Ad Click Prediction

Ankush Morey

2/27/2021

Contents

Introduction

Overview

Advertisements are really important in today's world and with increasing penetration of digital media. With the popularity of the Internet, businesses can put ads on the Internet, people can choose to watch, for the website and business mutual benefit. It is very interesting to see how people would like to spend their time on the internet every day and what kind of content could attract different age group, income level and location. From the analysis of how people interact with advertising on interest, businesses could create better ad to develop their companies.

Business Problem

We will predict which users are more likely to click an advertisement based on user's demographic data and features of advertisement.

Dataset Overview

The data is from public data recourse: https://www.kaggle.com/fayomi/advertising.

The dataset structure demonstrates that it is included by 1000 observations and 10 variables. The variables are the demographic data that describe the users as below:

- Daily.Time.Spend.On.Site: It is the total time (in minutes) that the user spends on the website (continuous numeric data)

- Age: Age of the users in years (continuous numeric data)
- Area.Icome: Income of the area where a user visited the website in US$ (continuous numeric data)
- Daily.Internet.Usage: Amount of time spent on the internet usage in minutes (continuous numeric data)
- Ad.Topic.Line: Title of the advertisements that popped up on the website (Character data)
- City: City of the user (Character data)
- Male: Categorical data whether the user is male or not (1 means user is a male and 0 means otherwise)
- Country: Region (Country) of the user (Character data)
- Timestamp: Timestamp when the user clicked the advertisement (date-time)
- Clicked.on.Ad: Categorical data if the user actually clicked on the website (1 = yes and 0 otherwise)

Approach

- We will divide the business problem into two major parts.
- We will conduct data cleaning and perform an exploratory data analysis to generate some insights on the clicking behavior of the users
- We will count the frequency of most repeating words from the advertisement title. These keywords should be the area of focus for designing the advertisement titles
- We will build features to enhance the predicting capability of the model
- We plan to develop Logistic Regression, Naïve Bayes, and Decision Tree for this classification problem
- We will be using the accuracy as a metric for model comparison

Exploratory Data Analysis

Import dataset and libraries

```
#Loading the dataset in our working environment and importing libraries
ad <- read.csv('C:\\Users\\ankus\\Desktop\\Ankush\\BANA\\Statistical Computing\\Project\\advertising.csv
library("ggplot2")
library("tidyverse")
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("RCurl")
library("XML")
library("OneR")
library("corrplot")
library("gridExtra")
library("lubridate")
library('caret')
library('rpart')
library('rpart.plot')
```

Data structure Analysis

```
# Analyzing the structure of dataframe and types of variables present
head(ad)
```

```
##   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
```

```
str(ad)
```

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

```
ad$Male <- as.factor(ad$Male)
str(ad$Male)
```

```
##  Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 2 2 2 ...
```

```
ad$Clicked.on.Ad <- as.factor(ad$Clicked.on.Ad)
str(ad$Clicked.on.Ad)
```

```
##  Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
```

```
# After using the str() function, we can observe that our dataframe consists of
# 1,000 observations in 10 variables

summary(ad)
```

```
##  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        0:519   Length:1000
##  Class :character   Class :character   1:481   Class :character
##  Mode  :character   Mode  :character           Mode  :character
##
##
##
##   Timestamp         Clicked.on.Ad
##  Length:1000        0:500
##  Class :character   1:500
##  Mode  :character
##
##
##
```

Missing Value treatment

```
sum(is.na(ad))
```

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

```
# Checking how many users actually clicked on the advertisements
sum(ad$Clicked.on.Ad == 1)
```

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

```
# Therefore 500 users out of 1,000 have actually selected the advertisements
```

Duplicate Records check

```
# Checking duplicate values in Ad Topic Line
dup_title <- ad$Ad.Topic.Line[duplicated(ad$Ad.Topic.Line)]
str(dup_title)
```

```
##  chr(0)
```

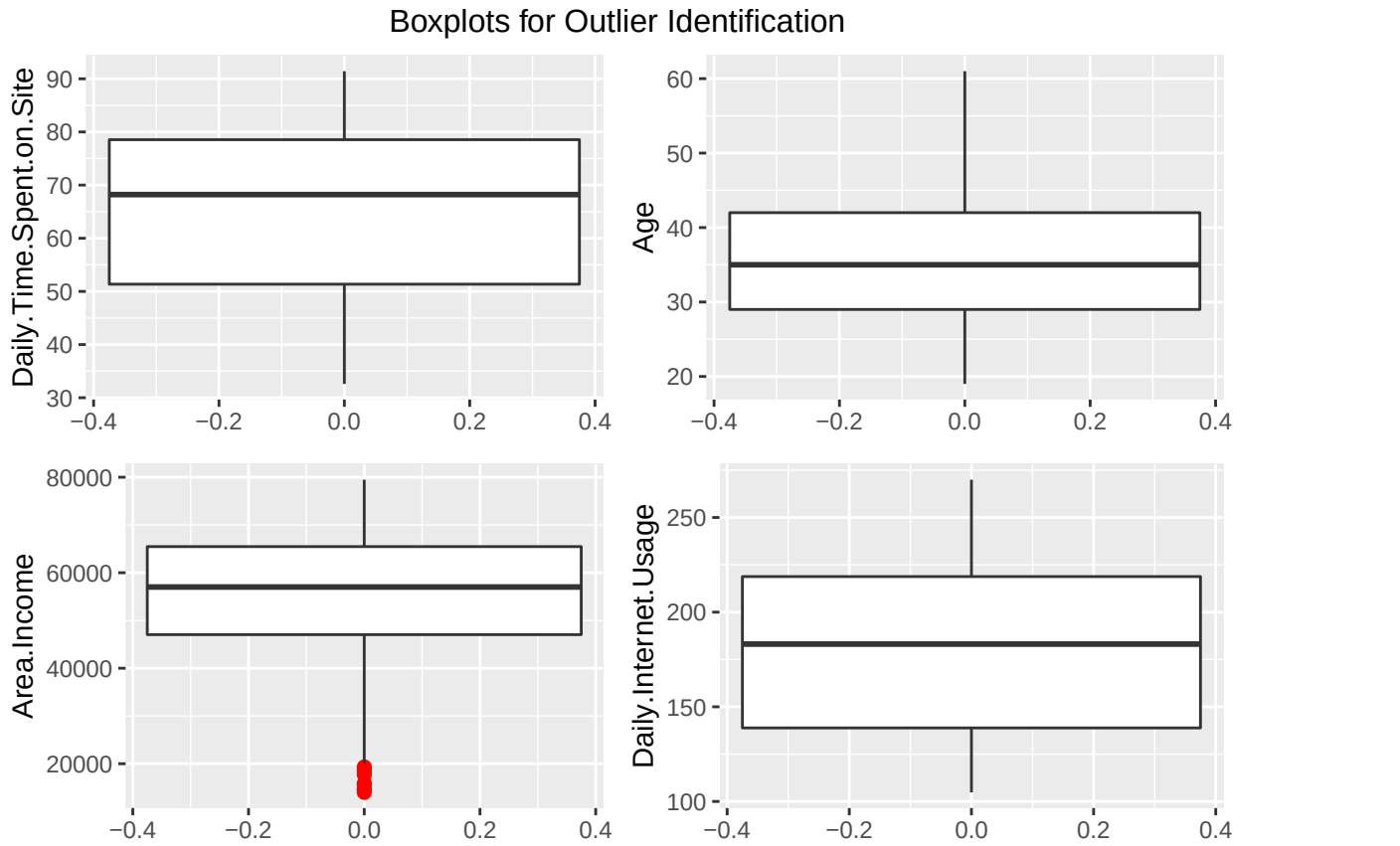
```
# Therefore there are no duplicate ad topic lines
```

Therefore there are no missing values in our dataset and no duplicate values in ad topic lines

Outlier Study

We will build boxplots for identifying the distribution and identifying outliers of our 4 numeric variables. We have used ggplot2 for building the boxplots and set the argument outlier.colour="red",outlier.size=2 inside the geom_boxplot function. We have also used grid.arrange () to set the plots adjacent to each other

```
out1 <- ggplot(ad, aes(y=Daily.Time.Spent.on.Site)) +
  geom_boxplot(outlier.colour="red",outlier.size=2)

out2 <- ggplot(ad, aes(y=Age)) +
  geom_boxplot(outlier.colour="red", outlier.size=2)

out3 <- ggplot(ad, aes(y=Area.Income)) +
  geom_boxplot(outlier.colour="red",outlier.size=2)

out4 <- ggplot(ad, aes(y=Daily.Internet.Usage)) +
  geom_boxplot(outlier.colour="red",outlier.size=2)

grid.arrange(out1, out2, out3, out4, ncol = 2, nrow = 2, top = "Boxplots for Outlier Identification")
```



```
outwhere <- ad %>% filter(Area.Income < 20000)
head(outwhere$Country)
```

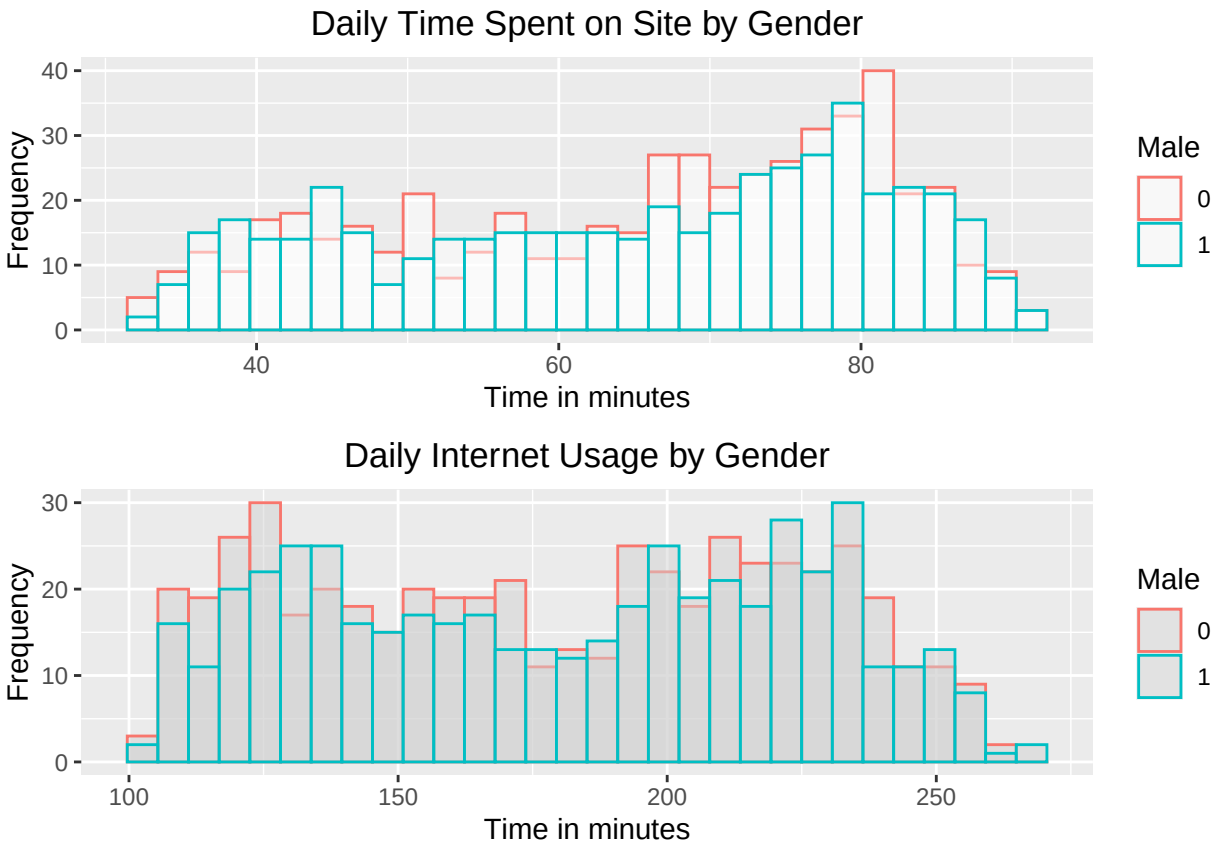
```
## [1] "Belize"                            "Madagascar"
## [3] "Heard Island and McDonald Islands" "Algeria"
## [5] "Azerbaijan"                        "Tajikistan"
```

Therefore, we have some outliers in Area Income column. These outliers lie below the 20,000 US$ range. We found the outliers using the filter(). The outliers in Area Income corresponded to the low-income regions, such as Algeria, Belize, Azerbaijan, and Madagascar

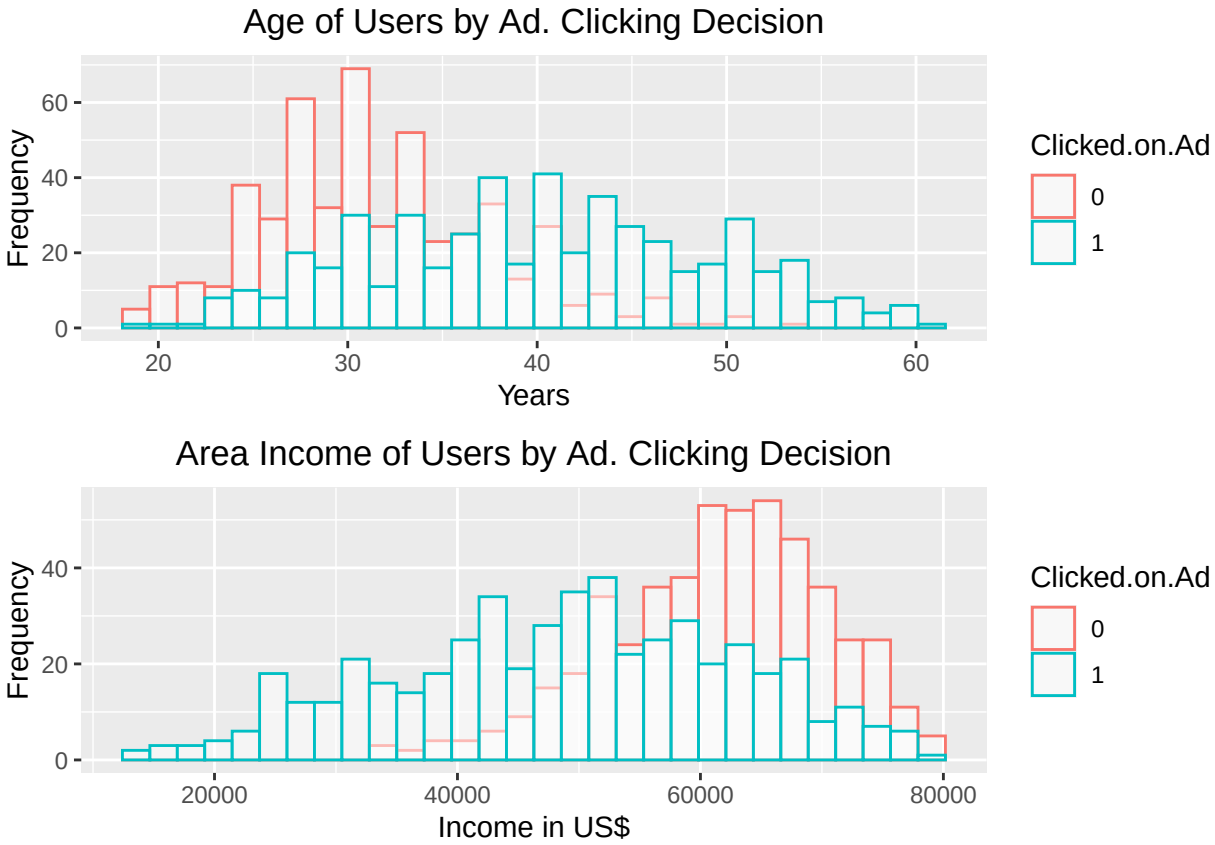
Distribution of Columns

We have built histograms to check the distribution of the values in addition to the boxplots above. We have used grid.arrange () function to keep the common histograms adjacent to each other. We have also added legends corresponded to 'Male' and 'Clicked on Add' categorical variables. Male = 1 means the user is male and 0 means user is not male. Similarly Clicked on Ads = 1 means the user has actually clicked the advertisement

```
hist_1 <- ggplot(ad, aes(x=Daily.Time.Spent.on.Site, color=Male)) +
  geom_histogram(fill = "White", alpha=0.5, position="identity") +
    ggtitle("Daily Time Spent on Site by Gender") +
      xlab("Time in minutes") + ylab("Frequency") +
        theme(plot.title = element_text(hjust = 0.5))

hist_2 <- ggplot(ad, aes(x=Daily.Internet.Usage, color=Male)) +
  geom_histogram(fill = "lightgrey", alpha=0.5, position="identity") +
   ggtitle("Daily Internet Usage by Gender") +
    xlab("Time in minutes") + ylab("Frequency") +
      theme(plot.title = element_text(hjust = 0.5))

hist_3 <- ggplot(ad, aes(x=Age, color=Clicked.on.Ad)) +
  geom_histogram(fill = "White", alpha=0.5, position="identity") +
  ggtitle("Age of Users by Ad. Clicking Decision") +
  xlab("Years") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5))


hist_4 <- ggplot(ad, aes(x=Area.Income, color=Clicked.on.Ad)) +
  geom_histogram(fill = "White", alpha=0.5, position="identity") +
  ggtitle("Area Income of Users by Ad. Clicking Decision") +
  xlab("Income in US$") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5))


grid.arrange(hist_1, hist_2, ncol = 1, nrow = 2)
```




```
grid.arrange(hist_3, hist_4, ncol = 1, nrow = 2)
```



Therefore, our 4 numeric columns: 'Daily Time Spent on Site' , 'Daily Internet Usage' , 'Age' , and 'Area Income' are not normally distributed. Another interesting insight from the histogram is: - Frequency of daily time spent on the site for women appears slightly more than men

Frequency of Countries and Scatter Plots

We have used dplyr package and piping operators to find the countries with maximum frequencies

```
country_freq <- ad %>% group_by(Country) %>% summarise(count = n()) %>% arrange(desc(count))
head(country_freq, n = 5) # Selecting the top 5 countries in terms of frequency
```

```
## # A tibble: 5 x 2
##   Country        count
##   <chr>          <int>
## 1 Czech Republic     9
## 2 France             9
## 3 Afghanistan        8
## 4 Australia          8
## 5 Cyprus             8
```

```
# Therefore maximum users are from Czech Republic and France in our dataset

ad_1 <- ad[ad$Clicked.on.Ad == 1,]
country_freq_1 <- ad_1 %>% group_by(Country) %>% summarise(count = n()) %>% arrange(desc(count))
head(country_freq_1, n = 5)
```

```
## # A tibble: 5 x 2
##   Country       count
##   <chr>         <int>
## 1 Australia         7
## 2 Ethiopia          7
## 3 Turkey            7
## 4 Liberia           6
## 5 Liechtenstein     6
```

```
# Therefore most number of users are from Australia, Ethiopia, and Turkey

#Age VS. Income ggplot
age.income <- ggplot(ad, aes(x = Age, y = Area.Income, shape = Clicked.on.Ad, color = Clicked.on.Ad)) +
                geom_point() +
                  ggtitle("Age vs Area Income") +
                      xlab("Age (in years)") + ylab("Income (in US$)") +
                        theme(plot.title = element_text(hjust = 0.5))


#Different age of customer spend on ad daily
age_Dailytimespend <- ggplot(ad, aes(x = Age, y = Daily.Time.Spent.on.Site, shape = Clicked.on.Ad, colo
                        geom_point() +
                          ggtitle("Age vs Daily Time Spent on Site") +
                            xlab("Age (in years)") + ylab("Time (in minutes)") +
                              theme(plot.title = element_text(hjust = 0.5))


grid.arrange(age.income, age_Dailytimespend, ncol = 1, nrow = 2)
```

Therefore maximum users are from Czech Republic and France in our dataset. However, users who actually click on the advertisements are mostly from from Australia, Ethiopia, and Turkey.

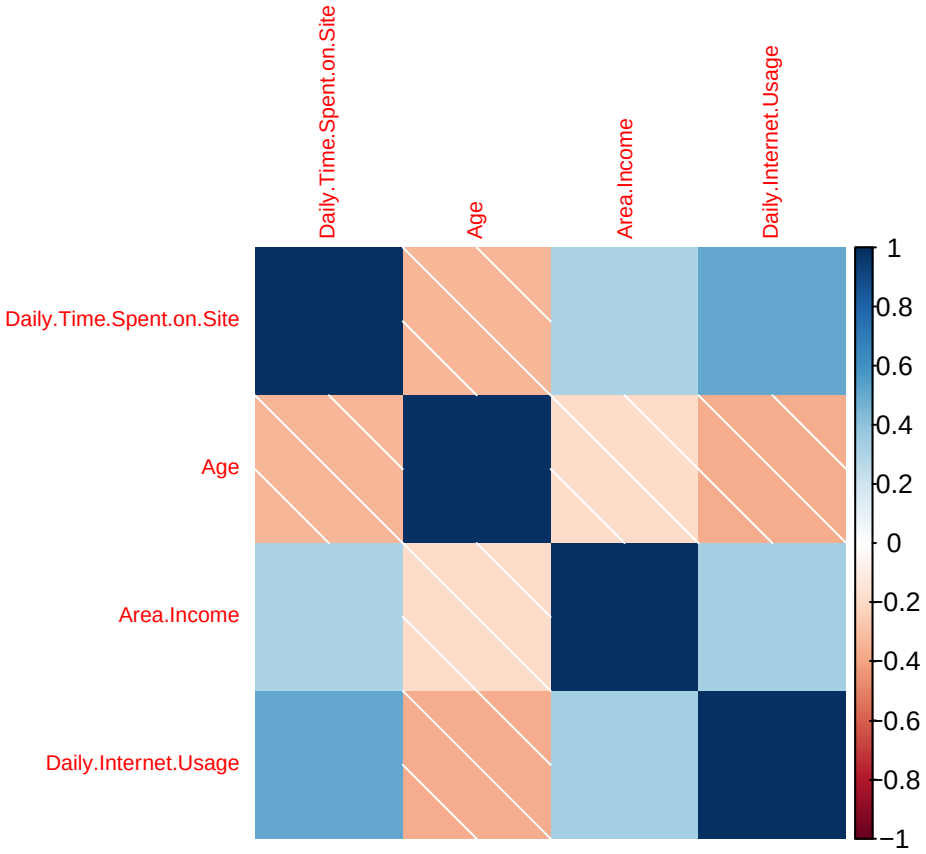
-Generally users from low income group actually clicks the website across all age groups

-Across all age groups, users who actually don't spent much time on the site click on the advertisements (this can also be the reason for page navigation; i.e. user click on the advertisement and therefore navigates away from the website

Correlation graphs

We have generated correlation plots of all the numerical variables. We have used corrplot package for generating the correlation plots

```
ad_numeric <- ad[c("Daily.Time.Spent.on.Site", "Age", "Area.Income", "Daily.Internet.Usage")]
c <- cor(ad_numeric)
corrplot(c, method = "shade", tl.cex=0.7)
```



We see that correlation is maximum for 'daily time on site' and 'daily internet usage'. This makes sense as more time spent on the website will lead to more internet usage. We also observed that age is negatively correlated to 'daily time spent on site' and 'daily internet usage'

Hypothesis Testing

```
#Hypothesis testing
cor.test(ad$Daily.Internet.Usage,ad$Daily.Time.Spent.on.Site)
```

```
##
##  Pearson's product-moment correlation
##
## data:  ad$Daily.Internet.Usage and ad$Daily.Time.Spent.on.Site
## t = 19.164, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4718365 0.5625633
## sample estimates:
##       cor
## 0.5186585
```

From the above results we can conclude that the correlation is statistically significant and its value is 0.52

```
cor.test(ad$Age,ad$Daily.Time.Spent.on.Site)
```

```
##
##  Pearson's product-moment correlation
##
## data:  ad$Age and ad$Daily.Time.Spent.on.Site
## t = -11.101, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3855821 -0.2751755
## sample estimates:
##       cor
## -0.3315133
```

From the above results we can conclude that the correlation is statistically significant, Age and Daily.Time.Spent.on.Site are negatively correlated with correlation coefficient of (-0.33)

Word Cloud

Bag of Words

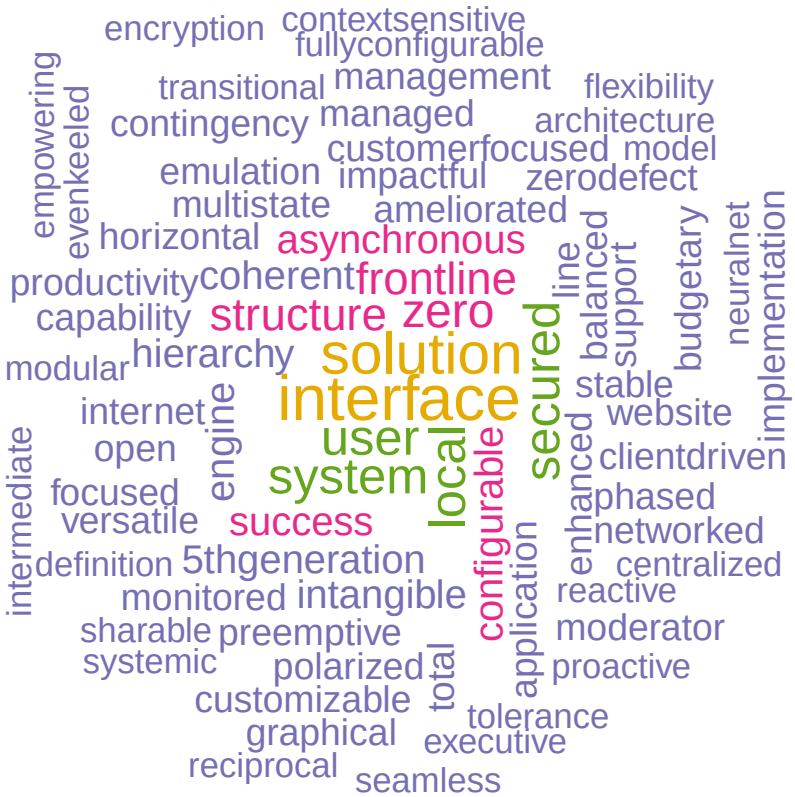
We have used 'Bag of words' method for extracting useful information from our variable 'Ad Topic Line'. We will store the text from this variable, clean the text data (removed punctuation marks and extra white spaces). We have not used text stemming (converting words into their root form) in this assignment. We have used libraries 'tm' , 'Snowballc' , 'wordcloud' , 'RColorBrewer' , 'RCurl' , and 'XML' for performing bag of words and building word cloud

```
# Creating Word Cloud
text <- ad$Ad.Topic.Line
docs <- Corpus(VectorSource(text))

# Text Cleaning (removing punctuations, and extra white spaces)
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, stripWhitespace)

# We learned another function for text stemming, but we are not using this for now
# docs <- tm_map(docs, stemDocument) # text stemming, converting words into their root form

# Build a term-doument matrix
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 20)
```

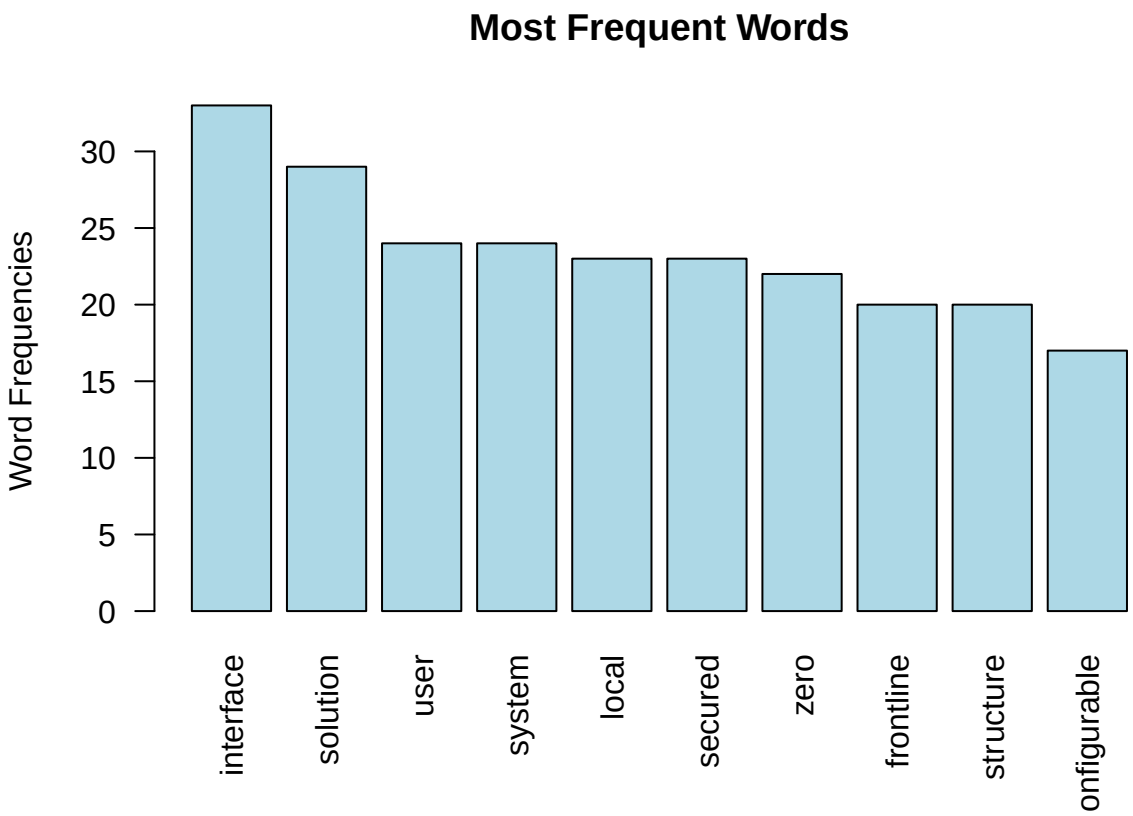
```
##                       word freq
## interface         interface   33
## solution           solution   29
## user                   user   24
## system               system   24
## local                 local   23
```

```
## secured             secured   23
## zero                   zero   22
## frontline          frontline   20
## structure          structure   20
## configurable    configurable   17
## asynchronous    asynchronous   17
## success             success   17
## 5thgeneration 5thgeneration   16
## coherent           coherent   16
## hierarchy          hierarchy   16
## engine               engine   16
## intangible       intangible   16
## enhanced           enhanced   15
## ameliorated     ameliorated   15
## impactful         impactful   15
```

```
set.seed(1221)
wordcloud(words = d$word, freq = d$freq, scale=c(2,.5),
          max.words=75, random.order=FALSE, rot.per=0.2,
          colors=brewer.pal(6, "Dark2"))
```




```
barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,
        col ="lightblue", main ="Most Frequent Words",
        ylab = "Word Frequencies")
```



We generated a word cloud having the font size corresponding to their frequency.

We observed that word 'Interface' occurred maximum number of times (33 times) in the advertisement topic line; followed by 'Solution' (24 times). As per this information, the marketing personnel can build advertisements with the following words in their title line to draw maximum attention of the users.

Feature Engineering

```
#Feature Engineering
#Create a duplicate data-frame to proceed with feature engineer and modeling
ad_copy <- ad

#Country column
length(unique(ad_copy$Country))
```

```
## [1] 237
```

We have 237 unique countries(regions) in our dataset.

Count Encoding

```
ad_copy$Country.Count <- as.numeric(ave(ad_copy$Country, ad_copy$Country, FUN = length))
ad_copy$Country<-NULL
```

Here we have created the country column by its frequency, this is count encoding. Converting a character column to numeric.

```
#City column
length(unique(ad_copy$City))
```

```
## [1] 969
```

```
ad_copy$City = NULL
```

969 distinct cities in 1000 rows, hence we delete this column.

```
#Timestamp
#We extract year, month, weekday and hour to derive insights
ad_copy$Timestamp <- as.POSIXct(ad_copy$Timestamp)
ad_copy$Month <- month(ad_copy$Timestamp) #7 month data
#ad$Day1 <- weekdays(ad$TimeStamp)
ad_copy$Day <- wday(ad_copy$Timestamp) #Sunday = 1....Saturday = 7
ad_copy$Hour <- hour(ad_copy$Timestamp) #24 hour scale
ad_copy$Year <- year(ad_copy$Timestamp)
ad_copy$year <- NULL
ad_copy$Timestamp<-NULL
```

This is 7-month data for the year 2016, hence all the values in column year are 2016, therefore we drop this column

```
#Converting Month, Day and Hour to factor
ad_copy$Month<-as.factor(ad_copy$Month)
ad_copy$Day<-as.factor(ad_copy$Day)
ad_copy$Hour<-as.factor(ad_copy$Hour)
```

Extracting key words based on frequency from AD.Topic.Line to provide as input to model

```
#Stemming
topic_stem <- tm_map(docs, stemDocument)
topic_dtm <- DocumentTermMatrix(topic_stem)
#Removing sparse columns
dtm <- removeSparseTerms(topic_dtm, 0.975)
#Columns reduced from 303 to 5
bow <- as.matrix(dtm)

#Adding word columns to our dataframe
ad_copy <- cbind(ad_copy, bow)
ad_copy$Ad.Topic.Line = NULL
```

Train Test Split Train Test split (We are splitting data to 80% for train and 20% for test. Since we have less data we would implement k fold cross validation)

```
set.seed(123)
train_idx <- sample(nrow(ad_copy), .80*nrow(ad_copy))

ad_train <- ad_copy[train_idx,]
ad_test <- ad_copy[-train_idx,]
```

```
# Define train control for k fold cross validation
train_control <- trainControl(method="cv", number=4)
```

K Fold Validation

Model Building

```
mod_log <- train(Clicked.on.Ad~.,data=ad_train, trControl=train_control, method="glm")
print(mod_log)
```

Logistic Model

```
## Generalized Linear Model
##
## 800 samples
##  15 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 599, 601, 600, 600
## Resampling results:
##
##   Accuracy  Kappa
##   0.938718  0.8773299
```

```
mod_nb <- train(Clicked.on.Ad~.,data=ad_train, trControl=train_control, method="nb")
print(mod_nb)
```

Naive Bayes

```
## Naive Bayes
##
## 800 samples
##  15 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 600, 600, 600, 600
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy  Kappa
##   FALSE           NaN         NaN
##    TRUE      0.95375   0.9074294
```

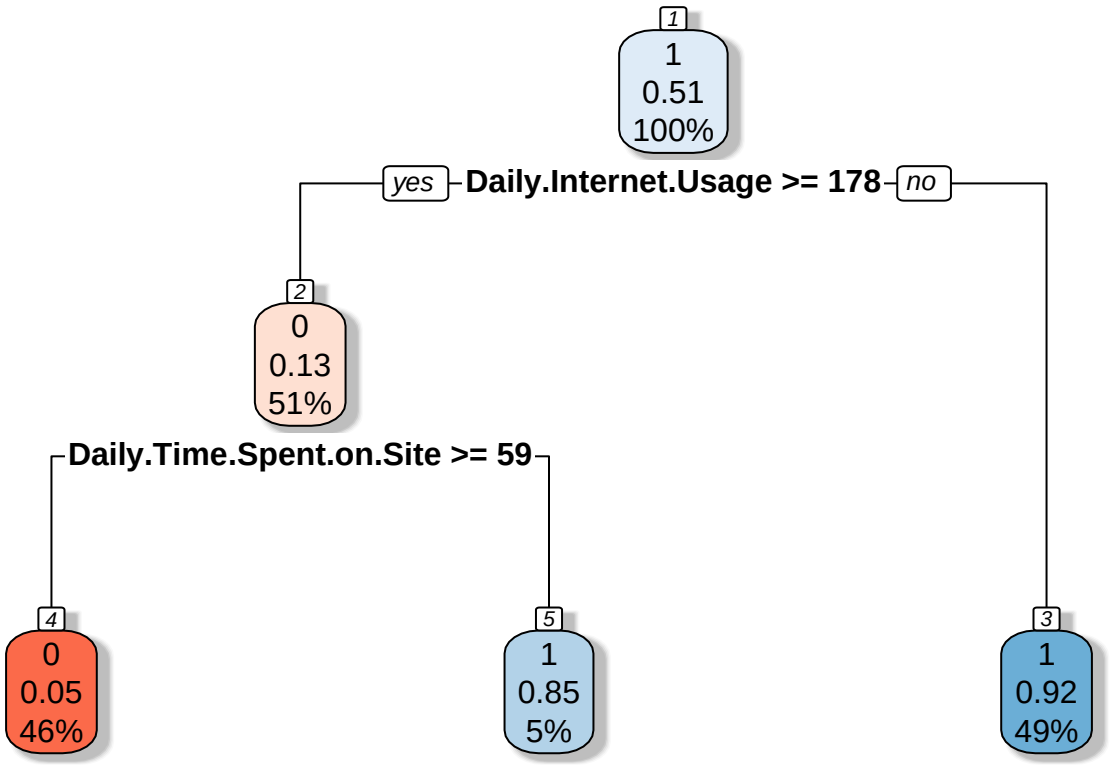
```
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```

```
mod_dt <- train(Clicked.on.Ad~.,data=ad_train, trControl=train_control, method="rpart")
print(mod_dt)
```

Decision Tree

```
## CART
##
## 800 samples
##  15 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 601, 600, 599, 600
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.02570694  0.9237242  0.8471731
##   0.07455013  0.9111614  0.8221266
##   0.78149100  0.7842024  0.5623293
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.02570694.
```

```
rpart.plot(mod_dt$finalModel, box.palette="RdBu", shadow.col="gray", nn=TRUE)
```



From above results, Naive Bayes gives us the best accuracy of 95%

Validation on test dataset

```
actual = ad_test$Clicked.on.Ad
ad_test$Clicked.on.Ad <- NULL
pred <- predict(mod_nb, ad_test)
confusionMatrix(pred, actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 108   3
##          1   3  86
##
##                Accuracy : 0.97
##                  95% CI : (0.9358, 0.9889)
##     No Information Rate : 0.555
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9393
##
```

```
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9730
##             Specificity : 0.9663
##          Pos Pred Value : 0.9730
##          Neg Pred Value : 0.9663
##              Prevalence : 0.5550
##          Detection Rate : 0.5400
##    Detection Prevalence : 0.5550
##       Balanced Accuracy : 0.9696
##
##        'Positive' Class : 0
##
```

The accuracy on test dataset for Naive Bayes model is 97%.

We finalize this model for predicting Ad Click prediction.

Conclusion

Outliers and Duplicate Values

Outliers: We had outliers in our dataset (under Area_Income); We have found the outliers but not removed them as these outliers were from expected low-income regions. There were no other outliers in our dataset

Distribution: None of the numerical variables were normally distributed

Insights on Advertisements

Frequency of daily time spent on the site for women appeared slightly more as compared to men

Maximum users are from Czech Republic and France in our dataset. However, Users who actually click on the Ad are mostly from Australia, Ethiopia, and Turkey.

Generally, users from low-income group actually clicks the website across all age groups

Across all age groups, users who actually don't spent much time on the site click on the advertisements (this can also be the reason for page navigation; i.e. user click on the advertisement and therefore navigates away from the website)

Correlation

We see that correlation is maximum for 'daily time on site' and 'daily internet usage'. This makes sense as more time spent on the website will lead to more internet usage. We also observed that age is negatively correlated to 'daily time spent on site' and 'daily internet usage

Bag of Words

Word 'Interface' occurred maximum number of times (33 times) in the advertisement topic line; followed by 'Solution' (24 times). As per this information, the marketing personnel can build advertisements with the following words in their title line to draw maximum attention of the users

Naive Byes Classifier

From Naive Byes classifier we are getting an accuracy of 97% on the test data. We will use this model to gain the insights as to which users are more prone to clicking on what type of advertisements. This will help us increase the click rate on the advertisements.

New Functions

Topics to be covered

cross validation, count encoding, Handling time and date data, naive bayes, Document-Term-Matrix, Decision Tree.

Cross validation:

The dataset that we are dealing with is relatively small (1000 instances), thus using a holdout set to estimate its performance will not give us an accurate result and will suffer from high variance. Here comes cross-validation into play, by dividing the dataset into k equal chunks or folds, then each time, training the model on k-1 folds and testing its performance on the remaining fold. This operation is repeated until every fold is used once as the holdout set. In R, this can be done through the following 2 steps:

1-use the function trainControl where one specify "cv" in method and the number of folds in k

2-feed the previous list to the function train along with the training set and other parameters related to the model.

Count Encoding:

Most machine learning models accept only numerical data, this is why string categorical/factor columns must be processed and transformed into numerical ones. Count encoding consists of replacing a categorical instance with the number of times that it has appeared in the dataset. This technique was applied on the column "Country". For instance, "Morocco" will be replaced by 3. If we have opted for the standard dummy encoding also known as one hot encoding we would have ended up with 237 columns instead of one in the case of count encoding. In R, we were able to count encode "Country" by:

-using the function ave with the parameter FUN set to length and ad$Country as grouping variable and input

-This function will group rows with the same value into sub-groups and then compute the length of each sub-group.

Handling time and date data:

In the dataset, the column "Timestamp" contains a character representing date and time (eg. '2016-03-27 00:53:11'). In order to extract the year, month, day, hour, minutes, and seconds information form this latter:

1- we converted "Timestamp" to a format that is more convenient to handle time and dates in R, using the function: as.POSIXct

2-we applied the functions year,month,wday, hour,minute, second on the previous object to extract the targeted quantities.

Naive Bayes:

Naive Bayes is a probabilistic classification algorithm based on Bayes theorem. It is widely used for text classification and spam filtering. Without going into heavy mathematical details, the core assumption of this algorithm is the conditional independence of features given the class/target. For each class, it computes the probability that an instance belongs to it, then outputs the predicted class as the one having the highest probability. In R, one can take advantage of this algorithm by setting the parameter method of the train function to "nb"

Document-Term matrix:

A Document-Term matrix is a text mining concept, used in text mining. The rows correspond to text/documents within the dataset and columns correspond to terms/words. For instance if row[i]column[j] = 5, it means that the document/text at the i th positions contains the term/world at the j th position 5 times. In R, this can be done through:

1-generating the Document-Term matrix using the function DocumentTermMatrix.

2-getting rid of sparse columns, using removeSparseTerms where we set sparse to 0.975, which means that columns/words with a sparsity rate higher than 0.975 will be dropped.

Decision tree(rpart):

Decision tree is a supervised learning algorithm used for solving classification problems. It splits data into 2 or more parts based on most appropriate attribute for the split. Which column to split on is determined by Information Gain (Entropy), Gini index(default), Misclassification error etc. Decision trees can be tuned by controlling the hyperparameters like depth of tree, min leaf size etc. It take both categorical as well as continuous data as input. In R, one can take advantage of this algorithm by setting the parameter method of the train function to "rpart". We have installed rpart and rpart.plot plackages for building and plotting decision trees. rpart.plot function plots the dendogram displaying the columns and the thresholds used for split.