Machine Learning 1

INN HOTELS

Prepared by
KARTHICK R

JUNE
2025

Table of Contents

Table of Contents

List of figures

List of figures

List of figures

List of figures

List of figures

List of Tables

List of Tables

List of Tables

Objective

- The INN Hotels Group, a chain of hotels based in Portugal, is experiencing a growing challenge with a high volume of booking cancellations. To address this issue, they have approached our firm for a data-driven solution. The objective of this project is to analyze the historical booking data to identify key factors that significantly influence booking cancellations. Based on these insights, a robust machine learning model will be developed to accurately predict the likelihood of a booking being canceled in advance. This predictive capability will empower the hotel group to implement more effective cancellation and refund policies, optimize resource allocation, and improve overall profitability.

We will be majorly focusing on these problems -

- The INN Hotels Group, a chain of hotels based in Portugal, is experiencing a growing challenge with a high volume of booking cancellations. To address this issue, they have approached our firm for a data-driven solution. The objective of this project is to analyze the historical booking data to identify key factors that significantly influence booking cancellations. Based on these insights, a robust machine learning model will be developed to accurately predict the likelihood of a booking being canceled in advance. This predictive capability will empower the hotel group to implement more effective cancellation and refund policies, optimize resource allocation, and improve overall profitability.

- The analysis will begin with an exploration of booking trends to determine the busiest months of the year, which will help the hotel plan for peak demand periods. Market segmentation will be examined to identify the primary sources of bookings, allowing for more targeted marketing strategies. Additionally, pricing dynamics will be analyzed to understand how room rates vary across different customer segments.

- Understanding the scope of the problem, the project will assess the overall cancellation rate and examine the behavior of repeat guests, who are essential to brand loyalty and revenue stability. Special attention will be given to guests with specific room or service requirements to determine if these requests have any bearing on cancellation behavior.

- A critical part of the analysis will involve identifying the most influential factors contributing to cancellations. These insights will directly inform the development of a robust machine learning model that predicts the likelihood of a booking being canceled. The final solution aims to support strategic decision-making, enhance customer retention, and improve financial performance across the hotel group.

Data overview

- The INN Hotels Group, a chain of hotels based in Portugal, is experiencing a growing challenge with a high volume of booking cancellations. To address this issue, they have approached our firm for a data-driven solution. The objective of this project is to analyze the historical booking data to identify key factors that significantly influence booking cancellations. Based on these insights, a robust machine learning model will be developed to accurately predict the likelihood of a booking being canceled in advance. This predictive capability will empower the hotel group to implement more effective cancellation and refund policies, optimize resource allocation, and improve overall profitability.

We will be majorly focusing on these problems -

- The INN Hotels Group, a chain of hotels based in Portugal, is experiencing a growing challenge with a high volume of booking cancellations. To address this issue, they have approached our firm for a data-driven solution. The objective of this project is to analyze the historical booking data to identify key factors that significantly influence booking cancellations. Based on these insights, a robust machine learning model will be developed to accurately predict the likelihood of a booking being canceled in advance. This predictive capability will empower the hotel group to implement more effective cancellation and refund policies, optimize resource allocation, and improve overall profitability.

Data overview

The dataset contains 36,275 bookings with 19 columns, capturing various aspects of hotel reservations.

- The Booking_ID, type_of_meal_plan, room_type_reserved, market_segment_type, booking_status columns are of object type while the rest columns are in numeric.

Displaying the first 5 rows of the dataset

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

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

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

Table 1: Top five rows of dataset

Checking the shape of the dataset

- The dataset contains 36275 rows and 19 columns

Checking the data types of the columns for the dataset

```
<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 types of the column

- The Booking_ID, type_of_meal_plan, room_type_reserved, market_segment_type, booking_status columns are of object type while the rest columns are in numeric.

Statistical summary of the dataset

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

Table 3: Statistical summary of the dataset

- The average price of avg_price_per_room is 103.42. the maximum price of a room is 540.
- The average no of adults visited are 1.8.
- The maximum no of adults visited are 4 and the maximum no of childrens visited are 10.

Value counts of booking_status

booking_status	proportion
0	0.67236
1	0.32764

dtype: float64

Table 4: Value counts of dataset

Checking for duplicate values

- There are no duplicate values in the data.

Checking for null values

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

dtype: int64

Table 5: Data on null values

- There are no null values in the dataset.

Exploratory Data Analysis

Rubic Problem 1

1.1 What are the busiest months in the hotel?

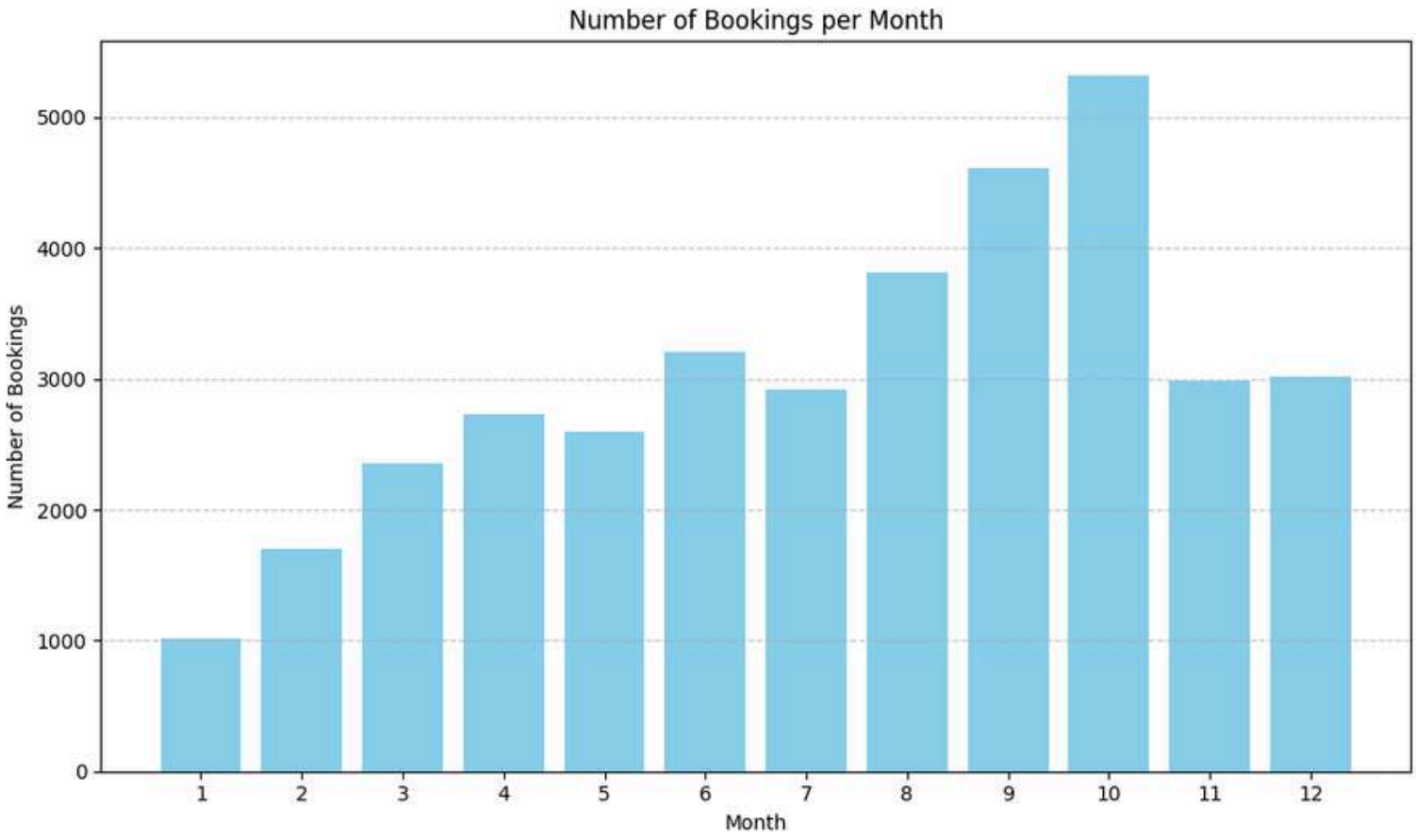


Figure 1: Busiest months

The busiest months for the hotel are:

- October (5,317 bookings)
- September (4,611 bookings)
- August (3,813 bookings)

Problem 2

1.2 Which market segment do most of the guests come from?

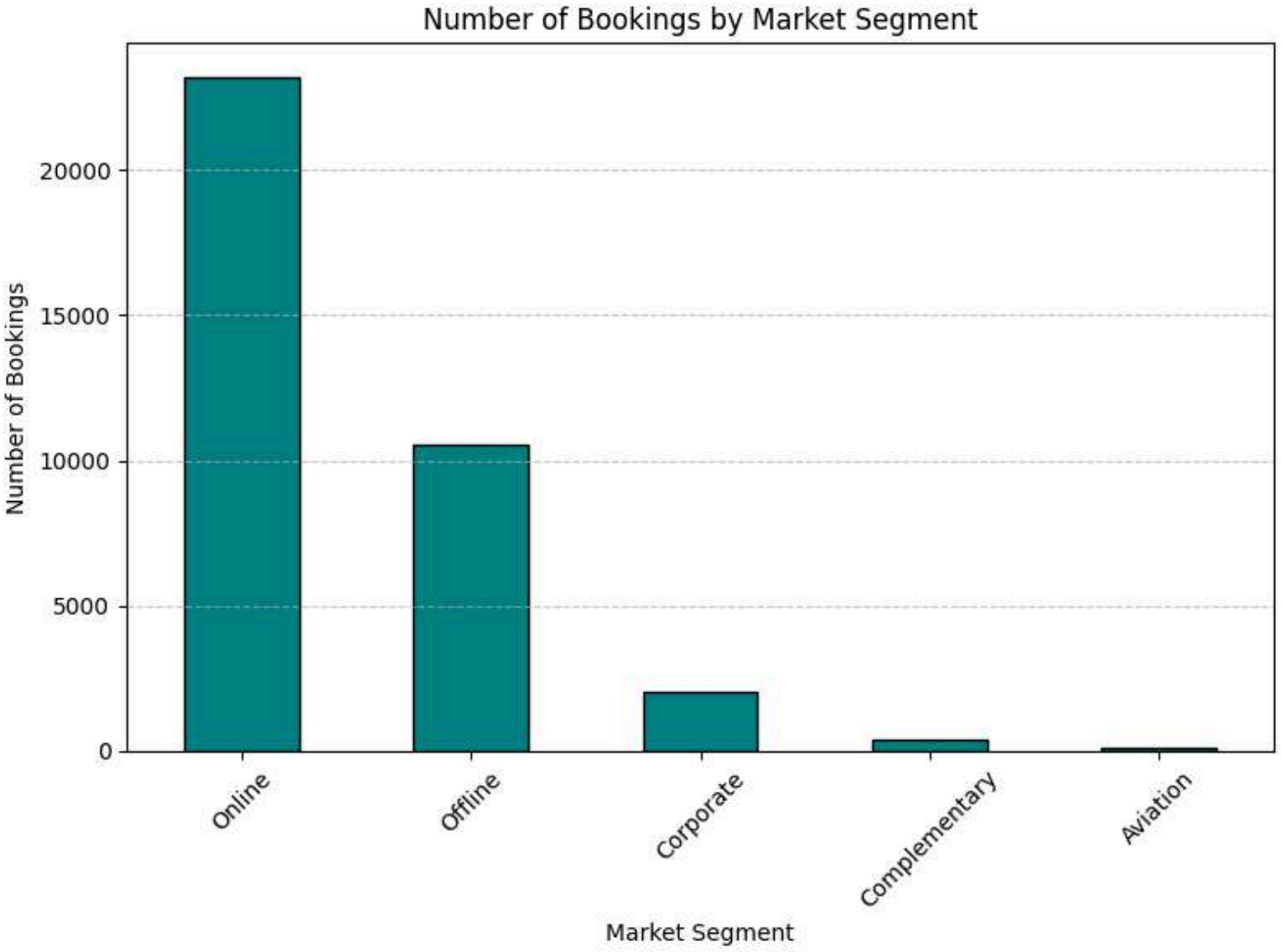


Figure 2: Number of guests from market segments

- Most of the guests come from the Online market segment, with 23,214 bookings.

Problem 3

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

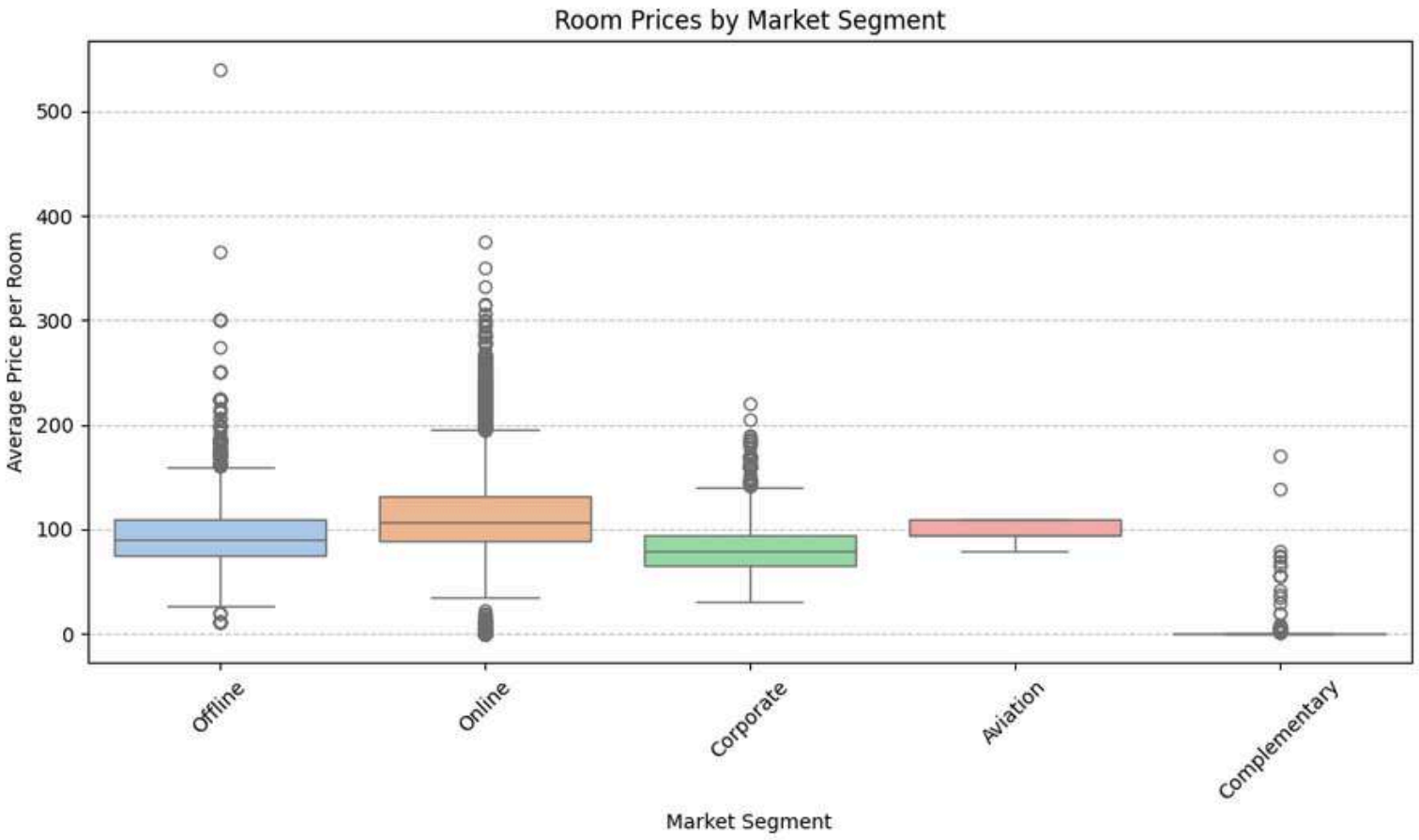


Figure 3: The differences in room prices in different market segments

- Online bookings have the highest average price per room, suggesting demand or higher willingness to pay.
- Offline and Corporate rates are moderately priced.
- Aviation guests tend to have consistently high rates with a narrow price range.
- Complementary segment has near-zero pricing (e.g., free stays for promotions, partners, or staff).

Problem 4

1.4 What percentage of bookings are canceled?

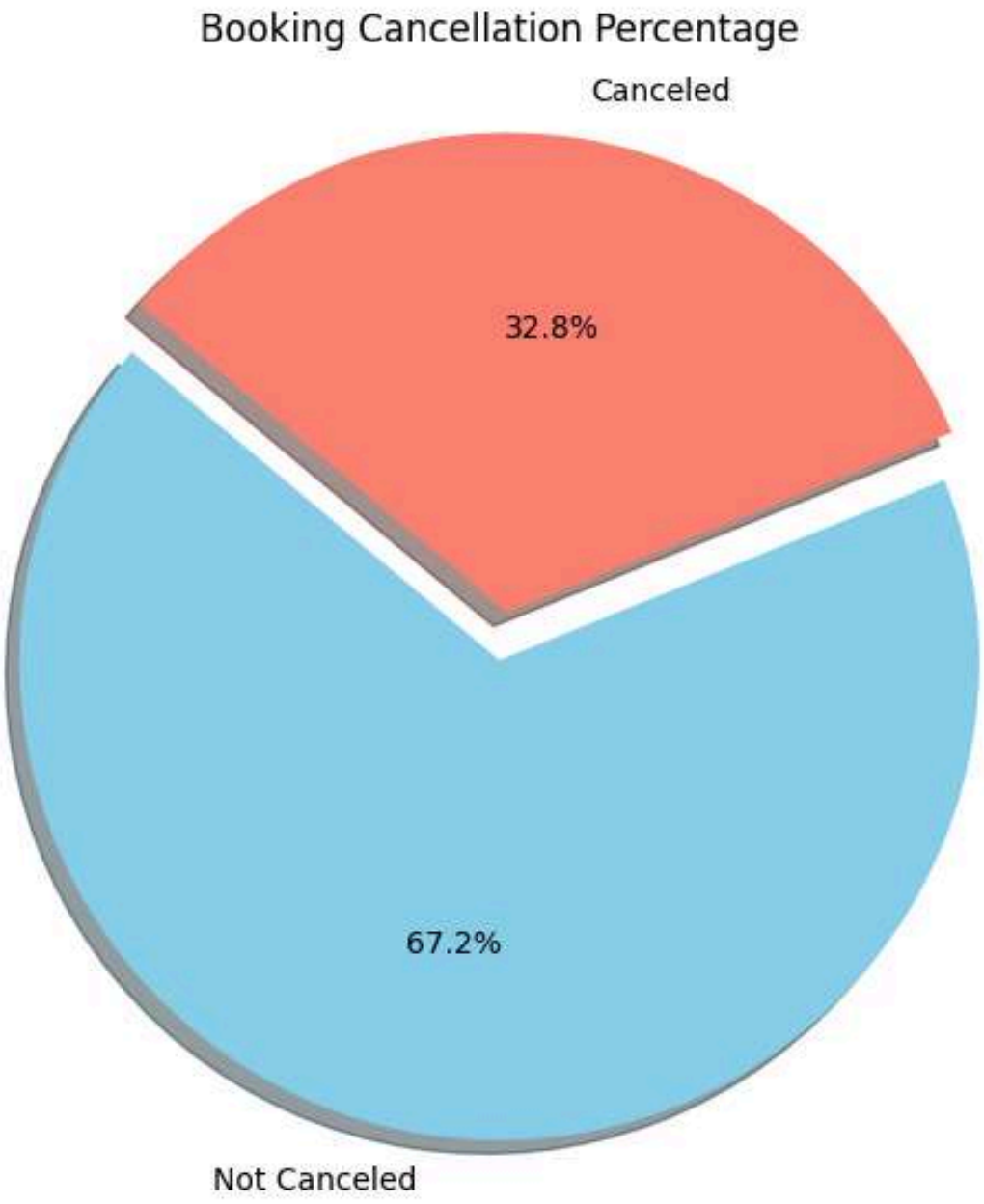


Figure 4: The percentage of bookings are canceled

- Approximately 32.76% of all bookings in the dataset are canceled.

Problem 5

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

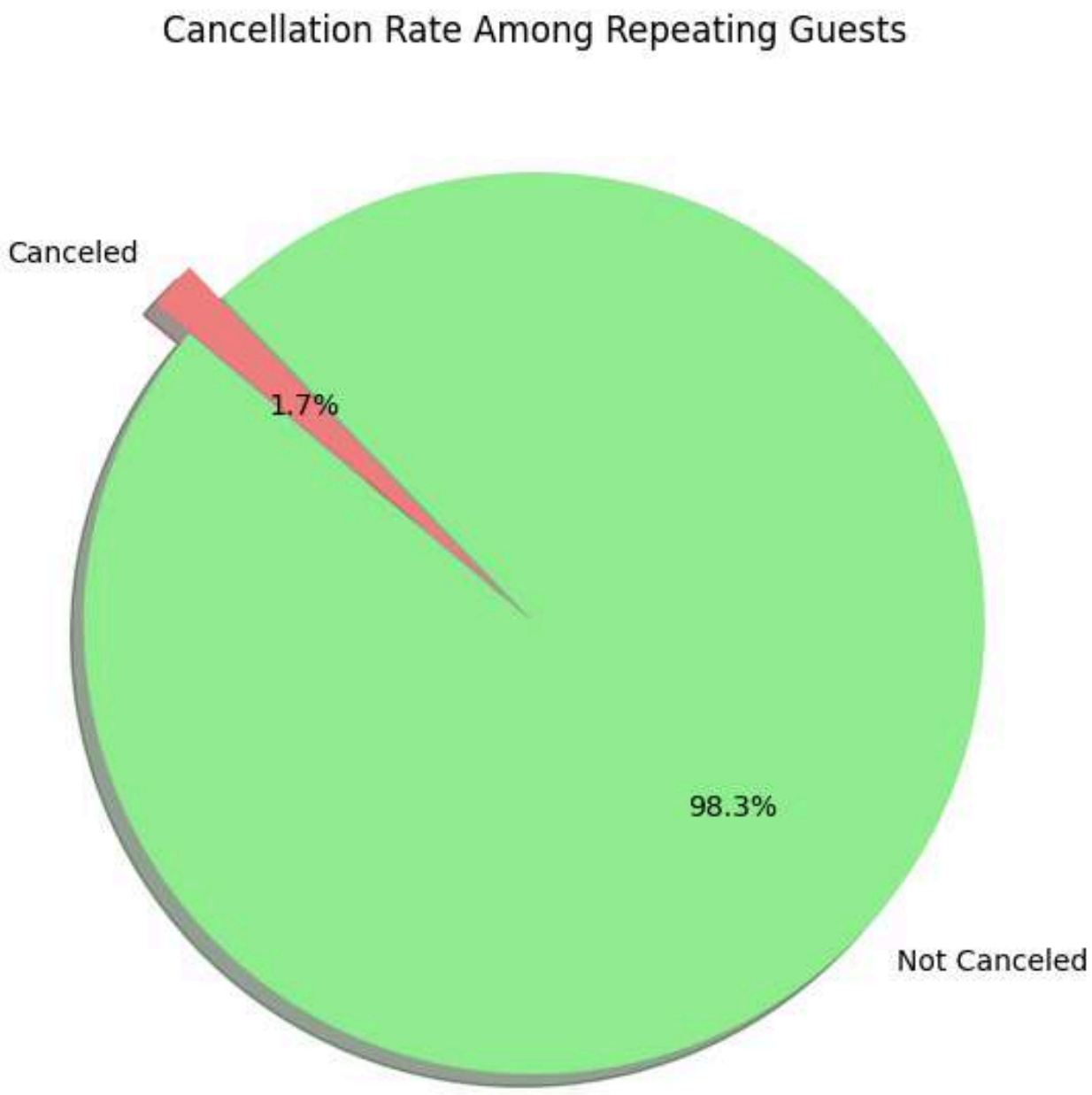


Figure 5: The repeating guests cancelling percentage

- Only about 1.72% of repeating guests cancel their bookings.
- This is significantly lower than the overall cancellation rate (~32.76%), highlighting that repeat guests are more reliable and possibly more loyal.

Problem 6

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

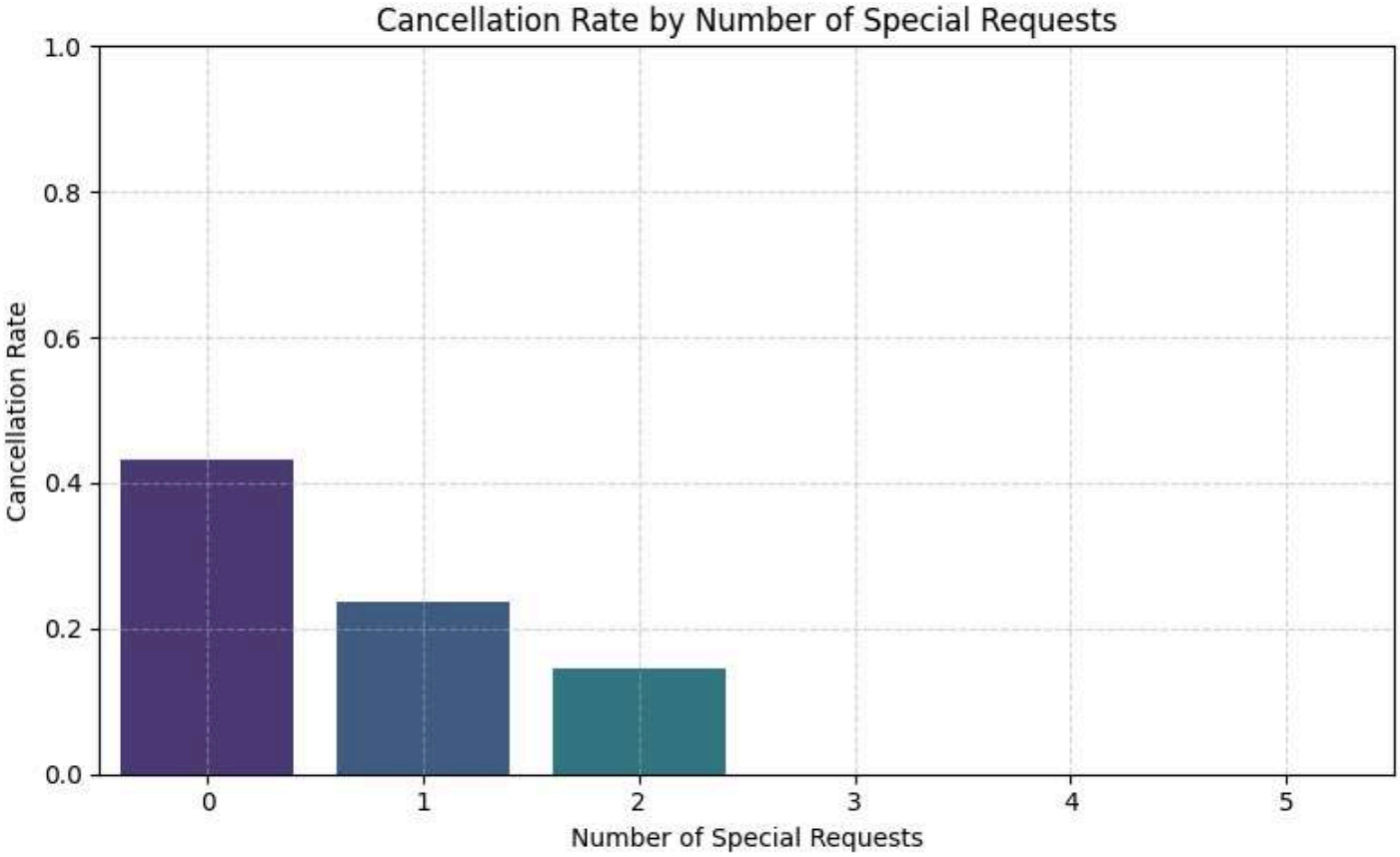


Figure 6: The requirements affecting booking cancellation

The bar chart shows the cancellation rate decreases as the number of special requests increases. This suggests:

- Guests with no special requests are more likely to cancel.
- Guests with more special requests tend to follow through with their bookings, likely due to stronger commitment or specific needs.

- In conclusion, special requirements are negatively associated with cancellations—they reduce the likelihood of a booking being canceled.

Dropping unwanted column

- Removing the **Booking_ID column,** because it doesn't help us to build the model.

Univariate analysis

Analysis on arrival_month

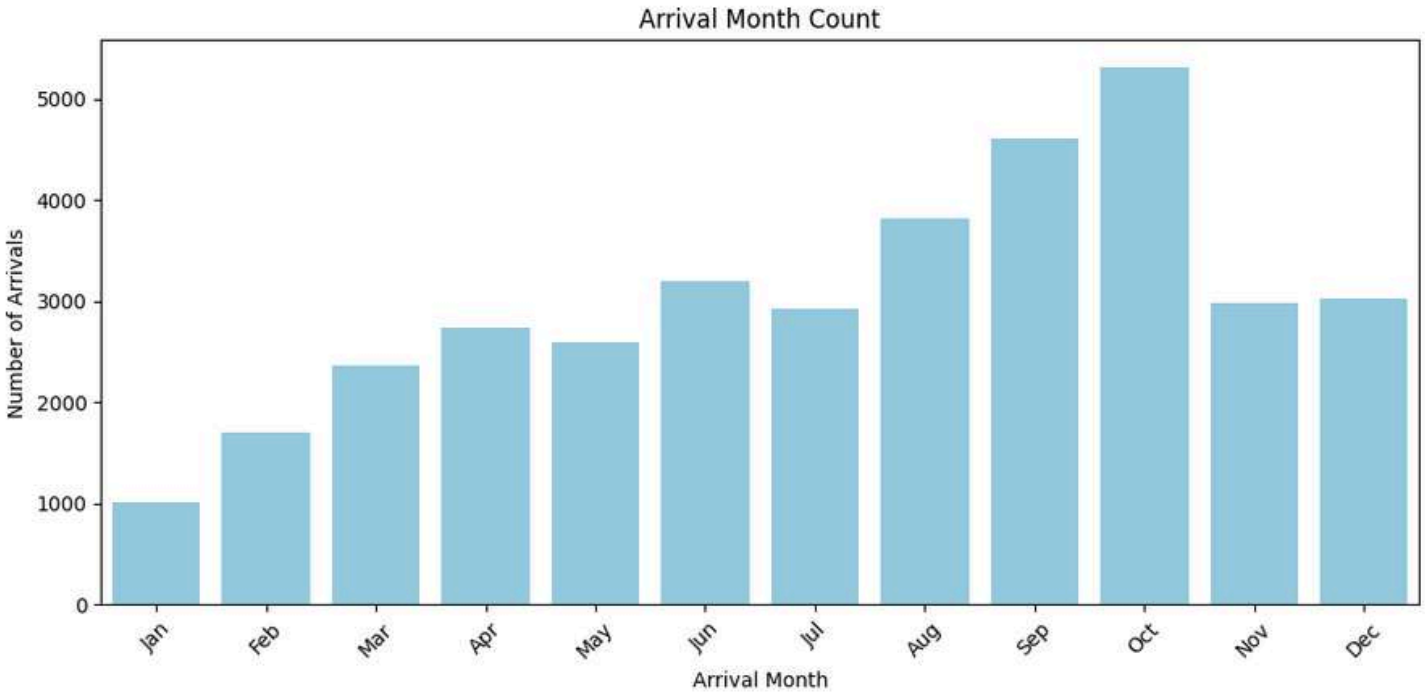


Figure 7: The Analysis on arrival_month

Observations on arrival_month

- October has the highest number of arrivals (~5,300), followed by September and August.
- January has the lowest number of arrivals (~1,000), likely due to post-holiday travel fatigue and winter off-season.
- From February to October, there's a steady increase in arrivals.
- This suggests late summer to early fall is the busiest period, possibly due to holidays, events, or favorable weather.

Analysis on lead_time

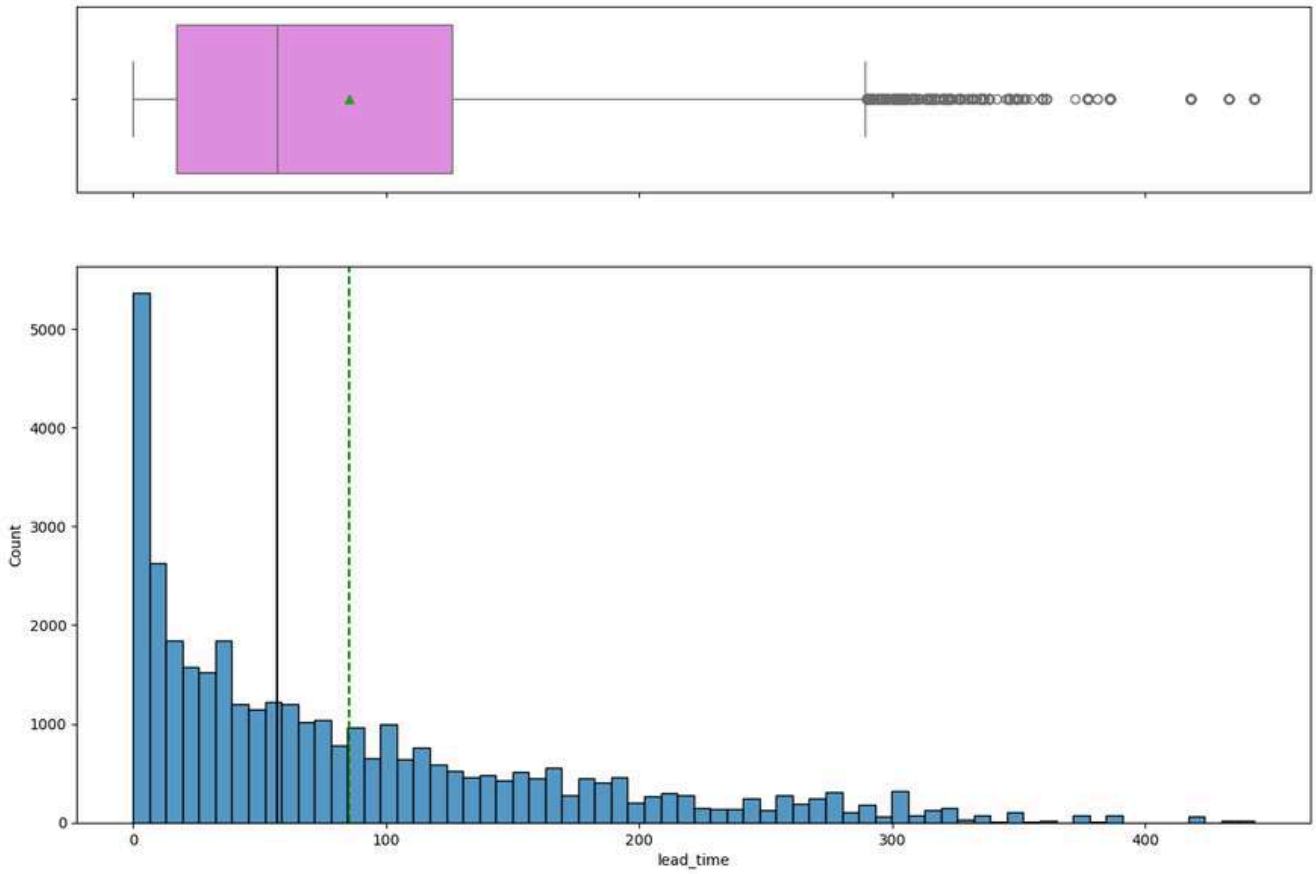


Figure 8: The Analysis on lead_time

Observations on lead_time

- Distribution is Right-Skewed.
- Most bookings are made with a short lead time.
- A large number of bookings occur with 0–50 days notice, especially close to the check-in date.
- Very few bookings are made >200 days in advance.
- The median (black line in boxplot) is around 50 days, meaning half of the bookings are made within 50 days of arrival.
- There are several outliers in lead time (bookings made over 400 days in advance).

Analysis on avg_price_per_room

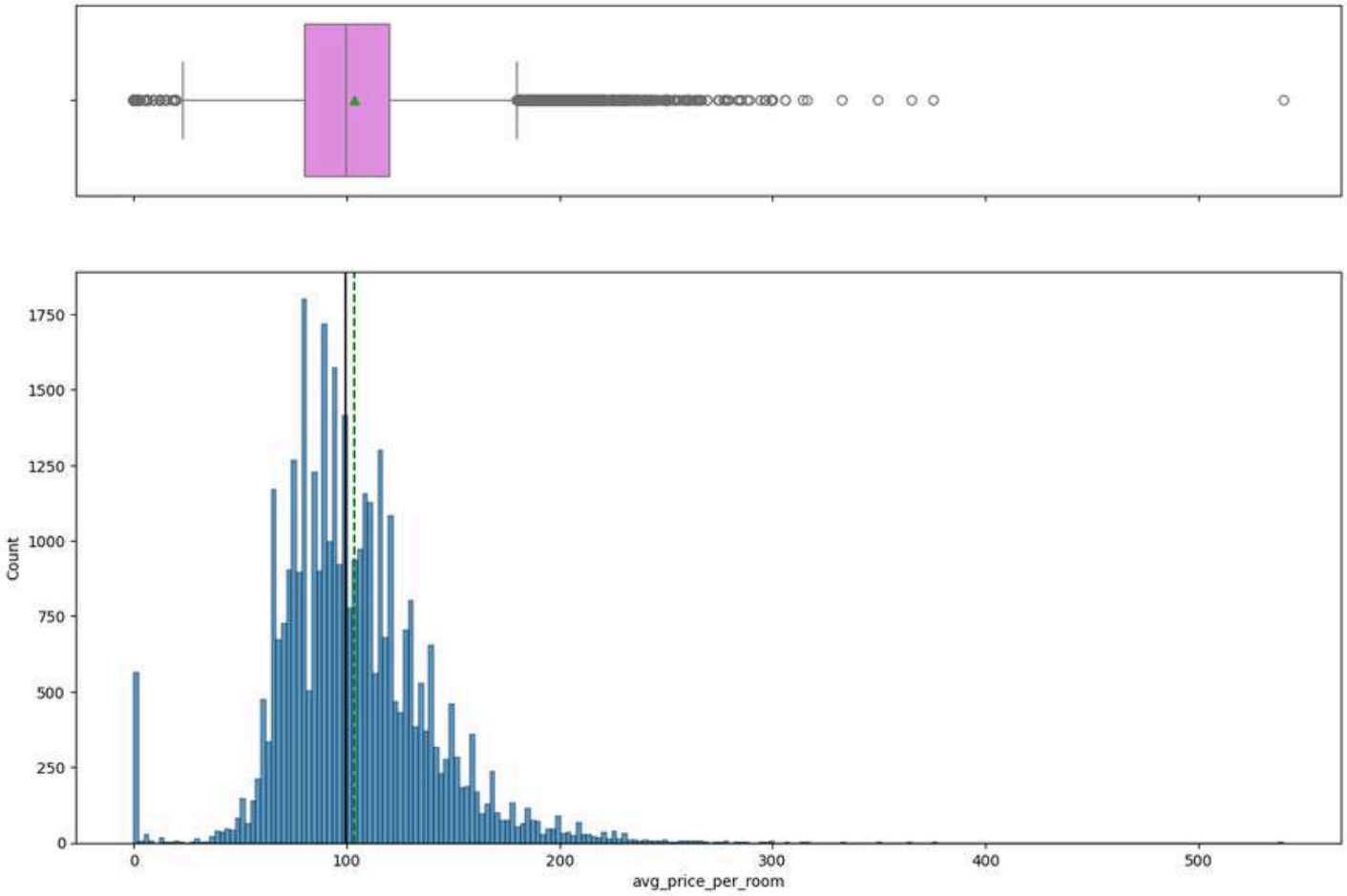


Figure 9: The Analysis on avg_price_per_room

Observations on avg_price_per_room

- The distribution is right-skewed, with most prices clustered between ₹60 and ₹150.
- A long tail extends to the right, with a few bookings priced as high as ₹500+
- The most frequent room rates are centered around ₹100, indicating this is the typical price point.

Analysis on Special guests

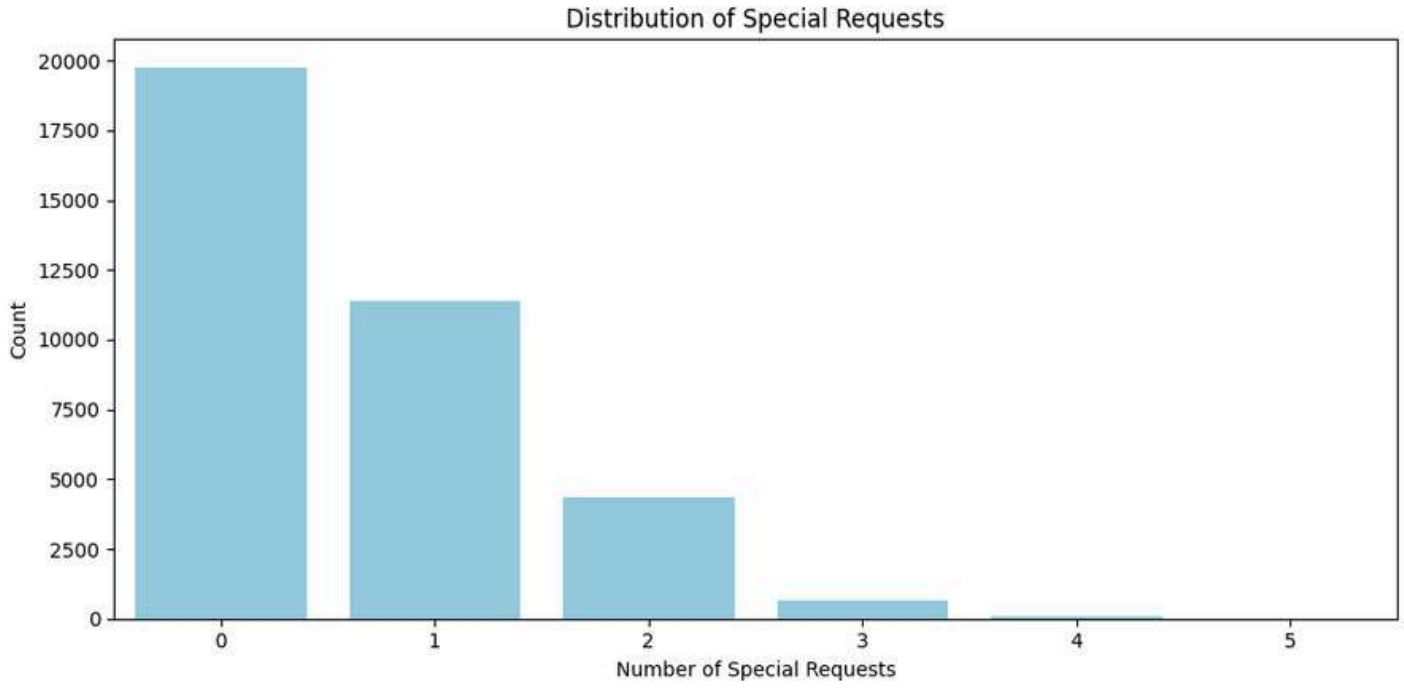


Figure 10: The Analysis on Special guests

Observations on Special guests

- Most of the there are no special guests.
- 11,000 Guests are accompanied with 1 guests.
- The maximum number of 5 special guest are accompanied in a room.

Analysis on type_of_meal_plan

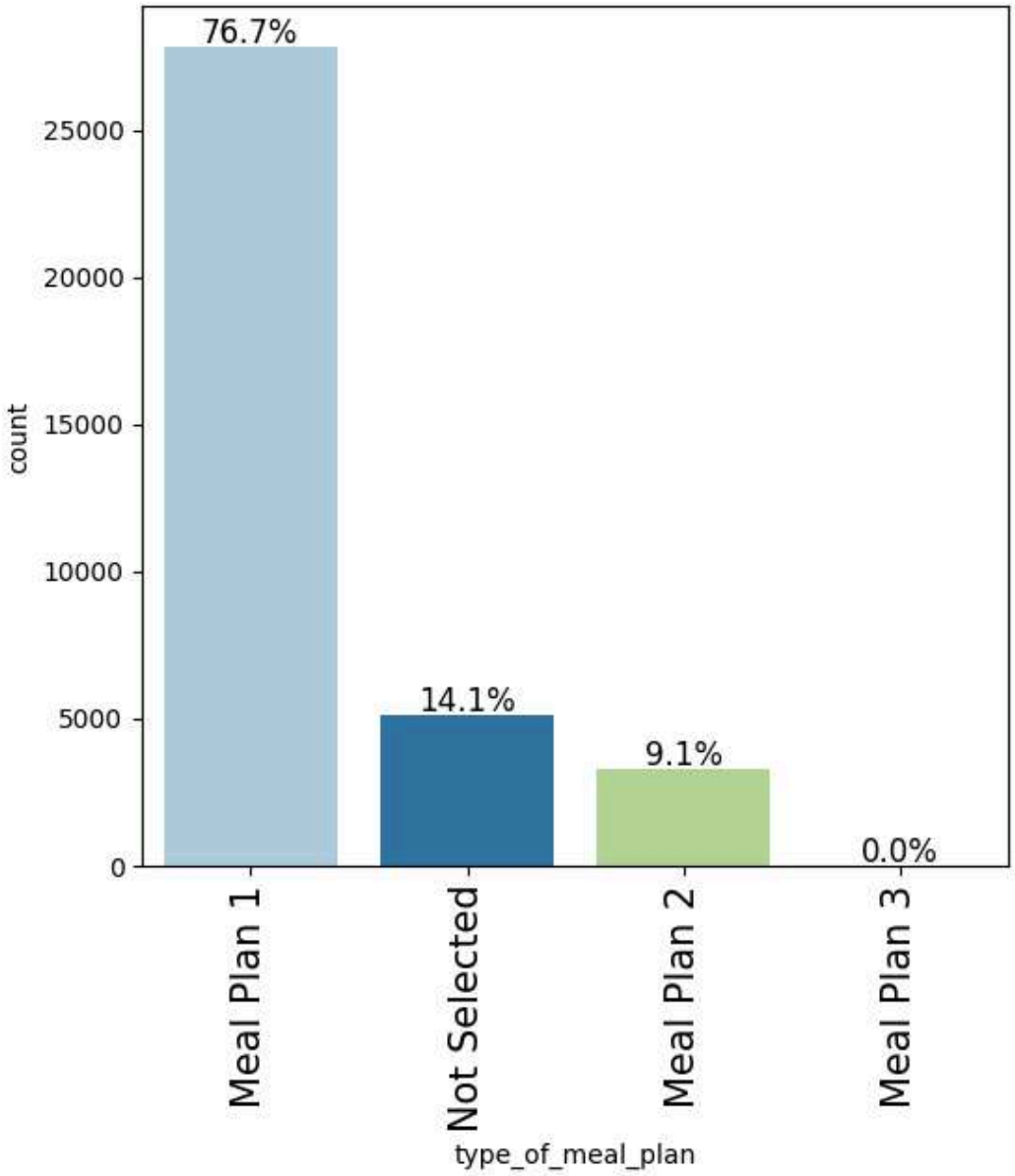


Figure 11: The Analysis on type_of_meal_plan

Observations on type_of_meal_plan

- 76.7% of guests opt for Meal Plan 1, indicating it is the default or most attractive option.
- 14.1% chose “Not Selected”, meaning they booked rooms without a meal plan.
- Only 9.1% selected Meal Plan 2, and 0% chose Meal Plan 3, making it essentially unused.

Analysis on room_type_reserved

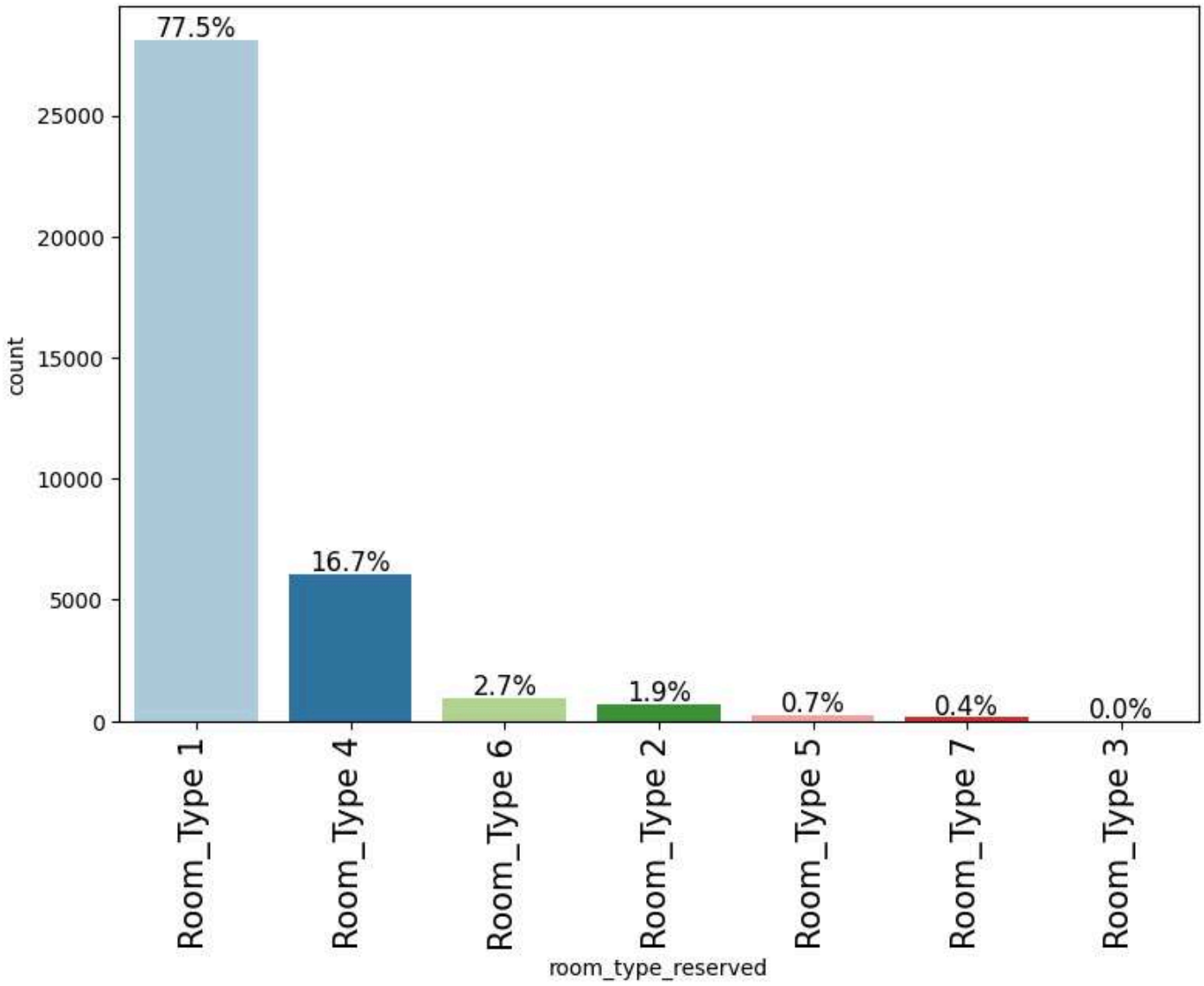


Figure 12: The Analysis on room_type_reserved

Observations on room_type_reserved

- 77.5% of bookings are for Room_Type 1, making it the default or most accessible option. could be Budget-friendly.
- 16.7% of bookings fall under Room_Type 4, making it the second-most popular.

Analysis on market_segment_type

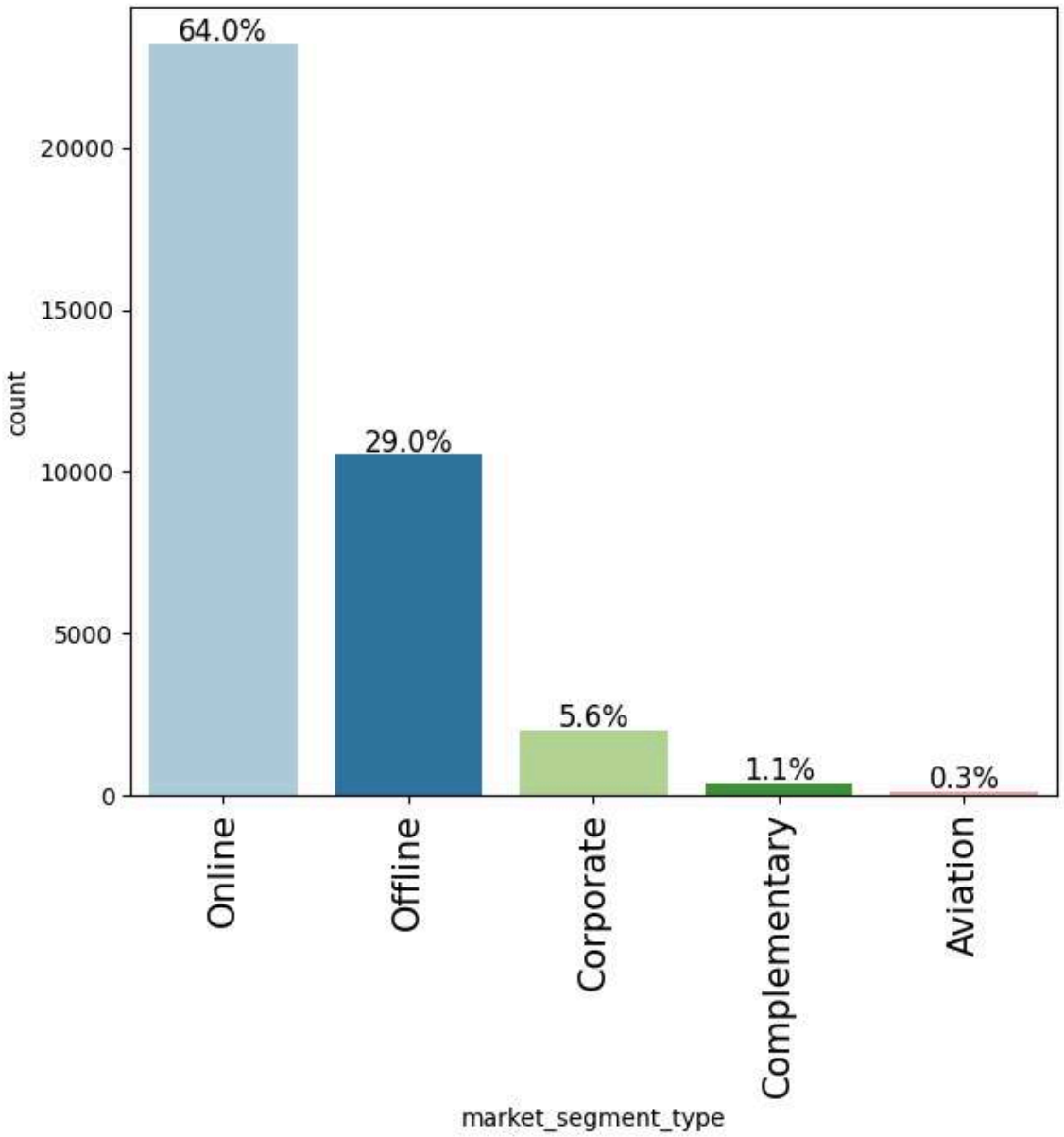


Figure 13: The Analysis on market_segment_type

Observations on market_segment_type

- 64.0% of all bookings are made through online platforms.Indicates a strong digital presence and consumer preference for convenience
- 29.0% of bookings come through offline channels.
- Business travel accounts for a small but notable share of 5.6%.

Analysis on booking_status

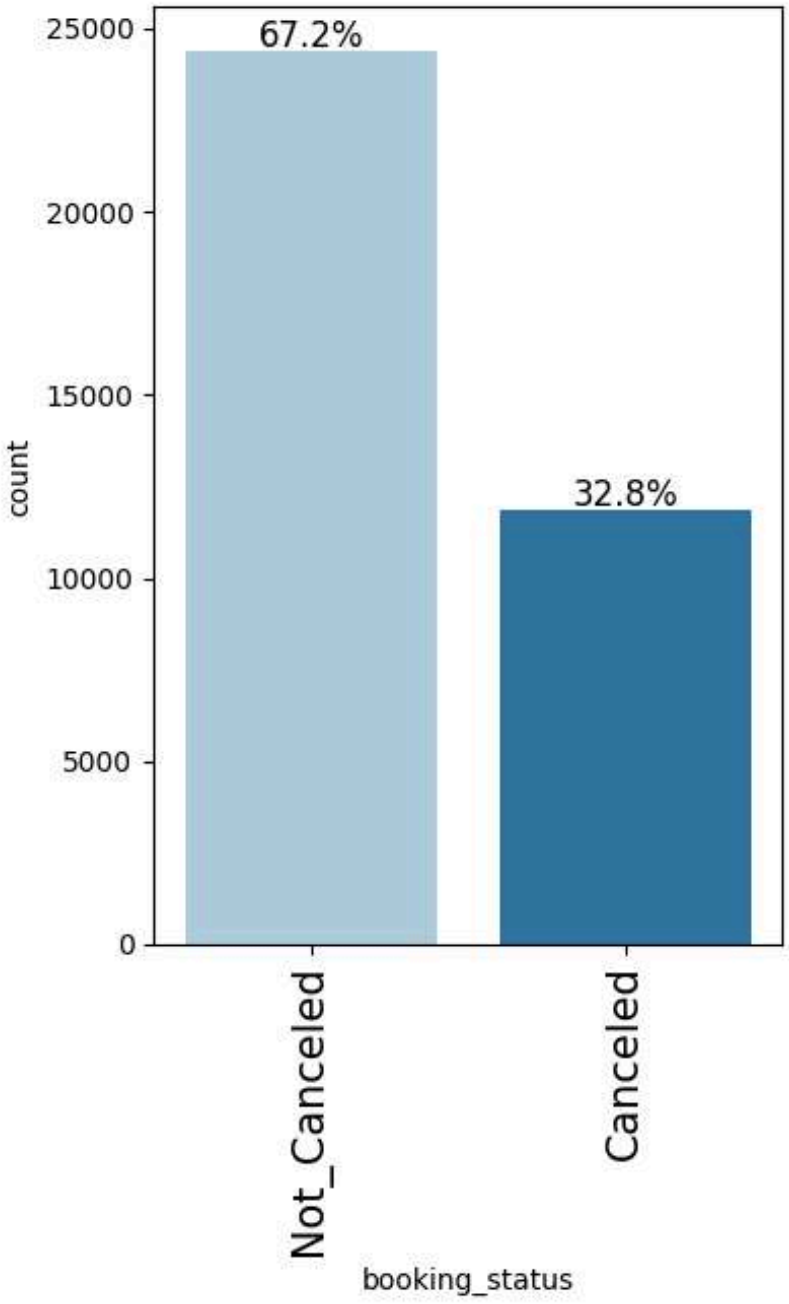


Figure 14: The Analysis on booking_status

Observations on booking_status

- Not Canceled – 67.2%, Most guests complete their bookings. Indicates good operational reliability
- Canceled – 32.8%, Nearly 1 in 3 bookings are canceled.

Bivariate Analysis

Analysis between room_type_reserved and avg_price_per_room

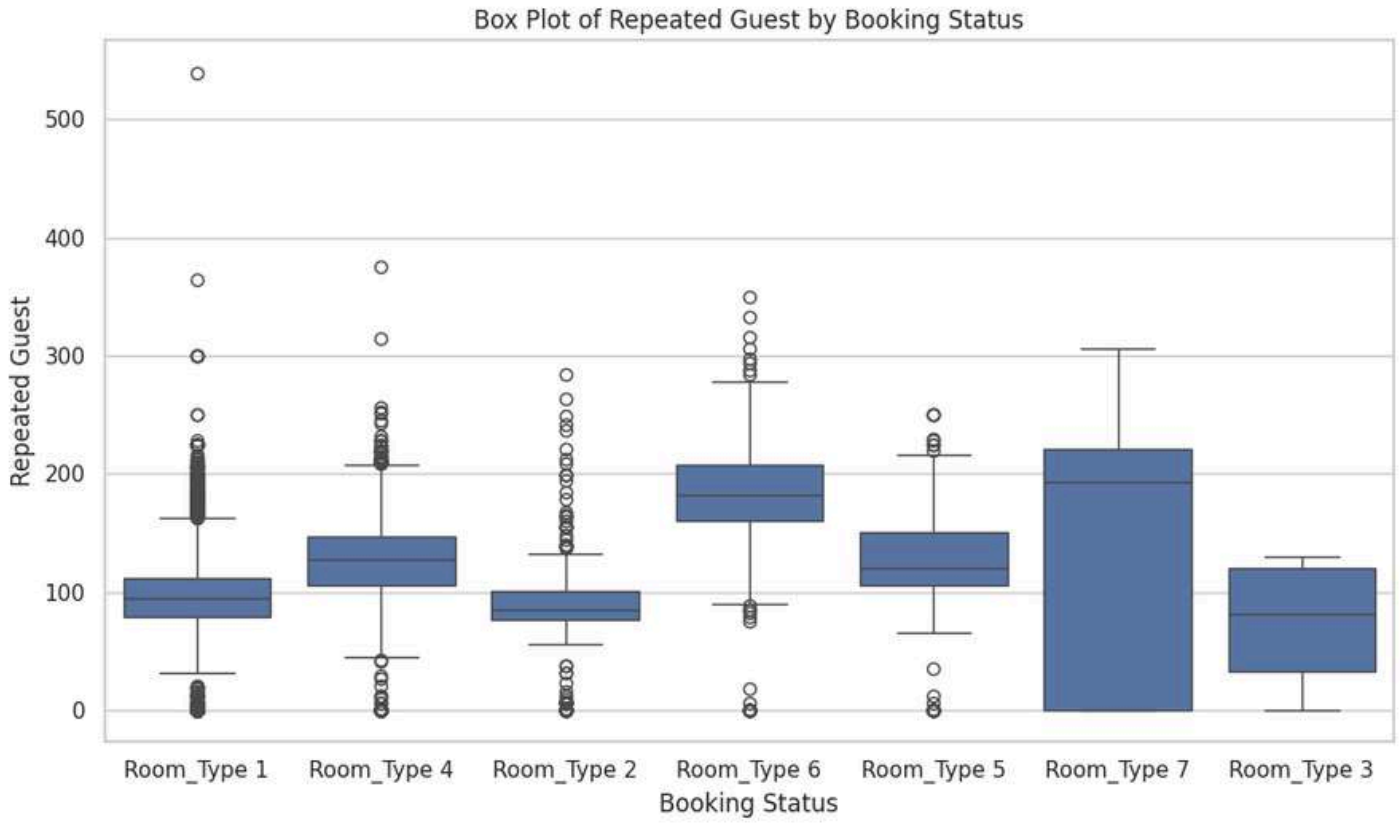


Figure 15: The Analysis between room_type_reserved and avg_price_per_room

Observations

- Room_Type 7 shows the highest median price and widest spread (interquartile range and whiskers).
- Room_Type 6 comes next in terms of median price and has a relatively consistent spread.
- Room_Type 1 and Room_Type 2 have lower medians and denser clusters of data points, indicating they may be standard or economy rooms.

Analysis between lead_time and avg_price_per_room

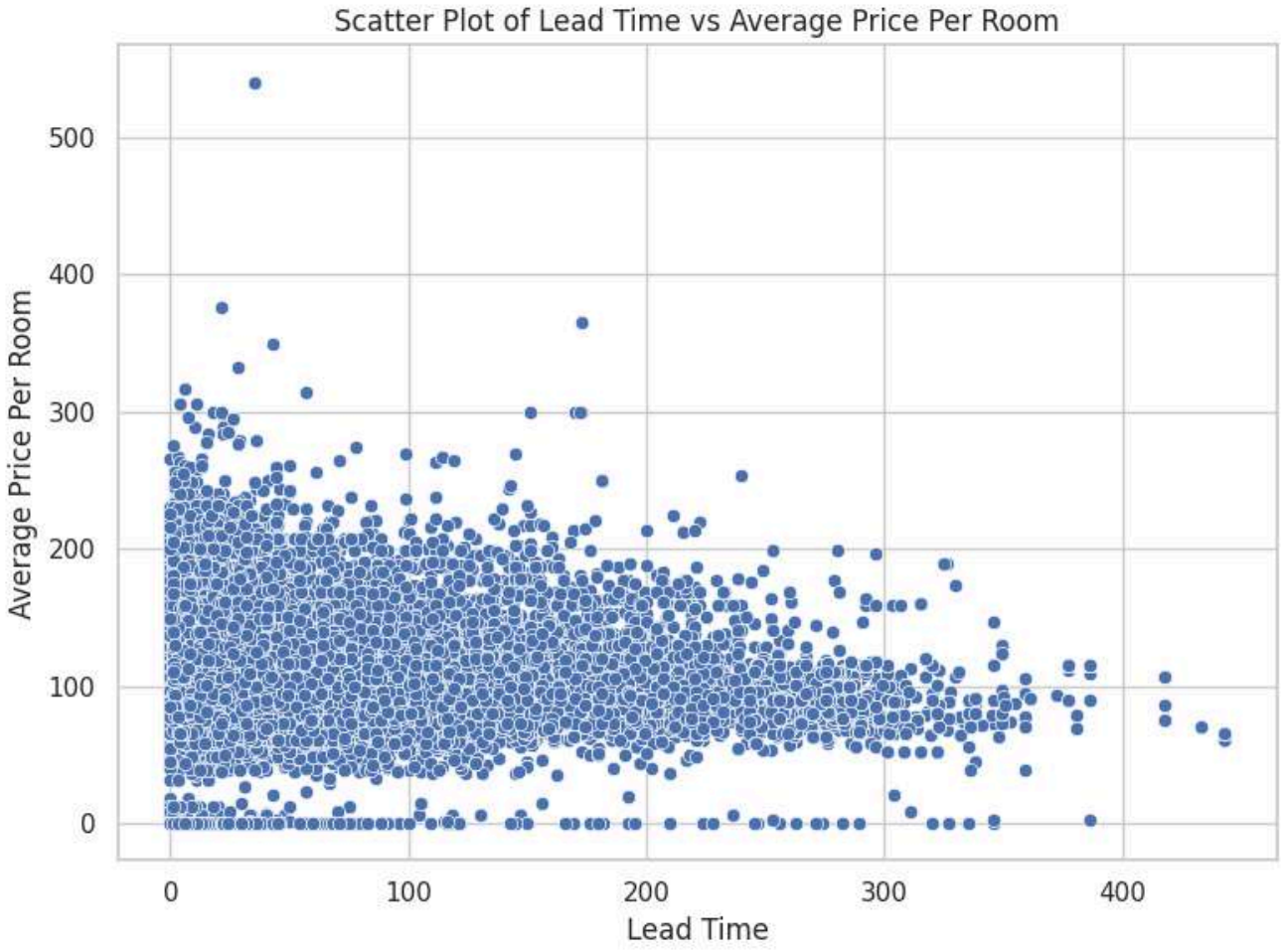


Figure 16: The Analysis between lead_time and avg_price_per_room

Observations

- There's a slight downward trend: as lead time increases, the average price per room tends to decrease.This suggests that early bookings (high lead time) may come with lower prices, possibly due to early bird discounts.
- A large number of bookings occur with low lead time (0–100 days), indicating many customers book rooms closer to their stay dates. Within this range, room prices vary widely—from very low to over 500 units.

Analysis between booking_status and market_segment_type

Figure 17: The Analysis between booking_status and market_segment_type

Observations

- The Online market segment is the largest contributor to both Canceled and Not_Canceled bookings. However, it accounts for a larger proportion of Canceled bookings, suggesting that online customers may have a higher cancellation tendency.
- The Offline segment makes up a larger share of Not_Canceled bookings compared to Canceled.
- Both segments are more prominent in Not_Canceled bookings, particularly the Corporate segment.

Analysis between lead_time and booking_status

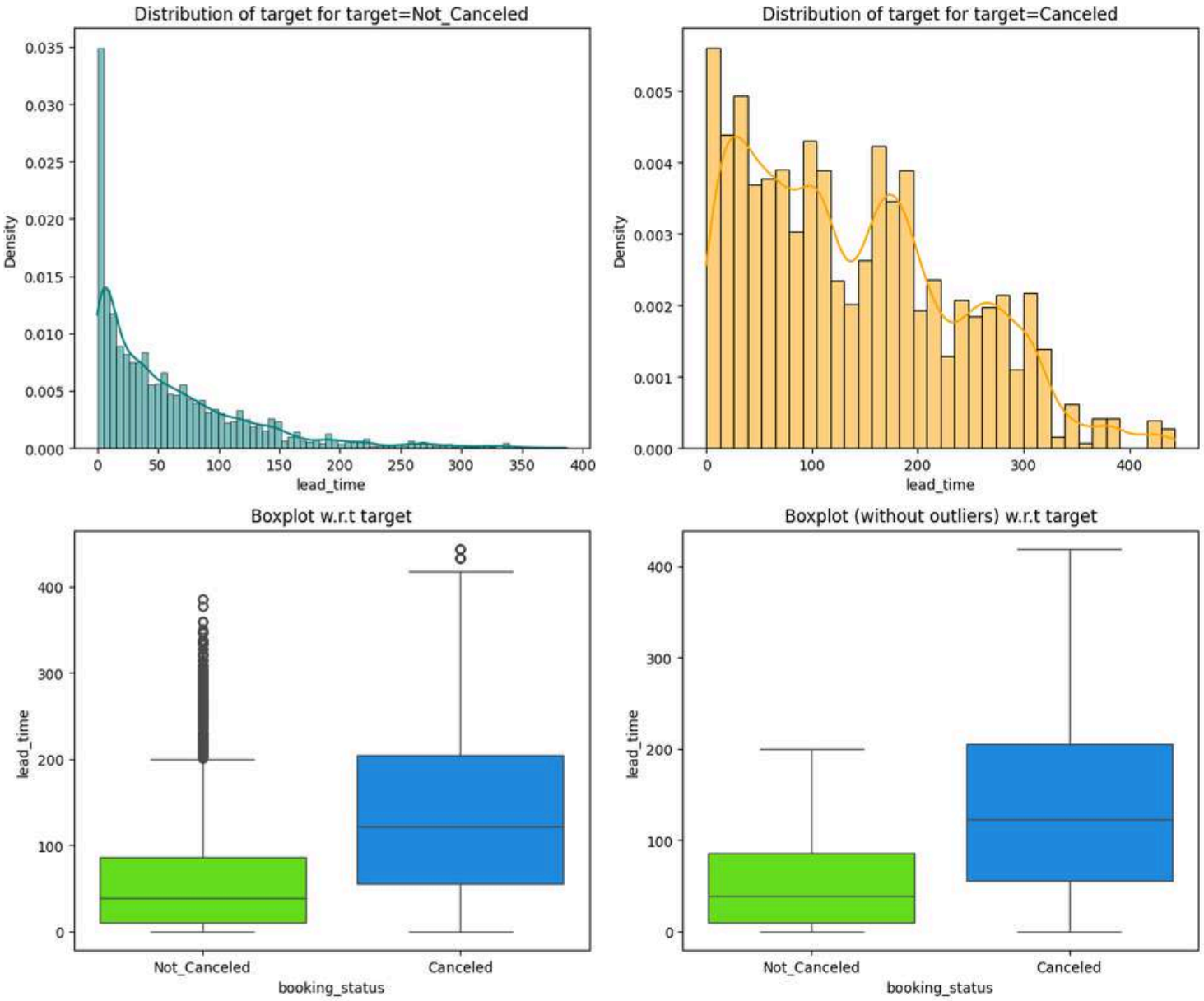


Figure 18: The Analysis between lead_time and booking_status

Observations

- Canceled bookings tend to have much longer lead times compared to not canceled ones.
- Short lead time bookings are more likely to be fulfilled, indicating higher commitment.
- Lead time is a strong predictor of cancellation risk and should be used in forecasting models or risk strategies.

Analysis between type_of_meal_plan and booking_status

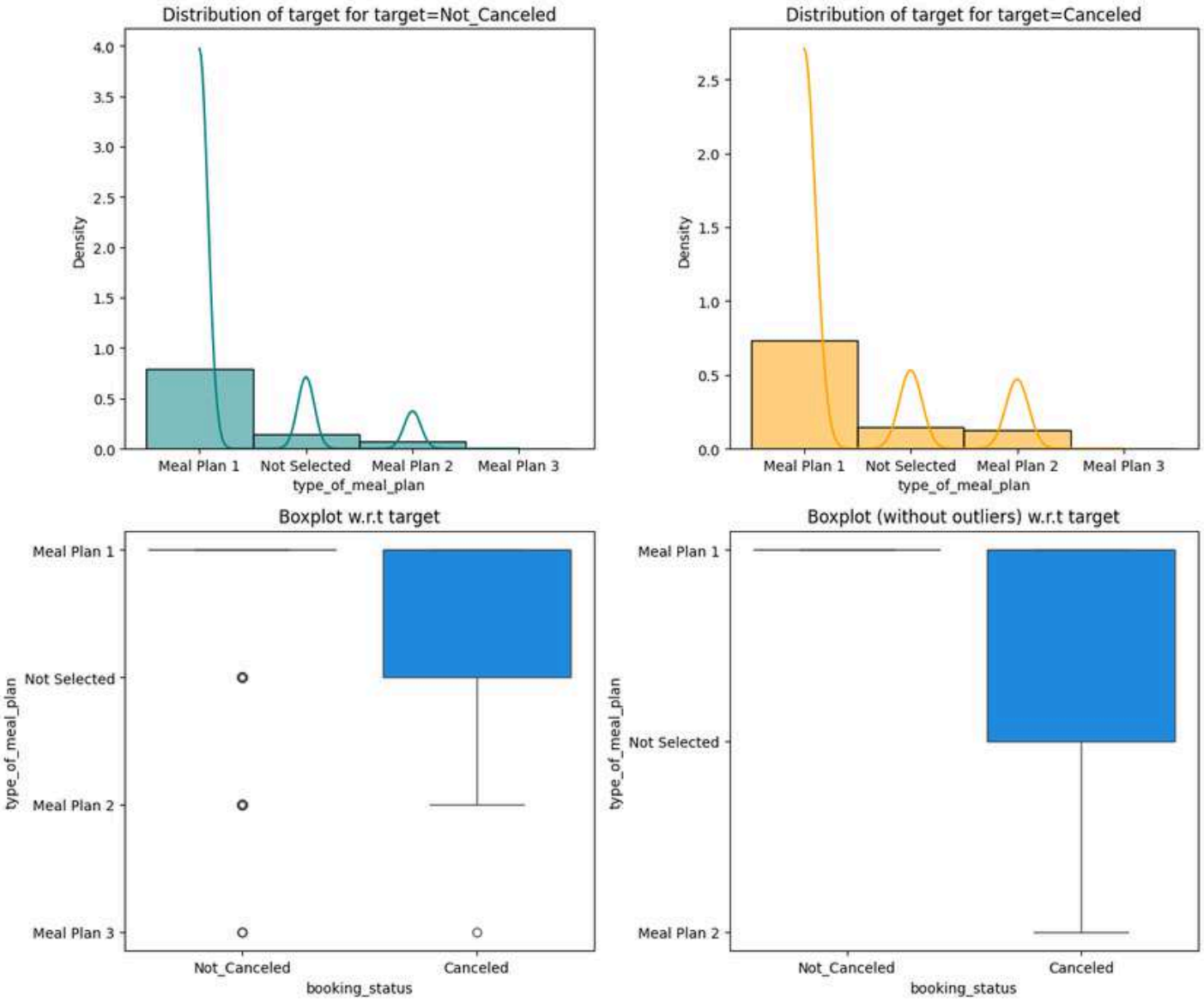


Figure 19: The Analysis between type_of_meal_plan and booking_status

Observations

- Meal Plan 1 is dominant across both canceled and not canceled bookings, indicating it's the most commonly chosen plan regardless of booking outcome.
- Alternative meal plans (like Meal Plan 2 and 'Not Selected') are more frequently associated with cancellations, suggesting less commitment or satisfaction.

Correlation matrix

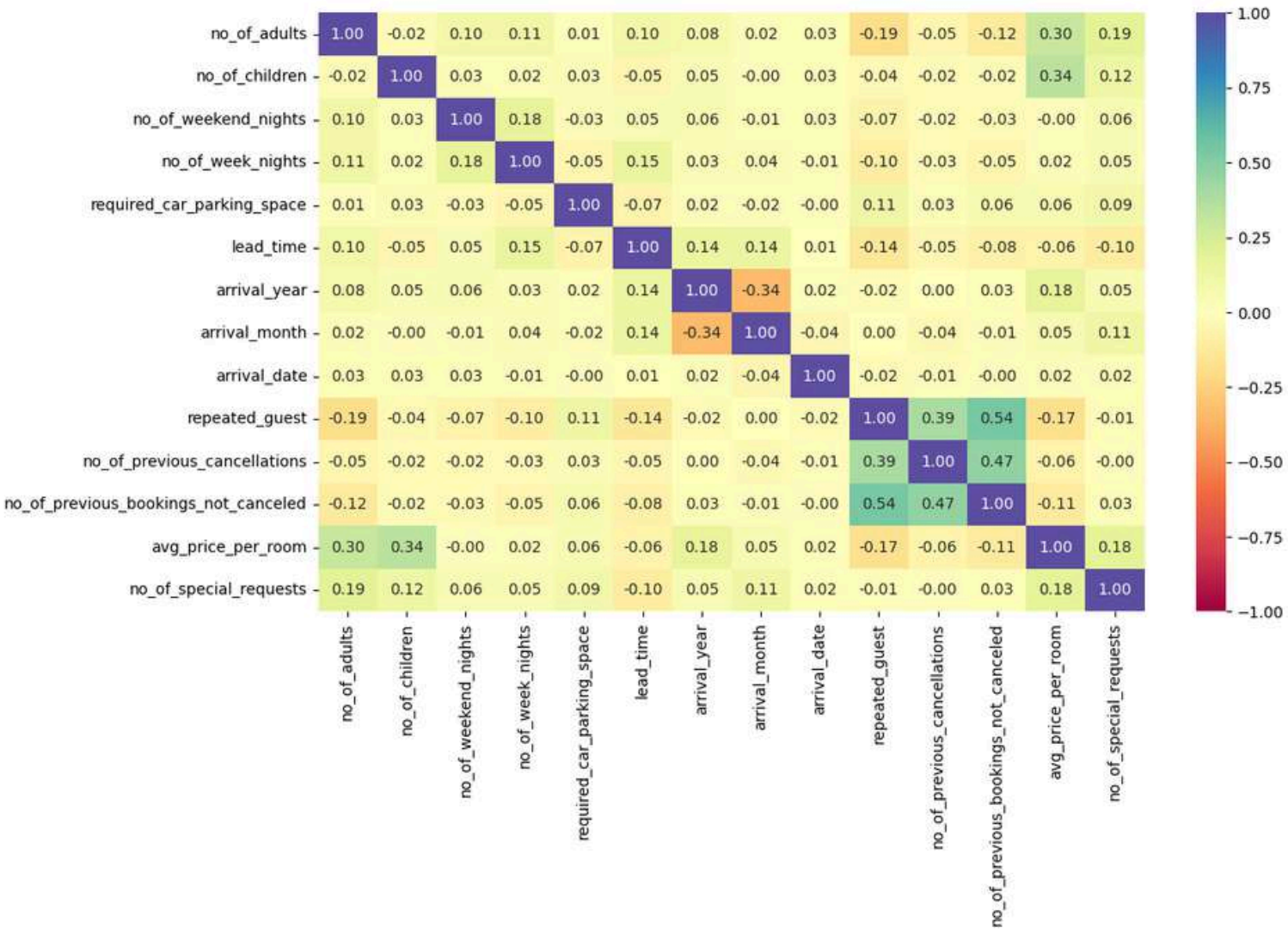


Figure 20: Correlation matrix

Observations

- repeated_guest correlates with both no_of_previous_bookings_not_canceled (0.54) and no_of_previous_cancellations (0.39).
- avg_price_per_room correlates with no_of_children (0.34) and no_of_adults (0.30), suggesting larger groups pay more.
- lead_time shows slight correlation with arrival_year and arrival_month (both 0.14), hinting at seasonal booking trends.

Data preprocessing

Outlier Detection

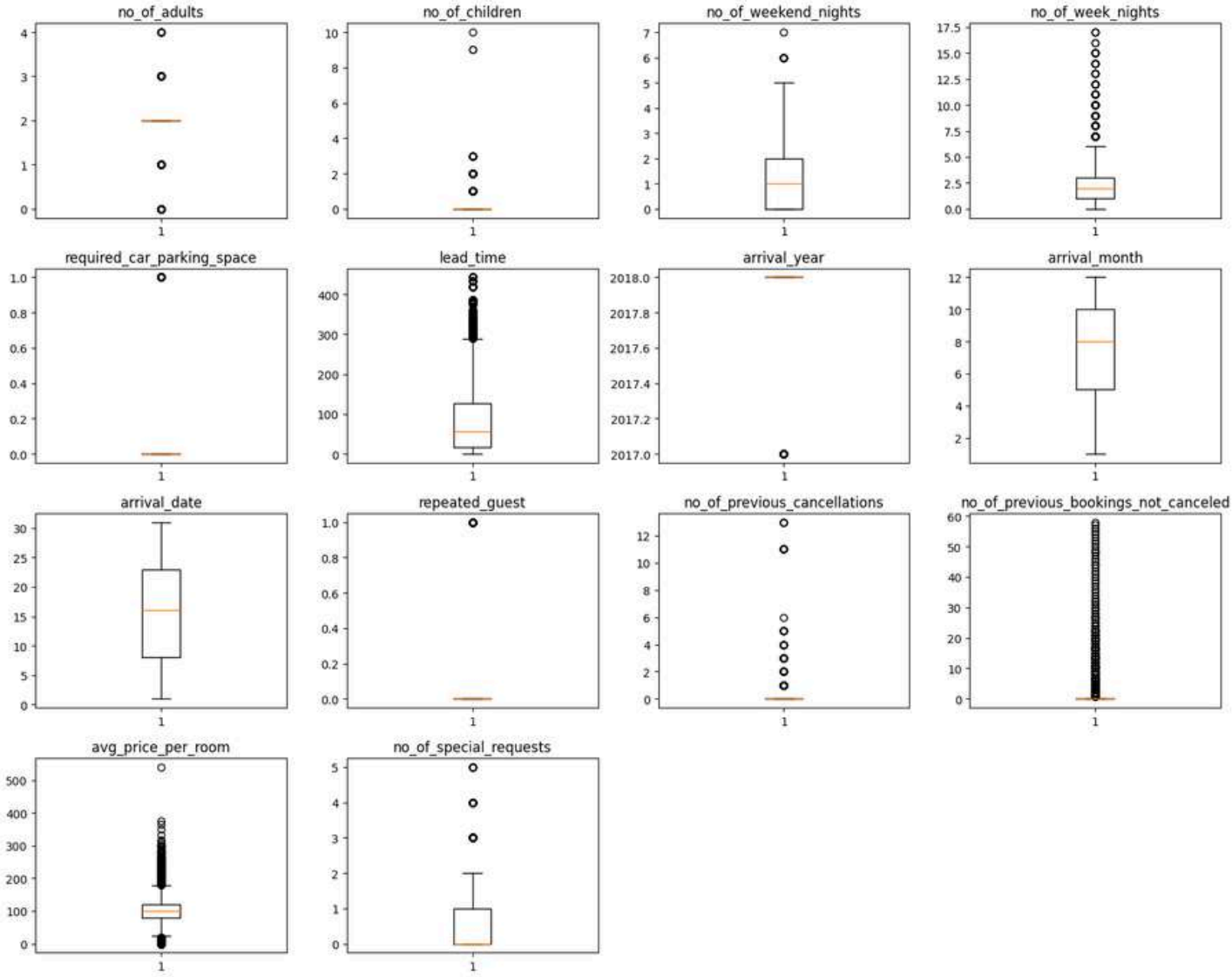


Figure 21: Outlier Detection

Observations

- Many variables (e.g., lead_time, avg_price_per_room, no_of_children, no_of_previous_cancellations) show significant outliers, indicating skewed distributions.
- Variables like required_car_parking_space and repeated_guest are binary (mostly 0), and arrival_year has almost all data from a single year (2018).
- Most bookings involve 2 adults, 0 children, 0 special requests, and no previous cancellations, suggesting 41 common booking patterns across users.

Checking for duplicate values

- There are no duplicate values in the data.

Checking for missing values

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

Table 6: Data on missing values

- There are no missing values in the data.

Outlier treatment

- We will not treat them as they are proper values.

Feature engineering

- The **pandas.get_dummies()** function in Python is used to perform one-hot encoding on categorical data. It converts categorical columns into multiple binary columns (0 or 1), which makes the data suitable for machine learning algorithms.

Adding a column Total nights

- Adding the column total nights by combining no_of_weekend_nights and no_of_week_nights

```
0        3
1        5
2        3
3        2
4        2
       ..
36270    8
36271    4
36272    8
36273    3
36274    3
Name: total_nights, Length: 36275, dtype: int64
```

Table 7: Adding a column Total nights

Data preparation for modeling

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	\
0	2	0	1	2	
1	2	0	2	3	
2	1	0	2	1	
3	2	0	0	2	
4	2	0	1	1	

	type_of_meal_plan	required_car_parking_space	room_type_reserved	lead_time	\
0	Meal Plan 1	0	Room_Type 1	224	
1	Not Selected	0	Room_Type 1	5	
2	Meal Plan 1	0	Room_Type 1	1	
3	Meal Plan 1	0	Room_Type 1	211	
4	Not Selected	0	Room_Type 1	48	

	arrival_year	arrival_month	arrival_date	market_segment_type
0	2017	10	2	Offline
1	2018	11	6	Online
2	2018	2	28	Online
3	2018	5	20	Online
4	2018	4	11	Online

	repeated_guest	no_of_previous_cancellations	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	no_of_previous_bookings_not_canceled	avg_price_per_room	\
0	0	65.00000	
1	0	106.68000	
2	0	60.00000	
3	0	100.00000	
4	0	94.50000	

	no_of_special_requests
0	0
1	1
2	0
3	0
4	0

0	0
1	0
2	1
3	1
4	1

Name: booking_status, dtype: int64

Table 8: Data preparation for modeling

- We want to predict the booking_status.
- Before we proceed to build a model, we'll have to encode categorical features
- We'll split the data into train and test to be able to evaluate the model that we build on the train data
- We will build a logistic Regression model using the train data and then check it's performance

Training and test set information

```
Shape of Training set :  (25392, 28)
Shape of test set :  (10883, 28)
Percentage of classes in training set:
booking_status
0   0.67399
1   0.32601
Name: proportion, dtype: float64
Percentage of classes in test set:
booking_status
0   0.66857
1   0.33143
Name: proportion, dtype: float64
```

Table 9: Training and test set

- Training set size: 25,392 rows; Test set size: 10,883 rows — both with 28 features.
- Class balance: ~67% not canceled (0), ~33% canceled (1) in both sets.
- Stratified split: Class proportions are consistent, ensuring fair model evaluation.

Model building - Logistic Regression

- Using **sm.add_constant(x_train)** adds a constant column (usually named "const") filled with 1s to the training dataset.
- Using **sm.add_constant(x_test)** does the same for the test dataset.

Using **sm.Logit(y_train, X_train_sm),** creates an Logistic Regression model object

- **y_train**: The target variable (dependent variable — that we want to predict).
- **x_train**: The feature variables (independent variables), including a constant (intercept) added earlier using sm.add_constant().

Using **fits** (trains) the model using the training data.

Using **result.predict(X_test_sm)** to Predict probabilities and labels

Logit Regression Results

Dep. Variable:	booking_status	No. Observations:	25392
Model:	Logit	Df Residuals:	25363
Method:	MLE	Df Model:	28
Date:	Sat, 31 May 2025	Pseudo R-squ.:	0.3293
Time:	18:41:57	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	0.0583	1.72e+05	3.39e-07	1.000	-3.37e+05	3.37e+05
no_of_week_nights	-0.0086	1.72e+05	-5.02e-08	1.000	-3.37e+05	3.37e+05
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
total_nights	0.0484	1.72e+05	2.82e-07	1.000	-3.37e+05	3.37e+05
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	9930.887
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

Table 10: Logistic Regression

- Some variables (e.g., multiple room_type_reserved_* and type_of_meal_plan_*) may exhibit multicollinearity due to one-hot encoding. Consider using Variance Inflation Factor (VIF) to identify and remove highly collinear features.

- Variables with p-values < 0.05 are statistically significant. These include:
  - no_of_adults, no_of_children, lead_time, repeated_guest, no_of_special_requests, avg_price_per_room, among others.
  - These should be retained as they have a meaningful relationship with booking status.
- Pseudo R-squared: 0.2393 suggests moderate model fit for classification tasks.

Using Variance Inflation Factor (VIF) to identify and remove highly collinear features.

```
Series before feature selection:

const                                 39468156.70600
no_of_adults                                1.34815
no_of_children                              1.97823
no_of_weekend_nights                            inf
no_of_week_nights                               inf
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
total_nights                                    inf
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
```

Table 11: Checking for Multicollinearity

- Major columns like market segment type Complementary, market_segment_type_Corporate, market_segment_type_Offline and market_segment_type_Online have high collinearity.

Removing the columns with high collinearity (VIF > 5)

	Feature	VIF
0	const	35303015.73364
1	no_of_adults	1.20612
2	no_of_children	1.16437
3	no_of_weekend_nights	1.05475
4	no_of_week_nights	1.07003
5	required_car_parking_space	1.03312
6	lead_time	1.39068
7	arrival_year	1.27739
8	arrival_month	1.24021
9	arrival_date	1.00484
10	repeated_guest	1.55629
11	no_of_previous_cancellations	1.33318
12	no_of_previous_bookings_not_canceled	1.59590
13	avg_price_per_room	1.42986
14	no_of_special_requests	1.21078
15	booking_status	1.42300

Table 12: Removing the columns with high collinearity

- Major columns like market segment type Complementary, market_segment_type_Corporate, market_segment_type_Offline and market_segment_type_Online with high collinearity has been removed.

Decision Tree Classifier

- Using **DecisionTreeClassifier(random_state=1)**

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

Table 13: Decision Tree Classifier

Confusion matrix for training set

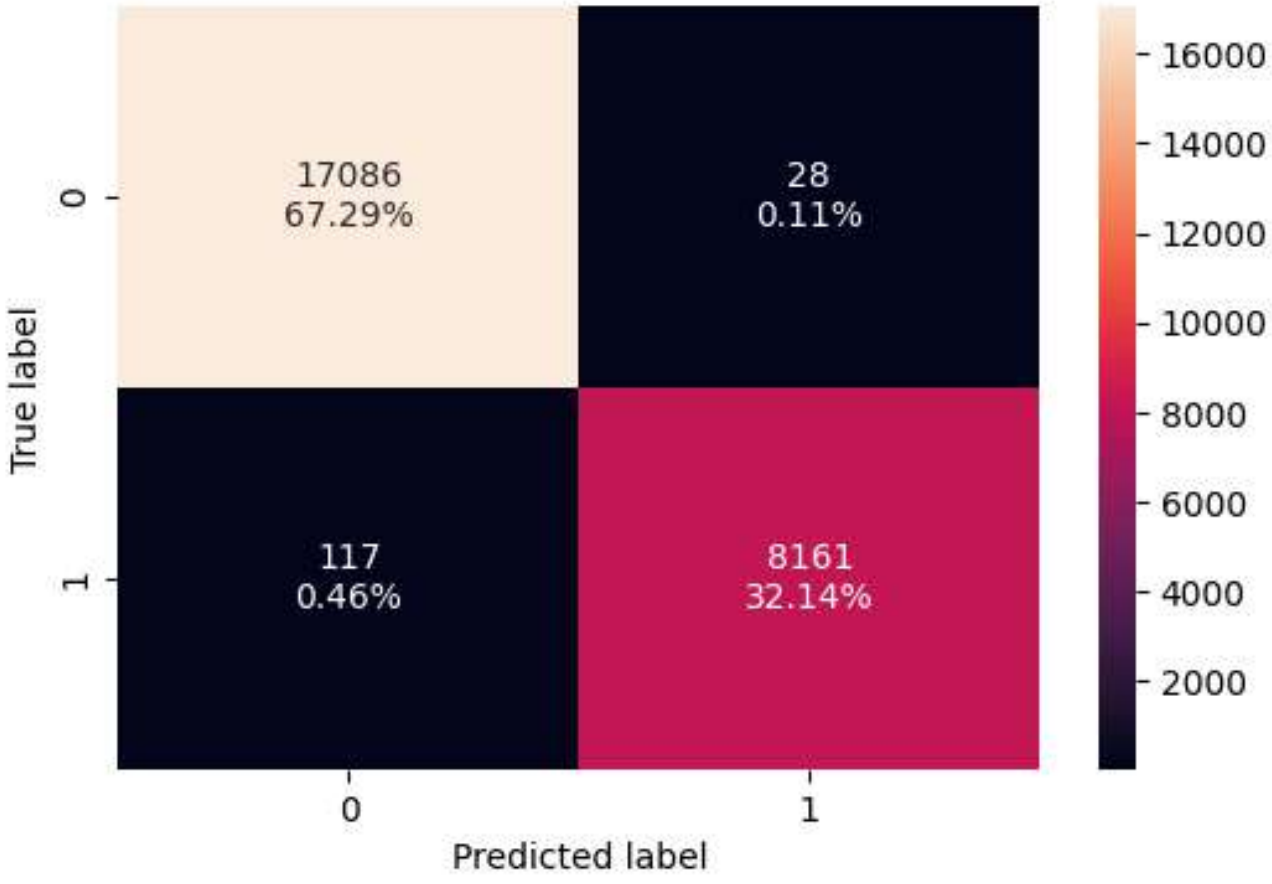


Figure 22: Confusion matrix for training set

High Overall Accuracy

- The model predicted correctly in 25,247 out of 25,392 cases.
- Majority class (not canceled) and minority class (canceled) are both well identified.

Low False Positives & False Negatives

- False Positives (Type I error): 28 cases where a non-cancellation was wrongly predicted as a cancellation.
- False Negatives (Type II error): 117 cases where actual cancellations were missed.
- These are low, indicating strong precision and recall.

Well-balanced Model Performance

- The model captured both classes with good balance:
  - True Negatives (0,0): 67.29%
  - True Positives (1,1): 32.14%
- Suggests good performance in a class-imbalanced setup.

Accuracy, Recall, Precision, F1 for training data

	Accuracy	Recall	Precision	F1
0	0.99429	0.98587	0.99658	0.99119

Table 14: Accuracy, Recall, Precision, F1 for training data

High Overall Accuracy (99.4%)

- The model correctly predicted most booking statuses, suggesting strong general performance.

Strong Recall (98.6%)

- Almost all actual positive cases (cancellations) were correctly identified, minimizing missed cancellations..

Very High Precision (99.66%)

- Of the bookings predicted as cancellations, nearly all were correct.
- Indicates the model makes very few false positive predictions.

Very High Precision (99.66%)

- The F1 score shows a perfect balance between precision and recall.
- Reinforces that the model is reliable for both identifying and avoiding false cancellations.

Confusion matrix for test set

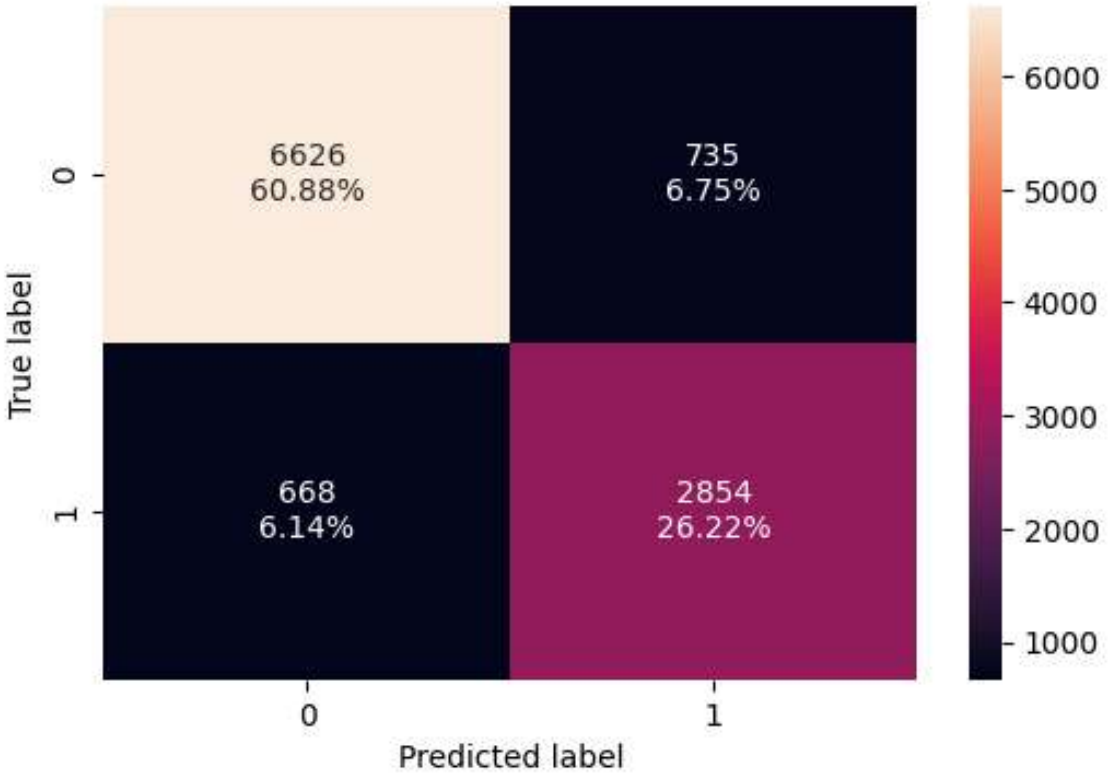


Figure 23: Confusion matrix for test set

Accuracy, Recall, Precision, F1 for test data

	Accuracy	Recall	Precision	F1
0	0.87108	0.81034	0.79521	0.80270

Table 15: Accuracy, Recall, Precision, F1 for test data

Moderate Accuracy (87.1%)

- The model performs reasonably well overall but is notably weaker than the previous model with ~99% accuracy.

Lower Recall (81.0%)

- It misses nearly 19% of actual cancellations, which could lead to operational inefficiencies.

Precision at 79.52%

- About 20% of predicted cancellations were incorrect (false positives).
- This can lead to unnecessary reservation holding or overbooking concerns.

F1 Score at 80.27%

- The F1 score reflects a balanced but not optimal trade-off between precision and recall.
- Indicates the model performs adequately, but not as reliably as the better-performing model.

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

- Creating a model with DecisionTreeClassifier with the class weight balanced and fitting the X_train and y_train in the model.

Creating confusion matrix with class weights for training set

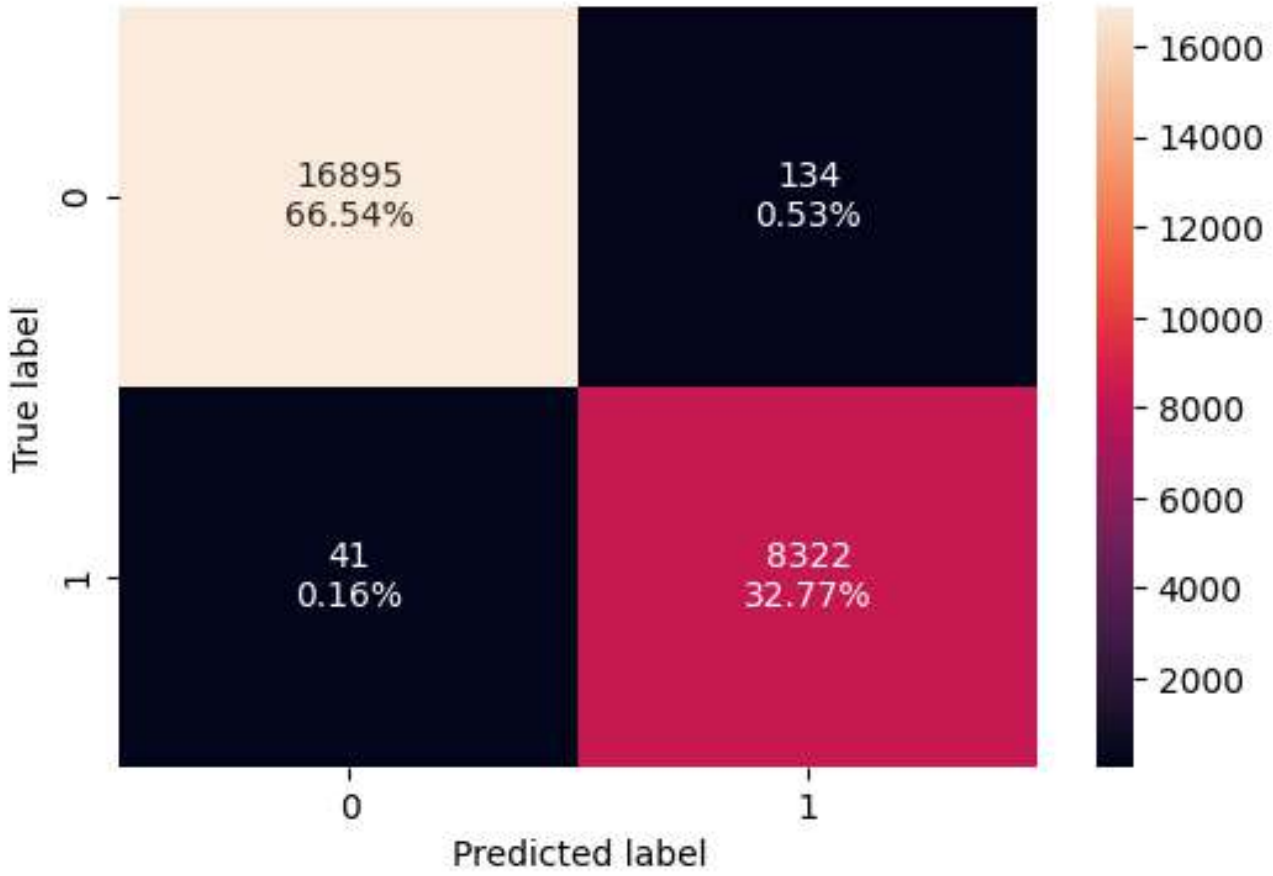


Figure 24: confusion matrix with class weights for training set

High Overall Accuracy

- True Positives (TP): 8,322
- True Negatives (TN): 16,895
- Together, these correct predictions make up 99.31% of the total predictions, indicating a highly accurate model.

Low False Negative Rate

- Only 41 actual positives were misclassified as negatives (False Negatives).
- This implies the model is very effective at detecting positives, which is crucial when positive cases are important to catch (e.g., cancellations or frauds).

- Class 0 (non-event) occurs slightly more frequently than Class 1 (event), as seen by the higher number of true negatives (16,895) versus true positives (8,322).
- However, the model still performs well across both classes.

Checking model performance classification with train set

	Accuracy	Recall	Precision	F1
0	0.99311	0.99510	0.98415	0.98960

Table 16: model performance classification with train set

Excellent Overall Accuracy

- The model achieved an accuracy of 99.31%, which means it correctly predicted outcomes in over 99% of the cases.

Very High Recall (99.51%)

- The recall score indicates the model correctly identified 99.51% of actual positive instances.
- This is especially valuable when minimizing false negatives is critical (e.g., detecting cancellations or fraud).

Strong Precision (98.42%)

- Precision shows that 98.42% of positive predictions were correct, suggesting a low false positive rate.
- This is important when acting on false positives could be costly or disruptive.

Balanced F1 Score (98.96%)

- The F1 score, which balances precision and recall, is very high at 98.96%, confirming the model is both accurate and reliable across positive predictions.

Creating confusion matrix with class weights for test set

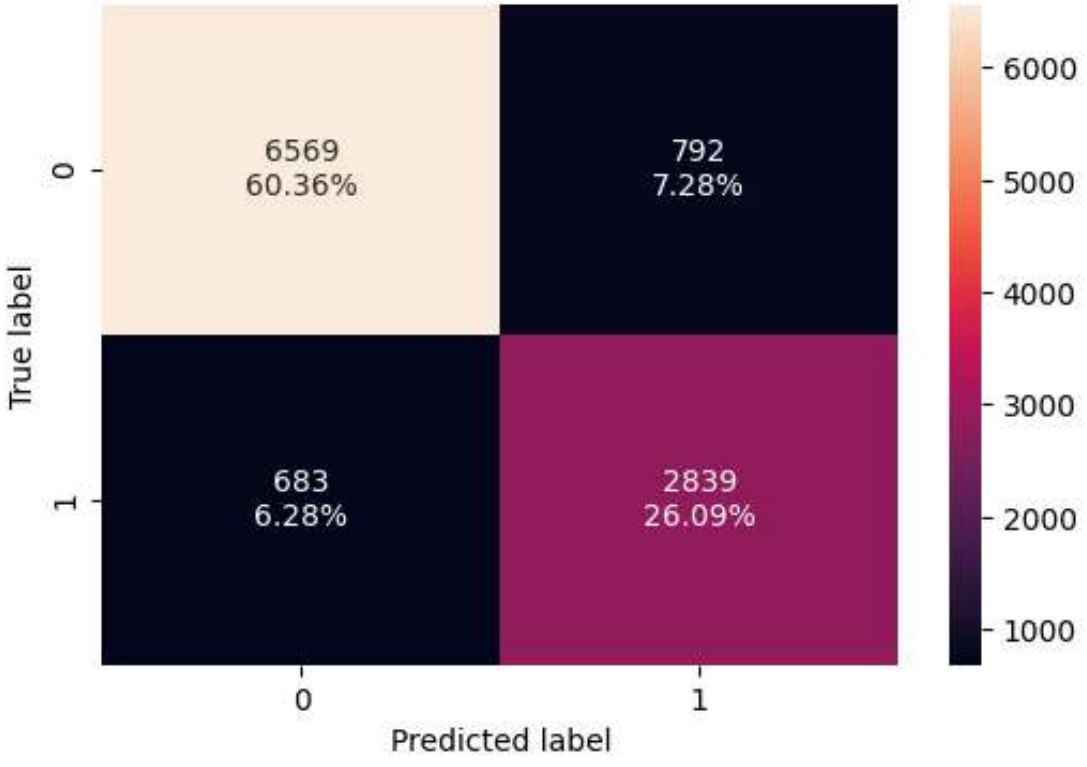


Figure 25: Confusion matrix with class weights for test set

- The model correctly classified 6,569 true negatives and 2,839 true positives, reflecting decent separation between classes.
- Misclassifications include 792 false positives and 683 false negatives, which are moderately high and may need attention depending on business cost.
- The true positive rate (recall for class 1) is approximately 80.6%, suggesting room for improvement in capturing all actual positives.
- About 13.56% of total predictions (false positives + false negatives) were incorrect, impacting overall model reliability.

Checking model performance classification with test set

	Accuracy	Recall	Precision	F1
0	0.86447	0.80608	0.78188	0.79379

Table 17: Model performance classification with test set

Accuracy (86.4%):

- The model performs well overall, correctly predicting the outcome in most instances.

Recall (80.6%):

- It captures approximately 81% of all actual positive cases, indicating good sensitivity.

Precision (78.2%):

- About 78% of the predicted positive cases are correct, while the rest are false positives.

F1-Score (79.4%):

- The harmonic mean of precision and recall shows balanced performance, though slightly lower precision suggests some trade-off.

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=np.int64(2),
                       max_leaf_nodes=50, min_samples_split=10, random_state=1)
```

Table 18: Decision Tree (Pre-pruning)

Decision Tree Classifier –

- The parameter class_weight='balanced' indicates the model accounts for class imbalance, assigning higher weights to the minority class (likely cancellations), which improves fairness in classification.

- The tree depth is limited with max_depth=np.int64(2), which prevents overfitting and ensures the model remains interpretable. However, this shallow depth might underfit if the data has complex patterns.

- Use of max_leaf_nodes=50 and min_samples_split=10 indicates pre-pruning strategies to control the growth of the tree, reducing overfitting and improving generalization.

- With random_state=1, the model's output is deterministic across runs, which is good practice for consistent results and debugging.

Creating a decision tree

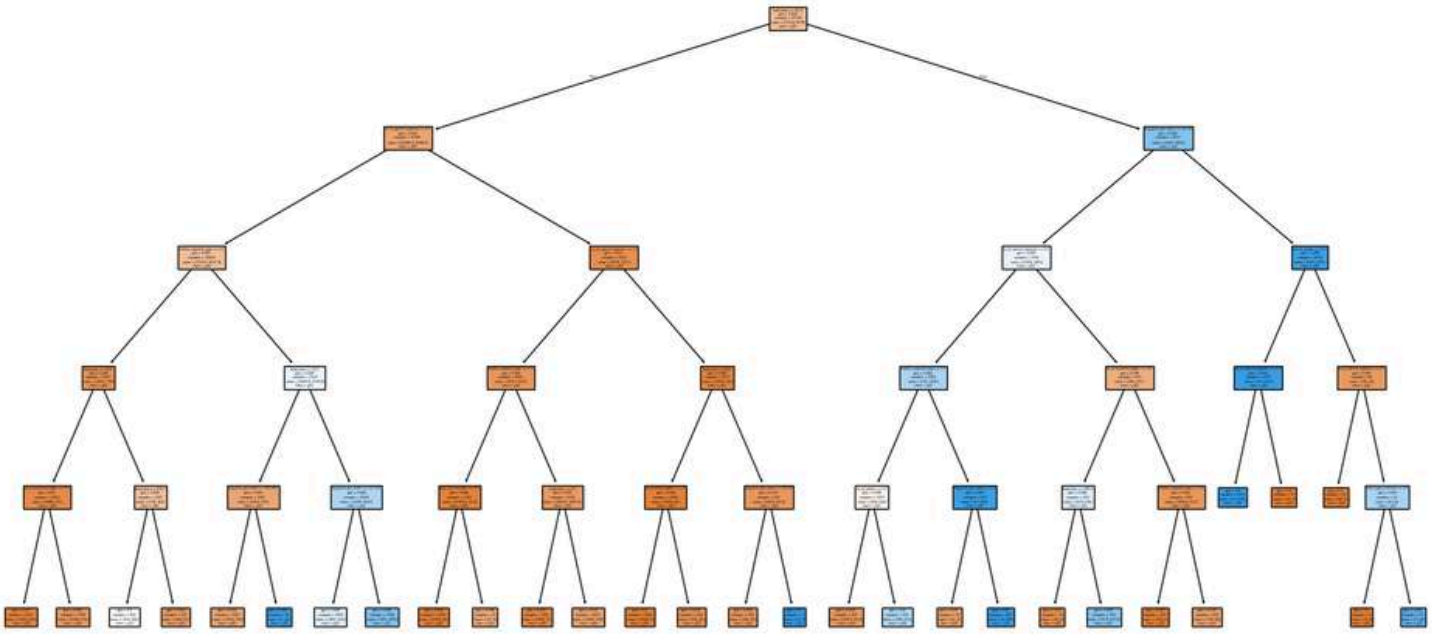


Figure 26: Decision Tree (Pre-pruning)

Insights from Decision Tree Visualization

Feature Importance Reflected by Top Nodes

- The root and upper-level nodes show the most influential features in predicting cancellations. For example, if lead_time or avg_price_per_room appears near the top, it indicates these have the highest predictive power.

Balanced Class Splits Visible

- The color gradient (orange for one class, blue for the other) and proportion of samples in each node suggest the tree is making a balanced attempt at classifying both booking outcomes, aligning with the class_weight='balanced' parameter.

Controlled Tree Depth Enhances Interpretability

- The tree depth is limited (likely ≤5), supporting pre-pruning. This keeps the model simple and interpretable without overfitting to noise in the training data.

Clear Decision Paths for Business Interpretation

- Each path from root to leaf can be translated into a set of if-else business rules. For instance:
- "If lead_time > 60 and no_of_special_requests = 0, then likely to cancel" – this can guide marketing or intervention strategies.

Test Accuracy: 0.84

Confusion_matrix for training dataset

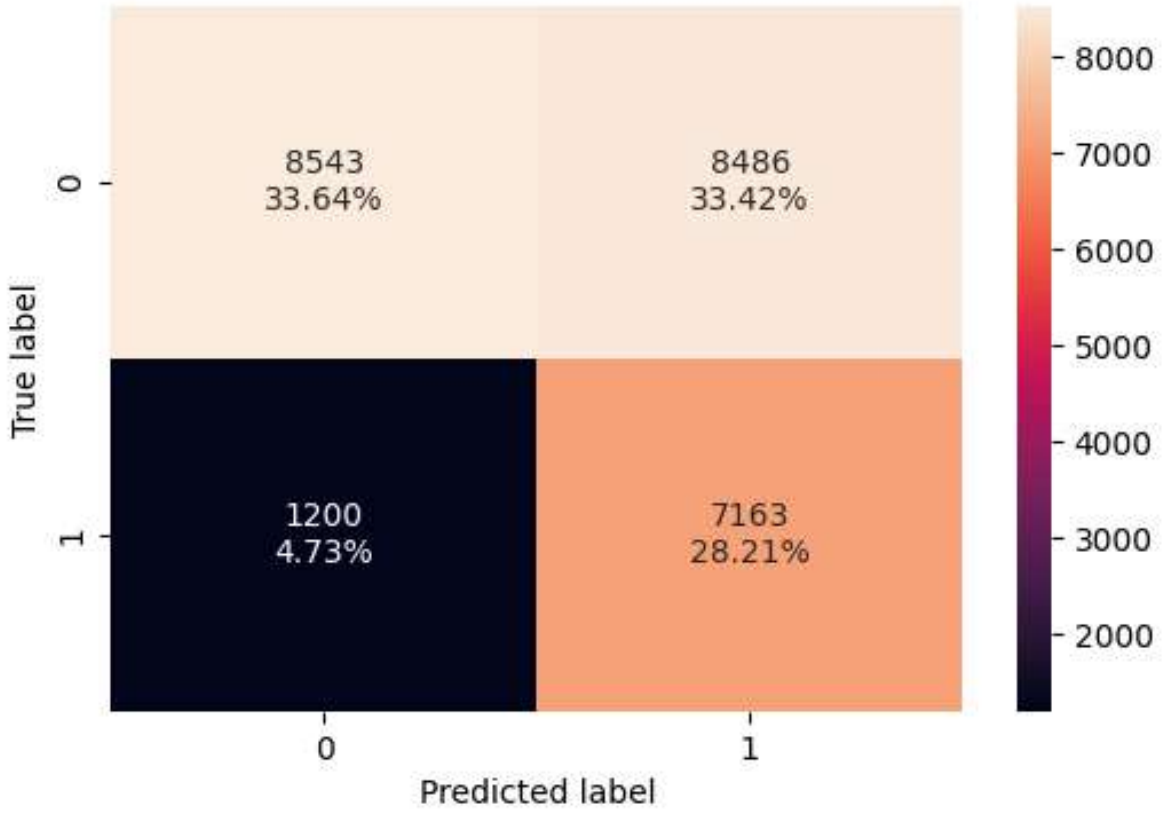


Figure 27: Confusion_matrix for training dataset

- **Balanced Classification:** The model predicts both classes (0 = no cancellation, 1 = cancellation) with almost equal frequency. The predicted positives (8486 + 7163) and negatives (8543 + 1200) are close, suggesting no major bias toward one class.

- **True Negative Rate is High:** The model correctly identified 8543 out of all actual class 0 samples, achieving a true negative rate of approximately 50.2% (8543 / (8543 + 8486)) — a reasonable result for non-cancellations.

- **Good True Positive Performance:** For class 1 (cancellations), the model correctly predicted 7163 instances, indicating a true positive rate (recall) of 85.6% (7163 / (7163 + 1200)). This shows the model is strong at catching cancellations.

- **Room for Reducing False Positives:** The model misclassified 8486 non-cancelled bookings as cancelled, contributing to potential overestimation of cancellations. This could negatively impact resource planning if relied upon operationally.

Model performance for training set

	Accuracy	Recall	Precision	F1
0	0.61854	0.85651	0.45773	0.59662

Table 19: Model performance for training set

- **High Recall (85.65%):** The model is very good at identifying actual non-cancellations, successfully catching most true negatives. This means it's less likely to miss genuine non-cancelled bookings.

- **Low Precision (45.77%):** Of all the bookings the model predicts as non-cancelled, less than half are actually correct. This indicates a high false positive rate—many cancelled bookings are wrongly predicted as non-cancelled.

- **Moderate F1-Score (59.66%):** The F1-score, which balances precision and recall, is moderate. This shows a trade-off: the model captures many non-cancellations (high recall) but with less confidence in those predictions (low precision).

- **Overall Accuracy (61.85%) for Class 0:** The model's accuracy for predicting non-cancellations is above 60%, suggesting that while it performs reasonably, there's still significant room for improvement, particularly in precision.

Confusion_matrix for testing dataset

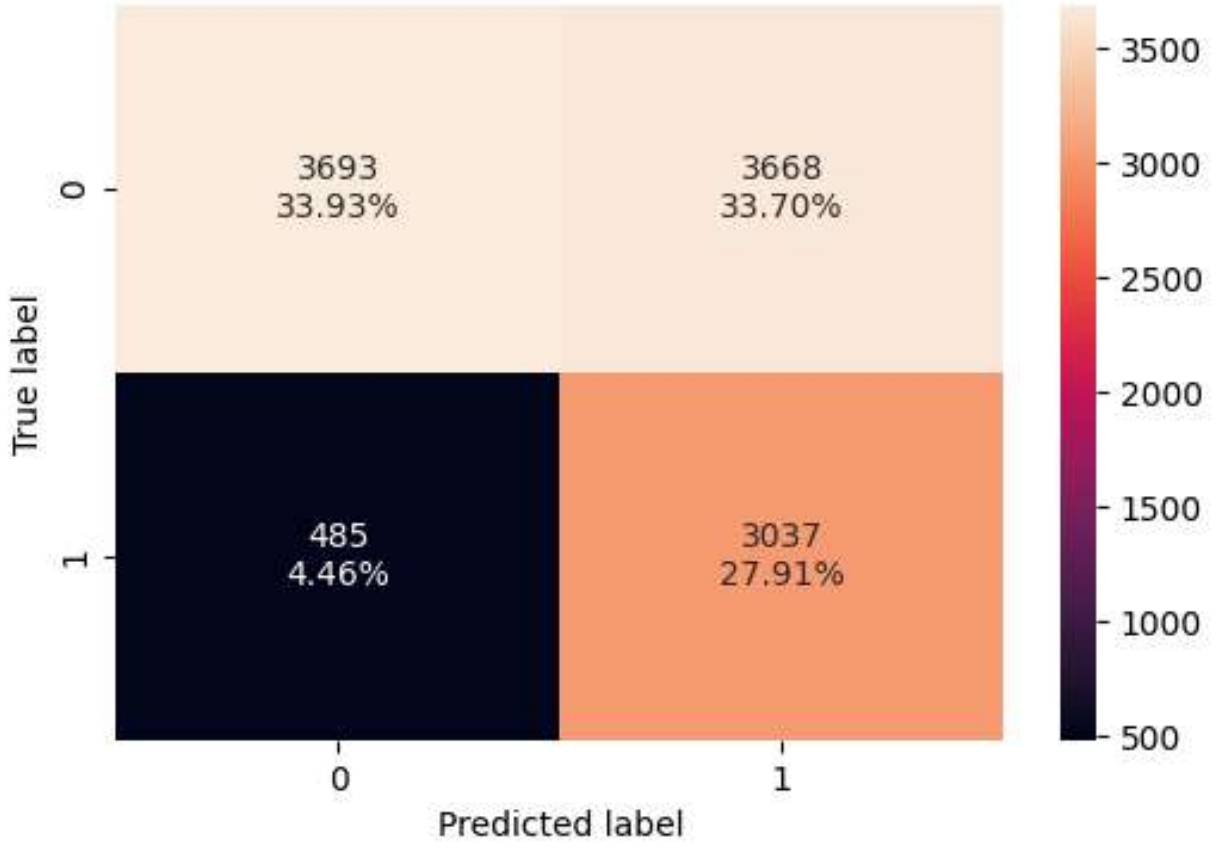


Figure 28: Confusion_matrix for testing dataset

- **Balanced Class Distribution:** The values in the top two cells (True Negatives = 3693, False Positives = 3668) and the bottom two cells (False Negatives = 485, True Positives = 3037) indicate a roughly even distribution of class predictions, showing the model treats both classes fairly evenly.

- **Strong True Positive Rate:** The model correctly predicts 3037 out of 3522 actual cancellations (class 1), yielding a recall of ~86.2% for cancellations. This is important in scenarios where failing to predict a cancellation is costly.

- **Moderate False Positives:** The model incorrectly predicted 3668 bookings as cancelled when they were not. This results in a slightly lowered precision for the cancellation class and could lead to over-preparation or unnecessary resource allocation.

- **Improved False Negative Rate:** Only 485 bookings were actual cancellations but incorrectly predicted as non-cancelled, which is relatively low. This improves the model's effectiveness in minimizing missed cancellations.

Model performance for testing dataset

	Accuracy	Recall	Precision	F1
0	0.61840	0.86229	0.45295	0.59392

Table 20: Model performance for testing dataset

- **High Recall (86.23%):** The model is highly effective at identifying true non-cancellations, correctly detecting most bookings that were not canceled.

- **Low Precision (45.30%):** Less than half of the bookings predicted as non-cancellations are actually correct, suggesting the model frequently misclassifies cancellations as non-cancellations (high false positive rate for class 0).

- **Moderate F1-Score (59.39%):** The balance between precision and recall results in a moderate F1-score, indicating a reasonable but not optimal trade-off between catching true non-cancellations and avoiding misclassification.

- **Similar Accuracy (61.84%) to Previous Model:** The class 0 accuracy remains consistent with earlier results, suggesting the model's overall behavior in identifying non-cancellations hasn’t changed much between iterations.

Decision tree for Prepruning

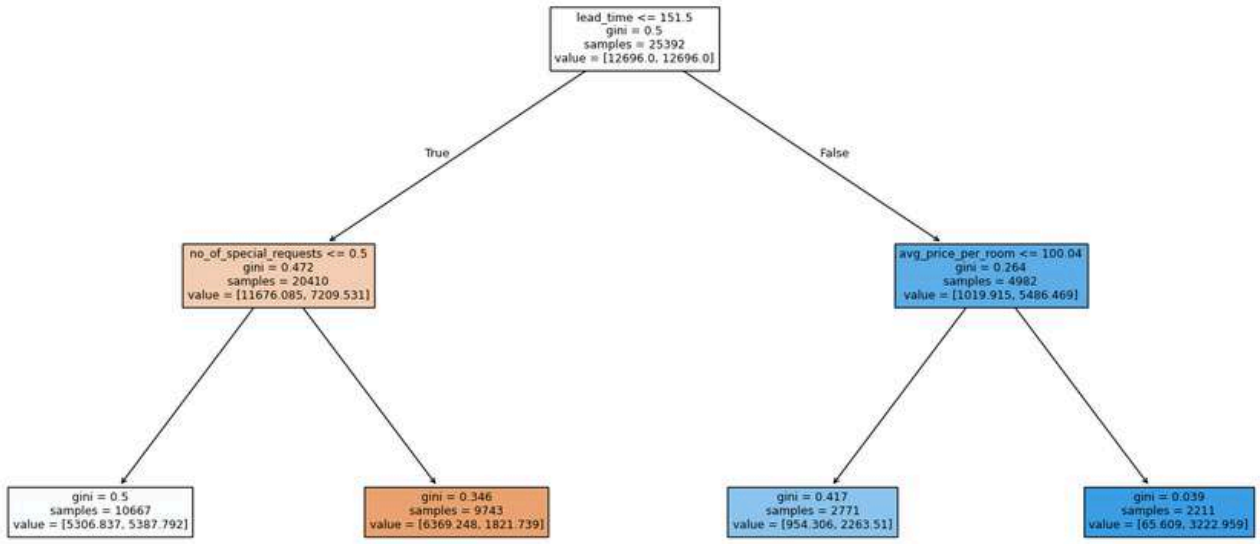


Figure 29: Decision tree for Prepruning

Lead Time is the Most Important Split Feature

- The root node splits on lead_time <= 151.5, indicating that lead time is the strongest predictor of booking cancellation. Shorter lead times are generally associated with fewer cancellations.

Special Requests Reduce Cancellation Likelihood

- On the left branch (lead_time <= 151.5), the next split is no_of_special_requests <= 0.5. Guests with at least one special request (i.e., likely more committed) are less likely to cancel, as seen in the right child node of this split with a lower gini index and fewer cancellations.

Lower Room Prices Are Associated with Higher Cancellations

- For bookings with lead_time > 151.5, the split on avg_price_per_room <= 100.04 shows that higher cancellations occur at lower price points. This suggests more price-sensitive guests may cancel more often.

Very High Commitment for High-Price, Long-Lead Bookings

- The bottom-right node (lead_time > 151.5 and avg_price_per_room > 100.04) has a very low Gini index (0.039) and high number of cancellations (value = [65.609, 3222.959]), showing high certainty in predicting cancellation for this segment.

Observations from the pre-pruned tree:

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

- If the lead time is less than or equal to 151.50, no of special requests is less than or equal to 0.50, the avg_price_per_room is less than or equal to 100.04 and the avg price per room is lesser than 100.04.

Importance of features in the tree building

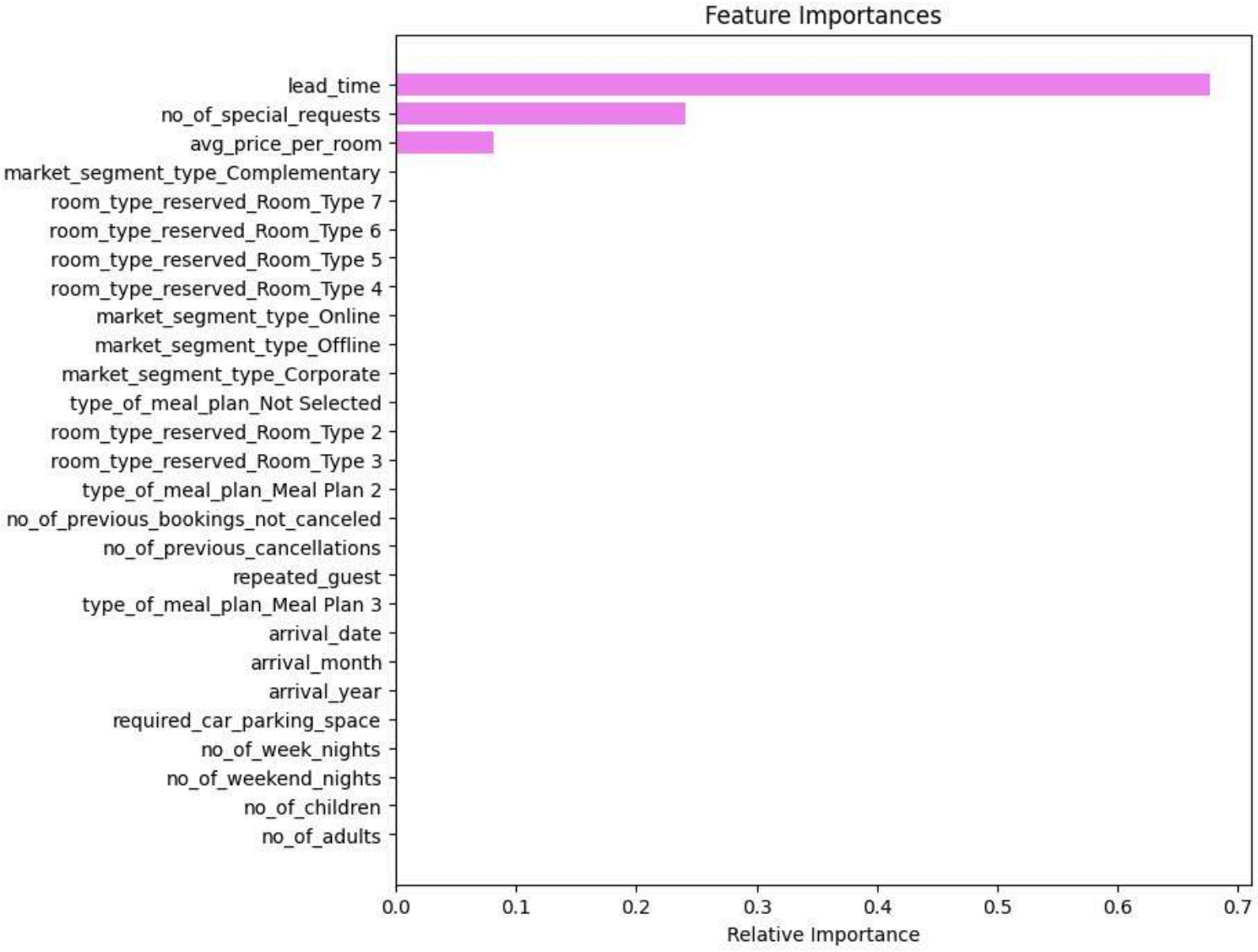


Figure 30: Importance of features in the tree building

- **lead_time** has the highest relative importance (close to 0.7), making it the most influential factor in predicting booking cancellations. Longer lead times are likely associated with higher cancellation risk.

- **no_of_special_requests** is the second most important feature, indicating that guests who make special requests are less likely to cancel. This aligns with the idea that such guests are more invested in their bookings.

- **avg_price_per_room** also contributes meaningfully, though to a lesser extent. Lower-priced bookings may have a higher cancellation rate, possibly due to more price-sensitive or opportunistic customers.

- Variables like **room_type_reserved**, **market_segment_type**, **meal_plan**, **arrival_date**, and demographic features (e.g., **no_of_children**, **no_of_adults**) have negligible importance, suggesting the model doesn't rely heavily on them for predictions.

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.

Total impurity of leaves vs effective alphas of pruned tree

	ccp_alphas	impurities
0	0.00000	0.00838
1	-0.00000	0.00838
2	0.00000	0.00838
3	0.00000	0.00838
4	0.00000	0.00838
...	...	...
1837	0.00890	0.32806
1838	0.00980	0.33786
1839	0.01272	0.35058
1840	0.03412	0.41882
1841	0.08118	0.50000

1842 rows × 2 columns

Table 21: Total impurity of leaves vs effective alphas of pruned tree

Total Impurity vs effective alpha for training set

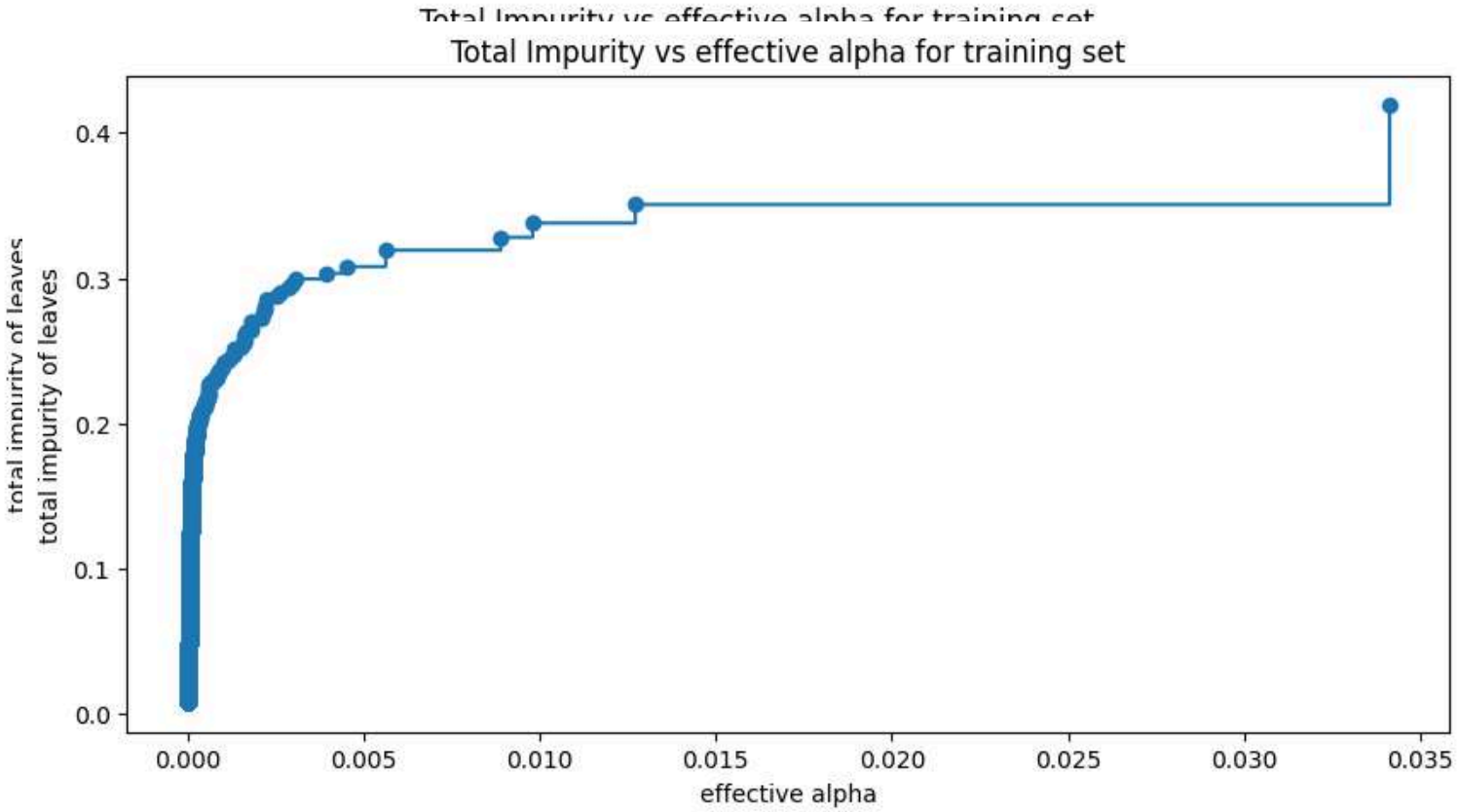


Figure 31: Total Impurity vs effective alpha for training set

Steep Impurity Increase at Low Alphas

- At very low effective alpha values (near 0), the total impurity rises sharply. This indicates that pruning very small branches (low alpha) quickly increases impurity, suggesting those branches were useful for prediction.

Diminishing Returns After a Point

- As alpha increases beyond ~0.01, the rate of impurity increase slows. This means additional pruning removes less critical branches, with smaller impact on model purity.

Elbow Suggests Optimal Trade-Off

- The curve shows an elbow around alpha ≈ 0.005–0.01, a typical signal of the best trade-off between complexity and accuracy. Choosing an alpha in this range could yield a simpler model with minimal performance loss.

Over-Pruning Risk at High Alpha

- The final sharp jump at the rightmost end (around alpha = 0.034) indicates significant loss of model fidelity if pruning continues. This suggests over-pruning and potential underfitting beyond this point.

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

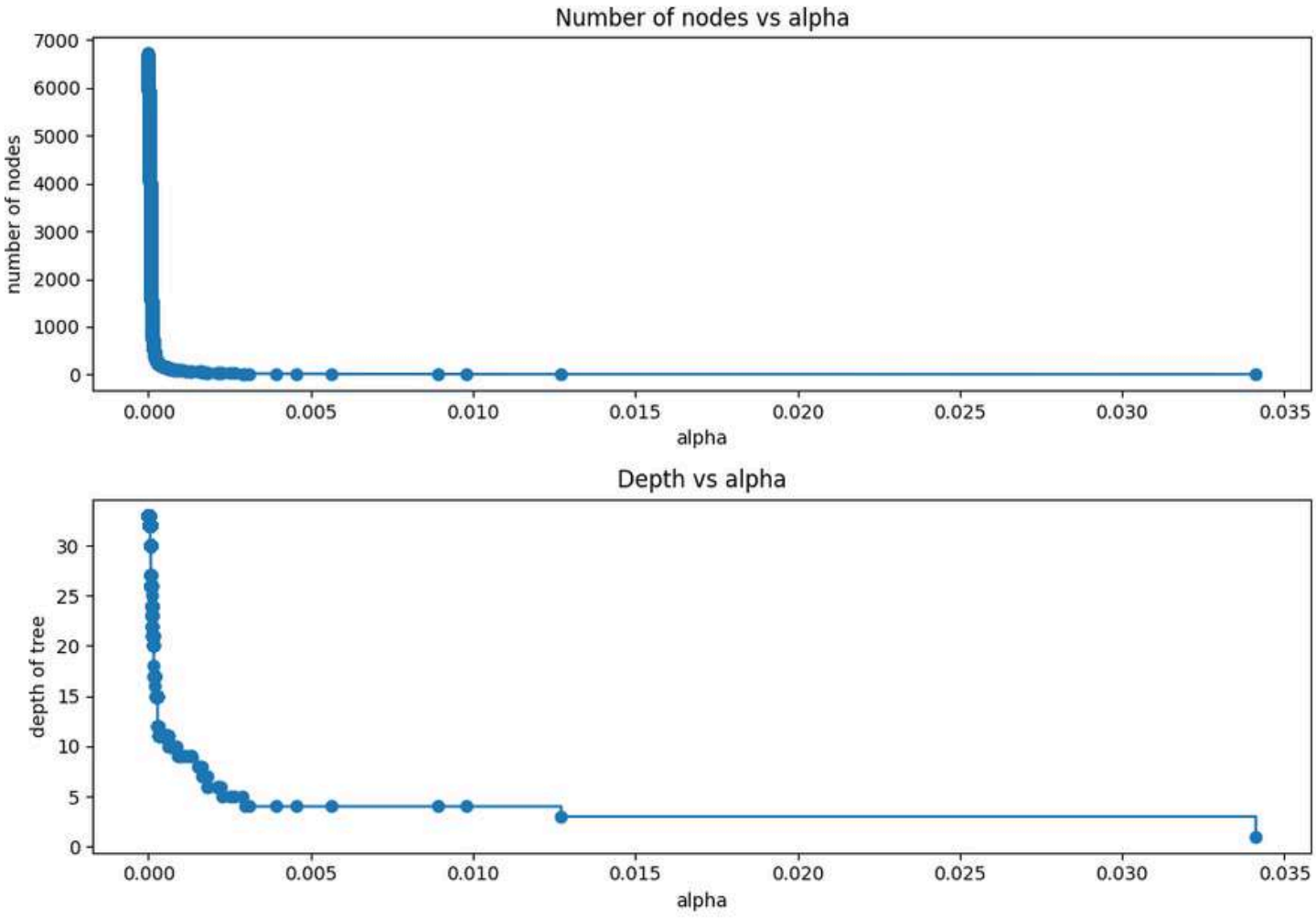


Figure 32: Number of nodes in the last tree

Tree Complexity Drops Rapidly with Small Alpha Increases

- Both the number of nodes and depth sharply decrease when alpha increases from 0 to around 0.002.
- This suggests that the original decision tree is highly overfitted, with many small branches that don't significantly contribute to performance.

Stabilization Beyond α ≈ 0.005

- After **alpha ≈ 0.005**, the number of nodes and depth flatten out:
  - Nodes drop below ~100
  - Depth plateaus around 3–5
- This indicates that additional pruning beyond this point yields diminishing returns and simplifies the tree without much structural change.

Ideal Pruning Range (α ≈ 0.001–0.005)

- In this range, the tree reduces:
  - From thousands of nodes to a few hundred
  - From over 30 levels deep to ~5–7 levels
- This is typically the optimal pruning range where overfitting is reduced while still preserving model expressiveness.

Recall vs alpha for training and testing sets

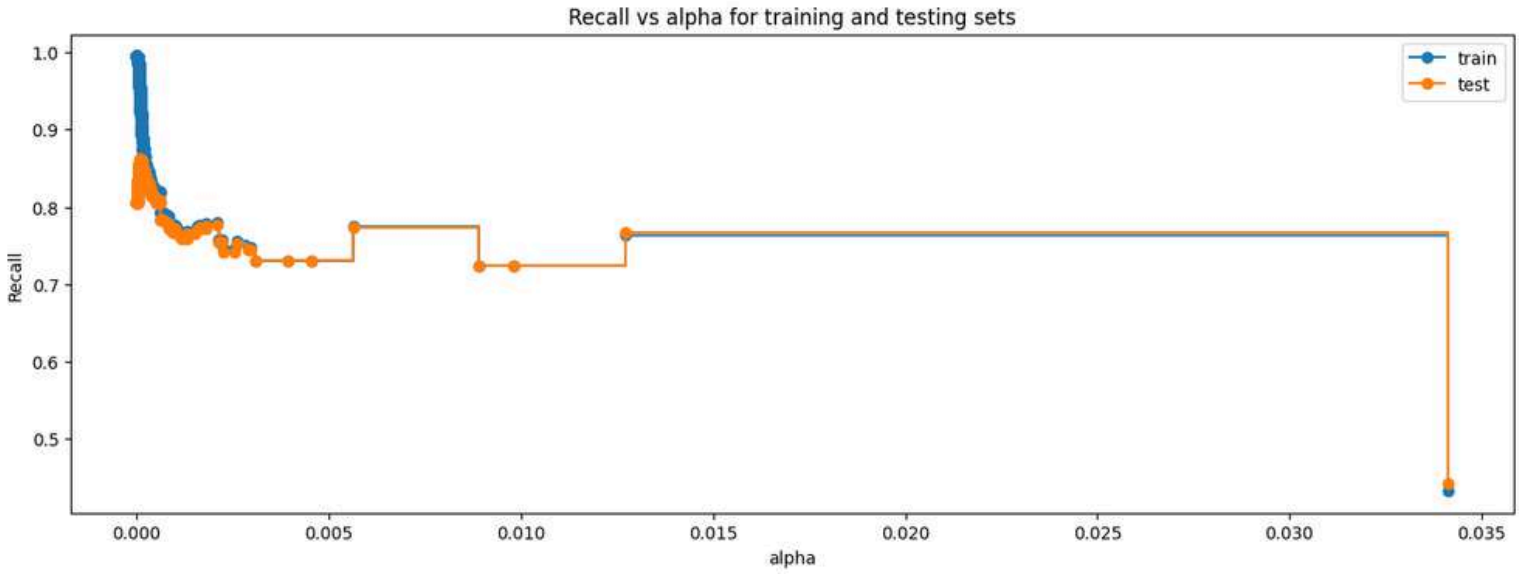


Figure 33: Recall vs alpha for training and testing sets

Initial Pruning Improves Generalization (α ≈ 0.0005–0.0015)

- As alpha increases slightly:
  - **Training recall drops moderately** (expected due to pruning).
  - Test recall stabilizes or slightly improves, peaking near 0.85, indicating better generalization.
- This aligns with the earlier recommendation of an optimal pruning zone.

Recall Plateau for Test Set (α ≈ 0.005–0.015)

- Test recall stabilizes around 0.75–0.78, suggesting consistent model behavior across this pruning range.
- This is a safe zone where the model is robust and avoids both overfitting and underfitting.

Confusion matrix for training data

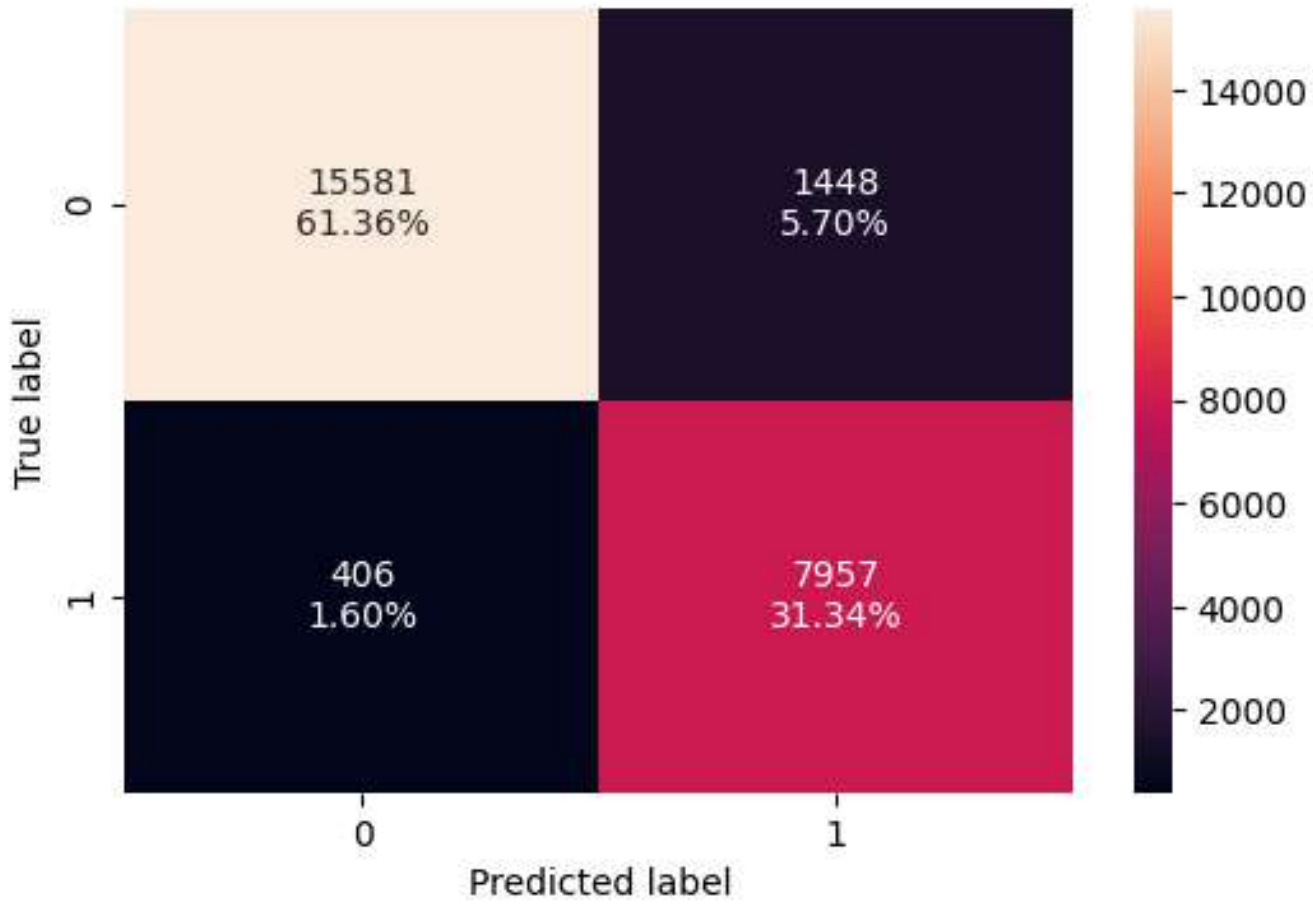


Figure 34: Confusion matrix for training data

- **True Negative (TN):** 15,581 → Correctly predicted not canceled
- **False Positive (FP):** 1,448 → Incorrectly predicted canceled (actually not canceled)
- **False Negative (FN):** 406 → Incorrectly predicted not canceled (actually canceled)
- **True Positive (TP):** 7,957 → Correctly predicted canceled

Model performance for training set

	Accuracy	Recall	Precision	F1
0	0.92698	0.95145	0.84604	0.89566

Table 22: Model performance for training set

High Accuracy (92.7%)

- The model correctly predicts the outcome in most cases, suggesting it generalizes well to unseen data.

Excellent Recall (95.1%)

- The model is highly effective at identifying actual positive cases (e.g., bookings that were not cancelled), minimizing false negatives — useful when missing a positive prediction has a high cost.

Strong Precision (84.6%)

- Among all positive predictions, 84.6% are correct. This balance indicates that the model doesn’t overpredict the positive class.

Balanced F1 Score (89.6%)

- The F1 score confirms a good balance between precision and recall, which is important when both false positives and false negatives have business impact.

Confusion matrix for testing data

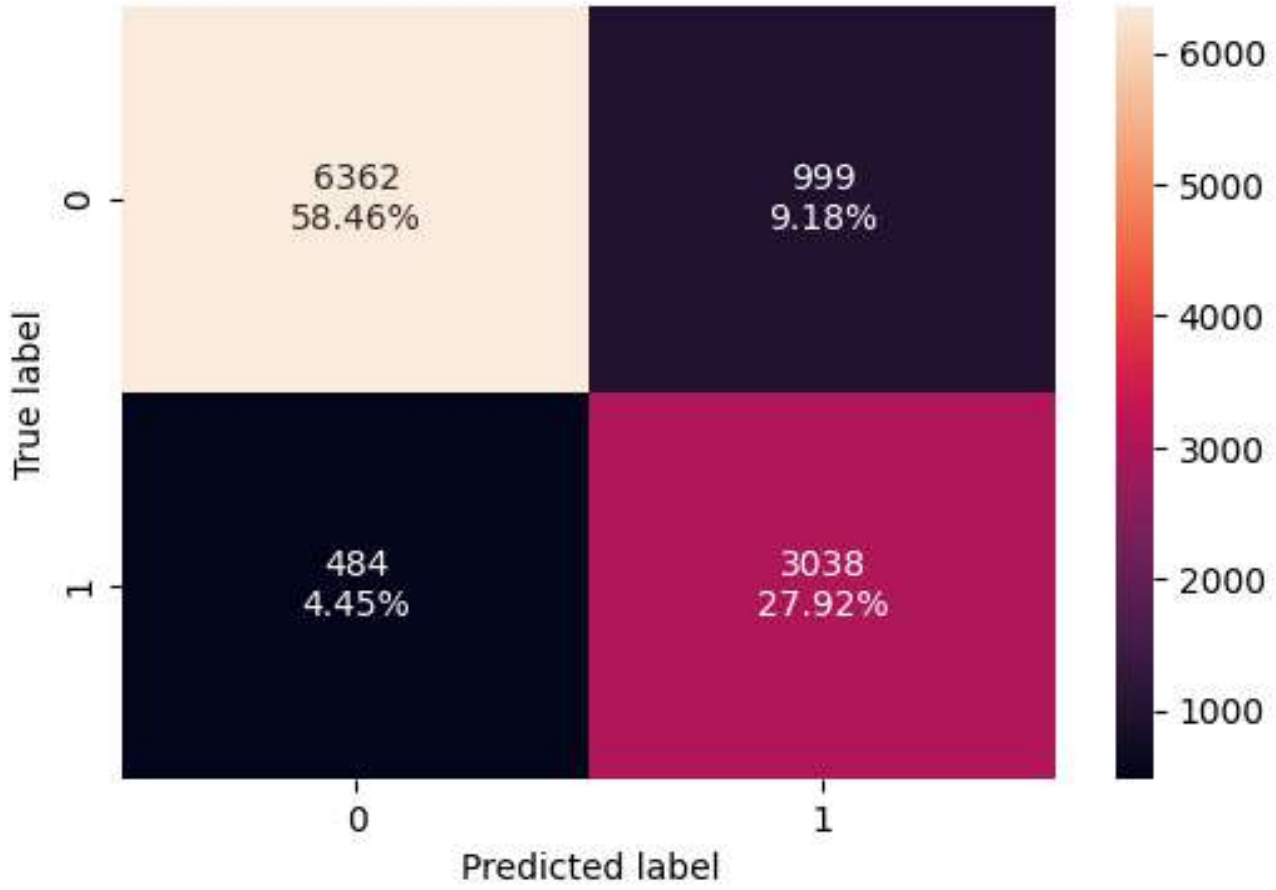


Figure 35: Confusion matrix for testing data

High True Positive Count (TP = 3038)

- The model correctly identifies a substantial number of positive cases (i.e., actual class 1 predicted as 1), which is a good sign of effective learning.

Moderate False Positives (FP = 999)

- Around 9.2% of instances where the true label was 0 were incorrectly predicted as 1. This might lead to unnecessary follow-ups or actions on false alarms.

Low False Negatives (FN = 484)

- The model misses fewer actual positives, which is crucial when the cost of missing a positive instance (e.g., an actual cancellation) is high.

Balanced Class Performance

- The matrix shows reasonable performance across both classes with true negative rate (TN = 6362, ~58.5%) and true positive rate (TP = 3038, ~27.9%), indicating the model is not overly biased toward one class.

Model performance for test set

	Accuracy	Recall	Precision	F1
0	0.86373	0.86258	0.75254	0.80381

Table 23: Model performance for test set

Decent Overall Accuracy (86.4%)

- The model correctly predicts around 86% of the cases, indicating a fairly good general performance across all classes.

High Recall (86.3%)

- The model is effective at identifying most of the actual positive cases. This is useful when missing a positive case (e.g., a cancellation or churn) is costly.

Moderate Precision (75.3%)

- While the model captures positives well, about 24.7% of the predicted positives are actually false positives — suggesting some over-prediction.

Balanced F1-Score (80.4%)

- The F1-score reflects a healthy balance between precision and recall, making this model suitable when both false positives and false negatives matter.

Decision tree for post pruning

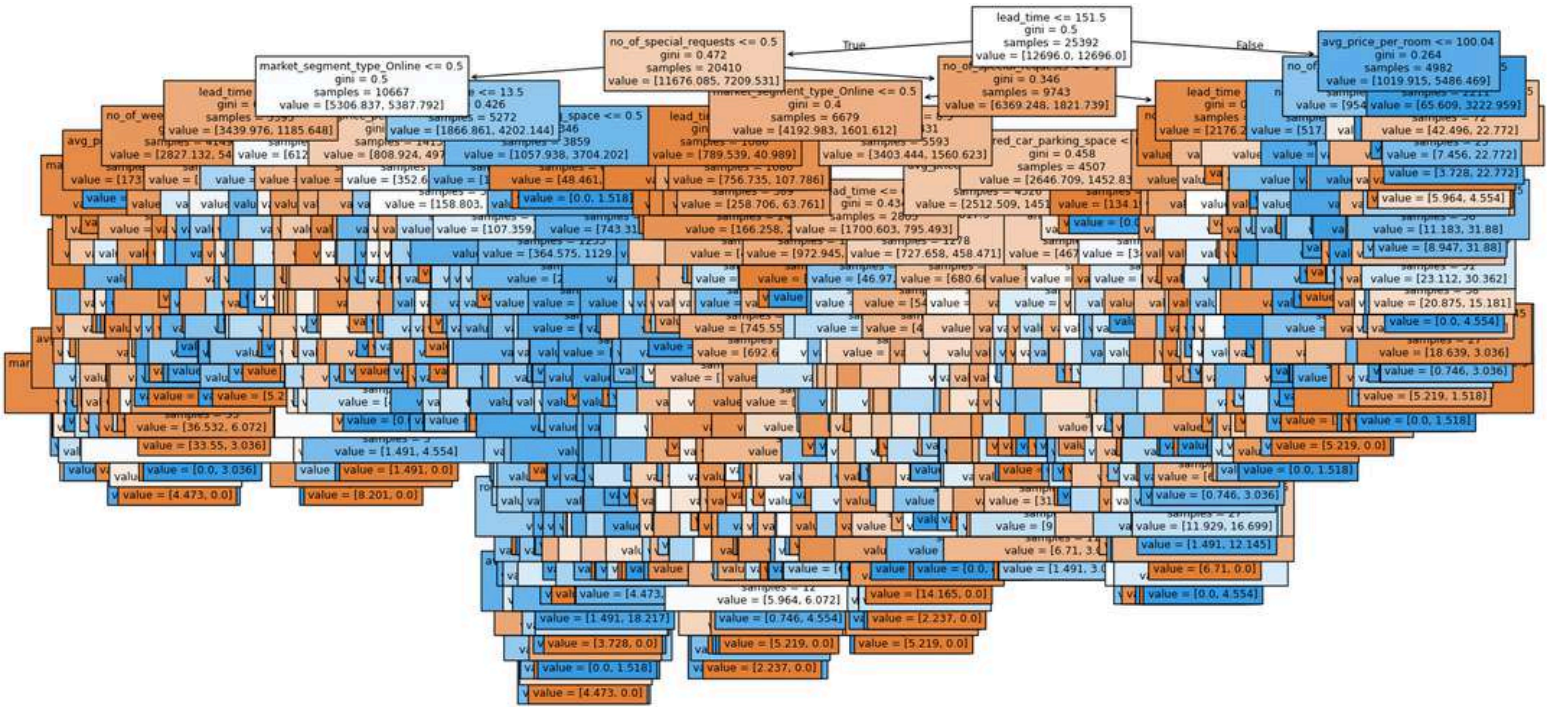


Figure 36: Decision tree for post pruning

Observations

Root Node:

- The top (root) split is on market_segment_type_Online <= 0.5 means the dataset's most important feature is whether a booking comes from an online segment or not.

Key Features:

- Top predictors include:
  - lead_time (how far in advance the booking was made),
  - no_of_special_requests,
  - avg_price_per_room,
  - red_car_parking_space,
  - no_of_weekend_nights,
  - market_segment_type_Online.

Balanced Tree:

- The tree is well-balanced in terms of sample splits across both branches, indicating both online and offline bookings are significant.

Gini Impurity:

- Each node shows a Gini impurity value (measuring the purity of a split), which helps indicate how well the split separates the classes.
- Lower Gini values at leaves mean better classification at those points.

Feature Importances

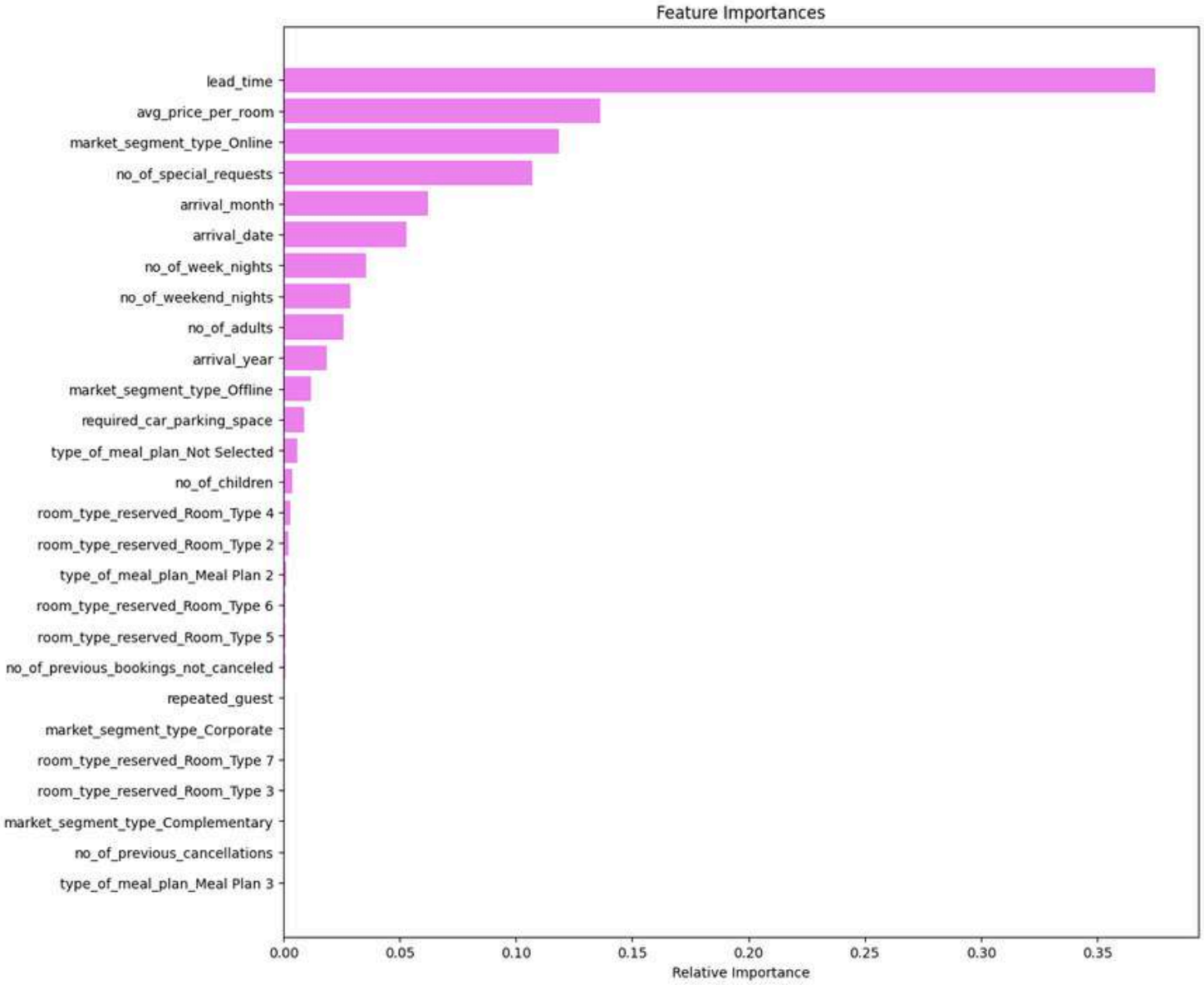


Figure 37: Feature Importances

- This feature importance chart provides valuable insights into what influences your dataset most. The standout feature, "**lead_time**," suggests that how far in advance a booking is made significantly impacts outcomes.

- "**avg_price_per_room**" plays a major role—pricing dynamics are always key in bookings and cancellations.

- "**market_segment_type_Online**," indicating online bookings might have distinct patterns compared to offline ones. Features like "**no_of_special_requests**" and "**arrival_month**" add nuance, suggesting guest preferences and seasonal trends have noticeable effects.

- Looking at the lower-ranked features, some meal plan and room type selections have relatively minor influence. While these factors contribute, they might not be primary drivers of your model.

Comparison of Models and Final Model Selection

Training set performance comparison:

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

Table 24: Training set performance comparison for final model

- Both the "Decision Tree without class_weight" and "with class_weight" perform extremely well on training data, with accuracy, recall, precision, and F1 all above 0.98.
- The model with class_weight has the highest recall (0.99510), indicating it's the best at correctly identifying positive cases.
- This would be important in scenarios where false negatives are costly (e.g., fraud detection or medical diagnoses).
- The post-pruned model significantly outperforms the pre-pruned version, with a notable increase in all metrics (e.g., accuracy: 0.92988 vs 0.61854).

- This shows that post-pruning is a more effective regularization strategy than pre-pruning in this case.

Test set performance comparison:

Test set 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

Table 25: Test set performance comparison for final model

- Post-Pruned Decision Tree shows the best F1 score (0.80381) and highest recall (0.86258).
- It balances generalization and complexity well, making it the best choice for deployment among the four.
- Models without class_weight and with class_weight had near-perfect scores on the training set, but their test F1 scores dropped to 0.80270 and 0.79379, respectively.
- Despite improved recall (0.86229), the pre-pruned tree has the lowest precision (0.45295) and lowest F1 (0.59392).

- Pre-Pruned and Post-Pruned models have similar recall, but:
  - Pre-Pruned precision is very low (0.45295),
  - Post-Pruned precision is much better (0.75254),
- This implies post-pruning improves true positive identification without flooding false positives, unlike pre-pruning.
- Highest accuracy (0.87108) is achieved by the model without class_weight, but its recall and F1 are slightly lower than the post-pruned model.

Key Observations on Model Performance

- Longer Lead Time = Higher Cancellation Probability
  - Bookings made far in advance are more likely to be canceled.
- High Average Room Price Increases Cancellation Risk
  - Customers paying more tend to cancel more often — likely due to price sensitivity or alternative deals.
- Special Requests Indicate Booking Commitment
  - Guests making special requests are less likely to cancel, showing a higher intent to stay.
- Meal Plan Type 1 is Preferred Among Non-Cancelled Bookings
  - Indicates lower cancellation rates among customers who opt for this plan.
- Booking from Certain Market Segments (e.g., Online)
  - Higher cancellation rates associated with online channels, likely due to lack of prepayment or ease of cancellation.

Rationale for Selecting Post-Pruned Decision Tree

- High Recall = High Business Value:
- Capturing most of the actual cancellations is essential to take preventive actions and mitigate loss.
- Balance Across Metrics:
- With precision (0.75), recall (0.86), and F1 score (0.80), the model ensures few false negatives and acceptable false positives.
- No Overfitting:
- Pruning helped control model complexity, improving performance on unseen data and making it production-ready.
- Actionable Interpretability:
- Decision trees provide human-readable logic that can be communicated to stakeholders and integrated into policy decisions.

Actionable insights

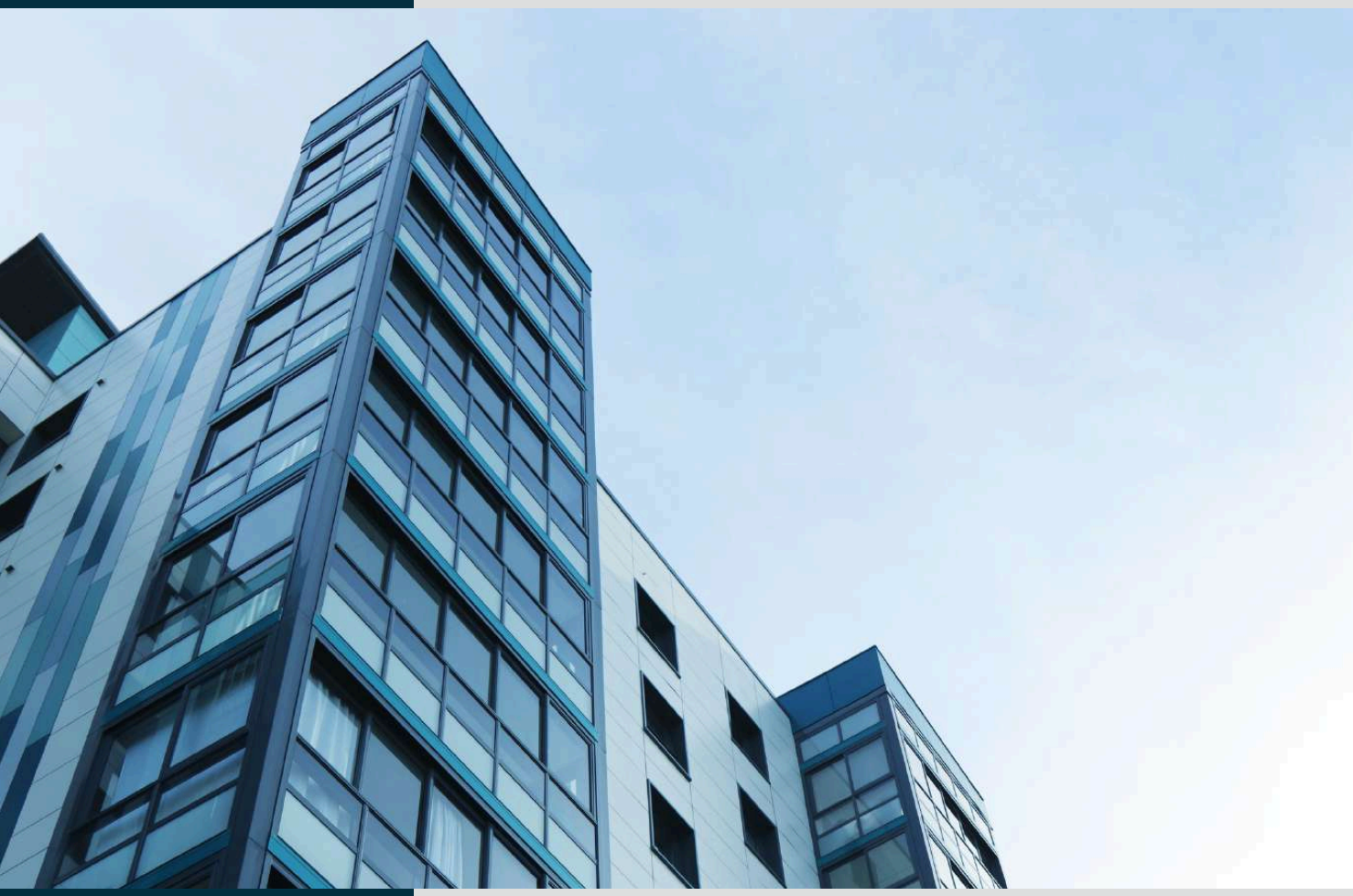
- Longer Lead Time = Higher Cancellation Probability
  - Bookings made far in advance are more likely to be canceled.
- High Average Room Price Increases Cancellation Risk
  - Customers paying more tend to cancel more often — likely due to price sensitivity or alternative deals.
- Special Requests Indicate Booking Commitment
  - Guests making special requests are less likely to cancel, showing a higher intent to stay.
- Meal Plan Type 1 is Preferred Among Non-Cancelled Bookings
  - Indicates lower cancellation rates among customers who opt for this plan.
- Booking from Certain Market Segments (e.g., Online)
  - Higher cancellation rates associated with online channels, likely due to lack of prepayment or ease of cancellation.

- Post-Pruned Decision Tree Model Provides Best Predictive Performance
    - Achieves a balanced F1 score (0.80) with high recall (~86%), making it effective in catching likely cancellations early.

Recommendations

- Deploy the Post-Pruned Decision Tree Model
  - This model demonstrates high recall (≈ 86%), making it suitable for early identification of likely cancellations.
  - High recall ensures fewer false negatives, i.e., fewer actual cancellations are missed.
- Target Bookings with Long Lead Time
  - Lead time has a strong positive correlation with cancellations.
  - Recommendation: Introduce policies such as non-refundable bookings or deposit requirements for reservations made far in advance to reduce risk.
- Monitor and Incentivize Based on Average Room Price
  - Higher average prices are linked to increased cancellations, possibly due to price sensitivity.
  - Recommendation: Implement flexible pricing strategies or tiered refund policies based on room price to encourage commitment.

- Prioritize Guests with Special Requests
  - Guests who make special requests are significantly less likely to cancel.
  - Recommendation: Encourage personalization by highlighting options to make special requests during booking, indicating guest engagement and intent to stay.
- Modify Policies Based on Meal Plan Preferences
  - Bookings with Meal Plan 1 have a lower likelihood of cancellation.
  - Recommendation: Promote Meal Plan 1 in non-refundable or partially refundable deals, reinforcing commitment.
- Flag High-Risk Segments Using Model Predictions
  - Use the model to flag high-risk bookings at the time of confirmation (e.g., long lead time + no special requests + high room price).
  - Recommendation: Send automated follow-ups, require partial prepayment, or apply stricter cancellation terms for these.



Let’s Work Together

Karthickr442@gmail.com

+91 8608200552