Customer Segmentation

Project Report

Submitted to:

Prof. Dr. Gopikrishnan S

Presented by:

L BHARADHWAJ REDDY	19BCD7047
REDDYBATHINA NAGA SAI RAM	19BCD7052
B N V R S ROHITH	19BCD7033
D V L SAI SRUTHI	19BCD7040

INDEX

1.INTRODUCTION

Sometimes referred to as market segmentation, customer segmentation is a method of analysing a client base and grouping customers into categories or segments which share particular attributes. Key differentials in segmentation include age, gender, education, location, spending patterns and socio-economic group. Relevant differentials are those which are expected to influence customer behaviour in relation to a business. The selected criteria are used to create customer segments with similar values, needs and wants.

When planning a targeted marketing campaign, it is also necessary to differentiate customers within these groupings according to their preferred means of communication.

2.OBJECTIVE

Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group. The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business.

3. LITERATURE REVIEW

K – Mean algorithm in one of the most popular centroid based algorithm. Suppose data set, D, contains n object in space. Partitioning methods distribute the object in D into K clusters. A centroid-based partitioning technique uses the centroid of a cluster, Ci to represent that cluster. Conceptually the centroid of a cluster is its center point. The different between an object p € Ci and Ci the representative of the cluster is measured by dist(p, Ci) where dist(x, y) is the Euclidean distance between two points x and y.

4. METHODOLOGY

Customer Segmentation is one the most important applications of unsupervised learning. Using clustering techniques, companies can identify the several segments of customers allowing them to target the potential user base. In this machine learning project, we will make use of K-means clustering. First we cleaned the dataset and then analysed some of the attributes present in dataset like Gender , age and Spending score. After that we fitted the model. While using the k-means clustering algorithm, the first step is to indicate the number of clusters (k) that we wish to produce in the final output.While working with clusters, you need to specify the number of clusters to use. We would like to utilize the optimal number of clusters. To help us in determining the optimal clusters, we used these three popular methods –

- **Elbow method**
- **Silhouette method**
- **Gap statistic**

After that we selected the optimal clusters and finally visualized the clusters.

5.RESULTS OBTAINED

5.1 Code for Exploratory Analysis

```
#Loading the test segmentation csv file into a data frame for customer segmentation
customer_segmentation=read.csv("test_segmentation.csv")
customer_segmentation

#Performing Exploratory analysis
#Class of data object
class(customer_segmentation)

#This function is used to Display Internal structure of data
str(customer_segmentation)

#This function is used to give Summary of data
summary(customer_segmentation)

#This function is used to give Column names
names(customer_segmentation)

#This function is used to give Dimensions of data
dim(customer_segmentation)

#This function is used to give the Data of the top
head(customer_segmentation)

#This function is used to give Data from the top
tail(customer_segmentation)
```

5.2 Output for Exploratory Analysis

```
> #Loading the test segmentation csv file into a data frame for customer segmentation
> customer_segmentation=read.csv("test_segmentation.csv")
> customer_segmentation
    i..ID Gender Ever_Married Age Graduated    Profession Work_Experience Spending_Score Family_Size Var_1 Segmentation
1  458989 Female          Yes  36       Yes      Engineer               0            Low           1 Cat_6            B
2  458994   Male          Yes  37       Yes    Healthcare               8        Average           4 Cat_6            A
3  458996 Female          Yes  69        No          <NA>               0            Low           1 Cat_6            A
4  459000   Male          Yes  59        No     Executive              11           High           2 Cat_6            B
5  459001 Female           No  19        No     Marketing              NA            Low           4 Cat_6            A
6  459003   Male          Yes  47       Yes        Doctor               0           High           5 Cat_4            C
7  459005   Male          Yes  61       Yes        Doctor               5            Low           3 Cat_6            D
8  459008 Female          Yes  47       Yes        Artist               1        Average           3 Cat_6            D
9  459013   Male          Yes  50       Yes        Artist               2        Average           4 Cat_6            B
10 459014   Male           No  19        No    Healthcare               0            Low           4 Cat_6            B
11 459015   Male           No  22        No    Healthcare               0            Low           3 Cat_6            D
12 459016 Female           No  22        No    Healthcare               0            Low           6 Cat_6            D
13 459024   Male          Yes  50       Yes        Artist               1        Average           5 Cat_6            A
14 459026   Male           No  27        No    Healthcare               8            Low           3 Cat_3            D
15 459032   Male           No  18        No        Doctor               0            Low           3 Cat_6            D
16 459033 Female          Yes  61       Yes        Artist               0            Low           1 Cat_6            C
17 459036 Female          Yes  20       Yes        Lawyer               1        Average           3 Cat_3            D
18 459039   Male          Yes  45       Yes        Artist               1        Average           2 Cat_6            B
19 459041   Male          Yes  55       Yes        Artist               8            Low           1 Cat_6            B
20 459045 Female          Yes  88       Yes        Lawyer               1        Average           4 Cat_6            C
21 459056   Male          Yes  63        No     Executive              NA           High           3 Cat_6            A
22 459057   Male          Yes  69        No        Lawyer              NA           High          NA Cat_6            D
23 459058   Male           No  42       Yes        Artist               0            Low           4 Cat_3            A
24 459059   Male          Yes  79        No     Executive              NA           High           2 Cat_6            B
25 459061 Female          Yes  35       Yes    Healthcare               9           High           3 Cat_6            B
26 459064   Male          Yes  27        No     Executive               5           High           4 Cat_6            B
```

```
> #Performing Exploratory analysis
> #Class of data object
> class(customer_segmentation)
[1] "data.frame"
> #This function is used to Display Internal structure of data
> str(customer_segmentation)
'data.frame':   2627 obs. of  11 variables:
 $ i..ID          : int  458989 458994 458996 459000 459001 459003 459005 459008 459013 459014 ...
 $ Gender         : chr  "Female" "Male" "Female" "Male" ...
 $ Ever_Married   : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Age            : int  36 37 69 59 19 47 61 47 50 19 ...
 $ Graduated      : chr  "Yes" "Yes" "No" "No" ...
 $ Profession     : chr  "Engineer" "Healthcare" NA "Executive" ...
 $ Work_Experience: int  0 8 0 11 NA 0 5 1 2 0 ...
 $ Spending_Score : chr  "Low" "Average" "Low" "High" ...
 $ Family_Size    : int  1 4 1 2 4 5 3 3 4 4 ...
 $ Var_1          : chr  "Cat_6" "Cat_6" "Cat_6" "Cat_6" ...
 $ Segmentation   : chr  "B" "A" "A" "B" ...
```

```
> #This function is used to give Summary of data
> summary(customer_segmentation)
     i..ID            Gender           Ever_Married            Age         Graduated          Profession         Work_Experience
 Min.   :458989   Length:2627        Length:2627        Min.   :18.00   Length:2627        Length:2627        Min.   : 0.000
 1st Qu.:461163   Class :character   Class :character   1st Qu.:30.00   Class :character   Class :character   1st Qu.: 0.000
 Median :463379   Mode  :character   Mode  :character   Median :41.00   Mode  :character   Mode  :character   Median : 1.000
 Mean   :463434                                         Mean   :43.65                                         Mean   : 2.553
 3rd Qu.:465696                                         3rd Qu.:53.00                                         3rd Qu.: 4.000
 Max.   :467968                                         Max.   :89.00                                         Max.   :14.000
                                                                                                              NA's   :269
 Spending_Score      Family_Size       Var_1            Segmentation
 Length:2627        Min.   :1.000   Length:2627        Length:2627
 Class :character   1st Qu.:2.000   Class :character   Class :character
 Mode  :character   Median :2.000   Mode  :character   Mode  :character
                    Mean   :2.825
                    3rd Qu.:4.000
                    Max.   :9.000
                    NA's   :113
> #This function is used to give Column names
> names(customer_segmentation)
 [1] "i..ID"           "Gender"          "Ever_Married"    "Age"             "Graduated"       "Profession"      "Work_Experience"
 [8] "Spending_Score"  "Family_Size"     "Var_1"           "Segmentation"
> #This function is used to give Dimensions of data
> dim(customer_segmentation)
[1] 2627   11
```

```
> #This function is used to give the Data of the top
> head(customer_segmentation)
   i..ID Gender Ever_Married Age Graduated Profession Work_Experience Spending_Score Family_Size Var_1 Segmentation
1 458989 Female          Yes  36       Yes   Engineer               0            Low           1 Cat_6            B
2 458994   Male          Yes  37       Yes Healthcare               8        Average           4 Cat_6            A
3 458996 Female          Yes  69        No       <NA>               0            Low           1 Cat_6            A
4 459000   Male          Yes  59        No  Executive              11           High           2 Cat_6            B
5 459001 Female           No  19        No  Marketing              NA            Low           4 Cat_6            A
6 459003   Male          Yes  47       Yes     Doctor               0           High           5 Cat_4            C
> #This function is used to give Data from the top
> tail(customer_segmentation)
      i..ID Gender Ever_Married Age Graduated    Profession Work_Experience Spending_Score Family_Size Var_1 Segmentation
2622 467950 Female           No  35       Yes Entertainment               1            Low           2 Cat_6            D
2623 467954   Male           No  29        No    Healthcare               9            Low           4 Cat_6            B
2624 467958 Female           No  35       Yes        Doctor               1            Low           1 Cat_6            A
2625 467960 Female           No  53       Yes Entertainment              NA            Low           2 Cat_6            C
2626 467961   Male          Yes  47       Yes     Executive               1           High           5 Cat_4            C
2627 467968 Female           No  43       Yes    Healthcare               9            Low           3 Cat_7            A
```

5.3 Code used for cleaning of the dataset

```
#Checking if there are any NA values in the data set
any(is.na(customer_segmentation))
# So from the output it is understood that there are NA values in the data set
#Let us extract the count of NA values in the data set
sum(is.na(customer_segmentation))

#Lets check the NA values column wise because there columns of both Numeric and Character

any(is.na(customer_segmentation$ID))
any(is.na(customer_segmentation$Gender))
any(is.na(customer_segmentation$Ever_Married))
any(is.na(customer_segmentation$Age))
any(is.na(customer_segmentation$Graduated))
any(is.na(customer_segmentation$Profession))
any(is.na(customer_segmentation$Work_Experience))
any(is.na(customer_segmentation$Spending_Score))
any(is.na(customer_segmentation$Family_Size))
any(is.na(customer_segmentation$Var_1))
any(is.na(customer_segmentation$Segmentation))

#So from the output it is evident that Ever_Married , Graduated , Profession and Work experience
#Family_size , Var_1 have NA values

class(customer_segmentation$Ever_Married)
#So Ever_Married is column which contains character values
#So we cannot replace with mean value

customer_segmentation$Ever_Married[is.na(customer_segmentation$Ever_Married)]="No"
```

```
class(customer_segmentation$Graduated)
#So Graduated is column which contains character values

customer_segmentation$Graduated[is.na(customer_segmentation$Graduated)]="Yes"

class(customer_segmentation$Profession)
#So Profession is column which contains character values

customer_segmentation$Profession[is.na(customer_segmentation$Profession)]="Engineer"

class(customer_segmentation$Work_Experience)
#So Experience is column which contains integer values

x=mean(customer_segmentation$Work_Experience,na.rm = TRUE)
x

# From the output we can see that we got a numeric value(i.e we got a decimal value)
#But work experience cannot be such a value. So lets either use floor or ceiling
y=floor(x)
y

customer_segmentation$Work_Experience[is.na(customer_segmentation$Work_Experience)]=y

class(customer_segmentation$Family_Size)
#So Family_Size is column which contains integer values

z=mean(customer_segmentation$Family_Size,na.rm = TRUE)
z
```

```
# From the output we can see that we got a numeric value(i.e we got a decimal value)
w=ceiling(z)
w

customer_segmentation$Family_Size[is.na(customer_segmentation$Family_Size)]=w

class(customer_segmentation$Var_1)
#So Var_1 is column which contains character values

customer_segmentation$Var_1[is.na(customer_segmentation$Var_1)]="Cat_2"

customer_segmentation

#Checking if there are any NA values now
any(is.na(customer_segmentation))
#From the result it is understood that there are NA values in the data set
#So the data is now cleaned

#Writing the updated ones into new csv file named credit_cards_details
write.csv(customer_segmentation,"customer_segmentation_cleaned.csv")
```

5.4 Outputs for cleaning of the data set

```
> #Checking if there are any NA values in the data set
> any(is.na(customer_segmentation))
[1] TRUE
> # So from the output it is understood that there are NA values in the data set
> #Let us extract the count of NA values in the data set
> sum(is.na(customer_segmentation))
[1] 526
> any(is.na(customer_segmentation$ID))
[1] FALSE
> any(is.na(customer_segmentation$Gender))
[1] FALSE
> any(is.na(customer_segmentation$Ever_Married))
[1] TRUE
> any(is.na(customer_segmentation$Age))
[1] FALSE
> any(is.na(customer_segmentation$Graduated))
[1] TRUE
> any(is.na(customer_segmentation$Profession))
[1] TRUE
> any(is.na(customer_segmentation$Work_Experience))
[1] TRUE
> any(is.na(customer_segmentation$Spending_Score))
[1] FALSE
> any(is.na(customer_segmentation$Family_Size))
[1] TRUE
> any(is.na(customer_segmentation$Var_1))
[1] TRUE
> any(is.na(customer_segmentation$Segmentation))
[1] FALSE
```

```
> class(customer_segmentation$Ever_Married)
[1] "character"
> customer_segmentation$Ever_Married[is.na(customer_segmentation$Ever_Married)]="No"
> class(customer_segmentation$Graduated)
[1] "character"
> customer_segmentation$Graduated[is.na(customer_segmentation$Graduated)]="Yes"
> class(customer_segmentation$Profession)
[1] "character"
> customer_segmentation$Profession[is.na(customer_segmentation$Profession)]="Engineer"
> class(customer_segmentation$Work_Experience)
[1] "integer"
> x=mean(customer_segmentation$Work_Experience,na.rm = TRUE)
> x
[1] 2.552587
> # From the output we can see that we got a numeric value(i.e we got a decimal value)
> #But work experience cannot be such a value. So lets either use floor or ceiling
> y=floor(x)
> y
[1] 2
> customer_segmentation$Work_Experience[is.na(customer_segmentation$Work_Experience)]=y
```

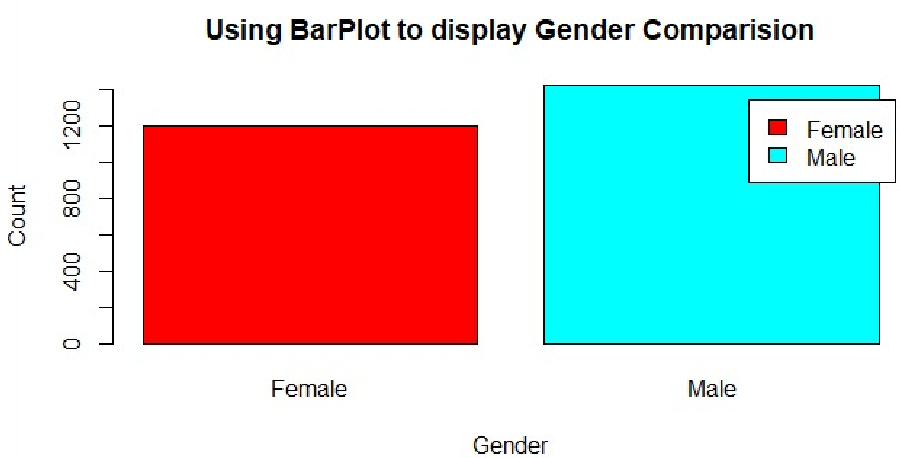
```
> class(customer_segmentation$Family_Size)
[1] "integer"
> z=mean(customer_segmentation$Family_Size,na.rm = TRUE)
> z
[1] 2.825378
> # From the output we can see that we got a numeric value(i.e we got a decimal va
> w=ceiling(z)
> w
[1] 3
> customer_segmentation$Family_Size[is.na(customer_segmentation$Family_Size)]=w
> class(customer_segmentation$Var_1)
[1] "character"
> customer_segmentation$Var_1[is.na(customer_segmentation$Var_1)]="Cat_2"
> customer_segmentation
```

```
> #Checking if there are any NA values now
> any(is.na(customer_segmentation))
[1] FALSE
>
> #Writing the updated ones into new csv file named credit_cards_details
> write.csv(customer_segmentation,"customer_segmentation_cleaned.csv")
```

5.5 Analyse the Data

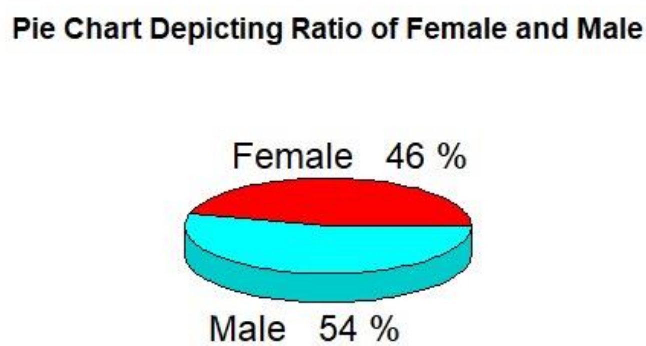
Gender Comparison

```
> #Customer Gender Visualization
> a=table(da$Gender)
> barplot(a,main="Using BarPlot to display Gender Comparision",
+         ylab="Count",
+         xlab="Gender",
+         col=rainbow(2),
+         legend=rownames(a))
```



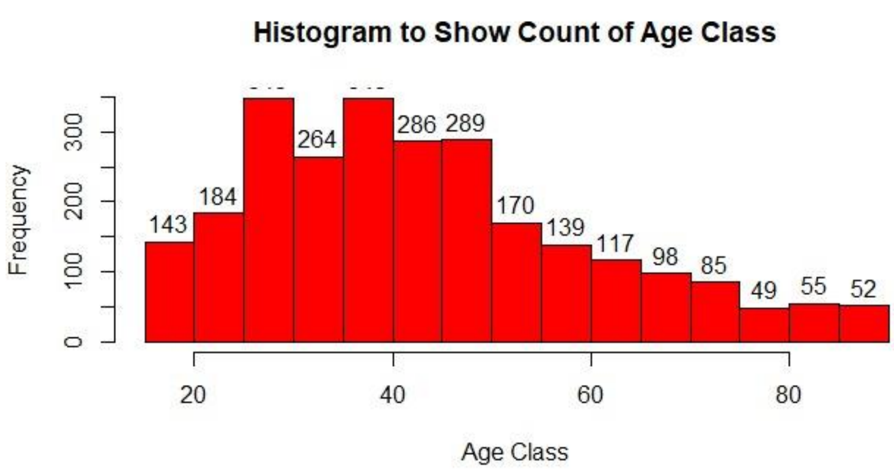
Ratio of Female and Male

```
The downloaded binary packages are in
        C:\Users\sruth\AppData\Local\Temp\Rtmp4UyKk1\downloaded_packages
> #visualizing a pie chart to observe the ratio of male and female distribution.
> pie=round(a/sum(a)*100)
> lbs=paste(c("Female","Male")," ",pie,"%",sep=" ")
> library(plotrix)
Warning message:
package 'plotrix' was built under R version 4.1.1
> pie3D(a,labels=lbs,
+       main="Pie Chart Depicting Ratio of Female and Male")
> 
```



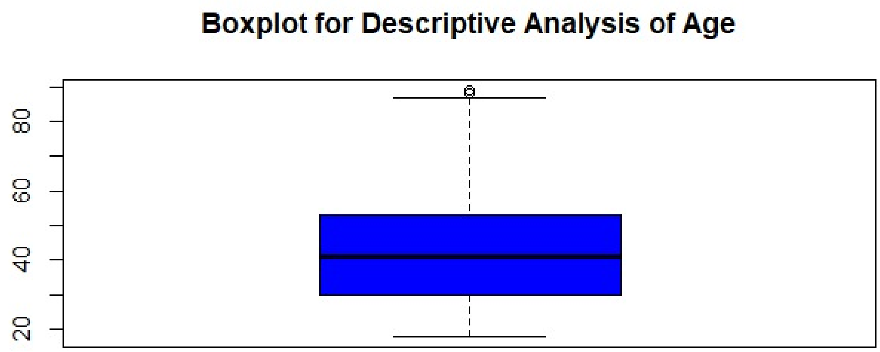
Histogram to show count of Age class

```
> hist(da$Age,
+       col="red",
+       main="Histogram to Show Count of Age Class",
+       xlab="Age Class",
+       ylab="Frequency",
+       labels=TRUE)
> 
```



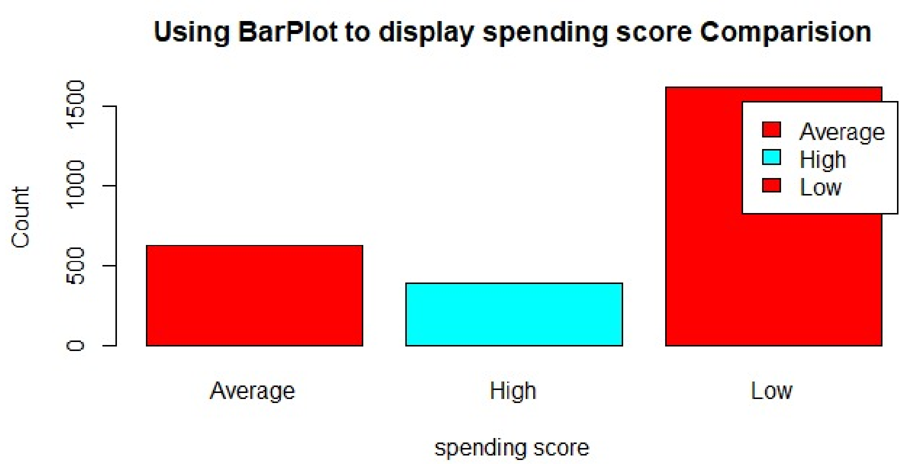
Boxplot for Descriptive Analysis of Age

```
> boxplot(da$Age,
+         col="blue",
+         main="Boxplot for Descriptive Analysis of Age")
>
```



Spending Score Comparison

```
> #Analysing the spending score of the customers
> b=table(da$Spending_Score)
> barplot(b,main="Using BarPlot to display spending score Comparision",
+         ylab="Count",
+         xlab="spending score",
+         col=rainbow(2),
+         legend=rownames(b))
>
```



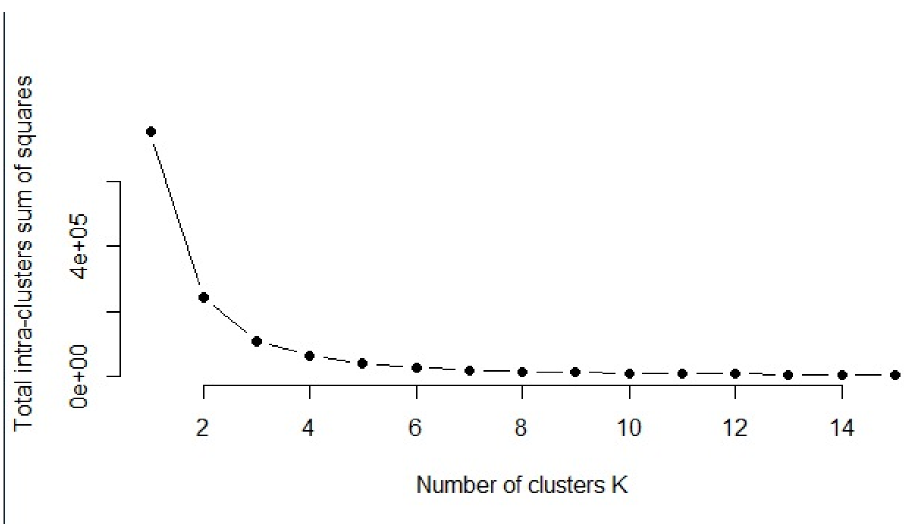
While working with clusters, you need to specify the number of clusters to use. We would like to utilize the optimal number of clusters. To help us in determining the optimal clusters, we used these three popular methods –

- **Elbow method**
- **Silhouette method**
- **Gap statistic**

5.6 K Means Clustering

1) Elbow Method

```
> #  To specify the number of clusters
> # using Elbow Method
> library(purrr)
> el <- function(k) {
+   kmeans(da[4],k,iter.max=100,nstart=100,algorithm="Lloyd" )$tot.withinss
+ }
> k.values <- 1:15
> ell_values <- map_dbl(k.values, el)
> plot(k.values, ell_values,
+      type="b", pch = 19, frame = FALSE,
+      xlab="Number of clusters K",
+      ylab="Total intra-clusters sum of squares")
```

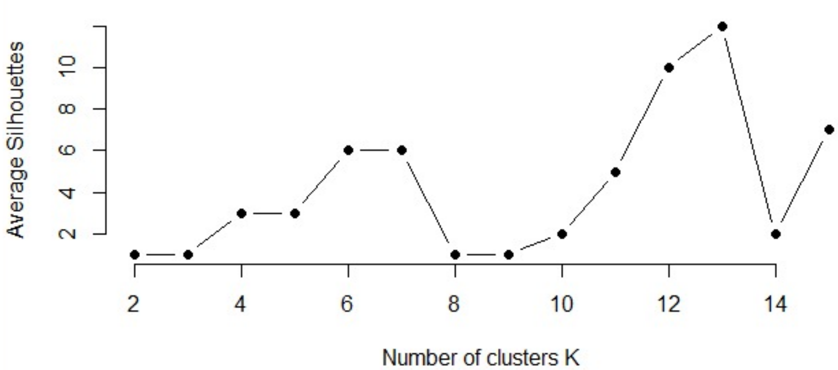


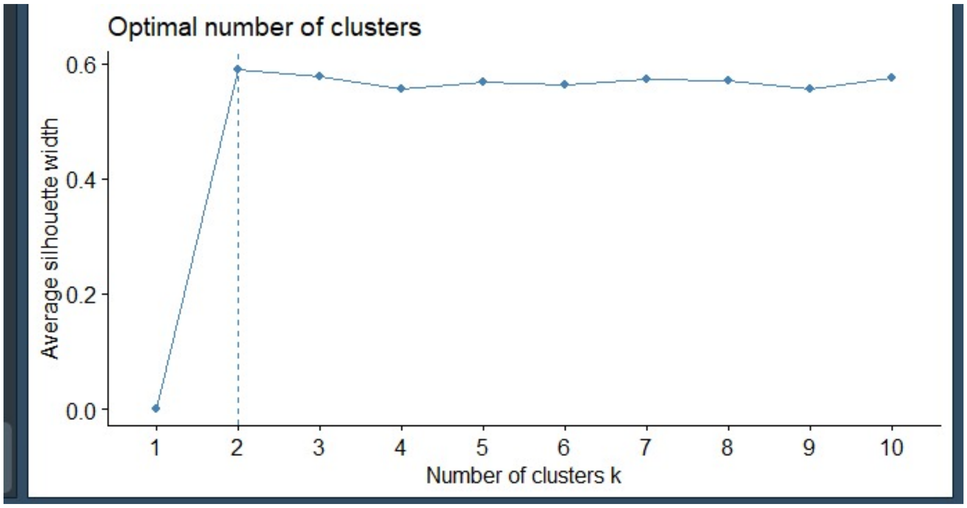
From the above graph, we conclude that 2 is the appropriate number of clusters since it seems to be appearing at the bend in the elbow plot.

2) Average Silhouette Method

We used the silhouette function in the cluster package to compute the average silhouette width. The following code computes this approach for 2-15 clusters.

```
> # 2nd Method
> #Average Silhouette Method
> library(cluster)
> library(gridExtra)
> library(grid)
> avg_sil <- function(k) {
+   km.res <- kmeans(da[4], centers = k, nstart = 25)
+   ss <- silhouette(km.res$cluster, dist(da))
+   mean(ss[3])
+ }
> # Compute and plot wss for k = 2 to k = 15
> k.values <- 2:15
> # extract avg silhouette for 2-15 clusters
> avg_sil_values <- map_dbl(k.values, avg_sil)
There were 14 warnings (use warnings() to see them)
> plot(k.values, avg_sil_values,
+      type = "b", pch = 19, frame = FALSE,
+      xlab = "Number of clusters K",
+      ylab = "Average Silhouettes")
> library(NbClust)
> library(factoextra)
> fviz_nbclust(da[4], kmeans, method = "silhouette")
```

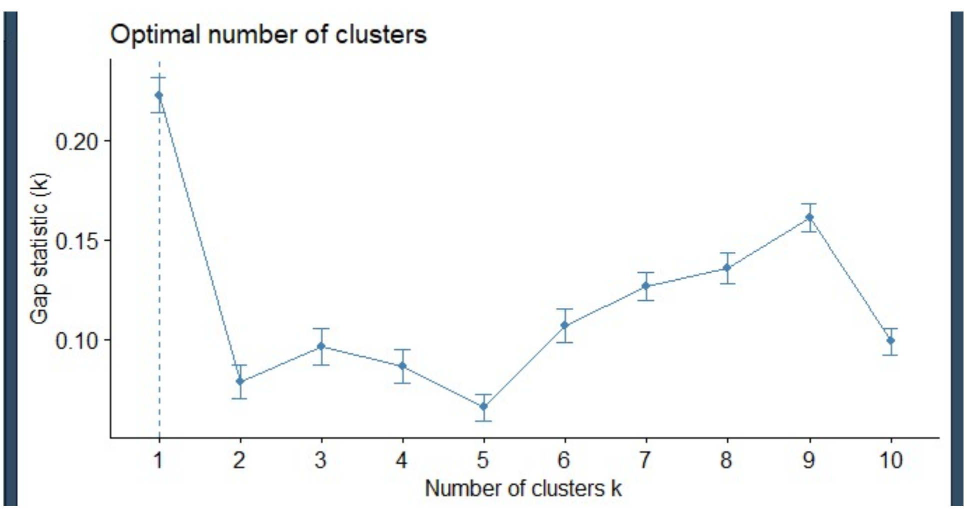




3) Gap Static Method

The gap statistic compares the total intracluster variation for different values of *k* with their expected values under null reference distribution of the data (i.e. a distribution with no obvious clustering). We can visualize the results with fviz_gap_stat which suggests four clusters as the optimal number of clusters.

```
>
> set.seed(125)
> set.seed(125)
> stat_gap <- clusGap(da[4], FUN = kmeans, nstart = 25,
+                    K.max = 10, B = 50)
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
.................................................. 50
There were 12 warnings (use warnings() to see them)
> fviz_gap_stat(stat_gap)
>
```



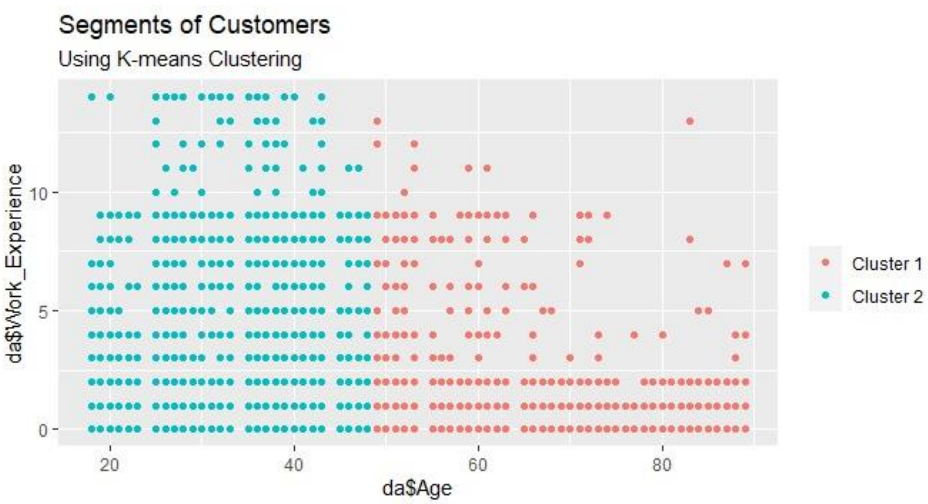
With most of these approaches suggesting 2 as the number of optimal clusters, we can perform the final analysis and extract the results using 2 clusters.

```
Console   Terminal ×   Jobs ×
R 4.1.0 · C:/Users/sruth/OneDrive/Desktop/CSE4027 LAB/
> k2<-kmeans(da[4],2,iter.max=100,nstart=50,algorithm="Lloyd")
> k2
K-means clustering with 2 clusters of sizes 872, 1755

Cluster means:
       Age
1 63.46445
2 33.80456

Clustering vector:
   [1] 2 2 1 1 2 2 1 2 1 2 2 2 1 2 2 1 2 2 1 1 1 1 2 1 2 2 1 2 1 1 1 1 2 1 2 1 1 1 1 1 2 1 2 1
  [45] 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 1 2 2 1 1 1 2 1 2 1 1 2 2 1 1 2
  [89] 2 2 1 1 2 2 2 1 2 2 1 1 1 2 2 2 2 2 2 1 2 1 1 1 2 1 1 1 2 2 1 2 1 2 2 2 2 1 2 1 2 2 1 1
 [133] 2 2 1 2 1 1 1 1 2 1 1 1 2 2 2 2 2 1 1 1 2 1 2 2 1 2 2 2 2 2 1 2 2 2 1 1 1 2 1 1 2 2 1 2
 [177] 1 1 1 1 2 2 1 1 2 2 2 1 2 1 1 1 2 2 2 2 1 2 1 1 2 2 1 1 2 1 2 2 2 1 2 1 2 1 2 2 1 2 1 2
 [221] 1 2 2 2 1 1 1 1 2 1 2 2 2 2 1 2 1 2 1 1 1 2 1 1 1 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 1 2 2
 [265] 2 1 1 2 2 1 2 1 1 1 1 2 1 2 2 2 1 1 1 1 2 2 2 1 2 2 1 2 1 1 2 1 2 2 2 1 2 2 2 1 2 1 1 2
 [309] 2 1 2 2 2 2 1 1 1 1 2 1 1 2 1 2 2 2 2 2 2 2 1 2 1 2 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2
 [353] 2 2 1 1 1 1 2 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
 [397] 2 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
 [441] 2 2 2 2 2 1 1 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 1 1 1
 [485] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2
 [529] 1 2 2 2 1 1 2 2 1 1 2 1 2 1 2 1 2 1 1 2 1 2 2 2 1 1 2 1 1 2 2 2 2 2 2 1 1 2 2 1 1 2 1
 [573] 1 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 1 1 1 2 2 2 1 2 1 2 1 2 2 1 2 2 1 2 2 2 2 2 1 2 1 1 2
 [617] 2 1 1 1 2 2 2 2 2 1 2 1 2 1 1 2 1 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 1 2 2
 [661] 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 1 1 1 2 2 2 1
 [705] 1 1 2 2 2 1 2 1 2 2 2 1 2 1 2 1 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
 [749] 2 1 2 2 1 2 2 1 1 1 2 2 2 2 2 2 2 1 1 1 1 2 2 2 1 2 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 2 1
 [793] 2 1 2 1 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2 1 2 2 1 2 2 1 2 2 1 2 1 1 2 1 1 2 1 2 1 2 2 1 1 1
 [837] 1 1 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 1 1 2 2 1 2 1 2 2 2 1 2 2 2 2 1 2 2 1 1 2 2 1
 [881] 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 1 2 2 1 1 1 2 1 2 2 2 1 2 2
 [925] 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 1 2 1 1 2 1 2 1 2 2 2 2 2 1 2 1
 [969] 2 2 2 2 1 2 1 1 2 1 2 1 1 1 2 2 2 2 2 2 2 2 1 1 2 2 2 1 1 1 2 1
 [ reached getOption("max.print") — omitted 1627 entries ]
```

```
 [ reached getOption("max.print") — omitted 1627 entries ]

Within cluster sum of squares by cluster:
[1] 117336.9 126160.0
 (between_SS / total_SS =  67.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
>
```

```
Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> pcclust=prcomp(da[4],scale=FALSE) #principal component analysis
> summary(pcclust)
Importance of components:
                         PC1
Standard deviation     16.97
Proportion of Variance  1.00
Cumulative Proportion   1.00
>
```

```
> #Visualizing The Clusters
> set.seed(1)
> ggplot(da, aes(x =da$Age , y = da$Work_Experience)) +
+   geom_point(stat = "identity", aes(color = as.factor(k2$cluster))) +
+   scale_color_discrete(name=" ",
+                        breaks=c("1", "2"),
+                        labels=c("Cluster 1", "Cluster 2")) +
+   ggtitle("Segments of Customers", subtitle = "Using K-means Clustering")
```



4) Evaluating the Model

In the output of our kmeans operation, we observe a list with several key information. From this, we conclude the useful information being –

- cluster – This is a vector of several integers that denote the cluster which has an allocation of each point.
- totss – This represents the total sum of squares.
- centers – Matrix comprising of several cluster centers
- withinss – This is a vector representing the intra-cluster sum of squares having one component per cluster.
- tot.withinss – This denotes the total intra-cluster sum of squares.
- betweenss – This is the sum of between-cluster squares.
- size – The total number of points that each cluster holds.

This is one method to evaluate the model

```
> k2$betweenss/k2$totss
[1] 0.6779022
>
```

We also additionally performed classification algorithms like Logistic Regression and decision Tree models.

5.7 Logistic Regression

We will first model the data , that is, split the dataset in the ratio in 80:20 ; training:testing respectively. Then Fit the model by using the glm() function.

```
R 4.1.0 · C:/Users/sruth/OneDrive/Desktop/CSE4027 LAB/
> getwd()
[1] "C:/Users/sruth/OneDrive/Desktop/CSE4027 LAB"
> setwd("C:/Users/sruth/OneDrive/Desktop/CSE4027 LAB")
> dir()
 [1] "BirthsKingCounty2001.txt"         "bmi_data (1).csv"
 [3] "bmi_data.csv"                     "colon.txt"
 [5] "COVID_country_wise_latest.csv"    "credit_cards.csv"
 [7] "CSE4027 LAB.Rproj"                "CSV"
 [9] "customer_segmentation_cleaned.csv" "diabetes.csv"
[11] "Diabetes_Updated.csv"             "Excel datasets"
[13] "House.xlsx"                       "Iris (2).csv"
[15] "Json"                             "lab sheeet 7_19BCD7040.docx"
[17] "Mall_Customers.csv"               "MRI.txt"
[19] "output.xlsx"                      "SalaryData.txt"
[21] "Shr.xlsx"                         "STATA"
[23] "Student_Data_cleaned.csv"         "Student_Data_Uncleaned.csv"
[25] "StudentsPerformance.csv"          "txt"
[27] "weatherHistory.csv"
> da=read.csv("customer_segmentation_cleaned.csv")
> da
    i..ID Gender Ever_Married Age Graduated    Profession Work_Experience Spending_Score
1  458989 Female          Yes  36       Yes      Engineer               0            Low
2  458994   Male          Yes  37       Yes    Healthcare               8        Average
3  458996 Female          Yes  69        No      Engineer               0            Low
4  459000   Male          Yes  59        No     Executive              11           High
5  459001 Female           No  19        No     Marketing               2            Low
6  459003   Male          Yes  47       Yes        Doctor               0           High
7  459005   Male          Yes  61       Yes        Doctor               5            Low
8  459008 Female          Yes  47       Yes        Artist               1        Average
9  459013   Male          Yes  50       Yes        Artist               2        Average
10 459014   Male           No  19        No    Healthcare               0            Low
```

```
> #data modelling
> library(caTools)
> set.seed(100)
> data_sample=sample.split(da$Gender,SplitRatio=0.80)
> train_data = subset(da,data_sample==TRUE)
> test_data = subset(da,data_sample==FALSE)
> dim(train_data)
[1] 2101   11
> dim(test_data)
[1] 526  11
```

```
> #Fitting The Logistic Regression Model
> Logistic_Model=glm(as.factor(Gender)~.,test_data,family=binomial())
> summary(Logistic_Model)

Call:
glm(formula = as.factor(Gender) ~ ., family = binomial(), data = test_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3164  -1.0564   0.3753   1.0011   2.1801

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              2.088e+01  1.761e+01   1.186 0.235812
i..ID                   -4.425e-05  3.782e-05  -1.170 0.242020
Ever_MarriedYes          1.121e+00  3.084e-01   3.634 0.000279 ***
Age                     -7.534e-03  9.535e-03  -0.790 0.429398
GraduatedYes            -4.207e-01  2.431e-01  -1.731 0.083456 .
ProfessionDoctor         2.086e-01  3.744e-01   0.557 0.577487
ProfessionEngineer      -1.037e+00  3.646e-01  -2.845 0.004447 **
ProfessionEntertainment  5.003e-01  3.683e-01   1.359 0.174271
ProfessionExecutive      2.349e+00  6.666e-01   3.524 0.000425 ***
ProfessionHealthcare     6.384e-01  3.812e-01   1.675 0.094001 .
ProfessionHomemaker     -1.268e+00  5.872e-01  -2.159 0.030828 *
ProfessionLawyer        -2.317e-01  4.524e-01  -0.512 0.608490
ProfessionMarketing      1.015e+00  5.572e-01   1.821 0.068564 .
Work_Experience         -5.417e-02  3.207e-02  -1.689 0.091147 .
Spending_ScoreHigh      -2.511e-01  4.009e-01  -0.626 0.531088
Spending_ScoreLow        3.696e-01  3.029e-01   1.220 0.222411
Family_Size              1.608e-01  7.266e-02   2.214 0.026851 *
Var_1Cat_2              -1.134e+00  1.404e+00  -0.808 0.419133
Var_1Cat_3              -1.025e+00  1.384e+00  -0.740 0.459149
Var_1Cat_4              -1.317e+00  1.383e+00  -0.952 0.340979
Var_1Cat_5               1.324e+01  6.233e+02   0.021 0.983051
Var_1Cat_6              -8.682e-01  1.358e+00  -0.639 0.522683
Var_1Cat_7              -1.513e+00  1.536e+00  -0.985 0.324701
```

```
Console   Terminal ×   Jobs ×
R 4.1.0 · C:/Users/sruth/OneDrive/Desktop/CSE4027 LAB/
Var_1Cat_3             -1.025e+00  1.384e+00  -0.740 0.459149
Var_1Cat_4             -1.317e+00  1.383e+00  -0.952 0.340979
Var_1Cat_5              1.324e+01  6.233e+02   0.021 0.983051
Var_1Cat_6             -8.682e-01  1.358e+00  -0.639 0.522683
Var_1Cat_7             -1.513e+00  1.536e+00  -0.985 0.324701
SegmentationB           1.578e-01  2.835e-01   0.557 0.577701
SegmentationC          -7.173e-02  2.856e-01  -0.251 0.801666
SegmentationD           2.013e-01  2.537e-01   0.793 0.427585
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 725.51  on 525  degrees of freedom
Residual deviance: 626.08  on 500  degrees of freedom
AIC: 678.08

Number of Fisher Scoring iterations: 13
```

Predicting the accuracy of the model by building the confusion matrix

```
> #predicting the accuracy
> res<-predict(Logistic_Model,test_data,type = "response")
> res
        11         17         32         35         41         57         71         75 
0.77276799 0.61955630 0.41323501 0.51591738 0.85026137 0.50944553 0.78046407 0.46164220 
        77         78         80         83         89         96         99        103 
0.79738149 0.59062440 0.63239573 0.56496562 0.47380179 0.56530932 0.58362326 0.80754380 
       106        110        120        126        144        158        166        171 
0.62068481 0.48451941 0.32424174 0.65352378 0.54644423 0.75335501 0.71755334 0.56293152 
       174        177        184        189        191        199        201        203 
0.15765058 0.41479671 0.46874931 0.52045997 0.93163103 0.63039509 0.76420504 0.53288401 
       211        219        225        235        239        240        246        247 
0.56808624 0.54507992 0.68274554 0.94467857 0.53483609 0.90128539 0.54383505 0.54644985 
       252        257        258        270        272        275        279        284 
0.17247733 0.49166888 0.76909827 0.78849906 0.54145857 0.73198117 0.82951764 0.61791076 
```

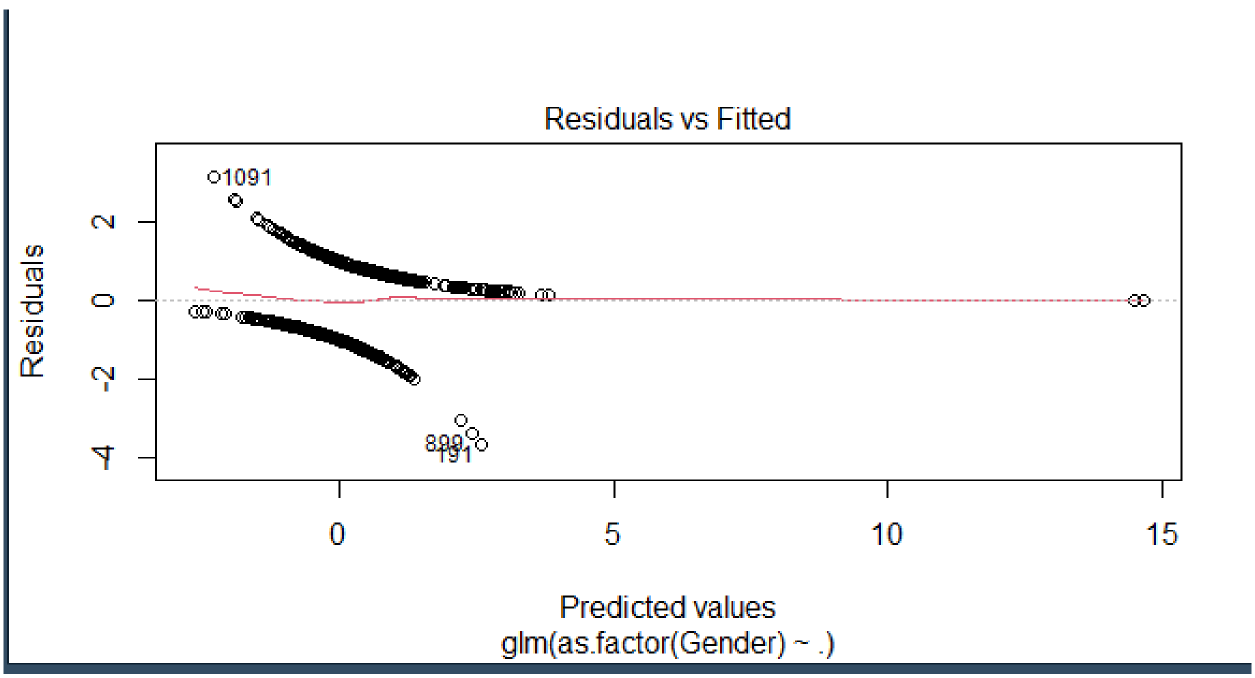
```
> restr<-predict(Logistic_Model,train_data,type = "response")
> restr
         1          2          3          4          5          6          7          8 
0.44501034 0.72481650 0.44845675 0.91343773 0.81379598 0.57088427 0.71706345 0.66235462 
         9         10         12         13         14         15         16         18 
0.67133227 0.79641960 0.84638112 0.68371732 0.64471165 0.69499798 0.59780382 0.61850899 
        19         20         21         22         23         24         25         26 
0.55906268 0.50483667 0.94344384 0.59617842 0.45395921 0.93647054 0.66201426 0.96236111 
        27         28         29         30         31         33         34         36 
0.38933485 0.60264142 0.50134839 0.35397597 0.61749604 0.56367018 0.86205525 0.21027164 
        37         38         39         40         42         43         44         45 
0.64979430 0.59797313 0.63774410 0.52735539 0.90988430 0.69543027 0.67696207 0.54546466 
        46         47         48         49         50         51         52         53 
0.49224756 0.63272939 0.63679283 0.79321198 0.72838005 0.42014703 0.51949975 0.41305360 
```

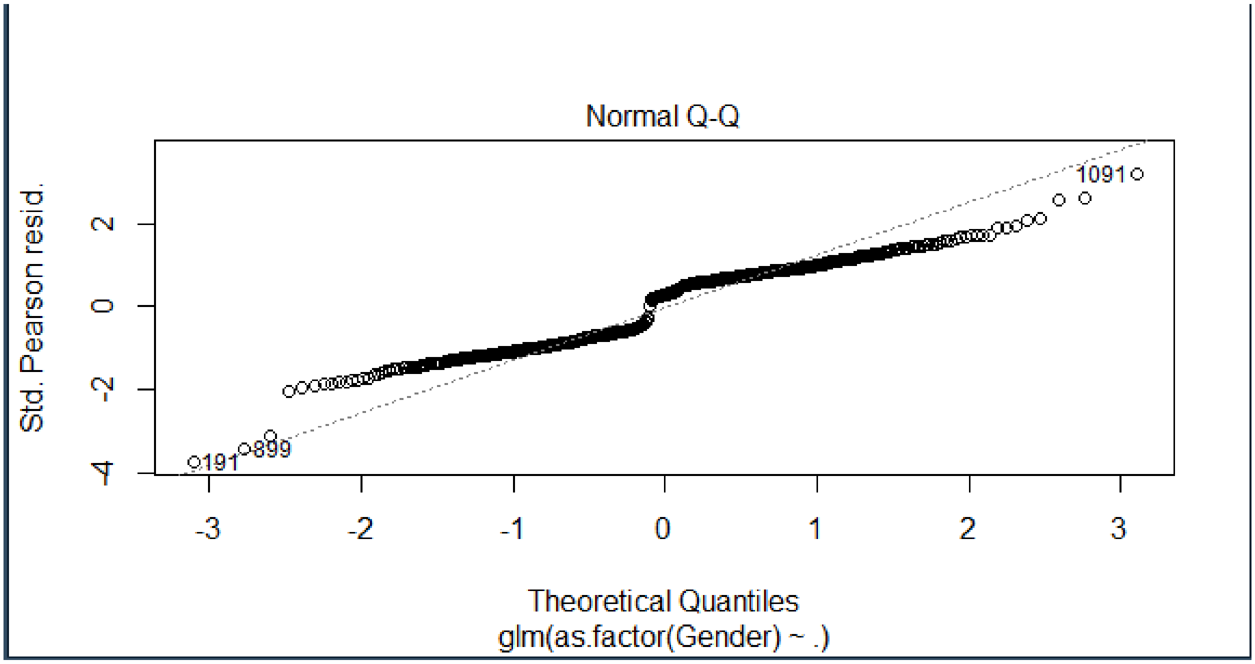
```
> #Building the confusion matrix for training data
> cm<-table(Actual_Value=train_data$Gender,Predicted_Value=restr>0.5)
> #Accuracy
> accuracy=(cm[[1,1]]+cm[[2,2]])/sum(cm)
> accuracy
[1] 0.6325559
```

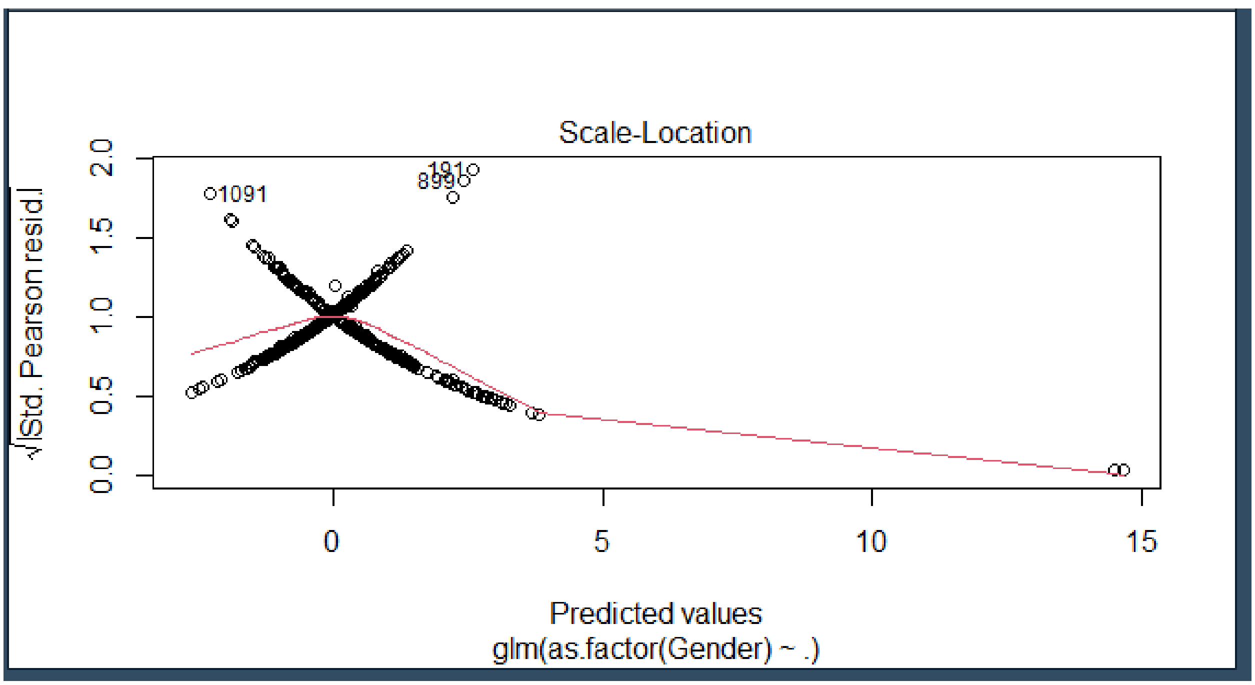
```
> #Building  the confusion matrix for testing Data
> cmte<-table(Actual_Value=test_data$Gender,Predicted_Value=res>0.5)
> cmte
            Predicted_Value
Actual_Value FALSE TRUE
      Female   143   98
      Male      76  209
> #accuracy
> accuracy=(cmte[[1,1]]+cmte[[2,2]])/sum(cmte)
> accuracy
[1] 0.6692015
```

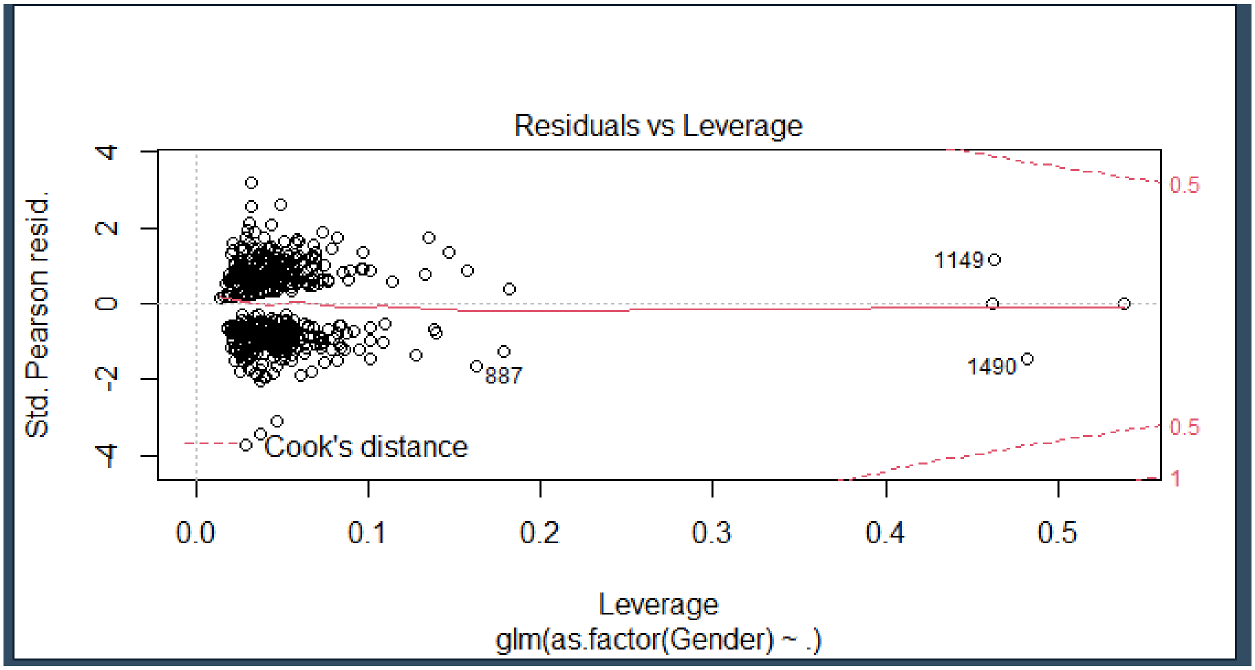
After we have summarised our model, we will visual it through the following plots

```
> plot(Logistic_Model)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
```




Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(as.factor(Gender) ~ .)
1091
191
899


Scale-Location
√|Std. Pearson resid.|
Predicted values
glm(as.factor(Gender) ~ .)
1091
191
899

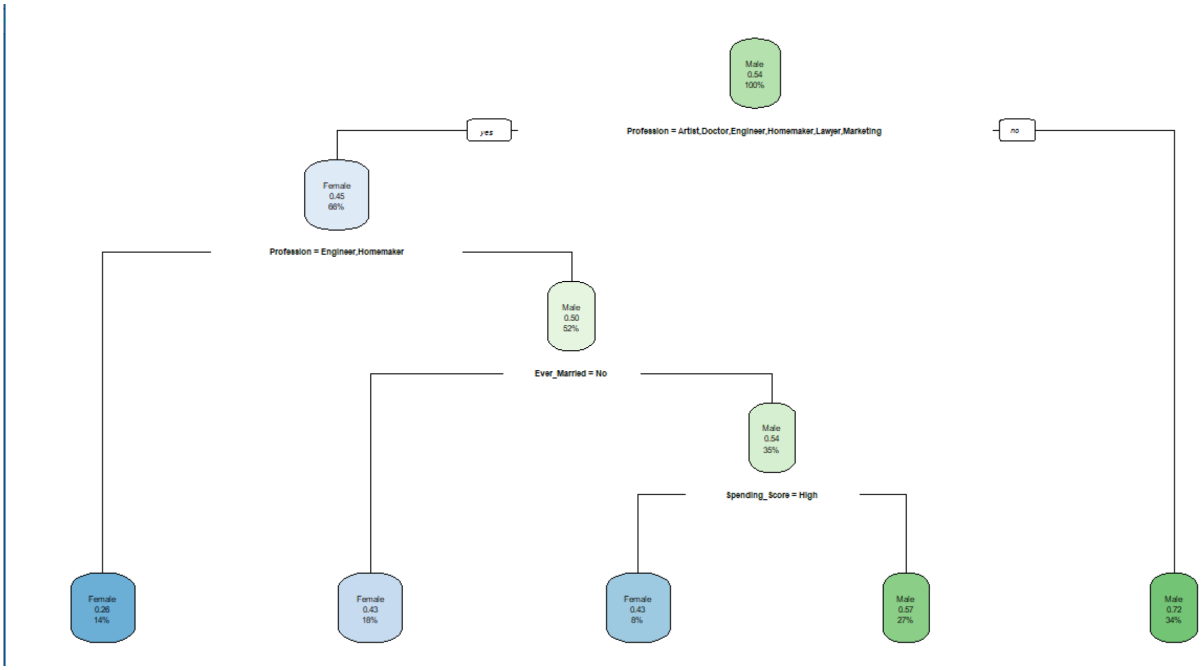


5.8 Decision Tree

We will now implement our decision tree model and will plot it using the rpart.plot() function. We will specifically use the recursive parting to plot the decision tree.

```
>
> #decision tree
> library(rpart)
Warning message:
package 'rpart' was built under R version 4.1.2
> library(rpart.plot)
Warning message:
package 'rpart.plot' was built under R version 4.1.2
> decisionTree_model <- rpart(as.factor(Gender) ~ . , da, method = 'class')
> predicted_val <- predict(decisionTree_model, da, type = 'class')
> probability <- predict(decisionTree_model, da, type = 'prob')
> rpart.plot(decisionTree_model)
>
```

Decision tree Model




```
R 4.1.0 · C:/Users/sruth/OneDrive/Desktop/CSE4027 LAB/
> #predicting the accuracy
> res<-predict(Logistic_Model,test_data,type = "response")
> res
        11         17         32         35         41         57         71         75 
0.77276799 0.61955630 0.41323501 0.51591738 0.85026137 0.50944553 0.78046407 0.46164220 
        77         78         80         83         89         96         99        103 
0.79738149 0.59062440 0.63239573 0.56496562 0.47380179 0.56530932 0.58362326 0.80754380 
       106        110        120        126        144        158        166        171 
0.62068481 0.48451941 0.32424174 0.65352378 0.54644423 0.75335501 0.71755334 0.56293152 
       174        177        184        189        191        199        201        203 
0.15765058 0.41479671 0.46874931 0.52045997 0.93163103 0.63039509 0.76420504 0.53288401 
       211        219        225        235        239        240        246        247 
0.56808624 0.54507992 0.68274554 0.94467857 0.53483609 0.90128539 0.54383505 0.54644985 
       252        257        258        270        272        275        279        284 
0.17247733 0.49166888 0.76909827 0.78849906 0.54145857 0.73198117 0.82951764 0.61791076 
       289        292        301        302        307        309        326        330 
0.33073461 0.64587947 0.53610532 0.31499518 0.73482775 0.82045904 0.39380722 0.27176437 
       339        341        344        348        349        355        370        371 
```

```
Levels: Female Male
> tr<-predict(decisionTree_model,train_data,type = "class")
> tr
     1      2      3      4      5      6      7      8      9     10     12     13     14 
Female   Male Female   Male Female Female   Male   Male   Male   Male   Male   Male   Male 
    15     16     18     19     20     21     22     23     24     25     26     27     28 
Female   Male   Male   Male   Male   Male Female Female   Male   Male   Male Female   Male 
    29     30     31     33     34     36     37     38     39     40     42     43     44 
Female Female   Male Female   Male Female   Male   Male   Male Female   Male   Male   Male 
    45     46     47     48     49     50     51     52     53     54     55     56     58 
Female Female   Male   Male   Male Female Female   Male Female   Male   Male Female Female 
    59     60     61     62     63     64     65     66     67     68     69     70     72 
```

```
> #Building the confusion matrix for training data
> cm<-table(Actual_Value=train_data$Gender,Predicted_Value= tr)
> cm
            Predicted_Value
Actual_Value Female Male
      Female    504  458
      Male      306  833
> #Accuracy
> accuracy=(cm[[1,1]]+cm[[2,2]])/sum(cm)
> accuracy
[1] 0.6363636
> #Building  the confusion matrix for testing Data
> cmte<-table(Actual_Value=test_data$Gender,Predicted_Value= tp)
> cmte
            Predicted_Value
Actual_Value Female Male
      Female    140  101
      Male       80  205
> #accuracy
> accuracy=(cmte[[1,1]]+cmte[[2,2]])/sum(cmte)
> #accuracy
> accuracy=(cmte[[1,1]]+cmte[[2,2]])/sum(cmte)
> accuracy
[1] 0.6558935
> 
```

6.CONCLUSION

Customer segmentation is a way to improve communication with the customer, to know the wishes of the customer, and customer activity so that appropriate communication can be built. Customer Segmentation needed to get potential customers used to increase profits. Potential customer data can be used to provide service characteristics of the customer including ecommerce services as a media buying and selling online. This paper discusses several components to do customer segmentation, Customer segmentation is an activity to divide customers or items into groups that have the same characteristics. Data that is needed for customer segmentation are internal data and external data. The internal data include demographic data and data purchase history, while the external data include cookies and server logs. Internal data can be obtained from a database when customers do registration or transactions and external data can be obtained from a web server or other source. We have concluded that K-means clustering is the best method. With the help of clustering, we can understand the variables much better, prompting us to take careful decisions. With the identification of customers, companies can release products and services that target customers based on several parameters like income, age, spending patterns, etc. Furthermore, more complex patterns like product reviews are taken into consideration for better segmentation.

7.REFERENCE

https://www.google.com/amp/s/www.intercom.com/blog/customer-segmentation/%3famp

https://www.mycustomer.com/hr-glossary/customer-segmentation

https://www.yieldify.com/blog/types-of-market-segmentation/

8.BIBLIOGRAPHY

(1) Al-Qaed F, Sutcliffe A. Adaptive Decision Support System (ADSS) for B2C E-Commerce. 2006 ICEC Eighth Int Conf Electron Commer Proc NEW E-COMMERCE Innov Conqu Curr BARRIERS, Obs LIMITATIONS TO Conduct Success Bus INTERNET. 2006:492-503.

(2) Mobasher B, Cooley R, Srivastava J. Automatic Personalization Based on Web Usage Mining. Commun ACM. 2000;43(8).

(3) Cherna Y, Tzenga G. Measuring Consumer Loyalty of B2C e-Retailing Service by Fuzzy Integral: a FANP-Based Synthetic Model. In: International Conference on Fuzzy Theory and Its Applications I FUZZY.; 2012:48-56.