

# HOTEL BOOKING CANCELLATION PREDICTION

## Post Graduate Program in Data Science and Engineering

**Location:Bangalore**

**Batch: PGPDSE - APR22**

**Submitted by**

EVELINE JEMIMA

SONA PAUL

HARSH PRIYADHARSHAN

ANKIT KUMAR

PARTHO SARATHI BISWAS

**Mentored by:** PRATIK SONAR

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Sl. NO | Topic | Page No |
| 1 | INTRODUCTION | 1 |
| 2 | BUSINESS PROBLEM STATEMENT | 2 |
| 3 | LITERATURE REVIEW | 3 |
| 4 | DATASET INFORMATION | 4 |
| 5 | VARIABLE IDENTIFICATION | 5 |
| 6 | APPROACH | 6 |
| 7 | DATA PREPROCESSING | 7 |
| 8 | STATISTICAL ANALYSIS | 11 |
| 9 | UNIVARIATE, BIVARIATE & MULTIVARIATE ANALYSIS | 13 |
| 10 | ENCODING | 22 |
| 11 | SCALING &  TRANSFORMATION | 23 |
| 12 | CHECKING FOR CLASSIMBALANCE | 24 |
| 13 | MODEL BUILDING | 25 |
| 14 | CONCLUSION | 43 |
| 15 | REFERENCES | 44 |

# INTRODUCTION:

Hotel booking cancellation prediction is a critical facet of the hospitality industry, influencing operational efficiency, revenue management, and customer satisfaction. This project aims to employ advanced machine learning techniques to forecast the likelihood of hotel booking cancellations, empowering establishments to make informed decisions and optimize resource utilization.

The significance of this project lies in its potential to revolutionize the way hotels manage their reservations. By leveraging historical booking data, alongside a multitude of variables such as booking lead time, seasonality, room type, and customer demographics, predictive models can be developed to anticipate the probability of a reservation being canceled. This foresight equips hoteliers with valuable insights to adapt their strategies, allocate resources effectively, and potentially mitigate revenue loss.

The methodology involves the utilization of various machine learning algorithms, such as logistic regression, decision trees, random forests to analyze and predict cancellation. Data preprocessing and feature engineering play a crucial role in enhancing the accuracy of these models. By identifying patterns and correlations within the data, the models can effectively learn and predict the likelihood of a booking getting canceled.

Moreover, this project prioritizes the interpretability of the models to enable hotel management to comprehend the reasons behind these forecasts. By understanding the key factors contributing to booking cancellations, hotels can proactively implement strategies to reduce cancellation rates. Whether it's through personalized offers, flexible cancellation policies, or targeted customer engagement, this knowledge can drive proactive measures to decrease cancellations and enhance customer satisfaction.

The implications of accurate cancellation prediction are far-reaching. It allows hotels to optimize their overbooking strategies, staff allocation, and inventory management, thereby reducing revenue loss resulting from cancellations. Additionally, it enables personalized customer service, as hotels can anticipate and cater to the needs of guests who are more likely to cancel, potentially retaining their business through tailored approaches.Ultimately, the project endeavors to create a robust, adaptable, and scalable solution that can be integrated into hotel management systems. By harnessing the power of predictive analytics, this project aims to assist hotels in making data-driven decisions, enhancing operational efficiency, and ultimately improving the overall guest experience.

# BUSINESS PROBLEM STATEMENT:

**Business Problem Understanding:**The business problem lies in the inability of hotels to accurately predict booking cancellations, leading to revenue loss and operational inefficiencies. The lack of a reliable forecasting mechanism hinders effective resource allocation, overbooking strategies, and personalized customer service. This project aims to address this challenge by leveraging machine learning to develop a predictive model. The goal is to empower hotels with actionable insights, enabling them to proactively manage cancellations, optimize revenue streams, and enhance overall operational efficiency. The solution holds the potential to transform the hospitality industry by providing a data-driven approach to mitigate the impact of unpredictable booking cancellations.

**Business Objective:** The primary business objective for implementing a hotel booking cancellation prediction system is to enhance operational efficiency, maximize revenue, and improve overall customer satisfaction within the hospitality industry. The specific business objectives include:

1. **Optimizing Resource Allocation:**
   * Improve efficiency in staff management, housekeeping, and other operational aspects by accurately predicting booking cancellations.
   * Minimize the impact of unexpected cancellations on staffing levels and resource allocation, ensuring optimal utilization of hotel resources.
2. **Revenue Maximization:**
   * Implement proactive overbooking strategies based on accurate cancellation predictions to capitalize on potential demand without risking overcapacity.
   * Identify opportunities to offer additional services or promotions to guests at risk of canceling, thereby retaining revenue that might otherwise be lost.
3. **Data-Driven Decision Making:**
   * Foster a culture of data-driven decision-making within the organization by leveraging predictive analytics to anticipate booking cancellations.
   * Provide actionable insights to hotel management for strategic planning and the development of targeted marketing and retention initiatives.
4. **Enhancing Customer Experience:**
   * Implement personalized services and retention strategies for guests identified as likely to cancel, improving overall customer satisfaction.
   * Proactively communicate with guests facing potential cancellations, offering flexible options and demonstrating a commitment to customer-centric practices.

# DATASET INFORMATION

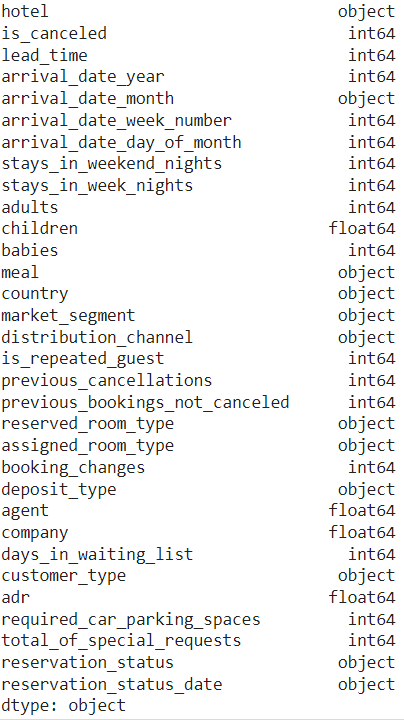
The dataset has been published in Kaggle. The dataset contains 32 attributes with 119390 observations. The datasets contain a mix of features with floating-point, integer, and string values. For this study, we intend to use the most common data attributes that can be readily available for any business. Hence, we have chosen 'reserved\_room\_type', 'assigned\_room\_type', 'booking\_changes' as our predicting variables and ‘is\_cancelled’ as our response variable. Our target variable is labeled with two classes. Hence, this scenario falls under the binary classification problem.

The booking changes feature indicates :Number of changes/amendments made to the booking from the moment the booking was entered on the Property Management System until the moment of check-in or cancellation. Calculated by adding the number of unique iterations that change some of the booking attributes, namely: persons, arrival date, nights, reserved room type or meal. The ‘leadtime’ feature: Number of changes/amendments made to the booking from the moment the booking was entered on the Property Management System until the moment of check-in or cancellation. Calculated by adding the number of unique iterations that change some of the booking attributes, namely: persons, arrival date, nights, reserved room type or meal.

Previous Cancellations feature indicates INumber of previous bookings that were canceled by the customer prior to the current booking. In case there was no customer profile associated with the booking, the value is set to 0. Otherwise, the value is the number of bookings with the same customer profile created before the current booking and canceled.

# VARIABLE IDENTIFICATION

Variables: The 32 columns are listed below :



**Fig:**Variables

# APPROACH

The approach to build a Hotel Booking Cancellation Prediction model typically involves the following steps:

1. Data collection: The project team will collect the relevant data from various sources. They will ensure the data is accurate, complete, and relevant to the project goals.
2. Data preprocessing: Clean and preprocess the data to remove outliers, missing values, and other errors. Aggregate the data into a format suitable for training machine learning models.
3. Feature engineering: Create new features or transform existing ones that can improve the accuracy of the model. For example, features such as countries, arrival\_date\_month, market\_segment may be included to help classify hotel booking cancellation.
4. Model selection: Select an appropriate machine learning algorithm for the Hotel Booking Cancellation Prediction task. Commonly used algorithms include logistic regression, decision trees, and random forests.
5. Model training: Train the selected machine learning model on the preprocessed data, using techniques such as cross-validation to optimize model parameters.
6. Model evaluation: Evaluate the trained model's accuracy and performance using metrics such as accuracy, precision and recall.
7. Model deployment: Deploy the trained model in a production environment to make predictions on new data. Integrate the model with inventory management systems to provide real-time insights and recommendations.
8. Model monitoring and updating: Monitor the model's performance over time and retrain or update the model as needed to ensure its accuracy and effectiveness.

Overall, the approach to Hotel Booking Cancellation Prediction involves a combination of data processing, feature engineering, machine learning, and integration with inventory management systems to provide real-time insights and recommendations to businesses.

# TARGET VARIABLE

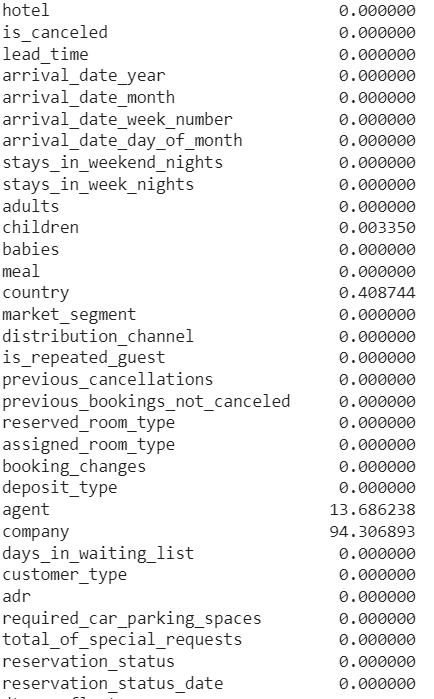
The target variable of the above dataset is is\_cancelled. We have to predict whether the booking will get cancelled or not. In the above dataset, 63% of bookings are not going to get cancelled and 37% of are going for cancellation. We observe that this data is very much balanced.

# DATA PREPROCESSING:

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model .When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this ,we use data pre-processing task.

A real-world data generally contains noises, missing values, and may be in an unusable format which cannot be directly used for machine learning models. Data pre- processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

**Missing Value Treatment:**



The next step of data pre-processing is to handle missing data in the datasets. Ifour dataset contains some missing data, then it may create a huge problem for our machine learning model.Hence it is necessary to handle missing values present in the dataset.

**Fig**:Missing Value

In dataset we have missing values in some features. children feature is integer datatype, replacing nan values with 0. Filling it with median imputation. Country column is categorical. So we are filling the nan values with mode value. We are dropping the column company since it is having null values of 94%.

## Imputation:

**Fig:**Heatmap of variables to show the null values

Imputing missing values is an important step in data preprocessing, and the choice of imputation method depends on the characteristics of the data and the nature of the missingness. Median imputation is one of the simplest and most commonly used methods for imputing missing values, particularly when the data contains outliers. The median is a robust measure of central tendency that is not affected by extreme values or outliers in the data. Therefore, median imputation can be a useful method when outliers are present in the data, as it provides a reasonable estimate of the missing value without being unduly influenced by extreme values.

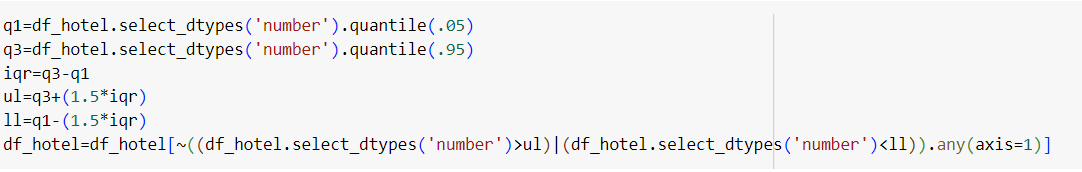
However,it is important to note that median imputation has some limitations. For example, it assumes that the data are approximately normally distributed, which may not always be the case. Additionally, median imputation can introduce bias in the estimates of other parameters, such as variances and covariances. Therefore, while median imputation can be a useful method for imputing missing values in the presence of outliers, it is important to carefully consider the assumptions and limitations of the method and to explore alternative methods if necessary.

## Checking for outliers:

## 

**Fig:**Boxplot of numerical variables

The box plots above are plotted for some numerical columns.



Nearly half of our dataset is having outliers. So we have taken Q1 as 0.05 and Q3 as 0.95 and treated the outliers.

# DROPPING OF COLUMNS

# 

We dropped some of the columns whose standard deviation value is 0.

# STATISTICALANALYSIS

### Chi-square test of independence:

Null hypothesis: There is no relationship between the two categorical variables.

Alternative hypothesis: There is a relationship between the two categorical variables.

Here , we passed all the categorical variables with target variable inside for loop. To check whether there is relationship between all categorical variables and target variable.

Here, we found out that all the categorical variables are significant.



**Fig:**Chi2 contingencyTest

### Mann-WhitneyTest

To check the significance of all numeric variables with target variable, we use the parametric test ttest\_ind or Wilcoxon test.The Wilcoxon signed-rank test, sometimes simply referred to as the Wilcoxon test, is a non-parametric statistical test used to compare two related or matched samples. It's an alternative to the t-test when the assumptions of normality are not met. Here we conclude that all the numeric variables are significant.

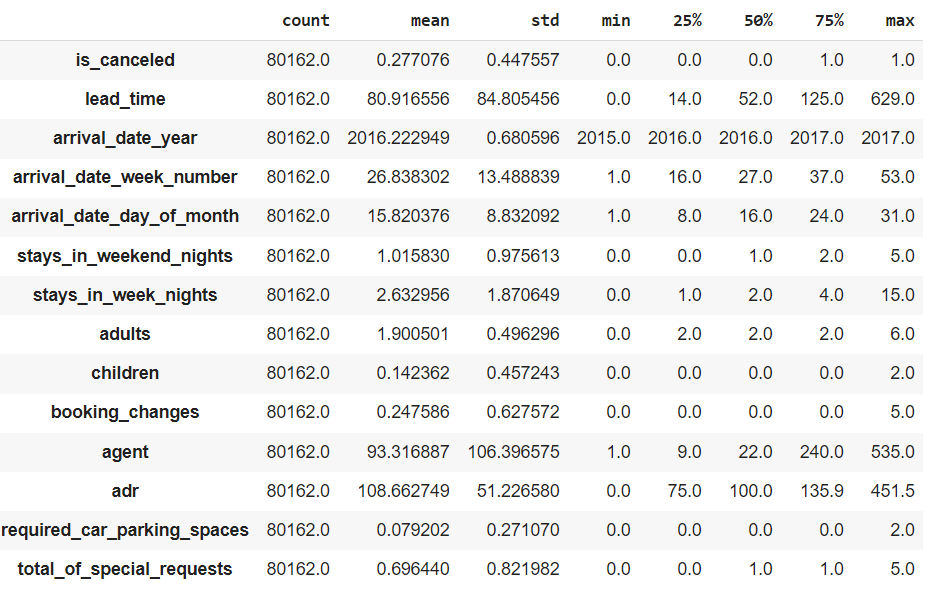


**Fig:**Mann-Whitneytest

**UNIVARIATE, BIVARIATE & MULTIVARIATE ANALYSIS**

### Univariate Analysis

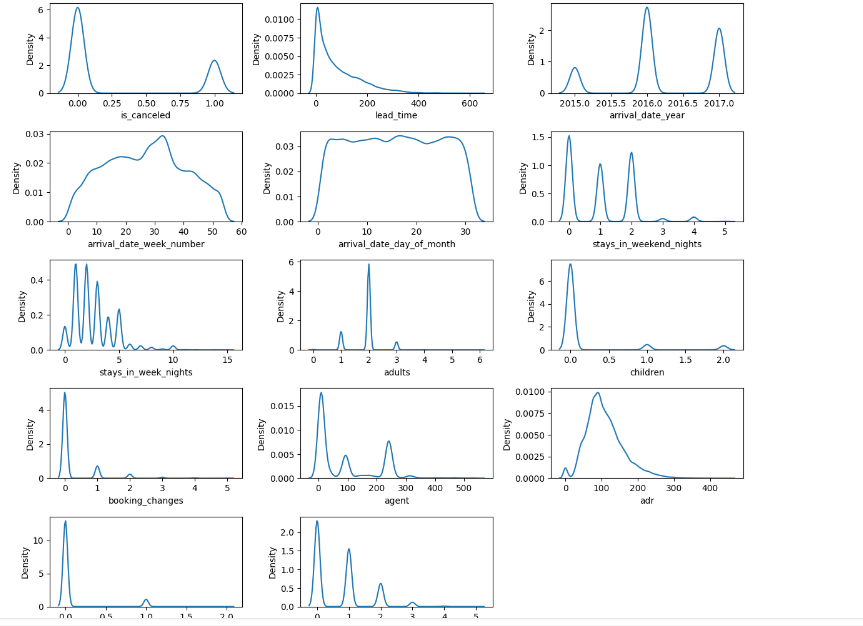
**Summary Statistics of Numerical Variable:**

****

**Fig:**5 point summary

There is a large difference between mean and median values , indicating that the variables are highly skewed. There is a large difference between the maximum and mean values, it typically indicates that there are a few extremely large values in the dataset that are significantly driving up the maximum value.

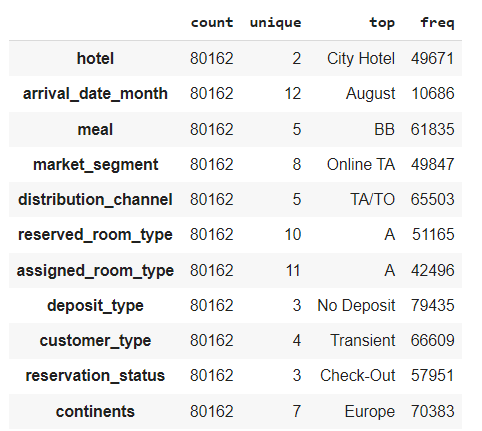
### KDE PLOT

****

**Fig:** KDE Plot of numerical variables

The arrival\_date\_day\_of\_month is moderately skewed .The variable adults and arrival\_date\_year are negatively skewed. Rest of the variables are highly positively skewed.

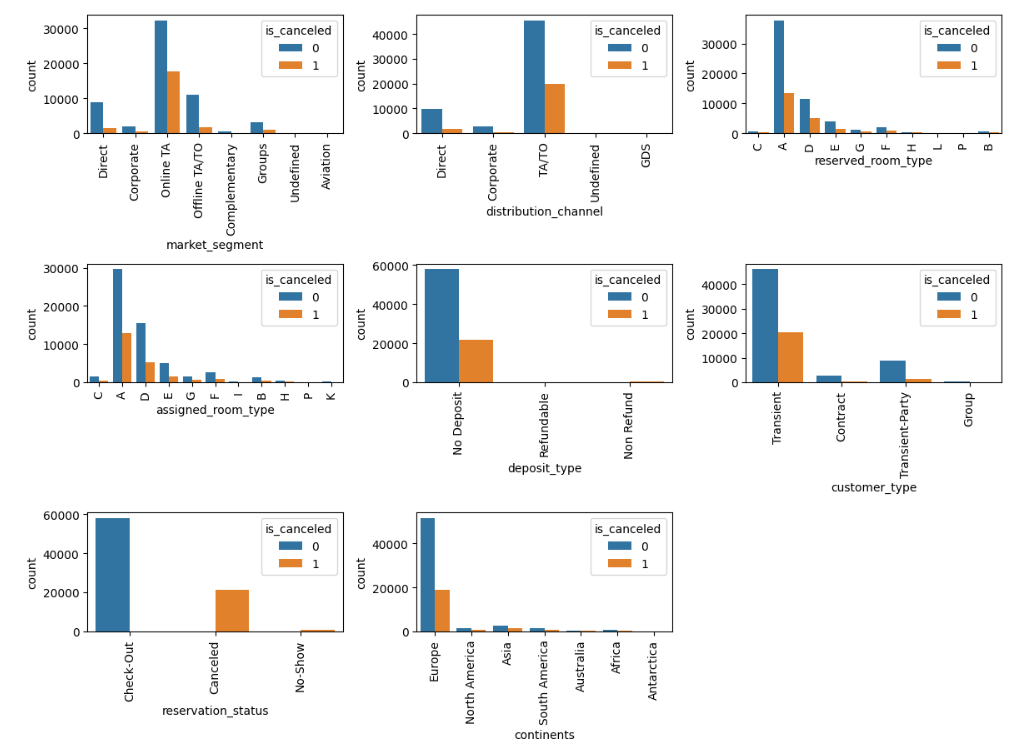
### Summary Statistics of Categorical Variable:

****

**Fig:** Summary Statistics of Categorical variables

There are eleven categorical variables: "hotel", "arrival\_date\_month", "meal", "market\_segment","distribution\_channel", "reserved\_room\_type",”deposit\_type”,”customer\_type”,”reservation\_status”,”continents” and "assigned\_room\_type. We can infer that City Hotel count is above 60000.Online Travel Agencies is the most preferred. Travel Agents and Tour Operators are widely utilized distribution channels. Most of the hotels are in Europe. Transient customer type is more. In the month of August, more customers came. The room type 'A' is assigned to more customers. Bed and Breakfast is taken by majority of the people.

**BIVARIATE ANALYSIS**

****

Based on the countplot of reservation\_status with is\_canceled, we can drop the column reservation\_status because it resembles the target column which we can see in the plot.

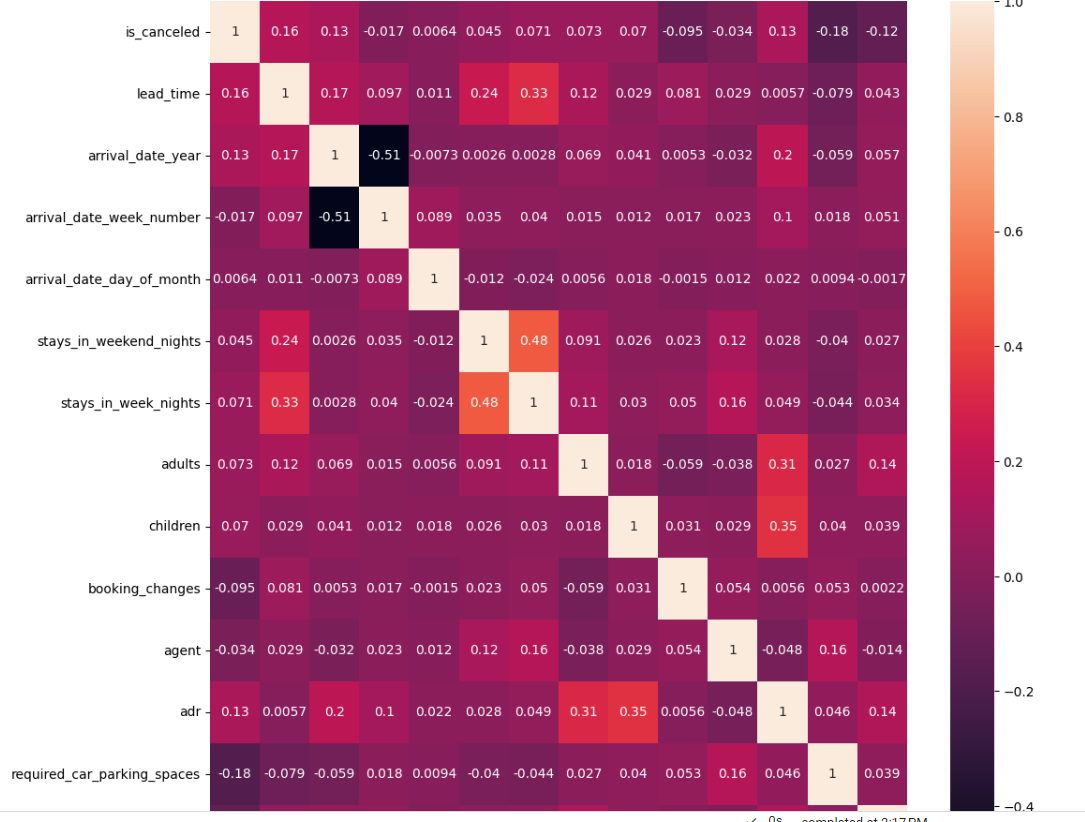
### BOXPLOT

### 

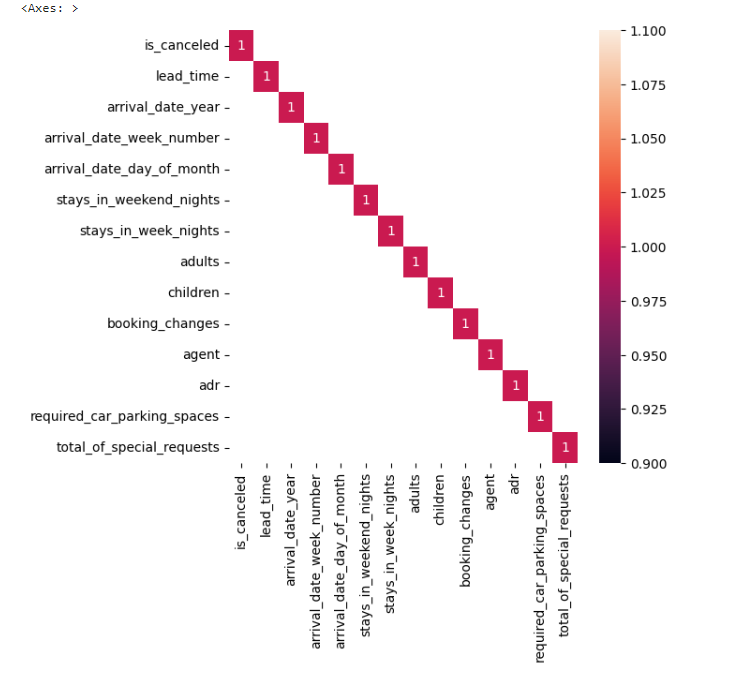
**Fig:** Boxplot of quantitative-target (is\_canceled)

The box plot is plotted between all numeric variables and target variable ‘is\_cancelled’. If the median of the categories of categorical variable is same, then we can conclude that the categorical variable is insignificant.

**HeatMap**

****

**Fig:** Heatmap (Numerical vs Numerical)



**Fig:** Heatmap (Correlation>0.65)

A heat map is a type of data visualization that uses color-coded squares or rectangles to represent the values of a two-dimensional dataset. It is particularly useful for visualizing large datasets and for identifying patterns or trends in the data. In a heat map, each row and column of the dataset is represented by a separate square or rectangle, and the color of the square represents the value of the corresponding cell in the dataset. The color scale typically ranges from a low value (e.g., blue or green) to a high value (e.g., red or yellow), with intermediate values represented by shades of the intermediate colors. They can also be used to visualize patterns or relationships in the data ,such as correlations between variables or clusters of similar observations.

From the correlation matrix, we can easily interpret that none of the features are correlated with each other and there’s absence of multicollinearity.

# ENCODING:

Encoding refers to the process of converting data from one representation to another. In the context of machine learning, encoding is often used to convert categorical variables(e.g.,colors,categories,labels) into a numerical representation that can be used by algorithms. This is important because many machine learning algorithms require numerical inputs.

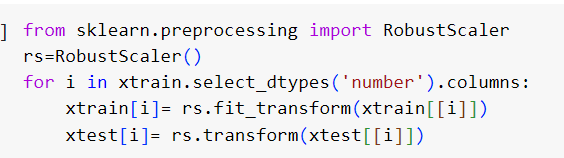
There are several methods of encoding categorical variables, including one-hot encoding, label encoding, and ordinal encoding. One-hot encoding involves creating a binary column for each category, where the column is 1 if the category is present and 0 otherwise. Label encoding involves assigning each category a numerical label (e.g.,0,1,2,etc.),which can be problematic if the labels have an order or hierarchy to them. Ordinal encoding is similar to label encoding, but itpreserves the order of the categories by assigning each category a numerical value based on its position in a predetermined order.

# SCALING&TRANSFORMATION

Scaling and transformation are two common data preprocessing techniques used to prepare data for analysis or modelling. Scaling involves transforming the values of the features in a dataset to a common scale, typically by subtracting the mean and dividing by the standard deviation. This is done to ensure that all features have the same scale and range, which can be important for certain algorithms, such as those that are distance-based. Transformation involves applying a mathematical function to the values of the features in a dataset to transform them in a specific way.

Robust scaling is a data normalization technique used in machine learning to scale numerical data so that it has a mean of 0 and a standard deviation of 1, while being robust to the presence of outliers. It is similar to standard scaling, but instead of using the mean and standard deviation of the data,it uses the median and interquartile range(IQR).

Yeo-Johnson Power Transformation is a data transformation technique used to normalize numerical data that may not follow a normal distribution. It is an extension of the Box-Cox transformation that can handle both positive and negative values. The Yeo-Johnson transformation applies a power transformation to the data using a lambda parameter that is chosen to maximize the normality of the transformed data.



We have done robust scaler here.

# CHECKING FOR CLASS IMBALANCE

In this case, the '0' class represents about 63% of the samples, while the '1' class represents only 37%of the samples. There is no class imbalance present here.

# MODEL BUILDING

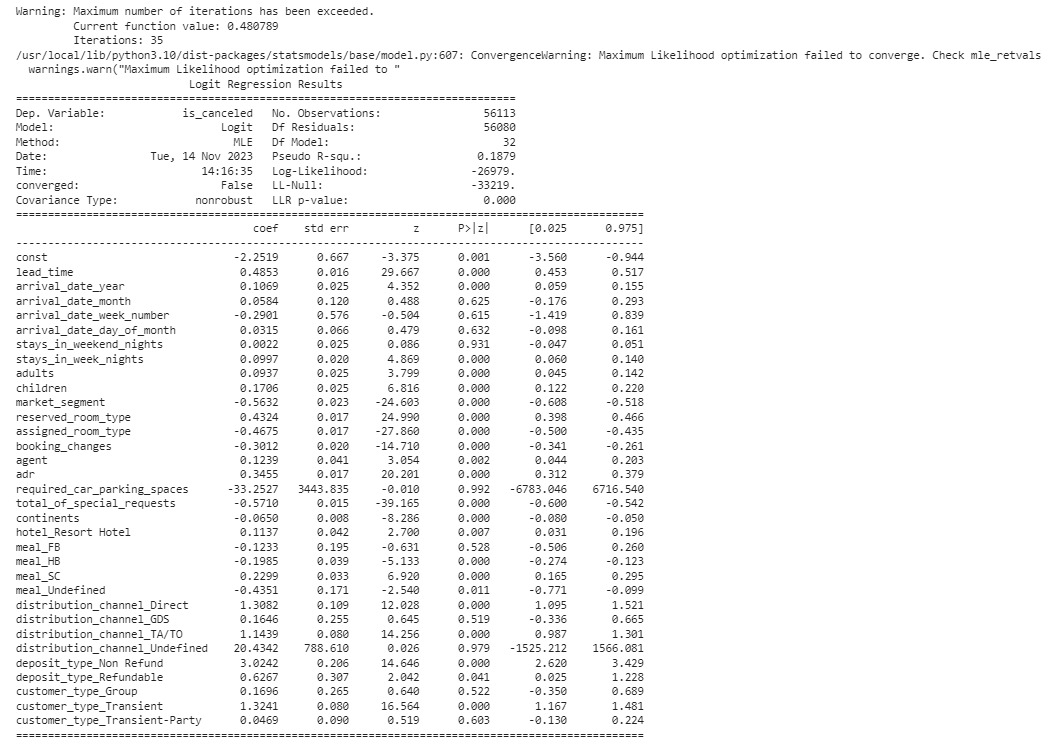
# Step by step approach for model building:-

1. Data Preprocessing: Clean and preprocess the data by handling missing values, encoding categorical features, scaling numerical features, and removing outliers.
2. Data Splitting: Split the data into training, validation, and testing sets. The training set is used to train the model, the validation set is used to tune the hyperparameters of the model, and the testing set is used to evaluate the final performance of the model.
3. Feature Engineering: Create new features from existing ones that can improve the model's predictive power. This can involve transforming or combining features, extracting new features from text or images, or engineering domain-specific features.
4. Model Selection: Select an appropriate machine learning model for the problem at hand, based on the type of data, the size of the dataset, and the performance metrics. Consider using simple models first, and gradually increasing complexity if necessary.
5. Hyperparameter Tuning: Tune the hyperparameters of the chosen model to optimize its performance on the validation set. This can involve selecting the best learning rate, regularization parameter, activation function, number of hidden layers, and other parameters.
6. Model Training: Train the final model on the entire training set using the optimized hyperparameters.
7. Model Evaluation: Evaluate the final model on the testing set using appropriate performance metrics such as accuracy, precision, recall, F1-score, or AUC-ROC curve. Compare the performance of the model with the baseline and previous state-of-the-art methods.
8. Model Deployment: Deploy the final model in a production environment, such as a web application, mobile app, or API, and monitor its performance over time. Update the model as necessary to improve its accuracy or adapt to changing data distributions.

Overall, this step-by-step approach provides a structured and systematic way to build, evaluate, and deploy machine learning models, and can help ensure the quality and robustness of the models.

**BASE MODEL (Logistic Regression Model)**

### LOGISTIC REGRESSION SUMMARY:

****

Based on the p-values, several of the predictor variables appear to be significantly associated with the probability of an items, including lead\_time, arrival\_date\_year, stays\_in\_week\_nights, adults, children, market\_segment, reserved\_room\_type, assigned\_room\_type, booking\_changes, agent, adr, total\_of\_special\_requests, continents, hotel\_Resort Hotel, meal\_HB, meal\_SC, meal\_Undefined, distribution\_channel\_Direct, distribution\_channel\_TA/TO, deposit\_type\_Non Refund, deposit\_type\_Refundable, customer\_type\_Transient.

The coefficients for arrival\_date\_month, arrival\_date\_week\_number, arrival\_date\_day\_of\_month, stays\_in\_weekend\_nights, required\_car\_parking\_spaces, meal\_FB, distribution\_channel\_GDS, distribution\_channel\_Undefined, customer\_type\_Group, customer\_type\_Transient-Party do not appear to be significant as their p-values are greater than 0.05. However, it's important to note that the model failed to converge and further analysis may be needed to determine the robustness of the results.

### TEST REPORT FOR BASE MODEL:

**A screenshot of a number

Description automatically generated**

**Fig:** Test Report (Base Model)

**Class 0 (Negative Instances):**

Precision (0.79): 79% of instances predicted as 0 were actually negative.

Recall (0.93): The model correctly identified 93% of all actual negative instances.

F1-score (0.86): The harmonic mean of precision and recall for negative instances is 86%.

Support (17493): There are 17493 instances of negative class in the dataset.

**Class 1 (Positive Instances):**

Precision (0.66): 66% of instances predicted as 1 were actually positive.

Recall (0.34): The model correctly identified 34% of all actual positive instances.

F1-score (0.45): The harmonic mean of precision and recall for positive instances is 45%.

Support (6556): There are 6556 instances of positive class in the dataset.

**Overall Model Performance:**

Accuracy (0.77): The accuracy of the model is 77%, meaning 77% of instances were correctly predicted.

**Weighted Average:**

Precision (0.75): The weighted precision, accounting for class imbalance, is 75%.

Recall (0.77): The weighted recall, accounting for class imbalance, is 77%.

F1-score (0.75): The weighted F1-score, accounting for class imbalance, is 75%.

**Analysis:**

The model performs well in accurately predicting negative instances (Class 0) with high precision and recall. However, there is room for improvement in predicting positive instances (Class 1), as reflected in lower precision and recall. The overall accuracy is 77%, and attention should be given to enhancing the model's ability to correctly identify positive instances.

### CONFUSION MATRIX

**A chart of a number of colored squares

Description automatically generated with medium confidence**

**Fig:** Confusion Matrix

* True Positives (TP): 2244 instances were correctly predicted as positive.
* True Negatives (TN): 16317 instances were correctly predicted as negative.
* False Positives (FP): 1176 instances were predicted as positive but were actually negative.
* False Negatives (FN): 4312 instances were predicted as negative but were actually positive.

This indicates that the model has relatively high accuracy in predicting negative instances (Class 0) but struggles with false negatives in predicting positive instances (Class 1).

### PERFORMANCE METRICS:

**A close up of numbers

Description automatically generated**

**Fig:** Performance Metrics (Base Model)

The model's accuracy is 0.771, which indicates that the model correctly classified 77.1% of the instances in the dataset. The precision of the model is 0.654, which means that out of all instances predicted as positive, 65.4% were actually positive. There call of the model is 0.342, which means that out of all actual positive instances, the model correctly identified 34.2%. The F1-score of the model is 0.449, which is the harmonic mean of precision and recall, and provides a balance between these two metrics.

The kappa value of 0.323 suggests that there is less agreement between the predictions made by the model and the actual values. This metric is useful in assessing inter-rater agreement, and values close to 1 indicate strong agreement, while values close to 0 indicate chance agreement. AUC-ROC is a metric that measures the area under the receiver operating characteristic (ROC) curve. The ROC curve is a plot of true positive rate (recall) versus false positive rate (1 - specificity) at various thresholds. In this case, the AUC-ROC is 0.788, indicating that the model performs moderately well at distinguishing between positive and negative instances.

Overall, the model appears to have moderate performance in terms of accuracy, precision, recall, F1-score, kappa value. There may be room for improvement in the model's performance, especially in identifying positive instances.

### 

### ROC CURVE

### A screen shot of a graph Description automatically generated

**Fig:**ROC CURVE

The AUC of logistic regression model on the test data is approximately 0.788. AUC-ROC is a metric that measures the area under the receiver operating characteristic (ROC) curve. The ROC curve is a plot of true positive rate (recall) versus false positive rate (1 -specificity) at various thresholds. In this case, the AUC-ROC is 0.78843 ,indicating that the model performs moderately well at distinguishing between positive and negative instances.

## Cross-Validation

A screenshot of a computer code

Description automatically generated

Fig (Cross- Validation)

The cross-validation results for a logistic regression model indicate an average accuracy of approximately 77.2% with a standard deviation of approximately 0.0045. This suggests that the model performs consistently across different folds, and the low standard deviation implies stability in the model's performance.

The average accuracy of 77.2% indicates that, on average, the model correctly predicts the target variable for 77.2% of the instances in the training data across the five folds. This provides a reasonable level of predictive performance.

The small standard deviation of 0.0045 suggests that the model's performance is robust and not highly sensitive to the specific subset of data used in each fold. A low standard deviation is desirable as it indicates less variability in the model's performance, contributing to its reliability.

In summary, the logistic regression model exhibits consistent and stable performance with an average accuracy of 77.2% across five-fold cross-validation. Further model evaluation and tuning can be performed to optimize its predictive capabilities.

## Random Forest Classifier:

**A screenshot of a computer

Description automatically generated**

**Fig:**Test Report (Random Forest Classifier)

The Random Forest classifier demonstrates strong predictive performance on the test set with an accuracy of 82%. The confusion matrix provides detailed insights into the model's classifications:

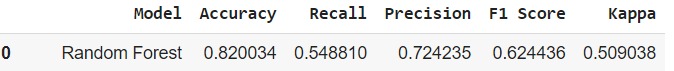
* True Negatives (16123): Instances correctly predicted as negative.
* False Positives (1370): Instances predicted as positive but are actually negative.
* False Negatives (2958): Instances predicted as negative but are actually positive.
* True Positives (3598): Instances correctly predicted as positive.

The precision-recall trade-off is evident in the classification report. Class 0 exhibits high precision (84%) and recall (92%), indicating effective identification of negative instances. However, Class 1 has lower precision (72%) and recall (55%), highlighting challenges in predicting positive instances.

The weighted average F1-score is 81%, emphasizing a balanced performance considering class imbalance. The model's macro-average F1-score is 75%, indicating a consistent measure across both classes.

In conclusion, the Random Forest model performs well overall, particularly in predicting negative instances. However, there's room for improvement in predicting positive instances, and further optimization or exploration of different algorithms may enhance the model's overall effectiveness.

### PERFORMANCEMETRICS:



**Fig:** Performance Metrics(Random Forest)

The Random Forest model achieves an accuracy of 82%, implying that it correctly predicts the target variable for 82% of instances in the dataset. While this indicates good overall predictive performance, a more detailed examination reveals nuances in the model's ability to handle positive instances (Class 1).

The recall for Class 1 is 54.88%, signifying that the model successfully captures 54.88% of actual positive instances. This metric is crucial in scenarios where correctly identifying positive instances is of high importance, such as identifying potential risks or rare events.

Precision for Class 1 is 72.42%, indicating that 72.42% of instances predicted as positive are indeed positive. Precision is valuable when the cost of false positives is high, as it represents the model's ability to make accurate positive predictions.

The F1 score, a harmonic mean of precision and recall, is 62.44%. This provides a balanced measure, considering both false positives and false negatives. It emphasizes the need for a model that can effectively balance precision and recall, especially in scenarios with imbalanced classes.

The Kappa statistic of 0.509 indicates moderate agreement beyond chance. While not extremely high, it suggests that the model's performance is better than random chance.

In conclusion, while the Random Forest model demonstrates strong overall accuracy, there is room for improvement in its ability to accurately identify positive instances. Further optimization or exploration of alternative models may enhance the model's performance, especially in scenarios where the correct identification of positive cases is critical.

## Comparison of Performance of Different Models

## 

Based on the evaluation metrics, the XG Boost model has the highest accuracy (0.82), recall (0.563), precision (0.716), F1 score (0.630), kappa score (0.514) among all the models evaluated. This indicates that the XG Boost model is the best-performing model for the given task.

## XGBOOST CLASSIFIER:

## A screenshot of a computer code Description automatically generated

**Fig:** Test Report (XGBoost Classifier)

The XG Boosting Classifier achieves an accuracy of 82.08% on the test set. The confusion matrix provides a detailed breakdown of the model's predictions:

* True Negatives (16182): Instances correctly predicted as negative.
* False Positives (1311): Instances predicted as positive but are actually negative.
* False Negatives (3178): Instances predicted as negative but are actually positive.
* True Positives (3378): Instances correctly predicted as positive.

Precision for Class 0 (negative instances) is high at 84%, indicating a good ability to correctly identify negative cases. However, precision for Class 1 (positive instances) is 72%, and recall is 52%, suggesting challenges in accurately predicting positive instances. The model demonstrates an overall F1-score of 60%.

The weighted average F1-score is 80%, considering class imbalance, and the macro-average F1-score is 74%, indicating a balanced measure across both classes. The model's accuracy is comparable to the Random Forest model but exhibits a trade-off between precision and recall for positive instances. Further optimization or tuning may enhance its performance, especially in scenarios where correctly identifying positive instances is crucial.

### PERFORMANCE METRICS:



**Fig:**Performance Metrics(XGBoosting Classifier)

The XGBoosting Classifier achieves an accuracy of 82%, indicating its ability to correctly predict the target variable for a significant portion of instances in the dataset. However, a more nuanced understanding of its performance is revealed through additional metrics:

* Recall (Sensitivity): The model exhibits a recall of 56.3% for positive instances (Class 1), implying that it successfully identifies 56.3% of all actual positive instances. A higher recall is desirable in scenarios where the cost of missing positive instances is high.
* Precision: The precision for Class 1 is 71.6%, indicating that 71.6% of instances predicted as positive are indeed positive. Precision is crucial in scenarios where the cost of false positives is significant.
* F1 Score: The F1 score, which considers both precision and recall, is 63%. This metric provides a balanced measure of the model's overall performance.

Kappa Statistic: The Kappa statistic of 51.4% suggests moderate agreement beyond chance. This metric accounts for the possibility of correct predictions occurring by random chance.

In summary, the XGB demonstrates solid overall accuracy, but there is room for improvement in correctly identifying positive instances. Further optimization or tuning may enhance its precision and recall balance, especially in scenarios where correctly identifying positive cases is crucial.

## Comparison of Performance of Different Models

**A screenshot of a graph

Description automatically generated**

**Fig:**Overall Comparison of Performance of all Models

**1. XGB Classifier:**

* Achieves the highest accuracy of 82.02%, demonstrating robust predictive performance.
* High recall (56.33%) for positive instances, indicating effective identification of actual positives.
* Good precision (71.65%) and balanced F1 score (63.07%) reflect strong overall model performance.
* Kappa statistic of 51.42% indicates substantial agreement beyond chance.

**2. Random Forest:**

* Like XGB, with an accuracy of 82.00%, precision, recall, and F1 score also comparable.
* Achieves a balanced trade-off between precision and recall for positive instances.
* Kappa statistic of 50.90% suggests substantial agreement beyond chance.

**3. Gradient Boosting Classifier:**

* Slightly lower accuracy at 81.33%, with balanced precision and recall.
* Demonstrates good precision (72.04%) but has a lower recall (51.53%) for positive instances.
* F1 score of 60.08% indicates a balanced measure of precision and recall.
* Kappa statistic of 48.33% suggests moderate agreement beyond chance.

**4. AdaBoost-rf:**

* Lower accuracy at 77.88%, with moderate recall (44.52%) for positive instances.
* Precision of 63.44% and F1 score of 52.33% indicate a trade-off between precision and recall.
* Kappa statistic of 38.50% suggests moderate agreement beyond chance.

**5. LogisticReg-skl:**

* Accuracy at 77.18%, with low recall (34.23%) for positive instances.
* Precision of 65.61% and F1 score of 44.99% indicate challenges in predicting positive instances.
* Kappa statistic of 32.34% suggests fair agreement beyond chance.

**6. KNearestNeighbour:**

* Accuracy at 76.93%, with a balanced trade-off between precision (58.48%) and recall (53.07%).
* F1 score of 55.64% indicates a balanced measure of precision and recall.
* Kappa statistic of 40.11% suggests moderate agreement beyond chance.

**7. Decision Tree-Gini:**

* Achieves 75.05% accuracy, with a good recall (56.07%) for positive instances.
* Precision of 54.08% and F1 score of 55.06% indicate a balanced measure of precision and recall.
* Kappa statistic of 37.80% suggests moderate agreement beyond chance.

**8. Gaussian NB:**

* Lowest accuracy at 49.62%, with high recall (88.99%) but low precision (33.86%) for positive instances.
* F1 score of 49.06% indicates an imbalance between precision and recall.
* Kappa statistic of 15.80% suggests poor agreement beyond chance.

**Overall:**

* XGB Classifier and Random Forest stand out with the highest accuracy and balanced precision-recall trade-off.
* Models vary in their performance, emphasizing the importance of choosing the right model for specific goals and datasets.
* Precision and recall metrics offer insights into the models' ability to handle positive instances, crucial in various real-world applications.

## HYPERPARAMETERTUNING

### XGBOOST Using GridSearchCV

**A close up of a text

Description automatically generated**

**Fig:**XGBoost using GridSearch CV

After hyperparameter tuning, the XGB Classifier demonstrates notable improvements in its performance:

* **Accuracy:** The model achieves an accuracy of 82.31%, showcasing enhanced predictive capability compared to the untuned version.
* **Recall:** The recall for positive instances (Class 1) increases to 57.06%, indicating improved sensitivity in identifying actual positive cases.
* **Precision:** Precision for Class 1 improves to 72.21%, suggesting a better ability to make accurate positive predictions.
* **F1 Score:** The F1 score reaches 63.58%, reflecting a more balanced measure of precision and recall. This improvement indicates a strengthened overall performance.
* **Kappa Statistic:** The Kappa statistic increases to 52.57%, signifying substantial agreement beyond chance. This suggests that the tuned XGB Classifier performs significantly better than random chance.

In summary, the hyperparameter tuning process enhances the XGB Classifier's accuracy, recall, precision, and overall F1 score. These improvements emphasize the effectiveness of selecting optimal hyperparameters to fine-tune the model for better performance on the given dataset.

## Cross-Validation

A screenshot of a computer code

Description automatically generated

Fig (Cross- Validation)

The cross-validation results for Extreme Gradient Boosting Classifier model indicate an average accuracy of approximately 81.9% with a standard deviation of approximately 0.0039. This suggests that the model performs consistently across different folds, and the low standard deviation implies stability in the model's performance.

The average accuracy of 81.9% indicates that, on average, the model correctly predicts the target variable for 81.9% of the instances in the training data across the five folds. This provides a reasonable level of predictive performance.

The small standard deviation of 0.0039 suggests that the model's performance is robust and not highly sensitive to the specific subset of data used in each fold. A low standard deviation is desirable as it indicates less variability in the model's performance, contributing to its reliability.

In summary, the logistic regression model exhibits consistent and stable performance with an good accuracy of 81.% across five-fold cross-validation. Further model evaluation and tuning can be performed to optimize its predictive capabilities.

## Feature Importance

The most important feature for the model is "deposit\_type\_non\_refund", which has an importance score of 0.38. The next most important features are "market\_segment\_online\_TA" and "total\_of\_special\_requests" with importance scores of 0.11 and 0.07, respectively.

It's important to note that feature importance scores can vary depending on the specific model anddataset used. Therefore, it's always good practice to evaluate the feature importance of a model to better understand which features are driving its predictions.

A screen shot of a graph

Description automatically generated

**Fig:** FeatureImportance

## CONCLUSION

By accurately predicting which bookings are likely to be canceled, hotels can optimize their room allocation, adjust pricing strategies, and minimize the impact of cancellations on revenue. This approach allows hotels to strike a balance between short-term gains and long-term stability, overcoming the challenges posed by initiatives like Booking Genius.

Random Forest and XGBoost exhibit relatively better overall performance compared to other models in terms of accuracy, AUC, recall, precision, and F1-score. These models are promising candidates for further consideration. After fine tuning the hyperparameters in XGBoost, we got the best model as XGBoost Classifier. We got the deposit\_type\_non\_refund feature as the important feature to consider.

So we need to focus more on it.

Our hotel booking cancellation prediction project has successfully developed a predictive model with commendable performance metrics, including accuracy, precision, and recall. The identified key predictors shed light on significant factors influencing booking cancellations, offering valuable insights for hotel management. These findings can inform strategic decisions related to resource optimization, targeted marketing, and overall guest satisfaction.

**REFERENCES**

1. https://www.kaggle.com/jessemostipak/hotel-booking-demand
2. <https://ieeexplore.ieee.org/document/10128484>
3. <https://www.researchgate.net/publication/361131872_Hotel_Booking_Prediction_using_Machine_Learning>
4. [https://towardsdatascience.com/predicting-hotel-cancellations-with-](https://towardsdatascience.com/recommendation-system-in-python-lightfm-61c85010ce17)machine-learning-fa669f93e794
5. https://triptease.com/blog/the-real-cost-of-free-cancellations/