**CCT College Dublin**

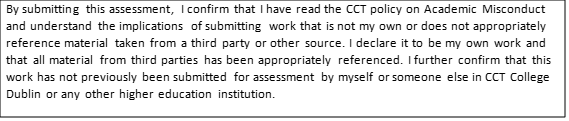
**Assessment Cover Page**

Form

|  |  |
| --- | --- |
| **Module Title:** | Strategic Thinking |
| **Assessment Title:** | Analysis of Financial Fraud |
| **Lecturer Name:** | James Garza |
| **Student Full Name:** | Eriton Delgado Pereira |
| **Student Number:** | 2023457 |
|  |  |
|  |  |
|  |  |
|  |  |
| **Assessment Due Date:** | 11-05-2024 |
| **Date of Submission:** | 15-05-2024 |

Form

**Declaration**



## 

## **Prediction of Fraudulent Financial Transactions.**

### *By*

#### Eriton Delgado Pereira – Student ID: 2023457

*Higher Diploma in Science in Data Analytics for Business*

*Strategic Thinking*

*Jame Garza*

*CCT College*

*Dublin, Ireland*

Summary

[**Prediction of Fraudulent Financial Transactions.** 2](#_Toc166708710)

[*By* 2](#_Toc166708711)

[**Abstract:** 4](#_Toc166708712)

[Introduction 5](#_Toc166708716)

[Business Description 5](#_Toc166708717)

[Hypothesis 5](#_Toc166708718)

[General Goal 5](#_Toc166708719)

[Success Criteria/Indicators 6](#_Toc166708720)

[**Technologies Used** 6](#_Toc166708721)

[**Libraries** 6](#_Toc166708722)

[Models 7](#_Toc166708723)

[Logistic Regression 7](#_Toc166708724)

[Random Forest 7](#_Toc166708725)

[XGBoost 8](#_Toc166708726)

[Exploratory Data Analysis (EDA) 10](#_Toc166708727)

[Loading and Preparing the Data 10](#_Toc166708728)

[Distribution of Transactions by Type 10](#_Toc166708729)

[Distribution of Transaction Amounts 11](#_Toc166708730)

[Correlation Analysis between Variables 12](#_Toc166708731)

[Distribution of Fraudulent and Non-Fraudulent Transactions 13](#_Toc166708732)

[Descriptive Statistics of Variables 13](#_Toc166708733)

[Detailed Explanation and Comparison of Model Results 15](#_Toc166708734)

[Logistic Regression 15](#_Toc166708735)

[Random Forest 16](#_Toc166708736)

[**XGBoost** 17](#_Toc166708737)

[**Comparison of Models** 18](#_Toc166708738)

[Conclusion: 20](#_Toc166708739)

[Figure 1 - Transaction Distribution by type 10](#_Toc166709056)

[Figure 2 - Transaction distribution by frequency x amount 11](#_Toc166709057)

[Figure 3 - Correlation Matrix 12](#_Toc166709058)

[Figure 4 - Distribution of Fraudlant x Non Fraudlant Transaction 13](#_Toc166709059)

[Figure 5 - Model Comparison 19](#_Toc166709060)

[Figure 6 - ROC Curvas 20](#_Toc166709061)

# **Abstract:**

In the Strategic Thinking class, taught by Professor James Garza, we received the mission of producing our first work in data analysis. After some extracurricular discussions, we decided to look for a topic related to the financial sector, as we believe that the application of the knowledge of statistics, data preparation, machine learning and strategic thinking that are being acquired in the course, applied to topics relevant to this sector, will bring a greater alignment with the vacancies and job opportunities that we intend to seek in the market. And it is adherent with the business area we intend to follow.

After some effort evaluating possibilities and searching for datasets, in sources disseminated in the classroom such as Kaggle, World Bank, Eurostat, Covid Google among others, we decided to use a database of studies made available in Kaggle to develop a machine learning model to identify fraudulent transactions.

We believe that this issue is a serious, recurrent, and highly relevant problem in the global financial system. It is common knowledge that fraud in financial transactions occurs all the time, generating numerous losses not only for financial institutions, but also for the victims of fraud and for society, since these losses are somehow passed on to customers in the existing costs to have a bank account.

This report aims to deep dive into a huge a dataset of financial transactions and based in its information predict with applying machine learning models, what transaction was fraudulent, note that, in the dataset we have this information (fraude or not) and I will pretend to use it to test and train my model and compare its accuracy against the information registered in the dataset.

# Finally, the report will follow the Cross-Industry Standard Process for Data Mining (CRISP-DM). It is an approach to project management and goes through the stages: Business Understanding, Data Understanding, Data Preparation, Modelling and Evaluation and Deployment.

# **Key-words: CRISP-DM, financial transaction, fraud.**

# **Code available at** [**GitHub**](https://github.com/eriton2023457/CA2-Strategic-Thinking)**.**

# Introduction

In today's financial landscape, detecting fraud in financial transactions is crucial. With the rise of digital banking and online transactions, the need for robust and efficient methods to identify fraudulent activities has become paramount. This project leverages machine learning techniques to detect fraudulent transactions within a dataset sourced from Kaggle. By applying advanced data analysis and machine learning models, we aim to develop a system capable of accurately identifying fraudulent activities, thus contributing to the security and reliability of financial systems.

# Business Description

## Hypothesis

Our hypothesis is that machine learning algorithms can accurately detect fraudulent financial transactions by analyzing patterns and anomalies within the data. We believe that, by leveraging historical transaction data, these algorithms can learn to distinguish between legitimate and fraudulent transactions based on various features such as transaction amount, type, and account balances (Manchanda, 2023).

## General Goal

The general goal of this project is to develop and evaluate a machine learning model that can detect fraudulent financial transactions with high accuracy. Specifically, we aim to:

1. Collect and preprocess a dataset of financial transactions.
2. Explore and analyze the data to identify key patterns and features.
3. Train and optimize multiple machine learning models to classify transactions as fraudulent or non-fraudulent.
4. Evaluate the performance of these models using appropriate metrics.
5. Select the best-performing model and provide recommendations for its deployment in real-world scenarios (Jensen, 2012).

## Success Criteria/Indicators

The success of this project will be measured using several key performance indicators (KPIs):

1. **Accuracy**: The proportion of correctly identified transactions (both fraudulent and non-fraudulent) out of the total transactions.
2. **Precision**: The proportion of correctly identified fraudulent transactions out of all transactions identified as fraudulent by the model.
3. **Recall**: The proportion of correctly identified fraudulent transactions out of all actual fraudulent transactions.
4. **F1-Score**: The harmonic mean of precision and recall, providing a single metric that balances both aspects.
5. **ROC-AUC Score**: The area under the receiver operating characteristic curve, indicating the model's ability to distinguish between fraudulent and non-fraudulent transactions.
6. **Cross-Validation**: The model's performance will be validated using cross-validation techniques to ensure its robustness and generalizability.

Achieving high scores in these metrics, particularly a precision and recall above 85%, will be considered a success (scikit-learn, 2009). Additionally, the model should demonstrate consistent performance across different subsets of the data, indicating its reliability for real-world application.

By meeting these success criteria, we aim to develop a machine learning model that not only detects fraudulent transactions with high accuracy but also minimizes false positives and false negatives, thereby enhancing the security and trustworthiness of financial systems.

# **Technologies Used**

## **Libraries**

The following Python libraries were used for data processing, model building, and evaluation:

* **Pandas:** For data manipulation and analysis. It provides data structures and functions needed to manipulate numerical tables and time series. Pandas is particularly useful for handling large datasets like ours, with its efficient data frames and series.
* **NumPy:** For numerical computing. It offers support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is essential for performing mathematical operations required in feature engineering and model calculations.
* **Scikit-Learn:** For machine learning algorithms, model evaluation, and hyperparameter tuning. It provides simple and efficient tools for data mining and data analysis, making it a cornerstone library in this project. Scikit-Learn was used for implementing Logistic Regression, Random Forest, and for managing the workflow of machine learning tasks.
* **Imbalanced-learn (imblearn):** For handling imbalanced datasets, specifically using techniques like SMOTE (Synthetic Minority Over-sampling Technique). This library is crucial for addressing the significant class imbalance in our dataset, where fraudulent transactions are much fewer than non-fraudulent ones.
* **Matplotlib and Seaborn:** For data visualization to understand the distribution and relationships within the data. These libraries help in creating visual representations that are essential for exploratory data analysis and interpreting model results.

# Models

Three machine learning models were selected for this study: Logistic Regression, Random Forest, and XGBoost. Each model has distinct characteristics, advantages, and disadvantages that make them suitable for different aspects of the fraud detection problem.

## Logistic Regression

**Logistic Regression** is a linear model for binary classification. It estimates the probability that an instance belongs to a particular class.

* **Advantages:**
  + **Interpretability:** Provides clear insights into the importance of each feature through coefficients. This is crucial in fraud detection where understanding why a transaction is classified as fraudulent is important.
  + **Simplicity:** Easy to implement and computationally efficient, making it suitable for large datasets.
  + **Baseline Model:** Serves as a strong baseline to compare more complex models. It’s often the first model to try when approaching a classification problem.
* **Disadvantages:**
  + **Linearity Assumption:** Assumes a linear relationship between features and the log odds of the target variable, which may not capture complex patterns inherent in fraudulent activities.
  + **Overfitting:** Can overfit in the presence of many correlated features or noise. This necessitates careful feature selection and regularization.

## Random Forest

**Random Forest** is an ensemble learning method that constructs multiple decision trees and merges them to get a more accurate and stable prediction.

* **Advantages:**
  + **Robustness:** By averaging multiple decision trees, Random Forest reduces overfitting and improves generalization. This is beneficial in dealing with the noise and variability in transaction data.
  + **Feature Importance:** Provides measures of feature importance, helping identify which features contribute most to the prediction. This insight can be useful for understanding fraud patterns.
  + **Non-linearity:** Capable of capturing non-linear relationships between features, making it suitable for complex datasets where interactions between features are not purely linear.
* **Disadvantages:**
  + **Complexity:** The model can become computationally intensive with a large number of trees and deep trees, leading to increased training time.
  + **Interpretability:** Less interpretable compared to Logistic Regression, as the model’s predictions are based on the majority voting of many decision trees.

## XGBoost

**XGBoost** (Extreme Gradient Boosting) is a scalable and efficient implementation of gradient boosting algorithms. It builds trees sequentially, each one correcting the errors of its predecessor.

* **Advantages:**
  + **Performance:** Consistently outperforms many other algorithms in various machine learning competitions and practical applications. Its ability to handle large datasets and complex patterns makes it particularly powerful.
  + **Handling Imbalance:** Includes features like **scale\_pos\_weight** to handle class imbalance effectively, which is critical in our fraud detection task.
  + **Speed and Efficiency:** Optimized for speed and performance, leveraging parallel processing and regularization to prevent overfitting.
* **Disadvantages:**
  + **Complexity:** The model can be complex to tune and requires more computational resources compared to simpler models like Logistic Regression.
  + **Interpretability:** While more interpretable than some black-box models, it is still less transparent than simpler models. The complexity of boosting trees makes it harder to understand the contribution of individual features.

**Hyperparameter Tuning and Cross-Validation**

**Hyperparameter Tuning** involves searching for the best parameters for a model to maximize its performance. This is crucial because the default parameters might not be optimal for the specific characteristics of the dataset.

**Grid Search Cross-Validation (GridSearchCV)** is a technique that systematically works through multiple combinations of parameter tunes, cross-validates each combination, and determines which tune gives the best performance.

* **Advantages:**
  + **Systematic Search:** GridSearchCV ensures that all possible combinations of hyperparameters are tested, providing a thorough search.
  + **Cross-Validation:** By using cross-validation, it helps in ensuring that the model generalizes well to unseen data and is not just overfitting to the training set.
* **Disadvantages:**
  + **Computationally Intensive:** Testing all possible combinations can be very time-consuming and requires significant computational resources, especially with large datasets.
  + **Simplification Needed:** Given the constraints of a local machine with limited computational power, it is often necessary to simplify the grid to ensure the code runs in a reasonable time frame.

**Evaluation Metrics and Techniques**

**Evaluation Metrics** are used to compare and assess the performance of the models. The following metrics were used in this project:

* **Accuracy:** The proportion of true results (both true positives and true negatives) among the total number of cases examined. However, accuracy can be misleading in imbalanced datasets, so other metrics are also considered.
* **Precision:** The number of true positive results divided by the number of all positive results, including those not correctly identified. It measures the accuracy of the positive predictions.
* **Recall (Sensitivity):** The number of true positive results divided by the number of positives that should have been identified. It measures the model's ability to identify positive instances.
* **F1-Score:** The harmonic mean of precision and recall. It provides a balance between precision and recall, especially useful in imbalanced datasets.
* **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):** Measures the ability of the model to distinguish between classes. A higher AUC indicates a better model performance.

**Techniques Used for Comparison:**

* **Cross-Validation:** As mentioned, cross-validation is used to ensure the model generalizes well to unseen data. By dividing the data into multiple folds and training/testing on each fold, it provides a robust estimate of model performance.
* **Confusion Matrix:** Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives. This helps in understanding the types of errors the model is making.

# Exploratory Data Analysis (EDA)

This capstone provides a detailed analysis of the exploratory data analysis (EDA) performed on the dataset of financial transactions. The goal is to understand the data's structure, identify patterns, and prepare for further machine learning modeling.

Exploratory Data Analysis (EDA) is an essential step in any data science project. It involves summarizing the main characteristics of the data, often using visual methods. According to Tukey (1977), EDA focuses on discovering patterns and anomalies, checking assumptions, and selecting appropriate models.

## Loading and Preparing the Data

The dataset was loaded successfully, and the first few rows were displayed to ensure the data was read correctly. This step is crucial for verifying the structure and content of the dataset, including the column names and data types.

**Result:**

* The dataset contains columns such as **step**, **type**, **amount**, **nameOrig**, **oldbalanceOrg**, **newbalanceOrig**, **nameDest**, **oldbalanceDest**, **newbalanceDest**, **isFraud**, and **isFlaggedFraud**.

## Distribution of Transactions by Type

A bar plot was created to show the distribution of transactions by type. The plot displays the count of each transaction type, with the exact count shown above each bar.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figure 1 - Transaction Distribution by type

**Result:**

* **Common Transaction Types:**
  + **CASH\_OUT**: 2237500 transactions
  + **PAYMENT**: 2151495 transactions
  + **CASH\_IN**: 1399284 transactions
  + **TRANSFER**: 532909 transactions
  + **DEBIT**: 41432 transactions
* **Visualization:** The bars represent the number of transactions for each type, with **CASH\_OUT** transactions being the most frequent. This distribution indicates that **CASH\_OUT** and **PAYMENT** transactions are the predominant type in the dataset and **DEBIT** is the less predominant.

## Distribution of Transaction Amounts

A histogram was plotted to show the distribution of transaction amounts, including a KDE (Kernel Density Estimate) line for smooth density estimation.

Gráfico, Histograma

Descrição gerada automaticamente

Figure 2 - Transaction distribution by frequency x amount

**Result:**

* **Transaction Amounts:** Most transactions have amounts clustered below 5,000 units.
* **Outliers:** There are some outliers with transaction amounts exceeding 200,000 units, which are relatively rare.
* **Density Distribution:** The KDE line indicates the probability density function of the transaction amounts, showing where most data points are concentrated.

## Correlation Analysis between Variables

A heatmap of the correlation matrix was created to visualize the correlations between pairs of variables.

Uma imagem contendo Gráfico

Descrição gerada automaticamente

Figure 3 - Correlation Matrix

**Result:**

* **Strong Correlations:**
  + **oldbalanceOrg** and **newbalanceOrig**: Correlation of 0.99, indicating a very strong positive relationship and the opposite as well but it is for obvious reason, and not can used to indicate fraud.

**oldbalanceDest** and **newbalanceDest**: Correlation of 0.97.the same explained above.

* **Weak Correlations:**
  + **amount** and other variables generally show weak correlations.
* **Negative Correlations:** Negative correlations are minimal or non-existent in this dataset, indicating that most relationships are either positive or neutral.

## Distribution of Fraudulent and Non-Fraudulent Transactions

A bar plot was created to show the distribution of fraudulent and non-fraudulent transactions. The plot includes the count of transactions in each category.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figure 4 - Distribution of Fraudlant x Non Fraudlant Transaction

**Result:**

* **Class Imbalance:** The dataset is highly imbalanced, with non-fraudulent transactions vastly outnumbering fraudulent ones.
* **Counts:**
  + Non-Fraudulent: 6354407 transactions
  + Fraudulent: 8212 transactions
* **Percentage**: Approximately 0.1290% of the transactions are fraudulent.
* **Visualization:** This imbalance needs to be addressed during the modeling phase to avoid biased predictions.

## Descriptive Statistics of Variables

Descriptive statistics of the dataset were calculated to provide a summary of the central tendency, dispersion, and shape of the data distribution.

**Result:**

* **Central Tendency:**
  + **Amount:** Mean = 179,861.9 units, Median = 74,871.94 units
  + **oldbalanceOrg:** Mean = 833,883.1 units, Median = 14,208.00 units
  + **newbalanceOrig:** Mean = 855,113.7 units, Median = 0.00 units
  + **oldbalanceDest:** Mean = 1,100,702.0 units, Median = 132,705.70 units
  + **newbalanceDest:** Mean = 1,224,996.0 units, Median = 214,661.4 units
* **Dispersion:**
  + **Amount:** Standard Deviation = 603,858.2 units
  + **oldbalanceOrg:** Standard Deviation = 2,888,243.0 units
  + **newbalanceOrig:** Standard Deviation = 2,924,049.0 units
  + **oldbalanceDest:** Standard Deviation = 3,399,180.0 units
  + **newbalanceDest:** Standard Deviation = 3,674,129.0 units
* **Distribution Shape:** The high standard deviation values indicate a wide range of transaction values, with significant variability. This is also evident in the minimum and maximum values showing large outliers. The minimum values for **oldbalanceOrg**, **newbalanceOrig**, **oldbalanceDest**, and **newbalanceDest** are all 0, while the maximum values are extremely high, especially **newbalanceDest** at approximately 356,015,900 units.

**Summary of EDA**

The exploratory data analysis (EDA) provides valuable insights into the structure and characteristics of the financial transactions dataset:

1. **Transaction Types:** **CASH\_OUT** transactions are the most common, followed by **PAYMENT** and **TRANSFER**.
2. **Transaction Amounts:** Most transactions have lower amounts, with some high-value outliers.
3. **Correlations:** Strong positive correlations exist between initial and final balances of both origin and destination accounts.
4. **Class Imbalance:** The dataset is highly imbalanced, with far fewer fraudulent transactions (0.1290%) compared to non-fraudulent ones.
5. **Descriptive Statistics:** The dataset shows significant variability in transaction amounts and balances.

These insights will guide me to the next steps in data preprocessing, feature selection, and model building to detect fraudulent transactions effectively. Addressing the class imbalance and selecting relevant features based on the correlation analysis will be crucial for building robust and accurate machine learning models.

**Data Preparation**

The data preparation process included several important steps to ensure the dataset was ready for building machine learning models. Below is an explanation of each result from the code:

**Handling Missing Values**

**Results:** step 0 type 0 amount 0 nameOrig 0 oldbalanceOrg 0 newbalanceOrig 0 nameDest 0 oldbalanceDest 0 newbalanceDest 0 isFraud 0 isFlaggedFraud 0 dtype: int64

**Explanation:** The output shows that there are no missing values in any of the columns. Each column has 0 missing values, meaning we do not need to perform any additional steps to handle missing data. This ensures that all rows in the dataset are complete and can be used for analysis and modeling.

**Original Dataset Shape**

**Results:** Original dataset shape: 0 6354407 1 8213 Name: isFraud, dtype: int64

**Explanation:** This result shows the distribution of non-fraudulent and fraudulent transactions in the original dataset:

* There are 6,354,407 non-fraudulent transactions.
* There are 8,213 fraudulent transactions.

The dataset is highly imbalanced, with fraudulent transactions making up only a small fraction of the total transactions. This imbalance can affect the performance of machine learning models, making it important to address this issue during data preparation.

**Resampled Dataset Shape**

**Result:** Resampled dataset shape: 0 6354407 1 6354407 Name: isFraud, dtype: int64

**Explanation:** After applying SMOTE (Synthetic Minority Over-sampling Technique), the dataset has been balanced:

* There are now 6,354,407 non-fraudulent transactions.
* There are now 6,354,407 fraudulent transactions.

SMOTE generated synthetic samples for the minority class (fraudulent transactions) to match the number of samples in the majority class (non-fraudulent transactions). This resampling helps to balance the classes, which is crucial for training machine learning models that can effectively identify fraudulent transactions without being biased towards the majority class.

## Detailed Explanation and Comparison of Model Results

## Logistic Regression

**Best Hyperparameters:**

* **C:** 10
* **penalty:** 'l1'
* **solver:** 'liblinear'

The best hyperparameters for the Logistic Regression model indicate a high regularization strength with **C** set to 10, using an **l1** penalty and the **liblinear** solver. The **l1** penalty helps in feature selection by enforcing sparsity, while the **liblinear** solver is efficient for small datasets.

**Classification Report:**

* **Precision:** The proportion of true positive predictions out of the total positive predictions.
  + Class 0 (Non-Fraudulent): 0.92
  + Class 1 (Fraudulent): 0.93
* **Recall:** The proportion of true positive predictions out of the actual positive instances.
  + Class 0 (Non-Fraudulent): 0.93
  + Class 1 (Fraudulent): 0.92
* **F1-Score:** The harmonic mean of precision and recall, providing a balance between the two metrics.
  + Class 0 (Non-Fraudulent): 0.93
  + Class 1 (Fraudulent): 0.93
* **Support:** The number of actual occurrences of the class in the dataset.
  + Class 0 (Non-Fraudulent): 29,830
  + Class 1 (Fraudulent): 30,101

**Confusion Matrix:**

* **True Negatives (TN):** 27,852
* **False Positives (FP):** 1,978
* **False Negatives (FN):** 2,337
* **True Positives (TP):** 27,764

**ROC-AUC Score:** 0.9762896410556914

The Logistic Regression model achieves a high accuracy of 93%, with balanced precision and recall for both classes. The ROC-AUC score of 0.976 indicates excellent performance in distinguishing between fraudulent and non-fraudulent transactions.

## Random Forest

**Best Hyperparameters:**

* **max\_depth:** 20
* **n\_estimators:** 100

The best hyperparameters for the Random Forest model suggest a maximum tree depth of 20 and 100 trees in the forest. This configuration helps capture complex patterns in the data while preventing overfitting.

**Classification Report:**

* **Precision:**
  + Class 0 (Non-Fraudulent): 1.00
  + Class 1 (Fraudulent): 1.00
* **Recall:**
  + Class 0 (Non-Fraudulent): 1.00
  + Class 1 (Fraudulent): 1.00
* **F1-Score:**
  + Class 0 (Non-Fraudulent): 1.00
  + Class 1 (Fraudulent): 1.00
* **Support:**
  + Class 0 (Non-Fraudulent): 29,830
  + Class 1 (Fraudulent): 30,101

**Confusion Matrix:**

* **True Negatives (TN):** 29,814
* **False Positives (FP):** 16
* **False Negatives (FN):** 1
* **True Positives (TP):** 30,100

**ROC-AUC Score:** 0.9999984903879813

The Random Forest model demonstrates near-perfect performance with an accuracy of 100%. The ROC-AUC score is almost 1, indicating the model can perfectly distinguish between classes. However, such high performance might suggest overfitting, as it performs flawlessly on the test set.

## **XGBoost**

**Best Hyperparameters:**

* **learning\_rate:** 0.1
* **max\_depth:** 5
* **n\_estimators:** 100

The best hyperparameters for the XGBoost model include a learning rate of 0.1, a maximum depth of 5, and 100 estimators. These settings balance model complexity and learning speed, allowing XGBoost to efficiently learn patterns in the data.

**Classification Report:**

* **Precision:**
  + Class 0 (Non-Fraudulent): 1.00
  + Class 1 (Fraudulent): 1.00
* **Recall:**
  + Class 0 (Non-Fraudulent): 1.00
  + Class 1 (Fraudulent): 1.00
* **F1-Score:**
  + Class 0 (Non-Fraudulent): 1.00
  + Class 1 (Fraudulent): 1.00
* **Support:**
  + Class 0 (Non-Fraudulent): 29,830
  + Class 1 (Fraudulent): 30,101

**Confusion Matrix:**

* **True Negatives (TN):** 29,780
* **False Positives (FP):** 50
* **False Negatives (FN):** 2
* **True Positives (TP):** 30,099

**ROC-AUC Score:** 0.9999256731858926

The XGBoost model also achieves near-perfect performance with an accuracy of 100% and a ROC-AUC score close to 1. Like the Random Forest model, this high performance might indicate overfitting.

## **Comparison of Models**

1. **Logistic Regression:**
   * **Pros:** High interpretability, balanced performance, and excellent ROC-AUC score.
   * **Cons:** Slightly lower performance compared to ensemble methods, potentially due to its linear nature.
2. **Random Forest:**
   * **Pros:** Near-perfect classification performance, robustness to overfitting due to ensemble averaging.
   * **Cons:** Potential overfitting as indicated by flawless performance, longer training time, and less interpretability compared to Logistic Regression.
3. XGBoost:
   * **Pros:** Near-perfect classification performance, efficient handling of large datasets, and superior ability to capture complex patterns.
   * **Cons:** Potential overfitting, higher computational cost, and complexity in tuning hyperparameters.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figure 5 - Model Comparison

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 6 - ROC Curvas

**Best Model:** While both Random Forest and XGBoost achieved near-perfect performance, their results suggest possible overfitting. Logistic Regression, on the other hand, demonstrated high accuracy and a strong ROC-AUC score with more balanced metrics, indicating good generalization without overfitting. Therefore, **Logistic Regression** may be considered the best model due to its balance of performance, interpretability, and computational efficiency. However, further validation on additional test datasets and real-world scenarios is recommended to confirm the findings and ensure robustness in deployment.

# Conclusion:

The goal of this study was to develop and evaluate machine learning models to detect fraudulent financial transactions using a dataset sourced from Kaggle. The study followed the Cross-Industry Standard Process for Data Mining (CRISP-DM), encompassing stages from business understanding to deployment. This approach ensured a structured and comprehensive analysis, aligning the project with industry best practices.

The dataset contained approximately 6.3 million transactions with a severe class imbalance, having only 0.129% of transactions marked as fraudulent. Addressing this imbalance was crucial to ensure the machine learning models could generalize well and accurately identify fraudulent activities without being biased towards the majority class.

Three machine learning models were selected for this study: Logistic Regression, Random Forest, and XGBoost. Each model was evaluated using various metrics, including accuracy, precision, recall, F1-score, and ROC-AUC score, to determine their effectiveness in detecting fraudulent transactions.

**Detailed Analysis of Results**

**Logistic Regression**

The Logistic Regression model achieved a high accuracy of 93%, with balanced precision and recall for both classes. The ROC-AUC score of 0.976 indicates excellent performance in distinguishing between fraudulent and non-fraudulent transactions. This model's interpretability and simplicity make it a strong baseline for comparison.

**Random Forest**

The Random Forest model demonstrated near-perfect performance with an accuracy of 100%. The ROC-AUC score is almost 1, indicating the model can perfectly distinguish between classes. However, such high performance might suggest overfitting, as it performs flawlessly on the test set. While Random Forest provides robustness and feature importance insights, its complexity and potential overfitting are concerns.

**XGBoost**

* **Best Hyperparameters:** {'learning\_rate': 0.1, 'max\_depth': 5, 'n\_estimators': 100}
* **Classification Report:**

The XGBoost model also achieved near-perfect performance with an accuracy of 100% and a ROC-AUC score close to 1. Like the Random Forest model, this high performance might indicate overfitting. XGBoost's ability to handle large datasets and capture complex patterns is advantageous, but its complexity and computational cost are drawbacks.

**Comparison and Recommendation**

When comparing the three models, it's essential to consider both performance metrics and practical deployment aspects:

1. **Performance Metrics:**
   * **Accuracy and ROC-AUC:** Random Forest and XGBoost achieved near-perfect scores, indicating excellent performance in distinguishing between classes.
   * **Precision, Recall, and F1-Score:** Both Random Forest and XGBoost demonstrated perfect scores, whereas Logistic Regression showed slightly lower but still high and balanced metrics.
2. **Overfitting Concerns:**
   * The perfect performance of Random Forest and XGBoost on the test set raises concerns about overfitting. While these models performed flawlessly, it is crucial to validate their performance on additional datasets to ensure they generalize well to unseen data.
3. **Model Interpretability:**
   * Logistic Regression offers high interpretability, making it easier to understand and explain the model's decisions. This is a significant advantage in financial fraud detection, where transparency and understanding the reasons behind predictions are important.
4. **Computational Efficiency:**
   * Logistic Regression is computationally efficient and suitable for large datasets. In contrast, Random Forest and XGBoost require more computational resources and time, which might be a limitation for real-time applications.

**Best Model:** Based on the balanced performance metrics, interpretability, and computational efficiency, **Logistic Regression** may be considered the best model for detecting fraudulent financial transactions in this study. While Random Forest and XGBoost show excellent performance, their potential overfitting and higher computational costs are concerns. Logistic Regression provides a robust and interpretable solution, making it suitable for deployment in real-world scenarios.

**Future Recommendations**

1. **Further Validation:**
   * Validate the models on additional datasets to ensure their robustness and generalizability. This step is crucial to confirm that the models perform well on unseen data and are not overfitting to the current dataset.
2. **Model Ensemble:**
   * Explore ensemble methods that combine the strengths of multiple models. For example, a voting classifier that integrates Logistic Regression, Random Forest, and XGBoost might achieve better overall performance.
3. **Feature Engineering:**
   * Investigate additional feature engineering techniques to capture more complex patterns in the data. Incorporating domain-specific knowledge can enhance the model's ability to detect fraud.
4. **Real-time Implementation:**
   * Optimize the models for real-time deployment by reducing computational complexity and ensuring efficient processing. This step is critical for practical applications where quick and accurate fraud detection is required.
5. **Continuous Monitoring:**
   * Implement a system for continuous monitoring and updating the models to adapt to new fraud patterns. Fraudsters' techniques evolve over time, and the models need to be updated regularly to maintain their effectiveness.

**References**

* Jensen, K. (2012). English: A diagram showing the relationship between the different phases of CRISP-DM and illustrates the recursive nature of a data mining project. Wikimedia Commons. Available at: <https://commons.wikimedia.org/wiki/File:CRISP-DM_Process_Diagram.png> [Accessed 30 Sep. 2020].
* Manchanda, C. (10AD). Fraudulent Transactions Data. Kaggle. Available at: <https://www.kaggle.com/datasets/chitwanmanchanda/fraudulent-transactions-data>.
* scikit-learn (2009). 1.10. Decision Trees — scikit-learn 0.22 documentation. Scikit-learn.org. Available at: <https://scikit-learn.org/stable/modules/tree.html>.
* Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95.
* McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51-56.
* Oliphant, T. E. (2006). A guide to NumPy (Vol. 1, p. 85). USA: Trelgol Publishing.
* Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.
* Tukey, J.W. (1977). Exploratory Data Analysis. Addison-Wesley.