**Food Delivery App Data Analysis**

In this project, you will be working on a real-world dataset of zomato, one of the most used food ordering platforms. This project aims on cleaning the dataset, analyze the given dataset, and mining informational quality insights.

**Project Description**

This project will help you understand how a real-world database is analyzed using SQL, how to get maximum available insights from the dataset, pre-process the data using python for better upcoming performance, and how a structured query language helps us retrieve useful information from the database.

##### Step by Step Process of Live Project Execution

**Module-1 Data PreProcessing (Week-1)**

In this module, you will query the dataset using structured query language to gain insights from the database. The problem statements to be solved will be provided to you, and you will need to provide the solution for the same using your logic. Different concepts of SQL will be used in this process, such as aggregating the data, grouping the data, ordering the data, etc.

**Task: 1 Pre-Processing the data and Removing Unwanted Columns**

In this module, you will query the dataset using structured query language to gain insights from the database. The problem statements to be solved will be provided to you and you need to provide the solution for the same using your logic. Different concepts of SQL will be used in this process such as aggregating the data, grouping the data, ordering the data, etc. Module 1 consists of subtasks which are as follows

Pre-processing the data:

Data Pre-processing is one of the important steps in data analytics because data that is not processed can lead to different unwanted results when the data will be used for further applications. This task includes sub-tasks such as handling null values, deletion or transformation of irrelevant values, datatype transformation, removing duplicates, etc. The tasks to be performed for cleaning the data set are given below:

Removing Unwanted Columns:

Removing unwanted columns refers to the process of eliminating irrelevant or unnecessary columns from a dataset. This can improve data analysis and visualization by reducing clutter and focusing on the most important information. It involves identifying and selecting the columns to be removed and executing the removal process using tools like programming languages, database management systems, or spreadsheet software.

**import pandas as pd**

**import numpy as np**

**import re**

**import warnings**

**warnings.filterwarnings("ignore")**

**def read\_data\_from\_csv():**

**hotels=pd.read\_csv('zomato.csv')**

**return hotels**

**def remove\_unwanted\_columns():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call read\_data\_from\_csv() function to get dataframe**

**hotels=read\_data\_from\_csv()**

**hotels = hotels.drop(['address','phone'], axis =1)**

**return hotels**

**Task: 2 Renaming and selecting Columns in a dataset**

Renaming columns involves changing the names of one or more columns in a dataset to make them more meaningful or consistent. Selecting columns refers to the process of choosing only specific columns to be included in a dataset while excluding all others. These techniques are useful for improving the organization and readability of data and can help streamline data analysis. By renaming and selecting only the relevant columns, data scientists can create a more focused and manageable dataset that is better suited for their specific analysis needs.

Only these columns are allowed in the dataset:

1.    Id 2.    Name 3.    online\_order 4.    book\_table 5.    rating 6.    votes 7.    location 8.    rest\_type

9.    dish\_liked 10.    cuisines 11.    approx\_cost 12.    type

**def rename\_columns():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call remove\_unwanted\_columns() function to get dataframe**

**hotels = remove\_unwanted\_columns()**

**hotels.rename(columns ={'rate':'rating','approx\_cost(for two people)':'approx\_cost','listed\_in(type)':'type'},inplace = True)**

**#task2: rename columns, only these columns are allowed in the dataset**

**return hotels**

**Task: 3 Dealing with Null values in a dataset**

Handling null values refers to the process of identifying and addressing missing or incomplete data in each column of a dataset. This involves using techniques like imputation, where missing values are replaced with estimated values based on other data, or deletion, where incomplete records are removed entirely. Proper handling of null values is critical for accurate data analysis and can help prevent bias and errors in results.

**#task3: handle null values of each column**

**def null\_value\_check():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call rename\_columns() function to get dataframe**

**hotels=rename\_columns()**

**#deleting null values of name column**

**hotels = hotels.dropna(subset = ['name'])**

**#handling null values of online\_order**

**hotels['online\_order'].fillna("NA",inplace = True)**

**#handling null values of book\_table**

**hotels['book\_table'].fillna("NA",inplace = True)**

**#handling null values of rating**

**hotels['rating'].fillna(0,inplace = True)**

**#handling null values of votes**

**hotels['votes'].fillna(0,inplace = True)**

**#handling null values of location**

**hotels['location'].fillna("NA",inplace = True)**

**#handling null values of rest\_type**

**hotels['rest\_type'].fillna("NA",inplace = True)**

**#handling null values of dishliked**

**hotels['dish\_liked'].fillna("NA",inplace = True)**

**#handling null values of cuisines**

**hotels['cuisines'].fillna("NA",inplace = True)**

**#handling null values of approxcost**

**hotels['approx\_cost'].fillna(0,inplace = True)**

**#handling null values of type**

**hotels['type'].fillna("NA",inplace = True)**

**return hotels**

**Task: 4 Identifying Duplicate data in a dataset.**

Finding duplicates in a dataset refers to the process of identifying records that are identical or nearly identical to one another. Duplicate data can skew analysis results and waste computational resources, so it is important to identify and remove duplicates before analyzing data. This can be achieved using algorithms that compare records and identify common attributes, or through manual inspection of the dataset.

**#task4 #find duplicates in the dataset**

**def find\_duplicates():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call null\_value\_check() function to get dataframe**

**hotels=null\_value\_check()**

**hotels = hotels.drop\_duplicates(keep = "first")**

**#droping the duplicates value keeping the first**

**return hotels**

**Task: 5 Text Cleaning**

Text cleaning refers to the process of removing irrelevant or unnecessary text from all the columns in a dataset. This is an essential step in data preprocessing and analysis, as it ensures that the data is accurate and reliable. Text cleaning can involve tasks such as removing stopwords, punctuation, and special characters, as well as correcting spelling and grammar errors.

**#task5 removing irrelevant text from all the columns**

**def removing\_irrelevant\_text():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call find\_duplicates() function to get dataframe**

**hotels= find\_duplicates()**

**hotels=hotels[hotels['online\_order'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['book\_table'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['rest\_type'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['online\_order'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['rating'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['name'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['location'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['approx\_cost'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['type'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['votes'].str.contains('RATED|Rated')==False]**

**hotels=hotels[hotels['dish\_liked'].str.contains('RATED|Rated')==False]**

**return hotels**

**Task: 6 Unique Value check and Irrelevant value handling**

The process of examining each column in a dataset to identify and handle any irrelevant data, while also verifying the uniqueness of values within each column. This helps ensure data accuracy and integrity in analysis and decision-making.

**#task6: check for unique values in each column and handle the irrelevant values**

**def check\_for\_unique\_values():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call removing\_irrelevant\_text() function to get dataframe**

**hotels=removing\_irrelevant\_text()**

**hotels=hotels[hotels['online\_order'].str.contains('Yes|No')==True]**

**# for i in range(0,len(hotels)):**

**# hotels.iloc[i].rating = hotels.iloc[i].rating[:3]**

**hotels['rating'] = hotels['rating'].apply(lambda x : 0 if x == 'NEW' else x)**

**hotels['rating'] = hotels['rating'].apply(lambda x : 0 if x == '-' else x)**

**#hotels['rating'] = hotels['rating'].str.slice(0, 3)**

**# hotels['rating'] = hotels['rating'].apply(lambda str(x): x[:3])**

**hotels['rating'] = hotels.rating.str.split('/').str[0]**

**hotels['rating'].fillna(0,inplace = True)**

**return hotels**

**Task: 7 Cleaning and exporting Zomato Dataset**

The process of cleaning the Zomato dataset by removing any unknown or unidentifiable characters and exporting the cleaned dataset to a new file named **"zomatocleaned.csv"**. This involves identifying and removing any symbols, special characters, or non-standard characters that may interfere with proper data analysis. By exporting the cleaned dataset to a new file, the original dataset can be preserved and the cleaned data can be easily accessed for further analysis and decision-making.

**#task7: remove the unknown character from the dataset and export it to "zomatocleaned.csv"**

**def remove\_the\_unknown\_character():**

**#DO NOT REMOVE FOLLOWING LINE**

**#call check\_for\_unique\_values() function to get dataframe**

**dataframe=check\_for\_unique\_values()**

**# task7:to remove the unknown charachter from name column:**

**# replace the string having regex [Ãx][^A-Za-z]+ with ''**

**dataframe['name'] = dataframe['name'].replace(to\_replace = "[Ãx][^A-Za-z]+" ,value="",regex=True)**

**#remove unknown character from dataset**

**#export cleaned Dataset to newcsv file named "zomatocleaned.csv"**

**dataframe.to\_csv('zomatocleaned\_v1.csv')**

**return dataframe**

**#check if mysql table is created using "zomatocleaned.csv"**

**#Use this final dataset and upload it on the provided database for performing analysis in MySQL**

**#To Run this task first Run the appliation for Terminal to create table named 'Zomato' and then run test.**

**def start():**

**remove\_the\_unknown\_character()**

**def task\_runner():**

**start()**

**Task: 8 Data Cleaning and Analysis with Excel and MySQL**

The process of performing data cleaning and analysis using both Excel and MySQL. The first step involves the deduplication of identical rows in an exported dataset using Excel. This helps ensure data accuracy and consistency. The second step involves uploading the cleaned dataset to a provided database and performing further analysis using MySQL. This allows for powerful data querying and visualization, enabling data-driven decision-making.

**Step1:** Download the exported data set **"zomatocleaned.csv"**

**Step 2:** Eliminate any identical rows in the Excel dataset by performing the deduplication task.

**Step 3:** Upload the final dataset to the database provided to conduct an analysis in MySQL.

**Step 4:** Retrieve the database host, username, password, and database information from the**"Database Info"** tab and input it into db.py.

**Step 5:** To complete the current task, click on **"Run Test"** and confirm that your table has been successfully created.

**Module: 2 Data Analysis using SQL(week – 2)**

In this module, you will be working on performing data analysis on the pre-processed data from the previous module and conducting Data Analysis using SQL. You will generate queries for given problem statements.

**Task : 1 For a high-level overview of the hotels, provide us the top 5 most voted hotels in the delivery category.**

**select distinct name,votes,rating**

**from b6175983.zomato**

**where type = 'delivery'**

**order by votes desc limit 5**

**Task : 2 The rating of a hotel is a key identifier in determining a restaurant’s performance. Hence for a particular location called Banashankari find out the top 5 highly rated hotels in the delivery category.**

**select name,rating, location, type**

**from b6175983.zomato**

**where location = 'Banashankari'**

**and type = 'delivery'**

**order by rating desc limit 5;**

**Task : 3 Compare the ratings of the cheapest and most expensive hotels in Indiranagar.**

**select (select rating from b6175983.zomato where location = 'Indiranagar' order by approx\_cost desc limit 1) as rating1,**

**(select rating from b6175983.zomato where location = 'Indiranagar' order by approx\_cost asc limit 1) as rating2**

**from b6175983.zomato**

**limit 1;**

**Task : 4 Online ordering of food has exponentially increased over time. Compare the total votes of restaurants that provide online ordering services and those that don’t provide online ordering services.**

**select**

**sum(votes) as total\_votes,online\_order**

**from b6175983.zomato**

**group by online\_order**

**Task : 5 Number of votes defines how much the customers are involved with the service provided by the restaurants For each Restaurant type, find out the number of restaurants, total votes, and average rating. Display the data with the highest votes on the top( if the first row of output is NA display the remaining rows).**

**select type,count(id) as number\_of\_restaurants,sum(votes) as total\_votes,avg(rating) as avg\_rating**

**from b6175983.zomato**

**group by type**

**having type NOT IN('NA')**

**order by total\_votes desc**

**Task : 6 What is the most liked dish of the most-voted restaurant on Zomato(as the restaurant has a tie-up with Zomato, the restaurant compulsorily provides online ordering and delivery facilities.**

**select name,dish\_liked,rating,votes from zomato**

**where type = "Delivery" and online\_order = "Yes"**

**order by rating desc,votes desc**

**limit 1**

**Task : 7 To increase the maximum profit, Zomato is in need to expand its business. For doing so Zomato wants the list of the top 15 restaurants which have min 150 votes, have a rating greater than 3, and is currently not providing online ordering. Display the restaurants with the highest votes on the top.**

**select name,rating,votes,online\_order**

**from zomato**

**where rating>3 and votes>150 and online\_order="No"**

**order by votes desc limit 15;**