ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY
ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ
DEPARTMENT OF STATISTICS

STATISTICAL MACHINE LEARNING

First Project

Professor: D.Karlis

17/04/2024
MSc in Statistics - AUEB
Grammenos Konstantinos
AM : f3612302
Title: Project 1 - Classification

Contents

List of Figures

List of Tables

Abstract

In this assignment, we address the challenge of predicting cancellations of hotel bookings using a dataset that includes plenty of variables such as booking ID, number of guests, stay duration, meal type, room type, other more. The very first thing we did was to apply some exploratory analysis in order to understand our data better and make some early findings. Then, we applied and compare six different models / algorithms which are a Logistic Regression model, Decision Tree, Random Forests, the Naïve Bayes classifier, the Linear Discriminant Analysis method and finally Support Vector Machines. We fitted all these models multiple times and we evaluated them based on some performance measures like the accuracy and the area under the curve, to end up with the best model, which will be able to classify better whether the booking has finally cancelled or not.

Introduction

In this project, we analyze data from a random sample of room bookings of a hotel and our goal was to examine whether the booking has finally cancelled or not. To do that, we first did some exploratory and description analysis and then we created models-algorithms to classify whether a booking will be cancelled or not. Also, we assessed how good are the predictions made by your models. The data set contains 2000 rows and 17 columns. More specifically, the variables of our dataset are displayed in the table below.

Table 1: Variables of our Dataset

Variable	Name	Type	Meaning	Value
1	Booking_ID	Character	Unique ID for booking	e.g. BID19169
2	number.of.adults	Numeric	# of adults in booking	From 0 to 3
3	number.of.children	Numeric	# of children in booking	From 0 to 3
4	number.of.weekend.nights	Numeric	# of weekend nights	From 0 to 6
5	number.of.week.nights	Numeric	# of week nights	From 0 to 14
6	type.of.meal	Character	Meal Type	e.g. Meal Plan1,…
7	car.parking.space	Character	Car parking space incl.	0=No or 1=Yes
8	room.type	Character	Room Type	e.g.Room_Type1,..
9	lead.time	Numeric	Days between booking and arrival	From 0 to 418
10	market.segment.type	Character	Market segment type	e.g. Offline, Online
11	repeated	Numeric	Repeat booking	0 or 1
12	P.C	Numeric	# of previous canceled bookings	From 0 to 3
13	P.not.C	Numeric	# of previous non-canceled bookings	From 0 to 41
14	average.price	Character	Average price	From 0 to 349.63
15	special.requests	Numeric	# of special requests	From 0 to 4
16	date.of.reservation	Character	Reservation Date	e.g. 8/17/2017
17	booking.status	Character	canceled or not	0 or 1

First, we did a descriptive analysis for the variables individually and then for the pair wise associations in order to examined the relations between them.

Descriptive analysis and exploratory data analysis

With the use of R Studio, we did some statistical analysis and plenty of graphs for our data. First of all we import our data in R and we put them into a data frame that is called data. First of all, we checked our dataset for missing values and we found out that there were no missing value. Furthermore, we check the class of each variable to see if is in the correct format and then we transform those that we should to the correct class. Also, because of some issues derived from the fact that specific levels in some variables that we transformed to factors like market.segment.type , room.type and type.of.meal had very few observations, we combined those levels as Other. In this way we avoided also the issues that may derive during the splitting in train and test set with some levels of the factors being only in the test and not the train or the opposite. Next we constructed a new variable for our response with the name booking.status.encoded, which will be 1 if the booking was cancelled and 0 otherwise.

Figure 1: Bar plots of categorical variables and for the special requests

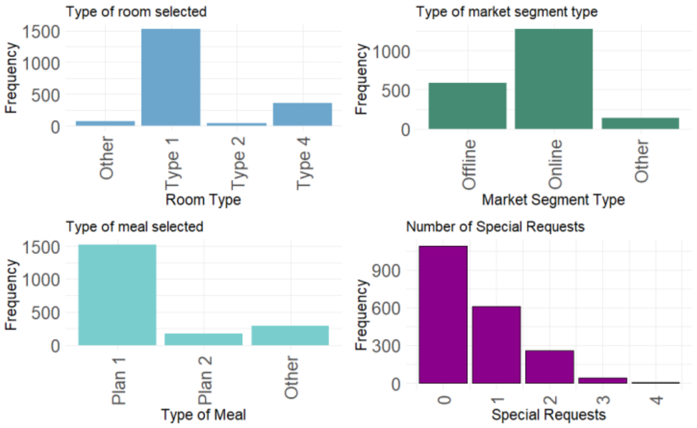


Figure 2: Bar plots of the discrete numeric variables

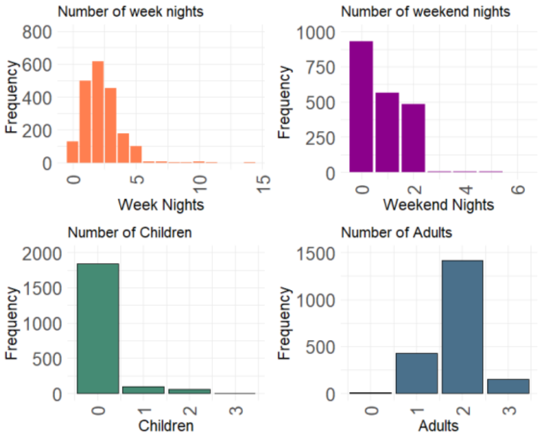


Figure 3: Bar plots of Booking Status, Repeated and Car Parking Space

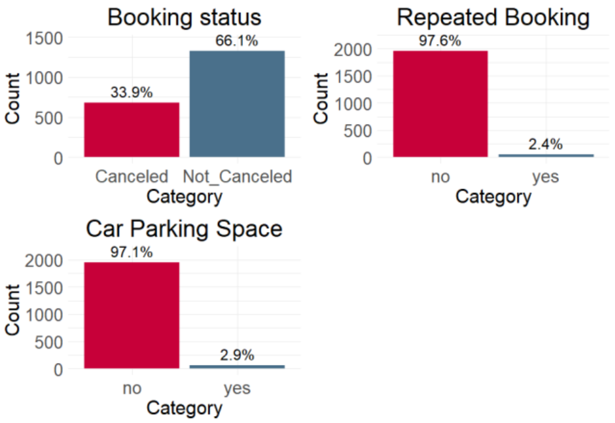
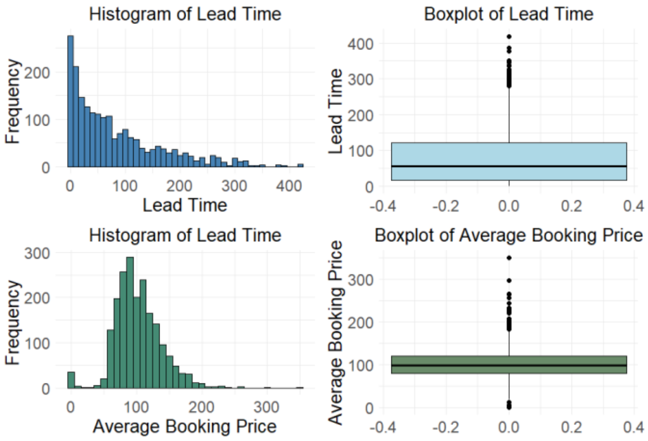


Figure 4: Histograms and Boxplots of the continuous variables Lead Time and Average Price



In Figure 1 we see the distribution of some of our categorical variables. More specifically, we see that out of all the room types the one that is most preferred is the Room_Type1 while for the market segments the online segment stands out as being the most common source of bookings. Also meal type 1 is more popular than others and as we also see in this figure, most bookings have 0 special requests, and the frequency decreasing as the number of requests increases. We will see later if all these variables may play a role and be an indicator of the likelihood of cancellation of a booking.

Some finding from Figure 2 are, that the most bookings are for 0-5 weeknights and also typically include 0-2 weekend nights. The majority of them involve no children and are predominantly made for two adults, suggesting couples or pairs are more common guests in the hotel.

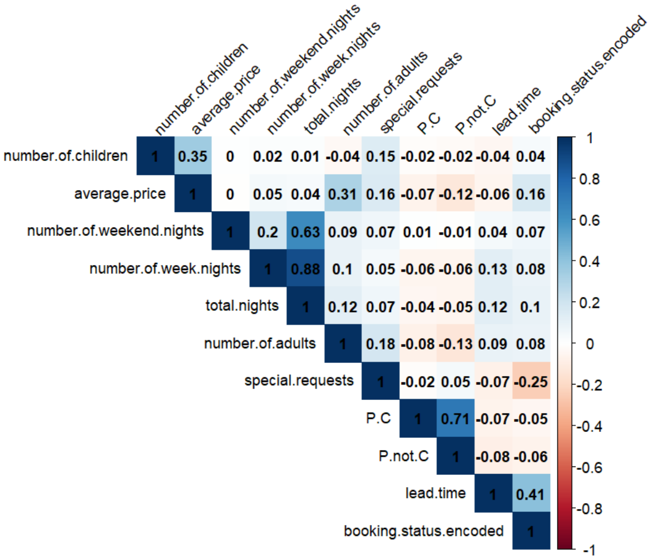
In Figure 3, we see that almost the 34% (33.9) of the hotel booking was cancelled, which is a considerable proportion of them which might demand some action. Also, we found out that the repeated guests are only 2.4% of the total number of guests and that only 2.9% request a car parking space which might indicates that is a less common necessity among guests.

Finally in Figure 4, in the histogram of lead time, we observe a right-skewed distribution for the lead time, indicating that most bookings are made relatively close to the arrival date with fewer of them made well in advance. In the boxplot, we see a median of almost 50 but we have several outliers indicating occasional much longer lead times. As regards the distribution of the average booking price this is also right-skewed, suggesting that more affordable bookings are common, with fewer bookings with higher price. The boxplot also shows us a median around 100 but also some outliers with much higher prices.

Pairwise comparisons

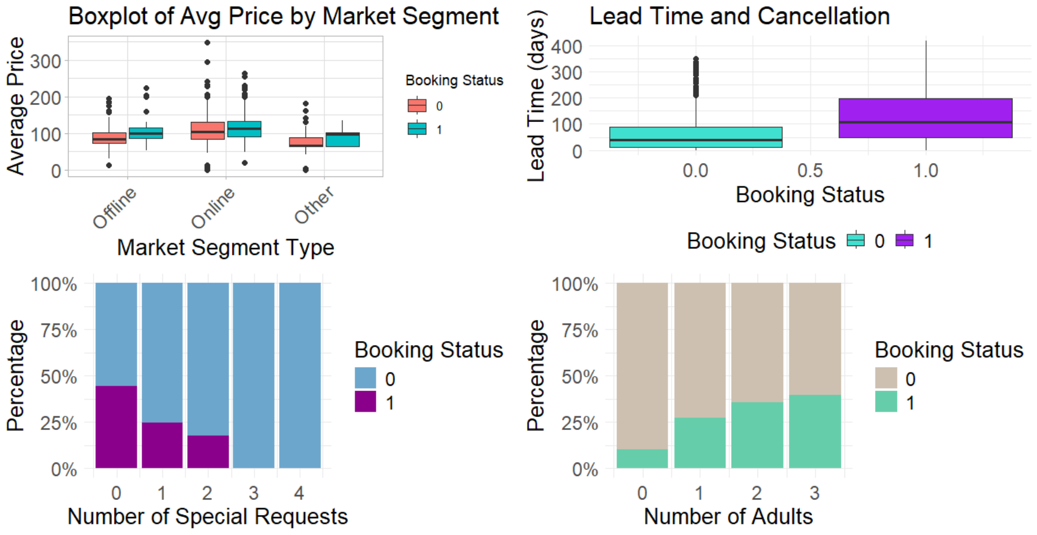
We continued our analysis by examining the pair association between our variables. Firstly and most significantly, we created a Pearson correlation matrix, in order to investigate the relationship between our most important continuous variables. Each cell in the figure 5 shows the correlation between two variables, and the value in the cell represents the correlation coefficient. We know that value 1 indicating a perfect positive correlation, the value -1 indicating a perfect negative correlation, and value 0 indicates no correlation.

Figure 5: Pearson Correlation Matrix



	number.of.children	average.price	number.of.weekend.nights	number.of.week.nights	total.nights	number.of.adults	special.requests	P.C	P.not.C	lead.time	booking.status.encoded
number.of.children	1	0.35	0	0.02	0.01	-0.04	0.15	-0.02	-0.02	-0.04	0.04
average.price		1	0	0.05	0.04	0.31	0.16	-0.07	-0.12	-0.06	0.16
number.of.weekend.nights			1	0.2	0.63	0.09	0.07	0.01	-0.01	0.04	0.07
number.of.week.nights				1	0.88	0.1	0.05	-0.06	-0.06	0.13	0.08
total.nights					1	0.12	0.07	-0.04	-0.05	0.12	0.1
number.of.adults						1	0.18	-0.08	-0.13	0.09	0.08
special.requests							1	-0.02	0.05	-0.07	-0.25
P.C								1	0.71	-0.07	-0.05
P.not.C									1	-0.08	-0.06
lead.time										1	0.41
booking.status.encoded											1

Figure 6: More Pairwise Comparisons - Boxplots and Stacked Bar Plots



In the Pearson correlation matrix in Figure 5, we see a moderate positive correlation (0.41) of the lead time with booking cancellations (booking.status.encoded), which may indicates that bookings made further in advance have a higher likelihood of being canceled. In addition we observe a negative moderate correlation (-0.25) between special requests and booking.status.encoded suggesting that while the number of requests increases the likelihood of the cancelation of a booking decreases. We expected to see a stronger correlation between the average price and the cancelation of a booking but at the end we found only a slight positive correlation (0.16) between them. Finally, the number of adults, the number of children and the number days of stay have a very weak but positive correlation with the cancelation of a booking.

In Figure 9 we also see the fact that longer lead time are associated with cancelled bookings, as the median is larger (about 110) and the range for cancelations (booking status = 1) is wider than the range of the non-cancelations. This shows us that those who book their room well in advance are more prone to cancel their booking. We also see a boxplot of average price by market segment type for both the cancelations and the non-cancelations. As regards the online and the offline the medians and the range between cancelations and non-cancelations are pretty similar in both categories but in the market segment other (Corporate, Aviation and Complementary) the range for cancelations is much wider and also the median is quite higher than the median of cancelations. In all categories we have plenty of outliers.

We also observe from the stacked bar plot of the special requests with the booking status that when special requests were 3 or 4 none of the bookings was canceled, while the highest percentage of cancelations was when we do not have any special request where the percentage was a bit lower than 50%. One more time we see here that bookings without or with just 1 or 2 special requests have a higher cancellation rate than those with 3 or 4 special requests, so as the number of special requests increases the likelihood of the cancelation decreases. We see that negative correlation that we observed also in the Pearson correlation matrix.

Finally, in the last plot the stacked bar plot of the number of adults with the booking status we observe that the cancellation rate appears consistent (around 25 %) when the adults are 1,2 or 3, suggesting that the number of adults may not be a significant predictor of cancellation as we also showed in the correlation matrix.

Classification models/Algorithms

After removing the columns of date.of.reservation, Booking_ID and booking.status (we do not need it as we have the new column booking.status.encoded) we randomly split the data to 80% training data and 20% test data.

Variable Selection for Logistic Regression

First of all, in order to fit a Logistic Regression model, we did Lasso and we found out a minimum lambda and also that the β coefficients of the variables P.C, P.not.C and the number.of.children were shrunk to zero. But it would not be correct to do Lasso only once to one train set and that is why we did lasso 100 times to different train data in order to see which variables s coefficients are shrunk to 0 most of the times. So in the below figure we see how many times the coefficient of each variable was shrunk to 0 after fitting the lasso model :

Table 2:Table of the count of the # of times the coefficients of each variable was shrunk to 0

Variable Name	# of times coefficient was 0
number.of.adults	24
number.of.children	40
number.of.weekend.nights	1
number.of.week.nights	16
type.of.mealMeal Plan1	63

type.of.mealMeal Plan2	24
type.of.mealOther	55
car.parking.space	1
room.typeRoom_Type 1	1
room.typeRoom_Type 2	31
room.typeRoom_Type 4	9
lead.Time	1
market.segment.typeOnline	1
market.segment.typeOther	1
repeated	0
P.C	99
P.not.C	99
average.price	1
special.requests	1

So, as we see in the figure above, variables like P.C and P.not.C have been shrunk to zero in nearly all iterations (99 out of 100), suggesting these variables are not contributing at all to the ability of a model to predict whether a booking will be canceled or not. Also, someone can say that the number of children and number of adults may do not contribute to the model while on the other the fact that some variables s coefficients are never 0, means that they are important and consistent predictors in determining the class of the booking.status.encoded (canceled , not canceled).

So, after these findings we tried fitting the Logistic Regression model without some of these variables and for sure without P.C and P.not.C but the results were almost the same in every approach, so we decided to use the full model.

Logistic Regression

The very first classification method that we did was logistic regression. More specifically we fit a GLM model with binomial family in the training data and we included all the variables. Basically, with this model we will estimate the βs that we need in order to calculate the probabilities. The logistic regression model predicts the probability that an individual (here a booking) belongs to a particular group-class (cancelled or not cancelled). This probability is calculated using this type:

$$P(Ci = 1) = \frac{\exp{(\beta * xi)}}{1+\exp{(\beta * xi)}}$$

where xi is a vector with covariates related to the i-th booking and β a vector of regression coefficients which for us are the coefficients of all the predictors in the trainset. For us class 1, that means the booking.status.encoded = 1 refers to a booking that was cancelled.
First, we run it once in order to check that it works in the correct way. For the train data we found that our model had almost 0.78 accuracy while the area under the curve was approximately 0.86. For the test data the accuracy was 0.80 and the area under the curve was 0.87. But, we have to run our logistic model multiple times and in different train and test sets in order to produce some performance measures and to be able to assess how good are the predictions made by our logistic regression model. So for this reason we will run a loop 100 times in which in every iteration we will randomly split our data to 80% train and 20% test and we will fit the logistic regression model in the training data and have predictions for both train and test data. After we run this , we observed some interesting performance measures. Below we can see the summary of the accuracy for the predictions

for the train data and the summary of accuracy for the predictions for the test data. The accuracy is estimated by the sum of the true positives and true negatives (TP + TN) over the sum of all the predictions (TP + FN + TN + FP). We observe that the mean accuracy for the train data is about 0.792 while the mean of the accuracy of the prediction on the test set is about 0.78. The mean area under the curve for the test set is about 0.85.

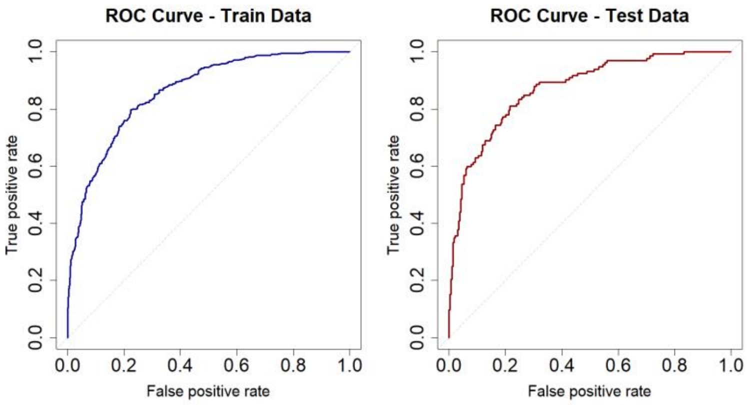
Table 3: Summary of Accuracy of predictions for the Train Data

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.7825	0.7892	0.7919	0.7923	0.7956	0.8063

Table 4: Summary of Accuracy of predictions for the Test Data

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.7325	0.7725	0.7887	0.7859	0.7975	0.8250

Figure 7: Example of ROC Curves for Train and Test Data



Decision Tree

First of all we need to say that there are 2 types of decision trees, the classification trees which predict a class and the response is categorical and the regression trees which aim to predict a value and in this case we have a continuous response. In our booking example we will fit a classification tree and of course we will transform our response (booking.status.encoded) to a factor with 2 levels, in order to fit the model in the correct way. In the below figure we see a decision tree which was fitted using the train data. Each node specifies a test on a variable and each leaf node indicates the value of the target variable. All the variables in decision trees must be discretized, the algorithm does the discitization for us and as we can see from the figure 14 for example lead time is a continuous variable but the algorithm split it to two parts one <149 and the other >149 where 149 is the point that we have the most information gain. The algorithm is using the Gini index (that is the default) in order to find the information that it takes from each variable. In our decision tree we see that the lead time is the most important variable for us as it is the first one in the tree and we grow the tree by selecting every time the variable that has the larger information gain. At the end, we stop when new nodes are not statistically significant that means the information gain is too small.

So, we fitted a tree using the tree library in R, to 100 different train sets and we made predictions for 100 different test sets and we found a mean accuracy of about 0.81. We know that we can never trust the predictions in train data in a decision tree as it has the tendency to overfit the data.

Figure 8: A typical classification decision tree for our dataset

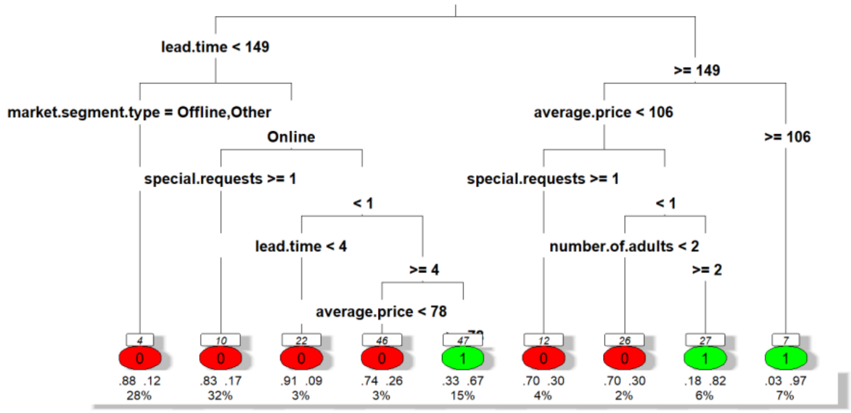


Table 5: Summary of Accuracy of predictions for the Test Data - Decision Tree Algorithm

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.7475	0.7925	0.8075	0.8056	0.8175	0.86

Random Forest

The third classification method that we did was a Random Forest using the library randomForest. First of all, we wanted to see the optimal combination of the number of trees and the number of variables for our algorithm. In order to do this, we fitted the Random Forest model in the train data using different combinations each time and we evaluated these combinations based on the Out Of Bag error rate (the Out of Bag is the Bootstrap Test Set). In order to fit the model right we transformed our response (booking.status.encoded) to a factor with levels 0 and 1.We fitted the model multiple times and for all the different combinations and we found out that the best combination seems to be the one with 500 trees and 3 variables, but we know that there is never a clear winner between combinations. When we say the best one we mean the combination which appear to have the smallest out of bag error, at most of the times.

So, after finding the optimal combination of these hyperparameters of the Random Forest model, we run 30 iterations where in each one, a random subset containing 80% of the dataset was selected as the training set and the remaining 20% as the test set and a Random Forest model with 500 trees and a variable sampling size of 3 per node was constructed using the training data. That means that in every node m variables are selected randomly in order to make a decision and we know that this m is constant throughout the process. The selection of m is extremely important. In particular, the way the algorithm works is to construct 500 trees, take one prediction from each tree and combine them all together to create a generic prediction. For each tree, we take a Bootstrap sample which will be the training set of the tree and the test set will be the Out Of Bag set, that is 36.8% of the data. As we know, we do not want to have similar trees in the forest because the better the individual tree the better the forest.

In each iteration we made predictions for the test data, we calculated the confusion matrix in order to compare the predicted and actual booking statuses and record the accuracy which as we have said is the proportion of the correct predictions made by our model. After running it 30 times we found that the mean accuracy of the predictions for the test set is about 0.84 (0.8393) while the mean out of bag error is about 0.16.

We also visualized the importance of each variable based on the random forest model. The importance of each variable is basically how much our random forest model uses this variable to make correct predictions. The more the model relies on this variable the more important it is.

In the graph below we see the MeanDecreaseAccuracy which is how much the appearance of a specific variable decreases the accuracy and the MeanDeacreseGini which is a measure based on the Gini index. In both measures lead time seems to be the most important variable and also extremely important ones are the special requests, the average price and the market segment type.

In the second graph below, we see the different error rates by the number of trees. We see 3 lines in the graph, the black one refers to the mean Out Of Bag error rate, the red refers to the minimum out of bag error rate and the green line represents the maximum error rate for the out of bag set.

Figure 9: Plot of the Importance of each variable

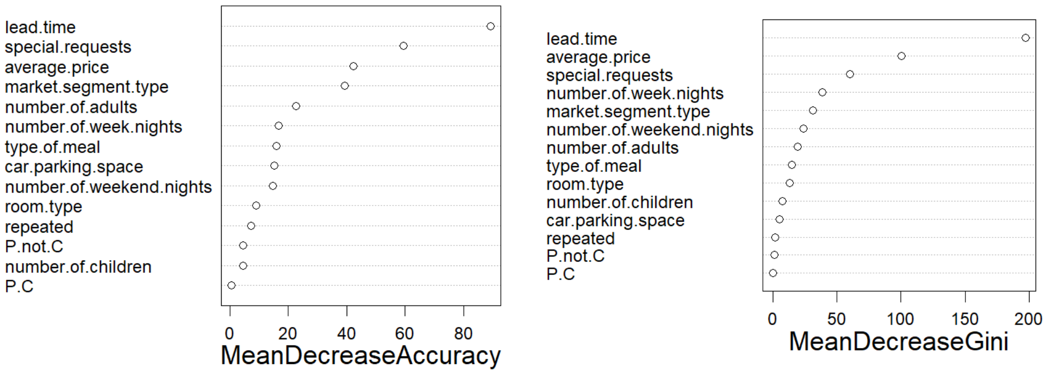


Figure 10: Random Forest Error rate by # of trees

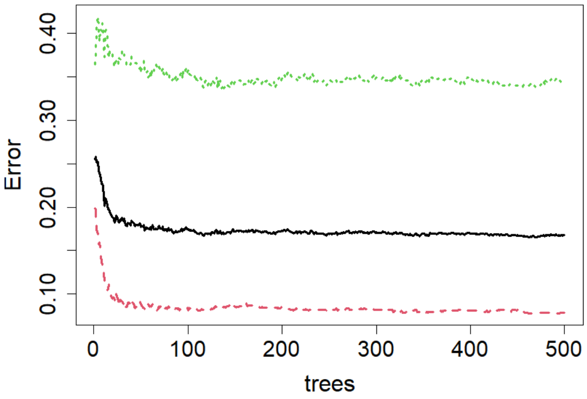


Table 6: Summary of Accuracy of predictions for the Test Data - Random Forest model

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0. 8150	0.8325	0.8438	0.8441	0.8525	0.8800

Naïve Bayes Classifier

The fourth model that we tried to do in order to predict if a booking will be cancelled or not was a Naïve Bayes Classifier. The Naïve Bayes classifier is a very simple model in which there is an unobserved class variable (in our case the booking.status.encoded.) that we want to predict, and several variables X1, X2, . . . , Xp that depend on the class variable.

Those X we can observe them. The good thing with this method is that it is quite easy to implement and it does not require large amount of data. The bad thing is that it relies on the conditional independence assumption which is a very strong one. As we understand, the key assumption of the Naive Bayes approach is that the variables are independent conditional on the class. Of course, the principle behind Naive Bayes is the Bayes theorem also known as the Bayes Rule. Basically, what this does is to measure the conditional probability of an event with a feature vector x1,x2,…,xp belonging to a particular class C (for example in the class=1 that for us means the booking is cancelled) with this type:

Figure 11: Bayes type in Naive Classifier

$$P(C|X_1, X_2, \ldots, X_p) = \frac{P(X_1, X_2, \ldots, X_p|C)P(C)}{P(X_1, X_2, \ldots, X_p)}$$

In order to implement this method we uses the libraries e1071 and caTools.
First we fitted a Naïve Bayes model using all the predictors in the train data and we found out that the accuracy of predictions both for train and for test data were extremely small something like 0.33 for the test data and 0.35 for the train data.

Variable Selection for Naïve Bayes Classifier

So, we did variable selection, by fitting in a loop the Naïve Bayes model without a different variable each time and each time we kept the fitted model without the variable that if we remove it from the model we have the better accuracy and by better I mean the largest. As a result after running this, after removing the variables P.not.C, P.C, number.of.week.nights and number.of.adults, we manage to have an accuracy of about 0.77.
So, now again we run a loop of 100 iterations where in each one we randomly split the data to 80% train and 20% test set and we fit a Naïve Bayes model without the variables that we removed from the variable selection process before and using the train data. Of course in each run, we do predictions using the test set and we calculate the accuracy of our predictions and the confusion matrix. After keeping all the accuracies of all the iterations we found a mean accuracy of 0.76 for the test data which is a bit smaller from the accuracies that we got from other different classification methods. Below we can see the summary of the accuracy for the test data.

Table 7: Summary of Accuracy of predictions for the Test Data - Naive Bayes Classifier Model

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0. 700	0.7425	0.7550	0.7574	0.77	0.82

Linear Discriminant Analysis (LDA)

One other method that we used to predict the booking cancelation is the Linear Discriminant Analysis and by this we mean the Fischer LDA to be precise. In our example we have 2 classes so we will have 1 discriminant function. This method requires no assumptions. The purpose of the linear discriminant analysis is to find combination of the variables that give best possible separation between groups (Canceled Booking and Not Cancelled Bookings) in our data set. Also here we run a loop of 100 iterations where in each one we randomly split the data to 80% train and 20% test set and we fit a LDA method in the train data using all the predictors (full model) and we calculated the accuracy of each model. Below can see the table of the summary of the accuracy over the 100 iterations and the histogram of it. We found out that the mean accuracy is about 0.78. Also, in Figure 12, is a scatter plot of the scores from an LDA model and as we can see the 2 classes (0 for

Not_Cancelled and 1 Canceled) are very well separated. In this plot the y-axis, represents the scores from the linear discriminant function. For example if the linear discriminant function from the model (from 1 fit with specific train and test data) is like this one :

LDA1 = 0.003539955 * number.of.adults + 0.099921978 * number.of.children + 0.068725400 * number.of.weekend.nights + ….. -0.881551241 * special.requests.

every time a new observation comes in we just replace all these variables with the values of the new observation.

Table 8: Summary of Accuracy of predictions for the Test Data - LDA method

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.7225	0.7694	0.7825	0.7818	0.7950	0.8250

Figure 12: Histogram of accuracies of predictions for the test data - LDA method

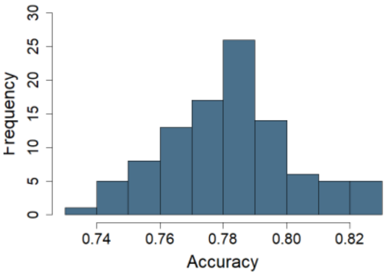
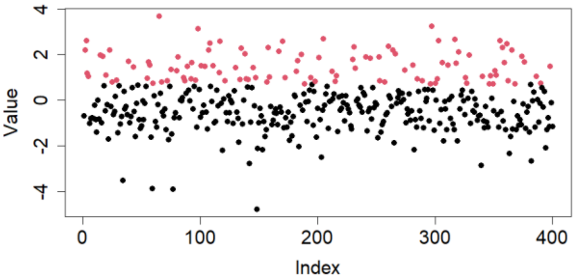


Figure 13: Scatter plot of the scores from the Linear Discriminant Analysis (LDA) model



Support Vector Machines (SVM)

The last classification method that we did is Support Vector Machines (SVM). The SVM method is used only for binary classifications and what id does is to separate the two classes by a hyperplane. SVM both minimize the empirical classification error and maximize the geometric margin at the same time. Here we want to find a line that maximizes the margin between the 2 classes. The support vectors are the data points which are closer to the decision plane or line. Only the support vectors are those who are being taken into account and contribute to the solution while outliers do not contribute at all to it.

In SVM method we have 2 important parameters :

1. The Cost parameter, which adds a penalty for each misclassified point. If the cost is small, the penalty is low we have a large margin and a greater number of misclassifications and the opposite is true if cost is large.

2. The γ parameter, which is used in Radial Basis function in kernels and basically it is how much we have to transform our data. Gamma parameter is used so to determine the boundaries that separate the two classes.

In order, to find the optimal values for those two parameters we will perform grid search with these values:

- For γ parameter: 0.7,1,1.2,1.5,2

- For the cost parameter: 0.5,1,1.5

After running the grid search multiple times we found out that the optimal parameters for gamma and cost are cost = 1 and γ = 0.7.

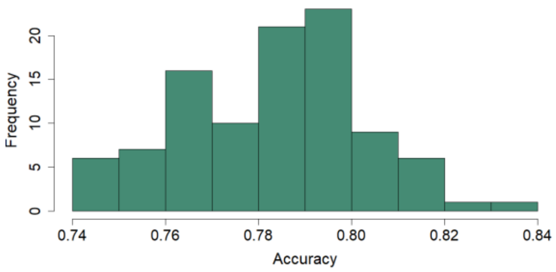
Furthermore, we did the same process and logic for variable selection by fitting in a loop the Support Vector Machines model without a different variable each time and each time we kept the fitted model without the variable that if we remove it from the model we have the better accuracy and by better I mean the largest. As a result after running the only variable that was removed was the number.of.week.nights and we did not manage to have a much better accuracy for our train data as we already had a extremely good accuracy of 0.91. Our new accuracy is just some decimal digits above.

So, again we run a loop of 100 iterations where in each one we randomly split the data to 80% train and 20% test set and we fit a SVM model in the train data using all the predictors (full model) because just removing this one variable did not gave us better accuracy, we calculated the accuracy of each model and we found a mean accuracy. Below we can see the summary of the accuracy of the SVM fitted models and a histogram of the accuracies. The mean accuracy of the SVM method is 0.78.

Table 9: Summary of Accuracy of predictions for the Test Data - SVM method

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.7425	0.7775	0.7875	0.7883	0.7981	0.8325

Figure 14: Histogram of accuracies of predictions for the test data - SVM method



Evaluation of the models

As regards the best possible model among all the ones that we fitted, the Random Forest is the best possible model, as it has the larger accuracy of all and that is 0.84. Random Forest was the best approach in classifying whether a booking will be cancelled or not, cause it has the larger proportion of correct predictions. Below, we see a confusion matrix that was produced by a random forest model:

Table 10: Confusion Matrix of a Random Forest model

Actual	**Predicted**	
	0	1
0	245	18
1	43	94

Conclusions

In conclusion, throughout our analysis we found out plenty of useful and interesting insights. First of all, we found some insights about our variables and we saw also the correlation between them and especially with our response (booking.status.encoded). We saw that lead time, and the average price are positively correlated with the cancelation of a booking while the special requests are negatively correlated with the likelihood of cancelation. Also, after fitted 5 different models, we reached to the conclusion that the Random Forest model is the most accurate one in classifying whether a booking is cancelled or not. All these insights derived from our analysis could assist in the optimization of booking management, and minimize the financial impact of booking cancellations.