Credit card Fraud

Sayantika sengupta

2022-12-24

Introduction:

Problem statenent:

> "Credit Card Frauds are the cases of using someone else's credit cards for financial transactions without the information of the card owner. Credit Cards were made available inorder for the people to increase their buying power, it is an agreement with your bank that lets the user use the money lended by the bank in exchange for the repayment of this lended money on the due date or incur interest charges. With the rise in the e-commerce and the recent boom of OTT platforms during the Coronavirus Pandemic, use of credit cards has risen exponentially along with other payment processes. As all the things in the nature are binary, cases of credit card frauds has also achieved high numbers. Global economy pays the price of more than $ 24 billion per year due to these frauds. Thus, it becomes essential to solve this problem and as a result a lot of startups have been born into this $ 30 billion industry. Thus, building automated models for such a rising problem statement is necessary and AI - ML is the key for it!"

Our aim is to :

- Classify and see if the credit card transaction is fraudulent or genuine.
- This is an *unsupervised learning* problem which does the Binary Classification.

Introduction to the data:

Dataset Attributes:

- V1 - V28 : Numerical features that are a result of PCA transformation.
- Time: Seconds elapsed between each transaction and the 1st transaction.
- Amount: Transaction amount.
- Class: Fraud or otherwise (1 or 0)

Collection of the dataset:

```
system.time(df<-read.csv("/home/sayantika/Desktop/Project/creditcard.csv"))
```

```
##    user  system elapsed
##  35.919   0.267  36.194
```

```
head(df)
```

```
##   Time         V1          V2        V3         V4          V5          V6
## 1    0 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.46238778
## 2    0  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.08236081
## 3    1 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.80049938
## 4    1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.24720317
## 5    2 -1.1582331  0.87773675 1.5487178  0.4030339 -0.40719338  0.09592146
```

```
## 6    2 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.02972755
##            V7          V8         V9         V10        V11         V12
## 1  0.23959855  0.09869790  0.3637870  0.09079417 -0.5515995 -0.61780086
## 2 -0.07880298  0.08510165 -0.2554251 -0.16697441  1.6127267  1.06523531
## 3  0.79146096  0.24767579 -1.5146543  0.20764287  0.6245015  0.06608369
## 4  0.23760894  0.37743587 -1.3870241 -0.05495192 -0.2264873  0.17822823
## 5  0.59294075 -0.27053268  0.8177393  0.75307443 -0.8228429  0.53819555
## 6  0.47620095  0.26031433 -0.5686714 -0.37140720  1.3412620  0.35989384
##          V13        V14        V15        V16         V17         V18
## 1 -0.9913898 -0.3111694  1.4681770 -0.4704005  0.20797124  0.02579058
## 2  0.4890950 -0.1437723  0.6355581  0.4639170 -0.11480466 -0.18336127
## 3  0.7172927 -0.1659459  2.3458649 -2.8900832  1.10996938 -0.12135931
## 4  0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.68409279  1.96577500
## 5  1.3458516 -1.1196698  0.1751211 -0.4514492 -0.23703324 -0.03819479
## 6 -0.3580907 -0.1371337  0.5176168  0.4017259 -0.05813282  0.06865315
##           V19         V20          V21          V22         V23         V24
## 1  0.40399296  0.25141210 -0.018306778  0.277837576 -0.11047391  0.06692807
## 2 -0.14578304 -0.06908314 -0.225775248 -0.638671953  0.10128802 -0.33984648
## 3 -2.26185710  0.52497973  0.247998153  0.771679402  0.90941226 -0.68928096
## 4 -1.23262197 -0.20803778 -0.108300452  0.005273597 -0.19032052 -1.17557533
## 5  0.80348692  0.40854236 -0.009430697  0.798278495 -0.13745808  0.14126698
## 6 -0.03319379  0.08496767 -0.208253515 -0.559824796 -0.02639767 -0.37142658
##          V25        V26          V27         V28 Amount Class
## 1  0.1285394 -0.1891148  0.133558377 -0.02105305 149.62     0
## 2  0.1671704  0.1258945 -0.008983099  0.01472417   2.69     0
## 3 -0.3276418 -0.1390966 -0.055352794 -0.05975184 378.66     0
## 4  0.6473760 -0.2219288  0.062722849  0.06145763 123.50     0
## 5 -0.2060096  0.5022922  0.219422230  0.21515315  69.99     0
## 6 -0.2327938  0.1059148  0.253844225  0.08108026   3.67     0
```

```
nrow(df)
```

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

Data Information:

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

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

```
##          Time            V1            V2            V3            V4
##  9.481386e+04  1.176324e-15  3.383673e-16 -1.396978e-15  2.094355e-15
##            V5            V6            V7            V8            V9
##  1.005890e-15  1.495474e-15 -5.638175e-16  1.145356e-16 -2.412173e-15
##           V10           V11           V12           V13           V14
##  2.235882e-15  1.698766e-15 -1.245865e-15  8.251477e-16  1.211625e-15
##           V15           V16           V17           V18           V19
##  4.888653e-15  1.435697e-15 -3.757572e-16  9.697262e-16  1.037610e-15
##           V20           V21           V22           V23           V24
##  6.409288e-16  1.613595e-16 -3.509494e-16  2.632029e-16  4.472927e-15
##           V25           V26           V27           V28        Amount
##  5.144033e-16  1.685498e-15 -3.658860e-16 -1.219590e-16  8.834962e+01
##         Class
##  1.727486e-03
```

Fraud data:

```
fraud = df[df$Class ==1,]
summary(fraud)
```

```
##       Time              V1                  V2                V3          
##  Min.   :   406   Min.   :-30.5524   Min.   :-8.402   Min.   :-31.104  
##  1st Qu.: 41242   1st Qu.: -6.0361   1st Qu.: 1.188   1st Qu.: -8.643  
##  Median : 75568   Median : -2.3425   Median : 2.718   Median : -5.075  
##  Mean   : 80747   Mean   : -4.7719   Mean   : 3.624   Mean   : -7.033  
##  3rd Qu.:128483   3rd Qu.: -0.4192   3rd Qu.: 4.971   3rd Qu.: -2.276  
##  Max.   :170348   Max.   :  2.1324   Max.   :22.058   Max.   :  2.250  
##        V4                V5                  V6                 V7           
##  Min.   :-1.313   Min.   :-22.1055   Min.   :-6.4063   Min.   :-43.5572  
##  1st Qu.: 2.373   1st Qu.: -4.7928   1st Qu.:-2.5015   1st Qu.: -7.9653  
##  Median : 4.177   Median : -1.5230   Median :-1.4246   Median : -3.0344  
##  Mean   : 4.542   Mean   : -3.1512   Mean   :-1.3977   Mean   : -5.5687  
##  3rd Qu.: 6.349   3rd Qu.:  0.2146   3rd Qu.:-0.4132   3rd Qu.: -0.9459  
##  Max.   :12.115   Max.   : 11.0951   Max.   : 6.4741   Max.   :  5.8025  
##        V8                 V9                V10                V11          
##  Min.   :-41.0443   Min.   :-13.4341   Min.   :-24.588   Min.   :-1.702  
##  1st Qu.: -0.1953   1st Qu.: -3.8724   1st Qu.: -7.757   1st Qu.: 1.973  
##  Median :  0.6215   Median : -2.2088   Median : -4.579   Median : 3.586  
##  Mean   :  0.5706   Mean   : -2.5811   Mean   : -5.677   Mean   : 3.800  
##  3rd Qu.:  1.7649   3rd Qu.: -0.7879   3rd Qu.: -2.614   3rd Qu.: 5.307  
##  Max.   : 20.0072   Max.   :  3.3535   Max.   :  4.031   Max.   :12.019  
##       V12                V13                V14                V15          
##  Min.   :-18.684   Min.   :-3.12779   Min.   :-19.214   Min.   :-4.49894  
##  1st Qu.: -8.688   1st Qu.:-0.97912   1st Qu.: -9.693   1st Qu.:-0.64354  
##  Median : -5.503   Median :-0.06557   Median : -6.730   Median :-0.05723  
##  Mean   : -6.259   Mean   :-0.10933   Mean   : -6.972   Mean   :-0.09293  
##  3rd Qu.: -2.974   3rd Qu.: 0.67296   3rd Qu.: -4.283   3rd Qu.: 0.60919  
##  Max.   :  1.376   Max.   : 2.81544   Max.   :  3.442   Max.   : 2.47136  
##       V16                V17                V18                V19          
##  Min.   :-14.130   Min.   :-25.163   Min.   :-9.49875   Min.   :-3.6819  
##  1st Qu.: -6.563   1st Qu.:-11.945   1st Qu.:-4.66458   1st Qu.:-0.2994  
##  Median : -3.550   Median : -5.303   Median :-1.66435   Median : 0.6468  
##  Mean   : -4.140   Mean   : -6.666   Mean   :-2.24631   Mean   : 0.6807  
##  3rd Qu.: -1.226   3rd Qu.: -1.342   3rd Qu.: 0.09177   3rd Qu.: 1.6493  
##  Max.   :  3.140   Max.   :  6.739   Max.   : 3.79032   Max.   : 5.2283  
##       V20                V21                V22                V23          
##  Min.   :-4.1282   Min.   :-22.79760   Min.   :-8.88702   Min.   :-19.25433  
##  1st Qu.:-0.1718   1st Qu.:  0.04179   1st Qu.:-0.53376   1st Qu.: -0.34218  
##  Median : 0.2847   Median :  0.59215   Median : 0.04843   Median : -0.07314  
##  Mean   : 0.3723   Mean   :  0.71359   Mean   : 0.01405   Mean   : -0.04031  
##  3rd Qu.: 0.8224   3rd Qu.:  1.24461   3rd Qu.: 0.61747   3rd Qu.:  0.30838  
##  Max.   :11.0590   Max.   : 27.20284   Max.   : 8.36199   Max.   :  5.46623  
##       V24                V25                V26                V27          
##  Min.   :-2.0280   Min.   :-4.78161   Min.   :-1.152671   Min.   :-7.26348  
##  1st Qu.:-0.4368   1st Qu.:-0.31435   1st Qu.:-0.259416   1st Qu.:-0.02003  
##  Median :-0.0608   Median : 0.08837   Median : 0.004321   Median : 0.39493  
##  Mean   :-0.1051   Mean   : 0.04145   Mean   : 0.051648   Mean   : 0.17058  
##  3rd Qu.: 0.2853   3rd Qu.: 0.45652   3rd Qu.: 0.396733   3rd Qu.: 0.82603  
##  Max.   : 1.0914   Max.   : 2.20821   Max.   : 2.745261   Max.   : 3.05236
```

```
##       V28                Amount             Class
##  Min.   :-1.86929   Min.   :   0.00   Min.   :1
##  1st Qu.:-0.10887   1st Qu.:   1.00   1st Qu.:1
##  Median : 0.14634   Median :   9.25   Median :1
##  Mean   : 0.07567   Mean   : 122.21   Mean   :1
##  3rd Qu.: 0.38115   3rd Qu.: 105.89   3rd Qu.:1
##  Max.   : 1.77936   Max.   :2125.87   Max.   :1
```

```
colMeans(fraud)
```

```
##          Time            V1            V2            V3            V4
##  8.074681e+04 -4.771948e+00  3.623778e+00 -7.033281e+00  4.542029e+00
##            V5            V6            V7            V8            V9
## -3.151225e+00 -1.397737e+00 -5.568731e+00  5.706359e-01 -2.581123e+00
##           V10           V11           V12           V13           V14
## -5.676883e+00  3.800173e+00 -6.259393e+00 -1.093338e-01 -6.971723e+00
##           V15           V16           V17           V18           V19
## -9.292875e-02 -4.139946e+00 -6.665836e+00 -2.246308e+00  6.806593e-01
##           V20           V21           V22           V23           V24
##  3.723194e-01  7.135884e-01  1.404888e-02 -4.030797e-02 -1.051303e-01
##           V25           V26           V27           V28        Amount
##  4.144889e-02  5.164813e-02  1.705748e-01  7.566729e-02  1.222113e+02
##         Class
##  1.000000e+00
```

```
nrow(fraud)
```

```
## [1] 492
```

Genuine data:

```
nofraud = df[df$Class == 0, ]
summary(nofraud)
```

```
##       Time              V1                    V2                   V3
##  Min.   :     0   Min.   :-56.40751   Min.   :-72.71573   Min.   :-48.32559
##  1st Qu.: 54230   1st Qu.: -0.91754   1st Qu.: -0.59947   1st Qu.: -0.88454
##  Median : 84711   Median :  0.02002   Median :  0.06407   Median :  0.18216
##  Mean   : 94838   Mean   :  0.00826   Mean   : -0.00627   Mean   :  0.01217
##  3rd Qu.:139333   3rd Qu.:  1.31622   3rd Qu.:  0.80045   3rd Qu.:  1.02837
##  Max.   :172792   Max.   :  2.45493   Max.   : 18.90245   Max.   :  9.38256
##        V4                 V5                    V6
##  Min.   :-5.68317   Min.   :-113.74331   Min.   :-26.16051
##  1st Qu.:-0.85008   1st Qu.:  -0.68940   1st Qu.: -0.76685
##  Median :-0.02241   Median :  -0.05346   Median : -0.27312
##  Mean   :-0.00786   Mean   :   0.00545   Mean   :  0.00242
##  3rd Qu.: 0.73762   3rd Qu.:   0.61218   3rd Qu.:  0.39962
##  Max.   :16.87534   Max.   :  34.80167   Max.   : 73.30163
##        V7                 V8                    V9
##  Min.   :-31.76495   Min.   :-73.21672   Min.   :-6.290730
##  1st Qu.: -0.55144   1st Qu.: -0.20863   1st Qu.:-0.640412
##  Median :  0.04114   Median :  0.02204   Median :-0.049964
##  Mean   :  0.00964   Mean   : -0.00099   Mean   : 0.004467
##  3rd Qu.:  0.57102   3rd Qu.:  0.32620   3rd Qu.: 0.598230
##  Max.   :120.58949   Max.   : 18.70925   Max.   :15.594995
##       V10                V11                   V12
```

```
##  Min.   :-14.741096   Min.   :-4.797473   Min.   :-15.14499
##  1st Qu.: -0.532880   1st Qu.:-0.763447   1st Qu.: -0.40210
##  Median : -0.091872   Median :-0.034923   Median :  0.14168
##  Mean   :  0.009824   Mean   :-0.006576   Mean   :  0.01083
##  3rd Qu.:  0.455135   3rd Qu.: 0.736362   3rd Qu.:  0.61921
##  Max.   : 23.745136   Max.   :10.002190   Max.   :  7.84839
##       V13                 V14                 V15
##  Min.   :-5.791881   Min.   :-18.39209   Min.   :-4.391307
##  1st Qu.:-0.648067   1st Qu.: -0.42245   1st Qu.:-0.582812
##  Median :-0.013547   Median :  0.05195   Median : 0.048294
##  Mean   : 0.000189   Mean   :  0.01206   Mean   : 0.000161
##  3rd Qu.: 0.662492   3rd Qu.:  0.49410   3rd Qu.: 0.648842
##  Max.   : 7.126883   Max.   : 10.52677   Max.   : 8.877742
##       V16                 V17                 V18
##  Min.   :-10.115560   Min.   :-17.09844   Min.   :-5.366660
##  1st Qu.: -0.465543   1st Qu.: -0.48264   1st Qu.:-0.497414
##  Median :  0.067377   Median : -0.06483   Median :-0.002787
##  Mean   :  0.007164   Mean   :  0.01154   Mean   : 0.003887
##  3rd Qu.:  0.523738   3rd Qu.:  0.39992   3rd Qu.: 0.501103
##  Max.   : 17.315112   Max.   :  9.25353   Max.   : 5.041069
##       V19                 V20                 V21
##  Min.   :-7.213527   Min.   :-54.49772   Min.   :-34.83038
##  1st Qu.:-0.456366   1st Qu.: -0.21176   1st Qu.: -0.22851
##  Median : 0.003117   Median : -0.06265   Median : -0.02982
##  Mean   :-0.001178   Mean   : -0.00064   Mean   : -0.00123
##  3rd Qu.: 0.457499   3rd Qu.:  0.13240   3rd Qu.:  0.18563
##  Max.   : 5.591971   Max.   : 39.42090   Max.   : 22.61489
##       V22                 V23                 V24
##  Min.   :-10.933144   Min.   :-44.80774   Min.   :-2.836627
##  1st Qu.: -0.542403   1st Qu.: -0.16170   1st Qu.:-0.354425
##  Median :  0.006736   Median : -0.01115   Median : 0.041082
##  Mean   : -0.000024   Mean   :  0.00007   Mean   : 0.000182
##  3rd Qu.:  0.528407   3rd Qu.:  0.14752   3rd Qu.: 0.439869
##  Max.   : 10.503090   Max.   : 22.52841   Max.   : 4.584549
##       V25                 V26                 V27
##  Min.   :-10.295397   Min.   :-2.604551   Min.   :-22.565679
##  1st Qu.: -0.317145   1st Qu.:-0.327074   1st Qu.: -0.070852
##  Median :  0.016417   Median :-0.052227   Median :  0.001230
##  Mean   : -0.000072   Mean   :-0.000089   Mean   : -0.000295
##  3rd Qu.:  0.350594   3rd Qu.: 0.240671   3rd Qu.:  0.090573
##  Max.   :  7.519589   Max.   : 3.517346   Max.   : 31.612198
##       V28                Amount              Class
##  Min.   :-15.43008   Min.   :    0.00   Min.   :0
##  1st Qu.: -0.05295   1st Qu.:    5.65   1st Qu.:0
##  Median :  0.01120   Median :   22.00   Median :0
##  Mean   : -0.00013   Mean   :   88.29   Mean   :0
##  3rd Qu.:  0.07796   3rd Qu.:   77.05   3rd Qu.:0
##  Max.   : 33.84781   Max.   :25691.16   Max.   :0
```

```
colMeans(nofraud)
```

```
##          Time            V1            V2            V3            V4
##  9.483820e+04  8.257737e-03 -6.270857e-03  1.217092e-02 -7.859868e-03
##            V5            V6            V7            V8            V9
##  5.453116e-03  2.418748e-03  9.636550e-03 -9.874712e-04  4.466569e-03
```

```
##           V10           V11           V12           V13           V14
##  9.823704e-03 -6.576104e-03  1.083172e-02  1.891994e-04  1.206439e-02
##           V15           V16           V17           V18           V19
##  1.608109e-04  7.164073e-03  1.153506e-02  3.887180e-03 -1.177864e-03
##           V20           V21           V22           V23           V24
## -6.442894e-04 -1.234847e-03 -2.431124e-05  6.975193e-05  1.819254e-04
##           V25           V26           V27           V28        Amount
## -7.172626e-05 -8.937579e-05 -2.951754e-04 -1.309404e-04  8.829102e+01
##         Class
##  0.000000e+00
```

```
nrow(nofraud)
```

```
## [1] 284315
```

Realisations:

- From the column means of fraud and genuine cases, we have For No Fraud cases, V1 - V28 mean values are almost 0 for all the cases. Mean Amount, 88.29, is less than the mean transaction amount, 122.21, of the Fraud cases.
- Time taken for No Fraud transactions is more than those for Fraud transactions.

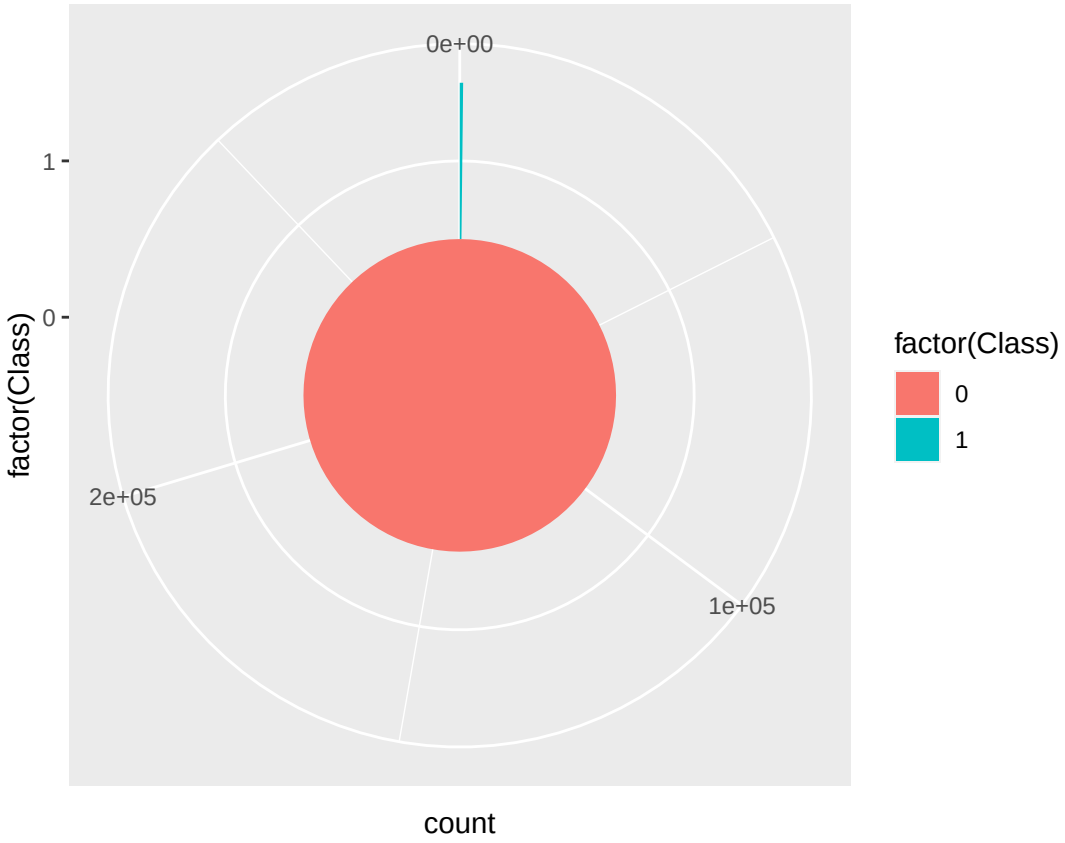
These can be some indications of fraud detection.

Data Visualisation

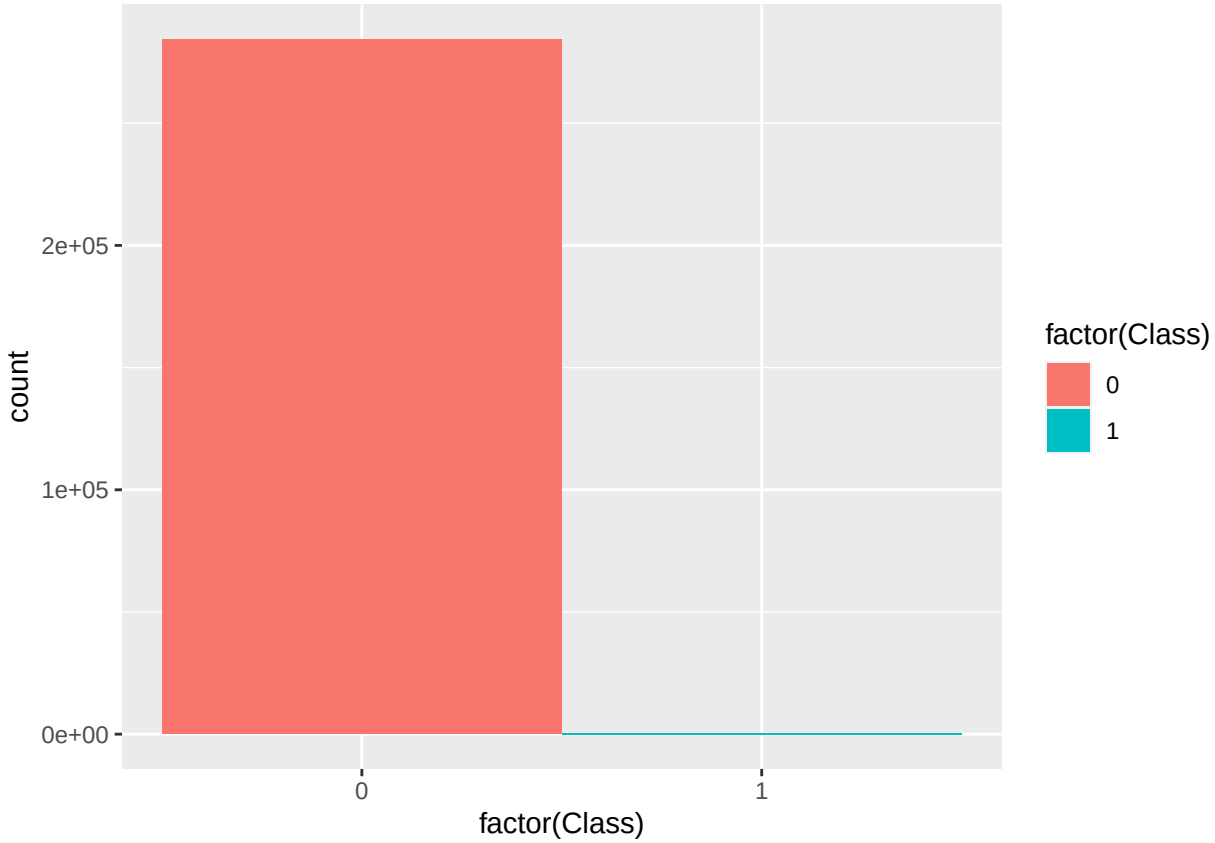
Target data visualisation:

We now will try to visualize the two target dataset.

```
library(ggplot2)
ggplot(df,aes(x = factor(Class),fill = factor(Class)))+
  geom_bar(width = 1)+
  coord_polar(theta = "y")
```




```
ggplot(df,aes(x = factor(Class),fill = factor(Class)))+
  geom_bar(width = 1)
```



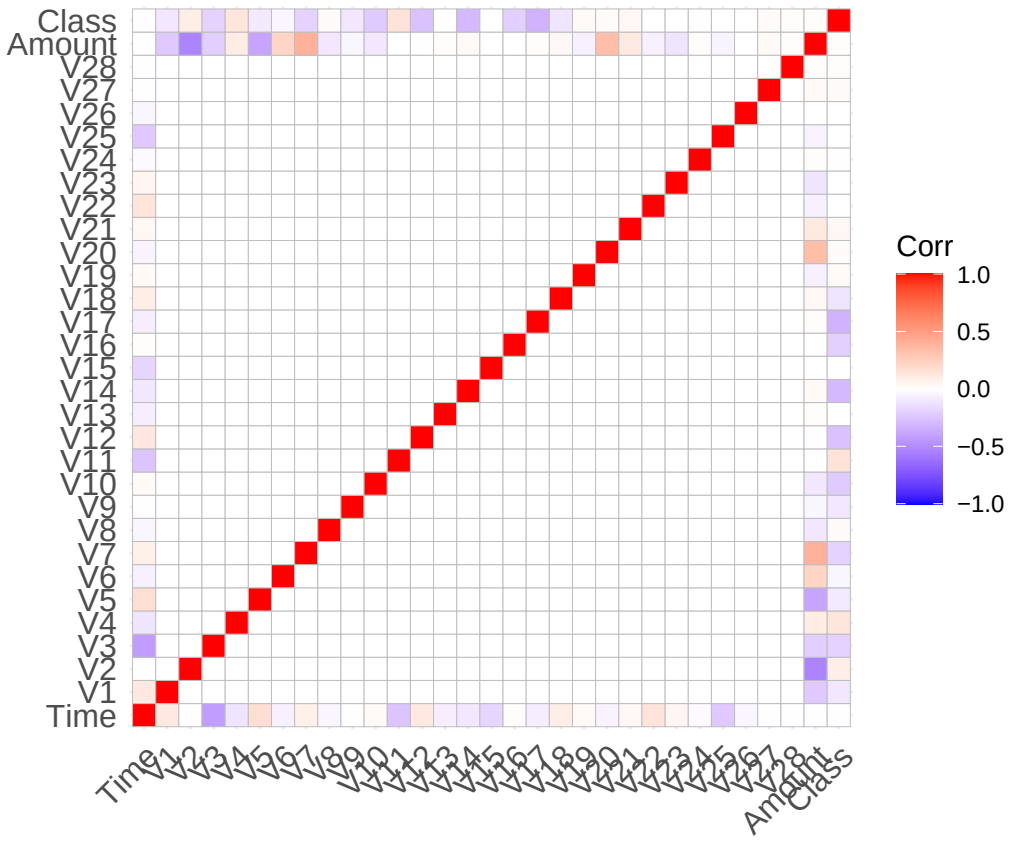
As we can see : * The data is highly unbalanced. * Due to highly unbalanced data, the classification model will bias its prediction towards the majority class, No Fraud. * We need to balance the data to do the analysis.

Feature selection for modeling:

We shall use correlation matrix to select features required for modeling.

Correlation matrix:

```
library(ggcorrplot)
corplot<-cor(df)
ggcorrplot(corplot)
```



* For feature selection, we will exclude the features having correlation values between [-0.1,0.1]. * V4, V11 are positively correlated and V7, V3, V16, V10, V12, V14, V17 are negatively correlated with the Class feature.

ANOVA Test:

```
ANOV<-aov(Class~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24+V
summary(ANOV)
```

```
##                Df Sum Sq Mean Sq   F value   Pr(>F)
## V1              1   5.04    5.04  6114.506  < 2e-16 ***
## V2              1   4.09    4.09  4961.015  < 2e-16 ***
## V3              1  18.29   18.29 22165.387  < 2e-16 ***
## V4              1   8.75    8.75 10601.268  < 2e-16 ***
## V5              1   4.43    4.43  5369.690  < 2e-16 ***
## V6              1   0.94    0.94  1133.886  < 2e-16 ***
## V7              1  17.22   17.22 20874.267  < 2e-16 ***
## V8              1   0.19    0.19   235.156  < 2e-16 ***
## V9              1   4.69    4.69  5686.129  < 2e-16 ***
## V10             1  23.10   23.10 28001.920  < 2e-16 ***
## V11             1  11.78   11.78 14279.180  < 2e-16 ***
## V12             1  33.35   33.35 40426.138  < 2e-16 ***
## V13             1   0.01    0.01    12.432 0.000422 ***
## V14             1  44.96   44.96 54489.551  < 2e-16 ***
## V15             1   0.01    0.01    10.618 0.001120 **
## V16             1  18.97   18.97 22995.043  < 2e-16 ***
## V17             1  52.35   52.35 63453.125  < 2e-16 ***
## V18             1   6.10    6.10  7398.980  < 2e-16 ***
```

```
## V19              1   0.59    0.59   720.230  < 2e-16 ***
## V20              1   0.20    0.20   240.276  < 2e-16 ***
## V21              1   0.80    0.80   972.271  < 2e-16 ***
## V22              1   0.00    0.00     0.386 0.534370
## V23              1   0.00    0.00     4.292 0.038289 *
## V24              1   0.03    0.03    31.040 2.53e-08 ***
## V25              1   0.01    0.01     6.513 0.010709 *
## V26              1   0.01    0.01    11.817 0.000587 ***
## V27              1   0.15    0.15   183.976  < 2e-16 ***
## V28              1   0.04    0.04    54.134 1.88e-13 ***
## Amount           1   0.08    0.08    91.319  < 2e-16 ***
## Residuals   284777 234.95    0.00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we will take all the significant variables.As they are not same with the correlation matrix, we will right two models based on the correlation matrix, and ANOVA test.

Dataset based on Correlation matrix:

Dataset based on Correlation matrix:

```
df1=data.frame(df$V3, df$V4, df$V7,df$V10,df$V11,df$V12,df$V14,df$V16,df$V17,df$Class)
head(df1)
```

```
##       df.V3      df.V4       df.V7      df.V10     df.V11      df.V12
## 1 2.5363467  1.3781552  0.23959855  0.09079417 -0.5515995 -0.61780086
## 2 0.1664801  0.4481541 -0.07880298 -0.16697441  1.6127267  1.06523531
## 3 1.7732093  0.3797796  0.79146096  0.20764287  0.6245015  0.06608369
## 4 1.7929933 -0.8632913  0.23760894 -0.05495192 -0.2264873  0.17822823
## 5 1.5487178  0.4030339  0.59294075  0.75307443 -0.8228429  0.53819555
## 6 1.1411093 -0.1682521  0.47620095 -0.37140720  1.3412620  0.35989384
##       df.V14     df.V16      df.V17 df.Class
## 1 -0.3111694 -0.4704005  0.20797124        0
## 2 -0.1437723  0.4639170 -0.11480466        0
## 3 -0.1659459 -2.8900832  1.10996938        0
## 4 -0.2879237 -1.0596472 -0.68409279        0
## 5 -1.1196698 -0.4514492 -0.23703324        0
## 6 -0.1371337  0.4017259 -0.05813282        0
```

```
ncol(df1)
```

```
## [1] 10
```

Dataset based on ANOVA test:

```
df2=data.frame(df$V1,df$V2,df$V3, df$V4,df$V5,df$V6, df$V7,df$V8,df$V9,df$V10,df$V11,df$V12,df$V13,df$V
head(df2)
```

```
##        df.V1       df.V2     df.V3      df.V4       df.V5       df.V6
## 1 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.46238778
## 2  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.08236081
## 3 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.80049938
## 4 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.24720317
## 5 -1.1582331  0.87773675 1.5487178  0.4030339 -0.40719338  0.09592146
## 6 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.02972755
```

```
##         df.V7       df.V8      df.V9      df.V10     df.V11      df.V12
## 1  0.23959855  0.09869790  0.3637870  0.09079417 -0.5515995 -0.61780086
## 2 -0.07880298  0.08510165 -0.2554251 -0.16697441  1.6127267  1.06523531
## 3  0.79146096  0.24767579 -1.5146543  0.20764287  0.6245015  0.06608369
## 4  0.23760894  0.37743587 -1.3870241 -0.05495192 -0.2264873  0.17822823
## 5  0.59294075 -0.27053268  0.8177393  0.75307443 -0.8228429  0.53819555
## 6  0.47620095  0.26031433 -0.5686714 -0.37140720  1.3412620  0.35989384
##       df.V13     df.V14     df.V16      df.V17      df.V18      df.V19
## 1 -0.9913898 -0.3111694 -0.4704005  0.20797124  0.02579058  0.40399296
## 2  0.4890950 -0.1437723  0.4639170 -0.11480466 -0.18336127 -0.14578304
## 3  0.7172927 -0.1659459 -2.8900832  1.10996938 -0.12135931 -2.26185710
## 4  0.5077569 -0.2879237 -1.0596472 -0.68409279  1.96577500 -1.23262197
## 5  1.3458516 -1.1196698 -0.4514492 -0.23703324 -0.03819479  0.80348692
## 6 -0.3580907 -0.1371337  0.4017259 -0.05813282  0.06865315 -0.03319379
##        df.V20       df.V21      df.V24     df.V26       df.V27      df.V28
## 1  0.25141210 -0.018306778  0.06692807 -0.1891148  0.133558377 -0.02105305
## 2 -0.06908314 -0.225775248 -0.33984648  0.1258945 -0.008983099  0.01472417
## 3  0.52497973  0.247998153 -0.68928096 -0.1390966 -0.055352794 -0.05975184
## 4 -0.20803778 -0.108300452 -1.17557533 -0.2219288  0.062722849  0.06145763
## 5  0.40854236 -0.009430697  0.14126698  0.5022922  0.219422230  0.21515315
## 6  0.08496767 -0.208253515 -0.37142658  0.1059148  0.253844225  0.08108026
##   df.Class
## 1        0
## 2        0
## 3        0
## 4        0
## 5        0
## 6        0
```

Data Balancing:

Data Balancing for Model based on Correlation Plot :

```
library(knitr)
suppressMessages(library(dplyr))
suppressMessages(library(ROSE))
predictor_variables <- df1[,-10] # Select everything except response
response_variable <- df1$Class
data_balanced_both1 <- ovun.sample(df.Class ~ ., data = df1, method = "both", p=0.5,
#data_balanced_both$df.Class
```

Data Balancing for Model based on ANOVA test :

```
library(knitr)
suppressMessages(library(dplyr))
suppressMessages(library(ROSE))
predictor_variables <- df2[,-10] # Select everything except response
response_variable <- df2$Class
data_balanced_both2 <- ovun.sample(df.Class ~ ., data = df2, method = "both", p=0.5,
#data_balanced_both$df.Class
```

Calculation for Data Balancing : Sampling Strategy : It is a ratio which is the common parameter for oversampling and undersampling. Sampling Strategy : (Samples of Minority Class) / (Samples of Majority Class) In this case,

Majority Class : No Fraud Cases : 284315 samples Minority Class : Fraud Cases : 492 samples

Undersampling : Trim down the majority class samples Sampling_Strategy = 0.1 0.1 = (492) / Majority Class Samples After undersampling,

Majority Class : No Fraud Cases : 4920 samples Minority Class : Fraud Cases : 492 samples

Oversampling : Increase the minority class samples Sampling_Strategy = 0.5 0.5 = (Minority Class Samples) / 4920 After oversampling,

Majority Class : No Fraud Cases : 4920 samples Minority Class : Fraud Cases : 2460 samples Final Class Samples :

Majority Class : No Fraud Cases : 4920 samples Minority Class : Fraud Cases : 2460 samples

- We have duplicated the data for imbalnced data to deal with the potential bias in the predictions. Hence evaluation of the model using accuracy would be wrong.
- We are going to use confusion matrix, ROC-AUC graph and ROC-AUC score for model evaluation.

Data Modelling:

Data modeling for Model based on Correlation Plot :

```
dim(train1)
```

```
## [1] 800  10
```

```
dim(test1)
```

```
## [1] 200  10
```

Logistic Regression:

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Call:
## glm(formula = df.Class ~ ., family = "binomial", data = train1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.58485  -0.31264  -0.04847   0.00000   2.78875
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.1169     0.2769 -11.255  < 2e-16 ***
## df.V3        -0.3575     0.1424  -2.511 0.012024 *
## df.V4         1.1295     0.1938   5.829 5.58e-09 ***
## df.V7        -0.4009     0.1607  -2.494 0.012622 *
## df.V10       -1.1131     0.3054  -3.645 0.000267 ***
## df.V11        0.5551     0.2342   2.370 0.017785 *
## df.V12       -0.8808     0.2943  -2.993 0.002766 **
## df.V14       -1.6613     0.3422  -4.854 1.21e-06 ***
## df.V16       -1.1385     0.3803  -2.994 0.002755 **
## df.V17       -1.1557     0.4969  -2.326 0.020040 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1107.76  on 799  degrees of freedom
## Residual deviance:  231.04  on 790  degrees of freedom
## AIC: 251.04
##
## Number of Fisher Scoring iterations: 12
```

Prediction based on the model:

Confusion matrix:

```
##    predict_reg1
##       0   1
##   0 102   2
##   1   7  89
```

Roc-Auc score:

```
## [1] 0.9539263
```

Data modeling for Model based on ANOVA :

```
library(caTools)
set.seed(1)
sample <- sample.split(data_balanced_both2, SplitRatio = 0.8)
train2  <- subset(data_balanced_both2, sample == TRUE)
test2   <- subset(data_balanced_both2, sample == FALSE)
colnames(train2)
```

```
##  [1] "df.V1"    "df.V2"    "df.V3"    "df.V4"    "df.V5"    "df.V6"
##  [7] "df.V7"    "df.V8"    "df.V9"    "df.V10"   "df.V11"   "df.V12"
## [13] "df.V13"   "df.V14"   "df.V16"   "df.V17"   "df.V18"   "df.V19"
## [19] "df.V20"   "df.V21"   "df.V24"   "df.V26"   "df.V27"   "df.V28"
## [25] "df.Class"
```

Logistic Regression:

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Call:
## glm(formula = df.Class ~ ., family = "binomial", data = train2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.01708  -0.26856  -0.00051   0.00000   3.03787
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.19054    0.56361  -7.435 1.04e-13 ***
## df.V1        0.83113    0.39239   2.118 0.034166 *
## df.V2       -0.86763    0.33224  -2.611 0.009016 **
## df.V3        0.38015    0.23250   1.635 0.102044
## df.V4        1.34576    0.27844   4.833 1.34e-06 ***
## df.V5        0.49369    0.23378   2.112 0.034708 *
```

```
## df.V6       -0.60237    0.27169  -2.217 0.026613 *
## df.V7       -0.26355    0.25818  -1.021 0.307340
## df.V8       -1.29693    0.42019  -3.087 0.002025 **
## df.V9       -0.49269    0.40351  -1.221 0.222085
## df.V10      -1.87491    0.68476  -2.738 0.006180 **
## df.V11       1.15834    0.42347   2.735 0.006232 **
## df.V12      -1.70835    0.66875  -2.555 0.010633 *
## df.V13      -0.42076    0.22455  -1.874 0.060963 .
## df.V14      -2.66888    0.76039  -3.510 0.000448 ***
## df.V16      -1.58859    0.62223  -2.553 0.010678 *
## df.V17      -2.36430    1.02503  -2.307 0.021079 *
## df.V18      -0.63945    0.44544  -1.436 0.151132
## df.V19       0.57641    0.29676   1.942 0.052092 .
## df.V20       0.05453    0.41016   0.133 0.894232
## df.V21       0.44181    0.43236   1.022 0.306849
## df.V24      -0.74490    0.40292  -1.849 0.064494 .
## df.V26       0.33769    0.39240   0.861 0.389483
## df.V27      -1.16550    1.14516  -1.018 0.308788
## df.V28       0.61910    1.62858   0.380 0.703836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1107.76  on 799  degrees of freedom
## Residual deviance:  204.26  on 775  degrees of freedom
## AIC: 254.26
##
## Number of Fisher Scoring iterations: 15
```

Prediction based on the model:

Confusion matrix:

```
##    predict_reg2
##       0   1
##   0 100   4
##   1   7  89
```

Roc-Auc score:

```
## [1] 0.9443109
```