Predicting customer clicks an ad

Victoria Maina

2022-03-29

Analysis to identify which individuals are most likely to click on ads when advertising on an online cryptography course

By Victoria Maina

1.Defining the Question

a) Specifying the Question

- To Find and deal with outliers, anomalies, and missing data within the dataset. Perform univariate and bivariate analysis using R
- To identify which individuals are most likely to click on her ads.

b) Defining the Metric for Success

This project will be successful when:

- When i identify which individuals are most likely to click on her ads.
- We will then create various classification models to predict which individuals will click on ads. We use confusion matrix and accuracy score as our metrics of success

c) Understanding the context

d) Recording the Experimental Design The following steps were taken:

1. Business Understanding
2. Reading the data
3. Checking our data
4. Data cleaning
5. Performing EDA(univariate,bivariate and multivariate analysis)
6. Conclusion

e) Data Relevance

Importing libraries

```r
# Importing libraries
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## 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

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

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

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

```r
library(kernlab)
```

```
##
## Attaching package: 'kernlab'

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(ggplot2)
library(ggcorrplot)
library(dplyr)
library(moments)
library(tinytex)
library(earth)
```

```
## Loading required package: Formula

## Loading required package: plotmo

## Loading required package: plotrix

## Loading required package: TeachingDemos
```

```r
library(Formula)
library(plotmo)
library(rpart)
library(plotrix)
library(purrr)
library(TeachingDemos)
library(prodlim)
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```r
library(iterators)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(gower)
library(numDeriv)
library(SQUAREM)
library(lava)
```

```
##
## Attaching package: 'lava'

## The following object is masked from 'package:dplyr':
##
##     vars

## The following object is masked from 'package:ggplot2':
##
##     vars
```

```
library(ipred)
```

```
##
## Attaching package: 'ipred'

## The following object is masked from 'package:lava':
##
##     cv
```

```
library(timeDate)
```

```
##
## Attaching package: 'timeDate'

## The following objects are masked from 'package:moments':
##
##     kurtosis, skewness
```

```
library(foreach)
```

```
##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```r
library(ModelMetrics)
```

```
##
## Attaching package: 'ModelMetrics'

## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, recall, sensitivity, specificity

## The following object is masked from 'package:base':
##
##     kappa
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:data.table':
##
##     dcast, melt

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(recipes)
```

```
##
## Attaching package: 'recipes'

## The following object is masked from 'package:stringr':
##
##     fixed

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

```r
library(plyr)
theme_set(theme_classic())
options(warn = -1)
```

2. Reading the Data

Loading the dataset

```r
advertising<-read.csv('http://bit.ly/IPAdvertisingData')
df<-advertising
```

3. Data Understanding

checking for first 5 rows

```r
head(df)
```

```
##   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
```

checking for last 5 rows

```r
tail(df)
```

```
##      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 data types

```
str(df)
```

```
## '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 ...
```

```
library(dplyr)
glimpse(df)
```

```
## Rows: 1,000
## Columns: 10
## $ Daily.Time.Spent.on.Site <dbl> 68.95, 80.23, 69.47, 74.15, 68.37, 59.99, 88.~
## $ Age                      <int> 35, 31, 26, 29, 35, 23, 33, 48, 30, 20, 49, 3~
## $ Area.Income              <dbl> 61833.90, 68441.85, 59785.94, 54806.18, 73889~
## $ Daily.Internet.Usage     <dbl> 256.09, 193.77, 236.50, 245.89, 225.58, 226.7~
## $ Ad.Topic.Line            <chr> "Cloned 5thgeneration orchestration", "Monito~
## $ City                     <chr> "Wrightburgh", "West Jodi", "Davidton", "West~
## $ Male                     <int> 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, ~
## $ Country                  <chr> "Tunisia", "Nauru", "San Marino", "Italy", "I~
## $ Timestamp                <chr> "2016-03-27 00:53:11", "2016-04-04 01:39:02",~
## $ Clicked.on.Ad            <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, ~
```

A description of the dataset

```
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  
##  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
```

```
class(df)
```

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

4.0 Data Cleaning

4.1 Completeness

```
# checking for the sum of missing values in each column

colSums(is.na(df))
```

```
## 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 are no missing values within our dataset.

4.2 Consistency

```
# checking for duplicates
duplicated_rows <- colSums(df[duplicated(df),])
duplicated_rows
```

```
## 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 no duplicates in the dataset

4.3 Uniformity

```r
# Changing the column namesto lower case
names(df) <- tolower(names(df))
names(df)
```

```
##  [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"
```

```r
library(stringr)
colnames(df) = str_replace_all(colnames(df), c(' ' = '_'))
colnames(df)
```

```
##  [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"
```

Checking for duplicates

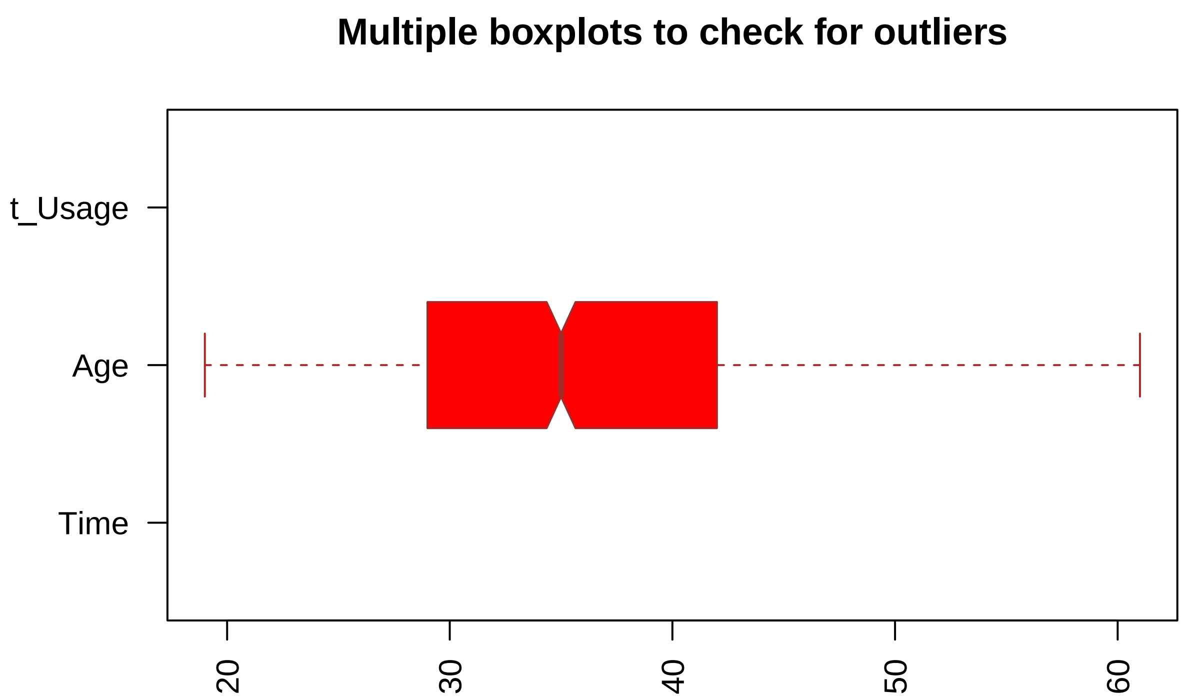
```r
anyDuplicated(df)
```

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

There are no duplicates in the dataset.

```r
# Using a boxplot to check for observations far away from other data points.
# We will Use all three double type columns: specifying each


daily_time_spent_on_site <- df$ daily_time_spent_on_site
age <- df$age
daily_internet_usage <- df$daily_internet_usage
area_income <- df$area_income
```

```
boxplot(daily_time_spent_on_site,age, daily_internet_usage,


main = "Multiple boxplots to check for outliers",
at = c(1,2,3),
names = c("Time", "Age","Iternet_Usage"),
las = 2,
col = c("orange","red","blue"),
border = "brown",
horizontal = TRUE,
notch = TRUE
)
```



The Daily_Time_Spent_on_Site,Age, Daily_Internet_Usage variables do not seem to have any outliers.

5.0 Exploratory Data Analysis

5.1 Univariate Analysis

```


numeric_columns = c("daily_time_spent_on_site", "age", "area_income", "daily_internet_usage",  "male",
mean(df$daily.time.spent.on.site)
```

5.1.1 Mean of Numeric Columns

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

```
mean(df$area.income)
```

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

```
mean(df$age)
```

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

```
mean(df$male)
```

```
## [1] 0.481
```

```
mean(df$daily.internet.usage)
```

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

The mean of daily time spent on site is 65.0002

the mean of age is 36.009

the mean of area income is 55000

the mean of male column is 0.481

the mean of internet usage column is 180.001 #### 5.1.2 Mode of Numeric Columns

```
# We create the mode function that will perform our mode operation for us
# ---
#
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

getmode(df$daily.time.spent.on.site)
```

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

```
getmode(df$age)
```

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

```
getmode(df$area.income)
```

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

```r
getmode(df$daily.internet.usage)
```

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

```r
getmode(df$male)
```

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

```r
getmode(df$timestamp)
```

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

mode of daily time spent on site is 62.26

mode of age is 31

mode of area income is 61833.9

mode of daily internet usage is 167.22

mode of male is 0

mode of time stamp column is “2016-03-27 00:53:11 UTC”

```r
median(df$daily.time.spent.on.site)
```

5.1.3 Median of the numerical columns

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

```r
median(df$age)
```

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

```r
median(df$area.income)
```

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

```r
median(df$daily.internet.usage)
```

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

```r
median(df$male)
```

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

median of daily time spent on site is 68.215

median of age is 35

median of area income is 57012.3

median of daily internet usage is 183.13

median of male is 0

```r
range(df$daily.time.spent.on.site)
```

5.1.4 Ranges of Numeric Columns

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

```r
range(df$age)
```

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

```r
range(df$area.income)
```

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

```r
range(df$daily.internet.usage)
```

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

```r
range(df$male)
```

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

```r
sd(df$daily.time.spent.on.site)
```

5.1.5 Standard Deviations of Numeric Columns

```
## [1] 15.85361
```

```r
sd(df$age)
```

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

```r
sd(df$area.income)
```

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

```r
sd(df$daily.internet.usage)
```

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

```r
sd(df$male)
```

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

```r
var(df$daily.time.spent.on.site)
```

5.1.6 Variance of the numerical cols

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

```r
var(df$age)
```

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

```r
var(df$area.income)
```

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

```r
var(df$daily.internet.usage)
```

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

```r
var(df$male)
```

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

```r
quantile(df$daily.time.spent.on.site)
```

5.1.7 Quantiles of Numeric Columns

```
##      0%     25%     50%     75%    100% 
## 32.6000 51.3600 68.2150 78.5475 91.4300
```

```r
quantile(df$age)
```

```
##   0%  25%  50%  75% 100% 
##   19   29   35   42   61
```

```r
quantile(df$area.income)
```

```
##       0%      25%      50%      75%     100% 
## 13996.50 47031.80 57012.30 65470.64 79484.80
```

```
quantile(df$daily.internet.usage)
```

```
##        0%      25%      50%      75%     100%
## 104.7800 138.8300 183.1300 218.7925 269.9600
```

```
quantile(df$male)
```

```
##   0%  25%  50%  75% 100%
##    0    0    0    1    1
```

```
skewness(df$daily.time.spent.on.site)
```

5.1.8 Skewness

```
## [1] -0.370646
## attr(,"method")
## [1] "moment"
```

```
skewness(df$age)
```

```
## [1] 0.4777052
## attr(,"method")
## [1] "moment"
```

```
skewness(df$area.income)
```

```
## [1] -0.6484229
## attr(,"method")
## [1] "moment"
```

```
skewness(df$daily.internet.usage)
```

```
## [1] -0.03343681
## attr(,"method")
## [1] "moment"
```

```
skewness(df$male)
```

```
## [1] 0.07594088
## attr(,"method")
## [1] "moment"
```

male,time stamp and age column are positively skewed while as time spent on a site ,area income and daily internet usage are negatively skewed.

```r
kurtosis(df$daily.time.spent.on.site)
```

kurtosis

```
## [1] -1.099864
## attr(,"method")
## [1] "excess"
```

```r
kurtosis(df$age)
```

```
## [1] -0.4097066
## attr(,"method")
## [1] "excess"
```

```r
kurtosis(df$area.income)
```

```
## [1] -0.1110924
## attr(,"method")
## [1] "excess"
```

```r
kurtosis(df$daily.internet.usage)
```

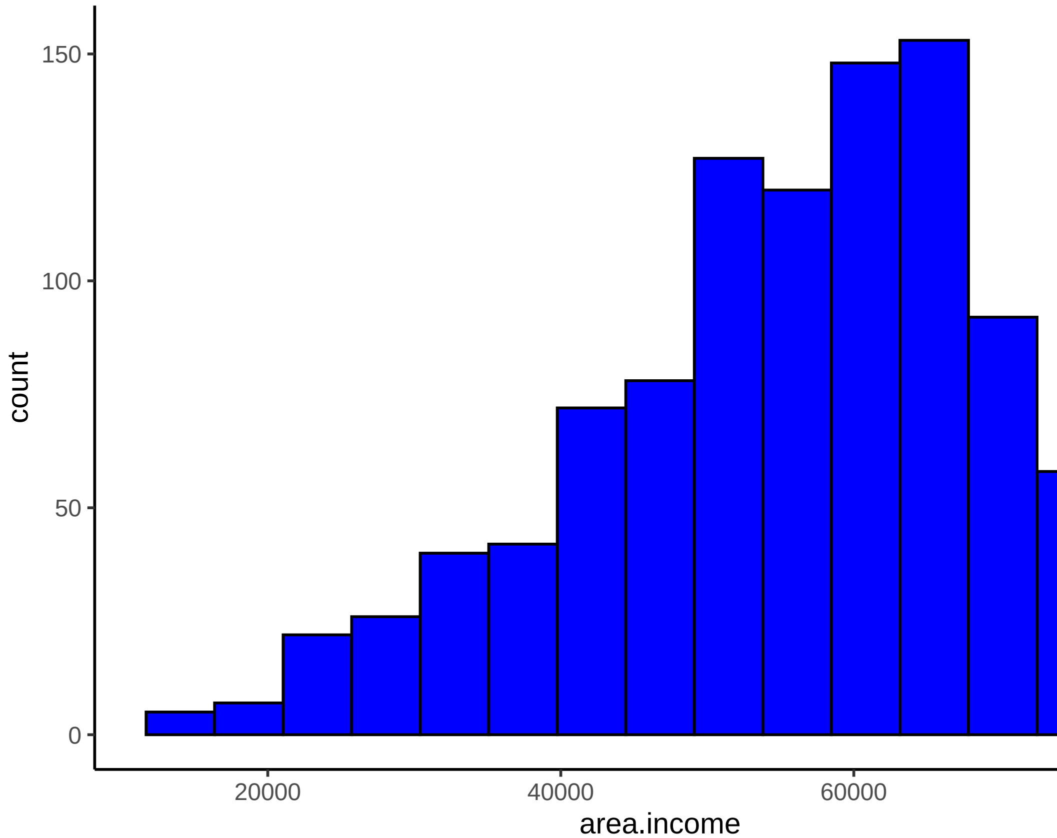
```
## [1] -1.275752
## attr(,"method")
## [1] "excess"
```

```r
kurtosis(df$male)
```

```
## [1] -1.996226
## attr(,"method")
## [1] "excess"
```

the data has a platykurtic distribution

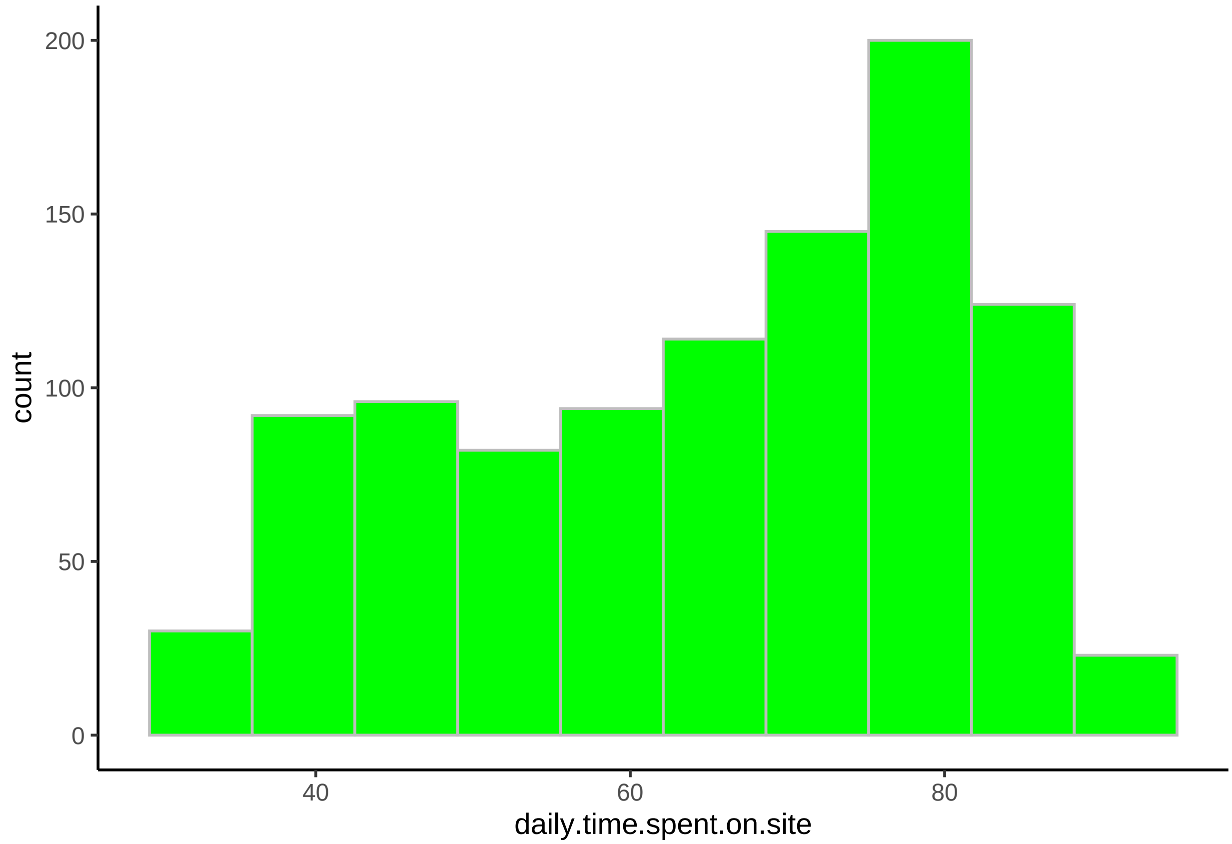
```r
# Histogram with density plot

ggplot(df, aes(x=area.income)) +
 geom_histogram(colour="black", fill="blue",bins=15)#+
```



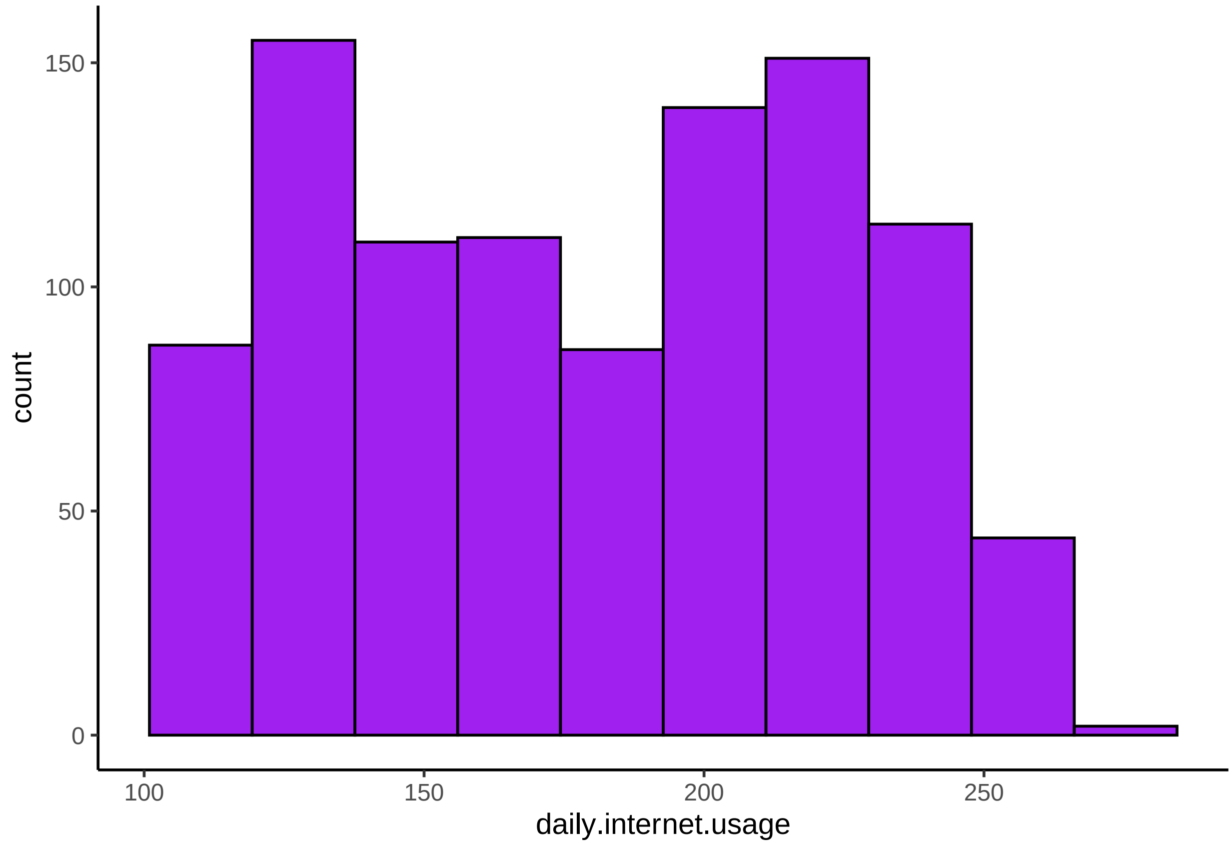
Histograms and Bar Chart

shows that most people receive incomes ranges between 60,000 and 70,000

```
# Histogram with density plot
ggplot(df, aes(x=daily.time.spent.on.site)) +
 geom_histogram(colour="grey", fill="green",bins=10)#+
```

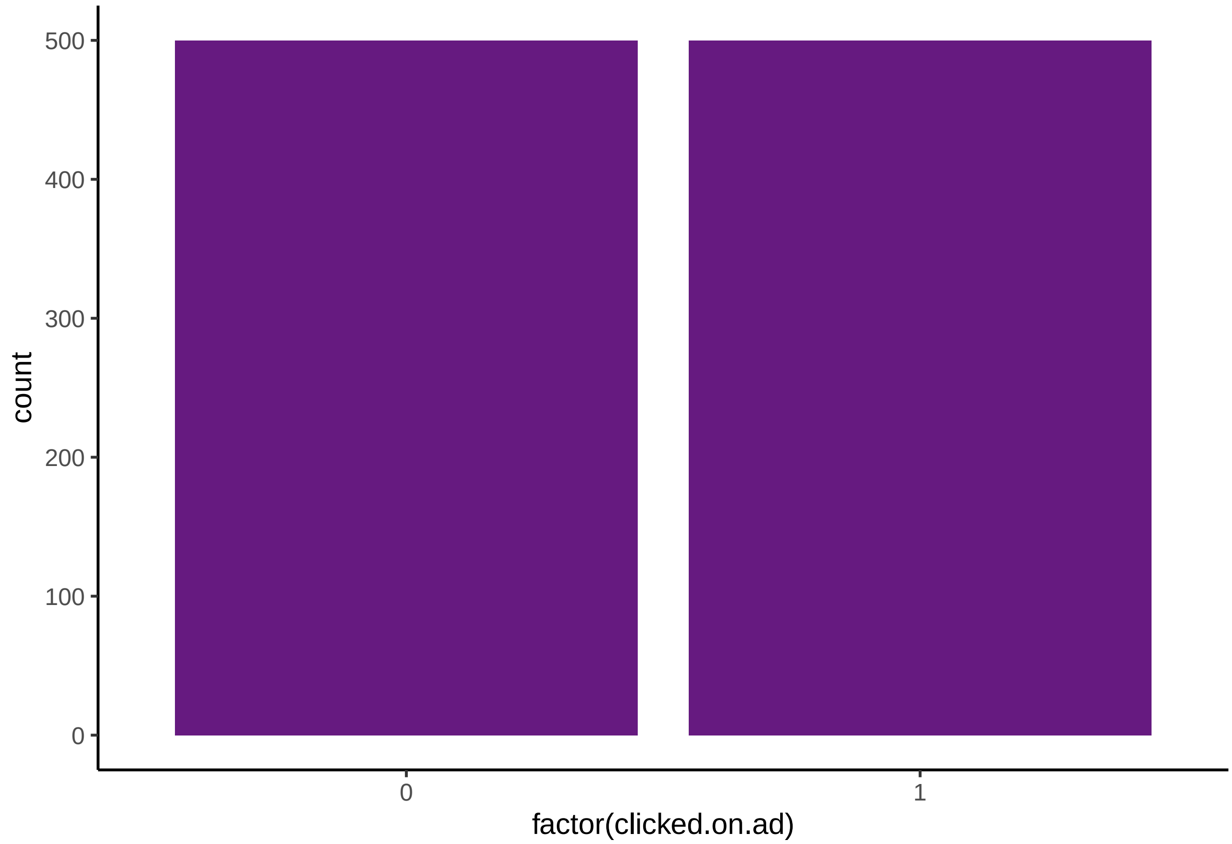



```
# Histogram with density plot
ggplot(df, aes(x=daily.internet.usage)) +
 geom_histogram(colour="black", fill="purple",bins=10)#+
```



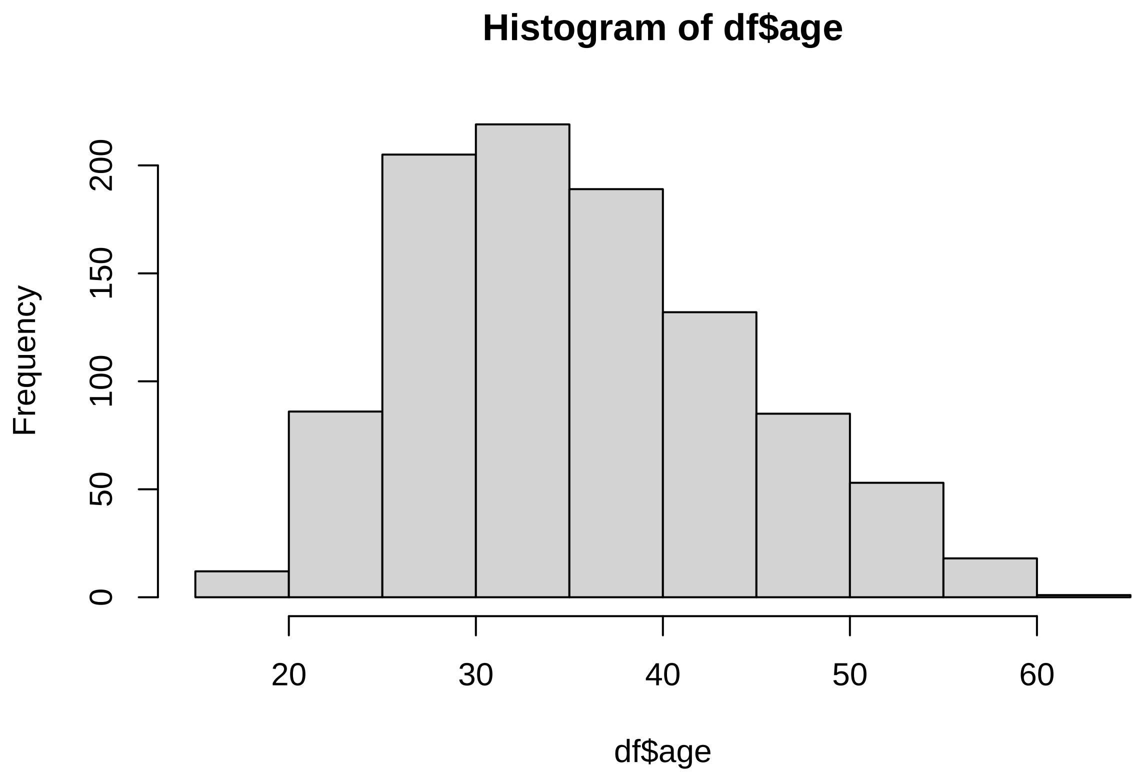
The Average Hours spent by users on the Internet is 180 minutes

```
ggplot(df, aes(x=factor(`clicked.on.ad`))) + geom_bar( fill=rgb(0.4,0.1,0.5))
```



The number of users on the site who clicked on the ad is equal to those that did not

```
# Creating a histogram for age
hist(df$age,)
```



Majority of the users are between the age 25 to 35.

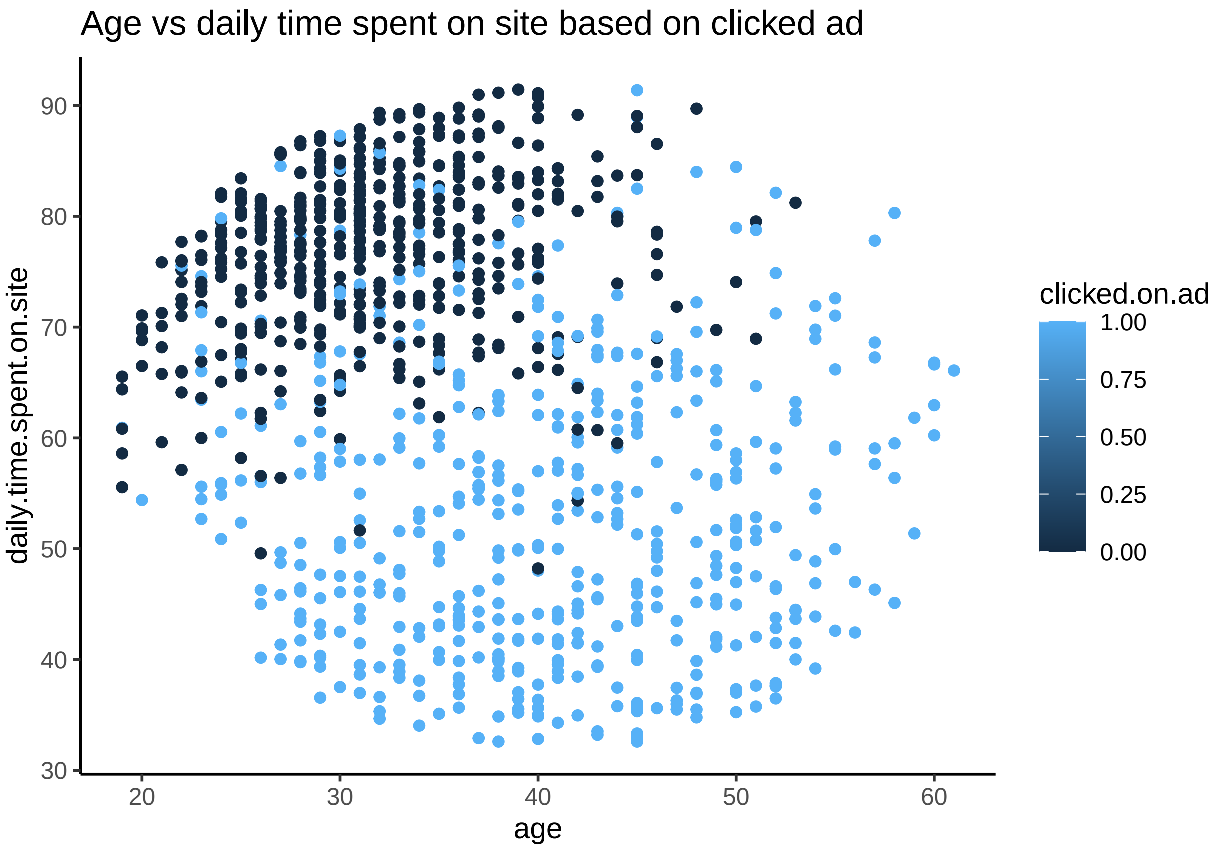
6.0 Bivariate analysis

```
ggplot(df, aes(x=area.income, y = daily.time.spent.on.site )) +  geom_point(aes(colour= as.factor(`clic
  labs(title="Area income vs daily time spent on site based on clicked ad")
```

6.1 Scatter Plots

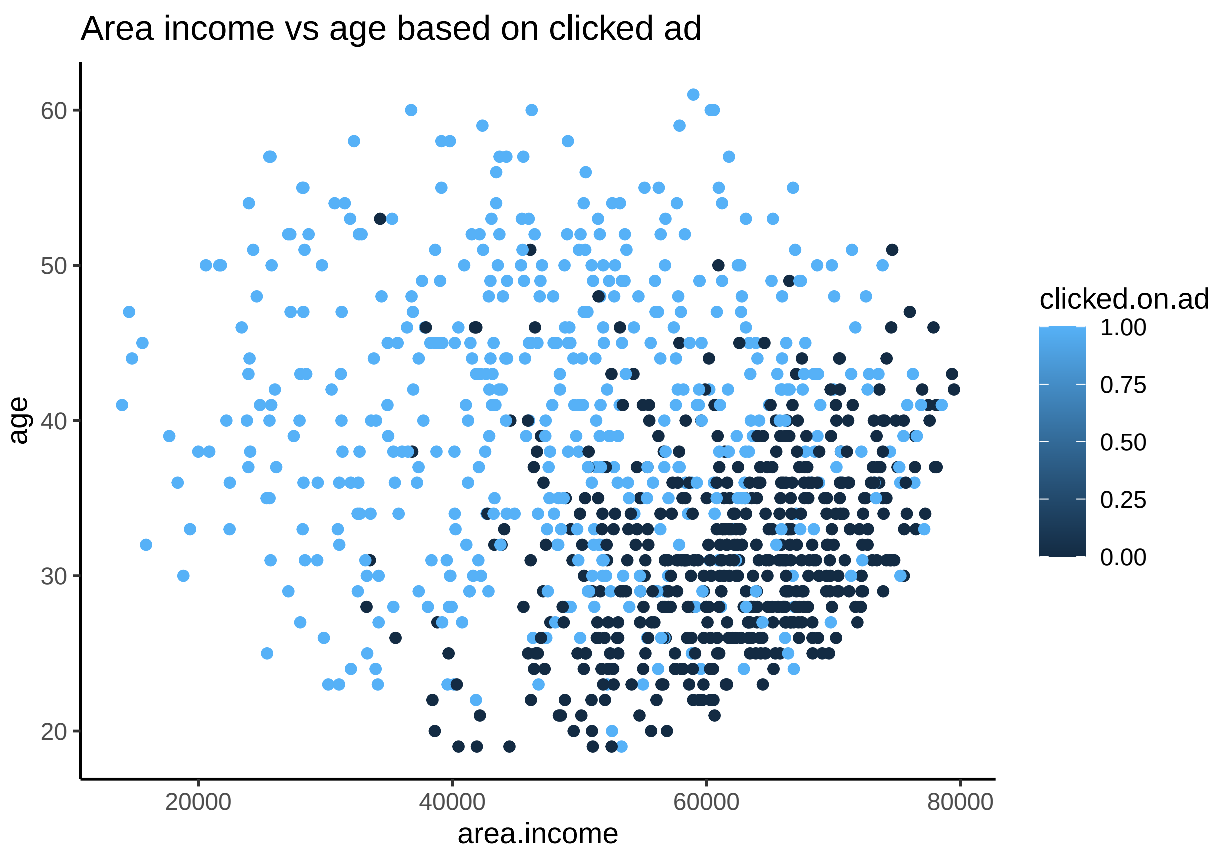
The scatter plot for the area _income against time spent on the site shows that high income earners were least likely to click on the ad despite the fact that they seemed to spend a over an hour a day on the site.

```
ggplot(data=df, aes(x=age, y=daily.time.spent.on.site))+
  geom_point(aes(color=clicked.on.ad))+
  labs(title="Age vs daily time spent on site based on clicked ad")
```



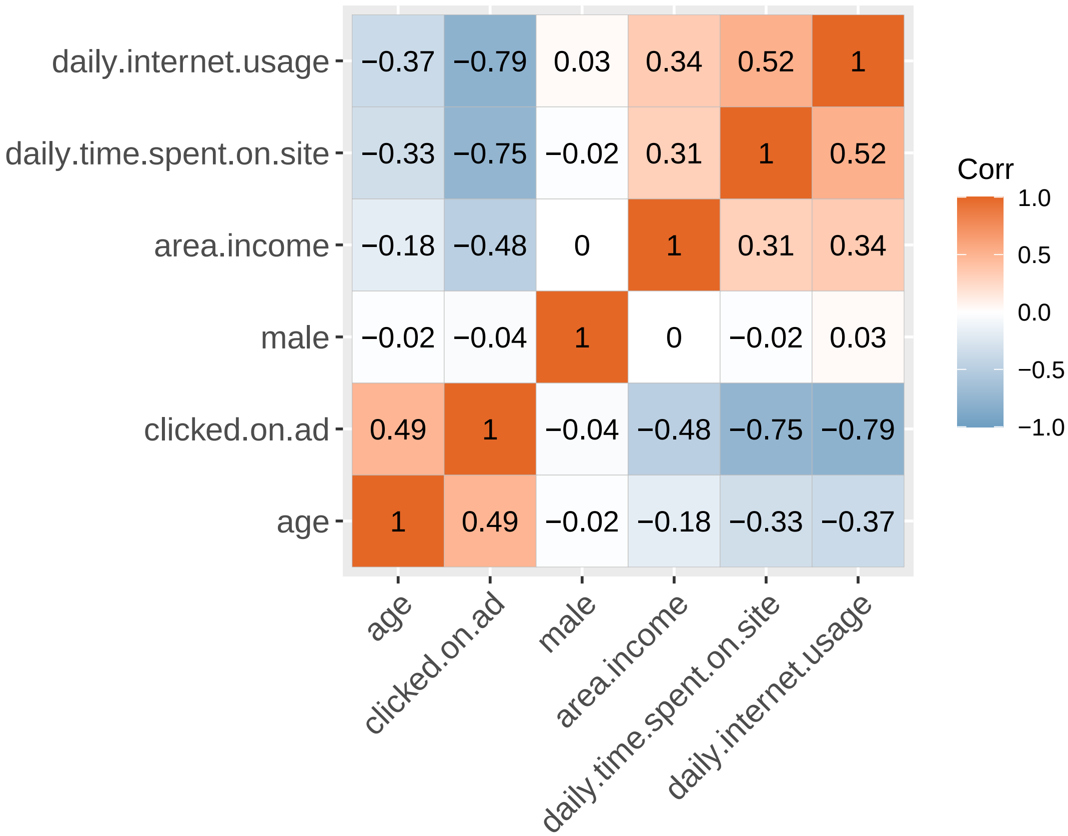
the Age against Time spent on the site show that the younger demographic are less tolerant to ads since are more likely to detect ads and avoid them while using the internet compared to their older counterparts

```
ggplot(data=df, aes(x=area.income, y=age))+
  geom_point(aes(color=clicked.on.ad))+
  labs(title="Area income vs age based on clicked ad")
```



The scatter plot for the area_income against Age showed that ,majority of the users who did not click on the ad were the high income earners and many were aged between 20 and 40 years.

```
corr = round(cor(select_if(df, is.numeric)), 2)
ggcorrplot(corr, hc.order = T, ggtheme = ggplot2::theme_gray,
   colors = c("#6D9EC1", "white", "#E46726"), lab = T)
```



6.2 Heat map

7.0 Conclusion

- The factors that seem to contribute the most to the click add activity are "daily_internet_usage","daily_time_spent_on
  and "area_income".
- area income showed a moderate negative relationship with click ad activity, where most click activity happened with those that earned above 40,000. However, earners from 66,000 less clicked on the ad.
- The people who clicked most on Ads were between age 28 to 43.
- Older people , those over 35 were more likely to click on the course ad.

8.0 Recommendations

- target users who were aged over 35 , as they were more likely to click on the ad.
- More focus should be on those earning a lower income i.e less than 60,000 because their indicate to be more beneficial as these consumers clicking on the ad .
- Finally the users who spend less time on the site and on the internet are more likely to click on the ads

Modelling

```r
df$clicked.on.ad <- as.factor(df$clicked.on.ad)
```

```r
df$clicked.on.ad <- as.numeric(df$clicked.on.ad)
```

```r
head(df)
```

```
##   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             1
## 2 2016-04-04 01:39:02             1
## 3 2016-03-13 20:35:42             1
## 4 2016-01-10 02:31:19             1
## 5 2016-06-03 03:36:18             1
## 6 2016-05-19 14:30:17             1
```

```r
df1 <- select(df, c(1,2,3,4,7,10))
#df1 <- select(df1, -c(7,8))
head(df1)
```

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

```r
#Create an index for data partitioning

set.seed(7)
library(caret)
index<- createDataPartition(df1$clicked.on.ad,p = 0.8 ,list = FALSE)
```

Splitting the data

```
#Using the indexes to split data into test and train set
df.train <- df1[index, ]
df.test <- df1[-index, ]
```

Decision Trees

```
#Fitting in the decision tree
TreeFit <- rpart(clicked.on.ad ~ ., data = df.train ,method = "class")

#Factor the Clicked.on.Ad vector in the test dataset
df.test$clicked.on.ad <- factor(df.test$clicked.on.ad)

#Using model to predict
TreePredict <- predict(TreeFit, newdata = df.test, type = "class")
confusionMatrix(TreePredict, df.test$clicked.on.ad)
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0  103
```

KNN

```
#Fitting model to training dataset
#Also we scale and center our data
knnModel <- train(clicked.on.ad ~ ., data = df.train, method = "knn", preProcess = c("center", "scale"))

#Using the model to predict
knnPredict <- predict(knnModel, newdata = df.test)

#Printing out the confusion matrix and statistics
confusionMatrix(knnPredict, df.test$clicked.on.ad)
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0   79
```

We can see both decision tree and knn have been correctly classified and have