ML-1 CODED PROJECT

Business Report

DSBA

Submitted By: Maheep Singh
Batch : PGP-DSBA (PGPDSBA.O.AUG24.A)

Table of Contents

List of Figures

List of Tables

Business Context & Data Dictionary

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 cancelled. 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 analyse 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 cancelled in advance, and help in formulating profitable policies for cancellations and refunds.

Data Description

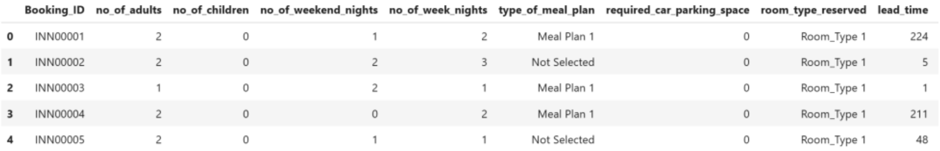
The data contains the different attributes of customers' booking details. The detailed data dictionary is given below: -

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

Rubric Question 1: Exploratory Data Analysis

Data Overview

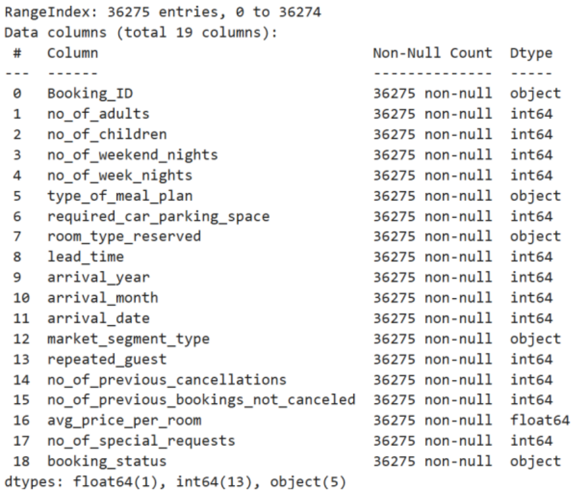
- **Load dataset** & display top 5 rows (Truncated view due to high no. of columns): -



	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	lead_time
0	INN00001	2	0	1	2	Meal Plan 1	0	Room_Type 1	224
1	INN00002	2	0	2	3	Not Selected	0	Room_Type 1	5
2	INN00003	1	0	2	1	Meal Plan 1	0	Room_Type 1	1
3	INN00004	2	0	0	2	Meal Plan 1	0	Room_Type 1	211
4	INN00005	2	0	1	1	Not Selected	0	Room_Type 1	48

Figure 1: Top 5 rows of the dataset

- There are **36275 rows & 19 columns** in the dataset
- **Checking datatypes**: -

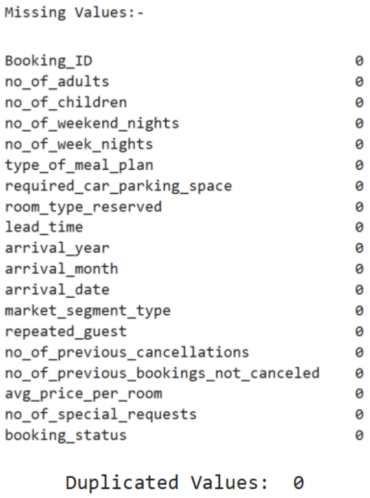


```
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Booking_ID                            36275 non-null  object 
 1   no_of_adults                          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)
```

Figure 2: Datatypes in the Dataset

- ✓ There are 19 columns - 5 object (string/category) type, 1 float (numeric) & 13 Int datatypes in the dataset.

- **Check & Treat Missing & Duplicate Values**: -
  - ✓ Upon checking, neither missing nor duplicate values were found. Hence, no treatment required.

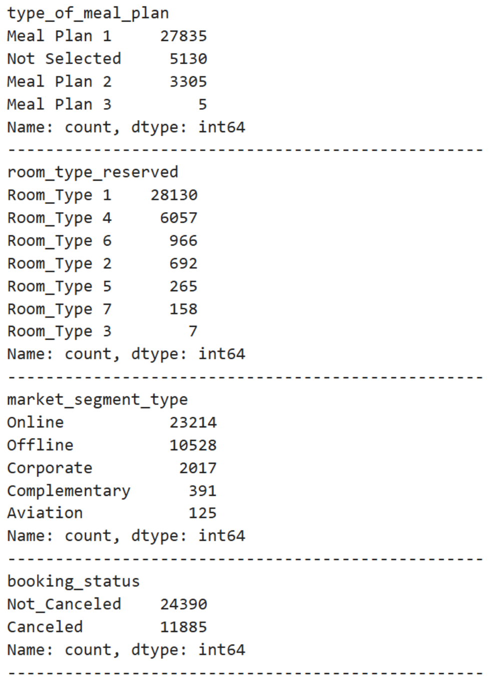


```
Missing Values:-

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

       Duplicated Values:  0
```

Figure 3: Missing/Duplicate Value-check

- **Dropping the 'Booking_ID' column** as it is the unique identifier of each item & serves no purpose in the analyses.

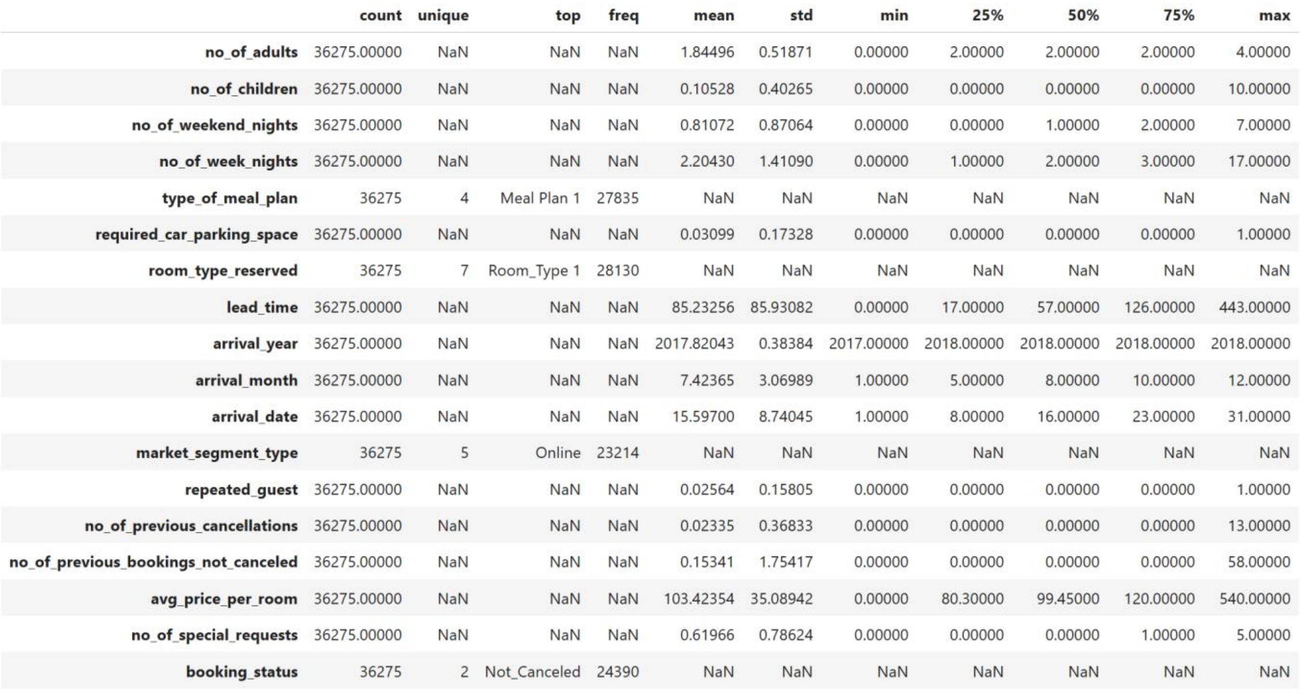
- Upon checking unique values of all categorical variables, **no error/mistypes were found** in the data. Hence, no treatment required.



```
type_of_meal_plan
Meal Plan 1     27835
Not Selected     5130
Meal Plan 2      3305
Meal Plan 3         5
Name: count, dtype: int64
--------------------------------------------------
room_type_reserved
Room_Type 1    28130
Room_Type 4     6057
Room_Type 6      966
Room_Type 2      692
Room_Type 5      265
Room_Type 7      158
Room_Type 3        7
Name: count, dtype: int64
--------------------------------------------------
market_segment_type
Online           23214
Offline          10528
Corporate         2017
Complementary      391
Aviation           125
Name: count, dtype: int64
--------------------------------------------------
booking_status
Not_Canceled    24390
Canceled        11885
Name: count, dtype: int64
--------------------------------------------------
```

Figure 4: Error-value check in the Dataset

- **Statistical Summary of the dataset**: -



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

Figure 5: Statistical Summary of the Dataset

Type	Columns	Observations & Insights
Numerical	no_of_adults	✓ Average adults range between 0 to 4. ✓ Mean is 1.8 adults & median is 2 adults. ✓ Standard Deviation is 0.52.
Numerical	no_of_children	✓ Average children range between 0 to 10. Clearly, there are outliers as which we may treat later (having 10 children is highly unlikely). ✓ Mean is 0.11 children (nearly zero) & median is 0 children. ✓ Standard Deviation is 0.40.
Numerical	no_of_weekend_nights	✓ Average weekend-nights booked range between 0 to 7. ✓ Mean is 0.81 (nearly 1) & median is 1 weekend-nights booked. ✓ Standard Deviation is 0.87.
Numerical	no_of_week_nights	✓ Average week-nights booked range between 0 to 17. ✓ Mean is 2.2 (nearly 2) & median is 2 week-nights booked. ✓ Standard Deviation is 1.41.
Categorical	type_of_meal_plan	✓ Multivariate variable with 4 unique values. ✓ 'Meal Plan 1' is the most dominant value, implying majority of the guests prefer to book rooms with Breakfast included.
Numerical	required_car_parking_space	✓ Inference to this variable is categorical but we would keep this as numerical as we would have to convert this into float for regression. ✓ 1 represents 'Car parking required' & 0 represents 'Car parking not required'. ✓ On an average, Car-parking is not required (mean is 0.03 & median is 0).
Categorical	room_type_reserved	✓ Multivariate variable with 7 unique values. ✓ 'Room_Type 1' is the most dominant value, implying majority of the guests prefer to book rooms with Type 1 (as per hotel-encoding).
Numerical	lead_time	✓ Guest booking lead time ranges between 0 to 443 days. ✓ Mean is 85 days (nearly 3 months) & Median is 57 days (nearly 2 months), implying guests plan their trip/vacation 2-3 months in advance. ✓ Standard Deviation is 85 days.
Numerical	arrival_year	✓ Data consists of 2 values – 2017 & 2018 ✓ Most of the data is for 2018 (Mean – 2017.8; Median – 2018)
Numerical	arrival_month	✓ Data is spread across all 12 months, implying guests book room all-round the year. ✓ Guests prefer to book room in the months July-September (Mean – 7.42; Median – 8), implying guests prefer to plan-their-trip/book in Summers. ✓ Standard deviation of data is 3.07 months.
Numerical	arrival_date	✓ Data is spread across all 31 days, implying guests book room on all days of the month. ✓ Guests prefer to start their booking in the middle of the month (Mean – 15.6; Median – 16). ✓ Standard deviation of data is 8.7 days.
Categorical	market_segment_type	✓ Multivariate variable with 5 unique values. ✓ 'Online' is the most dominant value, implying majority of the guests prefer to book online.
Numerical	repeated_guest	✓ Inference to this variable is categorical but we would keep this as numerical as we would have to convert this into float for regression. ✓ 1 represents 'Repeated Guest' & 0 represents 'Not a Repeated Guest'. ✓ On an average, guests are new rather than repeated/old (mean is 0.02 & median is 0).
Numerical	no_of_previous_cancellations	✓ Number of previous bookings that were cancelled by the customer prior to the current booking, range between 0 & 13. ✓ Mean is 0.02 (nearly 0) & Median is 0 previous cancellations. ✓ Standard deviation in the data is 0.36.
Numerical	no_of_previous_bookings_not_canceled	✓ Number of previous bookings not canceler by the customer prior to the current booking, range between 0 & 58. ✓ Mean is 0.15 (nearly 0) & Median is 0. ✓ Standard deviation in the data is 1.75.
Numerical	avg_price_per_room	✓ Average price per day of the reservation, range between 0 to 540 Euros. ✓ Mean is 103.4 Euros & Median is 99.45 Euros. ✓ Standard deviation in the data is 35.08.
Numerical	no_of_special_requests	✓ Total number of special requests made by the customer (e.g. high floor, view from the room, etc), range between 0 to 5 ✓ Mean is 0.6 (nearly 0) & Median is 0 special requests. ✓ Standard deviation in the data is 0.78.
Categorical	booking_status	✓ Bivariate variable with 2 unique values – 'Canceled' & 'Not_Canceled'. ✓ 'Not_Canceled' is the most dominant value, implying majority of the guests do not cancel their bookings.

Table 1: Statistical Summary – Observations

Univariate & Bivariate Analysis

- **Perform Univariate Analysis** – Use Histograms & Boxplots to analyse each numerical variable, followed by Barplots for categorical variables: -

   1. Distribution of **lead_time**: -

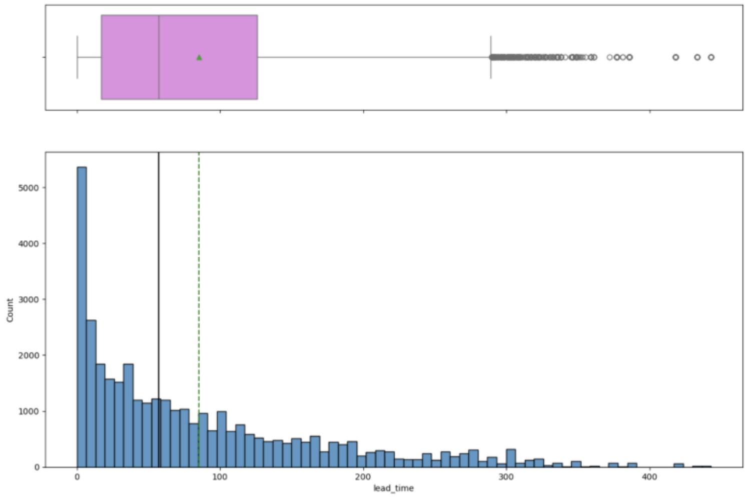


Figure 6: Univariate Analysis – lead_time

✓ **Observations & Insights** can be summarized below: -

- Distribution seems to be slightly right-skewed with different mean & median.
- Unimodal distribution having single peaks.
- Outliers observed, but we would keep the data as-is to avoid any loss of information

   2. Distribution of **avg_price_per_room**: -

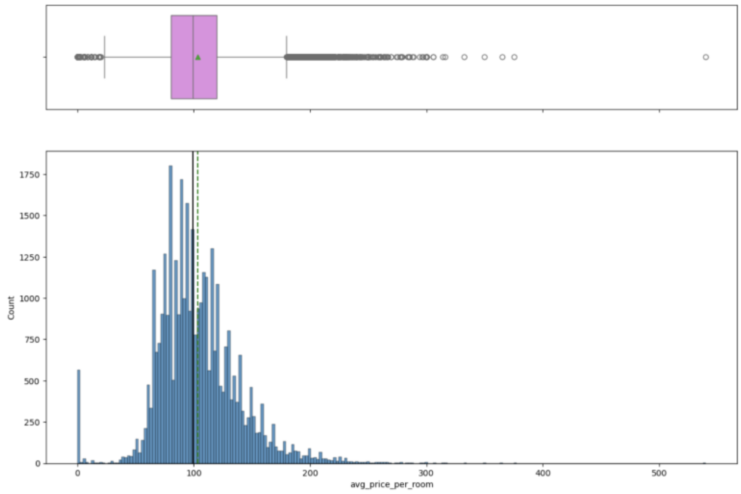


Figure 7: Univariate Analysis – avg_price_per_room

✓ **Observations & Insights** can be summarized below: -

- Distribution seems to be symmetrical with almost similar mean & median.
- Multimodal distribution having multiple peaks.
- Outliers observed, but we would keep the data as-is to avoid any loss of information

3. Distribution of **no_of_adults & no_of_children:** -

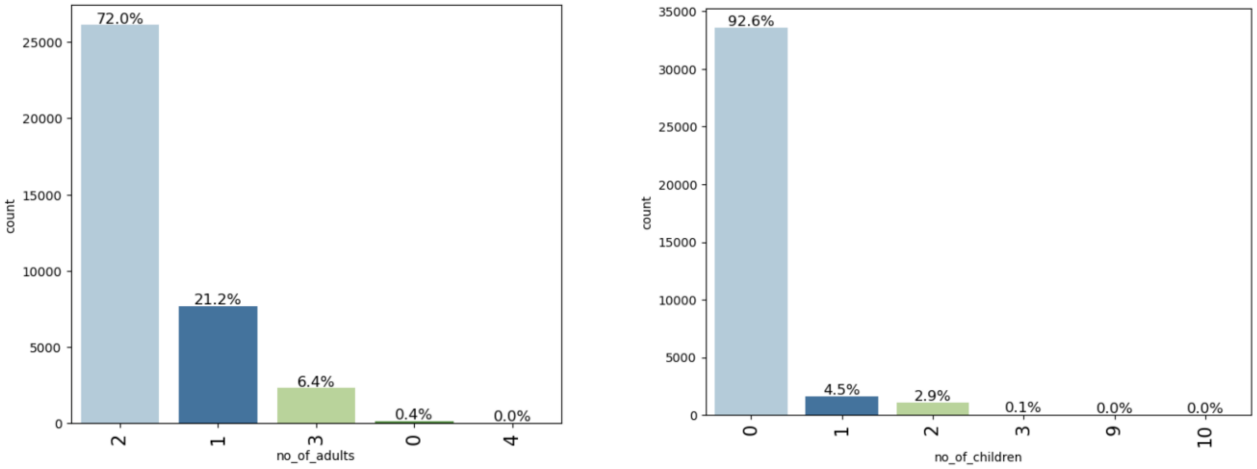


Figure 8: Univariate Analysis – no_of_adults & no_of_children

✓ **Observations & Insights** can be summarized below: -

- Majority of the guests come in pairs (i.e. 2), which contributes to 72% of the distribution | 21% are single travellers, implying they may be on a business trip | 6% come in a group of 3.
- Majority of the guests don't have children (92%) | Very few (7% have either 1 or 2 children)
- Very few guests have recorded 9 or 10 children which seems to be an error. We shall fix this later & replace by 3 children.

4. Distribution of **no_of_weekend_nights & no_of_week_nights**: -

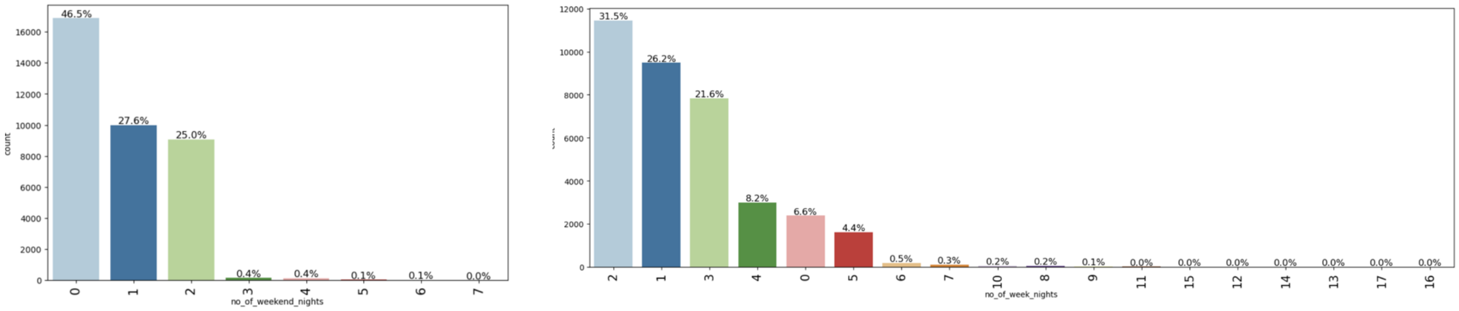


Figure 9: Univariate Analysis – no_of_weekend_nights & no_of_week_nights

✓ **Observations & Insights** can be summarized below: -

- Majority of the data (46%) shows that guests don't include a weekend-night & book 2-5 weeknights in their stay at the hotel

5. Distribution of **type_of_meal_plan & required_car_parking_space:** -

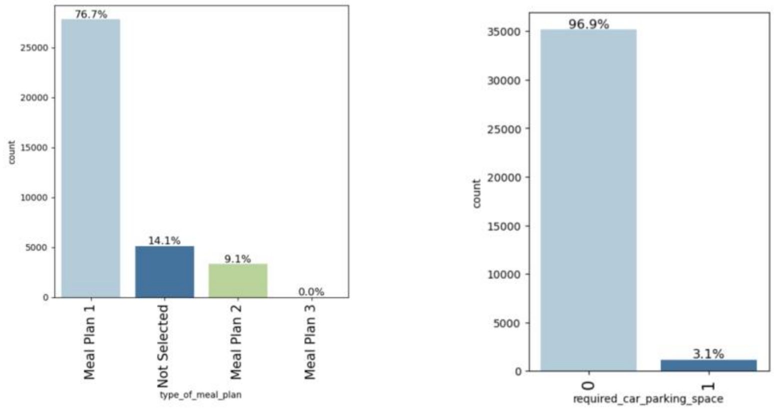


Figure 10: Univariate Analysis – type_of_meal_plan & required_car_parking_space

✓ **Observations & Insights** can be summarized below: -

- Majority of the data (76%) shows that guests prefer meal plan 1 (breakfast included) during their stay.
- Majority of the data suggests that guests don't prefer parking space along with their booking |0 – parking space not required (97%) ; 1 – parking space required (4%)

6. Distribution of **room_type_reserved & market_segment_type**: -

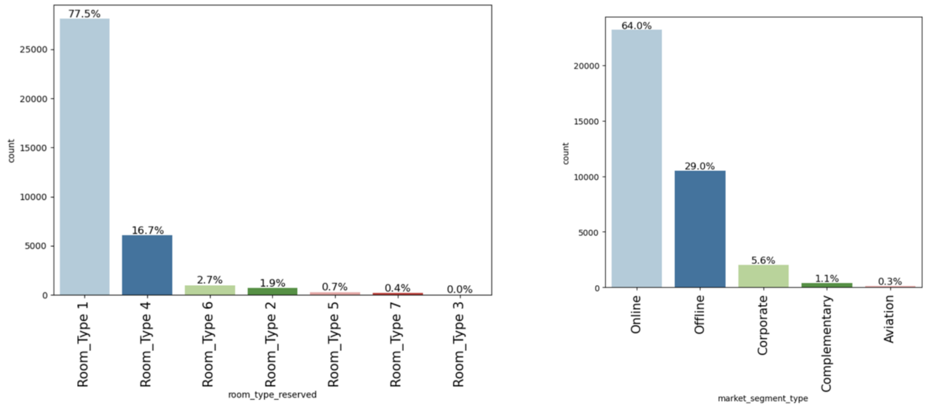


Figure 11: Univariate Analysis – room_type_reserved & market_segment_type

✓ **Observations & Insights** can be summarized below: -

- Majority of the data (78%) suggests that guests prefer Room_Type 1 followed by Room_Type 4 (17%) for their bookings.
- **Individual – Online segment (64%)** contributes the most to bookings followed by Individual – Offline segment (29%).

7. Distribution of **arrival_year, arrival_month & arrival_date**: -

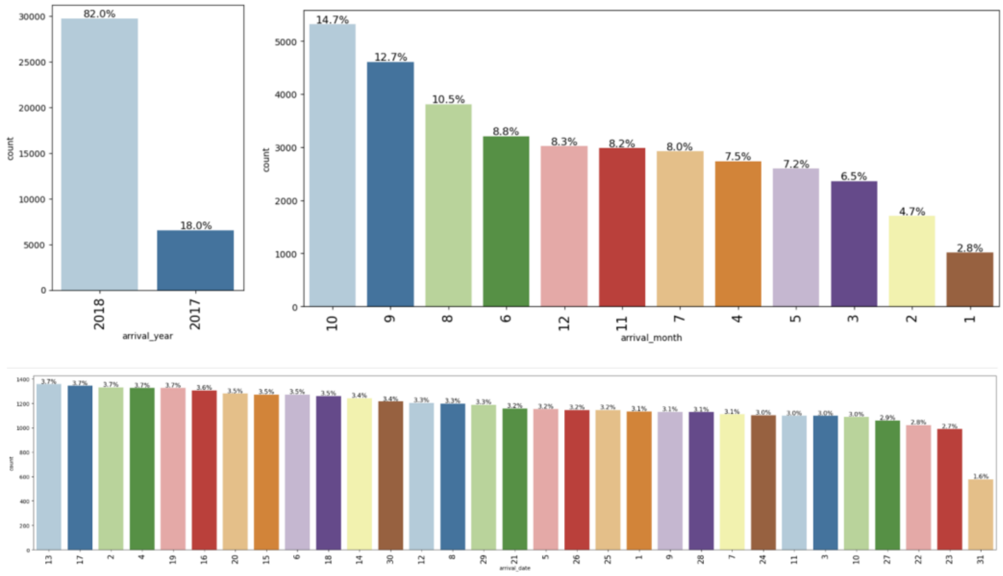


Figure 12: Univariate Analysis – arrival_year, arrival_month & arrival_date

✓ **Observations & Insights** can be summarized below: -

- Majority of the data is for the year 2018 (82%)
- **August, September & October (~15% for October only; and, collectively, 40%) are the busiest months** wherein the hotel receives maximum bookings, implying people prefer to book a hotel (go for a trip) during late summer months.
- Most of the bookings are start either at the start of the month (2-6) or middle of the month (13-19), suggesting people go on vacation either at the start or middle of the month.

8. Distribution of **no_of_previous_cancellations & no_of_previous_bookings_not_canceled**: -

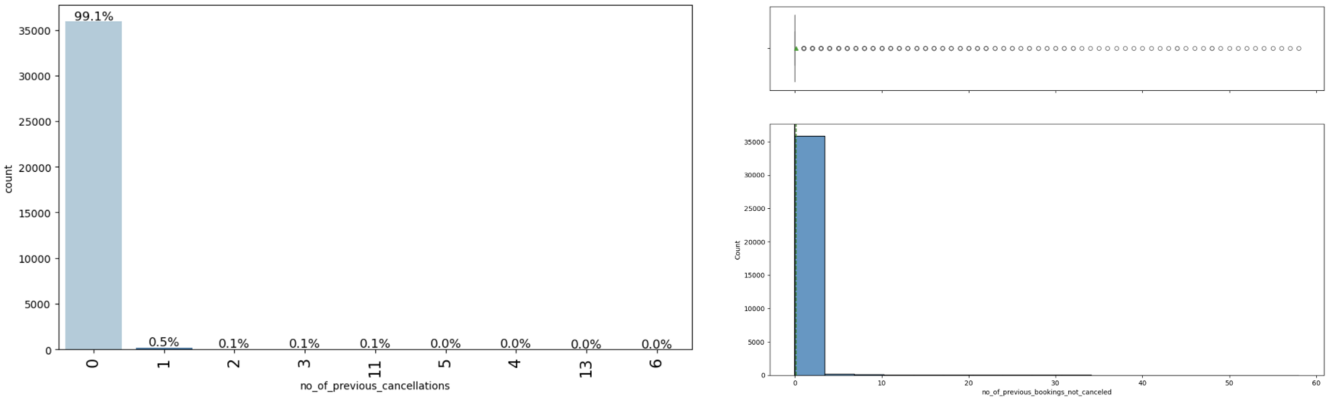


Figure 13: Univariate Analysis – no_of_previous_cancellations & no_of_previous_bookings_not_canceled

✓ **Observations & Insights** can be summarized below: -

- Majority of the data shows no cancelations for both the cases - Number of previous bookings that were cancelled by the customer prior to the current booking & the number of previous bookings not cancelled by the customer prior to the current booking

9. Distribution of **no_of_special_requests & repeated_guest**: -

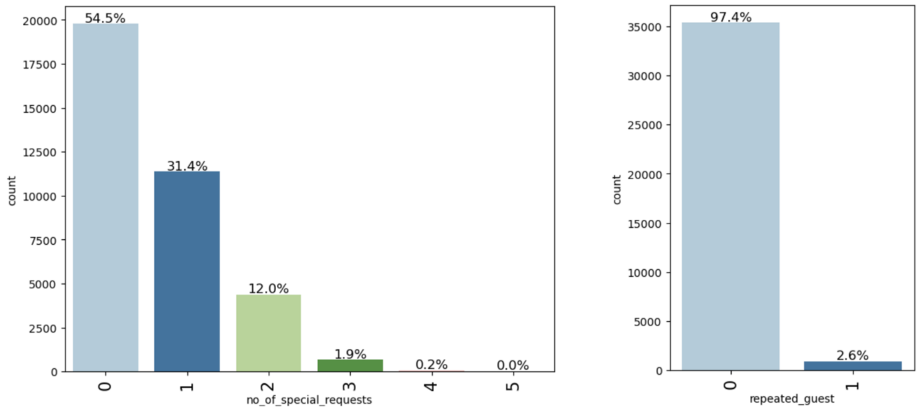


Figure 14: Univariate Analysis – no_of_special_requests & repeated_guest

✓ **Observations & Insights** can be summarized below: -

- Majority of the data suggests that guests have no special requests (55%) like high floor, view from the room, etc.
- Majority of the data suggests that 97% of the guests are new rather than old (i.e. not repeated) | 0 – not repeated (97.4%); 1 – repeated (2.6%)

10. Distribution of **booking_status**: -

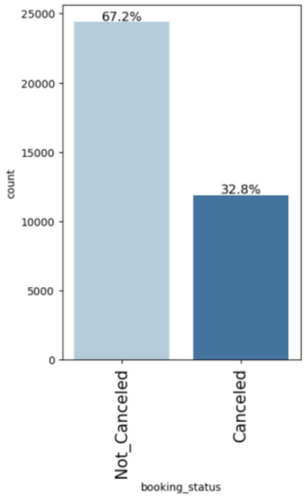


Figure 15: Univariate Analysis – booking_status

✓ **Observations & Insights** can be summarized below: -

- Majority of the data suggests that guests have bookings once done are not cancelled (67%).
- However, there seems to be an **opportunity for improvement here to reduce the cancellations that stand at ~33%.**

- **Perform Bivariate Analysis** – Use Heatmap to carry out bivariate analysis between numerical variables, followed by Distribution plots between numerical & categorical variables: -

1. **Numerical Variables**: -

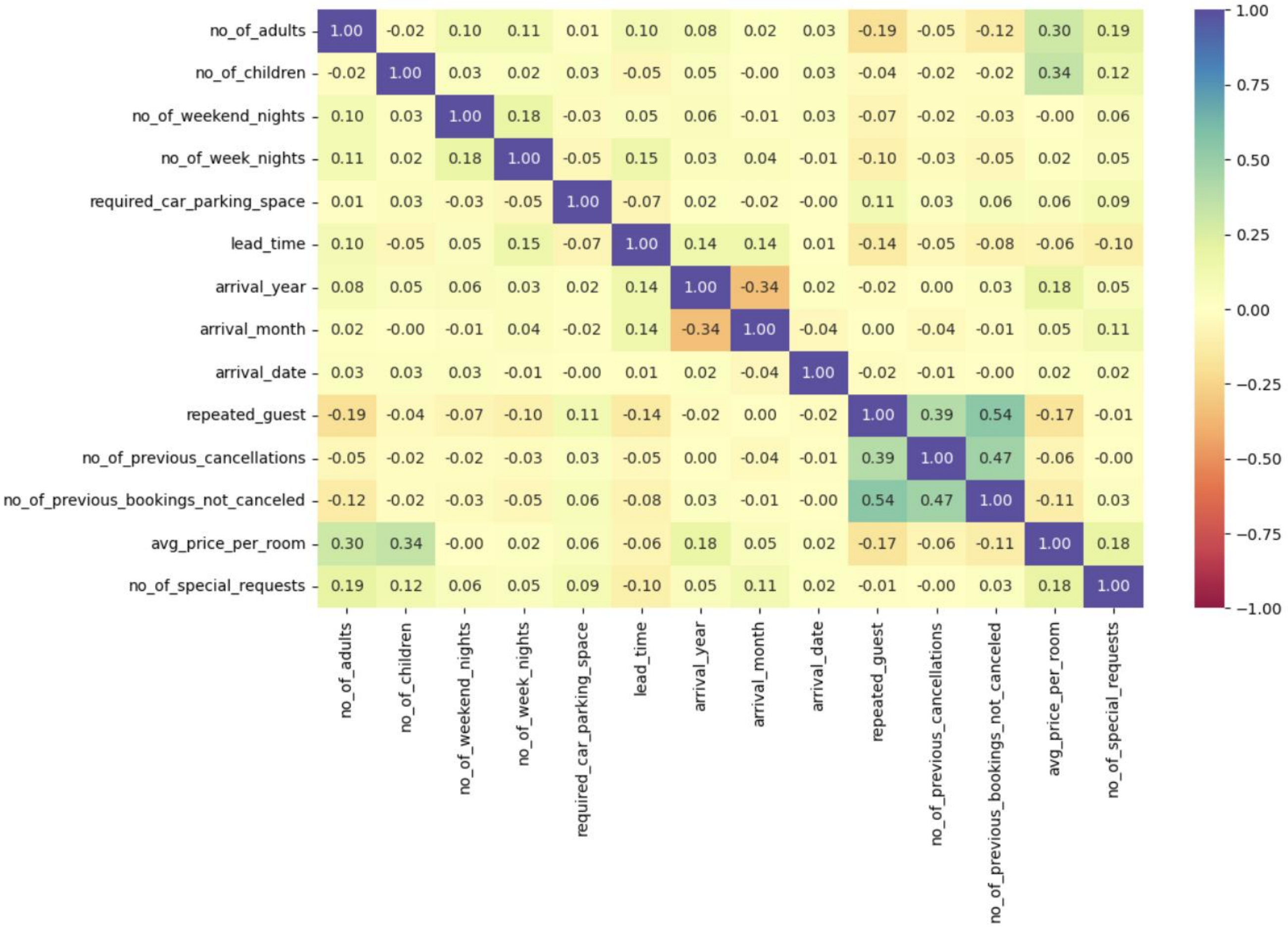


Figure 16: Bivariate Analysis – Numerical Variables

✓ **Observations & Insights** can be summarized below: -

- Clearly, there is no significant correlation trend between any of the numerical variables, i.e. none of the numerical variables are linearly correlated with each other.

2. **Market Segment vs Avg. Room Price**: -

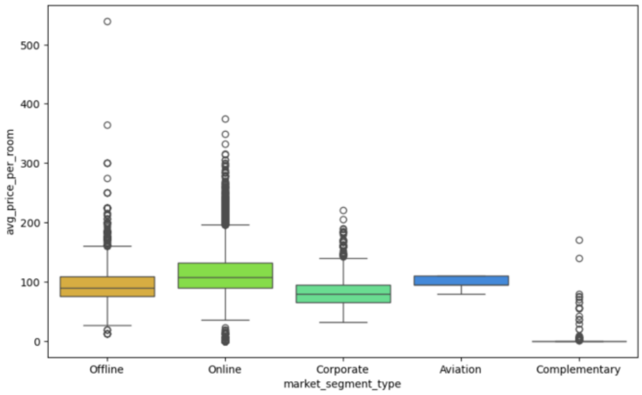
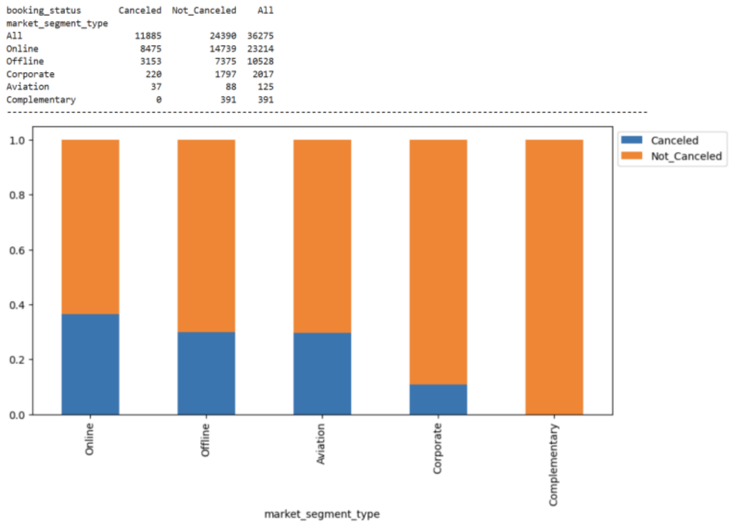


Figure 17: Bivariate Analysis – Market Segment vs Avg. Room Price

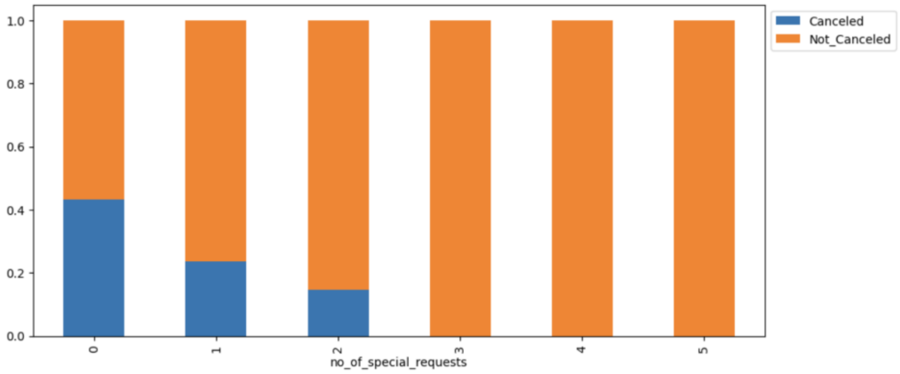
- ✓ **Observations & Insights** can be summarized below: -
  - ➢ **Offline and Online Segments:** These segments have relatively higher medians and broader price distributions compared to others. Outliers suggest the presence of premium-priced rooms.
  - ➢ **Corporate Segment:** This has a smaller IQR and median compared to Offline and Online, indicating more consistent pricing within this category.
  - ➢ **Aviation Segment:** Displays the narrowest IQR and lacks extreme outliers, implying a very uniform pricing structure.
  - ➢ **Complementary Segment**: Exhibits the lowest prices with minimal variation, as it likely includes heavily discounted or free offerings.
  - ➢ **The data suggests pricing differences are influenced by the market segment, with Offline and Online segments commanding higher and more varied prices, whereas Aviation and Complementary segments are more tightly clustered at lower price levels.**

3. **Relationship between Booking Status (Target variable) & other key variables: -**

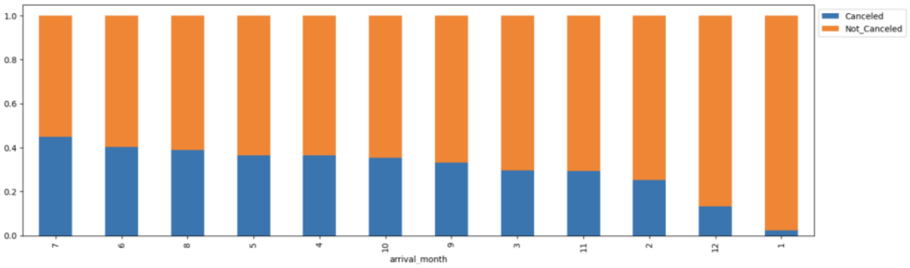


```
booking_status      Canceled  Not_Canceled    All
market_segment_type
All                    11885         24390  36275
Online                  8475         14739  23214
Offline                 3153          7375  10528
Corporate                220          1797   2017
Aviation                  37            88    125
Complementary              0           391    391
------------------------------------------------------------------------------------------------------------------------
```

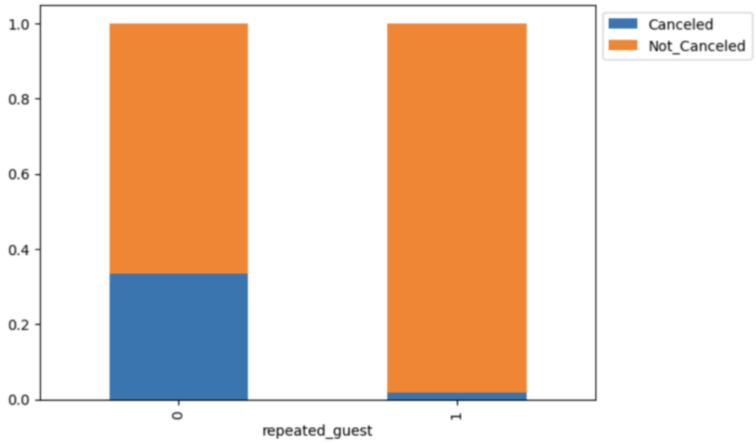
```
booking_status          Canceled  Not_Canceled    All
no_of_special_requests
All                        11885         24390  36275
0                           8545         11232  19777
1                           2703          8670  11373
2                            637          3727   4364
3  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



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



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



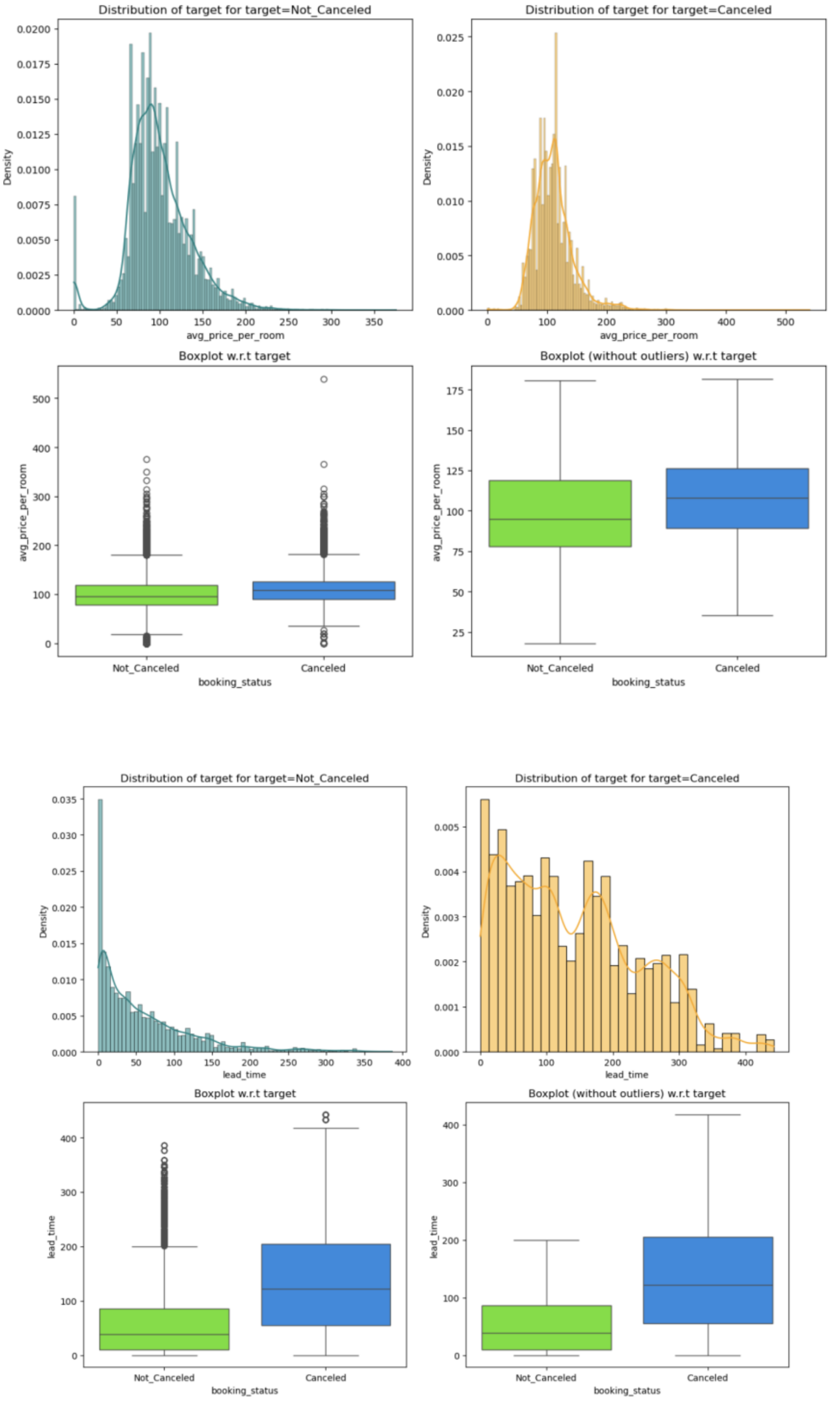


Figure 18: Bivariate Analysis – Relationship between Booking_Canceled (Target variable) & key variables

- ✓ **Observations & Insights** can be summarized below: -
  - ➢ **Booking Status & Market Segment**: -
    - The Online segment has the highest cancellation rate, possibly due to ease of booking and cancelling online.
    - Segments like Corporate, Aviation, and Complementary are more stable and show minimal cancellations, reflecting more predictable booking patterns.
    - These patterns highlight the need for targeted strategies to reduce cancellations, particularly in the Online and Offline segments.
  - ➢ **Booking Status & Special Requests**
    - Customers making no special requests are more likely to cancel their bookings, potentially indicating a lack of commitment or seriousness.
    - Conversely, customers with multiple special requests are less likely to cancel, possibly because these requests indicate higher engagement or a stronger intention to travel.
    - This pattern suggests that the number of special requests could be a useful predictor of booking reliability. Strategies to encourage customers to make requests (e.g., offering customization options) might help reduce cancellations.
  - ➢ **Booking Status & Arrival Month**
    - High Cancellation Rates (Peak Periods): Months 7, 8, and 9 (July, August, and September) have higher proportions of cancellations compared to other months. These months likely correspond to peak travel seasons or holidays, where demand is higher, and customers might make speculative bookings, leading to more cancellations.
    - Low Cancellation Rates (Off-Peak Periods): Months 11, 12, 1, and 2 (November, December, January, and February) show the lowest cancellation rates, indicating more reliable bookings. These months may represent off-peak travel seasons when customers are more deliberate about their travel plans.
    - Transition Months: Months like 4, 5, and 10 (April, May, and October) show moderate cancellation rates, falling between the peak and off-peak patterns.
    - Seasonality and Cancellation Behaviour – Cancellations tend to increase during high-demand months, possibly due to overbooking, speculative reservations, or higher competition for accommodations.
    - In contrast, lower-demand months see fewer cancellations, likely due to more committed travelers.
    - Actionable Takeaways:
      - Peak Periods: Hotels can implement stricter cancellation policies during peak months to mitigate losses from speculative bookings.
      - Off-Peak Periods: Focus on promotions and flexibility to attract more customers, as cancellations are already lower during this time.
  - ➢ **Booking Status & Repeated Guest**
    - Non-Repeated Guests (0): A significant proportion of bookings are cancelled (~33%), indicating that non-repeated guests are less reliable.
    - Repeated Guests (1): The cancellation rate is extremely low, with most bookings being not cancelled (almost 98% of bookings are honoured). This indicates that repeated guests are highly reliable.
    - Actionable Takeaways:
      - Focus on Loyalty Programs: Hotels should invest in programs to convert non-repeated guests into repeated ones by enhancing customer experience, offering incentives for repeat bookings, and maintaining strong communication.
      - Targeted Policies for Non-Repeated Guests: Stricter cancellation policies or deposits could be applied to non-repeated guests to reduce speculative bookings.
  - ➢ **Booking Status & Average Price per Room**
    - **Density Plots:**
      - Not Cancelled: The distribution is slightly right-skewed, with most prices concentrated between 50 to 150 units. There are some outliers with higher prices, but they are relatively uncommon.
      - Cancelled: The density plot for cancelled bookings also shows a peak around 50 to 100 units, but the distribution is more sharply skewed compared to not cancelled bookings. This suggests that lower prices are more common for cancelled bookings.
    - **Box Plots:**
      - Both categories have similar medians (approximately 100 units)
      - Cancelled bookings exhibit a wider range and more outliers, especially at the higher price levels.
      - Not cancelled bookings are more concentrated, with fewer extreme outliers.

- By removing outliers, the distributions appear more compact. The interquartile range (IQR) is slightly wider for cancelled bookings compared to not cancelled, indicating more variability in prices for cancelled bookings.

- **Summary:**
  - Lower-priced rooms are more likely to be cancelled, as indicated by the density plot.
  - Higher-priced rooms are less frequently cancelled, but when cancelled, they often appear as outliers.
  - Cancelled bookings show greater variability in prices, with a wider range and more extreme outliers.
  - Not cancelled bookings are more stable, with fewer deviations from the median.
- **Actionable Takeaways:**
  - Rooms priced at the lower range (50-100 units) may benefit from stricter cancellation policies or incentives to reduce cancellations.
  - For higher-priced rooms, providing enhanced customer engagement or premium cancellation options might mitigate losses.

➢ **Booking Status & Lead time**

- **Density Plots:**
  - Not Cancelled: The density plot shows that bookings that were not cancelled have a heavily skewed distribution, with most lead times concentrated at low values (0 to 50 days). The likelihood of not cancelling decreases as lead time increases, with very few bookings having lead times greater than 200 days.
  - Cancelled: The density plot for cancelled bookings is broader, indicating that cancellations occur across a wider range of lead times. Cancellations are more common with higher lead times, particularly between 50 and 200 days, and taper off beyond 300 days.
- **Box Plots:**
  - Cancelled bookings show a much wider range of lead times, with numerous outliers extending beyond 300 days.
  - Not cancelled bookings have a more compact distribution, with most lead times below 100 days.
  - By removing outliers, the median and interquartile range (IQR) are more clearly visible. Cancelled bookings have a higher median lead time compared to not cancelled bookings. The spread of lead times for cancelled bookings remains larger, even without outliers
- **Summary:**
  - Short Lead Times Are More Reliable: Bookings with short lead times (e.g., <50 days) are less likely to be cancelled, as shown by the higher density and smaller IQR in the "Not Cancelled" category.
  - Long Lead Times Are Riskier: Cancellations are significantly more common for bookings with long lead times, indicating greater uncertainty or indecision for bookings made far in advance.
  - Cancelled bookings show greater variability in lead times, while not cancelled bookings are more consistent and concentrated around shorter lead times.
- **Actionable Takeaways:**
  - Targeted Policies: Implement stricter cancellation policies or require deposits for bookings with long lead times to reduce the risk of cancellations. Offer flexible policies or incentives for short-lead-time bookings to encourage reliability.
  - Operational Planning: Use lead time as a predictive factor for cancellations, focusing on high-lead-time bookings for follow-ups or retention strategies. Allocate resources efficiently during peak cancellation windows (50–200 days lead time).

▪ **Answer EDA Questions in the Problem: -**

1. What are the busiest months in the hotel? – please refer link
2. Which market segment do most of the guests come from? – please refer link
3. Hotel rates are dynamic and change according to demand and customer demographics. What are the differences in room prices in different market segments? – please refer link
4. What percentage of bookings are cancelled? – please refer link
5. Repeating guests are the guests who stay in the hotel often and are important to brand equity. What percentage of repeating guests cancel? – please refer link1 & link2
6. Many guests have special requirements when booking a hotel room. Do these requirements affect booking cancellation? – please refer link1 & link2

Rubric Question 2: Data Preprocessing

Duplicate & Missing/Error Value-check

- Please refer Check & Treat Duplicate, Missing & Error Values section.
- No treatment required as explained in the section above.

Feature Engineering

- We would be **dropping the 'Booking_ID' column** as it is a unique identifier & serves no purpose in the analysis/correlation in the problem
- There are **2 variables which would be inferred as categorical but are numeric** in the dataset – **'required_car_parking_space' & 'repeated_guest'**. However, we prefer not to change it to object/category data type with values 'Yes' & 'No' (against 1 & 0) as we would have to convert them to float data type post encoding for regression. So, this seems to be a redundant step; so, we keep them as is.

Outlier Treatment

- Below is a summary of the histograms & outlier information for each numerical variable: -

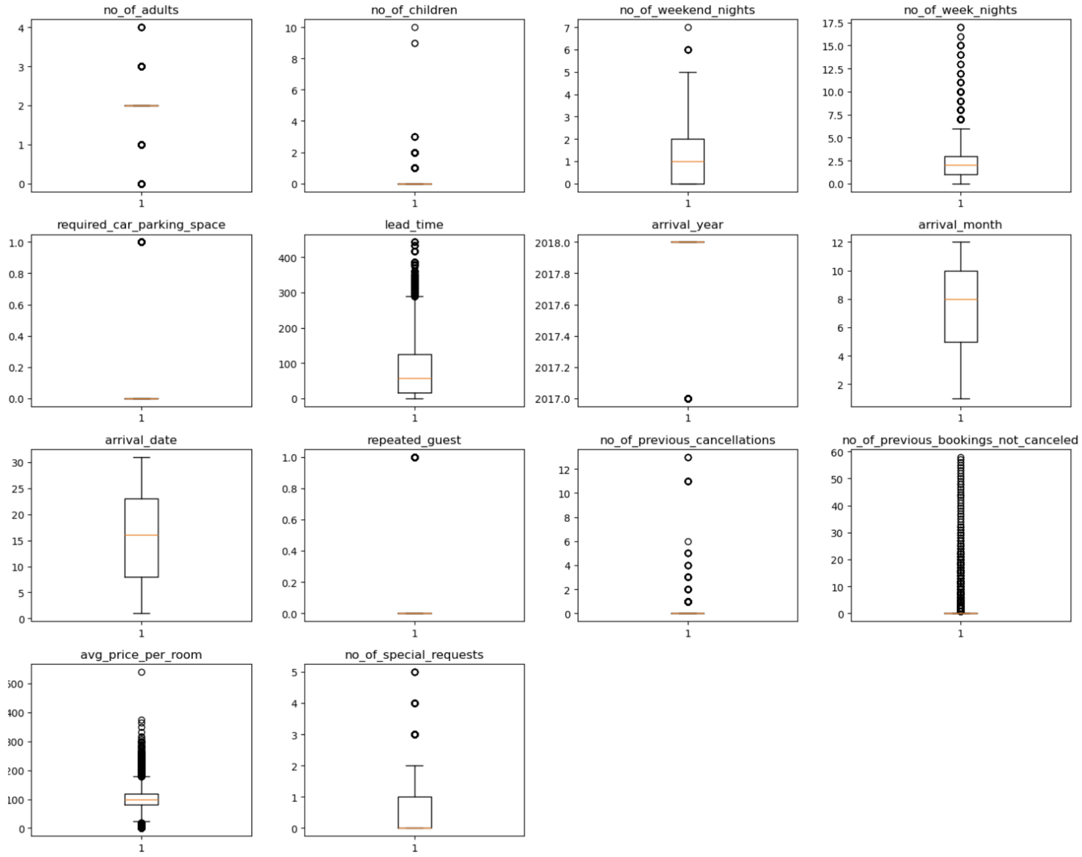
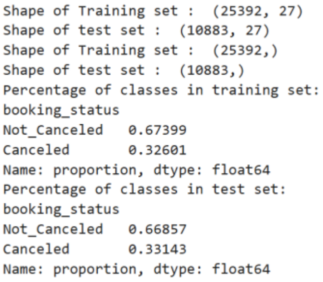


Figure 19: Outlier Inspection

- **'no_of_children' variable has values '9' & '10'** which seem to be **highly unlikely**. We would be **replacing** both values with the next value i.e. **'3'** so that we not only treat the outliers but ensure we don't lose information for these bookings, signifying, more than usual no. of children.
- There are **outliers in other numerical variables** also, but we would **keep them as-is to avoid any loss of information.**

Data Preparation for Modelling

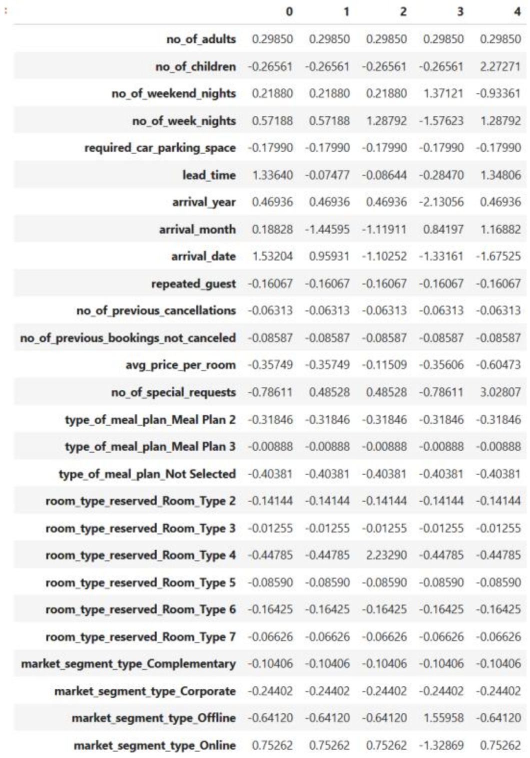
- As a data scientist, we want to analyse the data provided to find which factors have a high influence on booking cancellations, build a predictive model that can predict which booking is going to be cancelled in advance.
  - We split the data into Dependent (booking_status) & Independent variable (remaining others) Data Frames.
  - Before we proceed to build a model, we will encode categorical features.
  - We will split the data into Train and Test datasets to be able to evaluate the model that we build on the Train data. We will build a model using the Train dataset and then check its performance on Test dataset.
- **Creating Dummy Variables** & convert values to **Float datatype**: -
  - **One Hot Encoding** is a method for converting categorical variables into a binary format. It creates new binary columns (0s and 1s) for each category in the original variable. Each category in the original column is represented as a separate column, where a value of 1 indicates the presence of that category, and 0 indicates its absence.
  - It takes 'True' or 'False' as its values and is used to get (k-1) dummies out of k categorical levels (sorted in the ascending order of the alphabet) by removing the first level
- We will use **'random_state' value 42** (so that it returns a shuffled dataset) to **split the data into Train and Test Datasets in the ratio of 70:30**. Below is the summary of the split datasets: -



```
Shape of Training set :  (25392, 27)
Shape of test set :  (10883, 27)
Shape of Training set :  (25392,)
Shape of test set :  (10883,)
Percentage of classes in training set:
booking_status
Not_Canceled   0.67399
Canceled       0.32601
Name: proportion, dtype: float64
Percentage of classes in test set:
booking_status
Not_Canceled   0.66857
Canceled       0.33143
Name: proportion, dtype: float64
```

Figure 20: Train-Test Split Dataset Summary

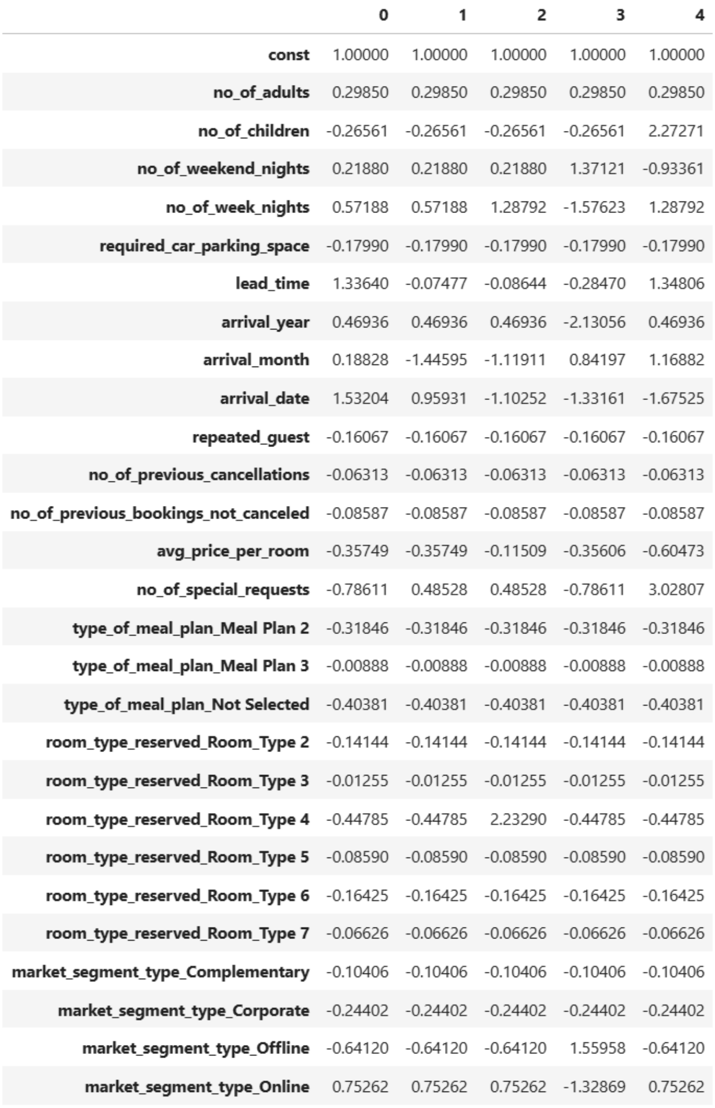
- **Scale the Train & Test Dataset** (of dependent variables i.e. X) & store as a new dataset to be used for **building KNN Classifier Model**. Below is the snapshot of the 1st 5 rows (in transpose for better visibility) of the scaled train dataset: -



	0	1	2	3	4
no_of_adults	0.29850	0.29850	0.29850	0.29850	0.29850
no_of_children	-0.26561	-0.26561	-0.26561	-0.26561	2.27271
no_of_weekend_nights	0.21880	0.21880	0.21880	1.37121	-0.93361
no_of_week_nights	0.57188	0.57188	1.28792	-1.57623	1.28792
required_car_parking_space	-0.17990	-0.17990	-0.17990	-0.17990	-0.17990
lead_time	1.33640	-0.07477	-0.08644	-0.28470	1.34806
arrival_year	0.46936	0.46936	0.46936	-2.13056	0.46936
arrival_month	0.18828	-1.44595	-1.11911	0.84197	1.16882
arrival_date	1.53204	0.95931	-1.10252	-1.33161	-1.67525
repeated_guest	-0.16067	-0.16067	-0.16067	-0.16067	-0.16067
no_of_previous_cancellations	-0.06313	-0.06313	-0.06313	-0.06313	-0.06313
no_of_previous_bookings_not_canceled	-0.08587	-0.08587	-0.08587	-0.08587	-0.08587
avg_price_per_room	-0.35749	-0.35749	-0.11509	-0.35606	-0.60473
no_of_special_requests	-0.78611	0.48528	0.48528	-0.78611	3.02807
type_of_meal_plan_Meal Plan 2	-0.31846	-0.31846	-0.31846	-0.31846	-0.31846
type_of_meal_plan_Meal Plan 3	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888
type_of_meal_plan_Not Selected	-0.40381	-0.40381	-0.40381	-0.40381	-0.40381
room_type_reserved_Room_Type 2	-0.14144	-0.14144	-0.14144	-0.14144	-0.14144
room_type_reserved_Room_Type 3	-0.01255	-0.01255	-0.01255	-0.01255	-0.01255
room_type_reserved_Room_Type 4	-0.44785	-0.44785	2.23290	-0.44785	-0.44785
room_type_reserved_Room_Type 5	-0.08590	-0.08590	-0.08590	-0.08590	-0.08590
room_type_reserved_Room_Type 6	-0.16425	-0.16425	-0.16425	-0.16425	-0.16425
room_type_reserved_Room_Type 7	-0.06626	-0.06626	-0.06626	-0.06626	-0.06626
market_segment_type_Complementary	-0.10406	-0.10406	-0.10406	-0.10406	-0.10406
market_segment_type_Corporate	-0.24402	-0.24402	-0.24402	-0.24402	-0.24402
market_segment_type_Offline	-0.64120	-0.64120	-0.64120	1.55958	-0.64120
market_segment_type_Online	0.75262	0.75262	0.75262	-1.32869	0.75262

Figure 21: Scaled Dataset

- **Add an Intercept to the Scaled Train & Test Datasets** (of dependent variables i.e. X) & store as a new dataset to be used for **Logistic Regression Model**. Below is the snapshot of the 1st 5 rows (in transpose for better visibility) of the scaled train dataset with intercept: -



	0	1	2	3	4
const	1.00000	1.00000	1.00000	1.00000	1.00000
no_of_adults	0.29850	0.29850	0.29850	0.29850	0.29850
no_of_children	-0.26561	-0.26561	-0.26561	-0.26561	2.27271
no_of_weekend_nights	0.21880	0.21880	0.21880	1.37121	-0.93361
no_of_week_nights	0.57188	0.57188	1.28792	-1.57623	1.28792
required_car_parking_space	-0.17990	-0.17990	-0.17990	-0.17990	-0.17990
lead_time	1.33640	-0.07477	-0.08644	-0.28470	1.34806
arrival_year	0.46936	0.46936	0.46936	-2.13056	0.46936
arrival_month	0.18828	-1.44595	-1.11911	0.84197	1.16882
arrival_date	1.53204	0.95931	-1.10252	-1.33161	-1.67525
repeated_guest	-0.16067	-0.16067	-0.16067	-0.16067	-0.16067
no_of_previous_cancellations	-0.06313	-0.06313	-0.06313	-0.06313	-0.06313
no_of_previous_bookings_not_canceled	-0.08587	-0.08587	-0.08587	-0.08587	-0.08587
avg_price_per_room	-0.35749	-0.35749	-0.11509	-0.35606	-0.60473
no_of_special_requests	-0.78611	0.48528	0.48528	-0.78611	3.02807
type_of_meal_plan_Meal Plan 2	-0.31846	-0.31846	-0.31846	-0.31846	-0.31846
type_of_meal_plan_Meal Plan 3	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888
type_of_meal_plan_Not Selected	-0.40381	-0.40381	-0.40381	-0.40381	-0.40381
room_type_reserved_Room_Type 2	-0.14144	-0.14144	-0.14144	-0.14144	-0.14144
room_type_reserved_Room_Type 3	-0.01255	-0.01255	-0.01255	-0.01255	-0.01255
room_type_reserved_Room_Type 4	-0.44785	-0.44785	2.23290	-0.44785	-0.44785
room_type_reserved_Room_Type 5	-0.08590	-0.08590	-0.08590	-0.08590	-0.08590
room_type_reserved_Room_Type 6	-0.16425	-0.16425	-0.16425	-0.16425	-0.16425
room_type_reserved_Room_Type 7	-0.06626	-0.06626	-0.06626	-0.06626	-0.06626
market_segment_type_Complementary	-0.10406	-0.10406	-0.10406	-0.10406	-0.10406
market_segment_type_Corporate	-0.24402	-0.24402	-0.24402	-0.24402	-0.24402
market_segment_type_Offline	-0.64120	-0.64120	-0.64120	1.55958	-0.64120
market_segment_type_Online	0.75262	0.75262	0.75262	-1.32869	0.75262

Figure 22: Intercept on Scaled Dataset

Rubric Question 3: Model building

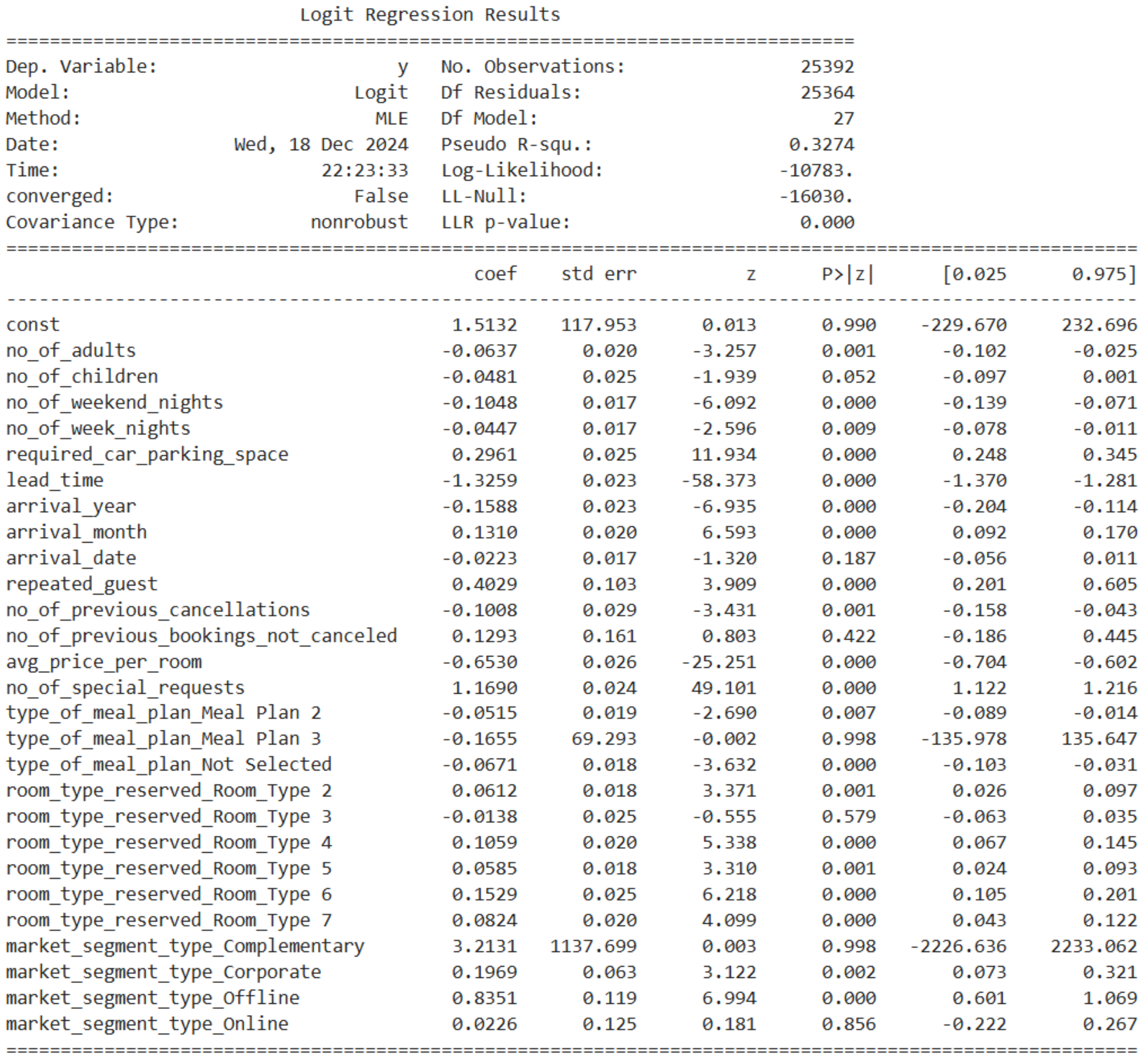
Model Evaluation Criteria

- **Accuracy**
  - Definition: Measures the proportion of correct predictions (both True Positives and True Negatives).
  - Business Implications: Accuracy is useful only if the dataset is balanced (equal distribution of cancelled and not-cancelled bookings), which is not in our case.
  - Verdict: **Accuracy is not the best metric for this problem**, especially if the dataset is imbalanced.
- **Precision**
  - Definition: Measures the proportion of correctly predicted "Not Cancelled" bookings out of all bookings predicted as "Not Cancelled".
  - Precision = TP / (TP + FP)
  - Business Implications: A model with high precision will result in fewer False Positives (fewer unnecessary preparations). High precision ensures the hotel does not waste resources preparing for guests who actually cancelled (False Positives).
  - Precision is more important when the cost of False Positives (FP) (e.g., wasted resources) is critical.
  - Verdict: **Precision is useful but less critical than Recall in this problem** because wasted resources (FP) have lower long-term damage than poor customer experience from overbooking (FN).
- **Recall**
  - Definition: Measures the proportion of correctly predicted "Not Cancelled" bookings out of all actual "Not Cancelled" bookings.
  - Recall = TP / (TP + FN)
  - Business Implications: High recall ensures the hotel accurately identifies most of the bookings that will not be cancelled. A model with high recall will result in fewer False Negatives (FN) (fewer missed bookings), reducing the risk of overbooking.
  - Missing a genuine booking (False Negative) can lead to significant business harm (Dissatisfied customers due to lack of room, refunds, compensation, and reputational damage).
  - Recall is critical in this problem because minimizing False Negatives (overbooking) is vital for business success.
  - Verdict: **Recall is the most important metric for the hotel booking problem**.
- **F1-Score**
  - Definition: The harmonic means of Precision and Recall, balancing the two.
  - F1 = 2 x (Precision x Recall) / (Precision + Recall)
  - Business Implications: F1-Score balances Precision and Recall. It's useful when you need to ensure both (a) Avoiding wasted resources (Precision) (b) Avoiding overbooking and dissatisfied customers (Recall).
  - When to Use F1-Score: When you want a trade-off between minimizing wasted resources (FP) and avoiding overbooking (FN).
  - Verdict: **F1-Score is a good secondary metric** but **Recall remains** the **primary metric to prioritize**.
- **Final Metric Verdict**
  - **Primary Metric: Recall:** Focus on maximizing Recall to minimize False Negatives (FN) and avoid overbooking situations.
  - **Secondary Metric: F1:** Use F1-Score to ensure a balance between Recall and Precision.

Logistic Regression Model

Build Model

- Train Dataset (with Intercept) is passed through the Logit Function.
- Below is the summary of the model output: -

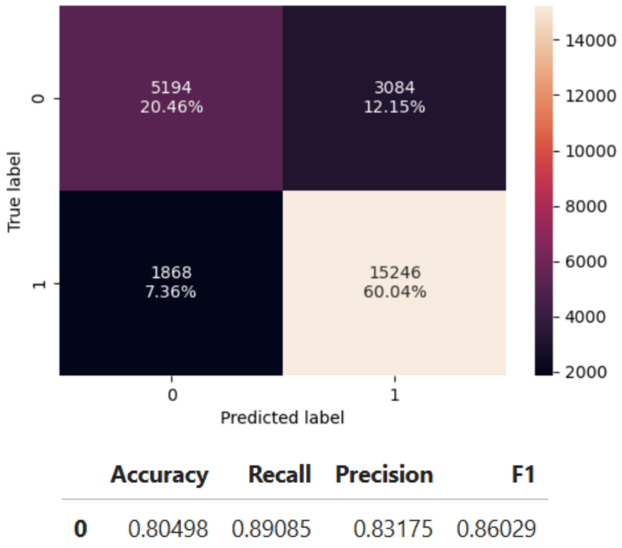


```
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                25392
Model:                          Logit   Df Residuals:                    25364
Method:                           MLE   Df Model:                           27
Date:                Wed, 18 Dec 2024   Pseudo R-squ.:                  0.3274
Time:                        22:23:33   Log-Likelihood:                -10783.
converged:                      False   LL-Null:                       -16030.
Covariance Type:            nonrobust   LLR p-value:                     0.000
========================================================================================================
                                          coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------
const                                   1.5132    117.953      0.013      0.990    -229.670     232.696
no_of_adults                           -0.0637      0.020     -3.257      0.001      -0.102      -0.025
no_of_children                         -0.0481      0.025     -1.939      0.052      -0.097       0.001
no_of_weekend_nights                   -0.1048      0.017     -6.092      0.000      -0.139      -0.071
no_of_week_nights                      -0.0447      0.017     -2.596      0.009      -0.078      -0.011
required_car_parking_space              0.2961      0.025     11.934      0.000       0.248       0.345
lead_time                              -1.3259      0.023    -58.373      0.000      -1.370      -1.281
arrival_year                           -0.1588      0.023     -6.935      0.000      -0.204      -0.114
arrival_month                           0.1310      0.020      6.593      0.000       0.092       0.170
arrival_date                           -0.0223      0.017     -1.320      0.187      -0.056       0.011
repeated_guest                          0.4029      0.103      3.909      0.000       0.201       0.605
no_of_previous_cancellations           -0.1008      0.029     -3.431      0.001      -0.158      -0.043
no_of_previous_bookings_not_canceled    0.1293      0.161      0.803      0.422      -0.186       0.445
avg_price_per_room                     -0.6530      0.026    -25.251      0.000      -0.704      -0.602
no_of_special_requests                  1.1690      0.024     49.101      0.000       1.122       1.216
type_of_meal_plan_Meal Plan 2          -0.0515      0.019     -2.690      0.007      -0.089      -0.014
type_of_meal_plan_Meal Plan 3          -0.1655     69.293     -0.002      0.998    -135.978     135.647
type_of_meal_plan_Not Selected         -0.0671      0.018     -3.632      0.000      -0.103      -0.031
room_type_reserved_Room_Type 2          0.0612      0.018      3.371      0.001       0.026       0.097
room_type_reserved_Room_Type 3         -0.0138      0.025     -0.555      0.579      -0.063       0.035
room_type_reserved_Room_Type 4          0.1059      0.020      5.338      0.000       0.067       0.145
room_type_reserved_Room_Type 5          0.0585      0.018      3.310      0.001       0.024       0.093
room_type_reserved_Room_Type 6          0.1529      0.025      6.218      0.000       0.105       0.201
room_type_reserved_Room_Type 7          0.0824      0.020      4.099      0.000       0.043       0.122
market_segment_type_Complementary       3.2131   1137.699      0.003      0.998   -2226.636    2233.062
market_segment_type_Corporate           0.1969      0.063      3.122      0.002       0.073       0.321
market_segment_type_Offline             0.8351      0.119      6.994      0.000       0.601       1.069
market_segment_type_Online              0.0226      0.125      0.181      0.856      -0.222       0.267
========================================================================================================
```

Figure 23: Logistic Regression Summary

Checking Model Performance

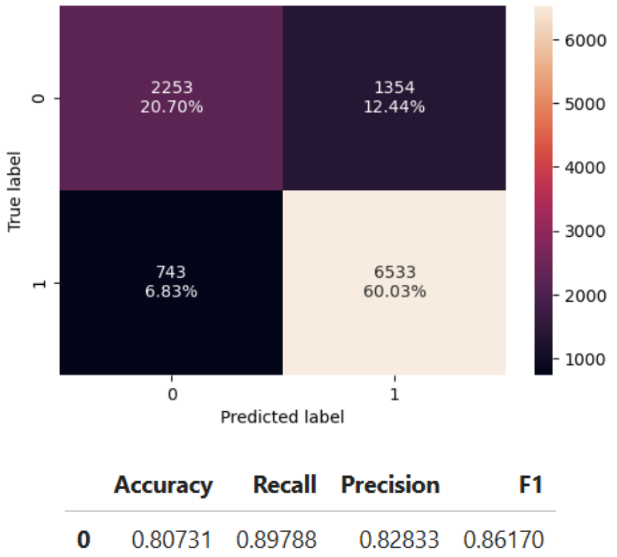
- Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -



	Accuracy	Recall	Precision	F1
0	0.80498	0.89085	0.83175	0.86029

Figure 24: Logistic Regression: Confusion Matrix & Metric Performance on Training Dataset

- Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -



	Accuracy	Recall	Precision	F1
0	0.80731	0.89788	0.82833	0.86170

Figure 25: Logistic Regression: Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**

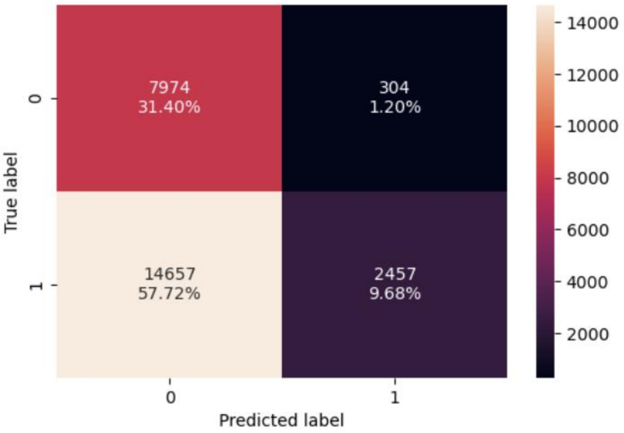
- Model Generalization: The metrics for both the training and testing datasets are very similar, indicating that the logistic regression model generalizes well and is not overfitting.
- Strengths:
  - ✓ High recall (around 89%) suggests that the model is very good at identifying positive cases (class 1).
  - ✓ Balanced F1 score (around 86%) indicates that the model effectively balances precision and recall.
- Weaknesses:
  - ✓ False positives and false negatives are not negligible, indicating room for improvement in distinguishing between the two classes.
- This model cannot be trusted fully as we have not dealt with multicollinearity yet & removed non-significant p-values.

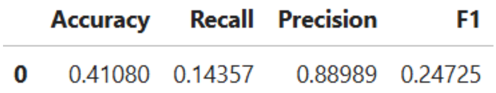
Naive- Bayes Classifier

Build Model & Check Model Performance

- Train Dataset is passed through the GaussianNB Function.

- Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -

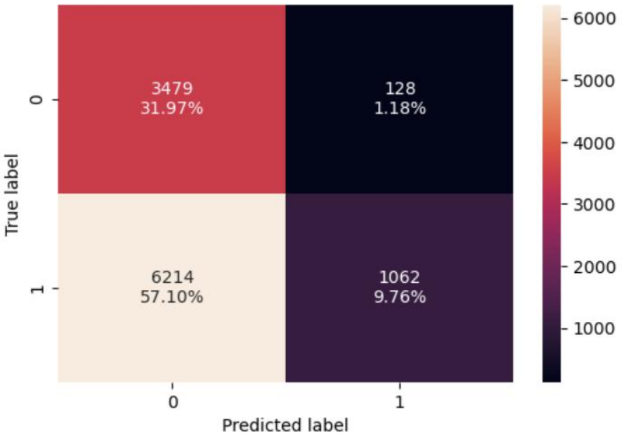


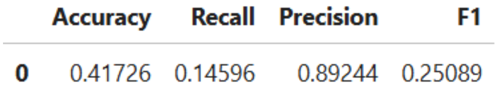


	Accuracy	Recall	Precision	F1
0	0.41080	0.14357	0.88989	0.24725

Figure 26: Naive - Bayes Classifier: Confusion Matrix & Metric Performance on Training Dataset

- Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -





	Accuracy	Recall	Precision	F1
0	0.41726	0.14596	0.89244	0.25089

Figure 27: Naive - Bayes Classifier: Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**

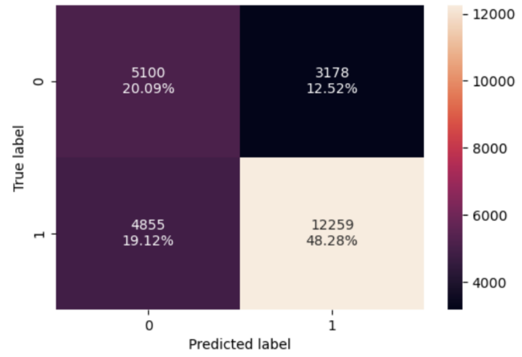
- Model Generalization: The metrics for both training and testing datasets are very similar, indicating that the Naive Bayes model generalizes consistently across datasets. However, the overall performance is poor, as it fails to effectively separate cancellations from non-cancellations.
- Strengths:
  - ✓ High Precision: The model achieves high precision (~89%), meaning that when it predicts a booking as a cancellation, it is correct most of the time.
- Weaknesses:
  - ✓ Low Recall: The model only captures ~14.5% of actual cancellations, leading to a large number of missed cancellations (false negatives).
  - ✓ Poor F1 Score: The low F1 score (~25%) highlights the imbalance between precision and recall, which is our secondary metric.
- While the Naive Bayes model generalizes consistently, **its poor recall and F1 score make it unsuitable** for solving the hotel booking cancellation problem effectively

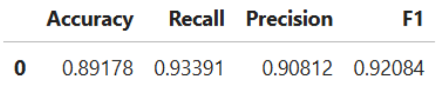
KNN Classifier

Build Model & Check Model Performance

- Scaled Training Dataset is passed through the KNeighborsClassifier Function with K = 5

- Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -

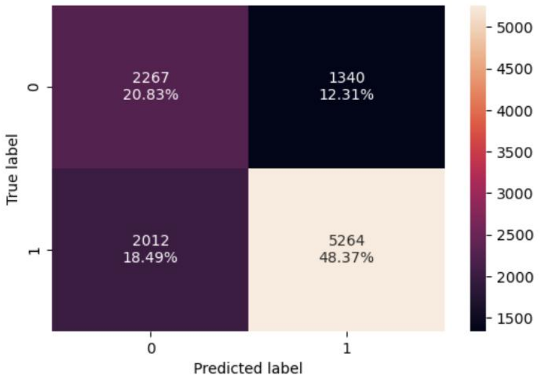


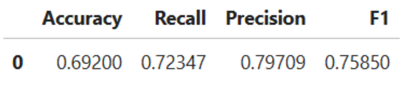


	Accuracy	Recall	Precision	F1
0	0.89178	0.93391	0.90812	0.92084

Figure 28: KNN Classifier: Confusion Matrix & Metric Performance on Training Dataset

- Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -





	Accuracy	Recall	Precision	F1
0	0.69200	0.72347	0.79709	0.75850

Figure 29: KNN Classifier: Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**

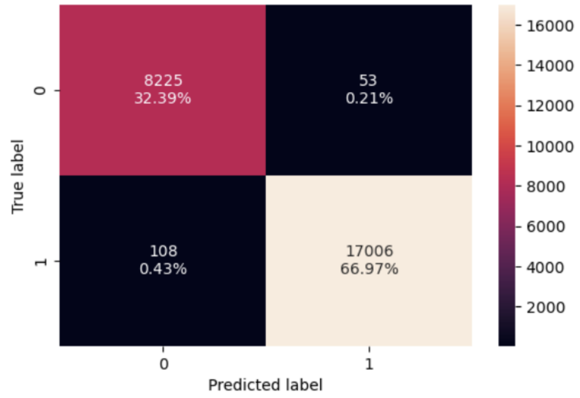
- Model Generalization: The KNN model performs significantly better on the training dataset than on the test dataset, suggesting potential overfitting. While the metrics for recall and F1 score on the training dataset are excellent, there is a notable drop in performance on the test dataset. This indicates that the model struggles to generalize to unseen data effectively.
- Strengths:
  - ✓ High Recall (Training Dataset): On the training dataset, recall is very high (93.39%), indicating that the model effectively identifies most cancellations (true positives).
  - ✓ High F1 (Training Dataset): On the training dataset, F1 is very high (90.81%), indicating that the model effectively balances Recall & Precision scores.
- Weaknesses:
  - ✓ Overfitting: The KNN model demonstrates much better performance on the training dataset than the test dataset, signalling overfitting. Poor F1 Score: The low F1 score (~25%) highlights the imbalance between precision and recall, which is our secondary metric.
  - ✓ Moderate Precision on Test Dataset: Precision (79.79%) on the test dataset indicates that a notable number of false positives (non-cancellations incorrectly predicted as cancellations) still exist, leading to potential operational inefficiencies for the hotel.
- The KNN model demonstrates good performance in Training dataset in recall and F1 score. However, the overfitting observed between training and test datasets limits its practicality for real-world implementation. While reasonably effective for cancellation prediction, the model's performance can be further optimized to better support hotel business operations.

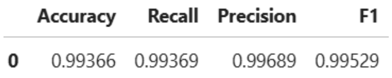
Decision Tree Classifier

Build Model & Check Model Performance

- Training Dataset is passed through the DecisionTreeClassifier Function with ‘random_state=42’ & ‘class_weight=balanced’ (by using class_weight='balanced', one can automatically adjust the weights for each class based on their frequency, helping to improve the model's performance on the minority class. This approach is particularly useful in scenarios where class imbalance is a significant issue)

- Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -

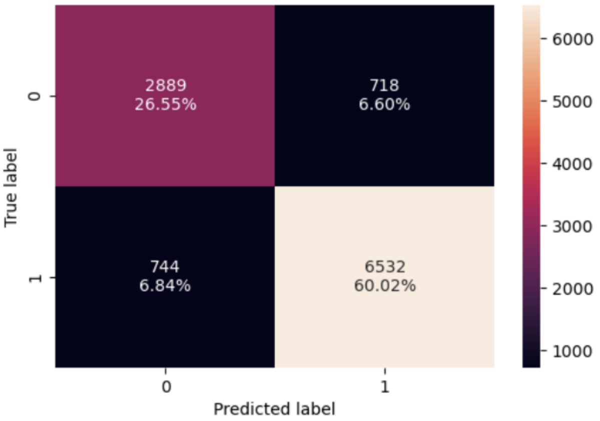


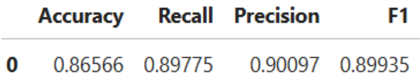


	Accuracy	Recall	Precision	F1
0	0.99366	0.99369	0.99689	0.99529

Figure 30: Decision Tree Classifier: Confusion Matrix & Metric Performance on Training Dataset

- Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -





	Accuracy	Recall	Precision	F1
0	0.86566	0.89775	0.90097	0.89935

Figure 31: Decision Tree Classifier: Confusion Matrix & Metric Performance on Test Dataset

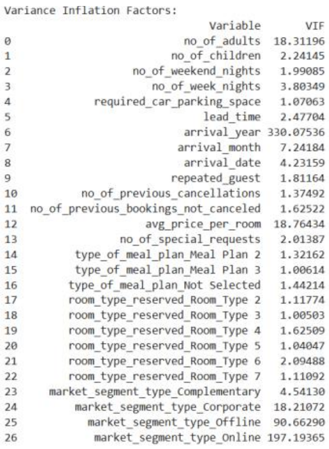
- **Summary of Performance: -**

- Model Generalization: The Decision Tree performs exceptionally well on the training dataset with near-perfect metrics (accuracy = 99.37%, F1 score = 99.53%), but its performance decreases slightly on the test dataset (accuracy = 86.57%, F1 score = 89.93%). This indicates slight overfitting, where the model memorizes the training data but struggles to generalize to unseen data.
- Strengths:
  - ✓ High Recall (Test Dataset): Recall of 89.77% on the test dataset suggests that the model identifies most of the actual cancellations, making it effective in anticipating revenue losses.
  - ✓ High F1 (Test Dataset): F1 of 89.76% on the test dataset suggests that the model effectively balances Recall & Precision scores.
- Weaknesses:
  - ✓ Slight Overfitting: Even though performance metrics are good for both Train & Test datasets, it demonstrates much better performance on the training dataset than the test dataset, signalling slight overfitting.
- The Decision Tree Classifier shows promise, particularly in identifying cancellations (high recall) and maintaining a balance between precision and recall (F1). However, its slight overfitting on the training data limits its reliability in real-world applications. By addressing overfitting by optimizing (pruning) the model, it could serve as a valuable tool for INN Hotels Group to optimize revenue management and mitigate losses due to cancellations.

Rubric Question 4: Model Performance Improvement

Logistic Regression Model – Tuning

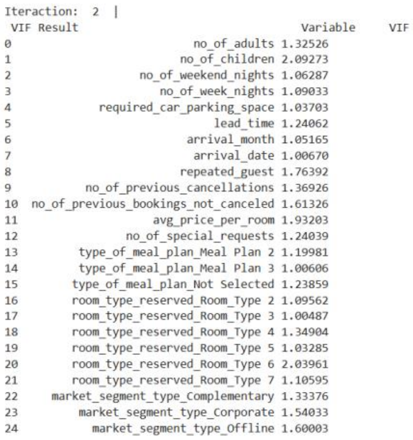
- **Dealing with Multicollinearity: -**
  - Multicollinearity occurs when predictor variables in a regression model are correlated. This correlation is a problem because predictor variables should be independent. If the correlation between variables is high, it can cause problems when we fit the model and interpret the results. When we have multicollinearity in the model, the coefficients that the model suggests are unreliable.
  - **Variance Inflation factor**: Variance inflation factors measure the inflation in the variances of the regression coefficients estimates due to collinearities that exist among the predictors. It is a measure of how much the variance of the estimated regression coefficient β_k is 'inflated' by the existence of correlation among the predictor variables in the model.
  - General Rule of Thumb while interpreting VIF: -
    - ✓ If VIF is 1, then there is no correlation among the kth predictor and the remaining predictor variables, and hence, the variance of β_k is not inflated at all.
    - ✓ If VIF exceeds 5, we say there is moderate VIF, and if it is 10 or exceeding 10, it shows signs of high multi-collinearity.
  - Below is the VIF for all independent variables: -



```
Variance Inflation Factors:
                                  Variable        VIF
0                             no_of_adults   18.31196
1                           no_of_children    2.24145
2                       no_of_weekend_nights    1.99085
3                          no_of_week_nights    3.80349
4                 required_car_parking_space    1.07063
5                                  lead_time    2.47704
6                               arrival_year  330.07536
7                              arrival_month    7.24184
8                               arrival_date    4.23159
9                             repeated_guest    1.81164
10              no_of_previous_cancellations    1.37492
11  no_of_previous_bookings_not_canceled    1.62522
12                        avg_price_per_room   18.76434
13                    no_of_special_requests    2.01387
14             type_of_meal_plan_Meal Plan 2    1.32162
15             type_of_meal_plan_Meal Plan 3    1.00614
16            type_of_meal_plan_Not Selected    1.44214
17            room_type_reserved_Room_Type 2    1.11774
18            room_type_reserved_Room_Type 3    1.00503
19            room_type_reserved_Room_Type 4    1.62509
20            room_type_reserved_Room_Type 5    1.04047
21            room_type_reserved_Room_Type 6    2.09488
22            room_type_reserved_Room_Type 7    1.11092
23       market_segment_type_Complementary    4.54130
24           market_segment_type_Corporate   18.21072
25             market_segment_type_Offline   90.66290
26              market_segment_type_Online  197.19365
```

Figure 32: VIF for Independent Variables (pre-Multicollinearity fix)

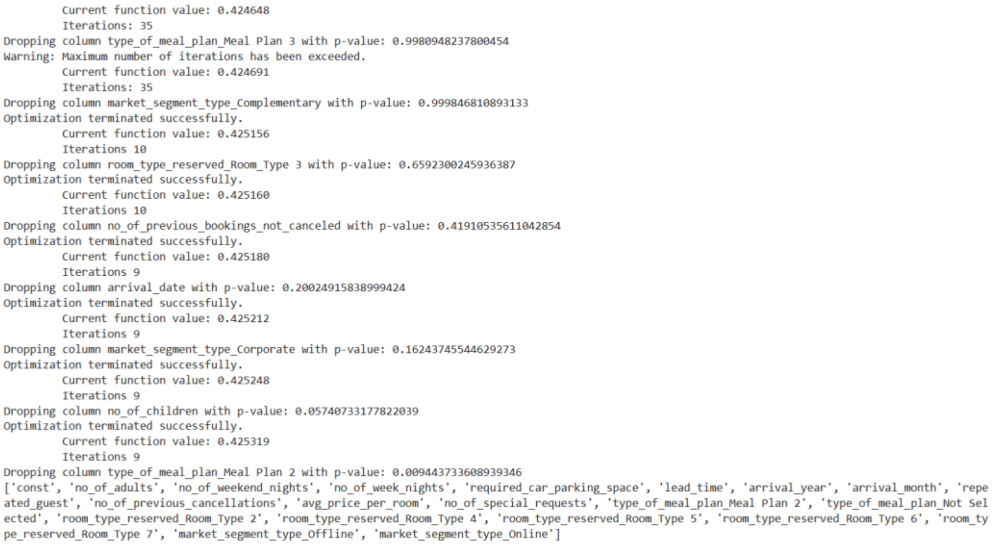
- Following steps to be taken to fix Multicollinearity (if any): -
  - ✓ In case any of the VIF was greater than 5, we follow the below steps: -
  - ✓ Drop every column one by one that has a VIF score greater than 5.
  - ✓ Check the VIF scores again.
  - ✓ Continue till you get all VIF scores under 5.
- Below is the final VIF Table after running the above algorithm, that ran for 2 iterations, removing 2 factors: -



```
Iteraction:  2  |
 VIF Result                                 Variable     VIF
0                           no_of_adults 1.32526
1                         no_of_children 2.09273
2                   no_of_weekend_nights 1.06287
3                      no_of_week_nights 1.09033
4             required_car_parking_space 1.03703
5                              lead_time 1.24062
6                          arrival_month 1.05165
7                           arrival_date 1.00670
8                         repeated_guest 1.76392
9           no_of_previous_cancellations 1.36926
10  no_of_previous_bookings_not_canceled 1.61326
11                    avg_price_per_room 1.93203
12                no_of_special_requests 1.24039
13         type_of_meal_plan_Meal Plan 2 1.19981
14         type_of_meal_plan_Meal Plan 3 1.00606
15        type_of_meal_plan_Not Selected 1.23859
16        room_type_reserved_Room_Type 2 1.09562
17        room_type_reserved_Room_Type 3 1.00487
18        room_type_reserved_Room_Type 4 1.34904
19        room_type_reserved_Room_Type 5 1.03285
20        room_type_reserved_Room_Type 6 2.03961
21        room_type_reserved_Room_Type 7 1.10595
22     market_segment_type_Complementary 1.33376
23         market_segment_type_Corporate 1.54033
24           market_segment_type_Offline 1.60003
```

Figure 33: VIF for Independent Variables (post-Multicollinearity fix)

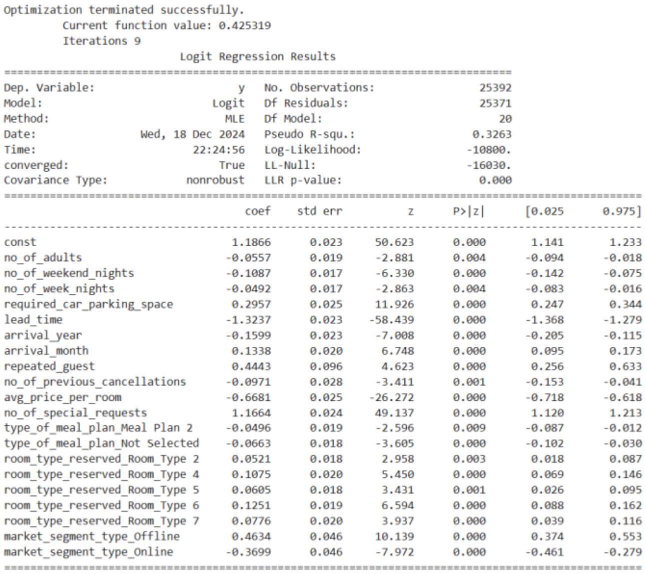
- **Dropping High p-value Variables**-
  - We will drop the predictor variables having a p-value greater than 0.05 as they do not significantly impact the target variable.
  - But sometimes p-values change after dropping a variable. So, we won't drop all variables at once.
  - Instead, we will do the following:
    1. Build a model, check the p-values of the variables, and drop the column with the highest p-value.
    2. Create a new model without the dropped feature, check the p-values of the variables, and drop the column with the highest p-value.
    3. Repeat the above two steps till there are no columns with p-value > 0.05.
  - By following the above steps, we ran multiple iterations to arrive at the final model. Below is the summary of each iteration showing which variables with what p-values were removed & the final variables which are significant: -



```
         Current function value: 0.424648
         Iterations: 35
Dropping column type_of_meal_plan_Meal Plan 3 with p-value: 0.9980948237800454
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.424691
         Iterations: 35
Dropping column market_segment_type_Complementary with p-value: 0.999846810893133
Optimization terminated successfully.
         Current function value: 0.425156
         Iterations 10
Dropping column room_type_reserved_Room_Type 3 with p-value: 0.6592300245936387
Optimization terminated successfully.
         Current function value: 0.425160
         Iterations 10
Dropping column no_of_previous_bookings_not_canceled with p-value: 0.41910535611042854
Optimization terminated successfully.
         Current function value: 0.425180
         Iterations 9
Dropping column arrival_date with p-value: 0.20024915838999424
Optimization terminated successfully.
         Current function value: 0.425212
         Iterations 9
Dropping column market_segment_type_Corporate with p-value: 0.16243745544629273
Optimization terminated successfully.
         Current function value: 0.425248
         Iterations 9
Dropping column no_of_children with p-value: 0.05740733177822039
Optimization terminated successfully.
         Current function value: 0.425319
         Iterations 9
Dropping column type_of_meal_plan_Meal Plan 2 with p-value: 0.009443733608939346
['const', 'no_of_adults', 'no_of_weekend_nights', 'no_of_week_nights', 'required_car_parking_space', 'lead_time', 'arrival_year', 'arrival_month', 'repe
ated_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 Sel
ected', '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_ty
pe_reserved_Room_Type 7', 'market_segment_type_Offline', 'market_segment_type_Online']
```

Figure 34: Features Removed with High p-value

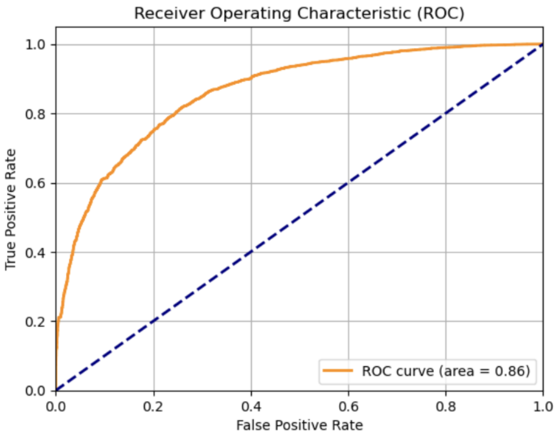
- Building the Logistic Regression Model again & displaying the summary: -

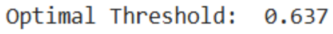


```
Optimization terminated successfully.
         Current function value: 0.425319
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                25392
Model:                          Logit   Df Residuals:                    25371
Method:                           MLE   Df Model:                           20
Date:                Wed, 18 Dec 2024   Pseudo R-squ.:                  0.3263
Time:                        22:24:56   Log-Likelihood:                -10800.
converged:                       True   LL-Null:                       -16030.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==================================================================================================
                                     coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------------
const                              1.1866      0.023     50.623      0.000       1.141       1.233
no_of_adults                      -0.0557      0.019     -2.881      0.004      -0.094      -0.018
no_of_weekend_nights              -0.1087      0.017     -6.330      0.000      -0.142      -0.075
no_of_week_nights                 -0.0492      0.017     -2.863      0.004      -0.083      -0.016
required_car_parking_space         0.2957      0.025     11.926      0.000       0.247       0.344
lead_time                         -1.3237      0.023    -58.439      0.000      -1.368      -1.279
arrival_year                      -0.1599      0.023     -7.008      0.000      -0.205      -0.115
arrival_month                      0.1338      0.020      6.748      0.000       0.095       0.173
repeated_guest                     0.4443      0.096      4.623      0.000       0.256       0.633
no_of_previous_cancellations      -0.0971      0.028     -3.411      0.001      -0.153      -0.041
avg_price_per_room                -0.6681      0.025    -26.272      0.000      -0.718      -0.618
no_of_special_requests             1.1664      0.024     49.137      0.000       1.120       1.213
type_of_meal_plan_Meal Plan 2     -0.0496      0.019     -2.596      0.009      -0.087      -0.012
type_of_meal_plan_Not Selected    -0.0663      0.018     -3.605      0.000      -0.102      -0.030
room_type_reserved_Room_Type 2     0.0521      0.018      2.958      0.003       0.018       0.087
room_type_reserved_Room_Type 4     0.1075      0.020      5.450      0.000       0.069       0.146
room_type_reserved_Room_Type 5     0.0605      0.018      3.431      0.001       0.026       0.095
room_type_reserved_Room_Type 6     0.1251      0.019      6.594      0.000       0.088       0.162
room_type_reserved_Room_Type 7     0.0776      0.020      3.937      0.000       0.039       0.116
market_segment_type_Offline        0.4634      0.046     10.139      0.000       0.374       0.553
market_segment_type_Online        -0.3699      0.046     -7.972      0.000      -0.461      -0.279
==================================================================================================
```

Figure 35: Logistic Regression Summary, post, Multicollinearity & P-value check

- Determining the Optimal Threshold using ROC Curve
  - The ROC curve illustrates the trade-off between the True Positive Rate (TPR) (Recall) and the False Positive Rate (FPR) for a binary classifier across various threshold values. The diagonal blue line represents a random classifier, with an area under the curve (AUC) of 0.5. A perfect classifier would achieve an AUC of 1.0.

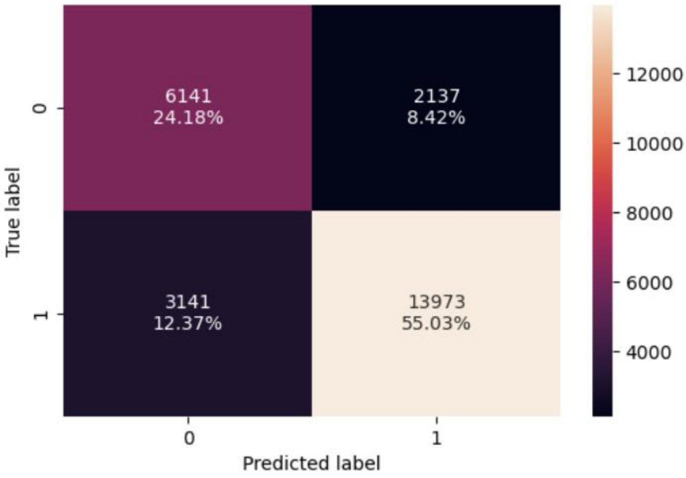


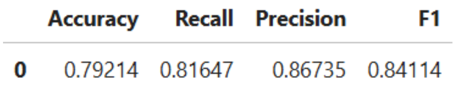


```
Optimal Threshold:  0.637
```

Figure 36: ROC Curve & Optimal Threshold Value

- Checking Model Performance post running the model with Optimal Threshold Value: -
  - Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -

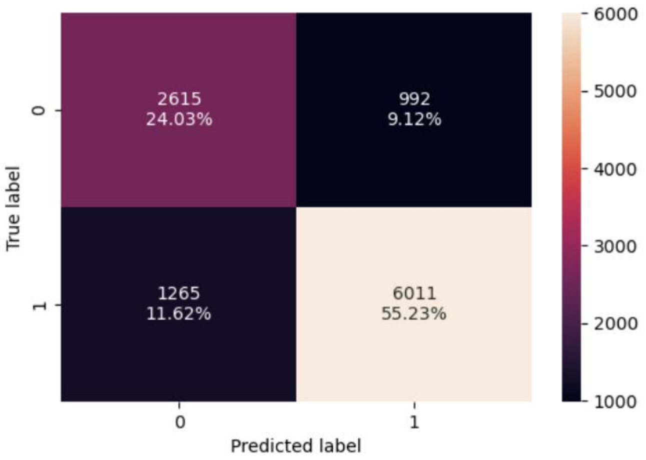


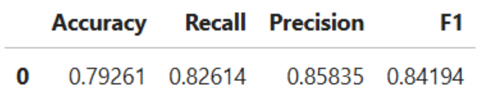


	Accuracy	Recall	Precision	F1
0	0.79214	0.81647	0.86735	0.84114

Figure 37: Logistic Regression: Confusion Matrix & Metric Performance on Training Dataset

- Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -





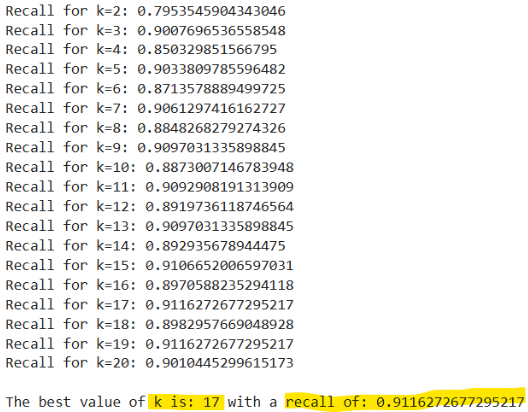
	Accuracy	Recall	Precision	F1
0	0.79261	0.82614	0.85835	0.84194

Figure 38: Logistic Regression: Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**
  - ✓ Model Generalization: The Logistic Regression model demonstrates consistent performance across training and test datasets, with similar metrics for accuracy, recall, precision, and F1 score. This indicates that the model generalizes well and is not overfitting.
  - ✓ Strengths:
    - High Recall | Training Recall: 81.65% | Test Recall: 82.61%: The model effectively identifies most cancellations (true positives), ensuring that the hotel can anticipate potential revenue losses.
    - High F1 | Training F1: 84.11% | F1 Recall: 84.19%: This indicates a good balance between precision and recall.
  - ✓ Logistic Regression provides a balanced and interpretable model for predicting hotel booking cancellations. Its consistent performance on both datasets ensures reliability, making it a strong candidate for deployment. By fine-tuning the decision threshold and leveraging the model's predictions, the INN Hotels Group can better anticipate cancellations, optimize resources, and improve profitability.

KNN Classifier – Tuning

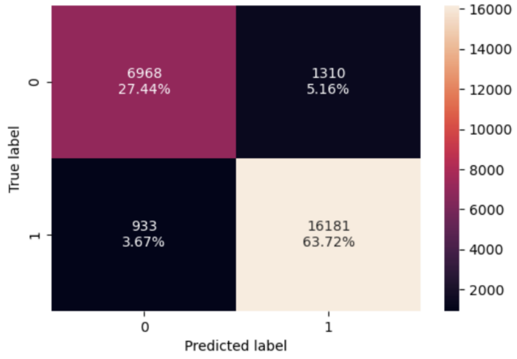
- We will run an algorithm to find the best Recall for all values of K ranging from 2 to 20. Below is the output of the K-value for best Recall: -

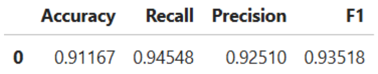


```
Recall for k=2: 0.7953545904343046
Recall for k=3: 0.9007696536558548
Recall for k=4: 0.850329851566795
Recall for k=5: 0.9033809785596482
Recall for k=6: 0.8713578889499725
Recall for k=7: 0.9061297416162727
Recall for k=8: 0.8848268279274326
Recall for k=9: 0.9097031335898845
Recall for k=10: 0.8873007146783948
Recall for k=11: 0.9092908191313909
Recall for k=12: 0.8919736118746564
Recall for k=13: 0.9097031335898845
Recall for k=14: 0.892935678944475
Recall for k=15: 0.9106652006597031
Recall for k=16: 0.8970588235294118
Recall for k=17: 0.9116272677295217
Recall for k=18: 0.8982957669048928
Recall for k=19: 0.9116272677295217
Recall for k=20: 0.9010445299615173

The best value of k is: 17 with a recall of: 0.9116272677295217
```

Figure 39: Best K-value with corresponding Recall-metric

- Checking Model Performance post running the model with 17 as K-value: -
  - Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -

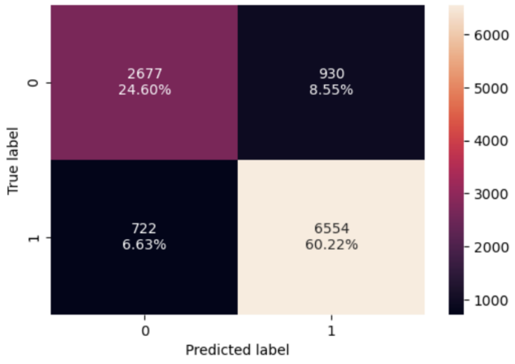


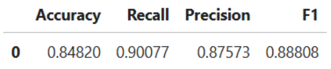


	Accuracy	Recall	Precision	F1
0	0.91167	0.94548	0.92510	0.93518

Figure 40: KNN Classifier: Confusion Matrix & Metric Performance on Training Dataset

  - Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -





	Accuracy	Recall	Precision	F1
0	0.84820	0.90077	0.87573	0.88808

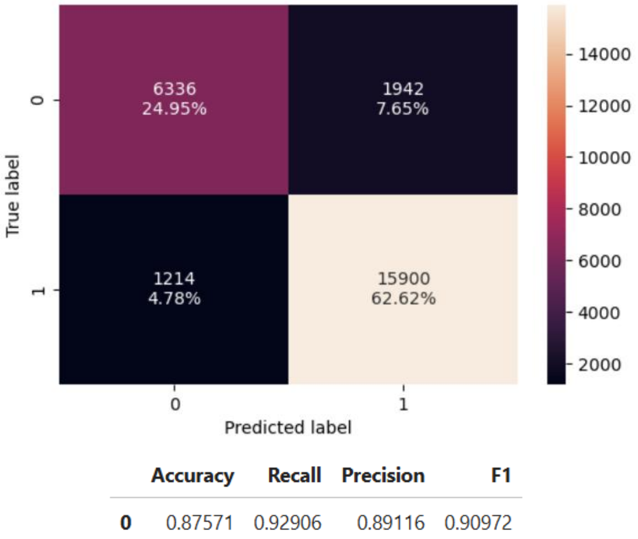
Figure 41: KNN Classifier: Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**
  - ✓ Model Generalization: The KNN model performs well on both the training and test datasets, but there is a noticeable gap in performance metrics, indicating slight overfitting.
  - ✓ Strengths:
    - High Recall | Training Recall: 94.55% | Test Recall: 90.07%: The model effectively identifies most cancellations, ensuring that the hotel is well-prepared to manage potential losses due to no-shows.
    - High F1 | Training F1: 93.52% | F1 Recall: 88.80%: This indicates a good balance between precision and recall.
  - ✓ Weaknesses:
    - Moderate Overfitting: The training metrics are higher than the test metrics, indicating the model slightly overfits the training data.
  - ✓ The KNN model with k=17 shows good performance in Recall & F1 scores. However, there is moderate overfitting in the model. We shall compare with other models in the end to finalize the best model for this problem.

Decision Tree Classifier – Tuning

Pre-pruning the Tree

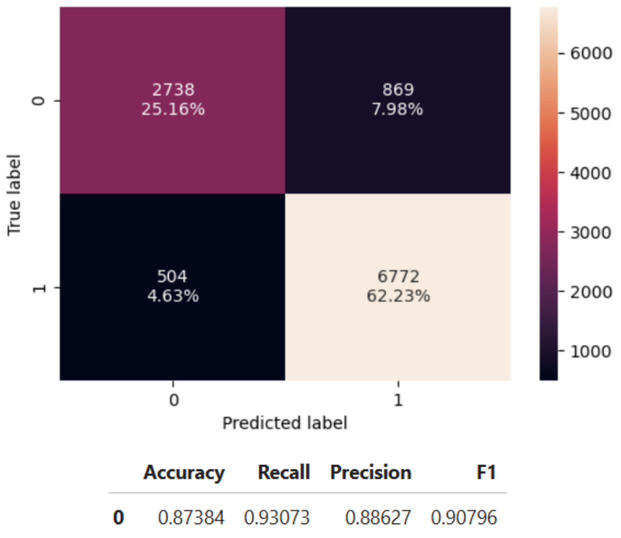
- Pre-pruning is a technique used in decision tree algorithms to limit the tree's growth and prevent overfitting by applying constraints during the training phase.
- Hyperparameters used to prune the tree: -
  - Maximum Depth: Set a limit on the maximum depth of the tree to restrict its complexity | Range of numbers between 5 to 13 with a step count of 2
  - Maximum Number of Leaf Nodes: Restrict the total number of leaf nodes in the tree. | Values – 10, 20, 30, 50, 70, 100
  - Minimum Samples to Split: minimum number of samples required to split a node | Values – 2, 6, 7, 10, 20, 30
- Using the GridSearchCV Function for hyperparameter tuning that helps us to find the best combination of hyperparameters for the model by systematically searching through a predefined grid of values. It evaluates model performance (in our case – Recall) for each combination of parameters using cross-validation.
- Using **optimal hyperparameters (from the above function) to build the Decision Tree Classifier Model**.
- **Checking Model Performance** post running the model again: -
  - Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -



	Accuracy	Recall	Precision	F1
0	0.87571	0.92906	0.89116	0.90972

Figure 42: Decision Tree Classifier (Pre-pruning): Confusion Matrix & Metric Performance on Training Dataset

- Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -



	Accuracy	Recall	Precision	F1
0	0.87384	0.93073	0.88627	0.90796

Figure 43: Decision Tree Classifier (Pre-pruning): Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**
  - ✓ Model Generalization: The pre-pruned decision tree performs well on both the training and test datasets, with consistent metrics indicating effective generalization.
  - ✓ Strengths:
    - High Recall | Training Recall: 92.91% | Test Recall: 93.07%: The model effectively identifies most cancellations (true positives), ensuring the hotel is prepared to handle potential revenue losses.
    - High F1 | Training F1: 90.97% | F1 Recall: 90.77%: This indicates a good balance between precision and recall.
- The **pre-pruned decision tree demonstrates excellent performance with balanced precision, recall, and generalization**. Its ability to effectively predict cancellations makes it a valuable tool for INN Hotels Group to minimize revenue loss, optimize room availability, and improve customer satisfaction. By leveraging this model in conjunction with dynamic pricing and proactive engagement strategies, the hotel can significantly enhance its operational efficiency and profitability.

▪ **Visualizing the Decision Tree**: -

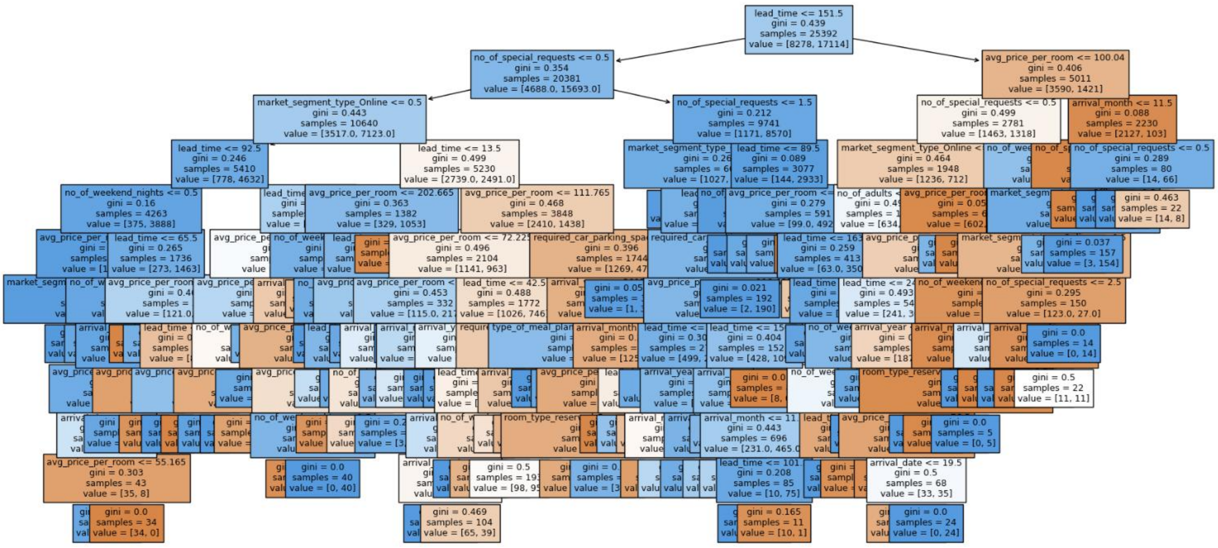


Figure 44: Visualization of pre-pruned Decision Tree

- **Analysing Feature Importance**: -

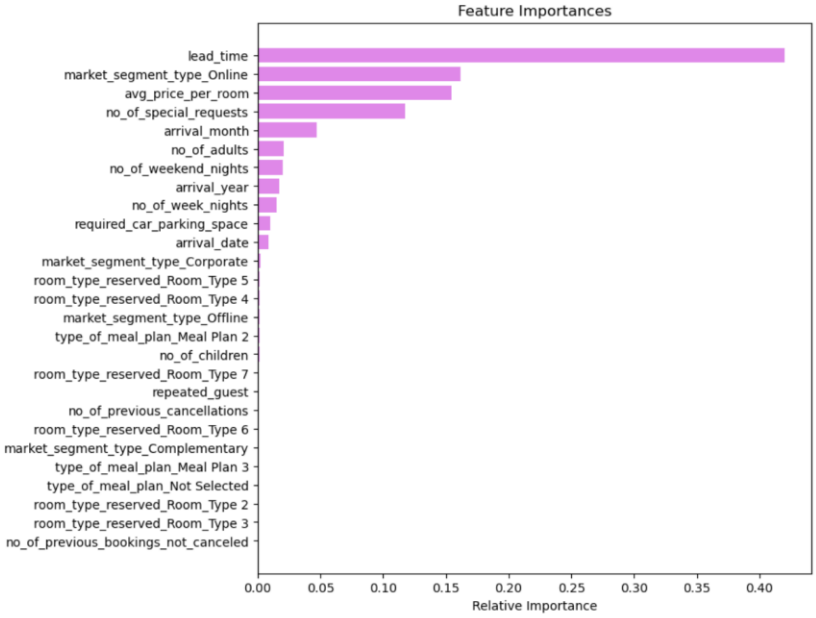
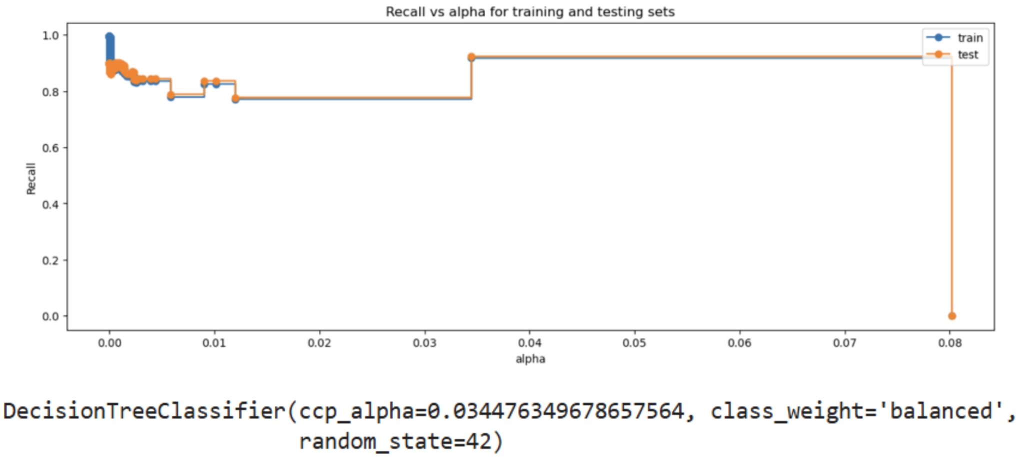


Figure 45: Feature Importances (Pre-pruned Decision Tree)

- Clearly, 'Lead-time' is the most significant feature that impacts booking cancelation, followed by 'market segment (Online)', 'Average Price of Room' & 'Special Requests', based on which the Business needs to vet upon & amend its marketing strategy.

Post-pruning the Tree

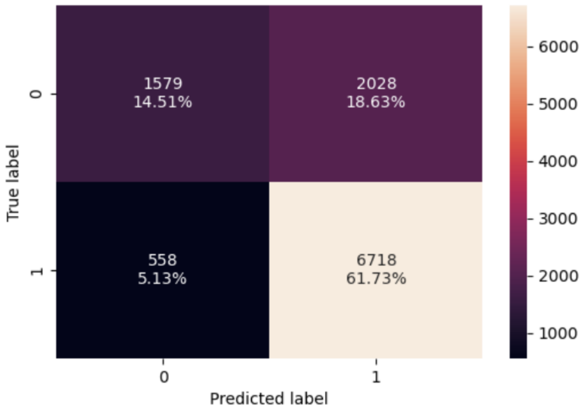
- Post-pruning (also known as "cost-complexity pruning") is a method to simplify a decision tree after it has been fully grown. This technique aims to remove branches that provide little predictive power, reducing overfitting and improving generalization.
- Steps to perform post-pruning: -
  1. Train the Full Tree: Start by training a fully grown decision tree without any constraints to allow it to overfit the data.
  2. Evaluate Cost-Complexity Pruning Path: Use cost_complexity_pruning_path to calculate the effective alpha (ccp_alpha – Relative error decrease per node) values that control the trade-off between tree complexity (number of leaf nodes) and its performance on the training dataset.
  3. Prune the Tree: Use the calculated ccp_alpha values to iteratively prune the tree and evaluate performance for each pruned version using cross-validation.
  4. Select the Optimal Tree: Choose the tree with the best balance of accuracy and generalization (e.g., best cross-validated performance on the test dataset).
- For each of the Decision Tree corresponding to ccp_alpha we compute Recall for both Training & Test datasets & finding the best ccp_alpha: -

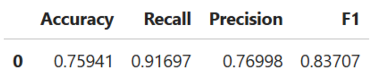


```
DecisionTreeClassifier(ccp_alpha=0.034476349678657564, class_weight='balanced',
                       random_state=42)
```

Figure 46: Optimal CCP_Alpha

- **Building the Decision Tree Classifier Model with ccp_alpha = 0.034**
- **Checking Model Performance** again: -
  - Below is the Confusion Matrix & Model Performance Metrics on Training dataset: -

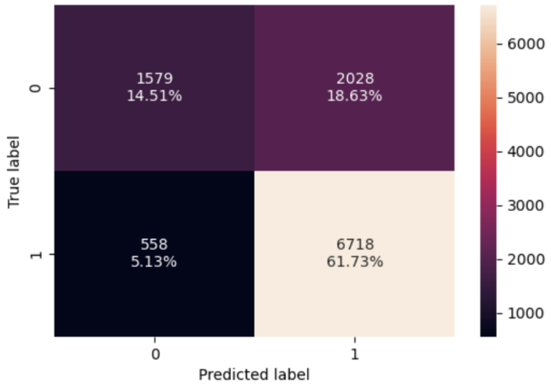


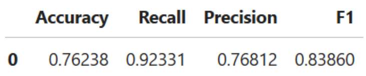


	Accuracy	Recall	Precision	F1
0	0.75941	0.91697	0.76998	0.83707

Figure 47: Decision Tree Classifier (Post-pruning): Confusion Matrix & Metric Performance on Train Dataset

  - Below is the Confusion Matrix & Model Performance Metrics on Test dataset: -





	Accuracy	Recall	Precision	F1
0	0.76238	0.92331	0.76812	0.83860

Figure 48: Decision Tree Classifier (Post-pruning): Confusion Matrix & Metric Performance on Test Dataset

- **Summary of Performance: -**
  - ✓ Model Generalization: The post-pruned decision tree shows similar performance on the training and test datasets, indicating effective generalization and reduced overfitting.
  - ✓ Strengths:
    - High Recall | Training Recall: 91.70% | Test Recall: 92.33%: The model effectively identifies most cancellations (true positives), ensuring the hotel is prepared to handle potential revenue losses.
    - High F1 | Training F1: 83.70% | F1 Recall: 83.86%: This indicates a good balance between precision and recall.
  - ✓ Weaknesses:
    - Moderate Precision | Training Precision: 77% | Test Recall: 76.81%: Precision is lower compared to recall, indicating a higher rate of false positives. This could result in unnecessary overbooking preparations.
  - ✓ The post-pruned decision tree offers a well-balanced and interpretable model for predicting hotel booking cancellations. Its high recall ensures that most cancellations are captured, allowing INN Hotels Group to plan effectively for cancellations and minimize revenue losses. However, we shall compare with other models as well to finalize the best model in the end.

- **Visualizing the Decision Tree: -**

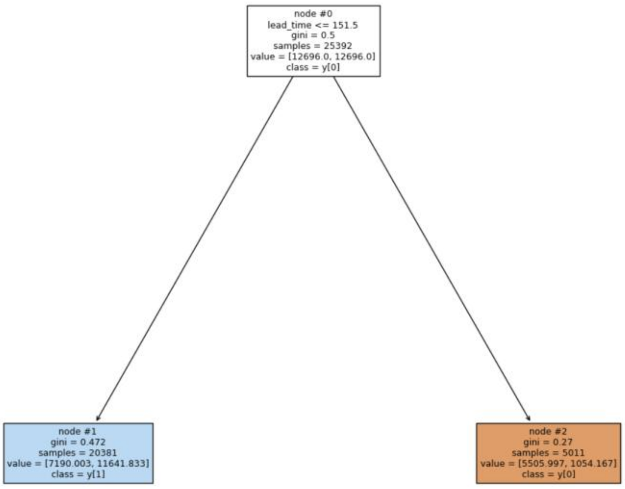


Figure 49: Visualization of post-pruned Decision Tree

- **Analysing Feature Importance: -**

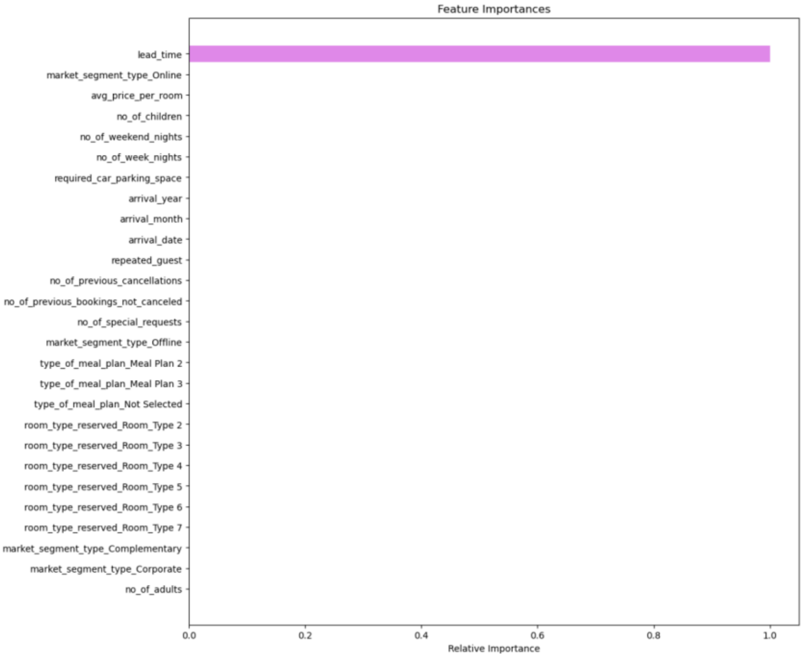


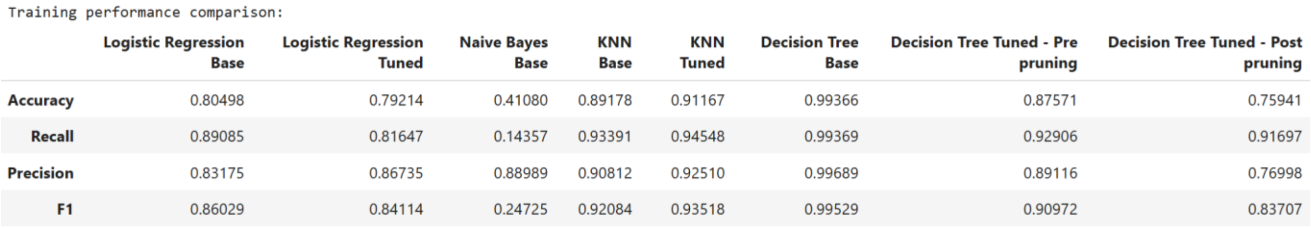
Figure 50: Feature Importances (Post-pruned Decision Tree)

- Clearly, ‘Lead-time’ is the only significant feature that impacts booking cancelation.

Rubric Question 5: Model Performance Comparison and Final Model Selection

Training Dataset Performance Comparison

- Below is the summary of performance comparison of Training dataset



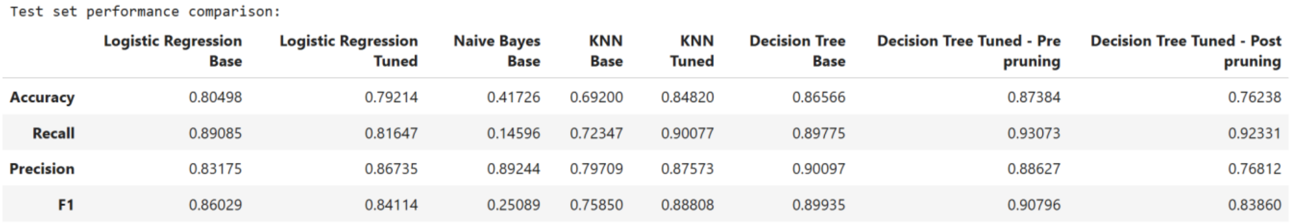
```
Training performance comparison:
```

	Logistic Regression Base	Logistic Regression Tuned	Naive Bayes Base	KNN Base	KNN Tuned	Decision Tree Base	Decision Tree Tuned - Pre pruning	Decision Tree Tuned - Post pruning
Accuracy	0.80498	0.79214	0.41080	0.89178	0.91167	0.99366	0.87571	0.75941
Recall	0.89085	0.81647	0.14357	0.93391	0.94548	0.99369	0.92906	0.91697
Precision	0.83175	0.86735	0.88989	0.90812	0.92510	0.99689	0.89116	0.76998
F1	0.86029	0.84114	0.24725	0.92084	0.93518	0.99529	0.90972	0.83707

Figure 51: Training Dataset Performance Comparison

Test Dataset Performance Comparison

- Below is the summary of performance comparison of Test dataset



```
Test set performance comparison:
```

	Logistic Regression Base	Logistic Regression Tuned	Naive Bayes Base	KNN Base	KNN Tuned	Decision Tree Base	Decision Tree Tuned - Pre pruning	Decision Tree Tuned - Post pruning
Accuracy	0.80498	0.79214	0.41726	0.69200	0.84820	0.86566	0.87384	0.76238
Recall	0.89085	0.81647	0.14596	0.72347	0.90077	0.89775	0.93073	0.92331
Precision	0.83175	0.86735	0.89244	0.79709	0.87573	0.90097	0.88627	0.76812
F1	0.86029	0.84114	0.25089	0.75850	0.88808	0.89935	0.90796	0.83860

Figure 52: Test Dataset Performance Comparison

Final Model Selection

- Below is the comparison of all models with their Strengths & Weakness: -

Model	Strengths	Weaknesses
Logistic Regression (Base and Tuned)	✓ Good balance between recall and precision, resulting in high F1 scores (~0.86 on both training and test datasets). ✓ Consistent generalization across datasets (minimal overfitting).	✓ Moderate recall (~81.65% after tuning) compared to other models, which means it misses more cancellations.
Naive Bayes (Base)	✓ High precision (~89%) indicates strong performance in avoiding false positives. ✓ Consistent generalization across datasets (minimal overfitting).	✓ Very low recall (~14.35%), leading to missed cancellations. This is unsuitable for the hotel problem where predicting cancellations is critical. ✓ Lowest F1 score (~0.25), making it the least effective model overall.
KNN (Base and Tuned)	✓ Tuned KNN achieves high recall (~94.55% on training and ~90.07% on test datasets), meaning it captures most cancellation.	✓ Slight overfitting in the tuned version.
Decision Tree (Base)	✓ Excellent recall (~99.37% on training and ~89.77% on test datasets).	✓ High overfitting
Decision Tree (Pre-Pruned)	✓ Well-balanced performance with a high recall (~93.07%) and precision (~88.63%) on the test dataset. ✓ High F1 score (~0.91 on test dataset) ✓ Minimal overfitting, making it reliable.	✓ Slight reduction in precision compared to the unpruned tree.
Decision Tree (Post-Pruned)	✓ Good recall (~92.33%) on the test dataset. ✓ Simplified and interpretable tree with less overfitting compared to the base version	✓ Lower F1 (our secondary performance metric) & Precision, resulting in more false positives compared to the pre-pruned version.

Table 2: Model Comparison

- **Best Model: Decision Tree (Pre-Pruned)**
  - There was a close contest between The Pre-pruned & post-pruned model, but we finalized the pre-pruned model. Please note following points leading us to this selection: -
    - ✓ Both models show similar performance on the training and test datasets, indicating **effective generalization** and reduced overfitting.
    - ✓ **Recall scores (primary metric) are marginally higher** in the pre-pruned model against the post-pruned model for both train & test datasets.
    - ✓ **FI scores (secondary metric) are higher** in the pre-pruned model against the post-pruned model for both train & test datasets.
    - ✓ **Other metric scores (Accuracy & Precision) are both significantly higher** in the pre-pruned model against the post-pruned model for both train & test datasets.
  - **Characteristics** of the Model: -
    - ✓ **High Recall**: Captures ~93.07% of actual cancellations, ensuring most cancellations are predicted correctly. This aligns with the hotel's need to proactively manage cancellations and minimize revenue loss.
    - ✓ **Balanced Metrics**: Precision (~88.63%) and F1 score (~0.91) ensure a balance between correctly predicting cancellations and minimizing false positives.
    - ✓ **Generalization:** Minimal performance gap between training and test datasets indicates reliable predictions on unseen data.

Rubric Question 6: Actionable Insights & Recommendations

- Now that we have selected our Final Model (Pre-pruned Decision Tree), let's analyse the feature importances (refer link1 & link2) in context to the business problem: -

Features	Importance	Insight	Business Implication/Recommendation
Lead Time (Most Important Feature)	~40%	✓ Lead time, or the number of days between booking and check-in, is the most influential factor in predicting cancellations. Longer lead times are often associated with higher cancellation probabilities, as customers are more likely to change plans over extended periods.	✓ Offer non-refundable rates or incentives (e.g., discounts or perks) for bookings with long lead times to reduce cancellations. ✓ Focus retention strategies (e.g., reminders, personalized offers) on bookings with high lead times.
Market Segment Type: Online	~15%	✓ Online bookings are a significant predictor of cancellations, likely due to the convenience of online platforms offering free or low-cost cancellations.	✓ Implement stricter cancellation policies for online bookings or require deposits to discourage cancellations. ✓ Analyse customer behaviour specific to online bookings to identify patterns leading to cancellations.
Average Price Per Room	~10%	✓ Higher room prices may contribute to cancellations, as customers could be price-sensitive and cancel if they find better deals elsewhere.	✓ Offer price guarantees or discounts for bookings at risk of cancellation. ✓ Promote bundled packages that offer better value, reducing the likelihood of cancellations.
Number of Special Requests	~8%	✓ Bookings with more special requests are less likely to cancel, as these customers are more committed to their plans.	✓ Use the number of special requests as a positive signal and prioritize customer service for such bookings. ✓ Encourage customers to make specific requests during booking to enhance engagement and reduce cancellations.
Arrival Month	~5%	✓ Seasonal trends influence cancellations, as bookings during peak months (Aug-Oct) or off-seasons may have different cancellation rates.	✓ Adjust pricing and policies based on seasonal trends. For example, stricter cancellation policies during peak months (Aug-Oct) and flexible options during off-season.
Number of Adults	~4%	✓ Bookings with a higher number of adults may be less likely to cancel, as such bookings are often for groups with stronger commitments.	✓ Encourage group bookings with tailored packages or discounts to minimize cancellation risk.

Table 3: Insights & Recommendations against Important Features

- **Strategic Actions for INN Hotels Group: -**
  1. **Dynamic Cancellation Policies**: Implement tiered cancellation policies based on lead time and market segment type. For example:
     - Longer lead times: Stricter refund terms.
     - Shorter lead times: Flexible cancellation policies.
  2. **Dynamic Pricing:** Adjust pricing or offer discounts to secure bookings with a high risk of cancellation.
  3. **Retention Strategies:**
     - Send personalized follow-up emails to bookings flagged as high risk of cancellation based on lead time and booking channel.
     - Offer incentives like discounts, room upgrades, or additional perks to customers with long lead times or high room prices.
  4. **Seasonal Adjustments**:
     - Leverage insights from the "arrival month" feature to adjust policies and pricing strategies during peak and off-peak seasons.
  5. **Promote Special Requests:**
     - Encourage customers to specify their preferences during booking, as higher engagement often reduces cancellation risk.
  6. **Overbooking Strategy**: Leverage the model's predictions to safely overbook rooms based on the likelihood of cancellations.

- **To Summarize** – The pre-pruned decision tree effectively identifies the most influential features for predicting cancellations, with lead time and online bookings standing out as critical drivers. By leveraging these insights, INN Hotels Group can proactively manage cancellations, optimize resource allocation, and enhance customer retention, ultimately improving profitability and operational efficiency.