# Project Note: Credit Card Fraud

# List of Contents

# List of Tables

## Version

| Version | Changes |
|---|---|
| V0 | The first version of the report |

Table 1: Version of the Report

# 1 Introduction

## 1.1 Background

A credit card is an electronic payment tool that uses a card issued by a bank or financial institution to make transactions. This card helps customers make payments with loans from banks and the user will pay off the debt with the nominal and time period determined by the bank. With a credit card, customers can make an easy and immediate transaction. In addition, the bank will give rewards in the form of points, the amount of which is determined based on the number of transactions made.

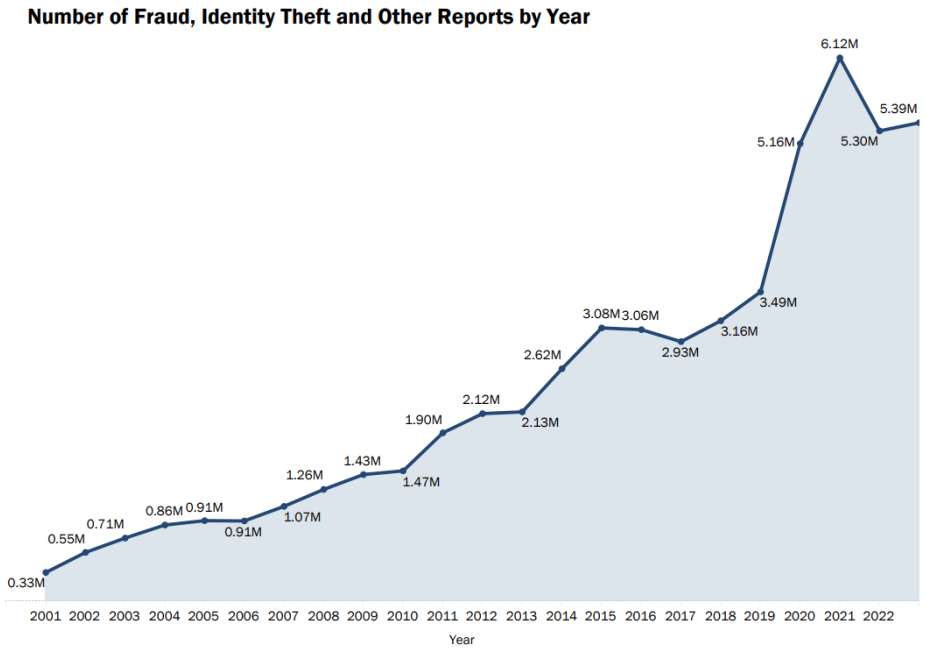



Figure 1: Credit Card Fraud Reports by Year (FTC Annual Data Book, 2023)

Transactions using credit cards have advantages that are quite attractive to customers. However, these transactions are not completely secured. Some credit card users have experienced bad things like fraud or identity theft.Based on **The Federal Trade Commision (FTC)** Annual Data Book pada 2023 that also shown in the Figure 1, fraud problems increased sharply from 2001 to 2021 by around 6 million customers and only slightly dropped by 700 thousand cases in the last three years. In addition, around 48% of the 5.4 million reports recorded were scam complaints, the rest were identity theft (19%) and other forms of fraud (33.6 %). Even though the recorded fraud is not that large compared to the number of credit card users, the total loss from this crime

reaches 10 billion USD. Fraud is a crime which is detrimental to both parties. Customers will unknowingly lose the money they have saved and the bank will suffer losses because it will not only have to compensate customers for their losses, but also lose the trust of its customers.

## 1.2 Goals

The goal of this project is:

1. Identifying the peak periods of fraud risk and quantifying the resulting financial losses.
2. Comparing machine learning models with 3 datasets that go through 3 preprocessing techniques, raw, imputation, and isolation forest.

## 1.3 Limitations

This machine learning project cannot be applied to companies or banks because the dataset used is a sample obtained from several banks in Belgium and has not been updated for more than 5 years. Futhermore, the research conducted did not involve experts or professionals who are experienced in the financial field, so the results obtained tend to be biased. Third, not all open-source, free models are tested here due to computing power limitations.

## 1.4 Tools

The library and its version we used in this project:

1. python 2.2.2
2. seaborn 0.13.2
3. matplotlib 3.10.0
4. scikit-learn 1.6.1
5. numpy 2.0.2
6. pandas 2.2.2
7. imbalanced-learn 0.13.0

# 2 Dataset and Features

## 2.1 Data Description

The data has been used, analyzed, and publshied by Machine Learning Group Université Libre de Bruxelles and this data were obtained free from Kaggle. The data contains:

1. The Time Transaction
2. Amount of Transasction
3. The Customer identity that had been changed
4. Class

| | Columns | Dtype | Role |
|---|---|---|---|
| 1 | Time | Int | Feature |
| 2 | Amount | float | feature |
| 3 | V1 - V34 | float | feature |
| 4 | Class | Boolean | label |

Table 2: Beispieltabelle

**Class** is a column that contains an indication of whether a transaction is fraudulent or not. Transactions that are indicated as fraudulent will be marked with the number 1, the rest with 0. The **Amount** feature denotes as the nominal amount used in the recorded transaction. Features **V1** to **V34** are samples of credit card user identity data such as name, credit card number, address, mobile phone number, Card Verification Code (CVC) or Card Verification Value (CVV), issue number, and other sensitive information that are transformed by the data provider using the Principle Component Analysis (PCA) method for security reasons.

# 3 Theory

## 3.1 Principle Component Analysis

PCA is an unsupervised machine learning algorithm that reduces high-dimensional input datasets (large number of features) with a series of specific processes. The technique can also filter out which features are interdependent or not related at all. (Strang, 2023). Columns V1 to V34 are data that initially consisted of the credit card user's identity which was then changed using the PCA method. This method is used to reduce columns in high-dimensional datasets (which have many columns and indexes). Table 3 will be an example demonstrating how the PCA works.

| Name | Age | Height (cm) | Weight (kg) | BMI |
|---|---|---|---|---|
| Participant 1 | 40 | 180 | 80 | 24.7 |
| Participant 2 | 32 | 177 | 48 | 15.3 |
| Participant 3 | 50 | 165 | 65 | 23.9 |
| Participant 4 | 38 | 174 | 68 | 22.5 |

Table 3: Table for PCA Example

PCA can be applied in various scientific fields, one of which is historical science. Researchers found and analyzed 88 samples of early insectivore tooth fossils of the mammal Kuehneotherium. These teeth are molars that are 200 billion years old. They used four bat species as a reference for comparison because they were not only well-studied of dietary differences, but also had a size that was almost similar to Kuehneotherium. In the world of bats, the nine roughness parameters differ greatly by species, and PCA separates each bat based on their daily food preferences. The result is that the components of axis 1 and axis 2 have values of 48 % and 40% respectively (with p-value ¡ 0.0001).If the value is more positive, the higher the proportion of "soft" prey, and vice versa (Gill et al., 2014).

The Zhou et al., 2019 research used the technique for data processing and two machine learning models, imported Ada Boost and single-layer decision tree, making the prediction accuracy rate reach 96.5% with the F-measure (F1 score), a measure of how well a machine learning model classifies data, reaching 97.3%. There are five steps in the PCA method, standardization, covariant metrics calculation, calculate the values and eigenvectors of the covariance matrix, feature vector, and recast.

### 3.1.1 Standardization

**Standardization is start line** of the whole process and will use the z-score method. This statistical method is a way to see how many data points are far from the expected value or average value $\mu$. In machine learning, This method can be used to assess the significance of the prediction model. The larger the z-score value (positive or negative) indicates that the prediction will be far from the average. This process is quite important if the model to be used depends on distance, such as Support Vector Machine and Principle Component Analysis. Z-score standardization can be used with the equation

$$Z = \frac{X - \mu}{\sigma} \tag{1}$$

where $X$, $\mu$, and $\sigma$ are features (variables), expectation value or average (mean) each feature, and standard deviation (std). After standardization process, each feature has $\mu = 0$ with standard deviation $\sigma = 0$ and turn into the Table 4

| Name | Age | Height | Weigth | BMI |
|---|---|---|---|---|
| Participant 1 | 0 | 1.07 | 1.29 | 0.83 |
| Participant 2 | -1.23 | 0.53 | -1.5 | -1.69 |
| Participant 3 | 1.54 | -1.60 | -0.02 | 0.62 |
| Participant 4 | -0.3 | 0 | 0.24 | 0.24 |

Table 4: Table After Z-score process

### 3.1.2 Covariant

The second stage of the Principle Component Analysis is **Calculating covariance**. **Covariance** is a measure of the combined variability or degree of association between two variables. The covariant calculation can be obtained by using 2.

$$cov(x_1, x_2) = \frac{(x_1 - \bar{x_1})(x_2 - \bar{x_2})}{n - 1} \tag{2}$$

where $x_1$ and $x_2$ are dataset variables, $\bar{x_1}$ and $\bar{x_2}$ denotes its mean, and $n$ is the number of observed index (Rice, 2007).

The covariant result usually forms a matrix according to the shaped of data that being processed. Since Table 3 is a 4 x 4 matrix, the calculation can be done by constructing the matrix below, and the results can be seen in Table 5.

$$\begin{bmatrix} cov(x_1, y_1) & cov(x_1, y_2) & cov(x_1, y_3) & cov(x_1, y_4) \\ cov(x_2, y_1) & cov(x_2, y_2) & cov(x_2, y_3) & cov(x_2, y_4) \\ cov(x_3, y_1) & cov(x_3, y_2) & cov(x_3, y_3) & cov(x_3, y_4) \\ cov(x_4, y_1) & cov(x_4, y_2) & cov(x_4, y_3) & cov(x_4, y_4) \end{bmatrix}.$$

| ... | Age | Height | Weight | BMI |
|---|---|---|---|---|
| Age | 1.33 | -1.04 | 0.58 | 0.98 |
| Height | -1.04 | 1.33 | 0.202 | -0.33 |
| Weight | 0.58 | 0.202 | 1.33 | 1.22 |
| BMI | 0.98 | -0.33 | 1.22 | 1.33 |

Table 5: Table After Covariance Calculation

### 3.1.3 Eigenvalue and Eigenvector

Eigenvalues are a special set of scalars associated with a linear equation, whereas eigenvectors are non-zero vectors that do not change when a linear transformation is applied. The word 'eigen' itself means 'characteristic' or 'proper'. The calculation of eigenvector and eigenvalue can be done by using equation 3 or 4.

$$\hat{A}X = \lambda X \tag{3}$$

$$(\hat{A} - \lambda I)X = 0 \tag{4}$$

where $\lambda$ is an eigenvalue, $\hat{A}$ represents the transformation operator, $I$ is identity matrix, and $X$ is an eigenvector. For a simple example, the operator $\hat{A}$.

$$\hat{A} = \begin{bmatrix} 5 & 2 & 10 \\ 4 & 3 & 1 \\ 3 & 8 & 4 \end{bmatrix}$$

multiplied by the matrix X

$$X = \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}$$

become

$$\begin{bmatrix} 5 & 2 & 10 \\ 4 & 3 & 1 \\ 3 & 8 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 19 \\ 25 \\ 30 \end{bmatrix}$$

The example above is an example of ordinary multiplication and cannot be considered as an eigenvector and eigenvalue because the vector $X$ is completely transformed into a completely different vector from before the transformation with the operator $\hat{A}$. The second instance use vector

$$X = \begin{bmatrix} -4 \\ -3 \\ 5 \end{bmatrix}$$

and an operator $\hat{A}$

$$\hat{A} = \begin{bmatrix} 22 & 16 & 0 \\ -9 & -2 & 0 \\ 5 & -10 & 0 \end{bmatrix}$$

By using equation $\hat{A}X = \lambda X$, the result will be

$$\begin{bmatrix} 22 & 16 & 0 \\ -9 & -2 & 0 \\ 5 & -10 & 0 \end{bmatrix} \begin{bmatrix} -4 \\ -3 \\ 5 \end{bmatrix} = \begin{bmatrix} -40 \\ 30 \\ -50 \end{bmatrix} = 10 \begin{bmatrix} -4 \\ 3 \\ -5 \end{bmatrix}$$

The example above is a correct eigenvector because, although it is affected by the operator $\hat{A}$, the vector $X$ does not change and the eigenvalue $\lambda = 10$ is formed. There are several ways to obtain eigenvectors and eigenvalues, one of which by using $det(\lambda I - \hat{A})$ or it can be called the characteristic equation (Strang, 2023). By using that math formulation, then the matrix 5 will be

$$\begin{bmatrix} 1.33 - \lambda_1 & -1.04 & 0.58 & 0.98 \\ -1.04 & 1.33 - \lambda_2 & 0.202 & -0.33 \\ 0.58 & 0.202 & 1.33 - \lambda_3 & 1.22 \\ 0.98 & -0.33 & 1.22 & 1.33 - \lambda_4 \end{bmatrix}$$

After calculation process, four eigenvalues were obtained $\lambda$

$$\begin{bmatrix} 3.44 & 1.83 & 0.053 & 0 \end{bmatrix}$$

If the eigenvalue is $\lambda < 1$, then the value is discarded. There are two

eigenvalues greater than one and then these values are used in rearranging the principal components. This can be obtained by the equation $(\hat{A} - \lambda I)X$. At the eigenvalue $\lambda = 3.44$, the eigenvector value is

$$\begin{bmatrix} -0.56 \\ 0.32 \\ -0.76 \\ 0.0095 \end{bmatrix}$$

### 3.2 Oversampling

Oversampling is a data augmentation technique by adding minority class data (fraud) to balance it with majority class data (non-fraud). An illustration of this technique can be seen in the Figure 2.

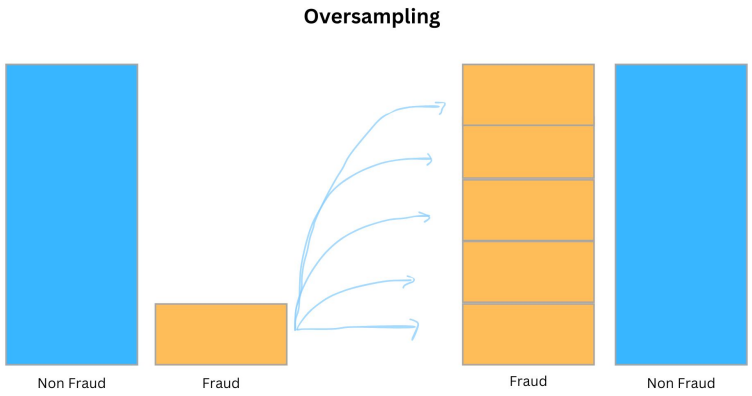



Figure 2: Oversampling

Oversampling have many variations, such as Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al, 2002), Support Vector Machine SMOTE (SVMSMOTE) (Tang et al, 2008), The Majority Weighted Minority Oversampling Technique (MWMOTE) (Barua et al, 2012), and etc. The benefit of this technique is model performance improvement and easy to applied. In Figure 3, publications on this technique are increasing year by year and are applied in fields such as health, finance, telecommunications, and so on. For instance, Chanda et al., 2023 utilizes a decision tree model with the SMOTE technique to predict credit cards.

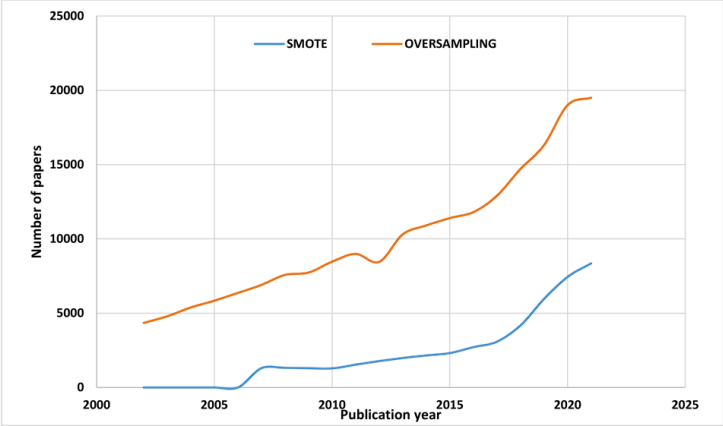



Figure 3: Oversampling Method Publication

One thing to note about using SMOTE is that, based on research from Alkhawaldeh et al., 2023 and Tarawneh et al., 2022, this technique is not recommended for datasets that are quite sensitive such as healthcare, cybersecurity, and several other vital objects because the model will experience overfitting and tend to create noisy data with large amounts. SMOTE is a technique developed by Chawla et al., 2002. According to the literature, this algorithm works in the following way:

1. Take a reference sample and its nearest neighbor
2. Multiply the distance difference by a random number from 0 to 1.
3. add this difference to the sample to create a new synthetic example in the feature space
4. repeat step 1 with a different reference sample

For example, looking at Figure 4, consider the first row (or any row in case the number of SMOTEs $N < 100$) and calculate the k nearest neighbors. Then, randomly select the nearest neighbor from the reference sample. After that, calculate the difference in distance between the two points (reference and one of the nearest neighbors) and multiply it by a random number from 0 to 1. This gives us additional synthetic instances along the distance between two points. If the data to be smote is 5000, then there are $\frac{5000}{100} = 50$ nearest neighbors connected to the reference point/data.

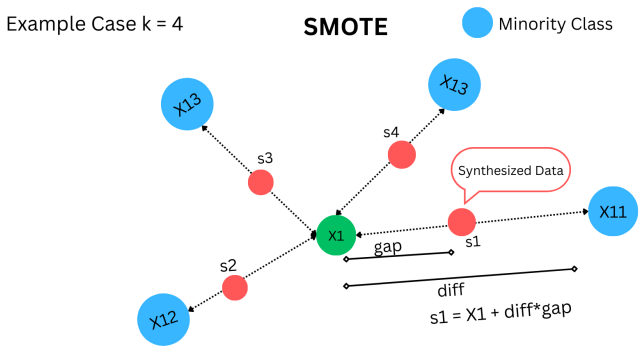



Figure 4: Demonstration of SMOTE

### 3.3 Machine Learning

Machine Learning is a subset or branch of artificial intelligence that focuses on developing machines that can learn and make predictions on patterns from data. This machine has three approaches, two of which are supervised learning and unsupervised learning. Supervised learning is a machine learning approach that uses labeled data as training to make predictions, while unsupervised is a machine learning approach that is used to analyze and group non-labeled data. The Supervised approach is divided into two types of tasks, regression and classification. Regression is a supervised learning method that uses algorithms to understand the relationship between dependent and independent variables, while classification is another supervised method that uses algorithms to assign data points to specific, pre-determined categories or classes (Volikatla, 2024). Unsupervised learning has three tasks, one of which is dimensionality reduction. The technique is used when the data to be processed has too many features or dimensions. Dimensionality reduction works like a filter, reducing the input data to a manageable size while maintaining the integrity of the dataset as much as possible. (Ciaburro, 2024).

Machine learning models are divided into two based on parameter values, namely parametric and non-parametric. A parametric model is a model that makes assumptions or predictions about mapping functions using fixed parameter values. This model will continue to use these parameter values even if the amount of data used becomes more and more complex. The advantages of parametric models are that they are simple, easy to interpret, and robust to small amounts of data. However, the models tends to produce biased as a result of data dimensionality and complexity.

A nonparametric model is the opposite of a parametric model - a model that does not use fixed parameter values to make assumptions of mapping function between input and output data. This model is free to make predictions without relying on fixed parameter values. Flexibility in terms of complex pattern mapping is the main advantage of this model. Nevertheless, due to the complex data and the absence of fixed parameter values, this model is difficult to interpret and takes a long time to make predictions (Triola, 2021). Some examples of non-parametric models are decision trees, random forests, and isolation forests.

### 3.3.1 Random Forest

Decision Tree is a non-parametric learning algorithm that forms trees that can be used for classification and regression tasks. The fundamental idea behind this model is to recursively divide the data into small sets (subsets) based on the values of different criteria (Rokach and Maimon, 2005, Mienye and Jere, 2024) and then generate. The process forms a hierarchical tree, as in Figure 5, consisting of a root node, branches, internal nodes, and leaf nodes. Each node represents a separate specific decision or attribute (Rivera-Lopez et al., 2022).

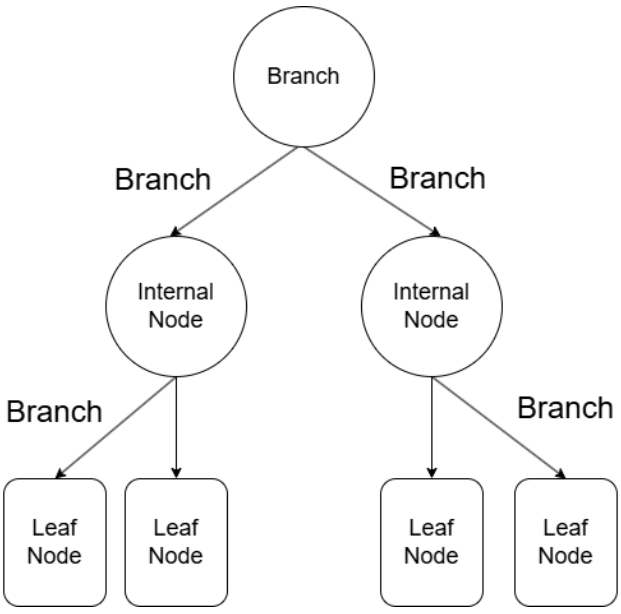



Figure 5: Decision Tree Diagram

Breimen, inspired by Amit and Geman, 1997's research on geometric feature selection, demonstrated ensembles of trees, where each tree in the en-

semble is grown based on its own parameters to improve the accuracy of classification and regression. Then, to get the final prediction, all the trees are combined (Bagging). This model is then called Random Forest, a collection of trees that each have their own parameters (Breiman, 1996, and Breiman, 2001). One modified model that adapts the tree algorithm is the Gradient Boost Decision Tree (GBDT). Basically, this model utilizes the error in the previous model - the value obtained from the difference between predictions and data - to improve model performance in the next training (Boosting). Unlike most models that focus on one model, this technique combines several weak models (usually decision trees) to form a model with higher accuracy (Freund, 1995, Freund and Schapire, 1996). The first step in using Gradient Boosting is to determine the initial value prediction $F_0$ with the equation 5.

$$F_0(x) = \arg\min_{\hat{y}} \sum_{i=1}^{n} L(y_i, \hat{y}) \tag{5}$$

where $L$ represent loss or cost function (e.g. mean absolute error, mean squared error, dan sebagainya), $y_i$ and $\hat{y}$ are actual and predicted value, respectively. The symbol arg min in the equation denotes prediction value $\hat{y}$ that can minimize the value of loss function. Previous predictions have errors that can be calculated using the residual score, which is the difference between the prediction and the actual value. Residuals for each data calculated by using Equation 6.

$$r_{mi} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}|_{F(x_i)=F_{m-1}(x_i)} \tag{6}$$

The equation calculates the residuals $r_{mi}$ by differentiating the Loss Function $L(y_i, F(x_i))$ from the previous prediction $F_{m-1}$, then multiplying it by -1. After getting the values, re-train the optimized model with those values and create terminal (leave) nodes $R_{jm}$ with total number of leaves $j = 1, 2, .., J_m$ in tree. Afterward, calculate the new prediction $\hat{y}$ by using equation 7

$$\hat{y}_{jm} = \arg\min_{\hat{y}} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \hat{y}) \tag{7}$$

for $j = 1, ...., J_m$. Last, update the prediction by using the new predicted values and the learning rate $\nu$ in the equation. 8.

$$F_m(x) = F_{m-1} + \nu \sum_{j=1}^{J_m} \hat{y}_{jm} \tag{8}$$

The $m$ denotes the tree index, whereas capital $M$ represents the total of tree.

The capital $J$ means the total of leave node. The value of learning rate $\nu$ between 0 and 1 (Friedman, 2000).

### 3.3.2 Feature Importances

Feature importances refers to a technique to calculate all input features on the model. A large score indicates that the feature has a large influence on the model's prediction. This technique is important because not only improve the performance of a model, but also allows us to understand what variables are irrelevant to the model.

The computation of the feature importances started from Gini Importances, an equation to calculate a node probability or node impurity. For instance, let us consider a tree with two nodes

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \tag{9}$$

where $ni_j$ denotes node importances, $w_j$ is weighted number of sample reaching node $j$, $C_j$ represents the impurity value of node $j$, $w_{left(j)}$ is weighted number of left node, and $w_{right(j)}$ denotes weighted number of right node. The equation 9 gives us importances of node j, which will be used to calculate the feature importance for each decision tree. The value of feature importance is obtained by the equation

$$fi_i = \frac{\sum_j ni_j}{\sum_{\in j \forall nodes} ni_j} \tag{10}$$

### 3.3.3 Confusion Matrix

One of the benchmarks of a classification model can be said to be good is by how many times it can guess something correctly. One of many ways use the Confusion Matrix table. Based on the Figure 6, the matrix consist of two basis, prediction and actual. In credit card fraud detection, a correct fraud prediction (True Positive) means the system successfully identified a fraudulent transaction, preventing potential financial loss. A correct legitimate prediction (True Negative) means the system correctly recognized a normal transaction as safe, ensuring smooth processing. The prediction of the model can be wrong sometimes. If the system mistakenly flags a legitimate transaction as fraud, it's called a False Positive—like a false alarm that causes an unnecessary security block. On the other hand, if the system incorrectly allows a fraudulent transaction to go through, that's a False Negative—meaning

it failed to catch the fraud in time.

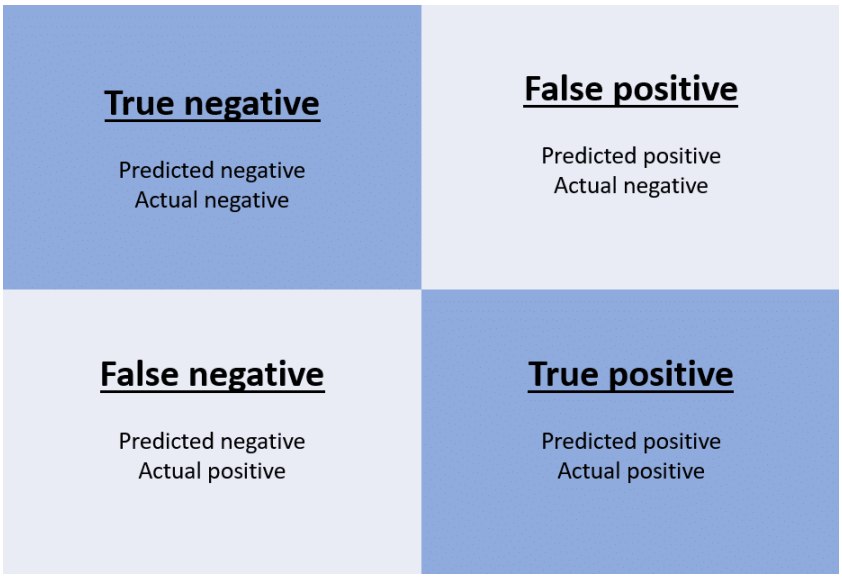



Figure 6: Confusion Matrix

### 3.3.4 Area Under the Receiver Operating Characteristic

Area under the Receiver Operating Characteristic (AUC-ROC) is a measure of how well a machine learning model is at separating two groups - such as identifying fraudulent transactions from legal ones. Imagine a security system that alerts you when something suspicious happens. If it's too sensitive, it might give false alarm;if it's too relaxed, it might miss real threats. The Receiving Operating Characteristic (ROC) curve visualize this balance, showing how well the model separates real cases from false guess. The Area Under Curve (AUC) gives a single score;the closer it is to 1, the better the model is making at correct predictions. A score near 0.5 means the model is guessing randomly.

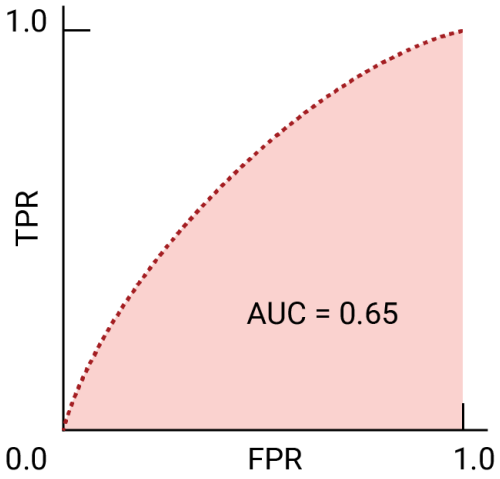



Figure 7: The AUC ROC Curve (Source: Google for Developer
)

Based on the Figure 7, the ROC curve consists of two components, False positive rate as x-axis and True positive rate as y-axis. True Positive Rate measure how well the model correctly identifies actual positives. Think of a medical test - if it correctly detects 90 out of 100 sick patients, the sensitivity is 90%. A higher value means fewer missed cases. TPR can be measured by equation 11.

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

where TP denotes True Positive and FN is False Negative (both terms has been explained in sub-section 3.3.3).

Specificity is part of the False Positive Rate (FPR). This shows how well a model avoids falsely labeling negative cases as positive. For instance, if a test correctly identifies 95 out of 100 healthly people as negative, its specificity is 95 %. A high specificity means fewer false alarms. It can be calculated using equation 12.

$$Specificity = \frac{TN}{TN + FN} \tag{12}$$

where TN denotes True Negative and FN is False Negatives. Meanwhile, the FPR is a measure how often the model incorrectly classifies as negatives as positives. For example, if 5 out of 100 healthy people are mistakenly diagnosed as sick, the false positive rate is 5%. A lower rate percentage means fewer

unnecessary alerts of false prediction. The FPR can be found using equation

$$FPR = 1 - Specificity. \tag{13}$$

### 3.4 Isolation Forest

Isolation Forest and Random Forest are two techniques that have similarities in the type of algorithm used, non-parametric ensemble trees. However, those methods has different framework and purpose. The random forest is an ensemble tree that is used to learn the data and make a prediction, whereas the isolation forest is used to detect and isolate the anomaly. This technique, introduced by Liu et al., 2008, works by randomly selecting two variables or features $\{x, y\}$ from the dataset, then comparing them with the range of values of the selected features, as in Figure 8.

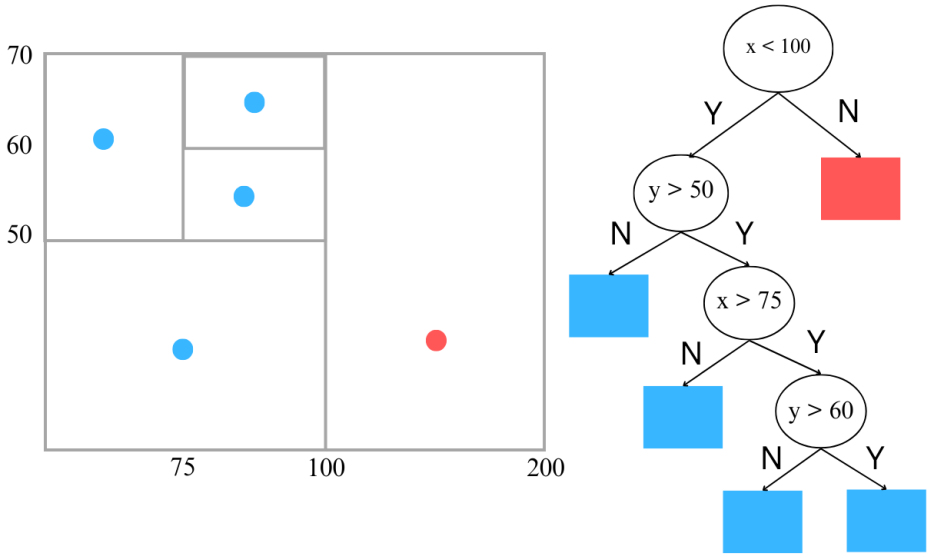



Figure 8: Isolation Forest

The anomaly of a data can be measured by the anomaly score. This value is obtained by the equation

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n}} \tag{14}$$

where $h(x)$ denotes number of splits or path of length, while $E(h(x))$ represent expectation value or average path of length.

## 3.5 Credit Card Fraud

Credit card fraud is when a user uses a credit card without the permission of the legal cardholder. Moreover, the illegal user has no relationship with the legal holder or the credit card issuing bank, and has no intention of returning it to the rightful owner. Credit card fraud is carried out in the following ways:

- Fraudulent acts using unauthorized accounts and/or personal information.
- using unauthorized accounts for personal gain.
- Misuse of account information to obtain goods and/or services.

Common fraud modus operandi includes lost or stolen cards, skimming, mail interception fraud, counterfeit cards, and other deceptive schemes.

There are several ways to deal with fraud, either by using technology or manually, such as manual review, address verification system, card verification method, positive and negative list, or payer authentication. Technology to detect fraud is developing very rapidly, such as rules based systems, neural networks, chip cards and biometrics are currently popular techniques Bhatla et al., 2003.

Credit card fraud detection research has been conducted by several researchers, using machine learning and not. Dal Pozzolo et al., 2014 provides answers to credit card fraud problems such as the scarcity of datasets due to confidentiality issues and algorithms that are unable to detect fraud due to high unbalanced, non-stationary distributed data. The answers written in the paper are based on the views of professionals by focusing on these problems.Then, Dal Pozzolo et al., 2015 conducted research using the undersampling method because users who are suspected of fraud are very small compared to legitimate users. In addition, he also used Bayes Minimum Risk theory to find the correct classification threshold because the undersampling method affects the *posteriori* probability - the chance that changes after fraud occurs - from the machine learning model.

There are several studies that focus on modifying machine learning to get better performance, such as Carcillo et al., 2019 which combines unsupervised and supervised learning methods in a machine learning model. Supervised learning is used to study past fraud behavior, while unsupervised techniques target the detection of new types of fraud. The accuracy of the machine learning model by combining the two techniques rises. Then, there is Xu et al.,

2023 who introduced the Deep Boosting Decision Tree (DBDT) model to detect fraud based on neutral network and gradient boosting. In the experiment, there are several imbalanced datasets used to test several models tested such as Random Forest, LightGBM, Gaussian Naive Bayes, Multilayer Perceptron, XGBoost, AdaBoost, and etc. The DBDT model has a better AUC score than all models in all datasets.

# 4 Result

## 4.1 Exploratory Data Analysis

### 4.1.1 Transaction

The dataset analyzed in this study consists of transactions recorded over a 48-hour period. As shown in Figure 9, the cash flow from customers' credit cards during this period ranges from approximately $200,000 to $900,000 for legitimate transactions. The total transaction volume in the morning (9th to 22nd hour) amounts to $11,000,000, while at night (22nd to 34th hour) it is slightly above $4,000,000. These transactions follow a periodic pattern, occurring frequently between morning and evening, with a significant decline during the evening and night.

However, this trend does not apply to fraudulent transactions. The amount of money associated with suspected fraudulent activities ranges from $50 to $6,000, with the total fraudulent transaction volume over the 48-hour period reaching $60,000. During the daytime (9th to 22nd hour), fraudulent transactions account for approximately $22,000, representing 0.2% of total daytime transactions. At night (22nd to 34th hour), fraudulent transactions total just over $12,000, making up 0.28% of the total nighttime transactions. Additionally, while the image does not reveal any repeating patterns, indicating that fraud can occur at any time, fraudsters tend to acquire more money during nighttime transactions.

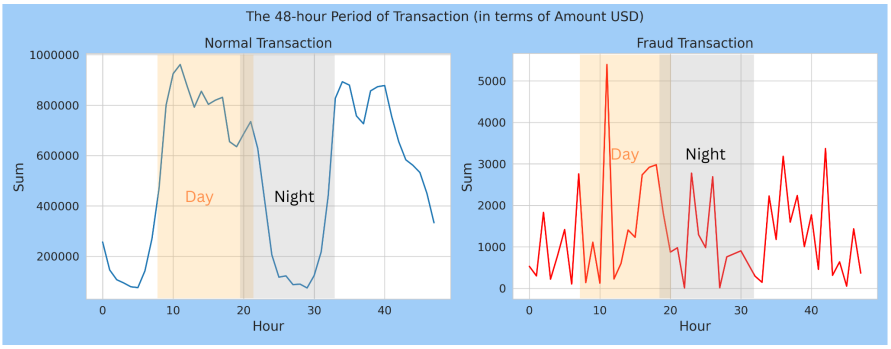



Figure 9: The transaction value in 48 hours

Figure 10 presents the transactions executed over a 48-hour period, following a pattern similar to the previous figure. During the daytime, 115 thousand credit card transactions were recorded, with just over 170 identified as fraudulent, resulting in a fraud probability of 0.15%. Meanwhile, at night, around 54

thousand transactions occurred, 106 of which were classified as fraud, yielding a fraud probability of just under 0.20%.

Fraudulent transactions peaked during the 10th hour, with 45 out of 8,000 transactions being fraudulent, corresponding to a fraud probability of approximately 0.56%. Interestingly, nighttime transactions showed a higher likelihood of fraud, with 35 out of 1,800 transactions being fraudulent, translating to a fraud probability of 1.94%. This indicates that fraud is 1.38% more likely to occur at night compared to the daytime.

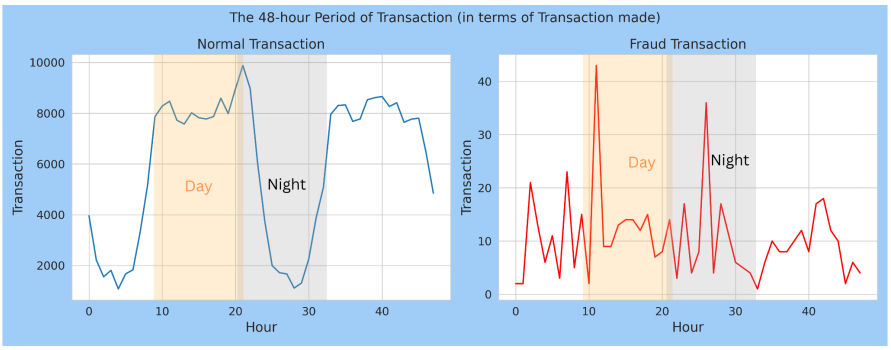



Figure 10: The Transaction Occurred in 48 hours

### 4.1.2 Imbalance

The Class column is a marker of whether the transaction is fraudulent or not. This column can be said as a *posteriori*, a calculation based on reports from users or the result of investigations from the company or bank that issued the credit card (Carcillo et al., 2019, Dal Pozzolo et al., 2015). If we see Figure 11, fraudulent transaction are very small in total, less than 0.5% of the total executed transactions. This means it is very rare to occurred.

This imbalanced data can be handled by using SMOTE (Chawla et al., 2002). In short, this technique is used to overcome this problem by compensating for the majority data (oversampling). The result of this method can be seen on the Figure 12.


Figure 11: Plot of Class Feature

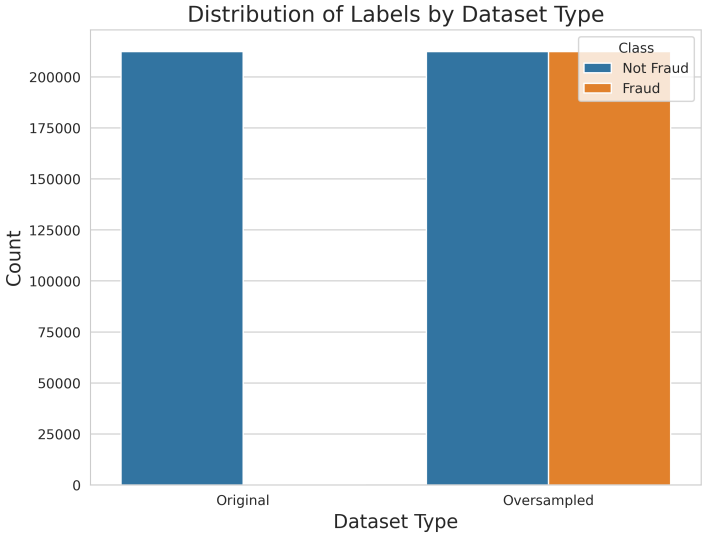
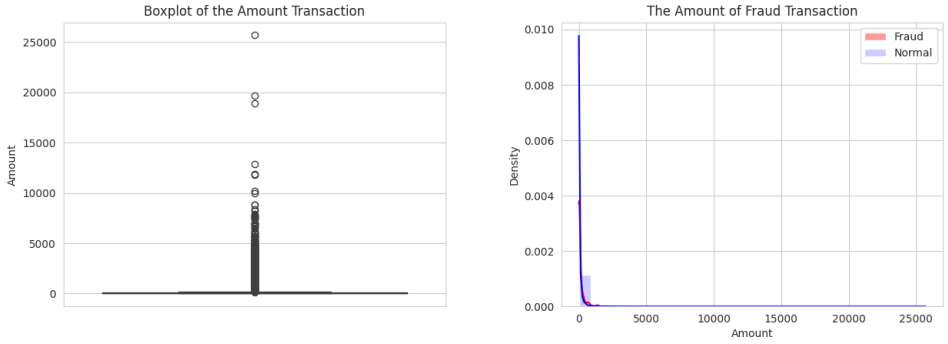



Figure 12: Class after Oversampling

### 4.1.3 Outlier and Skewness

**Outlier** is a data point that has a significant difference in value between the values in the dataset. One way to detect outliers is to use the Interquartile Range, which can be visualized with a boxplot, as in the **Amount** feature in Figure 13. Figure 13a illustrates a huge of outlier spots in the data, whereas

Figure 13b is the difference plot distribution between fraud and legit transaction in dollars. The plot shows that the majority of transactions occur in the nominal range of 100 to 1000 USD.




(a) Boxplot (b) Distribution Plot

Figure 13: Outlier Detection in Amount Feature

When all features are reviewed, the barplot in Figure 14 shows the number of outliers using the IQR method. Features V18, V19, and V20 are the features with the most outliers. In this project, outlier handling can be done in Imputation and Isolation Forest.

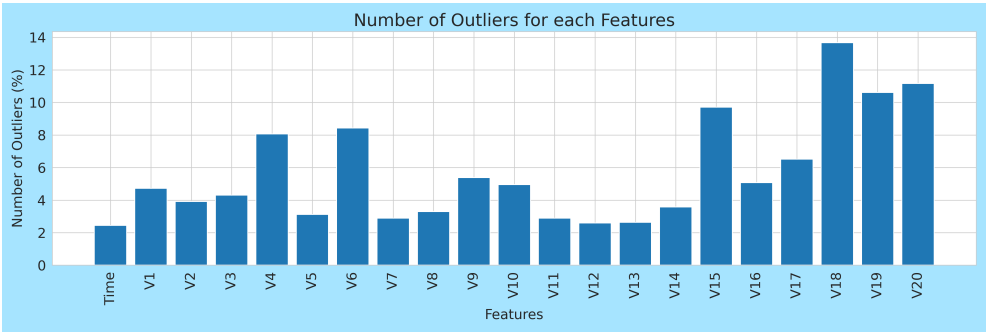



Figure 14: Number of Outlier in each Features

**Skewness**is a measure of how asymmetric the data distribution is. The coefficient value that can be said to be symmetric form, from both equations, if it is in the range of $-1 \leq skewness \leq 1$. The Figure 15 shows the skewnees score for every single features in the dataset. There are four features with the skewness score far outside the symmetric range, which is Class, Amount, V28, and V8.

The Figure 16 tells us the three plot of data distribution. Raw data is the dataset without the preprocessing treatment. The isolated forest post treament dataset shows no significance alteration. However, plot distribution

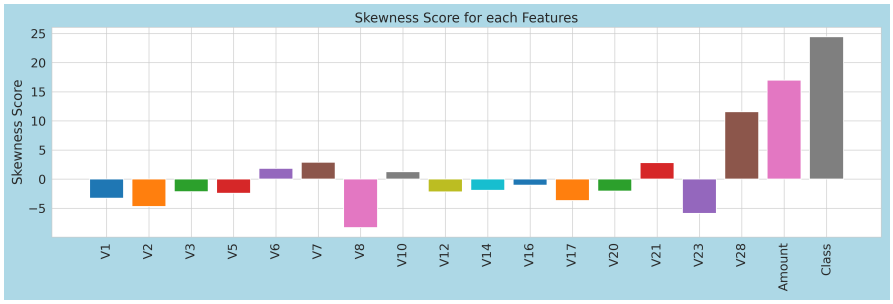



Figure 15: Skewness Score each Features

with imputation is totaly different. It shows a major transformation if we compare it with raw data. Both imputation and isolation forest consider the high value of transactions are outliers.

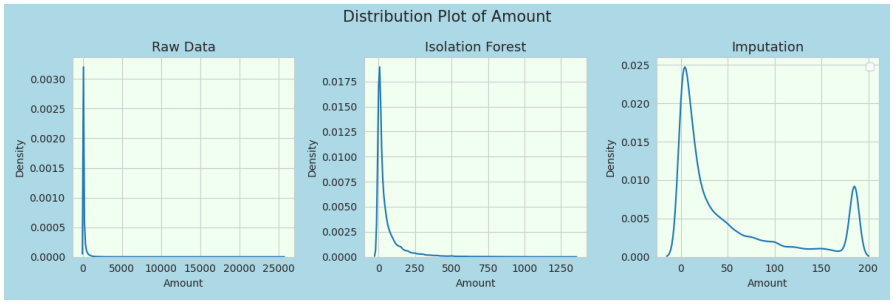



Figure 16: Distribution Plot of Amount

## 4.2 Machine Learning Result

The machine learning model employed in this project is Random Forest, a widely used open-source algorithm available in Python libraries. Random Forest was chosen due to its robustness in handling complex datasets and its non-parametric nature, which allows it to detect fraud without requiring predefined parameter values. This adaptability makes it well-suited for fraud detection, where patterns can be highly non-linear and variable across different transactions. Additionally, Random Forest excels in handling imbalanced datasets, a common challenge in fraud detection, by leveraging multiple decision trees to improve accuracy and reduce overfitting. Given these advantages, Random Forest provides a reliable foundation for building an effective fraud detection system.

The random forest will be trained with three datasets that have undergone different treatments, two of which have been handled with the Imputation and Isolation forest, while the rest are not treated at all (just raw data).

### 4.2.1 Feature Importances of each Dataset

The Random Forest is a machine learning model that can be used to detect credit card fraud, but can also see the contribution of each feature, which is represented by the feature importances score. In this project, features that have a score of more than 0.05 will be considered eligible. Based on the three feature importance plots using the Random Forest model, all features that have a score above 0.05 (red line) are features that are related to the user's identity. The selected features will be used to train the model. The raw dataset and imputation-preprocessed dataset has more features than the Isolation Forest.

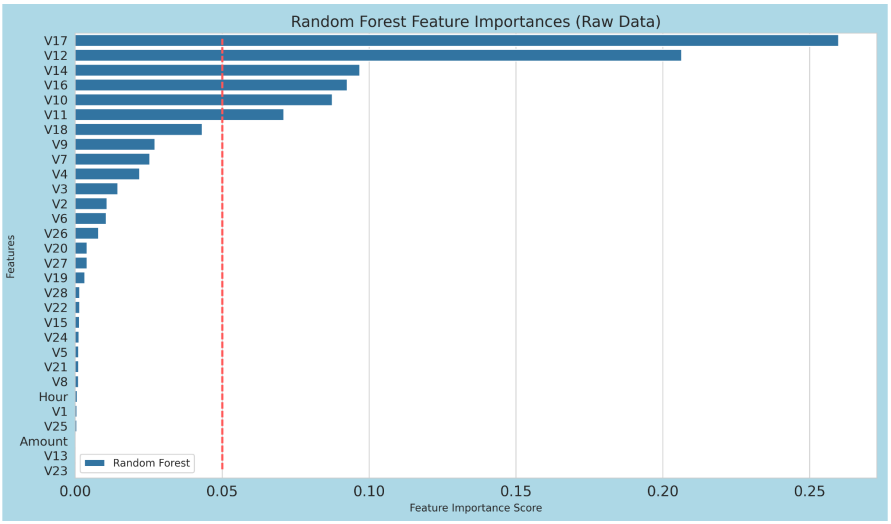



Figure 17: Raw data Feature Importances

### 4.2.2 Confusion Matrix

The confusion matrix explanation is provided in 3.3.3. As shown in Figure 20, the machine learning models across the three datasets exhibit nearly similar performance. However, the Isolation Forest model processes fewer data points than the other two datasets because, during preprocessing, the algorithm identifies certain data points as outliers (anomalies) and removes them. Notably, most of these anomalies correspond to data points labeled as fraud.

The classification report in Table 6 indicates that the model achieves 100% accuracy in detecting fraud across all datasets, suggesting flawless fraud detection. However, this result appears misleading when compared with the Confusion Matrix in Figure 20, which shows nonzero false negative (FN) and false positive (FP) values. This discrepancy implies that the model still strug-

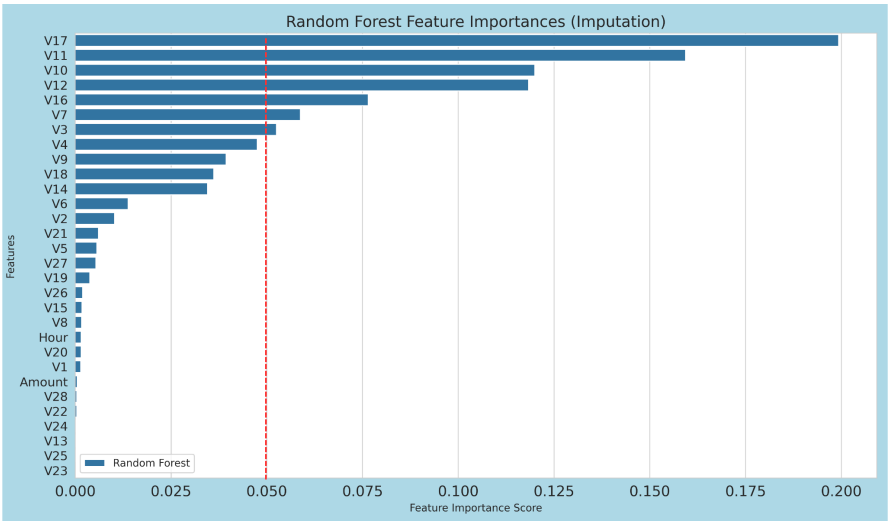



Figure 18: Imputation data feature importances

gles to correctly classify certain legitimate credit card transactions, despite its seemingly perfect accuracy score.

| Preprocessing Method | Accuracy | Recall | F1 Score | Precision |
|---|---|---|---|---|
| Raw or non-preprocessing | 100 | 67 | 74 | 83 |
| Imputation | 100 | 69 | 76 | 83 |
| Isolation Forest | 100 | 0 | 0 | 0 |

Table 6: The performance report of the model without oversampled dataset

After applying the oversampling technique, the confusion matrix results change, as illustrated in Figure 21. Using the raw and imputed datasets, the system correctly classifies 50,000 legitimate and fraudulent transactions, respectively. In comparison, the Isolation Forest model accurately identifies 44,000 legitimate transactions and 20,000 fraudulent ones. However, when examining False Negatives (FN)—instances where the system incorrectly classifies a fraudulent transaction as legitimate—the Isolation Forest model has the highest count, with 30,000 FN cases. This indicates that the machine learning model using the isolation-forest-preprocessed dataset requires further optimization. Althought if the model utilise Grid Search hyperparameter method, compared to the 21, there is no significant alteration.

Table 7 highlights a significant shift in model performance following the application of the oversampling method. While models trained on raw data and imputed datasets demonstrate high detection accuracy, the results for the Isolation Forest-preprocessed dataset differ notably. Although the model can still differentiate between fraudulent and legitimate transactions, its perfor-

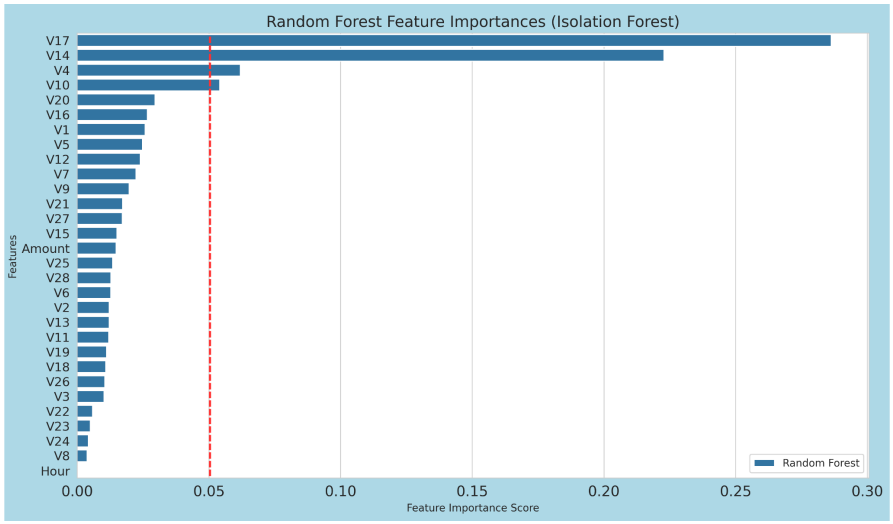



Figure 19: Feature Importance in Isolation Forest Data

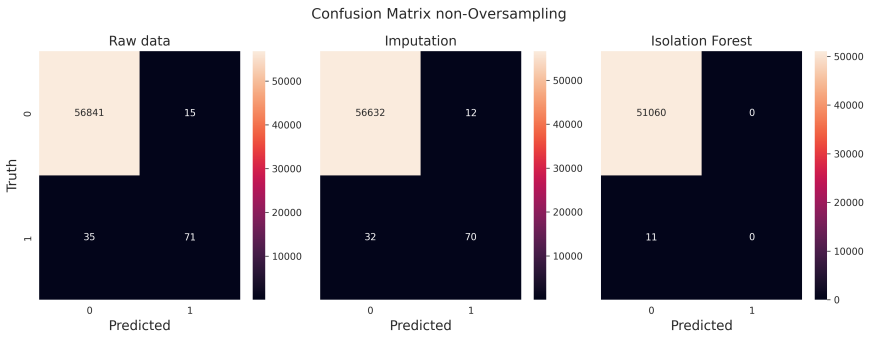



Figure 20: Confusion Matrix without Oversampling

mance indicators —accuracy, recall, precision, F1-score, and AUC — are considerably lower compared to the raw and imputed datasets. This discrepancy likely stems from Isolation Forest removing key fraud indicators, misclassifying them as anomalies during preprocessing. To mitigate this effect, adjusting the contamination level parameter could help preserve essential fraud-related patterns and improve model performance.

| Preprocessing Method | Accuracy | Recall | F1 Score | Precision |
|---|---|---|---|---|
| Raw or non-preprocessing | 92% | 86% | 92% | 98% |
| Imputation | 93% | 86% | 93% | 100% |
| Isolation Forest | 64% | 41% | 53% | 75% |

Table 7: The performance report of the model with oversampled dataset

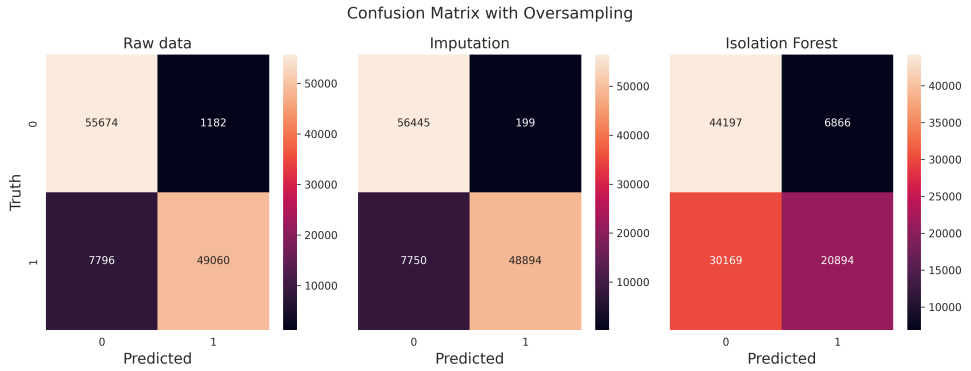



Figure 21: Oversampled Confusion Matrices

### 4.2.3 AUC and ROC Curve

The AUC-ROC curve evaluates how well the Random Forest model distinguishes between fraudulent and legitimate credit card transactions. In this study, we tested three different datasets, and the results show that the Raw and Imputation datasets performed best, each achieving an AUC score just above 90%. This indicates that the model can effectively differentiate between fraudulent and legitimate transactions.

The Isolation Forest model exhibited lower performance compared to the other two approaches. This suggests that, despite preprocessing, the dataset used with Isolation Forest still struggles to accurately identify fraudulent transactions.

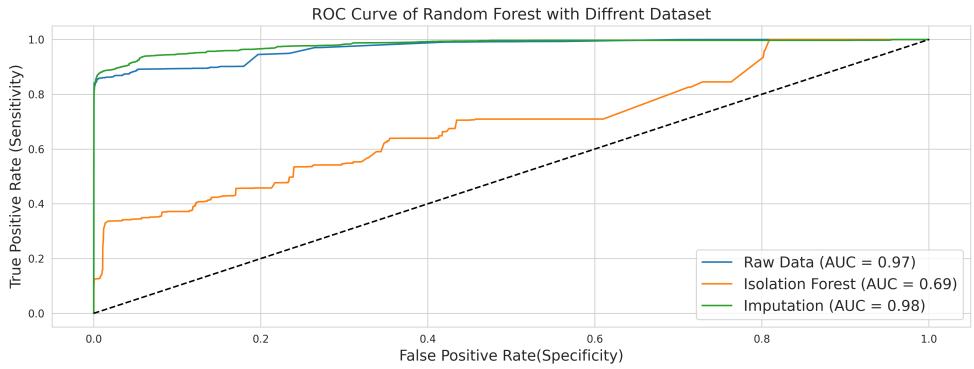



Figure 22: ROC Curve of Oversampled Dataset

# 5 Conclusion and Further Research

## 5.1 Conclusion

Based on the project, the conclusions are as follows:

1. The analysis of 48-hour transaction data shows that fraudulent activity is more frequent at night than during the day, as reflected in the pattern of detected fraud cases. Additionally, financial losses from nighttime fraud are higher compared to daytime, highlighting the increased risk during those hours. The fraudulent transactions over that period resulted in a total financial loss of $60,000.

2. Models trained on datasets from the raw and imputation approaches consistently outperform the data using the Isolation Forest.

## 5.2 Further Research

Suggestion for further research:

1. Try another machine learning model to predict credit card fraud.

2. Exploring alternative preprocessing models to enhance machine learning performance

## Literature


Alkhawaldeh, I., Albalkhi, I., & Nashwan, A. (2023). Challenges and limitations of synthetic minority oversampling techniques in machine learning. *World Journal of Methodology*, *13*. https://doi.org/10.5662/wjm.v13.i5.373

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, *9*(7), 1545–1588. https://doi.org/10.1162/neco.1997.9.7.1545

Barua, S., Islam, M. M., Yao, X., & Murase, K. (2014). Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, *26*(2), 405–425. https://doi.org/10.1109/TKDE.2012.232

Bhatla, T. P., Prabhu, V. C., & Dua, A. (2003). Understanding credit card frauds. https://api.semanticscholar.org/CorpusID:168440535

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010950718922

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, *24*(2), 123–140. https://doi.org/10.1023/A:1018054314350

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2019). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*. https://doi.org/10.1016/j.ins.2019.05.042

Chanda, R. K., Kumar Pagadala, P., Edukulla, C. K., Sai Archana, S., Gurram, S., & Maram, S. R. (2023). Enhancing credit card fraud prediction using decision trees, smote, and hyper-tuned random forests: A comprehensive approach. *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 794–799. https://doi.org/10.1109/I-SMAC58438.2023.10290611

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, *16*, 321–357. https://doi.org/10.1613/jair.953

Ciaburro, G. (2024). *Matlab for machine learning: Unlock the power of deep learning for swift and enhanced results*. Packt Publishing. https://books.google.co.id/books?id=0eXvEAAAQBAJ

Dal Pozzolo, A., Caelen, O., Johnson, R., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. https://doi.org/10.1109/SSCI.2015.33

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, *41*(10), 4915–4928. https://doi.org/https://doi.org/10.1016/j.eswa.2014.02.026

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, *121*(2), 256–285. https://doi.org/https://doi.org/10.1006/inco.1995.1136

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, 148–156.

Friedman, J. (2000). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*. https://doi.org/10.1214/aos/1013203451

Gill, P., Purnell, M., Crumpton, N., Robson Brown, K., Gostling, N., Stampanoni, M., & Rayfield, E. (2014). Dietary specializations and diversity in feeding ecology of the earliest stem mammals. *Nature*, *512*, 303–305. https://doi.org/10.1038/nature13622

Lebichot, B., Le Borgne, Y.-A., He, L., Oblé, F., & Bontempi, G. (2019). Deep-learning domain adaptation techniques for credit cards fraud detection. https://doi.org/10.1007/978-3-030-16841-4_8

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Mienye, I. D., & Jere, N. (2024). A survey of decision trees: Concepts, algorithms, and applications. *IEEE Access*, *12*, 86716–86727. https://doi.org/10.1109/ACCESS.2024.3416838

Rice, J. (2007). *Mathematical statistics and data analysis*. Thompson/Brooks/Cole. https://books.google.co.id/books?id=XKmRPwAACAAJ

Rivera-Lopez, R., Canul-Reich, J., Mezura-Montes, E., & Cruz-Chávez, M. A. (2022). Induction of decision trees as classification models through metaheuristics. *Swarm and Evolutionary Computation*, *69*, 101006. https://doi.org/https://doi.org/10.1016/j.swevo.2021.101006

Rokach, L., & Maimon, O. (2005). Decision trees. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 165–192). Springer US. https://doi.org/10.1007/0-387-25465-X_9

Strang, G. (2023). *Introduction to linear algebra* (6th ed.). CUP.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., & Krasser, S. (2009). Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(1), 281–288. https://doi.org/10.1109/TSMCB.2008.2002909

Tarawneh, A. S., Hassanat, A. B., Altarawneh, G. A., & Almuhaimeed, A. (2022). Stop oversampling for class imbalance learning: A review. *IEEE Access*, *10*, 47643–47660. https://doi.org/10.1109/ACCESS.2022.3169512

Triola, M. (2021). *Elementary statistics*. Pearson. https://books.google.co.id/books?id=FYfDzQEACAAJ

Volikatla, H. (2024). *Cloud based machine learning – practical guide to deploying ai models in the cloud*. RK Publication. https://books.google.co.id/books?id=lQEyEQAAQBAJ

Xu, B., Wang, Y., Liao, X., & Wang, K. (2023). Efficient fraud detection using deep boosting decision trees. *Decision Support Systems*, *175*, 114037. https://doi.org/https://doi.org/10.1016/j.dss.2023.114037

Zhou, H., Wei, L., Chen, G., Lin, P., & Lin, Y. (2019). Credit card fraud identification based on principal component analysis and improved ad-

aboost algorithm. *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, 507–510. https://doi.org/10.1109/ICICAS48597.2019.00111

---

Author: Fikri Abdillah

---