Projeto1_FraudDetectionFinal.R

mrp

2021-11-21

```
################################################################################
#                                                                              #
# This project was developed during my studies in the course Big Data Analytics  #
# with R and Azure2.0, offered by Data Science Academy.                       #
# (www.datascienceacademy.com.br).                                             #
#                                                                              #
################################################################################
#                                                                              #
# This project provides a simple predictive model to support the fraud detection #
# in traffic of clicks on mobile applications advertising. Three algorithms,    #
# including Random Forest, Naive Bayes and Support Vector Machines (SVM) were    #
# tested in this project.                                                       #
#                                                                              #
# It is relevant to mention that the train dataset was not used during the      #
# development of the model because the data set contains too much data to be     #
# processed in my machine, especially using R for this purpose. Therefore, the   #
# model creation was based on the train_sample data set.                        #
#                                                                              #
################################################################################

################################################################################
#                                                                              #
#                                    PART I                                    #
#                                                                              #
################################################################################


############################### LOADING PACKAGES #################################

# Loading packages
library(data.table)
library(dplyr)
```

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

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

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

```
library(timetk)
```

```
##
## Attaching package: 'timetk'

## The following object is masked from 'package:data.table':
##
##     :=
```

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(ggplot2)
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
library(randomForest)
```

```
## randomForest 4.6-14

## 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
```

```
library(e1071)

# Set seed for reproducibility purposes
set.seed(12345)
```

```r
##################### DATA LOADING AND PRE-PROCESSING #########################

# Loading training_sample dataset
data = fread('data/train_sample.csv')

# Visualizing the data
head(data)
```

```
##        ip app device os channel          click_time attributed_time
## 1:  87540  12      1 13     497 2017-11-07 09:30:38            <NA>
## 2: 105560  25      1 17     259 2017-11-07 13:40:27            <NA>
## 3: 101424  12      1 19     212 2017-11-07 18:05:24            <NA>
## 4:  94584  13      1 13     477 2017-11-07 04:58:08            <NA>
## 5:  68413  12      1  1     178 2017-11-09 09:00:09            <NA>
## 6:  93663   3      1 17     115 2017-11-09 01:22:13            <NA>
##    is_attributed
## 1:             0
## 2:             0
## 3:             0
## 4:             0
## 5:             0
## 6:             0
```

```r
# Checking data types of variables
glimpse(data)
```

```
## Rows: 100,000
## Columns: 8
## $ ip              <int> 87540, 105560, 101424, 94584, 68413, 93663, 17059, 121~
## $ app             <int> 12, 25, 12, 13, 12, 3, 1, 9, 2, 3, 3, 3, 3, 6, 2, 25, ~
## $ device          <int> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, ~
## $ os              <int> 13, 17, 19, 13, 1, 17, 17, 25, 22, 19, 22, 13, 22, 20,~
## $ channel         <int> 497, 259, 212, 477, 178, 115, 135, 442, 364, 135, 489,~
## $ click_time      <dttm> 2017-11-07 09:30:38, 2017-11-07 13:40:27, 2017-11-07 ~
## $ attributed_time <dttm> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ is_attributed   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

```r
# Checking for na values, considering each variable
sapply(colnames(data), function(x) {sum(is.na(data[,..x]))})
```

```
##              ip             app          device              os         channel
##               0               0               0               0               0
##      click_time attributed_time   is_attributed
##               0           99773               0
```

```r
# As the variable attributed_time presents more than 99% of missing data, this
# variable was removed.

# Transforming categorical variables to categorical type
data = data %>%
  select(-attributed_time) %>%
  mutate(ip = as.factor(ip),
         device = as.factor(device),
         os = as.factor(os),
         app = as.factor(app),
```

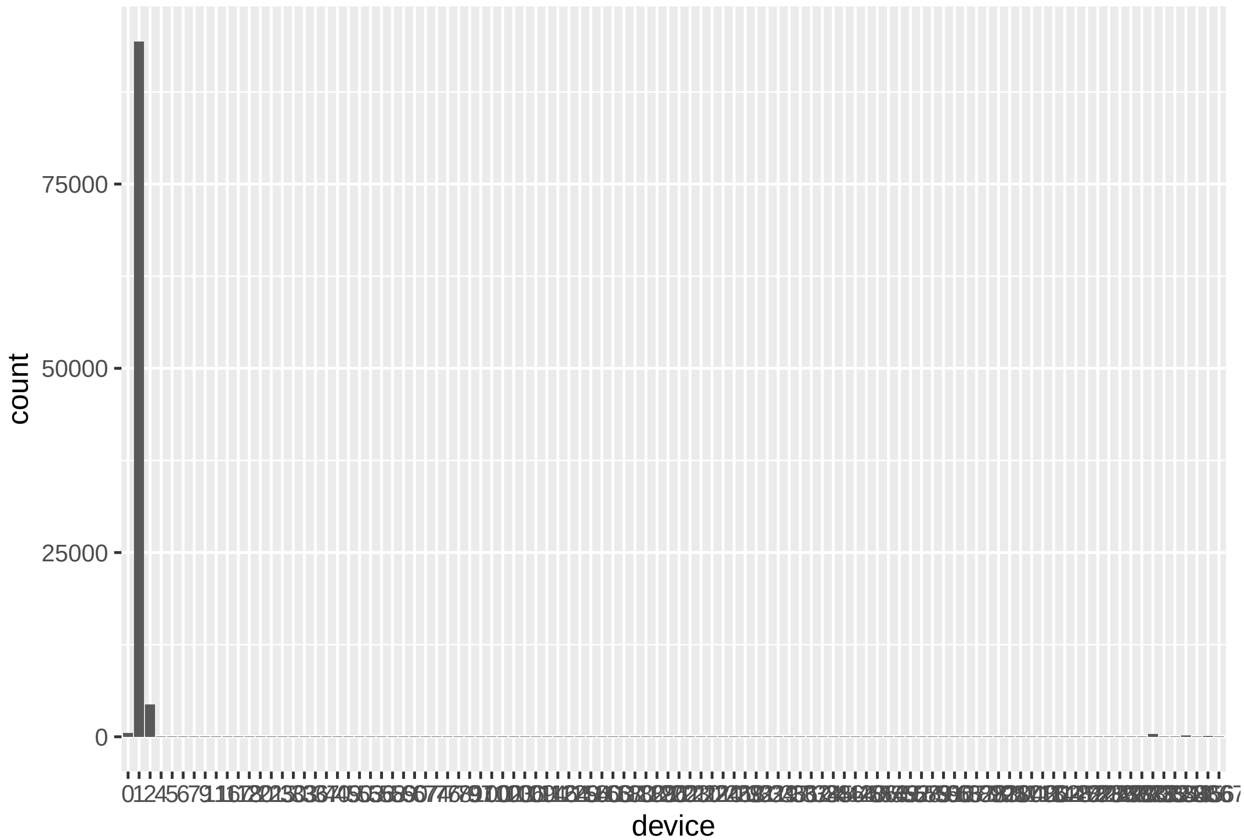
```r
        channel = as.factor(channel),
        is_attributed = as.factor(is_attributed))

###################### PRELIMINAR EXPLORATORY ANALYSIS ########################

# Visualizing some descriptive statistics
summary(data)
```

```
##        ip              app             device            os
##  5348   :  669   3      :18279   1      :94338   19     :23870
##  5314   :  616   12     :13198   2      : 4345   13     :21223
##  73487  :  439   2      :11737   0      :  541   17     : 5232
##  73516  :  399   9      : 8992   3032   :  371   18     : 4830
##  53454  :  280   15     : 8595   3543   :  151   22     : 4039
##  114276 :  219   18     : 8315   3866   :   93   10     : 2816
##  (Other):97378   (Other):30884   (Other):  161   (Other):37990
##     channel         click_time                  is_attributed
##  280    : 8114   Min.   :2017-11-06 16:00:00   0:99773
##  245    : 4802   1st Qu.:2017-11-07 11:34:09   1:  227
##  107    : 4543   Median :2017-11-08 07:07:50
##  477    : 3960   Mean   :2017-11-08 06:29:52
##  134    : 3224   3rd Qu.:2017-11-09 02:06:01
##  259    : 3130   Max.   :2017-11-09 15:59:51
##  (Other):72227
```

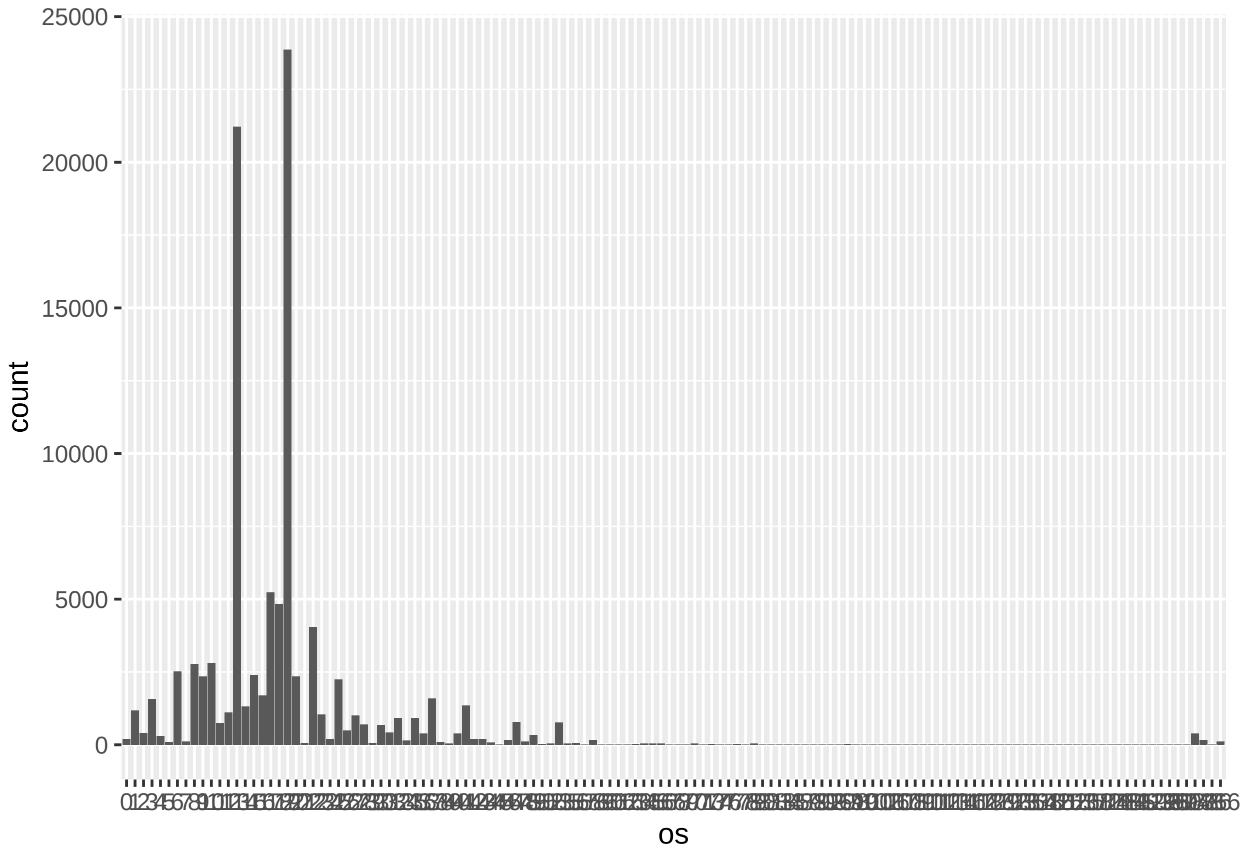
```r
# Checking the number of categories, considering each variable
apply(X = data, MARGIN = 2, FUN = function(x) {length(unique(x))})
```

```
##            ip           app        device            os       channel
##         34857           161           100           130           161
##    click_time is_attributed
##         80350             2
```

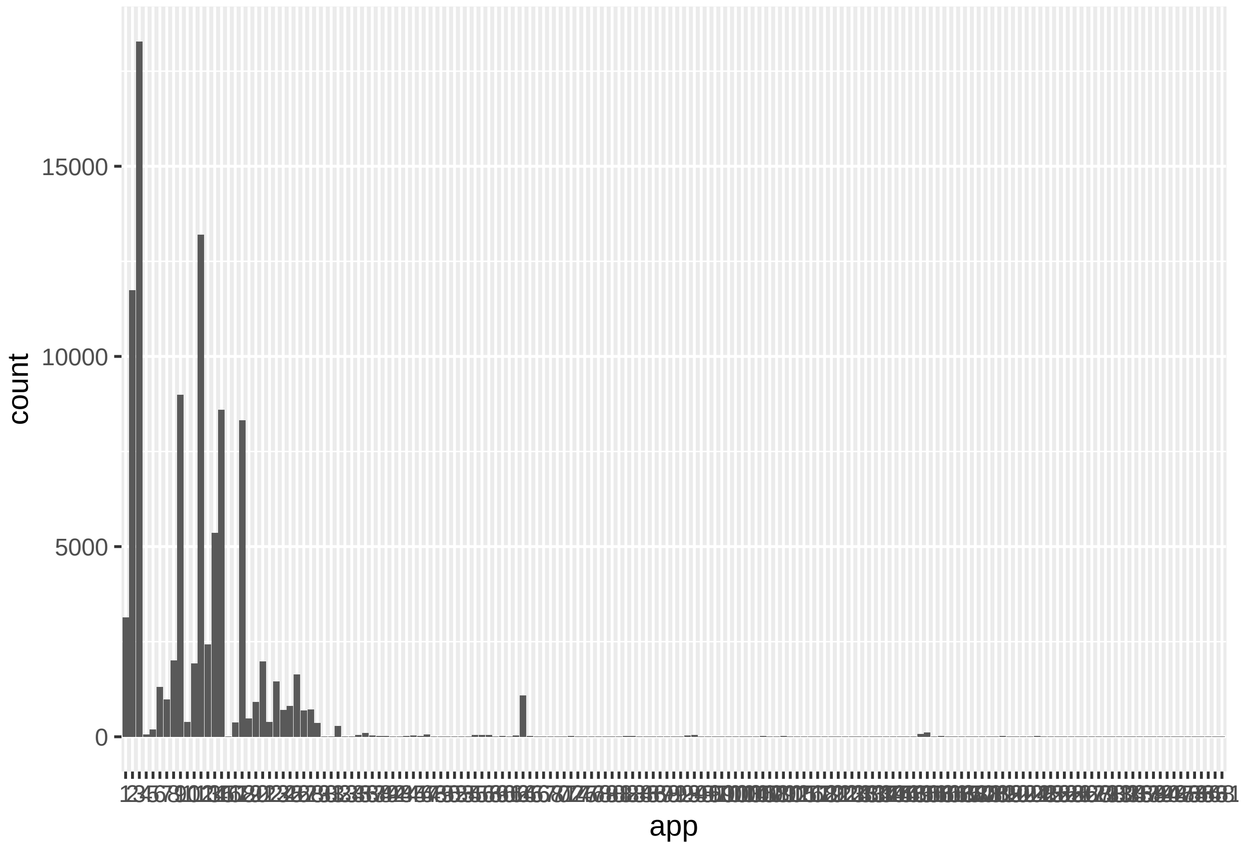
```r
# From the above analysis, both ip and click_time presents a lot of categories,
# and therefore, shouldn't be used in the model prediction. In this case, the
# click_time variable is almost like an ID.
# Let's plot some graphs to visualize the variables distribution.

# Distribution of device
ggplot(data = data) +
  geom_bar(aes(device))
```



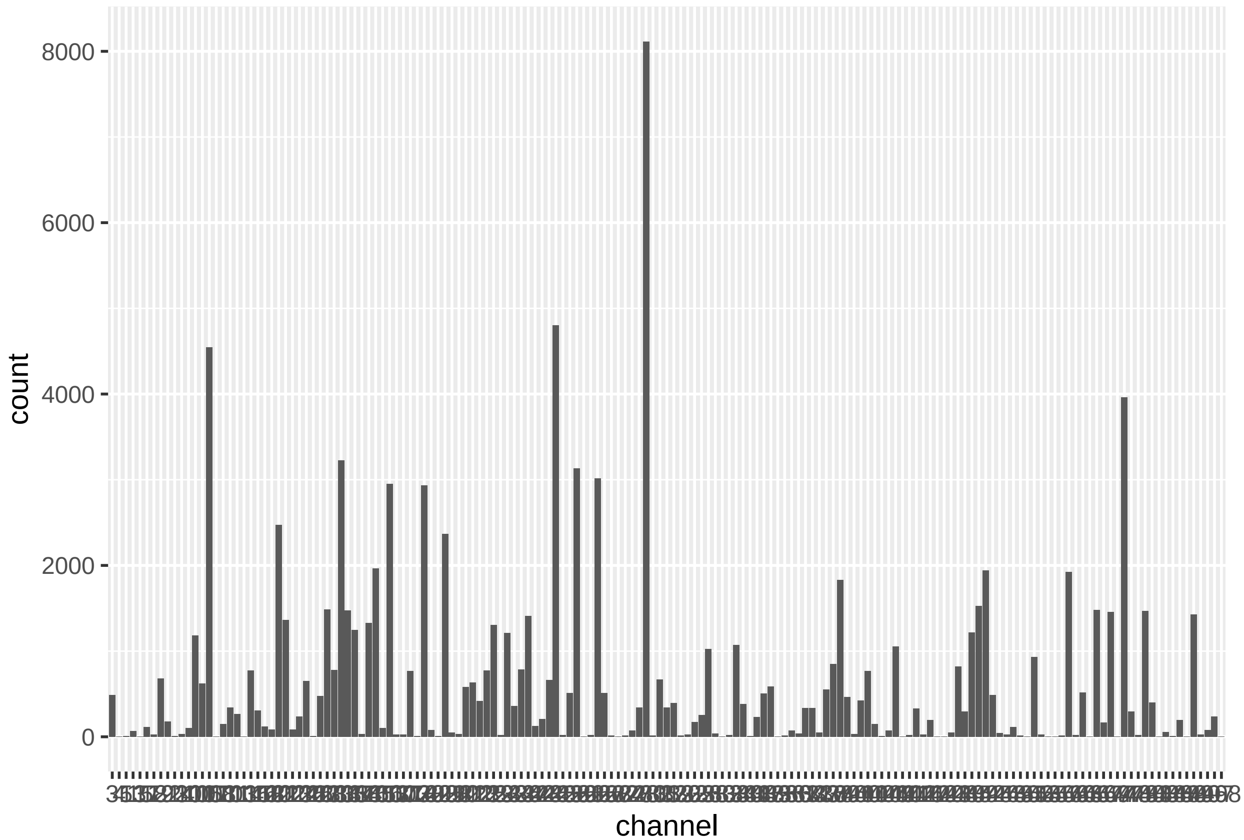

```r
# Distribution of os
ggplot(data = data) +
  geom_bar(aes(os))
```



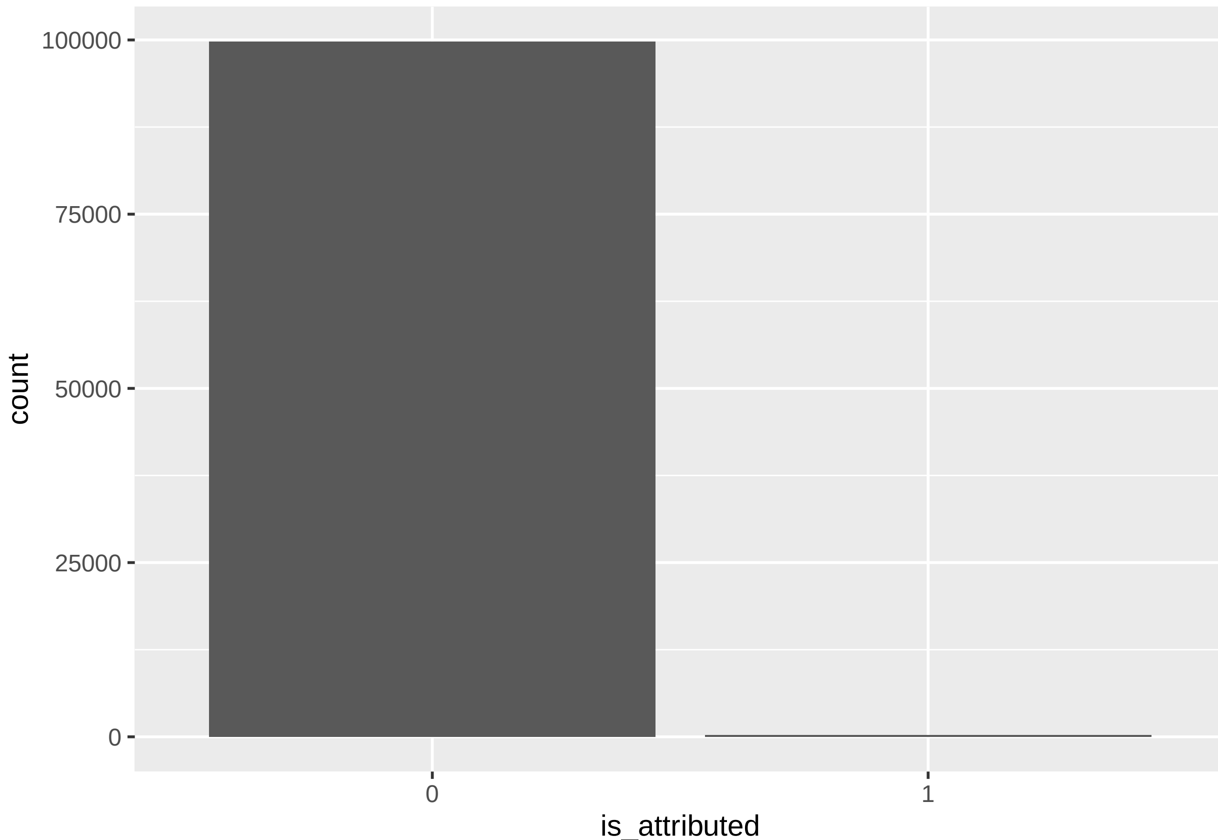

```r
# Distribution of app
ggplot(data = data) +
  geom_bar(aes(app))
```




```r
# Distribution of channel
ggplot(data = data) +
  geom_bar(aes(channel))
```




```r
# Distribution of is_attributed (target)
ggplot(data = data) +
  geom_bar(aes(is_attributed))
```




```
# It is relevant to mention that the target variable is unbalanced.

# The data balance step will be carried out only in the train set, therefore,
# it will be applied following the partition of the data in train and test sets.

################################ DATA PARTIONING ################################

# Creating a random index to separate data into train (80%) and test (20%)
train_index <- createDataPartition(data$is_attributed, p = 0.8, list = FALSE)

# Creating train and test sets based on train.index
train = data[ train_index,]
test  = data[-train_index,]

# Removing data, train_index and free unused memory
rm(train_index)
rm(data)
invisible(gc())

############################## DATA BALANCING ###################################

# Let's try three different balancing techniques:

### undersampling;
### oversampling with package ROSE;
### mix of undersampling and oversampling.
```

```r
# Checking unbalanced in train set
table(train$is_attributed)
```

```
##
##     0     1
## 79819   182
```

```r
# let's transform the click_time variable for factor type for balancing purposes,
# as the function ROSE cannot handle datetime type
train$click_time = as.factor(train$click_time)

### UNDERSAMPLING

# Separate majority and minority classes
train_majority = train[train$is_attributed==0]
train_minority = train[train$is_attributed==1]

# Creating a samples without replacement from train_majority based on the ip.
# The value XXX was select to produce a majority sample not too big compared
# to the minority sample
majority_under = sample(train_majority$ip, 100, replace=FALSE)

# Creating and filling in the indexes list for collecting all the repeated ips in
# the train set and considering the majority_under
j = 1
under_index = list()
for (i in 1:length(train$ip)) {
  if (train[i]$ip %in% majority_under) {
    under_index[[j]] = i
    j = j + 1}
}

# Creating the train_under with only the information of the majority_under
train_under = train[unlist(under_index), ]

# Binding the information of train_minority into the train_under
train_under = rbind(train_under, train_minority)

# Checking the distribution of target variable after balancing using the
# under sampling technique
table(train_under$is_attributed)
```

```
##
##    0    1
## 2795  184
```

```r
# Checking the data types of train_unde
glimpse(train_under)
```

```
## Rows: 2,979
## Columns: 7
## $ ip              <fct> 5348, 63925, 5348, 5348, 73487, 105475, 53454, 105475, 7~
## $ app             <fct> 8, 22, 58, 18, 3, 7, 18, 27, 53, 204, 12, 9, 3, 1, 9, 15~
## $ device          <fct> 1, 2, 3866, 1, 1, 1, 2, 1, 3866, 3543, 1, 1, 1, 1, 1, 1,~
## $ os              <fct> 11, 17, 866, 8, 8, 40, 97, 32, 866, 748, 19, 13, 25, 30,~
```

```
## $ channel       <fct> 145, 496, 347, 107, 280, 101, 121, 153, 347, 347, 409, 4~
## $ click_time    <fct> 2017-11-08 13:17:06, 2017-11-07 08:52:29, 2017-11-09 14:~
## $ is_attributed <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

```
### OVERSAMPLING

# Balancing the data using the package and function ROSE.
train_over = ROSE(is_attributed ~ ., data = train, N = 158000, seed=111)$data

# Checking the distribution of target variable after balancing using the
# oversampling technique using the function ROSE
table(train_over$is_attributed)
```

```
##
##     0     1
## 78907 79093
```

```
# Checking the data types of train_over
glimpse(train_over)
```

```
## Rows: 158,000
## Columns: 7
## $ ip            <fct> 77048, 114488, 36183, 108393, 99768, 53454, 58057, 68651~
## $ app           <fct> 9, 15, 2, 12, 2, 1, 21, 12, 9, 9, 14, 12, 3, 2, 9, 1, 2,~
## $ device        <fct> 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ os            <fct> 12, 25, 100, 3, 13, 17, 17, 18, 10, 2, 1, 11, 40, 6, 47,~
## $ channel       <fct> 466, 245, 205, 140, 219, 135, 128, 245, 334, 445, 118, 4~
## $ click_time    <fct> 2017-11-09 07:37:36, 2017-11-08 13:53:49, 2017-11-07 10:~
## $ is_attributed <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

```
### UNDERSAMPLING AND OVERSAMPLING

# Based on the number of observations for is_attributed == 0 for train_under,
# an over sample based on the train_under set was created using ROSE, resulting
# in the train_underOver set
train_underOver = ROSE(is_attributed ~ ., data = train_under, N = 4000, seed=111)$data

# Checking the distribution of target variable after balancing using the
# under sampling followed by over sampling techniques for train_under
table(train_underOver$is_attributed)
```

```
##
##    0    1
## 2010 1990
```

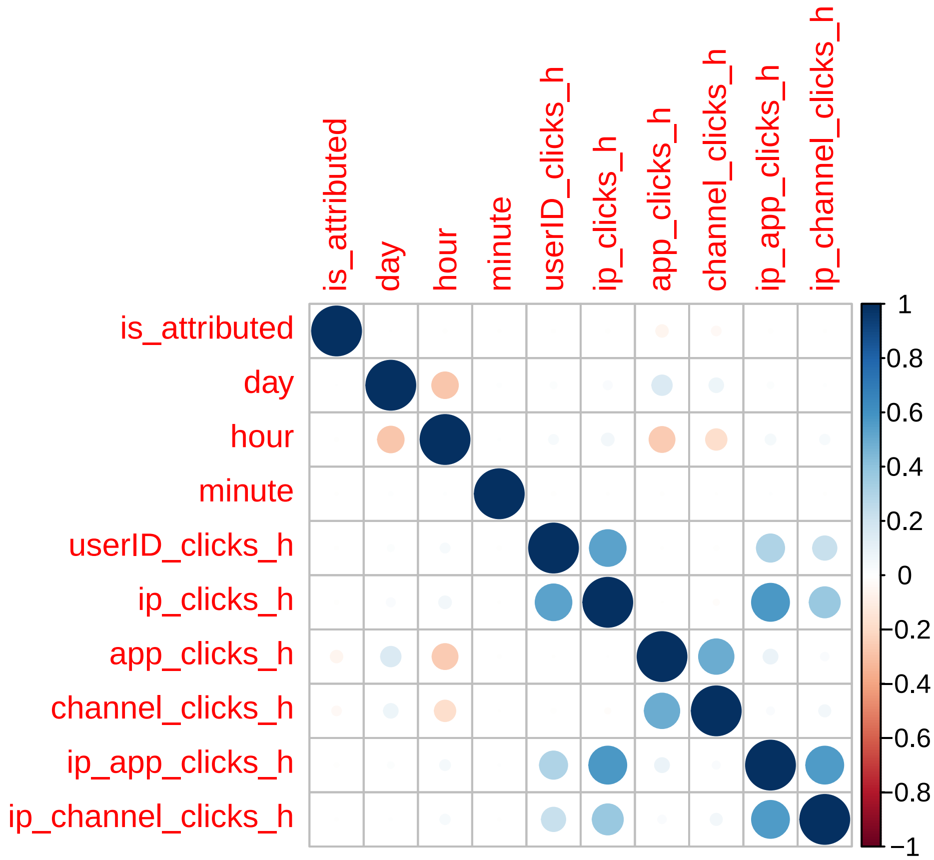
```
# Transforming click_time of train_under, train_over and train_underOver back to
# datetime type
train_over$click_time = ymd_hms(train_over$click_time)
train_under$click_time = ymd_hms(train_under$click_time)
train_underOver$click_time = ymd_hms(train_underOver$click_time)

############################ FEATURE ENGINEERING ###############################

# Let's consider that a click is associated to an specific user, which
# needs an 'os' installed in a 'device' using a specific 'ip' address. So, let's
# create a variable named 'userID' to represent this.
```

```r
# Let's consider the number of clicks in one hour per userID, ip, device, os, app, channel and
# create a few variables.

### day - day of the month that occurred an specific click;
### hour - hour of the day that occurred an specific click;
### minute - minute that occurred an specific click;
### userID_clicks_h - number of clicks per userID (ip + device + os) per hour;
### ip_clicks_h - number of clicks from a specific ip per hour;
### app_clicks_h - number of clicks in a specific app per hour;
### channel_clicks_h - number of clicks in a specific channel per hour;
### ip_app_clicks_h - number of clicks using a combination of ip and app per hour;
### ip_channel_clicks_h - number of clicks using a combination of ip and channel
# per hour.

# Creating a function to process the train and test (later on) sets
process_data <- function(df) {
  df = df %>%
    mutate(day = day(click_time),
           hour = hour(click_time),
           minute = minute(click_time)) %>%
    add_count(ip, device, os, day, hour) %>% rename(userID_clicks_h = n) %>%
    add_count(ip, day, hour) %>% rename(ip_clicks_h = n) %>%
    add_count(app, day, hour) %>% rename(app_clicks_h = n) %>%
    add_count(channel, day, hour) %>% rename(channel_clicks_h = n) %>%
    add_count(ip, app, day, hour) %>% rename(ip_app_clicks_h = n) %>%
    add_count(ip, channel, day, hour) %>% rename(ip_channel_clicks_h = n) %>%
    mutate(ip = NULL, click_time = NULL)
}

# Transform click_time of train set to datetime type
train$click_time = ymd_hms(train$click_time)

# Processing train
train = process_data(train)

# Processing train_over
train_over = process_data(train_over)

# Processing train_under
train_under = process_data(train_under)

# Processing train_underOver
train_underOver = process_data(train_underOver)

##############################  FEATURE SELECTION ################################

# Let's check for correlation considering our new variables and target variable

# Transforming the target variable to integer only for correlation purposes
train$is_attributed = as.integer(train$is_attributed)
train_over$is_attributed = as.integer(train_over$is_attributed)
train_under$is_attributed = as.integer(train_under$is_attributed)
train_underOver$is_attributed = as.integer(train_underOver$is_attributed)
```

```r
# Checking correlation on train
train_cor = cor(train[,c(5:14)])
train_cor
```

```
##                    is_attributed           day         hour        minute
## is_attributed       1.000000e+00  2.337994e-05 -0.004183406 -0.0022052854
## day                 2.337994e-05  1.000000e+00 -0.273322565  0.0049690953
## hour               -4.183406e-03 -2.733226e-01  1.000000000  0.0012856138
## minute             -2.205285e-03  4.969095e-03  0.001285614  1.0000000000
## userID_clicks_h    -3.501540e-03  1.498425e-02  0.033118164 -0.0049683536
## ip_clicks_h        -4.633191e-03  2.683992e-02  0.057400717 -0.0013564390
## app_clicks_h       -5.508750e-02  1.568939e-01 -0.258974134 -0.0029601268
## channel_clicks_h   -3.184878e-02  7.747184e-02 -0.173583994 -0.0005267984
## ip_app_clicks_h    -5.639064e-03  1.310200e-02  0.040503545  0.0010818933
## ip_channel_clicks_h -2.387282e-03  2.038519e-03  0.036219881 -0.0021123328
##                    userID_clicks_h   ip_clicks_h  app_clicks_h
## is_attributed          -0.003501540 -4.633191e-03 -5.508750e-02
## day                     0.014984254  2.683992e-02  1.568939e-01
## hour                    0.033118164  5.740072e-02 -2.589741e-01
## minute                 -0.004968354 -1.356439e-03 -2.960127e-03
## userID_clicks_h         1.000000000  5.326322e-01 -1.210354e-03
## ip_clicks_h             0.532632193  1.000000e+00 -6.559973e-05
## app_clicks_h           -0.001210354 -6.559973e-05  1.000000e+00
## channel_clicks_h       -0.007063484 -1.110633e-02  4.945489e-01
## ip_app_clicks_h         0.308535814  5.706450e-01  8.039337e-02
## ip_channel_clicks_h     0.226203088  3.770681e-01  2.357901e-02
##                    channel_clicks_h ip_app_clicks_h ip_channel_clicks_h
## is_attributed          -0.0318487770    -0.005639064        -0.002387282
## day                     0.0774718385     0.013101999         0.002038519
## hour                   -0.1735839940     0.040503545         0.036219881
## minute                 -0.0005267984     0.001081893        -0.002112333
## userID_clicks_h        -0.0070634837     0.308535814         0.226203088
## ip_clicks_h            -0.0111063265     0.570644987         0.377068105
## app_clicks_h            0.4945488537     0.080393366         0.023579011
## channel_clicks_h        1.0000000000     0.020279285         0.051205210
## ip_app_clicks_h         0.0202792847     1.000000000         0.568612529
## ip_channel_clicks_h     0.0512052103     0.568612529         1.000000000
```

```r
corrplot(train_cor)
```

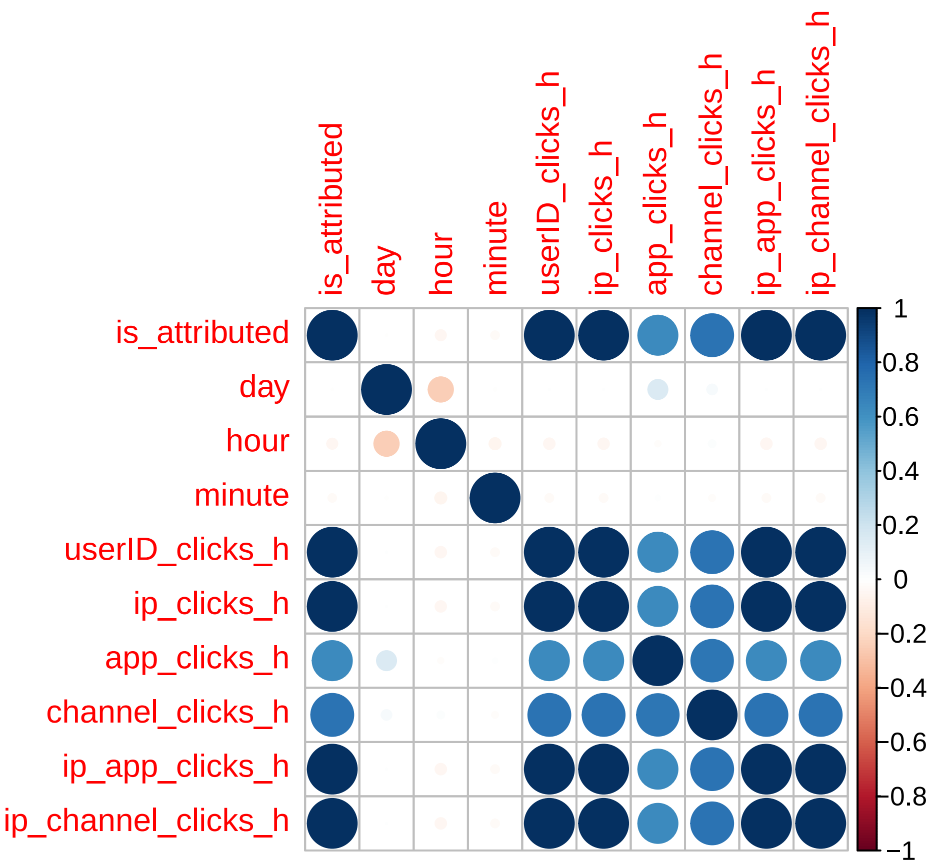



```
# There was no strong correlation between target variable and predictors using
# the original train set without any balance.

# Checking correlation on train_over
train_over_cor = cor(train_over[,c(5:14)])
train_over_cor
```

```
##                      is_attributed           day        hour       minute
## is_attributed          1.000000000 -0.0010668713 -0.04273077 -0.024784111
## day                   -0.001066871  1.0000000000 -0.24702904 -0.002953535
## hour                  -0.042730769 -0.2470290386  1.00000000 -0.051439756
## minute                -0.024784111 -0.0029535346 -0.05143976  1.000000000
## userID_clicks_h        0.997778890  0.0007248818 -0.04669101 -0.025927600
## ip_clicks_h            0.997240310  0.0005273523 -0.04578824 -0.025874319
## app_clicks_h           0.638549480  0.1515330356 -0.01215414  0.008977345
## channel_clicks_h       0.735193970  0.0398602603  0.01952198 -0.015144581
## ip_app_clicks_h        0.997812708  0.0006779099 -0.04669083 -0.025967447
## ip_channel_clicks_h    0.997813356  0.0006747045 -0.04670822 -0.025964033
##                      userID_clicks_h   ip_clicks_h app_clicks_h channel_clicks_h
## is_attributed           0.9977788898  0.9972403105  0.638549480       0.73519397
## day                     0.0007248818  0.0005273523  0.151533036       0.03986026
## hour                   -0.0466910053 -0.0457882369 -0.012154144       0.01952198
## minute                 -0.0259275998 -0.0258743194  0.008977345      -0.01514458
## userID_clicks_h         1.0000000000  0.9994525169  0.638529528       0.73426656
## ip_clicks_h             0.9994525169  1.0000000000  0.638464110       0.73399873
## app_clicks_h            0.6385295277  0.6384641099  1.000000000       0.72026972
## channel_clicks_h        0.7342665589  0.7339987303  0.720269721       1.00000000
```

```
## ip_app_clicks_h        0.9999114078  0.9994084137  0.638619105        0.73435399
## ip_channel_clicks_h    0.9999116432  0.9994073887  0.638607674        0.73435524
##                      ip_app_clicks_h ip_channel_clicks_h
## is_attributed           0.9978127076        0.9978133555
## day                     0.0006779099        0.0006747045
## hour                   -0.0466908308       -0.0467082210
## minute                 -0.0259674474       -0.0259640334
## userID_clicks_h         0.9999114078        0.9999116432
## ip_clicks_h             0.9994084137        0.9994073887
## app_clicks_h            0.6386191047        0.6386076741
## channel_clicks_h        0.7343539878        0.7343552398
## ip_app_clicks_h         1.0000000000        0.9999995971
## ip_channel_clicks_h     0.9999995971        1.0000000000
```

```
corrplot(train_over_cor)
```

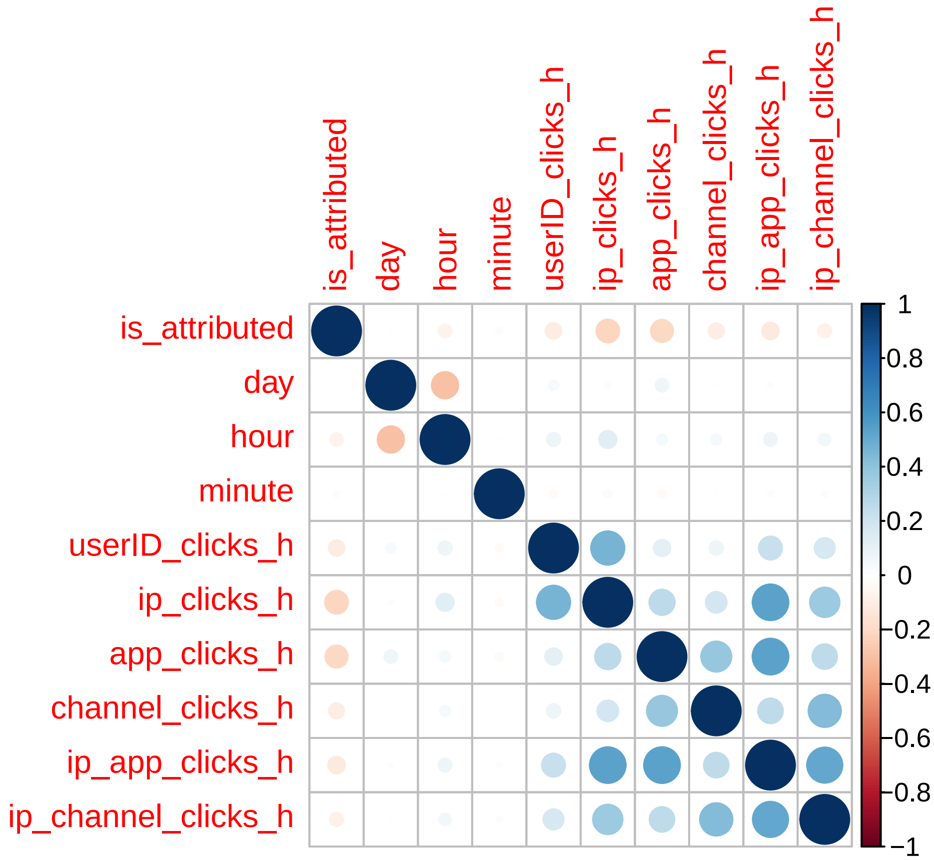



```
# In this case there appears to be a very strong linear correlation between target variable
# and userID_clicks_h, ip_clicks_h, ip_app_clicks_h and ip_channel_clicks_h
# Moreover, a relatively strong correlation between the target variable and
# app_clicks_h and channel_clicks_h was also observed.
# However, it is important to consider that most of those predictors are also
# correlated and should not be used altogether.

# Checking correlation on train_under_cor
train_under_cor = cor(train_under[,c(5:14)])
train_under_cor
```

```
##                      is_attributed          day          hour          minute
```

```
## is_attributed         1.000000000  0.0044780538 -0.069454476 -0.0172912481
## day                   0.004478054  1.0000000000 -0.290356890  0.0008137752
## hour                 -0.069454476 -0.2903568898  1.000000000  0.0072762025
## minute               -0.017291248  0.0008137752  0.007276203  1.0000000000
## userID_clicks_h      -0.102604101  0.0386031631  0.075198607 -0.0251512316
## ip_clicks_h          -0.218221990  0.0133455138  0.124712294 -0.0257905624
## app_clicks_h         -0.205825266  0.0699078285  0.045332144 -0.0287014242
## channel_clicks_h     -0.097226536  0.0050393793  0.043753893 -0.0035062905
## ip_app_clicks_h      -0.116793528  0.0126393382  0.071946410 -0.0114756994
## ip_channel_clicks_h  -0.072360004 -0.0044014402  0.057014470 -0.0168917464
##                     userID_clicks_h ip_clicks_h app_clicks_h channel_clicks_h
## is_attributed           -0.10260410 -0.21822199  -0.20582527     -0.097226536
## day                      0.03860316  0.01334551   0.06990783      0.005039379
## hour                     0.07519861  0.12471229   0.04533214      0.043753893
## minute                  -0.02515123 -0.02579056  -0.02870142     -0.003506290
## userID_clicks_h          1.00000000  0.46395368   0.11505165      0.078446998
## ip_clicks_h              0.46395368  1.00000000   0.26793589      0.180467118
## app_clicks_h             0.11505165  0.26793589   1.00000000      0.380457318
## channel_clicks_h         0.07844700  0.18046712   0.38045732      1.000000000
## ip_app_clicks_h          0.22309395  0.53388212   0.53731981      0.256678405
## ip_channel_clicks_h      0.17034284  0.36420196   0.25480530      0.437903706
##                     ip_app_clicks_h ip_channel_clicks_h
## is_attributed           -0.11679353         -0.07236000
## day                      0.01263934         -0.00440144
## hour                     0.07194641          0.05701447
## minute                  -0.01147570         -0.01689175
## userID_clicks_h          0.22309395          0.17034284
## ip_clicks_h              0.53388212          0.36420196
## app_clicks_h             0.53731981          0.25480530
## channel_clicks_h         0.25667840          0.43790371
## ip_app_clicks_h          1.00000000          0.51866529
## ip_channel_clicks_h      0.51866529          1.00000000
```

```
corrplot(train_under_cor)
```

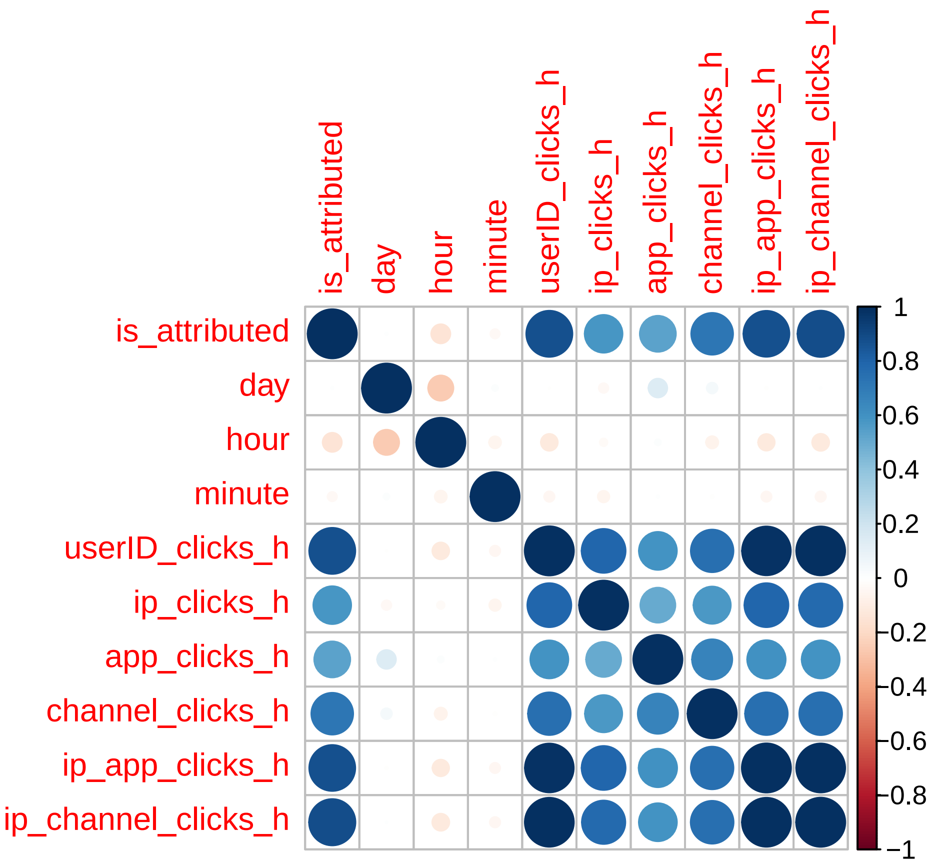



```
# No strong linear correlation was observed. However a few weak ones were observed
# In this case, let's select all variables and we can try different combinations but
# paying attention to the correlated predictors

# Checking correlation on train_under2_cor
train_underOver_cor = cor(train_underOver[,c(5:14)])
train_underOver_cor
```

```
##                     is_attributed          day        hour       minute
## is_attributed         1.000000000  0.002632042 -0.14781983 -0.034606590
## day                   0.002632042  1.000000000 -0.25989884  0.013592740
## hour                 -0.147819832 -0.259898839  1.00000000 -0.057560369
## minute               -0.034606590  0.013592740 -0.05756037  1.000000000
## userID_clicks_h       0.878067770 -0.001147684 -0.11466072 -0.042952841
## ip_clicks_h           0.588828363 -0.036795043 -0.02298851 -0.053201077
## app_clicks_h          0.533962461  0.143051090  0.01409193  0.003138878
## channel_clicks_h      0.721325635  0.046060308 -0.06010483 -0.004475516
## ip_app_clicks_h       0.873364042 -0.002959513 -0.11117945 -0.041742923
## ip_channel_clicks_h   0.887282705  0.001517575 -0.11715395 -0.043015254
##                     userID_clicks_h ip_clicks_h app_clicks_h channel_clicks_h
## is_attributed           0.878067770  0.58882836  0.533962461      0.721325635
## day                    -0.001147684 -0.03679504  0.143051090      0.046060308
## hour                   -0.114660723 -0.02298851  0.014091931     -0.060104827
## minute                 -0.042952841 -0.05320108  0.003138878     -0.004475516
## userID_clicks_h         1.000000000  0.79863688  0.594525623      0.750274069
## ip_clicks_h             0.798636885  1.00000000  0.504833403      0.571713118
## app_clicks_h            0.594525623  0.50483340  1.000000000      0.666986423
```

```
## channel_clicks_h        0.750274069  0.57171312  0.666986423      1.000000000
## ip_app_clicks_h         0.985914321  0.79909945  0.609719948      0.750792434
## ip_channel_clicks_h     0.991876642  0.77856700  0.595726090      0.757474357
##                     ip_app_clicks_h ip_channel_clicks_h
## is_attributed           0.873364042         0.887282705
## day                    -0.002959513         0.001517575
## hour                   -0.111179447        -0.117153949
## minute                 -0.041742923        -0.043015254
## userID_clicks_h         0.985914321         0.991876642
## ip_clicks_h             0.799099448         0.778566996
## app_clicks_h            0.609719948         0.595726090
## channel_clicks_h        0.750792434         0.757474357
## ip_app_clicks_h         1.000000000         0.991747195
## ip_channel_clicks_h     0.991747195         1.000000000
```

```
corrplot(train_underOver_cor)
```




```
# For the train_underOver set, it seems that the correlations results lay in
# between the previous results. There are a few strong correlations observed,
# but less strong compared to the train_over set.
# Let's try a few combination of predictors taking care with the correlated ones

# Transforming the target variable back to factor
train$is_attributed = as.factor(train$is_attributed)
train_over$is_attributed = as.factor(train_over$is_attributed)
train_under$is_attributed = as.factor(train_under$is_attributed)
train_underOver$is_attributed = as.factor(train_underOver$is_attributed)
```

```r
# Removing objects that are not need anymore and free unused memory
rm(i, j, majority_under, train_majority, train_minority, under_index)
invisible(gc())
# Now, before we go for the models' creation, let's do the same transformations
# that were done for train sets for the test set as well.

############################## PROCESSING TEST ##################################

# Processing test
test = process_data(test)

# Checking data types in test
glimpse(test)
```

```
## Rows: 19,999
## Columns: 14
## $ app                 <fct> 3, 3, 3, 1, 2, 3, 15, 23, 18, 11, 9, 2, 3, 18, 12,~
## $ device              <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ os                  <fct> 13, 18, 13, 13, 53, 13, 13, 10, 13, 19, 13, 13, 17~
## $ channel             <fct> 489, 280, 115, 115, 477, 115, 278, 153, 107, 481, ~
## $ is_attributed       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ day                 <int> 7, 7, 7, 7, 7, 7, 9, 8, 8, 8, 7, 7, 8, 8, 6, 8, 8,~
## $ hour                <int> 5, 1, 16, 17, 13, 3, 3, 16, 2, 18, 10, 10, 16, 23,~
## $ minute              <int> 3, 35, 19, 22, 28, 20, 20, 41, 7, 21, 31, 24, 4, 7~
## $ userID_clicks_h     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ ip_clicks_h         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ app_clicks_h        <int> 77, 76, 34, 6, 39, 70, 26, 6, 30, 3, 24, 44, 48, 2~
## $ channel_clicks_h    <int> 3, 37, 5, 2, 17, 2, 2, 12, 6, 1, 8, 2, 6, 13, 11, ~
## $ ip_app_clicks_h     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ ip_channel_clicks_h <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
```

```r
# Alright, all the processing is done, let's create a few models to perform
# the model selection.

############################ MODEL SELECTION ####################################

# Let's try two classification algorithms to create the models and identify
# the train sets that results in better accuracy. Then we can concentrate our
# final steps for model creation using the chosen train set.

# - Random Forest
# - Naive Bayes

###### RANDOM FOREST ######

# Checking the number of categories, considering each variable
apply(X = train, MARGIN = 2, FUN = function(x) {length(unique(x))})
```

```
##                app             device                 os            channel
##                150                 84                125                161
##      is_attributed                day               hour             minute
##                  2                  4                 24                 60
##    userID_clicks_h        ip_clicks_h       app_clicks_h   channel_clicks_h
##                  8                 21                237                127
```

```
##     ip_app_clicks_h ip_channel_clicks_h
##                   7                   6
```

```r
# With Random Forest can only handle 53 categories per variable, therefore the
# categorical variables will be not included in the random forest versions

##### TRAIN (non-balanced train set) #####

# Re-checking correlation of train
train_cor
```

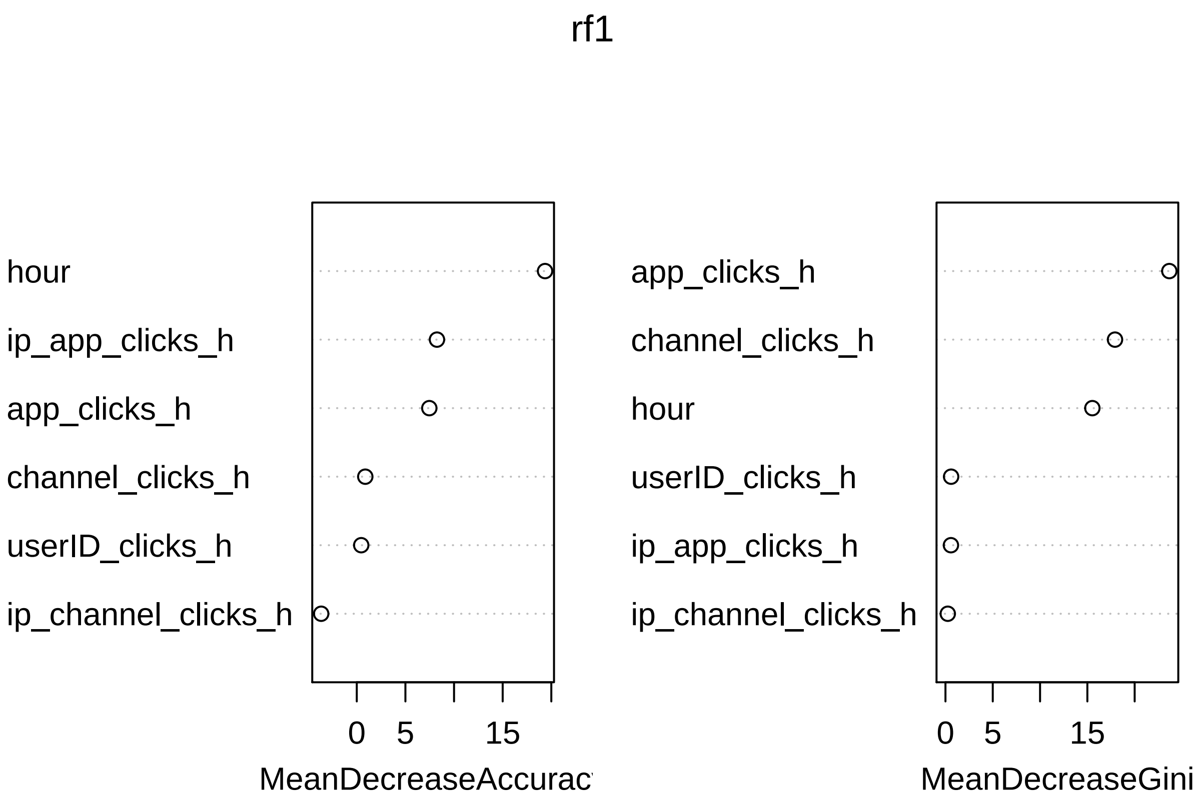
```
##                     is_attributed          day         hour        minute
## is_attributed        1.000000e+00  2.337994e-05 -0.004183406 -0.0022052854
## day                  2.337994e-05  1.000000e+00 -0.273322565  0.0049690953
## hour                -4.183406e-03 -2.733226e-01  1.000000000  0.0012856138
## minute              -2.205285e-03  4.969095e-03  0.001285614  1.0000000000
## userID_clicks_h     -3.501540e-03  1.498425e-02  0.033118164 -0.0049683536
## ip_clicks_h         -4.633191e-03  2.683992e-02  0.057400717 -0.0013564390
## app_clicks_h        -5.508750e-02  1.568939e-01 -0.258974134 -0.0029601268
## channel_clicks_h    -3.184878e-02  7.747184e-02 -0.173583994 -0.0005267984
## ip_app_clicks_h     -5.639064e-03  1.310200e-02  0.040503545  0.0010818933
## ip_channel_clicks_h -2.387282e-03  2.038519e-03  0.036219881 -0.0021123328
##                     userID_clicks_h   ip_clicks_h  app_clicks_h
## is_attributed          -0.003501540 -4.633191e-03 -5.508750e-02
## day                     0.014984254  2.683992e-02  1.568939e-01
## hour                    0.033118164  5.740072e-02 -2.589741e-01
## minute                 -0.004968354 -1.356439e-03 -2.960127e-03
## userID_clicks_h         1.000000000  5.326322e-01 -1.210354e-03
## ip_clicks_h             0.532632193  1.000000e+00 -6.559973e-05
## app_clicks_h           -0.001210354 -6.559973e-05  1.000000e+00
## channel_clicks_h       -0.007063484 -1.110633e-02  4.945489e-01
## ip_app_clicks_h         0.308535814  5.706450e-01  8.039337e-02
## ip_channel_clicks_h     0.226203088  3.770681e-01  2.357901e-02
##                     channel_clicks_h ip_app_clicks_h ip_channel_clicks_h
## is_attributed          -0.0318487770    -0.005639064        -0.002387282
## day                     0.0774718385     0.013101999         0.002038519
## hour                   -0.1735839940     0.040503545         0.036219881
## minute                 -0.0005267984     0.001081893        -0.002112333
## userID_clicks_h        -0.0070634837     0.308535814         0.226203088
## ip_clicks_h            -0.0111063265     0.570644987         0.377068105
## app_clicks_h            0.4945488537     0.080393366         0.023579011
## channel_clicks_h        1.0000000000     0.020279285         0.051205210
## ip_app_clicks_h         0.0202792847     1.000000000         0.568612529
## ip_channel_clicks_h     0.0512052103     0.568612529         1.000000000
```

```r
# Model creation
rf1 = randomForest(is_attributed ~ hour + userID_clicks_h + app_clicks_h +
                     channel_clicks_h + ip_app_clicks_h + ip_channel_clicks_h,
                   data = train, importance = TRUE)

# Visualizing the results of model
rf1
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + userID_clicks_h +      app_clicks_h + channel_clicks_h
```

```
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.23%
## Confusion matrix:
##       1 2  class.error
## 1 79818 1 1.252835e-05
## 2   182 0 1.000000e+00
```

```
# Visualizing the importance of variables
varImpPlot(rf1)
```




```
# Creating a vector to hold the observed values for the test set
obs = test$is_attributed
obs_oL = ifelse(obs == 0, 1, 2)
obs_oL = as.factor(obs_oL)

# Prediction using test data
pred_rf1 = predict(rf1, test[,-5])

# Calculating accuracy for model
accuracy_rf1 = sum(obs_oL == pred_rf1) / length(obs_oL)
accuracy_rf1
```

```
## [1] 0.9977499
```

```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf1)
```

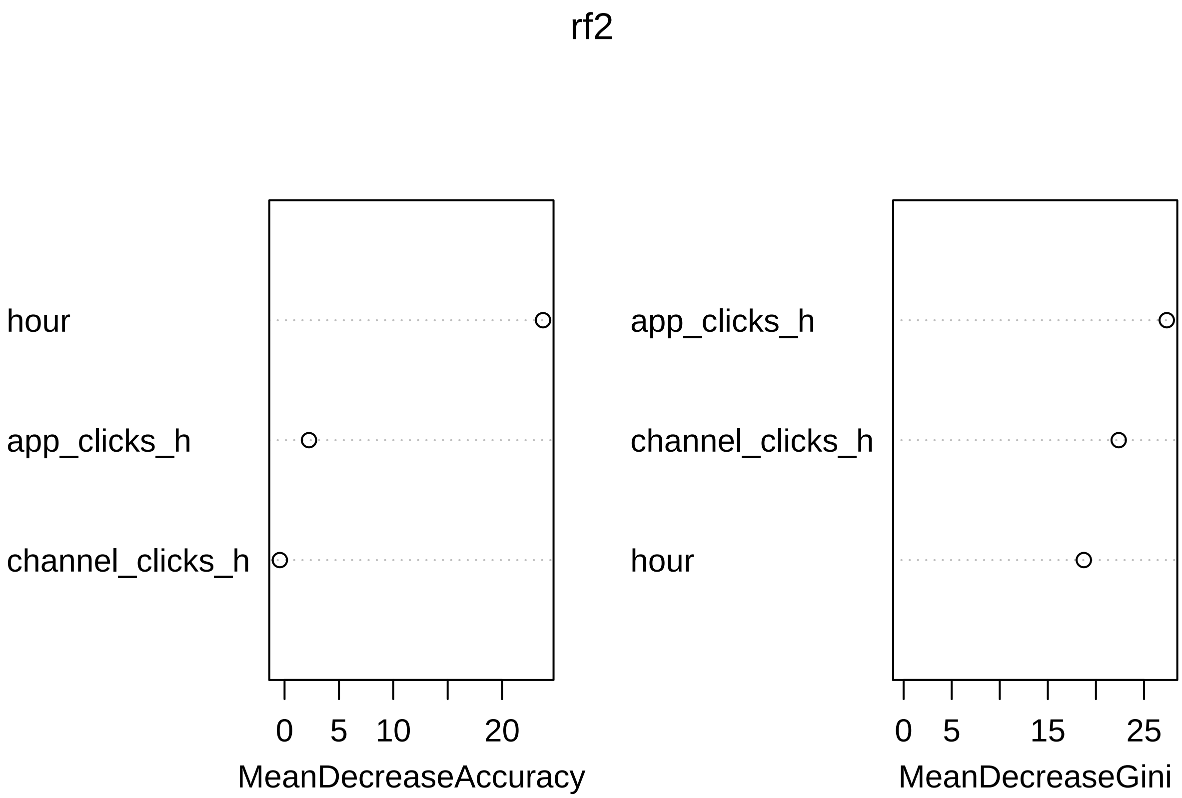
```
##       pred_rf1
```

```
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```
# The results shows that this model did not do a good job. It mistakes all
# the results from is_attributed that had a much lower number of observations.
# That's why it is important to balance the data.

# Let's create a new version of the model using the four most important variables,
# according to MeanDecreaseGini's index.


# Model creation
rf2 = randomForest(is_attributed ~ hour + app_clicks_h + channel_clicks_h,
                   data = train, importance = TRUE)

# Visualizing the results of model
rf2
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + app_clicks_h +      channel_clicks_h, data = train, in
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 0.23%
## Confusion matrix:
##       1 2 class.error
## 1 79819 0           0
## 2   182 0           1
```

```
# Visualizing the importance of variables
varImpPlot(rf2)
```




```
# Prediction using test data
pred_rf2 = predict(rf2, test[,-5])

# Calculating accuracy for model
accuracy_rf2 = sum(obs_oL == pred_rf2) / length(obs_oL)
accuracy_rf2
```

```
## [1] 0.9977499
```

```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf2)
```

```
##       pred_rf2
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```
# Well, it did not work any better than the previous version. Let's check some
# models using the train_over set


##### TRAIN_OVER #####

# Re-checking correlation of train_over
train_over_cor
```

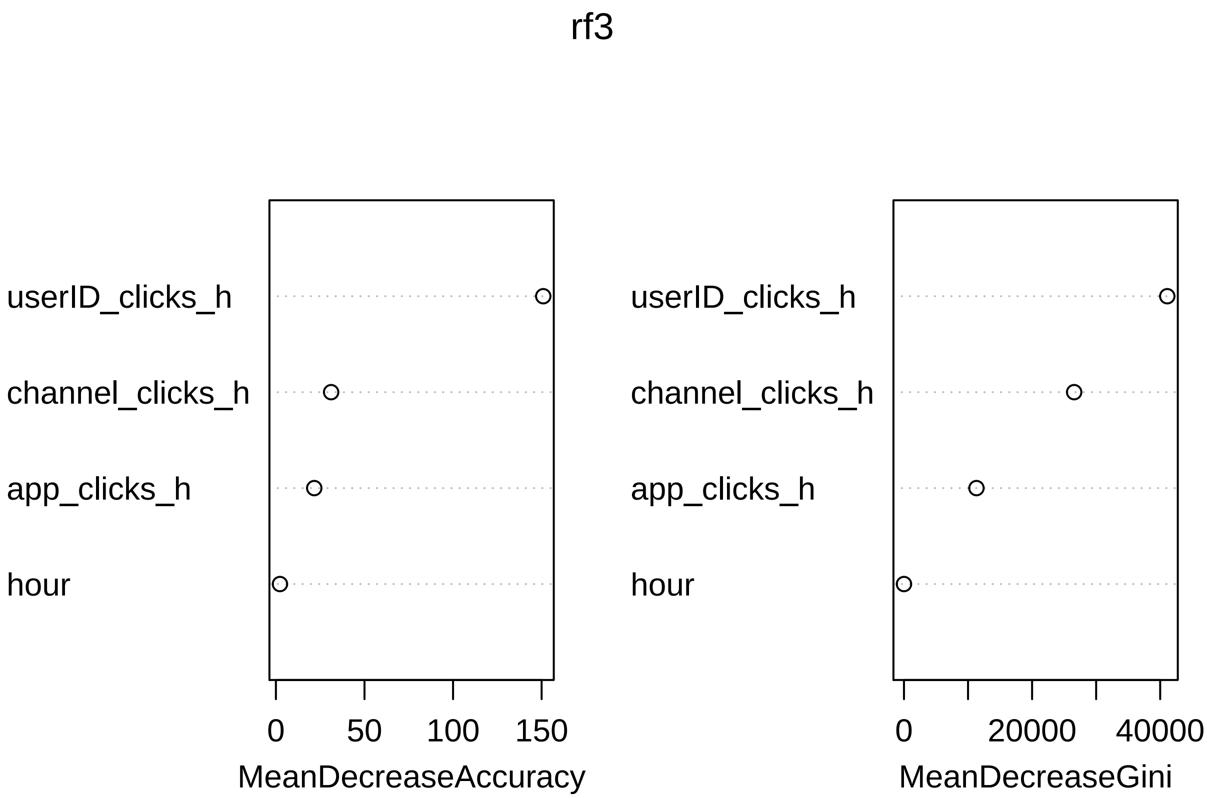
```
##                      is_attributed          day        hour       minute
## is_attributed          1.000000000 -0.0010668713 -0.04273077 -0.024784111
## day                   -0.001066871  1.0000000000 -0.24702904 -0.002953535
## hour                  -0.042730769 -0.2470290386  1.00000000 -0.051439756
## minute                -0.024784111 -0.0029535346 -0.05143976  1.000000000
```

```
## userID_clicks_h       0.997778890  0.0007248818 -0.04669101 -0.025927600
## ip_clicks_h           0.997240310  0.0005273523 -0.04578824 -0.025874319
## app_clicks_h          0.638549480  0.1515330356 -0.01215414  0.008977345
## channel_clicks_h      0.735193970  0.0398602603  0.01952198 -0.015144581
## ip_app_clicks_h       0.997812708  0.0006779099 -0.04669083 -0.025967447
## ip_channel_clicks_h   0.997813356  0.0006747045 -0.04670822 -0.025964033
##                     userID_clicks_h   ip_clicks_h app_clicks_h channel_clicks_h
## is_attributed          0.9977788898  0.9972403105  0.638549480       0.73519397
## day                    0.0007248818  0.0005273523  0.151533036       0.03986026
## hour                  -0.0466910053 -0.0457882369 -0.012154144       0.01952198
## minute                -0.0259275998 -0.0258743194  0.008977345      -0.01514458
## userID_clicks_h        1.0000000000  0.9994525169  0.638529528       0.73426656
## ip_clicks_h            0.9994525169  1.0000000000  0.638464110       0.73399873
## app_clicks_h           0.6385295277  0.6384641099  1.000000000       0.72026972
## channel_clicks_h       0.7342665589  0.7339987303  0.720269721       1.00000000
## ip_app_clicks_h        0.9999114078  0.9994084137  0.638619105       0.73435399
## ip_channel_clicks_h    0.9999116432  0.9994073887  0.638607674       0.73435524
##                     ip_app_clicks_h ip_channel_clicks_h
## is_attributed          0.9978127076        0.9978133555
## day                    0.0006779099        0.0006747045
## hour                  -0.0466908308       -0.0467082210
## minute                -0.0259674474       -0.0259640334
## userID_clicks_h        0.9999114078        0.9999116432
## ip_clicks_h            0.9994084137        0.9994073887
## app_clicks_h           0.6386191047        0.6386076741
## channel_clicks_h       0.7343539878        0.7343552398
## ip_app_clicks_h        1.0000000000        0.9999995971
## ip_channel_clicks_h    0.9999995971        1.0000000000
```

```
# Model creation
rf3 = randomForest(is_attributed ~ hour + userID_clicks_h + app_clicks_h +
                     channel_clicks_h,
                   data = train_over, importance = TRUE)

# Visualizing the results of model
rf3
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + userID_clicks_h +      app_clicks_h + channel_clicks_
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0%
## Confusion matrix:
##       1     2 class.error
## 1 78907     0           0
## 2     0 79093           0
```

```
# Visualizing the importance of variables
varImpPlot(rf3)
```




```r
# Prediction using test data
pred_rf3 = predict(rf3, test[,-5])

# Calculating accuracy of model
accuracy_rf3 = sum(obs_oL == pred_rf3) / length(obs_oL)
accuracy_rf3
```

```
## [1] 0.9977499
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf3)
```

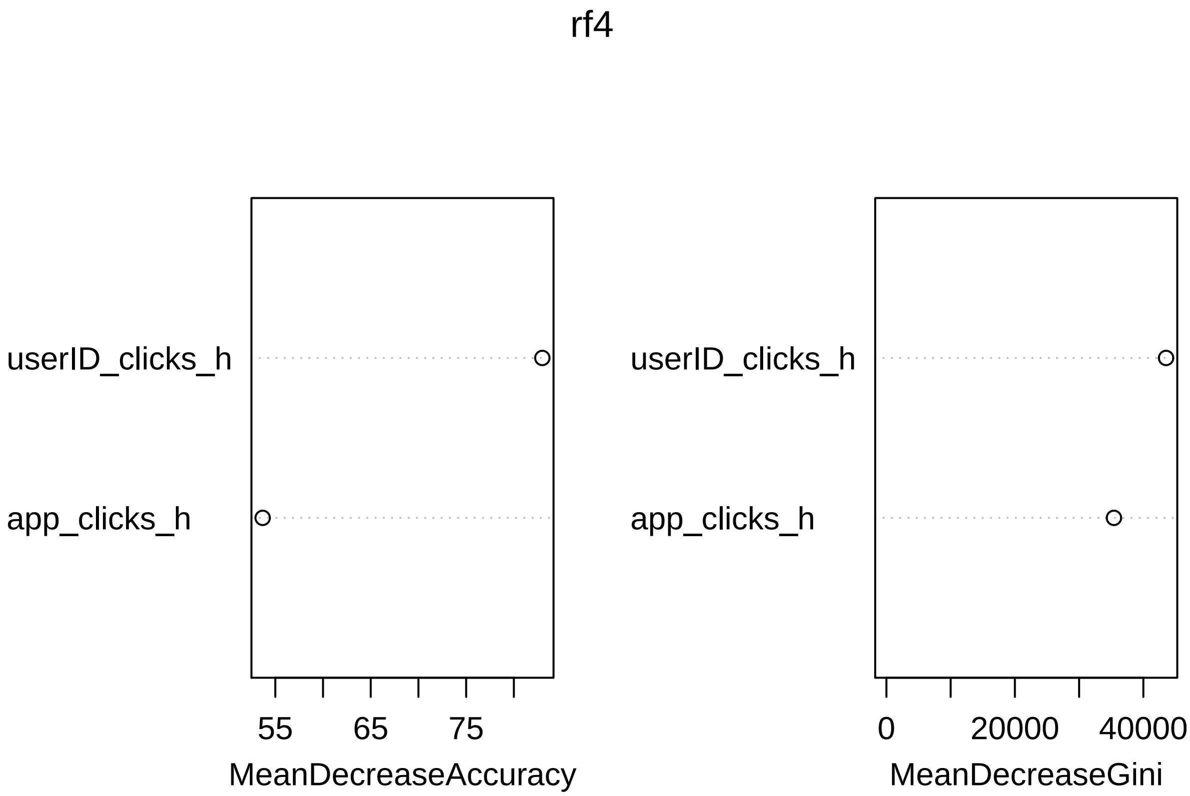
```
##         pred_rf3
## obs_oL      1     2
##       1 19954     0
##       2    45     0
```

```r
# The same problem as observed in rf1, where the model classifies all as the
# class that possess higher count


# Model creation
rf4 = randomForest(is_attributed ~ userID_clicks_h + app_clicks_h,
                   data = train_over, importance = TRUE)

# Visualizing the results of model
rf4
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ userID_clicks_h + app_clicks_h,      data = train_over, impo
```

```
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 0%
## Confusion matrix:
##       1     2 class.error
## 1 78907     0           0
## 2     0 79093           0
```

```
# Visualizing the importance of variables
varImpPlot(rf4)
```




```
# Prediction using test data
pred_rf4 = predict(rf4, test[,-5])

# Calculating accuracy of model
accuracy_rf4 = sum(obs_oL == pred_rf4) / length(obs_oL)
accuracy_rf4
```

```
## [1] 0.9977499
```

```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf4)
```

```
##       pred_rf4
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```r
# The same problem as observed in rf1, where the model classifies all as the
# class that possess higher count

# It seems that we are having some overfitting problem with the over sample train
# set. Let's try the train_under set.


##### TRAIN_UNDER #####

# Re-checking correlation of train_over
train_under_cor
```

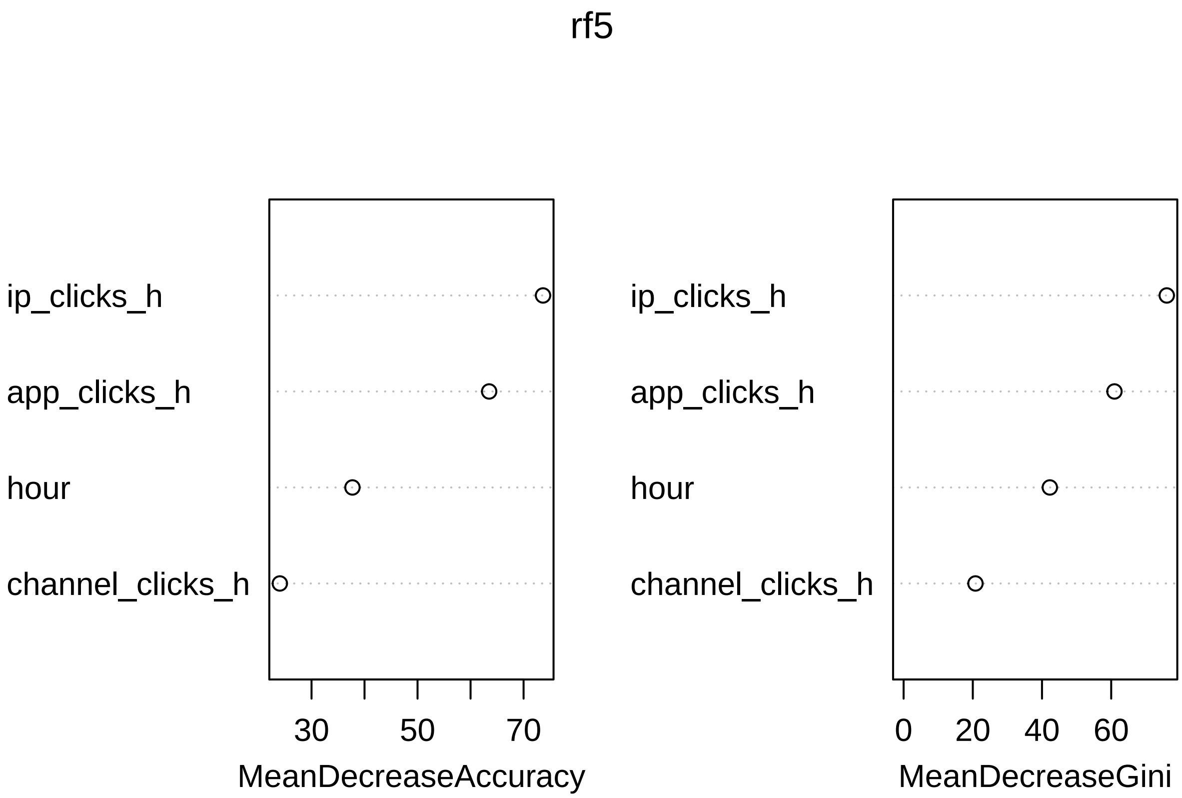
```
##                     is_attributed           day        hour        minute
## is_attributed         1.000000000  0.0044780538 -0.069454476 -0.0172912481
## day                   0.004478054  1.0000000000 -0.290356890  0.0008137752
## hour                 -0.069454476 -0.2903568898  1.000000000  0.0072762025
## minute               -0.017291248  0.0008137752  0.007276203  1.0000000000
## userID_clicks_h      -0.102604101  0.0386031631  0.075198607 -0.0251512316
## ip_clicks_h          -0.218221990  0.0133455138  0.124712294 -0.0257905624
## app_clicks_h         -0.205825266  0.0699078285  0.045332144 -0.0287014242
## channel_clicks_h     -0.097226536  0.0050393793  0.043753893 -0.0035062905
## ip_app_clicks_h      -0.116793528  0.0126393382  0.071946410 -0.0114756994
## ip_channel_clicks_h  -0.072360004 -0.0044014402  0.057014470 -0.0168917464
##                     userID_clicks_h ip_clicks_h app_clicks_h channel_clicks_h
## is_attributed           -0.10260410 -0.21822199  -0.20582527     -0.097226536
## day                      0.03860316  0.01334551   0.06990783      0.005039379
## hour                     0.07519861  0.12471229   0.04533214      0.043753893
## minute                  -0.02515123 -0.02579056  -0.02870142     -0.003506290
## userID_clicks_h          1.00000000  0.46395368   0.11505165      0.078446998
## ip_clicks_h              0.46395368  1.00000000   0.26793589      0.180467118
## app_clicks_h             0.11505165  0.26793589   1.00000000      0.380457318
## channel_clicks_h         0.07844700  0.18046712   0.38045732      1.000000000
## ip_app_clicks_h          0.22309395  0.53388212   0.53731981      0.256678405
## ip_channel_clicks_h      0.17034284  0.36420196   0.25480530      0.437903706
##                     ip_app_clicks_h ip_channel_clicks_h
## is_attributed           -0.11679353         -0.07236000
## day                      0.01263934         -0.00440144
## hour                     0.07194641          0.05701447
## minute                  -0.01147570         -0.01689175
## userID_clicks_h          0.22309395          0.17034284
## ip_clicks_h              0.53388212          0.36420196
## app_clicks_h             0.53731981          0.25480530
## channel_clicks_h         0.25667840          0.43790371
## ip_app_clicks_h          1.00000000          0.51866529
## ip_channel_clicks_h      0.51866529          1.00000000
```

```r
# Model creation
rf5 = randomForest(is_attributed ~ hour + ip_clicks_h +
                     app_clicks_h + channel_clicks_h,
                   data = train_under, importance = TRUE)

# Visualizing the results of model
rf5
```

```
##
```

```
## Call:
##  randomForest(formula = is_attributed ~ hour + ip_clicks_h + app_clicks_h +      channel_clicks_h, da
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 4.73%
## Confusion matrix:
##      1   2 class.error
## 1 2735  60  0.02146691
## 2   81 103  0.44021739
```

```
# Checking variable importance
varImpPlot(rf5)
```




```
# Prediction using test data
pred_rf5 = predict(rf5, test[,-5])

# Calculating accuracy for model
accuracy_rf5 = sum(obs_oL == pred_rf5) / length(obs_oL)
accuracy_rf5
```

```
## [1] 0.979549
```

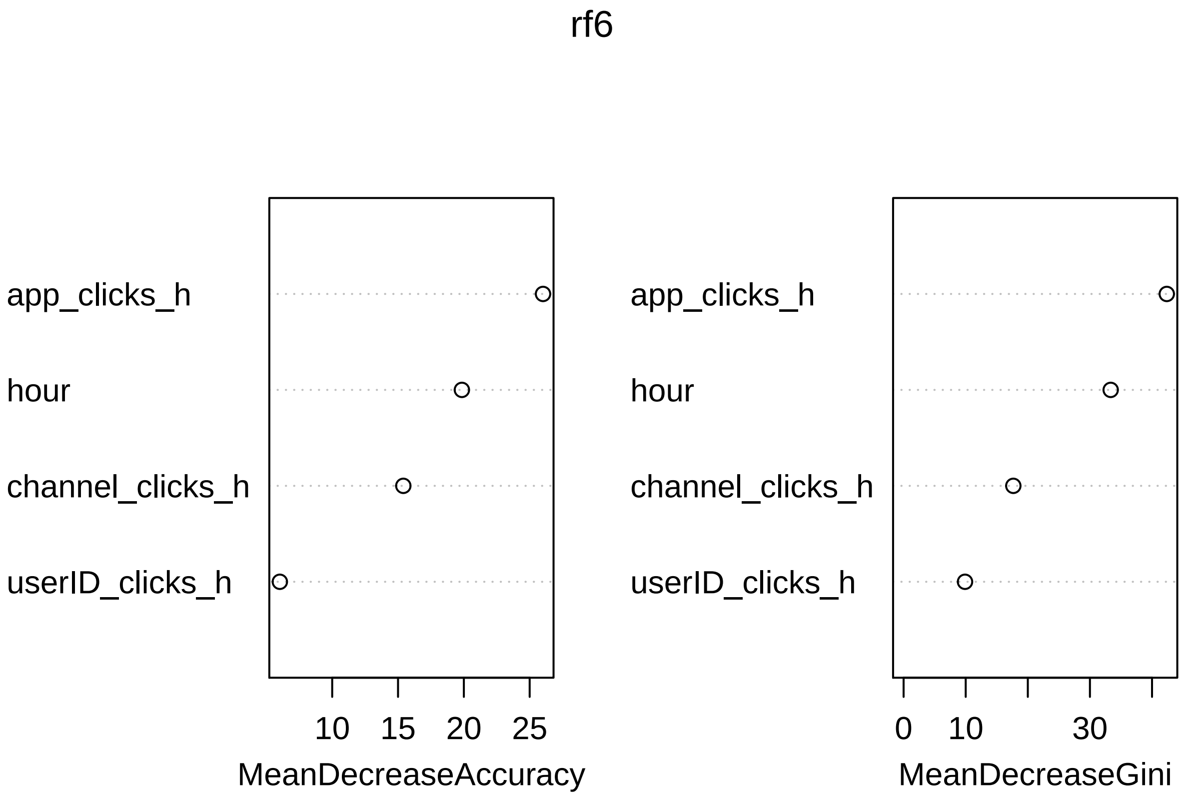
```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf5)
```

```
##       pred_rf5
## obs_oL     1     2
##      1 19571   383
##      2    26    19
```

```r
# Although it still makes some mistakes, this model seems to be doing a
# much better job than the previous versions.

# Let's try another version, using the userID_clicks_h instead of ip_clicks_h


# Model creation
rf6 = randomForest(is_attributed ~ hour + userID_clicks_h +
                     app_clicks_h + channel_clicks_h,
                   data = train_under, importance = TRUE)

# Visualizing the results of model
rf6
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + userID_clicks_h +      app_clicks_h + channel_clicks_
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 6.04%
## Confusion matrix:
##      1 2  class.error
## 1 2794 1 0.0003577818
## 2  179 5 0.9728260870
```

```r
# Checking variable importance
varImpPlot(rf6)
```



```r
# Prediction using test data
pred_rf6 = predict(rf6, test[,-5])

# Calculating accuracy for model
accuracy_rf6 = sum(obs_oL == pred_rf6) / length(obs_oL)
accuracy_rf6
```

```
## [1] 0.9973499
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf6)
```

```
##       pred_rf6
## obs_oL     1     2
##      1 19946     8
##      2    45     0
```

```r
# The version rf5 was better than the version rf6 and all the others tested
# until this point.

# Now, let's try the other train set >> train_underOver


##### TRAIN_UNDEROVER #####

# Re-checking correlation of train_over
train_underOver_cor
```

```
##                     is_attributed          day        hour       minute
## is_attributed         1.000000000  0.002632042 -0.14781983 -0.034606590
## day                   0.002632042  1.000000000 -0.25989884  0.013592740
## hour                 -0.147819832 -0.259898839  1.00000000 -0.057560369
## minute               -0.034606590  0.013592740 -0.05756037  1.000000000
## userID_clicks_h       0.878067770 -0.001147684 -0.11466072 -0.042952841
## ip_clicks_h           0.588828363 -0.036795043 -0.02298851 -0.053201077
## app_clicks_h          0.533962461  0.143051090  0.01409193  0.003138878
## channel_clicks_h      0.721325635  0.046060308 -0.06010483 -0.004475516
## ip_app_clicks_h       0.873364042 -0.002959513 -0.11117945 -0.041742923
## ip_channel_clicks_h   0.887282705  0.001517575 -0.11715395 -0.043015254
##                     userID_clicks_h ip_clicks_h app_clicks_h channel_clicks_h
## is_attributed           0.878067770  0.58882836  0.533962461      0.721325635
## day                    -0.001147684 -0.03679504  0.143051090      0.046060308
## hour                   -0.114660723 -0.02298851  0.014091931     -0.060104827
## minute                 -0.042952841 -0.05320108  0.003138878     -0.004475516
## userID_clicks_h         1.000000000  0.79863688  0.594525623      0.750274069
## ip_clicks_h             0.798636885  1.00000000  0.504833403      0.571713118
## app_clicks_h            0.594525623  0.50483340  1.000000000      0.666986423
## channel_clicks_h        0.750274069  0.57171312  0.666986423      1.000000000
## ip_app_clicks_h         0.985914321  0.79909945  0.609719948      0.750792434
## ip_channel_clicks_h     0.991876642  0.77856700  0.595726090      0.757474357
##                     ip_app_clicks_h ip_channel_clicks_h
## is_attributed           0.873364042         0.887282705
## day                    -0.002959513         0.001517575
## hour                   -0.111179447        -0.117153949
## minute                 -0.041742923        -0.043015254
```

```
## userID_clicks_h         0.985914321         0.991876642
## ip_clicks_h             0.799099448         0.778566996
## app_clicks_h            0.609719948         0.595726090
## channel_clicks_h        0.750792434         0.757474357
## ip_app_clicks_h         1.000000000         0.991747195
## ip_channel_clicks_h     0.991747195         1.000000000
```

```
# Model creation
rf7 = randomForest(is_attributed ~ hour + ip_channel_clicks_h + app_clicks_h,
                   data = train_underOver, importance = TRUE)

# Visualizing the results of model
rf7
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + ip_channel_clicks_h +      app_clicks_h, data = train_
##               Type of random forest: classification
##                     Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 0.27%
## Confusion matrix:
##      1    2 class.error
## 1 2010    0 0.000000000
## 2   11 1979 0.005527638
```

```
# Prediction using test data
pred_rf7 = predict(rf7, test[,-5])

# Calculating accuracy for model
accuracy_rf7 = sum(obs_oL == pred_rf7) / length(obs_oL)
accuracy_rf7
```

```
## [1] 0.9977499
```

```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf7)
```

```
##       pred_rf7
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```
# The same problem presente by rf1, the model classifies all as the class that
# possess higher count

# Let's try removing the predictor app_clicks_h


# Model creation
rf8 = randomForest(is_attributed ~ hour + ip_channel_clicks_h,
                   data = train_underOver, importance = TRUE)

# Visualizing the results of model
rf8
```

```
##
```

```
## Call:
##  randomForest(formula = is_attributed ~ hour + ip_channel_clicks_h,      data = train_underOver, impo
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 0.52%
## Confusion matrix:
##      1    2 class.error
## 1 2005    5 0.002487562
## 2   16 1974 0.008040201
```

```
# Prediction using test data
pred_rf8 = predict(rf8, test[,-5])

# Calculating accuracy for model
accuracy_rf8 = sum(obs_oL == pred_rf8) / length(obs_oL)
accuracy_rf8
```

```
## [1] 0.9977499
```

```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf8)
```

```
##       pred_rf8
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```
# The same problem as before, the model classifies all as the class that
# possess higher count


# Model creation
rf9 = randomForest(is_attributed ~ hour + userID_clicks_h + app_clicks_h,
                   data = train_underOver, importance = TRUE)

# Visualizing the results of model
rf9
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + userID_clicks_h +      app_clicks_h, data = train_unde
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 0.48%
## Confusion matrix:
##      1    2 class.error
## 1 2002    8 0.003980100
## 2   11 1979 0.005527638
```

```
# Prediction using test data
pred_rf9 = predict(rf9, test[,-5])

# Calculating accuracy for model
```

```r
accuracy_rf9 = sum(obs_oL == pred_rf9) / length(obs_oL)
accuracy_rf9
```

```
## [1] 0.9977499
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf9)
```

```
##       pred_rf9
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```r
# The same problem as before, the model classifies all as the class that
# possess higher count


# Model creation
rf10 = randomForest(is_attributed ~ hour + userID_clicks_h,
                    data = train_underOver, importance = TRUE)

# Visualizing the results of model
rf10
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ hour + userID_clicks_h,      data = train_underOver, importa
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 0.73%
## Confusion matrix:
##      1    2 class.error
## 1 1992   18 0.008955224
## 2   11 1979 0.005527638
```

```r
# Prediction using test data
pred_rf10 = predict(rf10, test[,-5])

# Calculating accuracy for model
accuracy_rf10 = sum(obs_oL == pred_rf10) / length(obs_oL)
accuracy_rf10
```

```
## [1] 0.9977499
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_rf10)
```

```
##       pred_rf10
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```r
# The same problem as before, the model classifies all as the class that
# possess higher count

# So, the undercsampling technique seems to have worked better, meaning that the
```

```r
# selected sample was able to produce a more accurate model, at least for the
# random forest algorithm.

# Let's try some versions using the Naive Bayes algorithm


###### NAIVE BAYES ######

# For the Naive Bayes version, we can also try to include the categorical
# variables for testing

##### TRAIN (non-balanced train set) #####

# Model creation
nb1 = naiveBayes(is_attributed ~ app + device + os + channel + hour +
                   userID_clicks_h + app_clicks_h + channel_clicks_h +
                   ip_app_clicks_h + ip_channel_clicks_h,
                 data = train, importance = TRUE)

# Visualizing the results of model
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace, importance = TRUE)
##
## A-priori probabilities:
## Y
##           1           2
## 0.997725028 0.002274972
##
## Conditional probabilities:
##    app
## Y              1            2            3            4            5
##   1 3.142109e-02 1.189065e-01 1.819867e-01 5.387189e-04 1.616157e-03
##   2 0.000000e+00 0.000000e+00 2.197802e-02 0.000000e+00 4.945055e-02
##    app
## Y              6            7            8            9           10
##   1 1.270374e-02 9.872336e-03 1.994513e-02 9.083050e-02 3.670805e-03
##   2 0.000000e+00 0.000000e+00 1.648352e-02 3.296703e-02 7.142857e-02
##    app
## Y             11           12           13           14           15
##   1 1.899297e-02 1.324497e-01 2.406695e-02 5.407234e-02 8.558113e-02
##   2 1.098901e-02 5.494505e-03 0.000000e+00 0.000000e+00 1.098901e-02
##    app
## Y             16           17           18           19           20
##   1 2.505669e-05 3.846202e-03 8.429071e-02 4.171939e-03 8.682143e-03
##   2 0.000000e+00 0.000000e+00 2.747253e-02 3.186813e-01 5.494505e-03
##    app
## Y             21           22           23           24           25
##   1 1.978226e-02 3.708390e-03 1.450782e-02 7.178742e-03 7.955499e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

```
##    app
## Y             26           27           28           29           30
##   1 1.639960e-02 6.752778e-03 7.065987e-03 3.307483e-03 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 9.890110e-02 0.000000e+00
##    app
## Y             31           32           33           34           35
##   1 0.000000e+00 2.956690e-03 1.127551e-04 1.252835e-05 2.129819e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02 1.098901e-01
##    app
## Y             36           37           38           39           42
##   1 1.127551e-03 4.009071e-04 1.628685e-04 1.503401e-04 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00
##    app
## Y             43           44           45           46           47
##   1 8.769842e-05 1.252835e-04 2.505669e-04 1.503401e-04 4.886055e-04
##   2 0.000000e+00 0.000000e+00 5.494505e-02 0.000000e+00 0.000000e+00
##    app
## Y             48           49           50           52           53
##   1 2.505669e-05 2.505669e-05 5.011338e-05 2.505669e-05 3.758504e-05
##   2 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    app
## Y             54           55           56           58           59
##   1 1.252835e-05 5.011338e-04 5.011338e-04 4.510204e-04 5.011338e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00
##    app
## Y             60           61           62           64           65
##   1 1.753968e-04 2.505669e-05 3.382653e-04 1.099989e-02 1.879252e-04
##   2 5.494505e-03 0.000000e+00 1.098901e-02 0.000000e+00 0.000000e+00
##    app
## Y             66           67           68           70           71
##   1 1.127551e-04 6.264173e-05 1.127551e-04 2.505669e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             72           74           75           76           78
##   1 7.517007e-05 5.011338e-05 2.505669e-05 7.517007e-05 0.000000e+00
##   2 2.747253e-02 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             79           80           81           82           83
##   1 1.252835e-05 0.000000e+00 0.000000e+00 2.129819e-04 1.378118e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02
##    app
## Y             84           85           86           87           88
##   1 5.011338e-05 2.505669e-05 3.758504e-05 1.252835e-05 6.264173e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             91           92           93           94           95
##   1 1.252835e-05 1.252835e-05 3.758504e-04 3.758504e-04 3.758504e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             96           97           99          100          101
##   1 0.000000e+00 0.000000e+00 1.252835e-05 1.252835e-05 3.758504e-05
##   2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            103          104          105          107          108
```

```
##   1 5.011338e-05 1.252835e-05 1.252835e-05 1.503401e-04 1.252835e-05
##   2 1.098901e-02 0.000000e+00 0.000000e+00 1.098901e-02 1.098901e-02
##    app
## Y            109          110          112          115          116
##   1 1.002268e-04 1.378118e-04 1.252835e-05 0.000000e+00 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03
##    app
## Y            117          118          119          121          122
##   1 3.758504e-05 2.505669e-05 3.758504e-05 3.758504e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            123          124          125          133          134
##   1 1.252835e-05 1.252835e-05 7.517007e-05 1.252835e-05 3.758504e-05
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    app
## Y            137          139          145          146          148
##   1 2.505669e-05 2.505669e-05 0.000000e+00 3.758504e-05 2.505669e-05
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    app
## Y            149          150          151          158          160
##   1 2.505669e-05 7.391724e-04 1.089966e-03 2.505669e-05 1.378118e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            161          163          165          168          170
##   1 2.505669e-05 1.252835e-05 1.252835e-05 1.252835e-05 8.769842e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            171          176          181          183          190
##   1 1.252835e-05 2.505669e-05 2.505669e-05 2.004535e-04 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            192          202          204          208          215
##   1 1.252835e-05 3.758504e-05 2.505669e-05 1.378118e-04 5.011338e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    app
## Y            216          232          233          261          266
##   1 1.252835e-05 8.769842e-05 0.000000e+00 0.000000e+00 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00
##    app
## Y            267          268          271          273          293
##   1 1.252835e-05 1.252835e-05 1.252835e-05 2.505669e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            302          310          315          347          363
##   1 1.252835e-05 2.505669e-05 3.758504e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            372          394          398          407          425
##   1 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            474          486          536          538          548
##   1 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

```
##    app
## Y            551
##   1 1.252835e-05
##   2 0.000000e+00
##
##    device
## Y              0            1            2            4            5
##   1 4.961225e-03 9.434721e-01 4.399955e-02 2.505669e-05 8.769842e-05
##   2 2.197802e-01 6.373626e-01 5.494505e-03 5.494505e-03 0.000000e+00
##    device
## Y              6            7            9           11           16
##   1 7.517007e-05 2.505669e-05 1.252835e-05 1.252835e-05 5.011338e-05
##   2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 1.648352e-02
##    device
## Y             17           18           20           21           25
##   1 1.252835e-05 1.252835e-05 2.505669e-05 1.252835e-05 3.758504e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00
##    device
## Y             30           33           36           37           40
##   1 1.252835e-05 2.505669e-05 0.000000e+00 1.252835e-05 5.011338e-05
##   2 5.494505e-03 5.494505e-03 5.494505e-03 0.000000e+00 5.494505e-03
##    device
## Y             49           50           53           56           58
##   1 1.252835e-05 2.505669e-05 0.000000e+00 1.252835e-05 0.000000e+00
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    device
## Y             59           60           67           74           76
##   1 1.503401e-04 2.505669e-05 2.505669e-05 0.000000e+00 1.252835e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    device
## Y             78           79           97          100          102
##   1 1.252835e-05 1.252835e-05 2.505669e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00 0.000000e+00
##    device
## Y            103          106          109          114          116
##   1 0.000000e+00 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03
##    device
## Y            124          129          154          160          163
##   1 2.505669e-05 0.000000e+00 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y            167          180          182          188          196
##   1 1.252835e-05 0.000000e+00 2.505669e-05 0.000000e+00 0.000000e+00
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    device
## Y            202          203          210          211          220
##   1 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y            241          268          291          321          329
##   1 0.000000e+00 1.252835e-05 0.000000e+00 0.000000e+00 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
```

```
## Y           347          351          362          374          385
##   1 1.252835e-05 0.000000e+00 0.000000e+00 1.252835e-05 1.252835e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           386          414          420          486          516
##   1 0.000000e+00 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           549          552          558          579          581
##   1 1.252835e-05 0.000000e+00 2.505669e-05 0.000000e+00 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03
##    device
## Y           596          607          657          828          883
##   1 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           928          957         1042         1080         1162
##   1 1.252835e-05 0.000000e+00 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y          1318         1422         1482         1728         1839
##   1 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y          2120         2429         2980         3032         3282
##   1 1.252835e-05 1.252835e-05 1.252835e-05 3.745975e-03 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y          3331         3543         3545         3866         3867
##   1 0.000000e+00 1.490873e-03 1.252835e-05 9.521542e-04 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##
##    os
## Y             0            1            2            3            4
##   1 1.753968e-03 1.171400e-02 3.846202e-03 1.569802e-02 2.906576e-03
##   2 1.208791e-01 0.000000e+00 5.494505e-03 5.494505e-03 1.098901e-02
##    os
## Y             5            6            7            8            9
##   1 9.646826e-04 2.554530e-02 9.897393e-04 2.701111e-02 2.347812e-02
##   2 0.000000e+00 1.098901e-02 2.747253e-02 0.000000e+00 0.000000e+00
##    os
## Y            10           11           12           13           14
##   1 2.811361e-02 7.667347e-03 1.126298e-02 2.120548e-01 1.286661e-02
##   2 1.648352e-02 0.000000e+00 0.000000e+00 1.153846e-01 1.098901e-02
##    os
## Y            15           16           17           18           19
##   1 2.448039e-02 1.725153e-02 5.249377e-02 4.883549e-02 2.385773e-01
##   2 2.197802e-02 1.098901e-02 2.747253e-02 1.098901e-02 1.813187e-01
##    os
## Y            20           21           22           23           24
##   1 2.335284e-02 5.261905e-04 4.037886e-02 1.028577e-02 1.791553e-03
##   2 1.098901e-02 3.296703e-02 2.747253e-02 5.494505e-03 1.098901e-01
##    os
## Y            25           26           27           28           29
```

```
##    1 2.263872e-02 4.735715e-03 1.008532e-02 7.078515e-03 4.510204e-04
##    2 1.648352e-02 1.098901e-02 3.296703e-02 0.000000e+00 6.043956e-02
##     os
## Y             30           31           32           34           35
##    1 6.602438e-03 4.234581e-03 9.195806e-03 1.478345e-03 9.170749e-03
##    2 1.098901e-02 5.494505e-03 1.098901e-02 0.000000e+00 5.494505e-03
##     os
## Y             36           37           38           39           40
##    1 4.009071e-03 1.598617e-02 8.268708e-04 4.384921e-04 3.733447e-03
##    2 1.098901e-02 5.494505e-03 1.648352e-02 0.000000e+00 0.000000e+00
##     os
## Y             41           42           43           44           45
##    1 1.365590e-02 2.017064e-03 1.804082e-03 7.767574e-04 8.769842e-05
##    2 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00 0.000000e+00
##     os
## Y             46           47           48           49           50
##    1 1.603628e-03 7.892858e-03 1.152608e-03 3.232313e-03 3.132086e-04
##    2 0.000000e+00 1.648352e-02 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             52           53           55           56           57
##    1 3.633220e-04 7.780102e-03 5.261905e-04 6.013606e-04 1.753968e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             58           59           60           61           62
##    1 1.566043e-03 1.002268e-04 3.758504e-05 0.000000e+00 1.628685e-04
##    2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##     os
## Y             63           64           65           66           67
##    1 2.129819e-04 3.257370e-04 5.763039e-04 3.883787e-04 2.505669e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             68           69           70           71           73
##    1 1.252835e-05 3.758504e-05 4.259637e-04 2.505669e-05 2.756236e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             74           76           77           78           79
##    1 2.505669e-05 1.503401e-04 2.756236e-04 0.000000e+00 4.510204e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             80           81           83           84           85
##    1 3.758504e-05 3.758504e-05 7.517007e-05 1.252835e-05 5.011338e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             87           88           90           92           96
##    1 3.758504e-05 1.252835e-05 1.252835e-04 5.011338e-05 1.002268e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y             97           98           99          100          102
##    1 2.756236e-04 1.002268e-04 1.252835e-05 1.127551e-04 5.011338e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     os
## Y            106          107          108          109          110
##    1 1.252835e-05 2.505669e-05 1.252835e-05 3.758504e-05 3.758504e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

```
##    os
## Y            111          112          113          114          116
##   1 2.505669e-05 5.011338e-05 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            117          118          127          129          132
##   1 2.505669e-05 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            133          135          137          138          142
##   1 1.252835e-05 1.252835e-05 0.000000e+00 2.505669e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            151          152          153          155          168
##   1 1.252835e-05 2.505669e-05 1.252835e-05 3.758504e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            172          174          178          184          185
##   1 0.000000e+00 1.252835e-05 5.011338e-05 2.505669e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            192          193          196          198          199
##   1 1.252835e-05 1.252835e-05 2.505669e-05 3.758504e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            207          607          748          836          866
##   1 1.252835e-05 3.871259e-03 1.653742e-03 1.252835e-05 1.140079e-03
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03
##
##    channel
## Y              3            4            5           13           15
##   1 4.936168e-03 1.252835e-05 1.252835e-05 6.514740e-04 2.505669e-05
##   2 0.000000e+00 0.000000e+00 2.197802e-02 0.000000e+00 0.000000e+00
##    channel
## Y             17           18           19           21           22
##   1 1.227778e-03 2.255102e-04 6.652551e-03 1.628685e-03 7.517007e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 7.692308e-02 0.000000e+00
##    channel
## Y             24           30          101          105          107
##   1 3.006803e-04 1.002268e-03 1.175159e-02 6.326814e-03 4.597903e-02
##   2 0.000000e+00 0.000000e+00 4.945055e-02 0.000000e+00 5.494505e-03
##    channel
## Y            108          110          111          113          114
##   1 6.264173e-05 1.578572e-03 3.257370e-03 2.280159e-03 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.153846e-01 5.494505e-03
##    channel
## Y            115          116          118          120          121
##   1 7.654819e-03 2.944161e-03 1.240306e-03 8.143424e-04 2.485624e-02
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03
##    channel
## Y            122          123          124          125          126
##   1 1.385635e-02 8.519275e-04 2.367857e-03 6.289229e-03 1.378118e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
```

```
## Y            127          128          130          134          135
##   1 4.898583e-03 1.477092e-02 7.792631e-03 3.287438e-02 1.454541e-02
##   2 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03 0.000000e+00
##    channel
## Y            137          138          140          145          150
##   1 1.173906e-02 3.257370e-04 1.331763e-02 1.963192e-02 9.521542e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.648352e-02 0.000000e+00
##    channel
## Y            153          160          171          173          174
##   1 2.932886e-02 2.756236e-04 1.879252e-04 7.604706e-03 1.002268e-04
##   2 0.000000e+00 0.000000e+00 2.197802e-02 5.494505e-03 0.000000e+00
##    channel
## Y            178          182          203          205          208
##   1 2.941655e-02 7.767574e-04 6.264173e-05 2.394167e-02 5.011338e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            210          211          212          213          215
##   1 3.382653e-04 5.700397e-03 6.414513e-03 3.520465e-03 7.968028e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 3.296703e-01 0.000000e+00
##    channel
## Y            219          224          232          234          236
##   1 1.289167e-02 2.004535e-04 1.227778e-02 3.658277e-03 7.993084e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            237          242          243          244          245
##   1 1.430737e-02 1.252835e-03 1.916837e-03 6.665080e-03 4.740726e-02
##   2 0.000000e+00 0.000000e+00 2.747253e-02 0.000000e+00 0.000000e+00
##    channel
## Y            253          258          259          261          262
##   1 2.255102e-04 5.011338e-03 3.079467e-02 1.252835e-05 2.004535e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            265          266          268          272          274
##   1 3.039377e-02 5.211792e-03 1.127551e-04 3.758504e-05 5.011338e-05
##   2 1.098901e-02 0.000000e+00 0.000000e+00 0.000000e+00 4.945055e-02
##    channel
## Y            277          278          280          282          315
##   1 7.391724e-04 3.332540e-03 8.110851e-02 1.378118e-04 6.627495e-03
##   2 0.000000e+00 0.000000e+00 1.098901e-02 2.747253e-02 0.000000e+00
##    channel
## Y            317          319          320          322          325
##   1 3.457823e-03 3.984014e-03 1.127551e-04 2.756236e-04 1.753968e-03
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    channel
## Y            326          328          330          332          333
##   1 2.643481e-03 1.032336e-02 4.134354e-04 3.758504e-05 1.503401e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.648352e-02
##    channel
## Y            334          340          341          343          347
##   1 1.091219e-02 3.883787e-03 5.011338e-05 2.142347e-03 4.986281e-03
##   2 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03 6.043956e-02
##    channel
## Y            349          353          356          360          361
##   1 5.976021e-03 2.505669e-05 1.503401e-04 7.015873e-04 3.758504e-04
```

```
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            364          371          373          376          377
##    1 3.382653e-03 3.345068e-03 5.261905e-04 5.700397e-03 8.381463e-03
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02
##     channel
## Y            379          386          391          400          401
##    1 1.839161e-02 4.710658e-03 3.382653e-04 4.209524e-03 7.842744e-03
##    2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            402          404          406          409          410
##    1 1.515930e-03 8.769842e-05 7.517007e-04 1.086208e-02 2.505669e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            411          412          416          417          419
##    1 2.004535e-04 3.194728e-03 2.380386e-04 2.067177e-03 1.252835e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 2.197802e-02
##     channel
## Y            420          421          424          430          435
##    1 5.011338e-05 4.635488e-04 8.581917e-03 3.081973e-03 1.247823e-02
##    2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            439          442          445          446          448
##    1 1.552262e-02 1.901803e-02 4.923640e-03 4.886055e-04 2.004535e-04
##    2 5.494505e-03 5.494505e-03 1.098901e-02 0.000000e+00 0.000000e+00
##     channel
## Y            449          450          451          452          453
##    1 1.215250e-03 1.628685e-04 1.252835e-05 9.308561e-03 2.630953e-04
##    2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            455          456          457          459          460
##    1 2.505669e-05 5.011338e-05 1.378118e-04 1.931871e-02 2.505669e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            463          465          466          467          469
##    1 5.061452e-03 0.000000e+00 1.497137e-02 1.641213e-03 1.464564e-02
##    2 0.000000e+00 5.494505e-03 1.098901e-02 0.000000e+00 0.000000e+00
##     channel
## Y            474          477          478          479          480
##    1 1.252835e-05 3.956451e-02 2.706123e-03 2.129819e-04 1.464564e-02
##    2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##     channel
## Y            481          483          484          486          487
##    1 4.021599e-03 2.505669e-05 4.886055e-04 8.769842e-05 1.841667e-03
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02
##     channel
## Y            488          489          490          496          497
##    1 5.011338e-05 1.429484e-02 2.756236e-04 7.642291e-04 2.430499e-03
##    2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##     channel
## Y            498
##    1 1.252835e-05
##    2 0.000000e+00
##
```

```
##    hour
## Y       [,1]     [,2]
##   1 9.317042 6.176409
##   2 8.774725 6.064007
##
##    userID_clicks_h
## Y       [,1]      [,2]
##   1 1.043123 0.2878074
##   2 1.021978 0.2340148
##
##    app_clicks_h
## Y        [,1]     [,2]
##   1 131.21702 95.64334
##   2  20.54945 51.60983
##
##    channel_clicks_h
## Y        [,1]     [,2]
##   1 39.219534 43.99727
##   2  9.818681 20.96393
##
##    ip_app_clicks_h
## Y       [,1]      [,2]
##   1 1.044601 0.2842518
##   2 1.010989 0.1045385
##
##    ip_channel_clicks_h
## Y       [,1]      [,2]
##   1 1.018993 0.1598381
##   2 1.010989 0.1045385
```

```r
# Prediction using test data
pred_nb1 = predict(nb1, test[,-5])

# Calculating accuracy for model
accuracy_nb1 = sum(obs_oL == pred_nb1) / length(obs_oL)
accuracy_nb1
```

```
## [1] 0.980549
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_nb1)
```

```
##       pred_nb1
## obs_oL     1     2
##      1 19572   382
##      2     7    38
```

```r
# This model looks much better than the previous ones and it seems that we
# have improved compared to the best version of random forest (rf5)

# Let's try a second version as well but let's keep this model to be further
# adjusted later on.


# Model creation
nb2 = naiveBayes(is_attributed ~ app + device + os + channel + hour +
```

```r
                   app_clicks_h + channel_clicks_h,
                 data = train, importance = TRUE)

# Visualizing the results of model
nb2
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace, importance = TRUE)
##
## A-priori probabilities:
## Y
##           1           2
## 0.997725028 0.002274972
##
## Conditional probabilities:
##    app
## Y              1            2            3            4            5
##   1 3.142109e-02 1.189065e-01 1.819867e-01 5.387189e-04 1.616157e-03
##   2 0.000000e+00 0.000000e+00 2.197802e-02 0.000000e+00 4.945055e-02
##    app
## Y              6            7            8            9           10
##   1 1.270374e-02 9.872336e-03 1.994513e-02 9.083050e-02 3.670805e-03
##   2 0.000000e+00 0.000000e+00 1.648352e-02 3.296703e-02 7.142857e-02
##    app
## Y             11           12           13           14           15
##   1 1.899297e-02 1.324497e-01 2.406695e-02 5.407234e-02 8.558113e-02
##   2 1.098901e-02 5.494505e-03 0.000000e+00 0.000000e+00 1.098901e-02
##    app
## Y             16           17           18           19           20
##   1 2.505669e-05 3.846202e-03 8.429071e-02 4.171939e-03 8.682143e-03
##   2 0.000000e+00 0.000000e+00 2.747253e-02 3.186813e-01 5.494505e-03
##    app
## Y             21           22           23           24           25
##   1 1.978226e-02 3.708390e-03 1.450782e-02 7.178742e-03 7.955499e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             26           27           28           29           30
##   1 1.639960e-02 6.752778e-03 7.065987e-03 3.307483e-03 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 9.890110e-02 0.000000e+00
##    app
## Y             31           32           33           34           35
##   1 0.000000e+00 2.956690e-03 1.127551e-04 1.252835e-05 2.129819e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02 1.098901e-01
##    app
## Y             36           37           38           39           42
##   1 1.127551e-03 4.009071e-04 1.628685e-04 1.503401e-04 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00
##    app
## Y             43           44           45           46           47
##   1 8.769842e-05 1.252835e-04 2.505669e-04 1.503401e-04 4.886055e-04
##   2 0.000000e+00 0.000000e+00 5.494505e-02 0.000000e+00 0.000000e+00
```

```
##    app
## Y             48           49           50           52           53
##   1 2.505669e-05 2.505669e-05 5.011338e-05 2.505669e-05 3.758504e-05
##   2 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    app
## Y             54           55           56           58           59
##   1 1.252835e-05 5.011338e-04 5.011338e-04 4.510204e-04 5.011338e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00
##    app
## Y             60           61           62           64           65
##   1 1.753968e-04 2.505669e-05 3.382653e-04 1.099989e-02 1.879252e-04
##   2 5.494505e-03 0.000000e+00 1.098901e-02 0.000000e+00 0.000000e+00
##    app
## Y             66           67           68           70           71
##   1 1.127551e-04 6.264173e-05 1.127551e-04 2.505669e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             72           74           75           76           78
##   1 7.517007e-05 5.011338e-05 2.505669e-05 7.517007e-05 0.000000e+00
##   2 2.747253e-02 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             79           80           81           82           83
##   1 1.252835e-05 0.000000e+00 0.000000e+00 2.129819e-04 1.378118e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02
##    app
## Y             84           85           86           87           88
##   1 5.011338e-05 2.505669e-05 3.758504e-05 1.252835e-05 6.264173e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             91           92           93           94           95
##   1 1.252835e-05 1.252835e-05 3.758504e-04 3.758504e-04 3.758504e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y             96           97           99          100          101
##   1 0.000000e+00 0.000000e+00 1.252835e-05 1.252835e-05 3.758504e-05
##   2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            103          104          105          107          108
##   1 5.011338e-05 1.252835e-05 1.252835e-05 1.503401e-04 1.252835e-05
##   2 1.098901e-02 0.000000e+00 0.000000e+00 1.098901e-02 1.098901e-02
##    app
## Y            109          110          112          115          116
##   1 1.002268e-04 1.378118e-04 1.252835e-05 0.000000e+00 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03
##    app
## Y            117          118          119          121          122
##   1 3.758504e-05 2.505669e-05 3.758504e-05 3.758504e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y            123          124          125          133          134
##   1 1.252835e-05 1.252835e-05 7.517007e-05 1.252835e-05 3.758504e-05
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    app
## Y            137          139          145          146          148
```

```
##   1 2.505669e-05 2.505669e-05 0.000000e+00 3.758504e-05 2.505669e-05
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    app
## Y           149          150          151          158          160
##   1 2.505669e-05 7.391724e-04 1.089966e-03 2.505669e-05 1.378118e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           161          163          165          168          170
##   1 2.505669e-05 1.252835e-05 1.252835e-05 1.252835e-05 8.769842e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           171          176          181          183          190
##   1 1.252835e-05 2.505669e-05 2.505669e-05 2.004535e-04 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           192          202          204          208          215
##   1 1.252835e-05 3.758504e-05 2.505669e-05 1.378118e-04 5.011338e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    app
## Y           216          232          233          261          266
##   1 1.252835e-05 8.769842e-05 0.000000e+00 0.000000e+00 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00
##    app
## Y           267          268          271          273          293
##   1 1.252835e-05 1.252835e-05 1.252835e-05 2.505669e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           302          310          315          347          363
##   1 1.252835e-05 2.505669e-05 3.758504e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           372          394          398          407          425
##   1 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           474          486          536          538          548
##   1 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    app
## Y           551
##   1 1.252835e-05
##   2 0.000000e+00
##
##    device
## Y             0            1            2            4            5
##   1 4.961225e-03 9.434721e-01 4.399955e-02 2.505669e-05 8.769842e-05
##   2 2.197802e-01 6.373626e-01 5.494505e-03 5.494505e-03 0.000000e+00
##    device
## Y             6            7            9           11           16
##   1 7.517007e-05 2.505669e-05 1.252835e-05 1.252835e-05 5.011338e-05
##   2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 1.648352e-02
##    device
## Y            17           18           20           21           25
##   1 1.252835e-05 1.252835e-05 2.505669e-05 1.252835e-05 3.758504e-05
```

```
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00
##    device
## Y            30           33           36           37           40
##   1 1.252835e-05 2.505669e-05 0.000000e+00 1.252835e-05 5.011338e-05
##   2 5.494505e-03 5.494505e-03 5.494505e-03 0.000000e+00 5.494505e-03
##    device
## Y            49           50           53           56           58
##   1 1.252835e-05 2.505669e-05 0.000000e+00 1.252835e-05 0.000000e+00
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    device
## Y            59           60           67           74           76
##   1 1.503401e-04 2.505669e-05 2.505669e-05 0.000000e+00 1.252835e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    device
## Y            78           79           97          100          102
##   1 1.252835e-05 1.252835e-05 2.505669e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00 0.000000e+00
##    device
## Y           103          106          109          114          116
##   1 0.000000e+00 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03
##    device
## Y           124          129          154          160          163
##   1 2.505669e-05 0.000000e+00 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           167          180          182          188          196
##   1 1.252835e-05 0.000000e+00 2.505669e-05 0.000000e+00 0.000000e+00
##   2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    device
## Y           202          203          210          211          220
##   1 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           241          268          291          321          329
##   1 0.000000e+00 1.252835e-05 0.000000e+00 0.000000e+00 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           347          351          362          374          385
##   1 1.252835e-05 0.000000e+00 0.000000e+00 1.252835e-05 1.252835e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           386          414          420          486          516
##   1 0.000000e+00 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y           549          552          558          579          581
##   1 1.252835e-05 0.000000e+00 2.505669e-05 0.000000e+00 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03
##    device
## Y           596          607          657          828          883
##   1 1.252835e-05 1.252835e-05 1.252835e-05 0.000000e+00 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
```

```
## Y           928          957         1042         1080         1162
##   1 1.252835e-05 0.000000e+00 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y          1318         1422         1482         1728         1839
##   1 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y          2120         2429         2980         3032         3282
##   1 1.252835e-05 1.252835e-05 1.252835e-05 3.745975e-03 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    device
## Y          3331         3543         3545         3866         3867
##   1 0.000000e+00 1.490873e-03 1.252835e-05 9.521542e-04 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##
##    os
## Y             0            1            2            3            4
##   1 1.753968e-03 1.171400e-02 3.846202e-03 1.569802e-02 2.906576e-03
##   2 1.208791e-01 0.000000e+00 5.494505e-03 5.494505e-03 1.098901e-02
##    os
## Y             5            6            7            8            9
##   1 9.646826e-04 2.554530e-02 9.897393e-04 2.701111e-02 2.347812e-02
##   2 0.000000e+00 1.098901e-02 2.747253e-02 0.000000e+00 0.000000e+00
##    os
## Y            10           11           12           13           14
##   1 2.811361e-02 7.667347e-03 1.126298e-02 2.120548e-01 1.286661e-02
##   2 1.648352e-02 0.000000e+00 0.000000e+00 1.153846e-01 1.098901e-02
##    os
## Y            15           16           17           18           19
##   1 2.448039e-02 1.725153e-02 5.249377e-02 4.883549e-02 2.385773e-01
##   2 2.197802e-02 1.098901e-02 2.747253e-02 1.098901e-02 1.813187e-01
##    os
## Y            20           21           22           23           24
##   1 2.335284e-02 5.261905e-04 4.037886e-02 1.028577e-02 1.791553e-03
##   2 1.098901e-02 3.296703e-02 2.747253e-02 5.494505e-03 1.098901e-01
##    os
## Y            25           26           27           28           29
##   1 2.263872e-02 4.735715e-03 1.008532e-02 7.078515e-03 4.510204e-04
##   2 1.648352e-02 1.098901e-02 3.296703e-02 0.000000e+00 6.043956e-02
##    os
## Y            30           31           32           34           35
##   1 6.602438e-03 4.234581e-03 9.195806e-03 1.478345e-03 9.170749e-03
##   2 1.098901e-02 5.494505e-03 1.098901e-02 0.000000e+00 5.494505e-03
##    os
## Y            36           37           38           39           40
##   1 4.009071e-03 1.598617e-02 8.268708e-04 4.384921e-04 3.733447e-03
##   2 1.098901e-02 5.494505e-03 1.648352e-02 0.000000e+00 0.000000e+00
##    os
## Y            41           42           43           44           45
##   1 1.365590e-02 2.017064e-03 1.804082e-03 7.767574e-04 8.769842e-05
##   2 0.000000e+00 0.000000e+00 1.098901e-02 0.000000e+00 0.000000e+00
##    os
## Y            46           47           48           49           50
```

```
##    1 1.603628e-03 7.892858e-03 1.152608e-03 3.232313e-03 3.132086e-04
##    2 0.000000e+00 1.648352e-02 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             52           53           55           56           57
##    1 3.633220e-04 7.780102e-03 5.261905e-04 6.013606e-04 1.753968e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             58           59           60           61           62
##    1 1.566043e-03 1.002268e-04 3.758504e-05 0.000000e+00 1.628685e-04
##    2 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03 0.000000e+00
##    os
## Y             63           64           65           66           67
##    1 2.129819e-04 3.257370e-04 5.763039e-04 3.883787e-04 2.505669e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             68           69           70           71           73
##    1 1.252835e-05 3.758504e-05 4.259637e-04 2.505669e-05 2.756236e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             74           76           77           78           79
##    1 2.505669e-05 1.503401e-04 2.756236e-04 0.000000e+00 4.510204e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             80           81           83           84           85
##    1 3.758504e-05 3.758504e-05 7.517007e-05 1.252835e-05 5.011338e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             87           88           90           92           96
##    1 3.758504e-05 1.252835e-05 1.252835e-04 5.011338e-05 1.002268e-04
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y             97           98           99          100          102
##    1 2.756236e-04 1.002268e-04 1.252835e-05 1.127551e-04 5.011338e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            106          107          108          109          110
##    1 1.252835e-05 2.505669e-05 1.252835e-05 3.758504e-05 3.758504e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            111          112          113          114          116
##    1 2.505669e-05 5.011338e-05 1.252835e-05 1.252835e-05 1.252835e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            117          118          127          129          132
##    1 2.505669e-05 1.252835e-05 1.252835e-05 1.252835e-05 1.252835e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            133          135          137          138          142
##    1 1.252835e-05 1.252835e-05 0.000000e+00 2.505669e-05 1.252835e-05
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            151          152          153          155          168
##    1 1.252835e-05 2.505669e-05 1.252835e-05 3.758504e-05 0.000000e+00
##    2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

```
##    os
## Y            172          174          178          184          185
##   1 0.000000e+00 1.252835e-05 5.011338e-05 2.505669e-05 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            192          193          196          198          199
##   1 1.252835e-05 1.252835e-05 2.505669e-05 3.758504e-05 1.252835e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    os
## Y            207          607          748          836          866
##   1 1.252835e-05 3.871259e-03 1.653742e-03 1.252835e-05 1.140079e-03
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 5.494505e-03
##
##    channel
## Y              3            4            5           13           15
##   1 4.936168e-03 1.252835e-05 1.252835e-05 6.514740e-04 2.505669e-05
##   2 0.000000e+00 0.000000e+00 2.197802e-02 0.000000e+00 0.000000e+00
##    channel
## Y             17           18           19           21           22
##   1 1.227778e-03 2.255102e-04 6.652551e-03 1.628685e-03 7.517007e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 7.692308e-02 0.000000e+00
##    channel
## Y             24           30          101          105          107
##   1 3.006803e-04 1.002268e-03 1.175159e-02 6.326814e-03 4.597903e-02
##   2 0.000000e+00 0.000000e+00 4.945055e-02 0.000000e+00 5.494505e-03
##    channel
## Y            108          110          111          113          114
##   1 6.264173e-05 1.578572e-03 3.257370e-03 2.280159e-03 0.000000e+00
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.153846e-01 5.494505e-03
##    channel
## Y            115          116          118          120          121
##   1 7.654819e-03 2.944161e-03 1.240306e-03 8.143424e-04 2.485624e-02
##   2 0.000000e+00 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03
##    channel
## Y            122          123          124          125          126
##   1 1.385635e-02 8.519275e-04 2.367857e-03 6.289229e-03 1.378118e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            127          128          130          134          135
##   1 4.898583e-03 1.477092e-02 7.792631e-03 3.287438e-02 1.454541e-02
##   2 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03 0.000000e+00
##    channel
## Y            137          138          140          145          150
##   1 1.173906e-02 3.257370e-04 1.331763e-02 1.963192e-02 9.521542e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 1.648352e-02 0.000000e+00
##    channel
## Y            153          160          171          173          174
##   1 2.932886e-02 2.756236e-04 1.879252e-04 7.604706e-03 1.002268e-04
##   2 0.000000e+00 0.000000e+00 2.197802e-02 5.494505e-03 0.000000e+00
##    channel
## Y            178          182          203          205          208
##   1 2.941655e-02 7.767574e-04 6.264173e-05 2.394167e-02 5.011338e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
```

```
## Y            210          211          212          213          215
##   1 3.382653e-04 5.700397e-03 6.414513e-03 3.520465e-03 7.968028e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 3.296703e-01 0.000000e+00
##    channel
## Y            219          224          232          234          236
##   1 1.289167e-02 2.004535e-04 1.227778e-02 3.658277e-03 7.993084e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            237          242          243          244          245
##   1 1.430737e-02 1.252835e-03 1.916837e-03 6.665080e-03 4.740726e-02
##   2 0.000000e+00 0.000000e+00 2.747253e-02 0.000000e+00 0.000000e+00
##    channel
## Y            253          258          259          261          262
##   1 2.255102e-04 5.011338e-03 3.079467e-02 1.252835e-05 2.004535e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            265          266          268          272          274
##   1 3.039377e-02 5.211792e-03 1.127551e-04 3.758504e-05 5.011338e-05
##   2 1.098901e-02 0.000000e+00 0.000000e+00 0.000000e+00 4.945055e-02
##    channel
## Y            277          278          280          282          315
##   1 7.391724e-04 3.332540e-03 8.110851e-02 1.378118e-04 6.627495e-03
##   2 0.000000e+00 0.000000e+00 1.098901e-02 2.747253e-02 0.000000e+00
##    channel
## Y            317          319          320          322          325
##   1 3.457823e-03 3.984014e-03 1.127551e-04 2.756236e-04 1.753968e-03
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    channel
## Y            326          328          330          332          333
##   1 2.643481e-03 1.032336e-02 4.134354e-04 3.758504e-05 1.503401e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.648352e-02
##    channel
## Y            334          340          341          343          347
##   1 1.091219e-02 3.883787e-03 5.011338e-05 2.142347e-03 4.986281e-03
##   2 0.000000e+00 0.000000e+00 5.494505e-03 5.494505e-03 6.043956e-02
##    channel
## Y            349          353          356          360          361
##   1 5.976021e-03 2.505669e-05 1.503401e-04 7.015873e-04 3.758504e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            364          371          373          376          377
##   1 3.382653e-03 3.345068e-03 5.261905e-04 5.700397e-03 8.381463e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02
##    channel
## Y            379          386          391          400          401
##   1 1.839161e-02 4.710658e-03 3.382653e-04 4.209524e-03 7.842744e-03
##   2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            402          404          406          409          410
##   1 1.515930e-03 8.769842e-05 7.517007e-04 1.086208e-02 2.505669e-05
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y            411          412          416          417          419
##   1 2.004535e-04 3.194728e-03 2.380386e-04 2.067177e-03 1.252835e-05
```

```
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 2.197802e-02
##    channel
## Y           420          421          424          430          435
##   1 5.011338e-05 4.635488e-04 8.581917e-03 3.081973e-03 1.247823e-02
##   2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y           439          442          445          446          448
##   1 1.552262e-02 1.901803e-02 4.923640e-03 4.886055e-04 2.004535e-04
##   2 5.494505e-03 5.494505e-03 1.098901e-02 0.000000e+00 0.000000e+00
##    channel
## Y           449          450          451          452          453
##   1 1.215250e-03 1.628685e-04 1.252835e-05 9.308561e-03 2.630953e-04
##   2 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y           455          456          457          459          460
##   1 2.505669e-05 5.011338e-05 1.378118e-04 1.931871e-02 2.505669e-04
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y           463          465          466          467          469
##   1 5.061452e-03 0.000000e+00 1.497137e-02 1.641213e-03 1.464564e-02
##   2 0.000000e+00 5.494505e-03 1.098901e-02 0.000000e+00 0.000000e+00
##    channel
## Y           474          477          478          479          480
##   1 1.252835e-05 3.956451e-02 2.706123e-03 2.129819e-04 1.464564e-02
##   2 0.000000e+00 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00
##    channel
## Y           481          483          484          486          487
##   1 4.021599e-03 2.505669e-05 4.886055e-04 8.769842e-05 1.841667e-03
##   2 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.098901e-02
##    channel
## Y           488          489          490          496          497
##   1 5.011338e-05 1.429484e-02 2.756236e-04 7.642291e-04 2.430499e-03
##   2 0.000000e+00 5.494505e-03 0.000000e+00 0.000000e+00 0.000000e+00
##    channel
## Y           498
##   1 1.252835e-05
##   2 0.000000e+00
##
##    hour
## Y       [,1]     [,2]
##   1 9.317042 6.176409
##   2 8.774725 6.064007
##
##    app_clicks_h
## Y        [,1]     [,2]
##   1 131.21702 95.64334
##   2  20.54945 51.60983
##
##    channel_clicks_h
## Y        [,1]     [,2]
##   1 39.219534 43.99727
##   2  9.818681 20.96393
```

```r
# Prediction using test data
pred_nb2 = predict(nb2, test[,-5])

# Calculating accuracy for model
accuracy_nb2 = sum(obs_oL == pred_nb2) / length(obs_oL)
accuracy_nb2
```

```
## [1] 0.9868493
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_nb2)
```

```
##       pred_nb2
## obs_oL     1     2
##      1 19699   255
##      2     8    37
```

```r
# Well, it seems that we could improve a little bit compared to nb1.
# This algorithm seems to work very well with the original unbalanced dataset.
# Could be a good option.


##### TRAIN_OVER #####

# Model creation
# Model creation
nb3 = naiveBayes(is_attributed ~ app + device + os + channel + hour +
                   userID_clicks_h + app_clicks_h +
                   channel_clicks_h,
                 data = train_over, importance = TRUE)

# Prediction using test data
pred_nb3 = predict(nb3, test[,-5])

# Calculating accuracy for model
accuracy_nb3 = sum(obs_oL == pred_nb3) / length(obs_oL)
accuracy_nb3
```

```
## [1] 0.9977499
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_nb3)
```

```
##       pred_nb3
## obs_oL     1     2
##      1 19954     0
##      2    45     0
```

```r
# Here we are back to overfitting problem. Therefore, using the over sampling
# technique in this data set generated some problems.
# From now on, the train_over set will not be used anymore.


##### TRAIN_UNDER #####

# Model creation
nb4 = naiveBayes(is_attributed ~ app + device + os + channel +
```

```r
                  hour + ip_clicks_h + app_clicks_h + channel_clicks_h,
                data = train_under)

# Visualizing the results of model
nb4
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          1          2
## 0.93823431 0.06176569
##
## Conditional probabilities:
##    app
## Y              1            2            3            4            5
##   1 0.0325581395 0.0948121646 0.1785330948 0.0000000000 0.0007155635
##   2 0.0000000000 0.0000000000 0.0217391304 0.0000000000 0.0489130435
##    app
## Y              6            7            8            9           10
##   1 0.0125223614 0.0121645796 0.0175313059 0.1041144902 0.0039355993
##   2 0.0000000000 0.0000000000 0.0163043478 0.0326086957 0.0706521739
##    app
## Y             11           12           13           14           15
##   1 0.0150268336 0.1656529517 0.0221824687 0.0457960644 0.0844364937
##   2 0.0108695652 0.0054347826 0.0000000000 0.0000000000 0.0108695652
##    app
## Y             16           17           18           19           20
##   1 0.0000000000 0.0042933810 0.0754919499 0.0032200358 0.0089445438
##   2 0.0000000000 0.0000000000 0.0271739130 0.3260869565 0.0054347826
##    app
## Y             21           22           23           24           25
##   1 0.0225402504 0.0042933810 0.0103756708 0.0042933810 0.0103756708
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y             26           27           28           29           30
##   1 0.0135957066 0.0096601073 0.0057245081 0.0053667263 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0978260870 0.0000000000
##    app
## Y             31           32           33           34           35
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0108695652 0.1086956522
##    app
## Y             36           37           38           39           42
##   1 0.0010733453 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0108695652 0.0000000000
##    app
## Y             43           44           45           46           47
##   1 0.0000000000 0.0003577818 0.0000000000 0.0007155635 0.0007155635
##   2 0.0000000000 0.0000000000 0.0543478261 0.0000000000 0.0000000000
```

```
##    app
## Y            48          49          50          52          53
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0054347826 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    app
## Y            54          55          56          58          59
##   1 0.0000000000 0.0007155635 0.0010733453 0.0003577818 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0000000000
##    app
## Y            60          61          62          64          65
##   1 0.0003577818 0.0000000000 0.0000000000 0.0125223614 0.0003577818
##   2 0.0054347826 0.0000000000 0.0108695652 0.0000000000 0.0000000000
##    app
## Y            66          67          68          70          71
##   1 0.0000000000 0.0003577818 0.0010733453 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            72          74          75          76          78
##   1 0.0003577818 0.0000000000 0.0000000000 0.0003577818 0.0000000000
##   2 0.0271739130 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            79          80          81          82          83
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0108695652
##    app
## Y            84          85          86          87          88
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            91          92          93          94          95
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            96          97          99         100         101
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           103         104         105         107         108
##   1 0.0000000000 0.0000000000 0.0000000000 0.0003577818 0.0000000000
##   2 0.0108695652 0.0000000000 0.0000000000 0.0108695652 0.0108695652
##    app
## Y           109         110         112         115         116
##   1 0.0000000000 0.0007155635 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0054347826
##    app
## Y           117         118         119         121         122
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           123         124         125         133         134
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    app
## Y           137         139         145         146         148
```

```
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    app
## Y           149          150          151          158          160
##   1 0.0003577818 0.0000000000 0.0032200358 0.0000000000 0.0010733453
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           161          163          165          168          170
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           171          176          181          183          190
##   1 0.0000000000 0.0000000000 0.0000000000 0.0007155635 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           192          202          204          208          215
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0003577818
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    app
## Y           216          232          233          261          266
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0000000000
##    app
## Y           267          268          271          273          293
##   1 0.0000000000 0.0000000000 0.0000000000 0.0007155635 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           302          310          315          347          363
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           372          394          398          407          425
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           474          486          536          538          548
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           551
##   1 0.0000000000
##   2 0.0000000000
##
##    device
## Y             0            1            2            4            5
##   1 0.0075134168 0.8654740608 0.1162790698 0.0000000000 0.0000000000
##   2 0.2228260870 0.6304347826 0.0054347826 0.0054347826 0.0000000000
##    device
## Y             6            7            9           11           16
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0163043478
##    device
## Y            17           18           20           21           25
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

```
##   2 0.0000000000 0.0000000000 0.0000000000 0.0108695652 0.0000000000
##    device
## Y             30           33           36           37           40
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0054347826 0.0054347826 0.0054347826 0.0000000000 0.0108695652
##    device
## Y             49           50           53           56           58
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    device
## Y             59           60           67           74           76
##   1 0.0003577818 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    device
## Y             78           79           97          100          102
##   1 0.0003577818 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0108695652 0.0000000000 0.0000000000
##    device
## Y            103          106          109          114          116
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0054347826
##    device
## Y            124          129          154          160          163
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y            167          180          182          188          196
##   1 0.0003577818 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    device
## Y            202          203          210          211          220
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y            241          268          291          321          329
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y            347          351          362          374          385
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    device
## Y            386          414          420          486          516
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y            549          552          558          579          581
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0054347826
##    device
## Y            596          607          657          828          883
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
```

```
## Y           928          957         1042         1080         1162
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          1318         1422         1482         1728         1839
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          2120         2429         2980         3032         3282
##   1 0.0000000000 0.0000000000 0.0000000000 0.0032200358 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          3331         3543         3545         3866         3867
##   1 0.0000000000 0.0025044723 0.0000000000 0.0035778175 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##
##    os
## Y             0            1            2            3            4
##   1 0.0025044723 0.0078711986 0.0075134168 0.0107334526 0.0021466905
##   2 0.1195652174 0.0000000000 0.0054347826 0.0054347826 0.0108695652
##    os
## Y             5            6            7            8            9
##   1 0.0021466905 0.0279069767 0.0014311270 0.0246869410 0.0339892665
##   2 0.0000000000 0.0108695652 0.0271739130 0.0000000000 0.0000000000
##    os
## Y            10           11           12           13           14
##   1 0.0218246869 0.0096601073 0.0096601073 0.1971377460 0.0093023256
##   2 0.0163043478 0.0000000000 0.0000000000 0.1141304348 0.0108695652
##    os
## Y            15           16           17           18           19
##   1 0.0218246869 0.0178890877 0.0479427549 0.0425760286 0.2393559928
##   2 0.0217391304 0.0108695652 0.0271739130 0.0108695652 0.1793478261
##    os
## Y            20           21           22           23           24
##   1 0.0286225403 0.0000000000 0.0307692308 0.0082289803 0.0014311270
##   2 0.0108695652 0.0326086957 0.0271739130 0.0054347826 0.1141304348
##    os
## Y            25           26           27           28           29
##   1 0.0193202147 0.0050089445 0.0128801431 0.0100178891 0.0000000000
##   2 0.0163043478 0.0108695652 0.0326086957 0.0000000000 0.0652173913
##    os
## Y            30           31           32           34           35
##   1 0.0071556351 0.0042933810 0.0121645796 0.0017889088 0.0103756708
##   2 0.0108695652 0.0054347826 0.0108695652 0.0000000000 0.0054347826
##    os
## Y            36           37           38           39           40
##   1 0.0060822898 0.0161001789 0.0039355993 0.0003577818 0.0060822898
##   2 0.0108695652 0.0054347826 0.0163043478 0.0000000000 0.0000000000
##    os
## Y            41           42           43           44           45
##   1 0.0118067979 0.0042933810 0.0032200358 0.0003577818 0.0000000000
##   2 0.0000000000 0.0000000000 0.0108695652 0.0000000000 0.0000000000
##    os
## Y            46           47           48           49           50
```

```
##   1 0.0032200358 0.0107334526 0.0003577818 0.0057245081 0.0000000000
##   2 0.0000000000 0.0163043478 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            52           53           55           56           57
##   1 0.0010733453 0.0085867621 0.0010733453 0.0017889088 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            58           59           60           61           62
##   1 0.0025044723 0.0003577818 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    os
## Y            63           64           65           66           67
##   1 0.0014311270 0.0010733453 0.0017889088 0.0014311270 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            68           69           70           71           73
##   1 0.0000000000 0.0007155635 0.0007155635 0.0000000000 0.0007155635
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            74           76           77           78           79
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0010733453
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            80           81           83           84           85
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            87           88           90           92           96
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            97           98           99          100          102
##   1 0.0010733453 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           106          107          108          109          110
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           111          112          113          114          116
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           117          118          127          129          132
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           133          135          137          138          142
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           151          152          153          155          168
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

```
##    os
## Y           172         174         178         184         185
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           192         193         196         198         199
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           207         607         748         836         866
##   1 0.0003577818 0.0035778175 0.0028622540 0.0000000000 0.0035778175
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0054347826
##
##    channel
## Y             3           4           5          13          15
##   1 0.0067978533 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0217391304 0.0000000000 0.0000000000
##    channel
## Y            17          18          19          21          22
##   1 0.0000000000 0.0007155635 0.0067978533 0.0003577818 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0760869565 0.0000000000
##    channel
## Y            24          30         101         105         107
##   1 0.0000000000 0.0010733453 0.0143112701 0.0042933810 0.0389982111
##   2 0.0000000000 0.0000000000 0.0489130435 0.0000000000 0.0054347826
##    channel
## Y           108         110         111         113         114
##   1 0.0000000000 0.0003577818 0.0014311270 0.0017889088 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.1141304348 0.0054347826
##    channel
## Y           115         116         118         120         121
##   1 0.0096601073 0.0035778175 0.0000000000 0.0025044723 0.0261180680
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0054347826
##    channel
## Y           122         123         124         125         126
##   1 0.0128801431 0.0014311270 0.0010733453 0.0082289803 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           127         128         130         134         135
##   1 0.0032200358 0.0157423971 0.0093023256 0.0343470483 0.0178890877
##   2 0.0000000000 0.0000000000 0.0054347826 0.0054347826 0.0000000000
##    channel
## Y           137         138         140         145         150
##   1 0.0125223614 0.0003577818 0.0135957066 0.0182468694 0.0007155635
##   2 0.0000000000 0.0000000000 0.0000000000 0.0163043478 0.0000000000
##    channel
## Y           153         160         171         173         174
##   1 0.0533094812 0.0003577818 0.0007155635 0.0071556351 0.0000000000
##   2 0.0000000000 0.0000000000 0.0217391304 0.0054347826 0.0000000000
##    channel
## Y           178         182         203         205         208
##   1 0.0264758497 0.0007155635 0.0000000000 0.0135957066 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
```

```
## Y           210          211          212          213          215
##   1 0.0007155635 0.0042933810 0.0039355993 0.0025044723 0.0110912343
##   2 0.0000000000 0.0000000000 0.0000000000 0.3369565217 0.0000000000
##    channel
## Y           219          224          232          234          236
##   1 0.0060822898 0.0000000000 0.0125223614 0.0135957066 0.0042933810
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           237          242          243          244          245
##   1 0.0125223614 0.0007155635 0.0014311270 0.0060822898 0.0479427549
##   2 0.0000000000 0.0000000000 0.0271739130 0.0000000000 0.0000000000
##    channel
## Y           253          258          259          261          262
##   1 0.0000000000 0.0053667263 0.0483005367 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           265          266          268          272          274
##   1 0.0325581395 0.0028622540 0.0000000000 0.0003577818 0.0000000000
##   2 0.0108695652 0.0000000000 0.0000000000 0.0000000000 0.0489130435
##    channel
## Y           277          278          280          282          315
##   1 0.0007155635 0.0017889088 0.0604651163 0.0000000000 0.0064400716
##   2 0.0000000000 0.0000000000 0.0108695652 0.0271739130 0.0000000000
##    channel
## Y           317          319          320          322          325
##   1 0.0035778175 0.0032200358 0.0000000000 0.0007155635 0.0000000000
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    channel
## Y           326          328          330          332          333
##   1 0.0304114490 0.0067978533 0.0007155635 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0163043478
##    channel
## Y           334          340          341          343          347
##   1 0.0053667263 0.0053667263 0.0000000000 0.0046511628 0.0128801431
##   2 0.0000000000 0.0000000000 0.0054347826 0.0054347826 0.0597826087
##    channel
## Y           349          353          356          360          361
##   1 0.0025044723 0.0000000000 0.0010733453 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           364          371          373          376          377
##   1 0.0000000000 0.0028622540 0.0007155635 0.0003577818 0.0067978533
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0108695652
##    channel
## Y           379          386          391          400          401
##   1 0.0168157424 0.0028622540 0.0000000000 0.0028622540 0.0071556351
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           402          404          406          409          410
##   1 0.0007155635 0.0000000000 0.0014311270 0.0096601073 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           411          412          416          417          419
##   1 0.0000000000 0.0032200358 0.0000000000 0.0032200358 0.0000000000
```

```
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0217391304
##    channel
## Y           420          421          424          430          435
##   1 0.0000000000 0.0003577818 0.0067978533 0.0021466905 0.0096601073
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           439          442          445          446          448
##   1 0.0128801431 0.0135957066 0.0071556351 0.0003577818 0.0003577818
##   2 0.0054347826 0.0054347826 0.0108695652 0.0000000000 0.0000000000
##    channel
## Y           449          450          451          452          453
##   1 0.0025044723 0.0000000000 0.0000000000 0.0125223614 0.0003577818
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           455          456          457          459          460
##   1 0.0000000000 0.0000000000 0.0010733453 0.0196779964 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           463          465          466          467          469
##   1 0.0035778175 0.0000000000 0.0171735242 0.0032200358 0.0157423971
##   2 0.0000000000 0.0054347826 0.0108695652 0.0000000000 0.0000000000
##    channel
## Y           474          477          478          479          480
##   1 0.0000000000 0.0379248658 0.0028622540 0.0003577818 0.0139534884
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    channel
## Y           481          483          484          486          487
##   1 0.0035778175 0.0000000000 0.0007155635 0.0010733453 0.0017889088
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0108695652
##    channel
## Y           488          489          490          496          497
##   1 0.0000000000 0.0114490161 0.0007155635 0.0007155635 0.0021466905
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           498
##   1 0.0000000000
##   2 0.0000000000
##
##    hour
## Y        [,1]     [,2]
##   1 10.550984 6.021064
##   2  8.809783 6.043713
##
##    ip_clicks_h
## Y       [,1]     [,2]
##   1 5.611091 4.822484
##   2 1.250000 1.936844
##
##    app_clicks_h
## Y       [,1]     [,2]
##   1 5.959928 4.569103
##   2 2.070652 1.955910
##
##    channel_clicks_h
```

```
## Y        [,1]     [,2]
##   1 2.325224 1.633070
##   2 1.673913 1.102493
```

```
# Prediction using test data
pred_nb4 = predict(nb4, test[,-5])

# Calculating accuracy for model
accuracy_nb4 = sum(obs_oL == pred_nb4) / length(obs_oL)
accuracy_nb4
```

```
## [1] 0.9062953
```

```
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_nb4)
```

```
##       pred_nb4
## obs_oL     1     2
##      1 18087  1867
##      2     7    38
```

```
# This model did reasonably but worse than the models created with the train set

# Let's try a second version with the train_under set

# Model creation
nb5 = naiveBayes(is_attributed ~ app + device + os + channel +
                   hour + userID_clicks_h + app_clicks_h + channel_clicks_h,
                 data = train_under)

# Visualizing the results of model
nb5
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          1          2
## 0.93823431 0.06176569
##
## Conditional probabilities:
##    app
## Y              1            2            3            4            5
##   1 0.0325581395 0.0948121646 0.1785330948 0.0000000000 0.0007155635
##   2 0.0000000000 0.0000000000 0.0217391304 0.0000000000 0.0489130435
##    app
## Y              6            7            8            9           10
##   1 0.0125223614 0.0121645796 0.0175313059 0.1041144902 0.0039355993
##   2 0.0000000000 0.0000000000 0.0163043478 0.0326086957 0.0706521739
##    app
## Y             11           12           13           14           15
##   1 0.0150268336 0.1656529517 0.0221824687 0.0457960644 0.0844364937
##   2 0.0108695652 0.0054347826 0.0000000000 0.0000000000 0.0108695652
```

```
##    app
## Y            16           17           18           19           20
##   1 0.0000000000 0.0042933810 0.0754919499 0.0032200358 0.0089445438
##   2 0.0000000000 0.0000000000 0.0271739130 0.3260869565 0.0054347826
##    app
## Y            21           22           23           24           25
##   1 0.0225402504 0.0042933810 0.0103756708 0.0042933810 0.0103756708
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            26           27           28           29           30
##   1 0.0135957066 0.0096601073 0.0057245081 0.0053667263 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0978260870 0.0000000000
##    app
## Y            31           32           33           34           35
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0108695652 0.1086956522
##    app
## Y            36           37           38           39           42
##   1 0.0010733453 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0108695652 0.0000000000
##    app
## Y            43           44           45           46           47
##   1 0.0000000000 0.0003577818 0.0000000000 0.0007155635 0.0007155635
##   2 0.0000000000 0.0000000000 0.0543478261 0.0000000000 0.0000000000
##    app
## Y            48           49           50           52           53
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0054347826 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    app
## Y            54           55           56           58           59
##   1 0.0000000000 0.0007155635 0.0010733453 0.0003577818 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0000000000
##    app
## Y            60           61           62           64           65
##   1 0.0003577818 0.0000000000 0.0000000000 0.0125223614 0.0003577818
##   2 0.0054347826 0.0000000000 0.0108695652 0.0000000000 0.0000000000
##    app
## Y            66           67           68           70           71
##   1 0.0000000000 0.0003577818 0.0010733453 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            72           74           75           76           78
##   1 0.0003577818 0.0000000000 0.0000000000 0.0003577818 0.0000000000
##   2 0.0271739130 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            79           80           81           82           83
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0108695652
##    app
## Y            84           85           86           87           88
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            91           92           93           94           95
```

```
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            96           97           99          100          101
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           103          104          105          107          108
##   1 0.0000000000 0.0000000000 0.0000000000 0.0003577818 0.0000000000
##   2 0.0108695652 0.0000000000 0.0000000000 0.0108695652 0.0108695652
##    app
## Y           109          110          112          115          116
##   1 0.0000000000 0.0007155635 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0054347826
##    app
## Y           117          118          119          121          122
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           123          124          125          133          134
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0003577818
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    app
## Y           137          139          145          146          148
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    app
## Y           149          150          151          158          160
##   1 0.0003577818 0.0000000000 0.0032200358 0.0000000000 0.0010733453
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           161          163          165          168          170
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           171          176          181          183          190
##   1 0.0000000000 0.0000000000 0.0000000000 0.0007155635 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           192          202          204          208          215
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0003577818
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    app
## Y           216          232          233          261          266
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0000000000
##    app
## Y           267          268          271          273          293
##   1 0.0000000000 0.0000000000 0.0000000000 0.0007155635 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           302          310          315          347          363
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

```
##    app
## Y           372          394          398          407          425
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           474          486          536          538          548
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           551
##   1 0.0000000000
##   2 0.0000000000
##
##    device
## Y             0            1            2            4            5
##   1 0.0075134168 0.8654740608 0.1162790698 0.0000000000 0.0000000000
##   2 0.2228260870 0.6304347826 0.0054347826 0.0054347826 0.0000000000
##    device
## Y             6            7            9           11           16
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0163043478
##    device
## Y            17           18           20           21           25
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0108695652 0.0000000000
##    device
## Y            30           33           36           37           40
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0054347826 0.0054347826 0.0054347826 0.0000000000 0.0108695652
##    device
## Y            49           50           53           56           58
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    device
## Y            59           60           67           74           76
##   1 0.0003577818 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    device
## Y            78           79           97          100          102
##   1 0.0003577818 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0108695652 0.0000000000 0.0000000000
##    device
## Y           103          106          109          114          116
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0054347826
##    device
## Y           124          129          154          160          163
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y           167          180          182          188          196
##   1 0.0003577818 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    device
```

```
## Y          202         203         210         211         220
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          241         268         291         321         329
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          347         351         362         374         385
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          386         414         420         486         516
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          549         552         558         579         581
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0054347826
##    device
## Y          596         607         657         828         883
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          928         957        1042        1080        1162
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    device
## Y         1318        1422        1482        1728        1839
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y         2120        2429        2980        3032        3282
##   1 0.0000000000 0.0000000000 0.0000000000 0.0032200358 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y         3331        3543        3545        3866        3867
##   1 0.0000000000 0.0025044723 0.0000000000 0.0035778175 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##
##    os
## Y            0           1           2           3           4
##   1 0.0025044723 0.0078711986 0.0075134168 0.0107334526 0.0021466905
##   2 0.1195652174 0.0000000000 0.0054347826 0.0054347826 0.0108695652
##    os
## Y            5           6           7           8           9
##   1 0.0021466905 0.0279069767 0.0014311270 0.0246869410 0.0339892665
##   2 0.0000000000 0.0108695652 0.0271739130 0.0000000000 0.0000000000
##    os
## Y           10          11          12          13          14
##   1 0.0218246869 0.0096601073 0.0096601073 0.1971377460 0.0093023256
##   2 0.0163043478 0.0000000000 0.0000000000 0.1141304348 0.0108695652
##    os
## Y           15          16          17          18          19
```

```
##   1 0.0218246869 0.0178890877 0.0479427549 0.0425760286 0.2393559928
##   2 0.0217391304 0.0108695652 0.0271739130 0.0108695652 0.1793478261
##    os
## Y            20           21           22           23           24
##   1 0.0286225403 0.0000000000 0.0307692308 0.0082289803 0.0014311270
##   2 0.0108695652 0.0326086957 0.0271739130 0.0054347826 0.1141304348
##    os
## Y            25           26           27           28           29
##   1 0.0193202147 0.0050089445 0.0128801431 0.0100178891 0.0000000000
##   2 0.0163043478 0.0108695652 0.0326086957 0.0000000000 0.0652173913
##    os
## Y            30           31           32           34           35
##   1 0.0071556351 0.0042933810 0.0121645796 0.0017889088 0.0103756708
##   2 0.0108695652 0.0054347826 0.0108695652 0.0000000000 0.0054347826
##    os
## Y            36           37           38           39           40
##   1 0.0060822898 0.0161001789 0.0039355993 0.0003577818 0.0060822898
##   2 0.0108695652 0.0054347826 0.0163043478 0.0000000000 0.0000000000
##    os
## Y            41           42           43           44           45
##   1 0.0118067979 0.0042933810 0.0032200358 0.0003577818 0.0000000000
##   2 0.0000000000 0.0000000000 0.0108695652 0.0000000000 0.0000000000
##    os
## Y            46           47           48           49           50
##   1 0.0032200358 0.0107334526 0.0003577818 0.0057245081 0.0000000000
##   2 0.0000000000 0.0163043478 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            52           53           55           56           57
##   1 0.0010733453 0.0085867621 0.0010733453 0.0017889088 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            58           59           60           61           62
##   1 0.0025044723 0.0003577818 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0054347826 0.0000000000 0.0054347826 0.0000000000
##    os
## Y            63           64           65           66           67
##   1 0.0014311270 0.0010733453 0.0017889088 0.0014311270 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            68           69           70           71           73
##   1 0.0000000000 0.0007155635 0.0007155635 0.0000000000 0.0007155635
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            74           76           77           78           79
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0010733453
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            80           81           83           84           85
##   1 0.0000000000 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            87           88           90           92           96
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

```
##    os
## Y             97           98           99          100          102
##   1 0.0010733453 0.0003577818 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            106          107          108          109          110
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            111          112          113          114          116
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            117          118          127          129          132
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            133          135          137          138          142
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            151          152          153          155          168
##   1 0.0000000000 0.0000000000 0.0003577818 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            172          174          178          184          185
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            192          193          196          198          199
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            207          607          748          836          866
##   1 0.0003577818 0.0035778175 0.0028622540 0.0000000000 0.0035778175
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0054347826
##
##    channel
## Y              3            4            5           13           15
##   1 0.0067978533 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0217391304 0.0000000000 0.0000000000
##    channel
## Y             17           18           19           21           22
##   1 0.0000000000 0.0007155635 0.0067978533 0.0003577818 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0760869565 0.0000000000
##    channel
## Y             24           30          101          105          107
##   1 0.0000000000 0.0010733453 0.0143112701 0.0042933810 0.0389982111
##   2 0.0000000000 0.0000000000 0.0489130435 0.0000000000 0.0054347826
##    channel
## Y            108          110          111          113          114
##   1 0.0000000000 0.0003577818 0.0014311270 0.0017889088 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.1141304348 0.0054347826
##    channel
```

```
## Y            115         116         118         120         121
##   1 0.0096601073 0.0035778175 0.0000000000 0.0025044723 0.0261180680
##   2 0.0000000000 0.0000000000 0.0000000000 0.0054347826 0.0054347826
##    channel
## Y            122         123         124         125         126
##   1 0.0128801431 0.0014311270 0.0010733453 0.0082289803 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y            127         128         130         134         135
##   1 0.0032200358 0.0157423971 0.0093023256 0.0343470483 0.0178890877
##   2 0.0000000000 0.0000000000 0.0054347826 0.0054347826 0.0000000000
##    channel
## Y            137         138         140         145         150
##   1 0.0125223614 0.0003577818 0.0135957066 0.0182468694 0.0007155635
##   2 0.0000000000 0.0000000000 0.0000000000 0.0163043478 0.0000000000
##    channel
## Y            153         160         171         173         174
##   1 0.0533094812 0.0003577818 0.0007155635 0.0071556351 0.0000000000
##   2 0.0000000000 0.0000000000 0.0217391304 0.0054347826 0.0000000000
##    channel
## Y            178         182         203         205         208
##   1 0.0264758497 0.0007155635 0.0000000000 0.0135957066 0.0003577818
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y            210         211         212         213         215
##   1 0.0007155635 0.0042933810 0.0039355993 0.0025044723 0.0110912343
##   2 0.0000000000 0.0000000000 0.0000000000 0.3369565217 0.0000000000
##    channel
## Y            219         224         232         234         236
##   1 0.0060822898 0.0000000000 0.0125223614 0.0135957066 0.0042933810
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y            237         242         243         244         245
##   1 0.0125223614 0.0007155635 0.0014311270 0.0060822898 0.0479427549
##   2 0.0000000000 0.0000000000 0.0271739130 0.0000000000 0.0000000000
##    channel
## Y            253         258         259         261         262
##   1 0.0000000000 0.0053667263 0.0483005367 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y            265         266         268         272         274
##   1 0.0325581395 0.0028622540 0.0000000000 0.0003577818 0.0000000000
##   2 0.0108695652 0.0000000000 0.0000000000 0.0000000000 0.0489130435
##    channel
## Y            277         278         280         282         315
##   1 0.0007155635 0.0017889088 0.0604651163 0.0000000000 0.0064400716
##   2 0.0000000000 0.0000000000 0.0108695652 0.0271739130 0.0000000000
##    channel
## Y            317         319         320         322         325
##   1 0.0035778175 0.0032200358 0.0000000000 0.0007155635 0.0000000000
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    channel
## Y            326         328         330         332         333
##   1 0.0304114490 0.0067978533 0.0007155635 0.0000000000 0.0000000000
```

```
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0163043478
##    channel
## Y           334          340          341          343          347
##   1 0.0053667263 0.0053667263 0.0000000000 0.0046511628 0.0128801431
##   2 0.0000000000 0.0000000000 0.0054347826 0.0054347826 0.0597826087
##    channel
## Y           349          353          356          360          361
##   1 0.0025044723 0.0000000000 0.0010733453 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           364          371          373          376          377
##   1 0.0000000000 0.0028622540 0.0007155635 0.0003577818 0.0067978533
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0108695652
##    channel
## Y           379          386          391          400          401
##   1 0.0168157424 0.0028622540 0.0000000000 0.0028622540 0.0071556351
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           402          404          406          409          410
##   1 0.0007155635 0.0000000000 0.0014311270 0.0096601073 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           411          412          416          417          419
##   1 0.0000000000 0.0032200358 0.0000000000 0.0032200358 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0217391304
##    channel
## Y           420          421          424          430          435
##   1 0.0000000000 0.0003577818 0.0067978533 0.0021466905 0.0096601073
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           439          442          445          446          448
##   1 0.0128801431 0.0135957066 0.0071556351 0.0003577818 0.0003577818
##   2 0.0054347826 0.0054347826 0.0108695652 0.0000000000 0.0000000000
##    channel
## Y           449          450          451          452          453
##   1 0.0025044723 0.0000000000 0.0000000000 0.0125223614 0.0003577818
##   2 0.0054347826 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           455          456          457          459          460
##   1 0.0000000000 0.0000000000 0.0010733453 0.0196779964 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           463          465          466          467          469
##   1 0.0035778175 0.0000000000 0.0171735242 0.0032200358 0.0157423971
##   2 0.0000000000 0.0054347826 0.0108695652 0.0000000000 0.0000000000
##    channel
## Y           474          477          478          479          480
##   1 0.0000000000 0.0379248658 0.0028622540 0.0003577818 0.0139534884
##   2 0.0000000000 0.0000000000 0.0054347826 0.0000000000 0.0000000000
##    channel
## Y           481          483          484          486          487
##   1 0.0035778175 0.0000000000 0.0007155635 0.0010733453 0.0017889088
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0108695652
##    channel
```

```
## Y            488          489          490          496          497
##   1 0.0000000000 0.0114490161 0.0007155635 0.0007155635 0.0021466905
##   2 0.0000000000 0.0054347826 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y            498
##   1 0.0000000000
##   2 0.0000000000
##
##    hour
## Y        [,1]     [,2]
##   1 10.550984 6.021064
##   2  8.809783 6.043713
##
##    userID_clicks_h
## Y       [,1]      [,2]
##   1 1.448658 1.0281267
##   2 1.021739 0.1462284
##
##    app_clicks_h
## Y       [,1]     [,2]
##   1 5.959928 4.569103
##   2 2.070652 1.955910
##
##    channel_clicks_h
## Y       [,1]     [,2]
##   1 2.325224 1.633070
##   2 1.673913 1.102493
```

```r
# Prediction using test data
pred_nb5 = predict(nb5, test[,-5])

# Calculating accuracy for model
accuracy_nb5 = sum(obs_oL == pred_nb5) / length(obs_oL)
accuracy_nb5
```

```
## [1] 0.9029951
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_nb5)
```

```
##       pred_nb5
## obs_oL     1     2
##      1 18020  1934
##      2     6    39
```

```r
# It did not get any better. Let's try the train_underOver set



##### TRAIN_UNDEROVER #####

# Model creation
nb6 = naiveBayes(is_attributed ~ app + device + os + channel +
                   hour + ip_channel_clicks_h + app_clicks_h,
                 data = train_underOver)
```

```r
# Visualizing the results of model
nb6
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##      1      2
## 0.5025 0.4975
##
## Conditional probabilities:
##    app
## Y              1            2            3            4            5
##   1 0.0427860697 0.1049751244 0.1810945274 0.0000000000 0.0009950249
##   2 0.0000000000 0.0000000000 0.0190954774 0.0000000000 0.0512562814
##    app
## Y              6            7            8            9           10
##   1 0.0114427861 0.0124378109 0.0174129353 0.1079601990 0.0024875622
##   2 0.0000000000 0.0000000000 0.0160804020 0.0412060302 0.0723618090
##    app
## Y             11           12           13           14           15
##   1 0.0149253731 0.1771144279 0.0124378109 0.0512437811 0.0766169154
##   2 0.0120603015 0.0055276382 0.0000000000 0.0000000000 0.0115577889
##    app
## Y             16           17           18           19           20
##   1 0.0000000000 0.0034825871 0.0771144279 0.0024875622 0.0104477612
##   2 0.0000000000 0.0000000000 0.0236180905 0.3125628141 0.0030150754
##    app
## Y             21           22           23           24           25
##   1 0.0203980100 0.0024875622 0.0049751244 0.0034825871 0.0069651741
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y             26           27           28           29           30
##   1 0.0134328358 0.0069651741 0.0029850746 0.0024875622 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.1040201005 0.0000000000
##    app
## Y             31           32           33           34           35
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0075376884 0.1100502513
##    app
## Y             36           37           38           39           42
##   1 0.0004975124 0.0000000000 0.0004975124 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0125628141 0.0000000000
##    app
## Y             43           44           45           46           47
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0009950249
##   2 0.0000000000 0.0000000000 0.0562814070 0.0000000000 0.0000000000
##    app
## Y             48           49           50           52           53
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

```
##   2 0.0075376884 0.0000000000 0.0070351759 0.0000000000 0.0000000000
##    app
## Y            54           55           56           58           59
##   1 0.0000000000 0.0004975124 0.0000000000 0.0009950249 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0065326633 0.0000000000
##    app
## Y            60           61           62           64           65
##   1 0.0000000000 0.0000000000 0.0000000000 0.0094527363 0.0004975124
##   2 0.0035175879 0.0000000000 0.0100502513 0.0000000000 0.0000000000
##    app
## Y            66           67           68           70           71
##   1 0.0000000000 0.0000000000 0.0014925373 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            72           74           75           76           78
##   1 0.0004975124 0.0000000000 0.0000000000 0.0014925373 0.0000000000
##   2 0.0226130653 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            79           80           81           82           83
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0004975124
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0150753769
##    app
## Y            84           85           86           87           88
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0004975124
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            91           92           93           94           95
##   1 0.0000000000 0.0000000000 0.0004975124 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y            96           97           99          100          101
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0065326633 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           103          104          105          107          108
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0115577889 0.0000000000 0.0000000000 0.0110552764 0.0115577889
##    app
## Y           109          110          112          115          116
##   1 0.0000000000 0.0009950249 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0035175879
##    app
## Y           117          118          119          121          122
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           123          124          125          133          134
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0025125628 0.0000000000 0.0000000000
##    app
## Y           137          139          145          146          148
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0055276382 0.0000000000 0.0000000000
##    app
```

```
## Y           149          150          151          158          160
##   1 0.0004975124 0.0000000000 0.0034825871 0.0000000000 0.0019900498
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           161          163          165          168          170
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           171          176          181          183          190
##   1 0.0000000000 0.0000000000 0.0000000000 0.0014925373 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           192          202          204          208          215
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0004975124
##   2 0.0000000000 0.0040201005 0.0000000000 0.0050251256 0.0000000000
##    app
## Y           216          232          233          261          266
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0075376884 0.0000000000
##    app
## Y           267          268          271          273          293
##   1 0.0000000000 0.0000000000 0.0000000000 0.0004975124 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           302          310          315          347          363
##   1 0.0000000000 0.0004975124 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           372          394          398          407          425
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           474          486          536          538          548
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    app
## Y           551
##   1 0.0000000000
##   2 0.0000000000
##
##    device
## Y             0            1            2            4            5
##   1 0.0064676617 0.8741293532 0.1074626866 0.0000000000 0.0000000000
##   2 0.2170854271 0.6386934673 0.0045226131 0.0065326633 0.0000000000
##    device
## Y             6            7            9           11           16
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0045226131 0.0000000000 0.0000000000 0.0000000000 0.0195979899
##    device
## Y            17           18           20           21           25
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0090452261 0.0000000000
##    device
## Y            30           33           36           37           40
```

```
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0040201005 0.0065326633 0.0065326633 0.0000000000 0.0105527638
##     device
## Y            49           50           53           56           58
##    1 0.0000000000 0.0004975124 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0055276382 0.0000000000 0.0045226131 0.0000000000
##     device
## Y            59           60           67           74           76
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0075376884 0.0000000000 0.0065326633 0.0000000000
##     device
## Y            78           79           97          100          102
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0105527638 0.0000000000 0.0000000000
##     device
## Y           103          106          109          114          116
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0065326633 0.0000000000 0.0045226131
##     device
## Y           124          129          154          160          163
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     device
## Y           167          180          182          188          196
##    1 0.0004975124 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0025125628 0.0000000000 0.0055276382 0.0000000000
##     device
## Y           202          203          210          211          220
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     device
## Y           241          268          291          321          329
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     device
## Y           347          351          362          374          385
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0035175879 0.0000000000 0.0000000000 0.0000000000
##     device
## Y           386          414          420          486          516
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     device
## Y           549          552          558          579          581
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0055276382 0.0050251256
##     device
## Y           596          607          657          828          883
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     device
## Y           928          957         1042         1080         1162
##    1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    2 0.0000000000 0.0045226131 0.0000000000 0.0000000000 0.0000000000
```

```
##    device
## Y          1318        1422        1482        1728        1839
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          2120        2429        2980        3032        3282
##   1 0.0000000000 0.0000000000 0.0000000000 0.0044776119 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    device
## Y          3331        3543        3545        3866        3867
##   1 0.0000000000 0.0034825871 0.0000000000 0.0029850746 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##
##    os
## Y             0           1           2           3           4
##   1 0.0014925373 0.0069651741 0.0074626866 0.0099502488 0.0019900498
##   2 0.1211055276 0.0000000000 0.0055276382 0.0060301508 0.0115577889
##    os
## Y             5           6           7           8           9
##   1 0.0019900498 0.0313432836 0.0009950249 0.0248756219 0.0323383085
##   2 0.0000000000 0.0100502513 0.0301507538 0.0000000000 0.0000000000
##    os
## Y            10          11          12          13          14
##   1 0.0194029851 0.0104477612 0.0119402985 0.2004975124 0.0119402985
##   2 0.0145728643 0.0000000000 0.0000000000 0.1100502513 0.0105527638
##    os
## Y            15          16          17          18          19
##   1 0.0233830846 0.0174129353 0.0472636816 0.0388059701 0.2393034826
##   2 0.0256281407 0.0090452261 0.0301507538 0.0090452261 0.1909547739
##    os
## Y            20          21          22          23          24
##   1 0.0313432836 0.0000000000 0.0298507463 0.0054726368 0.0019900498
##   2 0.0115577889 0.0286432161 0.0251256281 0.0040201005 0.1090452261
##    os
## Y            25          26          27          28          29
##   1 0.0174129353 0.0044776119 0.0109452736 0.0109452736 0.0000000000
##   2 0.0185929648 0.0140703518 0.0376884422 0.0000000000 0.0653266332
##    os
## Y            30          31          32          34          35
##   1 0.0059701493 0.0049751244 0.0144278607 0.0024875622 0.0114427861
##   2 0.0095477387 0.0040201005 0.0085427136 0.0000000000 0.0035175879
##    os
## Y            36          37          38          39          40
##   1 0.0059701493 0.0154228856 0.0034825871 0.0000000000 0.0029850746
##   2 0.0070351759 0.0040201005 0.0185929648 0.0000000000 0.0000000000
##    os
## Y            41          42          43          44          45
##   1 0.0149253731 0.0074626866 0.0014925373 0.0009950249 0.0000000000
##   2 0.0000000000 0.0000000000 0.0135678392 0.0000000000 0.0000000000
##    os
## Y            46          47          48          49          50
##   1 0.0064676617 0.0059701493 0.0000000000 0.0064676617 0.0000000000
##   2 0.0000000000 0.0125628141 0.0000000000 0.0000000000 0.0000000000
##    os
```

```
## Y            52          53          55          56          57
##   1 0.0019900498 0.0049751244 0.0004975124 0.0014925373 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            58          59          60          61          62
##   1 0.0029850746 0.0000000000 0.0009950249 0.0000000000 0.0000000000
##   2 0.0000000000 0.0045226131 0.0000000000 0.0060301508 0.0000000000
##    os
## Y            63          64          65          66          67
##   1 0.0014925373 0.0009950249 0.0014925373 0.0024875622 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            68          69          70          71          73
##   1 0.0000000000 0.0004975124 0.0009950249 0.0000000000 0.0004975124
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            74          76          77          78          79
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0004975124
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            80          81          83          84          85
##   1 0.0000000000 0.0004975124 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            87          88          90          92          96
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y            97          98          99         100         102
##   1 0.0009950249 0.0009950249 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           106         107         108         109         110
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           111         112         113         114         116
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           117         118         127         129         132
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           133         135         137         138         142
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           151         152         153         155         168
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           172         174         178         184         185
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

```
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           192          193          196          198          199
##   1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    os
## Y           207          607          748          836          866
##   1 0.0009950249 0.0049751244 0.0034825871 0.0000000000 0.0029850746
##   2 0.0000000000 0.0000000000 0.0025125628 0.0000000000 0.0070351759
##
##    channel
## Y             3            4            5           13           15
##   1 0.0079601990 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0256281407 0.0000000000 0.0000000000
##    channel
## Y            17           18           19           21           22
##   1 0.0000000000 0.0009950249 0.0074626866 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0743718593 0.0000000000
##    channel
## Y            24           30          101          105          107
##   1 0.0000000000 0.0014925373 0.0164179104 0.0039800995 0.0427860697
##   2 0.0000000000 0.0000000000 0.0427135678 0.0000000000 0.0030150754
##    channel
## Y           108          110          111          113          114
##   1 0.0000000000 0.0004975124 0.0009950249 0.0024875622 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.1170854271 0.0025125628
##    channel
## Y           115          116          118          120          121
##   1 0.0089552239 0.0024875622 0.0000000000 0.0024875622 0.0213930348
##   2 0.0000000000 0.0000000000 0.0000000000 0.0065326633 0.0050251256
##    channel
## Y           122          123          124          125          126
##   1 0.0154228856 0.0019900498 0.0014925373 0.0074626866 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           127          128          130          134          135
##   1 0.0024875622 0.0139303483 0.0089552239 0.0353233831 0.0139303483
##   2 0.0000000000 0.0000000000 0.0060301508 0.0060301508 0.0000000000
##    channel
## Y           137          138          140          145          150
##   1 0.0129353234 0.0000000000 0.0149253731 0.0164179104 0.0004975124
##   2 0.0000000000 0.0000000000 0.0000000000 0.0160804020 0.0000000000
##    channel
## Y           153          160          171          173          174
##   1 0.0467661692 0.0004975124 0.0004975124 0.0089552239 0.0000000000
##   2 0.0000000000 0.0000000000 0.0261306533 0.0030150754 0.0000000000
##    channel
## Y           178          182          203          205          208
##   1 0.0283582090 0.0004975124 0.0000000000 0.0159203980 0.0004975124
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           210          211          212          213          215
##   1 0.0000000000 0.0034825871 0.0034825871 0.0019900498 0.0109452736
##   2 0.0000000000 0.0000000000 0.0000000000 0.3422110553 0.0000000000
```

```
##    channel
## Y           219          224          232          234          236
##   1 0.0089552239 0.0000000000 0.0114427861 0.0129353234 0.0044776119
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           237          242          243          244          245
##   1 0.0114427861 0.0009950249 0.0019900498 0.0079601990 0.0398009950
##   2 0.0000000000 0.0000000000 0.0266331658 0.0000000000 0.0000000000
##    channel
## Y           253          258          259          261          262
##   1 0.0000000000 0.0039800995 0.0427860697 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           265          266          268          272          274
##   1 0.0402985075 0.0059701493 0.0000000000 0.0004975124 0.0000000000
##   2 0.0110552764 0.0000000000 0.0000000000 0.0000000000 0.0492462312
##    channel
## Y           277          278          280          282          315
##   1 0.0000000000 0.0014925373 0.0592039801 0.0000000000 0.0059701493
##   2 0.0000000000 0.0000000000 0.0135678392 0.0251256281 0.0000000000
##    channel
## Y           317          319          320          322          325
##   1 0.0034825871 0.0024875622 0.0000000000 0.0004975124 0.0000000000
##   2 0.0000000000 0.0000000000 0.0055276382 0.0000000000 0.0000000000
##    channel
## Y           326          328          330          332          333
##   1 0.0353233831 0.0089552239 0.0004975124 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0180904523
##    channel
## Y           334          340          341          343          347
##   1 0.0059701493 0.0049751244 0.0000000000 0.0029850746 0.0124378109
##   2 0.0000000000 0.0000000000 0.0025125628 0.0055276382 0.0537688442
##    channel
## Y           349          353          356          360          361
##   1 0.0049751244 0.0000000000 0.0014925373 0.0000000000 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           364          371          373          376          377
##   1 0.0000000000 0.0024875622 0.0000000000 0.0000000000 0.0069651741
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0115577889
##    channel
## Y           379          386          391          400          401
##   1 0.0179104478 0.0024875622 0.0000000000 0.0014925373 0.0094527363
##   2 0.0060301508 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           402          404          406          409          410
##   1 0.0004975124 0.0000000000 0.0004975124 0.0099502488 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##    channel
## Y           411          412          416          417          419
##   1 0.0000000000 0.0049751244 0.0000000000 0.0029850746 0.0000000000
##   2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0206030151
##    channel
## Y           420          421          424          430          435
```

```
##    1 0.0000000000 0.0004975124 0.0064676617 0.0014925373 0.0109452736
##    2 0.0000000000 0.0040201005 0.0000000000 0.0000000000 0.0000000000
##     channel
## Y            439          442          445          446          448
##    1 0.0159203980 0.0174129353 0.0084577114 0.0009950249 0.0000000000
##    2 0.0050251256 0.0065326633 0.0140703518 0.0000000000 0.0000000000
##     channel
## Y            449          450          451          452          453
##    1 0.0004975124 0.0000000000 0.0000000000 0.0124378109 0.0000000000
##    2 0.0045226131 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     channel
## Y            455          456          457          459          460
##    1 0.0000000000 0.0000000000 0.0019900498 0.0154228856 0.0000000000
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
##     channel
## Y            463          465          466          467          469
##    1 0.0034825871 0.0000000000 0.0144278607 0.0044776119 0.0129353234
##    2 0.0000000000 0.0045226131 0.0155778894 0.0000000000 0.0000000000
##     channel
## Y            474          477          478          479          480
##    1 0.0000000000 0.0323383085 0.0039800995 0.0004975124 0.0174129353
##    2 0.0000000000 0.0000000000 0.0030150754 0.0000000000 0.0000000000
##     channel
## Y            481          483          484          486          487
##    1 0.0049751244 0.0000000000 0.0009950249 0.0004975124 0.0019900498
##    2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0120603015
##     channel
## Y            488          489          490          496          497
##    1 0.0000000000 0.0164179104 0.0000000000 0.0000000000 0.0009950249
##    2 0.0000000000 0.0050251256 0.0000000000 0.0000000000 0.0000000000
##     channel
## Y            498
##    1 0.0000000000
##    2 0.0000000000
##
##     hour
## Y         [,1]     [,2]
##    1 10.525871 5.885127
##    2  8.745226 6.031251
##
##     ip_channel_clicks_h
## Y         [,1]      [,2]
##    1  1.841791 0.9506723
##    2 11.974874 3.6105499
##
##     app_clicks_h
## Y        [,1]      [,2]
##    1  6.21393  5.254783
##    2 17.23015 11.184332
```

```r
# Prediction using test data
pred_nb6 = predict(nb6, test[,-5])

# Calculating accuracy for model
```

```r
accuracy_nb6 = sum(obs_oL == pred_nb6) / length(obs_oL)
accuracy_nb6
```

```
## [1] 0.5437772
```

```r
# Visualizing the Confusion Matrix for the observed and predicted values
table(obs_oL, pred_nb6)
```

```
##       pred_nb6
## obs_oL     1     2
##      1 10847  9107
##      2    17    28
```

```r
# From the above analysis, the Naive Bayes algorithm did a much better job than
# the Random Forest algorithm. In the Naive Bayes, it was possible to include the
# categorical variables, which probably was one of the reasons for achieving
# better results with this algorithm. Moreover, the Naive Bayes also handled very
# well the unbalanced original train set.

# Remove all the variables from the environment
rm(process_data, train_over, train_under, train_underOver,
   train_over_cor, train_under_cor, train_underOver_cor,
   rf1, rf2, rf3, rf4, rf5, rf6, rf7, rf8, rf9, rf10,
   nb1, nb2, nb3, nb4, nb5, nb6,
   pred_rf1, pred_rf2, pred_rf3, pred_rf4, pred_rf5,
   pred_rf6, pred_rf7, pred_rf8, pred_rf9, pred_rf10,
   pred_nb1, pred_nb2, pred_nb3, pred_nb4, pred_nb5, pred_nb6,
   accuracy_rf1, accuracy_rf2, accuracy_rf3, accuracy_rf4,
   accuracy_rf5, accuracy_rf6, accuracy_rf7, accuracy_rf8,
   accuracy_rf9, accuracy_rf10, accuracy_nb1, accuracy_nb2,
   accuracy_nb3, accuracy_nb4, accuracy_nb5, accuracy_nb6)

# Clear console
cat("\014")
```

```r
# Clear plots and free unused memory
graphics.off()
invisible(gc())

# From now on, let's work with the original train set and test one more algorithm
# to create our final version of the model. Let's try the following algorithms:

# - Naive Bayes - already showed good results
# - SVM - linear and radial kernels

# Let's choose different sets of predictors to be used in different versions
# of the model.

# 1. app + device + os + channel + hour + userID_clicks_h + app_clicks_h + channel_clicks_h
# 2. app + device + os + channel + hour + userID_clicks_h + app_clicks_h
# 3. app + device + os + channel + hour + userID_clicks_h + channel_clicks_h
# 4. app + device + os + channel + hour + userID_clicks_h
# 5. app + device + os + channel + hour + app_clicks_h
# 6. app + device + os + channel + hour + channel_clicks_h

# 7. app + channel + hour + userID_clicks_h + app_clicks_h + channel_clicks_h
# 8. app + channel + hour + userID_clicks_h + app_clicks_h
# 9. app + channel + hour + userID_clicks_h + channel_clicks_h
# 10. app + channel + hour + userID_clicks_h

# 11. app + device + os + channel + hour + ip_clicks_h + app_clicks_h + channel_clicks_h
# 12. app + device + os + channel + hour + ip_clicks_h + app_clicks_h
# 13. app + device + os + channel + hour + ip_clicks_h + channel_clicks_h
# 14. app + device + os + channel + hour + ip_clicks_h
# 15. app + device + os + channel + hour + app_clicks_h
# 16. app + device + os + channel + hour + channel_clicks_h

# 17. app + channel + hour + ip_clicks_h + app_clicks_h + channel_clicks_h
# 18. app + channel + hour + ip_clicks_h + app_clicks_h
# 19. app + channel + hour + ip_clicks_h + channel_clicks_h
# 20. app + channel + hour + ip_clicks_h


# In total, there will be created 60 model version, being 20 constructed by the
# algorithm Naive Bayes and 40 by the SVM, including 20 using the linear kernel
# and 20 using the radial kernel.

# First, let's create a list to hold all the combinations of predictors (20)
predictors_list = list(
  "is_attributed ~ app + device + os + channel + hour + userID_clicks_h + app_clicks_h + channel_clicks_
  "is_attributed ~ app + device + os + channel + hour + userID_clicks_h + app_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + userID_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + app_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + channel_clicks_h",
  "is_attributed ~ app + channel + hour + userID_clicks_h + app_clicks_h + channel_clicks_h",
  "is_attributed ~ app + channel + hour + userID_clicks_h + app_clicks_h",
  "is_attributed ~ app + channel + hour + userID_clicks_h + channel_clicks_h",
```

```r
  "is_attributed ~ app + channel + hour + userID_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + ip_clicks_h + app_clicks_h + channel_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + ip_clicks_h + app_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + ip_clicks_h + channel_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + ip_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + app_clicks_h",
  "is_attributed ~ app + device + os + channel + hour + channel_clicks_h",
  "is_attributed ~ app + channel + hour + ip_clicks_h + app_clicks_h + channel_clicks_h",
  "is_attributed ~ app + channel + hour + ip_clicks_h + app_clicks_h",
  "is_attributed ~ app + channel + hour + ip_clicks_h + channel_clicks_h",
  "is_attributed ~ app + channel + hour + ip_clicks_h"
)


###### NAIVE BAYES ######

# Creating four lists to hold the results
nb_model_list = list()
nb_pred_list = list()
nb_accuracy_list = list()
nb_tables_list = list()

# Iterating throught the predictors_list to generate models
for (i in 1:length(predictors_list)) {

  # Model creation and filling in the nb_model_list
  nb_model_list[[i]] = naiveBayes(as.formula(predictors_list[[i]]),
                                  data = train)

  # Prediction using test data and filling in the nb_pred_list
  nb_pred_list[[i]] = predict(nb_model_list[[i]], test[,-5])

  # Calculating accuracy for model and filling in the nb_accuracy_list
  nb_accuracy_list[[i]] = sum(obs_oL == nb_pred_list[[i]]) / length(obs_oL)

  # Filling the list nb_tables_list with the confusion matrix of results
  nb_tables_list[[i]] = table(obs_oL, nb_pred_list[[i]])
  print(paste("Step", i, "of 20!!"))
}
```

```
## [1] "Step 1 of 20!!"
## [1] "Step 2 of 20!!"
## [1] "Step 3 of 20!!"
## [1] "Step 4 of 20!!"
## [1] "Step 5 of 20!!"
## [1] "Step 6 of 20!!"
## [1] "Step 7 of 20!!"
## [1] "Step 8 of 20!!"
## [1] "Step 9 of 20!!"
## [1] "Step 10 of 20!!"
## [1] "Step 11 of 20!!"
## [1] "Step 12 of 20!!"
## [1] "Step 13 of 20!!"
## [1] "Step 14 of 20!!"
```

```
## [1] "Step 15 of 20!!"
## [1] "Step 16 of 20!!"
## [1] "Step 17 of 20!!"
## [1] "Step 18 of 20!!"
## [1] "Step 19 of 20!!"
## [1] "Step 20 of 20!!"
```

```r
# Free unused memory
invisible(gc())


###### SVM - linear kernel ######

# Creating four lists to hold the results
svm1_model_list = list()
svm1_pred_list = list()
svm1_accuracy_list = list()
svm1_tables_list = list()


# Iterating throught the predictors_list to generate models
for (i in 1:length(predictors_list)) {

  # Model creation and filling in the svm1_model_list
  svm1_model_list[[i]] = svm(as.formula(predictors_list[[i]]),
                             data = train,
                             type = "C-classification",
                             kernel = "linear")

  # Prediction using test data and filling in the svm1_pred_list
  svm1_pred_list[[i]] = predict(svm1_model_list[[i]], test[,-5])

  # Calculating accuracy for model and filling in the svm1_accuracy_list
  svm1_accuracy_list[[i]] = sum(obs_oL == svm1_pred_list[[i]]) / length(obs_oL)

  # Filling the list svm1_tables_list with the confusion matrix of results
  svm1_tables_list[[i]] = table(obs_oL, svm1_pred_list[[i]])
  print(paste("Step", i, "of 20!!"))
}
```

```
## [1] "Step 1 of 20!!"
## [1] "Step 2 of 20!!"
## [1] "Step 3 of 20!!"
## [1] "Step 4 of 20!!"
## [1] "Step 5 of 20!!"
## [1] "Step 6 of 20!!"
## [1] "Step 7 of 20!!"
## [1] "Step 8 of 20!!"
## [1] "Step 9 of 20!!"
## [1] "Step 10 of 20!!"
## [1] "Step 11 of 20!!"
## [1] "Step 12 of 20!!"
## [1] "Step 13 of 20!!"
## [1] "Step 14 of 20!!"
## [1] "Step 15 of 20!!"
```

```
## [1] "Step 16 of 20!!"
## [1] "Step 17 of 20!!"
## [1] "Step 18 of 20!!"
## [1] "Step 19 of 20!!"
## [1] "Step 20 of 20!!"
```

```r
# Free unused memory
invisible(gc())


###### SVM - radial kernel ######

# Creating four lists to hold the results
svm2_model_list = list()
svm2_pred_list = list()
svm2_accuracy_list = list()
svm2_tables_list = list()

# Iterating throught the predictors_list to generate models
for (i in 1:length(predictors_list)) {

  # Model creation and filling in the svm1_model_list
  svm2_model_list[[i]] = svm(as.formula(predictors_list[[i]]),
                             data = train,
                             type = "C-classification",
                             kernel = "radial")

  # Prediction using test data and filling in the svm1_pred_list
  svm2_pred_list[[i]] = predict(svm2_model_list[[i]], test[,-5])

  # Calculating accuracy for model and filling in the svm1_accuracy_list
  svm2_accuracy_list[[i]] = sum(obs_oL == svm2_pred_list[[i]]) / length(obs_oL)

  # Filling the list svm1_tables_list with the confusion matrix of results
  svm2_tables_list[[i]] = table(obs_oL, svm2_pred_list[[i]])
  print(paste("Step", i, "of 20!!"))
}
```

```
## [1] "Step 1 of 20!!"
## [1] "Step 2 of 20!!"
## [1] "Step 3 of 20!!"
## [1] "Step 4 of 20!!"
## [1] "Step 5 of 20!!"
## [1] "Step 6 of 20!!"
## [1] "Step 7 of 20!!"
## [1] "Step 8 of 20!!"
## [1] "Step 9 of 20!!"
## [1] "Step 10 of 20!!"
## [1] "Step 11 of 20!!"
## [1] "Step 12 of 20!!"
## [1] "Step 13 of 20!!"
## [1] "Step 14 of 20!!"
## [1] "Step 15 of 20!!"
## [1] "Step 16 of 20!!"
## [1] "Step 17 of 20!!"
```

```
## [1] "Step 18 of 20!!"
## [1] "Step 19 of 20!!"
## [1] "Step 20 of 20!!"
```

```r
# Free unused memory
invisible(gc())

# Let's visualize the accuracy and confusion matrix of the models:
for (i in 1:length(nb_accuracy_list)) {
  print(paste("Accuracy for version", i, "- NB:", nb_accuracy_list[[i]]))
  print(paste("Accuracy for version", i, "- SVM_linear:", svm1_accuracy_list[[i]]))
  print(paste("Accuracy for version", i, "- SVM_radial:", svm2_accuracy_list[[i]]))
  print(paste("Confusion matrix for version", i, "- NB:", nb_tables_list[[i]]))
  print(paste("Confusion matrix for version", i, "- SVM_linear:", svm1_tables_list[[i]]))
  print(paste("Confusion matrix for version", i, "- SVM_radial:", svm2_tables_list[[i]]))
}
```

```
## [1] "Accuracy for version 1 - NB: 0.986149307465373"
## [1] "Accuracy for version 1 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 1 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 1 - NB: 19685"
## [2] "Confusion matrix for version 1 - NB: 8"
## [3] "Confusion matrix for version 1 - NB: 269"
## [4] "Confusion matrix for version 1 - NB: 37"
## [1] "Confusion matrix for version 1 - SVM_linear: 19951"
## [2] "Confusion matrix for version 1 - SVM_linear: 39"
## [3] "Confusion matrix for version 1 - SVM_linear: 3"
## [4] "Confusion matrix for version 1 - SVM_linear: 6"
## [1] "Confusion matrix for version 1 - SVM_radial: 19954"
## [2] "Confusion matrix for version 1 - SVM_radial: 45"
## [3] "Confusion matrix for version 1 - SVM_radial: 0"
## [4] "Confusion matrix for version 1 - SVM_radial: 0"
## [1] "Accuracy for version 2 - NB: 0.988249412470623"
## [1] "Accuracy for version 2 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 2 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 2 - NB: 19730"
## [2] "Confusion matrix for version 2 - NB: 11"
## [3] "Confusion matrix for version 2 - NB: 224"
## [4] "Confusion matrix for version 2 - NB: 34"
## [1] "Confusion matrix for version 2 - SVM_linear: 19951"
## [2] "Confusion matrix for version 2 - SVM_linear: 39"
## [3] "Confusion matrix for version 2 - SVM_linear: 3"
## [4] "Confusion matrix for version 2 - SVM_linear: 6"
## [1] "Confusion matrix for version 2 - SVM_radial: 19954"
## [2] "Confusion matrix for version 2 - SVM_radial: 45"
## [3] "Confusion matrix for version 2 - SVM_radial: 0"
## [4] "Confusion matrix for version 2 - SVM_radial: 0"
## [1] "Accuracy for version 3 - NB: 0.988499424971249"
## [1] "Accuracy for version 3 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 3 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 3 - NB: 19735"
## [2] "Confusion matrix for version 3 - NB: 11"
## [3] "Confusion matrix for version 3 - NB: 219"
## [4] "Confusion matrix for version 3 - NB: 34"
## [1] "Confusion matrix for version 3 - SVM_linear: 19951"
```

```
## [2] "Confusion matrix for version 3 - SVM_linear: 39"
## [3] "Confusion matrix for version 3 - SVM_linear: 3"
## [4] "Confusion matrix for version 3 - SVM_linear: 6"
## [1] "Confusion matrix for version 3 - SVM_radial: 19954"
## [2] "Confusion matrix for version 3 - SVM_radial: 45"
## [3] "Confusion matrix for version 3 - SVM_radial: 0"
## [4] "Confusion matrix for version 3 - SVM_radial: 0"
## [1] "Accuracy for version 4 - NB: 0.989799489974499"
## [1] "Accuracy for version 4 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 4 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 4 - NB: 19763"
## [2] "Confusion matrix for version 4 - NB: 13"
## [3] "Confusion matrix for version 4 - NB: 191"
## [4] "Confusion matrix for version 4 - NB: 32"
## [1] "Confusion matrix for version 4 - SVM_linear: 19951"
## [2] "Confusion matrix for version 4 - SVM_linear: 39"
## [3] "Confusion matrix for version 4 - SVM_linear: 3"
## [4] "Confusion matrix for version 4 - SVM_linear: 6"
## [1] "Confusion matrix for version 4 - SVM_radial: 19954"
## [2] "Confusion matrix for version 4 - SVM_radial: 45"
## [3] "Confusion matrix for version 4 - SVM_radial: 0"
## [4] "Confusion matrix for version 4 - SVM_radial: 0"
## [1] "Accuracy for version 5 - NB: 0.988249412470623"
## [1] "Accuracy for version 5 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 5 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 5 - NB: 19730"
## [2] "Confusion matrix for version 5 - NB: 11"
## [3] "Confusion matrix for version 5 - NB: 224"
## [4] "Confusion matrix for version 5 - NB: 34"
## [1] "Confusion matrix for version 5 - SVM_linear: 19951"
## [2] "Confusion matrix for version 5 - SVM_linear: 39"
## [3] "Confusion matrix for version 5 - SVM_linear: 3"
## [4] "Confusion matrix for version 5 - SVM_linear: 6"
## [1] "Confusion matrix for version 5 - SVM_radial: 19954"
## [2] "Confusion matrix for version 5 - SVM_radial: 45"
## [3] "Confusion matrix for version 5 - SVM_radial: 0"
## [4] "Confusion matrix for version 5 - SVM_radial: 0"
## [1] "Accuracy for version 6 - NB: 0.988649432471624"
## [1] "Accuracy for version 6 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 6 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 6 - NB: 19739"
## [2] "Confusion matrix for version 6 - NB: 12"
## [3] "Confusion matrix for version 6 - NB: 215"
## [4] "Confusion matrix for version 6 - NB: 33"
## [1] "Confusion matrix for version 6 - SVM_linear: 19951"
## [2] "Confusion matrix for version 6 - SVM_linear: 39"
## [3] "Confusion matrix for version 6 - SVM_linear: 3"
## [4] "Confusion matrix for version 6 - SVM_linear: 6"
## [1] "Confusion matrix for version 6 - SVM_radial: 19954"
## [2] "Confusion matrix for version 6 - SVM_radial: 45"
## [3] "Confusion matrix for version 6 - SVM_radial: 0"
## [4] "Confusion matrix for version 6 - SVM_radial: 0"
## [1] "Accuracy for version 7 - NB: 0.982499124956248"
## [1] "Accuracy for version 7 - SVM_linear: 0.997849892494625"
```

```
## [1] "Accuracy for version 7 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 7 - NB: 19612"
## [2] "Confusion matrix for version 7 - NB: 8"
## [3] "Confusion matrix for version 7 - NB: 342"
## [4] "Confusion matrix for version 7 - NB: 37"
## [1] "Confusion matrix for version 7 - SVM_linear: 19952"
## [2] "Confusion matrix for version 7 - SVM_linear: 41"
## [3] "Confusion matrix for version 7 - SVM_linear: 2"
## [4] "Confusion matrix for version 7 - SVM_linear: 4"
## [1] "Confusion matrix for version 7 - SVM_radial: 19954"
## [2] "Confusion matrix for version 7 - SVM_radial: 45"
## [3] "Confusion matrix for version 7 - SVM_radial: 0"
## [4] "Confusion matrix for version 7 - SVM_radial: 0"
## [1] "Accuracy for version 8 - NB: 0.987849392469623"
## [1] "Accuracy for version 8 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 8 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 8 - NB: 19719"
## [2] "Confusion matrix for version 8 - NB: 8"
## [3] "Confusion matrix for version 8 - NB: 235"
## [4] "Confusion matrix for version 8 - NB: 37"
## [1] "Confusion matrix for version 8 - SVM_linear: 19952"
## [2] "Confusion matrix for version 8 - SVM_linear: 41"
## [3] "Confusion matrix for version 8 - SVM_linear: 2"
## [4] "Confusion matrix for version 8 - SVM_linear: 4"
## [1] "Confusion matrix for version 8 - SVM_radial: 19954"
## [2] "Confusion matrix for version 8 - SVM_radial: 45"
## [3] "Confusion matrix for version 8 - SVM_radial: 0"
## [4] "Confusion matrix for version 8 - SVM_radial: 0"
## [1] "Accuracy for version 9 - NB: 0.989949497474874"
## [1] "Accuracy for version 9 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 9 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 9 - NB: 19762"
## [2] "Confusion matrix for version 9 - NB: 9"
## [3] "Confusion matrix for version 9 - NB: 192"
## [4] "Confusion matrix for version 9 - NB: 36"
## [1] "Confusion matrix for version 9 - SVM_linear: 19952"
## [2] "Confusion matrix for version 9 - SVM_linear: 41"
## [3] "Confusion matrix for version 9 - SVM_linear: 2"
## [4] "Confusion matrix for version 9 - SVM_linear: 4"
## [1] "Confusion matrix for version 9 - SVM_radial: 19954"
## [2] "Confusion matrix for version 9 - SVM_radial: 45"
## [3] "Confusion matrix for version 9 - SVM_radial: 0"
## [4] "Confusion matrix for version 9 - SVM_radial: 0"
## [1] "Accuracy for version 10 - NB: 0.990999549977499"
## [1] "Accuracy for version 10 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 10 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 10 - NB: 19786"
## [2] "Confusion matrix for version 10 - NB: 12"
## [3] "Confusion matrix for version 10 - NB: 168"
## [4] "Confusion matrix for version 10 - NB: 33"
## [1] "Confusion matrix for version 10 - SVM_linear: 19952"
## [2] "Confusion matrix for version 10 - SVM_linear: 41"
## [3] "Confusion matrix for version 10 - SVM_linear: 2"
## [4] "Confusion matrix for version 10 - SVM_linear: 4"
```

```
## [1] "Confusion matrix for version 10 - SVM_radial: 19954"
## [2] "Confusion matrix for version 10 - SVM_radial: 45"
## [3] "Confusion matrix for version 10 - SVM_radial: 0"
## [4] "Confusion matrix for version 10 - SVM_radial: 0"
## [1] "Accuracy for version 11 - NB: 0.986599329966498"
## [1] "Accuracy for version 11 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 11 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 11 - NB: 19695"
## [2] "Confusion matrix for version 11 - NB: 9"
## [3] "Confusion matrix for version 11 - NB: 259"
## [4] "Confusion matrix for version 11 - NB: 36"
## [1] "Confusion matrix for version 11 - SVM_linear: 19951"
## [2] "Confusion matrix for version 11 - SVM_linear: 39"
## [3] "Confusion matrix for version 11 - SVM_linear: 3"
## [4] "Confusion matrix for version 11 - SVM_linear: 6"
## [1] "Confusion matrix for version 11 - SVM_radial: 19954"
## [2] "Confusion matrix for version 11 - SVM_radial: 45"
## [3] "Confusion matrix for version 11 - SVM_radial: 0"
## [4] "Confusion matrix for version 11 - SVM_radial: 0"
## [1] "Accuracy for version 12 - NB: 0.988249412470623"
## [1] "Accuracy for version 12 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 12 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 12 - NB: 19730"
## [2] "Confusion matrix for version 12 - NB: 11"
## [3] "Confusion matrix for version 12 - NB: 224"
## [4] "Confusion matrix for version 12 - NB: 34"
## [1] "Confusion matrix for version 12 - SVM_linear: 19951"
## [2] "Confusion matrix for version 12 - SVM_linear: 39"
## [3] "Confusion matrix for version 12 - SVM_linear: 3"
## [4] "Confusion matrix for version 12 - SVM_linear: 6"
## [1] "Confusion matrix for version 12 - SVM_radial: 19954"
## [2] "Confusion matrix for version 12 - SVM_radial: 45"
## [3] "Confusion matrix for version 12 - SVM_radial: 0"
## [4] "Confusion matrix for version 12 - SVM_radial: 0"
## [1] "Accuracy for version 13 - NB: 0.988649432471624"
## [1] "Accuracy for version 13 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 13 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 13 - NB: 19738"
## [2] "Confusion matrix for version 13 - NB: 11"
## [3] "Confusion matrix for version 13 - NB: 216"
## [4] "Confusion matrix for version 13 - NB: 34"
## [1] "Confusion matrix for version 13 - SVM_linear: 19951"
## [2] "Confusion matrix for version 13 - SVM_linear: 39"
## [3] "Confusion matrix for version 13 - SVM_linear: 3"
## [4] "Confusion matrix for version 13 - SVM_linear: 6"
## [1] "Confusion matrix for version 13 - SVM_radial: 19954"
## [2] "Confusion matrix for version 13 - SVM_radial: 45"
## [3] "Confusion matrix for version 13 - SVM_radial: 0"
## [4] "Confusion matrix for version 13 - SVM_radial: 0"
## [1] "Accuracy for version 14 - NB: 0.989949497474874"
## [1] "Accuracy for version 14 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 14 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 14 - NB: 19766"
## [2] "Confusion matrix for version 14 - NB: 13"
```

```
## [3] "Confusion matrix for version 14 - NB: 188"
## [4] "Confusion matrix for version 14 - NB: 32"
## [1] "Confusion matrix for version 14 - SVM_linear: 19951"
## [2] "Confusion matrix for version 14 - SVM_linear: 39"
## [3] "Confusion matrix for version 14 - SVM_linear: 3"
## [4] "Confusion matrix for version 14 - SVM_linear: 6"
## [1] "Confusion matrix for version 14 - SVM_radial: 19954"
## [2] "Confusion matrix for version 14 - SVM_radial: 45"
## [3] "Confusion matrix for version 14 - SVM_radial: 0"
## [4] "Confusion matrix for version 14 - SVM_radial: 0"
## [1] "Accuracy for version 15 - NB: 0.988249412470623"
## [1] "Accuracy for version 15 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 15 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 15 - NB: 19730"
## [2] "Confusion matrix for version 15 - NB: 11"
## [3] "Confusion matrix for version 15 - NB: 224"
## [4] "Confusion matrix for version 15 - NB: 34"
## [1] "Confusion matrix for version 15 - SVM_linear: 19951"
## [2] "Confusion matrix for version 15 - SVM_linear: 39"
## [3] "Confusion matrix for version 15 - SVM_linear: 3"
## [4] "Confusion matrix for version 15 - SVM_linear: 6"
## [1] "Confusion matrix for version 15 - SVM_radial: 19954"
## [2] "Confusion matrix for version 15 - SVM_radial: 45"
## [3] "Confusion matrix for version 15 - SVM_radial: 0"
## [4] "Confusion matrix for version 15 - SVM_radial: 0"
## [1] "Accuracy for version 16 - NB: 0.988649432471624"
## [1] "Accuracy for version 16 - SVM_linear: 0.99789989499475"
## [1] "Accuracy for version 16 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 16 - NB: 19739"
## [2] "Confusion matrix for version 16 - NB: 12"
## [3] "Confusion matrix for version 16 - NB: 215"
## [4] "Confusion matrix for version 16 - NB: 33"
## [1] "Confusion matrix for version 16 - SVM_linear: 19951"
## [2] "Confusion matrix for version 16 - SVM_linear: 39"
## [3] "Confusion matrix for version 16 - SVM_linear: 3"
## [4] "Confusion matrix for version 16 - SVM_linear: 6"
## [1] "Confusion matrix for version 16 - SVM_radial: 19954"
## [2] "Confusion matrix for version 16 - SVM_radial: 45"
## [3] "Confusion matrix for version 16 - SVM_radial: 0"
## [4] "Confusion matrix for version 16 - SVM_radial: 0"
## [1] "Accuracy for version 17 - NB: 0.982799139956998"
## [1] "Accuracy for version 17 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 17 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 17 - NB: 19617"
## [2] "Confusion matrix for version 17 - NB: 7"
## [3] "Confusion matrix for version 17 - NB: 337"
## [4] "Confusion matrix for version 17 - NB: 38"
## [1] "Confusion matrix for version 17 - SVM_linear: 19952"
## [2] "Confusion matrix for version 17 - SVM_linear: 41"
## [3] "Confusion matrix for version 17 - SVM_linear: 2"
## [4] "Confusion matrix for version 17 - SVM_linear: 4"
## [1] "Confusion matrix for version 17 - SVM_radial: 19954"
## [2] "Confusion matrix for version 17 - SVM_radial: 45"
## [3] "Confusion matrix for version 17 - SVM_radial: 0"
```

```
## [4] "Confusion matrix for version 17 - SVM_radial: 0"
## [1] "Accuracy for version 18 - NB: 0.989249462473124"
## [1] "Accuracy for version 18 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 18 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 18 - NB: 19748"
## [2] "Confusion matrix for version 18 - NB: 9"
## [3] "Confusion matrix for version 18 - NB: 206"
## [4] "Confusion matrix for version 18 - NB: 36"
## [1] "Confusion matrix for version 18 - SVM_linear: 19952"
## [2] "Confusion matrix for version 18 - SVM_linear: 41"
## [3] "Confusion matrix for version 18 - SVM_linear: 2"
## [4] "Confusion matrix for version 18 - SVM_linear: 4"
## [1] "Confusion matrix for version 18 - SVM_radial: 19954"
## [2] "Confusion matrix for version 18 - SVM_radial: 45"
## [3] "Confusion matrix for version 18 - SVM_radial: 0"
## [4] "Confusion matrix for version 18 - SVM_radial: 0"
## [1] "Accuracy for version 19 - NB: 0.989949497474874"
## [1] "Accuracy for version 19 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 19 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 19 - NB: 19763"
## [2] "Confusion matrix for version 19 - NB: 10"
## [3] "Confusion matrix for version 19 - NB: 191"
## [4] "Confusion matrix for version 19 - NB: 35"
## [1] "Confusion matrix for version 19 - SVM_linear: 19952"
## [2] "Confusion matrix for version 19 - SVM_linear: 41"
## [3] "Confusion matrix for version 19 - SVM_linear: 2"
## [4] "Confusion matrix for version 19 - SVM_linear: 4"
## [1] "Confusion matrix for version 19 - SVM_radial: 19954"
## [2] "Confusion matrix for version 19 - SVM_radial: 45"
## [3] "Confusion matrix for version 19 - SVM_radial: 0"
## [4] "Confusion matrix for version 19 - SVM_radial: 0"
## [1] "Accuracy for version 20 - NB: 0.990999549977499"
## [1] "Accuracy for version 20 - SVM_linear: 0.997849892494625"
## [1] "Accuracy for version 20 - SVM_radial: 0.997749887494375"
## [1] "Confusion matrix for version 20 - NB: 19787"
## [2] "Confusion matrix for version 20 - NB: 13"
## [3] "Confusion matrix for version 20 - NB: 167"
## [4] "Confusion matrix for version 20 - NB: 32"
## [1] "Confusion matrix for version 20 - SVM_linear: 19952"
## [2] "Confusion matrix for version 20 - SVM_linear: 41"
## [3] "Confusion matrix for version 20 - SVM_linear: 2"
## [4] "Confusion matrix for version 20 - SVM_linear: 4"
## [1] "Confusion matrix for version 20 - SVM_radial: 19954"
## [2] "Confusion matrix for version 20 - SVM_radial: 45"
## [3] "Confusion matrix for version 20 - SVM_radial: 0"
## [4] "Confusion matrix for version 20 - SVM_radial: 0"
```

```
# First of all, all versions of SVM considering the radial kernel produced models
# that were probably over fitted and did not work at all with the test data. Althoug
# it presented a high level o accuracy (approx. 99.8%), it was not able to classify
# any register into one class (download the app >>> is_attributed = 1)

# In the case of SVM using the linear kernel, most of models presented similar results,
# with very high accuracy but a lot of wrong classification for one of the classes,
```

```r
# which is the one representing that an user did download the app >> is_attributed = 1

# The best versions of the model were created with the Naive Bayes algorithm. The
# accuracy varied from 98.25% (version 7) to 99.10% (versions 10 and 20). Let's
# check these versions a bit further

# Creating two new lists to hold the errors (typeI and typeII)
error_typeI = list()
error_typeII = list()

# Iterating through nb_tables_list and filling the errors lists
for (i in 1:length(nb_tables_list)) {
  error_typeI[[i]] = round(nb_tables_list[[i]][3])
  error_typeII[[i]] = round(nb_tables_list[[i]][2])
}

# Creating a data.frame with errors, model versions and accuracies.
errors = data.frame(error_typeI = unlist(error_typeI),
                    error_typeII = unlist(error_typeII),
                    model = sapply(c(1:20), function(x) {paste0("Version-",x)}),
                    accuracy = sapply(c(1:20), function(x) {round(nb_accuracy_list[[x]] * 100, 2)}))

# Sorting the data.frame based on error_typeI and error_typeII
errors %>%
  arrange(error_typeI, error_typeII)
```

```
##    error_typeI error_typeII      model accuracy
## 1          167           13 Version-20    99.10
## 2          168           12 Version-10    99.10
## 3          188           13 Version-14    98.99
## 4          191           10 Version-19    98.99
## 5          191           13  Version-4    98.98
## 6          192            9  Version-9    98.99
## 7          206            9 Version-18    98.92
## 8          215           12  Version-6    98.86
## 9          215           12 Version-16    98.86
## 10         216           11 Version-13    98.86
## 11         219           11  Version-3    98.85
## 12         224           11  Version-2    98.82
## 13         224           11  Version-5    98.82
## 14         224           11 Version-12    98.82
## 15         224           11 Version-15    98.82
## 16         235            8  Version-8    98.78
## 17         259            9 Version-11    98.66
## 18         269            8  Version-1    98.61
## 19         337            7 Version-17    98.28
## 20         342            8  Version-7    98.25
```

```r
# According to this problem, I guess it is better to classify a user as an user
# that will not download the app by mistake than the ones that will do because
# if someone that the model classified as they would not download the app, in
# the worst case scenario they will do the download. However, if I classify an
# user as potential "good" user and they don't download and keep on causing
# more clicks is worse. In summary we want a model with high sensitivity and high
```

```r
# specificity, however it is better to gain in sensitivity, not losing too much
# specificity. Therefore, my choice is the model version 9, which achieves
# a very low rate of error for the first case mentioned above, while achieving
# a still high rate of error, but lower than most of the model versions created
# in this project. Moreover, it achieved a very good accuracy of 98.99%.

# Now, let's try to improve it a little bit further if possible.

################################ MODEL TUNNING #################################

# ##Predictors used in version 9

#app + channel + hour + userID_clicks_h + channel_clicks_h

# Let's create a few variations just to make sure we still have the best features

#1. is_attributed ~ app + channel + hour + userID_clicks_h + channel_clicks_h **BASE
#2. is_attributed ~ app + channel + hour + userID_clicks_h
#3. is_attributed ~ app + channel + hour + channel_clicks_h
#4. is_attributed ~ app + channel + userID_clicks_h + channel_clicks_h
#5. is_attributed ~ app + channel + userID_clicks_h
#6. is_attributed ~ app + channel + channel_clicks_h
#7. is_attributed ~ app + hour + userID_clicks_h + channel_clicks_h
#8. is_attributed ~ app + hour + userID_clicks_h
#9. is_attributed ~ app + hour + channel_clicks_h
#10. is_attributed ~ channel + hour + userID_clicks_h + channel_clicks_h
#11. is_attributed ~ channel + hour + userID_clicks_h
#12. is_attributed ~ channel + hour + channel_clicks_h
#13. is_attributed ~ hour + userID_clicks_h + channel_clicks_h
#14. is_attributed ~ hour + userID_clicks_h
#15. is_attributed ~ app + userID_clicks_h + channel_clicks_h
#16. is_attributed ~ hour + channel_clicks_h
#17. is_attributed ~ app + userID_clicks_h
#18. is_attributed ~ app + channel_clicks_h
#19. is_attributed ~ channel + userID_clicks_h + channel_clicks_h
#10. is_attributed ~ channel + userID_clicks_h
#21. is_attributed ~ channel + channel_clicks_h

# Creating a list to hold all the combinations of predictors (21) for this step
predictors_list2 = list(
  "is_attributed ~ app + channel + hour + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ app + channel + hour + userID_clicks_h",
  "is_attributed ~ app + channel + hour + channel_clicks_h",
  "is_attributed ~ app + channel + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ app + channel + userID_clicks_h",
  "is_attributed ~ app + channel + channel_clicks_h",
  "is_attributed ~ app + hour + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ app + hour + userID_clicks_h",
  "is_attributed ~ app + hour + channel_clicks_h",
  "is_attributed ~ channel + hour + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ channel + hour + userID_clicks_h",
  "is_attributed ~ channel + hour + channel_clicks_h",
  "is_attributed ~ hour + userID_clicks_h + channel_clicks_h",
```

```r
  "is_attributed ~ hour + userID_clicks_h",
  "is_attributed ~ hour + channel_clicks_h",
  "is_attributed ~ app + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ app + userID_clicks_h",
  "is_attributed ~ app + channel_clicks_h",
  "is_attributed ~ channel + userID_clicks_h + channel_clicks_h",
  "is_attributed ~ channel + userID_clicks_h",
  "is_attributed ~ channel + channel_clicks_h"
)

# Creating four lists to hold the results
nb_model_list2 = list()
nb_pred_list2 = list()
nb_accuracy_list2 = list()
nb_tables_list2 = list()

# Iterating throught the predictors_list to generate models
for (i in 1:length(predictors_list2)) {

  # Model creation and filling in the nb_model_list
  nb_model_list2[[i]] = naiveBayes(as.formula(predictors_list2[[i]]),
                                   data = train)

  # Prediction using test data and filling in the nb_pred_list
  nb_pred_list2[[i]] = predict(nb_model_list2[[i]], test[,-5])

  # Calculating accuracy for model and filling in the nb_accuracy_list
  nb_accuracy_list2[[i]] = sum(obs_oL == nb_pred_list2[[i]]) / length(obs_oL)

  # Filling the list nb_tables_list with the confusion matrix of results
  nb_tables_list2[[i]] = table(obs_oL, nb_pred_list2[[i]])
  print(paste("Step", i, "of 20!!"))
}
```

```
## [1] "Step 1 of 20!!"
## [1] "Step 2 of 20!!"
## [1] "Step 3 of 20!!"
## [1] "Step 4 of 20!!"
## [1] "Step 5 of 20!!"
## [1] "Step 6 of 20!!"
## [1] "Step 7 of 20!!"
## [1] "Step 8 of 20!!"
## [1] "Step 9 of 20!!"
## [1] "Step 10 of 20!!"
## [1] "Step 11 of 20!!"
## [1] "Step 12 of 20!!"
## [1] "Step 13 of 20!!"
## [1] "Step 14 of 20!!"
## [1] "Step 15 of 20!!"
## [1] "Step 16 of 20!!"
## [1] "Step 17 of 20!!"
## [1] "Step 18 of 20!!"
## [1] "Step 19 of 20!!"
## [1] "Step 20 of 20!!"
```

```
## [1] "Step 21 of 20!!"
```

```r
# Creating two new lists to hold the errors (typeI and typeII)
error2_typeI = list()
error2_typeII = list()

# Iterating through nb_tables_list and filling the errors lists
for (i in 1:length(nb_tables_list2)) {
  error2_typeI[[i]] = round(nb_tables_list2[[i]][3])
  error2_typeII[[i]] = round(nb_tables_list2[[i]][2])
}

# Creating a data.frame with errors, model versions and accuracies.
errors2 = data.frame(error2_typeI = unlist(error2_typeI),
                    error2_typeII = unlist(error2_typeII),
                    model = sapply(c(1:21), function(x) {paste0("Version-",x)}),
                    accuracy = sapply(c(1:21), function(x) {round(nb_accuracy_list2[[x]] * 100, 2)}))

# Sorting the data.frame based on error_typeI and error_typeII
errors2 %>%
  arrange(error2_typeI, error2_typeII)
```

```
##    error2_typeI error2_typeII      model accuracy
## 1             0            45 Version-13    99.77
## 2             0            45 Version-14    99.77
## 3             0            45 Version-15    99.77
## 4             2            41 Version-11    99.78
## 5             2            41 Version-20    99.78
## 6             3            41 Version-12    99.78
## 7             3            41 Version-19    99.78
## 8             3            41 Version-21    99.78
## 9             4            41 Version-10    99.77
## 10            6            38  Version-8    99.78
## 11            9            37 Version-17    99.77
## 12           18            36 Version-18    99.73
## 13           19            36  Version-7    99.72
## 14           19            36  Version-9    99.72
## 15           20            36 Version-16    99.72
## 16          166            12  Version-5    99.11
## 17          168            12  Version-2    99.10
## 18          189            10  Version-3    99.00
## 19          189            10  Version-6    99.00
## 20          192             9  Version-1    98.99
## 21          192            10  Version-4    98.99
```

```r
# According to the results, the best option for our problem would be the version 1,
# which, in fact, exactly the same as version 9 from previous round.

# Therefore, this was the chosen final version for our problem.

# So, the chosen model, based on the accuracy evaluation is:
print(errors[9,])
```

```
##   error_typeI error_typeII     model accuracy
## 9         192            9 Version-9    98.99
```

```r
# Let's print out some evaluation parameters for the chosen model
confusionMatrix(nb_pred_list[[9]], obs_oL)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 19762     9
##          2   192    36
##
##                Accuracy : 0.9899
##                  95% CI : (0.9885, 0.9913)
##     No Information Rate : 0.9977
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.261
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9904
##             Specificity : 0.8000
##          Pos Pred Value : 0.9995
##          Neg Pred Value : 0.1579
##              Prevalence : 0.9977
##          Detection Rate : 0.9881
##    Detection Prevalence : 0.9886
##       Balanced Accuracy : 0.8952
##
##        'Positive' Class : 1
##
```

```r
################################## THE END #####################################
```