# Group Assignment Report for

## "Data Mining"


**SUBMITTED TO: GREAT LAKES INSTIUTE OF MANAGEMENT**

*In partial fulfilment of the requirements for the award of degree of*

*PGP – Business Analytics and Business Intelligence*


| **Submitted by: Group 6** | **Student ID** |
|---|---|
| Rupak | |
| Thokchom Joychandra Singh | E8DLVOV0YE |
| Madhuchandan S | |
| Nishitha Ramesh | |
| Amulya KS | |


greatlearning
*Learning for Life*

**PGP-BABI 2019-20**

# 1. Thera Bank – Loan Purchase Modelling

**Problem Statement:**

This case is about a bank (Thera Bank) which has a growing customer base. Majority of these customers are liability customers (depositors) with varying size of deposits. The number of customers who are also borrowers (asset customers) is quite small, and the bank is interested in expanding this base rapidly to bring in more loan business and in the process, earn more through the interest on loans. In particular, the management wants to explore ways of converting its liability customers to personal loan customers (while retaining them as depositors). A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise campaigns with better target marketing to increase the success ratio with a minimal budget. The department wants to build a model that will help them identify the potential customers who have a higher probability of purchasing the loan. This will increase the success ratio while at the same time reduce the cost of the campaign. The dataset has data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign. Link to the case file:

Thera Bank-Data Set.xlsx

You are brought in as a consultant and your job is to build the best model which can classify the right customers who have a higher probability of purchasing the loan. You are expected to do the following:

● EDA of the data available. Showcase the results using appropriate graphs

● Build appropriate models on both the test and train data (CART, Random Forest). Interpret all the model outputs and do the necessary modifications wherever eligible (such as pruning)

● Check the performance of all the models that you have built (test and train). Use all the model performance measures you have learned so far. Share your remarks on which model performs the best.

**Solution:**
Below is the explanatory data analysis report of the cereal dataset.

**1.1 Basic Data Structure:**

The given Thera Bank data set has 5000 observations of 14 variables. Out of the total 5000 observations, there are 18 rows that are missing data and the remaining 4982 are completed rows.

Of all the 14 variables, all the variables are numeric in nature however, Personal Loan, CD Account, Securities Account, online and CreditCard have levels “0” and “1” which means that the data is categorical in nature. Therefore, it should be converted into Factors. Also, Education columns has 3 levels 1, 2 and 3 which is also categorical in nature and hence should be converted into Ordered Factors.

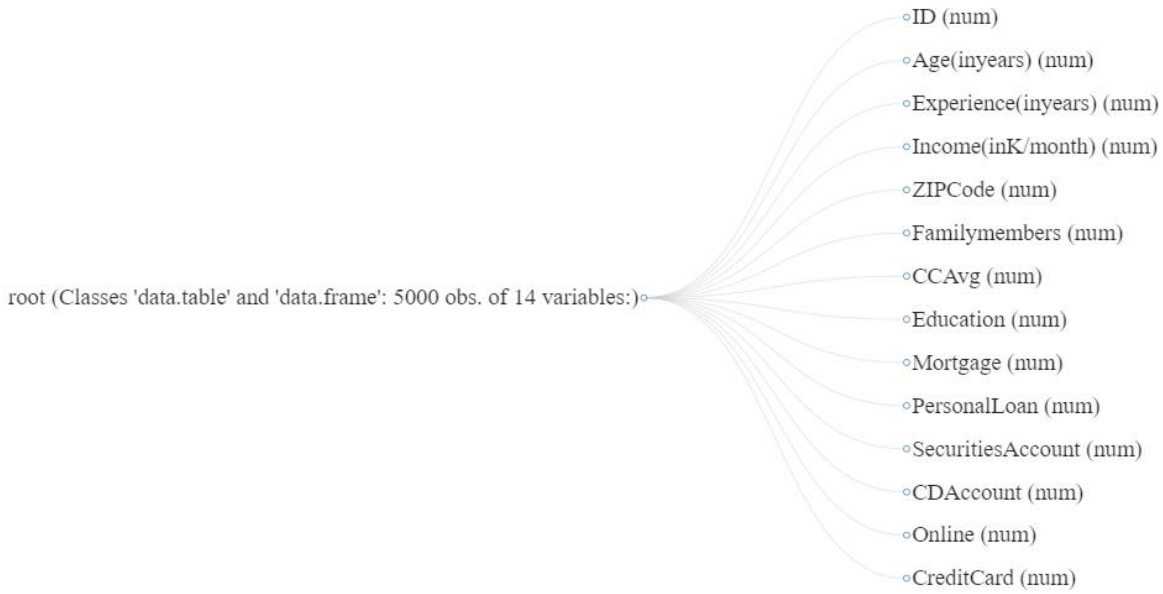



**Fig: Before Treatment**

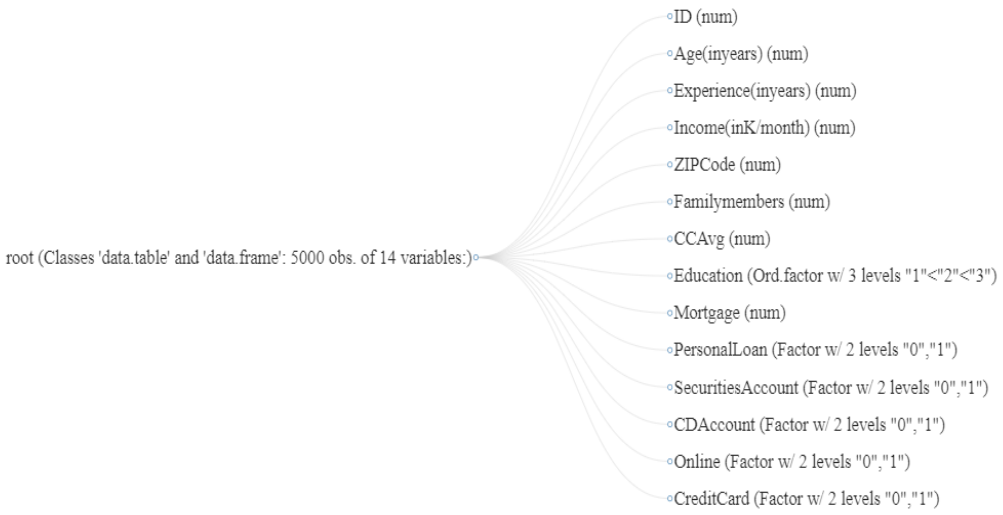



**Fig: After Treatment**

After treatment, we have 6 Discrete columns, Factors and the remaining data are continuous in nature. With this we are completed with the data formatting portion of the data. Now the next step will be the missing value treatment along with disturbance in data treatments.

**Null values and wrongly entered values.**

18 rows have "Family members" data missing in the provided data set. We will be treating this by replacing the missing value as "1" since the minimum number of person that user's family can have is at least the person i.e. himself/herself. And also, the mode of the overall data is 1 which justifies our point of view.

```
       ID        Age (in years)  Experience (in years) Income (in K/month)    ZIP Code     Family members      CCAvg
 Min.   :   1   Min.   :23.00   Min.   :-3.0          Min.   :  8.00       Min.   : 9307   Min.   :1.000   Min.   : 0.000
 1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0          1st Qu.: 39.00       1st Qu.:91911   1st Qu.:1.000   1st Qu.: 0.700
 Median :2500   Median :45.00   Median :20.0          Median : 64.00       Median :93437   Median :2.000   Median : 1.500
 Mean   :2500   Mean   :45.34   Mean   :20.1          Mean   : 73.77       Mean   :93153   Mean   :2.397   Mean   : 1.938
 3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0          3rd Qu.: 98.00       3rd Qu.:94608   3rd Qu.:3.000   3rd Qu.: 2.500
 Max.   :5000   Max.   :67.00   Max.   :43.0          Max.   :224.00       Max.   :96651   Max.   :4.000   Max.   :10.000
                                                                                            NA's   :18
   Education        Mortgage     Personal Loan   Securities Account   CD Account         Online        CreditCard
 Min.   :1.000   Min.   :  0.0   Min.   :0.000   Min.   :0.0000      Min.   :0.0000   Min.   :0.0000   Min.   :0.000
 1st Qu.:1.000   1st Qu.:  0.0   1st Qu.:0.000   1st Qu.:0.0000      1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
 Median :2.000   Median :  0.0   Median :0.000   Median :0.0000      Median :0.0000   Median :1.0000   Median :0.000
 Mean   :1.881   Mean   : 56.5   Mean   :0.096   Mean   :0.1044      Mean   :0.0604   Mean   :0.5968   Mean   :0.294
 3rd Qu.:3.000   3rd Qu.:101.0   3rd Qu.:0.000   3rd Qu.:0.0000      3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :3.000   Max.   :635.0   Max.   :1.000   Max.   :1.0000      Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

**Fig: Summary of data without missing value treatment**

```
       ID        Age (in years)  Experience (in years) Income (in K/month)    ZIP Code     Family members      CCAvg
 Min.   :   1   Min.   :23.00   Min.   :-3.0          Min.   :  8.00       Min.   : 9307   Min.   :1.000   Min.   : 0.000
 1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0          1st Qu.: 39.00       1st Qu.:91911   1st Qu.:1.000   1st Qu.: 0.700
 Median :2500   Median :45.00   Median :20.0          Median : 64.00       Median :93437   Median :2.000   Median : 1.500
 Mean   :2500   Mean   :45.34   Mean   :20.1          Mean   : 73.77       Mean   :93153   Mean   :2.392   Mean   : 1.938
 3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0          3rd Qu.: 98.00       3rd Qu.:94608   3rd Qu.:3.000   3rd Qu.: 2.500
 Max.   :5000   Max.   :67.00   Max.   :43.0          Max.   :224.00       Max.   :96651   Max.   :4.000   Max.   :10.000
   Education        Mortgage     Personal Loan   Securities Account   CD Account         Online        CreditCard
 Min.   :1.000   Min.   :  0.0   Min.   :0.000   Min.   :0.0000      Min.   :0.0000   Min.   :0.0000   Min.   :0.000
 1st Qu.:1.000   1st Qu.:  0.0   1st Qu.:0.000   1st Qu.:0.0000      1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
 Median :2.000   Median :  0.0   Median :0.000   Median :0.0000      Median :0.0000   Median :1.0000   Median :0.000
 Mean   :1.881   Mean   : 56.5   Mean   :0.096   Mean   :0.1044      Mean   :0.0604   Mean   :0.5968   Mean   :0.294
 3rd Qu.:3.000   3rd Qu.:101.0   3rd Qu.:0.000   3rd Qu.:0.0000      3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :3.000   Max.   :635.0   Max.   :1.000   Max.   :1.0000      Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

**Fig: Summary of data with missing value treatment**

```
> FamilySummary=table(mydata$`Family members`)
> FamilySummary

   1    2    3    4
1464 1292 1009 1217
```

**Fig: Mode of Family Members**

Experience seems to have some negative values which is not correct as it can't be negative in nature.

```
summary(mydata$`Experience (in years)`)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -3.0    10.0    20.0    20.1    30.0    43.0
```

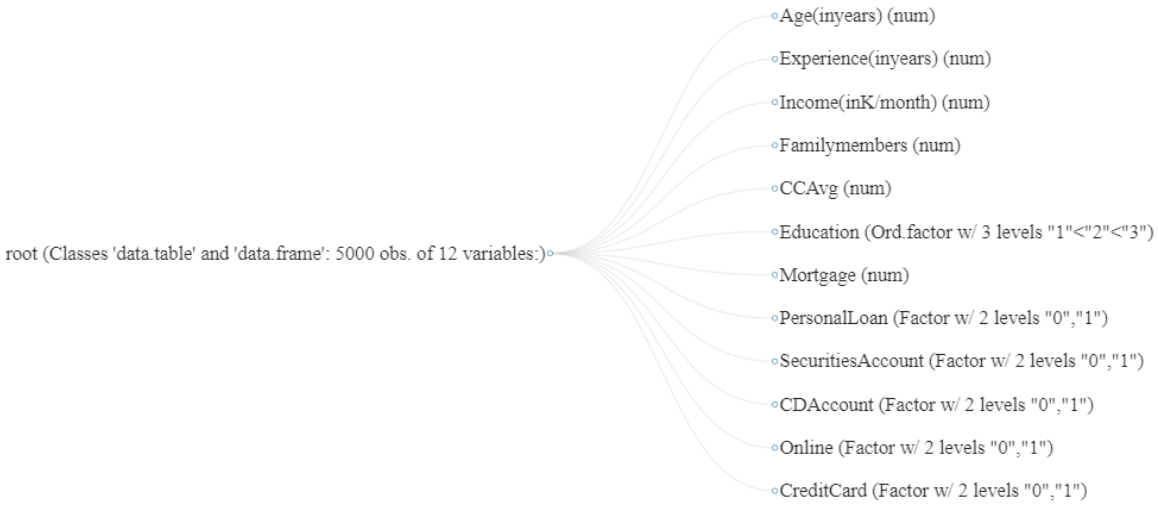
There are 6 IDs with negative experience and therefore, needs to be treated.

```
> head(mydata[mydata$`Experience (in years)`<0,])
# A tibble: 6 x 14
     ID `Age (in years)` `Experience (in~ `Income (in K/m~ `ZIP Code` `Family members` CCAvg Education Mortgage `Personal Loan`
  <dbl>            <dbl>             <dbl>           <dbl>      <dbl>            <dbl> <dbl> <ord>        <dbl> <fct>
1    90               25                -1             113      94303                4   2.3 3                0 0
2   227               24                -1              39      94085                2   1.7 2                0 0
3   316               24                -2              51      90630                3   0.3 3                0 0
4   452               28                -2              48      94132                2   1.75 3              89 0
5   525               24                -1              75      93014                4   0.2 1                0 0
6   537               25                -1              43      92173                3   2.4 2              176 0
# ... with 4 more variables: `Securities Account` <fct>, `CD Account` <fct>, Online <fct>, CreditCard <fct>
```

We are considering this as a data entry error and therefore, we have entered the absolute value of the data and hence, the data is rectified.

```
> summary(mydata$`Experience (in years)`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   10.00   20.00   20.13   30.00   43.00
```

ID and Zip Code doesn't seem to have any contribution on the overall outcome of the model and therefore, we will be removing the columns from the data.




### 1.2 Exploratory Data Analysis:

### Univariate Analysis:

### Numerical Data:

For the numerical data we will be creating boxplot to see if the data is normally distributed.

**Age:**

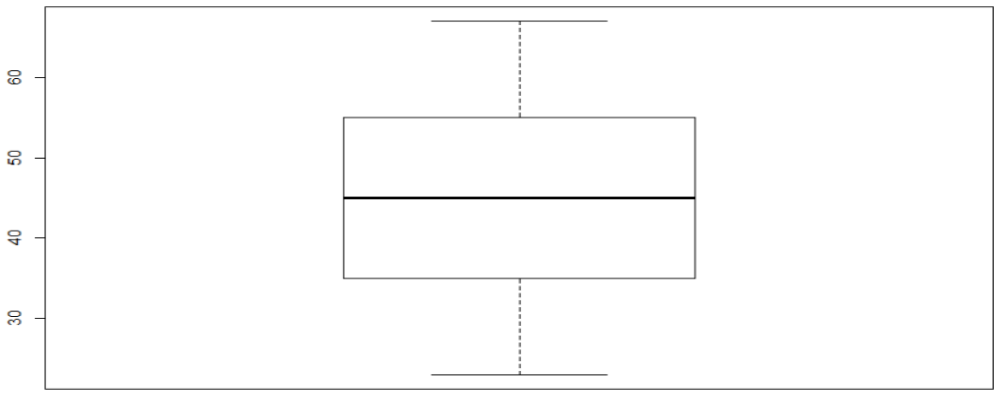



**Fig: Boxplot of Age**

Age seems to be normally distributed with minimum age from 23 years to maximum age of 67 years.

```
> summary(mydata$`Age (in years)`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.00   35.00   45.00   45.34   55.00   67.00
```

The median age and mean age are 45 years which shows that the data is normally distributed.

**Experience (in Years):**

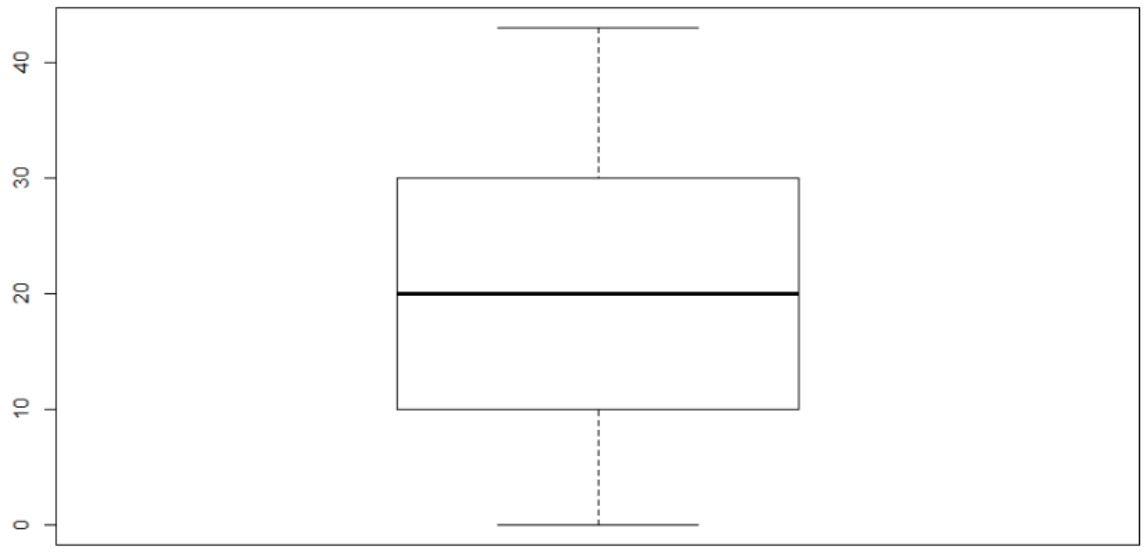



**Fig: Boxplot of Experience**

Experience is normally distributed with experience ranging from freshers to maximum 43 years of experience.

```
> summary(mydata$`Experience (in years)`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   10.00   20.00   20.13   30.00   43.00
```

The median as well as the mean experience is 20 years which means that the data is normally distributed.

**Income (inK/month):**

Income seems to have few outliers in the data. The minimum income is 8K/month whereas the maximum income is 224K/month. The mean and median income are also varied; mean is 73.77K/month whereas the median is 64K/month.

This shows the variance in income level and provide us the opportunity to provide different plans for different users.

```
> summary(mydata$`Income (in K/month)`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   8.00   39.00   64.00   73.77   98.00  224.00
```

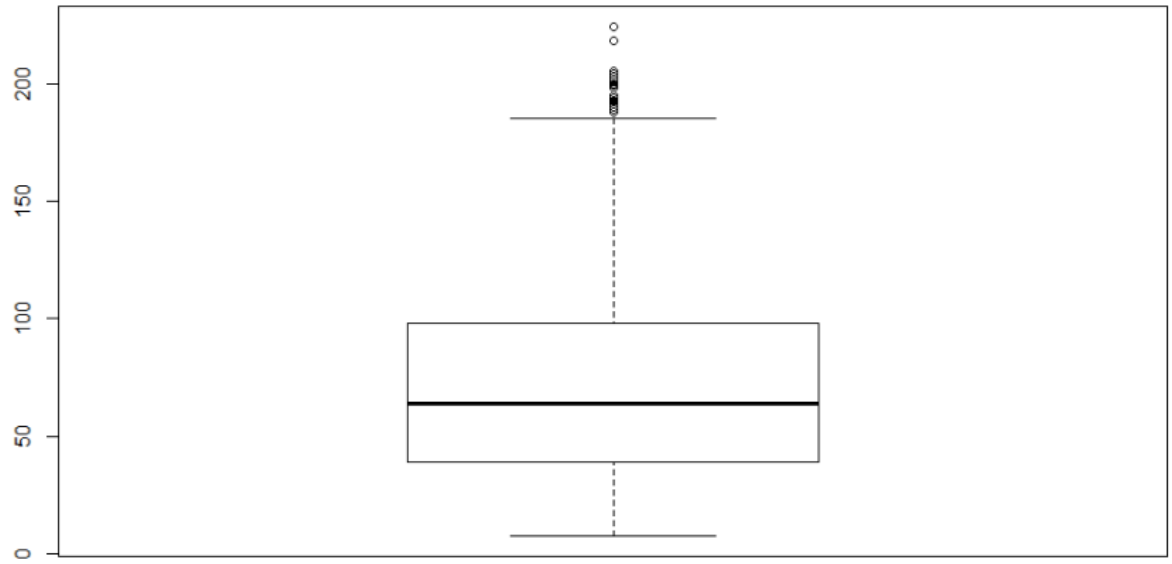
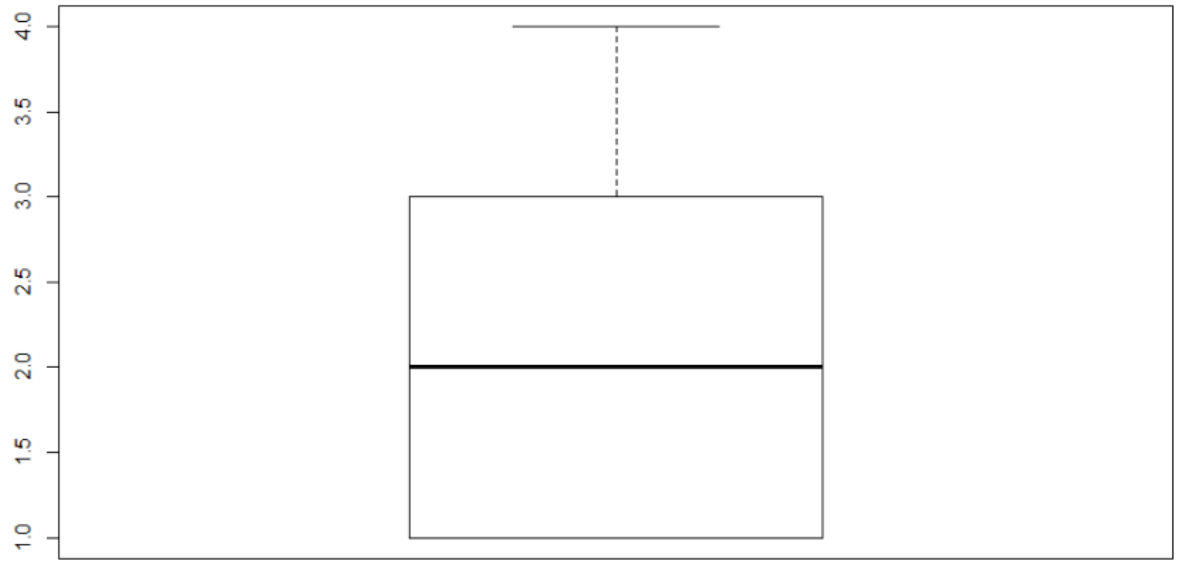



**Fig: Boxplot of Income (inK/month)**

**Family Members:**



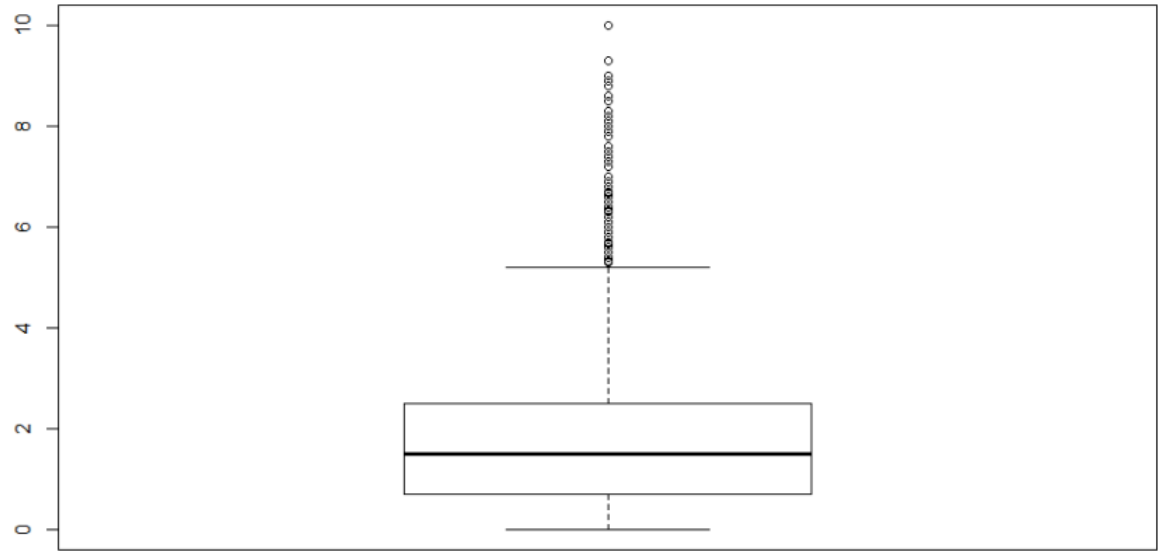

```
> summary(mydata$`Family members`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   2.392   3.000   4.000
```

From the boxplot, we can see that the data is normally distributed; family member ranges from 1 member i.e. themselves to maximum of 4 members.

**CCAvg:**

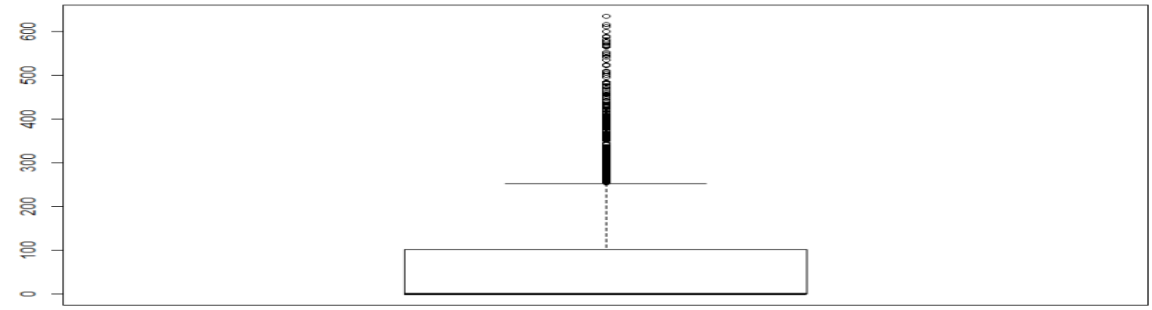
Credit card's average also have some oultiers which doesn't need any treatment as the income level of people varies the credit card spending will also vary. There are people with no credit cards while there are some which spends upto 10K.

```
summary(mydata$CCAvg)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.000   0.700   1.500   1.938   2.500  10.000
```




**Mortgage:**

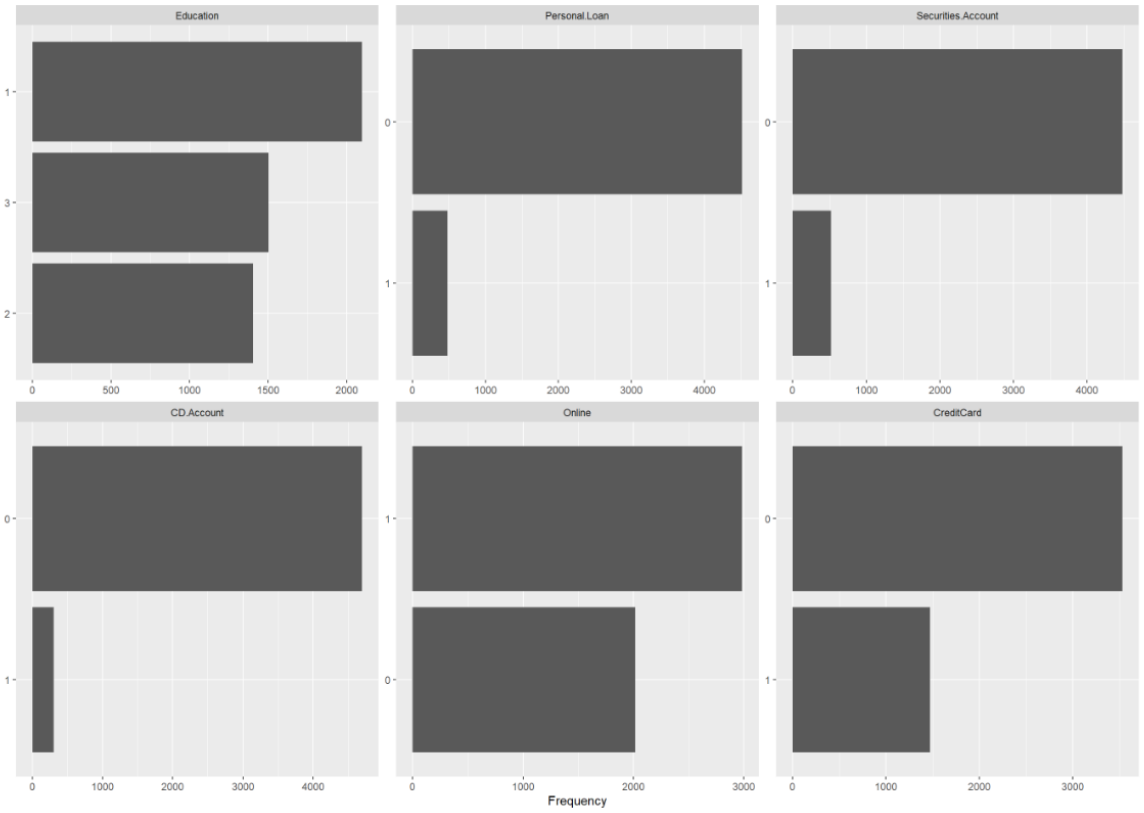
Mortgage data also has lot of outliers which also doesn't need treatments as the user profile varies the mortgage also varies.




```
summary(mydata$Mortgage)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0     0.0     0.0    56.5   101.0   635.0
```

**Bar chart for Categorical Variable:**

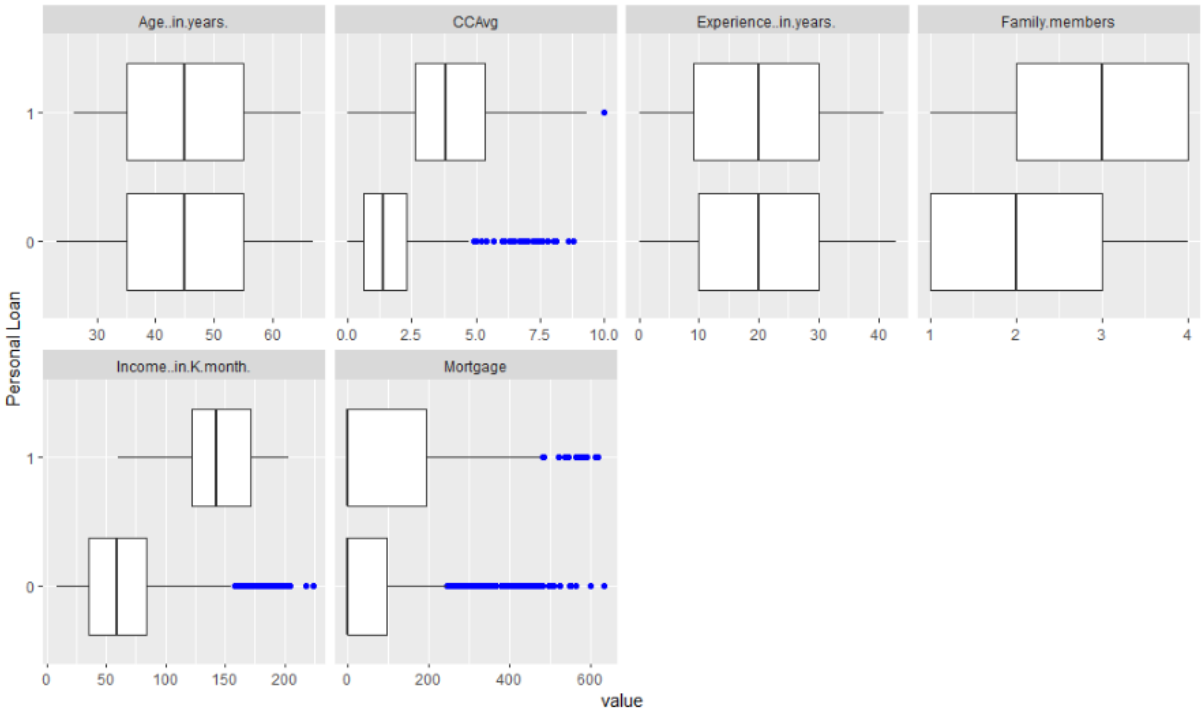
Bar Chart (by frequency)




The categorical variable is normal in nature and there doesn't seem to be any disturbance in the data.

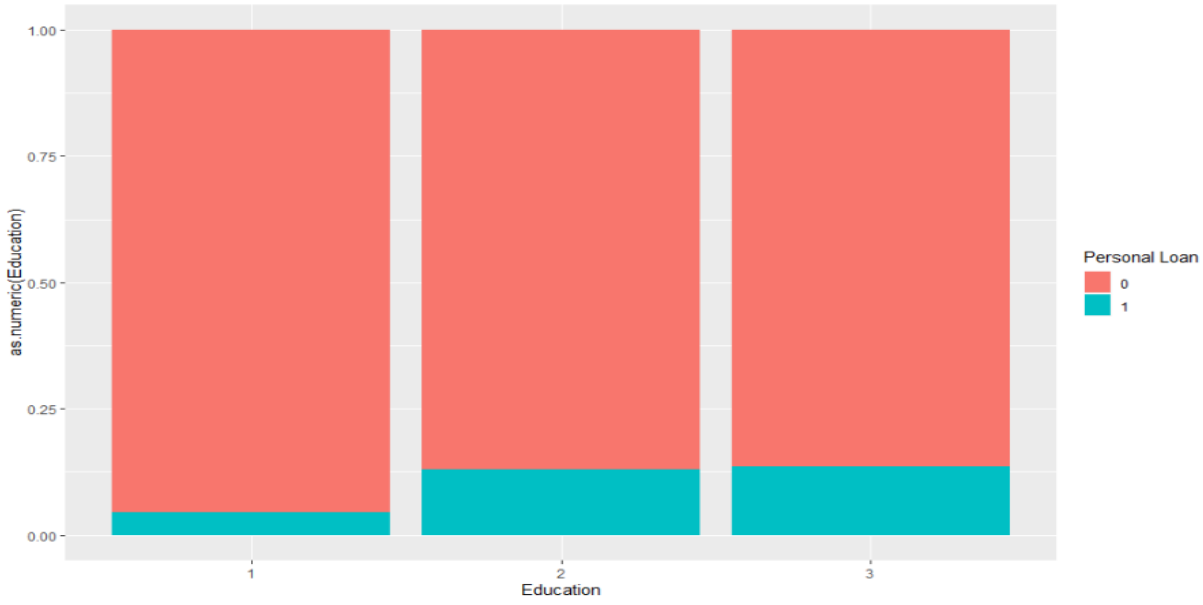
**Bivariate Analysis:**

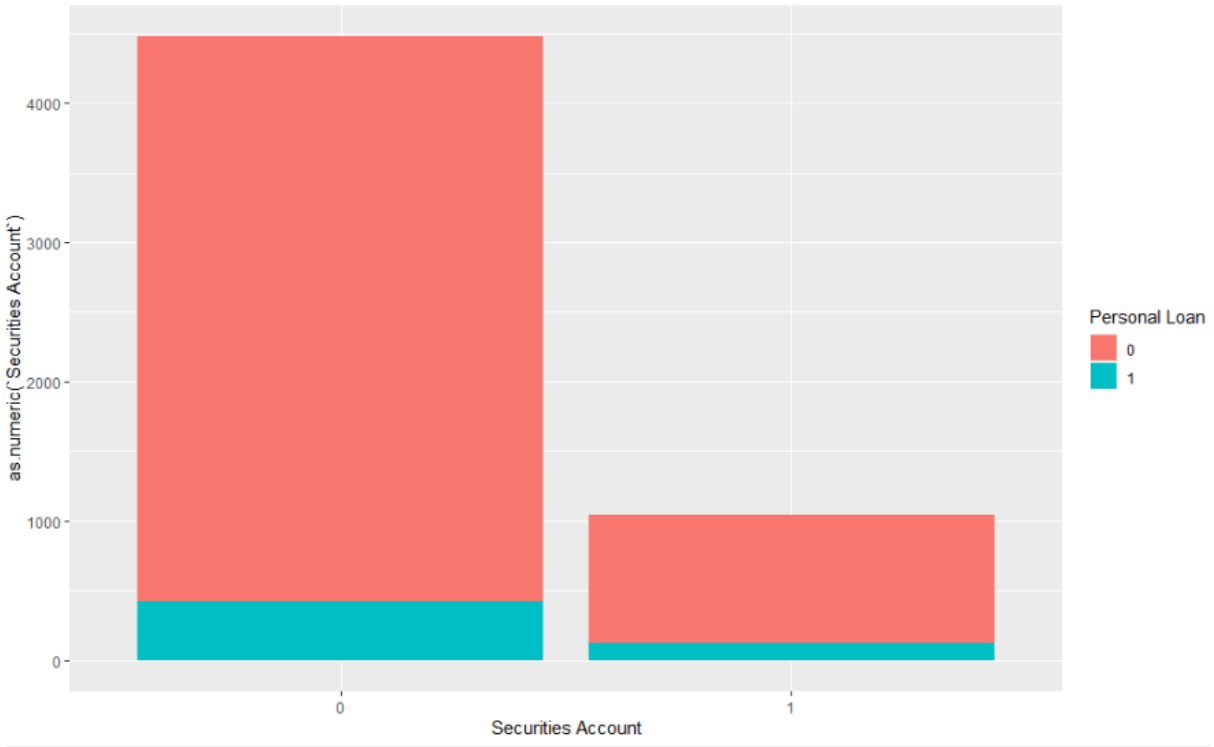
**Personal Loan (Y) vs all numerical variables:**

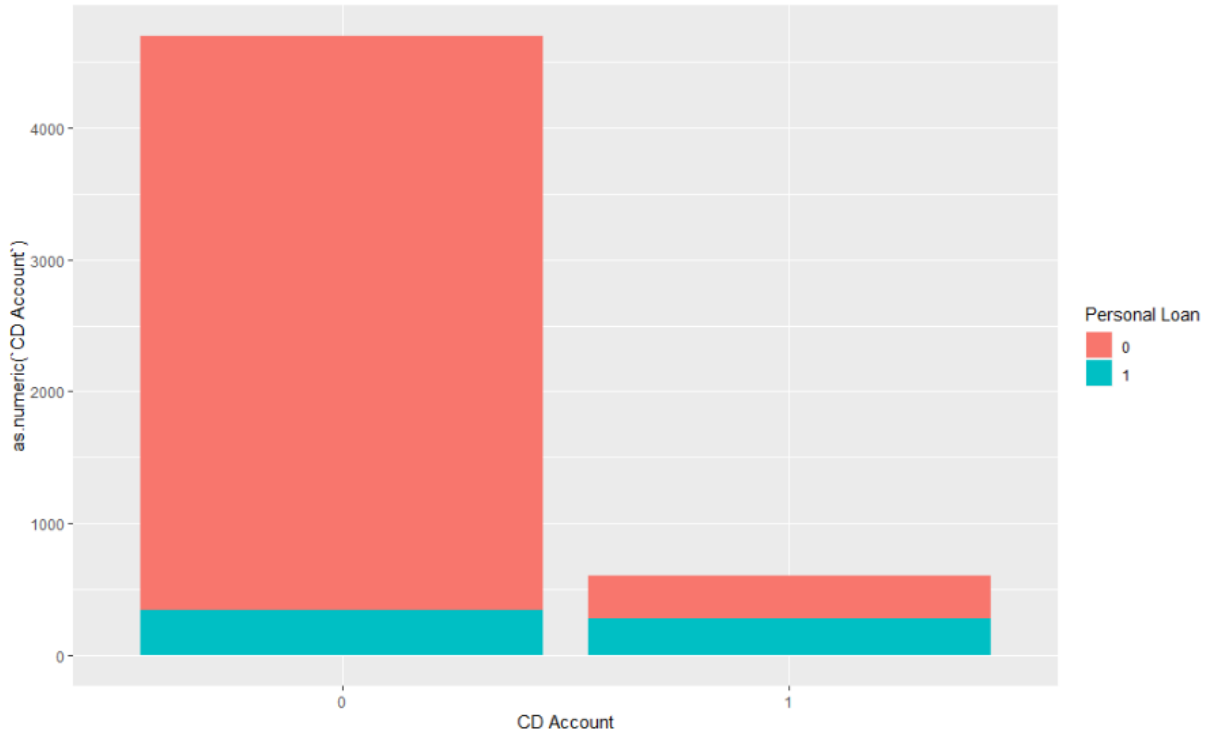


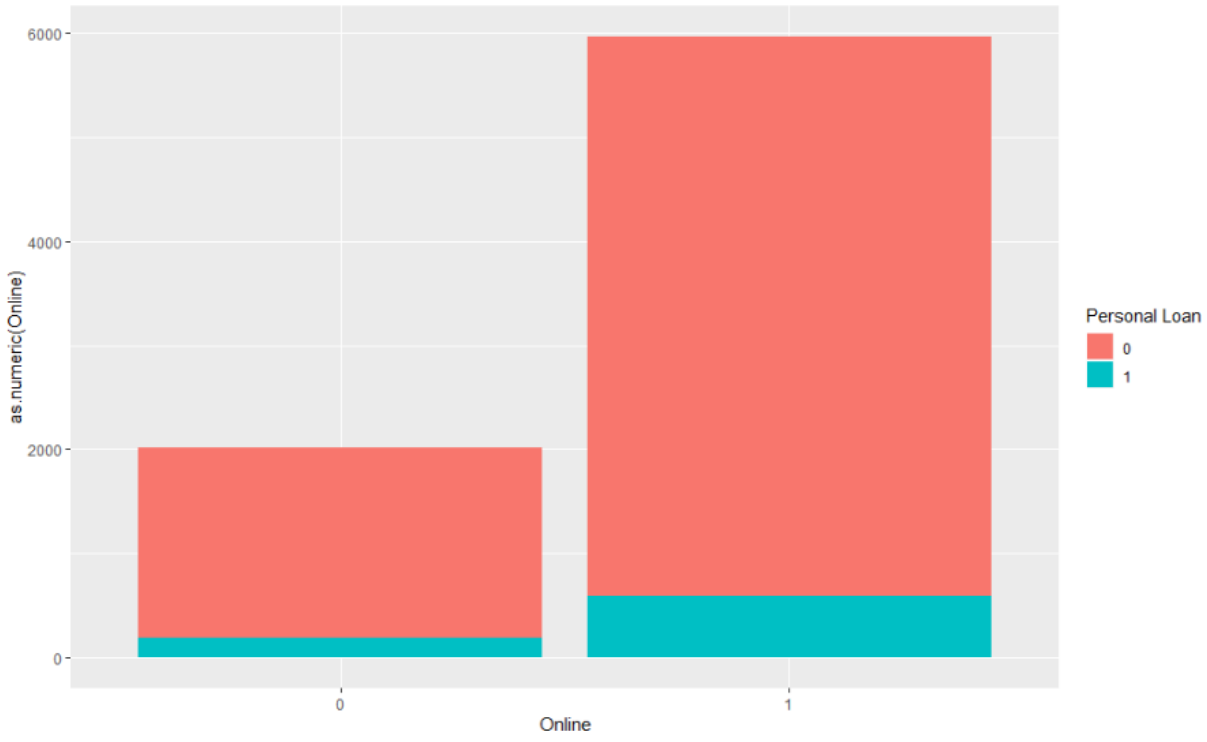

There are lot of non-person loan takers present as outliers in Credit Card, Mortgage and Income predictors.

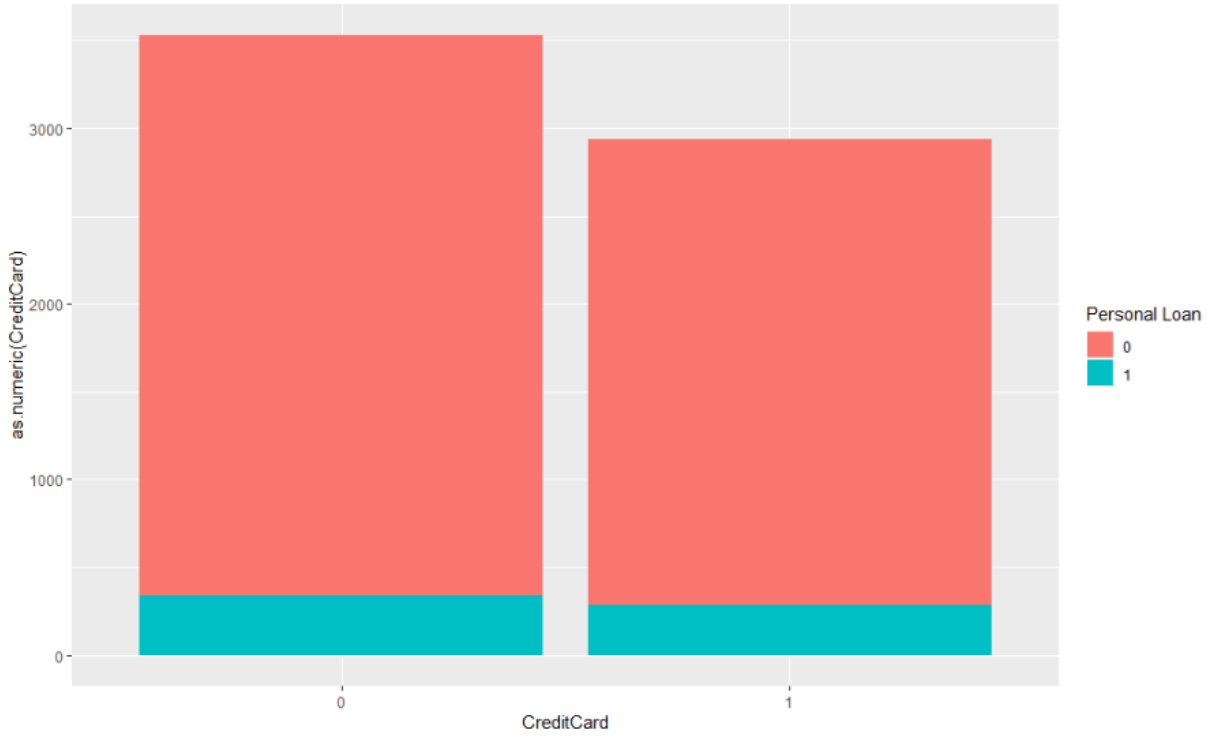
**Personal Loan (Y) vs categorical variables:**




4000
3000
2000
1000
0
as.numeric(`Securities Account`)
0
1
Securities Account
Personal Loan
0
1


4000
3000
2000
1000
0
as.numeric(`CD Account`)
0
1
CD Account
Personal Loan
0
1


6000
4000
2000
0
as.numeric(Online)
0
1
Online
Personal Loan
0
1


3000
2000
1000
0
as.numeric(CreditCard)
0
1
CreditCard
Personal Loan
0
1

**CART Model:**

Since we are done with data cleaning and EDA we will now build the model using CART (Classification & Regression Tree)

**Train & Test Set:**

As a rule of thumb, we are splitting the data into 70/30 i.e. 70% of the data for training and the remaining 30% is kept for testing once the model is built.

```
#Splitting of Dataset into Train - Test set
set.seed(1234)
## sampling 70% of data for training the algorithms using random sampling
mydata.index = sample(1:nrow(mydata), nrow(mydata)*0.70)
mydata.train = mydata[mydata.index,]
mydata.test = mydata[-mydata.index,]
dim(mydata.train)
dim(mydata.test)
```

Train & Test data structure:

```
> dim(mydata.train)
[1] 3500   12
> dim(mydata.test)
[1] 1500   12
>
```

Training data has 3500 observations while the test data has 1500 observations which justifies the split.

We will also check if the train and test data has equal split of personal loan values.

```
> prop.table(table(mydata.train$`Personal Loan`))

         0          1
0.90285714 0.09714286
> prop.table(table(mydata.test$`Personal Loan`))

         0          1
0.90666667 0.09333333
```

The Train and Test data has equal ration of personal loan values.

**Full Tree:**

**R Code:**

```
#Builing the full grown tree

library(rpart)
library(rpart.plot)
set.seed(1234)

tree_full = rpart(formula = `Personal Loan`~., data = mydata.train, cp=-1, minsplit=2, minbucket=1)
rpart.plot(tree_full, cex=0.7)

print(tree_full)
```

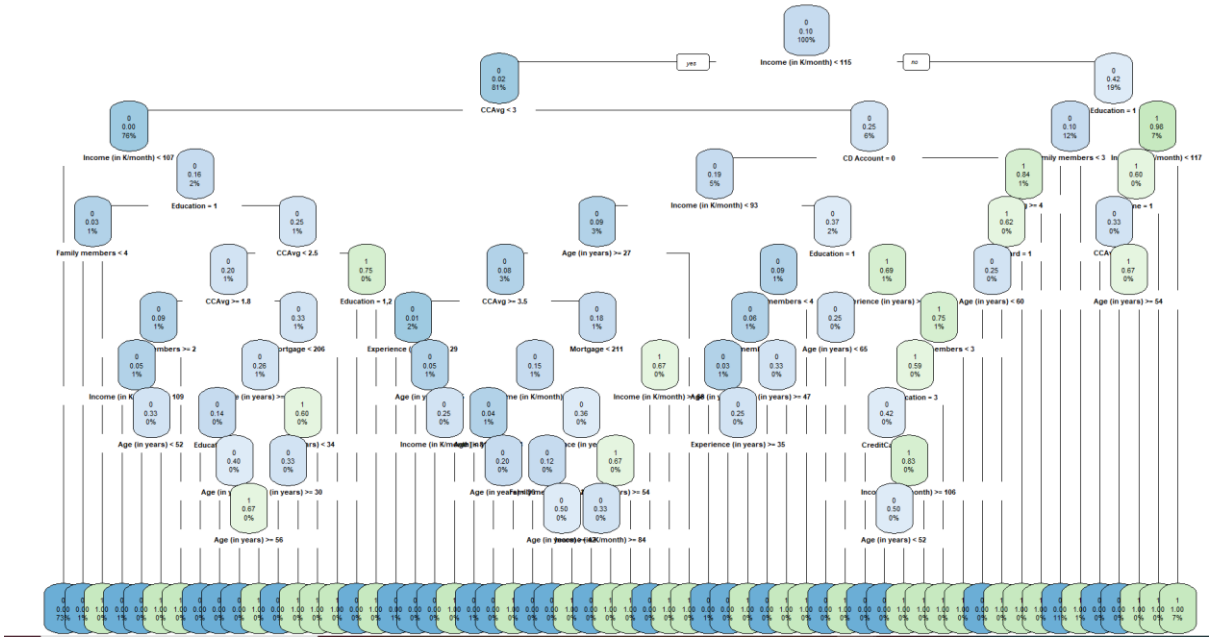


**Fig: Fully grown Tree plot**

**Prediction using the full tree:**

```
## Predict using the CART model
mydata.train$predict.class=predict(tree_full,mydata.train,type="class")
mydata.train$predict.score=predict(tree_full,mydata.train)
```

**Output:**

```
> ## Creating the confusion matrix
> tabtrain=with(mydata.train,table(`Personal Loan`,predict.class))
> tabtrain
              predict.class
Personal Loan    0    1
            0 3171    0
            1    0  329
```

The output shows that the model is overfitted. Since the model is a full fit model the model accuracy will be very high. However, it will not provide correct output for the test data.

This model will not suffice when real data is put into the model

```
> tabtest=with(mydata.test,table(`Personal Loan`,predict.class))
> tabtest
              predict.class
Personal Loan    0    1
            0 1333   16
            1   19  132

                      ACC       SENS       SPEC
tree_full_train 1.0000000 1.0000000 1.0000000
tree_full_test  0.9766667 0.8741722 0.9881394
```

**Prune Tree Model:**

**Pruning the tree:**

Pruning can be done by using two methods:

a) Manual Pruning
b) CP Method

We will try to prune the model using both the methods and measure the model performance of both the pruned models

**Manual Pruning:**

In manual pruning, we set the minsplit and minbucket manually. We will set the minsplit as 1% of train data and the minbucket as 1/3rd of min split

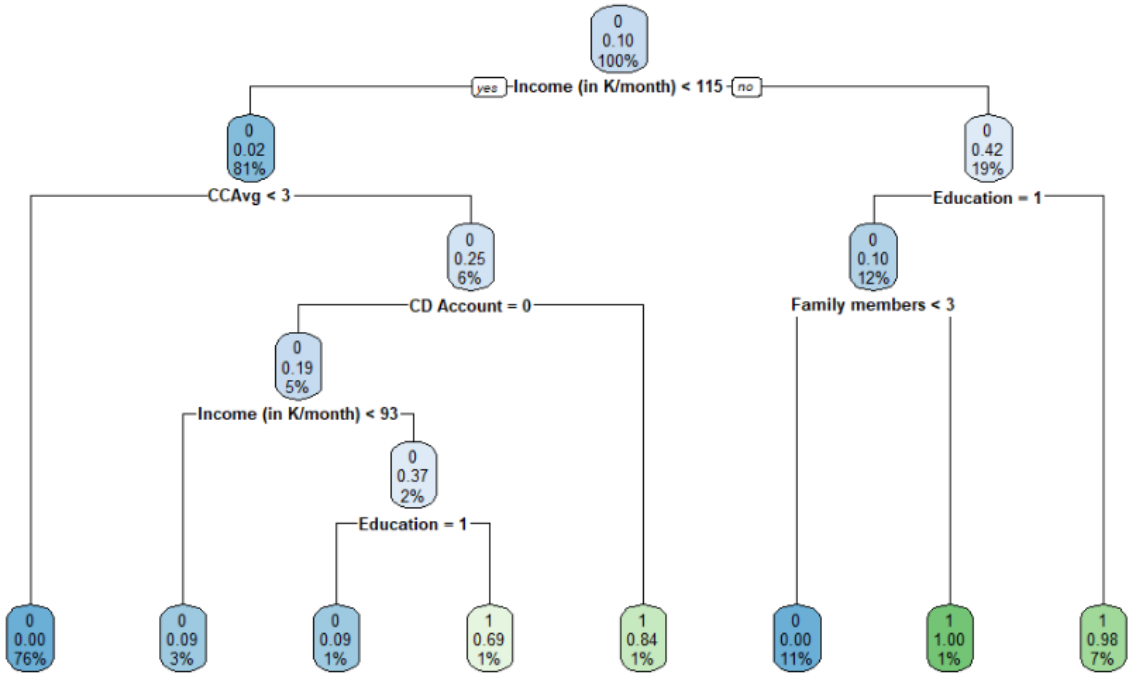


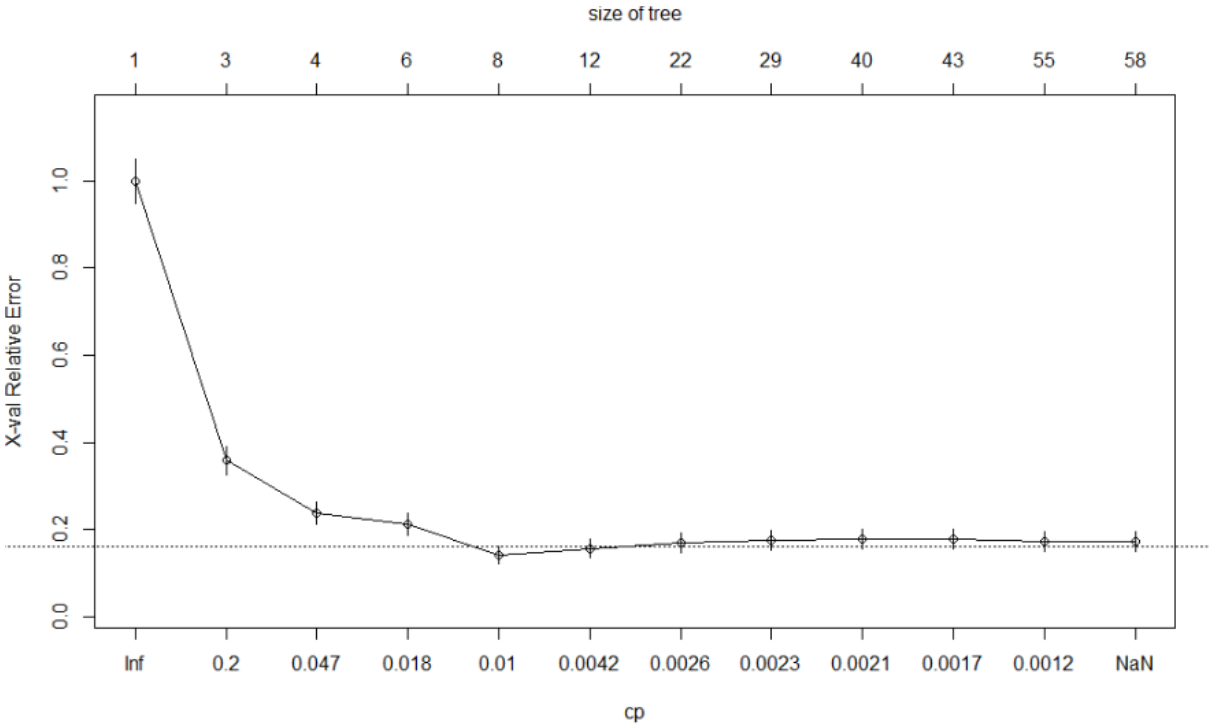

Fig: Pruned Tree

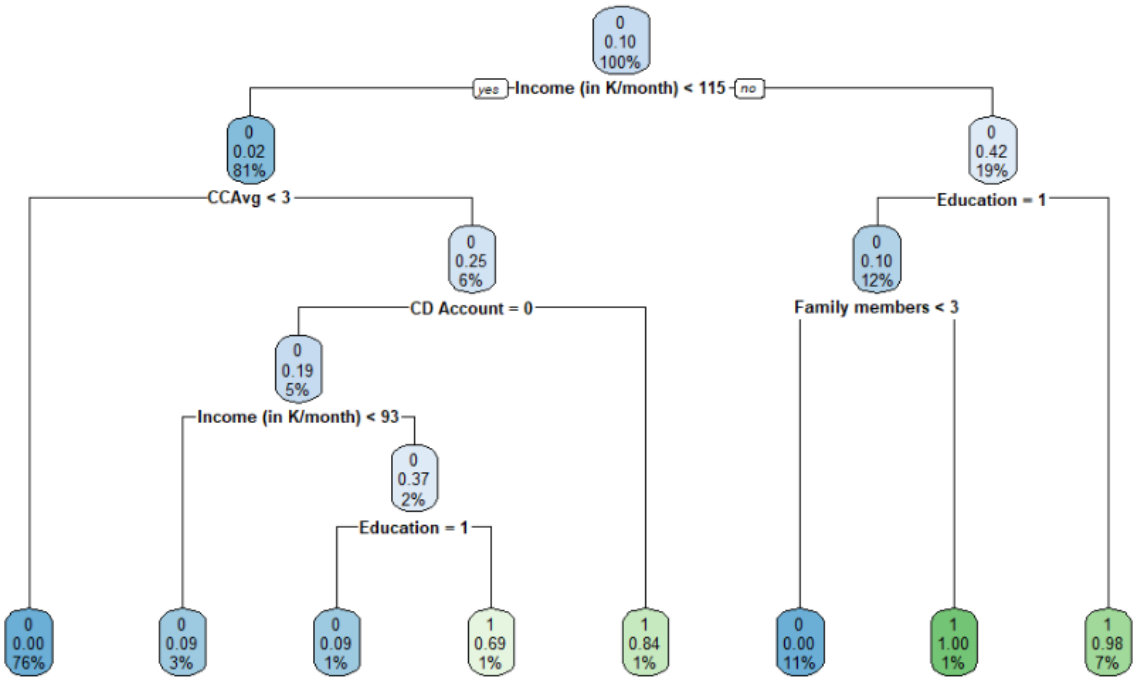
Confusion Matrix (Manual Pruned Model):

```
> tabtrain=with(mydata.train,table(`Personal Loan`,predict.class))
> tabtrain
             predict.class
Personal Loan    0    1
            0 3150   21
            1   24  305
```

```
> tabtest=with(mydata.test,table(`Personal Loan`,predict.class))
> tabtest
             predict.class
Personal Loan    0    1
            0 1338   11
            1   18  133

                              ACC       SENS       SPEC
tree_full_train         1.0000000 1.0000000 1.0000000
tree_full_test          0.9766667 0.8741722 0.9881394
tree_manual_prune_train 0.9871429 0.9270517 0.9933775
tree_manual_prune_test  0.9806667 0.8807947 0.9918458
```

**CP Prune Model:**






CP Pruned:

```
> tabtrain=with(mydata.train,table(`Personal Loan`,predict.class))
> tabtrain
             predict.class
Personal Loan    0    1
            0 3164    7
            1   23  306

> tabtest=with(mydata.test,table(`Personal Loan`,predict.class))
> tabtest
             predict.class
Personal Loan    0    1
            0 1341    8
            1   21  130
```

| | ACC | SENS | SPEC |
|---|---|---|---|
| tree_full_train | 1.000 | 1.000 | 1.000 |
| tree_full_test | 0.977 | 0.874 | 0.988 |
| tree_manual_prune_train | 0.987 | 0.927 | 0.993 |
| tree_manual_prune_test | 0.981 | 0.881 | 0.992 |
| cptree_train | 0.991 | 0.930 | 0.998 |
| cptree_test | 0.981 | 0.861 | 0.994 |

Test sensitivity is highest in the manual pruning and accuracy is also high.

**Random Forest Model – Full:**

Considering mtry = 3 and ntree = 800, we have plotted the full model

```
Call:
 randomForest(formula = Personal.Loan ~ ., data = traindata, ntree_1 = 800,      mtry_1 = 3, importance = TRUE,
set.seed(3021))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 1.07%
Confusion matrix:
     0   1 class.error
0 3612   4 0.001106195
1   39 345 0.101562500
```

**Random Forest Model – by changing the cut-off:**

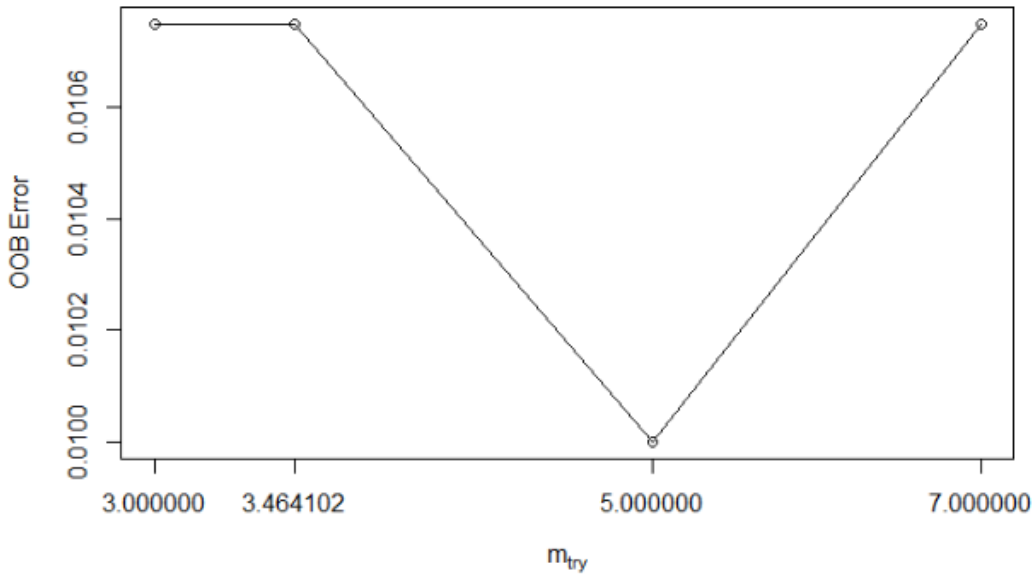
New cut-off is 60:40 with 60% being the positive class.

```
Call:
 randomForest(formula = Personal.Loan ~ ., data = traindata, ntree_1 = 800,      mtry_1 = 3, importance = TRUE, cutoff =
c(0.6, 0.4), set.seed(3021))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 0.92%
Confusion matrix:
     0   1 class.error
0 3608   8 0.002212389
1   29 355 0.075520833
```

**Random Forest Model –tuned:**

Tuning parameters:
mtryStart = mtry_1,
stepFactor = 1.5,
ntreeTry = 501,
improve = 0.01,
trace=TRUE,
plot=TRUE,
doBest=TRUE,
importance=TRUE

```
Call:
 randomForest(x = x, y = y, xtest = ..2, mtry = res[which.min(res[,      2]), 1],
importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 0.95%
Confusion matrix:
     0   1 class.error
0 3610   6 0.001659292
1   32 352 0.083333333
```


OOB Error
0.0100
0.0102
0.0104
0.0106
3.000000
3.464102
5.000000
7.000000
mtry

**Various Model Performance Measures (Confusion Matrix):**

**Confusion Matrix – CART Full Model:**

```
> ## Creating the confusion matrix
> tabtrain=with(mydata.train,table(`Personal Loan`,predict.class))
> tabtrain
             predict.class
Personal Loan    0    1
            0 3171    0
            1    0  329
```

**Fig: Train data**

```
> tabtest=with(mydata.test,table(`Personal Loan`,predict.class))
> tabtest
             predict.class
Personal Loan    0    1
            0 1333   16
            1   19  132
```

**Fig: Test Data**

**Confusion Matrix –CART Manual Pruned:**

```
> tabtrain=with(mydata.train,table(`Personal Loan`,predict.class))
> tabtrain
             predict.class
Personal Loan    0    1
            0 3150   21
            1   24  305
```

**Fig: Train Data**

```
> tabtest=with(mydata.test,table(`Personal Loan`,predict.class))
> tabtest
             predict.class
Personal Loan    0    1
            0 1338   11
            1   18  133
```

**Fig: Test Data**

**Confusion Matrix – CART CP Pruned:**

```
> tabtrain=with(mydata.train,table(`Personal Loan`,predict.class))
> tabtrain
             predict.class
Personal Loan    0    1
            0 3164    7
            1   23  306
```

**Fig: Train Data**

```
> tabtest=with(mydata.test,table(`Personal Loan`,predict.class))
> tabtest
              predict.class
Personal Loan    0    1
            0 1341    8
            1   21  130
```

**Fig: Test Data**

**Confusion Matrix – Random Forest Cut-off (RF2):**

```
                predict.loanclass
Personal.Loan    0    1
            0 3616    0
            1    0  384
```

**Fig: Train Data**

```
                predict.loanclass
Personal.Loan   0   1
            0 896   8
            1  14  82
[1] 0.978
[1] 0.8541667
[1] 0.9911504
```

**Fig: Test Data**

**Confusion Matrix – Random Forest Tuned (RF_tune):**

```
                predict.loanclass
Personal.Loan    0    1
            0 3616    0
            1    0  384
```

**Fig: Train Data**

```
                predict.loanclass
Personal.Loan   0   1
            0 900   4
            1  14  82
[1] 0.982
[1] 0.8541667
[1] 0.9955752
```

**Fig: Test Data**

**Model Performance Measure:**

**CART performance:**

| | ACC | SENS | SPEC |
|---|---|---|---|
| tree_full_train | 1.000 | 1.000 | 1.000 |
| tree_full_test | 0.977 | 0.874 | 0.988 |
| tree_manual_prune_train | 0.987 | 0.927 | 0.993 |
| tree_manual_prune_test | 0.981 | 0.881 | 0.992 |
| cptree_train | 0.991 | 0.930 | 0.998 |
| cptree_test | 0.981 | 0.861 | 0.994 |

**Random Forest Performance:**

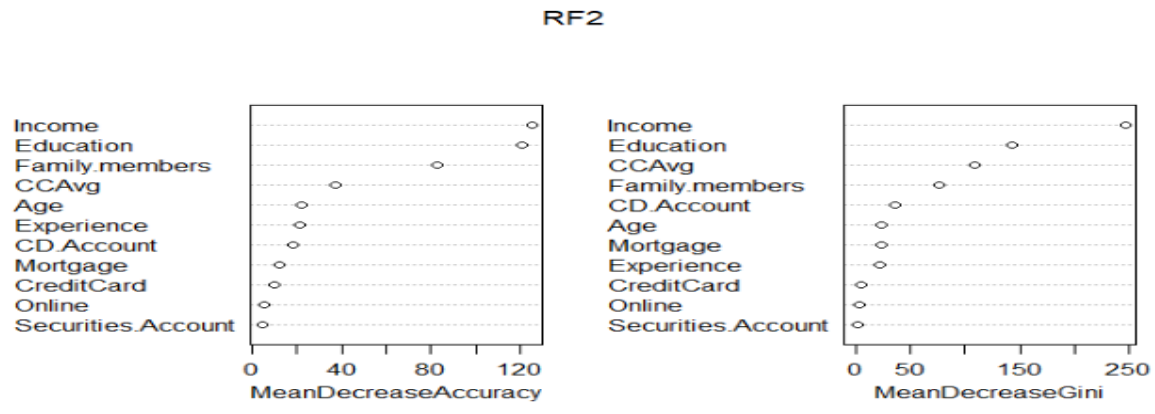
| | ACC <dbl> | SENS <dbl> | SPEC <dbl> |
|---|---|---|---|
| RF2_train | 1.000 | 1.0000000 | 1.0000000 |
| RF2_test | 0.978 | 0.8541667 | 0.9911504 |
| TuneRF_train | 1.000 | 1.0000000 | 1.0000000 |
| TuneRF_test | 0.982 | 0.8541667 | 0.9955752 |

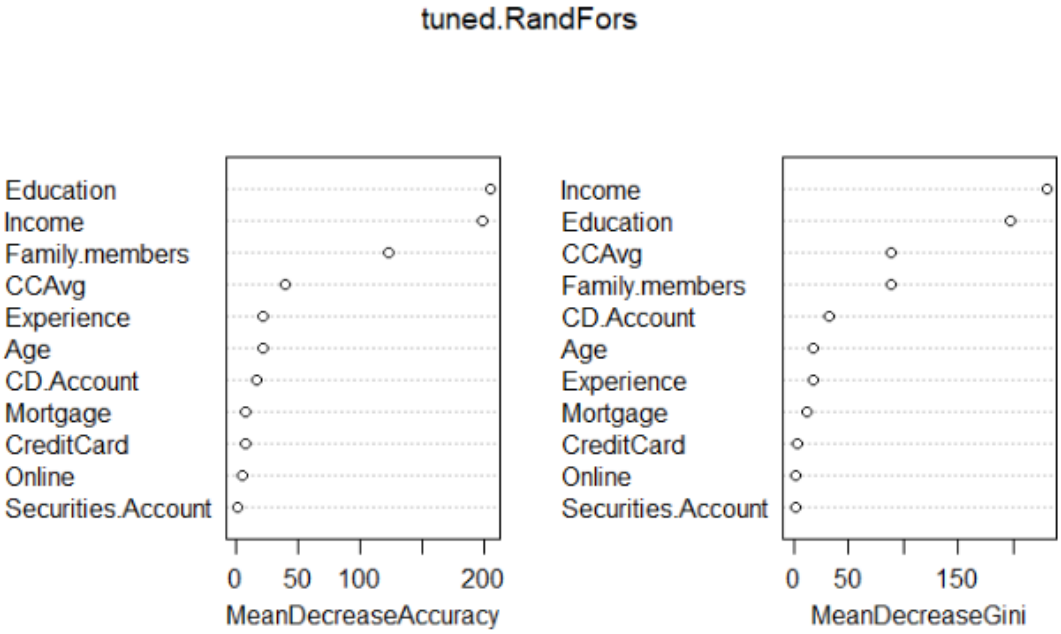
4 rows

The Accuracy of RF_Tune(tuned.RanFors) model is the highest and therefore, we will be proceeding with the model. Also the specificity of the model is higher than the rest and hence we will prepare algorithm with this model.

**Inferences:**

From both the Random Forest models we found out that Income, Education, Family Members and CC Average have the highest importance. Banks should concentrate on this variables for customers as these variables are the best factors deciding the probability of getting a loan.




tuned.RandFors
Education
Income
Family.members
CCAvg
Experience
Age
CD.Account
Mortgage
CreditCard
Online
Securities.Account
0
50
100
200
MeanDecreaseAccuracy
Income
Education
CCAvg
Family.members
CD.Account
Age
Experience
Mortgage
CreditCard
Online
Securities.Account
0
50
150
MeanDecreaseGini