Credit Card Fraud Detection Using AdaBoost

International Islamic University Malaysia

CSCI 4340 MACHINE LEARNING

Dr. Amelia Ritahani Bt. Ismail
05/06/2022

Arranged by:

Mohamed Moubarak Mohamed Misbahou Mkouboi (1820705)
Karim Fazlul (1822049)
Mohammad Afif Muhajir (1816203)

TABLE OF CONTENTS

Abstract

With many available ways of paying for transactions, especially in conventional transactions, credit cards remain one of the most selected media of payment due to their ease of use. Still, there are a few problems that come along with it, one prime case for that is fraud very for those credit cards. Although, there are ways of identifying information to prevent credit card frauds, with the introduction of pattern and information automatic computation such has been provided by machine learning method this problem can be a field of study and help to reduce cases thereof.

Keywords: Credit card, conventional transaction, fraud, machine learning.

1.0 INTRODUCTION

With many ways of automated computing methods that are now available due to the continuous research and growth of technology, we can accomplish many tasks that previously were strenuous, especially in the way of doing computed prediction. Computed prediction is generally the process that is being done under Artificial Intelligence and its derivatives such as Machine Learning and Deep Learning. These kinds of prediction that is being done from AI or Machine Learning process is a form of predictive analysis which is, based on SAS definition, "At its core, predictive analytics encompasses a variety of statistical techniques (including machine learning, predictive modeling, and data mining) and uses statistics (both historical and current) to estimate, or 'predict', future outcomes" (Wakefield, n.d.). With the help of this component in Machine Learning, we can do many nigh-accurate predictions, assuming it's provided with high-quality data to model with. In this particular project, we want to do a specific task of detecting credit card fraud amongst a full set of credit card data. We are going to use a method named "AdaBoost Algorithm" or Adaptive Boosting Algorithm, to tackle the problem of identifying and detecting data patterns for credit card fraud. As for the dataset itself, it consists of 284,807 instances of credit card transactions with over 28 attributes. Due to confidentiality issues, the name and details of the attributes can't be shared further but it has been confirmed that these attributes are the principal components of the data processing.

1.1 Project Objectives

a. To develop a working model for credit card fraud detection.

b. To research and do a proper implementation of AdaBoost Algorithm.

c. To evaluate accuracy and performance of Adaboost algorithm with other algorithms (kNN, logistic, SVM, Decision Tree)

1.2 Model Objectives

a. To determine a connection between each variable with credit card data that shows as a fraud.
b. To discover what's the biggest factor in determining credit card fraud in the prepared dataset.
c. To understand if the desired algorithm model can accurately determine credit card fraud data with a provided dataset.

2.0 EXPERIMENTAL SETUP

2.1 Literature Review

- Introduction to Credit Card Fraud

Credit card use has become a vital feature of modern economies in our day and age. It remains popular in society, particularly among people who enjoy making online transactions. A credit card is a bank-issued card that enables its owner to pay for products and services at businesses that accept card payments. Each credit card has a defined amount of cash (credit) that you may access through the card and use in advance to purchase goods and services, paying back the funds at the end of the month or according to the repayment schedule with or without interest charges. Moreover, in this topic, we are talking about credit card fraud which is an act perpetrated by anybody who, intending to defraud, utilizes a revoked, canceled, reported lost, or stolen credit card to receive

something of value Credit card fraud can also occur when a credit card number is used without the physical card being present. Stealing a person's identity to obtain a credit card is a more dangerous type of fraud since it works in tandem with identity theft. Credit card fraud is a widespread issue in the consumer credit business. It is one of the most rapidly rising kinds of fraud, as well as one of the most difficult to detect.

- Credit Card Default and Its Consequences

A credit card default can have major implications, including significant harm to your credit score. A credit card default happens when you fail to pay the minimum amount owing for multiple months. A default notice is usually delivered after six months of missing payments. Contrary to popular belief, it takes more than a handful of missed payments. Your credit card account will become late before it falls into default. This occurs if you have not made a payment in the last 30 days. If you do not make even the minimum payment on your credit card for six months, it will fall into default.

When you begin to get behind on payments, your lender will contact you to inquire about your circumstances and how you intend to settle your sum. If your card issuer is unhappy with your response, your account will be canceled and a no-payment report will be issued to credit bureaus. Late payments will be reported to credit bureaus in the run-up to a default. When you fail to make payments on time, your credit score suffers. Even if you try to rebuild your credit shortly after a default, many lenders may refuse you a new credit card or loan. They'll want to see any outstanding bills paid off or settled and a new track record of on-time payments. Lenders are concerned that you will default on any additional financial commitments unless you can demonstrate differently.

However, the ramifications of credit card fraud are

1. The account sent to collections

   Credit card companies can either shut your account and send the debt to a collection agency, which may sue you, or they can sue you. When this occurs, you will no longer interact with the credit card company, but with a collection agency. This may be highly inconvenient since you may receive a flood of calls from the collection agency and may face a lien on your wages.

2. Legal action

   Some creditors are more aggressive than others. If they need an urgent resolution, they may file a local court judgment against you, which might result in a payroll lien, which requires your company to transmit a percentage of your salary straight to your creditors.

3. Decrease your credit score

   The most significant thing on your credit record is a late payment. Failure to make regular payments for six months might result in a loss of hundreds of points on your credit score, which can take years to restore.

4. Increase in interest rates

   If you are 60 days late on a payment, your interest rate will most likely skyrocket. Because you're skipping payments and carrying a balance, your interest payments will only increase, making it much more challenging to settle the loans.

5. Decrease in credit limit

   When you default on a credit card, creditors see you as extremely hazardous. They may reduce their own risk by reducing the quantity of credit you have available to them. If you fail on a credit card, your credit limit on other cards may

be reduced. Lower credit limits bring additional issues, such as an increase in your credit use rate, which might affect your credit score.

- Machine Learning Used

  Classification algorithms will be employed in our project to determine whether or not the applicant is qualified for a credit card. Our initial project goal is to do research and correct implementation of the AdaBoost Algorithm. As a result, we selected to examine two-hybrid algorithms. We shall contrast the k-nearest Neighbour (KNN) method with the Adaboosting technique and the k-nearest Neighbour hybrid. In other words, we'll see which method performs best: a simple KNN algorithm or a hybrid with boosting. Furthermore, we will compare the accuracy and performance of the Adaboost algorithm to those of alternative methods (KNN, logistic, SVM, Decision Tree). The boosting algorithm is used to improve the performance of KNN, which has a high classification power.

  ➔ Logistics Regression

  Logistic regression is a supervised classification technique at its core. For a given collection of characteristics (or inputs), X, the target variable (or output), y, can only take discrete values in a classification issue. Logistic regression, contrary to common opinion, is a regression model.

  ➔ k-Nearest Neighbor

  KNN is an acronym for "K-Nearest Neighbour." It is a machine learning algorithm that is supervised. The algorithm can tackle classification and regression problem statements. The sign 'K' represents the number of nearest neighbours to a new unknown variable that must be predicted or categorized.

➔ SVM (Support Vector Machine)

SVM is a supervised machine learning technique that may be used for both classification and regression. Though we call them regression issues, they are best suited for categorization. The SVM algorithm's goal is to identify a hyperplane in an N-dimensional space that classifies the input points.

➔ Decision Tree

Decision Trees are a sort of Supervised Machine Learning (you describe what the input is and what the related output is in the training data) in which the data is continually separated based on a certain parameter. Two entities may explain the tree: decision nodes and leaves. The decisions or consequences are represented by the leaves. And the data is separated at the decision nodes.

➔ AdaBoost Algorithm

AdaBoost, also known as Adaptive Boosting, is a Machine Learning approach that is utilized as an Ensemble Method. The most frequent AdaBoost method is decision trees with one level, which is decision trees with just one split. These trees are often referred to as Decision Stumps. It is an ensemble learning approach (sometimes known as "meta-learning") that was developed to improve the performance of binary classifiers. AdaBoost employs an iterative strategy to improve poor classifiers by learning from their mistakes.

2.2. Modeling

- General Machine Learning Algorithms Procedure

Shown below is the steps and figure of implementation of the general process of using classification machine learning algorithms:

➢ Load up libraries and dataset

➢ Do dataset split into training and testing sets

➢ Proceed to do anomaly detection and clearance

➢ Apply the appropriate machine algorithm function

➢ Proceed to do training and testing of the model

➢ Finally, evaluate the accuracy result.

- Coding for implemented algorithms:

➔ Logistics Regression

```
log_pred = log_reg.predict(X_test)
print('---' * 45)
print("Logistic")
recall_scores.append(recall_score(y_test, log_pred))
print('Recall Score: {:.2f}'.format(recall_score(y_test, log_pred)))
precision_scores.append(precision_score(y_test, log_pred))
print('Precision Score: {:.2f}'.format(precision_score(y_test, log_pred)))
f1_scores.append(f1_score(y_test, log_pred))
print('F1 Score: {:.2f}'.format(f1_score(y_test, log_pred)))
accuracy_scores.append(accuracy_score(y_test, log_pred))
print('Accuracy Score: {:.2f}'.format(accuracy_score(y_test, log_pred)))
print('---' * 45)
```

➔ k-Nearest Neighbor

```
svc_pred = svc.predict(X_test)
print('---' * 45)
print("SVM")
recall_scores.append(recall_score(y_test, svc_pred))
print('Recall Score: {:.2f}'.format(recall_score(y_test, svc_pred)))
precision_scores.append(precision_score(y_test, svc_pred))
print('Precision Score: {:.2f}'.format(precision_score(y_test, svc_pred)))
f1_scores.append(f1_score(y_test, svc_pred))
print('F1 Score: {:.2f}'.format(f1_score(y_test, svc_pred)))
accuracy_scores.append(accuracy_score(y_test, svc_pred))
print('Accuracy Score: {:.2f}'.format(accuracy_score(y_test, svc_pred)))
print('---' * 45)
```

➔ SVM (Support Vector Machine)

```
svc_pred = svc.predict(X_test)
print('---' * 45)
print("SVM")
recall_scores.append(recall_score(y_test, svc_pred))
print('Recall Score: {:.2f}'.format(recall_score(y_test, svc_pred)))
precision_scores.append(precision_score(y_test, svc_pred))
print('Precision Score: {:.2f}'.format(precision_score(y_test, svc_pred)))
f1_scores.append(f1_score(y_test, svc_pred))
print('F1 Score: {:.2f}'.format(f1_score(y_test, svc_pred)))
accuracy_scores.append(accuracy_score(y_test, svc_pred))
print('Accuracy Score: {:.2f}'.format(accuracy_score(y_test, svc_pred)))
print('---' * 45)
```

➔ Decision Tree

```
tree_pred = tree_clf.predict(X_test)
print('---' * 45)
print("DT")
recall_scores.append(recall_score(y_test, tree_pred))
print('Recall Score: {:.2f}'.format(recall_score(y_test, tree_pred)))
precision_scores.append(precision_score(y_test, tree_pred))
print('Precision Score: {:.2f}'.format(precision_score(y_test, tree_pred)))
f1_scores.append(f1_score(y_test, tree_pred))
print('F1 Score: {:.2f}'.format(f1_score(y_test, tree_pred)))
accuracy_scores.append(accuracy_score(y_test, tree_pred))
print('Accuracy Score: {:.2f}'.format(accuracy_score(y_test, tree_pred)))
print('---' * 45)
```

➔ AdaBoost Algorithm

```
# make predictions using adaboost for classification
from sklearn.datasets import make_classification
from sklearn.ensemble import AdaBoostClassifier
# define the model
model = AdaBoostClassifier()
# fit the model on the whole dataset
model.fit(X_train, y_train)
# make a single prediction
boosted_pred = model.predict(X_test)

print('---' * 45)
print("AdaBoosting")
recall_scores.append(recall_score(y_test, boosted_pred))
print('Recall Score: {:.2f}'.format(recall_score(y_test, boosted_pred)))
precision_scores.append(precision_score(y_test, boosted_pred))
print('Precision Score: {:.2f}'.format(precision_score(y_test, boosted_pred)))
f1_scores.append(f1_score(y_test, boosted_pred))
print('F1 Score: {:.2f}'.format(f1_score(y_test, boosted_pred)))
accuracy_scores.append(accuracy_score(y_test, boosted_pred))
print('Accuracy Score: {:.2f}'.format(accuracy_score(y_test, boosted_pred)))
print('---' * 45)
```

3.0 Data Analysis

3.1 Dataset Description

The dataset consists of transaction data made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where around 284,807 instances of transactions have been gathered. It contains only numeric input variables which are the result of a PCA transformation. Features V1, V2, … V28 is the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes a value of 1 in case of fraud and 0 otherwise.

As previously stated due to confidentiality issues we can't share details on many of the attributes. We can only provide this much information:

No.	Attribute Name	Description
1	Time	Integer, time unit value
2	Amount	Float, continuous value
3	V1	Float, continuous value
4	V2	Float, continuous value
5	V3	Float, continuous value
6	V4	Float, continuous value
7	V5	Float, continuous value

8	V6	Float, continuous value
9	V7	Float, continuous value
10	V8	Float, continuous value
11	V9	Float, continuous value
12	V10	Float, continuous value
13	V11	Float, continuous value
14	V12	Float, continuous value
15	V13	Float, continuous value
16	V14	Float, continuous value
17	V15	Float, continuous value
18	V16	Float, continuous value
19	V17	Float, continuous value
20	V18	Float, continuous value
21	V19	Float, continuous value
22	V20	Float, continuous value
23	V21	Float, continuous value
24	V22	Float, continuous value
25	V23	Float, continuous value
26	V24	Float, continuous value
27	V25	Float, continuous value
28	V26	Float, continuous value
29	V27	Float, continuous value
30	V28	Float, continuous value

3.2 Data Preprocessing

The dataset we are working on has been through some preprocessing methods beforehand, therefore the current dataset is fairly clean and easier to manipulate, but we still have done the process of anomaly detection in the form of an outlier.

3.2.1 Anomaly Detection (Outliers)

As explained before, this dataset has been cleaned and refined by the previous dataset handler for easier manipulation, but even so it is still a concern to detect any possible anomaly, in this case being outliers. The importance of detecting outliers for this dataset is because outliers with extreme values will cause higher chances of great inaccuracy with our model.

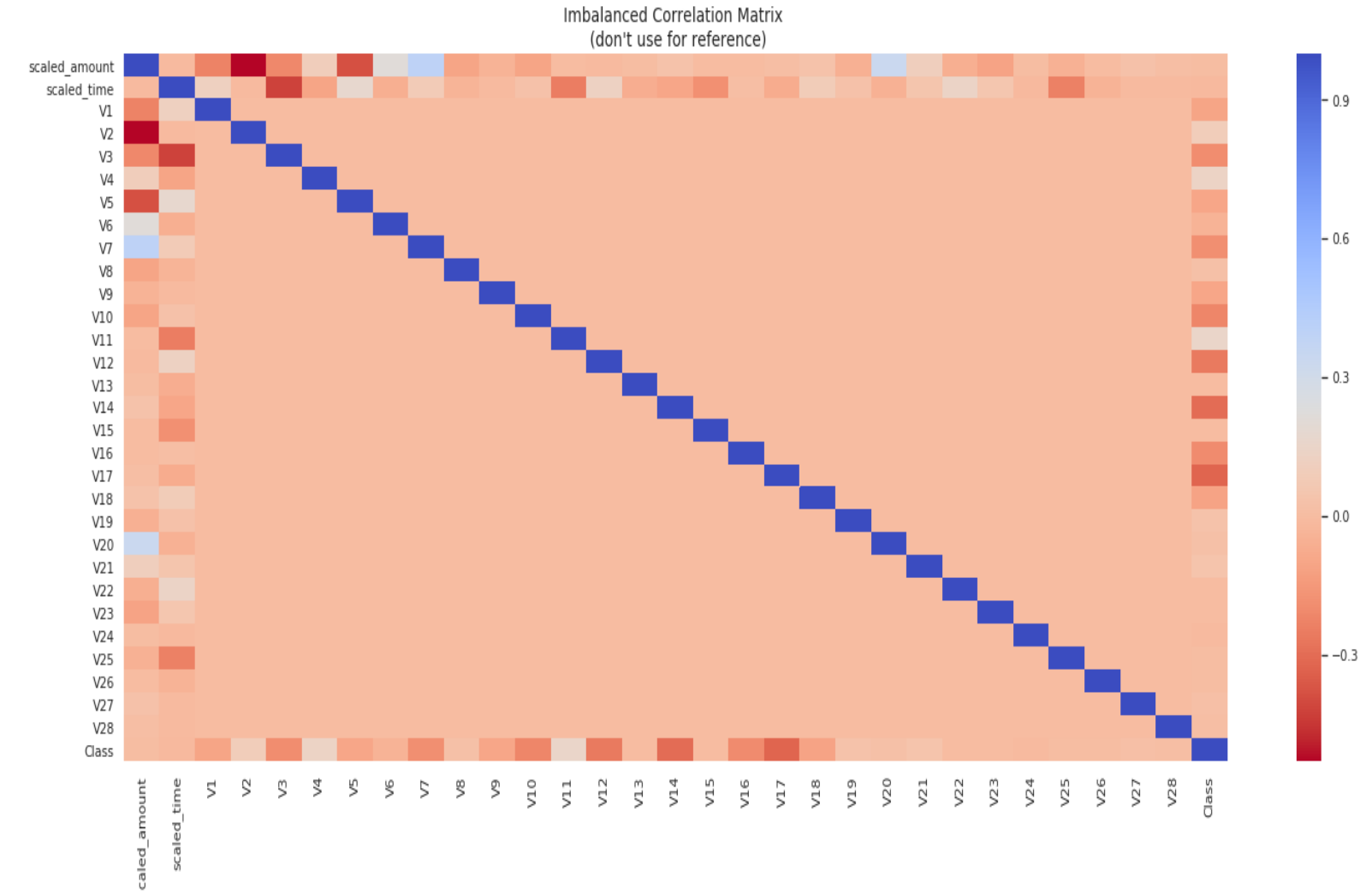
Our method of detecting any potential outliers in our dataset is by setting interquartile ranges (by calculating the differences for 25% and 75% percentile). Any data value beyond the set ranges will instantly be deleted. This method is also paired with a boxplot for easier to see the outliers' points.

3.3 Data Visualization

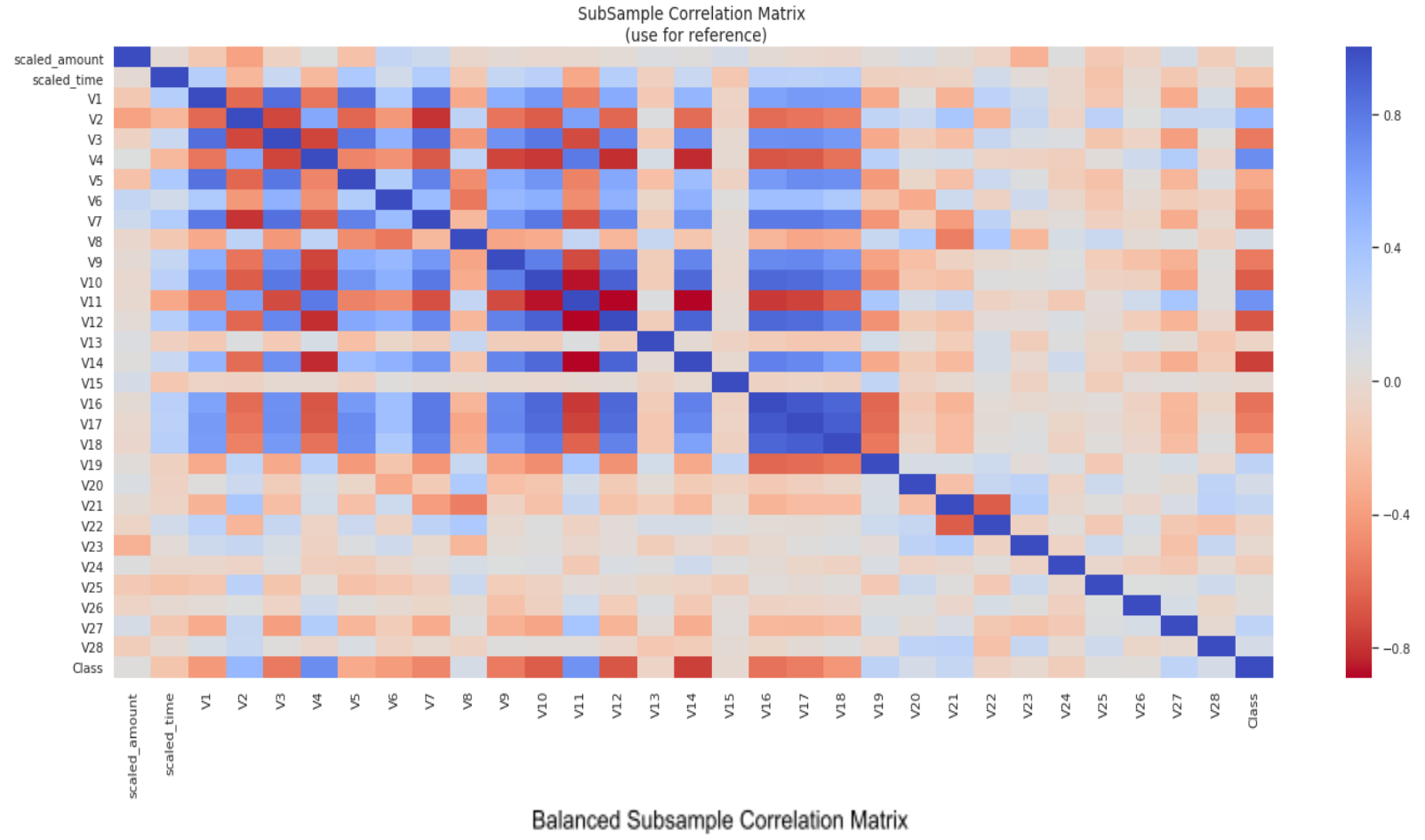
In this part, we will show the representation of our model in diagrams that presents the current analysis of data distributions, distributions, and learning curves for the dataset using many classification algorithms.

3.3.1 Data Correlation

In this project we want to find a clear correlation between each attribute as it can help determine if any specific feature or combination of features might identify a transaction data as a fraud. However, it is important that we use the correct dataframe (subsample) for us to see which features have a high positive or negative correlation concerning fraud transactions.



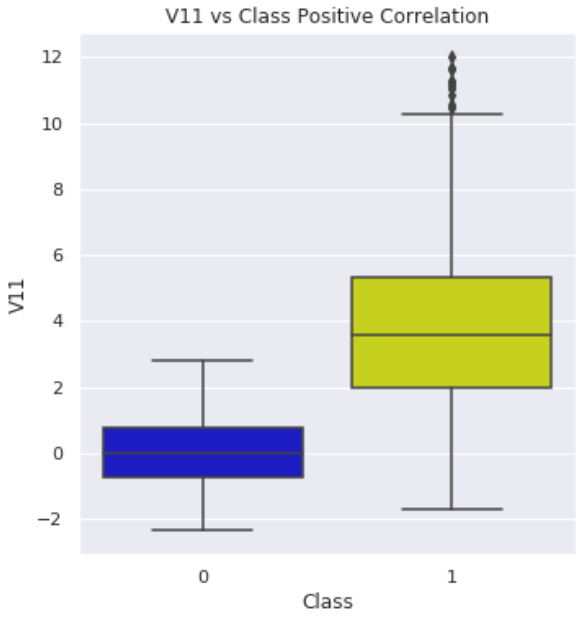
Imbalanced Correlation Matrix

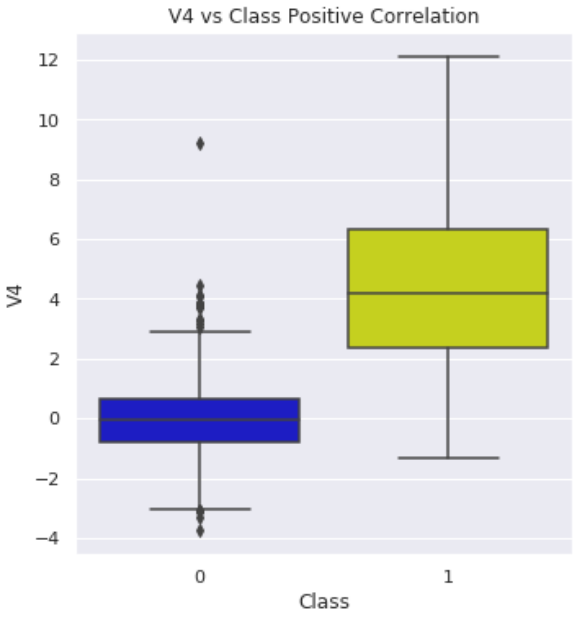


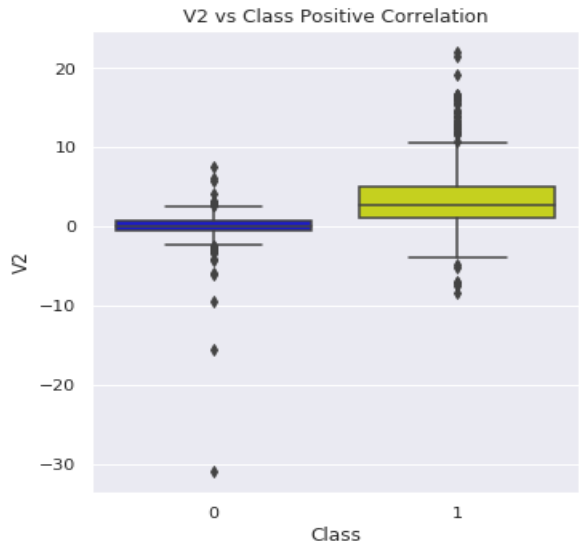
Balanced Subsample Correlation Matrix

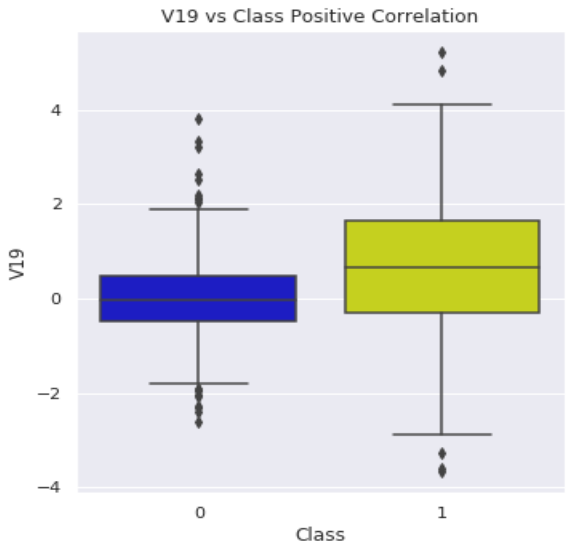
Subsampling is important for a highly imbalanced data set and it is depicted in the correlation plot as we can see in the correlation plot of the original data set Figure- (a) there is no significant correlation among the features but in the plot for balanced data set Figure- (b) we can observe distinct relationship among features where the attribute V14 has a very negative correlation with the class and transaction amount has a very high correlation with the class which is intuitive as the amount describe a lot of information about a transaction being normal or fraudulent.

In our further investigation of the correlations data for highly correlated data with predicted class, we found that column V11, V4, V2, and V19 provides unique details for the class label 1 which represents fraudulent transaction as the distribution for each case captures more information about the fraudulent case in comparison with the non-fraudulent transaction which is represented by class label 0, as can be seen in the figures below:

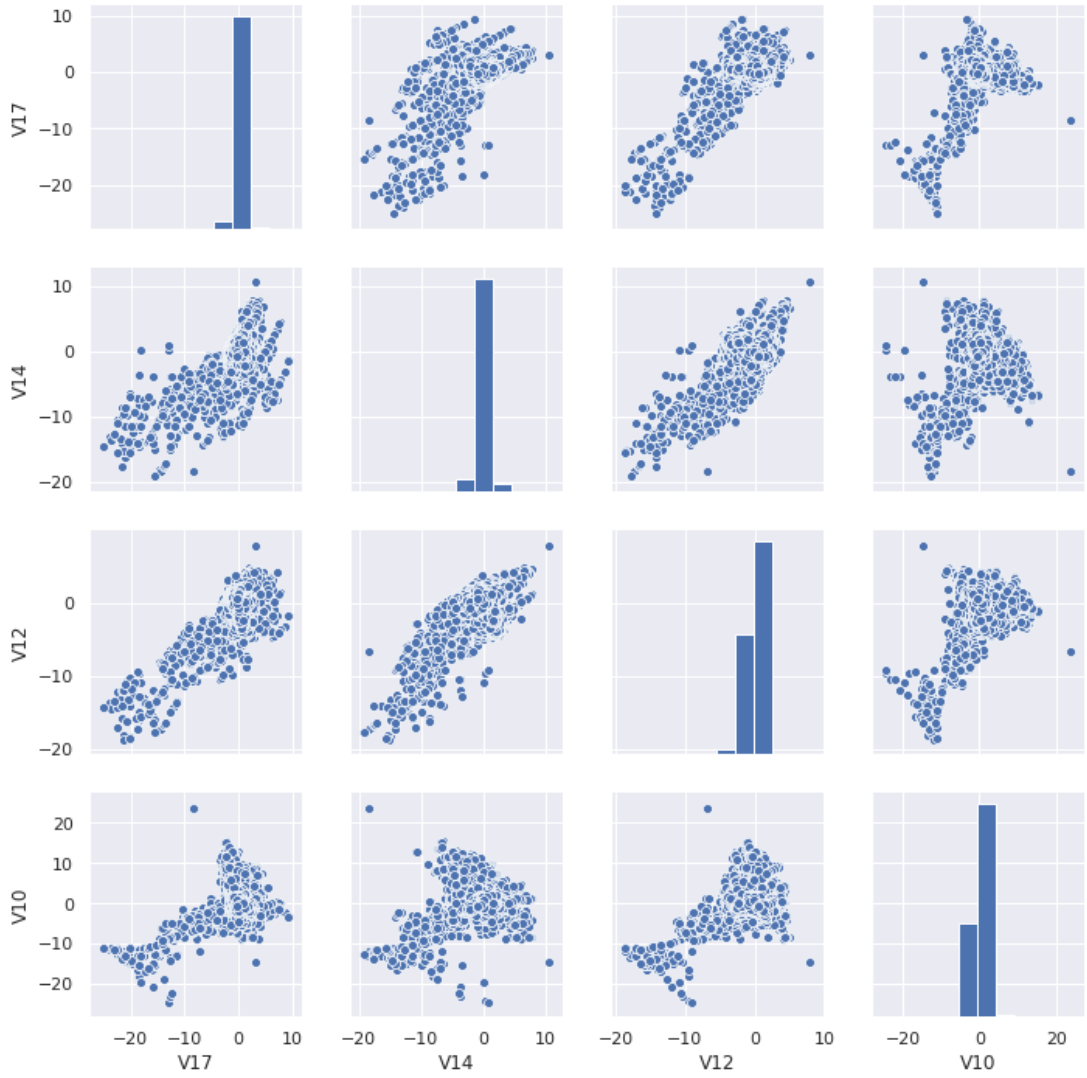






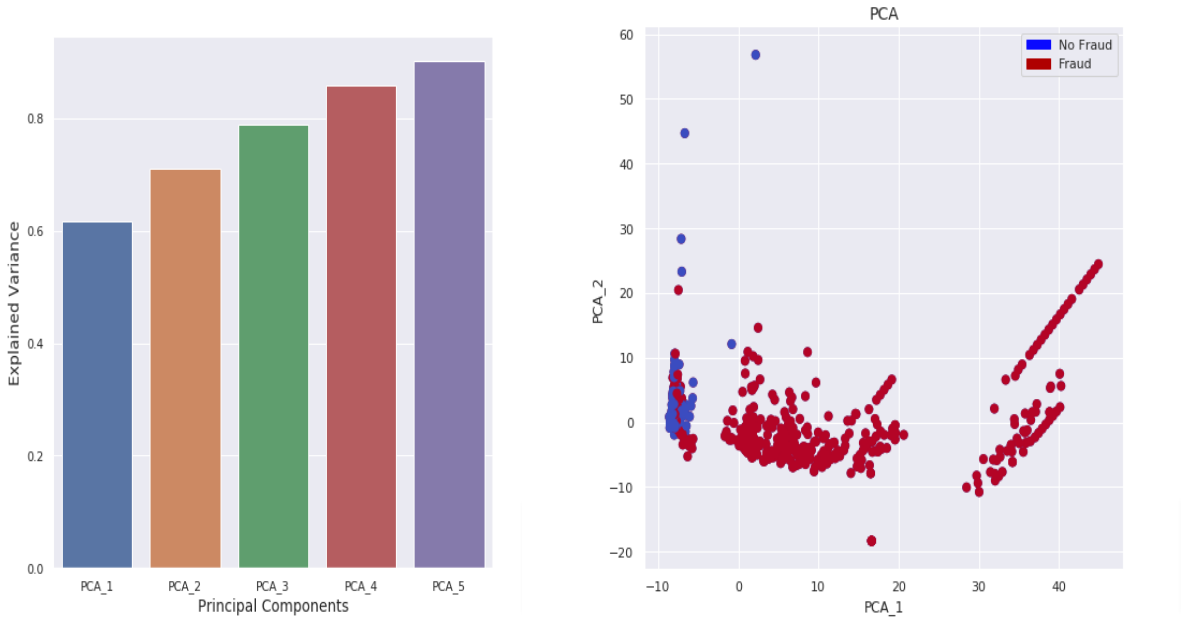


Other than those findings, we also managed to plot out other features that strongly correlate with each other as shown in the figure below:



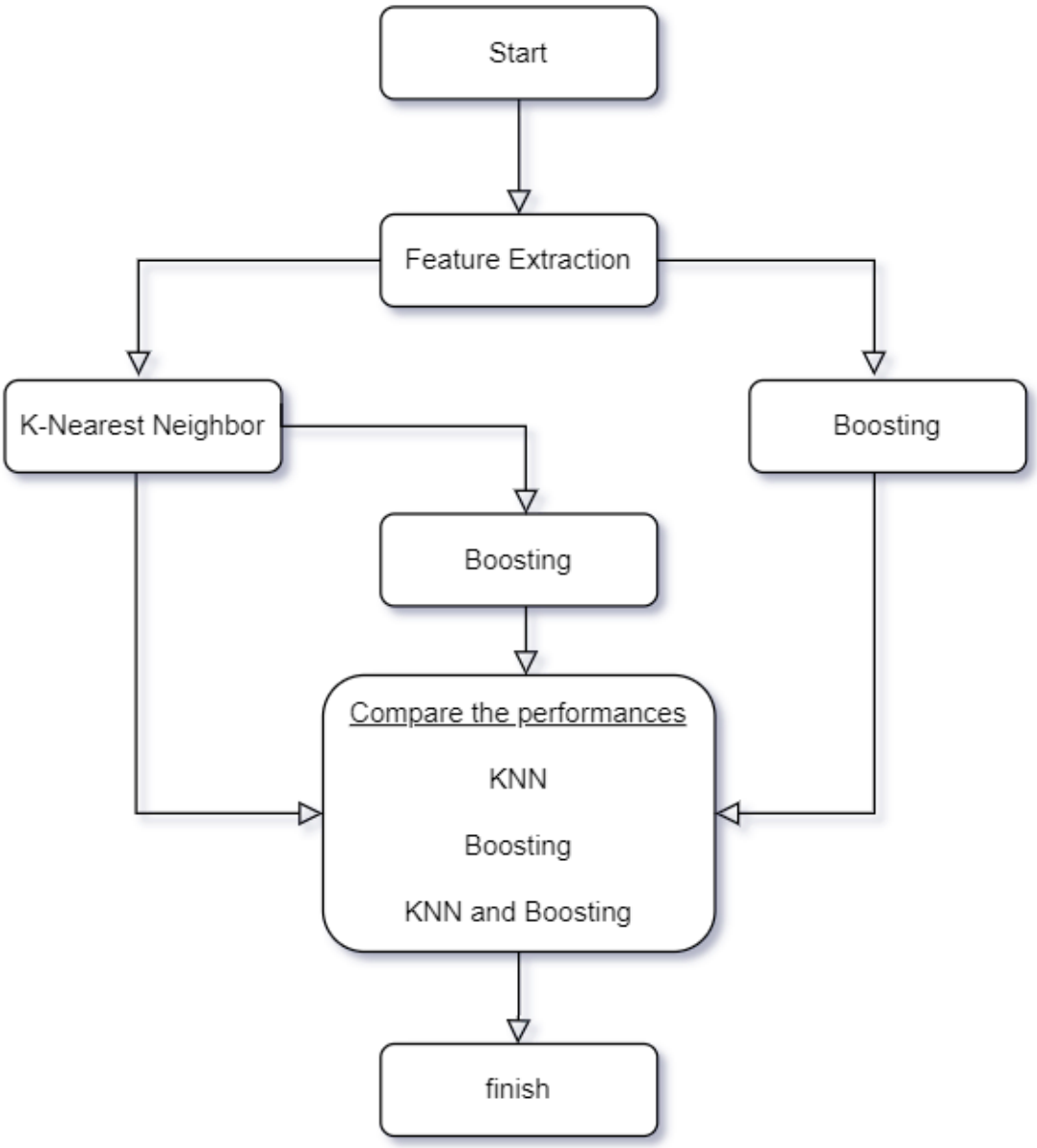
3.3.2 Principal Component Analysis

For our dataset, for it to be more manageable by compacting the size of overall data into smaller sets of components, we performed the Principal Component Analysis procedure. This procedure allows us to reduce the potential noise in our dataset by reducing it into principal component sets. The result can be seen in the figures below:

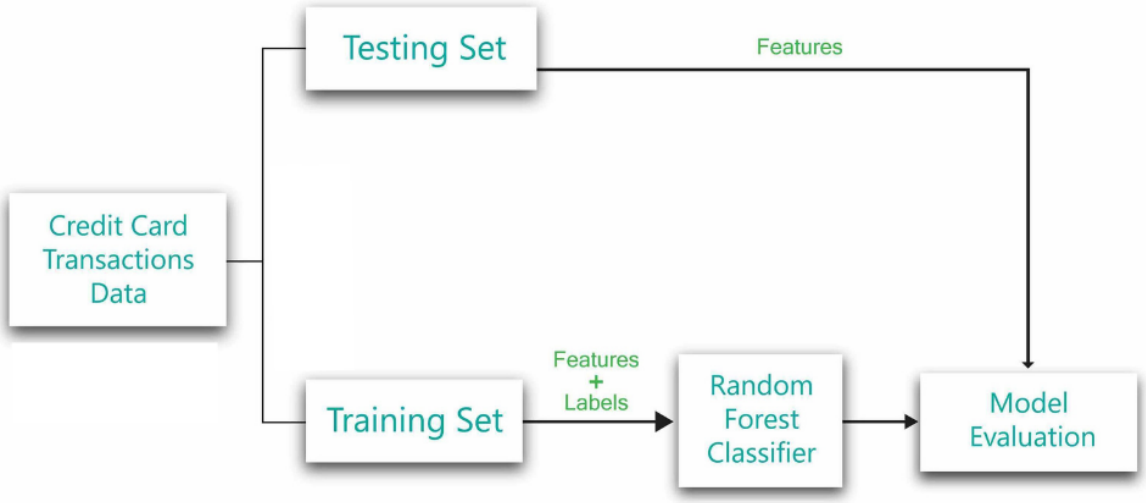


Based on the figure shown on the left, we can see that from our original dataset, 90% of its variances have been managed to be explained into 5 principal components. And after further attempts of testing our PCA result by trying to plot the first 2 principal components into a graph, we managed to do a successful clustering using those 2 principal components as shown in the right figure.

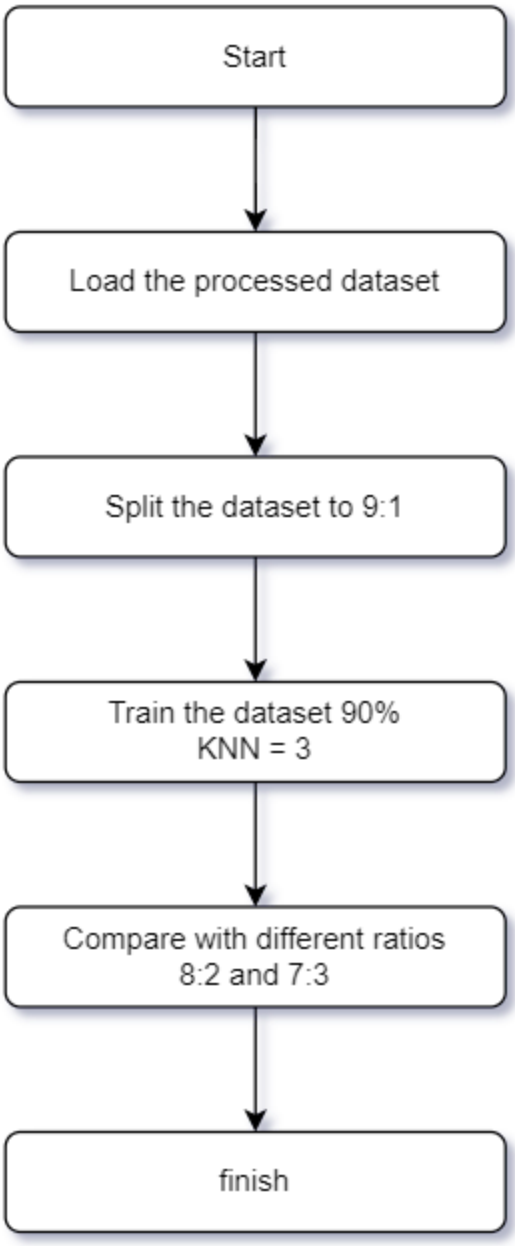
4.0 Model Development



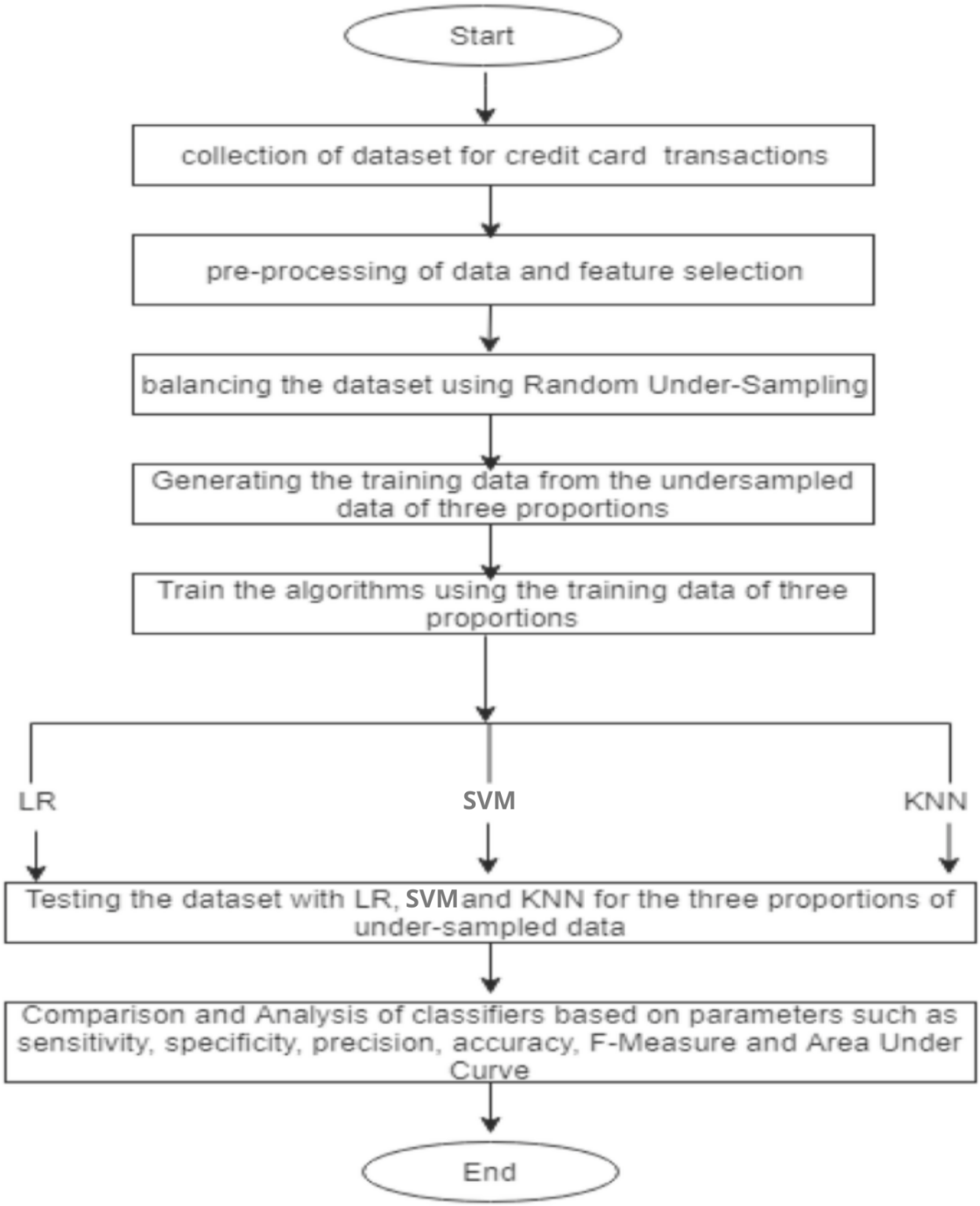
Flowchart for model development



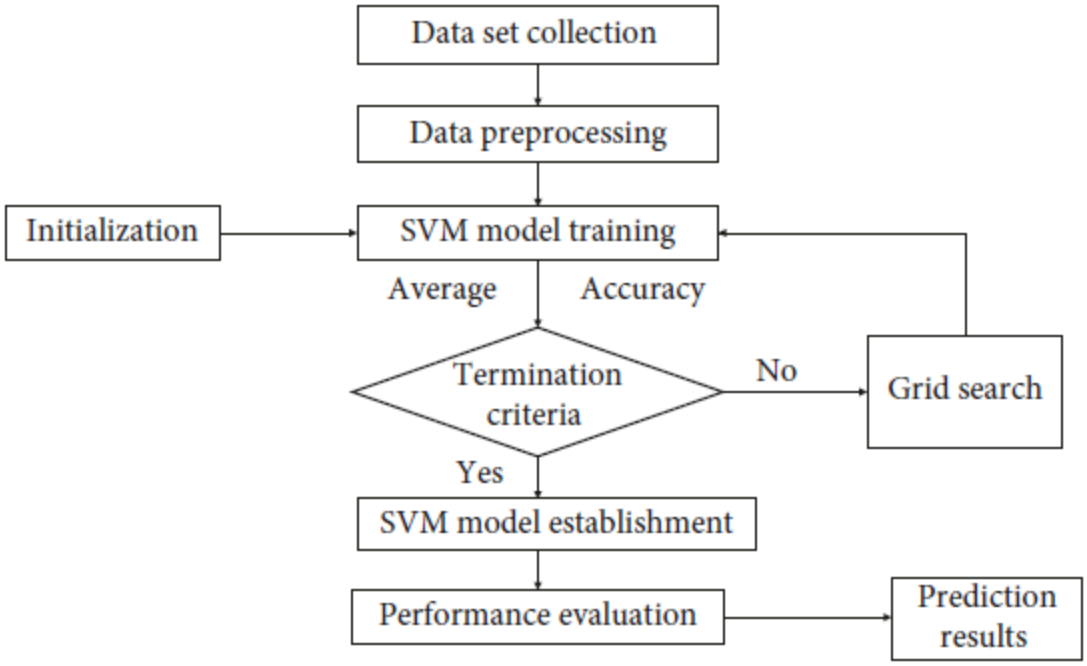
Flowchart for Decision Tree



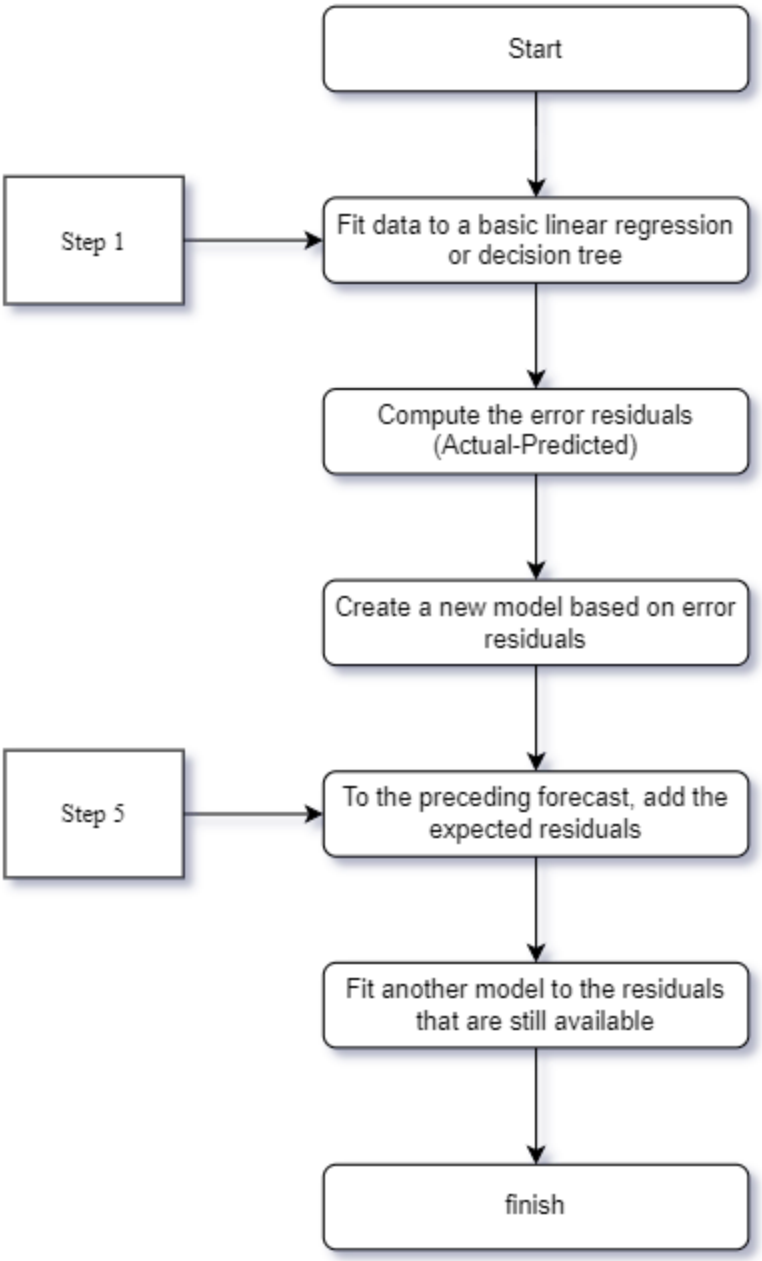
Flowchart for K-Nearest Neighbours algorithm



Flowchart for Logistics Regression



Flowchart of SVM.



Flowchart in AdaBoost algorithm

5.0 Result and Discussion

5.1 Results of accuracy from the myriad of Classification Machine Learning models

No.	Algorithm/Model Name	Accuracy Score	F1 Score	Recall Score	Precision Score
1	Logistics Regression	0.94	0.94	0.92	0.97
2	k-Nearest Neighbors	0.93	0.94	0.89	0.98
3	SVM	0.93	0.94	0.93	0.94
4	Decision Tree	0.91	0.95	0.92	0.93

We will use these results of accuracy and f1-score to be compared with AdaBoost Algorithm performances to determine which method is better suited for this problem of credit card fraud detection and dataset.

5.2 Result of accuracy from AdaBoost Algorithm model

No.	Algorithm/Model Name	Accuracy Score	F1 Score	Recall Score	Precision Score
1	Adaptive Boosting	0.93	0.93	0.92	0.94

References

5 consequences of a credit card default - bright. RSS. (n.d.). https://www.brightmoney.co/learn/5-consequences-of-a-credit-card-default#:~:text=your%20credit%20score.-,A%20credit%20card%20default%20can%20have%20severe%20consequences%2C%20leading%20to,seriously%20damaging%20your%20credit%20score.&text=A%20default%20on%20your%20credit,amount%20due%20for%20several%20months.

Lastname, C. (2008). Title of the source without caps except Proper Nouns or: First word after colon. *The Journal or Publication Italicized and Capped*, Vol#(Issue#), Page numbers.

Lastname, O. (2010). Online journal using DOI (digital object identifier). *Main Online Journal Name*, Vol#(Issue#), 159-192. https://doi.org/10.1000/182

Lastname, W. (2009). *Title of webpage*. Site Name. Retrieved July 3, 2019, from http://www.example.com