Prevendo Cliques Fraudulentos.R

Luiz Carlos Castro Cunha Junior

2019-11-08

```
# O objetivo do desenvolvimento do projeto consiste em identificar se um clique é fraudulento ou não.

#Configurando o working directory
setwd("C:\\FCD\\BigDataAnalytics-R-Azure\\Projeto")

#Carregando as biblíotecas
library(tidyverse)
```

```
## -- Attaching packages -----------------

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts --------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rmarkdown)


# Carregando uma amostra de 4.000.000 dados do dataset
dt<- read_csv('datasets/train.csv',n_max  = 4000000)
```

```
## Parsed with column specification:
## cols(
##   ip = col_double(),
##   app = col_double(),
```

```
##   device = col_double(),
##   os = col_double(),
##   channel = col_double(),
##   click_time = col_datetime(format = ""),
##   attributed_time = col_datetime(format = ""),
##   is_attributed = col_double()
## )
```

```r
# Verificando os valores NA do conjunto de dados
colSums(is.na(dt))
```

```
##              ip             app          device              os
##               0               0               0               0
##         channel      click_time attributed_time   is_attributed
##               0               0         3993058               0
```

```r
colSums(is.na(dt))
```

```
##              ip             app          device              os
##               0               0               0               0
##         channel      click_time attributed_time   is_attributed
##               0               0         3993058               0
```

```r
# Analisando a distribuição dos valores da variável target
table(dt$is_attributed)
```

```
##
##       0       1
## 3993058    6942
```

```r
# Podemos observar que há 3993058 valores NA na coluna attributed_time e 3993058 valores falsos(0) na c
# o que está completamente certo, pois quando is_attributed é verdadeira (1) aparece o tempo total do d
# E quando ele é 0(falso) é atribuído NA, então a coluna attributed_time será removida.
dt$attributed_time <- NULL

# Visualizando os dados
head(dt)
```

```
## # A tibble: 6 x 7
##       ip   app device    os channel click_time          is_attributed
##    <dbl> <dbl>  <dbl> <dbl>   <dbl> <dttm>                      <dbl>
## 1  83230     3      1    13     379 2017-11-06 14:32:21             0
## 2  17357     3      1    19     379 2017-11-06 14:33:34             0
## 3  35810     3      1    13     379 2017-11-06 14:34:12             0
## 4  45745    14      1    13     478 2017-11-06 14:34:52             0
## 5 161007     3      1    13     379 2017-11-06 14:35:08             0
## 6  18787     3      1    16     379 2017-11-06 14:36:26             0
```

```r
# Criando as colunas dia, mes e hora através da coluna click_time.
# Não será adicionadoa coluna ano, pois os dados são do ano de 2017
dt <- dt %>%
  mutate(dia = as.factor(day(click_time)),
         mes = month(click_time),
         hora = hour(click_time)
)


#Visualizando os dados após a mudança
```

```r
head(dt)
```

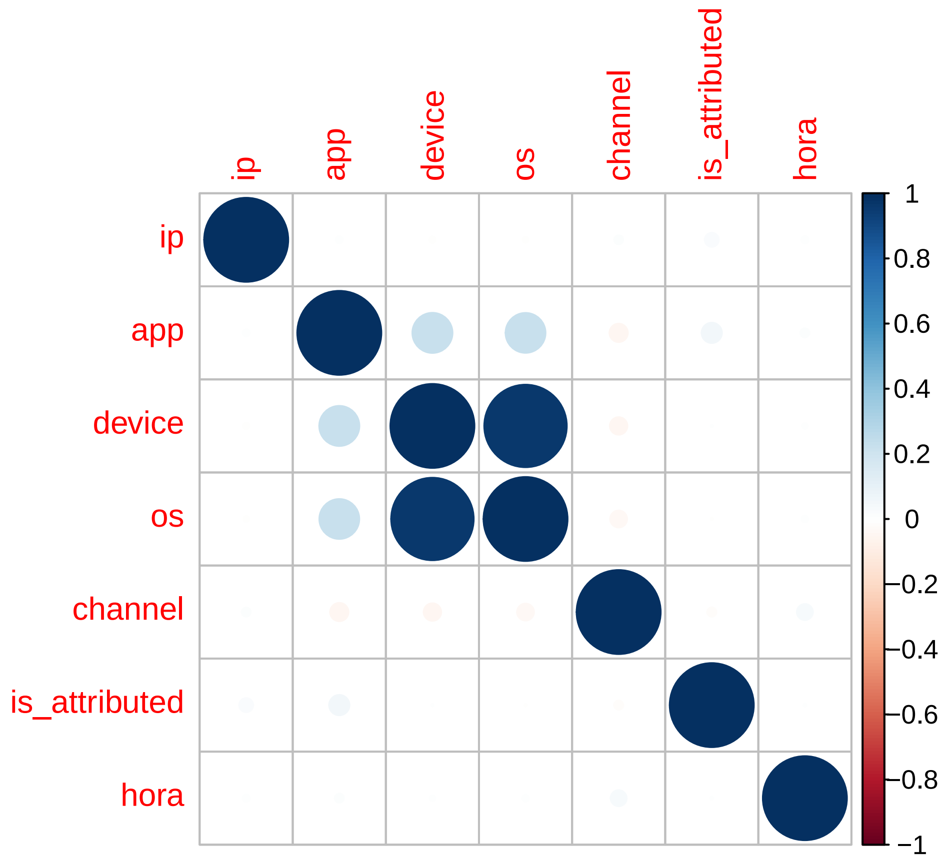
```
## # A tibble: 6 x 10
##       ip   app device    os channel click_time          is_attributed dia
##    <dbl> <dbl>  <dbl> <dbl>   <dbl> <dttm>                      <dbl> <fct>
## 1  83230     3      1    13     379 2017-11-06 14:32:21             0 6
## 2  17357     3      1    19     379 2017-11-06 14:33:34             0 6
## 3  35810     3      1    13     379 2017-11-06 14:34:12             0 6
## 4  45745    14      1    13     478 2017-11-06 14:34:52             0 6
## 5 161007     3      1    13     379 2017-11-06 14:35:08             0 6
## 6  18787     3      1    16     379 2017-11-06 14:36:26             0 6
## # ... with 2 more variables: mes <dbl>, hora <int>
```

```r
# Como as colunas foram preenchidas corretamente, não há sentido em manter a coluna click_time, então i
dt$click_time <- NULL

#Verificando os valores únicos dos dados
f_unicos <- function(x){
  length(unique(x))
}

lapply(dt, f_unicos)
```

```
## $ip
## [1] 55205
##
## $app
## [1] 289
##
## $device
## [1] 568
##
## $os
## [1] 229
##
## $channel
## [1] 159
##
## $is_attributed
## [1] 2
##
## $dia
## [1] 1
##
## $mes
## [1] 1
##
## $hora
## [1] 5
```

```r
# Após aplicar a função nos dados percebemos que as colunas dia e mês possuem apenas 1 valor único,
# que são respectivamente: 6(dia) e 11(mês).

# Removendo as duas colunas
dt$dia <- NULL
```

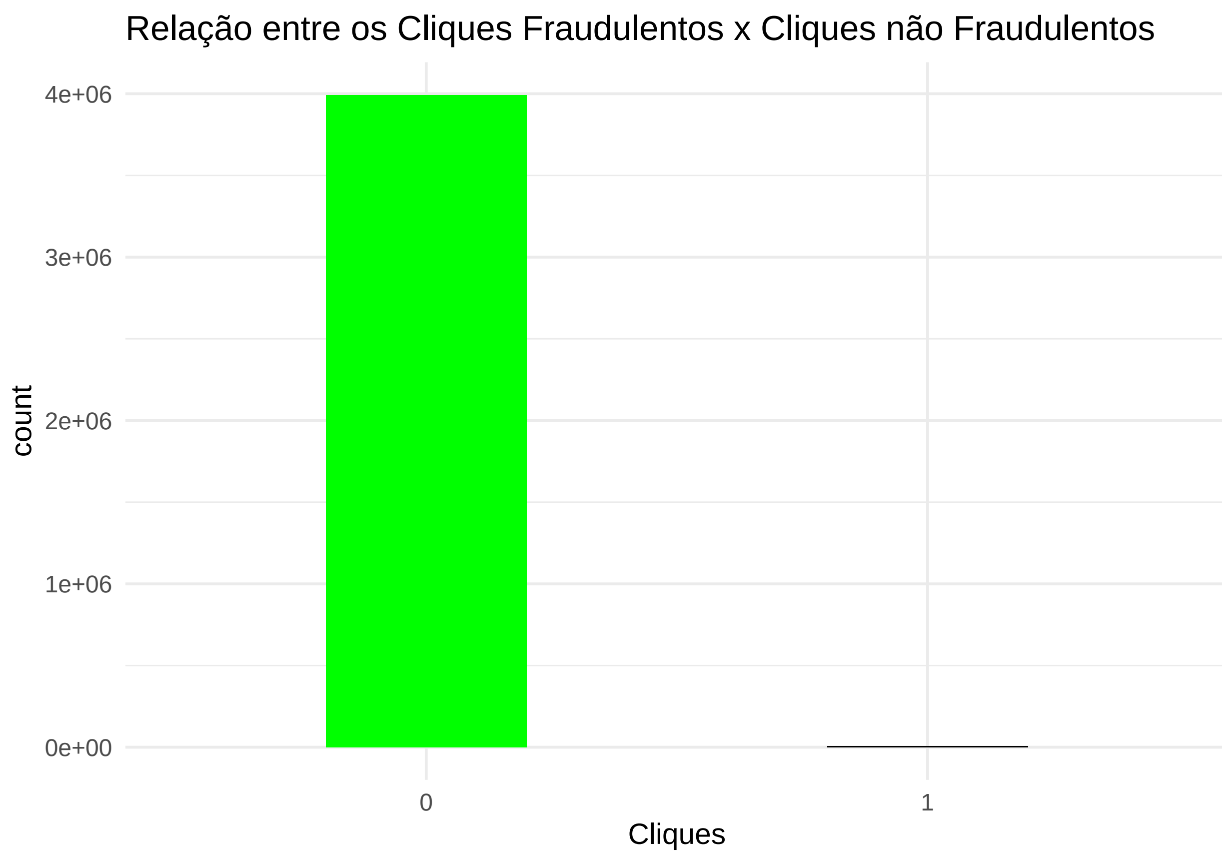
```r
dt$mes <- NULL

#Visualizando a correlação entre as variáveis
corrplot(cor(dt))
```



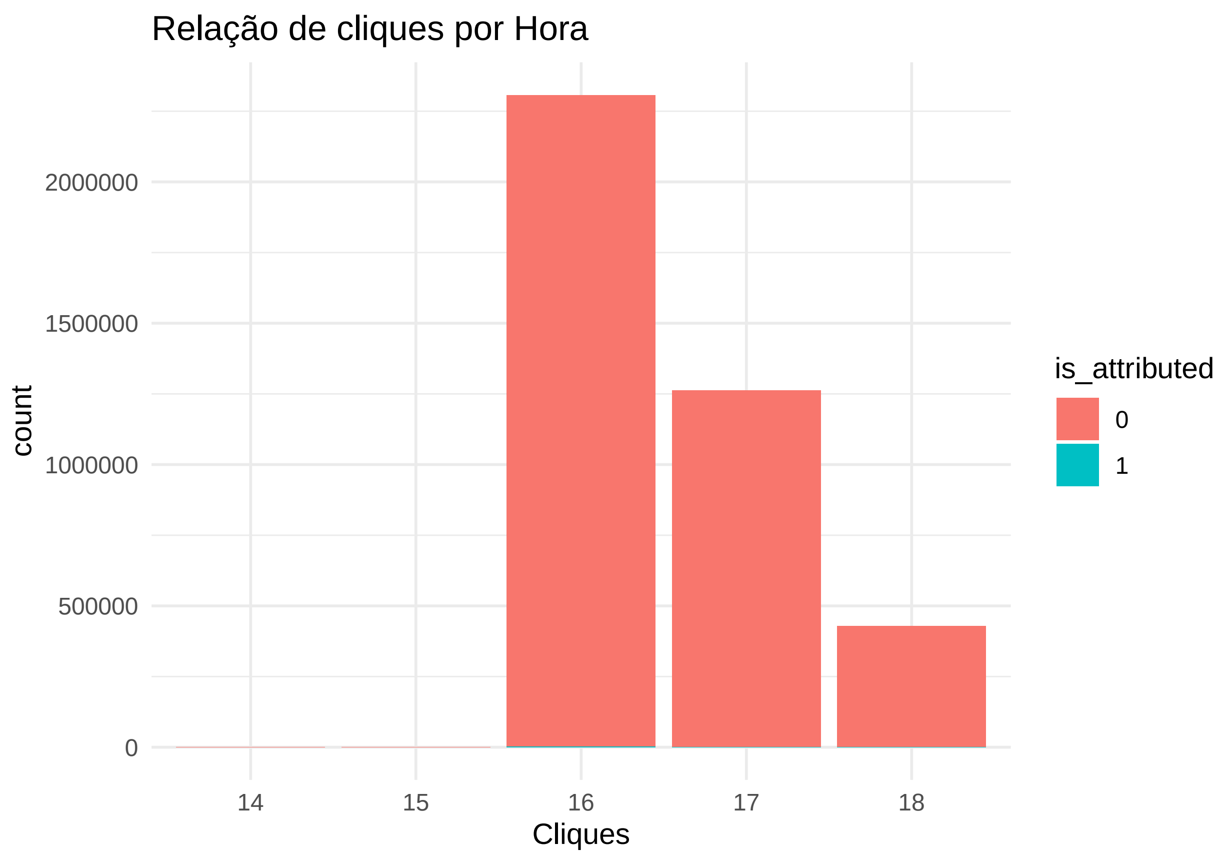

```r
#Convertendo as variáveis hora e is_attributed  para o tipo fator

dt$is_attributed <- factor(dt$is_attributed)
dt$hora <- factor(dt$hora)
str(dt)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 4000000 obs. of  7 variables:
##  $ ip           : num  83230 17357 35810 45745 161007 ...
##  $ app          : num  3 3 3 14 3 3 3 3 3 64 ...
##  $ device       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ os           : num  13 19 13 13 13 16 23 19 13 22 ...
##  $ channel      : num  379 379 379 478 379 379 379 379 379 459 ...
##  $ is_attributed: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ hora         : Factor w/ 5 levels "14","15","16",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
#Visualizando o total de cliques
ggplot(data = dt, aes(x = is_attributed))+
  geom_bar(stat = 'count', width = 0.4, fill = c('green', 'black'))+
  labs(title = "Relação entre os Cliques Fraudulentos x Cliques não Fraudulentos", x = 'Cliques')+
  theme_minimal()
```




```r
#Visualizando o total de cliques por hora
ggplot(data = dt, aes(x = hora, fill=  is_attributed))+
  geom_bar()+
  labs(title = 'Relação de cliques por Hora',x = "Cliques")+
  theme_minimal()
```




```r
#Separando os dados em treino e teste
library(caret)
set.seed(5000)
split <- createDataPartition(y = dt$is_attributed, p = 0.70, list = F)

# Criando dados de treino e de teste
dados_treino <- dt[split,]
dados_teste <- dt[-split,]

str(dados_treino)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2800001 obs. of  7 variables:
##  $ ip           : num  83230 17357 45745 18787 103022 ...
##  $ app          : num  3 3 14 3 3 3 3 3 3 3 ...
##  $ device       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ os           : num  13 19 13 16 23 19 13 25 18 13 ...
##  $ channel      : num  379 379 478 379 379 379 379 379 379 379 ...
##  $ is_attributed: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ hora         : Factor w/ 5 levels "14","15","16",..: 1 1 1 1 1 1 1 1 1 1 ...
```

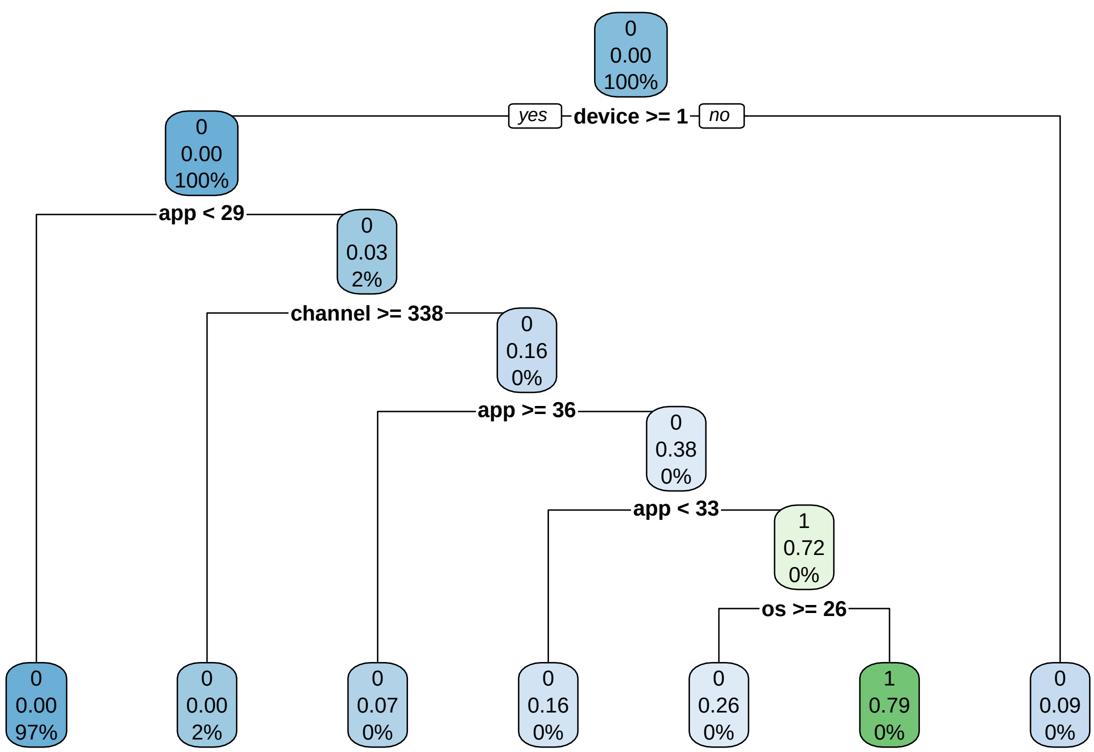
```r
#Verificando a proporção dos dados
round(prop.table(table(dados_teste$is_attributed)) * 100, 2)
```

```
##
##     0     1
## 99.83  0.17
```

```r
round(prop.table(table(dados_treino$is_attributed)) * 100, 2)
```

```
##
##     0     1
## 99.83  0.17
```

```r
#Utilizando o algoritmo de árvores de decisão
library(rpart)
set.seed(1234)

modelo <- rpart(is_attributed ~., data = dados_treino, method = "class")

#Criando um plot do modelo de árvores de decisão
library(rpart)
library(rpart.plot)
rpart.plot(modelo)
```




```r
#Fazendo previsões com o modelo
previsoes <- predict(modelo, dados_teste, type = 'class')

#Verificando o desempenho do modelo
round(prop.table(table(previsoes == dados_teste$is_attributed)) * 100, 2)
```

```
##
## FALSE  TRUE
##  0.15 99.85
```

```r
summary(modelo)
```

```
## Call:
## rpart(formula = is_attributed ~ ., data = dados_treino, method = "class")
##   n= 2800001
##
##           CP nsplit rel error    xerror       xstd
## 1 0.02423868      0 1.0000000 1.0000000 0.01433193
## 2 0.01851852      5 0.8788066 0.8788066 0.01343684
## 3 0.01000000      6 0.8602881 0.8602881 0.01329472
##
## Variable importance
##     app channel  device      os
##      50      36       7       7
##
## Node number 1: 2800001 observations,    complexity param=0.02423868
##   predicted class=0  expected loss=0.001735714  P(node) =1
##     class counts: 2.79514e+06  4860
##    probabilities: 0.998 0.002
##   left son=2 (2788722 obs) right son=3 (11279 obs)
##   Primary splits:
##       device  < 0.5    to the right, improve=164.56640, (0 missing)
##       app     < 28.5   to the left,  improve=125.94260, (0 missing)
##       os      < 0.5    to the right, improve= 43.61039, (0 missing)
##       channel < 103    to the right, improve= 27.56588, (0 missing)
##       ip      < 126551 to the left,  improve= 12.33873, (0 missing)
##   Surrogate splits:
##       os < 0.5    to the right, agree=0.997, adj=0.331, (0 split)
##
## Node number 2: 2788722 observations,    complexity param=0.02423868
##   predicted class=0  expected loss=0.00139096  P(node) =0.9959718
##     class counts: 2.78484e+06  3879
##    probabilities: 0.999 0.001
##   left son=4 (2725249 obs) right son=5 (63473 obs)
##   Primary splits:
##       app     < 28.5   to the left,  improve=121.064900, (0 missing)
##       os      < 0.5    to the right, improve= 36.936340, (0 missing)
##       channel < 103    to the right, improve= 25.688770, (0 missing)
##       device  < 3      to the left,  improve= 17.077610, (0 missing)
##       ip      < 126551 to the left,  improve=  7.077846, (0 missing)
##   Surrogate splits:
##       channel < 1.5    to the right, agree=0.977, adj=0, (0 split)
##       device  < 3470.5 to the left,  agree=0.977, adj=0, (0 split)
##
## Node number 3: 11279 observations
##   predicted class=0  expected loss=0.0869758  P(node) =0.004028213
##     class counts: 10298   981
##    probabilities: 0.913 0.087
##
## Node number 4: 2725249 observations
##   predicted class=0  expected loss=0.0006799379  P(node) =0.9733029
##     class counts: 2.7234e+06  1853
##    probabilities: 0.999 0.001
##
```

```
## Node number 5: 63473 observations,    complexity param=0.02423868
##   predicted class=0  expected loss=0.03191908  P(node) =0.02266892
##     class counts: 61447  2026
##    probabilities: 0.968 0.032
##   left son=10 (51414 obs) right son=11 (12059 obs)
##   Primary splits:
##       channel < 338    to the right, improve=465.05400, (0 missing)
##       app     < 35.5   to the right, improve=302.93710, (0 missing)
##       ip      < 126549 to the left,  improve= 44.34537, (0 missing)
##       os      < 22.5   to the right, improve= 26.30662, (0 missing)
##       device  < 2871.5 to the right, improve= 18.65924, (0 missing)
##   Surrogate splits:
##       app    < 101    to the left,  agree=0.830, adj=0.106, (0 split)
##       os     < 0.5    to the right, agree=0.812, adj=0.012, (0 split)
##       device < 3032.5 to the left,  agree=0.810, adj=0.001, (0 split)
##       ip     < 212759 to the left,  agree=0.810, adj=0.000, (0 split)
##
## Node number 10: 51414 observations
##   predicted class=0  expected loss=0.002606294  P(node) =0.01836214
##     class counts: 51280   134
##    probabilities: 0.997 0.003
##
## Node number 11: 12059 observations,    complexity param=0.02423868
##   predicted class=0  expected loss=0.1568953  P(node) =0.004306784
##     class counts: 10167  1892
##    probabilities: 0.843 0.157
##   left son=22 (8717 obs) right son=23 (3342 obs)
##   Primary splits:
##       app     < 35.5   to the right, improve=480.270100, (0 missing)
##       channel < 273    to the left,  improve=292.706400, (0 missing)
##       ip      < 126518 to the left,  improve=149.620100, (0 missing)
##       os      < 25.5   to the right, improve= 34.255030, (0 missing)
##       device  < 5      to the right, improve=  7.224918, (0 missing)
##   Surrogate splits:
##       channel < 206.5  to the left,  agree=0.789, adj=0.24, (0 split)
##
## Node number 22: 8717 observations
##   predicted class=0  expected loss=0.06951933  P(node) =0.003113213
##     class counts:  8111   606
##    probabilities: 0.930 0.070
##
## Node number 23: 3342 observations,    complexity param=0.02423868
##   predicted class=0  expected loss=0.3847995  P(node) =0.001193571
##     class counts:  2056  1286
##    probabilities: 0.615 0.385
##   left son=46 (1995 obs) right son=47 (1347 obs)
##   Primary splits:
##       app     < 33     to the left,  improve=502.94850, (0 missing)
##       channel < 271    to the left,  improve=267.13310, (0 missing)
##       os      < 25.5   to the right, improve= 46.24167, (0 missing)
##       ip      < 128131 to the left,  improve= 30.71654, (0 missing)
##       device  < 1.5    to the right, improve= 17.78570, (0 missing)
##   Surrogate splits:
##       channel < 61     to the right, agree=0.809, adj=0.526, (0 split)
```

```
##       os      < 86.5   to the left,  agree=0.601, adj=0.010, (0 split)
##       ip      < 894    to the right, agree=0.598, adj=0.004, (0 split)
##       hora    splits as  -RLLL,      agree=0.597, adj=0.001, (0 split)
##
## Node number 46: 1995 observations
##   predicted class=0  expected loss=0.1593985  P(node) =0.0007124997
##     class counts:  1677   318
##    probabilities: 0.841 0.159
##
## Node number 47: 1347 observations,    complexity param=0.01851852
##   predicted class=1  expected loss=0.281366  P(node) =0.0004810713
##     class counts:   379   968
##    probabilities: 0.281 0.719
##   left son=94 (184 obs) right son=95 (1163 obs)
##   Primary splits:
##       os      < 25.5   to the right, improve=91.4473900, (0 missing)
##       channel < 258.5  to the left,  improve=64.2760000, (0 missing)
##       ip      < 189840 to the left,  improve= 3.3511710, (0 missing)
##       app     < 34.5   to the left,  improve= 1.9960300, (0 missing)
##       hora    splits as  -RLLR,      improve= 0.8496116, (0 missing)
##   Surrogate splits:
##       device < 1516.5 to the right, agree=0.865, adj=0.011, (0 split)
##
## Node number 94: 184 observations
##   predicted class=0  expected loss=0.2554348  P(node) =6.571426e-05
##     class counts:   137    47
##    probabilities: 0.745 0.255
##
## Node number 95: 1163 observations
##   predicted class=1  expected loss=0.2080825  P(node) =0.000415357
##     class counts:   242   921
##    probabilities: 0.208 0.792
```

```
#Verificando a taxa de acerto do modelo com uma confussion matrix

confusionMatrix(previsoes, dados_teste$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction       0       1
##          0 1197790    1673
##          1     127     409
##
##                Accuracy : 0.9985
##                  95% CI : (0.9984, 0.9986)
##     No Information Rate : 0.9983
##     P-Value [Acc > NIR] : 1.322e-10
##
##                   Kappa : 0.312
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9999
##             Specificity : 0.1964
```

```
##          Pos Pred Value : 0.9986
##          Neg Pred Value : 0.7631
##              Prevalence : 0.9983
##          Detection Rate : 0.9982
##    Detection Prevalence : 0.9996
##       Balanced Accuracy : 0.5982
##
##        'Positive' Class : 0
##
```

```
#O modelo obteve uma taxa de acurácia de 99.85%
```