DATA MINING ASSIGNMENT

MODULE - 3

MODEL DEVELOPMENT REPORT ON

CUSTOMER LOAN PURCHASE CLASSIFICATION FOR THERA BANK

SUBMITTED BY:

YASHDEEP SINGH

PGP – BABI

24TH NOVEMBER 2018

CONTENTS

PREFACE

This report documents the work done on Thera bank's dataset to classify and build a predictive model using unsupervised and supervised machine learning algorithms.

This report shall give you an overview of the key metrics derived using clustering and classification techniques and their interpretation. This will be achieved through visualisations and model performance measures. Report shall also elaborate future decisions the management may take to achieve their desired outcome based on our findings.

OBJECTIVE

The aim of this exercise is to facilitate the management of Thera bank in their decision making capability to increase revenue by earning interest on loans. We will support them by targeting the customers that have a relatively higher probability of opting for a personal loan. Our preliminary findings are based on a campaign run on liability customers last year where a success rate of slightly more than 9% was achieved. Among the 5000 people targeted in the campaign, only 9.6% (480) accepted the personal loan that was offered to them.

The management wants to devise a marketing strategy where the success ratio increases with minimal budget, in other words we will build a model that will increase the success ratio while at the same time reduce the cost of the campaign. We are also supposed to explore ways to convert liability customers (depositors) to personal loan customers while retaining them as depositors.

INTRODUCTION

Environment Setup:

```
setwd("C:/Users/Hp/Desktop/R Programming")
getwd()
LoanData <- read.csv("Thera Bank_Personal_Loan_Dataset.csv", header =
TRUE)
```

We start by setting up our working directory and reading the csv file into Rstudio as LoanData.

LIBRARIES USED

1. **ggplot21**: For visualizations. Building plots step by step from multiple sources.
2. **cluster**: For finding groups in data and plotting clusters.
3. **NbClust:** For determining the optimal number of clusters using different distance measures and clustering methods.
4. **dplyr:** Data manipulation. (Splitting, applying & combining data)
5. **tidyverse:** Intuitive R package for data science
6. **caTools:** For splitting data set into in-sample and out-sample
7. **rpart**: Building CART based decision trees.
8. **rpart.plot**: Plotting decision trees.
9. **randomForrest**: Create a randomForrest model.
10. **data.table:** Creating rank ordered tables
11. **ROCR:** To plot ROC curve, calculating K.S and AUC.
12. **ineq**: Gini coefficient
13. **InformationValue**: Calculating concordance/discordance ratios

VARIABLE EXPLANATION:

1. **ID**: Customer ID
2. **Age**: Customer's age (Years)
3. **Experience**: Professional experience (Years)
4. **Income**: Annual Income ($000)
5. **Zip Code**: Home Address zip code
6. **Family**: Family size oof the customer
7. **CCAvg**: Average spending on credit cards per month($000)
8. **Education**: Levels of education where (1 = Undergraduate, 2 = Graduate, 3 = Advanced/Professional)
9. **Mortgage**: Value of house mortgage if any ($000)
10. **Personal Loan**: Did customer accept the personal loan offered in last campaign?
11. **Securities Account**: Does the customer have a securities account with the bank?
12. **CD Account**: Does the customer have a certificate of deposit account with the bank?

13. **Online**: Does the person use internet banking facilities?
14. **CreditCard**: Does the person use a credit card issue by the bank?

MISSING VALUE IDENTIFICATION:

```
summary(LoanData)

##        ID         Age..in.years.  Experience..in.years. Income..in.K.mont
h.
##  Min.   :   1   Min.   :23.00   Min.   :-3.0          Min.   :  8.00
##  1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0          1st Qu.: 39.00
##  Median :2500   Median :45.00   Median :20.0          Median : 64.00
##  Mean   :2500   Mean   :45.34   Mean   :20.1          Mean   : 73.77
##  3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0          3rd Qu.: 98.00
##  Max.   :5000   Max.   :67.00   Max.   :43.0          Max.   :224.00
##
##     ZIP.Code      Family.members       CCAvg           Education
##  Min.   : 9307   Min.   :1.000   Min.   : 0.000   Min.   :1.000
##  1st Qu.:91911   1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000
##  Median :93437   Median :2.000   Median : 1.500   Median :2.000
##  Mean   :93153   Mean   :2.397   Mean   : 1.938   Mean   :1.881
##  3rd Qu.:94608   3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000
##  Max.   :96651   Max.   :4.000   Max.   :10.000   Max.   :3.000
##                  NA's   :18

*output simplified for readability
```

It is observed that the minimum professional experience is -3 and there are N/A values present in Family members. Professional experience of anything below zero can be attributed to him/her not yet having started their professional career, or in other words, it would not be absolutely wrong to admit that negative value in experience equals to zero years of professional experience.

```
LoanData$`Total Experience`[LoanData$`Total Experience`< 1] <- 0
```

```
colSums(is.na(LoanData))

##                  ID                  Age    Total Experience
##                   0                    0                   0
##              Income                  Zip      Family Members
##                   0                    0                  18
##               CCAvg            Education            Mortgage
##                   0                    0                   0
##       Personal Loan Securities Account          CD Account
##                   0                    0                   0
##              Online           CreditCard
##                   0                    0
```

Missing values in family members is an indication that there are customers present in the data who for some reason (personal/factually incorrect information) have no record of members in their family. Hence common sense suggests that these customers have zero family members and the N/A values should be converted to zero.

```
LoanData[is.na(LoanData)] <- 0
```

FEATURE TRANSFORMATION:

First renaming column names for better readability.

```
colnames(LoanData) <- c("ID", "Age", "Total Experience", "Income", "Zip",
"Family Members", "CCAvg", "Education","Mortgage", "Personal Loan",
"Securities Account", "CD Account", "Online", "CreditCard")
```

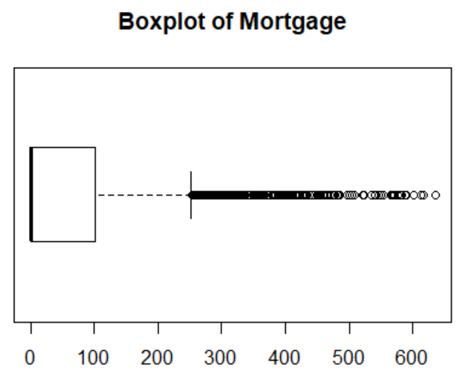
Next we will transform relevant variables from their current numeric status to factor (Eg family members, education, personal loan, securities account, CD account, online & credit card. We also remove any insignificant variable from the dataset that would not provide any additional value to our model building exercise (Eg ID & Zip)

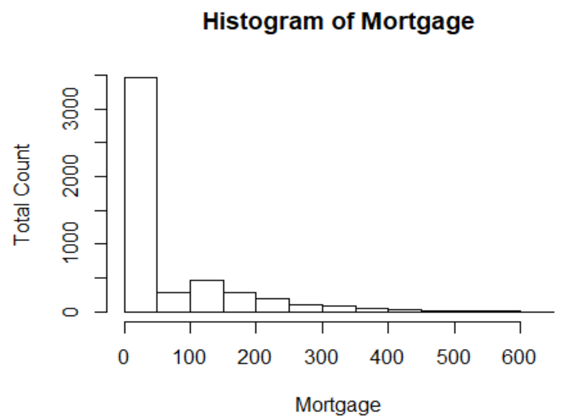
```
LoanData$`Family Members` <- as.factor(LoanData$`Family Members`)
LoanData$Education <- as.factor(LoanData$Education)
LoanData$`Personal Loan` <- as.factor(LoanData$`Personal Loan`)
LoanData$`Securities Account` <- as.factor(LoanData$`Securities Account`)
LoanData$`CD Account` <- as.factor(LoanData$`CD Account`)
LoanData$Online <- as.factor(LoanData$Online)
LoanData$CreditCard <- as.factor(LoanData$Online)
LoanData$ID <- NULL
LoanData$Zip <- NULL
str(LoanData)

## 'data.frame':    5000 obs. of  12 variables:
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Total Experience  : num  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ Family Members    : Factor w/ 5 levels "0","1","2","3",..: 5 4 2 2 5
5 3 2 4 2 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 3
2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2
...
##  $ Securities Account: Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1 1
...
```

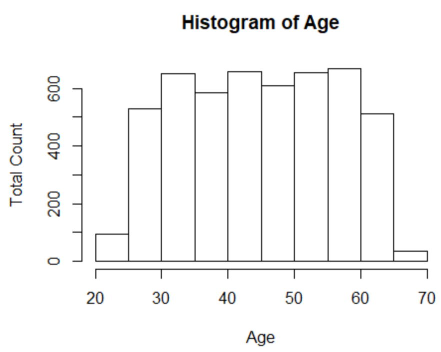
```
##  $ CD Account          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Online              : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1
...
##  $ CreditCard          : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1
...
```

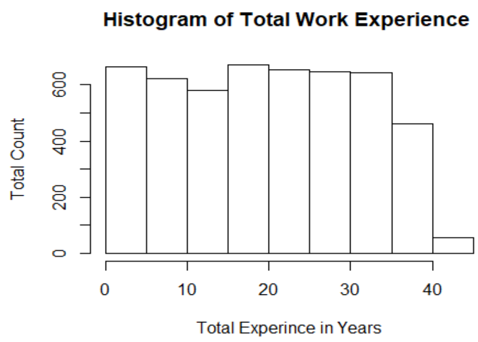
EXPLORATORY DATA ANALYSIS

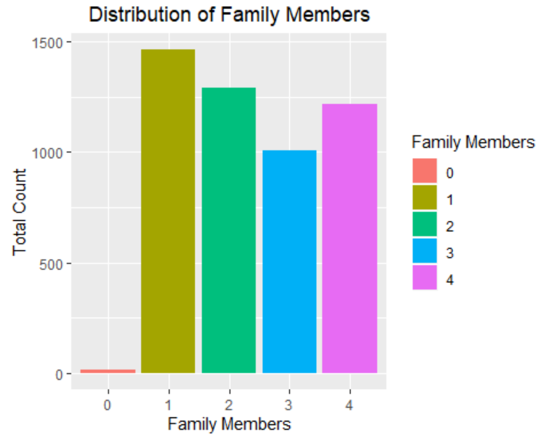
UNIVARIATE ANALYSIS:

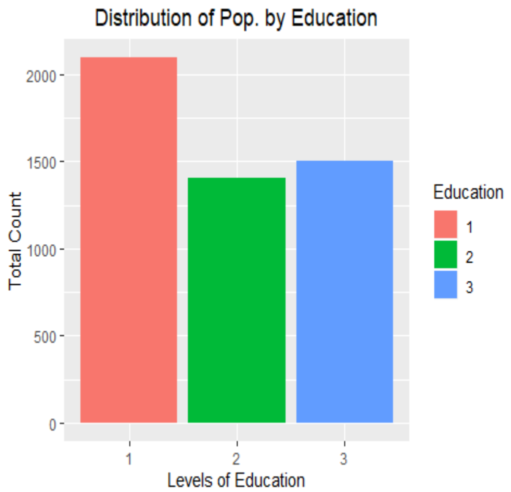


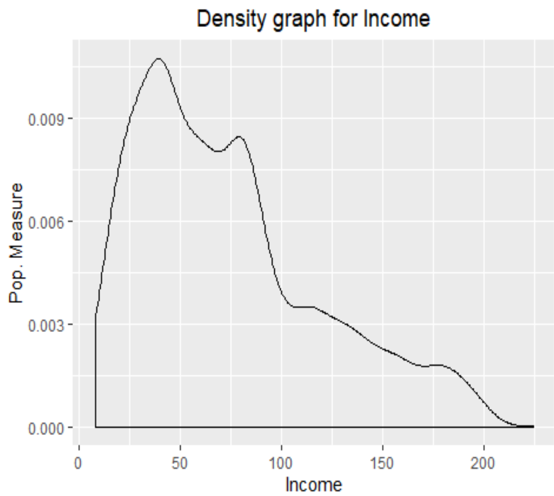


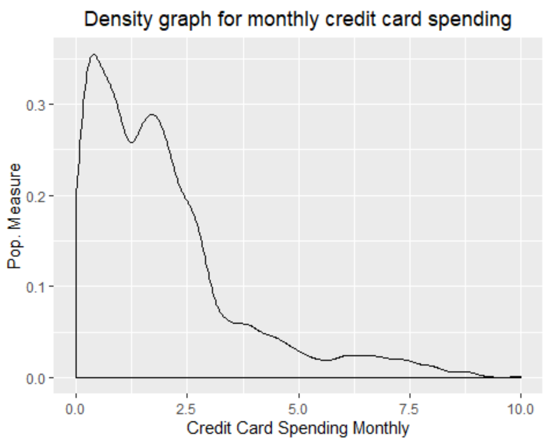


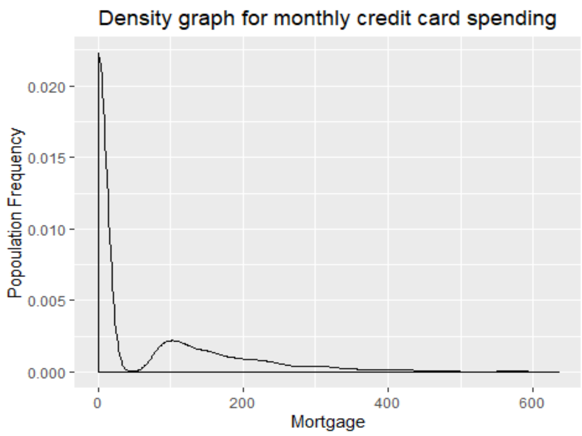










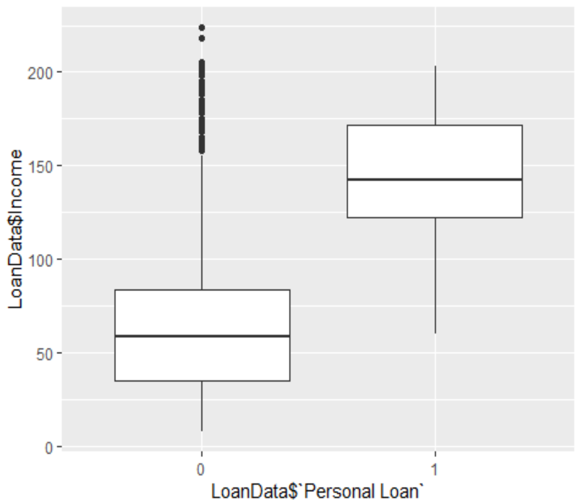


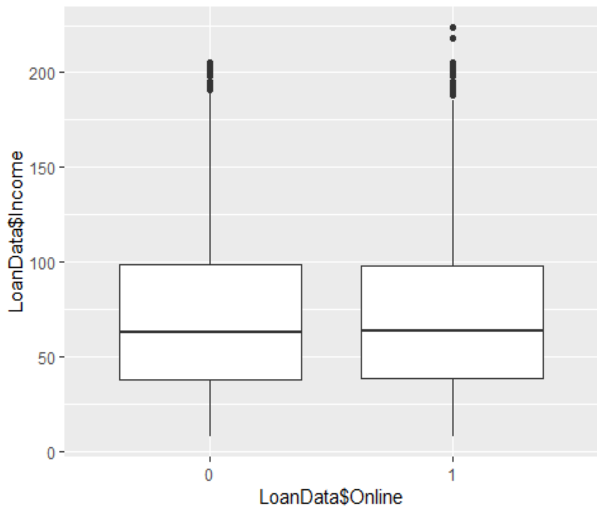
INTERPRETATION

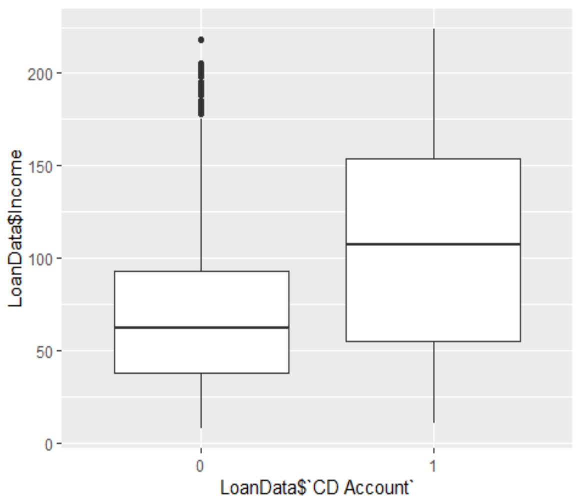
A few takeaways and insights from this univariate analysis are as follows:

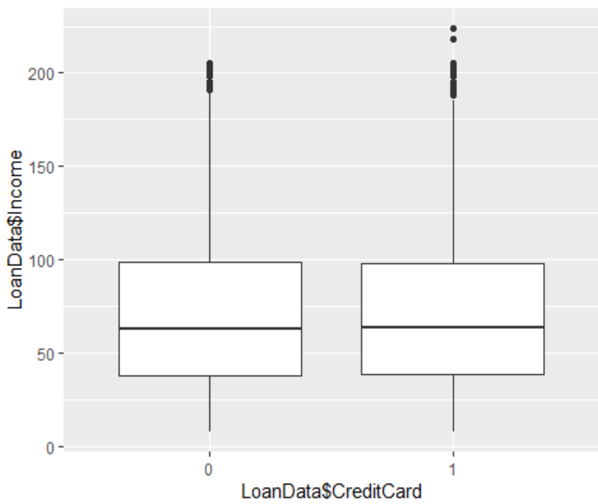
1. As expected, the data reveals that majority of people don’t have a mortgage. Additionally, we can also positively say there are outliers present in mortgage data i.e. the count of people having mortgage of upwards of $300,000 is very less but still present.
2. People with 40+ years of experience is relatively small.
3. Income, average monthly credit card spending & mortgage are all right skewed.
4. Undergraduates are relatively more than the graduates and professionals.
5. People with zero family members have the smallest count, while people with just one family member is relatively higher.

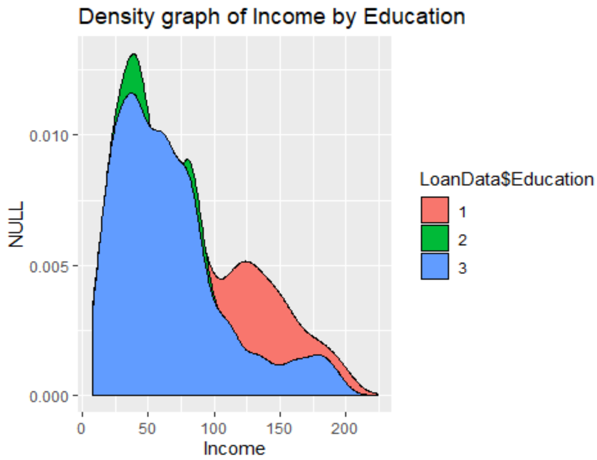
<u>BIVARIATE OR MULTIVARIATE ANALYSIS</u>

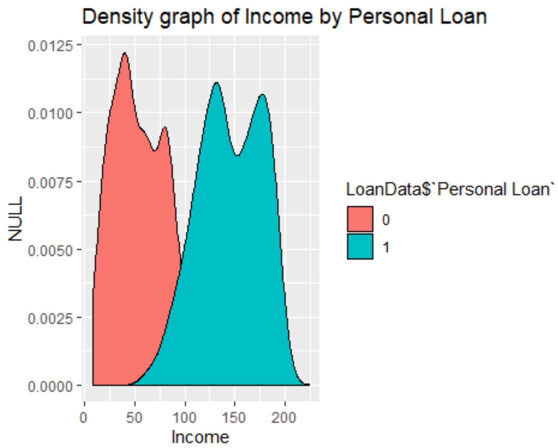


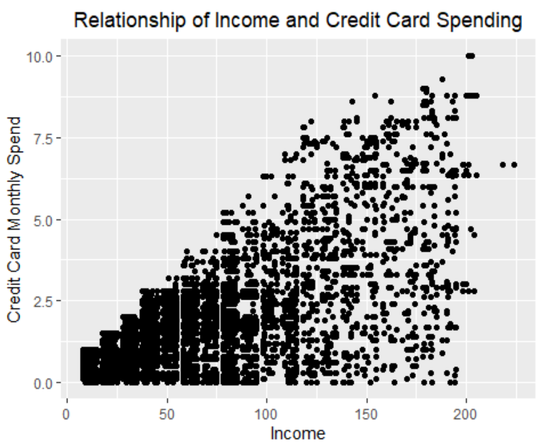


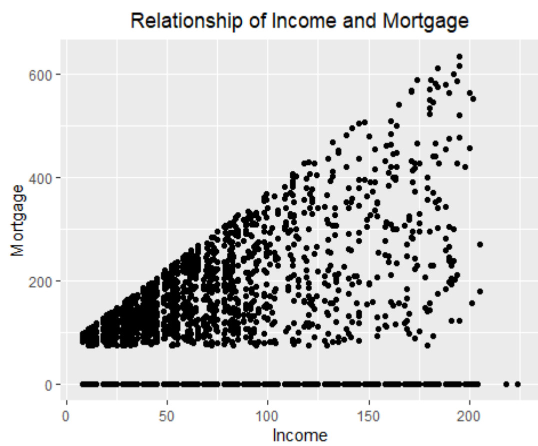


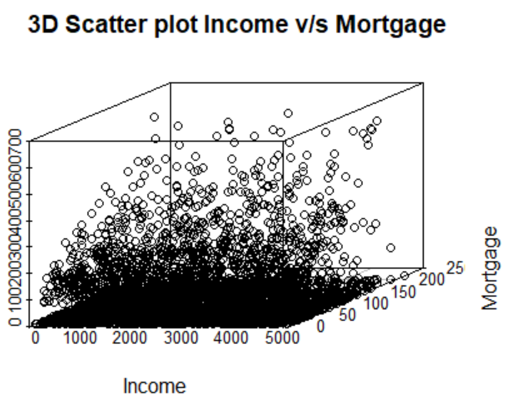


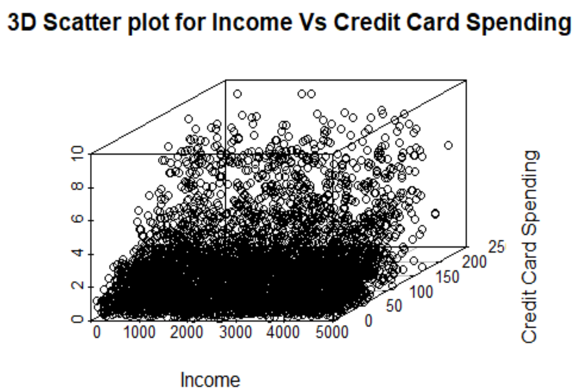

Density graph of Income by Education
NULL
0.000
0.005
0.010
0
50
100
150
200
Income
LoanData$Education
1
2
3

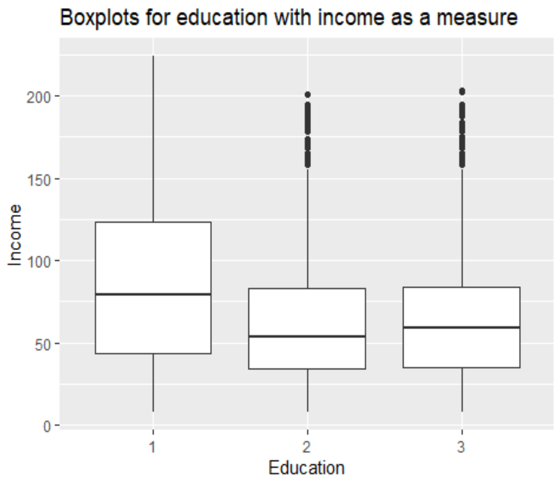

Density graph of Income by Personal Loan
NULL
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0
50
100
150
200
Income
LoanData$`Personal Loan`
0
1

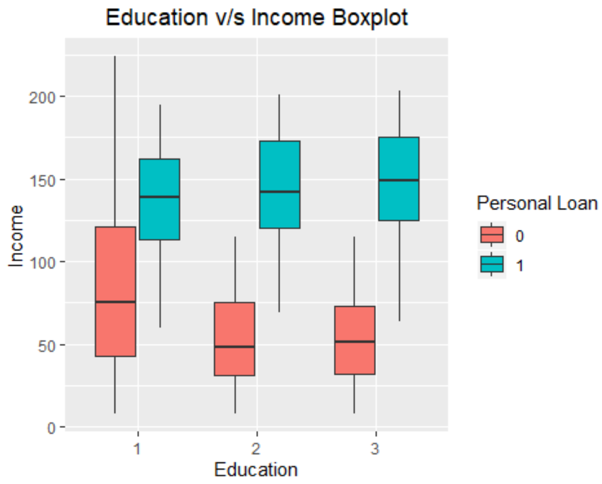

Relationship of Income and Credit Card Spending
Credit Card Monthly Spend
0.0
2.5
5.0
7.5
10.0
0
50
100
150
200
Income

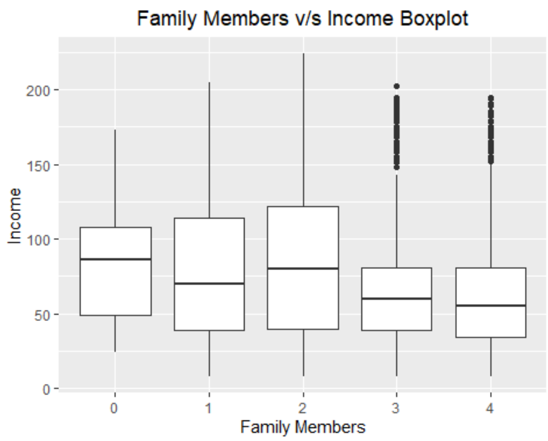

Relationship of Income and Mortgage
Mortgage
0
200
400
600
0
50
100
150
200
Income

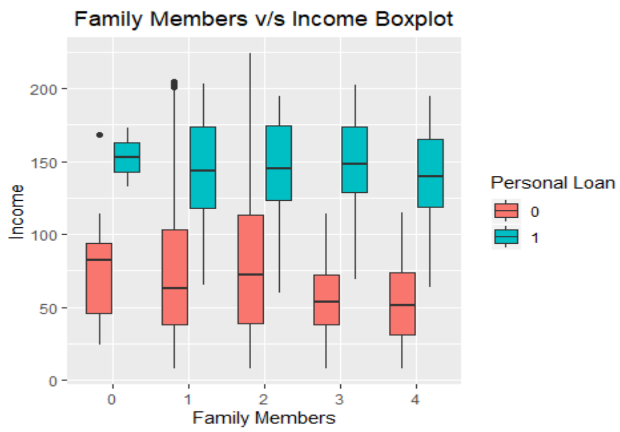

3D Scatter plot Income v/s Mortgage
0 100200300400500600700
0
1000
2000
3000
4000
5000
0
50
100
150
200
25
Mortgage
Income


3D Scatter plot for Income Vs Credit Card Spending
0
2
4
6
8
10
0
1000
2000
3000
4000
5000
0
50
100
150
200
25
Credit Card Spending
Income









INTERPRETATION

1. Income has a linear relationship with mortgage and average monthly credit card spending.
2. People who have a higher income generally are more prone to get a personal loan.

CLUSTERING (UNSUPERVISED LEARNING)

FEATURE SELECTION

```
LoanDataCluster <- LoanData[,c(1,2,3,5,7)]
head(LoanDataCluster)
```

```
##   Age Total Experience Income CCAvg Mortgage
## 1  25                1     49   1.6        0
## 2  45               19     34   1.5        0
## 3  39               15     11   1.0        0
## 4  35                9    100   2.7        0
## 5  35                8     45   1.0        0
## 6  37               13     29   0.4      155
```

We will be only using numeric values for our clustering mechanism in this exercise.

SCALING

```
Scaleddata <- scale(LoanDataCluster)
apply(Scaleddata, 2, mean)
```

```
##              Age Total Experience           Income            CCAvg
##     6.054575e-18     1.300502e-16     1.462101e-16     4.641524e-17
##         Mortgage
##    -2.653715e-17
```

```
apply(Scaleddata, 2, sd)
```

```
##              Age Total Experience           Income            CCAvg
##                1                1                1                1
##         Mortgage
##                1
```

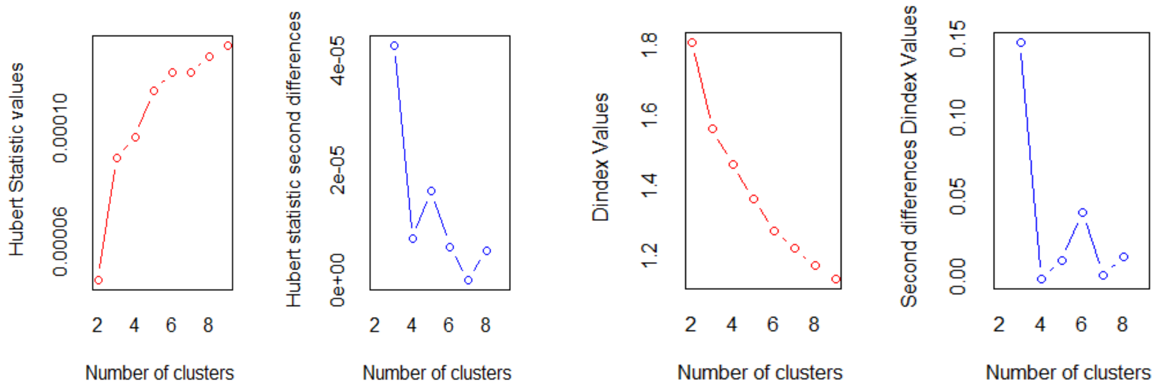
Here we apply scaling to our selected features. This is confirmed by checking the mean and standard deviation of our scaled data which is 0 and 1 respectively.

RECOMMENDING CLUSTERS

```
seed = 1000
set.seed(seed)
NBCLUSTER <- NbClust(Scaleddata, min.nc = 2, max.nc = 9, method = "ward.D"
)
```

```
## *******************************************************************
## * Among all indices:
## * 2 proposed 2 as the best number of clusters
## * 11 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 3 proposed 8 as the best number of clusters
## * 2 proposed 9 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
```

```
##
## *******************************************************************
```



The recommended number of optimal clusters is 3. (using Nbclust and ward.d method)

APPLICATION & INTERPRETATION

```
CustProfile <- aggregate(LoanDataCluster, list(LoanDataCluster$Cluster),
FUN = "mean")
```

```
##       Group.1 Age Total Experience     Income    CCAvg  Mortgage Cluster
## 1         1 35.11489          9.904832  60.15138 1.383412  44.74951       1
## 2         2 55.53604         30.233826  58.94177 1.367514  45.13494       2
## 3         3 43.68688         18.669554 147.69059 4.857463 116.42327       3
```

```
by(LoanDataCluster, INDICES = LoanDataCluster$Cluster, FUN = summary)
```

```
## LoanDataCluster$Cluster: 1
##        Age        Total Experience      Income           CCAvg
##  Min.   :23.00   Min.   : 0.000   Min.   :  8.00   Min.   :0.000
##  1st Qu.:30.00   1st Qu.: 5.000   1st Qu.: 35.00   1st Qu.:0.600
##  Median :35.00   Median :10.000   Median : 55.00   Median :1.300
##  Mean   :35.11   Mean   : 9.905   Mean   : 60.15   Mean   :1.383
##  3rd Qu.:40.00   3rd Qu.:15.000   3rd Qu.: 81.00   3rd Qu.:2.100
##  Max.   :46.00   Max.   :20.000   Max.   :194.00   Max.   :5.400
##     Mortgage       Cluster
##  Min.   :  0.00   1:2028
##  1st Qu.:  0.00   2:   0
##  Median :  0.00   3:   0
##  Mean   : 44.75
##  3rd Qu.: 90.00
##  Max.   :402.00
## --------------------------------------------------------
## LoanDataCluster$Cluster: 2
##        Age        Total Experience      Income           CCAvg
##  Min.   :45.00   Min.   :20.00    Min.   :  8.00   Min.   :0.000
##  1st Qu.:51.00   1st Qu.:25.00    1st Qu.: 33.00   1st Qu.:0.500
##  Median :56.00   Median :30.00    Median : 53.00   Median :1.300
##  Mean   :55.54   Mean   :30.23    Mean   : 58.94   Mean   :1.368
```

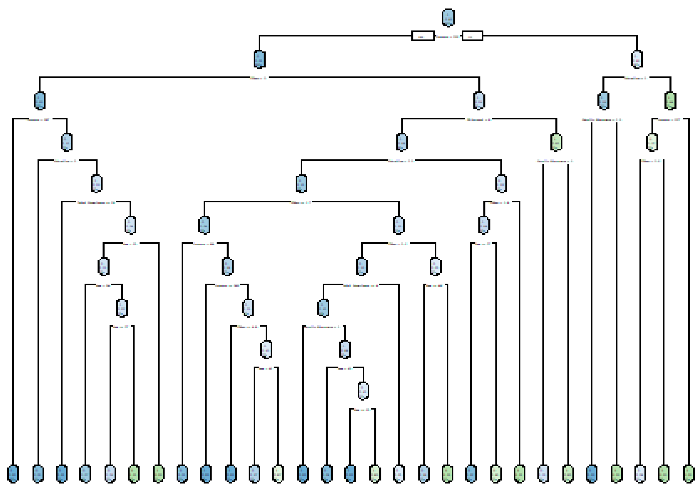
```
##  3rd Qu.:60.00   3rd Qu.:35.00    3rd Qu.: 80.00   3rd Qu.:2.000
##  Max.   :67.00   Max.   :43.00    Max.   :195.00   Max.   :4.900
##     Mortgage      Cluster
##  Min.   :  0.00   1:   0
##  1st Qu.:  0.00   2:2164
##  Median :  0.00   3:   0
##  Mean   : 45.13
##  3rd Qu.: 91.50
##  Max.   :427.00
## --------------------------------------------------------
## LoanDataCluster$Cluster: 3
##       Age        Total Experience     Income          CCAvg
##  Min.   :23.00   Min.   : 0.00    Min.   : 71.0   Min.   : 0.000
##  1st Qu.:36.00   1st Qu.:11.00    1st Qu.:125.0   1st Qu.: 3.400
##  Median :44.00   Median :19.00    Median :149.0   Median : 4.700
##  Mean   :43.69   Mean   :18.67    Mean   :147.7   Mean   : 4.857
##  3rd Qu.:51.00   3rd Qu.:25.00    3rd Qu.:173.2   3rd Qu.: 6.348
##  Max.   :65.00   Max.   :41.00    Max.   :224.0   Max.   :10.000
##     Mortgage     Cluster
##  Min.   :  0.0   1:  0
##  1st Qu.:  0.0   2:  0
##  Median :  0.0   3:808
##  Mean   :116.4
##  3rd Qu.:237.2
##  Max.   :635.0
```

CLASSIFICATION (SUPERVISED LEARNING)

CART MODEL (PLOT)

```
dim(Trainset)

## [1] 3500   12

sum(Trainset$`Personal Loan` == 1)/nrow(Trainset)*100
```

```
## [1] 9.6

sum(Trainset$`Personal Loan` == 0)/nrow(Trainset)*100

## [1] 90.4

CARTtree <- rpart(formula = Trainset$`Personal Loan` ~ . , data = Trainset
, method = "class",  minbucket = 3, cp = 0)

rpart.plot(CARTtree)
```



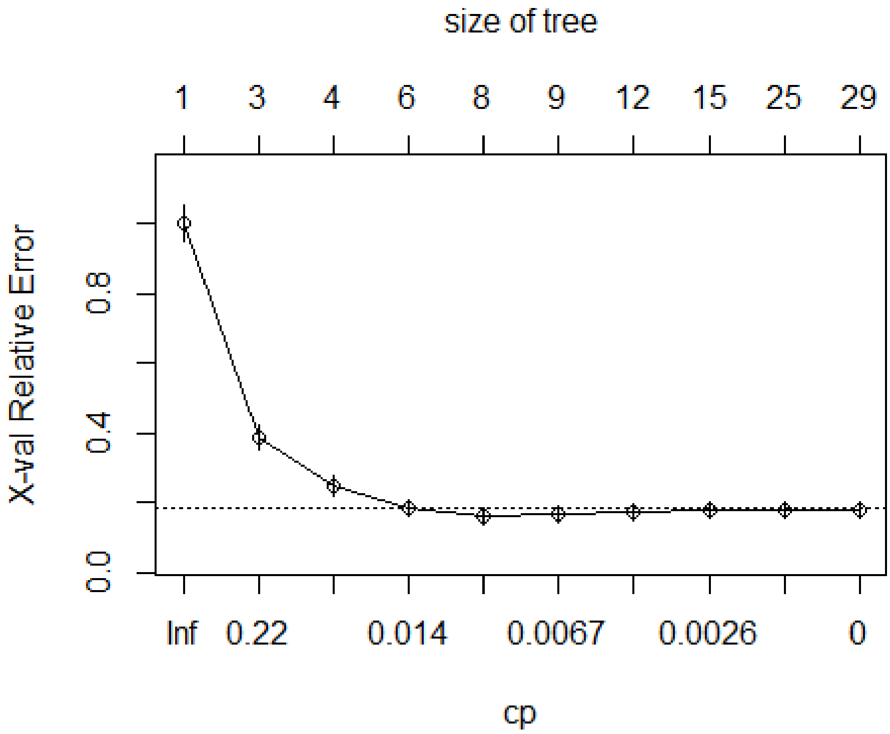
Using a 0 complexity parameter, we obtain a complex tree with 14 splits. The tree above is overfitting the data.

PRUNING

```
printcp(CARTtree)

##            CP nsplit rel error  xerror     xstd
## 1  0.3377976      0  1.000000 1.00000 0.051870
## 2  0.1398810      2  0.324405 0.38690 0.033298
## 3  0.0178571      3  0.184524 0.24702 0.026791
## 4  0.0104167      5  0.148810 0.18155 0.023041
## 5  0.0089286      7  0.127976 0.16071 0.021701
## 6  0.0049603      8  0.119048 0.16667 0.022093
## 7  0.0029762     11  0.104167 0.17262 0.022477
## 8  0.0022321     14  0.095238 0.17560 0.022667
## 9  0.0014881     24  0.068452 0.17560 0.022667
## 10 0.0000000     28  0.062500 0.17857 0.022855
```

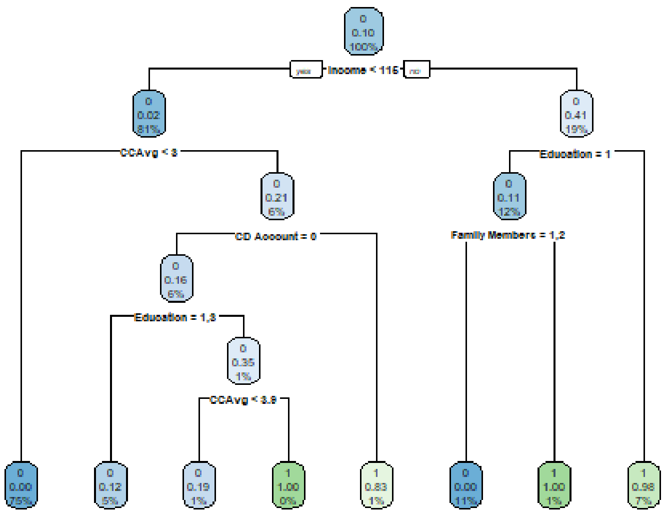
After printing out the complexity chart of our tree, it can be seen that the cross validation error (xerror) decreases till 7 splits, where xerror is 0.16071 and then starts increasing again. This is our cue that maybe, 7 or 8 splits is optimal. So for pruning, we may want to choose a C.P of 0.009.

```
plotcp(CARTtree)
```




```
PrunedTree <- prune(CARTtree, cp = 0.0090, "CP" )
PrunedTree

## n= 3500
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 3500 336 0 (0.904000000 0.096000000)
##    2) Income< 114.5 2827  57 0 (0.979837283 0.020162717)
##      4) CCAvg< 2.95 2613  11 0 (0.995790279 0.004209721) *
##      5) CCAvg>=2.95 214  46 0 (0.785046729 0.214953271)
##       10) CD Account=0 196  31 0 (0.841836735 0.158163265)
##         20) Education=1,3 162  19 0 (0.882716049 0.117283951) *
##         21) Education=2 34  12 0 (0.647058824 0.352941176)
##           42) CCAvg< 3.85 27   5 0 (0.814814815 0.185185185) *
##           43) CCAvg>=3.85 7   0 1 (0.000000000 1.000000000) *
##       11) CD Account=1 18   3 1 (0.166666667 0.833333333) *
##    3) Income>=114.5 673 279 0 (0.585438336 0.414561664)
```

```
##      6) Education=1 436  47 0 (0.892201835 0.107798165)
##       12) Family Members=1,2 389   0 0 (1.000000000 0.000000000) *
##       13) Family Members=3,4 47   0 1 (0.000000000 1.000000000) *
##      7) Education=2,3 237   5 1 (0.021097046 0.978902954) *

rpart.plot(PrunedTree)
```



This pruned tree yields the lowest cross-validated error. By looking at the tree we can also predict what kind of customer profile is more suited to purchasing a personal loan. Income is the predictor variable used for the primary split

We can suggest that people with an annual income of more than $116,000 and education level of graduate or working advanced professional has more chance of opting for a personal loan.

We may also suggest that people with an income less than $116,000 and an average monthly credit card spending of more than $3000 who has a C.D account with the bank has good chance of opting for a personal loan.

PREDICTION

```
Trainset$Pred <- predict(PrunedTree, data = Trainset, type = "class")
Trainset$Prob <- predict(PrunedTree, data = Trainset, type = "prob")[,"1"]
```

```
dim(Trainset)

## [1] 3500   14
```

```
Testset$Prediction <- predict(PrunedTree, Testset, type = "class")
Testset$Probability <- predict(PrunedTree, Testset, type = "prob")[,-1]
```

```
dim(Testset)

## [1] 1500   14
```

MODEL PERFORMANCE & INTERPRETATION

	CART	
	Train	Test
Confusion Matrix Accuracy	98.77%	98.07%
K.S	91.26	92.43
Gini	0.8718	0.8752
AUC	0.9822	0.9834
Concordance	0.9668	0.9693

CART model seems to be performing good after looking at all the model performance measures. Its in a healthy zone of it being neither over-fit or under-fit.

However, if we were to scrutinize this further and have an extremely small threshold when comparing model validations on both in-sample and out-sample data we may further say that, the confusion matrix accuracy of the model seems to be slightly overfitting the data (By 0.70 % to be precise) and all other model validations (K.S, gini, auc and concordance) are slightly under-fit.

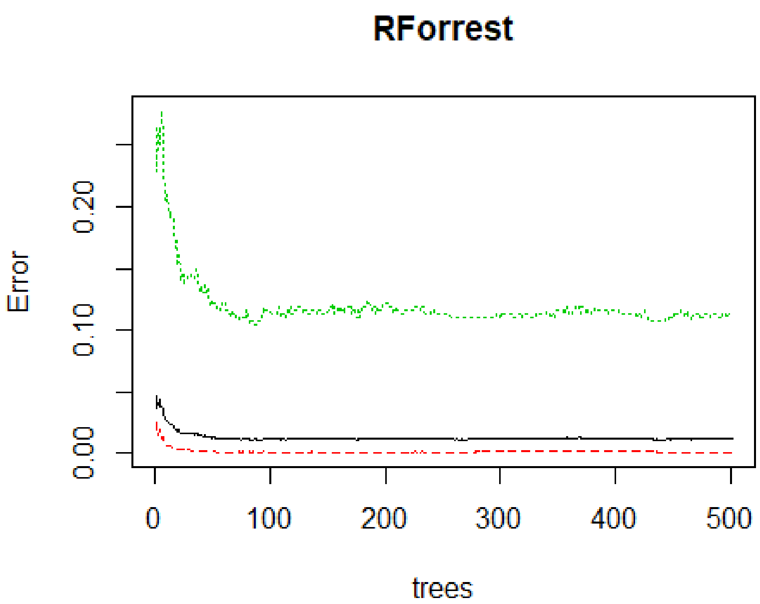
RANDOMFORREST MODEL (PLOT)

```
sum(Trainset$`Personal Loan` == 1)/nrow(Trainset)

## [1] 0.096

sum(Trainset$`Personal Loan`== 0)/nrow(Trainset)

## [1] 0.904

seed = 1000
set.seed(1000)
attach(Trainset)
RForrest <- randomForest(Trainset$`Personal Loan`~., data = Trainset, ntre
e = 501, mtry = 3, nodesize = 10, importance = TRUE)


RForrest

##
## Call:
##  randomForest(formula = Trainset$`Personal Loan` ~ ., data = Trainset,
ntree = 501, mtry = 3, nodesize = 10, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 501
```

```
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.17%
```

```
## Confusion matrix:
##      0   1  class.error
## 0 3161   3 0.0009481669
## 1   38 298 0.1130952381
```

```
plot(RForrest)
```



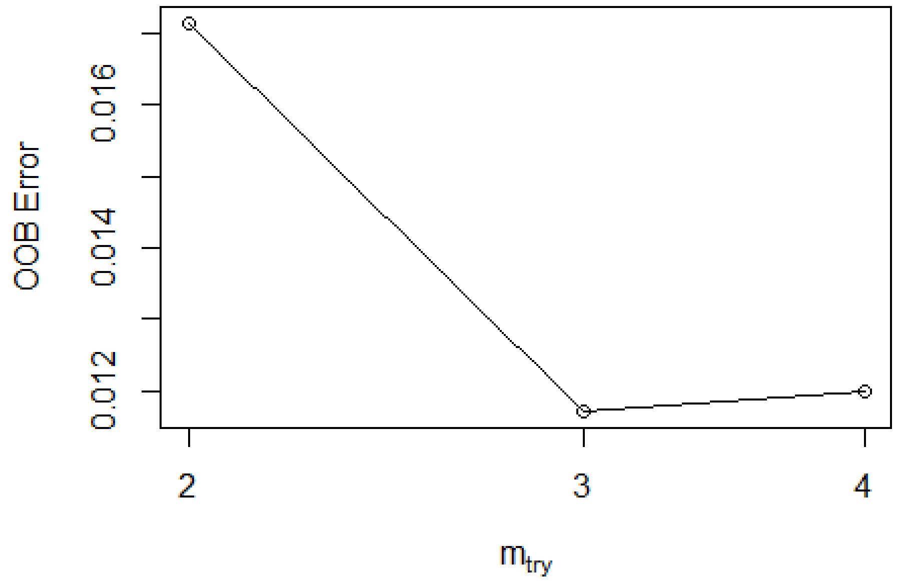
After plotting the random Forrest model, we observe that there is no visible improvement in the OOB roughly past 80 trees. So we use an odd number of 77 as our number of trees for tuning random Forrest model for optimal number of mtry.

<u>TUNING RANDOMFORREST MODEL</u>

```
set.seed(seed)
TRFOrrest <- tuneRF(x = Trainset[,-8], y = Trainset$`Personal Loan`, mtryS
tart = 3, nodesize = 10,
                    stepFactor = 1.5, ntreeTry = 77, improve = 0.0001, tra
ce = TRUE,
                    plot = TRUE, doBest = TRUE, importance = TRUE)
```

```
## mtry = 3  OOB error = 1.17%
## Searching left ...
## mtry = 2     OOB error = 1.71%
## -0.4634146 1e-04
## Searching right ...
```

```
## mtry = 4     OOB error = 1.2%
## -0.02439024 1e-04
```




```
TRFOrrest

##
## Call:
##  randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1],      no
desize = 10, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.17%
## Confusion matrix:
##      0   1  class.error
## 0 3161   3 0.0009481669
## 1   38 298 0.1130952381
```

There are three type-1 or false positives and 38 type-2 or true negatives in the confusion matrix of this tuned randomforrest.

PREDICTION

```
Trainset$PredictionF <- predict(TRFOrrest, Trainset, type = "class" )
Trainset$Probability <- predict(TRFOrrest, Trainset, type = "prob" )[,1]
dim(Trainset)
## [1] 3500   14
```

```
Testset$Pred <- predict(TRFOrrest, Testset, type = "class")
Testset$Prob <- predict(TRFOrrest, Testset, type = "prob")[,-1]
dim(Testset)
## [1] 1500   14
```

MODEL PERFORMANCE & INTERPRETATION

	RANDOMFORREST	
	Train	Test
Confusion Matrix Accuracy	99.28%	98.06%
K.S	98.83%	96.72%
Gini	0.9103	0.8845
AUC	0.9998	0.998
Concordance	0.9998	0.998

Random forrest model seems to be performing as expected with great accuracy both on in-sample and out-sample data. Though it may be overfitting in a couple of performance measures but that would be a call business has to tae. As per my understanding there should be no discussion that this model is neither overfitting the data or under fitting it.

CONCLUSION

Although CART model is outperforming randomforrest on confusion matrix validation parameter by 0.1% i.e accuracy on test set using CART model is 98.07% and accuracy on test set using randomforrest model is 98.06%, the randomforrest model supremely outperforms, as expected, when compared to CART model on other model evaluation parameters.

It is safe to assume that we may use randomforrest model for our future evaluations because of its accuracy and CART model for its ability to interpolating the decision tree.

CLOSING REMARKS

We may use the insights from the CART model and the plotted dendogram for its ability to target a particular group of customers who have a higher probability to respond to the personal loan campaign and hence avoid incurring any additional redundant costs. This will be useful when strategizing for future marketing campaigns.