Table of Contents

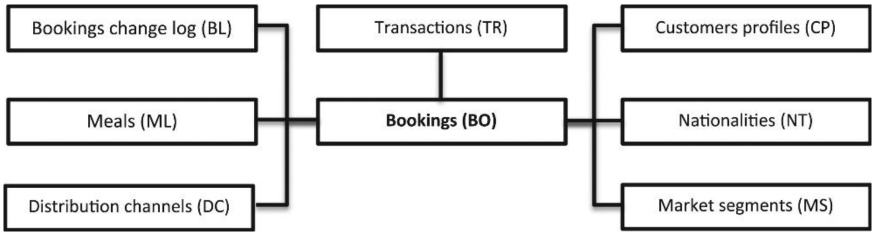
INTRODUCTION

A hotel is an establishment that provides paid accommodation, generally for a short duration of stay. Hotels often provide a number of additional guest services, such as restaurants, bars, swimming pools, healthcare, retail shops; business facilities like conference halls, banquet halls, boardrooms; and space
for private parties like birthdays, marriages, kitty parties, etc.

With increase in the spending capacity of the people, both leisure and business travel spending has seen YoY growth over the past 5 years. The hotel industry has been consistently growing with The Global Hotel Industry revenue in 2016/17 at a staggering value of $550 Bn and increasing in 2018, 2019 and 2020.
There are various factors that affect the day-to-day business in this industry. Some of the major factors affecting the business are high competition between hotels, improper marketing strategy, ineffective pricing model, customer dissatisfaction due to inability to satisfy their requests.

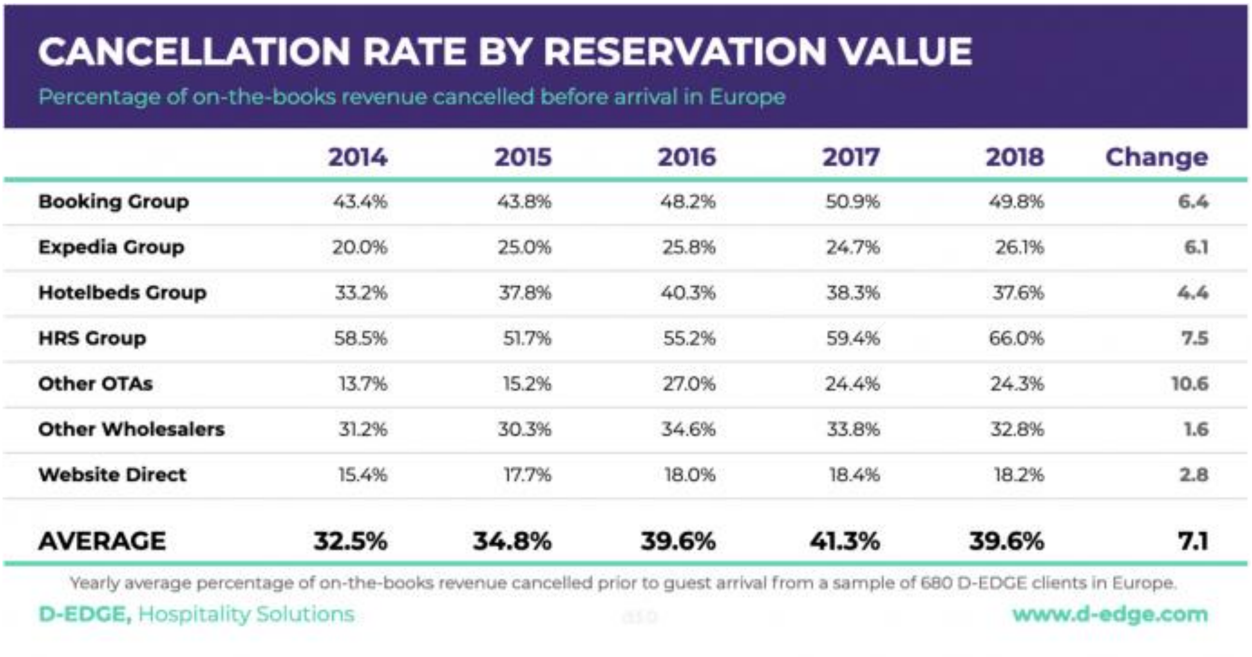
Dataset Information

This data article describes two datasets with hotel demand data. One of the hotels (H1) is a resort hotel and the other is a city hotel (H2). 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 cancelled. Since this is hotel real data, all data elements pertaining hotel or costumer identification were deleted. Instead of directly extracting variables from the bookings database table, when available, the variables' values were extracted from the bookings change log, with a timestamp relative to the day prior to arrival date. Not all variables in these datasets come from the bookings or change log database tables. Some come from other tables, and some are engineered from different variables from different tables. A diagram presenting the PMS database tables from where variables were extracted is presented in Fig: -



Problem Statement

Hotel cancellations have been rising in the past few years due to the presence of online websites (OTA) like Booking.com, expedia.com, etc. Numerical data shows that with the exception of 2018, every single channel has observed a marked increase in cancellation rate YoY. And, even in 2018, the number was 7.1 points above 2014.



CANCELLATION RATE BY RESERVATION VALUE

Percentage of on-the-books revenue cancelled before arrival in Europe

	2014	2015	2016	2017	2018	Change
Booking Group	43.4%	43.8%	48.2%	50.9%	49.8%	6.4
Expedia Group	20.0%	25.0%	25.8%	24.7%	26.1%	6.1
Hotelbeds Group	33.2%	37.8%	40.3%	38.3%	37.6%	4.4
HRS Group	58.5%	51.7%	55.2%	59.4%	66.0%	7.5
Other OTAs	13.7%	15.2%	27.0%	24.4%	24.3%	10.6
Other Wholesalers	31.2%	30.3%	34.6%	33.8%	32.8%	1.6
Website Direct	15.4%	17.7%	18.0%	18.4%	18.2%	2.8
AVERAGE	**32.5%**	**34.8%**	**39.6%**	**41.3%**	**39.6%**	**7.1**

Yearly average percentage of on-the-books revenue cancelled prior to guest arrival from a sample of 680 D-EDGE clients in Europe.

D-EDGE, Hospitality Solutions www.d-edge.com

OTA´s are encouraging customers to cancel by providing services like book now and cancel later, free of charge, whenever you want. This results in customers booking more than one hotel and making a final decision for their stay later on, hence resulting in cancellations. Examples of the impact of cancellations on a hotel:

- Loss of revenue when they cannot resell the room.
- Additional costs of distribution channels by increasing commissions or paying for publicity to help sell these rooms.
- Lowering prices last minute, so they can resell a room, resulting in reducing profit margin.

So, what can hotels do to reduce this uncertainty and maximize their product and revenue? A lot can be done with revenue management techniques when it comes to rates restrictions, like increasing the number of days until the customer can cancel their booking free of cost, hence giving you more time to resell the room. But now due to the presence of competition, we need to monitor these policies so that we don't loss customers due to stricter cancellation policies compared to our competitors Therefore, it would seem that we have a complex problem and not a viable solution. However, now we can use data science and machine learning techniques, to accurately predict which individual and specific reservations are going to cancelled.

Variable Categorization with Description

The dataset consists of 32 variables. Out of these variables 31 are independent variables and 1 is a target variable. The variables are a mixture of both numerical and categorical type.

Numerical

Sr No.	Variable	Datatype	Description
1	is_canceled	double	Value indicating if the booking was canceled (1) or not (0)
2	lead_time	double	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
3	arrival_date_year	double	Year of arrival date
4	arrival_date_week_number	double	Week number of year for arrival date
5	arrival_date_day_of_month	double	Day of arrival date
6	stays_in_weekend_nights	double	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
7	stays_in_week_nights	double	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
8	adults	double	Number of adults
9	children	double	Number of children
10	babies	Double	Number of babies
11	is_repeated_guest	double	Value indicating if the booking name was from a repeated guest (1) or not (0).
12	previous_cancellations	double	Number of previous bookings that were cancelled by the customer prior to the current booking.
13	previous_bookings_not_canceled	double	Number of previous bookings not cancelled by the customer prior to the current booking.
14	booking_changes	double	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
15	days_in_waiting_list	double	Number of days the booking was in the waiting list before it was confirmed to the customer
16	adr	double	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights

17	required_car_parking_spaces	double	Number of car parking spaces required by the customer
18	total_of_special_requests	double	Number of special requests made by the customer (e.g., twin bed or high floor)
19	reservation_status_date	double	Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel

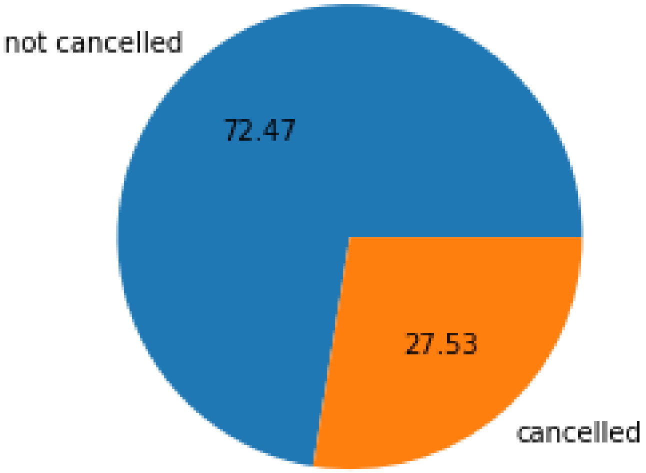
Categorical

Sr No.	Variable	Datatype	Description
1	hotel	character	Hotel (H1 = Resort Hotel or H2 = City Hotel)
2	arrival_date_month	character	Month of arrival date
3	meal	character	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
4	country	character	Country of origin. Categories are represented in the ISO 3155–3:2013 format
5	market_segment	character	Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
6	distribution_channel	character	Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”
7	reserved_room_type	character	Code of room type reserved. Code is presented instead of designation for anonymity reasons.
8	assigned_room_type	character	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.
9	deposit_type	character	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit – no deposit was made; non-Refund – a deposit was made in the value of the total stay cost; Refundable – a deposit was made with a value under the total cost of stay.

10	agent	character	ID of the travel agency that made the booking
11	company	character	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
12	customer_type	character	Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of contract associated to it; Group – when the booking is associated to a group; Transient – when the booking is not part of a group or contract, and is not associated to other transient booking; Transient-party – when the booking is transient, but is associated to at least other transient booking
13	reservation_status	character	Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out – customer has checked in but already departed; No-Show – customer did not check-in and did inform the hotel of the reason why

Target Variable

The target variable of the above dataset is is_canceled. We have to predict whether a booking is going to be cancelled or not.



In the above dataset, 72.47% of the bookings have not been cancelled and 27.53% of the bookings have been cancelled. **We observe that there is there is presence of moderate amount of class imbalance**.

Methodology to be Followed

CRISP-DM which stands for Cross Industry Standard Process for Data Mining is a methodology created to help shape data mining projects. It describes the different phases/tasks involved in the project and provides an overview of data mining life cycle.

1. Business Understanding - It focuses on determining the business requirements/objectives and understanding what outcome to achieve. Also determine the business units being affected. Convert this business problem into a data mining problem and carve out an initial plan.

- Determine the business objectives: Understand what is needed to be accomplished for the customer.
- Assess situation: Determine resources availability, project requirements, assess risks and contingencies, and conduct a cost-benefit analysis.
- Determine data mining goals: Convert business problem to a data mining problem and recognize the data mining problem type such as classification, regression or clustering, etc.
- Produce a project plan: Devise a step-to-step plan for executing the project.

2. Data understanding - This phase starts with collecting the data and then examining the data for its surface properties like data format, number of records, etc. The next step is to better understand the data by understanding each attribute and perform basic statistics on them. Understand the relationship between different attributes. Determine the quality of data by checking the missing values, outliers, duplicates, etc.

- Collect initial data: Acquire the data and load it into the analysis tool to be used.
- Describe data: Examine the data and document its surface properties like data format, number of records, or field identities. Understand the meaning of each attribute and attribute value in business terms. For each attribute, compute basic statistics so as to get a higher-level understanding.
- Explore data: Find insights from the data. Query it, visualize it, and identify relationships among the data.
- Verify data quality: Identify special values, missing attributes and null data. Determine how clean/dirty is the data.

3. Data preparation - This stage, which is often referred to as data wrangling, has the objective to develop the final data set for EDA and modelling. Covers all activities to construct the final dataset from the initial raw data. Some of the tasks include table, record and attribute selection as well as transformation and cleaning of data for modelling tools.

- Select data: Determine which attributes/features will be used and document reasons for inclusion/exclusion.
- Clean data: Correct, impute and remove the improper data.
- Extract data: Derive new attributes from the existing ones
- Integrate data: Create features by combining data from multiple sources.

- Format data: Re-format data as necessary. For example, convert string values to numeric values so as to perform mathematical operations.

4. Modelling - In this stage we build and assess different models built using various techniques from the training dataset.

- Select modelling technique: Determine the algorithms to be used to model the data based on the business requirement.
- Generate test design: In order to build and test the model, we need to divide the dataset into training and testing data set. In this step we divide the data into train and test data set.
- Build model: Based on the modelling technique selected, build the model on the input data set.
- Assess model: Compare the results of different models based on confusion matrix. The outcome of this step frequently leads to model tuning iterations until the best model is found.

5. Evaluation - Evaluate the models and review the steps executed to construct the model to be certain it properly achieves the business objectives.

- Evaluate results: Understand the data mining results and check how impactful they are in achieving the data mining goal. Select appropriate model based on confusion matrix.
- Review process: Review the work accomplished and make sure that nothing was overlooked and all steps were properly executed. Summarize the findings and correct anything if needed.
- Determine next steps: Based on the previous three tasks, determine whether to proceed to deployment, iterate further, or initiate new projects.

DATA PRE-PROCESSING

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre-processing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

The data consists of 119390 rows and 32 columns. Out of these we have 13 categorical columns and the rest as numerical.

Datatype Verification

We first check the data types of each of the columns of the data.

Attribute	Datatype
hotel	object
is_canceled	int64
lead_time	int64
arrival_date_year	int64
arrival_date_month	object
arrival_date_week_number	int64
arrival_date_day_of_month	int64
stays_in_weekend_nights	int64
stays_in_week_nights	int64
adults	int64
children	float64
babies	int64
meal	object
country	object
market_segment	object
distribution_channel	object
is_repeated_guest	int64
previous_cancellations	int64
previous_bookings_not_canceled	int64
reserved_room_type	object
assigned_room_type	object
booking_changes	int64
deposit_type	object
agent	float64
company	float64
days_in_waiting_list	int64
customer_type	object
adr	float64
required_car_parking_spaces	int64
total_of_special_requests	int64
reservation_status	object
reservation_status_date	object

From here we observe that is_repeated_guest, is_canceled are categorical data but are stated as numerical. We need to convert them to categorical data. Also, reservation_status_date is converted to date time format.

Attribute	Data Type
hotel	object

is_canceled	object
lead_time	int64
arrival_date_year	int64
arrival_date_month	object
arrival_date_week_number	int64
arrival_date_day_of_month	int64
stays_in_weekend_nights	int64
stays_in_week_nights	int64
adults	int64
children	float64
babies	int64
meal	object
country	object
market_segment	object
distribution_channel	object
is_repeated_guest	object
previous_cancellations	int64
previous_bookings_not_canceled	int64
reserved_room_type	object
assigned_room_type	object
booking_changes	int64
deposit_type	object
agent	object
company	float64
days_in_waiting_list	int64
customer_type	object
adr	float64
required_car_parking_spaces	int64
total_of_special_requests	int64
reservation_status	object
reservation_status_date	datetime64[ns]

Missing Value Treatment

The next step of data pre-processing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Attribute	Null Value Percentage
children	0.00335
country	0.408744
agent	13.686238
company	94.306893

According to the research paper(domain knowledge) null values are not missing values rather they are not applicable or unknown Values.
Research Paper:- https://www.sciencedirect.com/science/article/pii/S2352340918315191

- For Children the null means parents do not have any child. Hence replace it with 0.
- For country we replace null values by unknown.
- For agent, the booking was not done by any agent, hence replace null values by 0 which represents that the booking wasn't done by the agent.
- As we see that company column has largest number of null values. Hence, we drop this column.

Duplicate and Noisy Value Removal

Checking and removal of duplicate rows is important because presence of duplicates can lead us to make incorrect conclusions by leading us to believe that some observations are more common than they really are.

In our dataset, we have 32001 duplicate rows. Hence, we drop these duplicate rows. After removal of these duplicate rows, we have 87389 rows left.

Along with duplicate rows, we have 166 rows wherein the booking is made, but the number of adults, children and babies is zero. These rows indicate some anomaly in the booking entry. Hence, we remove these rows.

From the 5-point summary of the data, we observe that minimum value in ‘adr’ column is -6.38 which is not possible. Hence, we remove the columns containing negative ‘adr’ values.

After performing all these steps, we finally have data which has 87222 rows and 31 columns.

Check for Outliers

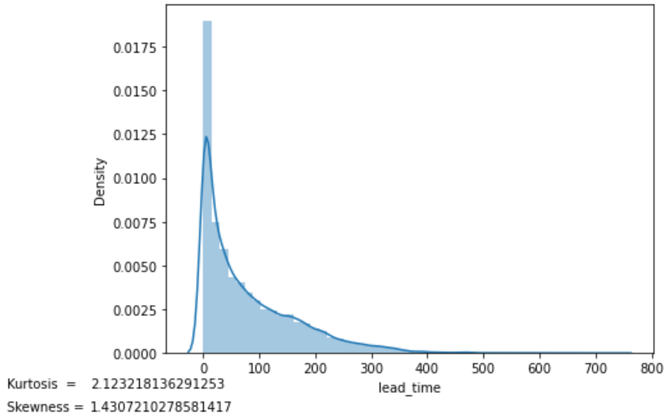
Data has outliers present in each of the numerical columns. For making the base model, we do not perform any outlier treatment and retain all the rows present in the data.

EXPLORATORY DATA ANALYSIS

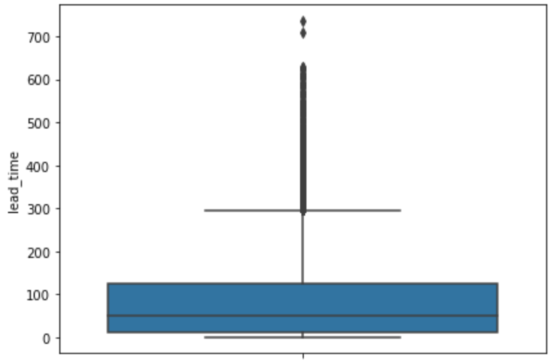
Univariate Analysis

For Numerical Variables:- We plot the distribution curve and box plot to study the variation of the numerical data.

1. lead_time

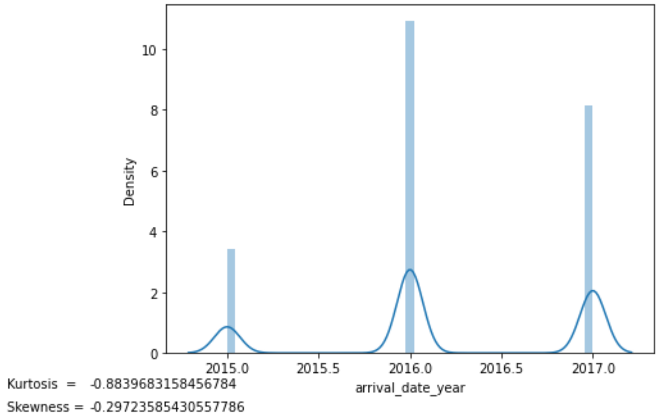


```
Kurtosis =   2.123218136291253
Skewness = 1.4307210278581417
```

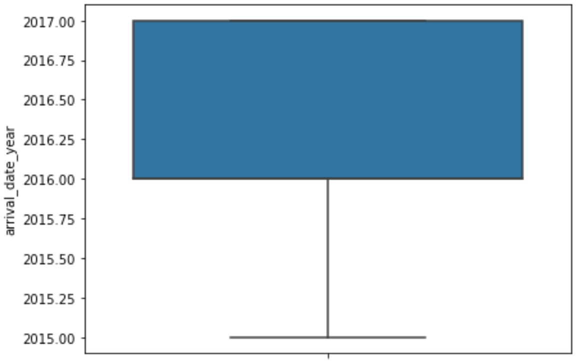


- Lead time is right skewed.
- It is leptokurtic, and has wide tail
- IQR of lead time lies from 0-300. Outliers are present.

2. arrival_date_year

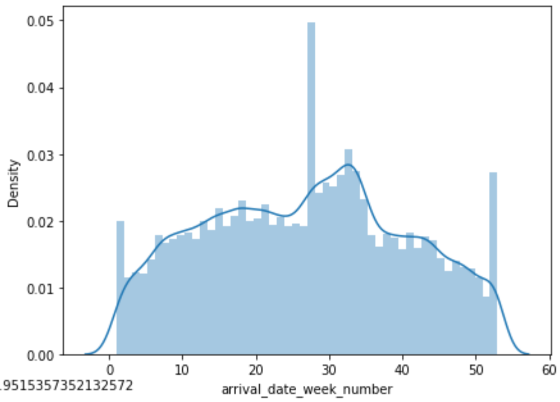


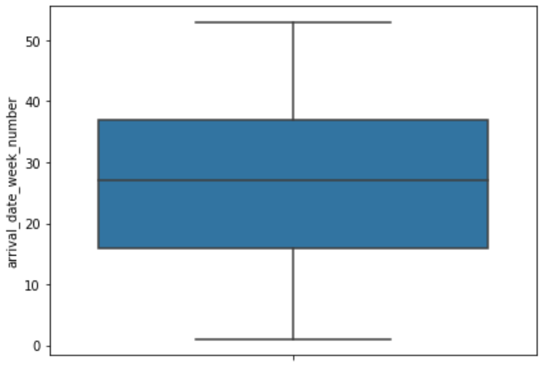
```
Kurtosis =   -0.8839683158456784
Skewness = -0.29723585430557786
```



- It has highest frequency for 2016 followed by 2017 and 2015.

3. arrival_date_week_number

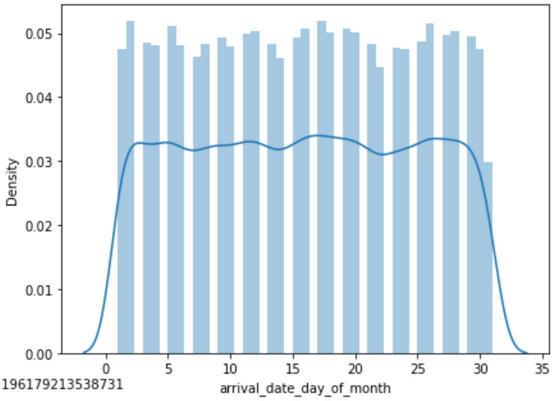


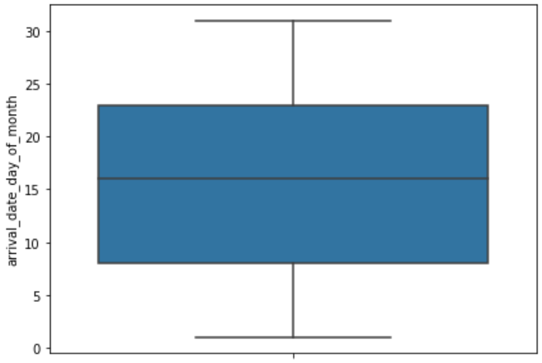


```
Kurtosis =   -0.9515357352132572
Skewness =  0.022401674404889965
```

- It varies from 1 to 52.
- It is a platykurtic curve.
- Highest frequency can be observed around the 27-28 followed by 52 and 1. This means that more bookings are made in the summer weeks and during the Christmas and New Years Eve.

4. arrival_date_day_of_month

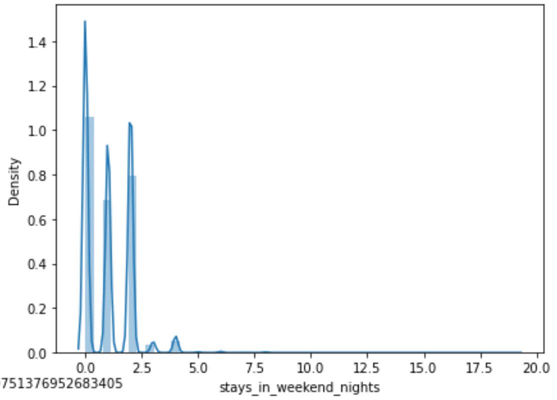


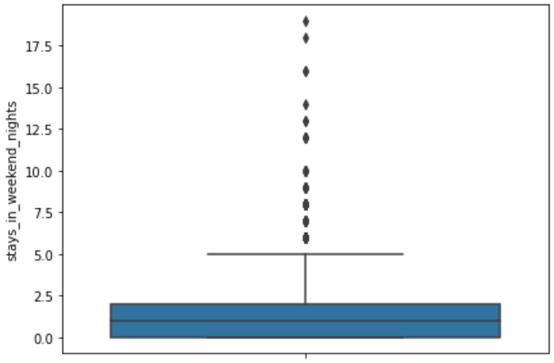


```
Kurtosis =   -1.196179213538731
Skewness =  0.00026174674141421955
```

- It is a platykurtic curve.
- It takes values from 1 to 31.

5. stays_in_weekend_nights

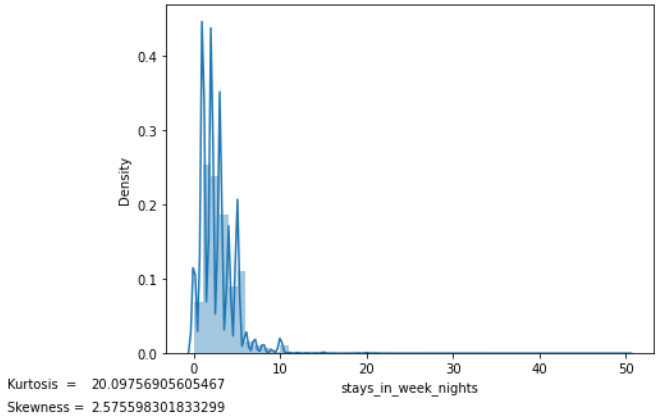




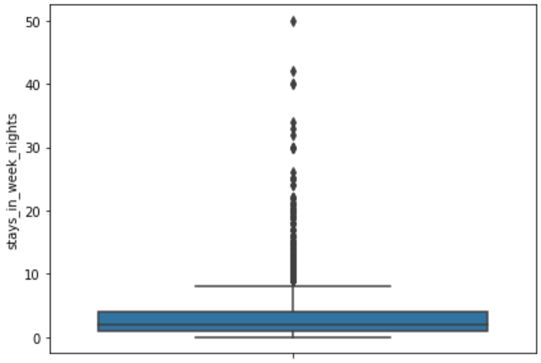
```
Kurtosis =   7.0751376952683405
Skewness = 1.3402646740157271
```

- It has a right skewed distribution.
- It is a leptokurtic curve and has wide tail.
- There are outliers present.
- IQR lies from 0 to 5.

6. stays_in_week_nights

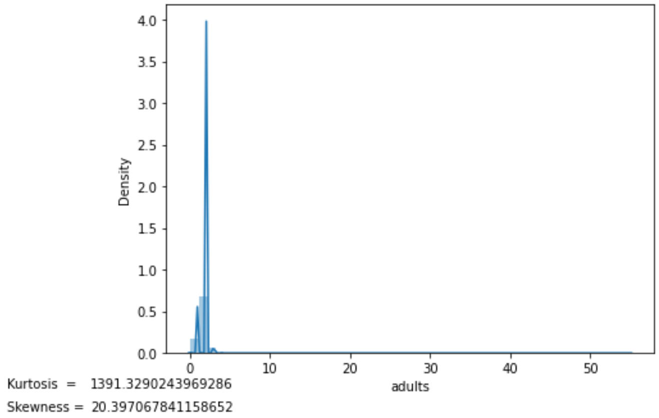


```
Kurtosis =   20.09756905605467
Skewness = 2.575598301833299
```

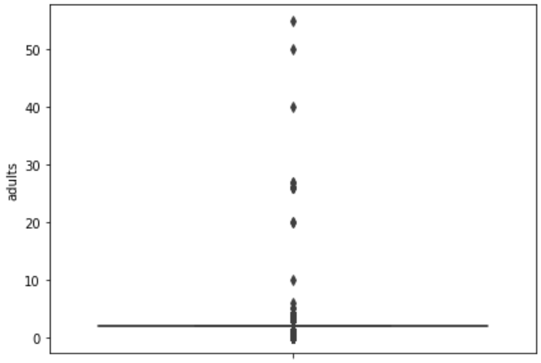


- It has right skewed distribution.
- It is a leptokurtic curve and has a wide tail.
- There is presence of outliers and IQR lies between 0-9.

7. adults

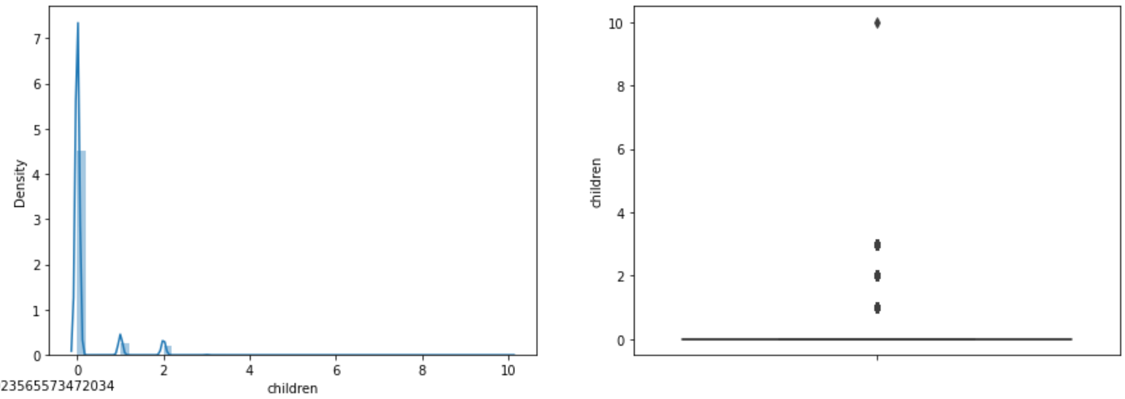


```
Kurtosis =   1391.3290243969286
Skewness = 20.397067841158652
```



- It is highly right skewed.
- It also has a high value of kurtosis and is a leptokurtic curve.
- Hence IQR is very narrow.
- It has a very high frequency at value 3.

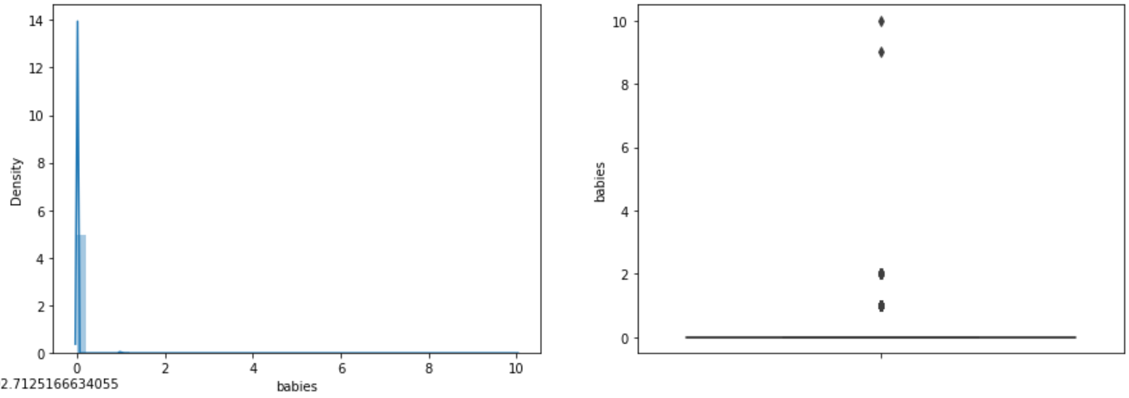
8. children



```
Kurtosis =   12.923565573472034
Skewness =  3.4594161347898664
```

- It has a very high frequency at value 0.
- It has a right skewed distribution.
- It is also leptokurtic and has wide tails.
- IQR is not present i.e. it has a very narrow IQR

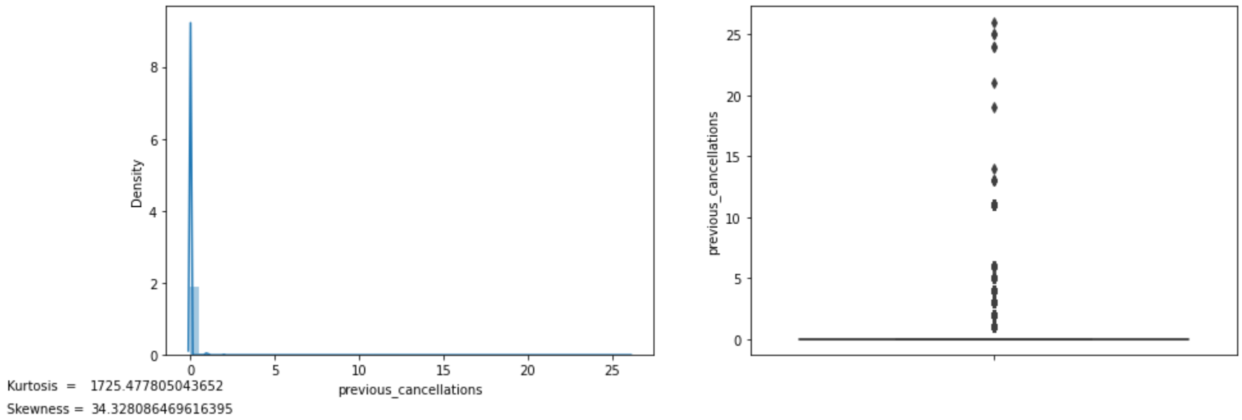
9. babies



```
Kurtosis =   1202.7125166634055
Skewness =  21.127463051912724
```

- It has a very high frequency at value 0.
- It has a right skewed distribution.
- It is also leptokurtic and has wide tails.
- IQR is not present i.e. it has a very narrow IQR

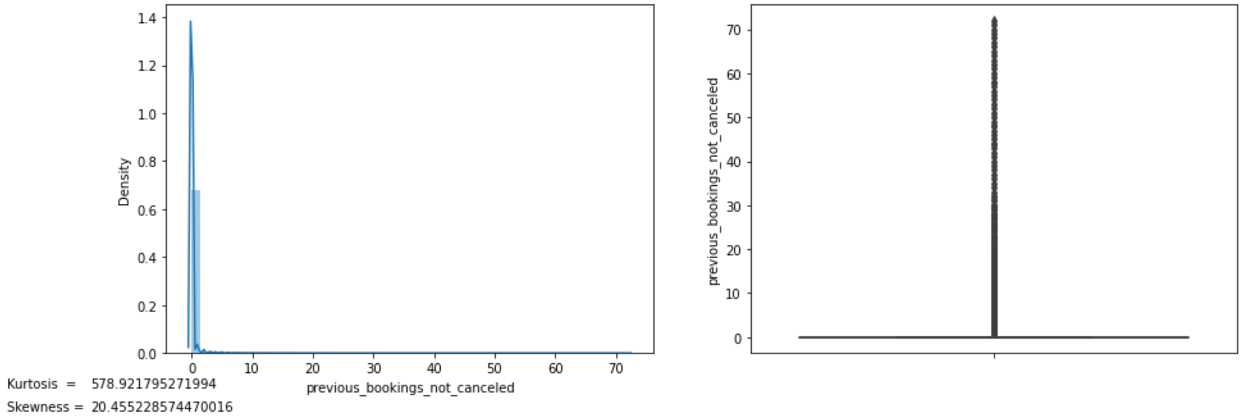
10. previous_cancellations



```
Kurtosis =   1725.477805043652
Skewness =  34.328086469616395
```

- It has a very high frequency at value 0.
- It has a highly right skewed distribution.
- It also has a high value of kurtosis and is a leptokurtic curve.
- IQR is not present i.e. it has a very narrow IQR
- It has a wide tail.

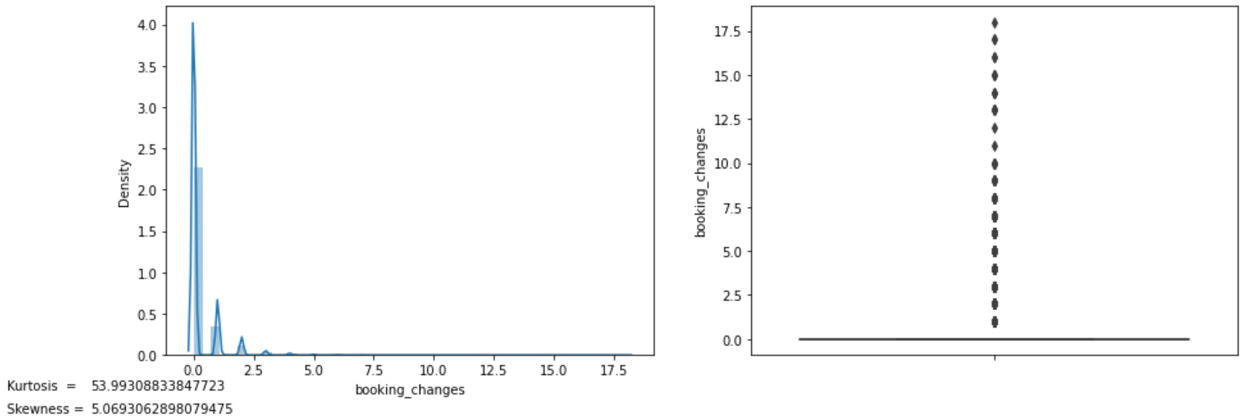
11. previous_bookings_not_canceled



```
Kurtosis =   578.921795271994
Skewness =  20.455228574470016
```

- It has a very high frequency at value 0.
- It has a highly right skewed distribution.
- It also has a high value of kurtosis and is a leptokurtic curve.
- IQR is not present i.e. it has a very narrow IQR
- It has a wide tail.

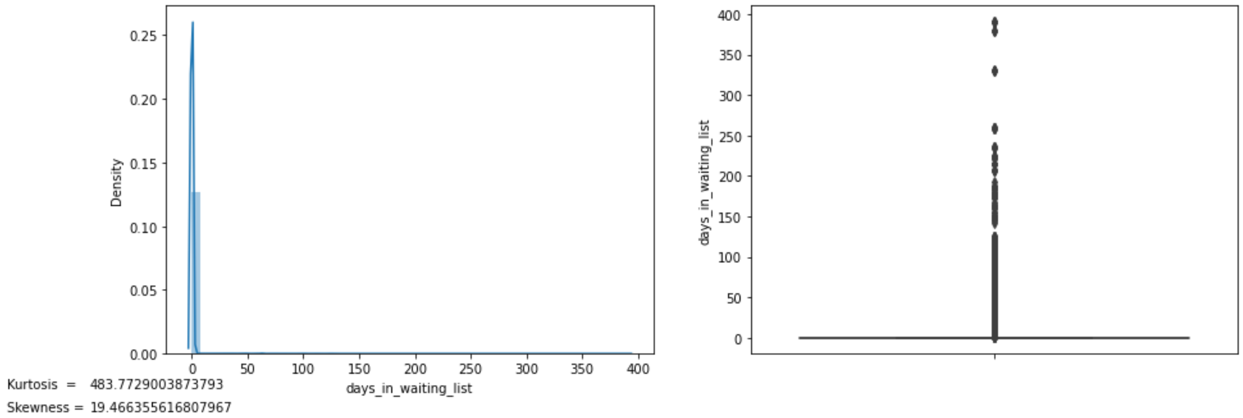
12. booking_changes



```
Kurtosis =   53.99308833847723
Skewness = 5.0693062898079475
```

- It has a very high frequency at value 0.
- It has a right skewed distribution.
- It is also leptokurtic and has wide tails.
- IQR is not present i.e. it has a very narrow IQR

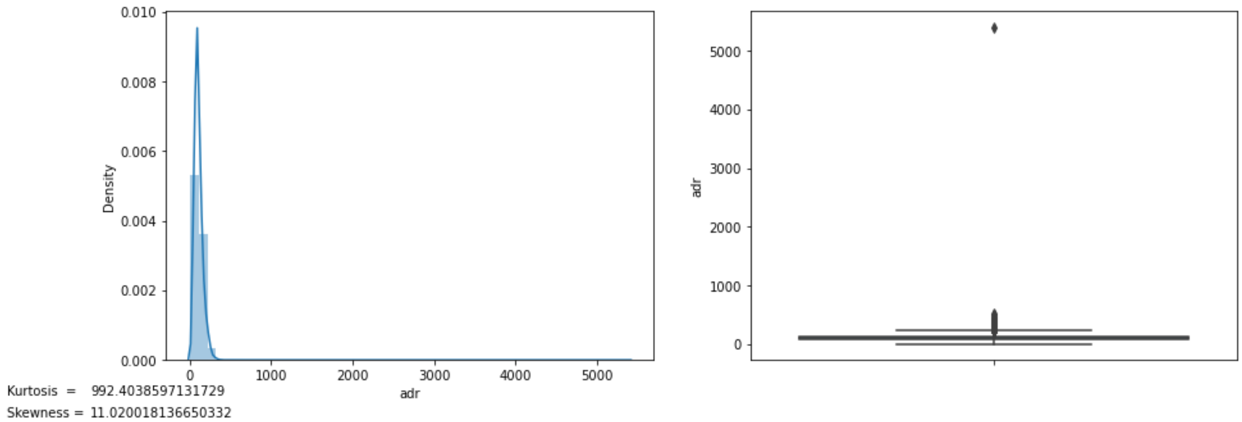
13. days_in_waiting_list



```
Kurtosis =   483.7729003873793
Skewness = 19.466355616807967
```

- It has a very high frequency at value 0.
- It has a highly right skewed distribution.
- It also has a high value of kurtosis and is a leptokurtic curve.
- IQR is not present i.e. it has a very narrow IQR
- It has a wide tail.

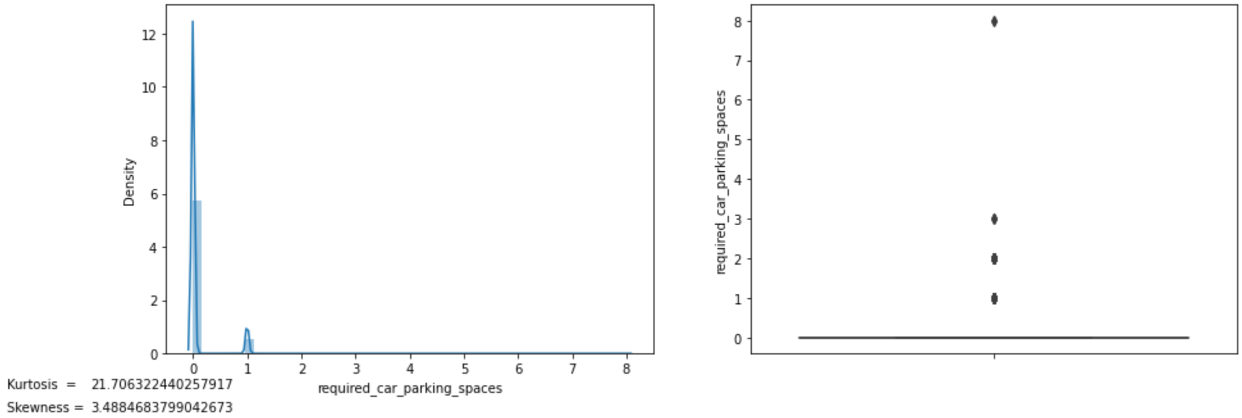
14. adr



```
Kurtosis =   992.4038597131729
Skewness = 11.020018136650332
```

- It has a highly right skewed distribution.
- It also has a high value of kurtosis and is a leptokurtic curve.
- IQR lies between 0 to 200.
- It has an outlier where adr is 5400.

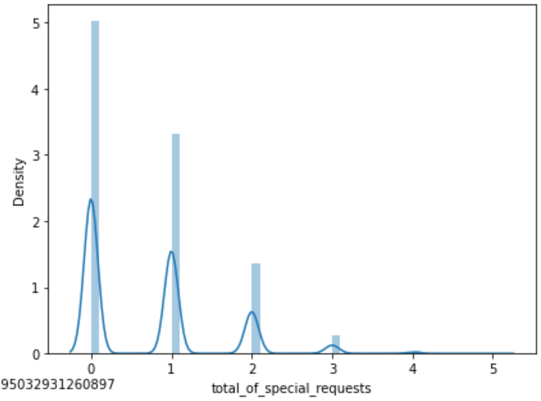
15. required_car_parking_spaces



```
Kurtosis =   21.706322440257917
Skewness = 3.4884683799042673
```

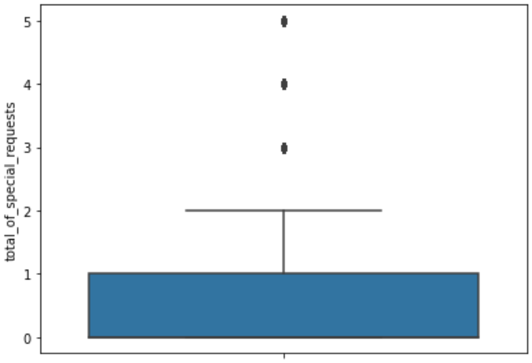
- It has a very high frequency at value 0.
- It has a right skewed distribution.
- It also has a high value of kurtosis and is a leptokurtic curve.
- IQR is not present i.e. it has a very narrow IQR
- It has a wide tail.

16. total_of_special_requests



Kurtosis = 0.8195032931260897

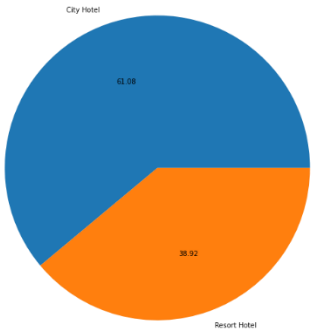
Skewness = 1.082363488181165



- It has a very high frequency at value 0.
- It has a right skewed distribution.
- It is a leptokurtic curve.
- Outliers are present.

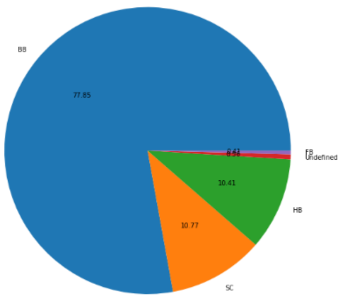
For Categorical Variables – We plot a combination of bar graph and pie chart to understand the distribution of categorical data in the dataset.

1. hotel



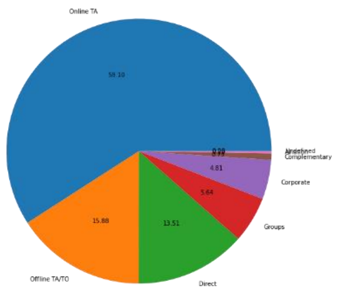
- 61% of the total bookings are done in City Hotel.
- 39% of the total bookings are done in Resort Hotel

2. meal



- 78% of the bookings made prefer a Bed and Breakfast stay.
- 10.5% of the bookings made prefer half board.
- 10.7% of the bookings prefer no meal.

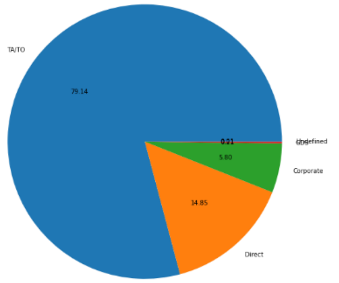
3. market_segment



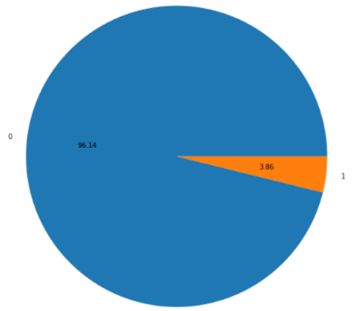
- 59% bookings are belong to Online Travel Agent market segment.
- 75% of the bookings belong to Travel Agent or Tour Operator (offline and online).
- 13.5% bookings are direct

4. distribution_channel

- 79% bookings are made through Travel Agents or Tour Operators.
- 15% bookings are done directly.



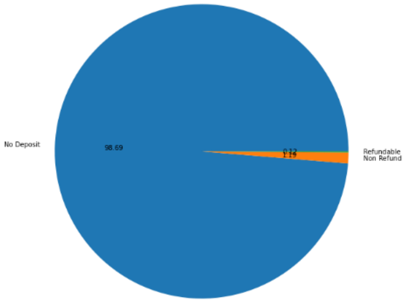
5. repeated_guest



- 96% of the bookings made was not from a repeated guest.

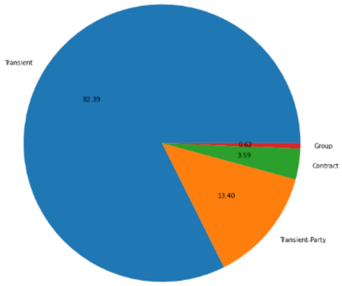
6. deposit_type

- 98.5% bookings are No Deposit bookings.



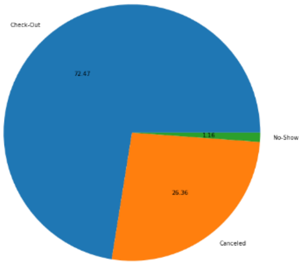
7. customer_type

- 82% of bookings come from Transient customers.

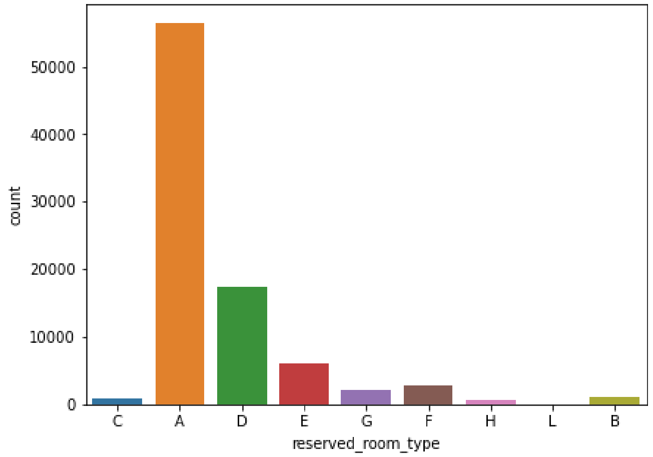


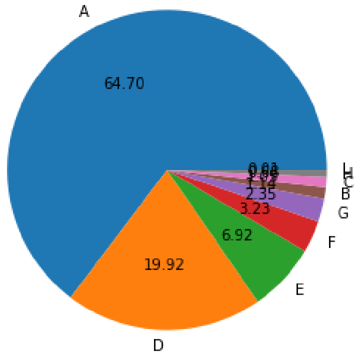
8. reservation_status

- 72.5% bookings have been checked out.
- 1.16% bookings are a no show.
- 26% bookings have been cancelled.



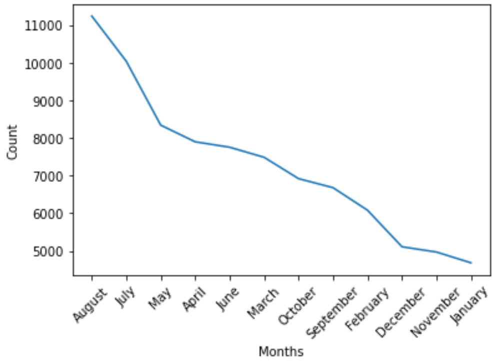
9. reserved_room_type





- 64% Bookings are done for room category A.
- 20% bookings are done for room category D.

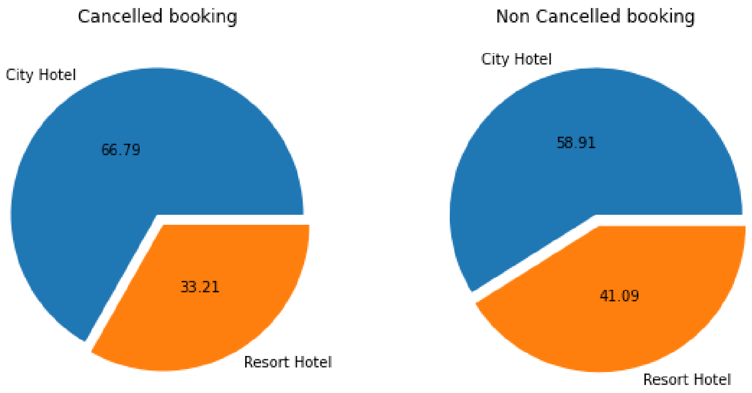
10. arrival_date_month



- Trend line to show the variation in bookings for different months.
- August has the highest number of bookings.
- Overall bookings increase in the months of summer.

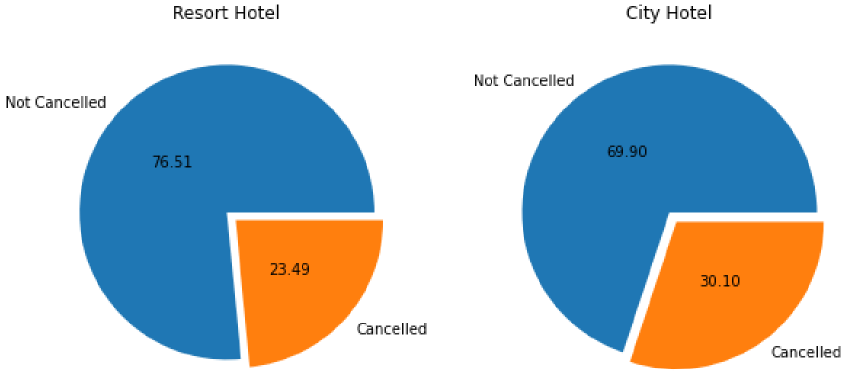
Bi-Variate Analysis

1. Hotel type vs is_canceled



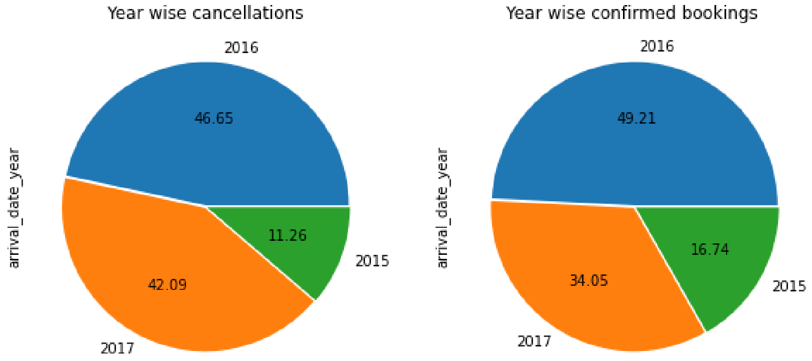
- Out of the total bookings cancelled, City hotel has contributed towards 67% cancellations.
- Out of the total confirmed bookings, City hotel has 59% of conformed bookings.

2. Percentage cancellations for each hotel



- 23.5% of the bookings made in Resort Hotel are cancelled.
- 30% of the bookings made in City Hotel are cancelled.
- City hotel has a higher cancellation percentage out of the total bookings made.

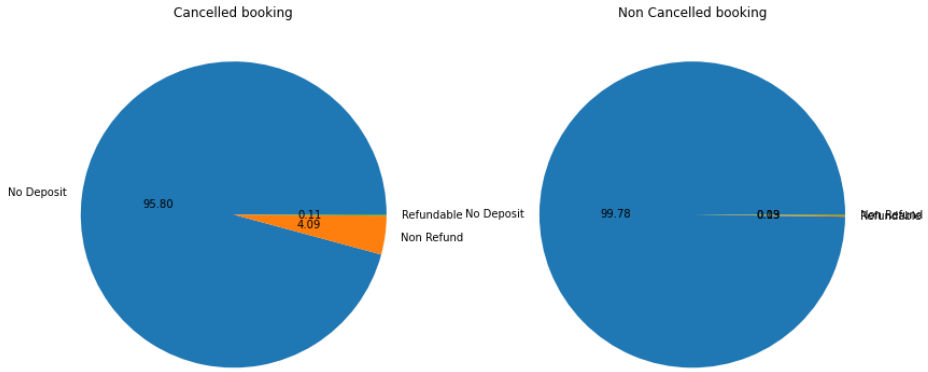
3. Year wise cancellations and confirmed bookings



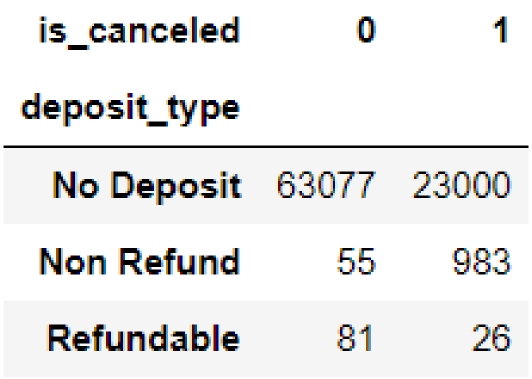
- Out of the total cancellations made for the 3 years, 46.5% cancellations are made in 2016.

- 2016 has the highest percentage confirmed and cancelled bookings.

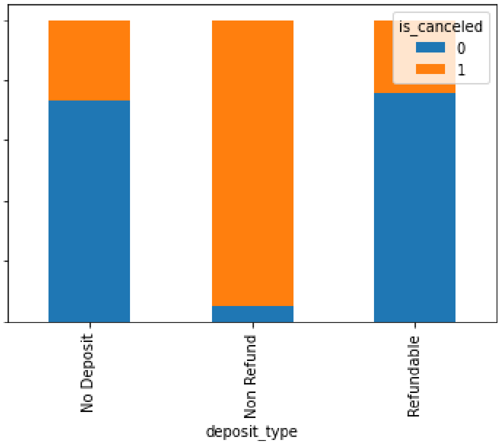
4. Deposit type vs cancellations



- Out of the bookings cancelled, 96% cancellations occurred for no deposit bookings.
- Out of the bookings confirmed, 99.78% confirmations occurred for no deposit bookings.

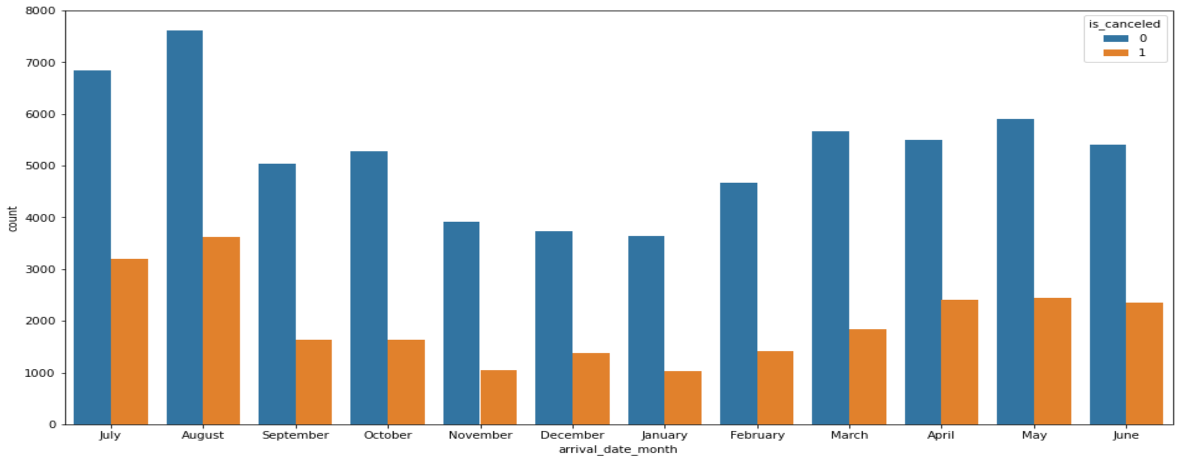


is_canceled	0	1
deposit_type		
No Deposit	63077	23000
Non Refund	55	983
Refundable	81	26

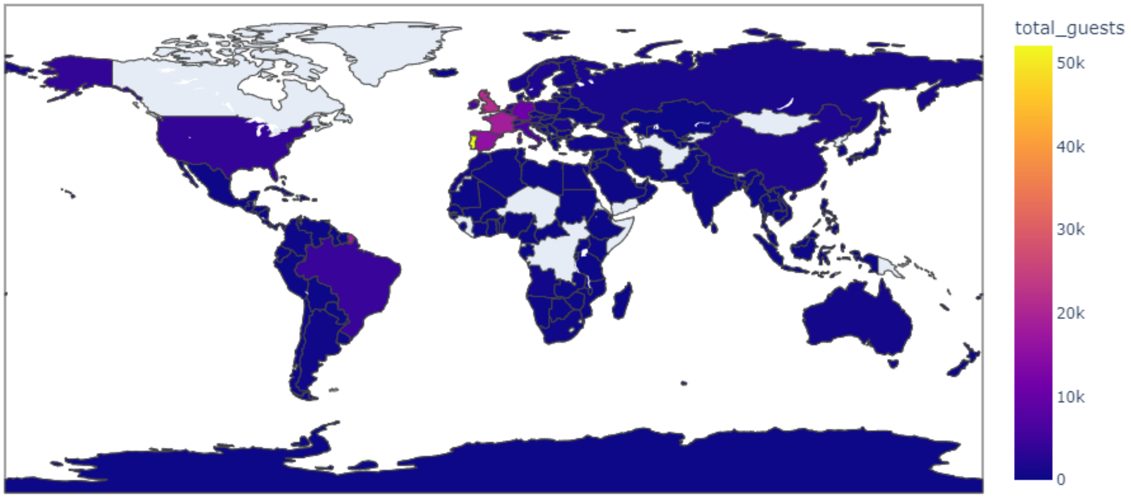


- Maximum ratio of cancellations occur for non refund bookings.
- Whereas the cancellation ratio for no deposit and refundable is almost the same.

5. Month vs cancellation
   - Maximum cancellations occurred for the month of August followed by July.
   - Also the number of confirmed bookings are highest for August followed by July.

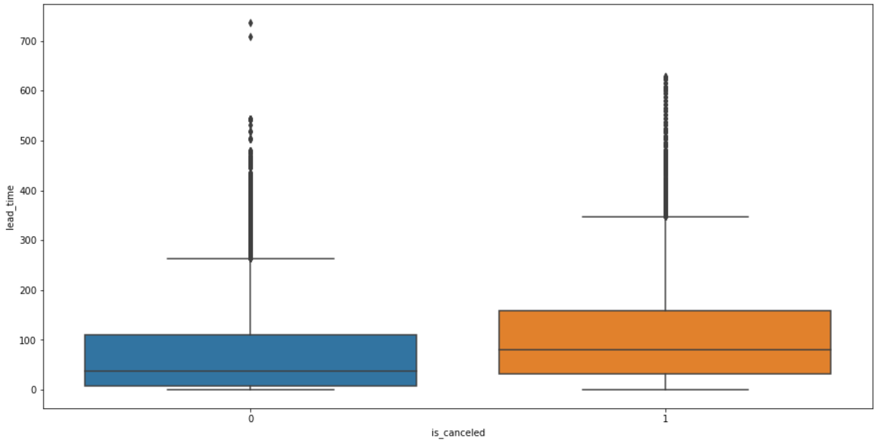


6. No of guests for each country



- More than 50,0000 visitors are from Portugal.
- Maximum visitors are from European region

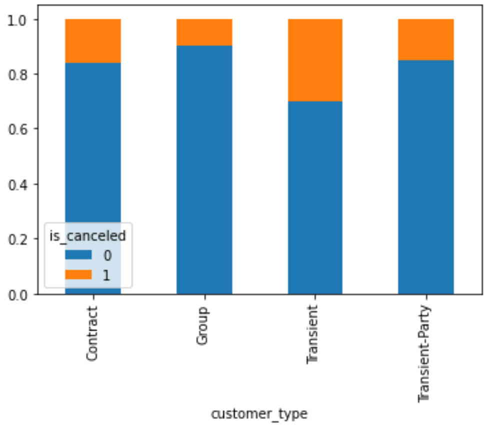
7. Lead time vs cancellation

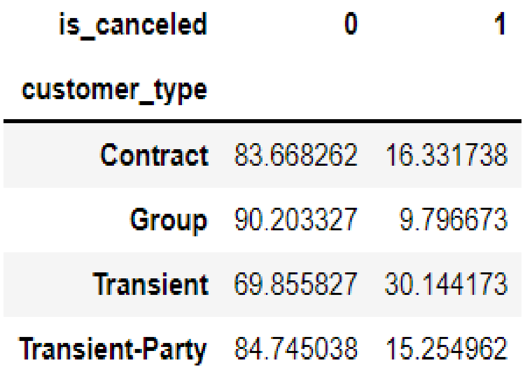


- We observe that lead time for cancelled bookings is greater than that of confirmed bookings.

- Hence, we can say that people tend to make the bookings in advance and then tend to cancel them due to various reasons.

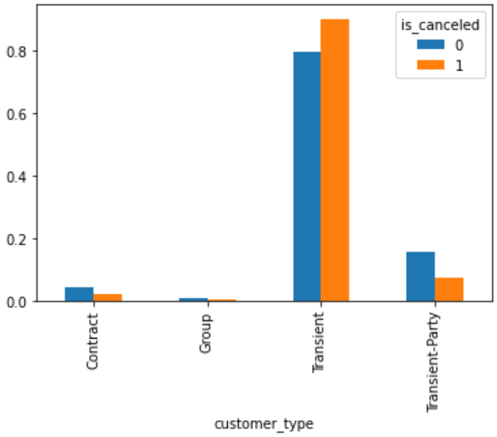
8. Customer type vs cancellations

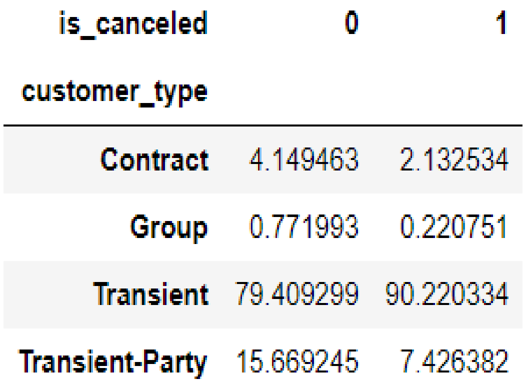




is_canceled	0	1
customer_type		
Contract	83.668262	16.331738
Group	90.203327	9.796673
Transient	69.855827	30.144173
Transient-Party	84.745038	15.254962

- For different types of customers, the ratio of cancellation is highest for transient customers.
- The ratio of cancellation for group bookings is the lowest.



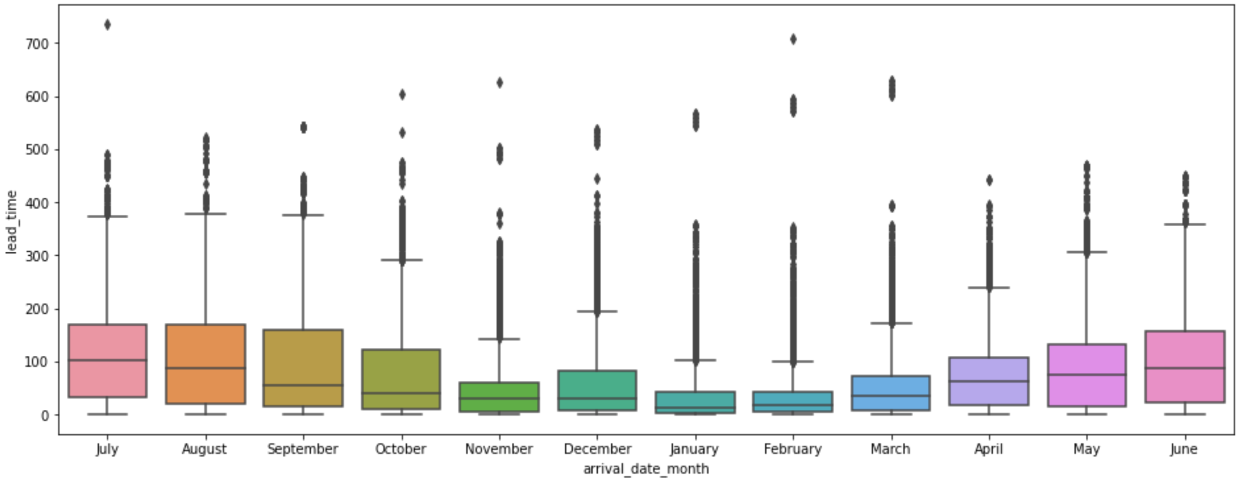


is_canceled	0	1
customer_type		
Contract	4.149463	2.132534
Group	0.771993	0.220751
Transient	79.409299	90.220334
Transient-Party	15.669245	7.426382

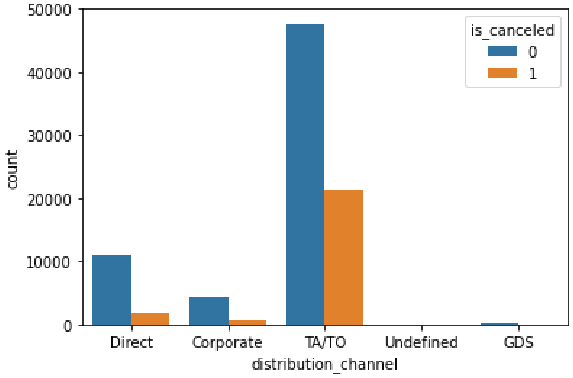
- Out of the total cancellations, maximum cancellations are made by transient customers.
- Maximum number of confirmed bookings also belong to transient customers.

9. Lead time vs month
   - Lead time is higher for May, June, July and August.
   - These months also have higher number of bookings compared to the other months.

- Hence for months having higher bookings, lead time is also more.

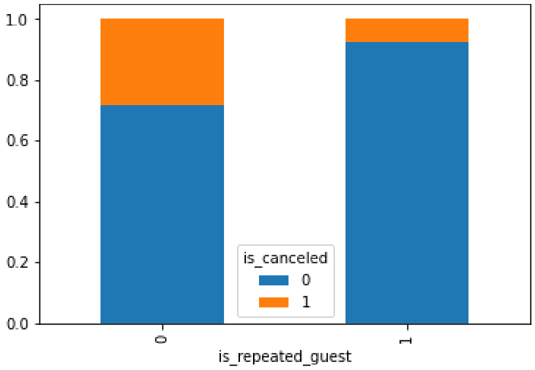


10. Distribution channel vs cancellation



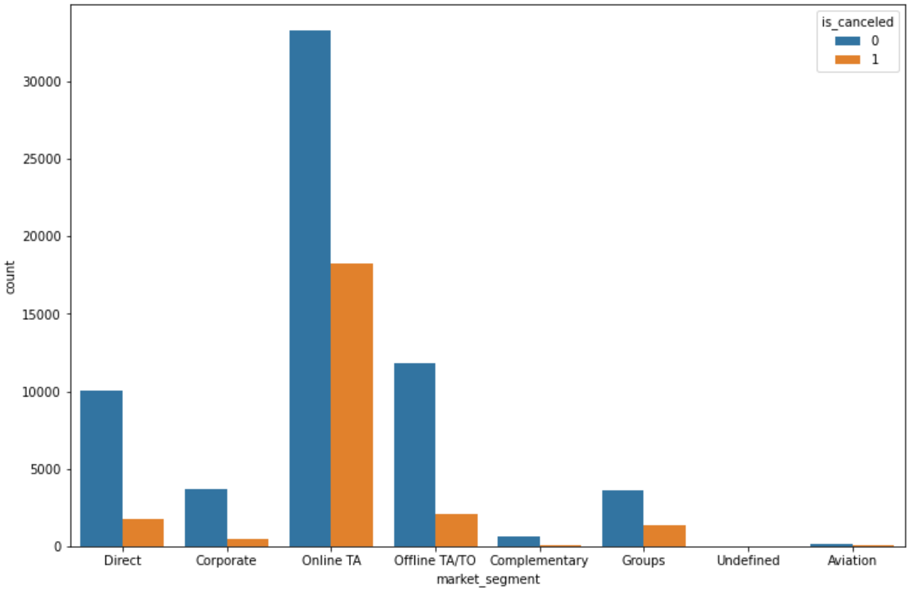
- Maximum cancellations occur for customers making booking through Travel Agents/Tour Operators.

11. Repeated guest vs cancellation



- The ratio of cancellation is less for repeated guest compared to non-repeated guests.
- Hence, we can say that repeated guests have a lesser tendency to cancel the bookings. This might be due to the fact that they would have enjoyed the experience at the hotel.

12. Market segment vs cancellation



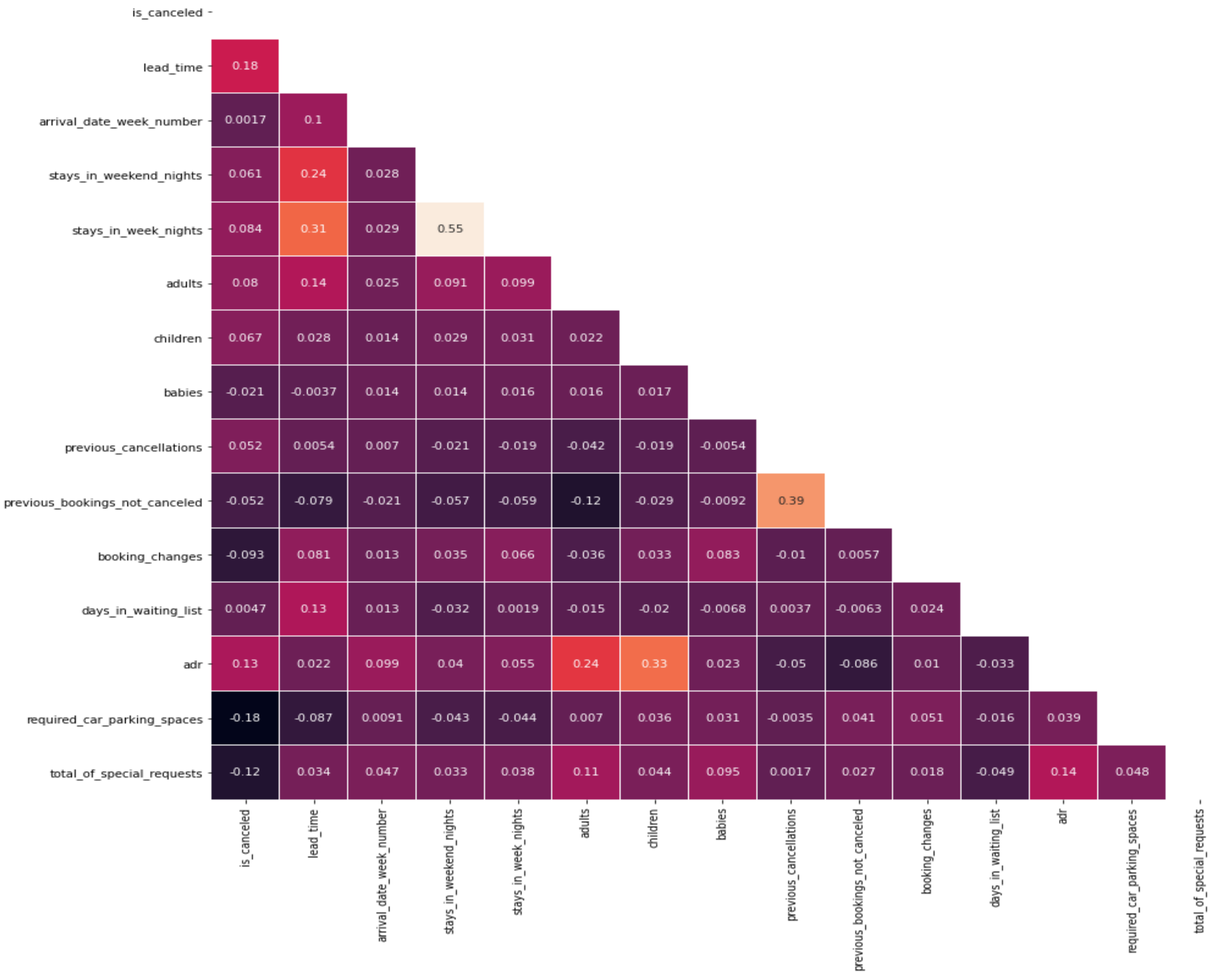
- Maximum cancellations are made by Online TA customers, followed by Offline TA/TO.

Correlation Matrix

Heat-Map - Pearson Correlation Matrix

(Assumption : For the Pearson correlation, both variables should be normally distributed. Other assumptions include linearity and homoscedasticity)

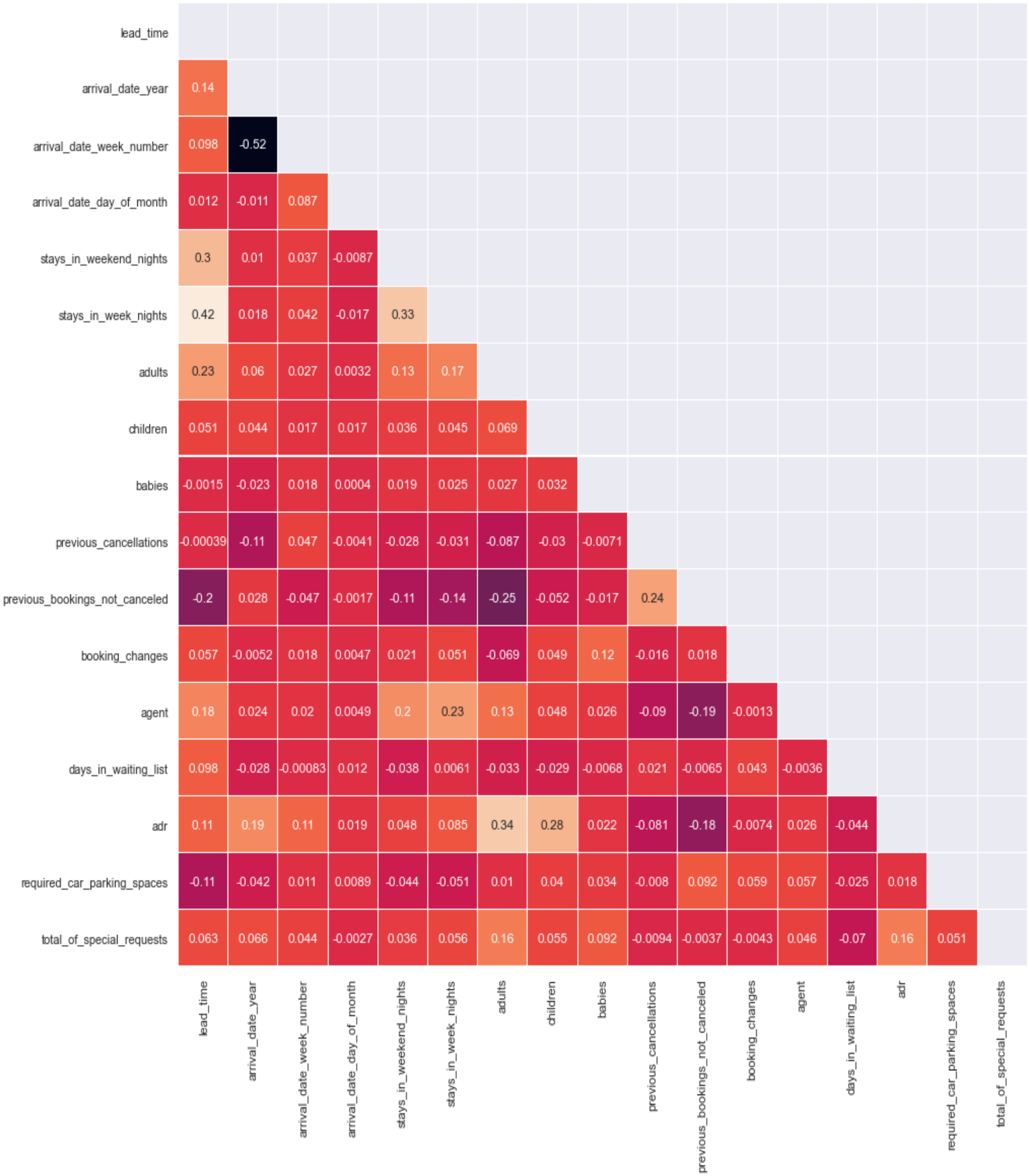
It gives a measure of how much two numeric variables are linearly correlated. It tries to obtain a best fit line between two numeric variables and how close the points are to a fitted line.



1. From the graph we can see that high level of collinearity is not present in our data.
2. But we can see some moderate level of correlations between:
   - Stays in week nights and stays in weekend nights in positive direction
   - Previous booking cancellations and Previous booking not cancelled
   - Number of children and average daily rate(adr)
   - Lead time and stays in week nights

Heat-Map - Spearman Correlation Matrix

Spearman rank correlation is a non-parametric test that is used to measure the degree of association between two variables. It does not assume normal distribution of data and looks for a monotonic relationship between variables. Two variables are monotonic correlated if any greater value of the one variable will result in a greater value of the other variable



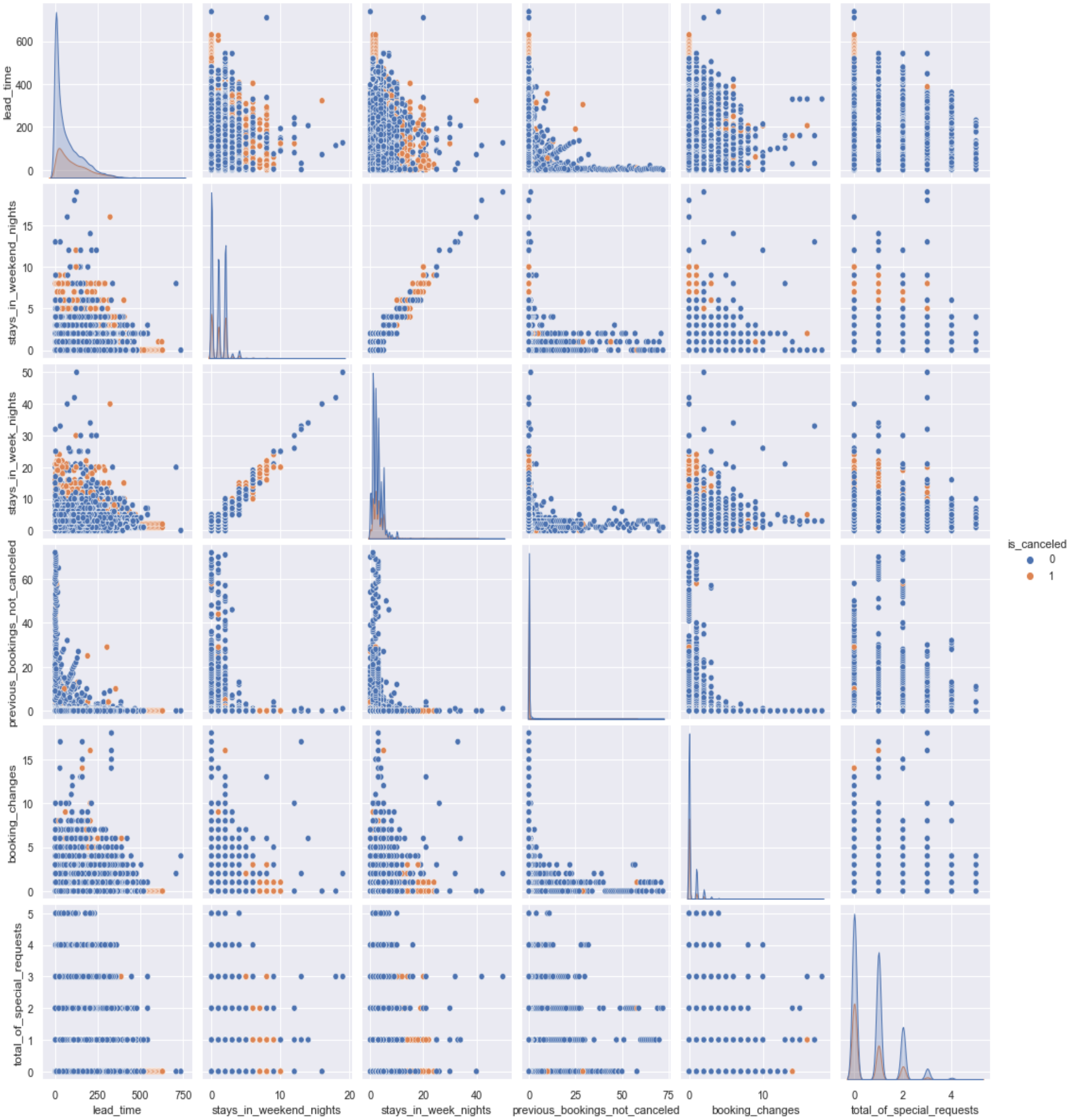
1. Apart from the Pearson correlation matrix:

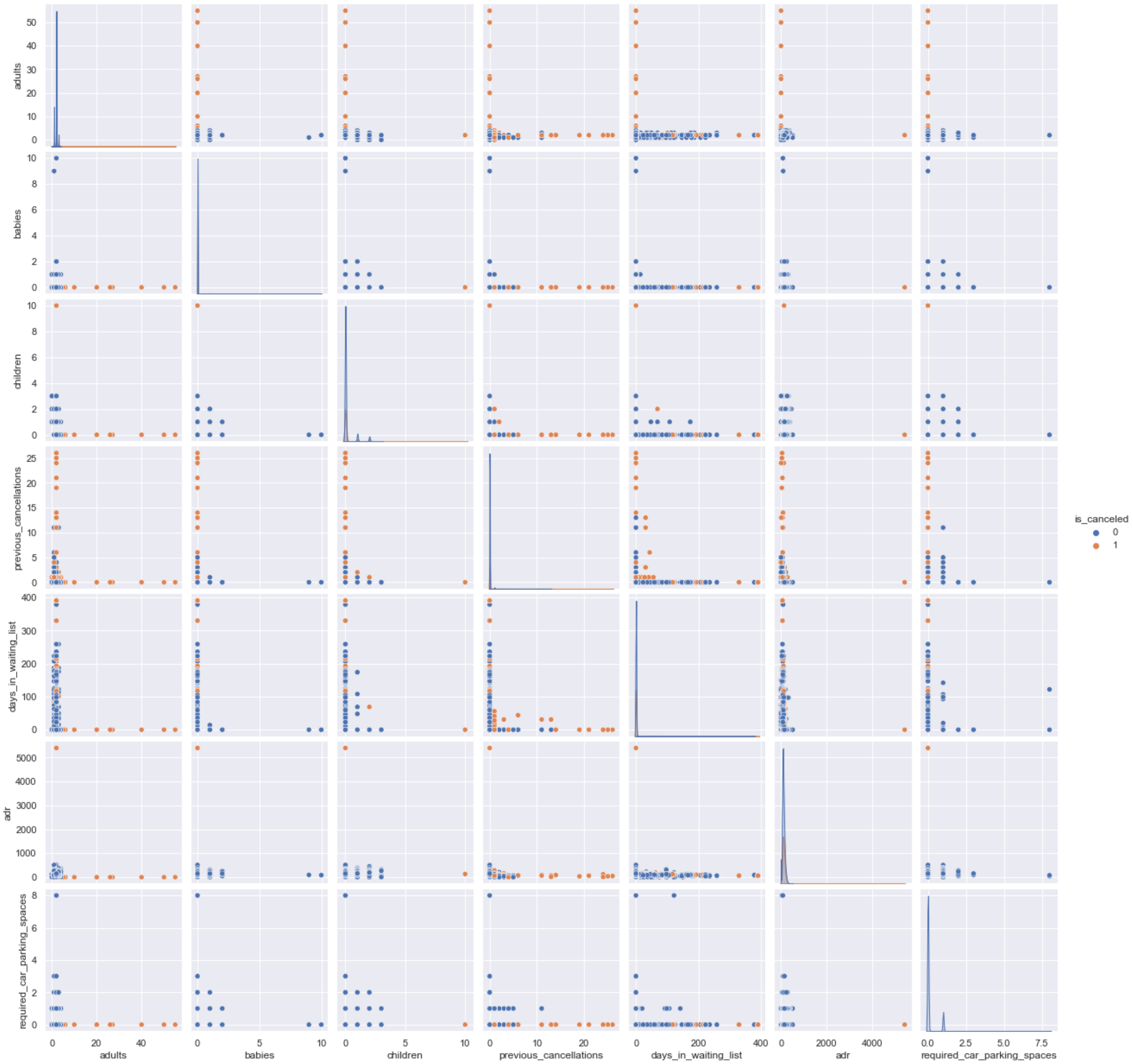
    - Arrival_date_week_number and arrival_date_year is showing moderate correlation in negative direction.

- The features stays_in_week_nights and lead_time have moderate positive relation.
- Average daily date(adr) is showing moderate correlation with number of adults in positive direction.

Multi-Variate Analysis

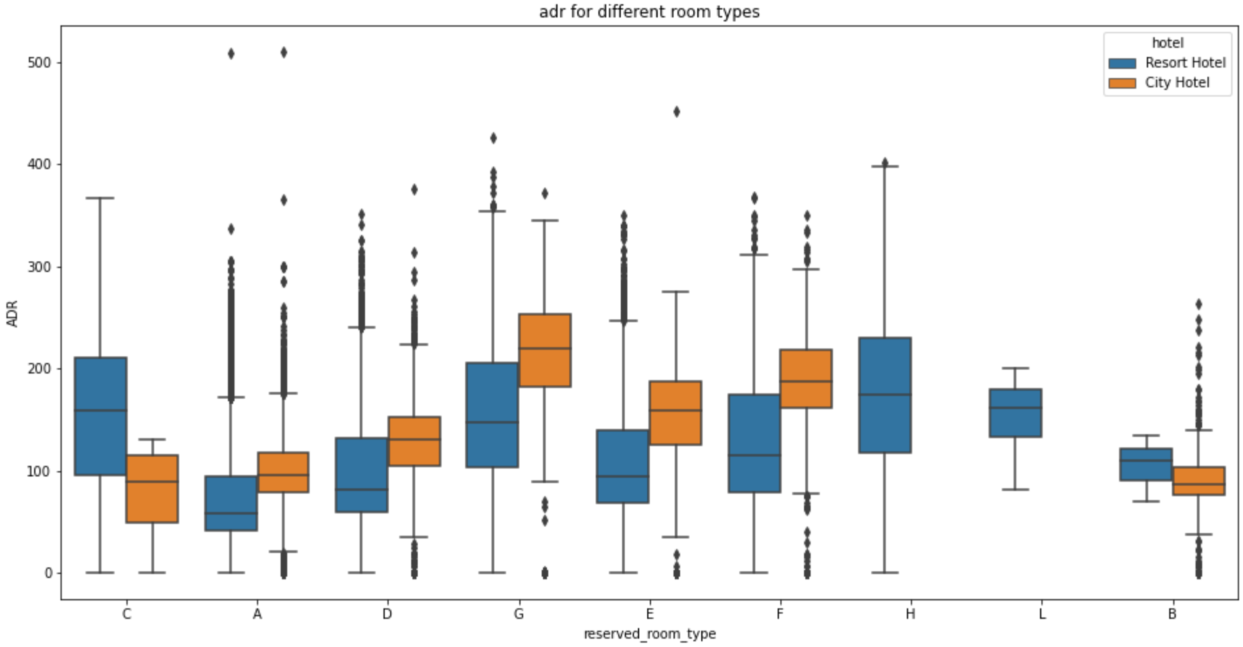
1. Pair-Plot





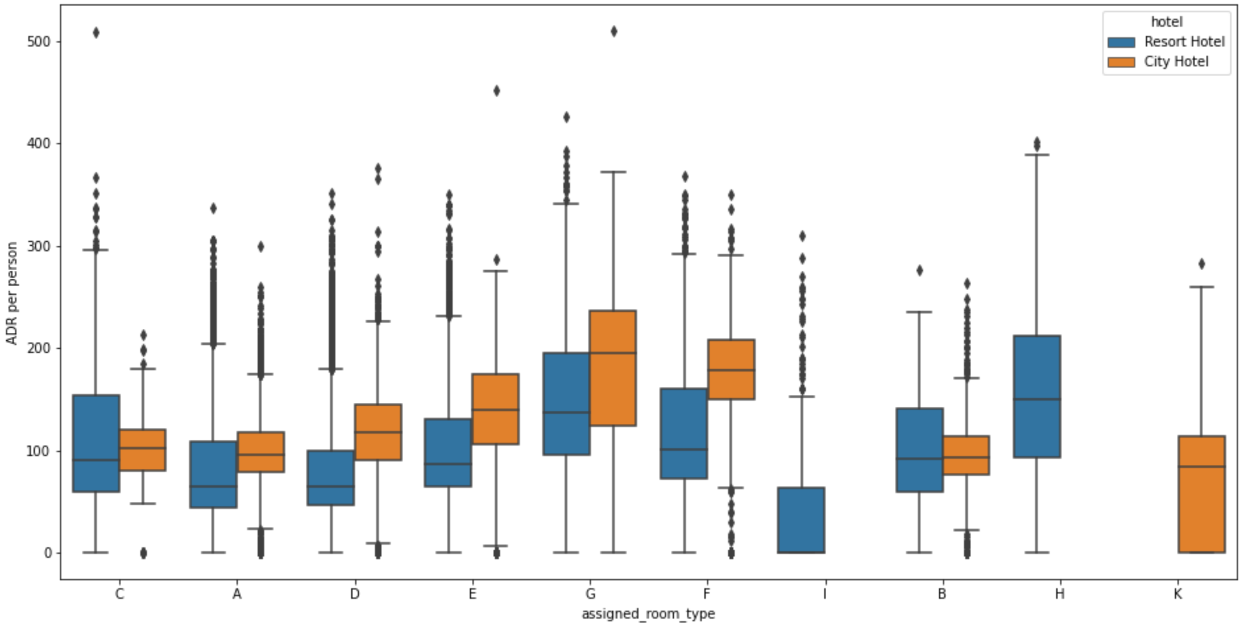
- From the pair plot we can see that none of the variables are normally distributed. The distribution of the features is right skewed.
- We can see linear relationship only between days_in_week_nights and days_in_weekend_nights.
- Except these features all the others have no relationship.
- High amount of overlapping is there with respect to variable is_canceled in the distribution of all variables.

2. Adr for reserved room type with hue



- For City Hotel, ADR for type G room is the highest.
- City Hotel does not have reservation for type H and L.
- For City Hotel ADR of room A,B and C is similar.
- For Resort Hotel, ADR for room C and H is highest.
- Except for room type C and B, the ADR for City hotel is greater than ADR of Resort hotel.
- No reservation is done for room I and K in City and Resort hotel.

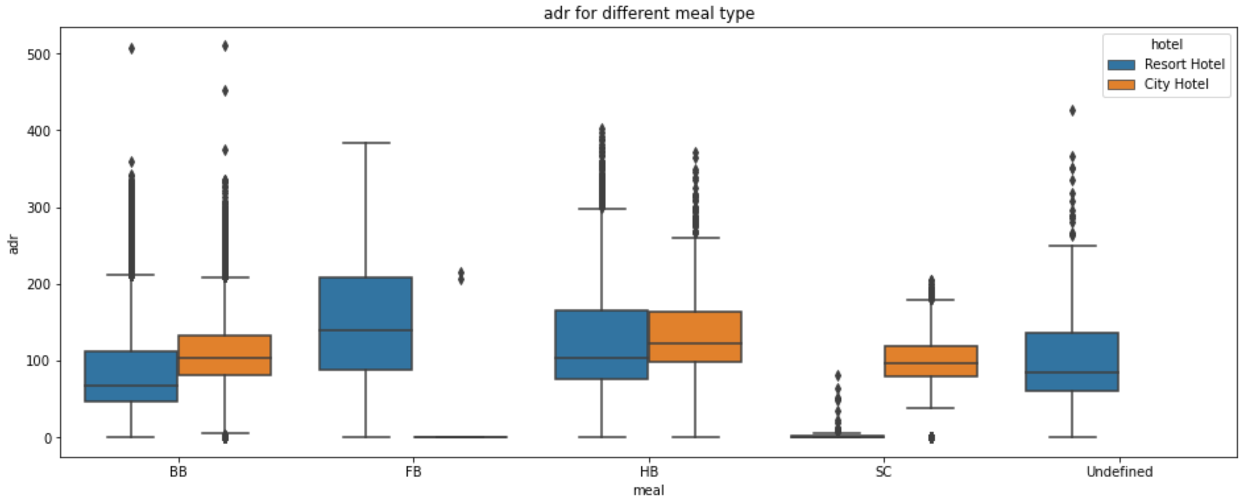
3. Adr for assigned room type with hue



- For City Hotel, ADR for type G room is the highest.

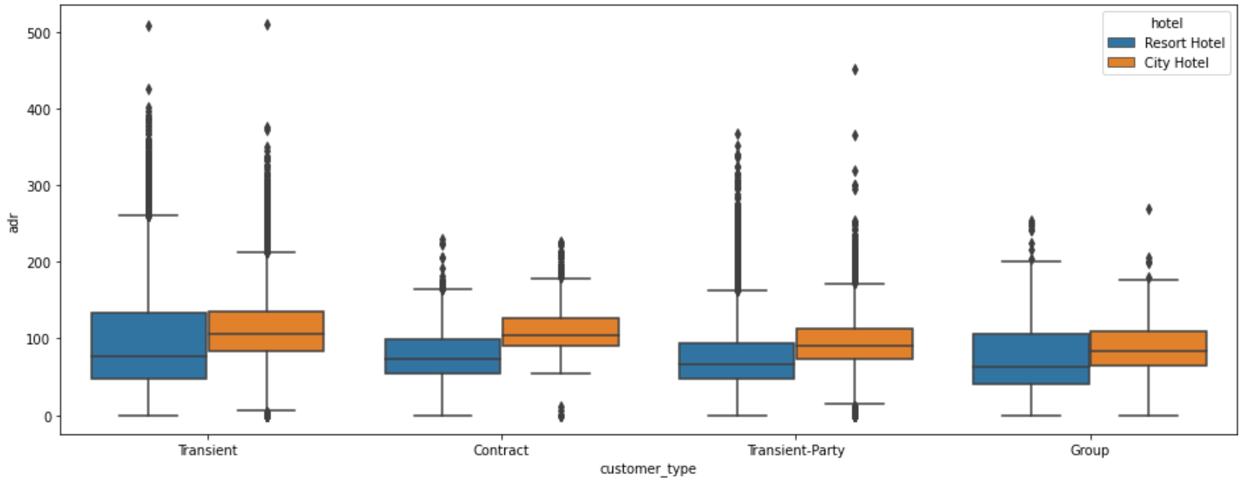
- No one stays in rooms I, H and L of City Hotel.
- No one stays in room K and L of the Resort hotel.
- For Resort Hotel, ADR for room G and H is highest.
- Except for room type C and B, the ADR for City hotel is greater than ADR of Resort hotel.

4. Adr for different meal types



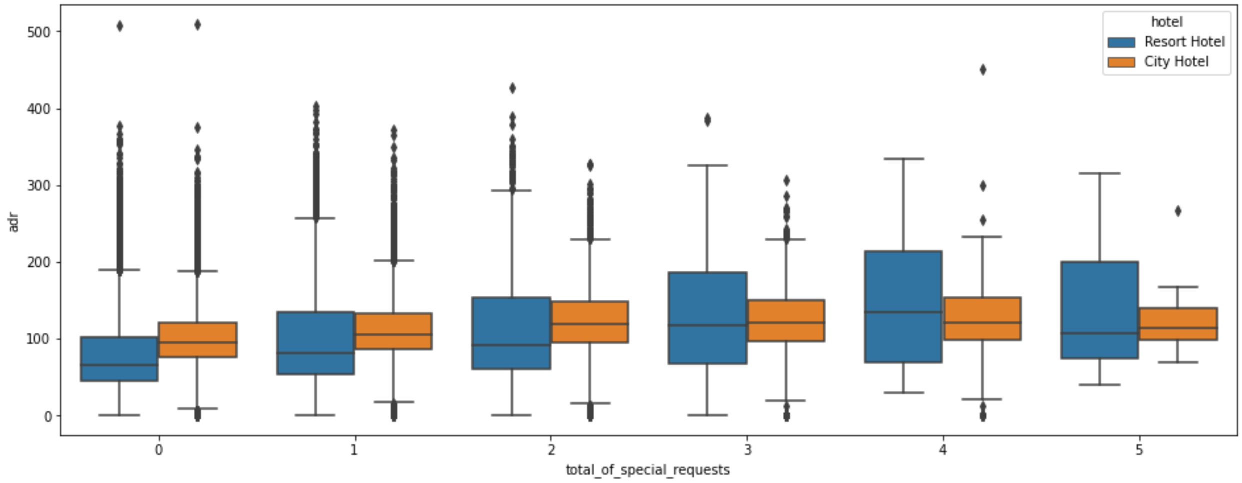
- ADR for Full Board is highest.
- For resort hotel, ADR of Full board > Half Board > Bed and Breakfast
- For City hotel ADR of Half Board > Bed and Breakfast > SC(no meal package)

5. Adr for different customers



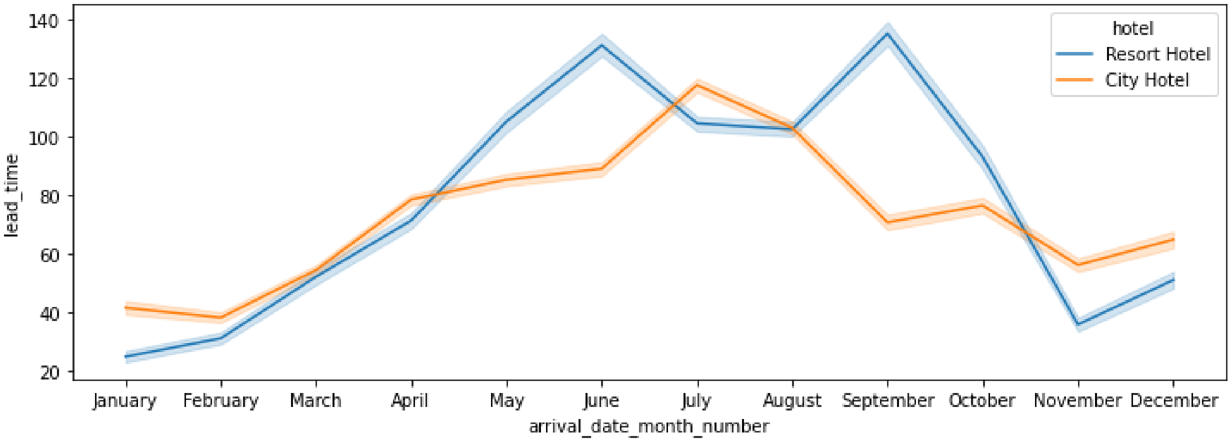
- 1. ADR of Transient customers of City hotel is highest.
- 2. ADR of all customers is more for City hotel than Resort hotel.
- 3. ADR for groups is the lowest. This is might due to the fact that the hotel might be offering discounts for Group bookings.

6. Variation of adr with special request



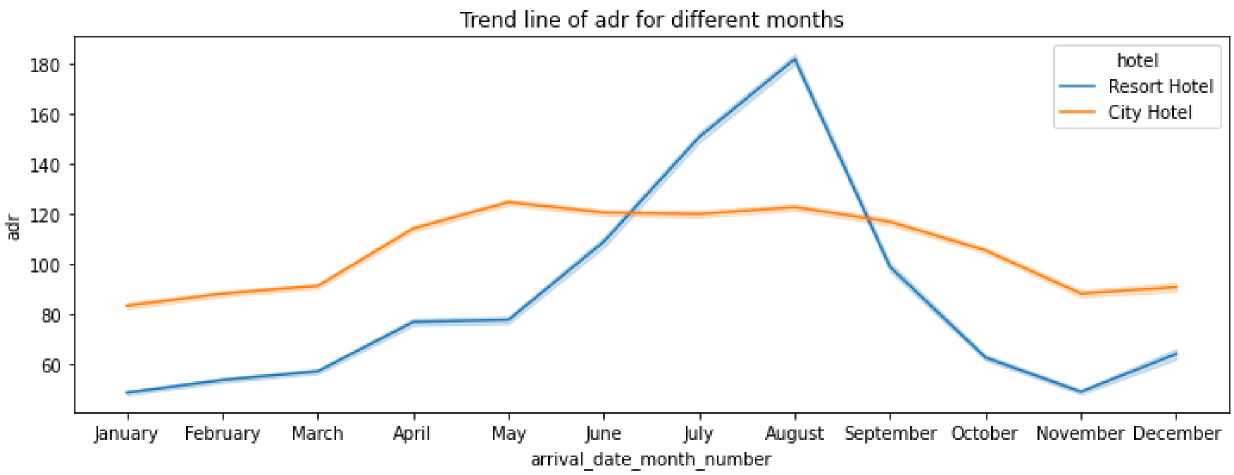
- ADR increases for Resort and City hotel as the number of special request increases except for 5.

7. Lead time for different months



- 1. Lead time for resort hotel is highest in June and September.
- 2. Lead time for city hotel is highest in July and August.
- 3. Lead time for both the hotel starts increasing during summer months and decreasing during winter months.

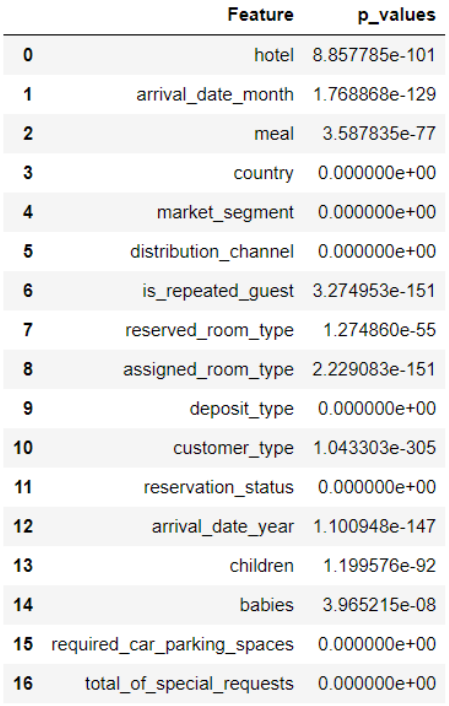
8. Adr for different months



- ADR of Resort hotel is highest in the months of July and August.
- ADR of Resort hotel rises sharply during the summer months whereas ADR of City hotel rises at a lesser rate.
- ADR of City hotel remains in the range of 80 to 120.

Statistical Tests

1. Categorical columns – For categorical columns we perform chi-square test to check for the significance of the categorical column with respect to is_canceled(target variable) column.



	Feature	p_values
0	hotel	8.857785e-101
1	arrival_date_month	1.768868e-129
2	meal	3.587835e-77
3	country	0.000000e+00
4	market_segment	0.000000e+00
5	distribution_channel	0.000000e+00
6	is_repeated_guest	3.274953e-151
7	reserved_room_type	1.274860e-55
8	assigned_room_type	2.229083e-151
9	deposit_type	0.000000e+00
10	customer_type	1.043303e-305
11	reservation_status	0.000000e+00
12	arrival_date_year	1.100948e-147
13	children	1.199576e-92
14	babies	3.965215e-08
15	required_car_parking_spaces	0.000000e+00
16	total_of_special_requests	0.000000e+00

Hypothesis of Chi-square test

H0 : Attributes are independent

H1 : Attributes are dependent

We observe that the p_values of all the columns are less than 0.05. Hence, we reject the null hypothesis. Therefore, we conclude that all the categorical features are significant.

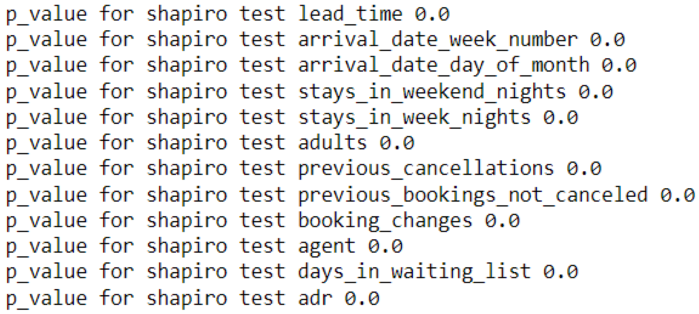
2. Shapiro test – We perform Shapiro test to identify whether the numerical data is normally distributed or not.

   Hypothesis of Shapiro test

   H0 : Attributes are normally distributed

   H1 : Attributes are not normally distributed

   p-values observed are-



```
p_value for shapiro test lead_time 0.0
p_value for shapiro test arrival_date_week_number 0.0
p_value for shapiro test arrival_date_day_of_month 0.0
p_value for shapiro test stays_in_weekend_nights 0.0
p_value for shapiro test stays_in_week_nights 0.0
p_value for shapiro test adults 0.0
p_value for shapiro test previous_cancellations 0.0
p_value for shapiro test previous_bookings_not_canceled 0.0
p_value for shapiro test booking_changes 0.0
p_value for shapiro test agent 0.0
p_value for shapiro test days_in_waiting_list 0.0
p_value for shapiro test adr 0.0
```

   We observe that p_value is less than 0.05. Hence, we reject the null hypothesis. Therefore, the numerical data is not normally distributed.

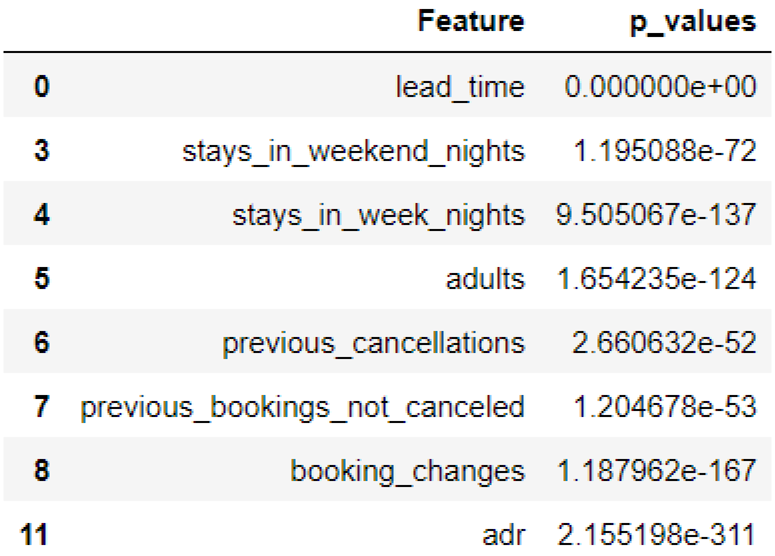
3. Numerical columns – We perform parametric and non-parametric tests for the numerical columns. Under parametric test we perform ANOVA and under non-parametric test we perform Mann Whitney U test.

   Hypothesis for numerical tests

   H0 : Two samples have the same mean (i.e insignificant)

   H1 : Two samples have different mean (i.e significant)

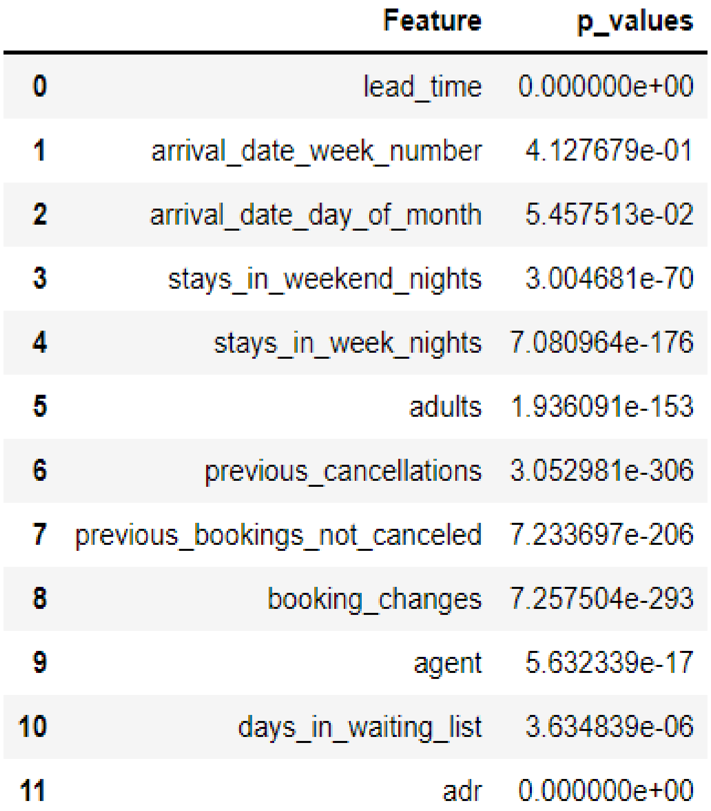
   - ANOVA test
     Significant variables identified from ANOVA test are -



	Feature	p_values
0	lead_time	0.000000e+00
3	stays_in_weekend_nights	1.195088e-72
4	stays_in_week_nights	9.505067e-137
5	adults	1.654235e-124
6	previous_cancellations	2.660632e-52
7	previous_bookings_not_canceled	1.204678e-53
8	booking_changes	1.187962e-167
11	adr	2.155198e-311

We observe that the p_values for some variables is greater than 0.05. Hence these variables are insignificant.

- Mann Whitney U test
  We observe that the p_values of all the variables is less than 0.05. Hence all the features are significant.



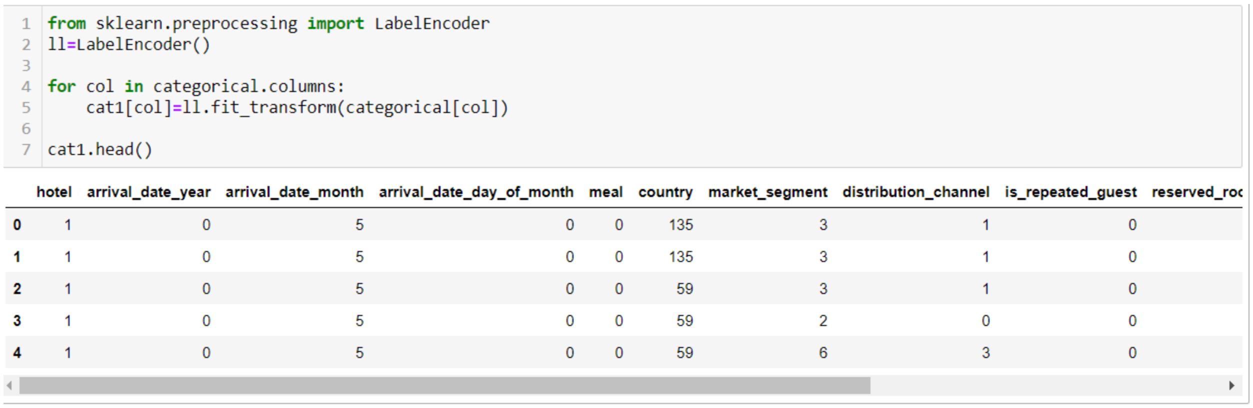
	Feature	p_values
0	lead_time	0.000000e+00
1	arrival_date_week_number	4.127679e-01
2	arrival_date_day_of_month	5.457513e-02
3	stays_in_weekend_nights	3.004681e-70
4	stays_in_week_nights	7.080964e-176
5	adults	1.936091e-153
6	previous_cancellations	3.052981e-306
7	previous_bookings_not_canceled	7.233697e-206
8	booking_changes	7.257504e-293
9	agent	5.632339e-17
10	days_in_waiting_list	3.634839e-06
11	adr	0.000000e+00

BASE MODEL

Logistic Regression Model

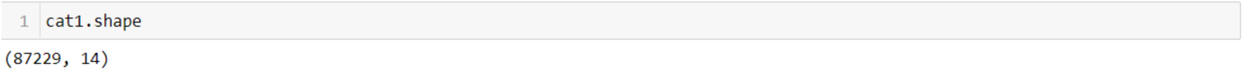
We have selected Logistic Regression as our base model. For this we have encoded all the categorical variables using Label Encoder and have kept the numerical columns as it is.

Encoding:



```
from sklearn.preprocessing import LabelEncoder
ll=LabelEncoder()

for col in categorical.columns:
    cat1[col]=ll.fit_transform(categorical[col])

cat1.head()
```

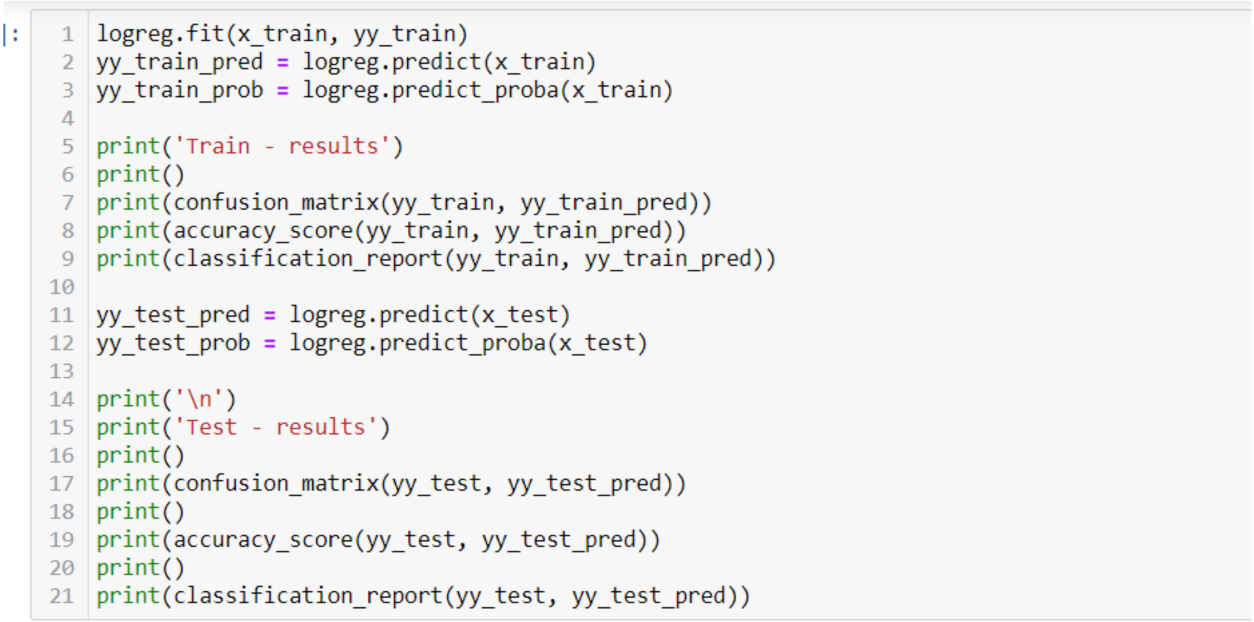
	hotel	arrival_date_year	arrival_date_month	arrival_date_day_of_month	meal	country	market_segment	distribution_channel	is_repeated_guest	reserved_roo
0	1	0	5	0	0	135	3	1	0	
1	1	0	5	0	0	135	3	1	0	
2	1	0	5	0	0	59	3	1	0	
3	1	0	5	0	0	59	2	0	0	
4	1	0	5	0	0	59	6	3	0	



```
cat1.shape
```

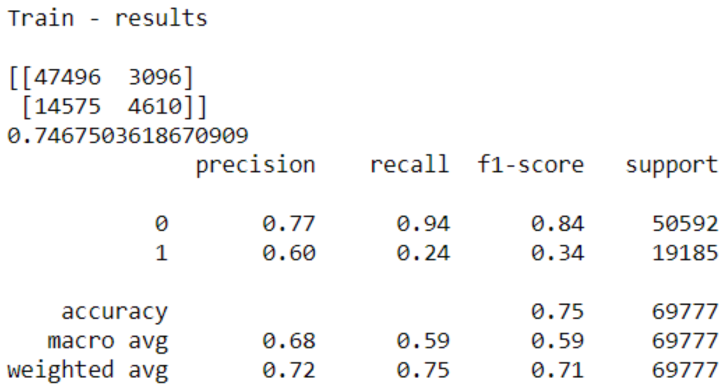
```
(87229, 14)
```

Logistic Regression model:

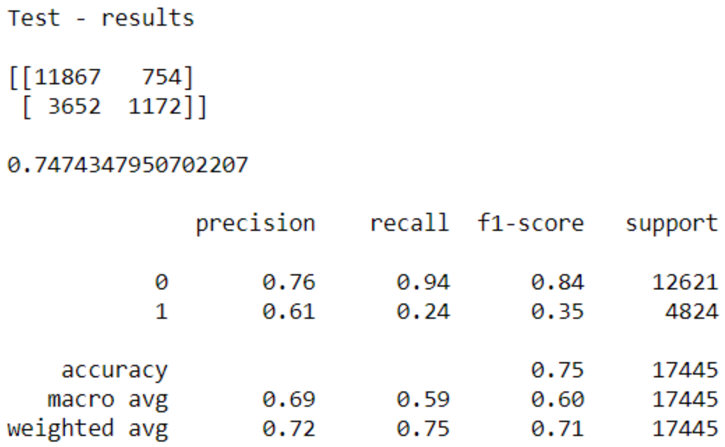


```
logreg.fit(x_train, yy_train)
yy_train_pred = logreg.predict(x_train)
yy_train_prob = logreg.predict_proba(x_train)

print('Train - results')
print()
print(confusion_matrix(yy_train, yy_train_pred))
print(accuracy_score(yy_train, yy_train_pred))
print(classification_report(yy_train, yy_train_pred))

yy_test_pred = logreg.predict(x_test)
yy_test_prob = logreg.predict_proba(x_test)

print('\n')
print('Test - results')
print()
print(confusion_matrix(yy_test, yy_test_pred))
print()
print(accuracy_score(yy_test, yy_test_pred))
print()
print(classification_report(yy_test, yy_test_pred))
```

Classification Report:

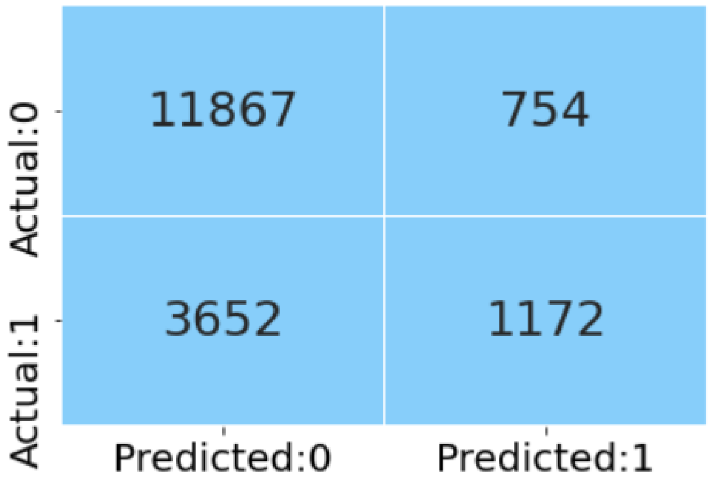


```
Train - results

[[47496  3096]
 [14575  4610]]
0.7467503618670909
              precision    recall  f1-score   support

           0       0.77      0.94      0.84     50592
           1       0.60      0.24      0.34     19185

    accuracy                           0.75     69777
   macro avg       0.68      0.59      0.59     69777
weighted avg       0.72      0.75      0.71     69777
```

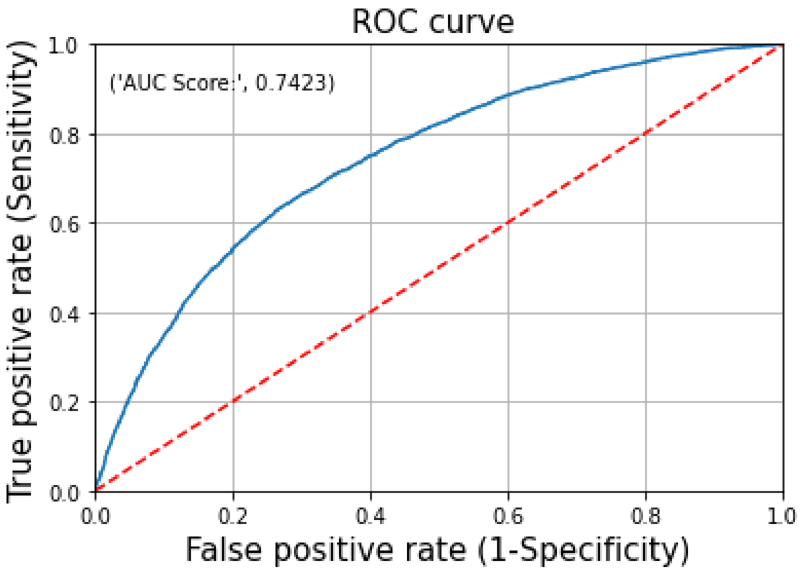


```
Test - results

[[11867   754]
 [ 3652  1172]]

0.7474347950702207

              precision    recall  f1-score   support

           0       0.76      0.94      0.84     12621
           1       0.61      0.24      0.35      4824

    accuracy                           0.75     17445
   macro avg       0.69      0.59      0.60     17445
weighted avg       0.72      0.75      0.71     17445
```

Confusion Matrix:



Roc-curve:



MODEL BUILDING AND METHODS

From EDA, we observed presence of high cardinality is certain categorical variables. In order to build models, we need to use appropriate encoding techniques to address this issue. Also, for building better models, we need to transform the numerical variables.

Encoding of Categorical Variables and Numerical Values Treatment and Feature Engineering

Under feature engineering we perform the following-

1. Extract total customers by combining adults, babies and children
2. Total bookings made previously by combing previous cancellations and previous bookings not cancelled.
3. Room change done from assigned room type and booked room type.
4. Arrival date by extracting each value from day, month and year column.

Under encoding we perform the following-

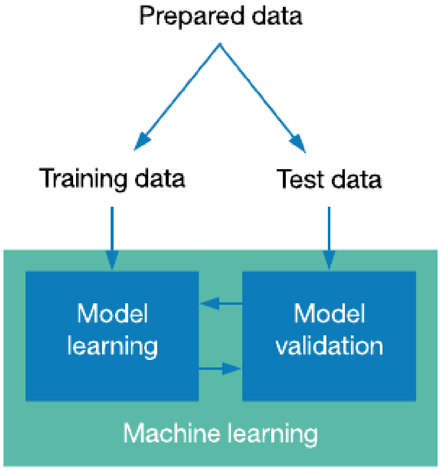
1. Encode month as January:1, February:2 etc.
2. Encode country based on the higher frequency of the countries present since this category has high cardinality.
3. Encode agent based on the higher frequency of the agents present since this category has high cardinality.
4. For rest of the columns we perform dummy encoding

For the numerical data we scale the data using minmax scaler.

Model Building

Step by step approach for model building: -

1. After performing encoding for the categorical features and transforming the numerical variables, we split the data into train data and test data. Model data uses train data to learn whereas test data is used to evaluate or validate the trained model.



2. The baseline model which we built was logistic regression without performing any transformation on numerical variables and using label encoder for categorical variables. Now after performing transformation and encoding the categorical variables, we again build logistic regression model.

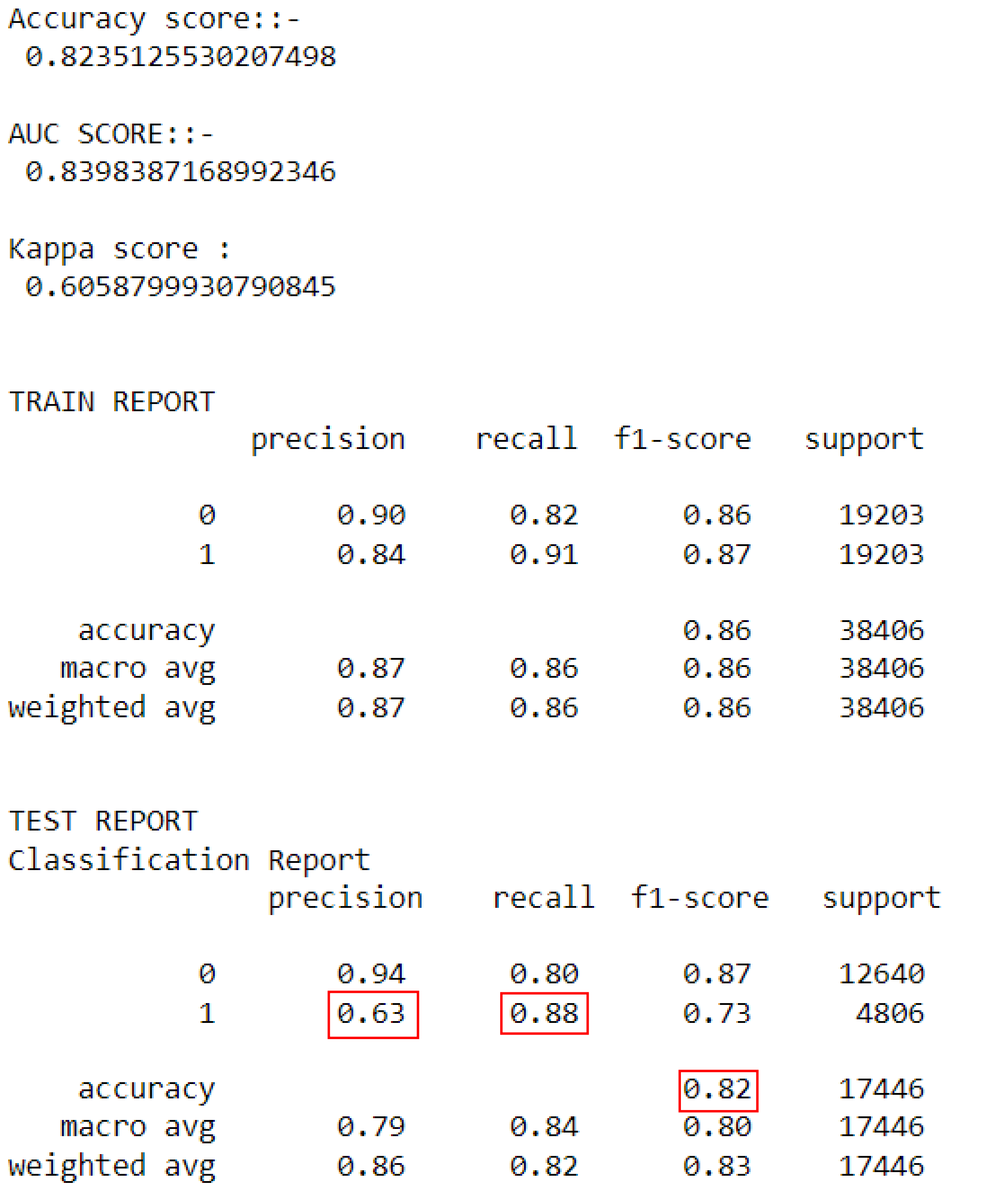
3. Next, we build non-linear models such as Decision Tree, Random Forest, Gradient Boost, Ada Boost and XG Boost Classifier. For these models, we perform hyper parameter tuning and feature selection techniques using SFS and RFE. Also, since there is presence of moderate amount of class imbalance, we perform oversampling and under sampling.

4. From these models, we do not achieve desired amount of accuracy, precision and recallEven though we achieve moderate level of accuracy for the model, we get low precision and recall values. In order to address this issue and the cardinality in the data, we build a model using CatBoost classifier.

5. A CatBoost model handles the categorical data on its own hence there is no need to do any encoding. Therefore, we split the data into train and test with test ratio of 0.2 and fit the model.

6. In order to further improve the model, we perform under-sampling and over-sampling to address the presence of moderate amount of class imbalance and again build the model. From here we observe that we obtain a better model with under sampling. Even though under sampling leads to loss of information, it contributes towards realistic data.

7. We also perform feature selection on the CatBoost model, but we get a model with lower recall, precision and accuracy. Hence, the best model we get is the CatBoost model with Under-Sampling.



```
Accuracy score::-
 0.8235125530207498

AUC SCORE::-
 0.8398387168992346

Kappa score :
 0.6058799930790845


TRAIN REPORT
              precision    recall  f1-score   support

           0       0.90      0.82      0.86     19203
           1       0.84      0.91      0.87     19203

    accuracy                           0.86     38406
   macro avg       0.87      0.86      0.86     38406
weighted avg       0.87      0.86      0.86     38406


TEST REPORT
Classification Report
               precision    recall  f1-score   support

           0       0.94      0.80      0.87     12640
           1       0.63      0.88      0.73      4806

    accuracy                           0.82     17446
   macro avg       0.79      0.84      0.80     17446
weighted avg       0.86      0.82      0.83     17446
```

Model Understanding

From all the models built, we get a better fitting model of **CatBoost** with **Under-Sampling**.

CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can work with diverse data types to help solve a wide range of problems that businesses face today. To top it up, it provides best-in-class accuracy.

It is especially powerful in two ways:

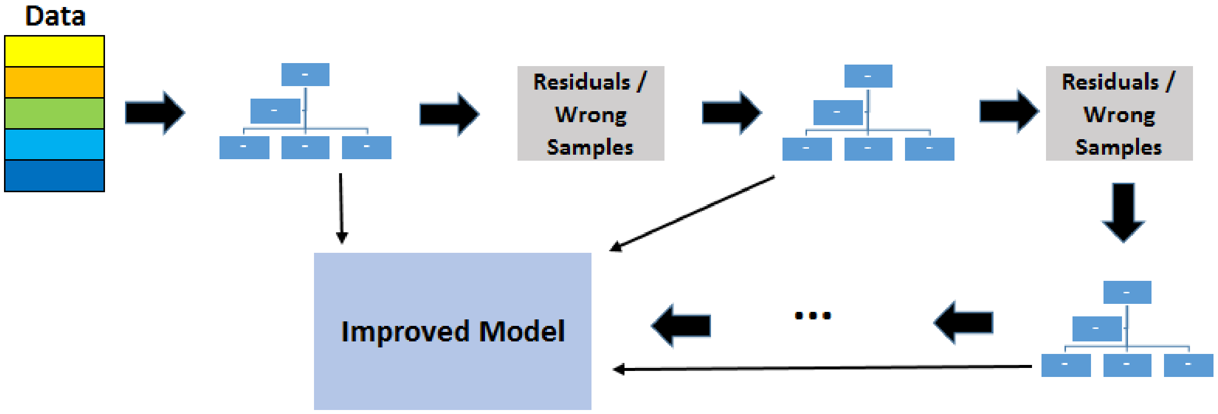
- It yields state-of-the-art results without extensive data training typically required by other machine learning methods, and
- Provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems.

"CatBoost" name comes from two words "**Cat**egory" and "**Boost**ing".
The library works well with multiple **Cat**egories of data, such as audio, text, image including historical data.
"**Boost**" comes from gradient boosting machine learning algorithm as this library is based on gradient boosting library. Gradient boosting is a powerful machine learning algorithm that is widely applied to multiple types of business challenges like fraud detection, recommendation items, forecasting and it performs well also. It can also return very good result with relatively less data, unlike DL models that need to learn from a massive amount of data. Hence CatBoost is an algorithm for gradient boosting on decision trees.

Some of the advantages of CatBoost algorithm are :-

- **Performance:** CatBoost provides state of the art results and it is competitive with any leading machine learning algorithm on the performance front.
- **Handling Categorical features automatically:** We can use CatBoost without any explicit pre-processing to convert categories into numbers. CatBoost converts categorical values into numbers using various statistics on combinations of categorical features and combinations of categorical and numerical features.
- **Robust:** It reduces the need for extensive hyper-parameter tuning and lower the chances of overfitting also which leads to more generalized models. Although, CatBoost has multiple parameters to tune and it contains parameters like the number of trees, learning rate, regularization, tree depth, fold size, bagging temperature and others.
- **Easy-to-use:** You can use CatBoost from the command line, using a user-friendly API.

In **Gradient Boosting**, each predictor tries to improve on its predecessor by reducing the errors. But the fascinating idea behind Gradient Boosting is that instead of fitting a predictor on the data at each iteration, it actually fits a new predictor to the residual errors made by the previous predictor.



Gradient boosting involves three elements:

1. A loss function to be optimized. The loss function used depends on the type of problem being solved. For a classification problem, logarithmic loss is used
2. A weak learner to make predictions. Decision trees are used as the weak learner in gradient boosting. Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini to minimize the loss.
3. An additive model to add weak learners to minimize the loss function. Trees are added one at a time, and existing trees in the model are not changed.

Over-Sampling and Under-Sampling

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary from a slight bias to a severe imbalance where there is one example in the minority class for hundreds, thousands, or millions of examples in the majority class or classes.

Imbalanced classifications pose a challenge for predictive modelling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class.

Imbalanced classification refers to a classification predictive modelling problem where the number of examples in the training dataset for each class label is not balanced. That is, where the class distribution is not equal or close to equal, and is instead biased or skewed.

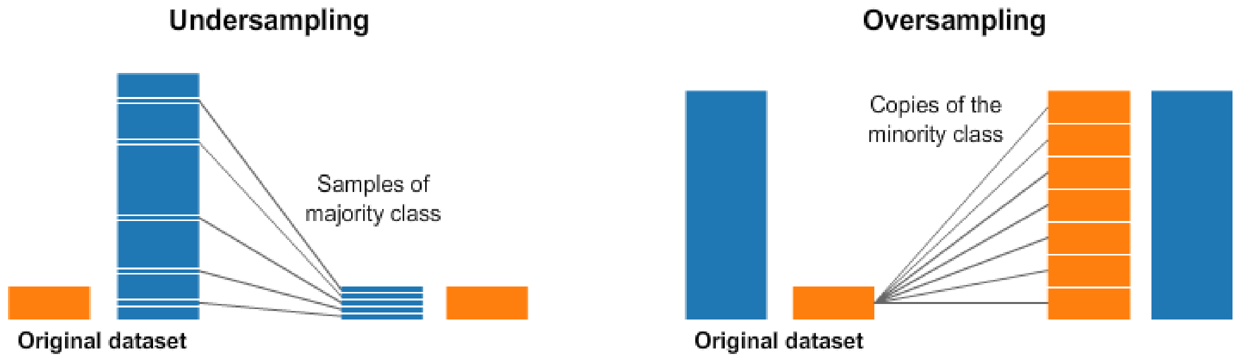
One approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called **under-sampling**, and to duplicate examples from the minority class, called **over-sampling**.

Random **over-sampling** involves randomly duplicating examples from the minority class and adding them to the training dataset.

Examples from the training dataset are selected randomly with replacement. This means that examples from the minority class can be chosen and added to the new "*more balanced*" training

dataset multiple times; they are selected from the original training dataset, added to the new training dataset, and then returned or "*replaced*" in the original dataset, allowing them to be selected again.
In some cases, seeking a balanced distribution for a severely imbalanced dataset can cause affected algorithms to overfit the minority class, leading to increased generalization error. The effect can be better performance on the training dataset, but worse performance on the holdout or test dataset.

Random **under-sampling** involves randomly selecting examples from the majority class to delete from the training dataset.
This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.
A limitation of under-sampling is that examples from the majority class are deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary. Given that examples are deleted randomly, there is no way to detect or preserve "*good*" or more information-rich examples from the majority class.



Precision and Recall Trade-Off

Precision: It is the accuracy of positive predictions.

$$Precision = \frac{True\ Positive}{True\ Postive + False\ Positive}$$

That mean, when the model predicts that a booking will be cancelled, it is correct around %precision times.
Recall: It is the ratio of positive instance that are correctly detected. It is also called sensitivity.

$$Recall = \frac{True\ Positive}{True\ Postive + False\ Negative}$$

Hence, for all the bookings that were actually cancelled, recall tells us how many the model correctly identified as being cancelled.

Accuracy: Accuracy is the ratio of the total number of correct predictions and the total number of predictions.

$$Accuracy = \frac{True\ Positive}{True\ Postive + False\ Negative + False\ Positive + False\ Negative}$$

Using accuracy as a defining metric for our model does make sense intuitively, but more often than not, it is always advisable to use Precision and Recall too. There might be other situations where our accuracy is very high, but our precision or recall is low.

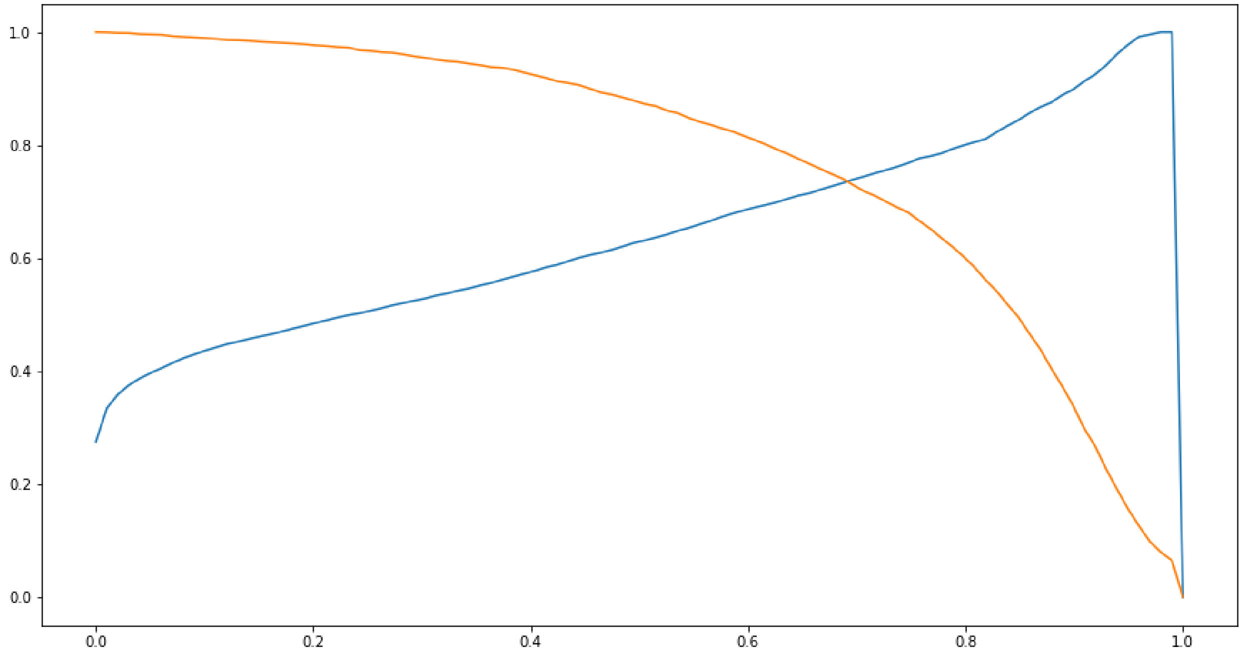
F1-score: F1-score is the Harmonic mean of the Precision and Recall.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Unfortunately, we can’t have both precision and recall high. If we increase precision, it will reduce recall, and vice versa. This is called the precision/recall trade-off.

In order to determine the best value for precision or recall, we plot a curve between precision, recall and threshold. For each threshold value, we have different values of precision and recall.

The plot for precision, recall and threshold for CatBoost model with Under-Sampling is given below:-

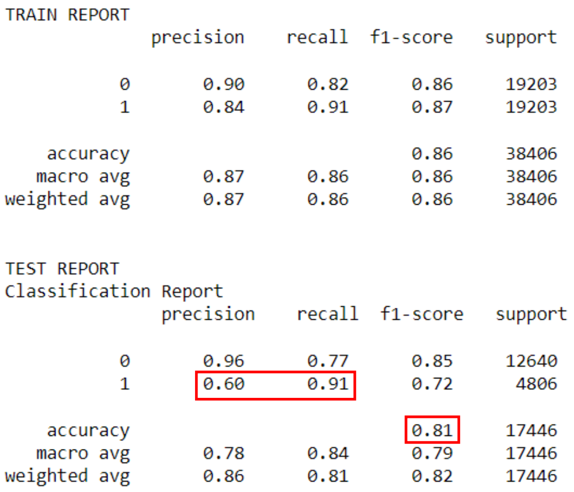


Case 1 - High Recall Value

By predicting the possibility of a booking being cancelled, the business can make some informed decisions, such as open the rooms for booking which are more likely to be cancelled. But if the recall of the model is low, this might lead to overbooking which would create a difficult scenario for the hotel managers. Hence for a good business of the hotel, we would like to preserve the recall of the model with decent amount of accuracy.

Also, MNCs in the hotel industry have a marketing team which can connect with the customers who are likely to cancel the booking. Hence for efficient operation of the marketing team, we need to have a higher recall, since a marketing agent would not like to contact a person who is less likely to cancel the booking.

From the graph, we take threshold of 0.44 for a good recall with decent accuracy. With this, we get a model with the following report: -

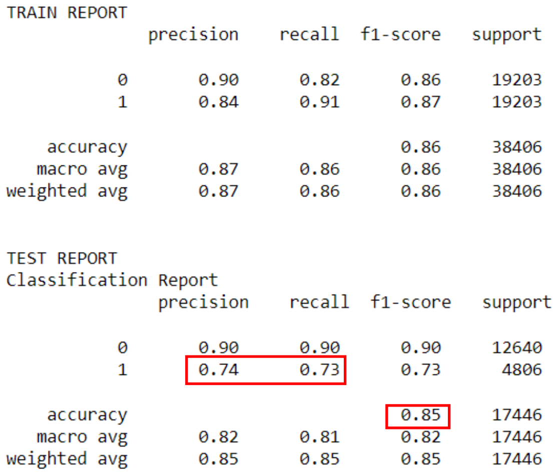


```
TRAIN REPORT
              precision    recall  f1-score   support

           0       0.90      0.82      0.86     19203
           1       0.84      0.91      0.87     19203

    accuracy                           0.86     38406
   macro avg       0.87      0.86      0.86     38406
weighted avg       0.87      0.86      0.86     38406


TEST REPORT
Classification Report
               precision    recall  f1-score   support

           0       0.96      0.77      0.85     12640
           1       0.60      0.91      0.72      4806

    accuracy                           0.81     17446
   macro avg       0.78      0.84      0.79     17446
weighted avg       0.86      0.81      0.82     17446
```

Case 2 – Decent precision and recall

Small business owners in the hotel industry do not have a marketing team. For them a model is required which will have a good precision as well as recall.

But if for the business, we need to have a good precision and recall, then from the graph, we use a threshold of 0.7. From here we get the following classification report: -



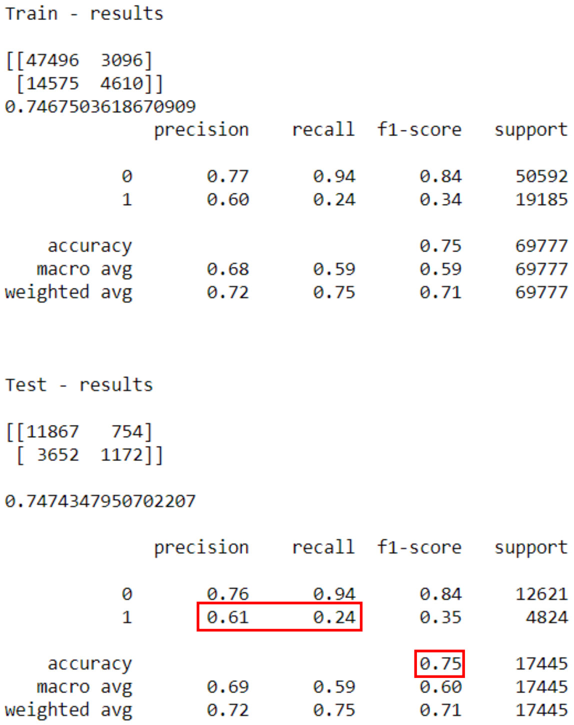
```
TRAIN REPORT
              precision    recall  f1-score   support

           0       0.90      0.82      0.86     19203
           1       0.84      0.91      0.87     19203

    accuracy                           0.86     38406
   macro avg       0.87      0.86      0.86     38406
weighted avg       0.87      0.86      0.86     38406


TEST REPORT
Classification Report
               precision    recall  f1-score   support

           0       0.90      0.90      0.90     12640
           1       0.74      0.73      0.73      4806

    accuracy                           0.85     17446
   macro avg       0.82      0.81      0.82     17446
weighted avg       0.85      0.85      0.85     17446
```

COMPARISON AND IMPLICATIONS

We compare the best model we get i.e. CatBoost model with Under-Sampling and the Logistic Regression model built by performing feature transformation and encoding.

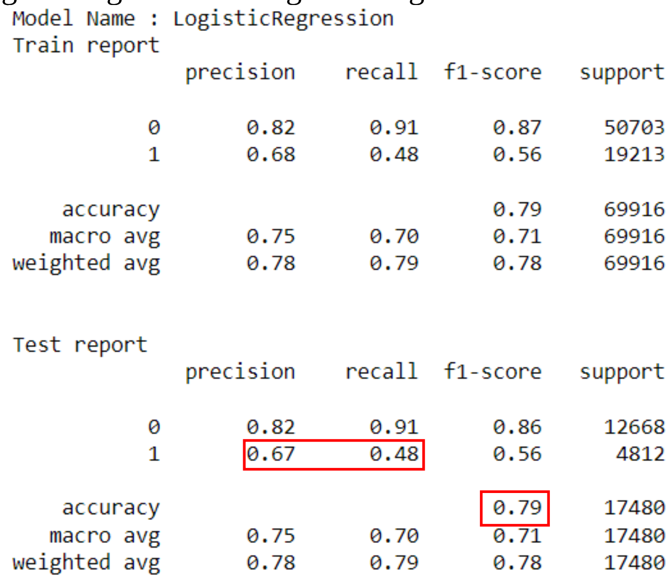
Comparison to Benchmark

Base Line Model – Logistic Regression without performing any feature transformation and using Label Encoder.



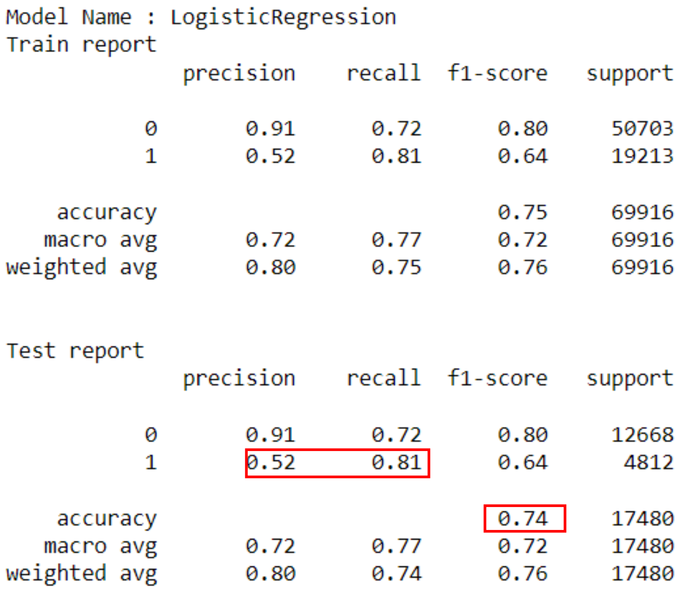
```
Train - results

[[47496  3096]
 [14575  4610]]
0.7467503618670909
              precision    recall  f1-score   support

           0       0.77      0.94      0.84     50592
           1       0.60      0.24      0.34     19185

    accuracy                           0.75     69777
   macro avg       0.68      0.59      0.59     69777
weighted avg       0.72      0.75      0.71     69777


Test - results

[[11867   754]
 [ 3652  1172]]

0.7474347950702207

              precision    recall  f1-score   support

           0       0.76      0.94      0.84     12621
           1       0.61      0.24      0.35      4824

    accuracy                           0.75     17445
   macro avg       0.69      0.59      0.60     17445
weighted avg       0.72      0.75      0.71     17445
```

Logistic Regression Model 1 – Logistic regression model by scaling the numerical data, performing feature engineering and encoding the categorical data.



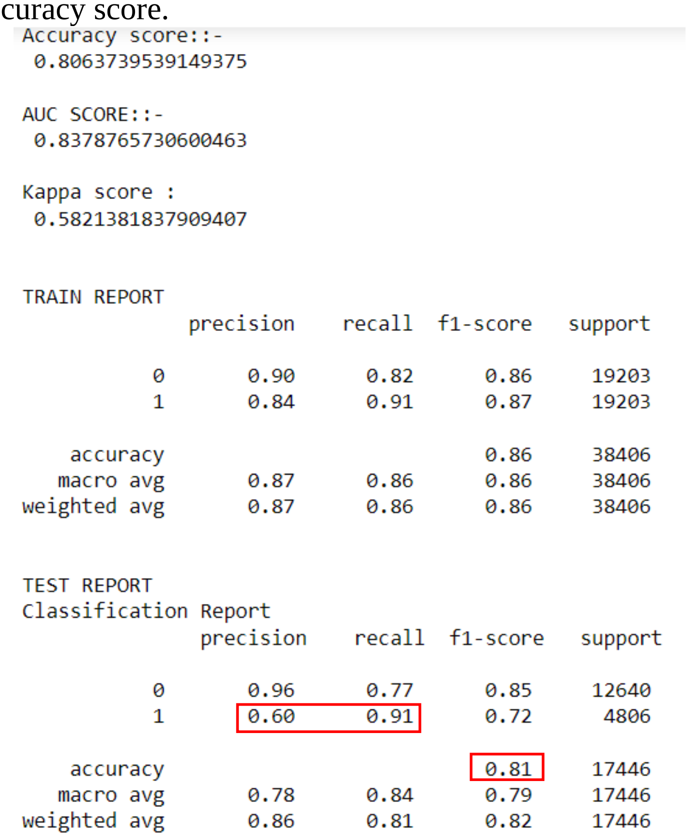
```
Model Name : LogisticRegression
Train report
              precision    recall  f1-score   support

           0       0.82      0.91      0.87     50703
           1       0.68      0.48      0.56     19213

    accuracy                           0.79     69916
   macro avg       0.75      0.70      0.71     69916
weighted avg       0.78      0.79      0.78     69916


Test report
              precision    recall  f1-score   support

           0       0.82      0.91      0.86     12668
           1       0.67      0.48      0.56      4812

    accuracy                           0.79     17480
   macro avg       0.75      0.70      0.71     17480
weighted avg       0.78      0.79      0.78     17480
```

Logistic Regression Model with hyper-parameter tuning using Randomized Search – We perform hyperparameter tuning for the final base line model obtained above using Randomised Search.



```
Model Name : LogisticRegression
Train report
              precision    recall  f1-score   support

           0       0.91      0.72      0.80     50703
           1       0.52      0.81      0.64     19213

    accuracy                           0.75     69916
   macro avg       0.72      0.77      0.72     69916
weighted avg       0.80      0.75      0.76     69916


Test report
              precision    recall  f1-score   support

           0       0.91      0.72      0.80     12668
           1       0.52      0.81      0.64      4812

    accuracy                           0.74     17480
   macro avg       0.72      0.77      0.72     17480
weighted avg       0.80      0.74      0.76     17480
```

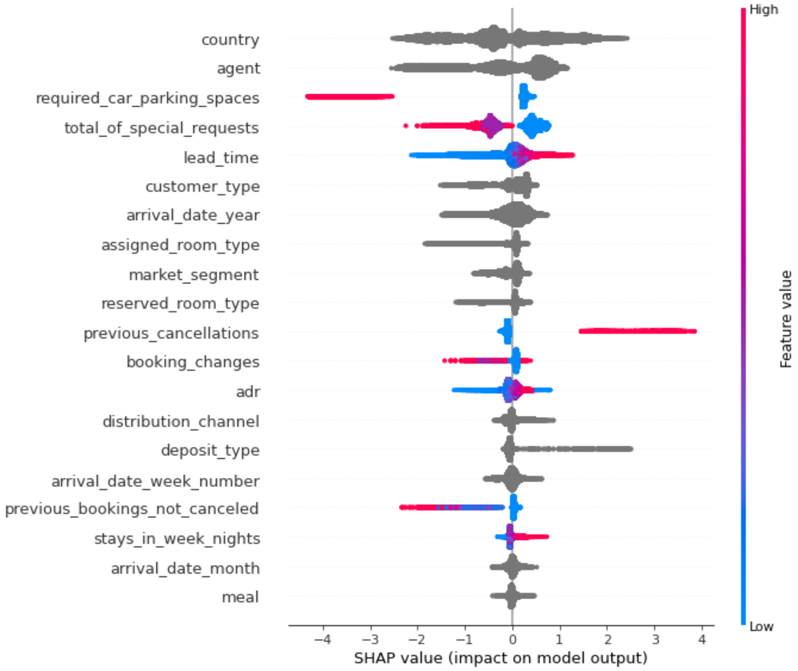
CatBoost model with Under-Sampling – CatBoost model with under-sampling having a good recall and a decent accuracy score.



```
Accuracy score::-
 0.8063739539149375

AUC SCORE::-
 0.8378765730600463

Kappa score :
 0.5821381837909407


TRAIN REPORT
              precision    recall  f1-score   support

           0       0.90      0.82      0.86     19203
           1       0.84      0.91      0.87     19203

    accuracy                           0.86     38406
   macro avg       0.87      0.86      0.86     38406
weighted avg       0.87      0.86      0.86     38406


TEST REPORT
Classification Report
               precision    recall  f1-score   support

           0       0.96      0.77      0.85     12640
           1       0.60      0.91      0.72      4806

    accuracy                           0.81     17446
   macro avg       0.78      0.84      0.79     17446
weighted avg       0.86      0.81      0.82     17446
```

Model Explanation

SHAP and LIME are libraries to explain the model. While implementing both of libraries we found SHAP library to be visually more explainable. Hence, we go ahead with SHAP library to explain the output of our model.



Summary plots are easy-to-read visualizations which bring the whole data to a single plot. All of the features are listed in y-axis in the rank order, the top one being the most contributor to the predictions and the bottom one being the least or zero-contributor. Shap values are provided in the x-axis. A value of zero represents no contribution whereas contributions increase as the SHAP value moves away from zero. Each circular dot in the plot represents a single data point. Colour of the dot denotes the value of that corresponding feature.
From the above plot, we can see that country, agent, car parking space, special requests, lead time, customer type, year, room type, market segment are some of the important features.

Inferences and Recommendations

Inference – Some of the important features are country, agent, car parking space, special requests, lead time, customer type, year, room type, market segment, etc.

1. Less number of car parking space leads to more cancellations.
2. Also, more amount of lead time leads to more cancellations.
3. The bookings which were previously cancelled lead to cancellations.
4. More number of week nights lead to more cancellations.
5. Less booking changes lead to more cancellations.
6. Less special requests also lead to more cancellations

Recommendations – Some business recommendations are-

1. Restrict the number of days until the customer can cancel their booking free of cost.

2. Also reduce the lead time in advance a customer can make the booking so that it can lead to lesser cancellations.
3. Keep a report of the customers cancelling their booking previously and target them with more offers so that they don’t cancel the booking next time.
4. Give certain offers, services to the customers staying for more nights so that they don’t cancel the booking.

LIMITATIONS, CHALLENGES AND SCOPE

Limitations of Data

Few of the limitations are: -

1. The dataset belongs to Portugal and consists of data of only two hotels. The model will be more robust if the data would have belonged from different regions of the world.
2. Also, the duration of data collected is from 1st of July of 2015 and to 31st of August 2017. Due to this there isn’t even distribution of the data.

Challenges

Few of the challenges faced are: -

1. High cardinality results in huge training effort in model tuning due to increase in model complexity (i.e. more number of features)
2. We also faced challenges on robust model tuning on all the models. Due to computational limitations, we are limited to using Randomized Search as a hyper parameter tuning technique instead of using Grid Search, HyperOpt etc.

Scope

Scope for some future work is: -

1. Perform hyper parameter tuning for the CatBoost model since due to lower processing power of our laptops, we couldn’t do that.
2. Exploring Google collab as an option for model training and tuning with faster lead time.
3. Exploring some robust data sampling technique as part of choosing smaller sample (a true representation of population data) from the population data.