Zomato Data Analaysis Project

In [137]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [46]:
```
data = pd.read_csv(r"C:\Users\Admin\Downloads\Zomatodata.csv")
```

In [32]:
```
data
```

Out[32]:

	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

In [33]:
```
data.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
```

In [34]:
```
data.duplicated().sum()
```

Out[34]:
```
0
```

In [35]:
```
data.describe()
```

Out[35]:

	votes	approx_cost(for two people)
count	148.000000	148.000000
mean	264.810811	418.243243
std	653.676951	223.085098
min	0.000000	100.000000
25%	6.750000	200.000000
50%	43.500000	400.000000
75%	221.750000	600.000000
max	4884.000000	950.000000

In [36]:
```
data.describe(include = "all")
```

Out[36]:

	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
count	148	148	148	148	148.000000	148.000000	148
unique	145	2	2	20	NaN	NaN	4
top	San Churro Cafe	No	No	3.8/5	NaN	NaN	Dining
freq	2	90	140	19	NaN	NaN	110
mean	NaN	NaN	NaN	NaN	264.810811	418.243243	NaN
std	NaN	NaN	NaN	NaN	653.676951	223.085098	NaN
min	NaN	NaN	NaN	NaN	0.000000	100.000000	NaN
25%	NaN	NaN	NaN	NaN	6.750000	200.000000	NaN
50%	NaN	NaN	NaN	NaN	43.500000	400.000000	NaN
75%	NaN	NaN	NaN	NaN	221.750000	600.000000	NaN
max	NaN	NaN	NaN	NaN	4884.000000	950.000000	NaN

In [37]:
```
data.isnull().sum()
```

Out[37]:
```
name                           0
online_order                   0
book_table                     0
rate                           0
votes                          0
approx_cost(for two people)    0
listed_in(type)                0
dtype: int64
```

In [21]:
```
data.head()
```

Out[21]:

	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

In [29]:
```
```

In [38]:
```
def handle_rate(value):
    value = str(value).split("/")
    value = value[0]
    return float(value)
```

In [40]:
```
data["rate"] = data["rate"].apply(handle_rate)
```

In [41]:
```
data
```

Out[41]:

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

148 rows × 7 columns

In [47]:
```
# data["rate"] = data["rate"].apply(lambda x: float(str(x).split("/")[0]) if x else
```

In [ ]:

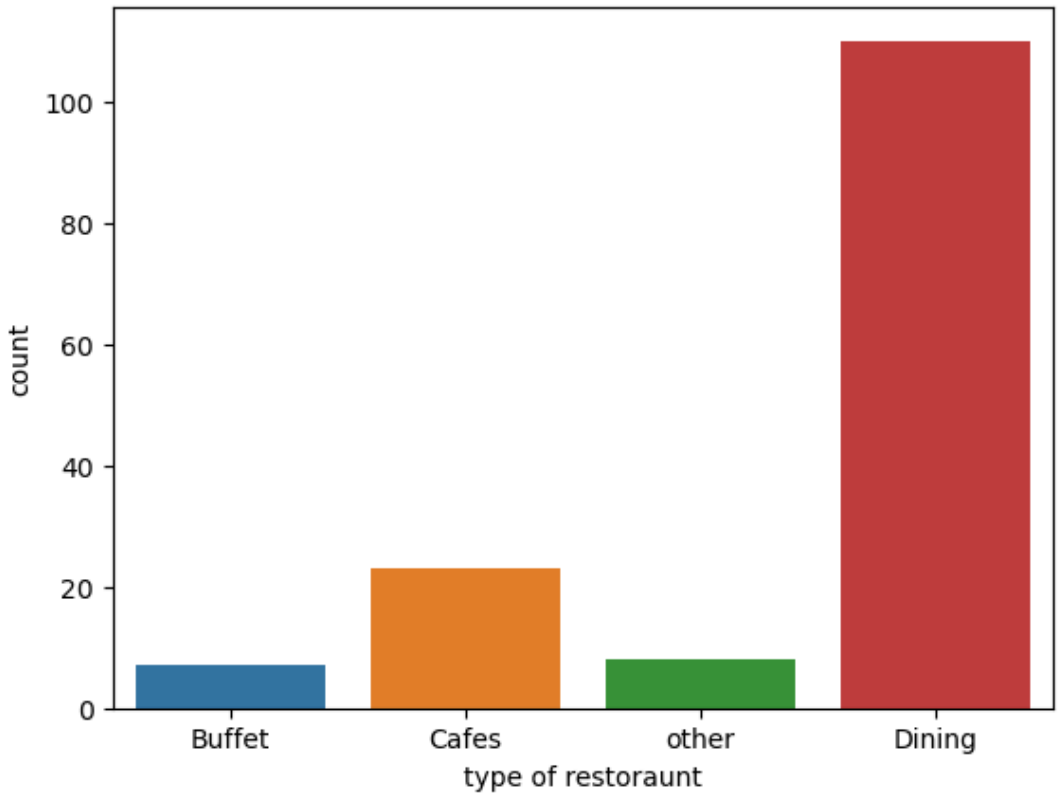
What type of restaurants do the majority of customers order from which majority of customers apply ?

In [51]:
```
data.groupby("listed_in(type)").aggregate({"name":"count"})
```

Out[51]:

listed_in(type)	name
Buffet	7
Cafes	23
Dining	110
other	8

In [62]:
```
sns.countplot(x = data["listed_in(type)"])
plt.xlabel("type of restaurant")
plt.xticks()
plt.yticks()
plt.show()
```



In [64]:
```
### conclusion :- majority of customers falls in dining category restaurant
```

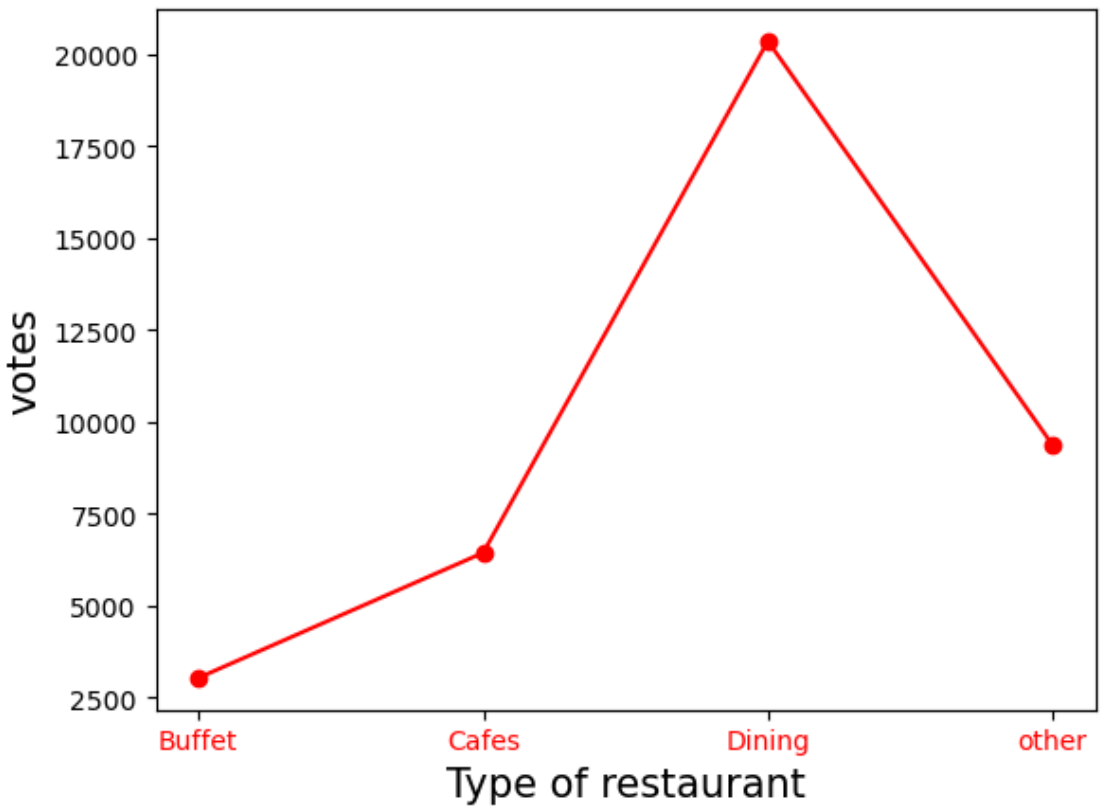
how many votes has each type of restaurant received from customers ?

In [77]:
```
group_data = data.groupby("listed_in(type)").aggregate({"votes":"sum"})
group_data
```

Out[77]:

listed_in(type)	votes
Buffet	3028
Cafes	6434
Dining	20363
other	9367

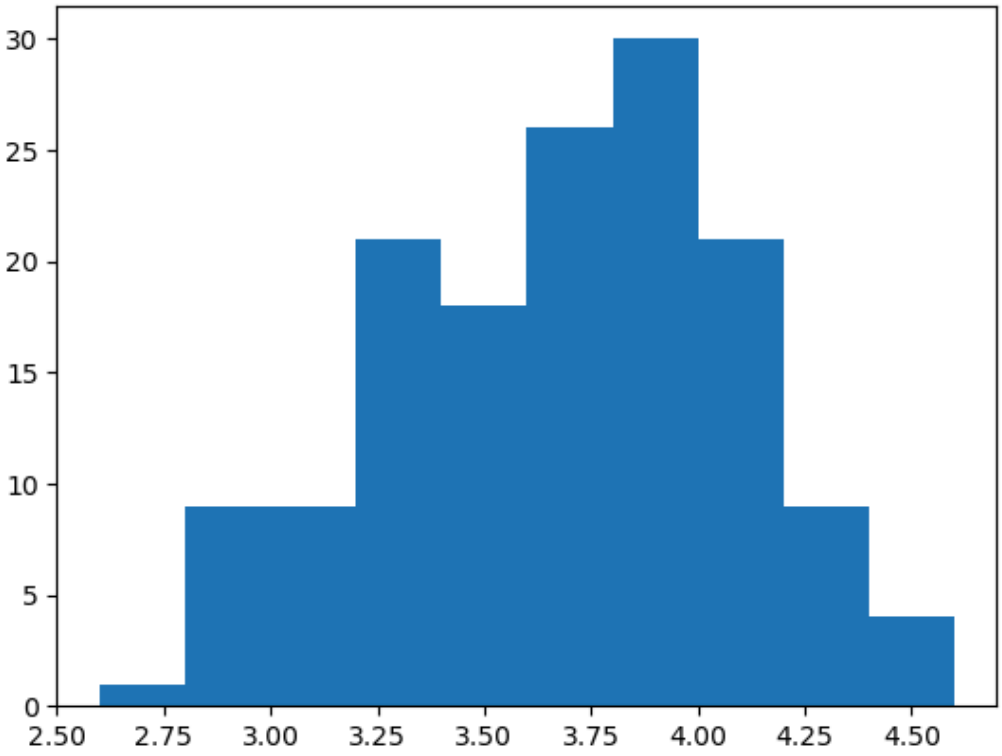
In [92]:
```
plt.plot(group_data, color = "r", marker = "o")
plt.xlabel("Type of restaurant", size = 15)
plt.xticks(color = "r")
plt.yticks()
plt.ylabel("votes", size = 15)
plt.show()
```



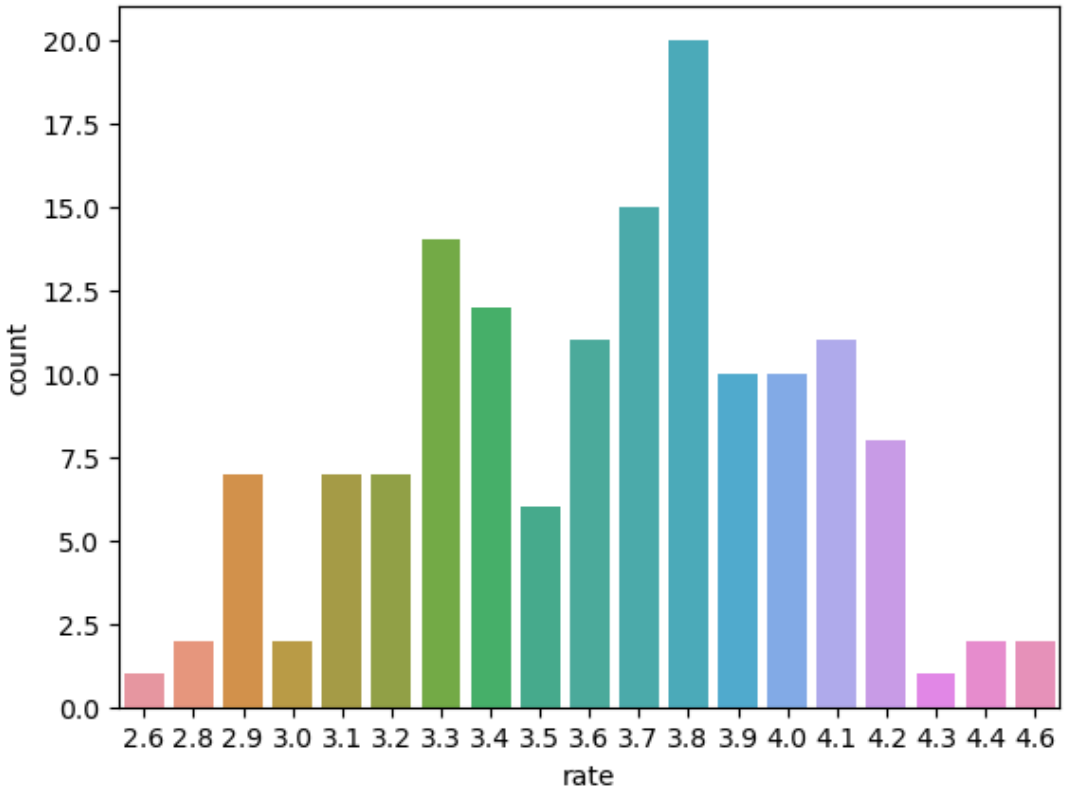
In [93]:
```
### conclusion : dining restaurant has received maximum votes
```

What are the ratings that majority of the restaurants received ?

In [96]:
```
plt.hist(data["rate"])
plt.show()
```



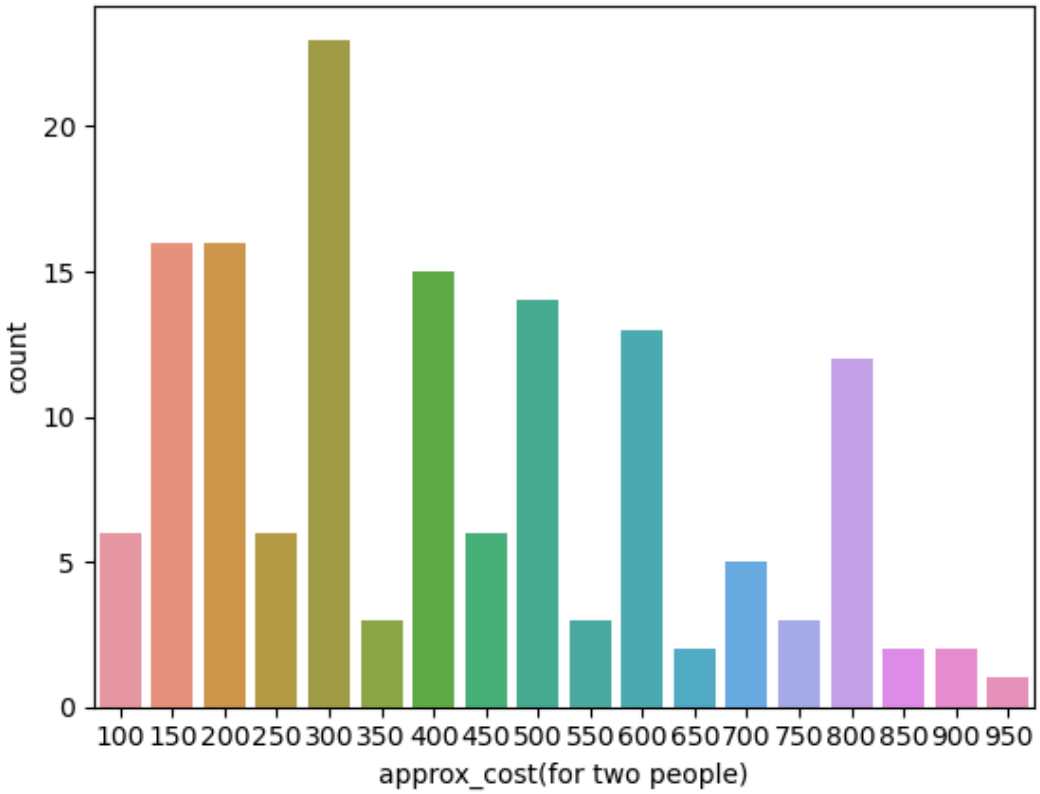
In [99]:
```
sns.countplot(x = data["rate"], data = data)
plt.show()
```



In [100]:
```
### conclusion : 3.8 is the rating given to highest no of restaurants
```

zomato has observed most of the couples order, what is there average spend on each order ?

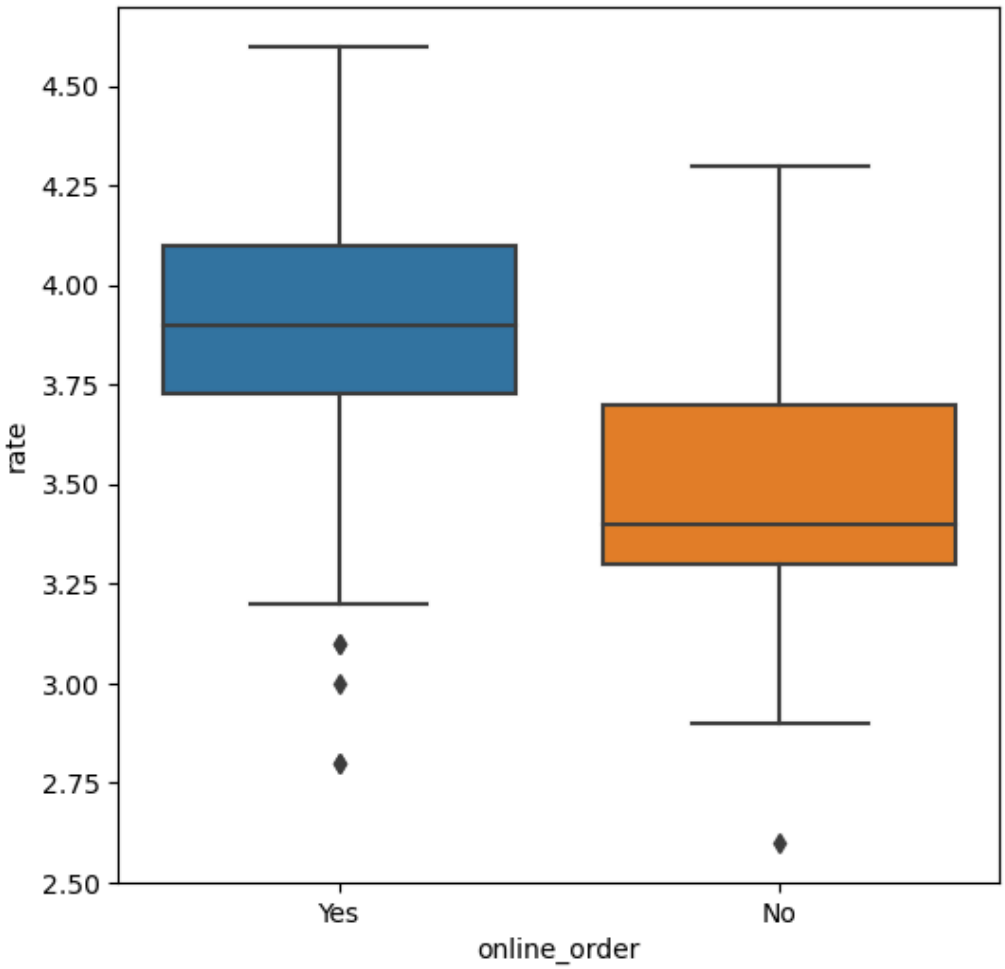
In [106]:
```
sns.countplot(x = data["approx_cost(for two people)"], data = data)
plt.show()
```



In [108]:
```
### conclusion : 300 rs is the money spend by most of the couples
```

Among online and offline which mode has received most rating ?

In [117]:
```
plt.figure(figsize = (6,6))
sns.boxplot(x = "online_order", y = "rate", data = data)
plt.show()
```

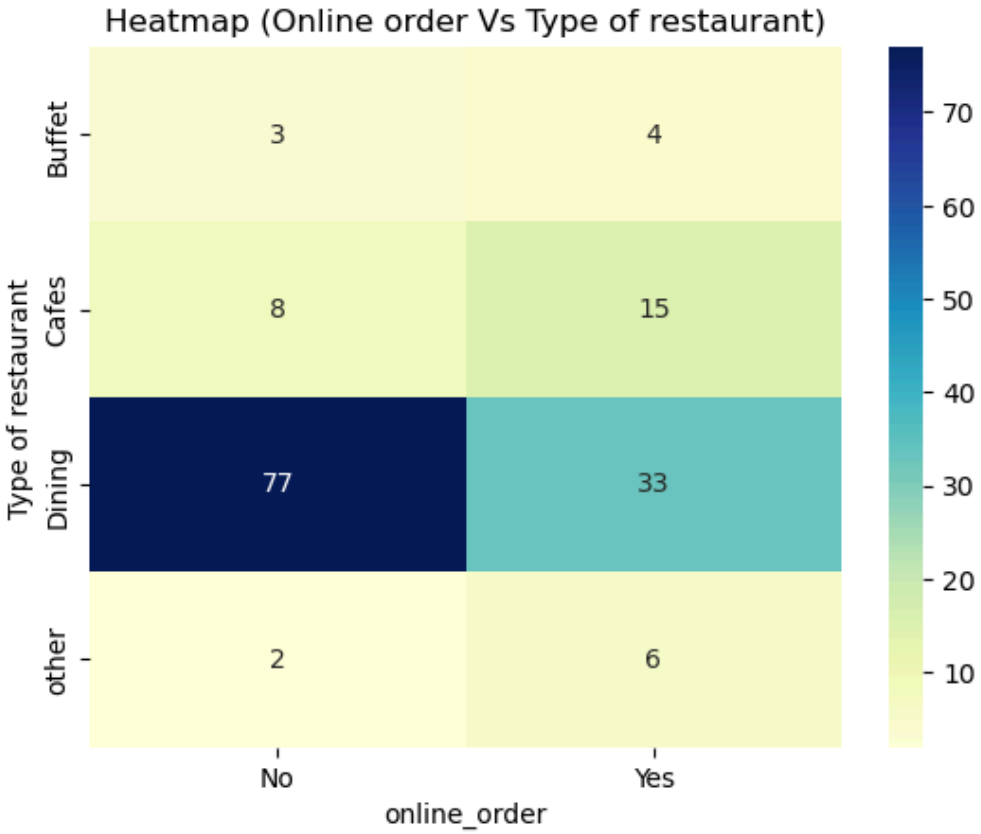


In [118]:
```
### conclusion :- offline order receive lower rating incomparison to offline ratin
```

which type of restaurant has received more orders ?

In [136]:
```
pivot_table = data.pivot_table(index ="listed_in(type)", columns = "online_order",

sns.heatmap(pivot_table, annot = True, cmap = "YlGnBu")
plt.ylabel("Type of restaurant")
plt.title("Heatmap (Online order Vs Type of restaurant)")
plt.show()
```



In [138]:
```
### conclusion : dining restaurants primarily accepts offline orders whereas cafes
### This suggests that customers offline order to feel in presence in dining but pl
```

In [ ]:
```

```