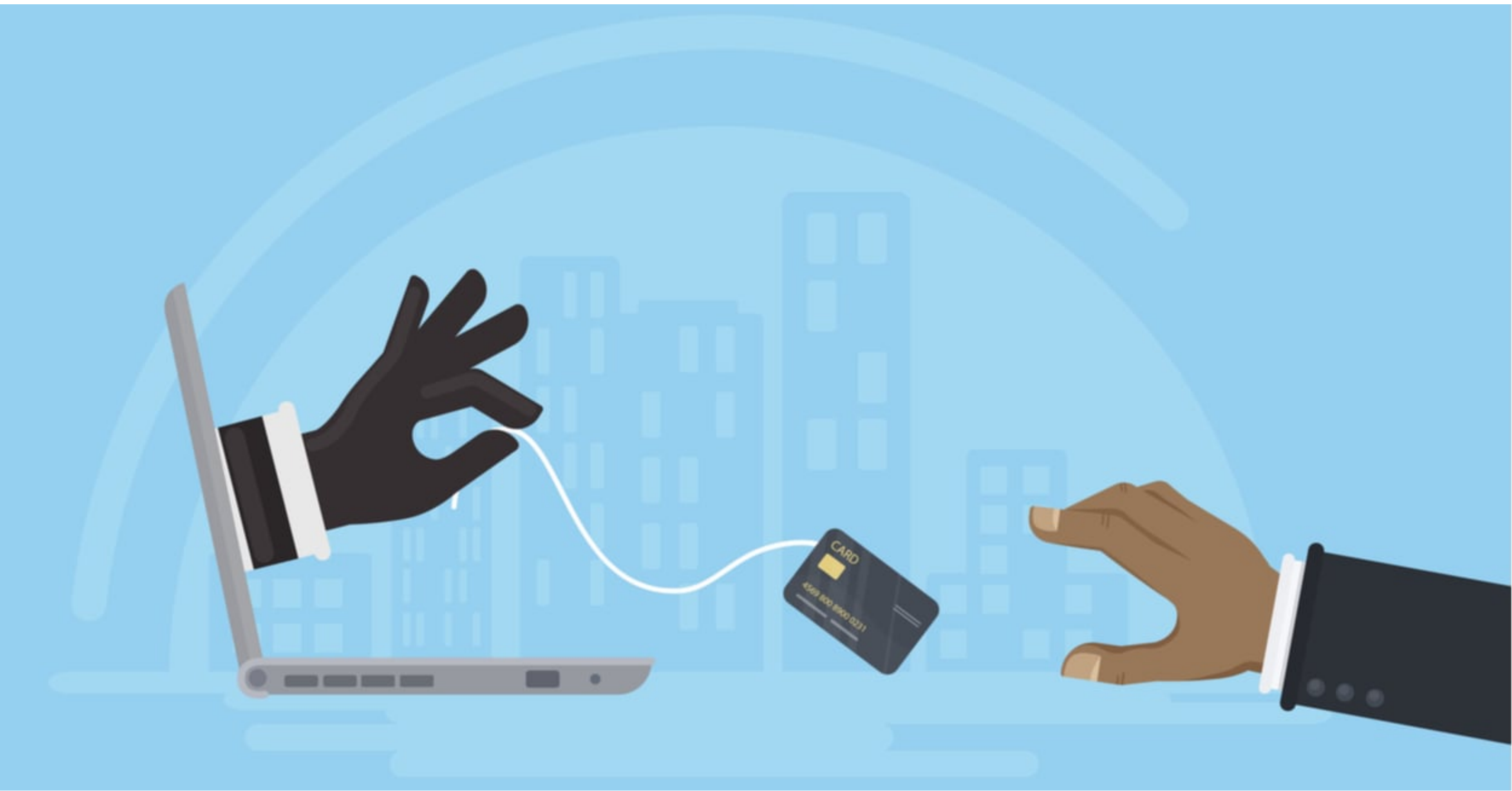
Demo of Using Machine Learning for Credit Card Fraud Detection

Chahid Chadi

May 2024



Contents

1 Introduction **1**

2 Data Exploration **2**

2.1 Data Familiarization 2

2.2 Data Cleaning 2

2.3 Descriptive Statistics 2

3 Data preprocessing **3**

3.1 Data Sampling 3

3.2 Data Visualisation 3

3.3 Feature enginering 6

3.4 Data Encoding 9

4 Trainig and testing the model **11**

4.1 Data Spliting 11

4.2 Model Training 11

4.3 Model's performance 12

4.4 Financial Analysis 13

5 Conclusion **15**

1 | Introduction

Credit card fraud is a pervasive issue that carries significant implications for both individuals and the broader economy. As our financial transactions increasingly shift towards digital platforms, the threat of fraudulent activities has grown exponentially. In this article, we will explore the impact of credit card fraud on the economy and shed light on the role of statistics in understanding and combating this pervasive problem.

Understanding the Magnitude of Credit Card Fraud

In today's interconnected world, credit cards have become an integral part of our daily lives, offering convenience and flexibility in making transactions. However, alongside this convenience comes the ever-present risk of credit card fraud. Statistics reveal the staggering magnitude of this issue. According to recent reports, credit card fraud accounts for billions of dollars in losses annually, making it one of the most significant financial crimes affecting individuals, businesses, and financial institutions worldwide.

Economic Consequences for Individuals

The economic consequences of credit card fraud are far-reaching and multifaceted. At the individual level, victims of fraud often experience financial hardships, including unauthorized charges and stolen funds. These incidents can lead to damaged credit scores, increased interest rates, and a loss of trust in financial systems. Moreover, the time and effort required to resolve fraudulent transactions can take a toll on individuals, causing stress and frustration.

Impact on Businesses and the Economy

Beyond the individual level, credit card fraud poses substantial challenges to businesses and the broader economy. As the incidence of credit card fraud rises, businesses, particularly small enterprises, bear the burden of financial losses, operational disruptions, and reputation damage. Such adverse impacts can hinder economic growth and stability, especially in sectors heavily reliant on consumer spending. Moreover, financial institutions face significant costs associated with fraud detection, prevention, and reimbursement, affecting their profitability and ability to provide competitive services.

The Role of Statistics in Combating Credit Card Fraud

To effectively address credit card fraud and mitigate its economic impact, a deep understanding of the underlying patterns and trends is crucial. This is where the power of statistics comes into play. By analyzing vast amounts of transactional data and employing advanced statistical techniques, experts can identify anomalous patterns, detect fraudulent activities, and develop robust fraud prevention and detection models. Statistical methods such as anomaly detection, machine learning algorithms, and predictive modeling enable financial institutions and law enforcement agencies to stay one step ahead of fraudsters and protect the integrity of financial systems.

Informing Policy and Regulation

Furthermore, statistics provide valuable insights into the evolving nature of credit card fraud, enabling policymakers and regulatory bodies to develop effective strategies and regulations. By studying the characteristics of fraudulent transactions, identifying vulnerable points in the payment ecosystem, and understanding the tactics employed by fraudsters, stakeholders can implement targeted measures to enhance security, promote awareness, and foster collaboration between financial institutions, merchants, and consumers. The continuous analysis of statistical data helps drive proactive measures to combat credit card fraud and safeguard the economy.

Keywords: Credit card fraud detection, machine learning, data imbalance, performance metrics, predictive modeling, feature engineering

2 | Data Exploration

2.1 | Data Familiarization

We have a vast dataset consisting of 5,705,508 transactions, each accompanied by eight variables that provide detailed descriptions of the transactions. This extensive collection of data allows us to gain valuable insights into various aspects of these transactions and analyze them from multiple perspectives.

Variable Name	Variable Type	Description
op_date	object	Operation date
amount	float64	Amount of the transaction
pan_hash	object	Hashed version of the primary account number (PAN)
pan_brand	object	Brand of the payment card (e.g., Visa, Mastercard)
pan_country	object	Country associated with the payment card
pdv	object	Point of sale or merchant identifier
mcc	object	Merchant category code
mutch_id	object	Unique identifier for the transaction

Before we continue, we will start by making a small modification to the variable names to provide more clarity and meaning.

- `amount` is now `TX_AMOUNT`: This modification aims to explicitly denote the transaction amount.
- `op_date` is now `TX_DATETIME`: This change reflects a shift towards more comprehensive timestamping, encompassing both date and time aspects of transactions.
- `pdv` is now `TERMINAL_ID`: This alteration provides a more descriptive identifier for the point-of-sale terminals involved in transactions.
- `mutch_id` is now `TX_FRAUD`: This renaming underscores the variable's role in flagging potentially fraudulent transactions.
- `pan_hash` is now `CUSTOMER_ID`: This adjustment clarifies the representation of customer identifiers.

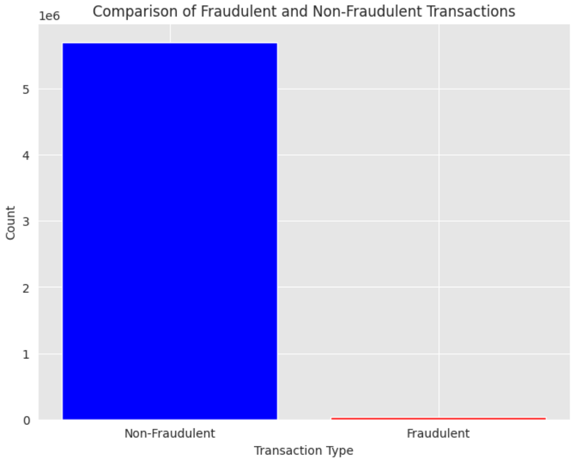
2.2 | Data Cleaning

Data cleaning is a vital process that tackles various data quality issues such as missing values, outliers, and inconsistencies. It is a critical step in preparing data for analysis, aiming to eliminate biases and errors. Initially, we addressed space in object variables. Subsequently, we identified 17 transactions with missing values in the `pan_country` variable. Given their insignificance in the dataset and the absence of fraudulent activity, we opted to remove them. Additionally, we encountered one missing value in the `pan_brand` variable, which we retained and replaced with "unknown" for clarity and completeness.

2.3 | Descriptive Statistics

The dataset contains several variables for analysis. Here is a general idea about the unique number of each variable :

Variable Name	Number of Unique V
CUSTOMER_ID	916244
pan_brand	4
pan_country	144
TERMINAL_ID	427
mcc	77
TX_FRAUD	2



The primary obstacle in this challenge lies within the dataset's imbalance, where the count of non-fraudulent transactions (5,704,680) vastly surpasses that of fraudulent transactions (414). Remarkably, fraudulent transactions account for less than 0.0073 percent of the dataset. This substantial class imbalance presents significant hurdles during machine learning model training. Models often exhibit a bias toward the majority class, impeding their ability to discern patterns within the minority class. Fortunately, numerous proposed solutions aim to mitigate this issue, ranging from algorithmic adjustments to sampling techniques and ensemble methods, each tailored to address the intricacies of imbalanced data.

3 | Data preprocessing

3.1 | Data Sampling

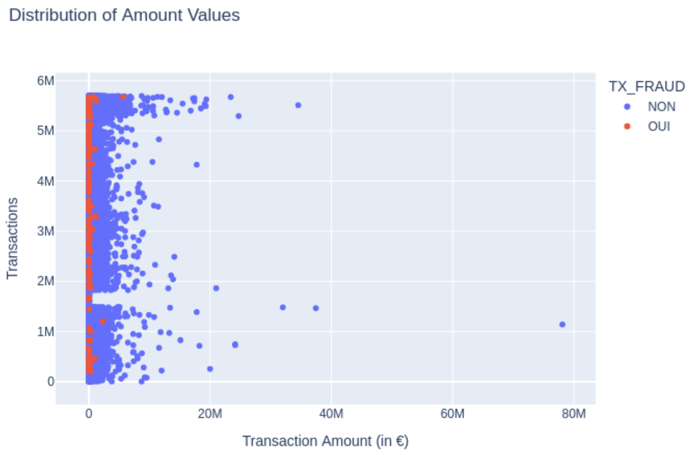
When confronted with extensive datasets, like the aforementioned 5 million transaction records, the computational demands of processing such vast amounts of data can swiftly become a bottleneck. In such situations, data undersampling emerges as a pragmatic strategy to navigate the challenges posed by this sheer volume of information. Following the undersampling process, we are left with 923,484 non-fraudulent transactions and 414 fraudulent transactions. Although the dataset retains its imbalance, the severity of this imbalance is notably reduced compared to the initial scenario. Thus, the objective was to retain approximately equivalent information while utilizing a reduced dataset size. To achieve this, we employed the Tomelinks undersampling technique, ensuring a balanced representation of both fraudulent and non-fraudulent transactions.

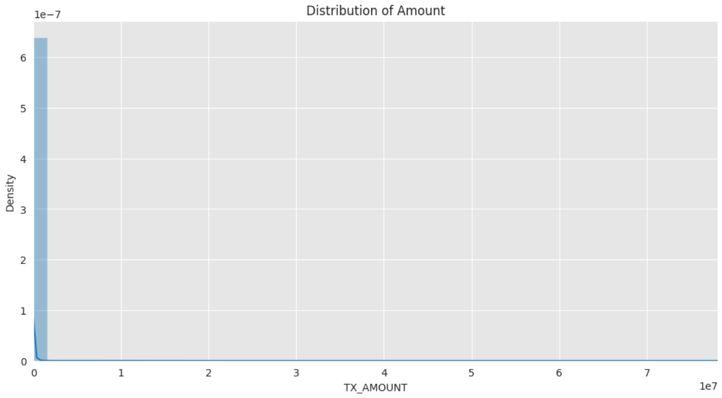
Variable Name	Number of Unique Values
CUSTOMER_ID	314946
pan_brand	4
pan_country	133
TERMINAL_ID	370
mcc	75
TX_FRAUD	2

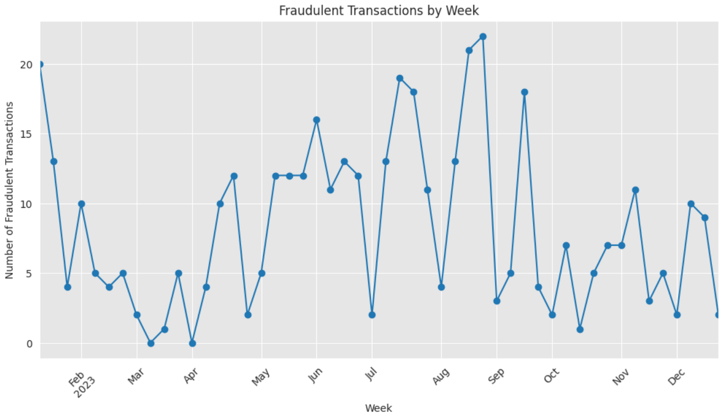
The table presented below depict the variable counts in the new dataset obtained after the sampling procedure. It is evident that the number of customers has decreased by a third, while the counts for the other variables remain relatively consistent. This outcome is favorable, indicating that we have successfully preserved the representation of all variables within a reduced dataset size.

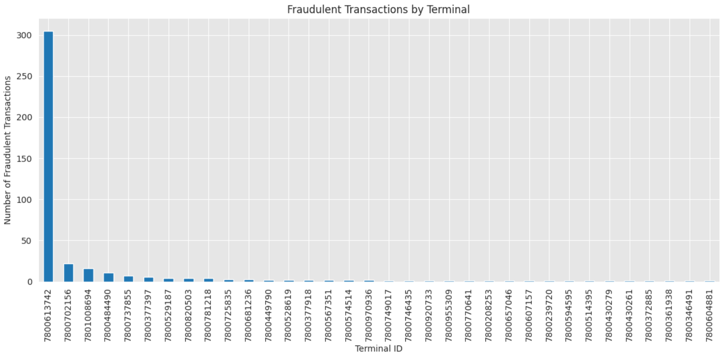
3.2 | Data Visualisation

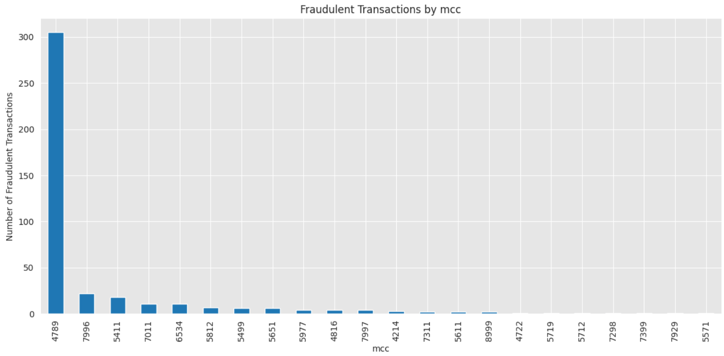
The dataset contains several variables for analysis. Here is a visualization that can help provide a general idea and extract new insights, especially about customer behavior in terms of fraudulent transactions.

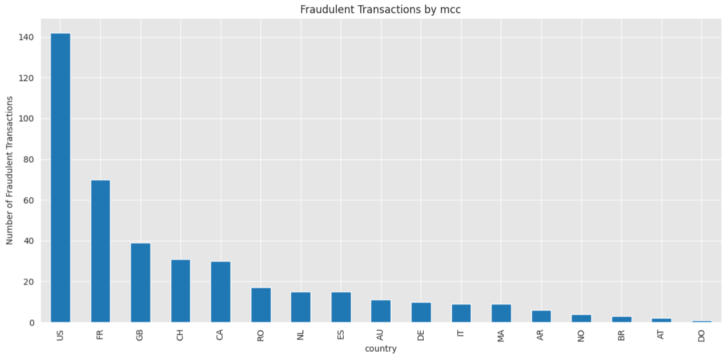

Distribution of Amount Values
TX_FRAUD
NON
OUI
Transactions
6M
5M
4M
3M
2M
1M
0
20M
40M
60M
80M
Transaction Amount (in €)

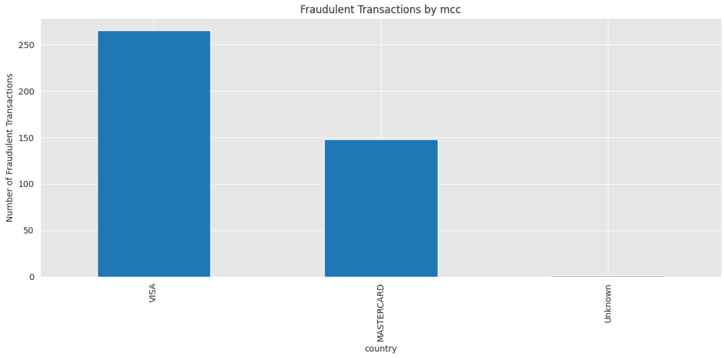

Distribution of Amount
1e−7
6
5
4
3
2
1
0
Density
0
1
2
3
4
5
6
7
TX_AMOUNT
1e7

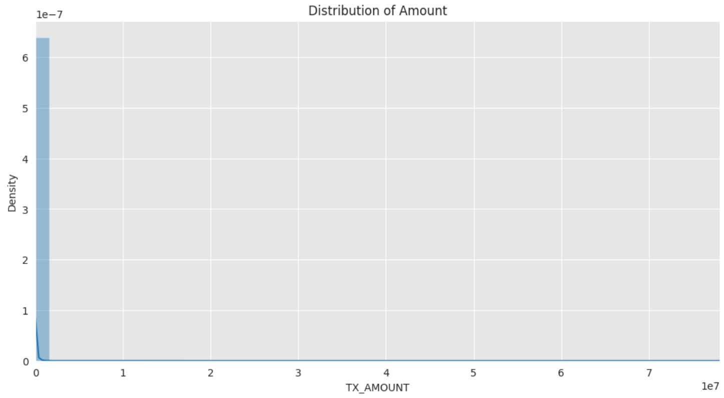

Fraudulent Transactions by Week
Number of Fraudulent Transactions
20
15
10
5
0
Feb
2023
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Week


Fraudulent Transactions by Terminal
Number of Fraudulent Transactions
300
250
200
150
100
50
0
78006613742
78007021 56
78010008694
78004484490
78007737855
78003377397
78005529187
78008820503
78007781218
78007725835
78006681236
78004449790
78005528619
78003377918
78005567351
78005574514
78009970936
78007749017
78007746435
78009920733
78009955309
78007770641
78002208253
78006657046
78006607157
78002239720
78005594595
78005514395
78004430279
78004430261
78003372885
78003361938
78003346491
78006604881
Terminal ID









These charts offer a comprehensive portrayal of fraudulent transaction patterns across various dimensions. They highlight the breadth of fraudulent activity, revealing consistent activity throughout the year with notable spikes, as well as the global nature of fraudulent transactions across different merchant category codes and countries. The charts emphasize the need for comprehensive detection and prevention measures to address the prevalent issue of fraudulent transactions.

Conclusion

By analyzing the distribution of fraudulent transactions across various factors, we can strategize on leveraging this information to create new variables that aid in fraud detection. For instance, we could devise risk scores for each factor, such as countries, MCC , or terminals. These risk scores could provide a quantitative measure of the likelihood of fraud associated with each factor, enabling more targeted and effective fraud detection mechanisms.

3.3 | Feature enginering

The main objective of machine learning is to extract patterns to turn data into knowledge. Since the beginning of this century, technological advances have drastically changed the size of data sets as well as the speed with which these data must be analyzed. Modern data sets may have a huge number of instances, a very large number of features, or both. In most applications, data sets are compiled by combining data from different sources and databases (containing both structured and unstructured data) where each source of information has its strengths and weaknesses. Before applying any machine learning algorithm, it is therefore necessary to transform these raw data sources into interesting features that better help the predictive models. This essential step, which is often denoted feature engineering, is of utmost importance in the machine learning process. We believe that data scientists should be well aware of the power of feature engineering and that they should share good practices. An important set of interesting features can be created based on the famous Recency, Frequency, Monetary (RFM) principle. Recency measures how long ago a certain event took place, whereas frequency counts the number specific events per unit of time. Besides recency features, we also present several other time-related features. Features related to monetary value measure the intensity of a transaction, typically expressed in a currency such as Euros or USD.

- **TRANSACTION_TX_AMOUNT_LOG**:
  We have applied a logarithmic transformation to the `TX_AMOUNT` variable. This transformation is particularly effective at mitigating the influence of extreme values or outliers. Given that our dataset includes transactions with exceptionally high or low amounts, applying the logarithmic transformation helps to reduce their impact on the model, thereby enhancing its robustness.

- **CUSTOMER_Z_SCORE**:
  The Z-score, also known as standard score, is a measure of how many standard deviations a data point is away from the mean of the dataset. Mathematically, it is calculated as:

  $$Z = \frac{X - \mu}{\sigma}$$

  where: X is the value of the data point, μ is the mean of the dataset, and σ is the standard deviation of the dataset.
  The Z-score measures how many standard deviations a data point is from the mean. A high Z-score for a transaction amount suggests that it is significantly different from the average transaction amount for that customer. Large deviations can indicate potential anomalies or fraudulent behavior.

- **transaction_amount_kurtosis**:
  Kurtosis is a statistical measure that describes the shape, or ”tailedness,” of a distribution. It measures the extent to which a distribution is peaked or flat compared to a normal distribution. Mathematically, kurtosis is calculated as:

  $$\text{Kurtosis} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^4/n}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2/n\right)^2} - 3$$

  where: x_i represents each individual data point, $\bar{x}$ is the mean of the dataset, and n is the number of data points. High kurtosis indicates a distribution with heavy tails, meaning it has more extreme values than a normal distribution. In the context of transaction amounts, high kurtosis could suggest that there are more transactions at the extreme ends of the distribution, which might be indicative of fraudulent behavior, such as large or unusually small transactions.

- **transaction_amount_skew**:
  Skewness is a statistical measure that quantifies the asymmetry of a distribution. It measures the degree to which the distribution of data points deviates from symmetry around the mean. Mathematically, skewness is calculated as:

  $$\text{Skewness} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3/n}{\left(\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2/n}\right)^3}$$

  where: x_i represents each individual data point, $\bar{x}$ is the mean of the dataset, and n is the number of data points.

  Skewness measures the direction and extent of asymmetry in the distribution. A positive skewness indicates a longer right tail, meaning the distribution is skewed to the right (positively skewed), while a negative skewness indicates a longer left tail, meaning the distribution is skewed to the left (negatively skewed).

  For identifying unusual patterns, skewness is helpful because: High skewness values indicate a significant deviation from symmetry, suggesting that the distribution is highly skewed. Unusual patterns in transaction amounts, such as a disproportionately large number of transactions on one side of the distribution (e.g., many small transactions but few large transactions), may be indicative of fraudulent activity. Skewness can help detect these asymmetries in the distribution, making it easier to identify potential fraud cases.

- **Proba_Von_Misses_Distriution**:
  The fundamental concept underlying this variable is to model transaction timestamps as periodic variables employing the von Mises distribution. By utilizing the mean and standard deviation (std), we can compute the von Mises distribution, enabling us to ascertain the likelihood that a transaction adheres to this distribution pattern.

- **TX_DURING_WEEKEND**:
  This feature serves to indicate whether a transaction occurred during the weekend. This distinction is particularly pertinent due to a notable prevalence of fraudulent activities transpiring during weekend periods.

- **TX_DURING_NIGHT**:
  The variable serves as an indicator of whether a transaction took place during nighttime hours. This distinction holds significant relevance, as nighttime transactions often represent a period of heightened risk for fraudulent activity.

- **TX_TIME_DAYS**:
  This variable transforms the transaction dates into days, representing the number of days elapsed since a reference date.

- **TX_TIME_SECONDS**:
  This variable transforms transaction dates into seconds, representing the duration elapsed since a reference date.

- **CUSTOMER_ID_NB_TX_IDAY_WINDOW**:
  This variable represents the number of transactions made by a customer within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days. Despite the varying temporal scopes, each iteration of this metric holds significant importance as it offers a condensed view of customer transactional behavior within specific timeframes.

- **CUSTOMER_ID_AVG_AMOUNT_1DAY_WINDOW**:
  The variable represents the average amount of money spent by a customer within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days. Each iteration of this metric offers valuable insights into customer spending habits within specific temporal windows.

- **CUSTOMER_ID_MEDIAN_TX_iDAY_WINDOW**:
  The variable represents the median amount of money spent by a customer within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days. Each iteration of this metric provides valuable insights into customer spending behavior within specific temporal windows.

- **CUSTOMER_ID_RANGE_TX_90DAY_WINDOW**:
  The variable represents the range of the amount of money spent by a customer within a 90-day time period. This metric offers insights into the variability of customer spending behavior over a medium-term timeframe.

- **TERMINAL_ID_NB_TX_iDAY_WINDOW**:
  This variable denotes the number of transactions made at a terminal within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days.

- **TERMINAL_ID_AVG_AMOUNT_iDAY_WINDOW**:
  This variable describes the average amount of money spent by customers at a terminal within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days.

- **TERMINAL_ID_MEDIAN_TX_iDAY_WINDOW**:
  This variable describes the median amount of money spent by customers at a terminal within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days.

- **TERMINAL_ID_RANGE_TX_iDAY_WINDOW**:
  This variable describes the range of transaction amounts made by customers at a terminal within various time periods, where i can take values such as 1, 7, 30, 90, or 365 days.

- **isFirstTransaction**:
  This variable indicates whether a particular transaction is the first transaction made by a customer. It serves as a binary indicator, with a value of 1 indicating that the transaction is indeed the customer's first transaction and a value of 0 indicating that it is not.

- **isCartLocal**:
  This variable indicates whether a customer account was created locally or not. It is a binary indicator, with a value of 1 suggesting that the account was created locally and a value of 0 suggesting otherwise.

- **IsSpecialDay** :
  This variable indicates whether a transaction occurred on a special day, such as a holiday, end of the

month, or other significant occasions throughout the year. It is a binary indicator, with a value of 1 suggesting that the transaction took place on a special day and a value of 0 indicating otherwise.

- **RISK_SCORE_TERMINAL_ID**:
  This variable denotes the risk score assigned to each terminal, indicating the level of risk associated with fraudulent activity at that terminal. The risk score is derived from the proportion of fraudulent transactions observed for each terminal, with higher scores indicating a higher likelihood of fraudulent activity. The risk scores typically range from 1 to 5, with 1 representing the lowest risk level and 5 representing the highest risk level.

- **RISK_SCORE_MCC**:
  This variable represents the risk score assigned to each Merchant Category Code (MCC), indicating the level of risk associated with fraudulent activity for transactions categorized under that MCC. The risk score is determined based on the proportion of fraudulent transactions observed within each MCC category. The risk scores are typically ranging from 1 to 5, with 1 indicating the lowest risk level and 5 indicating the highest risk level.

- **RISK_SCORE_COUNTRY**:
  This variable denotes the risk score assigned to each country, indicating the level of risk associated with fraudulent activity originating from that country. The risk score is derived from the proportion of fraudulent transactions observed for each country. Values: The risk scores are categorical and typically range from 1 to 5, with 1 representing the lowest risk level and 5 representing the highest risk level.

- **RISK_SCORE_TIME**:
  This variable represents the risk score assigned to different time periods within a day, dividing it into four distinct segments. Each segment corresponds to a specific part of the day, such as morning, afternoon, evening, and night. The risk score is assigned to each time segment based on the likelihood of fraudulent activity occurring during that period.

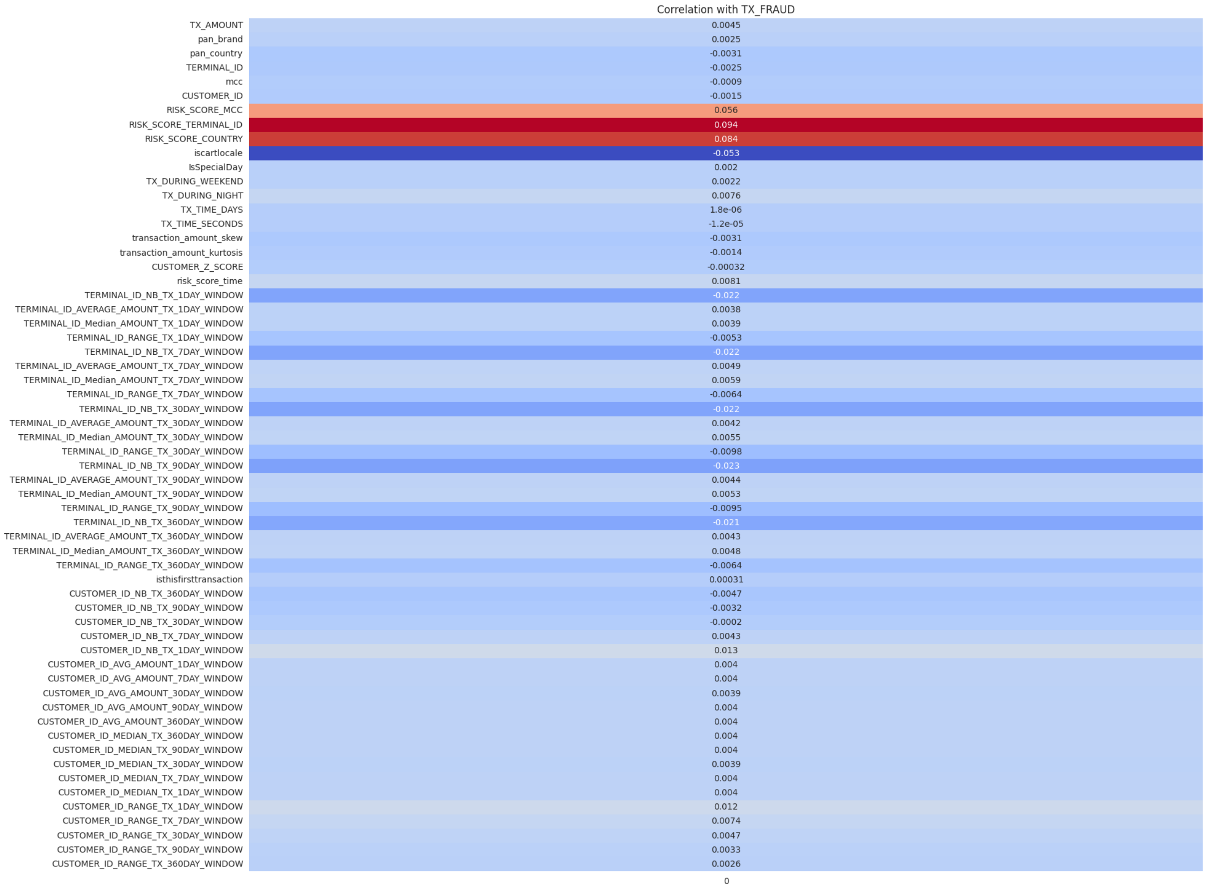
- **RISK_SCORE_AMOUNT**:
  This variable represents the risk score assigned to different spending amount intervals. The spending amounts are divided into distinct intervals or ranges, and each interval is assigned a risk score based on the likelihood of fraudulent activity associated with transactions falling within that range.

3.4 | Data Encoding

In our data encoding phase, we employed a combination of one-hot encoding, label encoding, and standard scaling techniques using the scikit-learn library.

1. **One-Hot Encoding:** This method was utilized to transform categorical variables into a binary format, enhancing the compatibility of our data with machine learning algorithms.
2. **Label Encoding:** Complementary to one-hot encoding, label encoding was selectively applied to categorical variables where a natural order among categories existed.
3. **Standard Scaling:** Additionally, we leveraged standard scaling to normalize numerical features across the dataset.

Before proceeding with the data splitting and training phase, it is prudent to examine the correlation between the target variable, `TX_FRAUD`, and the other variables we have created. Understanding these correlations provides valuable insights into the relationship between fraudulent transactions and the features engineered for prediction. By assessing the correlation, we can discern which variables exhibit strong associations with fraudulent activity and may thus be more influential in our predictive model. Conversely, variables showing weak or negligible correlations may warrant further investigation or potentially be excluded from the modeling process to streamline computational resources.



Based on this chart, it's evident that `RISK_SCORE_MCC` ,`RISK_SCORE_TERMINAL_ID`, `RISK_SCORE_COUNTRY`, and iscartelocale are the most strongly correlated variables with TX_FRAUD. This correlation is logical, given that the level of risk associated with fraudulent transactions often depends on factors such as the merchant category, terminal identity, country of origin, and whether the transaction occurred locally or not. These variables play a crucial role in assessing the likelihood of fraud, as they capture key aspects of transactional behavior and patterns that may indicate suspicious activity. Therefore, incorporating these variables into our predictive model is essential for accurately identifying and preventing fraudulent transactions.

4 | Trainig and testing the model

4.1 | Data Spliting

We conducted our analysis using a sample of 100,414 transactions extracted from the previous dataset. To ensure the robustness of our model, we partitioned 80 percent of the sampled data for training purposes, while the remaining 20 was earmarked for testing purposes.

(a) Resampled Class Counts

Class	Count
0	100,000
1	414

(b) Train Class Counts

Class	Count
0	79,997
1	334

(c) Test Class Counts

Class	Count
0	20,003
1	80

4.2 | Model Training

XGBoost

(Extreme Gradient Boosting) is a powerful ensemble machine learning algorithm that has gained widespread popularity due to its exceptional performance in a variety of tasks, including classification and regression. It is an implementation of gradient boosted decision trees designed to be highly efficient, flexible, and scalable.

Handling Imbalance

To address the class imbalance problem in my dataset, I employed the SMOTE (Synthetic Minority Over-sampling Technique) technique from the imblearn library. SMOTE is a widely used oversampling method that creates synthetic instances of the minority class by interpolating between existing minority instances. This helps to balance the class distribution and improve the model's ability to learn patterns from the minority class effectively.

hyperparameters tuning

After resampling the training data using SMOTE, I focused on tuning the hyperparameters of the XGBoost model to optimize its performance. XGBoost has several hyperparameters that can significantly impact the model's behavior, such as the number of estimators (trees), maximum depth of the trees, learning rate, regularization parameters, and others.

Performance Metrics for Credit Card Fraud Detection

In the context of credit card fraud detection, where the dataset is typically highly imbalanced (with a significantly larger number of legitimate transactions compared to fraudulent ones), it is crucial to evaluate the performance of the classification model using appropriate metrics. Here, we discuss the following metrics:

Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{4.1}$$

A high precision indicates that the model is making accurate predictions when it identifies a transaction as fraudulent.

Recall (Sensitivity)

Recall, also known as sensitivity, measures the proportion of actual positive instances that the model correctly identifies:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{4.2}$$

A high recall is desirable in fraud detection to minimize the number of fraudulent transactions that go undetected.

F1-score

The F1-score is the harmonic mean of precision and recall, providing a balanced measure between the two:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

The F1-score is useful when both precision and recall are important, as it combines them into a single metric.

Accuracy

Accuracy measures the proportion of correct predictions made by the model:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Instances}} \tag{4.4}$$

However, in imbalanced datasets, accuracy can be misleading as the majority class can dominate the metric, making it less informative.

Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

The AUC-ROC is a widely used metric for evaluating binary classification models. It measures the trade-off between true positive rate (recall) and false positive rate across different threshold settings:

$$\text{AUC-ROC} = \int_0^1 \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \, d\left(\frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}}\right) \tag{4.5}$$

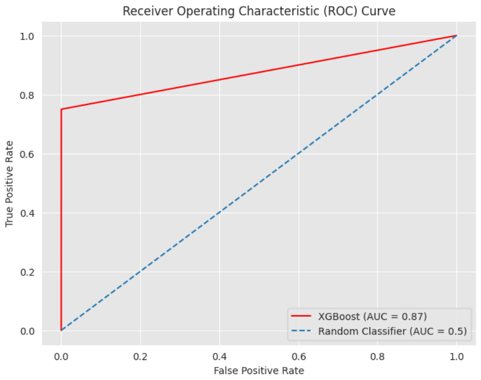
An AUC-ROC of 1.0 represents a perfect classifier, while an AUC-ROC of 0.5 corresponds to a random classifier.

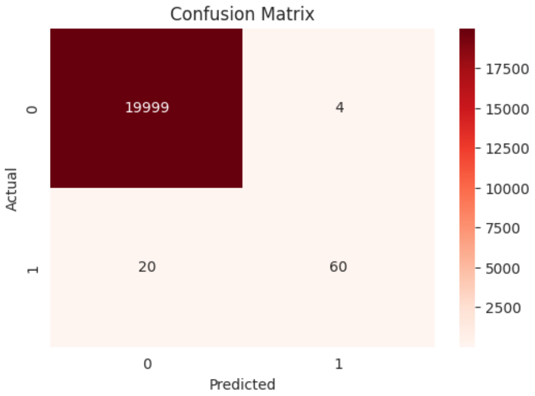
In the case of credit card fraud detection, where the cost of false negatives (missed fraudulent transactions) is typically higher than the cost of false positives, recall and AUC-ROC are often prioritized over other metrics.

4.3 | Model's performance

In this section, we evaluate the performance of our XGBoost classifier on the test data. The XGBoost model was trained on a binary classification problem, where the goal was to accurately identify fraudulent credit card transactions. Given the imbalanced nature of the dataset, with a significantly larger number of legitimate transactions compared to fraudulent ones, we employed the SMOTE (Synthetic Minority Over-sampling Technique) technique to balance the class distribution during training.

To assess the model's performance, we calculated several evaluation metrics, including precision, recall, F1-score, accuracy, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Additionally, we generated a classification report, which provides a detailed breakdown of the model's performance for each class.





Based on the results presented in the table below, our model demonstrates commendable performance. Specifically, the model exhibits high precision in predicting non-fraudulent transactions, achieving a perfect precision score of 1.00, implying that all non-fraudulent transactions were correctly identified by the model.
In terms of detecting fraudulent transactions, while the precision score slightly decreased to 0.94, indicating that there were some false positives, the model still achieved a substantial recall score of 0.75. This implies that the model successfully identified 75 percent of the fraudulent transactions present in the dataset.

	Scores			Support
Class	Precision	Recall	F1-score	
0	1.00	1.00	1.00	20,003
1	0.94	0.75	0.83	80
accuracy			1.00	20,083
macro avg	0.97	0.87	0.92	20,083
weighted avg	1.00	1.00	1.00	20,083

These results underscore the effectiveness of our model in accurately distinguishing between fraudulent and non-fraudulent transactions. Despite the presence of some false positives, the model's overall performance is robust, highlighting its potential for real-world applications in fraud detection.

Conclusion

Maximizing recall to identify the most fraudulent transactions can increase false positives, leading to customer dissatisfaction and operational challenges. Prioritizing precision to minimize false positives may overlook some fraud. The optimal balance between recall and precision depends on the organization's objectives and risk tolerance. Carefully weighing this trade-off is essential for an effective fraud detection strategy that safeguards financial integrity and maintains customer trust.

4.4 | Financial Analysis

The classification report shows that the XGBoost model has achieved excellent overall performance on the fraud detection task. However, we shouldn't forget that the main goal of this model is to help banks detect fraud so they don't lose money by stopping fraudulent transactions.

The total amount of money for the transactions that are fraud is a staggering 3,861,924.00. This figure represents the financial exposure the bank faces from fraudulent activities before the intervention of the fraud detection model. It is a critical metric that underscores the importance of having an effective fraud prevention system in place.
From a financial standpoint, the model's ability to identify 75% of fraudulent transactions carries significant weight. Extrapolating this detection rate to the financial realm suggests that the model has the potential to prevent losses equivalent to 75% of the total amount spent on fraudulent transactions. Given an

estimated total expenditure of approximately $3,861,924 on fraudulent transactions, if the model effectively intercepts 75% of these transactions, it could save the bank approximately $2,896,443. This outcome would undoubtedly yield substantial savings, underscoring the considerable financial benefits stemming from the model's predictive accuracy.

Category	Amount (USD)
Total amount of transactions that are fraud	3861924.00
Total amount of Transactions that are fraud and predicted as fraud	1813044.00
Total amount of Transactions that are fraud and predicted as not fraud	2048880.00
Total amount of transactions that are not fraud and predicted as fraud	1036100.00

Upon closer examination of the actual figures, it becomes apparent that the model's cost-saving efficacy is approximately 45%. While this marks a notable improvement, it falls short of optimal performance. Additionally, it's crucial to acknowledge that the model also yields false positives, incurring a cost of $1,036,100.

5 | Conclusion

The primary challenge in credit card fraud detection lies in the imbalance of the data. This makes machine learning a promising solution, as it can effectively address this issue through the use of various tuning techniques. By carefully tuning and combining different methods, we can achieve acceptable results. However, these results are highly dependent on the specific context and objectives of each entity. Despite these dependencies, machine learning remains a robust approach for tackling credit card fraud.
This work, while not fully realistic, provides a general insight into the mechanisms of machine learning in fraud detection. The main challenge moving forward is to implement these techniques in a more realistic scenario that mirrors real-life fraud detection more closely. Additionally, there is still significant work to be done in refining performance metrics. As highlighted in the financial analysis, a model with high recall is not necessarily sufficient to qualify as an effective model. It is crucial to balance recall with other metrics such as precision and accuracy to ensure a reliable and comprehensive fraud detection system.
Future work should focus on the following areas:

1. **Realistic Implementation**: Developing and testing models in real-world scenarios to validate their effectiveness and adaptability.
2. **Performance Metrics**: Enhancing the evaluation metrics to ensure a balanced trade-off between recall, precision, and other relevant measures.
3. **Data Preprocessing**: Improving data preprocessing techniques to better handle data imbalance and enhance model performance.
4. **Model Robustness**: Ensuring that models are robust and can generalize well across different datasets and fraud patterns.
5. **Scalability**: Making sure that the models can scale efficiently to handle large volumes of transactions without a significant drop in performance.

By addressing these areas, we can advance towards more effective and practical solutions for credit card fraud detection.

Then End