% false positive rate ensures that only a small fraction of legitimate transactions are flagged as potentially fraudulent, reducing inconvenience for customers and minimizing disruptions to their transactions. This balance between fraud detection and customer experience is essential for maintaining customer satisfaction and loyalty.

However, it's important to recognize that there's flexibility in setting the target false positive rate. By increasing the FPR target rate to 10%, we can achieve a 100% true positive rate (TPR), meaning that all fraudulent transactions are correctly classified. Nevertheless, this adjustment comes with the trade-off of misclassifying a higher proportion of legitimate transactions as fraud.

Determining the optimal target false positive rate involves weighing the priorities of fraud detection efficacy and customer experience. While a higher false positive rate may enhance fraud detection accuracy, it could potentially lead to increased customer dissatisfaction and transaction disruptions, referring to misclassification of legitimate transactions. Therefore, striking the right balance is essential in optimizing the effectiveness of the fraud detection system.

In conclusion, the best random forest model has significantly improved fraud detection capabilities, but careful consideration of the false positive rate target is necessary to ensure both effective fraud detection and a positive customer experience.

Insights and Recommendations

The random forest emerges as the top-performing model due to its ability to mitigate overfitting and deliver highly accurate predictions. Its superiority is evidenced by its substantial growth in the ROC curve, high accuracy rates, and exceptional performance on the holdout set. However, it's crucial to recognize that no single learning algorithm is universally superior to others, as highlighted by the No Free Lunch theorem.

One notable distinction between random forest and gradient boosting machine lies in their approaches to error correction and tree building. The gradient boosting machine tends to fit on noise, which can result in a higher risk of overfitting to the training data. In contrast, random forest is inclined towards generalization across various features, contributing to its robustness in handling different data scenarios.

The effectiveness of random forest, particularly in handling categorical data, underscores its versatility compared to other models like logistic regression. Random forest excels in scenarios where categorical data is predominant or where a combination of both categorical and numeric features exists, further solidifying its status as a preferred choice for diverse datasets.

While gradient boosting machines offer their own advantages such having parameters like a learning rate and flexibility in model variations. Random forest's ability to generalize across features and its strong performance with categorical data make it a compelling option for many classification tasks. This nuanced understanding of the strengths and weaknesses of each model is essential for selecting the most suitable approach based on the specific characteristics of the dataset and the objectives of the analysis.