Machine Learning I

Hotel California

Grou07

23 December 2022

Alexandra Pinto - 20211599

Francisco Farinha – 20211550

Ilona Nacu – 20211602

João Barradas - 20211590

Rafael Proença - 20211681

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

ABSTRACT

With the evolution of the hotel industry, cancellations had become more frequent. Our team has the goal to develop a predictive model able to predict if a customer will cancel their booking or not. For this, we will follow a proper Machine Learning pipeline. We started by treating and exploring the train and test data, then proceeded to feature selection. After selecting the features, we could start building our models and after trying a variety of different methods, we concluded that the Random Forest Classifier had the best performance for our specific dataset. With this, we focused on this algorithm and tried to boost its performance to the maximum by selecting the best hyperparameters. Those parameters were selected using the Grid Search technique, a technique very effective when it comes to selecting the best parameters for our model. But first, we had to drop two univariate features that we found (ArrivalYear and CompanyReservation) and another twelve that we consider irrelevant when selecting the best features to use in our model (those features are exposed in the results section). The models built were Model_Best and ModelRFC. Model_Best had its parameters chosen by Grid Search and Model_RFC is Model_Best but with some manual parameter adjustments to minimize the overfitting that model presented. The univariate variables don't provide useful information and so they were dropped. Some other discarded features were identified through some feature selection techniques like Pearson and Spearman Correlation, RFE, Lasso, and Chi-squared Test. A wide variety of models were tested with the test data on Kaggle, but the best ones had overfitting, so a Random Forest Classifier model without overfitting was built. The parameters were carefully chosen to reduce model overfitting but still have high performance. With all of this, we could accomplish the task of implementing a Machine Learning pipeline and provide useful results for Hotel California.

KEYWORDS

Random Forest Classifier; Grid Search; Overfitting; Classification

INTRODUCTION

The hotel industry has evolved, and tourism operators, overbooking, and free cancellation policies have become industry standard practices, resulting in more and more frequent cancellations. This presents issues for Hotel California, and its new management. They wish to avoid making the same mistakes the previous owners did and try to get ahead of potential issues.

This is where our team comes in. We were tasked with developing a suitable approach to reduce cancellation vacancies. The data provided to us is a representative sample of 2016 Hotel California bookings. Facing a classification problem, we must create a predictive model for booking cancellations. Our goal is to follow a proper Machine Learning Pipeline, making sure we do all the important steps to treat our data and make sure we only use the most important features, in the correct format, for our final model to be able to achieve the best results possible.

BACKGROUND

To evaluate our models, we make use of some metrics such as f1score and accuracy score, already explored in practical classes. To extend our criteria of evaluating the models we decided to do a confusion matrix and perform Specificity and Sensitivity. A confusion matrix is employed to describe how well a classification system performs. The effectiveness of a classification algorithm is shown and summarized via a confusion matrix.

An outcome where the model properly predicted the positive class is referred to as a true positive. Like a genuine positive, a true negative (TN) is a result for which the model accurately foresees the negative class. A false positive (FP) is a result when the model forecasts the positive class inaccurately. The proportion of non-fraud cases that were correctly identified to all non-fraud cases is known as the true negative (TN), or specificity. When a condition is not present, the test result is a true negative (TN).

A true positive (TP) rate or Sensitivity, that is calculated TP/(TP+FN) is a result when the model forecasts the positive class inaccurately.

The Specificity or the true negative (TN) rate is calculated TN/(TN+FP). The TPR measures the likelihood that a true positive (TP) will indeed test positive.

METHODOLOGY

We will proceed to describe the procedures and the materials used to conduct our project.

We started by exploring our dataset, to get to know it better, such as seeing the data types and then changing the data type of ‘Children’ from float to an integer (Figure 1.), checking for missing values, statistical data (Figure 2.) and seeing the distribution of some of the variables, using histograms. We also check for univariate variables (Figure 3.). We also divided the features depending on if they are continuous, discrete, categorical, or numerical since different techniques can use diverse types of variables, making the process of selecting variables easier. After performing this on the training dataset, we also did it on the test dataset.

Afterward, we divided our dataset into train and validation data, to be able to train our model and test it with validation data, for us to have an idea of how well our model is working in predicting the Cancelled variable and how well it will perform on test data.

We then scaled our data because it is a key step in pre-processing that involves transforming the features into a similar range so that high numeric feature values do not transcend the target variable [1].

During feature selection, we employ a variety of strategies to determine which features to preserve and which to discard.

To determine the non-significant numeric features, we used Spearman and Pearson Correlation Matrix (Figure 4. and Figure 5.), RFE, and Lasso Regression (Figure 6.). For the categorical ones, we used the Mutual Information Criterion and the Chi-square test. Taking into consideration all the methods applied we identified the features to be eliminated and dropped them from the dataset.

With all the steps taken, we can start building our models and tuning their parameters to increase their performance. To build the predictive models we use all the algorithms learned in the practical classes. After finding the best parameters for each model, we observed that the Random Forest Classifier (RFC) was the most suitable algorithm to make the target predictions for our specific dataset since it was the one with the best model performance.

RFC is an ensemble method. Ensemble methods combine several "weak learners" to produce a machine learning model that can outperform each model when used independently. So Random Forest algorithm's operation is based on a collection of trees, where each tree is dependent on a unique set of random variables. It employs a divide-and-conquer strategy. A forest is a grouping of several trees. This approach will produce several little decision trees from random selections of the input data.

Knowing this, we decided to concentrate on the RFC to improve its performance, therefore we used the Grid Search method to determine the best parameters for the RFC model. The Grid Search method is a technique for determining the best classifier parameters so that a model can accurately predict our target variable. A classification model includes numerous parameters and determining the appropriate arrangement of these parameters can be a difficult undertaking. Grid Search is one of the better solutions for these [2].

RESULTS

To reach our goal, which is to predict if a client is prone to cancel a reservation or not, we started to study our data, to see if there were any missing values, univariate features, and some statistical metrics such as mean, standard deviation, minimum and maximum value. In this process, we decided to remove the univariate features, ArrivalYear and CompanyReservation.

Through feature selection and using various models to be more accurate we decided to drop twelve features: ArrivalWeekNumber, ArrivalHour, WeekendStays, Babies, PreviousReservations, PreviousCancellations, DaysUntilConfirmation, FloorReserved, AffiliatedCustomer, DailyRateUSD, CountryofOriginAvgIncomeEuros (Year-2) and CountryofOriginHDI (Year-1) displayed in table 1.1.

After that, we started to apply the RandomForestClassifer and the Grid Search to select who were the best estimators to use in different hypermeters, we named this model as Model_Best. To deal with the overfitting of the Model_Best we decided to change and add some hyperparameters. With this, we ended up creating another model named ModelRFC and kept the hyperparameters selected by Grid Search in Model_Best. Their respective accuracy, f1 score, sensitivity and specificity are displayed in table 1.2. We also performed a confusion matrix that can be seen in Figure 7 and Figure 8.

Table 1.1 – Features selection

Predictor	**Spearman**	**Lasso**	**RFE**	**What do we decide to do?**
ArrivalMonth	Discard	Keep	Keep	Keep
ArrivalWeekNumber	Discard	Keep	Keep	Discard
ArrivalDayOfMonth	Keep	Discard	Keep	Keep
ArrivalHour	Keep	Discard	Keep	Discard
WeekendStays	Keep	Discard	Keep	Discard
WeekdayStays	Keep	Keep	Keep	Keep
Adults	Keep	Keep	Keep	Keep
Children	Keep	Keep	Keep	Keep
Babies	Keep	Discard	Discard	Discard
PreviousReservations	Discard	Discard	Keep	Discard
PreviousStays	Discard	Keep	Keep	Keep
PreviousCancellations	Keep	Discard	Keep	Discard
DaysUntilConfirmation	Keep	Discard	Keep	Discard
BookingChanges	Keep	Keep	Keep	Keep
BookingToArrivalDays	Keep	Keep	Keep	Keep
ParkingSpacesBooked	Keep	Keep	Keep	Keep
SpecialRequests	Keep	Keep	Keep	Keep
OrderedMealsPerDay	Keep	Keep	Keep	Keep
FloorReserved	Discard	Keep	Keep	Discard

FloorAssigned	Discard	Keep	Keep	Keep
BookingToArrivalDays	Keep	Keep	Keep	Keep
DailyRateEuros	Discard	Keep	Keep	Keep
DailyRateUSD	Discard	Discard	Keep	Discard
%PaidInAdvance	Keep	Keep	Keep	Keep
CountryofOriginAvgIncomeEuros (Year-2)	Discard	Keep	Keep	Discard
CountryofOriginAvgIncomeEuros (Year-1)	Discard	Keep	Keep	Keep
CountryofOriginHDI (Year-1)	Discard	Keep	Keep	Discard

Table 1.2 – Model comparison

Metrics	**Model_best_overfit**	**ModelRFC**
Accuracy Score in Train	0.98422	0.81758
Accuracy Score in Validation	0.785501	0.77977
Score on Kaggle	0.78373	0.77368
Sensitivity	0.86337	0.91247
Specificity	0.65583	0.65941

DISCUSSION

Analyzing some of the variables we dropped, starting with the univariate variables, i.e., variables that have variance equal to zero, since they may be harmful to our model and probably do not provide much useful information.

Other discarded features, mentioned in the previous section, were a result of the suggestions by the techniques we used for feature selection (Pearson and Spearman Correlation, RFE, Lasso, and Chi-squared Test). Looking at the correlation matrices, we found variables highly correlated with each other. For example, CountryofOriginAvgIncomeEuros (Year-2), CountryofOriginAvgIncomeEuros (Year-1), and CountryofOriginHDI (Year-1) are highly correlated features among themselves, so we chose to keep only one, the one that had a higher value for Lasso Regression, which is CountryofOriginAvgIncomeEuros (Year-1). This happens a few times, and we tend to keep only one of them, or, in some cases, neither, if Lasso and RFE also suggest its removal. When taking only Spearman Correlation Matrix, some other variables are also highly associated: ArrivalWeekNumber (removed) with ArrivalMonth, PreviousStays with PreviousReservations (removed), FloorAssigned with FloorReserved (removed), DailyRateUSD (removed) with DailyRateEuros.

From the feature selected methods, mainly Lasso and RFE, we can see that the Babies variable is not relevant, meaning that it has almost no impact on the target, according to them. That was our reasoning for dropping the other features we did as well. Some variables are not as clear and easy to decide to remove as this one, so we had to consider what the methods said together such as WeekendStays, which we removed, and PreviousStays, which we keep.

Many models were tested, and we checked their results on test data on Kaggle for them, but the models that gave us the best results had overfitting. After research and careful consideration, we built a final predictive model that, although it performed a bit worse, did not have the issue of overfitting. Said model is a Random Forest Classifier, with the following parameters: criterion='entropy', n_estimators = 500, bootstrap = False, max_depth= 20, max_samples= None, random_state=5, min_samples_leaf=15, max_features='log2', warm_start=True. We picked them after careful research on what would reduce overfitting while still giving high performance.

- criterion: determines how the impurity of a split will be measured. In the model, the criteria are defined by the 'entropy' which is characterized by a unit of measurement for information that depicts the disorder of the target's characteristics. The feature with the lowest entropy selects the optimal split, just like the Gini Index does.
- min_samples_leaf: when increased it can reduce overfitting. This parameter is the minimum number of samples needed to split a node.
- max_features: determines how many characteristics should be considered to produce the optimum split. In this case, we change the max_features=log2(features).
- warm_start: when set to True, the ensemble of estimators is expanded, and the previous call's solution is reused.

From Table 1.2 we can compare the model that gave us the best score (Model_best_overfit), which has overfitting, and our final model (ModelRFC). It is possible to see that, although the score obtained on Kaggle for the first model is better, for sensitivity and specificity we obtain better values on our model. This indicates how much the score on Model_best_overfit comes from memorization and not actually learning patterns to help in prediction.

Our model performs well on train and validation data, meaning that it was trained well, and validation is a good measure of how well it performs on test data. All three have similar values for the F1 score.

CONCLUSION

To summarize, we concluded that a Random Forest Classifier was the best solution for the problem presented to us. We obtained the following results: 4195 clients (approximately 70%) will not cancel their booking and 1779 (approximately 30%) will cancel.

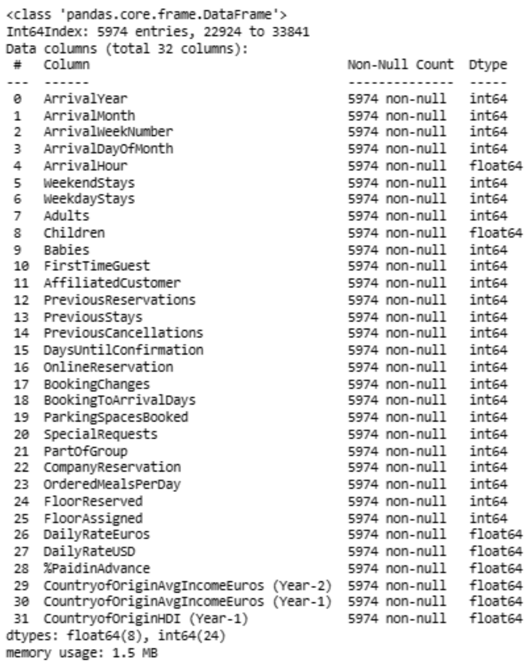
Our goal was to develop a prediction model for a classification problem, more specifically, a model able to identify whether a client will change his booking or not. This is essential for the company to stay afloat in complicated times for businesses of their type, being that times have changed.

We succeeded in achieving a model without overfitting, with similar results in train and validation data and a good F1 score on the test dataset. This makes us confident that we managed to implement a Machine Learning Pipeline and provide useful results for Hotel California.

REFERENCES

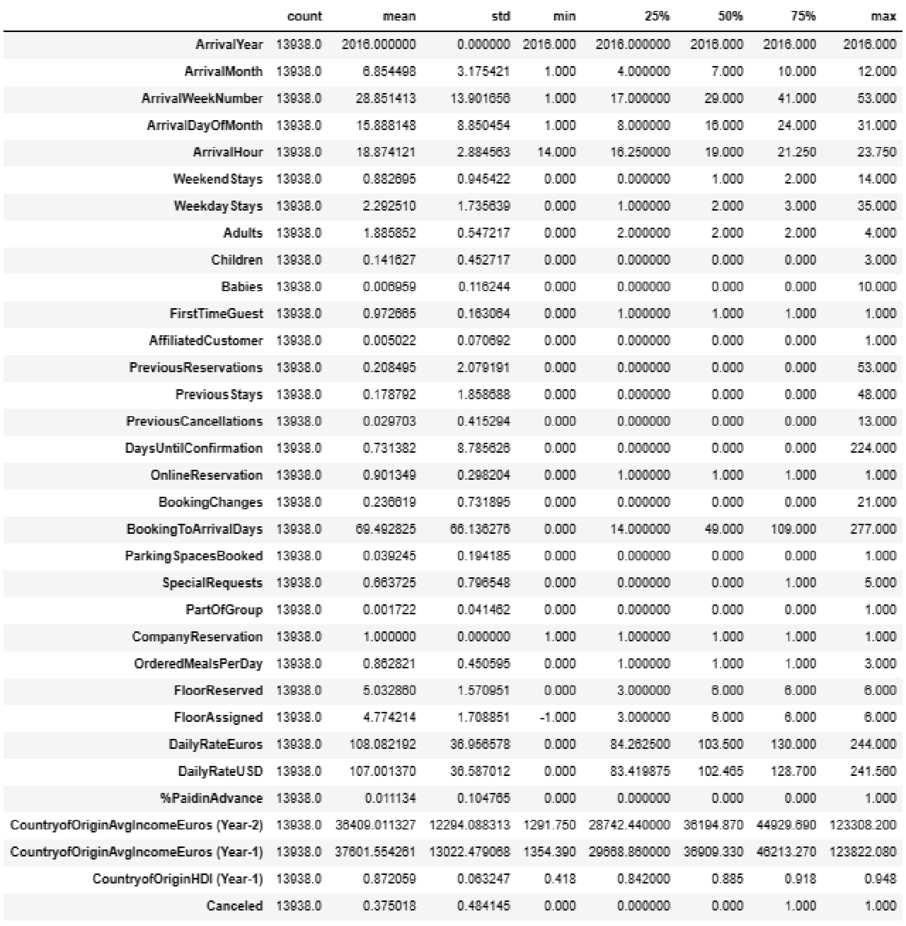
1. Singh, D. & Singh, B. Investigating the impact of data normalization on classification performance. *Appl Soft Comput* **97**, 105524 (2020).
2. Siji George, C. G. & Sumathi, B. Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. *International Journal of Advanced Computer Science and Applications* **11**, (2020).

ANNEXES



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5974 entries, 22924 to 33841
Data columns (total 32 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   ArrivalYear                           5974 non-null   int64  
 1   ArrivalMonth                          5974 non-null   int64  
 2   ArrivalWeekNumber                     5974 non-null   int64  
 3   ArrivalDayOfMonth                     5974 non-null   int64  
 4   ArrivalHour                           5974 non-null   float64
 5   WeekendStays                          5974 non-null   int64  
 6   WeekdayStays                          5974 non-null   int64  
 7   Adults                                5974 non-null   int64  
 8   Children                              5974 non-null   float64
 9   Babies                                5974 non-null   int64  
 10  FirstTimeGuest                        5974 non-null   int64  
 11  AffiliatedCustomer                    5974 non-null   int64  
 12  PreviousReservations                  5974 non-null   int64  
 13  PreviousStays                         5974 non-null   int64  
 14  PreviousCancellations                 5974 non-null   int64  
 15  DaysUntilConfirmation                 5974 non-null   int64  
 16  OnlineReservation                     5974 non-null   int64  
 17  BookingChanges                        5974 non-null   int64  
 18  BookingToArrivalDays                  5974 non-null   int64  
 19  ParkingSpacesBooked                   5974 non-null   int64  
 20  SpecialRequests                       5974 non-null   int64  
 21  PartOfGroup                           5974 non-null   int64  
 22  CompanyReservation                    5974 non-null   int64  
 23  OrderedMealsPerDay                    5974 non-null   int64  
 24  FloorReserved                         5974 non-null   int64  
 25  FloorAssigned                         5974 non-null   int64  
 26  DailyRateEuros                        5974 non-null   float64
 27  DailyRateUSD                          5974 non-null   float64
 28  %PaidinAdvance                        5974 non-null   float64
 29  CountryofOriginAvgIncomeEuros (Year-2)  5974 non-null   float64
 30  CountryofOriginAvgIncomeEuros (Year-1)  5974 non-null   float64
 31  CountryofOriginHDI (Year-1)           5974 non-null   float64
dtypes: float64(8), int64(24)
memory usage: 1.5 MB
```

Figure 1.



	count	mean	std	min	25%	50%	75%	max
ArrivalYear	13938.0	2016.000000	0.000000	2016.000	2016.000000	2016.000	2016.000	2016.000
ArrivalMonth	13938.0	6.854498	3.175421	1.000	4.000000	7.000	10.000	12.000
ArrivalWeekNumber	13938.0	28.851413	13.901656	1.000	17.000000	29.000	41.000	53.000
ArrivalDayOfMonth	13938.0	15.888148	8.850454	1.000	8.000000	16.000	24.000	31.000
ArrivalHour	13938.0	18.874121	2.884563	14.000	16.250000	19.000	21.250	23.750
WeekendStays	13938.0	0.882695	0.945422	0.000	0.000000	1.000	2.000	14.000
WeekdayStays	13938.0	2.292510	1.735639	0.000	1.000000	2.000	3.000	35.000
Adults	13938.0	1.885852	0.547217	0.000	2.000000	2.000	2.000	4.000
Children	13938.0	0.141627	0.452717	0.000	0.000000	0.000	0.000	3.000
Babies	13938.0	0.006959	0.116244	0.000	0.000000	0.000	0.000	10.000
FirstTimeGuest	13938.0	0.972665	0.163064	0.000	1.000000	1.000	1.000	1.000
AffiliatedCustomer	13938.0	0.005022	0.070692	0.000	0.000000	0.000	0.000	1.000
PreviousReservations	13938.0	0.208495	2.079191	0.000	0.000000	0.000	0.000	53.000
PreviousStays	13938.0	0.178792	1.858688	0.000	0.000000	0.000	0.000	48.000
PreviousCancellations	13938.0	0.029703	0.415294	0.000	0.000000	0.000	0.000	13.000
DaysUntilConfirmation	13938.0	0.731382	8.785626	0.000	0.000000	0.000	0.000	224.000
OnlineReservation	13938.0	0.901349	0.298204	0.000	1.000000	1.000	1.000	1.000
BookingChanges	13938.0	0.236619	0.731895	0.000	0.000000	0.000	0.000	21.000
BookingToArrivalDays	13938.0	69.492825	66.136276	0.000	14.000000	49.000	109.000	277.000
ParkingSpacesBooked	13938.0	0.039245	0.194185	0.000	0.000000	0.000	0.000	1.000
SpecialRequests	13938.0	0.663725	0.796548	0.000	0.000000	0.000	1.000	5.000
PartOfGroup	13938.0	0.001722	0.041462	0.000	0.000000	0.000	0.000	1.000
CompanyReservation	13938.0	1.000000	0.000000	1.000	1.000000	1.000	1.000	1.000
OrderedMealsPerDay	13938.0	0.862821	0.450595	0.000	1.000000	1.000	1.000	3.000
FloorReserved	13938.0	5.032860	1.570951	0.000	3.000000	6.000	6.000	6.000
FloorAssigned	13938.0	4.774214	1.708851	-1.000	3.000000	6.000	6.000	6.000
DailyRateEuros	13938.0	108.082192	36.956578	0.000	84.262500	103.500	130.000	244.000
DailyRateUSD	13938.0	107.001370	36.587012	0.000	83.419875	102.465	128.700	241.560
%PaidinAdvance	13938.0	0.011134	0.104765	0.000	0.000000	0.000	0.000	1.000
CountryofOriginAvgIncomeEuros (Year-2)	13938.0	36409.011327	12294.088313	1291.750	28742.440000	36194.870	44929.690	123308.200
CountryofOriginAvgIncomeEuros (Year-1)	13938.0	37601.554261	13022.479068	1354.390	29668.860000	36909.330	46213.270	123822.080
CountryofOriginHDI (Year-1)	13938.0	0.872059	0.063247	0.418	0.842000	0.885	0.918	0.948
Canceled	13938.0	0.375018	0.484145	0.000	0.000000	0.000	1.000	1.000

Figure 2.

```
ArrivalYear                               0.000000e+00
ArrivalMonth                              1.008330e+01
ArrivalWeekNumber                         1.932560e+02
ArrivalDayOfMonth                         7.833053e+01
ArrivalHour                               8.320704e+00
WeekendStays                              8.938226e-01
WeekdayStays                              3.012444e+00
Adults                                    2.994468e-01
Children                                  2.049529e-01
Babies                                    1.351259e-02
FirstTimeGuest                            2.659003e-02
AffiliatedCustomer                        4.997377e-03
PreviousReservations                      4.323033e+00
PreviousStays                             3.454721e+00
PreviousCancellations                     1.724692e-01
DaysUntilConfirmation                     7.718722e+01
OnlineReservation                         8.892550e-02
BookingChanges                            5.356698e-01
BookingToArrivalDays                      4.374007e+03
ParkingSpacesBooked                       3.770775e-02
SpecialRequests                           6.344894e-01
PartOfGroup                               1.719070e-03
CompanyReservation                        0.000000e+00
OrderedMealsPerDay                        2.030361e-01
FloorReserved                             2.467888e+00
FloorAssigned                             2.920173e+00
DailyRateEuros                            1.365789e+03
DailyRateUSD                              1.338609e+03
%PaidinAdvance                            1.097571e-02
CountryofOriginAvgIncomeEuros (Year-2)    1.511446e+08
CountryofOriginAvgIncomeEuros (Year-1)    1.695850e+08
CountryofOriginHDI (Year-1)               4.000203e-03
Canceled                                  2.343963e-01
dtype: float64
```

Figure 3.

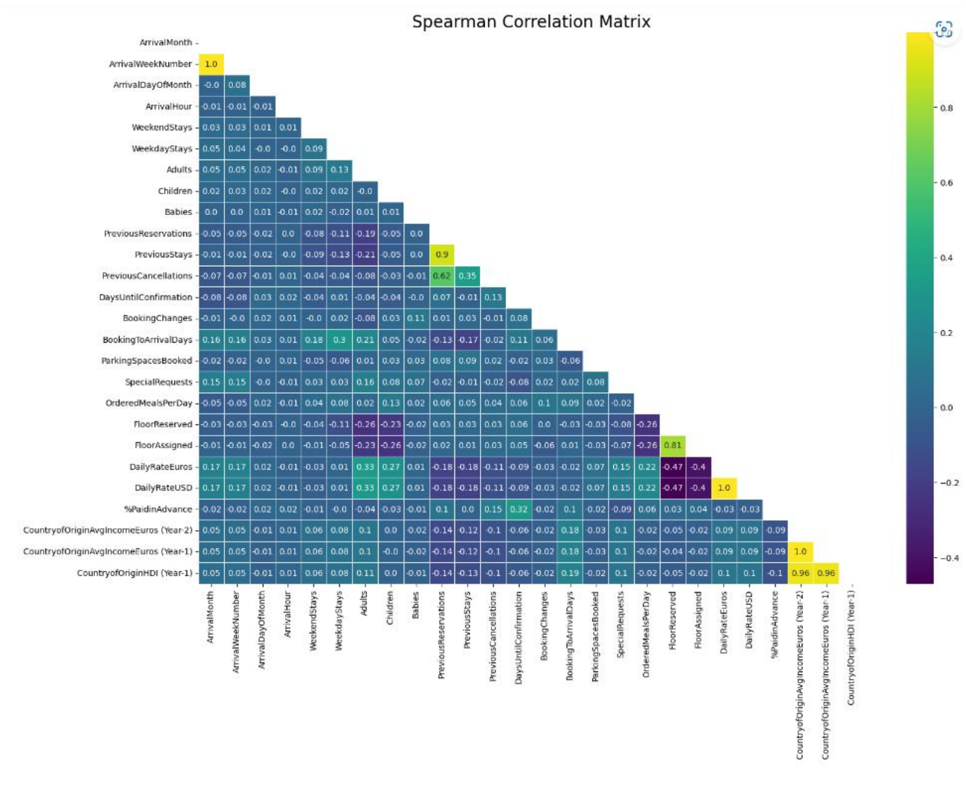


Figure 4.

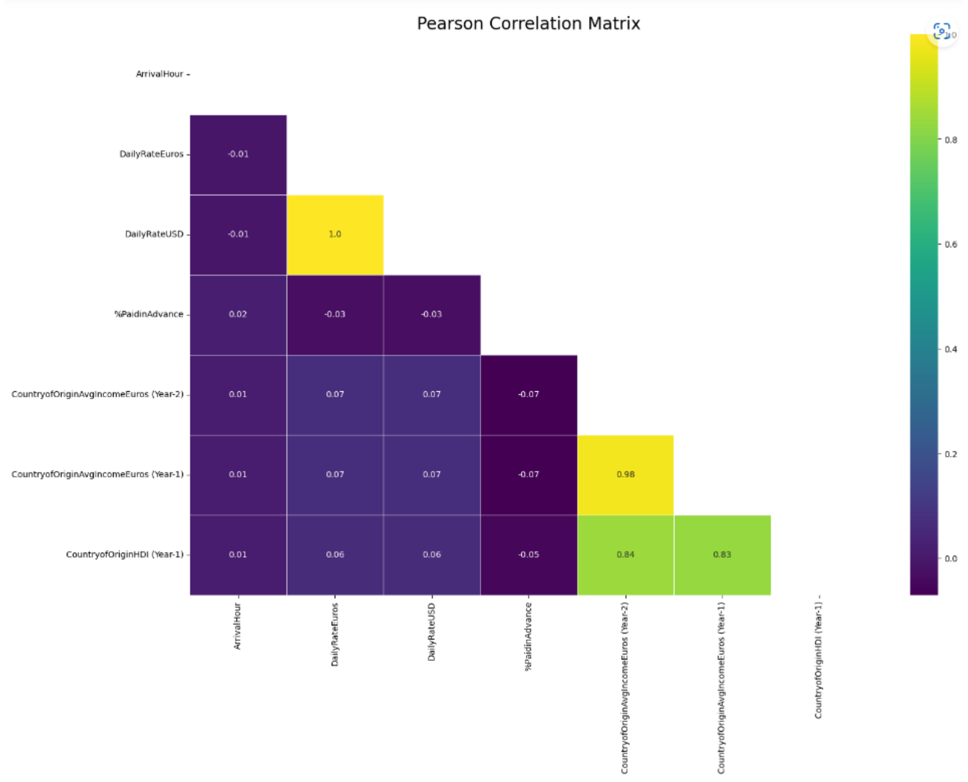


Figure 5.

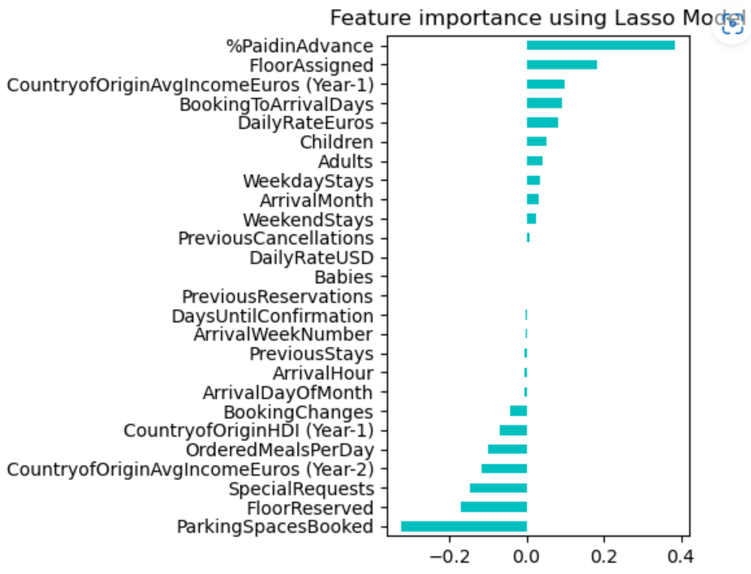


Figure 6.

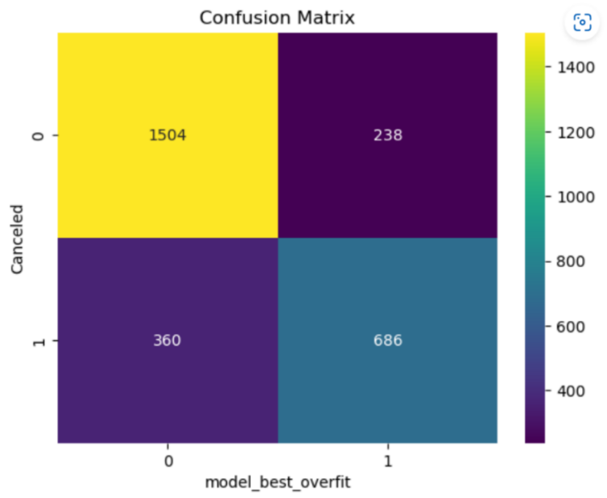


Figure 7.

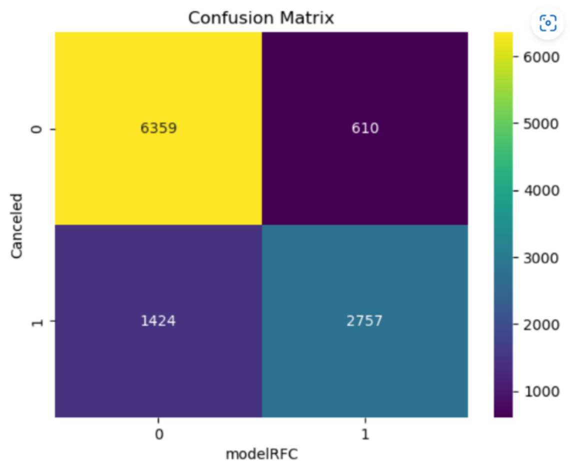


Figure 8.