ML-1 Coded Project Report

Prepared By: Parthasarathi Behura

CONTENTS: INN Hotels Group Page

LIST OF FIGURES:

LIST OF TABLES:

ML-1 Project

Business Context

A significant number of hotel bookings are called off due to cancellations or no-shows. The typical reasons for cancellations include change of plans, scheduling conflicts, etc. This is often made easier by the option to do so free of charge or preferably at a low cost which is beneficial to hotel guests but it is a less desirable and possibly revenue-diminishing factor for hotels to deal with. Such losses are particularly high on last-minute cancellations.

The new technologies involving online booking channels have dramatically changed customers' booking possibilities and behavior. This adds a further dimension to the challenge of how hotels handle cancellations, which are no longer limited to traditional booking and guest characteristics.

The cancellation of bookings impact a hotel on various fronts:
1. Loss of resources (revenue) when the hotel cannot resell the room.
2. Additional costs of distribution channels by increasing commissions or paying for publicity to help sell these rooms.
3. Lowering prices last minute, so the hotel can resell a room, resulting in reducing the profit margin.
4. Human resources to make arrangements for the guests.

Objective

The increasing number of cancellations calls for a Machine Learning based solution that can help in predicting which booking is likely to be canceled. INN Hotels Group has a chain of hotels in Portugal, they are facing problems with the high number of booking cancellations and have reached out to your firm for data-driven solutions. You as a data scientist have to analyze the data provided to find which factors have a high influence on booking cancellations, build a predictive model that can predict which booking is going to be canceled in advance, and help in formulating profitable policies for cancellations and refunds.

Data Description

The data contains the different attributes of customers' booking details. The detailed data dictionary is given below.

Data Dictionary:

- Booking_ID: the unique identifier of each booking
- no_of_adults: Number of adults
- no_of_children: Number of Children
- no_of_weekend_nights: Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
- no_of_week_nights: Number of weeknights (Monday to Friday) the guest stayed or booked to stay at the hotel

- type_of_meal_plan: Type of meal plan booked by the customer:
    - Not Selected – No meal plan selected
    - Meal Plan 1 – Breakfast
    - Meal Plan 2 – Half board (breakfast and one other meal)
    - Meal Plan 3 – Full board (breakfast, lunch, and dinner)
- required_car_parking_space: Does the customer require a car parking space? (0 - No, 1- Yes)
- room_type_reserved: Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels Group
- lead_time: Number of days between the date of booking and the arrival date
- arrival_year: Year of arrival date
- arrival_month: Month of arrival date
- arrival_date: Date of the month
- market_segment_type: Market segment designation.
- repeated_guest: Is the customer a repeated guest? (0 - No, 1- Yes)
- no_of_previous_cancellations: Number of previous bookings that were canceled by the customer prior to the current booking
- no_of_previous_bookings_not_canceled: Number of previous bookings not canceled by the customer prior to the current booking
- avg_price_per_room: Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
- no_of_special_requests: Total number of special requests made by the customer (e.g. high floor, view from the room, etc)
- booking_status: Flag indicating if the booking was canceled or not.

Imported the libraries for the Data are

- import pandas as pd
- import numpy as np
- import matplotlib.pyplot as plt
- import seaborn as sns
- from sklearn.model_selection import train_test_split
- from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
- from sklearn.neighbors import KNeighborsClassifier
- from sklearn.naive_bayes import GaussianNB
- from sklearn.tree import DecisionTreeClassifier
- import statsmodels.api as sm
- from sklearn import tree

DATA PROCESSING:

1. There are some information about the dataset, decision makers should have a look.

- The dataset is having 19 columns.
- There is a look on the 5 sample rows to check the data type.

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	room_type_reserved
0	INN00001	2	0	1	2	Meal Plan 1	0	Room_Type 1
1	INN00002	2	0	2	3	Not Selected	0	Room_Type 1
2	INN00003	1	0	2	1	Meal Plan 1	0	Room_Type 1
3	INN00004	2	0	0	2	Meal Plan 1	0	Room_Type 1
4	INN00005	2	0	1	1	Not Selected	0	Room_Type 1

lead_time	arrival_year	arrival_month	arrival_date	market_segment_type	repeated_guest	no_of_previous_cancellations
224	2017	10	2	Offline	0	0
5	2018	11	6	Online	0	0
1	2018	2	28	Online	0	0
211	2018	5	20	Online	0	0
48	2018	4	11	Online	0	0

no_of_previous_cancellations	no_of_previous_bookings_not_canceled	avg_price_per_room	no_of_special_requests	booking_status
0	0	65.00000	0	Not_Canceled
0	0	106.68000	1	Not_Canceled
0	0	60.00000	0	Canceled
0	0	100.00000	0	Canceled
0	0	94.50000	0	Canceled

Table 1: Top five rows of the dataset

2. While having a look on the data set information, it is found that there are 14 numerical and 5 categorial variables. The below table contains the same information.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
 4   no_of_week_nights                     36275 non-null  int64
 5   type_of_meal_plan                     36275 non-null  object
 6   required_car_parking_space            36275 non-null  int64
 7   room_type_reserved                    36275 non-null  object
 8   lead_time                             36275 non-null  int64
 9   arrival_year                          36275 non-null  int64
 10  arrival_month                         36275 non-null  int64
 11  arrival_date                          36275 non-null  int64
 12  market_segment_type                   36275 non-null  object
 13  repeated_guest                        36275 non-null  int64
 14  no_of_previous_cancellations          36275 non-null  int64
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64
 18  booking_status                        36275 non-null  object
dtypes: float64(1), int64(13), object(5)
memory usage: 5.3+ MB
```

Table 2: Basic information of the data type

3. Checking the statistical summary of the data.

	count	mean	std	min	25%	50%	75%	max
no_of_adults	36275.00000	1.84496	0.51871	0.00000	2.00000	2.00000	2.00000	4.00000
no_of_children	36275.00000	0.10528	0.40265	0.00000	0.00000	0.00000	0.00000	10.00000
no_of_weekend_nights	36275.00000	0.81072	0.87064	0.00000	0.00000	1.00000	2.00000	7.00000
no_of_week_nights	36275.00000	2.20430	1.41090	0.00000	1.00000	2.00000	3.00000	17.00000
required_car_parking_space	36275.00000	0.03099	0.17328	0.00000	0.00000	0.00000	0.00000	1.00000
lead_time	36275.00000	85.23256	85.93082	0.00000	17.00000	57.00000	126.00000	443.00000
arrival_year	36275.00000	2017.82043	0.38384	2017.00000	2018.00000	2018.00000	2018.00000	2018.00000
arrival_month	36275.00000	7.42365	3.06989	1.00000	5.00000	8.00000	10.00000	12.00000
arrival_date	36275.00000	15.59700	8.74045	1.00000	8.00000	16.00000	23.00000	31.00000
repeated_guest	36275.00000	0.02564	0.15805	0.00000	0.00000	0.00000	0.00000	1.00000
no_of_previous_cancellations	36275.00000	0.02335	0.36833	0.00000	0.00000	0.00000	0.00000	13.00000
no_of_previous_bookings_not_canceled	36275.00000	0.15341	1.75417	0.00000	0.00000	0.00000	0.00000	58.00000
avg_price_per_room	36275.00000	103.42354	35.08942	0.00000	80.30000	99.45000	120.00000	540.00000
no_of_special_requests	36275.00000	0.61966	0.78624	0.00000	0.00000	0.00000	1.00000	5.00000

Table 3: Statistical summary of the data

4. Checking the duplicate values.

Ans:- There are no duplicate values present in the data.

Table 4: Duplicate values in the dataset

5. Checking the null values:

Ans:- There are no missing values present in the dataset.

```
Booking_ID                            0
no_of_adults                          0
no_of_children                        0
no_of_weekend_nights                  0
no_of_week_nights                     0
type_of_meal_plan                     0
required_car_parking_space            0
room_type_reserved                    0
lead_time                             0
arrival_year                          0
arrival_month                         0
arrival_date                          0
market_segment_type                   0
repeated_guest                        0
no_of_previous_cancellations          0
no_of_previous_bookings_not_canceled  0
avg_price_per_room                    0
no_of_special_requests                0
booking_status                        0
dtype: int64
```

Table 5: Inspecting missing values in the dataset

6. There are all the unique values in Booking_ID. So dropped the column.

```
36275
```

- The `Booking_ID` column contains only unique values, so we can drop it

Table 6: Dropping Booking ID column

7. Exploratory Data Analysis

Univariate Analysis:

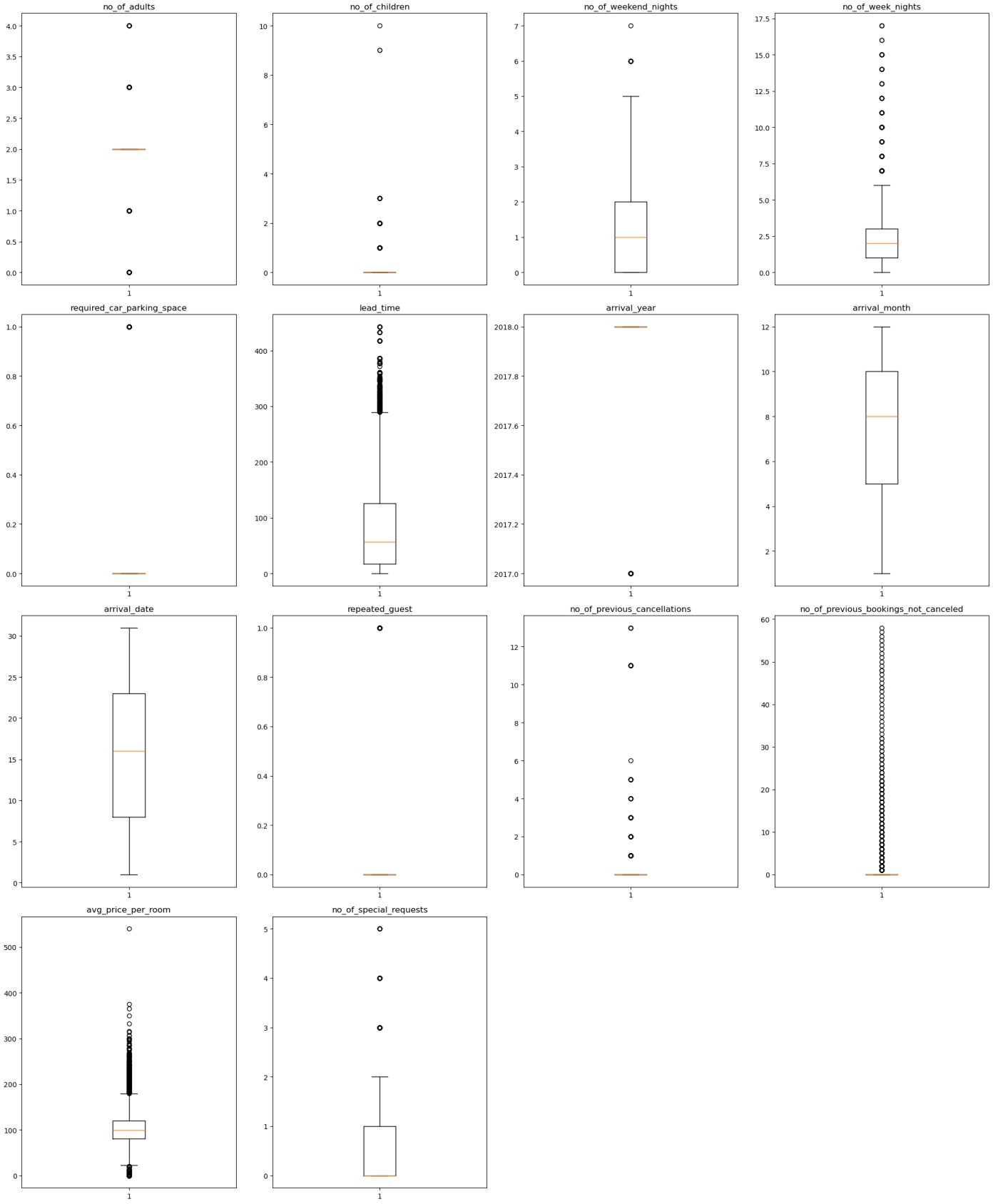
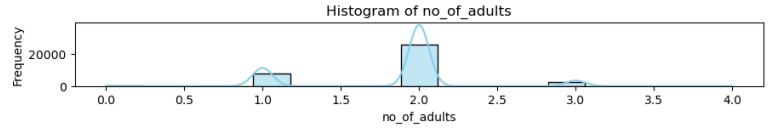
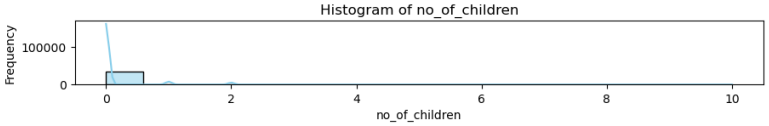
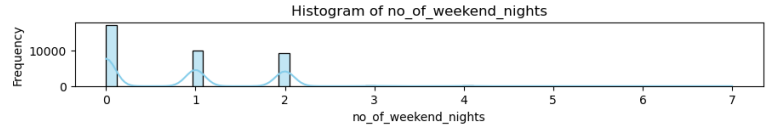
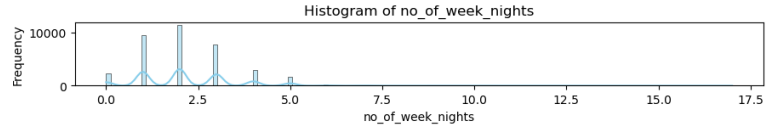


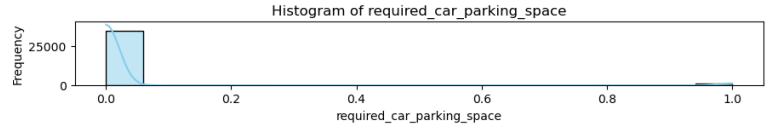
Figure-1 : Box plots of numerical variables

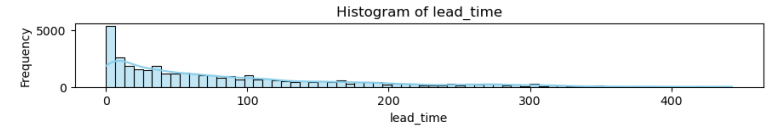

Histogram of no_of_adults
Frequency
20000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
no_of_adults

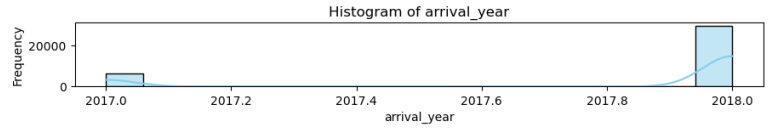

Histogram of no_of_children
Frequency
100000
0
0
2
4
6
8
10
no_of_children

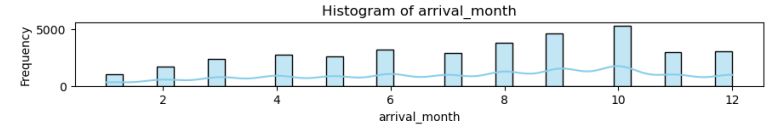

Histogram of no_of_weekend_nights
Frequency
10000
0
0
1
2
3
4
5
6
7
no_of_weekend_nights

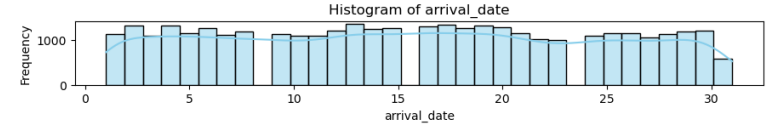

Histogram of no_of_week_nights
Frequency
10000
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
no_of_week_nights

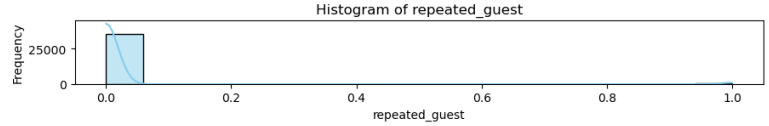

Histogram of required_car_parking_space
Frequency
25000
0
0.0
0.2
0.4
0.6
0.8
1.0
required_car_parking_space

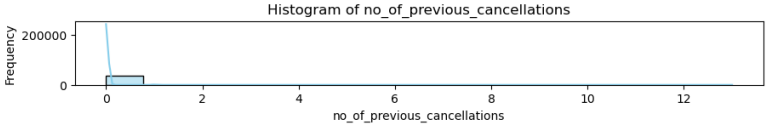

Histogram of lead_time
Frequency
5000
0
0
100
200
300
400
lead_time

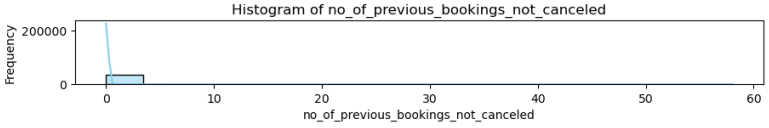

Histogram of arrival_year
Frequency
20000
0
2017.0
2017.2
2017.4
2017.6
2017.8
2018.0
arrival_year

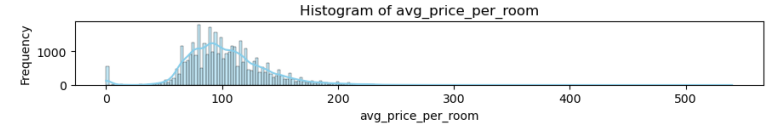

Histogram of arrival_month
Frequency
5000
0
2
4
6
8
10
12
arrival_month


Histogram of arrival_date
Frequency
1000
0
0
5
10
15
20
25
30
arrival_date


Histogram of repeated_guest
Frequency
25000
0
0.0
0.2
0.4
0.6
0.8
1.0
repeated_guest


Histogram of no_of_previous_cancellations
Frequency
200000
0
0
2
4
6
8
10
12
no_of_previous_cancellations


Histogram of no_of_previous_bookings_not_canceled
Frequency
200000
0
0
10
20
30
40
50
60
no_of_previous_bookings_not_canceled


Histogram of avg_price_per_room
Frequency
1000
0
0
100
200
300
400
500
avg_price_per_room

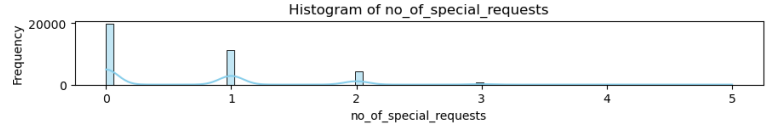

Histogram of no_of_special_requests
Frequency
20000
0
0
1
2
3
4
5
no_of_special_requests

Figure-2 : Histogram of numerical variables

Individually checking for graphs.

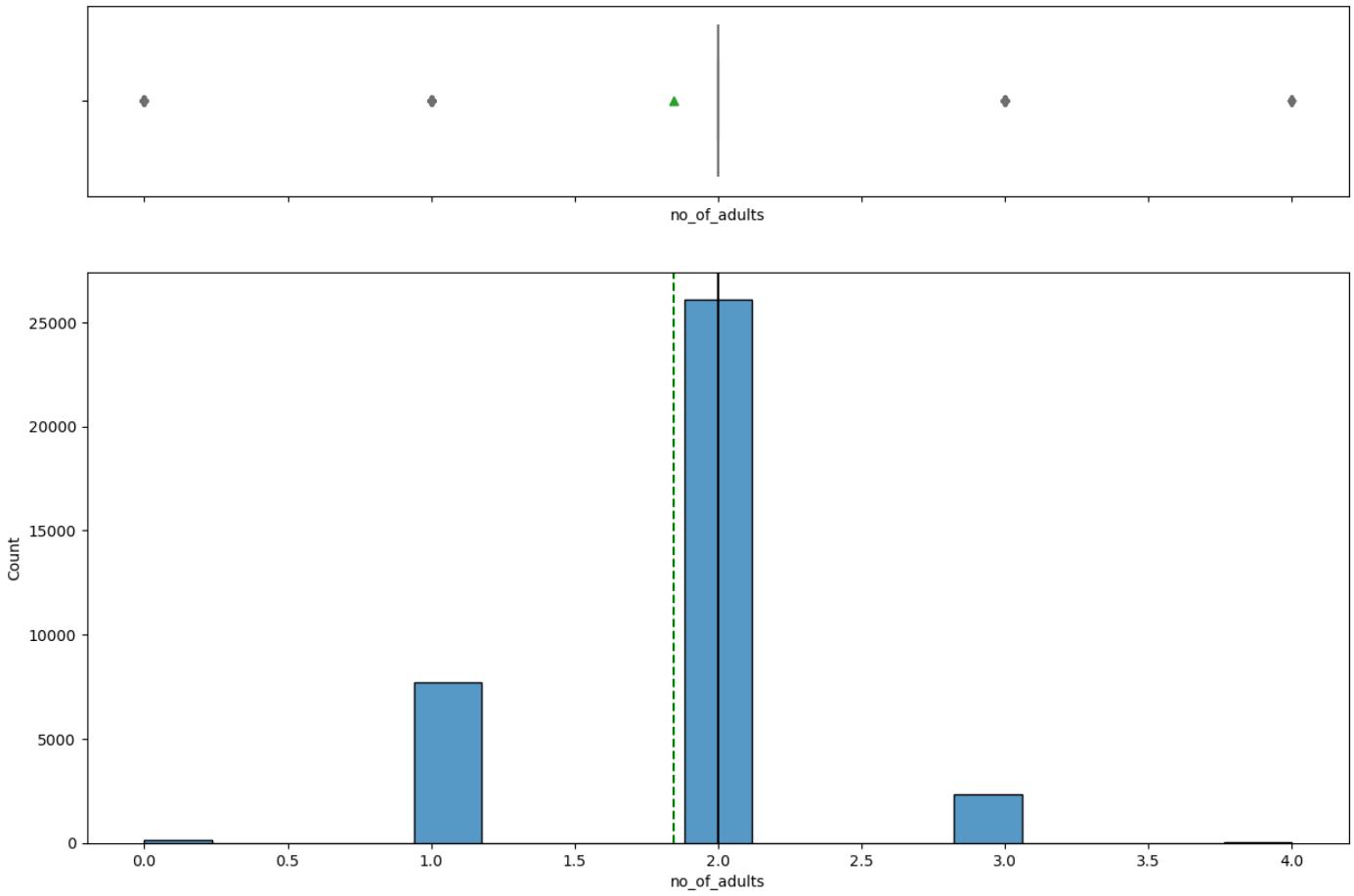


Figure-3 : Histo-box plot of no_of_adults

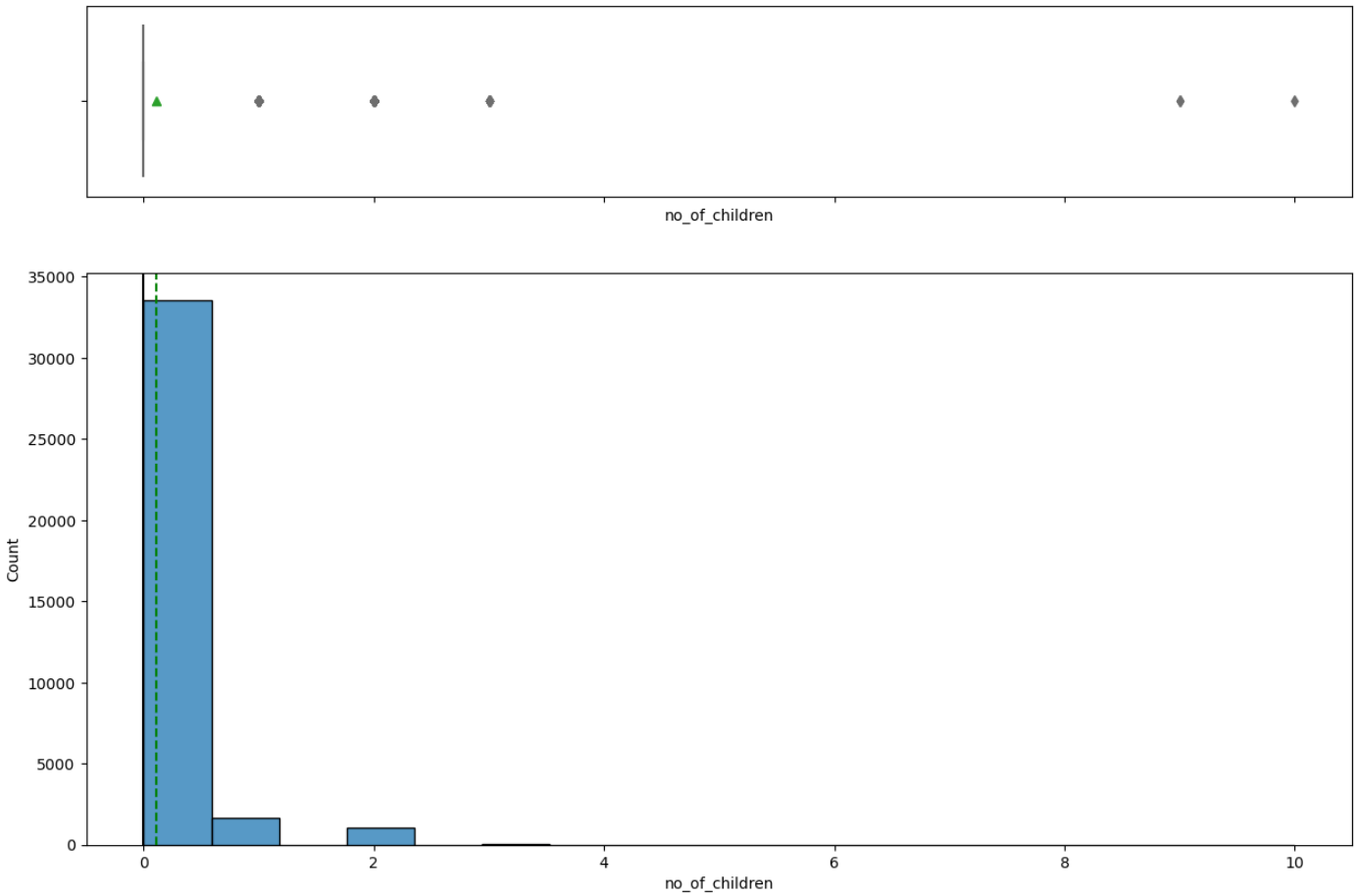


Figure-4 : Histo-box plot of no_of_children

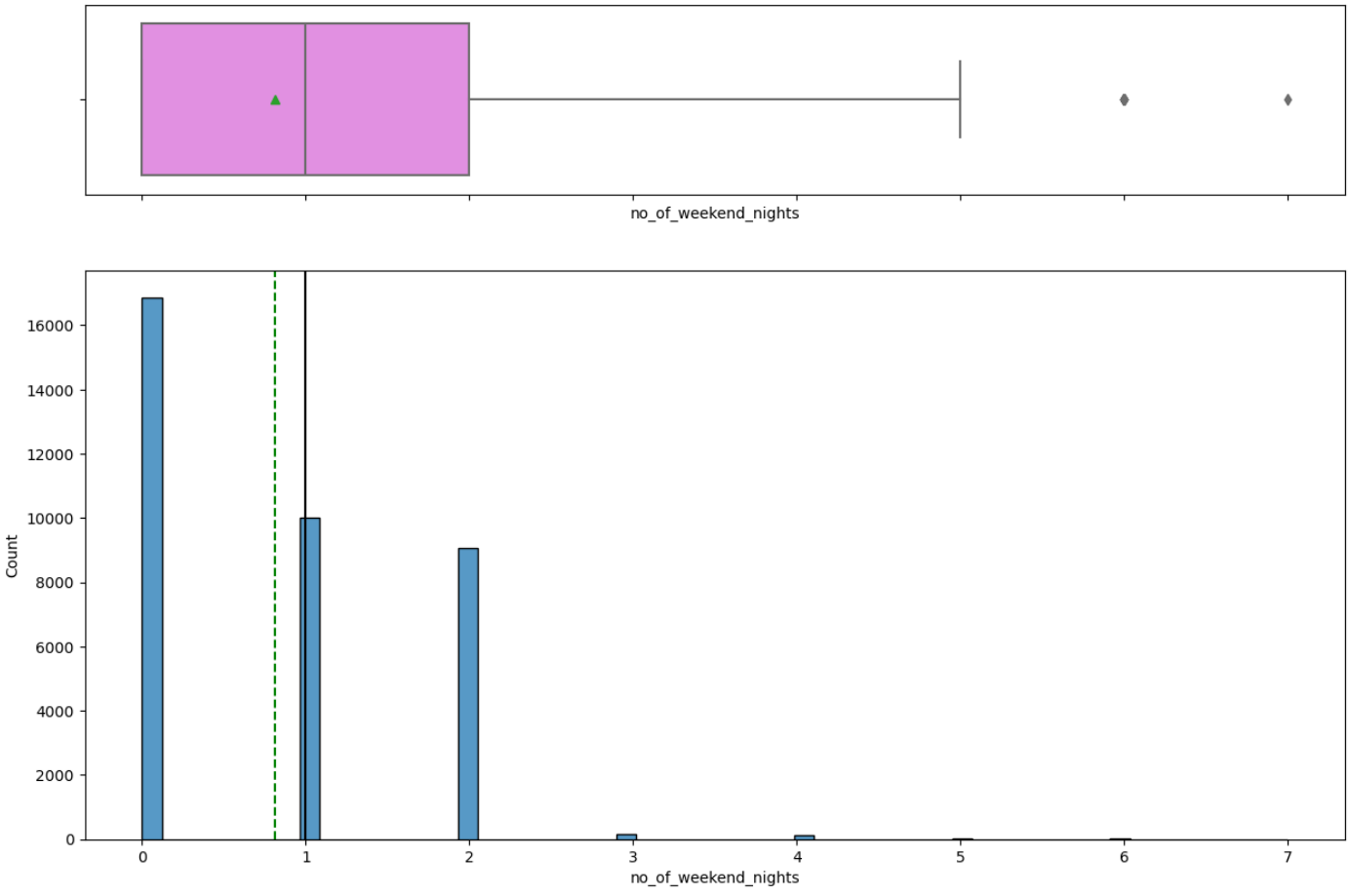


Figure-5 : Histo-box plot of no_of_weekend_nights

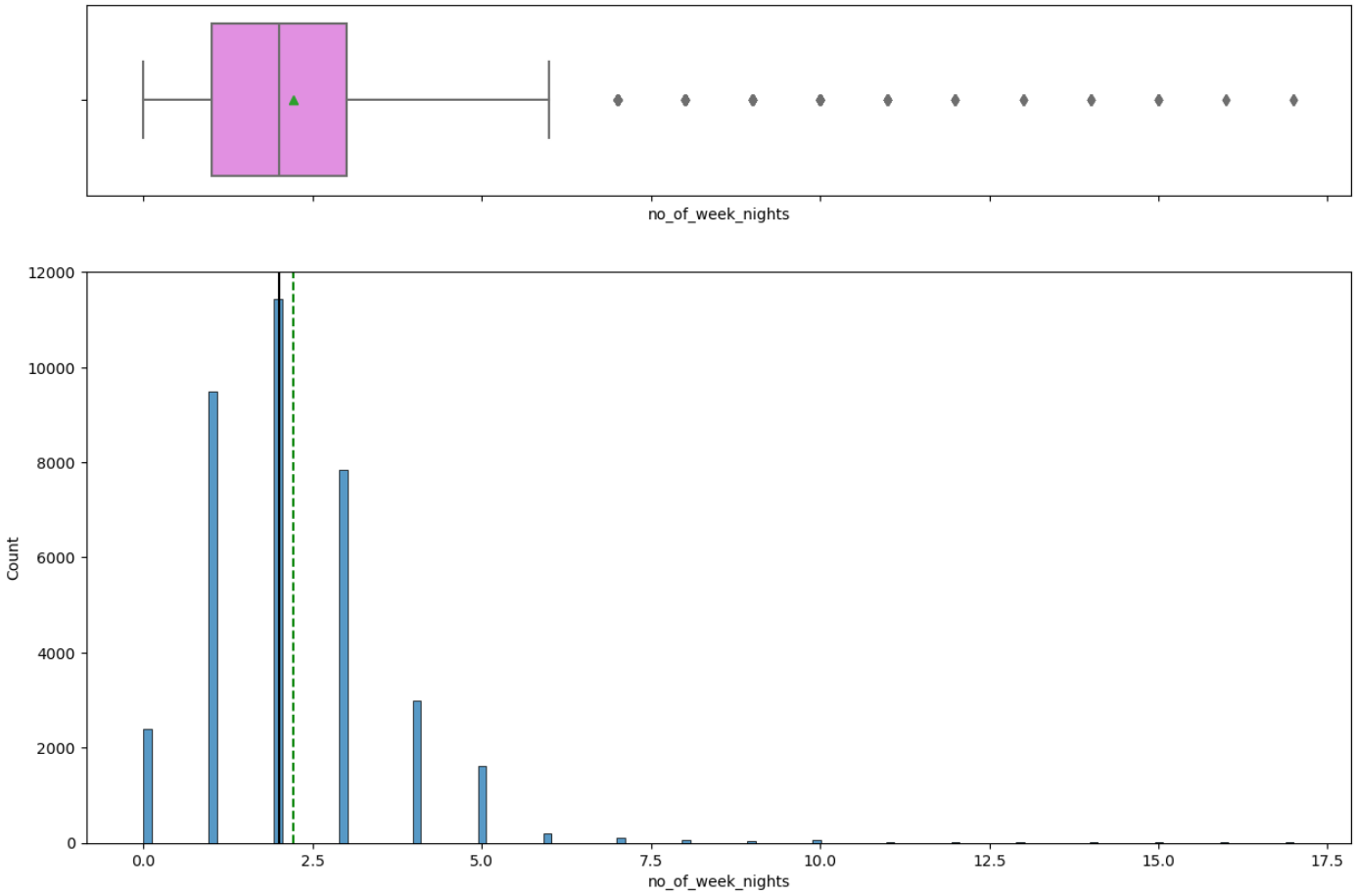


Figure-6 : Histo-box plot of no_of_week_nights

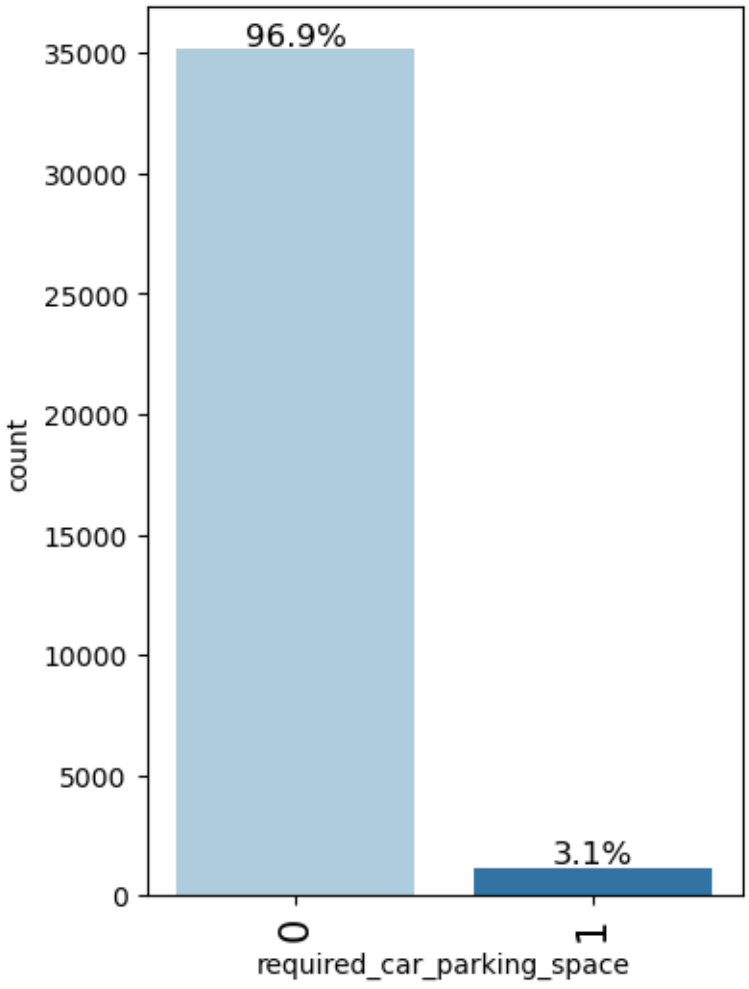


Figure-7 : Bar plot req. car-parking space

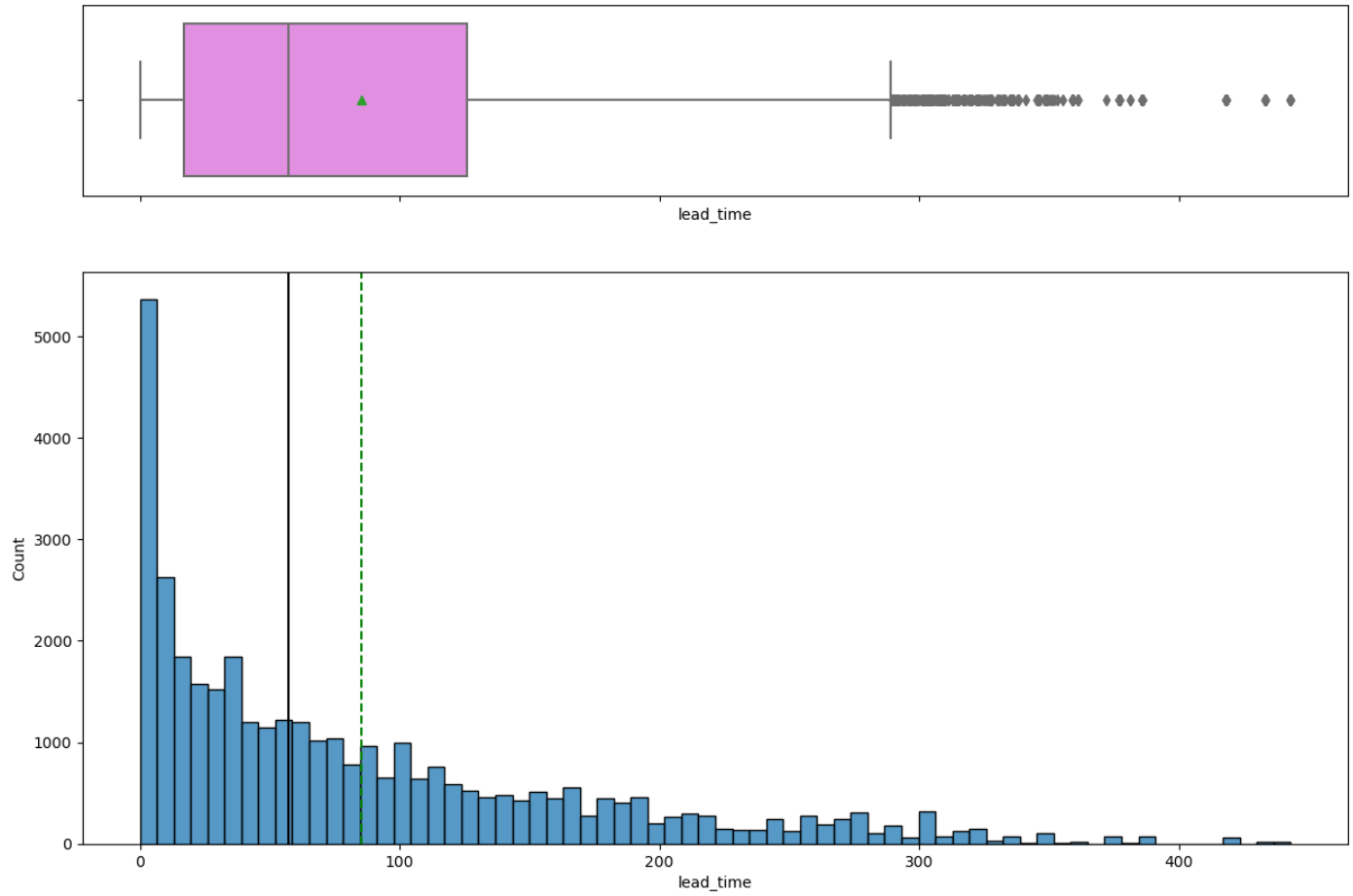


Figure-8 : Histo-box plot of lead_time

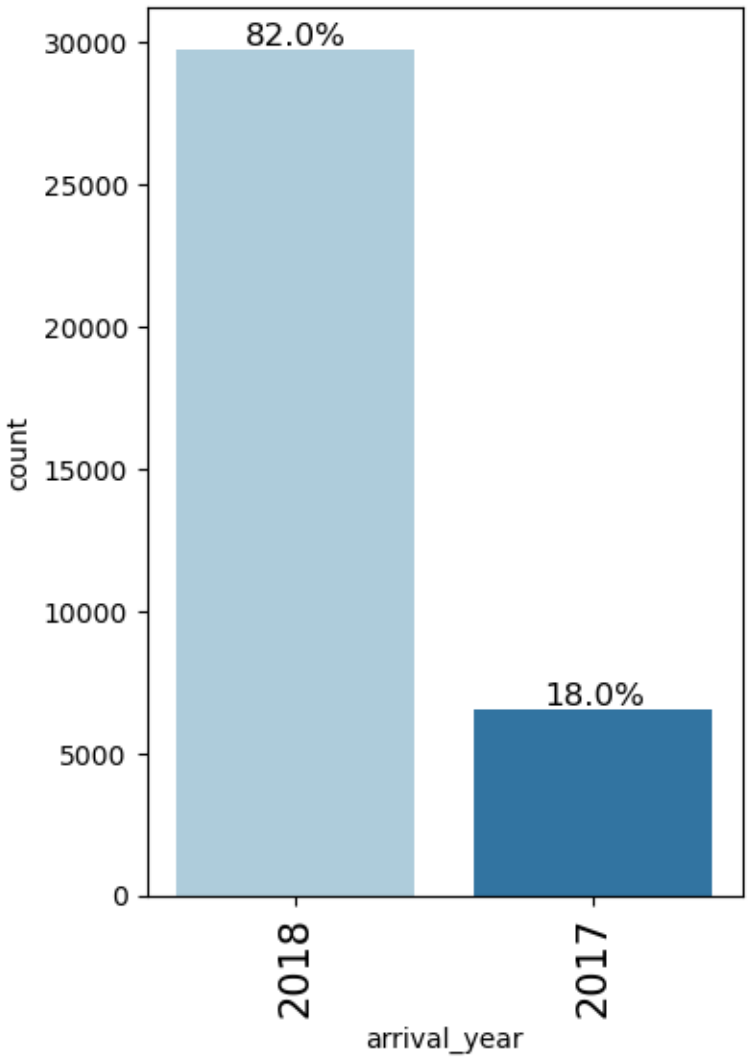


Figure-9 : Bar plot Arrival year

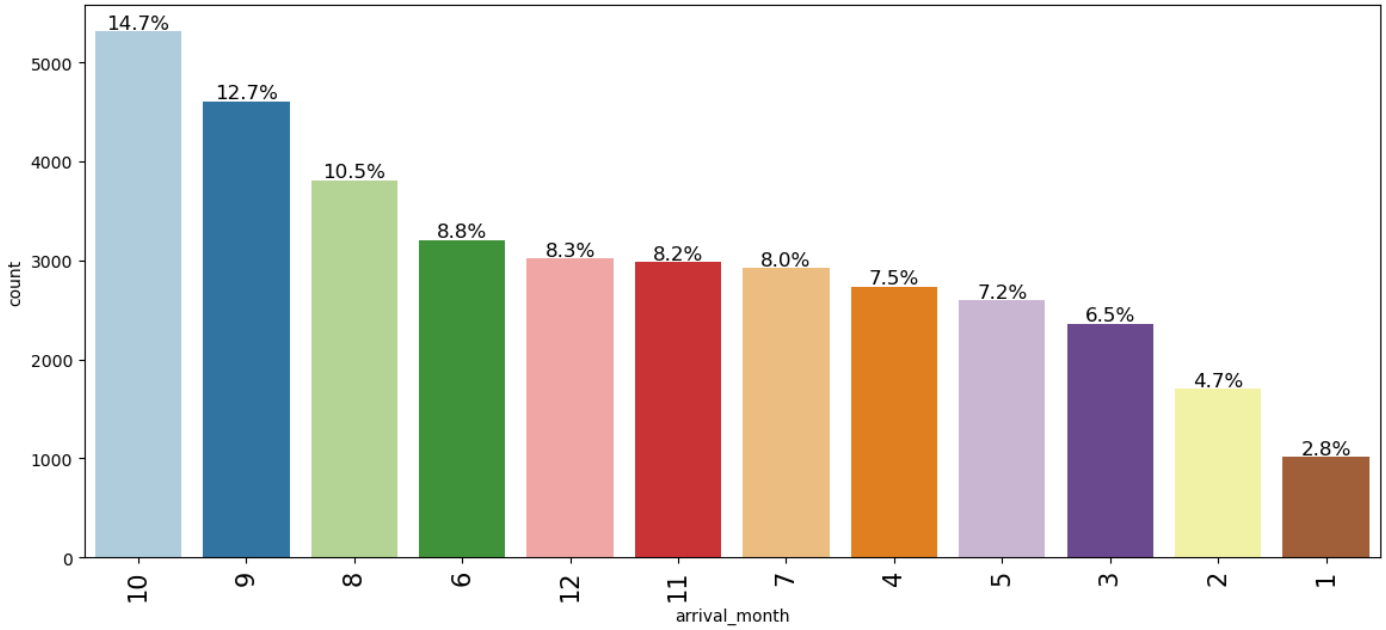


Figure-10 : Bar plot Arrival month

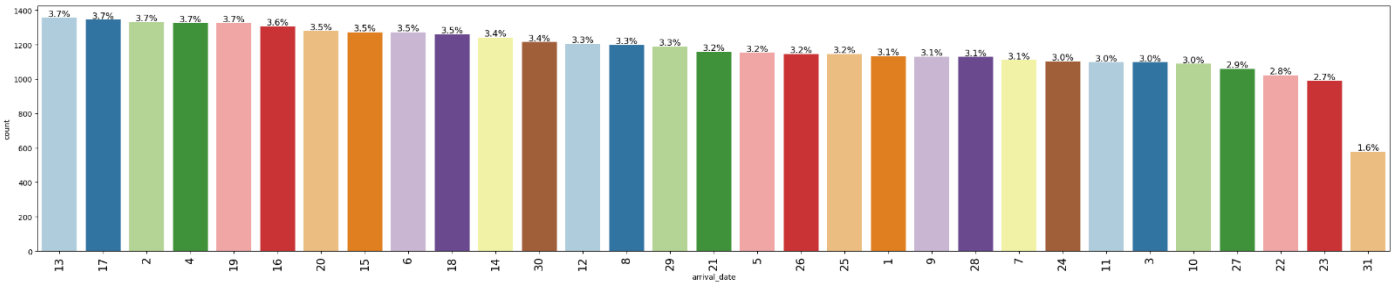


Figure-11 : Bar plot Arrival date

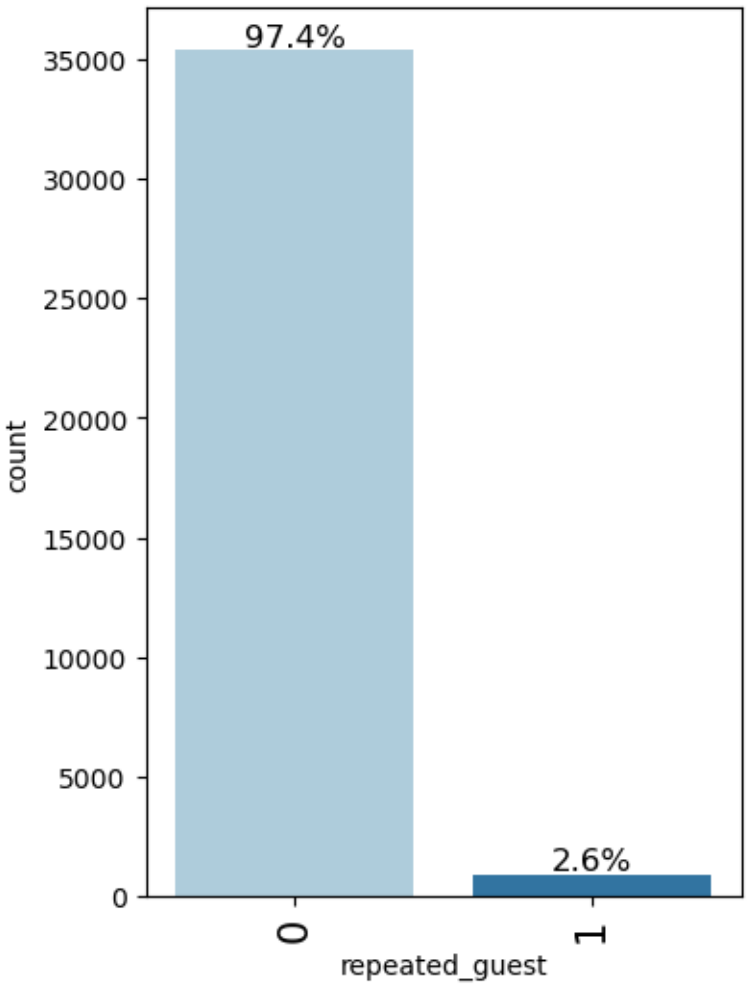


Figure-12 : Bar plot of Repeated guest

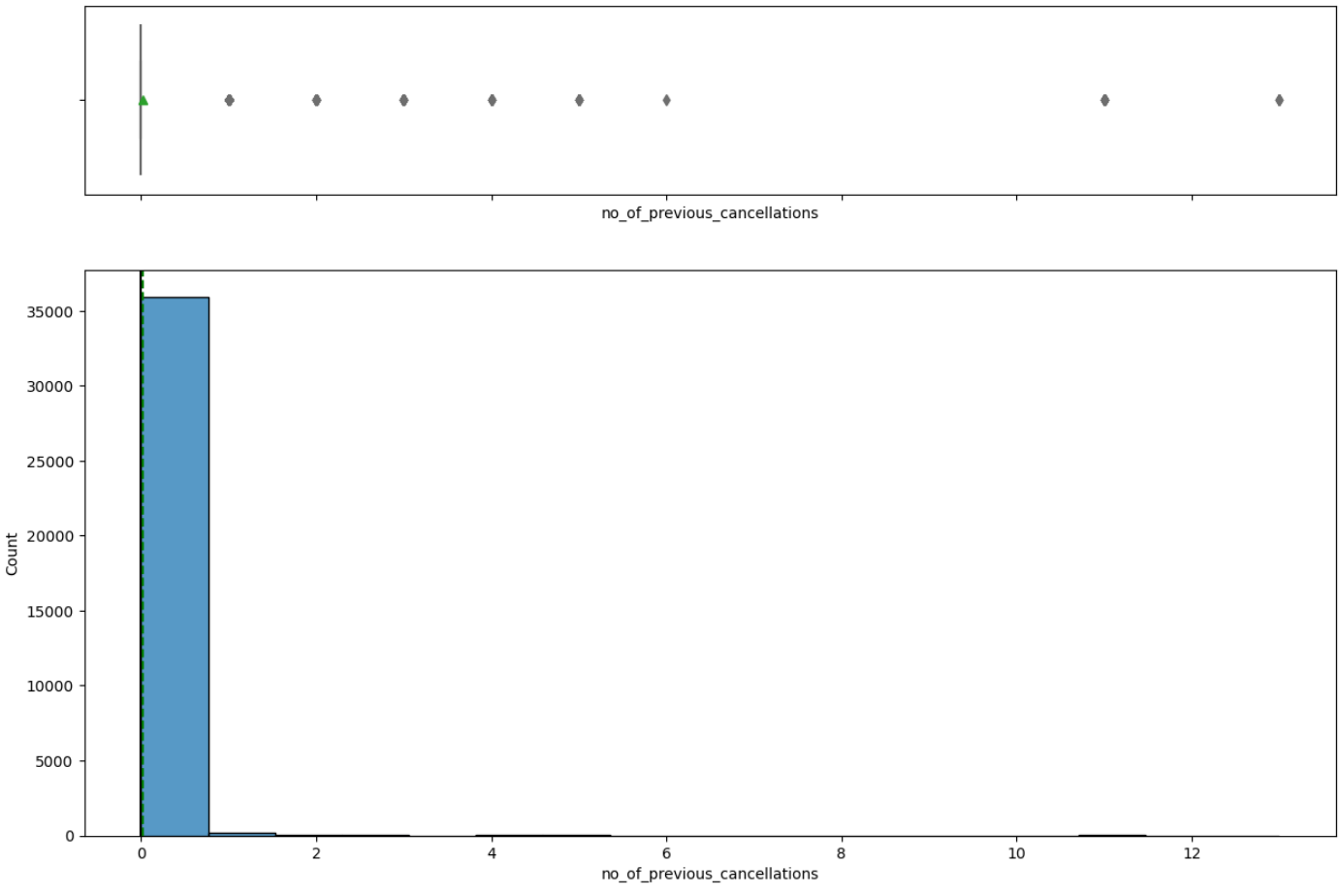


Figure-13 : Histo-box plot of no_of_cancellation

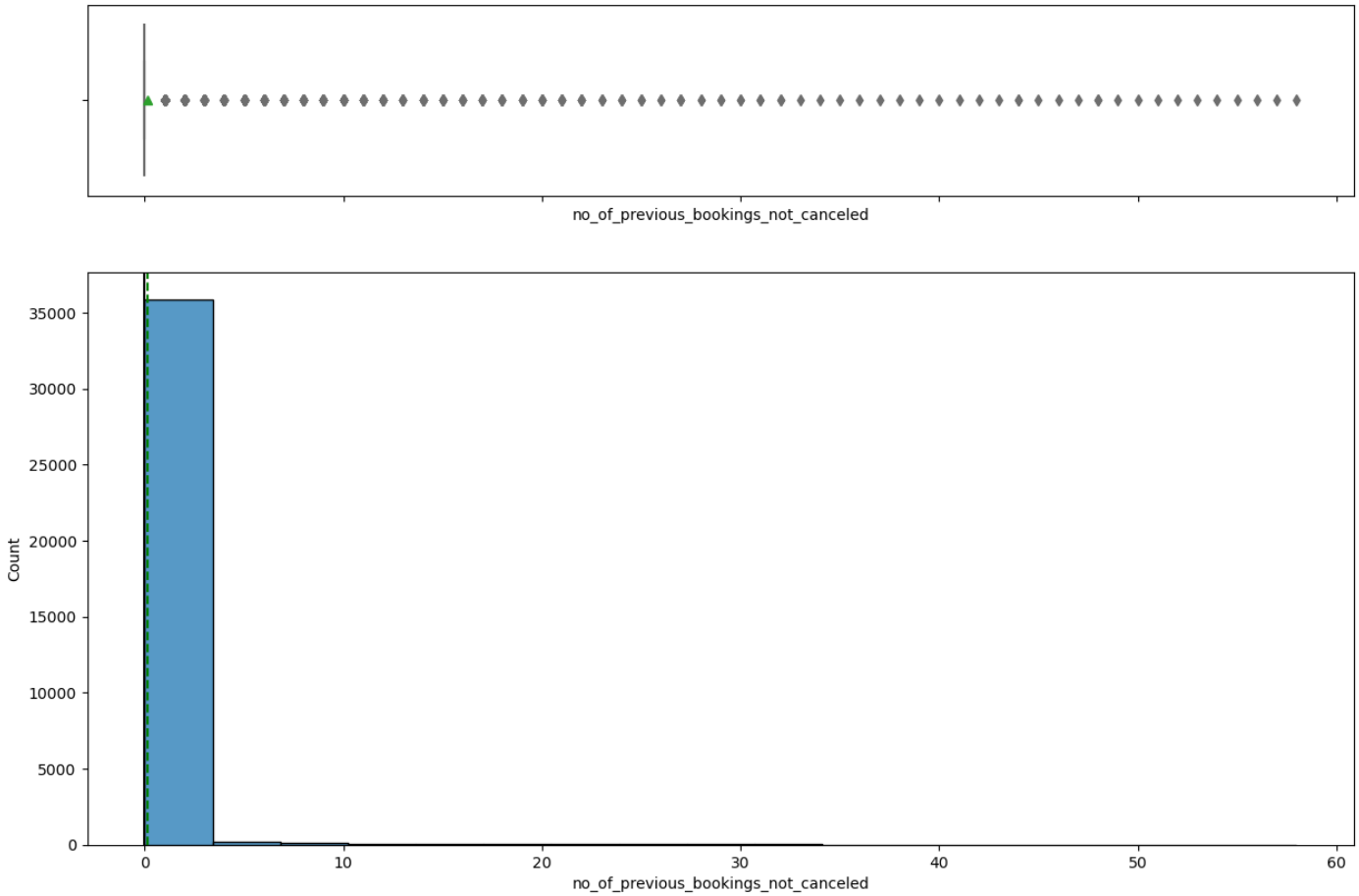


Figure-14 : Histo-box plot of no_of_bookings_not_canceled

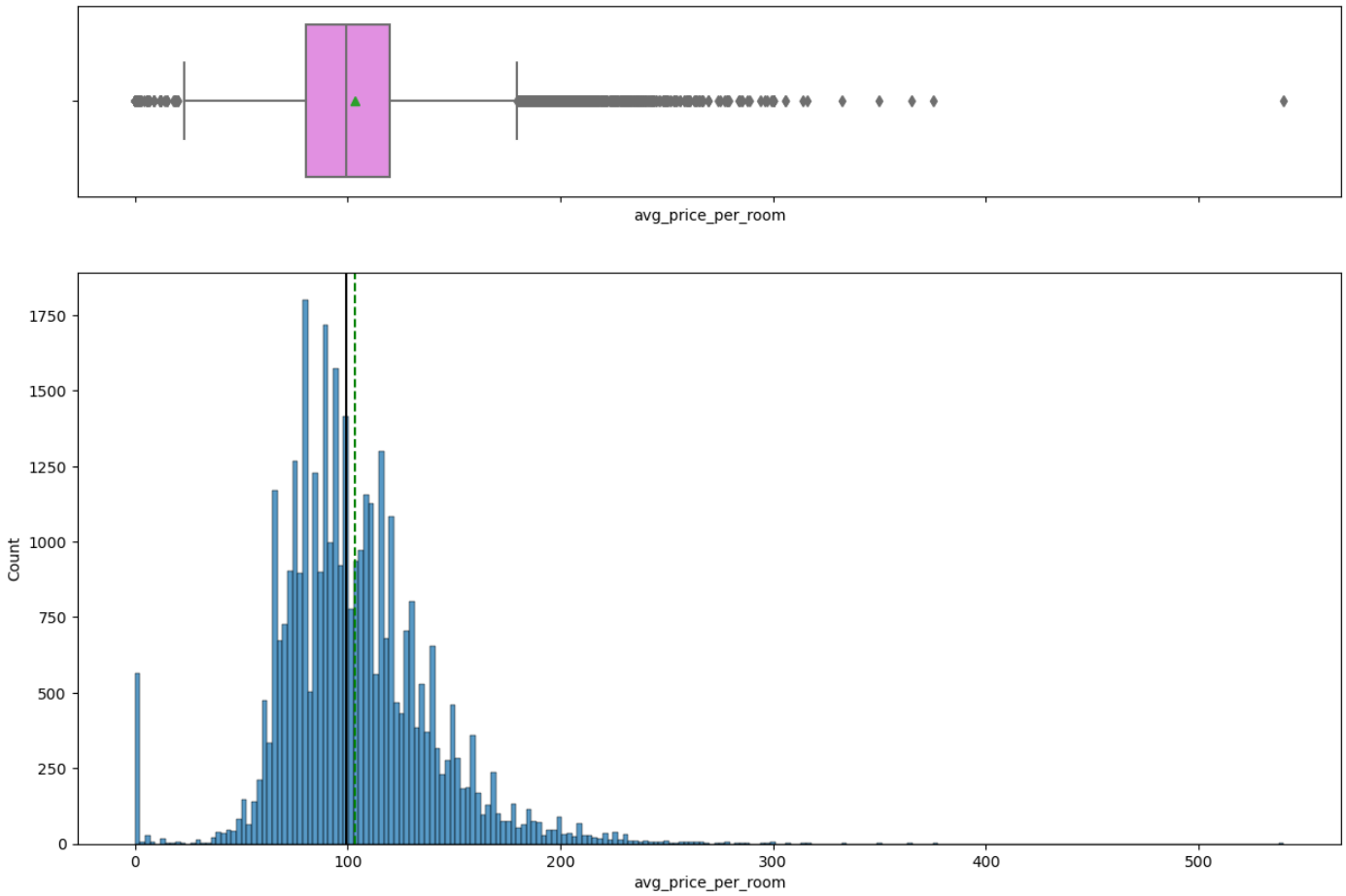


Figure-15 : Histo-box plot of avg. price per room

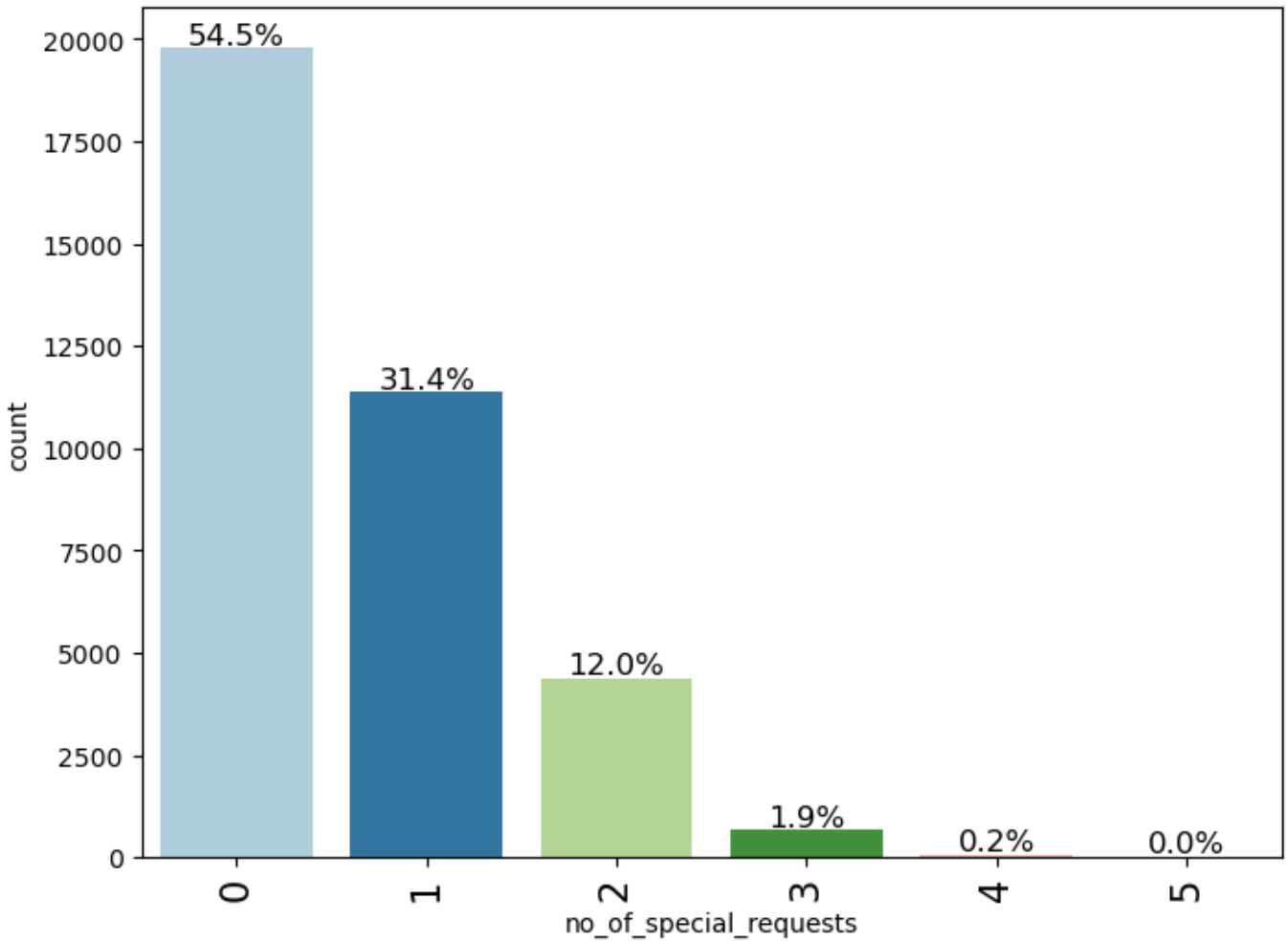


Figure-16 : Bar plot of no_of_special_requests

Bivariate Analysis:

Correlation Check-

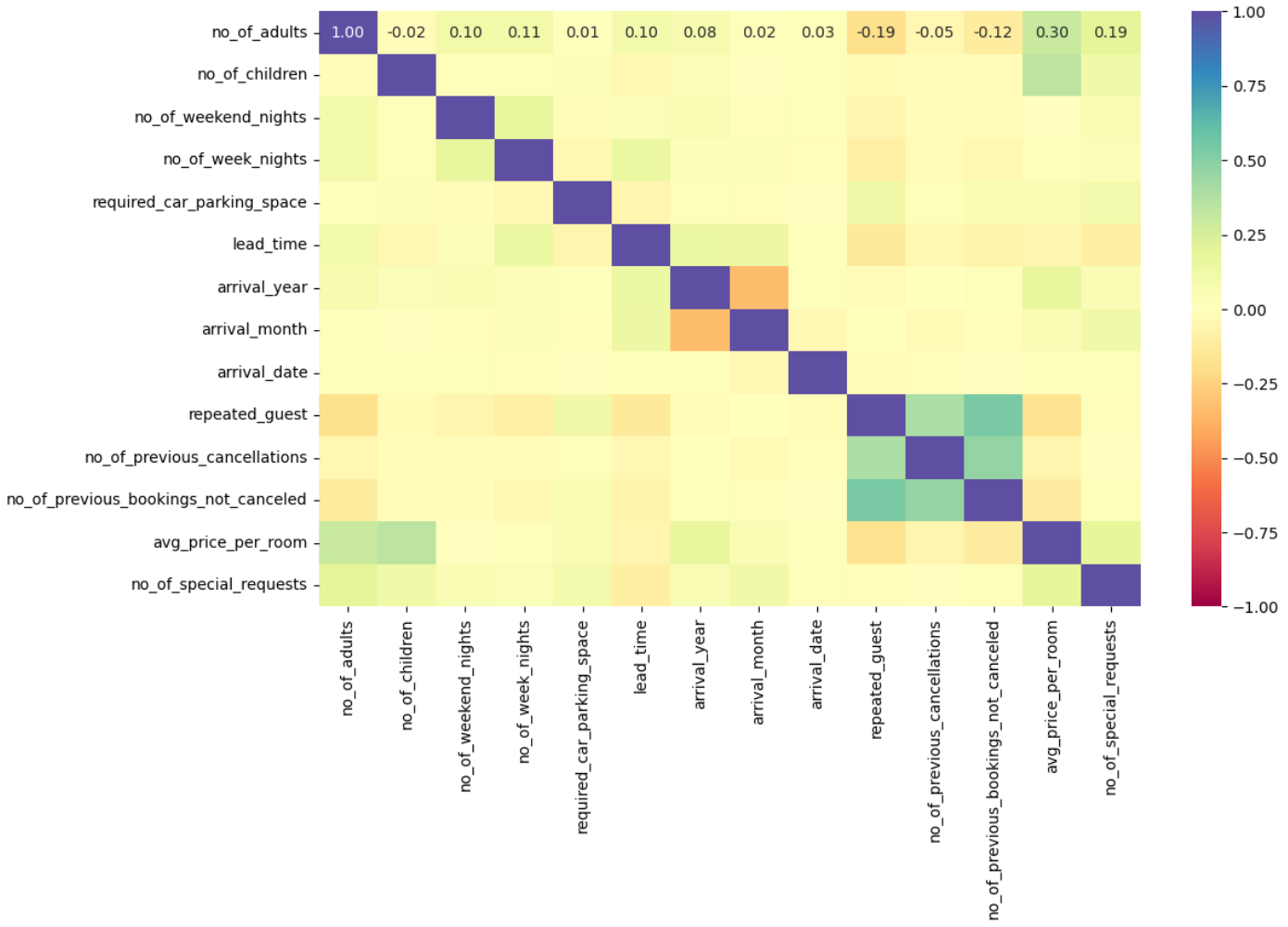


Figure-17 : Heat map of numeric variables

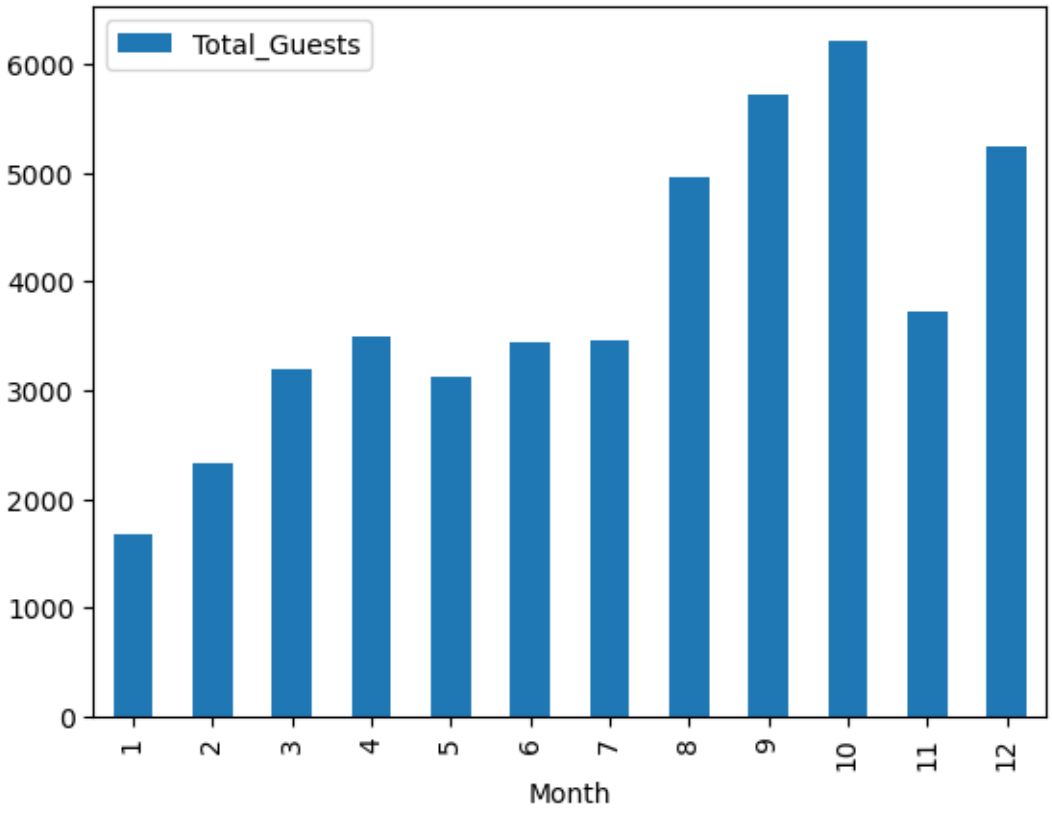


Figure-18 : Total guests on month wise

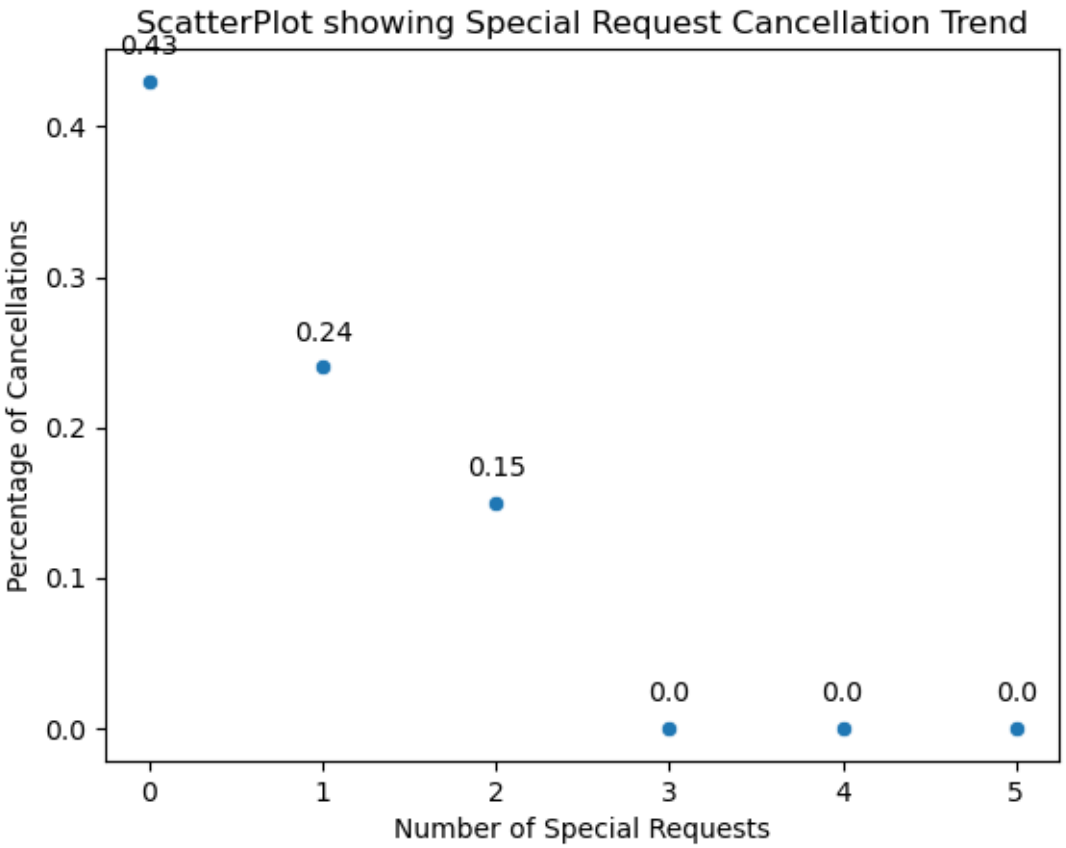


Figure-19 : Scatter plot monthly cancellation trend

EDA Questions:

1. What are the busiest months in the hotel?

Ans: We can see that the month of October is the busiest month in the hotel, then September then August.

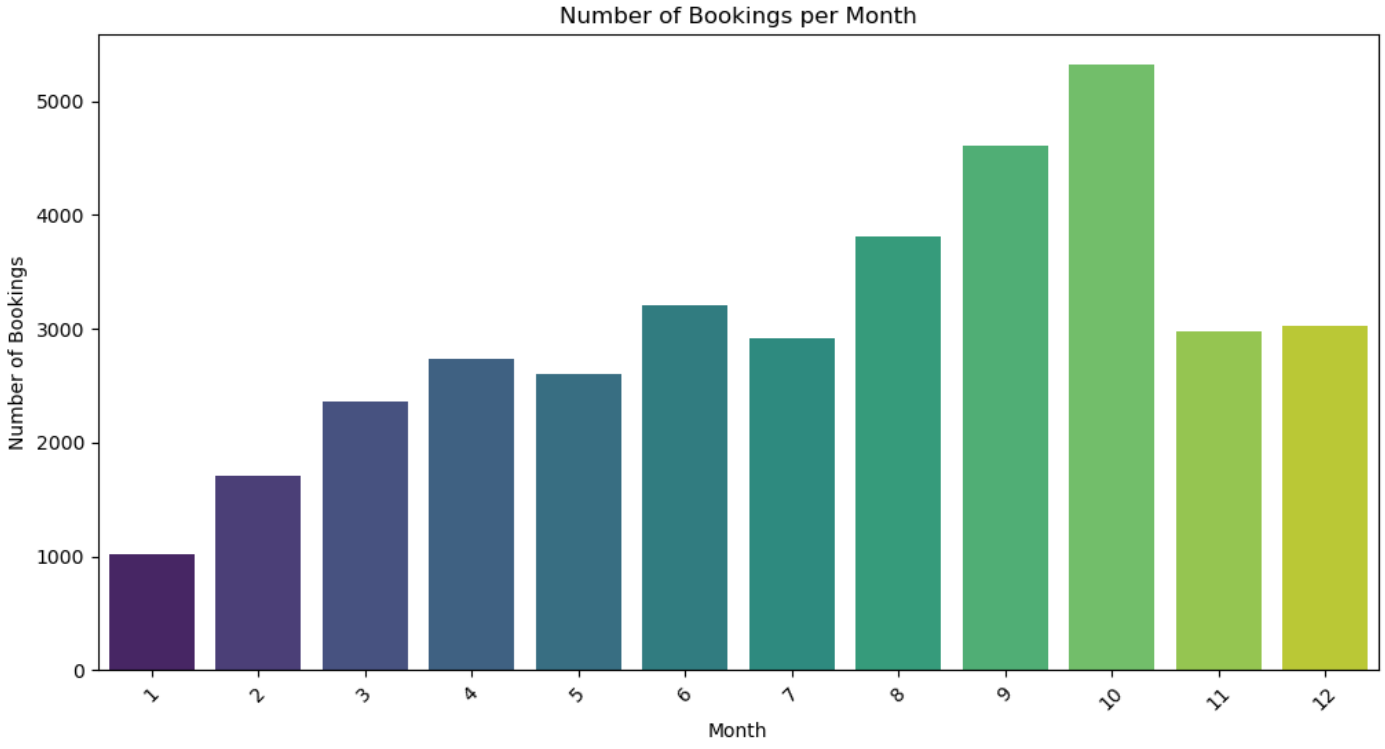


Figure-20 : Bookings month wise

2. Which market segment do most of the guests come from?

Ans:

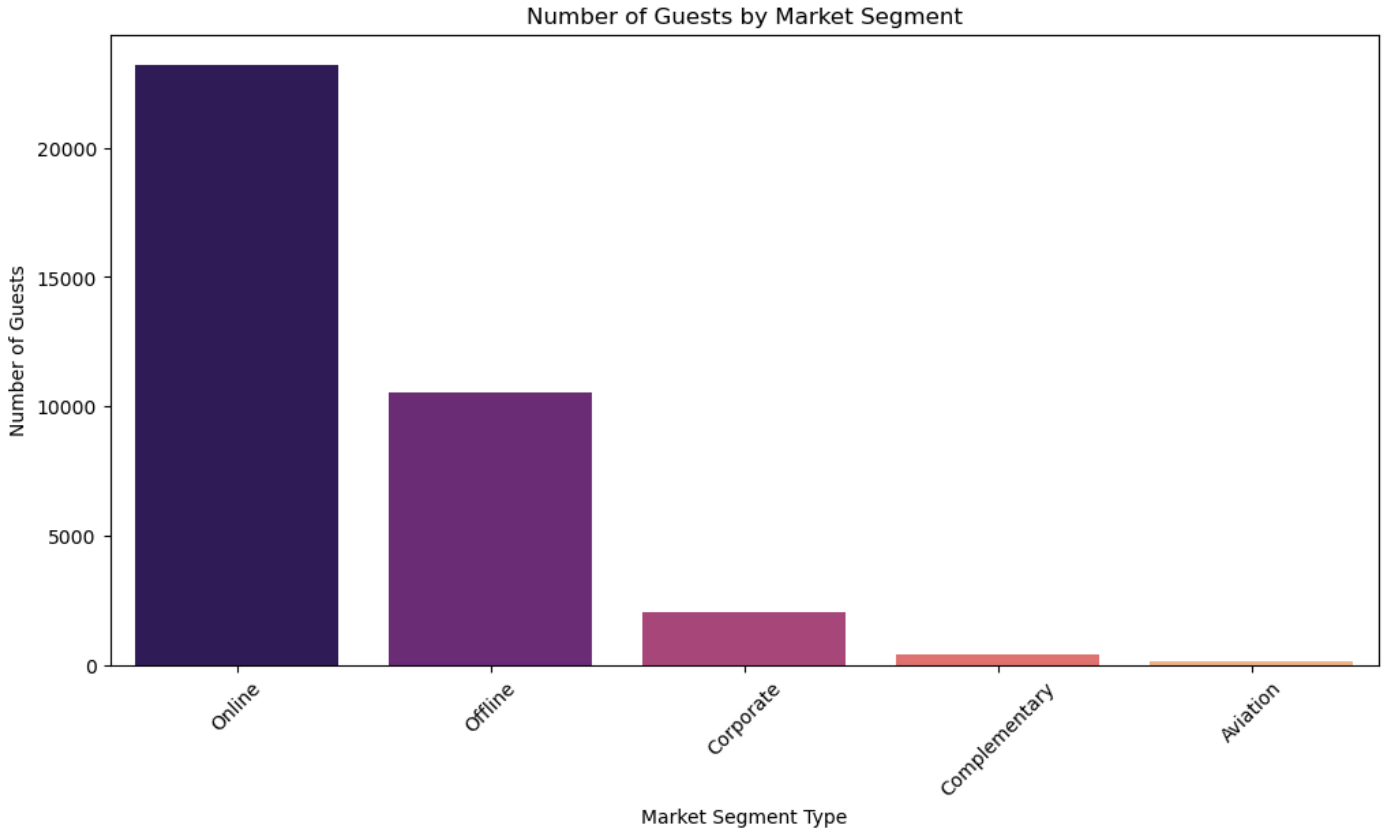


Figure-21 : Bookings from segments

	market_segment_type	count
0	Online	23214
1	Offline	10528
2	Corporate	2017
3	Complementary	391
4	Aviation	125

- Most of the guests come from online market segment.

3. Hotel rates are dynamic and change according to demand and customer demographics. What are the differences in room prices in different market segments?

Ans:

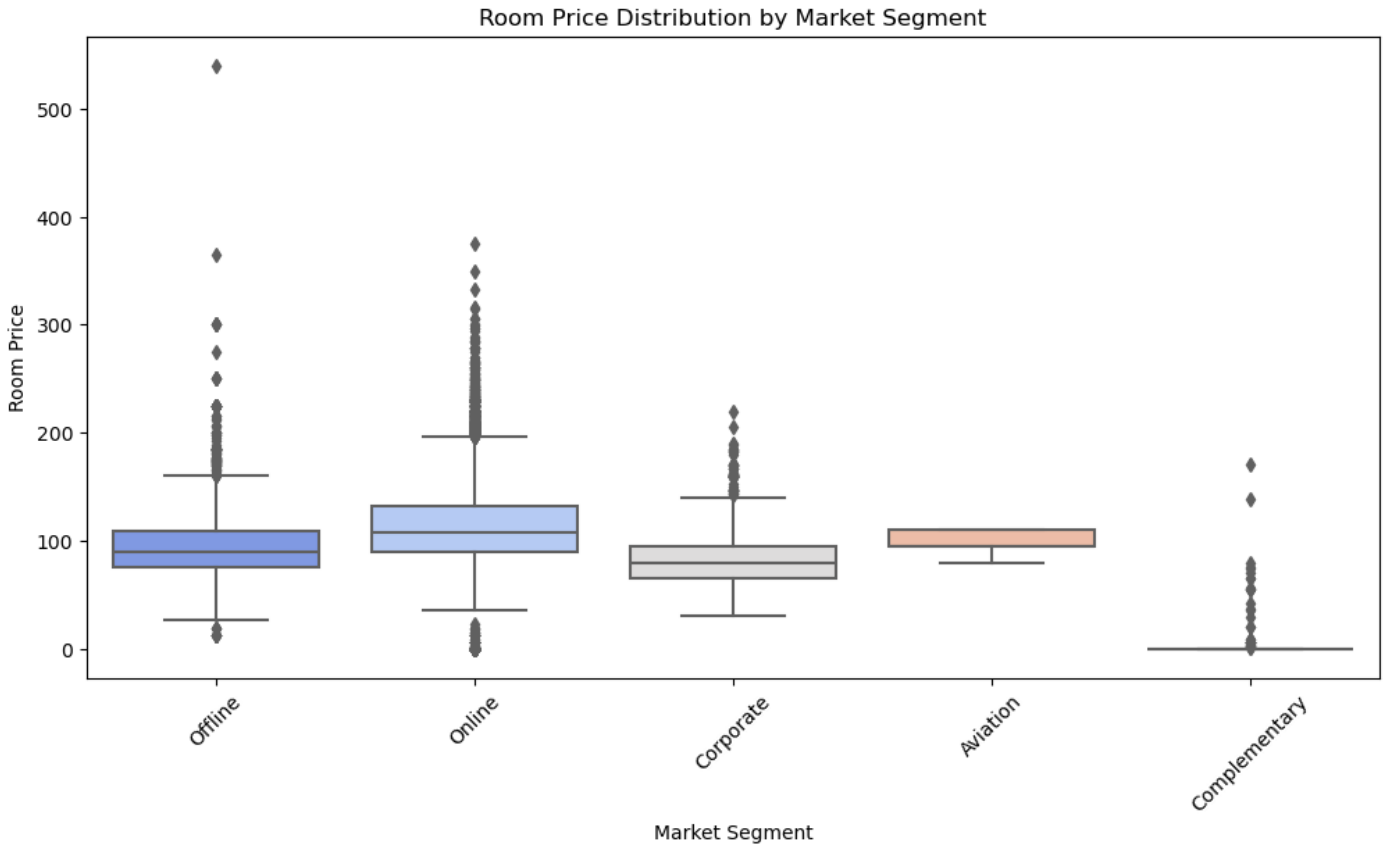


Figure-22 : Room price distribution market segment wise

The bookings by online market segment are having highest price for the rooms, then comes offline segment bookings and then aviation.

4. What percentage of bookings are canceled?

Ans:

```
Percentage of bookings that are canceled: 32.76%
```

5. Repeating guests are the guests who stay in the hotel often and are important to brand equity. What percentage of repeating guests cancel?

Ans:

```
1.7204301075268817
```

The percentage of repeating guests cancel booking is 1.72 %

6. Many guests have special requirements when booking a hotel room. Do these requirements affect booking cancellation?

Ans:

```
(20.24487816705055, 43.2067553218385)
```

The analysis shows the following cancellation rates:

- Guests with special requests: 20.24%
- Guests without special requests: 43.21%

This indicates that guests without special requests have a significantly higher cancellation rate compared to those with special requests. Thus, special requirements appear to affect booking cancellations, with guests who have special requests being less likely to cancel their bookings.

8. Data Preprocessing:

Missing Values:

```
no_of_adults                          0
no_of_children                        0
no_of_weekend_nights                  0
no_of_week_nights                     0
type_of_meal_plan                     0
required_car_parking_space            0
room_type_reserved                    0
lead_time                             0
arrival_year                          0
arrival_month                         0
arrival_date                          0
market_segment_type                   0
repeated_guest                        0
no_of_previous_cancellations          0
no_of_previous_bookings_not_canceled  0
avg_price_per_room                    0
no_of_special_requests                0
booking_status                        0
dtype: int64
```

There are no missing values in the data set.

Checking for outliers:

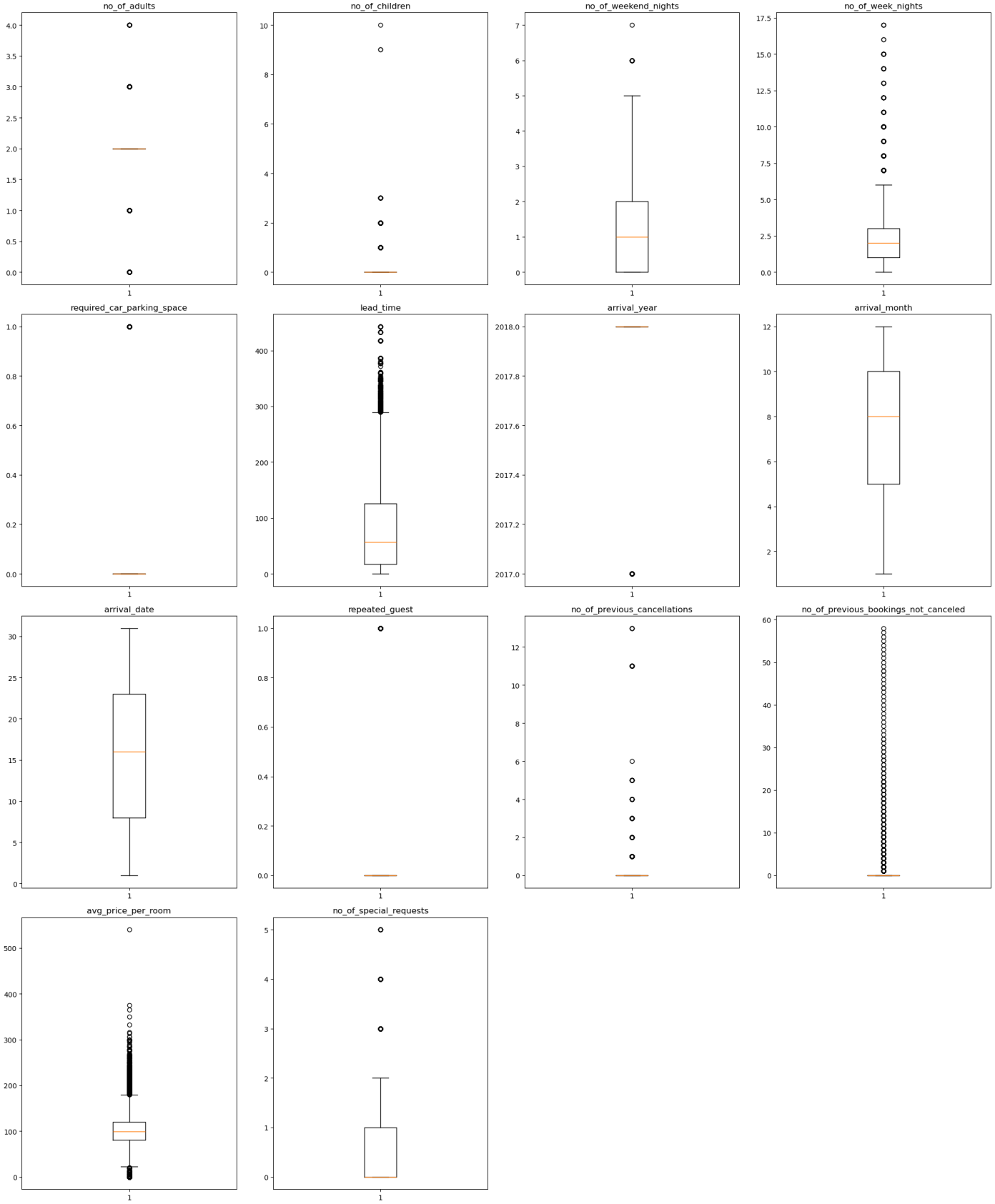


Figure-23 : Box plots for outliers

Observations:

While checking for outliers, I decided that these outliers for necessary for analysis as the data points could not be easily jusitified removal or scaling.

Feature Engineering:

Value Count for no_of_adults

no_of_adults

2 26091

1 7676

3 2316

0 139

4 16

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for no_of_children

no_of_children

0 33544

1 1616

2 1056

3 19

9 2

10 1

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for no_of_weekend_nights

no_of_weekend_nights

0 16872

1 9959

2 9071

3 152

4 129

5 34

6 20

7 1

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for no_of_week_nights

no_of_week_nights

2 11433

1 9481

3 7829

4 2989

0 2383

5 1611

6 189

7 112

10 62

8 62

9 34

11 17

15 10

12 9

14 7

13 5

17 3

16 2

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for required_car_parking_space

required_car_parking_space

0 35117

1 1121

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for lead_time

lead_time

0 1295

1 1077

2 643

4 627

3 626

...

300 1

353 1

328 1

352 1

351 1

Name: count, Length: 352, dtype: int64

To verify total counts 36238

****

Value Count for arrival_year

arrival_year

2018 29724

2017 6514

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for arrival_month

arrival_month

10 5317

9 4611

8 3813

6 3203

12 3021

11 2980

7 2920

4 2736

5 2598

3 2358

2 1667

1 1014

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for repeated_guest

repeated_guest

0 35312

1 926

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for no_of_previous_cancellations

no_of_previous_cancellations

0 35901

1 197

2 46

3 43

11 25

5 11

4 10

13 4

6 1

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for no_of_previous_bookings_not_canceled

no_of_previous_bookings_not_canceled

0 35429

1 227

2 112

3 80

4 65

5 60

6 36

7 24

8 23

9 19

10 18

11 14

12 12

14 9

15 8

16 7

13 7

18	6
20	6
21	6
17	6
19	6
22	6
25	3
27	3
24	3
23	3
44	2
29	2
48	2
28	2
30	2
32	2
31	2
26	2
46	1
55	1
45	1
57	1
53	1
54	1
58	1
41	1
40	1
43	1
35	1

50 1

56 1

33 1

37 1

42 1

51 1

38 1

34 1

39 1

52 1

49 1

47 1

36 1

Name: count, dtype: int64

To verify total counts 36238

****

Value Count for no_of_special_requests

no_of_special_requests

0 19751

1 11363

2 4363

3 675

4 78

5 8

Name: count, dtype: int64

To verify total counts 36238

****

Observations: There are m Many categorical types for int64, move forward with grouping

Table 7: Category wise value count

Train & Test data split:

```
Number of rows in train data = 25366
Number of rows in test data = 10872
Percentage of classes in training set:
booking_status_encoded
1   0.67074
0   0.32926
Name: proportion, dtype: float64
Percentage of classes in test set:
booking_status_encoded
1   0.67568
0   0.32432
Name: proportion, dtype: float64
```

- We had seen that around 67% of observations belongs to class 0 (Not cancelled) and 33% observations belongs to class 1 (cancelled), and this is preserved in the train and test sets

Table 8: Splitting data in 70:30 ratio

9. Model Building:

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=1)
```

Table 9: Fitting a decision tree model using the gini critera

Defining a function to compute different metrics to check performance of a classification model built using sklearn.

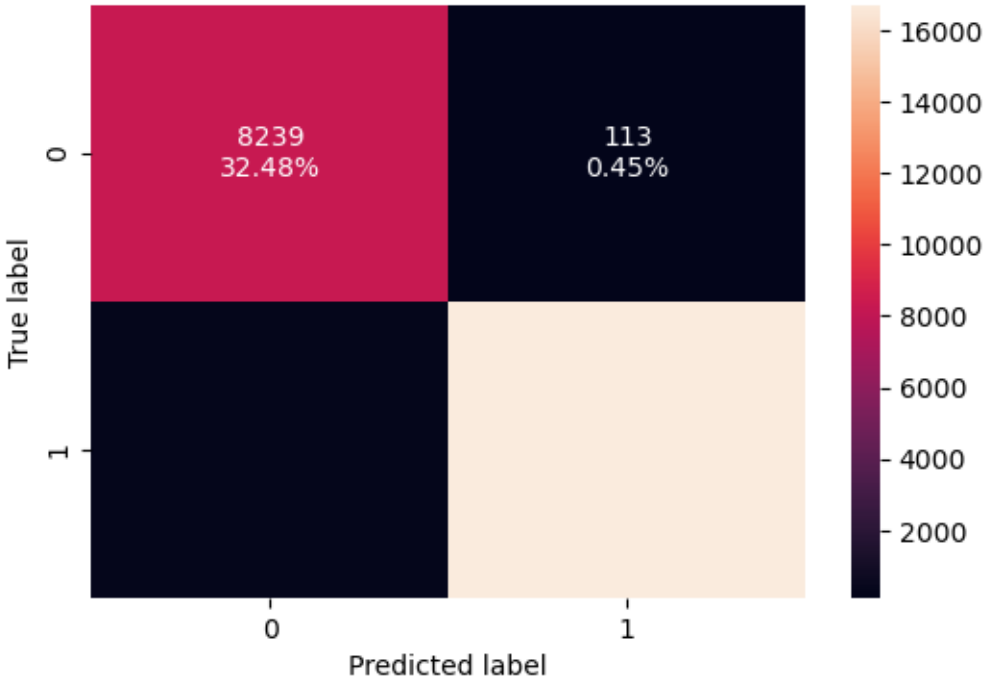


Figure-24 : Confusion matrix for X-train, y-train

Training performance:

	Accuracy	Recall	Precision	F1
0	0.98403	0.98284	0.99329	0.98804

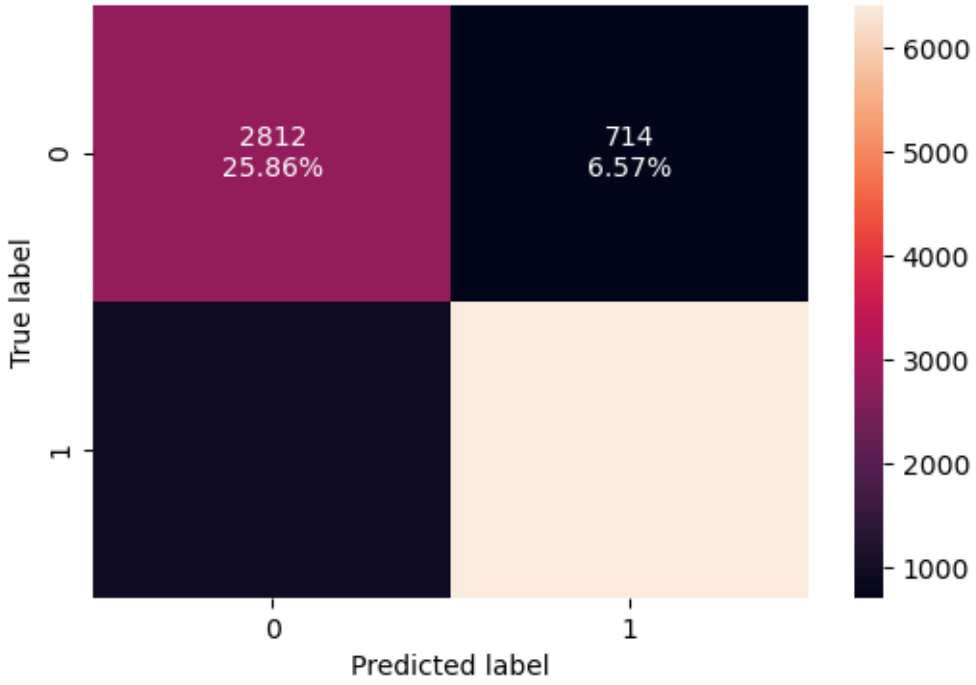


Figure-25 : Confusion matrix for X-test, y-test

Test performance:

	Accuracy	Recall	Precision	F1
0	0.84842	0.87286	0.89980	0.88612

Decision tree classifier balanced weight:

```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', random_state=1)
```

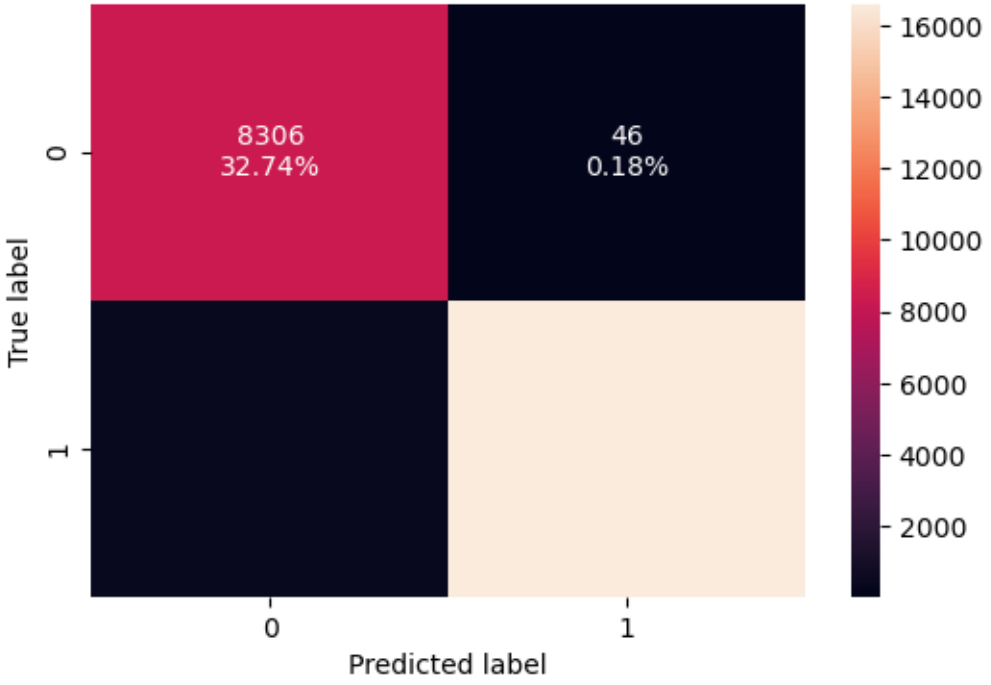


Figure-26 : Balanced Confusion matrix for X-train, y-train

Training performance: Balanced weighted

	Accuracy	Recall	Precision	F1
0	0.98179	0.97555	0.99724	0.98627

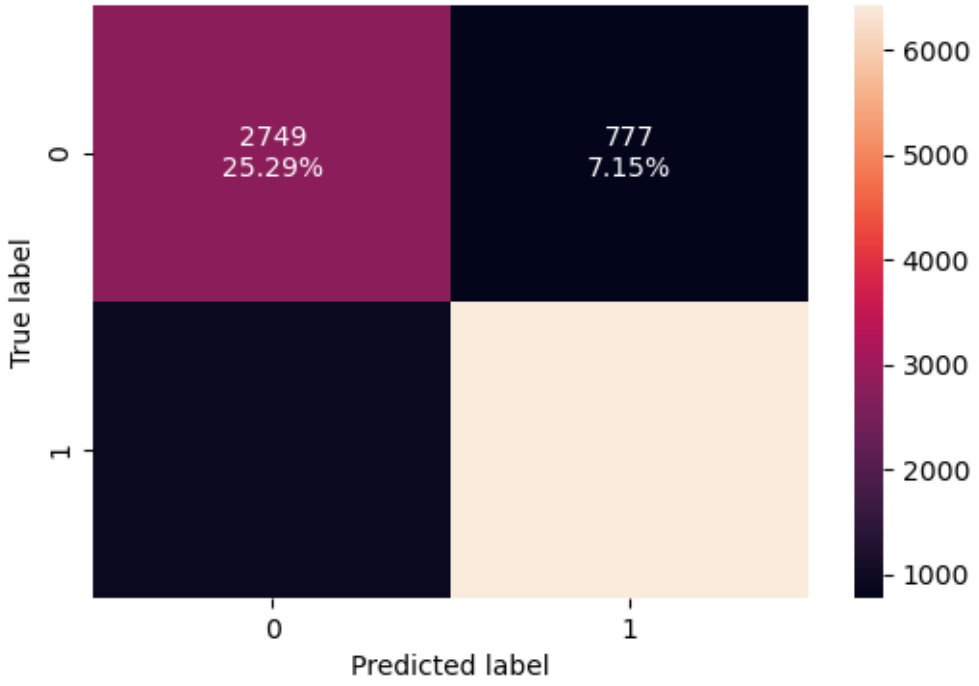


Figure-27 : Balanced Confusion matrix for X-test, y-test

Test performance: Balance weighted

	Accuracy	Recall	Precision	F1
0	0.84437	0.87544	0.89220	0.88374

Inference:

Observations:

As expected, the training data 's f1 score is very close to 1 while the testing data is nearing 90. While the model is capturing a good amount of test data, we can assume without tuning this model is overfitting.

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4, max_leaf_nodes=50, min_samples_split=70,
                       random_state=1)
```

Table 10: Decision tree classifier estimator

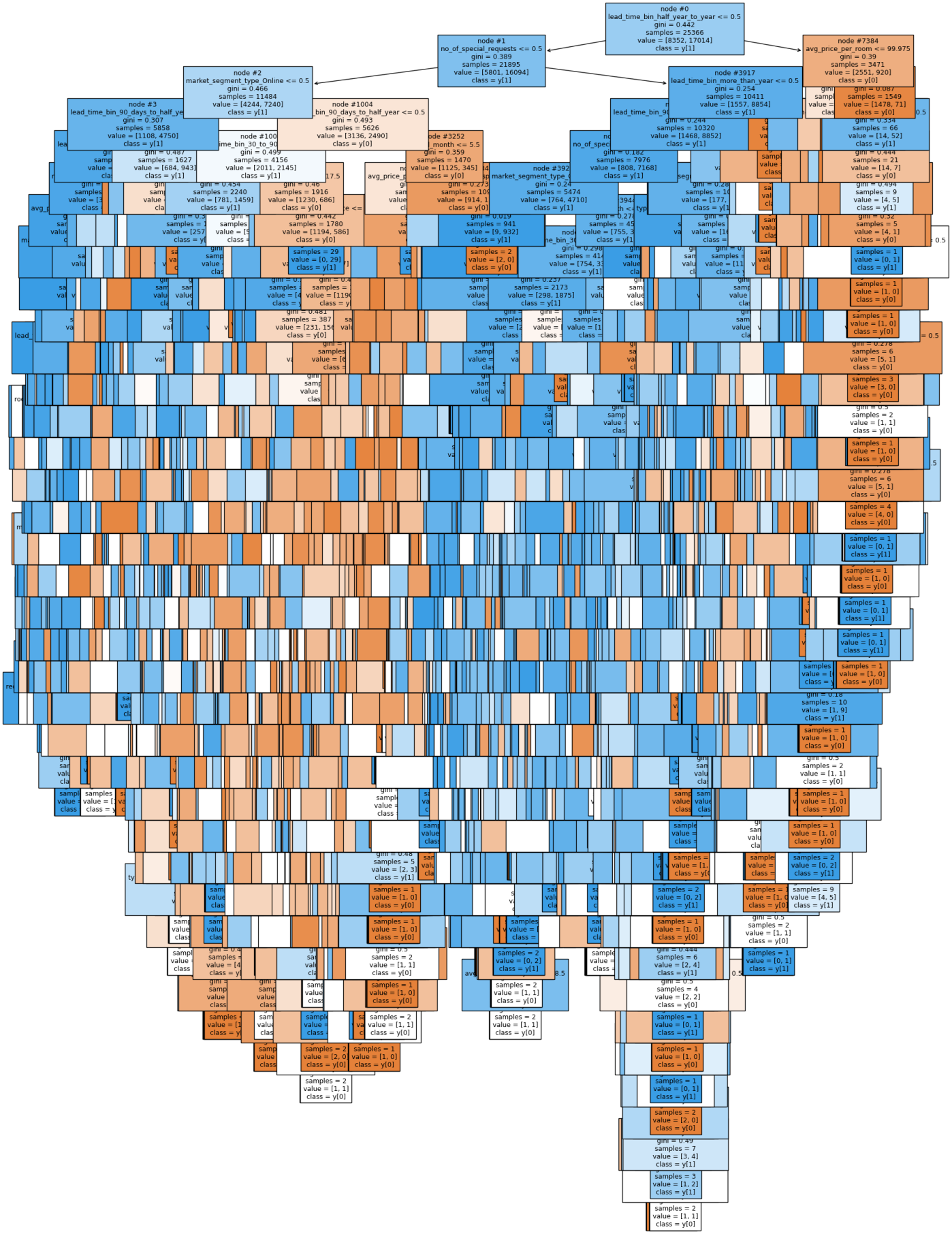


Figure-28 : Decision tree

Observation:

Due to the overfitting, the decision tree branchs are very complex. Should look at limiting depth and post pruning

Gather the model's important variables and sort by importance:

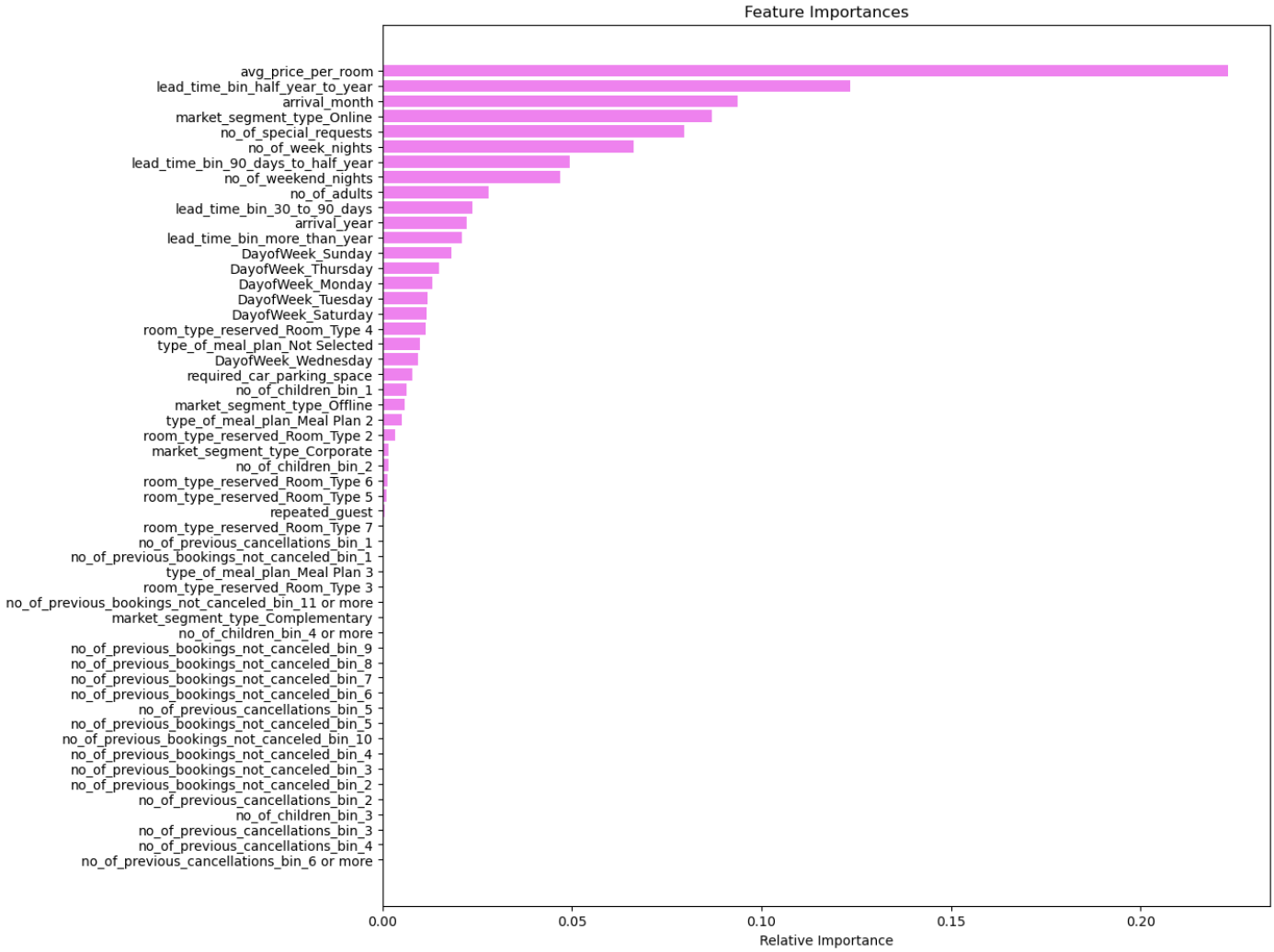


Figure-29 : Importance of datas

10. Model performance improvement:

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=15,
                       min_impurity_decrease=1e-06, random_state=1)
```

Table 11: Hyperparameter tuning

KNN classifier accuracy

```
KNN Classifier Accuracy: 0.8154893303899926
```

Naive bayes

```
Naive-Bayes Classifier Accuracy: 0.3589036055923473
```

Decision tree accuracy

```
Decision Tree Classifier Accuracy: 0.8513612950699043
```

Post tuning:

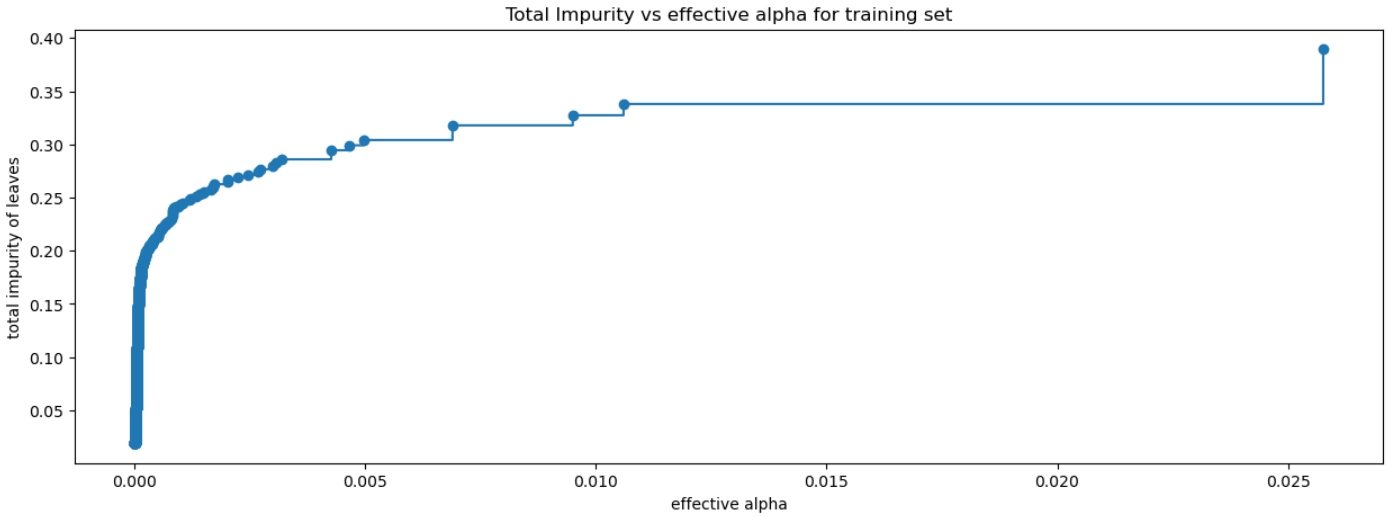


Figure-30 : Post-tuning impurity vs alfa training set

Defining nodes:

```
Number of nodes in the last tree is: 1 with ccp_alpha: 0.05218215082671268
```

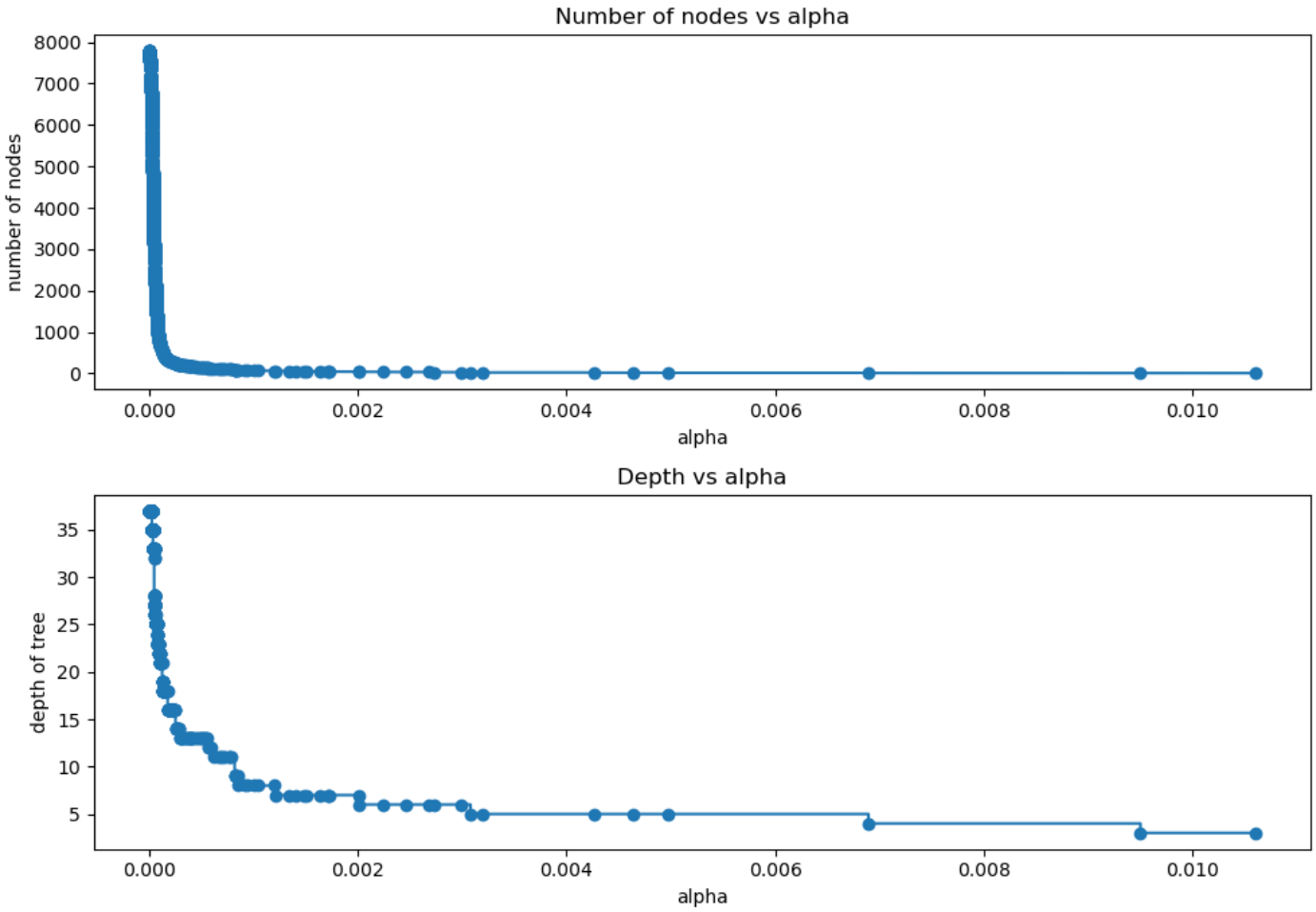


Figure-31 : Post-tuning no. of nodes & depth vs alfa training

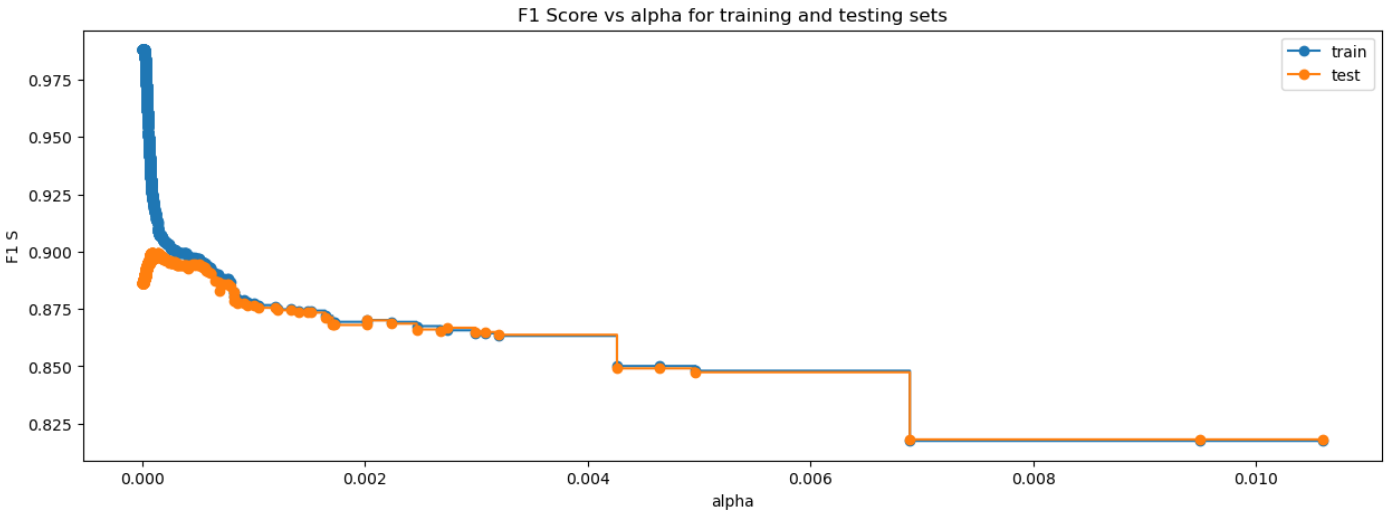


Figure-32 : F1 score vs alfa for training and test data

creating the model where we get highest train and test f1_score:

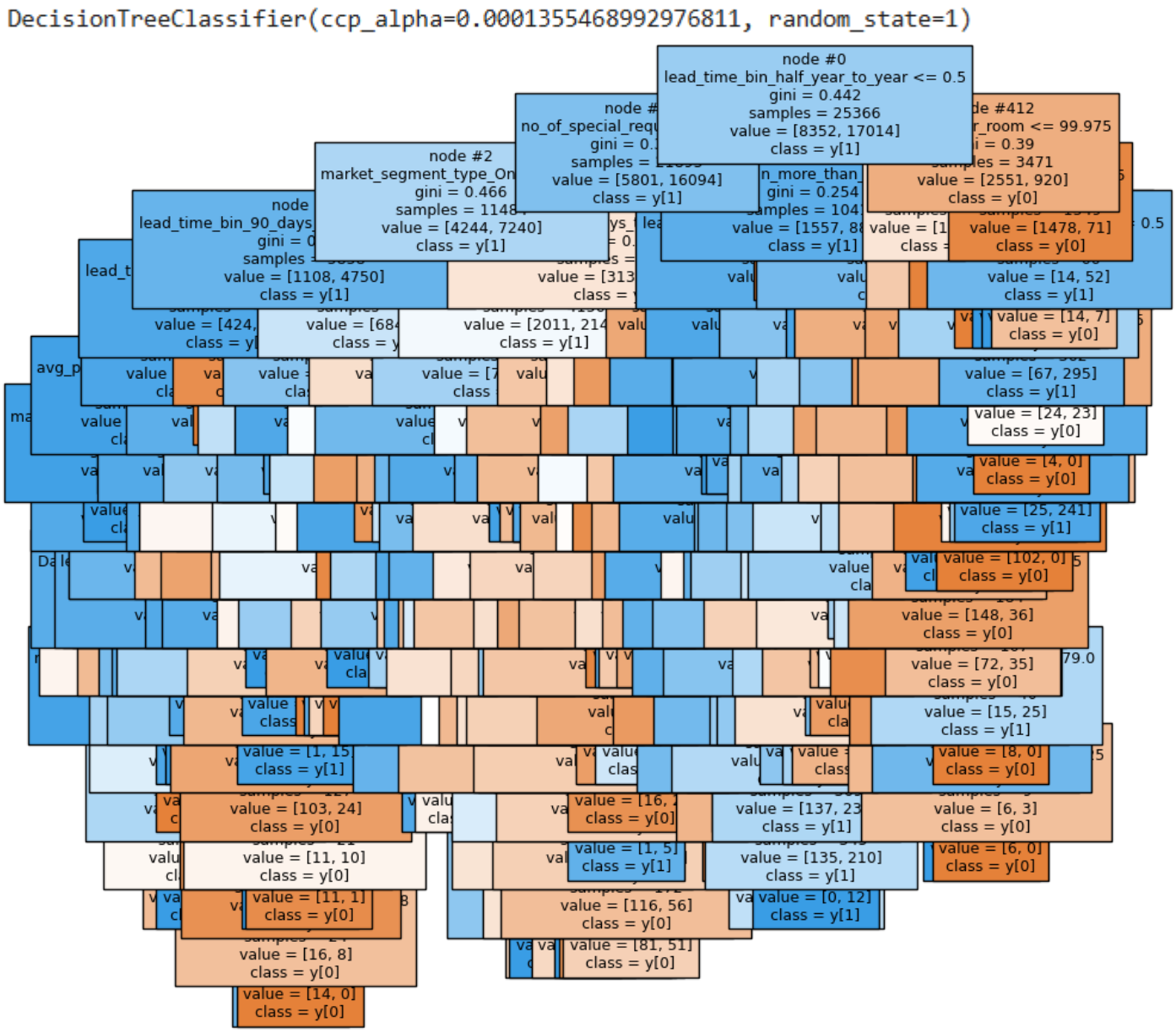


Figure-33 : Decision tree post tuning

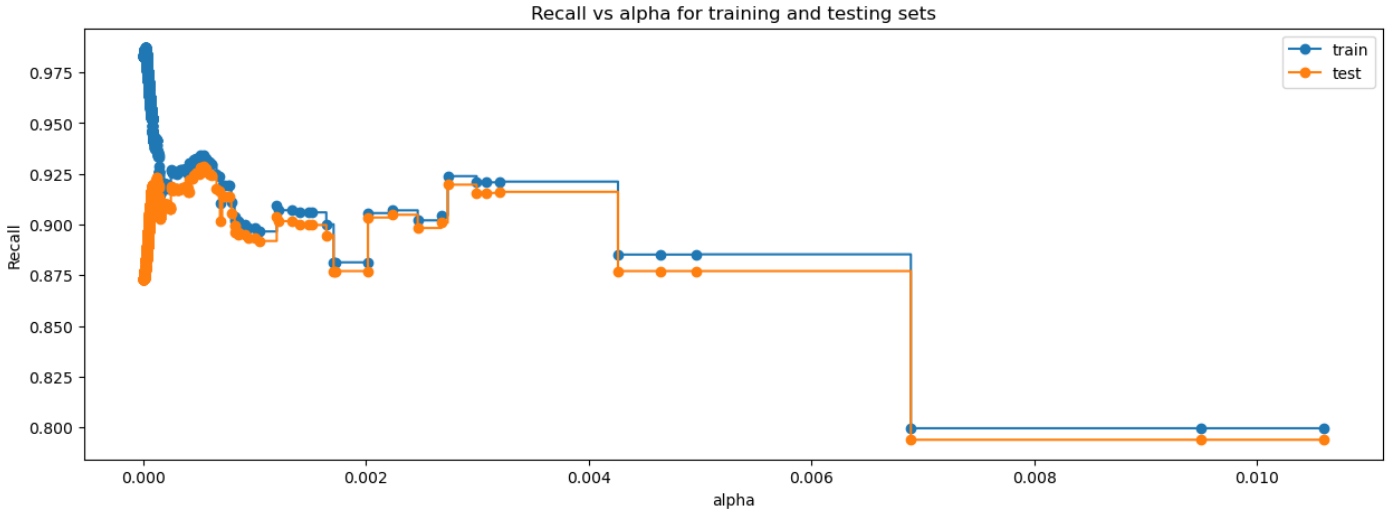


Figure-34 : Recall vs alfa for training & test data

Creating the model where we get highest train and test recall:

```
DecisionTreeClassifier(ccp_alpha=0.0005386455684601013, random_state=1)
```

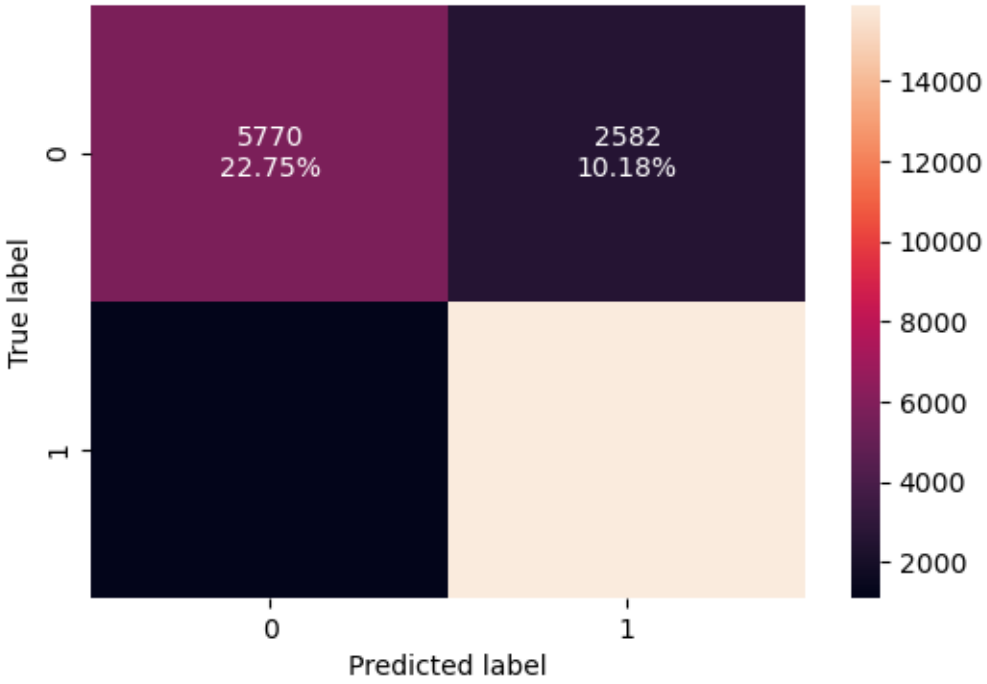


Figure-35 : Confusion matrix training best model

Best model training performance:

	Accuracy	Recall	Precision	F1
0	0.85406	0.93417	0.86025	0.89569

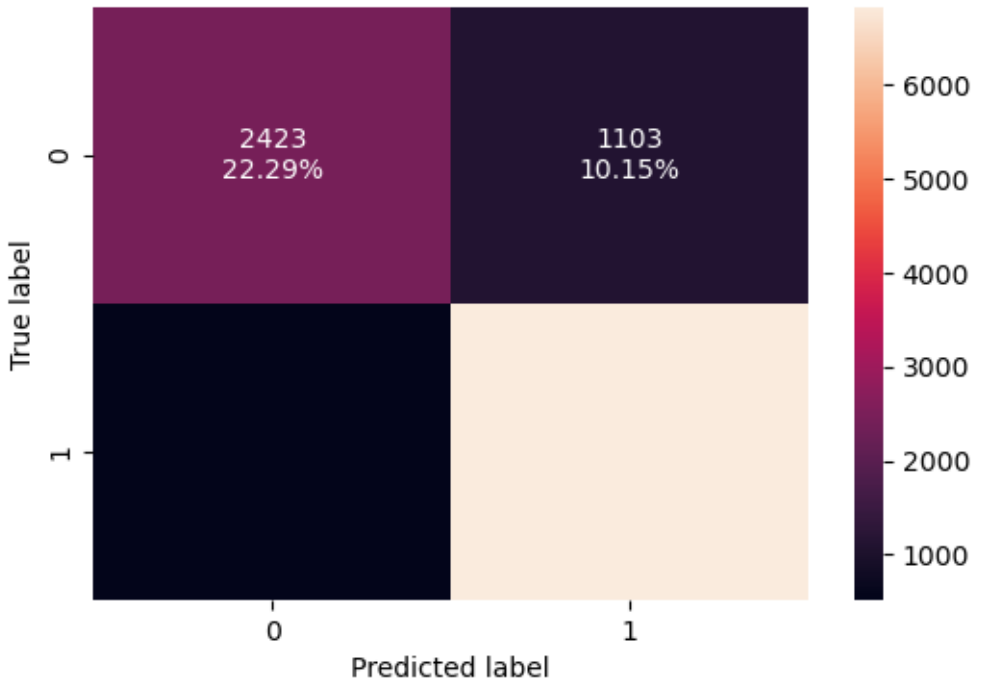


Figure-36 : Confusion matrix test best model

11. Model Performance Comparison and Final Model Selection:

Best model test performance:

	Accuracy	Recall	Precision	F1
0	0.85026	0.92853	0.86080	0.89339

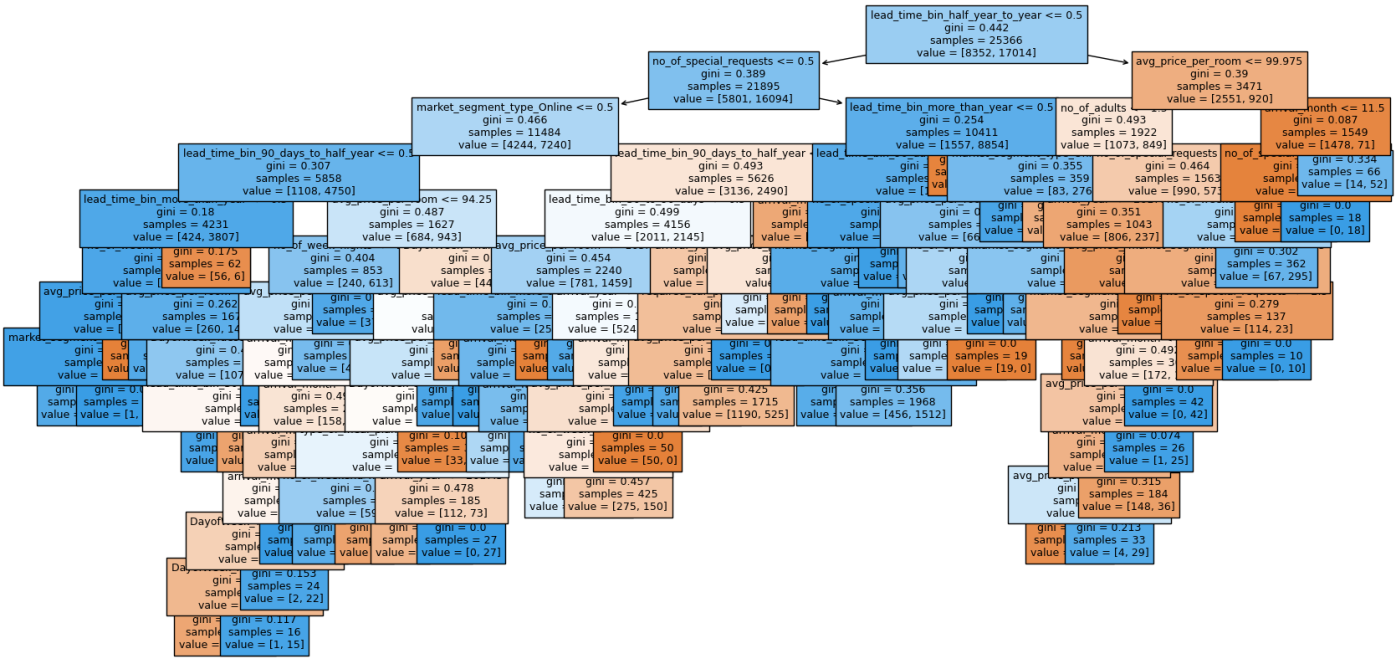


Figure-37 : Decision tree of the best model

Performance comparison:

Training performance comparison:

	Decision Tree (Initial)	Decision Tree (Pre-Pruning)	\
Accuracy	0.98403	0.89620	
Precision	0.98423	0.89547	
Recall	0.98403	0.89620	
F1 Score	0.98408	0.89569	

	Decision Tree (Post-Pruning)
Accuracy	0.85406
Precision	0.85274
Recall	0.85406
F1 Score	0.85006

Table 12: Performance comparison training data

Testing performance comparison:

	Decision Tree (Initial)	Decision Tree (Pre-Pruning)	\
Accuracy	0.84842	0.85688	
Precision	0.85144	0.85603	
Recall	0.84842	0.85688	
F1 Score	0.84956	0.85639	

	Decision Tree (Post-Pruning)
Accuracy	0.85026
Precision	0.84819
Recall	0.85026
F1 Score	0.84641

Table 13: Performance comparison test data

12. Inferences from the analysis: Actionable insights & recommendations:

Observation:

The post-pruning model, also known as model "best_model" shows the highest F1 score for testing data. I would use this model for future predictions.
Recommendations:

Knowing that lead time, price, and online bookings have the highest influence on cancellations we can infer that having cancellation clause during the online booking process would influence how customer's

book. Keep price of rooms near competitive pricing as it seems like guest will be book a room with the expectation of continued searches. Since repeating_guests have a very low cancellation rate, creating a loyalty program for those guest would help incentives other to move into that ca

tegory. During the online booking process, offering addtional customizations or special request would help reduce the likelihood of a cancelled booking.

END