

**Final Report: Hotel Booking Demand Dataset**

ALY 6020 -Predictive Analytics

Professor: Ajit Appari

Yi Yang

Fall 2020

Due December 10, 2020

**EXECUTIVE SUMMARY**

Competition in the hotel industry has been quite high and hotel companies have been trying to attract customers to book their rooms and preventing them from cancelling their booking. There are circumstances that customers want to cancel their booking for whatever reasons before their trip starts. Bookings were canceled is such a great loss for hotel business, and it is important for them to find some influential factors that lead to people cancel their booking and generate some policies to decrease the chance of people cancelling.

In order to achieve this, some machine learning methods are used to identify and predict various variables that affect most the chance of booking being canceled.

In this study, after getting the dataset including hotel booking selected, steps of data preparation like sampling, checking for null values and removing duplicate columns have been done. By modeling and analyzing the chance of canceling using 4 machine learning methods including K-Nearest Neighboring(KNN), Logistic Regression(LR), Decision Tree(DT) and Random Forest(RF) to predict multiple variables that affect customers most to cancel their booking, some business suggestions have been made for hotel business to improve their services. The compusionMatrix is used to output the result of predictions and generate the performance of all four models. Among these models, the RF algorithm turns out to have the highest accuracy. Turning algorithm is also used to optimal the RF model. Then, determining the importance of variables in each of the model to determine how important is each variable for the prediction of whether bookings are being canceled display the most factors that hotel business need to pay attention. Lastly, the comparative study shows that the RF method outperforms all other methods in terms of accuracy.

**INTRODUCTION**

**Problem Statement**- What factors that affect hotel cancellation most?

The customer booking is an important factor to influence the hotel income. Our group is very interested at exploring what factors that affect the hotel income, specifically in terms of preventing customers that have already booked from cancelling their booking. So, we want to apply machine learning modeling techniques to identify the top factors to help the hotel business reduce the room vacancy rate in order to earn more money.

For this data set, we want to predict whether customers will cancel their booking or not. Thus, we will preprocess our dataset and create the prediction models using KNN algorithm, Logistic Regression model, Decision Tree model and Random Forest to achieve our target.

**IMPLEMENTATION**

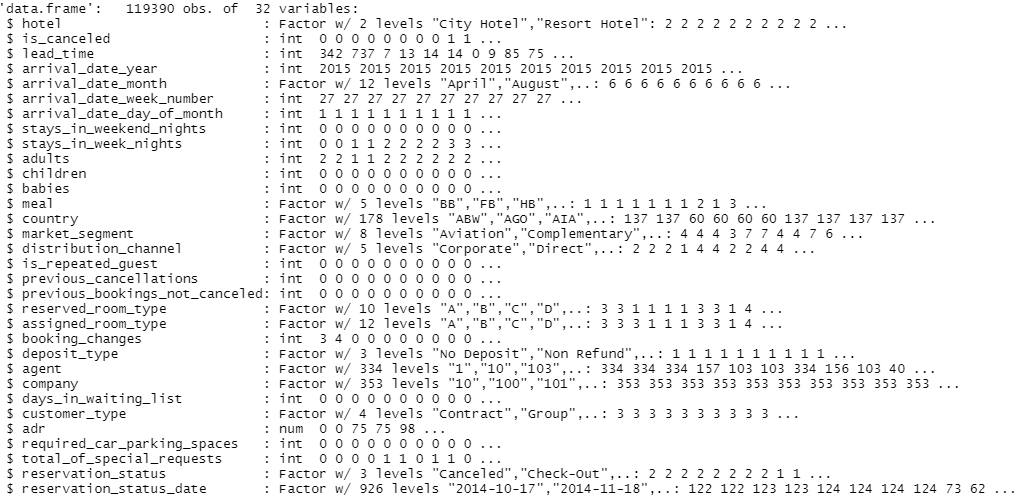
*Data Collection*

Our group decided to explore the data set that is about the customer hotel booking demand from public repository Kaggle (<https://www.kaggle.com/jessemostipak/hotel-booking-demand>). It contains over 119390 observations and 32 independent variables. Below is the table of the description of all variables.

|  |  |  |
| --- | --- | --- |
| Attribute | Attributes Description | Attribute Value Level |
| Hotel | Two types of hotel | City Hotel, Resort Hotel |
| **is\_canceled (Dependent Variable)** | Value indicating if the booking was canceled | 0: Not canceled  1: Canceled |
| Lead time | Number of days that elapsed between entering date of booking into property management system and arrival date | Range between 0 and 737 |
| Arrival Date Year | Year of arrival date | Range between 2015 and 2017 |
| Arrival Date Month | Month of arrival date | Range between Jan and Dec |
| Arrival Date Week Number | Week number of year for arrival date | Range between 1 and 53 |
| Arrival Date Day of Month | Day of arrival date | Range between 1 and 31 |
| Stays in Weekend Nights | Number of weekend nights (Sat/Sun) the guest stayed or booked to stay at the hotel | Range between 0 and 19 |
| Stays in Week Nights | Number of weekend nights (Mon-Fri) the guest stayed or booked to stay at the hotel | Range between 0 and 50 |
| Adults | Number of adults stayed at the hotel | Range between 0 and 55 |
| Children | Number of children stayed at the hotel | Range between 0 and 10 |
| Babies | Number of babies stayed at the hotel | Range between 0 and 10 |
| Meal | Type of meal booked | BB – Bed & Breakfast;  FB – Full board (breakfast, lunch and dinner);  HB – Half board (breakfast and one other meal – usually dinner);  Undefined/SC – no meal package; |
| Country | Country of origin (in ISO format) | PRT, GBR, FRA, ESP, DEU and other |
| Market Segment | Market segment designation | Online TA: Travel agents  Offline TO: Tour operators  Groups; Direct; Corporate; Complementary; other |
| Distribution Channel | Booking distribution channel | Corporate; Direct; GDS; TA/TO; undefined |
| Is Repeated Guest | value indicating if the booking name was from repeated guest | 0: No  1: Yes |
| Previous Cancellations | Number of previous bookings that were cancelled by the customer prior to the current booking | Range between 0 and 26 |
| Previous Booking not Canceled | Number of previous bookings not cancelled by the customer prior to the current booking | Range between 0 and 72 |
| Reserved Room Type | Code of room type reserved. Code is presented instead of designation for anonymity reasons | A; D; E; F; G; B; other |
| 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 to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons. | A; D; E; F; G; C; other |
| Booking Changes | Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation | Range between 0 and 21 |
| Deposit Type | Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories. | *1: No Deposit* – no deposit was made;  *2: Non Refund* – a deposit was made in the value of the total stay cost;  *3: Refundable* – a deposit was made with a value under the total cost of stay. |
| Agent | ID of the travel agency that made the booking | Null; 1; 7; 9; 14; 240; other |
| Company | ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons | Null; 40; 223; 67; 45; 153; other |
| Days in Waiting list | Number of days the booking was in the waiting list before it was confirmed to the customer | Range between 0 and 391 |
| Customer type | Type of booking, assuming one of four categories | *1: Contract* - when the booking has an allotment or other type of contract associated to it;  *2: Group* – when the booking is associated to a group;  *3: Transient* – when the booking is not part of a group or contract, and is not associated to other transient booking;  *4: Transient-party* – when the booking is transient, but is associated to at least other transient booking |
| Adr | Average Daily Rate defined as = | Range between -6.38 and 5400 |
| Required Car Parking Spaces | Number of car parking spaces required by the customer | Range between 0 and 8 |
| Total of Special Requests | Number of special requests made by the customer (e.g. twin bed or high floor) | Range between 0 and 5 |
| Reservation Status | Reservation last status, assuming one of three categories. | *1: Canceled* – booking was canceled by the customer;  *2: Check-Out* – customer has checked in but already departed;  *3: No-Show* – customer did not check-in and did inform the hotel of the reason why |
| Reservation Status Date | Date at which the last status was set. This variable can be used in conjunction with the *Reservation Status* to understand when was the booking canceled or when did the customer checked-out of the hotel | 2015-10-21;  2015-07-06;  2016-11-25;  2015-01-01;  2016-01-18;  2015-07-02;  other |

*Table 1*. Data Dictionary of Hotel Booking

Besides, by using the str() function in RStudio, each variable’s datatype is displayed below.



The key variable is “is\_canceled” that is a binary value contain 0 and 1 which stands for booking got canceled and not canceled, and there are some other variables “adult”, “children” and “customer\_type” etc. After getting to know about this data set, steps of data preparation are accomplished next to be ready for the prediction model building.

*Data Preparation*

Before the modeling part, we need to clean the original data set to get the data set which can be used in our models.

The first step is to obtain the sample data set. We choose to get 20 percent data set from the original data set. Because the “is\_canceled” column has two categories including 0 and 1, and it has some special ration between 0 and 1 data. So, we used the same ratio to get our sample data set.

This is the ratio of the original data set.图片包含 图示

描述已自动生成

This is the ration of our sample data set. 图片包含 图示

描述已自动生成

After the data sampling part, we used the summary() command to check the Null data. In our data set, there are two columns have the NULL data. And there are too many NULL data, so we decided to remove the “agent” column and “company” column.

表格

描述已自动生成

After removing the NULL data column, we checked the missing values (NA) in our dataset. We used the “sum(is.na(data))” command to check the NA data and there is only 1 NA data in our sample data set.

图片包含 徽标

描述已自动生成

So, we dropped the NA data by using the “hotel2<-na.omit(hotel2)” command.

Finishing the NA data and NULL data we can change the columns. For the first column Average Daily Rate (“adr”) column which data can't be 0 or negative. So, we used “hotel2<-hotel2[-which(hotel2$adr<=0),]” command to remove the data which adr less or equals 0.

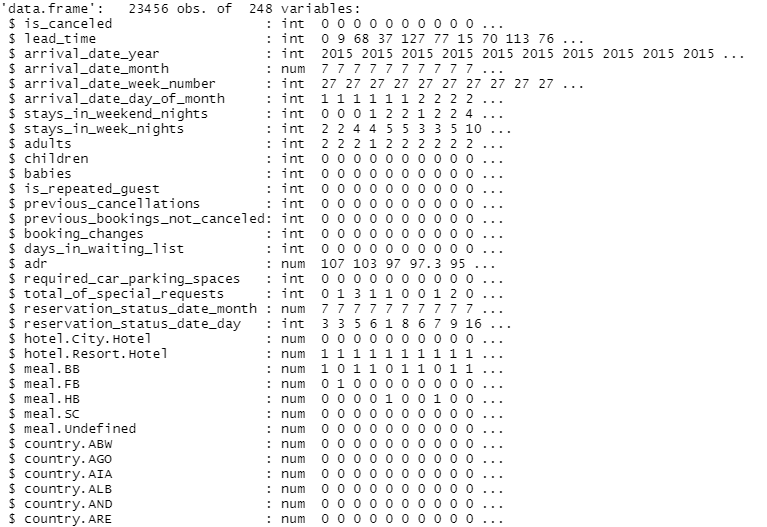
And the column “arrival\_date\_month” is not numeric data, so we transformed the “arrival\_date\_month” column into numeric data. We used “ifelse” command to transform the “arrival\_date\_month” column. The result is April equals 4, May equals 5, June equals 6.

After finishing the “arrival\_date\_month” column, we changed the “reservation\_status\_date” column. The “reservation\_status\_date” column is date data which is not numeric data. So, we extracted the month information and day information to generate two new columns “reservation\_status\_date\_month” and “reservation\_status\_date\_day”.

The next step is removing the duplicate columns “reservation\_status\_date” and “reservation\_status”. Since the “reservation\_status” column has the same information with “is\_canceled”, we need to remove this column.

The last step is creating the dummy variables for the columns which are not numeric type of data. We created the dummy variables for 9 columns that are “hotel”, “meal”, “country”, “market\_segment”, “distribution\_channel”, “reserved\_room\_type”, “assigned\_room\_type”, “deposit\_type”, and “customer\_type” as they will create one single binary column for each of their levels.

When we finished the data cleaning work, we were able to obtain our final data set which can be used to create the model. The final data set contains 23456 observations of 248 variables.



*Model Building*

Before starting the model building steps, we need to get the train data set and test data set. So, we split our final data set with a ratio of 80:20. The 80 percent of the final data set is the train data set, and the 20 percent of the final data set is test data set. Train data is used for model building while test data is used for prediction with the purpose of seeing if the built model works well if future data is added.

**Model 1: K-Nearest Neighboring**

KNN algorithm is a supervised ML algorithm used to find target variable belongs to which class by parsing through nearest neighbors i.e., new data point is compared with the nearest neighbors by taking a k values and hence the class or category of it is decided according to the frequency of neighbors.

Larger k values help to get accurate and less noisy outputs. Smoother decision boundaries can be helpful when we take higher k-values, but it will increase bias and cause discrepancies.

For KNN model, we created 9 models the k value from 3 to 19. 图片包含 表格

描述已自动生成

After the knn models are being created, we used the test data set and the “confusionMatrix” command to generate the confusion matrix.

图片包含 文本

描述已自动生成

The confusion matrixes have 9 accuracy results for each knn model. So, we set up the table for each k value and accuracy.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K value | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 |
| Accuracy | 0.8155 | 0.819 | 0.806 | 0.7991 | 0.7896 | 0.7832 | 0.7693 | 0.7627 | 0.7546 |

*Table 2.* Table of Accuracies with multiple K values

From Table 2, we can find the trend that when K value is less than 5, the accuracy increases as K value increases. When K value is large than 5, the accuracy decreases as K value increases. Also, we can find the biggest accuracy is when k equals 5.

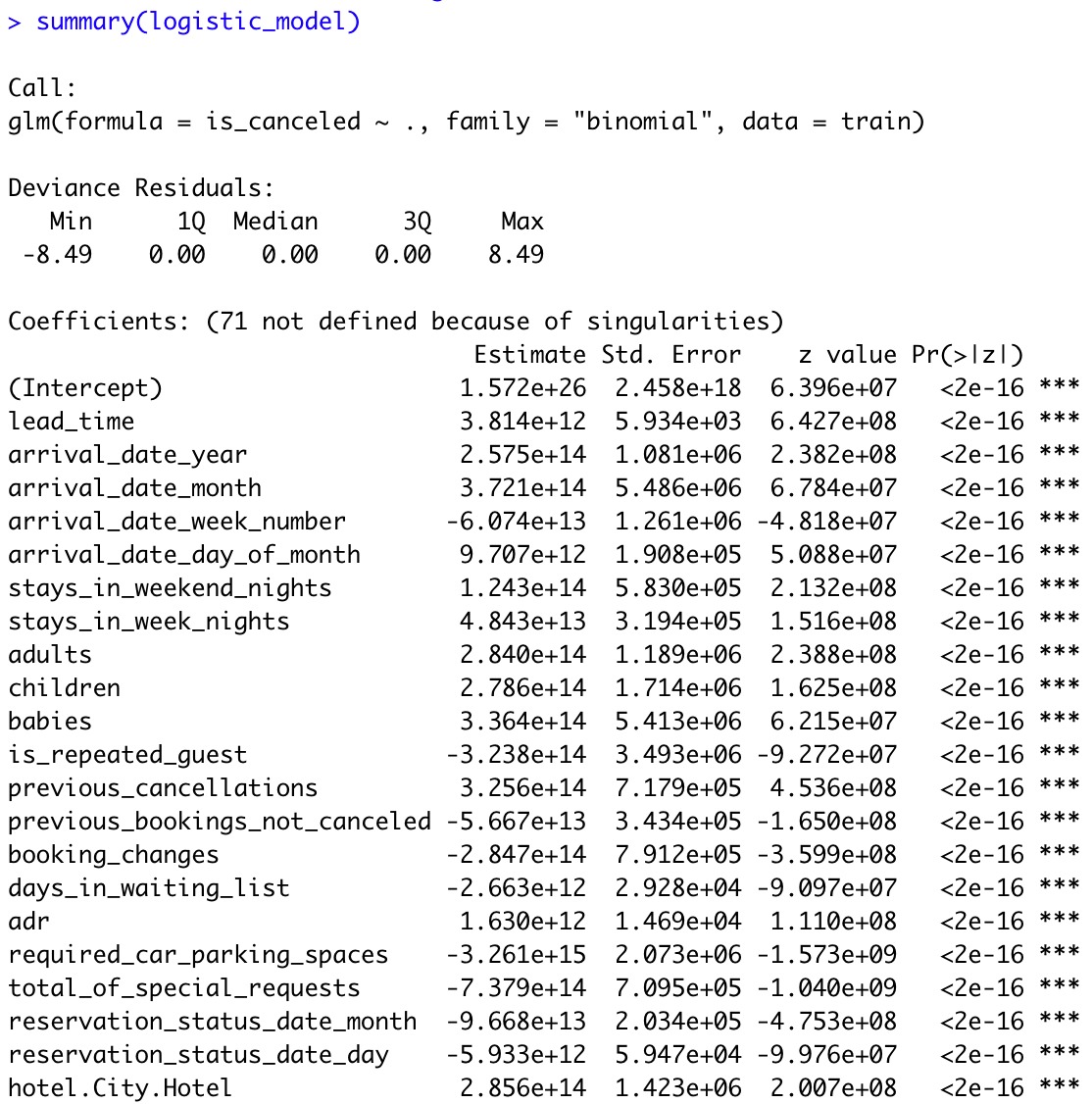
Therefore, the best KNN model is when K equals to 5, and the highest accuracy we generated is 81.9%.

**Model 2: Logistic Regression**

Besides KNN, another predictive algorithm that we used for our dataset is Logistic Regression. Logistic regression is used mainly when dependent variable is binary, and output is obtained in Yes/No format. In this data our target variable is ‘is\_canceled’ which helps to answer business question if bookings of hotel industry will be canceled or not. Logistic regression helps to find change in dependent variable when one or more independent variable values are adjusted. Hence, this technique can be used to determine relationship between one dependent and one or more independent variables. It helps to find correlation among variables in the dataset. There should not be NA variables in data as it can cause discrepancies in output. To build it, we used the glm() command which stands for “generalized linear models”, and set the family type into binomial that links to logistic regression model based on our train data.



Inside this model building, is\_canceled before ~ indicates it is target variable and everything after dot(.) indicates they are independent variables. Binomial family indicates output will be in the form of 0 (bookings not canceled) or 1 (bookings canceled). Then, we output the results by using the summary() command.

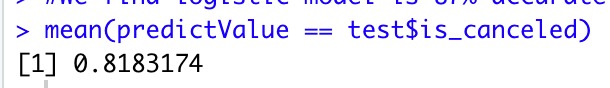


As seen in the above figure summary indicates which variables contribute significantly to model fitting process. Star sign beside each column shows variables significant at 90% CI (\*),95% CI (\*\*) or 99% confidence intervals (\*\*\*). Hence, this can be helpful to get estimates of each independent variables which can be considered or ignored for further model building process.

Furthermore, we used the predict() function to make predictions on test data and obtained the accuracy of our logistic model.

文本

描述已自动生成



Built model is checked if it will work well for future data or not. Prediction is done on test data and probabilities > 0.5 are checked. This helps to check relationship between actual and predicted values. As seen in above figure we find mean i.e. accuracy of the model is 81.83%.

**Model 3: Decision Tree**

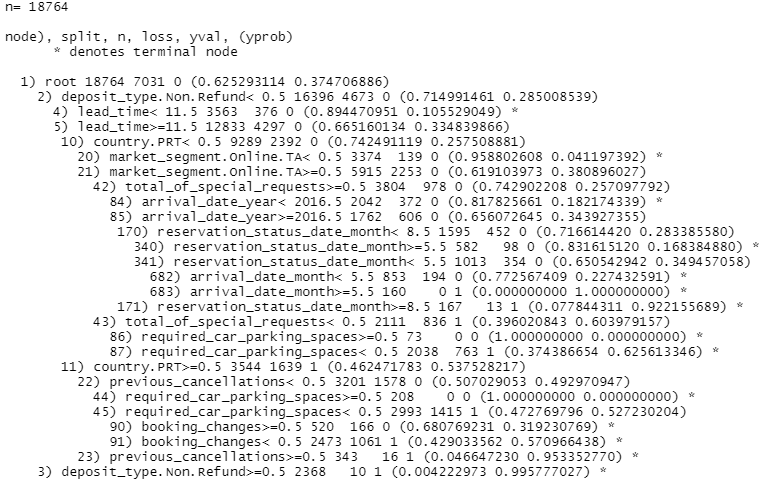
Apart from the KNN algorithm and the Logistic Regression algorithm, we also tried using another important predictive method which is Decision Tree, in our case, is classification tree since our predictor “is\_cancelled” is a binary variable. According to “Decision Trees in R Programming”(2020, April 5), Decision Tree algorithm is a type of useful supervised Machine learning technique and it is characterized by nodes and branches, where the tests on each attribute are represented at the nodes, the outcome of this process is represented at the branches and the class labels are represented at the leaf nodes. It works as a tree-like model based on multiple decisions that are used to generate their probable outcomes. This method is widely used since it is easy to explain.

Among various types of decision trees including M5, C4.5, C5.0 and others, we will use the CART modeling which is expanded as Classification and Regression Trees to grow our tree and display it graphically.

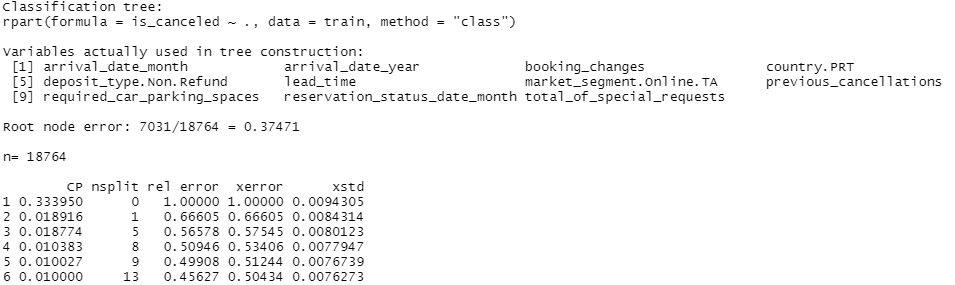
To grow our classification tree, we first converted the response variable which is an integer into a factor, then we installed and called the rpart() library and started growing the tree model by setting the formula as outcome variable: “is\_canceled” and all other attributes in our train data set, the method into “class” which stands for classification tree, and lastly specifies the data frame which is train data set.



After our tree is done, we can print the results down below.



And to be more specific, we used printcp() to print the table that contains complexity parameters and root node error.

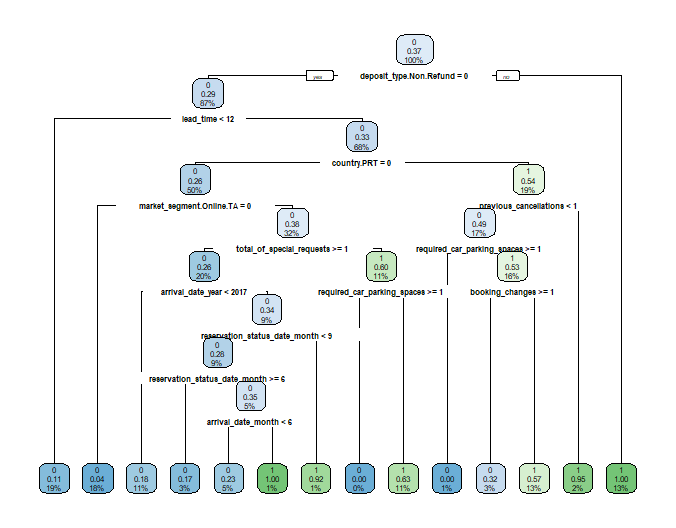


The results show that there are 11 attributes that impact the cancellation to create the decision tree model, and the root node error is around 0.37.

Besides, based on the variables used are as follows:

1. arrival date month
2. arrival date week year
3. booking changes
4. country PRT(Portugal)
5. deposit type non refund
6. lead time
7. market segment online TA
8. previous cancellations
9. required car parking space
10. reservation status date month
11. total of special requests

Moreover, we can visualize it by loading a library named rpart.plot(). And now the tree is showing below.

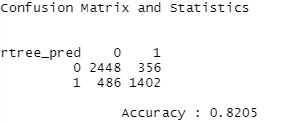


*Figure 1.* Decision Tree Plot

From the plot, we can observe that customers who have deposit type of non-refund is the top condition. If it equals to 1, means deposit is not refundable, then customers would 100% cancel their booking. Else, customers who select no deposit or deposit is refundable their booking. While this may not be true in real life since people tend not to cancel their booking if the bookings are not refundable since they cannot get the deposit back if they cancel it.

And if the deposit type of non-refund equals to 0 which means no deposit or refundable are selected and the lead time is greater than 12 days and people from Portugal, then there is 54% of customers canceled their booking. Based on this condition, if more than one previous cancellation occurred, the chance of cancellation will be 95%. More decisions can be generated through this plot.

Upon finishing our decision tree, we started predicting on our test data by calling the predict() function and specified the type of our prediction is classification. Also, to check the performance and accuracy of our model, we applied the confusion matrix function. To accomplish this, we first installed the Caret package. Then, we used the confusionMatrix() function and below is the result:

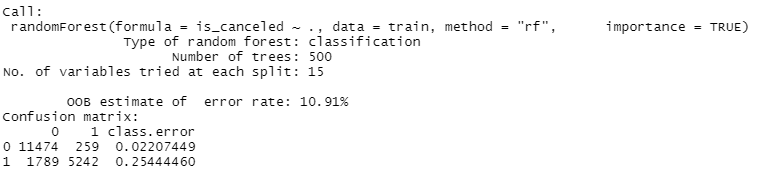


We can observe that in the class of 0, there are about 356 observations that are misclassified into class of 1, and in the class of 1, there are about 486 observations that are misclassified into class of 0. The overall accuracy we generated is 82.05% which is good.

**Model 4: Random Forest**

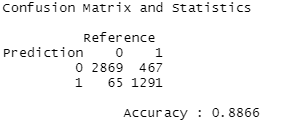
Since decision trees normally do not have the same level of predictive accuracy as other methods since they are not quite robust, and in order to minimize both error due to bias and error due to variance, we applied the random forest algorithm to our dataset. Similar to decision tree, random forest modelling works by creating multiple decision trees and then combining the output generated by each of the decision tree to better predict our outcome variable.

To build a random forest model, we installed and loaded the random forest library and set the method into rf, also we want to display the importance tables, so we set the importance into TRUE. Then, we looked at the result by looking at two parameters: one is number of trees N and another is mtry as number of variables available for splitting at each tree node. By default, N is 500 and mtry is 15.

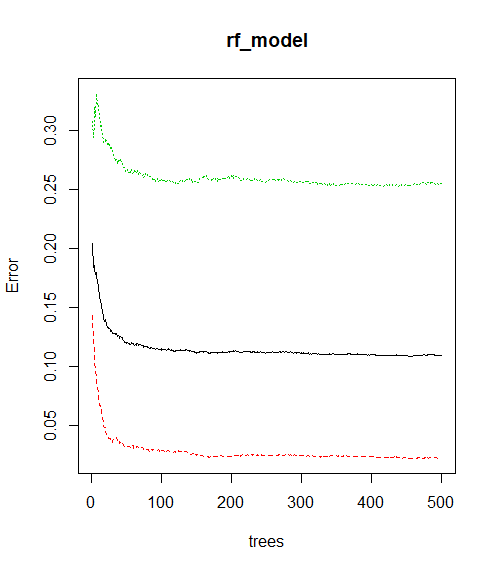


We observe that in the class of 0, there are about 259 observations that are misclassified into class of 1, and in the class of 1, there are about 1789 observations that are misclassified into class of 0 which is a lot. The classification error of class 1 is 0.2544 which is a lot higher than the classification error of class 0 that is 0.022. And the out of bag error is 10.91%.

Then we predicted on the test data and used the confusionMatrix() function to obtain the accuracy that is 88.66%.

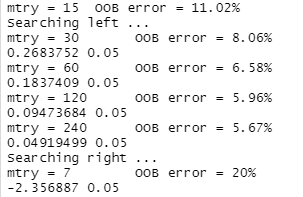


To optimize our random forest model, we plotted the error rate in terms of number of trees below.

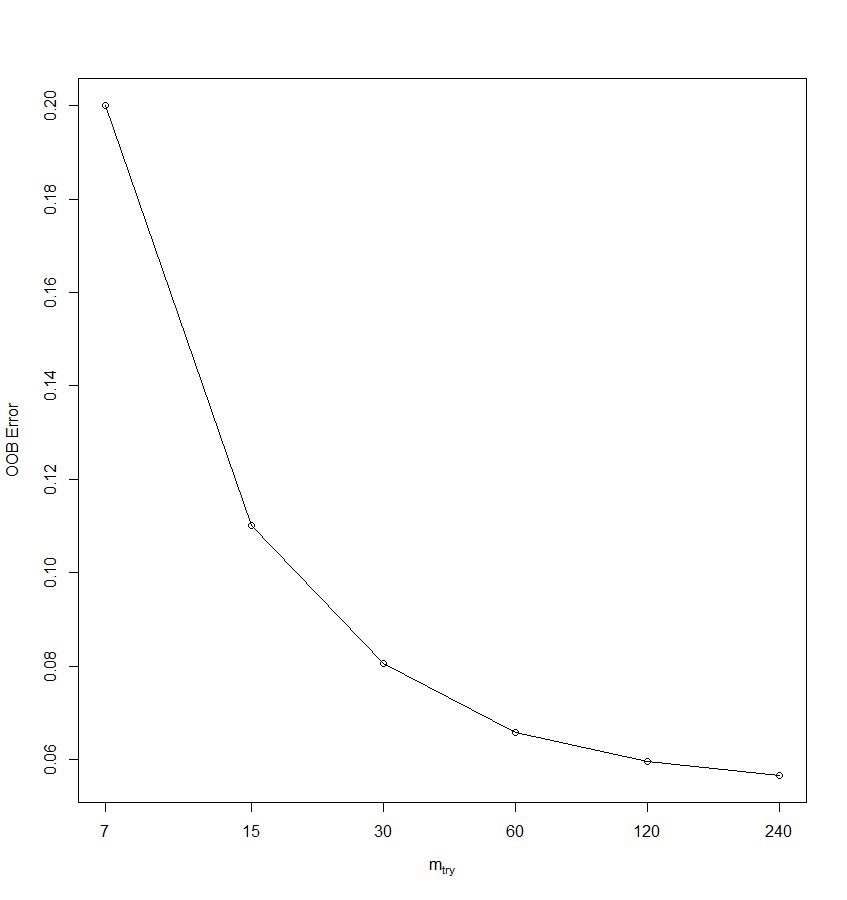


*Figure 2.* Error Rate Plot of Random Forest Model

From Figure 2, the red curve is the error rate for the Not Canceled (“is\_canceled” = 0), the green curve is the error rate for the Canceled (“is\_canceled” = 1), and the black curve is the Out-of-Bag error rate. We observe that the error rate tends to be stabilized as the number of trees gets after 200. We then used tuning algorithm to find the best mtry in terms of the lowest Out of bag error and we can observe from the right graph that when the mtry value equals to 240, it has the lowest OOB error which is 5.67%.

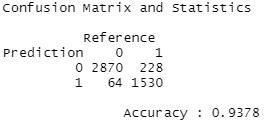


We can further visualize the results into the following plot.



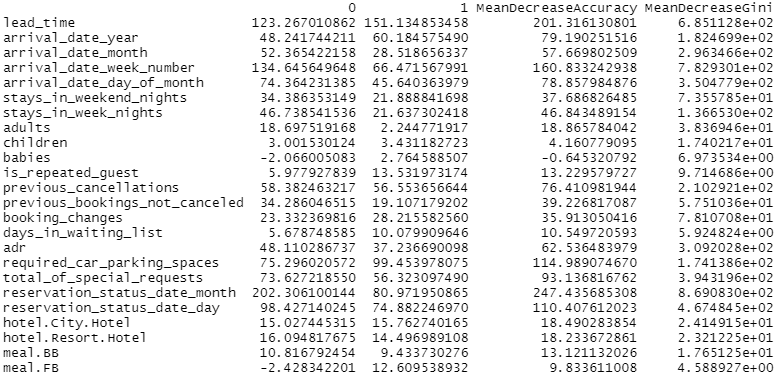
*Figure 3.* Plot of mtry vs OOB error

Figure 3 shows the most optimal mtry value is 240, with it, we obtained the lowest error rate of 5.67%. Now we rebuilt a model based on the train data with the number of trees equals to 200 and the mtry value equals to 240.

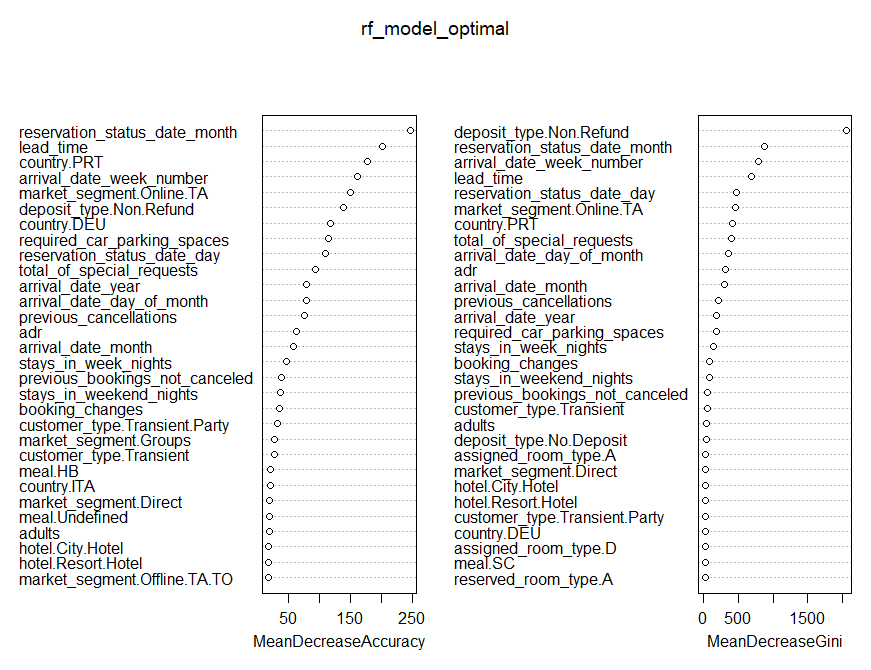


We observe that in the class of 0, there are about 228 observations that are misclassified into class of 1, and in the class of 1, there are about 64 observations that are misclassified into class of 0. With the confusionMatrix() function on the test data, we obtained the optimal accuracy that is 93.78%. The result showed that the accuracy has been improved than the previous random forest model.

We also used the importance() and varImpPlot() functions to check and plot the importance of independent variables.



*Figure 4.* Importance of variables



*Figure 5.* Plot of Random Forest Important Variables

Figure 4 and 5 show the importance of each variable in the model by the accuracy and the Gini Index.

From these two graphs, we observe that “deposit type non refund”, “lead time”, “reservation date month”, “arrival date year” and “country of Portugal” are most relevant to our optimized random forest model’s accuracy.

**DATA ANALYSIS**

Based on the results of all four models, we generating all findings and made a comparison table of 4 models’ accuracies below.

|  |  |  |
| --- | --- | --- |
| **Models** | **Accuracy** | **Error Rate** |
| KNN | 81.9% | / |
| Logistic Regression | 81.83% | / |
| Decision Tree | 82.05% | 37.471% |
| Random Forest(default) | 88.66% | 10.91% |
| Random Forest(optimized) | 93.78% | 5.67% |

*Table 3.* Comparison Table of Accuracies of all models

From Table 3, the best suited model for this dataset is random forest as it optimized the decision tree by creating multiple decision trees and combining the output generated by each decision tree to better predict our outcome variable. Random forest helps to reduce the correlation between the decision trees and removes the bias that decision trees might introduce in the system. Hence, tuning and optimization helps to improve decision tree’s accuracy and lower its error rate.

Compared to the random forest model with the default setting with optimized random forest model, the OOB error has decreased about 5%. And compared to the decision tree model, the accuracy of the optimized random forest model has improved about 11%.

**DISCUSSION**

Analysis of this dataset using various model building techniques helps to find significant factors causing change in bookings of hotel industry. The analysis also helps to answer various business questions like:

* **What main factors that leading to booking and cancellations?**

This correlation and dependencies will help the hotel industry to increase their revenue by planning arrangements of rooms, services in such a way that they provide better customer service thereby earning more profit.

The performance of the KNN and Logistic Regression models were significantly lower than the Decision Tree models and Random Forest models. However, the highest accuracy we obtained was with the Random Forest Model.

*Recommendations*

We recommend utilizing Random Forests for the highest accuracy, but Decision Tree models can be used as aid for non-technical managers since these are easy to interpret and understand.

We also recommend reducing lead time to decrease the number of cancellations. Also, non-refund deposit type could reduce the cancellation rate since it has negative impact on the hotel’s sales.

There are also some limitations of this study as this analysis is done on 20% of original data so that the accurate predictions may vary when data is trained and tested on larger samples. Also, as random forest is best suggested method the limitation is longer processing time. Random forest is suitable for this dataset as number of observations used for model building were comparatively less. Some further analysis like duplicate columns may be analyzed further to decide their need for model building before removing them.

**REFERENCES**

https://www.kaggle.com/jessemostipak/hotel-booking-demand.

*Decision Tree in R Programming.* (2020, April 5). Retrieved December 11, 2020 from https://www.geeksforgeeks.org/decision-tree-in-r-programming/