04-03-2020

DATA MINING PROJECT

THERA BANK

PRESENTED BY: SHILPA GIRIDHAR

CONTENTS

THERA BANK – PERSONAL LOAN CAMPAIGN 2
DESCRIPTION 2
Business Case 2
DATA EXPLORATION 3
1. EDA - Basic data summary, Univariate, Bivariate analysis, graphs 3
2. CART MODEL 9
2.1. Applying CART <plot the tree> - Decision Tree 9
2.2. Interpret the CART model output <pruning, remarks on pruning, plot the pruned tree> 9
3. K-Means Clustering 15
4. Random Forest 16

DESCRIPTION

Thera Bank - Loan Purchase Modeling

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_Personal_Loan_Modelling-dataset-1.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 - **(10 Marks)**
- 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) - **(30 Marks)**
- 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. - **(20 Marks)**
- **Hint :** split <- sample.split(Thera_Bank$Personal Loan, SplitRatio = 0.7)
  #we are splitting the data such that we have 70% of the data is Train Data and 30% of the data is my Test Data

```
train<- subset(Thera_Bank, split == TRUE)
test<- subset( Thera_Bank, split == FALSE)
```

BUSINESS CASE

Bank needs to identify best cluster of people who are likely to accept the personal loan offered to them in a campaign. It is understood that the bank's earlier campaign on all the 5000 customers resulted in only 480 (= 9.6%) who accepted the personal loan. So, the objective is to build an accurate model that classifies the subset of customers who are likely to accept the loan offer and target the campaign to these customers. This case is example of classification, as we are trying group and target certain type of customers for achieving the object. where the target variable is categorical (those are likely to say “yes” to the loan offer or “no” to the loan offer and the tree is used to identify the “class of people” who are likely fall into one of these categories.

DATA EXPLORATION

1. EDA - BASIC DATA SUMMARY, UNIVARIATE, BIVARIATE ANALYSIS, GRAPHS

Univariate (Continuous or Categorical Variables)

- Use Histogram, Boxplot, Density plot for continuous variable
- Use Barplot for categorical variables
- No of plots = No of variables

Bivariate

- both variables are categorical - Barplot

Univariate Analysis

– We have a mix of integer, numeric, and factor variables.

ID	Customer ID
Age	Customer's age in years
Experience	Years of professional experience
Income	Annual income of the customer ($000)
ZIPCode	Home Address ZIP code.
Family	Family size of the customer
CCAvg	Avg. spending on credit cards per month ($000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. ($000)
Personal Loan	Did this customer accept the personal loan offered in the last campaign?
Securities Account	Does the customer have a securities account with the bank?
CD Account	Does the customer have a certificate of deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by the bank?

– We use Data Summary to analyse the data structure

– There are 5000 rows, 14 columns in the "trainDS2" dataframe; There are missing values in certain columns

– Omitting the rows with missing values, we get a refined dataframe "trainDS1", in which there are 4982 rows, 14 columns

– Confirming that there are no missing values in the refined dataframe "trainDS1"

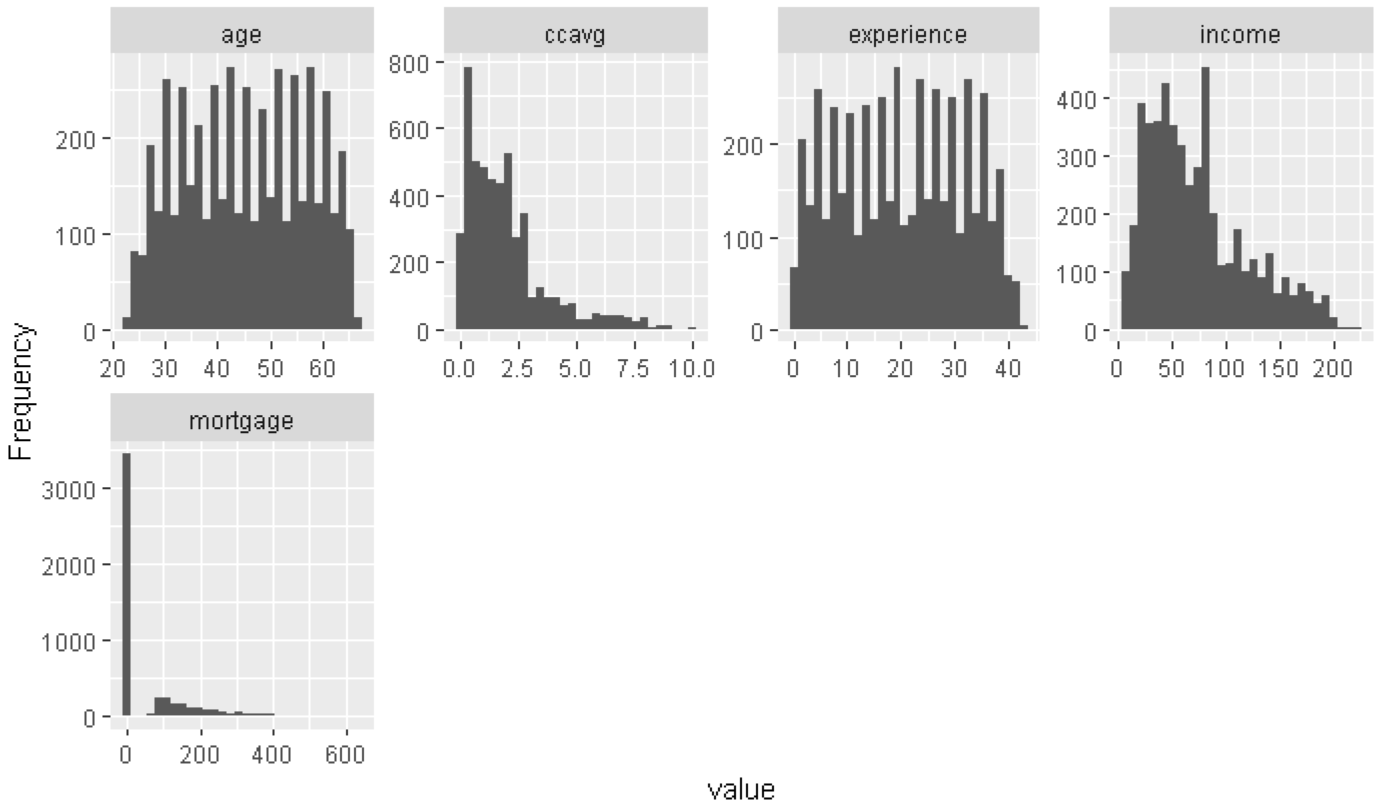
– Summary command shows that

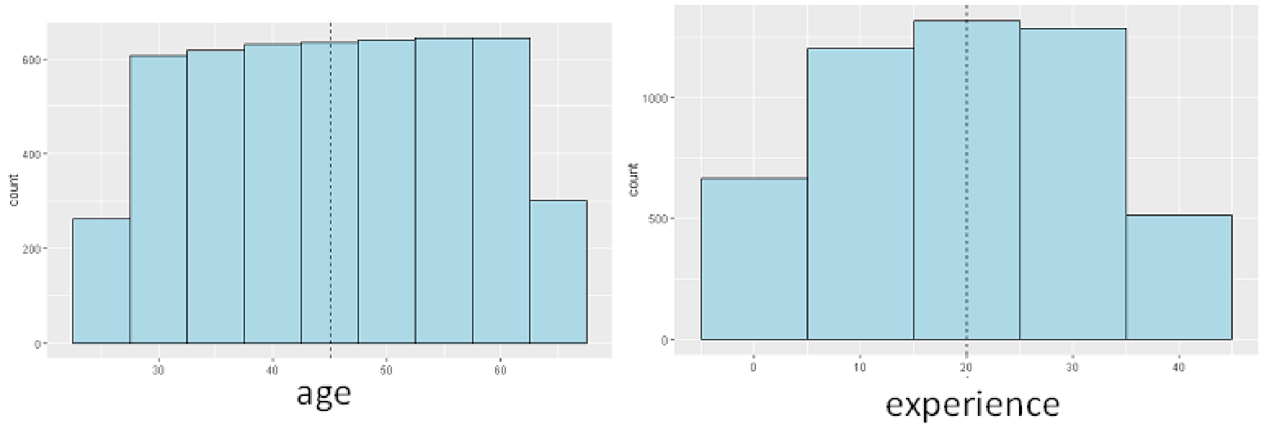
- Age ranges from 23 to 67, with Mean = 45 years
- Experience column has -ve values, which is invalid and is fixed using the 'abs' command
- Income ranges from $8000 to $2,24,000 per month
- Family members ranges from 1 to 4
- credit card spends per month ranges from 0 to $10,000 per month, with mean $1940
- House Mortgage ranges from 0 to $6,35,000; a lot of data people do not have a mortgage loan- so we exclude the values with zero while plotting

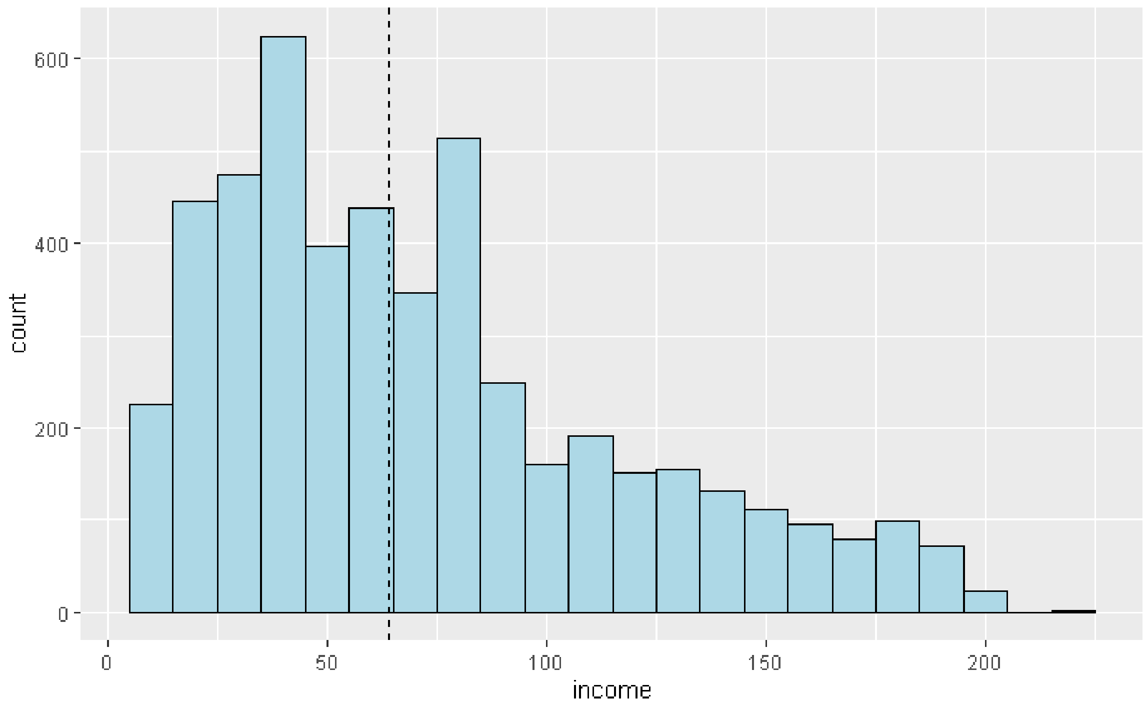
– Omitting the columns ID and fixing the -ve value in Experience column, we get new dataframe "trainDS" , in which there are 4982 rows, 13 columns

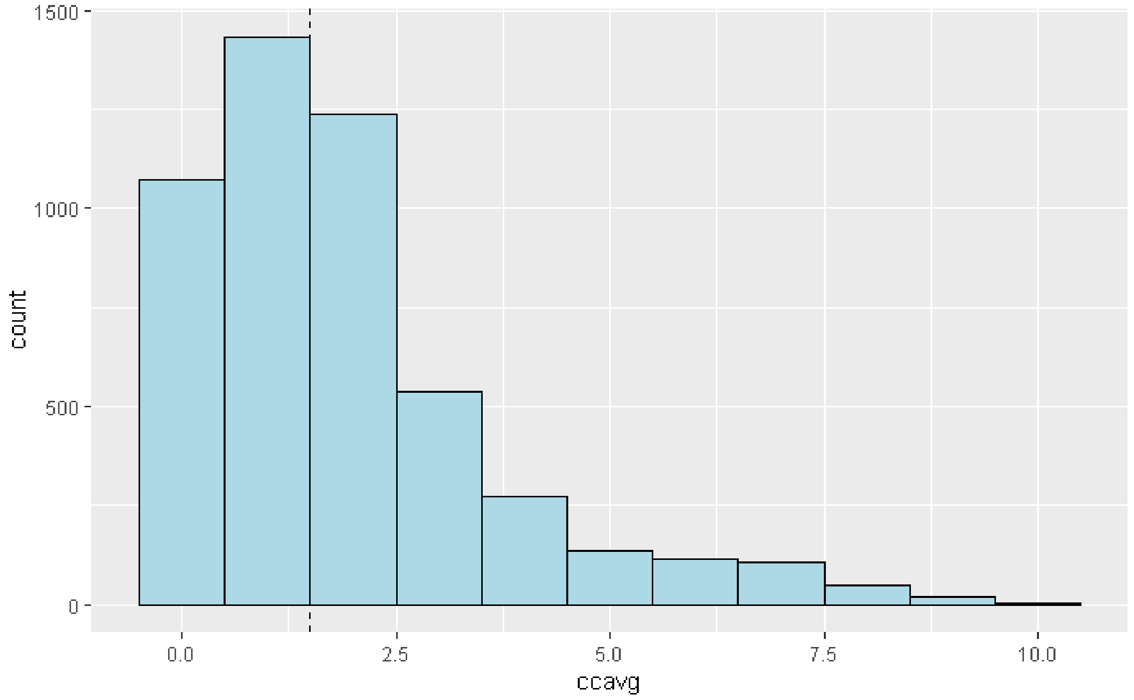
- Personal Loan column can be treated as the dependent target variable and rest 12 are independent variables

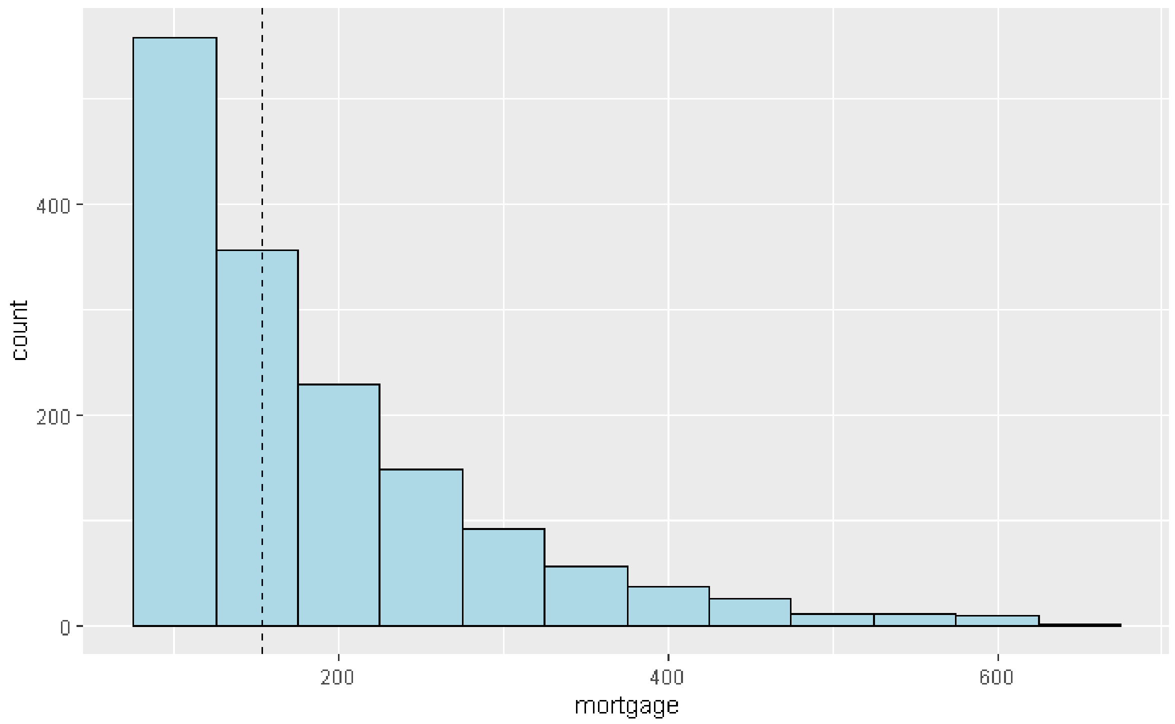
- We can use Histogram to plot continuous variables as shown below -





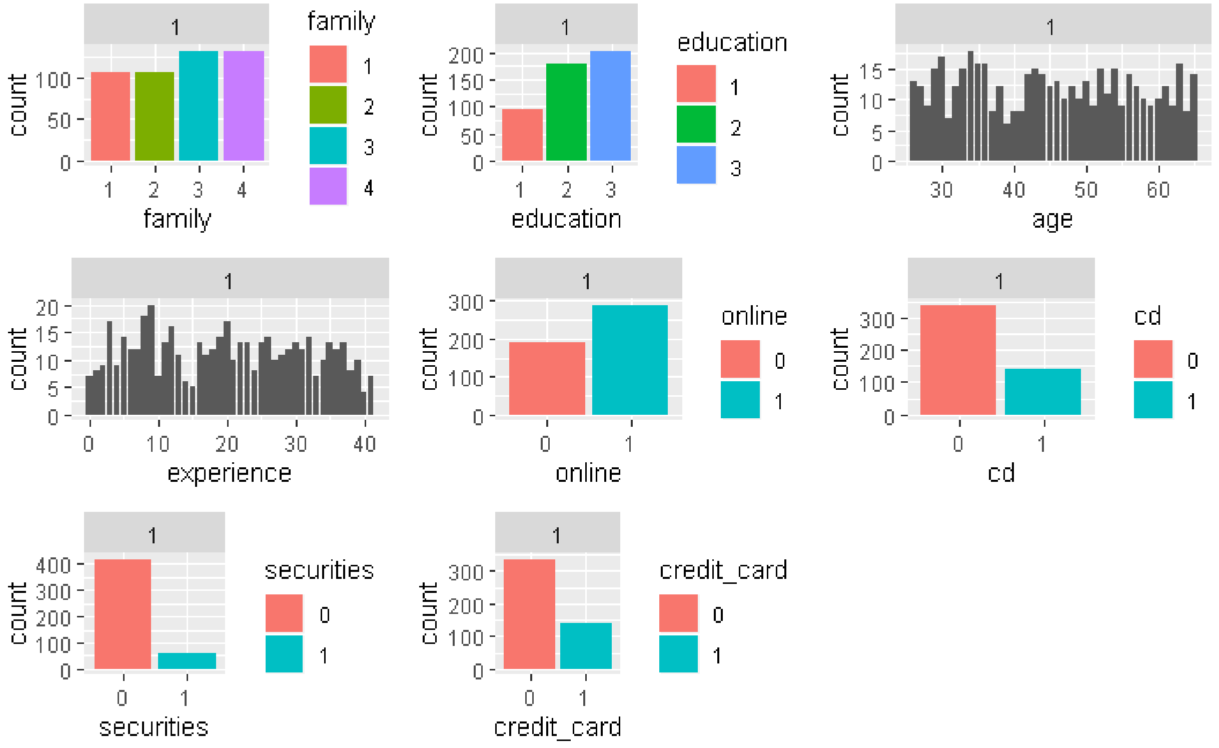

count
600
400
200
0
0
50
100
150
200
income


count
1500
1000
500
0
0.0
2.5
5.0
7.5
10.0
ccavg



Bivariate Analysis

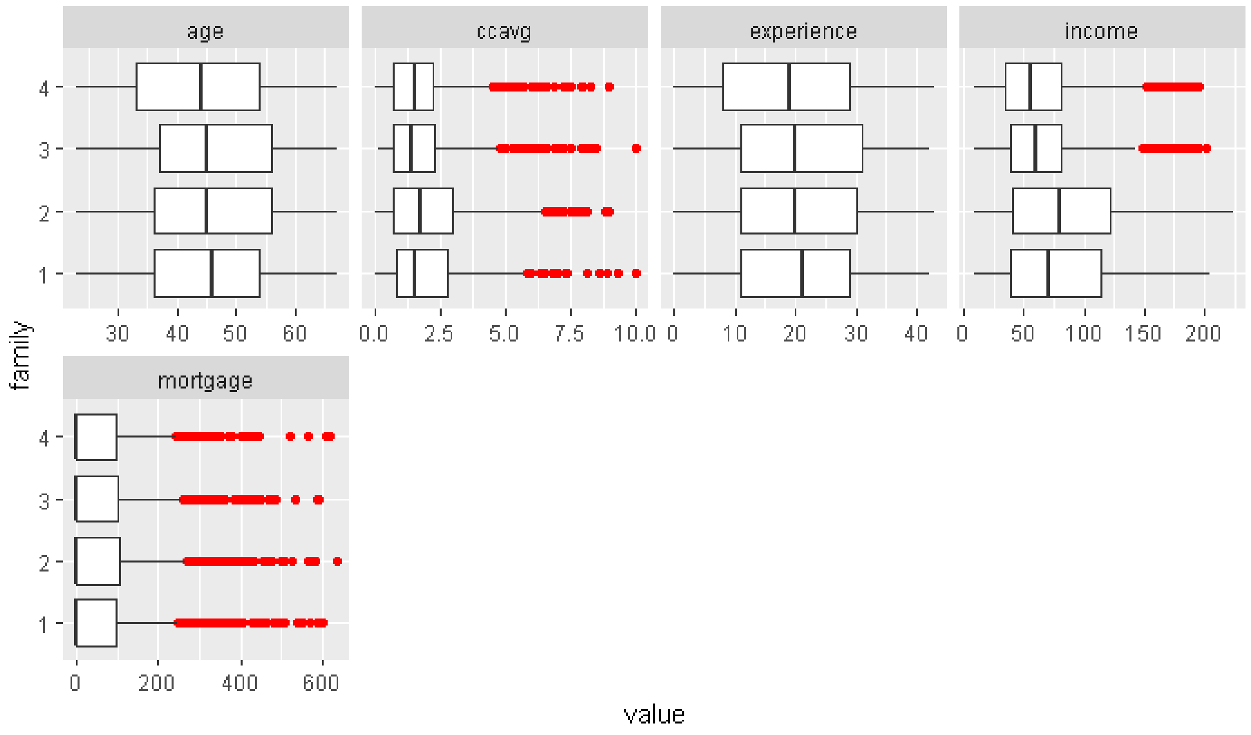
Plotting the No of people who showed interest in personal loan (Personal loan==1) offer - by family size , by education, experience, age, online, cd, securities, and credit card.
Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
It is clear that advanced professional have shown better interest than other education categories. Online method of loan offer seems to have more acceptance.



Find outliers using boxplot

```
## Using boxplot to find outliers
plot_boxplot(trainDS, by = "family", geom_boxplot_args = list("outlier.color" = "red"))
```

We can see from the box plots that “ccavg”, “income” and the “mortgage” columns have outliers



2. CART MODEL

2.1. APPLYING CART <PLOT THE TREE> - DECISION TREE

2.2. INTERPRET THE CART MODEL OUTPUT <PRUNING, REMARKS ON PRUNING, PLOT THE PRUNED TREE>

Next, the data is split into training and test subsets, and a decision tree is built on the training data.

Function rpart() is used to build a decision tree, and the tree with the minimum prediction error is selected.

After that, it is applied to new data to make prediction with function predict().

```
set.seed(1300)

## sampling 70% of data for training the algorithms using random sampling
trainDS.index = sample(1:nrow(trainDS), nrow(trainDS)*0.70)
p_train = trainDS[trainDS.index,]
p_test = trainDS[-trainDS.index,]
nrow(p_train)
nrow(p_test)
dim(p_train)
dim(p_test)
```

```
> dim(p_train)
[1] 3487   13
> dim(p_test)
[1] 1495   13
```

Checking if the proportion are right

p_train Dataset

```
         0          1
0.90450244 0.09549756
```

p_test Dataset

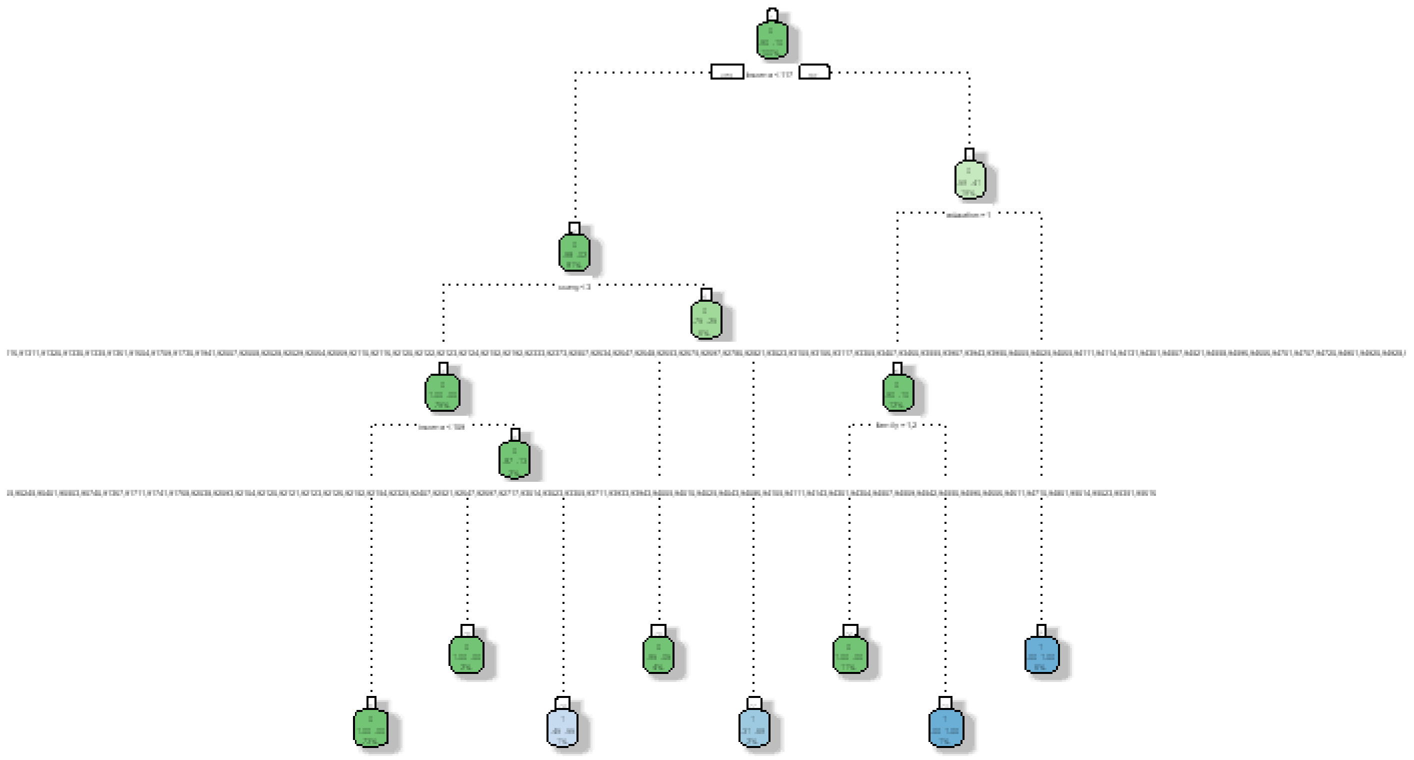
```
         0          1
0.90301003 0.09698997
```

Set the rpart parameters - >

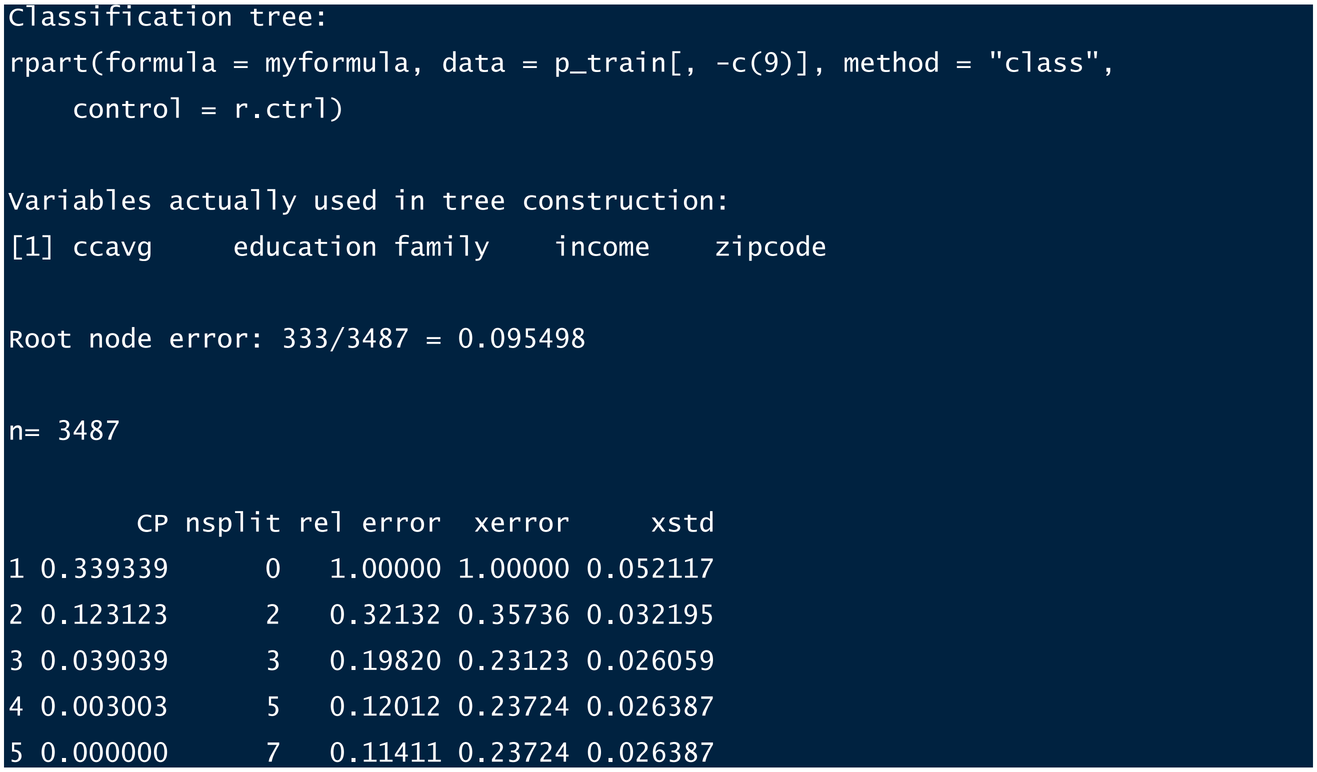
Min-bucket = Minimum no of observations that must be present in a leaf node

Min-split = 3 * Min-bucket = Minimum no of observations that must be present in a node for it to be considered for a split.

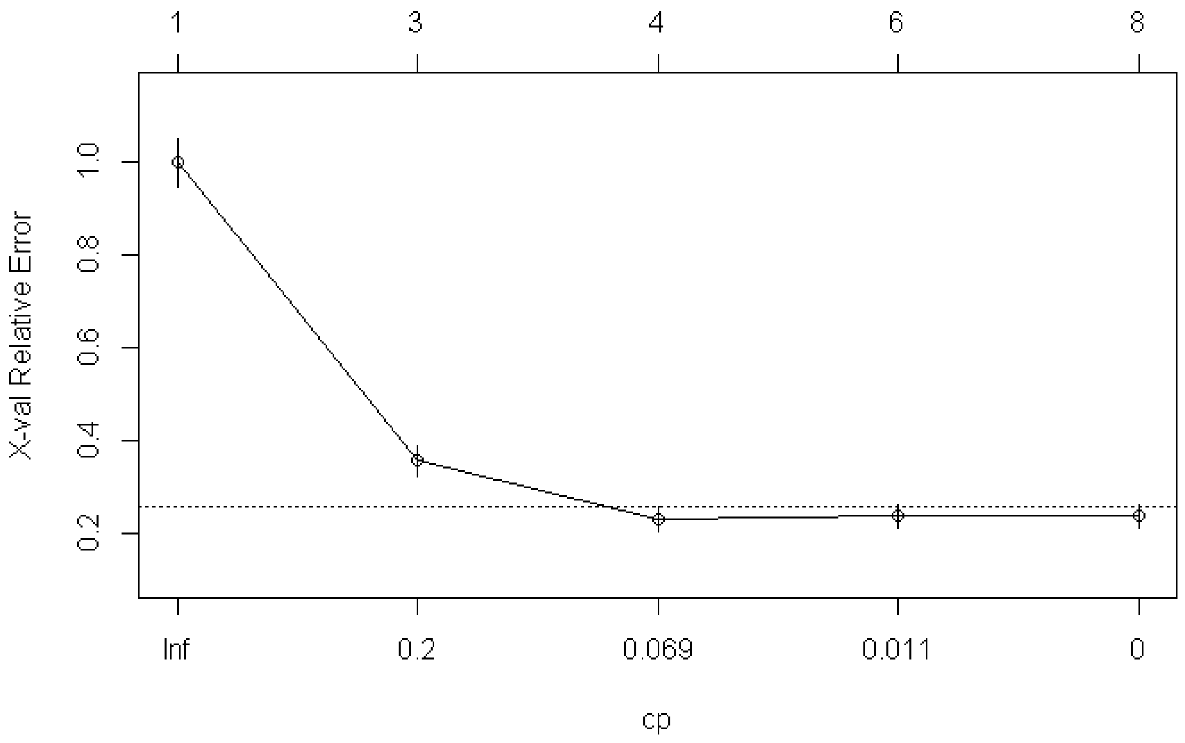
cp = how much is the error reducing, is the split helping to reduce the clutter



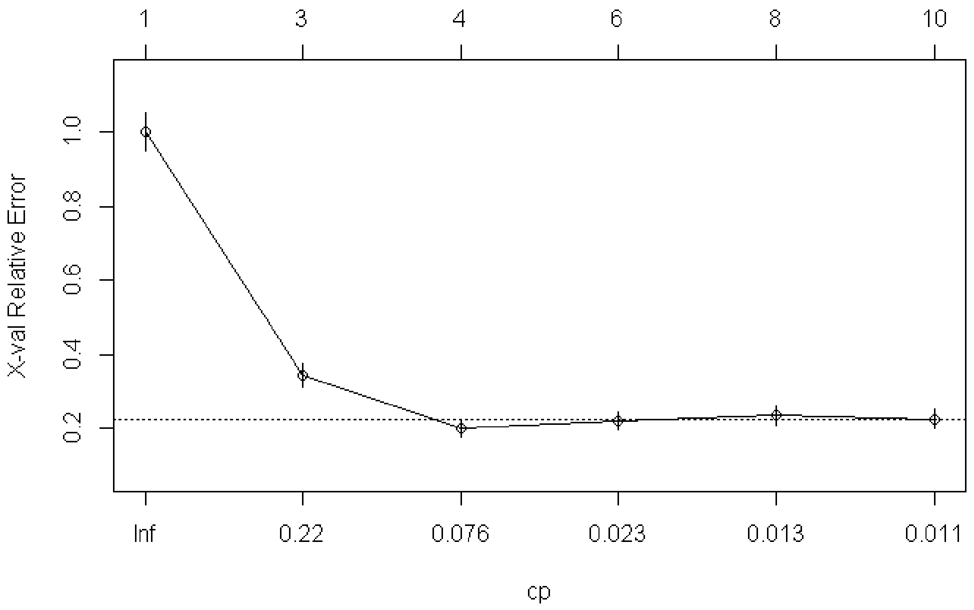
Rattle 2020-Mar-04 20:41:48 Shilpa

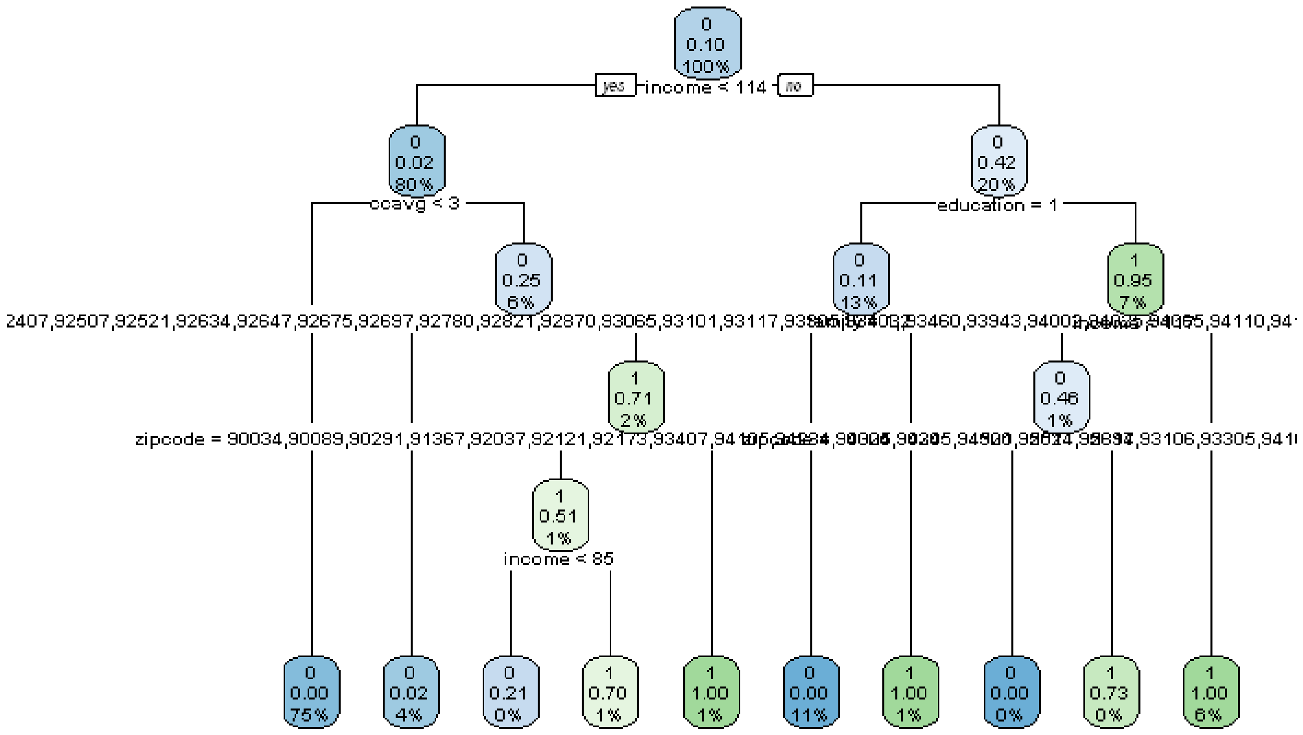


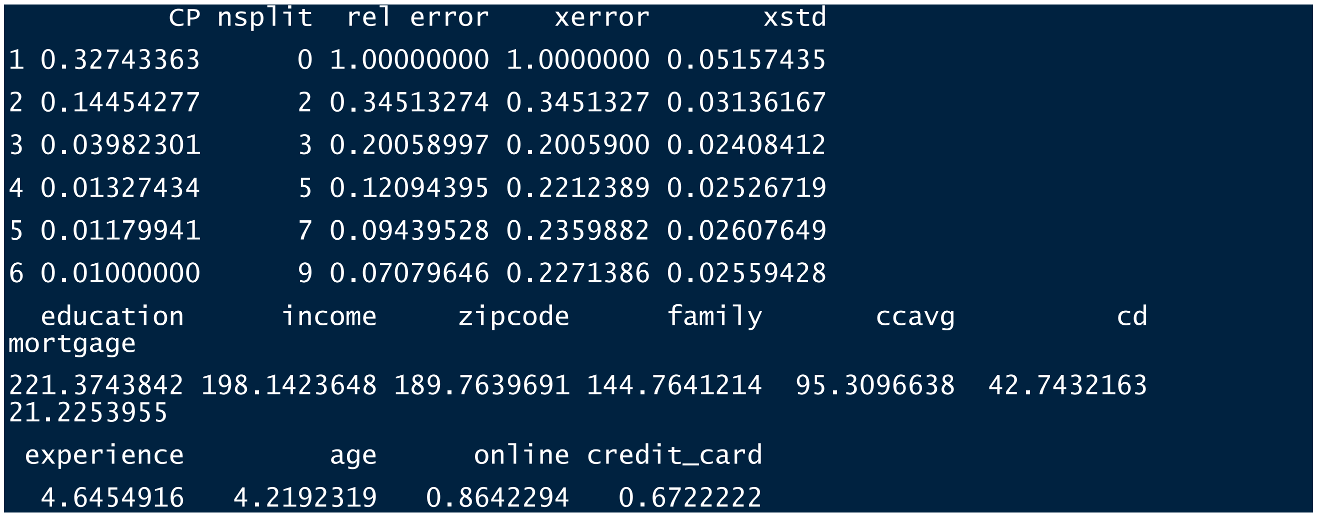
```
Classification tree:
rpart(formula = myformula, data = p_train[, -c(9)], method = "class",
    control = r.ctrl)

Variables actually used in tree construction:
[1] ccavg     education family    income    zipcode

Root node error: 333/3487 = 0.095498

n= 3487

        CP nsplit rel error  xerror     xstd
1 0.339339      0   1.00000 1.00000 0.052117
2 0.123123      2   0.32132 0.35736 0.032195
3 0.039039      3   0.19820 0.23123 0.026059
4 0.003003      5   0.12012 0.23724 0.026387
5 0.000000      7   0.11411 0.23724 0.026387
```



Using Gini Method







```
          CP nsplit  rel error    xerror       xstd
1 0.32743363      0 1.00000000 1.0000000 0.05157435
2 0.14454277      2 0.34513274 0.3451327 0.03136167
3 0.03982301      3 0.20058997 0.2005900 0.02408412
4 0.01327434      5 0.12094395 0.2212389 0.02526719
5 0.01179941      7 0.09439528 0.2359882 0.02607649
6 0.01000000      9 0.07079646 0.2271386 0.02559428
  education      income     zipcode      family       ccavg          cd
mortgage
221.3743842 198.1423648 189.7639691 144.7641214  95.3096638  42.7432163
21.2253955
 experience         age      online credit_card
  4.6454916   4.2192319   0.8642294   0.6722222
```

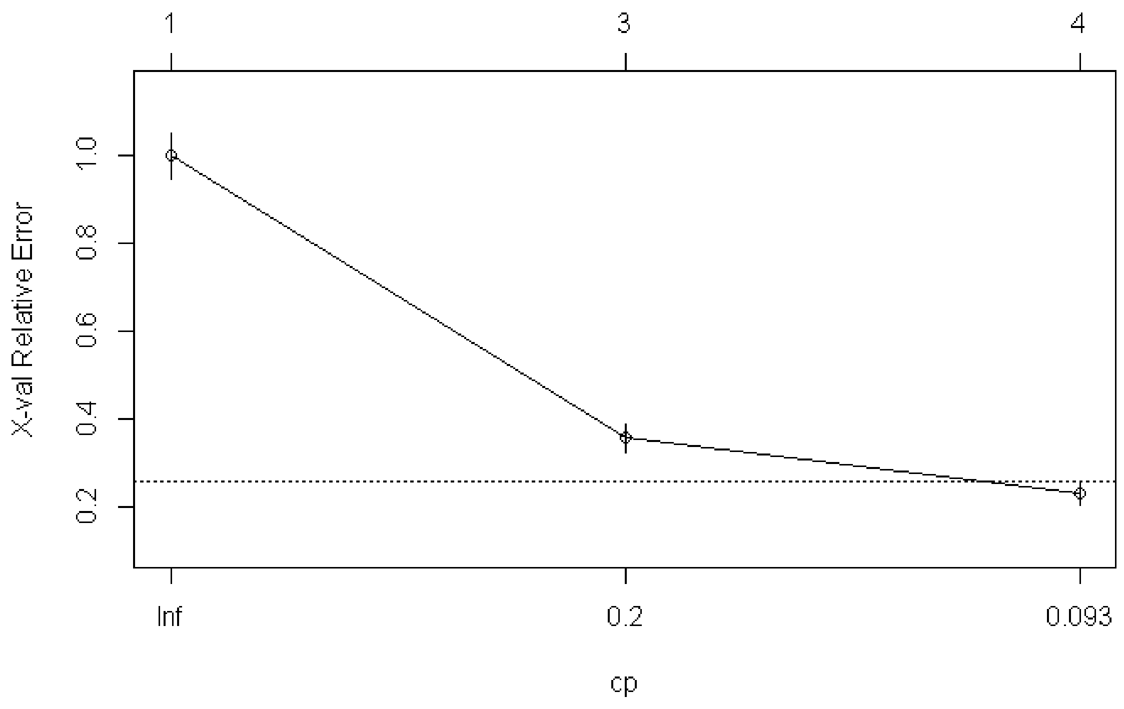
Pruning the tree

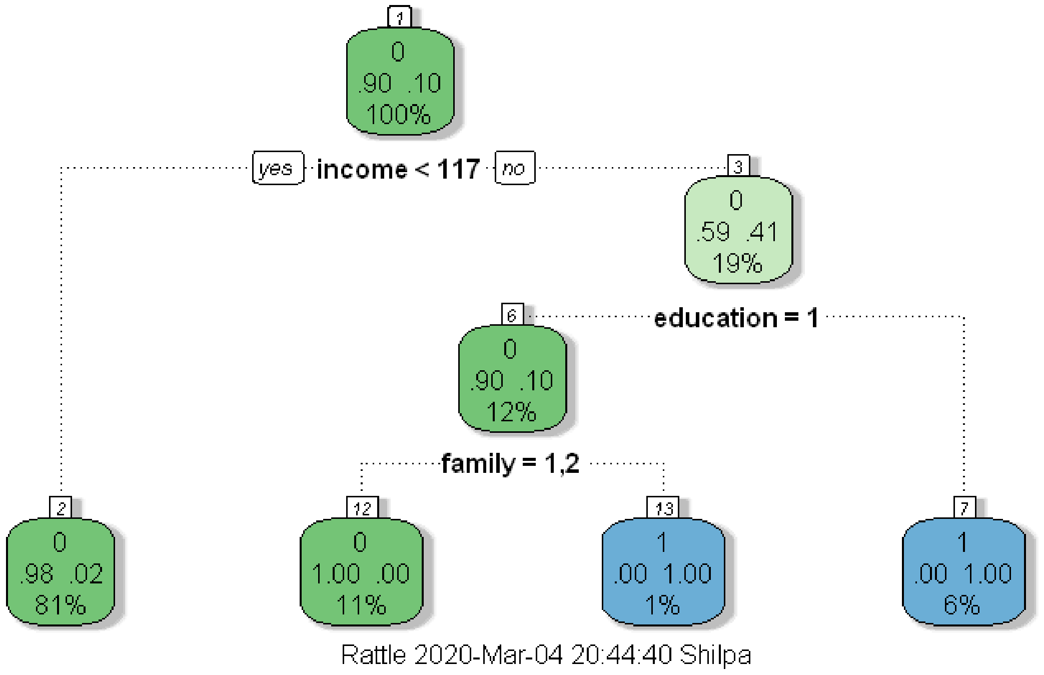
## Pruning the tree has started

## We are considering 0.07 as the pruned parameter and rebuild the tree

ptree<- prune(m2, cp= 0.07 ,"CP")
printcp(ptree)
plotcp(ptree)

```
Classification tree:
rpart(formula = myformula, data = p_train[, -c(9)], method = "class",
    control = r.ctrl)

Variables actually used in tree construction:
[1] education family    income

Root node error: 333/3487 = 0.095498

n= 3487

       CP nsplit rel error  xerror     xstd
1 0.33934      0   1.00000 1.00000 0.052117
2 0.12312      2   0.32132 0.35736 0.032195
3 0.07000      3   0.19820 0.23123 0.026059
```





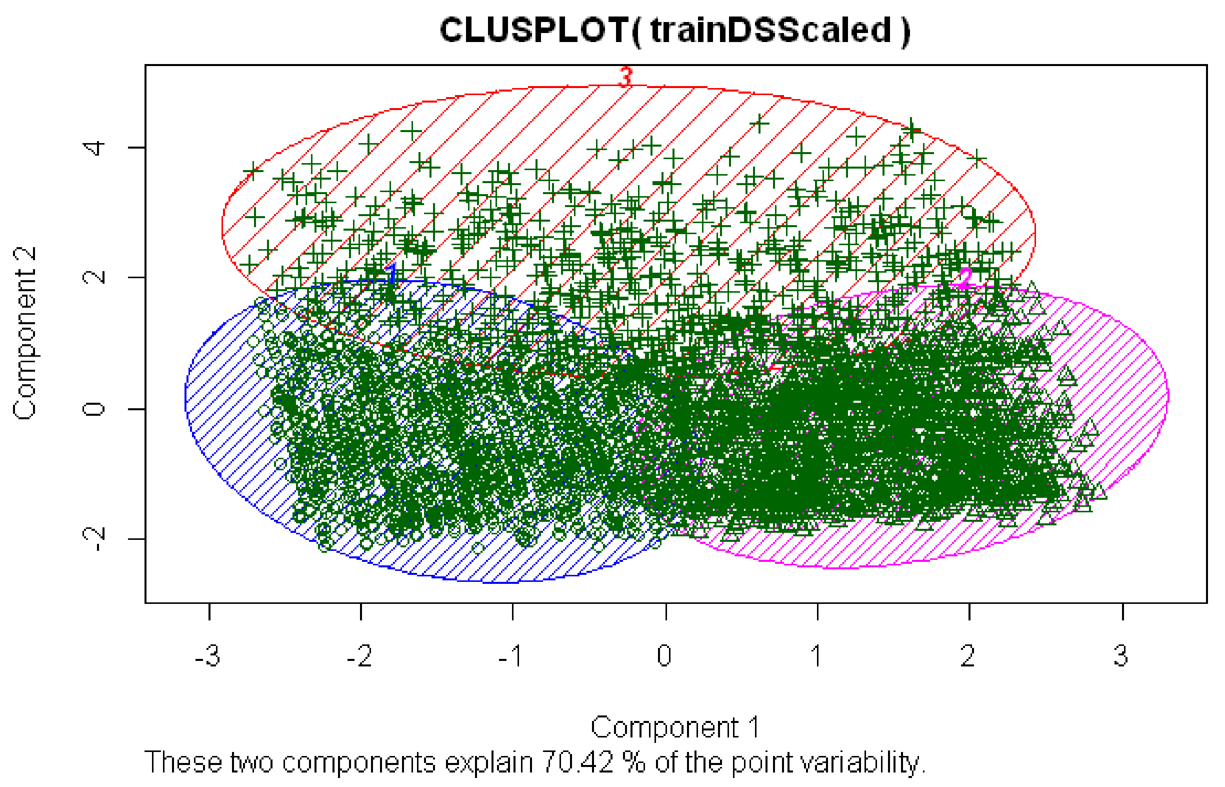
Observation
Income, Education and Family plays out as significant factors that decides personal loan

3. K-Means Clustering

Since the data size is large, we use K-means clustering. Based on the elbow curve we can see 3 clusters formed. K means cluster is performed on the refined data by removing the outliers identified in the data transformation stage.

```
K-means clustering with 3 clusters of sizes 1776, 1787, 819


Cluster means:
       age experience    income     ccavg   mortgage
1 -0.9335090 -0.9334823 -0.2102144 -0.2697066  0.05547239
2  0.8759760  0.8731447 -0.3872652 -0.3885018  0.05266850
3  0.1129949  0.1191148  1.3008348  1.4325416 -0.23521071
```

```
library(cluster)
income_outlier = boxplot(trainDS$income, plot=FALSE)
out_Income = income_outlier$out
trainDS_refined1 <- trainDS[-which(trainDS$income %in% out_Income),]


CCAvg_outlier <- boxplot(trainDS$ccavg, plot=FALSE)
out_ccavg<-CCAvg_outlier$out
trainDS_refined1 <- trainDS_refined1[-which(trainDS_refined1$ccavg %in% out_ccavg),]


Mort_outlier <- boxplot(trainDS$mortgage, plot=FALSE)
out_mort <- Mort_outlier$out
trainDS_refined1 <- trainDS_refined1[-which(trainDS_refined1$mortgage %in% out_mort),]

nrow(trainDS_refined1)

custData.Scaled = trainDS_refined1 %>% select_if(is.numeric)

trainDSScaled = scale(custData.Scaled, center = TRUE)

#print(trainDSScaled)

seed=1000
set.seed(seed) #since kmeans uses a randomized starting point for cluster centroids

clustk = kmeans(x=trainDSScaled, 3, nstart = 10)
print(clustk)
clusplot(trainDSScaled, clustk$cluster,
      color=TRUE, shade=TRUE, labels=4, lines=1)
```



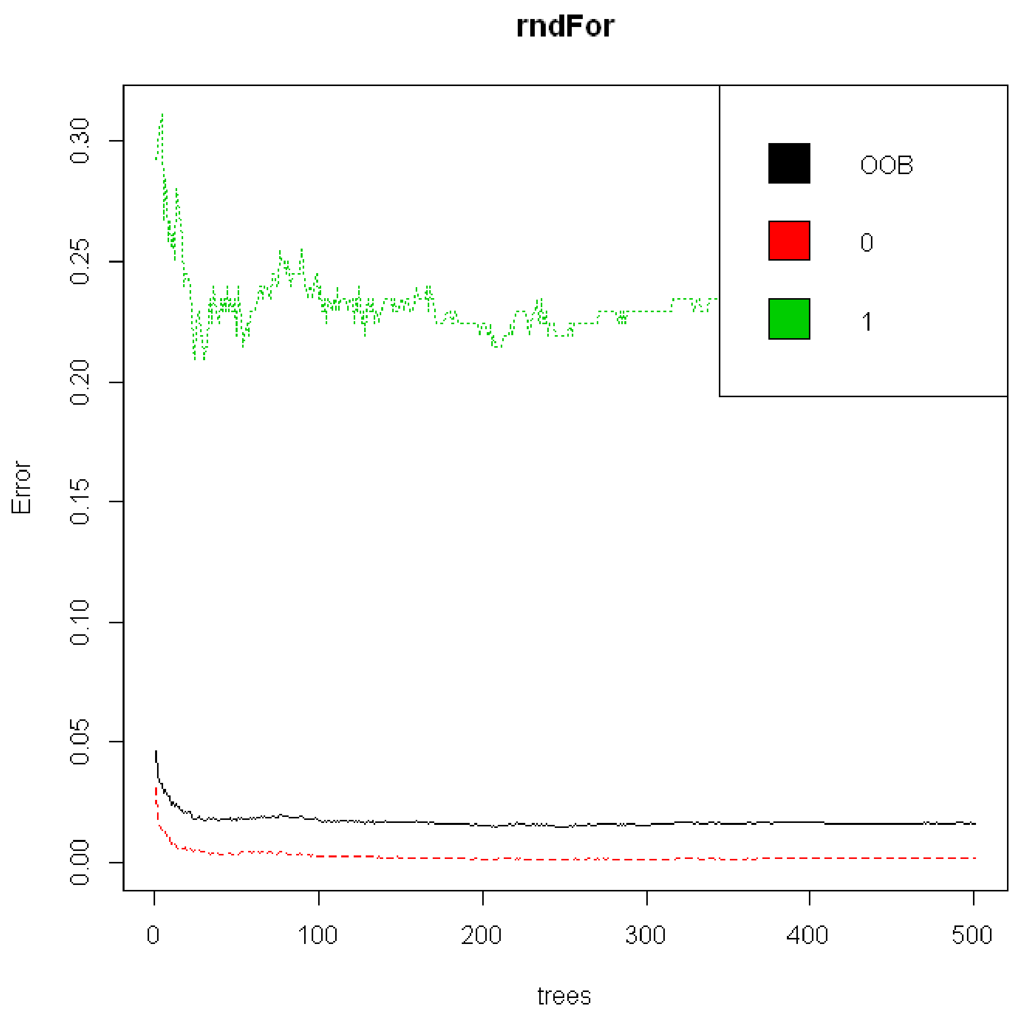
4. Random Forest

```
library(randomForest)

 ind <- sample(2, nrow(trainDS_refined1), replace=TRUE, prob=c(0.7, 0.3))
 trainData <- trainDS_refined1[ind==1,]
 testData <- trainDS_refined1[ind==2,]

head(trainData)
dim(trainData)

nrow(trainData)

print(sum(trainData$personal_loan=="1")/nrow(trainData))

seed=1000
set.seed(seed)
rndFor = randomForest(trainData$personal_loan ~ ., data = trainData[,-c(4)],
                ntree=501, mtry = 3, nodesize = 10,
                importance=TRUE)

print(rndFor)
```

```
Call:
 randomForest(formula = trainData$personal_loan ~ ., data = trainData[,      -c(4)
], 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: 1.63%
Confusion matrix:
     0   1 class.error
0 2870   6 0.002086231
1   44 152 0.224489796
```

Error Rate

```
            OOB          0         1
[1,] 0.04611212 0.03073967 0.2923077
[2,] 0.03646409 0.01942319 0.2972973
[3,] 0.03337784 0.01519468 0.3049645
[4,] 0.03271028 0.01372712 0.3109756
[5,] 0.02883922 0.01270208 0.2670455
[6,] 0.03052376 0.01333333 0.2841530
```


rndFor
Error
trees
OOB