# Mini Project - 4

## Project Report by James Peter

# Table of Contents

## Contents

# 1. Project Objective

- The objective is to build a classification model which accurately discriminates the customers who will respond to personal loan offers from the non-responders. The classification will be tried using three different models CART, Random Forest, Neural Network. Based on their performance both on the validation scores and holdout data one of them will be choosen as best discrimination model.
- Along with finding a best single model which can be intrepreted to find the most influential variables, for achieving higher accurary an ensemble model is build which will allow to make accurate predictions by combining the strengths of the three modelS albeit it results in a less interpretable model.

# 2. Data Exploration

**Check for presence of missing values**

```
                 CUST_ID                   TARGET                      AGE
                       0                        0                        0
                  GENDER                  BALANCE               OCCUPATION
                       0                        0                        0
                 AGE_BKT                      SCR           HOLDING_PERIOD
                       0                        0                        0
                ACC_TYPE              ACC_OP_DATE     LEN_OF_RLTN_IN_MNTH
                       0                        0                        0
         NO_OF_L_CR_TXNS          NO_OF_L_DR_TXNS        TOT_NO_OF_L_TXNS
                       0                        0                        0
NO_OF_BR_CSH_WDL_DR_TXNS        NO_OF_ATM_DR_TXNS        NO_OF_NET_DR_TXNS
                       0                        0                        0
       NO_OF_MOB_DR_TXNS        NO_OF_CHQ_DR_TXNS               FLG_HAS_CC
                       0                        0                        0
              AMT_ATM_DR        AMT_BR_CSH_WDL_DR               AMT_CHQ_DR
                       0                        0                        0
              AMT_NET_DR               AMT_MOB_DR                 AMT_L_DR
                       0                        0                        0
        FLG_HAS_ANY_CHGS AMT_OTH_BK_ATM_USG_CHGS     AMT_MIN_BAL_NMC_CHGS
                       0                        0                        0
    NO_OF_IW_CHQ_BNC_TXNS    NO_OF_OW_CHQ_BNC_TXNS      AVG_AMT_PER_ATM_TXN
                       0                        0                        0
 AVG_AMT_PER_CSH_WDL_TXN      AVG_AMT_PER_CHQ_TXN      AVG_AMT_PER_NET_TXN
                       0                        0                        0
     AVG_AMT_PER_MOB_TXN          FLG_HAS_NOMINEE         FLG_HAS_OLD_LOAN
                       0                        0                        0
                  random
                       0
```

**Observation:**

- The data doesn't contain any missing values, often the data has missing values which needs to be preprocessed before carrying out analysis and model training based on the type of model that is being conisdered suitable for the analsis.

### 2.1. Data description

```
'data.frame':   20000 obs. of  40 variables:
 $ CUST_ID                 : Factor w/ 20000 levels "C1","C10","C100",..: 17699 16532 11027 17984 2363 
 $ TARGET                  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ AGE                     : int  27 47 40 53 36 42 30 53 42 30 ...
 $ GENDER                  : Factor w/ 3 levels "F","M","O": 2 2 2 2 2 1 2 1 1 2 ...
 $ BALANCE                 : num  3384 287489 18217 71720 1671623 ...
 $ OCCUPATION              : Factor w/ 4 levels "PROF","SAL","SELF-EMP",..: 3 2 3 2 1 1 1 2 3 1 ...
 $ AGE_BKT                 : Factor w/ 7 levels "<25",">50","26-30",..: 3 7 5 2 5 6 3 2 6 3 ...
 $ SCR                     : int  776 324 603 196 167 493 479 562 105 170 ...
 $ HOLDING_PERIOD          : int  30 28 2 13 24 26 14 25 15 13 ...
 $ ACC_TYPE                : Factor w/ 2 levels "CA","SA": 2 2 2 1 2 2 2 1 2 2 ...
 $ ACC_OP_DATE             : Factor w/ 4869 levels "01-01-00","01-01-01",..: 3270 1806 3575 993 2861 86
 $ LEN_OF_RLTN_IN_MNTH     : int  146 104 61 107 185 192 177 99 88 111 ...
 $ NO_OF_L_CR_TXNS         : int  7 8 10 36 20 5 6 14 18 14 ...
 $ NO_OF_L_DR_TXNS         : int  3 2 5 14 1 2 6 3 14 8 ...
 $ TOT_NO_OF_L_TXNS        : int  10 10 15 50 21 7 12 17 32 22 ...
 $ NO_OF_BR_CSH_WDL_DR_TXNS: int  0 0 1 4 1 1 0 3 6 3 ...
 $ NO_OF_ATM_DR_TXNS       : int  1 1 1 2 0 1 1 0 2 1 ...
 $ NO_OF_NET_DR_TXNS       : int  2 1 1 3 0 0 1 0 4 0 ...
 $ NO_OF_MOB_DR_TXNS       : int  0 0 0 1 0 0 0 0 1 0 ...
 $ NO_OF_CHQ_DR_TXNS       : int  0 0 2 4 0 0 4 0 1 4 ...
 $ FLG_HAS_CC              : int  0 0 0 0 0 1 0 0 1 0 ...
 $ AMT_ATM_DR              : int  13100 6600 11200 26100 0 18500 6200 0 35400 18000 ...
 $ AMT_BR_CSH_WDL_DR       : int  0 0 561120 673590 808480 379310 0 945160 198430 869880 ...
 $ AMT_CHQ_DR              : int  0 0 49320 60780 0 0 10580 0 51490 32610 ...
 $ AMT_NET_DR              : num  973557 799813 997570 741506 0 ...
 $ AMT_MOB_DR              : int  0 0 0 71388 0 0 0 0 170332 0 ...
 $ AMT_L_DR                : num  986657 806413 1619210 1573364 808480 ...
 $ FLG_HAS_ANY_CHGS        : int  0 1 1 0 0 0 1 0 0 0 ...
 $ AMT_OTH_BK_ATM_USG_CHGS : int  0 0 0 0 0 0 0 0 0 0 ...
 $ AMT_MIN_BAL_NMC_CHGS    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ NO_OF_IW_CHQ_BNC_TXNS   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ NO_OF_OW_CHQ_BNC_TXNS   : int  0 0 1 0 0 0 0 0 0 0 ...
 $ AVG_AMT_PER_ATM_TXN     : num  13100 6600 11200 13050 0 ...
 $ AVG_AMT_PER_CSH_WDL_TXN : num  0 0 561120 168398 808480 ...
 $ AVG_AMT_PER_CHQ_TXN     : num  0 0 24660 15195 0 ...
 $ AVG_AMT_PER_NET_TXN     : num  486779 799813 997570 247169 0 ...
 $ AVG_AMT_PER_MOB_TXN     : num  0 0 0 71388 0 ...
 $ FLG_HAS_NOMINEE         : int  1 1 1 1 1 1 0 1 1 0 ...
 $ FLG_HAS_OLD_LOAN        : int  1 0 1 0 0 1 1 1 1 0 ...
 $ random                  : num  1.14e-05 1.11e-04 1.20e-04 1.37e-04 1.74e-04 ...
```

### 2.2. Data Summary

```
     TARGET             AGE         GENDER        BALANCE
 Min.   :0.0000   Min.   :21.00   F: 5433   Min.   :      0
 1st Qu.:0.0000   1st Qu.:30.00   M:14376   1st Qu.:  64754
 Median :0.0000   Median :38.00   O:  191   Median : 231676
 Mean   :0.1256   Mean   :38.42             Mean   : 511362
 3rd Qu.:0.0000   3rd Qu.:46.00             3rd Qu.: 653877
 Max.   :1.0000   Max.   :55.00             Max.   :8360431
```

```
   OCCUPATION        SCR        HOLDING_PERIOD  ACC_TYPE
 PROF    :5417   Min.   :100.0   Min.   : 1.00   CA: 4241
 SAL     :5855   1st Qu.:227.0   1st Qu.: 7.00   SA:15759
 SELF-EMP:3568   Median :364.0   Median :15.00
 SENP    :5160   Mean   :440.2   Mean   :14.96
                 3rd Qu.:644.0   3rd Qu.:22.00
                 Max.   :999.0   Max.   :31.00
 LEN_OF_RLTN_IN_MNTH NO_OF_L_CR_TXNS NO_OF_L_DR_TXNS  TOT_NO_OF_L_TXNS
 Min.   : 29.0       Min.   : 0.00   Min.   : 0.000   Min.   :  0.00
 1st Qu.: 79.0       1st Qu.: 6.00   1st Qu.: 2.000   1st Qu.:  9.00
 Median :125.0       Median :10.00   Median : 5.000   Median : 14.00
 Mean   :125.2       Mean   :12.35   Mean   : 6.634   Mean   : 18.98
 3rd Qu.:172.0       3rd Qu.:14.00   3rd Qu.: 7.000   3rd Qu.: 21.00
 Max.   :221.0       Max.   :75.00   Max.   :74.000   Max.   :149.00
 NO_OF_BR_CSH_WDL_DR_TXNS NO_OF_ATM_DR_TXNS NO_OF_NET_DR_TXNS
 Min.   : 0.000           Min.   : 0.000    Min.   : 0.000
 1st Qu.: 1.000           1st Qu.: 0.000    1st Qu.: 0.000
 Median : 1.000           Median : 1.000    Median : 0.000
 Mean   : 1.883           Mean   : 1.029    Mean   : 1.172
 3rd Qu.: 2.000           3rd Qu.: 1.000    3rd Qu.: 1.000
 Max.   :15.000           Max.   :25.000    Max.   :22.000
 NO_OF_MOB_DR_TXNS NO_OF_CHQ_DR_TXNS   FLG_HAS_CC       AMT_ATM_DR
 Min.   : 0.0000   Min.   : 0.000    Min.   :0.0000   Min.   :     0
 1st Qu.: 0.0000   1st Qu.: 0.000    1st Qu.:0.0000   1st Qu.:     0
 Median : 0.0000   Median : 2.000    Median :0.0000   Median :  6900
 Mean   : 0.4118   Mean   : 2.138    Mean   :0.3054   Mean   : 10990
 3rd Qu.: 0.0000   3rd Qu.: 4.000    3rd Qu.:1.0000   3rd Qu.: 15800
 Max.   :25.0000   Max.   :15.000    Max.   :1.0000   Max.   :199300
 AMT_BR_CSH_WDL_DR   AMT_CHQ_DR        AMT_NET_DR       AMT_MOB_DR
 Min.   :     0    Min.   :      0   Min.   :     0   Min.   :     0
 1st Qu.:  2990    1st Qu.:      0   1st Qu.:     0   1st Qu.:     0
 Median :340150    Median :  23840   Median :     0   Median :     0
 Mean   :378475    Mean   : 124520   Mean   :237308   Mean   : 22425
 3rd Qu.:674675    3rd Qu.:  72470   3rd Qu.:473971   3rd Qu.:     0
 Max.   :999930    Max.   :4928640   Max.   :999854   Max.   :199667
   AMT_L_DR       FLG_HAS_ANY_CHGS AMT_OTH_BK_ATM_USG_CHGS
 Min.   :      0   Min.   :0.0000   Min.   :  0.000
 1st Qu.: 237936   1st Qu.:0.0000   1st Qu.:  0.000
 Median : 695115   Median :0.0000   Median :  0.000
 Mean   : 773717   Mean   :0.1106   Mean   :  1.099
 3rd Qu.:1078927   3rd Qu.:0.0000   3rd Qu.:  0.000
 Max.   :6514921   Max.   :1.0000   Max.   :250.000
 AMT_MIN_BAL_NMC_CHGS NO_OF_IW_CHQ_BNC_TXNS NO_OF_OW_CHQ_BNC_TXNS
 Min.   :  0.000      Min.   :0.00000       Min.   :0.0000
 1st Qu.:  0.000      1st Qu.:0.00000       1st Qu.:0.0000
 Median :  0.000      Median :0.00000       Median :0.0000
 Mean   :  1.292      Mean   :0.04275       Mean   :0.0444
 3rd Qu.:  0.000      3rd Qu.:0.00000       3rd Qu.:0.0000
 Max.   :170.000      Max.   :2.00000       Max.   :2.0000
 AVG_AMT_PER_ATM_TXN AVG_AMT_PER_CSH_WDL_TXN AVG_AMT_PER_CHQ_TXN
 Min.   :    0       Min.   :     0          Min.   :     0
 1st Qu.:    0       1st Qu.:  1266          1st Qu.:     0
 Median : 6000       Median :147095          Median :  8645
 Mean   : 7409       Mean   :242237          Mean   : 25093
```

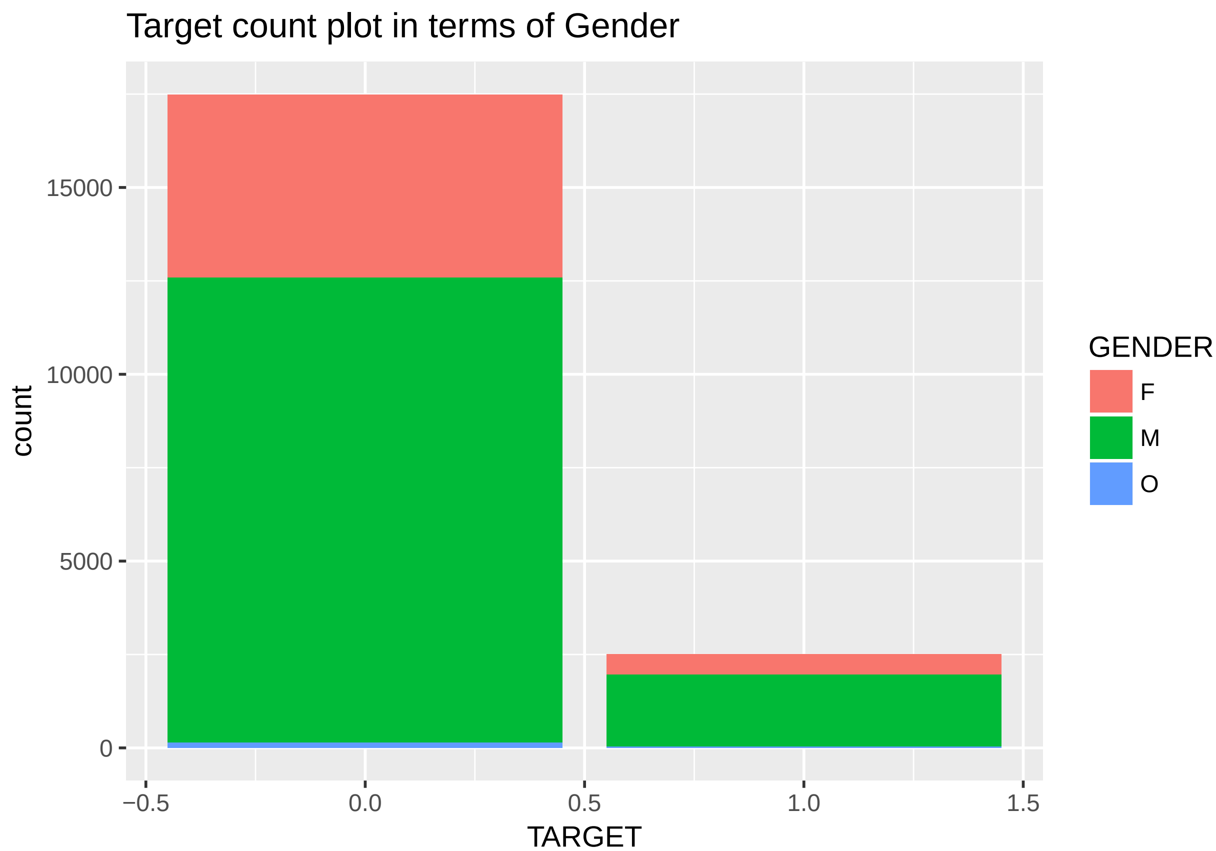
```
 3rd Qu.:13500       3rd Qu.:385000          3rd Qu.: 28605
 Max.   :25000       Max.   :999640          Max.   :537842
 AVG_AMT_PER_NET_TXN AVG_AMT_PER_MOB_TXN FLG_HAS_NOMINEE  FLG_HAS_OLD_LOAN
 Min.   :     0      Min.   :     0      Min.   :0.0000   Min.   :0.0000
 1st Qu.:     0      1st Qu.:     0      1st Qu.:1.0000   1st Qu.:0.0000
 Median :     0      Median :     0      Median :1.0000   Median :0.0000
 Mean   :179059      Mean   : 20304      Mean   :0.9012   Mean   :0.4929
 3rd Qu.:257699      3rd Qu.:     0      3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :999854      Max.   :199667      Max.   :1.0000   Max.   :1.0000
```

**Observation:**

- Summary of the data provides an easy way to get quick insights about the data, like the male observation is 3 times more than the females, the net transaction have higher initial amount value per transaction compared to other transaction as expected. The observations for savings account is more compared to current account. Based on thse the influence of the variables can be estimated while building the model, and help understand the results from the model with more clarity.

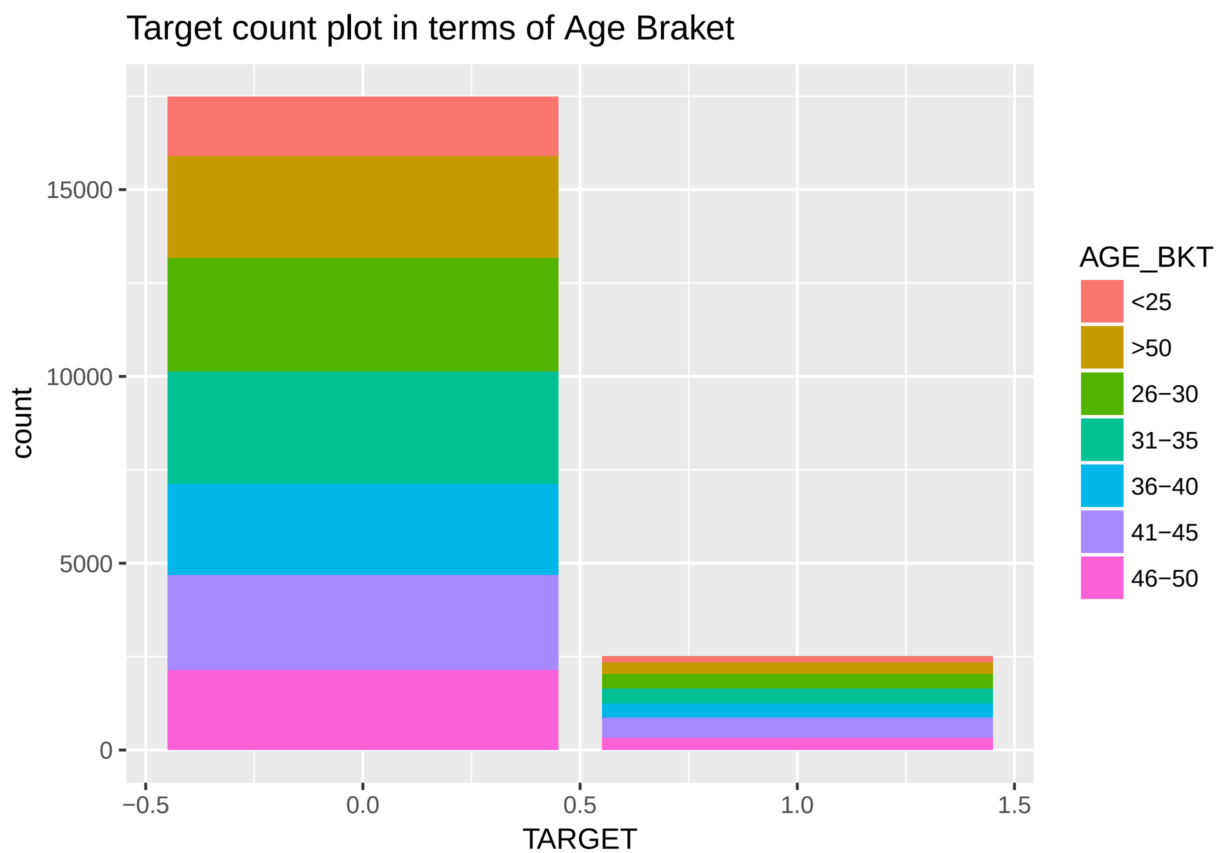
## 3. Exploratory Data Analysis

**Count plots to visualize ratio of responders for an attribute**



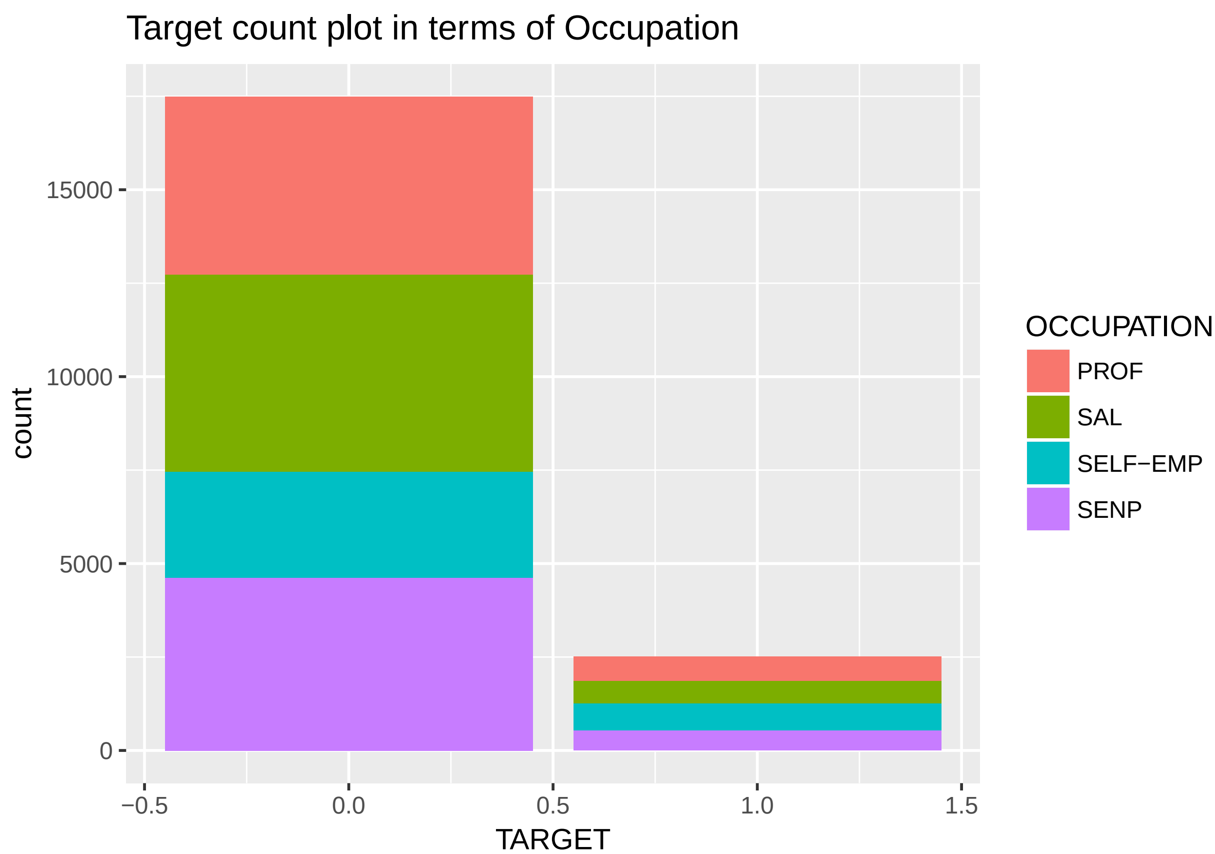

**Observation:**

- The representation for other category is relatively every less and observation for male forms the larger portion of the data.



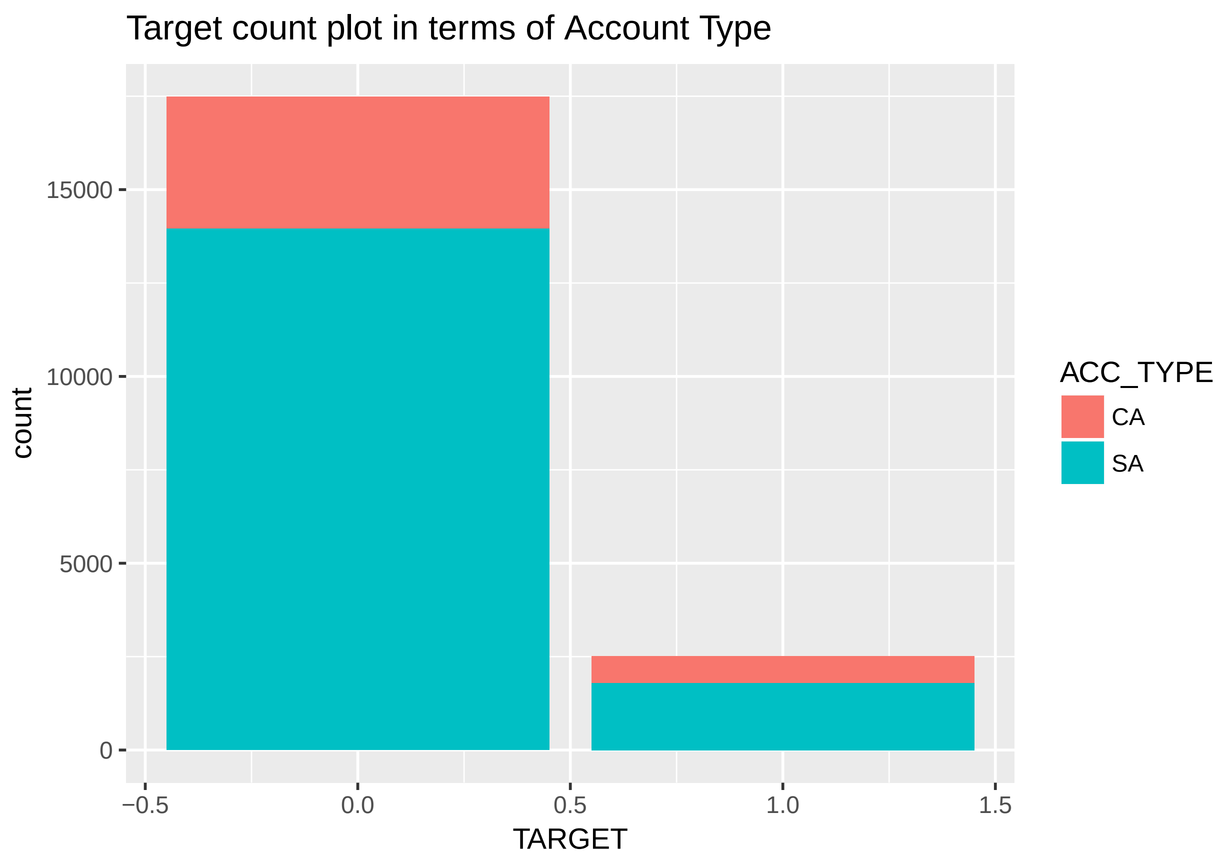

**Observation:**

- The ratio for age bracket in positive class and negative class are very similar which might not be a differentiator variable, if few age bracket had larger count in the positive class it would have been good to use that age bracket to have more influence, here there doesn't seem to be any particular difference in either of the classes. And given the dataset already has age variable, age braket may not add any more predictive power for the model.



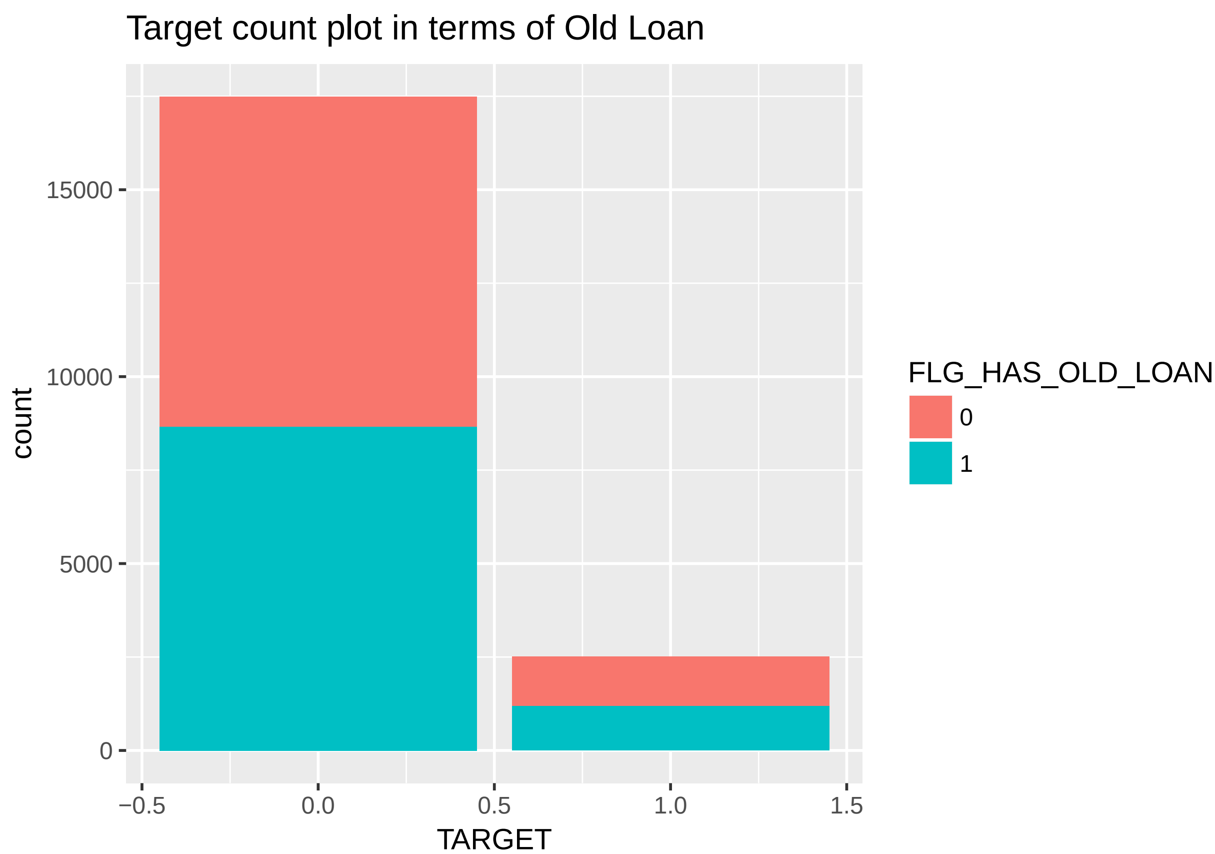

**Observation:**

- Occupation has a little difference between the positive and negative class, especially the self-employed has more count in the postive class, this would be a good predictor variable.



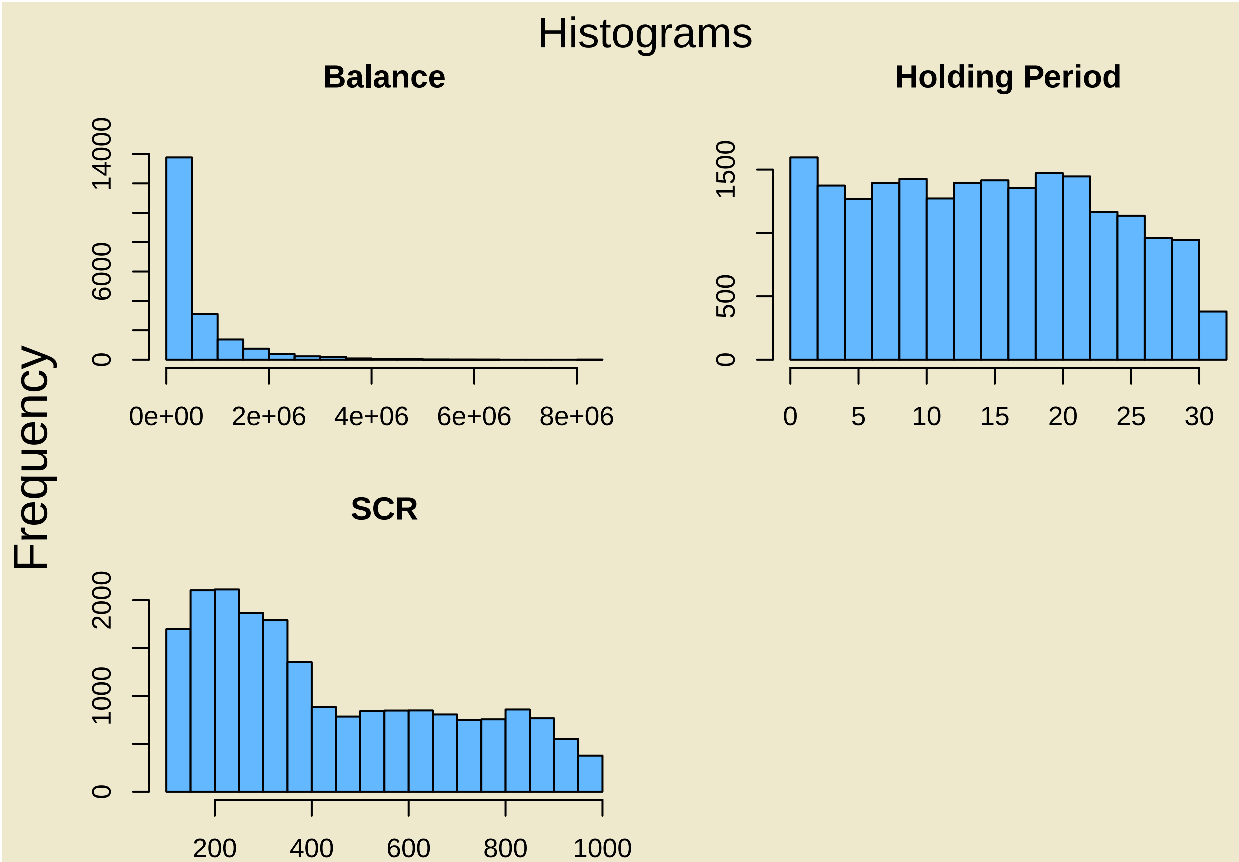

**Observation:**

- The count seems to have similar ratio between positive and negative class.




**Observation:**

- Since the intention to build models to predict who will respond postively old loan history will be a good predictor in conjuntion with SCR scores those with lower SCR scores and have not vailed loans can be excluded and those who have availed loand and have lower SCR scores should not be included to reduce the risk.




**Observation:**

- Holdin period has flat distribution, SCR has more towards lower half below 450 these would the ones which needs to excluded even though if the model returns a positive response in some way due to error. And the balance variables has more count indicating more accounts usually have less balance levels, and targeting those with some balance level would be less risky as there would likely be paying after availing the loand, and those with higher balance may not have a need for a loan so targeting them would be somewhat less fruitful.

## 3.1 Unbalanced data Analysis

**Unbalanced data:**

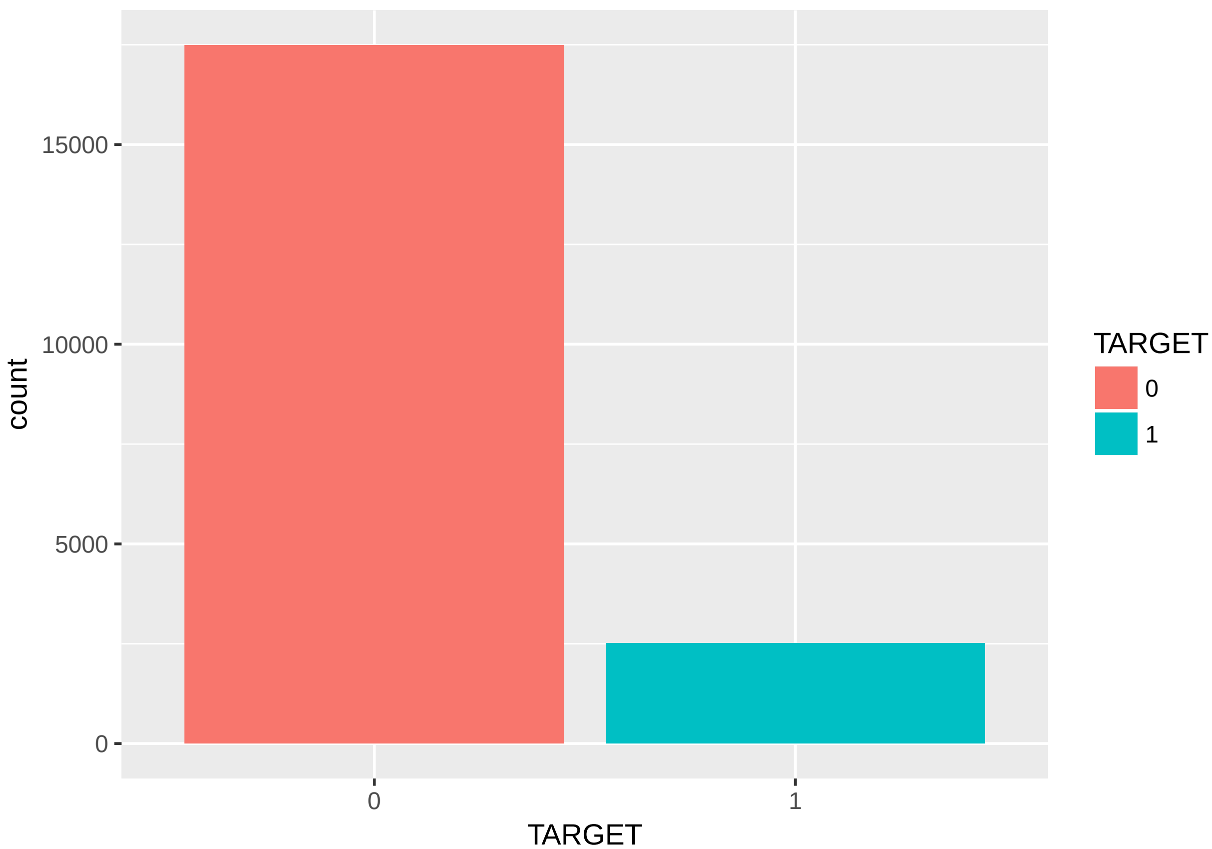
- The data with unbalanced observation would be less likely result in building a good model which can differentiate accurately between the positive and negative response class. In this regard sampling technique can be employed to make a data balanced.

```
# A tibble: 2 x 2
  TARGET class_count
   <int>       <int>
1      0       17488
2      1        2512
```

**Observation:**

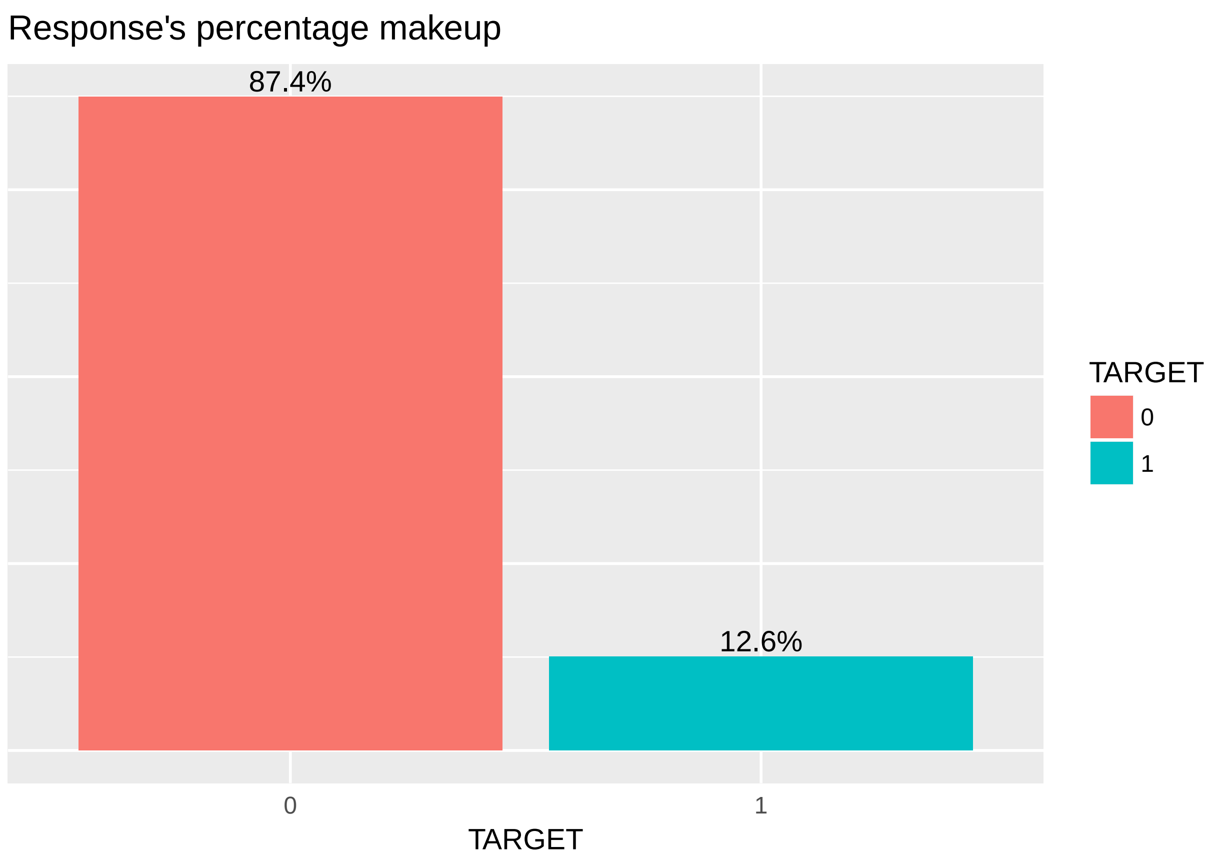
- The data has overall 20000 observation out of which 17488 are negative class and only about 12% of the observation belong to positive class which would result in a model which would not differentiate the positive class accurately.

**Count plot of response variable:**




**Observation:**

- The observation for negative class has a larger count.



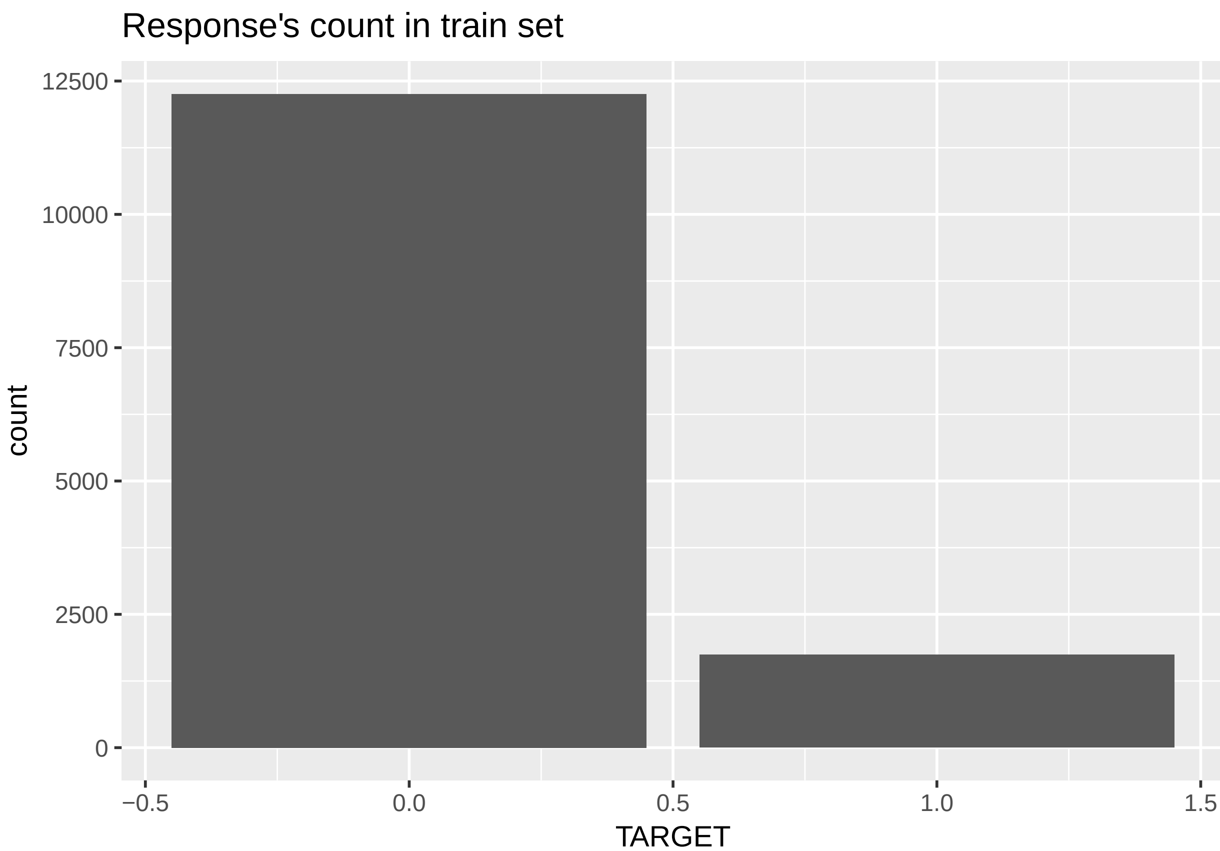

**Observation:**

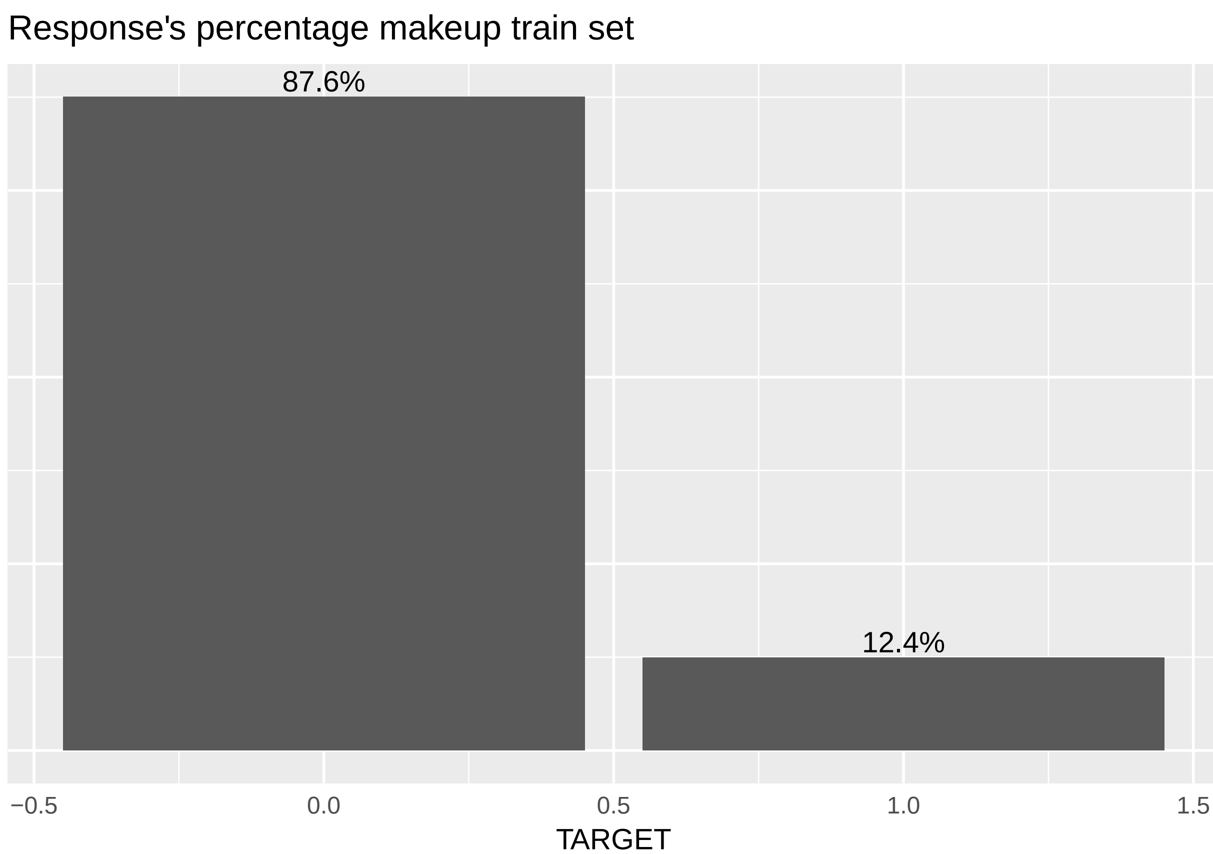
- The 12.6% of the postive class in the total observation will be problematic for building an accurate model.

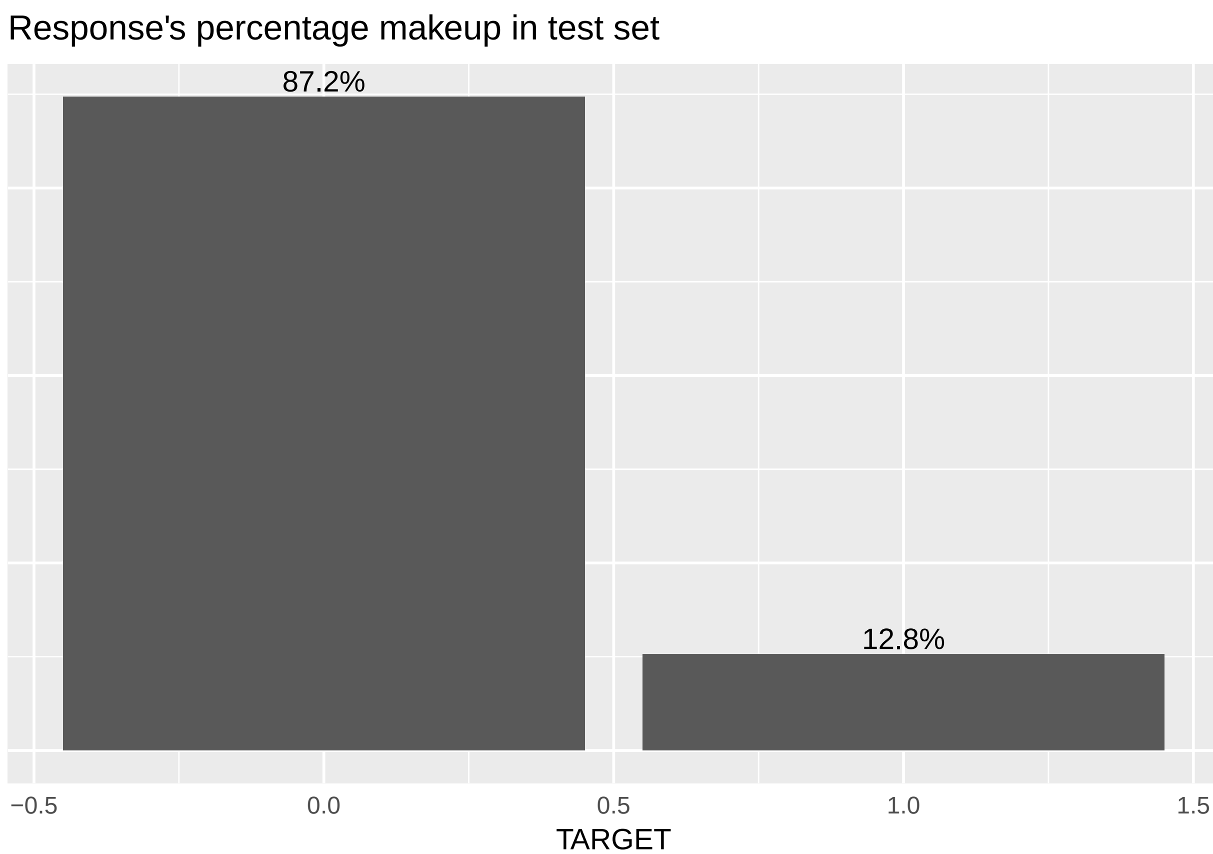
# 4. Splitting the data

## 4.1. Splitting the data into training and holdout sample

```
Dimension of total data:

20000 40

Dimension of training data:

14000 40

Dimension of test data:

6000 40
```




Response's percentage makeup train set
87.6%
12.4%
−0.5
0.0
0.5
1.0
1.5
TARGET




**Observation:**

- Using the train set which has unbalanced dataset would not give a good classification model, hence using sampling technique to transform the data into acceptable ratio of positive class to negative class would allow to build a better model.

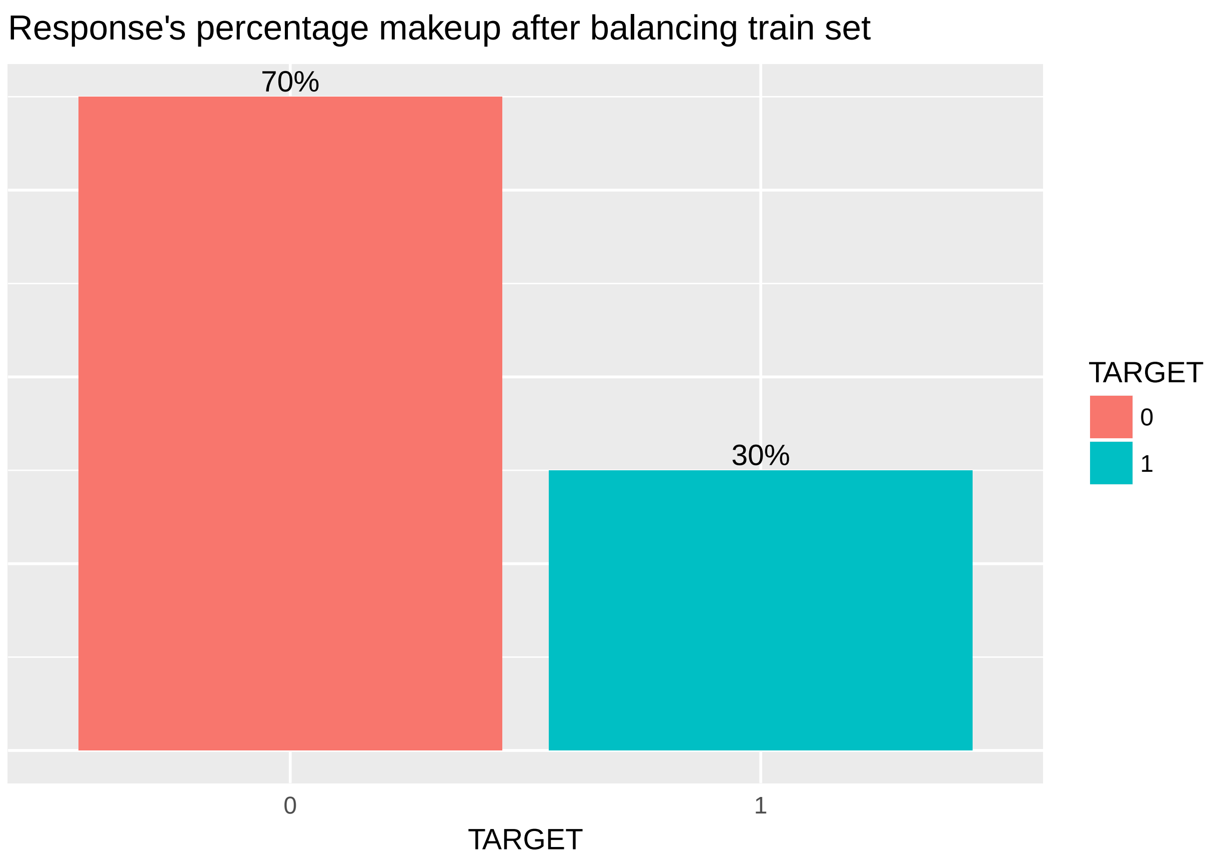
### 4.2 Generate Balanced data: Over sampling the positive class

```
cat('Response before balancing the Target class\n', table(train$TARGET))
```

```
Response before balancing the Target class
 12258 1742
```

```
train.balanced <- ovun.sample(TARGET~., data=train,
                              p=0.3,
                              seed=1, method="over")$data
```

```
Response after balancing the Target class
 12258 5253
```




**Observation:**

- The data has been transformed to make the positive(favourable) occur at 30% as previous 12.7% in the actual train dataset, now the balanced train dataset consist unfavourable:favourable response ratio to 70:30

# 5. CART Model

```r
r.ctrl = rpart.control(minsplit=100, minbucket = 12, cp = 0, xval = 10)


m1 <- rpart(formula = TARGET ~ .,
            data = train.balanced[,c(-1,-7,-11,-40)], method = "class",
            control = r.ctrl,
            parms = list(split = 'information'))
```

**Interpretation:**

- The minsplit has been set to perform split if the total observations are 100 or more and to the terminal node will have minimum of 12 observations.

## Analyzing the model

### Pruning Tree

```
Classification tree:
rpart(formula = TARGET ~ ., data = train.balanced[, c(-1, -7,
    -11, -40)], method = "class", parms = list(split = "information"),
    control = r.ctrl)

Variables actually used in tree construction:
 [1] AGE                     AMT_ATM_DR
 [3] AMT_BR_CSH_WDL_DR       AMT_CHQ_DR
 [5] AMT_L_DR                AMT_MOB_DR
 [7] AMT_NET_DR              AVG_AMT_PER_ATM_TXN
 [9] AVG_AMT_PER_CHQ_TXN     AVG_AMT_PER_CSH_WDL_TXN
[11] AVG_AMT_PER_MOB_TXN     AVG_AMT_PER_NET_TXN
[13] BALANCE                 FLG_HAS_CC
[15] FLG_HAS_OLD_LOAN        GENDER
[17] HOLDING_PERIOD          LEN_OF_RLTN_IN_MNTH
[19] NO_OF_BR_CSH_WDL_DR_TXNS NO_OF_CHQ_DR_TXNS
[21] NO_OF_L_CR_TXNS         NO_OF_L_DR_TXNS
[23] NO_OF_MOB_DR_TXNS       OCCUPATION
[25] SCR                     TOT_NO_OF_L_TXNS

Root node error: 5253/17511 = 0.29998

n= 17511

           CP nsplit rel error  xerror      xstd
1  0.01713307      0   1.00000 1.00000 0.0115438
2  0.01246907      2   0.96573 0.98744 0.0115019
3  0.00723396      6   0.90881 0.96288 0.0114173
4  0.00698014      8   0.89435 0.92043 0.0112623
5  0.00571102     11   0.87341 0.90253 0.0111935
6  0.00539374     12   0.86769 0.89720 0.0111727
7  0.00479091     17   0.84009 0.88864 0.0111387
```

```
8  0.00418808     25   0.80069 0.87417 0.0110803
9  0.00399772     26   0.79650 0.85684 0.0110086
10 0.00390253     28   0.78850 0.85151 0.0109861
11 0.00361698     30   0.78070 0.84142 0.0109430
12 0.00342661     31   0.77708 0.82410 0.0108673
13 0.00295069     33   0.77023 0.81211 0.0108137
14 0.00276033     35   0.76433 0.79421 0.0107318
15 0.00266514     37   0.75880 0.77822 0.0106566
16 0.00256996     49   0.72397 0.77518 0.0106420
17 0.00247478     51   0.71883 0.76223 0.0105795
18 0.00237959     52   0.71635 0.74110 0.0104746
19 0.00231160     56   0.70607 0.73863 0.0104621
20 0.00228441     71   0.66476 0.73310 0.0104340
21 0.00218923     73   0.66019 0.72701 0.0104027
22 0.00209404     76   0.65163 0.71692 0.0103502
23 0.00199886     80   0.64135 0.70798 0.0103030
24 0.00196713     82   0.63735 0.70588 0.0102919
25 0.00190367     95   0.60651 0.69941 0.0102572
26 0.00180849     99   0.59794 0.69237 0.0102190
27 0.00178945    103   0.59052 0.68532 0.0101805
28 0.00171331    108   0.58157 0.68418 0.0101742
29 0.00164985    128   0.52960 0.67447 0.0101202
30 0.00152294    131   0.52465 0.66419 0.0100622
31 0.00138016    135   0.51856 0.65524 0.0100109
32 0.00123739    142   0.50809 0.64249 0.0099366
33 0.00122379    145   0.50390 0.63449 0.0098892
34 0.00114220    154   0.49267 0.63050 0.0098653
35 0.00109461    157   0.48924 0.62041 0.0098043
36 0.00104702    161   0.48487 0.61946 0.0097985
37 0.00095184    164   0.48163 0.61260 0.0097565
38 0.00088045    168   0.47782 0.60784 0.0097270
39 0.00080000    183   0.46278 0.60270 0.0096949
```

**Interpretation:**

- The cp value to prune the tree was taken as 0.0008 after which reduction in the validation error is small, hence performing prune at 0.0008 was considered good.

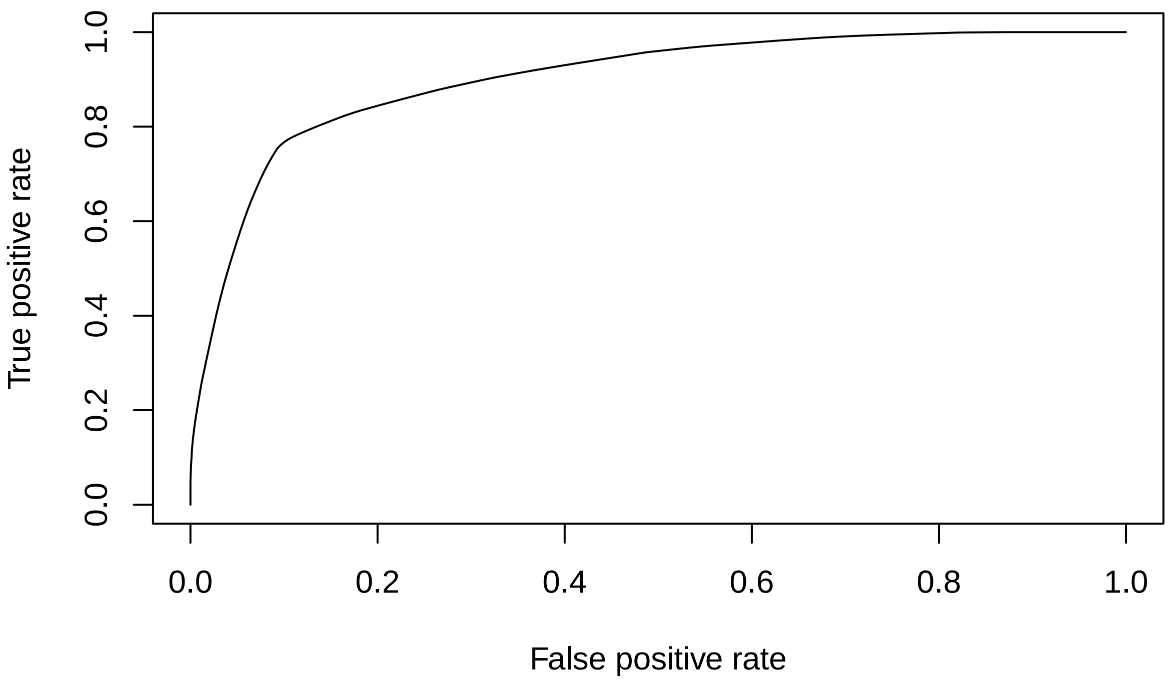
## 5.1 Scoring the classification tree

```
   deciles  cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 1819     1610          209 88.5%     1610          209
2:       9 1699     1304          395 76.8%     2914          604
3:       8 1753     1121          632 64.0%     4035         1236
4:       7 1930      458         1472 23.7%     4493         2708
5:       6 1585      263         1322 16.6%     4756         4030
6:       5 2173      268         1905 12.3%     5024         5935
   cum_rel_resp cum_rel_non_resp    ks
1:        30.6%             1.7% 28.94
2:        55.5%             4.9% 50.54
3:        76.8%            10.1% 66.73
4:        85.5%            22.1% 63.44
5:        90.5%            32.9% 57.66
```

```
6:        95.6%             48.4% 47.22
```

**Interpretation:**

- The KS value of 66.73 is achieved from the cross validation score, a KS of above 40 is considered to be good and here 66.73 is very encouraging.




```
      predict.class
TARGET     0     1
     0 11116  1142
     1  1289  3964


AUC:  0.9010003
KS:  0.6687697
Gini:  0.5614142
```

**Interpretation:**

- The tree measures are closely positive with KS of 0.67 and Gini of 0.56

## 5.2 Scoring Holdout sample

```
   deciles cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 610      278          332 45.6%      278          332
2:       9 615      188          427 30.6%      466          759
```

```
3:       8 580       58           522 10.0%      524          1281
4:       7 603       63           540 10.4%      587          1821
5:       6 599       51           548  8.5%      638          2369
6:       5 932       57           875  6.1%      695          3244
   cum_rel_resp cum_rel_non_resp    ks
1:        36.1%             6.4% 29.75
2:        60.5%            14.5% 46.01
3:        68.0%            24.5% 43.56
4:        76.2%            34.8% 41.41
5:        82.9%            45.3% 37.56
6:        90.3%            62.0% 28.23
```

```
      predict.class
TARGET    0    1
     0 4605  625
     1  346  424


AUC:  0.7813021
KS:  0.4600705
Gini:  0.6079377
```

**Interpretation:**

- The metrics of the holdout sample is above the general accepted values KS of 0.46 and Gini of 0.61 which is above the accepted 0.6 value.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4605  625
         1  346  424

               Accuracy : 0.8382
                 95% CI : (0.8286, 0.8474)
    No Information Rate : 0.8252
    P-Value [Acc > NIR] : 0.003943

                  Kappa : 0.3735
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9301
            Specificity : 0.4042
         Pos Pred Value : 0.8805
         Neg Pred Value : 0.5506
             Prevalence : 0.8252
         Detection Rate : 0.7675
   Detection Prevalence : 0.8717
      Balanced Accuracy : 0.6672

       'Positive' Class : 0
```

## Conclusion to CART model:

- The accuracy of 0.84 on the holdout sample is good and is encouraging. Given these measures CART model can be used make predictions for the unseen data. But there is still room for improvement and other model performance on the given data should be looked into to decide best model for making the predictions.

# 6. Random Forest Model

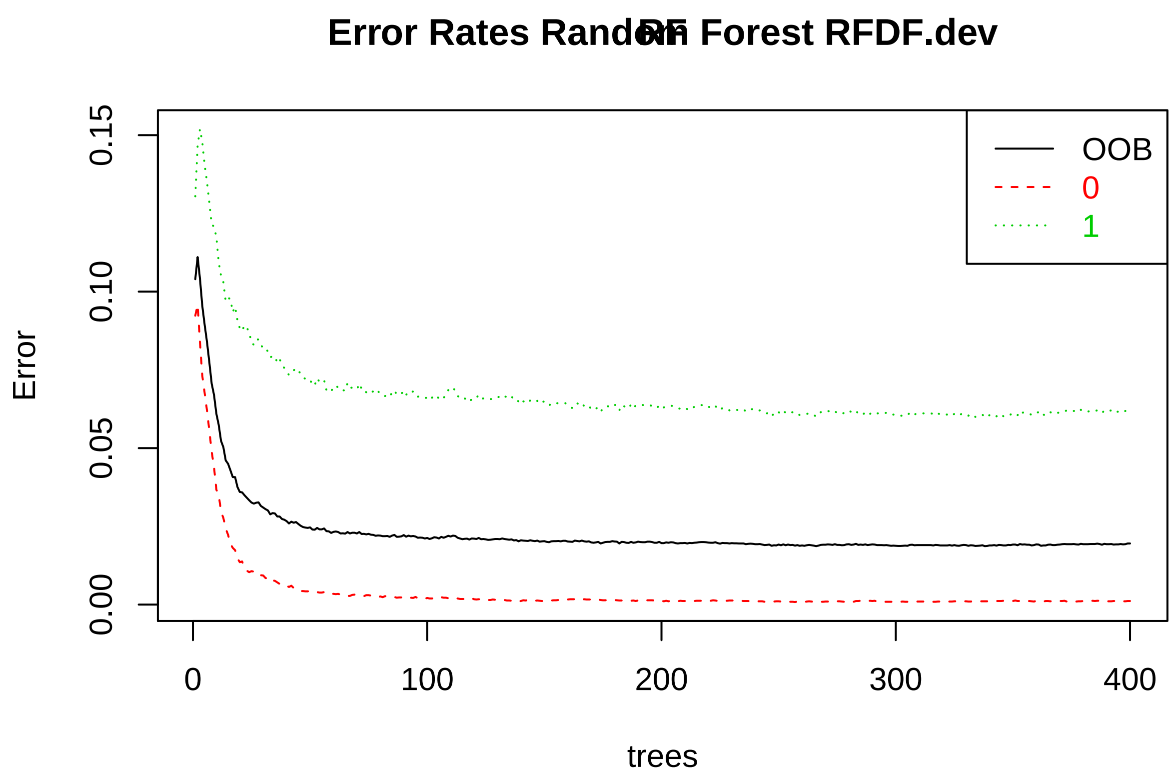
```
 [1] "CUST_ID"                 "TARGET"
 [3] "AGE"                     "GENDER"
 [5] "BALANCE"                 "OCCUPATION"
 [7] "AGE_BKT"                 "SCR"
 [9] "HOLDING_PERIOD"          "ACC_TYPE"
[11] "ACC_OP_DATE"             "LEN_OF_RLTN_IN_MNTH"
[13] "NO_OF_L_CR_TXNS"         "NO_OF_L_DR_TXNS"
[15] "TOT_NO_OF_L_TXNS"        "NO_OF_BR_CSH_WDL_DR_TXNS"
[17] "NO_OF_ATM_DR_TXNS"       "NO_OF_NET_DR_TXNS"
[19] "NO_OF_MOB_DR_TXNS"       "NO_OF_CHQ_DR_TXNS"
[21] "FLG_HAS_CC"              "AMT_ATM_DR"
[23] "AMT_BR_CSH_WDL_DR"       "AMT_CHQ_DR"
[25] "AMT_NET_DR"              "AMT_MOB_DR"
[27] "AMT_L_DR"                "FLG_HAS_ANY_CHGS"
[29] "AMT_OTH_BK_ATM_USG_CHGS" "AMT_MIN_BAL_NMC_CHGS"
[31] "NO_OF_IW_CHQ_BNC_TXNS"   "NO_OF_OW_CHQ_BNC_TXNS"
[33] "AVG_AMT_PER_ATM_TXN"     "AVG_AMT_PER_CSH_WDL_TXN"
[35] "AVG_AMT_PER_CHQ_TXN"     "AVG_AMT_PER_NET_TXN"
[37] "AVG_AMT_PER_MOB_TXN"     "FLG_HAS_NOMINEE"
[39] "FLG_HAS_OLD_LOAN"        "random"
```

```
Warning: package 'randomForest' was built under R version 3.4.4

randomForest 4.6-14

Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'

The following object is masked from 'package:dplyr':

    combine

The following object is masked from 'package:rattle':

    importance

The following object is masked from 'package:ggplot2':

    margin
```

## Training the Random Forest model

```

Call:
 randomForest(formula = as.factor(TARGET) ~ ., data = rf.data[,       c(-1, -11, -40)], ntree = 400, mtry
               Type of random forest: classification
                     Number of trees: 400
No. of variables tried at each split: 4

        OOB estimate of  error rate: 1.95%
Confusion matrix:
      0    1 class.error
```

```
0 12244    14 0.001142111
1   328 4925 0.062440510
```




**Interpretaion:**

- The generated model has good performance, with OOB error rate of 1.95%, however the peformance on the holdout sample be considered to check whether the model has overfit.

## List the importance of the variables

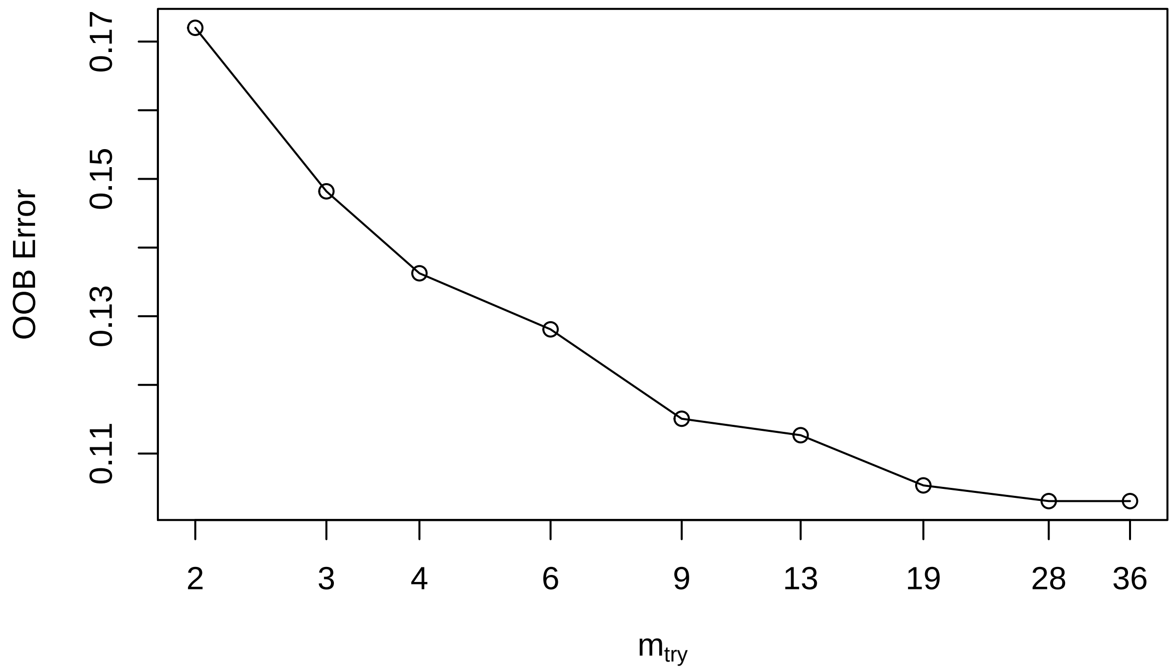
```
                             0     1 MeanDecreaseAccuracy MeanDecreaseGini
AGE_BKT                  48.65 57.48                59.07           318.66
AVG_AMT_PER_CSH_WDL_TXN  32.52 60.48                58.08           273.66
LEN_OF_RLTN_IN_MNTH      38.66 54.57                53.61           336.81
AMT_BR_CSH_WDL_DR        32.99 52.41                53.35           279.87
OCCUPATION               46.28 51.78                52.71           242.28
BALANCE                  39.95 52.96                52.03           410.66
AMT_L_DR                 37.01 46.66                50.45           341.68
AGE                      34.91 48.48                48.54           273.08
SCR                      40.50 48.29                48.10           405.95
HOLDING_PERIOD           31.81 47.11                46.01           387.62
AVG_AMT_PER_ATM_TXN      22.54 42.61                44.03           242.71
AMT_ATM_DR               25.27 38.39                41.27           250.14
NO_OF_L_CR_TXNS          31.00 38.36                40.81           300.09
TOT_NO_OF_L_TXNS         30.03 37.66                39.99           302.88
```

| | | | | |
|---|---|---|---|---|
| FLG_HAS_CC | 33.12 | 35.81 | 35.97 | 140.63 |
| AMT_CHQ_DR | 26.63 | 34.07 | 35.71 | 228.91 |
| AVG_AMT_PER_NET_TXN | 23.08 | 32.63 | 33.17 | 183.32 |
| FLG_HAS_OLD_LOAN | 22.08 | 33.77 | 33.12 | 58.66 |
| AVG_AMT_PER_CHQ_TXN | 25.12 | 31.73 | 32.43 | 223.68 |
| FLG_HAS_NOMINEE | 14.57 | 30.65 | 30.61 | 31.24 |
| AMT_NET_DR | 22.16 | 29.66 | 30.48 | 186.99 |
| NO_OF_BR_CSH_WDL_DR_TXNS | 20.39 | 28.99 | 28.66 | 127.58 |
| GENDER | 22.42 | 26.59 | 26.78 | 67.21 |
| NO_OF_L_DR_TXNS | 19.57 | 27.73 | 26.46 | 188.15 |
| NO_OF_CHQ_DR_TXNS | 19.80 | 24.19 | 24.89 | 119.98 |
| NO_OF_IW_CHQ_BNC_TXNS | 11.85 | 23.31 | 24.32 | 20.23 |
| AVG_AMT_PER_MOB_TXN | 14.11 | 24.34 | 23.75 | 101.61 |
| AMT_MOB_DR | 15.77 | 22.43 | 23.06 | 104.67 |
| NO_OF_OW_CHQ_BNC_TXNS | 11.48 | 21.22 | 22.15 | 20.43 |
| FLG_HAS_ANY_CHGS | 17.36 | 22.01 | 22.13 | 39.09 |
| ACC_TYPE | 16.53 | 16.27 | 17.81 | 42.59 |
| NO_OF_NET_DR_TXNS | 11.94 | 17.43 | 17.08 | 61.73 |
| NO_OF_ATM_DR_TXNS | 14.73 | 16.98 | 16.90 | 74.10 |
| AMT_MIN_BAL_NMC_CHGS | 5.30 | 10.17 | 11.05 | 3.78 |
| NO_OF_MOB_DR_TXNS | 6.84 | 9.14 | 8.98 | 24.55 |
| AMT_OTH_BK_ATM_USG_CHGS | 4.89 | 5.68 | 6.74 | 1.97 |

**Interpretation:**

- The top three important variables for the model is the Length of relationship with the bank, cash withdrawal amount and age. These three variables gives important information about loyalty of a customer, the financial status based on level of balance a customer hold in the account and the age the active repayment years or working years still left for a customer.

```
mtry = 3  OOB error = 14.82%
Searching left ...
mtry = 2    OOB error = 17.2%
-0.1606936 0.001
Searching right ...
mtry = 4    OOB error = 13.63%
0.0805395 0.001
mtry = 6    OOB error = 12.81%
0.05993294 0.001
mtry = 9    OOB error = 11.51%
0.1016496 0.001
mtry = 13   OOB error = 11.27%
0.02084367 0.001
mtry = 19   OOB error = 10.54%
0.06487582 0.001
mtry = 28   OOB error = 10.31%
0.02168022 0.001
mtry = 36   OOB error = 10.31%
0 0.001
```




## 6.1 Measuring the model performance

```
  CUST_ID TARGET AGE GENDER   BALANCE OCCUPATION AGE_BKT SCR
1  C14034      1  38      M 124050.44       SENP   36-40 758
2   C3658      0  43      M 385208.13       PROF   41-45 745
3  C12690      1  26      M 216951.51        SAL   26-30 486
4  C19526      0  35      M 179917.28        SAL   31-35 238
5  C13021      1  44      M   7701.45       PROF   41-45 134
6  C13042      1  54      M  40223.52   SELF-EMP     >50 807
  HOLDING_PERIOD ACC_TYPE ACC_OP_DATE LEN_OF_RLTN_IN_MNTH NO_OF_L_CR_TXNS
1              5       SA    02-11-01                 196              11
2             25       SA    01-09-13                  53               6
3             20       SA  11/17/2008                 103              11
4              8       SA   3/19/2008                 110               4
5             13       CA   8/22/2002                 177              38
6             22       CA    12-08-04                 150              11
  NO_OF_L_DR_TXNS TOT_NO_OF_L_TXNS NO_OF_BR_CSH_WDL_DR_TXNS
1               6               17                        1
2               1                7                        0
3               7               18                        3
4               6               10                        2
5              15               53                        1
6               4               15                        0
  NO_OF_ATM_DR_TXNS NO_OF_NET_DR_TXNS NO_OF_MOB_DR_TXNS NO_OF_CHQ_DR_TXNS
1                 1                 0                 0                 4
```

```
2                 0                 0                 0                 1
3                 1                 1                 0                 2
4                 1                 1                 0                 2
5                 2                 4                 1                 7
6                 2                 1                 0                 1
  FLG_HAS_CC AMT_ATM_DR AMT_BR_CSH_WDL_DR AMT_CHQ_DR AMT_NET_DR AMT_MOB_DR
1          0      11600            981870      77380          0          0
2          1          0                 0      46220          0          0
3          0      14500            145270      43880     225458          0
4          0      19500            151650      41410     268613          0
5          1      47700            822300      91080     970687       3386
6          0      35400                 0      43180     635414          0
  AMT_L_DR FLG_HAS_ANY_CHGS AMT_OTH_BK_ATM_USG_CHGS AMT_MIN_BAL_NMC_CHGS
1  1070850                0                       0                    0
2    46220                0                       0                    0
3   429108                0                       0                    0
4   481173                0                       0                    0
5  1935153                0                       0                    0
6   713994                0                       0                    0
  NO_OF_IW_CHQ_BNC_TXNS NO_OF_OW_CHQ_BNC_TXNS AVG_AMT_PER_ATM_TXN
1                     0                     0               11600
2                     0                     0                   0
3                     0                     0               14500
4                     0                     0               19500
5                     0                     0               23850
6                     0                     0               17700
  AVG_AMT_PER_CSH_WDL_TXN AVG_AMT_PER_CHQ_TXN AVG_AMT_PER_NET_TXN
1               981870.00            19345.00                 0.0
2                    0.00            46220.00                 0.0
3                48423.33            21940.00            225458.0
4                75825.00            20705.00            268613.0
5               822300.00            13011.43            242671.8
6                    0.00            43180.00            635414.0
  AVG_AMT_PER_MOB_TXN FLG_HAS_NOMINEE FLG_HAS_OLD_LOAN    random
1                   0               1                0 0.4716295
2                   0               1                0 0.2404372
3                   0               1                0 0.3703603
4                   0               0                1 0.8377543
5                3386               1                1 0.8577152
6                   0               1                1 0.9167304
  predict.class predict.score.0 predict.score.1
1             1           0.034           0.966
2             0           0.996           0.004
3             0           0.612           0.388
4             0           0.906           0.094
5             1           0.098           0.902
6             0           0.512           0.488
```

```
   deciles  cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 1756     1755            1  100%     1755            1
2:       9 1764     1731           33   98%     3486           34
3:       8 1739     1300          439   75%     4786          473
4:       7 1765      321         1444   18%     5107         1917
5:       6 1739       75         1664    4%     5182         3581
```

```
6:       5 1754       43         1711    2%     5225          5292
   cum_rel_resp cum_rel_non_resp   ks
1:          33%               0% 0.33
2:          66%               0% 0.66
3:          91%               4% 0.87
4:          97%              16% 0.81
5:          99%              29% 0.70
6:          99%              43% 0.56
```

```
KS:  0.8819228
```

```
AUC:  0.9833527
```

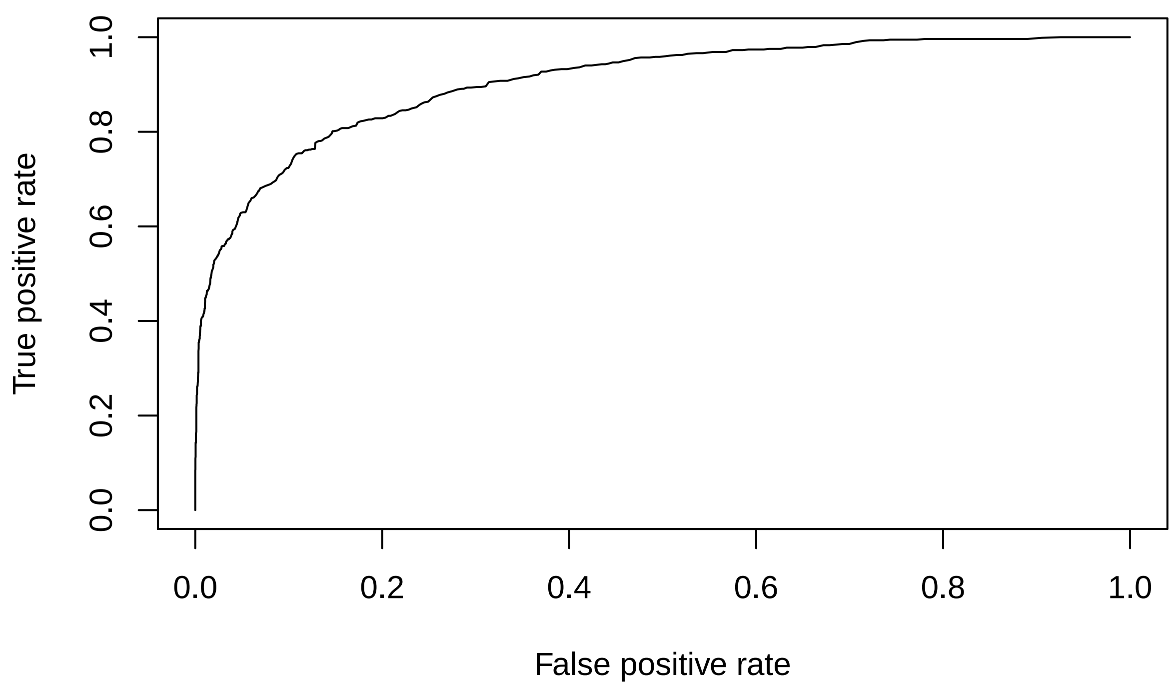
```
Gini: 0.5537327
```

```
      predict.class
TARGET     0     1
     0 12149   109
     1  1177  4076
```

**Interpretation:**

- A KS of 0.87 looks susceptible and whether the model has overfit needs to be looked based on the performance on the holdout sample, the Gini of 0.55 is below the general accepted level of 0.6

## 6.2 Scoring Holdout sample



```
AUC:  0.9052622

KS:  0.6544536

Gini: 0.5234223

      predict.class
TARGET    0    1
     0 5145   85
     1  393  377
```

**Interpretation:**

- The metrics on the holdout sample comes close to the general achievable level given the data set of 0.64, but the Gini of 0.51 is still below the 0.6 level

### Holdout Sample Confusion matrix

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5145   85
         1  393  377

               Accuracy : 0.9203
                 95% CI : (0.9132, 0.9271)
    No Information Rate : 0.923
    P-Value [Acc > NIR] : 0.7886

                  Kappa : 0.5707
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9290
            Specificity : 0.8160
         Pos Pred Value : 0.9837
         Neg Pred Value : 0.4896
             Prevalence : 0.9230
         Detection Rate : 0.8575
   Detection Prevalence : 0.8717
      Balanced Accuracy : 0.8725

       'Positive' Class : 0
```

**Interpretation:**

- The model accuracy on the holdout sample 0f 92% is very good given the model was able to ahieve is level of accuracy on the unseen dataset is encouraging, still as the data was unbalanced better data with resonable proportion between positive and negative class will reduce false negatives and more importantly false postives.

## Conclusion to Random Forest Model

- The random forest model accuracy of 92% is 10% more than the CART model, which makes random forest a better model for this dataset, however alternative models performance can be considered before deciding and even better a ensemble of models can constructed to measure the peformance of all the models.

# 7. Neural Network Model

## 7.1 Neural Network model variables inclusion

```
 [1] "TARGET"                    "AGE"
 [3] "GENDERF"                   "GENDERM"
 [5] "GENDERO"                   "BALANCE"
 [7] "OCCUPATIONPROF"            "OCCUPATIONSAL"
 [9] "OCCUPATIONSELF_EMP"        "OCCUPATIONSENP"
[11] "SCR"                       "HOLDING_PERIOD"
[13] "ACC_TYPECA"                "ACC_TYPESA"
[15] "LEN_OF_RLTN_IN_MNTH"       "NO_OF_L_CR_TXNS"
[17] "NO_OF_L_DR_TXNS"           "TOT_NO_OF_L_TXNS"
[19] "NO_OF_BR_CSH_WDL_DR_TXNS" "NO_OF_ATM_DR_TXNS"
[21] "NO_OF_NET_DR_TXNS"         "NO_OF_MOB_DR_TXNS"
[23] "NO_OF_CHQ_DR_TXNS"         "FLG_HAS_CC"
[25] "AMT_ATM_DR"                "AMT_BR_CSH_WDL_DR"
[27] "AMT_CHQ_DR"                "AMT_NET_DR"
[29] "AMT_MOB_DR"                "AMT_L_DR"
[31] "FLG_HAS_ANY_CHGS"          "AMT_OTH_BK_ATM_USG_CHGS"
[33] "AMT_MIN_BAL_NMC_CHGS"      "NO_OF_IW_CHQ_BNC_TXNS"
[35] "NO_OF_OW_CHQ_BNC_TXNS"     "AVG_AMT_PER_ATM_TXN"
[37] "AVG_AMT_PER_CSH_WDL_TXN"   "AVG_AMT_PER_CHQ_TXN"
[39] "AVG_AMT_PER_NET_TXN"       "AVG_AMT_PER_MOB_TXN"
[41] "FLG_HAS_NOMINEE"           "FLG_HAS_OLD_LOAN"
```

**Interpretation:**

- The neural network model requires the variables to be numeric and converting the categorical variables to dummy variables makes it possible to include the categorical variables for training the model. The Age bracket variables has not been considered training the model and it constributes less towards the predictive power of the model given customer age is already present in the dataset.

## 7.2 Model Training using the scaled data

```
hidden: 3    thresh: 0.1    rep: 1/1    steps:      500  min thresh: 0.1385284426
                                                    593  error: 657.89316    time: 9.86 secs
```

**Interpretation:**

- The hidden layers of 3 has been considered and sum of squared error as error measure, the minimum decrease in the Gradient(slope) of 0.1 is taken.

## 7.3 Scoring the NN model

```
   deciles  cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 1402      560          842   40%      560          842
2:       9 1398      285         1113   20%      845         1955
3:       8 1400      267         1133   19%     1112         3088
4:       7 1400      140         1260   10%     1252         4348
5:       6 1401       93         1308    7%     1345         5656
```

```
6:       5 1400      125         1275    9%     1470         6931
   cum_rel_resp cum_rel_non_resp   ks
1:          32%               7% 0.25
2:          49%              16% 0.33
3:          64%              25% 0.39
4:          72%              35% 0.37
5:          77%              46% 0.31
6:          84%              57% 0.27
```

**Interpretation:**

- The KS of 0.39 is achieved which is close to the acceptable 0.4 level.

### Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 11967   291
         1  1367   375

               Accuracy : 0.8815714
                 95% CI : (0.876104, 0.8868798)
    No Information Rate : 0.9524286
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2605678
 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.8974801
            Specificity : 0.5630631
         Pos Pred Value : 0.9762604
         Neg Pred Value : 0.2152698
             Prevalence : 0.9524286
         Detection Rate : 0.8547857
   Detection Prevalence : 0.8755714
      Balanced Accuracy : 0.7302716

       'Positive' Class : 0
```

**Interpretation:**

- The accuracy for the model without using the over sampling method is 88%, with sensitivity(true positive rate) of 90% and specificity(true negative rate) 0.56

## 7.3 Scoring the holdout sample

```
           0%            1%            5%           10%           25% 
0.01234440023 0.01234440023 0.01234440056 0.01234470913 0.02924813912 
          50%           75%           90%           95%           98% 
```

```
0.05489879574 0.19518399737 0.22823218629 0.43935020199 0.55243616441
          99%          100%
0.55251567070 0.55251620087
```

```
   deciles cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 600      182          418   30%      182          418
2:       9 600      131          469   22%      313          887
3:       8 600      127          473   21%      440         1360
4:       7 600       80          520   13%      520         1880
5:       6 603       54          549    9%      574         2429
6:       5 597       56          541    9%      630         2970
   cum_rel_resp cum_rel_non_resp   ks
1:          24%               8% 0.16
2:          41%              17% 0.24
3:          57%              26% 0.31
4:          68%              36% 0.32
5:          75%              46% 0.29
6:          82%              57% 0.25
```

**Interpretation:**

- The KS of 0.32 is ahieved which lesser than the training data and CART and Random Forest models.

**Confusion Matrix for holdout sample**

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5068  162
         1  657  113

               Accuracy : 0.8635
                 95% CI : (0.8545529, 0.8720921)
    No Information Rate : 0.9541667
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1594971
 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.8852402
            Specificity : 0.4109091
         Pos Pred Value : 0.9690249
         Neg Pred Value : 0.1467532
             Prevalence : 0.9541667
         Detection Rate : 0.8446667
   Detection Prevalence : 0.8716667
      Balanced Accuracy : 0.6480746

       'Positive' Class : 0
```

**Interpretation:**

- The accuracy of 86% achieved doesn't predict the positive response accurately as the dataset has more non-reponders number the 86% just represents 5068 cases which were corretly classified as non-responders, and here again the unbalanced nature of the data is causing the model to have high accuracy but the actual performance doesn't look every encouraging which miss classifies a lot of the responders.

## 7.4 Training the model using balanced data(over sampling method)

```
hidden: 3    thresh: 0.1    rep: 1/1    steps:     500  min thresh: 0.3190523784
                                                  1000  min thresh: 0.1342862939
                                                  1426  error: 1464.37244   time: 27.56 secs
```

```
           0%            1%            5%           10%           25% 
0.05469241565 0.05469241565 0.05469241565 0.05469250058 0.15040667476 
          50%           75%           90%           95%           98% 
0.18107460842 0.42935028611 0.69741550681 0.69762789913 0.69762790614 
          99%          100% 
0.69762790614 0.69762790614 
```

```
   deciles  cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 1752     1186          566   68%     1186          566
2:       9 1752      989          763   56%     2175         1329
3:       8 1750      746         1004   43%     2921         2333
4:       7 1753      686         1067   39%     3607         3400
5:       6 1749      430         1319   25%     4037         4719
6:       5 1751      270         1481   15%     4307         6200
   cum_rel_resp cum_rel_non_resp   ks
1:          23%               5% 0.18
2:          41%              11% 0.30
3:          56%              19% 0.37
4:          69%              28% 0.41
5:          77%              38% 0.39
6:          82%              51% 0.31
```

#### Confusion matrix of over sampled data

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 11967   291
         1  1367   375
                                         
               Accuracy : 0.8815714      
                 95% CI : (0.876104, 0.8868798)
    No Information Rate : 0.9524286      
    P-Value [Acc > NIR] : 1              
                                         
                  Kappa : 0.2605678      
 Mcnemar's Test P-Value : <0.0000000000000002
                                         
            Sensitivity : 0.8974801      
            Specificity : 0.5630631      
         Pos Pred Value : 0.9762604      
```

```
         Neg Pred Value : 0.2152698
             Prevalence : 0.9524286
         Detection Rate : 0.8547857
   Detection Prevalence : 0.8755714
      Balanced Accuracy : 0.7302716

       'Positive' Class : 0
```

**Interpretation:**

- The trained model using the balanced data gives a greater error than the model trained using only the acutal train split data. And it can be looked as the duplicate information is not making the model to learning any better about the patterns in the data, it just creating more difficulties for the model to learn the actual pattern as the accuracy has not improved.

**Percetiles for the test set**

```
           0%            1%            5%           10%           25% 
0.05469241565 0.05469241565 0.05469241565 0.05469253867 0.15040667476 
          50%           75%           90%           95%           98% 
0.18105731539 0.40351747437 0.68266470774 0.69762786653 0.69762790614 
          99%          100% 
0.69762790614 0.69762790614 
```

```
   deciles cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
1:      10 600      220          380   37%      220          380
2:       9 600      101          499   17%      321          879
3:       8 600      101          499   17%      422         1378
4:       7 600      105          495   18%      527         1873
5:       6 600       54          546    9%      581         2419
6:       5 600       35          565    6%      616         2984
   cum_rel_resp cum_rel_non_resp   ks
1:          29%               7% 0.22
2:          42%              17% 0.25
3:          55%              26% 0.29
4:          68%              36% 0.32
5:          75%              46% 0.29
6:          80%              57% 0.23
```

**Confusion Matrix for the test set**

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4756  474
         1  531  239
                                          
               Accuracy : 0.8325          
                 95% CI : (0.8228069, 0.8418687)
    No Information Rate : 0.8811667       
    P-Value [Acc > NIR] : 1.0000000       
```

```
                 Kappa : 0.2269211
 Mcnemar's Test P-Value : 0.0773179

            Sensitivity : 0.8995650
            Specificity : 0.3352034
         Pos Pred Value : 0.9093690
         Neg Pred Value : 0.3103896
             Prevalence : 0.8811667
         Detection Rate : 0.7926667
   Detection Prevalence : 0.8716667
      Balanced Accuracy : 0.6173842

       'Positive' Class : 0
```

**Interpretation:**

- The Accuracy acieved using the over sampled data doesn't lead in a more accurate model, so trying principal component analysis to reduce the number of variables would be a good method given the neural network model performs better with less covaried data in the training data set.

## 7.5 Performing PCA for the training dataset

**Using the actual training set without over sampling**

**Bartlett test to check whether PCA is possible**

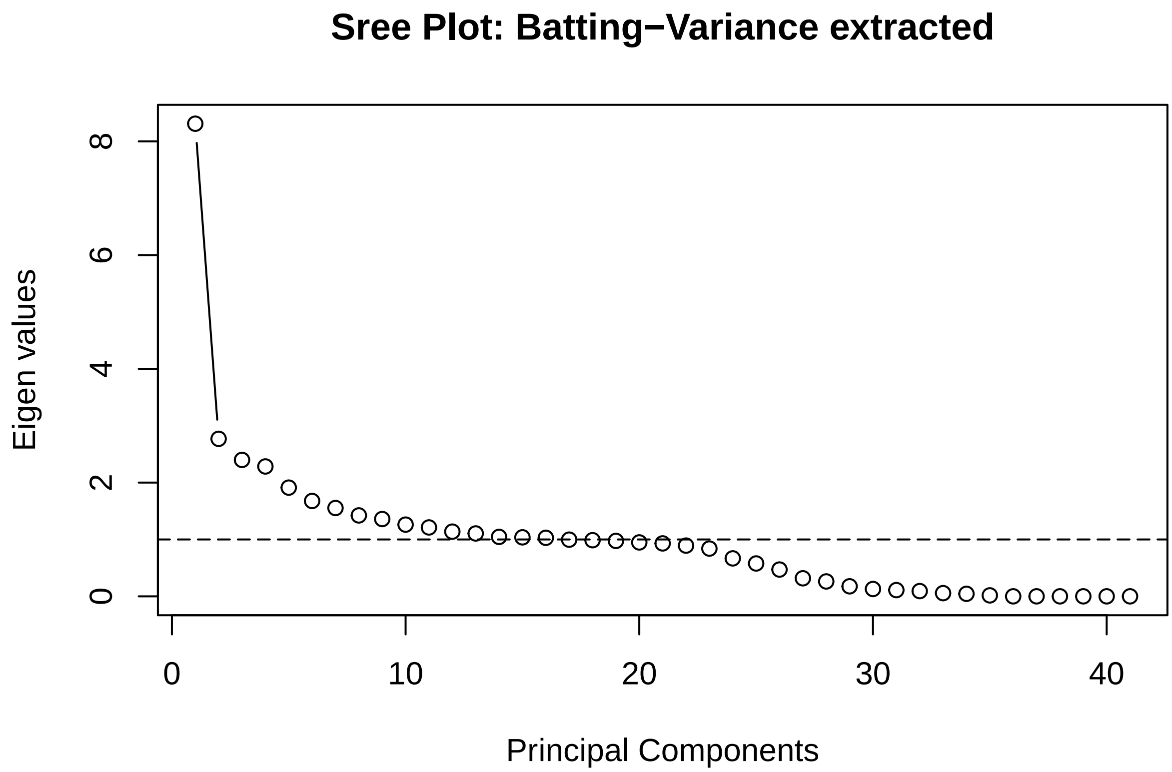
```
$chisq
[1] 3035312.651

$p.value
[1] 0

$df
[1] 820
```

**Interpretation:**

- The p-value from the test tells that the PCA is possible for the data.

**Sree Plot**

```
Warning in plot.xy(xy, type, ...): plot type 'both' will be truncated to
first character
```




**Interpretation:**

- The sree plot provides visual representation to choose the optimal number PCA components those which have eigen values of 1 or greater, and accordingly there are 14 PCA variables which can be considered good to include to train the nerual network model to check the impact on the classification power of the trained model using the PCA variables as against the actual and over sampled training data.

**Cummulative variance explained**

```
 [1]  20.27026329  27.02533951  32.87627535  38.44602097  43.10864026
 [6]  47.19765973  50.98481497  54.45431505  57.76916365  60.84476041
[11]  63.79904230  66.57380877  69.27002397  71.81984388  74.35049252
[16]  76.86064261  79.29154320  81.70014636  84.07725388  86.38898429
[21]  88.65832196  90.83411927  92.88181832  94.50960157  95.91862872
[26]  97.06746092  97.84051009  98.47865004  98.90831555  99.22249724
[31]  99.48988227  99.71440847  99.85119566  99.95915745  99.99995628
[36] 100.00000000 100.00000000 100.00000000 100.00000000 100.00000000
[41] 100.00000000
```

**Interpretation:**

- The fist 14 PCA explains the 70% variability in the data

```
TARGET              PC1              PC2             PC3             PC4
```

```
2275        0  0.5313581957 -0.88546064768  1.1776801490 -0.2492443157
12446       0  1.6407451352 -0.20275254223  0.4676613438  0.6825361193
12185       1 -2.1529218481  0.94787376086 -0.7318284763  1.0678629384
12466       0 -0.7265408539 -3.06609887493  0.3763737569  0.5757846760
17215       1 -1.0012763495  0.05819420764 -0.8243366533  0.2902222338
12804       1 -4.4005039758  1.32334652203 -2.5451495438  1.6104748770
```

## 7.6 Training the neural network model with PCA variables

```
hidden: 3    thresh: 0.1    rep: 1/1    steps:     1000  min thresh: 0.1296927106
                                                   1134  error: 691.22426    time: 11.69 secs
```

**Interpretation:**

- The error for the trained is slightly greater the neural network model trained with actural training set, so PCA is not making the model perform any better than. The error for the model on acutal train set is error value of 658 for this model it is 691 when the model converged, which would not result in acutual difference in classifiying the test set.
- Hence of the three neural network model trained used three training dataset: acutal training dataset, balanced training data set(using over sampling method) and training dataset after performing PCA. The neural network model created with actual training dataset is better and its results on test dataset will be considered to build the ensemble model.

# 8. Ensemble Model

## Ensemble advantages

- Ensemble learning helps improve machine learning results by combining several models. Ensemble methods combine several machine learning results(techniques) into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking). Here using three ensemble methods to improve classification of the test dataset: Majority voting, Weighted average, Averaging.
- The techinque of the using the results from the models as variables to train the new model on top of the bottom layer models is also widely employed which provides better performance as against the simpler three formula methods.

## 8.1 Majority voting

-

```
test$pred_majority<-as.factor(ifelse(rf.test$predict.class==1 & nn.test$Class==1,1,ifelse(rf.test$predi
```

## 8.2 Averaging

```
#Taking average of predictions
test$pred_avg<-(rf.test$predict.score[,2]+nn.test$Predict.score+cart.test$predict.score)/3

#Splitting into binary classes at 0.5
test$pred_avg_class<-as.factor(ifelse(test$pred_avg>0.5,1,0)[,2])
```

## 8.3 Weighted Average

```
#Taking weighted average of predictions
test$pred_weighted_avg<-(rf.test$predict.score[,2]*0.5)+(nn.test$Predict.score*0.2)+(cart.test$predict.s

#Splitting into binary classes at 0.5
test$pred_weighted_avg<-as.factor(ifelse(test$pred_weighted_avg>0.5,1,0)[,2])

head(test[,c(1, 43, 45, 46)], 10)
```

```
   TARGET pred_majority pred_avg_class pred_weighted_avg
1       0             0              0                 0
2       0             0              0                 0
3       0             0              0                 0
4       0             0              0                 0
6       0             0              0                 0
9       0             0              0                 0
10      0             0              0                 0
11      0             0              0                 0
14      0             0              0                 0
18      0             0              0                 0
```

## 8.4 Scoring the ensemble results

### Confusion matrix of Majority method

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5122  108
         1  446  324

               Accuracy : 0.9076667
                 95% CI : (0.9000586, 0.9148756)
    No Information Rate : 0.928
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4922648
 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.9198994
            Specificity : 0.7500000
         Pos Pred Value : 0.9793499
         Neg Pred Value : 0.4207792
             Prevalence : 0.9280000
         Detection Rate : 0.8536667
   Detection Prevalence : 0.8716667
      Balanced Accuracy : 0.8349497

       'Positive' Class : 0
```

**Interpretation:**

- The accuracy of the majority is better than the CART model and Neural Network holdout sample results. The Random forest model performs better than the majority method.

### Confusion matrix of Average method

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5140   90
         1  496  274

               Accuracy : 0.9023333
                 95% CI : (0.8945416, 0.9097313)
    No Information Rate : 0.9393333
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4368489
 Mcnemar's Test P-Value : <0.0000000000000002
```

```
            Sensitivity : 0.9119943
            Specificity : 0.7527473
         Pos Pred Value : 0.9827916
         Neg Pred Value : 0.3558442
             Prevalence : 0.9393333
         Detection Rate : 0.8566667
   Detection Prevalence : 0.8716667
      Balanced Accuracy : 0.8323708

       'Positive' Class : 0
```

#### Interpretation

- The average method also performs better than the CART and neural network model, but cannot outpeform the random forest model

#### Confusion matrix of Weighted Average method

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5131   99
         1  447  323

               Accuracy : 0.909
                 95% CI : (0.901439, 0.9161606)
    No Information Rate : 0.9296667
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4961643
 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.9198638
            Specificity : 0.7654028
         Pos Pred Value : 0.9810707
         Neg Pred Value : 0.4194805
             Prevalence : 0.9296667
         Detection Rate : 0.8551667
   Detection Prevalence : 0.8716667
      Balanced Accuracy : 0.8426333

       'Positive' Class : 0
```

#### Interpretation:

- Based on its individual perform of the Random Forest model the highest weight of 0.5 was assigned to it and 0.3 to CART model and 0.2 to neural network model. Which againg provides results better than the CART and Neural Network but below the Random Forest. Which may have pushed more towards the predictions influenced by Random Forest as it was a better model a higher weight was assigned to differentiate between the Majority and Average method results.

# 9. Project Conclusion

- The binary classification problem is very common in business problems and ability to analyse to provide best accurate insights to take informed decision allows the decision makers to adopt more confident actions based on the measures gained from the classification model results. So, it is very important to have the ability to build such a model which provides accurate predictions which will allow to reduce the uncertainty associated with data and not add to the uncontrolable variance which are not captured by the data.