MACHINE LEARNING – 1
INN HOTELS GROUP PROJECT

By: BENITA MERLIN.E
PGP-Data Science and Business Analytics.
BATCH: PGP DSBA. O. MAY24.A

Contents

1. INTRODUCTION **8**
1.1 PROJECT OVERVIEW 8
1.2. OBJECTIVE 8
1.3. CONSTRAINTS AND APPROACH 8
1.4. PROBLEM DEFINITION 8
2. DATA DESCRIPTION **9**
2.1. DATA DICTIONARY 9
2.2 DATA INFORMATION 11
3. STATISTICAL ANALYSIS: **11**
4. EXPLORATORY DATA ANALYSIS (EDA) **14**
4.1. Univariate Analysis 14
4.1.1. AVERAGE PRICE PER ROOM 14
4.1.2 LEAD TIME 15
4.1.3. NO_OF_ADULTS 17
4.1.4. NO.OF. CHILDREN 18
4.1.5. NO_OF_WEEKEND_NIGHTS 18
4.1.6. ARRIVAL_MONTH 21
4.1.7. MARKET_SEGMENT_TYPE 22
4.1.8 NO.OF. SPECIAL REQUEST 23
4.1.9 TYPE_OF_MEAL_PLAN 24
4.1.10 REQUIRED_CAR_PARKING_SPACE 25
4.1.11 ROOM_TYPE_RESERVED 25
4.1.12 BOOKING STATUS 26
4.1.13 REPEATED_GUEST 27
4.1.14 NO_OF_PREVIOUS_CANCELLATIONS 27
4.2 Bivariate Analysis 28
4.2.1 MARKET SEGMENT TYPE Vs AVERAGE PRICE PER ROOM 28
4.2.2 REPEATED_GUEST VS BOOKING_STATUS 29
4.2.3 NO_OF_SPECIAL_REQUESTS VS BOOKING_STATUS 29
4.2.4 MARKET_SEGMENT_TYPE VS BOOKING_STATUS 30
4.2.5 NO_OF_SPECIAL_REQUESTS VS BOOKING_STATUS 31
4.2.6 BOXPLOT OF NO_OF_SPECIAL_REQUESTS VS AVG_PRICE_PER_ROOM 31
4.2.7 ARRIVAL_MONTH VS BOOKING_STATUS 32
4.2.8 ARRIVAL_MONTH VS AVG_PRICE_PER_ROOM 32
4.2.9 LEAD_TIME VS BOOKING_STATUS 33

4.2.10 AVG_PRICE_PER_ROOM VS BOOKING_STATUS 33
4.3 MULTIVARIATE ANALYSIS 35
4.3.1 RELATIONSHIP AMONG NUMERICAL VARIABLES – HEAT MAP 35
4.3.2 PAIR PLOT FOR NUMERIC DATA 36
5. QUESTIONS 37
1. What are the busiest months in the hotel? 37
2. Which market segment do most of the guests come from? 37
3. Hotel rates are dynamic and change according to demand and customer demographics. What are the differences in room prices in different market segments? 38
4. What percentage of bookings are canceled? 39
5. Repeating guests are the guests who stay in the hotel often and are important to brand equity. What percentage of repeating guests cancel? 40
6. Many guests have special requirements when booking a hotel room. Do these requirements affect booking cancellation? 41
6. KEY MEANINGFUL OBSERVATIONS ON INDIVIDUAL VARIABLES AND THE RELATIONSHIP BETWEEN VARIABLES 42
1. Do special requests impact the prices of the room? 42
2. How does price per room impact booking status? 42
3. How does lead time impact booking status? 44
4. Number of bookings cancelled each month 46
5. Family_members vs booking_status 47
6. Arrival_month vs booking_status 48
7. Repeated_Guest Vs Booking_Status 49
8. Grouping the data on arrival months and extracting the count of bookings 50
7. DATA PREPROCESSING 50
7.1 Outlier Detection 51
7.2 Feature Engineering 51
8. LOGISTIC REGRESSION 52
8.1 Model Building - Logistic Regression 52
8.2 Model Performance Evaluation 53
8.3 Detecting and Dealing with Multicollinearity 55
9. KNN CLASSIFIER (SKLEARN) 71
9.1 Model Evaluation 71
9.2 K- Nearest Neighbor 72
9.3 K with different values 74
10. NAIVE BAYES 74
10.1 Check Model Performance 75

10.1.1 Model Performance on Training set 75
10.1.2 Model Performance on Test set 76
10.2 Comparison of Models and Final Model Selection 76
10.2.1 Training performance comparison 76
10.2.2 Test set performance comparison 77
11. DECISION TREE 77
11.1 Decision Tree (default) 77
11.1.1 Model Performance on Training set 78
11.1.2 Model Performance on Test set 79
11.2 Decision Tree (with class_weights) 79
11.2.1 Model Performance on Training set 80
11.2.2 Model Performance on Test set 81
11.3 Decision Tree (Pre-pruning) 81
11.3.1 Model Performance on Training set 82
11.3.2 Model Performance on Test set 83
11.3.3 Pre-Pruned Tree 84
11.3.4 Text report showing the rules of a decision tree 84
11.3.5 Importance of Features in the Tree Building 85
11.4 Decision Tree (Post pruning) 85
11.4.1 Total impurity of leaves vs effective alphas of pruned tree 86
11.4.2 Decision Tree using effective alphas 86
11.4.3 Decision tree classifier 88
11.4.4 Post - Pruned Tree 90
11.4.5 Text report for Post Pruned Tree 90
11.5 Comparison of Models and Final Model Selection 91
11.5.1 Training performance comparison 91
11.5.2 Testing performance comparison 92
12. MODEL PERFORMANCE COMPARISON AND FINAL MODEL SELECTION 92
12.1 Training Performance 92
12.2 Test Set Performance 92
13 ACTIONABLE INSIGHTS AND RECOMMENDATIONS: 93
13.1. Recommendation 93
13.2. Actionable Insights 94

LIST OF FIGURES

Figure 1 Statistical Summary 12
Figure 2 Distribution of Average Price Per Room Analysis 14
Figure 3 Distribution of Lead time Analysis 15
Figure 4 Barplot of No.of.Adults 17
Figure 5 Barplot of no.of. children 18
Figure 6 Barplot of No_Of_Weekend_Nights 19
Figure 7 Barplot of No_Of_Week_Nights 20
Figure 8 Barplot Of Arrival_Month 21
Figure 9 Barplot of Market_Segment_Type 22
Figure 10 Barplot Of No.of.Special Request 23
Figure 11 barplot of Type_of_Meal_Plan 24
Figure 12 Barplot of Required_Car_Parking_Space 25
Figure 13 Barplot of Room_Type_Reserved 25
Figure 14 Barplot of Booking Status 26
Figure 15 Barplot of Repeated_Guest 27
Figure 16 Barplot of No_of_Previous_Cancellations 27
Figure 17 Boxplot of Market Segment Type Vs Average Price Per Room 28
Figure 18 Stacked Barplot of Repeated_Guest Vs Booking_Status 29
Figure 19 Catplot of No_of_Special_Requests Vs Booking_Status 29
Figure 20 Stacked Barplot of Market_Segment_Type Vs Booking_Status 30
Figure 21 Stacked Barplot of No_of_Special_Requests Vs Booking_Status 31
Figure 22 Boxplot of No_of_Special_Requests Vs Avg_Price_Per_Room 31
Figure 23 Stacked Barplot of Arrival_Month Vs Booking_Status 32
Figure 24 Line plot of Arrival_Month Vs Avg_Price_Per_Room 32
Figure 25 Boxplot of Lead_Time Vs Booking_Status 33
Figure 26 Distribution_Plot_Wrt_Target of Avg_Price_Per_Room Vs Booking_Status 34
Figure 27 Heatmap for Numerical Data 35
Figure 28 Pair Plot for Numeric Data 36
Figure 29 Distribution of Arrival Month 37
Figure 30 Barplot of Market Segment Type 37
Figure 31 Barplot of Market Segment Vs Avg Price Per Room 38
Figure 32 Barplot of Booking Status 39
Figure 33 Stacked_Barplot of Repeated_Guest Vs Booking_Status 40
Figure 34 Stacked_Barplot No_of_Special_Requests Vs Booking_Status 41
Figure 35 boxplot of no_of_special_requests vs avg_price_per_room 42
Figure 36 Distribution_Plot_Wrt_Target of Avg_Price_Per_Room Vs Booking_Status 43
Figure 37 Distribution_Plot_Wrt_Target of Lead_Time Vs Booking_Status 44
Figure 38 Stacked_Barplot of Arrival_Month Vs Booking_Status 46
Figure 39 stacked barplot of no.of family members vs booking status 47
Figure 40 Stacked Barplot of No.of Family Members Vs Booking Status 48
Figure 41 Stacked_Barplot Of Repeated_Guest Vs Booking_Status 49
Figure 42 Splitting data in train and test sets 52
Figure 43 Logistic Regression Result 53

Figure 44 Confusion matrix 54
Figure 45 Training Performance 55
Figure 46 VIF Series before feature selection 56
Figure 47 VIF Series 2 before removing multicollinearity 57
Figure 48 VIF Series 2 58
Figure 49 training performance 58
Figure 50 Logistic Regression Result after removing multicollinearity 59
Figure 51 Logistic Regression Result after removing high p- value 60
Figure 52 Converting coefficients to odds 61
Figure 53 Confusion Matrix 61
Figure 54 Training Set Performance Of The New Model 62
Figure 55 Confusion Matrix 62
Figure 56 Test Set Performance Of The New Model 63
Figure 57 ROC -AUC On Training Set 64
Figure 58 Confusion Matrix 65
Figure 59 Model Performance For Training Set 65
Figure 60 ROC-AUC On Test Set 66
Figure 61 Confusion Matrix 67
Figure 62 Figure 63 Model Performance on Test Set 67
Figure 64 Precision - Recall curve 68
Figure 65 Confusion Matrix 69
Figure 66 Model Performance on Training set 69
Figure 67 Confusion Matrix 70
Figure 68 Test performance 70
Figure 69 Confusion Matrix for KNN =3 72
Figure 70 Model Performance on Training set 73
Figure 71 Confusion Matrix (KNN =3) for Test set 73
Figure 72 Model Performance on Test set 73
Figure 73 KNN Recall Score for Different Values of K 74
Figure 74 Confusion Matrix 75
Figure 75 Model Performance on Training set 75
Figure 76 Confusion Matric 76
Figure 77 Model Performance on Test set 76
Figure 78 Comparison On Training Performance Model 76
Figure 79 Comparison on Test Performance Model 77
Figure 80 Confusion Matrix 78
Figure 81 Model Performance on Training set 78
Figure 82 confusion matrix 79
Figure 83 Model Performance on Test set 79
Figure 84 Confusion Matrix 80
Figure 85 Model Performance on Training set 80
Figure 86 confusion matrix 81
Figure 87 Model Performance on Test set 81
Figure 88 confusion matrix 82
Figure 89 Model Performance on Training set 82
Figure 90 confusion matrix 83
Figure 91 Model Performance on Test Set 83
Figure 92 Pre-Pruned Tree 84

Figure 93 Text report of Pre- Pruned Tree 84
Figure 94 Feature Importance of pre -pruned Tree 85
Figure 95 Total Impurity Vs Effective Alpha For Training Set 86
Figure 96 No.Of Nodes Vs Alpha And Depth Vs Alpha 87
Figure 97 Recall vs alpha for training and testing sets 88
Figure 98 confusion matrix 89
Figure 99 Model Performance on Training set 89
Figure 100 confusion matrix 89
Figure 101 Model Performance on Test set 90
Figure 102 Post Pruned Tree 90
Figure 103 Text report for Post Pruned Tree 90
Figure 104 Feature Importance of Post Pruned Tree 91
Figure 105 Training performance comparison 91
Figure 106 Testing performance comparison 92
Figure 107 Training Performance comparison 92
Figure 108 Testing performance comparison 92

LIST OF TABLES

Table 1 Data Dictionary 10
Table 2 Data Information 11

1. INTRODUCTION

1.1 PROJECT OVERVIEW

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.

1.2. 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.

1.3. CONSTRAINTS AND APPROACH

We will use historical data to perform exploratory data analysis (EDA), feature

engineering, and build a model. The insights gained will guide recommendations for business strategies

1.4. PROBLEM DEFINITION

- INN Hotels Group is experiencing a high number of booking cancellations across its chain of hotels in Portugal, negatively impacting revenue and operational efficiency.
- They need a data-driven solution to predict which bookings are likely to be canceled in advance.
- Analyze the historical booking data to identify key factors that influence cancellations.
- Develop a Machine Learning model to predict the likelihood of a booking being canceled

2. DATA DESCRIPTION

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

2.1. DATA DICTIONARY

SI.NO	VARIABLE	DESCRIPTION
1	Booking_ID	the unique identifier of each booking
2	no_of_adults	Number of adults
3	no_of_children	Number of Children
4	no_of_weekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
5	no_of_week_nights	Number of weeknights (Monday to Friday) the guest stayed or booked to stay at the hotel
6	type_of_meal_plan	• Type of meal plan booked by the customer: o Not Selected – No meal plan selected o Meal Plan 1 – Breakfast o Meal Plan 2 – Half board (breakfast and one other meal) o Meal Plan 3 – Full board (breakfast, lunch, and dinner)
7	required_car_parking_space	• Does the customer require a car parking space? (0 - No, 1- Yes)
8	room_type_reserved	Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels Group
9	lead_time	Number of days between the date of booking and the arrival date
10	arrival_year	Year of arrival date

11	arrival_month	Month of arrival date
12	arrival_date	Date of the month
13	market_segment_type	Market segment designation
14	repeated_guest	• Is the customer a repeated guest? (0 - No, 1- Yes)
15	no_of_previous_cancellations	Number of previous bookings that were canceled by the customer prior to the current booking
16	no_of_previous_bookings_not_canceled	• Number of previous bookings not canceled by the customer prior to the current booking
17	avg_price_per_room	• Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
18	no_of_special_requests	• Total number of special requests made by the customer (e.g. high floor, view from the room, etc)
19	booking_status	• Flag indicating if the booking was canceled or not.

Table 1 Data Dictionary

2.2 DATA 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 Data Information

- There are in total 36275 rows and 19 columns present.
- There are 14 numeric (1 float and 13 int type) and 5 string (object type) columns in the data.
- The target variable is the booking_status, which is of object type
- There are no duplicates.
- There are no missing values in all the columns.

3. STATISTICAL ANALYSIS:

The statistical analysis provides a summary of the key metrics for the numerical columns in the dataset. This includes measures such as the count, mean, standard deviation, minimum, 25th percentile (Q1), median (50th percentile, Q2), 75th percentile (Q3), and maximum values for each column.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Booking_ID	36275	36275	INN00001	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
no_of_adults	36275.00000	NaN	NaN	NaN	1.84496	0.51871	0.00000	2.00000	2.00000	2.00000	4.00000
no_of_children	36275.00000	NaN	NaN	NaN	0.10528	0.40265	0.00000	0.00000	0.00000	0.00000	10.00000
no_of_weekend_nights	36275.00000	NaN	NaN	NaN	0.81072	0.87064	0.00000	0.00000	1.00000	2.00000	7.00000
no_of_week_nights	36275.00000	NaN	NaN	NaN	2.20430	1.41090	0.00000	1.00000	2.00000	3.00000	17.00000
type_of_meal_plan	36275	4	Meal Plan 1	27835	NaN	NaN	NaN	NaN	NaN	NaN	NaN
required_car_parking_space	36275.00000	NaN	NaN	NaN	0.03099	0.17328	0.00000	0.00000	0.00000	0.00000	1.00000
room_type_reserved	36275	7	Room_Type 1	28130	NaN	NaN	NaN	NaN	NaN	NaN	NaN
lead_time	36275.00000	NaN	NaN	NaN	85.23256	85.93082	0.00000	17.00000	57.00000	126.00000	443.00000
arrival_year	36275.00000	NaN	NaN	NaN	2017.82043	0.38384	2017.00000	2018.00000	2018.00000	2018.00000	2018.00000
arrival_month	36275.00000	NaN	NaN	NaN	7.42365	3.06989	1.00000	5.00000	8.00000	10.00000	12.00000
arrival_date	36275.00000	NaN	NaN	NaN	15.59700	8.74045	1.00000	8.00000	16.00000	23.00000	31.00000
market_segment_type	36275	5	Online	23214	NaN	NaN	NaN	NaN	NaN	NaN	NaN
repeated_guest	36275.00000	NaN	NaN	NaN	0.02564	0.15805	0.00000	0.00000	0.00000	0.00000	1.00000
no_of_previous_cancellations	36275.00000	NaN	NaN	NaN	0.02335	0.36833	0.00000	0.00000	0.00000	0.00000	13.00000
no_of_previous_bookings_not_canceled	36275.00000	NaN	NaN	NaN	0.15341	1.75417	0.00000	0.00000	0.00000	0.00000	58.00000
avg_price_per_room	36275.00000	NaN	NaN	NaN	103.42354	35.08942	0.00000	80.30000	99.45000	120.00000	540.00000
no_of_special_requests	36275.00000	NaN	NaN	NaN	0.61966	0.78624	0.00000	0.00000	0.00000	1.00000	5.00000
booking_status	36275	2	Not_Canceled	24390	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 1 Statistical Summary

Observation:

- There are 36,275 bookings in total
- The average number of adults per booking is around 1.84, which suggests most bookings are for 2 adults (as reflected in the median and mode).
- The average number of children per booking is low, at 0.105, indicating that the majority of bookings are for adults without children. There are a few instances where up to 10 children were part of a booking (the maximum value).
- Weekend nights: On average, bookings include 0.81 weekend nights, with most bookings either including no weekend nights or just one weekend night.
- Weekday nights: The average number of weeknights is 2.2, with a maximum of 17 weeknights for certain bookings. The mode and median are around 2 or 3 nights, indicating that most bookings last for 1-3 weekdays.
- The average lead time for bookings is 85.23 days, meaning most guests book around three months in advance. However, the lead time ranges from 0 to 443 days (more than a year), which highlights a wide variation in booking behavior.
- The arrival year is centered around 2017-2018 (the dataset contains bookings primarily from these years).
- The arrival months range from 1 to 12, with an average of July (7), indicating a tendency for bookings in mid-year. The months of May, August, October, and December seem significant based on the 25th, 50th, and 75th percentiles.

- The arrival date (within a month) has an average of the 15th day of the month, with the dates spread across the entire month (ranging from 1 to 31).
- Room Type Reserved: There are 7 distinct room types available, with Room_Type 1 being the most common (appears 28,130 times).
- Meal Plan: There are 4 distinct meal plans, and Meal Plan 1 is by far the most popular, as it appears in 27,835 bookings.
- Special Requests: On average, each booking has 0.62 special requests, but a large number of bookings have no special requests. The highest number of special requests for a single booking is 5.
- Very few customers require car parking, with only about 3% of the bookings needing a parking space (0.03099 mean for required_car_parking_space).
- There are 5 market segments, with "Online" bookings dominating, as this segment accounts for 23,214 bookings.
- Repeated Guests: Only about 2.56% of guests are repeat customers, indicating that the majority are first-time visitors.
- The average number of previous cancellations is quite low, at 0.02 per booking, but there are a few guests who have canceled up to 13 times.
- Non-canceled bookings: Most guests haven't canceled previously, with the median and mode both being 0, but some customers have as many as 58 prior bookings that were not canceled.
- The average price per room is $103.42, with a significant spread (standard deviation of 35.09). Prices range from 0 to 540, though most prices are concentrated between 80.30 and 120.
- Out of 36,275 bookings, 24,390 are marked as "Not Canceled", which indicates a cancellation rate of around 33%, implying that one-third of bookings in the dataset have been canceled.

4. EXPLORATORY DATA ANALYSIS (EDA)

4.1. Univariate Analysis

4.1.1. AVERAGE PRICE PER ROOM

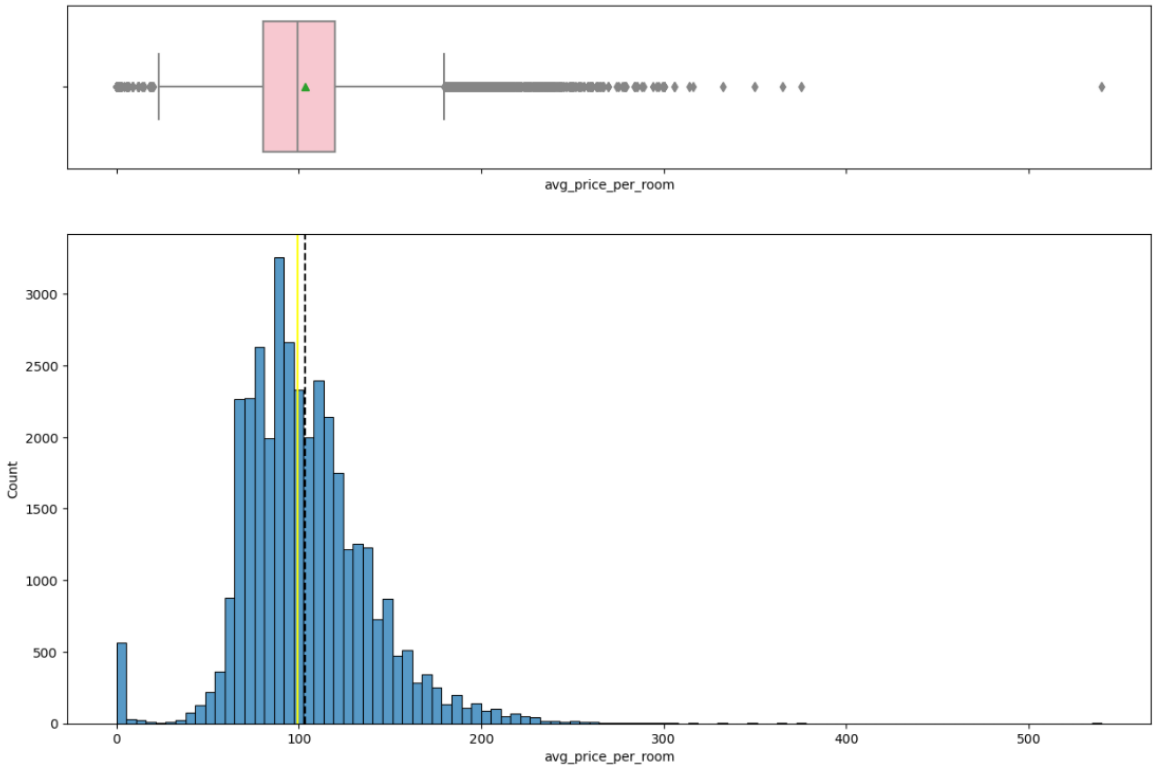


Figure 2 Distribution of Average Price Per Room Analysis

Observation:

1. Boxplot (Top Plot):

- Outliers: There are several outliers on both ends, especially on the higher end (room prices above 200). These extreme values could indicate either luxury or premium bookings.
- Median: The green triangle shows the median value of avg_price_per_room, which is located near the center of the interquartile range (IQR), indicating that most room prices are clustered around this value.
- Skewness: The long tail on the right side suggests that the distribution is positively skewed, meaning there are some very high room prices that drive the average up.

2. Histogram (Bottom Plot):

- Distribution Shape: The histogram shows that the distribution is unimodal and right-skewed, with most room prices concentrated between 50 and 150. A significant number of bookings fall in this range.
- Mean vs. Median: The mean (yellow line) is slightly higher than the median (black dashed line), further indicating a positive skew in the data. This suggests that while most prices are lower, the higher outliers increase the average room price.

- Price Range: Very few rooms have a price exceeding 200, which could indicate a niche market for high-end rooms or pricing strategies that target specific customer segments.

3. Overall Insights:

- Outliers: The presence of outliers, particularly high-priced rooms, could be important for understanding whether premium or luxury bookings are more prone to cancellation.
- Skewness: The right-skewed distribution could mean that higher-priced rooms may contribute to a different cancellation behavior compared to mid-range bookings.
- Majority of Bookings: Since most room prices are within a reasonable range (50-150), this price segment could be where most cancellations or successful bookings occur. Understanding customer behavior in this range might help predict cancellation likelihood.

4.1.2 LEAD TIME

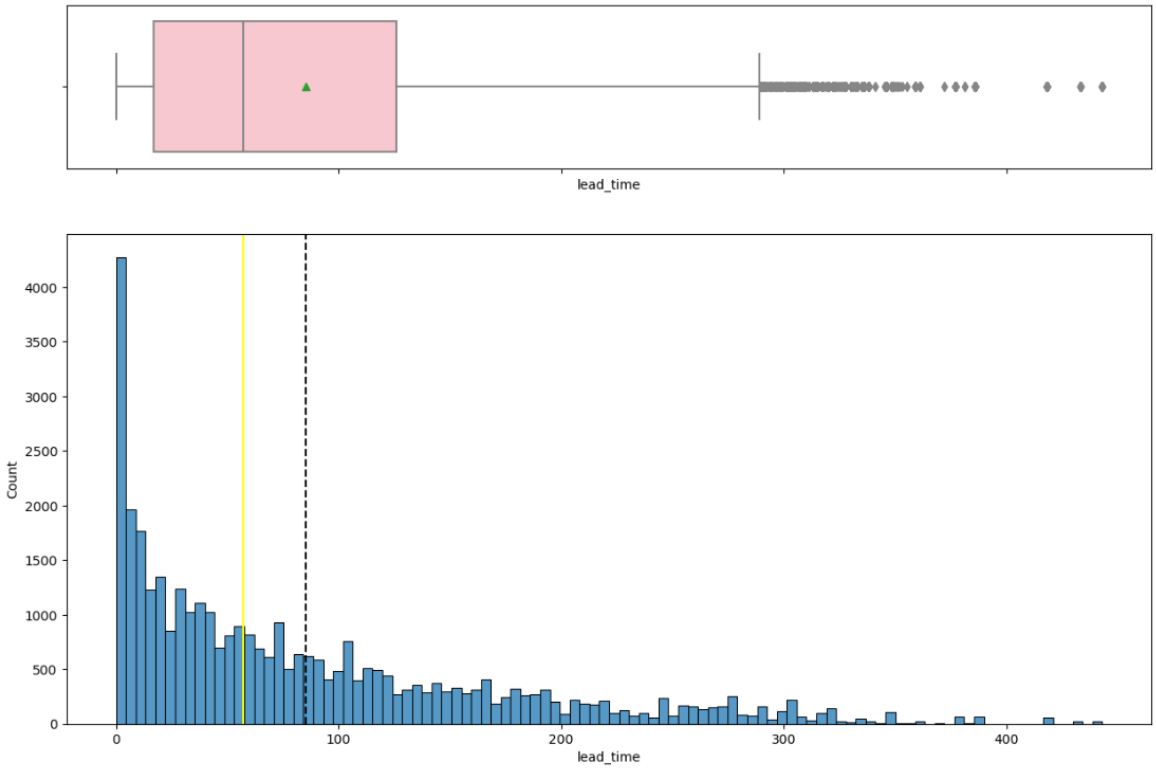


Figure 3 Distribution of Lead time Analysis

Observation:

1. Boxplot (Top Plot):

- Outliers: There are a significant number of outliers at the high end of the lead time, with values extending beyond 300 days up to 440+ days. These outliers indicate that some bookings are made almost a year in advance.

- Interquartile Range (IQR): The bulk of the data lies within a lead time of approximately 0 to 130 days. The box covers the middle 50% of the data, which shows that most bookings are made between 17 and 126 days in advance (from previous insights).
- Median Lead Time: The median value (black line in the box) is around 57 days, meaning half of the bookings are made within two months of the stay.

2. Histogram (Bottom Plot):

- Distribution Shape: The distribution is right-skewed, meaning that most bookings have a shorter lead time, but there are still a significant number of bookings with longer lead times.
- Mode: The highest frequency (peak) of bookings has a lead time of 0-10 days, showing that many customers book their stays at short notice.
- Mean vs. Median: The mean lead time (yellow line) is greater than the median (black dashed line), indicating that the long tail of higher lead times pulls the mean up. The mean lead time is around 85 days, as mentioned in earlier statistics, but most bookings are made much earlier.
- Long Tail: The lead time distribution extends beyond 300 days, though the frequency of such long lead times is low compared to shorter bookings.

3. Overall Insights:

- Short-Notice Bookings: A large portion of bookings occurs within the first 0-30 days of the stay date, which could suggest a tendency for last-minute planning by customers.
- Long-Term Planners: Some customers plan far in advance, but they represent a minority. The long tail in the histogram shows that while it’s rare, some bookings are made almost a year in advance.
- Cancellations and Lead Time: Given the wide spread in lead times, it's possible that both short-term and long-term bookings may have different cancellation behaviors. Shorter lead times might correlate with spontaneous trips and possibly higher cancellation rates, while longer lead times could suggest more planned and firm bookings.

4.1.3. NO_OF_ADULTS

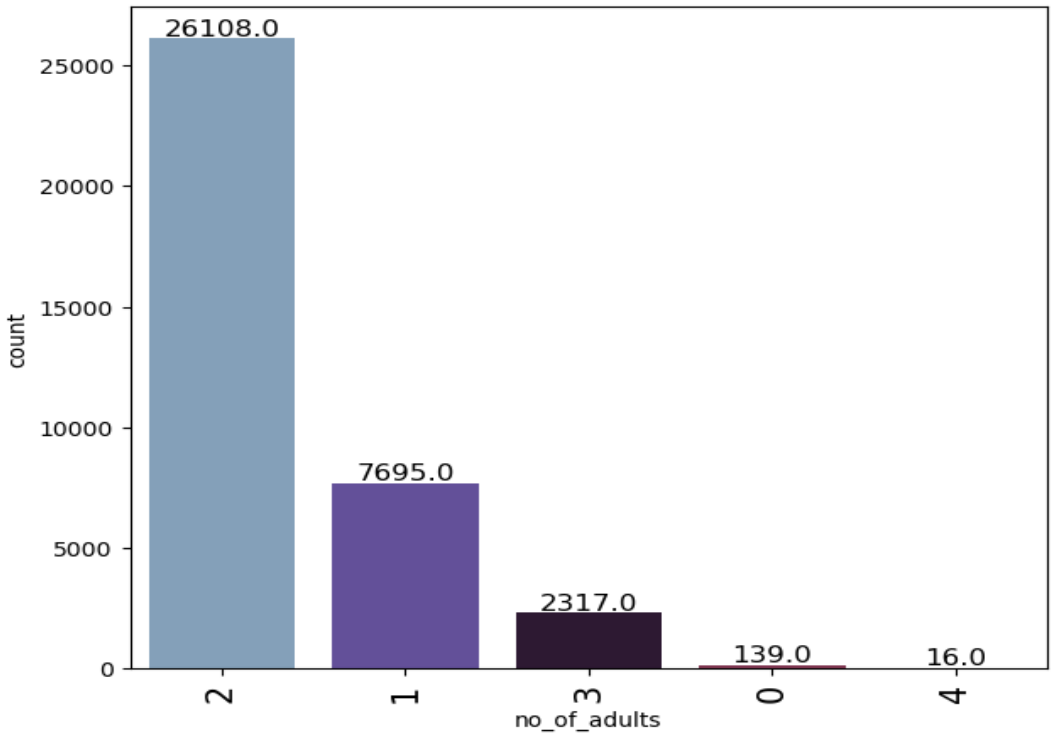


Figure 4 Barplot of No.of.Adults

Observation:

- The majority of the bookings (26,108) involve two adults. This indicates that the most common booking is made by couples or two individuals traveling together.
- The second-largest group (7,695 bookings) consists of single adult bookings. This suggests a significant number of solo travelers or single occupancy in rooms.
- A smaller number (2,317) of bookings have three adults, likely representing group travel with three individuals or small families with an additional adult.
- There are 139 bookings with zero adults, which could be data entry errors, as it's unlikely to have a booking without at least one adult. These entries may need further investigation or cleaning.

There are 16 bookings involving four adults. These might represent larger families or groups traveling together.

4.1.4. NO.OF. CHILDREN

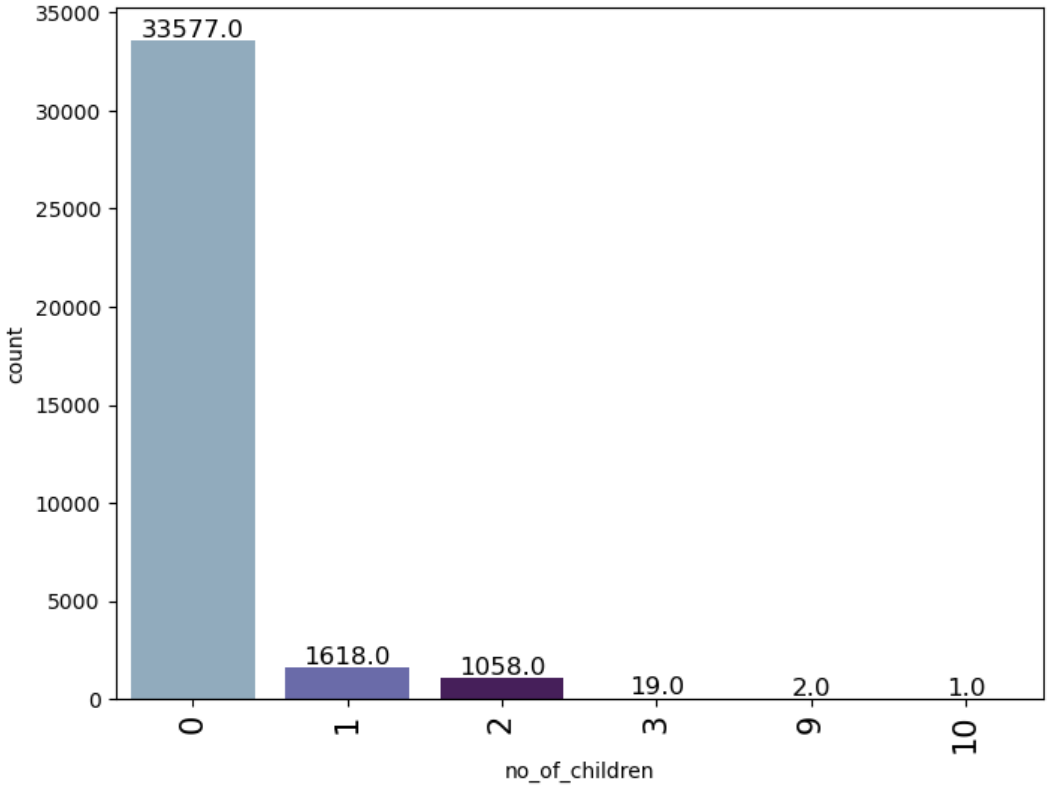


Figure 5 Barplot of no.of. children

Observation:

- The majority of the bookings (33,577) involve zero children. This indicates that the most common booking is made by couples or two individuals traveling together.
- There are 1618 bookings with one children.
- There are 1058 bookings with two children which is quite lesser than one children.
- Only 19 bookings with three children.
- Two bookings have 9 children.These might represent larger families or groups traveling together.

There is 1 booking with 10 children. These might represent larger families or groups traveling together.

4.1.5. NO_OF_WEEKEND_NIGHTS

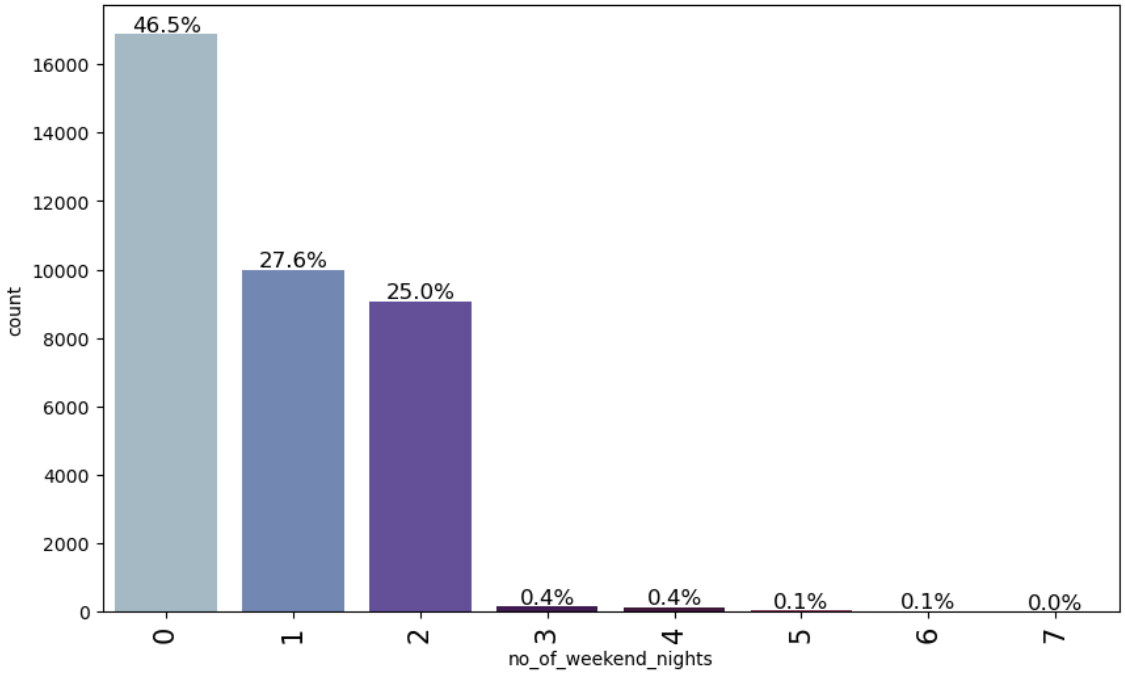


Figure 6 Barplot of No_Of_Weekend_Nights

Observation

- 46.5% of the bookings were not made for weekend nights.
- 27.6% of the bookings were made for 1 weekend night.
- 25% of the bookings were made for 2 weekend night.
- Only 0.4% of the bookings were made for 3 and 4 weekend nights.
- Only 0.1% of the bookings were made for 5 weekend nights.

This shows that a majority of bookings either avoided weekends altogether or included just one weekend, with very few involving three, four and 5 weekend nights.

4.1.6. NO_OF_WEEK_NIGHTS

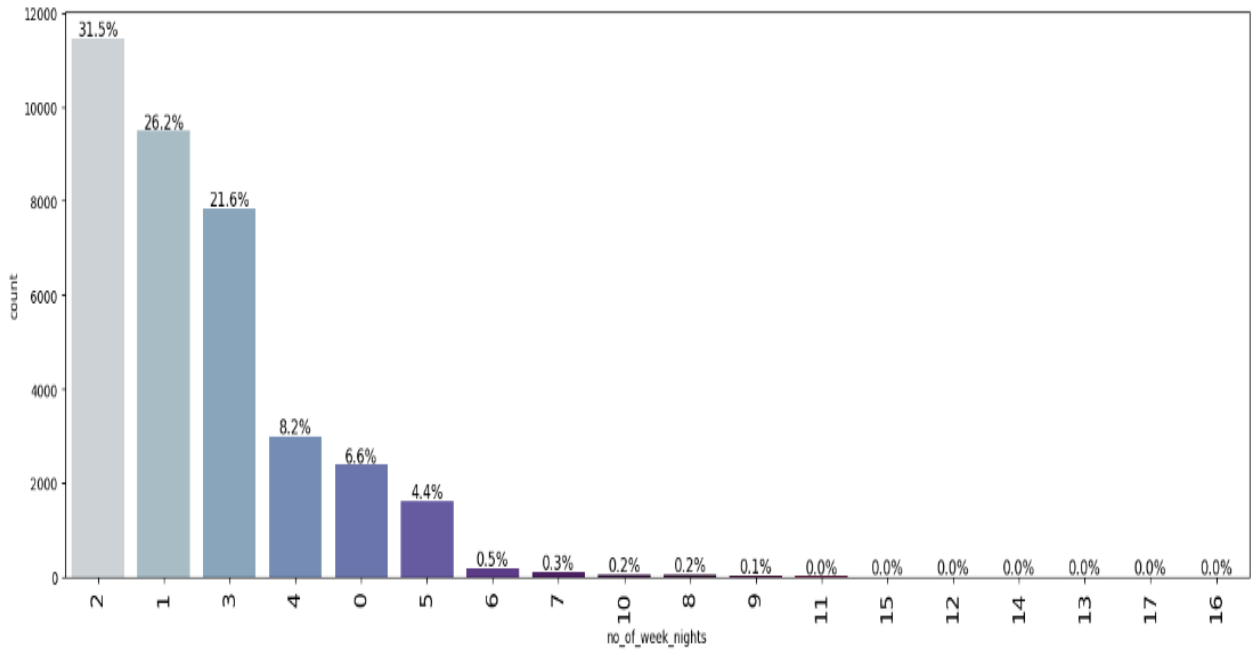


Figure 7 Barplot of No_Of_Week_Nights

Observation:

- 31.5% of the bookings were made for 2 week nights.
- 26.2% of the bookings were made for 1 week night.
- 21.6% of the bookings were made for 3 week nights.
- 8.2% of the bookings were made for 4 week nights.
- 6.6% of the bookings were not made for any week night.
- 4.4% of the bookings were made for 5 week nights.
- 0.5% of the bookings were made for 6 week nights
- 0.3% of the bookings were made for 7 week nights
- 0.2% of the bookings were made for 8 and 10 week nights
- 0.1 of the bookings were made for 9 week nights

This shows that most bookings were made for up to 3 weeknights, with very few extending beyond 5 weeknights.

4.1.6. ARRIVAL_MONTH

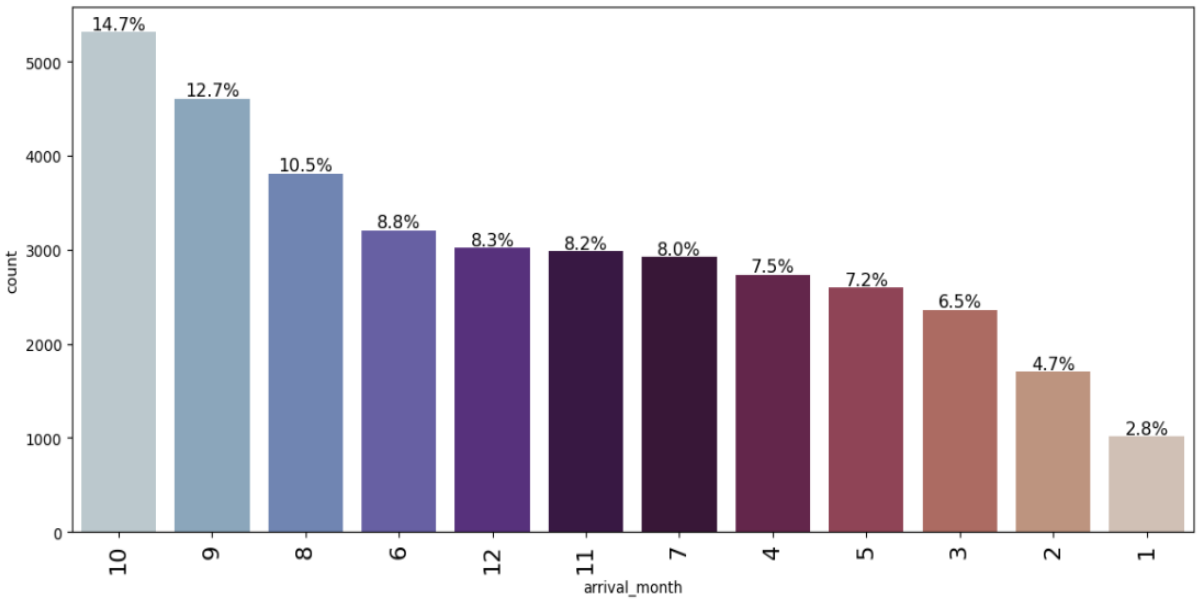


Figure 8 Barplot Of Arrival_Month

Observation:

- 14.7% bookings were made on october.
- 12.7% bookings were made on september.
- 10.5% bookings were made on August.
- 8.8% bookings were made on June.
- 8.3% bookings were made on December.
- 8.2% bookings were made on November.
- 8% bookings were made on July.
- 7.5% bookings were made on April.
- 7.2% bookings were made on May.
- 6.5% bookings were made on March.
- 4.7% bookings were made on February.
- 2.8% bookings were made on January.

This indicates that the highest number of bookings occurred in the months of October, then in September, and August, while the fewest bookings were made in January and February.

4.1.7. MARKET_SEGMENT_TYPE

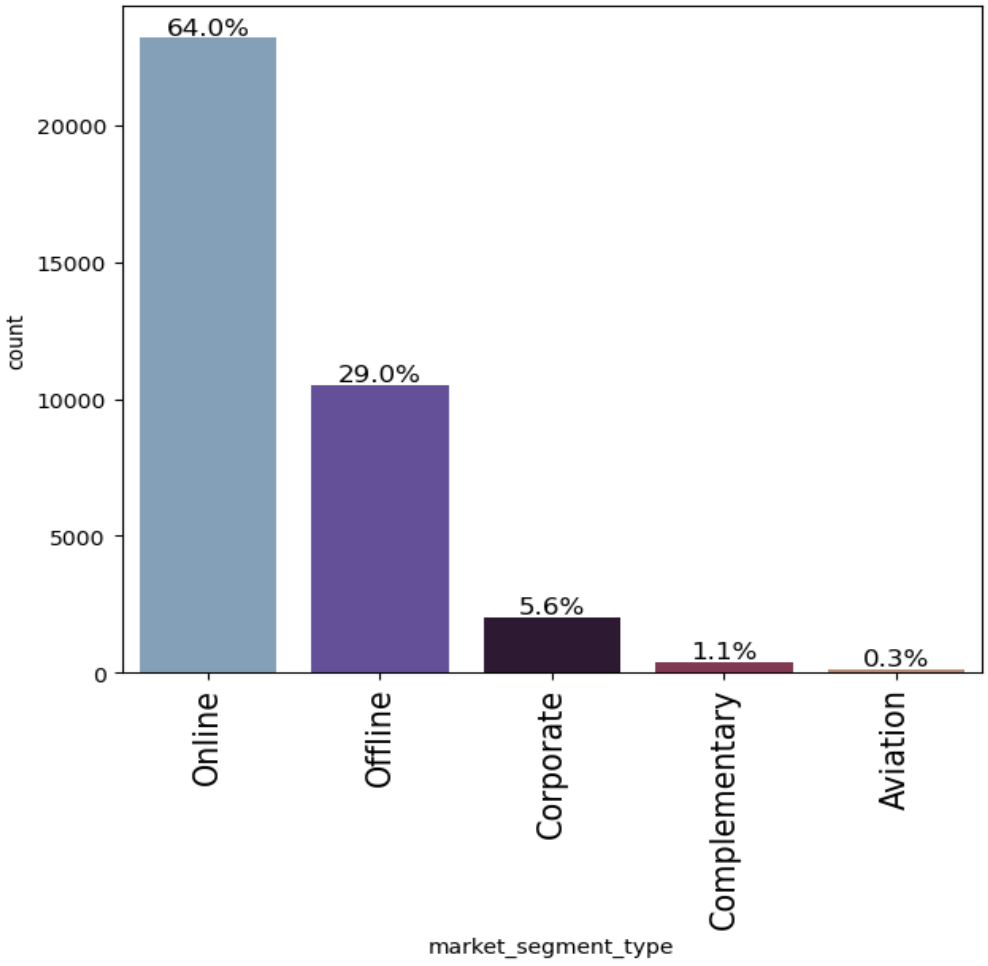


Figure 9 Barplot of Market_Segment_Type

Observation:

- 64.0% of the bookings were made through the Online market segment.
- 29.0% of the bookings were made through the Offline segment.
- 5.6% of the bookings came from the Corporate segment.
- 1.1% of the bookings were from the Complementary segment.
- Only 0.3% of the bookings were made through the Aviation segment.

This indicates that the majority of bookings were made online, followed by offline, with significantly fewer bookings from corporate, complementary, and aviation channels.

4.1.8 NO.OF. SPECIAL REQUEST

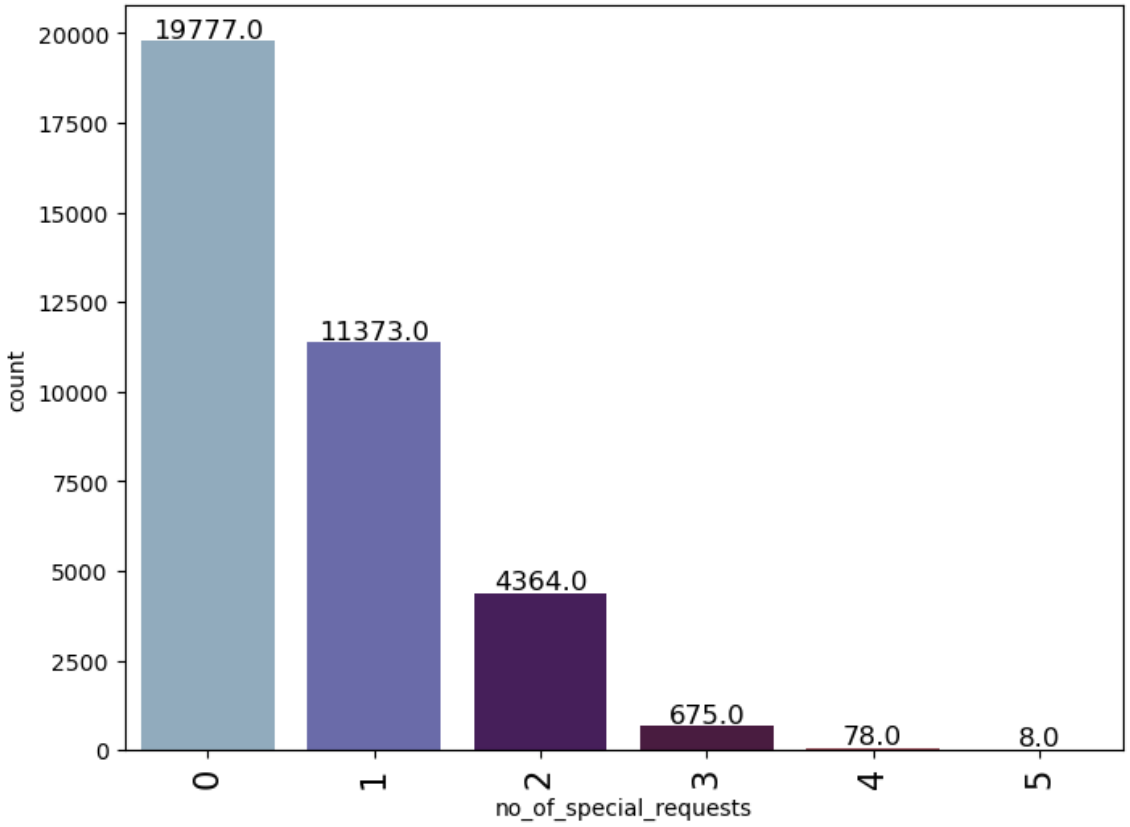


Figure 10 Barplot Of No.of.Special Request

Observation:

- 19,777 bookings had 0 special requests.
- 11,373 bookings had 1 special request.
- 4,364 bookings had 2 special requests.
- 675 bookings had 3 special requests.
- 78 bookings had 4 special requests.
- 8 bookings had 5 special requests.

The majority of bookings (almost half) had no special requests, while only a small number involved more than three requests.

4.1.9 TYPE_OF_MEAL_PLAN

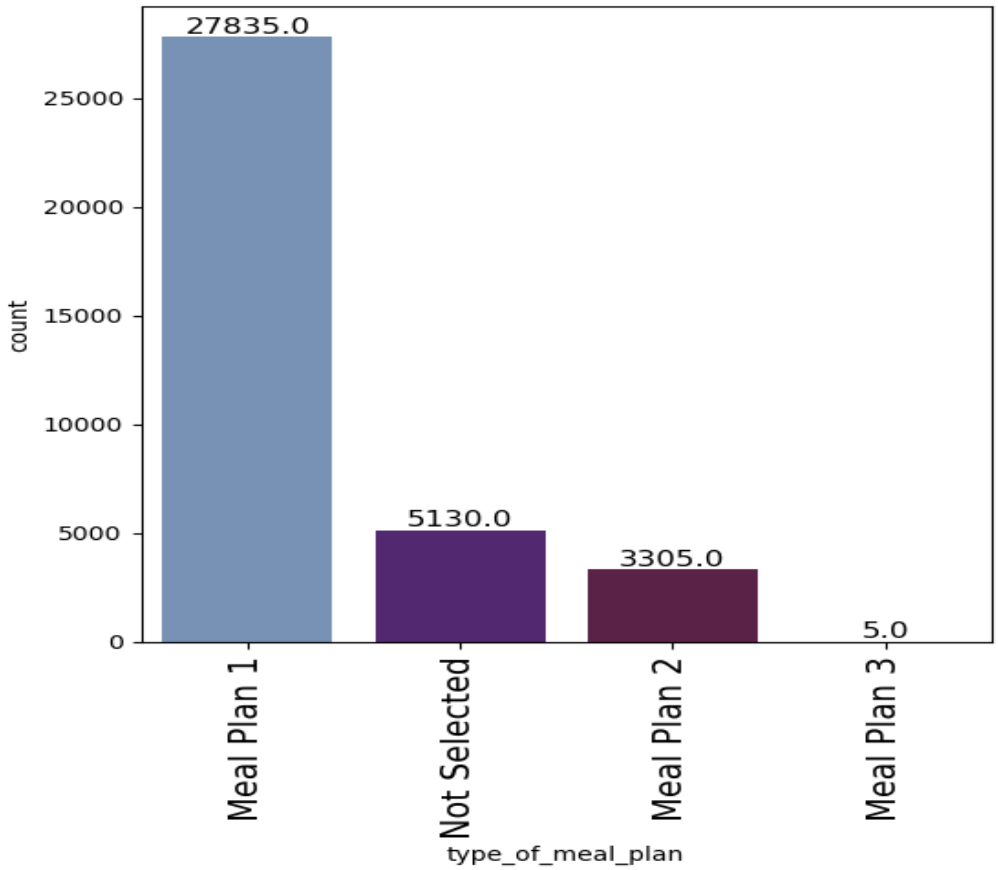


Figure 11 barplot of Type_of_Meal_Plan

Observation:

- Meal Plan 1 was selected for 27,835 bookings.
- 5,130 bookings did not select any meal plan (Not Selected).
- 3,305 bookings were made with Meal Plan 2.
- 5 bookings chose Meal Plan 3.

The vast majority of bookings opted for Meal Plan 1, with significantly fewer bookings either not selecting a meal plan or choosing Meal Plan 2. Meal Plan 3 had an extremely low selection rate.

4.1.10 REQUIRED_CAR_PARKING_SPACE

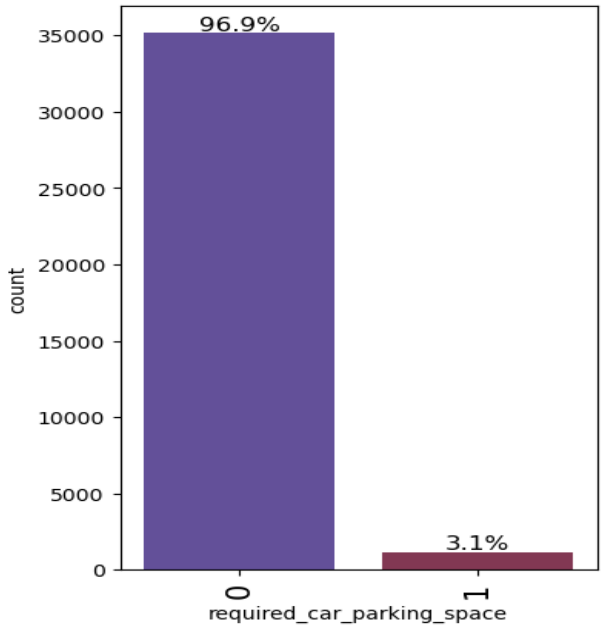


Figure 12 Barplot of Required_Car_Parking_Space

Observation:

- 96.9% bookings did not require a car parking space.
- only 3.1 bookings required a car parking space.

4.1.11 ROOM_TYPE_RESERVED

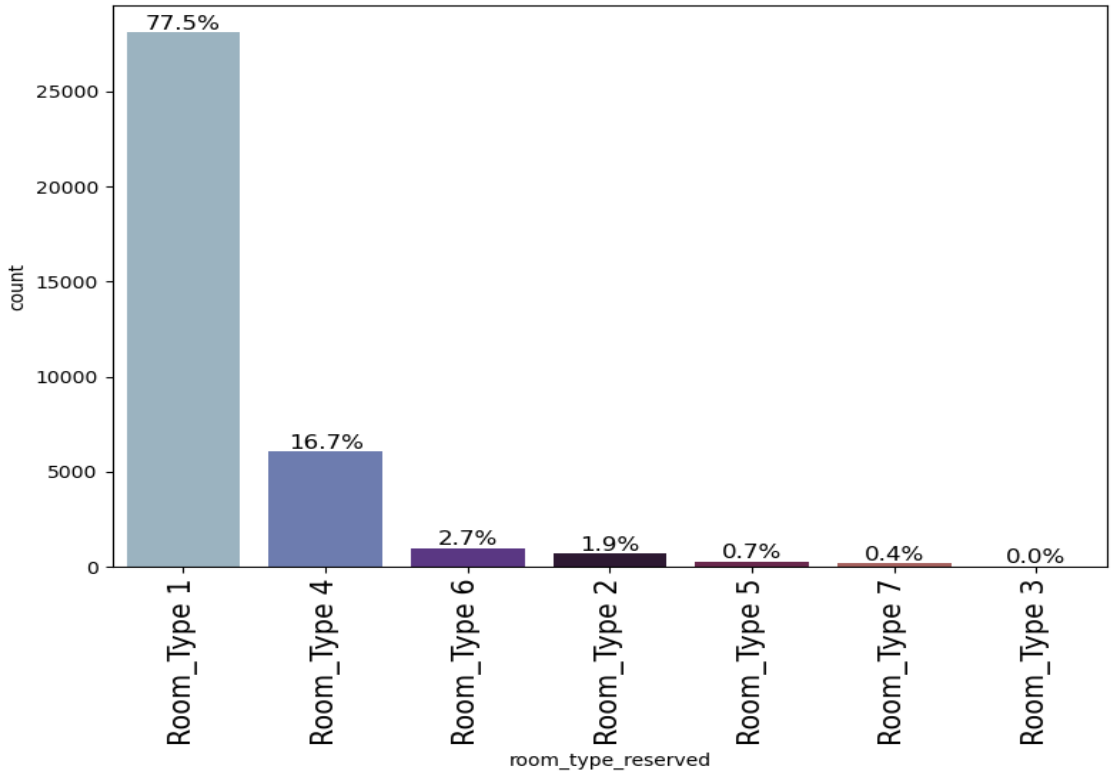


Figure 13 Barplot of Room_Type_Reserved

Observation:

- 77.5% of the bookings were made for room type 1
- 16.7% of the bookings were made for room type 4
- 2.7 % of the bookings were made for room type 6
- 1.9 % of the bookings were made for room type 2
- 0.7 % of the bookings were made for room type 5
- 0.4 % of the bookings were made for room type 7
- No bookings were made for room type 3

The majority of bookings, 77.5%, were made for Room Type 1

4.1.12 BOOKING STATUS

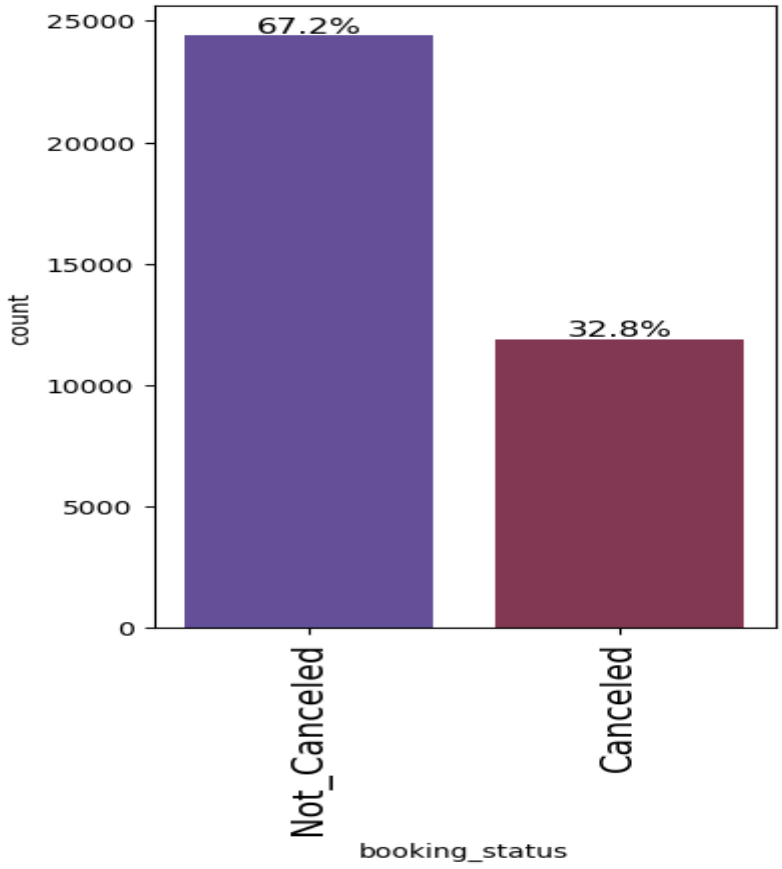


Figure 14 Barplot of Booking Status

Observation:

- 67.2% of bookings were not cancelled while 32.8% of bookings were cancelled

4.1.13 REPEATED_GUEST

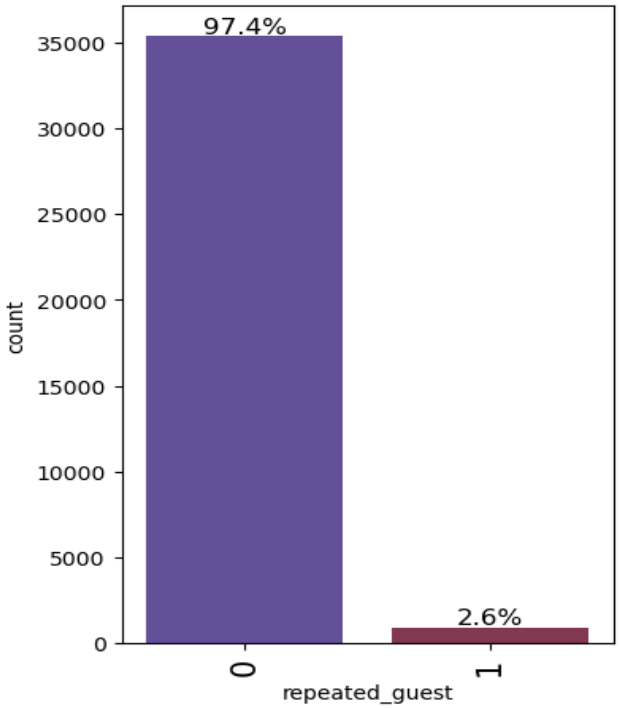


Figure 15 Barplot of Repeated_Guest

Observation:

- Majority of bookings, accounting for 97.4%, are made by new guests, while only 2.6% of bookings come from repeated guests.

4.1.14 NO_OF_PREVIOUS_CANCELLATIONS

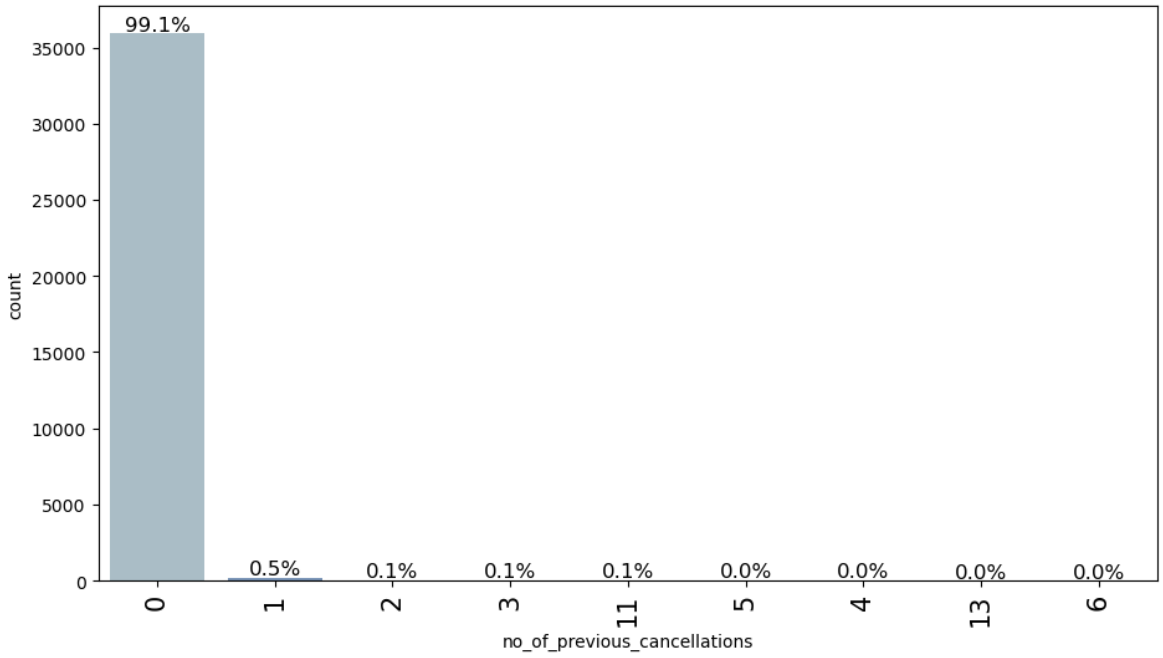


Figure 16 Barplot of No_of_Previous_Cancellations

Observation:

- 99.1% of bookings had no previous cancellations, while only 0.5% of bookings had a single previous cancellation

4.2 Bivariate Analysis

4.2.1 MARKET SEGMENT TYPE Vs AVERAGE PRICE PER ROOM

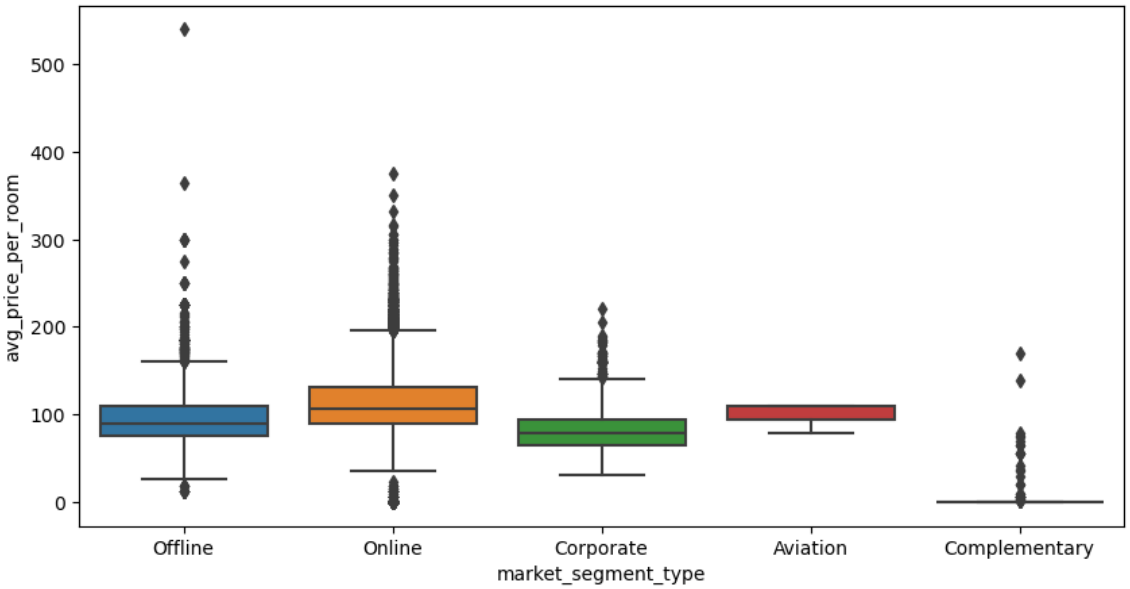


Figure 17 Boxplot of Market Segment Type Vs Average Price Per Room

Observation:

- Online booking is the highest in room price.
- Aviation, Offline, and Corporate are generally slightly lower priced with Corporate edging out for the lowest.
- Complimentary are of course free.

4.2.2 REPEATED_GUEST VS BOOKING_STATUS

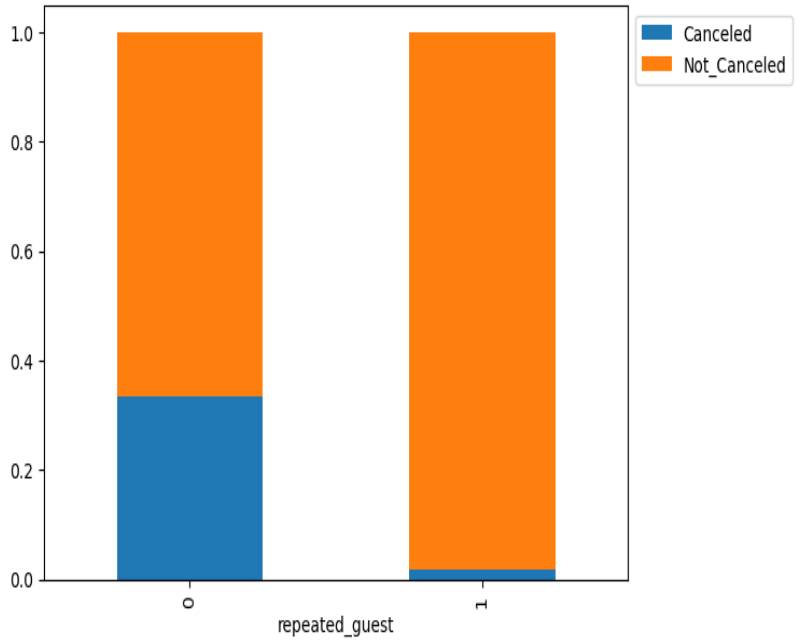


Figure 18 Stacked Barplot of Repeated_Guest Vs Booking_Status

Observation:

- Most repeat guests do not cancel their bookings, indicating a potential source of brand loyalty among these guests.
- Only 16 repeat guest have cancelled their bookings.

4.2.3 NO_OF_SPECIAL_REQUESTS VS BOOKING_STATUS

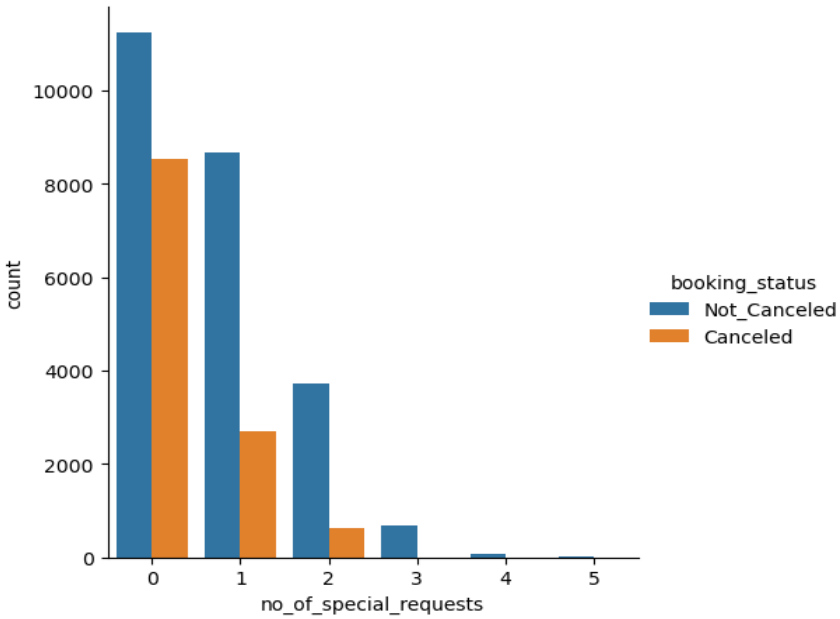


Figure 19 Catplot of No_of_Special_Requests Vs Booking_Status

Observation:

- The absence of special request increases the likelihood of cancellation.
- The addition of special request begins to reduce the likelihood of cancellation at one and progressively reduces cancellation to Zero on the instance of a third request.

4.2.4 MARKET_SEGMENT_TYPE VS BOOKING_STATUS

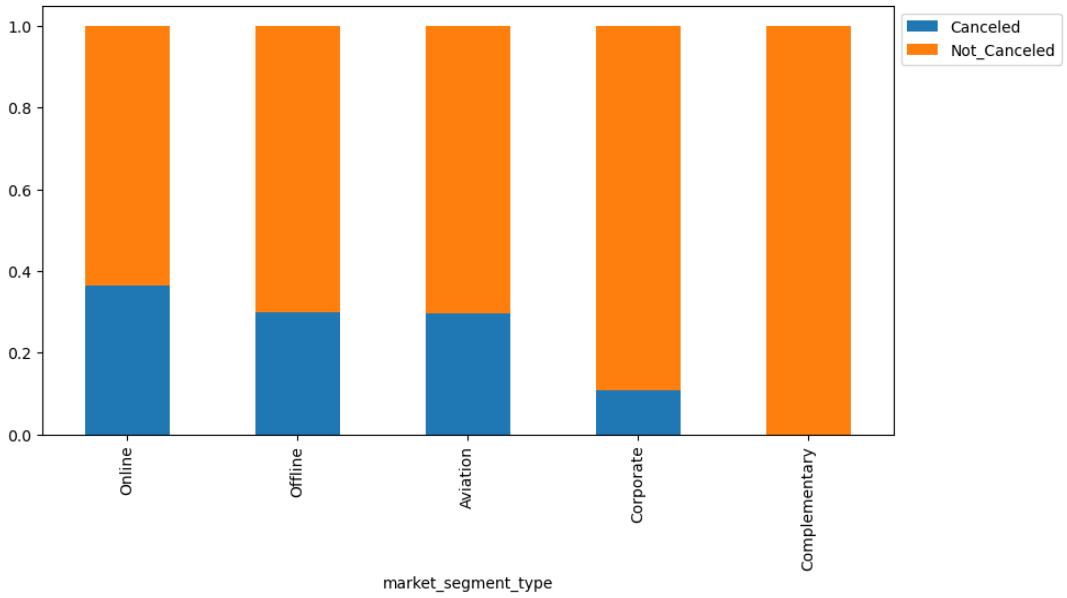


Figure 20 Stacked Barplot of Market_Segment_Type Vs Booking_Status

Observation:

- The majority of bookings come from the **online market segment** with **23,214** total bookings.
- The online and offline market segments dominate the total bookings.
- Corporate and aviation bookings have lower cancellation rates compared to online and offline bookings.
- Complementary bookings have a 0% cancellation rate, likely because they are provided as a benefit or bonus.

4.2.5 NO_OF_SPECIAL_REQUESTS VS BOOKING_STATUS

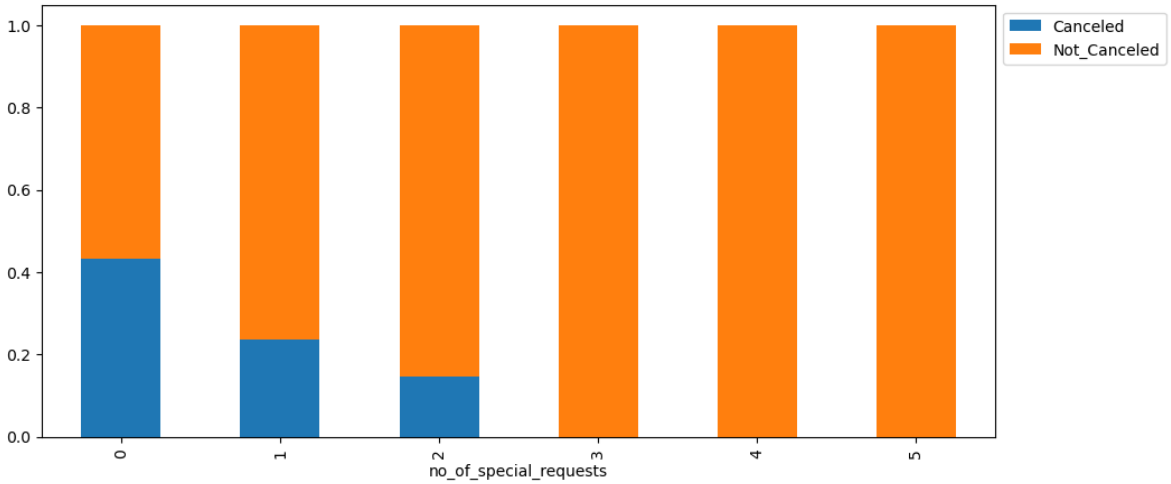


Figure 21 Stacked Barplot of No_of_Special_Requests Vs Booking_Status

Observation:

- The absence of special request increases the likelihood of cancellation.
- The addition of special request begins to reduce the likelihood of cancellation at one and progressively reduces cancellation to Zero on the instance of a third request.

4.2.6 BOXPLOT OF NO_OF_SPECIAL_REQUESTS VS AVG_PRICE_PER_ROOM

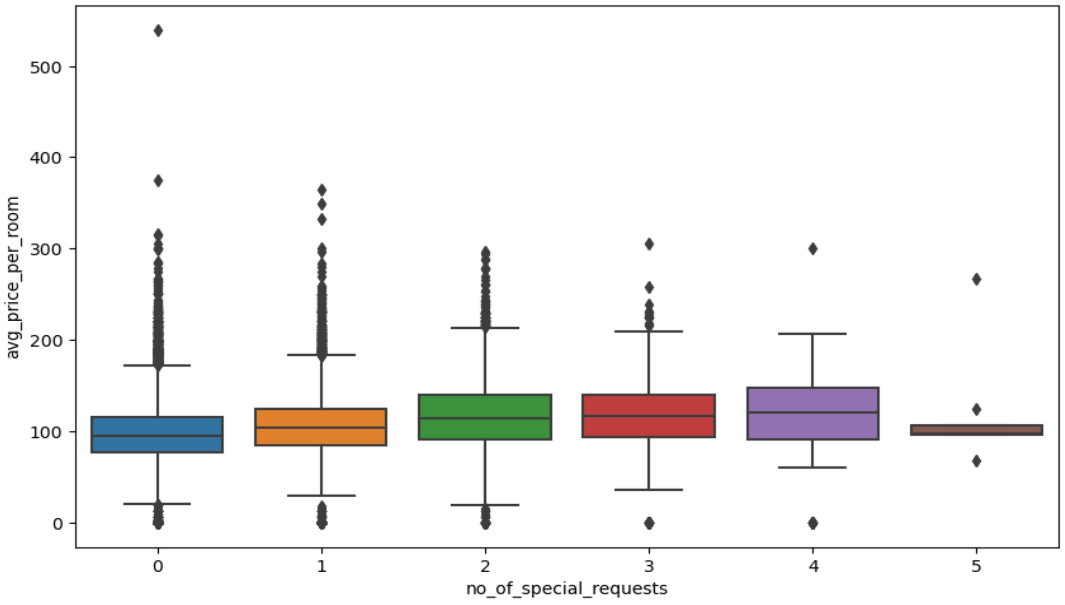


Figure 22 Boxplot of No_of_Special_Requests Vs Avg_Price_Per_Room

Observation:

- The price of the room does not seem to vary too much if there is little requests given, but when there is about 4 or more special requests the price seems to suffer variation.

4.2.7 ARRIVAL_MONTH VS BOOKING_STATUS

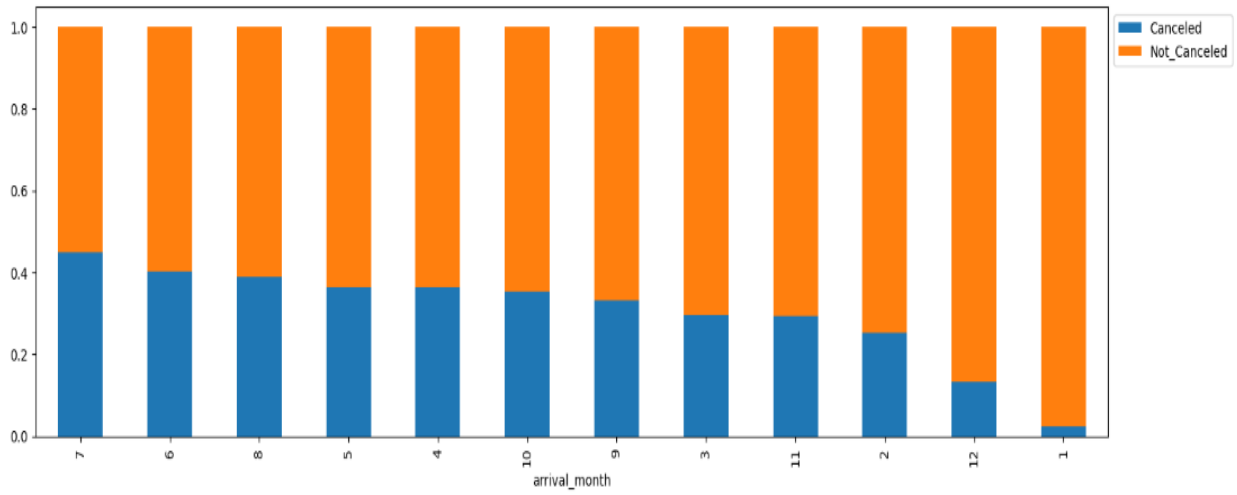


Figure 23 Stacked Barplot of Arrival_Month Vs Booking_Status

Observation:

- The highest cancellation rates are observed in July (45.0%) and June (40.3%).
- The lowest cancellation rate is in January (2.4%) and December (13.3%).
- Overall, there is significant variation in cancellation rates across different months, with summer months showing higher rates compared to winter months.

4.2.8 ARRIVAL_MONTH VS AVG_PRICE_PER_ROOM

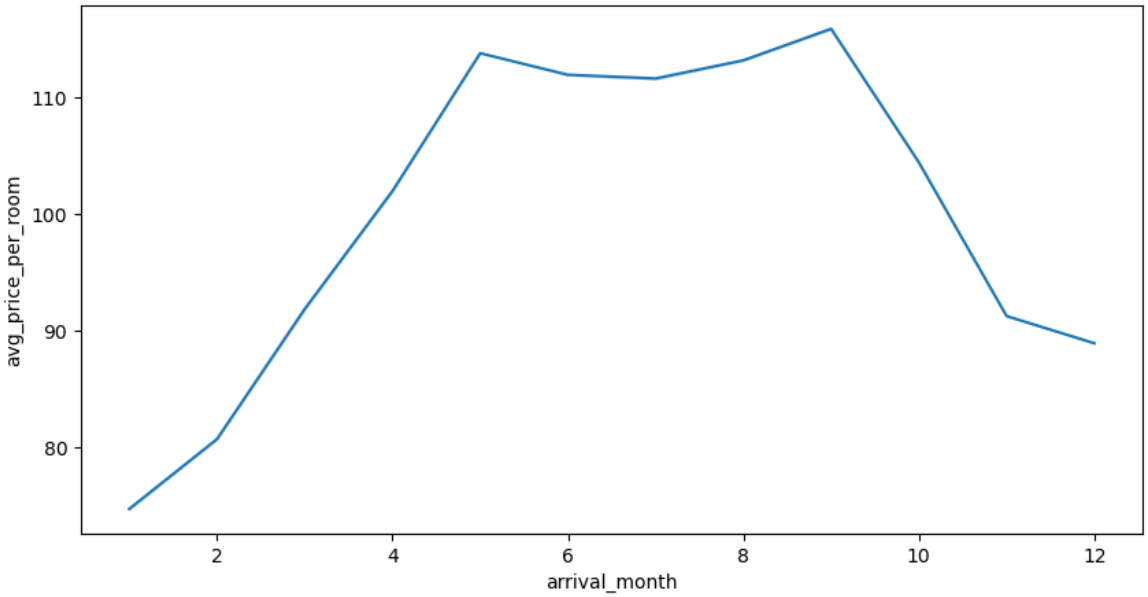


Figure 24 Line plot of Arrival_Month Vs Avg_Price_Per_Room

Observation: The prices are the highest in the summer, since most people go on vacation then it peaks at around October.

4.2.9 LEAD_TIME VS BOOKING_STATUS

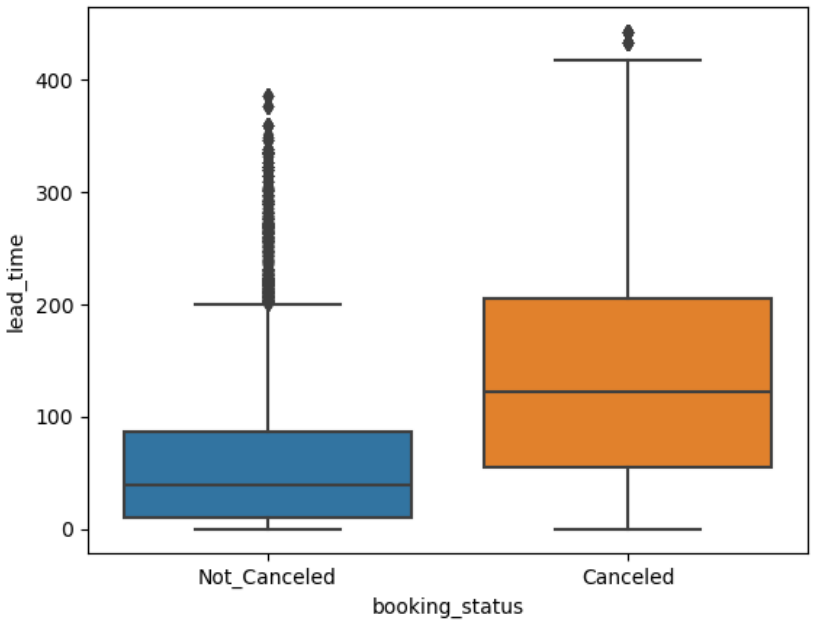
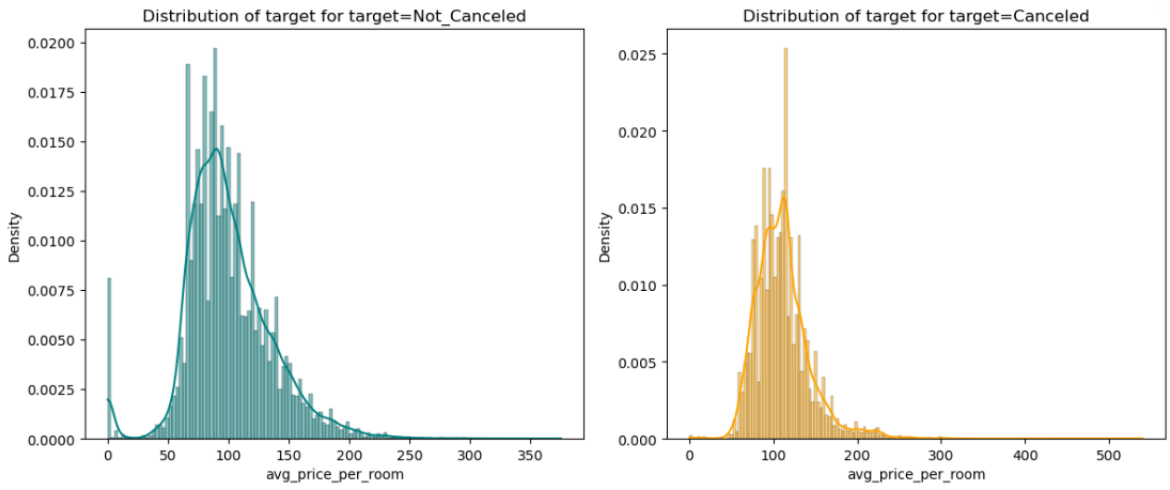


Figure 25 Boxplot of Lead_Time Vs Booking_Status

Observation:

An increase in lead time is observed to significantly influence the likelihood of booking cancellations, with longer lead times generally associated with higher cancellation rates.

4.2.10 AVG_PRICE_PER_ROOM VS BOOKING_STATUS



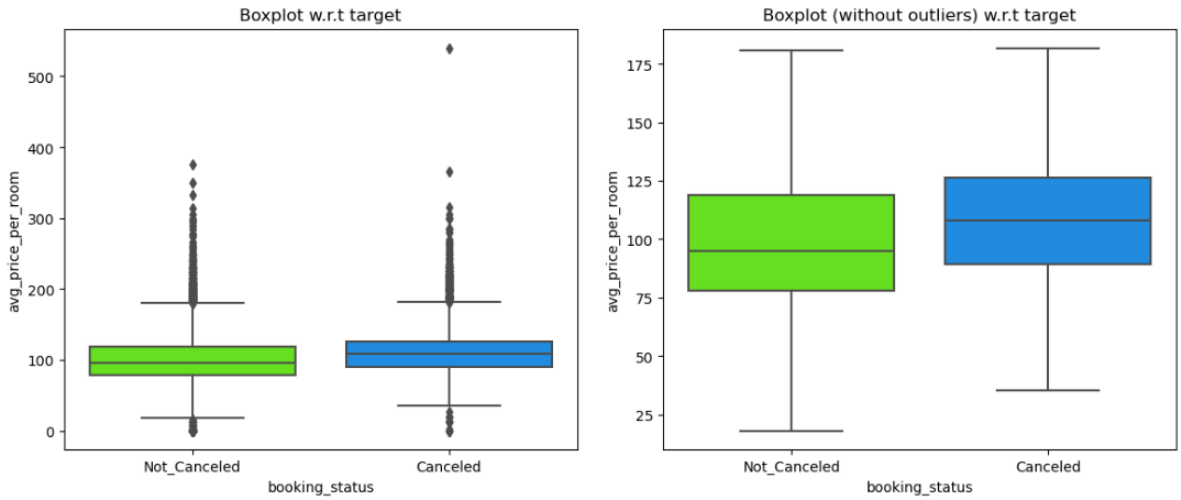


Figure 26 Distribution_Plot_Wrt_Target of Avg_Price_Per_Room Vs Booking_Status

Observation:

1.Histogram (Top plot)

1. **Not Canceled Bookings**:
    - The average price per room peaks around a lower price range (approximately 100-150).
    - The density gradually decreases as the price increases, indicating that most non-canceled bookings are at lower price points.
2. **Canceled Bookings**:
    - The peak density is slightly higher than for non-canceled bookings, suggesting a higher frequency of cancellations at certain price points.
    - The distribution is more spread out, with a noticeable number of cancellations even at higher price points (up to 500).

Insights:

- **Price Sensitivity**: Lower-priced rooms tend to have fewer cancellations, possibly indicating that customers are more likely to keep their bookings when the cost is lower.
- **Higher Price Cancellations**: There is a significant number of cancellations at higher price points, which could be due to various factors such as budget constraints or changes in travel plans.

Box plot (Bottom Plot)

1. **With Outliers**:
    - **Not Canceled**: The median price is lower, and there are numerous outliers, indicating a wide range of prices.
    - **Canceled**: The median price is higher, with many outliers, suggesting cancellations occur across a broader price range.

2. **Without Outliers**:
    - **Not Canceled**: The interquartile range (IQR) is more compact, showing most bookings fall within a narrower price range.
    - **Canceled**: The IQR is wider, indicating more variability in the prices of canceled bookings.

Insights:

- **Price Influence**: Higher average prices per room seem to correlate with a higher likelihood of cancellation.
- **Variability**: Canceled bookings show more price variability, suggesting that cancellations are influenced by a range of factors, not just price.

4.3 MULTIVARIATE ANALYSIS

4.3.1 RELATIONSHIP AMONG NUMERICAL VARIABLES – HEAT MAP

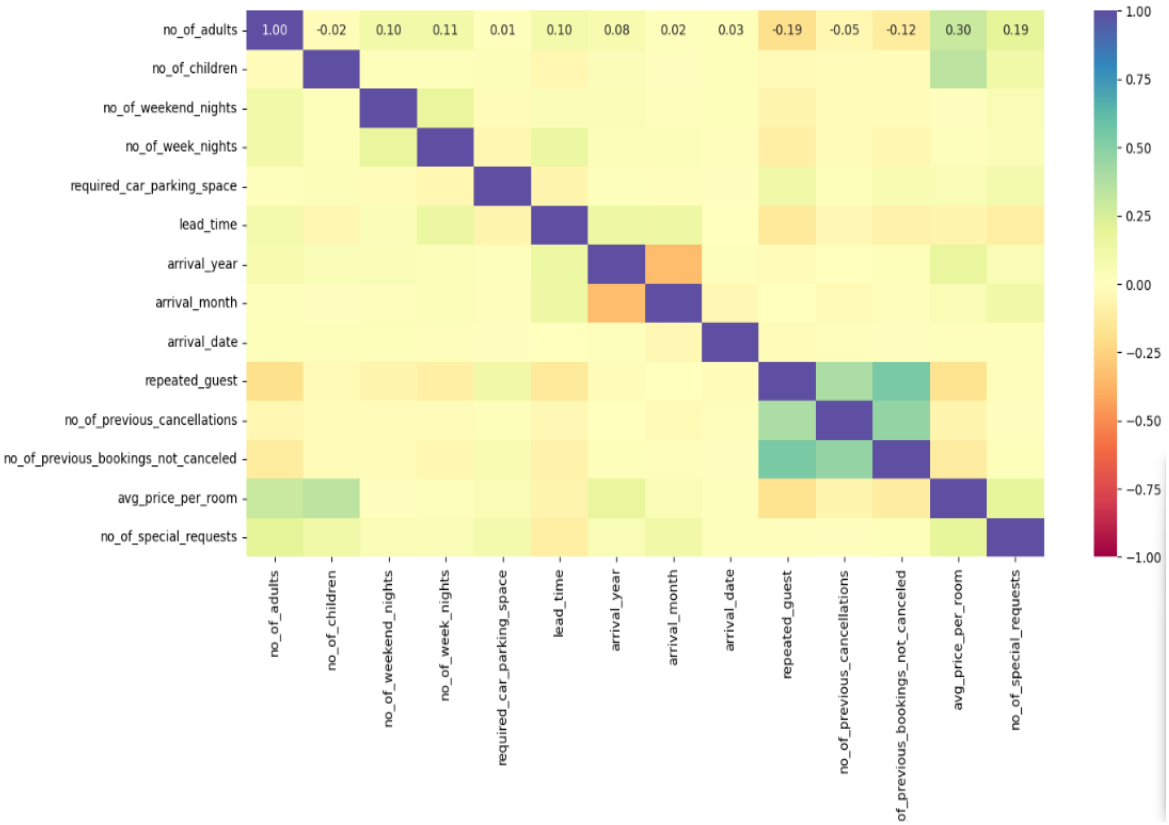


Figure 27 Heatmap for Numerical Data

Observation:

- There are high correlation between repeated guest and no.of previous bookings not cancelled.
- This shows repeat guests do not cancel their bookings, indicating a potential source of brand loyalty among these guests.
- The correlation between the number of previous cancellations and the number of previous bookings that were not canceled is 0.47, indicating a moderate positive relationship between these two variables.

4.3.2 PAIR PLOT FOR NUMERIC DATA

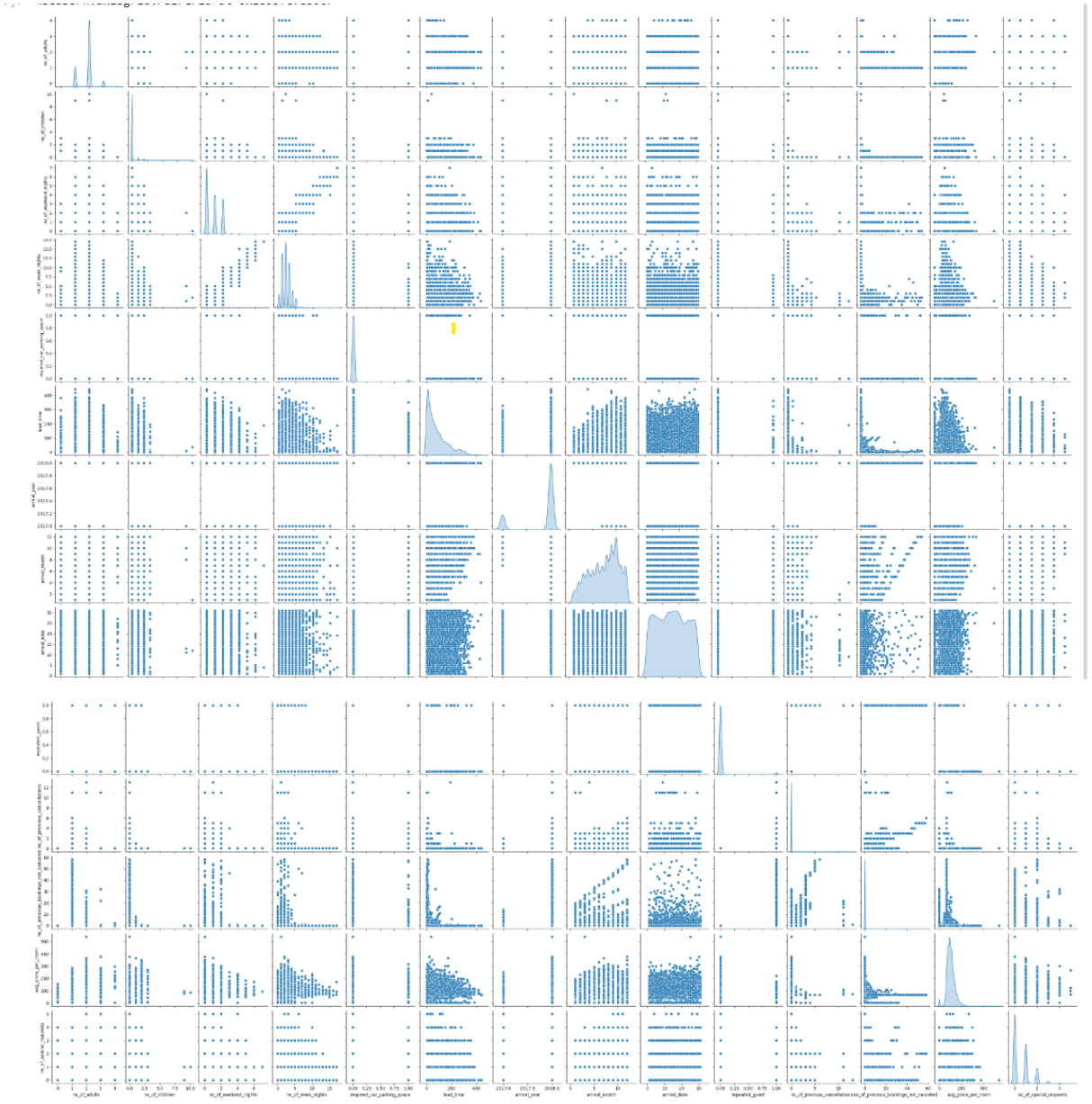


Figure 28 Pair Plot for Numeric Data

5. QUESTIONS

1. What are the busiest months in the hotel?

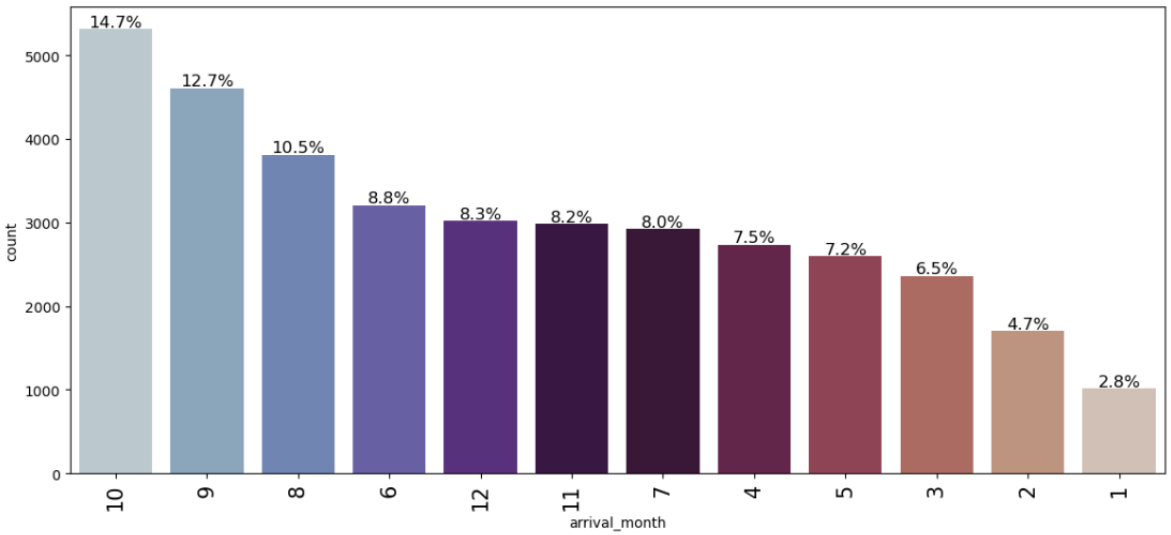


Figure 29 Distribution of Arrival Month

Observation:

- From this graph we can say, October month is the busiest months.
- It has 14.7 percentage.
- It has 5317 counts.

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

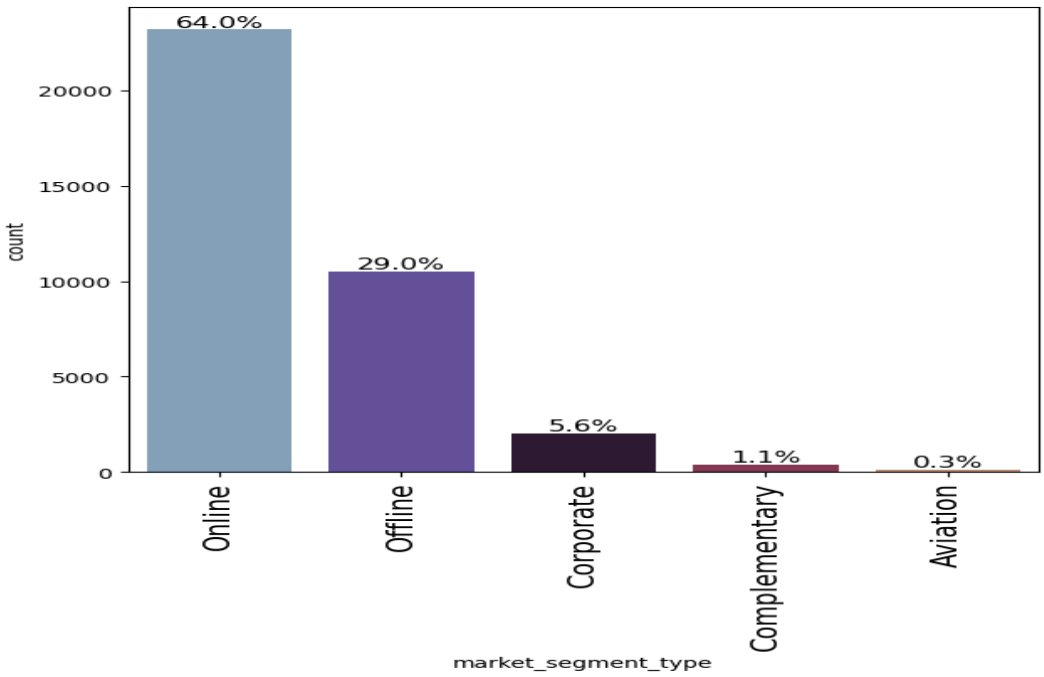


Figure 30 Barplot of Market Segment Type

Observation:

- Online has the highest count 23214
- 64% of the bookings come via the internet.
- The **online** and **offline** market segments dominate the total bookings.
- **Corporate** and **aviation** bookings have lower cancellation rates compared to online and offline bookings.
- **Complementary bookings** have a **0% cancellation rate**, likely because they are provided as a benefit or bonus.

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

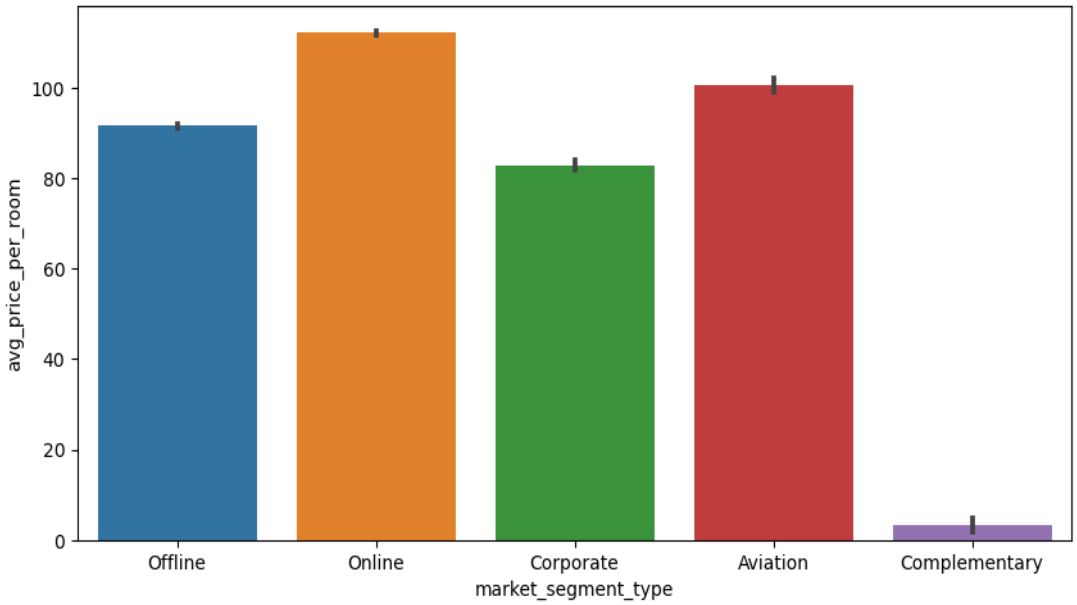


Figure 31 Barplot of Market Segment Vs Avg Price Per Room

Observation:

- Online booking is the highest in room price.
- Aviation, Offline, and Corporate are generally slightly lower priced with Corporate edging out for the lowest.
- Complimentary are of course free.

4. What percentage of bookings are canceled?

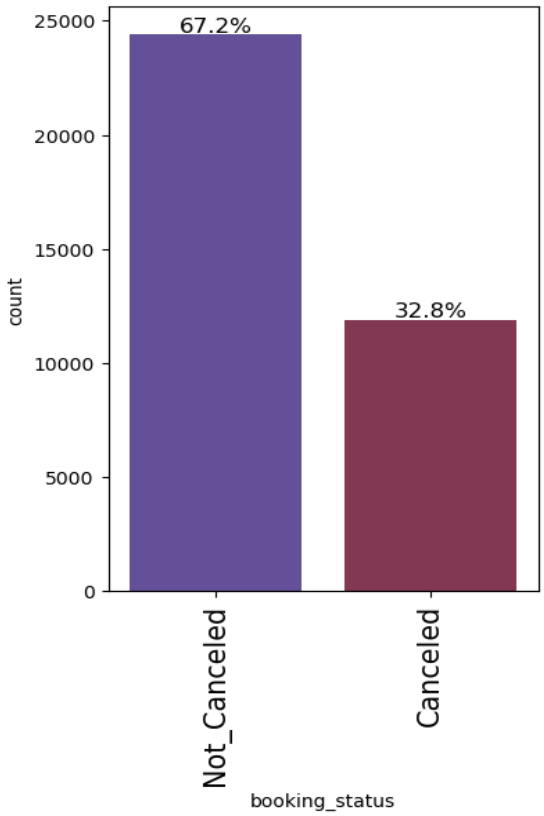


Figure 32 Barplot of Booking Status

Observation:

- 32.8 percentage bookings got cancelled.
- About 1/2 (11885) of bookings are cancelled in the given data.

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

booking_status	Canceled	Not_Canceled	All
repeated_guest			
All	11885	24390	36275
0	11869	23476	35345
1	16	914	930

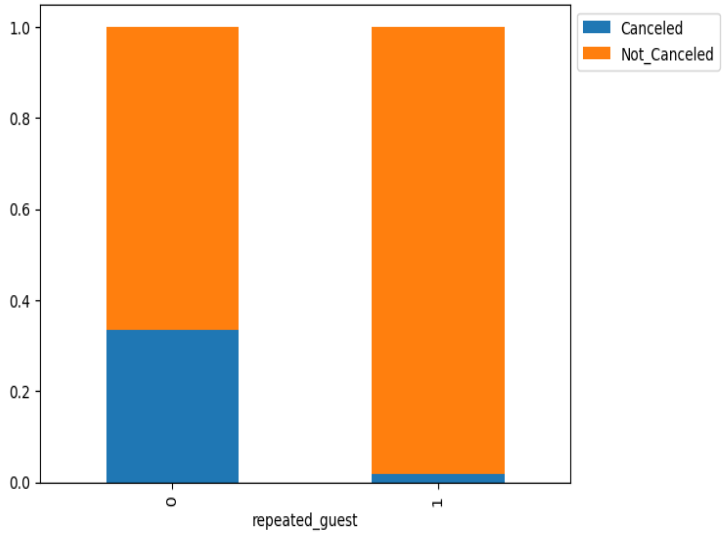


Figure 33 Stacked_Barplot of Repeated_Guest Vs Booking_Status

Observation:

- Only 16 repeated guest have cancelled the bookings.
- 1.72% of repeating guest rarely cancel the bookings.

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

booking_status	Canceled	Not_Canceled	All
no_of_special_requests			
All	11885	24390	36275
0	8545	11232	19777
1	2703	8670	11373
2	637	3727	4364
3	0	675	675
4	0	78	78
5	0	8	8

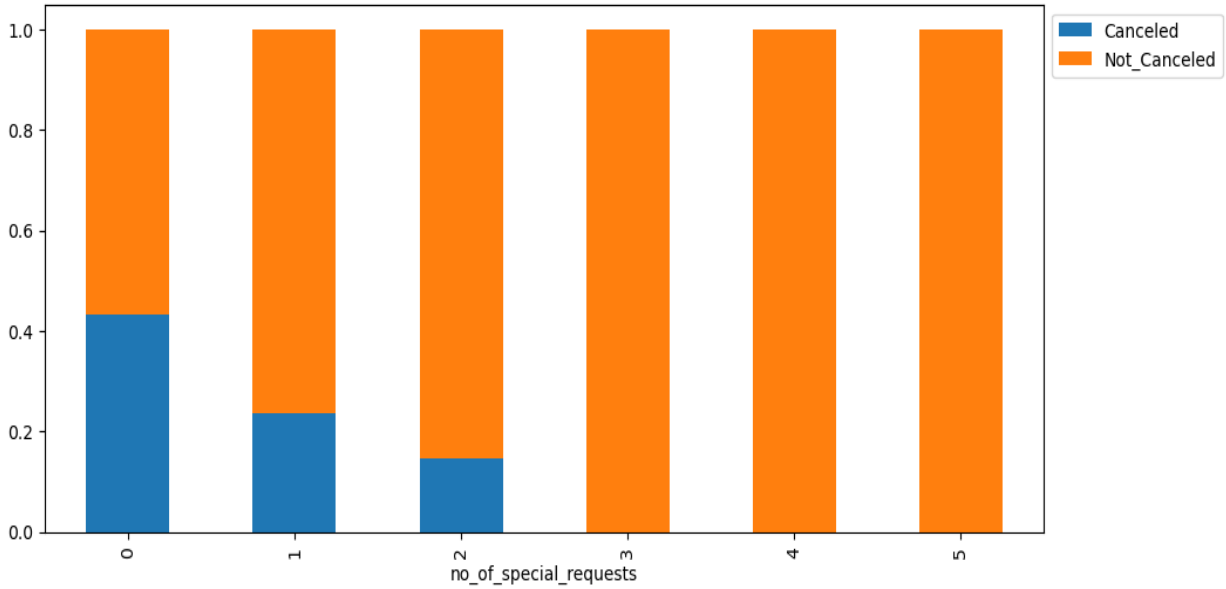


Figure 34 Stacked_Barplot No_of_Special_Requests Vs Booking_Status

Observation:

- The absence of special request increases the likelihood of cancellation, the addition of special request begins to reduce the likelihood of cancellation at one and progressively reduces cancellation to Zero on the instance of a third request.

Yes, special requirements affect the booking cancellation.

6. KEY MEANINGFUL OBSERVATIONS ON INDIVIDUAL VARIABLES AND THE RELATIONSHIP BETWEEN VARIABLES.

1. Do special requests impact the prices of the room?

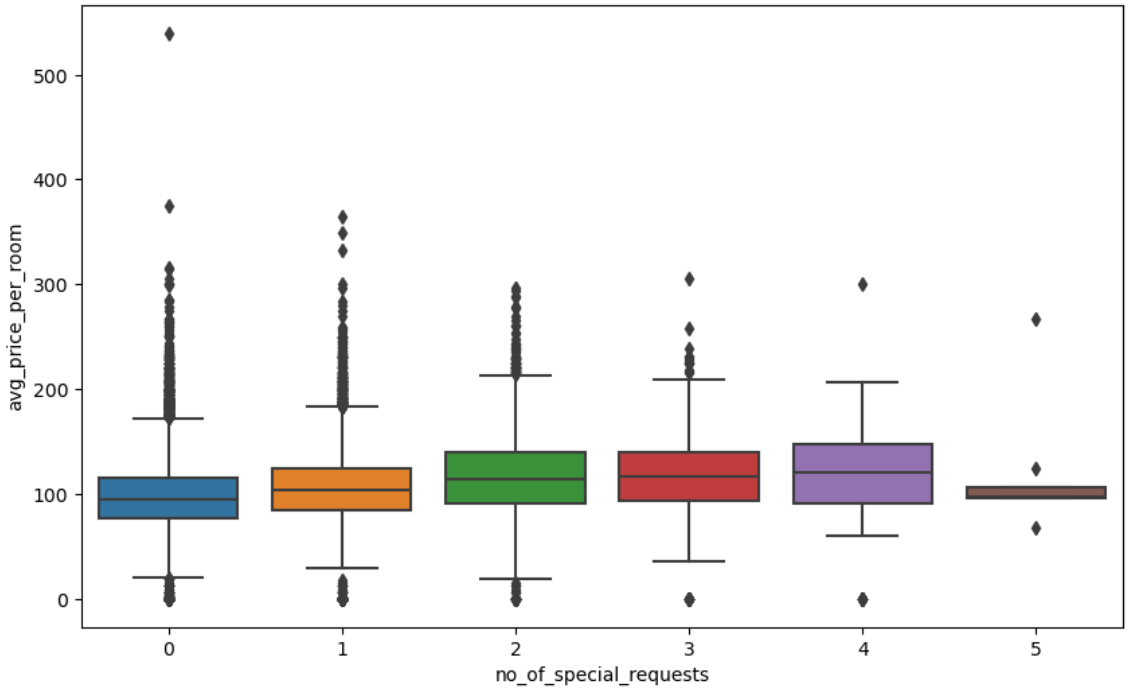
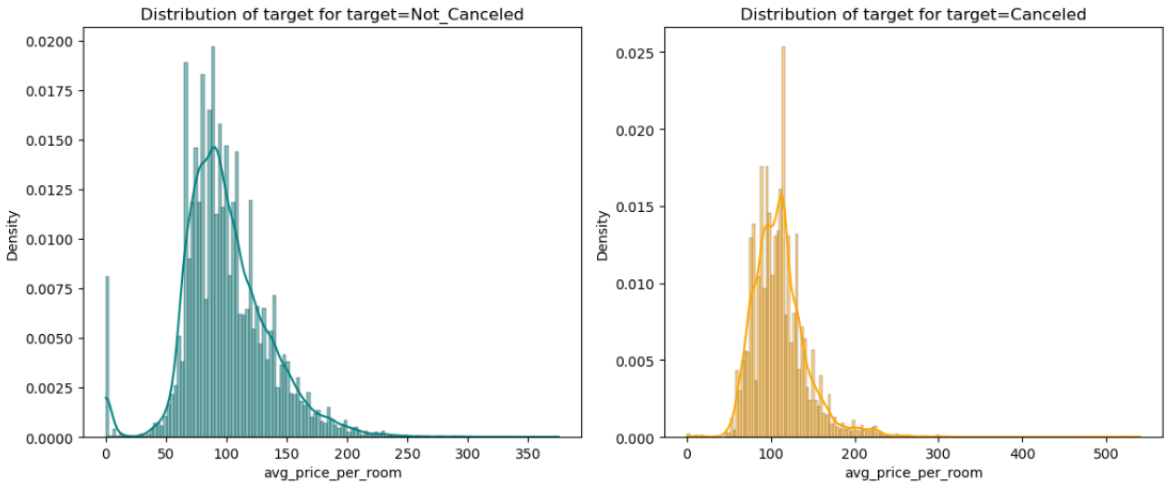


Figure 35 boxplot of no_of_special_requests vs avg_price_per_room

Observation:

The price of the room does not seem to vary too much if there is little requests given, but when there is about 4 or more special requests the price seems to suffer variation

2. How does price per room impact booking status?



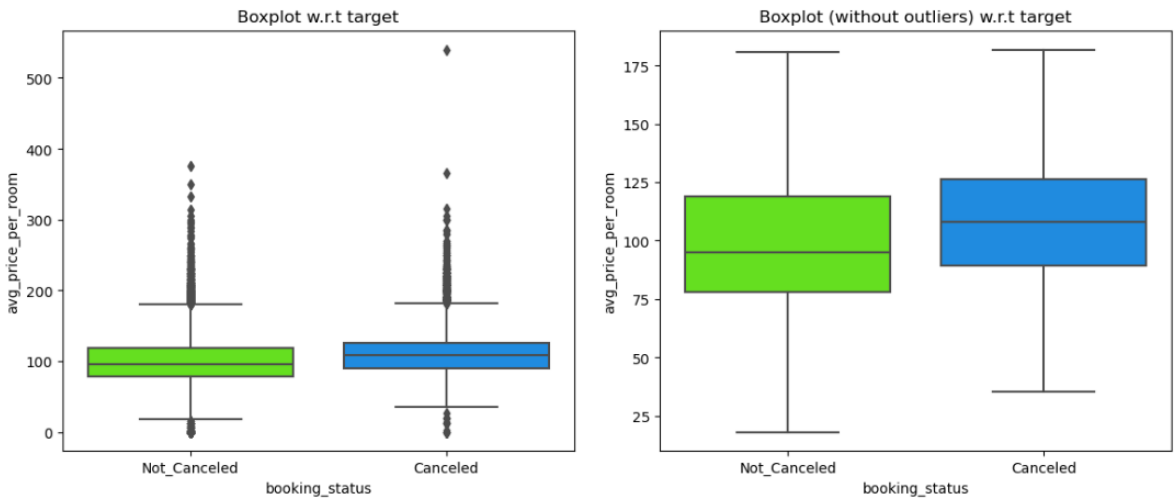


Figure 36 Distribution_Plot_Wrt_Target of Avg_Price_Per_Room Vs Booking_Status

Observation:

1.histogram (Top plot)

3. **Not Canceled Bookings**:
    - The average price per room peaks around a lower price range (approximately 100-150).
    - The density gradually decreases as the price increases, indicating that most non-canceled bookings are at lower price points.
4. **Canceled Bookings**:
    - The peak density is slightly higher than for non-canceled bookings, suggesting a higher frequency of cancellations at certain price points.
    - The distribution is more spread out, with a noticeable number of cancellations even at higher price points (up to 500).

Insights:

- **Price Sensitivity**: Lower-priced rooms tend to have fewer cancellations, possibly indicating that customers are more likely to keep their bookings when the cost is lower.
- **Higher Price Cancellations**: There is a significant number of cancellations at higher price points, which could be due to various factors such as budget constraints or changes in travel plans.

2.Box plot(Bottom Plot)

1.With Outliers:

- **Not Canceled**: The median price is lower, and there are numerous outliers, indicating a wide range of prices.
- **Canceled**: The median price is higher, with many outliers, suggesting cancellations occur across a broader price range.

2.Without Outliers:

- **Not Canceled**: The interquartile range (IQR) is more compact, showing most bookings fall within a narrower price range.
- **Canceled**: The IQR is wider, indicating more variability in the prices of canceled bookings.

Insights:

- **Price Influence**: Higher average prices per room seem to correlate with a higher likelihood of cancellation.
- **Variability**: Canceled bookings show more price variability, suggesting that cancellations are influenced by a range of factors, not just price.

3. How does lead time impact booking status?

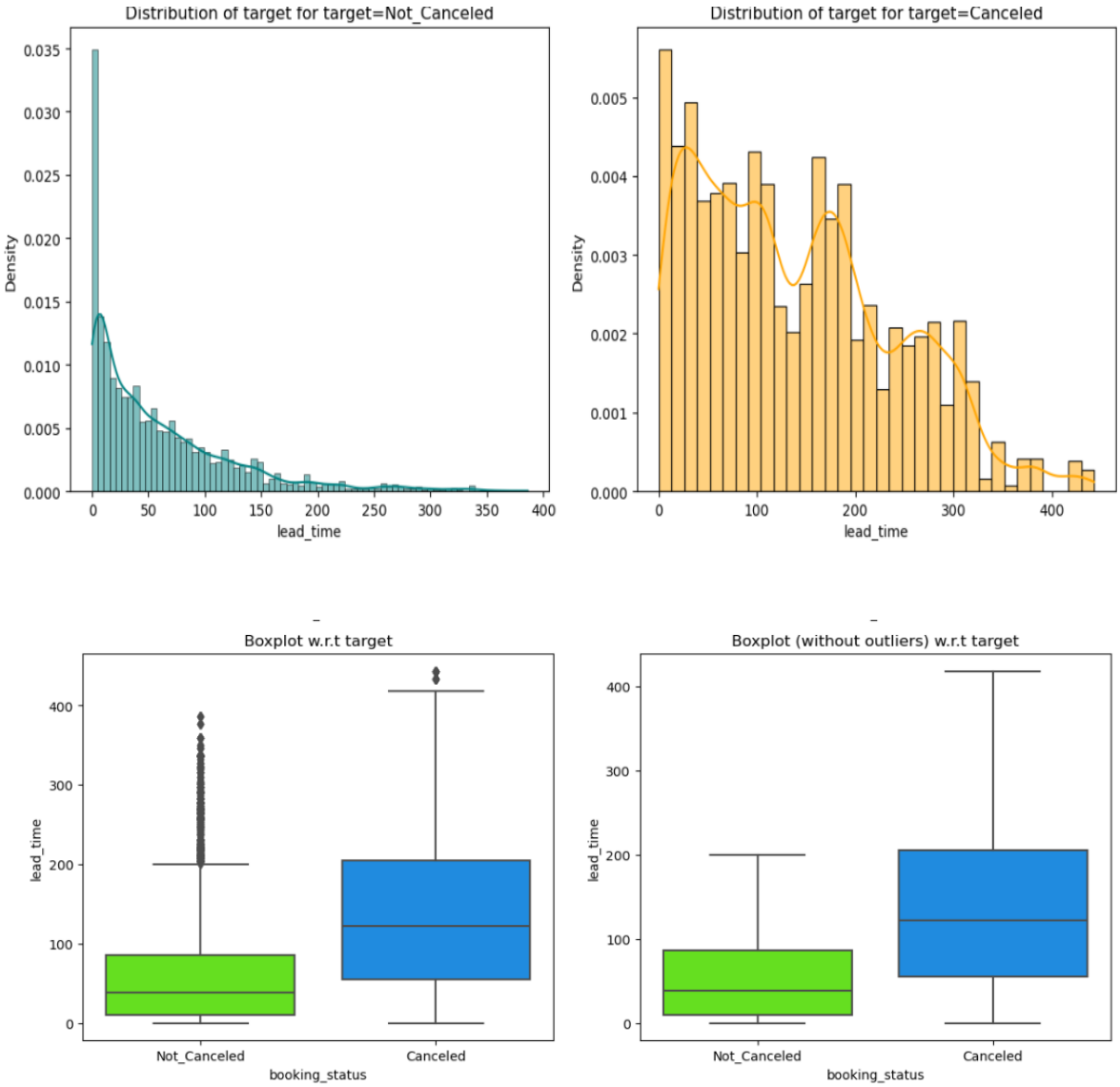


Figure 37 Distribution_Plot_Wrt_Target of Lead_Time Vs Booking_Status

Observation:

1.Histogram (Top Plot):

1. **Not Canceled Bookings**:
    - The distribution is right-skewed, with most bookings having shorter lead times.
    - The density decreases as the lead time increases, indicating that bookings made closer to the stay date are less likely to be canceled.
2. **Canceled Bookings**:
    - This distribution is also right-skewed but less steep, suggesting cancellations occur more evenly across different lead times.
    - There is a noticeable density even at higher lead times, indicating that bookings made well in advance are more prone to cancellation.

Insights:

- **Lead Time Impact**: Shorter lead times are associated with fewer cancellations, possibly because plans are more certain closer to the stay date.
- **Advance Bookings**: Higher lead times show a higher likelihood of cancellation, which could be due to changes in plans or other unforeseen circumstances.

2.Box Plot (Bottom Plot)

1. **Boxplot with Outliers**:
    - **'Canceled' Bookings**: There are numerous outliers, indicating that some bookings have significantly higher real-time values. This suggests variability and potential issues in the booking process for canceled reservations.
    - **'Not_Canceled' Bookings**: The distribution is more compact with fewer outliers, indicating more consistent real-time values.
    - **Median Comparison**: The median real-time for 'Canceled' bookings is higher than for 'Not_Canceled' bookings, suggesting that canceled bookings generally take longer.
2. **Boxplot without Outliers**:
    - **'Canceled' Bookings**: Without the outliers, the data shows a clearer picture. The median real-time remains higher for canceled bookings, but the range is more comparable to not canceled bookings.
    - **'Not_Canceled' Bookings**: The distribution remains consistent, showing that most bookings fall within a similar range of real-time values.
    - **Overall Comparison**: Removing outliers helps in understanding the core distribution of the data, highlighting that even without extreme values, canceled bookings tend to have higher real-time values.

4. Number of bookings cancelled each month

booking_status	Canceled	Not_Canceled	All
arrival_month			
All	11885	24390	36275
10	1880	3437	5317
9	1538	3073	4611
8	1488	2325	3813
7	1314	1606	2920
6	1291	1912	3203
4	995	1741	2736
5	948	1650	2598
11	875	2105	2980
3	700	1658	2358
2	430	1274	1704
12	402	2619	3021
1	24	990	1014

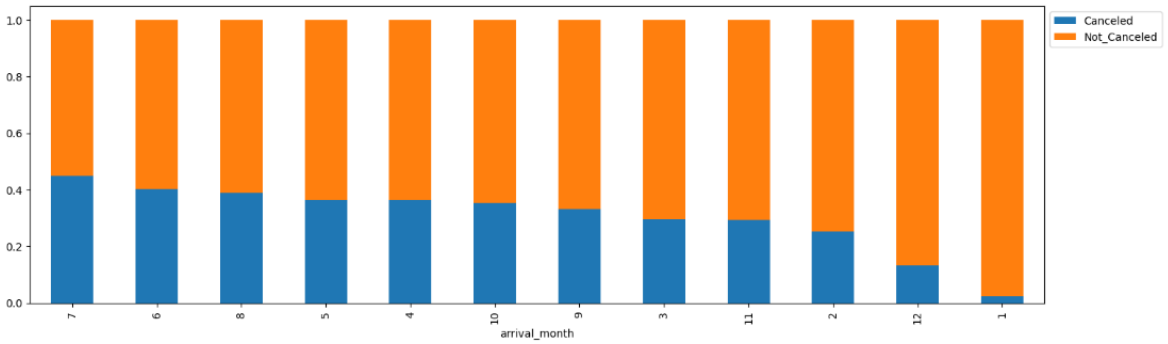


Figure 38 Stacked_Barplot of Arrival_Month Vs Booking_Status

Observation:

- July has the most percentage of cancellations and January has the least

5. Family_members vs booking_status

booking_status	0	1	All
no_of_family_members			
All	18456	9985	28441
2	15506	8213	23719
3	2425	1368	3793
4	514	398	912
5	10	5	15
11	0	1	1
12	1	0	1

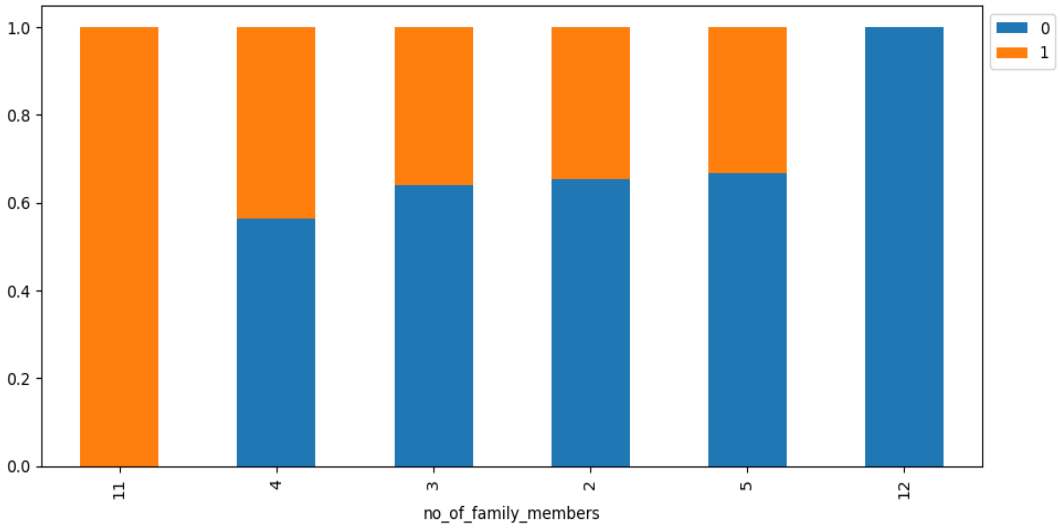


Figure 39 stacked barplot of no.of family members vs booking status

Observation:

- **Families with 2 members** make up the majority of the bookings (around **83%** of total bookings) and have a cancellation rate consistent with the overall average.
- **Families with 4 members** have the highest cancellation rate (**43.6%**), which may suggest that larger family trips face more uncertainty or risk of cancellation.
- The data for families with **5 or more members** is minimal and therefore doesn't allow for strong conclusions.

6. Arrival_month vs booking_status

booking_status	0	1	All
arrival_month			
All	24390	11885	36275
10	3437	1880	5317
9	3073	1538	4611
8	2325	1488	3813
7	1606	1314	2920
6	1912	1291	3203
4	1741	995	2736
5	1650	948	2598
11	2105	875	2980
3	1658	700	2358
2	1274	430	1704
12	2619	402	3021
1	990	24	1014

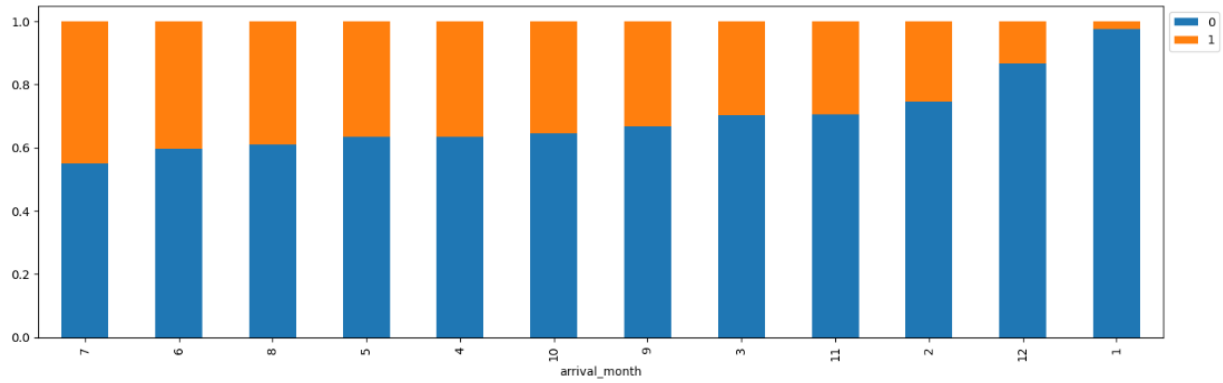


Figure 40 Stacked Barplot of No.of Family Members Vs Booking Status

Observation:

- **December and January** have the **lowest cancellation rates** (~13.3% and 2.4%, respectively), indicating these are more stable months for bookings. This may be due to holidays and higher travel commitment.
- **July and August** have **higher cancellation rates** (45% and 39%, respectively), suggesting these months experience more uncertainty in travel plans.
- **October and September** are the peak months in terms of booking volume, with **October** having the highest total bookings but a moderate cancellation rate.

7. Repeated_Guest Vs Booking_Status

booking_status	0	1	All
repeated_guest			
All	24390	11885	36275
0	23476	11869	35345
1	914	16	930

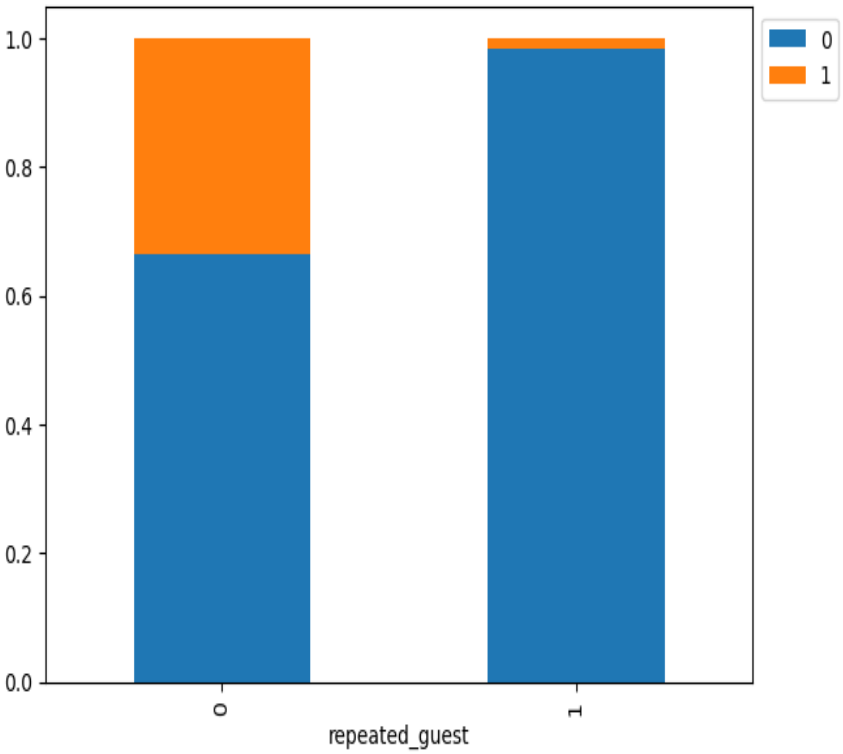
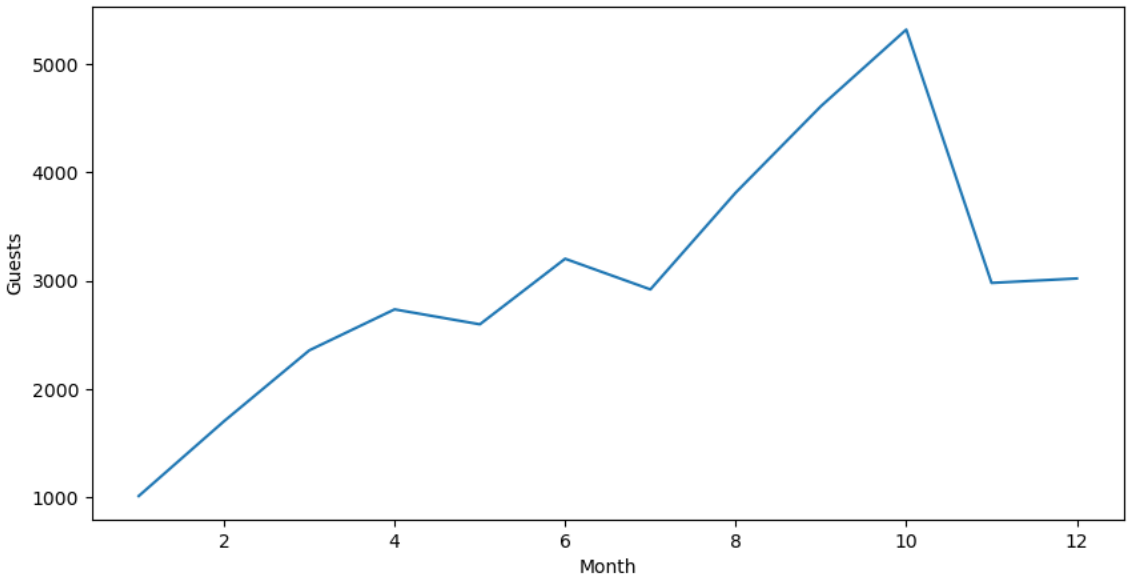


Figure 41 Stacked_Barplot Of Repeated_Guest Vs Booking_Status

- **Repeated guests** (those who have stayed before) are far more reliable in terms of keeping their bookings, with an **extremely low cancellation rate** (1.7%). This suggests that customer loyalty or satisfaction leads to more stable bookings.
- **Non-repeated guests** have a significantly higher cancellation rate (**33.6%**), indicating that first-time guests are more likely to cancel their bookings compared to returning customers.

8. Grouping the data on arrival months and extracting the count of bookings



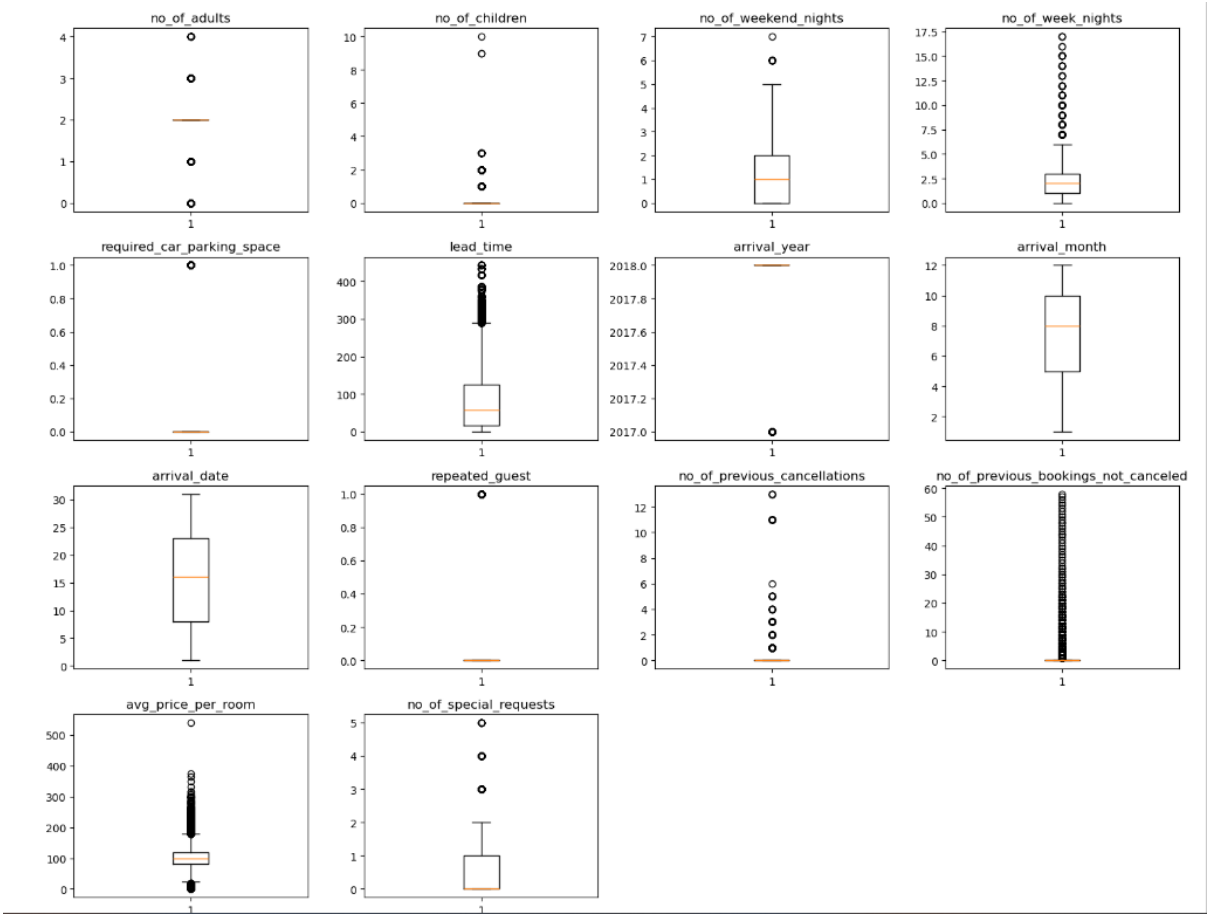
Observation:

- The line plot shows a peak in bookings during the month of October, indicating that the majority of bookings were made during this period.

7. DATA PREPROCESSING

- There are no duplicated values in the data
- There are no missing values in the data
- Outliers are detected.

7.1 Outlier Detection



Observation:

Since these values represent genuine outliers, we have chosen not to remove them from the analysis, as they reflect real-world booking behavior that could provide important insights for understanding customer patterns and trends.

7.2 Feature Engineering

- For the column booking status, replaced the not cancelled value with '***0***' and cancelled value with '1'.
- Created dummy variables for the columns mainly having datatype object and category.
- Splatted the data y with booking status and x with rest of the columns.
- Splitting the data in 70:30 ratio for train to test data.

```
Shape of Training set :  (25392, 27)
Shape of test set :  (10883, 27)
Percentage of classes in training set:
booking_status
0   0.67064
1   0.32936
Name: proportion, dtype: float64
Percentage of classes in test set:
booking_status
0   0.67638
1   0.32362
Name: proportion, dtype: float64
```

Figure 42 Splitting data in train and test sets

- We had seen that around 67% of observations belongs to class 0 (Not canceled) and 32% observations belongs to class 1 (canceled), and this is preserved in the train and test sets

8. LOGISTIC REGRESSION

8.1 Model Building - Logistic Regression

- We will now perform logistic regression using statsmodels, a Python module that provides functions for the estimation of many statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Using statsmodels, we will be able to check the statistical validity of our model - identify the significant predictors from p-values that we get for each predictor variable.

Logit Regression Results

Dep. Variable:	booking_status	No. Observations:	25392
Model:	Logit	Df Residuals:	25364
Method:	MLE	Df Model:	27
Date:	Sat, 07 Sep 2024	Pseudo R-squ.:	0.3293
Time:	20:43:48	Log-Likelihood:	-10793.
converged:	False	LL-Null:	-16091.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-924.5923	120.817	-7.653	0.000	-1161.390	-687.795
no_of_adults	0.1135	0.038	3.017	0.003	0.040	0.187
no_of_children	0.1563	0.057	2.732	0.006	0.044	0.268
no_of_weekend_nights	0.1068	0.020	5.398	0.000	0.068	0.146
no_of_week_nights	0.0398	0.012	3.239	0.001	0.016	0.064
required_car_parking_space	-1.5939	0.138	-11.561	0.000	-1.864	-1.324
lead_time	0.0157	0.000	58.868	0.000	0.015	0.016
arrival_year	0.4570	0.060	7.633	0.000	0.340	0.574
arrival_month	-0.0415	0.006	-6.418	0.000	-0.054	-0.029
arrival_date	0.0005	0.002	0.252	0.801	-0.003	0.004
repeated_guest	-2.3469	0.617	-3.805	0.000	-3.556	-1.138
no_of_previous_cancellations	0.2664	0.086	3.108	0.002	0.098	0.434
no_of_previous_bookings_not_canceled	-0.1727	0.153	-1.131	0.258	-0.472	0.127
avg_price_per_room	0.0188	0.001	25.404	0.000	0.017	0.020
no_of_special_requests	-1.4690	0.030	-48.790	0.000	-1.528	-1.410
type_of_meal_plan_Meal Plan 2	0.1768	0.067	2.654	0.008	0.046	0.307
type_of_meal_plan_Meal Plan 3	17.8379	5057.771	0.004	0.997	-9895.212	9930.888
type_of_meal_plan_Not Selected	0.2782	0.053	5.245	0.000	0.174	0.382
room_type_reserved_Room_Type 2	-0.3610	0.131	-2.761	0.006	-0.617	-0.105
room_type_reserved_Room_Type 3	-0.0009	1.310	-0.001	0.999	-2.569	2.567
room_type_reserved_Room_Type 4	-0.2821	0.053	-5.305	0.000	-0.386	-0.178
room_type_reserved_Room_Type 5	-0.7176	0.209	-3.432	0.001	-1.127	-0.308
room_type_reserved_Room_Type 6	-0.9456	0.147	-6.434	0.000	-1.234	-0.658
room_type_reserved_Room_Type 7	-1.3964	0.293	-4.767	0.000	-1.971	-0.822
market_segment_type_Complementary	-41.8798	8.42e+05	-4.98e-05	1.000	-1.65e+06	1.65e+06
market_segment_type_Corporate	-1.1935	0.266	-4.487	0.000	-1.715	-0.672
market_segment_type_Offline	-2.1955	0.255	-8.625	0.000	-2.694	-1.697
market_segment_type_Online	-0.3990	0.251	-1.588	0.112	-0.891	0.093

Figure 43 Logistic Regression Result

Observations

- Negative values of the coefficient show that the probability of Person Cancelling decreases with the increase of the corresponding attribute value.
- Positive values of the coefficient show that the probability of Person Cancelling increases with the increase of the corresponding attribute value.
- p-value of a variable indicates if the variable is significant or not. If we consider the significance level to be 0.05 (5%), then any variable with a p-value less than 0.05 would be considered significant.

8.2 Model Performance Evaluation

- Predicting a booking cancellation will not cancelled but in reality, the booking will cancel (FN)
- Predicting a booking cancellation will cancel but in reality, the booking will not cancel (FP)

Which case is more important?

- If we predict that a booking cancellation will not cancelled but in reality, the booking will cancel then the hotel will have to bear the cost of cancellation like revenue loss
- If we predict that a booking cancellation will cancel but in reality, the booking will not cancel then the hotel will have to arrange rooms in other hotel for rental or they should give discount
- This cost is generally less compared to the cost of cancellation.

How to reduce this loss?

- We need to reduce both False Negatives and False Positives
- f1_score should be maximized as the greater the f1_score, the higher the chances of reducing both False Negatives and False Positives and identifying both the classes correctly
    - fi_score is computed as

$$f1\_score = (2*Precision*Recall)/ (Precision+Recall)$$

First, let's create functions to calculate different metrics and confusion matrix so that we don't have to use the same code repeatedly for each model.

- The model_performance_classification_statsmodels function will be used to check the model performance of models.
- The confusion_matrix_statsmodels function will be used to plot confusion matrix.
- **Model Evaluation**: The confusion matrix helps in understanding the types of errors your model is making and can guide you in improving the model.

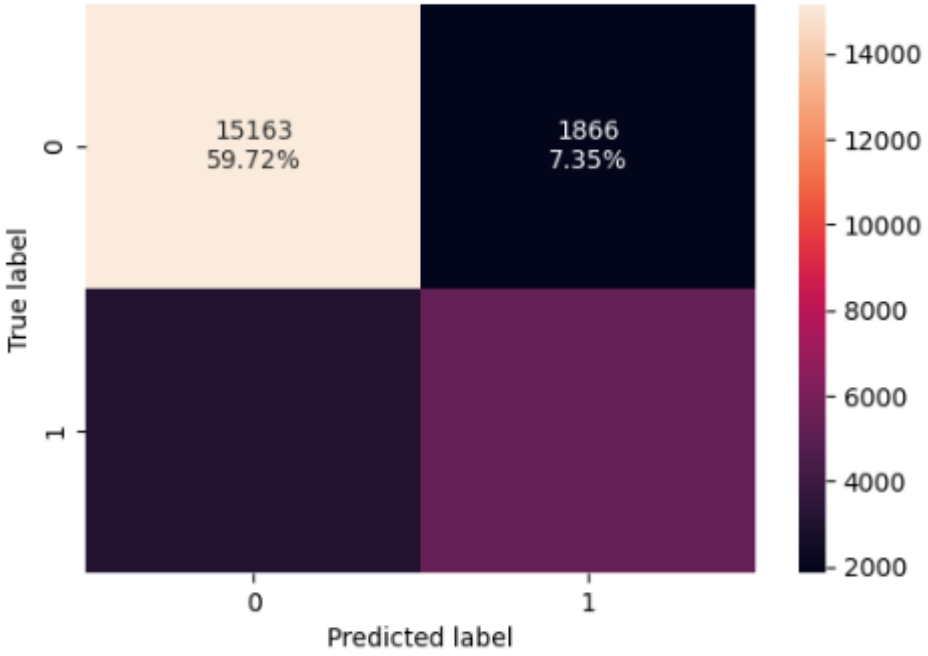


Figure 44 Confusion matrix

- **True Positives (TP)**: The top-left cell (15163, 59.72%) represents the number of correctly predicted positive cases.

- **False Positives (FP)**: The top-right cell (1866, 7.35%) represents the number of negative cases incorrectly predicted as positive.
- **True Negatives (TN)**: The bottom-right cell represents the number of correctly predicted negative cases.
- **False Negatives (FN)**: The bottom-left cell represents the number of positive cases incorrectly predicted as negative.
- **Threshold Adjustment**: By analyzing the confusion matrix, you can adjust the decision threshold to balance precision and recall according to your specific needs.

Model Training Performance check:

Training performance:

	Accuracy	Recall	Precision	F1
0	0.80604	0.63422	0.73975	0.68293

Figure 45 Training Performance

Observations

- The f1_score of the model is ~0.68 and we will try to maximize it further
- The variables used to build the model might contain multicollinearity, which will affect the p-values
- We will have to remove multicollinearity from the data to get reliable coefficients and p-values

8.3 Detecting and Dealing with Multicollinearity

There are different ways of detecting (or testing for) multicollinearity. One such way is using the Variation Inflation Factor (VIF).

- **Variance Inflation factor**: Variance inflation factors measure the inflation in the variances of the regression coefficients estimates due to collinearities that exist among the predictors. It is a measure of how much the variance of the estimated regression coefficient βkβk is "inflated" by the existence of correlation among the predictor variables in the model.
- **General Rule of thumb**:
    - If VIF is 1 then there is no correlation among the kkth predictor and the remaining predictor variables, and hence the variance of βkβk is not inflated at all
    - If VIF exceeds 5, we say there is moderate multicollinearity
    - If VIF is equal or exceeding 10, it shows signs of high multi-collinearity
- The purpose of the analysis should dictate which threshold to use

```
const                                  39468156.70600
no_of_adults                                 1.34815
no_of_children                               1.97823
no_of_weekend_nights                         1.06948
no_of_week_nights                            1.09567
required_car_parking_space                   1.03993
lead_time                                    1.39491
arrival_year                                 1.43083
arrival_month                                1.27567
arrival_date                                 1.00674
repeated_guest                               1.78352
no_of_previous_cancellations                 1.39569
no_of_previous_bookings_not_canceled         1.65199
avg_price_per_room                           2.05042
no_of_special_requests                       1.24728
type_of_meal_plan_Meal Plan 2                1.27185
type_of_meal_plan_Meal Plan 3                1.02522
type_of_meal_plan_Not Selected               1.27218
room_type_reserved_Room_Type 2               1.10144
room_type_reserved_Room_Type 3               1.00330
room_type_reserved_Room_Type 4               1.36152
room_type_reserved_Room_Type 5               1.02781
room_type_reserved_Room_Type 6               1.97307
room_type_reserved_Room_Type 7               1.11512
market_segment_type_Complementary            4.50011
market_segment_type_Corporate               16.92844
market_segment_type_Offline                 64.11392
market_segment_type_Online                  71.17643
dtype: float64
```

Figure 46 VIF Series before feature selection

Observation:

- market_segment_type_Corporate, market_segment_type_Online and market_segment_type_ Offline exhibit high multicollinearity.
- We will drop market segment type corporate and market segment type online.

Removing market_segment_type_Corporate

const	39398297.48593
no_of_adults	1.34412
no_of_children	1.97810
no_of_weekend_nights	1.06742
no_of_week_nights	1.09382
required_car_parking_space	1.03992
lead_time	1.39417
arrival_year	1.42861
arrival_month	1.27519
arrival_date	1.00672
repeated_guest	1.77838
no_of_previous_cancellations	1.39514
no_of_previous_bookings_not_canceled	1.64962
avg_price_per_room	2.04989
no_of_special_requests	1.24702
type_of_meal_plan_Meal Plan 2	1.27165
type_of_meal_plan_Meal Plan 3	1.02522
type_of_meal_plan_Not Selected	1.27173
room_type_reserved_Room_Type 2	1.10139
room_type_reserved_Room_Type 3	1.00330
room_type_reserved_Room_Type 4	1.35423
room_type_reserved_Room_Type 5	1.02717
room_type_reserved_Room_Type 6	1.97298
room_type_reserved_Room_Type 7	1.11512
market_segment_type_Complementary	1.37679
market_segment_type_Offline	5.78845
market_segment_type_Online	6.42358

dtype: float64

Figure 47 VIF Series 2 before removing multicollinearity

- market_segment_type_Online and market_segment_type_Offline exhibit high multicollinearity.

Removing market_segment_type_Online

```
const                                 39391371.31459
no_of_adults                                1.33178
no_of_children                              1.97735
no_of_weekend_nights                        1.06904
no_of_week_nights                           1.09512
required_car_parking_space                  1.03979
lead_time                                   1.39064
arrival_year                                1.42838
arrival_month                               1.27463
arrival_date                                1.00672
repeated_guest                              1.78019
no_of_previous_cancellations                1.39545
no_of_previous_bookings_not_canceled        1.65175
avg_price_per_room                          2.04959
no_of_special_requests                      1.24242
type_of_meal_plan_Meal Plan 2               1.27150
type_of_meal_plan_Meal Plan 3               1.02522
type_of_meal_plan_Not Selected              1.27039
room_type_reserved_Room_Type 2              1.10127
room_type_reserved_Room_Type 3              1.00330
room_type_reserved_Room_Type 4              1.35600
room_type_reserved_Room_Type 5              1.02781
room_type_reserved_Room_Type 6              1.97273
room_type_reserved_Room_Type 7              1.11500
market_segment_type_Complementary           1.33825
market_segment_type_Corporate               1.52777
market_segment_type_Offline                 1.59742
dtype: float64
```

Figure 48 VIF Series 2

- Dropping market_segment_type_Online fixes the multicollinearity in market_segment.

Model Performance Check:

	Accuracy	Recall	Precision	F1
0	0.80577	0.63374	0.73929	0.68246

Figure 49 training performance

Observations:

1.market_segment_type_Corporate, market_segment_type_Online and market_segment_type_Offline doesn't have a significant impact on the model performance.

2. We can choose any model to proceed to the next steps.

Logit Regression Results

Dep. Variable:	booking_status	No. Observations:	25392
Model:	Logit	Df Residuals:	25365
Method:	MLE	Df Model:	26
Date:	Sat, 07 Sep 2024	Pseudo R-squ.:	0.3292
Time:	20:43:52	Log-Likelihood:	-10794.
converged:	False	LL-Null:	-16091.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-933.3324	120.655	-7.736	0.000	-1169.813	-696.852
no_of_adults	0.1060	0.037	2.841	0.004	0.033	0.179
no_of_children	0.1542	0.057	2.694	0.007	0.042	0.266
no_of_weekend_nights	0.1075	0.020	5.439	0.000	0.069	0.146
no_of_week_nights	0.0405	0.012	3.295	0.001	0.016	0.065
required_car_parking_space	-1.5907	0.138	-11.538	0.000	-1.861	-1.320
lead_time	0.0157	0.000	58.933	0.000	0.015	0.016
arrival_year	0.4611	0.060	7.711	0.000	0.344	0.578
arrival_month	-0.0411	0.006	-6.358	0.000	-0.054	-0.028
arrival_date	0.0005	0.002	0.257	0.797	-0.003	0.004
repeated_guest	-2.3140	0.618	-3.743	0.000	-3.526	-1.102
no_of_previous_cancellations	0.2633	0.086	3.074	0.002	0.095	0.431
no_of_previous_bookings_not_canceled	-0.1728	0.152	-1.136	0.256	-0.471	0.125
avg_price_per_room	0.0187	0.001	25.374	0.000	0.017	0.020
no_of_special_requests	-1.4709	0.030	-48.891	0.000	-1.530	-1.412
type_of_meal_plan_Meal Plan 2	0.1794	0.067	2.694	0.007	0.049	0.310
type_of_meal_plan_Meal Plan 3	19.8256	1.36e+04	0.001	0.999	-2.67e+04	2.67e+04
type_of_meal_plan_Not Selected	0.2745	0.053	5.181	0.000	0.171	0.378
room_type_reserved_Room_Type 2	-0.3640	0.131	-2.784	0.005	-0.620	-0.108
room_type_reserved_Room_Type 3	-0.0018	1.310	-0.001	0.999	-2.569	2.566
room_type_reserved_Room_Type 4	-0.2763	0.053	-5.207	0.000	-0.380	-0.172
room_type_reserved_Room_Type 5	-0.7182	0.209	-3.436	0.001	-1.128	-0.308
room_type_reserved_Room_Type 6	-0.9408	0.147	-6.402	0.000	-1.229	-0.653
room_type_reserved_Room_Type 7	-1.3891	0.293	-4.743	0.000	-1.963	-0.815
market_segment_type_Complementary	-47.7454	7.09e+06	-6.74e-06	1.000	-1.39e+07	1.39e+07
market_segment_type_Corporate	-0.8033	0.103	-7.807	0.000	-1.005	-0.602
market_segment_type_Offline	-1.7995	0.052	-34.577	0.000	-1.902	-1.698

Figure 50 Logistic Regression Result after removing multicollinearity

Now that we do not have multicollinearity in our data, the p-values of the coefficients have become reliable and we can remove the non-significant predictor variables all together having p value > 0.05

Removing high p-value variables

- For other attributes present in the data, the p-values are high only for few dummy variables and since only one (or some) of the categorical levels have a high p-value we will drop them iteratively as sometimes p-values change after dropping a variable. So, we'll not drop all variables at once.
- Instead, we will do the following repeatedly using a loop:
    - Build a model, check the p-values of the variables, and drop the column with the highest p-value.
    - Create a new model without the dropped feature, check the p-values of the variables, and drop the column with the highest p-value.
    - Repeat the above two steps till there are no columns with p-value > 0.05.

Note: The above process can also be done manually by picking one variable at a time that has a high p-value, dropping it, and building a model again. But that might be a little tedious and using a loop will be more efficient.

Logit Regression Results

Dep. Variable:	booking_status	No. Observations:	25392
Model:	Logit	Df Residuals:	25370
Method:	MLE	Df Model:	21
Date:	Sat, 07 Sep 2024	Pseudo R-squ.:	0.3283
Time:	20:43:53	Log-Likelihood:	-10809.
converged:	True	LL-Null:	-16091.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-917.2860	120.456	-7.615	0.000	-1153.376	-681.196
no_of_adults	0.1086	0.037	2.914	0.004	0.036	0.182
no_of_children	0.1522	0.057	2.660	0.008	0.040	0.264
no_of_weekend_nights	0.1086	0.020	5.501	0.000	0.070	0.147
no_of_week_nights	0.0418	0.012	3.403	0.001	0.018	0.066
required_car_parking_space	-1.5943	0.138	-11.561	0.000	-1.865	-1.324
lead_time	0.0157	0.000	59.218	0.000	0.015	0.016
arrival_year	0.4531	0.060	7.591	0.000	0.336	0.570
arrival_month	-0.0424	0.006	-6.568	0.000	-0.055	-0.030
repeated_guest	-2.7365	0.557	-4.915	0.000	-3.828	-1.645
no_of_previous_cancellations	0.2289	0.077	2.983	0.003	0.078	0.379
avg_price_per_room	0.0192	0.001	26.343	0.000	0.018	0.021
no_of_special_requests	-1.4699	0.030	-48.892	0.000	-1.529	-1.411
type_of_meal_plan_Meal Plan 2	0.1654	0.067	2.487	0.013	0.035	0.296
type_of_meal_plan_Not Selected	0.2858	0.053	5.405	0.000	0.182	0.389
room_type_reserved_Room_Type 2	-0.3560	0.131	-2.725	0.006	-0.612	-0.100
room_type_reserved_Room_Type 4	-0.2826	0.053	-5.330	0.000	-0.387	-0.179
room_type_reserved_Room_Type 5	-0.7352	0.208	-3.529	0.000	-1.143	-0.327
room_type_reserved_Room_Type 6	-0.9650	0.147	-6.572	0.000	-1.253	-0.677
room_type_reserved_Room_Type 7	-1.4312	0.293	-4.892	0.000	-2.005	-0.858
market_segment_type_Corporate	-0.7928	0.103	-7.711	0.000	-0.994	-0.591
market_segment_type_Offline	-1.7867	0.052	-34.391	0.000	-1.889	-1.685

Figure 51 Logistic Regression Result after removing high p- value

Now no categorical feature has p-value greater than 0.05, so we'll consider the features in *X_train2* as the final ones and *lg3* as final model.

Coefficient Interpretations

- Coefficient of no_of_adults, no_of_children, no_of_weekend_nights, lead_time ,arrival_year, no_of_previous_cancellations, type_of_meal_plan_Meal Plan 2, room_type_reserved_Room_Type 4, and market_segment_type_Online are positive an increase in these will lead to increase in chances of cancelling on booking.
- Coefficient of required_car_parking_space, repeated_guest, no_of_special_requests, room_type_reserved_Room_Type 2, and market_segment_type_Offline are negative increase in these will lead to decrease in chances of cancelling on booking.

Converting coefficients to odds

- The coefficients (βs) of the logistic regression model are in terms of log(odds) and to find the odds, we have to take the exponential of the coefficients
- Therefore, **odds=exp(β)**

- The percentage change in odds is given as (exp(b)-1)*100

	const	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	required_car_parking_space	lead_time	arrival_year	arrival_month
Odds	0.00000	1.11475	1.16436	1.11475	1.04264	0.20305	1.01584	1.57324	0.95853
Change_odd%	-100.00000	11.47536	16.43601	11.47526	4.26363	-79.69523	1.58352	57.32351	-4.14725

Figure 52 Converting coefficients to odds

Coefficient interpretations

- Repeated Guest: Being a repeated guest reduces the odds of cancellation by 95.21%, making it the strongest factor against cancellation.
- Required Car Parking Space: Guests who request parking are 74% less likely to cancel their bookings.
- Market Segment (Online): Online bookings are 176.75% more likely to cancel compared to other segments.
- Number of Special Requests: Each additional special request decreases the odds of cancellation by 76.23%, indicating that guests with more requests are less likely to cancel.
- Number of Children: Each additional child increases the odds of cancellation by 47.58%, suggesting family bookings are more likely to cancel.

Checking performance of the new model:

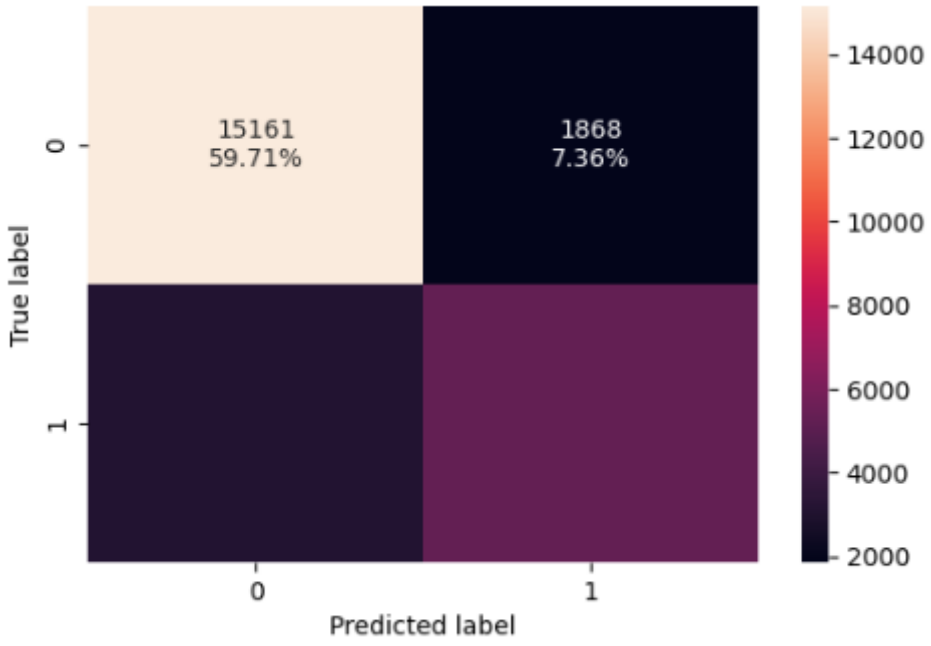


Figure 53 Confusion Matrix

- The model has a high number of true positives, indicating it is good at correctly identifying positive cases.

- The false negatives, while not extremely high, indicate there are some positive cases the model is missing. Improving recall could be beneficial.
- **Areas for Improvement**: Reducing false positives and false negatives can help improve the model's precision and recall, respectively.

Training set performance:

	Accuracy	Recall	Precision	F1
0	0.80541	0.63255	0.73903	0.68166

Figure 54 Training Set Performance Of The New Model

- The model has a high recall, meaning it is effective at identifying most of the true positive cases.
- The precision is lower, indicating that there are a significant number of false positives. This could be problematic in scenarios where false alarms are costly.
- **Areas for Improvement**: Enhancing precision without significantly compromising recall would improve the model's overall performance. Techniques such as adjusting the decision threshold, using more balanced training data, or employing more sophisticated algorithms might help.

Test set performance

- *We have to first drop the columns from the test set that were dropped from the training set.*

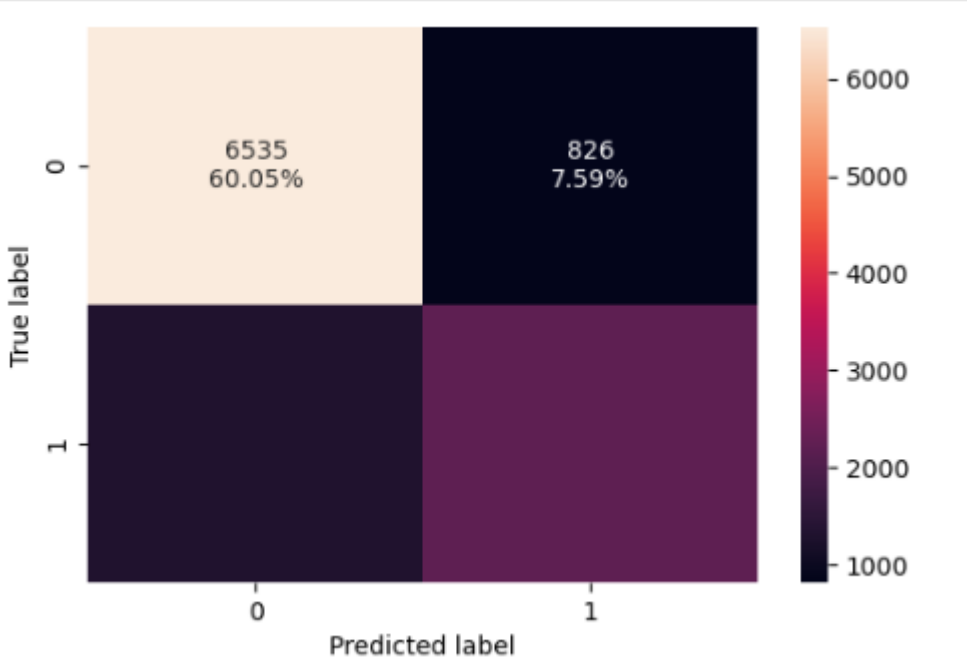


Figure 55 Confusion Matrix

- The model has a high number of true negatives, indicating it is good at correctly identifying negative cases.

- The false positives, while not extremely high, indicate there are some negative cases the model is incorrectly predicting as positive. Improving precision could be beneficial.
- **Areas for Improvement**: Reducing false positives and false negatives can help improve the model's precision and recall, respectively. Techniques such as adjusting the decision threshold, using more balanced training data, or employing more sophisticated algorithms might help.

	Accuracy	Recall	Precision	F1
0	0.80465	0.63089	0.72900	0.67641

Figure 56 Test Set Performance Of The New Model

The model is giving a good f1_score of ~0.681 and ~0.676 on the train and test sets respectively

- As the train and test performances are comparable, the model is not overfitting
- Moving forward we will try to improve the performance of the model

Model Performance Improvement

- Let's see if the f1_score can be improved further by changing the model threshold
- First, we will check the ROC curve, compute the area under the ROC curve (ROC-AUC), and then use it to find the optimal threshold
- Next, we will check the Precision-Recall curve to find the right balance between precision and recall as our metric of choice is f1_score

ROC Curve and ROC-AUC:

(ROC) curve, which is used to evaluate the performance of a binary classification model.

- ROC-AUC on training set:

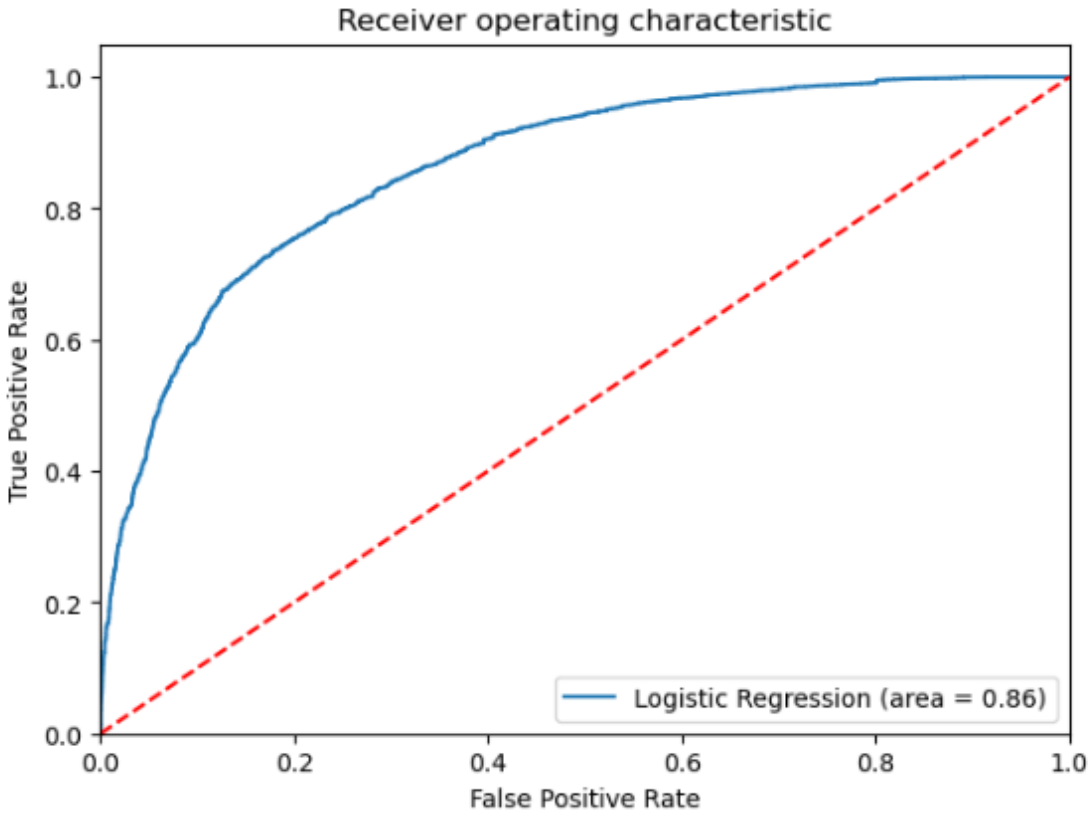


Figure 57 ROC -AUC On Training Set

- **Logistic Regression Model**: The solid blue line represents the performance of a Logistic Regression model. The Area Under the Curve (AUC) for this model is 0.86, indicating good predictive ability. An AUC of 1.0 represents a perfect model, while an AUC of 0.5 represents a model that performs no better than random chance.
- **Reference Line**: The dashed red diagonal line represents random guessing, with an AUC of 0.5. This serves as a baseline to compare the model's performance.
- **True Positive Rate (TPR)**: Also known as sensitivity or recall, this is plotted on the y-axis. It measures the proportion of actual positives correctly identified by the model.
- **False Positive Rate (FPR)**: This is plotted on the x-axis and measures the proportion of actual negatives incorrectly identified as positives by the model.
- The ROC curve helps in understanding the trade-off between the TPR and FPR for different threshold settings of the model. The closer the ROC curve is to the top-left corner, the better the model's performance.

- Logistic Regression model is giving a good performance on training set.

Optimal threshold using AUC-ROC curve:

The optimal cut off would be where tpr is high and fpr is low.

- Optimal threshold using AUC-ROC curve = 0.371046662348869

Checking model performance on training set

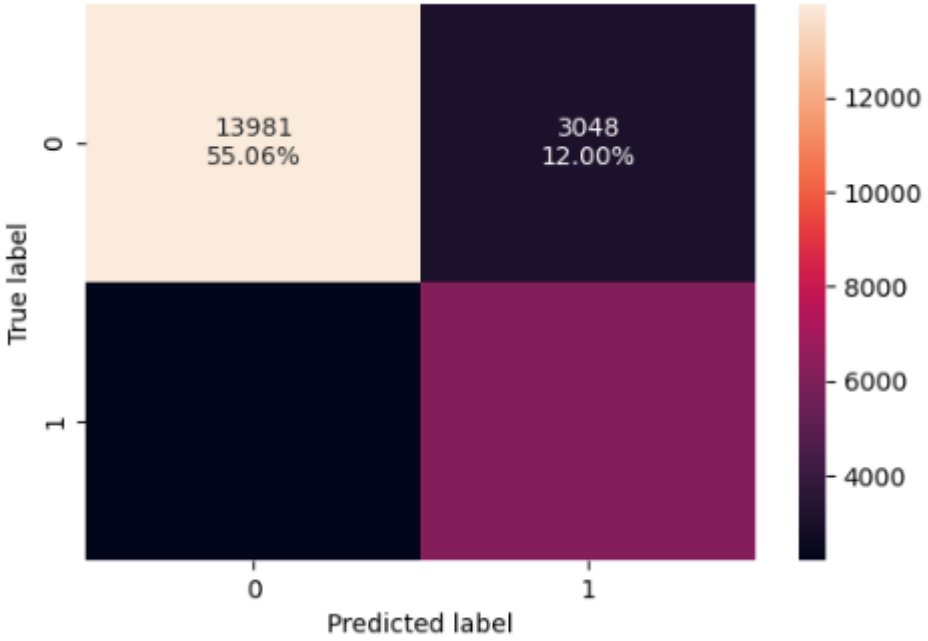


Figure 58 Confusion Matrix

Checking model performance for Training set:

	Accuracy	Recall	Precision	F1
0	0.79289	0.73562	0.66870	0.70056

Figure 59 Model Performance For Training Set

- Precision of model has increased but the other metrics have reduced.
- The model is still giving a good performance.

Checking model performance on test set:

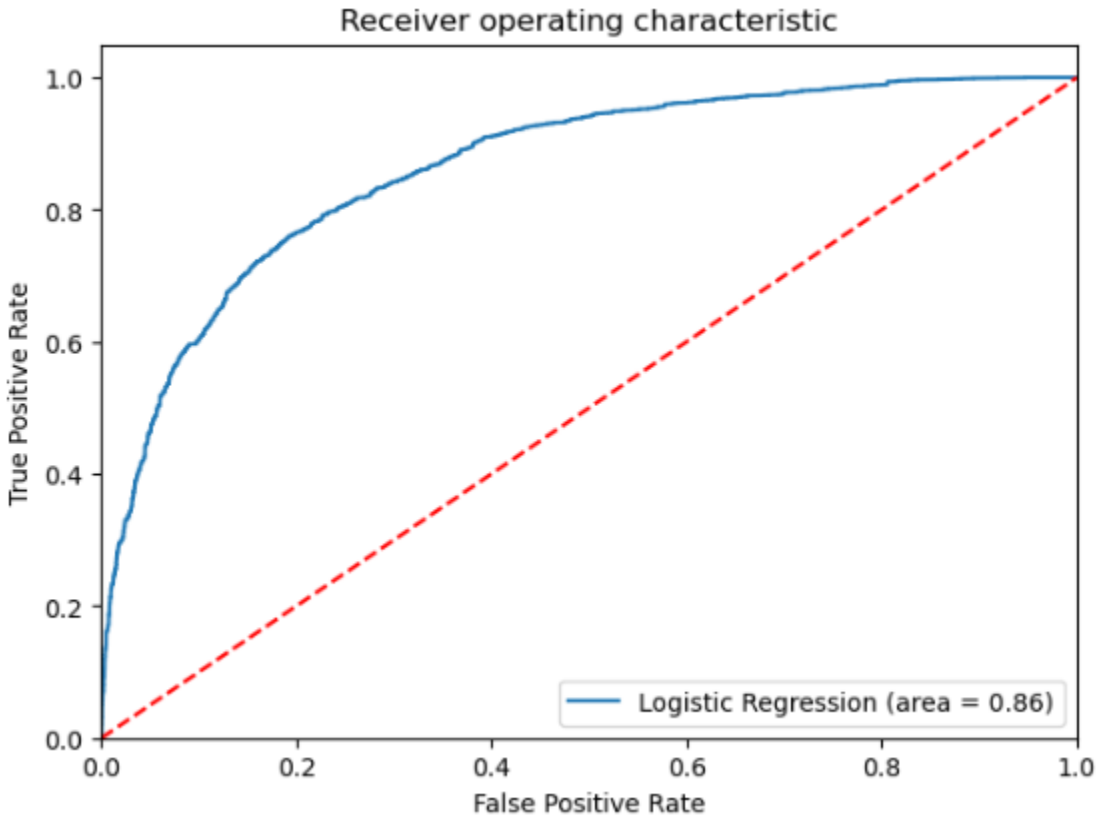


Figure 60 ROC-AUC On Test Set

- The closer the ROC curve is to the top-left corner, the better the model's performance. It indicates a higher true positive rate and a lower false positive rate.
- An AUC of 0.86 suggests that the Logistic Regression model has a good level of separability. It means that there is an 86% chance that the model will correctly distinguish between a randomly chosen positive instance and a randomly chosen negative instance.

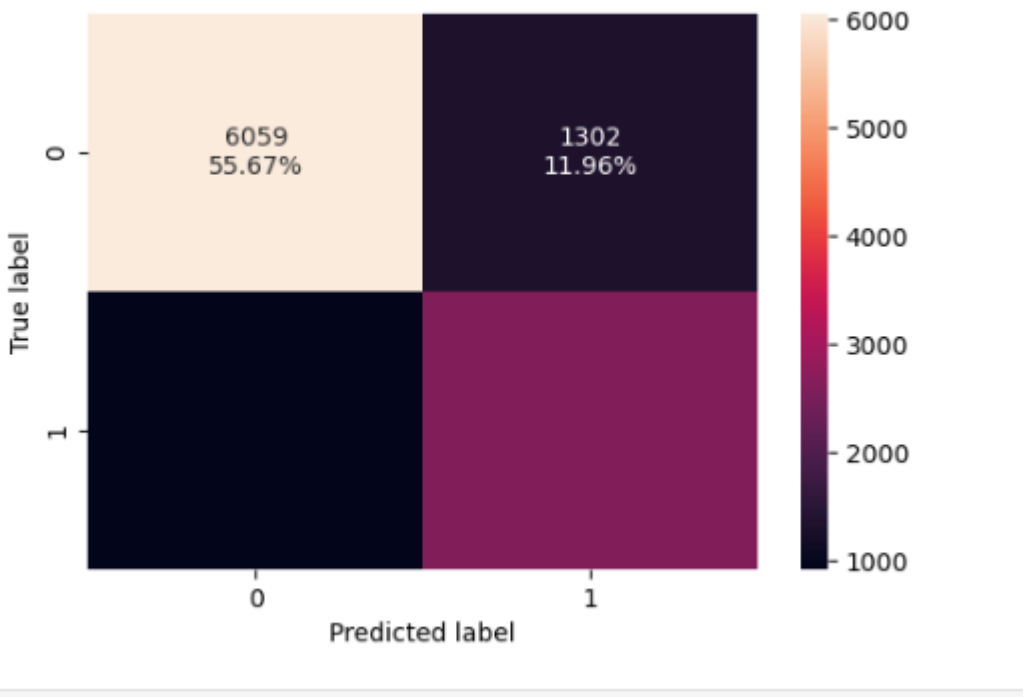


Figure 61 Confusion Matrix

Checking model performance on Test set:

	Accuracy	Recall	Precision	F1
0	0.79601	0.73935	0.66667	0.70113

Figure 62 Figure 63 Model Performance on Test Set

Precision-Recall Curve:

We use precision-recall curves to evaluate the performance of classification models, especially when dealing with imbalanced datasets**.**

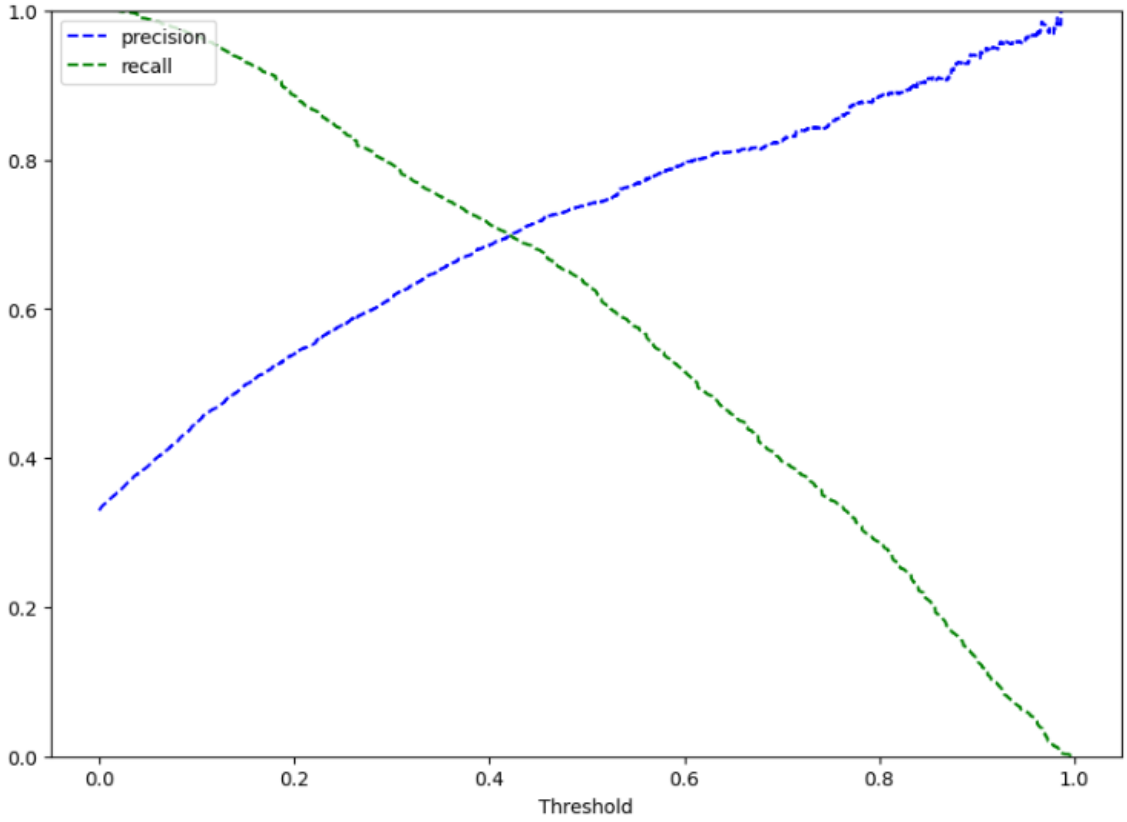


Figure 64 Precision - Recall curve

- **Precision** (dashed blue line): As the threshold increases, precision also increases. This means that as the model becomes more confident in its positive predictions, the proportion of true positives among all positive predictions improves.
- **Recall** (dotted green line): Conversely, as the threshold increases, recall decreases. This indicates that the model is identifying fewer true positives out of all actual positives as it becomes more conservative in its predictions.
- At the threshold of 0.42, we get balanced recall and precision.

Let's set the optimal_threshold_curve = 0.42.

Checking model performance on training set:

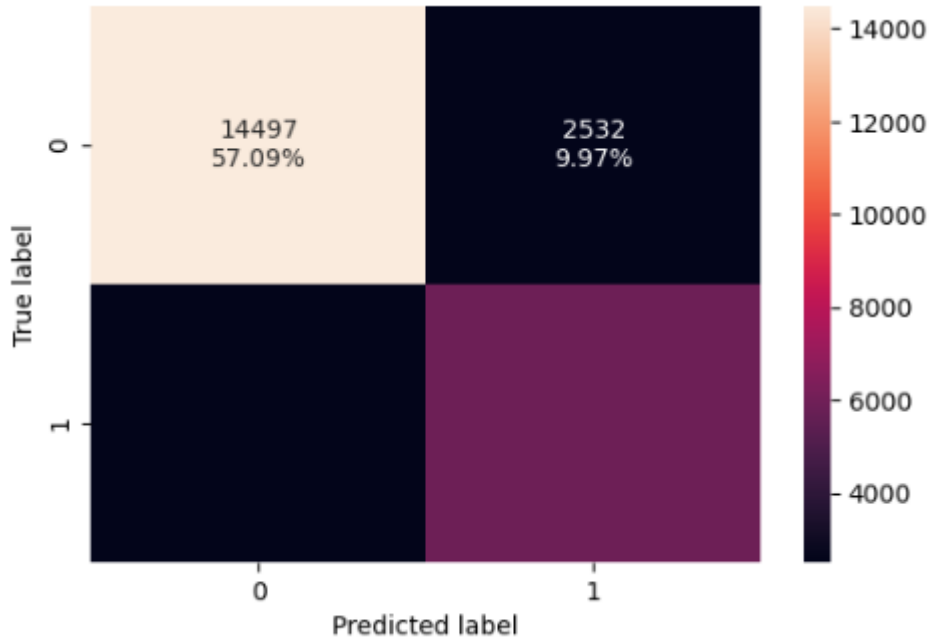


Figure 65 Confusion Matrix

	Accuracy	Recall	Precision	F1
0	0.80128	0.69939	0.69789	0.69864

Figure 66 Model Performance on Training set

- Model is performing well on training set.
- There's not much improvement in the model performance as the default threshold is 0.50 and here we get 0.42 as the optimal threshold.

Checking model performance on test set:

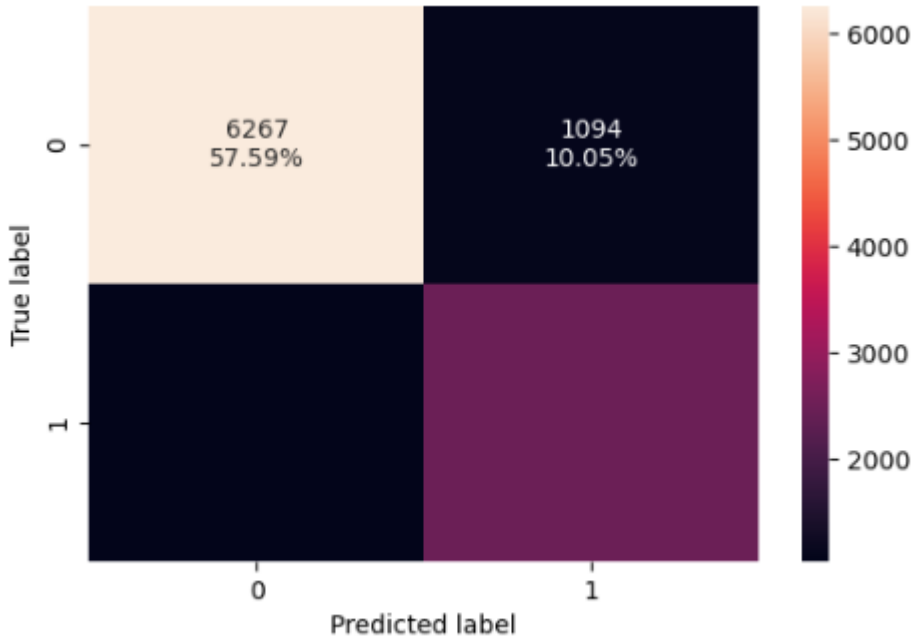


Figure 67 Confusion Matrix

	Accuracy	Recall	Precision	F1
0	0.80364	0.70386	0.69381	0.69880

Figure 68 Test performance

Model Performance Comparison and Final Model Selection

Training performance comparison:

	Logistic Regression-default Threshold (0.5)	Logistic Regression- 0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80541	0.79289	0.80128
Recall	0.63255	0.73562	0.69939
Precision	0.73903	0.66870	0.69789
F1	0.68166	0.70056	0.69864

Test set performance comparison:

	Logistic Regression-default Threshold (0.5)	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80465	0.79601	0.80364
Recall	0.63089	0.73935	0.70386
Precision	0.72900	0.66667	0.69381
F1	0.67641	0.70113	0.69880

- Almost all the three models are performing well on both training and test data without the problem of overfitting
- The model with an Optimal threshold using AUC-ROC curve(0.37) is giving the best F1 score. Therefore, it can be selected as the final model

9. KNN CLASSIFIER (SKLEARN)

The **k-nearest neighbors (KNN) classifier** is a simple, yet powerful, supervised machine learning algorithm used for both classification and regression tasks.

- **Non-Parametric**: KNN is non-parametric, meaning it doesn't make any assumptions about the underlying data distribution.
- **Instance-Based**: It is an instance-based learning algorithm, also known as a lazy learner, because it doesn't explicitly learn a model. Instead, it memorizes the training dataset and performs classification or regression at the time of prediction.
- **Distance-Based**: KNN classifies a data point based on how its neighbors are classified. The "k" in KNN refers to the number of nearest neighbors considered when making the classification.

How Does KNN Work?

1. **Store Training Data**: The algorithm stores all the training data.
2. **Calculate Distance**: For a new data point, it calculates the distance (commonly Euclidean) to all other points in the training data.
3. **Find Nearest Neighbors**: It identifies the k-nearest neighbors to the new data point.
4. **Majority Vote**: For classification, it assigns the class that is most common among the k-nearest neighbors. For regression, it averages the values of the k-nearest neighbors.

9.1 Model Evaluation

Model evaluation criterion:

Model can make wrong predictions as:

- Predicting a booking cancellation will not cancelled but in reality, the booking will cancel (FN)
- Predicting a booking cancellation will cancel but in reality, the booking will not cancel (FP)

Which case is more important?

- If we predict that a booking cancellation will not cancelled but in reality, the booking will cancel then the hotel will have to bear the cost of cancellation like revenue loss
- If we predict that a booking cancellation will cancel but in reality, the booking will not cancel then the hotel will have to arrange rooms in other hotel for rental or they should give discount
- This cost is generally less compared to the cost of cancellation.

How to reduce the losses?

The Hotel would want the recall to be maximized, greater the recall score higher are the chances of minimizing the False Negatives.

9.2 K- Nearest Neighbor

In order to optimize our model, it's essential to experiment with different values of k to find the most suitable fit for our data. We can commence this process by setting k equal to 3 and gradually exploring other values to assess their impact on the model's performance.

- We'll only consider odd values of K as the classification will be done based on majority voting.

K=3

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

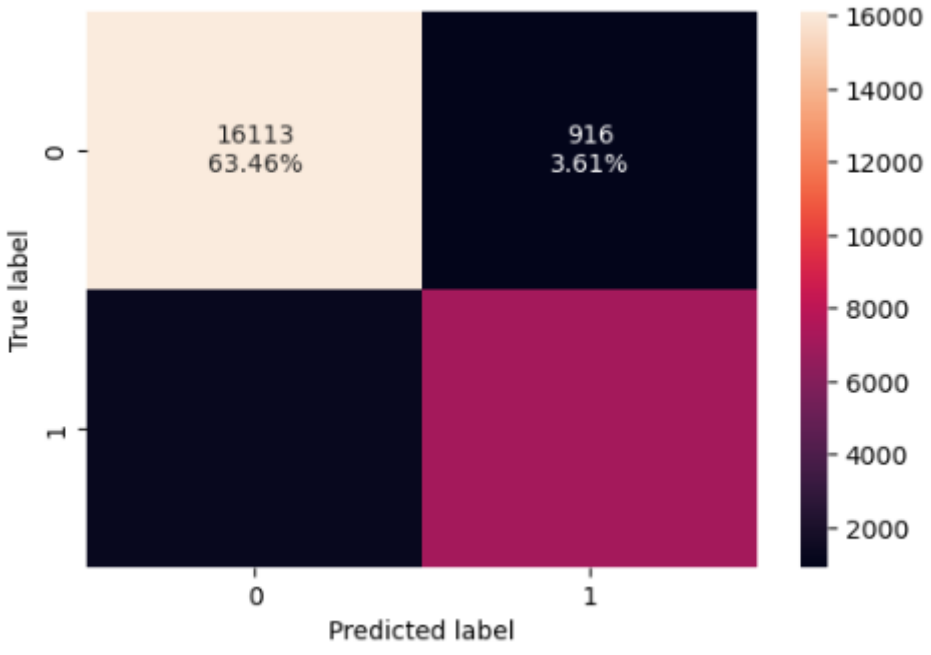


Figure 69 Confusion Matrix for KNN =3

Model Performance on Training set:

	Accuracy	Recall	Precision	F1
0	0.91588	0.85412	0.88634	0.86993

Figure 70 Model Performance on Training set

These metrics suggest that the model is performing well, with a good balance between precision and recall, making it reliable for classification tasks.

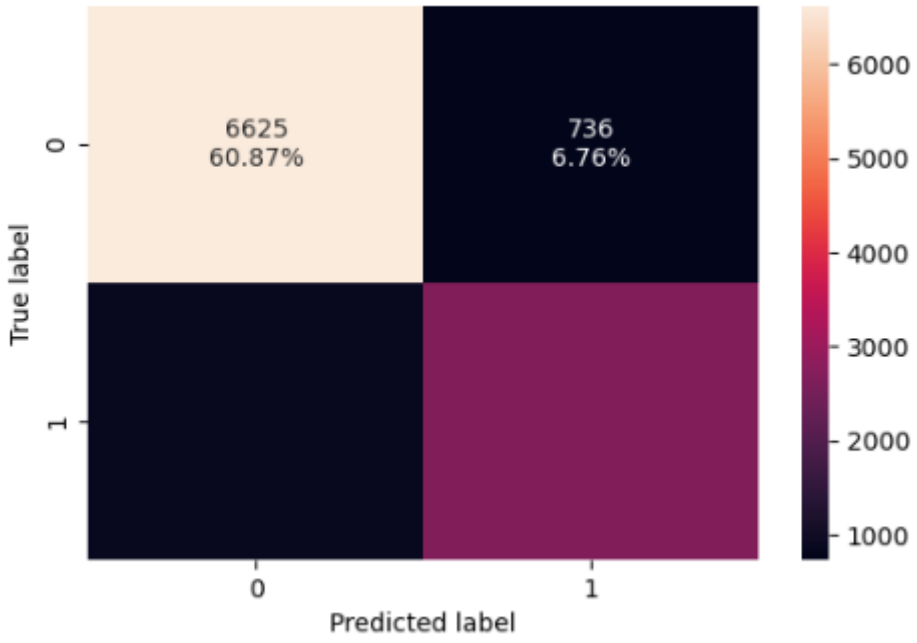


Figure 71 Confusion Matrix (KNN =3) for Test set

	Accuracy	Recall	Precision	F1
0	0.85271	0.75383	0.78295	0.76812

Figure 72 Model Performance on Test set

Let's run the KNN with no of neighbours to be 1,3,5..19 and find the optimal number of neighbours from the above list using the recall score

9.3 K with different values

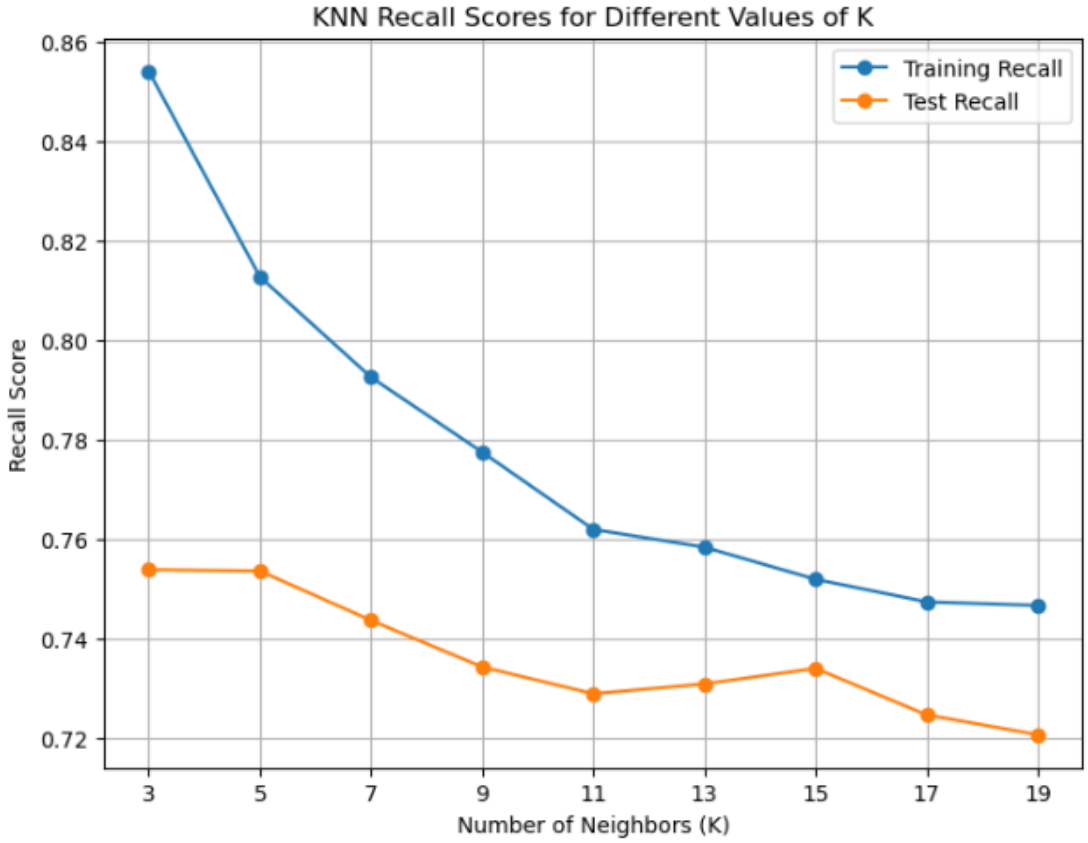


Figure 73 KNN Recall Score for Different Values of K

- The recall scores for both training and test sets are highest when k=3. This suggests that with k=3, the model is better at identifying positive instances in both the training and test data compared to other values of k.
- As the value of k increases beyond 3, the recall scores tend to decrease for both training and test sets. This indicates a potential risk of the model not being able to identify the underlying patterns in the data.
- Therefore, based on the provided recall scores, k=3 appears to be the most suitable choice for balancing model performance between capturing positive instances effectively and generalizing well to new data.

10. NAIVE BAYES

Naive Bayes classifiers are probabilistic classifiers that apply Bayes' Theorem with the "naive" assumption that features are conditionally independent given the class label. This means that the presence of a particular feature in a class is unrelated to the presence of any other feature.

10.1 Check Model Performance

```
GaussianNB
GaussianNB()
```

10.1.1 Model Performance on Training set

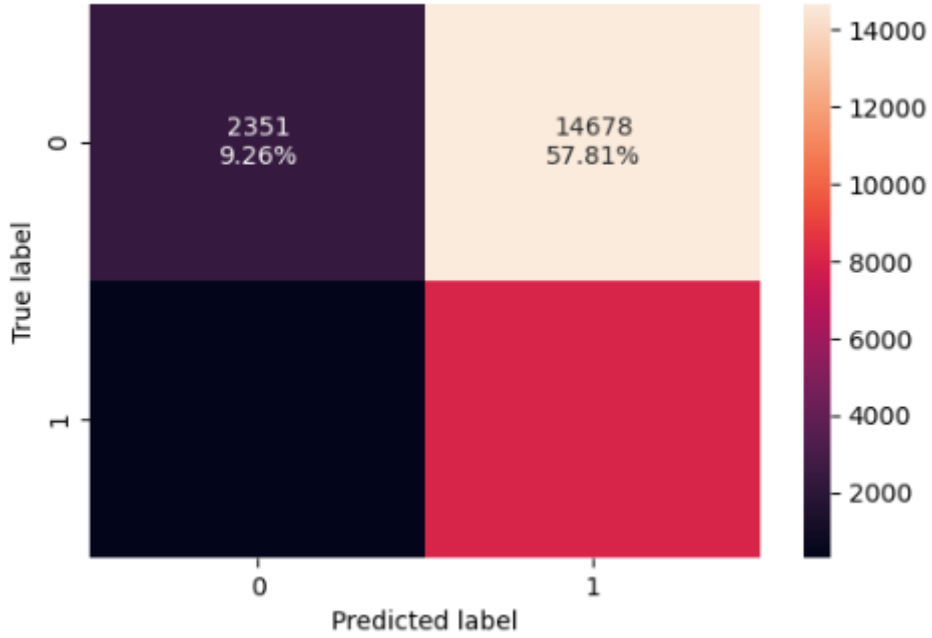


Figure 74 Confusion Matrix

	Accuracy	Recall	Precision	F1
0	0.40970	0.96281	0.35425	0.51793

Figure 75 Model Performance on Training set

10.1.2 Model Performance on Test set

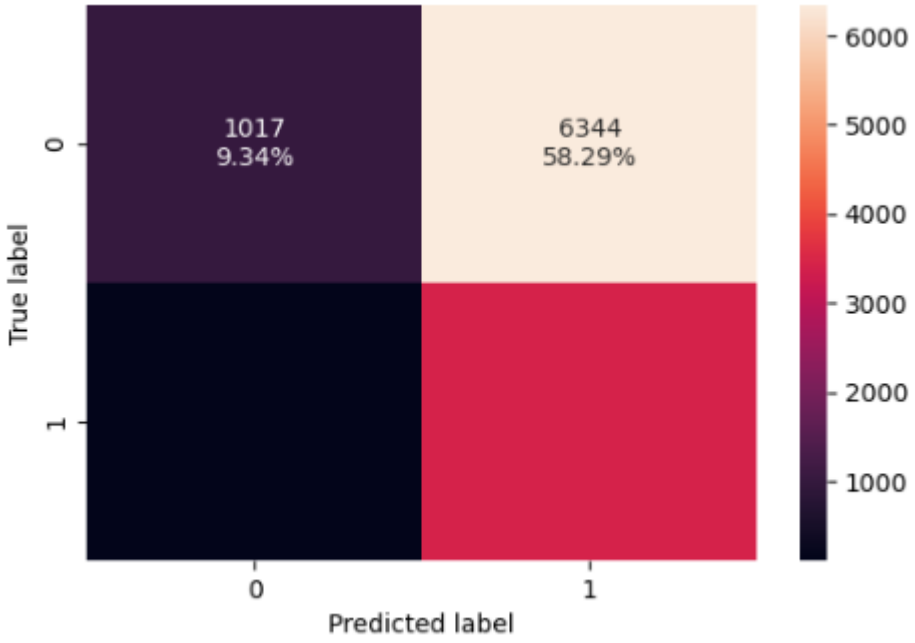


Figure 76 Confusion Matric

	Accuracy	Recall	Precision	F1
0	0.40614	0.96621	0.34913	0.51292

Figure 77 Model Performance on Test set

10.2 Comparison of Models and Final Model Selection

10.2.1 Training performance comparison

	K Nearest Neighbor k=3	Naive Bayes
Accuracy	0.91588	0.40970
Recall	0.85412	0.96281
Precision	0.88634	0.35425
F1	0.86993	0.51793

Figure 78 Comparison On Training Performance Model

10.2.2 Test set performance comparison

	K Nearest Neighbor k=3	Naive Bayes
Accuracy	0.85271	0.40614
Recall	0.75383	0.96621
Precision	0.78295	0.34913
F1	0.76812	0.51292

Figure 79 Comparison on Test Performance Model

- In both the training and test sets, the K Nearest Neighbor model with k=3 demonstrates the highest recall among all compared models.
- This indicates that the model with k=3 is better at correctly identifying positive instances compared to the models with different k values and Naive Bayes.
- Naive Bayes consistently shows lower recall values compared to K Nearest Neighbor models with different k values.
- This suggests that Naive Bayes may struggle to capture positive instances as effectively as K Nearest Neighbor models in both training and test datasets, highlighting potential limitations in its performance for this specific task.

11. DECISION TREE

A **Decision Tree** is a type of supervised machine learning algorithm that is used for both classification and regression tasks. It works by recursively splitting the dataset into smaller subsets based on certain criteria, creating a tree-like structure of decisions. The goal is to learn decision rules from the data that can be used to predict the target variable.

11.1 Decision Tree (default)

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

Model can make wrong predictions as:

- Predicting a booking cancellation will not cancelled but in reality, the booking will cancel (FN)
- Predicting a booking cancellation will cancel but in reality, the booking will not cancel (FP)

Which case is more important?

- If we predict that a booking cancellation will not cancelled but in reality, the booking will cancel then the hotel will have to bear the cost of cancellation like revenue loss
- If we predict that a booking cancellation will cancel but in reality, the booking will not cancel then the hotel will have to arrange rooms in other hotel for rental or they should give discount
- This cost is generally less compared to the cost of cancellation.

How to reduce the losses?

- The Hotel would want the recall to be maximized, greater the recall score higher are the chances of minimizing the False Negatives.

11.1.1 Model Performance on Training set

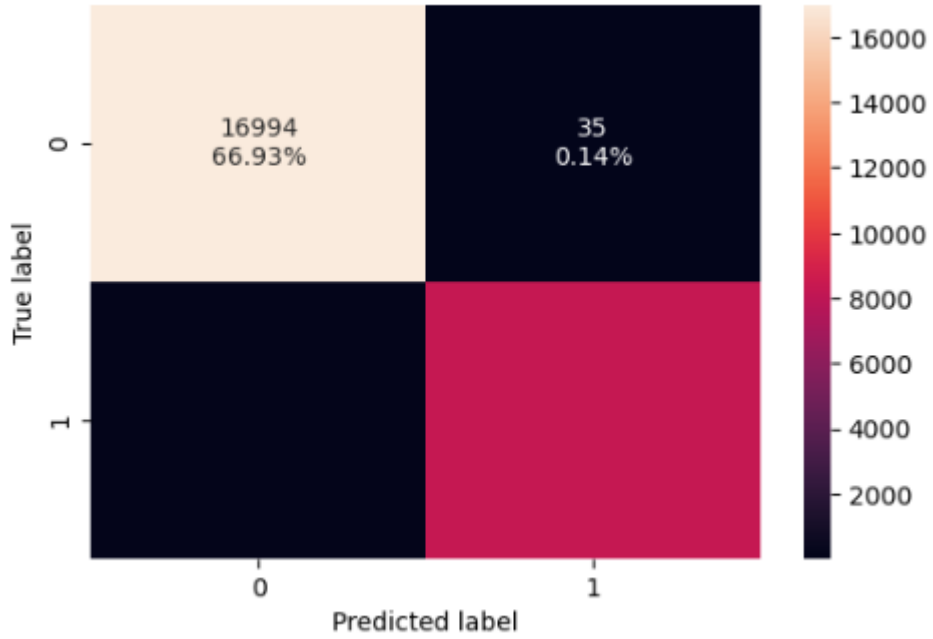


Figure 80 Confusion Matrix

	Accuracy	Recall	Precision	F1
0	0.99421	0.98661	0.99578	0.99117

Figure 81 Model Performance on Training set

- This model performs exceptionally well across all key metrics. The **high accuracy** and **precision** show that it makes very few mistakes in both positive and negative predictions.
- The **slightly lower recall** compared to precision suggests that the model might miss a small number of actual cancellations but doesn't often wrongly predict cancellations.
- Given the **high F1-score**, this model is well-suited for predicting hotel booking cancellations with minimal trade-offs between false positives and false negatives.

11.1.2 Model Performance on Test set

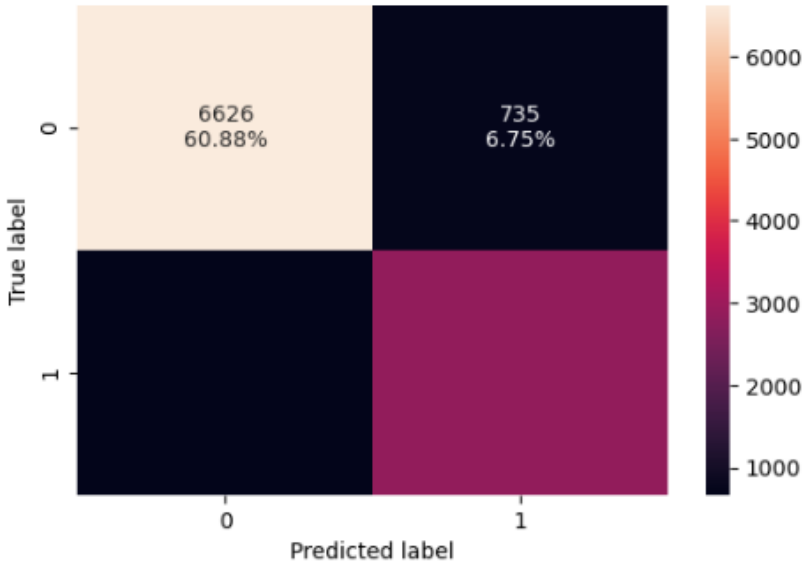


Figure 82 confusion matrix

	Accuracy	Recall	Precision	F1
0	0.87108	0.81034	0.79521	0.80270

Figure 83 Model Performance on Test set

- The **accuracy** is lower than the training model, suggesting that this model struggles more in predicting cancellations correctly overall.
- **Recall** is slightly better than precision, meaning the model is better at identifying true positives (actual cancellations) than avoiding false positives.
- **Precision** below 80% indicates that the model may have a tendency to incorrectly predict some bookings as cancellations when they are not.
- The **F1-score** indicates that there is a trade-off between precision and recall, but neither of them is particularly high.
- Both recall and precision can be improved to boost overall model performance. Depending on the business need (e.g., minimizing false positives or maximizing true positives), you may want to tune the model further by adjusting hyperparameters, trying other algorithms, or feature engineering.

11.2 Decision Tree (with class_weights)

- If the frequency of class A is 10% and the frequency of class B is 90%, then class B will become the dominant class and the decision tree will become biased toward the dominant classes
- In this case, we will set class_weight = "balanced", which will automatically adjust the weights to be inversely proportional to the class frequencies in the input data

- class_weight is a hyperparameter for the decision tree classifier

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

11.2.1 Model Performance on Training set

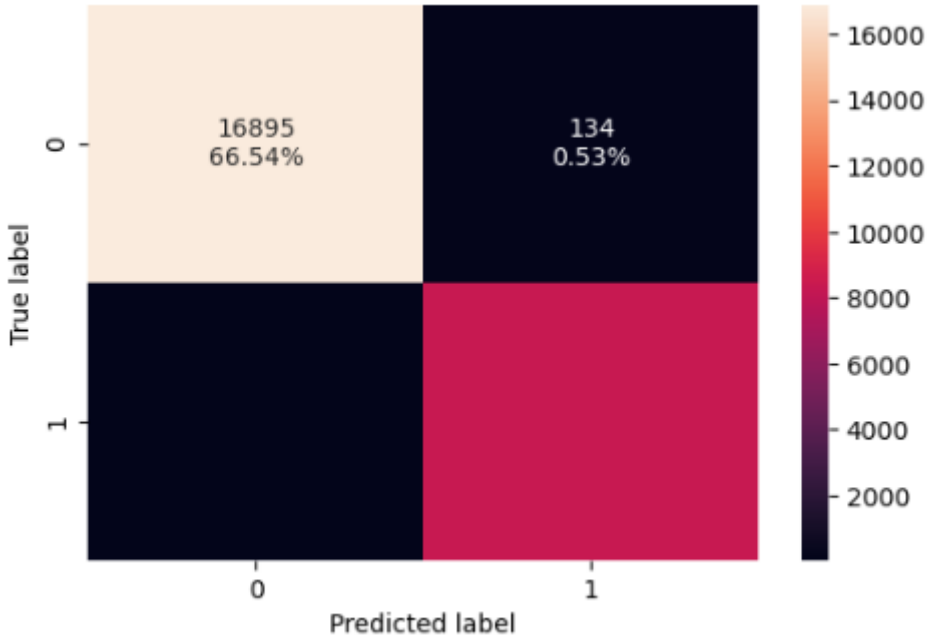


Figure 84 Confusion Matrix

	Accuracy	Recall	Precision	F1
0	0.99311	0.99510	0.98415	0.98960

Figure 85 Model Performance on Training set

- Model is able to perfectly classify all the data points on the training set.
- As we know a decision tree will continue to grow and classify each data point correctly if no restrictions are applied as the trees will learn all the patterns in the training set.
- This generally leads to overfitting of the model as Decision Tree will perform well on the training set but will fail to replicate the performance on the test set.

11.2.2 Model Performance on Test set

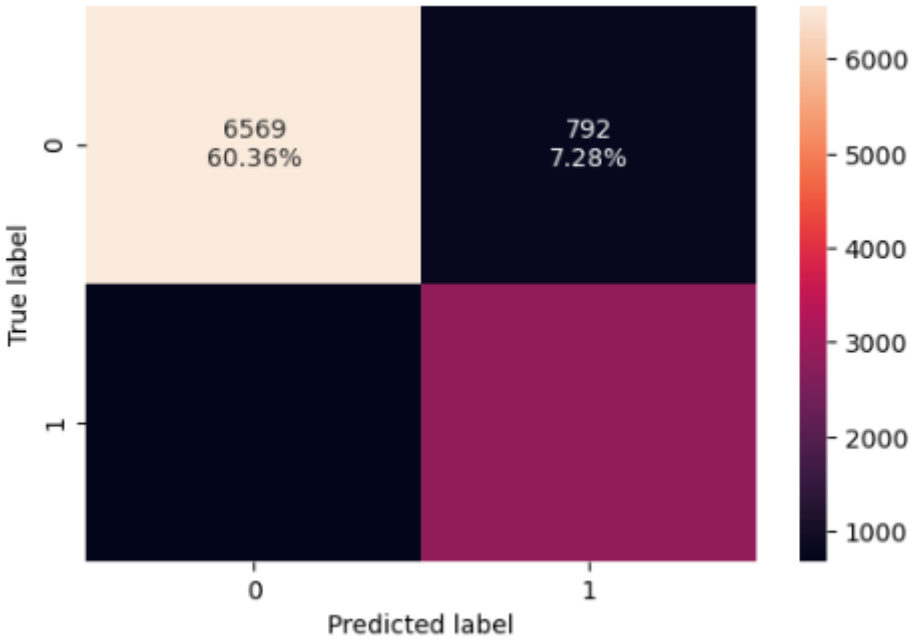


Figure 86 confusion matrix

	Accuracy	Recall	Precision	F1
0	0.86447	0.80608	0.78188	0.79379

Figure 87 Model Performance on Test set

- There is a huge disparity in performance of model on training set and test set, which suggests that the model is overfiiting.

Let's use pruning techniques to try and reduce overfitting.

11.3 Decision Tree (Pre-pruning)

Using GridSearch for Hyperparameter tuning of our tree model

- Hyperparameter tuning is also tricky in the sense that there is no direct way to calculate how a change in the hyperparameter value will reduce the loss of your model, so we usually resort to experimentation. i.e we'll use Grid search
- Grid search is a tuning technique that attempts to compute the optimum values of hyperparameters.
- It is an exhaustive search that is performed on a the specific parameter values of a model.

- The parameters of the estimator/model used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', max_depth=2, max_leaf_nodes=50,
                       min_samples_split=10, random_state=1)
```

11.3.1 Model Performance on Training set

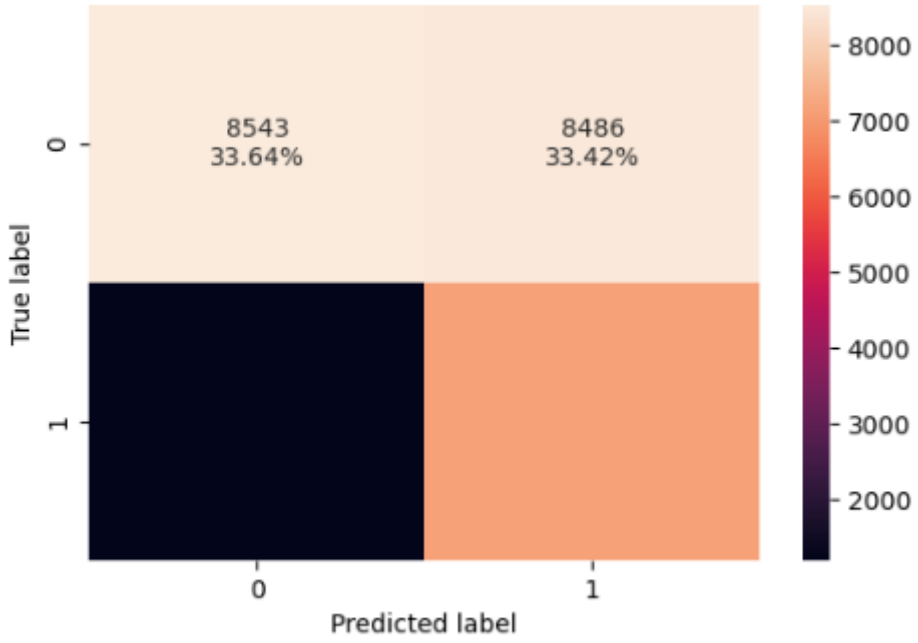


Figure 88 confusion matrix

	Accuracy	Recall	Precision	F1
0	0.61854	0.85651	0.45773	0.59662

Figure 89 Model Performance on Training set

11.3.2 Model Performance on Test set

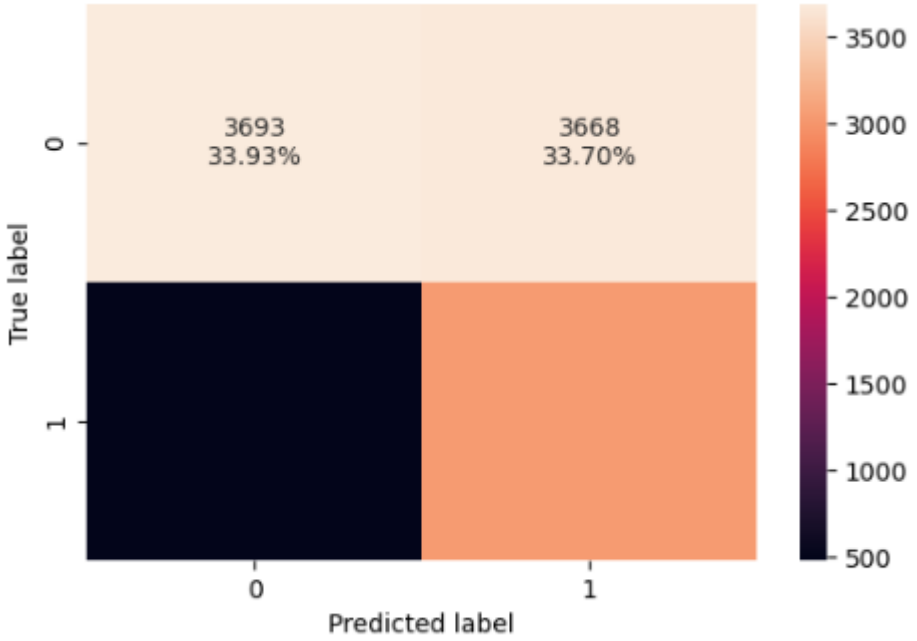


Figure 90 confusion matrix

	Accuracy	Recall	Precision	F1
0	0.61840	0.86229	0.45295	0.59392

Figure 91 Model Performance on Test Set

The model is giving a generalized result now since the recall scores on both the train and test data are coming to be around 0.96 which shows that the model is able to generalize well on unseen data.

11.3.3 Pre-Pruned Tree

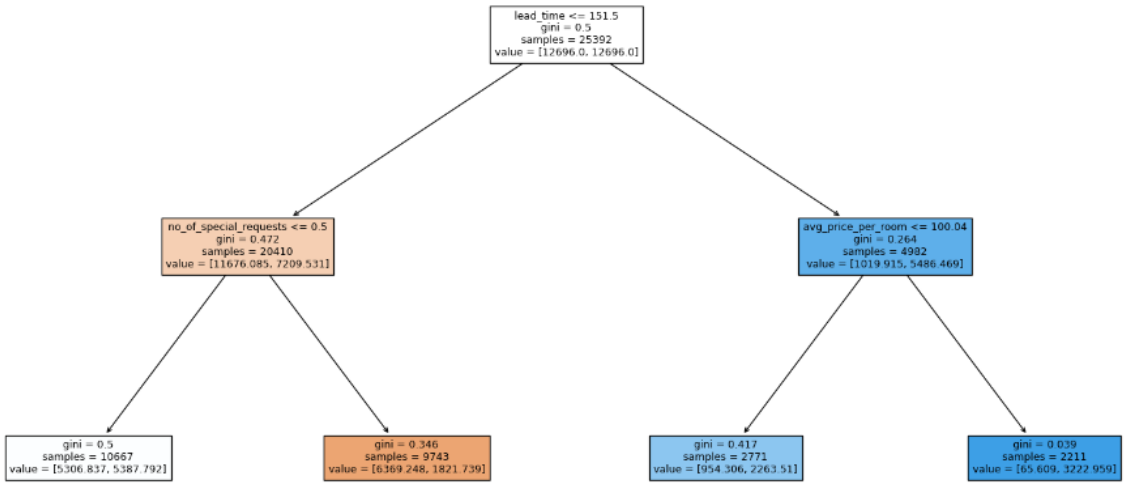


Figure 92 Pre-Pruned Tree

11.3.4 Text report showing the rules of a decision tree

```
|--- lead_time <= 151.50
|   |--- no_of_special_requests <= 0.50
|   |   |--- weights: [5306.84, 5387.79] class: 1
|   |--- no_of_special_requests >  0.50
|   |   |--- weights: [6369.25, 1821.74] class: 0
|--- lead_time >  151.50
|   |--- avg_price_per_room <= 100.04
|   |   |--- weights: [954.31, 2263.51] class: 1
|   |--- avg_price_per_room >  100.04
|   |   |--- weights: [65.61, 3222.96] class: 1
```

Figure 93 Text report of Pre- Pruned Tree

Observations from the pre-pruned tree:

Using the above extracted decision rules we can make interpretations from the decision tree model like:

*Lead time ≤ 151.50 & special requests ≤ 0.50: Predict class 1, with fairly balanced weights between the two classes.

- Lead time ≤ 151.50 & special requests > 0.50: Predict class 0, strongly supported by the weights.
- Lead time > 151.50 & avg_price_per_room ≤ 100.04: Predict class 1, but weights favor class 0.
- Lead time > 151.50 & avg_price_per_room > 100.04: Predict class 1, despite class 0 having much higher weight.
- Lead Time is a critical factor, with shorter lead times tending to favor class 1, while higher lead times show more variability based on the room price.

- Number of Special Requests also plays a significant role for short lead times, and if there are any requests, it tends to favor class 0.
- Average Price Per Room impacts predictions for longer lead times, with lower prices generally leaning toward class 1.

Interpretations from other decision rules can be made similarly

11.3.5 Importance of Features in the Tree Building

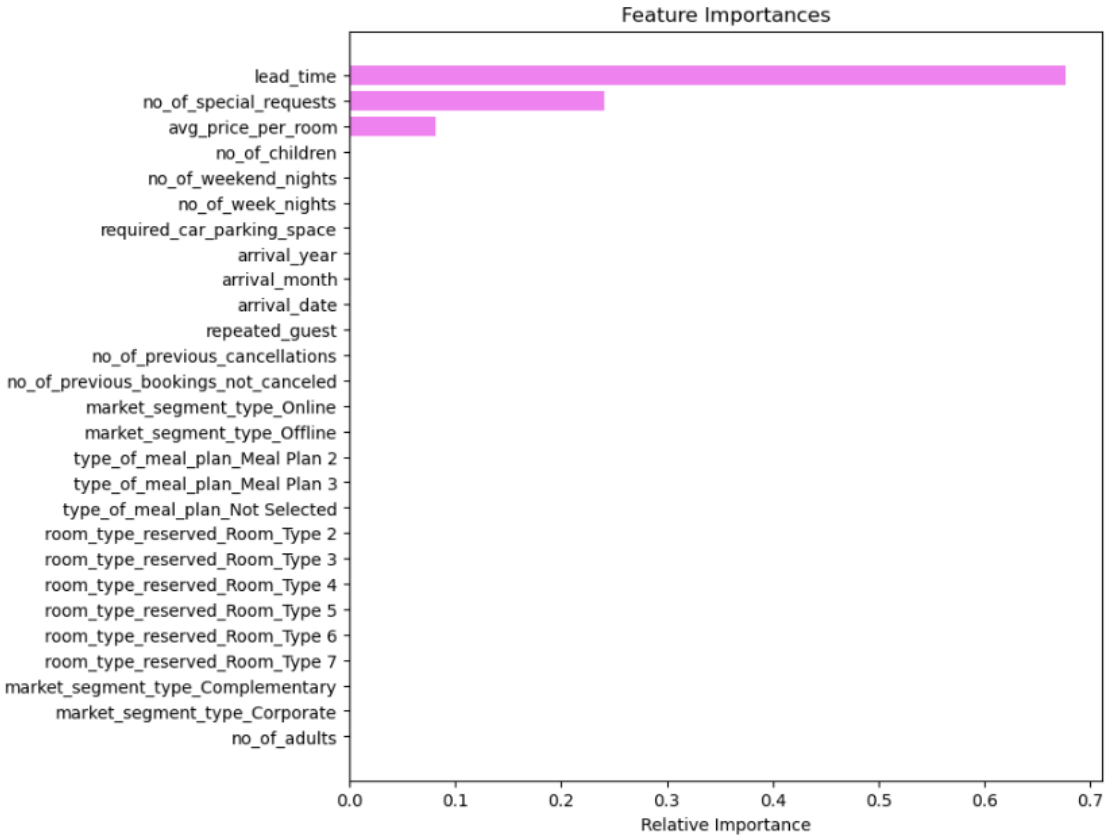


Figure 94 Feature Importance of pre -pruned Tree

According to the graph, lead time is the most important feature.

11.4 Decision Tree (Post pruning)

The DecisionTreeClassifier provides parameters such as min_samples_leaf and max_depth to prevent a tree from overfiting. Cost complexity pruning provides another option to control the size of a tree. In DecisionTreeClassifier, this pruning technique is parameterized by the cost complexity parameter, ccp_alpha. Greater values of ccp_alpha increase the number of nodes pruned. Here we only show the effect of ccp_alpha on regularizing the trees and how to choose a ccp_alpha based on validation scores.

11.4.1 Total impurity of leaves vs effective alphas of pruned tree

Minimal cost complexity pruning recursively finds the node with the "weakest link". The weakest link is characterized by an effective alpha, where the nodes with the smallest effective alpha are pruned first. To get an idea of what values of ccp_alpha could be appropriate, scikit-learn provides DecisionTreeClassifier.cost_complexity_pruning_path that returns the effective alphas and the corresponding total leaf impurities at each step of the pruning process. As alpha increases, more of the tree is pruned, which increases the total impurity of its leaves.

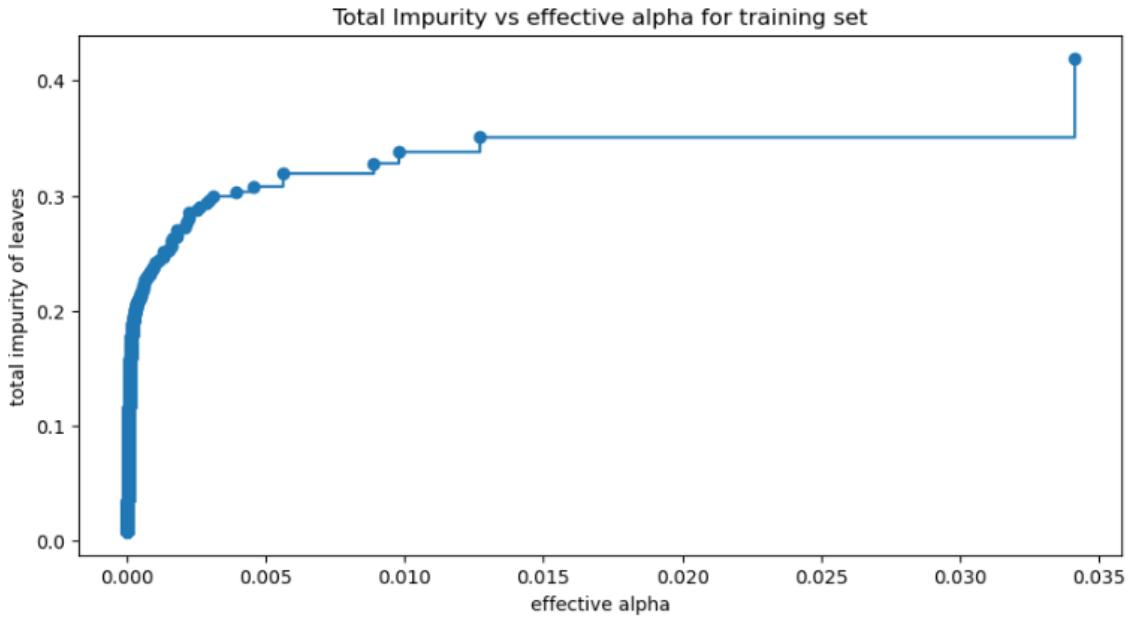


Figure 95 Total Impurity Vs Effective Alpha For Training Set

This plot helps in choosing the optimal **alpha** value for cost-complexity pruning. A smaller alpha may lead to overfitting, while a larger alpha oversimplifies the tree. The ideal value of alpha would lie before the final large jump, where the impurity increases sharply without any significant gain in generalization. This ensures a balanced model with enough complexity to capture the data patterns without overfitting.

11.4.2 Decision Tree using effective alphas

Next, we train a decision tree using the effective alphas. The last value in ccp_alphas is the alpha value that prunes the whole tree, leaving the tree, clfs[-1], with one node.

Number of nodes in the last tree is: 1 with ccp_alpha: 0.08117914389136943

For the remainder, we remove the last element in clfs and ccp_alphas, because it is the trivial tree with only one node. Here we show that the number of nodes and tree depth decreases as alpha increases.

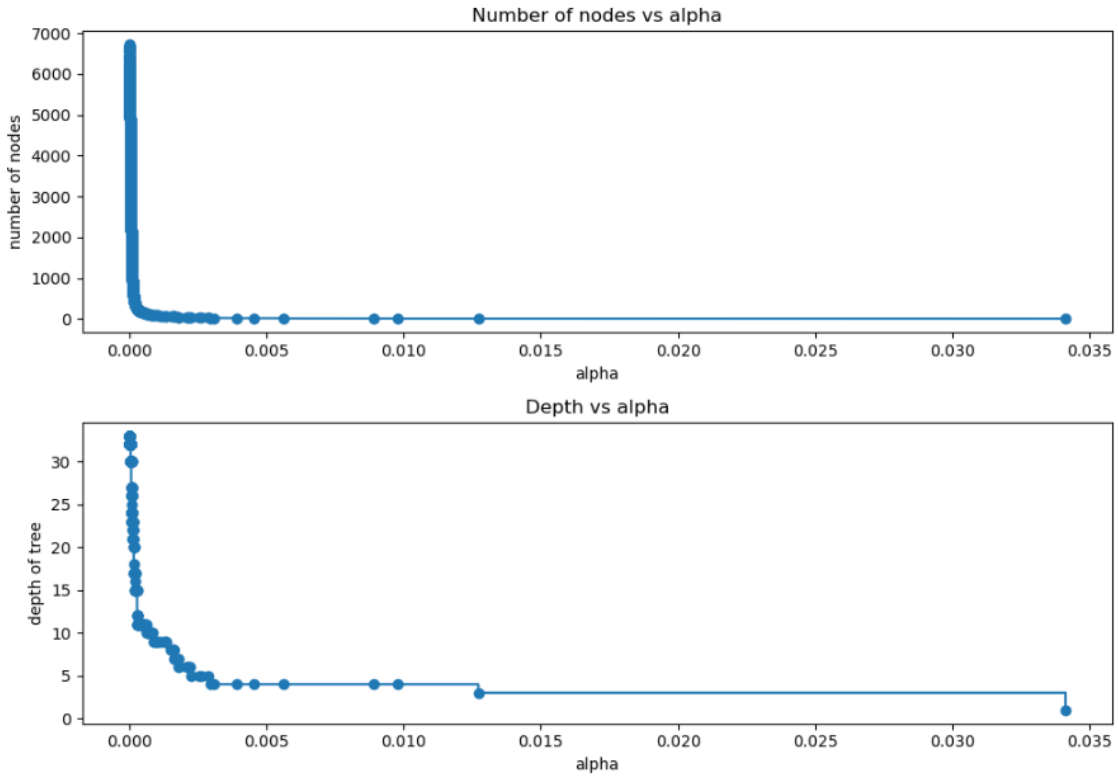


Figure 96 No.Of Nodes Vs Alpha And Depth Vs Alpha

1. **Number of Nodes vs Alpha**:
    - As alpha increases from 0 to approximately 0.005, the number of nodes decreases sharply.
    - Beyond 0.005, the number of nodes levels off and remains relatively constant.
2. **Depth vs Alpha**:
    - Similarly, the depth decreases significantly as alpha increases from 0 to around 0.005.
    - After this point, the depth stabilizes and shows little change with further increases in alpha.

Interpretation:

- **Initial Sensitivity**: Both the number of nodes and depth are highly sensitive to changes in alpha at lower values (0 to 0.005).
- **Stabilization**: Once alpha exceeds 0.005, further increases have minimal impact on these variables, indicating a stabilization point.

Figure 97 Recall vs alpha for training and testing sets

Observations:

1. **Recall for Training Set**:
    - The recall value for the training set, marked with blue squares, starts high but drops sharply as alpha increases from 0 to around 0.005.
    - After this initial drop, the recall value levels off and remains relatively stable as alpha continues to increase.
2. **Recall for Testing Set**:
    - The recall value for the testing set, marked with orange circles, follows a similar pattern to the training set.
    - It also starts high, decreases sharply at low alpha values, and then stabilizes beyond alpha = 0.005.

Interpretation:

- **Initial Sensitivity**: Both the training and testing recall values are highly sensitive to changes in alpha at lower values (0 to 0.005).
- **Stabilization**: Once alpha exceeds 0.005, further increases have minimal impact on recall, indicating a stabilization point.

11.4.3 Decision tree classifier

DecisionTreeClassifier(ccp_alpha=8.046650001478856e-05, class_weight='balanced',

random_state=1)

11.4.3.1 Model Performance on Training set

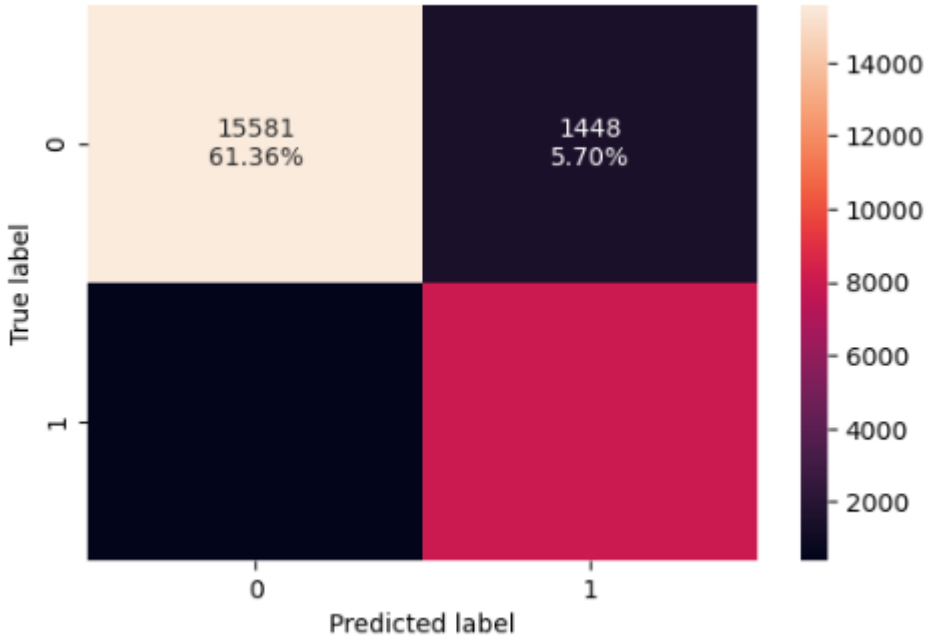


Figure 98 confusion matrix

	Accuracy	Recall	Precision	F1
0	0.92698	0.95145	0.84604	0.89566

Figure 99 Model Performance on Training set

11.4.3.2 Model Performance on Test set

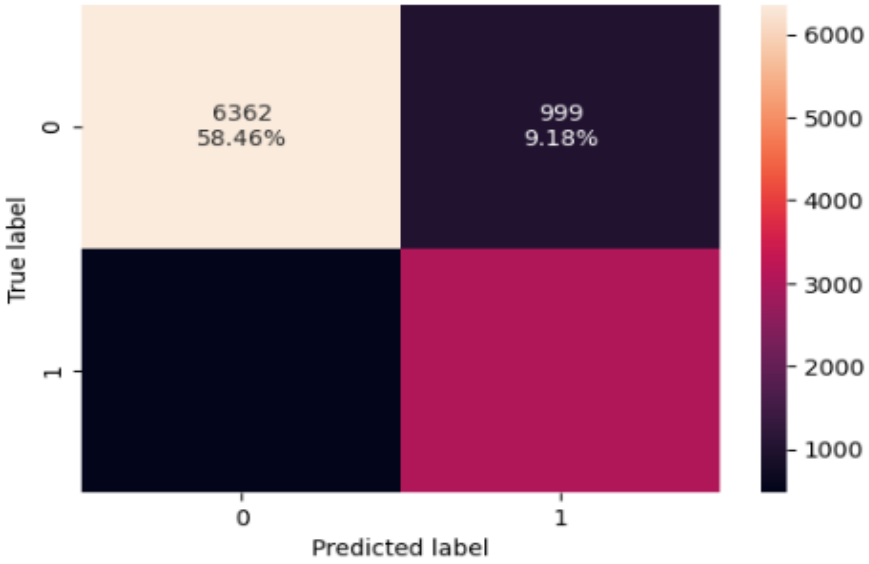


Figure 100 confusion matrix

	Accuracy	Recall	Precision	F1
0	0.86373	0.86258	0.75254	0.80381

Figure 101 Model Performance on Test set

- In the post-pruned tree also, the model is giving a generalized result since the recall scores on both the train and test data are coming to be around 0.95 and 0.86 respectively which shows that the model is able to generalize well on unseen data.

11.4.4 Post - Pruned Tree

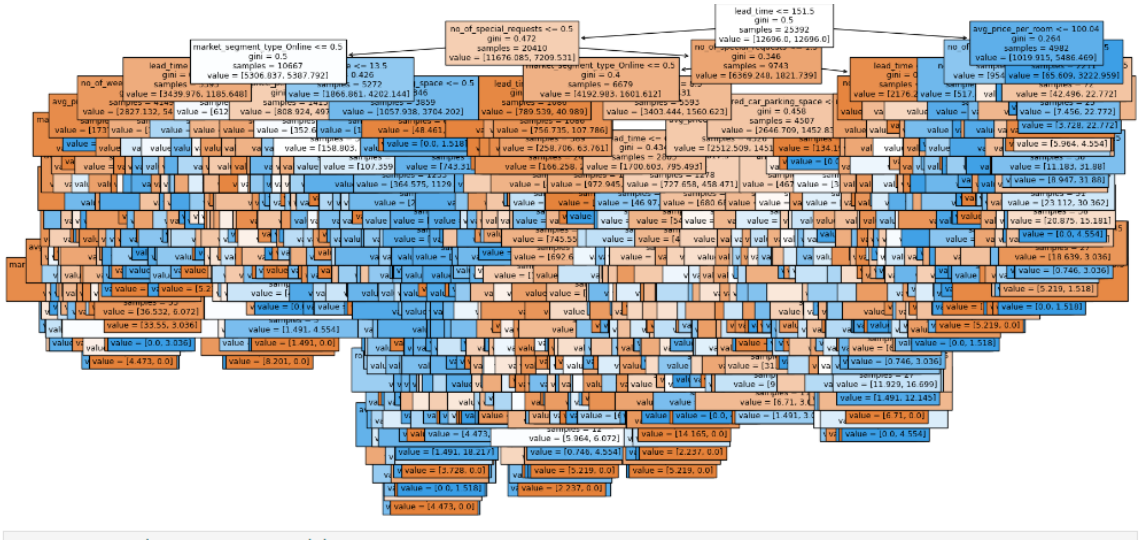


Figure 102 Post Pruned Tree

11.4.5 Text report for Post Pruned Tree

```
|--- lead_time <= 151.50
|   |--- no_of_special_requests <= 0.50
|   |   |--- market_segment_type_Online <= 0.50
|   |   |   |--- lead_time <= 90.50
|   |   |   |   |--- no_of_weekend_nights <= 0.50
|   |   |   |   |   |--- avg_price_per_room <= 196.50
|   |   |   |   |   |   |--- market_segment_type_Offline <= 0.50
|   |   |   |   |   |   |   |--- lead_time <= 16.50
|   |   |   |   |   |   |   |   |--- avg_price_per_room <= 68.50
|   |   |   |   |   |   |   |   |   |--- weights: [207.26, 10.63] class: 0
|   |   |   |   |   |   |   |   |--- avg_price_per_room >  68.50
|   |   |   |   |   |   |   |   |   |--- arrival_date <= 29.50
|   |   |   |   |   |   |   |   |   |   |--- no_of_adults <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |--- no_of_adults >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |--- arrival_date >  29.50
|   |   |   |   |   |   |   |   |   |   |--- avg_price_per_room <= 115.50
```

Figure 103 Text report for Post Pruned Tree

- We can see that the observation we got from the pre-pruned tree is also matching with the decision tree rules of the post pruned tree.

Feature Importance of Post Pruned Tree:

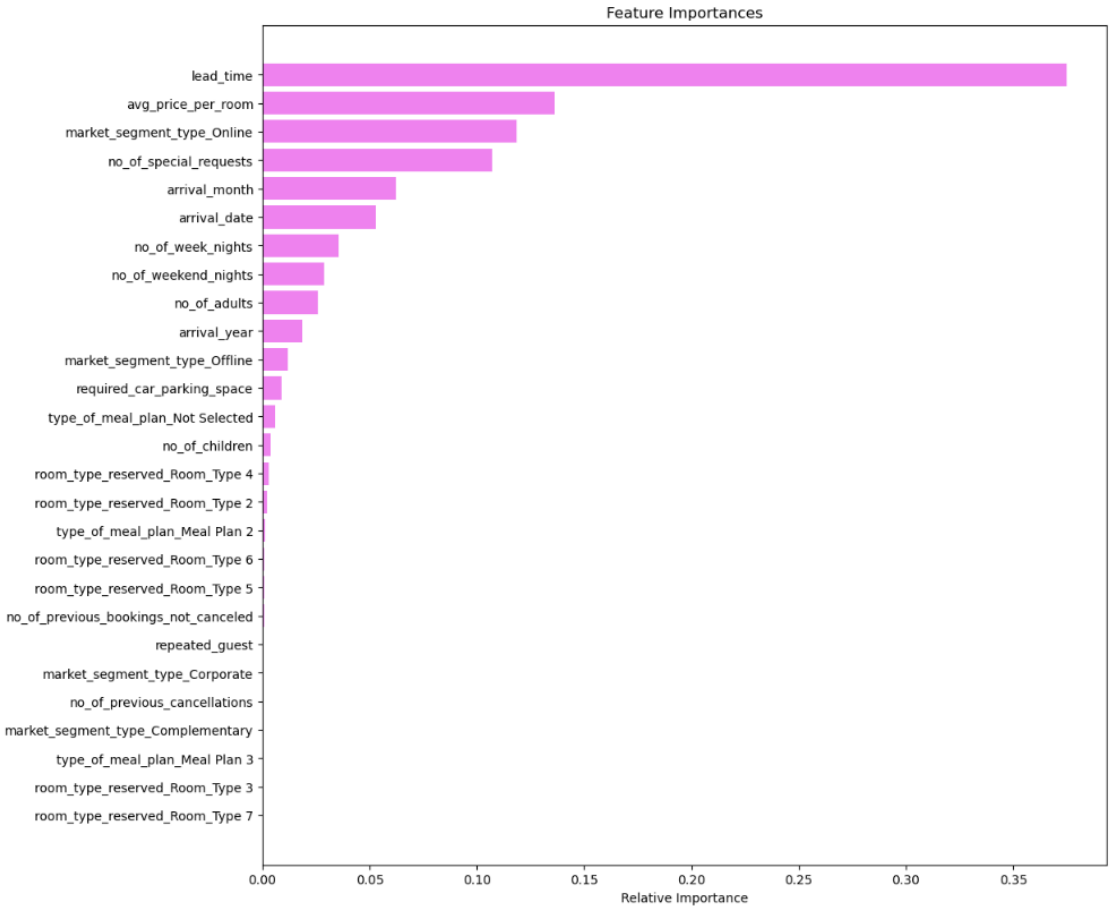


Figure 104 Feature Importance of Post Pruned Tree

11.5 Comparison of Models and Final Model Selection

11.5.1 Training performance comparison

	Decision Tree without class_weight	Decision Tree with class_weight	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.99421	0.99311	0.61854	0.92698
Recall	0.98661	0.99510	0.85651	0.95145
Precision	0.99578	0.98415	0.45773	0.84604
F1	0.99117	0.98960	0.59662	0.89566

Figure 105 Training performance comparison

- **Class Weights**: Applying class weights slightly reduces accuracy but improves recall and precision, indicating better handling of class imbalance.
- **Pruning**: Pre-pruning significantly lowers performance metrics, while post-pruning improves them, suggesting that post-pruning is more effective in maintaining model performance.

11.5.2 Testing performance comparison

	Decision Tree without class_weight	Decision Tree with class_weight	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.87108	0.86447	0.61840	0.86373
Recall	0.81034	0.80608	0.86229	0.86258
Precision	0.79521	0.78188	0.45295	0.75254
F1	0.80270	0.79379	0.59392	0.80381

Figure 106 Testing performance comparison

- **Class Weights**: Applying class weights slightly reduces accuracy, recall, precision, and F1 score, indicating a trade-off when handling class imbalance.
- **Pruning**: Pre-pruning significantly lowers performance metrics, while post-pruning improves them, suggesting that post-pruning is more effective in maintaining model performance.

12. MODEL PERFORMANCE COMPARISON AND FINAL MODEL SELECTION

12.1 Training Performance

Training performance comparison:

	Logistic Regression-default Threshold (0.5)	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold	K Nearest Neighbor k=3	Naive Bayes	Decision Tree without class_weight	Decision Tree with class_weight	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.80541	0.79289	0.80128	0.91588	0.40970	0.99421	0.99311	0.61854	0.92698
Recall	0.63255	0.73562	0.69939	0.85412	0.96281	0.98661	0.99510	0.85651	0.95145
Precision	0.73903	0.66870	0.69789	0.88634	0.35425	0.99578	0.98415	0.45773	0.84604
F1	0.68166	0.70056	0.69864	0.86993	0.51793	0.99117	0.98960	0.59662	0.89566

Figure 107 Training Performance comparison

12.2 Test Set Performance

Test set performance comparison:

	Logistic Regression-default Threshold (0.5)	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold	K Nearest Neighbor k=3	Naive Bayes	Decision Tree without class_weight	Decision Tree with class_weight	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.80465	0.79601	0.80364	0.85271	0.40614	0.87108	0.86447	0.61840	0.86373
Recall	0.63089	0.73935	0.70386	0.75383	0.96621	0.81034	0.80608	0.86229	0.86258
Precision	0.72900	0.66667	0.69381	0.78295	0.34913	0.79521	0.78188	0.45295	0.75254
F1	0.67641	0.70113	0.69880	0.76812	0.51292	0.80270	0.79379	0.59392	0.80381

Figure 108 Testing performance comparison

Observation:

- **Decision Trees (both weighted and pruned)** deliver consistently high performance across all key metrics on both the training and test sets, making them the most robust model.
- **KNN (k=3)** strikes a good balance between precision and recall, with competitive accuracy, making it a reliable alternative.
- **Naive Bayes** achieves very high recall, but it faces significant challenges with precision, particularly in handling false positives.
- **Logistic Regression** performs adequately, but its effectiveness varies depending on threshold adjustments, with trade-offs between precision and recall.

Insights:

- Among these models, **Decision Trees (both weighted and pruned)** are the best. They exhibit the highest performance across all metrics (accuracy, precision, recall, and F1-score) on both the training and test sets, making them the most reliable and robust for predicting hotel booking cancellations.
- While KNN and Logistic Regression are decent alternatives, and Naive Bayes excels in recall, **Decision Trees** provide the best overall balance of predictive power, interpretability, and generalization.

13 ACTIONABLE INSIGHTS AND RECOMMENDATIONS:

13.1. Recommendation

1. Customer Loyalty (Repeated Guests):
    - Repeated guests have an extremely low cancellation rate (~1.7%), indicating that they are less likely to cancel bookings.
    - Action: Utilize customer loyalty programs or incentives to encourage repeat bookings, as this group is highly stable.
2. Booking Channel Impact (Market Segment):
    - The online segment has a relatively higher cancellation rate (~36.5%) compared to offline and corporate bookings.
    - Action: Implement stricter cancellation policies or provide flexible booking options with insurance for online bookings to reduce cancellations.
3. Booking Month:
    - High cancellation rates are observed during July (45%), August (39%), and June (40.3%). In contrast, December (13.3%) and January (2.4%) have low cancellation rates.
    - Action: Introduce special offers or discounts during months with higher cancellation rates to reduce cancellations. Additionally, focus marketing efforts on the more stable months, like December and January.

4. Number of Family Members:

- Larger families (4+ members) have higher cancellation rates (43.6% for families with 4 members).
- Action: Offer family-friendly packages or flexible booking policies that cater to larger groups to reduce cancellations.

13.2. Actionable Insights

- **Decision Trees Perform Best**:
    - The **Decision Tree with class weights** consistently delivers the best performance across both training and test sets, with high accuracy, precision, recall, and F1 scores.
    - Balancing class weights helps minimize bias toward any particular class, resulting in more reliable predictions.
    - The **post-pruned decision tree** also performs strongly on the test set by reducing overfitting, as seen by a controlled drop in training accuracy compared to the non-pruned version.
    - In contrast, **pre-pruned decision trees** show a significant performance drop, emphasizing the need for careful application of pruning techniques.
- **KNN (k=3): A Balanced Model**:
    - **KNN (k=3)** offers a solid balance between accuracy, precision, and recall, with consistent performance across both training and test sets, indicating good generalization.
    - However, due to the computational intensity and memory requirements of KNN, it may not be ideal for very large datasets compared to faster algorithms.
- **Naive Bayes: High Recall, Low Precision**:
    - **Naive Bayes** excels in recall, identifying almost all positive instances, making it suitable for cases where missing positive instances is costly (e.g., disease detection).
    - Despite high recall, the model suffers from low precision, leading to a high number of false positives, which could be problematic in applications where precision is more critical.
- **Threshold Adjustments in Logistic Regression**:
    - **Logistic Regression** allows flexibility through threshold adjustment. A lower threshold (e.g., 0.37) improves recall, while a higher threshold (e.g., 0.42) offers a better balance between precision and recall.

- While Logistic Regression performs well, it tends to lag slightly behind Decision Trees and KNN in terms of accuracy and F1 score. Nonetheless, it remains a strong baseline model due to its interpretability and simplicity.