SHRI RAMSWAROOP MEMORIAL COLLEGE OF ENGINEERING AND MANAGEMENT, LUCKNOW, UTTAR PRADESH

MASTER OF COMPUTER APPLICATION
2024-25
(Odd Semester)
PROJECT

SHRI RAMSWAROOP MEMORIAL COLLEGE OF ENGINEERING AND MANAGEMENT, LUCKNOW, UTTAR PRADESH

MASTER OF COMPUTER APPLICATION
2024-25
(Odd Semester)
PROJECT

Zomato Data Analysis Project: Unlocking Insights

Welcome to our exploration of Zomato data! We'll dive deep into analyzing sales, customer behavior, and competition to uncover valuable insights for driving growth and improving strategies.

Presented By:

- Vaibhav Kamal

[redacted]

Introduction

Deepinder Goyal, CEO

This project focuses on analyzing Zomato’ s extensive dataset to uncover meaningful insights related to customer behavior, restaurant performance, and delivery trends. By applying data analysis techniques, the goal is to enhance user experience, improve operational efficiency, and inform strategic decision-making.

Through exploratory data analysis (EDA), visualization, and predictive modeling, the project identifies key patterns that can drive business growth. Whether optimizing delivery times, improving customer satisfaction, or identifying market opportunities, this analysis provides actionable insights to help Zomato and its partners stay competitive in the food delivery industry.

Objectives of the Project

1

Analyze Sales and Performance

By analyzing sales data and identifying patterns, trends, and challenges, the project seeks to optimize business strategies and improve Zomato's overall performance.

2

Understand Customer Behavior

The project explores customer demographics, ordering habits, and loyalty, providing insights into customer engagement and satisfaction.

3

Competitor Analysis

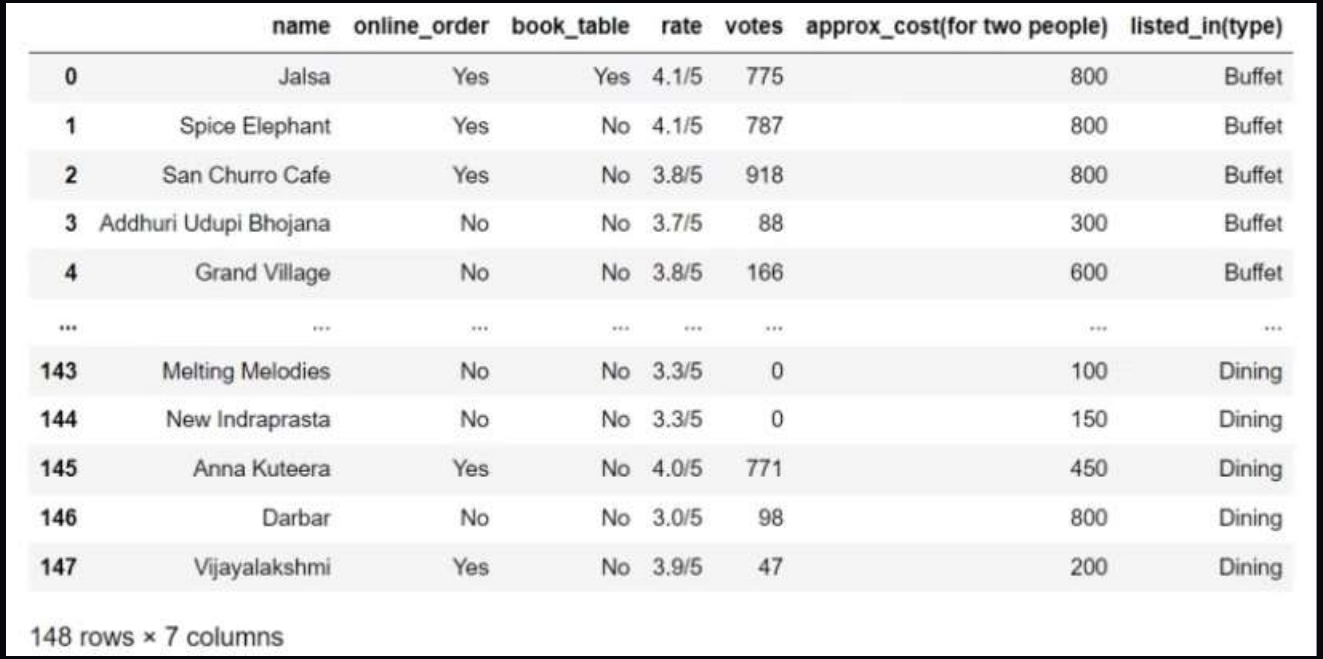
Comparing Zomato's platform with competitors such as Swiggy and Uber Eats to understand market position, pricing strategies, and customer preferences.

4

Restaurant Popularity

Identifying which restaurants perform well based on order volumes, customer ratings, and reviews.

[redacted]



	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1/5	775	800	Buffet
1	Spice Elephant	Yes	No	4.1/5	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8/5	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	300	Buffet
4	Grand Village	No	No	3.8/5	166	600	Buffet
...	...	...	...	...	...	...	...
143	Melting Melodies	No	No	3.3/5	0	100	Dining
144	New Indraprasta	No	No	3.3/5	0	150	Dining
145	Anna Kuteera	Yes	No	4.0/5	771	450	Dining
146	Darbar	No	No	3.0/5	98	800	Dining
147	Vijayalakshmi	Yes	No	3.9/5	47	200	Dining

148 rows × 7 columns

Dataset of Zomato

First step is to upload the dataset of Zomato API, (named Zomato_data.csv) into the Jupyter Notebook.

Uploading a dataset into Jupyter Notebook is a simple process that allows for efficient data analysis. You can either load data directly from local files (like CSV, Excel) using libraries such as **Pandas**, or from online sources using URLs. For example, `pd.read_csv()` is commonly used to read CSV files. Ensuring that the dataset is clean and properly formatted before analysis is key to a smooth workflow in Jupyter.

Libraries Used

1

import pandas as pd

The command `import pandas as pd` is used to import the **Pandas** library in Python, assigning it the alias `pd` for convenience. Pandas is a powerful library for data manipulation and analysis, allowing you to work efficiently with structured data, such as DataFrames.

2

import numpy as np

The command `import numpy as np` imports the **NumPy** library in Python and assigns it the alias `np`. NumPy is essential for numerical computing, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

3

import matplotlib.pyplot as plt

The command `import matplotlib.pyplot as plt` imports the **Matplotlib** library's `pyplot` module, which is commonly used for creating visualizations in Python. By assigning it the alias `plt`, it simplifies the process of generating plots, such as charts, and histograms

4

import seaborn as sns

The command `import seaborn as sns` imports the **Seaborn** library in Python and assigns it the alias `sns`. Seaborn is built on top of Matplotlib and provides a high-level interface for creating heatmaps, violin plots, and pair plots.

[redacted]

Creation of a DataFrame

A **DataFrame** is a two-dimensional, tabular data structure provided by the **Pandas** library. It is similar to a spreadsheet or SQL table, where data is organized into rows and columns. Here, it is used to help Jupyter Notebook to fetch the CSV file of Zomato_data.

Following are some of the steps to perform operations:

1. ### Create a variable:

   Create a variable named "dataframe"

2. ### Create the alias pd:

   ```
   dataframe = pd.read_csv()
   ```

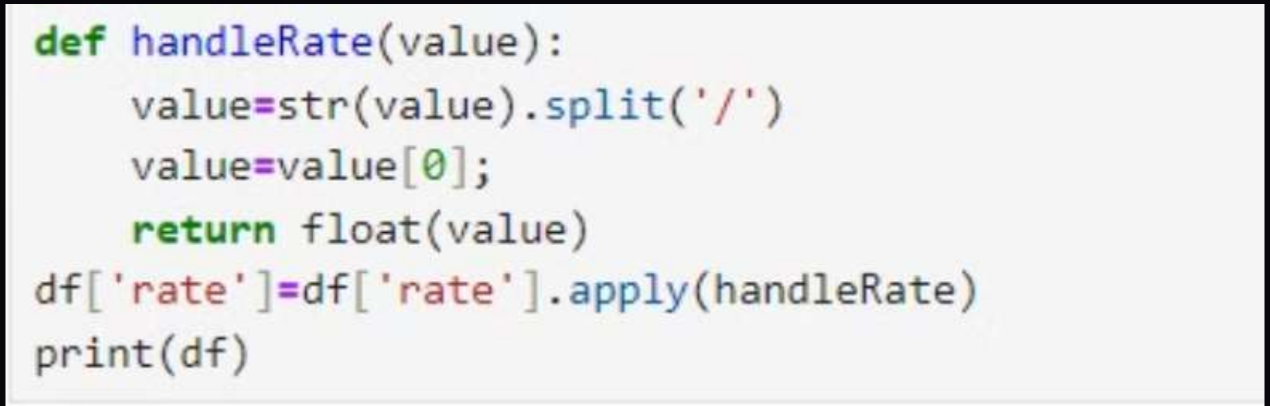
3. ### Assigning file name:

   ```
   dataframe = pd.read_csv("Zomato_data.csv")
   ```

4. ### Print the dataframe:

   ```
   dataframe = pd.read_csv("Zomato_data.csv")
   print(dataframe)
   ```

Data Cleaning & Data Preprocessing



```
def handleRate(value):
    value=str(value).split('/')
    value=value[0];
    return float(value)
df['rate']=df['rate'].apply(handleRate)
print(df)
```

Convert the data type of column - rate:

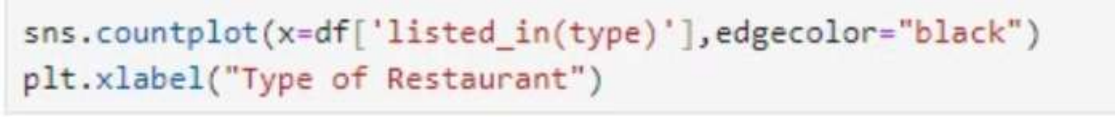
The given code defines a function `handleRate` to clean and process the `rate` column in a DataFrame.

- `handleRate(value)`: This function takes a value (likely in the form of a string) from the `rate` column, splits it by the `/` character, and retrieves the first part (before the `/`). It then converts this value to a float.
- `df['rate'] = df['rate'].apply(handleRate)`: The `apply()` function is used to apply the `handleRate` function to each entry in the `rate` column, transforming it into a numeric value.
- `print(df.head())`: This prints the first 5 rows of the modified DataFrame to show the updated `rate` column.

This approach is useful for cleaning and converting string representations of ratings or scores into numeric values for further analysis.

What type of restaurant do the majority of customers order from?

Input:



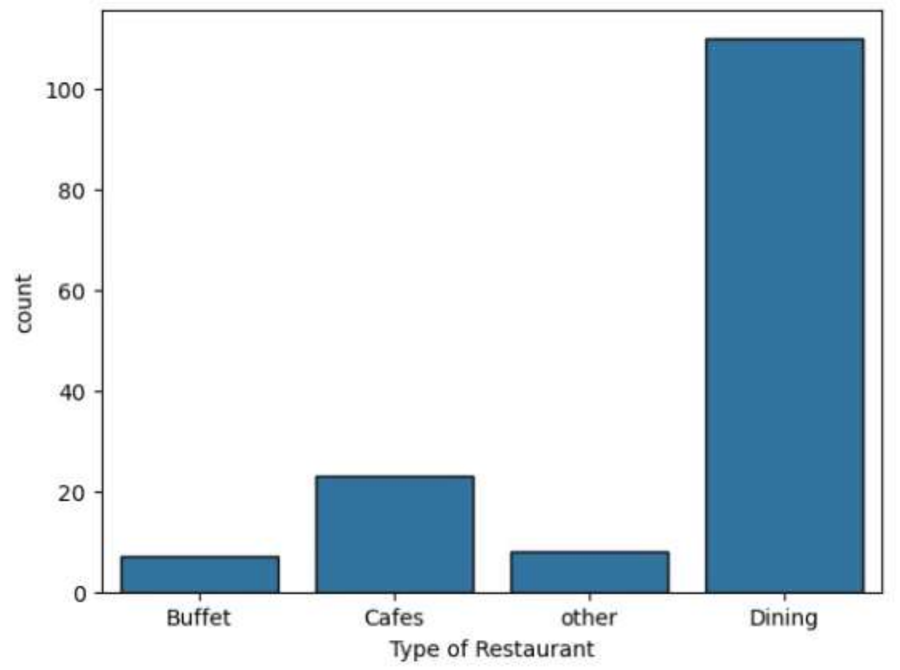
```
sns.countplot(x=df['listed_in(type)'],edgecolor="black")
plt.xlabel("Type of Restaurant")
```

The code uses **Seaborn** and **Matplotlib** to visualize the distribution of restaurant types from a dataset.

- `sns.countplot()`: This creates a count plot, showing the number of occurrences for each category in the `listed_in(type)` column of the DataFrame. It counts how many times each restaurant type appears in the data.
- `plt.xlabel("Type of restaurant")`: This sets the label for the x-axis of the plot to "Type of restaurant," making the chart more readable.

This visualization helps identify the frequency distribution of various restaurant types in the dataset.

Output:



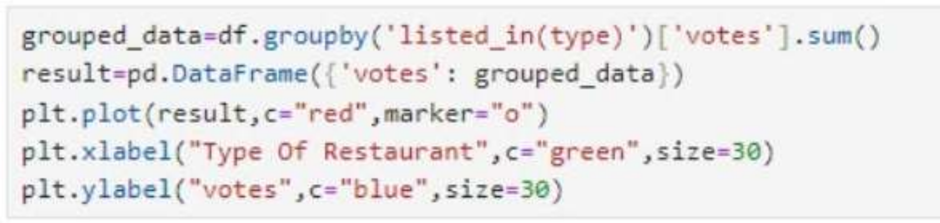
Conclusion:

The majority of the restaurants fall into the dining category.

Dining restaurants are preferred by a larger number of individuals.

How many votes has each type of restaurant received from customers?

Input:



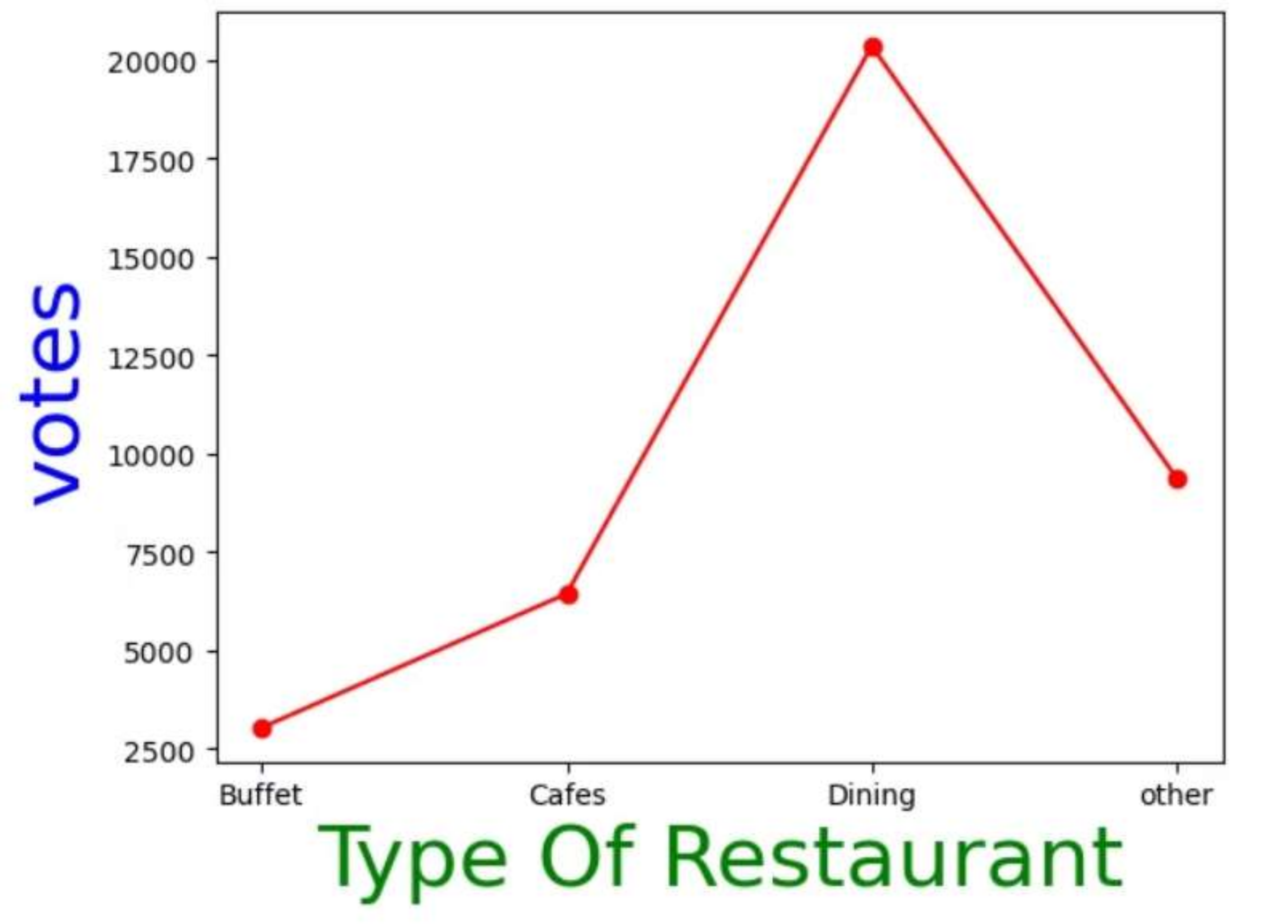
```
grouped_data=df.groupby('listed_in(type)')['votes'].sum()
result=pd.DataFrame({'votes': grouped_data})
plt.plot(result,c="red",marker="o")
plt.xlabel("Type Of Restaurant",c="green",size=30)
plt.ylabel("votes",c="blue",size=30)
```

This code groups the data by restaurant type and sums the votes for each type, then plots the results.

- `grouped_data = df.groupby('listed_in(type)')['votes'].sum()`: Groups the data by the `listed_in(type)` column (restaurant types) and sums up the
- `result = pd.DataFrame({'votes': grouped_data})`: Converts the grouped data into a new DataFrame with the column `votes`.
- `plt.plot(result, c="red", marker="o")`: Plots the summed votes for each restaurant type, with green markers shaped as circles.
- `plt.xlabel("Type of restaurant", c="green", size=30)` and `plt.ylabel("Votes", c="blue", size=30)`: Labels the x-axis as "Type of restaurant" and the y-axis as "Votes", both in red with a font size of 20.

This code visualizes the relationship between restaurant types and the total votes they received.

Output:

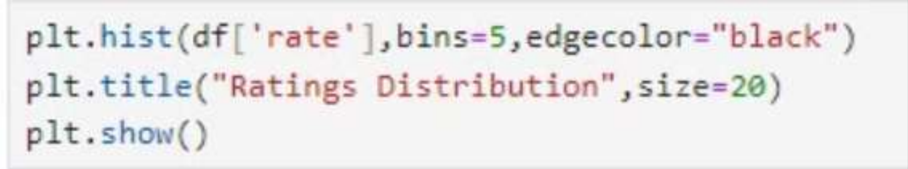


Conclusion:

The majority of restaurants received ratings.

What are the ratings that the majority of restaurants have received?

Input:



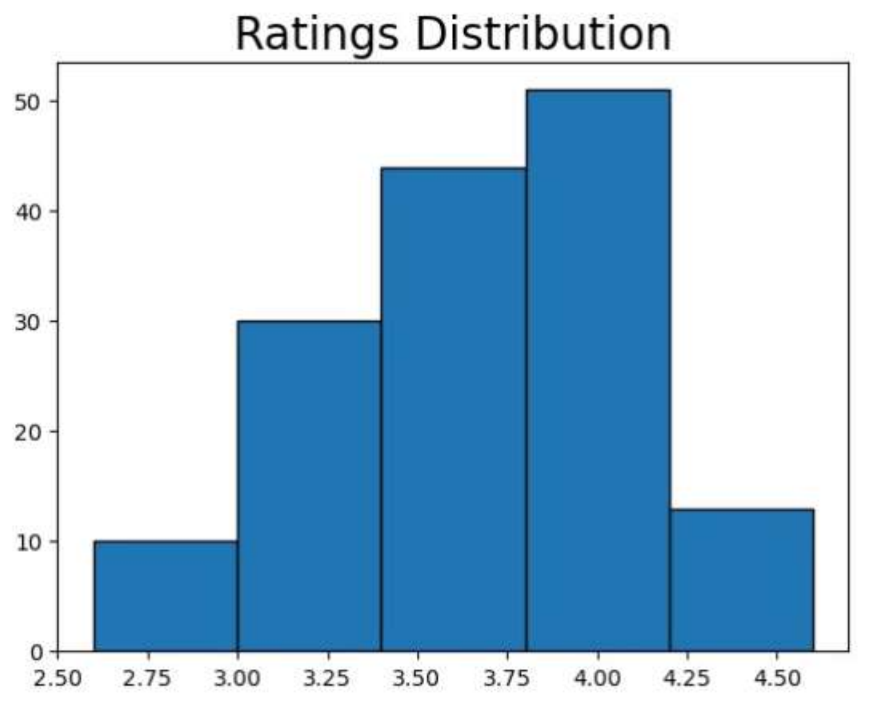
```
plt.hist(df['rate'],bins=5,edgecolor="black")
plt.title("Ratings Distribution",size=20)
plt.show()
```

This code creates a histogram to visualize the distribution of ratings from the `rate` column.

- `plt.hist(df['rate'], bins=5)`: Plots a histogram of the `rate` values in the dataframe with 5 bins, which divides the
- `plt.title("Ratings Distribution")`: Adds a title "Ratings Distribution" to the plot.
- `plt.show()`: Displays the histogram.

The code is used to visualize how ratings are distributed across the dataset by grouping them into 5 intervals.

Output:

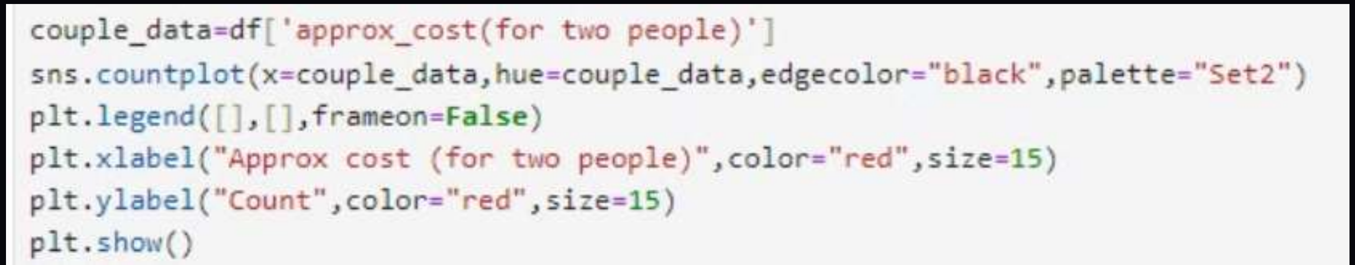


Conclusion:

The majority of restaurants received ratings ranging from 3.5 to 4.

Zomato has observed that most couples order most of their food online. What is their average spending on each order?

Input:



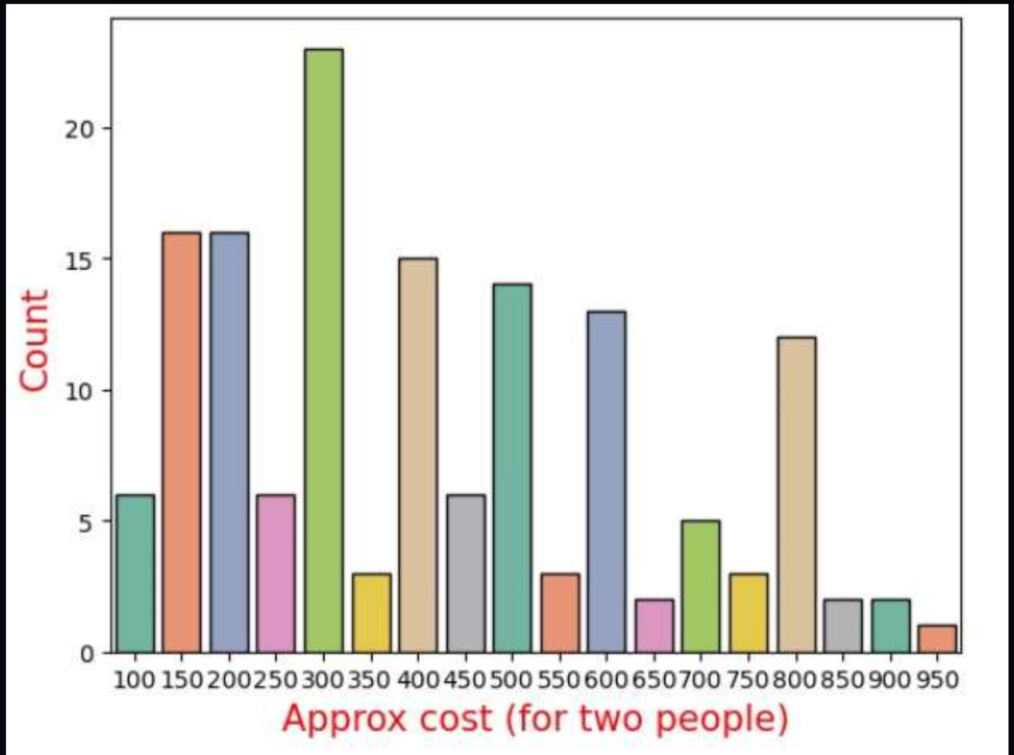
```
couple_data=df['approx_cost(for two people)']
sns.countplot(x=couple_data,hue=couple_data,edgecolor="black",palette="Set2")
plt.legend([],[],frameon=False)
plt.xlabel("Approx cost (for two people)",color="red",size=15)
plt.ylabel("Count",color="red",size=15)
plt.show()
```

This code is used to create a count plot of the approximate cost for two people from the given dataset:

- `couple_data = df['approx_cost(for two people)']`: Extracts the column labeled `'approx_cost(for two people)'** from the dataframe and assigns it to the`
- `sns.countplot(x=couple_data)`: Creates a Seaborn count plot that shows the frequency distribution of the different cost values in `couple_data`.

This visualization helps analyze how frequently certain cost ranges appear in the dataset.

Output:

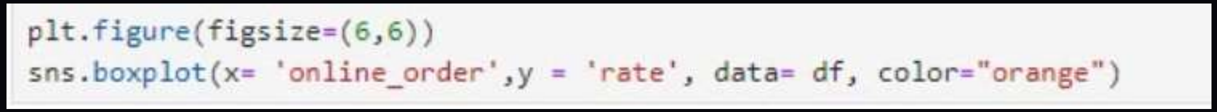


Conclusion:

The majority of couples prefer restaurants with an approximate cost of Rs.300.

Which mode (online or offline) has received the maximum rating?

Input:



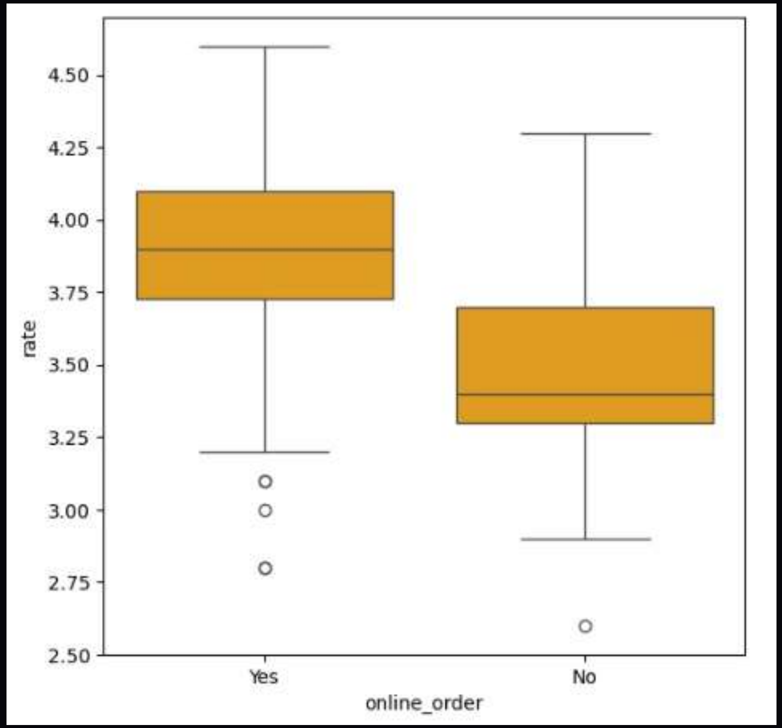
```
plt.figure(figsize=(6,6))
sns.boxplot(x= 'online_order',y = 'rate', data= df, color="orange")
```

The code generates a boxplot using the Seaborn library to visualize the relationship between two variables from a DataFrame:

- `x='online_order'`: This represents the categorical variable for the x-axis, indicating whether an online order option is available.
- `y='rate'`: This represents the numerical variable for the y-axis, showing the ratings or scores.
- `data=df`: Refers to the dataset being used for the plot.
- `figsize=(6,6)`: This sets the size of the plot to 6x6 inches.

The boxplot helps to visualize the distribution of ratings (`rate`) across different categories of online ordering options (`online_order`).

Output:



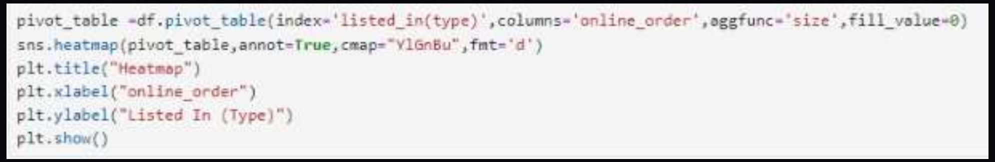
Conclusion:

Offline orders received lower ratings in comparison to online orders, which obtained excellent ratings.

While online orders receive higher ratings than offline orders.

Which type of restaurant received more offline orders, so that Zomato can provide those customers with some good offers?

Input:



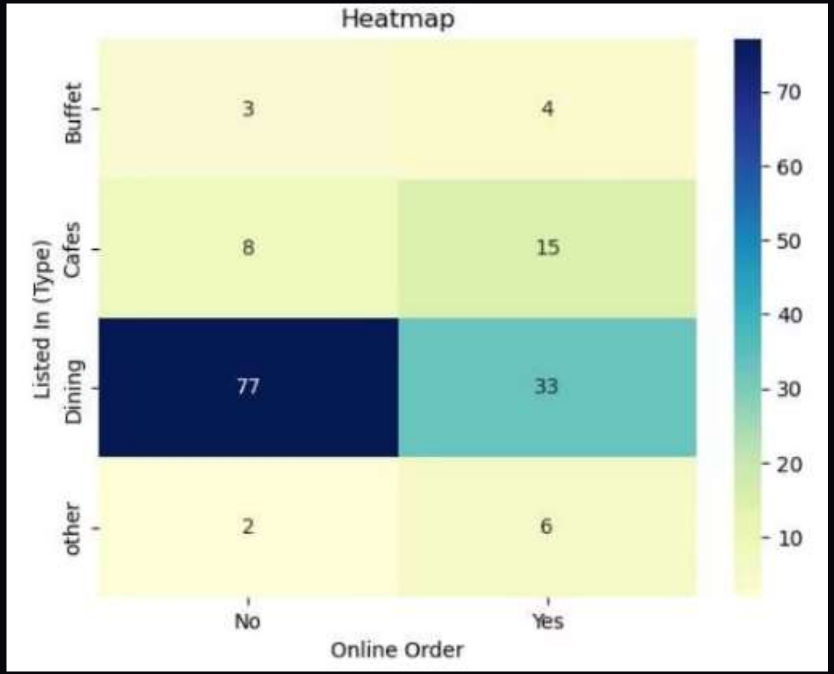
```
pivot_table =df.pivot_table(index='listed_in(type)',columns='online_order',aggfunc='size',fill_value=0)
sns.heatmap(pivot_table,annot=True,cmap="YlGnBu",fmt='d')
plt.title("Heatmap")
plt.xlabel("online_order")
plt.ylabel("Listed In (Type)")
plt.show()
```

The given code generates a heatmap to visualize the frequency of occurrences between two categorical variables using a pivot table:

- `pivot_table`: A pivot table is created from the DataFrame with `listed_in(type)` as the index and `online_order` as the columns.
- `sns.heatmap`: A Seaborn heatmap is plotted with the pivot table data.
  - `annot=True`: Annotates each cell with the corresponding value from the pivot table.
  - `cmap="YlGnBu"`: A color map (yellow-green-blue) is used to represent the data.
  - `fmt='d'`: Specifies that the values are formatted as integers.
- `plt.title`, `plt.xlabel`, `plt.ylabel`: These set the title and axis labels for the heatmap.

The heatmap provides a visual representation of the relationship between the two categorical variables (`online_order` and `listed_in(type)`) based on their frequency.

Output:



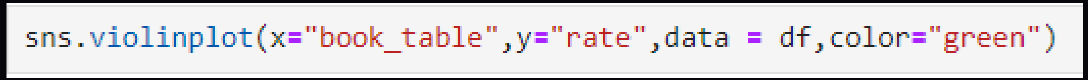
Conclusion:

Dining restaurants primarily accept offline orders, whereas cafes primarily receive online orders.

This suggests to clients prefer to place order in person at restaurants, but prefer online ordering at cafes.

Do restaurants that allow table booking have higher ratings compared to those that don’t?

Input:



```
sns.violinplot(x="book_table",y="rate",data = df,color="green")
```

The code you provided uses the `seaborn` library to create a violin plot. Here's a breakdown:

- **sns.violinplot**: This function generates a violin plot, which is a combination of a box plot and a kernel density plot, used to visualize the distribution of numerical data.

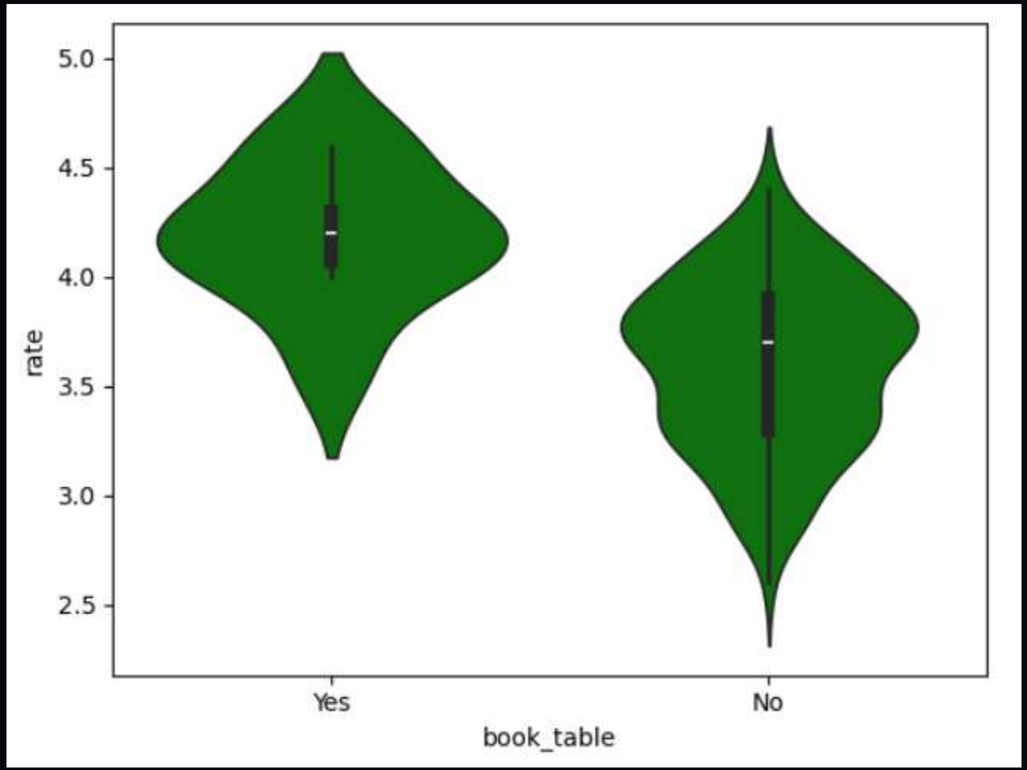
- **x="book_table"**: The `x` axis represents the `book_table` variable (likely a categorical variable from your DataFrame `df`).

- **y="rate"**: The `y` axis represents the `rate` variable, which is likely a numerical variable from the DataFrame.

- **data=df**: Specifies that the data is coming from the DataFrame `df`.
- **color="green"**: The color of the violin plot is set to green.

In summary, this code generates a green violin plot that shows the distribution of the "rate" values based on the "book_table" categories from the DataFrame `df`.

Output:

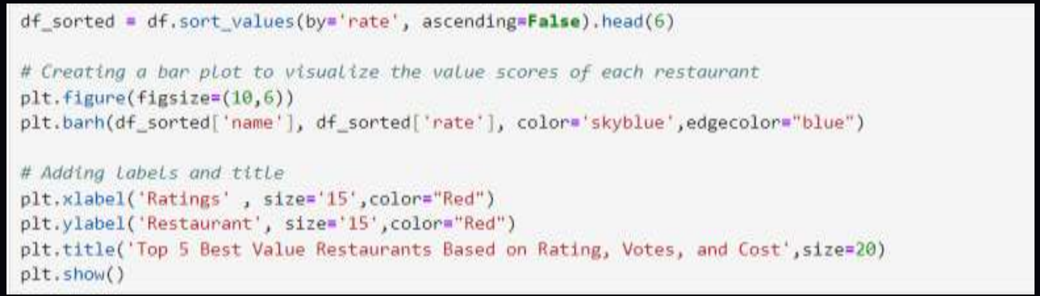


Conclusion:

Restaurants which have booking table option have slightly higher ratings than those restaurants which does not have booking table options.

Which are the top 5 restaurants that offers the best value based on the combination of rating, votes and cost for 2 peoples?

Input:



```
df_sorted = df.sort_values(by='rate', ascending=False).head(6)

# Creating a bar plot to visualize the value scores of each restaurant
plt.figure(figsize=(10,6))
plt.barh(df_sorted['name'], df_sorted['rate'], color='skyblue',edgecolor="blue")

# Adding labels and title
plt.xlabel('Ratings' , size='15',color="Red")
plt.ylabel('Restaurant', size='15',color="Red")
plt.title('Top 5 Best Value Restaurants Based on Rating, Votes, and Cost',size=20)
plt.show()
```

The code provided creates a bar plot to visualize the top 6 restaurants based on their ratings. Here's a breakdown:

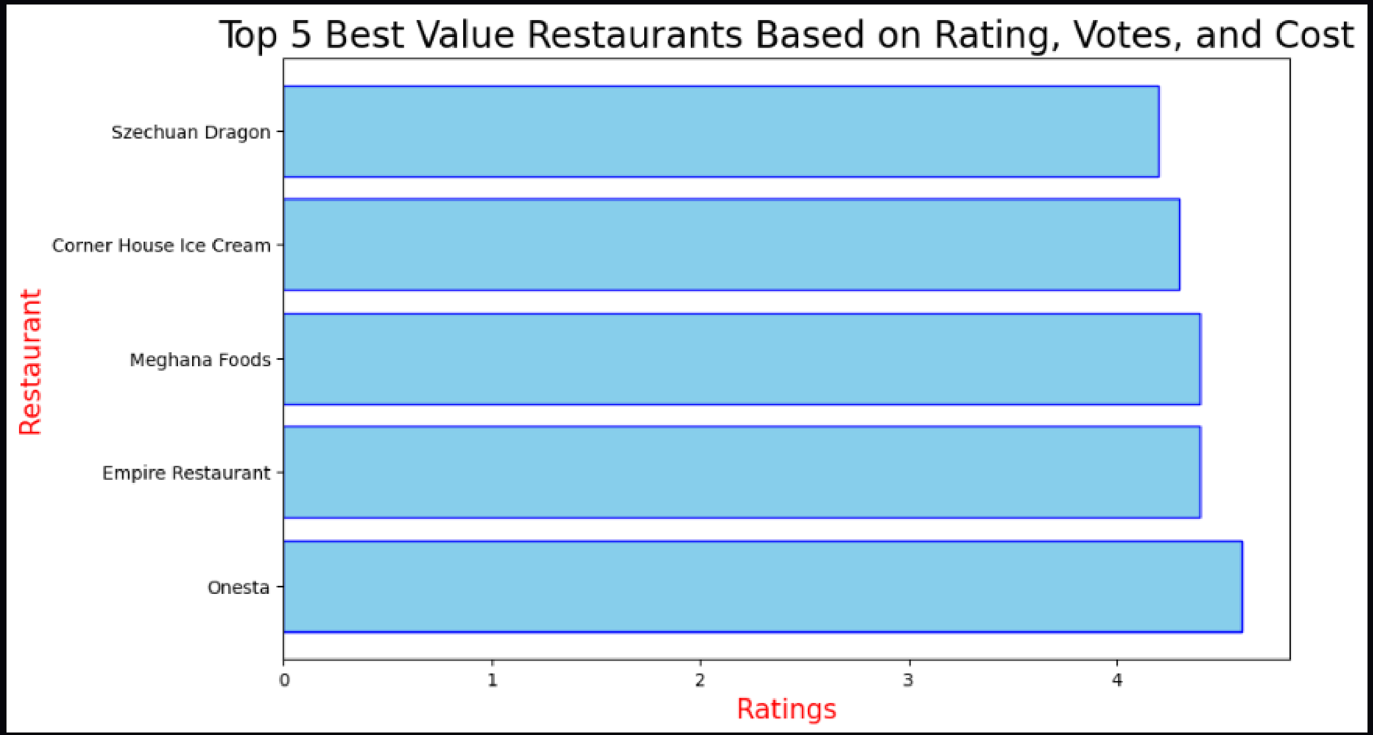
1. **df_sorted = df.sort_values(by='rate', ascending=False).head(6)**:
   - This line sorts the DataFrame `df` by the 'rate' column in descending order (highest to lowest).
2. **plt.figure(figsize=(10,6))**:
   - Defines the figure size for the plot, with dimensions 10x6 inches.

3. **plt.barh(df_sorted['name'], df_sorted['rate'], color='skyblue', edgecolor='blue')**:
   - Creates a horizontal bar

4. **plt.xlabel('Ratings', size=15, color="Red")** and **plt.ylabel('Restaurant', size=15, color="Red")**:
   - These lines add red-colored axis labels with a font size of 15.

5. **plt.title('Top 5 Best Value Restaurants Based on Rating, Votes, and Cost', size=20)**:
   - Adds a title to the plot with a font size of 20.

Output:

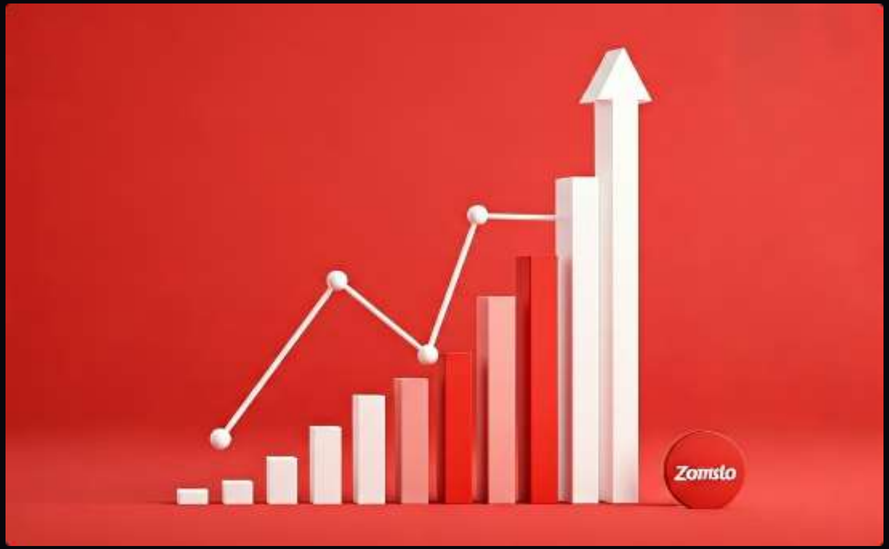


Conclusion:

Onesta, Empire Restaurant, Meghana Foods, Corner House Ice Cream and Szechuan Dragon are the top 5 restaurants.

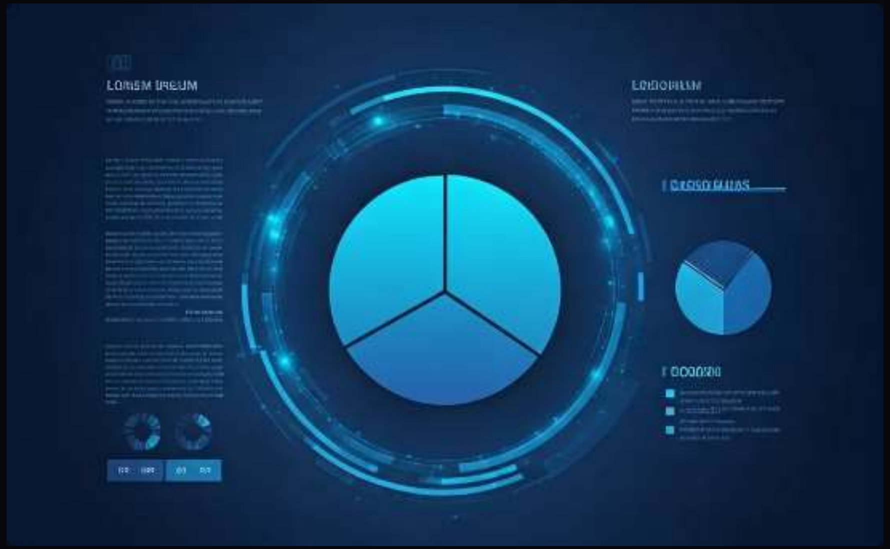
Applications :

1) Exploratory Data Analysis



Sales Trends

We analyzed sales data over time, identifying patterns and seasonality, and pinpointing peak periods and potential challenges. This helps us understand the overall performance of the platform.



Customer Demographics

We explored customer characteristics like age, location, and ordering habits. This provides valuable information for targeted marketing and personalized experiences.

Restaurant Popularity

We identified popular restaurants based on order volume, ratings, and customer reviews. This helps us understand which restaurants are performing well and why.

[redacted]

2) Customer Behavior and Trends

1

Order Frequency

We analyzed how often customers order from Zomato, identifying frequent users and potential areas for engagement.

2

Order Value

We explored factors influencing order value, such as customer demographics, promotions, and restaurant choice.

3

Customer Loyalty

We analyzed customer retention rates, identifying factors contributing to repeat orders and customer satisfaction.

4

Customer Feedback

We analyzed customer reviews and feedback, identifying areas for improvement and opportunities for personalized service.

Advantages

Improved Sales Insights

Analyzing sales data reveals trends and peak periods, optimizing operations and efficiency.

Customer Behavior Understanding

Gaining insights into demographics, ordering habits, and spending patterns allows for targeted marketing and personalized experiences.

Restaurant Performance Insights

The project identifies popular restaurants based on order volume, customer reviews, and ratings.

Enhanced Customer Experience

Personalized recommendations, targeted marketing campaigns, and loyalty programs improve the customer experience and increase retention.

Disadvantages

1

Data Quality Issues

Accurate insights depend on high-quality data. Incompleteness or inconsistencies can lead to flawed conclusions.

2

Limited Generalizability

Insights might be specific to Zomato's context and may not universally apply without further validation.

3

Scalability of Models

Models might need frequent retraining as business or customer behavior evolves.

4

Ethical and Privacy Concerns

Data analysis involving customer behavior requires careful attention to privacy and secure handling of sensitive information.

5

Bias in Recommendations

Machine learning models for recommendations can develop biases, impacting fairness and inclusivity.

[redacted]

Real-World Applications

Zomato's data analysis has numerous applications in the food and restaurant industry. These insights can optimize restaurant operations, improve customer experience, and drive business growth.

Zomato's data can also be used to improve the efficiency of delivery operations. For example, data analysis can help optimize delivery routes, reduce delivery times, and minimize delivery costs.

Data can also be used to improve the accuracy of restaurant information and ensure that the platform is up-to-date. This helps customers make informed decisions and increases the credibility of the platform.

Finally, Zomato's data analysis can help the company develop new features and services that enhance the user experience. For example, Zomato can use data to create personalized recommendations, develop new payment options, and launch new delivery services.

Future Scope

1

Personalized Recommendations

The project can enhance the platform’s ability to recommend restaurants or dishes to users based on their past orders, preferences, and real-time data. Machine learning algorithms can offer dynamic, personalized suggestions, increasing user engagement and satisfaction.

2

Consumer Behaviour Analysis

Through deeper analysis of consumer behavior, Zomato can forecast future trends, such as popular cuisines or peak ordering times. Understanding these behaviors will allow Zomato to offer more targeted promotions and campaigns, as well as improve restaurant performance metrics.

3

Geospatial Analysis

Zomato can utilize geospatial data to analyse customer locations and restaurant densities. This would help optimize delivery zones, identify underserved areas, and open opportunities for strategic restaurant partnerships or marketing efforts based on geographic trends.

4

Integration with IoT and Smart Devices

As IoT devices become more common, Zomato can explore integrations with smart devices like voice assistants or smart refrigerators. This would allow customers to order food through voice commands or receive notifications when a favorite restaurant is offering a deal.

5

Customer Experience Enhancement

By leveraging data on customer feedback, ordering patterns, and preferences, the project can contribute to improving overall customer experience. Features like quicker delivery, better customer service, and personalized discounts could be tailored to individual users or groups.

Conclusion and Next Steps

In conclusion, the development of a Zomato Database System for a MCA project underscores the practical application of computer application principles in addressing real-world business challenges. This project highlights the integration of machine learning, database management, software engineering, and user interface design, crucial in creating a system that enhances efficiency, accuracy, and security.

By analyzing Zomato data, we gained valuable insights into sales performance, customer behavior, and competitive dynamics. These findings inform our recommendations for improving strategies and driving growth.

[redacted]