PROJECT ON DATA ANALYSIS

Amodemaja Olalekan Quzim 500662930

2023-12-03

Introduction

Leading mobile phone service provider BangorTelco is addressing the issue of retention of clients. The data science team wants to create a prediction model that will help them determine which clients are most likely to leave when their contracts expire. This focused strategy will cut expenses while optimising incentive offers.

Data Retrieval

The BangorTelco IT department has provided access to the corporate database, which is an extensive collection of data that includes 20,000 previous clients. The dataset, kept in a database called 'BangorTelco_Customers,' contains vital information about customers, such as usage trends, demographics, and the variable 'LEAVE,' which indicates whether a client left or stayed at the end of their contract.

We will retrieve all the necessary information to start our research using SQL. To build a decision tree model that can forecast the probability if a customer will leave or stay, it is important that this data retrieval stage be completed to obtain insights into consumer behavior.

```
#install my sql and load library
options(repos = c(CRAN = "https://cran.r-project.org"))

install.packages("RMySQL")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'RMySQL' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'RMySQL'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Windows\AppData\Local\R\win-library\4.3\00LOCK\RMySQL\libs\x64\RMySQL.dll
## to C:\Users\Windows\AppData\Local\R\win-library\4.3\RMySQL\libs\x64\RMySQL.dll:
## Permission denied

## Warning: restored 'RMySQL'

##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
library(RMySQL)
```

```
## Loading required package: DBI
```

```
library(ggplot2)
USER<- "root"
PASSWORD<- "Baba4every"
HOST<- "localhost"
DBNAME<- "world"            #the database we want to connect, that is the one we  created during the in

SQLdatabase<- dbConnect(MySQL(), user= USER, password<-PASSWORD, host = HOST, dbname = DBNAME, port=330
bangortelcodata<- dbGetQuery(SQLdatabase, statement = "SELECT * from world.bangorcustomerchurn")
dbDisconnect(SQLdatabase)
```

```
## [1] TRUE
```

```
#understand the data
head (bangortelcodata)
```

```
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE
## 1 BTLC-007761    zero  89318       0        0 162233           266
## 2 BTLC-007682     one 142814     187       17 346690           716
## 3 BTLC-002228    zero  55675       0       32 792662           257
## 4 BTLC-011752     one  39559       0        0 416439           165
## 5 BTLC-015958    zero 145081       0        0 341108           583
## 6 BTLC-013969     one 120631      66       17 467811           884
##   OVER_15MINS_CALLS_PER_MONTH AVERAGE_CALL_DURATION REPORTED_SATISFACTION
## 1                           1                    12                 unsat
## 2                          24                     4                 unsat
## 3                           1                     1            very_unsat
## 4                           0                    15              very_sat
## 5                           0                     9                   avg
## 6                           4                     6                   sat
##   REPORTED_USAGE_LEVEL CONSIDERING_CHANGE_OF_PLAN LEAVE
## 1          very_little                considering  STAY
## 2                 high                considering LEAVE
## 3          very_little              never_thought  STAY
## 4                 high                considering  STAY
## 5                  avg                         no LEAVE
## 6            very_high                considering LEAVE
```

```
tail(bangortelcodata)
```

```
##        CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE
## 19995 BTLC-004882     one  79197      64        0 268908           378
## 19996 BTLC-002655     one  50798      47        0 554096           268
## 19997 BTLC-006427     one 116094      54       40 952072           813
## 19998 BTLC-005080     one 127584       0       55 268961           513
## 19999 BTLC-011352     one  46954       0        0 735459           155
## 20000 BTLC-000554    zero 141086      77       86 157602           479
##       OVER_15MINS_CALLS_PER_MONTH AVERAGE_CALL_DURATION REPORTED_SATISFACTION
```

```
## 19995                          5                    13           very_unsat
## 19996                          3                     8                unsat
## 19997                          3                    13                unsat
## 19998                          5                     2                unsat
## 19999                          0                     8                unsat
## 20000                          3                     1           very_unsat
##       REPORTED_USAGE_LEVEL CONSIDERING_CHANGE_OF_PLAN LEAVE
## 19995                 high                considering LEAVE
## 19996            very_high   actively_looking_into_it  STAY
## 19997               little                         no  STAY
## 19998            very_high                considering LEAVE
## 19999          very_little                         no  STAY
## 20000                  avg                considering  STAY
```

```
dim(bangortelcodata )
```

```
## [1] 20000    13
```

```
class(bangortelcodata$LEAVE)
```

```
## [1] "character"
```

Data Exploaration

In this stage, we will be performing some basic explorations such as understanding the data,viewing the nature of our data,converting attributes to the necessary data type. Key insights into customer behaviour and attributes can be obtained by examining the BangorTelco customer dataset. COLLEGE, INCOME, OVERAGE, REPORTED_SATISFACTION, and other factors are among the ones we examine. Finding trends and possible indicators of client attrition, this investigation offers a sophisticated comprehension of the data. # Findings: Demographics; College Distribution- Level of Education Income level Overcharge per month

Usage Pattern; Consideration of plans Usage level Satisfaction Level

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
View(bangortelcodata)

head(select(.data=bangortelcodata, REPORTED_SATISFACTION:LEAVE,1:10))
```

```
##   REPORTED_SATISFACTION REPORTED_USAGE_LEVEL CONSIDERING_CHANGE_OF_PLAN LEAVE
## 1                 unsat          very_little                considering  STAY
## 2                 unsat                 high                considering LEAVE
## 3            very_unsat          very_little              never_thought  STAY
## 4              very_sat                 high                considering  STAY
## 5                   avg                  avg                         no LEAVE
## 6                   sat            very_high                considering LEAVE
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE
## 1 BTLC-007761    zero  89318       0        0 162233           266
## 2 BTLC-007682     one 142814     187       17 346690           716
## 3 BTLC-002228    zero  55675       0       32 792662           257
## 4 BTLC-011752     one  39559       0        0 416439           165
## 5 BTLC-015958    zero 145081       0        0 341108           583
## 6 BTLC-013969     one 120631      66       17 467811           884
##   OVER_15MINS_CALLS_PER_MONTH AVERAGE_CALL_DURATION
## 1                           1                    12
## 2                          24                     4
## 3                           1                     1
## 4                           0                    15
## 5                           0                     9
## 6                           4                     6
```

```
summarise(.data=bangortelcodata,
          AverageIncome=mean(INCOME,na.rm=TRUE))
```

```
##   AverageIncome
## 1      80281.45
```

```
sample_n(tbl=bangortelcodata,10)
```

```
##     CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE
## 1  BTLC-004112     one 115506       0       89 957014           351
## 2  BTLC-014606     one  81896       0        6 995366           390
## 3  BTLC-012978    zero 106966     198        0 834935           769
## 4  BTLC-021384    zero  52012      54        0 215884           278
## 5  BTLC-013199    zero  48023       0        0 235924           137
## 6  BTLC-016211     one  23865       0       16 934835           227
## 7  BTLC-011314    zero  29745       0       24 583768           148
## 8  BTLC-014735    zero  37562     184        0 656851           206
## 9  BTLC-003343    zero  40582       0       86 281426           151
## 10 BTLC-004035    zero  29713     222        0 536210           138
##    OVER_15MINS_CALLS_PER_MONTH AVERAGE_CALL_DURATION REPORTED_SATISFACTION
## 1                            0                     1              very_sat
## 2                            1                     6                   avg
## 3                           20                    15              very_sat
## 4                            3                    10                 unsat
## 5                            0                    14                   avg
## 6                            0                     4              very_sat
## 7                            1                     5                   avg
## 8                           19                    14                 unsat
## 9                            1                     2            very_unsat
## 10                          16                    12            very_unsat
##    REPORTED_USAGE_LEVEL CONSIDERING_CHANGE_OF_PLAN LEAVE
```

```
## 1           very_little                         no LEAVE
## 2                   avg               considering  STAY
## 3           very_little             never_thought LEAVE
## 4             very_high                         no  STAY
## 5             very_high             never_thought  STAY
## 6             very_high                         no  STAY
## 7             very_high                         no  STAY
## 8             very_high                         no LEAVE
## 9           very_little   actively_looking_into_it  STAY
## 10            very_high                considering  STAY
```

```
head(sample_frac(tbl=bangortelcodata,0.01))
```

```
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE
## 1 BTLC-006171     one 106779      47        0 361315           427
## 2 BTLC-010202     one  22951      46       48 690672           202
## 3 BTLC-004046    zero  68845       0        0 278871           250
## 4 BTLC-002744     one  82608     179       21 528544           214
## 5 BTLC-014652    zero 105391     238       17 461297           430
## 6 BTLC-013598    zero  96567       0       88 187751           372
##   OVER_15MINS_CALLS_PER_MONTH AVERAGE_CALL_DURATION REPORTED_SATISFACTION
## 1                           4                    15              very_sat
## 2                           3                     6            very_unsat
## 3                           1                     9                 unsat
## 4                          29                     5            very_unsat
## 5                          15                     4            very_unsat
## 6                           0                     2              very_sat
##   REPORTED_USAGE_LEVEL CONSIDERING_CHANGE_OF_PLAN LEAVE
## 1               little   actively_looking_into_it LEAVE
## 2            very_high                         no  STAY
## 3          very_little               considering LEAVE
## 4                 high               considering  STAY
## 5                  avg   actively_looking_into_it LEAVE
## 6            very_high               considering LEAVE
```

```
#rename columns
colnames(bangortelcodata)
```

```
##  [1] "CUSTOMERID"                 "COLLEGE"
##  [3] "INCOME"                     "OVERAGE"
##  [5] "LEFTOVER"                   "HOUSE"
##  [7] "HANDSET_PRICE"              "OVER_15MINS_CALLS_PER_MONTH"
##  [9] "AVERAGE_CALL_DURATION"      "REPORTED_SATISFACTION"
## [11] "REPORTED_USAGE_LEVEL"       "CONSIDERING_CHANGE_OF_PLAN"
## [13] "LEAVE"
```

```
bangortelcodata<- bangortelcodata %>% rename(RETENTION = LEAVE)
bangortelcodata<- bangortelcodata %>% rename(USAGE_LEVEL=REPORTED_USAGE_LEVEL)
bangortelcodata<- bangortelcodata %>% rename(PLUS_15MINS = OVER_15MINS_CALLS_PER_MONTH)
bangortelcodata<- bangortelcodata %>% rename(PLAN_CHANGE=CONSIDERING_CHANGE_OF_PLAN)

bycustomer_retention<-group_by(.data=bangortelcodata,RETENTION)
```

```
customer_retention_summary<- summarize(.data= bycustomer_retention,

                       #Summarise the table grouped by LEAVE COLUMN
                        count=n(),                          #Count rows in each  group
                        AverageIncome=mean(INCOME,na.rm=TRUE),
                        .groups="drop")
customer_retention_summary
```

```
## # A tibble: 2 x 3
##   RETENTION count AverageIncome
##   <chr>     <int>         <dbl>
## 1 LEAVE      9852        84356.
## 2 STAY      10148        76326.
```

```
bycustomer_retention<-group_by(.data=bangortelcodata,RETENTION,PLAN_CHANGE)
customer_retention_summary<- summarize(.data= bycustomer_retention,

                       #Summarise the table grouped by LEAVE COLUMN
                        count=n(),                          #Count rows in each  group
                        AverageIncome=mean(INCOME,na.rm=TRUE),
                        .groups="drop")
customer_retention_summary
```

```
## # A tibble: 10 x 4
##    RETENTION PLAN_CHANGE              count AverageIncome
##    <chr>     <chr>                    <int>         <dbl>
##  1 LEAVE     actively_looking_into_it  2453        84124.
##  2 LEAVE     considering               3879        83951.
##  3 LEAVE     never_thought              966        83559.
##  4 LEAVE     no                        2013        85537.
##  5 LEAVE     perhaps                    541        85337.
##  6 STAY      actively_looking_into_it  2541        76523.
##  7 STAY      considering               4041        76254.
##  8 STAY      never_thought             1029        76022.
##  9 STAY      no                        2025        76345.
## 10 STAY      perhaps                    512        76445.
```

```
bycustomer_retention<-group_by(.data=bangortelcodata,RETENTION,PLAN_CHANGE,USAGE_LEVEL)
customer_retention_summary<- summarize(.data= bycustomer_retention,

                       #Summarise the table grouped by LEAVE COLUMN
                        count=n(),                          #Count rows in each  group
                        AverageIncome=mean(INCOME,na.rm=TRUE),
                        .groups="drop")
customer_retention_summary
```

```
## # A tibble: 50 x 5
##    RETENTION PLAN_CHANGE              USAGE_LEVEL count AverageIncome
##    <chr>     <chr>                    <chr>       <int>         <dbl>
##  1 LEAVE     actively_looking_into_it avg           122        88253.
##  2 LEAVE     actively_looking_into_it high          216        83885.
```

```
##  3 LEAVE     actively_looking_into_it little        973        84052.
##  4 LEAVE     actively_looking_into_it very_high     621        82659.
##  5 LEAVE     actively_looking_into_it very_little   521        85139.
##  6 LEAVE     considering              avg           185        88907.
##  7 LEAVE     considering              high          381        83461.
##  8 LEAVE     considering              little       1531        85381.
##  9 LEAVE     considering              very_high    1022        81735.
## 10 LEAVE     considering              very_little   760        83086.
## # i 40 more rows
```

```
select(bycustomer_retention, RETENTION,USAGE_LEVEL,PLAN_CHANGE)
```

```
## # A tibble: 20,000 x 3
## # Groups:   RETENTION, PLAN_CHANGE, USAGE_LEVEL [50]
##    RETENTION USAGE_LEVEL PLAN_CHANGE
##    <chr>     <chr>       <chr>
##  1 STAY      very_little considering
##  2 LEAVE     high        considering
##  3 STAY      very_little never_thought
##  4 STAY      high        considering
##  5 LEAVE     avg         no
##  6 LEAVE     very_high   considering
##  7 LEAVE     very_high   considering
##  8 STAY      high        considering
##  9 LEAVE     little      actively_looking_into_it
## 10 STAY      little      considering
## # i 19,990 more rows
```

```
# as 'COLLEGE' is the variable with values 'zero' and 'one'
# Converting 'zero' to 0 and 'one' to 1
bangortelcodata <- bangortelcodata%>%
  mutate(COLLEGE=ifelse(COLLEGE == 'one', 1, 0),
         RETENTION=ifelse(RETENTION=="STAY",0,1))
head(bangortelcodata)
```

```
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE PLUS_15MINS
## 1 BTLC-007761       0  89318       0        0 162233           266           1
## 2 BTLC-007682       1 142814     187       17 346690           716          24
## 3 BTLC-002228       0  55675       0       32 792662           257           1
## 4 BTLC-011752       1  39559       0        0 416439           165           0
## 5 BTLC-015958       0 145081       0        0 341108           583           0
## 6 BTLC-013969       1 120631      66       17 467811           884           4
##   AVERAGE_CALL_DURATION REPORTED_SATISFACTION USAGE_LEVEL   PLAN_CHANGE
## 1                    12                 unsat very_little   considering
## 2                     4                 unsat        high   considering
## 3                     1            very_unsat very_little never_thought
## 4                    15              very_sat        high   considering
## 5                     9                   avg         avg            no
## 6                     6                   sat   very_high   considering
##   RETENTION
## 1         0
## 2         1
## 3         0
```

```
## 4         0
## 5         1
## 6         1
```

```
# Assuming 'bangortelcodata' is your dataframe and 'REPORTED_SATISFACTION' is the column
bangortelcodata$REPORTED_SATISFACTION <- factor(bangortelcodata$REPORTED_SATISFACTION)


head(bangortelcodata)
```

```
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE PLUS_15MINS
## 1 BTLC-007761       0  89318       0        0 162233           266           1
## 2 BTLC-007682       1 142814     187       17 346690           716          24
## 3 BTLC-002228       0  55675       0       32 792662           257           1
## 4 BTLC-011752       1  39559       0        0 416439           165           0
## 5 BTLC-015958       0 145081       0        0 341108           583           0
## 6 BTLC-013969       1 120631      66       17 467811           884           4
##   AVERAGE_CALL_DURATION REPORTED_SATISFACTION USAGE_LEVEL   PLAN_CHANGE
## 1                    12                unsat very_little   considering
## 2                     4                unsat        high   considering
## 3                     1           very_unsat very_little never_thought
## 4                    15             very_sat        high   considering
## 5                     9                  avg         avg            no
## 6                     6                  sat   very_high   considering
##   RETENTION
## 1         0
## 2         1
## 3         0
## 4         0
## 5         1
## 6         1
```

```
# Assuming 'bangortelcodata' is your dataframe and 'USAGE_LEVEL' is the column
bangortelcodata$USAGE_LEVEL <- factor(bangortelcodata$USAGE_LEVEL)


head(bangortelcodata)
```

```
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE PLUS_15MINS
## 1 BTLC-007761       0  89318       0        0 162233           266           1
## 2 BTLC-007682       1 142814     187       17 346690           716          24
## 3 BTLC-002228       0  55675       0       32 792662           257           1
## 4 BTLC-011752       1  39559       0        0 416439           165           0
## 5 BTLC-015958       0 145081       0        0 341108           583           0
## 6 BTLC-013969       1 120631      66       17 467811           884           4
##   AVERAGE_CALL_DURATION REPORTED_SATISFACTION USAGE_LEVEL   PLAN_CHANGE
## 1                    12                unsat very_little   considering
## 2                     4                unsat        high   considering
## 3                     1           very_unsat very_little never_thought
## 4                    15             very_sat        high   considering
## 5                     9                  avg         avg            no
## 6                     6                  sat   very_high   considering
##   RETENTION
```

```
## 1         0
## 2         1
## 3         0
## 4         0
## 5         1
## 6         1
```

```
# Assuming 'bangortelcodata' is your dataframe and 'CHANGE OF PLANS' is the column
bangortelcodata$PLAN_CHANGE <- factor(bangortelcodata$PLAN_CHANGE)

head(bangortelcodata)
```

```
##    CUSTOMERID COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE PLUS_15MINS
## 1 BTLC-007761       0  89318       0        0 162233           266           1
## 2 BTLC-007682       1 142814     187       17 346690           716          24
## 3 BTLC-002228       0  55675       0       32 792662           257           1
## 4 BTLC-011752       1  39559       0        0 416439           165           0
## 5 BTLC-015958       0 145081       0        0 341108           583           0
## 6 BTLC-013969       1 120631      66       17 467811           884           4
##   AVERAGE_CALL_DURATION REPORTED_SATISFACTION USAGE_LEVEL   PLAN_CHANGE
## 1                    12                unsat very_little   considering
## 2                     4                unsat        high   considering
## 3                     1           very_unsat very_little never_thought
## 4                    15             very_sat        high   considering
## 5                     9                  avg         avg            no
## 6                     6                  sat   very_high   considering
##   RETENTION
## 1         0
## 2         1
## 3         0
## 4         0
## 5         1
## 6         1
```

Task 1: Decision Tree

Methodology: Based on the 'LEAVE' variable, we build a predictive model for customer churn using the decision tree algorithm. The decision tree categorises consumers into those who are likely to stay or leave based on a set of rules deduced from the dataset. ### Model Performance: The accuracy, precision, and recall of the decision tree model are measured through training and evaluation. The ability of the model to forecast loss of customers is essential for optimizing retention tactics.

```
#install packages and load library
install.packages("rpart")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'rpart' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'rpart'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Windows\AppData\Local\R\win-library\4.3\00LOCK\rpart\libs\x64\rpart.dll
## to C:\Users\Windows\AppData\Local\R\win-library\4.3\rpart\libs\x64\rpart.dll:
## Permission denied

## Warning: restored 'rpart'

##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
install.packages("rpart.plot")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'rpart.plot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
install.packages("caret")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'caret' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'caret'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Windows\AppData\Local\R\win-library\4.3\00LOCK\caret\libs\x64\caret.dll
## to C:\Users\Windows\AppData\Local\R\win-library\4.3\caret\libs\x64\caret.dll:
## Permission denied

## Warning: restored 'caret'

##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
install.packages("glmnet")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'glmnet' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'glmnet'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Windows\AppData\Local\R\win-library\4.3\00LOCK\glmnet\libs\x64\glmnet.dll
## to C:\Users\Windows\AppData\Local\R\win-library\4.3\glmnet\libs\x64\glmnet.dll:
## Permission denied

## Warning: restored 'glmnet'

##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
install.packages("caTools")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'caTools' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'caTools'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Windows\AppData\Local\R\win-library\4.3\00LOCK\caTools\libs\x64\caTools.dll
## to
## C:\Users\Windows\AppData\Local\R\win-library\4.3\caTools\libs\x64\caTools.dll:
## Permission denied

## Warning: restored 'caTools'

##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
#library
library(dplyr)
library(forcats)
library(caret)
```

```
## Loading required package: lattice
```

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
library(caTools)
library(rpart)
library(rpart.plot)
data_tree<- bangortelcodata %>% select (-CUSTOMERID)
set.seed(123)
sample_split<- sample.split(data_tree$RETENTION,SplitRatio =0.7)
#train and test the set for the decision tree
traindata <- subset(data_tree,sample_split == TRUE)
testdata <- subset(data_tree,sample_split == FALSE)
```

```r
#Model for the decision tree
decisiontree_model <- rpart(RETENTION ~ ., data = traindata, method = "class", minbucket = 5, maxdepth=
prediction<-predict(decisiontree_model, traindata, type="class")
confusion_matrix<- table(prediction, traindata$RETENTION)

print(confusion_matrix)
```

```
##
## prediction    0    1
##          0 4524 1544
##          1 2580 5352
```

```r
TP<-confusion_matrix[2,2]
FP<-confusion_matrix[2,1]
TN<-confusion_matrix[1,1]
FN<-confusion_matrix[1,2]
accuracy<- (TP+TN)/sum(confusion_matrix)
precision<- TP/(TP+FP)
recall<- TP/ (TP+FN)
f1_score<- 2 * (precision = recall)/ (precision + recall)
print(paste('Accuracy:', accuracy))
```

```
## [1] "Accuracy: 0.705428571428571"
```

```r
print(paste('Precision:', precision))
```

```
## [1] "Precision: 0.776102088167053"
```

```r
print(paste('Recall:', recall))
```

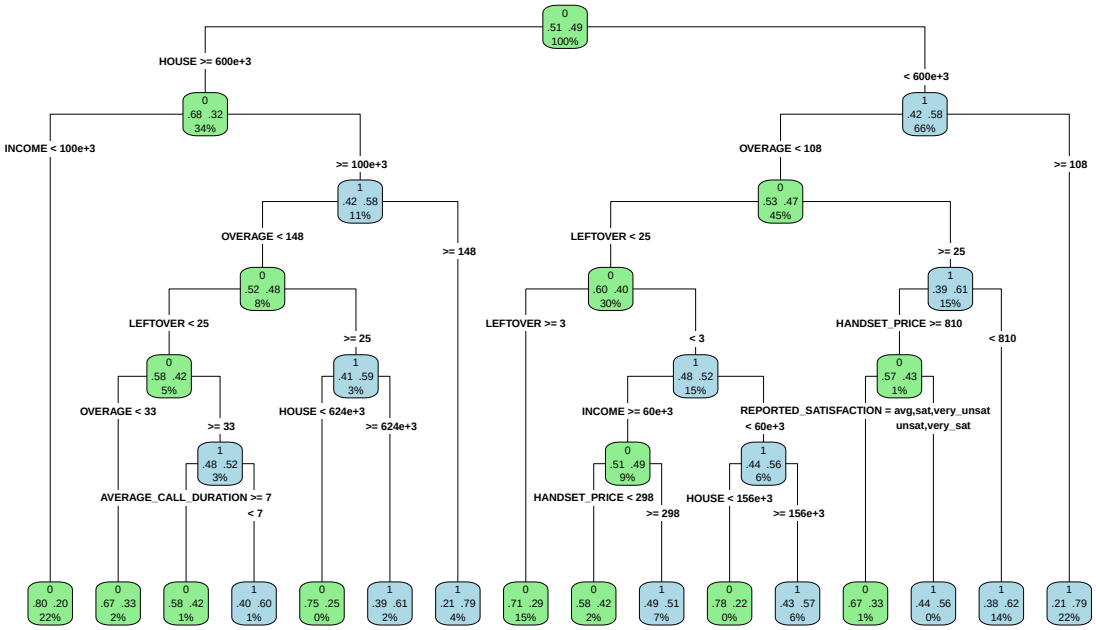
```
## [1] "Recall: 0.776102088167053"
```

```r
print(paste('f1_score:', f1_score))
```

```
## [1] "f1_score: 1"
```

```r
rpart.plot(decisiontree_model, box.palette = c("lightgreen", "lightblue"),
           nn.cex = 0.8, # Adjust the node label size if needed
           fallen.leaves = TRUE, main ="BANGOR TELCO DECISION TREE", extra=104,type=4)
```

BANGOR TELCO DECISION TREE




```
head(data_tree)
```

```
##   COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE PLUS_15MINS
## 1       0  89318       0        0 162233           266          1
## 2       1 142814     187       17 346690           716         24
## 3       0  55675       0       32 792662           257          1
## 4       1  39559       0        0 416439           165          0
## 5       0 145081       0        0 341108           583          0
## 6       1 120631      66       17 467811           884          4
##   AVERAGE_CALL_DURATION REPORTED_SATISFACTION USAGE_LEVEL   PLAN_CHANGE
## 1                    12                 unsat very_little   considering
## 2                     4                 unsat        high   considering
## 3                     1            very_unsat very_little never_thought
## 4                    15              very_sat        high   considering
## 5                     9                   avg         avg            no
## 6                     6                   sat   very_high   considering
##   RETENTION
## 1         0
## 2         1
## 3         0
## 4         0
## 5         1
## 6         1
```

Interpretation:

accuracy: The percentage of accurate predictions made by the model is 70.54%, which is its overall accuracy. Precision: With a precision of 77.61%, the model is accurate 77.61% of the time when it predicts a consumer would depart (class 1). Recall:A recall percentage of 77.61% means that 77.61% of the consumers who genuinely left are being captured by the model. F1_Score: A very high F1 score of 1 could indicate overfitting or a problem with the data. To guarantee model robustness, more research is advised. The costs associated with false positives and false negatives may dictate that we prioritise recall or precision, depending on the particular business situation.

```
# save the model
saveRDS(data_tree, "C:/Users/Windows/Desktop/New folder/MyTreeModel.RDS") # change this to the location
```

In the next step, we will run a summary statistics and check the structure of our data to verify it's fitness to run a logistics regression.

```
#check summary and structure of data_tree
summary(decisiontree_model)
```

```
## Call:
## rpart(formula = RETENTION ~ ., data = traindata, method = "class",
##     minbucket = 5, maxdepth = 6, cp = 0.001)
##   n= 14000
##
##             CP nsplit rel error    xerror        xstd
## 1  0.208236659      0 1.0000000 1.0000000 0.008578055
## 2  0.057134571      1 0.7917633 0.7927784 0.008370746
## 3  0.035527842      3 0.6774942 0.6783643 0.008093258
## 4  0.010440835      4 0.6419664 0.6432715 0.007982782
## 5  0.007758121      5 0.6315255 0.6370360 0.007961836
## 6  0.003335267      7 0.6160093 0.6190545 0.007899153
## 7  0.002755220      8 0.6126740 0.6250000 0.007920257
## 8  0.002610209     10 0.6071636 0.6276102 0.007929403
## 9  0.002175174     11 0.6045534 0.6250000 0.007920257
## 10 0.001885151     12 0.6023782 0.6239849 0.007916680
## 11 0.001450116     13 0.6004930 0.6219548 0.007909495
## 12 0.001015081     14 0.5990429 0.6236949 0.007915656
## 13 0.001000000     15 0.5980278 0.6229698 0.007913093
##
## Variable importance
##               OVERAGE                 HOUSE           PLUS_15MINS
##                    21                    18                    17
##                INCOME         HANDSET_PRICE              LEFTOVER
##                    14                    11                    10
## AVERAGE_CALL_DURATION
##                     9
##
## Node number 1: 14000 observations,    complexity param=0.2082367
##   predicted class=0  expected loss=0.4925714  P(node) =1
##     class counts:  7104  6896
##    probabilities: 0.507 0.493
##   left son=2 (4690 obs) right son=3 (9310 obs)
##   Primary splits:
```

```
##       HOUSE         < 600479.5 to the right, improve=397.33930, (0 missing)
##       OVERAGE       < 87.5     to the left,  improve=352.53680, (0 missing)
##       PLUS_15MINS   < 7.5      to the left,  improve=286.03800, (0 missing)
##       INCOME        < 100072   to the left,  improve= 90.52039, (0 missing)
##       HANDSET_PRICE < 397.5    to the left,  improve= 67.52063, (0 missing)
##   Surrogate splits:
##       HANDSET_PRICE < 897.5    to the right, agree=0.665, adj=0, (0 split)
##
## Node number 2: 4690 observations,    complexity param=0.03552784
##   predicted class=0  expected loss=0.3247335  P(node) =0.335
##     class counts:  3167  1523
##    probabilities: 0.675 0.325
##   left son=4 (3127 obs) right son=5 (1563 obs)
##   Primary splits:
##       INCOME        < 100357.5 to the left,  improve=301.629900, (0 missing)
##       HANDSET_PRICE < 401.5    to the left,  improve=215.389400, (0 missing)
##       OVERAGE       < 68.5     to the left,  improve= 26.187890, (0 missing)
##       PLUS_15MINS   < 7.5      to the left,  improve= 17.771760, (0 missing)
##       LEFTOVER      < 21.5     to the left,  improve=  6.836182, (0 missing)
##   Surrogate splits:
##       HANDSET_PRICE < 400      to the left,  agree=0.935, adj=0.806, (0 split)
##       HOUSE         < 600742.5 to the right, agree=0.667, adj=0.001, (0 split)
##
## Node number 3: 9310 observations,    complexity param=0.05713457
##   predicted class=1  expected loss=0.4228786  P(node) =0.665
##     class counts:  3937  5373
##    probabilities: 0.423 0.577
##   left son=6 (6231 obs) right son=7 (3079 obs)
##   Primary splits:
##       OVERAGE               < 107.5    to the left,  improve=405.075000, (0 missing)
##       PLUS_15MINS           < 7.5      to the left,  improve=324.849500, (0 missing)
##       LEFTOVER              < 24.5     to the left,  improve= 69.829810, (0 missing)
##       AVERAGE_CALL_DURATION < 3        to the right, improve= 46.272770, (0 missing)
##       HOUSE                 < 313084.5 to the left,  improve=  3.766729, (0 missing)
##   Surrogate splits:
##       PLUS_15MINS < 7.5      to the left,  agree=0.937, adj=0.811, (0 split)
##       INCOME      < 20056.5  to the right, agree=0.670, adj=0.002, (0 split)
##       HOUSE       < 150024   to the right, agree=0.669, adj=0.001, (0 split)
##
## Node number 4: 3127 observations
##   predicted class=0  expected loss=0.1979533  P(node) =0.2233571
##     class counts:  2508   619
##    probabilities: 0.802 0.198
##
## Node number 5: 1563 observations,    complexity param=0.007758121
##   predicted class=1  expected loss=0.4216251  P(node) =0.1116429
##     class counts:   659   904
##    probabilities: 0.422 0.578
##   left son=10 (1059 obs) right son=11 (504 obs)
##   Primary splits:
##       OVERAGE               < 147.5    to the left,  improve=65.186770, (0 missing)
##       PLUS_15MINS           < 7.5      to the left,  improve=48.413090, (0 missing)
##       LEFTOVER              < 22.5     to the left,  improve=11.372140, (0 missing)
##       AVERAGE_CALL_DURATION < 3        to the right, improve= 5.297969, (0 missing)
```

```
##       INCOME               < 152831.5 to the right, improve= 3.705814, (0 missing)
##   Surrogate splits:
##       PLUS_15MINS < 7.5      to the left,  agree=0.930, adj=0.784, (0 split)
##       INCOME      < 159711.5 to the left,  agree=0.679, adj=0.006, (0 split)
##       HOUSE       < 997737   to the left,  agree=0.678, adj=0.002, (0 split)
##
## Node number 6: 6231 observations,    complexity param=0.05713457
##   predicted class=0  expected loss=0.4734393  P(node) =0.4450714
##     class counts:  3281  2950
##    probabilities: 0.527 0.473
##   left son=12 (4142 obs) right son=13 (2089 obs)
##   Primary splits:
##       LEFTOVER              < 24.5     to the left,  improve=116.153200, (0 missing)
##       AVERAGE_CALL_DURATION < 3        to the right, improve= 80.206480, (0 missing)
##       OVERAGE               < 1.5      to the left,  improve= 53.018340, (0 missing)
##       PLUS_15MINS           < 2        to the left,  improve= 38.019760, (0 missing)
##       INCOME                < 48102.5  to the right, improve=  7.595966, (0 missing)
##   Surrogate splits:
##       AVERAGE_CALL_DURATION < 3        to the right, agree=0.934, adj=0.802, (0 split)
##       INCOME                < 20091.5  to the right, agree=0.665, adj=0.001, (0 split)
##       HOUSE                 < 598112   to the left,  agree=0.665, adj=0.001, (0 split)
##
## Node number 7: 3079 observations
##   predicted class=1  expected loss=0.2130562  P(node) =0.2199286
##     class counts:   656  2423
##    probabilities: 0.213 0.787
##
## Node number 10: 1059 observations,    complexity param=0.007758121
##   predicted class=0  expected loss=0.4787535  P(node) =0.07564286
##     class counts:   552   507
##    probabilities: 0.521 0.479
##   left son=20 (699 obs) right son=21 (360 obs)
##   Primary splits:
##       LEFTOVER              < 24.5     to the left,  improve=12.572350, (0 missing)
##       AVERAGE_CALL_DURATION < 3        to the right, improve= 7.657377, (0 missing)
##       HOUSE                 < 963577   to the right, improve= 7.567501, (0 missing)
##       OVERAGE               < 32.5     to the left,  improve= 6.759196, (0 missing)
##       INCOME                < 153233   to the right, improve= 4.958613, (0 missing)
##   Surrogate splits:
##       AVERAGE_CALL_DURATION < 3        to the right, agree=0.935, adj=0.808, (0 split)
##       HANDSET_PRICE         < 179      to the right, agree=0.663, adj=0.008, (0 split)
##       HOUSE                 < 998923   to the left,  agree=0.661, adj=0.003, (0 split)
##
## Node number 11: 504 observations
##   predicted class=1  expected loss=0.2123016  P(node) =0.036
##     class counts:   107   397
##    probabilities: 0.212 0.788
##
## Node number 12: 4142 observations,    complexity param=0.01044084
##   predicted class=0  expected loss=0.4048769  P(node) =0.2958571
##     class counts:  2465  1677
##    probabilities: 0.595 0.405
##   left son=24 (2084 obs) right son=25 (2058 obs)
##   Primary splits:
```

```
##       LEFTOVER              < 2.5      to the right, improve=103.74970, (0 missing)
##       AVERAGE_CALL_DURATION < 7        to the left,  improve= 72.61678, (0 missing)
##       OVERAGE               < 15.5     to the left,  improve= 62.95222, (0 missing)
##       PLUS_15MINS           < 2        to the left,  improve= 46.46714, (0 missing)
##       INCOME                < 49950    to the right, improve= 10.65840, (0 missing)
##   Surrogate splits:
##       AVERAGE_CALL_DURATION < 7        to the left,  agree=0.927, adj=0.852, (0 split)
##       INCOME                < 41404    to the right, agree=0.522, adj=0.038, (0 split)
##       HANDSET_PRICE         < 326.5    to the right, agree=0.515, adj=0.023, (0 split)
##       PLUS_15MINS           < 2        to the left,  agree=0.514, adj=0.022, (0 split)
##       HOUSE                 < 209860.5 to the right, agree=0.512, adj=0.017, (0 split)
##
## Node number 13: 2089 observations,    complexity param=0.002610209
##   predicted class=1  expected loss=0.3906175  P(node) =0.1492143
##     class counts:   816  1273
##    probabilities: 0.391 0.609
##   left son=26 (130 obs) right son=27 (1959 obs)
##   Primary splits:
##       HANDSET_PRICE < 809.5    to the right, improve=8.845140, (0 missing)
##       HOUSE         < 186285.5 to the left,  improve=3.927207, (0 missing)
##       LEFTOVER      < 82.5     to the right, improve=3.010959, (0 missing)
##       INCOME        < 149389.5 to the left,  improve=2.839275, (0 missing)
##       OVERAGE       < 47.5     to the left,  improve=2.762015, (0 missing)
##
## Node number 20: 699 observations,    complexity param=0.00275522
##   predicted class=0  expected loss=0.4234621  P(node) =0.04992857
##     class counts:   403   296
##    probabilities: 0.577 0.423
##   left son=40 (343 obs) right son=41 (356 obs)
##   Primary splits:
##       OVERAGE     < 32.5     to the left,  improve=12.655550, (0 missing)
##       PLUS_15MINS < 2        to the left,  improve= 8.722317, (0 missing)
##       HOUSE       < 960996.5 to the right, improve= 5.321848, (0 missing)
##       LEFTOVER    < 8.5      to the right, improve= 3.853225, (0 missing)
##       INCOME      < 151823.5 to the right, improve= 3.841633, (0 missing)
##   Surrogate splits:
##       PLUS_15MINS           < 2        to the left,  agree=0.930, adj=0.857, (0 split)
##       INCOME                < 147755   to the right, agree=0.535, adj=0.052, (0 split)
##       HOUSE                 < 931613   to the right, agree=0.535, adj=0.052, (0 split)
##       HANDSET_PRICE         < 509.5    to the left,  agree=0.535, adj=0.052, (0 split)
##       REPORTED_SATISFACTION splits as  RLRLL,        agree=0.529, adj=0.041, (0 split)
##
## Node number 21: 360 observations,    complexity param=0.001450116
##   predicted class=1  expected loss=0.4138889  P(node) =0.02571429
##     class counts:   149   211
##    probabilities: 0.414 0.586
##   left son=42 (20 obs) right son=43 (340 obs)
##   Primary splits:
##       HOUSE                 < 624143   to the left,  improve=4.784641, (0 missing)
##       INCOME                < 119059   to the right, improve=4.306391, (0 missing)
##       OVERAGE               < 82       to the right, improve=3.405957, (0 missing)
##       HANDSET_PRICE         < 578      to the right, improve=2.311760, (0 missing)
##       REPORTED_SATISFACTION splits as  LRRRR,        improve=1.393124, (0 missing)
##
```

```
## Node number 24: 2084 observations
##   predicted class=0  expected loss=0.293666  P(node) =0.1488571
##     class counts:  1472   612
##    probabilities: 0.706 0.294
##
## Node number 25: 2058 observations,    complexity param=0.003335267
##   predicted class=1  expected loss=0.4825073  P(node) =0.147
##     class counts:   993  1065
##    probabilities: 0.483 0.517
##   left son=50 (1221 obs) right son=51 (837 obs)
##   Primary splits:
##       INCOME                < 59923.5  to the right, improve=4.348429, (0 missing)
##       HOUSE                 < 155469   to the left,  improve=4.025778, (0 missing)
##       REPORTED_SATISFACTION splits as  RLRRR,        improve=2.268146, (0 missing)
##       PLUS_15MINS           < 7.5      to the left,  improve=2.208880, (0 missing)
##       COLLEGE               < 0.5      to the left,  improve=2.065778, (0 missing)
##   Surrogate splits:
##       HANDSET_PRICE < 249.5    to the right, agree=0.878, adj=0.699, (0 split)
##       HOUSE         < 153820   to the right, agree=0.599, adj=0.013, (0 split)
##       OVERAGE       < 94.5     to the left,  agree=0.596, adj=0.006, (0 split)
##
## Node number 26: 130 observations,    complexity param=0.001015081
##   predicted class=0  expected loss=0.4307692  P(node) =0.009285714
##     class counts:    74    56
##    probabilities: 0.569 0.431
##   left son=52 (75 obs) right son=53 (55 obs)
##   Primary splits:
##       REPORTED_SATISFACTION splits as  LLRRL,        improve=3.365967, (0 missing)
##       PLUS_15MINS           < 11.5     to the left,  improve=2.328685, (0 missing)
##       INCOME                < 152559   to the left,  improve=2.112899, (0 missing)
##       HANDSET_PRICE         < 893.5    to the left,  improve=1.737453, (0 missing)
##       PLAN_CHANGE           splits as  RLRLL,        improve=1.694322, (0 missing)
##   Surrogate splits:
##       HANDSET_PRICE < 881      to the left,  agree=0.631, adj=0.127, (0 split)
##       PLAN_CHANGE   splits as  LLRRR,        agree=0.615, adj=0.091, (0 split)
##       INCOME        < 152240   to the left,  agree=0.608, adj=0.073, (0 split)
##       LEFTOVER      < 84.5     to the left,  agree=0.585, adj=0.018, (0 split)
##       HOUSE         < 160428   to the right, agree=0.585, adj=0.018, (0 split)
##
## Node number 27: 1959 observations
##   predicted class=1  expected loss=0.3787647  P(node) =0.1399286
##     class counts:   742  1217
##    probabilities: 0.379 0.621
##
## Node number 40: 343 observations
##   predicted class=0  expected loss=0.3265306  P(node) =0.0245
##     class counts:   231   112
##    probabilities: 0.673 0.327
##
## Node number 41: 356 observations,    complexity param=0.00275522
##   predicted class=1  expected loss=0.4831461  P(node) =0.02542857
##     class counts:   172   184
##    probabilities: 0.483 0.517
##   left son=82 (164 obs) right son=83 (192 obs)
```

```
##   Primary splits:
##       AVERAGE_CALL_DURATION < 7        to the right, improve=5.619145, (0 missing)
##       LEFTOVER              < 2.5      to the left,  improve=4.713891, (0 missing)
##       HOUSE                 < 970142   to the right, improve=3.932756, (0 missing)
##       INCOME                < 157268.5 to the right, improve=1.960412, (0 missing)
##       PLAN_CHANGE           splits as  LRLRR,        improve=1.827506, (0 missing)
##   Surrogate splits:
##       LEFTOVER      < 2.5      to the left,  agree=0.919, adj=0.823, (0 split)
##       HANDSET_PRICE < 355.5    to the left,  agree=0.567, adj=0.061, (0 split)
##       COLLEGE       < 0.5      to the left,  agree=0.565, adj=0.055, (0 split)
##       PLAN_CHANGE   splits as  RRLLR,        agree=0.562, adj=0.049, (0 split)
##       OVERAGE       < 124.5    to the right, agree=0.559, adj=0.043, (0 split)
##
## Node number 42: 20 observations
##   predicted class=0  expected loss=0.25  P(node) =0.001428571
##     class counts:    15     5
##    probabilities: 0.750 0.250
##
## Node number 43: 340 observations
##   predicted class=1  expected loss=0.3941176  P(node) =0.02428571
##     class counts:   134   206
##    probabilities: 0.394 0.606
##
## Node number 50: 1221 observations,    complexity param=0.002175174
##   predicted class=0  expected loss=0.4905815  P(node) =0.08721429
##     class counts:   622   599
##    probabilities: 0.509 0.491
##   left son=100 (232 obs) right son=101 (989 obs)
##   Primary splits:
##       HANDSET_PRICE         < 297.5    to the left,  improve=3.009195, (0 missing)
##       PLUS_15MINS           < 7.5      to the left,  improve=2.440678, (0 missing)
##       HOUSE                 < 249216   to the right, improve=1.849889, (0 missing)
##       REPORTED_SATISFACTION splits as  RLRRR,       improve=1.725748, (0 missing)
##       INCOME                < 60586.5  to the left,  improve=1.702532, (0 missing)
##   Surrogate splits:
##       INCOME < 61486.5  to the left,  agree=0.811, adj=0.004, (0 split)
##
## Node number 51: 837 observations,    complexity param=0.001885151
##   predicted class=1  expected loss=0.4432497  P(node) =0.05978571
##     class counts:   371   466
##    probabilities: 0.443 0.557
##   left son=102 (23 obs) right son=103 (814 obs)
##   Primary splits:
##       HOUSE         < 156489   to the left,  improve=5.447254, (0 missing)
##       COLLEGE       < 0.5      to the left,  improve=3.269684, (0 missing)
##       HANDSET_PRICE < 199.5    to the left,  improve=1.399755, (0 missing)
##       INCOME        < 56828    to the left,  improve=1.215027, (0 missing)
##       USAGE_LEVEL   splits as  LRRLL,       improve=1.213971, (0 missing)
##
## Node number 52: 75 observations
##   predicted class=0  expected loss=0.3333333  P(node) =0.005357143
##     class counts:    50    25
##    probabilities: 0.667 0.333
##
```

```
## Node number 53: 55 observations
##   predicted class=1  expected loss=0.4363636  P(node) =0.003928571
##     class counts:    24    31
##    probabilities: 0.436 0.564
##
## Node number 82: 164 observations
##   predicted class=0  expected loss=0.4207317  P(node) =0.01171429
##     class counts:    95    69
##    probabilities: 0.579 0.421
##
## Node number 83: 192 observations
##   predicted class=1  expected loss=0.4010417  P(node) =0.01371429
##     class counts:    77   115
##    probabilities: 0.401 0.599
##
## Node number 100: 232 observations
##   predicted class=0  expected loss=0.4181034  P(node) =0.01657143
##     class counts:   135    97
##    probabilities: 0.582 0.418
##
## Node number 101: 989 observations
##   predicted class=1  expected loss=0.4924166  P(node) =0.07064286
##     class counts:   487   502
##    probabilities: 0.492 0.508
##
## Node number 102: 23 observations
##   predicted class=0  expected loss=0.2173913  P(node) =0.001642857
##     class counts:    18     5
##    probabilities: 0.783 0.217
##
## Node number 103: 814 observations
##   predicted class=1  expected loss=0.4336609  P(node) =0.05814286
##     class counts:   353   461
##    probabilities: 0.434 0.566
```

```
str(data_tree)
```

```
## 'data.frame':    20000 obs. of  12 variables:
##  $ COLLEGE              : num  0 1 0 1 0 1 1 0 0 0 ...
##  $ INCOME               : int  89318 142814 55675 39559 145081 120631 59162 117488 82304 46786 ...
##  $ OVERAGE              : int  0 187 0 0 0 66 0 53 170 44 ...
##  $ LEFTOVER             : int  0 17 32 0 0 17 55 12 34 0 ...
##  $ HOUSE                : int  162233 346690 792662 416439 341108 467811 251345 810740 517128 964756
##  $ HANDSET_PRICE        : int  266 716 257 165 583 884 396 205 369 193 ...
##  $ PLUS_15MINS          : int  1 24 1 0 0 4 1 4 26 5 ...
##  $ AVERAGE_CALL_DURATION: int  12 4 1 15 9 6 1 4 2 8 ...
##  $ REPORTED_SATISFACTION: Factor w/ 5 levels "avg","sat","unsat",..: 3 3 5 4 1 2 4 5 5 5 ...
##  $ USAGE_LEVEL          : Factor w/ 5 levels "avg","high","little",..: 5 2 5 2 1 4 4 2 3 3 ...
##  $ PLAN_CHANGE          : Factor w/ 5 levels "actively_looking_into_it",..: 2 2 3 2 4 2 2 2 1 2 ...
##  $ RETENTION            : num  0 1 0 0 1 1 1 0 1 0 ...
```

TASK 2: LOGISTIC REGRESSION

Introduction:

The goal of Task 2 is to estimate the probability that a customer will leave BangorTelco by using logistic regression as a predictive modelling method. Using different input features to forecast the likelihood of customer churn, logistic regression provides a more straightforward method, building on the understanding obtained from the decision tree analysis in Task 1.

Logistic regression models

Model development: Using the supplied dataset, which includes details on 20,000 BangorTelco customers, we will build a logistic regression model. The goal of the model is to identify trends in the characteristics of customers that influence their propensity to leave the business. In contrast to decision trees, logistic regression computes the likelihood of an occurrence directly, which makes it an effective tool for binary classification issues like churn prediction.

```
#Run a logistic regression using the retention column as the dependent variable and others as the indep
bangortelcoLog <- glm(formula = RETENTION ~  INCOME + OVERAGE + LEFTOVER + HOUSE + HANDSET_PRICE + PLUS_
               data = data_tree,                                    #  the model on the iris data
               family = binomial)                                 # binomial means we will be using logist


#Inspect the model:
bangortelcoLog
```

```
##
## Call:  glm(formula = RETENTION ~ INCOME + OVERAGE + LEFTOVER + HOUSE +
##     HANDSET_PRICE + PLUS_15MINS + AVERAGE_CALL_DURATION + REPORTED_SATISFACTION +
##     USAGE_LEVEL + PLAN_CHANGE, family = binomial, data = data_tree)
##
## Coefficients:
##                      (Intercept)                            INCOME
##                        -5.306e-01                         3.430e-06
##                          OVERAGE                          LEFTOVER
##                         4.968e-03                         8.328e-03
##                             HOUSE                     HANDSET_PRICE
##                        -1.873e-06                         4.378e-04
##                       PLUS_15MINS             AVERAGE_CALL_DURATION
##                         1.381e-02                         2.801e-02
##        REPORTED_SATISFACTIONsat       REPORTED_SATISFACTIONunsat
##                        -1.050e-01                         7.997e-02
##   REPORTED_SATISFACTIONvery_sat  REPORTED_SATISFACTIONvery_unsat
##                         4.743e-02                         7.142e-02
##                 USAGE_LEVELhigh                USAGE_LEVELlittle
##                        -2.682e-02                        -1.923e-02
##            USAGE_LEVELvery_high           USAGE_LEVELvery_little
##                         2.683e-02                         2.086e-02
##          PLAN_CHANGEconsidering         PLAN_CHANGEnever_thought
##                         8.005e-03                         1.625e-02
##                    PLAN_CHANGEno               PLAN_CHANGEperhaps
##                         6.135e-02                         8.959e-02
```

```
##
## Degrees of Freedom: 19999 Total (i.e. Null);  19980 Residual
## Null Deviance:       27720
## Residual Deviance: 25250     AIC: 25290
```

```
summary(bangortelcoLog)
```

```
##
## Call:
## glm(formula = RETENTION ~ INCOME + OVERAGE + LEFTOVER + HOUSE +
##     HANDSET_PRICE + PLUS_15MINS + AVERAGE_CALL_DURATION + REPORTED_SATISFACTION +
##     USAGE_LEVEL + PLAN_CHANGE, family = binomial, data = data_tree)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -5.306e-01  1.048e-01  -5.061 4.17e-07 ***
## INCOME                         3.430e-06  5.262e-07   6.518 7.14e-11 ***
## OVERAGE                        4.968e-03  2.767e-04  17.957  < 2e-16 ***
## LEFTOVER                       8.328e-03  7.539e-04  11.048  < 2e-16 ***
## HOUSE                         -1.873e-06  6.142e-08 -30.493  < 2e-16 ***
## HANDSET_PRICE                  4.378e-04  1.026e-04   4.267 1.99e-05 ***
## PLUS_15MINS                    1.381e-02  2.656e-03   5.199 2.00e-07 ***
## AVERAGE_CALL_DURATION          2.801e-02  4.571e-03   6.128 8.92e-10 ***
## REPORTED_SATISFACTIONsat      -1.050e-01  8.202e-02  -1.280    0.201
## REPORTED_SATISFACTIONunsat     7.997e-02  5.819e-02   1.374    0.169
## REPORTED_SATISFACTIONvery_sat  4.743e-02  5.613e-02   0.845    0.398
## REPORTED_SATISFACTIONvery_unsat 7.142e-02  5.319e-02   1.343    0.179
## USAGE_LEVELhigh               -2.682e-02  8.269e-02  -0.324    0.746
## USAGE_LEVELlittle             -1.923e-02  7.170e-02  -0.268    0.789
## USAGE_LEVELvery_high           2.683e-02  7.386e-02   0.363    0.716
## USAGE_LEVELvery_little         2.086e-02  7.547e-02   0.276    0.782
## PLAN_CHANGEconsidering         8.005e-03  3.845e-02   0.208    0.835
## PLAN_CHANGEnever_thought       1.625e-02  5.645e-02   0.288    0.773
## PLAN_CHANGEno                  6.135e-02  4.503e-02   1.363    0.173
## PLAN_CHANGEperhaps             8.959e-02  7.220e-02   1.241    0.215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27722  on 19999  degrees of freedom
## Residual deviance: 25252  on 19980  degrees of freedom
## AIC: 25292
##
## Number of Fisher Scoring iterations: 4
```

Logistic regression Interpretations:

Intercept: -0.5306

when all prediction variable is zero, the log odd of the response variable is the intercept. INCOME: 3.43e-06

A 3.43e-06 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable INCOME. OVERAGE: 0.00497

A 0.00497 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable. LEFTOVER: 0.00833

A 0.00833 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable LEFTOVER. HOUSE: -1.87e-06

A -1.87e-06 decrease in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable HOUSE. HANDSET_PRICE: 0.000438

A 0.000438 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable HANDSET_PRICE. PLUS_15MINS: 0.01381

A 0.01381 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable PLUS_15MINS. AVERAGE_CALL_DURATION: 0.02801

A 0.02801 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable AVERAGE_CALL_DURATION. REPORTED_SATISFACTION (sat): -0.105

A -0.105 decrease in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable REPORTED_SATISFACTACTION associated to satisfaction. REPORTED_SATISFACTION (unsat): 0.07997

A 0.07997 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable REPORTED_SATISFACTACTION associated to unsatisfaction. REPORTED_SATISFACTION (very_sat): 0.04743

A 0.04743 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable REPORTED_SATISFACTACTION associated to very-satisfied. REPORTED_SATISFACTION (very_unsat): 0.07142

A 0.07142 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable REPORTED_SATISFACTACTION associated to very-unsatisfied. USAGE_LEVEL (high): -0.02682

A -0.02682 decrease in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable USAGE_LEVEL associated to high. USAGE_LEVEL (little): -0.01923

A -0.01923 decrease in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable USAGE_LEVEL associated to little. USAGE_LEVEL (very_high): 0.02683

A 0.02683 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable USAGE_LEVEL associated to very_high. USAGE_LEVEL (very_little): 0.02086

A 0.02086 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable USAGE_LEVEL associated to very_little. PLAN_CHANGE (considering): 0.008005

A 0.008005 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable PLAN_CHANGE associated to considering. PLAN_CHANGE (never_thought): 0.01625

A 0.01625 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable PLAN_CHANGE associated to never_thought. PLAN_CHANGE (no): 0.06135

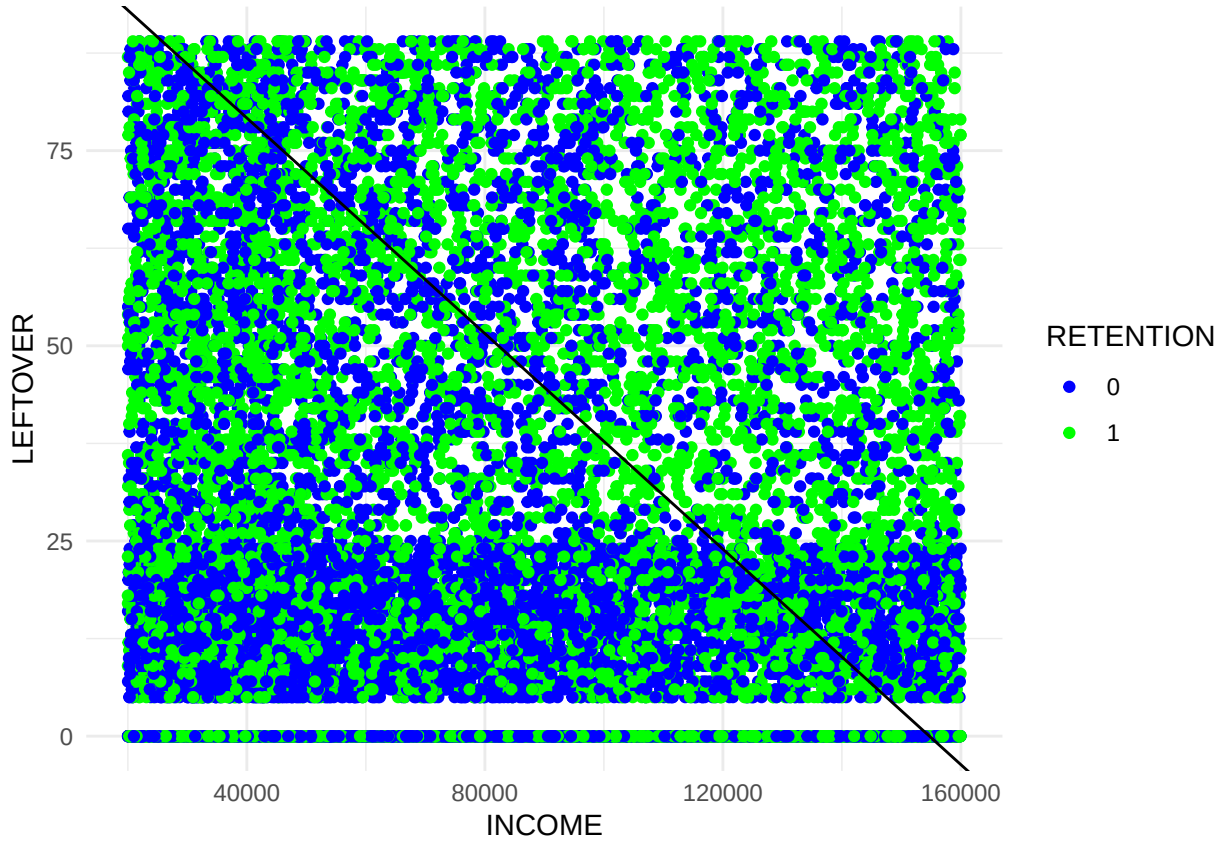
A 0.01635 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable PLAN_CHANGE associated to no. PLAN_CHANGE (perhaps): 0.08959

A 0.08959 increase in the coefficient of the dependent variable 'RETENTION',is associated with a one-unit increase of the independent variable PLAN_CHANGE associated to perhaps.

```
head(data_tree)
```

```
##   COLLEGE INCOME OVERAGE LEFTOVER  HOUSE HANDSET_PRICE PLUS_15MINS
## 1       0  89318       0        0 162233           266          1
## 2       1 142814     187       17 346690           716         24
## 3       0  55675       0       32 792662           257          1
## 4       1  39559       0        0 416439           165          0
## 5       0 145081       0        0 341108           583          0
## 6       1 120631      66       17 467811           884          4
##   AVERAGE_CALL_DURATION REPORTED_SATISFACTION USAGE_LEVEL   PLAN_CHANGE
## 1                    12                unsat very_little   considering
## 2                     4                unsat        high   considering
## 3                     1           very_unsat very_little never_thought
## 4                    15             very_sat        high   considering
## 5                     9                  avg         avg            no
## 6                     6                  sat   very_high   considering
##   RETENTION
## 1         0
## 2         1
## 3         0
## 4         0
## 5         1
## 6         1
```

```
library(ggplot2)


# Calculate the Slope and Intercept
Slope <- -coef(bangortelcoLog)[2] / coef(bangortelcoLog)[3]
Intercept <- -coef(bangortelcoLog)[1] / coef(bangortelcoLog)[3]

# Plotting using ggplot
ggplot(data =data_tree, aes(x = INCOME, y = LEFTOVER, color = factor(RETENTION))) +
  geom_point() +
  geom_abline(slope = Slope, intercept = Intercept) +
  labs(x = "INCOME", y = "LEFTOVER", color = "RETENTION") +  # Label axes and legend
  scale_color_manual(values = c("blue", "green")) +  # Adjust colors for RETENTION
  theme_minimal()
```




```
install.packages("corrplot")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'corrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
library(corrplot)
```

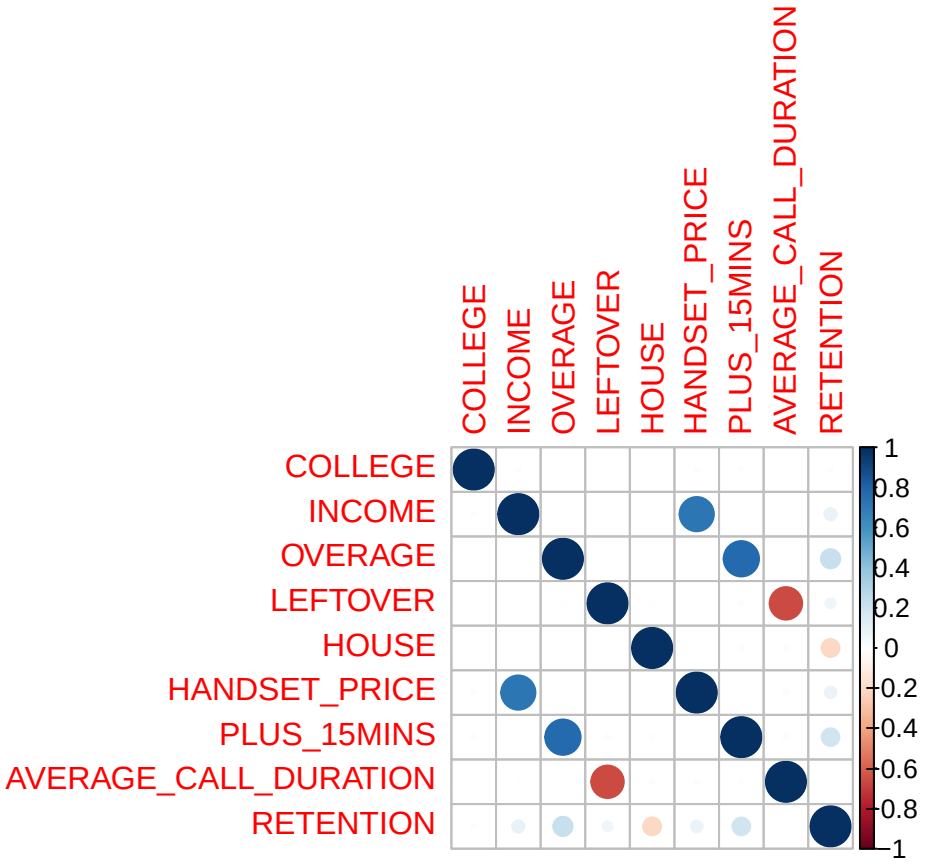
```
## corrplot 0.92 loaded
```

```
# Select only numeric columns from traindata
numeric_data <- traindata[sapply(traindata, is.numeric)]

# Compute correlation matrix
Correlations <- cor(numeric_data)
Correlations
```

```
##                            COLLEGE       INCOME      OVERAGE     LEFTOVER
## COLLEGE                1.0000000000  0.013257736 -0.007428526 -0.002041481
## INCOME                 0.0132577360  1.000000000 -0.001241853  0.002394858
```

```
## OVERAGE               -0.0074285255 -0.001241853  1.000000000 -0.007421243
## LEFTOVER              -0.0020414814  0.002394858 -0.007421243  1.000000000
## HOUSE                  0.0044357963 -0.007723535  0.008961009  0.009864740
## HANDSET_PRICE          0.0120964735  0.727641561 -0.003098744  0.005474413
## PLUS_15MINS           -0.0134656747  0.006902055  0.774671493 -0.012665922
## AVERAGE_CALL_DURATION -0.0002543916 -0.009779928  0.005877415 -0.653884100
## RETENTION              0.0130290524  0.091631411  0.229565092  0.061548652
##                              HOUSE HANDSET_PRICE  PLUS_15MINS
## COLLEGE                0.004435796   0.012096473 -0.013465675
## INCOME                -0.007723535   0.727641561  0.006902055
## OVERAGE                0.008961009  -0.003098744  0.774671493
## LEFTOVER               0.009864740   0.005474413 -0.012665922
## HOUSE                  1.000000000  -0.002380118  0.011408162
## HANDSET_PRICE         -0.002380118   1.000000000  0.004871106
## PLUS_15MINS            0.011408162   0.004871106  1.000000000
## AVERAGE_CALL_DURATION -0.012241965  -0.014319693  0.014950833
## RETENTION             -0.200496448   0.082558599  0.194814956
##                       AVERAGE_CALL_DURATION    RETENTION
## COLLEGE                       -0.0002543916  0.013029052
## INCOME                        -0.0097799276  0.091631411
## OVERAGE                        0.0058774154  0.229565092
## LEFTOVER                      -0.6538841001  0.061548652
## HOUSE                         -0.0122419651 -0.200496448
## HANDSET_PRICE                 -0.0143196933  0.082558599
## PLUS_15MINS                    0.0149508332  0.194814956
## AVERAGE_CALL_DURATION          1.0000000000 -0.008065079
## RETENTION                     -0.0080650789  1.000000000
```

```r
# Visual representation of correlation matrix
corrplot(Correlations)
```



TASK 3: K - NEAREST NEIGHBOUR

```
#load library
library(caret)
knn_data<-data_tree
knn_data<- knn_data %>% mutate( RETENTION = factor(RETENTION))
#scale data
knn_data[,2:8]<-scale(knn_data[,2:8])

#we would split the data into train and test split
set.seed(123)
intrain<-createDataPartition(knn_data$RETENTION, p = 0.70, list = FALSE)
train_data<-knn_data[intrain,]
test_data<-knn_data[-intrain,]

?knn
```

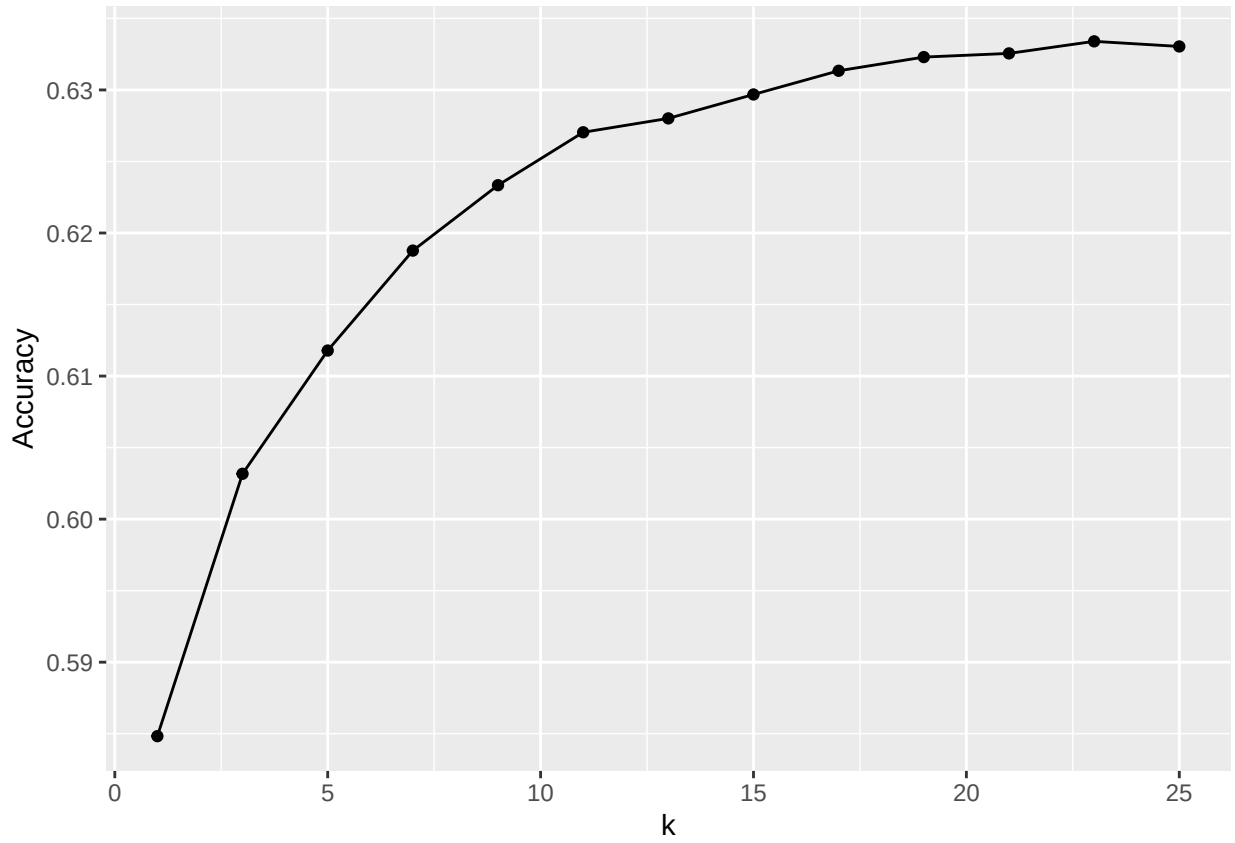
```
## starting httpd help server ... done
```

```
Grid_values<- expand.grid(k=seq(1 , 25, by =2))

knn_reg_fit<- train(RETENTION~.,data = train_data, method = 'knn',
      preProcess= c('center', 'scale'),
      trControl= trainControl(method = 'repeatedcv',number =10, repeats = 5), tuneGrid = Grid_values)
knn_reg_fit
```

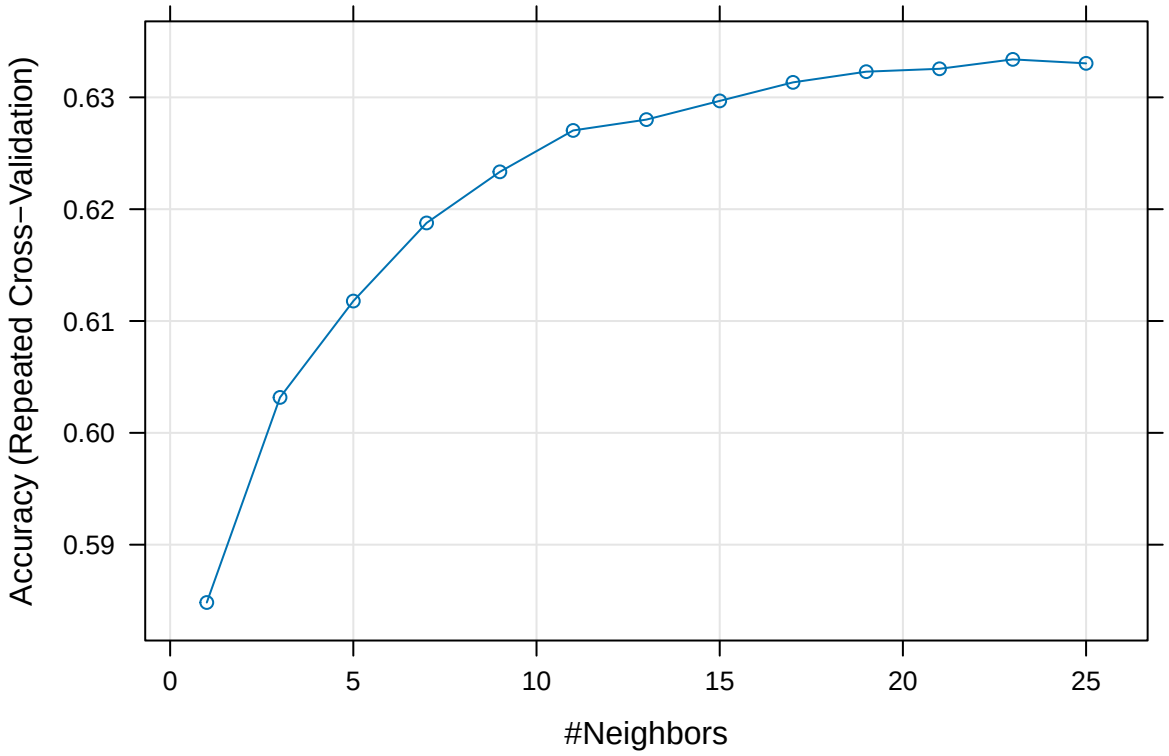
```
## k-Nearest Neighbors
```

```
##
## 14001 samples
##    11 predictor
##     2 classes: '0', '1'
##
## Pre-processing: centered (20), scaled (20)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 12601, 12601, 12602, 12600, 12601, 12601, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.5848298  0.1694064
##    3  0.6031692  0.2059566
##    5  0.6117838  0.2230519
##    7  0.6187696  0.2368267
##    9  0.6233401  0.2457627
##   11  0.6270399  0.2530067
##   13  0.6280111  0.2547766
##   15  0.6296833  0.2580179
##   17  0.6313399  0.2612146
##   19  0.6322971  0.2630536
##   21  0.6325537  0.2635332
##   23  0.6333964  0.2651528
##   25  0.6330392  0.2643848
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 23.
```

```
#plot model
results <- knn_reg_fit$results
results |> ggplot(aes(x = k, y = Accuracy)) + geom_point() + geom_line()
```




```
plot(knn_reg_fit)
```




```
confusionMatrix(knn_reg_fit)
```

```
## Cross-Validated (10 fold, repeated 5 times) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##            Reference
## Prediction    0    1
##          0 35.8 21.7
##          1 14.9 27.5
##
##  Accuracy (average) : 0.6334
```

INTERPRETATION ON KNN MODEL

Accuracy: 0.6317 is the stated overall accuracy (average across resamples). Interpreting the Accuracy With an accuracy of 0.6317, the model correctly classified 63.17% of cases on average. But accuracy by itself could not give the whole narrative.Taking into account precision, recall, and F1-score might help you gain a better understanding of the model's performance.

```
# Make predictions on test data
TestPred <- predict(knn_reg_fit, newdata = test_data)

# Generate confusion matrix
```

```r
conf_mat_knn <- confusionMatrix(data = TestPred, reference = test_data$RETENTION)

# Extract the metrics
knn_accuracy <- conf_mat_knn$overall['Accuracy']
knn_precision <- conf_mat_knn$byClass['Precision']
knn_recall <- conf_mat_knn$byClass['Recall']
knn_F1_score <- conf_mat_knn$byClass['F1']

# lets display metrics
knn_accuracy
```

```
##  Accuracy
## 0.6252709
```

```r
knn_precision
```

```
## Precision
## 0.6167155
```

```r
knn_recall
```

```
##    Recall
## 0.6908673
```

```r
knn_F1_score
```

```
##        F1
## 0.6516889
```

```r
conf_mat_knn # Confusion Matrix details in full
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2103 1307
##          1  941 1648
##
##                Accuracy : 0.6253
##                  95% CI : (0.6129, 0.6375)
##     No Information Rate : 0.5074
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.249
##
##  Mcnemar's Test P-Value : 1.379e-14
##
##             Sensitivity : 0.6909
##             Specificity : 0.5577
##          Pos Pred Value : 0.6167
```

```
##          Neg Pred Value : 0.6365
##              Prevalence : 0.5074
##          Detection Rate : 0.3506
##    Detection Prevalence : 0.5684
##       Balanced Accuracy : 0.6243
##
##        'Positive' Class : 0
##
```

#INTERPRETATION FOR ACCURACY, PRECISION, RECALL, AND F1-SCORE in KNN MODEL Accuracy:

The model's overall accuracy is 0.6254, which indicates that roughly 62.54% of cases were correctly classified.

Precision: The computed precision is 0.6174. The ratio of accurately predicted positive observations to the total number of predicted positives is known as precision. In this instance, it indicates that approximately 61.74% of the occurrences that the model projected to be positive (1) were in fact positive.

Recall: The calculation for recall, sometimes referred to as sensitivity or true positive rate, is 0.6886. The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. In this instance, it indicates that the model accurately predicted approximately 68.86% of the actual positive events.

F1 Score: The harmonic mean of recall and precision is the F1 score. Recall and precision must be balanced. 0.6510 is the reported F1 score.

```
library(pROC)                #Package for ROC calculation
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

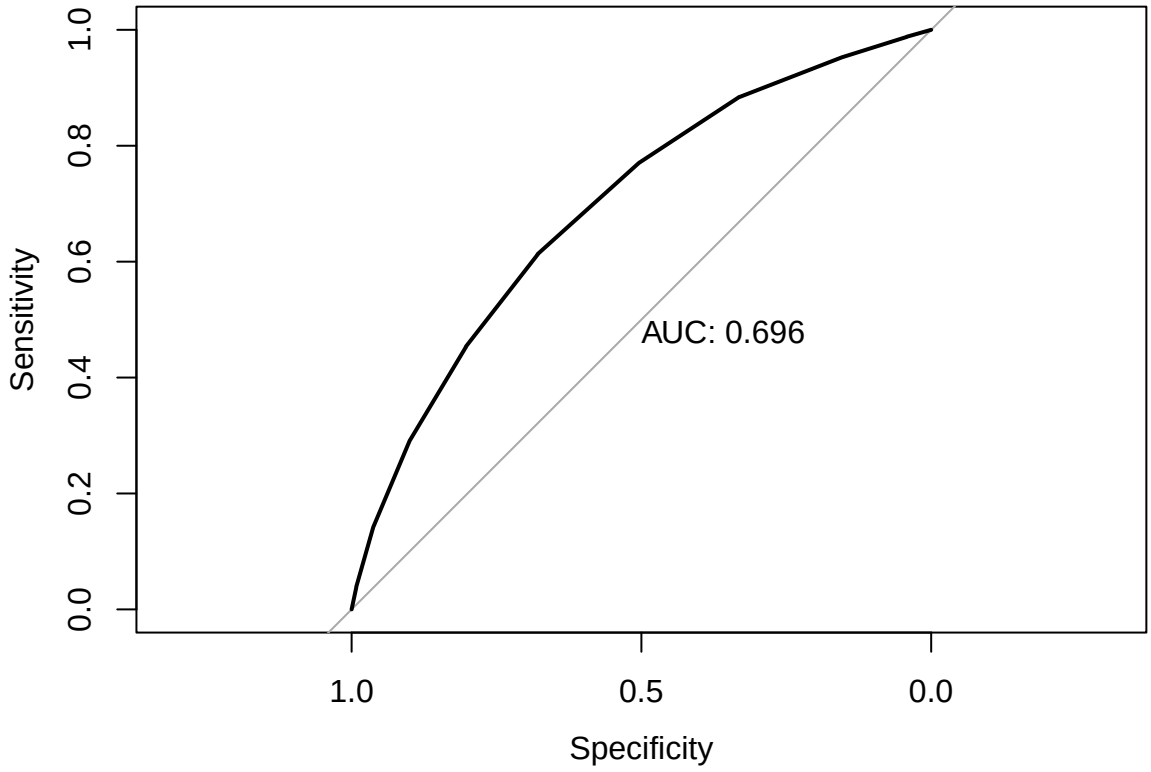
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(dplyr)               #For data manipulation


#split train and test equally.
set.seed(10)
SplitIndex <- sample(x = c("Train", "Test"), size = nrow(knn_data), replace = T, prob = c(0.5,0.5))
TrainData <- filter(knn_data, SplitIndex == "Train")
TestData <- filter(knn_data, SplitIndex == "Test")

#Build the model on training data
set.seed(5)
KnnModel <- train(form = RETENTION ~ .,
                  data = TrainData,
                  method = 'knn')

#Predicted probabilities
KnnProbs <- predict(object = KnnModel, newdata = TestData, type = "prob")
```

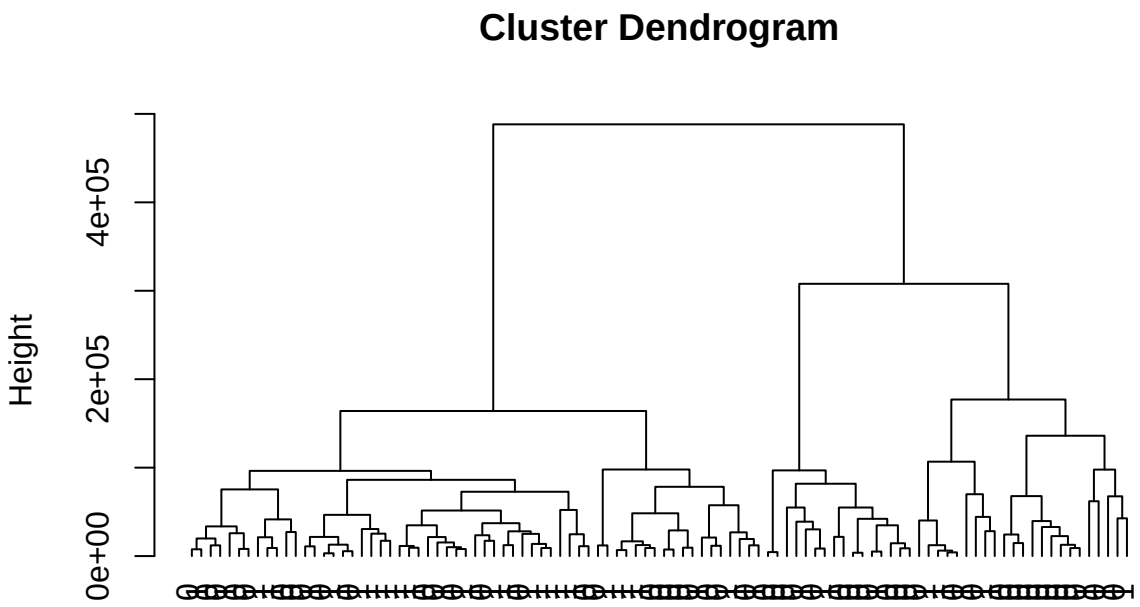
```r
# head(KnnProbs)
KnnProbs <- KnnProbs[,2]      #Only want one probability for each row

#Generate the ROC
KnnROC <- roc(response = TestData$RETENTION, predictor = KnnProbs)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
plot(KnnROC, print.auc = T)
```




```r
library(dplyr)

#for the next plot to be legible, we will change the sample size to 100
set.seed(1)
Sample_bangotelco <- sample_n(tbl = bangortelcodata, size = 100)

#Calculate and plot Hierarchical clutering
#dist: firstly, we will calculate the distance matrix first for the sample_bangotelco, by default Eucli
HClust <- hclust(d = dist(x=Sample_bangotelco[,2:8]), method = "average") #Hierarchical cluster analysi

#hang = -1: this will show the labels of retention on the plot and how they will be displayed. as hang
#See ?plot.hclust for more details
plot(x = HClust, hang = -1, labels=Sample_bangotelco$RETENTION)
```




```
install.packages("dendextend")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'dendextend' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
```

```
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ----------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:rpart':
##
##     prune
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(colorspace)
```

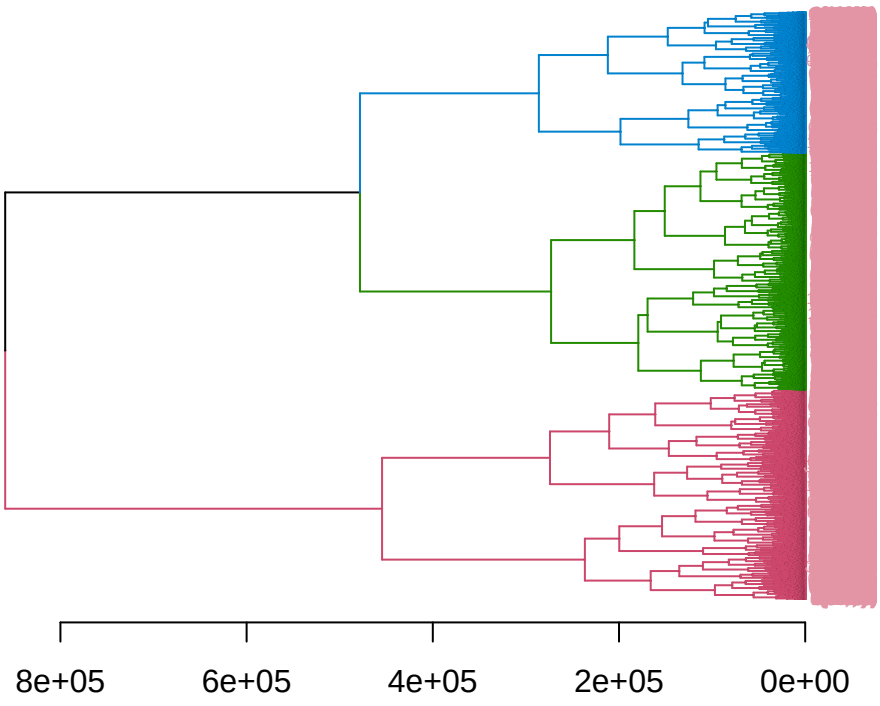
```
##
## Attaching package: 'colorspace'
```

```
## The following object is masked from 'package:pROC':
##
##     coords
```

```
#determine the distances in the bangortelcodata (excluding retention), generate hierarchical clusters
Dist_Bangortelco <- dist(x = bangortelcodata[,2:8], method = "euclidean")
Hc_Bangortelco <- hclust(d = Dist_Bangortelco, method = "complete")     # In complete linkage clusterin
Bangortelco_Dend <- as.dendrogram(Hc_Bangortelco)                       # dendrogram object


# retention column levels saved
RetentionLevs <- rev(levels(bangortelcodata[,13]))

# Color the branches based on the clusters:
Bangortelco_Dend <- color_branches(dend = Bangortelco_Dend, k=3)

#  we will match labels, as much as we can, to the real classification of the column:
# assign colour  ordered by the dendrogram
labels_colors(Bangortelco_Dend) <-
   rainbow_hcl(3)[sort_levels_values(
     as.numeric(bangortelcodata[,13])[order.dendrogram(Bangortelco_Dend)]
     )]

# add the column type to the labels: RETENTION  which is ordered by the dendrogram
labels(Bangortelco_Dend) <- paste(as.character(bangortelcodata[,13])[order.dendrogram(Bangortelco_Dend)]
                          "(",labels(Bangortelco_Dend),")",           #include numbers observed
                          sep = "")
# hang bangortelco_dend:
Bangortelco_Dend <- hang.dendrogram(Bangortelco_Dend,hang_height=0.1)

# label size reduced:
Bangortelco_Dend <- set(dend = Bangortelco_Dend, what = "labels_cex",value = 0.5)
```

```r
# And plot:
par(mar = c(3,3,3,7))
plot(Bangortelco_Dend,
     main = "Clustered bangortelco data set (the labels give the true flower species)",  # set the main
     horiz =  TRUE,   #plots the dendrogram in most cases horizontally
     nodePar = list(cex = .007)) # sets the nodes to have  labels at 7% of the default text size.
```

tered bangortelco data set (the labels give the true flower species)




```r
#circlize graph
install.packages("circlize")
```

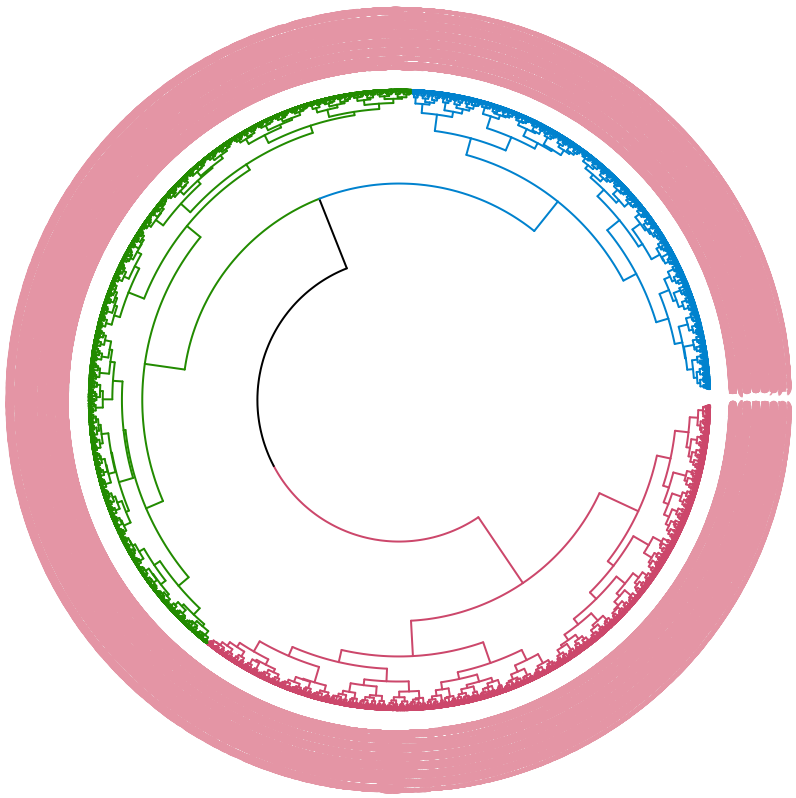
```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'circlize' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```r
library(circlize)
```

```
## ========================================
## circlize version 0.4.15
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
```

```
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
par(mar = rep(1,4))
circlize_dendrogram(Bangortelco_Dend)
```



```
#import cluster
install.packages("cluster")
```

```
## Installing package into 'C:/Users/Windows/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'cluster' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'cluster'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
```

```
## C:\Users\Windows\AppData\Local\R\win-library\4.3\00LOCK\cluster\libs\x64\cluster.dll
## to
## C:\Users\Windows\AppData\Local\R\win-library\4.3\cluster\libs\x64\cluster.dll:
## Permission denied
```

```
## Warning: restored 'cluster'
```

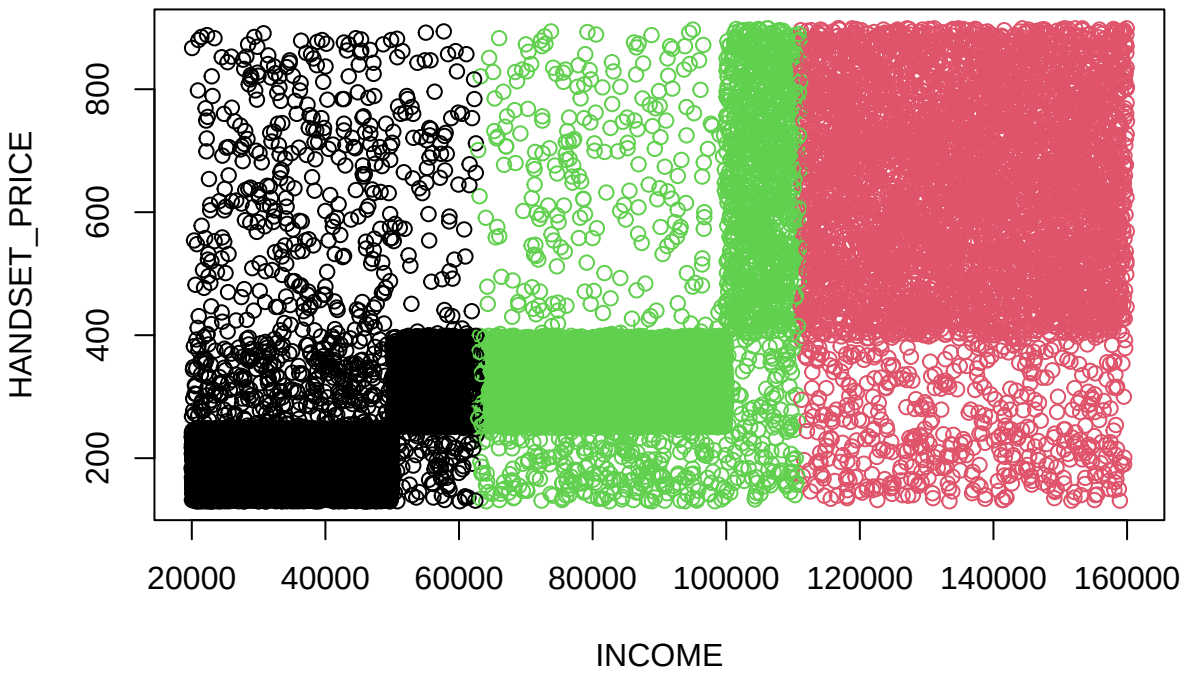
```
##
## The downloaded binary packages are in
##  C:\Users\Windows\AppData\Local\Temp\RtmpOgoVdG\downloaded_packages
```

```
library(cluster)
```

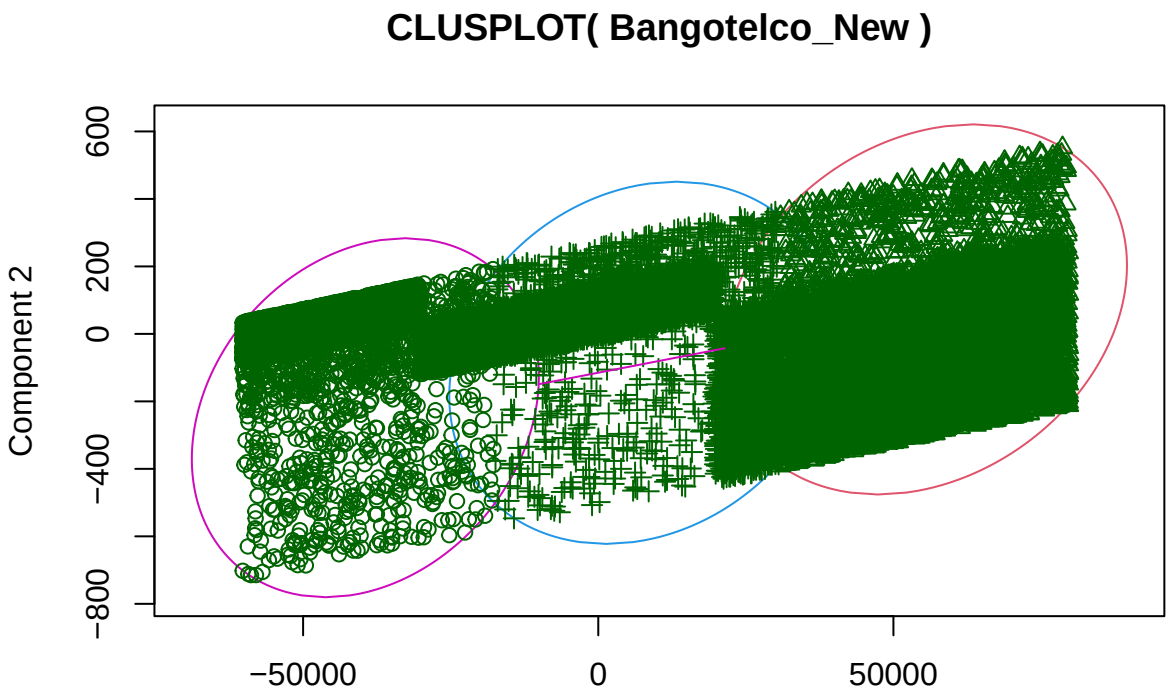
```
Bangotelco_New <- data_tree[,c("INCOME", "HANDSET_PRICE")]

#we will make use of KMeans for the algorithm, and then specify that we want k=3 clusters
Kmean_bangortelcodata <- kmeans(x = Bangotelco_New, center = 3) # neat right!

#Noting that the clusters for the data not having RETENTION , does a fair job
table(RETENTION = bangortelcodata$RETENTION, Cluster = Kmean_bangortelcodata$cluster)
```

```
##          Cluster
## RETENTION    1    2    3
##         0 4545 2349 3254
##         1 3803 3157 2892
```

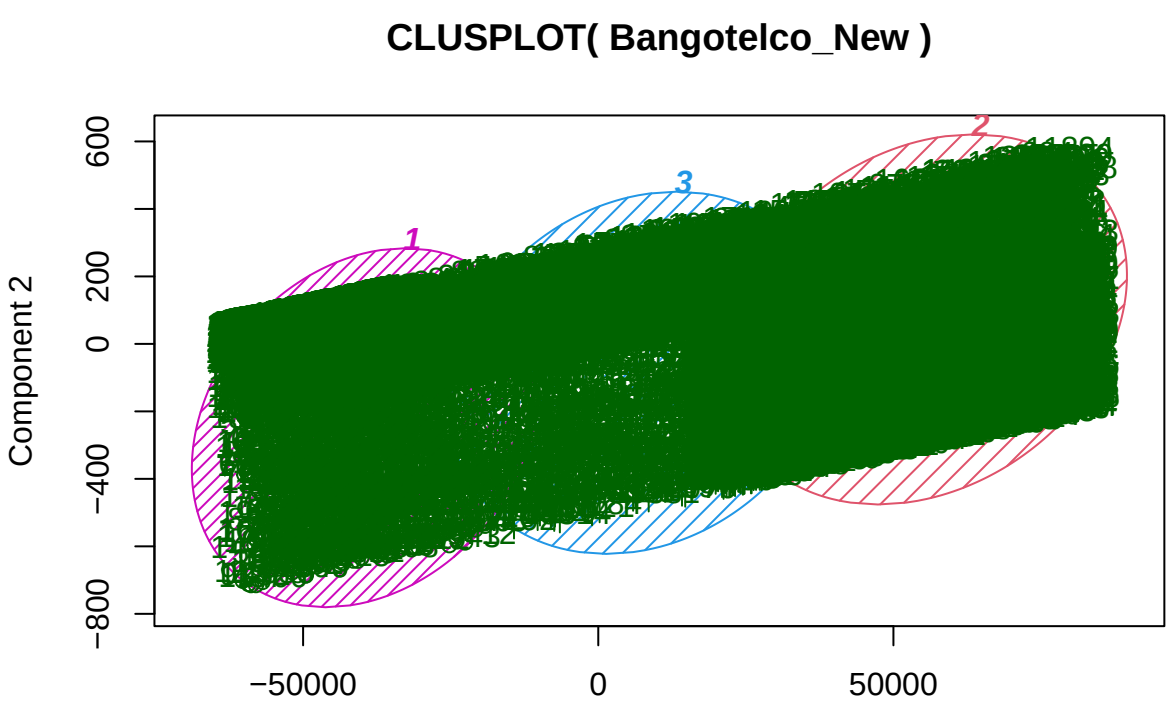
```
# lets plot INCOME and HANDSET
# coloring the points according to the cluster assignments from k-means.
plot(Bangotelco_New[,c("INCOME", "HANDSET_PRICE")], col=Kmean_bangortelcodata$cluster)
# Adds the cluster centers to the plot as points with a different plotting character (pch=8 which is a
points(Kmean_bangortelcodata$centers[,c("INCOME", "HANDSET_PRICE")], col=1:3, pch=8, cex=2)
```




```
#plot cluster
clusplot(Bangotelco_New, Kmean_bangortelcodata$cluster, color = TRUE)
```

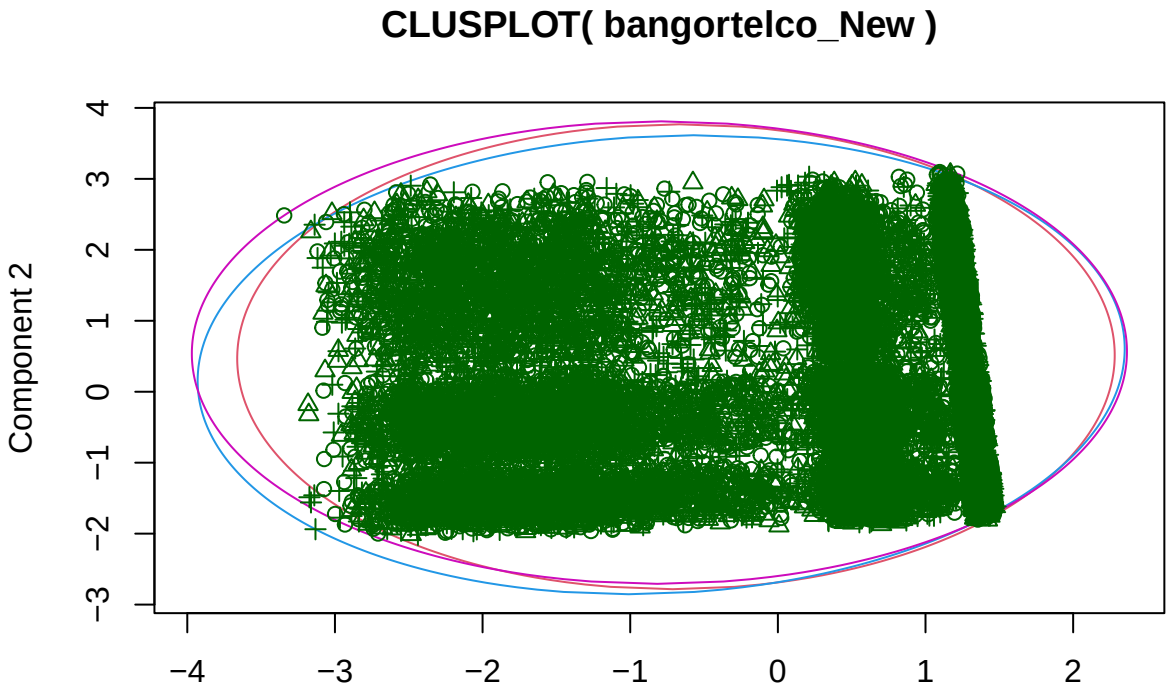



```
clusplot(Bangotelco_New, Kmean_bangortelcodata$cluster, color=TRUE, shade=TRUE,
         labels=2, lines=0)
```



INTERPRETATION OF CLUSTERING(COMPONENT 1) The two component which is INCOME AND HANDSET_PRICE expalins 100% of the point variability

```
# New data without retention
bangortelco_New <- bangortelcodata[, 2:8]

# Lets partition around mediods - fit 3 medoids to the Bangortelco_New data
Bangortelco_Med <- pam(x = bangortelco_New, k = 3)$clustering

# are the clusters well distinguished?
clusplot(bangortelco_New, Bangortelco_Med, color = TRUE)
```



INTERPRETATION OF CLUSTERING(COMPONENT 2) The two component which is INCOME AND HANDSET_PRICE expalins 49.8% of the point variability

```
tinytex::install_tinytex(force = TRUE)
```