# STACY IP

2022-03-30

```
# Define the question :A Kenyan entrepreneur has created an online cryptography course and would want t

# The metric for success:To identify which individuals are most likely to click on her ads.

# The context:To advertise the the cryptography course

# Experimental design taken:Univariate and Bivariate analysis of data using R language.

# The appropriateness of the available data to answer the given question:The data collected provided in
```

```
library (caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(moments)
library(gridExtra)
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(ggplot2)
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.2
## 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:stats':
##
##     cutree
```

```r
library(rpart,quietly = TRUE)
```

```
##
## Attaching package: 'rpart'
```

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

```r
library(caret,quietly = TRUE)
library(rpart.plot,quietly = TRUE)
library(rattle)
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
# Load Data
data<- read.csv('http://bit.ly/IPAdvertisingData')
```

```r
# Preview data
head(data)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                           Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

```
library(magrittr)
```

```
##
## Attaching package: ’magrittr’

## The following object is masked from ’package:purrr’:
##
##     set_names

## The following object is masked from ’package:tidyr’:
##
##     extract
```

```
library(dplyr)
```

```
colnames(data)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

```
colnames(data)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

```
# Get the number of rows and columns in our dataset
dim(data)
```

```
## [1] 1000   10
```

```
# List the columns and data types
str(data)
```

```
# Identifying the numeric class in the data
class(data)
```

```
## [1] "data.frame"
```

```
# Find missing values
colSums(is.na(data))
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                        0                        0                        0
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                        0                        0                        0
##                     Male                  Country                Timestamp
##                        0                        0                        0
##            Clicked.on.Ad
##                        0
```

```
#find out total missing values in each column
# by using the function colSums()
colSums(is.na(data))
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                        0                        0                        0
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                        0                        0                        0
##                     Male                  Country                Timestamp
##                        0                        0                        0
##            Clicked.on.Ad
##                        0
```

```
# to omit all rows containing missing values.
omit<- na.omit(data)
head(omit)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                            Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5          Robust logistical utilization   South Manuel    0    Iceland
## 6        Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

```
# List all the column names
colnames(data)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

```
#Check data types of each column
str(data)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestration" "Monitored national standardi
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ Timestamp               : chr  "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20:35:42"
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

```
# Drop the columns with 'character data type'
df <- data[ -c(5,6,8,9) ]
colnames(df)
```

```
## [1] "Daily.Time.Spent.on.Site" "Age"
## [3] "Area.Income"              "Daily.Internet.Usage"
## [5] "Male"                     "Clicked.on.Ad"
```

```
# Check data types in each column
str(df)
```

```
## 'data.frame':    1000 obs. of  6 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

```
# Convert to numeric
num<- lapply(df, as.numeric)
```

```
# Confirm column names of df
str(df)
```

```
## 'data.frame':    1000 obs. of  6 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

```
# UNIVARIATE ANALYSIS
## Find min,quantile,median, mean, max
summary(df)
```

```
##  Daily.Time.Spent.on.Site      Age         Area.Income    Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##       Male       Clicked.on.Ad
##  Min.   :0.000   Min.   :0.0
##  1st Qu.:0.000   1st Qu.:0.0
##  Median :0.000   Median :0.5
##  Mean   :0.481   Mean   :0.5
##  3rd Qu.:1.000   3rd Qu.:1.0
##  Max.   :1.000   Max.   :1.0
```

```
#  skewness
library(moments)
skewness(df,na.rm =FALSE)
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##              -0.37120261               0.47842268              -0.64939670
##     Daily.Internet.Usage                     Male            Clicked.on.Ad
##              -0.03348703               0.07605493               0.00000000
```

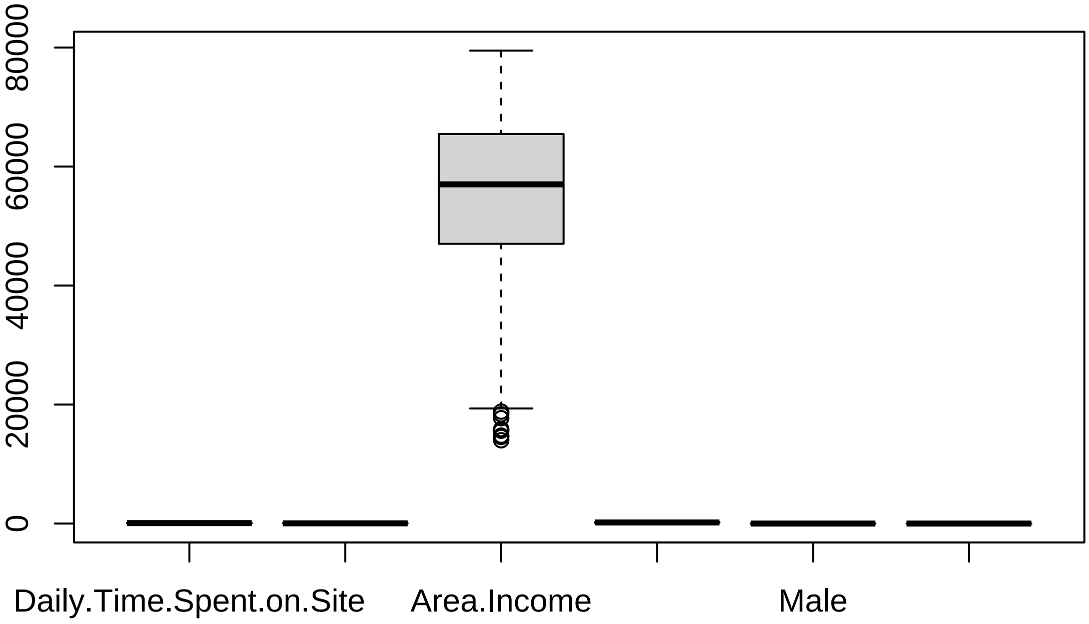
```
# Calculate kurtosis
kurtosis(df,na.rm =FALSE)
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                 1.903942                 2.595482                 2.894694
##     Daily.Internet.Usage                     Male            Clicked.on.Ad
##                 1.727701                 1.005784                 1.000000
```

```
#Standard deviation
sapply(df, sd)
```

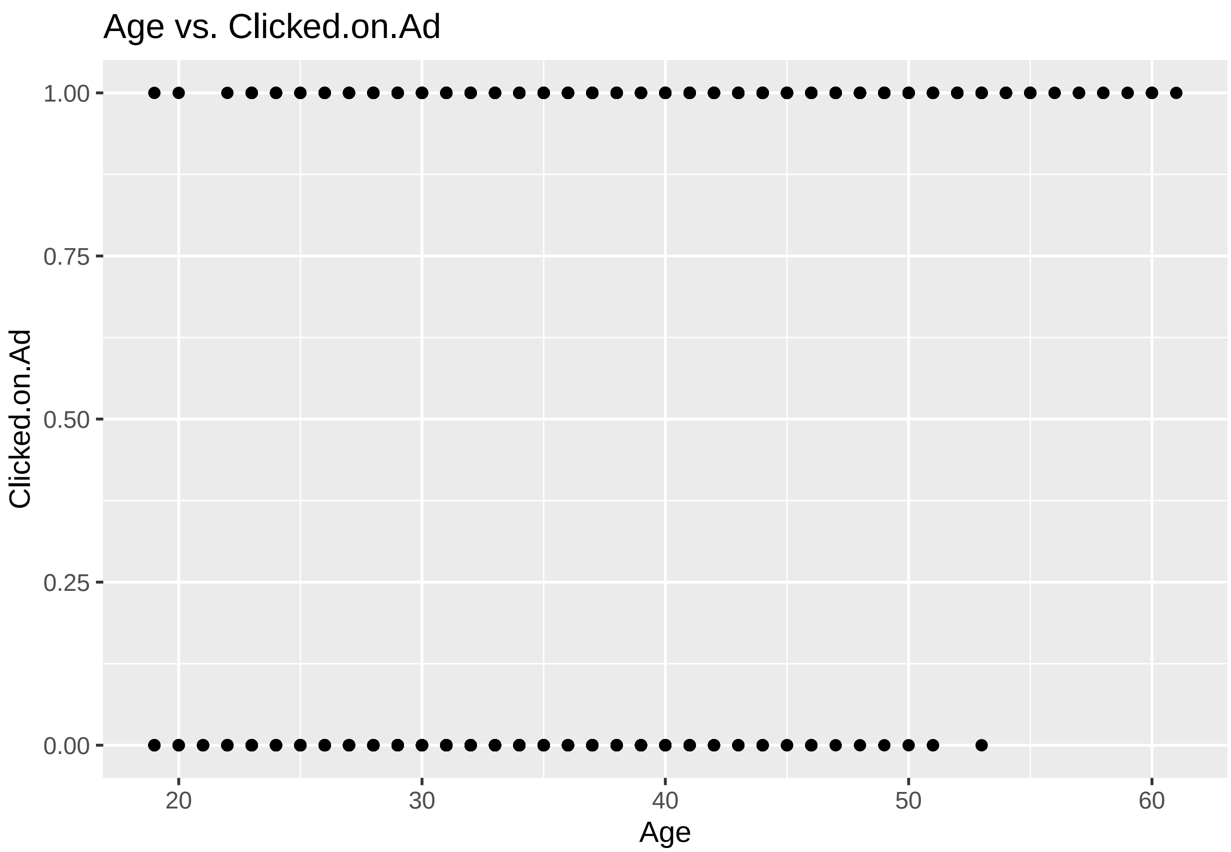
```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##             1.585361e+01             8.785562e+00             1.341463e+04
##     Daily.Internet.Usage                     Male             Clicked.on.Ad
##             4.390234e+01             4.998889e-01             5.002502e-01
```

```
# Outliers
boxplot(df)
```



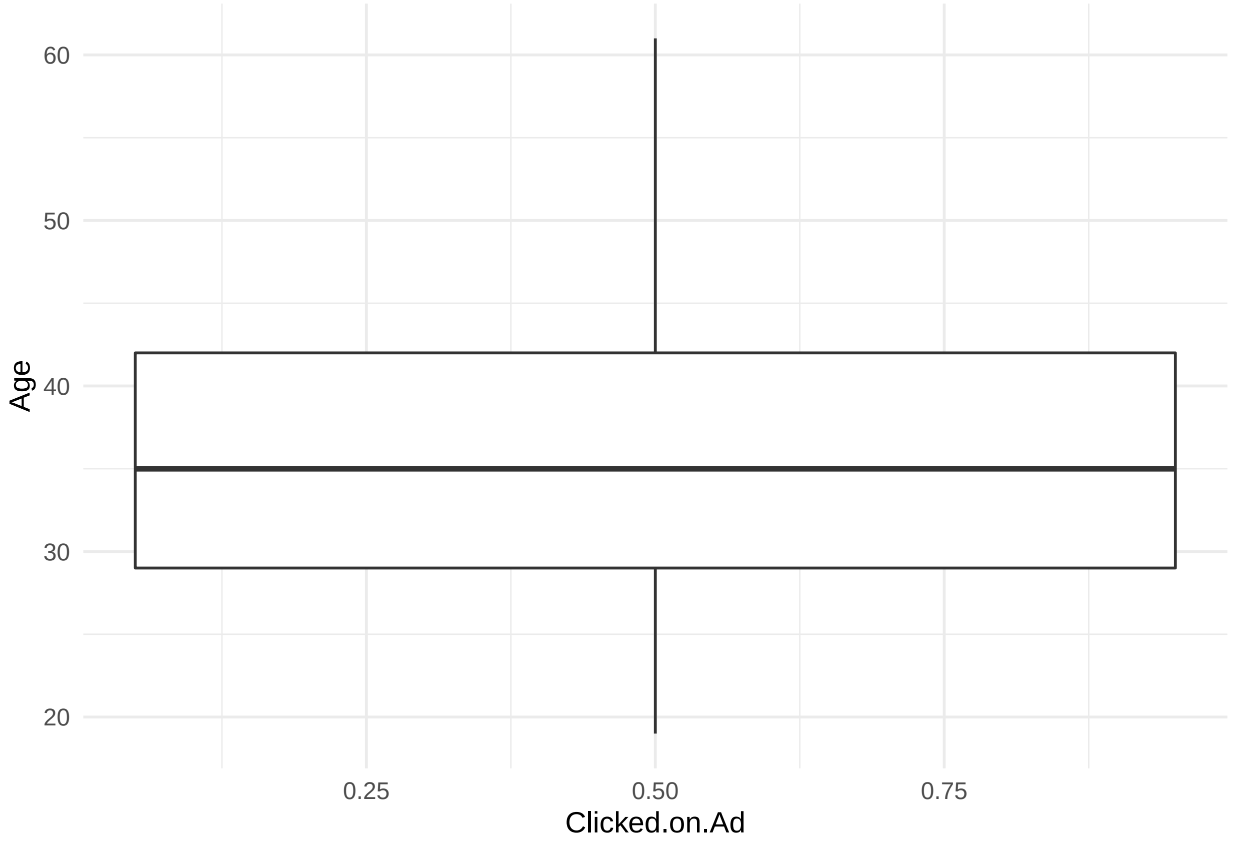
```
library(ggplot2)

qplot(x = Age,
      y = Clicked.on.Ad,
      data = data,geom = "point",
      xlab = "Age",
      ylab = "Clicked.on.Ad",
      main = "Age vs. Clicked.on.Ad");
```



```
library("ggplot2")
# Box plot
bp <- ggplot(data, aes(Clicked.on.Ad, Age )) +
  geom_boxplot(aes(fill = Age)) +
  theme_minimal() +
  theme(legend.position = "top")
bp
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



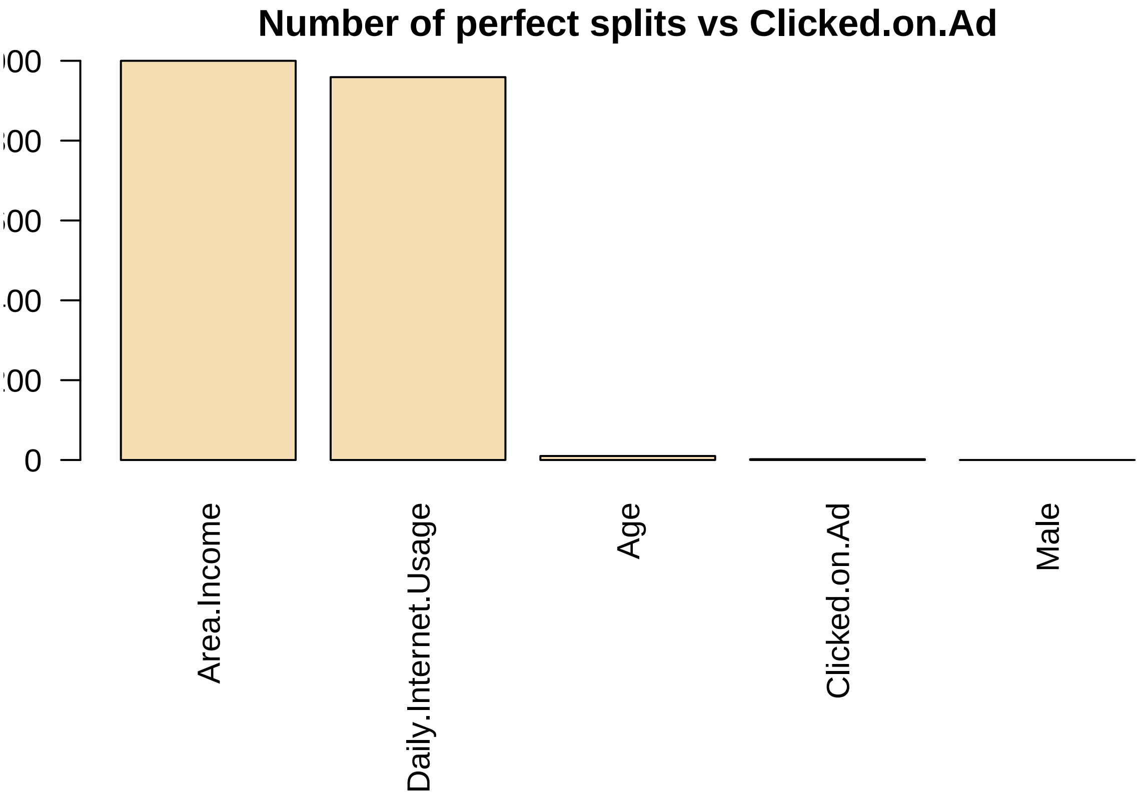
```
# Supervised Learning  DECISION TREE
# Splitting data into training and test data sets

indxTrain <- createDataPartition(y =df$'Clicked.on.Ad',
                                 p = 0.75,list= FALSE)
training <- df[indxTrain,]
testing <- df[-indxTrain,]
```

```
number.perfect.splits <- apply(X=df[-1], MARGIN = 2, FUN = function(col){
t <- table(df$'Clicked.on.Ad',col)
sum(t == 0)})
```

```
# Descending order of perfect splits
order <- order(number.perfect.splits,decreasing = TRUE)
number.perfect.splits <- number.perfect.splits[order]
```

```
# Plot graph
par(mar=c(10,2,2,2))
barplot(number.perfect.splits,
main="Number of perfect splits vs Clicked.on.Ad",
xlab="",ylab="Feature",las=2,col="wheat")
```

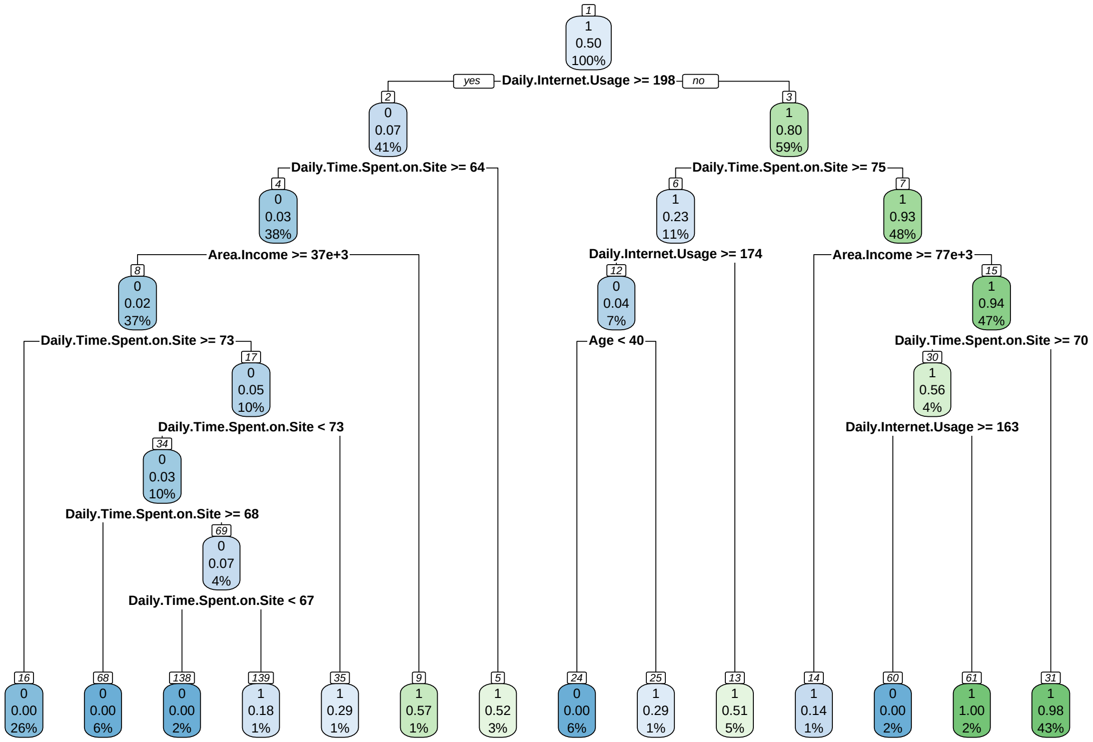


```
#data splicing
set.seed(12345)
train <- sample(1:nrow(data),
                size = ceiling(0.80*nrow(data)),
                replace = FALSE)
# training set
data_train <- df[train,]
# test set
data_test <- df[-train,]
```

```
# penalty matrix
penalty.matrix <- matrix(c(0,1,10,0), byrow=TRUE, nrow=2)
```

```
# building the classification tree with rpart
tree <- rpart(Clicked.on.Ad ~.,data= data_train,
parms = list(loss = penalty.matrix),method = "class")
```

```
# Visualize the decision tree with rpart.plot
rpart.plot(tree, nn=TRUE)
```



```r
#Testing the model
pred <- predict(object=tree,data_test,type="class")
```

```r
data_test = na.omit(data_test)
```

```r
#Calculating accuracy
levels <- levels(pred)
levels <- levels[order(levels)]
table(ordered(pred,levels), ordered(data_test$Clicked.on.Ad, levels))
```

```
##
##      0  1
##   0 81  3
##   1 18 98
```