

**Hotel Reservation Cancellation Prediction**

**Final Report**



Team:

|  |  |
| --- | --- |
| Batch details | PGPDSE CHN Jan23 (Group – 2) |
| Team members | 1. Praveen Arokia Raj A 2. G.Raghav 3. Sathish J 4. Harisri Karthikeyan 5. Balachandar P 6. Mohana Devan R |
| Domain of Project | Tourism Industry |
| Proposed project title | Hotel Reservation Cancellation Prediction |
| Group Number | Group 2 |
| Team Leader | Balachandar P |
| Mentor Name | Mr Jayveer Nanda |

**Contents**

[**1.** **Summary of the problem statement, Data and findings** 4](#_Toc137639637)

[1.1. Problem Statement 4](#_Toc137639638)

[1.2. Project Objectives: 4](#_Toc137639639)

[**2.** **EDA** 6](#_Toc137639640)

[2.1. Approach: 6](#_Toc137639641)

[2.2. Analysis: 7](#_Toc137639642)

[2.2.2 Bivariate Analysis: 18](#_Toc137639643)

[2.2.3 Multivariate Analysis: 22](#_Toc137639644)

[2.2.4 Heatmap: 23](#_Toc137639645)

[2.2.5 VIF: 23](#_Toc137639646)

[**3**. **Statistical Tests**: 24](#_Toc137639647)

[3.1 Outlier and Distribution of the numerical variable: 25](#_Toc137639648)

[4. **Base Model Summary**: 26](#_Toc137639649)

[4.1 Model Building: 28](#_Toc137639650)

[4.2 Log of Odds: 30](#_Toc137639651)

[5. **Building Base Model using Sklearn Logistic Regression**: 31](#_Toc137639652)

[6. **Model built using Decision Tree**: 32](#_Toc137639653)

[7. **Model built using Random Forest Classifier**: 34](#_Toc137639654)

[8. **Model built using XG Boost Classifier**: 35](#_Toc137639655)

[9. **Model built using Ada Boost**: 37](#_Toc137639656)

[10. **Model built using Cat Boost**: 39](#_Toc137639657)

[11. **Model built using Hyper Parameter Tuning**: 41](#_Toc137639658)

[11.1 GridSearchCV using Random Forest Parameters: 41](#_Toc137639659)

[11.2 GridSearchCV using Decision Tree Parameters: 43](#_Toc137639660)

[12. **Business Inference** 45](#_Toc137639661)

# 

# **Summary of the problem statement, Data and findings**

## Problem Statement

Develop a classification model to predict the booking status (canceled or not canceled) based on the provided booking information. The model should use features such as the number of adults and children, length of stay, meal plan, parking requirements, room type, lead time, and previous booking and cancellation history to accurately classify whether a booking will be canceled or not. The objective is to develop a reliable model that can help hotel management to identify potential cancellations in advance and take appropriate measures to reduce the impact on their business.

The model's performance should be evaluated based on metrics such as accuracy, precision, recall, and F1 score. The model should be able to provide actionable insights to the hotel management team for better decision-making and business planning."

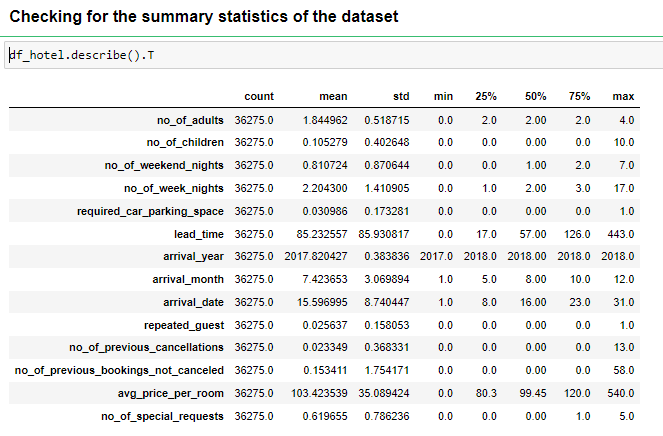
## Project Objectives:

The business objectives for the Hotel Reservations Classification Dataset include maximizing revenue, improving customer satisfaction, reducing cancellations, and streamlining operations. By analyzing the data, hotels can identify trends and patterns that can inform pricing and revenue management strategies, address negative guest experiences, predict and prevent cancellations, and streamline operations to enhance efficiency and profitability. The ultimate goal is to improve the overall guest experience while optimizing hotel operations for maximum success.

* 1. **Data & Findings :**
* Details about the data and dataset files are given in below link,  
  <https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset>
* Data Dictionary :
  1. Booking\_ID: unique identifier of each booking
  2. no\_of\_adults: Number of adults
  3. no\_of\_children: Number of Children
  4. no\_of\_weekend\_nights: Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
  5. no\_of\_week\_nights: Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
  6. type\_of\_meal\_plan: Type of meal plan booked by the customer:
  7. required\_car\_parking\_space: Does the customer require a car parking space? (0 - No, 1- Yes)
  8. room\_type\_reserved: Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels.
  9. lead\_time: Number of days between the date of booking and the arrival date
  10. arrival\_year: Year of arrival date
  11. arrival\_month: Month of arrival date
  12. arrival\_date: Date of the month
  13. market\_segment\_type: Market segment designation.
  14. repeated\_guest: Is the customer a repeated guest? (0 - No, 1- Yes)
  15. no\_of\_previous\_cancellations: Number of previous bookings that were canceled by the customer prior to the current booking
  16. no\_of\_previous\_bookings\_not\_canceled: Number of previous bookings not canceled by the customer prior to the current booking
  17. avg\_price\_per\_room: Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
  18. no\_of\_special\_requests: Total number of special requests made by the customer (e.g. high floor, view from the room, etc)
  19. booking\_status: Flag indicating if the booking was canceled or not.

# **EDA**

## 2.1. Approach:



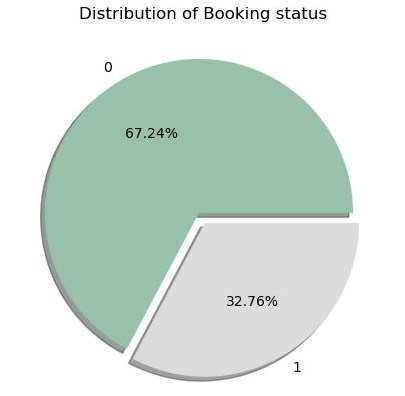
**Inference:**

From above statistics we get to know:

* + - The maximum number of guests who occupied the hotel room over the age of 18 is 4
    - The maximum number of guests who occupied the hotel room below the age of 18 is 10
    - On an average 1 weekend night (Saturday or Sunday) the guest has stayed or booked to stay at the hotel
    - The request for car parking space has majorly been asked for as a requirement (again this is a categorical variable)
    - The Lead time between the date of arrival and the date of booking on an average is 85 days and a maximum of 443 days
    - The data collected is for the years 2017, 2018
    - The arrival month we can get to see which month has the highest bookings, 12, December
    - With the arrival date we can classify or infer the following
  + The time of the booking , month end or the mid-month, here most frequent bookings happened for the month end
  + Weekday or the Weekend, where the cancellations happen the most
    - In the collected data, this organisation has had more of repeated guest than the new visits.
    - The maximum number of previous bookings that were cancelled and not cancelled by the customer prior to the current booking is 13 and 58 respectively
    - Price of the room ranges from $80 to $540 , however the average booking price has been ~$103
    - Comparatively there were not special requests made by the customer (e.g. high floor, view from the room, etc), the maximum number itself is 5

## 2.2. Analysis:

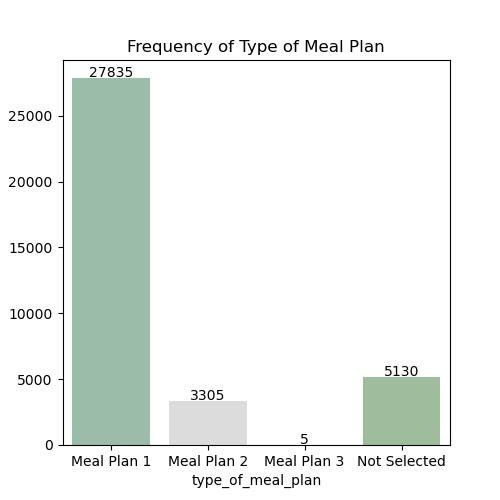
* **Target Variable analysis:** 
  + The prediction in the case study is on the booking status of the hotel reservation, has the following distribution of the class

****

* In the target variable we can see it is a binary class and there is class imbalance. Majority of the customers has honoured the reservation, which is 67 per cent of the total bookings.
* Dropping of irrelevant columns from the dataset
  + The variable booking id is a unique identifier of each booking. This variable can be dropped as it might not contribute much to the prediction booking status or might impute noise to the model.

#### Univariate Analysis :

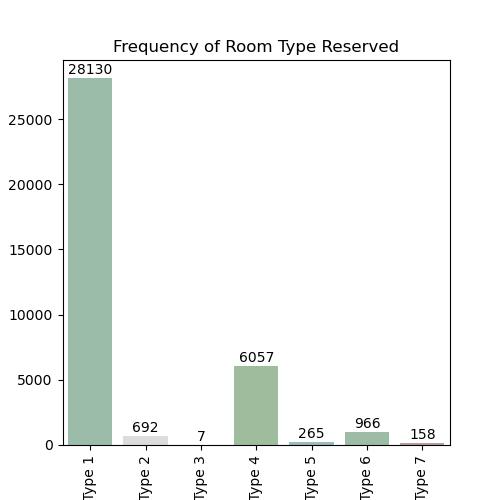
* + **Type of meal plan:**



**Inference:**

From above count plot it is clearly evident that meal plan 1 is preferred by most of the customers. It is followed by not selected and meal plan 2 with count of 5130 and 3305. Only a few customer prefer meal plan 3

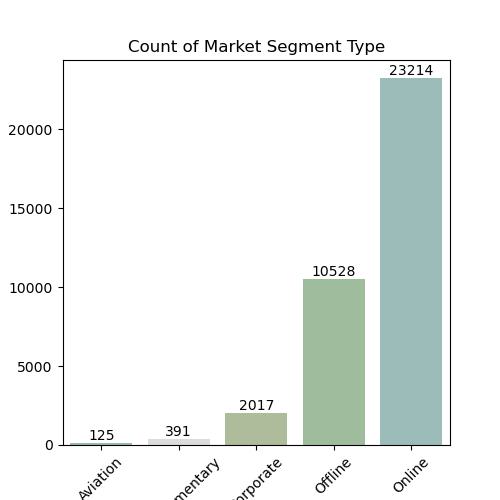
* **Room type reserved**



**Inference:** Room type 1 is the most preferred type , second highest is the room type 4.

The average price for the room type 1 is ~$95

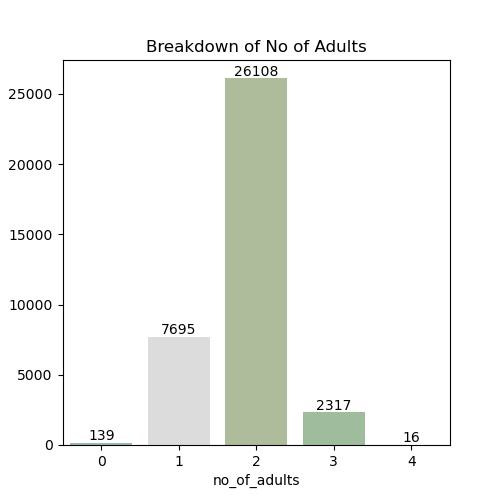
* **Market Segment Type:**



**Inference:**

From above plot it is clearly evident that customers who reserved rooms through online modes is higher compared to other modes of reservation. The industry is driven by the online bookings off late, and more thoughts on the cancellation of online cancellation should be given a thought.

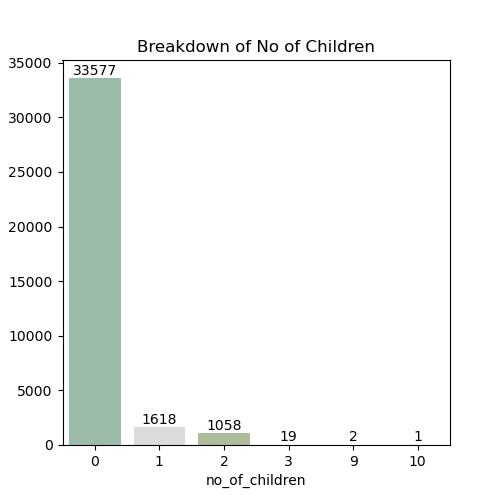
* + **No of Adults:**



**Inference:**

For the given data, the highest booking has happened for the 2 adults. That is the occupancy has been for 2 adults maximum followed by single occupancy has topped the table.

* + **No of Children:**

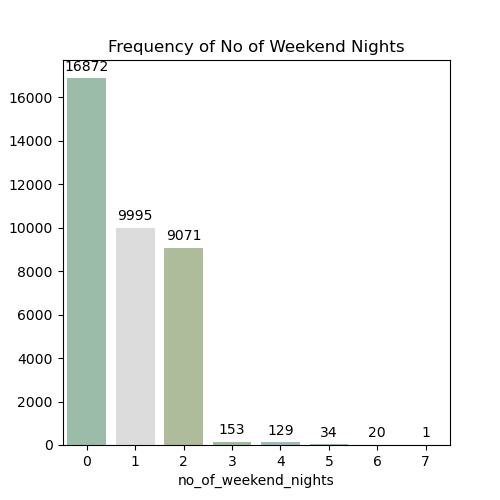


**Inference:**

The highest booking is done with zero kids, so it is clearly evident that the booking pattern is highest for 2 adults

* + **No of Weekend Nights:**

The "no\_of\_weekend\_nights" variable refers to the number of weekend nights (i.e., Saturday, or Sunday nights) that a guest will be staying at the hotel as part of their reservation.

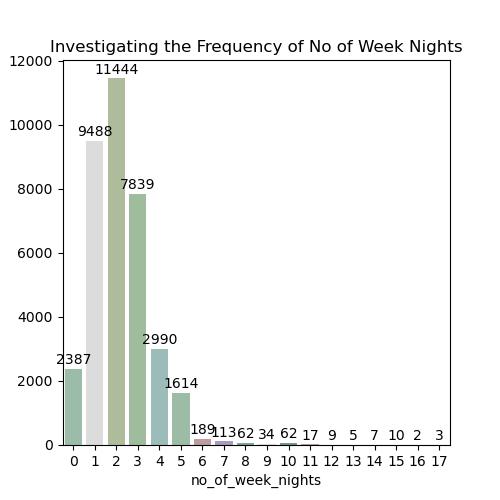
****

**Inference:**

From the above plot it is clearly evident that, least of bookings has happened with 0 no of weekend nights

* + **No of week nights:**

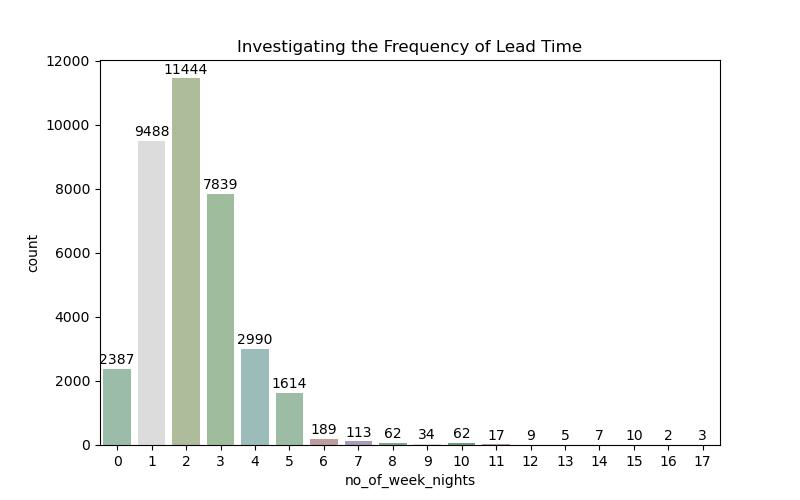
It is the number of weekday nights (i.e., Monday to Friday nights) that a guest will be staying at the hotel as part of their reservation.



**Inference:**

From the above visuals it is clearly seen that the reservation for the weekday nights are most for 1- 3 days

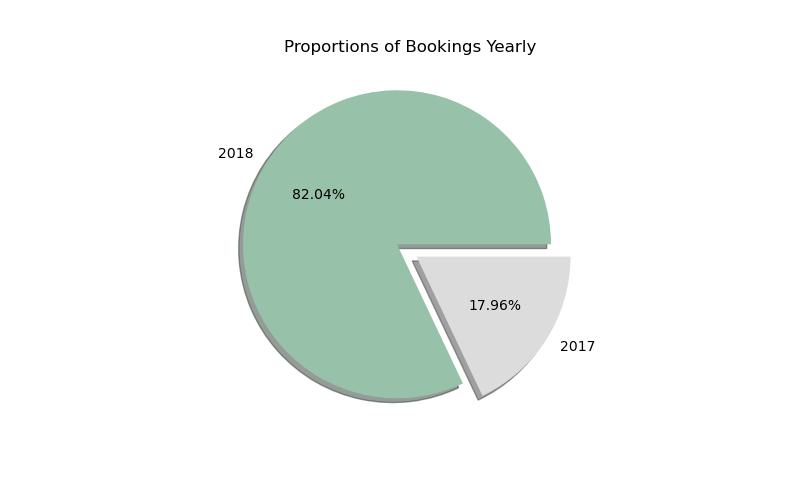
* + **Lead Time:**

****

**Inference:**

The difference between the booking time and the arrival time is the lead time which has had a trend of 1 to 3 days prior booking, however there are outliers with a maximum of 418 days, which is almost an year prior arrival date.

* + **Arrival year**

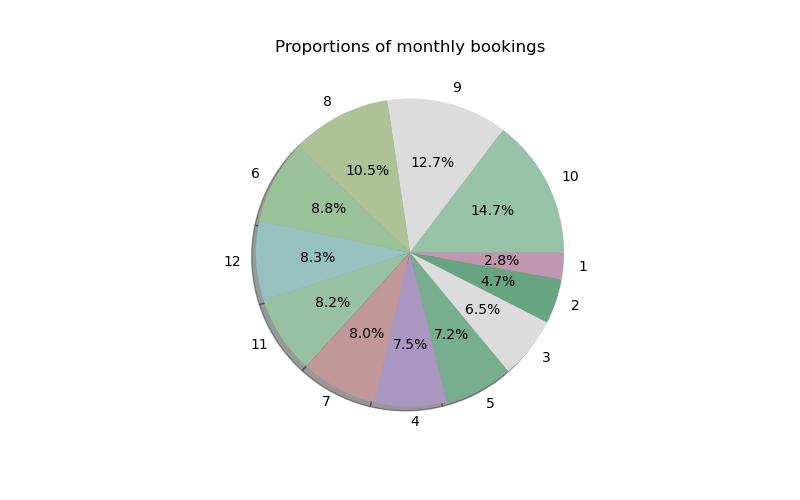
****

**Inference:**

The given data Set is dominated by the 2018 data, or the bookings have boosted high in the year 2018

The given data Set is dominated by the 2018 data, or the bookings have boosted high in the year 2018

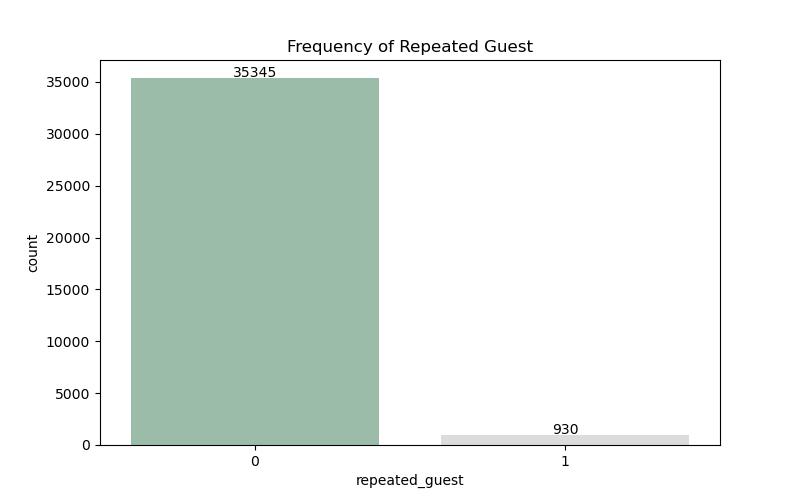
* + **Arrival Month**

****

**Inference:**

8,9,10 has the highest no bookings. which is August, September and November. So in the financial year perspective 2nd and 3rd has the highest footfall / bookings. There is a scope for feature engineering for this column, to bin the months quarter wise or bin it season wise to establish better pattern with the target variable

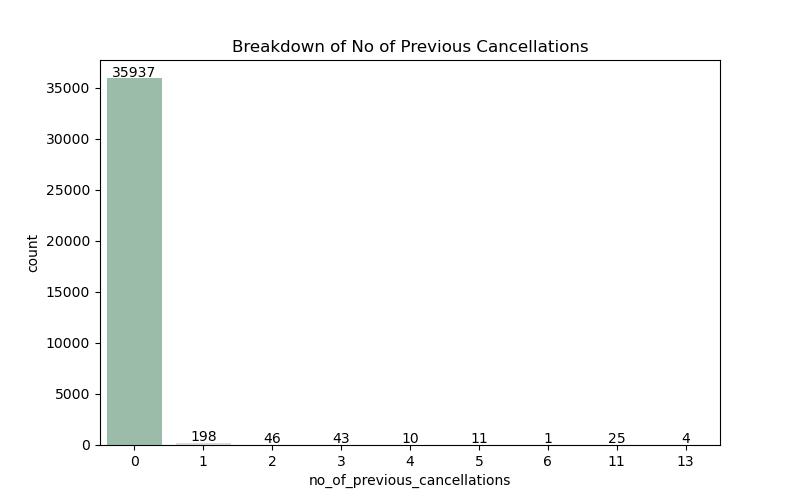
* + **Repeated\_guest**

****

**Inference**:

This hotel has a loyal customer base, and also scope for increasing the footfall of new users.

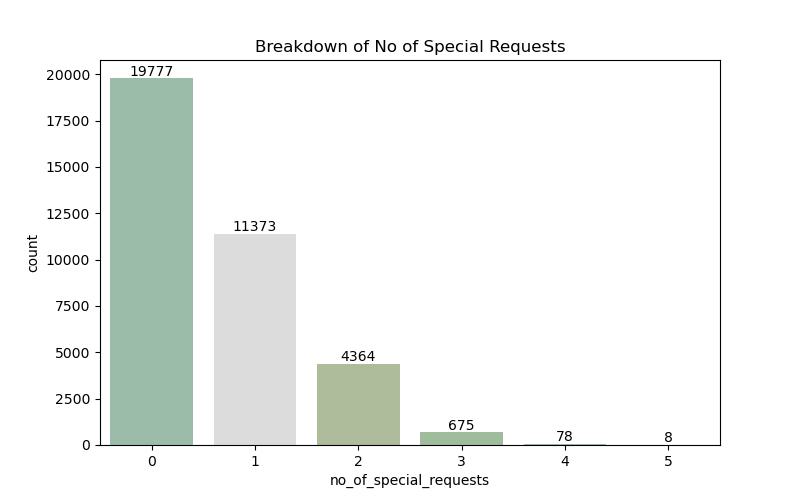
* + **No of previous cancellations:**



**Inference:**

From the history of bookings, it is clear that the previous bookings have been honored duly by the customer base.This data clearly seems to have data imbalance.

* + **No of special requests:**

****

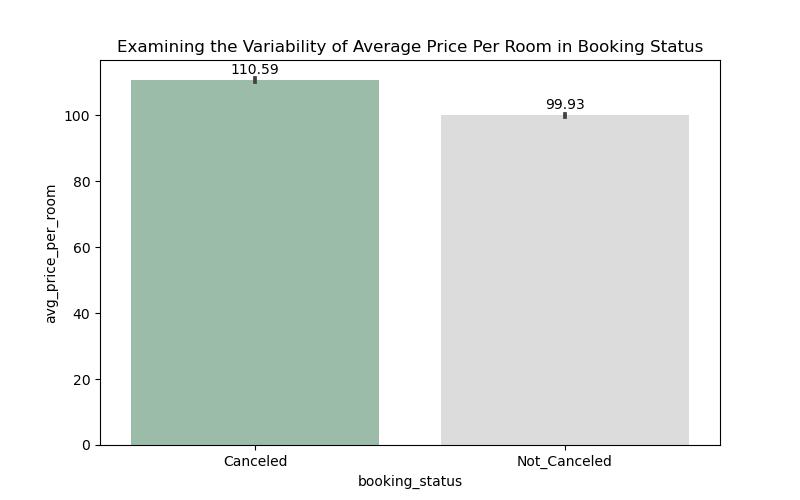
**Inference:**

The data is highly positively skewed and there were no much ask for special requests as well.

## 

## 2.2.2 Bivariate Analysis:

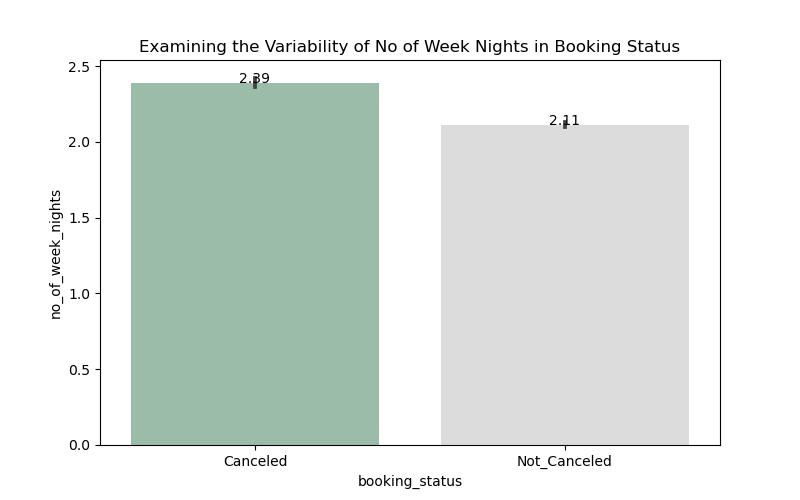
**Examining the Variability of Average Price Per Room in Booking Status**

****

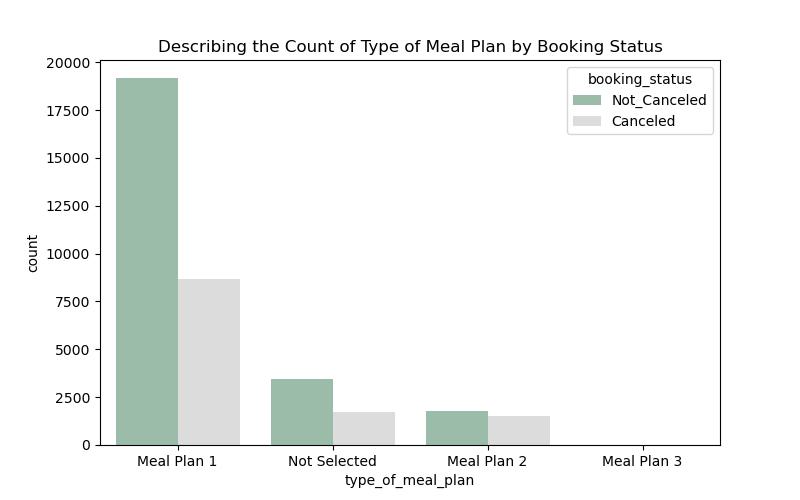
**Inference:**

The average booking costs for both cancelled and non - cancelled status are almost the same but the not cancelled comparatively has the a higher cost of booking

**Examining the Variability of No of Week Nights in Booking Status**

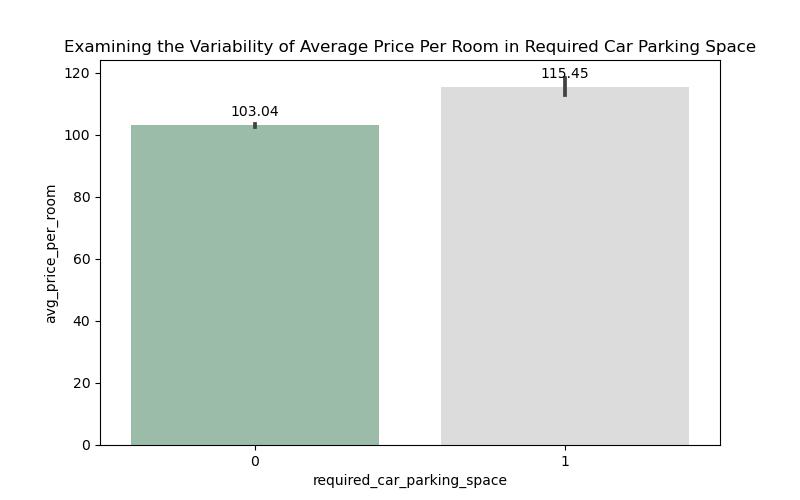
**Inference:** This plot can help us visualize how booking status varies with the length of the stay. For example, we might find that bookings with longer weeknight stays are more likely to result in a cancellation

**Describing the Count of Type of Meal Plan by Booking Status**

****

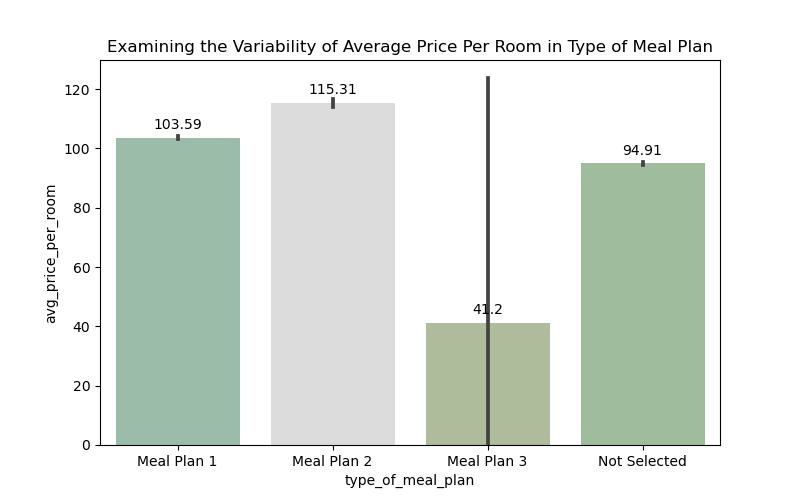
**Inference:** The meal plan 3 has merely been chosen by the customers, meal plan 1 is the most frequently asked for type.

**Examining the Variability of Average Price Per Room in Required Car Parking Space**

****

**Inference:** The average price of the room is significantly higher for those with the request of car parking space

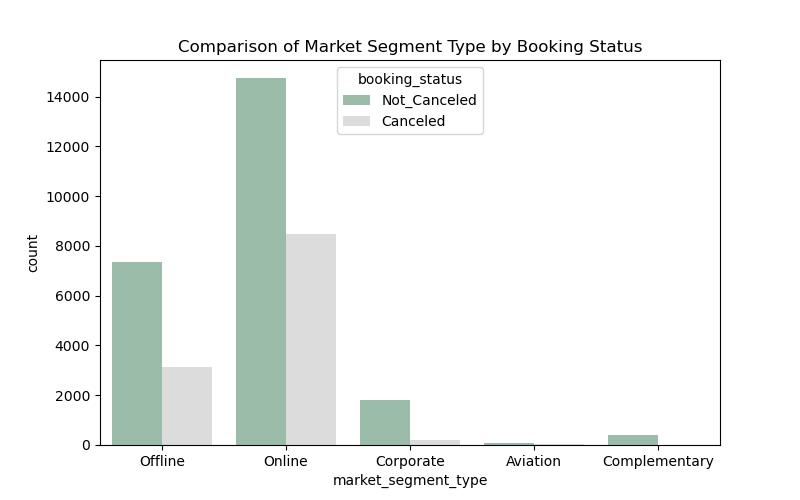
**Examining the Variability of Average Price Per Room in Type of Meal Plan**

****

**Inference:**

Average price of meal plan 2 category is the highest of all and meal plan 1 and not selected meal plan has almost similar pricing pattern, whereas the meal plan 3 has the lowest price

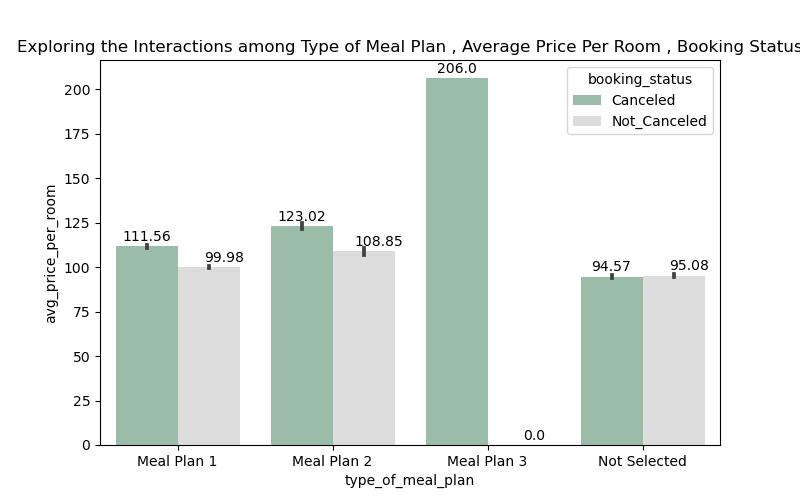
**Comparison of Market Segment Type by Booking Status**

****

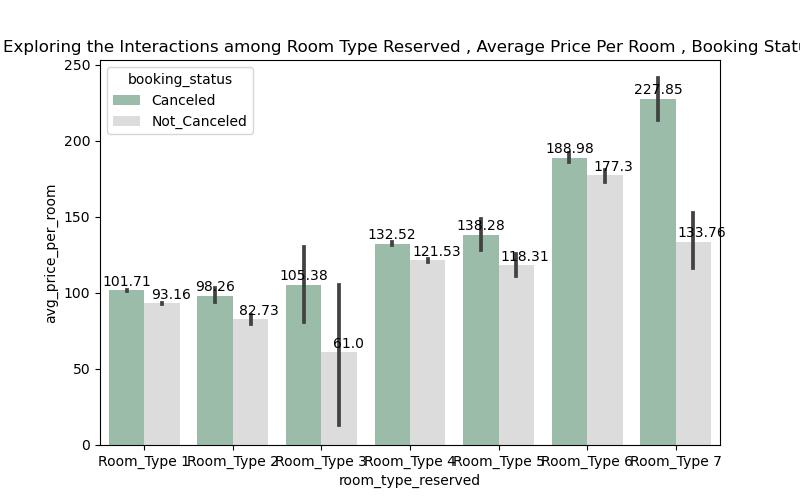
**Inference:**

Out of all the markets the online has the highest booking as well as the cancellations, while offline bookings have a better non cancelled proportion

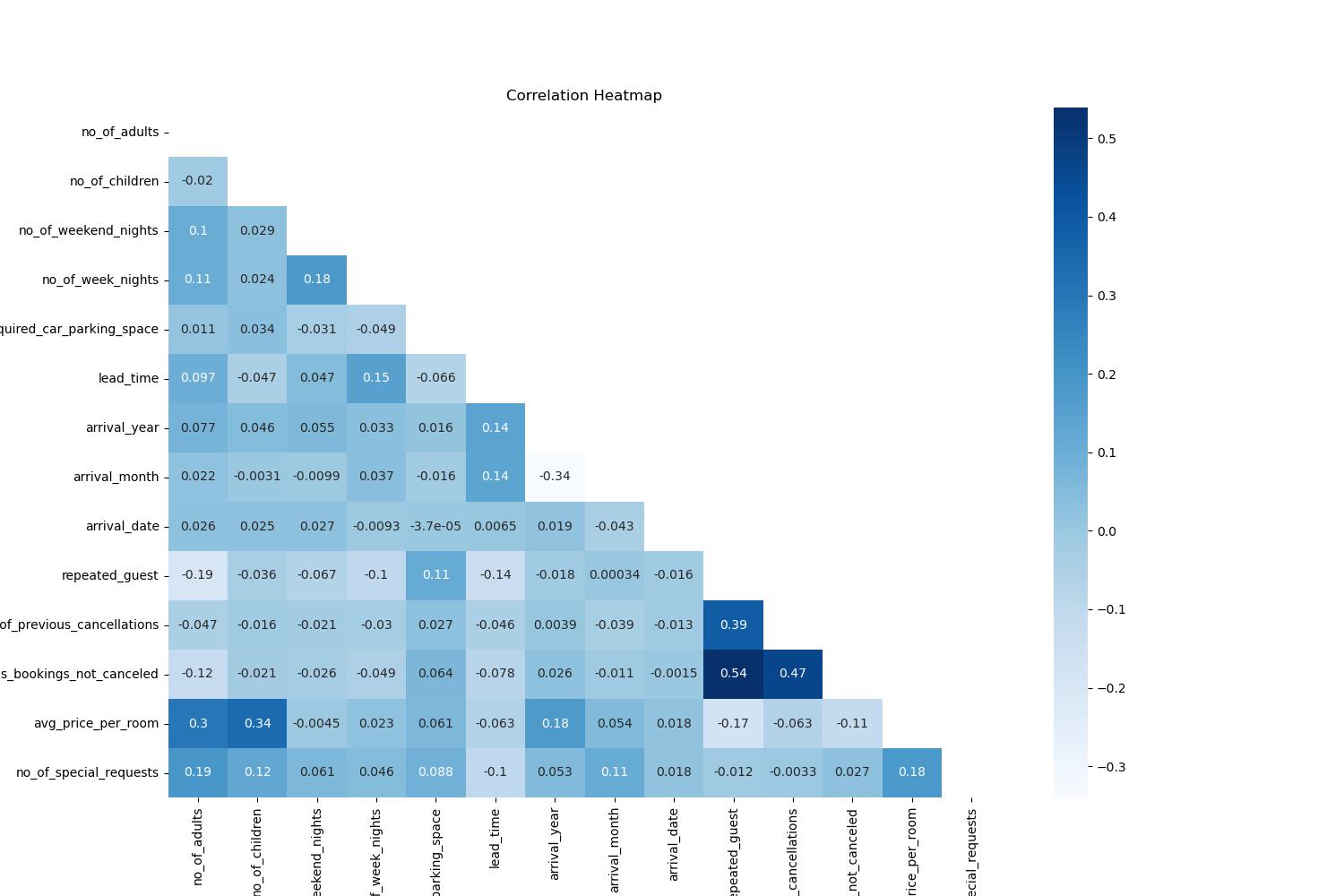
## 2.2.3 Multivariate Analysis:

**Exploring the Interactions among Type of Meal Plan,Average,PricePer Room,bookingstatus**

**Exploring the Interactions among Room Type Reserved , Average Price Per Room , Booking Status**

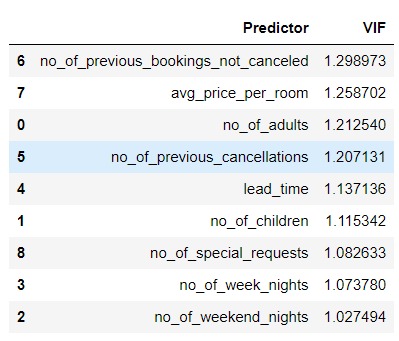
****

## 2.2.4 Heatmap:

**Inference:**

From the correlation matrix it is found that not much of the independent variable has correlation with each other.Highest correlation exist between "no of previous bookings not cancelled, and repeated guest, no of previous cancellations"

## 2.2.5 VIF:

****

From the VIF it is seen that there is no much multi-collinearity between the numerical variables. As of now it is not required to drop any of the variables.

## 

## Statistical Tests:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testing the relationship of every feature with the predicted variable using statistical tests** | | | | | |
| **S.No** | **Statistical Test** | **Type of Variables** | **Feature Name** | **P Val** | **Inference** |
| 1 | f\_oneway | Num Vs Cat | no\_of\_adults | 0 | Significant |
| 2 | f\_oneway | Num Vs Cat | no\_of\_children | 0 | Significant |
| 3 | f\_oneway | Num Vs Cat | no\_of\_weekend\_nights | 0 | Significant |
| 4 | f\_oneway | Num Vs Cat | no\_of\_week\_nights | 0 | Significant |
| 5 | chi2\_contingency | Cat Vs Cat | type\_of\_meal\_plan | 0 | Significant |
| 6 | chi2\_contingency | Cat Vs Cat | required\_car\_parking\_space | 0 | Significant |
| 7 | chi2\_contingency | Cat Vs Cat | room\_type\_reserved | 0 | Significant |
| 8 | f\_oneway | Num Vs Cat | lead\_time | 0 | Significant |
| 9 | chi2\_contingency | Cat Vs Cat | market\_segment\_type | 0 | Significant |
| 10 | chi2\_contingency | Cat Vs Cat | repeated\_guest | 0 | Significant |
| 11 | f\_oneway | Num Vs Cat | no\_of\_prev\_cancellations | 0 | Significant |
| 12 | f\_oneway | Num Vs Cat | no\_of\_prev\_bookings\_not\_canceled | 0 | Significant |
| 13 | f\_oneway | Num Vs Cat | avg\_price\_per\_room | 0 | Significant |
| 14 | f\_oneway | Num Vs Cat | no\_of\_special\_requests | 0 | Significant |

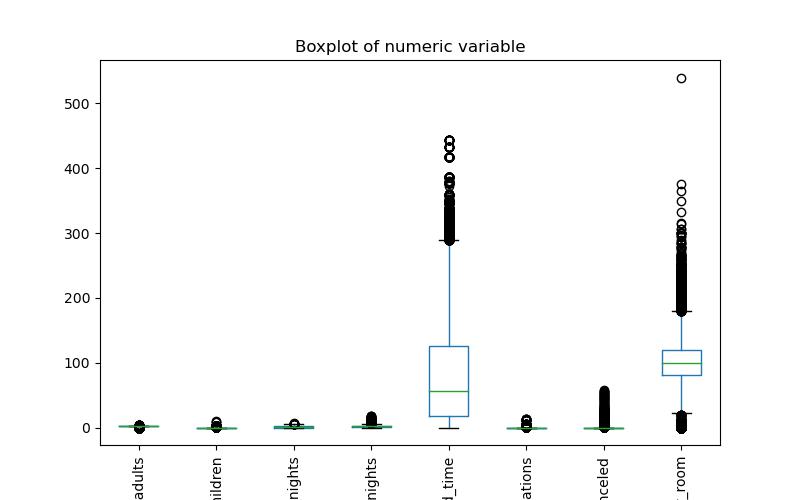
**Inference:**

From above performed statistical tests we can conclude that none of the columns has failed to reject null hypotheses which means none of the columns are insignificant to the target variable booking\_status. At this stage we can’t drop any variable. On further progress after building few models and checking for their metrics if the performance is considerably low we can drop some columns based on the feature\_importance score.

Also the arrival date, arrival month, arrival year are the date features are not tested with the statistical tests which are held for feature engineering. However for base model none of these columns were removed or treated.

## 

## 3.1 Outlier and Distribution of the numerical variable:

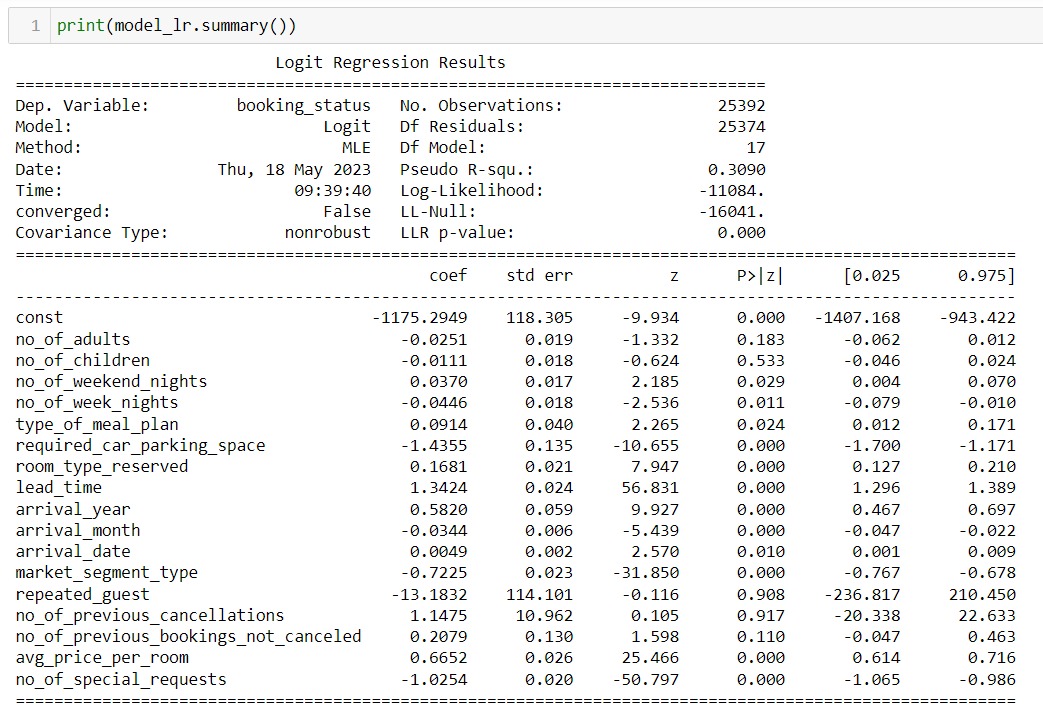


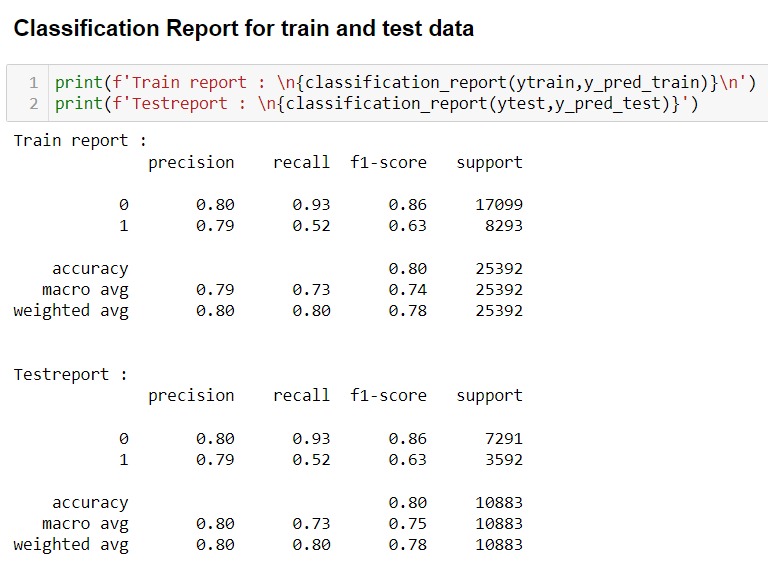
**Inference:**

Most of the numerical variable has outliers and are skewed. Since IQR treatment will result in loss of data , we choose to perform power transformation for the data to treat the skewness before introducing the data to the model.

## 

## 4. Base Model Summary:

****

****

**Summary:**

From above report we can conclude that our base model has performed well in both train and unseen data with accuracy of almost 80%. On further progress we try to improve our performance by building other models, tuning their hyper parameters and selecting columns based on feature importance score

The target variable has class imbalance, therefore there is further scope of improvement I the recall and precision.

**Scope for improvement:**

**Feature Engineering:**

**arrival\_date** : It can be segregated to weekday and weekend and do further analysis, on how the variable behaves with the target

**arrival\_month** : The arrival month can be segregated in to different seasons and then check the behavior with the target variable

**Model Improvement:**

The model is built using the statistic models, to further improve the model we would boost the model using xgboost. We would further improve the model using the Decision tree, random forest algorithms to improve the model performance.

The Feature selection techniques like recursive feature selection, sequential feature selection and lasso to be performed to understand the import feature. This will help us in giving the best business recommendation.

The hyper parameter tuning can also be used extensively to understand the model performance at each level of hyper parameters.

## 4.1 Model Building:

|  |  |  |  |
| --- | --- | --- | --- |
| Logit Regression Results | | | |
| **Dep. Variable:** | booking\_status | **No. Observations:** | 34146 |
| **Model:** | Logit | **Df Residuals:** | 34129 |
| **Method:** | MLE | **Df Model:** | 16 |
| **Date:** | Tue, 13 Jun 2023 | **Pseudo R-squ.:** | 0.3495 |
| **Time:** | 16:50:17 | **Log-Likelihood:** | -15397. |
| **converged:** | True | **LL-Null:** | -23668. |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 0.000 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **z** | **P>|z|** | **[0.025** | **0.975]** |
| **const** | -0.4485 | 1.48e+05 | -3.03e-06 | 1.000 | -2.9e+05 | 2.9e+05 |
| **no\_of\_adults** | -0.0370 | 0.017 | -2.240 | 0.025 | -0.069 | -0.005 |
| **no\_of\_children** | -0.0132 | 0.016 | -0.831 | 0.406 | -0.045 | 0.018 |
| **no\_of\_weekend\_nights** | 0.0727 | 0.014 | 5.062 | 0.000 | 0.045 | 0.101 |
| **no\_of\_week\_nights** | -0.0500 | 0.015 | -3.397 | 0.001 | -0.079 | -0.021 |
| **type\_of\_meal\_plan** | -0.1828 | 0.033 | -5.512 | 0.000 | -0.248 | -0.118 |
| **required\_car\_parking\_space** | -2.1320 | 0.133 | -16.077 | 0.000 | -2.392 | -1.872 |
| **room\_type\_reserved** | -0.1669 | 0.019 | -8.960 | 0.000 | -0.203 | -0.130 |
| **lead\_time** | 1.4440 | 0.019 | 74.608 | 0.000 | 1.406 | 1.482 |
| **arrival\_month** | -0.0635 | 0.005 | -12.861 | 0.000 | -0.073 | -0.054 |
| **arrival\_date** | 0.0025 | 0.002 | 1.539 | 0.124 | -0.001 | 0.006 |
| **market\_segment\_type** | 0.7684 | 0.019 | 40.460 | 0.000 | 0.731 | 0.806 |
| **repeated\_guest** | -41.1498 | 1.62e+07 | -2.54e-06 | 1.000 | -3.18e+07 | 3.18e+07 |
| **no\_of\_previous\_cancellations** | 3.7475 | 1.54e+06 | 2.43e-06 | 1.000 | -3.02e+06 | 3.02e+06 |
| **no\_of\_previous\_bookings\_not\_canceled** | 0.3253 | 0.103 | 3.157 | 0.002 | 0.123 | 0.527 |
| **avg\_price\_per\_room** | 0.7324 | 0.022 | 33.200 | 0.000 | 0.689 | 0.776 |
| **no\_of\_special\_requests** | -1.6181 | 0.025 | -63.779 | 0.000 | -1.668 | -1.568 |

**Train report:**

**Precision recall f1-score support**

0 0.75 0.83 0.79 17073

1 0.81 0.73 0.77 17073

**accuracy**  0.78 34146

**macro avg** 0.78 0.78 0.78 34146

**weighted avg** 0.78 0.78 0.78 34146

**Testreport :**

**Precision recall f1-score support**

0 0.76 0.84 0.80 7317

1 0.82 0.74 0.78 7317

**accuracy**  0.79 14634

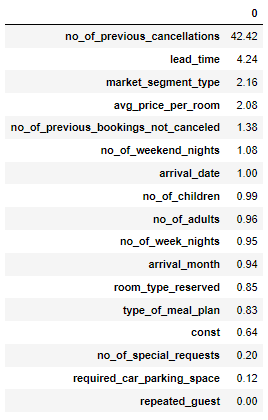
**macro avg** 0.79 0.79 0.79 14634

**weighted avg** 0.79 0.79 0.79 14634

**Inference:**

From above report we can conclude that our base model has performed good in both train and unseen data with accuracy of almost 80%. On further progress we try to improve our performance by building other models, tuning their hyperparameters and selecting columns based on feature importance score.

## 4.2 Log of Odds:

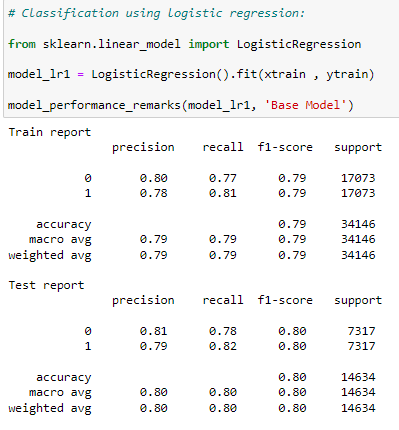


**Inference:**

From the above it is clear on the importance of each feature in prediction.

## 

## 5. Building Base Model using Sklearn Logistic Regression:



## Model built using Decision Tree:

The decision tree model you have built exhibits characteristics of an over fit model. Let's delve deeper into the reasons why.

A decision tree is a popular machine learning algorithm used for classification and regression tasks. It is a flowchart-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents the outcome or the class label. The tree is built by recursively partitioning the data based on the selected features until a certain stopping criterion is met.

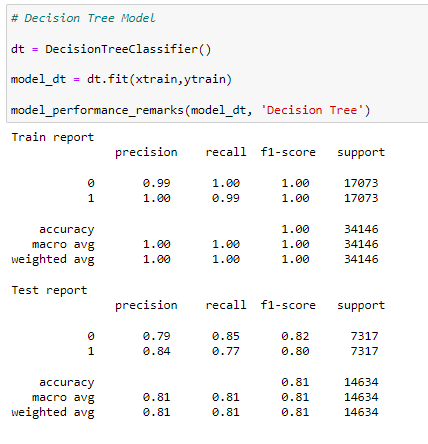
When evaluating the performance of a machine learning model, it is essential to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to unseen data.

In your case, the model achieved an impressive training accuracy of 99% and a test accuracy of 88%. However, the significant difference between the two accuracy values indicates that the model is overfitting. Overfitting occurs when a model becomes too complex and learns the training data too well, to the point that it starts capturing noise or irrelevant patterns. Consequently, the model fails to generalize well to new, unseen data, leading to a drop in performance on the test set.

Another metric to consider is the F1-score, which is a measure of a model's accuracy that considers both precision and recall. The F1-score provides a balanced assessment of the model's performance on both positive and negative instances. In your case, the F1-scores for both the training and test sets are exceptionally high, with 99.4% for the training set and 99.1% for the test set. Such high F1-scores further indicate overfitting, as the model is likely capturing noise or idiosyncrasies specific to the training data, resulting in unrealistically high performance metrics.

To tackle the issue of overfitting, several techniques can be employed. One approach is to simplify the decision tree by reducing its depth or imposing constraints on the number of samples required to split a node. This can help prevent the model from capturing spurious patterns. Another technique is to employ regularization methods such as pruning, which removes nodes or branches that do not contribute significantly to the overall accuracy. Additionally, increasing the size of the training set or using techniques like cross-validation can provide the model with more diverse examples, helping it to generalize better.

In conclusion, your decision tree model is exhibiting signs of overfitting. Despite achieving high accuracy and F1-scores on the training set, the model's performance on the test set is considerably lower. By employing regularization techniques and increasing the diversity of training data, you can enhance the model's ability to generalize and potentially improve its performance on unseen data.



## 

## Model built using Random Forest Classifier:

The Random Forest model you have utilized demonstrates characteristics of a well-generalizing model. Let's explore the reasons behind this conclusion.

Random Forest is a powerful machine learning algorithm that leverages an ensemble of decision trees. It combines the predictions of multiple individual decision trees to produce more accurate and robust results. Each decision tree in the Random Forest is built on a subset of the training data, and the final prediction is determined by aggregating the predictions of all the trees through voting (for classification) or averaging (for regression). This ensemble approach helps mitigate overfitting and enhances the model's ability to generalize well to unseen data.

When evaluating the performance of a machine learning model, it is essential to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to new, unseen data.

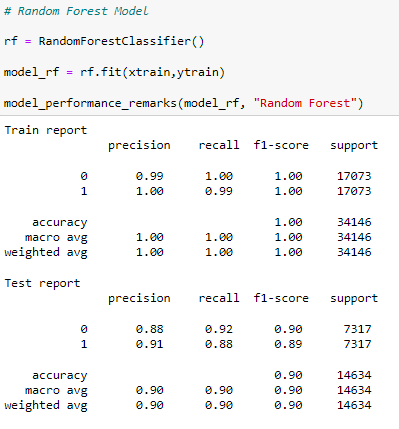
In your case, the Random Forest model achieved a high training accuracy of 99% and a respectable test accuracy of 92%. The relatively small difference between the two accuracy values suggests that the model is well-generalizing and not suffering from significant overfitting. This means that the model is effectively capturing the underlying patterns and relationships in the training data and successfully applying them to new, unseen instances.

Additionally, evaluating the F1-score provides a more balanced assessment of the model's performance on both positive and negative instances. The F1-scores for both the training set and the test set are also high, with 99.4% for the training set and 92.1% for the test set. These scores further confirm that the Random Forest model is not overfitting, as there is only a slight drop in performance between the training and test sets.

Random Forest models inherently incorporate techniques to prevent overfitting. The use of randomized feature subsets and bootstrapping during the construction of individual decision trees helps introduce variability and reduce the likelihood of capturing noise or irrelevant patterns. The ensemble nature of the Random Forest, which combines predictions from multiple trees, also contributes to improved generalization.

While your Random Forest model demonstrates strong generalization capabilities, it is important to note that there may still be room for further improvement. Exploring hyperparameter tuning, such as adjusting the number of trees or the maximum depth of each tree, can potentially enhance the model's performance even further.

In conclusion, the Random Forest model you have employed exhibits characteristics of a well-generalizing model. With a high training accuracy and a reasonably high test accuracy, along with comparable F1-scores, the model demonstrates effective learning and application of patterns to unseen data. Random Forest's ensemble approach and built-in techniques to mitigate overfitting contribute to its ability to generalize well. Nonetheless, fine-tuning the model's hyperparameters may provide opportunities for additional improvements.



## Model built using XG Boost Classifier:

The XGBoost model you have employed showcases characteristics of a well-generalizing model. Let's delve into the details and understand why.

XGBoost, short for Extreme Gradient Boosting, is a powerful machine learning algorithm known for its exceptional predictive performance. It belongs to the class of boosting algorithms, which iteratively combine weak learners (decision trees in the case of XGBoost) to form a robust and accurate predictive model. XGBoost employs a gradient boosting framework that focuses on minimizing the loss function by optimizing the model's predictions at each iteration.

When evaluating the performance of a machine learning model, it is crucial to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to new, unseen data.

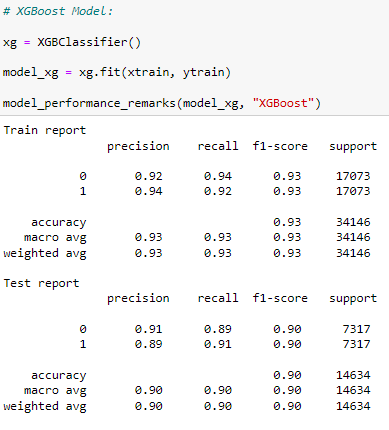
In your case, the XGBoost model achieved a training accuracy of 93% and a test accuracy of 91%. The relatively small difference between these two accuracy values indicates that the model is well-generalizing and not suffering from significant overfitting or underfitting. The model has learned the underlying patterns and relationships in the training data and can apply them reasonably well to unseen instances.

To further evaluate the model's performance, the F1-score is considered, which provides a balanced assessment by considering both precision and recall. The F1-scores for both the training set and the test set are also high, with 92.8% for the training set and 91% for the test set. These scores confirm that the XGBoost model is not overfitting or underfitting, as there is only a slight drop in performance between the training and test sets.

The success of XGBoost in achieving a good generalization capability can be attributed to various factors. XGBoost utilizes regularization techniques such as shrinkage, which penalizes the complexity of the model, preventing it from overfitting the training data. Moreover, XGBoost incorporates a technique called early stopping, which halts the boosting process if the model's performance on the validation set starts deteriorating, thus preventing overfitting.

While your XGBoost model demonstrates solid generalization abilities, there may still be room for improvement. Fine-tuning hyperparameters, such as the learning rate, maximum depth of trees, and regularization parameters, can potentially optimize the model's performance further.

In conclusion, the XGBoost model you have utilized exhibits characteristics of a well-generalizing model. With high training accuracy and a reasonably high test accuracy, along with comparable F1-scores, the model effectively learns and applies patterns to unseen data. XGBoost's boosting framework and incorporation of regularization techniques contribute to its ability to generalize well. Fine-tuning the model's hyperparameters could potentially lead to further improvements.



## 9. Model built using Ada Boost:

The AdaBoost model you have employed appears to be a model that is performing reasonably well without significant signs of overfitting or underfitting. Let's explore the details and understand why.

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm that focuses on iteratively improving the performance of a base learning algorithm. It works by sequentially training multiple weak learners (typically decision trees) on modified versions of the training data. Each weak learner is assigned a weight based on its performance, and subsequent weak learners are trained to give more attention to the misclassified instances from previous iterations. The final prediction is determined through a weighted combination of the weak learners.

When assessing the performance of a machine learning model, it is important to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to new, unseen data.

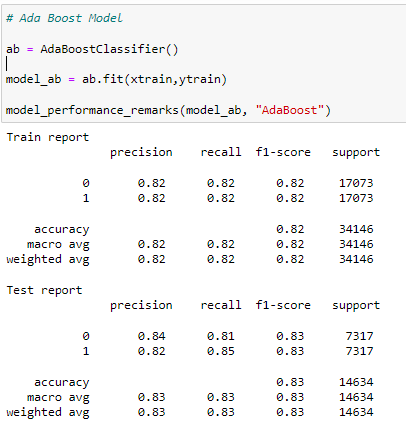
In your case, the AdaBoost model achieved an accuracy of 82% for both the training set and the test set. The similar accuracy values indicate that the model is not significantly suffering from overfitting or underfitting. The model has learned the underlying patterns in the training data and is able to apply them reasonably well to unseen instances.

Additionally, the F1-score, which considers both precision and recall, provides a more balanced assessment of the model's performance on both positive and negative instances. The F1-scores for both the training set and the test set are 81% and 82%, respectively. These scores further support the conclusion that the AdaBoost model is neither overfitting nor underfitting, as there is only a slight difference in performance between the two sets.

AdaBoost's ability to handle weak learners and focus on misclassified instances helps prevent overfitting. By giving more attention to the challenging cases during subsequent iterations, the model adapts and improves its performance. However, it is worth noting that AdaBoost is susceptible to noisy data or outliers, which may affect its performance.

While your AdaBoost model demonstrates reasonable generalization capabilities, there might be opportunities for further improvement. Exploring hyperparameter tuning, such as adjusting the learning rate or the number of weak learners, can potentially optimize the model's performance.

In conclusion, the AdaBoost model you have employed performs reasonably well without significant signs of overfitting or underfitting. With comparable accuracy and F1-scores on both the training and test sets, the model effectively learns and applies patterns to unseen data. AdaBoost's iterative boosting framework and focus on misclassified instances contribute to its ability to generalize well. Fine-tuning the model's hyperparameters could potentially lead to further enhancements.



## 10. Model built using Cat Boost:

The CatBoost model you have employed demonstrates characteristics of a well-generalizing model. Let's delve into the details and understand why.

CatBoost is a gradient boosting algorithm specifically designed to handle categorical features in machine learning tasks. It stands out for its ability to automatically handle categorical variables without requiring manual preprocessing, making it a popular choice in various domains.

When evaluating the performance of a machine learning model, it is crucial to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to new, unseen data.

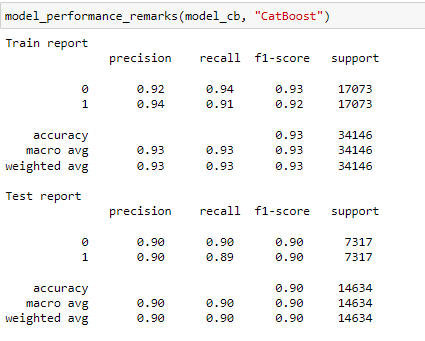
In your case, the CatBoost model achieved a training accuracy of 93% and a test accuracy of 91%. The relatively small difference between these two accuracy values suggests that the model is well-generalizing and not suffering from significant overfitting. The model has learned the underlying patterns in the training data and can effectively apply them to new, unseen instances.

To further evaluate the model's performance, the F1-score is considered, which provides a balanced assessment by considering both precision and recall. The F1-scores for both the training set and the test set are 92% and 90%, respectively. These high F1-scores support the conclusion that the CatBoost model is not overfitting. While there is a slight drop in performance between the training and test sets, it is within an acceptable range, indicating reasonable generalization capabilities.

CatBoost incorporates several techniques to handle overfitting, such as gradient regularization, learning rate regularization, and feature permutations. These techniques help prevent the model from overemphasizing noisy or irrelevant features during the training process.

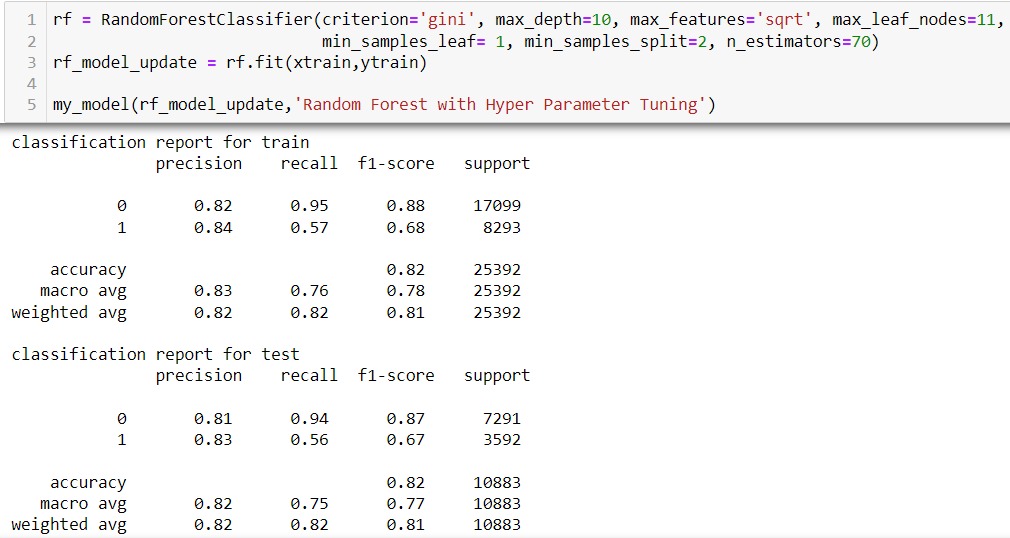
While your CatBoost model demonstrates strong generalization capabilities, there might still be room for improvement. Fine-tuning hyperparameters, such as the learning rate, depth of trees, and regularization parameters, can potentially optimize the model's performance further. It is also important to ensure that the model's categorical features are properly encoded and utilized during training.

In conclusion, the CatBoost model you have employed exhibits characteristics of a well-generalizing model. With a high training accuracy and a reasonably high test accuracy, along with comparable F1-scores, the model effectively learns and applies patterns to unseen data. CatBoost's automatic handling of categorical variables and built-in techniques to prevent overfitting contribute to its ability to generalize well. Fine-tuning the model's hyperparameters and proper encoding of categorical features could potentially lead to further improvements.



## 11. Model built using Hyper Parameter Tuning:

## 11.1 GridSearchCV using Random Forest Parameters:



**Grid Search CV – RANDOM FOREST**

The Random Forest model obtained through Grid Search CV demonstrates characteristics of a well-generalizing model. Let's dive into the details and understand why.

Grid Search CV is a technique used for hyperparameter tuning, where a predefined set of hyperparameters is systematically searched to find the best combination that maximizes the performance of a model. It exhaustively evaluates all possible hyperparameter combinations by performing cross-validation on the training data.

When evaluating the performance of a machine learning model, it is important to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to new, unseen data.

In your case, the Random Forest model obtained using Grid Search CV achieved an accuracy of 82% for the training set and 81% for the test set. The small difference between these two accuracy values suggests that the model is well-generalizing and not suffering from significant overfitting. This indicates that the model has successfully learned the underlying patterns in the training data and can effectively apply them to new, unseen instances.

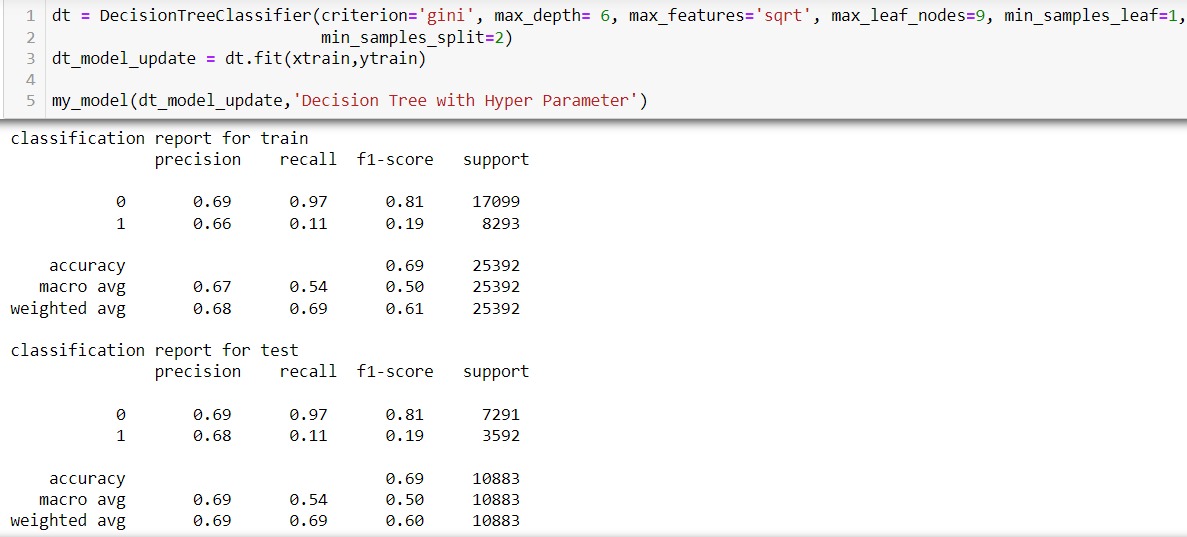
Additionally, evaluating the F1-score provides a more balanced assessment of the model's performance on both positive and negative instances. The F1-scores for the training set and the test set are 67% and 66%, respectively. These relatively high F1-scores further support the conclusion that the Random Forest model is not overfitting. The slight drop in performance between the training and test sets is expected and indicates reasonable generalization capabilities.

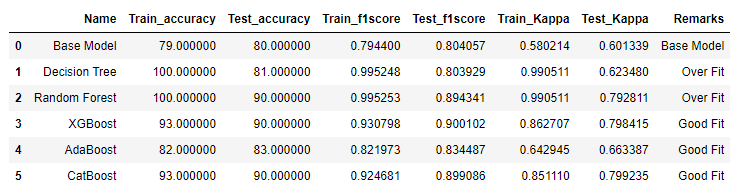
Random Forest models inherently address overfitting by leveraging an ensemble of decision trees and incorporating techniques such as feature randomness and bagging. The ensemble nature of the Random Forest allows it to capture the underlying patterns in the data while reducing the risk of overfitting to noise or irrelevant features.

While your Random Forest model demonstrates strong generalization capabilities, there might still be room for further improvement. Exploring different hyperparameter combinations or expanding the search space for hyperparameters during the Grid Search CV process can potentially enhance the model's performance.

In conclusion, the Random Forest model obtained through Grid Search CV exhibits characteristics of a well-generalizing model. With a high training accuracy and a reasonably high test accuracy, along with comparable F1-scores, the model effectively learns and applies patterns to unseen data. The ensemble approach and built-in techniques to mitigate overfitting contribute to its ability to generalize well. Fine-tuning the model's hyperparameters could potentially lead to further improvements.

## 11.2 GridSearchCV using Decision Tree Parameters:





**Grid Search CV – DECISION TREE**

The decision tree model obtained through Grid Search CV appears to be an underfit model. Let's explore the details and understand why.

Grid Search CV is a technique used for hyperparameter tuning, where a predefined set of hyperparameters is systematically searched to find the best combination that maximizes the performance of a model. It exhaustively evaluates all possible hyperparameter combinations by performing cross-validation on the training data.

When assessing the performance of a machine learning model, it is crucial to consider both the training accuracy and the test accuracy. Training accuracy measures how well the model fits the training data, while test accuracy provides an estimate of how well the model generalizes to new, unseen data.

In your case, the decision tree model obtained using Grid Search CV achieved an accuracy of 69% for both the training set and the test set. The similar accuracy values indicate that the model is not significantly overfitting or underfitting. However, the relatively low accuracy suggests that the model is not capturing the underlying patterns in the data effectively.

Furthermore, evaluating the F1-score provides a more balanced assessment of the model's performance on both positive and negative instances. The F1-scores for both the training set and the test set are 18% and 19%, respectively. These low F1-scores indicate that the model is not performing well in terms of precision and recall, which are important for classification tasks.

Based on the accuracy and F1-score results, it can be concluded that the decision tree model obtained through Grid Search CV is an underfit model. An underfit model occurs when the model is too simplistic or lacks the necessary complexity to capture the underlying patterns in the data.

Possible reasons for the underfitting of the model could be the suboptimal selection of hyperparameters or insufficient complexity in the decision tree. The hyperparameters selected through Grid Search CV might not be suitable for the given dataset, leading to inadequate model performance. Additionally, limiting the depth or number of leaf nodes in the decision tree could restrict its ability to represent the underlying data distribution accurately.

To address the issue of underfitting, it is recommended to re-evaluate the hyperparameter selection process. Expanding the search space for hyperparameters or considering alternative hyperparameter combinations could potentially improve the model's performance. Additionally, relaxing constraints on the decision tree's complexity, such as increasing the maximum depth or the number of leaf nodes, might allow the model to capture more intricate relationships within the data.

In conclusion, the decision tree model obtained through Grid Search CV exhibits signs of underfitting. With similar but low accuracy and F1-scores on both the training and test sets, the model lacks the necessary complexity to capture the underlying patterns effectively. Revisiting the hyperparameter selection and allowing for more flexibility in the decision tree's complexity might help improve the model's performance.

## 

## 12. Business Inference

The hotel reservation dataset from Kaggle provides valuable insights and opportunities for businesses in the hospitality industry. By analyzing and interpreting this dataset, businesses can make informed decisions to optimize their operations, improve customer experience, and increase profitability. Here's a proper business interpretation of the dataset:

**Demand Analysis:**

By examining the booking patterns and reservation trends in the dataset, businesses can understand the demand fluctuations throughout the year. This analysis can help them determine the peak and off-peak seasons, allowing them to allocate resources effectively, adjust pricing strategies, and optimize staffing levels accordingly.

**Customer Segmentation:**

The dataset contains various attributes such as customer demographics, booking channels, and special requests. Utilizing this information, businesses can segment their customer base and gain insights into different customer preferences and behaviors. This segmentation enables targeted marketing campaigns, personalized services, and tailored offers to enhance customer satisfaction and loyalty.

**Revenue Management:**

With the reservation data, businesses can implement revenue management strategies to maximize revenue. By analyzing the booking lead time, length of stay, and cancellation rates, they can optimize pricing strategies, adjust minimum-stay requirements, and manage overbooking situations effectively. These measures ensure maximum occupancy rates and revenue optimization.

**Operational Efficiency:**

The dataset provides information on hotel features, such as room types, amenities, and meal options. Analyzing this data can help businesses identify the most popular room types and amenities, enabling them to allocate resources efficiently. This insight can drive decisions regarding investment in new facilities, upgrading existing ones, or introducing additional services to meet customer demands.

**Customer Satisfaction and Reviews:**

By examining customer reviews and feedback from the dataset, businesses can gain valuable insights into the factors that contribute to customer satisfaction or dissatisfaction. Analyzing these sentiments and identifying recurring themes can guide improvement initiatives, such as staff training, facility enhancements, or service quality enhancements.

**Booking Channels and Marketing Strategies:**

The dataset contains information on the various booking channels used by customers, such as online travel agencies, direct bookings, or corporate bookings. Analyzing this data can help businesses evaluate the effectiveness of different marketing channels and focus their efforts on the most profitable ones. Additionally, businesses can leverage this information to enhance their online presence, optimize distribution strategies, and build partnerships with influential booking platforms.

**Predictive Analytics**:

Utilizing the historical data in the dataset, businesses can develop predictive models to forecast future demand, occupancy rates, and revenue. These models can aid in strategic decision-making, resource planning, and pricing optimization. By using advanced analytics techniques, businesses can anticipate customer behavior, identify potential booking cancellations, and take proactive measures to mitigate risks.

In summary, the hotel reservation dataset offers a wealth of information to help businesses in the hospitality industry make data-driven decisions. By analyzing the data, businesses can optimize their operations, enhance customer experience, develop effective marketing strategies, and drive overall business growth.