Unsupervised learning approach to Credit card Fraud

Joshua Dalphy, Choongil Kim, Gouri Kulkarni, Jinping Bai

October 18, 2020

Business Introduction

Background

What is Fraud ? Fraud is the intentional perversion of truth in order to induce another to part with something of value or to surrender a legal right, an act of deceiving or misrepresenting, a wrongful or criminal deception intended to result in financial or personal gain. A fraudulent person is one who is not what he or she pretends to be.

Day by day, we are becoming an increasingly cashless society. The World Payments Report from Capgemini is the leading source for data, trends and insights on global and regional non-cash payments, the key regulatory and industry initiatives (KRIIs), and today's dynamic payments environment. The number of consumers who make 51–100% of monthly purchases via e-commerce nearly doubled during the pandemic, and the transition from retail to e-commerce will continue even after the virus has been contained.

Banks that offer a seamless, secure, and speedy digital interface will see a positive impact on revenue, while those that don't will erode value and potentially lose business. Modern banking demands faster risk decisions (such as real-time payments) so banks must strike the right balance between managing fraud and handling authorized transactions instantly.

Undetected fraud is becoming costlier day by day due to the increasing number of non cash transactions.

One of the drivers of cyber crime is to gain and misuse credit card information. With the growing number of threats from vulnerabilities such as new technologies and increasing transaction volumes, credit card issuers need to obtain a complete view of financial crime as it evolves.

Banks and credit card companies take fraud very seriously, and have highly sophisticated security systems and teams of experts in place to monitor transactions, protect customers and prevent and detect credit card fraud. In response to fraud, the bank must:

-identify and authenticate the customer

-monitor and detect transactions and behavioral anomalies

-respond to and mitigate risks and issues

Credit card issuers need to detect fraud almost in realtime and report it to the cardholder.

There is competition among banks in being the fastest and best at promptly reporting fraud to the cardholder and being more effective in preventing future fraudulent activities by taking appropriate action.

Credit card issuers offer complimentary fraud detection services to proactively notify clients of potential fraud to your account. Also they need to come up with fraud detection and prevention plans.

Data mining techniques have been used for suspicious transaction monitoring for years. However, with the growing number of transactions, the volume of data that needs to be ingested, understood, analyzed and modeled is also increasing. New trends arise by the hour and the number of dimensions that need to be addressed is forever changing.

Clustering techniques have been widely used fo detecting fraud. Clusters formed after analysis can identify transactions which are low risk, high risk, extremely risky and so on which can be inputs for the credit card issuer to apply to systems and tools to communicate to the client.

An example of a red flag could be an online purchase made at a merchant in another country if the client does not have prior such history. In terms of transactions, an outlier is an observation which is different from others and generates suspicion -is it genuine, or a fradulent transacton. Unlike supervised learning our concern is not the history of the data, but observations which are different from normal observations.

Clustering

Clustering is the process of grouping a set of data objects into multiple groups of clusters so that objects within a cluster have high similarity , but are very dissimilar to objects in other clusters. Dissimilarities and similarities are assesses based on the attribute values describing the objects and often involve distance or other measures. The goal is to discover groupings.

Different clustering methods can result in different clustering outputs of the same dataset.

Clustering can be used to gain insight into data distribution , observe characteristics of each cluster and focus on a particular set of clusters for further analysis. Or, it can be a preprocessing step for other algorithms. Another name for clustering is automatic classification, or data segmentation. When used for outlier detection, where outliers may be more interesting than common cases. Credit card fraud detection is an example use case of outlier detection.

Requirements of clustering algorithms

When used to solve business problem, clustering algorithms should be:

-scalable to changing dataset sizes

-able to deal with increasing attributes

-ability to visualize clusters of various shapes

-domain knowledge and understanding of the business

-ability to deal with noisy data

-ability to perform incremental clustering, insensitivity to input order

-ability to cluster highly dimensional data

-ability to satisfy constraints

-interpretability and usability

-partitioning criteria

-separation of clusters

-ability to measure similarity

-ability to define cluster subspaces

Clustering algorithms that address these points can provide meaningful insights.

Clustering methods

Partitioning methods are mostly distance based. They construct k partitions(clusters) of the data from n objects (dataset observations) k-means , k-medoids are examples of algorithms using partitioning methods and are demonstrated here. They can be used to find mutually exclusive clusters of spherical shape and may use cluster centers. They work well on small to medium sized data sets. The starting point is the number of clusters. The partitioning algorithm organizes the objects into k partitions (k<= n) where each partition represents a cluster.

Evaluation of clustering analysis methods

After having tried out clustering on a dataset, how do we evaluate whether the clustering results are good?

Evaluation goals:

-assess clustering tendency : does a non random structure exist in the data? Clusters obtained from blindly applying a clustering method on a dataset will return misleading clusters Hopkins statistic : given a dataset D, how far is a random variable from being uniformly distributed in the data space If h >.5, then it is unlikely that the dataset has statistically significant clusters

-determine the number of clusters in the dataset estimating the optimum number of clusters before using the clustering algorithm determining the right number of clusters depends on the distribution shape, scale, and clustering resolution required. 1.Elbow method - increasing the number of clusters can help reduce the sum of within cluster variance of each cluster. The turning point of the curve of the sum of within -cluster variances is used 2.Cross validation method :divide the dataset into m parts , use m-1 parts to build a clustering model , use the remaining part to test the quality of the cluster.

-measure cluster quality How well do the clusters fit the dataset? How well do the clusters match the ground truth?

-cluster scores comparing two clustering results on the same dataset. We examine how well the clusters are separated and how compact the clusters are. The silhoutte coefficient uses the average distance between each object and all other objects in the cluster , It lies between -1 and 1. The average silhoutte coefficient value of all objects in the dataset can be used to measure the quality of a clustering.

Clustering methods used for more than 10 dimensions are :

1.subspace clustering

2.dimensionality reduction

We have attempted to demonstrate dimensionality reduction here.

In our case, we want to cluster transactions based on attributes.

A transaction is a vector of attributes. Our dataset has 31 attributes.

As the number of dimensions increases , noise increases. Clusters created using traditional distance measures may not be meaningful for high dimensional data.

We use the following approaches on data with higher dimensions:

-subspaces

-dimensionality reduction

Subspace clustering methods search for clusters with a similarity that is measured using conventional metrics such as distance or density

Correlation based clustering methods use PCA. They apply PCA to derive a set of new , uncorrelated dimensions.

With Biclustering methods, the transaction-attributes matrix can be analyzed in two dimensions, the transaction dimension and characteristics dimension

1. treating each transaction as an object and characteristic as attribute, we can find groups of transactions that have similar characteristics
2. using the characteristics as objects and transactions as attributes, we can find groups of characteristics and the transaction that fall within them

Data Understanding

The data set for this project was obtained from the data.world website, https://data.world/raghu543/credit-card-fraud-data

The datasets contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation.

PCA used for dimension reduction consists of the following steps:

1. data is standardised , normalized
2. calculate covariance matrix
3. calculate eigen vectors
4. sort eigen vectors
5. select the first k eigen vectors which will be the new k dimensions

Due to confidentiality issues, we do not have the original features and background information about the data. The labeled variables are Time, Amount and Class. V1 -V28 variables are anonymized to protect sensitive user information. We do not know the labels for these variables.

The input data has been collected over a short period of two days and sent for analysis.

Class identifies a transaction as Normal or Fraudulent and could be the output of prior supervised learning.

Amount is the amount of the transaction.

Data Exploration

The dimensions of the dataset

```
## [1] 284807     31
```

Explore the types of each variable

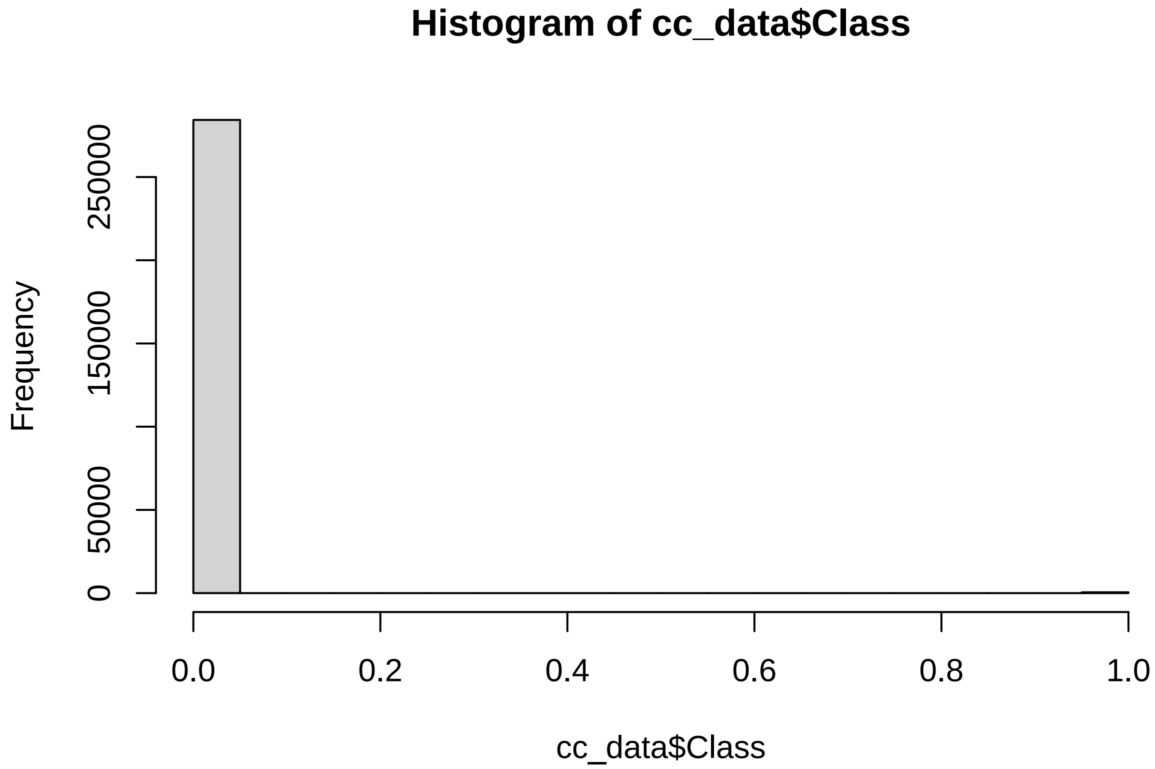
```
## Classes 'data.table' and 'data.frame':   284807 obs. of  31 variables:
##  $ Time  : num  0 0 1 1 2 2 4 7 7 9 ...
##  $ V1    : num  -1.36 1.192 -1.358 -0.966 -1.158 ...
##  $ V2    : num  -0.0728 0.2662 -1.3402 -0.1852 0.8777 ...
##  $ V3    : num  2.536 0.166 1.773 1.793 1.549 ...
##  $ V4    : num  1.378 0.448 0.38 -0.863 0.403 ...
##  $ V5    : num  -0.3383 0.06 -0.5032 -0.0103 -0.4072 ...
##  $ V6    : num  0.4624 -0.0824 1.8005 1.2472 0.0959 ...
##  $ V7    : num  0.2396 -0.0788 0.7915 0.2376 0.5929 ...
##  $ V8    : num  0.0987 0.0851 0.2477 0.3774 -0.2705 ...
##  $ V9    : num  0.364 -0.255 -1.515 -1.387 0.818 ...
##  $ V10   : num  0.0908 -0.167 0.2076 -0.055 0.7531 ...
##  $ V11   : num  -0.552 1.613 0.625 -0.226 -0.823 ...
##  $ V12   : num  -0.6178 1.0652 0.0661 0.1782 0.5382 ...
##  $ V13   : num  -0.991 0.489 0.717 0.508 1.346 ...
##  $ V14   : num  -0.311 -0.144 -0.166 -0.288 -1.12 ...
##  $ V15   : num  1.468 0.636 2.346 -0.631 0.175 ...
##  $ V16   : num  -0.47 0.464 -2.89 -1.06 -0.451 ...
##  $ V17   : num  0.208 -0.115 1.11 -0.684 -0.237 ...
##  $ V18   : num  0.0258 -0.1834 -0.1214 1.9658 -0.0382 ...
##  $ V19   : num  0.404 -0.146 -2.262 -1.233 0.803 ...
##  $ V20   : num  0.2514 -0.0691 0.525 -0.208 0.4085 ...
##  $ V21   : num  -0.01831 -0.22578 0.248 -0.1083 -0.00943 ...
##  $ V22   : num  0.27784 -0.63867 0.77168 0.00527 0.79828 ...
##  $ V23   : num  -0.11 0.101 0.909 -0.19 -0.137 ...
##  $ V24   : num  0.0669 -0.3398 -0.6893 -1.1756 0.1413 ...
##  $ V25   : num  0.129 0.167 -0.328 0.647 -0.206 ...
##  $ V26   : num  -0.189 0.126 -0.139 -0.222 0.502 ...
##  $ V27   : num  0.13356 -0.00898 -0.05535 0.06272 0.21942 ...
##  $ V28   : num  -0.0211 0.0147 -0.0598 0.0615 0.2152 ...
##  $ Amount: num  149.62 2.69 378.66 123.5 69.99 ...
##  $ Class : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Raw data understaing

We have now looked at our raw data and are ready to explore the data

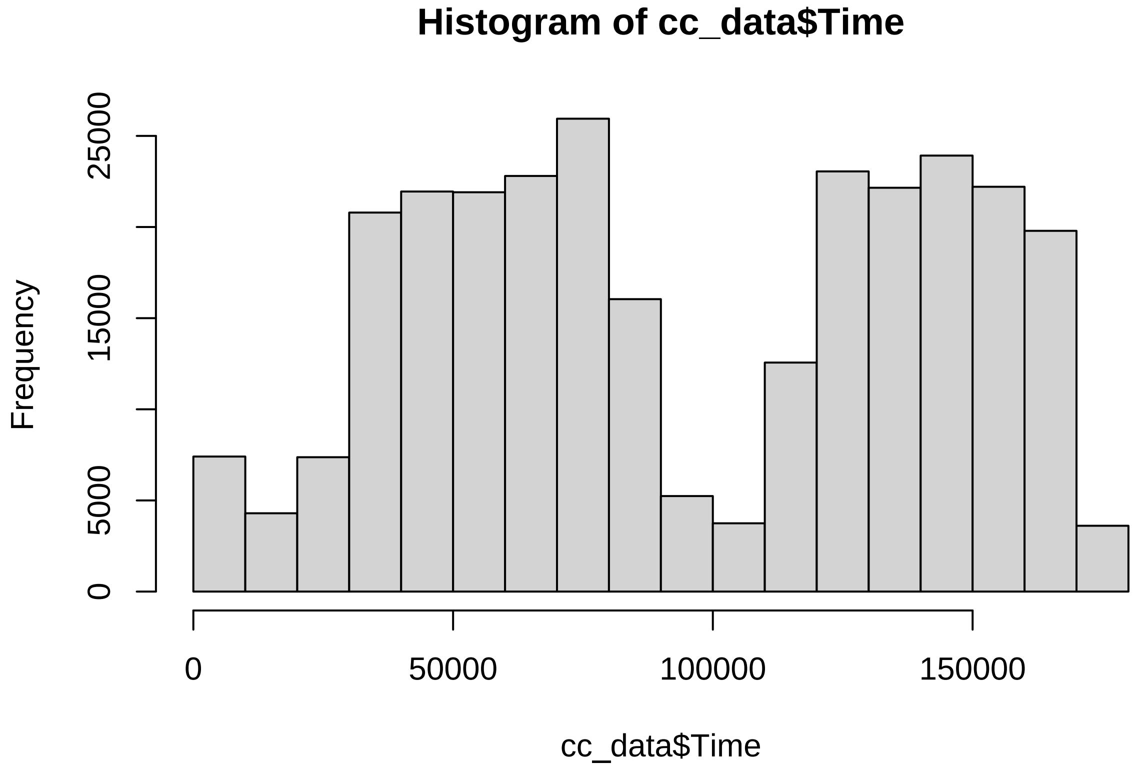
Variable Exploration

Explore the frequency of Class (Normal / Fraud)

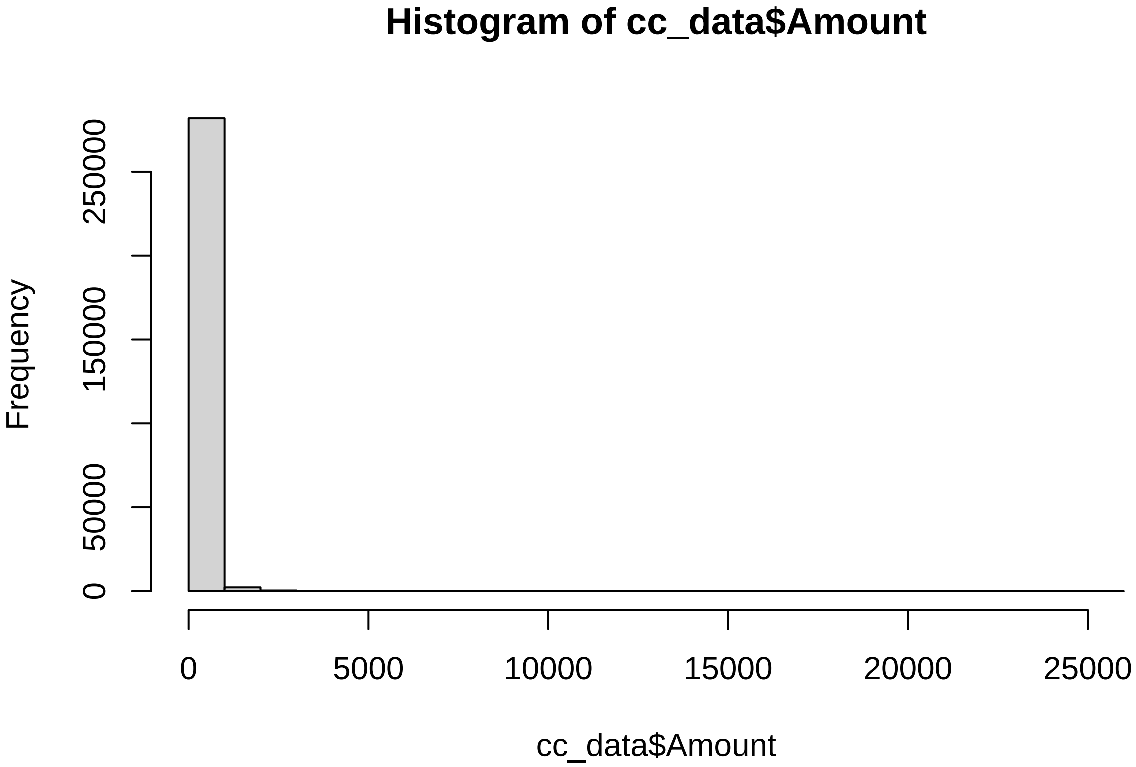


Most transactions are not fraudulent, only a few are fraudulent. However, not indentifying the fraudulent cases in time puts the card issuer at risk for fraud liability and can cost the card issuer millions of dollars.

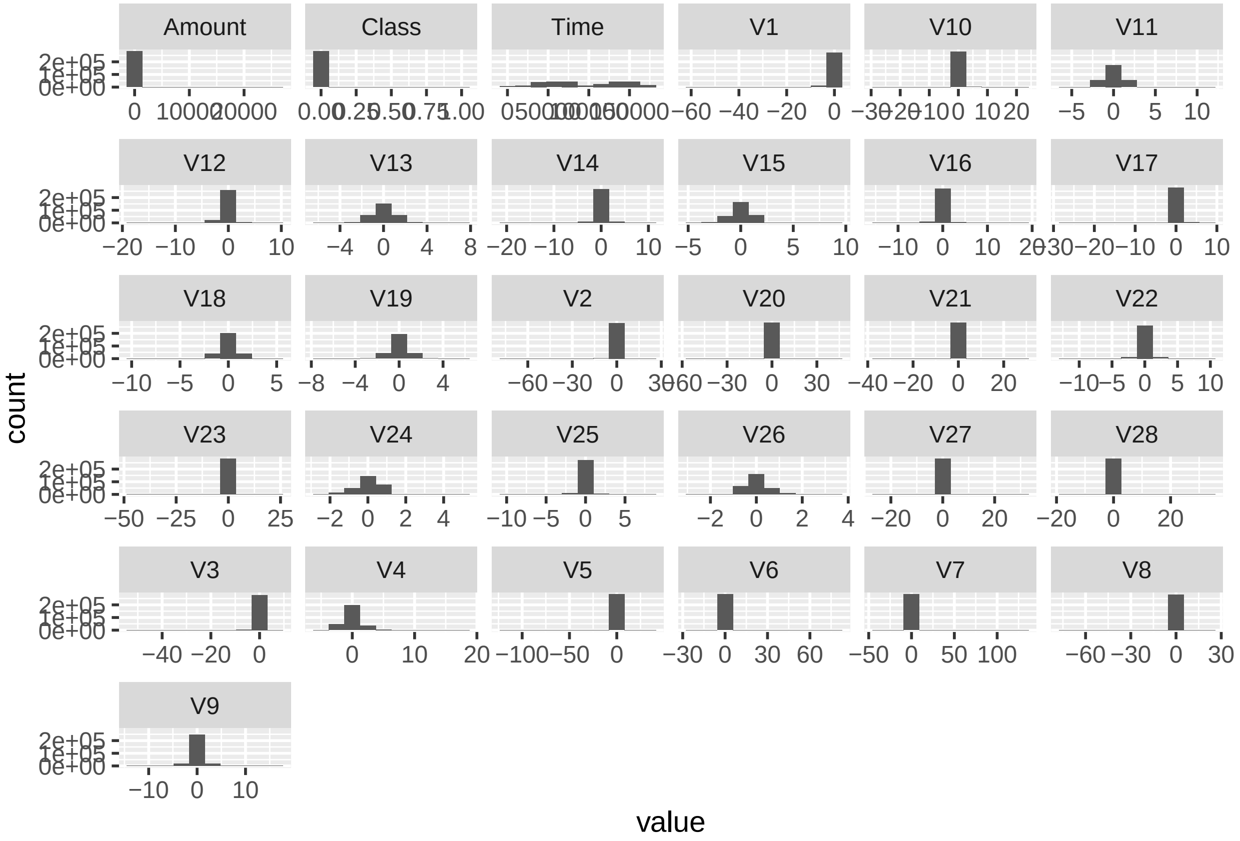
Time : Most transactions happen during a certain time of the day



Amount - Most transactions are small amounts

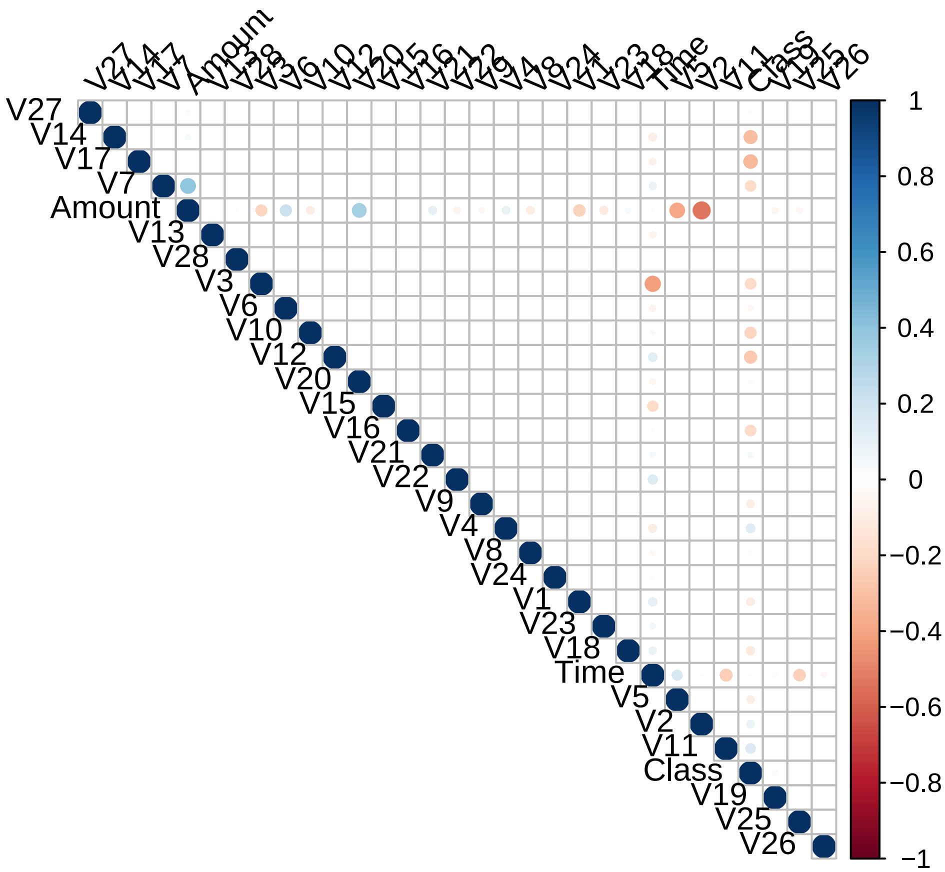


Histogram distribution of all variables



We do not know what the variables V2 - V28 stand for. They are characteristics of each transaction. We still need to know the correlation between the variables.

Simple correlation matrix

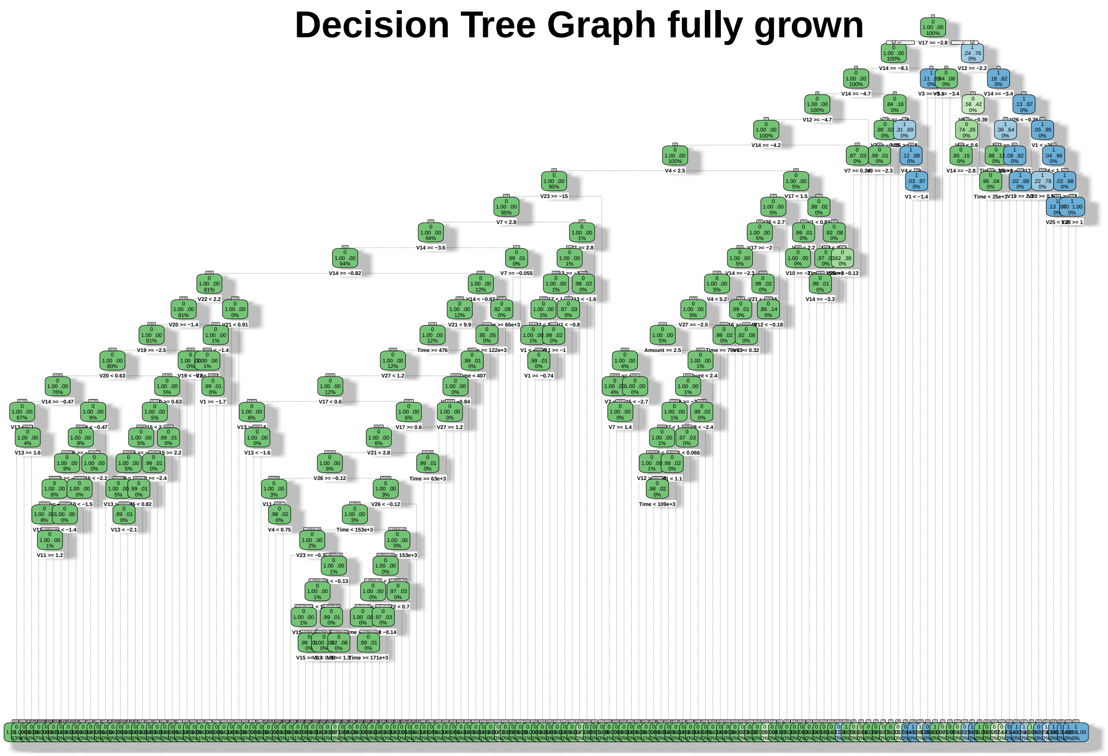


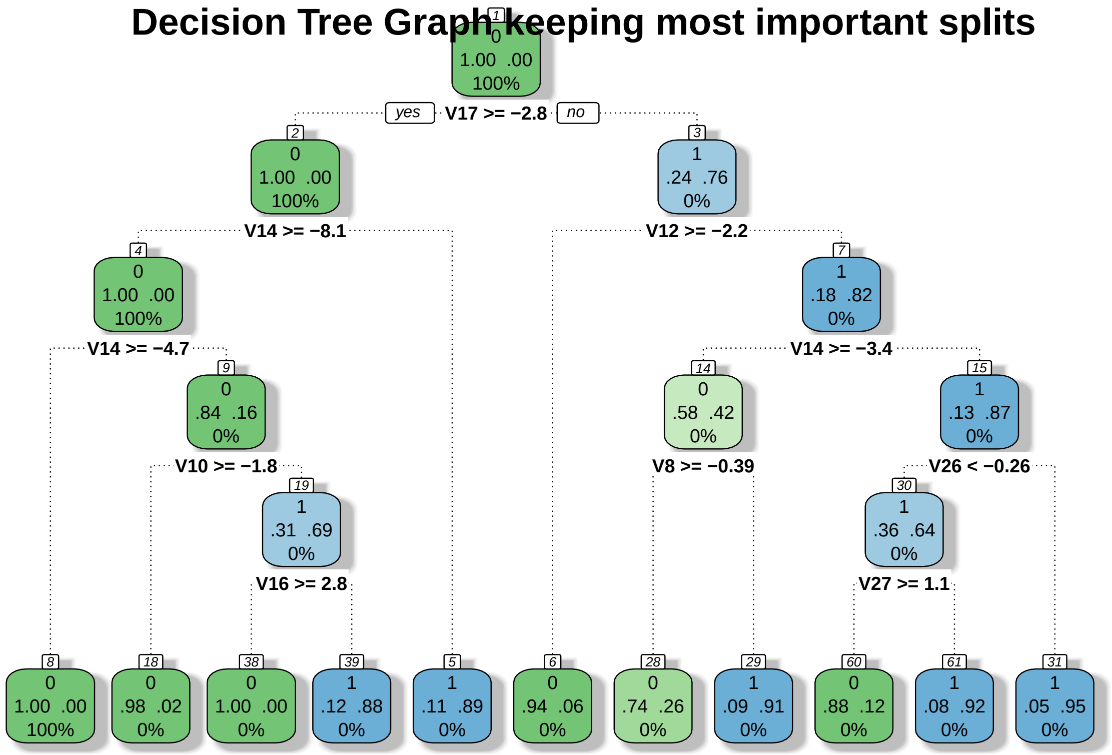
The result of the correlation will be used later when using PCA on the high dimensional data set (29 variables)

Decision trees for anomaly detection

Credit card fraud is an anomalous event, we indentify events that don't match the expected pattern. How can decision trees help analyze the events? Decision trees don't make an assumption about the data. They can be used to understand the variables. Decision tree is a machine-learning algorithm that can be a classification or regression tree analysis. The decision tree can be represented by graphical representation as a tree with leaves and branches. The leaves are generally the data points and branches are the condition to make decisions for the class data set. rpart() is used to create the decision trees in R. The output of rpart shows that a certain combination of variables result in a certain event (leaf)

The root node is the Class label, we then have decision nodes and sub-nodes.



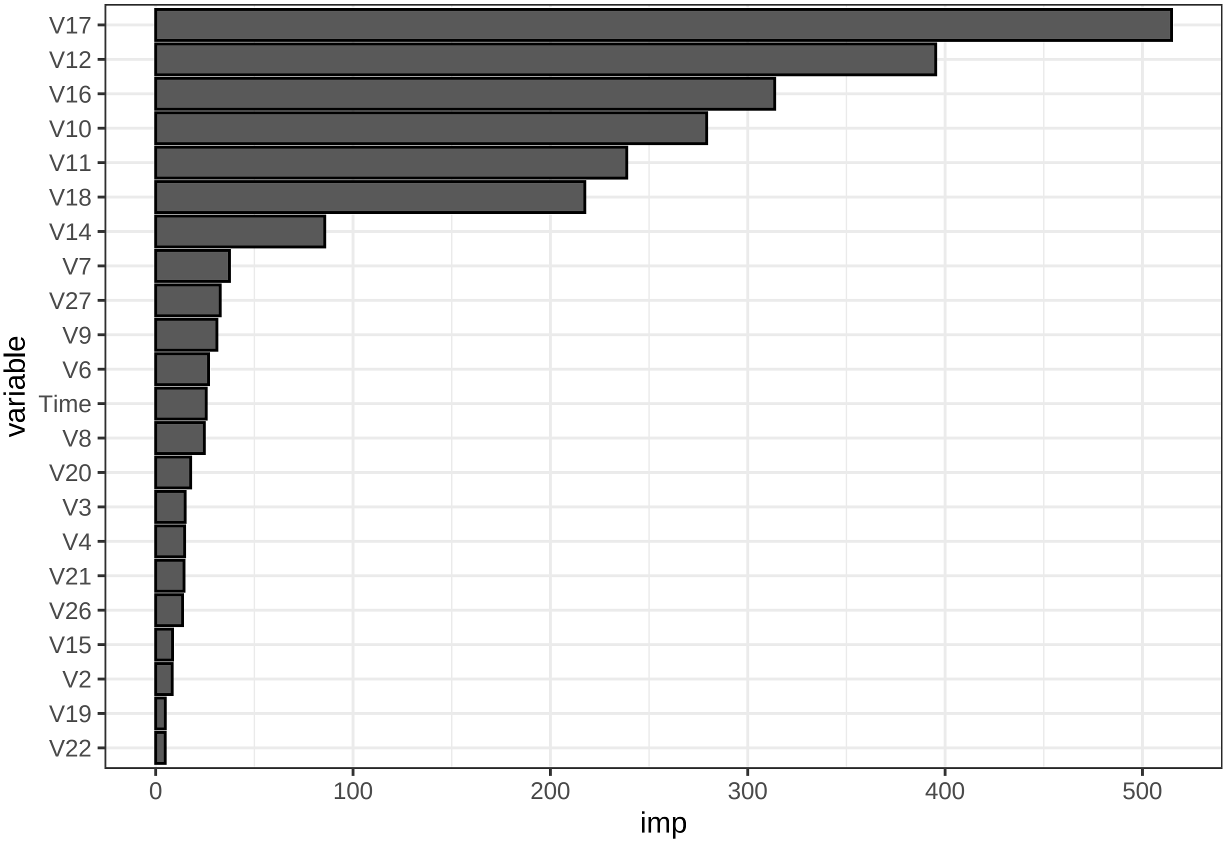


The algorithm has determined that V17 was the best variable for further analysis. Down the tree,it picks other variables and outputs leafs. The entire tree shows the relation,importance of variables,choice of variables which result in a transaction being labeled as Fraud or Normal.

We ran the decision tree on the full dataset , and again removing the complexities, producing the two trees.

Feature importance

Using the decision tree with complexities removed, we get the histogram showing feature importance



Plot of variable importance

Without the aid of machine learning, the best way to identify a fraudulent event would be to hire an investigator to manually monitor the activities on an account. Card issuers have fraud investigation departments for this purpose. The challenge posed by big data and ever increasing volumes is that issuers need to strategically decide which transactions to investigate. The decision tree model comes useful based off some initial features. If the bank wants to aggressively investigate transactions, the investigating team can use the output from decision trees.

Outlier detection

In fraud detection we pay special attention to transactions that are different from typical cases. We identify suspicious cases. Our goal is to detect such transactions as soon as they occur and notify the cardholder. For example, when a credit card is compromised, the transaction behaviour is different from normal.

Outlier detection or anomaly detection is the process of finding data objects with behaviours different from normal.

Outlier detection and clustering analysis are related tasks. Outlier detection as the first step , captures the exceptional cases.

Noise is not the same as an outlier. Noise in a transaction could be an unusually large transaction and detecting nois as outliers could lead to errors or false alarms and losses for the credit card issuer.

Outliers could be -Global, using the whole dataset -Contextual, using certain variable as context -Collective, using a collection of objects as a whole to form an outlier

Approaches for outlier detection are unsupervised (clustering analysis when we have no prior knowledge of the data and labels), supervised (modeling using classification on labeled data), semisupervised (modeling using labeled data that is normalized)

Our data does not have labels for the outliers. We use the unsupervised approach.

Outlier detection methods could be univariate or multivariate, parametric(statistical) or non parametric(model free), distance based or clustering techniques).

The output of the outlier detection can be a label or a score

Using Grubb's test : This is a statistical test. According to Grubb's, an outlier is one that appears to deviate markedly from other members of the sample. Causes of outliers could be Human errors, Instrument errors, Experimental errors, Intentional errors, Data processing errors, Sampling errors or Natural.

In our use case, the outliers , faudulent transactions are a natural event, and we are actually exploring the outliers. A customer's purchase behaviour can be modeled as a random variabe. A customer may generate some "noise transactions" which could seem like random errors. Treating them as outliers could prove costly to the bank.

The R outliers package, (https://www.rdocumentation.org/packages/outliers/versions/0.14) has a series of functions that can be used to test for outliers.

Grubb's test can be used on numerical variables. Our dtaa consistes of numerical variables.

From the histograms, we see the normal patterns in the data.

Grubb's test lets us sequentially identify outliers. The algorithm first considers the data value with the highest absolute value. If the null hypothesis that such a value is not an outlier is rejected, the considered value is detected as an outlier and excluded from further analysis. Subsequently, a value with the second-highest absolute value is considered, and its quality is again evaluated using the Grubbs test. This procedure is repeated until no outlier is detected.

Thus Grubbs' test assesses whether the value that is farthest from the mean is an outlier - the value could be either the maximum or minimum value. Grubbs test can be run on each variable and a function can be written to automate runnuing the test on each variable. We have 29 variables. Grubbs test assumes the data is normally distributed

Performed on one variable V1:

```
##
## Attaching package: 'outliers'
```

```
## The following object is masked from 'package:randomForest':
##
##     outlier
```

```
##
##  Grubbs test for one outlier
##
## data:  cc_data$V1
## G = 28.79844, U = 0.99709, p-value < 2.2e-16
## alternative hypothesis: lowest value -56.407509631329 is an outlier
```

```
## [1] 39768
```

```
## [1] 193546
```

Grubbs tests can be performed with Type = 10, 11 or 20 We can specify how we want the outliers detected.

Various other algorithms can be used for anomaly detection, such as KNN.

Clustering analysis

We demonstrate three partitioning methods :

1. Heirarchical clustering
2. Kmeans clustering
3. Kmedoids clustering

Ingesting the data, we explore the structure and missing values

Hierarchical Clustering

For simplicity, we conducted clustering analysis on the first 10,000 observations

Simply stated, clustering is a technique which is used to group similar data points in a manner which leads to all points belonging to the same group to be more similar to each other. Each distinct group of in the data is referred to as a cluster. Hierarchical clustering is one of the more popular clustering techniques and can be subdivided into two categories:

1. Agglomerative
2. Divisive

For agglomerative clustering, each data point is initially considered as an individual cluster. For each iteration, similar clusters are merged until either one or k clusters remain. The steps of the agglomerative clustering technique are outlined below:

1. Compute the proximity matrix
2. Let each data point be a cluster
3. Merge the two closest clusters
4. Update the proximity matrix
5. Repeat steps 1-4 until either one or k clusters remain

For divise hierarchical clustering, all the data points are initially considered to be a single cluster. For each iteration, data points which are not similar are separated from the original cluster. Once the algorithm has finished running, there will be k clusters left.

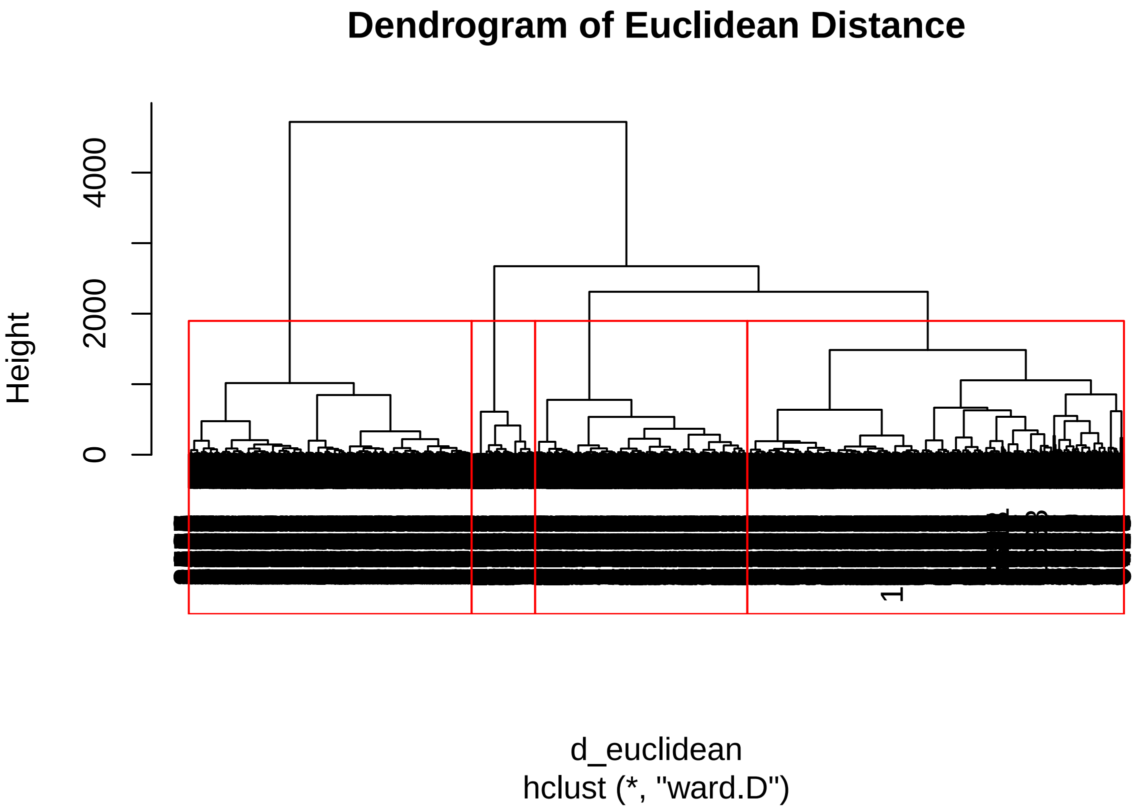
For the current application, the agglomerative clustering approach was used and we investigated the technique using both the Euclidean and Manhattan method to calculate the distance. Given a set of two points, A and B, whose coordinates are (x1,y1) and (x2,y2) the Euclidean distance can be calculated using the equation below:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The Manhattan distance can be calculated using the following equation:

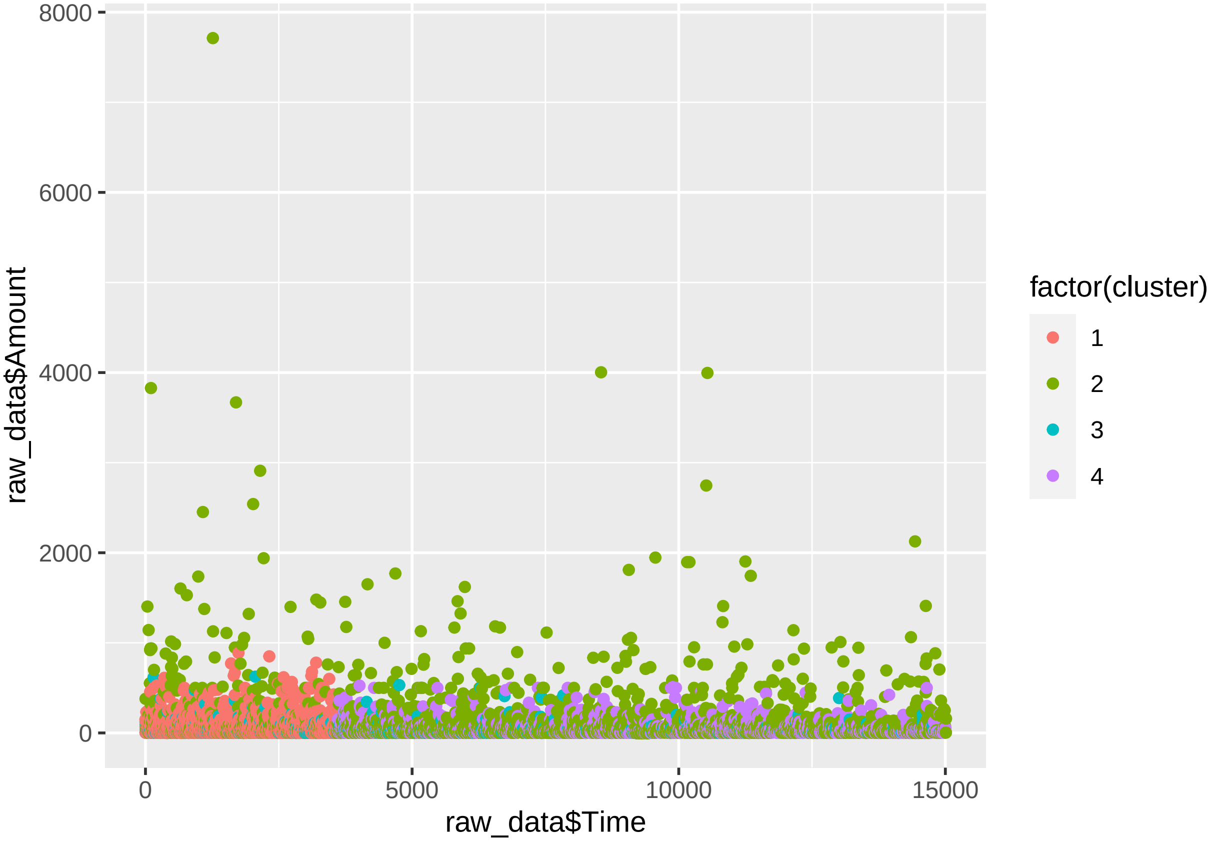
$$d = |x_2 - x_1| + |y_2 - y_1|$$

Using the results from the Euclidean distance proximity matrix, the hierarchical clustering technique was applied and a dendrogram was produced for $k = 4$.

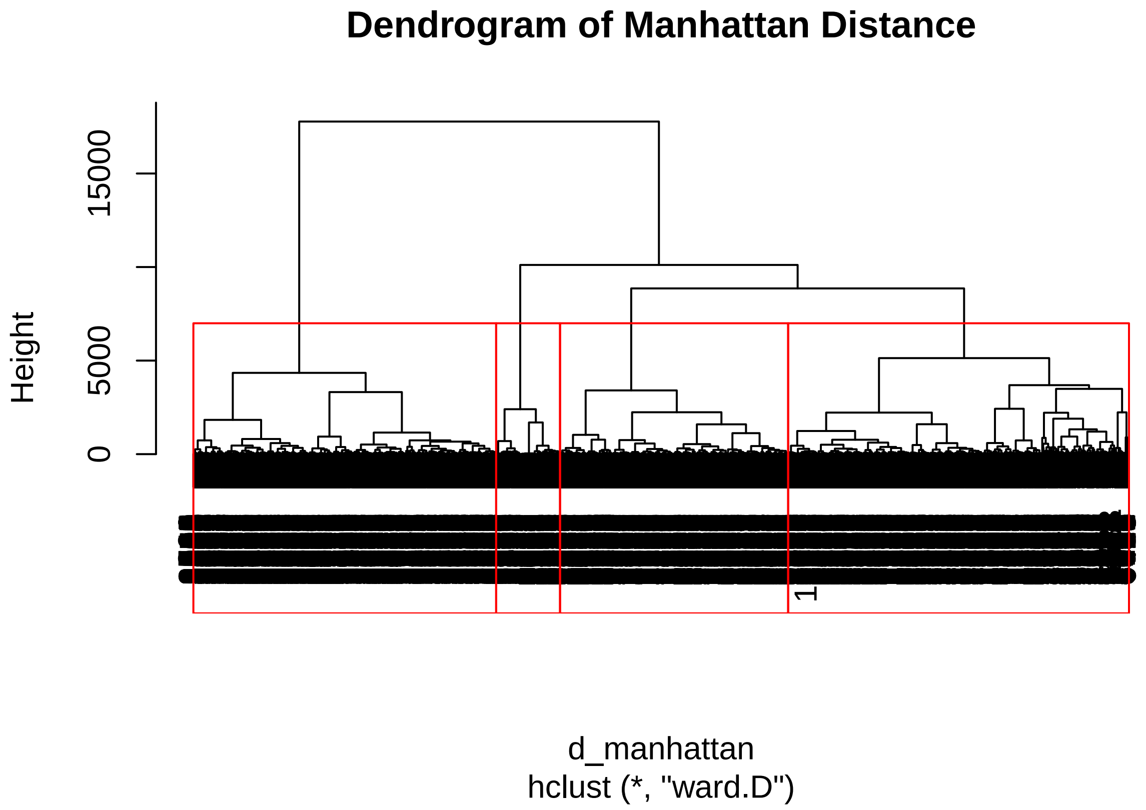

Dendrogram of Euclidean Distance
Height
0
2000
4000
d_euclidean
hclust (*, "ward.D")

Additionally, we can use the clusters to aid in evaluating potential relationships between features in the dataset. This can help us better understand our data. For the current application, we examined the trend between the amount and time variables. The results are shown in the figure below.

```
##   cluster    n
## 1       1 3024
## 2       2 4028
## 3       3  679
## 4       4 2269
```



Using the results from the Manhattan distance proximity matrix, the hierarchical clustering technique was applied and a dendrogram produced for $k = 4$.



Then, we examined the trend between the amount and time variables, using the clusters. The results are shown in the figure below.

```
##   cluster    n
## 1       1 3236
## 2       2 3643
## 3       3  683
## 4       4 2438
```

K-means Clustering

For simplicity, we conducted clustering analysis on the first 10,000 observations

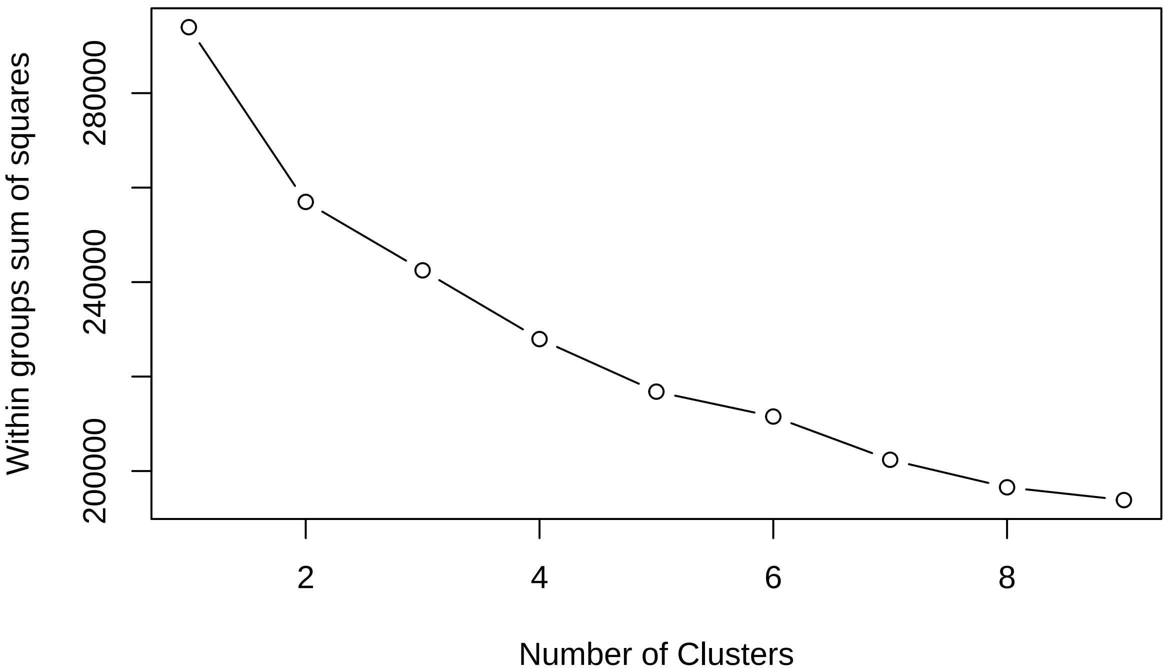
K-means clustering is a technique used to partition data into k clusters, where data points in a given cluster are similar and data points in different are farther apart. The measure of similarity between two points is assessed by determining the distance between them.

```
## $names
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
##
## $class
## [1] "kmeans"
```

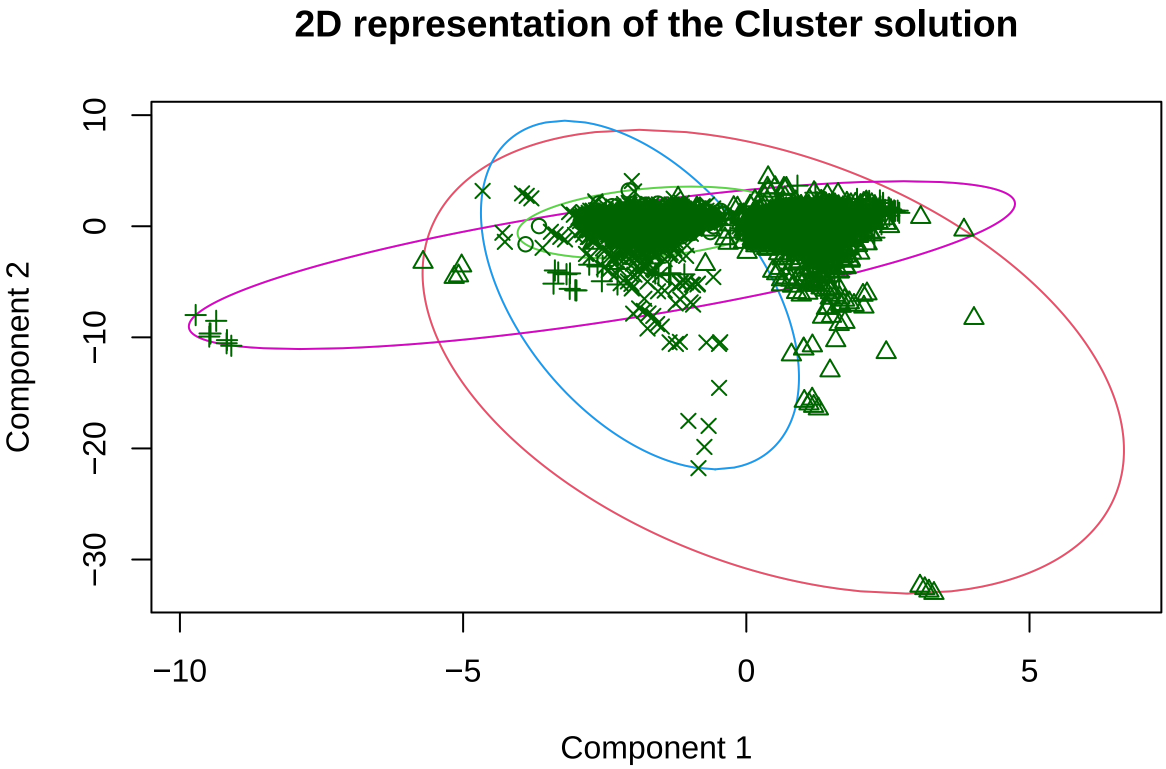
```
##          V1         V2        V3         V4         V5         V6          V7
## 1  1.034250 -0.3092943 0.3133692  0.5216765 -0.4609215 0.06091179 -0.21315827
## 2 -1.349683  0.6516634 1.4605953  0.3148845  0.2139163 0.17204415  0.07770404
## 3  1.069914 -0.1502698 0.4081073  0.5539805 -0.3493624 0.20039701 -0.52862303
## 4 -1.132128  0.6690380 1.1408578 -0.2868908  0.2235338 0.04828848  0.35663623
##            V8         V9         V10        V11        V12         V13
## 1  0.03376904 0.07661325 -0.02243507 0.18494988  0.4037543 -0.06415952
## 2 -0.19087418 1.24413643 -0.24995785 1.05549743 -2.3300516  1.50985083
## 3  0.07909264 1.34885485 -0.42144911 1.31958241 -2.2671005  1.50756520
## 4 -0.13253235 0.03766438 -0.07524233 0.09939666  0.2270910 -0.19997554
##           V14        V15         V16         V17         V18         V19
## 1 -0.06325975  0.3338250 -0.01726977 -0.02931418 -0.17572995 -0.10286014
## 2  1.20153936 -0.4930572 -0.08750450  0.61676515  0.12838633 -0.08393965
## 3  1.25213232 -0.3065976  0.26394352  0.57089004 -0.02985055 -0.19516963
## 4 -0.12099125  0.2594211 -0.21916026 -0.14261901 -0.08708727  0.11066845
##            V20         V21          V22          V23        V24         V25
## 1  0.009631391 -0.06855813 -0.211988712 -0.078619977 0.01974088  0.36347992
## 2  0.014549587 -0.04523451 -0.133309376  0.008109543 0.02485170 -0.09618047
## 3 -0.003119050 -0.14435146 -0.268302935 -0.043610957 0.00743350  0.27246103
## 4  0.089777493  0.05681248 -0.007556546 -0.046380853 0.03406732 -0.06119595
##           V26          V27         V28
## 1  0.06636666  0.006033614  0.03149080
## 2  0.13421430 -0.011620272 -0.01046493
## 3  0.24352672 -0.029554856  0.01775925
## 4 -0.05768755  0.067439500 -0.01435323
```

```
## [1] 1460 3122 2915 2503
```

Determine the number the optimal number of clusters

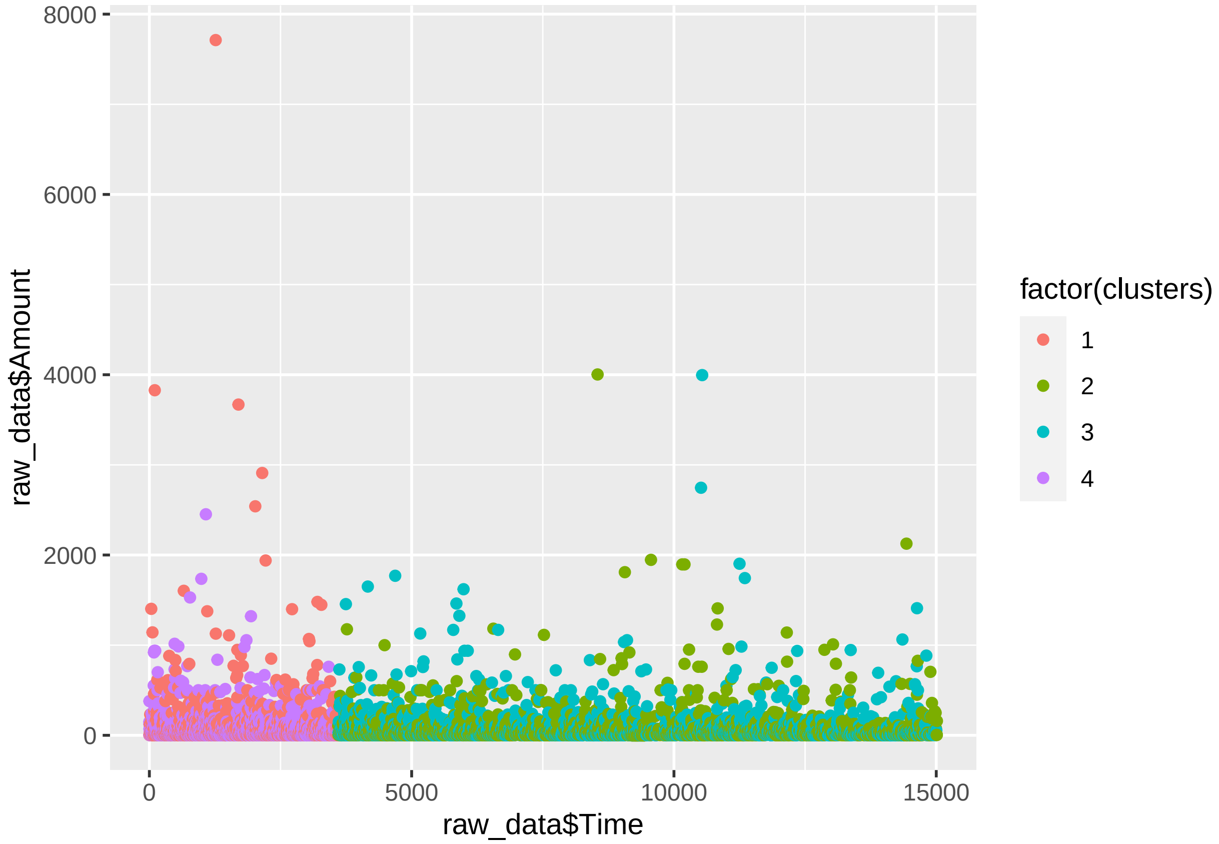


Then, we can visualize the clusters using *clusplot*



These two components explain 15.73 % of the point variability.

Lastly, similar to the approach used in the hierarchical clustering section, we examined the trend between the amount and time variables, using the clusters. The results are shown in the figure below.



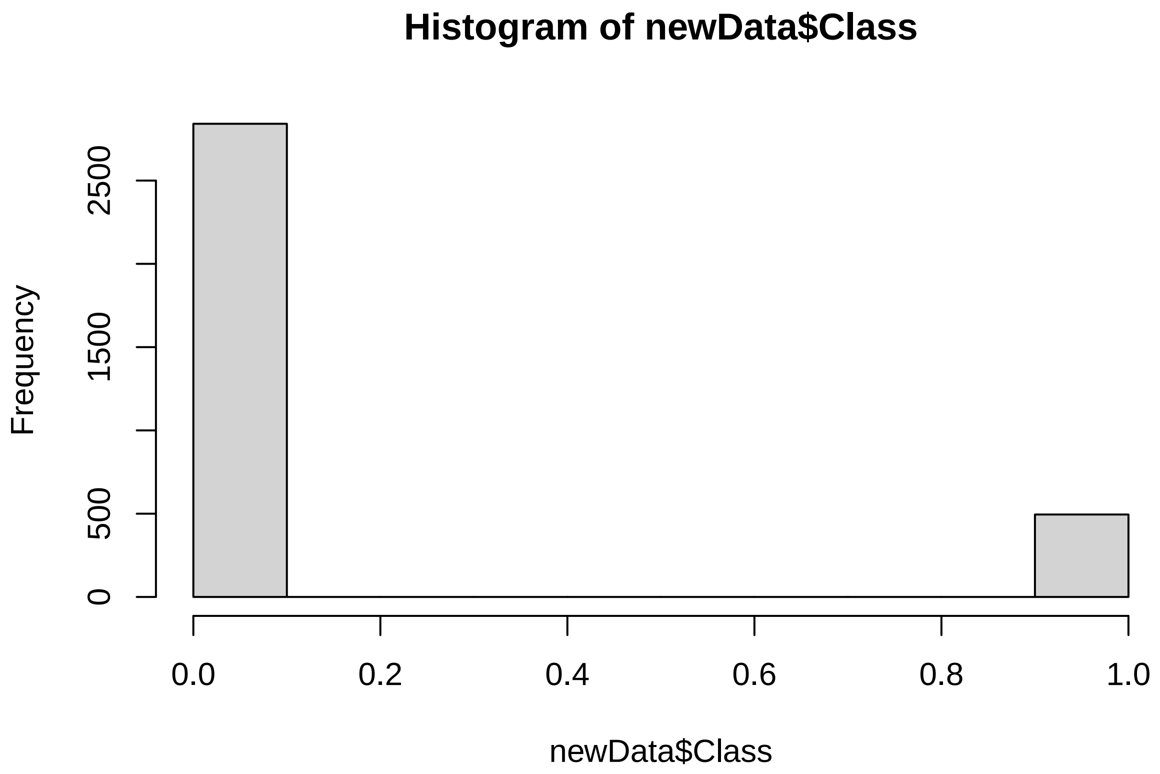
k medoids : PAM

Load the data

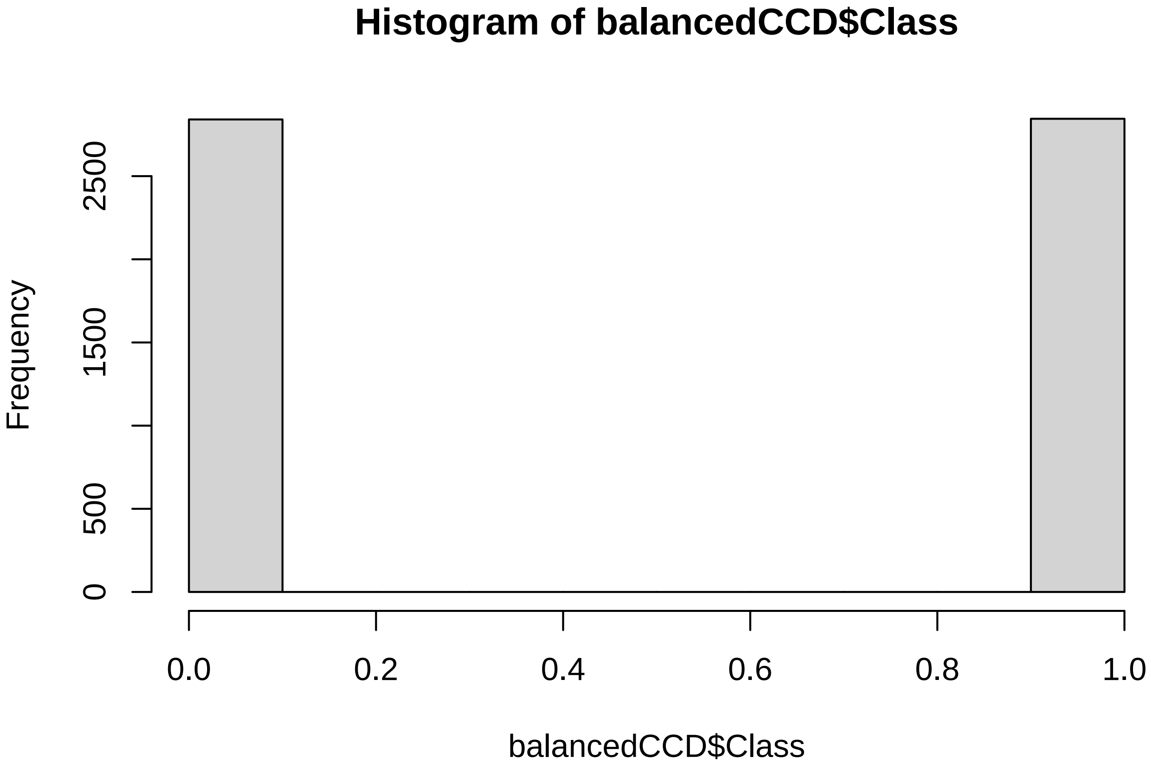
Data exploration No missing data, higly imbalanced data

Since the data is imbalanced, we are going to make balanced dataset. By making a balanced dataset, we can have a better cluster group.

1. under sample the non fraud Data, keeping the existing Fraud Data



2. Over sampling for Fraud data and save the sampled data



3. Make a copy of test data for cluster evaluation

Decision Tree from balanced Dataset

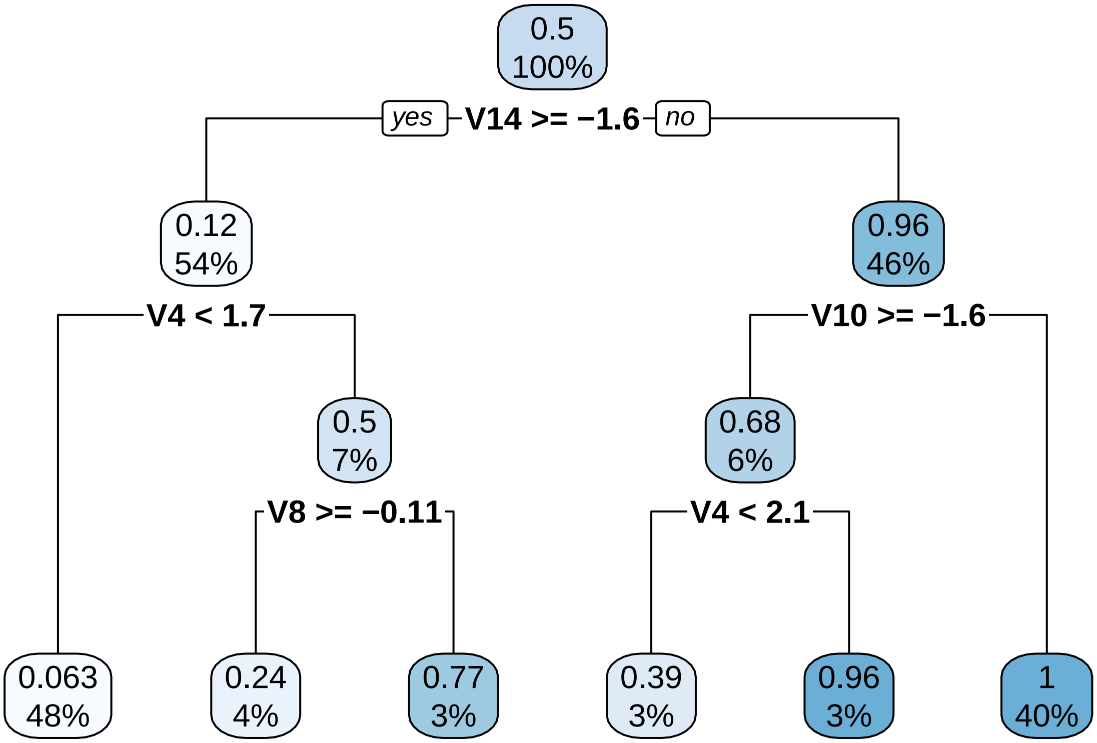
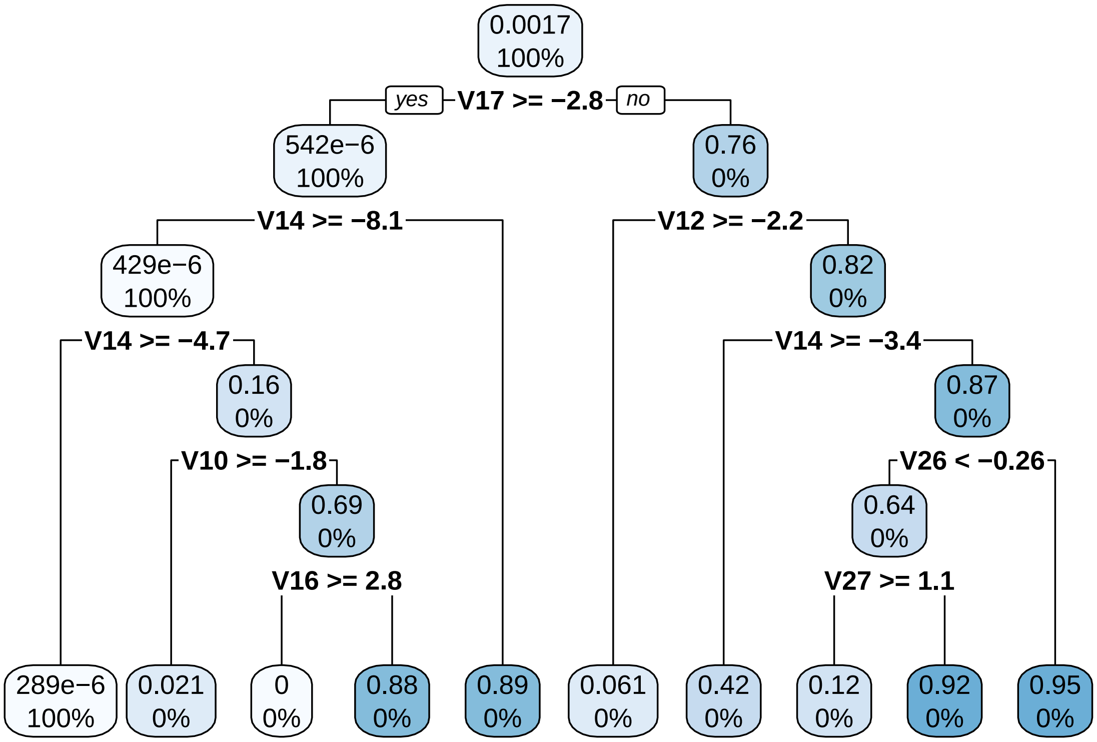
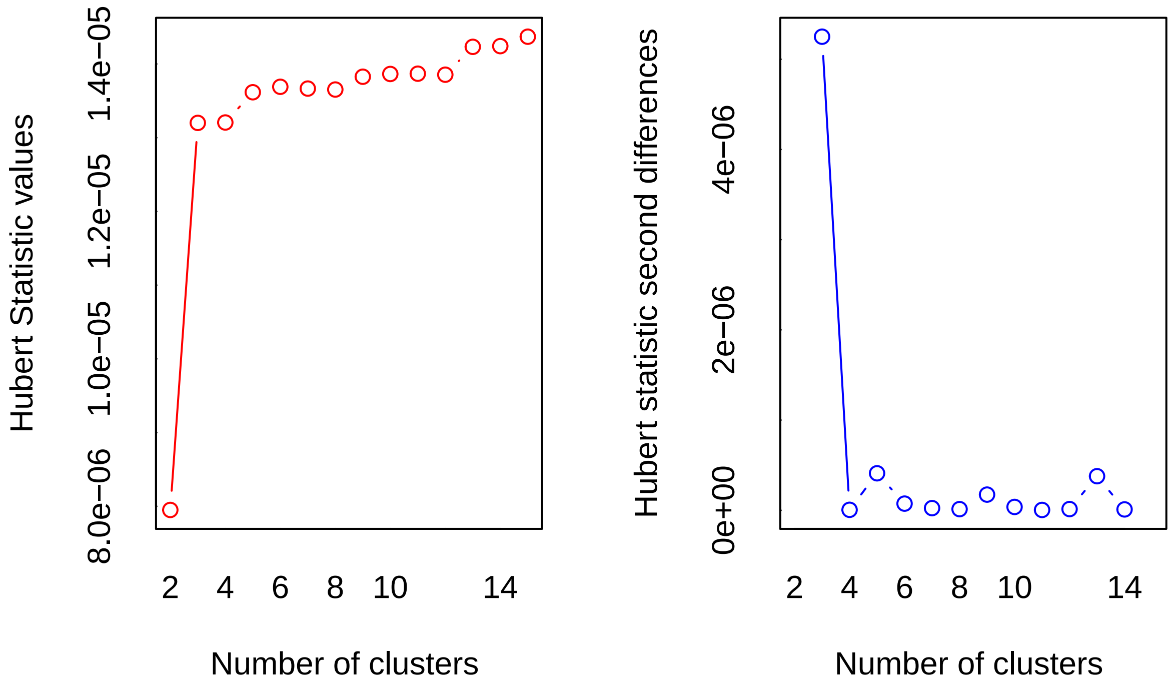


Figure out the important features using decision tree

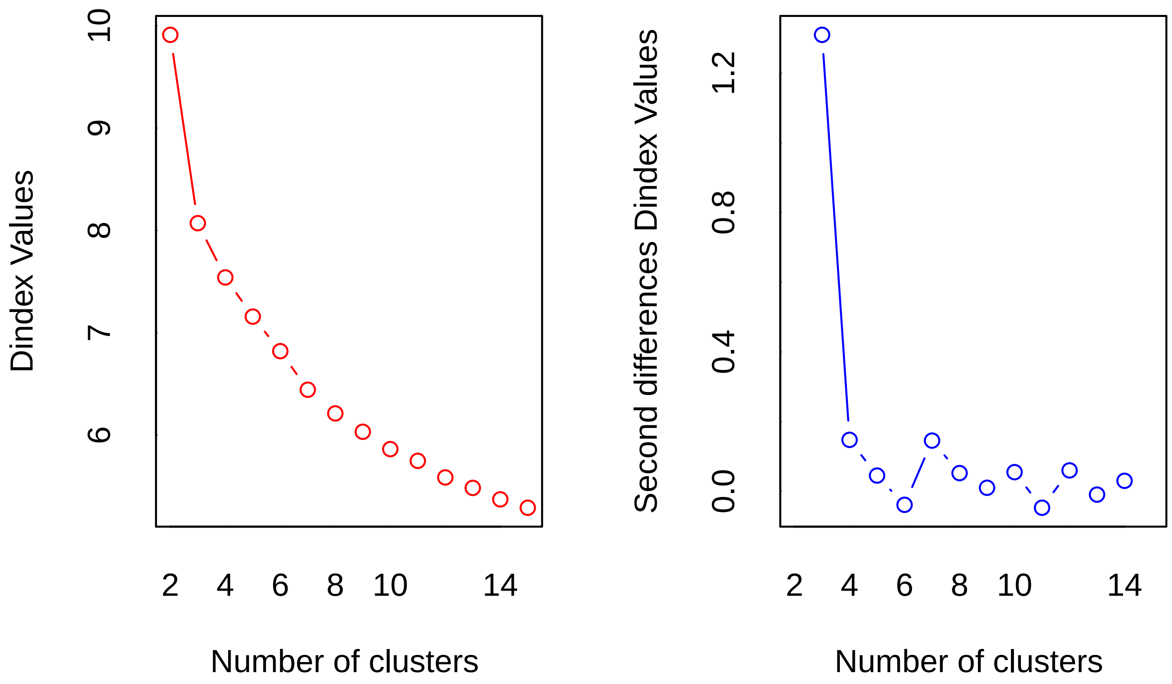


Time, Amount are not important features according to the decision tree, so we will disregard. And the sampled data and its size affects the importance; so for general clustering result, the following will consider the most of the features, except Amount and Time. While several trials were made to check the features' importance, Amount and Time were not included in the importance. And the features existence was not important in the clustering result external evaluation.

Find the best cluster number K The elbow point is around 4

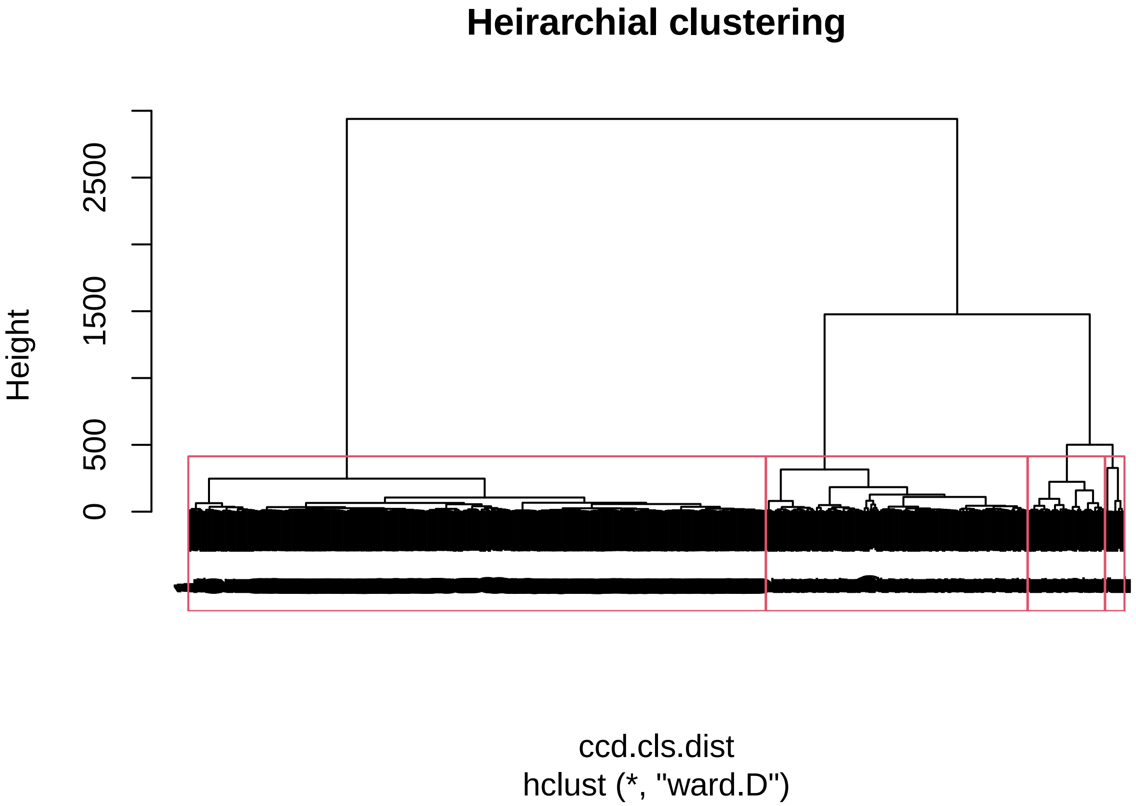


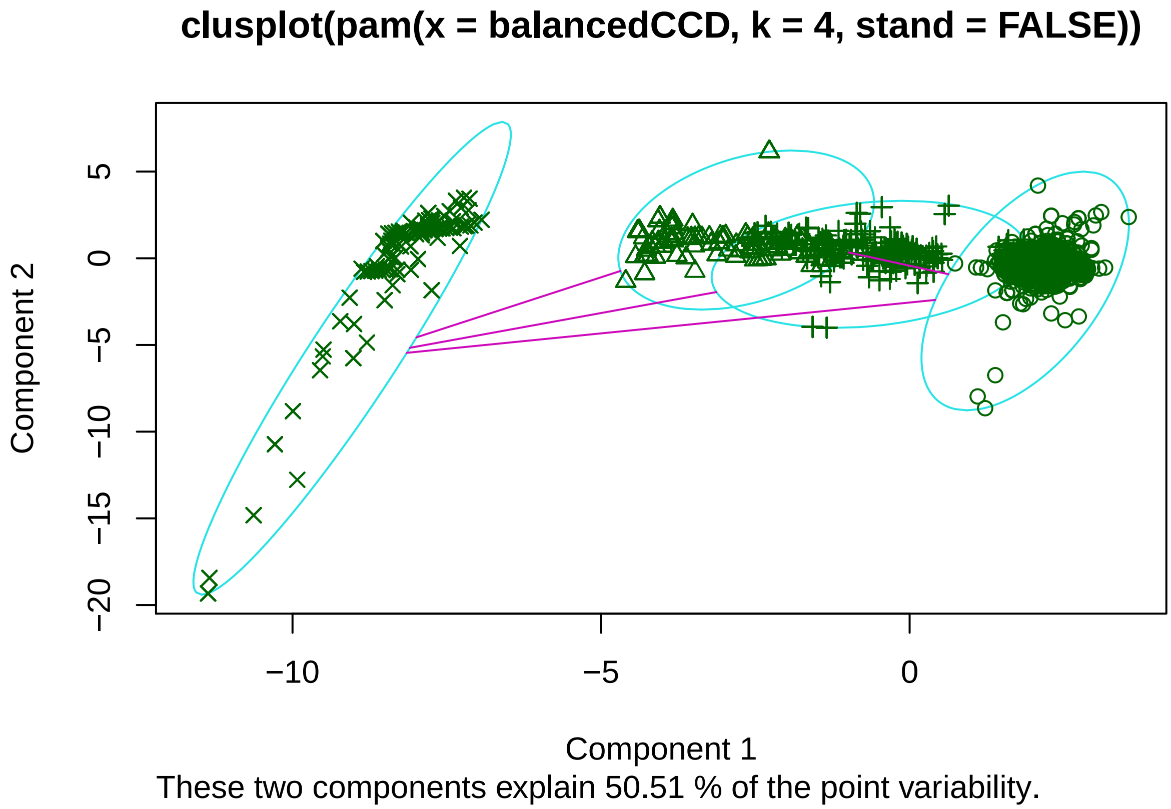
```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 7 proposed 3 as the best number of clusters
## * 3 proposed 4 as the best number of clusters
## * 4 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 2 proposed 7 as the best number of clusters
## * 4 proposed 12 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

Use k=4 since it looks the best clustering number


Heirarchial clustering
Height
0
500
1500
2500
ccd.cls.dist
hclust (*, "ward.D")



Cluster VS Actual

1. External Accuracy Measures

1) Rand Index - “the ratio of matching and unmatched observations among two clustering structures” Higher the value, better the score.

*RAND SCORE = a + d / (a + b + c + d) where • a = observations which are available in the same cluster in both structures (C1 and C2) • b = observations which are available in a cluster in C1 and not in the same cluster in C2 • c = observations which are available in a cluster in C2 and not in the same cluster in C1 • d = observations which are available in different clusters in C1 and C2 1) Rand Index

```
## Loading required package: sp
```

```
## Loading required package: maps
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:cluster':
##
##     votes.repub
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
## Loading required package: shapefiles
```

```
## Loading required package: foreign
```

```
##
## Attaching package: 'shapefiles'
```

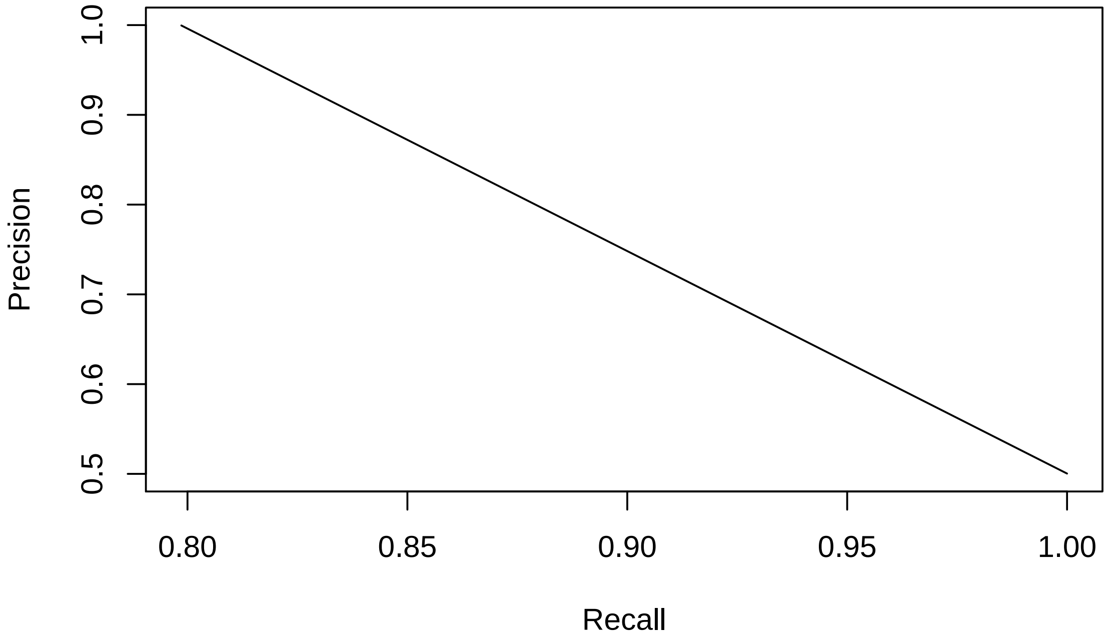
```
## The following objects are masked from 'package:foreign':
##
##     read.dbf, write.dbf
```

```
## [1] "Rand Index: 81.85 %"
```

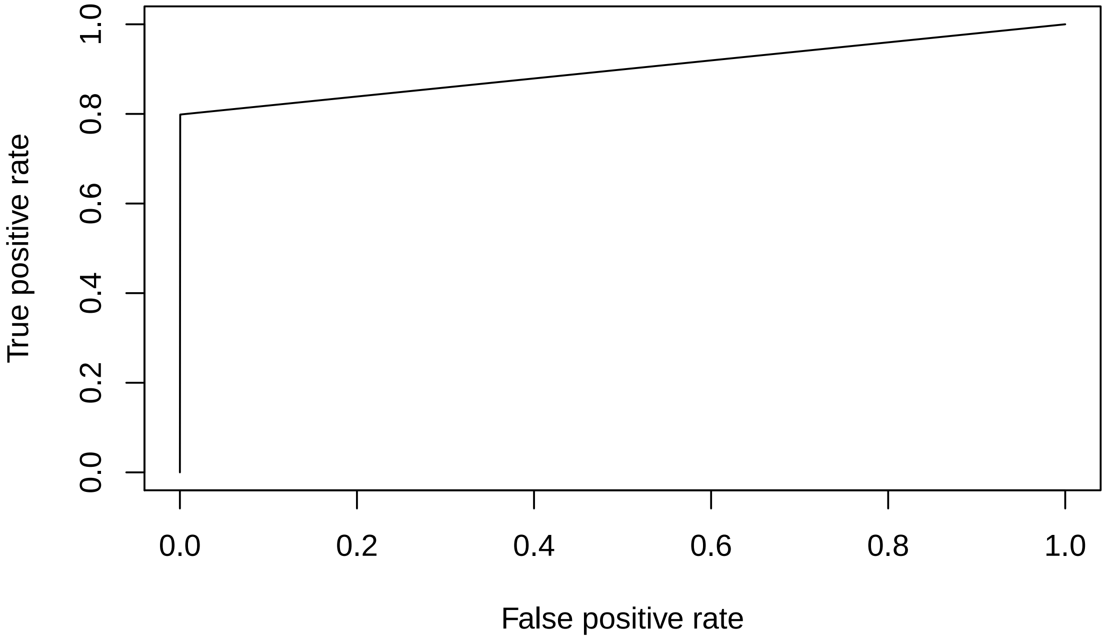
2) Precision Recall Measure

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2840  573
##          1    1 2272
##
##                Accuracy : 0.8991
##                  95% CI : (0.8909, 0.9068)
##     No Information Rate : 0.5004
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7981
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9996
##             Specificity : 0.7986
##          Pos Pred Value : 0.8321
##          Neg Pred Value : 0.9996
##              Prevalence : 0.4996
##          Detection Rate : 0.4995
##    Detection Prevalence : 0.6002
##       Balanced Accuracy : 0.8991
##
##        'Positive' Class : 0
##
```

```
## [1] "Precision: 99.96 %"
## [1] "recall: 79.86 %"
## [1] "F-measure: 88.78 %"
```



3) ROC, AUC curve



```
## [1] "Accuracy: 89.91 %"
```

Using PCA and SVM

Import the orignal data to process PCA and LDA models

This is a 2 days credit card trasaction record, starting from "0" sencond to "171792" sendonds. The feature of "Time" has not any important value to the target feature. Remove "Time"

Convert column name" Amount" to "V29" modeling.

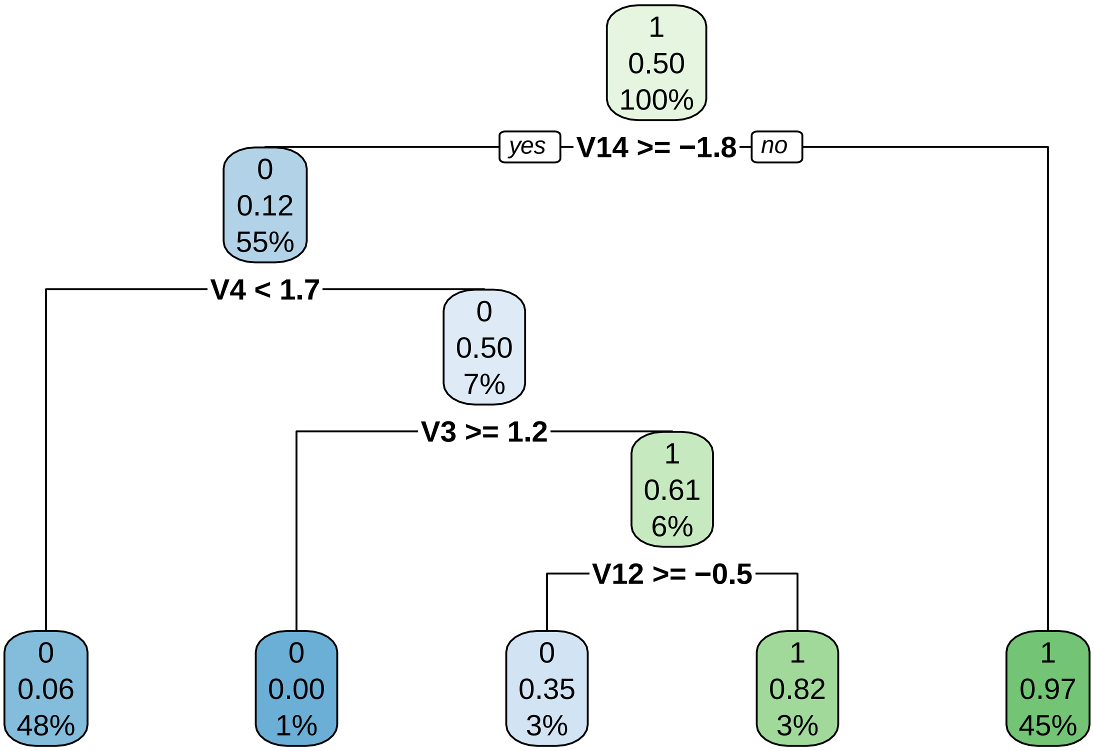
Split the data to train and test datasets.

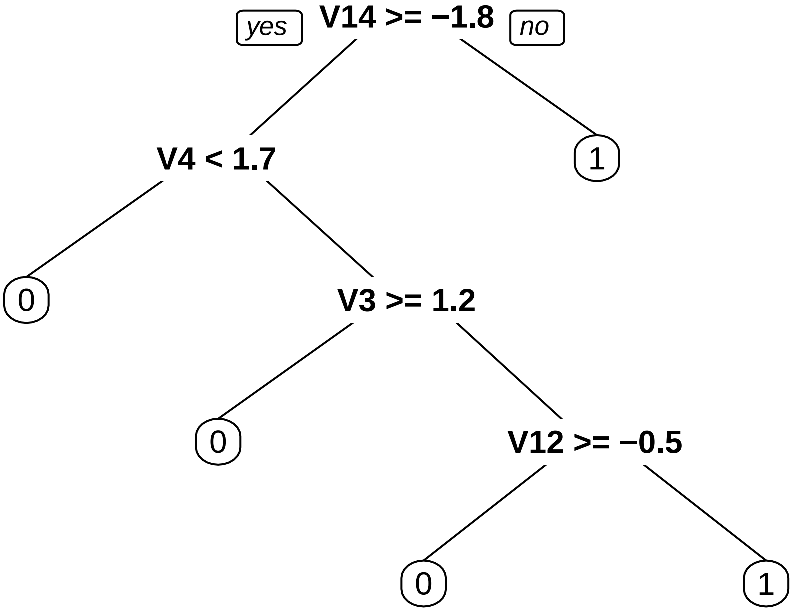
Scale the Amount, V29.

```
##
##           0           1
## 0.998272514 0.001727486
```

The dependant variable is an extremely imbalanced. Treat the imbalanced target variable by the method of "both".

Make decision tree to check feature important for feature selection





V14, V4, V3, V12 contribute 100% variables that affect the target feature, Class. Make a new trainBalanced dataset and test dataset only contains V14,V4,V3,V12 and target variable, V30.

```
## [1] 199364      5
```

```
## [1] 85443     5
```

Applying the decision tree model with the only 5 selected features

```
##
##         0     1
##   0 81654    13
##   1  3641   135
```

Applying Principal Component Analysis-PCA

Principal Component Analysis-PCA can help for data noise filtering, visualizaion, feature extraction.The target feature , Class , has two levels: "0" and "1". So that we set pcaComp into 2 conponents.

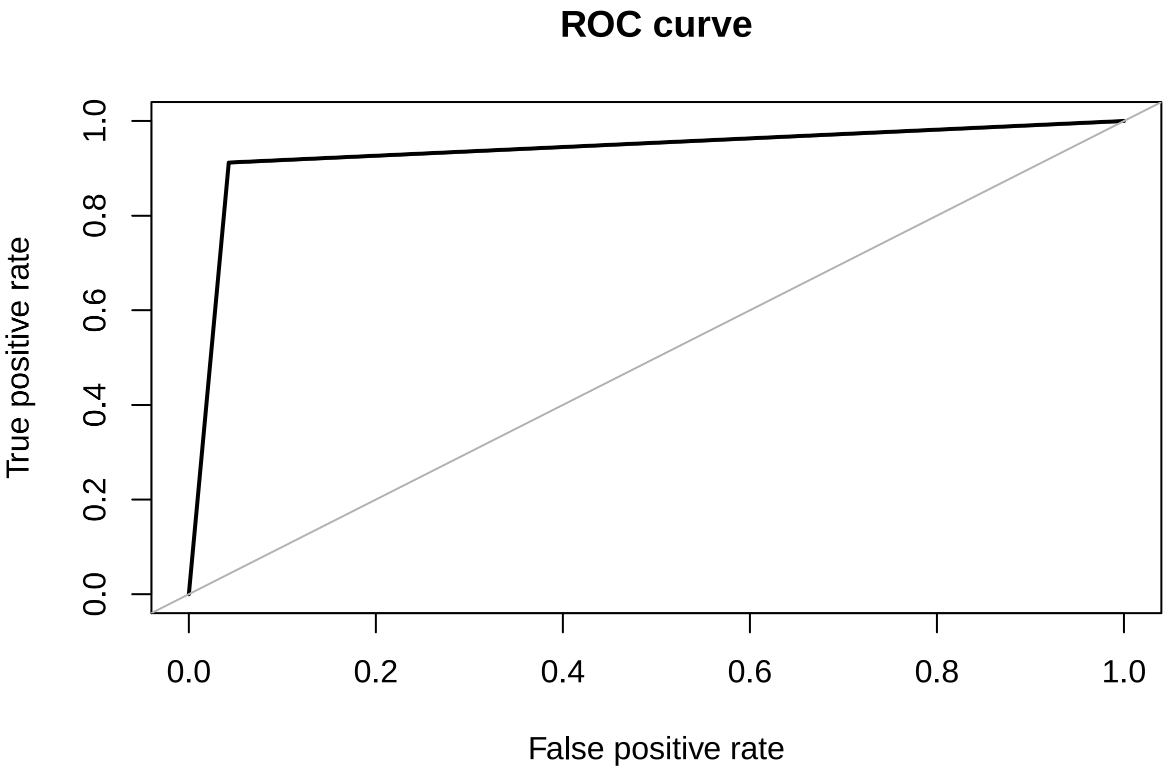
```
##          PC1         PC2 Class
## 1 -1.6214240 -0.29059288     0
## 2 -1.2716250  0.27092609     0
## 3 -0.9728174  0.48671672     0
## 4 -1.3933569 -0.20577187     0
## 5 -1.2848201  0.06506074     0
## 6 -1.6462228  0.05323086     0
```

```
##          PC1        PC2 Class
## 2  -1.391012  0.1202642     0
## 4  -1.600773 -0.1763443     0
## 5  -1.336856 -0.1952519     0
## 8  -1.392333 -0.2109593     0
## 11 -1.541674 -0.1070628     0
## 16 -1.525719 -0.3759904     0
```

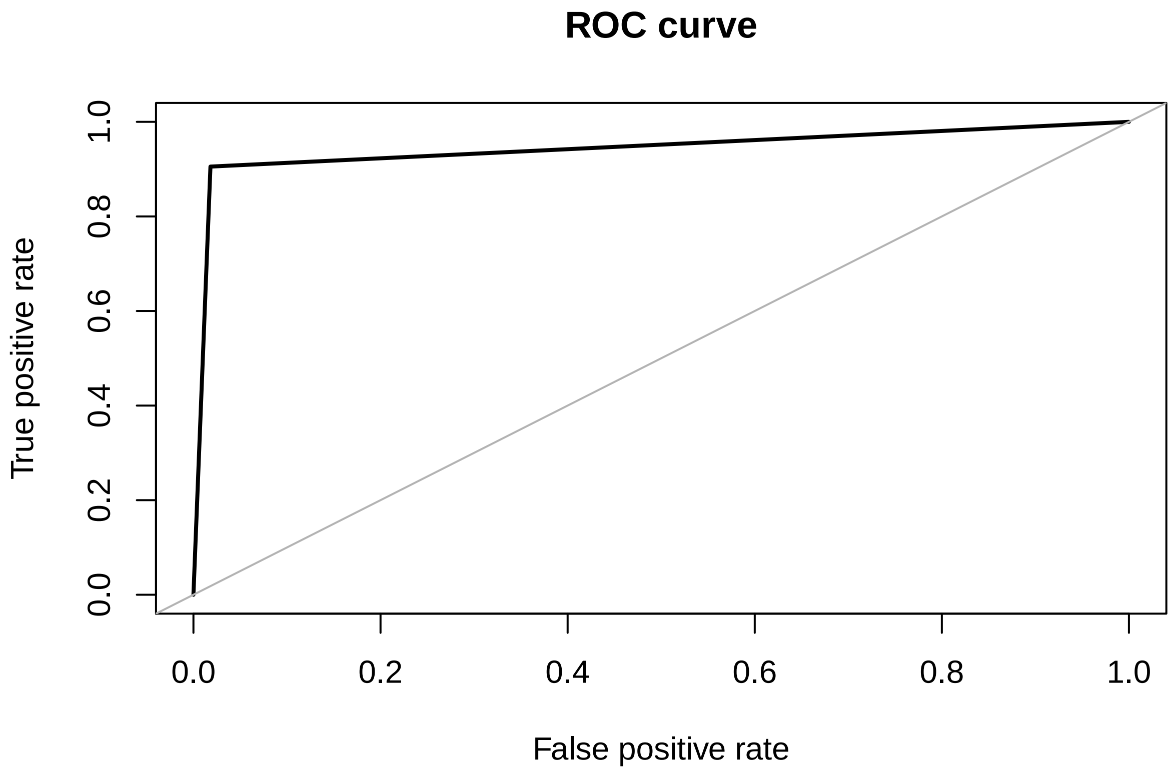
Using PCA to apply decision tree model

```
##
##         0     1
##   0 83742    14
##   1  1553   134
```

Comparing the decision tree model before after PCA



```
## Area under the curve (AUC): 0.935
```



```
## Area under the curve (AUC): 0.944
```

The PCA get better AUC, however the True posicive rate are almost the same.

```
##
## Call:
## accuracy.meas(response = as.factor(new.test$Class), predicted = y_predTree$`1`)
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.036
## recall: 0.912
## F: 0.034
```

```
##
## Call:
## accuracy.meas(response = as.factor(testPca$Class), predicted = y_pca.predTree$`1`)
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.079
## recall: 0.905
## F: 0.073
```

The PCA decision tree has better AUC, however still not very good model.

Applying Linear Discriminant Analysis Lda model

Pca is an unsupervised machine learning technique, however since our data has already shows the target variable so that we can apply Linear Discrininant Analysis_Lda model which is a kind of supervised machine learning technique. Lda can maximize the space for the class-seperation, for a better classification. We will use back the firt training and testing datasets.

```
##    class posterior.0   posterior.1         LD1
## 1      0           1 5.090351e-136  0.02946577
## 3      0           1 1.175690e-135  0.06325069
## 6      0           1 5.278221e-136  0.03092850
## 7      0           1 2.917921e-137 -0.08592534
## 9      0           1 3.037487e-137 -0.08430452
## 10     0           1 4.114215e-136  0.02087322
```

Because we have 2 levels of the target variable, so we get one LD1. We don't need the posterior.0 and posterior.1 for the prediction.

```
##            LD1 class
## 1   0.02946577     0
## 3   0.06325069     0
## 6   0.03092850     0
## 7  -0.08592534     0
## 9  -0.08430452     0
## 10  0.02087322     0
```

So the same to the test dataset to LDA.

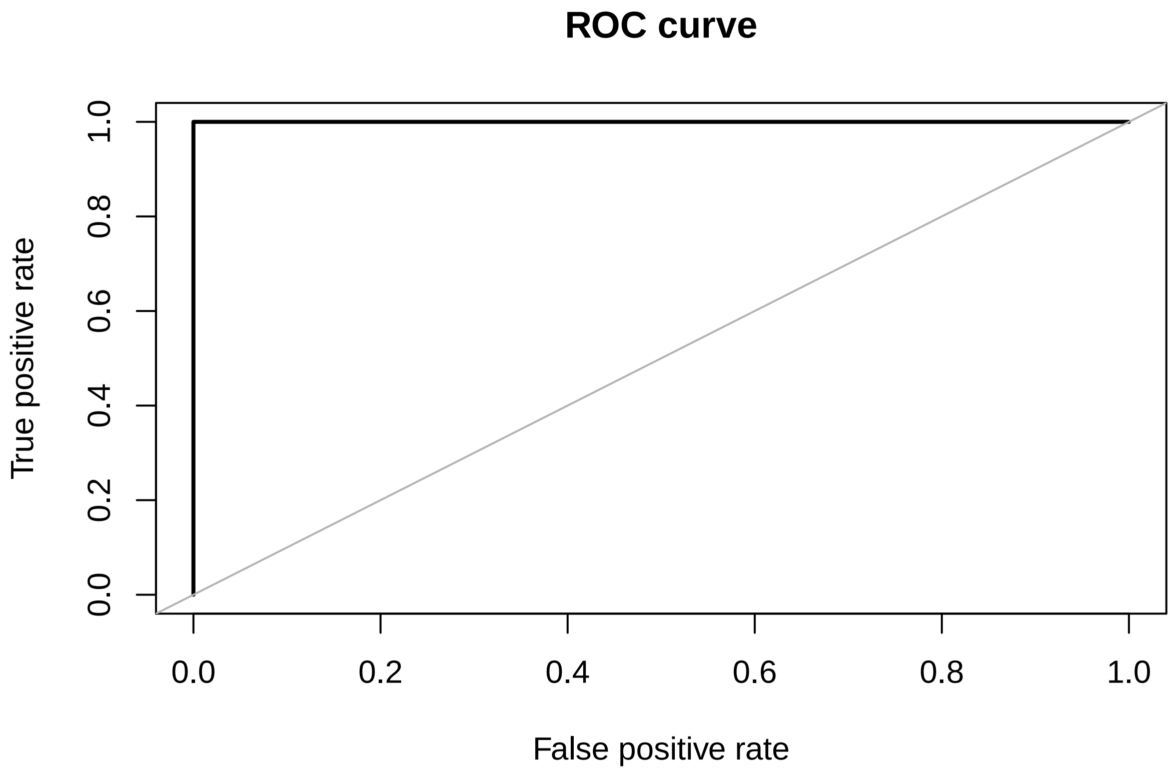
```
##             LD1 class
## 2  -0.004938518     0
## 4  -0.021697179     0
## 5  -0.012931428     0
## 8   0.217533820     0
## 11 -0.073895430     0
## 16 -0.048856346     0
```

Apply LDA to SVM model.

```
##    lda_pred
##         0     1
##   0 85313     1
##   1     0   129
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 85313     0
##          1     1   129
##
##                Accuracy : 1
##                  95% CI : (0.9999, 1)
##     No Information Rate : 0.9985
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9961
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.000000
##             Specificity : 0.999988
##          Pos Pred Value : 0.992308
##          Neg Pred Value : 1.000000
##              Prevalence : 0.001510
##          Detection Rate : 0.001510
##    Detection Prevalence : 0.001521
##       Balanced Accuracy : 0.999994
##
##        'Positive' Class : 1
##
```

We get 100% Accuracy, 100% Sensitivity and 99.99% Specificity which means we get a perfect model to detect the fraud.



```
## Area under the curve (AUC): 1.000
```

We thus recommend the LDA SVM model.

Clustering analysis deployment

We deployed the Kmeans Kmedoids models and shared them at shinyapps.io.

https://ml-lab.shinyapps.io/creditCardDataClustering/

Due to limitations of the free account, we were unable to load the entire data and decided to demonstrate the models using a subset of the data.

Conclusion

We have demonstrated use of Kmeans and Kmedoids , which are clustering techniques suitable for data with dimensions less than 10 in general. We also demonstrated using decision trees and PCA for dimensionality reduction. This technique can be used effectively perform cluster analysis on highly dimensional datasets.

Github link to the project: https://github.com/csml1000groupc/UnsupervisedLearning_CreditCardFraud

Bibliography

https://worldpaymentsreport.com/#

https://www.mckinsey.com/