Detecção de Fraude - Mini Projeto DSA

Diretorio do Projeto e opções

Importando os pacotes utilizados

```r
library(readr)
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
```

```r
library(tidyr)
library("corrgram")
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

```r
library(lubridate)
```

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

## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union

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

```r
library(BBmisc)
```

```
##
## Attaching package: 'BBmisc'

## The following objects are masked from 'package:dplyr':
##
##     coalesce, collapse

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

Carregando os dados

```r
dataset_train <- read_csv("C:/FCD/1-BigDataRAzure/ProjetoFeedBack/Projeto1/data/train_sample.csv")
```

```
## 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()
## )
```

Visualizar geral dos dados

```r
head(dataset_train)
```

```
## # A tibble: 6 x 8
##       ip   app device    os channel click_time          attributed_time
##    <dbl> <dbl>  <dbl> <dbl>   <dbl> <dttm>              <dttm>
## 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
## # ... with 1 more variable: is_attributed <dbl>
```

```
## tibble [100,000 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ip             : num [1:100000] 87540 105560 101424 94584 68413 ...
##  $ app            : num [1:100000] 12 25 12 13 12 3 1 9 2 3 ...
##  $ device         : num [1:100000] 1 1 1 1 1 1 1 1 2 1 ...
##  $ os             : num [1:100000] 13 17 19 13 1 17 17 25 22 19 ...
##  $ channel        : num [1:100000] 497 259 212 477 178 115 135 442 364 135 ...
##  $ click_time     : POSIXct[1:100000], format: "2017-11-07 09:30:38" "2017-11-07 13:40:27" ...
##  $ attributed_time: POSIXct[1:100000], format: NA NA ...
##  $ is_attributed  : num [1:100000] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. 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()
##   .. )
```

——————————-Data fields———————————————— Each row of the training data contains a click record, with the following features.

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 download 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

```
#Checando NA
apply(dataset_train, 2, function(x) any(is.na(x)))
```

```
##              ip             app          device              os         channel
##           FALSE           FALSE           FALSE           FALSE           FALSE
##      click_time attributed_time   is_attributed
##           FALSE            TRUE           FALSE
```

```
#Quantidades de IP, devices, app, channel e ip.
apply(dataset_train,2,function(x) length(unique(x)))
```

```
##              ip             app          device              os         channel
##           34857             161             100             130             161
##      click_time attributed_time   is_attributed
##           80350             228               2
```

Criando variaveis diarias

```
dataset_train <- separate(dataset_train, col = 'click_time', into = c('data','horario'), sep = ' ')
dataset_train$Dia_Semana <- wday(dataset_train$data)
dataset_train$horario <- hour(as.POSIXct(dataset_train$horario
                                         , format = c("%H:%M:%S")))
#Deletando a variavel attributed time visto que a maioria de seus valores sao NA.
dataset_train$attributed_time <- NULL
dataset_train$data <- NULL
head(dataset_train)
```

```
## # A tibble: 6 x 8
##       ip   app device    os channel horario is_attributed Dia_Semana
##    <dbl> <dbl>  <dbl> <dbl>   <dbl>   <int>         <dbl>      <dbl>
## 1  87540    12      1    13     497       9             0          3
## 2 105560    25      1    17     259      13             0          3
## 3 101424    12      1    19     212      18             0          3
## 4  94584    13      1    13     477       4             0          3
## 5  68413    12      1     1     178       9             0          5
## 6  93663     3      1    17     115       1             0          5
```

Analisando a distribuição de dados

```
table(as.factor(dataset_train$is_attributed))
```

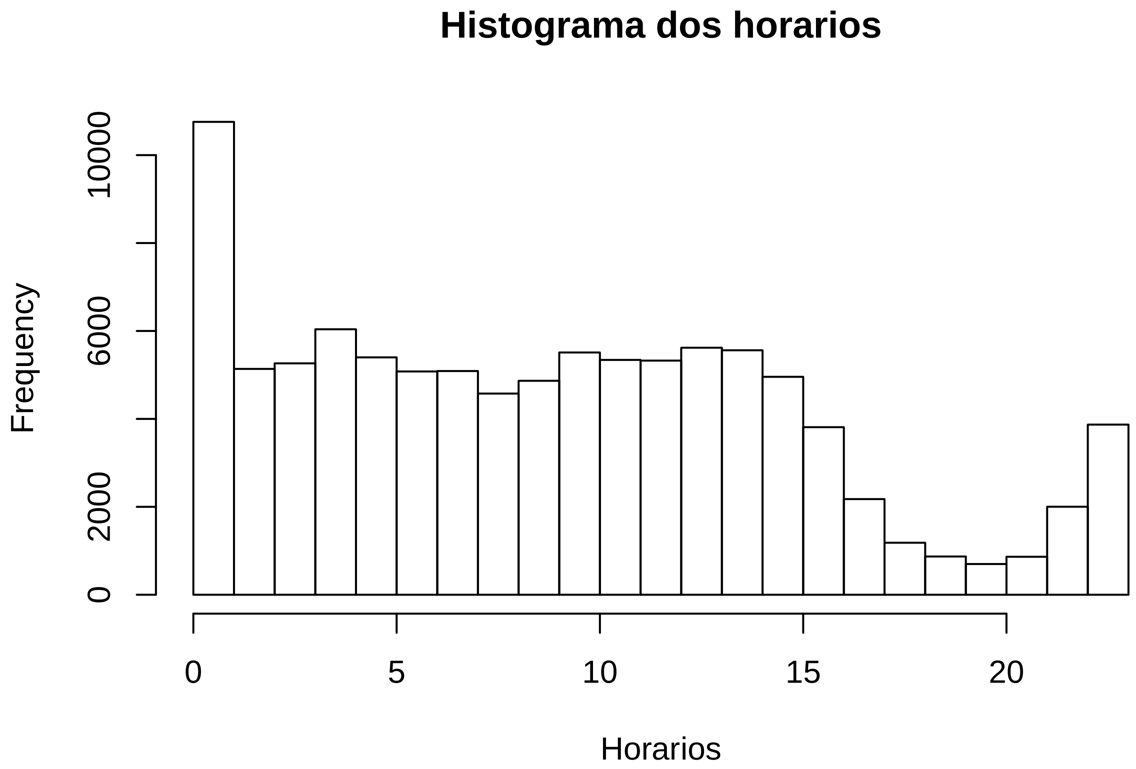
```
##
##     0     1
## 99773   227
```

```
print("Dados Desbalanceados")
```

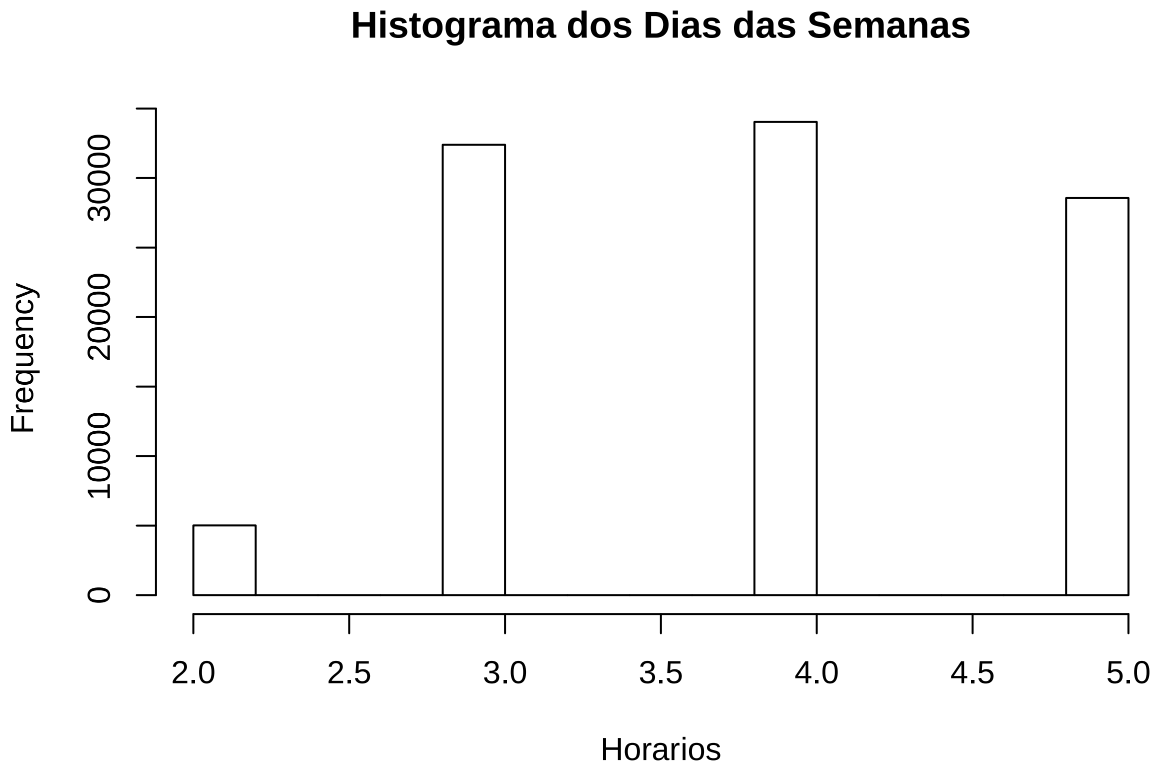
```
## [1] "Dados Desbalanceados"
```

Analise de dados de modo grafico

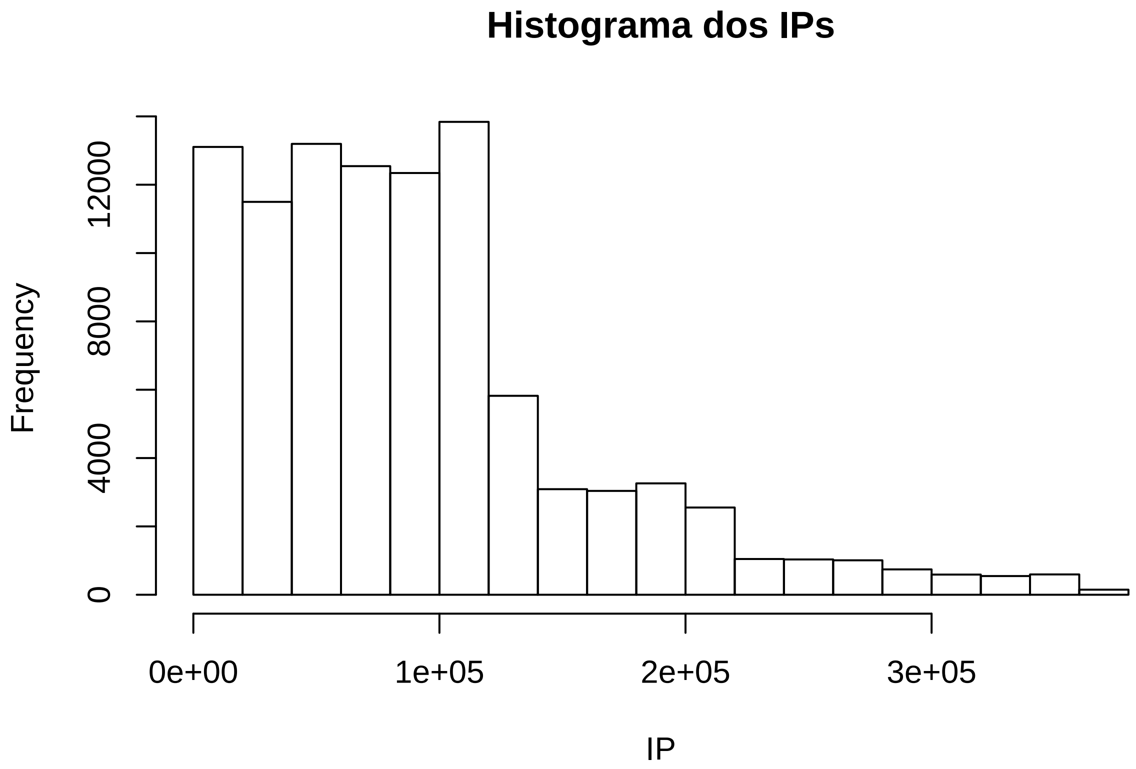
```
library(ggplot2)
hist(dataset_train$horario, xlab = "Horarios" , main =  "Histograma dos horarios")
```



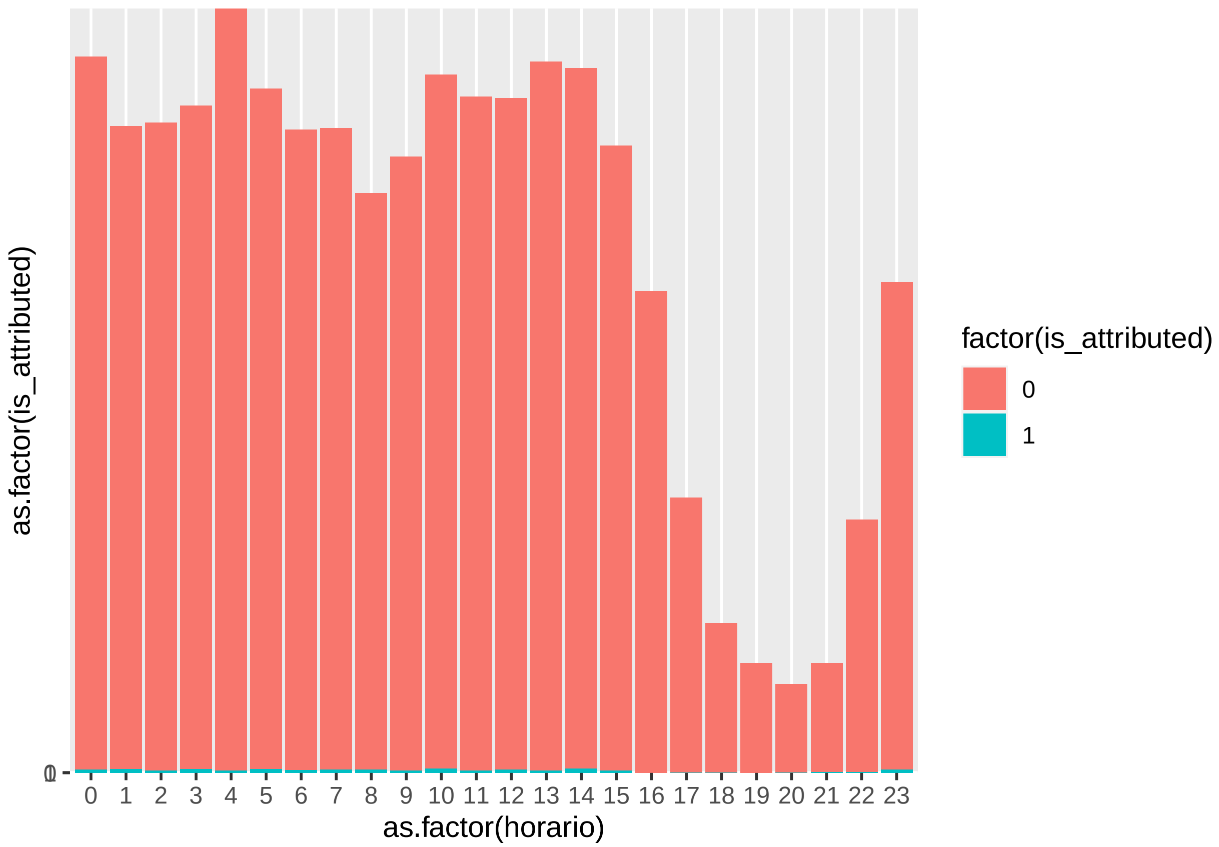
```r
hist(dataset_train$Dia_Semana, xlab = "Horarios" , main =  "Histograma dos Dias das Semanas")
```



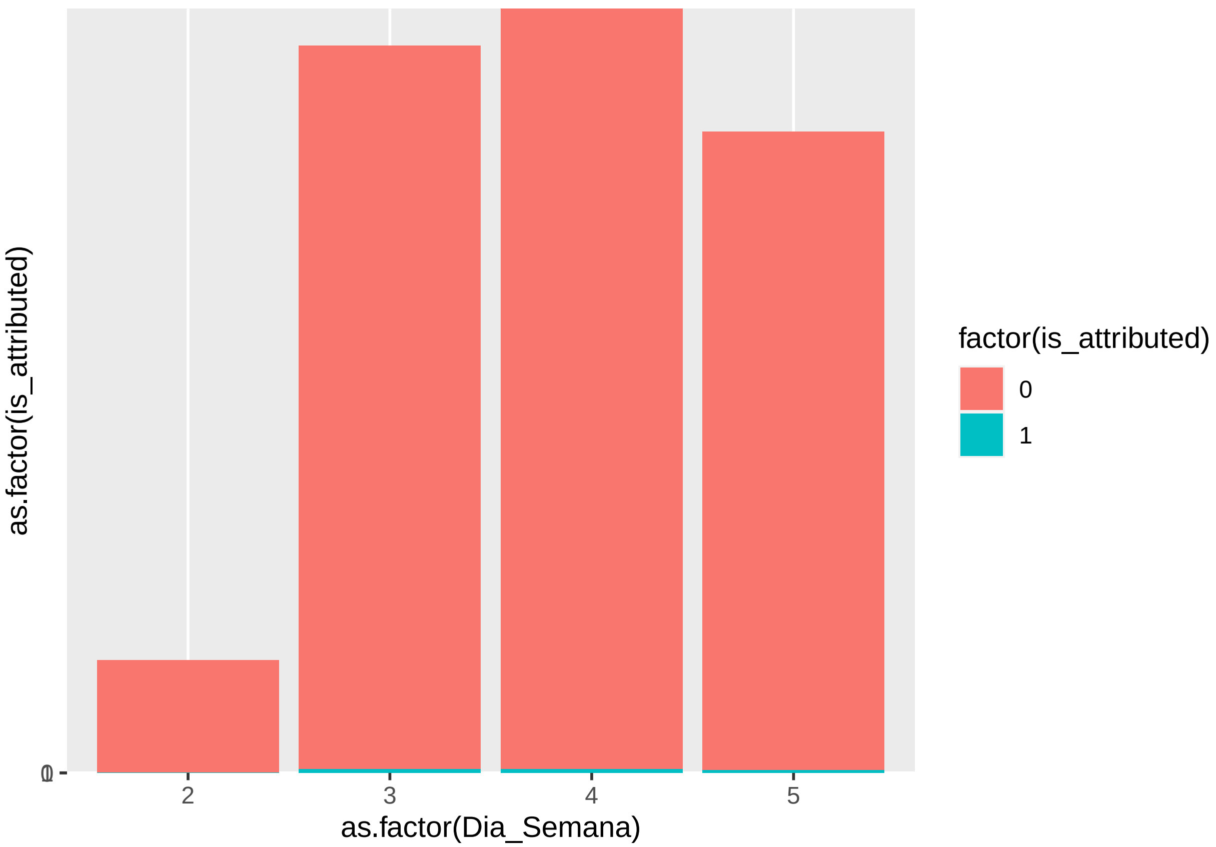
```r
hist(x = dataset_train$ip, xlab = "IP" , main =  "Histograma dos IPs")
```



```
ggplot(dataset_train, aes(x = as.factor(horario),
                          y = as.factor(is_attributed) ,fill=factor(is_attributed)))   +  geom_bar(stat=
```

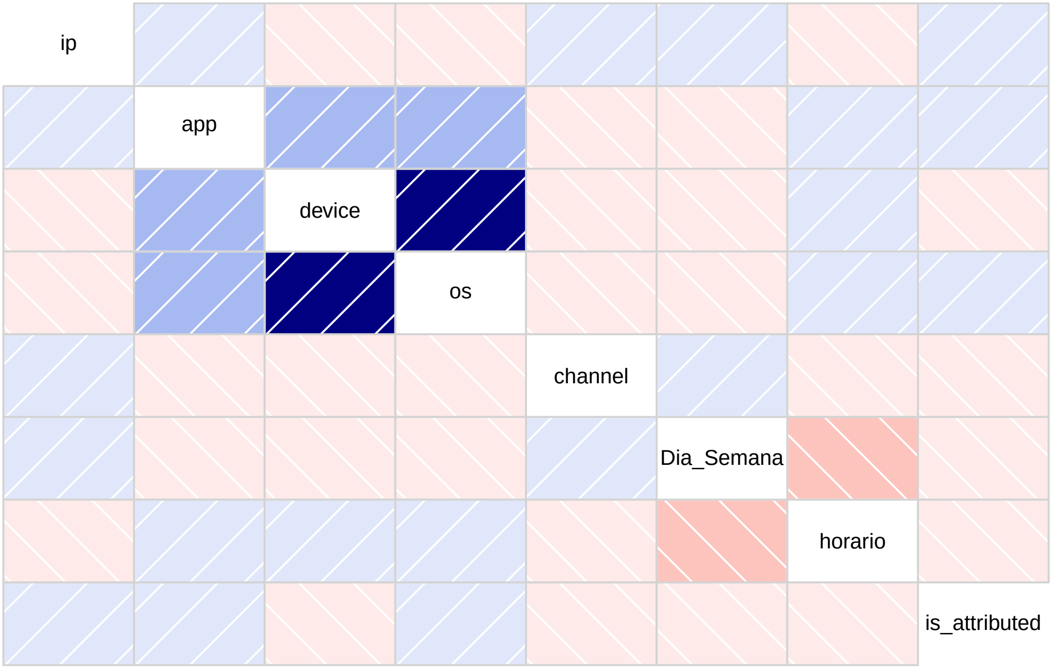


```
ggplot(dataset_train, aes(x = as.factor(Dia_Semana),
                          y = as.factor(is_attributed) ,fill=factor(is_attributed))) +
  geom_bar(stat="identity")
```



Verificação, escolha e ajuste das variaveis preditivas e target.

```r
#Grafico de correlação
var <- c("ip", "app", "device", "os", "channel", "Dia_Semana", "horario", "is_attributed" )
corrgram(dataset_train[,var])
```



```r
#Verificaçao
dim(dataset_train)
```

```
## [1] 100000      8
```

```r
apply(dataset_train, 2, function(x) any(is.na(x)))
```

```
##            ip           app        device            os       channel
##         FALSE         FALSE         FALSE         FALSE         FALSE
##       horario is_attributed    Dia_Semana
##         FALSE         FALSE         FALSE
```

```r
#Colocando a variavel target como fator
dataset_train$is_attributed <- as.factor(dataset_train$is_attributed)
```

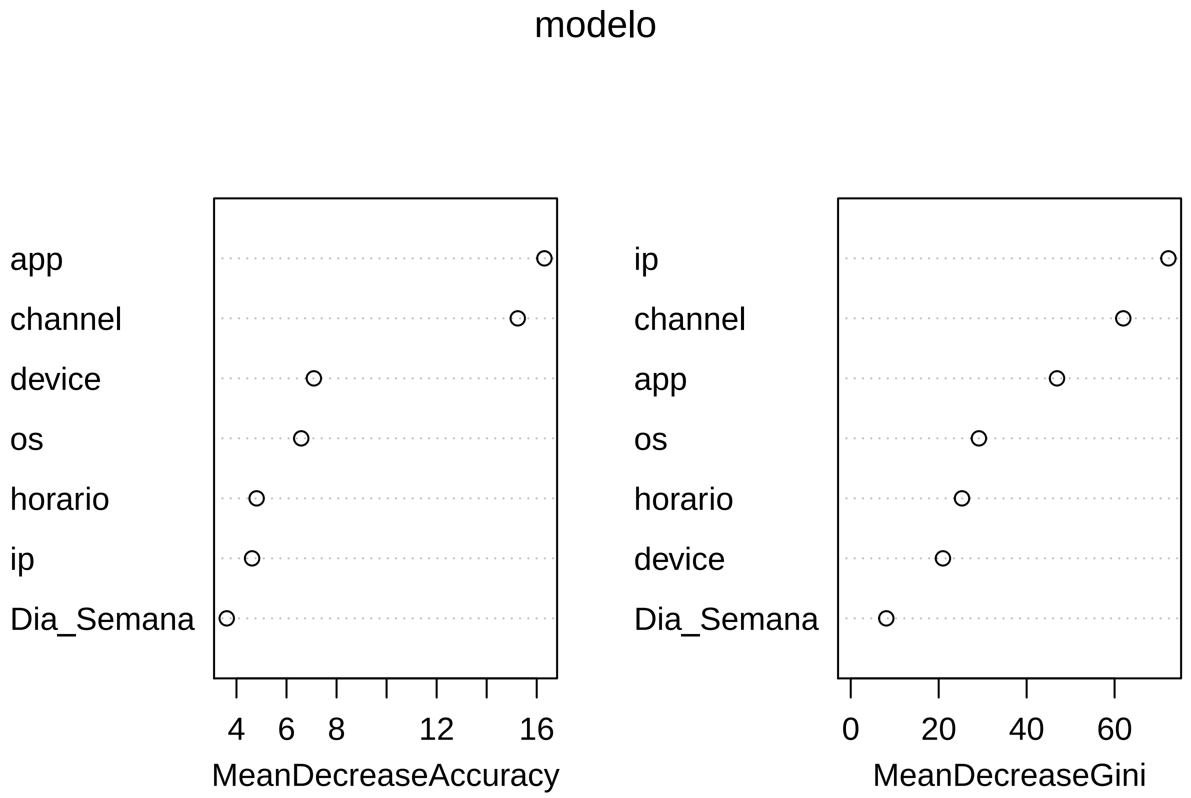
Feature Selection

```r
# Criando um modelo para identificar os atributos com maior importância para o modelo preditivo
require(randomForest)
```

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

```
# Avalidando a importância de todas as variaveis
modelo <- randomForest(is_attributed ~ .,
                       data = dataset_train,
                       ntree = 100, nodesize = 10, importance = T)


# Plotando as variáveis por grau de importância
varImpPlot(modelo)
```



```
modelo$importance
```

```
##                         0           1 MeanDecreaseAccuracy MeanDecreaseGini
## ip          -9.568078e-05 0.092343131         1.161734e-04        72.234613
## app          1.820200e-03 0.149739288         2.158266e-03        46.904366
## device       6.008380e-04 0.014623909         6.337824e-04        20.964112
## os           3.624488e-04 0.042597162         4.590342e-04        29.140795
## channel      1.644749e-03 0.078639944         1.822724e-03        61.977125
## horario      4.122924e-05 0.017024399         8.087132e-05        25.298525
## Dia_Semana   4.830078e-05 0.001998894         5.227489e-05         8.081725
```

Separação das variaveis em treino e teste

```r
#Separando dados de treino e de teste
# Funcao para gerar dados de treino e dados de teste
splitData <- function(dataframe) {
  index <- 1:nrow(dataframe)
  trainindex <- sample(index, trunc(length(index)/2))
  trainset <- dataframe[trainindex, ]
  testset <- dataframe[-trainindex, ]
  list(trainset = trainset, testset = testset)
}

# Gerando dados de treino e de teste
splits <- splitData(dataset_train)

# Separando os dados
dados_treino <- splits$trainset
dados_teste <- splits$testset
```

Criando o Modelo de Classificação com os dados de treino Realizando previsões e avaliações com os dados de teste

```r
# Construindo o modelo
modelo <- randomForest(is_attributed ~ .,
                       data = dados_treino,
                       ntree = 150, nodesize = 10)
print(modelo)
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ ., data = dados_treino,      ntree = 150, nodesize = 10)
##                Type of random forest: classification
##                      Number of trees: 150
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.23%
## Confusion matrix:
##       0  1  class.error
## 0 49876  1 2.004932e-05
## 1   112 11 9.105691e-01
```

```r
# Previsoes e analise
previsoes <- data.frame(observado = dados_teste$is_attributed,
                        previsto = predict(modelo, newdata = dados_teste))

table(previsoes)
```

```
##          previsto
## observado     0     1
##         0 49895     1
##         1   100     4
```

```r
prop.table(table(previsoes),2)
```

```
##          previsto
## observado         0         1
##         0 0.9979998 0.2000000
##         1 0.0020002 0.8000000
```

Optmização 1- Matriz de custo no algoritmo Random Florest

```
# Optimizacao do projeto

# Criando uma Cost Function
# Colocando um custo mais pesado caso de um falso positivo
Cost_func <- matrix(c(0, 0.5, 1, 0), nrow = 2, dimnames = list(c("1", "2"), c("1", "2")))



modelo_v2 <- randomForest(is_attributed ~ . -device -Dia_Semana,
                       data = dados_treino,
                       cost = Cost_func,
                       ntree = 150, nodesize = 10)

print(modelo_v2)
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ . - device - Dia_Semana,      data = dados_treino, cost = Cos
##                Type of random forest: classification
##                      Number of trees: 150
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.22%
## Confusion matrix:
##       0  1  class.error
## 0 49867 10 0.0002004932
## 1   102 21 0.8292682927
```

Optmização 1- Matriz de custo no algoritmo Random Florest

```
# Optimizacao do projeto

# Criando uma Cost Function
# Colocando um custo mais pesado caso de um falso positivo
Cost_func <- matrix(c(0, 0.5, 1, 0), nrow = 2, dimnames = list(c("1", "2"), c("1", "2")))



modelo_v2 <- randomForest(is_attributed ~ . -device -Dia_Semana,
                       data = dados_treino,
                       cost = Cost_func,
                       ntree = 150, nodesize = 10)

print(modelo_v2)
```

```
##
## Call:
##  randomForest(formula = is_attributed ~ . - device - Dia_Semana,      data = dados_treino, cost = Co
##                Type of random forest: classification
##                      Number of trees: 150
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.22%
## Confusion matrix:
```

```
##       0  1  class.error
## 0 49866 11 0.0002205425
## 1   101 22 0.8211382114
```

```
# Previsoes e analise
# Dataframes com valores observados e previstos Modelo V2
previsoes2 <- data.frame(observado = dados_teste$is_attributed,
                         previsto = predict(modelo_v2, newdata = dados_teste))
table(previsoes2)
```

```
##          previsto
## observado     0     1
##         0 49892     4
##         1    87    17
```

```
prop.table(table(previsoes2),2)
```

```
##          previsto
## observado           0           1
##         0 0.998259269 0.190476190
##         1 0.001740731 0.809523810
```

Optmização 2- C50 e matriz de custo

```
#C50 Modelo para tentar optimizacao
require(C50)
```

```
## Loading required package: C50
```

```
modelo_C50  <- C5.0(is_attributed ~ .,
                   data = dados_treino,
                   trials = 100,
                   cost = Cost_func)
summary(modelo_C50)
```

```
##
## Call:
## C5.0.formula(formula = is_attributed ~ ., data = dados_treino, trials =
##  100, cost = Cost_func)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue May 19 13:12:41 2020
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 50000 cases (8 attributes) from undefined.data
##
## -----  Trial 0:  -----
##
## Decision tree:
##
## app <= 28: 0 (48509/71)
## app > 28:
## :...channel > 278: 0 (1170/5)
##     channel <= 278:
##     :...ip > 158663:
```

```
##         :...channel > 232: 1 (6)
##         :   channel <= 232:
##         :   :...channel <= 105: 1 (21/9)
##         :       channel > 105:
##         :       :...ip <= 329646: 0 (36/7)
##         :           ip > 329646: 1 (3)
##         ip <= 158663:
##         :...horario > 3: 0 (195/5)
##             horario <= 3:
##             :...app > 35: 0 (39/4)
##                 app <= 35:
##                 :...Dia_Semana > 4: 0 (6)
##                     Dia_Semana <= 4:
##                     :...app <= 32: 0 (5/1)
##                         app > 32: 1 (10/1)
##
## -----  Trial 1:  -----
##
## Decision tree:
##
## device <= 0: 1 (3351.5/426.8)
## device > 0:
## :...device > 11: 1 (2347/275.4)
##     device <= 11:
##     :...app <= 28: 0 (40195/4143.2)
##         app > 28: 1 (4106.5/1890.5)
##
## -----  Trial 2:  -----
##
## Decision tree:
##
## ip <= 161031:
## :...os <= 0: 1 (1156.9/387.8)
## :   os > 0: 0 (36119.2/3261.5)
## ip > 161031:
## :...channel <= 111: 0 (1661.2/6.5)
##     channel > 111:
##     :...app <= 4: 0 (1365.1)
##         app > 4:
##         :...channel <= 213: 1 (6097.4/1096.4)
##             channel > 213: 0 (3600.3/960.6)
##
## -----  Trial 3:  -----
##
## Decision tree:
##
## device <= 0: 1 (3890.9/1842.7)
## device > 0:
## :...device > 11: 0 (3242.6/1467.7)
##     device <= 11:
##     :...channel <= 114: 0 (5906.4/2453.9)
##         channel > 114:
##         :...app <= 28: 0 (30557.5/1625.7)
##             app > 28:
```

```
##             :...channel <= 278: 1 (1697.7/551.1)
##                 channel > 278: 0 (4704.8/409)
##
## -----  Trial 4:  -----
##
## Decision tree:
##
## device > 11: 1 (3225.6/1420.1)
## device <= 11:
## :...channel <= 114:
##     :...channel <= 111: 0 (3694.3/626.8)
##     :   channel > 111: 1 (2827.5/166)
##     channel > 114:
##     :...channel <= 140: 0 (7185.4/81.8)
##         channel > 140:
##         :...channel > 486: 0 (665.1/305.4)
##             channel <= 486:
##             :...channel > 445: 0 (4454.1/152.7)
##                 channel <= 445:
##                 :...channel > 442: 1 (545.2/87.1)
##                     channel <= 442:
##                     :...channel > 420: 0 (1061.7)
##                         channel <= 420:
##                         :...channel <= 417: 0 (26096.8/3686.5)
##                             channel > 417: 1 (244.2)
##
## -----  Trial 5:  -----
##
## Decision tree:
##
## device > 957: 0 (4852.3)
## device <= 957:
## :...device > 11: 1 (2195.6/773.7)
##     device <= 11:
##     :...channel <= 114:
##         :...os <= 25: 0 (4540.2/1136)
##         :   os > 25: 1 (2426.2/610.1)
##         channel > 114:
##         :...channel <= 140: 0 (5687.5/93.3)
##             channel > 140:
##             :...ip > 186810:
##                 :...os > 748: 1 (93.3)
##                 :   os <= 748:
##                 :   :...Dia_Semana <= 2: 1 (561.9/133.7)
##                 :       Dia_Semana > 2: 0 (5883.4/1731.6)
##                 ip <= 186810:
##                 :...channel > 486: 1 (579.3/251)
##                     channel <= 486:
##                     :...horario > 22: 0 (1244.9/440.4)
##                         horario <= 22:
##                         :...os > 29: 0 (2654.5)
##                             os <= 29:
##                             :...channel > 420: 0 (5530.9/120.3)
##                                 channel <= 420:
```

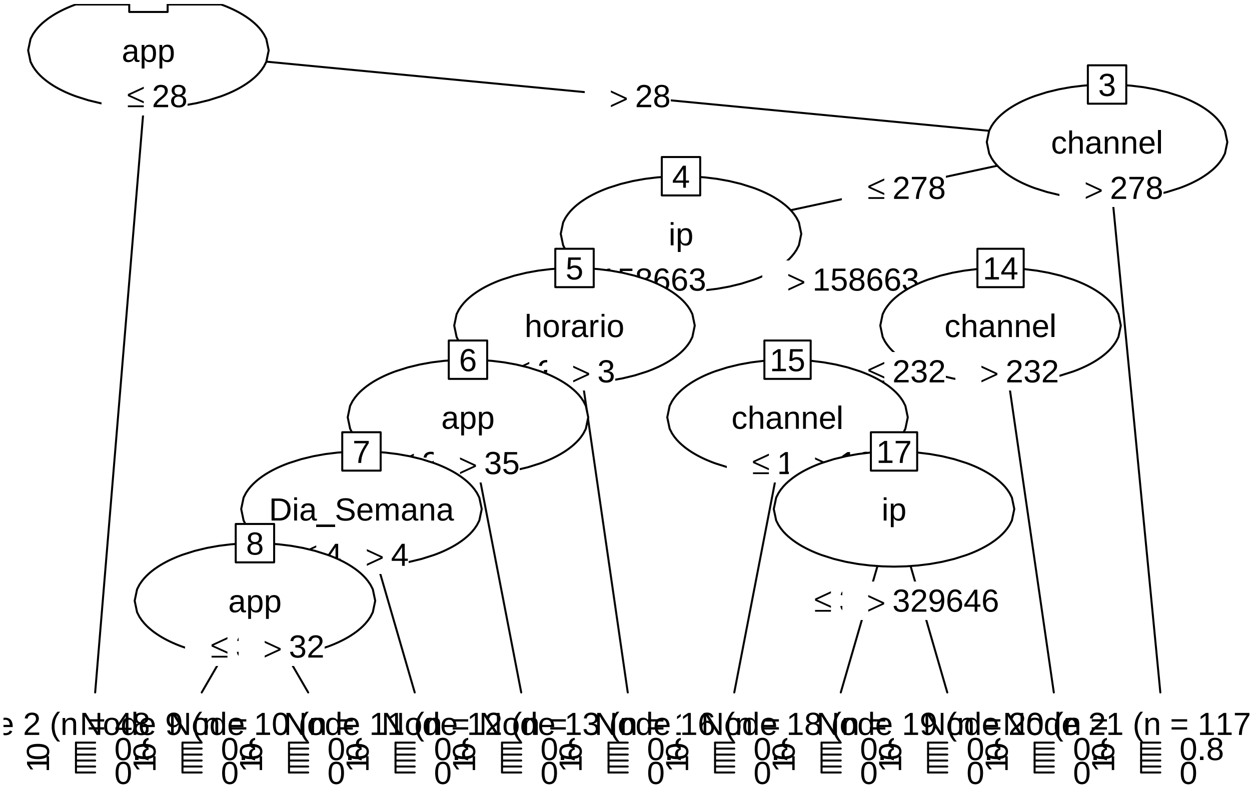
```
##                                  :...channel > 417: 1 (96.2)
##                                      channel <= 417:
##                                      :...channel > 377: 0 (704.9)
##                                          channel <= 377:
##                                          :...channel <= 376: 0 (12692.6/1960.1)
##                                              channel > 376: 1 (256.4/92.2)
##
## -----  Trial 6:  -----
##
## Decision tree:
##
## ip > 210631:
## :...ip > 358438: 0 (182.6)
## :   ip <= 358438:
## :   :...horario <= 3: 0 (931.9/116.7)
## :       horario > 3:
## :       :...os <= 23: 0 (4370.1/1613.5)
## :           os > 23: 1 (2759.5/1002.3)
## ip <= 210631:
## :...os > 61: 0 (4050.2)
##     os <= 61:
##     :...os > 60: 1 (169.7)
##         os <= 60:
##         :...device > 11: 0 (1551.9/754.5)
##             device <= 11:
##             :...os > 32: 0 (4290.9)
##                 os <= 32:
##                 :...app <= 18: 0 (20619.6/1732.6)
##                     app > 18:
##                     :...channel > 420: 0 (1341.7)
##                         channel <= 420:
##                         :...device > 1: 0 (214.2)
##                             device <= 1:
##                             :...ip > 202934: 0 (175.5)
##                                 ip <= 202934:
##                                 :...ip > 202760: 1 (76)
##                                     ip <= 202760:
##                                     :...ip > 196491: 0 (181.3)
##                                         ip <= 196491:
##                                         :...channel <= 101: 1 (1026.3/379.6)
##                                             channel > 101:
##                                             :...channel <= 160: 0 (1461.3/176.9)
##                                                 channel > 160:
##                                                 :...ip <= 187227: 0 (6363.4/2023.8)
##                                                     ip > 187227: 1 (233.9/30.8)
##
## -----  Trial 7:  -----
##
## Decision tree:
##
## device > 957: 0 (3289.4)
## device <= 957:
## :...device > 11: 1 (2143/871.3)
##     device <= 11:
```

```
##     :...os > 36: 0 (6946.7/451.7)
##         os <= 36:
##         :...channel > 347:
##             :...ip <= 342889: 0 (11210.6/889.2)
##             :   ip > 342889: 1 (359.9/37.1)
##             channel <= 347:
##             :...device > 1: 0 (732.6)
##                 device <= 1:
##                 :...ip <= 5328: 0 (546.8)
##                     ip > 5328:
##                     :...ip <= 6437: 1 (326.4/139.8)
##                         ip > 6437:
##                         :...channel > 212: 0 (10379.6/3727.3)
##                             channel <= 212:
##                             :...os > 35: 1 (313.1/105.6)
##                                 os <= 35:
##                                 :...channel > 171: 0 (1541.1)
##                                     channel <= 171:
##                                     :...channel <= 160: 0 (11797.4/2517.2)
##                                         channel > 160: 1 (413.4/128)
##
## -----  Trial 8:  -----
##
## Decision tree:
##
## device > 957: 0 (2644.3)
## device <= 957:
## :...ip <= 6437: 0 (6132/325.8)
##     ip > 6437:
##     :...ip <= 6486: 1 (157.7/1.2)
##         ip > 6486:
##         :...channel > 347: 0 (11234.1/1062.2)
##             channel <= 347:
##             :...ip <= 7318: 0 (136.9)
##                 ip > 7318:
##                 :...ip > 358440: 0 (101.6)
##                     ip <= 358440:
##                     :...ip > 358384: 1 (47.7/0.1)
##                         ip <= 358384:
##                         :...ip <= 7391: 1 (48.4/3.1)
##                             ip > 7391:
##                             :...ip <= 11450: 0 (299.4)
##                                 ip > 11450:
##                                 :...ip <= 11498: 1 (103.4/1)
##                                     ip > 11498:
##                                     :...ip <= 15187: 0 (262.3)
##                                         ip > 15187:
##                                         :...ip <= 15229: 1 (122.5/0.7)
##                                             ip > 15229:
##                                             :...channel <= 212: 0 (15629.8/3545.1)
##                                                 channel > 212:
##                                                 :...app > 146: 0 (549)
##                                                     app <= 146:
##                                                     :...ip <= 32206: 0 (508.4)
##                                                         ip > 32206:
##                                                         :...os > 24: 0 (3027.3/571.8)
##                                                             os <= 24: [S1]
##
## SubTree [S1]
##
## channel <= 282: 1 (7531.7/3464.1)
## channel > 282: 0 (1463.3/478.3)
##
## -----  Trial 9:  -----
##
## Decision tree:
##
## app <= 18:
## :...app > 11: 0 (13072.8/284.1)
## :   app <= 11:
## :   :...channel <= 114: 1 (3111.4/1329.5)
## :       channel > 114: 0 (16221/1618.2)
## app > 18:
## :...channel > 420: 0 (1016.9)
##     channel <= 420:
##     :...channel > 412: 1 (195.9)
##         channel <= 412:
##         :...device > 957: 0 (854.3)
##             device <= 957:
##             :...device > 928: 1 (74)
##                 device <= 928:
##                 :...os > 29: 0 (1863.4/145.3)
##                     os <= 29:
##                     :...ip > 232401: 1 (2794.9/1097.9)
##                         ip <= 232401:
##                         :...ip <= 5341: 0 (239.9)
##                             ip > 5341:
##                             :...device > 188: 0 (140.6)
##                                 device <= 188:
##                                 :...horario <= 5: 1 (3760.9/1782.5)
##                                     horario > 5: 0 (6654.1/1936.1)
##
## -----  Trial 10:  -----
##
## Decision tree:
##
## channel <= 111: 0 (9458.7/922.3)
## channel > 111:
## :...channel <= 114: 1 (2384.7/876)
##     channel > 114:
##     :...app > 15: 0 (17892.4/4310.3)
##         app <= 15:
##         :...os > 25: 0 (2926.2)
##             os <= 25:
##             :...os <= 11: 0 (2684)
##                 os > 11:
##                 :...horario <= 2: 0 (2000.1)
##                     horario > 2:
```

```
##                     :...ip <= 6479: 0 (1572.5)
##                         ip > 6479:
##                         :...ip <= 11977: 1 (444.7/160.3)
##                             ip > 11977: 0 (10636.8/1656.9)
##
## -----  Trial 11:  -----
##
## Decision tree:
##
## device <= 0: 1 (3968.3/1945.8)
## device > 0:
## :...app > 28:
##     :...os <= 43: 1 (6013/2834.2)
##     :   os > 43: 0 (653.2)
##     app <= 28:
##     :...ip <= 120141: 0 (26197.8/1299.4)
##         ip > 120141:
##         :...ip <= 120259: 1 (349.3/16.2)
##             ip > 120259: 0 (12818.3/2367.1)
##
## -----  Trial 12:  -----
##
## Decision tree:
##
## ip <= 160123: 0 (36152.8/3814.6)
## ip > 160123:
## :...app <= 4: 0 (1399.8)
##     app > 4:
##     :...device > 596: 0 (212.1)
##         device <= 596:
##         :...device > 516: 1 (84.3)
##             device <= 516:
##             :...os > 836: 1 (66.6)
##                 os <= 836:
##                 :...ip > 358440: 0 (202.6)
##                     ip <= 358440:
##                     :...device > 97: 0 (110.9)
##                         device <= 97:
##                         :...device > 17: 1 (335.5/84.7)
##                             device <= 17:
##                             :...app <= 10: 1 (3190.9/1268.7)
##                                 app > 10:
##                                 :...channel > 424: 0 (924)
##                                     channel <= 424:
##                                     :...app <= 28: 0 (4759.9/841.5)
##                                         app > 28: 1 (2560.5/1179.3)
##
## -----  Trial 13:  -----
##
## Decision tree:
##  0 (50000/8072.1)
##
## *** boosting reduced to 13 trials since last classifier is very inaccurate
##
```

```
##
## Evaluation on training data (50000 cases):
##
## Trial        Decision Tree
## -----      ----------------
##    Size      Errors
##
##    0     11  103( 0.2%)
##    1      4 1875( 3.8%)
##    2      6 1580( 3.2%)
##    3      6  419( 0.8%)
##    4     10  806( 1.6%)
##    5     16 2020( 4.0%)
##    6     18  580( 1.2%)
##    7     13  697( 1.4%)
##    8     18 9661(19.3%)
##    9     13 2801( 5.6%)
##   10      9  958( 1.9%)
##   11      6 1515( 3.0%)
##   12     12 1128( 2.3%)
## boost            103( 0.2%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##   49872     5    (a): class 0
##      98    25    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% ip
##  100.00% app
##  100.00% device
##  100.00% channel
##   100.00%    os
##   85.21% horario
##    7.84% Dia_Semana
##
##
## Time: 1.2 secs
```

```r
plot(modelo_C50)
```



```
# Previsoes e analise

# Dataframes com valores observados e previstos Modelo V2
previsoes_c50 <- data.frame(observado = dados_teste$is_attributed,
                          previsto = predict(modelo_C50, newdata = dados_teste))
table(previsoes_c50)
```

```
##          previsto
## observado     0     1
##         0 49889     7
##         1    96     8
```

```
prop.table(table(previsoes_c50),2)
```

```
##          previsto
## observado           0           1
##         0 0.998079424 0.466666667
##         1 0.001920576 0.533333333
```

Avaliação por ROC

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

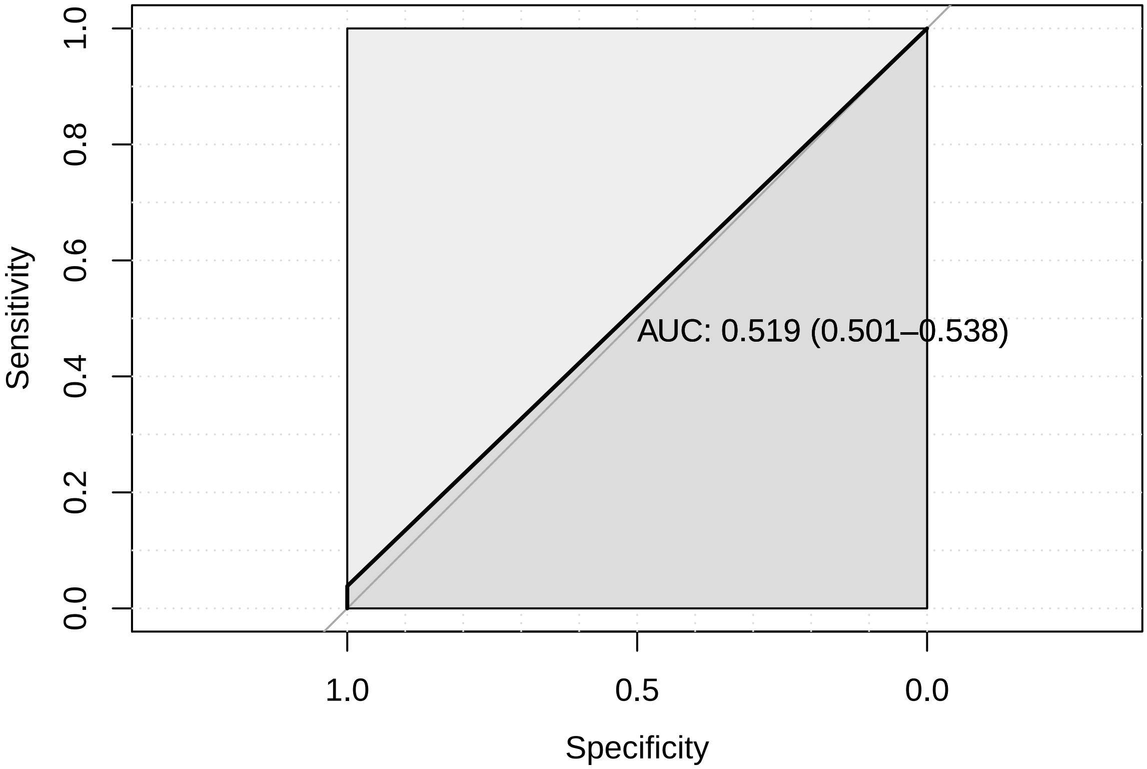
```
## The following objects are masked from 'package:stats':
```

```
##
##     cov, smooth, var
```

```
pROC_obj <- roc(as.numeric(previsoes$observado),as.numeric(previsoes$previsto),
                smoothed = TRUE,
                # arguments for ci
                ci=TRUE, ci.alpha=0.9, stratified=FALSE,
                # arguments for plot
                plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
                print.auc=TRUE, show.thres=TRUE)
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

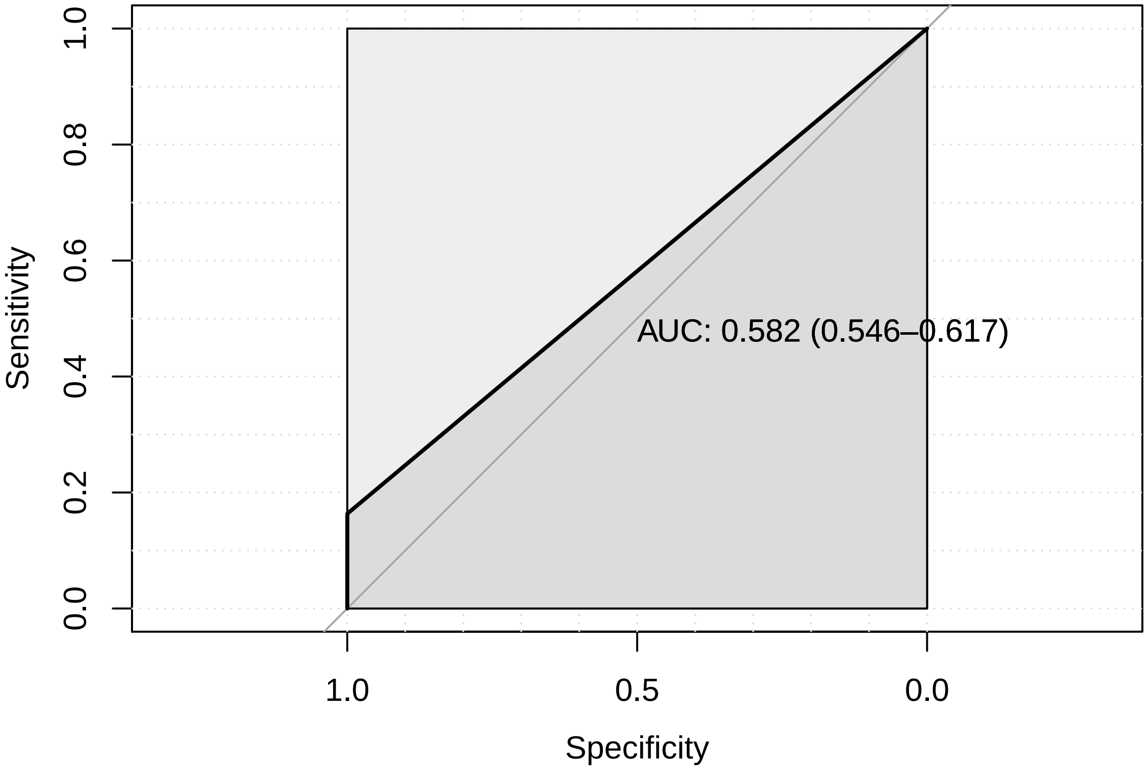


```
print(pROC_obj)
```

```
##
## Call:
## roc.default(response = as.numeric(previsoes$observado), predictor = as.numeric(previsoes$previsto),
##
## Data: as.numeric(previsoes$previsto) in 49896 controls (as.numeric(previsoes$observado) 1) < 104 case
## Area under the curve: 0.5192
## 95% CI: 0.5007-0.5378 (DeLong)
```

```
pROC_obj2 <- roc(as.numeric(previsoes2$observado),as.numeric(previsoes2$previsto),
                 smoothed = TRUE,
                 # arguments for ci
                 ci=TRUE, ci.alpha=0.9, stratified=FALSE,
                 # arguments for plot
                 plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
                 print.auc=TRUE, show.thres=TRUE)
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

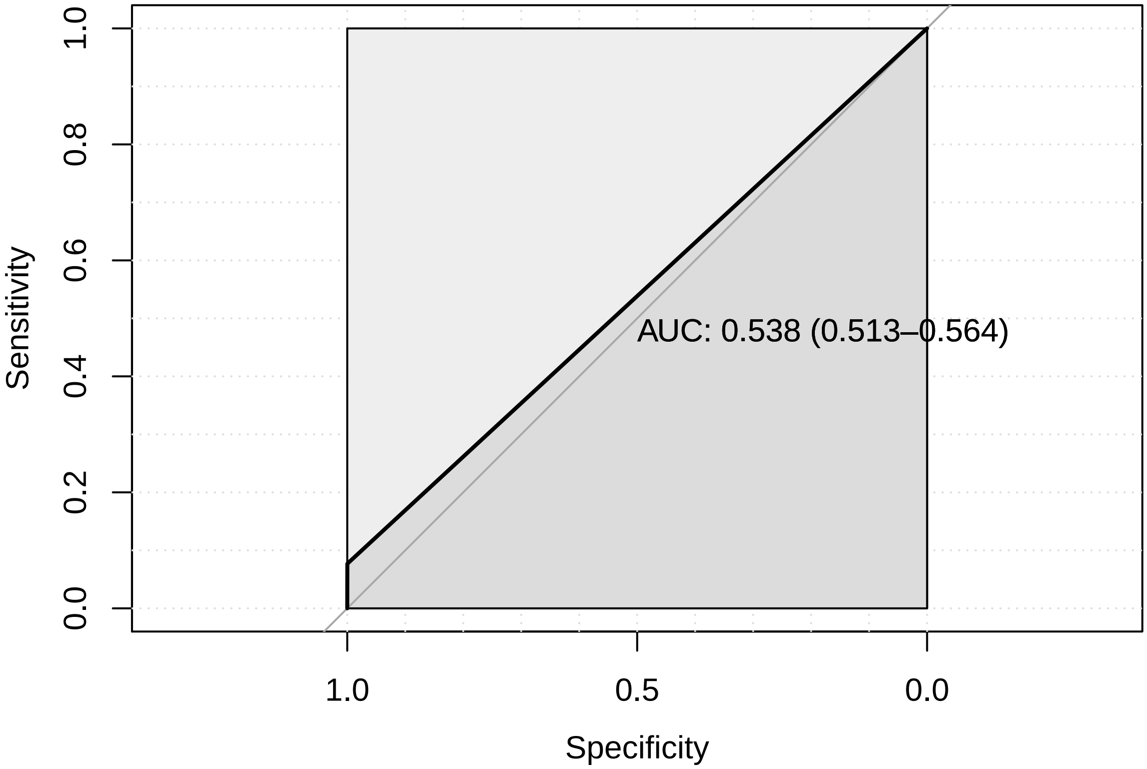


```
print(pROC_obj2)
```

```
##
## Call:
## roc.default(response = as.numeric(previsoes2$observado), predictor = as.numeric(previsoes2$previsto)
##
## Data: as.numeric(previsoes2$previsto) in 49896 controls (as.numeric(previsoes2$observado) 1) < 104 ca
## Area under the curve: 0.5817
## 95% CI: 0.546-0.6174 (DeLong)
```

```
pROC_objc50 <- roc(as.numeric(previsoes_c50$observado),as.numeric(previsoes_c50$previsto),
                 smoothed = TRUE,
                 # arguments for ci
                 ci=TRUE, ci.alpha=0.9, stratified=FALSE,
                 # arguments for plot
                 plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
                 print.auc=TRUE, show.thres=TRUE)
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```



```r
print(pROC_objc50)
```

```
##
## Call:
## roc.default(response = as.numeric(previsoes_c50$observado), predictor = as.numeric(previsoes_c50$pre
##
## Data: as.numeric(previsoes_c50$previsto) in 49896 controls (as.numeric(previsoes_c50$observado) 1) <
## Area under the curve: 0.5384
## 95% CI: 0.5127-0.5641 (DeLong)
```

Bibliografias utilizada

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

https://www.rdocumentation.org/

DSA cursos e scripts

https://rviews.rstudio.com/2019/01/17/roc-curves/

https://rviews.rstudio.com/2019/03/01/some-r-packages-for-roc-curves/

https://cran.r-project.org/web/packages/C50/vignettes/C5.0.html