

Airline Passenger Satisfaction Classification

Süleyman Erim, Giacomo Schiavo, Mattia Varagnolo

30.05.2023

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# import libraries
```

```
library(tidyverse)
```

```
library(knitr)
```

```
# import datasets
```

```
data_train = read.csv("train.csv")
```

```
data_test = read.csv("test.csv")
```

```
# merge train and test data
```

```
data = rbind(data_train, data_test)
```

```
attach(data)
```

```
# dimension of data
```

```
print(dim(data))
```

```
## [1] 129880    25
```

```
##           X           id           Gender           Customer.Type
## Min.      :    0   Min.      :    1   Length:129880   Length:129880
## 1st Qu.: 16235   1st Qu.: 32471   Class :character   Class :character
## Median : 38964   Median : 64941   Mode  :character   Mode  :character
## Mean      : 44159   Mean      : 64941
## 3rd Qu.: 71433   3rd Qu.: 97410
## Max.      :103903   Max.      :129880
##
##           Age           Type.of.Travel           Class           Flight.Distance
## Min.      : 7.00   Length:129880   Length:129880   Min.      : 31
## 1st Qu.:27.00   Class :character   Class :character   1st Qu.: 414
## Median :40.00   Mode  :character   Mode  :character   Median : 844
## Mean      :39.43
## 3rd Qu.:51.00
## Max.      :85.00
##                                     Mean      :1190
##                                     3rd Qu.:1744
##                                     Max.      :4983
```

```
##
## Inflight.wifi.service Departure.Arrival.time.convenient Ease.of.Online.booking
## Min. :0.000 Min. :0.000 Min. :0.000
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000
## Median :3.000 Median :3.000 Median :3.000
## Mean :2.729 Mean :3.058 Mean :2.757
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000
## Max. :5.000 Max. :5.000 Max. :5.000
##
## Gate.location Food.and.drink Online.boarding Seat.comfort
## Min. :0.000 Min. :0.000 Min. :0.000 Min. :0.000
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000
## Median :3.000 Median :3.000 Median :3.000 Median :4.000
## Mean :2.977 Mean :3.205 Mean :3.253 Mean :3.441
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:5.000
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000
##
## Inflight.entertainment On.board.service Leg.room.service Baggage.handling
## Min. :0.000 Min. :0.000 Min. :0.000 Min. :1.000
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:3.000
## Median :4.000 Median :4.000 Median :4.000 Median :4.000
## Mean :3.358 Mean :3.383 Mean :3.351 Mean :3.632
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:5.000
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000
##
## Checkin.service Inflight.service Cleanliness Departure.Delay.in.Minutes
## Min. :0.000 Min. :0.000 Min. :0.000 Min. : 0.00
## 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:2.000 1st Qu.: 0.00
## Median :3.000 Median :4.000 Median :3.000 Median : 0.00
## Mean :3.306 Mean :3.642 Mean :3.286 Mean : 14.71
## 3rd Qu.:4.000 3rd Qu.:5.000 3rd Qu.:4.000 3rd Qu.: 12.00
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :1592.00
##
## Arrival.Delay.in.Minutes satisfaction
## Min. : 0.00 Length:129880
## 1st Qu.: 0.00 Class :character
## Median : 0.00 Mode :character
## Mean : 15.09
## 3rd Qu.: 13.00
## Max. :1584.00
## NA's :393
```

```
# replace dots with underscores in column names
names(data) = gsub("\\.", "_", names(data))
```

```
# print column names
print(names(data))
```

```
## [1] "X" "id"
## [3] "Gender" "Customer_Type"
## [5] "Age" "Type_of_Travel"
## [7] "Class" "Flight_Distance"
## [9] "Inflight_wifi_service" "Departure_Arrival_time_convenient"
```

```
## [11] "Ease_of_Online_booking"      "Gate_location"
## [13] "Food_and_drink"              "Online_boarding"
## [15] "Seat_comfort"                "Inflight_entertainment"
## [17] "On_board_service"            "Leg_room_service"
## [19] "Baggage_handling"            "Checkin_service"
## [21] "Inflight_service"            "Cleanliness"
## [23] "Departure_Delay_in_Minutes"  "Arrival_Delay_in_Minutes"
## [25] "satisfaction"
```

```
# drop X and id column
#TODO: explain why
data = data %>% select(-X, -id)
```

```
# insert all categorical variables into a list
```

```
categorical_var = c(data$Cleanliness,
  data$Inflight_wifi_service,
  data$Departure_Arrival_time_convenient,
  data$Ease_of_Online_booking,
  data$Gate_location,
  data$Food_and_drink,
  data$Online_boarding,
  data$Seat_comfort,
  data$Inflight_entertainment,
  data$On_board_service,
  data$Leg_room_service,
  data$Baggage_handling,
  data$Checkin_service,
  data$Inflight_service,
  data$Cleanliness
)
```

```
# convert categorical variables to factors and then to numeric
```

```
categorical_var = as.factor(categorical_var)
categorical_var = as.numeric(categorical_var)
```

```
# convert gender to numeric and then to factor
```

```
data$Gender = as.numeric(as.factor(data$Gender))
```

```
# change type of customer to 0 and disloyal customer to 1
```

```
data$Customer_Type = as.numeric(factor(data$Customer_Type, levels = c("Loyal Customer", "disloyal Customer")))
```

```
# change type of travel to 0 and personal travel to 1
```

```
data$Type_of_Travel = as.numeric(factor(data$Type_of_Travel, levels = c("Personal Travel", "Business Travel")))
```

```
# change class Business is 2, Eco Plus is 1 and Eco is 0
```

```
data$Class = as.numeric(factor(data$Class, levels = c("Business", "Eco Plus", "Eco")) - 1)
```

```
data$satisfaction = as.numeric(factor(data$satisfaction, levels = c("neutral or dissatisfied", "satisfied")))
```

```
# drop na values in Arrival Delay in Minutes
```

```
# TODO: explain why (now it's dropped to simplify the analysis)
```

```
data = data %>% drop_na(Arrival_Delay_in_Minutes)
```

```
# DATA BALANCE: quite balanced
kable(prop.table(table(data$satisfaction)),caption="dataset distribution",col.names = c("Satisfaction",
```

Table 1: dataset distribution

Satisfaction	Frequency
0	0.5655008
1	0.4344992

```
# Train-test split
set.seed(123)
train_index = sample(1:nrow(data), 0.8*nrow(data))
# 80% of data is used for training
train = data[train_index,]
# 20% of data is used for testing
test = data[-train_index,]

# print dimension of train and test data
dim(train)
```

```
## [1] 103589    23
```

```
dim(test)
```

```
## [1] 25898    23
```

```
# save true values of test satisfaction column
test_true = test$satisfaction
```

```
# drop satisfaction column from test data
test = test %>% select(-satisfaction)
```

```
# print proportion of satisfied and dissatisfied customers in train and test data
prop.table(table(train$satisfaction))
```

```
##
##          0          1
## 0.5668845 0.4331155
```

```
prop.table(table(test_true))
```

```
## test_true
##          0          1
## 0.559966 0.440034
```

DATA ANALYSIS

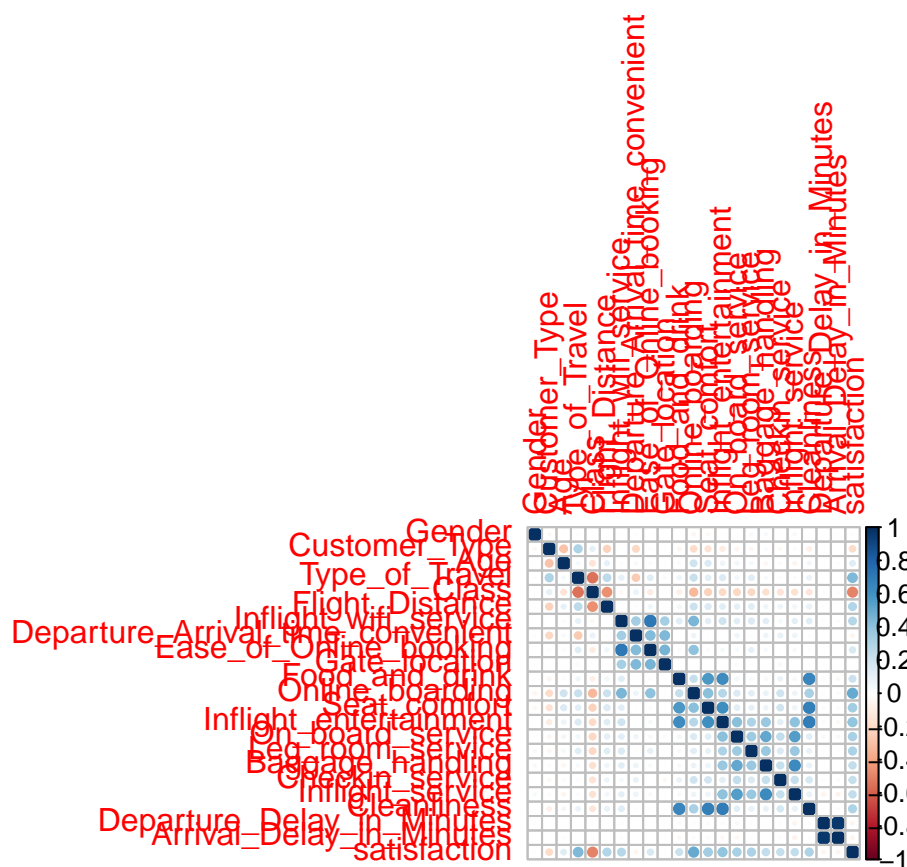
```
# save satisfaction column of train data
train_satisfaction = train$satisfaction
```

```
# correlation matrix only for numeric variables
correlation_matrix = cor(train[, sapply(train, is.numeric)])
```

```
# plot correlation matrix
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(correlation_matrix, method = "circle")
```



train

LOGISTIC REGRESSION

```
# Model definition:
glm_compl<- glm(data = train,
                 satisfaction ~ .,
                 family = "binomial")
# We compute the reference level R-Squared
```

```
s<- summary(glm_compl)
r2<- 1 - (s$deviance/s$null.deviance)
1/(1-r2)
```

```
## [1] 2.045321
```

```
# VIF Iteration 1
# Using the VIF function and comparing the obtained values with the
# computed quantity:
# (The process is done iteratively where we delete one variable at time)
library(regclass)
```

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

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

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

```
## Loading required package: stats4
```

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

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

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

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
```

```
## Attaching package: 'randomForest'
```

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

```
##
```

```
##      combine
```

```
## The following object is masked from 'package:ggplot2':
```

```
##
```

```
##      margin
```

```
## Important regclass change from 1.3:
```

```
## All functions that had a . in the name now have an _
```

```
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```

vif_values <- VIF(glm_compl)

# Create a data frame with variable names and their corresponding VIF values
vif_df <- data.frame(Variable = names(vif_values), VIF = vif_values, row.names = NULL)

# Sort the data frame in decreasing order of VIF values
sorted_df <- vif_df[order(-vif_df$VIF), ]

# Print the sorted data frame
print(sorted_df)

```

```

##              Variable      VIF
## 22  Arrival_Delay_in_Minutes 14.214669
## 21  Departure_Delay_in_Minutes 14.179713
## 14    Inflight_entertainment  3.256814
##  9    Ease_of_Online_booking  2.605357
## 20              Cleanliness  2.466629
##  7    Inflight_wifi_service  2.227467
## 13            Seat_comfort  2.050528
## 11          Food_and_drink  2.019669
## 19          Inflight_service  2.011267
##  4          Type_of_Travel  1.846157
## 17          Baggage_handling  1.818810
##  8  Departure_Arrival_time_convenient 1.716303
## 15          On_board_service  1.635092
##  2          Customer_Type  1.593352
##  5              Class  1.580339
## 10          Gate_location  1.523870
## 12          Online_boarding  1.492857
##  6          Flight_Distance  1.321514
## 16          Leg_room_service  1.218411
## 18          Checkin_service  1.208941
##  3              Age  1.180740
##  1              Gender  1.007157

```

```

# VIF Iteration 2
# Using the VIF function and comparing the obtained values with the
# computed quantity:
# (The process is done iteratively where we delete one variable at time)
# Model definition:
glm_compl<- glm(data = train,
                satisfaction ~ .-Arrival_Delay_in_Minutes,
                family = "binomial")
vif_values <- VIF(glm_compl)

# Create a data frame with variable names and their corresponding VIF values
vif_df <- data.frame(Variable = names(vif_values), VIF = vif_values, row.names = NULL)

# Sort the data frame in decreasing order of VIF values
sorted_df <- vif_df[order(-vif_df$VIF), ]

```

```
# Print the sorted data frame
print(sorted_df)
```

```
##              Variable      VIF
## 14      Inflight_entertainment 3.253083
## 9       Ease_of_Online_booking 2.604730
## 20              Cleanliness 2.462982
## 7       Inflight_wifi_service 2.226307
## 13              Seat_comfort 2.050264
## 11              Food_and_drink 2.015772
## 19              Inflight_service 2.010121
## 4              Type_of_Travel 1.844680
## 17              Baggage_handling 1.819082
## 8  Departure_Arrival_time_convenient 1.717985
## 15              On_board_service 1.635354
## 2              Customer_Type 1.591163
## 5              Class 1.580708
## 10              Gate_location 1.524352
## 12              Online_boarding 1.491323
## 6              Flight_Distance 1.321942
## 16              Leg_room_service 1.218441
## 18              Checkin_service 1.208792
## 3              Age 1.180280
## 21      Departure_Delay_in_Minutes 1.019189
## 1              Gender 1.007198
```

```
# VIF Iteration 3
# Using the VIF function and comparing the obtained values with the
# computed quantity:
# (The process is done iteratively where we delete one variable at time)
# Model definition:
glm_compl<- glm(data = train,
               satisfaction ~ .-Arrival_Delay_in_Minutes-Inflight_entertainment,
               family = "binomial")
vif_values <- VIF(glm_compl)

# Create a data frame with variable names and their corresponding VIF values
vif_df <- data.frame(Variable = names(vif_values), VIF = vif_values,row.names = NULL)

# Sort the data frame in decreasing order of VIF values
sorted_df <- vif_df[order(-vif_df$VIF), ]

# Print the sorted data frame
print(sorted_df)
```

```
##              Variable      VIF
## 9       Ease_of_Online_booking 2.597679
## 7       Inflight_wifi_service 2.179638
## 19              Cleanliness 2.069021
## 13              Seat_comfort 1.921144
## 18              Inflight_service 1.871872
## 4              Type_of_Travel 1.808991
## 16              Baggage_handling 1.777057
```



```
## 11          Food_and_drink 1.749368
## 8  Departure_Arrival_time_convenient 1.714919
## 5          Class 1.568851
## 2          Customer_Type 1.542831
## 14         On_board_service 1.529663
## 10         Gate_location 1.524519
## 12         Online_boarding 1.473553
## 6          Flight_Distance 1.321644
## 15         Leg_room_service 1.203573
## 3          Age 1.175796
## 17         Checkin_service 1.168239
## 20         Departure_Delay_in_Minutes 1.017509
## 1          Gender 1.005687
```

```
# VIF Iteration 4
# Using the VIF function and comparing the obtained values with the
# computed quantity:
# (The process is done iteratively where we delete one variable at time)
# Model definition:
glm_compl<- glm(data = train,
               satisfaction ~ .-Arrival_Delay_in_Minutes-Inflight_entertainment
               -Ease_of_Online_booking ,
               family = "binomial")
vif_values <- VIF(glm_compl)

# Create a data frame with variable names and their corresponding VIF values
vif_df <- data.frame(Variable = names(vif_values), VIF = vif_values,row.names = NULL)

# Sort the data frame in decreasing order of VIF values
sorted_df <- vif_df[order(-vif_df$VIF), ]

# Print the sorted data frame
print(sorted_df)
```

```
##          Variable      VIF
## 18      Cleanliness 2.057085
## 12      Seat_comfort 1.907140
## 17      Inflight_service 1.872313
## 4       Type_of_Travel 1.786666
## 15      Baggage_handling 1.777868
## 10      Food_and_drink 1.747112
## 8  Departure_Arrival_time_convenient 1.583838
## 5          Class 1.564422
## 7      Inflight_wifi_service 1.550276
## 13      On_board_service 1.529146
## 2          Customer_Type 1.520480
## 11      Online_boarding 1.413533
## 9          Gate_location 1.393553
## 6          Flight_Distance 1.321174
## 14      Leg_room_service 1.198613
## 3          Age 1.168891
## 16      Checkin_service 1.166815
## 19      Departure_Delay_in_Minutes 1.016333
## 1          Gender 1.004951
```

```

# VIF Iteration 5
# Using the VIF function and comparing the obtained values with the
# computed quantity:
# (The process is done iteratively where we delete one variable at time)
# Model definition:
glm_compl<- glm(data = train,
                satisfaction ~ .-Arrival_Delay_in_Minutes-Inflight_entertainment
                -Ease_of_Online_booking - Cleanliness ,
                family = "binomial")
vif_values <- VIF(glm_compl)

# Create a data frame with variable names and their corresponding VIF values
vif_df <- data.frame(Variable = names(vif_values), VIF = vif_values,row.names = NULL)

# Sort the data frame in decreasing order of VIF values
sorted_df <- vif_df[order(-vif_df$VIF), ]

# Print the sorted data frame
print(sorted_df)

```

```

##              Variable      VIF
## 17      Inflight_service 1.881686
## 15      Baggage_handling 1.787458
## 4       Type_of_Travel 1.776955
## 8  Departure_Arrival_time_convenient 1.587989
## 12              Seat_comfort 1.585879
## 5              Class 1.563525
## 7      Inflight_wifi_service 1.551518
## 13      On_board_service 1.534811
## 2              Customer_Type 1.512988
## 10      Food_and_drink 1.431732
## 11      Online_boarding 1.408212
## 9       Gate_location 1.401531
## 6      Flight_Distance 1.321957
## 14      Leg_room_service 1.202873
## 3              Age 1.166393
## 16      Checkin_service 1.164245
## 18      Departure_Delay_in_Minutes 1.013281
## 1              Gender 1.004409

```

```

glm_compl<- glm(data = train,
                satisfaction ~ .-Arrival_Delay_in_Minutes-Inflight_entertainment
                -Ease_of_Online_booking - Cleanliness ,
                family = "binomial")
# Observation of the model summary:
summary(glm_compl)

```

```

##
## Call:
## glm(formula = satisfaction ~ . - Arrival_Delay_in_Minutes - Inflight_entertainment -
##      Ease_of_Online_booking - Cleanliness, family = "binomial",
##      data = train)

```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -8.823e+00  8.653e-02 -101.973 < 2e-16 ***
## Gender          7.017e-02  1.929e-02   3.638 0.000275 ***
## Customer_Type  -2.047e+00  2.906e-02 -70.452 < 2e-16 ***
## Age            -8.049e-03  7.022e-04 -11.463 < 2e-16 ***
## Type_of_Travel  2.757e+00  3.074e-02  89.667 < 2e-16 ***
## Class          -3.243e-01  1.264e-02 -25.653 < 2e-16 ***
## Flight_Distance -2.802e-06  1.103e-05  -0.254 0.799509
## Inflight_wifi_service 3.202e-01  9.479e-03  33.776 < 2e-16 ***
## Departure_Arrival_time_convenient -1.682e-01  7.817e-03 -21.520 < 2e-16 ***
## Gate_location    -1.325e-02  8.707e-03  -1.522 0.128025
## Food_and_drink    8.632e-02  8.915e-03   9.683 < 2e-16 ***
## Online_boarding    6.124e-01  9.978e-03  61.379 < 2e-16 ***
## Seat_comfort     1.865e-01  9.774e-03  19.080 < 2e-16 ***
## On_board_service  3.241e-01  9.740e-03  33.278 < 2e-16 ***
## Leg_room_service  2.540e-01  8.403e-03  30.228 < 2e-16 ***
## Baggage_handling  1.522e-01  1.116e-02  13.629 < 2e-16 ***
## Checkin_service   3.329e-01  8.317e-03  40.023 < 2e-16 ***
## Inflight_service   1.445e-01  1.149e-02  12.575 < 2e-16 ***
## Departure_Delay_in_Minutes -4.336e-03  2.619e-04 -16.553 < 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: 141746  on 103588  degrees of freedom
## Residual deviance:  70165  on 103570  degrees of freedom
## AIC: 70203
##
## Number of Fisher Scoring iterations: 5
```

```
glm_compl<- glm(data = train,
                 satisfaction ~ .-Arrival_Delay_in_Minutes-Inflight_entertainment
                 -Ease_of_Online_booking - Cleanliness
                 -Flight_Distance -Gate_location,
                 family = "binomial")
# Observation of the model summary:
summary(glm_compl)
```

```
##
## Call:
## glm(formula = satisfaction ~ . - Arrival_Delay_in_Minutes - Inflight_entertainment -
##      Ease_of_Online_booking - Cleanliness - Flight_Distance -
##      Gate_location, family = "binomial", data = train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -8.8521191  0.0835376 -105.966 < 2e-16 ***
## Gender          0.0701859  0.0192865   3.639 0.000274 ***
## Customer_Type  -2.0469787  0.0281350 -72.755 < 2e-16 ***
## Age            -0.0080418  0.0007009 -11.474 < 2e-16 ***
## Type_of_Travel  2.7515722  0.0303740  90.590 < 2e-16 ***
```

```
## Class -0.3229737 0.0119904 -26.936 < 2e-16 ***
## Inflight_wifi_service 0.3178024 0.0093146 34.119 < 2e-16 ***
## Departure_Arrival_time_convenient -0.1735864 0.0069860 -24.848 < 2e-16 ***
## Food_and_drink 0.0866219 0.0089116 9.720 < 2e-16 ***
## Online_boarding 0.6145055 0.0098663 62.283 < 2e-16 ***
## Seat_comfort 0.1857609 0.0097617 19.030 < 2e-16 ***
## On_board_service 0.3245154 0.0097307 33.350 < 2e-16 ***
## Leg_room_service 0.2543085 0.0083920 30.304 < 2e-16 ***
## Baggage_handling 0.1527092 0.0111576 13.687 < 2e-16 ***
## Checkin_service 0.3332325 0.0083121 40.090 < 2e-16 ***
## Inflight_service 0.1448930 0.0114864 12.614 < 2e-16 ***
## Departure_Delay_in_Minutes -0.0043390 0.0002619 -16.567 < 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: 141746 on 103588 degrees of freedom
## Residual deviance: 70167 on 103572 degrees of freedom
## AIC: 70201
##
## Number of Fisher Scoring iterations: 5
```

```
glm_compl<- glm(data = train,
  satisfaction ~ .-Arrival_Delay_in_Minutes-Inflight_entertainment
  -Ease_of_Online_booking - Cleanliness
  -Flight_Distance -Gate_location,
  family = "binomial")
```

```
# Computing the predictions with the model on the test set:
pred_glm_compl<- predict(glm_compl, test, type = "response")
```

```
# Converting the prediction in {0,1} according to the chosen threshold:
```

```
threshold4<- 0.4
threshold5<- 0.5
threshold6<- 0.6
```

```
pred_glm_compl_04<- ifelse(pred_glm_compl > threshold4, 1, 0)
pred_glm_compl_05<- ifelse(pred_glm_compl > threshold5, 1, 0)
pred_glm_compl_06<- ifelse(pred_glm_compl > threshold6, 1, 0)
```

```
# Confusion matrix with threshold = 0.4
table(test_true, pred_glm_compl_04)
```

```
##      pred_glm_compl_04
## test_true      0      1
##      0 12382  2120
##      1  1543  9853
```

```
mean(pred_glm_compl_04==test_true)
```

```
## [1] 0.8585605
```

```
# Confusion matrix with threshold = 0.5
```

```
table(test_true, pred_glm_compl_05)
```

```
##           pred_glm_compl_05
## test_true      0      1
##           0 13057 1445
##           1  1908 9488
```

```
mean(pred_glm_compl_05==test_true)
```

```
## [1] 0.8705305
```

```
# Confusion matrix with threshold = 0.6
```

```
table(test_true, pred_glm_compl_06)
```

```
##           pred_glm_compl_06
## test_true      0      1
##           0 13574  928
##           1  2381 9015
```

```
mean(pred_glm_compl_06==test_true)
```

```
## [1] 0.8722295
```