Hotel Booking Cancellation Prediction

Group – 3

Sri Ram Jonnalagadda

Rohan Naga Venkata Mayukh Ungarala

Priya Khandelwal

Neetu Rasinger Babu

Raga Poojitha Bottlagunta

Table of Contents

1. Introduction

1.1 Project Background and Problem Definition

The hotel sector is among the top sectors. Daily cancellations of hotel reservations are common for a variety of reasons. Cancellations of bookings have a big impact on demand management in the hotel sector. Hotel cancellations can be done via their hotel website, app, or by giving them a call. Hotels use strict cancellation policies and overbooking techniques to lessen the impact of cancellations, however these strategies can have a detrimental effect on income and the hotel's reputation. A machine learning-based system prototype was created to lessen this impact. Machine learning models can learn about the new patterns and predict hotel booking cancellations. Classification models are trained using data from the hotel's property management system on a day-to-day basis to determine which reservations are "likely to cancel" and as a result, to determine net demand.

Due to overbooking, the hotel is forced to refuse service, which can be awful for the customer and hurt the hotel's reputation as well as its immediate financial situation. It may also indicate potential revenue loss from unhappy clients who refuse to make another reservation at the same hotel. However, strict cancellation policies, especially those that are non-refundable, run the risk of not only reducing bookings but also decreasing revenue because of the imposition of significant discounts.

Revenue managers and other hotel staff members can take steps to prevent potential cancellations by identifying which reservations are likely to be canceled and offering services such as room upgrades, discounts, or other perks. However, some customers (such as corporate customers) are not price sensitive and these offers may not always apply. However, booking classification is not the only potential benefit. Running the model daily for all bookings of a book yields significant results. Number of nights expected to be canceled per next day. Using this amount, the hotel can derive value from demand by calculating net demand. Accurate demand figures help hotel managers create more effective overbooking and cancellation policies.

The cancellation prediction system described in this project uses data from a hotel property management system (PMS) to identify hotel reservations that are likely to be canceled. It is based on a machine learning model. This study demonstrates how a machine learning cancellation prediction model can be a great tool, but also shows how the system was tested at two hotels in a real production environment.

Once a machine learning model has been developed, it is important that the model is able to analyze all cancelled rooms and the customer details in a timely manner. The model will be required to be to be retrained constantly to ensure that it never results in a concept drift.

1.2 Project Objectives

The main objective of this project is to create machine learning model which predicts about whether a hotel reservation made in the future can be canceled or not. The below mentioned are the other objectives of this project.

- Understanding Hotel Booking system and predicting the cancellations
- Collecting data on hotel booking cancellations
- Performing Exploratory data analysis to understand the dataset
- Data Pre-processing for training Machine Learning models
- Implementing Various Machine models and comparing them
- Fine tuning the best model

- Deployment of the model.

1.3 Project Requirements

Processing each cancellation and classifying it as a cancelled booking are functional requirements. A cancelled flag has already been added to the current dataset. We must create a machine learning algorithm that learns about canceled transactions by using this data as training. Once trained, the model should be able to anticipate canceled reservations in real-time hotel management transactions. Real-time events must be accepted by the system, and a new flag called "is cancelled" must be created.

Machine learning requirements include identifying suitable models for hotel booking cancellations. We will check the different machine learning models and, in that model, we will select the best model and it is hyper parameter tuned and used for the prediction. Optuna was used in our project.

1.4 Project Deliverables

- The processing, preparation, and creation of training data related to cancellations.
- List of chosen machine learning models and justification of selecting them.
- Train, test ML models, and preserve performance results and models.
- Detect the hotel cancellations and let the hotels know about them.
- Retrain the machine learning models every day using the newly predicted data.

1.5 Technology and Solution Survey

The subject of hotel cancellations has been the subject of numerous different projects. The majority of the machine learning techniques implemented in each of those projects are largely comparable to retrospect, including KNN, logistic regression, Cat boost, linear regression, AdaBoosting, and Gradient Boosting. All currently used classification-based learning algorithms in machine learning.

All machine learning models fall under the category of supervised learning, which means that every data point has the specified label. Nevertheless, depending on the dataset, it is still feasible to build a model using unsupervised learning, semi-supervised learning, or reinforcement learning.

The various machine learning methods that have been applied in the past to the project's target problem are contrasted below. This would assist us in reducing the number of algorithms we need to include in our model design in order to achieve the best outcomes. The table 1 illustrates the technology and literature survey.

Table 1

Technology and literature survey

Authors	**Objective**	**Datasets**	**Models Used**	**Results**
Cho, V. (2003)	Comparison of three different approaches to tourist arrival forecasting in Hong Kong where we know the demand for the hotels which can be useful for the project where we can predict how many cancellations were done.	Hong Kong Tourist Association (1974–2000)	compares exponential smoothing, seasonal ARIMA and ANN in forecasting	Neural Network gave the higher accuracy

Sánchez-Medina, A. J., & Eleazar, C. (2020)	Using machine learning and big data for efficient forecasting of hotel booking cancellations so that hotels can minimize lost profits. And also, it identifies guests likely to cancel a hotel booking.	four-star hotel partner located in Gran Canaria (Spain) were used	Random forest, Support vector machine, ANN (GA optimized), ANN (GA optimized)	ANN (GA optimized) outperformed and gave higher accuracy
Adil, M., Ansari, M. F., Alahmadi, A., Wu, J. Z., & Chakrabortty, R. K. (2021).	Solving the Problem of Class Imbalance in the Prediction of Hotel Cancellations	Portuguese hotel chain data	Random forest with SMOTE-ENN	Solved the Imbalanced problem
Antonio, N., de Almeida, A., & Nunes, L. (2017, December).	Predicitng hotel bookings cancellation with a machine learning classification model	Property Management Systems data in Portugal were.	XGBoost is used.	Identified the most frequent cancelled bookings customers details and predicted the hotel rooms which are “likely to cancel”.

2 Data and Project Management Plan

2.1 Data Management Plan

Data has been collected from the research paper performed by Nuno Antonio on hotel cancellation. All raw, processed and training data will be stored in google drive. After data processing, the dataframes are converted into CSV files and stored in google drive. So, in modelling phase we will use the previous CSV file will be converted to dataframe for models training. Highest accuracy model will be hyper parameter tuned and then it is saved in pickle format by using the pickle library. Because of this change to pickle format, it is helpful for model deployment. Permission for accessing the data from google drive are given only to the five team members.

2.2 Project Development Methodology

Zoom meetings and Microsoft teams served as task management tools for us. For current and completed documentation, we have used Google Drive as a storage system. All the team members have complete access to the project individually. Their access is controlled through SJSU-provided email accounts. Each team member will be in charge of particular aspects of the project. Data collection, exploratory data analysis, data cleaning, model training, hyperparameter tuning, and deployment are some examples of these tasks.

The group will be required to define a topic that addresses a significant societal issue. In this instance, hotel management's interest-worthy subject was hotel booking cancellation. The Cross Industry Standard Process for Data Mining (**CRISP-DM**) CRISP-DM methodology has been used to carry out our project. The following table 1lists the tasks that must be completed at each project stage.

2.2.1 CRISP – DM

Table 2

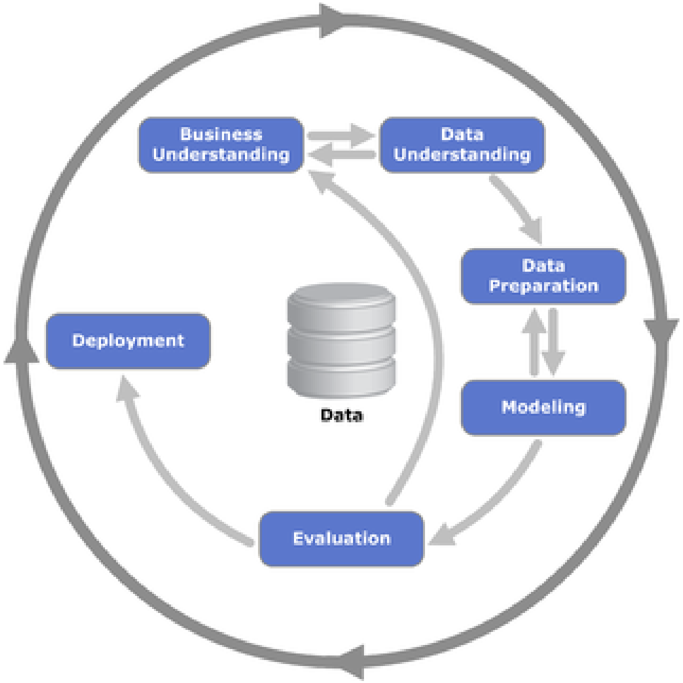
CRISP -DM methodology explanation

Business Understanding	Business Understanding is the first and most important step in our project using the CRISP-DM approach. The main objective of this project is to understand the hotel booking cancellation in the property management system that were provided and separate and prioritize the most vital information. Possibility of cancellation happening and how to identify that it is a valid cancelled booking.
Data understanding	In this step, our team collected various datasets and explored the data. Understand the purpose of each feature in the data and how using it can help you test the model. Identify which of the features provide the most importance for the model Understand how the selected features support to the business understanding
Data Preparation	Clean the data and organize the data into a form that can be used for the data model. We have differentiated the numerical and categorical features by using the D- type function. Here we have 10 categorical

	columns, and the remaining all are numerical columns. Also we have dropped columns that were not useful and normalized numerical variables. Split the values for training and testing.
Model Development	Identify classification models that support cancellation booking detection. Explore different Python Libraries that provide accurate and fast execution. Develop models and train them.
Model Evaluation	This section covers model architecture, model fine-tuning (there are 11 models), and model testing. Determine which accuracy scoring method provides the best prediction for cancellation. Choosing the best model which has the highest accuracy.
Deployment	We will begin implementing the Deployment Stage, as soon as we get the approved model from the evaluation stage, which has the maximum accuracy as mentioned above in the evaluation phase. Once the model evaluations are completed use the best models for prediction and ensemble the results for hotel booking cancellations.

Figure 1

CRISP – DM Flow Diagram



2.3 Project Organization Plan

For this project, one team member will serve as the group leader. Throughout the project, the group leader will be in charge of delegating tasks and updating the team. In order to prevent multiple team members from working on the same task, the team leader assigns each team member a task using Zoom meetings. This aids in setting expectations for how long each task ought to take each person, as well as determining whether the project is progressing according to schedule. Each task owner provides a projected completion date, and daily standup meetings are used to monitor progress.

2.4 Project Resource Requirement and Plan

Compute: Google Colab Pro Plus

Config: NVIDIA V100 Processor, Cost - $50

Storage: Google Drive

Software: Python, Google Drive

Tools: Python notebook, Google Docs, Google Sheets, Zoom

The approximate cost expected to accrue by Google colab pro plus is 75$ for the project timeline (2 months). Other tools we have used were either open-source or free for SJSU student accounts.

2.5 Project Schedule

Below mentioned are our project milestones and the final (target) dates for each phase to complete in time successfully.

1) **Business and Data Understanding - 9/7/2022 to 9/28/2022**

- Hotel Booking Cancellation schema understanding.
- Understanding the sample dataset
- Business use cases for applying ML model

2) **Design and Model Analysis – 9/29/2022 to 10/13/2022**

- Brainstorm classification models
- System design
- Architectural decisions
- ABT and Feature analysis

3) **Model Implementation – 10/14/2022 to 10/31/2022**

- Hotel booking cancellation model implementation
- Training and testing the model
- Measuring model performance

4) **Model Evaluation – 11/2/2022 to 11/16/2022**

- Comparing the model and performance score
- Choosing the best model
- Saving the model for using in real time hotel booking cancellations.

5) **Deployment – 11/18/2022 to 12/1/2022**

- Deployment of the model
- Monitoring and Reporting

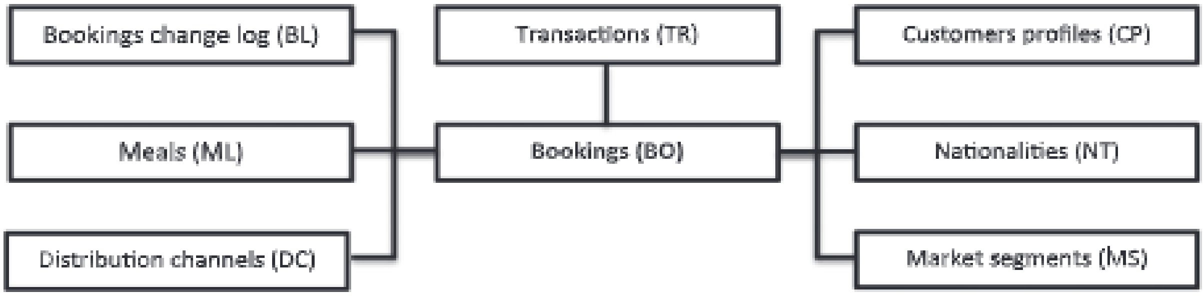
3 Data Engineering

3.1 Data Process

The whole procedure for gathering, processing, validating, and ultimately sorting the data will be covered in this process. Most of the research on Revenue Management demand forecasting and prediction issues in the tourism and travel-related industries uses data from the aviation industry in the Passenger Name Record format (PNR). The aviation industry created a format. Without industry-specific data, it is impossible to completely understand the requirements and peculiarities of the remaining tourism and travel sectors, such as hospitality, cruise, theme parks, etc. To assist overcome this restriction, two hotel datasets with Demand data was shared. The statistics that are presently accessible were gathered with the intention of creating prediction models that would categorize the possibility that a hotel reservation will be canceled. However, these datasets use transcend beyond this cancellation prediction issue because of the properties of the variables they include. The data contain multiple database values that are shown in Figure 2.

Figure 2

The Dataflow of Databases



3.2 Data Collection

The primary data was gathered from two source Hotels and a Resort Figure 3 is illustrating the ratio of data. It can be seen in Figure 3, 33.6% of the data belongs to Resort hotel and 66.4 % of the data belongs to City Hotel. Table 3 illustrates 31 attributes that data contain. One of Nuno Antonio's surveys on hotel management served as the source of the data for this study's literature review.

Figure 3

The Ratio of Hotel and Resort Data

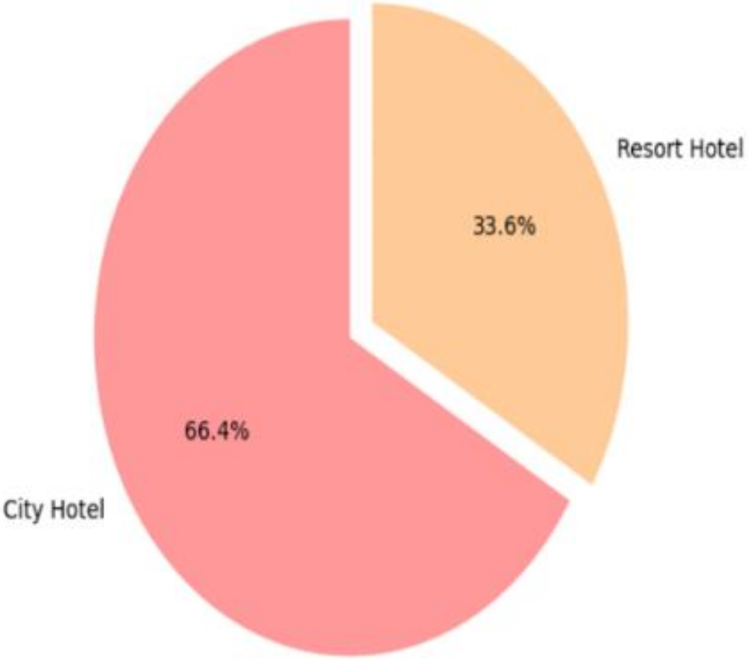
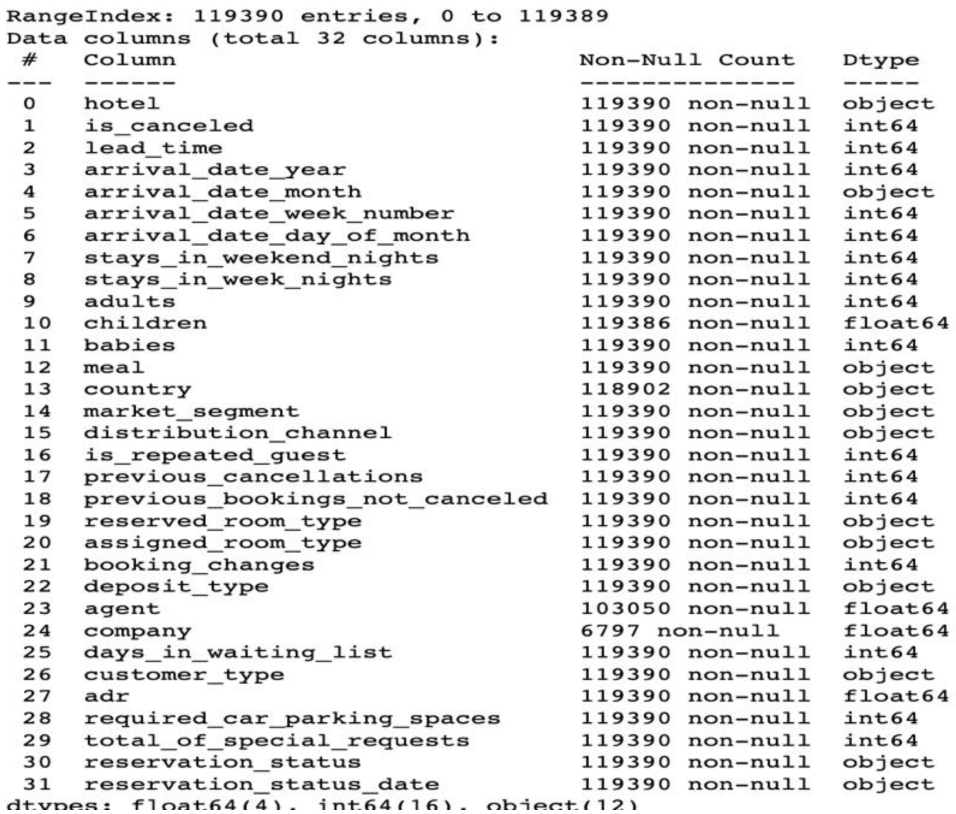


Table 3

Data attributes



```
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   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)
```

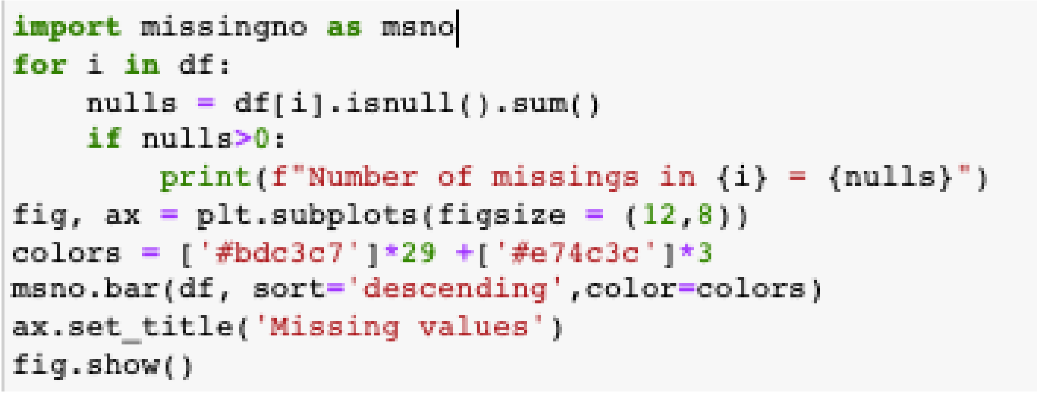
3.3 Data Pre-Processing

Checking the missing values in the data

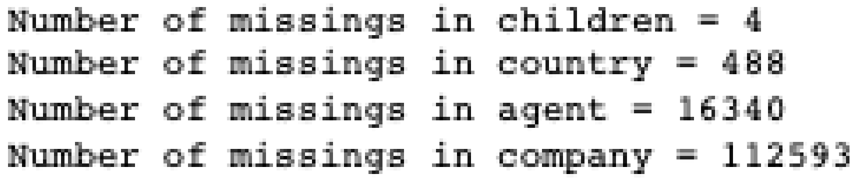
The first step we performed for data preprocessing is to check the missing values of data to enhance the quality of data. As can be seen in Figure 5 Country, Agent and company attribute has fewer values than other attributes.

Figure 4

Missing Values



```
import missingno as msno
for i in df:
    nulls = df[i].isnull().sum()
    if nulls>0:
        print(f"Number of missings in {i} = {nulls}")
fig, ax = plt.subplots(figsize = (12,8))
colors = ['#bdc3c7']*29 +['#e74c3c']*3
msno.bar(df, sort='descending',color=colors)
ax.set_title('Missing values')
fig.show()
```



```
Number of missings in children = 4
Number of missings in country = 488
Number of missings in agent = 16340
Number of missings in company = 112593
```

Figure 5

Missing Attributes Values

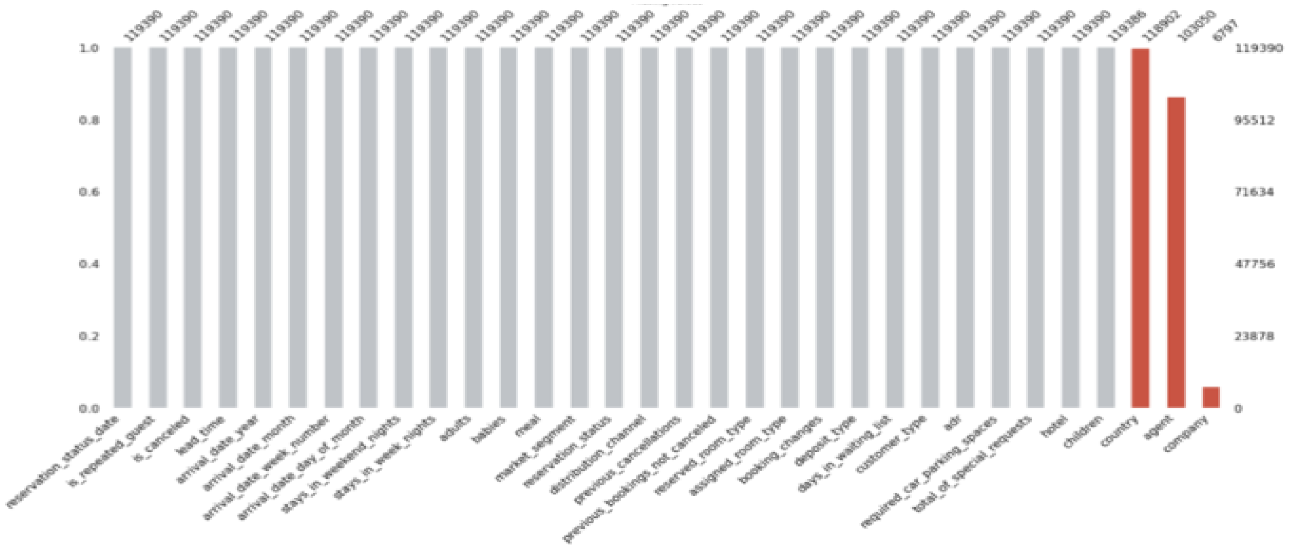


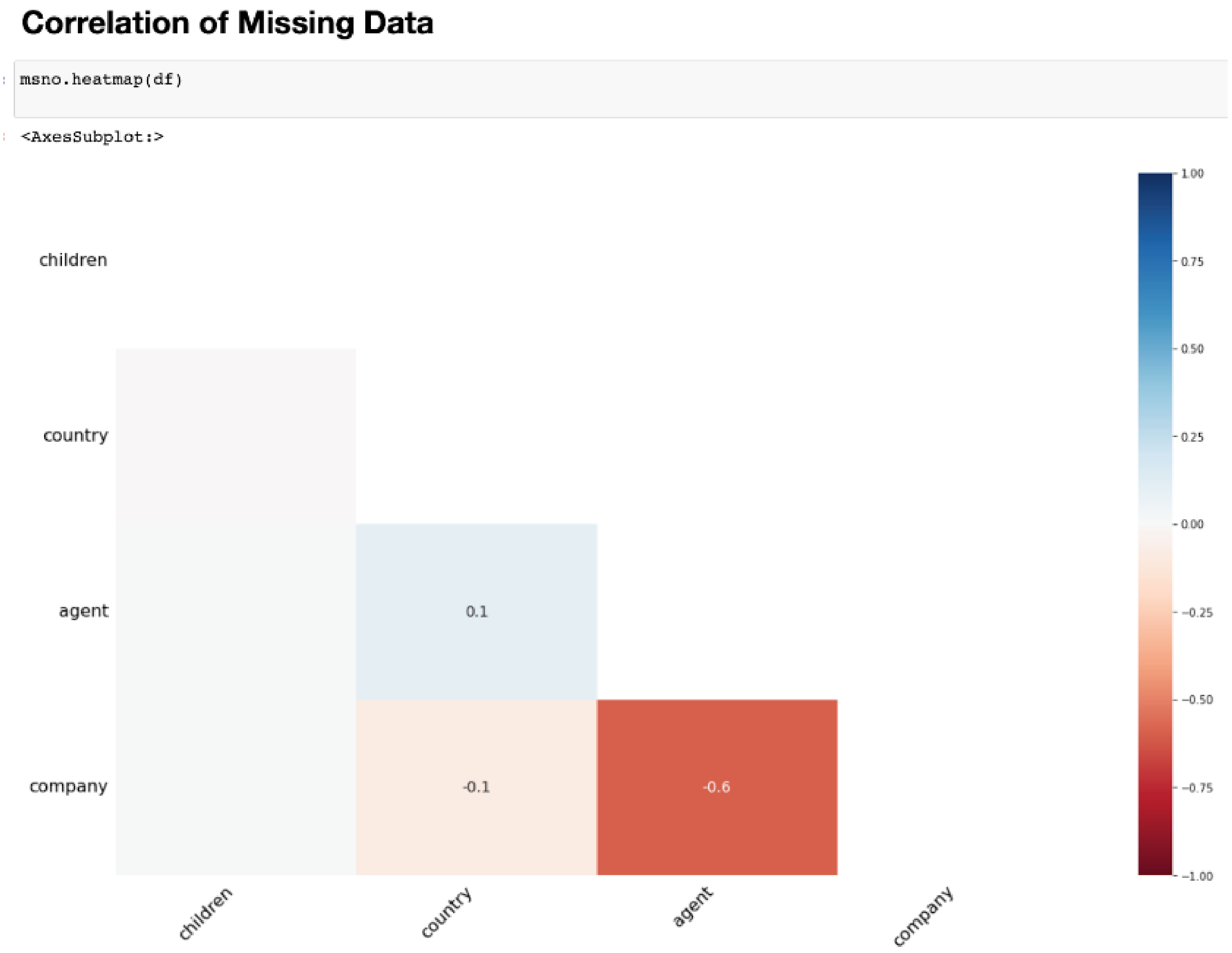
Figure 6 shows the correlation of the missing values, where it can be seen that the agent and company has less than 1 % relationship, we removed these attributes from the data using the given code syntax. df=df.drop(['agent','company'],axis=1). And Country has 488 rows with NaN values. 488 rows out of

119390 are negligible hence we just removed the rows.

df = df.dropna(axis = 0)

Figure 6

Heatmap



Correlation of Missing Data

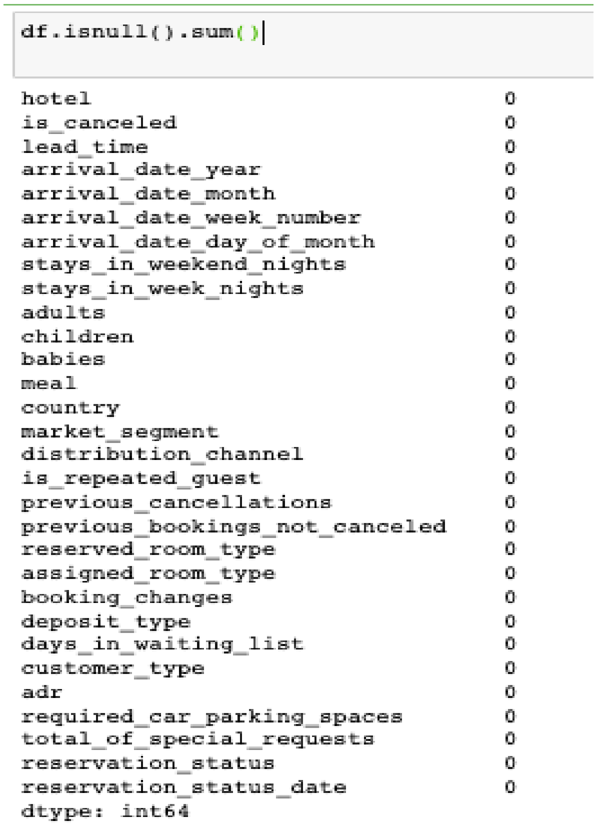
```
msno.heatmap(df)
```

```
<AxesSubplot:>
```

It can be seen in Figure 7 there are no null values in data.

Figure 7

Check Null Values

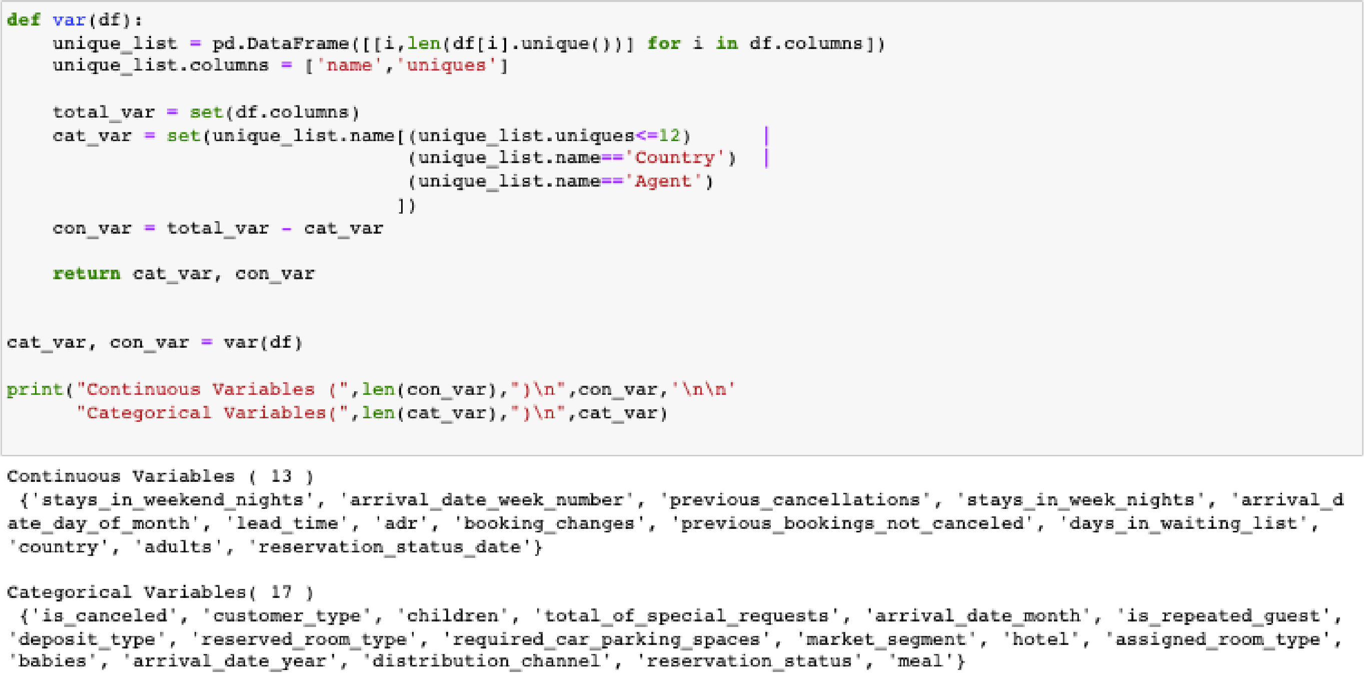


```
df.isnull().sum()
```

```
hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           0
stays_in_week_nights              0
adults                            0
children                          0
babies                            0
meal                              0
country                           0
market_segment                    0
distribution_channel              0
is_repeated_guest                 0
previous_cancellations            0
previous_bookings_not_canceled    0
reserved_room_type                0
assigned_room_type                0
booking_changes                   0
deposit_type                      0
days_in_waiting_list              0
customer_type                     0
adr                               0
required_car_parking_spaces       0
total_of_special_requests         0
reservation_status                0
reservation_status_date           0
dtype: int64
```

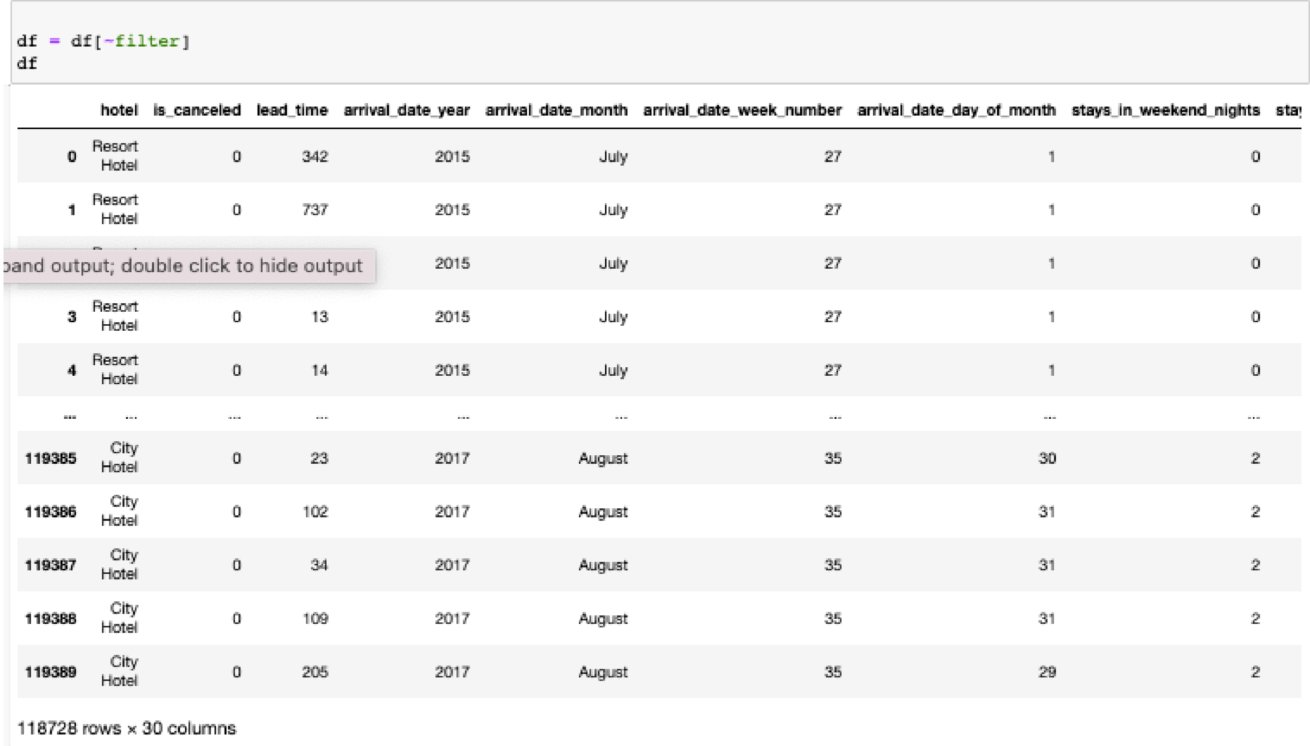
Checking the Categorical and Continuous value

It can be seen in the output that data includes 13 continuous and 17 categorical attributes. List of Continuous Variables with Outliers identified through Summary Statistics are: Lead_time, days_in_waiting_list, stays_in_weekend_nights, adults, previous_bookings_not_canceled, previous_cancellations. We inspected list of Continuous Variables which should be Categorical eg. previous_bookings_not_canceled, previous_cancellations. babies, adults, and children cannot be zero at the same time, so we removed all the instances having zero at the same time.



```
def var(df):
    unique_list = pd.DataFrame([[i,len(df[i].unique())] for i in df.columns])
    unique_list.columns = ['name','uniques']

    total_var = set(df.columns)
    cat_var = set(unique_list.name[(unique_list.uniques<=12)      |
                                  (unique_list.name=='Country')   |
                                  (unique_list.name=='Agent')
                                 ])
    con_var = total_var - cat_var

    return cat_var, con_var


cat_var, con_var = var(df)

print("Continuous Variables (",len(con_var),")\n",con_var,'\n\n'
      "Categorical Variables(",len(cat_var),")\n",cat_var)
```

```
Continuous Variables ( 13 )
 {'stays_in_weekend_nights', 'arrival_date_week_number', 'previous_cancellations', 'stays_in_week_nights', 'arrival_d
ate_day_of_month', 'lead_time', 'adr', 'booking_changes', 'previous_bookings_not_canceled', 'days_in_waiting_list',
'country', 'adults', 'reservation_status_date'}

Categorical Variables( 17 )
 {'is_canceled', 'customer_type', 'children', 'total_of_special_requests', 'arrival_date_month', 'is_repeated_guest',
'deposit_type', 'reserved_room_type', 'required_car_parking_spaces', 'market_segment', 'hotel', 'assigned_room_type',
'babies', 'arrival_date_year', 'distribution_channel', 'reservation_status', 'meal'}
```



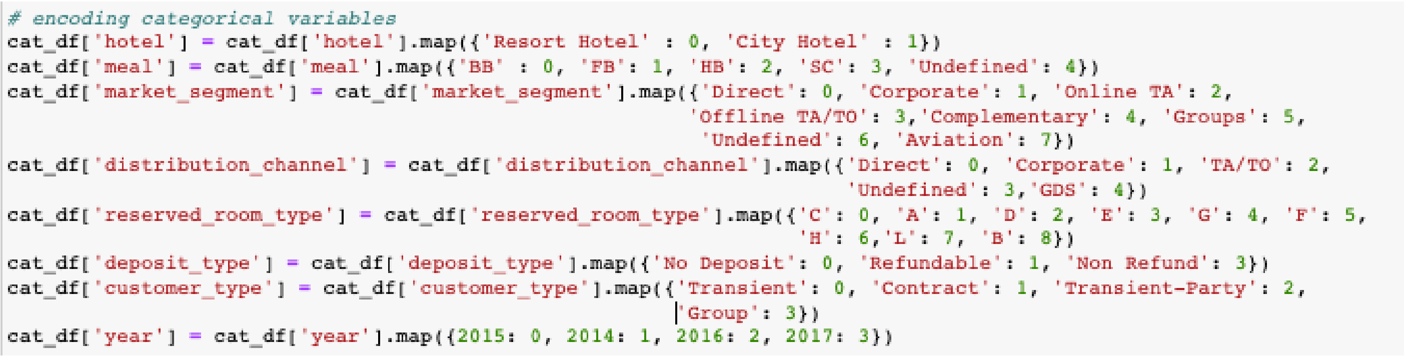
```
df = df[~filter]
df
```

pand output; double click to hide output

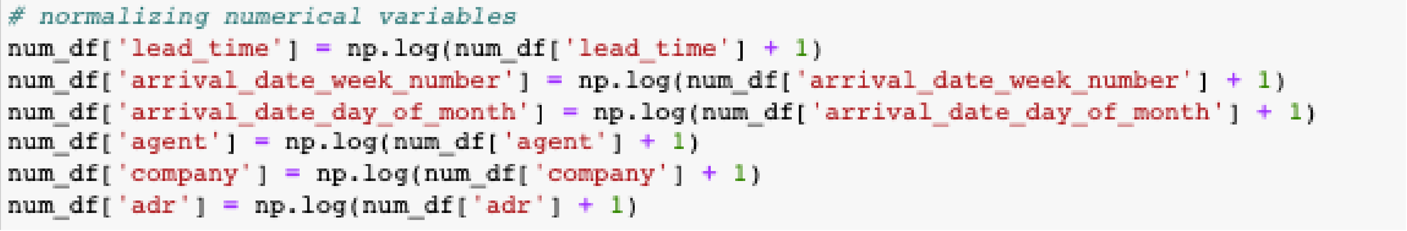
	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	sta
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	
[illegible]	[illegible]	[illegible]	[illegible]	2015	July	27	1	0	
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	
...	...	...	...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	35	30	2	
119386	City Hotel	0	102	2017	August	35	31	2	
119387	City Hotel	0	34	2017	August	35	31	2	
119388	City Hotel	0	109	2017	August	35	31	2	
119389	City Hotel	0	205	2017	August	35	29	2	

118728 rows × 30 columns

After filtering the values performed encoding on categorical variables to make numerical using 0,1,2,.. values. Finally normalized the value before performing the modeling.



```
# encoding categorical variables
cat_df['hotel'] = cat_df['hotel'].map({'Resort Hotel' : 0, 'City Hotel' : 1})
cat_df['meal'] = cat_df['meal'].map({'BB' : 0, 'FB': 1, 'HB': 2, 'SC': 3, 'Undefined': 4})
cat_df['market_segment'] = cat_df['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2,
                                                         'Offline TA/TO': 3,'Complementary': 4, 'Groups': 5,
                                                          'Undefined': 6, 'Aviation': 7})
cat_df['distribution_channel'] = cat_df['distribution_channel'].map({'Direct': 0, 'Corporate': 1, 'TA/TO': 2,
                                                                     'Undefined': 3,'GDS': 4})
cat_df['reserved_room_type'] = cat_df['reserved_room_type'].map({'C': 0, 'A': 1, 'D': 2, 'E': 3, 'G': 4, 'F': 5,
                                                                 'H': 6,'L': 7, 'B': 8})
cat_df['deposit_type'] = cat_df['deposit_type'].map({'No Deposit': 0, 'Refundable': 1, 'Non Refund': 3})
cat_df['customer_type'] = cat_df['customer_type'].map({'Transient': 0, 'Contract': 1, 'Transient-Party': 2,
                                                        'Group': 3})
cat_df['year'] = cat_df['year'].map({2015: 0, 2014: 1, 2016: 2, 2017: 3})
```



```
# normalizing numerical variables
num_df['lead_time'] = np.log(num_df['lead_time'] + 1)
num_df['arrival_date_week_number'] = np.log(num_df['arrival_date_week_number'] + 1)
num_df['arrival_date_day_of_month'] = np.log(num_df['arrival_date_day_of_month'] + 1)
num_df['agent'] = np.log(num_df['agent'] + 1)
num_df['company'] = np.log(num_df['company'] + 1)
num_df['adr'] = np.log(num_df['adr'] + 1)
```

3.4 Data Transformation

After performing data preparation data filtering removed the missing values feature of greater than 40 % and removed the instance that has missing values. After encoding the categorical values and attributes were transformed into numerical values and performed normalization using the NumPy log function.

3.5 Data Preparation

By using A k-fold variant called StratifiedKFold produces stratified folds, where each set has about the same proportion of samples from each target class as the entire set. Finding the index inside the original dataset after applying the KFold results in the creation of the X-train, X-test, Y-train, and Y-test, which guarantees that every time the model is run, every dataset will be entirely random. The training and testing set of data are ready to be fed into the models once the KFold is finished.

3.6 Data Statistics

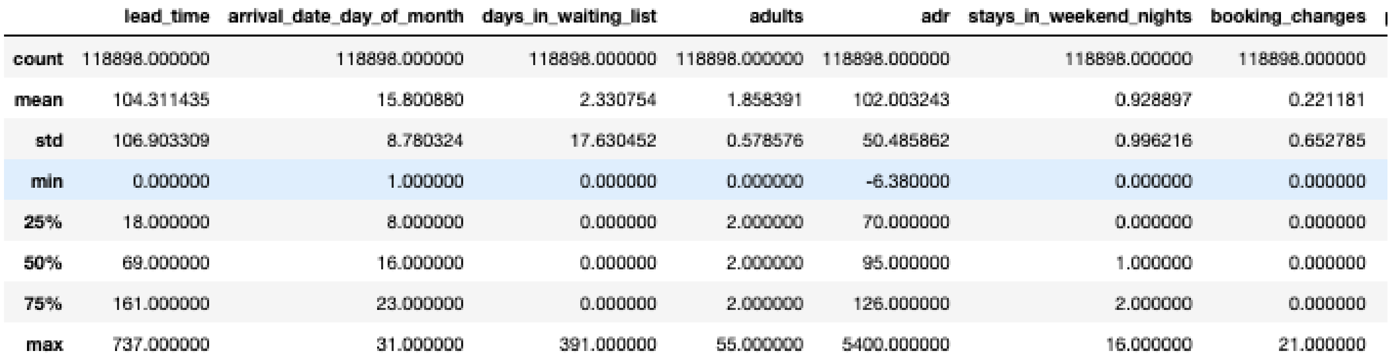
3.6.1 Data Cardinality

It can be seen in the given cardinality table that the target variable is_cancelled has 118898 values

That is the same count that all column has hence, we can assume that no missing value for target variable. Tables 4,5, and 6 display the statistics for continuous variable data and Table 7 displays the statistics for categorical variables, the count of feature instances, mean, and the standard deviation value of the feature as std, as well as the 25%, 50%, and 75% value ranges and the maximum value of the feature value, can be seen in given tables. The number of feature occurrences and the feature's unique value is shown in Table 7 category feature statistics.

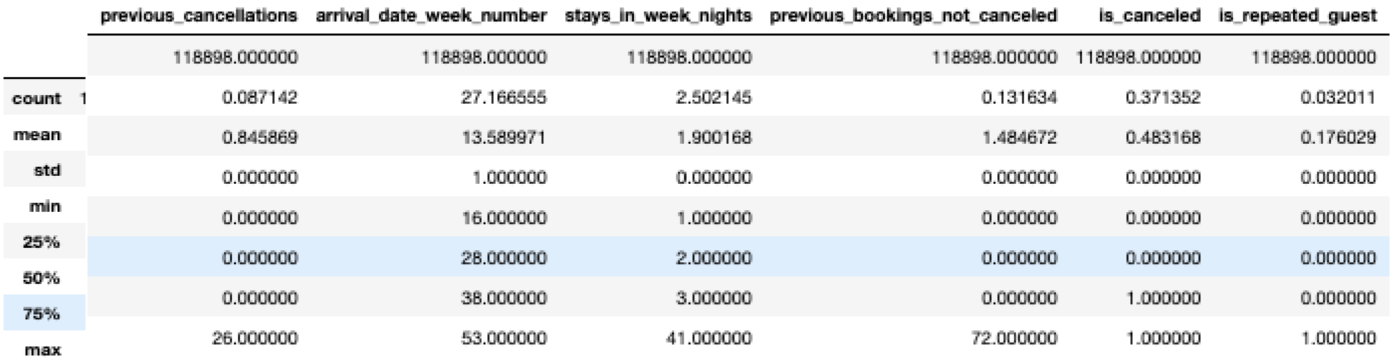
The top value and most common value are displayed in the table.

Table 4



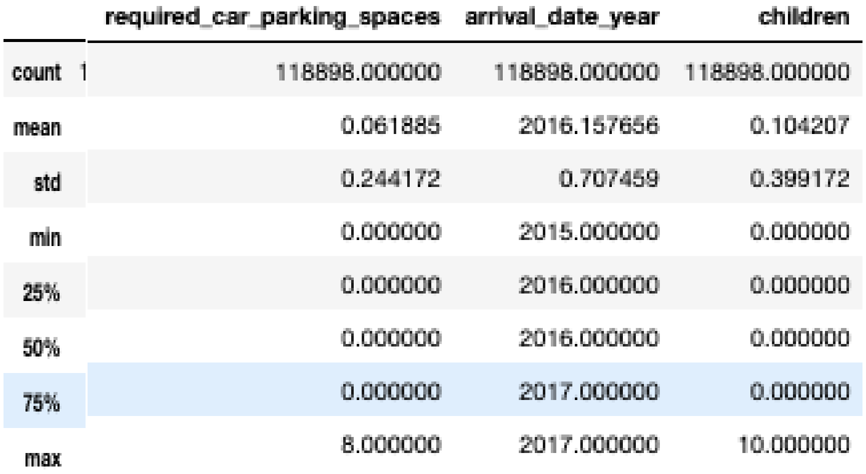
	lead_time	arrival_date_day_of_month	days_in_waiting_list	adults	adr	stays_in_weekend_nights	booking_changes
count	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000
mean	104.311435	15.800880	2.330754	1.858391	102.003243	0.928897	0.221181
std	106.903309	8.780324	17.630452	0.578576	50.485862	0.996216	0.652785
min	0.000000	1.000000	0.000000	0.000000	-6.380000	0.000000	0.000000
25%	18.000000	8.000000	0.000000	2.000000	70.000000	0.000000	0.000000
50%	69.000000	16.000000	0.000000	2.000000	95.000000	1.000000	0.000000
75%	161.000000	23.000000	0.000000	2.000000	126.000000	2.000000	0.000000
max	737.000000	31.000000	391.000000	55.000000	5400.000000	16.000000	21.000000

Table 5



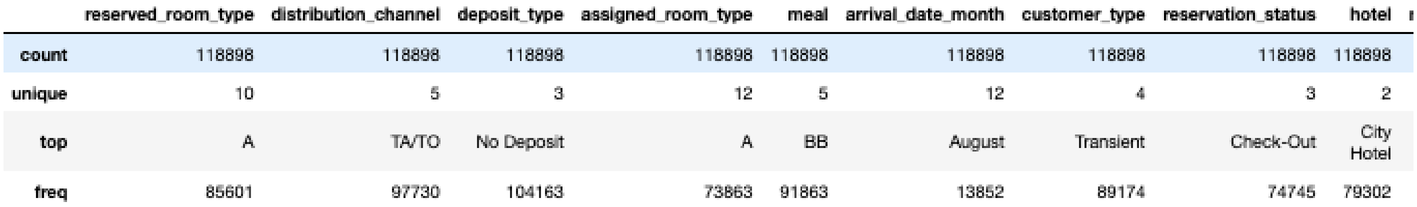
	previous_cancellations	arrival_date_week_number	stays_in_week_nights	previous_bookings_not_canceled	is_canceled	is_repeated_guest
count	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000
mean	0.087142	27.166555	2.502145	0.131634	0.371352	0.032011
std	0.845869	13.589971	1.900168	1.484672	0.483168	0.176029
min	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	16.000000	1.000000	0.000000	0.000000	0.000000
50%	0.000000	28.000000	2.000000	0.000000	0.000000	0.000000
75%	0.000000	38.000000	3.000000	0.000000	1.000000	0.000000
max	26.000000	53.000000	41.000000	72.000000	1.000000	1.000000

Table 6



	required_car_parking_spaces	arrival_date_year	children
count	118898.000000	118898.000000	118898.000000
mean	0.061885	2016.157656	0.104207
std	0.244172	0.707459	0.399172
min	0.000000	2015.000000	0.000000
25%	0.000000	2016.000000	0.000000
50%	0.000000	2016.000000	0.000000
75%	0.000000	2017.000000	0.000000
max	8.000000	2017.000000	10.000000

Table 7



	reserved_room_type	distribution_channel	deposit_type	assigned_room_type	meal	arrival_date_month	customer_type	reservation_status	hotel
count	118898	118898	118898	118898	118898	118898	118898	118898	118898
unique	10	5	3	12	5	12	4	3	2
top	A	TA/TO	No Deposit	A	BB	August	Transient	Check-Out	City Hotel
freq	85601	97730	104163	73863	91863	13852	89174	74745	79302

3.6.2 Data Quality Report

Table 8

Feature	Data Quality Issue	Potential Handling Strategies
'is_canceled'	None	Checked is it containing values other than binary
'is_repeated_guest'	None	Checked is it containing values other than binary

'arrival_date_month'	None	Checked for outlier and format
'total_of_special_requests'	None	Checked for outlier with data distribution
'hotel'	None	Checked for outlier and other than 2 string values
'children'	None	Checked for outlier

country	Remove null instances	Checked for outlier and terms of countries name
lead_time	None	Checked for outlier and format of data values
arrival_date_day_of_month	None	Checked for outlier and format
days_in_waiting_list	None	Checked for outlier
adults	None	verified if all the values are non-negative, then remove the row if there is a negative value.
adr	None	verified if all the values are non-negative, then remove the row if there is a negative value.
stays_in_weekend_nights	None	Checked for outlier and handled if any
booking_changes	None	None
previous_cancellations	None	Checked for outlier and verified if all the values are non-negative, then remove the row if there is a negative value.
stays_in_week_nights	None	Checked for outlier verified if all the values are non-negative, then remove the row if there is a negative value.
previous_bookings_not_canceled	None	Checked for outlier

3.6.3 Analytical Base Table

Figure 8

Analytical Base Table for Hotel Cancellation Prediction

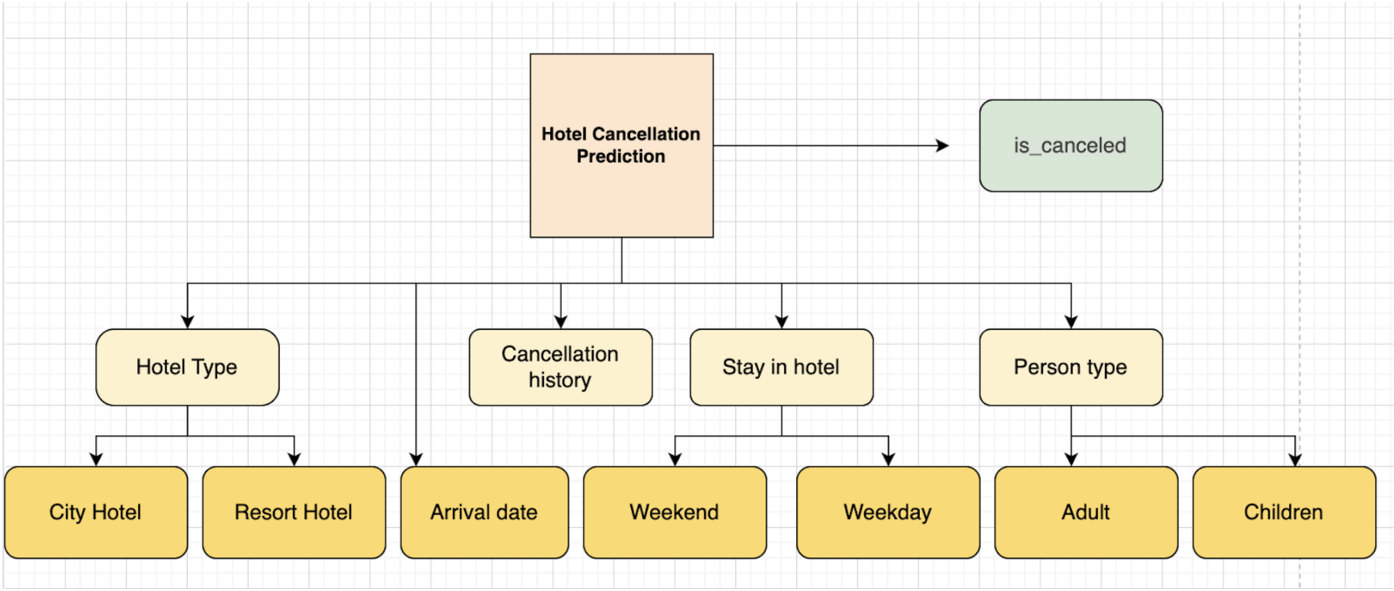
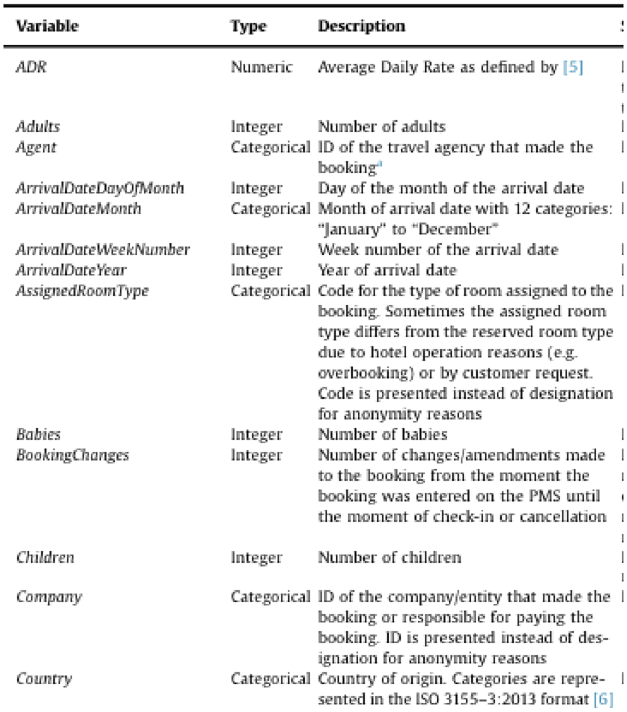
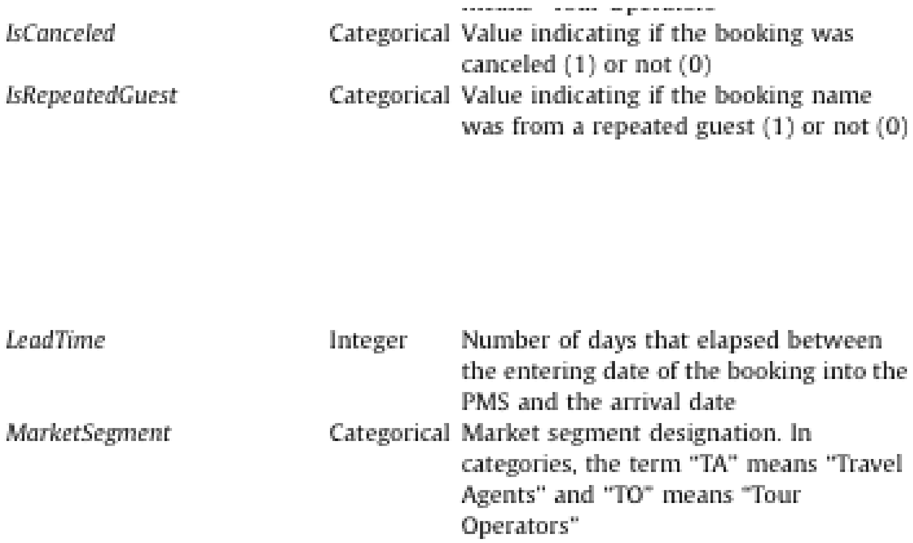


Table 9

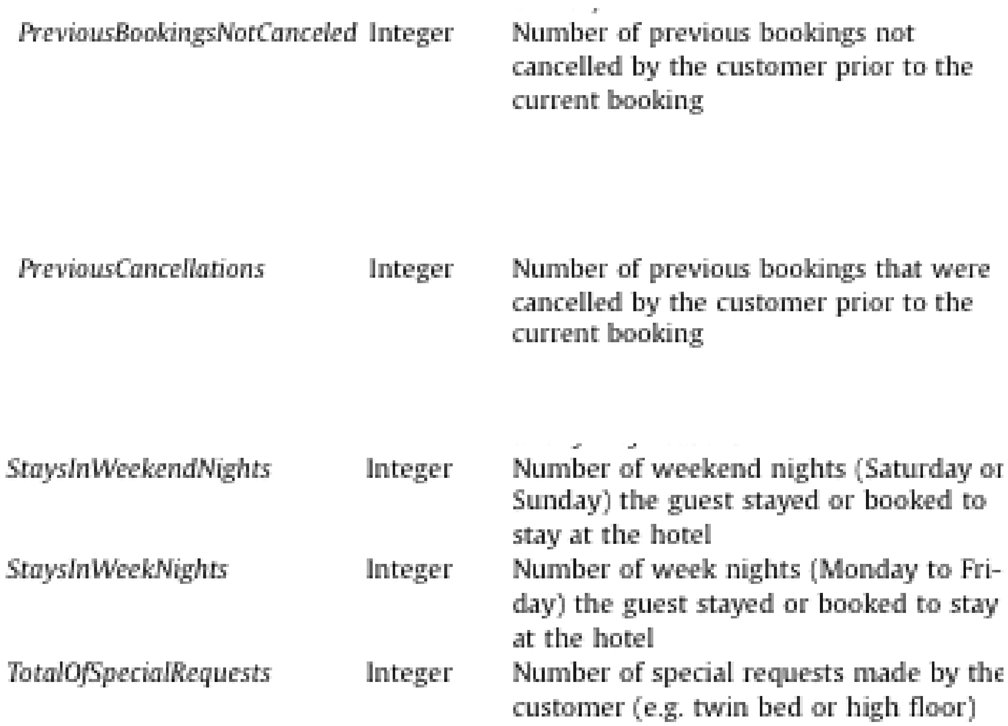
Analytical Base Table



Variable	Type	Description
ADR	Numeric	Average Daily Rate as defined by [5]
Adults	Integer	Number of adults
Agent	Categorical	ID of the travel agency that made the booking[a]
ArrivalDateDayOfMonth	Integer	Day of the month of the arrival date
ArrivalDateMonth	Categorical	Month of arrival date with 12 categories: "January" to "December"
ArrivalDateWeekNumber	Integer	Week number of the arrival date
ArrivalDateYear	Integer	Year of arrival date
AssignedRoomType	Categorical	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons
Babies	Integer	Number of babies
BookingChanges	Integer	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
Children	Integer	Number of children
Company	Categorical	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
Country	Categorical	Country of origin. Categories are represented in the ISO 3155–3:2013 format [6]



IsCanceled	Categorical	Value indicating if the booking was canceled (1) or not (0)
IsRepeatedGuest	Categorical	Value indicating if the booking name was from a repeated guest (1) or not (0)
LeadTime	Integer	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
MarketSegment	Categorical	Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"



PreviousBookingsNotCanceled	Integer	Number of previous bookings not cancelled by the customer prior to the current booking
PreviousCancellations	Integer	Number of previous bookings that were cancelled by the customer prior to the current booking
StaysInWeekendNights	Integer	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
StaysInWeekNights	Integer	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
TotalOfSpecialRequests	Integer	Number of special requests made by the customer (e.g. twin bed or high floor)

3.7 Data Analytics Results

Here we are going to show some analytics that we have performed on our data using some Visualization. Figure 9 shows the plot, from where most of the guests are coming or we can say from where the booking is has made.

Figure 9

Chart Showing Country Data

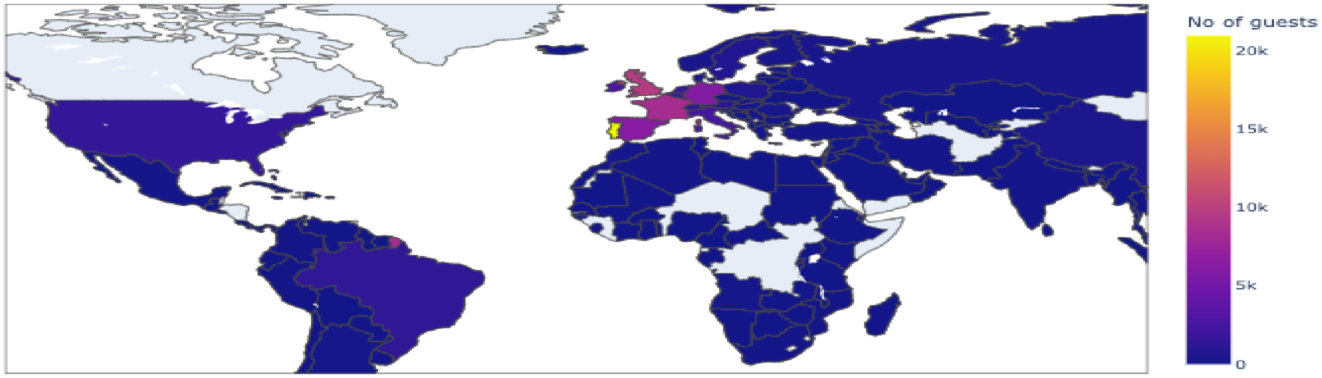
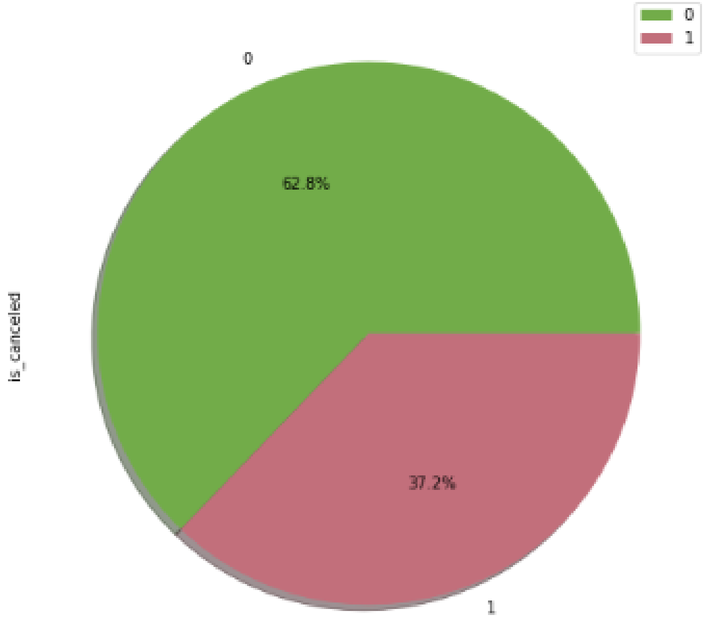


Figure 10 displays client cancellations of reservations, and it is clear that 37.2% of reservations were canceled. Figure 16 shows the total number of visitors every month, and it is quite evident that The most popular month to stay at a hotel in Be Seen City is august.

Figure 10

Cancellation Data Ratio



As seen in Figure 11, August is the busiest month. It follows that hotel rates will also be higher than normal at this time. The City hotel has more guests during spring and autumn, when the prices are also highest, In January and February, there are fewer 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.

Figure 11

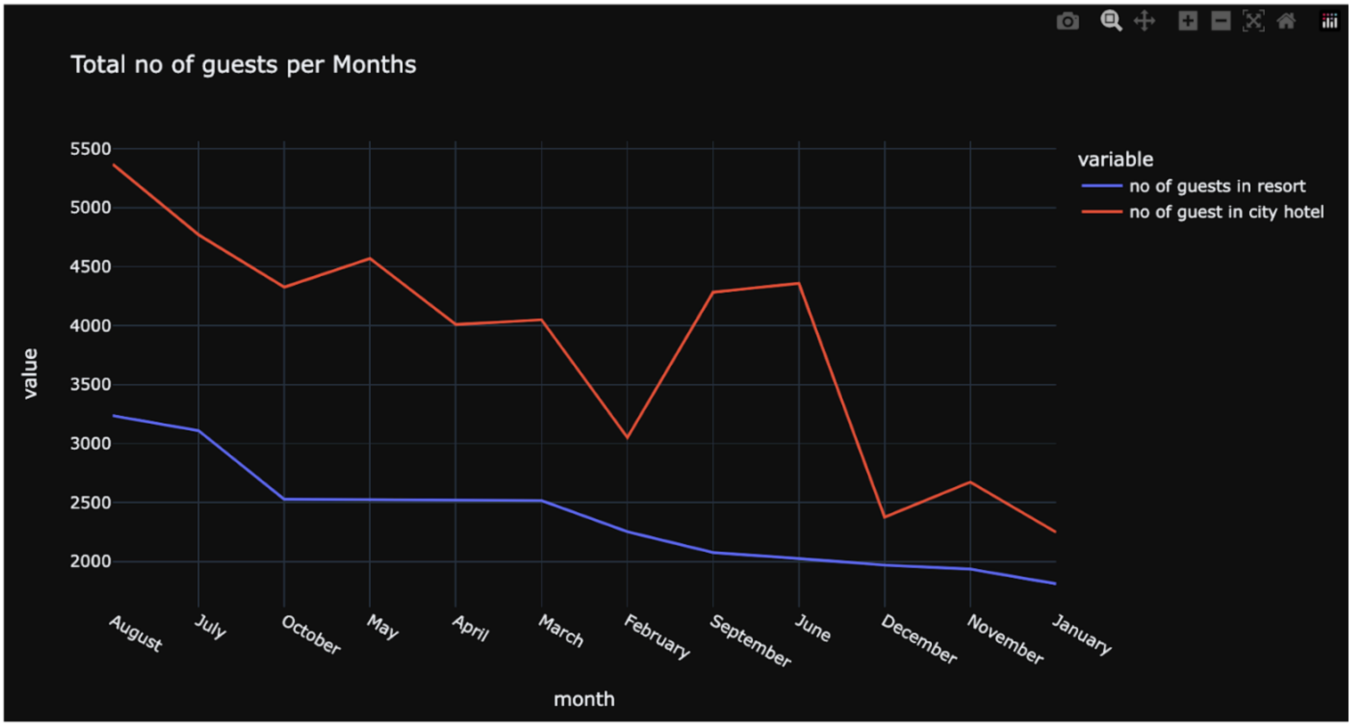


Figure 13 depicts the cancellation of each hotel and resort reservation separately. Additionally, resort hotels have a 50% lower rate of cancellations than city hotels.

Figure 12

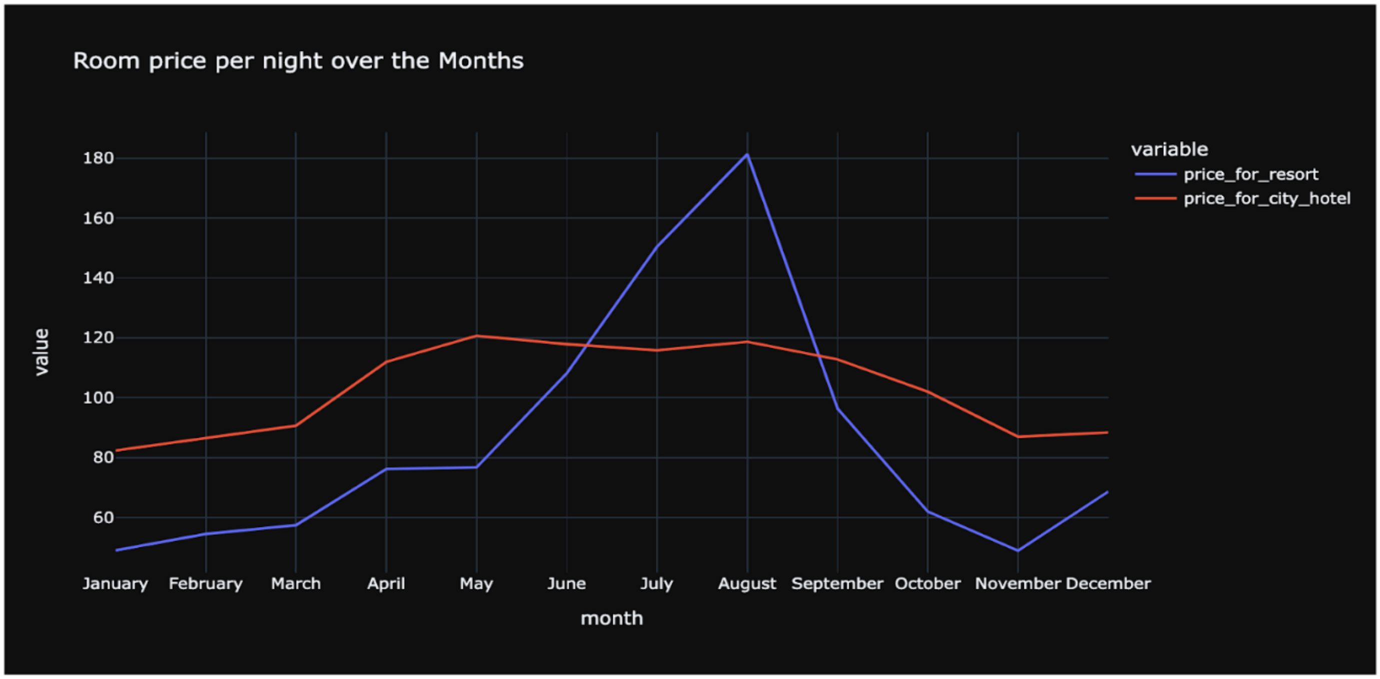


Figure 13

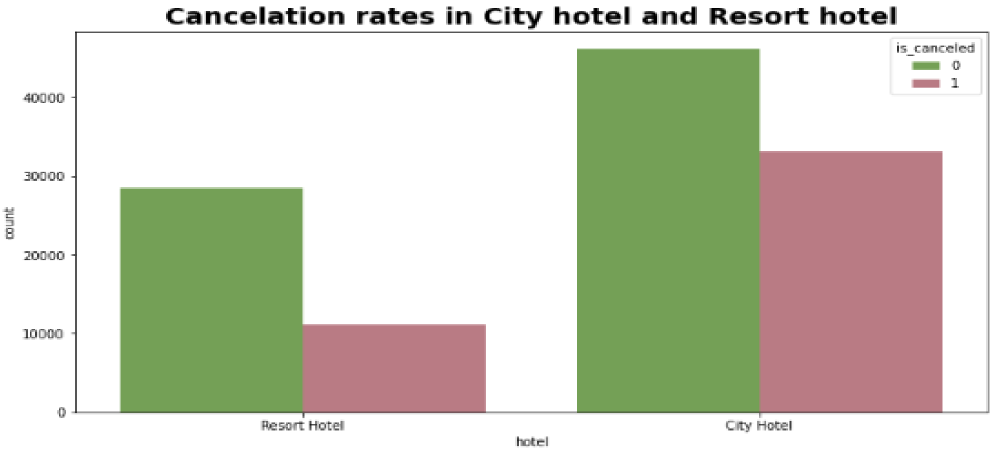
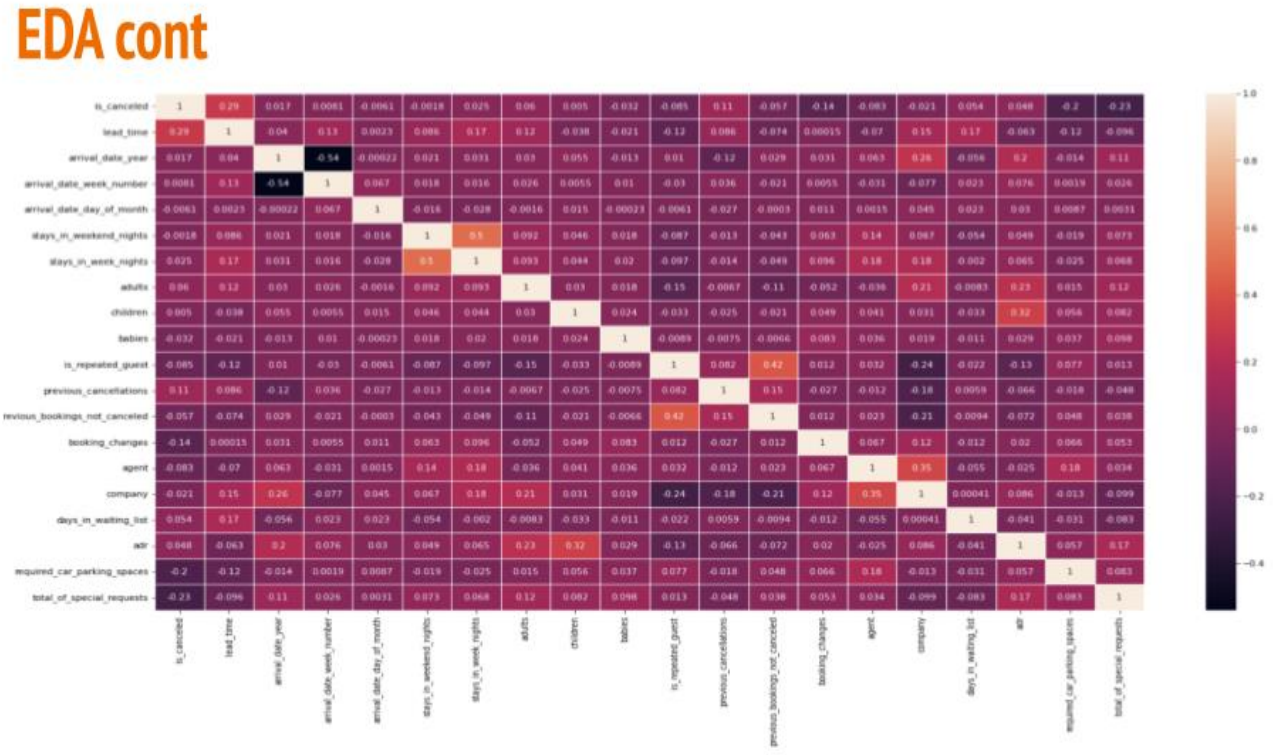
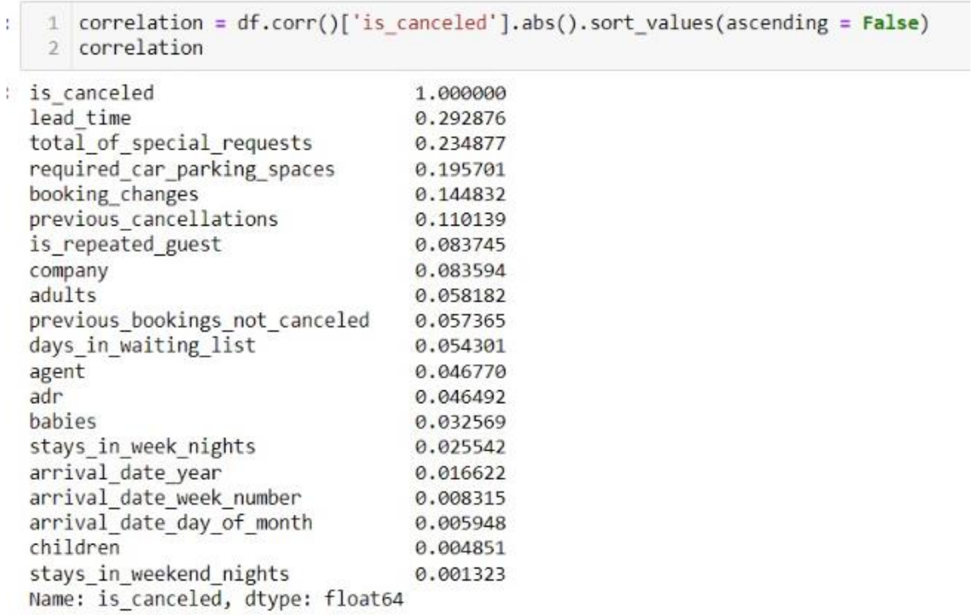


Figure 14 illustrates the correlation of features. We can observe from the correlation heatmap that there are not many features with low correlation values, We deleted the remaining four less important features from the list, such as the arrival date, the week number, and weekend stays.

Figure 14

Corelation heatmap


EDA cont


```
correlation = df.corr()['is_canceled'].abs().sort_values(ascending = False)
correlation
```

```
is_canceled                       1.000000
lead_time                         0.292876
total_of_special_requests         0.234877
required_car_parking_spaces       0.195701
booking_changes                   0.144832
previous_cancellations            0.110139
is_repeated_guest                 0.083745
company                           0.083594
adults                            0.058182
previous_bookings_not_canceled    0.057365
days_in_waiting_list              0.054301
agent                             0.046770
adr                               0.046492
babies                            0.032569
stays_in_week_nights              0.025542
arrival_date_year                 0.016622
arrival_date_week_number          0.008315
arrival_date_day_of_month         0.005948
children                          0.004851
stays_in_weekend_nights           0.001323
Name: is_canceled, dtype: float64
```

4 Model development

4.1 Model Proposals

4.1.1 Model comparison for classification

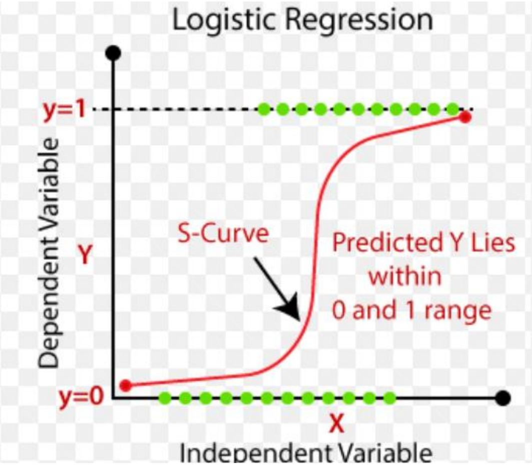
The prediction of hotel booking cancellation is perceived as a classification issue. As a result, depending on the problem statement that we have studied, there are a few models that would be useful in understanding which model delivers a greater prediction outcome.

4.1.2 Logistic Regression

A machine learning technique which also behaves as supervised then it is called logistic regression. It is an interpretation and classification model that returns a probability value after assigning data to a binary set of classes. Logistic regression uses a collection of independent factors to predict the categorical dependent variable. The categorical dependent variable is_cancelled can have a value of 0 or 1. The sigmoid function is used to convert anticipated values to probabilities. It represents the values between 0 and 1. The logistic regression uses the threshold value to compute the S Shape. The value over the threshold is usually 1 (canceled), whereas the value below the threshold is usually 0 (not canceled).

Figure 15

Figure of Logistic Regression



4.1.3 KNN algorithm

The KNN algorithm expects that new and existing data will be comparable. A unique data point is classified using the K-NN algorithm based on similarity after all the existing data has been stored. During the training phase, the KNN algorithm saves the dataset. When employing the KNN method, the euclidean distance and number of data points denoted by K value are the two crucial factors. The most often used distance measure is euclidean distance. Depending on the needs, Hamming distance, Manhattan distance, and Minkowski distance may be utilized

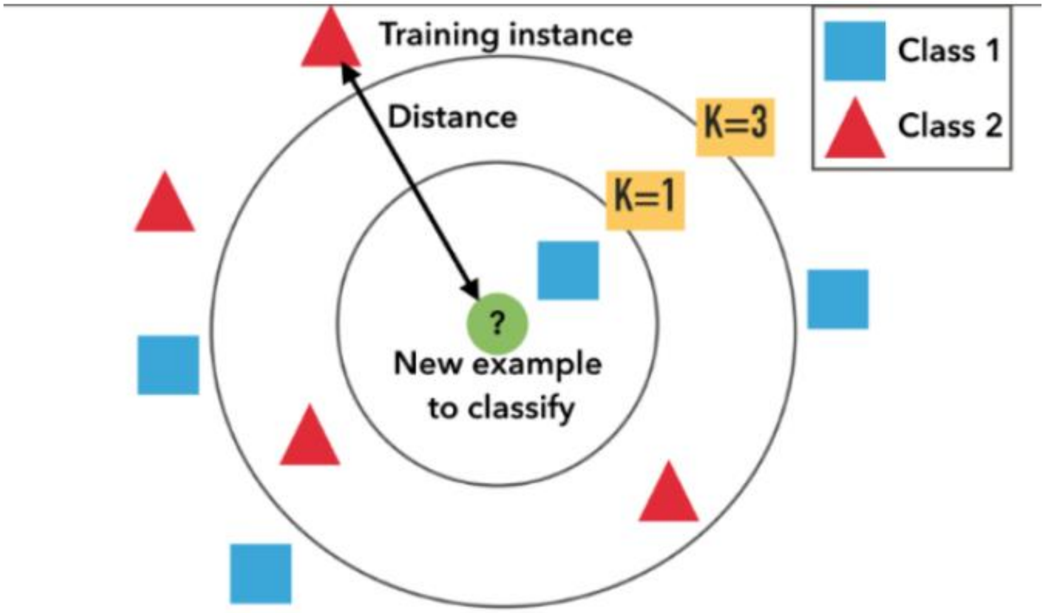
The Euclidean distance, computed as follows, by using the metric:

$$E_d = \sqrt[2]{\Delta x + \Delta y + \ldots + \Delta n}$$

where the goal and training set's binary properties are x, y, and n. Each incoming booking is evaluated using this algorithm for its proximity to the most recent booking and indicates whether it will be canceled or not.

Figure 16

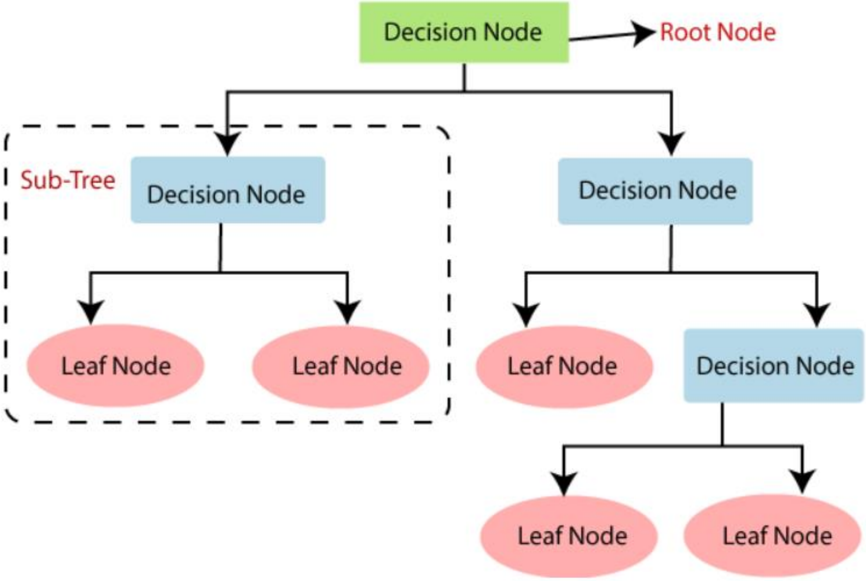
Figure for KNN-algorithm



4.1.4 Decision Tree Classifier

Based on specific thresholds, the decision tree will divide features. The model could decide whether or not the reservations would be canceled by dividing the decisions as they proceeded down the tree. The decision tree's cornerstone is the Gini index or information gain calculation. Decision trees categorize instances by arranging them in a tree starting from the root to a leaf node, which gives the instance's categorization. As seen in the Figure 18, to classify an instance, one tests the attribute given by the root node of the tree before continuing down the branch of the tree that corresponds to the attribute's value. The subtree ingrained at the new instance is then subjected to the same procedure.

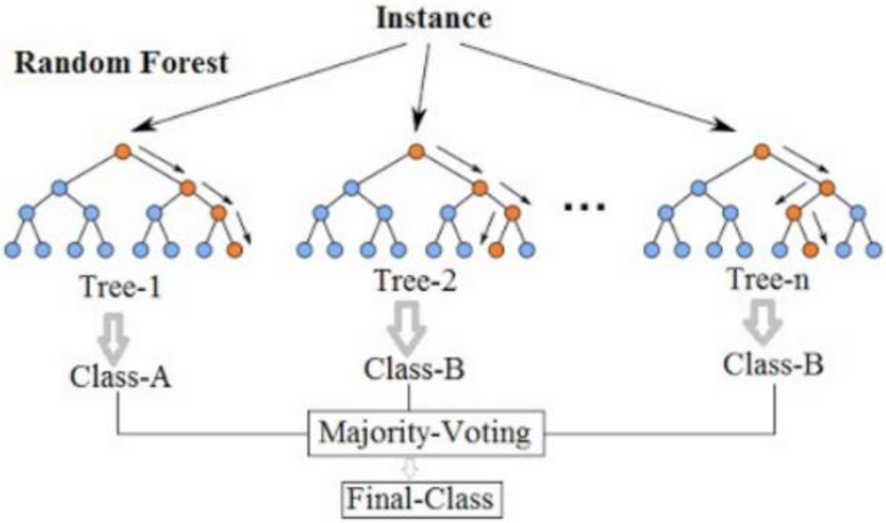
Figure 17



4.1.5 Random Forest

A bagging method called Random Forest uses several models of decision trees in distinct collections of existing information. The Random Forest creates recommendations from a number of decision trees as opposed to the Decision Tree, which only develops one decision tree, and uses the most popular forecast as the final outcome.

Figure 18



4.1.6 AdaBoost Classifier

AdaBoost, also known as Adaptive Boost, is an ensemble technique that makes use of boosting and bagging to increase prediction accuracy. This classification difficulty arises from the binary nature of the target column and is canceled. Each data point will first be given certain weights. All of the weights will start off being equal.

Calculating the sample weights by using the below formula

$$w(x_i, y_i) = \frac{1}{N}, \ i = 1, 2, \ldots . n$$

N stands for the overall number of data points. By calculating each stump's Gini index and choosing the one with the lowest value, you may determine which one categorizes the new sample collection the best. To update the prior sample weights and equalize the new sample weights, compute the "Amount of Say" and "Total error."These instructions should follow repeatedly unless and until a relatively low error is attained.

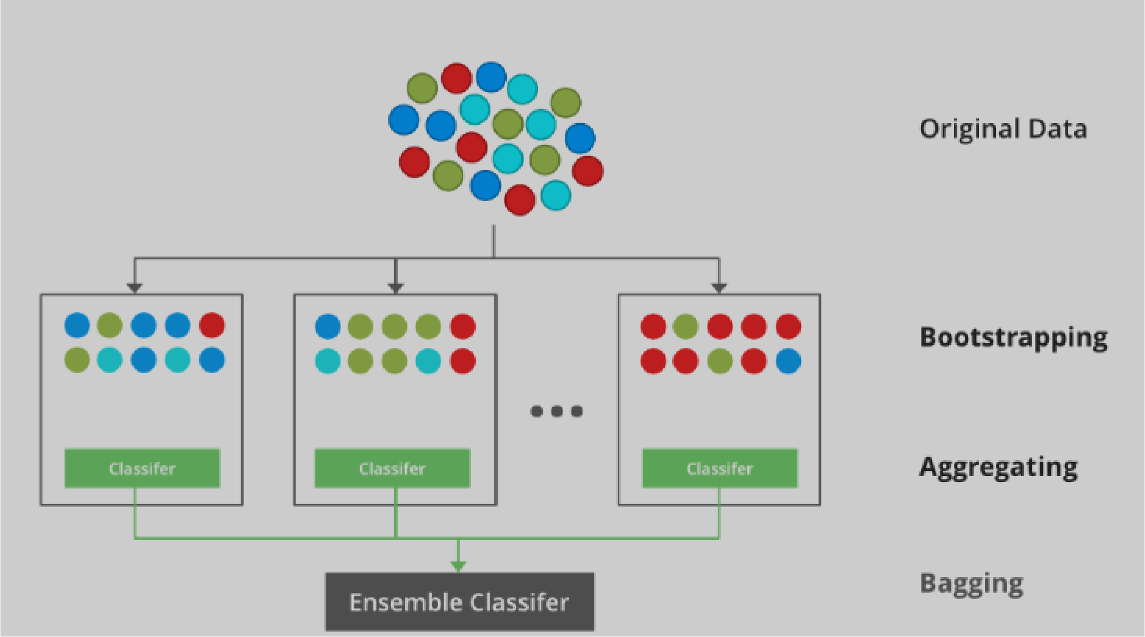
4.1.7 Gradient Boosting Classifier

The primary concept underlying this method is to create models in succession while attempting to minimize the flaws of the prior model. Creating a base model to forecast the data in the training data is the first stage in gradient boosting. In the following stage, the pseudo residuals, which are observed value and predicted value) and create additional predictions using is_canceled as the target feature and the other attributes since decreasing the residuals will gradually increase the accuracy and predictive capacity of the model.

4.1.8 XgBoost

XGBoost is a technique for ensemble learning. The foundation learners, sometimes referred to as the ensemble's models may come from a single learning algorithm or from other learning algorithms and b uild an ensemble model to forecast the cancellation using regression techniques as the base learners. The error rate or MSE should be minimized when the model is first initialized. Later, the residuals are used to teach each learner. The resultant errors at each step are used to model each of the additional learners in boosting. On the surface, it appears that boosting learners employ the similarities in residual mistakes. When boosting reaches its maximum accuracy, the revenues appear to be arbitrarily dispersed with no discernible pattern.

Figure 19



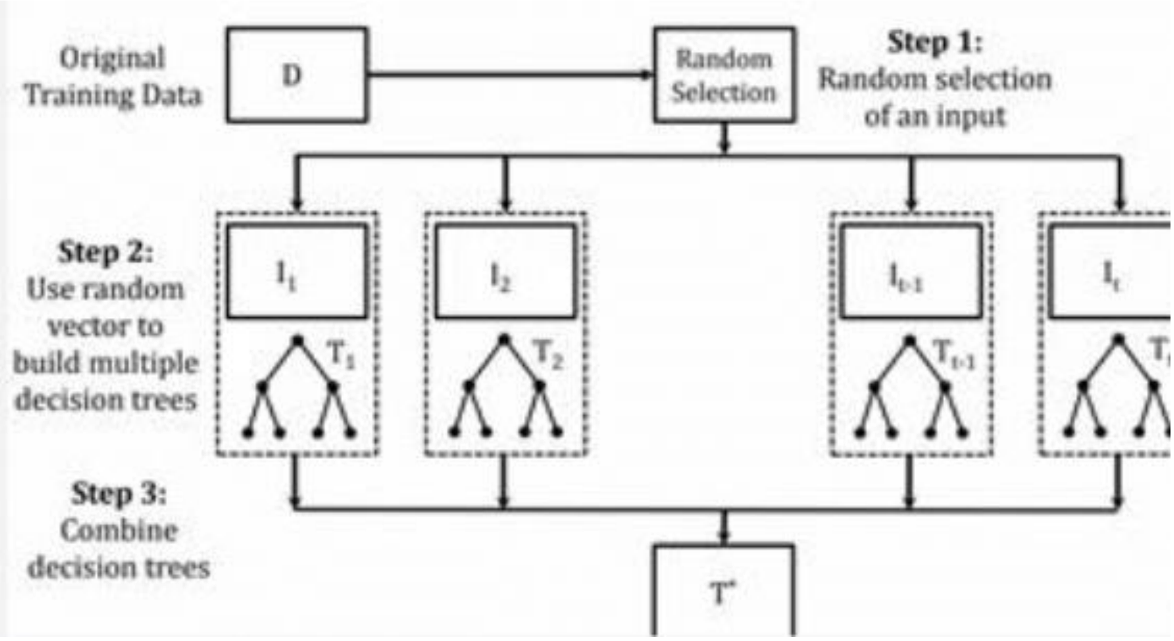
4.1.9 Cat Boost

Without any specific pre-processing, CatBoost may be used to turn categories into numbers. CatBoost uses various statistics on categorical feature combinations and categorical and numerical feature combinations to translate categorical values into numbers. Target statistics are ordered in CatBoost. The algorithms used in online learning, which obtain training examples in a chronological order, are the inspiration for ordered target statistics. Artificial time is introduced, which is a randomized permutation of the training instances. In order to prevent target leakage, it will only depend on the training instances it has already seen in the past (samples taken before that specific sample in the fake time).

4.1.10 Extra Trees Classifier

A sort of ensemble learning algorithm called Extremely Randomized Trees Classifier combines the output of many de-correlated decision trees. The first training sample is used to build each prediction model which is a decision tree in the Extra Trees Forest. Then, each tree is given a random selection of k features from the feature set at each test node, from which it must choose the best feature to divide the data according to certain mathematical criterion which is Gini Index. There are several de-correlated decision trees produced as a result of this random sampling of characteristics.

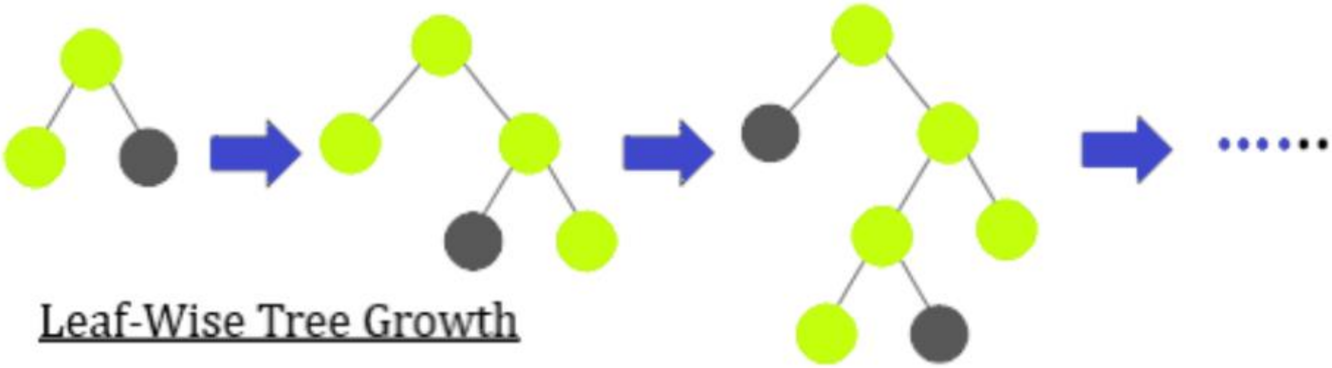
Figure 20


Original Training Data
D
Random Selection
Step 1: Random selection of an input
Step 2: Use random vector to build multiple decision trees
I1
I2
It-1
It
T1
T2
Tt-1
Tt
Step 3: Combine decision trees
T*

4.1.11 LGBM

LightGBM is a decision tree-based gradient boosting framework that improves the model's efficiency while using less memory. It employs exclusive feature bundling and gradient-based one side sampling, two innovative approaches (EFB). The under-trained cases, which have bigger gradients, will provide more information gain. To maintain the precision of the gain, GOSS retains those instances with big gradients (e.g., larger than a preset threshold or in the top percentiles) and only randomly rejects those occurrences with small gradients.

Figure 21



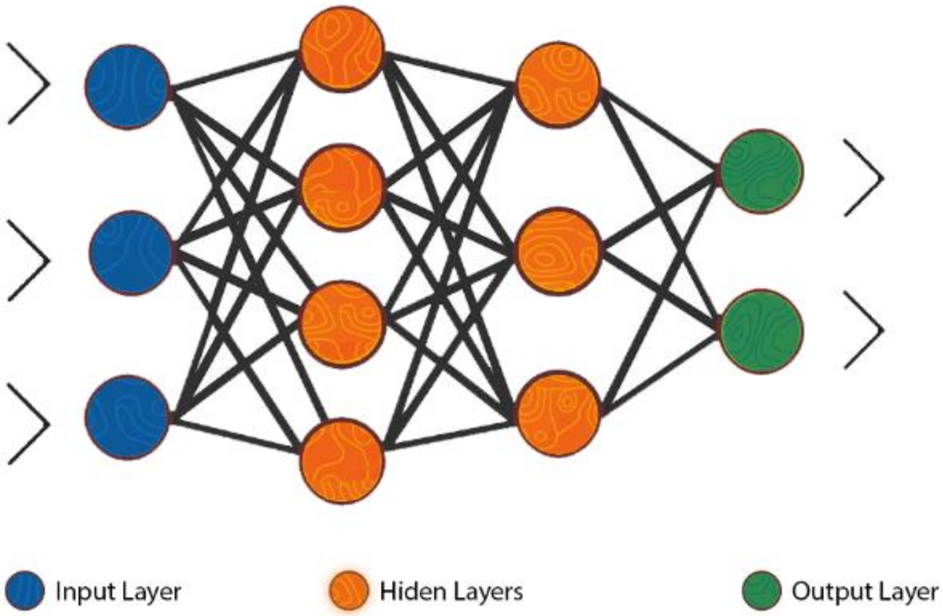
4.1.12. Voting Classifier

A machine learning model that learns from a collection of many models and forecasts an output (class) based on the class that has the highest likelihood of being selected as the output which is called voting classifier. It merely averages the results of each classifier that is fed into the voting classifier and forecasts the output class based on the vote with the largest majority. The Voting Classifier handles two different voting methods. In a hard voting situation, the class with the largest number of votes receives the projected output. Soft voting involves averaging the probabilities assigned to each class, with the output class being the forecast based on that average.

4.1.13 ANN

ANN is adaptable by nature; it has the capacity to modify the network's weights. To map outputs to a particular range, the activation function after learning and modifying the weights. The activation function's primary goal is to make the network less linear. ANN has a strong capacity for prediction. Data quality problems are simply handled by it. It can identify patterns in data that hasn't been seen before and has a greater threshold for noisy data.

Figure 22



4.2 Model Evaluation Methods

To create the models after cleaning the data and dividing it into a 70% training set (about 83,000 rows) and a 30% test set (about 36,000 rows). Before the model is into production on untested data, it should be able to increase its overall predictive power by evaluating its performance using several criteria. When a machine learning model is applied to unexplored data, failing to properly evaluate it using a variety of assessment measures and relying simply on accuracy might result in inaccurate predictions.

The percentage of the situations that were accurately predicted really came true is explained by precision. Precision is helpful when False Positive is more of a worry than False Negative. Recall describesthat how many of the actual positive cases that the model was able to properly anticipate. It provides a synthesis of the Precision and Recall measurements. It reaches its optimum when Precision and Recall are equal.

Figure 23

$$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

In most commercial situations, need not focus much on True Negatives, yet they may contribute significantly to accuracy. False Negative and False Positive are typically associated with business expenses (tangible and intangible); therefore, F1 Score may be a more appropriate metric to utilize.

4.3 Model Validation and Evaluation Results

Accuracy metrics may be used by different machine learning models to determine whether or not they are accurate. As investigated the findings using the f1 score measure. When compared to other models that are presented and estimated in this project, the catboost models offer a greater accuracy of 99.5 percent.

Table 10

Models	**Accuracy**
Logistic Regression	0.809775
KNN	0.88628
Decision Tree Classifier	0.9477
Random Forest Classifier	0.9516
AdaBoost Classifier	0.9466
Gradient Boosting Classifier	0.9100
XgBoost	0.9823
Cat Boost	**0.9959**
Extra Trees Classifier	0.9485
LGBM	0.9631
Voting Classifier	0.9636
ANN	0.9800

4.4 Model Results Discussion

Models including Logistic Regression, KNN, Random Forest Classifier, Ada Boost, Gradient Boosting Classifier, XgBoost, Cat Boost, Extra Trees Classifier, LGBM, Voting Classifier, and ANN are employed in the project. In comparison to previous models, the CatBoost model initially offered a better F1 score of 99.5%. As a result, here Optuna is choosed to use to implement hyperparameter adjustment for the CatBoost model. A software framework called Optuna is used to automate the optimization of these hyperparameters. Through trial and error, it automatically identifies the ideal hyperparameter settings for great performance. Consequently, the CatBoost model gave an F1 score of 100 per cent.

5 System Design and Architecture

5.1 System Requirements Analysis

The hotel sector continues to face a serious problem with booking cancellations as flexible booking alternatives spread. The hotel management team might be informed of the cancellations after forecasting them.

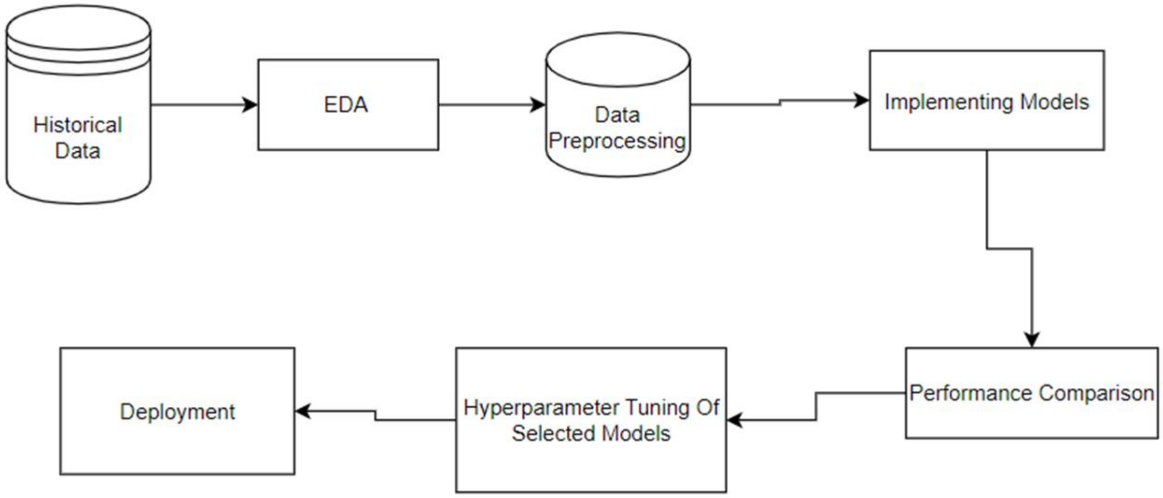
Cases:

Customers occasionally cancel their reservations due to a change in plans, inclement weather, cancelled flights, illness, etc. All of these situations result in significant revenue losses for hotels, which is why machine learning is used in the sector. A crucial component of hotel dynamic pricing is decision support systems built on sophisticated models that forecast the likelihood of reservation cancellation.

5.2 System Design & Architecture

Initially, some research had been done on data and finalized the dataset “hotel_bookings” and then did the exploratory data analysis. Next, some data preprocessing techniques have been applied to the finalized dataset. We have checked the missing and null values and dropped them and then checked the categorical and continuous values. After performing data preparation data filtering removed the missing values feature greater than 40 % and removed the instance that has missing values. After encoding the categorical values, attributes were transformed into numerical values and performed normalization using the NumPy log function. For Data Preparation, used A k-fold variant called StratifiedKFold produces stratified folds, where each set has about the same proportion of samples from each target class as the entire set. In the next step, implemented a few models and then compared the performance of the models. Out of all the models implemented, for the best-performed model fine-hyper tuning was done and then deployed the model was.

Figure 24



5.3 System Supporting Environment

Operating System: Windows

Programming languages: Anaconda Python

Machine learning Libraries: scikit-learn

Other Python Libraries used:

Pandas: dataframe manipulation

Numpy: mathematical manipulation

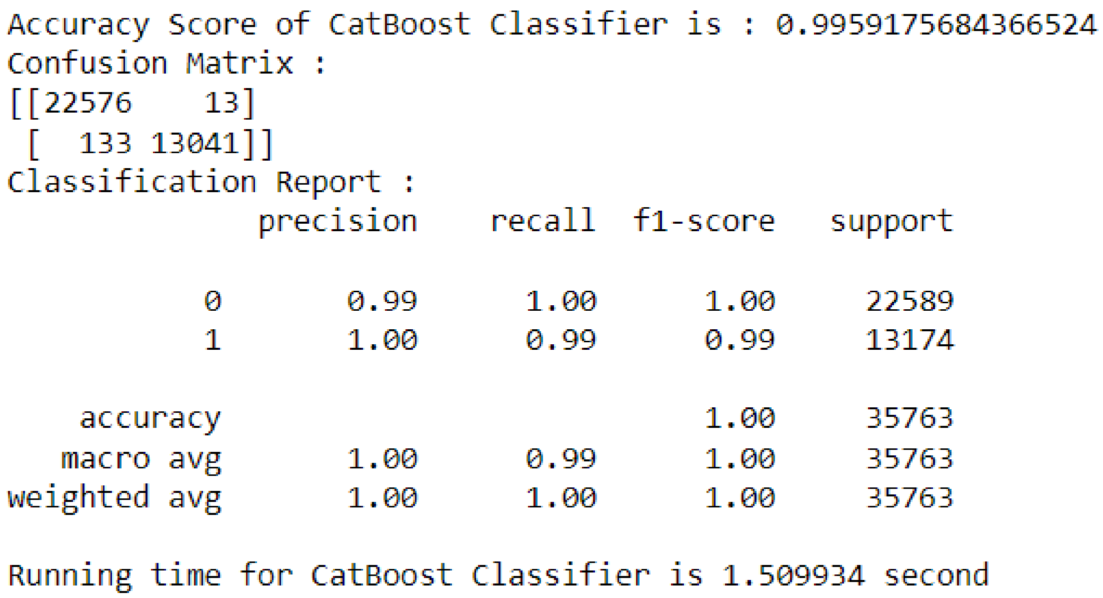
Folium: Interactive and used for dashboard building.

6 System Evaluation and Visualization

6.1 Analysis of Model Execution and Evaluation Results

Accuracy metrics may be used by different machine learning models to determine whether or not they are accurate. To give a better solution, here 11 models are created to assess the problem statement. All of the models that were employed more or less offered a respectable level of accuracy. Compared to the other, which had a 99.5% accuracy, CatBoost Model had a higher level of precision.

Figure 25

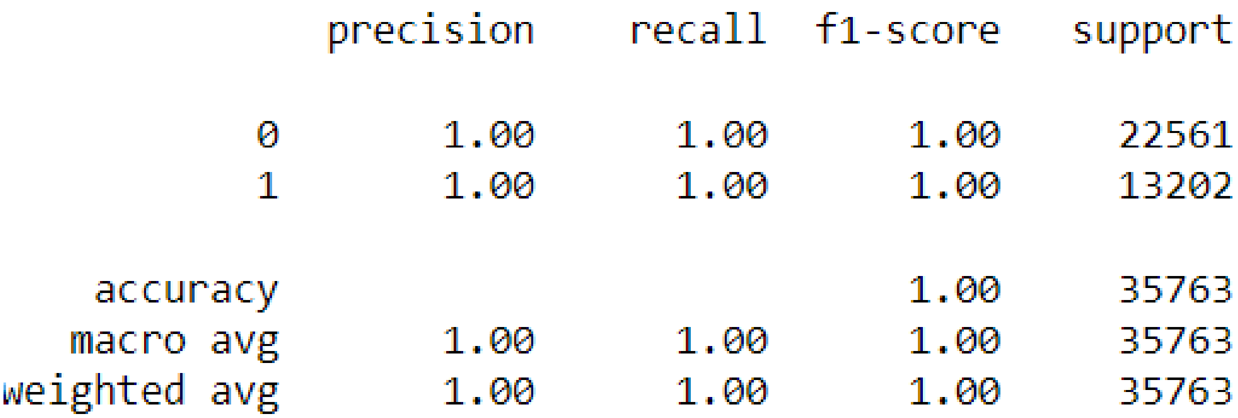


```
Accuracy Score of CatBoost Classifier is : 0.9959175684366524
Confusion Matrix :
[[22576    13]
 [  133 13041]]
Classification Report :
              precision    recall  f1-score   support

           0       0.99      1.00      1.00     22589
           1       1.00      0.99      0.99     13174

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

Running time for CatBoost Classifier is 1.509934 second
```

The F1 score was consequently raised to 100% by doing hyperparameter optimization for the CatBoost model using Optuna.

Figure 26

After hyperparameter tuning



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     22561
           1       1.00      1.00      1.00     13202

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

6.2 Achievements and Constraints

Achievements

For practical usage, we deployed our model using an EC2 instance so users (hotel managers and staff) can interact with the model by providing parameters to check if the customer's probability of cancellation is high or low.

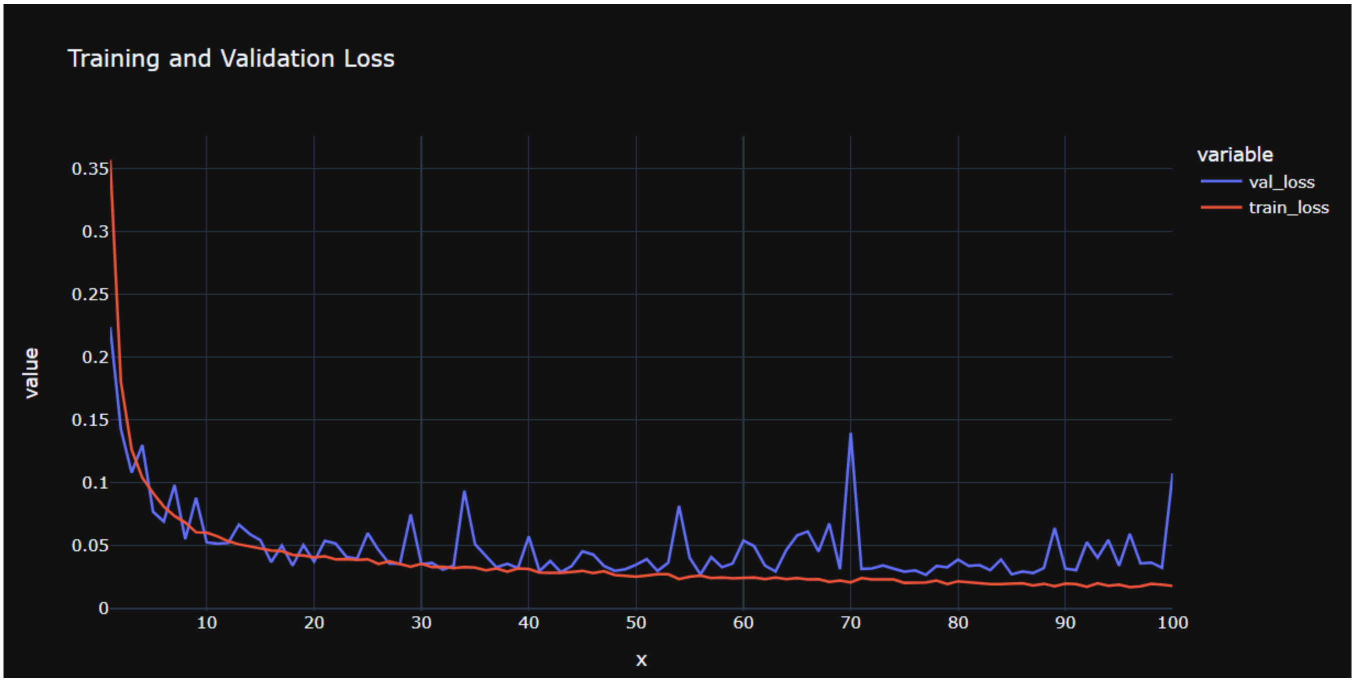
Constraints

At the production stage, the algorithms would take only specific inputs such as **“**Lead time”, “Total Number Of Special Requests”,” Required Car Parking Space”, “Booking Changes”, “Previous Cancellation”, and “Repeated Guests”. To get new parameters as input we need to retrain our model with new features (parameters).

6.3 System Quality Evaluation of Model Functions and Performance

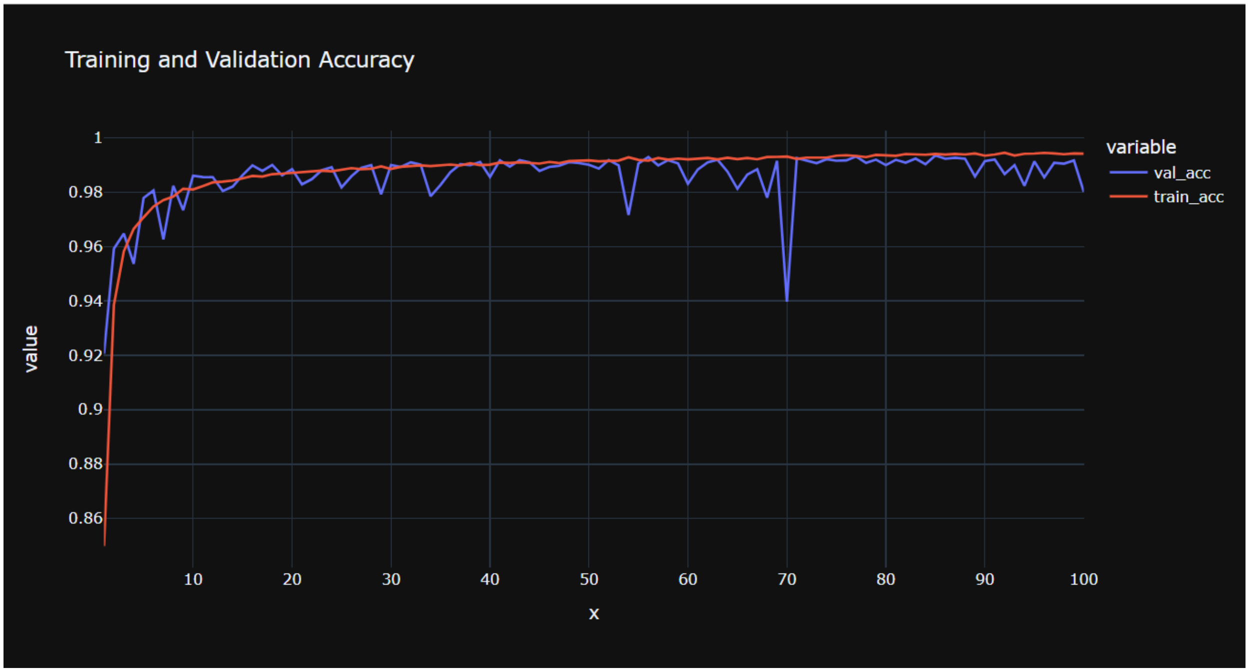
All the developed machine learning methods attain close to a 0.89 to 1 ratio. Each algorithm's execution time, however, varied uniquely. The training and validation loss and accuracy are shown in figure 28.

Figure 27



Validation loss became constant after 10 iterations and a significant shift in validation loss in the first 10 validations.

Figure 28

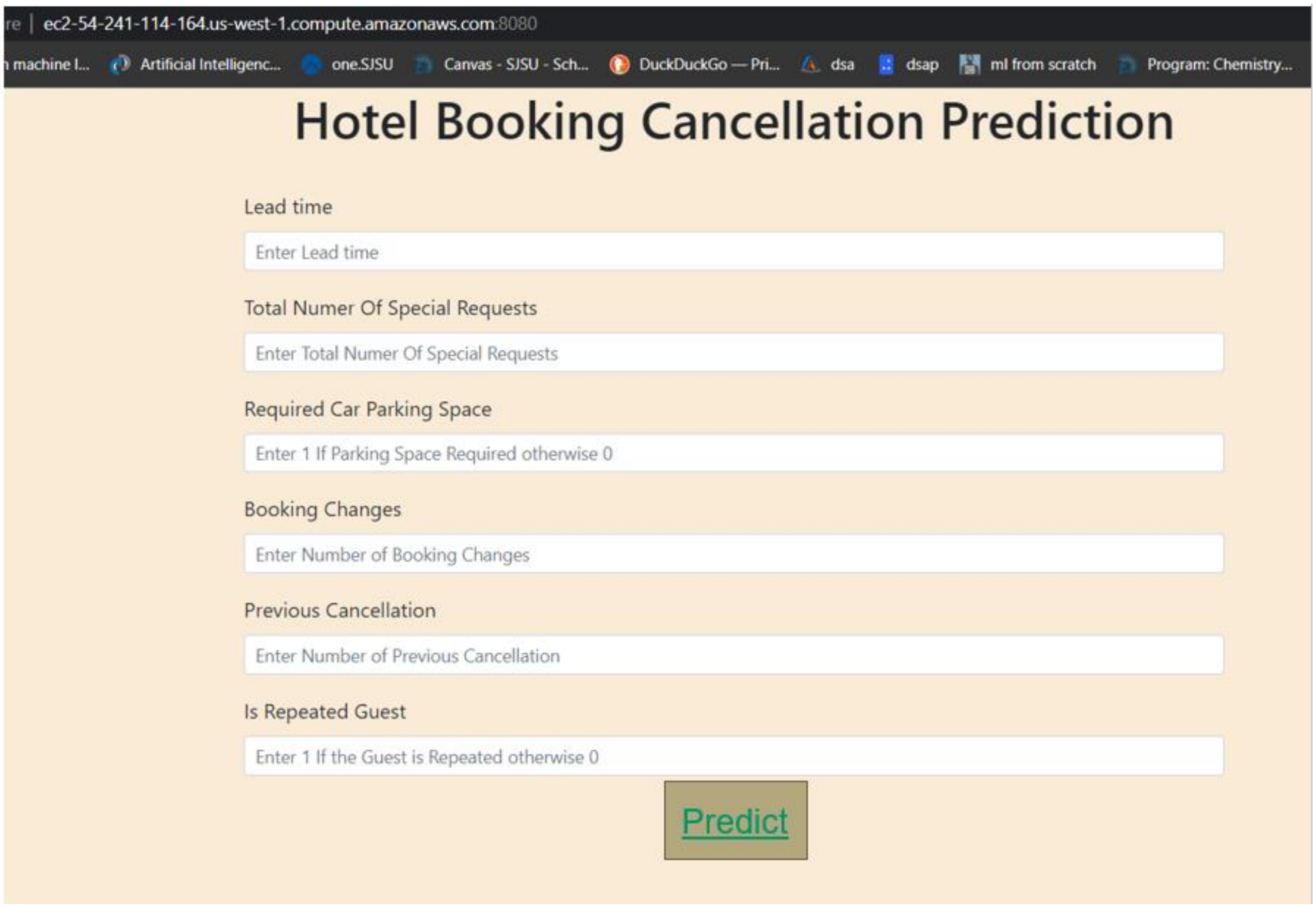


The accuracy of the validation was poor up to 10 iterations. And after 20 iterations, the models appear to have achieved a plateau. The training and validation accuracy was about 0.98 at this point.

6.4 System Visualizations

Here a static website has been created for predicting the Hotel booking cancellation in “ec2” which is shown in Figure 29. Here Lead time, Total Number Of Special Requests, Required Car Parking Space, Booking Changes, Previous Cancellations and Repeated Guests are taken for predicting the hotel booking cancellation. Lead time means when a customer booked a room, then that is the time difference between the reservation that customer has made and the actual arrival date. The total number of Special Requests means the customization requests from the customer to the hotel. Required Car Parking Space means the number of parking spaces that customers requested at the hotel. Booking changes mean the number of times a customer has been changed to his/her booking reservation. Previous Cancellation tells about how many times the customer cancelled their reservation after they have been booked in the past and Is Repeated Guest tells that is the guest reserved in the same hotel in the past or the new customer. Based on these input parameters, hotel booking cancellation rate (High LTV, Mid LTV or Low LTV) will be predicted

Figure 29



Hotel Booking Cancellation Prediction

Lead time

Enter Lead time

Total Numer Of Special Requests

Enter Total Numer Of Special Requests

Required Car Parking Space

Enter 1 If Parking Space Required otherwise 0

Booking Changes

Enter Number of Booking Changes

Previous Cancellation

Enter Number of Previous Cancellation

Is Repeated Guest

Enter 1 If the Guest is Repeated otherwise 0

7 Evaluation and Reflection

The major objective of this study was to develop a machine learning model-based prediction that may aid in foreseeing hotel reservation cancellations and able to develop several models for this project, including logistic regression, KNN, decision trees, Random Forest, XGBoost, and AdaBoost. The CatBoost model was tuned with Optional. To forecast the cancellation of the hotel reservation and also developed a static website for predictions.

7.1 Benefits

The CatBoost model has the best predicting ability which is determined. With the highest degree of accuracy, this model predicts whether or not a reservation will be cancelled. Therefore, using the catboost model to create prediction models for booking cancellations is a smart idea. As a consequence, this approach would enable hotels to estimate occupancy more precisely, manage their operations accordingly, and boost income. These algorithms not only enable hotel management to act on reservations that have been flagged as "perhaps going to be canceled," but also to provide more accurate demand estimates.

7.2 Experience and Lessons Learned

Model Evaluation Scores - Focused more on F1 score than the accuracy as true negative predictions might dominate the accuracy in a reverse direction.

Pickle - Joblib may also be utilized in Pickle cases. However, we opted for pickle because it loads and saves models more quickly than Joblib.

As part of a wider data model, we did normalization, which entails structuring data based on specified properties. Eliminating duplicate data, reducing data update mistakes, and streamlining the model training process are the major goals of database normalization.

Data encoding is maintained throughout test, train, and validation. This made sure that the encoding method we utilized was constant across our project.

7.3 Recommendations for Future Work

Unless other conditions are met, the lead time data implies it could be wise to limit reservations made far in advance because they have a tendency to cancel more frequently. However, this should be further looked into to see how near to the reservation date they were cancelled as if they also cancel well in advance, it could not have a big impact on the bottom line.

Comparing the outcomes to other hotels of a similar calibre is one approach to further enhance this attempt. This would reveal if these tendencies are general or exclusive to this hotel, and whether they are constant across the board. Finding these distinctive patterns or qualities might be crucial since doing so would level the playing field and concentrate on them can offer the hotel an advantage over rivals. Customer testimonials would be quite helpful in identifying these variances.

7.4 Contributions and Impacts on Society

The outcomes enable hotel managers to precisely estimate net demand, create better projections, enhance cancellation procedures, specify better overbooking techniques, and employ more forceful pricing and inventory management approaches.

References

Cho, V. (2003). A comparison of three different approaches to tourist arrival forecasting. *Tourism management*, *24*(3), 323-330.

Sánchez-Medina, A. J., & Eleazar, C. (2020). Using machine learning and big data for efficient forecasting of hotel booking cancellations. *International Journal of Hospitality Management*, *89*, 102546.

Adil, M., Ansari, M. F., Alahmadi, A., Wu, J. Z., & Chakrabortty, R. K. (2021). Solving the problem of class imbalance in the prediction of hotel cancellations: A hybridized machine learning approach. *Processes*, *9*(10), 1713.

https://ieeexplore.ieee.org/abstract/document/8260781?casa_token=3VMBccz3ToAAAAA:usMFNMfQOI3OcTRAeEnOt7C2MG6ba1QsjKvoYdce-DWPyYtYfRqL4K2Xz7mSXW0lKaZOAYT25l8

https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

https://static.javatpoint.com/tutorial/machine-learning/images/linear-regression-vs-logistic-regression.png

https://miro.medium.com/max/650/1*OyYyr9qY-w8RkaRh2TKo0w.png

https://upload.wikimedia.org/wikipedia/commons/7/76/Random_forest_diagram_complete.png

https://media.geeksforgeeks.org/wp-content/uploads/20200520035248/Leaf-Wise-Tree-Growth.png

https://www.researchgate.net/publication/341967355/figure/fig1/AS:901875410948097@1592035262515/Visual-Representation-of-Extra-Trees-Classifier.ppm

https://media.geeksforgeeks.org/wp-content/uploads/20210707140912/Bagging.png

https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/ANN-Graph.gif

https://static.javatpoint.com/tutorial/machine-learning/images/decision-tree-classification-algorithm.png

https://www.researchgate.net/profile/Lizawati-Mi Yusuf/publication/342743065/figure/fig2/AS:945640284639233@1602469621568/Overview-of-Work-Process-of-Grid-Search-with-Cross-Validation.png