Predicting Credit Card Fraudulent Transactions Using Synthetic Data Generation

Libraries

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.4
```

```
## corrplot 0.84 loaded
```

```r
library(ROSE)
```

```
## Warning: package 'ROSE' was built under R version 3.4.4
```

```
## Loaded ROSE 0.0-3
```

```r
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.4.3
```

Credit card dataset

credit card dataset is downloaded from kaggle.com. data contains 31 variables namely Time,V1,v2,...V28,Amount,Class and having 284,807 observations. data is already scaled using PCA.Data is highly imbalanced that means,there are only 492 fraudulant transcations out of 284807 transactions.

Loading dataset

```
data=read.csv("C:\\Users\\AJIT\\Documents\\creditcard.csv")
```

Preprocessing

Since data is already scaled so we already prepeared for Exploratory data Analaysis,but before that we will check if there is any missing values.

```
sum(is.na(data))    ## No Missing data
```

```
## [1] 0
```

Exploratory Data analysis

Know about data

```
str(data)
```

```
## '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 ...
```

Discriptive measures

```
summary(data)
```

```
##       Time                V1                  V2
##  Min.   :     0   Min.   :-56.40751   Min.   :-72.71573
##  1st Qu.: 54202   1st Qu.: -0.92037   1st Qu.: -0.59855
```

```
##  Median : 84692   Median :  0.01811   Median :  0.06549
##  Mean   : 94814   Mean   :  0.00000   Mean   :  0.00000
##  3rd Qu.:139321   3rd Qu.:  1.31564   3rd Qu.:  0.80372
##  Max.   :172792   Max.   :  2.45493   Max.   : 22.05773
##        V3                  V4                  V5
##  Min.   :-48.3256   Min.   :-5.68317   Min.   :-113.74331
##  1st Qu.: -0.8904   1st Qu.:-0.84864   1st Qu.:  -0.69160
##  Median :  0.1799   Median :-0.01985   Median :  -0.05434
##  Mean   :  0.0000   Mean   : 0.00000   Mean   :   0.00000
##  3rd Qu.:  1.0272   3rd Qu.: 0.74334   3rd Qu.:   0.61193
##  Max.   :  9.3826   Max.   :16.87534   Max.   :  34.80167
##        V6                  V7                  V8
##  Min.   :-26.1605   Min.   :-43.5572   Min.   :-73.21672
##  1st Qu.: -0.7683   1st Qu.: -0.5541   1st Qu.: -0.20863
##  Median : -0.2742   Median :  0.0401   Median :  0.02236
##  Mean   :  0.0000   Mean   :  0.0000   Mean   :  0.00000
##  3rd Qu.:  0.3986   3rd Qu.:  0.5704   3rd Qu.:  0.32735
##  Max.   : 73.3016   Max.   :120.5895   Max.   : 20.00721
##        V9                  V10                 V11
##  Min.   :-13.43407   Min.   :-24.58826   Min.   :-4.79747
##  1st Qu.: -0.64310   1st Qu.: -0.53543   1st Qu.:-0.76249
##  Median : -0.05143   Median : -0.09292   Median :-0.03276
##  Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.00000
##  3rd Qu.:  0.59714   3rd Qu.:  0.45392   3rd Qu.: 0.73959
##  Max.   : 15.59500   Max.   : 23.74514   Max.   :12.01891
##       V12                 V13                 V14
##  Min.   :-18.6837   Min.   :-5.79188   Min.   :-19.2143
##  1st Qu.: -0.4056   1st Qu.:-0.64854   1st Qu.: -0.4256
##  Median :  0.1400   Median :-0.01357   Median :  0.0506
##  Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.0000
##  3rd Qu.:  0.6182   3rd Qu.: 0.66251   3rd Qu.:  0.4931
##  Max.   :  7.8484   Max.   : 7.12688   Max.   : 10.5268
##       V15                 V16                  V17
##  Min.   :-4.49894   Min.   :-14.12985   Min.   :-25.16280
##  1st Qu.:-0.58288   1st Qu.: -0.46804   1st Qu.: -0.48375
##  Median : 0.04807   Median :  0.06641   Median : -0.06568
##  Mean   : 0.00000   Mean   :  0.00000   Mean   :  0.00000
```

```
##  3rd Qu.: 0.64882   3rd Qu.:  0.52330   3rd Qu.:  0.39968
##  Max.   : 8.87774   Max.   : 17.31511   Max.   :  9.25353
##       V18                 V19                 V20
##  Min.   :-9.498746   Min.   :-7.213527   Min.   :-54.49772
##  1st Qu.:-0.498850   1st Qu.:-0.456299   1st Qu.: -0.21172
##  Median :-0.003636   Median : 0.003735   Median : -0.06248
##  Mean   : 0.000000   Mean   : 0.000000   Mean   :  0.00000
##  3rd Qu.: 0.500807   3rd Qu.: 0.458949   3rd Qu.:  0.13304
##  Max.   : 5.041069   Max.   : 5.591971   Max.   : 39.42090
##       V21                 V22                  V23
##  Min.   :-34.83038   Min.   :-10.933144   Min.   :-44.80774
##  1st Qu.: -0.22839   1st Qu.: -0.542350   1st Qu.: -0.16185
##  Median : -0.02945   Median :  0.006782   Median : -0.01119
##  Mean   :  0.00000   Mean   :  0.000000   Mean   :  0.00000
##  3rd Qu.:  0.18638   3rd Qu.:  0.528554   3rd Qu.:  0.14764
##  Max.   : 27.20284   Max.   : 10.503090   Max.   : 22.52841
##       V24                V25                 V26
##  Min.   :-2.83663   Min.   :-10.29540   Min.   :-2.60455
##  1st Qu.:-0.35459   1st Qu.: -0.31715   1st Qu.:-0.32698
##  Median : 0.04098   Median :  0.01659   Median :-0.05214
##  Mean   : 0.00000   Mean   :  0.00000   Mean   : 0.00000
##  3rd Qu.: 0.43953   3rd Qu.:  0.35072   3rd Qu.: 0.24095
##  Max.   : 4.58455   Max.   :  7.51959   Max.   : 3.51735
##       V27                  V28                Amount
##  Min.   :-22.565679   Min.   :-15.43008   Min.   :    0.00
##  1st Qu.: -0.070840   1st Qu.: -0.05296   1st Qu.:    5.60
##  Median :  0.001342   Median :  0.01124   Median :   22.00
##  Mean   :  0.000000   Mean   :  0.00000   Mean   :   88.35
##  3rd Qu.:  0.091045   3rd Qu.:  0.07828   3rd Qu.:   77.17
##  Max.   : 31.612198   Max.   : 33.84781   Max.   :25691.16
##      Class
##  Min.   :0.000000
##  1st Qu.:0.000000
##  Median :0.000000
##  Mean   :0.001728
##  3rd Qu.:0.000000
##  Max.   :1.000000
```

Correlation

Since ,data is generated using PCA that means there is no corrleation among them and this can be verify as below

```
cordata=subset(data,select=-c(Time,Class,Amount))
corre=cor(cordata)
corre
```

```
##                V1            V2            V3            V4            V5
## V1   1.000000e+00 -6.965284e-17 -5.689257e-16 -2.602863e-16  3.146931e-16
## V2  -6.965284e-17  1.000000e+00  5.207402e-17 -1.613213e-16  1.119124e-16
## V3  -5.689257e-16  5.207402e-17  1.000000e+00 -2.229734e-16 -6.014871e-16
## V4  -2.602863e-16 -1.613213e-16 -2.229734e-16  1.000000e+00 -1.841492e-15
## V5   3.146931e-16  1.119124e-16 -6.014871e-16 -1.841492e-15  1.000000e+00
## V6   1.492868e-16  3.870545e-16  1.427210e-15 -4.247485e-16  6.266854e-16
## V7   7.841775e-17 -1.307637e-16  2.297393e-16 -7.423988e-17 -2.011798e-17
## V8  -5.446145e-17 -2.461255e-17 -7.356493e-17  6.405396e-16  5.160094e-16
## V9   3.813198e-17 -1.123192e-16  1.037123e-16  5.956330e-16  4.855550e-16
## V10  5.323676e-17 -1.342760e-16  2.047704e-16 -1.044370e-16  1.097669e-16
## V11  3.002286e-16  3.438822e-16  1.136658e-16 -2.921831e-16  7.268487e-16
## V12  1.818542e-16 -3.249832e-16  2.105842e-16 -1.939196e-16  3.915888e-16
## V13 -4.924522e-17 -3.781395e-17 -3.519517e-17  1.732999e-17 -2.925906e-16
## V14  3.872776e-16 -3.806711e-16  6.698461e-16 -9.668574e-17  2.428524e-16
## V15 -9.135222e-17  6.457046e-17 -6.312482e-17  1.844076e-16  1.151762e-16
## V16  3.349857e-16  4.068406e-17  5.714038e-16 -4.182935e-17  6.014895e-16
## V17 -2.373744e-17 -6.403585e-16  9.221563e-17 -3.727928e-16  4.239453e-16
## V18  1.468961e-16  2.334236e-16  3.128313e-16 -1.514837e-17  4.134664e-16
## V19  1.649928e-16  1.202548e-17  3.456581e-16 -2.884334e-16 -1.192412e-16
## V20  1.432581e-16  8.194049e-17  7.004887e-17 -1.837991e-16 -1.930386e-16
## V21 -9.271675e-17  8.039593e-17 -1.592155e-16 -5.925622e-17 -7.207268e-17
## V22  9.611336e-17  1.701033e-16 -2.257641e-16  2.371879e-16  2.278784e-17
## V23  1.757891e-16  1.346719e-16 -7.683090e-17  2.000434e-16  1.102508e-16
## V24 -5.157132e-17 -1.071030e-16  2.526865e-17  1.606241e-16 -9.709665e-16
## V25 -2.390623e-16  1.157084e-16  1.145955e-16  6.473123e-16 -1.058767e-16
## V26 -1.264191e-16  2.620792e-16 -2.164134e-16 -4.040848e-16  3.387285e-16
## V27  9.657711e-17 -5.267197e-16  5.247791e-16 -1.059009e-16  4.526686e-16
```

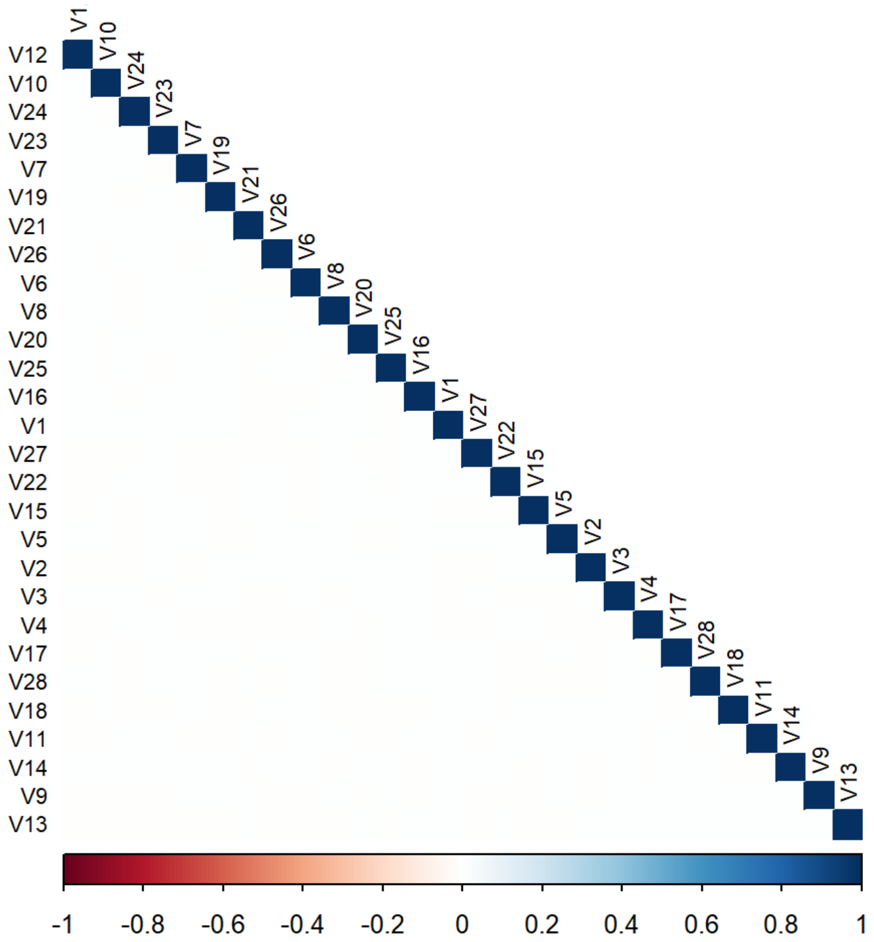
```
## V28  3.679910e-16 -3.781747e-16  7.328569e-16 -3.463299e-18 -1.776307e-16
##                V6            V7            V8            V9           V10
## V1   1.492868e-16  7.841775e-17 -5.446145e-17  3.813198e-17  5.323676e-17
## V2   3.870545e-16 -1.307637e-16 -2.461255e-17 -1.123192e-16 -1.342760e-16
## V3   1.427210e-15  2.297393e-16 -7.356493e-17  1.037123e-16  2.047704e-16
## V4  -4.247485e-16 -7.423988e-17  6.405396e-16  5.956330e-16 -1.044370e-16
## V5   6.266854e-16 -2.011798e-17  5.160094e-16  4.855550e-16  1.097669e-16
## V6   1.000000e+00 -4.980567e-17 -3.464946e-16 -9.720859e-17  1.367542e-16
## V7  -4.980567e-17  1.000000e+00 -1.220995e-17  7.581563e-18  3.058215e-16
## V8  -3.464946e-16 -1.220995e-17  1.000000e+00  4.234618e-16 -1.375841e-18
## V9  -9.720859e-17  7.581563e-18  4.234618e-16  1.000000e+00 -2.699398e-16
## V10  1.367542e-16  3.058215e-16 -1.375841e-18 -2.699398e-16  1.000000e+00
## V11  8.797284e-16 -3.654117e-16  1.371280e-16  3.139639e-16 -3.362664e-16
## V12  2.777281e-16  6.628940e-16  3.125071e-17 -1.250532e-15  8.314966e-16
## V13 -1.586079e-16 -6.222310e-17 -2.956807e-16  9.315873e-16 -4.311178e-16
## V14  3.377903e-16  3.323236e-17 -2.671262e-16  9.308315e-16  6.226081e-16
## V15 -1.122062e-16 -3.686690e-17  1.063869e-16 -8.886415e-16  4.221862e-16
## V16 -1.039022e-16  4.924499e-16  1.624649e-16 -4.609106e-16  1.765313e-16
## V17  1.246951e-16  5.445838e-16 -3.623854e-16  7.046948e-16  6.929137e-16
## V18  5.574127e-17  2.001104e-16 -3.325984e-16  1.454444e-16  4.809759e-16
## V19  8.169762e-17 -7.326312e-17 -3.349560e-16  1.175618e-16  2.297130e-17
## V20  1.161439e-16  2.160135e-16  1.261930e-16 -3.553584e-16 -1.270898e-15
## V21 -8.657658e-17  9.842322e-18  2.685700e-17  2.371978e-16  1.055033e-15
## V22 -1.153466e-16 -6.600083e-16  2.519185e-17 -1.715969e-16 -2.589804e-16
## V23  3.484483e-17 -2.641793e-16  1.918773e-16 -8.975970e-17  2.352341e-16
## V24 -1.073779e-15 -7.012260e-18 -2.115920e-16 -2.817883e-16 -8.473482e-17
## V25  5.546756e-16  1.861577e-17 -1.568089e-16  2.428573e-16 -3.467882e-16
## V26 -2.540491e-16 -7.833061e-16  2.096443e-18 -9.565914e-17 -3.766310e-16
## V27 -1.387765e-16 -1.856769e-16  3.203139e-16 -1.730431e-16 -3.667056e-16
## V28  4.321112e-16  8.208381e-17 -5.808313e-16  7.961430e-16  2.289423e-16
##               V11           V12           V13           V14           V15
## V1   3.002286e-16  1.818542e-16 -4.924522e-17  3.872776e-16 -9.135222e-17
## V2   3.438822e-16 -3.249832e-16 -3.781395e-17 -3.806711e-16  6.457046e-17
## V3   1.136658e-16  2.105842e-16 -3.519517e-17  6.698461e-16 -6.312482e-17
## V4  -2.921831e-16 -1.939196e-16  1.732999e-17 -9.668574e-17  1.844076e-16
## V5   7.268487e-16  3.915888e-16 -2.925906e-16  2.428524e-16  1.151762e-16
## V6   8.797284e-16  2.777281e-16 -1.586079e-16  3.377903e-16 -1.122062e-16
```

```
## V7  -3.654117e-16  6.628940e-16 -6.222310e-17  3.323236e-17 -3.686690e-17
## V8   1.371280e-16  3.125071e-17 -2.956807e-16 -2.671262e-16  1.063869e-16
## V9   3.139639e-16 -1.250532e-15  9.315873e-16  9.308315e-16 -8.886415e-16
## V10 -3.362664e-16  8.314966e-16 -4.311178e-16  6.226081e-16  4.221862e-16
## V11  1.000000e+00 -6.271674e-16  4.003475e-16 -7.695011e-17  2.088049e-16
## V12 -6.271674e-16  1.000000e+00 -2.294537e-14  4.339276e-16 -2.845353e-16
## V13  4.003475e-16 -2.294537e-14  1.000000e+00  1.432712e-15 -1.094370e-16
## V14 -7.695011e-17  4.339276e-16  1.432712e-15  1.000000e+00 -2.954878e-16
## V15  2.088049e-16 -2.845353e-16 -1.094370e-16 -2.954878e-16  1.000000e+00
## V16  1.680389e-16  4.961492e-16  4.758591e-16 -8.132066e-16  9.896690e-16
## V17  6.731052e-16 -3.581485e-16  7.757847e-17  1.149633e-15 -5.770624e-16
## V18  9.846238e-17 -6.057951e-16  2.424779e-16 -2.203375e-16  6.815925e-16
## V19 -1.095230e-15  1.822685e-16 -1.202426e-16  2.346856e-16 -1.439421e-15
## V20 -2.069629e-16  2.525286e-16  3.699481e-17 -2.180906e-17  1.754788e-16
## V21  9.506204e-18  5.754933e-16  1.423086e-16 -2.100761e-16  5.272469e-17
## V22  1.040834e-17 -6.489571e-17 -4.945052e-17  6.148449e-16 -3.438947e-16
## V23  1.282108e-16  2.837502e-16 -6.830450e-16  2.297548e-16  9.564145e-17
## V24  1.649224e-15  4.385884e-16 -6.517493e-16  3.197605e-17 -4.483148e-16
## V25 -6.049823e-16 -1.158139e-17 -9.572245e-17 -3.550131e-17  2.180887e-16
## V26 -1.124197e-16  1.755297e-16 -1.371594e-16 -2.415534e-17  1.018833e-16
## V27 -1.687641e-16 -3.083437e-16 -4.856405e-16  4.264244e-18 -1.248456e-15
## V28 -3.465876e-16  7.010326e-16  1.084762e-15  2.404893e-15 -1.121411e-15
##               V16           V17           V18           V19           V20
## V1   3.349857e-16 -2.373744e-17  1.468961e-16  1.649928e-16  1.432581e-16
## V2   4.068406e-17 -6.403585e-16  2.334236e-16  1.202548e-17  8.194049e-17
## V3   5.714038e-16  9.221563e-17  3.128313e-16  3.456581e-16  7.004887e-17
## V4  -4.182935e-17 -3.727928e-16 -1.514837e-17 -2.884334e-16 -1.837991e-16
## V5   6.014895e-16  4.239453e-16  4.134664e-16 -1.192412e-16 -1.930386e-16
## V6  -1.039022e-16  1.246951e-16  5.574127e-17  8.169762e-17  1.161439e-16
## V7   4.924499e-16  5.445838e-16  2.001104e-16 -7.326312e-17  2.160135e-16
## V8   1.624649e-16 -3.623854e-16 -3.325984e-16 -3.349560e-16  1.261930e-16
## V9  -4.609106e-16  7.046948e-16  1.454444e-16  1.175618e-16 -3.553584e-16
## V10  1.765313e-16  6.929137e-16  4.809759e-16  2.297130e-17 -1.270898e-15
## V11  1.680389e-16  6.731052e-16  9.846238e-17 -1.095230e-15 -2.069629e-16
## V12  4.961492e-16 -3.581485e-16 -6.057951e-16  1.822685e-16  2.525286e-16
## V13  4.758591e-16  7.757847e-17  2.424779e-16 -1.202426e-16  3.699481e-17
## V14 -8.132066e-16  1.149633e-15 -2.203375e-16  2.346856e-16 -2.180906e-17
```

```
## V15  9.896690e-16 -5.770624e-16  6.815925e-16 -1.439421e-15  1.754788e-16
## V16  1.000000e+00  1.676170e-15 -2.711204e-15  1.119911e-15  3.468227e-16
## V17  1.676170e-15  1.000000e+00 -5.244170e-15  3.767476e-16 -8.851568e-16
## V18 -2.711204e-15 -5.244170e-15  1.000000e+00 -2.674692e-15 -3.714489e-16
## V19  1.119911e-15  3.767476e-16 -2.674692e-15  1.000000e+00  2.875816e-16
## V20  3.468227e-16 -8.851568e-16 -3.714489e-16  2.875816e-16  1.000000e+00
## V21 -4.003975e-16 -9.524938e-16 -1.207426e-15  5.810910e-16 -1.172015e-15
## V22  2.544008e-16 -3.249489e-16 -5.371814e-16 -1.007031e-15  9.587679e-16
## V23  7.052180e-16  4.373451e-16 -2.962968e-16  6.691001e-16  1.100574e-16
## V24 -3.522772e-16 -1.631683e-16 -1.808092e-16 -8.718833e-17  1.617068e-16
## V25 -3.331055e-16  7.892950e-17 -2.498278e-16  8.223861e-16 -4.976490e-18
## V26 -4.660470e-16  2.542018e-16  2.920778e-16  5.501523e-16 -3.499391e-16
## V27  8.110078e-16  6.945843e-16  2.268477e-16 -1.545547e-16 -9.887404e-16
## V28  7.028481e-16 -8.344534e-17  8.010596e-16 -1.361453e-15 -2.264586e-16
##               V21           V22           V23           V24           V25
## V1  -9.271675e-17  9.611336e-17  1.757891e-16 -5.157132e-17 -2.390623e-16
## V2   8.039593e-17  1.701033e-16  1.346719e-16 -1.071030e-16  1.157084e-16
## V3  -1.592155e-16 -2.257641e-16 -7.683090e-17  2.526865e-17  1.145955e-16
## V4  -5.925622e-17  2.371879e-16  2.000434e-16  1.606241e-16  6.473123e-16
## V5  -7.207268e-17  2.278784e-17  1.102508e-16 -9.709665e-16 -1.058767e-16
## V6  -8.657658e-17 -1.153466e-16  3.484483e-17 -1.073779e-15  5.546756e-16
## V7   9.842322e-18 -6.600083e-16 -2.641793e-16 -7.012260e-18  1.861577e-17
## V8   2.685700e-17  2.519185e-17  1.918773e-16 -2.115920e-16 -1.568089e-16
## V9   2.371978e-16 -1.715969e-16 -8.975970e-17 -2.817883e-16  2.428573e-16
## V10  1.055033e-15 -2.589804e-16  2.352341e-16 -8.473482e-17 -3.467882e-16
## V11  9.506204e-18  1.040834e-17  1.282108e-16  1.649224e-15 -6.049823e-16
## V12  5.754933e-16 -6.489571e-17  2.837502e-16  4.385884e-16 -1.158139e-17
## V13  1.423086e-16 -4.945052e-17 -6.830450e-16 -6.517493e-16 -9.572245e-17
## V14 -2.100761e-16  6.148449e-16  2.297548e-16  3.197605e-17 -3.550131e-17
## V15  5.272469e-17 -3.438947e-16  9.564145e-17 -4.483148e-16  2.180887e-16
## V16 -4.003975e-16  2.544008e-16  7.052180e-16 -3.522772e-16 -3.331055e-16
## V17 -9.524938e-16 -3.249489e-16  4.373451e-16 -1.631683e-16  7.892950e-17
## V18 -1.207426e-15 -5.371814e-16 -2.962968e-16 -1.808092e-16 -2.498278e-16
## V19  5.810910e-16 -1.007031e-15  6.691001e-16 -8.718833e-17  8.223861e-16
## V20 -1.172015e-15  9.587679e-16  1.100574e-16  1.617068e-16 -4.976490e-18
## V21  1.000000e+00  3.489827e-15  6.459116e-16  1.391805e-16 -1.058544e-16
## V22  3.489827e-15  1.000000e+00  2.998995e-16  3.180808e-17 -9.676148e-16
```

```
## V23  6.459116e-16  2.998995e-16  1.000000e+00  6.662704e-17 -7.284999e-16
## V24  1.391805e-16  3.180808e-17  6.662704e-17  1.000000e+00  1.240324e-15
## V25 -1.058544e-16 -9.676148e-16 -7.284999e-16  1.240324e-15  1.000000e+00
## V26 -4.803701e-16 -3.920807e-17  1.279253e-15  1.863838e-16  2.435465e-15
## V27 -1.398538e-15  1.635775e-16  4.325298e-16 -3.050278e-16 -5.961657e-16
## V28  2.025134e-16 -5.377144e-16  1.367329e-15 -2.770212e-16  3.734279e-16
##               V26           V27           V28
## V1  -1.264191e-16  9.657711e-17  3.679910e-16
## V2   2.620792e-16 -5.267197e-16 -3.781747e-16
## V3  -2.164134e-16  5.247791e-16  7.328569e-16
## V4  -4.040848e-16 -1.059009e-16 -3.463299e-18
## V5   3.387285e-16  4.526686e-16 -1.776307e-16
## V6  -2.540491e-16 -1.387765e-16  4.321112e-16
## V7  -7.833061e-16 -1.856769e-16  8.208381e-17
## V8   2.096443e-18  3.203139e-16 -5.808313e-16
## V9  -9.565914e-17 -1.730431e-16  7.961430e-16
## V10 -3.766310e-16 -3.667056e-16  2.289423e-16
## V11 -1.124197e-16 -1.687641e-16 -3.465876e-16
## V12  1.755297e-16 -3.083437e-16  7.010326e-16
## V13 -1.371594e-16 -4.856405e-16  1.084762e-15
## V14 -2.415534e-17  4.264244e-18  2.404893e-15
## V15  1.018833e-16 -1.248456e-15 -1.121411e-15
## V16 -4.660470e-16  8.110078e-16  7.028481e-16
## V17  2.542018e-16  6.945843e-16 -8.344534e-17
## V18  2.920778e-16  2.268477e-16  8.010596e-16
## V19  5.501523e-16 -1.545547e-16 -1.361453e-15
## V20 -3.499391e-16 -9.887404e-16 -2.264586e-16
## V21 -4.803701e-16 -1.398538e-15  2.025134e-16
## V22 -3.920807e-17  1.635775e-16 -5.377144e-16
## V23  1.279253e-15  4.325298e-16  1.367329e-15
## V24  1.863838e-16 -3.050278e-16 -2.770212e-16
## V25  2.435465e-15 -5.961657e-16  3.734279e-16
## V26  1.000000e+00 -2.851245e-16 -2.952380e-16
## V27 -2.851245e-16  1.000000e+00  3.001876e-17
## V28 -2.952380e-16  3.001876e-17  1.000000e+00
```

```
corrplot(corre, order = "FPC", method = "color",
         type = "lower", tl.cex = 0.7, tl.col = rgb(0, 0, 0))
```



As,we see,there is no corrleation.

Distribution of Class variable

given, probelm is binary classification having two class 1 and 0.

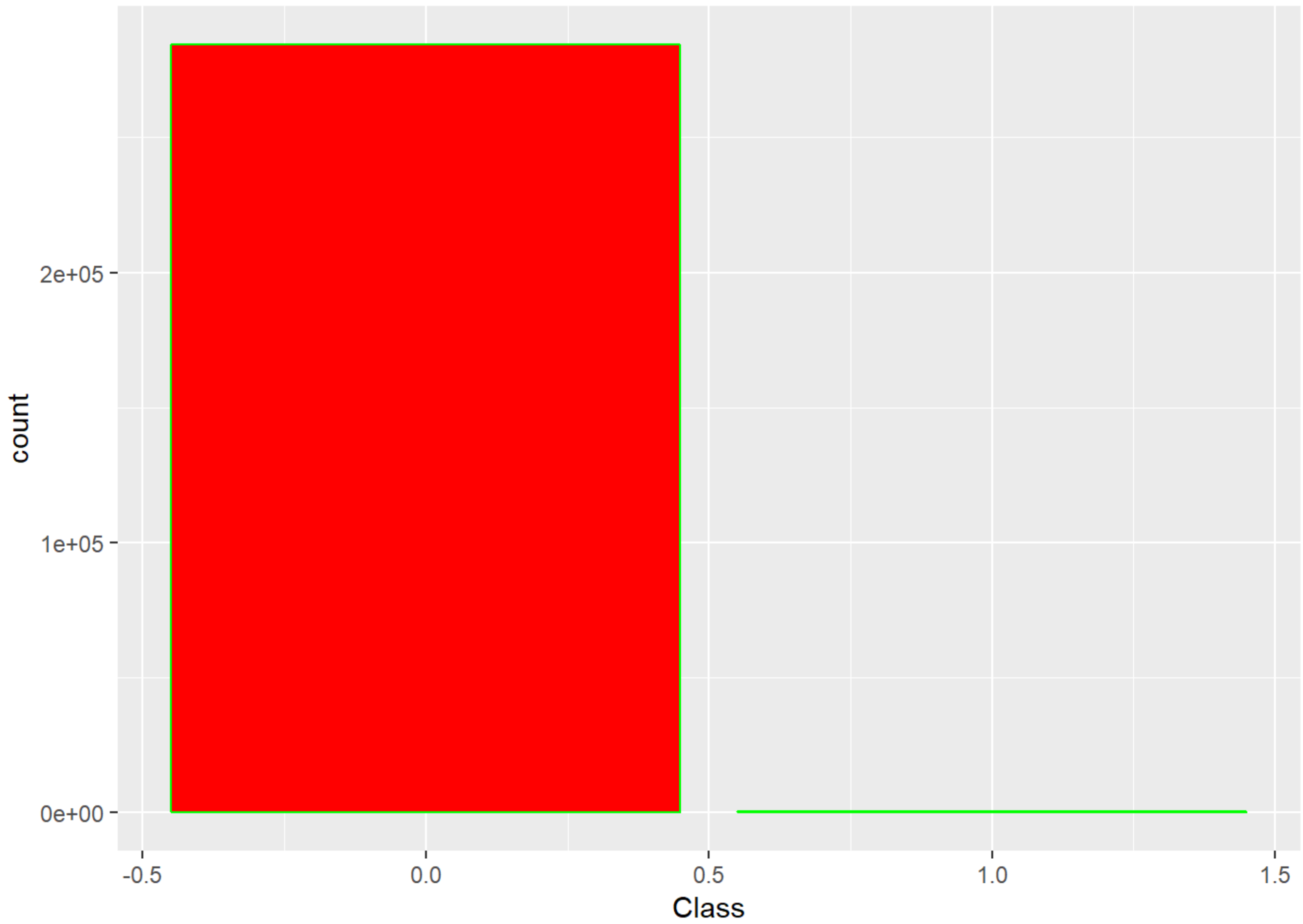
```
table(data$Class)
```

```
##
##      0      1
## 284315    492
```

```
prop.table(table(data$Class))*100
```

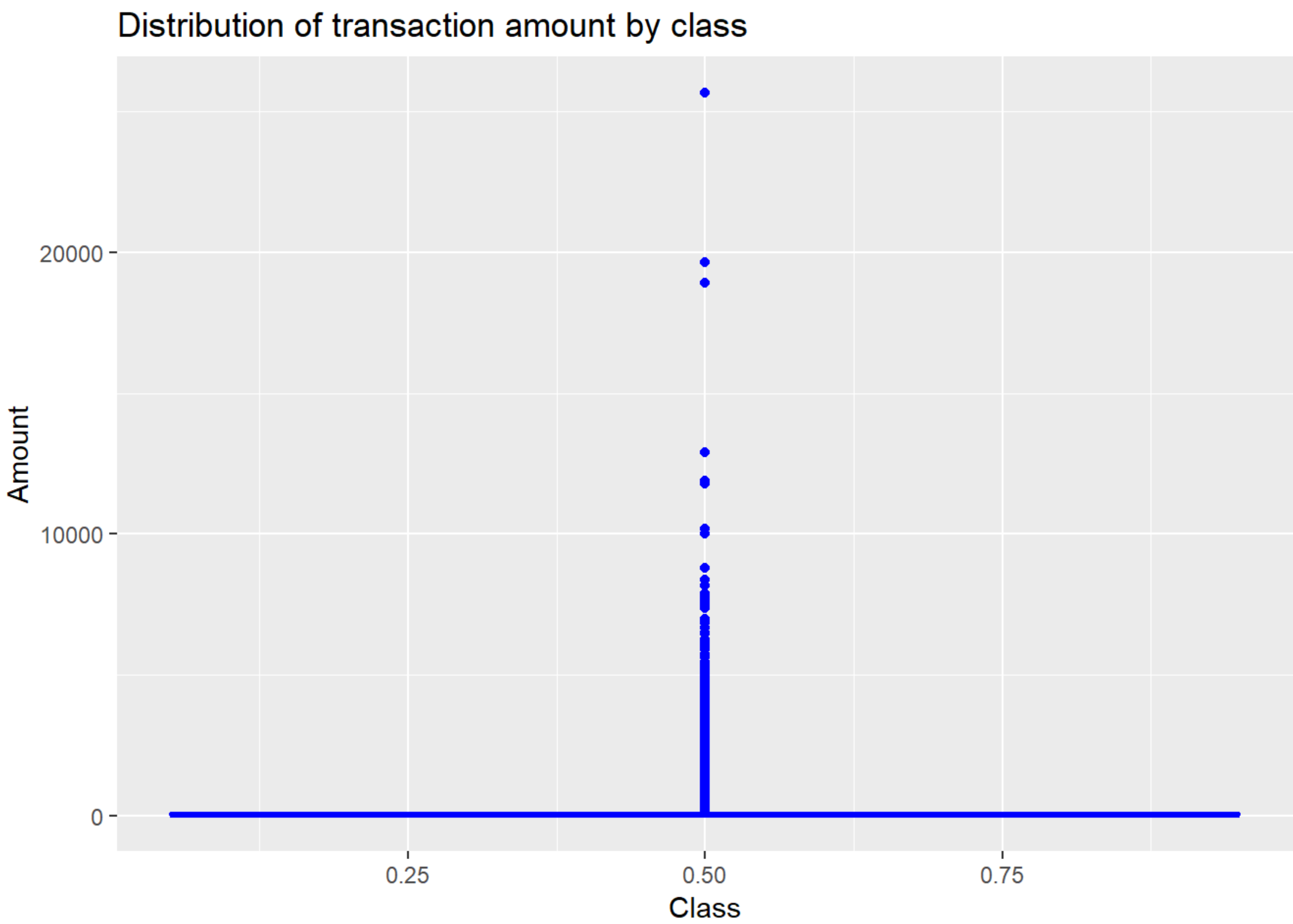
```
##
##          0          1
## 99.8272514  0.1727486
```

```
ggplot(data,aes(x=Class))+geom_bar(color="green",fill="red")
```




```
ggplot(data, aes(x = Class, y = Amount)) + geom_boxplot(color="blue") +
  ggtitle("Distribution of transaction amount by class")
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Cleary ,data is highly imbalanced

with 492 observation from positive class and 284315 from negative class. # Data Spliting

```
size<- floor(0.75 * nrow(data))
set.seed(123)
train_ind <- sample(seq_len(nrow(data)), size =size)
train <- data[train_ind, ]
test <- data[-train_ind, ]
```

Methods for Imbalanced Classification Problem

below methods are sampling methods used for imbalanced dataset.

Undersampling

Oversampling

Synthetic data generation

Cost sensitive Learning

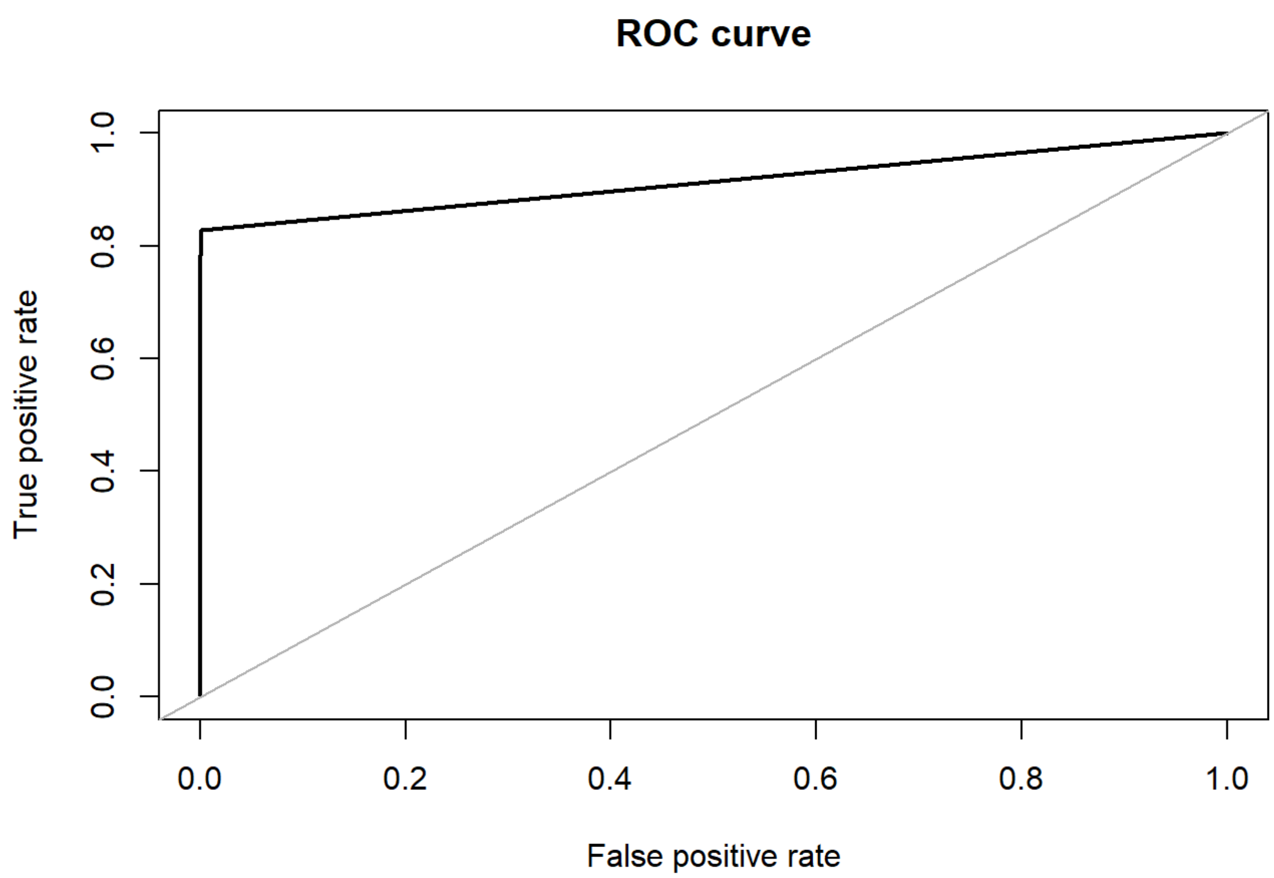
here,we use only Synthetic data generation method ,Since this method is robust one than first two methods. but,before that we will check how model perform without this method. # Modelling ## Decision tree without sampling method We will use ROC curve as metrics ,since accuarcy is not good choice while working with imbalanced data classification problem.

```
dt<- rpart(Class~ .,train)
pred<- predict(dt,test)
accuracy.meas(test$Class, pred)
```

```
##
## Call:
## accuracy.meas(response = test$Class, predicted = pred)
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.875
## recall: 0.731
## F: 0.398
```

Above metrics is not enough to evaluate our model,so we use AUC.

```
roc.curve(test$Class, pred, plotit = T)
```




```
## Area under the curve (AUC): 0.914
```

Decision tree with sampling method

In R,there is package called ROSE(Random Over Sapmling Examples) used for implementing sampling method.

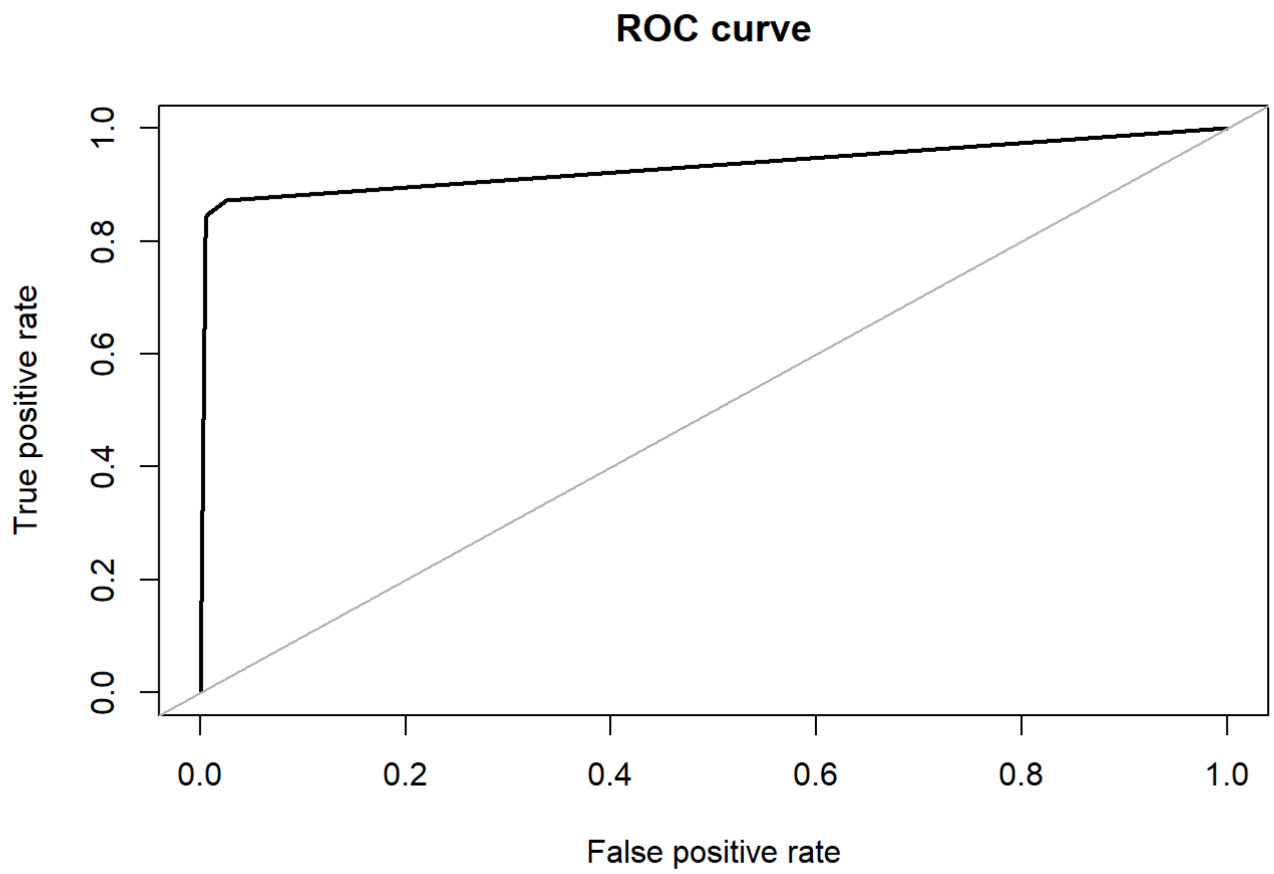
```
data.rose <- ROSE(Class~.,train, seed = 1)$data
table(data.rose$Class)
```

```
## 
##      0      1 
## 106697 106908
```

```
dt.rose <- rpart(Class ~ .,data.rose)
pred.tree.rose <- predict(dt.rose,test)
accuracy.meas(test$Class, pred.tree.rose)
```

```
## 
## Call: 
## accuracy.meas(response = test$Class, predicted = pred.tree.rose)
## 
## Examples are labelled as positive when predicted is greater than 0.5 
## 
## precision: 0.139
## recall: 0.851
## F: 0.120
```

```
roc.curve(test$Class, pred.tree.rose,plotit = T)
```




```
## Area under the curve (AUC): 0.932
```

Clearly sampling method is robust one with AUC 0.932

Logistic Regression Without Sampling

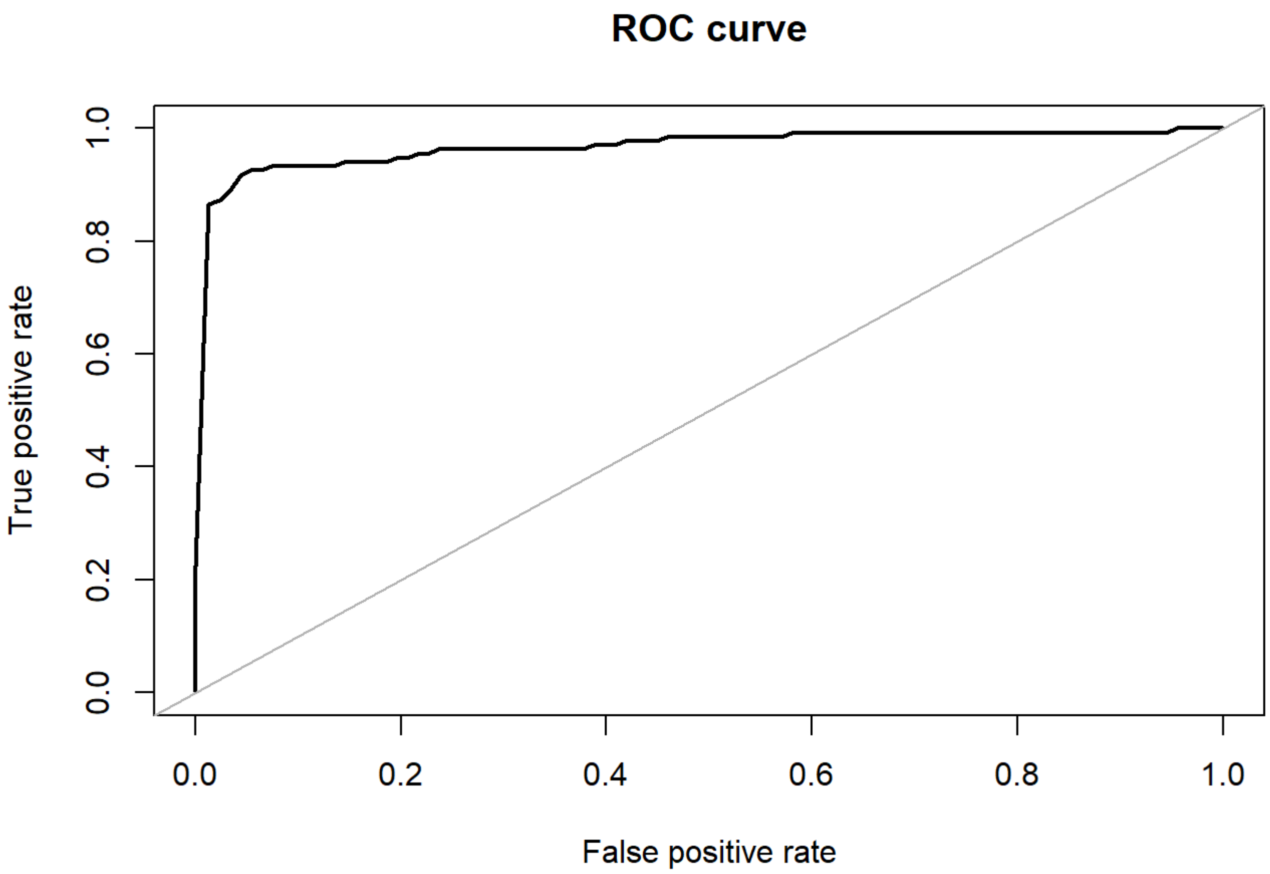
```
glm=glm(Class~.,train,family = binomial)
```

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

```
pre<- predict(glm,test)
accuracy.meas(test$Class, pre)
```

```
##
## Call:
## accuracy.meas(response = test$Class, predicted = pre)
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.883
## recall: 0.507
## F: 0.322
```

```
roc.curve(test$Class, pre,plotit = T)
```




```
## Area under the curve (AUC): 0.967
```

Logistic Regression With Sampling

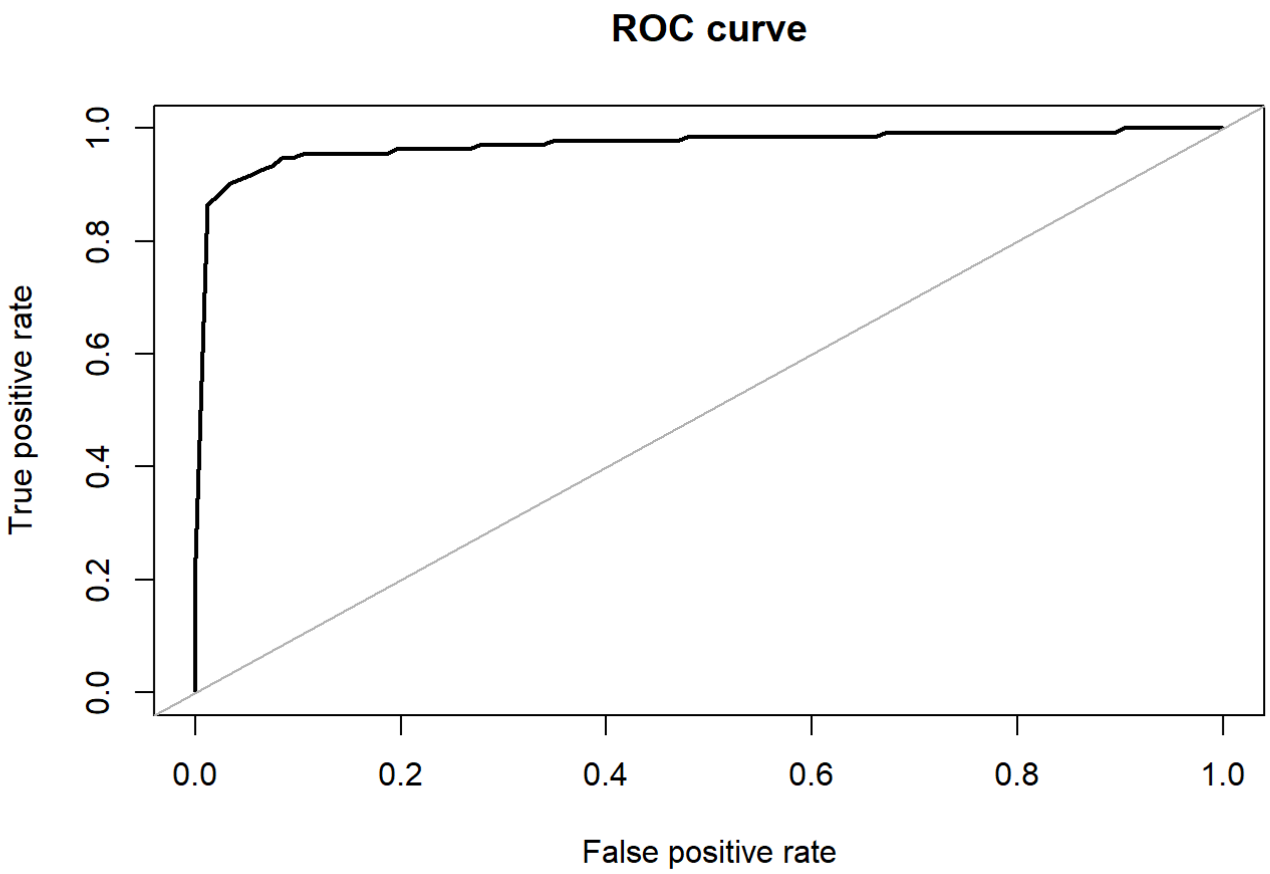
```
glm=glm(Class~.,data.rose,family = binomial)
```

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

```
pre<- predict(glm,test)
accuracy.meas(test$Class, pre)
```

```
##
## Call:
## accuracy.meas(response = test$Class, predicted = pre)
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.186
## recall: 0.858
## F: 0.153
```

```
roc.curve(test$Class, pre,plotit = T)
```




```
## Area under the curve (AUC): 0.971
```

Again Sampling technique outperformed with AUC 0.971

Summary

Here,we have implement only two models decion tree and Logistic regression. we get robust model logistic regression with sampling. we,can still improve our AUC while trying other models. we can also use parameter tuning technique to optimized our models.