Untitled

Credit Card Fraud Detection

Many Financial institutions across the Globe are faced with various kinds of frauds. This project deals with one of them and that is Credit Card Frauds. The main aim of this project is to develop a Credit Card Fraud Detection using various techniques. The main problem with this kind of data is the imbalance between the fraud and non-fradulent transactions which is a major challenge in developing an accurate prediction model. This can be classified as Imbalanced Classification. The term imbalanced refer to the disparity encountered in the dependent (response) variable. Therefore, an imbalanced classification is one in which the dependent variable has imbalanced proportion of classes. In other words, a data set that exhibits an unequal distribution between its classes is considered to be imbalanced.

Library

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

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

Data

Initial Data Findings

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

The basic structure of the credit card data provides following initial details:-

1. The dataset has in total 284,807 transactions
2. Out of which only 492 are classified as fraud transaction
3. This shows that 0.172% of all transactions are fraud transactions
4. Columns V1 to V28 have been tranformed using PCA due to privacy reasons
5. Time column displays seconds elasped between each transaction and the first transaction
6. Amount column displays transaction amount
7. Lastly class column is a response variable and takes only binary values such as 1 for fraud transactions and 0 for otherwise

Getting Data Ready for Analysis

As you can see, the data set contains 31 variable. Class is the response variable. While Time, V1 to V28 and Amount are dependent variables. Let's check the severity of imbalance in this data set:

```
## '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 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

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

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

```
##
## Call:
## accuracy.meas(response = testing$Class, predicted = PredictionTree[,
##     2])
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.957
## recall: 0.732
## F: 0.415
```

```
## Area under the curve (AUC): 0.898
```

As we see, this data set contains only 1% of positive cases and 99% of negative cases. This is a severely imbalanced data set. We'll use the sampling techniques and try to improve this prediction accuracy.

Methods Used for Analysis of Imbalanced Dataset

The methods that I am using are called Sampling Methods. These methods aim to modify an imbalanced data into balanced distribution. The modification occurs by altering the size of original data set and provide the same proportion of balance. Following are the sampling methods that will used in my analysis:-

1. Undersampling
2. Oversampling
3. Synthetic Data Generation
4. Cost Sensitive Learning

Oversampling

```
##
##      0      1
## 213237 213237
```

In this case, originally we had 213237 negative observations. So, I instructed this line of code to over sample minority class until it reaches 213237 and the total data set comprises of 426474 samples.

Undersampling

```
##
##   0   1
## 369 369
```

Balanced Both

```
##
##      0      1
## 106697 106909
```

Syntehtic Data Generation

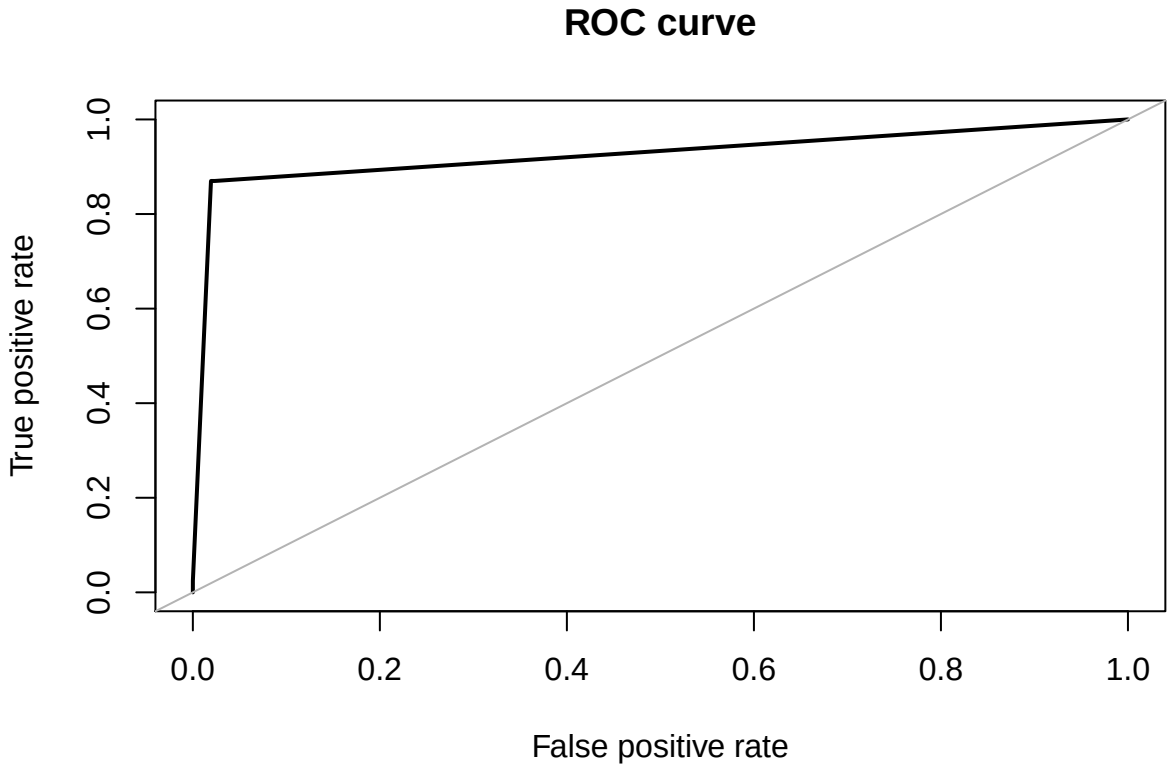
```
##
##      0      1
## 106697 106909
```

This generated data has size equal to the original data set (213606 observations). Now, we've balanced data sets using 4 techniques. Let's compute the model using each data and evaluate its accuracy.

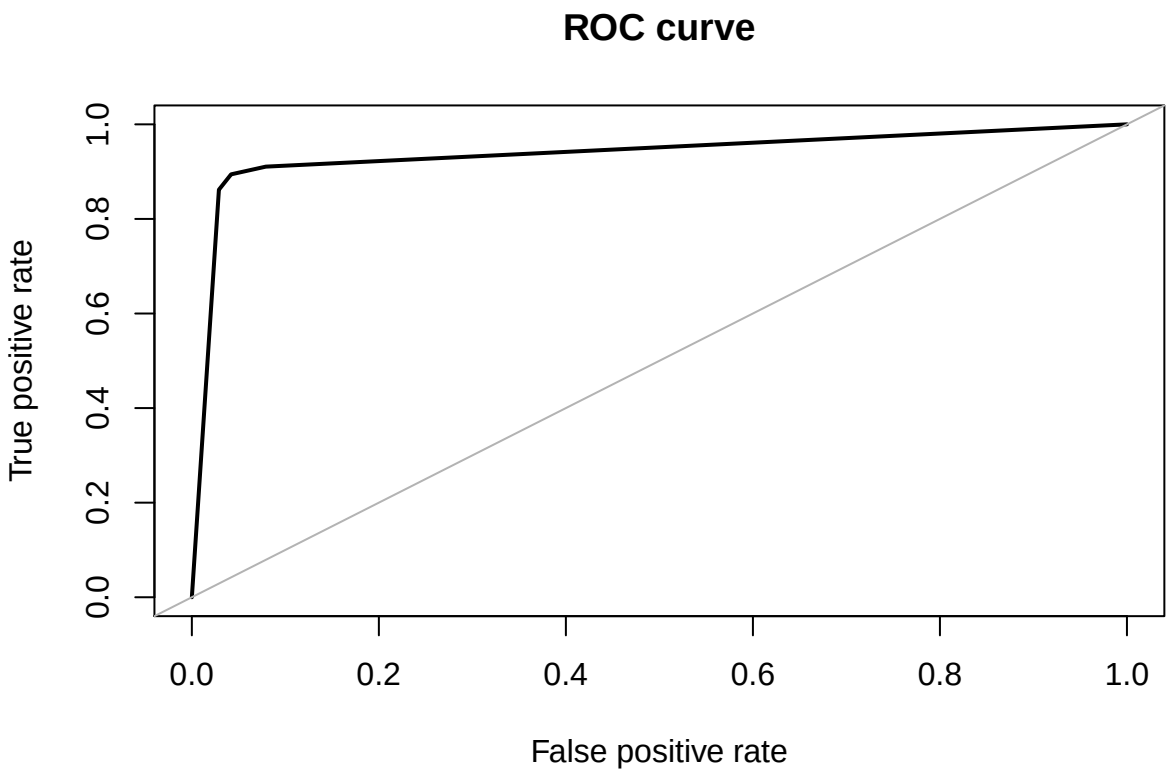
Build Decision Tree Models

Prediction on unseen data

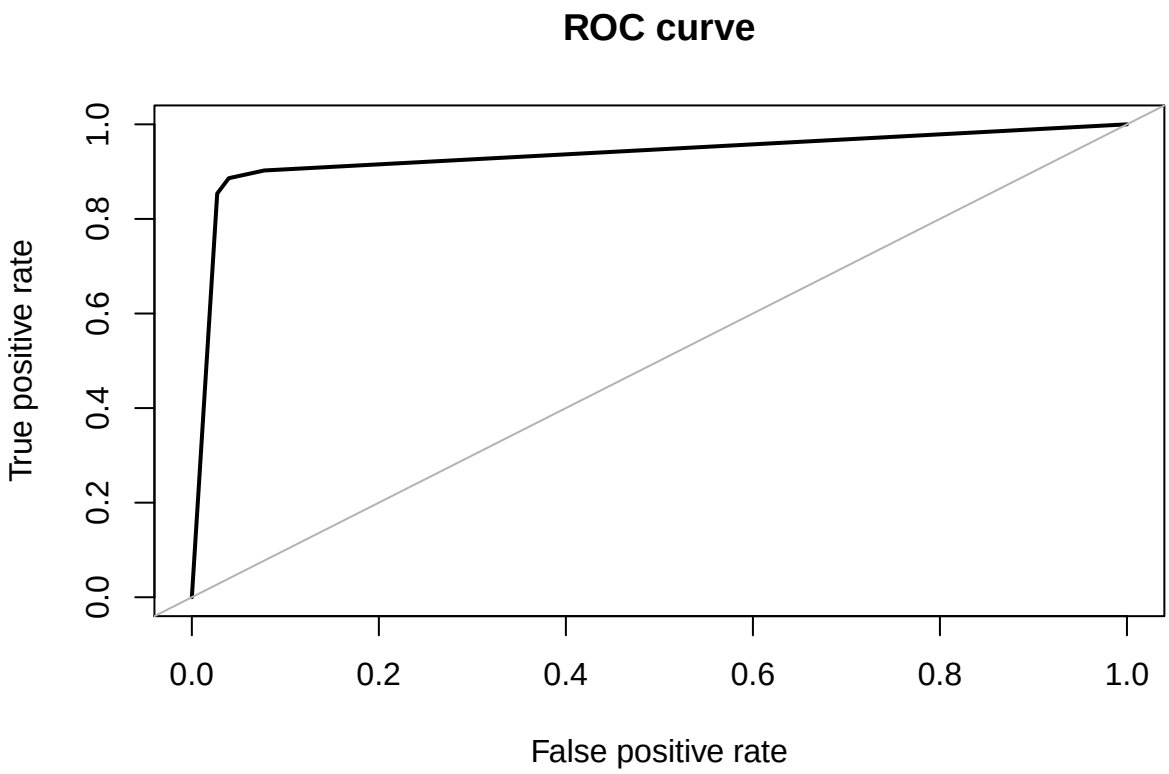
Accuracy of Models



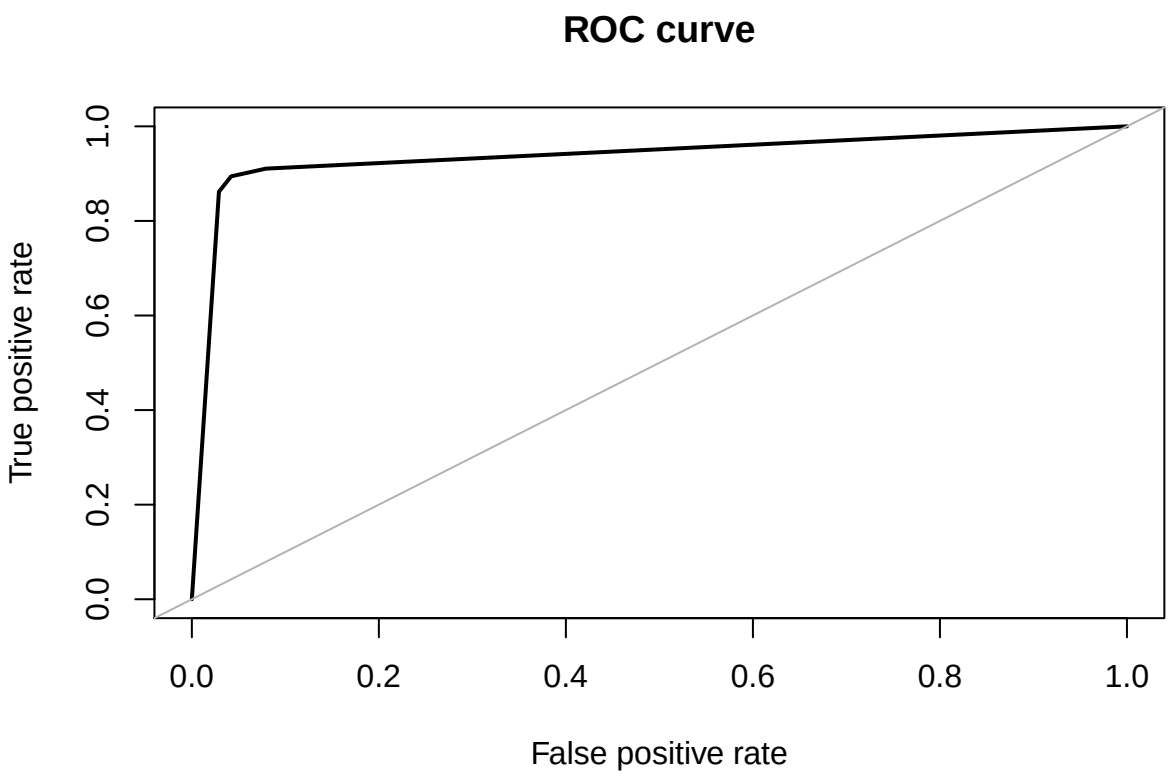

```
## Area under the curve (AUC): 0.925
```




```
## Area under the curve (AUC): 0.937
```




```
## Area under the curve (AUC): 0.934
```




```
## Area under the curve (AUC): 0.937
```

```
##
## Call:
## ROSE.eval(formula = Class ~ ., data = training, learner = rpart,
##     extr.pred = function(obj) obj[, 2], method.assess = "holdout",
##     seed = 1)
##
## Holdout estimate of auc: 0.932
```

Conclusion

Hence, we get the highest accuracy from data obtained using Over, Under and Both algorithm. Another method to check the model accuracy is using holdout and bagging. This helps us to ensure that our resultant predictions doesn't suffer from high variance.

We see that our accuracy retains at ~ 0.93 and shows that our predictions aren't suffering from high variance.