University of Sunderland

The Detection of Credit Card Fraud with Machine Learning Algorithm

Student Name: Adigun Abdul Azeez Oladipupo
Student Number: 219143011

Introduction

Fraudulent activity main objective can be said to be the attainment of a particular or monetary benefit by means of deceit.as a result, in other to avoid fraud from happening, the two important approach is to detect and prevent it (Yousefi, Alaghband and Garibay, 2019) . They are several types of cards use for payment that are broadly available, such as, prepaid card, credit card, or debit card. And they are the common way in which payment is done in lot of countries. In fact, the digital technology development has made provision for different ways of handling money, specifically in the ways payment is made that has shifted from action done physically into a digital action done electronically. Financial policies have been modernized as a result of the present environment, together with the way businesses plan and operate in either small or large organisations. Deceptive utilization of someone credit card information for purchasing items or services is called credit fraud. The transaction could be done in a physical or digital way (Sahoo and Gupta, 2019).For transaction done physically, there is a physical presence of the credit card. whereas the transactions that are done digitally occurs on mobile phone or the internet. The individuals that the card belongs to usually give personal information such as the number on the card, the number use to verify the card, and the expiring date of the card on the internet or mobile phones. As a result of how rapidly e-commerce is growing over the years, they has been enormous increment in the usage of credit card (Zainal, Md Som and Mohamed, 2017).

According to (Alladi *et al.*, 2020) in year 2011, roughly 317 million transactions were made using credit card in Malaysia. And by 2018 an increment of over 440 million was recorded. In 2015, a worldwide report of $21.84billion credit card fraud was recorded (Paschen, Kietzmann and Kietzmann, 2019). They have been growth in fraudulent cases has the use of credit card increases. Although , several techniques to verify it were established, the amount of credit card fraudulent cases have not drastically reduced. The possibility of significant financial benefits, together with regular development of financial services generates broad openings for frauds (Gianini *et al.*, 2020). The money from fraud payment card is frequently use for illegal actions that are mostly difficult to avoid, such as, supporting terrorism (Dal Pozzolo *et al.*, 2018) .Scammers have a preference of being on the cyberspace because they will be able to hide their whereabout and individuality. Current rise in credit card fraudulent action has strongly affected banking industry. The losses as a result of credit card fraud primarily affect the merchants since the whole cost is on them, as well as the charges by the issuer of their card, managerial charges with various other expenses. As a result of the merchants bearing all the cost, it leads to rise in price of product and discount reduction. Therefore, the reduction of this deficit is extremely necessary (Alenzi and Aljehane, 2020). Efficient way of detecting fraud is needed for reducing the amount of fraud cases.

Machine learning algorithm

Decision tree is a technique that allows the approximation of target functions that are separately estimated, whereby the acquired function is illustrated with a decision tree. Decision tree helps with classifying instances, where all the nodes within the tree indicates an analysis of various element of the instance, every branch moving down the node is linked with a probable value in the attribute. Decision tree can be seen as a schema where the node in it represents a test attribute that denotes the result of the attribute. The classification of an instance begins from the root node in a tree, analysing the attribute that the node identified, and moving down the

branches of the tree that links with the attributes value of a specified instances. The same procedure will be repeated with the subtree at the root of the new node. In developing how the decision tree will be calculated. The dataset was divided into train dataset and test dataset established on the dataset in consideration. And the prediction technique is use for measuring how the test dataset is performing (Hudali *et al.,* 2019).

Synthetic Minority Oversampling Technique (SMOTE) can be described as a machine learning method use for classifying data, it helps prove solution to machine learning difficulty of a class subjugating another class, which is called class imbalance, This happens as a result of a class having a way larger number than the other class. SMOTE assist with differentiating the instances and synthesizes the smaller instances. SMOTE function helps with shrinking the smaller instances from the actual instances (Sahayasakila, Aishwaryasikhakolli and Yasaswi, 2019). SMOTE methods will be used to train the dataset. With the utilization of the SMOTE method on the data, that is, the transactions will be trained. In this research the smote method is primarily use for differentiating the fraudulent transaction from the legitimate ones that the owners of the card made.

Pre-Processing on real or simulated data

The dataset was obtained from Kaggle, a site for data analysis which assist with the provision of dataset. The dataset contains a total number of 31 columns, and 28 of these columns are labelled v1-v28 so as to maintain the confidentiality of the users. The three other columns are Time, Amount, and Class. The interval of when the first and other subsequent transaction were made is indicated within the Time column. The Amount column indicates the total sum of money used in the transactions. Lately, the Class column have two variables 0 and 1, where 0 is the legitimate transitions while 1 is the fraudulent Transaction.

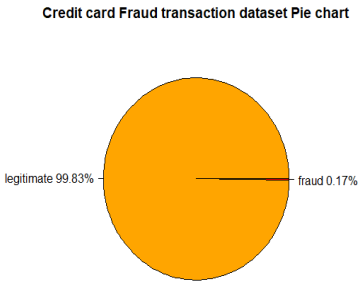
R Programming Content With Results

The class column was converted into a factor column with two factors which are 0 and 1, where 0 represent the legitimate transaction and 1 represent Fraudulent transaction.

```
 $ Amount: num  80.1 1 11.1 0 29.7 ...
 $ Class : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
>
```

The summary of the dataset indicates that 284315 are legitimate transaction while only 492 are fraudulent transaction and quite a few of the columns are integer having a mean of Zero. The dataset were inspected for missing values with is.na function, and the result indicated that they were no missing values. The number of fraudulent and legitimate transaction in the dataset were inspected by getting the distribution of the fraudulent and legitimate transaction within the dataset. The result above indicated that most of the transaction are legitimate with a total 284315 and the fraudulent transaction are only 492.The percentage of the dataset shows that 99.83 % are legitimate transaction while 0.17% are fraudulent transaction which clearly indicate that the dataset is Imbalanced

```
> #checking how Fraud and Legitimate are distributed within the dataset
> table(credit_card$Class)

     0      1
284315    492
> #Checking the Percentage of Fraud and Legitimate transaction in dataset
> prop.table(table(credit_card$Class))

         0          1
0.998272514 0.001727486
> #Displaying the dataset for credit card fraud transaction with a pie chart
> labels <- c("legitimate", "fraud")
> labels <- paste(labels, round(100*prop.table(table(credit_card$Class)), 2))
> labels <- paste0(labels, "%")
> pie(table(credit_card$Class), labels, col = c("orange","red"),
+     main ="Credit card Fraud transaction dataset Pie chart")
>
```

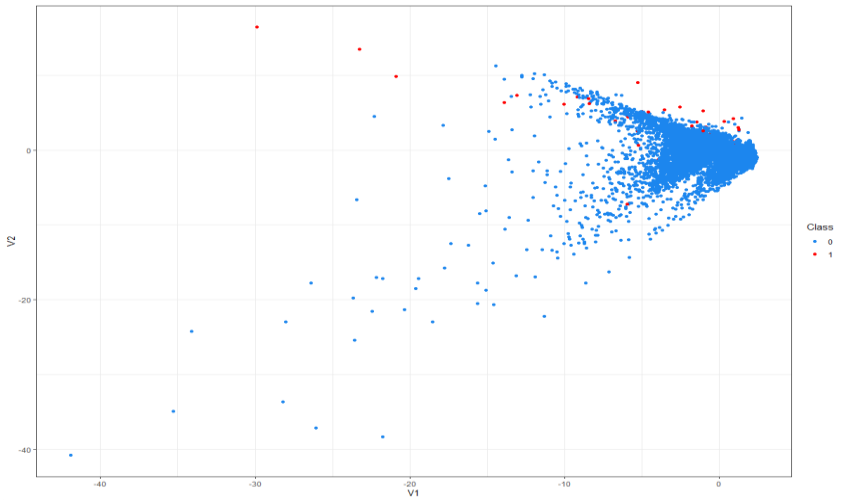


A model was created on the dataset to get how accurate they are without a machine learning model by predicting that every transaction within the dataset is legitimate.

```
> #------------------------------------------------------------------------
> #predictions with no model
> predictions <- rep.int(0, nrow(credit_card))
> predictions <- factor(predictions,levels =c(0 ,1)) # 0= legit, 1= Fraudd
> confusionMatrix(data = predictions, reference = credit_card$Class)
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 284315    492
         1      0      0

               Accuracy : 0.9983
                 95% CI : (0.9981, 0.9984)
    No Information Rate : 0.9983
    P-Value [Acc > NIR] : 0.512
```

The result above shows that, all legitimate transactions are correctly classified but none of the fraudulent transaction are classified. The legitimate cases have a true positive with a total of 284315, and the true negative of 0 which indicates that the fraudulent transaction is wrongly classified. Hence, all the 492 fraudulent transactions are flagged as legitimate transactions which are False positives. Therefore, it will be wrong to use the accuracy of the model due to fact that it does not represent anything, moreover, the objective is to maximize the true negative in other to classify most of the fraudulent transactions as fraudulent transaction.

A model was built using a small subset of the dataset for faster computation after which the model will be applied to the entire dataset.



As indicated in the scattered plot above, they are only few red colours which represent the fraud cases. Therefore, if a model is trained on this data, it will not be able to learn much because the

number of fraud cases are few. Hence, a different technique will be used for balancing the dataset before building a model on the data.

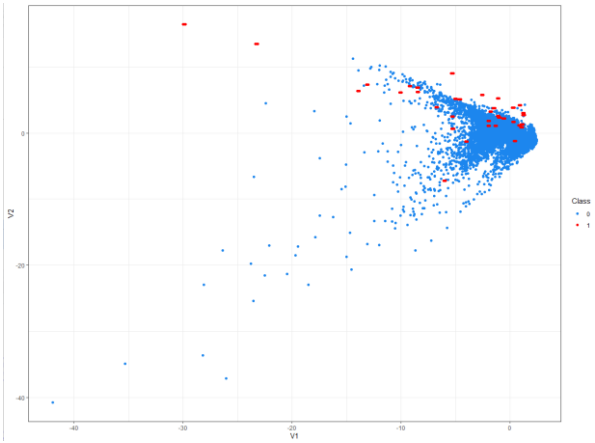
In preparation for balancing the dataset a training and test set where created, in which the balancing is been used on the training set to train the model and not the test set.

```
> dim(train_data)
[1] 22785    31
> dim(test_data)
[1] 5696   31
```

The model will be trained on the train data after which it has been balanced, thereafter, the performance will be tested using confusion matrix on the test data. The dimension result shows that the train data contains 22785 rows and 31 columns, while the test data shows a sum of 5696 rows and 31 column

A random over sampling (ROS) has been utilized in other to increase the amount of fraud causes with the data set, Hence the minority will be oversampled.

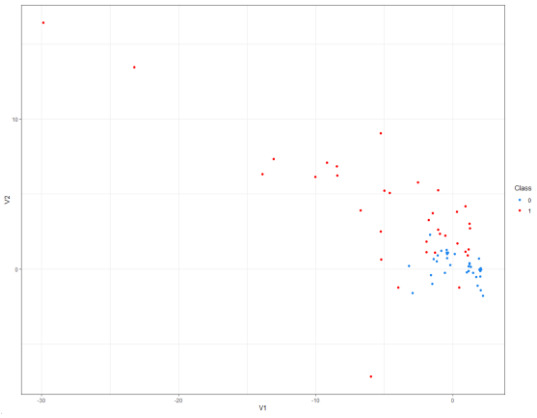
```
> oversampling_result <- ovun.sample(Class ~ . ,
+                                    data = train_data,
+                                    method = "over",
+                                    N = new_n_total,
+                                    seed = 2019)
> View(oversampling_result)
> oversampled_credit <- oversampling_result$data
> table(oversampled_credit$Class)

    0     1
22750 22750
> ggplot(data = oversampled_credit, aes(x = V1 , y =V2, col = Class)) +
+   geom_point(position = position_jitter(width = 0.1))+
+   theme_bw()+
+   scale_color_manual(values = c('dodgerblue2','red'))
>
```



As seen in the scattered plot above, the red point which are the fraudulent cases are still not showing to be more than the legitimate cases which are the blue points. This is because only duplicated points are created, a lot of the points are overlapping on one another due to the reason that random over sampling only create duplicate point which are already present in the dataset. Random over sampling only creates duplicate value and it has been suggested to be a condition that is not good.

A random under sampling has been use to reduce the number of legitimate cases, and make it equal to the number of fraudulent cases.

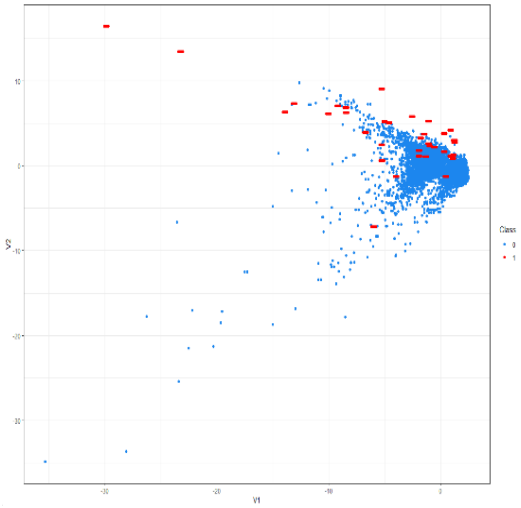
```
> #random under-sampling (RUS)
> table(train_data$Class)

    0     1
22750    35
> n_fraud <- 35
> new_frac_fraud <-0.50
> new_n_total <- n_fraud/new_frac_fraud # = 35/0.50
> new_n_total
[1] 70
> undersampling_result <- ovun.sample(Class ~ . ,
+                                     data = train_data,
+                                     method = "under",
+                                     N = new_n_total,
+                                     seed = 2019)
> undersampled_credit <- undersampling_result$data
> table(undersampled_credit$Class)

 0  1
35 35
> ggplot(data = undersampled_credit, aes(x = V1 , y = V2, col = Class)) +
+   geom_point()+
```



As seen in the scattered plot above, with random under sampling the number of legitimate cases has been reduced to less or equal to the number of fraudulent cases, although, this means that a lot of the data will be lost, and it will not be good for generalization.

Both random over sampling and under sampling were also use for the analysis as shown below

```
> n_new <- nrow(train_data) # = 22785
> fraction_fraud_new <- 0.50
> sampling_result <- ovun.sample(Class ~ . ,
+                                data = train_data,
+                                method = "both",
+                                N = n_new,
+                                p = fraction_fraud_new,
+                                seed = 2019)
> sampled_credit <- sampling_result$data
> table(sampled_credit$Class)

    0     1
11430 11355
> prop.table(table(sampled_credit$Class))

        0         1
0.5016458 0.4983542
> ggplot(data = sampled_credit, aes(x = V1 , y = V2, col = Class)) +
+   geom_point(position = position_jitter(width = 0.2)) +
+   theme_bw()+
+   scale_color_manual(values = c('dodgerblue2', 'red'))
```

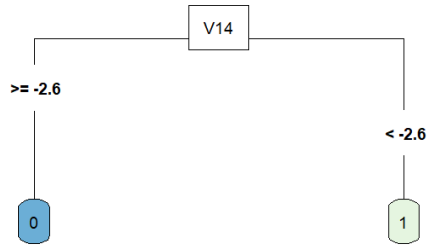


The result of using both random over sampling and under sampling indicate that 51% of the data are legitimate transaction and 49% of the data are fraud transection. Some of the numbers of the legitimate cases have been reduced, and the number of fraudulent cases have increase. Both are almost at the same ratio, although the amount of fraud cases are overlapping on one another due to the fact that they are just duplications, while the legitimate cases are scattered around the plot.

SMOTE method is used and this will allow synthetic samples in the dataset without the creation of duplicate, that is, with regard to this research, fraud cases.

With the use of SMOTE, the number of legitimate cases in the dataset is 60% and fraudulent cases is 40%. Using the SMOTE method, a lot of synthetic points have been added as indicate in the scattered plot above, Where the blue point represents class 0 which is the amount of legitimate cases, while the red point represent the number of fraud causes where synthetic points are added using the SMOTE method. With the creation of this, the model can therefore be trained on.

A decision tree is built using the SMOTE data created in other to predict whether a transaction is legitimate or fraudulent.



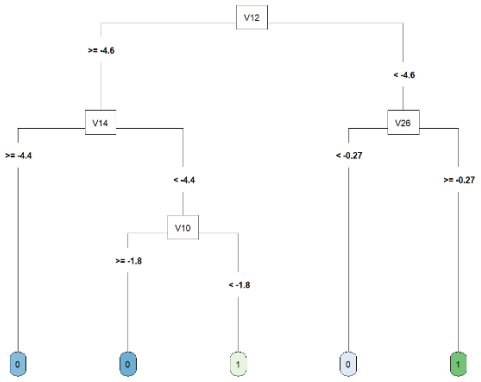
As indicated in the decision tree built with the SMOTE model above, V14 which is a column in the dataset has been used to classify different samples. Within this particular column V14, if the value is greater than -2.6 the transaction will be classified as legitimate transaction and if the value is less than -2.6 the transaction will be classified as fraudulent transaction.

Prediction was also made on the test sets to identify the number of samples that are correctly and incorrectly classified with a confusion matrix.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5625    2
         1   62    7

               Accuracy : 0.9888
                 95% CI : (0.9857, 0.9913)
    No Information Rate : 0.9984
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1772
```

The confusion matrix of the test data has shown below indicate a True positive of 5625 and a True negative of 7, therefore, out of 9 samples that are within the test data, 7 fraudulent cases were correctly classified by the model that was built using the smote data.

A decision tree without using the smote data was inspected to identify the number of samples that are correctly or incorrectly classified.



Above is the decision tree that the model will use to classify the various samples with regards to the legitimate or fraudulent cases.

A comparison was made using the whole data, which will therefore indicate the number of samples that will be correctly classified by the model that was built on smote and by the model that was built on the original data which is imbalanced data.

```
> CART_model <- rpart(Class ~ . , credit_smote)
> predicted_val <- predict(CART_model, credit_card [-1], type = 'class')
> confusionMatrix(predicted_val, credit_card$Class)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 28155     4
         1   282    40

               Accuracy : 0.99
                 95% CI : (0.9887, 0.9911)
    No Information Rate : 0.9985
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2164
```

Using the model that was built on credit smote a prediction was made on the whole credit card dataset, the confusion matrix indicated that out of the 44 samples that was in the dataset 40 fraud cases where correctly classified while using the smote data as seen above.

The same model was also use on the original train data without balancing factor that is SMOTE.

```
> #Decision tree without the use of SMOTE
> CART_model <- rpart(Class ~ . , train_data[, -1])
> predicted_val <- predict(CART_model, credit_card [-1], type = 'class
> confusionMatrix(predicted_val, credit_card$Class)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 28433     9
         1     4    35

               Accuracy : 0.9995
                 95% CI : (0.9992, 0.9998)
    No Information Rate : 0.9985
    P-Value [Acc > NIR] : 3.99e-08
```

the confusion matrix indicate that 35 fraudulent cases were detected. This indicate that 5 more samples can be detected when making use of SMOTE. Therefore, it can be suggested that making use of smote is preferable because it balances the dataset first before making prediction, which makes it easy for it to see variables to classify larger fraud cases using the smote that was 40, than the original dataset that is unbalanced.

REFERENCES

Alenzi, H. Z. and Aljehane, N. O. (2020) 'Fraud Detection in Credit Cards using Logistic Regression', *International Journal of Advanced Computer Science and Applications*, 11(12), pp. 540–551. doi: 10.14569/IJACSA.2020.0111265.

Alladi, T. *et al.* (2020) 'Consumer IoT: Security Vulnerability Case Studies and Solutions', *IEEE Consumer Electronics Magazine*, 9(2), pp. 17–25. doi: 10.1109/MCE.2019.2953740.

Credit Card Fraud Detection | Kaggle. Available at: https://www.kaggle.com/mlg-ulb/creditcardfraud (Accessed: 17 January 2022).

Dal Pozzolo, A. *et al.* (2018) 'Credit card fraud detection: A realistic modeling and a novel learning strategy', *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), pp. 3784–3797. doi: 10.1109/TNNLS.2017.2736643.

Gianini, G. *et al.* (2020) 'Managing a pool of rules for credit card fraud detection by a Game Theory based approach', *Future Generation Computer Systems*, 102, pp. 549–561. doi: 10.1016/j.future.2019.08.028.

Hudali, J. A. *et al.* (2019) 'Credit Card Fraud Detection by using ANN and Decision Tree', 2(3), pp. 1–4. Available at: https://ieeexplore.ieee.org/xpl/conhome.

Paschen, J., Kietzmann, J. and Kietzmann, T. C. (2019) 'Artificial intelligence (AI) and its implications for market knowledge in B2B marketing', *Journal of Business and Industrial Marketing*, 34(7), pp. 1410–1419. doi: 10.1108/JBIM-10-2018-0295.

Sahayasakila, V., Aishwaryasikhakolli, D. and Yasaswi, V. (2019) 'Credit card fraud detection system using smote technique and whale optimization algorithm', *International Journal of Engineering and Advanced Technology*, 8(5), pp. 190–192.

Sahoo, S. R. and Gupta, B. B. (2019) 'Classification of various attacks and their defence mechanism in online social networks: a survey', *Enterprise Information Systems*, 13(6), pp. 832–864. doi: 10.1080/17517575.2019.1605542.

Yousefi, N., Alaghband, M. and Garibay, I. (2019) 'A Comprehensive Survey on Machine Learning Techniques and User Authentication Approaches for Credit Card Fraud Detection', pp. 1–27. Available at: http://arxiv.org/abs/1912.02629.

Zainal, R., Md Som, A. and Mohamed, N. (2017) 'a Review on Computer Technology Applications in Fraud Detection and Prevention', *Management and Accounting Review (MAR)*, 16(2), p. 59. doi: 10.24191/mar.v16i2.671.

APPENDIX

```r
#Importing Dataset
credit_card <- read.csv(file.choose(),header=TRUE)

#List of library used
#library(caret)
#library(dplyr)
#library(ggplot2)
#library(caTools)
#library(ROSE)
#library(smotefamily)
#library(rpart)
#library(rpart.plot)

#looking at the structure of the dataset
str(credit_card)

#Covertion of Class column to a factor Variable where:0 = Legitimate, 1 =
Fraud
credit_card$Class <- factor(credit_card$Class, levels=c(0, 1))

#For dataset summary
summary(credit_card)

#To check for missing values
sum(is.na(credit_card))

#------------------------------------------------

#checking how Fraud and Legitimate are distributed within the dataset
table(credit_card$Class)

#Checking the Percentage of Fraud and Legitimate transaction in dataset
prop.table(table(credit_card$Class))


#Displaying the dataset for credit card fraud transaction with a pie chart
labels <- c("legitimate", "fraud")
labels <- paste(labels, round(100*prop.table(table(credit_card$Class)), 2))
labels <- paste0(labels, "%")

pie(table(credit_card$Class), labels, col = c("orange","red"),
    main ="Credit card Fraud transaction dataset Pie chart")

#----------------------------------------

#predictions with no model
predictions <- rep.int(0, nrow(credit_card))
predictions <- factor(predictions,levels =c(0 ,1)) # 0= legit, 1= Fraudd

#install.packages('caret')
library(caret)
confusionMatrix(data = predictions, reference = credit_card$Class)

#----------------------------------------

#Loading dplyr
library(dplyr)
#A smalll subset of the dataset is taken for faster computational merging -
#Then a model can be developed for the whole dataset
set.seed(1)
```

```r
credit_card <- credit_card %>% sample_frac(0.1)

#checking the distribution of class seed
table(credit_card$Class)
#plotting a scattered plot on V1 and V2
library(ggplot2)

ggplot(data = credit_card, aes(x = V1 , y = V2, col = Class)) +
  geom_point() +
  theme_bw()+
  scale_color_manual(values = c('dodgerblue2', 'red'))
#-------------------------------------------------

#For credit card Fraud Detection Model, a Training and Test set is created

#install.packages ('caTools')
library(caTools)
set.seed(123)

data_sample =sample.split(credit_card$Class,SplitRatio = 0.80)

train_data = subset(credit_card,data_sample==TRUE)
test_data = subset(credit_card,data_sample==FALSE)

#TO check the dimension of train data and test data
dim(train_data)
dim(test_data)

#-------------------------------------------------

#In other to balance the dataset different Techniques will be utilized
#First Technique is Random Over-sampling(ROS)

table(train_data$Class)

n_legit <- 22750
new_frac_legit <- 0.50
new_n_total <- n_legit/new_frac_legit # = 22750/0.50

#To perform a random Oversampling a package called 'ROSE' was instilled
#install.package('ROSE')
library(ROSE)
oversampling_result <- ovun.sample(Class ~ . ,
                                   data = train_data,
                                   method = "over",
                                   N = new_n_total,
                                   seed = 2019)
oversampled_credit <- oversampling_result$data
table(oversampled_credit$Class)

#plotting Scatter plot with gg function
ggplot(data = oversampled_credit, aes(x = V1 , y =V2, col = Class)) +
  geom_point(position = position_jitter(width = 0.1))+
  theme_bw()+
  scale_color_manual(values = c('dodgerblue2','red'))

#-------------------------------------------------

#random under-sampling (RUS)
table(train_data$Class)

```

```r
n_fraud <- 35
new_frac_fraud <-0.50
new_n_total <- n_fraud/new_frac_fraud # = 35/0.50

library(ROSE)
undersampling_result <- ovun.sample(Class ~ . ,
                                    data = train_data,
                                    method = "under",
                                    N = new_n_total,
                                    seed = 2019)
undersampled_credit <- undersampling_result$data
table(undersampled_credit$Class)

#Plotting a scattered plot to check the distribution classes of V1 and V2

ggplot(data = undersampled_credit, aes(x = V1 , y = V2, col = Class)) +
  geom_point()+
  theme_bw()+
  scale_color_manual(values = c('dodgerblue2', 'red'))
#---------------------------------------------------

# Both ROS and RUS where perfore to see how the data looks like

n_new <- nrow(train_data) # = 22785
fraction_fraud_new <- 0.50

sampling_result <- ovun.sample(Class ~ . ,
                               data = train_data,
                               method = "both",
                               N = n_new,
                               p = fraction_fraud_new,
                               seed = 2019)
sampled_credit <- sampling_result$data
table(sampled_credit$Class)
prop.table(table(sampled_credit$Class))

ggplot(data = sampled_credit, aes(x = V1 , y = V2, col = Class)) +
  geom_point(position = position_jitter(width = 0.2)) +
  theme_bw()+
  scale_color_manual(values = c('dodgerblue2', 'red'))
#---------------------------------------------------

#The utilization of SMOTE for dataset Balancing
#install.packages("smotefamily")
library(smotefamily)

table(train_data$Class)
n0 <- 22750
n1 <- 35
r0 <- 0.6

#For calculating the number of times to get SMOTE
#Value of dup_size parameter of SMOTE
ntimes <- ((1 - r0) / r0) * (n0 / n1) - 1

smote_output = SMOTE(X = train_data[ , -c(1, 31)],
                     target = train_data$Class,
                     K = 5,
                     dup_size = ntimes)
credit_smote <- smote_output$data
colnames(credit_smote)[30] <- "Class"
```

```r
prop.table (table(credit_smote$Class))

#The Original dataset classification distribution
ggplot(train_data, aes(x = V1, y =V2, color = Class)) +
  geom_point()+
  scale_color_manual(values = c('dodgerblue2','red'))

#The classification distribution of the over-sampled dataset using SMOTE
ggplot(credit_smote, aes(x = V1, y = V2, color = Class)) +
  geom_point() +
  scale_color_manual(values = c('dodgerblue2', 'red'))

#-----------------------------------------------------------

#Building a decision tree to check if the data is fraudlent or legitimate
#install.packages('rpart')
#install.packages('rpart.plot')

library(rpart)
library(rpart.plot)

CART_model <- rpart(Class ~ . , credit_smote)
rpart.plot(CART_model, extra = 0, type = 5, tweak = 1.2)

#prediction of fraudulent class
predicted_val <- predict(CART_model, test_data, type = 'class')

#build confusion matrix for prediction
library(caret)
confusionMatrix(predicted_val, test_data$Class)

#-------------------------------------------------------------

#Decision tree without the use of SMOTE
CART_model <- rpart(Class ~ . , train_data[, -1])
rpart.plot(CART_model, extra = 0, type = 5, tweak = 1.2)

#prediction of fraud classes
predicted_val <- predict(CART_model, test_data [,-1], type = 'class')

library(caret)
confusionMatrix(predicted_val, test_data$Class)

#-------------------------------------------------------------

# comparing the two model using the whole data
predicted_val <- predict(CART_model, credit_card [-1], type = 'class')
confusionMatrix(predicted_val, credit_card$Class)
```