UNSUPERVISED LEARNING

Report

ANOMALY DETECTION IN BANK TRANSACTIONS

PREPARED BY :
Sehara Sooriarachchi

Abstract

The detection of fraudulent transactions is a critical challenge in the banking sector, with financial institutions facing increasing threats from sophisticated fraud schemes. Traditional supervised learning approaches require labeled data, which can be scarce or outdated due to the evolving nature of fraud. This project explores the application of unsupervised learning techniques to detect anomalous and potentially fraudulent bank transactions without relying on predefined labels. By leveraging clustering algorithms such as K-Means and DBSCAN, along with exploratory data analysis, transaction patterns are identified, and deviations from typical behavior are flagged as suspicious. The study demonstrates how several variables can be utilized to uncover fraudulent activity, offering a scalable and adaptable solution for early fraud detection.

Introduction

Bank fraud has emerged as a growing concern in the digital economy, causing substantial losses and undermining customer trust. As transaction volumes increase and fraud tactics evolve, conventional rule-based systems and supervised machine learning models often fall short due to their dependence on labeled data and predefined rules. This highlights the need for more flexible and adaptive methods capable of identifying new and previously unseen fraudulent behaviors.

Unsupervised learning offers a powerful alternative, allowing systems to discover hidden patterns and anomalies within transaction data without requiring labeled outcomes. Techniques such as clustering and anomaly detection help isolate outliers that may signify fraudulent transactions. In this project, real-world transaction data is utilized to apply unsupervised learning methods, analyze customer behavior, and flag irregularities that deviate from typical transaction patterns. This study aims to contribute to the development of automated fraud detection systems that are both accurate and resilient to emerging threats.

Problem Statement

Financial institutions are constantly targeted by fraudulent activities that result in significant economic losses and reputational damage. Traditional fraud detection systems rely heavily on labeled data and predefined rules, which limits their effectiveness against novel and adaptive fraud tactics. There is a pressing need for scalable and intelligent systems that can detect fraud dynamically, without relying on historical fraud labels.

This project addresses the following problem:

> How can unsupervised machine learning techniques be utilized to detect fraudulent bank transactions in the absence of labeled data?

About the Dataset

This dataset is a detailed look into a bank's transactional behavior and financial activity patterns. It contains 2,512 samples of transaction data. It includes the following columns:

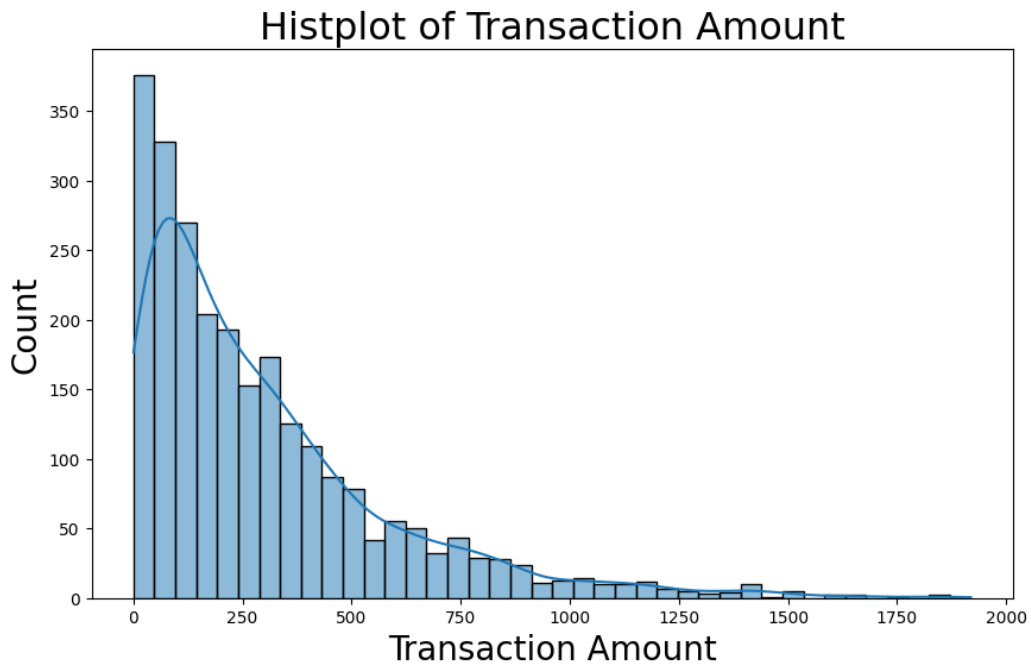
	Variable name	Variable Type	Description
1	TransactionID	Categorical	Unique identifier of each transaction
2	AccountID	Categorical	Unique identifier of each account, with multiple transactions per account
3	TransactionAmount	Integer	The monetary value of each transaction
4	TransactionDate	Date/Time	Timestamp of each transaction, capturing date and time
5	TransactionType	Categorical	Categorical field indicating 'Credit' or 'Debit' transactions
6	Location	Categorical	Geographic location of the transaction (US city names)
7	DeviceID	Categorical	An identifier for devices used to perform the transaction
8	IP Address	Categorical	The IPv4 address associated with the transaction
9	MerchantID	Categorical	Unique identifier for merchants
10	AccountBalance	Integer	Balance in the account post-transaction
11	PreviousTransactionDate	Date/Time	Timestamp of the last transaction for the account
12	Channel	Categorical	Channel through which the transaction was performed (Online/ATM/Branch)
13	CustomerAge	Integer	Age of the account holder
14	CustomerOccupation	Categorical	Occupation of the account holder (Doctor/Engineer/Student/Retired)
15	TransactionDuration	Integer	Duration of the transactions in seconds
16	LoginAttempts	Integer	Number of login attempts before the transaction

Dataset link: https://www.kaggle.com/datasets/valakhorasani/bank-transaction-dataset-for-fraud-detection/data

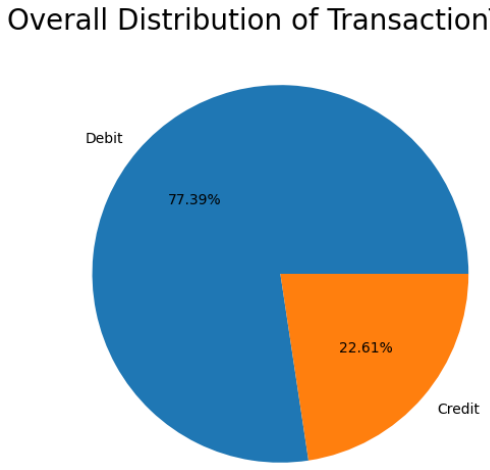
Pre-processing

- Reviewed the dataset for any duplicates and missing values. There were no missing values and no duplicate entries found.
- Two new variables named 'DayOfWeek' and 'Hour' is created by extracting day names and hour from 'TransactionDate' respectively.

Important Results of the Descriptive Analysis

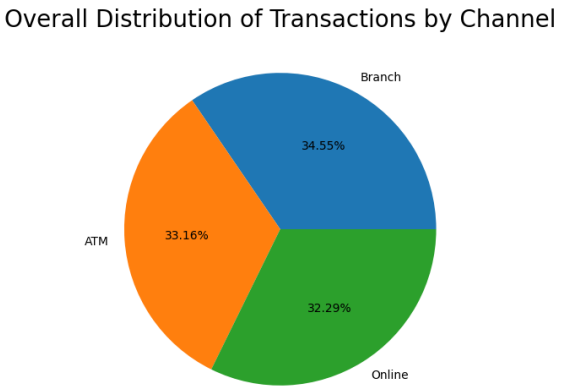


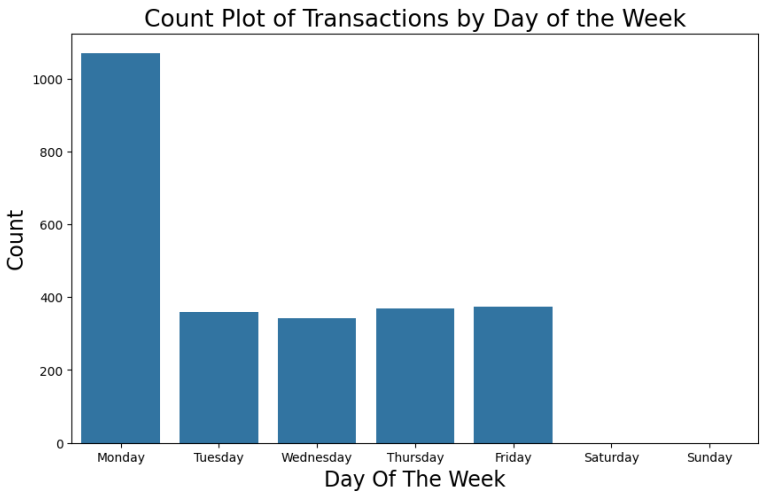
The above plot displays the distribution of amounts within the dataset, which reveals a right-skewed distribution, where the majority of transactions are concentrated in the lower range, particularly below 500. This indicates that low-value transactions are far more common. The frequency of high-value transactions decreases sharply as the amount increases. Such a pattern is typical in financial transaction data, where high-value transactions are relatively rare compared to low-value ones.



This pie chart illustrates the proportion of debit and credit transactions in the dataset. The chart reveals that the majority of transactions are debit transactions, accounting for approximately 77.89% of the total. In contrast, credit transactions make up only 22.61%. This significant imbalance suggests that users predominantly engage in debit transactions, which may include payments for goods and services, bill settlements, or ATM withdrawals. Credit transactions, being fewer, may represent inflows such as refunds, salary deposits, or transfers from other accounts.

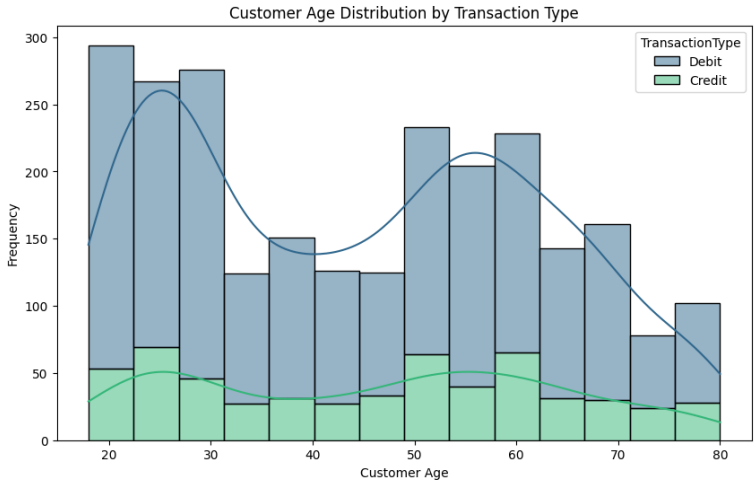
This pie chart provides an overview of how transactions are conducted across different banking channels: Branch, ATM, and Online. The chart shows a relatively balanced distribution among the three, with branch transactions leading slightly at 34.55%, followed closely by ATM transactions at 33.16%, and online transactions at 32.99%. This suggests that customers utilize all three channels fairly evenly, indicating a diverse range of banking preferences.

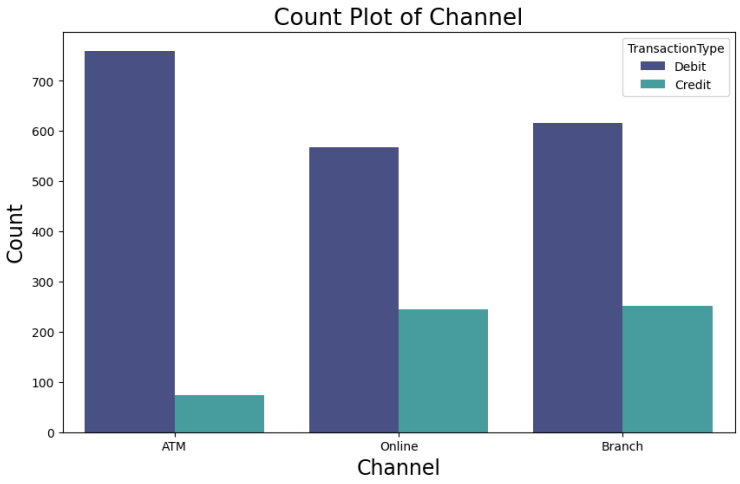




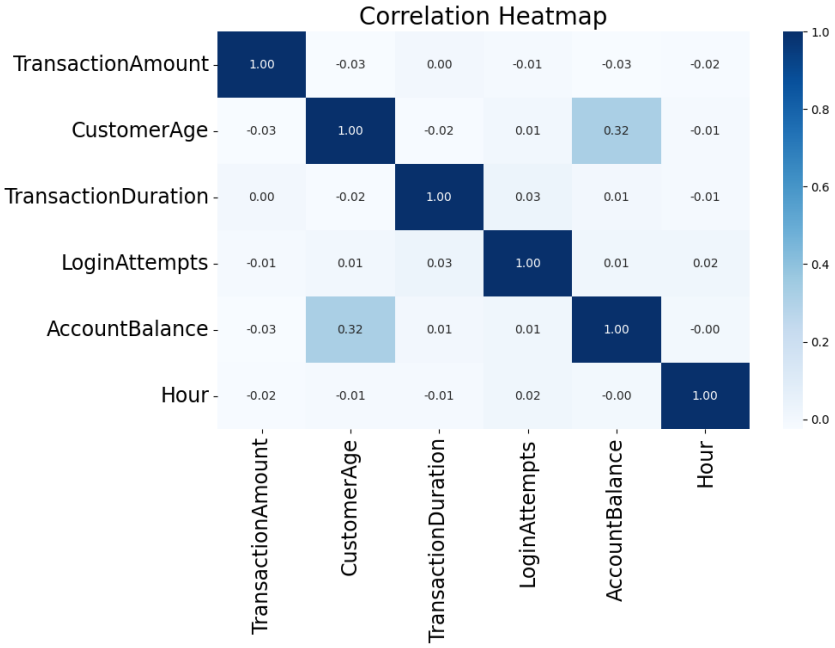
This bar plot provides a visual representation of the volume of bank transactions occurring each day. The data reveals a significant spike in transactions on Monday, far surpassing other days of the week, with over 1,000 transactions recorded. From Tuesday to Friday, the transaction volumes remain relatively consistent, each day accounting for around 350 to 380 transactions. Saturday and Sunday show no recorded transactions.

This histogram illustrates the frequency of debit and credit transactions across various age groups. The debit transactions dominate across all age brackets, with notable peaks among younger adults aged 18-30 and another surge among individuals aged 50-65. Credit transactions, while consistently lower in frequency, follow a similar age distribution trend.

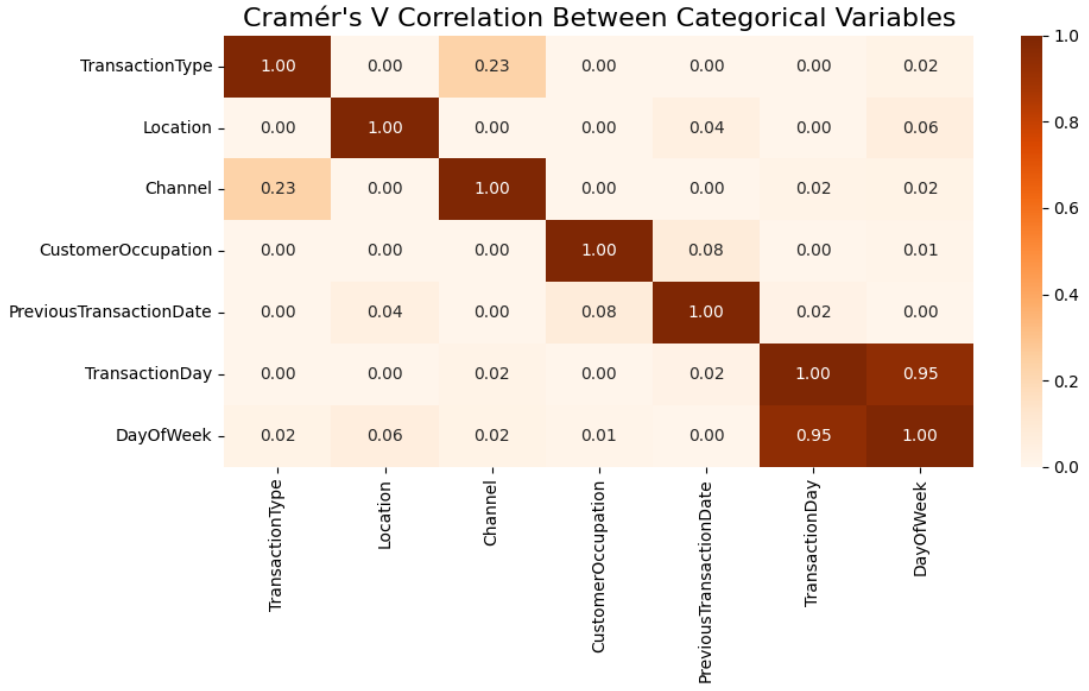




This count plot provides insights into the distribution of debit and credit transactions across three primary banking channels: ATM, Online, and Branch. It is evident that debit transactions dominate across all channels, with ATMs showing the highest volume of debit activity, followed by branches and online platforms. In contrast, credit transactions are relatively lower overall, though they are more evenly distributed between online and branch channels.



This matrix displays the relationships between numerical variables: ‘TransactionAmount’, ‘CustomerAge’, ‘TransactionDuration’, ‘LoginAttempts’, ‘AccountBalance’, and ‘Hour’. The values range from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 means no correlation. For example, ‘AccountBalance’ and ‘CustomerAge’ have a positive correlation of 0.32, indicating a weak positive relationship. Meanwhile, ‘TransactionAmount’ and ‘Hour’ show a negative correlation of -0.02, suggesting a weak negative relationship. This heatmap indicates that there was no multicollinearity in the numerical variables.



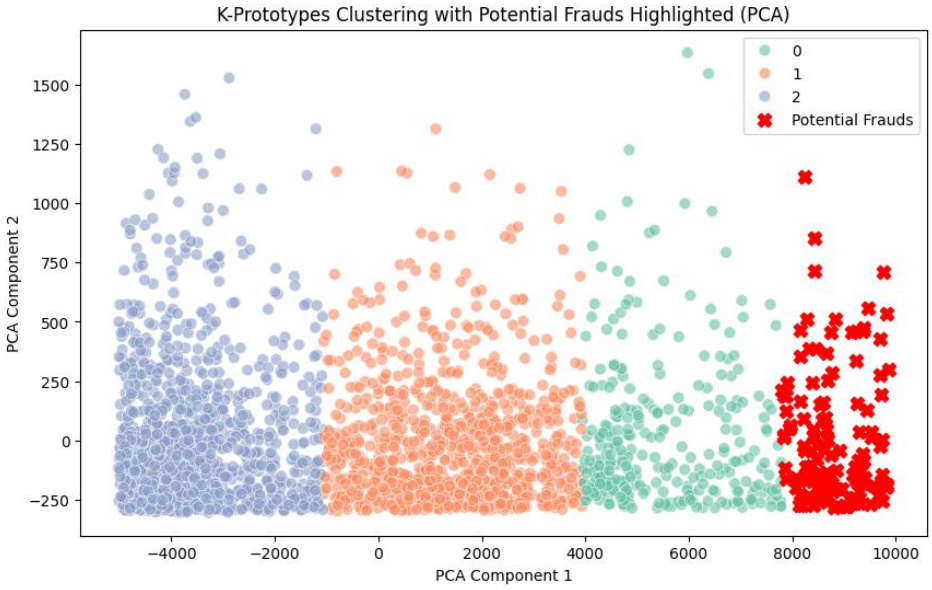
The Cramer's V heatmap above shows the strength of association between pairs of categorical variables in the dataset. Most categorical features exhibit low correlation with one another (values close to 0). Notably, TransactionDay and DayOfWeek show an extremely high correlation (0.95). Hence TransactionDate variable is dropped. Overall, the low multicollinearity among categorical features is exhibited.

Summary of EDA

- Most of the transactions are debit transactions.
- Transactions are fairly evenly split across the three channels.
- The highest number of transactions occurs on Mondays. Debit and credit transactions are common across a wide age range but are particularly concentrated among younger adults and middle-aged customers.

Advanced Analysis

To move beyond basic exploration and dive into pattern detection, the K-Prototypes clustering algorithm is used. This method is well-suited for the dataset due to its mixed data types. This algorithm calculates distances based on Euclidean distance for numerical variables and simple matching dissimilarity for categorical variables.



The PCA-based visualization of the K-Prototypes clustering illustrates the separation of mixed-type transaction data into three distinct clusters, each representing different customer or transaction behavior profiles. The clustering algorithm was applied to a combination of. variables, including numerical features TransactionAmount, CustomerAge, TransactionDuration, LoginAttempts, AccountBalance, and Hour, as well as categorical features such as TransactionType, Location, Channel, CustomerOccupation, and DayOfWeek. In this plot, clusters are color-coded (0,1, and 2), and potential fraudulent transactions are highlighted with red crosses. These potential frauds concentrate in the rightmost segment of the plot.

After performing K-Prototypes clustering and identifying potential fraud patterns, the cluster labels are added as a new feature to the original dataset. This enrichment allows us to leverage the insights from unsupervised learning to enhance the performance of supervised models. The cluster labels serve as a synthesized feature representing underlying transaction behavior, which can be particularly useful for distinguishing between fraudulent and legitimate transactions.

To further improve the detection of fraud, a set of classification models was trained using the label data, where transactions previously identified as potential fraud were treated as positive fraud cases. Various classification algorithms were experimented with, such as:

- Logistic Regression
- Random Forest
- XGBoost
- Support Vector Machine
- Neural Networks

The suggested models were fitted to the complete data set (80% training and 20% testing). We used dummy Encoding for the categorical variables.

In the context of fraud transaction classification, where class imbalance is a common issue due to the rarity of fraudulent cases compared to legitimate ones, traditional evaluation metrics such as accuracy can be misleading. Instead, more appropriate metrics include precision, recall, F1-score, and AUC score. Precision evaluates how many of the transactions identified as fraudulent are actually fraudulent, helping to reduce false positives. Recall focuses on identifying all actual fraud cases, minimizing false negatives. The F1-score, which is the harmonic mean of precision and recall, provides a balanced assessment when both false positives and false negatives carry significant consequences.

The results of the models are as follows:

Logistic Regression

Logistic regression is a fundamental classification algorithm used to model the probability of a binary outcome. The output of this project is a probability score indicating the risk of fraud, which can then be thresholder to classify transactions.

	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	0.96	1.00	0.98

Random Forest

In fraud transaction classification with class imbalance, Random Forest is well-suited due to its robustness and ability to handle skewed data. It provides reliable predictions by averaging results across multiple decision trees, reducing overfitting.

	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00

XGBoost

XGBoost is highly effective for fraud transaction classification, especially in the presence of class imbalance. Its boosting approach focuses on minimizing misclassification by giving more weight to hard-to-predict cases, such as rare fraudulent transactions. With its regularization and handling of missing data, XGBoost delivers strong predictive performance.

	Precision	Recall	F1-score

0	1.00	1.00	1.00
1	1.00	1.00	1.00

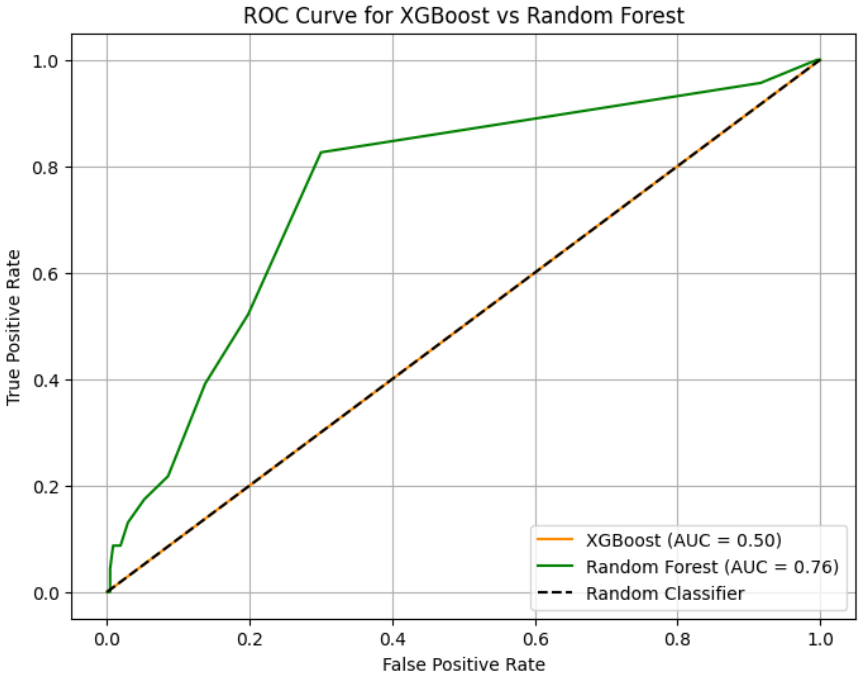
SVM

Support Vector Machine (SVM) is a powerful classification algorithm well-suited for fraud detection, particularly when the dataset is imbalanced. By finding the optimal hyperplane that maximizes the margin between classes, SVM can effectively separate fraudulent and non-fraudulent transactions. With the use of kernel functions, it can handle non-linear patterns common in fraud data.

	Precision	Recall	F1-score
0	0.95	1.00	0.98
1	0.00	0.00	0.00

Best Model

Among the models tested , Random Forest, XGBoost, Logistic Regression, and SVM, XGBoost and Random Forest demonstrated the strongest performance based on evaluation metrics. To determine the best model between them, the AUC (Area Under the Curve) metric was used, as it effectively captures the trade-off between true positive and false positive rates in imbalanced classification problems.



Above plot represent the Receiver Operating Characteristic (ROC) curves for comparing the performance of two classification models: XGBoost and Random Forest. The ROC curve plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across various threshold settings, which helps evaluate the diagnostic ability of classifiers.

In the above plot, the XGBoost model's performance is poor, showing an AUC (Area Under the Curve) of 0.50, which suggests no better performance than random guessing (equivalent to the diagonal line on the ROC graph). This indicates that XGBoost failed to distinguish between positive and negative classes in this context. In contrast, the Random Forest model shows a significantly better performance with an AUC of 0.76, indicating good discriminatory power and a much better capability of correctly classifying positive and negative cases.

The ROC curves for Random Forest rise well above the diagonal random classifier line, particularly between FPR values of 0 and 0.4, and reach a True Positive Rate of around 0.9, showing the model's strong classification ability. The consistent appearance of these plots across different images indicates robustness in the Random Forest model's performance, while the XGBoost model remains equivalent to chance.

Hence it is concluded that the best model is Random Forest.

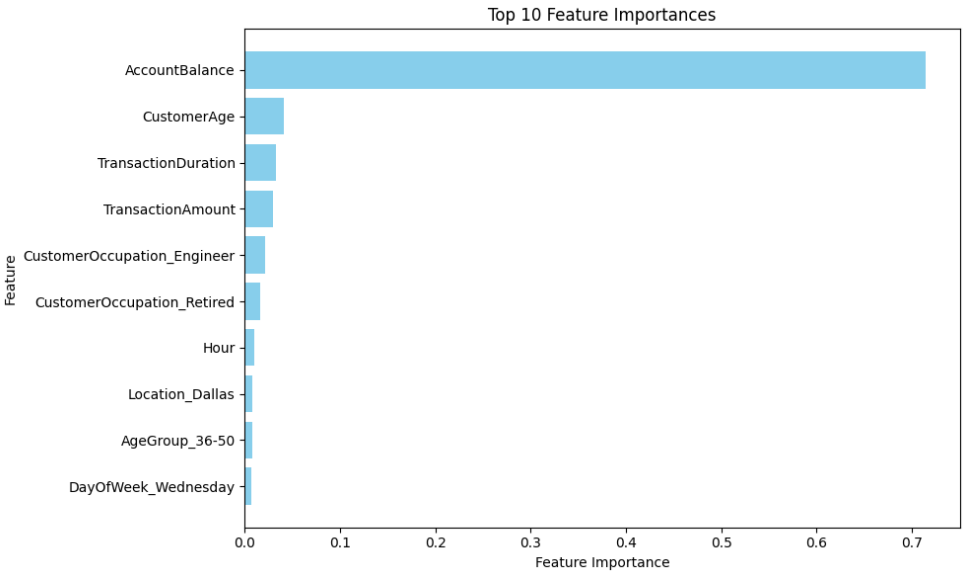
Hyperparameter tuning

To optimize the performance of the Random Forest classifier for fraud detection, Randomized Search Cross-Validation (RandomizedSearchCV) was employed to efficiently explore a wide range of hyperparameter combinations. This method randomly samples from the specified parameter grid and evaluates model performance using cross-validation, reducing computation time compared to GridSearchCV. The best parameters identified were:

n_estimators	154
max_depth	19
min_samples_split	5
min_samples_leaf	3
max_features	sqrt

These tuned parameters enhanced the model's ability to capture complex patterns in the data while controlling overfitting, leading to improved classification of fraudulent transactions.

Top 10 features



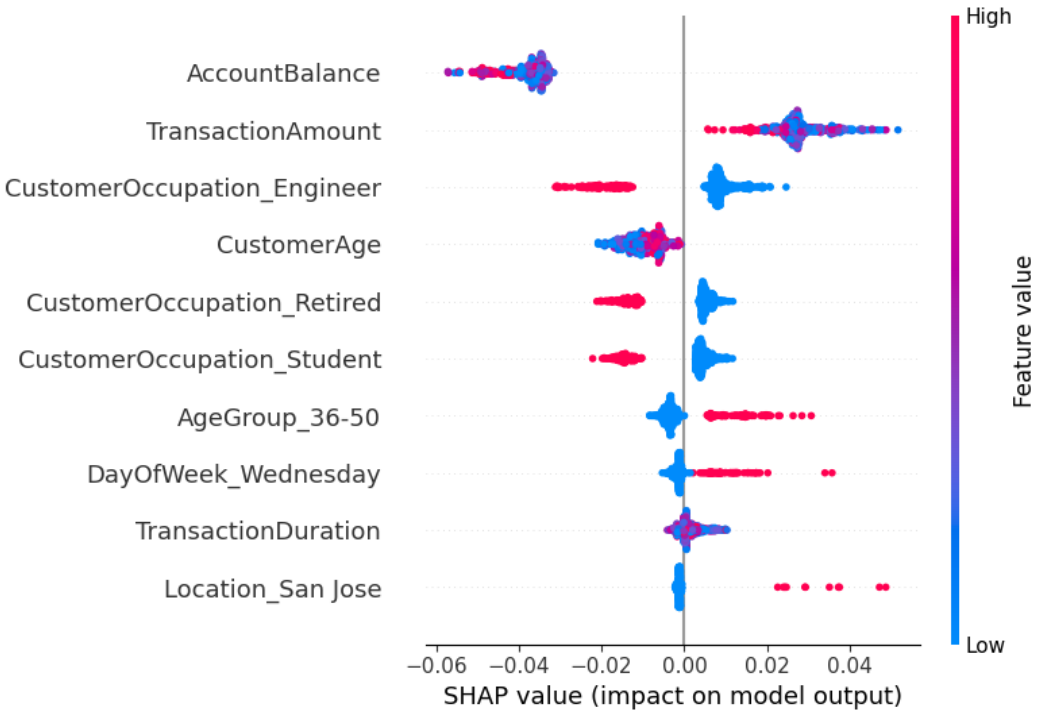
Each bar's length indicates the relative contribution of each feature to the model's prediction power.

According to above graph, AccountBalance is overwhelmingly the most important feature, with a feature importance value exceeding 0.7, demonstrating that it plays a dominant role in predicting the target variable. The next most important feature is CustomerAge, but its importance is much smaller, around 0.05 or less, highlighting a stark contrast. Other features like TransactionDuration, TransactionAmount, CustomerOccupation_Engineer, and CustomerOccupation_Retired contribute minimally to the model's predictions. The bottom-ranked features such as Hour, Location_Dallas, AgeGroup_36-50, and DayOfWeek_Wednesday have negligible impact on the model.

The consistent pattern across different graphs suggests that AccountBalance is a decisive predictor, overshadowing other features. This could indicate potential data imbalance or overfitting toward this particular feature. The model may heavily rely on this financial metric to distinguish between classes or outcomes, which, while effective, might limit its generalizability if account balance is not consistently available or representative

SHAP Plot

A SHAP plot is a way to generate explanations for a single prediction. It shows features that contribute to pushing the output from the base value (average model output) to the actual predicted value. The color-coding indicates the feature value, with red representing high values and blue representing low values.



Among the features, AccountBalance stands out as the most influential factor, with a wide range of SHAP values, suggesting that both high and low account balances can significantly impact the prediction outcome. TransactionAmount, CustomerOccupation_Engineer, and CustomerAge also display moderate importance, with their SHAP values indicating meaningful contributions to the model's output. In contrast, features such as Location_San Jose, DayOfWeek_Wednesday, and TransactionDuration show minimal spread in SHAP values, implying a negligible role in the predictions. Overall, these plots illustrate not just which features are important, but also how the value of each feature influences the model's prediction direction and magnitude. This understanding is crucial for interpreting the model's decision-making process and for identifying which factors are most impactful in determining the predicted outcomes.

Discussion

In this project, we implemented an unsupervised approach to detect fraudulent transactions within a financial dataset. Given the scarcity of labeled fraud data and the evolving nature of fraud patterns, unsupervised methods provide a valuable alternative for identifying anomalies without relying on prior knowledge.

Our approach involved applying clustering and anomaly detection techniques, such as Isolation Forest and K Prototypes, to uncover patterns that deviate significantly from normal transactional behavior. The results indicate that these models can effectively isolate suspicious

transactions by identifying outliers based on transaction features like amount, frequency, and time of occurrence.

However, several challenges remain. Firstly, the imbalance and rarity of fraudulent events mean that evaluation metrics such as precision and recall can be difficult to interpret without labeled ground truth. Although we used proxy indicators and domain knowledge to validate detected anomalies, a lack of labeled data limits the ability to fully assess detection performance. Secondly, false positives remain a concern, as some legitimate transactions may appear anomalous due to unusual but valid behavior. This highlights the need for combining unsupervised detection with expert review or additional verification mechanisms.

Moreover, feature selection and data preprocessing significantly impacted model performance. Features related to transaction velocity, location consistency, and merchant behavior enhanced the detection capability. Future work could explore feature engineering techniques and incorporate external data sources, such as device fingerprints or customer profiles, to improve detection robustness.

Finally, the evolving tactics of fraudsters require adaptive models that can update in near real-time. Integrating online learning algorithms or hybrid supervised-unsupervised frameworks may enhance detection over time.

Conclusion

This project demonstrated the feasibility of using unsupervised learning methods for detecting fraudulent transactions in the absence of labeled data. By leveraging anomaly detection algorithms, we were able to identify suspicious transactions that warrant further investigation. While the models showed a promising ability to highlight outliers, the limitations of unsupervised methods underscore the importance of combining algorithmic detection with domain expertise and continuous model refinement.

Future research should focus on improving detection accuracy through enhanced feature engineering, incorporation of additional contextual data, and development of adaptive systems that evolve with fraud patterns. Ultimately, unsupervised fraud detection offers a scalable and flexible tool to assist financial institutions in mitigating risk and protecting customers against fraud.