FITE7410 Financial Fraud Analytics

Second Semester, 2022-2023

Assignment 1

Lee Chung Ho (3036030037)

Part I: EDA analysis on the dataset "Synthetic Financial Datasets For Fraud Detection"

Step 1: Distinguish and understand attributes

After reading the dataset into R, the first thing to do is to look into the internal structure and summary of the dataset.

Figure 1 in appendix shows the internal structure of the dataset. The dataset consists of 11 attributes, 8 in numeric type and 3 in character type, with 6,362,620 data entries. "isFraud" is the fraud class label of the dataset.

Figure 2 gives an overview of the data, in which basic statistics, such as mean, median, min, max values are shown. It is noticeable that the medians of the amount and account balance attributes are much smaller than their third quartile and maximum values.

Step 2: Univariate Analysis

In univariate analysis, we check the class balance and the transaction type balance.

Figure 3(a) shows the fraud to non-fraud ratio by a pie chart. The dataset is extremely unbalanced as 99.871% of data are in non-fraud class while only 0.129% in fraud class.

Figure 3(b) shows a distribution of transaction type. The most frequent transaction type is shown to be CASH_OUT(2237500), which is 5.4 times more than the least one is DEBIT(414324). These differences should be aware of during sampling procedures before modelling.

Step 3: Bi-/Multi-variate Analysis

First, we investigate the number frauds in different types of transactions. In figure 4, we can see that fraudulent transactions are only in 2 transaction types: CASH-OUT and TRANSFER. Modelling could be focused on these two types of transactions.

Secondly, we investigate transactions over time step attribute. Figure 5 shows the distribution of all, non-fraudulent and fraudulent transactions over time. Obviously, fraudulent transactions is approximately a uniform distribution over time, while non-fraudulent transactions have a non-uniform distribution, similar to the distribution of all transactions in the dataset. It implies that genuine transactions are not happening at the same rate in every hour, while fraudulent transaction does.

Thirdly, it is also important to notice how attribute correlation changes over classes. Figure 6 is the correlation plots between attributes for all, non-fraudulent and fraudulent transactions. Reading from (a) to (c), correlations do not change much from all to non-frauds, but changes drastically from non-frauds to frauds. For example, indicating by red arrows, correlations between "amount" and "newbalanceOrig", as well as "amount" and "oldbalanceOrig" differs a lot over fraud classes. These features could be used to strengthen classification models.

Step 4: Detect aberrant and missing values

Search results for aberrant and missing values are summarized in the red boxes in figure 7. No missing data, NA entries, or negative values are found in the dataset.

Step 5: Detect outliers

Boxplot is an efficient tool for outlier detection. In figure 8, with the outliers in red, it is clear that outliers are distributed differently between classes in "amount", "oldbalanceDest", "newbalanceDest"; there are many extreme outliers detected in genuine transactions but no extreme outliers are found in the above-mentioned 3 attributes of fraudulent transactions. It indicates that people may keep the transaction amount and account balances small to avoid attention and escape from fraud detection.

Step 6: Feature Engineering

Two features are created for further modelling.

Plotted in figure 9, time step modulo 24 is created as a feature to the distribution of transactions over a day. There are some hours, probably night time, that transactions happen much less frequently. This is a feature that cannot be seen solely by the time step column.

Apart from that, the character labels, such as "nameOrig" and "nameDest", are converted to numerical IDs and the total number of transactions by each account is also summarized. Figure 10 shows a table of some of the transaction recipient unique IDs with the number of transactions they are involved. These may be useful features because the same person may make fraudulent transactions through the same account for multiple times.

Part II: k-means clustering analysis on the dataset "Credit Card Fraud Detection Data"

Step 1: Data re-sampling

After reading the dataset into R, the dataset is split into a train and test dataset in a 7:3 ratio. In figure 11, the class tables of the original, train and test table are shown.

Since the dataset is highly unbalanced, the train dataset is re-sampled based on class label using 2 methods: Synthesized Minority Oversampling Technique and Random Over-Sampling Examples. Figure 12 shows that class is balanced after re-sampling. Class label and time step are then removed and the datasets are scaled for clustering.

Step 2: Find the optimal number of cluster

The elbow method compares the within-cluster similarity of kmeans models generated by different number of clusters, aiming at minimizing within-cluster sum of square error while preventing over-clustering. Figure 13 are the plots of the within-cluster sum of square against number of clusters. The sudden change of curve slope, i.e. elbow, indicated by red arrows, appears at 3 clusters for SMOTE and 4 clusters for ROSE.

Step 3: Build the model

Using the kmeans function from the stat package, the datasets are clustered in using the cluster number defined above.

Observations are represented by points in figure 14, using principal components. Edges of the cluster is drawn by lining up the outer-most points.

The cluster number and class labels are appended to the train dataset. Figure 15 shows the class in different clusters.

For SMOTE, frauds are mainly in cluster 1 and 2, so cluster 1 and 2 can be named fraud clusters, while cluster 3 can be named non-fraud cluster according as it has a much lower fraud-to-non-fraud ratio.

For ROSE, frauds are mainly in cluster 1, 2 and 4, so cluster 1, 2 and 4 can be named fraud clusters, while cluster 3 can be named non-fraud cluster according as it has a much lower fraud-to-non-fraud ratio.

Step 4: Make prediction using the trained models

The test dataset split at the beginning is fitted into the clusters defined in the models by finding the squared Euclidean distance from each sample to each cluster center.

The prediction result is shown in figure 16. Clearly, most of the fraudulent transactions are in cluster 2 for both methods. The test dataset is highly unbalanced, referring to figure 11(c), with only ~0.2% fraud to non-fraud ratio, but from figure 16, we are able to obtain a ratio of ~71% and ~66% in the models, revealing the power of clustering.

Step 5: Model evaluation

In the cluster-modelling results, there are clusters contain a large number of fraud after modeling but a small number of frauds in testing.

One reason may be both sampling methods have created new features in the dataset, which is a common drawback of re-sampling methods.

Another reason may be over-fitting; the model over-learned the features of the dataset, resulting in a difference between training and testing accuracy.

Quantitatively, we can evaluate the models using the confusion matrix of the prediction results in figure 17.

Both models predicts the fraudulent transactions quite well, with Recall (Ture+/sum(True+, False+)) close to 1. That means the models can correctly determine the real frauds. However, the precision (True+/sum(True+, False-)) of both models are not close to 1 because the models predicted a large number of genuine transaction to be fraudulent transactions, causing the F1 scores of both models also close to 0.

Though, SMOTE model is slightly better in terms of specificity (True-/sum(Ture-, False+)), mainly due to the higher non-fraud class prediction accuracy; SMOTE has 60512 True- but rose only has 43334 True-.

Step 6: Discussion

There are too many false alarms in the models. It will take a large cost for the credit card company to investigate into such a large number of "suspicious" cases.

If these models are implemented in real life, it is possible to improve the performance by, referring to figure 16, naming ONLY the cluster 2 of both models to be fraud cluster, and others to be non-fraud clusters. If we do so, we can obtain a precision/F1 score close to 1, but some frauds will be leaked to the non-fraud prediction class.

Another way to improve the models is to introduce some other modeling methods, such as logistic regression or ensemble, to perform within the clusters to further distinguish the frauds from the cluster.

Appendix

```
> str(PaySim)
spec_tbl_df [6,362,620 × 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ step          : num [1:6362620] 1 1 1 1 1 1 1 1 1 1 ...
 $ type          : chr [1:6362620] "PAYMENT" "PAYMENT" "TRANSFER" "CASH_OUT" ...
 $ amount        : num [1:6362620] 9840 1864 181 181 11668 ...
 $ nameOrig      : chr [1:6362620] "C1231006815" "C1666544295" "C1305486145" "C840083671" ...
 $ oldbalanceOrg : num [1:6362620] 170136 21249 181 181 41554 ...
 $ newbalanceOrig: num [1:6362620] 160296 19385 0 0 29886 ...
 $ nameDest      : chr [1:6362620] "M1979787155" "M2044282225" "C553264065" "C38997010" ...
 $ oldbalanceDest: num [1:6362620] 0 0 0 21182 0 ...
 $ newbalanceDest: num [1:6362620] 0 0 0 0 0 ...
 $ isFraud       : num [1:6362620] 0 0 1 1 0 0 0 0 0 0 ...
 $ isFlaggedFraud: num [1:6362620] 0 0 0 0 0 0 0 0 0 0 ...
```

Figure 1 Internal structure of dataset

```
> summary(PaySim)
      step           type              amount           nameOrig         oldbalanceOrg
 Min.   :  1.0   Length:6362620     Min.   :       0   Length:6362620     Min.   :       0
 1st Qu.:156.0   Class :character   1st Qu.:   13390   Class :character   1st Qu.:       0
 Median :239.0   Mode  :character   Median :   74872   Mode  :character   Median :   14208
 Mean   :243.4                      Mean   :  179862                      Mean   :  833883
 3rd Qu.:335.0                      3rd Qu.:  208721                      3rd Qu.:  107315
 Max.   :743.0                      Max.   :92445517                      Max.   :59585040
 newbalanceOrig       nameDest         oldbalanceDest      newbalanceDest         isFraud
 Min.   :       0   Length:6362620     Min.   :       0   Min.   :       0   Min.   :0.000000
 1st Qu.:       0   Class :character   1st Qu.:       0   1st Qu.:       0   1st Qu.:0.000000
 Median :       0   Mode  :character   Median :  132706   Median :  214661   Median :0.000000
 Mean   :  855114                      Mean   : 1100702   Mean   : 1224996   Mean   :0.001291
 3rd Qu.:  144258                      3rd Qu.:  943037   3rd Qu.: 1111909   3rd Qu.:0.000000
 Max.   :49585040                      Max.   :356015889   Max.   :356179279   Max.   :1.000000
 isFlaggedFraud
 Min.   :0.0e+00
 1st Qu.:0.0e+00
 Median :0.0e+00
 Mean   :2.5e-06
 3rd Qu.:0.0e+00
 Max.   :1.0e+00
```

Figure 2 Summary of dataset

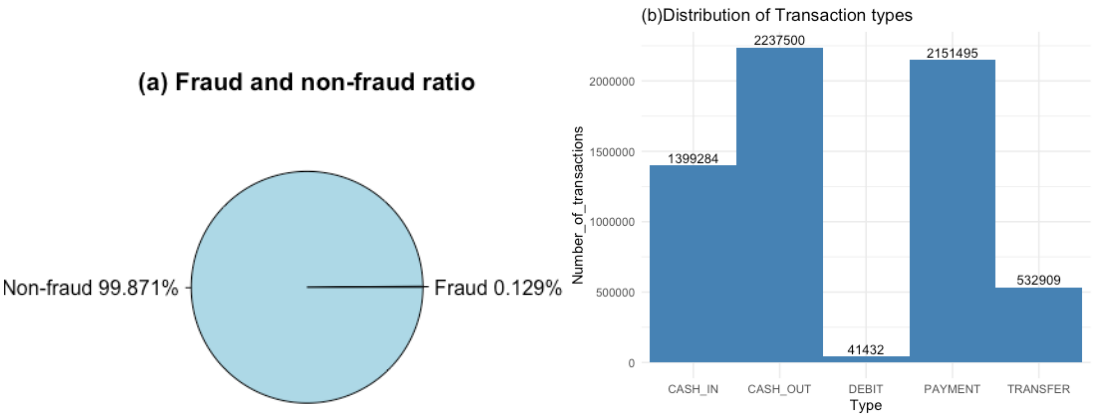

Figure 3 (a) Fraud-to-non-fraud ratio, (b) Distribution of transaction types

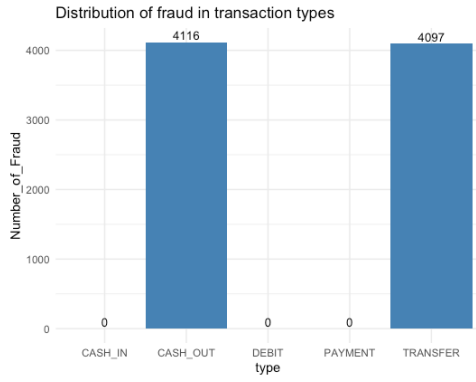

Figure 4 Distribution of fraud in transaction types

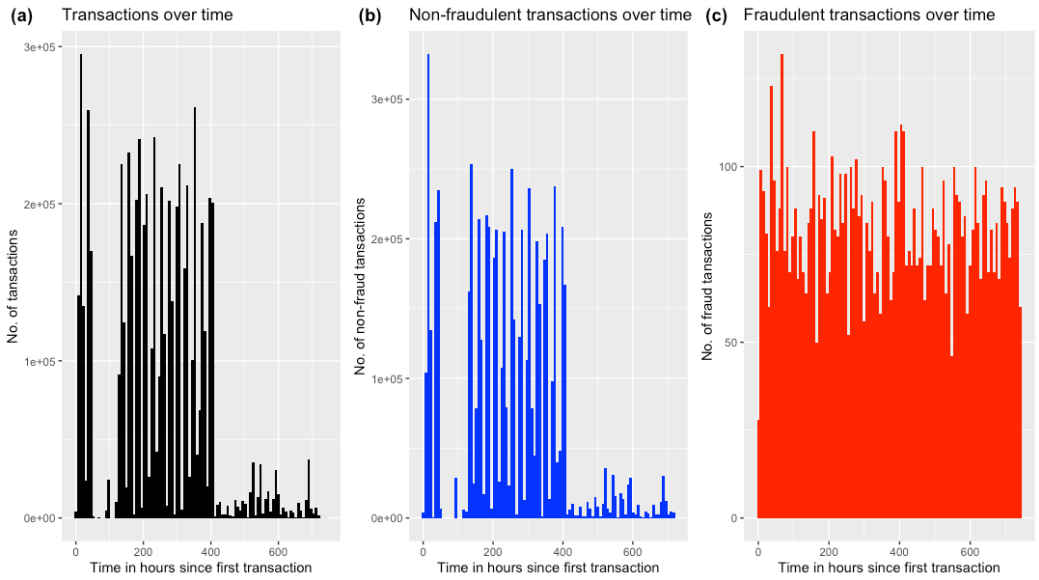

Figure 5 Distribution of (a) all (b) non-fraudulent (c) fraudulent transactions over time

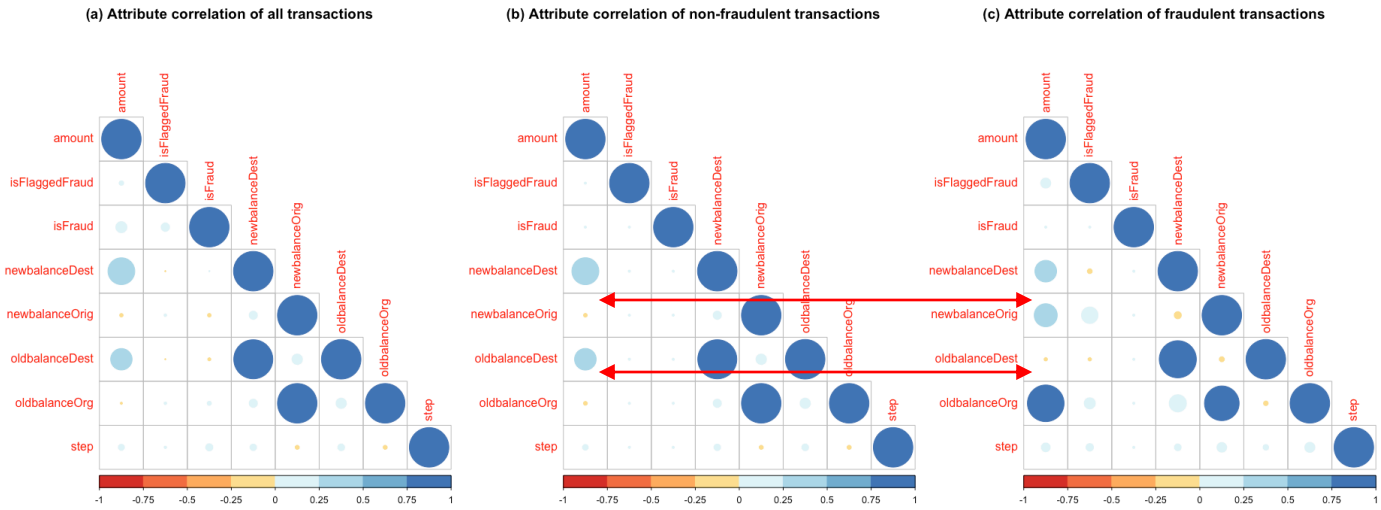

Figure 6 Attribution correlation in (a) non-fraud (b) fraud cases

```
> skim(PaySim)
── Data Summary ────────────────────
                           Values
Name                       PaySim
Number of rows             6362620
Number of columns          11
-------------------------
Column type frequency:
  character                3
  numeric                  8
-------------------------
Group variables            None

── Variable type: character ──────────────────────────────────────────────────────
  skim_variable n_missing complete_rate min max empty n_unique whitespace
1 type                  0             1   5   8     0        5          0
2 nameOrig              0             1   5  11     0  6353307          0
3 nameDest              0             1   2  11     0  2722362          0

── Variable type: numeric ────────────────────────────────────────────────────────
  skim_variable  n_missing complete_rate     mean           sd p0    p25     p50      p75       p100 hist
1 step                  0             1 2.43e+2     142.      1   156     239      335        743   ▇▇▁▁▁
2 amount                0             1 1.80e+5  603858.      0 13390.  74872.  208721.  92445517.   ▇▁▁▁▁
3 oldbalanceOrg         0             1 8.34e+5 2888243.      0     0   14208   107315.  59585040.   ▇▁▁▁▁
4 newbalanceOrig        0             1 8.55e+5 2924049.      0     0       0   144258.  49585040.   ▇▁▁▁▁
5 oldbalanceDest        0             1 1.10e+6 3399180.      0     0  132706.  943037. 356015889.   ▇▁▁▁▁
6 newbalanceDest        0             1 1.22e+6 3674129.      0     0  214661. 1111909. 356179279.   ▇▁▁▁▁
7 isFraud               0             1 1.29e-3       0.0359  0     0       0        0         1   ▇▁▁▁▁
8 isFlaggedFraud        0             1 2.51e-6       0.00159 0     0       0        0         1   ▇▁▁▁▁
> #Check NA values
> sum(is.na(PaySim))
[1] 0
> #Check if there are negative values
> sum(PaySim<0)
[1] 0
```

Figure 7 Results of aberrant and missing values search

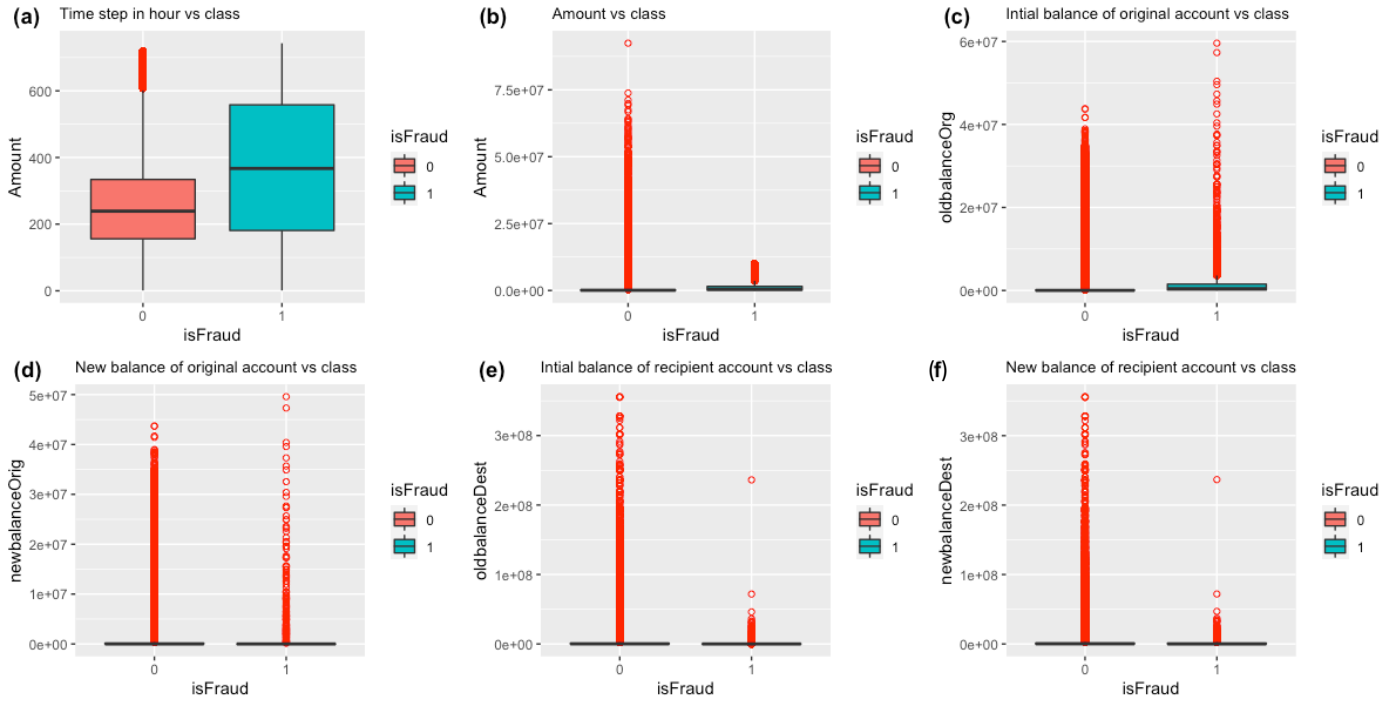

Figure 8 Boxplots of (a) step (b) amount (c) oldbalanceOrg (d) newbalanceOrig (e) oldbalanceDest (f) newbalanceDest vs class

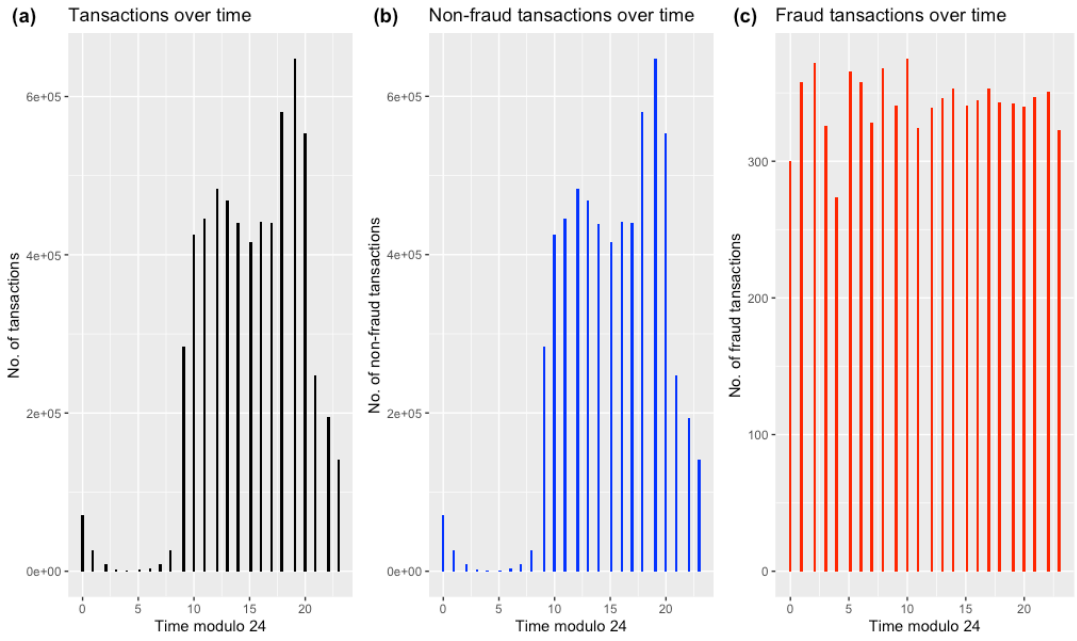

Figure 9 Distribution of (a) all (b) non-fraudulent (c) fraudulent transactions over a day

```
> table(namedest$nameDest_total_trans)
```

1	2	3	4	5	6	7	8
2262704	77054	57289	45115	36821	30416	25807	21900
9	10	11	12	13	14	15	16
18627	16343	14336	12385	10981	9709	8651	7711
17	18	19	20	21	22	23	24
6706	6097	5469	4914	4410	3941	3473	3174
25	26	27	28	29	30	31	32
2749	2614	2289	2066	1924	1755	1569	1413
33	34	35	36	37	38	39	40
1325	1128	1033	986	852	785	706	654
41	42	43	44	45	46	47	48
538	499	477	411	330	291	283	248

Figure 10 Examples of recipient ID with corresponding number of transactions

(a)
```
> table(Credit$Class)

     0      1
284315    492
```

(b)
```
> table(train$Class)

     0      1
199026    338
```

(c)
```
> table(test$Class)

     0      1
85289    154
```

Figure 11 Distribution of class in (a) original (b) train (c) test dataset

(a)
```
> table(smote_train$Class)

   0    1
1352 1014
```

(b)
```
> table(rose_train$Class)

    0     1
99455 99909
```

Figure 12 Distribution of class in train dataset after (a) SMOTE (b) ROSE

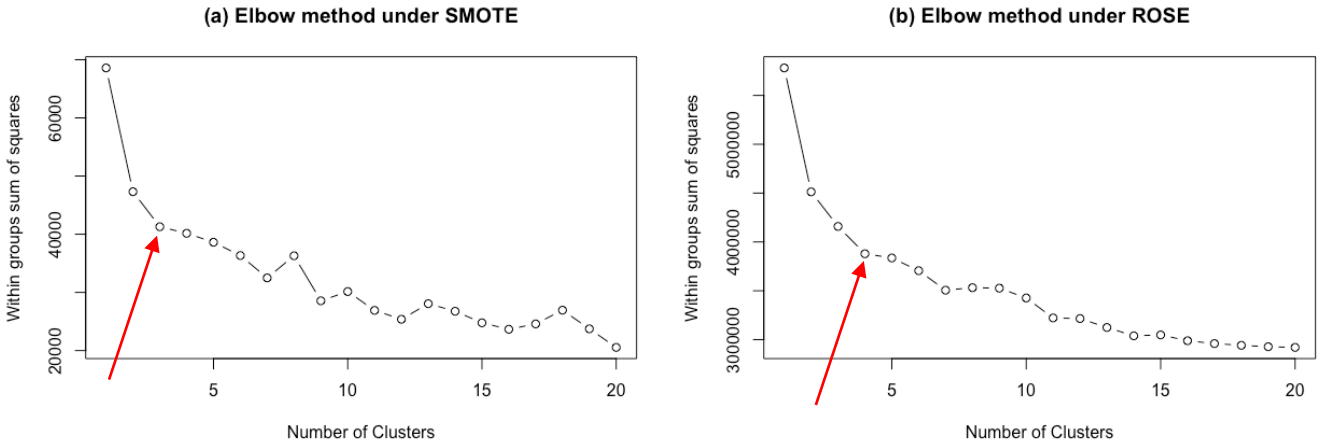

Figure 13 Elbow method using the (a) SMOTE (b) ROSE dataset

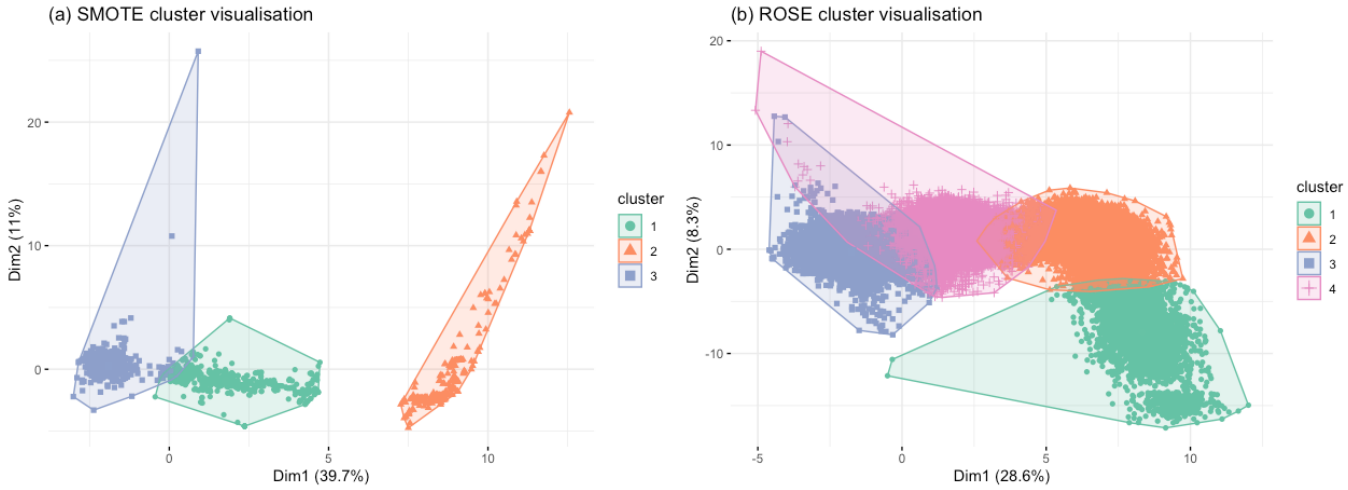

Figure 14 Elbow method using the (a) SMOTE (b) ROSE dataset

(a)
```
> table(kmeans_smote_3_with_class$cluster,
  kmeans_smote_3_with_class$fraud)

     0    1
1    0  548
2    0  231
3 1352  235
```

(b)
```
> table(kmeans_rose_with_class$cluster,
  kmeans_rose_with_class$fraud)

      0     1
1     2  3655
2     9 20788
3 99414 27426
4    30 48040
```

Figure 15 Distribution of class sorted by clusters using (a) SMOTE (b) ROSE dataset

(a)

```
> SMOTE_test_resul
  Non-fraud Fraud
1     24730    28
2        47   119
3     60512     7
```

```
> round(prop.table(SMOTE_test_result,margin=1)*100, digits=2)
     non_fraud Fraud
[1,]     99.89  0.11
[2,]     28.31 71.69
[3,]     99.99  0.01
```

(b)

```
> ROSE_test_result
  Non-fraud Fraud
1       193    10
2        57   112
3     43334     5
4     41705    27
```

```
> round(prop.table(ROSE_test_result,margin=1)*100, digits=2)
     non_fraud Fraud
[1,]     95.07  4.93
[2,]     33.73 66.27
[3,]     99.99  0.01
[4,]     99.94  0.06
```

Figure 16 Distribution of class in different clusters using (a) SMOTE (b) ROSE dataset

(a)

```
> smote_matrix
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 60512     7
         1 24777   147

               Accuracy : 0.7099
                 95% CI : (0.7069, 0.713)
    No Information Rate : 0.9982
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0082

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.954545
            Specificity : 0.709494
         Pos Pred Value : 0.005898
         Neg Pred Value : 0.999884
              Precision : 0.005898
                 Recall : 0.954545
                     F1 : 0.011723
             Prevalence : 0.001802
         Detection Rate : 0.001720
   Detection Prevalence : 0.291703
      Balanced Accuracy : 0.832020

       'Positive' Class : 1
```

(b)

```
> rose_matrix
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 43334     5
         1 41955   149

               Accuracy : 0.5089
                 95% CI : (0.5056, 0.5123)
    No Information Rate : 0.9982
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0035

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.967532
            Specificity : 0.508084
         Pos Pred Value : 0.003539
         Neg Pred Value : 0.999885
              Precision : 0.003539
                 Recall : 0.967532
                     F1 : 0.007052
             Prevalence : 0.001802
         Detection Rate : 0.001744
   Detection Prevalence : 0.492773
      Balanced Accuracy : 0.737808

       'Positive' Class : 1
```

Figure 17 Confusion matrix and statistics on prediction result using (a) SMOTE (b) ROSE dataset