IP 3rd Sept

Ian_Tirok

September 1, 2021

Defining the Question

Identify individuals who are most likely to will clickon an Ad.

Problem Statement

A Kenyan entrepreneur has created an online cryptography course and would want to advertise it on her blog. She currently targets audiences originating from various countries. In the past, she ran ads to advertise a related course on the same blog and collected data in the process. She would now like to employ your services as a Data Science Consultant to create a solution that would allow her to determine whether ads targeted to audiences of certain characteristics i.e. city, male country, ad topic, etc. would click on her ads.

Metrics for Success

Identify users who are likely to click an ad.

##Experimental Design Taken

Installing packages and loading libraries required

Loading the data

Exploratory Data Analysis

Data Cleaning

Visualizations Modelling Random Forest Predictions and Evaluation of the Model Conclusion

```
r = getOption("repos")
r["CRAN"] = "http://cran.us.r-project.org"
options(repos = r)
install.packages("weatherData")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

```
## Warning: package 'weatherData' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

##Importing Libraries we need for this Project analysis.

```r
##Importing the required packages
install.packages("iterators")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```r
install.packages("rlang")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

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

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\hp\Documents\R\win-library\4.1\00LOCK\rlang\libs\x64\rlang.dll to C:
## \Users\hp\Documents\R\win-library\4.1\rlang\libs\x64\rlang.dll: Permission
## denied

## Warning: restored 'rlang'

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

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

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (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\hp\Documents\R\win-library\4.1\00LOCK\caret\libs\x64\caret.dll to C:
## \Users\hp\Documents\R\win-library\4.1\caret\libs\x64\caret.dll: Permission
## denied

## Warning: restored 'caret'

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

```
install.packages('ranger')
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages('caTools')
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages("caretEnsemble")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages("e1071")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

## package 'e1071' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\hp\AppData\Local\Temp\RtmpEPJMSe\downloaded_packages
```

```
install.packages("randomForest")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages("ggcorrplot")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages('ranger')
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages('caTools')
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

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

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
install.packages("iterators")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

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

```
library(lattice)
library(rpart)
library("rpart.plot")
```

```
# loading the dataset
#dataset_url = http://bit.ly/IPadvertData
advert <- read.csv("http://bit.ly/IPAdvertisingData")
```

```
# printing out the dataset
head(advert)
```

```
##   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
```

```
# printing out the last rows of the dataset
tail(advert)
```

```
##      Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 995                     43.70  28    63126.96               173.01
## 996                     72.97  30    71384.57               208.58
## 997                     51.30  45    67782.17               134.42
## 998                     51.63  51    42415.72               120.37
## 999                     55.55  19    41920.79               187.95
## 1000                    45.01  26    29875.80               178.35
##                             Ad.Topic.Line          City Male
## 995         Front-line bifurcated ability  Nicholasland    0
## 996         Fundamental modular algorithm     Duffystad    1
## 997       Grass-roots cohesive monitoring   New Darlene    1
## 998          Expanded intangible solution South Jessica    1
## 999  Proactive bandwidth-monitored policy   West Steven    0
## 1000      Virtual 5thgeneration emulation   Ronniemouth    0
##                      Country           Timestamp Clicked.on.Ad
## 995                  Mayotte 2016-04-04 03:57:48             1
```

```
## 996                 Lebanon 2016-02-11 21:49:00            1
## 997  Bosnia and Herzegovina 2016-04-22 02:07:01            1
## 998                Mongolia 2016-02-01 17:24:57            1
## 999               Guatemala 2016-03-24 02:35:54            0
## 1000                 Brazil 2016-06-03 21:43:21            1
```

```
# checking for the number of rows and columns
dim(advert)
```

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

The dataset has 1000 rows and 10 columns

```
# this is to check for attributes
sapply(advert, class)
```

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

```
# this is to get a summary statistics of the dataset
summary(advert)
```

```
##  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
##  Ad.Topic.Line          City                Male         Country
##  Length:1000        Length:1000        Min.   :0.000   Length:1000
##  Class :character   Class :character   1st Qu.:0.000   Class :character
##  Mode  :character   Mode  :character   Median :0.000   Mode  :character
##                                        Mean   :0.481
##                                        3rd Qu.:1.000
##                                        Max.   :1.000
##   Timestamp         Clicked.on.Ad
##  Length:1000        Min.   :0.0
##  Class :character   1st Qu.:0.0
##  Mode  :character   Median :0.5
##                     Mean   :0.5
##                     3rd Qu.:1.0
##                     Max.   :1.0
```

```
# Data cleaning
## check for missing values in our data
colSums(is.na(advert))
```

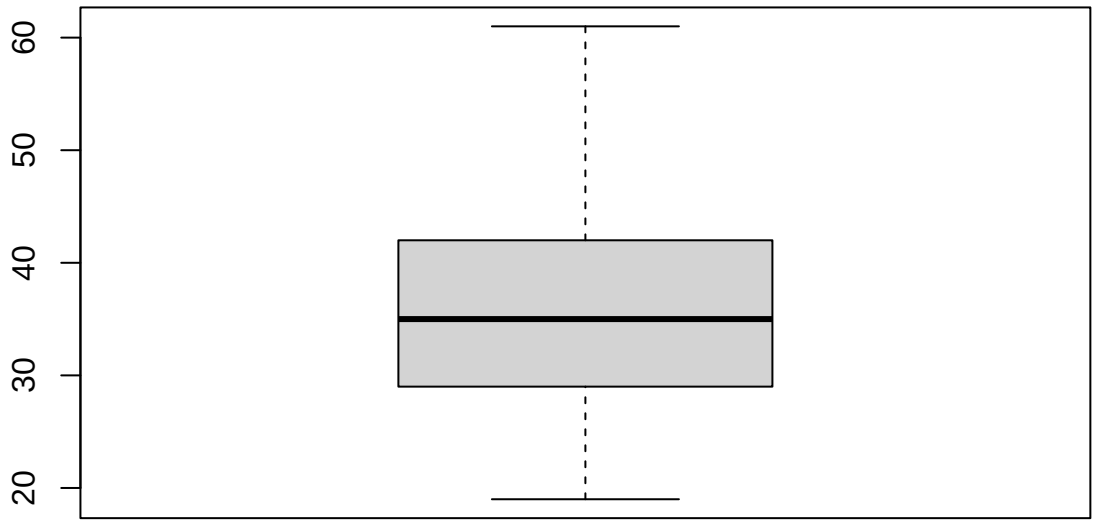
```
## Daily.Time.Spent.on.Site                       Age              Area.Income
##                        0                         0                        0
##     Daily.Internet.Usage            Ad.Topic.Line                      City
##                        0                         0      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

There doesn't seem to be any missing rows

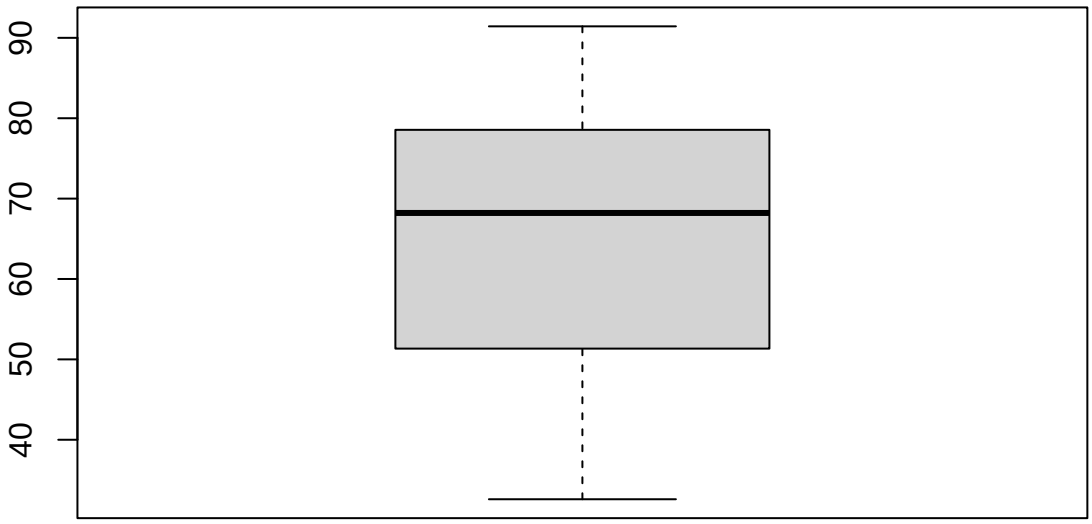
```
# Check for duplicates in our data
duplicated_rows <- advert[duplicated(advert),]
duplicated_rows
```

```
##  [1] Daily.Time.Spent.on.Site Age                      Area.Income
##  [4] Daily.Internet.Usage     Ad.Topic.Line            City
##  [7] Male                     Country                  Timestamp
## [10] Clicked.on.Ad
## <0 rows> (or 0-length row.names)
```

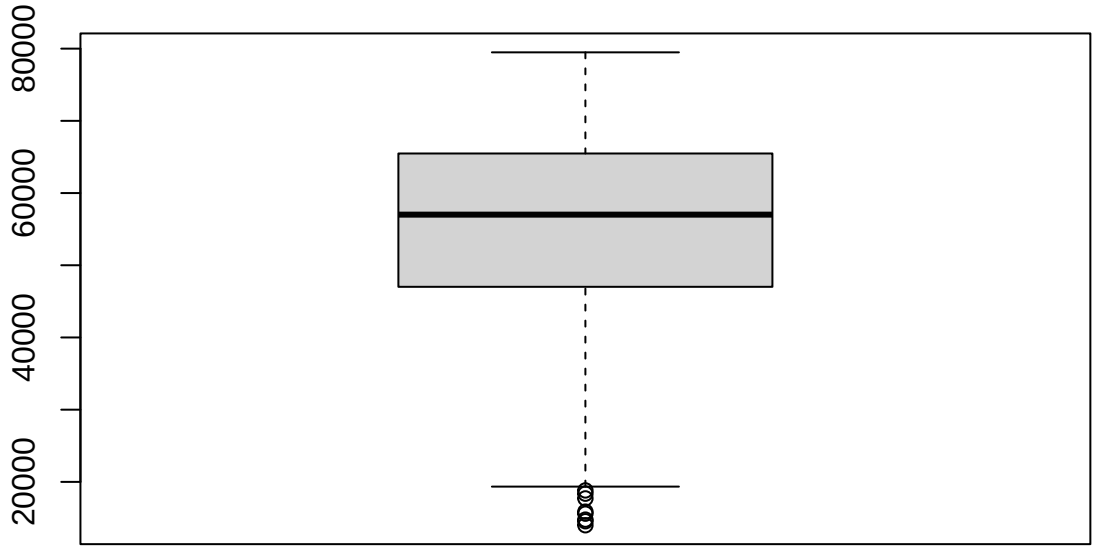
```
#Checking the outliers in the Age Column.
boxplot(advert$Age)
```




```
#Checking the outliers in the Time spent on site Column.
boxplot(advert$'Daily.Time.Spent.on.Site')
```

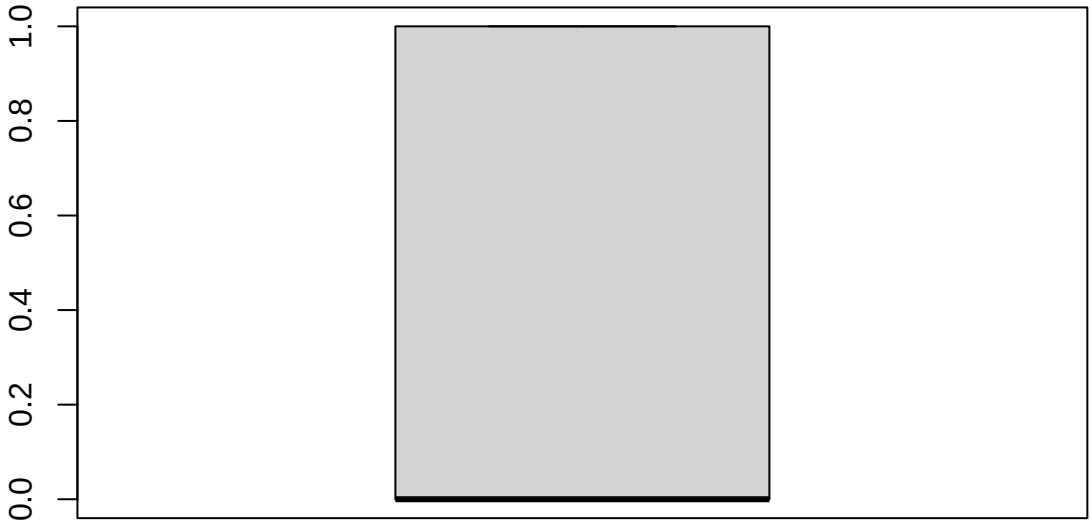



```
#Checking the outliers in the Area income Column.

boxplot(advert$'Area.Income')
```



There are outliers with the area income but it is possible for people to earn outside the innterquatile range. so we will not remove the outliers

```
#Checking the outliers in the Male Column.

boxplot(advert$'Male')
```




```
head(advert)
```

```
##   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
```

##Exploring the data EXPLAROTARY DATA ANALYSIS

Measures of dispersion

```
dt.mean <- mean(advert$'Daily.Time.Spent.on.Site')
dt.mean
```

```
## [1] 65.0002
```

```
dt.median <- median(advert$'Daily.Time.Spent.on.Site')
dt.median
```

```
## [1] 68.215
```

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
dt.mode <- getmode(advert$'Daily.Time.Spent.on.Site')
dt.mode
```

```
## [1] 62.26
```

Daily Time Spent on Site Measures of central tendency

Mean - 65.002 Median - 68.215 Mode - 62.26

```
#Checking the mean, median and mode of the age column
age.mean <- mean(advert$Age)
age.mean
```

```
## [1] 36.009
```

```
age.median <- median(advert$Age)
age.median
```

```
## [1] 35
```

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
age.mode <- getmode(advert$Age)
age.mode
```

```
## [1] 31
```

Age Measures of central tendency

Mean - 36.009 Median - 35 Mode - 31

```r
#Checking the mean, median and mode of the area income column
ai.mean <- mean(advert$'Area.Income')
ai.mean
```

```
## [1] 55000
```

```r
ai.median <- median(advert$'Area.Income')
ai.median
```

```
## [1] 57012.3
```

```r
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
ai.mode <- getmode(advert$'Area.Income')
ai.mode
```

```
## [1] 61833.9
```

Area income Measures of central tendency

Mean - 55000.00008 Median - 57012.3 Mode -61833.9

```r
#Checking the mean, median and mode of the daily internet usage column
diu.mean <- mean(advert$'Daily.Internet.Usage')
diu.mean
```

```
## [1] 180.0001
```

```r
diu.median <- median(advert$'Daily.Internet.Usage')
diu.median
```

```
## [1] 183.13
```

```r
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
diu.mode <- getmode(advert$'Daily.Internet.Usage')
diu.mode
```

```
## [1] 167.22
```

Daily Internet Usage Measures of central tendency

Mean - 180.0001 Median - 183.13 Mode -167.22

```r
#Checking the mode of the country column
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
country.mode <- getmode(advert$Country)
country.mode
```

```
## [1] "Czech Republic"
```

Czech Republic is the most frequent country.

```
#Checking the mode of the city column
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
city.mode <- getmode(advert$City)
city.mode
```

```
## [1] "Lisamouth"
```

Lisamouth is the most frequent city.

```
#Checking the mode of the sex column
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
Male.mode <- getmode(advert$Male)
Male.mode
```

```
## [1] 0
```

```
#Checking the mode of the Daily.Internet.Usage column
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
diu.mode <- getmode(advert$'Daily.Internet.Usage')
diu.mode
```

```
## [1] 167.22
```

Majority of the site visitors had a daily internet usage of 167.22

```
#Checking the mode of the Year column
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
yr.mode <- getmode(advert$Timestamp
                   )
yr.mode
```

```
## [1] "2016-03-27 00:53:11"
```

The site had most traffic on 27th March 2016 at 53 minutes past midnight

Measures of dispersion Range

```r
#Finding the Range
range(advert$'Daily.Time.Spent.on.Site')
```

```
## [1] 32.60 91.43
```

```r
range(advert$'Age')
```

```
## [1] 19 61
```

```r
range(advert$'Area.Income')
```

```
## [1] 13996.5 79484.8
```

```r
range(advert$'Daily.Internet.Usage')
```

```
## [1] 104.78 269.96
```

```r
range(advert$'Male')
```

```
## [1] 0 1
```

```r
range(advert$'Clicked.on.Ad')
```

```
## [1] 0 1
```

The time spent on the site ranges from 32 minutes to 91 minutes

The range of the age is 19 to 61 years old old

The range of the area income is between 13,996 to 79,848 dollars

The range of daily internet usage is betweem 104 minutes to 269 minutes

The range of gender is 1 because there are only 2 possibilities

The range of whether ad was clicked or not is 1. Because there are only 2 possibilites

Standard Deviation

```r
#Finding the Standard Deviation

sd(advert$'Daily.Time.Spen.on.Site')
```

```
## [1] NA
```

```r
sd(advert$'Age')
```

```
## [1] 8.785562
```

```
sd(advert$'Area.Income')
```

```
## [1] 13414.63
```

```
sd(advert$'Daily.Internet.Usage')
```

```
## [1] 43.90234
```

```
sd(advert$'Male')
```

```
## [1] 0.4998889
```

```
sd(advert$'Clicked.on.Ad')
```

```
## [1] 0.5002502
```

Standard deviation of Daily time spent on site is 15.85361 Standard deviation of Age is 8.785562 Standard deviation of Area income is 13414.63 Standard deviation of Daily internet usage is 43.90234 Standard deviation of Gender is 0.4998889 Standard deviation of Clicked on Ad is 0.5002502

Calculating Variance

```
#Finding the Standard Deviation

var(advert$'Daily.Time.Spent.on.Site')
```

```
## [1] 251.3371
```

```
var(advert$'Age')
```

```
## [1] 77.18611
```

```
var(advert$'Area.Income')
```

```
## [1] 179952406
```

```
var(advert$'Daily.Internet.Usage')
```

```
## [1] 1927.415
```

```
var(advert$'Male')
```

```
## [1] 0.2498889
```

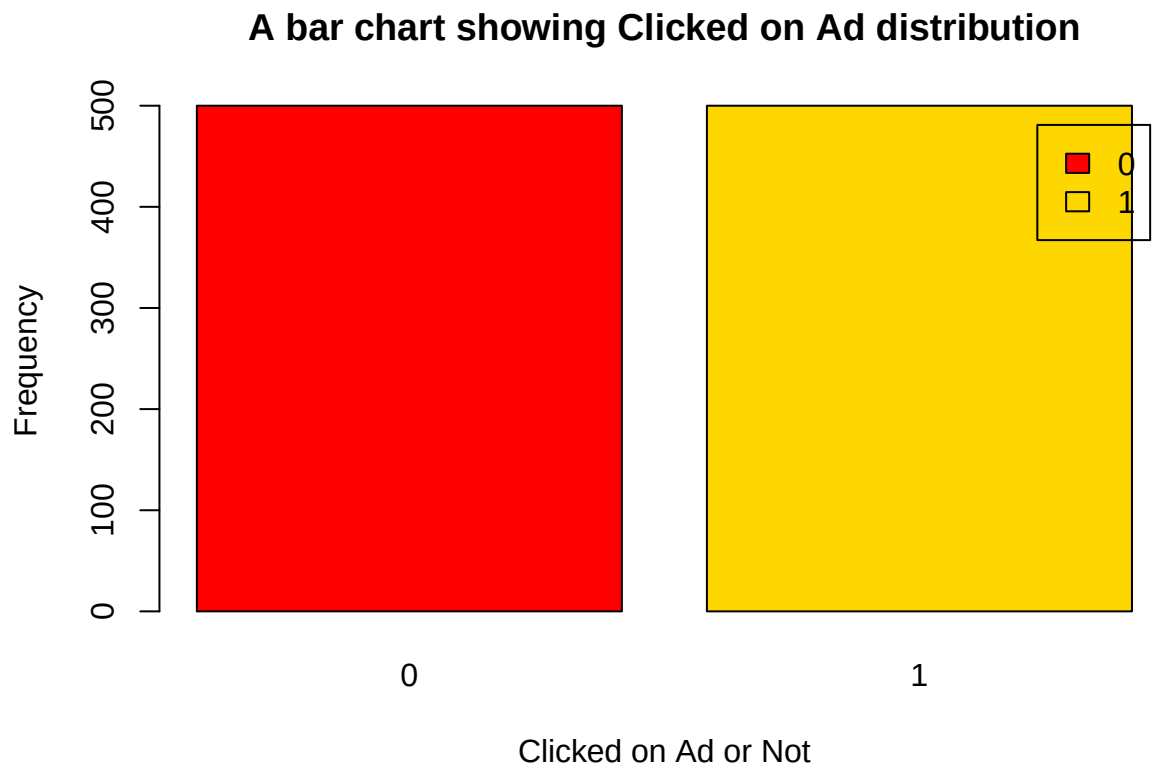
```
var(advert$'Clicked.on.Ad')
```

```
## [1] 0.2502503
```

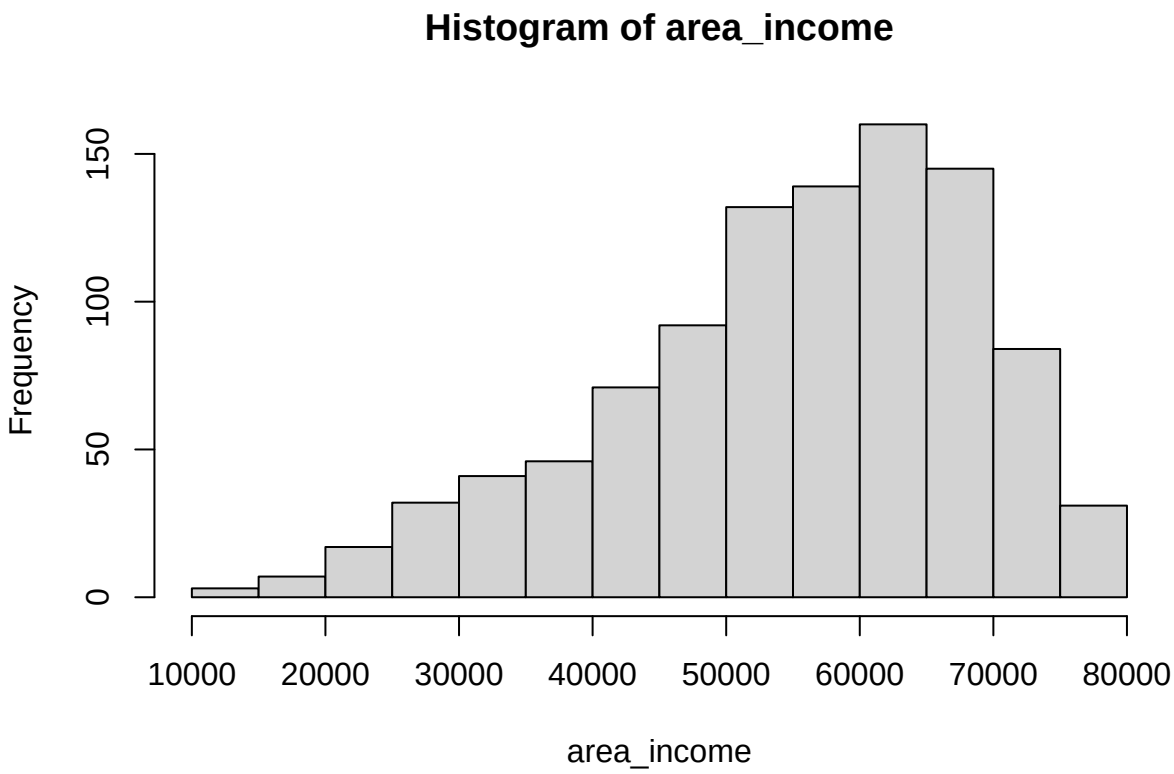
Variance of Daily time spent on site is 251.3371 Variance of Age is 77.18611 Variance of Area income is 179952406 Variance of Daily internet usage is 1927.415 Variance of Gender is 0.2498889 Variance of Clicked on Ad is 0.2502503

##Univariate analysis

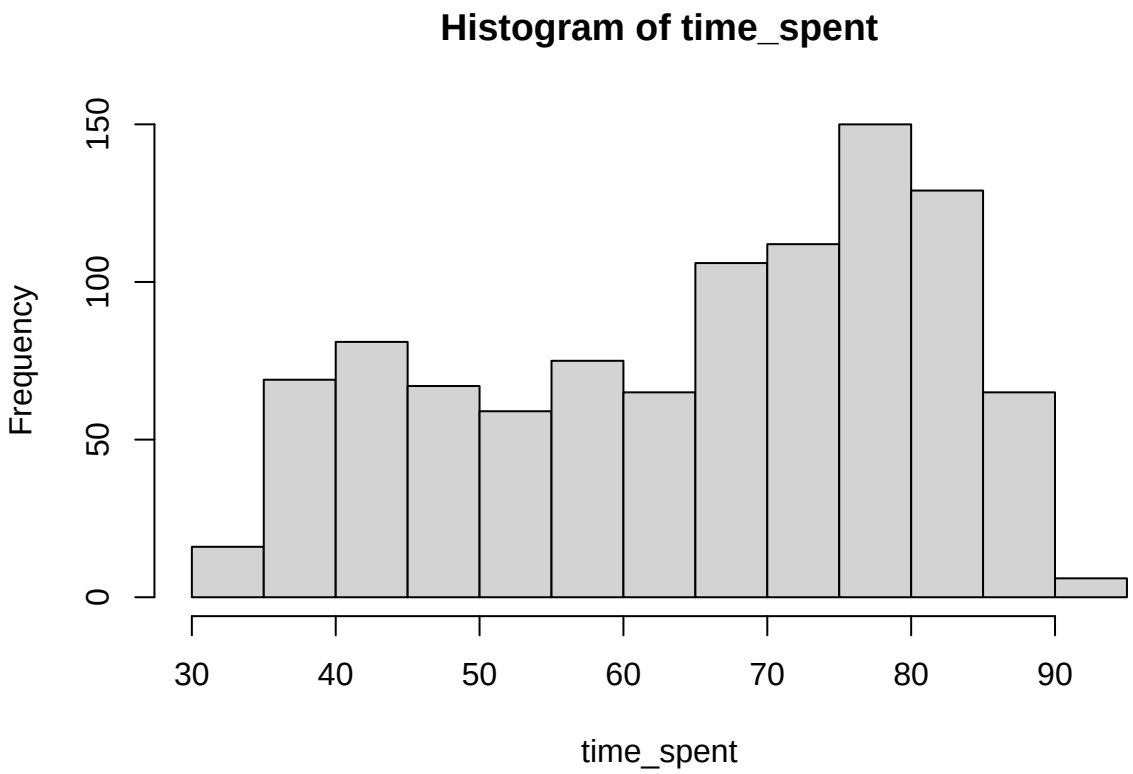
```r
#stacked bars

counts <- table(advert$Clicked.on.Ad)
barplot(counts,
        main="A bar chart showing Clicked on Ad distribution",
        xlab="Clicked on Ad or Not",
        ylab = "Frequency",
        col=c("red","gold"),
        legend = rownames(counts))
```



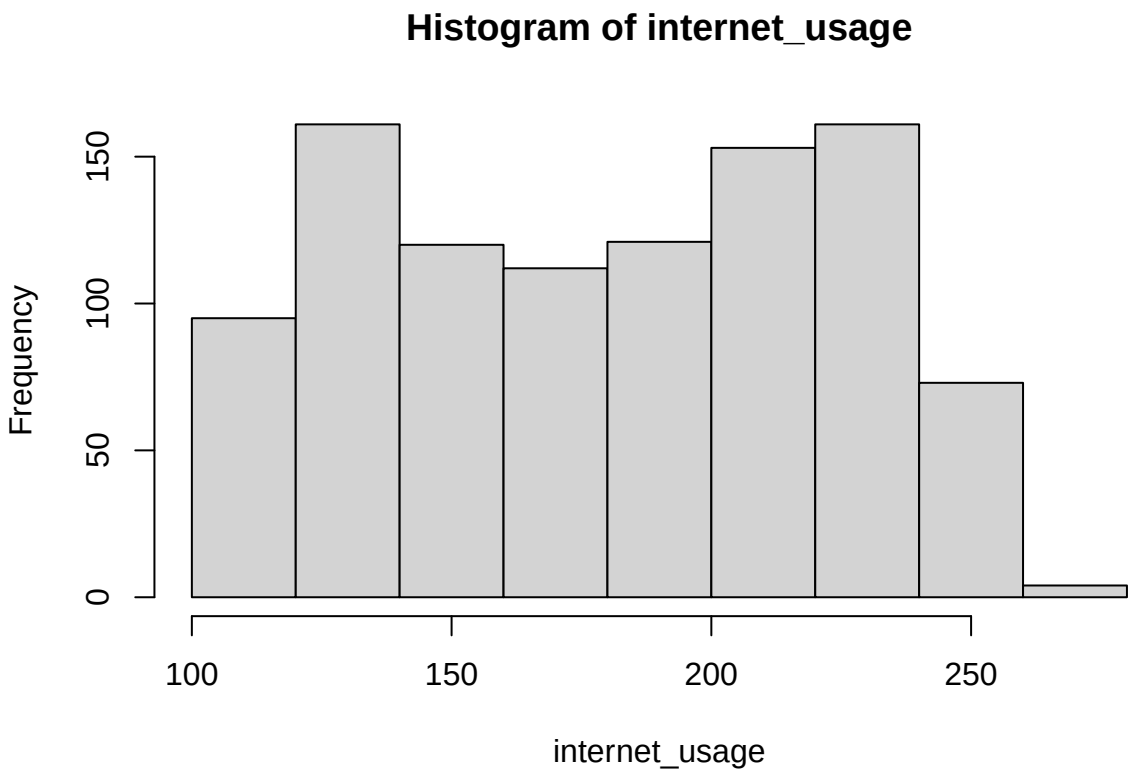

```r
# Histogram
area_income<-(advert$Area.Income)
time_spent<-(advert$Daily.Time.Spent.on.Site)
internet_usage<-(advert$Daily.Internet.Usage)
hist(area_income)
```




```
hist(time_spent)
```

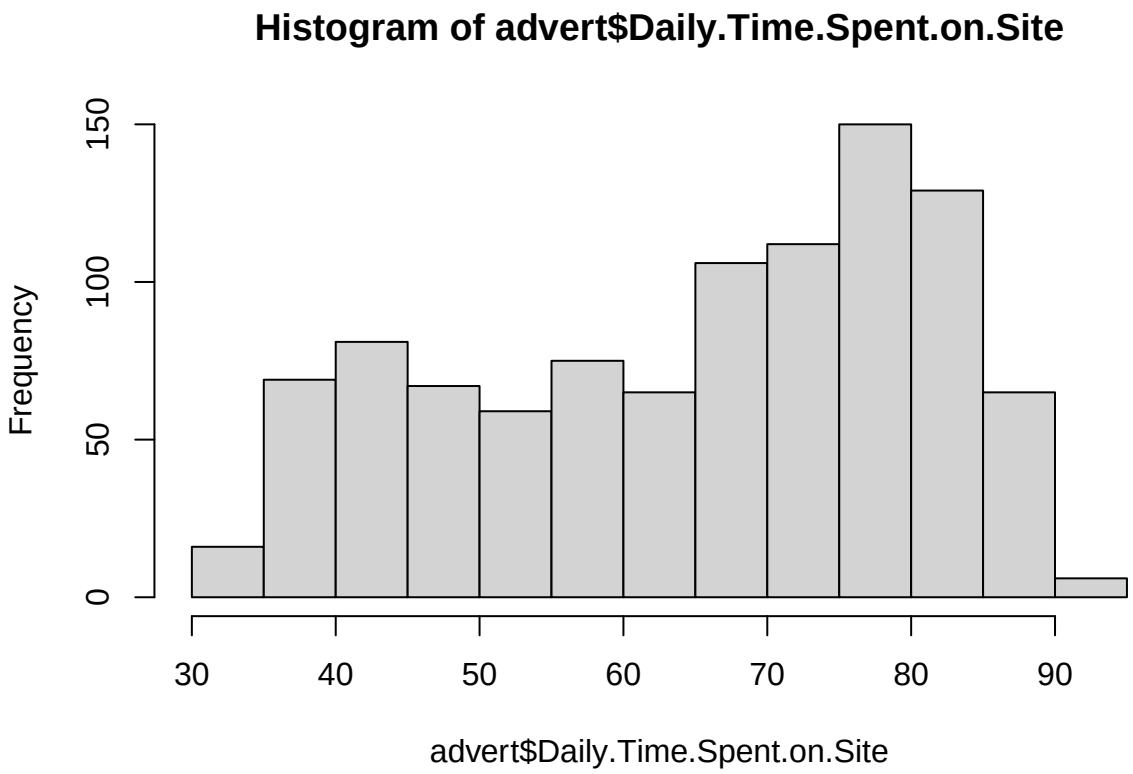



```
hist(internet_usage)
```



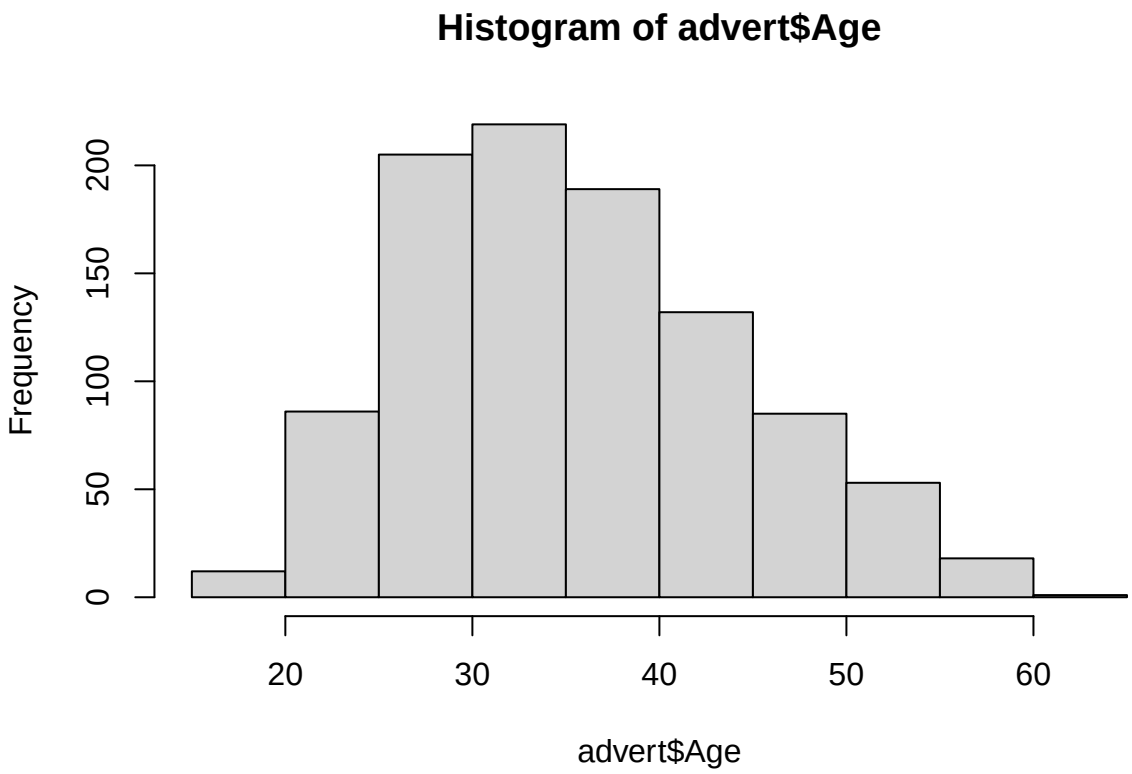
Observations 1. Majority of the website visitors earn between 55,000 nd 70,000 2. Majority of the website visitors spend between 70 and 85 minutes on the website 3. Majority of the website visitors spend either 120 - 140 minutes or 170 to 230 minutes on the website

```
# Histogram for Daily Time Spent on Site
hist(advert$`Daily.Time.Spent.on.Site`)
```

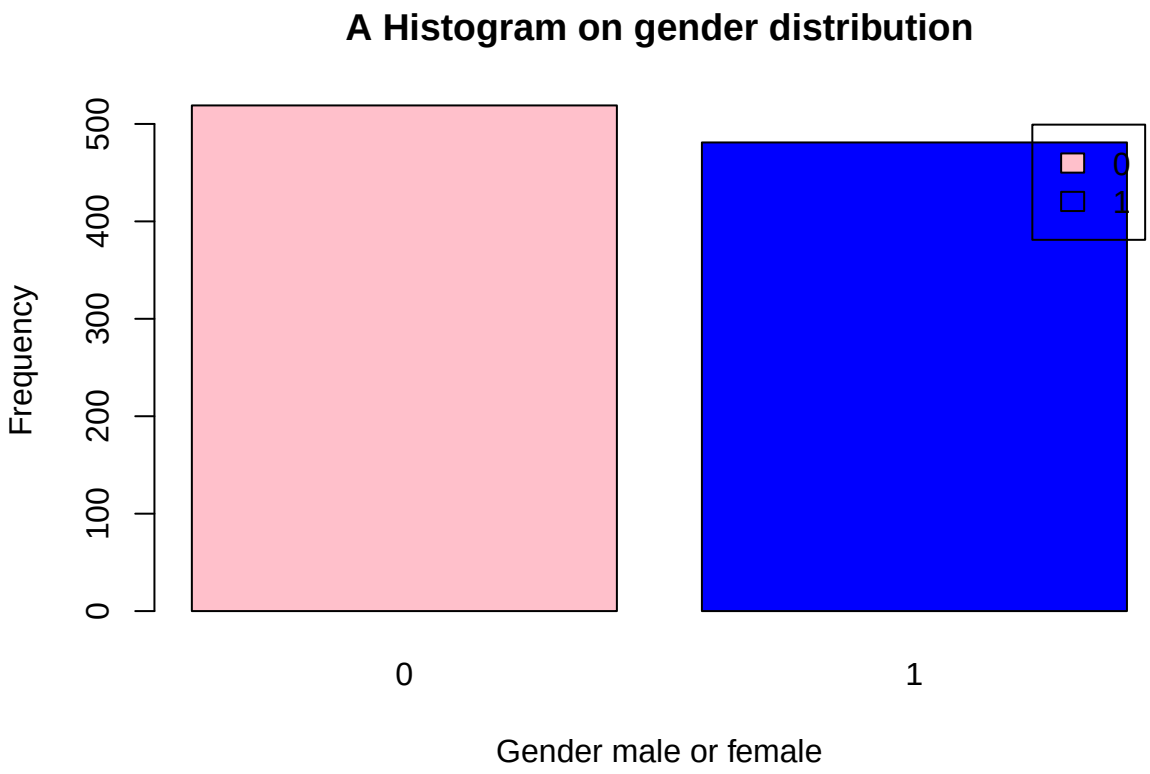


Most people spend about 75 minutes on the website

```
# Histogram for age
hist(advert$`Age`)
```




```
#stacked bars

counts <- table(advert$Male)
barplot(counts,
        main="A Histogram on gender distribution",
        xlab="Gender male or female",
        ylab = "Frequency",
        col=c("pink","blue"),
        legend = rownames(counts))
```



Male is represented by blue This shows that there are more female than male visitors of the website ##BIVARIATE ANALYSIS **Covariance** Covariance measures the directional relationship between the returns on two assets. A positive covariance means that asset returns move together while a negative covariance means they move inversely.

```
###Bivariate analysis
#Covariance of age and click on ad
age <- advert$Age
click <- advert$'Clicked.on.Ad'
gender <- advert$Male
cov(age, click)
```

```
## [1] 2.164665
```

In this instance, the positive covariance shows a positive correlation between the 2 varibles.

```
###Bivariate analysis
#Covariance of time spent on site and click on ad
time <- advert$Daily.Time.Spent.on.Site

click <- advert$'Clicked.on.Ad'
gender <- advert$Male
cov(time, click)
```

```
## [1] -5.933143
```

This means the more time spent on the website, the less likely the user will click on your ad

```
###Bivariate analysis
#Covariance income and click on ad
income <- advert$Area.Income

click <- advert$'Clicked.on.Ad'
gender <- advert$Male
cov(income, click)
```

```
## [1] -3195.989
```

This is a very high negative covariance - meaning there is no correlation between user's income and whether they click on the ad

```
###Bivariate analysis
#Covariance of internet and click on ad
click <- advert$'Clicked.on.Ad'

intusage <- advert$'Daily.Internet.Usage'
gender <- advert$Male
cov(intusage, click)
```

```
## [1] -17.27409
```

The more time spent online by the user, the less likely they will click on your ad

```
#Finding the correlation
cor <- cor(advert[, unlist(lapply(advert, is.numeric))])
round(cor, 3)
```

```
##                           Daily.Time.Spent.on.Site    Age Area.Income
## Daily.Time.Spent.on.Site                    1.000 -0.332       0.311
## Age                                        -0.332  1.000      -0.183
## Area.Income                                 0.311 -0.183       1.000
## Daily.Internet.Usage                        0.519 -0.367       0.337
## Male                                       -0.019 -0.021       0.001
## Clicked.on.Ad                              -0.748  0.493      -0.476
##                           Daily.Internet.Usage   Male Clicked.on.Ad
## Daily.Time.Spent.on.Site                0.519 -0.019        -0.748
## Age                                    -0.367 -0.021         0.493
## Area.Income                             0.337  0.001        -0.476
## Daily.Internet.Usage                    1.000  0.028        -0.787
## Male                                    0.028  1.000        -0.038
## Clicked.on.Ad                          -0.787 -0.038         1.000
```

```
#selecting Clicked.on.Ad data that had 1
clicked <- advert[advert$Clicked.on.Ad == 1,]
head(clicked)
```

```
##    Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
```

```
## 8                     66.00  48    24593.33               131.76
## 11                    47.64  49    45632.51               122.02
## 13                    69.57  48    51636.92               113.12
## 15                    42.95  33    30976.00               143.56
## 16                    63.45  23    52182.23               140.64
## 17                    55.39  37    23936.86               129.41
##                           Ad.Topic.Line             City Male
## 8               Reactive local challenge Port Jefferybury    1
## 11        Centralized neutral neural-net  West Brandonton    0
## 13 Centralized content-based focus group   West Katiefurt    1
## 15         Grass-roots coherent extranet     West William    0
## 16    Persistent demand-driven interface   New Travistown    1
## 17    Customizable multi-tasking website   West Dylanberg    0
##                   Country           Timestamp Clicked.on.Ad
## 8               Australia 2016-03-07 01:40:15             1
## 11                 Qatar 2016-03-16 20:19:01             1
## 13                 Egypt 2016-06-03 01:14:41             1
## 15              Barbados 2016-03-24 09:31:49             1
## 16                 Spain 2016-03-09 03:41:30             1
## 17 Palestinian Territory 2016-01-30 19:20:41             1
```

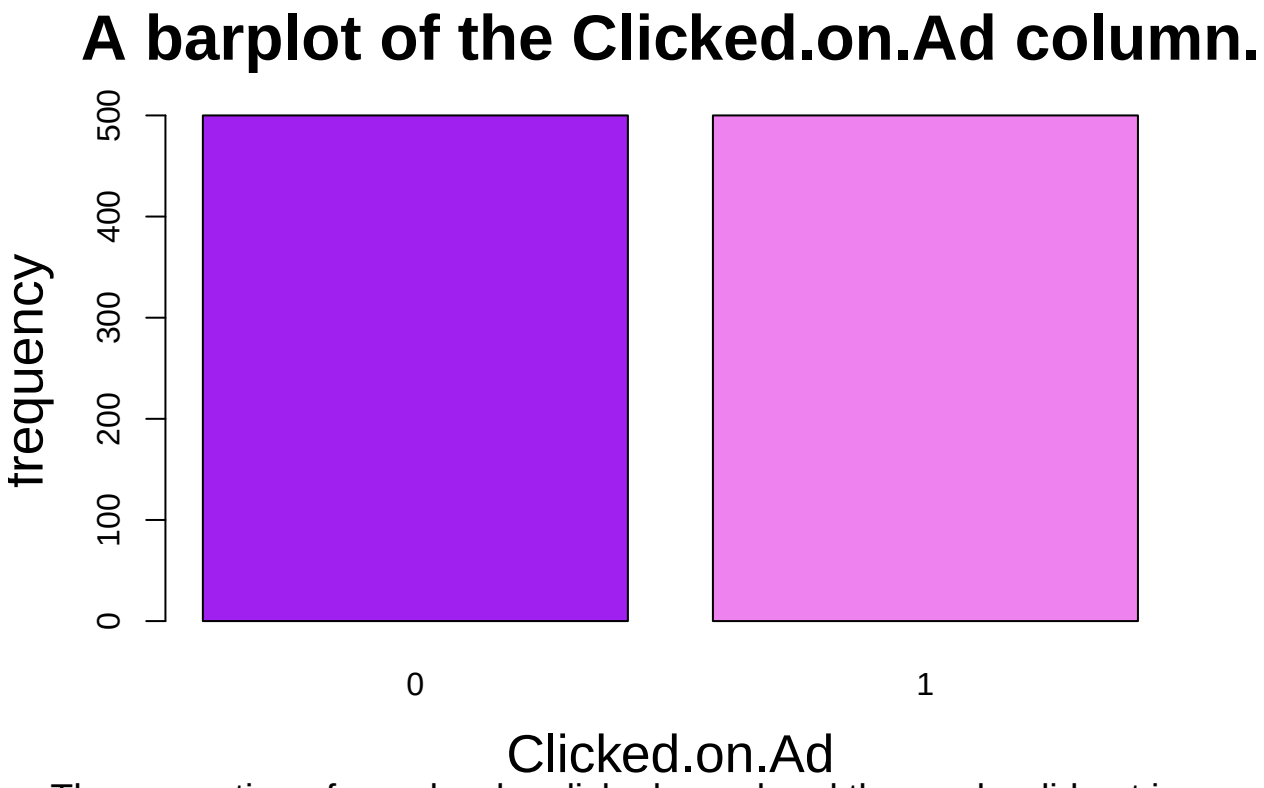
```
dim(clicked)
```

```
## [1] 500  10
```

```
#Frequency table of clicked on ad
Clicked.on.Ad_freq <- table(advert$Clicked.on.Ad)
Clicked.on.Ad_freq
```

```
##
##   0   1
## 500 500
```

```
#Bar graph to show frequency distribution of clicked on ad
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(Clicked.on.Ad_freq), main="A barplot of the Clicked.on.Ad column.",
        xlab="Clicked.on.Ad",
        ylab="frequency",
        sub="The proportion of people who clicked on ad and those who did not is equal.",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("purple","violet"))
```



The proportion of people who clicked on ad and those who did not is equa

```
#Frequency table of gender
gender_freq <- table(advert$Male)
gender_freq
```

```
##
##   0   1
## 519 481
```

From the graph we can see that females(0) are more than males(1)

```
#comparison of area income and clicked on ad
sort(table(clicked$Area.Income), decreasing = TRUE)[1:5]
```

```
##
##  13996.5 14548.06  14775.5 15598.29  15879.1
##        1        1        1        1        1
```

```
#comparison of age and clicked on ad
sort(table(clicked$Age), decreasing = TRUE)[1:5]
```

```
##
## 45 36 38 41 42
## 27 25 25 22 20
```

```
#comparison of country and clicked on ad
sort(table(clicked$Country), decreasing = TRUE)[1:5]
```

```
##
##     Australia      Ethiopia        Turkey       Liberia Liechtenstein
##             7             7             7             6             6
```

```
#comparison of city and clicked on ad
sort(table(clicked$City), decreasing = TRUE)[1:5]
```

```
##
##   Lake David   Lake James    Lisamouth Michelleside   Millerbury
##            2            2            2            2            2
```

```
#comparison of daily time spent on site and clicked on ad
sort(table(clicked$Daily.Time.Spent.on.Site), decreasing = TRUE)[1:5]
```

```
##
## 75.55  32.6 35.49 35.66 35.98
##     3     2     2     2     2
```

Feature Engineering

```
head(advert)
```

```
##   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
```

```
#dropping the year, country, city and ad topic line columns
advert$Ad.Topic.Line <- NULL
advert$City <- NULL
advert$Country <- NULL
advert$Year <- NULL
advert$Timestamp <- NULL
head(advert)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage Male
## 1                    68.95  35    61833.90               256.09    0
## 2                    80.23  31    68441.85               193.77    1
## 3                    69.47  26    59785.94               236.50    0
## 4                    74.15  29    54806.18               245.89    1
## 5                    68.37  35    73889.99               225.58    0
## 6                    59.99  23    59761.56               226.74    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

```
advert$Clicked.on.Ad =as.factor(advert$Clicked.on.Ad)
head(advert)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage Male
## 1                    68.95  35    61833.90               256.09    0
## 2                    80.23  31    68441.85               193.77    1
## 3                    69.47  26    59785.94               236.50    0
## 4                    74.15  29    54806.18               245.89    1
## 5                    68.37  35    73889.99               225.58    0
## 6                    59.99  23    59761.56               226.74    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

```
advert$Male <- as.numeric(as.character(advert$Male))
head(advert)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage Male
## 1                    68.95  35    61833.90               256.09    0
## 2                    80.23  31    68441.85               193.77    1
## 3                    69.47  26    59785.94               236.50    0
## 4                    74.15  29    54806.18               245.89    1
## 5                    68.37  35    73889.99               225.58    0
## 6                    59.99  23    59761.56               226.74    1
##   Clicked.on.Ad
```

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

```
# Normalizing the dataset so that no particular attribute
# has more impact on modeling algorithm than others.
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))
}
#data$Age<- normalize(data$Age)
advert$Area.Income<- normalize(advert$Area.Income)
advert$Daily.Internet.Usage<- normalize(advert$Daily.Internet.Usage)
advert$Daily.Time.Spent.on.Site<- normalize(advert$Daily.Time.Spent.on.Site)
advert$Male<- normalize(advert$Male)
advert$Age<- normalize(advert$Age)
head(advert)
```

```
##   Daily.Time.Spent.on.Site       Age Area.Income Daily.Internet.Usage Male
## 1                0.6178820 0.3809524   0.7304725            0.9160310    0
## 2                0.8096209 0.2857143   0.8313752            0.5387456    1
## 3                0.6267211 0.1666667   0.6992003            0.7974331    0
## 4                0.7062723 0.2380952   0.6231599            0.8542802    1
## 5                0.6080231 0.3809524   0.9145678            0.7313234    0
## 6                0.4655788 0.0952381   0.6988280            0.7383460    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

```
advert$Male <- NULL
head(advert)
```

```
##   Daily.Time.Spent.on.Site       Age Area.Income Daily.Internet.Usage
## 1                0.6178820 0.3809524   0.7304725            0.9160310
## 2                0.8096209 0.2857143   0.8313752            0.5387456
## 3                0.6267211 0.1666667   0.6992003            0.7974331
## 4                0.7062723 0.2380952   0.6231599            0.8542802
## 5                0.6080231 0.3809524   0.9145678            0.7313234
## 6                0.4655788 0.0952381   0.6988280            0.7383460
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

Decision Trees

```
install.packages("rattle")
```

```
## Installing package into 'C:/Users/hp/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

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

```
#Loading libraries
library(rpart,quietly = TRUE)
library(caret,quietly = TRUE)
library(rpart.plot,quietly = TRUE)
library(rattle)
```

```
## Loading required package: tibble
```

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

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

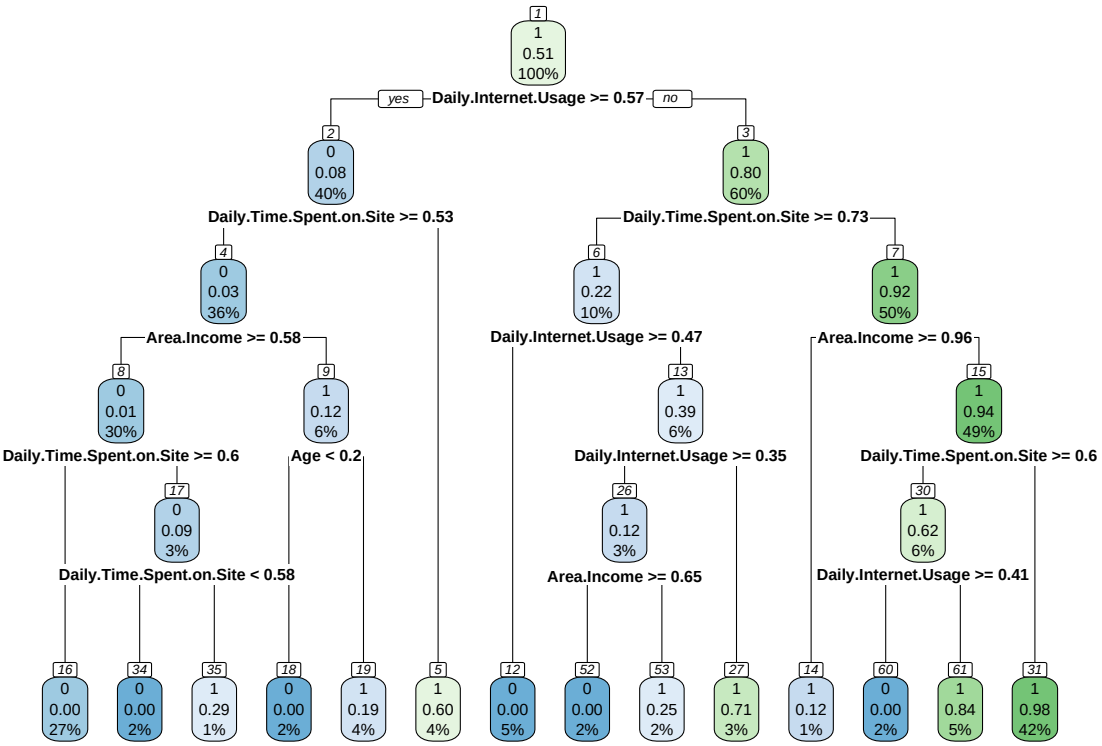
```
#data splicing
set.seed(123)
train <- sample(1:nrow(advert),size = ceiling(0.80*nrow(advert)),replace = FALSE)
# training set
ad_train <- advert[train,]
# test set
ad_test <- advert[-train,]
```

```
#Penalty matrix
penalty.matrix <- matrix(c(0, 1, 10,0), byrow = TRUE, nrow = 2)
#Building our model
tree <- rpart(Clicked.on.Ad ~., data = ad_train, parms=list(loss=penalty.matrix), method = 'class')
tree
```

```
## n= 800
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 389 1 (0.486250000 0.513750000)
##    2) Daily.Internet.Usage>=0.5662308 319 270 0 (0.915360502 0.084639498)
##      4) Daily.Time.Spent.on.Site>=0.5281319 289  90 0 (0.968858131 0.031141869)
##        8) Area.Income>=0.5787783 238  30 0 (0.987394958 0.012605042)
##         16) Daily.Time.Spent.on.Site>=0.6013089 215  10 0 (0.995348837 0.004651163) *
```

```
##         17) Daily.Time.Spent.on.Site< 0.6013089 23  20 0 (0.913043478 0.086956522)
##           34) Daily.Time.Spent.on.Site< 0.5802312 16   0 0 (1.000000000 0.000000000) *
##           35) Daily.Time.Spent.on.Site>=0.5802312 7   5 1 (0.714285714 0.285714286) *
##        9) Area.Income< 0.5787783 51  45 1 (0.882352941 0.117647059)
##         18) Age< 0.202381 19   0 0 (1.000000000 0.000000000) *
##         19) Age>=0.202381 32  26 1 (0.812500000 0.187500000) *
##      5) Daily.Time.Spent.on.Site< 0.5281319 30  12 1 (0.400000000 0.600000000) *
##    3) Daily.Internet.Usage< 0.5662308 481  97 1 (0.201663202 0.798336798)
##      6) Daily.Time.Spent.on.Site>=0.7324494 83  65 1 (0.783132530 0.216867470)
##       12) Daily.Internet.Usage>=0.4720002 37   0 0 (1.000000000 0.000000000) *
##       13) Daily.Internet.Usage< 0.4720002 46  28 1 (0.608695652 0.391304348)
##         26) Daily.Internet.Usage>=0.3478932 25  22 1 (0.880000000 0.120000000)
##           52) Area.Income>=0.6463641 13   0 0 (1.000000000 0.000000000) *
##           53) Area.Income< 0.6463641 12   9 1 (0.750000000 0.250000000) *
##         27) Daily.Internet.Usage< 0.3478932 21   6 1 (0.285714286 0.714285714) *
##      7) Daily.Time.Spent.on.Site< 0.7324494 398  32 1 (0.080402010 0.919597990)
##       14) Area.Income>=0.9611263 8   7 1 (0.875000000 0.125000000) *
##       15) Area.Income< 0.9611263 390  25 1 (0.064102564 0.935897436)
##         30) Daily.Time.Spent.on.Site>=0.6013089 50  19 1 (0.380000000 0.620000000)
##           60) Daily.Internet.Usage>=0.4080094 13   0 0 (1.000000000 0.000000000) *
##           61) Daily.Internet.Usage< 0.4080094 37   6 1 (0.162162162 0.837837838) *
##         31) Daily.Time.Spent.on.Site< 0.6013089 340   6 1 (0.017647059 0.982352941) *
```

```
#visualizing the tree
rpart.plot(tree, nn=TRUE)
```



```
#making predictions with our model
pred <- predict(object = tree, ad_test[,-6], type = 'class')
#calculating accuracy
t <- table(ad_test$Clicked.on.Ad, pred)
confusionMatrix(t)
```

```
## Confusion Matrix and Statistics
##
##    pred
##      0  1
##   0 88 23
##   1  1 88
##
##                Accuracy : 0.88
##                  95% CI : (0.8267, 0.9216)
##     No Information Rate : 0.555
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7629
##
##  Mcnemar's Test P-Value : 1.814e-05
##
##             Sensitivity : 0.9888
##             Specificity : 0.7928
##          Pos Pred Value : 0.7928
##          Neg Pred Value : 0.9888
##              Prevalence : 0.4450
##          Detection Rate : 0.4400
##    Detection Prevalence : 0.5550
##       Balanced Accuracy : 0.8908
##
##        'Positive' Class : 0
##
```

#8. Challenging the solution

SVM

```
library('caret')
intrain <- createDataPartition(y = advert$Clicked.on.Ad, p= 0.7, list = FALSE)
training <- advert[intrain,]
testing <- advert[-intrain,]
dim(training)
```

```
## [1] 700   5
```

```
dim(testing)
```

```
## [1] 300   5
```

```r
#building our model
#
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
svm_Linear <- train(Clicked.on.Ad ~., data = training, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)
svm_Linear
```

```
## Support Vector Machines with Linear Kernel
##
## 700 samples
##   4 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 630, 630, 630, 630, 630, 630, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9671429  0.9342857
##
## Tuning parameter 'C' was held constant at a value of 1
```

```r
#making predictions
test_pred <- predict(svm_Linear, newdata = testing)
test_pred
```

```
##   [1] 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0
##  [38] 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 1 1
##  [75] 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0
## [112] 0 0 1 0 0 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0 1 1 0 1 1 0 0 1
## [149] 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 0 0
## [186] 1 1 1 0 1 0 0 0 1 0 0 0 1 1 0 1 1 1 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 1
## [223] 1 1 1 1 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 0 0 0 0
## [260] 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 0
## [297] 1 1 1 1
## Levels: 0 1
```

```r
#checking accuracy of model
confusionMatrix(table(test_pred, testing$Clicked.on.Ad))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred   0   1
##         0 146   8
##         1   4 142
##
##                Accuracy : 0.96
##                  95% CI : (0.9312, 0.9792)
```

```
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : 0.3865
##
##             Sensitivity : 0.9733
##             Specificity : 0.9467
##          Pos Pred Value : 0.9481
##          Neg Pred Value : 0.9726
##              Prevalence : 0.5000
##          Detection Rate : 0.4867
##    Detection Prevalence : 0.5133
##       Balanced Accuracy : 0.9600
##
##        'Positive' Class : 0
##
```

```
#Hyperparameter tuning
grid <- expand.grid(C = c(0,0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2,5))
svm_Linear_Grid <- train(Clicked.on.Ad ~., data = training, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneGrid = grid,
tuneLength = 10)
```

```
## Warning: model fit failed for Fold01.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold02.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold03.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold04.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold05.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold06.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold07.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold08.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters
```

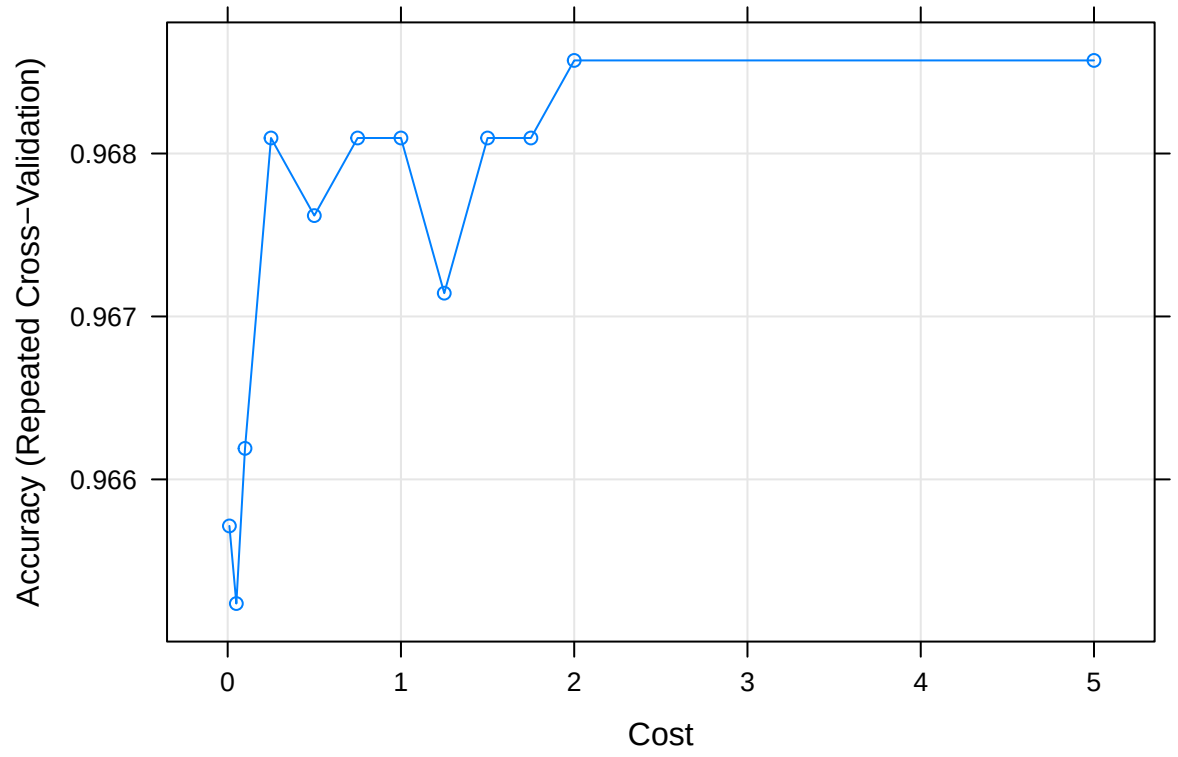
```
## Warning: model fit failed for Fold09.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold10.Rep1: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold01.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold02.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold03.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold04.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold05.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold06.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold07.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold08.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold09.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold10.Rep2: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold01.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold02.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold03.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold04.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold05.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters
```

```
## Warning: model fit failed for Fold06.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold07.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold08.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold09.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold10.Rep3: C=0.00 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.

## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
svm_Linear_Grid
```

```
## Support Vector Machines with Linear Kernel
##
## 700 samples
##   4 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 630, 630, 630, 630, 630, 630, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.00       NaN       NaN
##   0.01  0.9657143  0.9314286
##   0.05  0.9652381  0.9304762
##   0.10  0.9661905  0.9323810
##   0.25  0.9680952  0.9361905
##   0.50  0.9676190  0.9352381
##   0.75  0.9680952  0.9361905
##   1.00  0.9680952  0.9361905
##   1.25  0.9671429  0.9342857
##   1.50  0.9680952  0.9361905
##   1.75  0.9680952  0.9361905
##   2.00  0.9685714  0.9371429
##   5.00  0.9685714  0.9371429
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 2.
```

```
plot(svm_Linear_Grid)
```




```
#Making predictions with the model after tuning.
test_pred_grid <- predict(svm_Linear_Grid, newdata = testing)
test_pred_grid
```

```
##   [1] 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0
##  [38] 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 1 1
##  [75] 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0
## [112] 0 0 1 0 0 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0 1 1 0 1 1 0 0 1
## [149] 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 0 0
## [186] 1 1 1 0 1 0 0 0 1 0 0 0 1 1 0 1 1 1 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 1
## [223] 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 0 0 0 0
## [260] 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 0
## [297] 1 1 1 1
## Levels: 0 1
```

```
#checking the accuracy
confusionMatrix(table(test_pred_grid, testing$Clicked.on.Ad))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred_grid   0   1
```

```
##              0 145   8
##              1   5 142
##
##                Accuracy : 0.9567
##                  95% CI : (0.927, 0.9767)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9133
##
##  Mcnemar's Test P-Value : 0.5791
##
##             Sensitivity : 0.9667
##             Specificity : 0.9467
##          Pos Pred Value : 0.9477
##          Neg Pred Value : 0.9660
##              Prevalence : 0.5000
##          Detection Rate : 0.4833
##    Detection Prevalence : 0.5100
##       Balanced Accuracy : 0.9567
##
##        'Positive' Class : 0
##
```

Conclusion

- The age and gender do not determine whether an individual clicks on an ad. This is probably because their interests on the internet are different from what the ad is about.
- Daily time spent on a site has a negative correlation on whether an individual clicks on an ad probably because they are already on the site and are aware of what the ad is about.
- The model created using SVM performs better with an accuracy of 95.6% than the one created using decision trees which has an accuracy of 88.5%.
- Hyperparameter tuning doesn't do much in improving the svm model performance.
- We achieved our metric of success since both our models achieved an accuracy score of above 85%.

Recommendations

- More resorces should be chanelled towards maximizing the ad clicks gotten at 9am and during the month of February as these are the times with the highest number of ad clicks.
- Ads that are more appealing could be created so as to increase the ad clicks from men.
- We recommend the use of the SVM model in making predictions as it achieved the highest accuracy score of 95.6%.

##9. Follow up questions

###a) Did we have the right data? `Yes we did. Our data set had a good number of variables that helped us study the individuals and determine who was likely to click on an ad..`
###b) Do we need other data to answer our question? `Not necessarily, however further research is needed to help gain deeper insight on the study.` ###c) Did we have the right question?

The question was to create a model that consistently and accurately predicted whether an individual was most likely to click on an ad. We were able to do that by analysing the given dataset.