# Airline passengers satisfaction

## Guillermo Peña

## 03/01/2021

This dataset contains an airline passenger satisfaction survey. The main questions that comes to mind are:

1. What factors are highly correlated to a satisfied (or dissatisfied) passenger?
2. Can you predict passenger satisfaction?

We are looking into a dataset that has been split in train and test datasets with 22 variables subject to explain the satisfaction variable which takes two values: "satisfied" or "neutral or dissatisfied".

Necessary libraries for the analysis

```
if (!require(broom)) install.packages('broom')
```

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

```
if (!require(tidyverse)) install.packages('tidyverse')
```

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

```
## -- Attaching packages -----------------------------------------------------
---------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```
if (!require(caret)) install.packages('caret')
```

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

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

```
##
## Attaching package: 'caret'
```

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

```
if (!require(MASS)) install.packages('MASS')
```

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

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
if (!require(ROCR)) install.packages('ROCR')
```

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

```
## Warning: package 'ROCR' was built under R version 4.0.2
```

```
if (!require(readr)) install.packages('readr')
library(broom)
library(tidyverse)
library(caret)
library(MASS)
library(ROCR)
library(readr)
```

```
url_train <- "https://github.com/guillepena/Passenger-Satisfaction/blob/master/train.csv"
train_data <- read.csv("train.csv")
url_test <- "https://github.com/guillepena/Passenger-Satisfaction/blob/master/test.csv"
test_data <- read.csv("test.csv")
```

The proportion of people for the two satisfaction levels are:

```
table(train_data$satisfaction)
```

```
##
## neutral or dissatisfied               satisfied
##                   58879                   45025
```

We first transform the variables that came from a rating to factor format.

```
train_data$Inflight.wifi.service = as.factor(train_data$Inflight.wifi.service)
train_data$Departure.Arrival.time.convenient = as.factor(train_data$Departure.Arriv
al.time.convenient)
train_data$Ease.of.Online.booking = as.factor(train_data$Ease.of.Online.booking)
train_data$Gate.location = as.factor(train_data$Gate.location)
train_data$Food.and.drink = as.factor(train_data$Food.and.drink)
train_data$Online.boarding = as.factor(train_data$Online.boarding)
train_data$Seat.comfort = as.factor(train_data$Seat.comfort)
train_data$Inflight.entertainment = as.factor(train_data$Inflight.entertainment)
train_data$On.board.service = as.factor(train_data$On.board.service)
train_data$Leg.room.service = as.factor(train_data$Leg.room.service)
train_data$Baggage.handling = as.factor(train_data$Baggage.handling)
train_data$Checkin.service = as.factor(train_data$Checkin.service)
train_data$Inflight.service = as.factor(train_data$Inflight.service)
train_data$Cleanliness = as.factor(train_data$Cleanliness)
train_data$satisfaction = as.factor(train_data$satisfaction)
str(train_data)
```

```
## 'data.frame':    103904 obs. of  25 variables:
##  $ X                                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ id                               : int  70172 5047 110028 24026 119299 111157
82113 96462 79485 65725 ...
##  $ Gender                           : chr  "Male" "Male" "Female" "Female" ...
##  $ Customer.Type                    : chr  "Loyal Customer" "disloyal Customer"
"Loyal Customer" "Loyal Customer" ...
##  $ Age                              : int  13 25 26 25 61 26 47 52 41 20 ...
##  $ Type.of.Travel                   : chr  "Personal Travel" "Business travel" "
Business travel" "Business travel" ...
##  $ Class                            : chr  "Eco Plus" "Business" "Business" "Bus
iness" ...
##  $ Flight.Distance                  : int  460 235 1142 562 214 1180 1276 2035 8
53 1061 ...
##  $ Inflight.wifi.service            : Factor w/ 6 levels "0","1","2","3",..: 4 4
3 3 4 4 3 5 2 4 ...
##  $ Departure.Arrival.time.convenient: Factor w/ 6 levels "0","1","2","3",..: 5 3
3 6 4 5 5 4 3 4 ...
##  $ Ease.of.Online.booking           : Factor w/ 6 levels "0","1","2","3",..: 4 4
3 6 4 3 3 5 3 4 ...
##  $ Gate.location                    : Factor w/ 6 levels "0","1","2","3",..: 2 4
3 6 4 2 4 5 3 5 ...
##  $ Food.and.drink                   : Factor w/ 6 levels "0","1","2","3",..: 6 2
6 3 5 2 3 6 5 3 ...
##  $ Online.boarding                  : Factor w/ 6 levels "0","1","2","3",..: 4 4
6 3 6 3 3 6 4 4 ...
##  $ Seat.comfort                     : Factor w/ 6 levels "0","1","2","3",..: 6 2
6 3 6 2 3 6 4 4 ...
##  $ Inflight.entertainment           : Factor w/ 6 levels "0","1","2","3",..: 6 2
6 3 4 2 3 6 2 3 ...
##  $ On.board.service                 : Factor w/ 6 levels "0","1","2","3",..: 5 2
5 3 4 4 4 6 2 3 ...
##  $ Leg.room.service                 : Factor w/ 6 levels "0","1","2","3",..: 4 6
4 6 5 5 4 6 3 4 ...
##  $ Baggage.handling                 : Factor w/ 5 levels "1","2","3","4",..: 4 3
4 3 4 4 4 5 1 4 ...
##  $ Checkin.service                  : Factor w/ 6 levels "0","1","2","3",..: 5 2
5 2 4 5 4 5 5 5 ...
##  $ Inflight.service                 : Factor w/ 6 levels "0","1","2","3",..: 6 5
5 5 4 5 6 6 2 4 ...
##  $ Cleanliness                      : Factor w/ 6 levels "0","1","2","3",..: 6 2
6 3 4 2 3 5 3 3 ...
##  $ Departure.Delay.in.Minutes       : int  25 1 0 11 0 0 9 4 0 0 ...
##  $ Arrival.Delay.in.Minutes         : num  18 6 0 9 0 0 23 0 0 0 ...
##  $ satisfaction                     : Factor w/ 2 levels "neutral or dissatisfie
d",..: 1 1 2 1 2 1 1 2 1 1 ...
```

```
test_data$Inflight.wifi.service = as.factor(test_data$Inflight.wifi.service)
test_data$Departure.Arrival.time.convenient = as.factor(test_data$Departure.Arrival
.time.convenient)
test_data$Ease.of.Online.booking = as.factor(test_data$Ease.of.Online.booking)
test_data$Gate.location = as.factor(test_data$Gate.location)
test_data$Food.and.drink = as.factor(test_data$Food.and.drink)
test_data$Online.boarding = as.factor(test_data$Online.boarding)
test_data$Seat.comfort = as.factor(test_data$Seat.comfort)
test_data$Inflight.entertainment = as.factor(test_data$Inflight.entertainment)
test_data$On.board.service = as.factor(test_data$On.board.service)
test_data$Leg.room.service = as.factor(test_data$Leg.room.service)
test_data$Baggage.handling = as.factor(test_data$Baggage.handling)
test_data$Checkin.service = as.factor(test_data$Checkin.service)
test_data$Inflight.service = as.factor(test_data$Inflight.service)
test_data$Cleanliness = as.factor(test_data$Cleanliness)
test_data$satisfaction = as.factor(test_data$satisfaction)
str(test_data)
```

```
## 'data.frame':    25976 obs. of  25 variables:
##  $ X                                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ id                               : int  19556 90035 12360 77959 36875 39177 7
9433 97286 27508 62482 ...
##  $ Gender                           : chr  "Female" "Female" "Male" "Male" ...
##  $ Customer.Type                    : chr  "Loyal Customer" "Loyal Customer" "di
sloyal Customer" "Loyal Customer" ...
##  $ Age                              : int  52 36 20 44 49 16 77 43 47 46 ...
##  $ Type.of.Travel                   : chr  "Business travel" "Business travel" "
Business travel" "Business travel" ...
##  $ Class                            : chr  "Eco" "Business" "Eco" "Business" ...
##  $ Flight.Distance                  : int  160 2863 192 3377 1182 311 3987 2556 
556 1744 ...
##  $ Inflight.wifi.service            : Factor w/ 6 levels "0","1","2","3",..: 6 2 
3 1 3 4 6 3 6 3 ...
##  $ Departure.Arrival.time.convenient: Factor w/ 6 levels "0","1","2","3",..: 5 2 
1 1 4 4 6 3 3 3 ...
##  $ Ease.of.Online.booking           : Factor w/ 6 levels "0","1","2","3",..: 4 4 
3 1 5 4 6 3 3 3 ...
##  $ Gate.location                    : Factor w/ 5 levels "1","2","3","4",..: 4 1 
4 2 3 3 5 2 2 2 ...
##  $ Food.and.drink                   : Factor w/ 6 levels "0","1","2","3",..: 4 6 
3 4 5 6 4 5 6 4 ...
##  $ Online.boarding                  : Factor w/ 6 levels "0","1","2","3",..: 5 5 
3 5 2 6 6 5 6 5 ...
##  $ Seat.comfort                     : Factor w/ 5 levels "1","2","3","4",..: 3 5 
2 4 2 3 5 5 5 4 ...
##  $ Inflight.entertainment           : Factor w/ 6 levels "0","1","2","3",..: 6 5 
3 2 3 6 6 5 6 5 ...
##  $ On.board.service                 : Factor w/ 6 levels "0","1","2","3",..: 6 5 
5 2 3 5 6 5 3 5 ...
##  $ Leg.room.service                 : Factor w/ 6 levels "0","1","2","3",..: 6 5 
2 2 3 4 6 5 3 5 ...
##  $ Baggage.handling                 : Factor w/ 5 levels "1","2","3","4",..: 5 4 
3 1 2 1 5 4 5 4 ...
##  $ Checkin.service                  : Factor w/ 5 levels "1","2","3","4",..: 2 3 
2 3 4 1 4 5 3 5 ...
##  $ Inflight.service                 : Factor w/ 6 levels "0","1","2","3",..: 6 5 
3 2 3 3 6 5 4 5 ...
##  $ Cleanliness                      : Factor w/ 6 levels "0","1","2","3",..: 6 6 
3 5 5 6 4 4 6 5 ...
##  $ Departure.Delay.in.Minutes       : int  50 0 0 0 0 0 0 77 1 28 ...
##  $ Arrival.Delay.in.Minutes         : num  44 0 0 6 20 0 0 65 0 14 ...
##  $ satisfaction                     : Factor w/ 2 levels "neutral or dissatisfie
d",..: 2 2 1 2 2 2 2 2 2 2 ...
```

Perform a copy of the datasets.

```
train_data_copy = train_data
test_data_copy = test_data
```

We want to make sure that we have no NAs in the dataset

```
NA_position_train <- which(is.na(train_data_copy$Arrival.Delay.in.Minutes))
train_data_copy$Arrival.Delay.in.Minutes[NA_position_train] = mean(train_data_copy$
Arrival.Delay.in.Minutes, na.rm = TRUE)
NA_position_test <- which(is.na(test_data_copy$Arrival.Delay.in.Minutes))
test_data_copy$Arrival.Delay.in.Minutes[NA_position_test] = mean(test_data_copy$Arr
ival.Delay.in.Minutes, na.rm = TRUE)
```

We try first to perform a logistic regression model on the dataset.

```
est_mod <- glm(satisfaction ~ Gender + Customer.Type + Age +
                 Type.of.Travel + Class + Flight.Distance + Inflight.wifi.service +
                 Departure.Arrival.time.convenient + Ease.of.Online.booking +
                 Gate.location + Food.and.drink + Online.boarding + Seat.comfort +
                 Inflight.entertainment + On.board.service + Leg.room.service +
                 Baggage.handling + Checkin.service + Inflight.service +
                 Cleanliness + Departure.Delay.in.Minutes + Arrival.Delay.in.Minute
s , data = train_data_copy, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(est_mod)
```

```
##
## Call:
## glm(formula = satisfaction ~ Gender + Customer.Type + Age + Type.of.Travel +
##     Class + Flight.Distance + Inflight.wifi.service + Departure.Arrival.time.con
venient +
##     Ease.of.Online.booking + Gate.location + Food.and.drink +
##     Online.boarding + Seat.comfort + Inflight.entertainment +
##     On.board.service + Leg.room.service + Baggage.handling +
##     Checkin.service + Inflight.service + Cleanliness + Departure.Delay.in.Minute
s +
##     Arrival.Delay.in.Minutes, family = "binomial", data = train_data_copy)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6966  -0.2130  -0.0471   0.1327   4.4049
##
## Coefficients: (3 not defined because of singularities)
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           9.510e+00  9.961e+03   0.001 0.999238
## GenderMale                            4.641e-02  2.730e-02   1.700 0.089115 .
## Customer.Typedisloyal Customer     -3.354e+00  4.953e-02 -67.719  < 2e-16 ***
## Age                                -2.309e-03  1.017e-03  -2.271 0.023147 *
## Type.of.TravelPersonal Travel      -4.273e+00  5.507e-02 -77.585  < 2e-16 ***
## ClassEco                           -6.296e-01  3.720e-02 -16.923  < 2e-16 ***
## ClassEco Plus                      -8.366e-01  6.048e-02 -13.832  < 2e-16 ***
## Flight.Distance                     7.223e-06  1.535e-05   0.470 0.638010
## Inflight.wifi.service1             -2.402e+01  8.868e+01  -0.271 0.786540
## Inflight.wifi.service2             -2.427e+01  8.868e+01  -0.274 0.784329
## Inflight.wifi.service3             -2.432e+01  8.868e+01  -0.274 0.783935
## Inflight.wifi.service4             -2.276e+01  8.868e+01  -0.257 0.797414
## Inflight.wifi.service5             -1.720e+01  8.868e+01  -0.194 0.846245
## Departure.Arrival.time.convenient1  3.144e-01  9.296e-02   3.382 0.000720 ***
## Departure.Arrival.time.convenient2  4.302e-01  8.959e-02   4.802 1.57e-06 ***
## Departure.Arrival.time.convenient3  2.415e-01  8.631e-02   2.799 0.005134 **
## Departure.Arrival.time.convenient4 -6.774e-01  7.733e-02  -8.761  < 2e-16 ***
## Departure.Arrival.time.convenient5 -9.128e-01  8.491e-02 -10.750  < 2e-16 ***
## Ease.of.Online.booking1             3.064e+00  9.139e-01   3.352 0.000801 ***
## Ease.of.Online.booking2             2.995e+00  9.139e-01   3.277 0.001049 **
## Ease.of.Online.booking3             3.495e+00  9.137e-01   3.825 0.000131 ***
## Ease.of.Online.booking4             4.341e+00  9.134e-01   4.752 2.02e-06 ***
## Ease.of.Online.booking5             3.710e+00  9.138e-01   4.060 4.92e-05 ***
## Gate.location1                     -1.876e+01  6.523e+03  -0.003 0.997705
## Gate.location2                     -1.868e+01  6.523e+03  -0.003 0.997715
## Gate.location3                     -1.885e+01  6.523e+03  -0.003 0.997695
## Gate.location4                     -1.910e+01  6.523e+03  -0.003 0.997663
## Gate.location5                     -1.931e+01  6.523e+03  -0.003 0.997638
## Food.and.drink1                    -3.282e-01  1.745e+00  -0.188 0.850835
## Food.and.drink2                    -4.633e-02  1.745e+00  -0.027 0.978818
## Food.and.drink3                    -1.760e-01  1.744e+00  -0.101 0.919632
## Food.and.drink4                    -1.320e-01  1.745e+00  -0.076 0.939677
## Food.and.drink5                    -2.865e-01  1.745e+00  -0.164 0.869560
## Online.boarding1                   -3.623e+00  9.175e-01  -3.949 7.84e-05 ***
## Online.boarding2                   -3.543e+00  9.174e-01  -3.862 0.000112 ***
## Online.boarding3                   -3.774e+00  9.171e-01  -4.115 3.87e-05 ***
## Online.boarding4                   -2.128e+00  9.168e-01  -2.321 0.020291 *
## Online.boarding5                   -8.786e-01  9.170e-01  -0.958 0.337987
## Seat.comfort1                       2.047e+01  6.523e+03   0.003 0.997496
## Seat.comfort2                       1.995e+01  6.523e+03   0.003 0.997560
## Seat.comfort3                       1.889e+01  6.523e+03   0.003 0.997689
## Seat.comfort4                       1.959e+01  6.523e+03   0.003 0.997603
## Seat.comfort5                       2.044e+01  6.523e+03   0.003 0.997500
## Inflight.entertainment1             3.970e+01  1.515e+03   0.026 0.979101
## Inflight.entertainment2             4.045e+01  1.515e+03   0.027 0.978704
## Inflight.entertainment3             4.129e+01  1.515e+03   0.027 0.978265
## Inflight.entertainment4             4.096e+01  1.515e+03   0.027 0.978438
## Inflight.entertainment5             4.020e+01  1.515e+03   0.027 0.978839
## On.board.service1                  -2.335e+01  4.051e+03  -0.006 0.995402
## On.board.service2                  -2.320e+01  4.051e+03  -0.006 0.995432
## On.board.service3                  -2.267e+01  4.051e+03  -0.006 0.995536
## On.board.service4                  -2.258e+01  4.051e+03  -0.006 0.995553
## On.board.service5                  -2.205e+01  4.051e+03  -0.005 0.995658
## Leg.room.service1                  -2.400e+00  9.579e-01  -2.506 0.012210 *  
## Leg.room.service2                  -2.127e+00  9.574e-01  -2.222 0.026274 *  
## Leg.room.service3                  -2.244e+00  9.572e-01  -2.344 0.019056 *  
## Leg.room.service4                  -1.546e+00  9.573e-01  -1.614 0.106420    
## Leg.room.service5                  -1.384e+00  9.571e-01  -1.446 0.148230    
## Baggage.handling2                  -2.192e-01  7.601e-02  -2.884 0.003925 ** 
## Baggage.handling3                  -8.441e-01  7.099e-02 -11.890  < 2e-16 ***
## Baggage.handling4                  -2.459e-01  6.902e-02  -3.563 0.000366 ***
## Baggage.handling5                   5.155e-01  7.337e-02   7.026 2.12e-12 ***
## Checkin.service1                   -1.426e+00  5.429e-02 -26.262  < 2e-16 ***
## Checkin.service2                   -1.235e+00  5.401e-02 -22.860  < 2e-16 ***
## Checkin.service3                   -7.263e-01  4.346e-02 -16.712  < 2e-16 ***
## Checkin.service4                   -7.456e-01  4.324e-02 -17.243  < 2e-16 ***
## Checkin.service5                           NA         NA      NA       NA    
## Inflight.service1                  -4.820e-01  7.645e-02  -6.304 2.90e-10 ***
## Inflight.service2                  -7.017e-01  6.933e-02 -10.120  < 2e-16 ***
## Inflight.service3                  -1.394e+00  5.729e-02 -24.332  < 2e-16 ***
## Inflight.service4                  -6.947e-01  4.493e-02 -15.460  < 2e-16 ***
## Inflight.service5                          NA         NA      NA       NA    
## Cleanliness1                       -9.970e-01  7.512e-02 -13.273  < 2e-16 ***
## Cleanliness2                       -9.543e-01  7.303e-02 -13.067  < 2e-16 ***
## Cleanliness3                       -4.690e-01  6.144e-02  -7.633 2.30e-14 ***
## Cleanliness4                       -6.023e-01  6.021e-02 -10.004  < 2e-16 ***
## Cleanliness5                               NA         NA      NA       NA    
## Departure.Delay.in.Minutes          4.452e-03  1.260e-03   3.532 0.000412 ***
## Arrival.Delay.in.Minutes           -8.336e-03  1.246e-03  -6.689 2.24e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 142189  on 103903  degrees of freedom
## Residual deviance:  37004  on 103828  degrees of freedom
## AIC: 37156
## 
## Number of Fisher Scoring iterations: 17
```

We can see some variables with low significancy. To improve the robustness of the model we can rebuild it with the variables that have higher significancy.

```
est_mod_1 <- glm(satisfaction ~ Customer.Type + Age + Type.of.Travel + Class +
                  Departure.Arrival.time.convenient + Ease.of.Online.booking +
                  Online.boarding  +
                  Baggage.handling + Checkin.service + Inflight.service +
                  Cleanliness + Departure.Delay.in.Minutes + Arrival.Delay.in.Minut
es , data = train_data_copy, family = "binomial")

summary(est_mod_1)
```

```
## 
## Call:
## glm(formula = satisfaction ~ Customer.Type + Age + Type.of.Travel + 
##     Class + Departure.Arrival.time.convenient + Ease.of.Online.booking + 
##     Online.boarding + Baggage.handling + Checkin.service + Inflight.service + 
##     Cleanliness + Departure.Delay.in.Minutes + Arrival.Delay.in.Minutes, 
##     family = "binomial", data = train_data_copy)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.3643  -0.3607  -0.0818   0.2713   4.3324  
## 
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                        -2.263e+01  3.682e+02  -0.061   0.9510    
## Customer.Typedisloyal Customer     -2.707e+00  3.581e-02 -75.590  < 2e-16 ***
## Age                                -3.452e-03  7.969e-04  -4.332 1.48e-05 ***
## Type.of.TravelPersonal Travel      -3.706e+00  3.911e-02 -94.744  < 2e-16 ***
## ClassEco                           -5.346e-01  2.663e-02 -20.077  < 2e-16 ***
## ClassEco Plus                      -7.382e-01  4.418e-02 -16.710  < 2e-16 ***
## Departure.Arrival.time.convenient1  4.799e-01  6.775e-02   7.084 1.40e-12 ***
## Departure.Arrival.time.convenient2  6.014e-01  6.558e-02   9.170  < 2e-16 ***
## Departure.Arrival.time.convenient3  4.334e-01  6.401e-02   6.771 1.28e-11 ***
## Departure.Arrival.time.convenient4 -5.023e-01  5.767e-02  -8.710  < 2e-16 ***
## Departure.Arrival.time.convenient5 -7.054e-01  6.041e-02 -11.677  < 2e-16 ***
## Ease.of.Online.booking1            -2.927e+00  9.280e-02 -31.536  < 2e-16 ***
## Ease.of.Online.booking2            -3.152e+00  9.074e-02 -34.733  < 2e-16 ***
## Ease.of.Online.booking3            -2.860e+00  8.942e-02 -31.979  < 2e-16 ***
## Ease.of.Online.booking4            -1.542e+00  8.567e-02 -17.996  < 2e-16 ***
## Ease.of.Online.booking5            -1.113e+00  8.748e-02 -12.719  < 2e-16 ***
## Online.boarding1                   -1.216e+00  1.002e-01 -12.134  < 2e-16 ***
## Online.boarding2                   -1.442e+00  9.885e-02 -14.593  < 2e-16 ***
## Online.boarding3                   -1.640e+00  9.744e-02 -16.833  < 2e-16 ***
## Online.boarding4                    3.939e-01  9.612e-02   4.098 4.16e-05 ***
## Online.boarding5                    2.314e+00  9.931e-02  23.298  < 2e-16 ***
## Baggage.handling2                  -1.118e-01  5.767e-02  -1.938   0.0526 .  
## Baggage.handling3                  -2.329e-01  5.395e-02  -4.316 1.59e-05 ***
## Baggage.handling4                   6.043e-01  5.253e-02  11.505  < 2e-16 ***
## Baggage.handling5                   1.280e+00  5.590e-02  22.897  < 2e-16 ***
## Checkin.service1                   -2.582e-01  3.367e+02  -0.001   0.9994    
## Checkin.service2                   -1.172e-01  3.367e+02   0.000   0.9997    
## Checkin.service3                    3.306e-01  3.367e+02   0.001   0.9992    
## Checkin.service4                    2.925e-01  3.367e+02   0.001   0.9993    
## Checkin.service5                    1.022e+00  3.367e+02   0.003   0.9976    
## Inflight.service1                   1.299e+01  1.735e+02   0.075   0.9403    
## Inflight.service2                   1.299e+01  1.735e+02   0.075   0.9403    
## Inflight.service3                   1.283e+01  1.735e+02   0.074   0.9411    
## Inflight.service4                   1.372e+01  1.735e+02   0.079   0.9370    
## Inflight.service5                   1.433e+01  1.735e+02   0.083   0.9342    
## Cleanliness1                        1.163e+01  8.905e+01   0.131   0.8961    
## Cleanliness2                         1.185e+01  8.905e+01   0.133   0.8942    
## Cleanliness3                         1.217e+01  8.905e+01   0.137   0.8913    
## Cleanliness4                         1.239e+01  8.905e+01   0.139   0.8894    
## Cleanliness5                         1.282e+01  8.905e+01   0.144   0.8855    
## Departure.Delay.in.Minutes           4.450e-03  1.054e-03   4.223 2.41e-05 ***
## Arrival.Delay.in.Minutes            -8.637e-03  1.042e-03  -8.286  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 142189  on 103903  degrees of freedom
## Residual deviance:  54729  on 103862  degrees of freedom
## AIC: 54813
## 
## Number of Fisher Scoring iterations: 11
```

The power of the model is already looking much better.

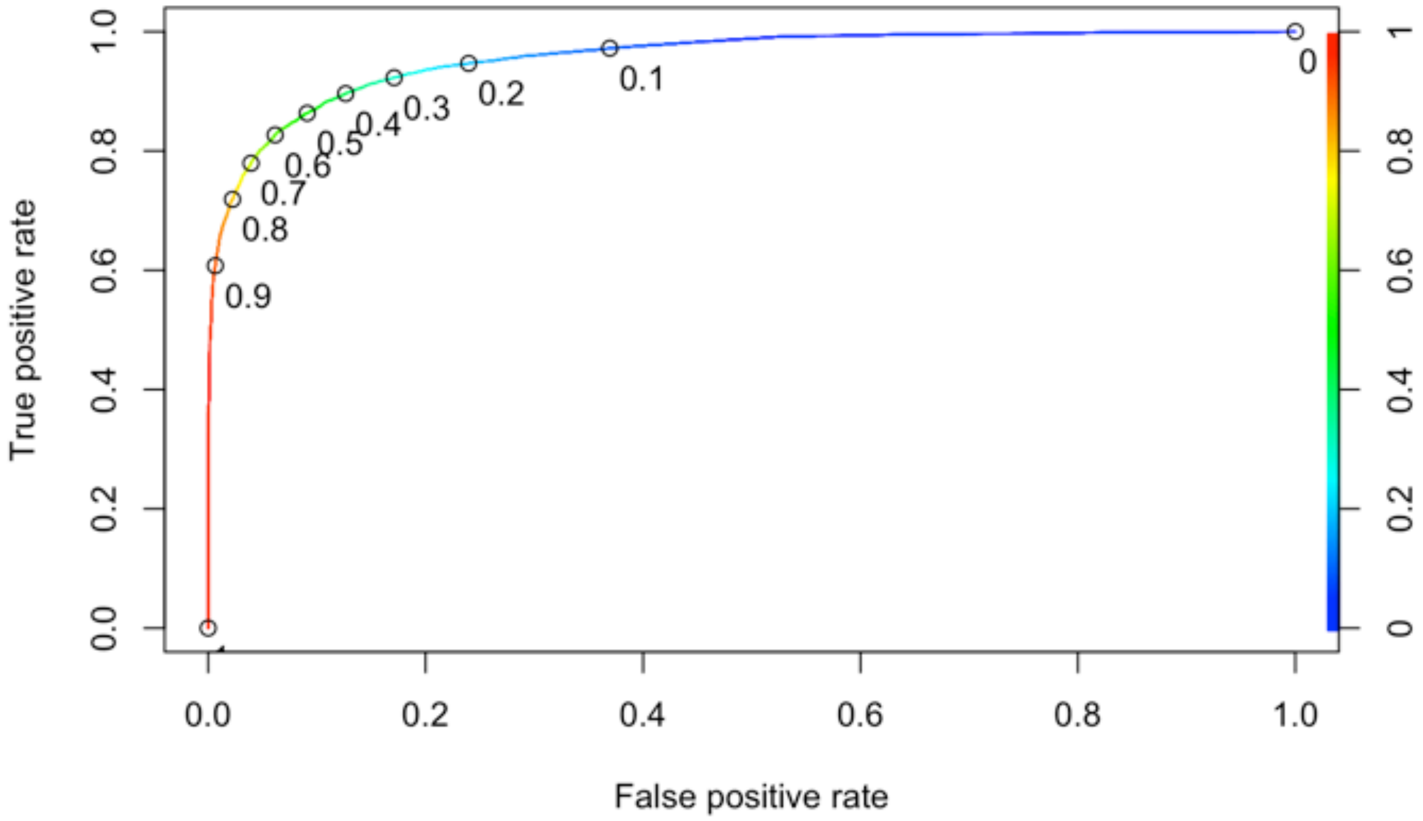
```
predict <- predict(est_mod_1, type = 'response' , newdata=test_data_copy)

summary(predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 0.0000002 0.0254417 0.3020626 0.4402136 0.9236887 0.9998681
```

ROC curve. The ROC curve plots sensitivity (TPR) versus 1 - specificity or the false positive rate (FPR). It gives us an idea of the trade-offs to make when choosing a cutoff for prediction. In this analysis we are going to benefit accuracy.

```
ROCRpred <- prediction(predict, test_data_copy$satisfaction)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, print.cutoffs.at=seq(0,1,by=0.1),text.adj = c(-0.2,
1.7))
```



The area under the curve (AUC) of the ROC plot is:

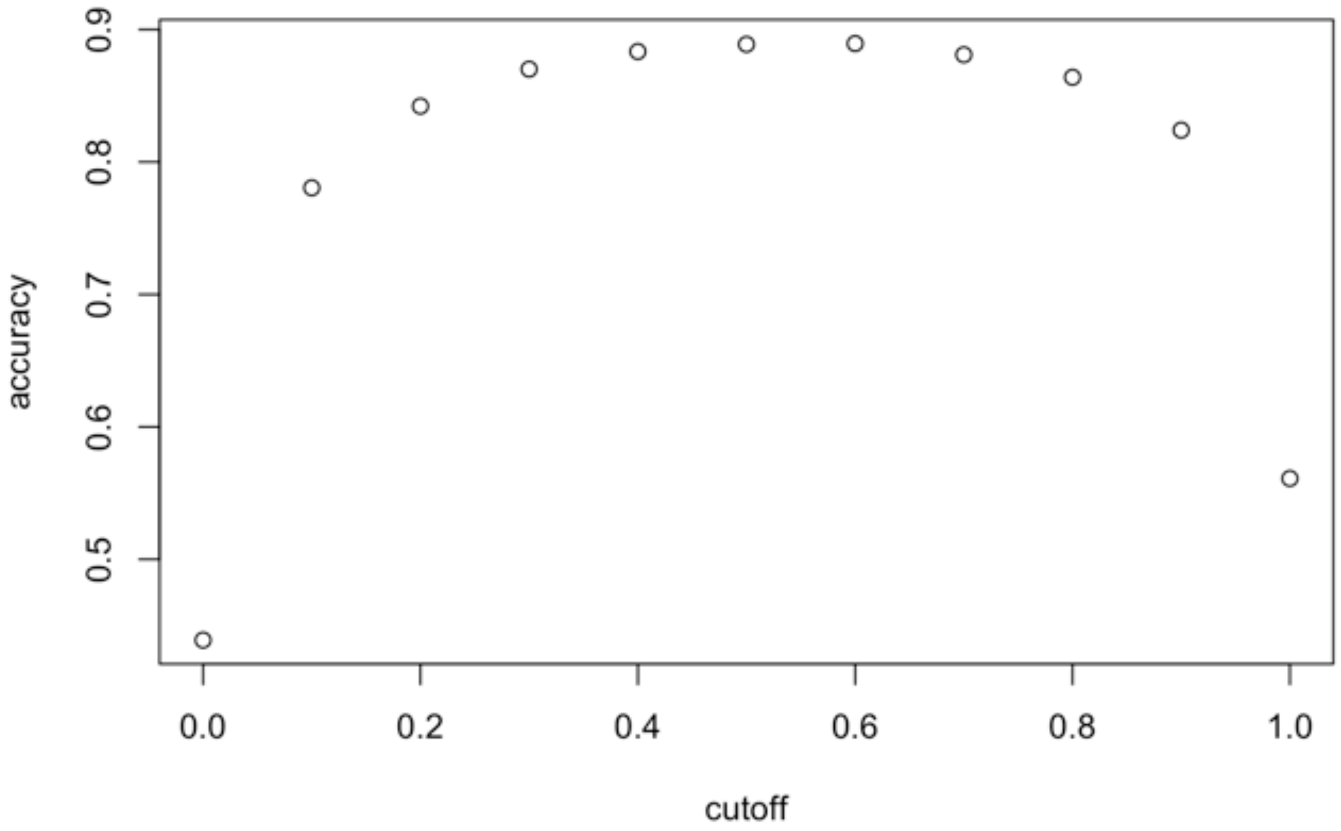
```
AUC <- as.numeric(performance(ROCRpred, "auc")@y.values)
AUC
```

```
## [1] 0.9564346
```

In short, this measure ranging from 0 to 1, shows how well the classification model is performing in general, where the higher the number the better.

```
cutoff <- seq(0,1,.1)
accuracy <- map_dbl(cutoff, function(x){
    y_hat <- ifelse(as.numeric(predict) > x,"satisfied", "neutral or dissatisfied")
    mean(y_hat == test_data_copy$satisfaction) })

plot(cutoff,accuracy)
```



```
max(accuracy)
```

```
## [1] 0.8893979
```

```
best_cutoff <- cutoff[which.max(accuracy)]
best_cutoff
```

```
## [1] 0.6
```

From the ROC Curve, we found 0.6 is the optimum threshold value for Cut-off.

Confussion Matrix

```
y_hat <- ifelse(as.numeric(predict) > best_cutoff,"satisfied", "neutral or dissatis
fied")
cm <- confusionMatrix(data = as.factor(y_hat), reference = test_data_copy$satisfact
ion)

cm$overall["Accuracy"]
```

```
##  Accuracy
## 0.8893979
```

```
cm$byClass[c("F1","Sensitivity","Specificity","Prevalence")]
```

```
##          F1 Sensitivity Specificity  Prevalence
##   0.9049462   0.9384478   0.8267123   0.5610179
```

Decision tree model:

We want to compare the logistic regression model with a decision tree model, looking at what model performs best overall.

```
if (!require(rpart)) install.packages('rpart')
```

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

```
library(rpart)
```

```
tree <- rpart(satisfaction ~ Gender + Customer.Type + Age +
               Type.of.Travel + Class + Flight.Distance + Inflight.wifi.service +
               Departure.Arrival.time.convenient + Ease.of.Online.booking +
               Gate.location + Food.and.drink + Online.boarding + Seat.comfort +
               Inflight.entertainment + On.board.service + Leg.room.service +
               Baggage.handling + Checkin.service + Inflight.service +
               Cleanliness + Departure.Delay.in.Minutes + Arrival.Delay.in.Minutes
,
             data = train_data_copy, method = 'class', minbucket=25)
```

Analyzing the importance of the variables in the tree model using varImp function.

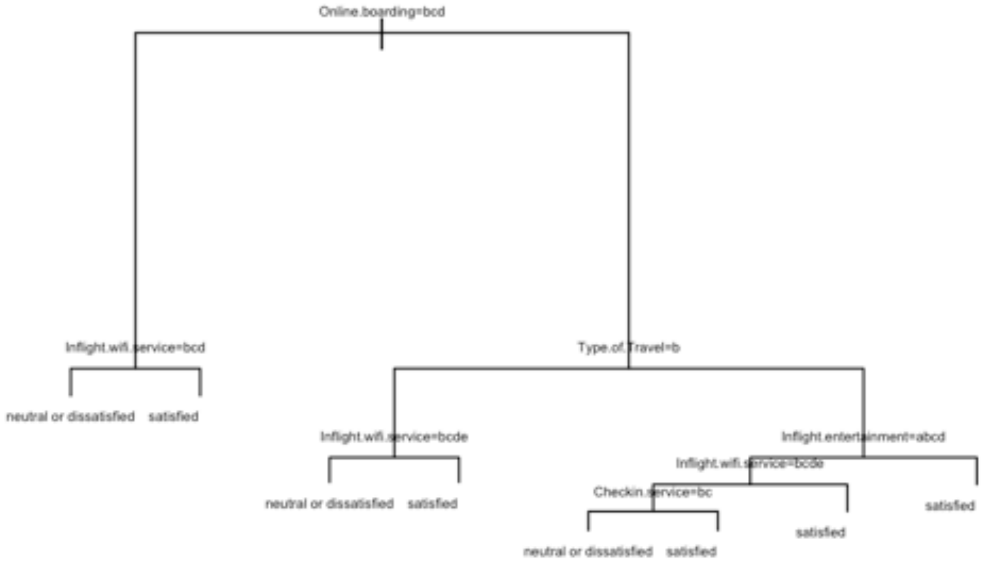
```
varImp(tree)
```

```
##                                       Overall
## Age                                 178.5503
## Arrival.Delay.in.Minutes            184.3497
## Baggage.handling                   1182.1223
## Checkin.service                    2579.5914
## Class                             19424.6440
## Cleanliness                         778.8102
## Ease.of.Online.booking             1821.1027
## Inflight.entertainment            13164.1445
## Inflight.service                   1216.5199
## Inflight.wifi.service             20748.8184
## Leg.room.service                   3784.7539
## On.board.service                   1227.2620
## Online.boarding                   19436.9550
## Seat.comfort                       1045.1440
## Type.of.Travel                    17950.0202
## Gender                                0.0000
## Customer.Type                         0.0000
## Flight.Distance                       0.0000
## Departure.Arrival.time.convenient     0.0000
## Gate.location                         0.0000
## Food.and.drink                        0.0000
## Departure.Delay.in.Minutes            0.0000
```

We rebuid the model with only the variables with higher importance in the model.

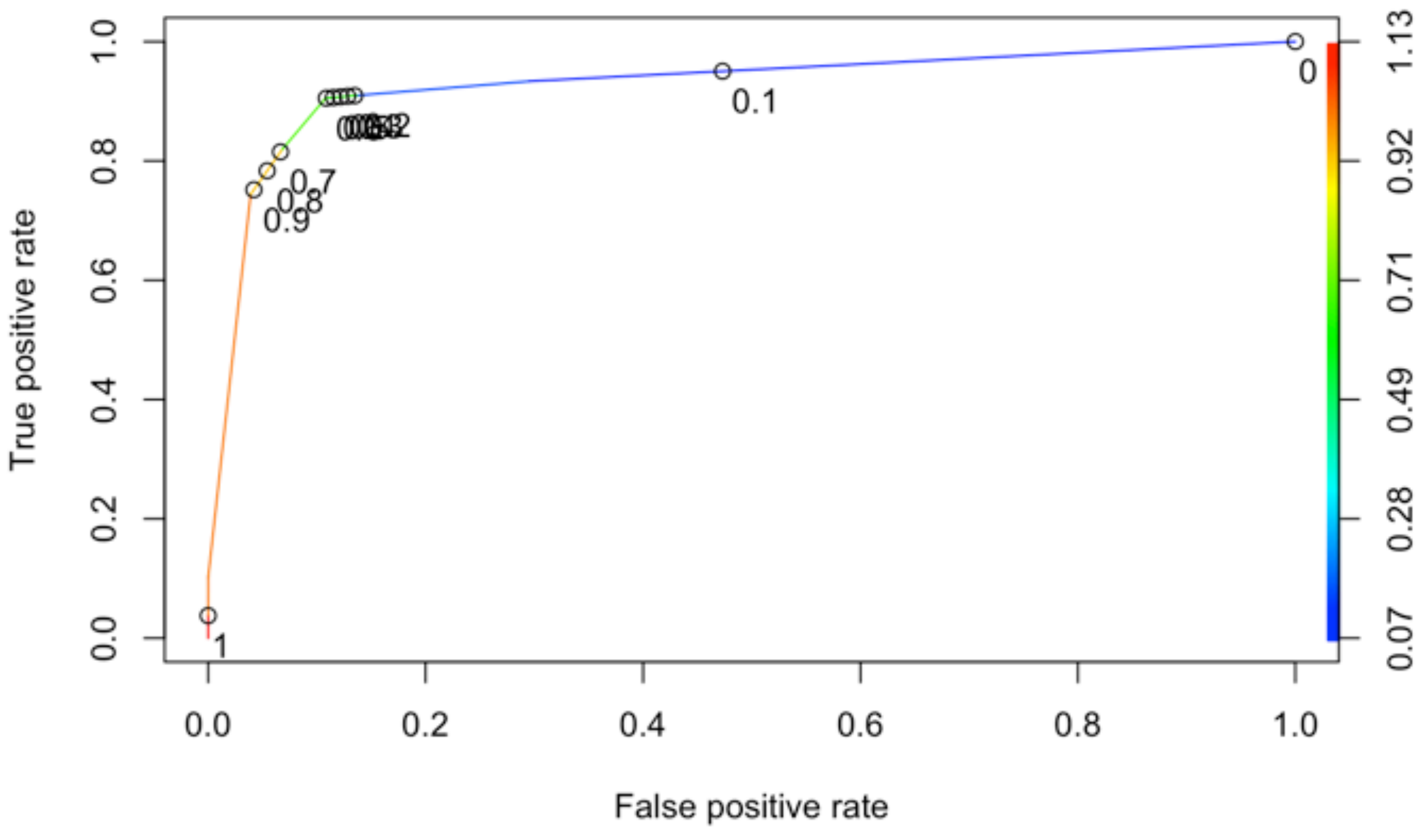
```
tree1 <- rpart(satisfaction ~  Age + Type.of.Travel + Class + Inflight.wifi.service
+
                Ease.of.Online.booking + Online.boarding + Seat.comfort +
               Inflight.entertainment + On.board.service + Leg.room.service +
               Baggage.handling + Checkin.service + Inflight.service +
               Cleanliness + Arrival.Delay.in.Minutes,
               data = train_data_copy, method = 'class', minbucket=25)

plot(tree1, margin = 0.1)
text(tree1, cex = 0.4)
```



```
predict_cart <- predict(tree1, newdata = test_data_copy, class="tree")
```

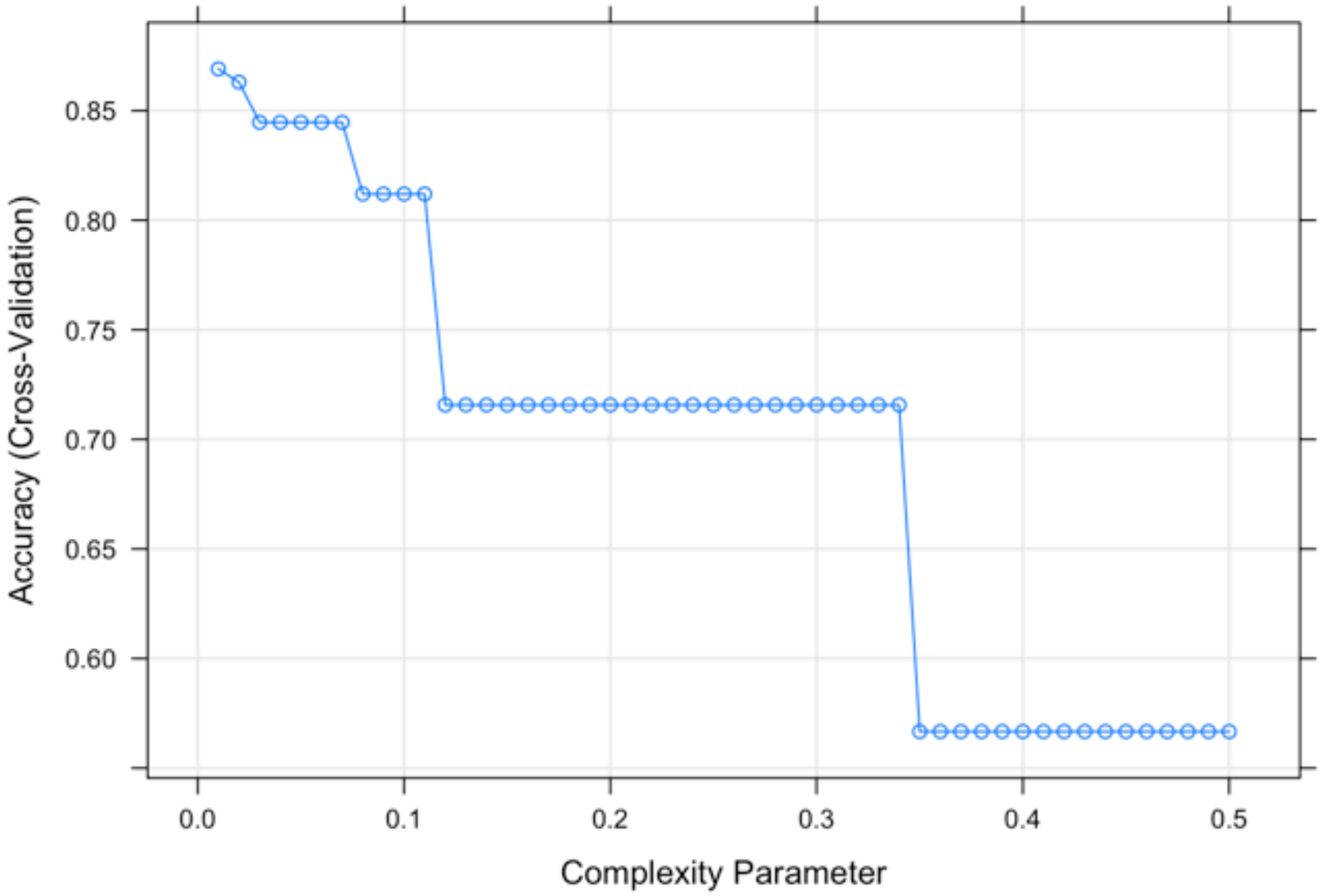
```
#Plotting the ROC Curve
pred <- prediction(predict_cart[,2], test_data_copy$satisfaction)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE, print.cutoffs.at=seq(0,1,by=0.1),text.adj = c(-0.2,1.7)
)
```



```
############################ Optimization Part
# Define cross-validation experiment
numFolds = trainControl( method = "cv", number = 10 )
cpGrid = expand.grid( .cp = seq(0.01,0.5,0.01))
```

```
train_rpart <- train(satisfaction ~  Age + Type.of.Travel + Class + Inflight.wifi.s
ervice +
        Ease.of.Online.booking + Online.boarding + Seat.comfort +
        Inflight.entertainment + On.board.service + Leg.room.service +
        Baggage.handling + Checkin.service + Inflight.service +
        Cleanliness + Arrival.Delay.in.Minutes,
      data = train_data_copy, method = "rpart", trControl = numFolds, tuneGrid = cp
Grid )

plot(train_rpart)
```



```
y_hat_cart <- predict(train_rpart,test_data_copy)
cm_cart <- confusionMatrix(data = as.factor(y_hat_cart), reference = test_data_copy
$satisfaction)

cm_cart$overall["Accuracy"]
```

```
##  Accuracy
## 0.8639128
```

```
cm_cart$byClass[c("F1","Sensitivity","Specificity","Prevalence")]
```

```
##          F1 Sensitivity Specificity  Prevalence
##   0.8865059   0.9473684   0.7572569   0.5610179
```

Very often it is useful to have a single number as a summary of performace, for example for optimization purposes when we don't want to work wih many objective functions. One metric that is preferred over overall accuracy is an average of specificity and sensitivity, referred to as balanced accuracy. Because specificity and sensitivity are rates, it is more appropriate to compute the harmonic average. In fact, the F1-score, a widely used one-number summary, is the harmonic average of precision and recall.

Looking at the F1 meassure of the models we can see that the logistic regression performs better.

The F1 meassure for logistic regression is:

```
cm$byClass["F1"]
```

```
##        F1
## 0.9049462
```

The F1 meassure for the tree model is:

```
cm_cart$byClass["F1"]
```

```
##        F1
## 0.8865059
```