Frauds Detection

Vitor Marques

2020-12-25

```
# Data Dictionary
# ip: ip address of click.
# app: app id for marketing.
# device: device type id of user mobile phone (e.g., iphone 6 plus, iphone 7, huawei mate 7, etc.)
# os: os version id of user mobile phone
# channel: channel id of mobile ad publisher
# click_time: timestamp of click (UTC)
# attributed_time: if user download the app for after clicking an ad, this is the time of the app downl
# is_attributed: the target that is to be predicted, indicating the app was downloaded
### Note that ip, app, device, os, and channel are encoded.
# The test data is similar, with the following differences:
# click_id: reference for making predictions
# is_attributed: not included


# Loading library
#install.packages("markdown")
# install.packages('caret')
# install.packages('data.table')
# install.packages("corrplot")
# install.packages('e1071')
# install.packages('ROSE')
library(markdown)
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

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

```
## data.table 1.13.4 using 6 threads (see ?getDTthreads).  Latest news: r-datatable.com
```

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

```
## The following object is masked _by_ '.GlobalEnv':
##
##     .N
```

```
library(caret)
```

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

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

```
## Use suppressPackageStartupMessages() to eliminate package startup
## messages
```

```
library(e1071)
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
```

```
# Loading datasets

train <- fread("talkingdata-adtracking-fraud-detection/train.csv")
test <- fread("talkingdata-adtracking-fraud-detection/test.csv")
sample_subm <- fread("talkingdata-adtracking-fraud-detection/sample_submission.csv")

#Data Exploration
str(train)
```

```
## Classes 'data.table' and 'data.frame':   184903890 obs. of  8 variables:
##  $ ip             : int  83230 17357 35810 45745 161007 18787 103022 114221 165970 74544 ...
##  $ app            : int  3 3 3 14 3 3 3 3 3 64 ...
##  $ device         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ os             : int  13 19 13 13 13 16 23 19 13 22 ...
##  $ channel        : int  379 379 379 478 379 379 379 379 379 459 ...
##  $ click_time     : chr  "2017-11-06 14:32:21" "2017-11-06 14:33:34" "2017-11-06 14:34:12" "2017-11-0
##  $ attributed_time: chr  "" "" "" "" ...
##  $ is_attributed  : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(test)
```

```
## Classes 'data.table' and 'data.frame':   18790469 obs. of  7 variables:
##  $ click_id  : int  0 1 2 3 4 5 6 7 9 8 ...
##  $ ip        : int  5744 119901 72287 78477 123080 110769 12540 88637 14932 123701 ...
##  $ app       : int  9 9 21 15 12 18 3 27 18 12 ...
##  $ device    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ os        : int  3 3 19 13 13 13 1 19 10 53 ...
##  $ channel   : int  107 466 128 111 328 107 137 153 107 424 ...
##  $ click_time: chr  "2017-11-10 04:00:00" "2017-11-10 04:00:00" "2017-11-10 04:00:00" "2017-11-10 04
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(sample_subm)
```

```
## Classes 'data.table' and 'data.frame':   18790469 obs. of  2 variables:
##  $ click_id     : int  0 1 2 3 4 5 6 7 9 8 ...
```

```
##  $ is_attributed: int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
View(train)
View(test)
View(sample_subm)
#Table
prop.table(table(train$is_attributed))
```

```
##
##           0           1
## 0.997529279 0.002470721
```

```
#Data Split
set.seed(283)
indextrain <- sample(nrow(train), size = 0.0001 * nrow(train))
indextest <- sample(nrow(test), size = 0.001 * nrow(test))


df_train <- train[indextrain,]
str(df_train)
```

```
## Classes 'data.table' and 'data.frame':   18490 obs. of  8 variables:
##  $ ip             : int  8499 52052 73516 33738 33835 192625 26577 5314 77048 93963 ...
##  $ app            : int  12 2 12 7 12 1 26 8 25 12 ...
##  $ device         : int  1 1 1 1 1 1 1 2 1 1 ...
##  $ os             : int  19 19 19 19 19 14 6 11 13 32 ...
##  $ channel        : int  259 205 326 101 178 17 266 140 259 265 ...
##  $ click_time     : chr  "2017-11-08 23:39:05" "2017-11-08 01:03:56" "2017-11-06 18:21:22" "2017-11-
##  $ attributed_time: chr  "" "" "" "" ...
##  $ is_attributed  : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
df_test <- test[indextest,]
str(df_test)
```

```
## Classes 'data.table' and 'data.frame':   18790 obs. of  7 variables:
##  $ click_id  : int  16059230 17028288 10618832 14803412 11917152 12564969 4949169 16693217 10483662
##  $ ip        : int  56063 125222 60136 75422 112530 48062 119901 41186 14792 8694 ...
##  $ app       : int  9 21 17 3 21 18 2 3 9 9 ...
##  $ device    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ os        : int  1 20 22 13 13 18 17 19 19 19 ...
##  $ channel   : int  466 232 128 489 232 379 469 409 442 107 ...
##  $ click_time: chr  "2017-11-10 14:09:30" "2017-11-10 14:26:51" "2017-11-10 10:27:42" "2017-11-10 13
##  - attr(*, ".internal.selfref")=<externalptr>
```

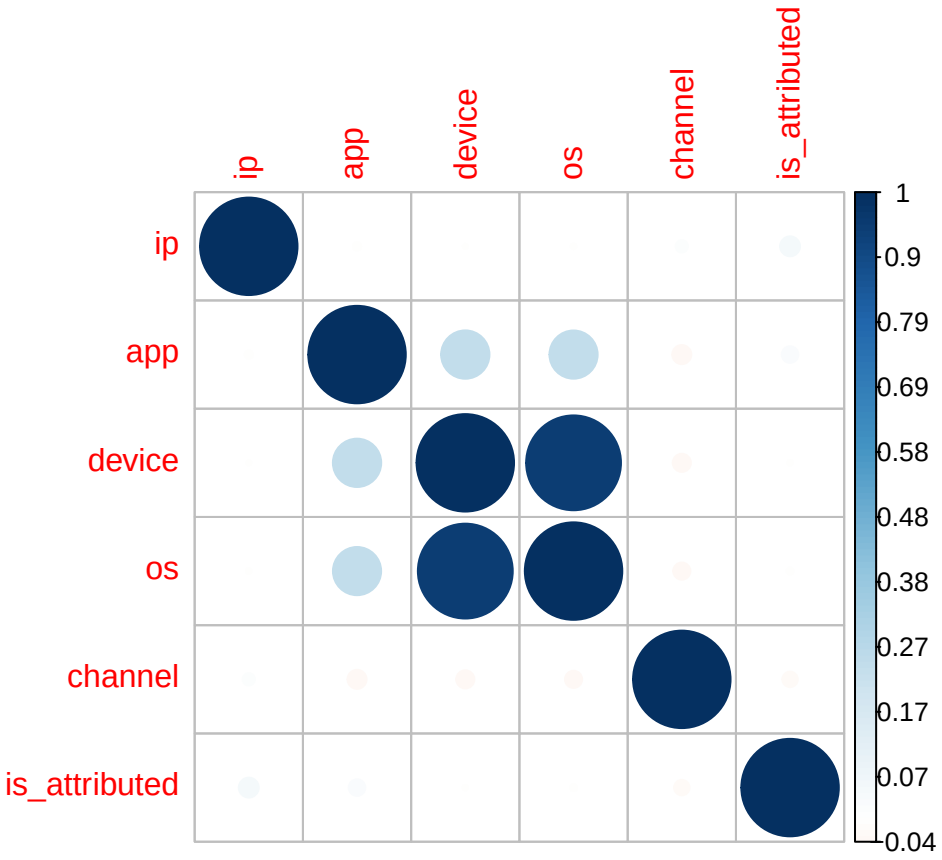
```
#Removing click_time & attributed_time
df_train$click_time <- NULL
df_test$click_time <- NULL
df_train$attributed_time <- NULL

#Correlation between features
cor(df_train)
```

```
##                         ip          app       device           os
## ip             1.000000000 -0.006712387 -0.002422125 -0.003096289
## app           -0.006712387  1.000000000  0.245671152  0.244206103
```

```
## device         -0.002422125  0.245671152  1.000000000  0.947166050
## os             -0.003096289  0.244206103  0.947166050  1.000000000
## channel         0.015479754 -0.037906680 -0.034395220 -0.030784488
## is_attributed   0.041625783  0.028760646 -0.002847128 -0.004673887
##                    channel is_attributed
## ip              0.01547975   0.041625783
## app            -0.03790668   0.028760646
## device         -0.03439522  -0.002847128
## os             -0.03078449  -0.004673887
## channel         1.00000000  -0.024842316
## is_attributed -0.02484232   1.000000000
```

```
corrplot(cor(df_train), is.corr = F )
```

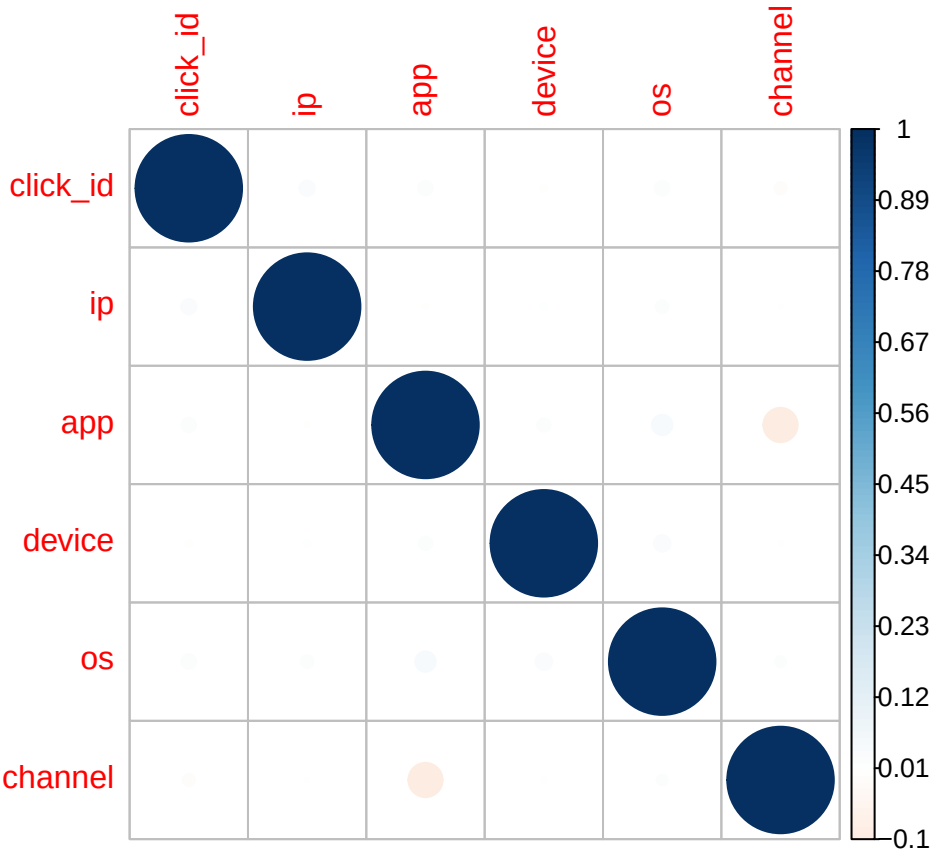



```
cor(df_test)
```

```
##              click_id            ip          app       device          os
## click_id  1.000000000  0.0203604717  0.018241131 -0.003813941 0.01911509
## ip        0.020360472  1.0000000000 -0.001811038  0.003465447 0.01420335
## app       0.018241131 -0.0018110383  1.000000000  0.015983221 0.03639227
## device   -0.003813941  0.0034654471  0.015983221  1.000000000 0.02381966
## os        0.019115091  0.0142033472  0.036392267  0.023819659 1.00000000
## channel  -0.012892077  0.0007992651 -0.103422068  0.002011807 0.01027796
##                channel
## click_id -0.0128920775
## ip        0.0007992651
## app      -0.1034220681
```

```
## device    0.0020118068
## os        0.0102779587
## channel   1.0000000000
```

```
corrplot(cor(df_test), is.corr = F)
```




```
#Removing IP & click_id
df_train$ip <- NULL
df_test$ip <- NULL
df_test$click_id <- NULL


str(df_test)
```

```
## Classes 'data.table' and 'data.frame':   18790 obs. of  4 variables:
##  $ app    : int  9 21 17 3 21 18 2 3 9 9 ...
##  $ device : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ os     : int  1 20 22 13 13 18 17 19 19 19 ...
##  $ channel: int  466 232 128 489 232 379 469 409 442 107 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(df_train)
```

```
## Classes 'data.table' and 'data.frame':   18490 obs. of  5 variables:
##  $ app          : int  12 2 12 7 12 1 26 8 25 12 ...
##  $ device       : int  1 1 1 1 1 1 1 2 1 1 ...
##  $ os           : int  19 19 19 19 19 14 6 11 13 32 ...
##  $ channel      : int  259 205 326 101 178 17 266 140 259 265 ...
```

```
##  $ is_attributed: int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
#Table of target feature
table(df_train$is_attributed)
```

```
##
##     0     1
## 18445    45
```

```
#Applying undersampling.
# Using Under Both or Over
df_train1 <- ovun.sample(is_attributed ~., data = df_train, method = 'both', N = 45000)$data
View(df_train1)
table(df_train1$is_attributed)
```

```
##
##     0     1
## 22665 22335
```

```
#Function for automating variable categorization
catfun <- function(dataset, features){
  for (feature in features) {
    dataset[[feature]] <- as.factor(dataset[[feature]])
  }
  return(dataset)
}

#Function for  Normalization
catnorm <- function(dataset, features){
  for(feature in features){
    dataset[[feature]] <- scale(dataset[[feature]], center = T, scale = T)
  }
  return(dataset)
}

#Features
testnorm <- c('channel', 'os', 'device', 'app')
trainnorm <- c('app', 'device', 'os', 'channel')
traincat <- c('is_attributed')

# Categorization
df_train <- catfun(df_train1, traincat)

str(df_train)
```

```
## 'data.frame':    45000 obs. of  5 variables:
##  $ app          : int  21 14 1 15 3 3 32 3 3 13 ...
##  $ device       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ os           : int  25 13 6 40 22 8 6 13 13 13 ...
##  $ channel      : int  128 467 377 480 205 211 376 280 130 477 ...
##  $ is_attributed: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Normalization
df_train <- catnorm(df_train, trainnorm)
df_test <- catnorm(df_test, testnorm)
```

```r
str(df_train)
```

```
## 'data.frame':    45000 obs. of  5 variables:
##  $ app          : num [1:45000, 1] 0.289 -0.187 -1.072 -0.119 -0.936 ...
##   ..- attr(*, "scaled:center")= num 16.7
##   ..- attr(*, "scaled:scale")= num 14.7
##  $ device       : num [1:45000, 1] -0.079 -0.079 -0.079 -0.079 -0.079 ...
##   ..- attr(*, "scaled:center")= num 17
##   ..- attr(*, "scaled:scale")= num 203
##  $ os           : num [1:45000, 1] 0.0984 -0.1765 -0.3368 0.4419 0.0297 ...
##   ..- attr(*, "scaled:center")= num 20.7
##   ..- attr(*, "scaled:scale")= num 43.7
##  $ channel      : num [1:45000, 1] -0.799 1.721 1.052 1.818 -0.226 ...
##   ..- attr(*, "scaled:center")= num 235
##   ..- attr(*, "scaled:scale")= num 135
##  $ is_attributed: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
str(df_test)
```

```
## Classes 'data.table' and 'data.frame':   18790 obs. of  4 variables:
##  $ app    : num [1:18790, 1] -0.279 0.782 0.428 -0.81 0.782 ...
##   ..- attr(*, "scaled:center")= num 12.2
##   ..- attr(*, "scaled:scale")= num 11.3
##  $ device : num [1:18790, 1] -0.0251 -0.0251 -0.0251 -0.0251 -0.0251 ...
##   ..- attr(*, "scaled:center")= num 1.75
##   ..- attr(*, "scaled:scale")= num 29.7
##  $ os     : num [1:18790, 1] -1.587 0.116 0.296 -0.511 -0.511 ...
##   ..- attr(*, "scaled:center")= num 18.7
##   ..- attr(*, "scaled:scale")= num 11.2
##  $ channel: num [1:18790, 1] 1.484 -0.242 -1.009 1.654 -0.242 ...
##   ..- attr(*, "scaled:center")= num 265
##   ..- attr(*, "scaled:scale")= num 136
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
#Model training
modelv1 <- train(is_attributed ~., data = df_train, method = 'rf')
modelv1
```

```
## Random Forest
##
## 45000 samples
##     4 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 45000, 45000, 45000, 45000, 45000, 45000, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9903396  0.9806805
##   3     0.9908568  0.9817147
##   4     0.9909341  0.9818692
##
## Accuracy was used to select the optimal model using the largest value.
```

```
## The final value used for the model was mtry = 4.
```

```
predc <- predict(modelv1, df_test)
View(predc)

#Submission
View(predc)
```