Data Mining

THERA BANK – Loan Purchase Modelling

Contents

Introduction:

Please have a look at the colour codes used, so that it will be easy for you to interpret the work done.

Colour & Font Size used in this report (for ease of interpretation :

Categories	Font Colour & Size
Main Header	**Blue Bold Font** – Arial 11
Sub-Header	**Blue Bold Font** – Arial 10
Sub-Sub Header	**Blue Bold Font** – Arial 9
Description	Dark Blue Font – Arial 10
>R commands & Interpretations	Dark Red Lucida Consc 9
Comments against R Commands (given as #Steps)	Blue Accent 5 Lucida Consc 9
>R Output	Black Lucida Consc 9

READING THE DATA : We have named our file as "**data**"

```
> data <- read_excel("Thera Bank-Data Set.xlsx", sheet = "Bank_Personal_Loan_
Modelling")
```

So here we go, in a step-wise manner starting with Exploratory Data Analysis :

I. Exploratory Data Analysis (EDA) :

Basic Data Summary :

Data Cleansing : There are 5000 row items and 14 variables in the data-set given, where we will try to do t he cleansing of the data and fill in any missing values.

```
#Step1 - checking the dimensions
> dim(data)
[1] 5000   14
```

There We checked for missing values and found that there are 18 cells missing under the column "Family Members". Since out of the total 5000, only 18 records of no. of family members are missing and also sinc e there are other predictor variables in these 18 records, which will be valuable for our analysis, we thoug ht it prudent to replace it with zero instead of deleting the rows. The below R-commands and results are a s shown below :

```
#Step 2 - Checking for missing values in the data-set
> any(is.na(data))
[1] TRUE
```

```
# Step - 3 checking columns which have missing values
> sapply(data, function(x){sum(is.na(x))})
```

```
ID        Age (in years) Experience (in years)   Income (in K/month)
0                   0                     0                     0
ZIP Code        Family members            CCAvg              Education
0                  18                     0                     0
Mortgage         Personal Loan     Securities Account            CD Accout
0                   0                     0                     0
Online             CreditCard
  0                   0
```

```
#Step 4 - replacing missing values with 0
> data[is.na(data)]=0
```

```
# Step 5 - re-checking for missing values after replacing with zero
> any(is.na(data))
[1] FALSE
```

Interpretation / Comments : *By using the above commands we have replaced the missing values with "0" to have our basic data-set complete and re-checked if there are any further missing values and then proceeded further as below :*

Step 6 - checking the data summary again
```
> summary(data)
```
```
 ID         Age (in years)   Experience (in years) Income (in K/month)    ZIP Code    
  Min.   :   1   Min.   :23.00   Min.   :-3.0          Min.   :  8.00      Min.   : 9307  
  1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0          1st Qu.: 39.00      1st Qu.:91911  
  Median :2500   Median :45.00   Median :20.0          Median : 64.00      Median :93437  
  Mean   :2500   Mean   :45.34   Mean   :20.1          Mean   : 73.77      Mean   :93153  
  3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0          3rd Qu.: 98.00      3rd Qu.:94608  
  Max.   :5000   Max.   :67.00   Max.   :43.0          Max.   :224.00      Max.   :96651  
  Family members       CCAvg          Education       Mortgage       Personal Loan  
  Min.   :0.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0   Min.   :0.000  
  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1st Qu.:0.000  
  Median :2.000   Median : 1.500   Median :2.000   Median :  0.0   Median :0.000  
  Mean   :2.389   Mean   : 1.938   Mean   :1.881   Mean   : 56.5   Mean   :0.096  
  3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0   3rd Qu.:0.000  
  Max.   :4.000   Max.   :10.000   Max.   :3.000   Max.   :635.0   Max.   :1.000  
  Securities Account    CD Account          Online          CreditCard   
  Min.   :0.0000     Min.   :0.0000    Min.   :0.0000    Min.   :0.000  
  1st Qu.:0.0000     1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.000  
  Median :0.0000     Median :0.0000    Median :1.0000    Median :0.000  
  Mean   :0.1044     Mean   :0.0604    Mean   :0.5968    Mean   :0.294  
  3rd Qu.:0.0000     3rd Qu.:0.0000    3rd Qu.:1.0000    3rd Qu.:1.000  
  Max.   :1.0000     Max.   :1.0000    Max.   :1.0000    Max.   :1.000  
```

Initial Data Insights : *The insights from the summary above : i) The average age of the customers is ~45 and also average experience in years is ~20 years, which means the population is a matured population who are responsible for their families spending and earning.*

‘’’

```
> dim(data)
[1] 5000   14
```

Data Transformation : *Now we again look at the data and try to convert the column variables into the right unit of factor/numericals/ordinal data etc.. Also we look at converting any negative values like experience i nto positive. Steps 7 to 14 covers the same.*

```
# Step 7 – Look at the internal structure of the bank dataset. Now each of 14 variable
s are in num. So it has to be converted into the right class or units.
> str(data)
```
```
 Classes ‘tbl_df’, ‘tbl’ and 'data.frame':  5000 obs. of  14 variables:
  $ ID                    : num  1 2 3 4 5 6 7 8 9 10 ...
  $ Age (in years)        : num  25 45 39 35 35 37 53 50 35 34 ...
  $ Experience (in years): num  1 19 15 9 8 13 27 24 10 9 ...
  $ Income (in K/month)   : num  49 34 11 100 45 29 72 22 81 180 ...
  $ ZIP Code              : num  91107 90089 94720 94112 91330 ...
  $ Family members        : num  4 3 1 1 4 4 2 1 3 1 ...
  $ CCAvg                 : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
  $ Education             : num  1 1 1 2 2 2 2 3 2 3 ...
  $ Mortgage              : num  0 0 0 0 0 155 0 0 104 0 ...
  $ Personal Loan         : num  0 0 0 0 0 0 0 0 0 1 ...
  $ Securities Account    : num  1 1 0 0 0 0 0 0 0 0 ...
  $ CD Account            : num  0 0 0 0 0 0 0 0 0 0 ...
  $ Online                : num  0 0 0 0 0 1 1 0 1 0 ...
  $ CreditCard            : num  0 0 0 0 1 0 0 1 0 0 ...
```

```
#Step 8 - dropping the first and 5th columns of the dataset i.e. ID and Zip Code as
they are not relevant data for working as such.
> mydata = data[,-c(1,5)]
> View(mydata)

# Step 9 : We convert the following multiple columns into factor columns
> col = c("Education","Personal Loan", "Securities Account", "CD Account", "Online", "
CreditCard")
> mydata[col] = lapply(mydata[col], factor)

#Step 10 : converting education into ordinals
```

```
> mydata$Education = factor(mydata$Education, levels = c("1","2","3"), order = TRUE)

# Step 11 : Abbreviating variable names
> mydata = mydata%>% rename(Age = "Age (in years)", Experience = "Experience (in years
)", Income = "Income (in K/year)")
> View(mydata)

# Step 12 : checking for negative values in experience col
> head(mydata[mydata$Experience<0,])
```

```
# A tibble: 6 x 12
Age Experience Income `Family members` CCAvg Education Mortgage `Personal Loan` `Securit
ies Acc~
  <dbl>      <dbl>  <dbl>            <dbl> <dbl> <ord>         <dbl> <fct>              <fct
>
1    25         -1    113                4  2.3   3                0 0                0
2    24         -1     39                2  1.7   2                0 0                0
3    24         -2     51                3  0.3   3                0 0                0
4    28         -2     48                2  1.75  3               89 0                0
5    24         -1     75                4  0.2   1                0 0                0
6    25         -1     43                3  2.4   2              176 0                0

# ... with 3 more variables: `CD Account` <fct>, Online <fct>, CreditCard <fct
```

```
# Step 13 : converting negative values to positive values
> mydata$Experience = abs(mydata$Experience)
>
> dim(mydata)
[1] 5000   12

# Step 14 : Now we look at the summary data again
> summary(mydata)
```

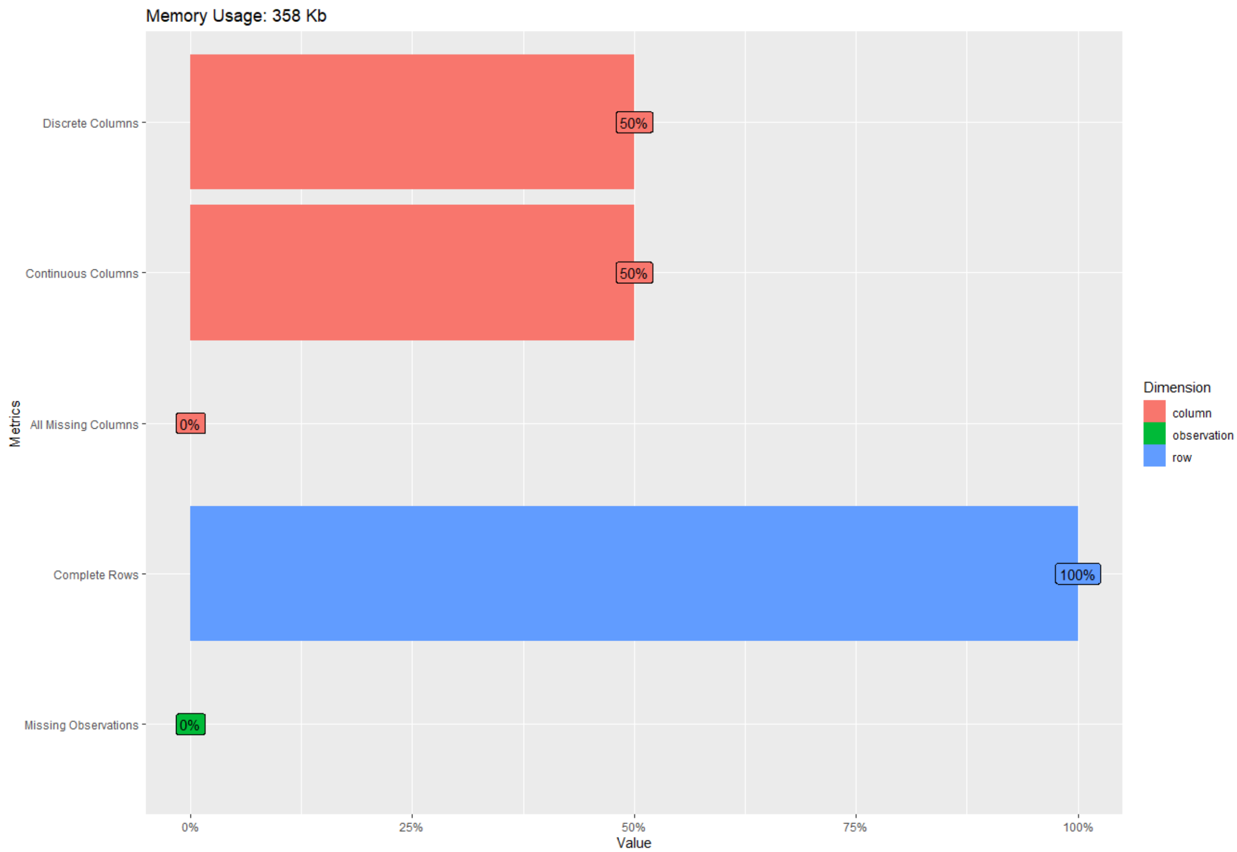
```
 Age           Experience       Income        Family members     CCAvg          Education
 Min.   :23.00   Min.   : 0.00   Min.   :  8.00   Min.   :0.000   Min. 0.000    1:2096
 1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000   1st Qu.: 0.700   2:1403
 Median :45.00   Median :20.00   Median : 64.00   Median :2.000   Median : 1.500   3:1501
 Mean   :45.34   Mean   :20.13   Mean   : 73.77   Mean   :2.389   Mean   : 1.938
 3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 98.00   3rd Qu.:3.000   3rd Qu.: 2.500
 Max.   :67.00   Max.   :43.00   Max.   :224.00   Max.   :4.000   Max.   :10.000

  Mortgage      Personal Loan Securities Account CD Account Online   CreditCard
 Min.   :  0.0   0:4520        0:4478             0:4698     0:2016   0:3530
 1st Qu.:  0.0   1: 480        1: 522             1: 302     1:2984   1:1470
 Median :  0.0
 Mean   : 56.5
 3rd Qu.:101.0
 Max.   :635.0
```

Interpretation of steps 7 to 14 : *In the above steps, we have tried to simplify and rationalize the data by changing the units and removing any discrepancies like -ve values for experience etc as per the steps 7 to 14 above.*

‘’’

Metrics of EDA : *Now, lets look at the metrics for the entire Data. Now all the missing variables and missing observations have got removed as shown in the metrics below :*

```
>plot_intro(mydata)
```



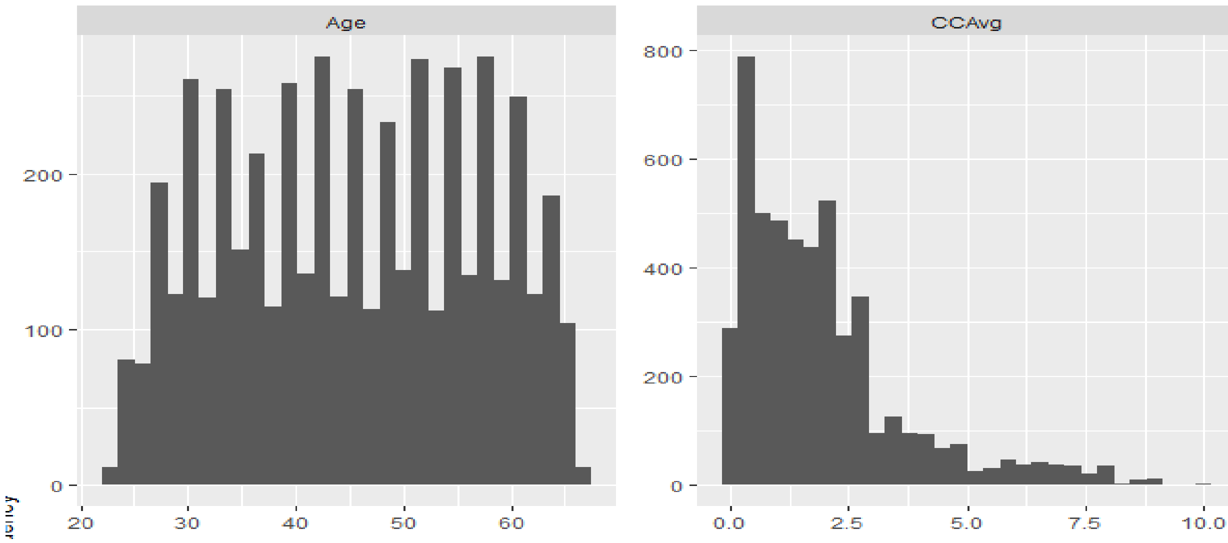
Interpretation : *Now there are no missing rows or columns or any missing observations. The number of discrete and continuous columns are both equal in proportion*

‘,,,

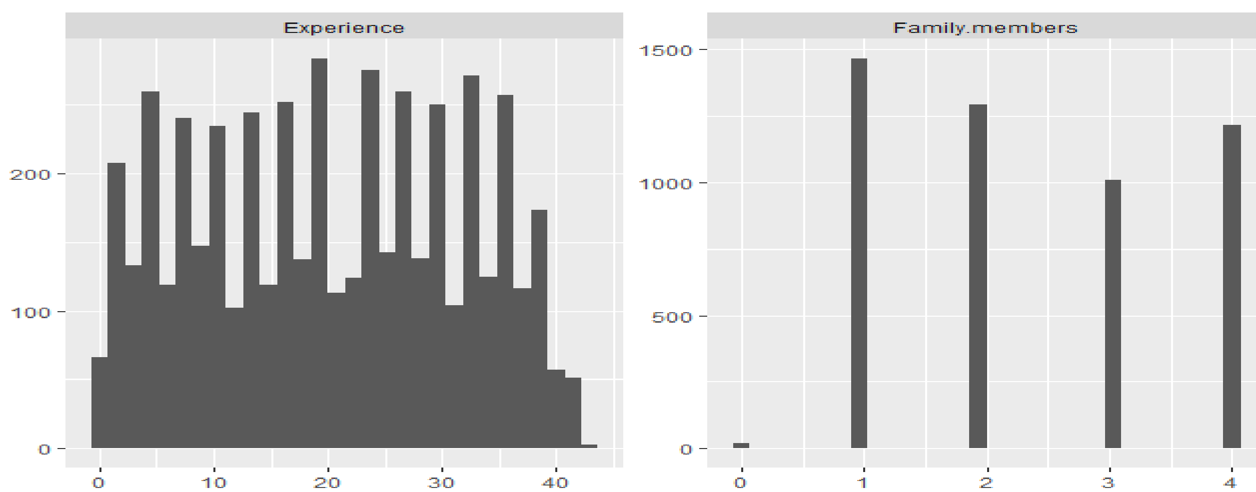
Univariate Analysis : *Now, lets try to plot the Histogram for each of the numerical variables and try to interpret the results*

Histogram for Numerical Variables -Step 15

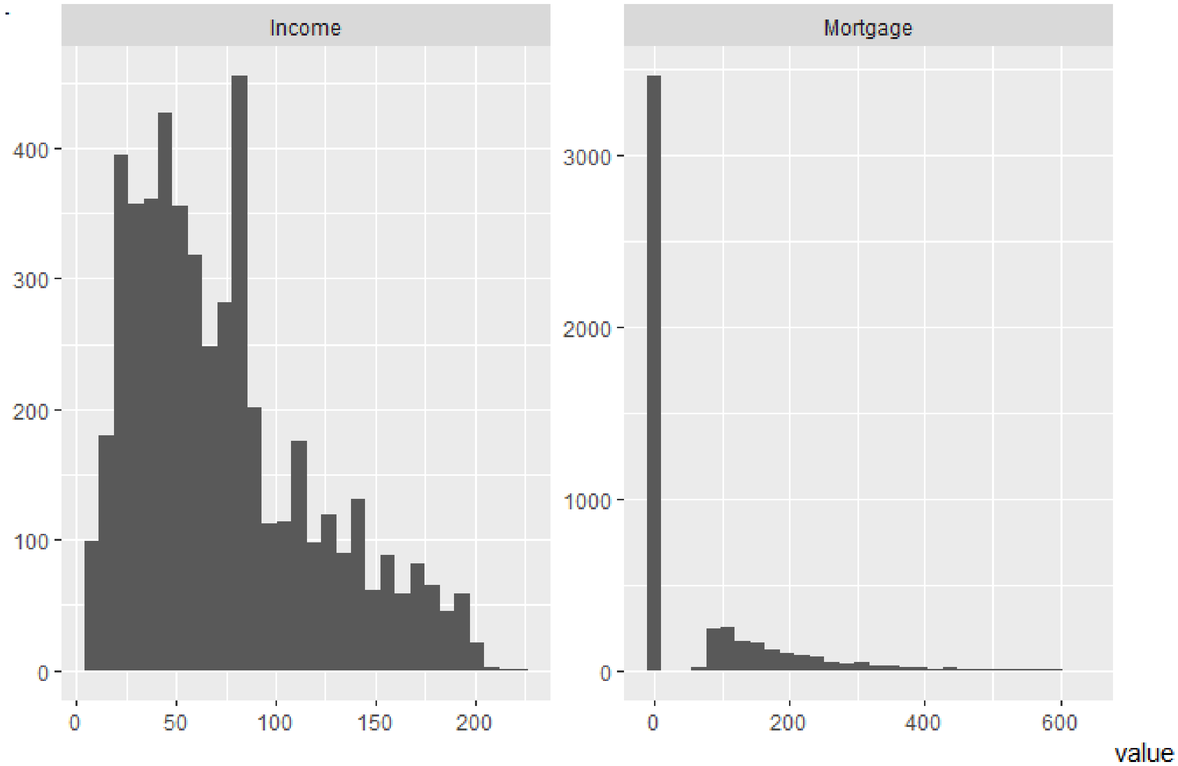
```
> plot_histogram(mydata)
```



Interpretation	
Age	CCAvg
Most of the customers are in the age group between 30 and 60 which means, earning members of the family	*The **C**redit **C**ard **A**vergage spending of majority of the customers is only upto $2500. But there are another approx.. 50% customers who spend a little more which means there are people who take more money on credit card. These set of customers can possibly become the potential liability customers to be converted as Asset customers and sold the Personal Loan Product.*



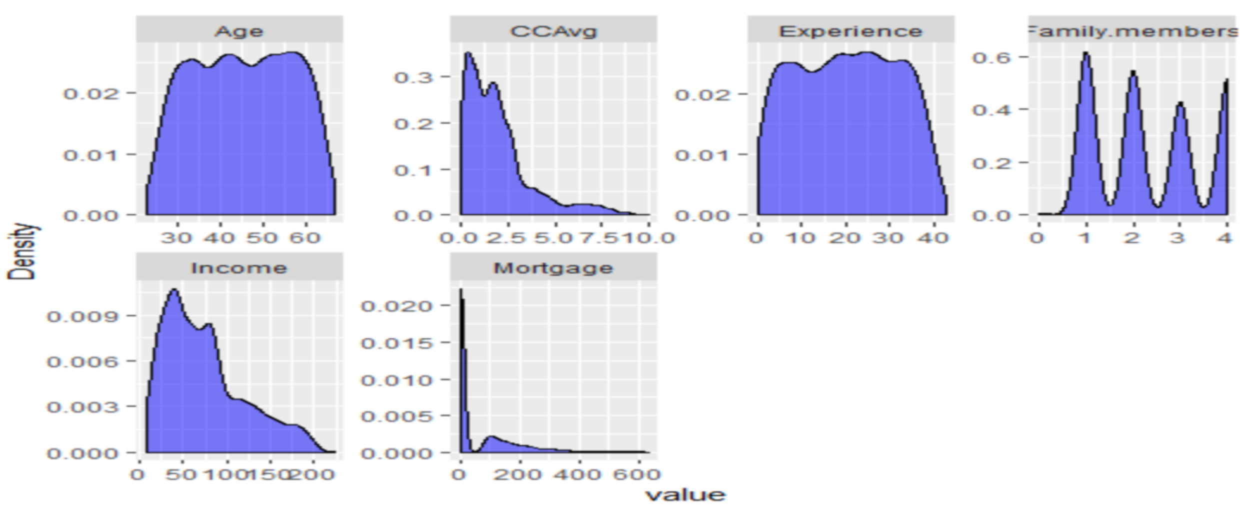
Interpretation	
Experience	Family Members
The professional work experience of the population is u niformly distributed between 2 years upto 40 years. So money needs have to be highlighted for the different life stages to the customers.	*There are close to 1464 customers who are single and another1293 customers who are just 2 members. The remaining population are having children. So the bank can adopt different selling strategies for each such segment of customer population.*



Interpretation	
Income	**Mortgage**
50% of the population have income upto approx. $64000 annualy per month and the remaining 50% have average income upto $1,00,000. So we can say there are almost 50% people who are lower income who may need money. And the rest 50% earning high may have higher consumption and hence may go for some loan requirements. This is a preliminary observation	*Only 15% i.e. 731 customers have mortgaged their house for some loan and the remaining do not have any such mortgage. It can also mean that customers do not have long term liability or any major loan EMI currently and **are likely** go for short duration loans like personal loans.*

Density for Numerical Variables -Step 16

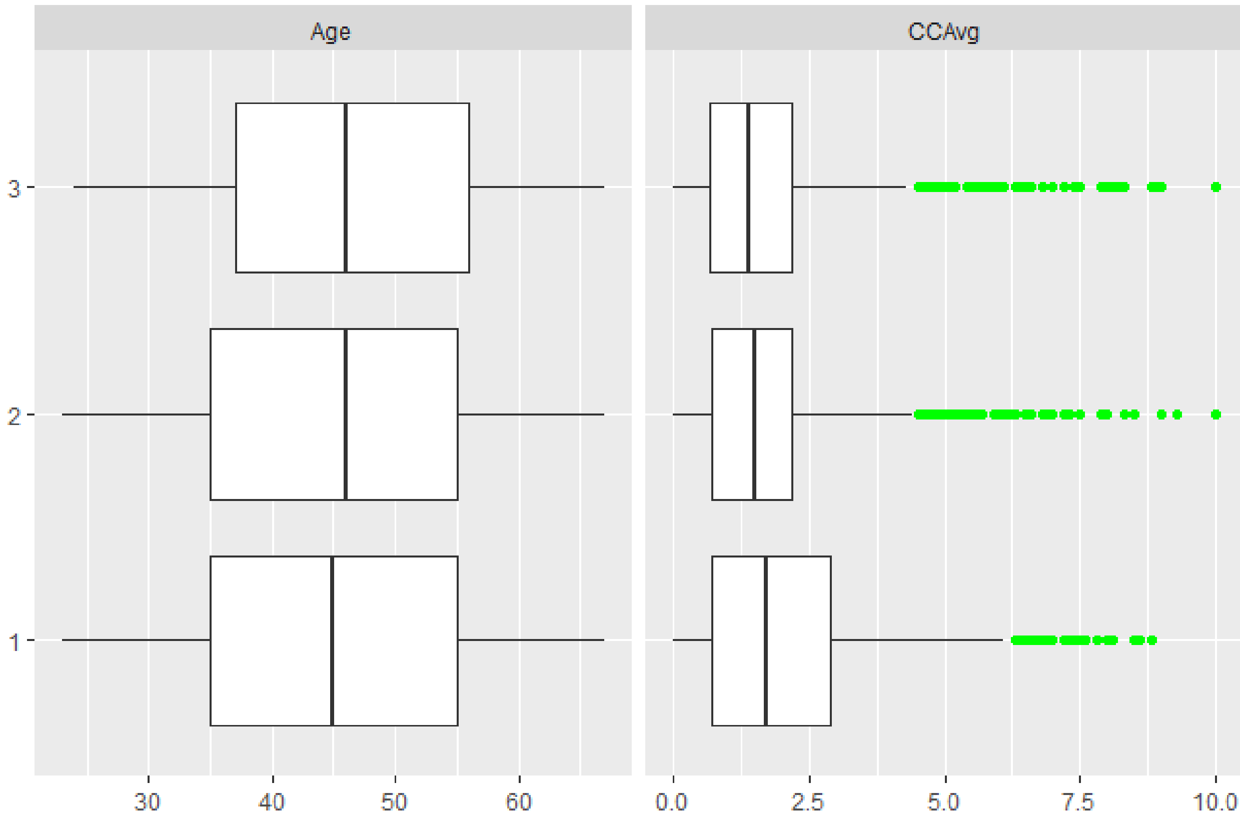
```
> plot_density(mydata,geom_density_args = list(fill="blue", alpha=0.5))
```



Bivariate Analysis : *Now, lets try to draw box plots for each of the variables & try to interpret the results*

Boxplots by Education - Step 16

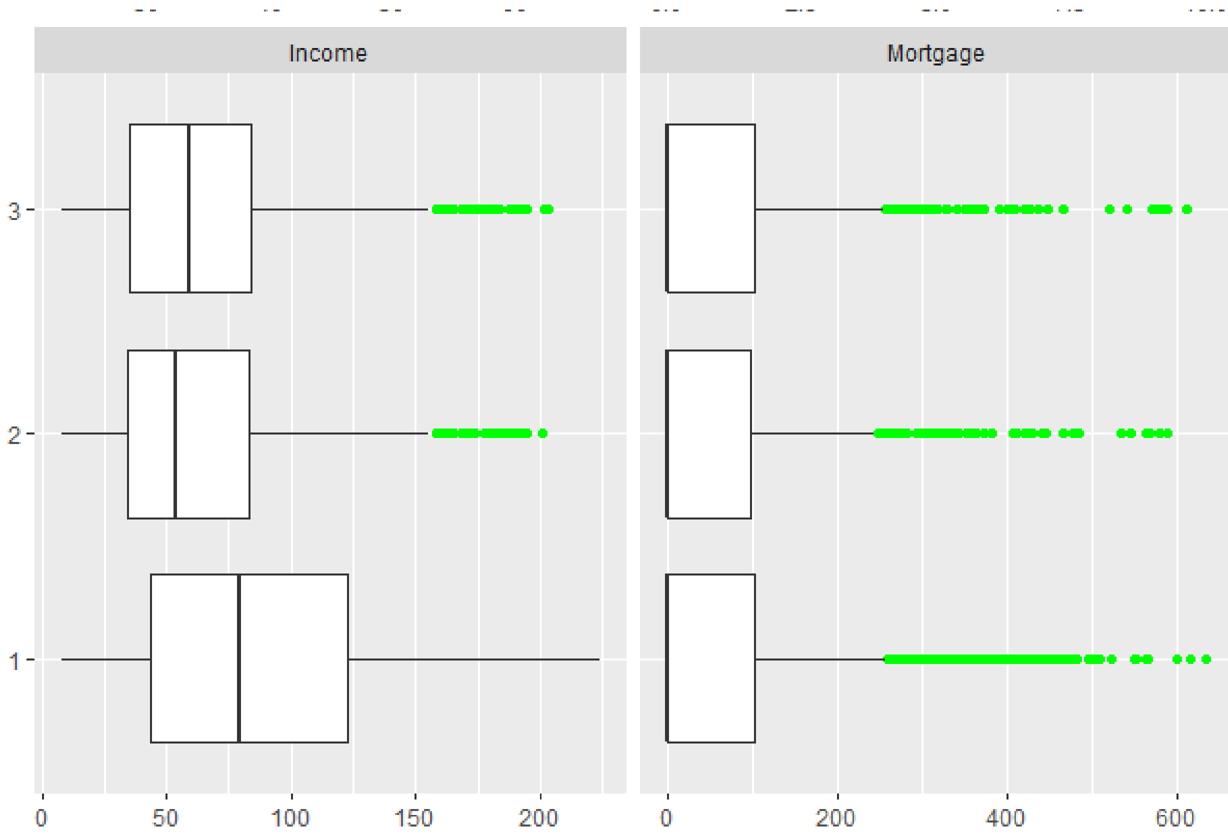
```
> plot_boxplot(mydata, by = "Education", geom_boxplot_args = list("outlier.color" = "g
reen"))
```



Interpretation	
Age vs Education	CCAvg vs Education
As you can read from the box-plot of age vs education, t here is an even distribution of population across the und ergraduates, graduates and Advanced professional cust omer base and the average age is around 45	*In each of the educational class of under-graduate, graduate and advanced professional customer group, it is a right skewed plot with lot of outliers i.e. the average spend on ccredit card is around $1900 across all the customers, however there are many customers(**Outliers**) who spend more than $5000 per month across the 3 levels of education. So it will help us target such customers who have high credit card spend to convert as personal loan customers.*

class	count	total sum of age	Avg age
UnderGrad	2096	94244	44.96
Graduate	1403	63191	45.04
Adv,. Professional	1501	69257	46.14

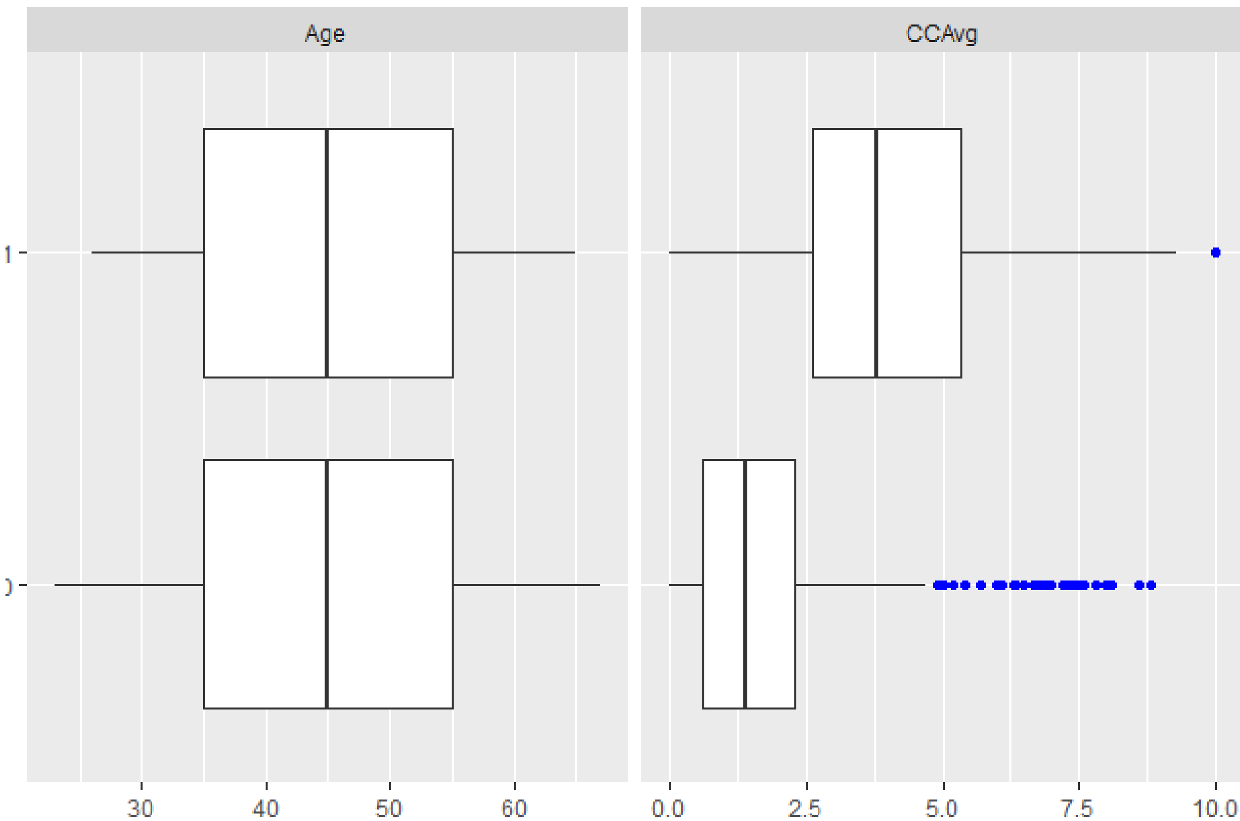
Interpretation	
Experience vs Education	Family Members vs Education
In each of the educational the average work experience in years is ~20 years, which means the population is a m atured population who are responsible for their families spending and earning	*There is no direct relation as such between education and the no. of family members*



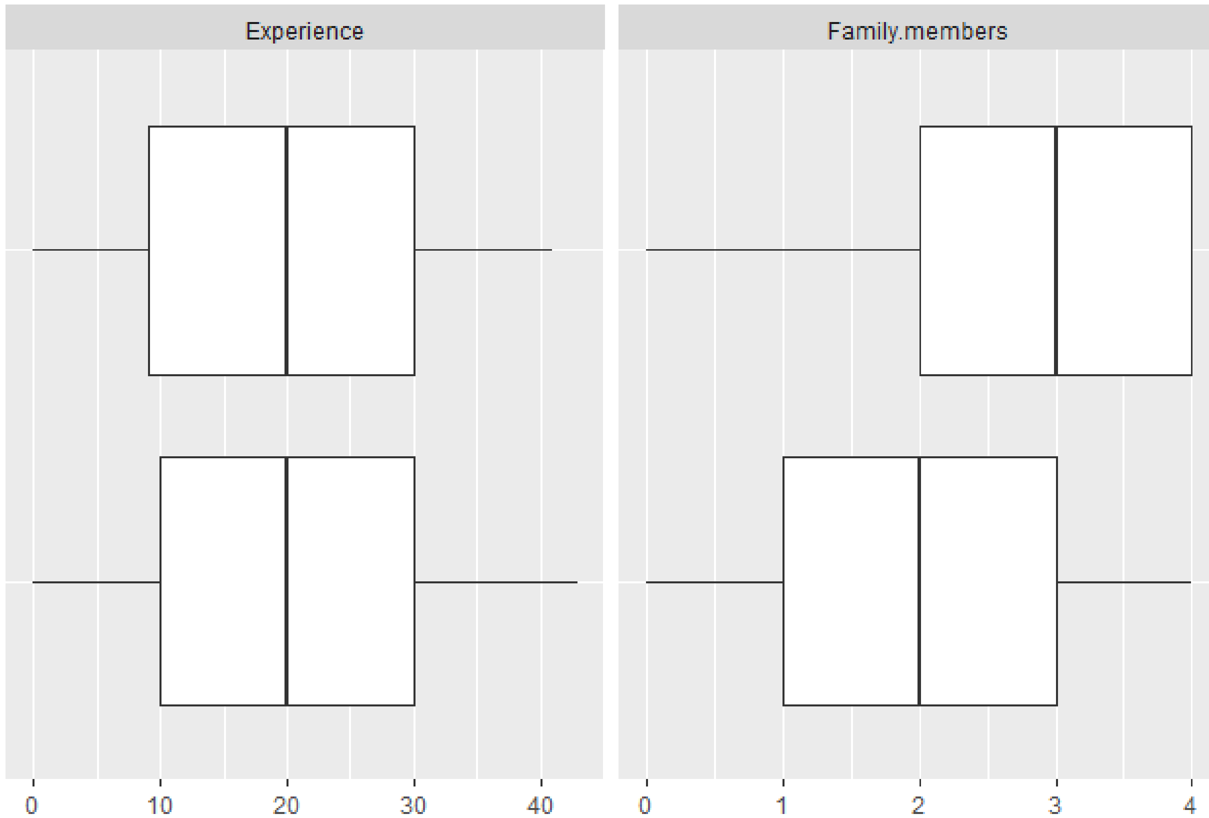
Interpretation	
Income vs Education	Mortgage vs Education
*In the undergraduate class has a higher average incom e of around $73000 per year as compared to the gradua te class or the advanced professional, however in the gr qraduate and advanced professional group, we have cu stomers(**Outliers**), who have a higher income above $1 50000 dollars also. So this gives an insight that there ar e customers who have good amount of income for bank to lend a personal loan.*	*Irrespective of the class of Education, there are only few people(**mostly outliers**) around 15% who have mortgaged their house. However, there is no direct relation between Education and Mortgage done by customers. So most of the people are those who do not have mortgage loans, which means there is a potential to sell short term personal loans to these people based on their monthly income.*

Boxplot by Personal Loan - Step 17

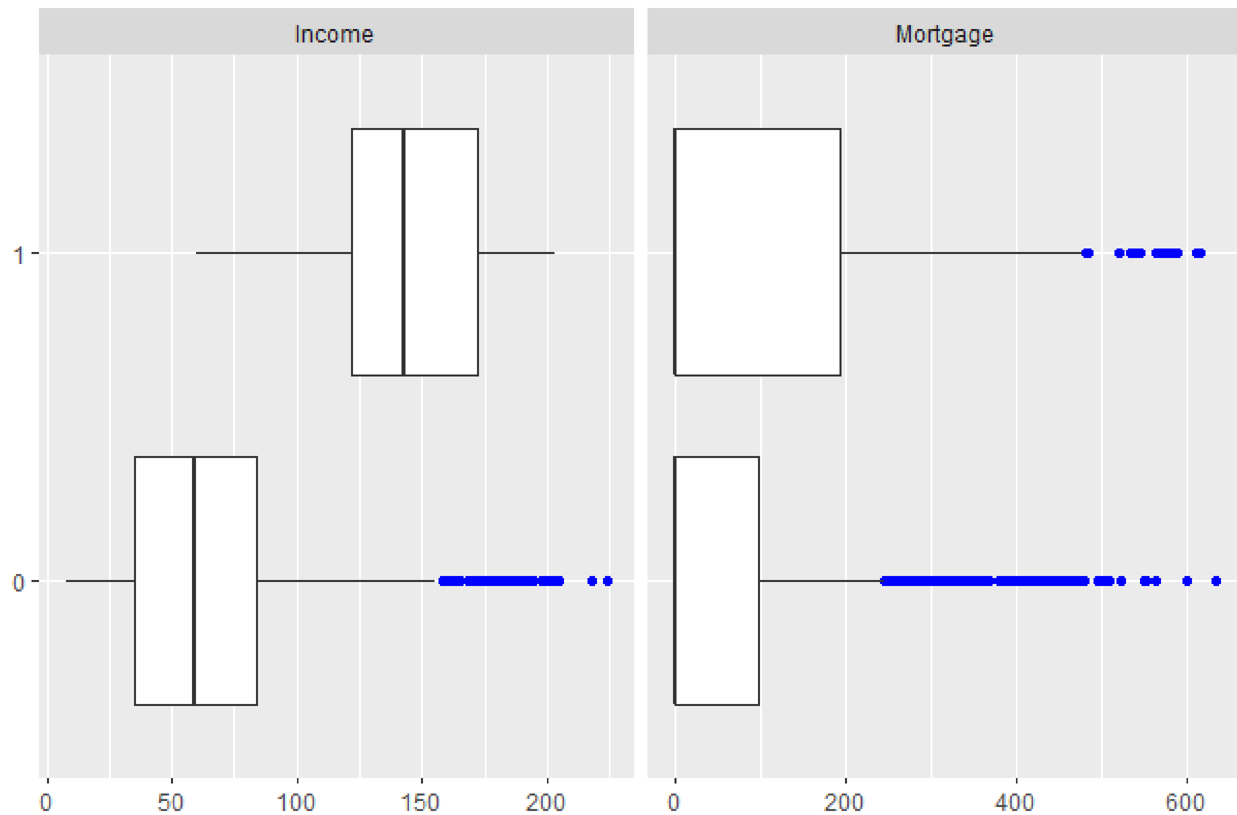
```
> plot_boxplot(mydata, by = "Personal Loan", geom_boxplot_args = list("outlier.color"
= "blue"))
```



Interpretation	
Age vs Personal Loan	CCAvg vs Personal Loan
There is no specific relation to age for availing of personal loan. From the data summary we know that 480 customers have availed personal loan in the previous marketing campaign and their average age is around 45 years. The remaining 4520 customers who have not availed the personal loan in the previous campaign also have an average age of 45 years. So age probably does not have direct relationship for availing personal loan. There are other factors to be considered to identify the potential customer for conversion as an asset customer.	*From the box-plot above it is evident that out of the **480 customers** who availed the personal loan in the previous campaign, their **average credit card spend which is $3.91k** is much higher than the average credit card spend of $1.73k of the remaining 4520 customers, who did not avail personal loan in the previous campaign. So in a way it shows that the customers spending more amount using their credit card are also more likely to go for a personal loan. This is just an initial observation based on this box plot. Also if you see there are many outliers(high credit card spending customers) in the population of customers who did not avail the personal loan, who can be looked at for converting into an asset customer.*



Interpretation	
Experience vs Personal Loan	Family Members vs Personal Loan
In each of the category of population who have availed or not availed personal loan in the previous marketing campaign, the average work experience in years is ~20 years, which means the population is a matured population who are responsible for their families spending and earning.	*The box plot graph shows that in the previous marketing campaign, the customers who have availed personal loan have an average family member size of 3, which means if the family size is more then the customer has more inclination to avail for a personal loan. Obviously more family members means, more money requirement.*



Interpretation	
Income vs Personal Loan	Mortgage vs Personal Loan
It is quite evident that the average income of customers who have availed personal loan in the previous marketin g campaign was much higher at $144.75k as compared to the average annual income of customers who did not avail the personal loan ($66.24k). This means higher inc ome class customers are the spending population and h ence they are the ones who will be more inclined to avai l a personal loan amongst other factors.	*From the above, we are getting an indication that those customers who availed the personal loan in the previou s marketing campaign mostly did not have any mortgag e loan. The ones who have the higher mortgage loans, will not be inclined to take additional personal loan.*

Previous campaign	count	total sum Income	Avg Annual Income $'000
No pers loan	4520	299393	66.24
Pers. Loan Availed	480	69478	144.75

We are getting some kind of indication by these box plot analysis, which we have to validate as we go further. However we can summarise as below :

Insight :
So in a way, we should target customers with reasonably higher income and high credit card sp end but with less exposure on mortgage loans.

Corelation between numeric features : *Lets see which variables are more co-related to each other*

```
cor_data<-mydata[,c(1,2,3,5,7)]
resp<-cor(cor_data)
round(resp,2)
```

```
              Age  Experience Income CCAvg Mortgage
Age          1.00        0.99  -0.06 -0.05    -0.01
Experience   0.99        1.00  -0.05 -0.05    -0.01
Income      -0.06       -0.05   1.00  0.65     0.21
CCAvg       -0.05       -0.05   0.65  1.00     0.11
Mortgage    -0.01       -0.01   0.21  0.11     1.00
```

Interpretation : *From the above result we can infer that*
a) Age and Experience are very positively correlated
b) Income and the average credit card spend are also highly correlated

‘,,

Multivariate Analysis : *Now, lets try to plot the two independent variables in different combinations again st the class of customers who have availed personal loan and those customers who have not availed the personal loan*

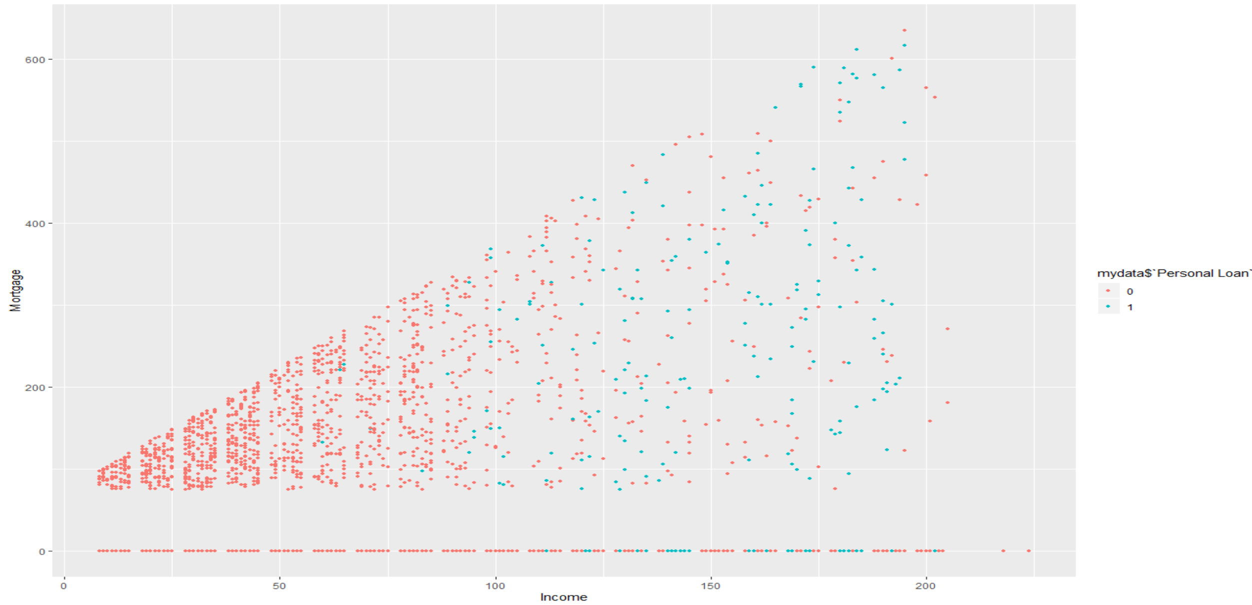
Plotting the of personal loan availed and not availed population against two predictor variables
(in different combinations) - > ## Step 18 –

Now since our interest lies in understanding which set of customers wi ll avail the personal loan, we need to do a plotting of the two catego ries of customers – i.e. “customers availing personal loan” and “custo mer not availing personal loan” and see how they are placed against he various other predictor variables

```
> ## Income vs Mortgage (scatter)
> ## Income (density)
> ## Mortgage (density)
> ## Age (density)
> ## Experience (density)
> ## Income vs Education (histogram)
```

```
p1 = ggplot(mydata, aes(Income, fill= "Personal Loan")) + geom_density(alpha=0.5)
> p2 = ggplot(mydata, aes(Mortgage, fill= "Personal Loan")) + geom_density(alpha=0.5
)
> p3 = ggplot(mydata, aes(Age, fill= "Personal Loan")) + geom_density(alpha=0.5)
> p4 = ggplot(mydata, aes(Experience, fill= "Personal Loan")) + geom_density(alpha=0
.5)
> p5 = ggplot(mydata, aes(Income, fill= Education)) + geom_histogram(alpha=0.5, bins
= 70)
> p6 = ggplot(mydata, aes(Income, Mortgage, color = "Personal Loan")) +
+   geom_point(alpha = 0.5)
> grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2, nrow = 3)
```

Income vs Mortgage(scatter)



Interpretation :
If you notice, in the scatter plot above, those who have availed personal loans(blue dots) are in the higher income bracket and who also has some mortage loans.

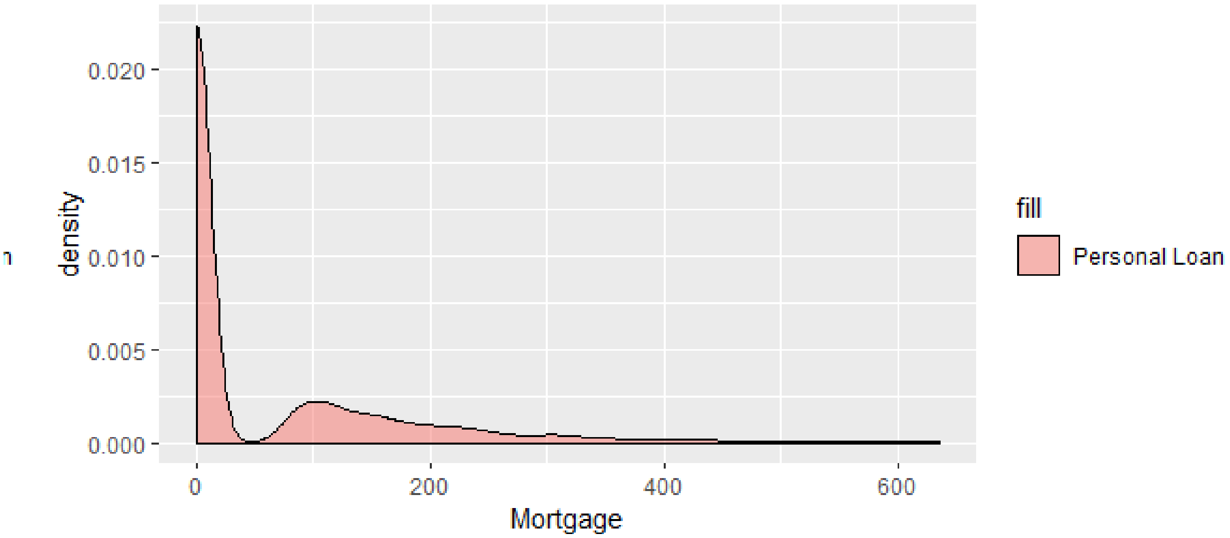
Population Density of Income vis-à-vis Personal Loan Availed

```
> p1 = ggplot(mydata, aes(Income, fill= "Personal Loan")) + geom_density(alpha=0.5)
```

Interpretation :
The maximum number of people who have availed personal loan are in the income bracket upto $60-70K annual income. So it's the needy people with medium income who prefer to go for personal loans.

Population Density of Mortgage cases vis-à-vis Personal Loan Availed

```
> p2 = ggplot(mydata, aes(Mortgage, fill= "Personal Loan")) + geom_density(alpha=0.5)
```



Interpretation :
The maximum number of people who have availed personal loan are people who do not have any mortga ges

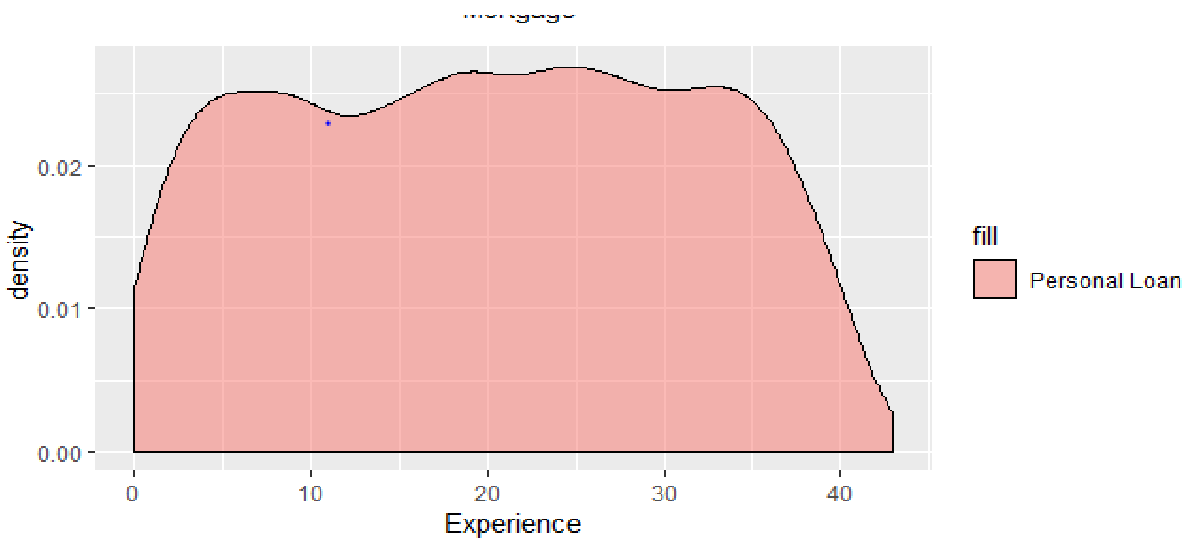
Population Density of "Age" cases vis-à-vis Personal Loan Availed

```
> p3 = ggplot(mydata, aes(Age, fill= "Personal Loan")) + geom_density(alpha=0.5)
```

Interpretation :
There is no relation of age to availment of personal loan as such as you can see from the above graph.

Population Density of "Experience" cases vis-à-vis Personal Loan Availed

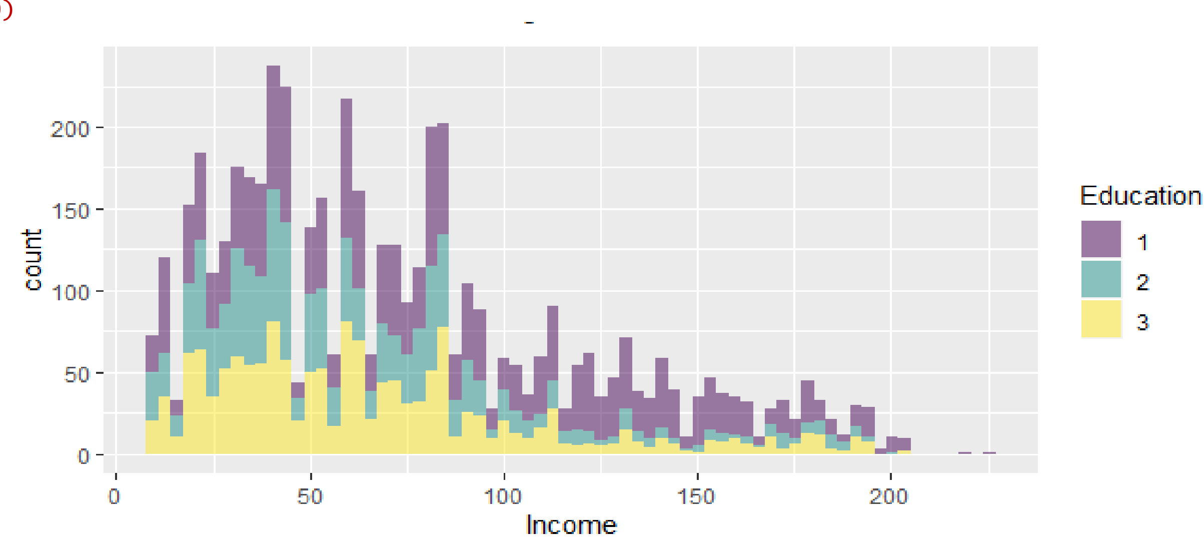
```
> p4 = ggplot(mydata, aes(Experience, fill= "Personal Loan")) + geom_density(alpha=0.5
)
```



Interpretation :
The years of work experience as such does not have a bearing on the availment of personal loan as the personal loan is availed across all tenure of work experience in no. of years .

Income vs Education

```
> p5 = ggplot(mydata, aes(Income, fill= Education)) + geom_histogram(alpha=0.5, bins =
70)
```



Interpretation :
The level of education as such is also not linked to the income of the customers, although upto 50% customers are in the middle income group range.

Income vs Mortgage

```
> p6 = ggplot(mydata, aes(Income, Mortgage, color = "Personal Loan"))
+   geom_point(alpha = 0.5)
```

```
> grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2, nrow = 3)
```

Interpretation :
The loans are preferred by customers in the lower to middle income group and also where they have a lesser mortgage exposure

‘’’

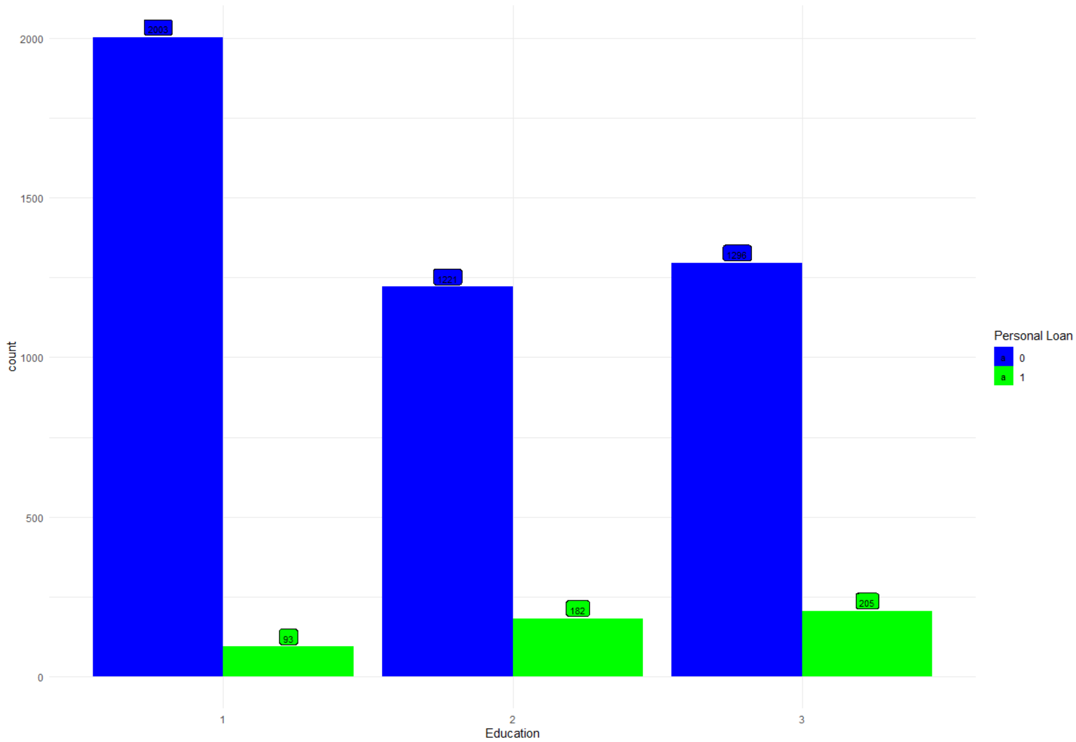
Number of Personal Loans taken (based on Education)

```
p1 = ggplot(mydata, aes(Income, fill= "Personal Loan")) + geom_density(alpha=0.5)
> p2 = ggplot(mydata, aes(Mortgage, fill= "Personal Loan")) + geom_density(alpha=0.5
)
> p3 = ggplot(mydata, aes(Age, fill= "Personal Loan")) + geom_density(alpha=0.5)
```

```
> p4 = ggplot(mydata, aes(Experience, fill= "Personal Loan")) + geom_density(alpha=0
.5)
> p5 = ggplot(mydata, aes(Income, fill= Education)) + geom_histogram(alpha=0.5, bins
= 70)
> p6 = ggplot(mydata, aes(Income, Mortgage, color = "Personal Loan")) +
+   geom_point(alpha = 0.5)
> grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2, nrow = 3)
```

```
> summary(mydata$`Personal Loan`)
```

```
0    1
4520  480
```

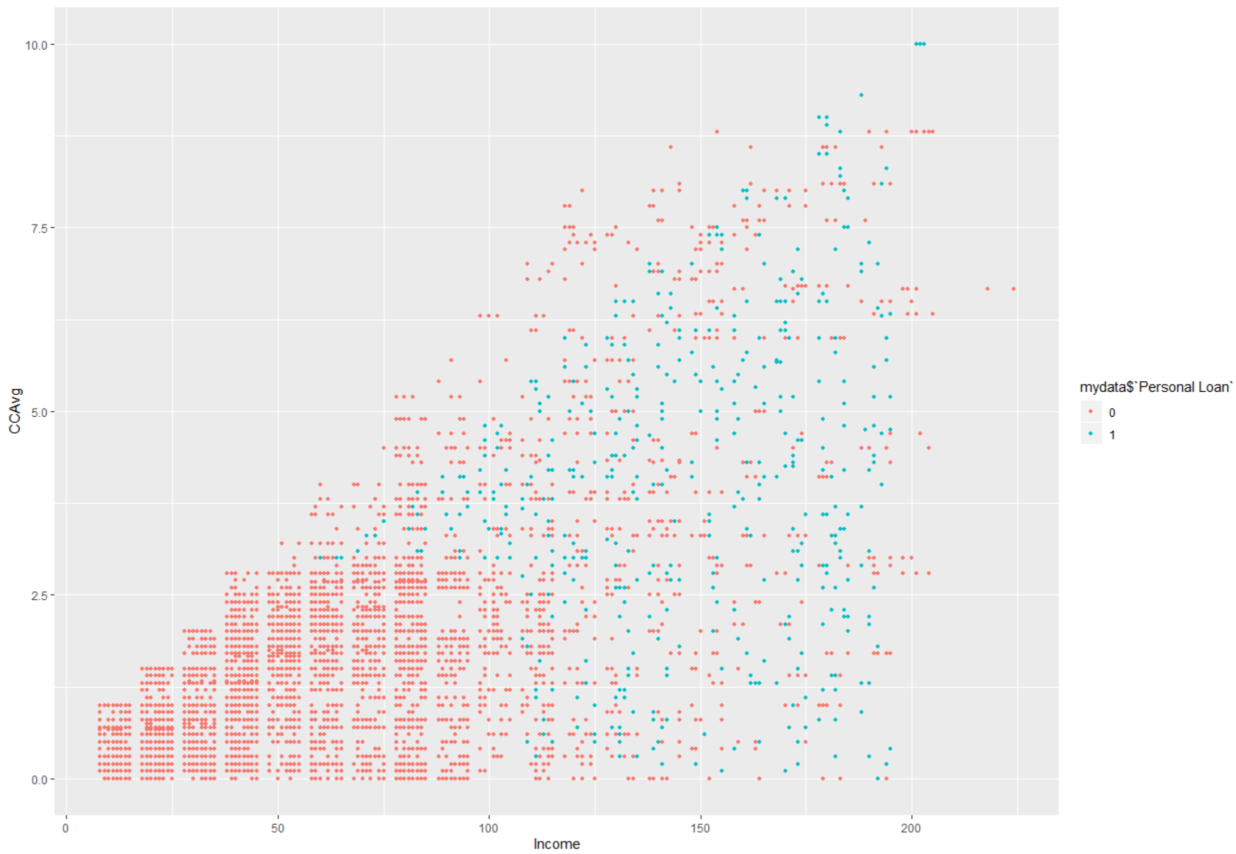


Interpretation :

The number of customers who have not availed personal loan in the previous campaign is higher across all the three levels of education i.e. graduate(1), undergraduate(2) and advanced professional(3).
This means there is a good no. of customers who can still be tapped for converting into a personal loan asset customer.
Out of the 480 personal loan availed cases, there are 205 advanced professionals, 182 graduates and 93 undergraduates. So level of education seems to have some role.

Number of Personal Loans taken (based on Credit Card Spend and Income)

```
ggplot(mydata, aes(Income,y = CCAvg, color = mydata$`Personal Loan`)) +
+   geom_point(size = 1)
```



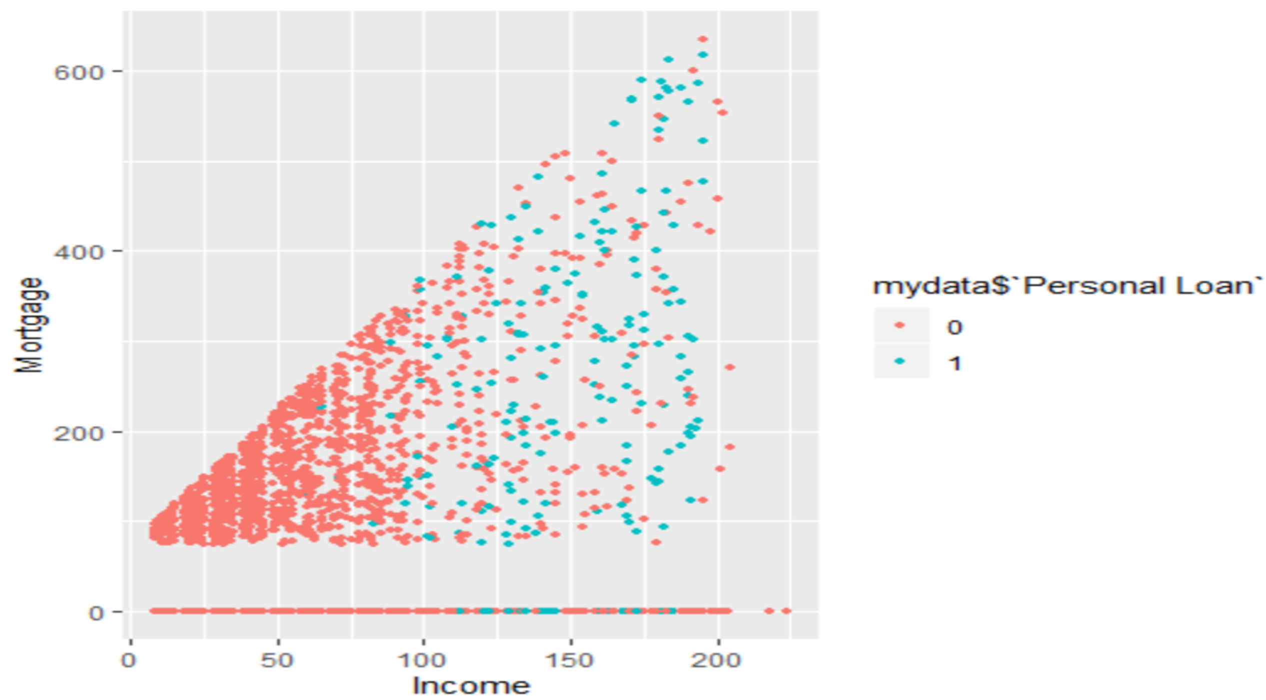
Interpretation :

- *If you notice, in the scatter plot above, those who have availed personal loans(blue dots) are having a higher Credit Card Spending, because they sometimes may avail a personal loan to repay their debts.*
- *Also, those who have availed personal loans(blue dots) are in the higher income bracket , and since all such high income people have not availed the personal loan, it gives good opportunity to the marketing team to target such high income customers by offering them personal loan to buy high lifestyle products*
- *We can also target customers who are in the middle income group of 40k to 100k annual income and who are spending less on credits cards as of now. So it a potential target customer group.*

Insight : Credit card spend and income is a good indicator of whom we need to target

Number of Personal Loans taken (based on Mortgage)

```
> ggplot(mydata, aes(Income,y = Mortgage, color = mydata$`Personal Loan`)) +
+   geom_point(size = 1)
```



Interpretation :

- *There are many customers who are having zero mortgage or low mortgages. So it's a good segm ent to target as they will not be having any high monthly EMI payout currently.*
- *Also those with higher Mortgages (above 150k) can also be targeted, since these customers would need additional money at lower interest rates to meet their repayment or daily needs*
- *.Hence Mortgage also plays a crucial role here in targeting customers for personal loan offering.*

Insight : Both high and low mortgage customers can be targeted using specific marketing strategy

II. CLUSTERING :

Applying Clustering Algorithm :

We need to divide the data into groups of customers having similar characteristics so as to have a targeted approach to sell the personal loan product to the potential customers at a minimal budget. We have in Unsupervised learning, both hierarchical and non-hierarchical methods of clustering. So we need to decide on the best clustering algorithm option.

Type of Algorithm : We have decided to use K-Means clustering , where we have to decide on the number of clusters. K-Means is a non-hierarchical clustering mechanism.

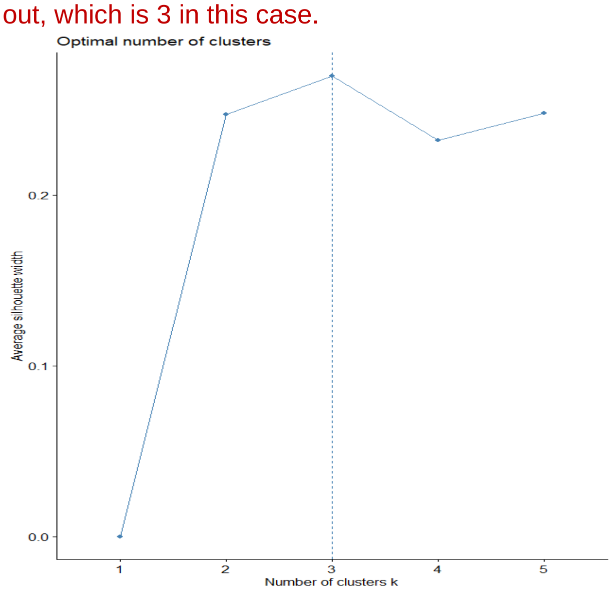
Rationale : The rationale for using a K-means , is that the data-set is quite large (5000 rows). If we use a hierarchical clustering it will be quite time consuming and tedious. Whereas in K-means clustering we can arrive at the number of defined clusters and then the rows or objects will get assigned to the designated clusters using a selected measure of distance (we have used Euclidean distance as the measure of distance). K-means non-hierarchical clustering is also relatively fast as compared to the hierarchical clustering.

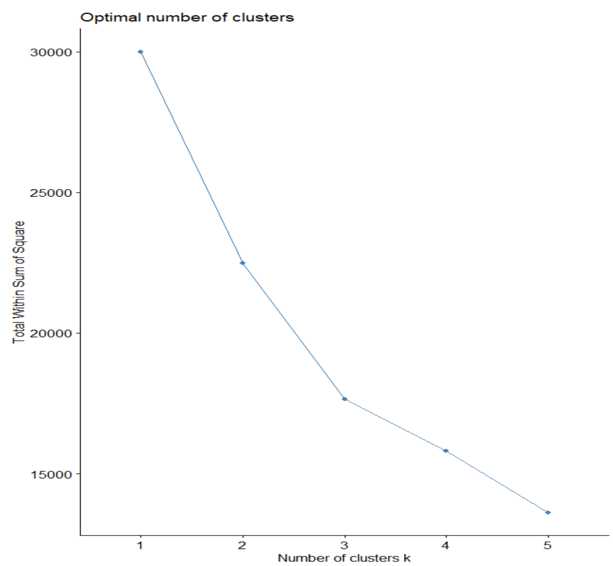
Checking Optimal Clusters : First we need to arrive at the optimal number of clusters, by identifying the number variables in each rows, scaling the cluster to do a fair clustering job and also calculating the Euclidean Distance . **Also we are considering only numerical variables for clustering purpose, as we calculate the Euclidean distances to form the clusters.**

```
mydata.clust = mydata %>% select_if(is.numeric)
>
> mydata.scale = scale(mydata.clust, center = TRUE)  #scaling the cluster
>
> mydata.dist = dist(mydata.scale, method = "euclidean") #calculating the euclidean
distance
```

```
> p12 = fviz_nbclust(mydata.scale, kmeans, method = "silhouette", k.max = 5) # k-mea
ns clustering is used
> p21 = fviz_nbclust(mydata.scale, kmeans, method = "wss", k.max = 5)
>
> grid.arrange(p12, p21, ncol=2)
```

Number of Clusters : The **number of clusters** which we need to arrive at **is 3 using the elbow method** as shown in the below graph. The number is clusters is taken as the point at which the graph plateaus out, which is 3 in this case.



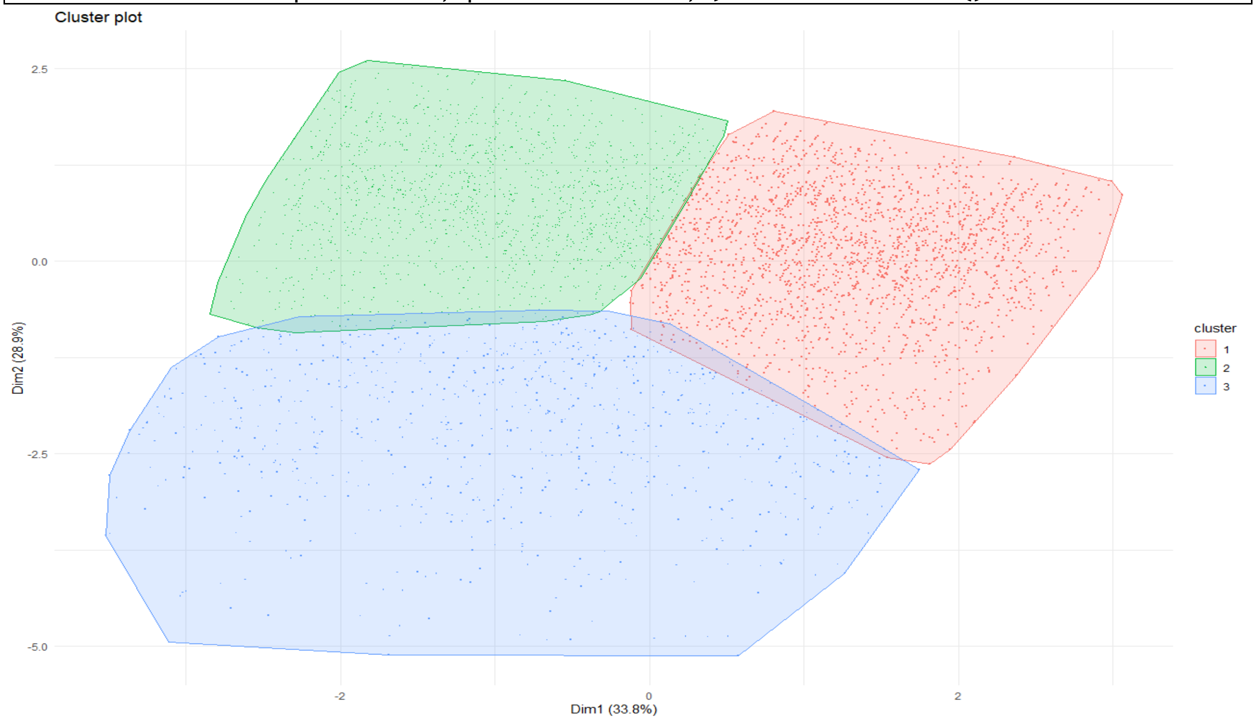


So now we need to run the K-Means clustering algorithm with 3 cluster centers and nstart as 10 times as shown below

K-Means Clustering : First need to set a seed and run as below :

```
set.seed(8787)
> mydata.clusters = kmeans(mydata.scale, 3, nstart = 10)
```

```
> fviz_cluster(mydata.clusters, mydata.scale, geom = "point",
+              ellipse = TRUE, pointsize = 0.2, ) + theme_minimal()
```



Clustering Output Interpretation :

Remarks : [for making a meaningful interpretation] :

- The 5000 rows data-set has been transformed to only 3 clusters which helps us in a great way.
- So the data has been divided into 3 distinct clusters which means it can be either on education class levels or income levels buckets of lower middle or higher income group. So intuitively, if education is the basis for clustering, then the educated customers can be targeted who will have good earning potential and thereby an aspiration to improve their lifestyle needs and financial requirement.

Dendrogram : Note that there is no dendogram involved here, since we have used K-means clustering which is a non-hierarchical clustering mechanism.

Insight :

K-Means Clustering Algorithm has made the existing 5000 row data-set simpler by partitioning i t into 3 manageable clusters which will help the marketing team to adopt targeted strategy for t he three class of customers.

‘ ’

III. Applying Supervised Learning Techniques (Test & Train) :

Splitting dataset into train and test data :

```
set.seed(1233)
```

We will Sample 70% of data for training the algorithms using random sampling method :

```
> mydata.index = sample(1:nrow(mydata), nrow(mydata)*0.70)
> mydata.train = mydata[mydata.index,]
> mydata.test = mydata[-mydata.index,]
```

30% of the rows is used for testing

```
> dim(mydata.test)
```

```
[1] 1500   12
```

The remaining 70% is taken for training

```
> dim(mydata.train)
```

```
[1] 3500   12
```

Here we check the ratio of personal loan taken in the ***training data*** *which is as below*

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

```
   0    1
3151  349
```

Now we check the ratio of personal loan taken in the ***test data*** *which is as below*

```
> table(mydata.test$`Personal Loan`)
```

```
   0    1
1369  131
```

Interpretation :

349 out of 3500 customers availed personal loan in the training data. And 131 out of 1500 customers availed personal loan in the test sample data. Table shown below :

Data Split	No Personal Loan (0)	Availed Personal Loan (1)	Total Sample Data
My Data – Training Sample	3151	349	3500
My Data – Test Sample	1369	131	1500

Applying CART Model :

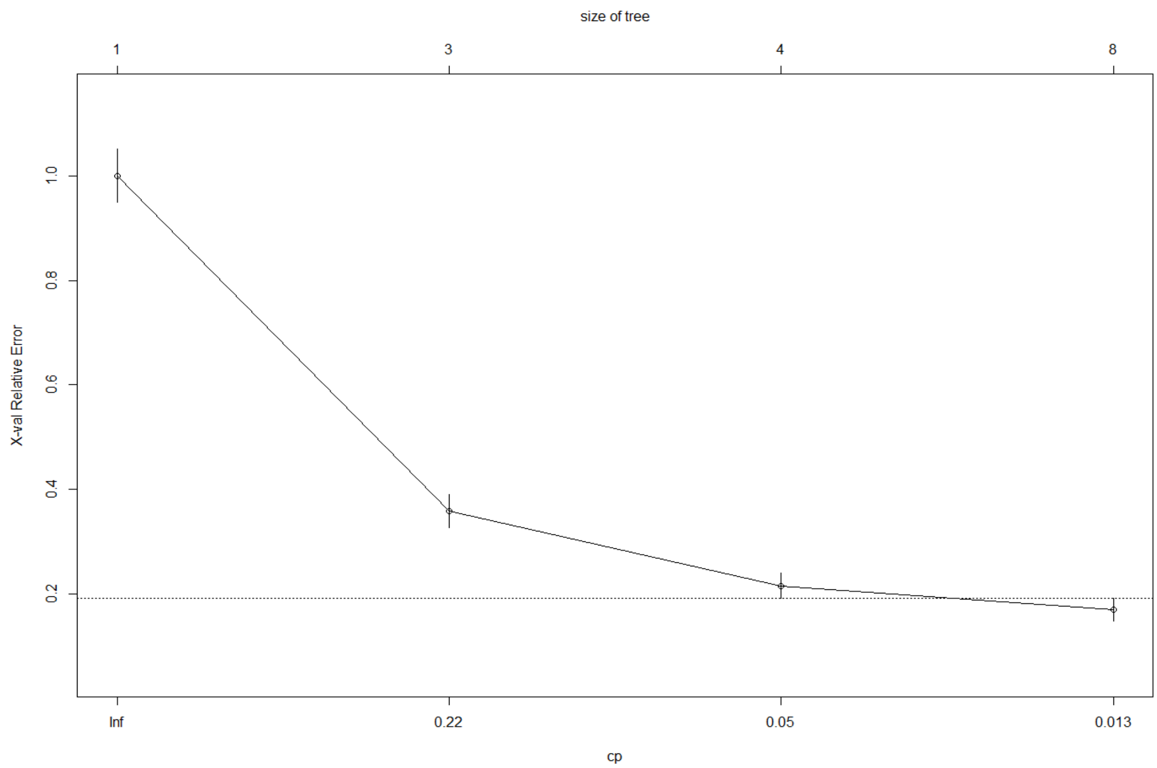
Classification and regression trees are the most popular predictive analytics techniques used

```
set.seed(233)
> cart.model.gini = rpart(mydata.train$`Personal Loan`~., data = mydata.train, metho
d = "class",
+                          parms = list(split="gini"))
```

Checking the Complexity Parameter :

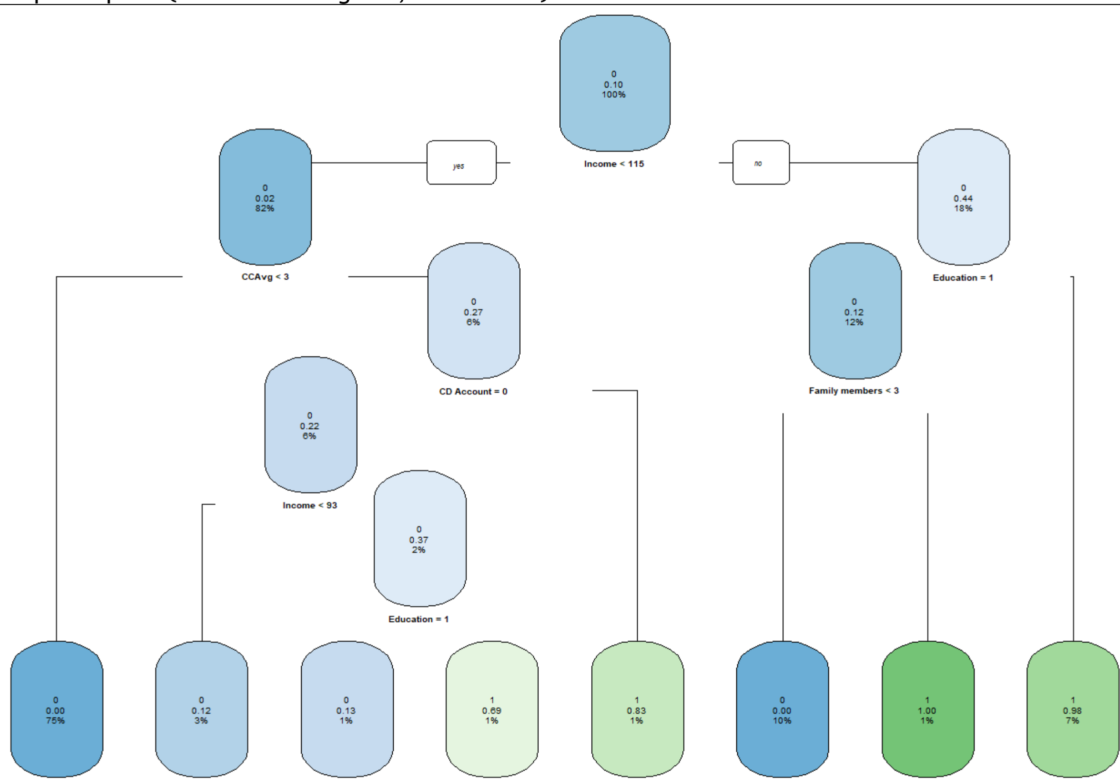
```
> plotcp(cart.model.gini)
```

The output of the complexity parameter would be as below :



Plotting the Tree : We will now plot the classification tree as below

```
> rpart.plot(cart.model.gini, cex =0.6)
```



Now, we create a Complexity Parameter table(*cptable*) to check or gauge the best cross-validated error out of the complexity parameter

```
> cart.model.gini$cptable
```

	CP	nsplit	rel error	xerror	xstd
1	0.32521490	0	1.0000000	1.0000000	0.05078991
2	0.14326648	2	0.3495702	0.3696275	0.03193852
3	0.01719198	3	0.2063037	0.2263610	0.02517855
4	0.01000000	7	0.1346705	0.1977077	0.02356543

Checking for the variable importance for splitting of the tree:

```
> cart.model.gini$variable.importance
```

```
Education     Income        Family members     CCAvg          CD Account
232.137107    188.541598    142.501489         106.606257     56.904176

Mortgage     Experience      Age          Online
27.306276    3.445512        3.437672     1.751040
```

Insight :
Now with this CART Model in place we are able to arrive at some conclusions which will help us understand the data better.

1. **Education, Income , Family Member, CC Avg and CD Account are important predictors on which data is split by tree algorithm**
2. **The CART TREE built as shown above also clearly reflects the importance of these predictors**
3. **First split happens on whether Income is less than or greater than $ 115K**
4. **Complexity parameter almost lowers to 0.05 (graph) with relative 0.2 as the cross validated error**

‘ ,

Interpreting the CART model Output :

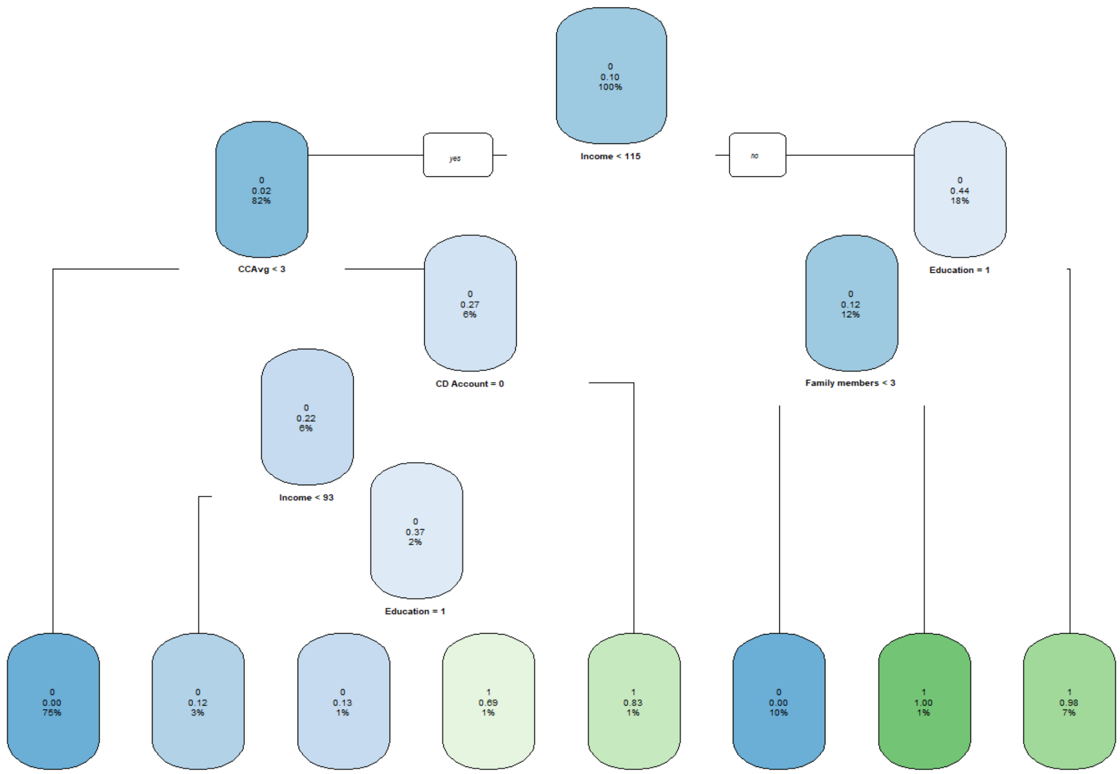
Pruning the tree : Pruning the Trees using the best complexity parameter

```
pruned.model = prune(cart.model.gini, cp = 0.015)
```

Remarks on Pruning : [for making a meaningful interpretation] :

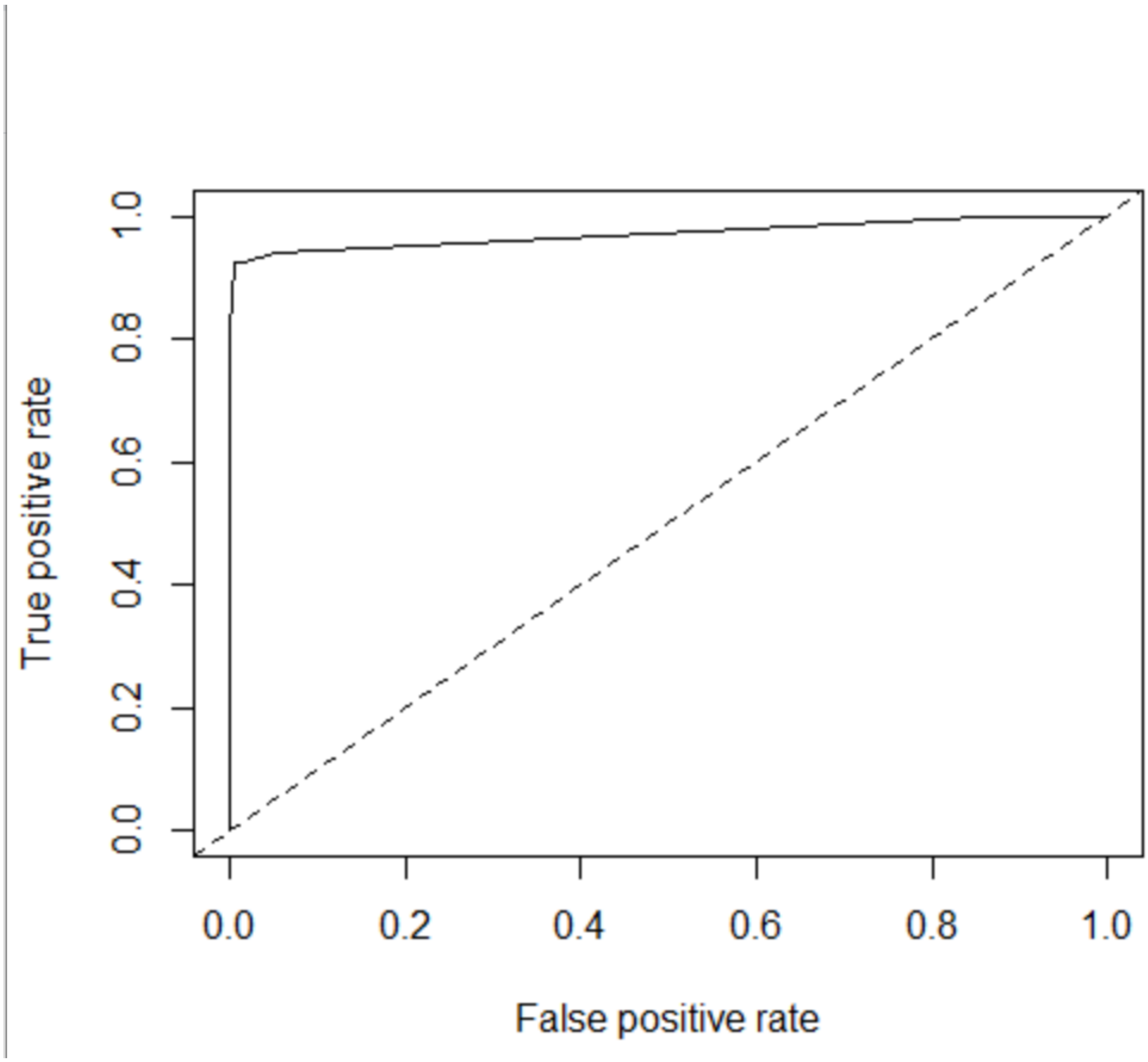
Plotting the Pruned Tree :

```
rpart.plot(pruned.model, cex=0.65)
```



ROC Curve for Pruned Tree :

```
library("ROCR")
cart.pred = predict(pruned.model, mydata.test, type = "prob")[,2]
pred2=prediction(cart.pred,mydata.test$`Personal Loan`)
plot(performance(pred2,"tpr","fpr"))
abline(0,1,lty=2)
```



Plotting Area Under Curve :
Since this is a loan prediction and we want to be more careful to weed out possible defaulters rather then

```
auc.tmp<-performance(pred2,"auc")
auc<-as.numeric(auc.tmp@y.values)
print(auc)
```

```
[1] 0.971105
```

Interpretation :
The area under the curve is approx. 97%. So we can infer that the CART Model has given 97.1 % accuracy in predicting the people who will take the personal loan (this is basis the test data)

KS :
KS is 91.78%

```
perf<-performance(pred2,"tpr","fpr")
KS<-max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])
KS
```

```
 [1] 0.9178204
```

CART Prediction :

```
> cart.pred = predict(pruned.model, mydata.test, type = "prob")
>
> cart.pred.prob.1 = cart.pred[,1]
> head(cart.pred.prob.1, 10)
```

```
     1          2          3          4          5
0.99734244 0.99734244 0.99734244 0.99734244 0.02109705
     6          7          8          9         10
0.31428571 0.02109705 0.99734244 0.02109705 0.99734244
```

Setting threshold & Using Confusion Matrix :
Since this is a loan prediction and we want to be more careful to weed out possible defaulters rather then deny the disbursal to deerving prospects We will set the threshold for probability as high as 0.70

All the predicted probabilities >=0.7 will be considered as class “1” and rest class “0”

Using the Confusison Matrix to gauge the performance of Models

Threshold :
we will set a threshold based on which the probability can be considered as “1”

```
> threshold = 0.70
> mydata.test$loanprediction = ifelse(cart.pred.prob.1 >= threshold, 1, 0)
>
> mydata.test$loanprediction = as.factor(mydata.test$loanprediction)
>
> Cart.Confusion.Matrix = confusionMatrix(mydata.test$loanprediction,
+                     reference = mydata.test$`Personal Loan`, positive = "1")
> Cart.Confusion.Matrix
Confusion Matrix and Statistics
```

Confusion Matrix and Statistics :

Please see the output of the confusion matrix :

```
Reference
Prediction    0     1
        0        8     121
        1     1361    10

              Accuracy       : 0.012
                95% CI         : (0.0071, 0.0189)
    No Information Rate  : 0.9127
    P-Value [Acc > NIR]  : 1

                 Kappa : -0.1738

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

           Sensitivity        : 0.076336
           Specificity        : 0.005844
        Pos Pred Value   : 0.007294
        Neg Pred Value   : 0.062016
            Prevalence      : 0.087333
        Detection Rate     : 0.006667
  Detection Prevalence : 0.914000
     Balanced Accuracy  : 0.041090

      'Positive' Class : 1
```

Insight :
We can see that even Pruned CART tree has very low accuracy of just 1.2 % **even after tuning it s complexity parameter.**

‘ ,

Applying Random Forests :

Random forest is an ensemble method used by combining weak and strong learners to give a better accuracy or output. Its a combination of multiple trees each chosen randomly to grow on dataset It makes use of the concept of averaging in the sense that the weak and strong learners combined produce better results rather than a single CART tree.

Two packages have been used to model the training dataset

1. Random Forest
2. Ranger (better than random forest)

Creating the Random Forest Model : We create the Random Forest Model as below :

```
set.seed(1233)
>
> RF.model = randomForest(formula = mydata.train$`Personal Loan`~
+                             (mydata.train$Age+mydata.train$Experience+mydata.train$Income
+                                mydata.train$`Family members`+mydata.train$CCAvg+
+                                mydata.train$Education+mydata.train$Mortgage+
+                                mydata.train$`Securities Account`+mydata.train$`CD Account
+                                mydata.train$Online+mydata.train$CreditCard),data = mydata
.train)
> print(RF.model)
```

```
Call:
 randomForest(formula = mydata.train$`Personal Loan` ~ (mydata.train$Age +      mydata.train$Experience + mydata.train$Income + mydata.train$`Family members` +      mydata.train$CCAvg + mydata.train$Education + mydata.train$Mortgage +      mydata.train$`Securities Account` + mydata.train$`CD Account` +      mydata.train$Online + mydata.train$CreditCard), data = mydata.train)

Type of random forest: classification
Number of trees: 500

No. of variables tried at each split: 3

OOB estimate of  error rate: 1.31%

Confusion matrix:
     0       1   class.error
0  3147      4   0.001269438
1  42      307   0.120343840
```

Printing the Error Rate :

```
> err = RF.model$err.rate
> head(err)
```

```
            OOB           0           1
[1,] 0.04441041  0.02815700  0.1865672
[2,] 0.03998119  0.02405858  0.1822430
[3,] 0.03709369  0.01994907  0.1930502
[4,] 0.03410641  0.01472810  0.2147887
[5,] 0.03294946  0.01560837  0.1921824
[6,] 0.02968176  0.01190476  0.1890244
```

Out of the Bag Error Rate :

```
> oob_err = err[nrow(err), "OOB"]
> print(oob_err)  ## depicts the final out of bag error for all the samples
```

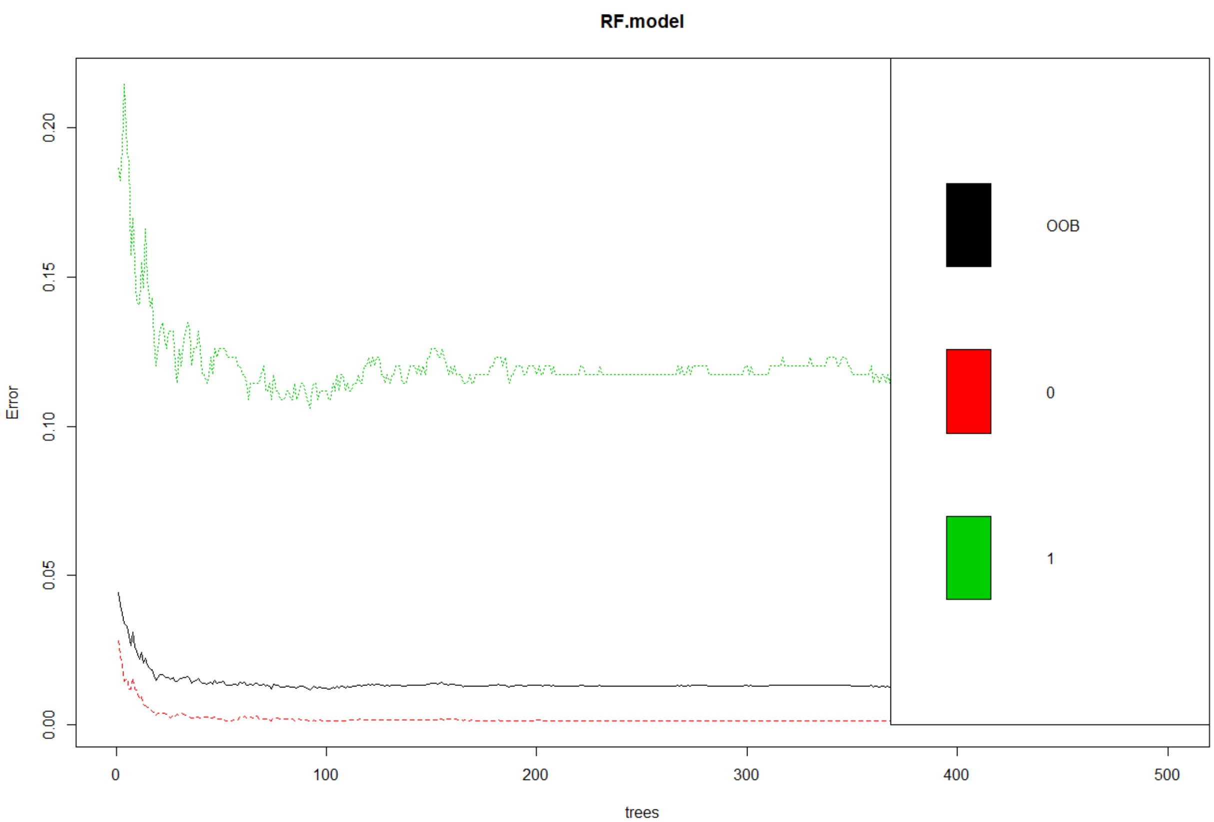
```
OOB
0.01314286
```

Following plot depicts the Out of Bag error for Class 0 and Class 1 and Overall OOB error.Also suggests the optimal trees we can use to tune Random forest model

Somewhere 100 trees should suffice as it saves time to train less trees and achieve same or even better results depending on cases

Plot the OOB Error :

```
> plot(RF.model)
> legend(x="topright", legend = colnames(err), fill = 1:ncol(err))
```



Interpreting the RF Model Output :

Remarks on the RF model output – Prediction on the Train Set for the Random Forest Model :

```
> RF.pred = predict(RF.model, mydata.train, type = "prob")[,1]
>
> mydata.train$RFpred = ifelse(RF.pred>=0.8,"1","0")
>
> mydata.train$RFpred = as.factor(mydata.train$RFpred)
>
> levels(mydata.train$RFpred)
```

```
[1] "0" "1"
```

```
> RFConf.Matx = confusionMatrix(mydata.train$RFpred, mydata.train$`Personal
Loan`, positive = "1")
> RFConf.Matx
```

Confusion Matrix and Statistics – Random Forest :

Please see the output of the confusion matrix :

```
          Reference
Prediction      0      1
         0      4    349
         1   3147      0

               Accuracy : 0.0011
                 95% CI : (3e-04, 0.0029)
    No Information Rate : 0.9003
    P-Value [Acc > NIR] : 1

                  Kappa : -0.2188

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

            Sensitivity : 0.0000000
            Specificity : 0.0012694
         Pos Pred Value : 0.0000000
         Neg Pred Value : 0.0113314
             Prevalence : 0.0997143
         Detection Rate : 0.0000000
   Detection Prevalence : 0.8991429
      Balanced Accuracy : 0.0006347

       'Positive' Class : 1
```

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

```
   0    1
3151  349
```

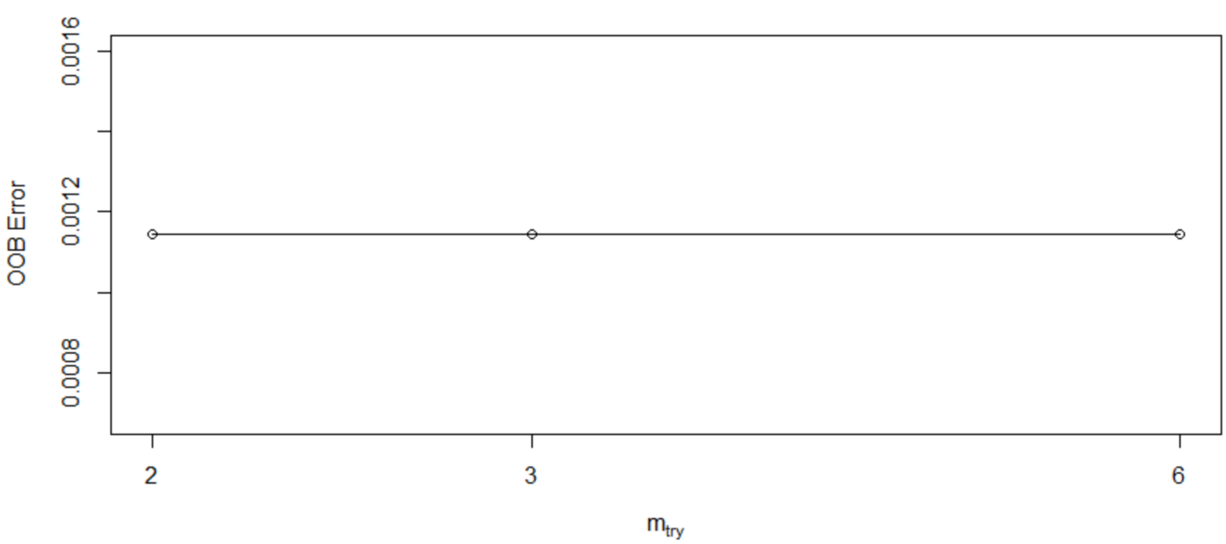
```
> table(mydata.test$`Personal Loan`)
```

```
   0    1 
1369  131 
```

Tuning the Random Forest Algorithm : Using the RF Function in Random Forest Algorithm to get some idea about improving the performance of the model

```
> set.seed(333)
>
> tuned.RandFors = tuneRF(x = mydata.train[,-8],
+                         y= mydata.train$`Personal Loan`,
+                         ntreeTry = 501, doBest = T)
```

```
    mtry = 3  OOB error = 0.11%
Searching left ...
mtry = 2    OOB error = 0.11%
0 0.05
Searching right ...
mtry = 6    OOB error = 0.11%
0 0.05
```




```
> print(tuned.RandFors)
```

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

        OOB es`timate of  error rate: 0.11%
Confusion matrix:
     0   1 class.error
0 3147   4 0.001269438
1    0 349 0.000000000
```

Interpreting the RF Model Output with ntree =100 :

Number of trees , we take as 100 here, as we saw above, it may give a better model for prediction

Remarks on the RF model output – Prediction on the Test & Train Set for the Random Forest Model :

```
trainIndex<-createDataPartition(data$`Personal Loan`,
                                p=0.7,
                                list = FALSE,
                                times = 1)
base_data_2<-data[,-5]
train.data <-base_data_2[trainIndex,2:length(base_data_2)]
colnames(train.data)<-c('Age_in_years','Experience_years','Income_Monthly',
'Family_members','CCAvg','Education','Mortgage',
                        'Personal_loan','Securities_Account','CD_Account','
Online','CreditCard')
train.data$Personal_loan<-as.factor(train.data$Personal_loan)
train.data<-na.omit(train.data)
test.data  <- base_data_2[-trainIndex,2:length(base_data_2) ]
colnames(test.data)<-c('Age_in_years','Experience_years','Income_Monthly','
Family_members','CCAvg','Education','Mortgage',
                       'Personal_loan','Securities_Account','CD_Account','O
nline','CreditCard')
test.data<-na.omit(test.data)
test.data$Personal_loan<-as.factor(test.data$Personal_loan)
model1 <- randomForest(Personal_loan ~ ., ntree = 100,data = train.data, im
portance = TRUE)
model1
```

```
Call:
 randomForest(formula = Personal_loan ~ ., data = train.data,      ntree =
100, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 3

        OOB estimate of  error rate: 1.35%
Confusion matrix:
     0   1 class.error
0 3163   6 0.001893342
1   41 275 0.129746835
```

```
Pred_rf <- predict(model1, test.data, type = 'class')
confusionMatrix(test.data$Personal_loan, Pred_rf)
```

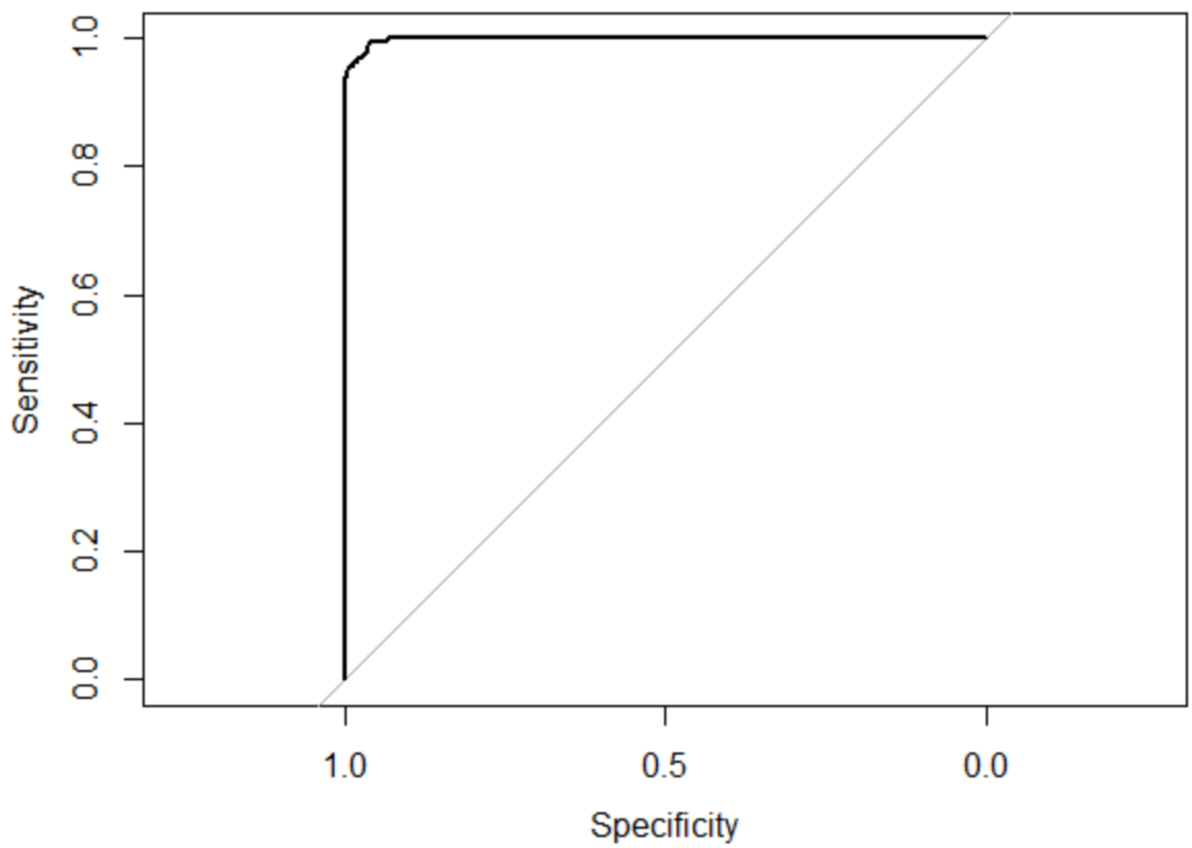
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1334    1
         1   15  147

               Accuracy : 0.9893
                 95% CI : (0.9827, 0.9939)
    No Information Rate : 0.9011
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9424

 Mcnemar's Test P-Value : 0.001154

            Sensitivity : 0.9889
            Specificity : 0.9932
         Pos Pred Value : 0.9993
         Neg Pred Value : 0.9074
             Prevalence : 0.9011
         Detection Rate : 0.8911
   Detection Prevalence : 0.8918
      Balanced Accuracy : 0.9911

       'Positive' Class : 0
```

Insight :
In the above confusion matrix, TP = 147, TN=1334 and total is FP =15, FN=1. So Accuracy = TP+ TN / TN+FN+FP+TP, which is 147+1334/1497 = 98.93% accuracy

Random forest has perfomed very well with 99.9% accuracy on the test data

ROC Curve for RandomForest :

```
library("ROCR")
Pred_rf <- predict(model1, test.data, type = 'prob')[,2]
require(pROC)
rf.roc<-roc(test.data$Personal_loan,Pred_rf)
plot(rf.roc)
```

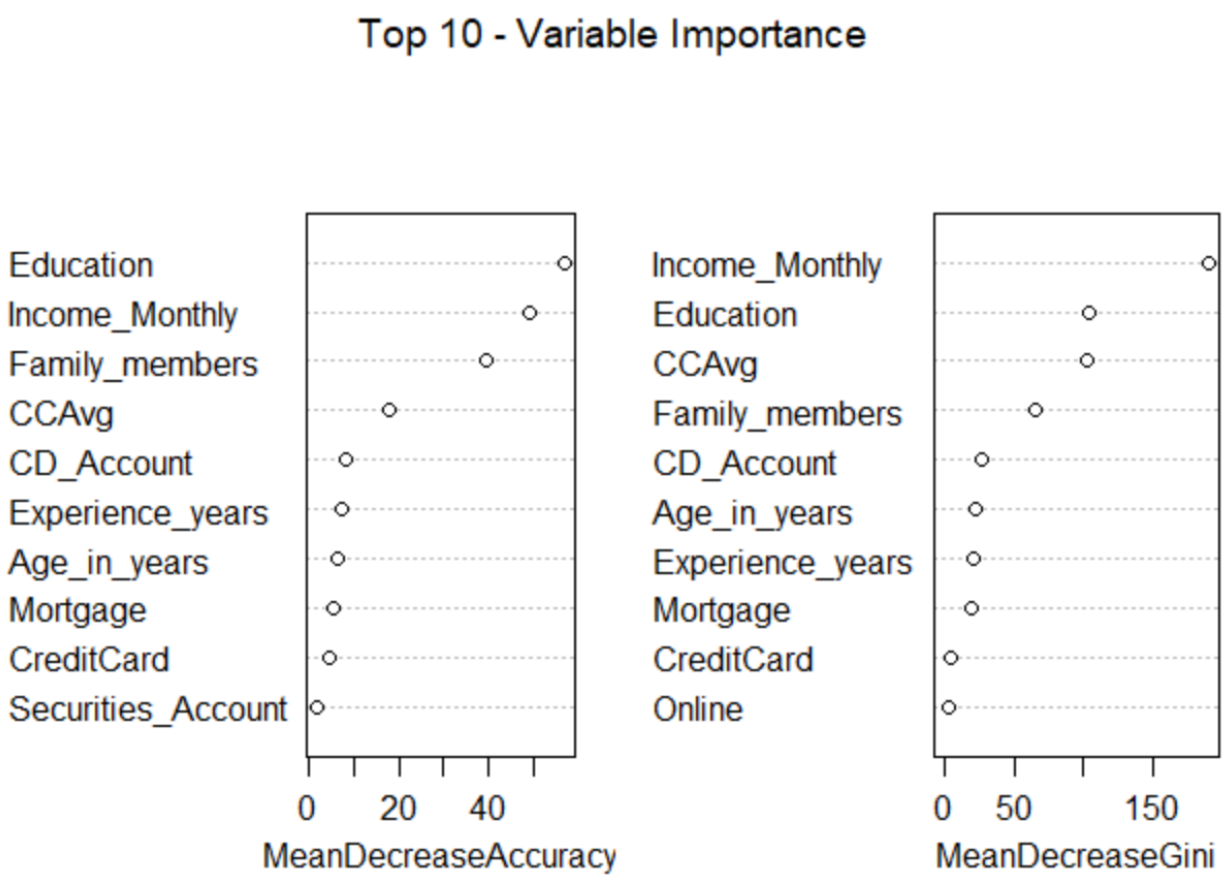



```
auc(rf.roc)
```

```
Area under the curve: 0.9984
```

Insight :
Its an ideal curve with AUC 99.84%

```
varImpPlot(model1,
           sort = T,
           n.var=10,
           main="Top 10 - Variable Importance")
```



Insight : Monthly Income and Education are the most significant factor that decides personal loan

, ,

IV. Model Performance Measures (Test & Train) :

Confusion Matrix Interpretation :

We have shown in the sections above the interpretation of confusion matrix in CART Model as wel l as the Random Forest Model

Interpretation of other Model Performance Measures :

KS, AUC and GINI : All three has been explained in the preceding paragraphs

Remarks on Model Validation Exercise :

We saw in the preceding paragraphs in this document regarding Clustering, CART Model and Random Forest Model

Clustering : We applied the Unsupervised clustering technique on the dataset which gave us 3 distinct cl usters, meaning 3 would have been the optimum number of clusters (using the ELBOW Method) and we c ould also intuitively make out that probably people spending higher on their credit card or people having a higher income and education could opt for or go for a personal loan if approached.
However intuition alone would not work and we need a model which would be able to test or validate.

Then we tried out the Supervised Learning methods as below :

CART Model : in the CART Model, we created a train data of 70% and a test data of 30% of the dataset provided. We checked the complexity parameter, plotted the classification tree . Using the complexity par ameter we gauged the best cross validated error . Then when we checked the "variable importance" for s plitting the tree, we got **Education, Income, Family members and CCAvg** as the prominent variables to split the tree. Complexity parameter almost lowers to 0.05 (graph) with relative 0.2 as the cross validated error. So with CART Tree, we got to know the importance of these predictor variables.

The area under the curve is approx. 97%. But the accuracy under the CART Model was not so encouraging

RANDOM Forest Model : However if you see, when we come to Random Forest model again with the tra in and test data, we could infer that it can bring more accuracy. The Confusion Matrix shows an accuracy of Predicted Positive value of 99.93% for the test data. The AUC for the test data is 99.84%. When we plo tted the Variable importance, **Education and Monthly Income proved to be the most significant facto r impacting the likelihood of people going for a Personal Loan.**

Best Performed Model : So we can say that the RANDOM FOREST MODEL is the best performed model for accuracy, however clustering technique was also useful while working on this data set.

Model Building using the Algorithm in the last step :
Interpretation of Results : xxxxxxxxx
I
So when we have a large data set and we need to get an accurate model built, Random Forest is a good option, although we need to keep a control on the optimum number of trees we will use to build the forest

Insight :

As per the confusion matrix, table in the Random Forest Model, we have an accuracy of 98.93%

TP = 147, TN=1334 and total is FP =15, FN=1.
So Accuracy = TP+TN **/** TN+FN+FP+TP,
which is 147+1334/1497 = **98.93% accuracy**

So the last model Random forest which we used in the last step has performed very well with 99.9% accuracy on the test data

‘ ,

………………………………………………….x End of Document x…………………………………………………..