DATA PREDICTION ON HOTEL BOOKING CANCELLATION

A FINAL REPORT

SUBMITTED BY

Mohammed Anaz

D K Tarun Venkatesh

Malini

Vishal Raj

Sangeetha Elangovan

In partial fulfilment for the award of the course

Of

DATA SCIENCE & ENGINEERING

GREAT LAKES INSTITUTE OF MANAGEMENT,

OMR, CHENNAI-44

NOVEMBER 2023

Capstone Project: Final Report

Batch details	DSE CHN Apr'23
Team members	Mohammed Anaz, D K Tarun Venkatesh, Malini, Vishal Raj, Sangeetha Elangovan
Domain of Project	Travel and Tourism
Proposed project title	Hotel Booking Cancellation Prediction
Group Number	2
Team Leader	Vishal Raj
Mentor Name	Ankush Bansal

Ankush Bansal

Signature of the Mentor

Vishal Raj

Signature of the Team Leader

Table of contents

- **Introduction**
  - Brief overview of the Cancellation Prediction

The goal of this machine learning project is to develop a predictive model that can accurately predict whether customers who book hotel rooms will show up for their reservation or cancel it at the last minute. By achieving this objective, we aim to assist hotels in optimizing their operations, improving resource allocation, and reducing revenue losses due to cancellations.

- Importance of hotel booking cancellation prediction

By studying the previous data of customer bookings and building a model using machine learning techniques to predict the actual footfall of customers who are booked for the Hotel stay. By implementing the Trained model, our aim is to create a predictive model with good accuracy to detect whether the booked customers are going to appear on the booked date.

Expected Benefits:

- Improved resource allocation for hotels, reducing operational costs.
- Enhanced customer experience by reducing booking cancellations.
- Increased revenue through optimized room management.
- Data-driven insights for hotel management to make informed decisions.

- Objectives of the analysis

Develop a Booking Cancellation prediction system that allows hotel owners to determine whether the guests who made the booking will come or not. Reduce the risk of knowing the cancellation at the last moment and be prepared with the prediction system.

- **Data Collection and Description**
    - Source of the dataset

Dataset Name	Hotel Booking Cancellation Prediction
Dataset Provider	The data is originally from the article Hotel Booking demand Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.
Data Website	Kaggle Link - https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand Article Link - https://www.sciencedirect.com/science/article/pii/S2352340918315191

- Data description and attributes

S.No	Variable Name	Description	Type
1	hotel	Resort hotel or city hotel	object
2	is_canceled	Indicates if the booking is cancelled or not	int64
3	Lead_time	No. Of days that elapsed between the entering date of the booking and arrival date	int64
4	Arrival_date_year	Year of arrival date	int64
5	Arrival_date_month	Month of arrival date	object
6	Arrival_date_week_number	Week number of the year of arrival date	int64
7	Arrival_date_day_of_month	Day of arrival date	int64
8	Stays_in_weekend_nights	No of weekend nights (saturday or sunday). The guest stayed or booked to stay at the hotel	int64
9	Stays_in_week_nights	No of weeknights (monday to friday). The guests stayed or booked to stay at the hotel	int64
10	adults	No of adults	int64
11	children	No of children	float64
12	babies	No of babies	int64
13	meal	Type of meal booked	object
14	country	Country of origin	object
15	Market_segment	Method of booking	object
16	Distribution_channel	Bookings distributed through	object
17	Is_repeated_guest	Is the booking guest has already visited or not	int64
18	Previous_cancellations	No of previous booking that are cancelled by the customer	int64
19	Previous_bookings_not_cancelled	No of previous booking that are not cancelled by the customer	int64

20	Reserved_room_type	Room type reserved by the customer at the time of booking	object
21	Assigned_room_type	Room type given to the customer	object
22	Booking_changes	No of changes made from the moment of booking initiated	int64
23	Deposit_type	Indicates wheather the customer made any deposits upon the booking	object
24	agent	Agent ID	float64
25	company	Company ID	float64
26	Days_in_waiting_list	No of days the booking was in WL before confirmed to customer	int64
27	Customer_type	Type of customer booked	object
28	adr	Average daily rate as defined by dividing the sum of all lodging transactions by the total number of staying nights	float64
29	Required_car_parking_spaces	Wheather customer required car parking	int64
30	Total_of_special_request	No of special request made by the customer	int64
31	Reservation_status	Reservation status of booking	object
32	Reservation_status_date	The date at which the reservation is made	object

- Categories in the dataset

Total Observations	119390
Variables	32
Number of Numeric columns	20
Number of Categorical columns	12

- Types of Variables

int64	16
float64	4
object	12

- Missing value counts

Variable Name	Number of missing values	Percentage of missing values
children	4	0.003
country	488	0.40
agent	16340	13.6
company	112593	94.3

- Redundant Column

The variables Is_canceled & Reservation_status Both the variables have same kind of information hence we are dropping Reservation_status.

- Classes in target variable

Is_canceled	Count of classes	Percentage of classes
0	75166	62.95
1	44224	37.04

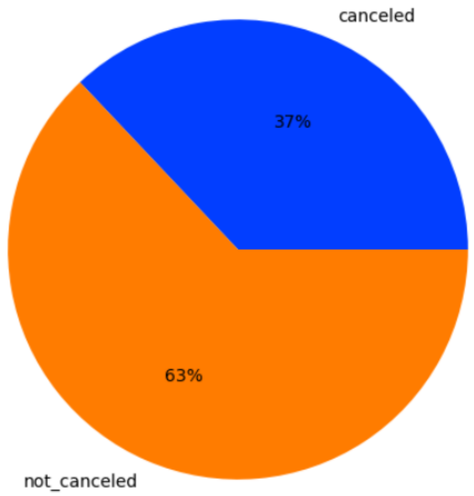
Here 0 represents no cancellation and 1 represents booking cancelled / No show

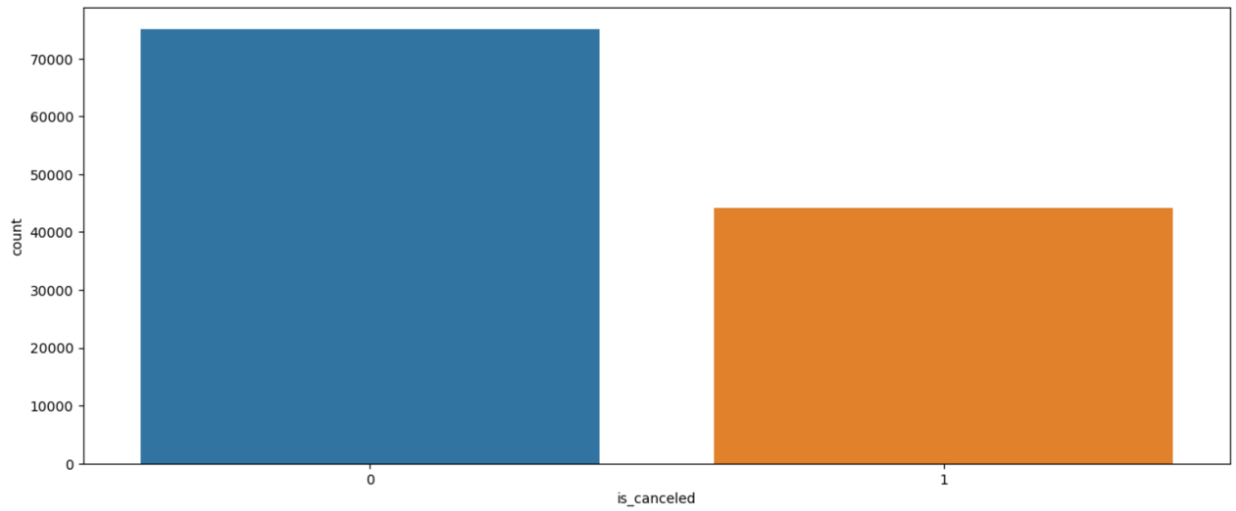
- **Exploratory Data Analysis (EDA)**

  - **Data visualization**

➢ **Univariate Analysis**

❖ ‘is_canceled’ variable



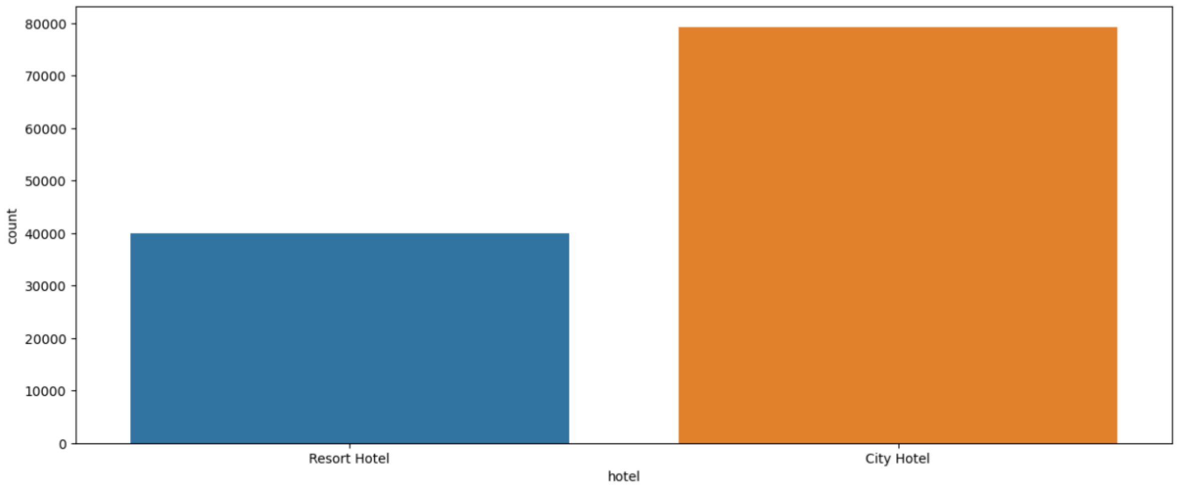


Inference –

Count plot for the variable shows that there are more number of people appearing at the hotel than that of people canceling their booking

There are nearly 45000 counts of booking cancellation present in the dataset

- ‘hotel’ Variable



Inference –

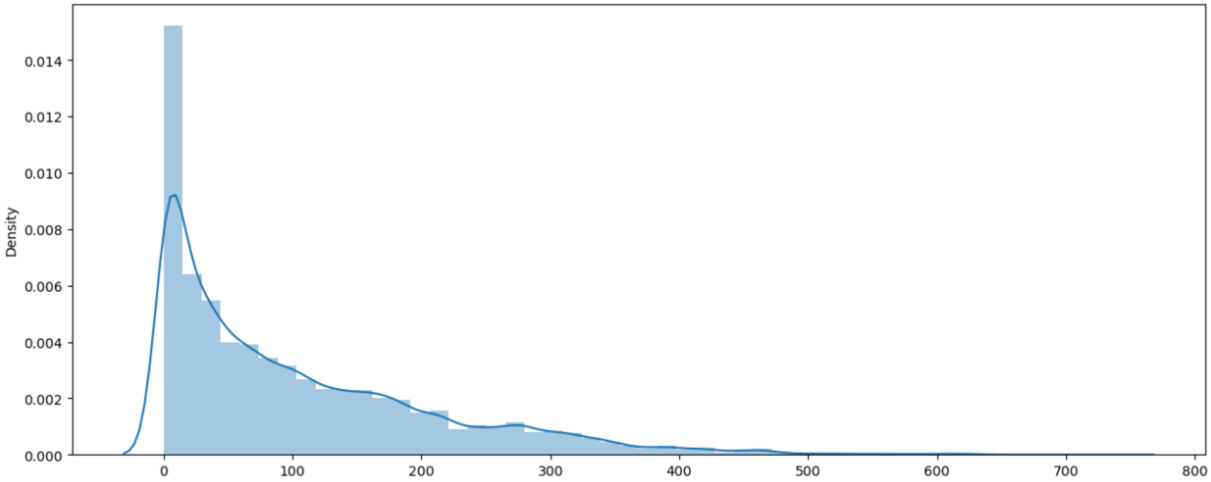
We can see that the variable 'hotel has two subclasses

66% of the hotels are 'city hotel'

33% of the hotels are 'resort hotel'

With the data we can see that there are double the amount of 'city hotel' than 'resort hotel'

- ‘lead_time’ Variable



Inference –

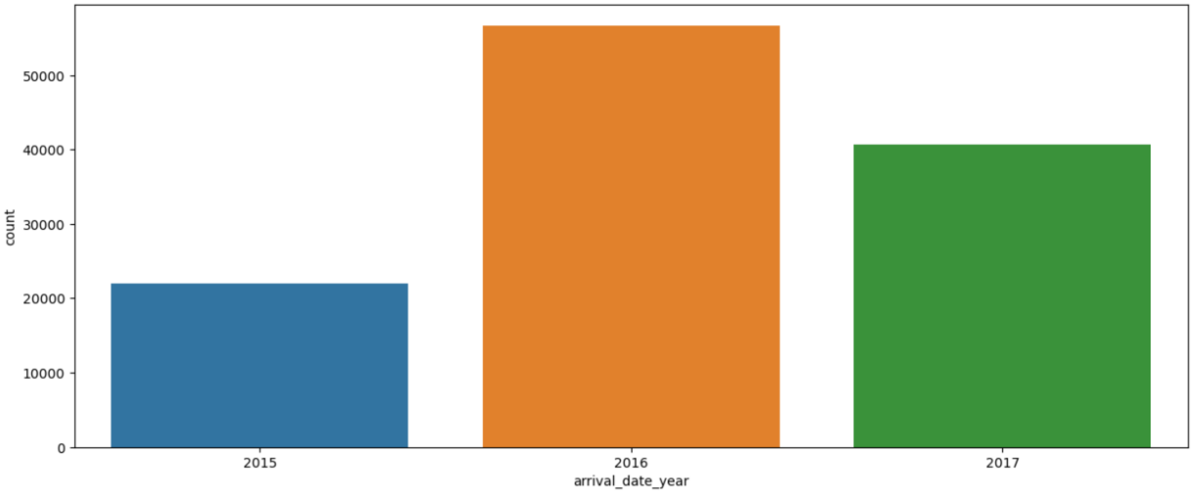
Number of days that elapsed between the entering date of the booking into the system and the arrival date

From the above graph we can see that most of the people arrive immediately after date of booking

People who book in advance are less

The data is highly right skewed

- ‘arrival_date_year’ Variable

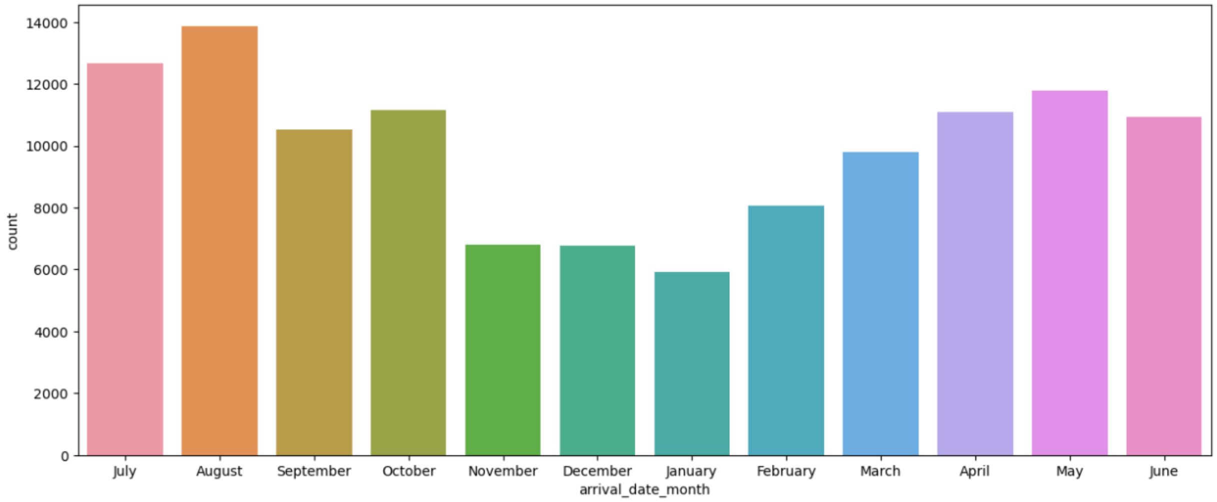


Inference –

We can see from the data that
34.0% of the data is from 2017

47.4% of the data is from 2016
18.4% of the data is from 2015
It is observed that more people have visited hotels in the year 2016

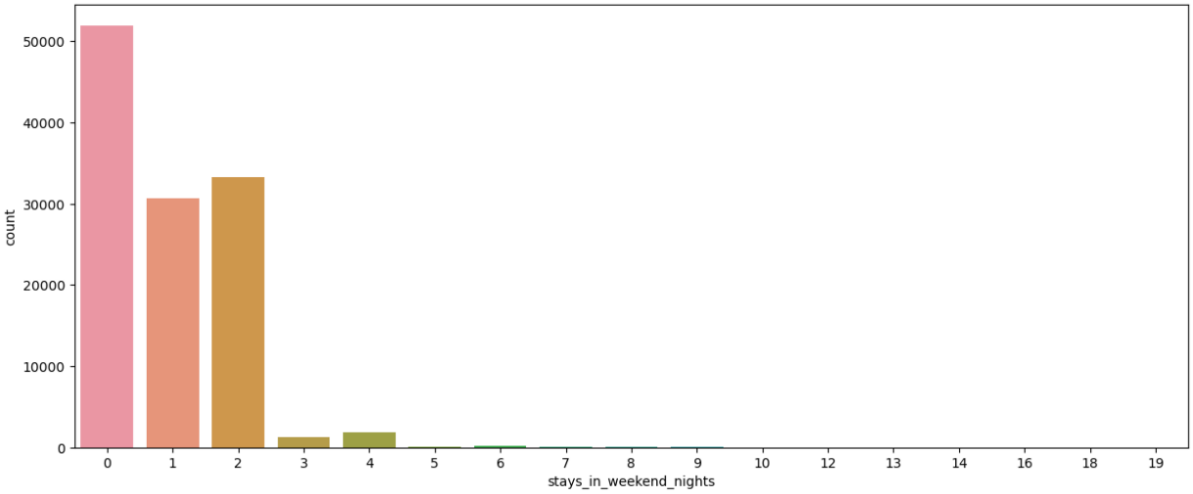
- ‘arrival_date_month’ Variable

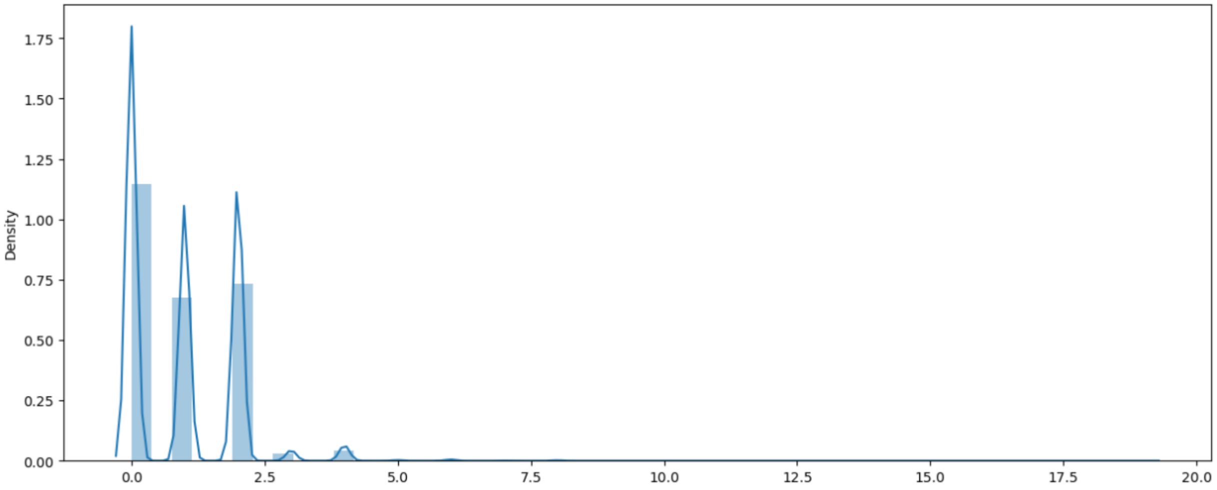


Inference –

We can see that the hotel bookings are high in July and August
This hike in hotel bookings can be due to many reasons
Summer Vacation Season:
In many countries, July and August coincide with the summer vacation season when schools are out, and families and individuals take extended breaks. This leads to increased travel and, consequently, higher hotel bookings.
Good Weather:
July and August are typically associated with warm weather and longer daylight hours in many parts of the world. This pleasant weather encourages people to travel, go on outdoor adventures, and visit tourist destinations.
International Tourism:
July and August are prime months for international travel, with tourists from around the world exploring different countries and regions. This influx of international travelers contributes to higher hotel occupancy rates.

- ‘stay_in_weekend_nights’ Variable





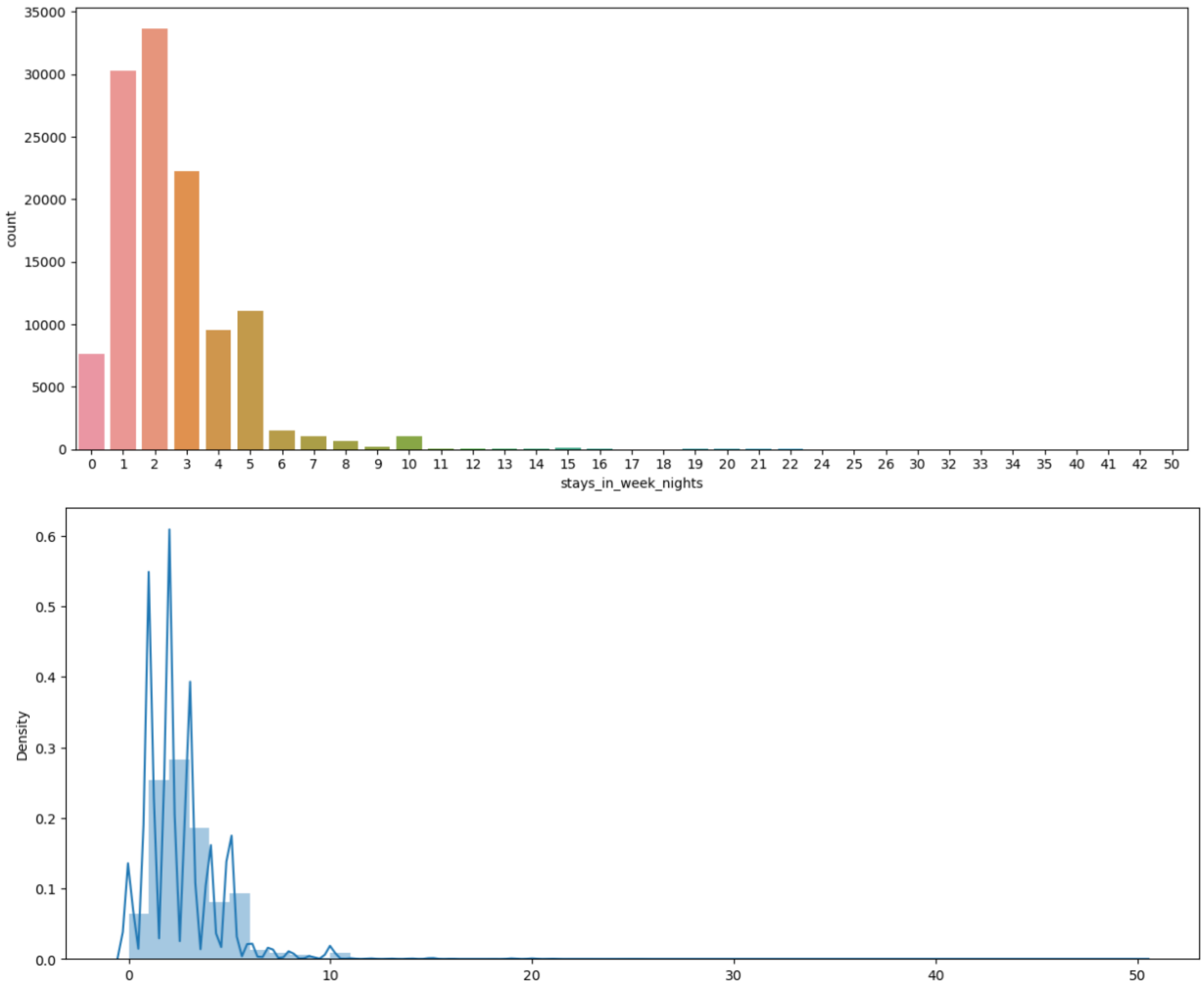
Inference –

Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
We can see that most customers prefer to stay in the weekend nights mostly once or twice
43.5% of the people do not prefer to stay in weekend nights
25.7% of the people stay once in the weekend nights
27.8% of the people stay twice in the weekend nights
Leisure and Getaways:

Weekends offer a break from the routine of work and daily responsibilities, making them an ideal time for leisure travel. People take advantage of weekends to go on getaways, explore new places, and relax.
Extended Vacations:

Many travellers plan longer vacations that include weekends to maximize their time away from work. They may book hotel stays to cover both weekdays and weekends during these trips.

❖ ‘stay_in_week_nights’ Variable



Inference –

Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel

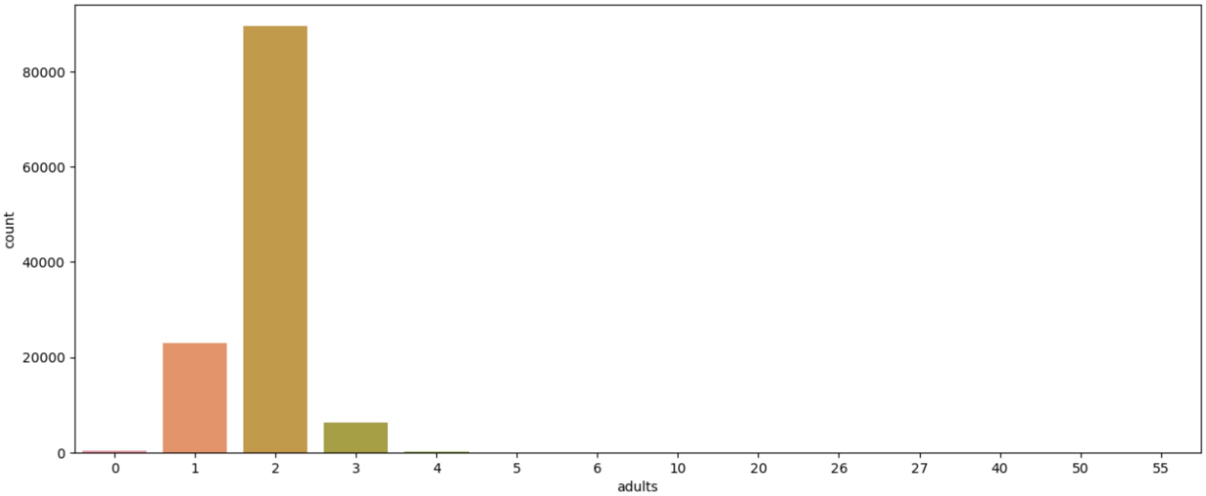
From the data we can see that maximum people who stay in week nights usually stay from one to three days

We can also observe that the data is highly skewed

Business Travel:

Weeknight hotel stays are often driven by business travel. Professionals travel for work-related meetings, conferences, and training sessions during the workweek and may stay in hotels for convenience and proximity to their work commitments.

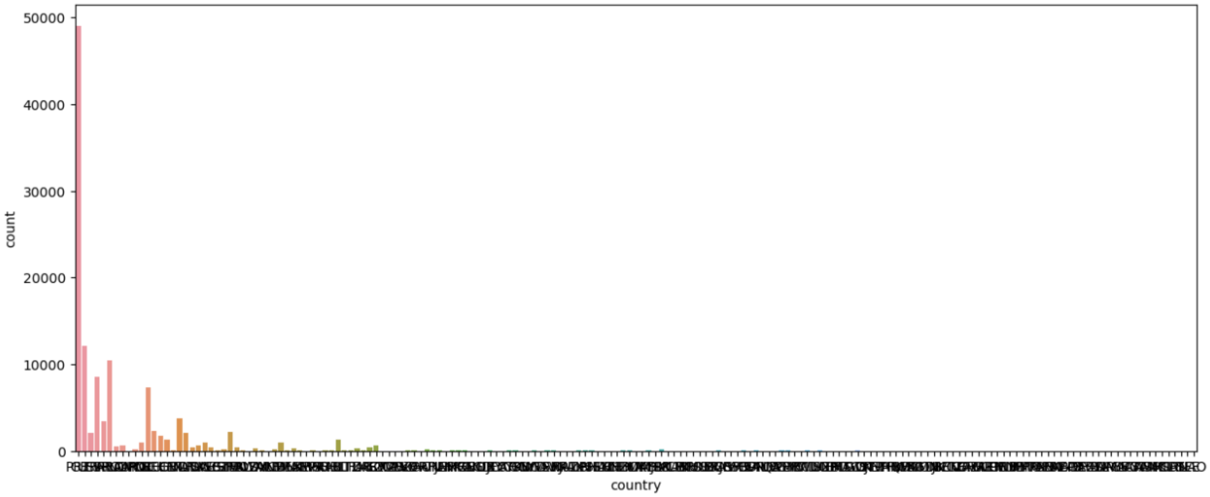
❖ ‘adults’ Variable



Inference –

Adults who come in twos are very high counted above 80000
75% of the customers come in twos
19% of the customers come ones

❖ ‘country’ Variable

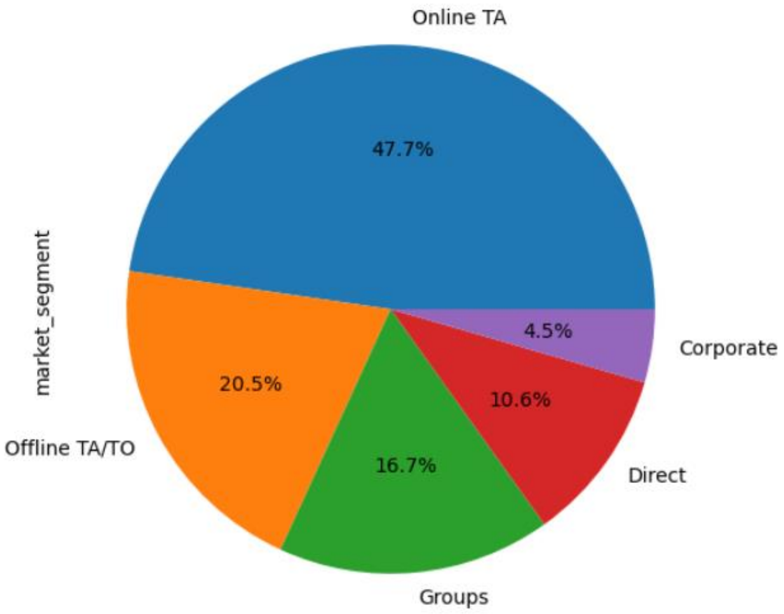


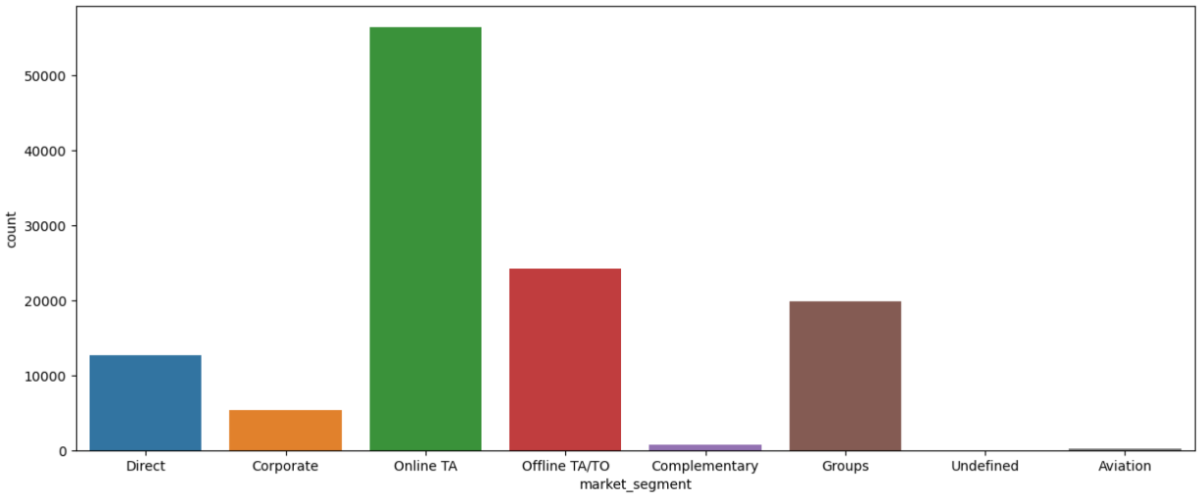
Major Class in the ‘country’ variable	Abbreviation	Percentage of counts
PRT	Portugal	41.107295
GBR	Great Britain	10.159142
FRA	France	8.723511
ESP	Spain	7.176480
DEU	Germany	6.103526

Inference –

we can see that the maximum number of bookings occured from Portugal, followed by Great Britain

❖ ‘market_segment’ Variable





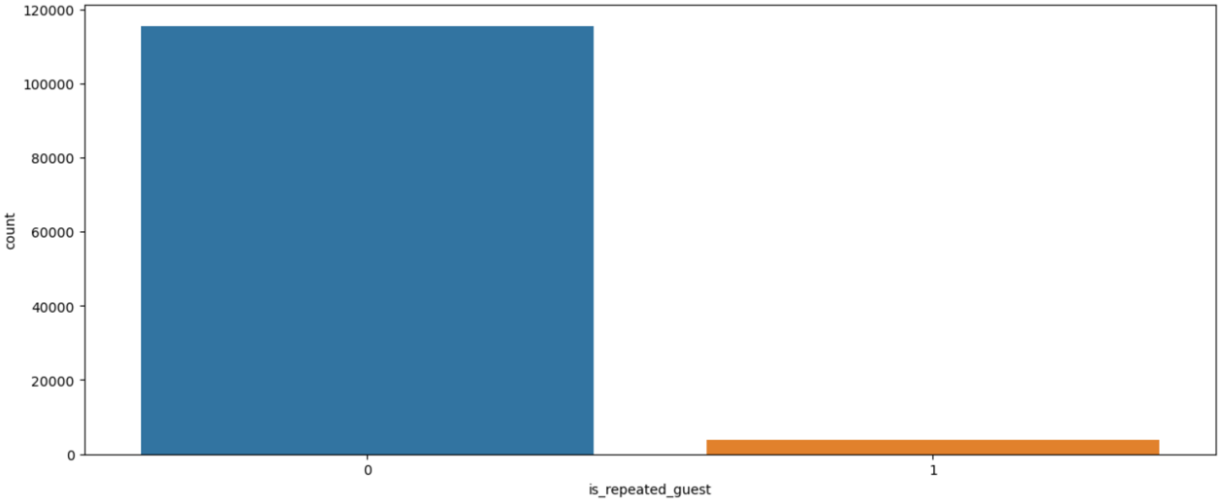
Online TA	47.304632
Offline TA/TO	20.285619
Groups	16.593517
Direct	10.558673

Corporate	4.435045
Complementary	0.622330
Aviation	0.198509
Undefined	0.001675

Inference –

47 percent of the people have booked through online travel agency

❖ ‘is_repeated_guest’ Variable



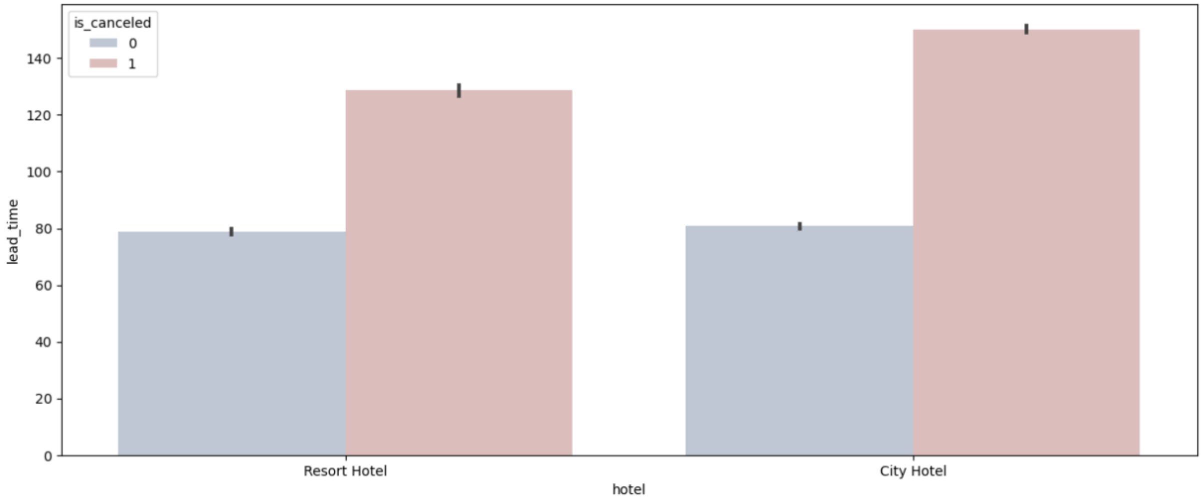
Classes present in ‘is_repeated_guest’ variable	Value counts of the classes	Percentage of the classes
0	115580	96.80
1	3810	3.19

Inference –

Here there only few repeated guest booked the hotel
96.80% of people appearing are not repeated guests

- **Bi-variate Analysis**

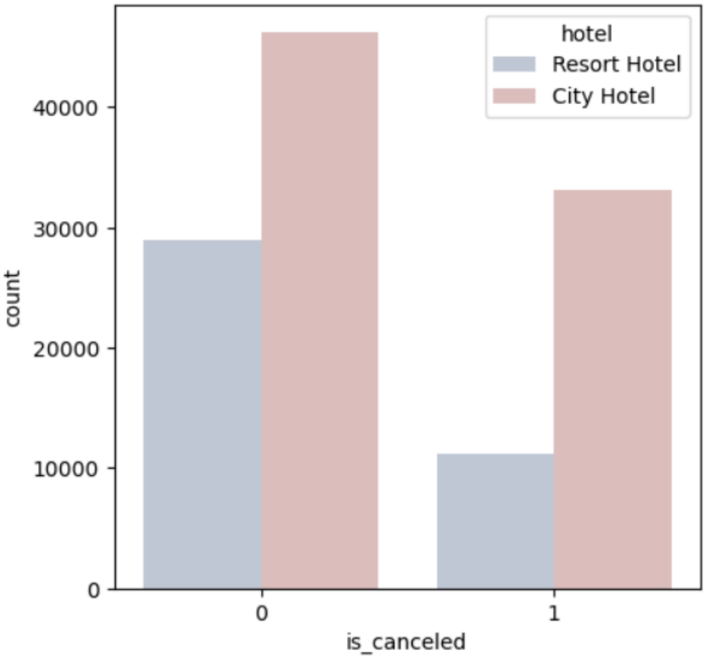
- ‘hotel’ Vs ‘lead_time’



Inference –

In both the hotels the bookings with more than lead time of 80 have higher chances of cancellations

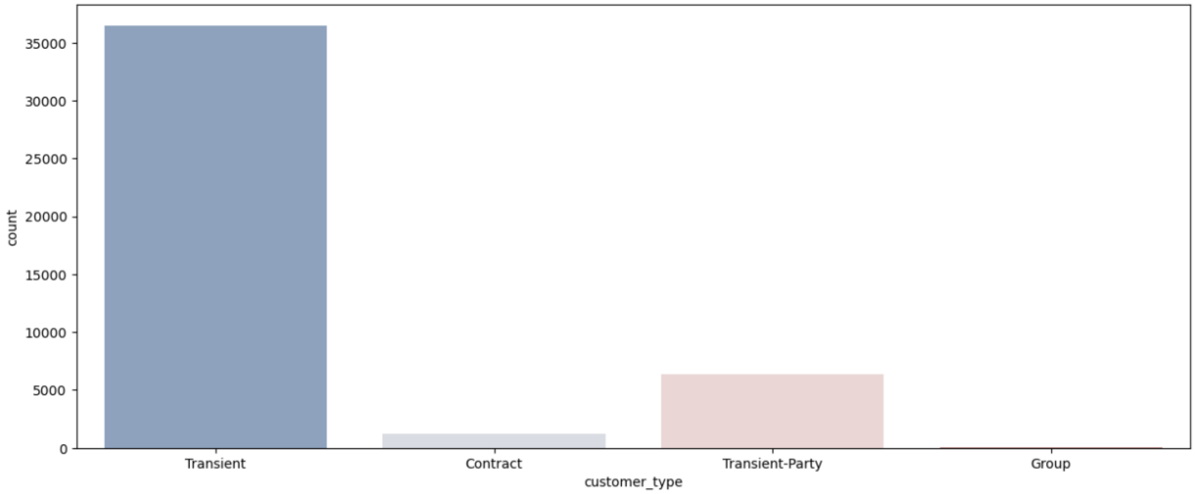
- ‘hotel’ Vs ‘is_canceled’



Inference –

In the cancelled bookings we can see that more number of cancellations occurred at City hotel

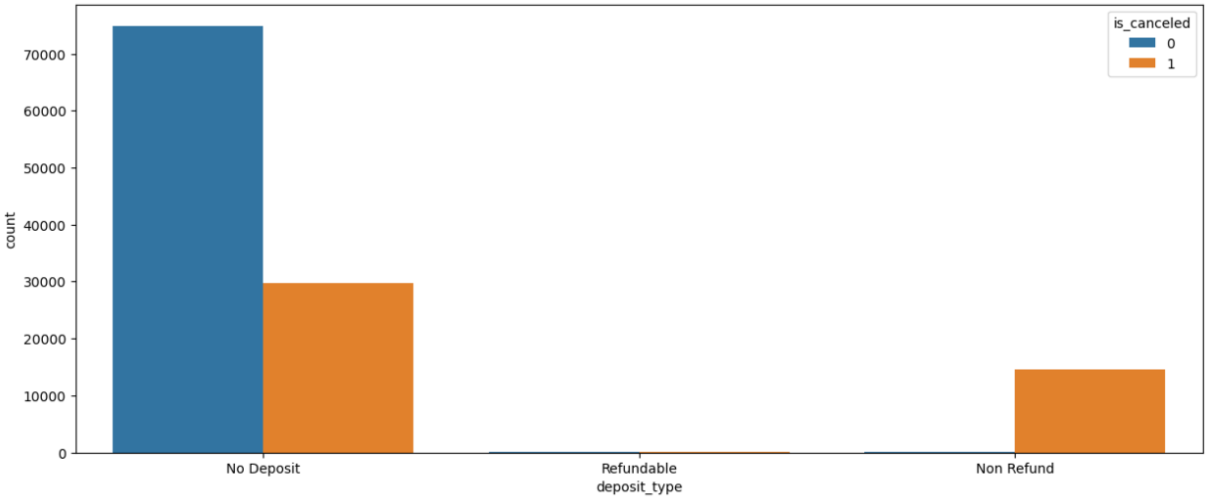
- ❖ ‘customer_type’ vs ‘is_canceled



Inference –

The ones with customer_type as transient are the ones with a short stay who have no special requirements, notably the cancellations are higher for the transient customer_types

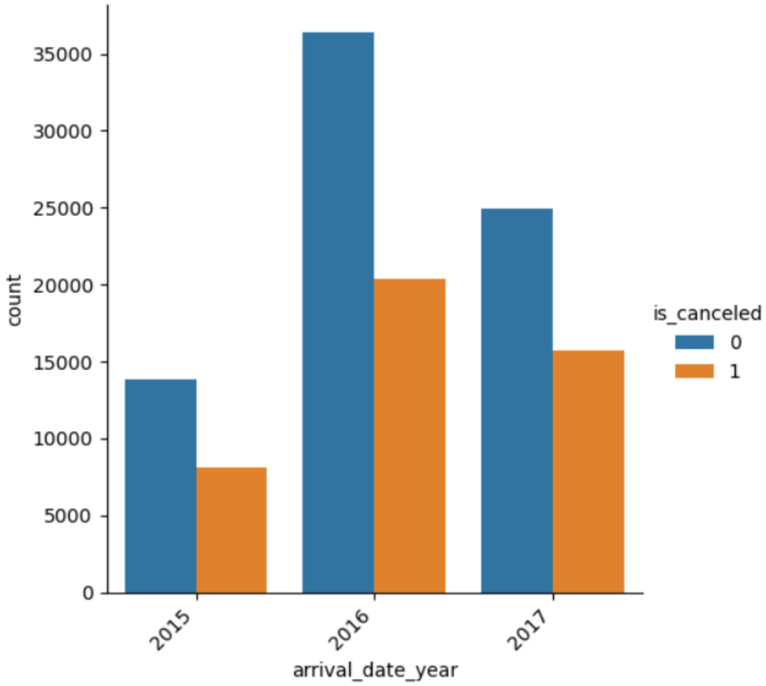
- ❖ ‘deposit_type’ Vs ‘is_canceled’



Inference –

Most of the people with non-refundable deposit type have cancelled

- ‘arrival_date_year’ Vs ‘is_canceled’

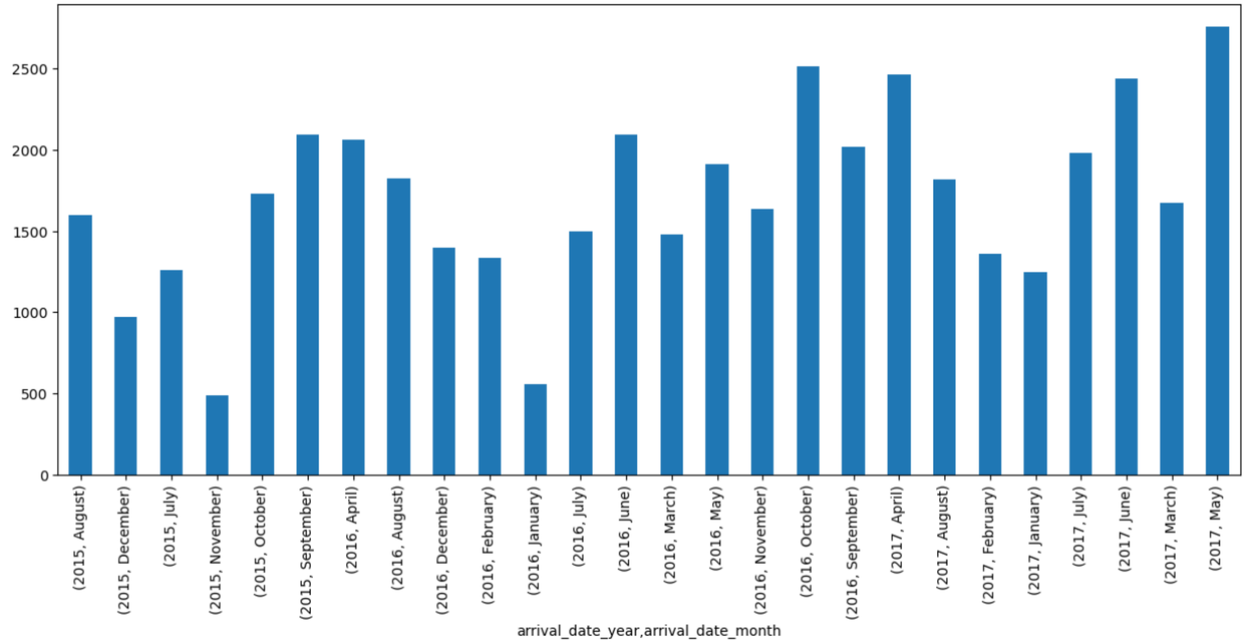


Inference –

From the above graph we could infer that the maximum number of bookings has occurred in the Year 2016 as well as maximum number of cancellations has also occurred in the year 2016.

- **Multi-variate Analysis**

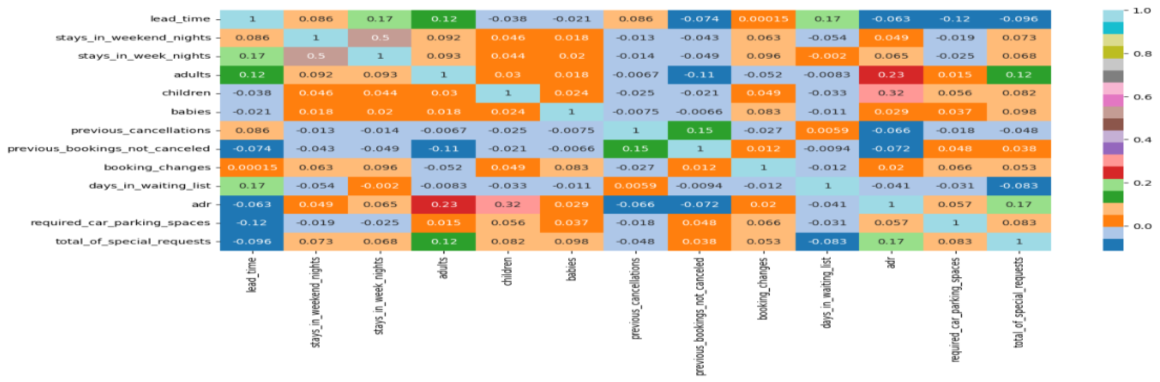
❖ 'arrival_date_year' Vs 'arrival_date_month' Vs 'is_canceled'



Inference –

For the year 2015 it is observed that the maximum number of cancellations occurred in the month of September.
For the year 2016 it is observed that the maximum number of cancellations occurred in the month of October.
For the year 2017 it is observed that the maximum number of cancellations occurred in the month of May.

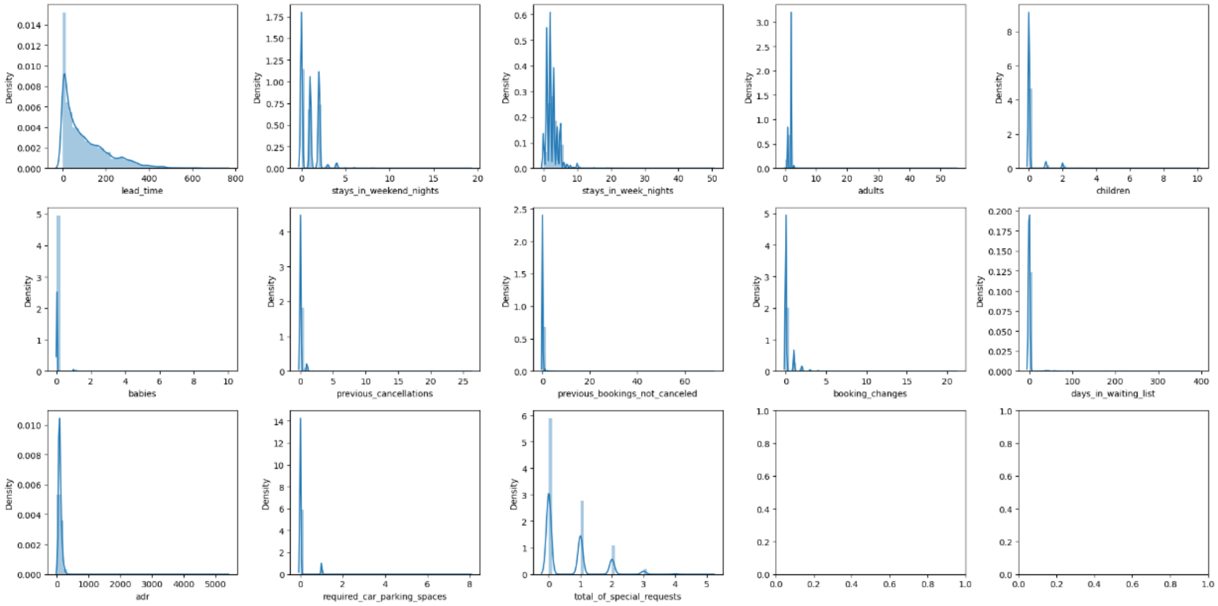
❖ Checking for Co-relation

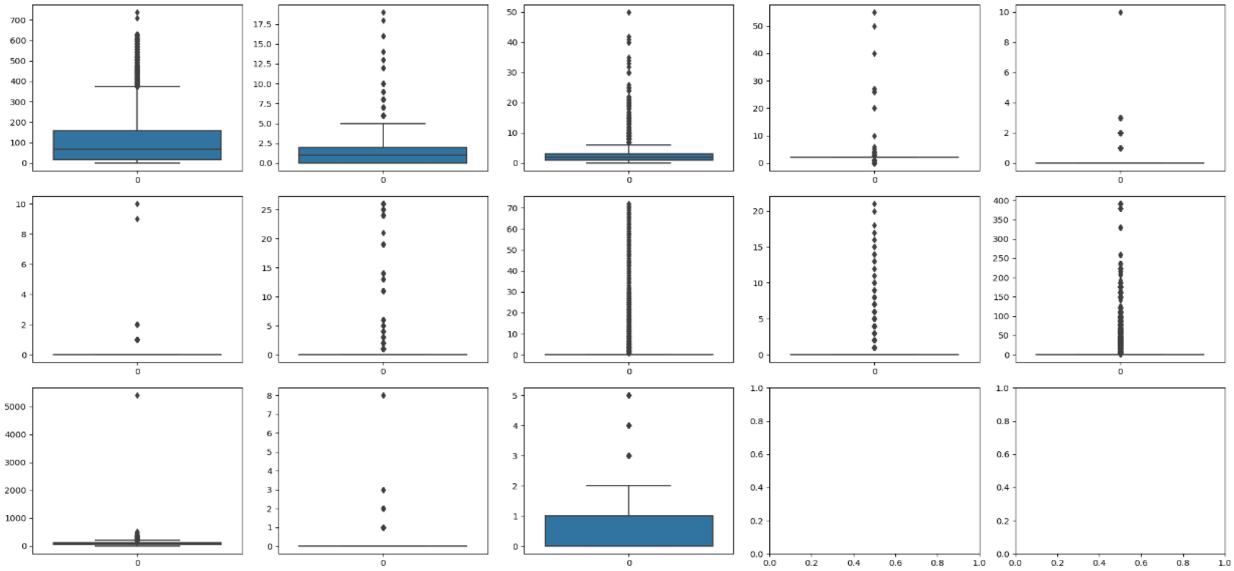


Inference –

We can observe the correlation between the variables.

- **Outlier Detection**





Inference –

The data exhibits a slight right-skew, as evident from the distribution plots. Notably, the presence of numerous data points exceeding the upper whisker
boundary strongly suggests the existence of outliers within the dataset

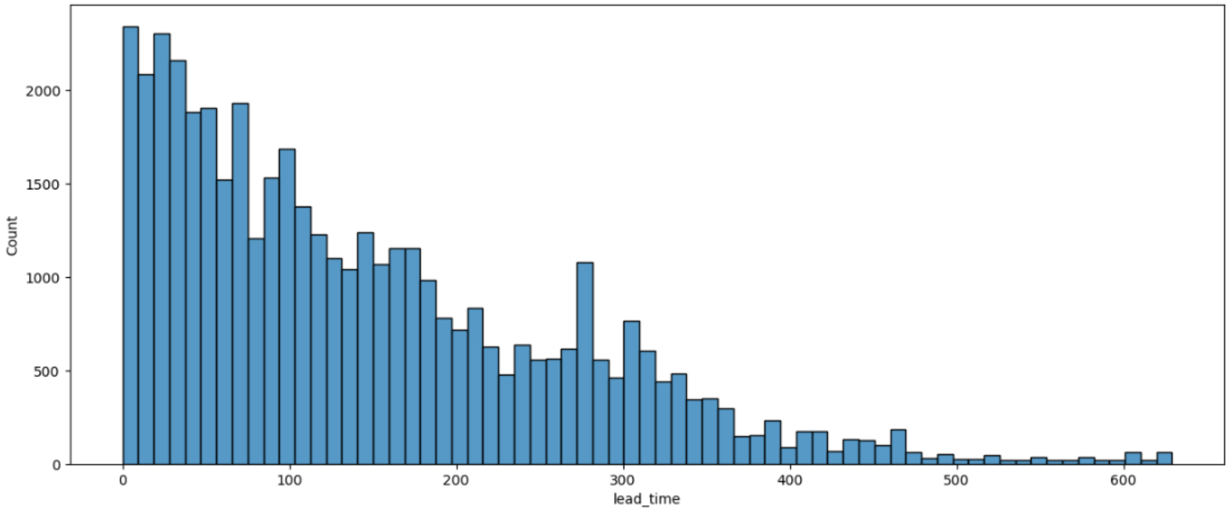
- **Hypothesis Testing**

❖ ‘is_canceled Vs ‘lead_time

➢ Assumptions

- H0 - higher the lead time does not impact cancellation
- H1 - higher the lead time impact cancellation

  It is a Right tailed test



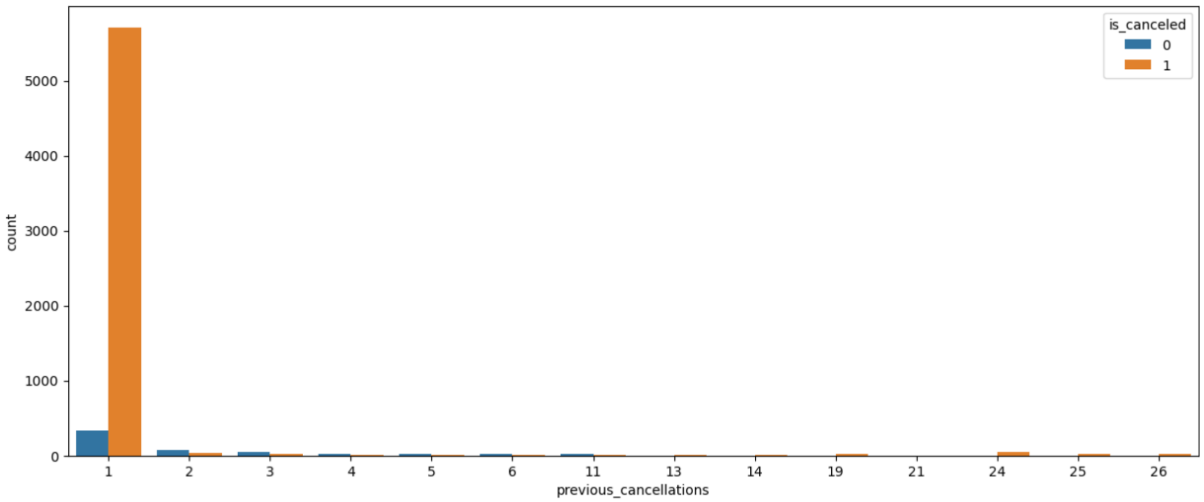
Result – p_value = 1.0

- Failed to reject null hypothesis.
- Failing to reject the null hypothesis confirms that our assumption that earlier the bookings made (higher lead time), higher the chances of cancellation

❖ ‘previous_cancellations’ Vs ‘is_canceled’

➢ Assumptions

- H0 - The cancellation status of the current booking is independent of the previous cancellation.
- H1 - The cancellation status of the current booking is dependent of the previous cancellation



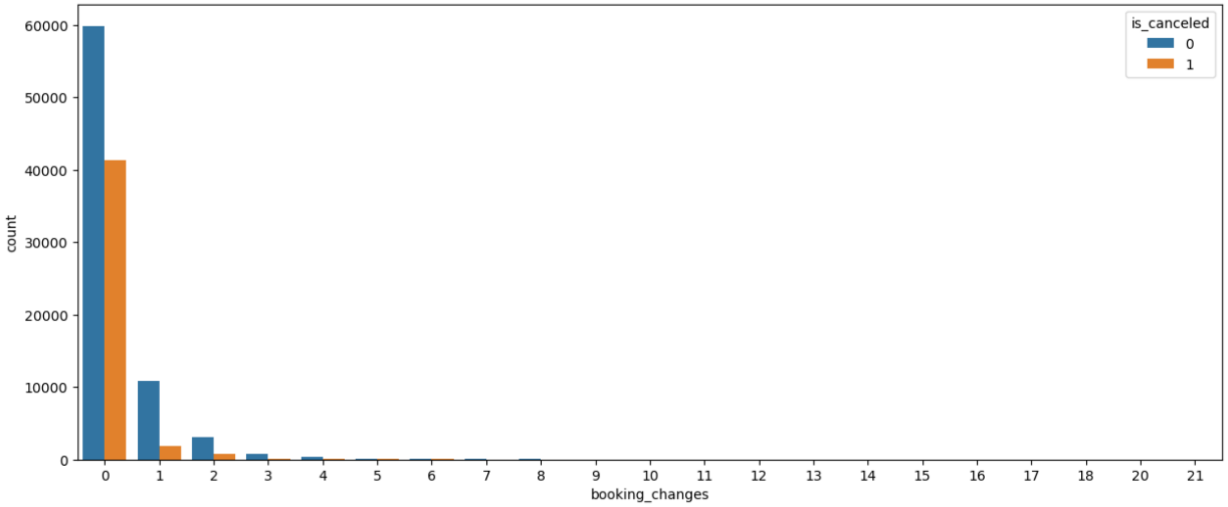
Result – P_value = 6.5036876520562995e-06

- Rejecting null hypothesis, states that the current booking is dependent on the previous c ancellation

❖ 'booking_changes' Vs 'is_canceled'

➢ Assumptions

- H0 – The booking changes does not have an impact on the cancellation
- H1 – The number of booking changes has an impact on the number of cancellations



Result – P_value = 0.9745985018985022

- Failed to reject null hypothesis, people with changes have cancelled the booking, which justifies our assumption ‘days_in_waiting_list’ Vs ‘hotel’

- Assumptions
  - H0 - City hotel has more waiting period
  - H1 - Resort hotel has more waiting period

Result – P_value = 1.0

- Failed to reject null hypothesis, the same can be observed from the graph that the city h otel has more waiting days

- **Feature Engineering**

Feature engineering is the process of getting data ready for machine learning by carefully choosing and improving the input variables, known as "features." It's like preparing the ingredients for a recipe – you want to make sure you have the right ones, and they're in the best possible form for cooking. Feature engineering can involve tasks like cleaning data, handling missing values, and transforming variables to make them more useful for machine learning algorithms.

  - Feature selection

Feature selection is about picking the most important features from a bunch of potential ones. Imagine you're packing for a trip, and you want to bring only the most essential items in your suitcase to keep it light. Similarly, in feature selection, we choose the most relevant features to include in our model. This not only simplifies the model but also makes it faster and less prone to errors.

  - Feature importance

Feature importance is like figuring out which players are the most valuable on a sports team. In machine learning, it helps us understand which features have the most impact on a model's predictions. By identifying these key features, we can focus our efforts on improving them or use them to make more informed decisions. It's a bit like finding the most influential pieces in a puzzle – they have a bigger say in solving the overall picture.

In summary, feature engineering involves preparing data for machine learning, including tasks like cleaning and transforming. Feature selection is all about choosing the most critical features, while feature importance helps us understand which features matter the most for accurate predictions, akin to selecting the right ingredients, packing wisely, and recognizing star players on a team. These steps collectively lead to better machine learning models.

  - Feature engineering techniques applied

➢ Family Size Categorization

To better capture the dynamics of guests traveling with children, babies, or family members, we created a consolidated column called "Family." This column was further segmented into three categories - small, medium, and large-sized families based on the number of accompanying family members. This allows us to account for the varying needs and preferences of different family sizes during their stay.

- Seasonal Arrival Date

We transformed the "Arrival date month" feature into a more intuitive representation by associating each month with a specific season. For instance, if a reservation falls in December, it is categorized as winter. This simplifies the understanding of booking patterns and helps us make season-specific recommendations or adjustments.

- Room Preference Analysis

To gain a deeper insight into room allocation satisfaction, we combined the "reserved room type" and "assigned room type" columns into a single feature called "Room Type." This feature indicates whether guests received their preferred room type or not, providing valuable information on guest satisfaction and potential room allocation issues.

- Geographic Clustering of Countries

We grouped countries based on their respective continents. This geographic categorization enables us to identify trends and preferences among guests from specific regions and tailor services or marketing strategies accordingly. It also simplifies the analysis of regional performance.

- Holiday Identification

By analyzing reservation status dates, we categorized bookings as either holidays or non-holidays. This distinction allows us to account for the impact of holidays on booking patterns, occupancy rates, and guest behavior, enabling us to optimize staffing and services during peak holiday periods.

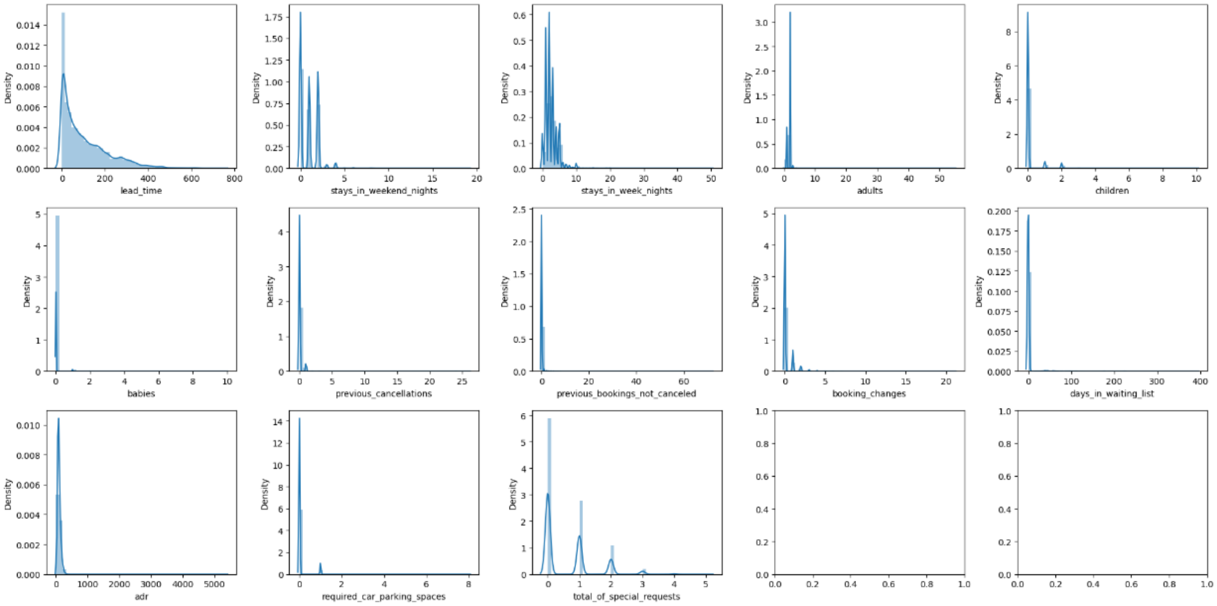
- Agent Booking Tiers

Agents responsible for bookings were classified into three distinct groups based on their booking volume. This segmentation helped us identify the contributions of different agent tiers to overall bookings and guest satisfaction. It also informs our strategy for agent incentives and management.

- **Data Preprocessing**

    ○ Outlier Detection

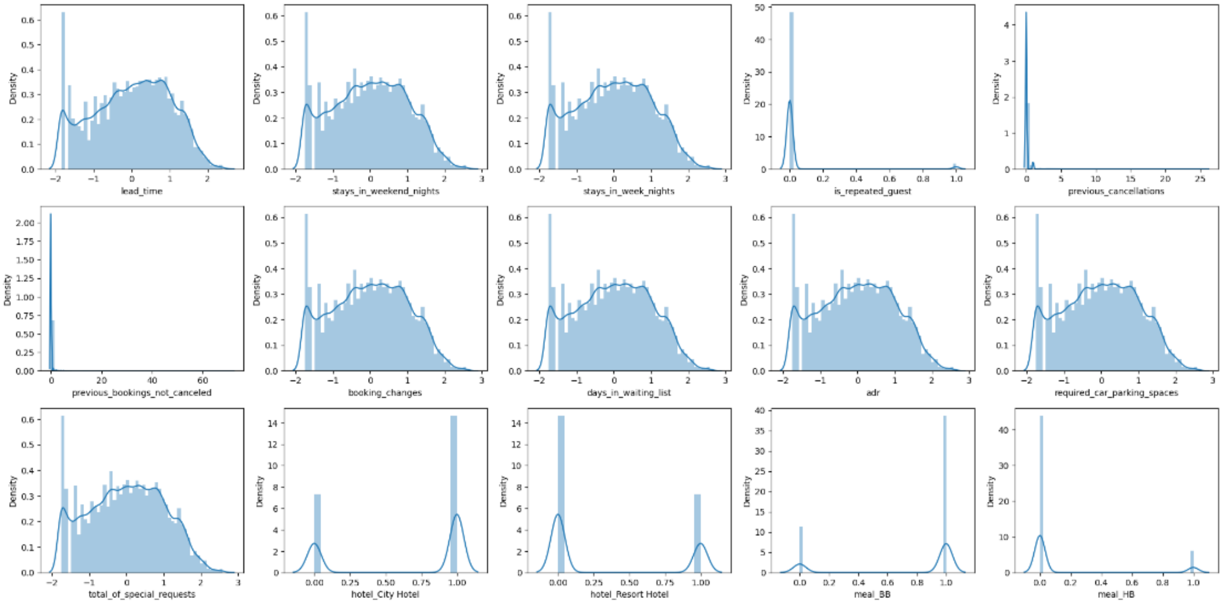
In the data preprocessing phase, we applied a visualization technique to assess the presence of substantial skewness in our dataset. Skewed data distributions can have a significant impact on the performance of our analytical models. This visual representation allowed us to observe the shape of the data distribution for each feature. From these distributions, we can observe that the data exhibits pronounced skewness, indicating that further attention to address this skewness may be necessary for our data preprocessing efforts.



    ○ Transformation

We transformed and scaled both the training and test datasets to improve our machine learning model's effectiveness and adaptability. This process ensures accurate performance and the ability to handle new data.

To maintain fair evaluations, we strictly separated the training and test datasets to avoid data leakage. Consistently applying these transformations to both datasets enhance our model's reliability when making predictions on new, unseen data. This strengthens our model's ability to generalize and make dependable predictions, bolstering our machine learning solution's overall robustness.



- Encoding

We applied one-hot encoding using the 'get_dummies' method from Pandas to convert categorical features into binary columns. This transformation prepares the data for machine learning by ensuring that the model can interpret and use categorical information correctly during training, without introducing any unintended ordinal relationships.

- **Model Building**
    - Splitting data into training and testing sets

Separating the data into different parts is an important first step in our project's data preparation. With the provided code, we've split our data into two main groups:

'X' includes all the things we use to make predictions, like customer information and booking details. It's like the ingredients we use to cook a meal.

'y' is specifically for the thing we're trying to predict, which is whether a booking will be canceled or not. It's like the result we want to find, such as whether the meal will turn out tasty or not.

So, 'X' has all the important information, and 'y' is where we keep what we're trying to figure out. This separation helps us organize our data and makes it easier to build and evaluate our predictive model.

- Base Model Building

In our machine learning project, we meticulously followed a systematic approach to model development, ensuring that each step was carried out with precision and care. A crucial aspect of our data preprocessing pipeline involved separately transforming the training and test datasets, which included essential steps like scaling to improve model performance and generalization.

For the modeling phase, we opted to utilize a Logistic Regression classifier from the Scikit-Learn library as our base model. This choice was based on its suitability for our classification task and its interpretability, making it a solid starting point for our analysis.

The initial results from our base model were promising. While accuracy serves as a valuable initial indicator of the model's predictive capability, we recognize the need for a more comprehensive performance assessment. Therefore, we conducted a thorough evaluation of our model, considering additional metrics beyond accuracy to gain a deeper understanding of its performance.

In this classification report, we observe that the precision for class 0 is 0.76, meaning that when our model predicted class 0, it was correct 76% of the time. The recall for class 0 is 0.95, indicating that our model identified 95% of the instances of class 0 correctly. The F1-score for class 0 is 0.84, which is a harmonic mean of precision and recall.

For class 1, our model achieved a precision of 0.84, a recall of 0.49, and an F1-score of 0.62. These metrics provide a more nuanced view of our model's performance, showing that it excels at correctly identifying class 0 instances but has room for improvement in classifying class 1 instances.

This comprehensive evaluation helps us make informed decisions about model refinement and potential areas for improvement in our machine learning project, ensuring that we consider precision, recall, and F1-score for each class to better understand our model's capabilities. Proceeding further as our model falls under Classification, we have decided to use various models namely:

1. Logistic Regression Model
2. Decision Tree Model
3. Random Forest Model
4. KNN Model
5. Ada Boost Technique
6. XG Boost Technique
7. Gradient Boosting Technique

❖ Classification Report – Base Model

	Precision	Recall	F1 score	Support
0	0.84	0.91	0.87	22654
1	0.82	0.70	0.75	13163
Accuracy			0.83	35817
Macro avg	0.83	0.80	0.81	35817
Weighted avg	0.83	0.83	0.83	35817

Hyper parameter Tuning:

By training a model with existing data, we are able to fit the model parameters. However, there is another kind of parameter, known as Hyper parameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Models can have many hyper parameters and finding the best combination of parameters can be treated as a search problem. The best strategies for Hyper parameter tuning are:

GridSearchCV

In GridSearchCV approach, the machine learning model is evaluated for a range of hyper parameter values. This approach is called GridSearchCV, because it searches for the best set of hyper parameters from a grid of hyper parameters values.

1. Decision Tree Regressor:

A Decision Tree Regressor is a supervised machine learning algorithm used for regression asks. It creates a tree-like structure where each internal node represents a feature, and each leaf node represents a predicted numerical value. The decision tree splits the data based on feature values to minimize the mean squared error (MSE) or other regression loss functions. Decision Trees can capture non-linear relationships in data and are interpretable.

2. Random Forest Regressor:

A Random Forest Regressor is an ensemble learning method that combines multiple decision tree regressors to make more accurate predictions. It fits a collection of decision tree regressors on different subsets of the dataset and averages their predictions to reduce overfitting and improve predictive accuracy.

3. Gradient Boosting Regressor:

Similar to the Gradient Boosting Classifier, the Gradient Boosting Regressor builds an additive model in a forward stage-wise fashion for regression tasks. It fits a sequence of regression trees to the negative gradient of the loss function, optimizing for regression loss functions like mean squared error (MSE).

4. XGBoost Regressor:

XGBoost can also be used for regression tasks. It shares many features with the XGBoost Classifier but is applied to predict continuous numerical values. XGBoost includes enhancements like parallel computing, cache optimization, and regularization to improve regression accuracy and prevent overfitting.

5. AdaBoost Regressor:

AdaBoost (Adaptive Boosting) is an ensemble learning technique that can be applied to regression problems as well. The AdaBoost Regressor works similarly to the AdaBoost Classifier but focuses on improving regression performance. It combines multiple weak regression models (usually decision trees with limited depth) to create a strong ensemble model.

6. Stack Classifier (Stacking):

Stacking is an ensemble learning technique used for classification problems, but it can also be adapted to regression tasks (Stacking Regressor). Stacking combines predictions from multiple diverse models to improve overall predictive performance.

- Improvements to base model with Feature Selection

S.no.	**Model**	**Accuracy**	**Recall**	**Precision**	**F1 Score**
1	Base model	0.831895	0.695054	0.820097	0.752416
2	Knn model with Feature Selection	0.851104	0.775203	0.811258	0.792821
3	Decision tree model with Feature Selection	0.816205	0.582390	0.875914	0.699612
4	Random Forest model with Feature Selection	0.795293	0.466687	0.951665	0.626262
5	Bagging using Knn with Feature Selection	0.852751	0.776723	0.814078	0.794962
6	Bagging using dt with Feature Selection	0.876288	0.791461	0.860707	0.824633
7	Stacking using lr,knn,dt with Feature Selection	0.872882	0.790929	0.852522	0.820571
8	**Ada boost with random forest and Feature Selection**	**0.881118**	**0.785535**	**0.878132**	**0.829257**
9	Gradient boost with Feature Selection	0.866711	0.783560	0.842716	0.812062
10	Extreme gradient boost with Feature Selection	0.872658	0.791385	0.851619	0.820398

- Hyper tuning the Models
  - Grid Search CV for KNN Model

N_Neighbors	3,5,7
P	1,2

  - Best Params for KNN Model

N_Neighbors	7
P	1

  - Grid Search CV for Decision Tree Model

Criterion	Entropy, Gini
Max Depth	3,5,7

  - Best Params for Decision Tree Model

Criterion	Gini
Max Depth	7

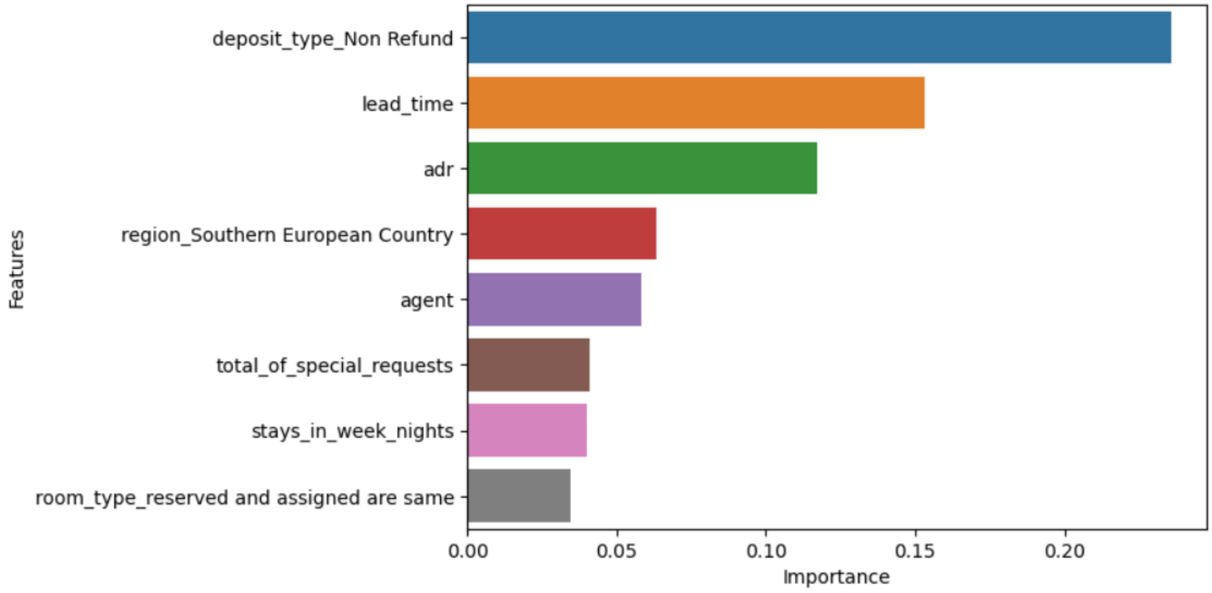
- Grid Search CV for Random Forest Model

Criterion	Entropy, Gini
Max Depth	10,15
Max Features	Sqrt,log2
Min Samples split	2,8
Min Samples Leaf	5,9
Max leaf nodes	8,11
N_estimators	100

- Grid Search CV for Random Forest Model

Criterion	Gini
Max Depth	10
Max Features	Sqrt
Min Samples split	2
Min Samples Leaf	9
Max leaf nodes	11
N_estimators	100

❖ Feature Importance Chart



Note –

In the above chart showing importance of first 8 features

- Business Interpretations

Project Objective: The goal of this project is to develop supervised classification models to predict whether a customer who books a hotel room will show up for their reservation or not. By leveraging a comprehensive database of hotel-related features, we aim to assist the hotel industry in optimizing resource allocation, enhancing customer experience, and improving revenue management

Data Description: The dataset consists of numerous columns, including hotel type (e.g., resort or city hotel), booking lead time, arrival date information, guest demographics, reservation details (e.g., meal plan, room type), booking channel, historical guest behavior (e.g., previous cancellations and bookings), and other relevant attributes.

Model Building: A variety of machine learning algorithms are employed to build classification models. The dataset is typically divided into training and testing sets, and predictive models are trained using algorithms such as logistic regression, decision trees, random forests, KNN, bagging, and boosting models have been built. Features are pre-processed, and model performance is evaluated using metrics like accuracy, precision, recall, and F1-score.

1. **No-Show Prediction:** The primary use of these models is to predict whether a customer is likely to show up for their hotel reservation. This information can be used to manage hotel resources more efficiently, such as overbooking rooms during periods of high no-show probability or offering incentives to guests with high no-show risk to encourage them to show up.

2. **Resource Allocation:** Hotels can optimize resource allocation based on no-show predictions. For example, they can reduce staffing or service levels during periods with a low likelihood of no-shows, while being prepared for potential no-shows during high-risk periods.

3. **Revenue Optimization:** Accurate predictions can help hotels maximize revenue. By identifying high-risk reservations, hotels can offer additional rooms to customers when they might otherwise have been fully booked, potentially increasing revenue.

4. **Customer Segmentation:** We can segment customers based on their likelihood of showing up. For example, regular customers who rarely no-show could be offered loyalty rewards, while high-risk customers might be prompted to provide a deposit.

5. **Marketing and Communication:** Customize communication and marketing strategies for customers based on their likelihood of showing up. For example, high-risk customers could receive reminders or incentives to confirm their reservations.

6. **Pricing Strategies:** Adjust pricing strategies based on no-show predictions. For example, offer discounts or promotions to customers with a high likelihood of not showing up to encourage them to keep their reservations.

7. **Operational Efficiency:** The models can help in more efficient handling of last-minute changes, as rooms may become available due to no-shows. This can lead to smoother operations and better customer satisfaction.

8. **Customer Experience:** Knowing that a customer is likely to show up or not can influence how they are treated during their stay. High-risk customers can be given extra attention to ensure their satisfaction and likelihood of returning.

9. **Tracking Model Performance:** Continuously monitor the accuracy and performance of your classification models. Make adjustments as needed to improve predictions and business outcomes.

10**.** **Ethical Considerations:** Ensuring the project complies with legal and ethical standards, including data privacy regulations, is of paramount importance.

In conclusion, this project focuses on building classification models to predict hotel booking no-shows, with the objective of benefiting the hotel industry through optimized resource allocation, enhanced revenue management, improved customer experience, and operational efficiency. It emphasizes the ethical and legal aspects of data usage and underlines the importance of ongoing model maintenance and improvement

- **Conclusion**

In this phase of our hotel booking cancellation prediction project, we have executed a series of critical processes, each contributing to a deeper understanding of our dataset and an enhanced foundation for future model development.

Data Pre-processing: During this stage, we meticulously prepared our data for modeling. Key steps included:

Data Type Standardization: We standardized data types across columns, ensuring alignment with our understanding of the hotel booking dataset.

Column Removal: We thoughtfully removed redundant columns that exhibited a one-to-one relationship with our target variable, as they did not provide additional information.

Missing Value Treatment: Missing values were addressed diligently, preserving data integrity and preventing bias in our analysis.

Exploratory Data Analysis (EDA):
Our EDA efforts were insightful and revealed crucial patterns and relationships within the dataset:

Correlation Analysis: We identified high correlations between certain independent variables, offering insights into potential multi collinearity issues.

Feature Significance: By scrutinizing the relationship between independent variables and our target variable, we gained valuable insights into the significance of each feature in predicting hotel booking cancellations.

Outlier Detection: The presence of outliers was noted and will be addressed in our subsequent modeling phase.

Feature Engineering: To enhance the predictive power of our model, we strategically engineered new variables:

Handling Data Imbalance: Variables were engineered to address imbalances within the dataset, contributing to a more robust predictive framework.

High Cardinality Variables: Techniques for managing high cardinality variables were applied, ensuring they can be effectively utilized in modeling.

Model Building - Base Model:
We initiated our modeling journey with the construction of a base model using Logistic Regression:

Data Splitting: We carefully partitioned the dataset into independent and target variables, creating distinct training and testing sets.

Outlier Transformation: A data transformation technique was employed to mitigate the impact of outliers without sacrificing data integrity or causing leakage.

Performance Metrics: Our base model exhibited promising accuracy, correctly predicting 78% of booking cancellations, setting a strong foundation for subsequent model iterations.

Future Steps: As we move forward, we are committed to advancing our modeling efforts. Our objectives include:

Multicollinearity Mitigation: We will actively address multicollinearity among independent variables, ensuring that model inputs remain independent and reliable.

Model Diversification: A suite of classification techniques, including K-Nearest Neighbors (KNN), Naïve Bayes, and Ensemble techniques, will be implemented, allowing us to evaluate their respective performances.

Performance Evaluation: Comprehensive evaluation, using metrics aligned with our business objectives, will guide the selection of the most effective model for predicting hotel booking cancellations.

Our dedication to improving model performance and gaining deeper insights into booking behaviors remains steadfast. We anticipate that these efforts will provide valuable tools for the hotel industry to optimize booking management strategies and enhance customer satisfaction.

- **References**

  - Kaggle Link - https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand
  - Article Link - https://www.sciencedirect.com/science/article/pii/S2352340918315191