Predictions whether a user will download an app after clicking a mobile app advertisement - FINAL MODEL

Diogo F. dos Santos

August 10th, 2020

% !TEX encoding = UTF-8 Unicode

PART TWO This script got the main tidying lines of part one to tidy the full training dataset, nominated train.csv.

```
# Removes all existing objects and packages from the current workspace
rm(list = ls())
# Working directory
setwd("~/Documents/learning_Data_Science/R_learnings/Project_1_in_R")
# getwd()

# Packages
library(dplyr)
```

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

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

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

```
library(data.table)
```

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

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

```
library(caret)
```

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

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

```
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(DMwR)
```

```
## Loading required package: grid

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(knitr)
library(rmarkdown)
```

```
# Number of rows in the train dataset
# The train dataset named train.csv can be found on the web site
# https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data
n_rows <- fread(file = 'train.csv', header = T, select = 'is_attributed')
n_rows <- nrow(n_rows)
n_rows    # 184.903.890 rows
```

```
## [1] 184903890
```

```
gc()
```

```
##            used  (Mb) gc trigger  (Mb)  max used  (Mb)
## Ncells  2224175 118.8    4021857 214.8   3670593 196.1
## Vcells 12990512  99.2  102896125 785.1 105482783 804.8
```

```
# Calculating the number of batches
for (i in c(15:100)) {
  if (n_rows%%i == 0) {
    print(c(i, n_rows/i))
  }
}             # 15 seems better for my computer capacity
```

```
## [1]       15 12326926
## [1]      30 6163463
## [1]      73 2532930
```

```
rm(i)

# Batches
n = 15
train_set <- data.frame(is_attributed = c(),
                        app = c(),
                        channel = c(),
                        repetitions_fac = c(),
                        app_fac = c())
for (i in c(0:(n-1))) {
  if (i == 0) {
```

```
    # The train dataset named train.csv can be found on the web site
    # https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data
    train <- fread(file = 'train.csv', header = T,
                   skip = n_rows/n*i, nrows = n_rows/n,
                   select = c('is_attributed', 'ip', 'app', 'channel'))
                  } else {
    train <- fread(file = 'train.csv', header = F,
                    skip = n_rows/n*i, nrows = n_rows/n,
                    select = c(8,1,2,5))
    names(train) <- c('is_attributed', 'ip', 'app', 'channel')
  }

  # ip feature
  # Repeated ips in order
  n_dupl_ips <- train %>%
    count(ip, wt = n(), name = 'repetitions') %>%
    arrange(desc(repetitions))

  # Number of duplicate ips column
  train <- left_join(train, n_dupl_ips, by = 'ip')
  train$ip <- NULL

  # repetitions classes
  train$repetitions_fac <- cut(train$repetitions,
                               breaks = c(0,5,nrow(train)),
                               labels = c(1, 2))
  train$repetitions <- NULL

  # app classes
  train$app_fac <- cut(train$app,
                       breaks = c(0, 3, 12, 18, nrow(train)),
                       right = F, labels = c(1, 2, 3, 4))

  # is_attributed classes
  train <- train %>%
    mutate(is_attributed = factor(is_attributed, levels = c(1,0)))
  head(train_set)

  # Balancing the target class
  train <- SMOTE(is_attributed ~ ., data  = train)

  # Binding the train dataset
  train_set <- rbind(train_set, train)

  rm(n_dupl_ips, train)
  gc()
  print(i)
}
```

```
## [1] 0
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

```
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
```

```
# training data set dimension
dim(train_set)
```

```
## [1] 3197922       5
```

```
# Number of downloads, indicated by "1"
table(train_set$is_attributed)
```

```
##
##       1       0
## 1370538 1827384
```

```
# Features types
str(train_set)
```

```
## Classes 'data.table' and 'data.frame':   3197922 obs. of  5 variables:
##  $ is_attributed  : Factor w/ 2 levels "1","0": 2 2 2 2 2 2 2 2 2 2 ...
##  $ app            : num  26 15 3 9 12 15 18 15 2 9 ...
##  $ channel        : num  266 140 452 334 497 245 121 140 205 258 ...
##  $ repetitions_fac: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ app_fac        : Factor w/ 4 levels "1","2","3","4": 4 3 2 2 3 3 4 3 1 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

PART THREE In this part, the tidying training dataset was taken with the best model acquired in part one to train the model, but the number of the trees of the random forest model was reduced due to my notebook capacity.

```
# Random forest model
model15 <- randomForest(is_attributed ~ repetitions_fac * app +
                          channel * app_fac,
                        data = train_set,
                        ntree = 10,
                        nodesize = 1)
```

PART FOUR In this part, the trained model was applied to the provided test dataset, test.csv. Afterward, the predicted results were matched with the click_id to produce the submission file.

The test dataset is similar to the training dataset, with the following differences: click_id: reference for making predictions is_attributed: not included

```
# Loading the test file
# The test dataset named test.csv can be found on the web site
# https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data
test_set <- fread(file = 'test.csv', header = T,
                  select = c('click_id', 'ip', 'app', 'channel'))

# ip feature
```

```
# Repeated ips in order
n_dupl_ips <- test_set %>%
  count(ip, wt = n(), name = 'repetitions') %>%
  arrange(desc(repetitions))

# Number of duplicate ips column
test_set <- left_join(test_set, n_dupl_ips, by = 'ip')
test_set$ip <- NULL
rm(n_dupl_ips)

# repetitions classes
test_set$repetitions_fac <- cut(test_set$repetitions,
                                breaks = c(0,5,nrow(test_set)),
                                labels = c(1, 2))
test_set$repetitions <- NULL

# app classes
test_set$app_fac <- cut(test_set$app,
                        breaks = c(0, 3, 12, 18, nrow(test_set)),
                        right = F, labels = c(1, 2, 3, 4))
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5566151 297.3   13128697  701.2  13128697  701.2
## Vcells 99736171 761.0  547515494 4177.3 680144856 5189.1
```

Predictions of the machine learning model

```
# Predictions using the model 15s
predictions15 <- predict(model15, test_set, type = "prob")
head(predictions15)
```

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

The submission file with the calculated probabilities

```
# for the is_attributed variable
test_set_results <- data.frame(click_id = test_set$click_id,
                               is_attributed = predictions15[,1])
head(test_set_results)
```

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

```r
dim(test_set_results)
```

```
## [1] 18790469        2
```

```r
# Saving the submission file
# write.csv(x = test_set_results, file = 'submission_file.csv', row.names = F)

# Number yes (1) or no (0) is_attributed variable
table(round(test_set_results[,2]))
```

```
##
##        0        1
## 18346343   444126
```

Cleaning the house

rm(list = ls()) gc()

THE END