Hotel Booking Prediction using Machine Learning Algorithms

Group Info.

Group - 1			
Name	**ID**	**Mail Ids**	**Contribution**
Ketha Tirumuru	11597873	kethatirumuru@my.unt.edu	Pre-processing, modeling, Analysis
Chandana Polakonda	11645295	chandanapolakonda@my.unt.edu	Pre-processing, modeling, Analysis
Arun Thotakuri	11600012	arunthotakuri@my.unt.edu	Pre-processing, modeling, Analysis
Numitha Devi Oguri	11647630	numithadevioguri@my.unt.edu	Pre-processing, modeling, Analysis
Lalitha Nali	11649335	lalithanali@my.unt.edu	Pre-processing, modeling, Analysis

Abstract:

In order to create precise predictive models for booking cancellations, this research focuses on evaluating hotel reservation data. Our goal is to discover the critical elements driving booking cancellations and develop trustworthy models that can forecast cancellation probabilities using machine learning algorithms. Both hotel management and guests will profit from the understandings obtained from this analysis, helping them to make wise hotel reservation selections. The best time to book, the optimum duration of stay to get the greatest daily rate, and the chance of getting special requests are just a few of the crucial concerns we try to address. Our objective is to maximize hotel occupancy rates and raise customer satisfaction by utilizing the power of predictive modeling.

We use a dataset with 119,390 observations that includes booking data for a resort hotel and a city hotel in order to achieve our goals. Each observation represents a hotel reservation and is linked to 31 variables that offer pertinent information about the reservation. These factors include the date of the reservation, the length of the stay, the number of adults, kids, and infants, the accessibility of parking spaces, and more. The cancellation status, which indicates whether a reservation was canceled or not, is the key variable for our research.

We preprocess the dataset by handling missing values and encoding categorical variables to ensure the accuracy and reliability of the data. Depending on the situation, appropriate approaches, such as imputation or deletion, are used to deal with missing values. The

successful use of categorical variables in our machine learning models is made possible by their encoding into numerical representations. We start the project design process with a preprocessed dataset and move through steps including data exploration, feature selection, model training, and evaluation. We seek to build reliable prediction models that correctly predict booking cancellations and offer insightful information about the variables behind these cancellations by utilizing cutting-edge technologies and frameworks.

Data Specification:

A total of 119,390 observations make up the hotel reservation data that was used in this project. Each observation represents a hotel reservation and is linked to a number of features that offer helpful information about the reservation. The objective of this research is to create supervised learning-based predictive models for booking cancellations.

hotel: The type of hotel is indicated by this categorical variable, which distinguishes between a city hotel and a resort hotel. It aids in capturing the many features and services connected to each type of lodging.
is_canceled: This binary variable indicates whether or not a reservation was canceled. A number of 1 denotes that the reservation was canceled, while a value of 0 denotes that it was not. Since the objective is to predict booking cancellations, this feature acts as the target variable for prediction.
lead_time: This numerical attribute indicates how many days there are between the date of the reservation and the day of arrival. It offers information about how much time the guest had between making the appointment and when they planned to arrive, which may affect the possibility that they would cancel.
arrival_date_month: The month of the arrival date is indicated by this categorical attribute. It makes it possible to analyze seasonality and possible fluctuations in booking cancellations throughout the course of the year.
arrival_date_week_number: The week number of the arrival date is shown by the numerical characteristic arrival_date_week_number. In order to investigate any weekly patterns in booking cancellations, it offers an additional temporal dimension.
arrival_date_day_of_month: The day of the month of the arrival is indicated by this number attribute. It makes it possible to look into any distinct trends or patterns connected to particular dates throughout the month.
stays_in_weekend_nights: This numerical attribute indicates how many Saturday or Sunday or other weekend nights the visitor stayed. It offers information about the visitor's preferences and length of stay on weekends.
stays_in_week_nights: This digit indicates how many weekday nights (Monday through Friday) the visitor spent there. By collecting the visitor's preferences and length of stay on weekdays, it enhances the prior feature.
adults, children, babies: Number of adults, kids, and babies in the reservation is indicated by the characters "adults," "kids," and "babies," accordingly. They reveal details on the make-up and size of the party or group making the reservation.
meal: This categorical characteristic denotes the kind of meal arrangement that the visitors have made, such as "Bed & Breakfast," "Half Board," "Full Board," or "No Meal." It records the

diners' preferences and may provide insight into how they felt about their entire booking experience.
market_segment: This categorized attribute identifies the market sector related to the reservation, such as "Online Travel Agents," "Direct," "Corporate," or "Groups." It offers information on the various target markets and distribution methods used to make reservations.
distribution_channel: This categorized attribute identifies the method by which the reservation was made, such as "Travel Agents," "Direct," or "Corporate." It aids in understanding the various platforms and channels used to make bookings.
is_repeated_guest: This binary feature lets users know whether a visitor has already visited before or not. A number of 1 denotes a returning guest, while a value of 0 denotes a first-time visitor. It records the guest's level of loyalty or repeat business.
reserved_room_type: This categorical characteristic identifies the kind of accommodation that the visitor has booked. It reveals the initial preferences and expectations of the guests for the accommodation.
assigned_room_type: The type of room that was given to the visitor at check-in is represented by the categorical characteristic called assigned_room_type. It highlights any potential adjustments or differences between the booked room type and the one that was actually given.
booking_changes: This numerical attribute represents the quantity of booking changes. It records any changes or additions made to the reservation prior to the visitor's arrival.
deposit_type:Using the category feature "deposit_type," you may specify whether a deposit was made for a reservation as "No Deposit," "Non-Refundable," or "Refundable." The payment and deposit preferences of the visitors are shown.
agent: This attribute displays the ID of the travel company or reservation agent that is linked to the reservation. It aids in locating the particular company or agent in charge of making the reservation.
company: This feature shows the ID of the organization or company making the reservation. It aids in locating the company or group responsible for the reservation.
days_in_waiting_list: This metric indicates how long a reservation was held on a waiting list before it was verified. It records any delays or waiting times connected to the reservation.
customer_type: This categorized attribute describes the kind of client, such as "Transient," "Contract," "Group," or "Transient-Party." It offers information on the many client segments connected to the bookings.
required_car_parking_spaces: This integer attribute indicates how many parking spaces the visitor needs. It represents the visitors' preferences and parking requirements.
total_of_special_requests: This numerical attribute represents all of the special requests that the visitor has submitted. It covers requests for extra beds, certain room demands, and any other specific needs.
reservation_status: This categorical attribute describes the current reservation status, such as "Canceled," "Check-Out," or "No-Show." It aids in tracking the reservation's development and outcome.
reservation_status_date: The date that the reservation status was most recently changed is represented by this feature. It offers details on the progression and background of the reservation's status changes.

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_segment	dist
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	0.0	0	BB	PRT	Direct	
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	0.0	0	BB	PRT	Direct	
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	0.0	0	BB	GBR	Direct	
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	0.0	0	BB	GBR	Corporate	
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	0.0	0	BB	GBR	Online TA	

Hotel Dataset

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeated_guest	previous_cancell
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119386.000000	119390.000000	119390.000000	119390.
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599	2.500302	1.856403	0.103890	0.007949	0.031912	0.
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613	1.908286	0.579261	0.398561	0.097436	0.175767	0.
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000	2.000000	0.000000	0.000000	0.000000	0.
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000	2.000000	0.000000	0.000000	0.000000	0.
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000	3.000000	2.000000	0.000000	0.000000	0.000000	0.
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	50.000000	55.000000	10.000000	10.000000	1.000000	26.

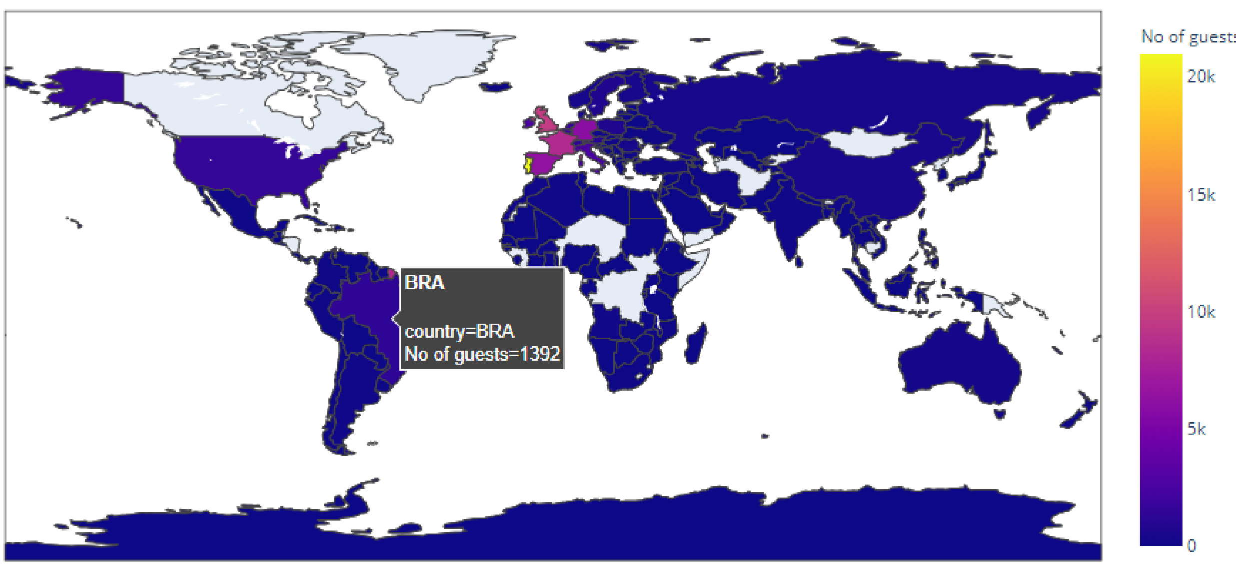
Hotel Dataset Description

```
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

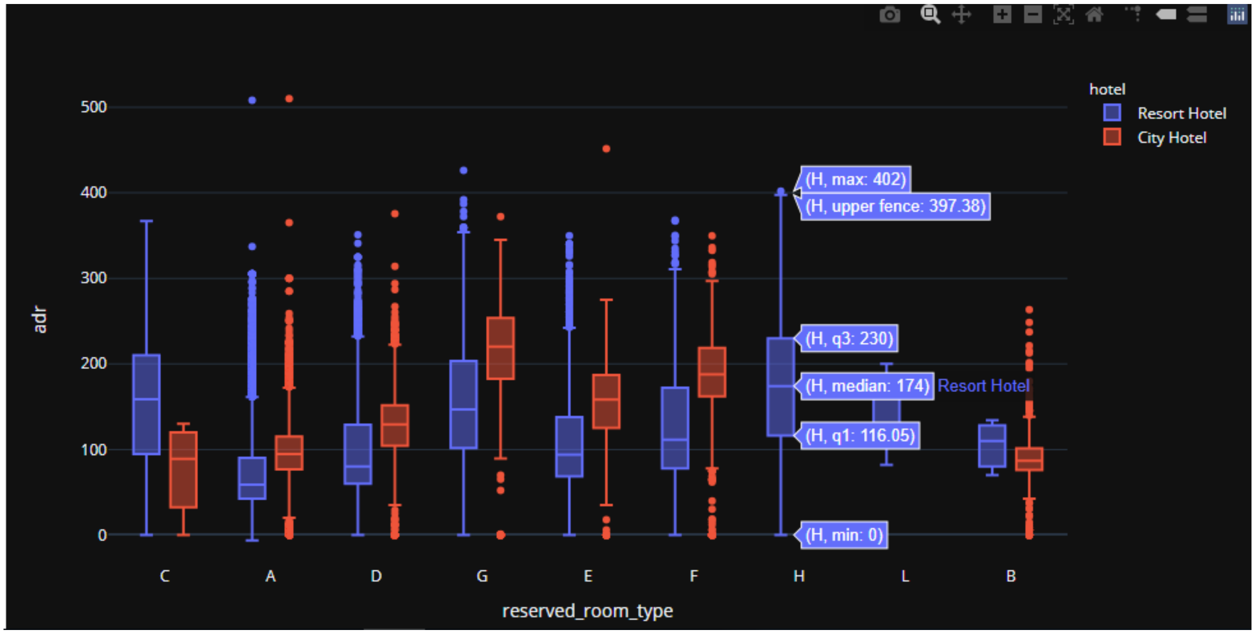
Hotel Dataset Info

	country	No of guests
0	PRT	20977
1	GBR	9668
2	FRA	8468
3	ESP	6383
4	DEU	6067
...	...	...
161	NPL	1
162	GUY	1
163	MRT	1
164	ATF	1
165	NAM	1

From where the guests are coming



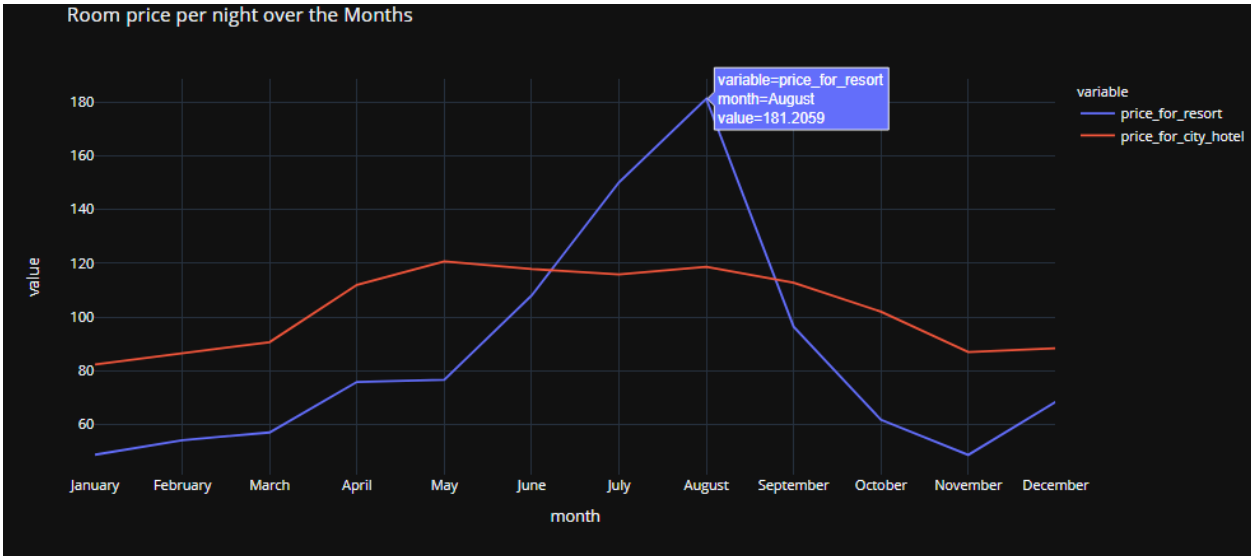
People from all over the world are staying in these two hotels. Most guests are from Portugal and other countries in Europe.



Both hotels have different room types and different meal arrangements. Seasonal factors are also important, So the prices varies a lot.
The figure shows that the average price per room depends on its type and the standard deviation.

	month	price_for_resort	price_for_city_hotel
0	April	75.867816	111.962267
1	August	181.205892	118.674598
2	December	68.410104	88.401855
3	February	54.147478	86.520062
4	January	48.761125	82.330983
5	July	150.122528	115.818019
6	June	107.974850	117.874360
7	March	57.056838	90.658533
8	May	76.657558	120.669827
9	November	48.706289	86.946592
10	October	61.775449	102.004672
11	September	96.416860	112.776582

Month wise price of both hotels



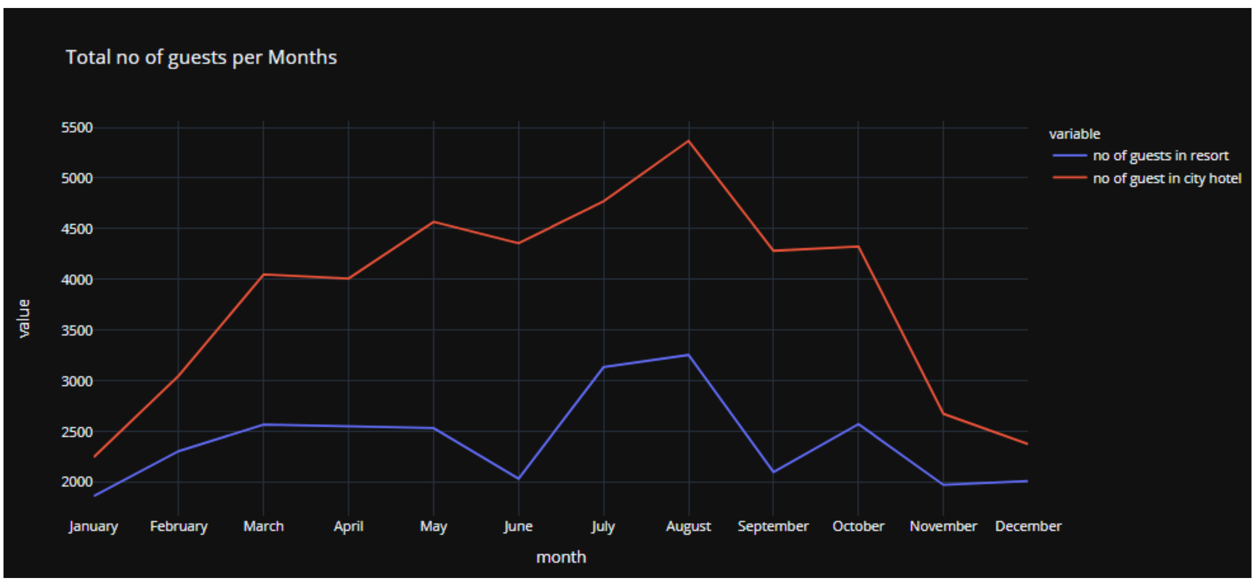
This plot clearly shows that prices in the Resort Hotel are much higher during the summer and prices of city hotel varies less and are most expensive during Spring and Autumn.

	month	no of guests
0	August	3257
1	July	3137
2	October	2575
3	March	2571
4	April	2550
5	May	2535
6	February	2308
7	September	2102
8	June	2037
9	December	2014
10	November	1975
11	January	1866

No.of guests per month

	month	no of guests in resort	no of guest in city hotel
0	August	3257	5367
1	July	3137	4770
2	October	2575	4326
3	March	2571	4049
4	April	2550	4010
5	May	2535	4568
6	February	2308	3051
7	September	2102	4283
8	June	2037	4358
9	December	2014	2377
10	November	1975	2676
11	January	1866	2249

Month wise guests per month



The City hotel has more guests during spring and autumn, when the prices are also highest. In July and August there are less visitors, although prices are lower. Guest numbers for the Resort hotel go down slightly from June to September, which is also when the prices are highest. Both hotels have the fewest guests during the winter.

Project Design:

In this Project, hotel reservation data were analyzed and prediction models for cancellations of reservations were created using a variety of tools, frameworks, and models. Python was utilized as the programming language, and a number of libraries, including Pandas, NumPy, scikit-learn, and Matplotlib, were also used.

One would need to have Python and the necessary libraries installed in order to replicate the results. It would also be important to have access to the project's dataset, which contains data on hotel reservations. The dataset would require preprocessing and transformation into an analysis-ready format.

Catoboost, Gradient boosting, decision trees, random forests, and logistic regression were among the models employed in the project. These models were chosen because they can handle both numerical and category information and are effective for binary classification tasks. These models were implemented using the scikit-learn library, which offered a reliable and effective framework for developing and testing the models.

The program's data preprocessing, feature selection, and model evaluation all included significant design choices. We carefully checked the dataset for anomalies, outliers, and discrepancies. To assure the dataset's quality, data cleaning procedures like imputation and outlier removal were used. The most pertinent elements for the prediction job were found using feature selection approaches like correlation analysis and feature importance.

The project's data preprocessing techniques (such as handling missing values and encoding categorical variables), feature selection techniques (such as correlation analysis and feature importance), model training and evaluation techniques, and prediction techniques were among the most crucial functions and methods. For better code organization and reusability, some functions were divided into distinct modules or classes and modularized.

Following a disciplined and modular approach, the coding was finished. The stages for data preprocessing were carried out first, then those for feature selection, model training, and model evaluation. To make sure that each step was accurate and correct, it was rigorously checked. To improve readability and maintainability, the code was commented on and given docstrings.

The user's experience of the program's functionality included importing the dataset, preparing the data (managing missing values, encoding categorical variables, etc.), choosing pertinent features, training and comparing several models, and forecasting cancellations of reservations. The application provides predictions based on the chosen model and insights into the key factors impacting cancellations. The user might evaluate the effectiveness of various models and choose the best one for their own requirements.

The feature selection procedure could serve as an illustration of complex logic. It involved employing ensemble models, such as random forests or gradient boosting, to calculate feature importance scores. Training these models and extracting the feature significance scores were required by the reasoning. The most important features for model training and prediction were then ranked and chosen based on the scores.

i. Preprocessing and data loading: A pandas DataFrame is used to import the hotel booking dataset, and missing values are handled accordingly. Variables that can be categorized are encoded for additional study.

ii. Data division into training and test sets: For the purpose of training the models and assessing their efficacy, the dataset is split into a training set and a testing set.
iii. Model for logistic regression: The training set serves as the basis for the Logistic Regression model's evaluation using the F1 score. This model has an F1 score of 0.86.
Iv. K-Nearest Neighbors (KNN) model: The F1 score is used to assess the KNN model, which is trained on the training set. This model has an F1 score of 0.92.
V. Model for a decision tree classifier: The F1 score is used to assess the Decision Tree Classifier model, which is trained on the training set. This model has an F1 score of 0.96.
Vi. Model of the Random Forest Classifier: The Random Forest Classifier model is tested using the F1 score after being trained on the training set. This model has an F1 score of 0.96.
Vii. Model XGBoost: Using the F1 score, the XGBoost model is assessed after being trained on the training set. This model has an F1 score of 0.99.
Viii. CatBoost model: The CatBoost model is tested using the F1 score after being trained on the training set. This model has an F1 score of 1.00.

Milestones:

1. Data Exploration and Pre-processing: The project's initial milestone involved exploring the hotel booking dataset and carrying out the appropriate data pretreatment procedures. This involved looking at the dataset's structure, spotting outliers and missing values, and managing them accordingly. Categorical variables were encoded for further analysis after the dataset had been cleaned.

2. Feature Engineering and Selection: Additional features were created from the existing dataset at this milestone to collect more pertinent data. One feature, for instance, included the total number of visitors (adults, kids, and newborns). Following feature engineering, feature selection methods were used to determine which features were crucial for the prediction task. To choose the ideal set, many techniques like correlation analysis, feature importance from ensemble models, and others were used.

3. Model Training and Evaluation: The following milestone entailed using the chosen features to train several machine learning models. The dataset was used to implement and train gradient boosting, decision trees, random forests, and logistic regression models. Utilizing suitable criteria including accuracy, precision, recall, and F1 score, the models were assessed. Techniques for cross-validation were used to achieve a reliable evaluation.

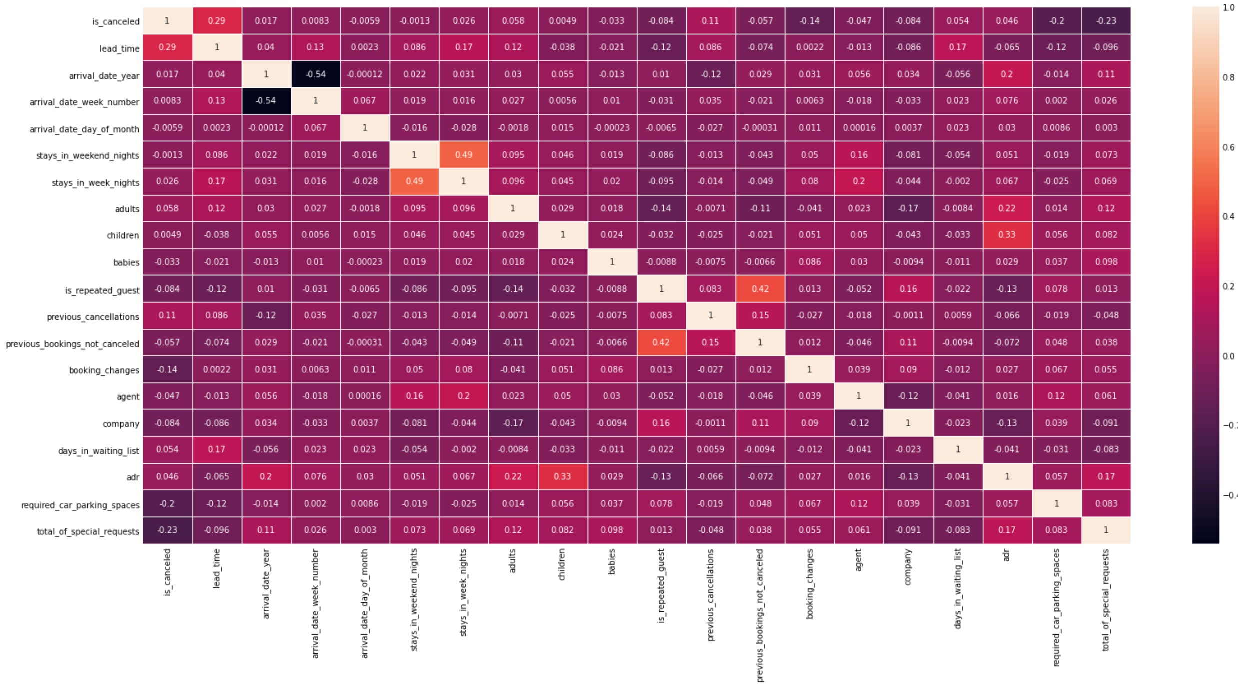
4. Model Performance Comparison and Selection: This milestone concentrated on evaluating the effectiveness of various models and choosing the best one for the task of forecasting cancellations of reservations. The model with the greatest F1 score (or any other desired statistic) was selected as the final model after the evaluation results were examined.

1. Data loading and preprocessing: Completed
  - Dataset loaded into pandas DataFrame
  - Missing values handled appropriately

- Categorical variables encoded

2. Splitting the data into training and testing sets: Completed

- Data divided into training set and testing set



3. Logistic Regression model: Completed

- Model trained on the training set
- Evaluation using F1 score: 0.86

```
Accuracy Score of Logistic Regression is : 0.8106422839247267
Confusion Matrix :
[[21339  1097]
 [ 5675  7652]]
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.95      0.86     22436
           1       0.87      0.57      0.69     13327

    accuracy                           0.81     35763
   macro avg       0.83      0.76      0.78     35763
weighted avg       0.82      0.81      0.80     35763
```

4. K-Nearest Neighbors (KNN) model: Completed

- Model trained on the training set

- Evaluation using F1 score: 0.92

```
Accuracy Score of KNN is : 0.8920951821715181
Confusion Matrix :
[[21692   744]
 [ 3115 10212]]
Classification Report :
              precision    recall  f1-score   support

           0       0.87      0.97      0.92     22436
           1       0.93      0.77      0.84     13327

    accuracy                           0.89     35763
   macro avg       0.90      0.87      0.88     35763
weighted avg       0.90      0.89      0.89     35763
```

5. Decision Tree Classifier model: Completed

- Model trained on the training set
- Evaluation using F1 score: 0.96

```
Accuracy Score of Decision Tree is : 0.9490534910382239
Confusion Matrix :
[[21578   858]
 [  964 12363]]
Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     22436
           1       0.94      0.93      0.93     13327

    accuracy                           0.95     35763
   macro avg       0.95      0.94      0.95     35763
weighted avg       0.95      0.95      0.95     35763
```

6. Random Forest Classifier model: Completed

- Model trained on the training set
- Evaluation using F1 score: 0.96

```
Accuracy Score of Random Forest is : 0.9531638844615944
Confusion Matrix :
[[22287   149]
 [ 1526 11801]]
Classification Report :
              precision    recall  f1-score   support

           0       0.94      0.99      0.96     22436
           1       0.99      0.89      0.93     13327

    accuracy                           0.95     35763
   macro avg       0.96      0.94      0.95     35763
weighted avg       0.96      0.95      0.95     35763
```

7. XGBoost model: Completed

- Model trained on the training set
- Evaluation using F1 score: 0.99

```
Accuracy Score of XG Boost Classifier is : 0.9840897016469535
Clear output

executed by Tirumuru Ketha
11:30 PM (5 minutes ago)          :
executed in 19.078s          on    recall  f1-score   support

           0       0.98      1.00      0.99     22612
           1       1.00      0.96      0.98     13151

    accuracy                           0.98     35763
   macro avg       0.99      0.98      0.98     35763
weighted avg       0.98      0.98      0.98     35763
```

8. CatBoost model: Completed

- Model trained on the training set
- Evaluation using F1 score: 1.00

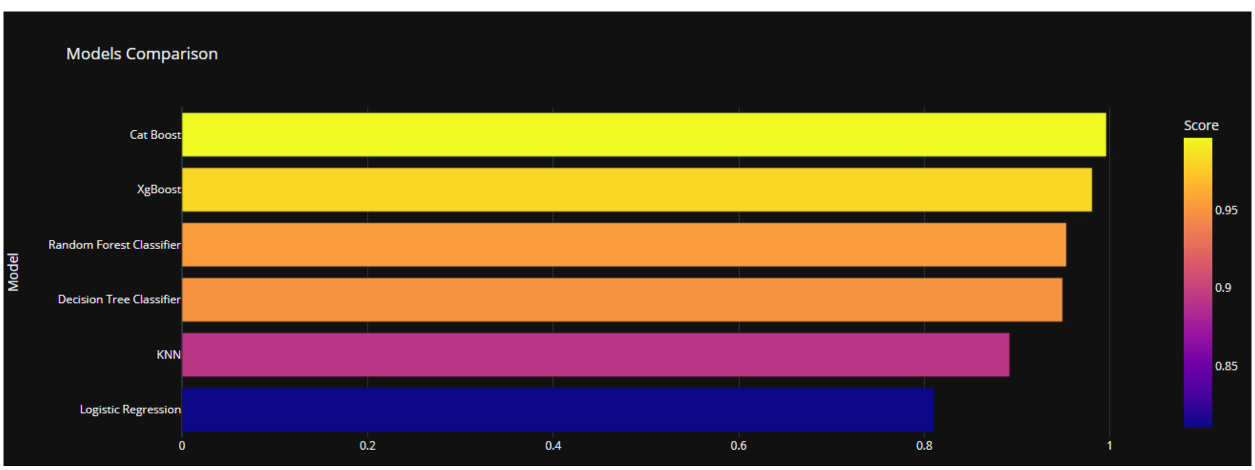
```
Accuracy Score of  CatBoost Classifier is : 0.9954142549562397
Confusion Matrix :
[[22602    10]
 [  154 12997]]
Classification Report :
              precision    recall  f1-score   support

           0       0.99      1.00      1.00     22612
           1       1.00      0.99      0.99     13151

    accuracy                           1.00     35763
   macro avg       1.00      0.99      1.00     35763
weighted avg       1.00      1.00      1.00     35763
```

9. Model comparison and Selection:

	Model	Score
5	Cat Boost	0.996253
4	XgBoost	0.980958
3	Random Forest Classifier	0.953164
2	Decision Tree Classifier	0.949053
1	KNN	0.892095
0	Logistic Regression	0.810642



Incremental Features:

1.Additional Features: New features were developed as part of the feature engineering process to capture significant facets of the booking data. For instance, existing features were used to calculate the total number of guests and the total number of special requests. These incremental characteristics enhanced the performance of prediction by giving the models more data to learn from.
2.Selecting Advanced Features: In the beginning, simple feature selection methods like correlation analysis were used. More sophisticated methods, like feature importance from ensemble models, were introduced into the project as it went along. The accuracy of the models was increased and the most important characteristics were identified using these incremental feature selection techniques.
3. Model ensembles: Later on in the research, ensemble models like gradient boosting and random forests were added. These models combine a number of weak learners to create a more accurate prediction. The project improved forecast accuracy and robustness by using model ensembles.

Repo Link:
https://github.com/tketha/SD-for-AI---Group1/tree/main

References:

1. https://www.kaggle.com/code/niteshyadav3103/hotel-booking-prediction-99-5-acc/input
2. https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand
3. https://www.sciencedirect.com/science/article/pii/S2352340918315191