Zomato Data Analysis Using Python

The following libraries were used to analyse Zomato's Database.

Numpy– with Numpy arrays, complex computations are executed quickly, and large calculations are handled efficiently.

Matplotlib– It has a wide range of features for creating high-quality plots, charts, histograms, scatter plots, and more.

Pandas– The library simplifies the loading of data frames into 2D arrays and provides functions for performing multiple analysis tasks in a single operation.

Seaborn– It offers a high-level interface for creating visually appealing and informative statistical graphics.

Database used: Zomato_data

1. **IMPORT REQUIRED LIBRARIES FOR ANALYSIS:**
2. **IMPORT DATASET TO JUPYTER NOTEBOOK:**

SYNTAX:

```
# Lets import the required libraries for analysis
Import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# The dataframe is named as Zomato
Zomato = pd.read_csv("Zomato_data.csv")
Zomato.head()
```

OUTPUT:

	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

3. **The datatype of rate is converted to float and the denominator is eliminated**.

SYNTAX:

```
# let's convert the data type of the "rate" column to float and remove the denominator.
def handleRate(value):
    value = str(value).split('/')
    value = value[0];
    return float(value)
```

```
Zomato['rate'] = Zomato['rate'].apply(handleRate)
print(Zomato.head())
```

OUTPUT:

```
                  name online_order book_table  rate  votes  \
0               Jalsa          Yes        Yes   4.1    775
1        Spice Elephant          Yes         No   4.1    787
2       San Churro Cafe          Yes         No   3.8    918
3  Addhuri Udupi Bhojana           No         No   3.7     88
4          Grand Village           No         No   3.8    166

   approx_cost(for two people) listed_in(type)
0                          800          Buffet
1                          800          Buffet
2                          800          Buffet
3                          300          Buffet
4                          600          Buffet
```

4. UNDERSTAND THE SUMMARY OF THE DATAFRAME:

```
Zomato.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   name                         148 non-null    object
 1   online_order                 148 non-null    object
 2   book_table                   148 non-null    object
 3   rate                         148 non-null    float64
 4   votes                        148 non-null    int64
 5   approx_cost(for two people)  148 non-null    int64
 6   listed_in(type)              148 non-null    object
dtypes: float64(1), int64(2), object(4)
memory usage: 8.2+ KB
```

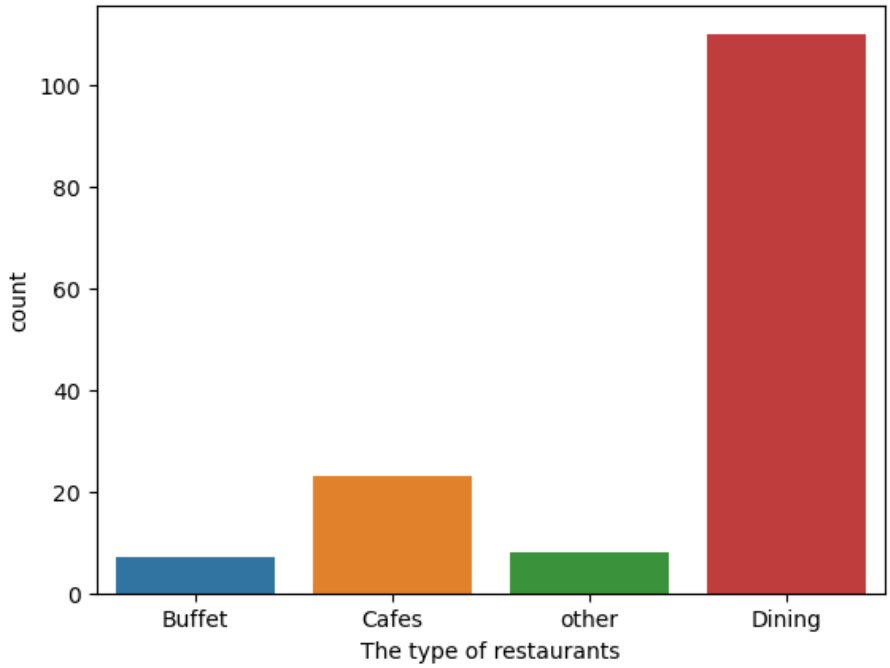
5. FOCUS ON THE TYPE OF DINING THAT THE RESTAURANT HOLDS :

SYNTAX:

```
# Let's analyse the type of restaurants present in the dataset:
sns.countplot(x= Zomato['listed_in(type)'])
plt.xlabel("The type of restaurants")
```

OUTPUT:

```
Text(0.5, 0, 'The type of restaurants')
```



<u>On observation, it is evident that – Most of the restaurants offer dining compared to Buffet and Cafes.</u>

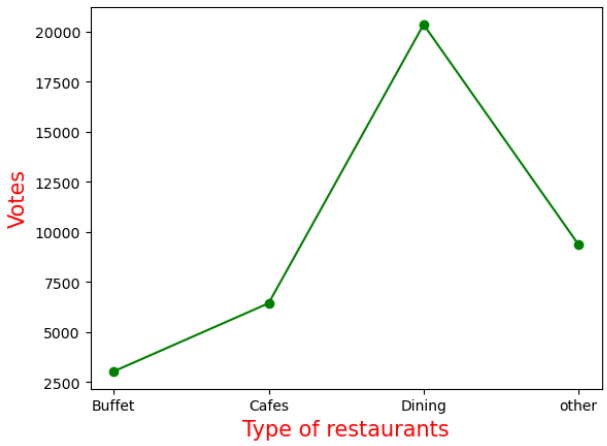
6. **TYPE OF RESTAURANTS PREFERRED BY INDIVIDUALS:**

SYNTAX:

```
grouped_data = Zomato.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes': grouped_data})
plt.plot(result, c= "green", marker = "o")
plt.xlabel("Type of restaurants", c="red", size = 15)
plt.ylabel("Votes", c = "red", size = 15)
```

OUTPUT:

```
22]: Text(0, 0.5, 'Votes')
```



- On observation, the dining restaurants are most preferred by individuals.

7. DETERMINE THE RESTAURANT THAT RECEIVED THE MAXIMUM NUMBER OF VOTES?

SYNTAX:

```
max_votes = Zomato['votes'].max()
restaurant_with_max_votes = Zomato.loc[Zomato['votes'] == max_votes, 'name']
print("Restaurants with the maximum votes:")
print(restaurant_with_max_votes)
```

OUTPUT:

```
Restaurants with the maximum votes:
38    Empire Restaurant
Name: name, dtype: object
```

8. EXPLORE THE COLUMN – ONLINE ORDER

SYNTAX:

```
sns.countplot(x=Zomato['online_order'])
plt.title("Restaurants with online order")
```

OUTPUT:

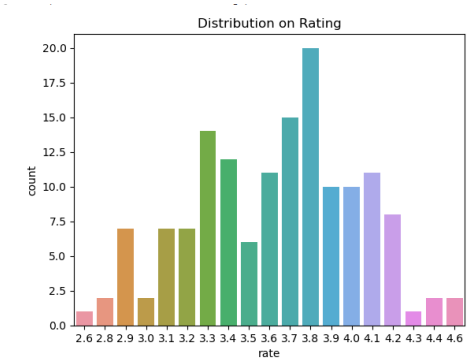
- Majority of the restaurants does not accept online orders.

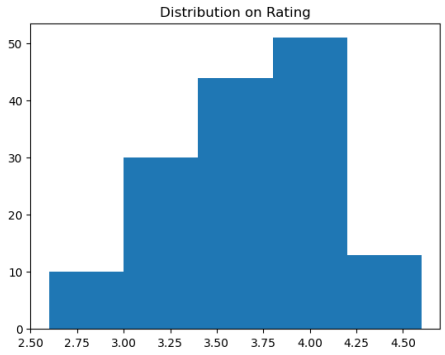
9. EXPLORE COLUMN – RATE:

SYNTAX:

```
plt.hist(Zomato['rate'], bins = 5)
plt.title("Distribution on Rating")
plt.show()
```

OUTPUT:





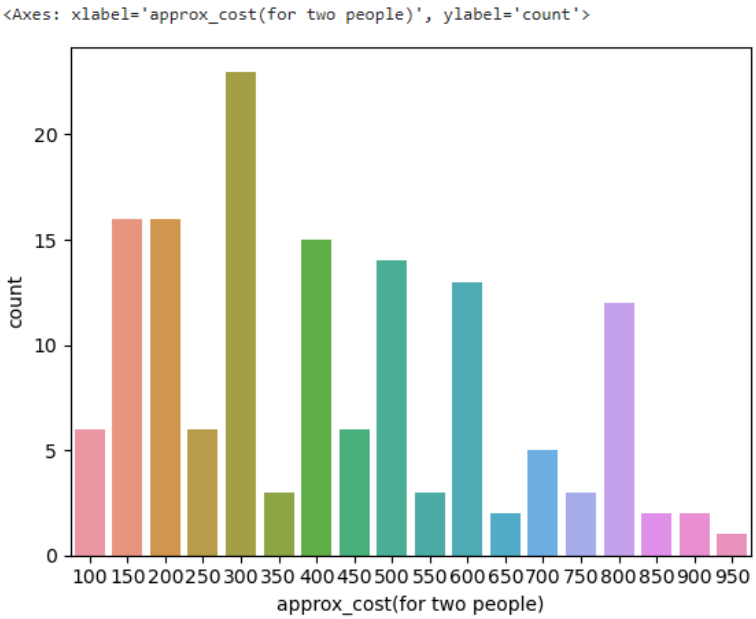
- Most of the restaurants hold the rating between 3.5 – 4.

10. EXPLORE COLUMN - APPROXIMATE COST FOR TWO PEOPLE

SYNTAX

```
couple_data = Zomato['approx_cost(for two people)']
sns.countplot(x= couple_data)
```

OUTPUT



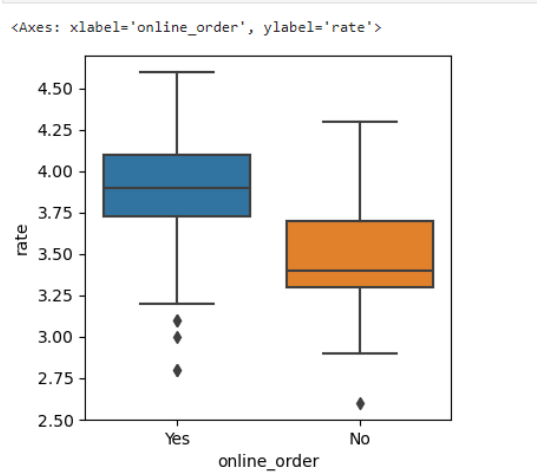
- The approximate cost for two people being 300 is the most preferred among customers.

11. EXPLORE RATING BETWEEN ONLINE AND OFFLINE ORDERS:

SYNTAX:

```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = Zomato)
```

OUTPUT:



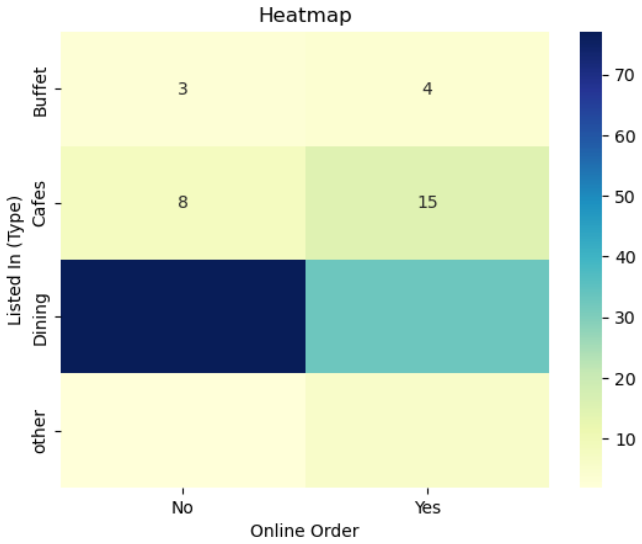
- Offline orders received lesser ratings compared to online orders.

12. COMPARSION BETWEEN DINING RESTAURANTS AND CAFES.

SYNTAX:

```
pivot_table = Zomato.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()
```

OUTPUT:



- Dining restaurants receive offline orders, whereas cafes receive more online orders comparatively. In clear, customers would love to take away from the restaurants but are comfortable in placing online orders on cafes.