Hotel Management System

Podeti Nithish Goud
podeti.n@northeastern.edu

Manasa Krishnan
krishnan.man@northeastern.edu

Anugu Varsha Reddy
anugu.v@northeastern.edu

Abstract

Database management systems are prevalent across various fields for displaying information about a system, querying user details, updating, creating and deleting user information. In this project, we use a system that provides information regarding different hotels and their management system. The system aims at creating, deleting and updating guest information on their database. It also provides an interactive user interface in the form of a dashboard that analyses the hotel information in the form of graphical statistical analysis. The system utilizes Python for visualisation and analysis using prediction and classification algorithms and MySQL workbench to create queries that can potentially make changes to the database. With this system, hotels can improve the robustness of their services by recognizing important patterns such as cancellation of reservations.

Category: Database Management System, Database Application, Data Visualisation, Supervised Machine Learning

Introduction

Hotel Management is one of the most growing enterprise handling processes in the world. Every enterprise dealing with hospitals, hotels, banks etc should have a lucid system or maintenance and support. Managing hotels and resorts need not only have a bright business mind but also a knack for the best method of marketing, providing the best service to the customers eventually leading to increased customer satisfaction.

This problem is interesting as the data mining techniques might result in a discovery that will optimize the entire management by correcting the flaws of delayed responses from the company to the customers, service to customers, transaction mechanisms, cancellation of stay by customers etc.

In this project, we deal with improving and providing robust services to guests of a hotel management system. This system provides an interactive session to see how the hotel manages its influx of visitors across the globe and analysis of the hotel's management. It also assesses how the hotel can improve its business by analyzing the reasons for guests cancelling their reservations by using prediction and classification algorithms.

The goal is to make hotel management and administration as simple as possible by providing automated hotel management analysis that allows for room bookings, cancellations, cash billing, room service, and total billing, among other things. It is possible to keep detailed records or information on an endless number of consumers. As a result, the hotel can better manage its operations and reduce paper work and labour.

Dataset

The database management system we have chosen for this project is Hotel Management System. The link to this dataset is https://www.kaggle.com/jessemostipak/hotel-booking-demand. This system gives information about the resort(hotel), customers who stay in the hotel, type of customers, number of customers, old and new customers, the room type, room reservation, and room cancellation information. We have a relational database consisting of 29 features(columns) and about 100,000 records(rows). It will be analysed in the form of a csv file.

Methodologies

E-R Diagram

Hotel Management system is the main class. The admin in the hotel makes the bookings class and has 1-to-N relationship with the booking class.

Guest class has the primary key, Customer_ID and other related information and is contained inside the hotel management system class. It has a total relationship with the main class. Guest makes a booking and has one-to-many relationship with booking class as one guest can make any number of bookings and changes. Guests will N-to-N relationship with the rooms class as many guests can book any number of rooms. Guest makes payments and so it has a 1-to-N relationship with transaction class. It can also be assumed that the class is n total participation with the transaction class. Guest is connected to Stay class where 1 customer can have any number of stays and thus give rise to the Guest_History class.

The booking class is connected to the room class and is assumed to have N-to-N relationship. The stay class is connected to the room class and is considered to be a weak entity and Room is the strong entity as all of its attributes have their values in room class and is extended from that.

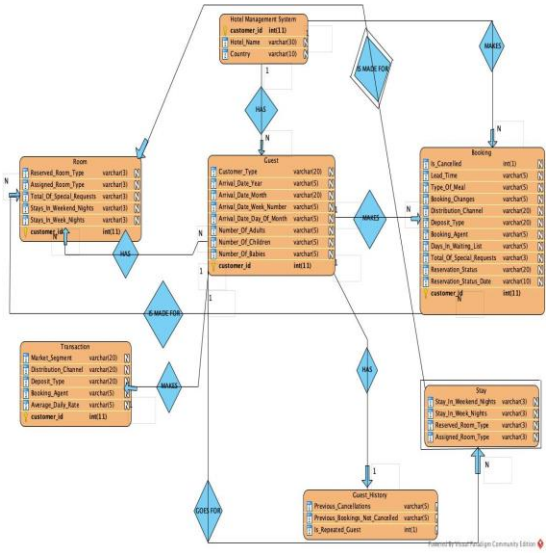

Figure 1. ERD and Schema Diagram for Hotel Management System

Database Design

The platform we used for the database design is MySQL Workbench and we created the following components in our database design:

Views

The five views that we created are regarding the total cancellations, cancellations done when an agent made a booking, total guests arrived, type of room and arrival day of the guest in that particular year.

Stored Procedures

The six stored procedures that we did in our project are regarding the assigned room type of the customer, customer's lead time taken for a booking, meal chosen by a customer, total number of bookings made by a particular agent, maximum customers arrived in a month and maximum lead time taken by a hotel to make a booking.

Functions

The four functions that we created are finding out the customer country, customer booking under an agent, total number of customers from a country, and total customers that arrived on a particular date of the year.

Transactions

The three transactions that we created are inserting new customers, finding the number of customers in each hotel till date and deleting the room and rolled back.

Triggers

The two triggers that we created are regarding the meal of the customer, when not given automatically takes breakfast and bed (BB) and the other one is cancellation which is directly proportional to reservation status for that particular booking.

Optimization

We also perform query optimization by limiting the result of any query to be 50 records and avoided null values in necessary columns. We have normalized tables so that the data will be non-redundant and used appropriate data types and also avoided unnecessary columns in the select clause for the SQL queries to run in seconds.

Application Design

The browser-specific components are created using HTML standards, and the dynamism of the system is achieved by focusing on Java Server Pages constructs. The components included in the application are buttons and labels for creating new guest records, deleting user information, updating user information and reading user description like meal type, lead time, cancellations etc.

Data Analysis

Data Cleaning

Data pre-processing is done to see if there any imbalances in the dataset. The null values are dropped and the missing values for both numerical and categorical attributes are handled by using Pipeline architecture as it reduces the number of sparse values too. OneHotEncoder of Python is used to replace all categorical content with numeric values which are suited better for performing analysis. Scaling of values is done for every feature in our dataset for better prediction and classification.

```
customer_ID                      1.186971e+09
Is_Canceled                      2.343480e-01
Lead_Time                        2.520964e+00
Arrival_Date_Year                4.992893e-01
Arrival_Date_Week_Number         1.842816e+02
Arrival_Date_Day_Of_Month        5.068079e-01
Stays_In_Weekend_Nights          9.891829e-01
Stays_In_Week_Nights             3.572044e+00
No_Of_Adults                     2.315691e-01
No_Of_Children                   1.597913e-01
No_Of_Babies                     3.800716e-03
Is_Repeated_Guest                2.708617e-02
Previous_Cancellations           7.204557e-01
Previous_Bookings_Not_Cancelled  2.091370e+00
Booking_Changes                  3.980688e-01
Booking_Agent                    3.496853e+00
Days_In_Waiting_List             5.076378e-01
Average_Daily_Rate               2.419944e+03
Total_Of_Special_Requests        1.931905e-01
total_staying_nights             2.214772e-01
Arrival_Date_Week_number         4.395448e-01
Stays_In_Weekend_Nights          2.511793e-01
dtype: float64
```

Figure 2. Normalized values

Exploratory Data Analysis

Statistical analysis of the data is performed to see an overview of the entire hotel's spread, customer information, room information, days the customers waited to get the reservation, their arrival date, special requests made by the customers and stay duration on weekdays and weekends.

Folium library and marker cluster algorithms are used to see how the hotels are spread across the globe by taking their latitudes and longitudes. Following this and using matplotlib library a pie chart is plotted to interpret the per cent of guests coming from different countries. Portugal and United Kingdom occupy the major per cent. Market Segment column gives information regarding the types of intermediaries for payments between the guests and the hotel's admin. A bar chart relating these two columns is plotted. More than other segments, groups and online TAs have a non-refund policy. By using the average daily rate column, every guest of the resort hotel pays about 48 euros (assumed to be euros as the major locations of the hotels are United Kingdom and Portugal) on an average per night for their stay. The number of reservations done for rooms B, C, G, H, L are comparatively lower than the others and the hotel should find a way to prevent this. A bar chart is also plotted using seaborn library for the number of reservations made in each and every month.

The correlation matrix for all the features is given as follows:

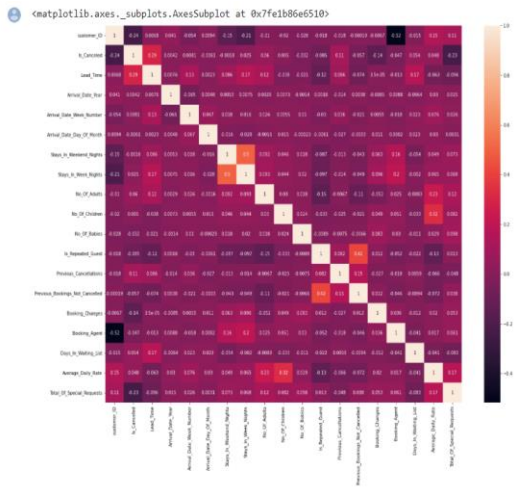

Figure 3. Correlation heatmap

Prediction of Booking Cancellations

The main reason for a hotel's business to go down can be verified using factors such as wait time of the customers, the kind of deposit they are asked for etc. Supervised Machine Learning algorithms provide an effective way to overcome this problem. 'Is_Canceled' column is taken as the response variable whereas other features are taken as the predictor variables. After finding the correlation between these two kinds of variables, it is understood that Reservation_Status, Deposit_Type, Lead_Time, Country and Total_Of_Special_Requests are the top 5 most influential factors on the response variable. We use Logistic Regression for prediction and 4 algorithms including Decision Tree, Gaussian Naïve Bayes, XGBoost and Random Forest for classification.

1. Prediction

Logistic Regression is used to predict the cancellation of customers. After pre-processing the

dataset by converting the object data types to int or float, the accuracy given by the model is just about 67 per cent which is very low. This is because, logistic regression fails to identify the randomness in the dataset and assumes linearity between the dependent and the independent variables. That is why we perform classification.

2. Classification

We perform binary classification of cancellations made by the customer and compare the results of 4 algorithms. Random Forest works the best with an accuracy of around 85 per cent. We also apply hyper parameter tuning in efforts to increase the accuracy rate and receive an accuracy of about 86 per cent using 300 estimators and 9 max_features. Random Forest works better as it captures the hidden randomness in our dataset.

Dashboard

Dashboard of Python provides an interactive user interface to retrieve information about an entity in the form of graphs, charts, dataframes and text. gridspec of matplotlib of Python is used here to create a dashboard. We create two dashboards, one to get the distribution of the predictors and another to know the spread of the values of each variable in the form of box plot. We toggle between different variables and see its distribution and perform individual interpretation.

Results and Discussions

Type	Name	Hyperparameters	Accuracy
Prediction	Logistic Regression	penalty = l2, C = 0.1	68.01
Classification	Random Forest	n_estimators = 300, max_features = 9	86.28
	Decision Tree	Default	81.48
	XGBoost	Default	81.27
	Gaussian NB	Default	60.7

Table 1. Accuracy values of the algorithms

We see that the random forest classification works better than the other models.

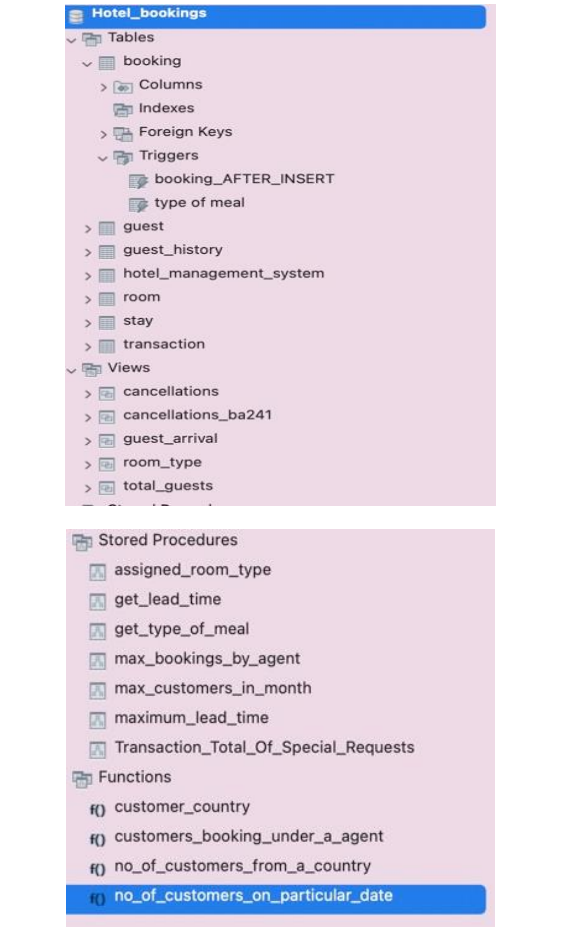

Figure 4. Components inside SQL

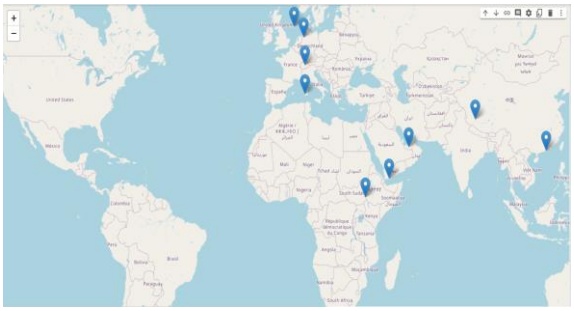

Figure 5. Geographical spread of the origin countries of the guests

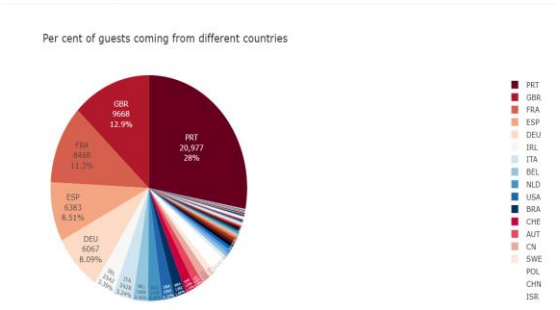

Figure 6. Percentage and number of guests from different countries

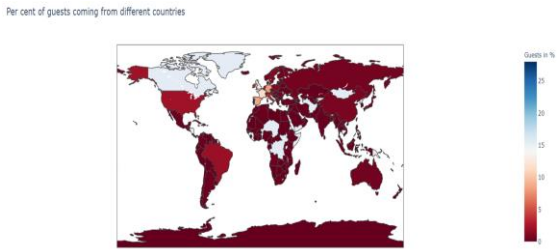

Figure 7. Choropleth of the above chart

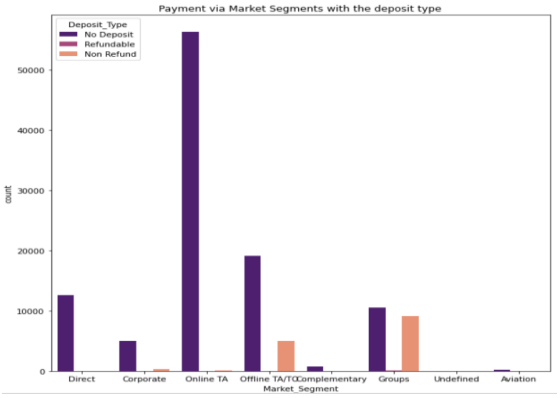

Figure 8. Distribution of market segments with the type of payment

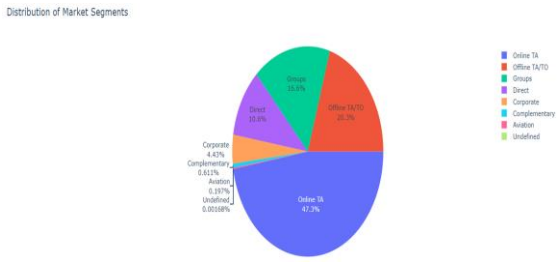

Figure 9. Different types of market segments

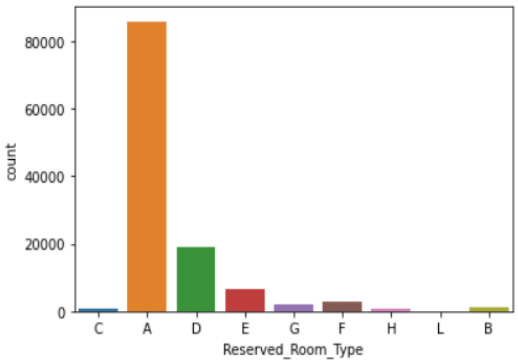

Figure 10. Distribution of reserved room types

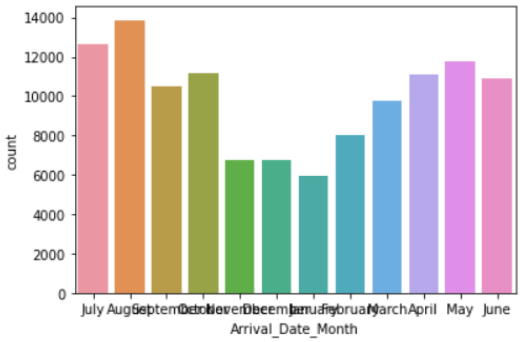

Figure 11. Distribution of arrival months

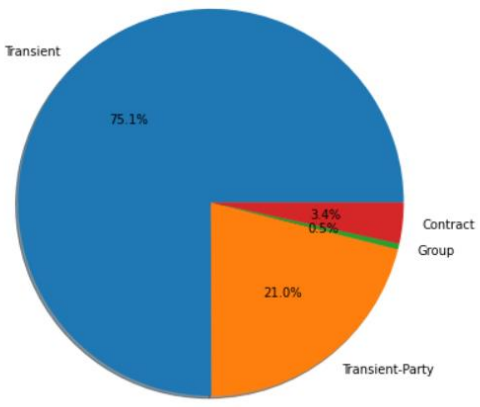

Figure 12. Pie chart for the type of customers

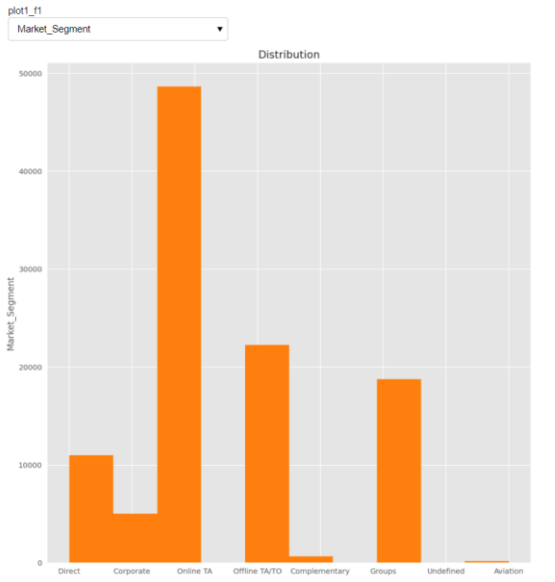

Figure 13. Dashboard for displaying the histograms of all features

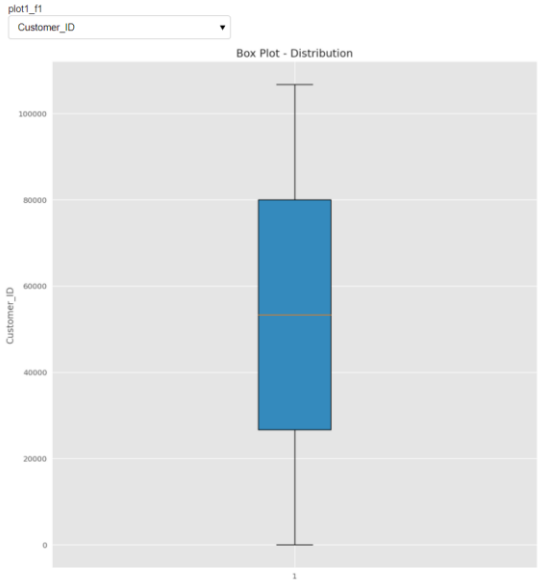

Figure 14. Dashboard for displaying the box plot of all features

Source Code

GitHub: https://github.com/PodetiNithish/Hotel-Management-system

Conclusion

The ideal outcome of this project (Hotel management system) is to show the visualization of the influx of visitors and guests to the resort hotel and city hotel via a dashboard that helps in the visualization of the distribution of features of the dataset and arrive at the best prediction and classification algorithms to find out whether a guest stays or not. Random Forest works the best for classifying the cancellation of customers. The statistical diagrammatic descriptions are also provided. This will in evidently improve the overall management of the system.

Future Works

While the system we have at hand is very efficient in terms of getting user information, the database of hotel management system tends to scale and thus we need a highly scalable machine to handle large datasets. We can then process data using Big Data algorithms like MapReduce and Big Data technologies like cloud storage, distributed and parallel systems and improve ACID properties of the systems. There might be a situation where we have to check for multicollinearity by which we have to apply Principal Component Analysis on the dataset for feature selection. If the dataset is skewed, it is necessary that we apply SMOTE or resampling. The user interface must be reliable enough and should satisfy CAP theorem and fault tolerance.