King Saud University
College of Computer and Information Sciences
Information Technology department

IT 326: Data Mining

Course Project

Bank Personal Loan Modelling

Project Report

Group #: **1** LAB Day-Time: **Wed - 8 AM**

Group members:

Name	ID	Section
Deema AlFuaim		
Razan AlDhafian		
Yara AlManea		

[**5 / 8 / 2022**]

1 Problem

Personal loans are considered a major revenue generating method for banks. Therefore, to campaign for their personal loans, banks reach out to potential customers randomly. Hence, this campaign ends up as annoying marketing calls rather than being an effective means for accepting a personal loan.

In our problem, we have introduced a bank (Thera Bank) which has a growing customer base. In this bank, most of the customers are liability customers ,or depositors, with varying sizes of deposits. In addition, the number of customers who are borrowers , known as asset customers, is quite small . To conclude, the dataset is studying the ways of converting its liability customers to personal loan customers while keeping them as depositors .

Since the bank is interested in expanding the loan interest base rapidly. It will do so by encouraging the marketing department to devise campaigns that better target the customers . Hence, this will increase the success ratio of the personal loan campaign with minimal budget, therefore, bringing in an increased profit through the loan interest.

2 Data Mining Task

In solving the problem, the **data mining tasks** used will be **Classification and Clustering**.

In classification, we will seek to identify the potential customers willing to purchase the loan**.** Therefore, the class attribute will be the probability of customers to accept a personal loan {high probability, low probability}, using the **"personal loan" attribute as a class label**. The **goal of classification** is to accurately predict the target class for each customer in the data, thus, **predicting the likelihood of a liability customer buying personal loans**.

In clustering , we will partition a set of numeric data objects into subsets (clusters) , where objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. The **goal of clustering** is to cluster the data with minimal overlap such that the intra class similarity is maximized and the inter class similarity is minimized ,hence, we will **study the customers' behavior in order to group them based on their similar characteristics.**

3 Data

Our Source: https://www.kaggle.com/krantiswalke/bank-personal-loan-modelling
Number of objects: 5000
Number of attributes: 14

Attributes	Data Type	Possible Values
ID	int (nominal)	1 - 5000
Age	int (numeric ratio)	23 - 67
Experience	int (numeric ratio)	(-3) - 43
Income	int (numeric ratio)	8 - 224
ZIP Code	int (nominal)	9307 - 96.7k
Family	int (ordinal)	1 - 4
CCAvg	dec (numeric ratio)	0 - 10
Education	int (ordinal)	1 - 3
Mortgage	int (numeric ratio)	0 - 635
Personal Loan	int (Binary)	0 - 1
Securities Account	int (Binary)	0 - 1
CD Account	int (Binary)	0 - 1
Online	int (Binary)	0 - 1
Credit card	int (Binary)	0 - 1

- ID: Customer ID
- Age: Customer's age in completed years
- Experience: #years of professional experience
- Income: Annual income of the customer ($000)
- ZIP Code: Home Address ZIP code.
- Family: the 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: Do customers use internet banking facilities?
- Credit card: Does the customer use a credit card issued by UniversalBank?

➔ Missing Values:

```
> sum(is.na(dataset))
[1] 0
```

➔ Statistical Measures:

```
> summary(dataset)
       ID              Age            Experience       Income          ZIP.Code         Family     
 Min.   :   1   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   : 9307   Min.   :1.000  
 1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:91911   1st Qu.:1.000  
 Median :2500   Median :45.00   Median :20.0   Median : 64.00   Median :93437   Median :2.000  
 Mean   :2500   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :93152   Mean   :2.396  
 3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:94608   3rd Qu.:3.000  
 Max.   :5000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :96651   Max.   :4.000  
     CCAvg          Education       Mortgage     Personal.Loan   Securities.Account   CD.Account      
 Min.   : 0.000   Min.   :1.000   Min.   :  0.0   Min.   :0.000   Min.   :0.0000     Min.   :0.0000  
 1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1st Qu.:0.000   1st Qu.:0.0000     1st Qu.:0.0000  
 Median : 1.500   Median :2.000   Median :  0.0   Median :0.000   Median :0.0000     Median :0.0000  
 Mean   : 1.938   Mean   :1.881   Mean   : 56.5   Mean   :0.096   Mean   :0.1044     Mean   :0.0604  
 3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0   3rd Qu.:0.000   3rd Qu.:0.0000     3rd Qu.:0.0000  
 Max.   :10.000   Max.   :3.000   Max.   :635.0   Max.   :1.000   Max.   :1.0000     Max.   :1.0000  
     Online        CreditCard    
 Min.   :0.0000   Min.   :0.000  
 1st Qu.:0.0000   1st Qu.:0.000  
 Median :1.0000   Median :0.000  
 Mean   :0.5968   Mean   :0.294  
 3rd Qu.:1.0000   3rd Qu.:1.000  
 Max.   :1.0000   Max.   :1.000
```

➔ **Outliers:**

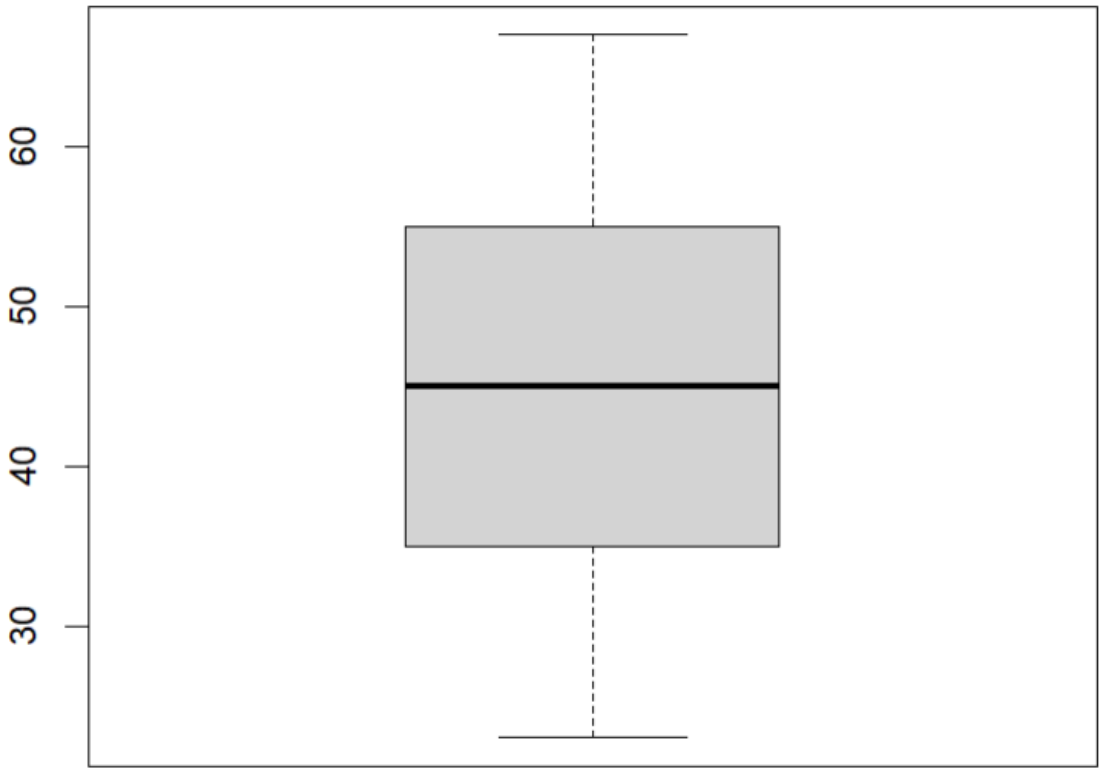
```
> boxplot(dataset$Experience)$out
numeric(0)
> boxplot(dataset$Age)$out
numeric(0)
> boxplot(dataset$CCAvg)$out
  [1]  8.90  8.10  5.70  8.00  5.70  5.60  7.20  7.40  7.50  6.50  6.50  7.80  7.90
 [33]  5.70  8.30  5.50  6.90  6.10  6.00  8.00  6.80  6.33  7.80  7.20  6.50  6.80
 [65]  6.00  5.90  8.10  5.40  8.80  5.40  6.33  8.10  8.80  5.70  7.60  7.30  7.00
 [97]  6.90  9.00  6.00  8.60  5.90  5.40  7.40  6.33  6.80  5.40  7.30  7.40  6.70
[129]  8.60  8.30  5.80  7.80  6.00  5.40  7.40  8.10  6.67  6.00  6.33  6.00  6.50
[161]  6.80  5.90 10.00  7.50  5.60  6.67  6.10  7.50  8.00  6.10  6.70  8.80  7.40
[193]  8.00  6.10  5.70  5.40  7.20  8.80  7.00  6.50  7.90  6.30  6.90  7.60  6.00
[225]  6.90  8.00  7.50  6.10  5.40  8.80  7.80  6.00  5.33  7.20  8.60  6.70  8.00
[257]  5.40  6.67  6.80  8.80  7.80  6.50  9.00  7.20  6.67  6.00  9.30  5.60  7.50
[289]  7.40  6.00  6.10  6.40  6.00  8.10  5.60  6.30  7.30  8.50  5.30  6.00  5.40
[321]  7.50  8.60  5.30  6.67
> boxplot(dataset$Income)$out
 [1] 193 194 190 188 195 191 200 205 204 195 192 194 202 195 200 193 192 195 191 188
[50] 204 198 201 201 191 191 195 190 188 190 195 195 205 198 190 191 191 195 194 194
>
```

➔ **Boxplots:**

<u>Age boxplot:</u>

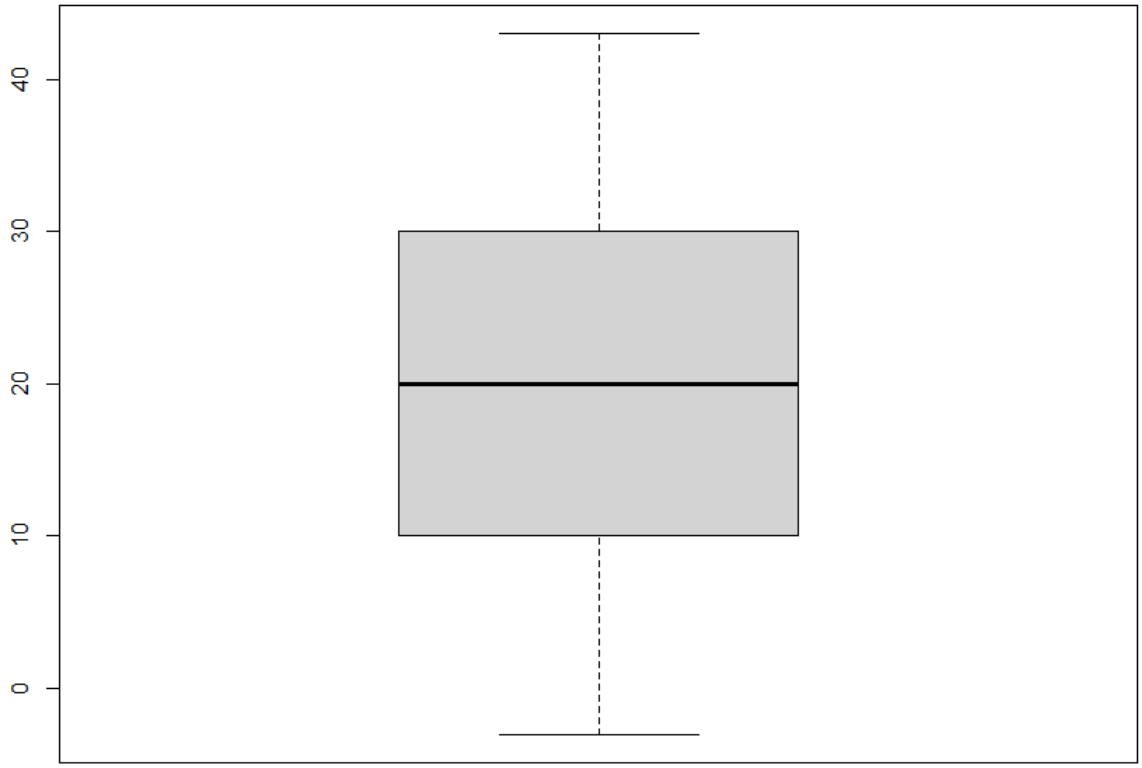
After plotting the age attribute, we noticed that most of the customers mainly range from 35 - 55 years old ; as indicated by the boundaries of the 1st and 3rd quartile as well as the median(2nd Quartile) which is estimated to be 45 years old by looking at the plot.

- *We can divide the age attribute into partitions such that anyone ranging from 20-29 will be classified as 20's , and so on with 30's,40's ,50's. This method is called* ***discretization*** *and* ***will help make the data processing simpler ,understandable, and easy to deal with.***



Experience boxplot:

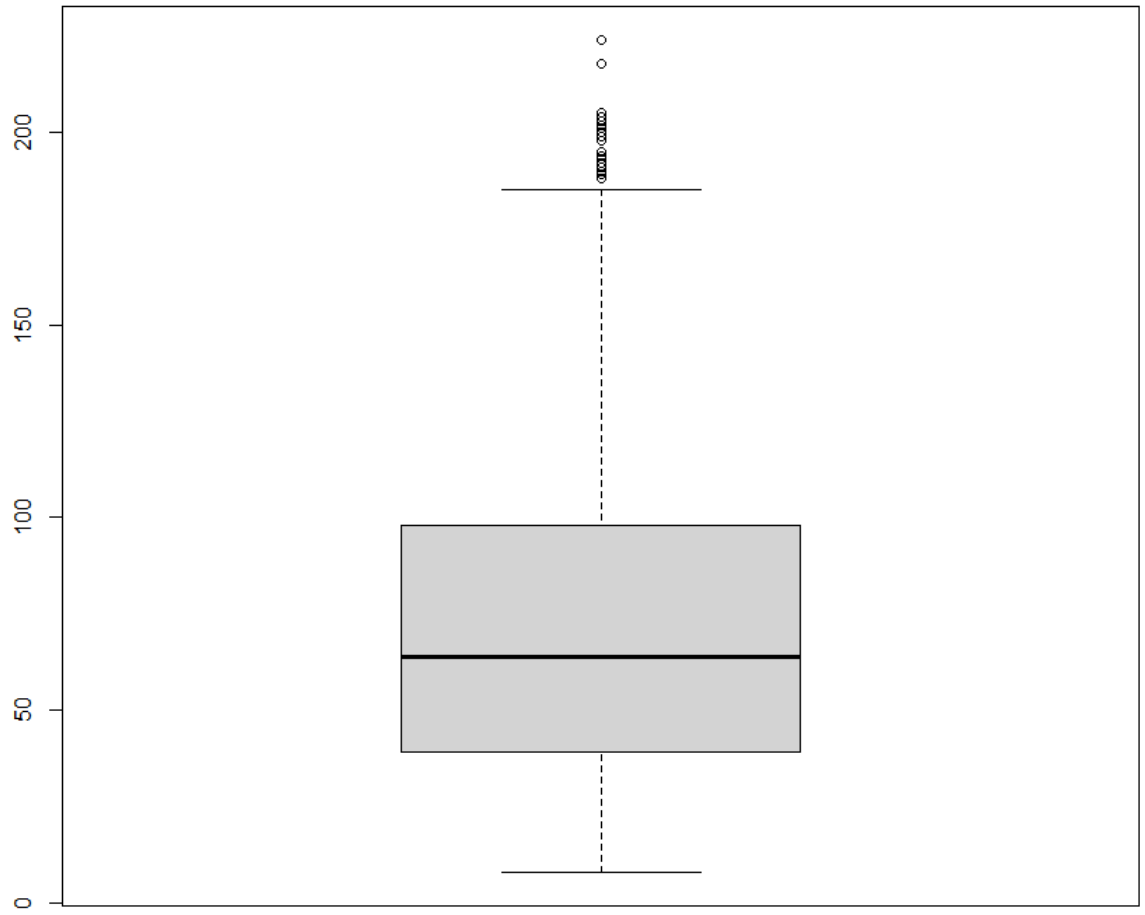
- From the graph, we noticed that there exists negative values which are inconsistent, since the attribute refers to the number of years of professional experience, which can't be negative. ***Therefore, we decided to smooth the data using bin means.***



Income boxplot:

From the plot, we can tell that there are outliers deviating from the possible values, as signified by the income range which is greater than approximately 200k annually(higher than 1.5*IQR of the 3rd quartile). Moreover, the annual income range for the majority of customers ranges from 40k to 100k as the 1st and 3rd quartile portray, whereas the median falls at almost 65k.

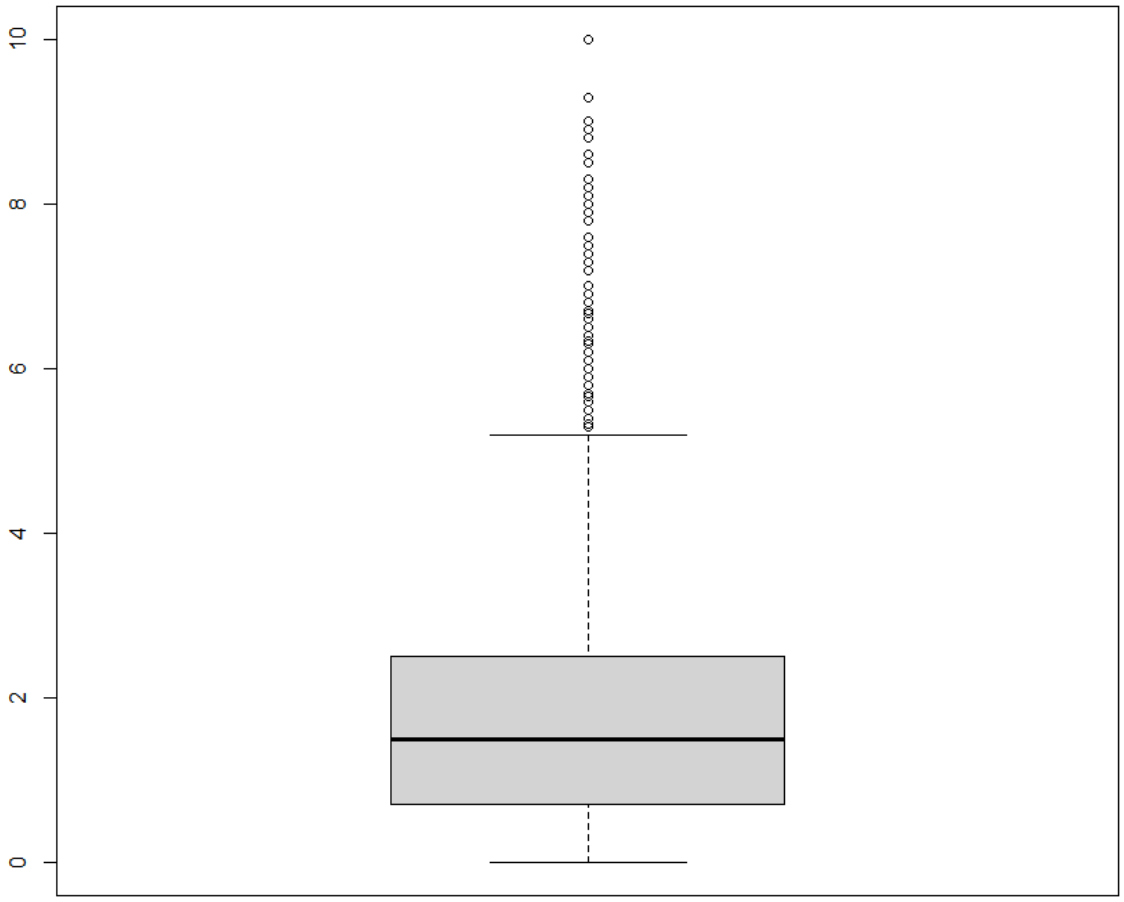
- *Because the Income ranges to 200K, this is considered a significant number that might affect how the data is distributed, thus, making analysis ambiguous.* ***Hence, we decided to normalize it.***



CCAvg boxplot:

In the below plot, it shows that the Avg. spending of customers on credit cards per month mostly lies between 0.5k to 2.5k, there are outliers to the attribute which have an Avg. spending of 5.5k reaching to 10k per month

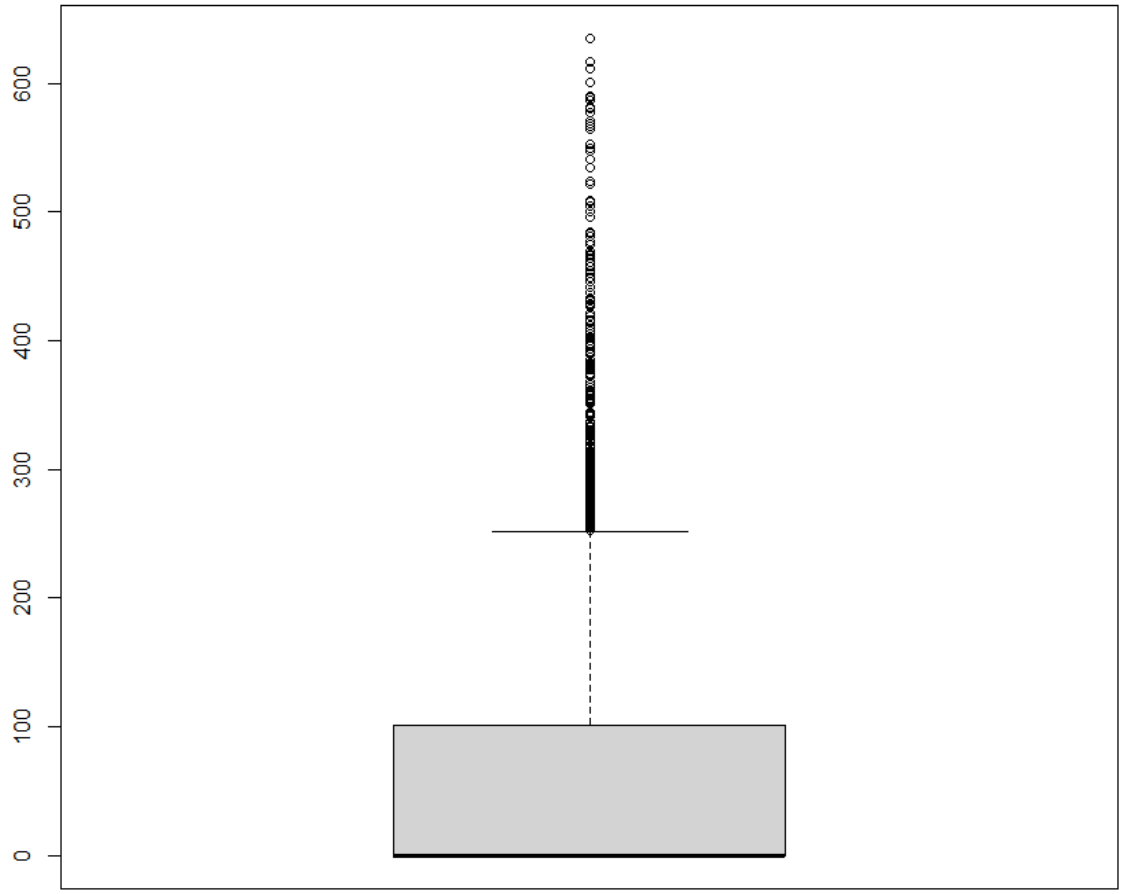
- ***There are many outliers, therefore, we have to apply data cleaning techniques.***



Mortgage boxplot:

By looking at the plot, one can tell that the median and 1st quartile are equal ; meaning that the value of house mortgage for most of the customers begins from 0k and extends to a 100k, there are huge numbers of outliers in this attribute shown by mortgage values going from 250k to 630k .

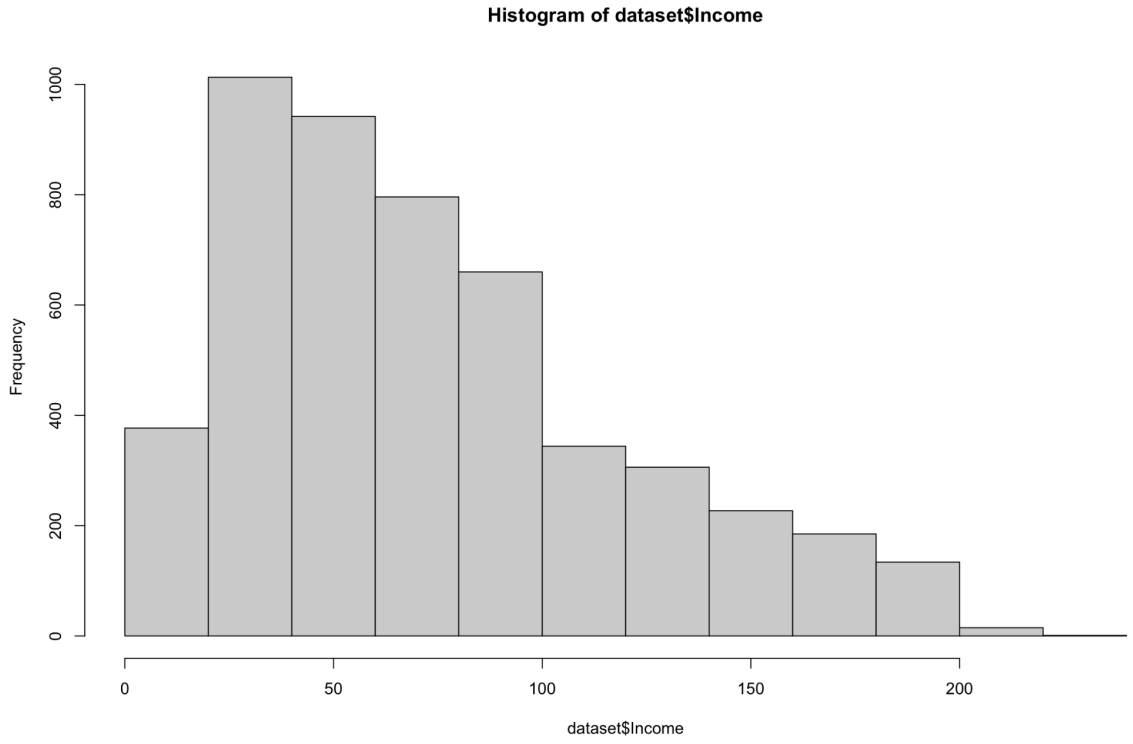
- *The mortgage range is quite huge which might affect how the data is distributed and will make analysis ambiguous too.* ***Hence, we decided to normalize it.***



➔ Histogram *:*

This chart shows the frequency of each customer's total income in the dataset.
Through the histogram, it was found that most of the customers have an annual income of 25k.

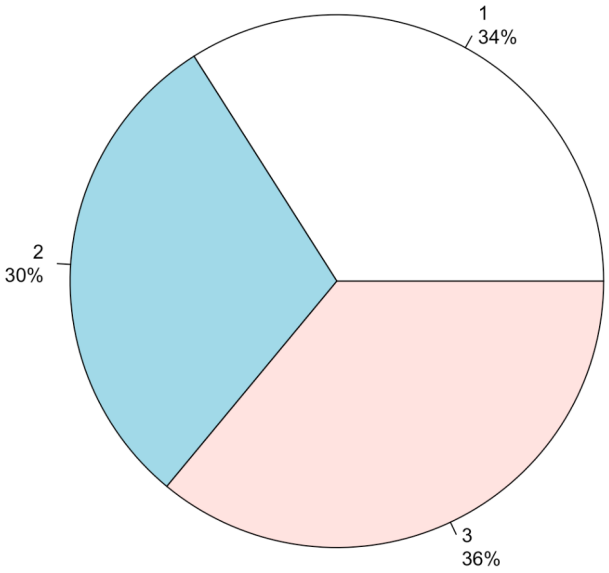
-We decided to check for duplicate tuples to see if the highest frequency resulted from it



➔ Pie Chart :

The below pie chart illustrates the percentage of each education level for all customers and how it is distributed amongst them. In the chart, the highest percentage (36%) is for level (3) which is the Advanced/ Professional level . In contrast, the lowest percentage (30%) is for level (2) which is Graduate, whereas the middle percentage (34%) is for level (1) which is Undergrad.

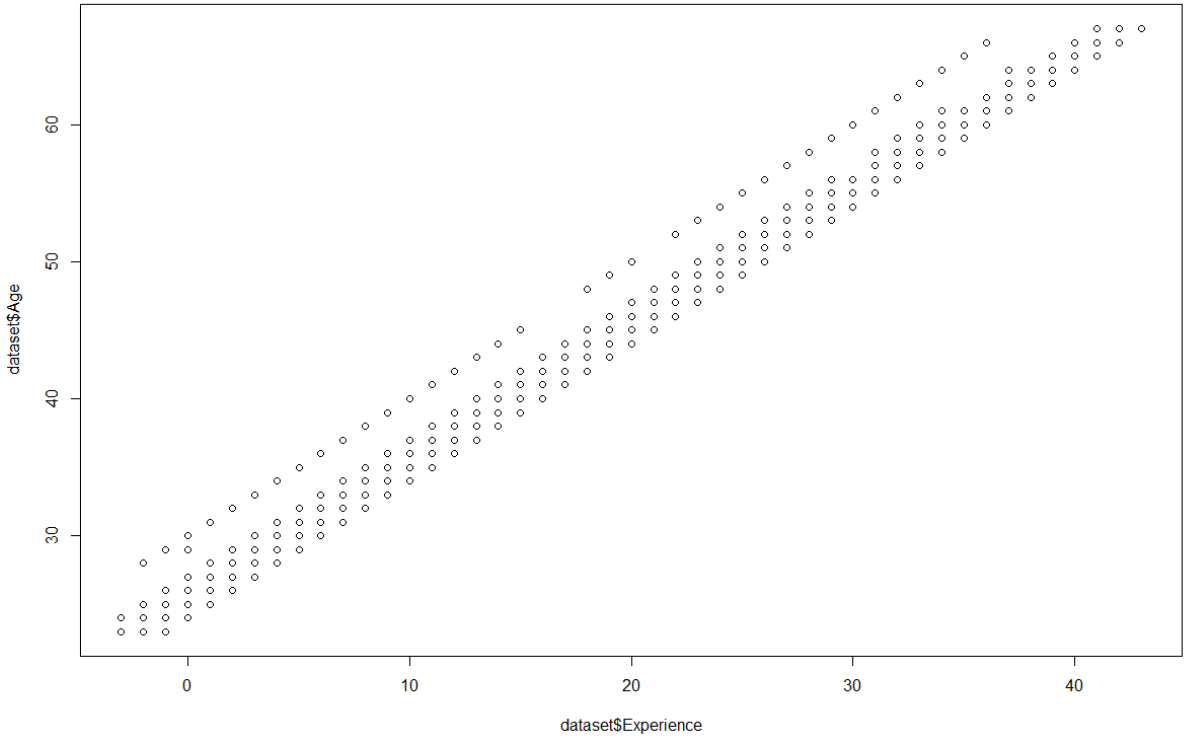
- ***We decided to check for duplicate tuples to see if the highest frequency resulted from it.***



➔ Scatter :

The below scatter plot portrays the relation between age and experience. From the figure we can see that as the number of years of professional experience increases the age also increases correspondingly.

- ***The scatter plot did not shed light on a preprocessing method to apply.***



4 Data preprocessing

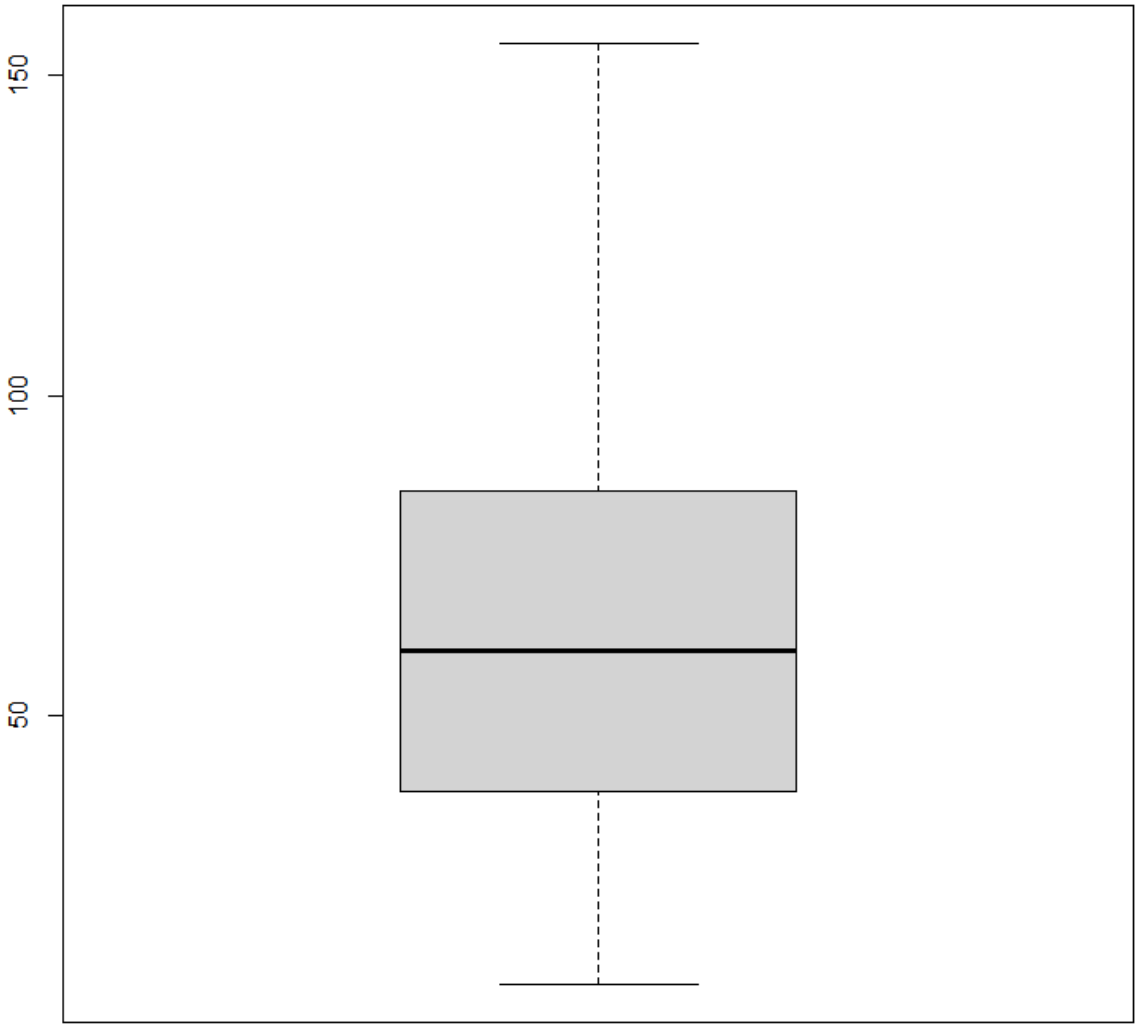
➔ Data cleaning :

1) Removing outliers:

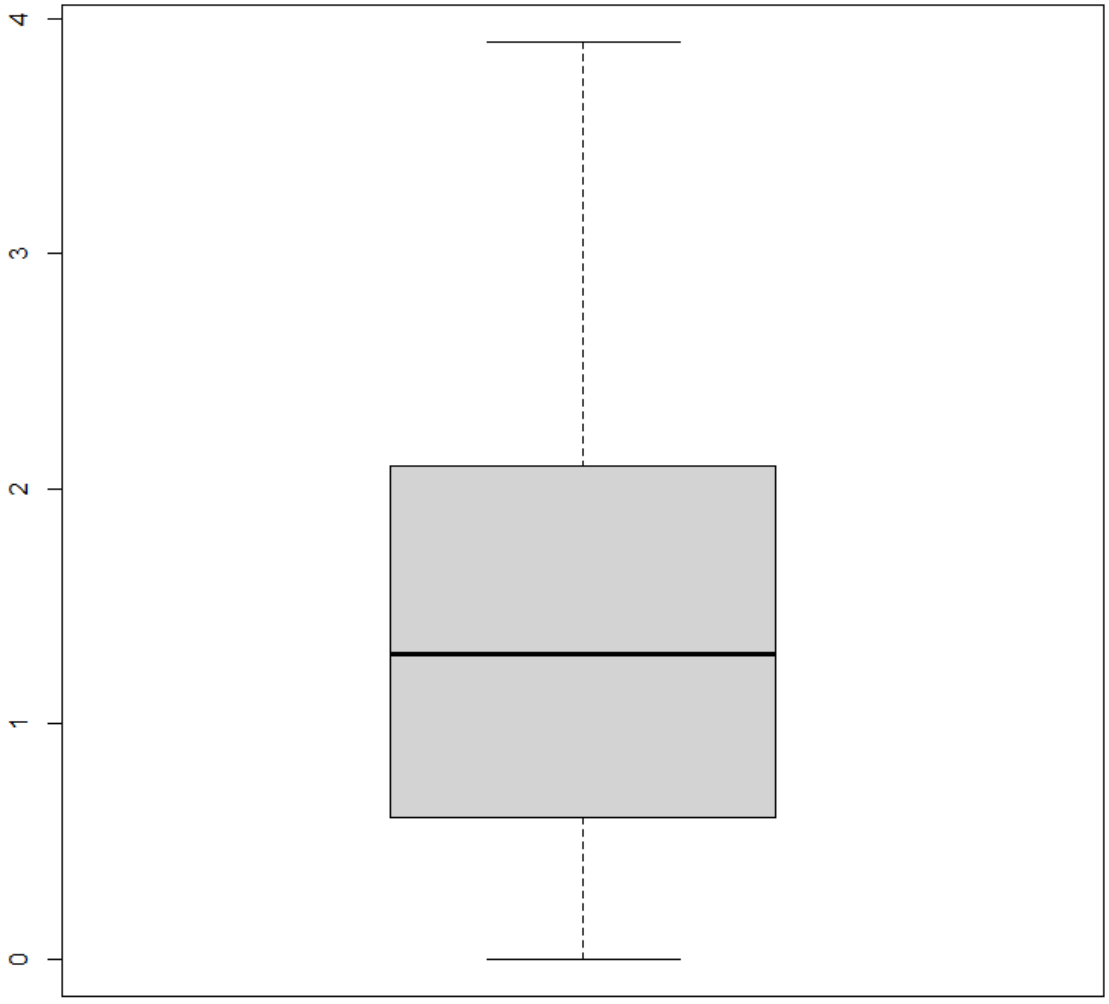
We eliminated outliers to make the outcome statistically significant, such that results of data analysis are not affected by these outliers . In addition, we dropped the mortgage attribute; as it generated issues with the removal of outliers since it contains many zero values that affected the q1 and q3, however it doesn't have an impact on our model.

```
#mortgage drop
dataset <- dataset[,-c(1,5,9)]
#income
boxplot(dataset$Income)$out #? yes
dataset=subset(dataset, Income<=155)
boxplot(dataset$Income)
boxplot(dataset$Income)$out #=0
#ccavg
boxplot(dataset$CCAvg)$out #? yes
boxplot(dataset$CCAvg)
dataset=subset(dataset, CCAvg<=3.9)
boxplot(dataset$CCAvg)$out #=0
```

- **Boxplot after removing outliers (Income)**



- **Boxplot after removing outliers (CCAvg)**



2) Smoothing by bin means:

We smoothed out the data to get rid of any existing noisy data that impacts how data is analyzed. In our case, the noisy data was negative values.

```
no_of_bins <- 5
round(ave(dataset$Experience, rep(1:length(dataset$Experience), each = no_of_bins, length.out = length(dataset$Experience))))
```

```
> round(ave(dataset$Experience, rep(1:length(dataset$Experience), each = no_of_bins, length.out = length(dataset$Experience))))
   [1] 10 10 10 10 10 17 17 17 17 17 28 28 28 28 28 22 22 22 22 22 18 18 18 18 18 20 20 20 20 20 18 18 18 18 18 23 23 23
  [39] 23 23 17 17 17 17 17 20 20 20 20 20 16 16 16 16 16 17 17 17 17 17 21 21 21 21 21 29 29 29 29 29 17 17 17 17 17 23
  [77] 23 23 23 23 21 21 21 21 21 16 16 16 16 16 26 26 26 26 26 24 24 24 24 24 26 26 26 26 26 12 12 12 12 12 23 23 23 23
 [115] 23 25 25 25 25 25 23 23 23 23 23 16 16 16 16 16 14 14 14 14 14 32 32 32 32 32 14 14 14 14 14 26 26 26 26 26 24 24
 [153] 24 24 24 12 12 12 12 12 16 16 16 16 16  8  8  8  8  8 19 19 19 19 19 23 23 23 23 23 14 14 14 14 14 25 25 25 25 25
 [191] 27 27 27 27 27 16 16 16 16 16 17 17 17 17 17 20 20 20 20 20 25 25 25 25 25 16 16 16 16 16 16 16 16 16 16 16 16 16
 [229] 16 16 18 18 18 18 18 20 20 20 20 20 24 24 24 24 24 17 17 17 17 17 27 27 27 27 27 24 24 24 24 24 17 17 17 17 17 27
 [267] 27 27 27 27 15 15 15 15 15 14 14 14 14 14 21 21 21 21 21 17 17 17 17 17 15 15 15 15 15 20 20 20 20 20 23 23 23 23
 [305] 23 26 26 26 26 26 22 22 22 22 22 17 17 17 17 17 31 31 31 31 31 25 25 25 25 25 26 26 26 26 26 17 17 17 17 17 18 18
 [343] 18 18 18 13 13 13 13 13 19 19 19 19 19 15 15 15 15 15 17 17 17 17 17 21 21 21 21 21 21 21 21 21 21 11 11 11 11 11
 [381] 29 29 29 29 29 14 14 14 14 14 23 23 23 23 23 19 19 19 19 19 21 21 21 21 21 26 26 26 26 26 23 23 23 23 23 17 17 17
 [419] 17 17 19 19 19 19 19 16 16 16 16 16 18 18 18 18 18 25 25 25 25 25 32 32 32 32 32 24 24 24 24 24 18 18 18 18 18 16
 [457] 16 16 16 16 22 22 22 22 22 19 19 19 19 19 24 24 24 24 24 30 30 30 30 30 15 15 15 15 15 24 24 24 24 24 21 21 21 21
 [495] 21 16 16 16 16 16 19 19 19 19 19 25 25 25 25 25 10 10 10 10 10 17 17 17 17 17 20 20 20 20 20 20 20 20 20 20 21 21
 [533] 21 21 21 16 16 16 16 16 13 13 13 13 13 19 19 19 19 19 20 20 20 20 20 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18
 [571] 16 16 16 16 16 22 22 22 22 22 12 12 12 12 12 13 13 13 13 13 12 12 12 12 12 15 15 15 15 15 22 22 22 22 22 11 11 11
 [609] 11 11 28 28 28 28 28 27 27 27 27 27 13 13 13 13 13 19 19 19 19 19 24 24 24 24 24 26 26 26 26 26 20 20 20 20 20 21
 [647] 21 21 21 21 17 17 17 17 17 22 22 22 22 22 29 29 29 29 29 32 32 32 32 32 20 20 20 20 20 21 21 21 21 21 22 22 22 22
 [685] 22 16 16 16 16 16 15 15 15 15 15 21 21 21 21 21 18 18 18 18 18 21 21 21 21 21 20 20 20 20 20 26 26 26 26 26 28 28
 [723] 28 28 28 27 27 27 27 27 18 18 18 18 18 22 22 22 22 22 25 25 25 25 25 25 25 25 25 25 22 22 22 22 22 31 31 31 31 31
 [761] 15 15 15 15 15 19 19 19 19 19 19 19 19 19 19 32 32 32 32 32 22 22 22 22 22 19 19 19 19 19 20 20 20 20 20 12 12 12
 [799] 12 12 19 19 19 19 19 30 30 30 30 30 16 16 16 16 16 27 27 27 27 27 19 19 19 19 19 22 22 22 22 22 19 19 19 19 19 18
 [837] 18 18 18 18 22 22 22 22 22 20 20 20 20 20 15 15 15 15 15 30 30 30 30 30 23 23 23 23 23 27 27 27 27 27 12 12 12 12
 [875] 12 20 20 20 20 20 23 23 23 23 23 16 16 16 16 16 17 17 17 17 17 23 23 23 23 23 18 18 18 18 18 20 20 20 20 20 28 28
 [913] 28 28 28 22 22 22 22 22 14 14 14 14 14 21 21 21 21 21 18 18 18 18 18 27 27 27 27 27 19 19 19 19 19 18 18 18 18 18
 [951] 20 20 20 20 20 26 26 26 26 26 13 13 13 13 13 29 29 29 29 29 25 25 25 25 25 30 30 30 30 30 21 21 21 21 21 28 28 28
 [989] 28 28 11 11 11 11 11 18 18 18 18 18
 [ reached getOption("max.print") -- omitted 4000 entries ]
```

3) checking duplicate:

The existence of redundant data alters how the data is portrayed, therefore, we must work on deleting any duplicate data in our dataset.

```
> nrow(dataset)
[1] 4257
> dataset[duplicated(dataset),]
     Age Experience Income Family CCAvg Education Personal.Loan Securities.Account CD.Account Online CreditCard
631   32          7     35      3  1.30         1             0                   0          0      0          1
800   29          3     39      4  2.10         3             0                   0          0      1          0
1027  28          4     43      3  0.10         2             0                   0          0      1          0
1202  35          8     38      4  1.00         2             0                   0          0      1          0
1527  36         10     80      4  2.20         2             0                   0          0      1          0
1677  46         20     74      4  2.60         3             0                   0          0      1          0
2032  60         35     80      3  0.50         1             0                   0          0      1          0
2629  33          6     78      4  2.00         2             0                   1          0      1          0
2682  37         11     35      2  0.80         3             0                   0          0      0          0
2729  39         13     58      3  2.10         1             0                   0          0      1          0
2807  53         27     59      2  0.80         3             0                   0          0      1          0
2956  54         29     44      2  2.30         3             0                   0          0      1          0
3051  50         25     58      1  1.30         2             0                   0          0      1          0
3362  31          5     85      3  1.60         1             0                   0          0      1          1
3454  29          3     31      4  0.30         2             0                   0          0      1          0
3578  39          9     32      3  2.00         3             0                   0          0      1          0
3695  38          8     21      1  0.67         3             0                   0          0      1          0
3903  45         21     39      2  2.10         3             0                   0          0      0          1
4059  39         15     65      1  1.50         3             0                   0          0      0          0
4483  40         14     28      2  0.80         3             0                   0          0      0          0
4617  66         41    114      1  0.80         3             0                   0          0      1          1
4733  39         13     69      3  0.10         1             0                   0          0      0          0
4744  50         26     21      1  0.20         1             0                   0          0      1          0
4745  44         20     72      3  0.30         3             0                   0          0      1          0
4747  31          7     18      1  0.40         3             0                   0          0      1          0
4829  52         28     62      1  1.80         3             0                   0          0      1          0
4837  54         24     72      3  1.40         3             0                   0          0      0          1
> dataset<- dataset [!duplicated(dataset),]
>
> nrow(dataset)  #duplicates are removed
[1] 4230
```

4) Data reduction

Through our understanding of data we noticed that (mortgage , ID , ZIP.Code) attributes don't affect our data set, hence we don’t need them in our study so we decided to drop them in order to prevent any conflict or data manipulation.

```
#mortgage, ID , ZIP CODE drop

dataset <- dataset[,-c(1,5,9)]
```

➔ Data Transformation:

1) Discretization :

By dividing the range of continuous attributes into intervals and then replacing them into small interval labels, this will enhance the understanding of data and improve its representation, making it simplified.

```
GAge=cut(dataset$Age, br=c(19, 29, 39,49,59 ,67))
GAge
table(GAge)
GAge=cut(dataset$Age, br=c(19, 29, 39,49,59 ,67), labels=c("20's", "30's","40's","50's","60's"))
table(GAge)
```

```
> table(GAge)
GAge
(19,29] (29,39] (39,49] (49,59] (59,67]
    488    1245    1254    1334     674
> GAge=cut(dataset$Age, br=c(19, 29, 39,49,59 ,67), labels=c("20's", "30's","40's","50's","60's"))
> table(GAge)
GAge
20's 30's 40's 50's 60's
 488 1245 1254 1334  674
>
```

2) Normalization :

When we do further analysis, the attribute income will intrinsically influence the result more due to its larger value,therefore, we need to normalize both the income and mortgage attributes. By doing so, the normalized data will be in a particular range that makes it simple to deal with and the large size will no longer affect the low values in the data.

```
#create function
normalize  <- function (x) {
  return ((x - min(x)) / (max (x)- min (x))) }

dataset$Income<-normalize(dataWithoutNormalization$Income) #assign


dataset$Mortgage<-normalize(dataWithoutNormalization$Mortgage)


str(dataset)
```

```
> str(dataset)
'data.frame':   4230 obs. of  11 variables:
 $ Age                : int  25 45 39 35 35 37 53 50 35 65 ...
 $ Experience         : num  10 10 10 10 10 23 23 23 23 23 ...
 $ Income             : num  0.2789 0.1769 0.0204 0.6259 0.2517 ...
 $ Family             : int  4 3 1 1 4 4 2 1 3 4 ...
 $ CCAvg              : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 2.4 ...
 $ Education          : int  1 1 1 2 2 2 2 3 2 3 ...
 $ Personal.Loan      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
 $ CD.Account         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Online             : int  0 0 0 0 0 1 1 0 1 0 ...
 $ CreditCard         : int  0 0 0 0 1 0 0 1 0 0 ...
```

Data before preprocessing :

```
> str(dataset)
'data.frame':   5000 obs. of  14 variables:
 $ ID                 : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Age                : int  25 45 39 35 35 37 53 50 35 34 ...
 $ Experience         : int  1 19 15 9 8 13 27 24 10 9 ...
 $ Income             : int  49 34 11 100 45 29 72 22 81 180 ...
 $ ZIP.Code           : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
 $ Family             : int  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          : int  1 1 1 2 2 2 2 3 2 3 ...
 $ Mortgage           : int  0 0 0 0 0 155 0 0 104 0 ...
 $ Personal.Loan      : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
 $ CD.Account         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Online             : int  0 0 0 0 0 1 1 0 1 0 ...
 $ CreditCard         : int  0 0 0 0 1 0 0 1 0 0 ...
> summary(dataset)
       ID              Age          Experience       Income          ZIP.Code         Family          CCAvg
 Min.   :   1   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   : 9307   Min.   :1.000   Min.   : 0.000
 1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:91911   1st Qu.:1.000   1st Qu.: 0.700
 Median :2500   Median :45.00   Median :20.0   Median : 64.00   Median :93437   Median :2.000   Median : 1.500
 Mean   :2500   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :93153   Mean   :2.396   Mean   : 1.938
 3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:94608   3rd Qu.:3.000   3rd Qu.: 2.500
 Max.   :5000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :96651   Max.   :4.000   Max.   :10.000
   Education        Mortgage     Personal.Loan   Securities.Account   CD.Account         Online          CreditCard
 Min.   :1.000   Min.   :  0.0   Min.   :0.000   Min.   :0.0000      Min.   :0.0000   Min.   :0.0000   Min.   :0.000
 1st Qu.:1.000   1st Qu.:  0.0   1st Qu.:0.000   1st Qu.:0.0000      1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
 Median :2.000   Median :  0.0   Median :0.000   Median :0.0000      Median :0.0000   Median :1.0000   Median :0.000
 Mean   :1.881   Mean   : 56.5   Mean   :0.096   Mean   :0.1044      Mean   :0.0604   Mean   :0.5968   Mean   :0.294
 3rd Qu.:3.000   3rd Qu.:101.0   3rd Qu.:0.000   3rd Qu.:0.0000      3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :3.000   Max.   :635.0   Max.   :1.000   Max.   :1.0000      Max.   :1.0000   Max.   :1.0000   Max.   :1.000
>
```

Data after preprocessing :

```
> str(dataset)
'data.frame':   4230 obs. of  11 variables:
 $ Age               : int  25 45 39 35 35 37 53 50 35 65 ...
 $ Experience        : num  10 10 10 10 10 23 23 23 23 23 ...
 $ Income            : num  0.2789 0.1769 0.0204 0.6259 0.2517 ...
 $ Family            : int  4 3 1 1 4 4 2 1 3 4 ...
 $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 2.4 ...
 $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
 $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
 $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
 $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
> summary(dataset)
      Age          Experience        Income           Family          CCAvg         Education     Personal.Loan
 Min.   :23.00   Min.   : 5.00   Min.   :0.0000   Min.   :1.000   Min.   :0.000   Min.   :1.000   0:4064
 1st Qu.:35.00   1st Qu.:17.00   1st Qu.:0.1769   1st Qu.:1.000   1st Qu.:0.600   1st Qu.:1.000   1: 166
 Median :46.00   Median :20.00   Median :0.3197   Median :2.000   Median :1.300   Median :2.000
 Mean   :45.56   Mean   :20.32   Mean   :0.3571   Mean   :2.458   Mean   :1.405   Mean   :1.923
 3rd Qu.:56.00   3rd Qu.:24.00   3rd Qu.:0.5034   3rd Qu.:4.000   3rd Qu.:2.100   3rd Qu.:3.000
 Max.   :67.00   Max.   :35.00   Max.   :1.0000   Max.   :4.000   Max.   :3.900   Max.   :3.000
 Securities.Account   CD.Account          Online          CreditCard
 Min.   :0.0000     Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000     1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.0000     Median :0.00000   Median :1.0000   Median :0.0000
 Mean   :0.1047     Mean   :0.04232   Mean   :0.5931   Mean   :0.2957
 3rd Qu.:0.0000     3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :1.0000     Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
>
```

5 Data Mining Technique

Classification:

Since the class label attribute is available with its name, it is convenient to use the classification method because it uses the supervised approach, in which we can determine whether the customer is interested in taking a personal loan or not through the model.

So we divide the dataset into training data and test data , we used (**party**) package so we can build the decision tree using the method **ctree,** and (**caret**) package to evaluate the model by using the **confusion matrix** method

Clustering :

In our dataset, we used k-means as our main clustering method. The k-means method relies on randomly selecting k objects as cluster centers and assigning the rest of the objects to the nearest center. The reason behind choosing the K-means method is because it can handle large datasets.

In addition, in our usage of k-means function, we have the (nstart) option which attempts multiple initial configurations and reports on the best one; thus, making our results more accurate. In our implementation of the k-means function, we started by scaling the non-numeric columns, however, we found out that it negatively affects our clustering and will display inaccurate results. Therefore, we dropped the non-numeric columns and started to apply clustering on these numeric data types only {Age , Experience , Income , CCAvg} . The next step was finding the optimal number of clusters for the dataset using the elbow and silhouette methods. Furthermore, we then applied the k-means method on 3 different numbers of clusters and visualized these clusters to see which one provides the best result. Finally, we found the silhouette width for each cluster and the avg of all clusters. As for the packages, we used (**factoextra) , (cluster), (GGally) , (plotly)** whereas the methods are **scale() , fviz_nbclust() , geom_vline() , labs() , kmeans() , fviz_cluster() , silhouette() , fviz_silhouette() , as.factor(), ggparcoord(), ggplotly().**

❖ Evaluation and Comparison

<table>
<tr><th>Mining task</th><th>Comparison Criteria</th></tr>
<tr><td>Classification</td><td>

Training set 70% , Testing set 30%

Evaluation metrics: Accuracy, precision, sensitivity, specificity.

	3 different Sizes of training set and testing set:
Accuracy	0.9816
precision	0.9826
sensitivity	0.9983
specificity	0.6379

Training set 90% , Testing set 10%

Evaluation metrics: Accuracy, precision, sensitivity, specificity.

	3 different Sizes of training set and testing set:
Accuracy	0.9887
precision	0.9883
sensitivity	1
specificity	0.7368

Training set 80% , Testing set 20%

</td></tr>
</table>

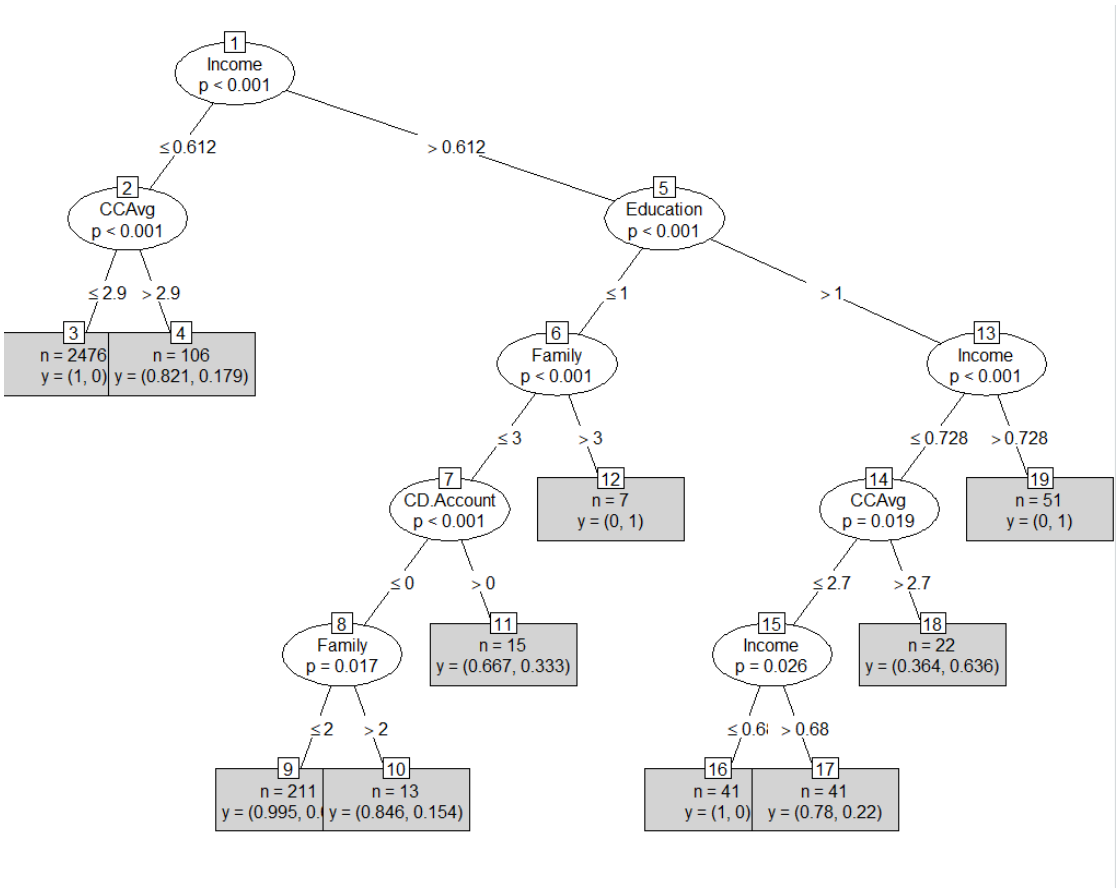
Evaluation metrics: Accuracy, precision, sensitivity, specificity.

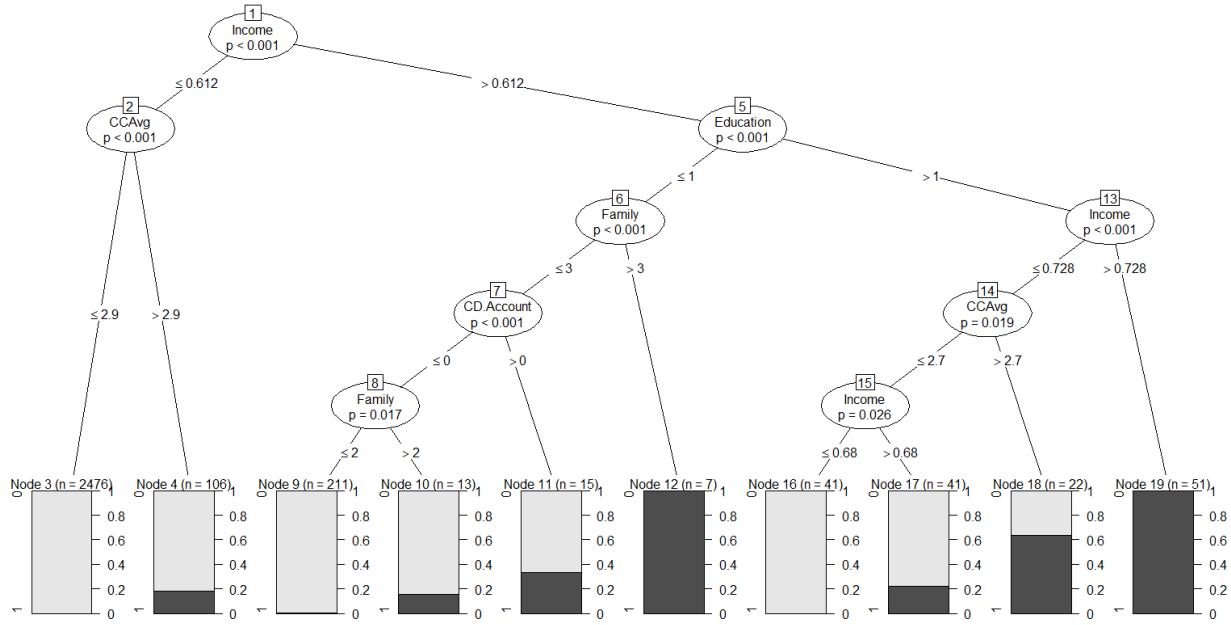
	3 different Sizes of training set and testing set:
Accuracy	0.9804
precision	0.9816
sensitivity	0.9981
specificity	0.6250

1- Training (70%) Test (30%)

```
Accuracy
98.15557
> results
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1187   21
         1    2   37

               Accuracy : 0.9816
                 95% CI : (0.9725, 0.9883)
    No Information Rate : 0.9535
    P-Value [Acc > NIR] : 8.789e-08

                  Kappa : 0.7537

 Mcnemar's Test P-Value : 0.0001746

            Sensitivity : 0.9983
            Specificity : 0.6379
         Pos Pred Value : 0.9826
         Neg Pred Value : 0.9487
             Prevalence : 0.9535
         Detection Rate : 0.9519
   Detection Prevalence : 0.9687
      Balanced Accuracy : 0.8181

       'Positive' Class : 0
```

Figures :

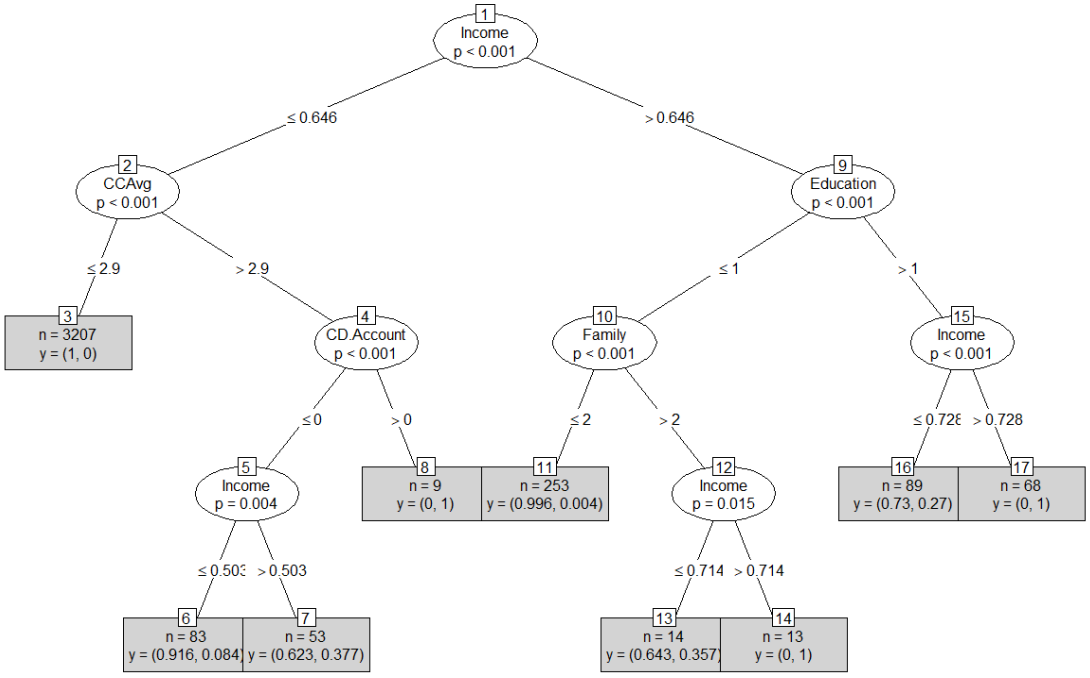

1
Income
p < 0.001
≤ 0.612
> 0.612
2
CCAvg
p < 0.001
5
Education
p < 0.001
≤ 2.9
> 2.9
≤ 1
> 1
3
n = 2476
y = (1, 0)
4
n = 106
y = (0.821, 0.179)
6
Family
p < 0.001
13
Income
p < 0.001
≤ 3
> 3
≤ 0.728
> 0.728
7
CD.Account
p < 0.001
12
n = 7
y = (0, 1)
14
CCAvg
p = 0.019
19
n = 51
y = (0, 1)
≤ 0
> 0
≤ 2.7
> 2.7
8
Family
p = 0.017
11
n = 15
y = (0.667, 0.333)
15
Income
p = 0.026
18
n = 22
y = (0.364, 0.636)
≤2
> 2
≤ 0.6
> 0.68
9
n = 211
y = (0.995, 0.
10
n = 13
y = (0.846, 0.154)
16
n = 41
y = (1, 0)
17
n = 41
y = (0.78, 0.22)

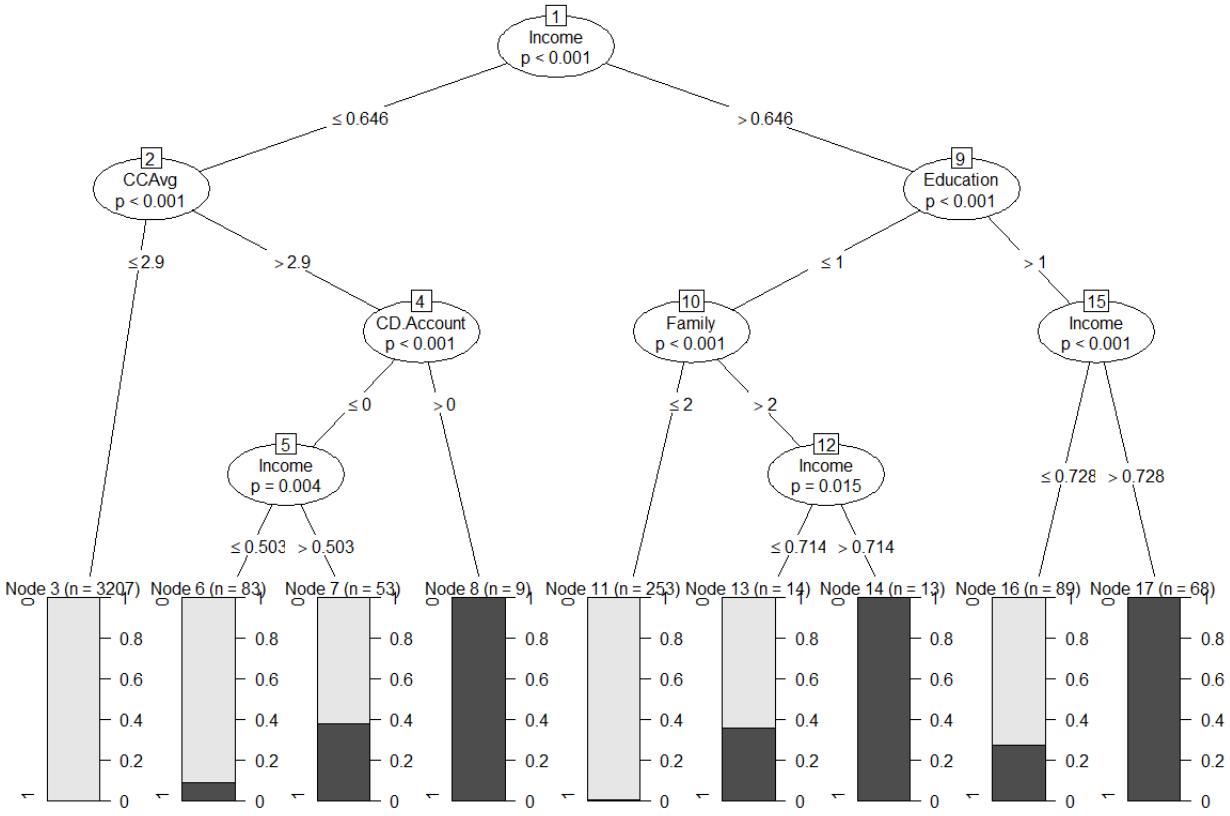

1
Income
p < 0.001
≤ 0.612
> 0.612
2
CCAvg
p < 0.001
5
Education
p < 0.001
≤ 2.9
> 2.9
≤ 1
> 1
6
Family
p < 0.001
13
Income
p < 0.001
≤ 3
> 3
≤ 0.728
> 0.728
7
CD.Account
p < 0.001
14
CCAvg
p = 0.019
≤ 0
> 0
≤ 2.7
> 2.7
8
Family
p = 0.017
15
Income
p = 0.026
≤2
> 2
≤ 0.68
> 0.68
Node 3 (n = 2476)
Node 4 (n = 106)
Node 9 (n = 211)
Node 10 (n = 13)
Node 11 (n = 15)
Node 12 (n = 7)
Node 16 (n = 41)
Node 17 (n = 41)
Node 18 (n = 22)
Node 19 (n = 51)
1
0.8
0.6
0.4
0.2
0

2- Training (90%) Test (10%)

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 422   5
         1   0  14

               Accuracy : 0.9887
                 95% CI : (0.9737, 0.9963)
    No Information Rate : 0.9569
    P-Value [Acc > NIR] : 0.0001215

                  Kappa : 0.8427

 Mcnemar's Test P-Value : 0.0736383

            Sensitivity : 1.0000
            Specificity : 0.7368
         Pos Pred Value : 0.9883
         Neg Pred Value : 1.0000
             Prevalence : 0.9569
         Detection Rate : 0.9569
   Detection Prevalence : 0.9683
      Balanced Accuracy : 0.8684

       'Positive' Class : 0
```

Figures:

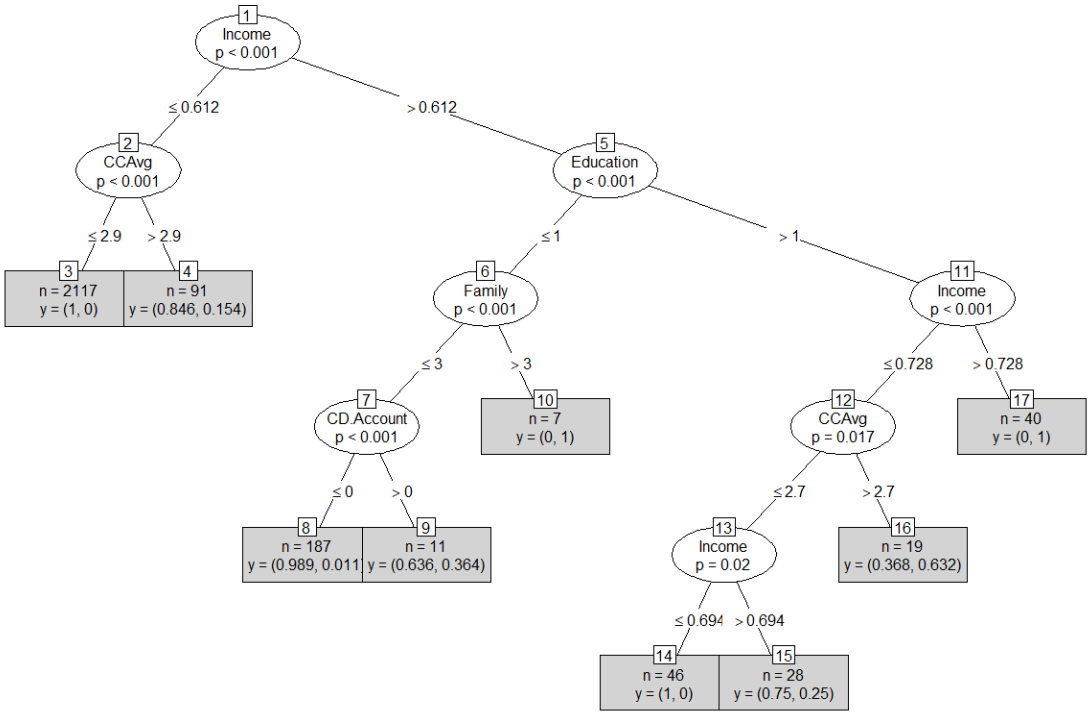

1
Income
p < 0.001
≤ 0.646
> 0.646
2
CCAvg
p < 0.001
9
Education
p < 0.001
≤ 2.9
> 2.9
≤ 1
> 1
3
n = 3207
y = (1, 0)
4
CD.Account
p < 0.001
10
Family
p < 0.001
15
Income
p < 0.001
≤ 0
> 0
≤ 2
> 2
≤ 0.728
> 0.728
5
Income
p = 0.004
8
n = 9
y = (0, 1)
11
n = 253
y = (0.996, 0.004)
12
Income
p = 0.015
16
n = 89
y = (0.73, 0.27)
17
n = 68
y = (0, 1)
≤ 0.503
> 0.503
≤ 0.714
> 0.714
6
n = 83
y = (0.916, 0.084)
7
n = 53
y = (0.623, 0.377)
13
n = 14
y = (0.643, 0.357)
14
n = 13
y = (0, 1)

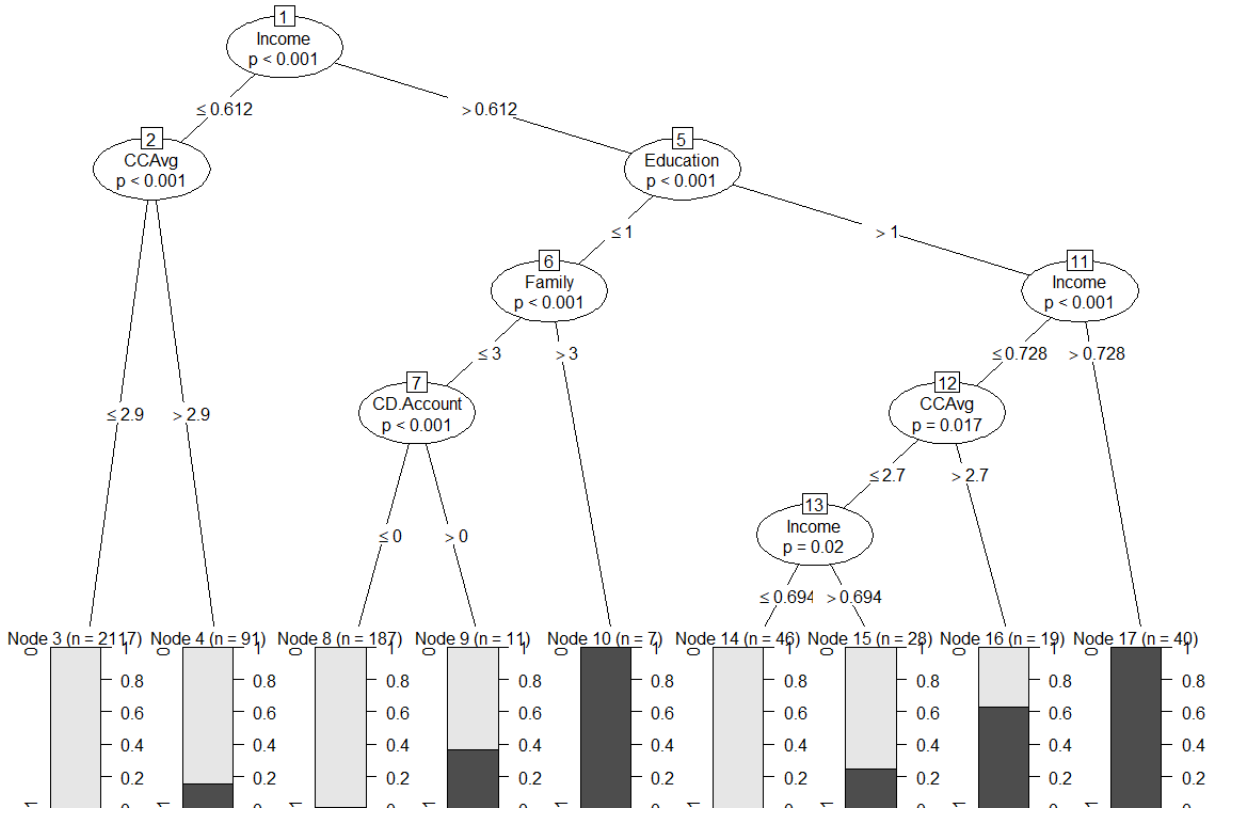

1
Income
p < 0.001
≤ 0.646
> 0.646
2
CCAvg
p < 0.001
9
Education
p < 0.001
≤ 2.9
> 2.9
≤ 1
> 1
4
CD.Account
p < 0.001
10
Family
p < 0.001
15
Income
p < 0.001
≤ 0
> 0
≤ 2
> 2
≤ 0.728
> 0.728
5
Income
p = 0.004
12
Income
p = 0.015
≤ 0.503
> 0.503
≤ 0.714
> 0.714
Node 3 (n = 3207)
Node 6 (n = 83)
Node 7 (n = 53)
Node 8 (n = 9)
Node 11 (n = 253)
Node 13 (n = 14)
Node 14 (n = 13)
Node 16 (n = 89)
Node 17 (n = 68)
1
0.8
0.6
0.4
0.2
0

3- Training (60%) Test (40%)

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1601   30
         1    3   50

               Accuracy : 0.9804
                 95% CI : (0.9726, 0.9865)
    No Information Rate : 0.9525
    P-Value [Acc > NIR] : 1.108e-09

                  Kappa : 0.7421

 Mcnemar's Test P-Value : 6.011e-06

            Sensitivity : 0.9981
            Specificity : 0.6250
         Pos Pred Value : 0.9816
         Neg Pred Value : 0.9434
             Prevalence : 0.9525
         Detection Rate : 0.9507
   Detection Prevalence : 0.9685
      Balanced Accuracy : 0.8116

       'Positive' Class : 0
```

Figures:

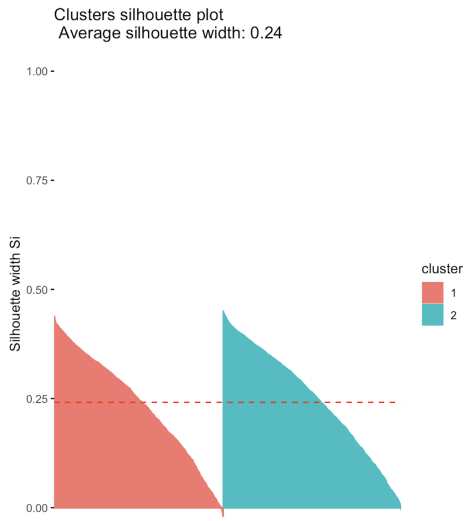

1
Income
p < 0.001
≤ 0.612
> 0.612
2
CCAvg
p < 0.001
5
Education
p < 0.001
≤ 2.9
> 2.9
≤ 1
> 1
3
n = 2117
y = (1, 0)
4
n = 91
y = (0.846, 0.154)
6
Family
p < 0.001
11
Income
p < 0.001
≤ 3
> 3
≤ 0.728
> 0.728
7
CD.Account
p < 0.001
10
n = 7
y = (0, 1)
12
CCAvg
p = 0.017
17
n = 40
y = (0, 1)
≤ 0
> 0
≤ 2.7
> 2.7
8
n = 187
y = (0.989, 0.011)
9
n = 11
y = (0.636, 0.364)
13
Income
p = 0.02
16
n = 19
y = (0.368, 0.632)
≤ 0.694
> 0.694
14
n = 46
y = (1, 0)
15
n = 28
y = (0.75, 0.25)

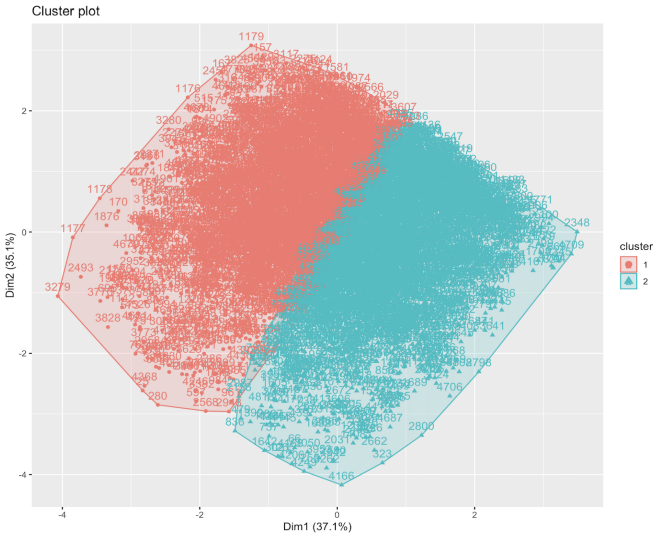

1
Income
p < 0.001
≤ 0.612
> 0.612
2
CCAvg
p < 0.001
5
Education
p < 0.001
≤ 2.9
> 2.9
≤ 1
> 1
6
Family
p < 0.001
11
Income
p < 0.001
≤ 3
> 3
≤ 0.728
> 0.728
7
CD.Account
p < 0.001
12
CCAvg
p = 0.017
≤ 0
> 0
≤ 2.7
> 2.7
13
Income
p = 0.02
≤ 0.694
> 0.694
Node 3 (n = 2117)
Node 4 (n = 91)
Node 8 (n = 187)
Node 9 (n = 11)
Node 10 (n = 7)
Node 14 (n = 46)
Node 15 (n = 28)
Node 16 (n = 19)
Node 17 (n = 40)
0
0.2
0.4
0.6
0.8
1

Mining Task	Comparison Criteria
Clustering	**2 clusters**

# of cluster	**cluster #1**	**cluster#2**
Silhouette width for each cluster	0.23	0.25
Silhouette width for all clusters	**0.24**	

VISUALIZATION



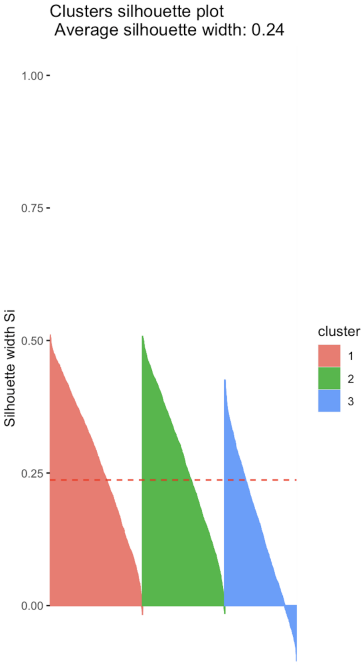


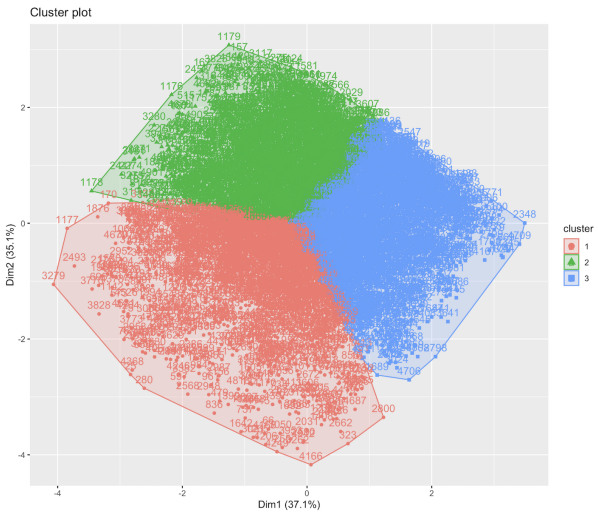
Clustering

3 clusters

# of cluster	cluster #1	cluster#2	cluster#3
Silhouette width for each cluster	0.15	0.27	0.28
Silhouette width for all clusters	**0.24**		

VISUALIZATION



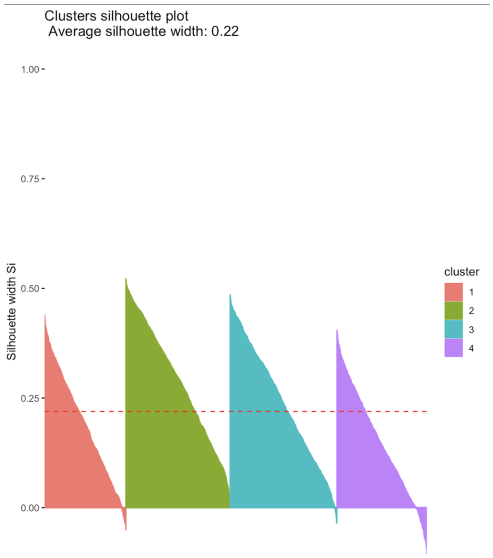


Clustering

4 clusters

# of cluster	cluster #1	cluster#2	cluster#3	cluster#4
Silhouette width for each cluster	0.24	0.15	0.19	0.29
Silhouette width for all clusters	**0.22**			

VISUALIZATION



6 Findings:

Putting everything into consideration, we began our study by identifying the dataset's attributes and knowing how each one impacts every other attribute by drawing plots that show how well correlated(affect each other) the attributes are. In addition, we looked to see if the dataset contains any outliers/missing data in order to apply data cleaning and transformation techniques to our dataset properly which will help us identify the potential customers willing to purchase the loan. Finally , to solve the data mining problem, we needed to implement data mining techniques , including both classification and clustering .

*In our **classification**, we studied the dataset, and understood the attributes and how it will affect each other to help with predicting the best result, so we came up with these results after applying data mining technique:*

-Training set 70% and Testing set 30% accuracy = 98.15%

-Training set 90% and Testing set 10% accuracy = 98.87%

-Training set 60% and Testing set 40% accuracy = 98.04%

The best evaluation model which has best accuracy was the second model which is [90% , 10%]

the number customers who accept a personal loan was affected by (Income,Education, Family , CCAvg= low monthly spending average and CD Account = certificate of deposit)

so it learned better than other evaluation models .

So [90% , 10%] model is the best since it learned better than the other models

Some interesting relationships we noticed :

- *high income*
- *high education level*

_____

∴ *take personal loan*

- *high income ∧ low education level*
- *many family members*

_____

∴ *take personal loan*

- *low income*
- *high monthly spending average ∧ has CD account*

_____

∴ *take personal loan*

- *low income*
- *low monthly spending average*

_____

∴ *don't take personal loan*

- *low income ∧ low education level*
- *low family members*

_____

∴ *don't take personal loan*

To conclude, our suggestion is to focus on promoting and offering the personal loan to the interested groups since the chance of them purchasing the personal loan is quite higher than other groups, also the bank can offer a certificate of deposit (CD) to the customers with high monthly spending since it will raise the chance of them buying a personal loan significantly as :

- *low income ∧ no CD account*
- *high monthly spending*

_____

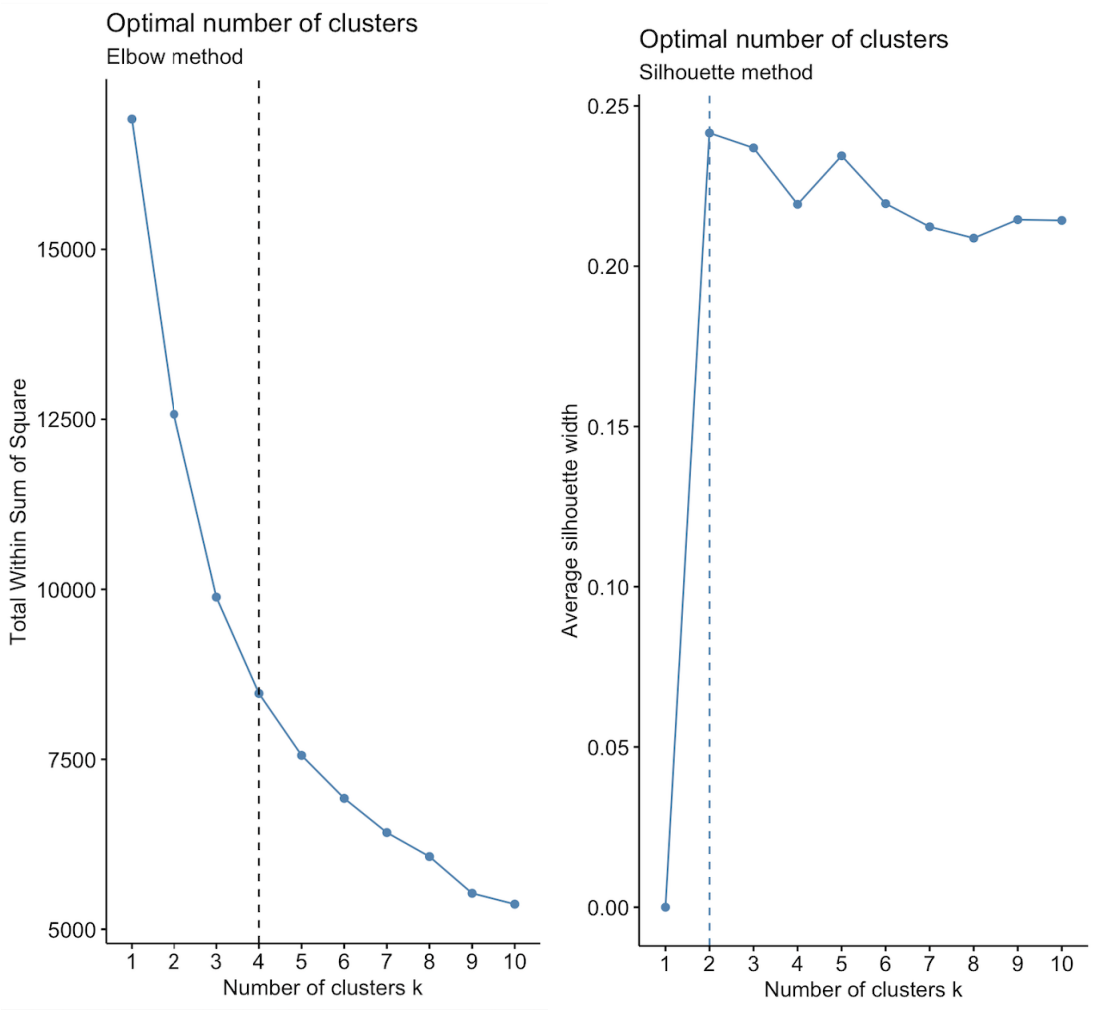
∴ *don't take personal loan*

- *low income ∧ has CD account*
- *high monthly spending*

_____

∴ *take personal loan*

Whereas in **clustering**, which is simply partitioning data into groups where each group has data that is similar to one another and dissimilar from other groups. To cluster the data, we applied the k-means method to our dataset. By doing so , we chose 3 different numbers of clusters to test them out and find out which one offers the best results.



To evaluate the different number of clusters, we used elbow and silhouette methods to help us determine the optimal number of clusters. By doing so, we inferred from the elbow method that the optimal number of clusters is 4 . On the other hand , the silhouette method indicated that 2 clusters were the optimal, which were inaccurate results, since we did further analysis by plotting the 3 different clusters and seeing which visualization shows the best clustering results. After looking at them, 4 clusters represents the clearest grouping between the clusters with no apparent overlapping; signifying that the characteristics of each cluster is distinct from others, providing a better clustering result. As a result, 4 clusters proved to be the optimal number of clusters.

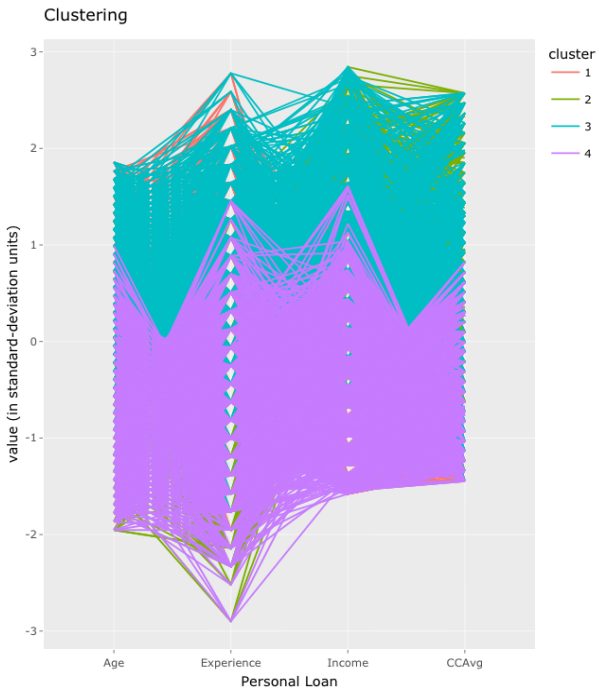
when k=2
cluster size :
cluster #1 : 2064
cluster #2 : 2166

when k=3
cluster size :
cluster #1 : 1587
cluster #2 : 1412
cluster #3 : 1231

when k=4
cluster size :
cluster #1 : 900
cluster #2 : 1154
cluster #3 : 1183
cluster #4 : 993

Additionally, we implemented a code for clustering analysis, precisely to identify the patterns found within the 4 clusters, the results in the plot below show that observations which belong to the same group tend to have similar characteristics.

For example in the blue cluster we noticed that it has high age, experience, income, and CCAvg values ,while in contrast the red cluster got categorized by the low age, experience, income, and CCAvg values ,also we can see that the compactness in these clusters is significant due to the huge numbers of objects in our dataset.



Although we displayed a visualization to interpret the clustering, one drawback of the plot is that it gets cluttered as the number of objects increase, and since our dataset has more than 4000 objects, this makes it difficult to analyse the clusters using a plot.

Hence, we will evaluate our clusters by looking at the centers of each cluster. This is beneficial since the center of a cluster is considered a representative of the observations of that cluster.

	Age	**Experience**	**Income**	**CCAvg**
#1	0.7874592	0.5589504	-0.7000224	-0.6936062
#2	-0.8029563	-0.6513380	0.8240874	0.8710967
#3	0.7678951	0.6873401	0.7088348	0.6982482
#4	-0.8018488	-0.6266723	-0.5390751	-0.5722113

By seeing the values of each attribute in each cluster, one can imply that there is a variation in the values from one cluster to the other , therefore, the centers are not overlapping and we applied the goal of clustering which is to cluster the data with minimal overlap such that the intra class similarity is maximized and the inter class similarity is minimized.

Code

Preprocessing:

```
dataset = read.csv('Bank_Personal_Loan_Modelling.csv')
#dataset2 = dataset


###### data before processing########3
str(dataset)
summary(dataset)
View(dataset)

############# Removing  Outliers ###############

#age no outliers

boxplot(dataset$Age)$out # = 0

#Experience no outliers

boxplot(dataset$Experience)$out # = 0

#mortgage drop

dataset <- dataset[,-c(1,5,9)]

#income

boxplot(dataset$Income)$out #? yes

dataset=subset(dataset, Income<=155)

boxplot(dataset$Income)

boxplot(dataset$Income)$out #=0

#ccavg

boxplot(dataset$CCAvg)$out #? yes

boxplot(dataset$CCAvg)

dataset=subset(dataset, CCAvg<=3.9)

boxplot(dataset$CCAvg)$out #=0
```

```


######boxplot After removing outliers

boxplot(dataset$Income)
boxplot(dataset$CCAvg)


###########duplicate###################

nrow(dataset)
dataset[duplicated(dataset),]
dataset<- dataset [!duplicated(dataset),]

nrow(dataset)  #duplicates are removed


#########discretization################

summary(dataset$Age)

datasett = dataset

GAge=cut(datasett$Age, br=c(22, 29, 39,49,59 ,67), labels=c("20's","30's","40's","50's","60's"))
table(GAge)

PL=cut(dataset$Personal.Loan, br=c(-0.01,0.5,1), labels=c("0","1"))
table(PL)

dataset$Personal.Loan=PL
datasett$Age=GAge


############## bining by mean ###################

no_of_bins <- 5
dataset$Experience=round(ave(dataset$Experience, rep(1:length(dataset$Experience), each = no_of_bins, length.out = length(dataset$Experience))))




################ Normalization ##################

```

Classification:

```r

summary(dataset)
# I decided what to normalize by observing the ranges

dataWithoutNormalization <- dataset
#create function
normalize  <- function (x) {
  return ((x - min(x)) / (max (x)- min (x))) }

dataset$Income<-normalize(dataset$Income) #assign


########## data after processing #############
str(dataset)
summary(dataset)


############ phase 2


dataset2 = dataset


#split data
set.seed(1234)
#Training (70%)  Test (30%).
ind <- sample(2, nrow(dataset2), replace=TRUE, prob=c(0.7, 0.3))
trainData <- dataset2[ind==1,]
testData <- dataset2[ind==2,]


library(party)
#myFormula specifies that Species is the target variable and all other variables are independent variables.
myFormula <- Personal.Loan ~ Age + Experience + Income + Family + CCAvg + Education + Securities.Account + CD.Account + Online + CreditCard
#Build decision tree ctree()
data_ctree <- ctree(myFormula, data=trainData)

#check the prediction
#makes prediction for data.
table(predict(data_ctree), trainData$Personal.Loan)
print(data_ctree)
plot(data_ctree,type="simple")
plot(data_ctree)

```

```r
# predict on test data
#the built tree is tested with test data
testPred <- predict(data_ctree, newdata = testData)

#Evaluate the model
#Create the confusion matrix
table(testPred, testData$Personal.Loan)



#accuracy and cm

library(caret)
results <- confusionMatrix(testPred, testData$Personal.Loan)
acc <- results$overall["Accuracy"]*100
acc
results
#_________________


set.seed(1234)
#Training (90%)  Test (10%).
ind <- sample(2, nrow(dataset2), replace=TRUE, prob=c(0.9, 0.1))
trainData <- dataset2[ind==1,]
testData <- dataset2[ind==2,]



#myFormula specifies that Species is the target variable and all other variables are independent variables.
myFormula <- Personal.Loan ~ Age + Experience + Income + Family + CCAvg + Education + Securities.Account + CD.Account + Online + CreditCard
#Build decision tree ctree()
data_ctree <- ctree(myFormula, data=trainData)

#check the prediction
#makes prediction for data.
table(predict(data_ctree), trainData$Personal.Loan)
print(data_ctree)
plot(data_ctree,type="simple")
plot(data_ctree)


# predict on test data
#the built tree is tested with test data
testPred <- predict(data_ctree, newdata = testData)
```

```r
testPred <- predict(data_ctree, newdata = testData)

#Evaluate the model
#Create the confusion matrix
table(testPred, testData$Personal.Loan)




#accuracy and cm

results <- confusionMatrix(testPred, testData$Personal.Loan)
acc <- results$overall["Accuracy"]*100
acc
results

#_________________


set.seed(1234)
#Training (60%)  Test (40%).
ind <- sample(2, nrow(dataset2), replace=TRUE, prob=c(0.60, 0.40))
trainData <- dataset2[ind==1,]
testData <- dataset2[ind==2,]



#myFormula specifies that Species is the target variable and all other variables are independent variables.
myFormula <- Personal.Loan ~ Age + Experience + Income + Family + CCAvg + Education + Securities.Account + CD.Account + Online + CreditCard
#Build decision tree ctree()
data_ctree <- ctree(myFormula, data=trainData)

#check the prediction
#makes prediction for data.
table(predict(data_ctree), trainData$Personal.Loan)
print(data_ctree)
plot(data_ctree,type="simple")
plot(data_ctree)


```

Clustering:

```r
# predict on test data
#the built tree is tested with test data
testPred <- predict(data_ctree, newdata = testData)

#Evaluate the model
#Create the confusion matrix
table(testPred, testData$Personal.Loan)

#accuracy and cm

results <- confusionMatrix(testPred, testData$Personal.Loan)
acc <- results$overall["Accuracy"]*100
acc
results


# **** CLUSTERING ****

str(dataset)

#dropping unnecessary columns ( non-numeric columns )
dataset3 <- dataset[,-c(4,6,7,8,9,10,11)]

str(dataset3)

#scaling data
dataset3 <- scale(dataset3)


#checking appropriate number of clusters
library(factoextra)

fviz_nbclust(dataset3,  kmeans , method = "wss") +
  geom_vline(xintercept = 4, linetype = 2) +
  labs(subtitle = "Elbow method")


fviz_nbclust(dataset3, kmeans , method = "silhouette") +
  labs(subtitle="Silhouette method")



```

```r
#visualizing data with two clusters
km2 = kmeans(dataset3, centers = 2, nstart = 25)
fviz_cluster(km2, dataset3)

#interpreting cluster
km2$centers

#average for each cluster
library(cluster)
avg_sil <- silhouette( km2$cluster , dist(dataset3))
fviz_silhouette(avg_sil)



#visualizing data with 3 clusters
km3 = kmeans(dataset3, centers = 3, nstart = 25)
fviz_cluster(km3,dataset3)

#interpreting cluster
km3$centers

#average for each cluster
avg_sil <- silhouette( km3$cluster , dist(dataset3))
fviz_silhouette(avg_sil)



#visualizing data with 4 clusters
km4 <- kmeans(dataset3, centers = 4, nstart = 25)
fviz_cluster(km4, dataset3)

#interpreting cluster
km4$centers

#average for each cluster
avg_sil <- silhouette( km4$cluster , dist(dataset3))
fviz_silhouette(avg_sil)


```

```r
#clustering patterns interpretation
library(GGally)
library(plotly)

dataset3$cluster <- as.factor(km4$cluster)
p <- ggparcoord(data = dataset3, columns = c(1:4), groupColumn = "cluster", scale = "std") + labs(x = "Personal Loan", y = "value (in standard-deviation units)", title = "Clustering")
ggplotly(p)


```

7 References

Use IEEE format
https://libraryguides.vu.edu.au/ieeereferencing/gettingstarted

You can use any resource for reference generating such as

https://www.citethisforme.com/citation-generator/ieee

8 Tasks Distribution

ID	Name	Responsibilities
	Yara AlManea	*Tasks were distributed evenly*
	Razan AlDhafian	*Tasks were distributed evenly*
	Deema AlFuaim	*Tasks were distributed evenly*