Investigation of Reservation Cancellations and Predict Their Probability

Rümeysa KURT

4/1/23

Calling the data set.

The dataset is taken from Kaggle.

```
library(readr)
hotel_bookings <- read_csv("hotel_bookings.csv")
```

1.Detail your task with the problem, features, and target.

Problem

Reservations made at hotels can be canceled for many different reasons. Canceled reservations cause great losses to hotel owners. Here the probability of whether the hotel reservation will be canceled or not will be estimated.

Features

1. hotel (H1 = Resort Hotel or H2 = City Hotel)
2. is_canceled : Value indicating if the booking was canceled (1) or not (0).
3. lead_time : Number of days that elapsed between the entering date of the booking into the PMS and the arrival date.
4. arrival_date_year : Year of arrival date.
5. arrival_date_month : Month of arrival date.
6. arrival_date_week_number : Week number of year for arrival date.
7. arrival_date_day_of_month : Day of arrival date.
8. stays_in_weekend_night : Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel.

9. stays_in_week_nights : Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel.
10. adults : Number of adults.
11. children : Number of children.
12. babies : Number of babies.
13. meal : Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal.
14. country: Country of origin. Categories are represented in the ISO 3155–3:2013 format.
15. market_segment : Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators".
16. distribution_channel : Booking distribution channel. The term "TA" means "Travel Agents" and "TO" means "Tour Operators".
17. is_repeated_guest : Value indicating if the booking name was from a repeated guest (1) or not (0).
18. previous_cancellations : Number of previous bookings that were cancelled by the customer prior to the current booking.
19. previous_bookings_not_canceled : Number of previous bookings not cancelled by the customer prior to the current booking.
20. reserved_room_type : Code of room type reserved. Code is presented instead of designation for anonymity reasons.
21. assigned_room_type : Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due.
22. booking_changes : Number of changes/amendments made to the booking from the moment the booking was entered on the PMS.
23. deposit_type : Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No.
24. agent : ID of the travel agency that made the booking.
25. company : ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for.
26. days_in_waiting_list : Number of days the booking was in the waiting list before it was confirmed to the customer.
27. customer_type : Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of.
28. adr : Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights.
29. required_car_parking_spaces : Number of car parking spaces required by the customer.
30. total_of_special_requests : Number of special requests made by the customer (e.g. twin bed or high floor).
31. reservation_status : Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out.
32. reservation_status_date : Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to.

Target

is_canceled : Value indicating if the booking was canceled (1) or not (0).

2.Describe the dataset in your task in terms of the dimension, variable type, and some other that you want to add.

```
install.packages("DALEX")
install.packages("ROCR")
install.packages("caret")
library(DALEX)
library(ROCR)
library(caret)
```

In this data set we have 4 NA observations.Missing observations were removed because the data set was large enough.

```
hotel_bookings <- na.exclude(hotel_bookings)
```

The `str()` function is used to browse the dataset. In this way, the number of observations in the dataset, the number of features, the type of features, and the dimension of the dataset were reached.

Longitude, latitude, housing median age, total rooms, total bedrooms, population, households, median income, median house value features are numerical, ocean proximity feature is categorical.

The size of the dataset is [119,386 $\times$ 32].

```
str(hotel_bookings)
```

```
tibble [119,386 x 32] (S3: tbl_df/tbl/data.frame)
 $ hotel                         : chr [1:119386] "Resort Hotel" "Resort Hotel" "Resort Hotel
 $ is_canceled                   : num [1:119386] 0 0 0 0 0 0 0 0 1 1 ...
 $ lead_time                     : num [1:119386] 342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : num [1:119386] 2015 2015 2015 2015 2015 ...
 $ arrival_date_month            : chr [1:119386] "July" "July" "July" "July" ...
 $ arrival_date_week_number      : num [1:119386] 27 27 27 27 27 27 27 27 27 27 ...
 $ arrival_date_day_of_month     : num [1:119386] 1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights       : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : num [1:119386] 0 0 1 1 2 2 2 2 3 3 ...
```

```
$ adults                         : num [1:119386] 2 2 1 1 2 2 2 2 2 2 ...
$ children                       : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ babies                         : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ meal                           : chr [1:119386] "BB" "BB" "BB" "BB" ...
$ country                        : chr [1:119386] "PRT" "PRT" "GBR" "GBR" ...
$ market_segment                 : chr [1:119386] "Direct" "Direct" "Direct" "Corporate" ...
$ distribution_channel           : chr [1:119386] "Direct" "Direct" "Direct" "Corporate" ...
$ is_repeated_guest              : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ previous_cancellations         : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ previous_bookings_not_canceled: num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ reserved_room_type             : chr [1:119386] "C" "C" "A" "A" ...
$ assigned_room_type             : chr [1:119386] "C" "C" "C" "A" ...
$ booking_changes                : num [1:119386] 3 4 0 0 0 0 0 0 0 0 ...
$ deposit_type                   : chr [1:119386] "No Deposit" "No Deposit" "No Deposit" "No
$ agent                          : chr [1:119386] "NULL" "NULL" "NULL" "304" ...
$ company                        : chr [1:119386] "NULL" "NULL" "NULL" "NULL" ...
$ days_in_waiting_list           : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ customer_type                  : chr [1:119386] "Transient" "Transient" "Transient" "Transi
$ adr                            : num [1:119386] 0 0 75 75 98 ...
$ required_car_parking_spaces    : num [1:119386] 0 0 0 0 0 0 0 0 0 0 ...
$ total_of_special_requests      : num [1:119386] 0 0 0 0 1 1 0 1 1 0 ...
$ reservation_status             : chr [1:119386] "Check-Out" "Check-Out" "Check-Out" "Check-
$ reservation_status_date        : Date[1:119386], format: "2015-07-01" "2015-07-01" ...
- attr(*, "na.action")= 'exclude' Named int [1:4] 40601 40668 40680 41161
  ..- attr(*, "names")= chr [1:4] "40601" "40668" "40680" "41161"
```

The "country", "agent", "company" and "reservation date" attributes in the dataset have been removed as they have multiple classes. Problematic in model training are categorical features because there are many classes.

```
hotel_bookings <- hotel_bookings[c(-4, -5, -14, -21, -24, -25, -31, -32)]
```

Splitting the data set

The sample() function is used to split the dataset into test and train sets. Using the index object, the observations are set to train and test.

```
set.seed(123) # for reproducibility
index <- sample(1 : nrow(hotel_bookings), round(nrow(hotel_bookings) * 0.80))
train <- hotel_bookings[index, ]
test  <- hotel_bookings[-index, ]
```

The dim() function is used to see how many rows and how many columns are in data sets separated into test and train.

```
dim(test)
```

```
[1] 23877    24
```

```
dim(train)
```

```
[1] 95509    24
```

While there are 23877 observations and 24 features in the test part, there are 95509 observations and 24 features in the train part.

3. Train a logistic regression model

The "glm()" function is used to train a logistic regression model. "Y" is the target variable that deals with features to be predicted and "x"s are features that use the information to predict the target variable. family=binomial defines the target property. We have two levels(or outcome, possibility) here so it's coded as binomial.

```
lr_model <- glm(is_canceled ~ ., family = "binomial" , data=train)
```

```
lr_model
```

```
Call:  glm(formula = is_canceled ~ ., family = "binomial", data = train)

Coefficients:
                  (Intercept)                hotelResort Hotel
                   -2.490e+00                        6.482e-02
                    lead_time         arrival_date_week_number
                    3.979e-03                       -5.513e-04
     arrival_date_day_of_month          stays_in_weekend_nights
                    1.100e-03                        5.267e-02
          stays_in_week_nights                           adults
                    5.217e-02                        1.015e-01
```

```
                     children                         babies
                    1.994e-01                      2.692e-01
                       mealFB                         mealHB
                    6.688e-01                     -1.617e-01
                       mealSC                   mealUndefined
                    1.429e-01                     -4.908e-01
   market_segmentComplementary          market_segmentCorporate
                    6.263e-01                     -2.625e-01
          market_segmentDirect             market_segmentGroups
                    2.118e-02                     -8.470e-02
   market_segmentOffline TA/TO          market_segmentOnline TA
                   -6.919e-01                      6.691e-01
    distribution_channelDirect          distribution_channelGDS
                   -5.936e-01                     -1.079e+00
     distribution_channelTA/TO   distribution_channelUndefined
                   -7.519e-02                      2.789e+01
             is_repeated_guest           previous_cancellations
                   -5.973e-01                      2.795e+00
previous_bookings_not_canceled              reserved_room_typeB
                   -5.097e-01                      5.926e-02
           reserved_room_typeC              reserved_room_typeD
                   -1.227e-02                     -6.738e-02
           reserved_room_typeE              reserved_room_typeF
                    6.639e-02                     -4.317e-01
           reserved_room_typeG              reserved_room_typeH
                   -2.134e-01                     -2.393e-02
           reserved_room_typeL              reserved_room_typeP
                    5.647e-01                      1.572e+01
               booking_changes           deposit_typeNon Refund
                   -3.915e-01                      5.554e+00
        deposit_typeRefundable             days_in_waiting_list
                    1.892e-01                     -4.899e-04
             customer_typeGroup           customer_typeTransient
                   -9.453e-02                      8.431e-01
  customer_typeTransient-Party                             adr
                    4.133e-01                      4.577e-03
   required_car_parking_spaces        total_of_special_requests
                   -4.088e+01                     -7.177e-01

Degrees of Freedom: 95508 Total (i.e. Null);  95463 Residual
Null Deviance:      126000
Residual Deviance: 82160    AIC: 82250
```

You can see more details about the model using the "summary()" function. Here, when interpreting the levels of categorical variables, we can say that if even one of them is significant, the others are also significant.

```
summary(lr_model)
```

```
Call:
glm(formula = is_canceled ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.7387  -0.4226   0.2187   5.7823

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -2.490e+00  1.999e-01 -12.454  < 2e-16 ***
hotelResort Hotel               6.482e-02  2.235e-02   2.899 0.003739 **
lead_time                       3.979e-03  1.035e-04  38.447  < 2e-16 ***
arrival_date_week_number       -5.513e-04  6.715e-04  -0.821 0.411615
arrival_date_day_of_month       1.100e-03  9.777e-04   1.125 0.260607
stays_in_weekend_nights         5.267e-02  9.910e-03   5.315 1.07e-07 ***
stays_in_week_nights            5.217e-02  5.246e-03   9.945  < 2e-16 ***
adults                          1.015e-01  1.716e-02   5.914 3.33e-09 ***
children                        1.994e-01  2.740e-02   7.279 3.36e-13 ***
babies                          2.692e-01  9.950e-02   2.706 0.006816 **
mealFB                          6.688e-01  1.179e-01   5.672 1.41e-08 ***
mealHB                         -1.617e-01  3.016e-02  -5.360 8.33e-08 ***
mealSC                          1.429e-01  2.885e-02   4.954 7.28e-07 ***
mealUndefined                  -4.908e-01  1.092e-01  -4.494 7.00e-06 ***
market_segmentComplementary     6.263e-01  2.485e-01   2.520 0.011726 *
market_segmentCorporate        -2.625e-01  1.945e-01  -1.350 0.177073
market_segmentDirect            2.118e-02  2.156e-01   0.098 0.921762
market_segmentGroups           -8.470e-02  2.035e-01  -0.416 0.677280
market_segmentOffline TA/TO    -6.919e-01  2.041e-01  -3.389 0.000701 ***
market_segmentOnline TA         6.691e-01  2.034e-01   3.289 0.001005 **
distribution_channelDirect     -5.936e-01  1.040e-01  -5.705 1.16e-08 ***
distribution_channelGDS        -1.079e+00  2.262e-01  -4.768 1.86e-06 ***
distribution_channelTA/TO      -7.519e-02  7.791e-02  -0.965 0.334512
distribution_channelUndefined   2.789e+01  8.925e+02   0.031 0.975073
is_repeated_guest              -5.973e-01  9.119e-02  -6.550 5.77e-11 ***
previous_cancellations          2.795e+00  6.624e-02  42.193  < 2e-16 ***
```

```
previous_bookings_not_canceled -5.097e-01  2.832e-02 -18.000  < 2e-16 ***
reserved_room_typeB             5.926e-02  8.421e-02   0.704 0.481621
reserved_room_typeC            -1.227e-02  1.025e-01  -0.120 0.904713
reserved_room_typeD            -6.738e-02  2.455e-02  -2.744 0.006064 **
reserved_room_typeE             6.639e-02  3.959e-02   1.677 0.093584 .
reserved_room_typeF            -4.317e-01  6.473e-02  -6.670 2.55e-11 ***
reserved_room_typeG            -2.134e-01  7.565e-02  -2.821 0.004781 **
reserved_room_typeH            -2.393e-02  1.275e-01  -0.188 0.851102
reserved_room_typeL             5.647e-01  9.429e-01   0.599 0.549240
reserved_room_typeP             1.572e+01  2.723e+02   0.058 0.953950
booking_changes                -3.915e-01  1.711e-02 -22.884  < 2e-16 ***
deposit_typeNon Refund          5.554e+00  1.243e-01  44.695  < 2e-16 ***
deposit_typeRefundable          1.892e-01  2.306e-01   0.820 0.412013
days_in_waiting_list           -4.899e-04  5.454e-04  -0.898 0.369121
customer_typeGroup             -9.453e-02  1.897e-01  -0.498 0.618340
customer_typeTransient          8.431e-01  5.965e-02  14.136  < 2e-16 ***
customer_typeTransient-Party    4.133e-01  6.349e-02   6.510 7.53e-11 ***
adr                             4.577e-03  2.338e-04  19.573  < 2e-16 ***
required_car_parking_spaces    -4.088e+01  1.319e+02  -0.310 0.756603
total_of_special_requests      -7.177e-01  1.284e-02 -55.900  < 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: 125994  on 95508  degrees of freedom
Residual deviance:  82160  on 95463  degrees of freedom
AIC: 82252

Number of Fisher Scoring iterations: 13
```

4. Measuring model performance

In here we use additional argument.(type=response) When we set the type as a response we can calculate the probability of the target feature.

When calculating the probability of the target variable, its actual values were excluded from the test set.

```r
predicted_probs <- predict(lr_model, test[,-2], type = "response")
head(predicted_probs)
```

```
        1         2         3         4         5         6
0.3517356 0.1619818 0.2125263 0.2873658 0.3437163 0.5325684
```

Here, the result is not a probability value because we convert the probabilities into classes or labels. If the values of the estimated probabilities are greater than 0.5, they will be equal to 1, and if they are smaller, they will be equal to 0.

(predicted_probs $>$ 0.5 is a condition)

```r
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)
head(predicted_classes)
```

```
1 2 3 4 5 6
0 0 0 0 0 1
```

According to the values of performance metrics, the model classifies the observations with 0.80 accuracy. Its precision is 0.59 means that the model classify only 59% of the positive class, which is canceled reservations in the problem, correctly.

In that case, we must check the imbalancedness of the classes in the target feature.

```r
TP <- sum(predicted_classes[which(test$is_canceled == "1")] == 1)
FP <- sum(predicted_classes[which(test$is_canceled == "1")] == 0)
TN <- sum(predicted_classes[which(test$is_canceled == "0")] == 0)
FN <- sum(predicted_classes[which(test$is_canceled == "0")] == 1)
```

```r
recall      <- TP / (TP + FN)
specificity <- TN / (TN + FP)
precision   <- TP / (TP + FP)
accuracy    <- (TN + TP) / (TP + FP + TN + FN)

recall
```

```
[1] 0.8362275
```

```r
specificity
```

```
[1] 0.8001136
```

```
precision
```

```
[1] 0.5987225
```

```
accuracy
```

```
[1] 0.8096076
```

In this step, 62% of the train set observation belongs to the class whose reservation has not been canceled, and 37% belongs to the class whose reservation has been cancelled.

So there is imbalanced between these classes. It means that number of observations in this two classes are not equal.In this case is called imbalancedness problem in classification. If we want to solve this problem, we can use the following methods: - Increase the number of less observed class it is also called minority class. - Decrease the number of observation in the majority class.

This may cause the model learn less from the minority class and normally the model couldn't classify correctly the observation in majority class.

```
table(train$is_canceled) / dim(train)[1]
```

```
        0         1
0.6287994 0.3712006
```

The `confusionMatrix()` function in the `{caret}` package is used to calculate the metrics.

When we compare the accuracy(0.8096) and pos pred value(0.5987) we can see that the model does not perform well in both of the classes because pos pred value is not good as in general performance. So it means that there is a sign about imbalanced problem in model.

If there is an imbalance problem in the model, it is suggested to use balanced accuracy instead of accuracy as a general performance matrix.

```
confusionMatrix(table(ifelse(test$is_canceled == "1", "1", "0"),
                      predicted_classes),
                positive = "1")
```

```
Confusion Matrix and Statistics

   predicted_classes
        0     1
  0 14082  1028
  1  3518  5249

               Accuracy : 0.8096
                 95% CI : (0.8046, 0.8146)
    No Information Rate : 0.7371
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5643

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8362
            Specificity : 0.8001
         Pos Pred Value : 0.5987
         Neg Pred Value : 0.9320
             Prevalence : 0.2629
         Detection Rate : 0.2198
   Detection Prevalence : 0.3672
      Balanced Accuracy : 0.8182

       'Positive' Class : 1
```

The following steps were followed to obtain the ROC curve in R.

```r
explain_lr <- explain(model   = lr_model,                  # trained model
                      data    = test[, -2],                # test set without target
                      y       = test$is_canceled == "1",   # observed values of target
                                                           # with reference class
                      type    = "classification",          # type of task
                      verbose = FALSE)                     # remove some messages
```

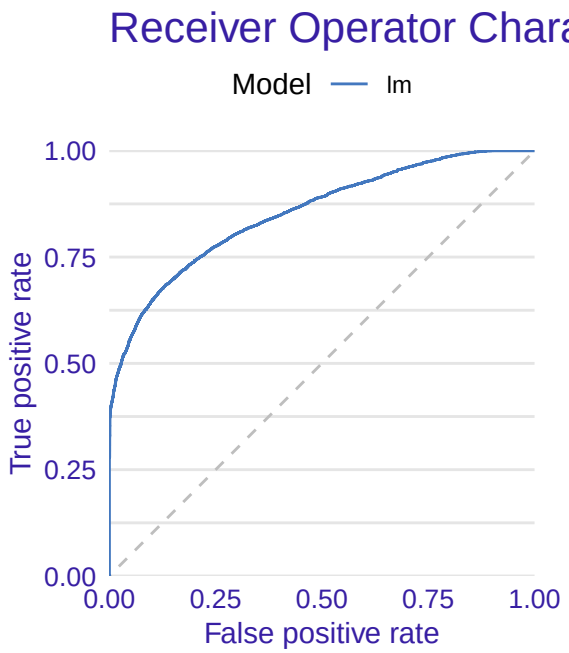
With the explainer object created above, the `model_performance()` function from the `{DALEX}` package is used to plot ROC curves and others.

The ROC curve tells us how good the model is at predicting. It is one of the metrics commonly used to evaluate the performance of machine learning algorithms, especially in situations with imbalanced data sets.

If the ROC curve is close to the reference line it means that model is a random model and it does not perform well.

Here, the ROC curve is close to the left top, we can say that the model performance is excellent.

```
performance_lr <- model_performance(explain_lr)
plot(performance_lr, geom = "roc")
```



Area under the curve (AUC) is used as a statistics to show the summarization of ROC curve.

The `performance_lr` object is used to calculate the area under the curve (AUC). The higher AUC indicates higher performance. The AUC is between 0 and 1. Here the AUC value is 0.85. This shows us that the performance of the model is close to perfect.

```
performance_lr
```

```
Measures for:  classification
recall     : 0.5987225
precision  : 0.8362275
f1         : 0.6978197
accuracy   : 0.8096076
auc        : 0.8534894
```

```
Residuals:
           0%           10%           20%           30%           40%
-1.000000e+00 -3.943172e-01 -2.838372e-01 -2.039740e-01 -1.514084e-01
          50%           60%           70%           80%           90%
-8.828341e-02 -2.220446e-16  5.484042e-03  3.697501e-01  6.305511e-01
         100%
 9.999999e-01
```