

# Zomato Data Analysis Using Python

Zomato has an average of 17.5 million monthly transacting customers for its food delivery business' average monthly active food delivery restaurant partners on Zomato's platform have also increased by 8.7% year-on-year, from 208,000 to 226,000. You are working in a data-driven role at Zomato. You have a dataset of customers. As a data professional, you need to analyze the data, perform EDA (Exploratory Data Analysis) and visualization, and answer the following questions:

# Zomato Data Analysis Using Python

[4]:

	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

# Zomato Data Analysis Using Python

- ▶ 1) What type of restaurant do the majority of customers order from?
- ▶ 2) How many votes has each type of restaurant received from customers?
- ▶ 3) What are the ratings that the majority of restaurants have received?
- ▶ 4) Zomato has observed that most couples order most of their food online. What is their average spending on each order?
- ▶ 5) Which mode (online or offline) has received the maximum rating?
- ▶ 6) Which type of restaurant received more offline orders, so that Zomato can provide those customers with some good offers?

# Zomato Data Analysis Using Python

## Import Libraries

```
[66]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

## Create Data Frame

```
[4]: dataframe
```

```
[4]:
```

	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

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

## Convert the data type of 'rate' column into float and remove denominator

```
[77]: import pandas as pd

# Assuming you have your data in a DataFrame called df
# If your data is in a CSV or another format, load it like this:
# df = pd.read_csv("your_file.csv")

# Convert the 'rate' column to string type, then remove '/5' and convert to float
dataframe['rate'] = dataframe['rate'].astype(str).str.replace('/5', '', regex=False)

# Convert the 'rate' column to float
dataframe['rate'] = dataframe['rate'].astype(float)

# Display the updated DataFrame
print(dataframe.head())
```

	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

```
## Summary of the data frame..
```

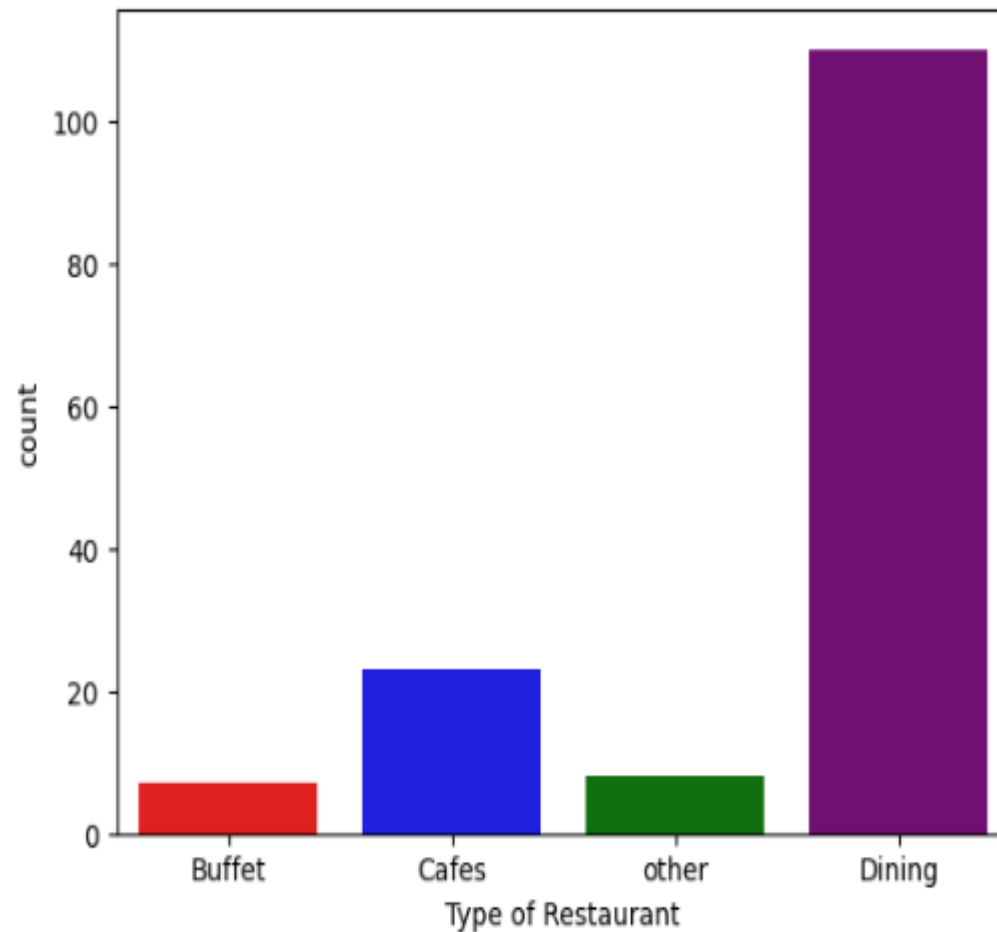
```
[26]: dataframe.info()
```

```
<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
```

## Restaurent Category ratio with Chart.

```
[61]: sns.countplot(x=dataframe['listed_in(type)'], palette=["red", "blue", "green", "purple"], hue=dataframe['listed_in(type)'])  
  
# Label the x-axis  
plt.xlabel("Type of Restaurant")
```

```
[61]: Text(0.5, 0, 'Type of Restaurant')
```



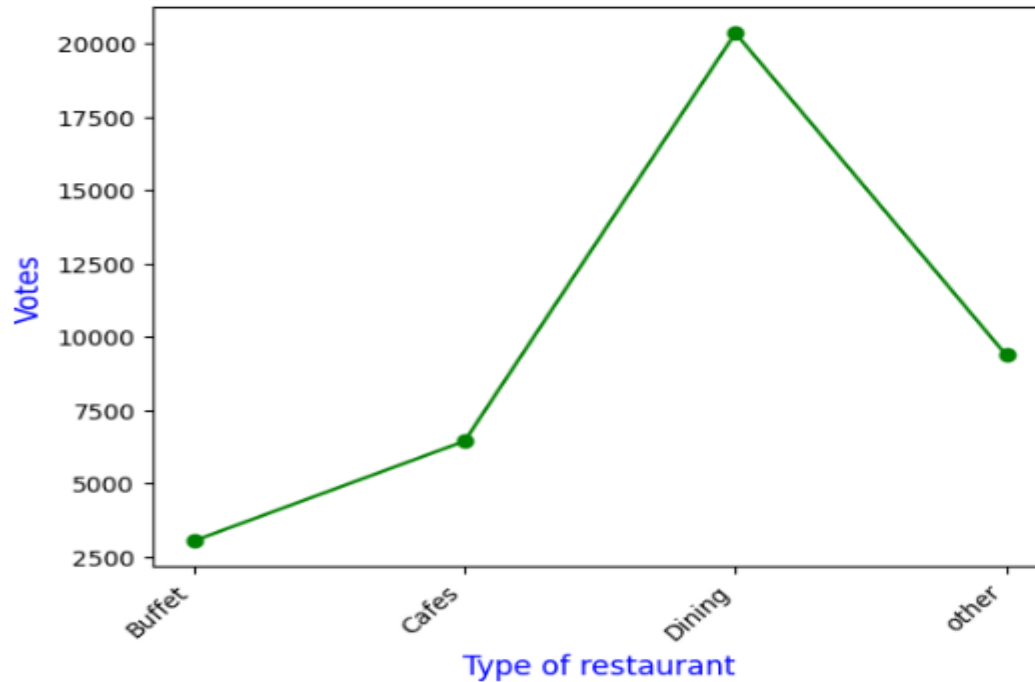
## Rating ration of Restaurents acording to customer votes.

```
[65]: grouped_data = dataframe.groupby('listed_in(type)')['votes'].sum() # Add parentheses to sum
      result = pd.DataFrame({'votes': grouped_data}) # Create a DataFrame

# Plot the data
plt.plot(result.index, result['votes'], c="green", marker="o") # Specify the x values (index) and y values (votes)
plt.xlabel("Type of restaurant", c="blue", size=12) # Label for the x-axis
plt.ylabel("Votes", c="blue", size=12) # Label for the y-axis

# Optional: format the x-ticks for better clarity
plt.xticks(rotation=45, ha='right') # Rotate x-axis labels if needed

plt.show() # Display
```

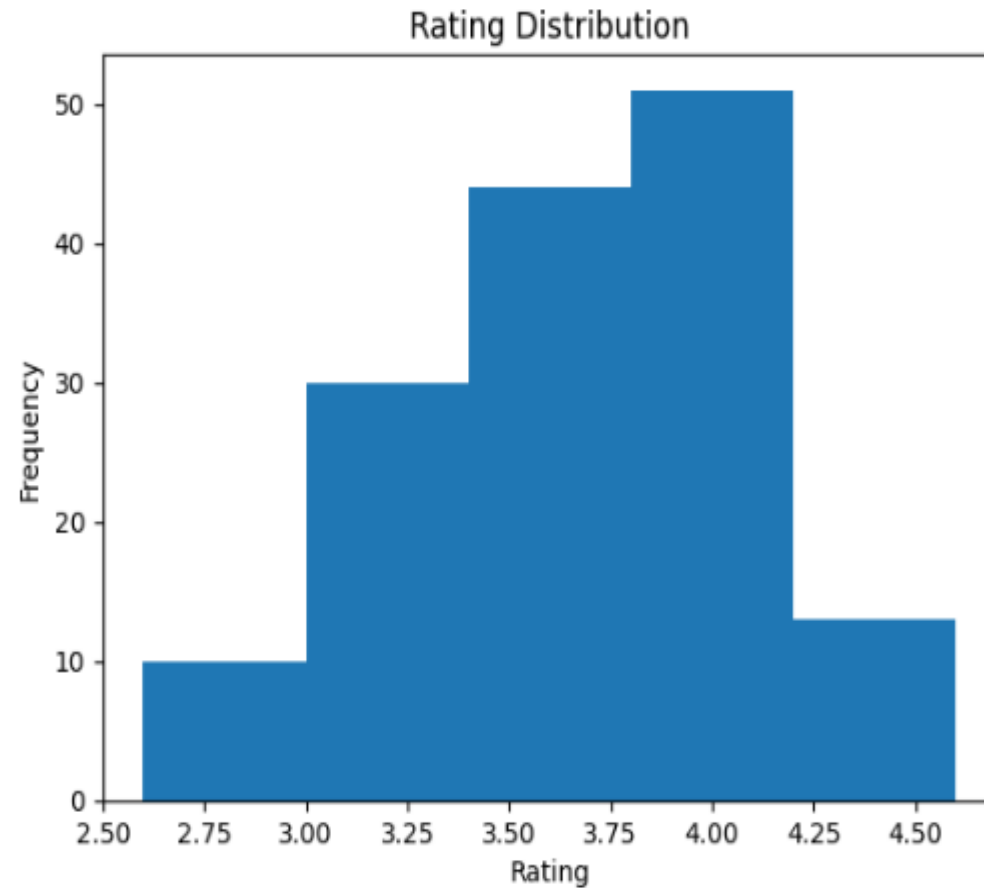


The above visualization shows that dinning restaurents take highest votes and customer satisfaction.



## ▼ Highest rating range from total range of 5

```
[78]: plt.hist(dataframe['rate'], bins=5) # bins=5 for 5 bins in the histogram  
plt.title("Rating Distribution")  
plt.xlabel('Rating')  
plt.ylabel('Frequency')  
plt.show()
```



- ▼ The above histogram shows that majority of restaurants gain rating 3.75 to 4.25 out of 5.

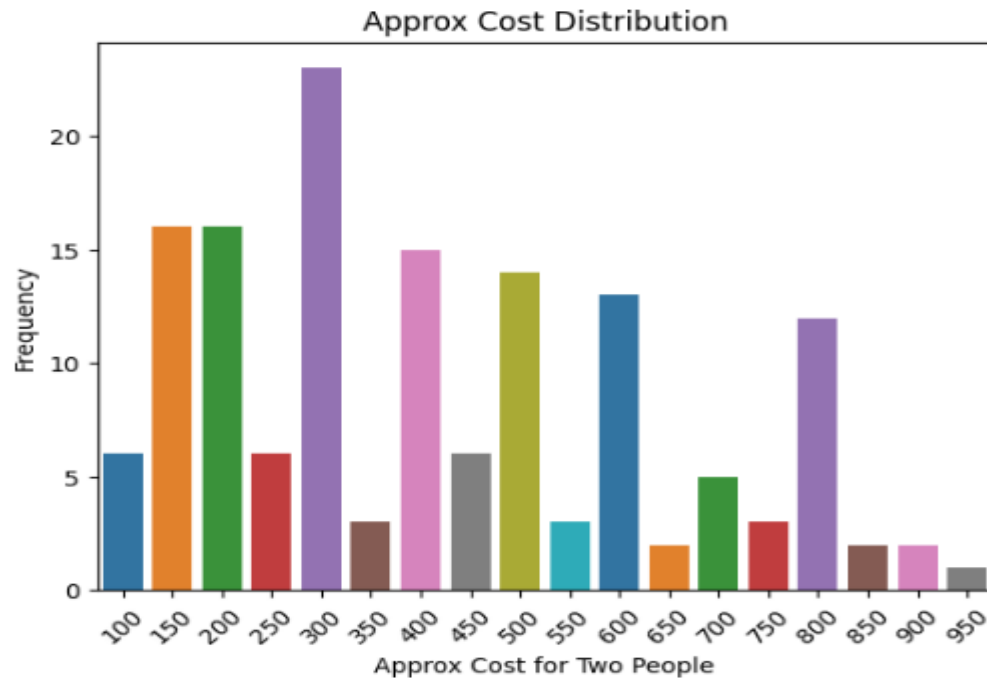
## Customer preference price ratio..

```
[81]: couple_data = dataframe['approx_cost(for two people)']

# Create the countplot with a palette of different colors, using 'hue'
sns.countplot(data=dataframe, x='approx_cost(for two people)', hue='approx_cost(for two people)', palette='tab10')

# Optional: Turn off the legend as it's not necessary when hue is the same as x
plt.legend([],[], frameon=False)

# Add Labels and title
plt.title('Approx Cost Distribution')
plt.xlabel('Approx Cost for Two People')
plt.ylabel('Frequency')
plt.xticks(rotation=45) # Rotate x-axis labels if necessary for readability
plt.show()
```



The above chart shows the majority of customer prefer the restaurants of maximum of 300 cost.

## Rating ratio between online and offline orde

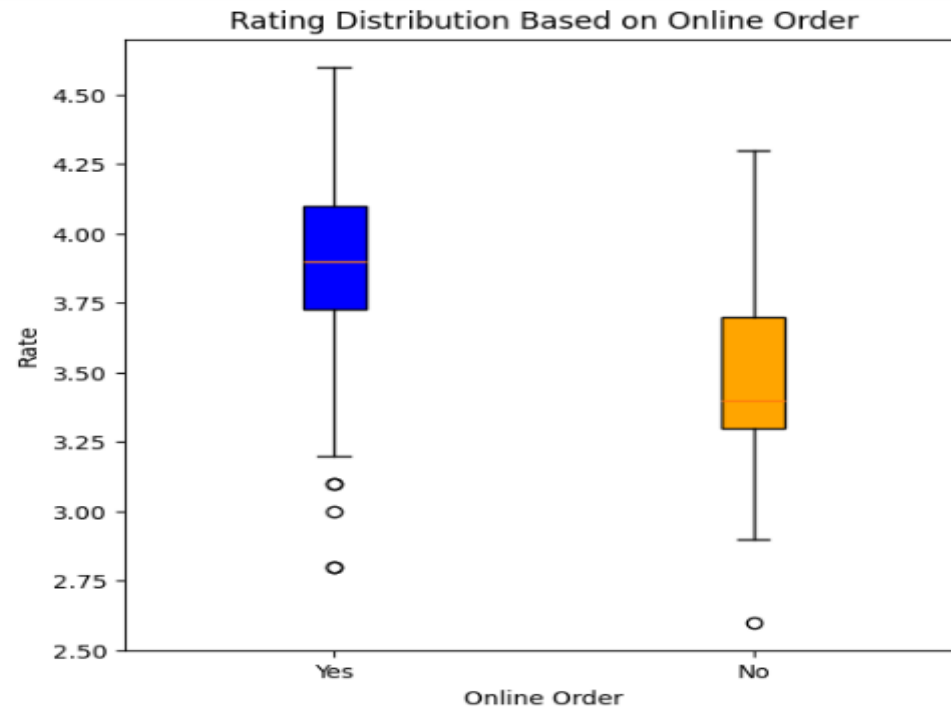
```
[88]: plt.figure(figsize=(6,6))

# Create the boxplot
box = plt.boxplot([dataframe[dataframe['online_order'] == 'Yes']['rate'],
                    dataframe[dataframe['online_order'] == 'No']['rate']],
                  patch_artist=True, # This makes the boxes filled with color
                  tick_labels=['Yes', 'No'])

# Set colors for the boxes
colors = ['blue', 'orange']
for patch, color in zip(box['boxes'], colors):
    patch.set_facecolor(color)

# Set titles and labels
plt.title('Rating Distribution Based on Online Order')
plt.xlabel('Online Order')
plt.ylabel('Rate')

plt.show()
```



online order received greater rating as compared to offline order.

## online/offline ratio between different categories of restaurants,

```
[91]: pivot_table = dataframe.pivot_table(index='listed_in(type)', columns='online_order', aggfunc='count')

# Create a heatmap
sns.heatmap(pivot_table, annot=True, cmap="YlGnBu", fmt='d')

# Set the title and labels
plt.title("HeatMap")
plt.xlabel("Online Order")
plt.ylabel("Listed In (Type)")

# Display the heatmap
plt.show()
```

