Zomato Restaurant Data Analytics - Customer Behavior Analysis

January 22, 2026

1 Zomato Restaurant Data Analytics: Customer Behavior Analysis

- Understanding customer preferences and restaurant trends helps food businesses make better decisions.
- In this notebook, we analyze the Zomato restaurant dataset using Python to extract useful insights.

1.0.1 Importing necessary libraries and data

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

```python
[2]: df = pd.read_csv('Zomato-data.csv')
     df
```

```
[2]:                      name online_order book_table   rate  votes  \
     0                   Jalsa          Yes        Yes  4.1/5    775
     1           Spice Elephant          Yes         No  4.1/5    787
     2           San Churro Cafe          Yes         No  3.8/5    918
     3     Addhuri Udupi Bhojana          No         No  3.7/5     88
     4             Grand Village          No         No  3.8/5    166
     ..                    …            …          …      …      …
     143         Melting Melodies          No         No  3.3/5      0
     144          New Indraprasta          No         No  3.3/5      0
     145             Anna Kuteera         Yes         No  4.0/5    771
     146                  Darbar          No         No  3.0/5     98
     147            Vijayalakshmi         Yes         No  3.9/5     47

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

```
..                          ...        ...
143                         100        Dining
144                         150        Dining
145                         450        Dining
146                         800        Dining
147                         200        Dining

[148 rows x 7 columns]
```

1.0.2 Data Cleaning and Preparation

```
[3]: def handleRate(value):
         try:
             return float(str(value).split('/')[0])
         except(ValueError, TypeError):
             return None
```

```
[4]: df['rate'] = df['rate'].apply(handleRate)
```

```
[5]: df.head()
```

```
[5]:                     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
```

```
[6]: df.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
```

[7]: `df.isnull().sum()`

```
[7]: name                       0
     online_order               0
     book_table                 0
     rate                       0
     votes                      0
     approx_cost(for two people)  0
     listed_in(type)            0
     dtype: int64
```
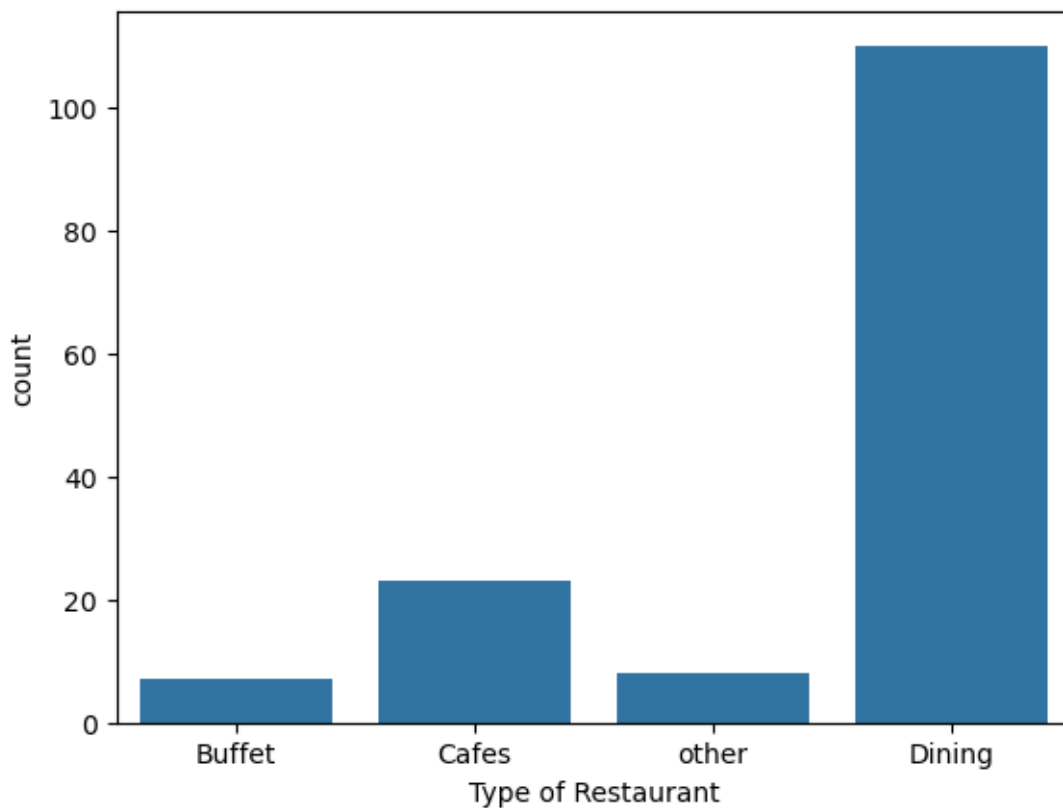
1.0.3 Exploring Restaurant Types

let's see the popular restaurant categories

[8]:
```python
sns.countplot(x=df['listed_in(type)'])
plt.xlabel("Type of Restaurant")
```

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

1.0.4 Insights

Majority of the restaurants are comes under the dinner category

1.0.5 Votes by Restaurant types

```
[9]: grouped_data = df.groupby('listed_in(type)')['votes'].sum()
     grouped_data
```

```
[9]: listed_in(type)
     Buffet      3028
     Cafes       6434
     Dining     20363
     other       9367
     Name: votes, dtype: int64
```
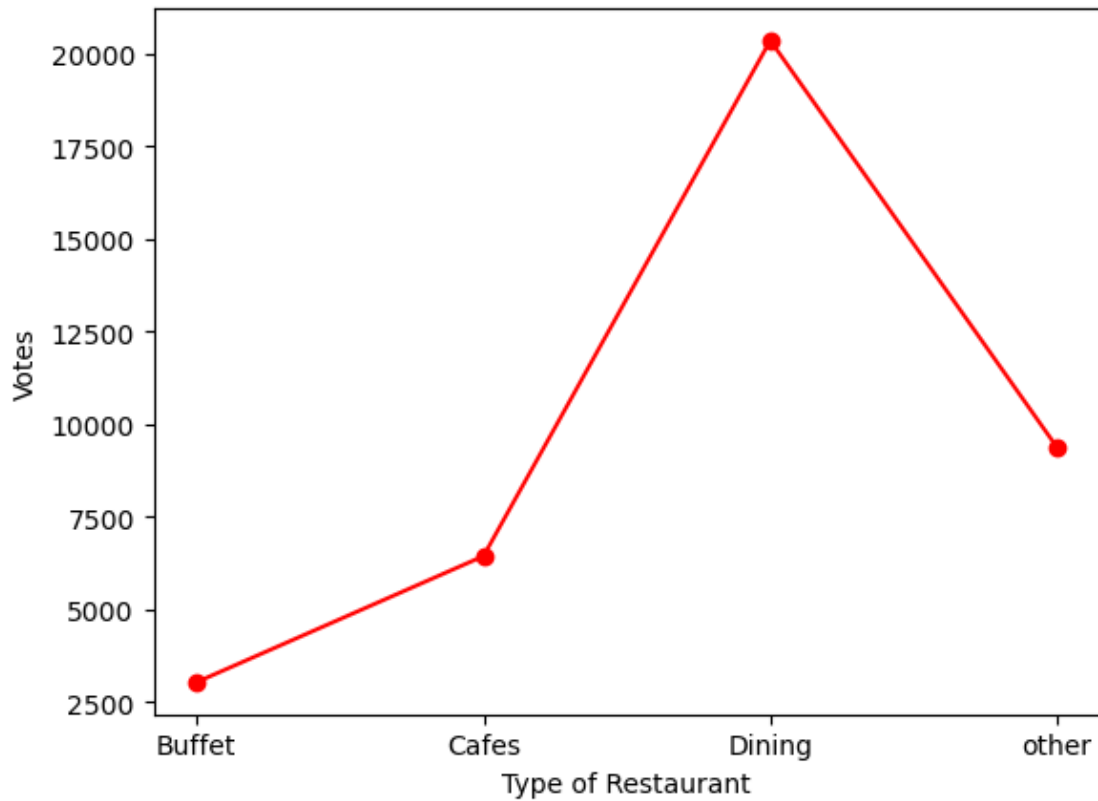
```
[10]: res_votes = pd.DataFrame(grouped_data)
      res_votes
```

```
[10]:                 votes
      listed_in(type)
      Buffet           3028
      Cafes            6434
      Dining          20363
      other            9367
```

```
[11]: plt.plot(res_votes, c='red', marker='o')
      plt.xlabel('Type of Restaurant')
      plt.ylabel('Votes')
```

```
[11]: Text(0, 0.5, 'Votes')
```

1.0.6 Insights

Dining restaurants are more voted or liked by customers

```
[12]: max_votes = df['votes'].max()
      restaurant_with_max_votes = df.loc[df['votes'] == max_votes, 'name']

      print('Restaurant(s) with the maximum votes:')
      print(restaurant_with_max_votes)
```
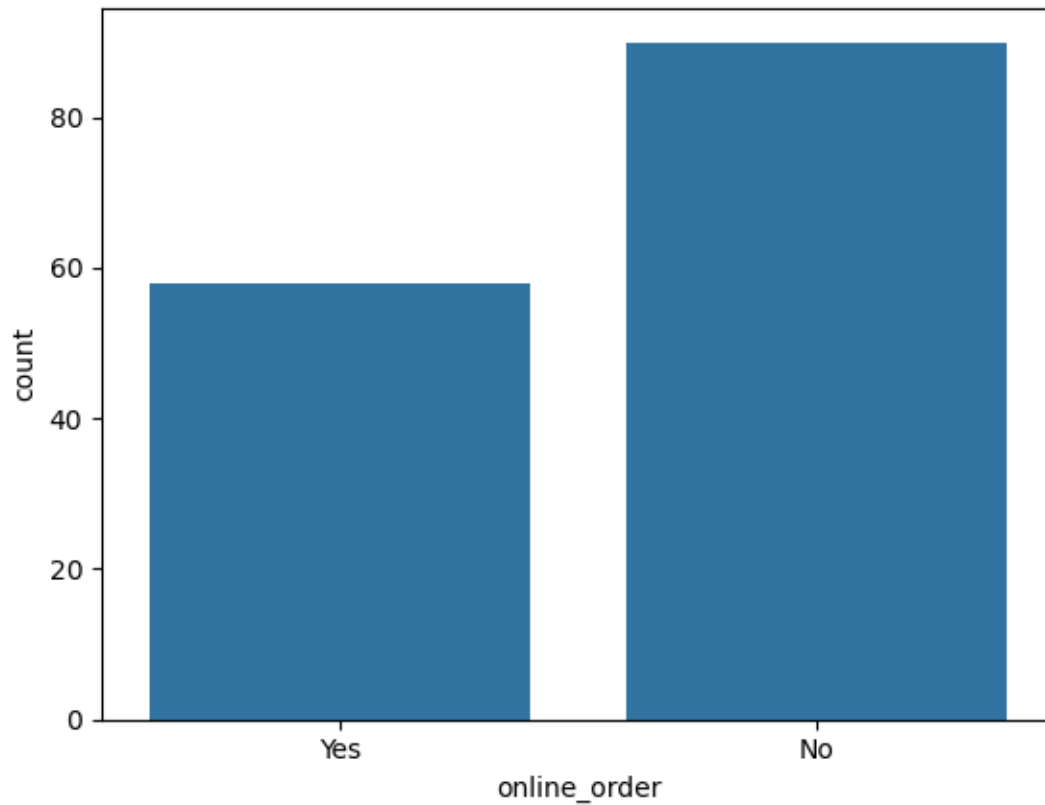
```
Restaurant(s) with the maximum votes:
38      Empire Restaurant
Name: name, dtype: object
```

1.0.7 Online Order Availability

```
[13]: sns.countplot(x=df['online_order'])
```
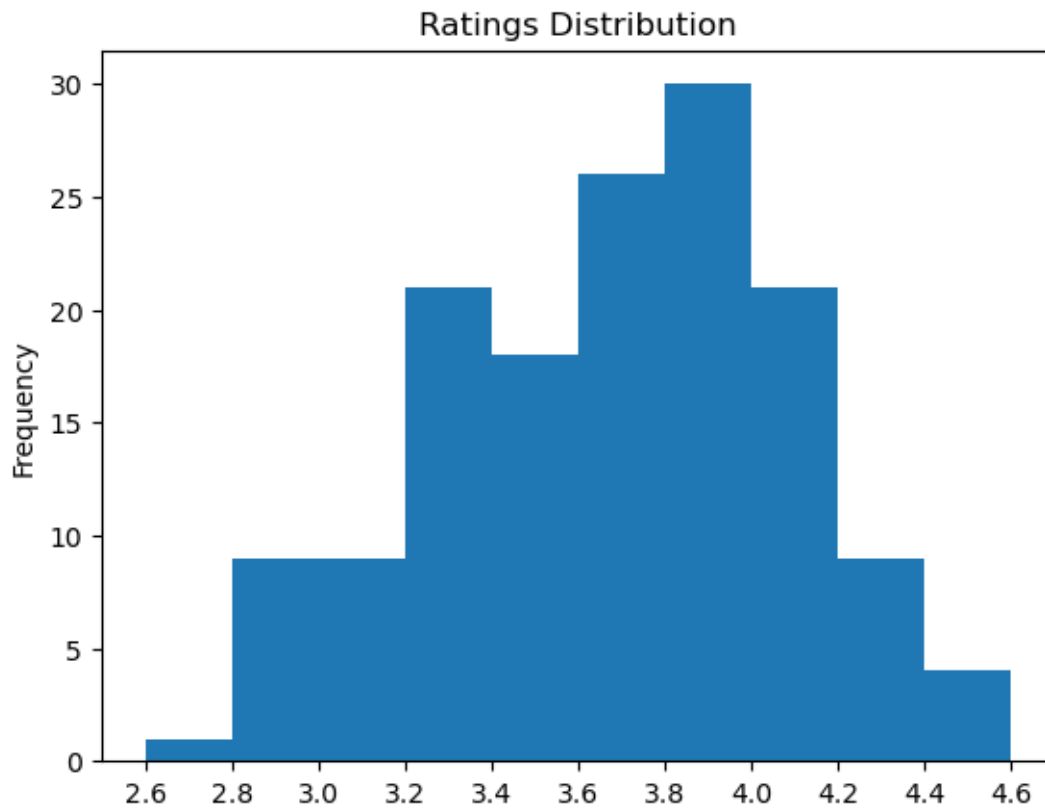
```
[13]: <Axes: xlabel='online_order', ylabel='count'>
```

1.0.8 Insights

Majority of the restaurants do not accept online orders.

1.0.9 Analyze Ratings

```
[14]: count, bin_edges = np.histogram(df['rate'])
```

```
[15]: df['rate'].plot(kind='hist', xticks=bin_edges)
      plt.title('Ratings Distribution')
      plt.show()
```
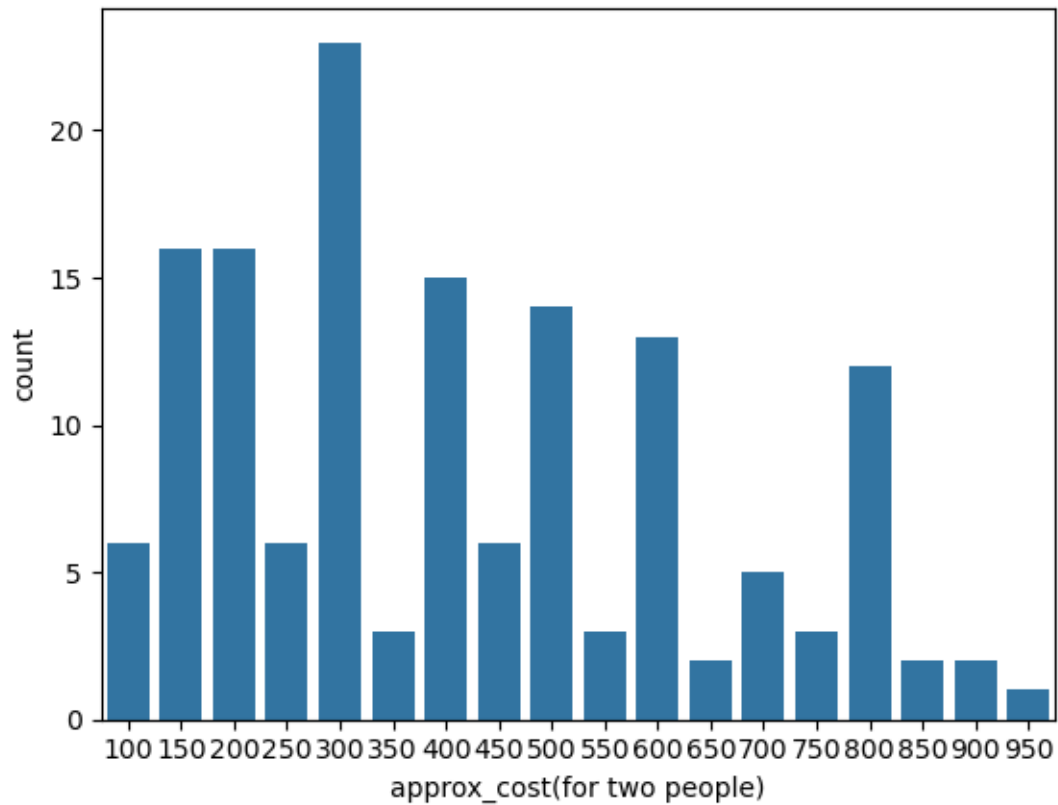
Ratings Distribution

1.0.10 Insights

The majority of restaurants received ratings ranging from 3.5 to 4

1.0.11 Approximate Cost for couples

```
[16]: couple_data=df['approx_cost(for two people)']
      sns.countplot(x=couple_data)
```
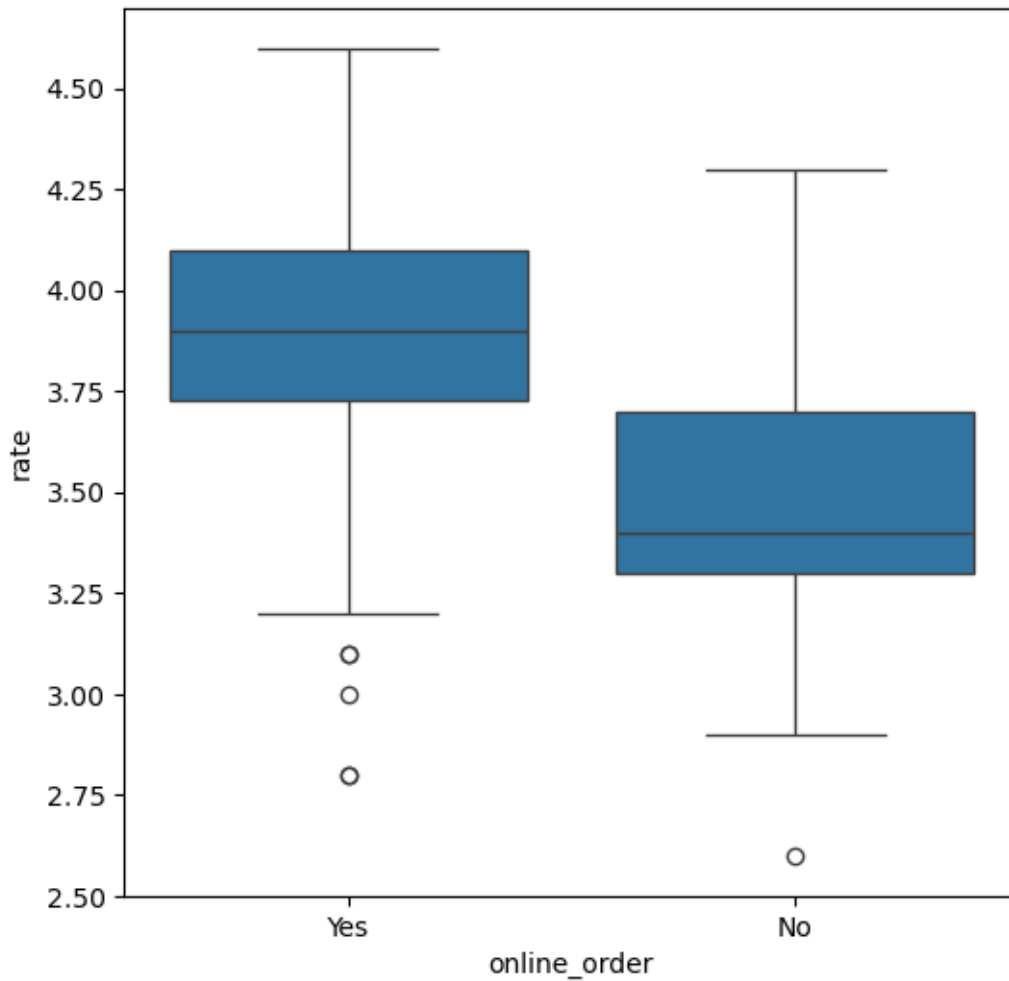
```
[16]: <Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```

1.0.12 Insight

The majority of couples prefer restaurants with approximate cost of 300 rupees.

```python
[17]: plt.figure(figsize = (6,6))
      sns.boxplot(x = 'online_order', y = 'rate', data = df)
```

```
[17]: <Axes: xlabel='online_order', ylabel='rate'>
```
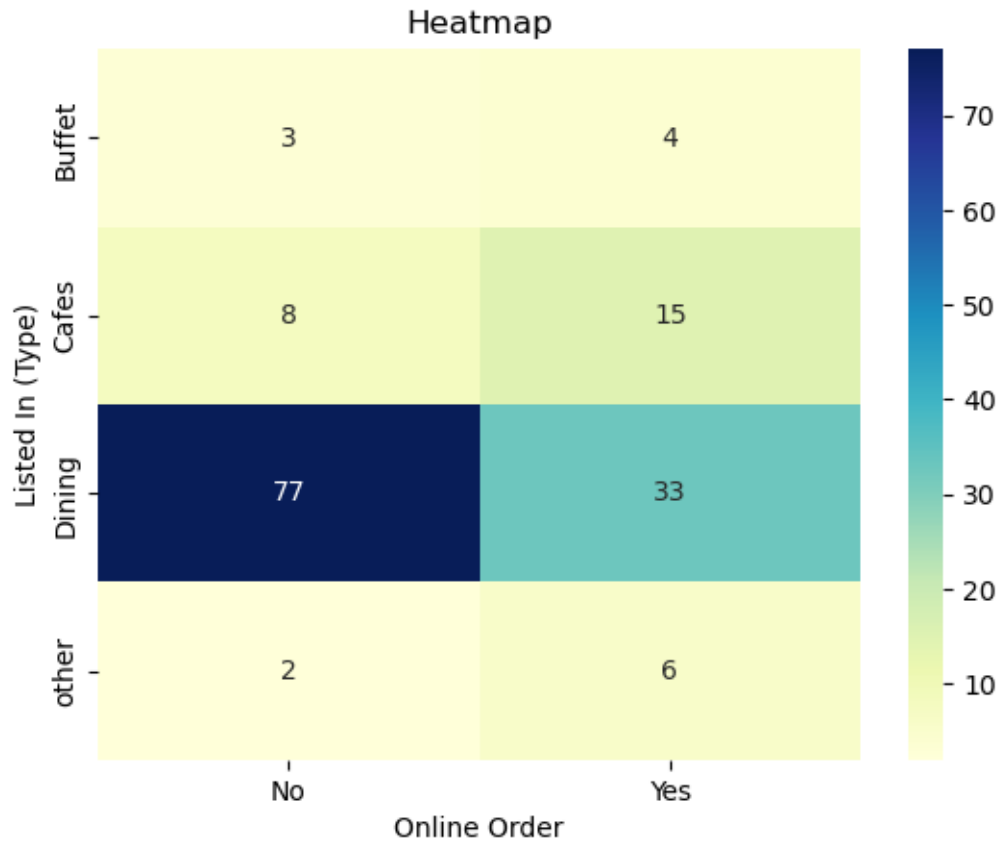
1.0.13 Insights

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

1.0.14 Order mode preferences based on Restaurant type

```
[18]: 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()
```

Heatmap

1.0.15 Insights

- Dining restaurants mainly receive offline orders.
- Cafes receive more online orders.
- Customers prefer in-person dining at restaurants and online ordering for cafes.

1.0.16 Final Conclusion

- Dining restaurants dominate the market.

- Online delivery is less common overall but popular among cafes.

- Mid-range pricing (300 rupees for two) is preferred by couples.

- Online-order restaurants tend to receive higher ratings.