

ALGEBRA UNIVERSITY COLLEGE

Z A G R E B

Student:

Mateja Aristan

PROJECT

- Hotel booking demand -

MACHINE LEARNING METHODS PROJECT

Mentor:

Lovro Sindičić

Zagreb, 2020

Problem description

Reservations are an essential part of the hotel industry and they represent a contract between a customer and the hotel. This contract gives the customer the right to use the service in the future at a settled price, usually with an option to cancel the agreement before the service provision. The problem occurs when we cannot accurately estimate booking cancellations. Nowadays, with more online reservations usage, there is also a higher cancellation rate because it is relatively easy for users to cancel their bookings. Hotels often implement a combination of overbooking and cancellation policies to manage the risk of cancellations. Overbooking can harm both the hotel's reputation and revenue, and possibly the potential loss of future income from dissatisfied customers who will not book again to stay at the hotel. Cancellation policies, especially non-refundable policies, can reduce the number of bookings. Predicting reservations that can be canceled and preventing these cancellations would create a much higher value for the hotels and help with their successful operation. Results allow hotel managers to predict net demand accurately, build better forecasts, improve cancellation policies, define better overbooking tactics, and use more assertive pricing and inventory allocation strategies.

Aim and hypothesis

Using the Kaggle dataset (available at <https://www.kaggle.com/jessemostipak/hotel-booking-demand>), we want to build a model to classify bookings with high cancellation probability and using this information to forecast cancellations.

The aim is to anticipate:

- number of canceled reservations in a given period
- what is the probability that a specific reservation will be canceled

Hypotheses:

- part of the attribute specifies the reservation as a candidate for cancellation
- based on the given data, a good enough predictive model can be made with precision greater than 90%.


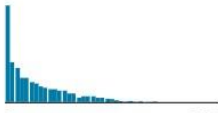
<div> <div>▲ hotel</div> <div>≡</div> </div> <div> Hotel (H1 = Resort Hotel or H2 = City Hotel) </div>	<div> <div># is_canceled</div> <div>≡</div> </div> <div> Value indicating if the booking was canceled (1) or not (0) </div>	<div> <div># lead_time</div> <div>≡</div> </div> <div> Number of days that elapsed between the entering date of the booking into the PMS and the arrival date </div>	<div> <div># arrival_date_year</div> <div>≡</div> </div> <div> Year of arrival date </div>	<div> <div>▲ arrival_date_month</div> <div>≡</div> </div> <div> Month of arrival date </div>
<div> <div>City Hotel</div> <div>66%</div> </div> <div> <div>Resort Hotel</div> <div>34%</div> </div>				<div> <div>August</div> <div>12%</div> </div> <div> <div>July</div> <div>11%</div> </div> <div> <div>Other (92852)</div> <div>78%</div> </div>
Resort Hotel	0	342	2015	July
Resort Hotel	0	737	2015	July
Resort Hotel	0	7	2015	July
Resort Hotel	0	13	2015	July
Resort Hotel	0	14	2015	July
Resort Hotel	0	14	2015	July
Resort Hotel	0	0	2015	July

Image 1. The example of ‘Hotel booking demand’ dataset

This dataset contains booking information for a four-star Resort hotel (H1) and a City hotel (H2) located in Portugal with an accommodation capacity of more than 200 rooms.

Both datasets share the same structure, with 31 variables describing the 40,060 observations of H1 and 79,330 observations of H2. Each observation represents a hotel booking. Both datasets comprehend bookings due to arrive between the 1st of July of 2015 and the 31st of August 2017, including bookings that effectively arrived and bookings that were canceled. Data from the mentioned dataset also includes information such as when the reservation was made, when the reservation was canceled, date of arrival of the guests, the number of nights guests stayed at a hotel, number of guests, where are they from, the cost for a room, length of stay, the number of available parking spaces, etc.

For a better understanding of the dataset, here are some questions to answer:

- 1) How many bookings were canceled?
- 2) Which month has the highest number of visitors?
- 3) What is the monthly average daily rate per person over the year?
- 4) Which country has the most number of hotel visitors?
- 5) Which customer type contributes to the most hotel booking cancellations?
- 6) Which month has the highest number of cancellations?

Materials, methodology and research plan

As a supervised machine learning technique, classification is often used to obtain predictions results from the selected data set. Machine learning classification algorithms used in this project will be Random forest, K Nearest Neighbors, XGBoost, and AdaBoost. The efficiency of the algorithms mentioned will be compared, and the model will be built using cross-validation. Techniques and tools planned to use: Jupyter (Python language) with Pandas packages, Scikit-learn. Using data sets from hotels and addressing booking cancellation prediction as a classification problem in data science, we hope to build a model for predicting booking cancellations with accuracy results of over 90%.

Steps:

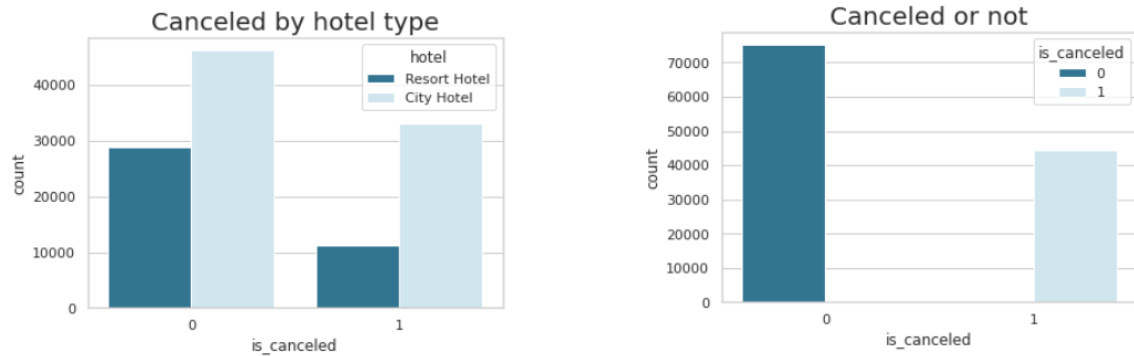
Data Analysis
Data Processing (Checking and handling missing values, Dropping irrelevant features, Filtering out the numerical features, Transformation)
Split the dataset into train and test sets 80%-20%
Fitting Classifiers (RandomForestClassifier, KNeighborsClassifier, XGBoostClassifier, AdaBoostClassifier)
Model Selection

Project steps

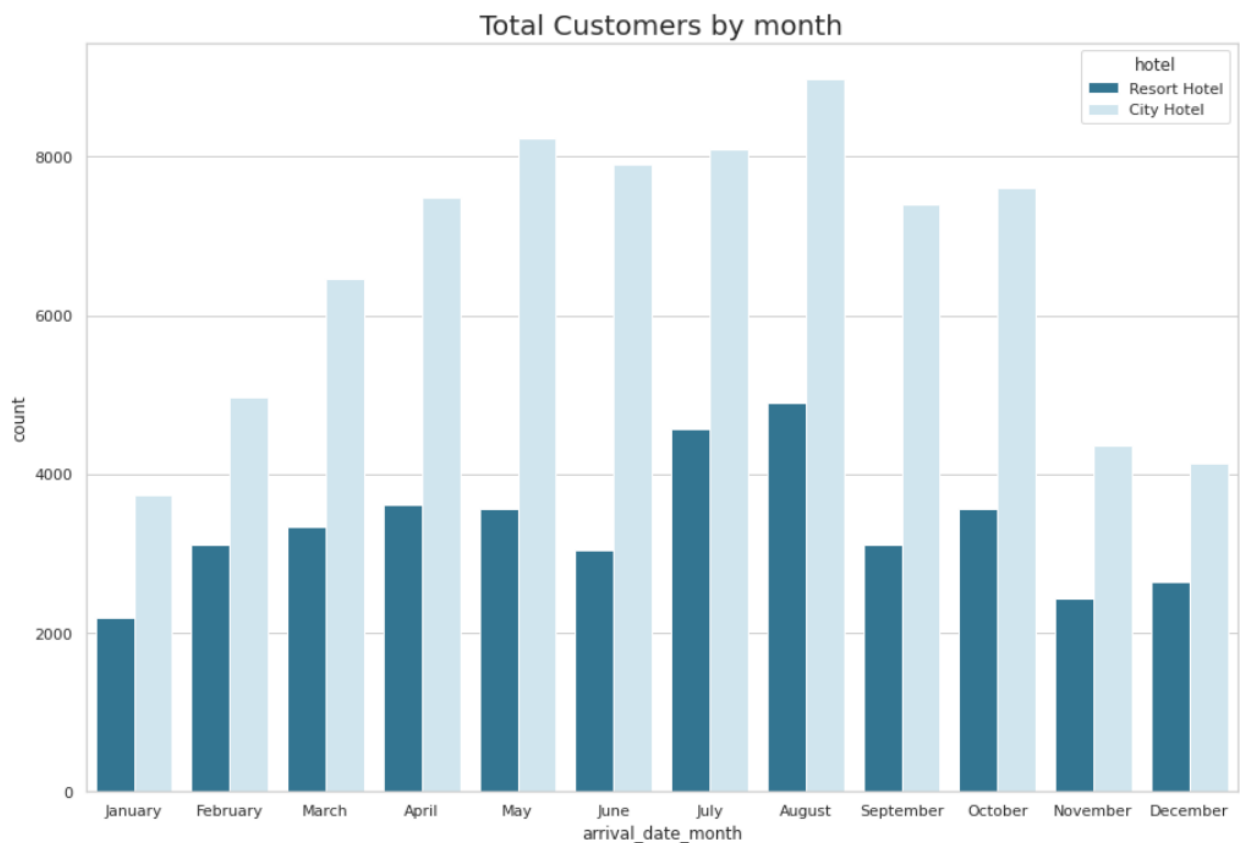
After importing the dataset, it was shown that the dataset contains 119390 rows and 32 columns. Columns are:

```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',  
      'arrival_date_month', 'arrival_date_week_number',  
      'arrival_date_day_of_month', 'stays_in_weekend_nights',  
      'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',  
      'country', 'market_segment', 'distribution_channel',  
      'is_repeated_guest', 'previous_cancellations',  
      'previous_bookings_not_canceled', 'reserved_room_type',  
      'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',  
      'days_in_waiting_list', 'customer_type', 'adr',  
      'required_car_parking_spaces', 'total_of_special_requests',  
      'reservation_status', 'reservation_status_date', 'month', 'year',  
      'day'],  
      dtype='object')
```

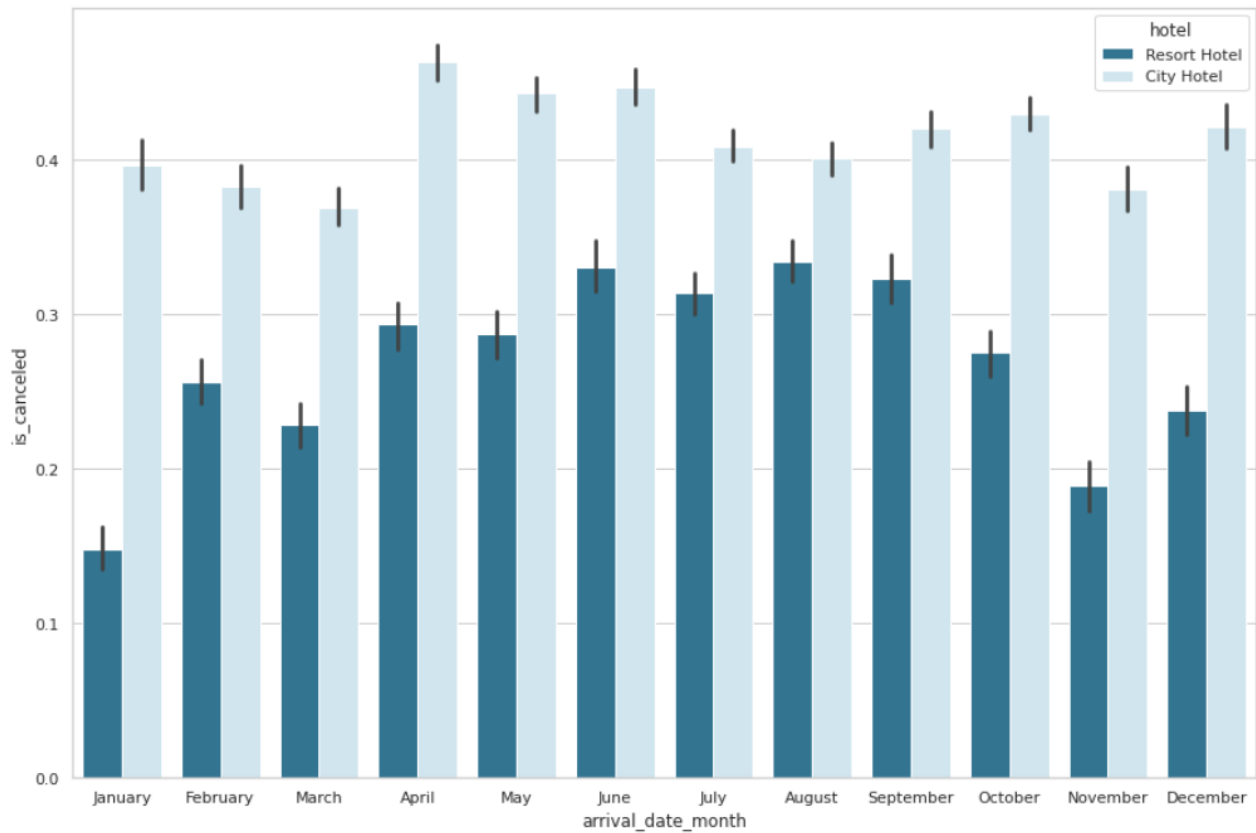
In exploratory data analysis section, I wanted to show distribution of cancelled and not cancelled reservations. In the images below we can see that we have data from a Resort Hotel and City hotel. In x axis we see the 0 represents not cancelled and 1 represents cancelled reservations. On the left image we can see the comparison of cancellations by type of hotel and we can conclude that there are more cancellations in the City hotel. The right image shows distribution of total number of cancellations for both hotels.



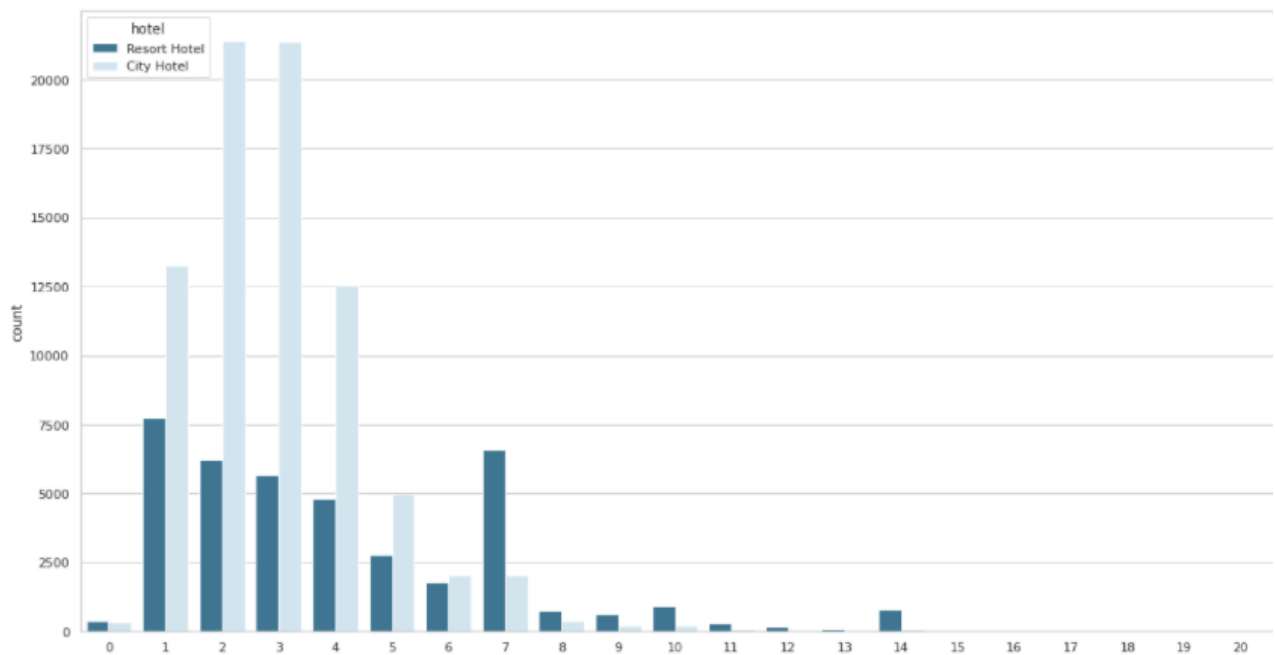
In the next plot total number of customers is presented by their month arrival. Here we can notice that both hotels have the most number of guest during Summer in July and August.



In the image below, I wanted to check also reservation cancellation by months for both City and Resort hotels. Here cancellations ratio is bigger in Spring and Summer and have a lowest ratio in the winter for Resort Hotel, but City hotel has highest ratio of cancellation during Sprint, in April, May and June, the cancellation rate drops from January to March and in November.



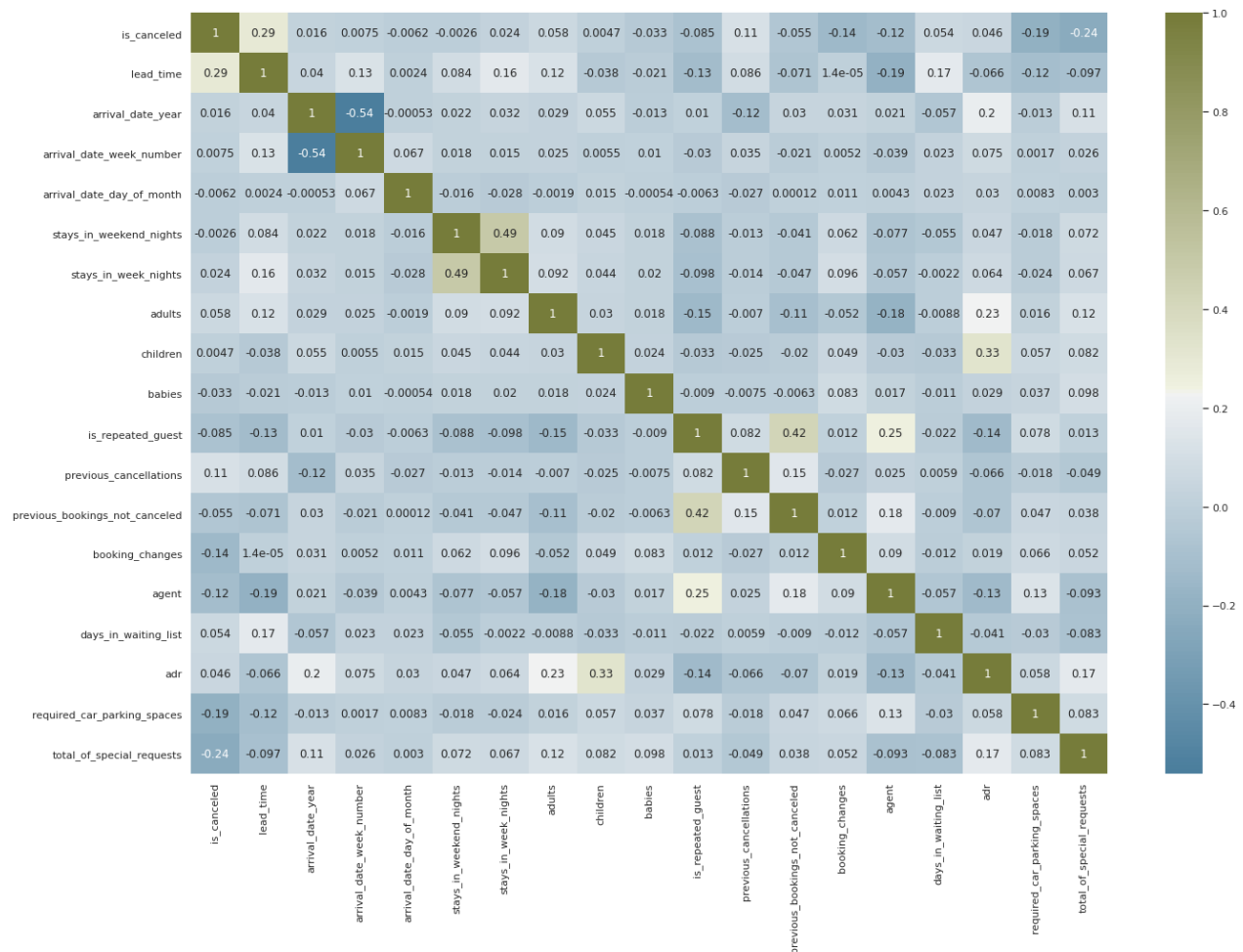
I was also interested to see what is the most often number of days those guests stay in both City and Resort hotel. We can see that in the city hotel people tend to stay more 1 to 4 night, where in resort hotel people tend to stay longer.



After data visualization the data types and null values were explored. After seeing that there are 4 columns ('children', 'country', 'agent' and 'company') that contain null values I decided to explore them further to decide what to do with them. In the screenshot below we can see that there are 94.3% missing values in column '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) and because of that I will drop the column as it is seeming unnecessary to keep it for further work. There are 4 missing values in column 'children', which I will fill out with value 0 taking in consideration that those guests do not have children. In the column 'agent' (ID of the travel agency that made the booking), there are 16340 missing values, or 13.69% of total values. This is probably because the guest didn't make reservation through agency. I will probably fill out those values with random ID "999" in the code below. In the column 'country' there are 0.41% missing values for which I later decided to drop the data.

hotel	0.000000
is_canceled	0.000000
lead_time	0.000000
arrival_date_year	0.000000
arrival_date_month	0.000000
arrival_date_week_number	0.000000
arrival_date_day_of_month	0.000000
stays_in_weekend_nights	0.000000
stays_in_week_nights	0.000000
adults	0.000000
children	0.003350
babies	0.000000
meal	0.000000
country	0.408744
market_segment	0.000000
distribution_channel	0.000000
is_repeated_guest	0.000000
previous_cancellations	0.000000
previous_bookings_not_canceled	0.000000
reserved_room_type	0.000000
assigned_room_type	0.000000
booking_changes	0.000000
deposit_type	0.000000
agent	13.686238
company	94.306893
days_in_waiting_list	0.000000
customer_type	0.000000
adr	0.000000
required_car_parking_spaces	0.000000
total_of_special_requests	0.000000
reservation_status	0.000000
reservation_status_date	0.000000
dtype:	float64

Now that there are no null values in any of the columns, I decided to explore the feature correlation with the cancellation. Looking into heatmap, it is visible there is high correlation between stays_in_weekend_nights and stays_in_week_nights, then between is_repeated_guest and previous_bookings_not_canceled, then adr and children.



```
is_canceled      1.000000
lead_time        0.291940
total_of_special_requests  0.235595
required_car_parking_spaces  0.194801
booking_changes  0.144669
agent            0.118595
previous_cancellations  0.109914
is_repeated_guest  0.085185
adults           0.058408
previous_bookings_not_canceled  0.055495
days_in_waiting_list  0.054008
adr              0.046133
babies           0.032523
stays_in_week_nights  0.024103
arrival_date_year  0.016339
arrival_date_week_number  0.007481
arrival_date_day_of_month  0.006173
children         0.004740
stays_in_weekend_nights  0.002639
Name: is_canceled, dtype: float64
```

After correlation exploration, I decided to drop columns: 'hotel', 'meal', 'country', 'market_segment', 'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type', 'reservation_status', 'arrival_date_month', 'reservation_status_date'.

Further on, the dummiy variables were made for the categorical variables 'hotel', 'meal', 'country', 'market_segment', 'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type', 'reservation_status'.

After which the dataset contained 118902 rows and 241 columns.

The dataset was split into test and train in 70%:30% using sklearn method `train_test_split()`:

```
# Splitting the dataset 70:30 using sklearn method train_test_split()
y = result["is_canceled"]
X = result.drop(["is_canceled"], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 42)
```

For the prediction of a cancellation of a booking, I decided to use different models and compare the results.

- RandomForestClassifier - randomly selects observations (bootstrap samples) and a subset of the features from the train data and constructs a decision tree for every sample. From each decision tree, it will get the prediction results and based on the majority votes of predictions, it averages the results to predict the final output
- KNeighborsClassifier - looks for the nearest neighbors.
- XGBoostClassifier - builds multiple trees on top of each other to correct the prediction errors of the previous tree
- AdaBoostClassifier - It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations.

The Accuracy for all models are presented in table below:

RandomForestClassifier	1.0
KNeighborsClassifier	0.8635866670404531
XGBoostClassifier	1.0
AdaBoostClassifier	1.0
Gradient Boosting Classifier	1.0

References

- 1) <https://www.kaggle.com/jessemostipak/hotel-booking-demand>
- 2) <https://www.sciencedirect.com/science/article/pii/S2352340918315191>
- 3) <https://pdfs.semanticscholar.org/0f5f/3a506360b9be0a7ab52d77974695f1c48a4d.pdf>
- 4) <https://www.datacamp.com/>