Machine Learning - 1 Project

BUSINESS REPORT

Table of Contents

List of Tables

List of Figures

Problem Statement

Context

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

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

The cancellation of bookings impacts a hotel on various fronts:

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

Objective

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

Data Dictionary

- Booking_ID: the unique identifier of each booking
- no_of_adults: Number of adults
- no_of_children: Number of Children

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

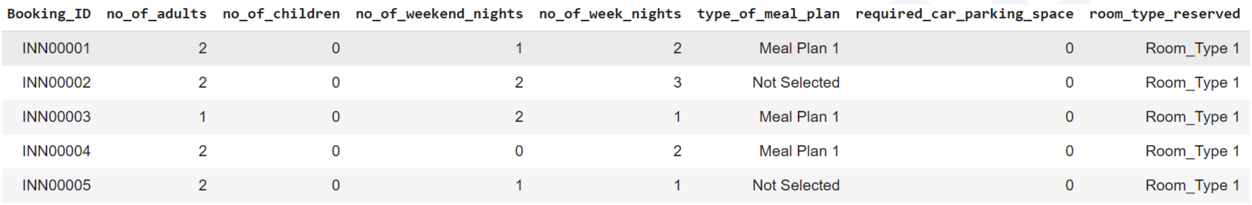
Loading the Dataset and Data Overview

The dataset provided was loaded into a pandas dataframe for analysis.

Getting the first 5 and last 5 rows

The dataset has been loaded successfully. It has 36275 rows and 19 columns.

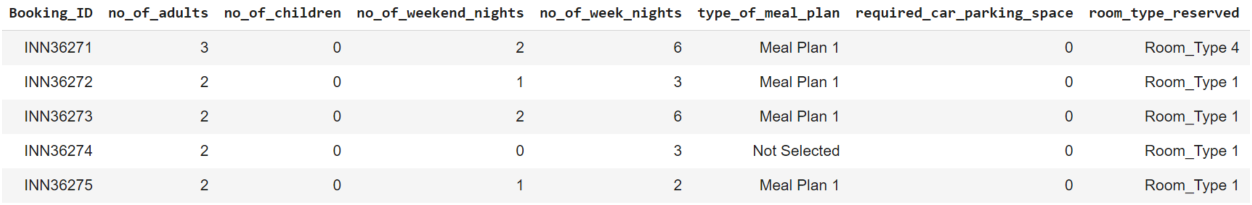
- First 5 rows of the dataset:



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
INN00001	2	0	1	2	Meal Plan 1	0	Room_Type 1
INN00002	2	0	2	3	Not Selected	0	Room_Type 1
INN00003	1	0	2	1	Meal Plan 1	0	Room_Type 1
INN00004	2	0	0	2	Meal Plan 1	0	Room_Type 1
INN00005	2	0	1	1	Not Selected	0	Room_Type 1

Table 1: First 5 rows of the dataset

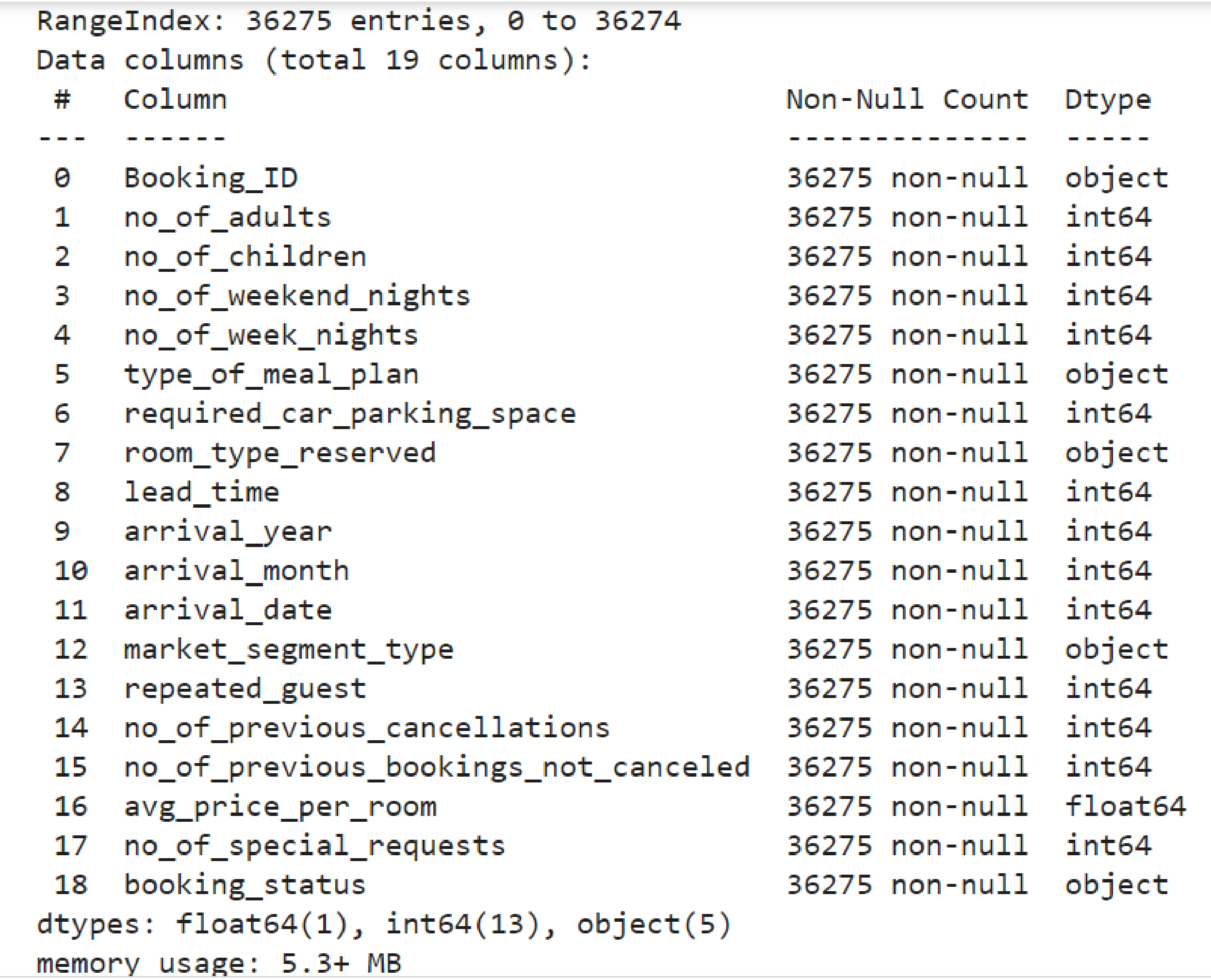
- Last 5 rows of the dataset:



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
INN36271	3	0	2	6	Meal Plan 1	0	Room_Type 4
INN36272	2	0	1	3	Meal Plan 1	0	Room_Type 1
INN36273	2	0	2	6	Meal Plan 1	0	Room_Type 1
INN36274	2	0	0	3	Not Selected	0	Room_Type 1
INN36275	2	0	1	2	Meal Plan 1	0	Room_Type 1

Table 2: Last 5 rows of the dataset

Information about the dataset



```
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 3: Information about the dataset

Checking for Duplicate values

There are no duplicate values in the dataset.

Data types of the variables

Data type	No. of variables
object	5

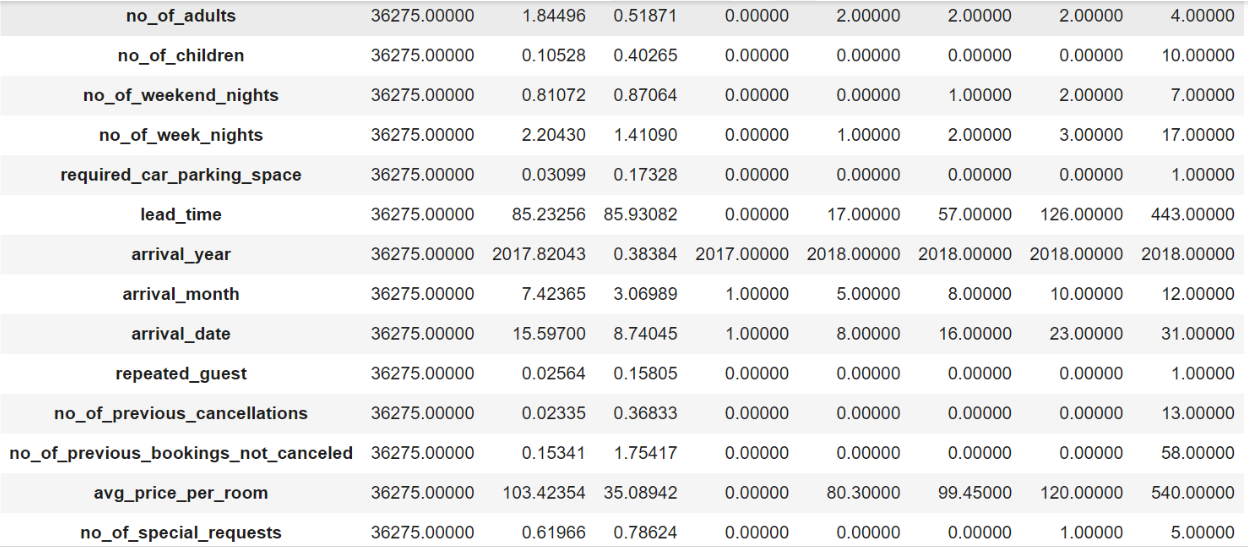
float64	1
int64	13

Table 4: Datatypes of the variables in the dataset

- Booking_ID is an object because it contains unique identifiers for all the bookings. However, this is not used in any model building or EDA and is therefore dropped.
- We are now left with 18 variables.
- The last column, **booking_status** is the target variable and contains values (‘Cancelled’ and ‘Not_Cancelled’)

Statistical Description of the Dataset

A description of the columns of the dataset is given below:



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 5: Statistical Summary of the dataset

From the descriptive statistics, we can conclude that:

- The number of adults ranges from 0 to 4, which is usual.
- The maximum value in the number of children column is 10, which is a bit unusual and might require a check.
- The range of the number of weeks and weekend nights seems fine. Though 7 weekends

might be a very long stay.

- At least 75% of the customers do not require car parking space.
- On average the customers book 85 days in advance. There's also a very huge difference in the 75th percentile and maximum value which indicates there might be outliers present in this column.
- We have two years of data, 2017 and 2018.
- At least 75% of the customers are not repeating customers.
- The average price per room is 103 euros. There's a huge difference between the 75th percentile and the maximum value which indicates there might be outliers present in this column.

Exploratory Data Analysis

Univariate Analysis

In this section, we will analyze the distribution of independent variables. It will help us identify the pattern among the variables and the effects they have on the target variable.

First, let us see how the target variable (booking_status) is distributed.

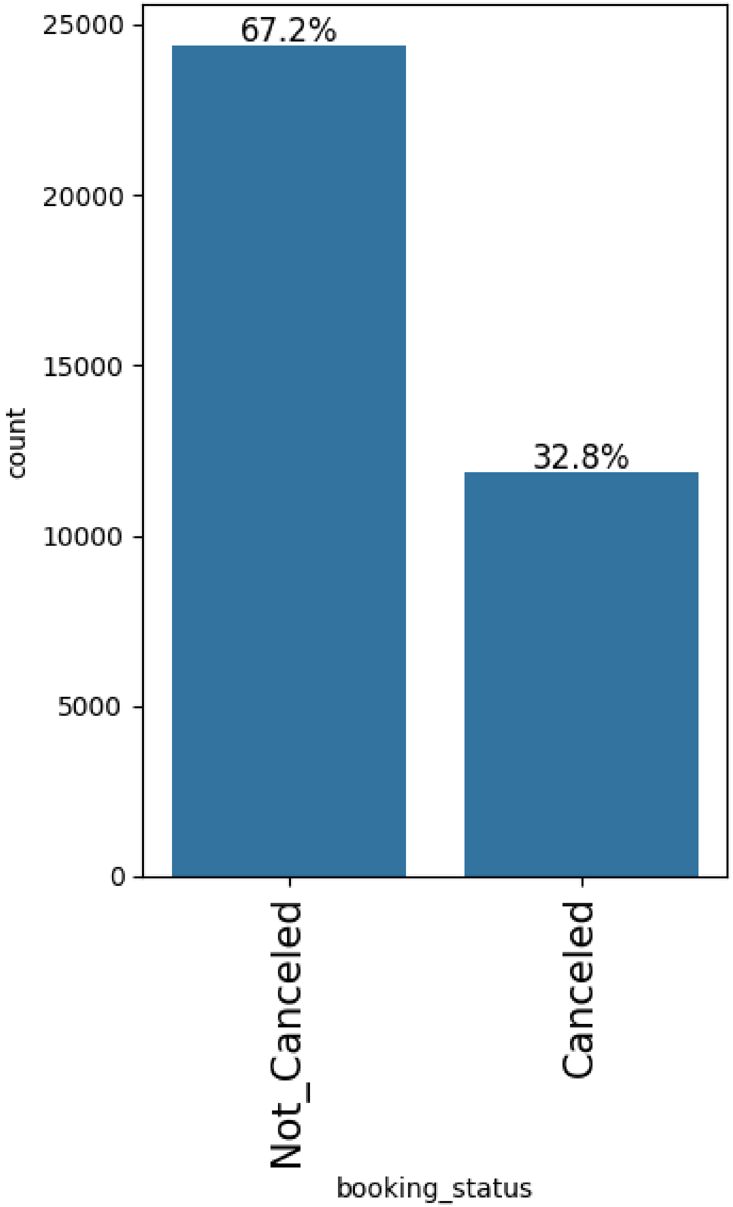
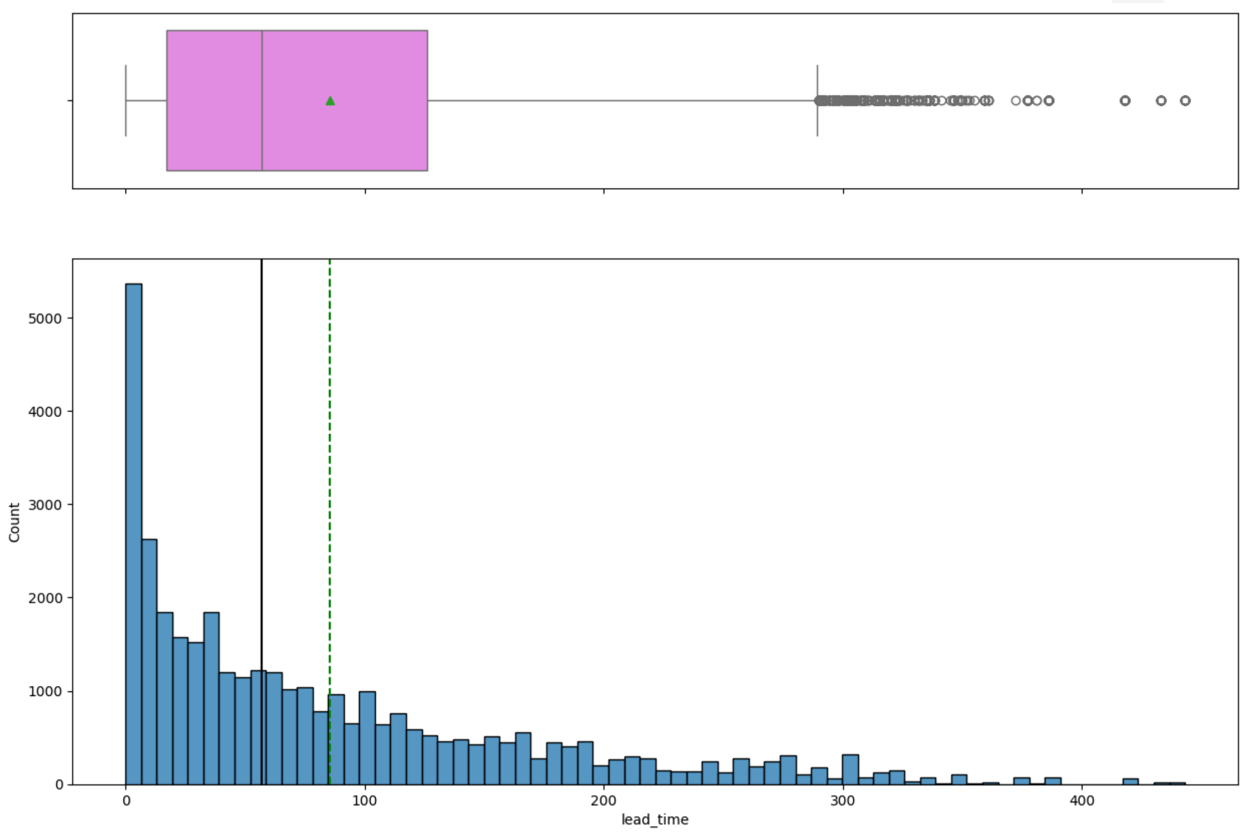


Fig 1: Countplot of the target variable (booking_status)

From the above plot, we can see that 32.8% of the bookings were canceled by the customers.

Now, let’s understand the distribution of other variables.



Observations on lead_time

Fig 2: Observations on lead_time

- The distribution of lead time is right-skewed, and there are many outliers.
- Some customers made bookings around 500 days in advance.
- Many customers have made the booking on the same day of arrival as well.

Observations on avg_price_per_room

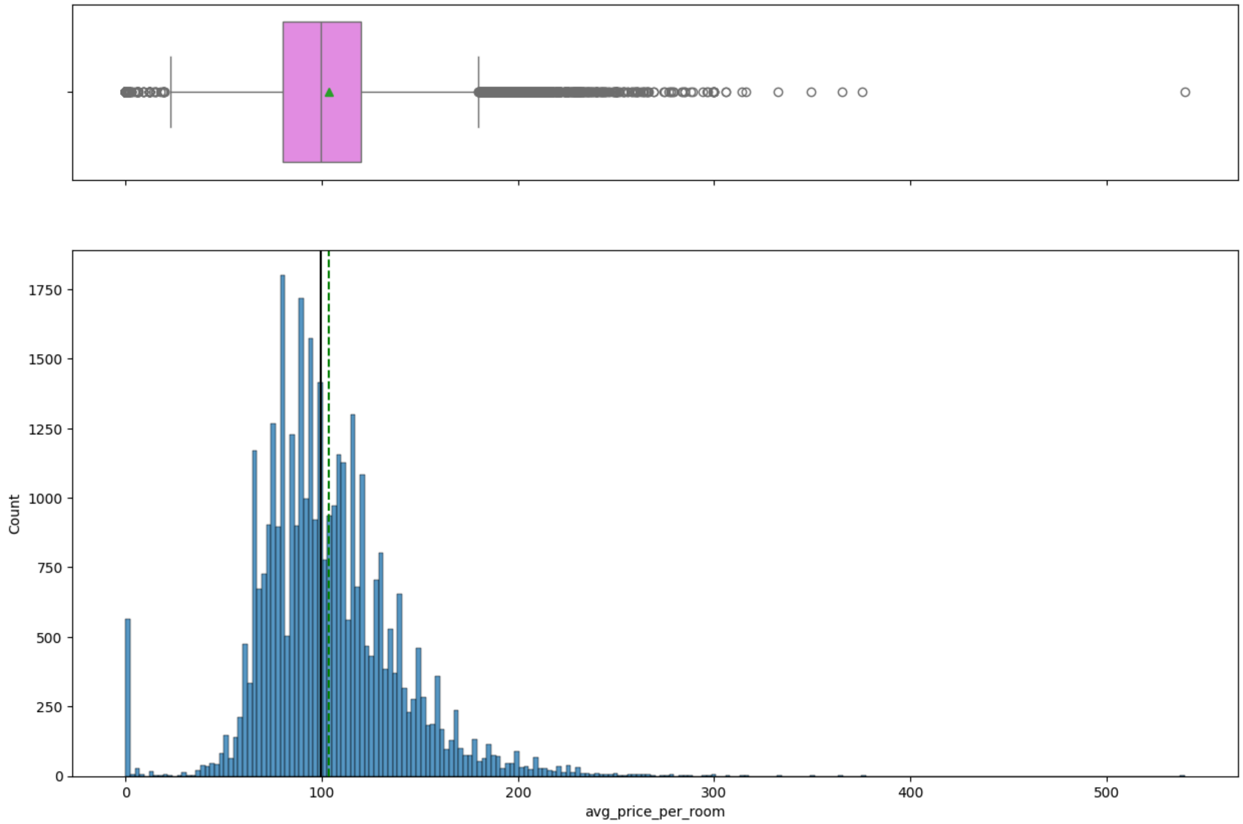


Fig 3: Observations on avg_price_per_room

- The distribution of the average price per room is skewed to the right. There are outliers on both sides.
- The average price of a room is around ~100 euros.
- There is 1 observation where the average price of the room is more than 500 euros. This observation is quite far away from the rest of the values. Instead of dropping it, we will clip this to the upper whisker (Q3 + 1.5 * IQR).
- Interestingly some rooms have a price equal to 0.
- It makes sense that most values with room prices equal to 0 are the rooms given as complimentary service by the hotel.
- The rooms booked online must be a part of some promotional campaign done by the hotel.

Observations on the no_of_adults

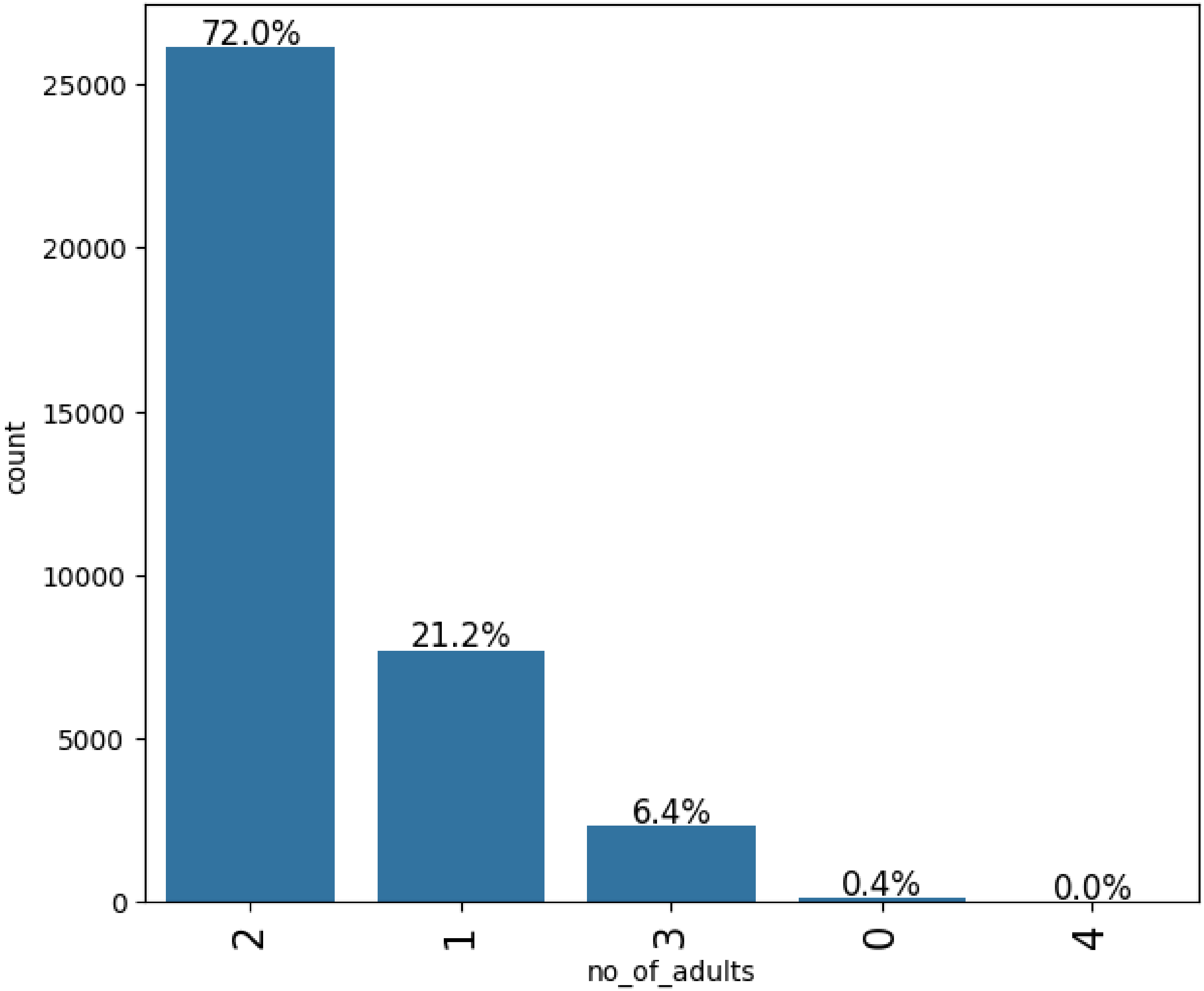


Fig 4: Observations on the no_of_adults

- 72% of the bookings were made for 2 adults.

Observations on no_of_children

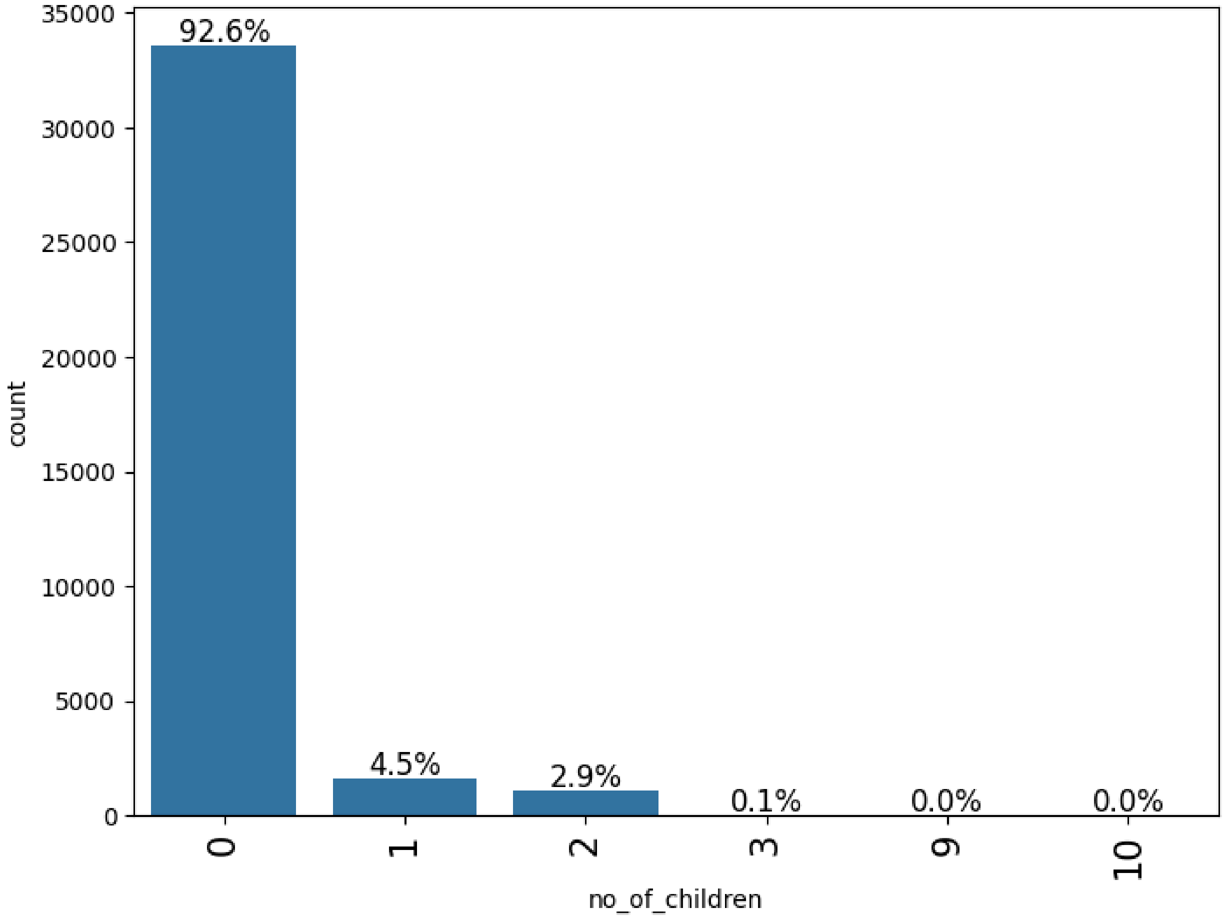


Fig 5: Observations on no_of_children

- 93% of the customers didn't make reservations for children.
- There are some values in the data where the number of children is 9 or 10, which is highly unlikely.
- We will replace these values with the maximum value of 3 children.

Observations on no_of_week_nights

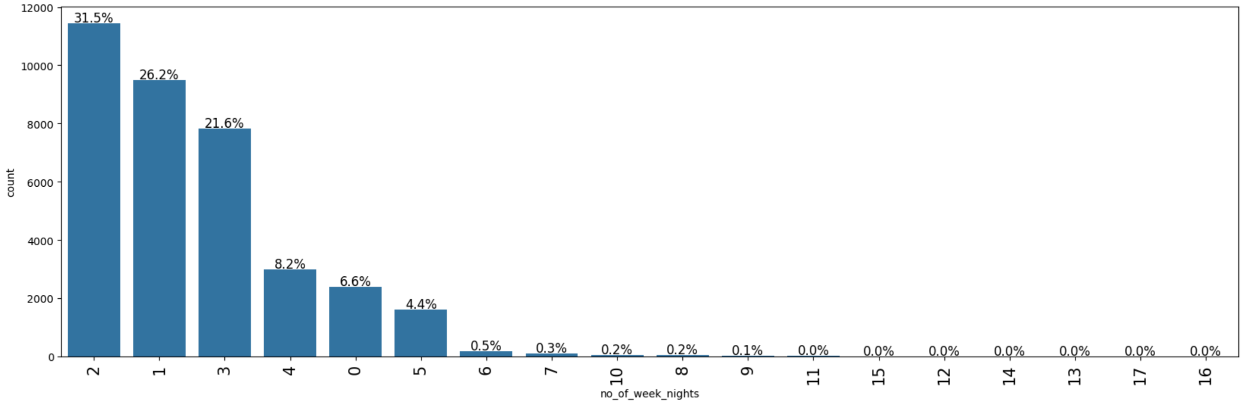


Fig 6: Observations on no_of_week_nights

- Most bookings are made for 2 nights (31.5%) followed by 1 night (26.2%).
- A very small proportion of customers made the booking for more than 10 days.

Observations on required_car_parking_space

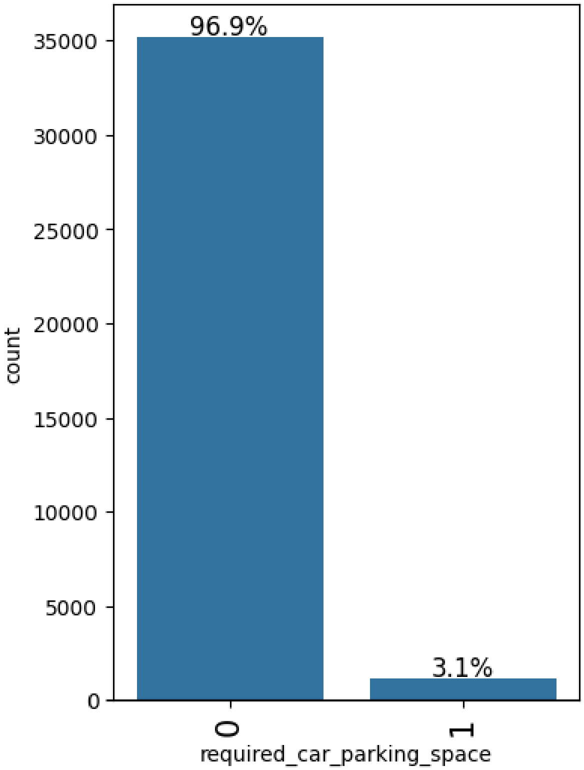


Fig 7: Observations on required_car_parking_space

- 96.9% of the customers do not require a car parking space.

Observations on type_of_meal_plan

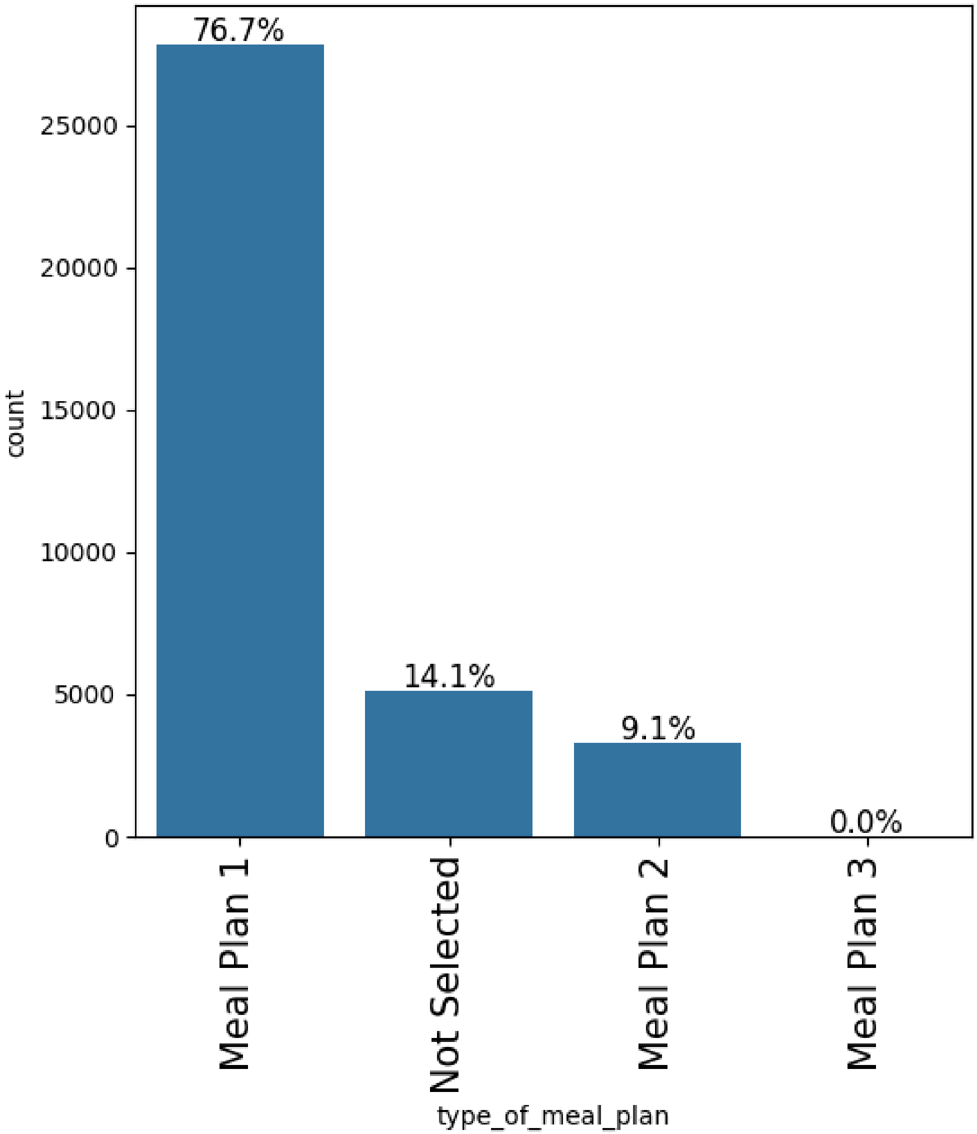


Fig 8: Observations on type_of_meal_plan

- Most of the customers prefer meal plan 1 which is only breakfast.
- 14.1% of the customers didn't select a meal plan.

Observations on arrival_month

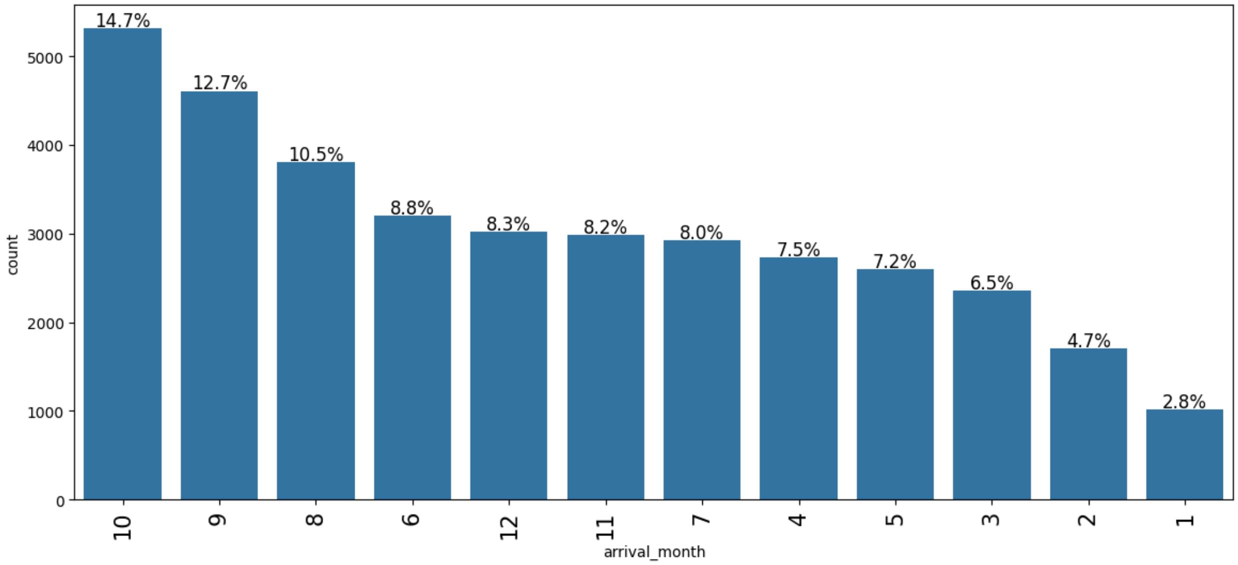


Fig 9: Observations on arrival_month

- October is the busiest month for the hotel followed by September.
- 14.7% of the bookings were made in October.

Bivariate Analysis:

For bivariate analysis, we can analyze the contribution of variables in determining the booking_status which is the target variable. This tells us whether the booking was canceled or not.

Observations on market_segment_type and avg_price_per_room

Hotel rates are dynamic and change according to demand and customer demographics. Let's see how prices vary across different market segments.

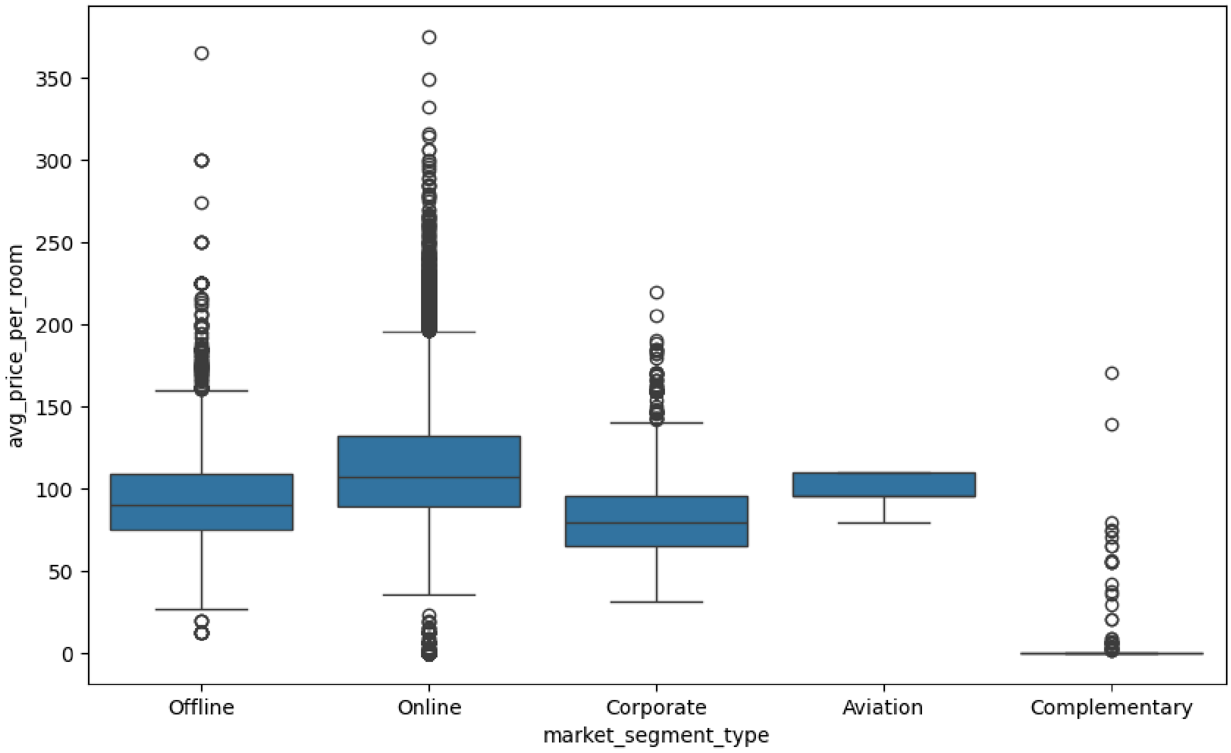


Fig 10: Observations on market_segment_type and avg_price_per_room

- Rooms booked online have high variations in prices.
- The offline and corporate room prices are almost similar.
- The complementary market segment gets the rooms at meager prices, which makes sense.

Let's see if the special requests made by the customers impact the prices of a room

Observations on no_of_special_requests and avg_price_per_room

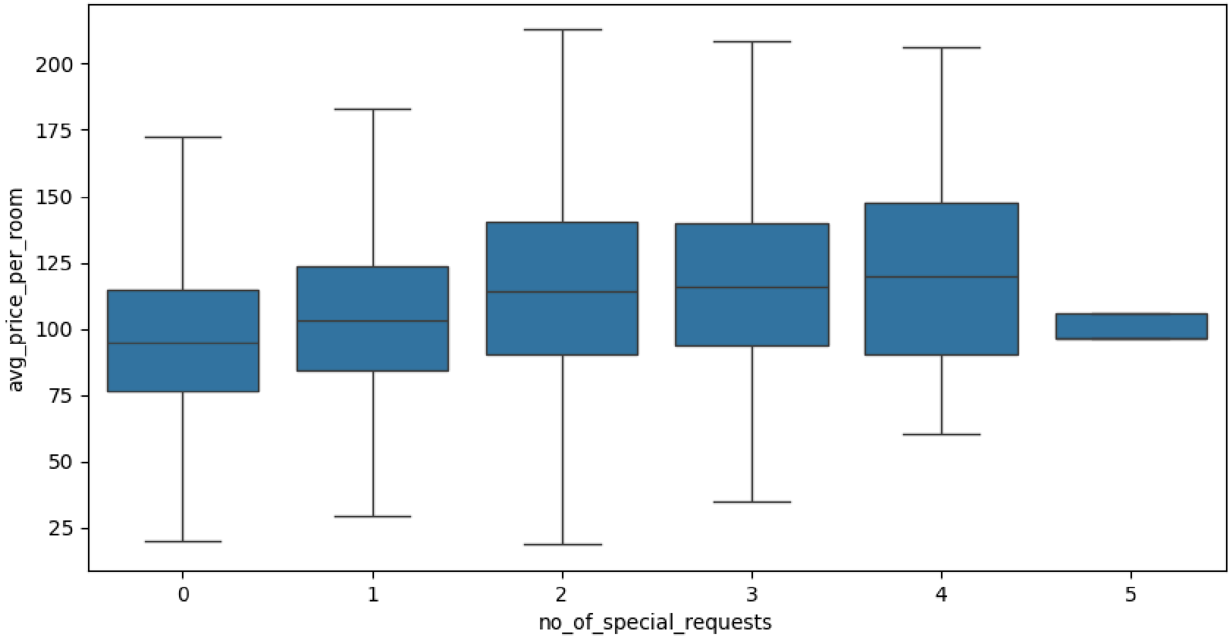


Fig 11: Observations on no_of_special_requests and avg_price_per_room

- The median prices of the rooms where the customers made some special requests are slightly higher than those where customers didn't.

We saw earlier that there is a positive correlation between booking status and average price per room. Let's analyze it

Observations on avg_price_room and booking_status

The company's first interaction with leads should be compelling and persuasive. Let's see if the channels of the first interaction have an impact on the conversion of leads

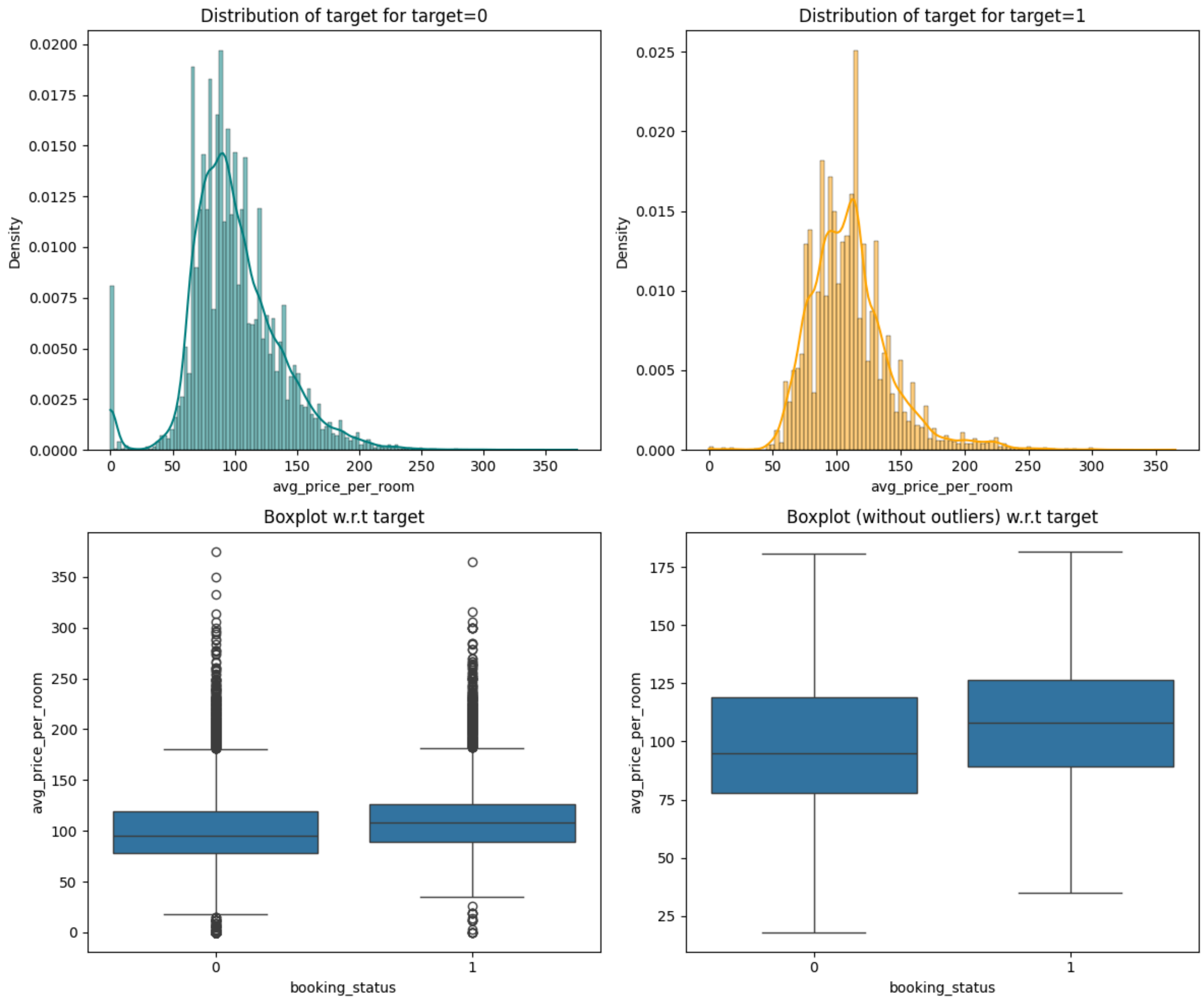


Fig 12: Observations on avg_price_room and booking_status

- The distribution of prices for canceled bookings and not canceled bookings is quite similar.
- The prices for the canceled bookings are slightly higher than the bookings that were not canceled.

Observations on market_segment_type and booking_status

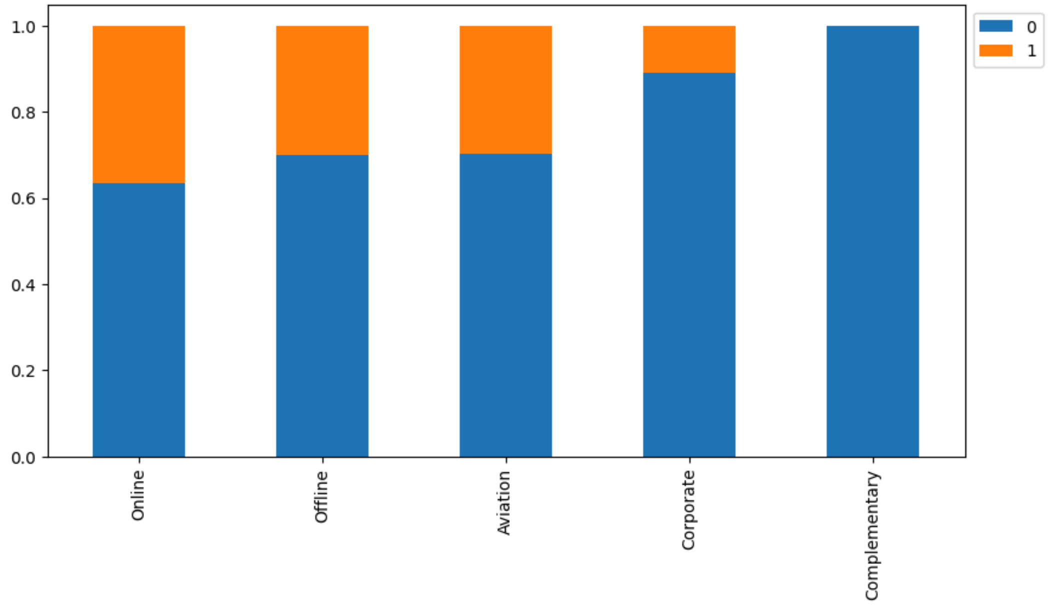


Fig 13: Observations on market_segment_type and booking_status

- Around 40% of the online bookings were canceled.
- Bookings made offline are less prone to cancellations.
- The corporate segment shows very low cancellations.

Observations on no_of_special_requests and booking_status

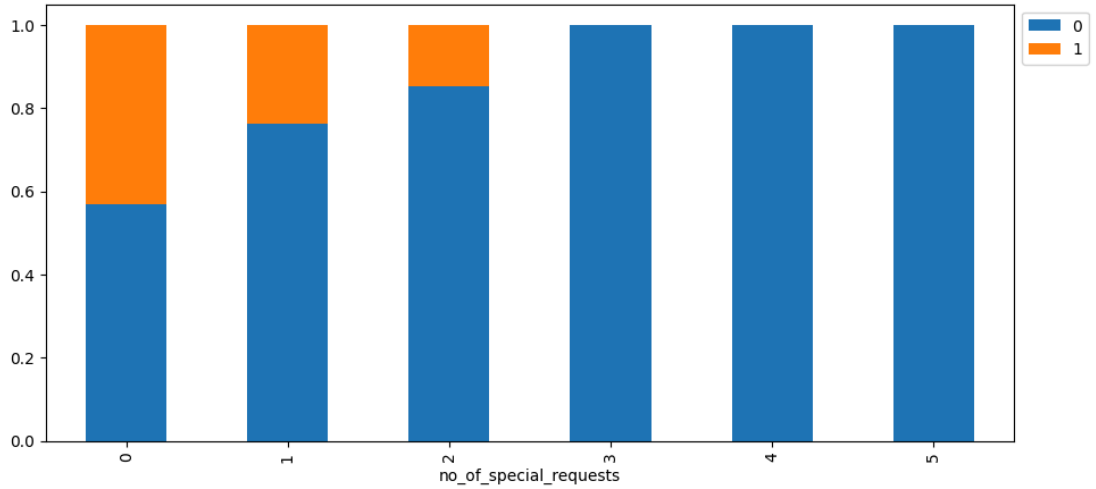


Fig 14: Observations on no_of_special_requests and booking_status

- If a customer has made more than 2 requests there's a very high chance that the booking will not be canceled.

Observations on lead_time and booking_status

There is a positive correlation between booking status and lead time. Let's analyze it further

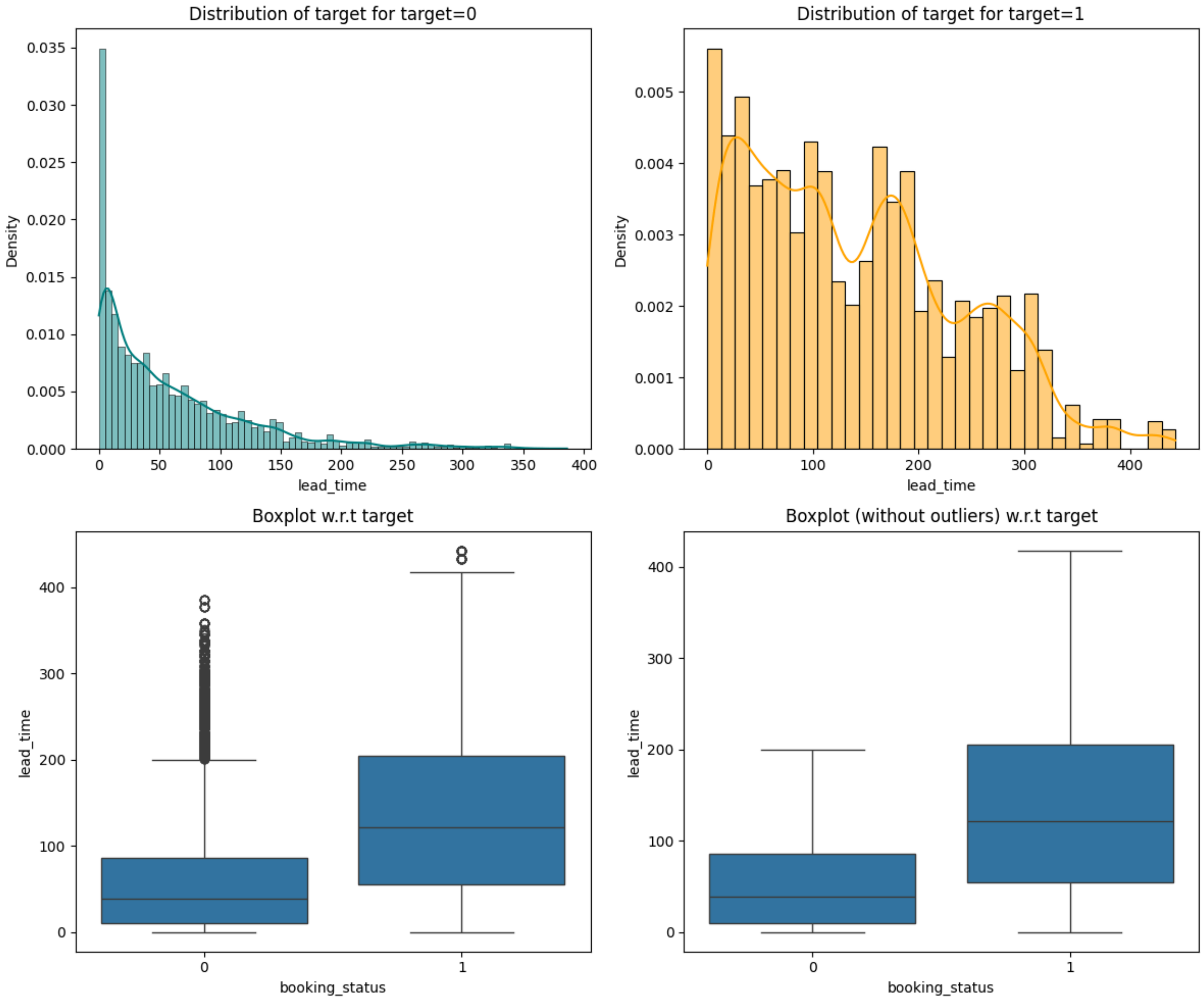


Fig 15: Observations on lead_time and booking_status

- There's a big difference in the median value of lead time for bookings that were canceled and bookings that were not canceled.
- The higher the lead time higher the chances of a booking being canceled.

Generally, people travel with their spouses and children for vacations or other activities.

Observations on no_of_family_members and booking_status

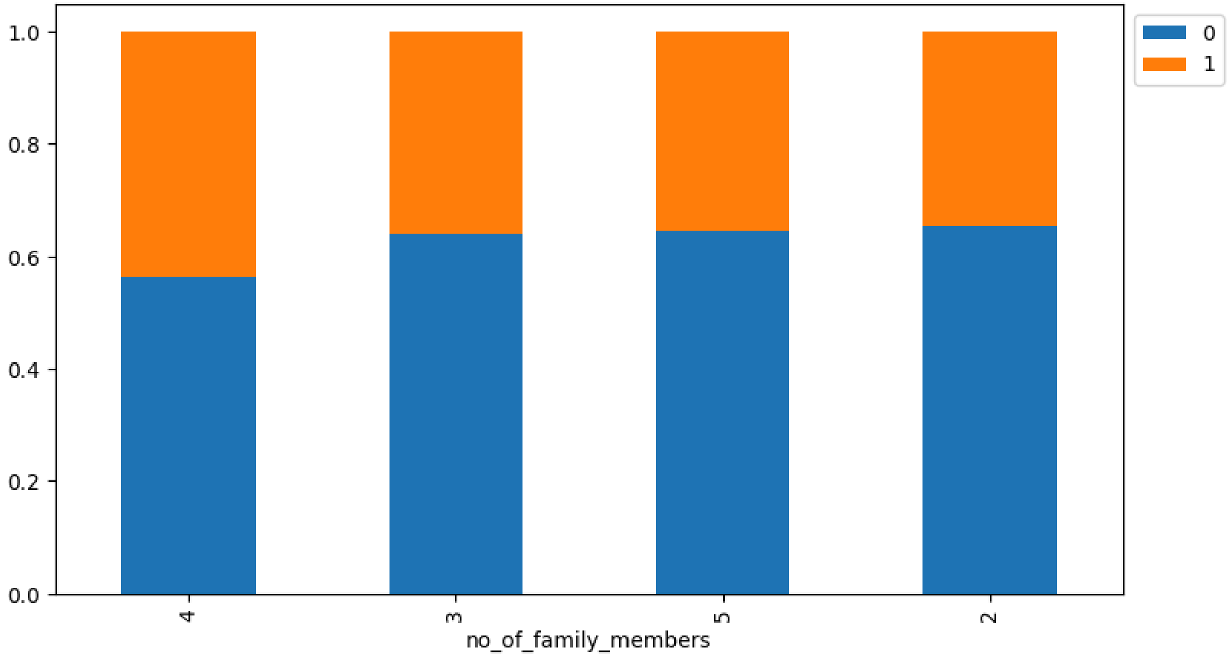


Fig 16: Observations on no_of_family_members and booking_status

- There's a ~40% chance of a booking getting canceled if the booking is made for 4 family members

Let's do a similar analysis for the customer who stays for at least a day at the hotel

Observations on total_days and booking_status

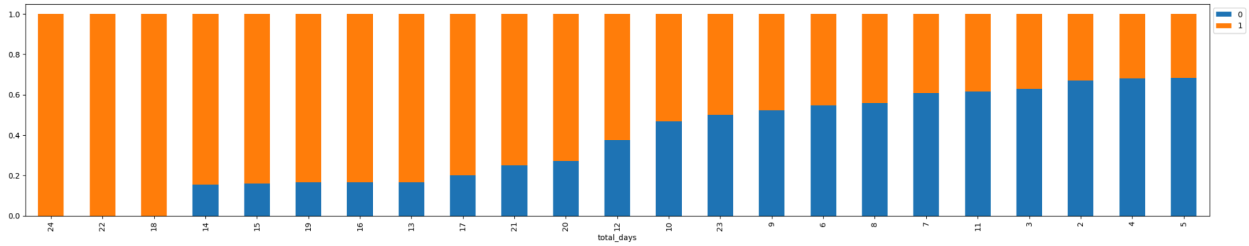


Fig 17: Observations on total_days and booking_status

- The general trend is that the chances of cancellation increase as the number of days the customer plans to stay at the hotel increases.

repeated_guests and booking_status

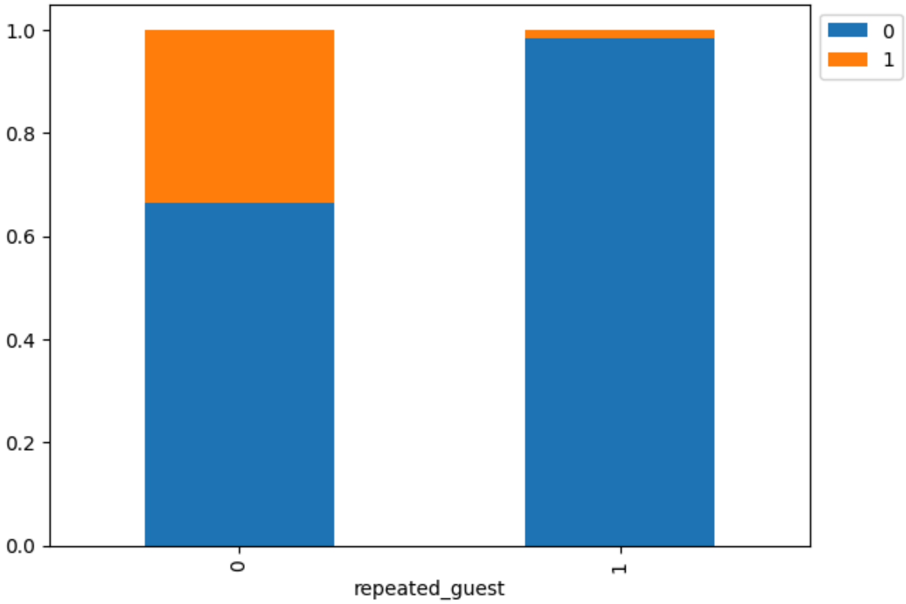


Fig 18: Observations on repeated_guests and booking_status

- There are very few repeat customers but the cancellations among them are very low.
- This is a good indication that repeat customers are important for the hospitality industry as they can help in spreading word of mouth.
- A loyal guest is usually more profitable for the business because they are more familiar with what is on offer at a hotel they have visited before.
- Attracting new customers is tedious and costs more as compared to a repeated guest.

Let's find out what are the busiest months in the hotel.

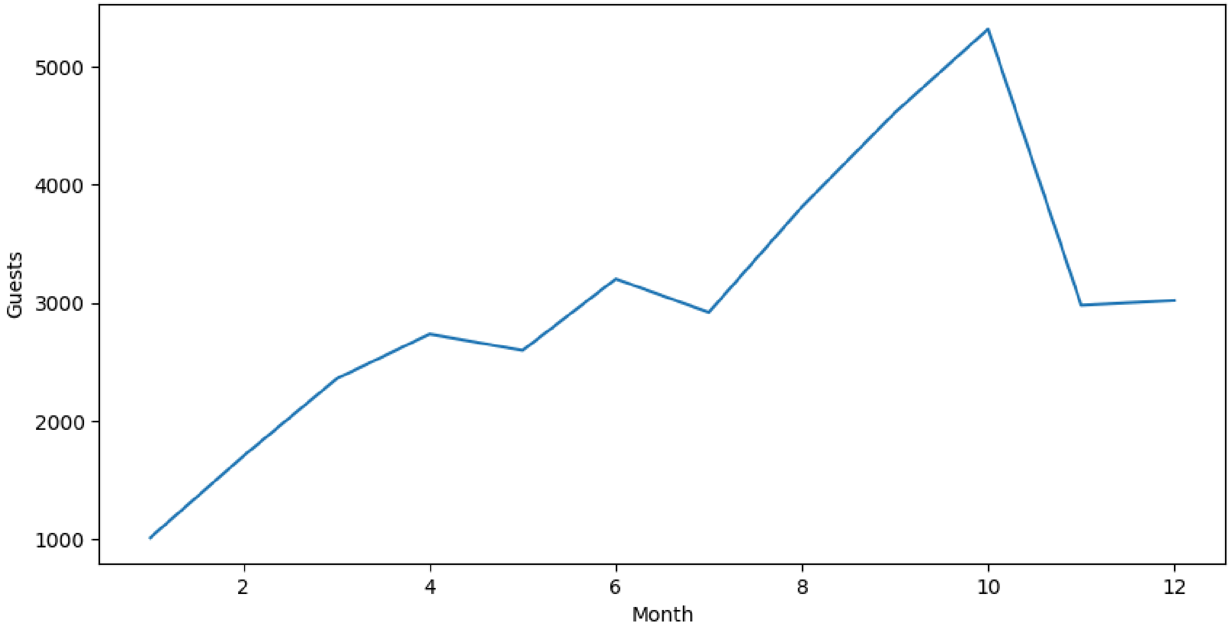


Fig 19: Observations on Month vs Number of Guests

- The trend shows the number of bookings remains consistent from April to July and the hotel sees around 3000 to 3500 guests.
- Most bookings were made in October - more than 5000 bookings.
- The last bookings were made in January - around 1000 bookings.

Let's check the percentage of bookings canceled each month

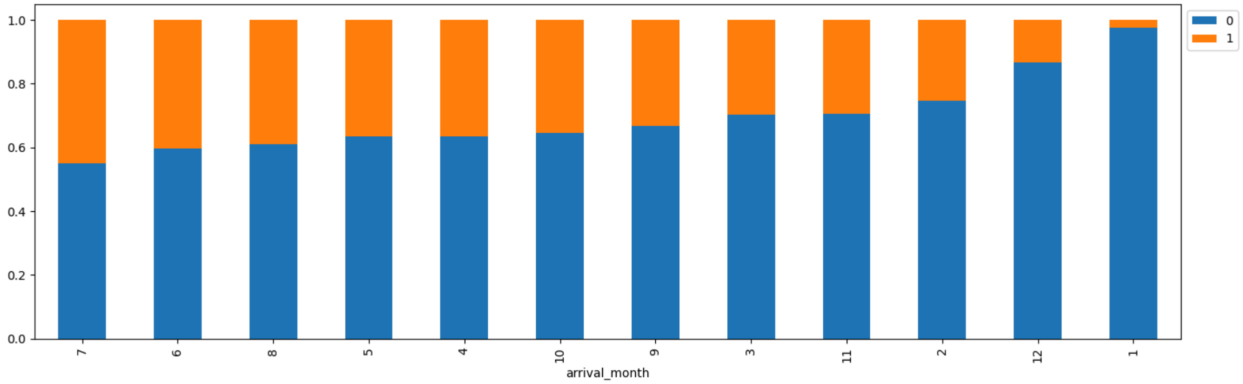


Fig 20: Observations on canceled bookings

- We see that even though the highest number of bookings were made in September and October - around 40% of these bookings got canceled.
- Least bookings were canceled in December and January - customers might have traveled to celebrate Christmas and New Year.

Correlation Analysis

Since many variables are continuous, the heatmap of the correlation matrix can give a very good idea of the correlations between the independent variables and the dependent variable.

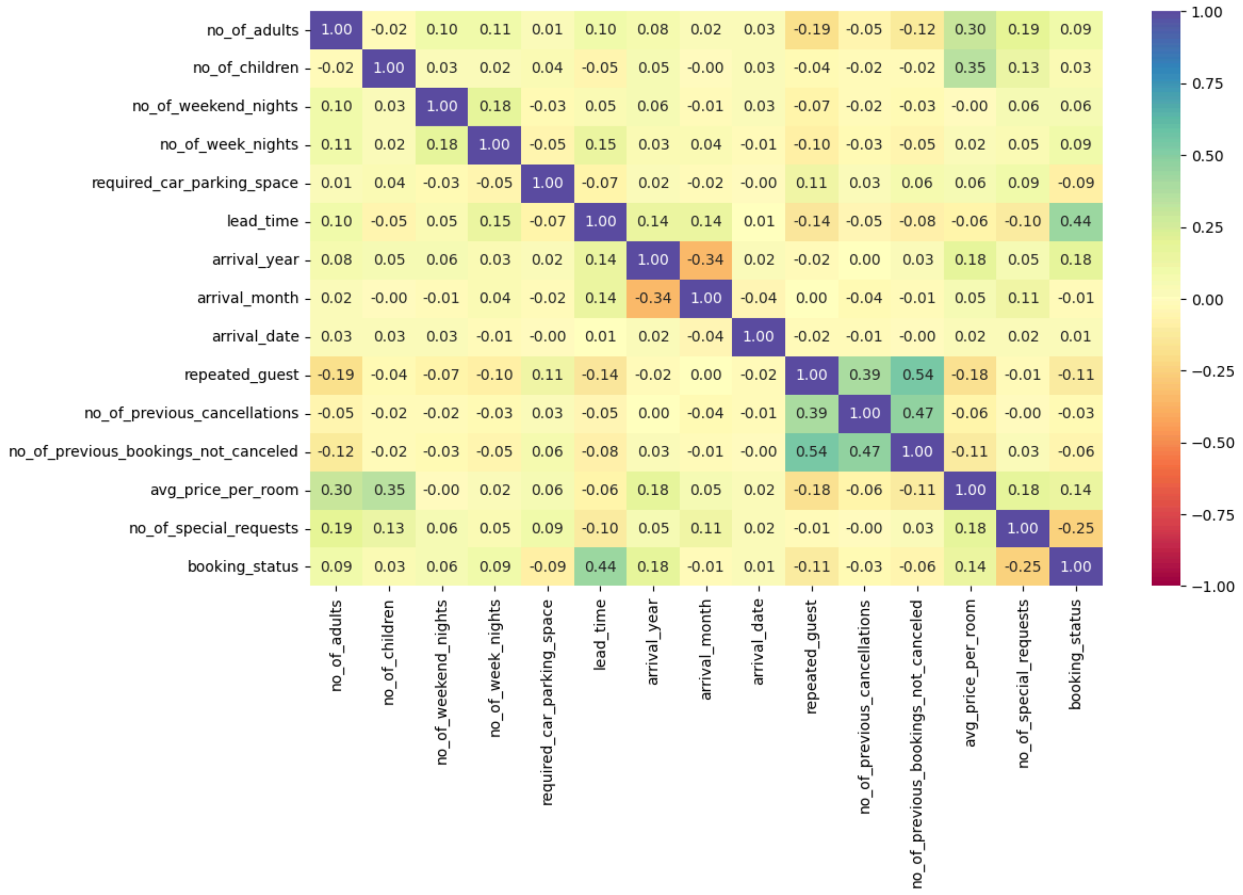


Fig 21: Correlation Heatmap

- There's a positive correlation between the number of customers (adults and children) and the average price per room.
- This makes sense as the more the number of customers more rooms they will require thus increasing the cost.

- There's a negative correlation between average room price and repeated guests. The hotel might be giving some loyalty benefits to the customers.
- There's a positive correlation between the number of previous bookings canceled and previous bookings not canceled by a customer and repeated guest.
- There's a positive correlation between lead time and the number of weeknights a customer is planning to stay in the hotel.
- There's a positive correlation between booking status and lead time, indicating higher the lead time higher the chances of cancellation. We will analyze it further.
- There's a negative correlation between the number of special requests from the customer and the booking status, indicating if a customer has made some special requests the chances of cancellation might decrease. We will analyze it further.

Data Preprocessing

Outliers Check

Outliers are values within a dataset that vary greatly from the others—they're either much larger or significantly smaller.

We can check the outliers present in the given dataset using the boxplots. It helps us identify data points that stand out from the rest of the data.

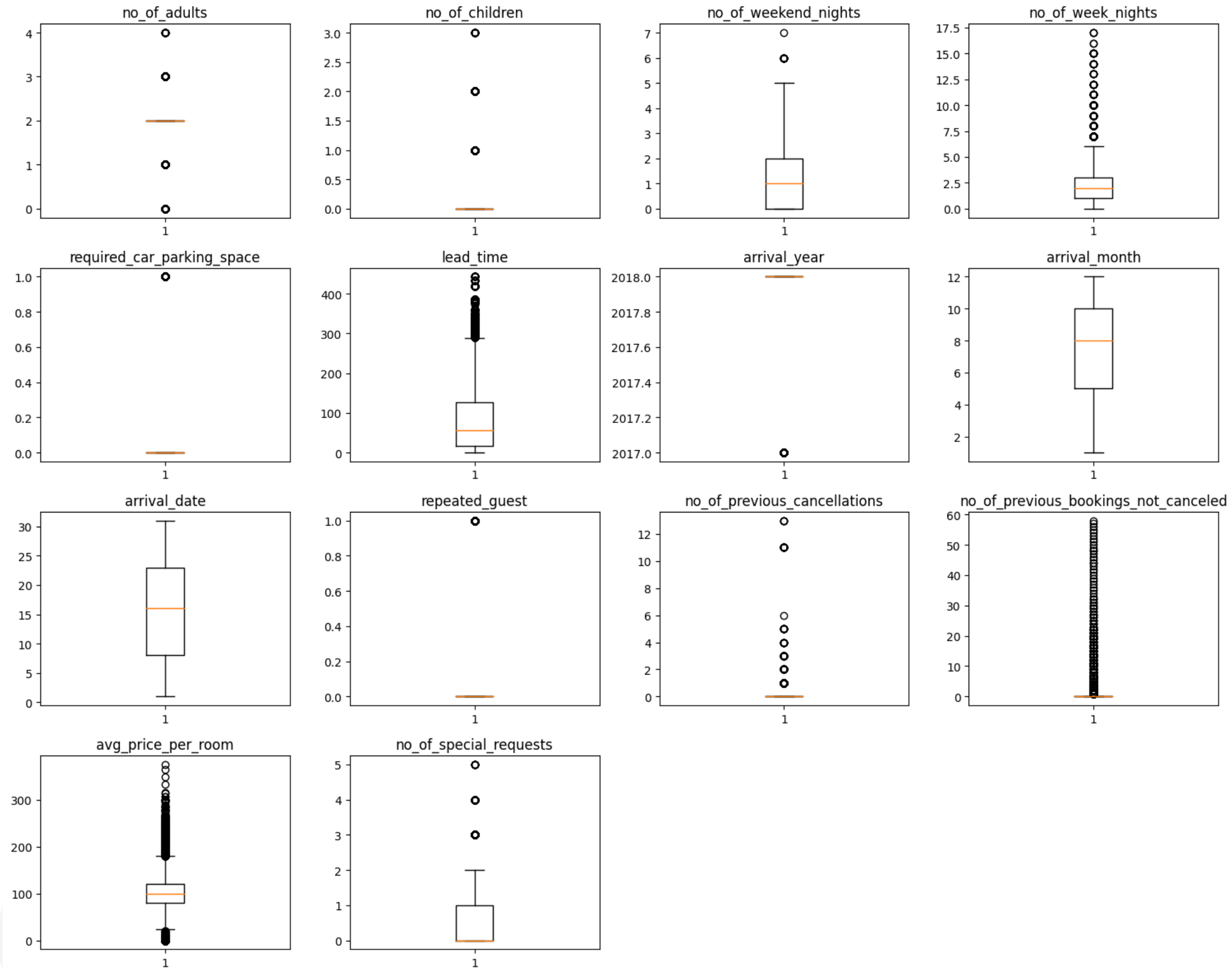


Fig 22: Outliers in the dataset

There are quite a few outliers present in the dataset. However, due to them being actual values, we will not treat the outliers and leave them in the dataset.

Data Preparation for Modeling

- We want to predict which bookings will be canceled.

- 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.

Model Building

Model Evaluation Criterion

The model can make wrong predictions as:

1. Predicting a customer will not cancel their booking but in reality, the customer will cancel their booking.

2. Predicting a customer will cancel their booking but in reality, the customer will not cancel their booking.

Which case is more important?

Both cases are important as:

- If we predict that a booking will not be canceled and the booking gets canceled then the hotel will lose resources and will have to bear additional costs of distribution channels.
- If we predict that a booking will get canceled and the booking doesn't get canceled the hotel might not be able to provide satisfactory services to the customer by assuming that this booking will be canceled. This might damage the brand equity.

How to reduce the losses?

The hotel chain would want the `F1 Score` to be maximized, the greater the F1 score higher the chances of minimizing False Negatives and False Positives.

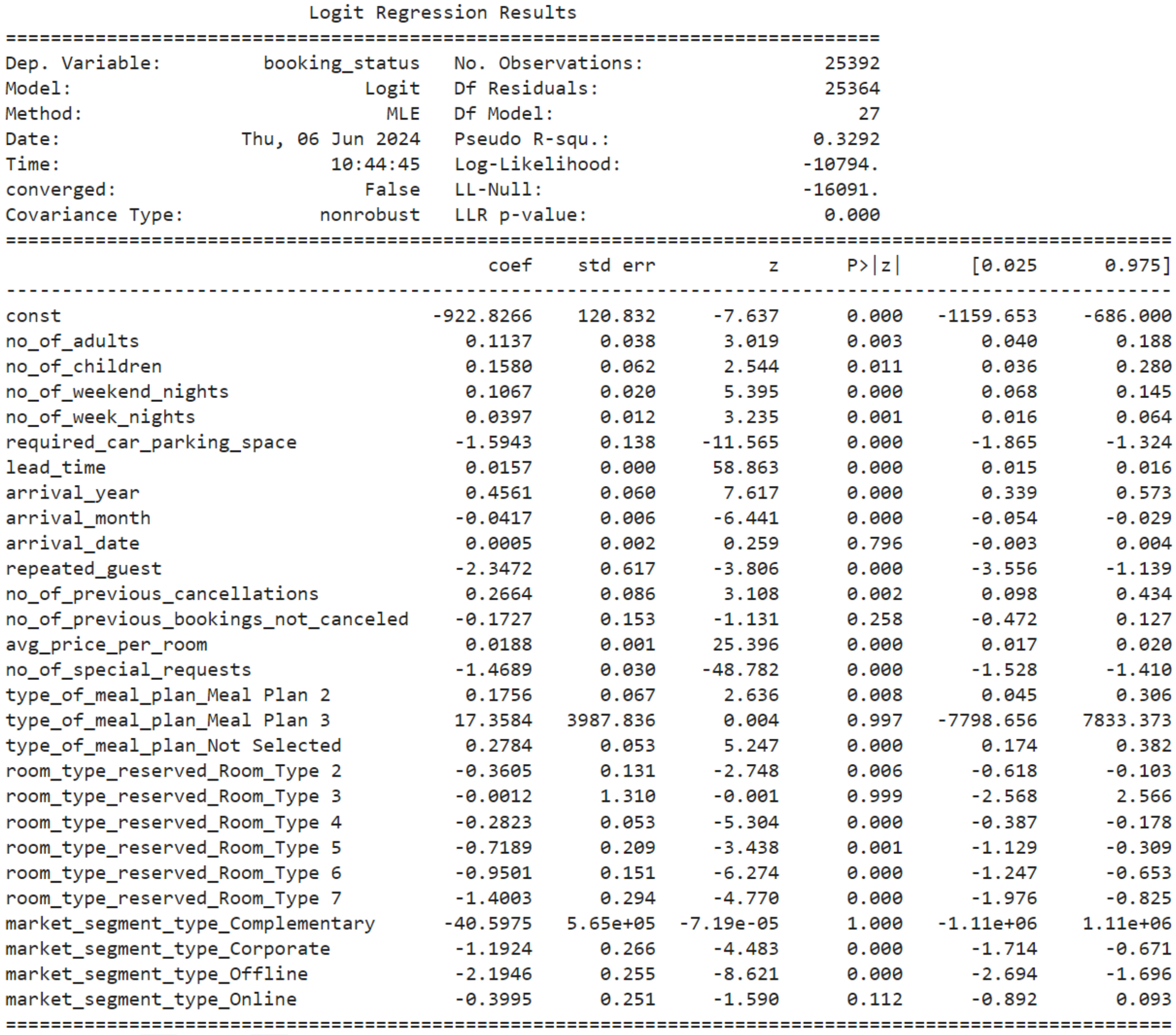
Logistic Regression

A constant term is added to the independent variable matrix to account for the intercept in the linear regression model.

The logistic regression (logit function) model helps predict whether the booking will be canceled or not because it can model the nonlinear relationship between predictors and the probability of conversion.

We use the Statsmodel Library for building the logistic regression model. The statsmodels library for logistic regression offers detailed statistical inference and diagnostics, aiding in hypothesis testing and interpretation, while sklearn focuses more on predictive modeling without the same level of statistical analysis.

Here is what the model summary looks like after the model is built:



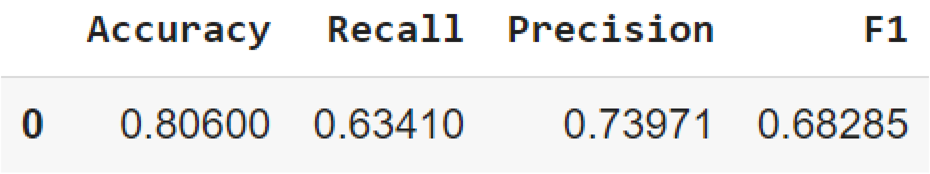
```
                           Logit Regression Results
==============================================================================
Dep. Variable:         booking_status   No. Observations:                25392
Model:                          Logit   Df Residuals:                    25364
Method:                           MLE   Df Model:                           27
Date:                Thu, 06 Jun 2024   Pseudo R-squ.:                  0.3292
Time:                        10:44:45   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                                 -922.8266    120.832     -7.637      0.000    -1159.653    -686.000
no_of_adults                             0.1137      0.038      3.019      0.003        0.040       0.188
no_of_children                           0.1580      0.062      2.544      0.011        0.036       0.280
no_of_weekend_nights                     0.1067      0.020      5.395      0.000        0.068       0.145
no_of_week_nights                        0.0397      0.012      3.235      0.001        0.016       0.064
required_car_parking_space              -1.5943      0.138    -11.565      0.000       -1.865      -1.324
lead_time                                0.0157      0.000     58.863      0.000        0.015       0.016
arrival_year                             0.4561      0.060      7.617      0.000        0.339       0.573
arrival_month                           -0.0417      0.006     -6.441      0.000       -0.054      -0.029
arrival_date                             0.0005      0.002      0.259      0.796       -0.003       0.004
repeated_guest                          -2.3472      0.617     -3.806      0.000       -3.556      -1.139
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     25.396      0.000        0.017       0.020
no_of_special_requests                  -1.4689      0.030    -48.782      0.000       -1.528      -1.410
type_of_meal_plan_Meal Plan 2            0.1756      0.067      2.636      0.008        0.045       0.306
type_of_meal_plan_Meal Plan 3           17.3584   3987.836      0.004      0.997    -7798.656    7833.373
type_of_meal_plan_Not Selected           0.2784      0.053      5.247      0.000        0.174       0.382
room_type_reserved_Room_Type 2          -0.3605      0.131     -2.748      0.006       -0.618      -0.103
room_type_reserved_Room_Type 3          -0.0012      1.310     -0.001      0.999       -2.568       2.566
room_type_reserved_Room_Type 4          -0.2823      0.053     -5.304      0.000       -0.387      -0.178
room_type_reserved_Room_Type 5          -0.7189      0.209     -3.438      0.001       -1.129      -0.309
room_type_reserved_Room_Type 6          -0.9501      0.151     -6.274      0.000       -1.247      -0.653
room_type_reserved_Room_Type 7          -1.4003      0.294     -4.770      0.000       -1.976      -0.825
market_segment_type_Complementary      -40.5975   5.65e+05  -7.19e-05      1.000    -1.11e+06    1.11e+06
market_segment_type_Corporate           -1.1924      0.266     -4.483      0.000       -1.714      -0.671
market_segment_type_Offline             -2.1946      0.255     -8.621      0.000       -2.694      -1.696
market_segment_type_Online              -0.3995      0.251     -1.590      0.112       -0.892       0.093
========================================================================================================
```

Table 6: Logistic Regression - Model Summary

Logistic Regression Model - Training Performance

We need to evaluate the model's performance on training and test data to see how the logistic regression model performs. We will plot the confusion matrix and analyze performance metrics like accuracy, precision, recall, and f1-score to evaluate the model's performance.

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.80600	0.63410	0.73971	0.68285

Table 7: Performance Metrics for Logistic regression Training data

Confusion matrix:

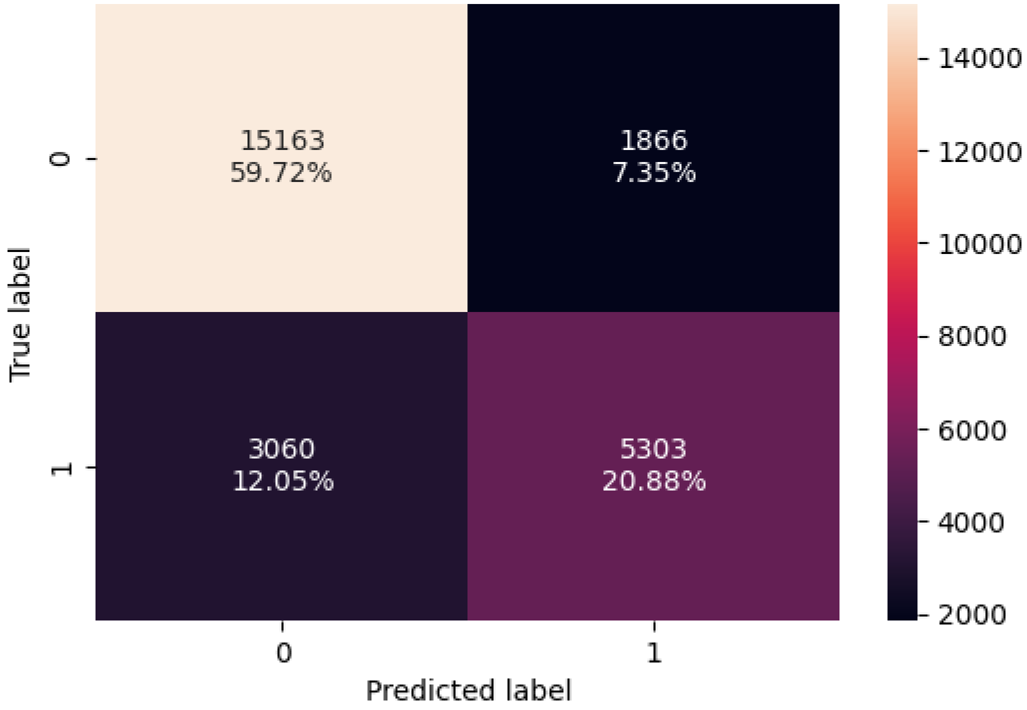
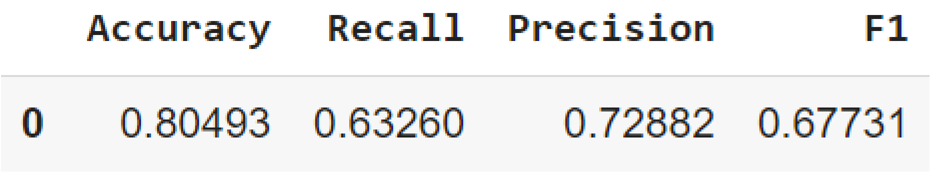


Fig 23: Confusion matrix for Logistic regression Training data

Logistic Regression Model - Test Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.80493	0.63260	0.72882	0.67731

Table 8: Performance Metrics for Logistic regression Test data

Confusion matrix:

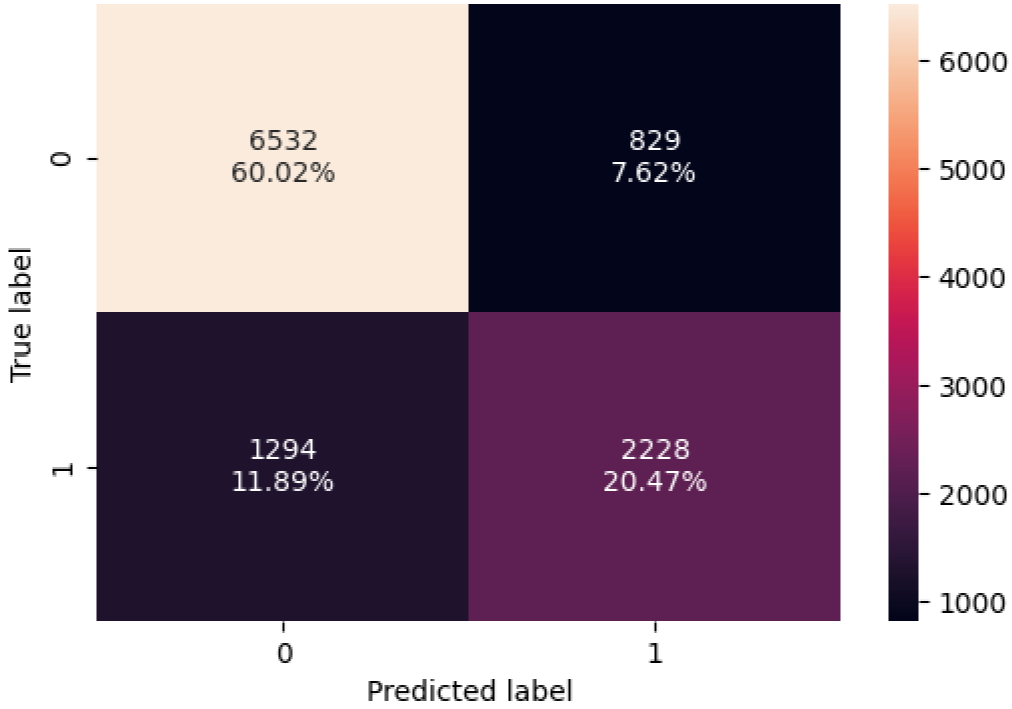


Fig 24: Confusion matrix for Logistic regression Test data

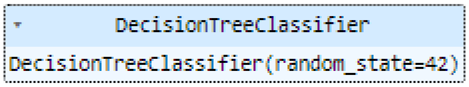
Logistic Regression Model - Performance Observations

We can observe that this model is not performing well using the above performance metrics and confusion matrix. The F1 score for both training and test data is average. So, we will try building other models on the same data.

Decision Tree Classifier

Decision Trees are renowned for their interpretability, making them valuable for understanding the factors influencing lead conversion. The tree-like structure provides a clear visual representation of the decision-making process, enabling easy identification of important features. This model is versatile and can handle numerical and categorical data without extensive preprocessing. Furthermore, decision trees can naturally capture nonlinear relationships between features and the target variable and since it is not dependent on the scale of the individual attributes, we don't need to do feature scaling for building decision trees.

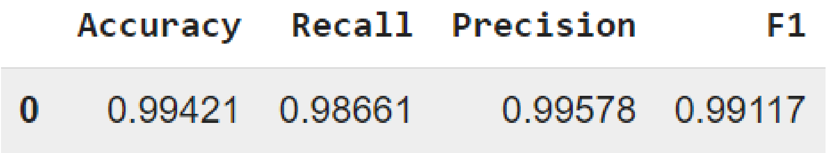
By leveraging the strengths of these algorithms, businesses can effectively predict lead conversion and optimize their sales and marketing efforts.



```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

Decision Tree Classifier - Training Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.99421	0.98661	0.99578	0.99117

Table 9: Performance Metrics for Decision Tree Classifier Training data

Confusion Matrix:

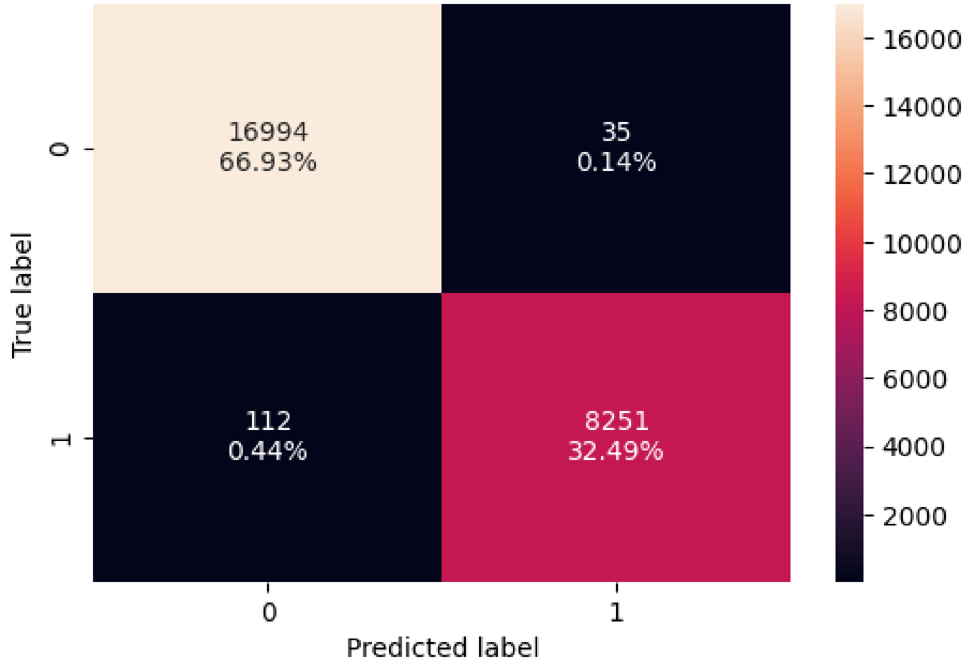
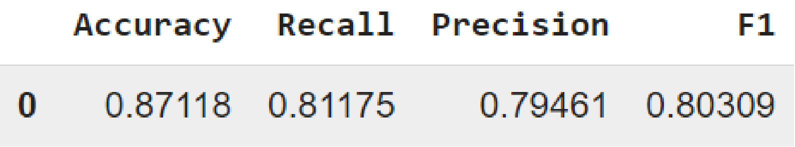


Fig 25: Confusion matrix for Decision Tree Classifier Training data

Decision Tree Classifier - Test Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.87118	0.81175	0.79461	0.80309

Table 10: Performance Metrics for Decision Tree Classifier Test data

Confusion Matrix:

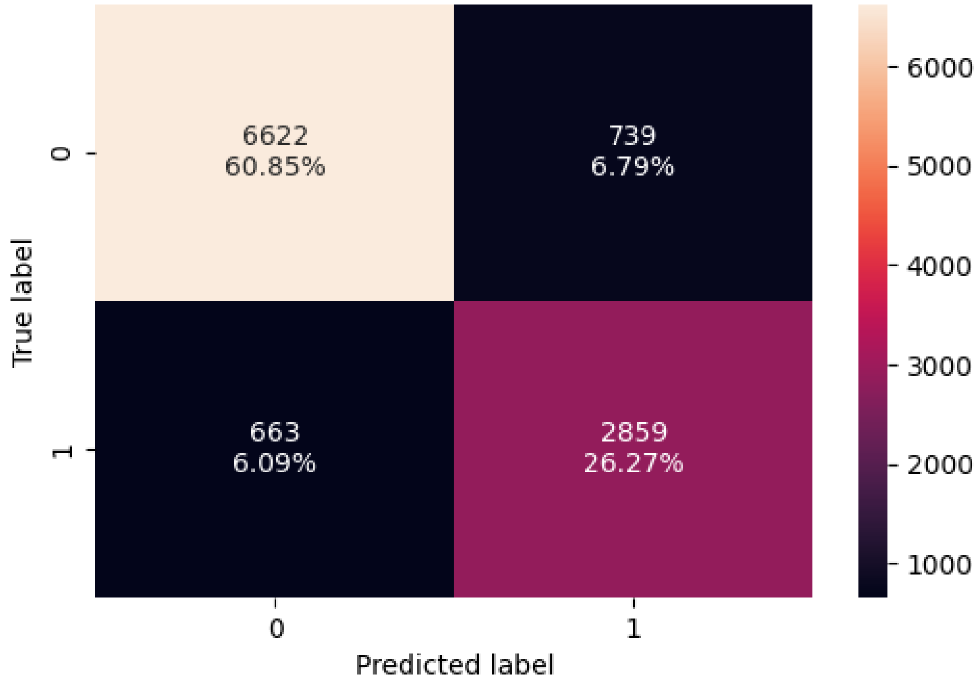


Fig 26: Confusion matrix for Decision Tree Classifier Test data

Decision Tree - Performance Observations

- The model has performed very well 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.
- The decision tree model overfits the data as expected and cannot be generalized well on the test set.

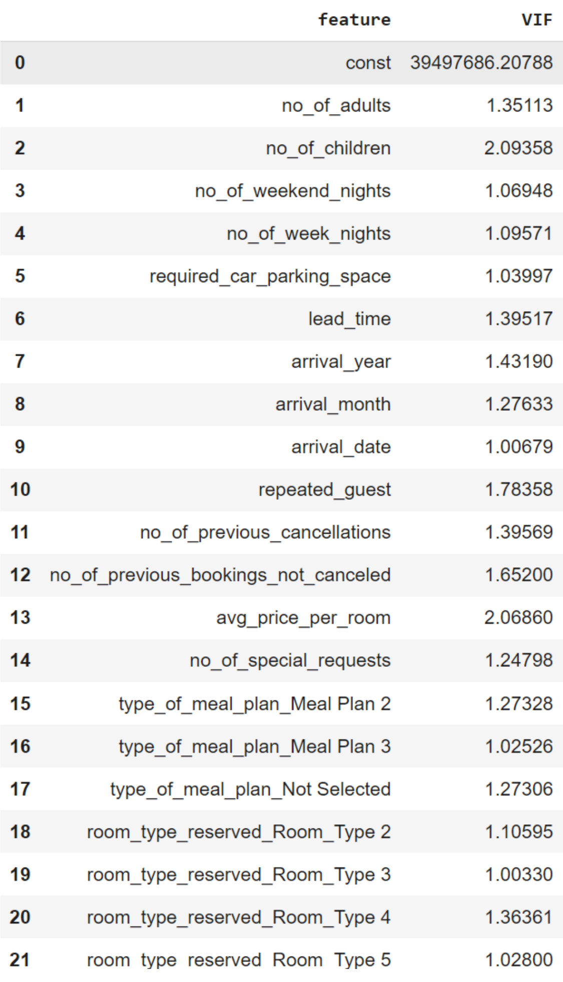
Therefore, we need to tune all of these models to enhance their performance and then compare again to figure out which model best fits our use case.

Model Performance Improvement

Logistic Regression (dealing with multicollinearity, removing high p-value variables, determining optimal threshold using ROC curve)

The first step is to address the problem of multicollinearity. This occurs when the independent variables are highly correlated to each other. Variance Inflation Factor (VIF) measures the extent to which the variance of the estimated regression coefficients is increased due to multicollinearity.

The VIF criterion will be used to exclude variables with high multicollinearity. If VIF > 5, then exclude these variables from the model. The VIFs of the different variables in the dataset are as follows:



	feature	VIF
0	const	39497686.20788
1	no_of_adults	1.35113
2	no_of_children	2.09358
3	no_of_weekend_nights	1.06948
4	no_of_week_nights	1.09571
5	required_car_parking_space	1.03997
6	lead_time	1.39517
7	arrival_year	1.43190
8	arrival_month	1.27633
9	arrival_date	1.00679
10	repeated_guest	1.78358
11	no_of_previous_cancellations	1.39569
12	no_of_previous_bookings_not_canceled	1.65200
13	avg_price_per_room	2.06860
14	no_of_special_requests	1.24798
15	type_of_meal_plan_Meal Plan 2	1.27328
16	type_of_meal_plan_Meal Plan 3	1.02526
17	type_of_meal_plan_Not Selected	1.27306
18	room_type_reserved_Room_Type 2	1.10595
19	room_type_reserved_Room_Type 3	1.00330
20	room_type_reserved_Room_Type 4	1.36361
21	room_type_reserved_Room_Type 5	1.02800

Table 11: VIF of the variables

Since no variable has VIF > 5, we will keep the dataset as it is.

The next step in the process is to identify variables that have high p-values (> 0.05).

P-values tell us how likely it is that a feature doesn't matter. If the p-value is too high (usually above 0.05), it means the feature probably isn't important, so we can remove it to simplify our model.

Think of a p-value as a "doubt score". A low p-value (less than 0.05, commonly used) means it's unlikely the result happened by chance. In other words, this feature likely has a real impact on what you're trying to predict (lead conversions, in our example). So we keep this feature in our model.

On the other hand, a high p-value (above 0.05) means there's a good chance the result is just random noise. This feature probably doesn't have a strong influence on what we're predicting. So we can drop this feature to keep your model simpler and more focused on the important stuff.

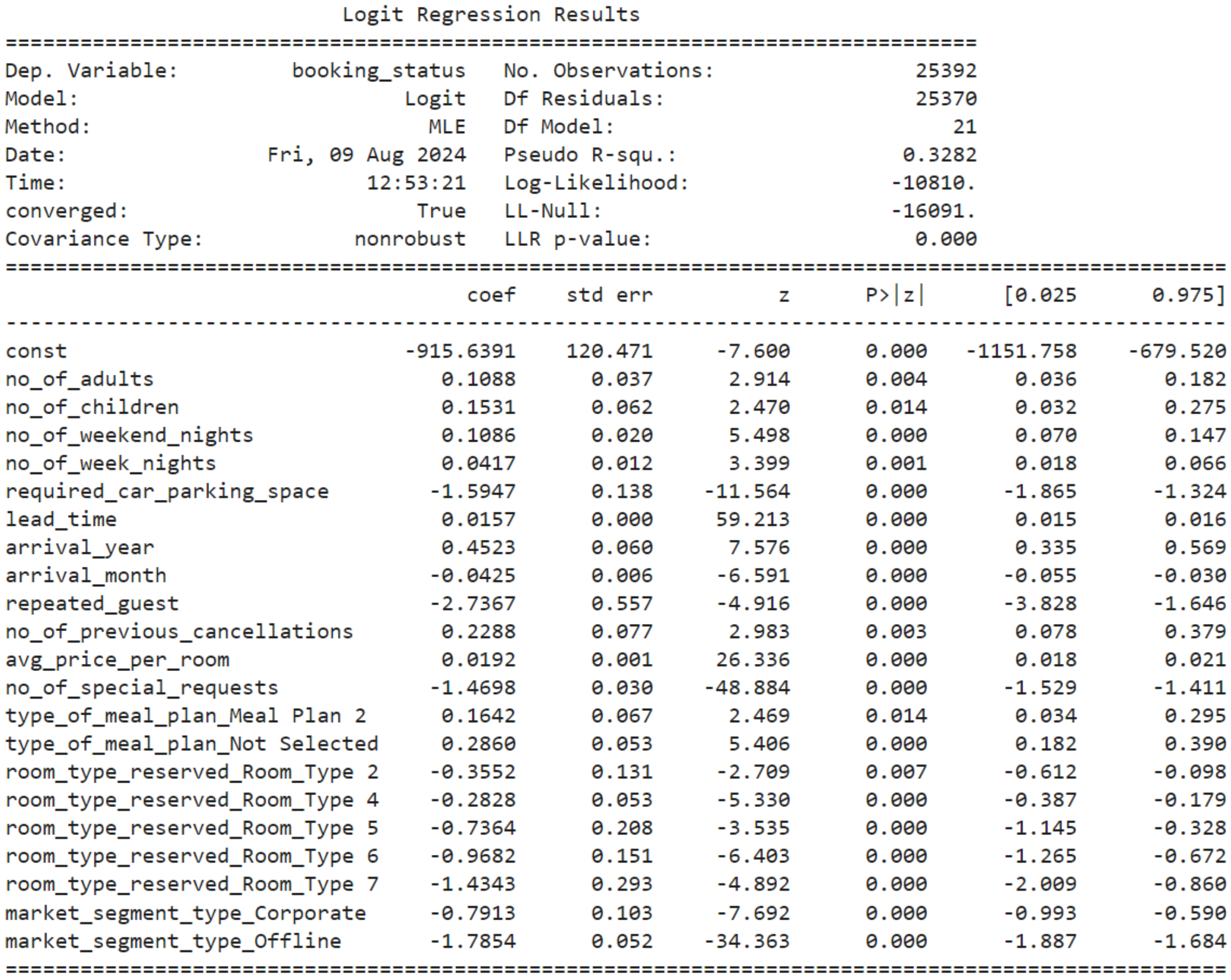
By combining VIF and p-value analysis, we can effectively identify and remove variables that are either highly correlated with other predictors or have minimal impact on the target variable, leading to a more robust and predictive model.

After dropping the attributes with high p-values, we are left with the following attributes:

- 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
- repeated_guest
- no_of_previous_cancellations
- avg_price_per_room
- no_of_special_requests
- type_of_meal_plan_Meal Plan 2
- type_of_meal_plan_Not Selected
- room_type_reserved_Room_Type 2

- room_type_reserved_Room_Type 4
- room_type_reserved_Room_Type 5
- room_type_reserved_Room_Type 6
- room_type_reserved_Room_Type 7
- market_segment_type_Corporate
- market_segment_type_Offline

After removing the columns with high p-values, we are left with 22 columns. The next step is to build the logistic regression model with the remaining variables. After fitting the logistic regression model to the new data, the summary of the model is given below:



```
                           Logit Regression Results                           
==============================================================================
Dep. Variable:         booking_status   No. Observations:                25392
Model:                          Logit   Df Residuals:                    25370
Method:                           MLE   Df Model:                           21
Date:                Fri, 09 Aug 2024   Pseudo R-squ.:                  0.3282
Time:                        12:53:21   Log-Likelihood:                -10810.
converged:                       True   LL-Null:                       -16091.
Covariance Type:            nonrobust   LLR p-value:                     0.000
====================================================================================================
                                       coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------------
const                             -915.6391    120.471     -7.600      0.000   -1151.758    -679.520
no_of_adults                         0.1088      0.037      2.914      0.004       0.036       0.182
no_of_children                       0.1531      0.062      2.470      0.014       0.032       0.275
no_of_weekend_nights                 0.1086      0.020      5.498      0.000       0.070       0.147
no_of_week_nights                    0.0417      0.012      3.399      0.001       0.018       0.066
required_car_parking_space          -1.5947      0.138    -11.564      0.000      -1.865      -1.324
lead_time                            0.0157      0.000     59.213      0.000       0.015       0.016
arrival_year                         0.4523      0.060      7.576      0.000       0.335       0.569
arrival_month                       -0.0425      0.006     -6.591      0.000      -0.055      -0.030
repeated_guest                      -2.7367      0.557     -4.916      0.000      -3.828      -1.646
no_of_previous_cancellations         0.2288      0.077      2.983      0.003       0.078       0.379
avg_price_per_room                   0.0192      0.001     26.336      0.000       0.018       0.021
no_of_special_requests              -1.4698      0.030    -48.884      0.000      -1.529      -1.411
type_of_meal_plan_Meal Plan 2        0.1642      0.067      2.469      0.014       0.034       0.295
type_of_meal_plan_Not Selected       0.2860      0.053      5.406      0.000       0.182       0.390
room_type_reserved_Room_Type 2      -0.3552      0.131     -2.709      0.007      -0.612      -0.098
room_type_reserved_Room_Type 4      -0.2828      0.053     -5.330      0.000      -0.387      -0.179
room_type_reserved_Room_Type 5      -0.7364      0.208     -3.535      0.000      -1.145      -0.328
room_type_reserved_Room_Type 6      -0.9682      0.151     -6.403      0.000      -1.265      -0.672
room_type_reserved_Room_Type 7      -1.4343      0.293     -4.892      0.000      -2.009      -0.860
market_segment_type_Corporate       -0.7913      0.103     -7.692      0.000      -0.993      -0.590
market_segment_type_Offline         -1.7854      0.052    -34.363      0.000      -1.887      -1.684
====================================================================================================
```

Table 12: Logistic regression - Improved Model summary

After building the new logistic regression model with significant features, we have to find the optimal threshold value for the improved model.

- The threshold value in Logistic Regression determines the point at which predicted probabilities are classified into different classes. Finding the optimal threshold is useful as it allows for adjusting the trade-off between precision and recall, ensuring the model's predictions align with the specific needs of the problem at hand, such as minimizing false negatives to identify potential leads who will convert.

- Using the ROC - AUC curve to determine the value of the optimal threshold, this value comes out to be ~ 0.14. This means that if for any customer, the predicted probability value for the status variable is greater than 0.14, it will be classified as canceled (1). On the other hand, if the predicted probability is less than 0.14, the booking will be classified as not canceled (0).

These findings can be justified by plotting the ROC - AUC curve for the data:

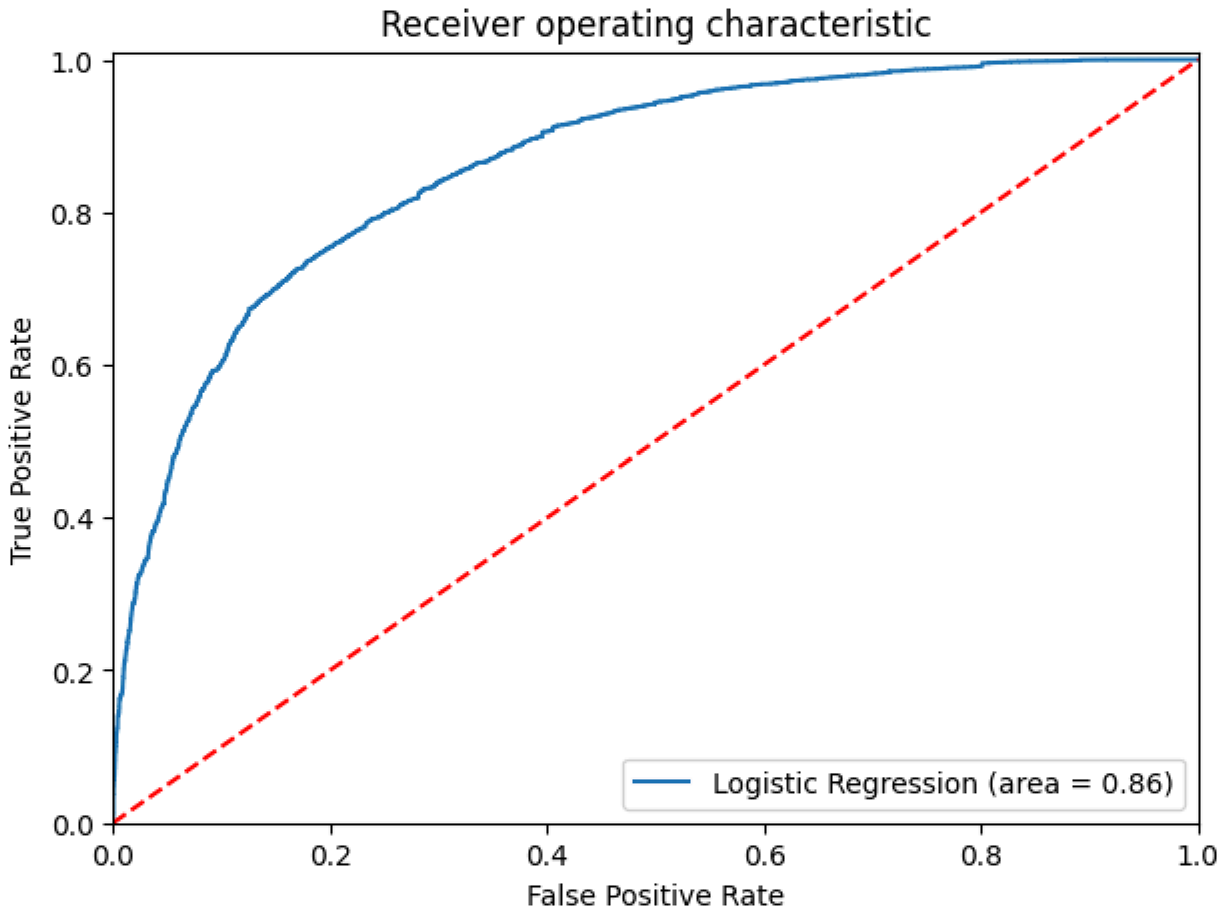
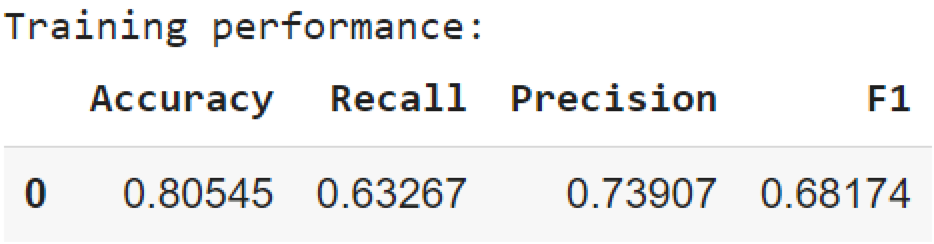


Fig 27: ROC - AUC Curve

Logistic Regression Model - Training Performance:

After making certain adjustments to the logistic regression model, like dropping columns with high p-values from the dataset and finding the optimal threshold value, we will check the model's performance again on the training and test sets to see if there is any improvement in its performance.

Performance Metrics:



Training performance:

	Accuracy	Recall	Precision	F1
0	0.80545	0.63267	0.73907	0.68174

Table 13: Performance Metrics for Logistic Regression (Improved) - Training data

Confusion matrix:

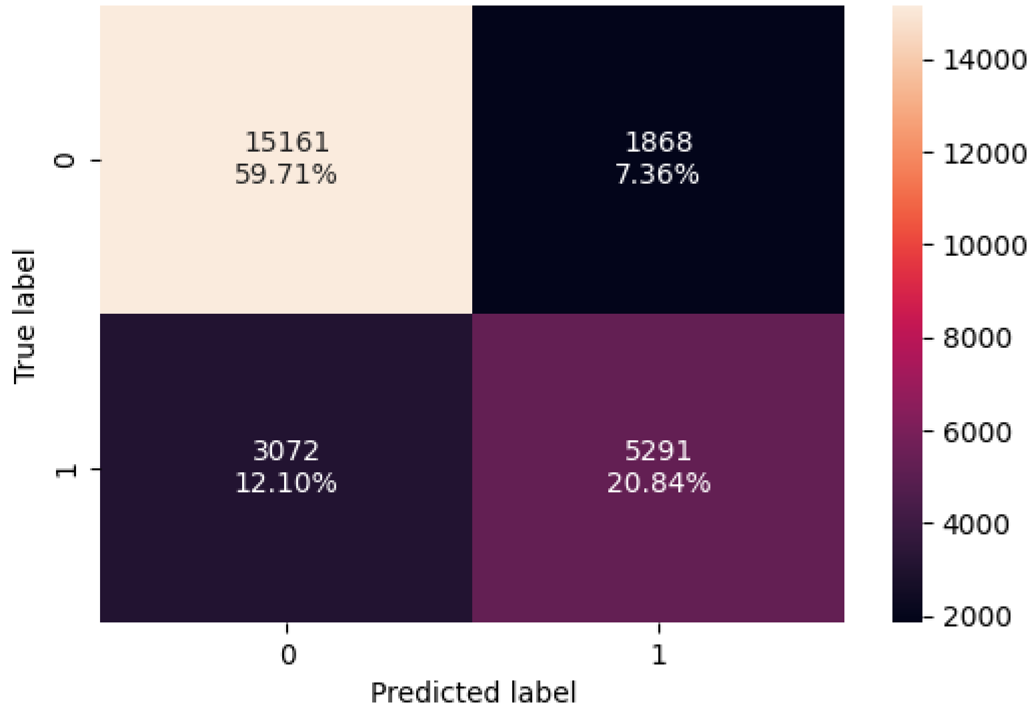
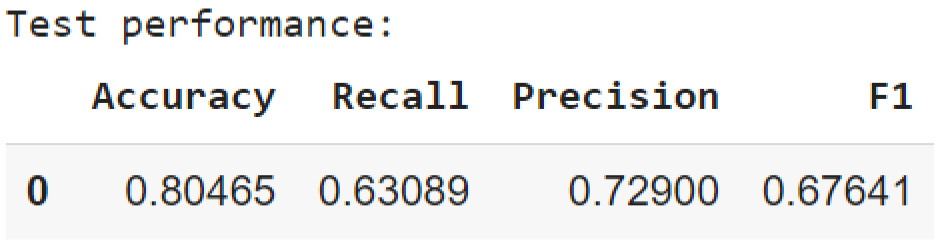


Fig 28: Confusion matrix for Logistic Regression (Improved) - Training data

Logistic Regression Model - Test Performance:

Performance Metrics:



Test performance:

	Accuracy	Recall	Precision	F1
0	0.80465	0.63089	0.72900	0.67641

Table 14: Performance Metrics for Logistic Regression (Improved) - Test data

Confusion matrix:

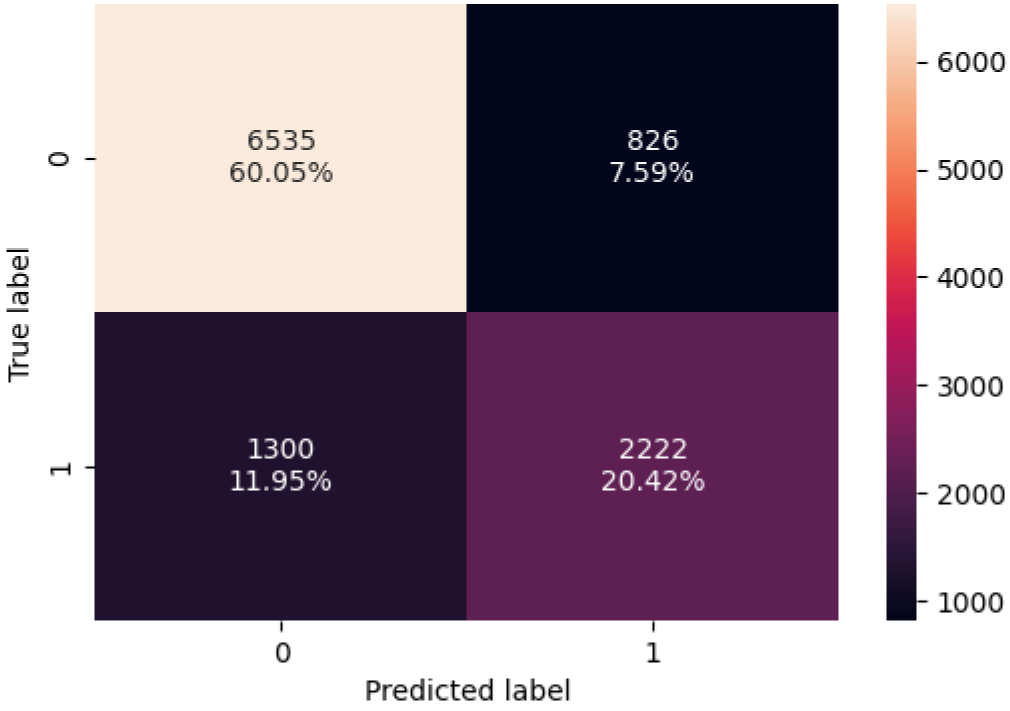


Fig 29: Confusion matrix for Logistic Regression (Improved) - Test data

Logistic Regression Model - Performance Observations:

Analyzing the performance results of the improved logistic regression model above, we can draw the following conclusions.

- The model gives a generalized performance on training and test sets.
- ROC-AUC score of 0.86 on training is quite good.

Decision Tree Classifier (pre-pruning)

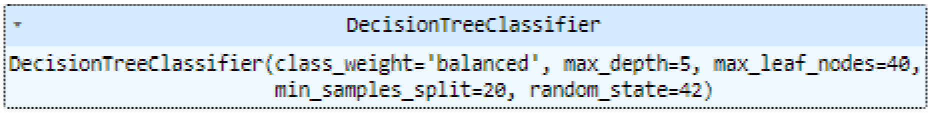
We will employ pre-pruning techniques to improve the performance of the Decision Tree model. Pre-pruning stops a decision tree from growing too deep too soon. This prevents overfitting, where the model becomes too specific to the training data and performs poorly on new data. By limiting the tree's size, pre-pruning helps the model generalize better and make more accurate predictions on unseen data.

We will be using Hyperparameter tuning with GridSearchCV to improve the performance of our Decision Tree model.

Hyperparameter tuning with GridSearchCV involves systematically searching through a specified grid of hyperparameter values for a machine learning algorithm. It evaluates each combination using cross-validation and selects the one that maximizes a specified performance metric, thus optimizing the model's effectiveness and generalization.

We will check different combinations of values for several parameters like max_depth, min_samples_split, max_leaf_nodes, and class_weight for the DecisionTreeClassifier function from sklearn to see which combination performs the best for the given data.

After performing GridSearchCV for Hyperparameter tuning, the best combination of parameters for the DecisionTreeClassifier comes out to be:

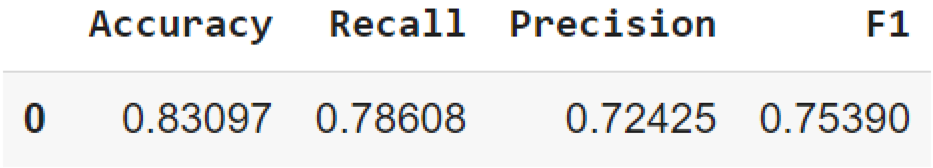


```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', max_depth=5, max_leaf_nodes=40,
                       min_samples_split=20, random_state=42)
```

Now, we select the best model that uses these parameters and our next step is to evaluate the performance of the improved Decision Tree model on the training and test sets.

Decision Tree Classifier - Training Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.83097	0.78608	0.72425	0.75390

Table 15: Performance Metrics for Decision Tree Classifier (Improved) - Training data

Confusion Matrix:

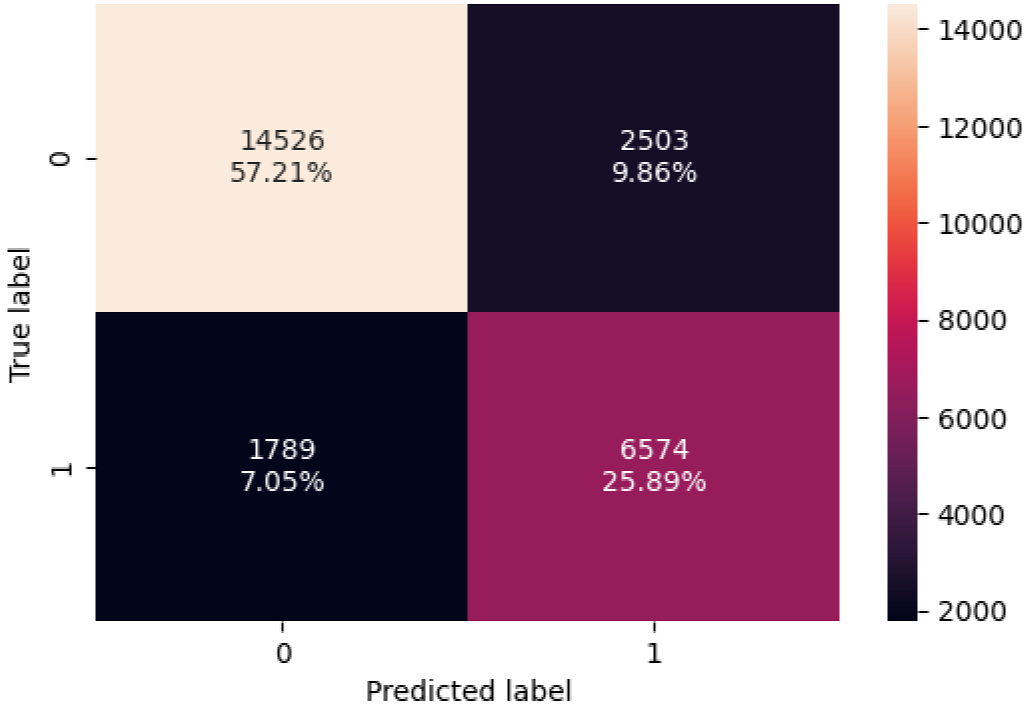
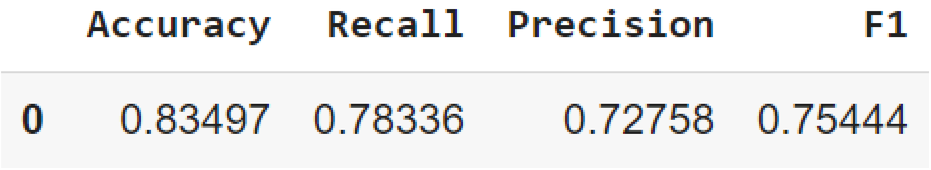


Fig 30: Confusion matrix for Decision Tree Classifier (Improved) - Training data

Decision Tree Classifier - Test Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.83497	0.78336	0.72758	0.75444

Table 16: Performance Metrics for Decision Tree Classifier (Improved) - Test data

Confusion Matrix:

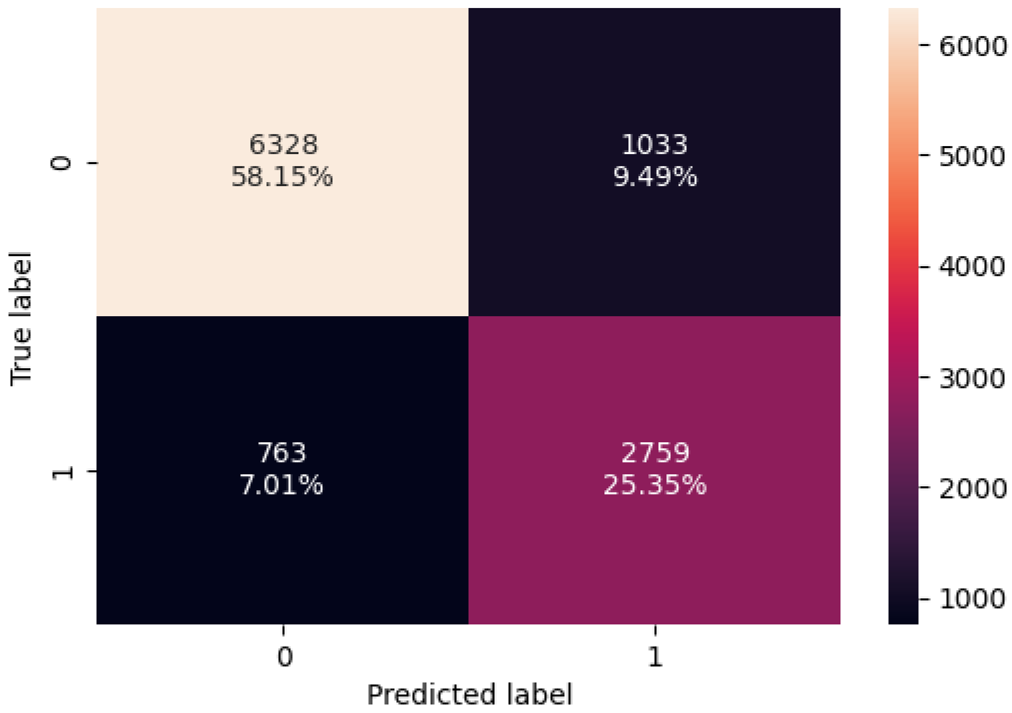


Fig 31: Confusion matrix for Decision Tree Classifier (Improved) - Test data

Decision Tree Classifier - Performance Observations

- The difference between the F1 scores of the training and test sets is negligible, indicating that the model has generalized well.
- The similar accuracy and F1 score indicate that the model is predicting both classes well with few errors.

Decision Tree Classifier (post-pruning)

Post-pruning, also known as cost complexity pruning, is a technique used to simplify decision trees by removing nodes that provide little predictive power, thereby reducing overfitting. This method involves pruning branches of the tree after it has been fully grown, aiming to balance the trade-off between model complexity and accuracy.

ccp_alpha: In cost complexity pruning, the parameter `ccp_alpha` (Cost Complexity Pruning Alpha) plays a crucial role. It controls the trade-off between the tree's complexity and its fit to the training data.

Increasing `ccp_alpha` will lead to more aggressive pruning, resulting in a simpler tree with fewer terminal nodes. Conversely, a lower `ccp_alpha` allows the tree to remain more complex.

ccp_alpha vs. Impurities

The choice of `ccp_alpha` affects the tree's impurity values. With a high `ccp_alpha`, the model prioritizes simplicity over the purity of the nodes. This results in a tree with fewer nodes but potentially higher impurity in the remaining nodes. On the other hand, a lower `ccp_alpha` allows the tree to retain more detailed splits, potentially leading to lower impurity but at the cost of increased complexity.

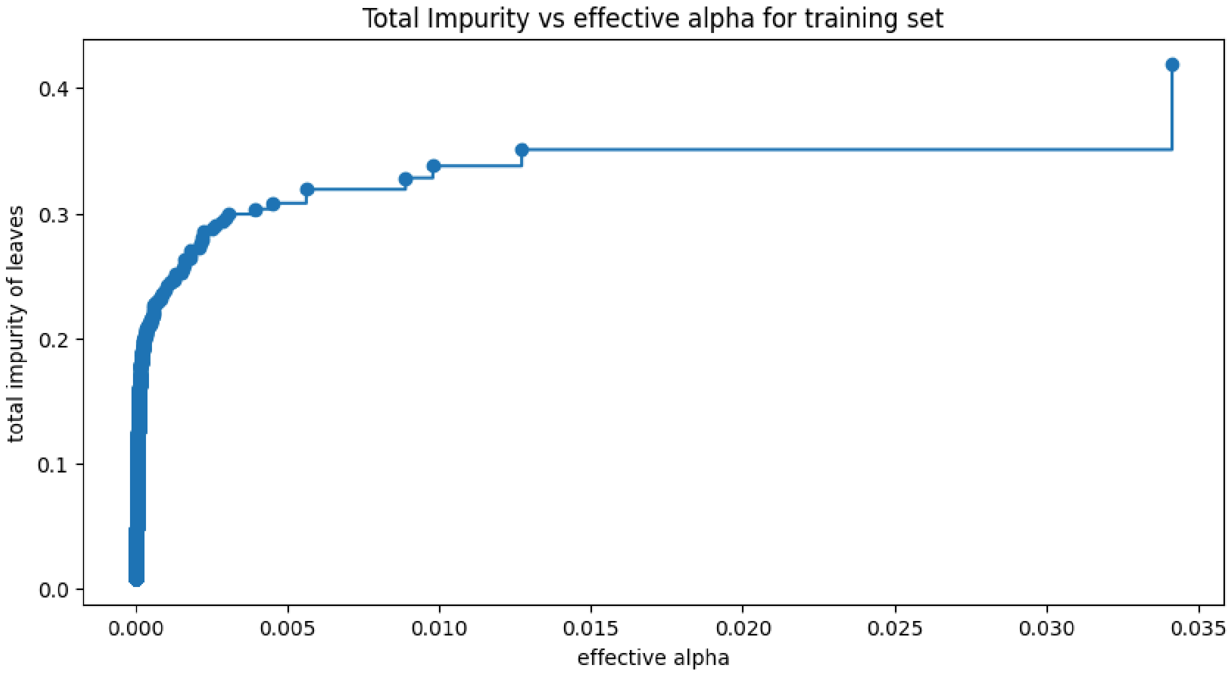


Fig 32: Impurity vs alpha

Choosing the best `ccp_alpha`

The optimal value of `ccp_alpha` is typically chosen by evaluating the model's performance on both training and validation datasets. The goal is to find a balance that maximizes the F1 score, which is a measure of a model's accuracy considering both precision and recall. The best `ccp_alpha` is selected based on the trade-off between a high F1 score on the training data and a good generalization performance on the test data. This ensures the model is not overly complex while maintaining predictive power.

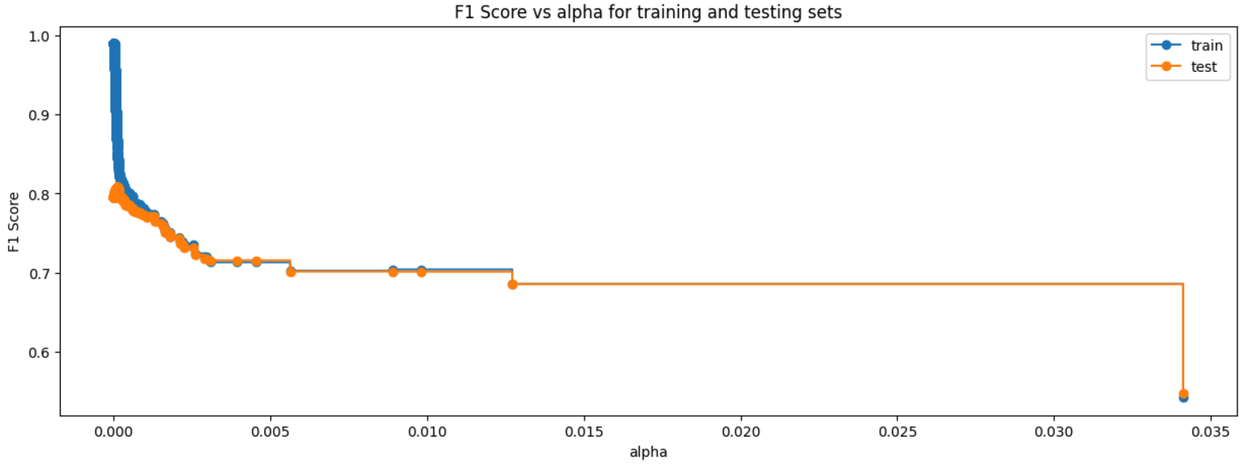
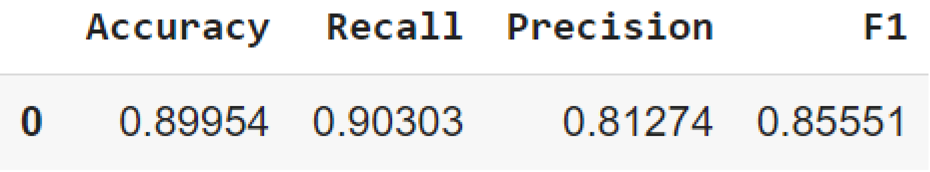


Fig 33: F1 Score vs alpha

From the above figure, we can conclude that the best ‘ccp_apha’ value is 0.00012267633155167043

Decision Tree Classifier - Training Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.89954	0.90303	0.81274	0.85551

Table 17: Performance Metrics for Decision Tree Classifier (Improved) - Training data

Confusion Matrix:

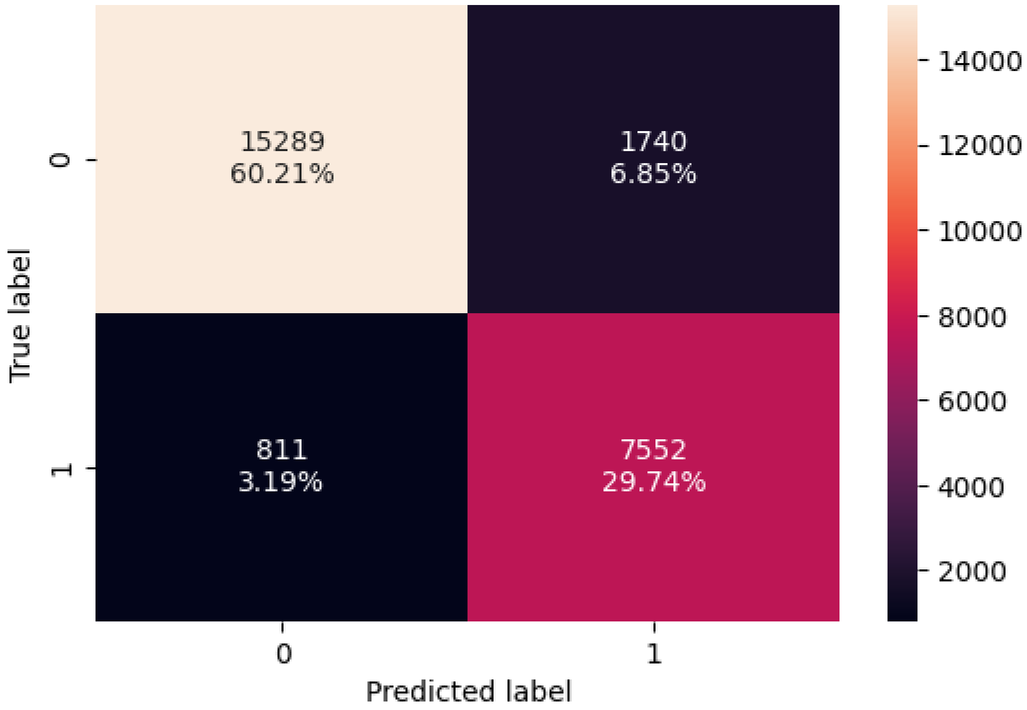
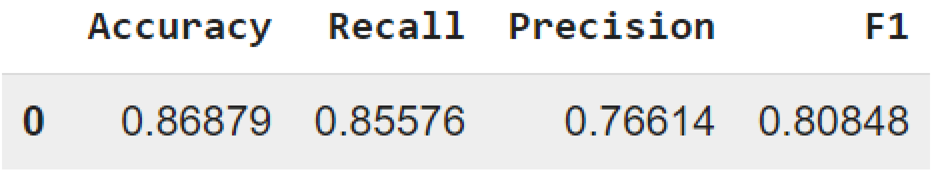


Fig 34: Confusion matrix for Decision Tree Classifier (Improved) - Training data

Decision Tree Classifier - Test Performance

Performance Metrics:



	Accuracy	Recall	Precision	F1
0	0.86879	0.85576	0.76614	0.80848

Table 18: Performance Metrics for Decision Tree Classifier (Improved) - Test data

Confusion Matrix:

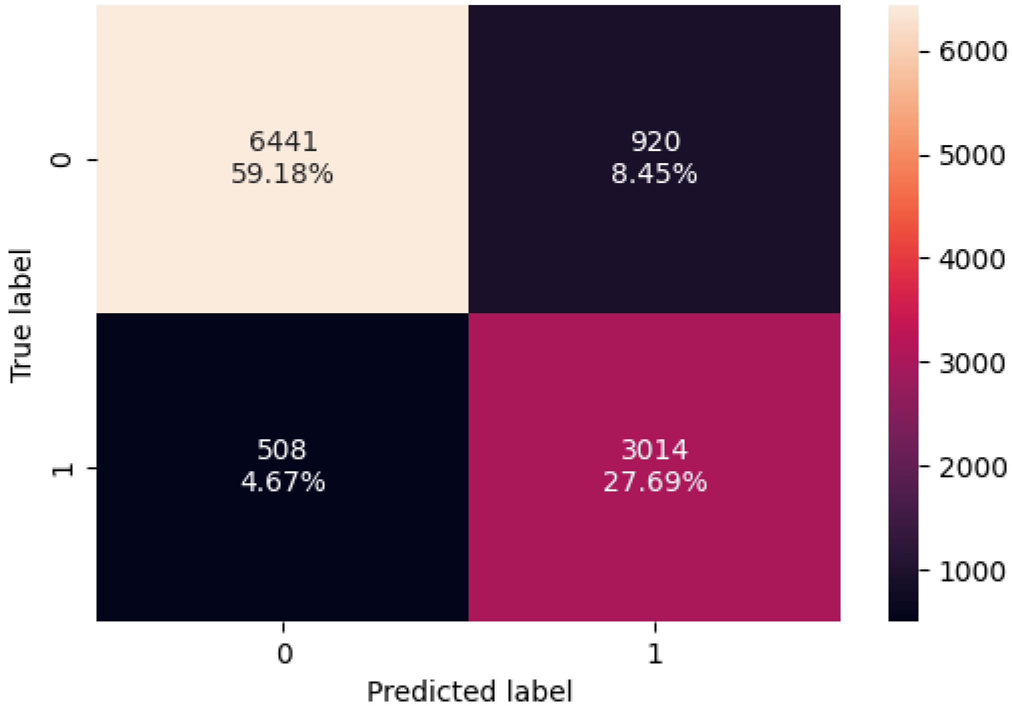


Fig 35: Confusion matrix for Decision Tree Classifier (Improved) - Test data

Decision Tree Classifier - Performance Observations

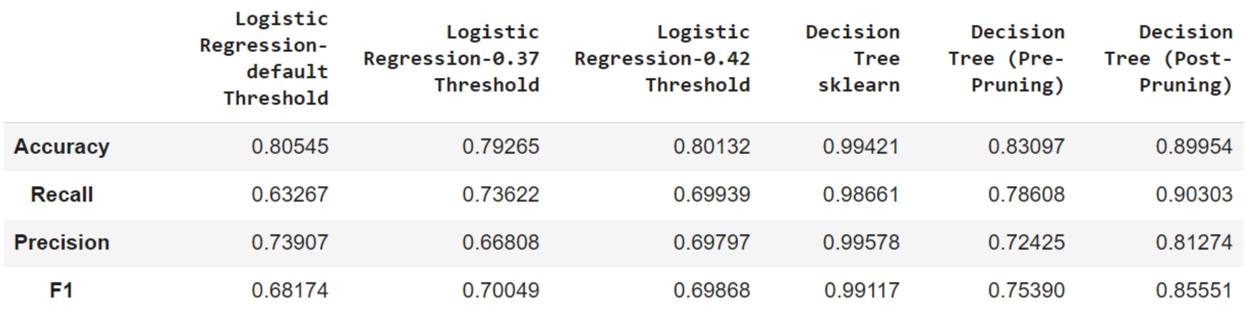
- After post pruning the decision tree the performance has generalized on training and test set.
- With this model, we are getting high recall, but the difference between recall and precision has increased.

Model Comparison and Final Model Selection:

Model Comparison

Based on the evaluation results of the improved models for lead conversion prediction, it is evident that all three models demonstrate notable enhancements in performance metrics compared to their default counterparts. We can analyze this with the help of the performance comparison data below:

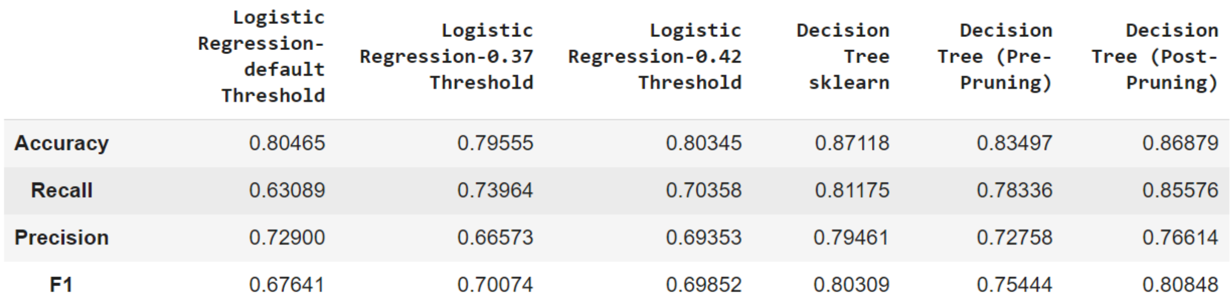
Training Data:



	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.80545	0.79265	0.80132	0.99421	0.83097	0.89954
Recall	0.63267	0.73622	0.69939	0.98661	0.78608	0.90303
Precision	0.73907	0.66808	0.69797	0.99578	0.72425	0.81274
F1	0.68174	0.70049	0.69868	0.99117	0.75390	0.85551

Table 19: Model Performance Comparison for Training Data

Test Data:



	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.80465	0.79555	0.80345	0.87118	0.83497	0.86879
Recall	0.63089	0.73964	0.70358	0.81175	0.78336	0.85576
Precision	0.72900	0.66573	0.69353	0.79461	0.72758	0.76614
F1	0.67641	0.70074	0.69852	0.80309	0.75444	0.80848

Table 20: Model Performance Comparison for Test Data

Final Model Selection:

- The decision tree model with default parameters is overfitting the training data and cannot generalize well.
- The difference between the F1 scores of the training and test sets is negligible for the pre-pruned model, indicating that the model has generalized well.

- Although the difference in scores for the pre-pruned model is negligible, the post-pruned model shows comparatively higher scores. In real-world scenarios, a difference of 0.05 is generally considered negligible.
- The hotel will be able to maintain a balance between resources and brand equity using the post-pruned decision tree model.

We'll move ahead with the post-pruned decision tree model as our final model.

For the post-pruned Decision Tree model, the most important features utilized in identifying the target variable, i.e., booking_status, are:

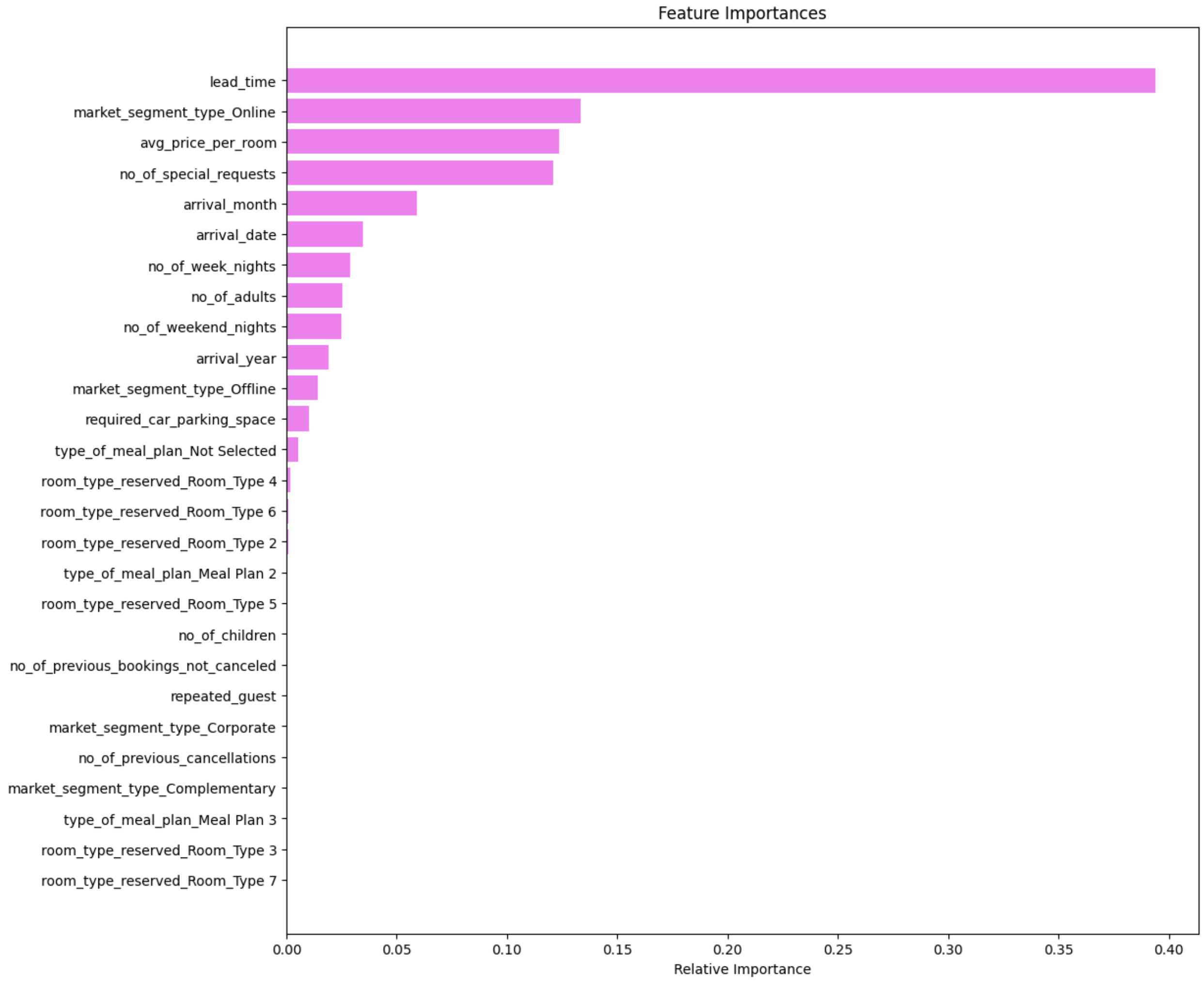


Fig 36: Feature Importance in Model Building

The top 5 attributes affecting the model's predictions are:

1. lead_time
2. market_segment_type_Online
3. avg_price_per_room
4. no_of_special_requests
5. arrival_month

Although other attributes may influence the model performance, the above-given attributes are more likely to be the primary deciding factors in determining whether a customer will cancel his booking or not.

Visualization of the decision tree

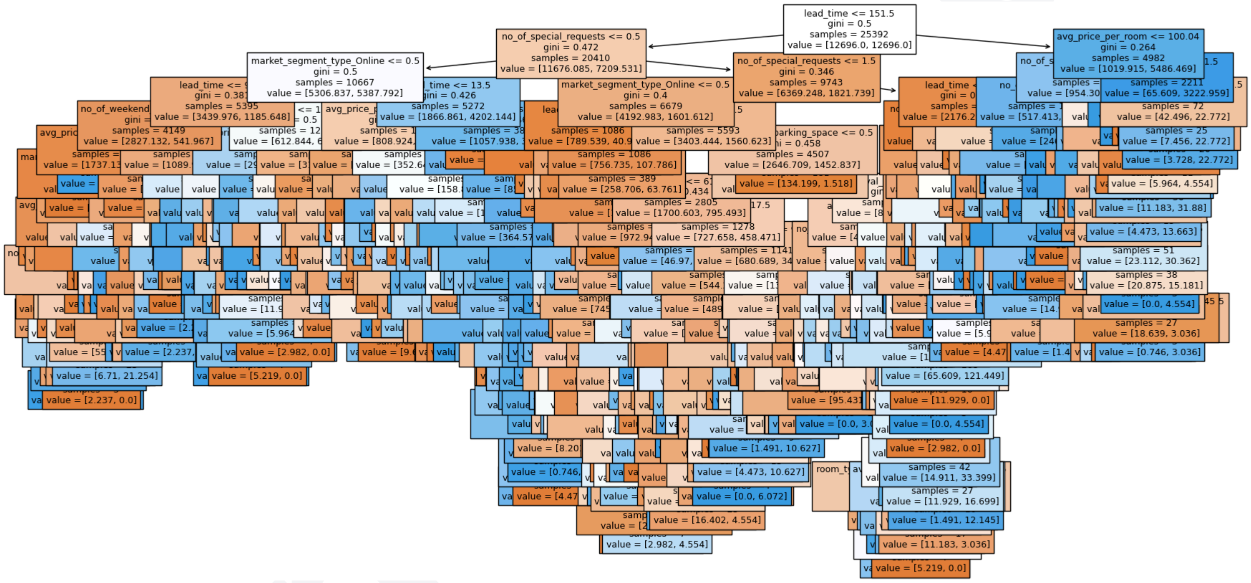


Fig 37: Visualization of the decision tree

- The tree is complex and the decision rules are not clearly visible, but one can expand the same and find out the exact rules.

Conclusions and Recommendations

Conclusions:

- Overall, the Decision Tree model performs better on the dataset.
- Looking at important variables based on p-values in Logistic regression and feature importance in the Decision Tree model
  - Lead Time, Number of special requests, and Average price per room are important in both model
  - From the Logistic Regression model, we observe that Lead Time and Average price per room have a positive relation with bookings getting canceled. And the number of special requests has a negative relation with bookings getting canceled.

Recommendations:

1) The lead time and the number of special requests made by the customer play a key role in identifying whether a booking will be canceled. Bookings where a customer has made a special request and the booking was done within 151 days of the date of arrival are less likely to be canceled.
   - Using this information, the hotel can take the following actions:
     - Set up a system that can send a prompt like an automated email to the customers before the arrival date asking for a re-confirmation of their booking and any changes they would like to make in their bookings.
     - Remind guests about imminent deadlines.
   - The response given by the customer will give the hotel ample time to re-sell the room or make preparations for the customer's requests.

2) Stricter cancellation policies can be adopted by the hotel.

- The bookings where the average price per room is high, and there were special requests associated should not get a full refund as the loss of resources will be high in these cases.
- Ideally, the cancellation policies should be consistent across all market segments but as noticed in our analysis high percentage of bookings done online are

canceled. The booking canceled online should yield less percentage of refund to the customers.

The refunds, cancellation fees, etc should be highlighted on the website/app before a customer confirms their booking to safeguard guests' interest.

3) The length of stay at the hotel can be restricted.

- We saw in our analysis that bookings, where the total length of stay was more than 5 days, had higher chances of getting cancelled.
- Hotels can allow bookings for up to 5 days only and then customers should be asked to re-book if they wish to stay longer. These policies can be relaxed for corporate and Aviation market segments. For other market segments, the process should be fairly easy to not hamper their experience with the hotel.

4) Such restrictions can be strategized by the hotel to generate additional revenue.

- In December and January cancellation to non-cancellation ratio was low. Customers might travel to celebrate Christmas and New Year. The hotel should ensure that enough human resources are available to cater to the needs of the guests.
- October and September saw the highest number of bookings but also a high number of cancellations. This should be investigated further by the hotel.

5) Post-booking interactions can be initiated with the customers.

- Post-booking interactions will show the guests the level of attention and care they will receive at the hotel.
- To give guests a personalized experience, information about local events, nearby places to explore, etc can be shared from time to time.

6) Improving the experience of repeat customers.

- Our analysis shows that there are very few repeat customers and the cancellations among them are very low which is a good indication as repeat customers are important for the hospitality industry as they can help in spreading word of mouth.

- A loyal guest is usually more profitable for the business because they are more familiar with offerings from the hotel they have visited before.
- Attracting new customers is tedious and costs more as compared to a repeated guest.
- A loyalty program that offers - special discounts, access to services in hotels, etc for these customers can help improve their experience.