Codes

Akshay

2023-12-04

Installing necessary packages

```
# install.packages("rstatix")
# install.packages("vctrs")
# install.packages("devtools")
# devtools::install_github("dongyuanwu/RSBID")
# install.packages("dplyr")
# install.packages("corrplot")
# install.packages("caret")
# install.packages("magrittr")
# install.packages("MLmetrics")
# install.packages("rpart.plot")
# install.packages("e1071")
library(rstatix)
```

```
## Warning: package 'rstatix' was built under R version 4.2.3
```

```
##
## Attaching package: 'rstatix'
```

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

```
library("RSBID")
```

```
## Loading required package: FNN
```

```
## Warning: package 'FNN' was built under R version 4.2.3
```

```
## Loading required package: clustMixType
```

```
## Warning: package 'clustMixType' was built under R version 4.2.3
```

```
## Loading required package: klaR
```

```
## Warning: package 'klaR' was built under R version 4.2.3
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:rstatix':
##
##     select
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

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

```
## The following object is masked from 'package:MASS':
##
##     select
```

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

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

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

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

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 4.2.3
```

```
library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.2.3
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
```

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

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.3
```

```
## Loading required package: rpart
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

Loading Data

Link to the Dataset:
https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset

```
set.seed(12345)
df <- read.csv("…/online_shoppers_intention.csv", stringsAsFactors = TRUE)
```

Understanding data

```
# View(df)
dim(df)
```

```
## [1] 12330    18
```

```
str(df)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 0 2 3 ...
##  $ ProductRelated_Duration: num  0 64 0 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : Factor w/ 10 levels "Aug","Dec","Feb",..: 3 3
3 3 3 3 3 3 3 3 ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : Factor w/ 3 levels "New_Visitor",..: 3 3 3 3 3
3 3 3 3 3 ...
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```r
summary(df)
```

```
##  Administrative   Administrative_Duration Informational
##  Min.   : 0.000   Min.   :   0.00         Min.   : 0.0000
##  1st Qu.: 0.000   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.000   Median :   7.50         Median : 0.0000
##  Mean   : 2.315   Mean   :  80.82         Mean   : 0.5036
##  3rd Qu.: 4.000   3rd Qu.:  93.26         3rd Qu.: 0.0000
##  Max.   :27.000   Max.   :3398.75         Max.   :24.0000
##
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :   0.00        Min.   :  0.00   Min.   :    0.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  184.1
##  Median :   0.00        Median : 18.00   Median :  598.9
##  Mean   :  34.47        Mean   : 31.73   Mean   : 1194.8
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1464.2
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##
##   BounceRates         ExitRates         PageValues        SpecialDay
##  Min.   :0.000000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.003112   Median :0.02516   Median :  0.000   Median :0.00000
##  Mean   :0.022191   Mean   :0.04307   Mean   :  5.889   Mean   :0.06143
##  3rd Qu.:0.016813   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##
##      Month      OperatingSystems    Browser           Region
##  May    :3364   Min.   :1.000    Min.   : 1.000   Min.   :1.000
##  Nov    :2998   1st Qu.:2.000    1st Qu.: 2.000   1st Qu.:1.000
##  Mar    :1907   Median :2.000    Median : 2.000   Median :3.000
##  Dec    :1727   Mean   :2.124    Mean   : 2.357   Mean   :3.147
##  Oct    : 549   3rd Qu.:3.000    3rd Qu.: 2.000   3rd Qu.:4.000
##  Sep    : 448   Max.   :8.000    Max.   :13.000   Max.   :9.000
##  (Other):1337
##   TrafficType                VisitorType     Weekend          Revenue
##  Min.   : 1.00   New_Visitor      : 1694   Mode :logical   Mode :logical
##  1st Qu.: 2.00   Other            :   85   FALSE:9462      FALSE:10422
##  Median : 2.00   Returning_Visitor:10551   TRUE :2868      TRUE :1908
##  Mean   : 4.07
##  3rd Qu.: 4.00
##  Max.   :20.00
##
```

```r
table(df$Revenue)
```

```
##
## FALSE  TRUE
## 10422  1908
```

```r
names(df)
```

```
##  [1] "Administrative"          "Administrative_Duration"
##  [3] "Informational"           "Informational_Duration"
##  [5] "ProductRelated"          "ProductRelated_Duration"
##  [7] "BounceRates"             "ExitRates"
##  [9] "PageValues"              "SpecialDay"
## [11] "Month"                   "OperatingSystems"
## [13] "Browser"                 "Region"
## [15] "TrafficType"             "VisitorType"
## [17] "Weekend"                 "Revenue"
```

Required pre-processes

1) Converting to factor (Done)
2) missing values (Done)
3) Duplicate values (Done)
4) Outliers ()
5) Variable selection by correlation plot
6) balancing (Done)
7) Scaling (Done)

```
Names <-
c("SpecialDay","Month","OperatingSystems","Browser","Region","TrafficType","V
isitorType","Weekend","Revenue" )

for (i in Names) {
  df[,i] <- as.factor(df[,i])

}

str(df)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 0 2 3 ...
##  $ ProductRelated_Duration: num  0 64 0 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : Factor w/ 6 levels "0","0.2","0.4",..: 1 1 1 1
1 1 3 1 5 3 ...
##  $ Month                  : Factor w/ 10 levels "Aug","Dec","Feb",..: 3 3
3 3 3 3 3 3 3 3 ...
##  $ OperatingSystems       : Factor w/ 8 levels "1","2","3","4",..: 1 2 4 3
3 2 2 1 2 2 ...
##  $ Browser                : Factor w/ 13 levels "1","2","3","4",..: 1 2 1
2 3 2 4 2 2 4 ...
```

```
##  $ Region                 : Factor w/ 9 levels "1","2","3","4",..: 1 1 9 2
1 1 3 1 2 1 ...
##  $ TrafficType            : Factor w/ 20 levels "1","2","3","4",..: 1 2 3
4 4 3 3 5 3 2 ...
##  $ VisitorType            : Factor w/ 3 levels "New_Visitor",..: 3 3 3 3 3
3 3 3 3 3 ...
##  $ Weekend                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1
1 2 1 1 ...
##  $ Revenue                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1
1 1 1 1 ...
```

```
# Checking for missing values
sum(is.na.data.frame(df))
```

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

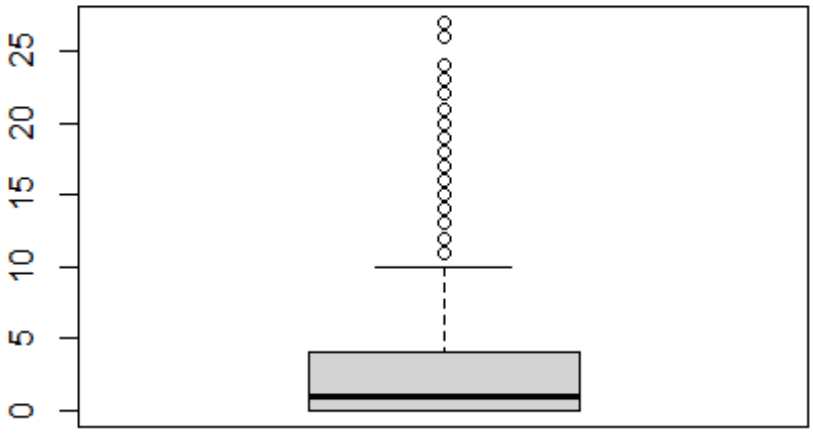
```
# Removing duplicate entries
library(dplyr)
dim(df)
```

```
## [1] 12330    18
```

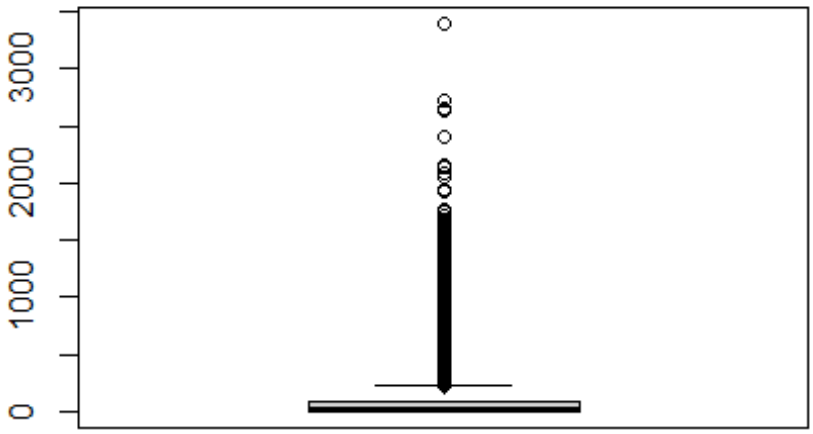
```
df <- distinct(df)
dim(df)
```

```
## [1] 12205    18
```

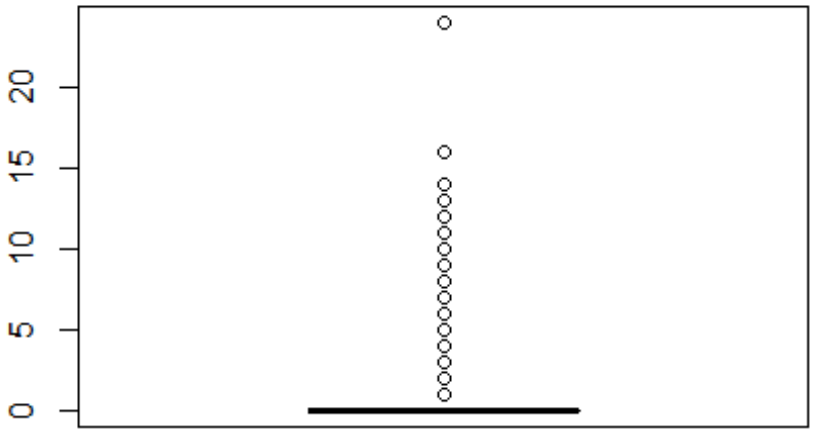
```
# plotting boxplots to identify outliers

X <-
c("Administrative","Administrative_Duration","Informational","Informational_D
uration","ProductRelated","ProductRelated_Duration","BounceRates","ExitRates"
,"PageValues")

for (j in X) {
  boxplot(df[,j], xlab = j)

}
```



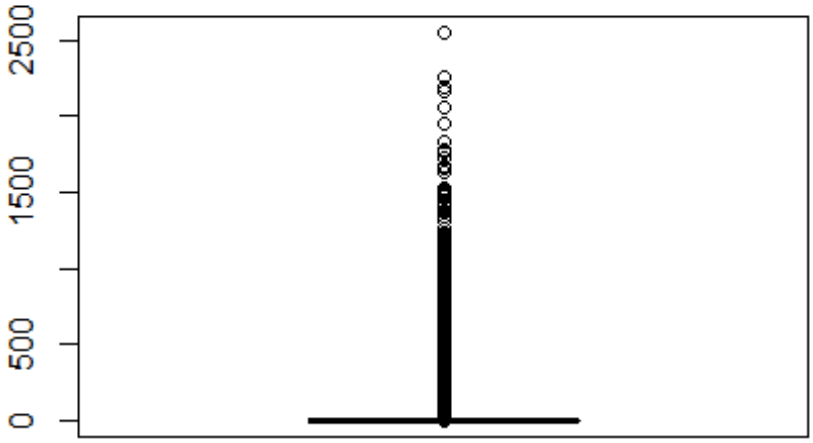
Administrative



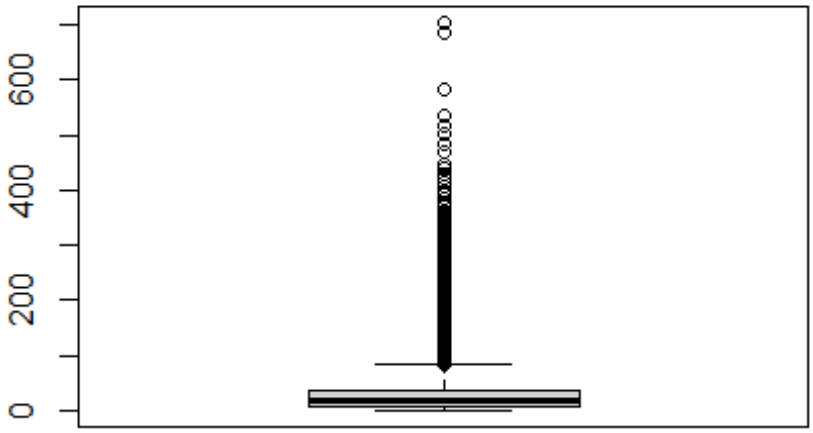
Administrative_Duration



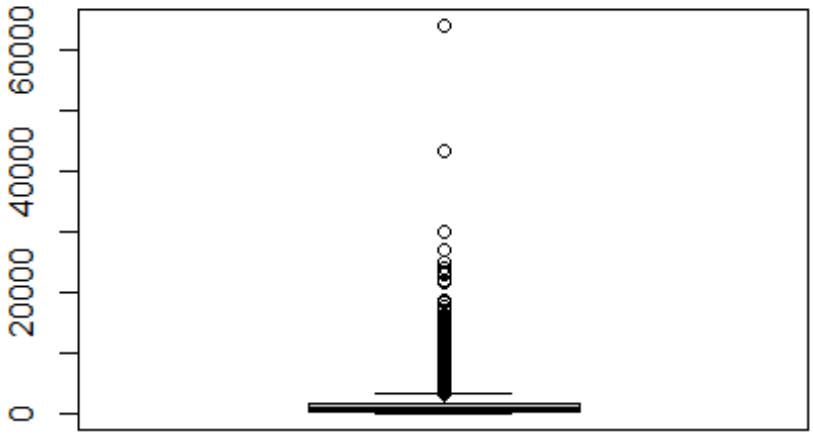
Informational



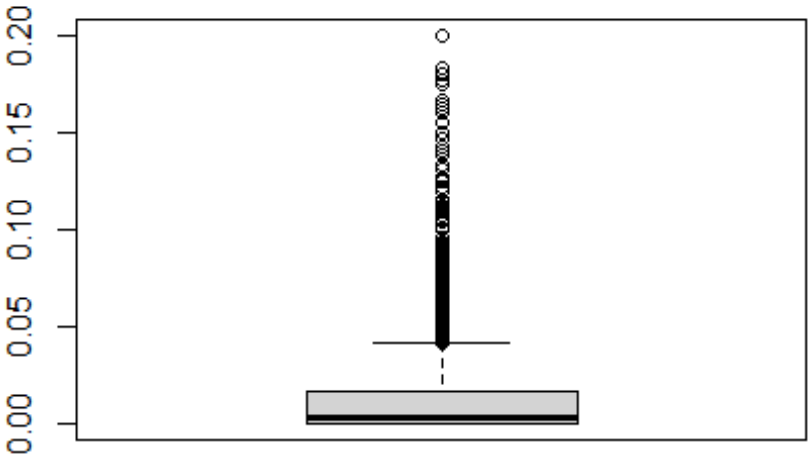
Informational_Duration



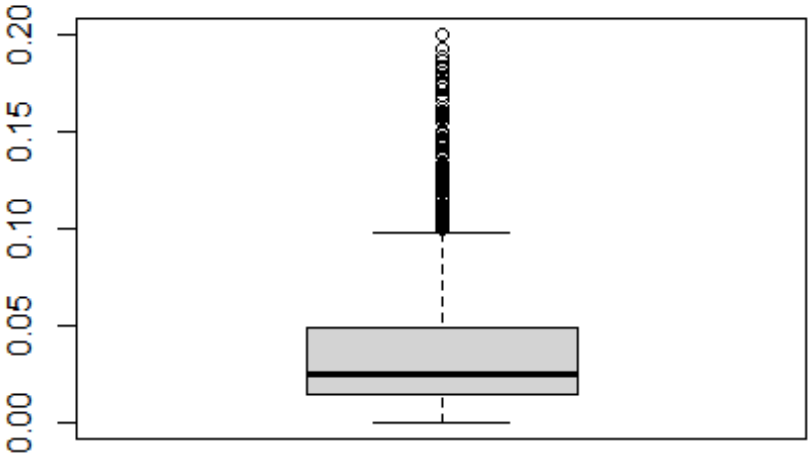
ProductRelated



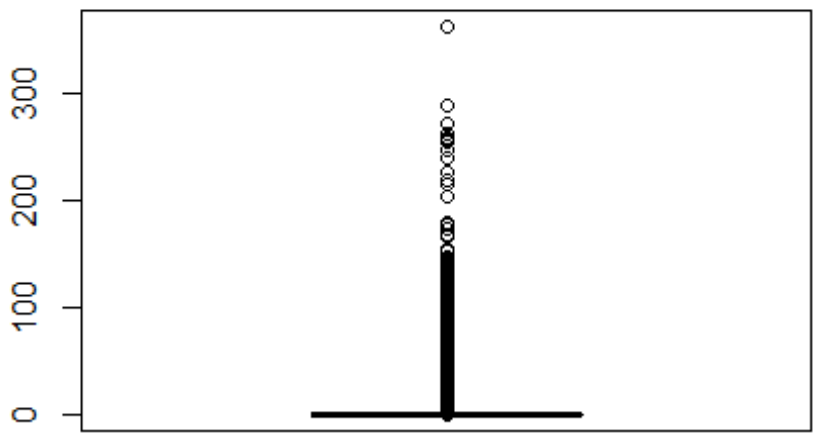
ProductRelated_Duration



BounceRates



ExitRates



PageValues

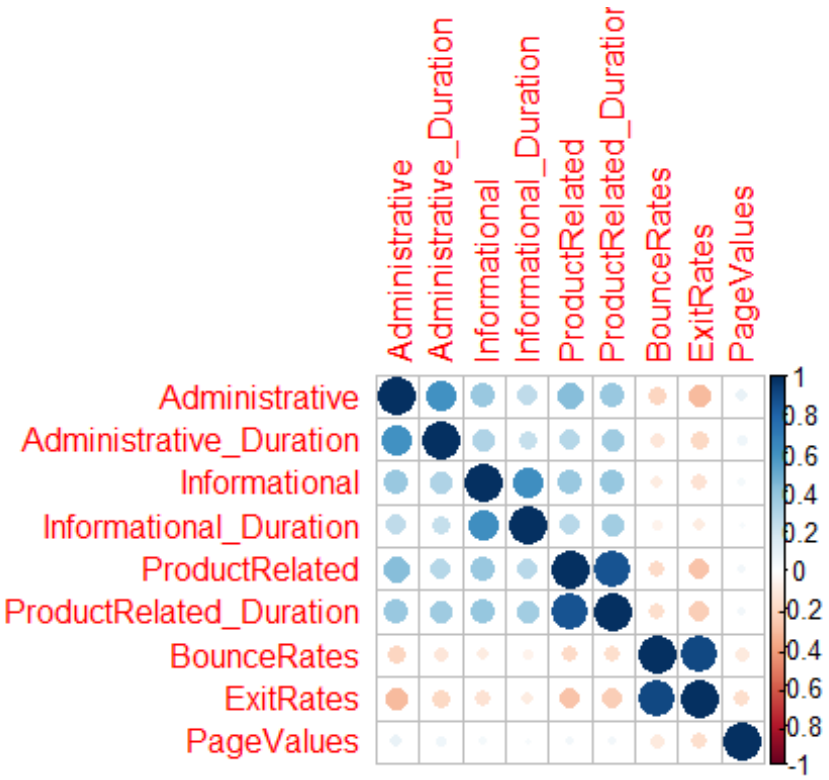
```
# There seems to be too many outliers to be seen from the boxlots
# Counting the extreme outliers in each variable

for (j in X) {
  Outliers <- identify_outliers(j, data = df)
  print("No. of outliers in")
  print (j)
  print(nrow(Outliers[Outliers$is.extreme == TRUE,]))

}
```

```
## Warning: Using an external vector in selections was deprecated in
tidyselect 1.1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(j)
##
##   # Now:
##   data %>% select(all_of(j))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## [1] "No. of outliers in"
## [1] "Administrative"
```

```
## [1] 51
## [1] "No. of outliers in"
## [1] "Administrative_Duration"
## [1] 540
## [1] "No. of outliers in"
## [1] "Informational"
## [1] 2631
## [1] "No. of outliers in"
## [1] "Informational_Duration"
## [1] 2405
## [1] "No. of outliers in"
## [1] "ProductRelated"
## [1] 446
## [1] "No. of outliers in"
## [1] "ProductRelated_Duration"
## [1] 398
## [1] "No. of outliers in"
## [1] "BounceRates"
## [1] 924
## [1] "No. of outliers in"
## [1] "ExitRates"
## [1] 666
## [1] "No. of outliers in"
## [1] "PageValues"
## [1] 2730
```

There are too many extreme outliers according to this result. So, removing them is not a wise decision. Instead models that are tolerant to outliers will be used to construct the classifier

```
# Correlation pot
df.cor <- cor(df[-c(10:18)])
corrplot(df.cor)
```




```
# calculating the extreme correlations
cor(df$ProductRelated,df$ProductRelated_Duration)
```

```
## [1] 0.8603299
```

```
cor(df$BounceRates,df$ExitRates)
```

```
## [1] 0.9021444
```

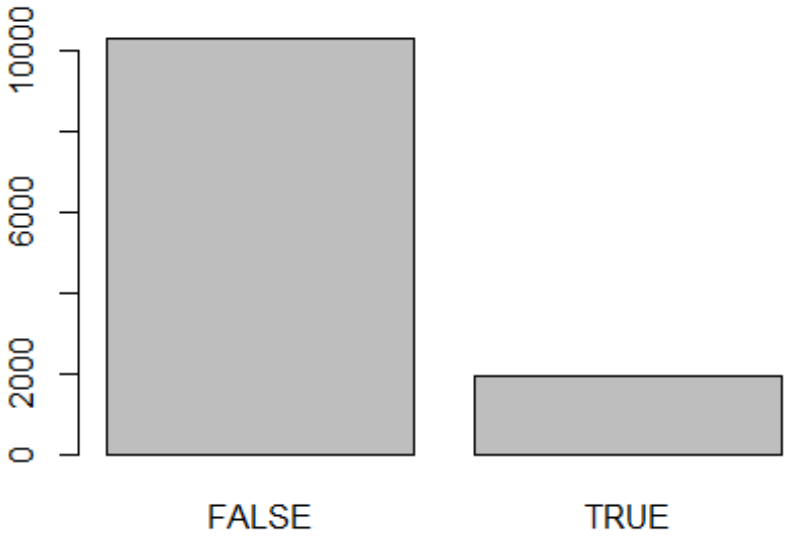
```
cor(df$Administrative,df$Administrative_Duration)
```

```
## [1] 0.6004568
```

```
# removing features
 df <- df[-c(5,7)]

# Data balancing with SMOTE_NC

# Class distribution before SMOTE
(T <- table(df$Revenue))
```

```
##
## FALSE  TRUE
## 10297  1908
```

```
barplot(T)
```




```
# SMOTE
df <- SMOTE_NC(df,16)

## Variables are continous and categorical, SMOTE_NC could be used.

##   |
|                                                                      |   0%
|                                                                      |   1%
|=                                                                     |   1%
|=                                                                     |   2%
|==                                                                    |   2%
|==                                                                    |   3%
|==                                                                    |   4%
|===                                                                   |   4%
|===                                                                   |   5%
|====                                                                  |   5%
|====                                                                  |   6%
```

```
|
|=====                                                                 |   6%
|
|=====                                                                 |   7%
|
|=====                                                                 |   8%
|
|======                                                                |   8%
|
|======                                                                |   9%
|
|=======                                                               |   9%
|
|=======                                                               |  10%
|
|=======                                                               |  11%
|
|========                                                              |  11%
|
|========                                                              |  12%
|
|=========                                                             |  12%
|
|=========                                                             |  13%
|
|=========                                                             |  14%
|
|==========                                                            |  14%
|
|==========                                                            |  15%
|
|===========                                                           |  15%
|
|===========                                                           |  16%
|
|============                                                          |  16%
|
|============                                                          |  17%
|
|============                                                          |  18%
|
|=============                                                         |  18%
|
|=============                                                         |  19%
|
|==============                                                        |  19%
|
|==============                                                        |  20%
|
|==============                                                        |  21%
```

```
|
|===============                                                  |  21%
|
|===============                                                  |  22%
|
|================                                                 |  22%
|
|================                                                 |  23%
|
|================                                                 |  24%
|
|=================                                                |  24%
|
|=================                                                |  25%
|
|==================                                               |  25%
|
|==================                                               |  26%
|
|===================                                              |  26%
|
|===================                                              |  27%
|
|===================                                              |  28%
|
|====================                                             |  28%
|
|====================                                             |  29%
|
|=====================                                            |  29%
|
|=====================                                            |  30%
|
|=====================                                            |  31%
|
|======================                                           |  31%
|
|======================                                           |  32%
|
|=======================                                          |  32%
|
|=======================                                          |  33%
|
|=======================                                          |  34%
|
|========================                                         |  34%
|
|========================                                         |  35%
|
|=========================                                        |  35%
```

```
|
|========================                                 |  36%
|
|=========================                                |  36%
|
|=========================                                |  37%
|
|==========================                               |  38%
|
|==========================                               |  38%
|
|===========================                              |  39%
|
|===========================                              |  39%
|
|============================                             |  40%
|
|============================                             |  41%
|
|=============================                            |  41%
|
|=============================                            |  42%
|
|=============================                            |  42%
|
|==============================                           |  43%
|
|==============================                           |  44%
|
|===============================                          |  44%
|
|===============================                          |  45%
|
|===============================                          |  45%
|
|================================                         |  46%
|
|================================                         |  46%
|
|=================================                        |  47%
|
|=================================                        |  48%
|
|==================================                       |  48%
|
|==================================                       |  49%
|
|===================================                      |  49%
|
|===================================                      |  50%
```

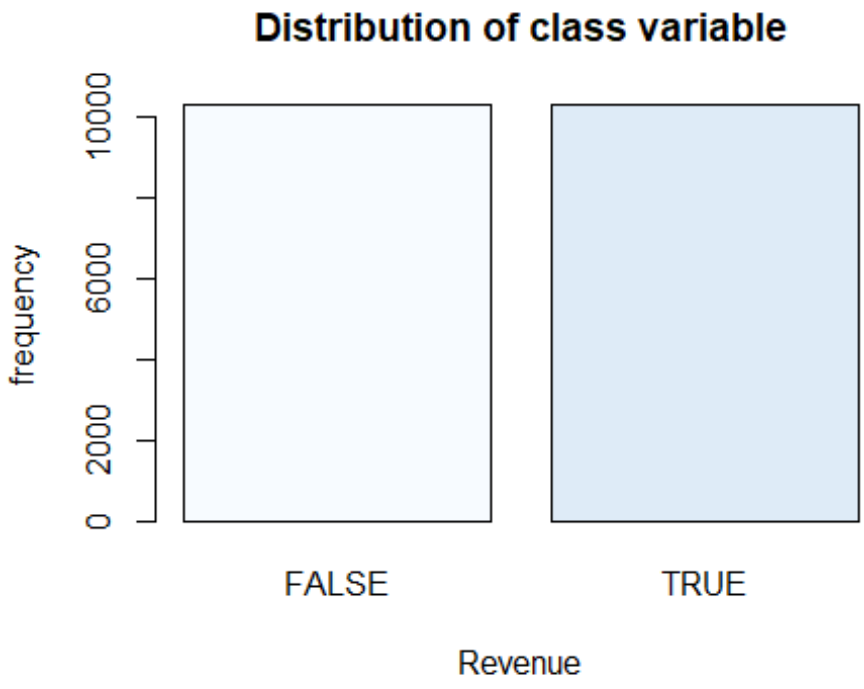
```
|
|==================================                                    |  51%
|
|===================================                                   |  51%
|
|====================================                                  |  52%
|
|=====================================                                 |  52%
|
|======================================                                |  53%
|
|=======================================                               |  54%
|
|========================================                              |  54%
|
|=========================================                             |  55%
|
|==========================================                            |  55%
|
|===========================================                           |  56%
|
|============================================                          |  56%
|
|=============================================                         |  57%
|
|==============================================                        |  58%
|
|==============================================                        |  58%
|
|===============================================                       |  59%
|
|================================================                      |  59%
|
|=================================================                     |  60%
|
|==================================================                    |  61%
|
|==================================================                    |  61%
|
|===================================================                   |  62%
|
|====================================================                  |  62%
|
|=====================================================                 |  63%
|
|======================================================                |  64%
|
|======================================================                |  64%
|
|=======================================================               |  65%
```

```
|
|=============================================                         |  65%
|
|==============================================                        |  66%
|
|===============================================                       |  66%
|
|================================================                      |  67%
|
|=================================================                     |  68%
|
|=================================================                     |  68%
|
|==================================================                    |  69%
|
|==================================================                    |  69%
|
|===================================================                   |  70%
|
|====================================================                  |  71%
|
|====================================================                  |  71%
|
|=====================================================                 |  72%
|
|=====================================================                 |  72%
|
|======================================================                |  73%
|
|======================================================                |  74%
|
|=======================================================               |  74%
|
|=======================================================               |  75%
|
|========================================================              |  75%
|
|========================================================              |  76%
|
|=========================================================             |  76%
|
|=========================================================             |  77%
|
|==========================================================            |  78%
|
|==========================================================            |  78%
|
|===========================================================           |  79%
|
|============================================================          |  79%
```

```
|
|=============================================================          |  80%
|
|==============================================================         |  81%
|
|===============================================================        |  81%
|
|================================================================       |  82%
|
|=================================================================      |  82%
|
|==================================================================     |  83%
|
|===================================================================    |  84%
|
|====================================================================   |  84%
|
|=====================================================================  |  85%
|
|====================================================================== |  85%
|
|======================================================================= |  86%
|
|========================================================================|  86%
|
|========================================================================|  87%
|
|========================================================================|  88%
|
|========================================================================|  88%
|
|========================================================================|  89%
|
|========================================================================|  89%
|
|========================================================================|  90%
|
|========================================================================|  91%
|
|========================================================================|  91%
|
|========================================================================|  92%
|
|========================================================================|  92%
|
|========================================================================|  93%
|
|========================================================================|  94%
|
|========================================================================|  94%
```

```
|
|=================================================================     |  95%
|
|==================================================================    |  95%
|
|===================================================================   |  96%
|
|====================================================================  |  96%
|
|===================================================================== |  97%
|
|====================================================================== |  98%
|
|======================================================================= |  98%
|
|======================================================================== |  99%
|
|=========================================================================|  99%
|
|=========================================================================| 100%
```

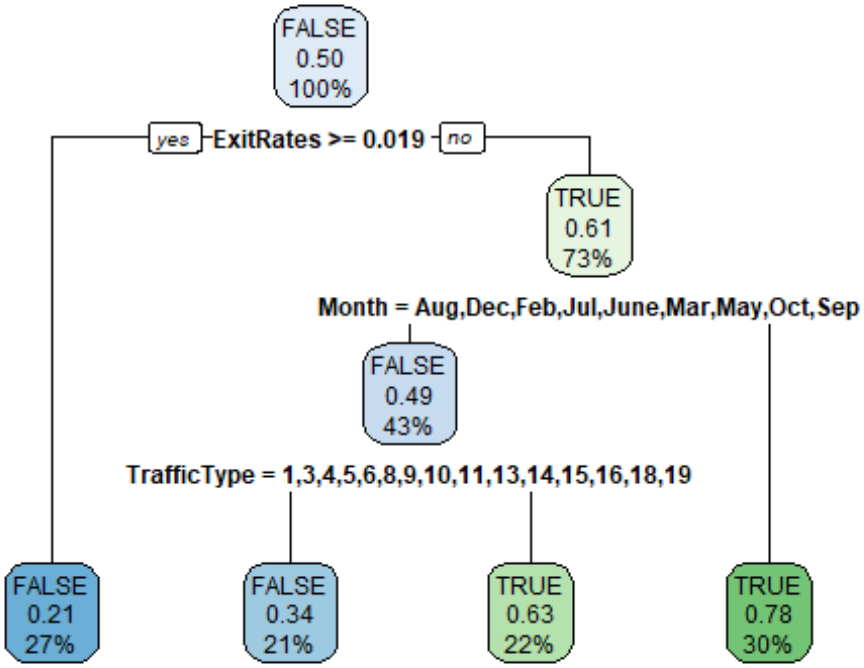
```r
# Class distribution after SMOTE
Y <- table (df$Revenue)
barplot(Y, main = "Distribution of class variable", xlab = "Revenue", ylab =
"frequency", col = blues9)
```



```r
# Scaling

Y <-
c("Administrative","Administrative_Duration","Informational","Informational_D
uration","ProductRelated_Duration","ExitRates","PageValues")
for (j in Y) {
  df[,j] <- scale(df[,j])

}

summary(df)
```

```
##   Administrative.V1  Administrative_Duration.V1  Informational.V1
##  Min.   :-0.814608   Min.   :-0.530245          Min.   :-0.460093
##  1st Qu.:-0.814608   1st Qu.:-0.530245          1st Qu.:-0.460093
##  Median :-0.366052   Median :-0.369864          Median :-0.460093
##  Mean   : 0.000000   Mean   : 0.000000          Mean   : 0.000000
##  3rd Qu.: 0.410765   3rd Qu.: 0.110504          3rd Qu.: 0.088239
##  Max.   : 7.174024   Max.   :18.631435          Max.   :17.845091
##
##  Informational_Duration.V1 ProductRelated_Duration.V1    ExitRates.V1
##  Min.   :-0.283980         Min.   :-0.703392          Min.   :-0.854843
##  1st Qu.:-0.283980         1st Qu.:-0.555823          1st Qu.:-0.527104
##  Median :-0.283980         Median :-0.315042          Median :-0.317015
##  Mean   : 0.000000         Mean   : 0.000000          Mean   : 0.000000
##  3rd Qu.:-0.257117         3rd Qu.: 0.154367          3rd Qu.: 0.062826
##  Max.   :17.310563         Max.   :29.697496          Max.   : 4.388980
##
##     PageValues.V1    SpecialDay      Month      OperatingSystems
##  Min.   :-0.543071   0  :19334   Nov    :7588   2      :13262
##  1st Qu.:-0.543071   0.2:  178   May    :5028   1      : 3686
##  Median :-0.527962   0.4:  243   Dec    :2413   3      : 3026
##  Mean   : 0.000000   0.6:  361   Mar    :2392   4      :  513
##  3rd Qu.: 0.187555   0.8:  324   Oct    : 900   8      :   75
##  Max.   :13.636028   1  :  154   Sep    : 655   6      :   19
##                                  (Other):1618   (Other):   13
##     Browser          Region      TrafficType              VisitorType
##  2      :15061   1      :9903   2      :9661   New_Visitor      : 2911
##  1      : 3438   3      :3923   1      :3511   Other            :   81
##  4      :  871   2      :1588   3      :2496   Returning_Visitor:17602
##  5      :  512   4      :1556   4      :1448
##  6      :  175   7      :1037   13     : 807
##  10     :  169   6      :1030   10     : 591
##  (Other):  368   (Other):1557   (Other):2080
##   Weekend       Revenue
##  FALSE:16805   FALSE:10297
##  TRUE : 3789   TRUE :10297
##
##
##
```

```
##
##

# Splitting the data into train(80%) and test(20%)

df <- df[,-7]
names(df)

##  [1] "Administrative"          "Administrative_Duration"
##  [3] "Informational"           "Informational_Duration"
##  [5] "ProductRelated_Duration" "ExitRates"
##  [7] "SpecialDay"              "Month"
##  [9] "OperatingSystems"        "Browser"
## [11] "Region"                  "TrafficType"
## [13] "VisitorType"             "Weekend"
## [15] "Revenue"

train.index <- df$Revenue%>%
  createDataPartition(p = 0.8, list = FALSE)
train <- df[train.index,]
test <- df[-train.index,]

# Classification tree
CT_Model <- rpart(train$Revenue~., data = train)
CT_Model

## n= 16476
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 16476 8238 FALSE (0.5000000 0.5000000)
##    2) ExitRates>=0.01866119 4518  954 FALSE (0.7888446 0.2111554) *
##    3) ExitRates< 0.01866119 11958 4674 TRUE (0.3908680 0.6091320)
##      6) Month=Aug,Dec,Feb,Jul,June,Mar,May,Oct,Sep 7013 3431 FALSE
(0.5107657 0.4892343)
##       12) TrafficType=1,3,4,5,6,8,9,10,11,13,14,15,16,18,19 3378 1149
FALSE (0.6598579 0.3401421) *
##       13) TrafficType=2,7,20 3635 1353 TRUE (0.3722146 0.6277854) *
##      7) Month=Nov 4945 1092 TRUE (0.2208291 0.7791709) *

rpart.plot(CT_Model)
```




```
CT_pred <- predict(CT_Model, test, type = 'class')
confusionMatrix(CT_pred,test$Revenue)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1497  543
##      TRUE    562 1516
##
##                Accuracy : 0.7317
##                  95% CI : (0.7179, 0.7452)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.4633
##
##  Mcnemar's Test P-Value : 0.5882
##
##             Sensitivity : 0.7271
##             Specificity : 0.7363
##          Pos Pred Value : 0.7338
##          Neg Pred Value : 0.7295
##              Prevalence : 0.5000
##          Detection Rate : 0.3635
##    Detection Prevalence : 0.4954
##       Balanced Accuracy : 0.7317
```

```
##
##        'Positive' Class : FALSE
##
```

```
print("Accuracy")
```

```
## [1] "Accuracy"
```

```
Accuracy(y_pred = CT_pred, y_true = test$Revenue)
```

```
## [1] 0.7316659
```

```
print("Precision")
```

```
## [1] "Precision"
```

```
Precision(y_pred = CT_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.7338235
```

```
print("Recall")
```

```
## [1] "Recall"
```

```
Recall(y_pred = CT_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.727052
```

```
print("F1 Score")
```

```
## [1] "F1 Score"
```

```
F1_Score(y_pred = CT_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.7304221
```

```
# c) SVM Model


SVM_Model <- svm(train$Revenue~., data=train, kernel="linear",
cost=0.10,scale=FALSE)
summary(SVM_Model)
```

```
##
## Call:
## svm(formula = train$Revenue ~ ., data = train, kernel = "linear",
##     cost = 0.1, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
```

```
## Number of Support Vectors:  9037
##
##  ( 4524 4513 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
SVM_pred <- predict(SVM_Model, test)
confusionMatrix(SVM_pred,test$Revenue)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1499  413
##      TRUE    560 1646
##
##                Accuracy : 0.7637
##                  95% CI : (0.7504, 0.7766)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5274
##
##  Mcnemar's Test P-Value : 2.861e-06
##
##             Sensitivity : 0.7280
##             Specificity : 0.7994
##          Pos Pred Value : 0.7840
##          Neg Pred Value : 0.7461
##              Prevalence : 0.5000
##          Detection Rate : 0.3640
##    Detection Prevalence : 0.4643
##       Balanced Accuracy : 0.7637
##
##        'Positive' Class : FALSE
##
```

```
print("Accuracy")
```

```
## [1] "Accuracy"
```

```
Accuracy(y_pred = SVM_pred, y_true = test$Revenue)
```

```
## [1] 0.7637203
```

```
print("Precision")
```

```
## [1] "Precision"
```

```r
Precision(y_pred = SVM_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.7839958
```

```r
print("Recall")
```

```
## [1] "Recall"
```

```r
Recall(y_pred = SVM_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.7280233
```

```r
print("F1 Score")
```

```
## [1] "F1 Score"
```

```r
F1_Score(y_pred = SVM_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.7549736
```

```r
# Random Forest Model
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.2.3
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
RF_Model <- randomForest(train$Revenue~., data = train, ntree = 500)
summary(RF_Model)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted      16476   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes          32952   matrix numeric
## oob.times      16476   -none- numeric
## classes            2   -none- character
## importance        14   -none- numeric
## importanceSD       0   -none- NULL
```

```
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y               16476  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
RF_pred <- predict(RF_Model, test)
confusionMatrix(RF_pred,test$Revenue)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1741  268
##      TRUE    318 1791
##
##                Accuracy : 0.8577
##                  95% CI : (0.8467, 0.8682)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7154
##
##  Mcnemar's Test P-Value : 0.04295
##
##             Sensitivity : 0.8456
##             Specificity : 0.8698
##          Pos Pred Value : 0.8666
##          Neg Pred Value : 0.8492
##              Prevalence : 0.5000
##          Detection Rate : 0.4228
##    Detection Prevalence : 0.4879
##       Balanced Accuracy : 0.8577
##
##        'Positive' Class : FALSE
##
```

```
print("Accuracy")
```

```
## [1] "Accuracy"
```

```
Accuracy(y_pred = RF_pred, y_true = test$Revenue)
```

```
## [1] 0.8576979
```

```
print("Precision")
```

```
## [1] "Precision"
```

```r
Precision(y_pred = RF_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.8666003
```

```r
print("Recall")
```

```
## [1] "Recall"
```

```r
Recall(y_pred = RF_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.8455561
```

```r
print("F1 Score")
```

```
## [1] "F1 Score"
```

```r
F1_Score(y_pred = RF_pred, y_true = test$Revenue, positive = NULL)
```

```
## [1] 0.8559489
```