Hotel Booking Analysis

Aarti Gade

Kunal Badgujar

Vivek Tanagawade

Data science trainees, Alma Better, Bangalore

Abstract:

Hotels plays a major role in tourism industry. People travel around the world andto relax they definitely need a hotel to stay and have food of different cultures. We had concern of many things related to hotels during our journey.

They are, when the best time of year to book a hotel room is? Or the optimal length of stay in order to get the best daily rate? What if you wanted to predict whether or not a hotel was likely to receive a disproportionately high number of special requests?

In order to achieve this, we can use data visualization method with several datasets and predict the possibility of the bestoutcome for the customer to be get satisfied.

1. Problem Statement

Data of different hotels (i.e., excluding the personal information) like booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things are gathered. Using this information obtain the results which helps the customer to solve different problems like when the best time of year to book a hotel room is? Or the optimal length of stay in order to get the best daily rate?

What if you wanted to predict whether or not a hotel was likely to receive a disproportionately high number of special requests.

This approach builds a customer friendly platform in order to find the best results and solve the multiple problems, The data includes.

- Number of hotels available.
- How many people are cancelled the booking?
- 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, company
- Days in waiting list
- Customer type
- Adr, required car parking spaces
- Total of special requests
- Reservation status
- Reservation status date

2. Introduction

The hotel booking analysis is a customer oriented approach, where they can analyze the solution for different problems which are most often faced by everyone.

In order to achieve the solutions, we require a dataset containing all the information related to the previous booking done by the customers excluding their personal information.

To handle such dataset, we need some python libraries to work with. we need panda's library for, data visualization we use Matplotlib and another is NumPy for mathematical operation where pandas act as a wrapper for these two libraries. We require Seaborn library which is built on top of the matplotlib which used for making statistical graphs. Dataset contains 119390 rows and 32 columns (i.e., 119390, 32).

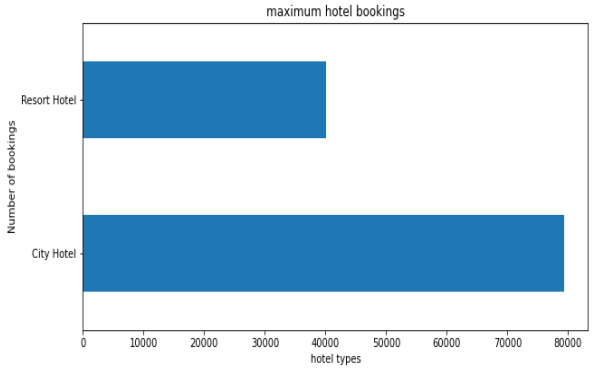
2. List of observations

1. In the given dataset, to find which hotel contain maximum bookings. This can be done by plotting the graphs.

```
[ ]  df.hotel.agg(["value_counts"])
```

	value_counts
City Hotel	79330
Resort Hotel	40060

In the above figure we get the value counts for each hotel bookings.



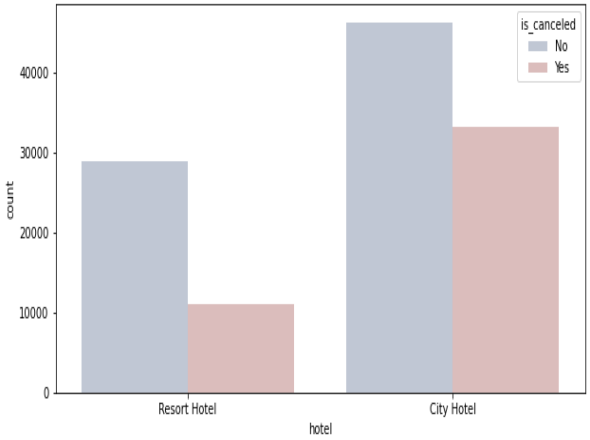
After plotting the graph, we get this figure as an output, this is a bar graph which are labelled with City hotel and Resort Hotel.

2. In the same way we can plot the graphfor bar graph, i.e. for the hotel have the maximum number of cancellation of bookings

```
[ ]  df.groupby(["hotel"])["is_canceled"].agg(["value_counts"])
```

		value_counts
hotel	is_canceled	
City Hotel	No	46228
	Yes	33102
Resort Hotel	No	28938
	Yes	11122

In the above figure we can observe that we group each hotel with is canceled either yes or no and get the total count respectively.



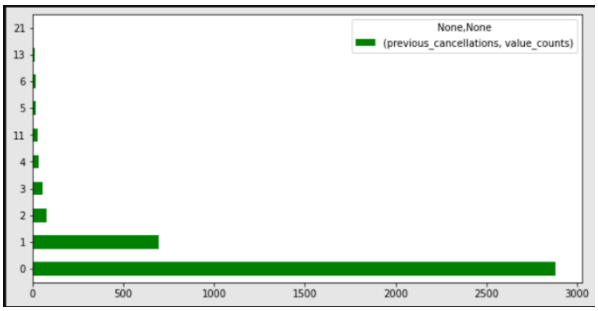
This is the graph which indicates the person is cancelled the booking for Resort or City hotel. We have the count of cancelled and not cancelled list of each hotel from previous figure.

3. To count the previous cancellation by repeated guests. In order to get this, we need to check is_repeated_guest is Yes or No.

```
[ ] df[["previous_cancellations"]][df["is_repeated_guest"] == "Yes"].agg(["value_counts"])
```

	previous_cancellations
	value_counts
0	2883
1	693
2	76
3	52
4	31
11	27
5	19
6	16
13	12
21	1

The above figure indicates previous cancellation and their total value counts respectively.

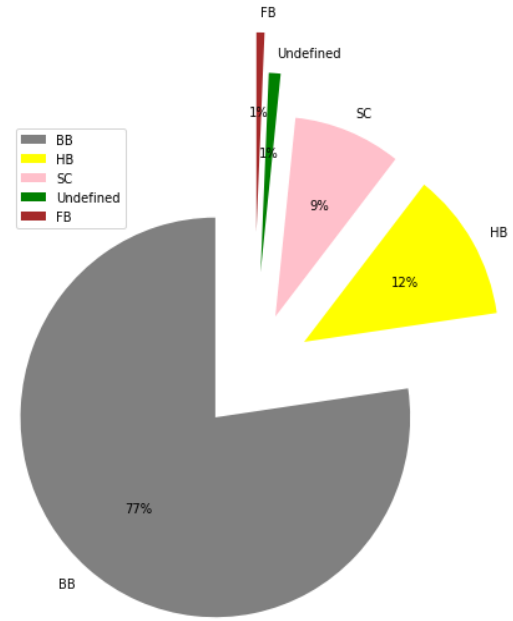


Plotting the graph for the above table which provides total value counts for previous cancellation.

4. Now we need to analyze the preference of the guest like what they basically prefer.

```
[ ] df["meal"].agg(["value_counts"])
```

	value_counts
BB	92310
HB	14463
SC	10650
Undefined	1169
FB	798



In the above figure we can see that totalmeal counts are displayed for BB, HB, SC FB and Undefined respectively.

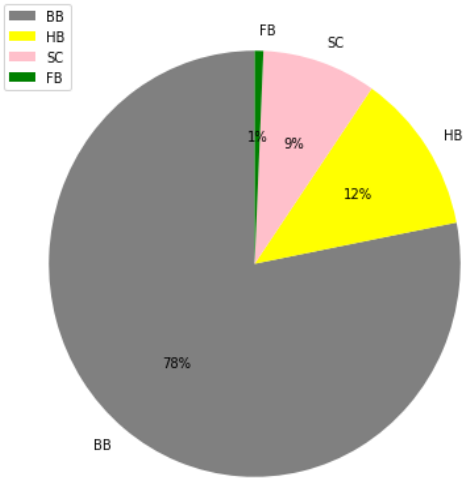
In this figure a pie graph is plotted for the previous data table which provides total meal counts for respective data frames.

Like the same way,

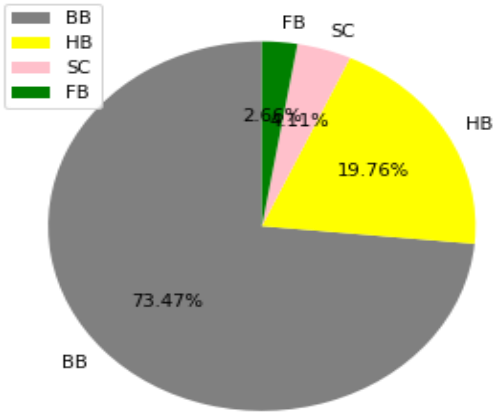
- Adults favourite and least favourite meal is.

```
[ ] df.iloc[np.where((df["adults"] > 0) & (df["meal"] != "Undefined"))]["meal"].agg(["value_counts"])
```

	value_counts
BB	92020
HB	14454
SC	10546
FB	798



- The babies favorite and least favorite meals

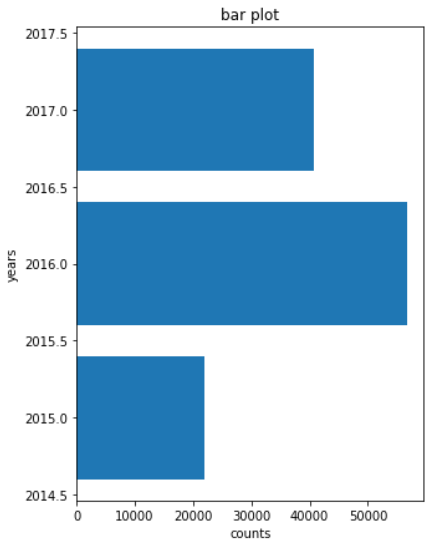


5. The most busy year could be found by using arrival date year field. Using agg function we get the total value count for arrival date year.

```
[ ] df.arrival_date_year.agg(["value_counts"])
```

	value_counts
2016	56707
2017	40687
2015	21996

There are three data fields obtained after running the above code. 2016,2017 and 2016 along with their value count.



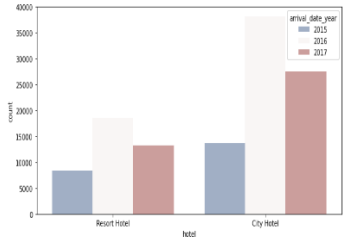
Upon plotting a graph for the arrival date year total value count we get this as a output.

6. How many guests arrived year-wise could be found by grouping hotel into arrival date year and fetching total view count by using agg function.

```
df.groupby(["hotel"])["arrival_date_year"].agg(["value_counts"])
```

hotel	arrival_date_year	value_counts
City Hotel	2016	38140
	2017	27508
	2015	13682
Resort Hotel	2016	18567
	2017	13179
	2015	8314

The above figure shows the total view count for arrival date year for respective hotels.
Upon plotting the graph we get,

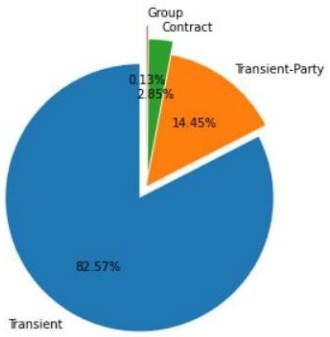


7. Which type of customers are more likely to cancel the booking, this can be get by mapping customer type with is cancelled field is yes.

```
df["customer_type"][df["is_canceled"] == "Yes"].agg(["value_counts"])
```

	value_counts
Transient	36514
Transient-Party	6389
Contract	1262
Group	59

The above figure gives the total count for the customers are more likely to cancel the booking.

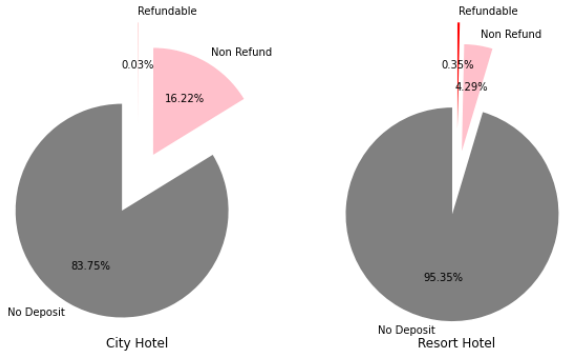


Above graphs are obtained from the table results.

8. Deposit Type hotel-wise, to get this group by is to be done for the hotel and deposit type and finding the total value count.

```
df.groupby(["hotel"])["deposit_type"].agg(["value_counts"])
```

hotel	deposit_type	value_counts
City Hotel	No Deposit	66442
	Non Refund	12868
	Refundable	20
Resort Hotel	No Deposit	38199
	Non Refund	1719
	Refundable	142



9. Number of guests who had not cancelled their booking, this can be achieved by using Is Cancelled is No and mapping with country. Using agg function total value count is obtained.

```
[ ] df[df["is_canceled"] =="No"]["country"].agg(["value_counts"])
```

	value_counts
PRT	21071
GBR	9676
FRA	8481
ESP	6391
DEU	6069
...	...
BHR	1
DJI	1
MLI	1
NPL	1
FRO	1

166 rows × 1 columns

10. From where the most guests coming,this can be easily achieved by total value count for each country.

```
df["country"].agg(["value_counts"])
```

	value_counts
PRT	48590
GBR	12129
FRA	10415
ESP	8568
DEU	7287
...	...
DJI	1
BWA	1
HND	1
VGB	1
NAM	1

178 rows × 1 columns

11. Maximum number of stays in week nights in each hotel, this can be achieved by grouping by hotel with stays_in_week_night field and finding the max value by agg function.

```
[ ] df.groupby(["hotel"])["stays_in_week_nights"].agg(["max"])
```

hotel	max
City Hotel	41
Resort Hotel	50

12. Maximum number of stays in weekend nights in each hotel, this can be achieved by grouping by hotel with stays_in_week end night field and finding the max value by agg function.

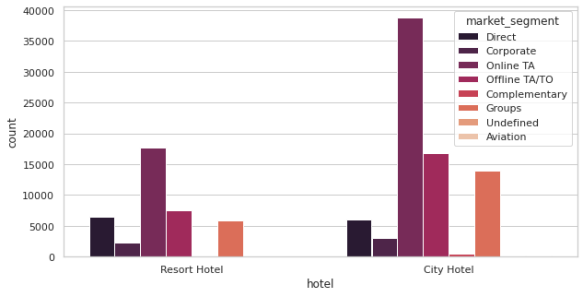
```
[ ] df.groupby(["hotel"])["stays_in_weekend_nights"].agg(["max"])
```

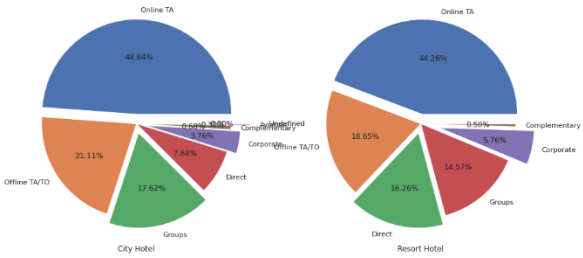
hotel	max
City Hotel	16
Resort Hotel	19

13. From where most of the bookings were made, this can be achieved by grouping by hotel with market segment field andfinding the value count by agg function.

```
[ ] df.groupby(["hotel"])["market_segment"].agg(["value_counts"])
```

hotel	market_segment	value_counts
City Hotel	Online TA	38748
	Offline TA/TO	16747
	Groups	13975
	Direct	6093
	Corporate	2986
	Complementary	542
	Aviation	237
	Undefined	2
Resort Hotel	Online TA	17729
	Offline TA/TO	7472
	Direct	6513
	Groups	5836
	Corporate	2309
	Complementary	201





The above graphs are plotted by the results obtained from the table in previous figure.

Conclusion:

After all the data visualization it's being concluded that data of different hotels (i.e., excluding the personal information) like booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things are gathered.

Using this information obtain the results which helps the customer to solve different problems like when the best time of year to book a hotel room is? Or the optimal length of stay in order to get the best daily rate? What if you wanted to predict whether or not a hotel was likely to receive? disproportionately high number of special requests? etc. Are easily identified and a proper way of analyzing the huge datasetand converting into easily understandable method will be a better choice.

Challenges:

(1) There was a lot of duplicate data.
(2) Data was present in wrong datatype format.
(3) Choosing appropriate visualization techniques to use was difficult.
(4) A lot of null values were there in the dataset.

References-

- Pandas user guide: https://pandas.pydata.org/docs/user_guide/index.html
- Matplotlib user guide: https://matplotlib.org/3.3.1/users/index.html
- Seaborn user guide & tutorial: https://seaborn.pydata.org/tutorial.html