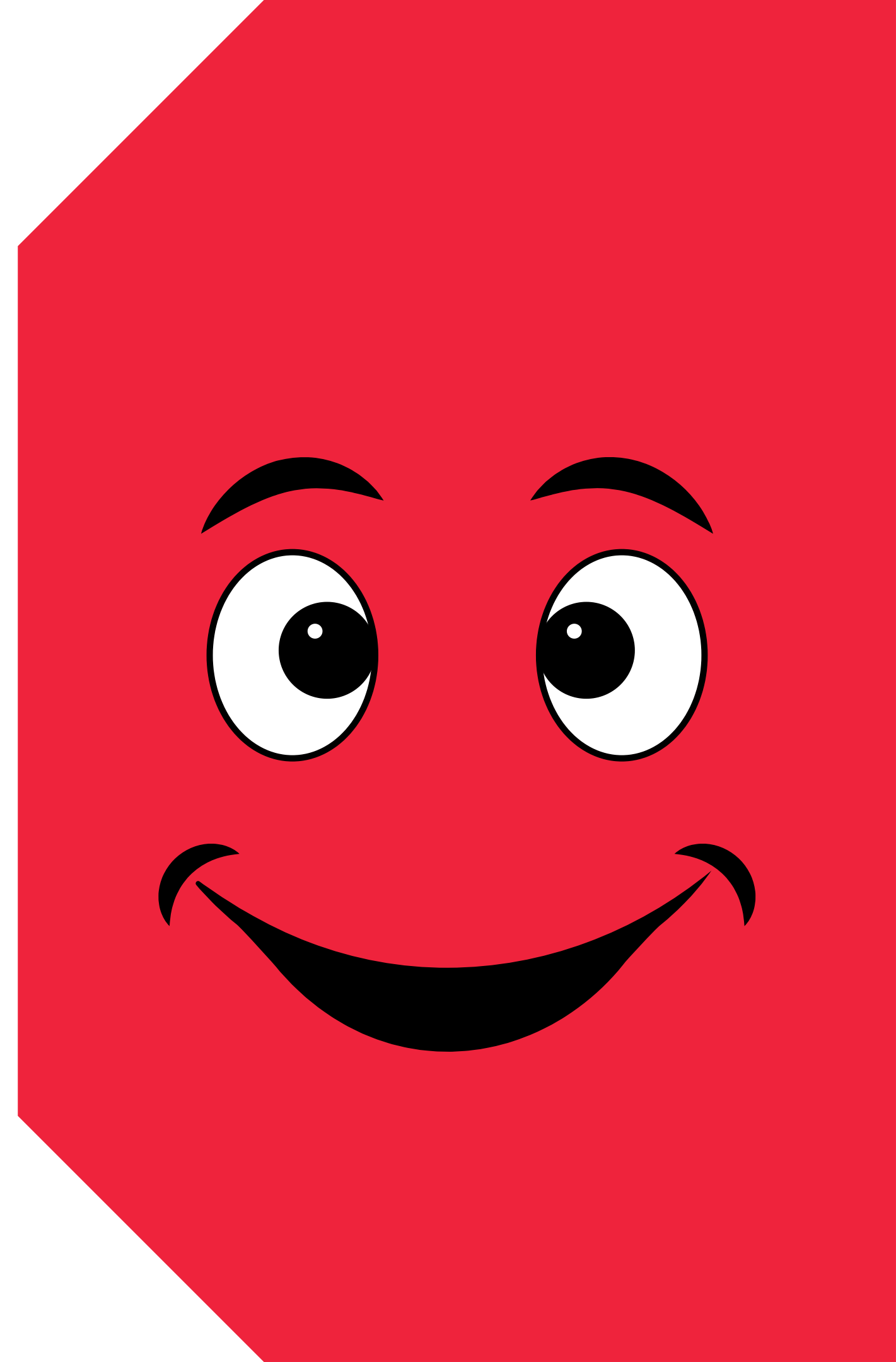


@mohammad-kaif1

# Zomato

An Exploratory Data Analysis  
using python



# Zomato

A solid red circle is positioned on the left side of the slide, partially cut off by the edge.

Zomato had over 11.2 million transacting users on a monthly average during the first half of the financial year 2020, a significant increase from the previous year. The food delivery company offered its delivery services in about 500 cities across India with over 200 thousand delivery partners at that time.

Zomato is an Indian startup whose business is mainly restaurant aggregation and food delivery services. The company was founded by Deepinder Goyal and Pankaj Chaddah in 2008.

**I'll address these questions 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?

## Importing Libraries:

- The first step is importing essential libraries in Python. The libraries used are:
  - **pandas: For data manipulation and analysis.**
  - **numpy: For numerical operations.**
  - **matplotlib.pyplot: For plotting and visualization.**
  - **seaborn: For statistical data visualization.**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

## Loading the Dataset:

- The dataset named "**Zomato data .csv**" is loaded into a DataFrame using the **pd.read\_csv()** function.

```
dataframe = pd.read_csv("Zomato data .csv")
```

dataframe							
	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

## Previewing the Data:

- **The DataFrame is previewed, showing 7 columns:**
  - a.name: Name of the restaurant.
  - b.online\_order: Indicates whether the restaurant offers online ordering.
  - c.book\_table: Shows whether table booking is available.
  - d.rate: Rating of the restaurant (e.g., 4.1/5).
  - e.votes: Number of votes the restaurant received.
  - f.approx\_cost(for two people): Approximate cost for two people.
  - g.listed\_in(type): Category in which the restaurant is listed (e.g., Buffet, Dining).
- **The dataset contains 148 rows and 7 columns, as seen in the last image.**

## dataframe.shape

- Purpose: The .shape attribute in Pandas is used to determine the size or dimensions of the DataFrame. It returns a tuple that represents the number of rows and columns in the DataFrame.
- Output (148, 7):
  - 148: This is the number of rows in the DataFrame.
  - 7: This is the number of columns in the DataFrame.

```
dataframe.shape
```

```
(148, 7)
```

```
dataframe.columns
```

```
Index(['name', 'online_order', 'book_table', 'rate', 'votes',  
      'approx_cost(for two people)', 'listed_in(type)'],  
      dtype='object')
```

## dataframe.columns

- Purpose: The .columns attribute in Pandas returns the labels of the columns of the DataFrame. It essentially lists the names of all the columns in the DataFrame

# Checking the DataFrame Information

- **Code: dataframe.info()**
- The output gives details about the DataFrame:
- The DataFrame has 148 entries (rows), indexed from 0 to 147.
- There are 7 columns with information on non-null counts and data types:
- Columns like name, online\_order, book\_table, rate, and listed\_in(type) have the object data type.
- votes and approx\_cost(for two people) have the int64 data type.

**Purpose: This step is useful to understand the data structure, identify null values, and know the types of data in each column.**

```
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    object
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: int64(2), object(5)
memory usage: 8.2+ KB
```

## Converting the Data Type of the rate Column

- Problem: The rate column contains ratings in a string format like "4.1/5". To perform numerical operations, we need to convert it to a float.
- Solution: A custom function is created to handle this:

### Explanation:

- The handlerate function splits the value by the "/" character, extracts the first part (the rating before the "/"), and converts it to a float.
- The apply function is used to apply this transformation to each value in the rate column.

```
def handlerate(value):  
    value=str(value).split("/")  
    value=value[0];  
    return float(value)  
  
dataframe["rate"]=dataframe["rate"].apply(handlerate)  
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 Udipi 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

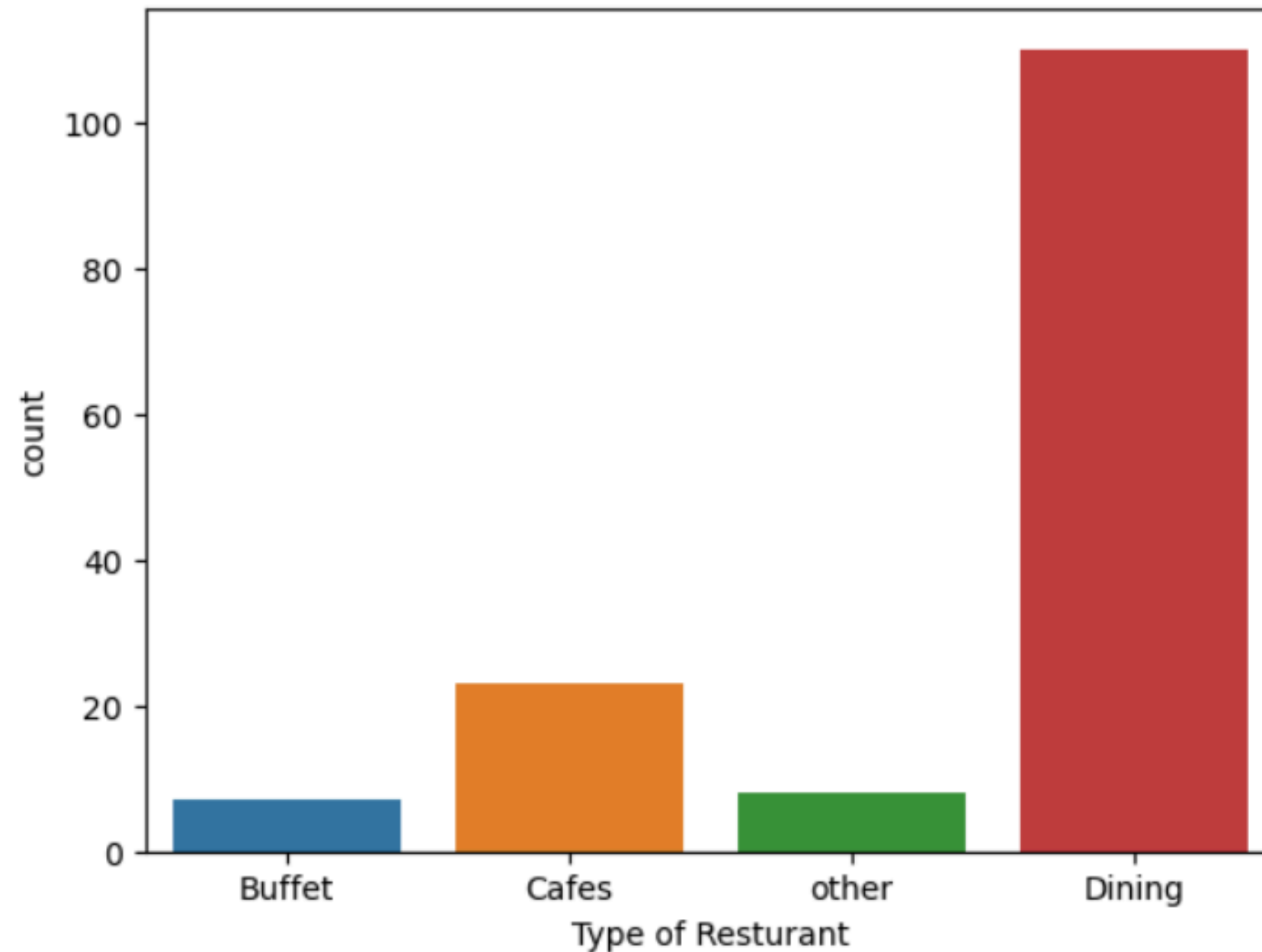


## Q.what type of restaurant do the majority of customers order from?

### Type of Resturant

```
sns.countplot(x=dataframe["listed_in(type)"])  
plt.xlabel("Type of Resturant")
```

```
Text(0.5, 0, 'Type of Resturant')
```

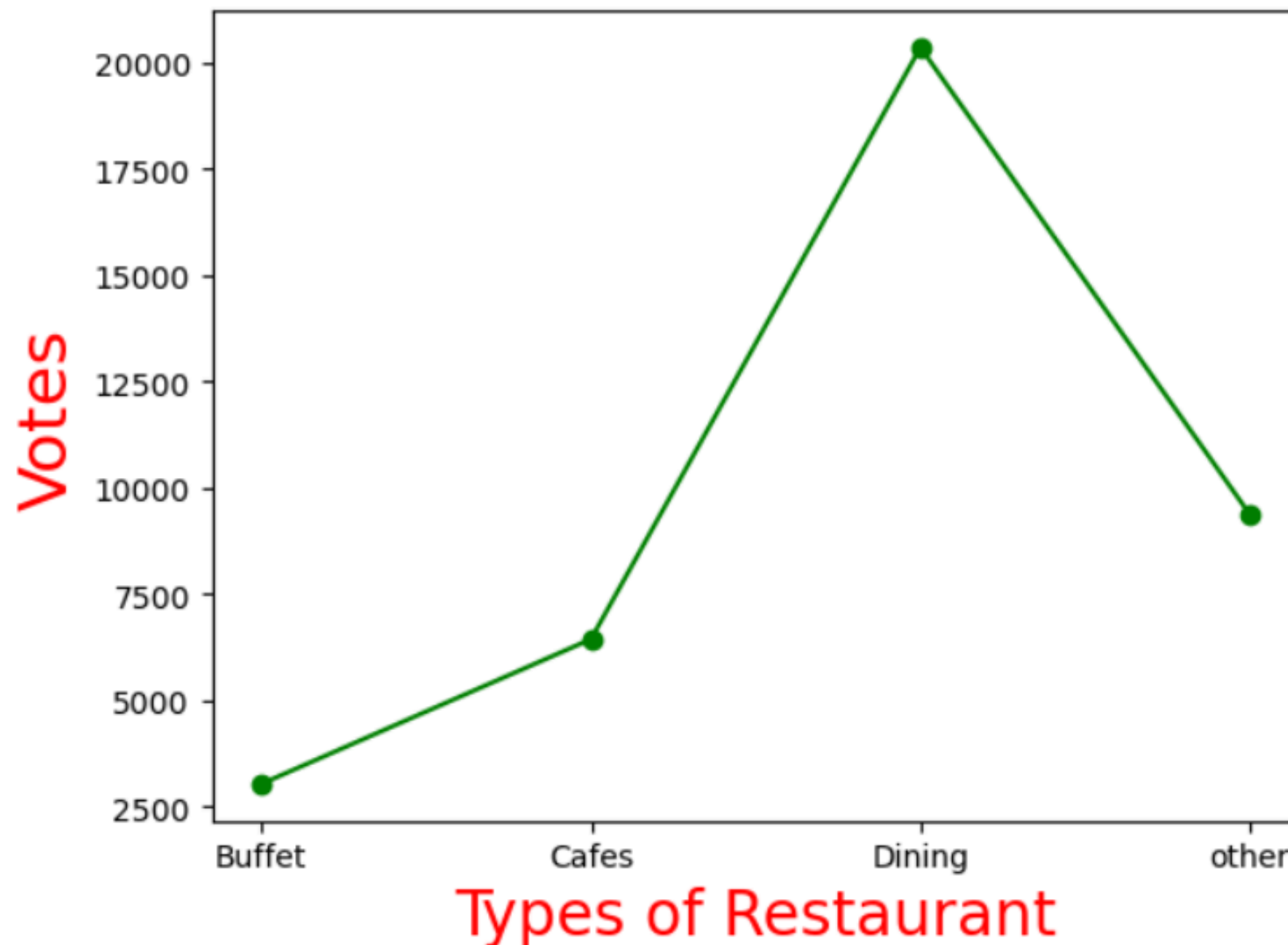


majority of restaurant falls in Dining category

## Q2.How many votes has each type of restaurant received from customers?

```
grouped_data = dataframe.groupby("listed_in(type)")["votes"].sum()

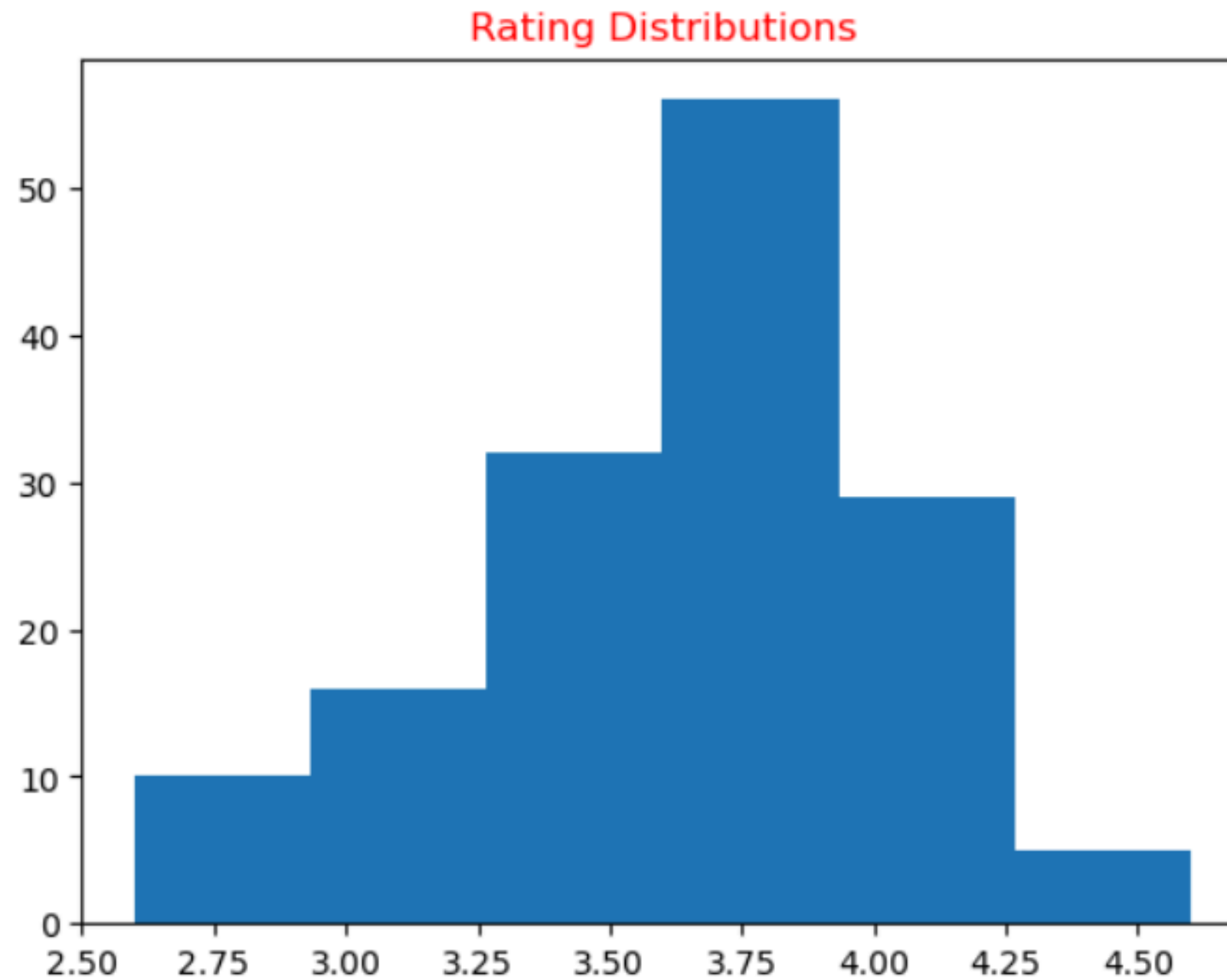
plt.plot(grouped_data.index, grouped_data.values, color="green", marker="o")
plt.xlabel("Types of Restaurant", color="red", fontsize=22)
plt.ylabel("Votes", color="red", fontsize=22)
plt.show()
```



dining restaurant has recieved maximum Votes

### Q3.What are the ratings that the majority of restaurants have received?

```
plt.hist(dataframe["rate"],bins=6)  
plt.title("Rating Distributions",color="red")  
plt.show()
```

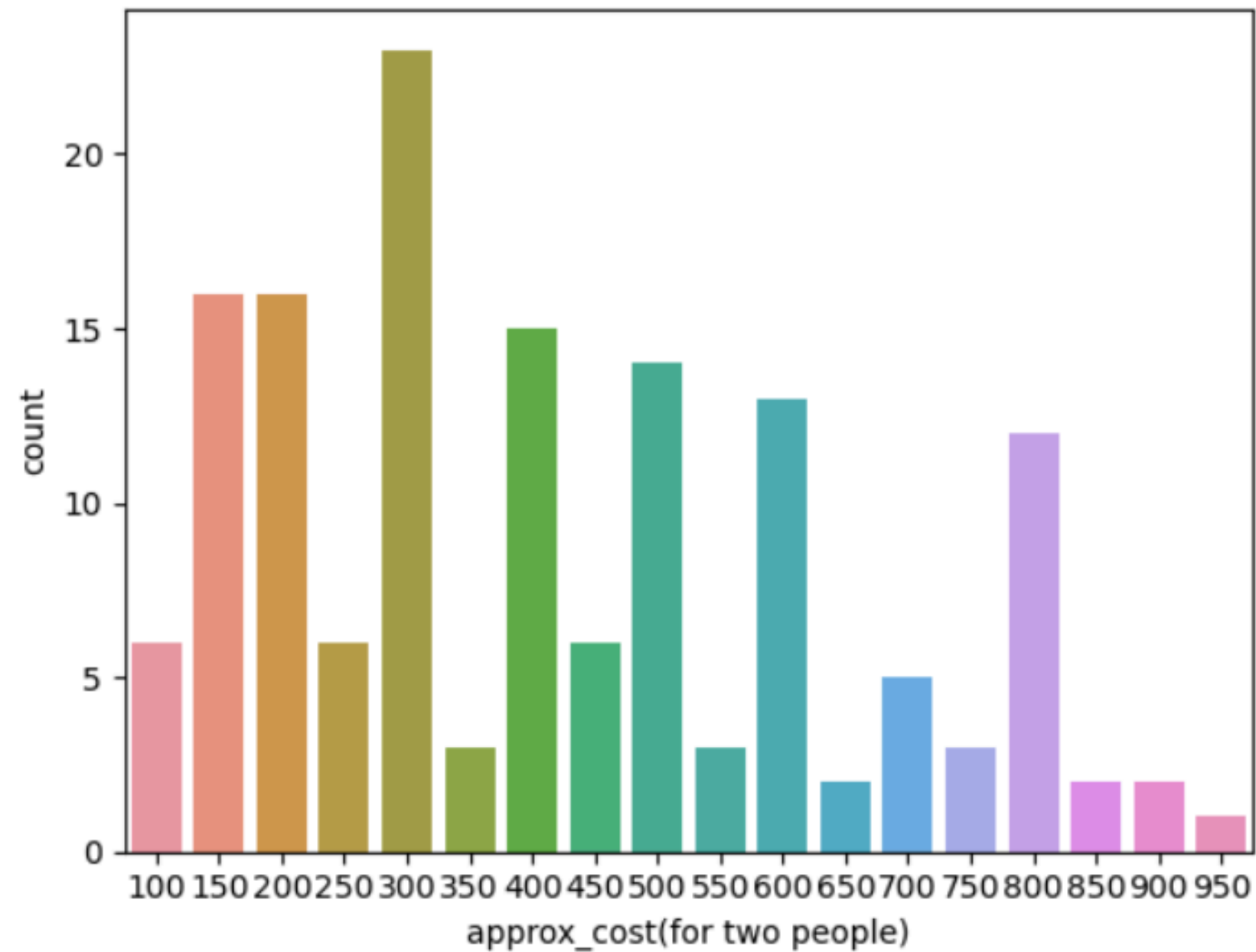


The majority rasturants recieved rating from 3.5 to 4.5

**Q4.Zomato has observed that most couples order most of their food online. What is their average spending on each order?**

```
coupledata = dataframe["approx_cost(for two people)"]  
sns.countplot(x=coupledata)
```

<Axes: xlabel='approx\_cost(for two people)', ylabel='count'>

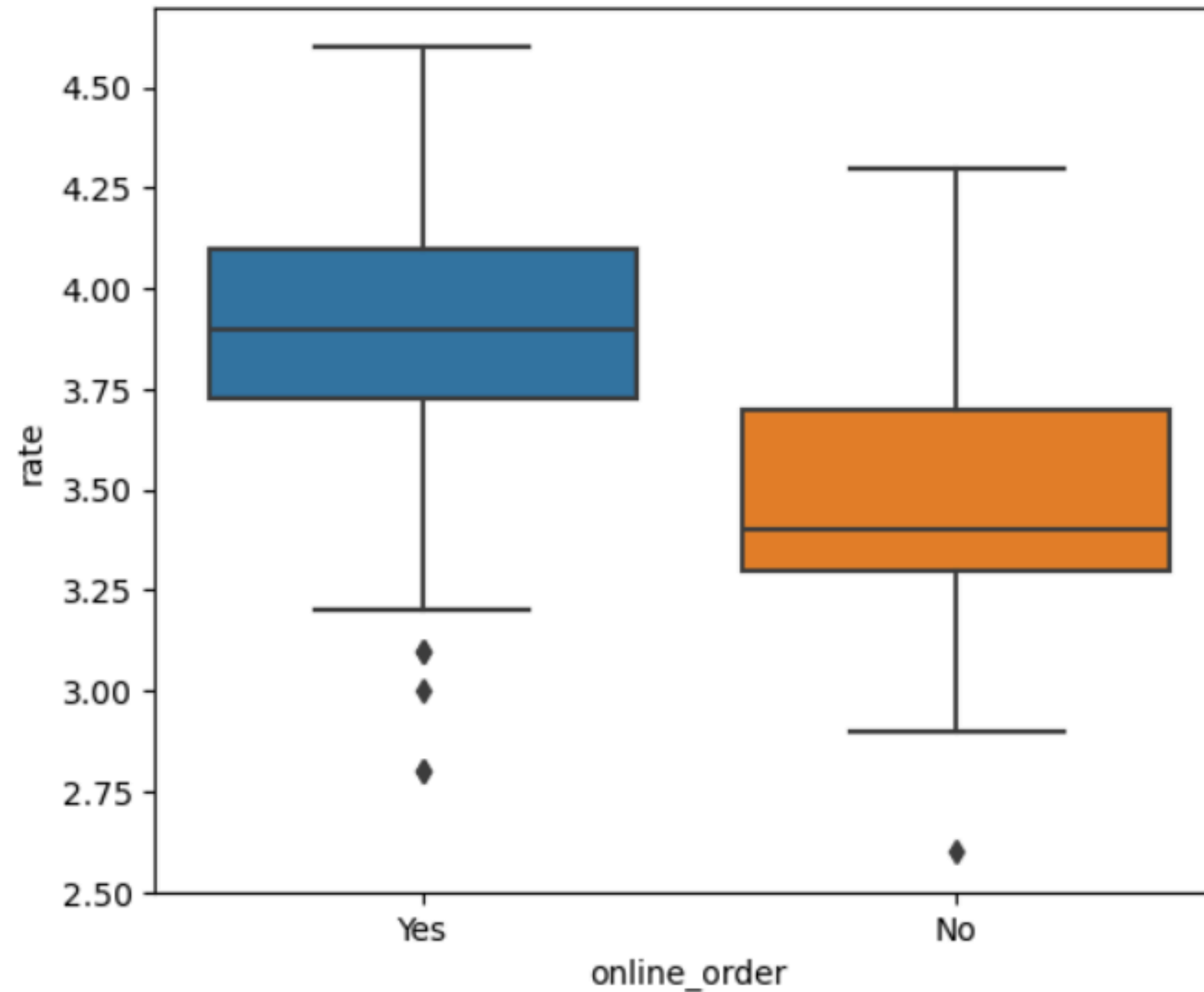


the majority of couples prefer restaurants with an approximate cost of 300 rupees

## Q5.Which mode (online or offline) has received the maximum rating?

```
plt.figure(figsize=(6,5))  
sns.boxplot(x = "online_order", y = "rate",data = dataframe)
```

<Axes: xlabel='online\_order', ylabel='rate'>

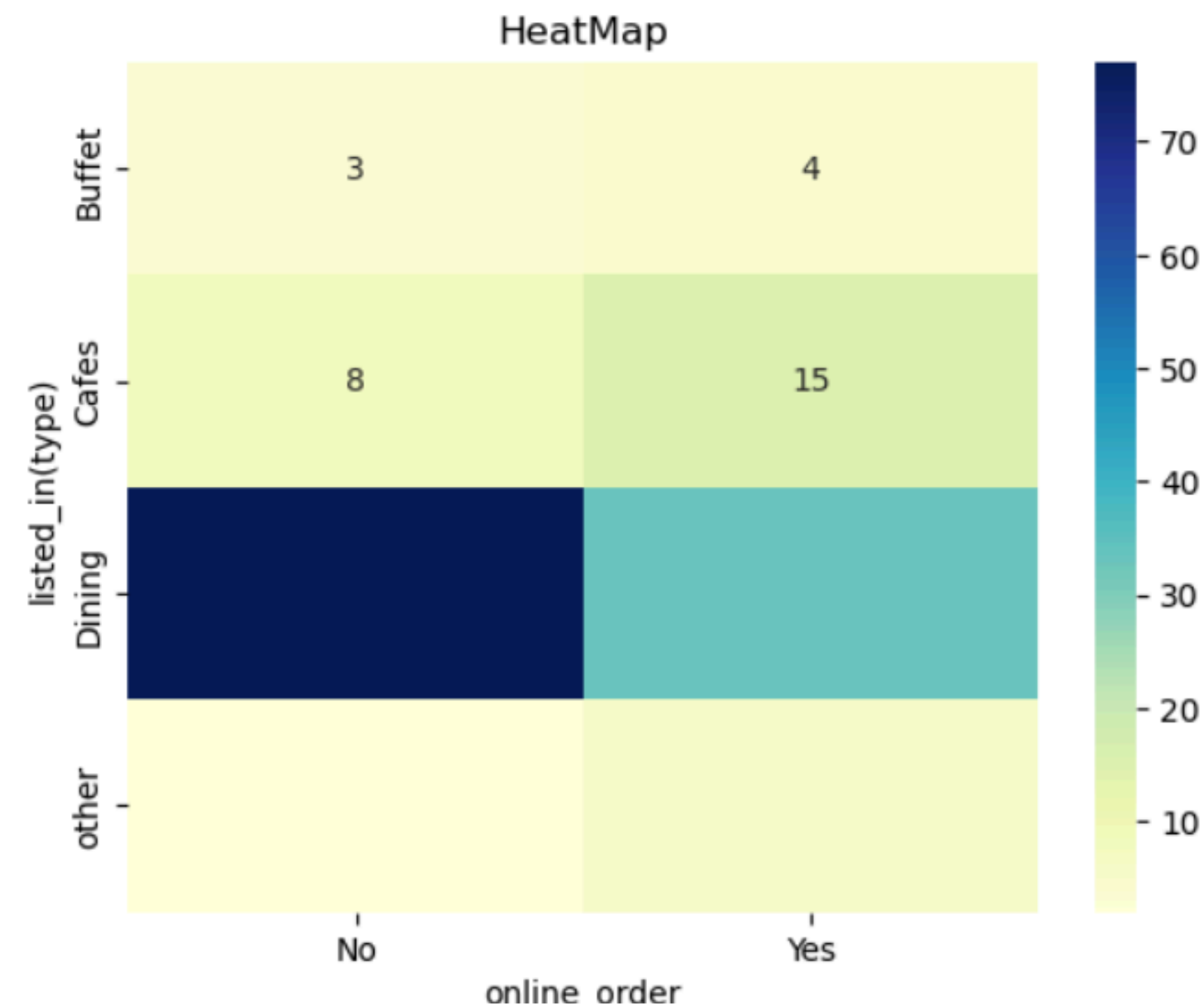


Offline orders recieved lower rating in comparison to Online rating

## Q6.Which type of restaurant received more offline orders, so that Zomato can provide those customers with some good offers?

```
pivot = dataframe.pivot_table(index="listed_in(type)",columns="online_order", aggfunc="size", fill_value=0)
sns.heatmap(pivot, annot=True, cmap="YlGnBu", fmt="d")
plt.title("HeatMap")
plt.xlabel("online_order")
plt.ylabel("listed_in(type)")
plt.show
```

<function matplotlib.pyplot.show(close=None, block=None)>



Thank  
you

@mohammad-kaif1