CAPSTONE PROJECT

PRESENTATION

AGENDA

INTRODUCTION

UNDERSTANDING THE DATASET

BASIC STEPS TO PERFORM

DATASET PREPROCESSING

HYPOTHESIS

OBSERVATION

RECOMMENDATION

CONCLUSION

INTRODUCTION

INTRODUCTION

- In this project, we aim to analyze Zomato restaurant data to identify key factors that contribute to the success of restaurants, as measured by their ratings. By exploring various features such as location, cuisine, pricing, and service offerings, we aim to provide insights that can help restaurant owners and Zomato users make informed decisions.

- **Zomato** is a technology platform that connects customers, restaurant partners, and delivery partners, offering services like restaurant discovery, food delivery, table reservations, and more. It began as a food directory in **2008** and has since expanded into a **global online food delivery** and **restaurant aggregator**.

WHAT TO EXPECT?

- Exploratory Data Analysis (EDA) on Zomato, featuring key attributes, including restaurant names, price, rating, cuisines etc.
- Insights derived from the dataset using Python libraries and visualization tools.
- We will give the summary of data with our conclusion.

UNDERSTANDING THE DATASET

DATASET DESCRIPTION

Attributes in the dataset :

- **res_id:** specifies restaurant id
- **name**: name of restaurant
- **establishment**: type of restaurant
- **url**: link to order from zomato
- **address**: address of restaurant
- **city**
- **city_id**
- **locality**
- **latitude**: geographical latitude location
- **longitude**: geographical longitude location
- **zipcode**
- **country_id**
- **locality_verbose**: (locality,city)
- **cuisines**: types of cuisines
- **timings**: opening, closing time
- **average_cost_for_two:** cost for 2 people
- **price_range:** 1= cheapest, 4= most expensive
- **currency**
- **highlights:** facilities offered by restaurants
- **aggregate_rating:** rating between (1-5)
- **rating_text:** good, very good, average, excellent,poor, etc.
- **votes**
- **photo_count**
- **opentable_support:** restaurant allows table booking (0: not supported, 1: supported)
- **delivery**: online booking and delivery (0: not suppported, 1: supported)
- **takeaway**: (0: not supported, 1: supported, -1: not available)

BASIC STEPS TO PERFORM

1.Import Python Library:

- We import the libraries first to use the functions of that libraries.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

2. Loading the dataset:

- Load the dataset using panda library on which we will do our analysis.

```
zomato = pd.read_csv("Indian-Resturants.csv")
```

zomato

3. Describing Data:

- We use .describe() function to give the mean, min, max, std, count etc.

```
zomato.describe()
```

	res_id	city_id	latitude	longitude	country_id	average_cost_for_two
count	2.119440e+05	211944.000000	211944.000000	211944.000000	211944.0	211944.000000
mean	1.349411e+07	4746.785434	21.499758	77.615276	1.0	595.812229
std	7.883722e+06	5568.766386	22.781331	7.500104	0.0	606.239363
min	5.000000e+01	1.000000	0.000000	0.000000	1.0	0.000000
25%	3.301027e+06	11.000000	15.496071	74.877961	1.0	250.000000
50%	1.869573e+07	34.000000	22.514494	77.425971	1.0	400.000000
75%	1.881297e+07	11306.000000	26.841667	80.219323	1.0	700.000000
max	1.915979e+07	11354.000000	10000.000000	91.832769	1.0	30000.000000

4. Data Information:

- We use .info() function it give the datatype of the given columns on dataset.

```
[5]: zomato.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   res_id                211944 non-null  int64
 1   name                  211944 non-null  object
 2   establishment         211944 non-null  object
 3   url                   211944 non-null  object
 4   address               211810 non-null  object
 5   city                  211944 non-null  object
 6   city_id               211944 non-null  int64
```

zomato

3. Describing Data:

- We use .describe() function to give the mean, min, max, std, count etc.

```
zomato.describe()
```

	res_id	city_id	latitude	longitude	country_id	average_cost_for_two
count	2.119440e+05	211944.000000	211944.000000	211944.000000	211944.0	211944.000000
mean	1.349411e+07	4746.785434	21.499758	77.615276	1.0	595.812229
std	7.883722e+06	5568.766386	22.781331	7.500104	0.0	606.239363
min	5.000000e+01	1.000000	0.000000	0.000000	1.0	0.000000
25%	3.301027e+06	11.000000	15.496071	74.877961	1.0	250.000000
50%	1.869573e+07	34.000000	22.514494	77.425971	1.0	400.000000
75%	1.881297e+07	11306.000000	26.841667	80.219323	1.0	700.000000
max	1.915979e+07	11354.000000	10000.000000	91.832769	1.0	30000.000000

4. Data Information:

- We use .info() function it give the datatype of the given columns on dataset.

```
[5]: zomato.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   res_id                211944 non-null  int64
 1   name                  211944 non-null  object
 2   establishment         211944 non-null  object
 3   url                   211944 non-null  object
 4   address               211810 non-null  object
 5   city                  211944 non-null  object
 6   city_id               211944 non-null  int64
```

zomato

DATASET PREPROCESSING

zomato

Data Preprocessing includes :-

- Handling missing values
- Remove duplicates if any
- Reducing data
- Detecting and handling outliers

HANDLING MISSING VALUES:-

- Before handling missing values we find missing values first.

```
zomato.isnull().sum()/len(zomato)*100
```

```
[8]:  res_id                0.000000
      name                  0.000000
      establishment         0.000000
      url                   0.000000
      address               0.063224
      city                  0.000000
      city_id               0.000000
      locality              0.000000
      latitude              0.000000
      longitude             0.000000
      zipcode              76.995338
      country_id            0.000000
```

```
## Handling the object missing values
for k in zomato.select_dtypes(["object"]):
    zomato[k] = zomato[k].fillna(zomato[k].mode()[0])
```

- After finding missing values we handle those missing values.

```
res_id                0.0
name                  0.0
establishment         0.0
url                   0.0
address               0.0
city                  0.0
city_id               0.0
locality              0.0
latitude              0.0
longitude             0.0
country_id            0.0
locality_verbose      0.0
```

REDUCING VALUES:-

- Reducing the dataset is also a very important step to do in data preprocessing .
- As the missing value percentage of “zipcode” is more than 25% so if the missing values are more than 25% than we can drop/ delete that data.
- After dropping the “zipcode column” from the data.

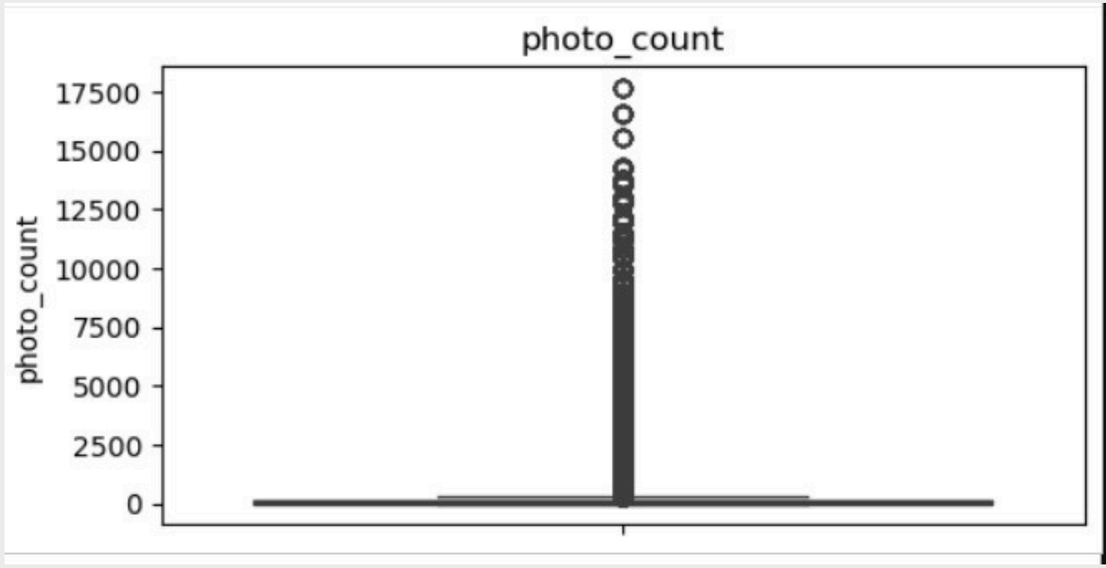
```
8]:  res_id                0.000000
     name                  0.000000
     establishment         0.000000
     url                   0.000000
     address               0.063224
     city                  0.000000
     city_id               0.000000
     locality              0.000000
     latitude              0.000000
     longitude             0.000000
     zipcode              76.995338
     country_id            0.000000
     locality_verbose      0.000000
     cuisines              0.656305
```

```
# As we can see the missing value percentage  is
zomato.drop("zipcode", axis = 1, inplace= True)
```

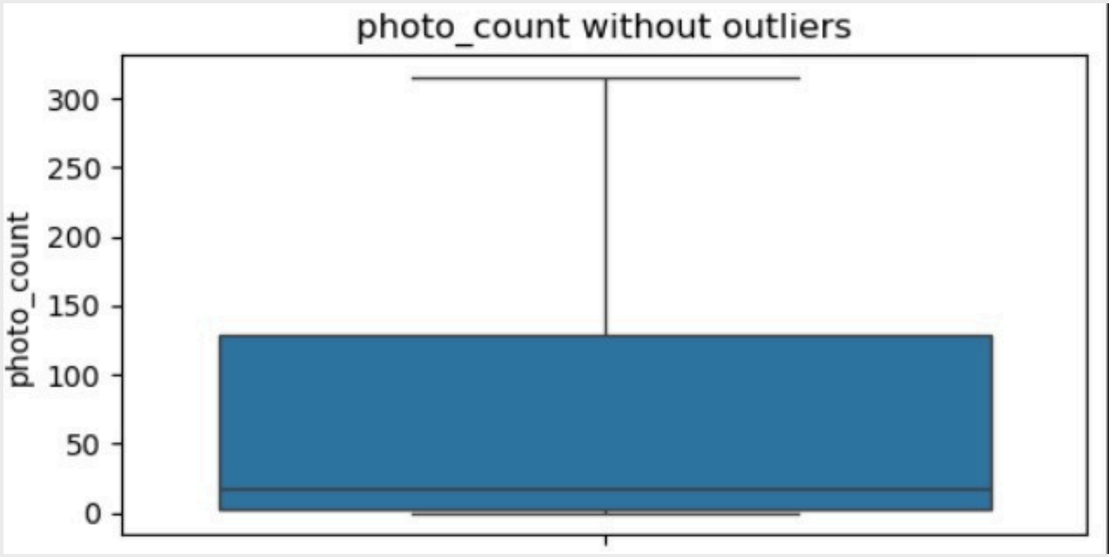
```
res_id                0.0
name                  0.0
establishment         0.0
url                   0.0
address               0.0
city                  0.0
city_id               0.0
locality              0.0
latitude              0.0
longitude             0.0
country_id            0.0
locality_verbose      0.0
cuisines              0.0
```

HANDLING OUTLIERS

- Handling outliers of dataset is another important aspect of data preprocessing as it helps to manage the outliers.



- After handling the outliers this is the result.



Handling outliers using IQR method

- It is defined as the difference between the 75th and 25th percentiles of the data.

- IQR=Q3−Q1 this formula is used to calculate IQR.

```
## using Inter Quartile Method (IQR) method

outlier_list1 = [ "average_cost_for_two", "price_range","votes", "photo_count","aggregate_rating"]

for j in outlier_list1:
    q1 = zomato[j].quantile(0.25)
    q3 = zomato[j].quantile(0.75)
    iqr = q3 - q1

    print("IQR:", iqr)
    print()
    print("Q1:", q1)
    print()
    print("Q2:", q3)
    print()

    ul = q3 + 1.5 * iqr
    ll = q1 - 1.5 * iqr
    print(ul)
    print(ll)

    ## setting the outlier values
    zomato[j] = np.where(zomato[j]>ul,ul,
                         np.where(zomato[j]<ll,ll,
                                  zomato[j]))
```

```
IQR: 346.0

Q1: 16.0

Q2: 362.0

881.0
-503.0
```

```
IQR: 450.0

Q1: 250.0

Q2: 700.0

1375.0
-425.0
```

```
IQR: 125.0

Q1: 3.0

Q2: 128.0

315.5
-184.5
```

```
IQR: 1.0

Q1: 1.0

Q2: 2.0

3.5
-0.5
```

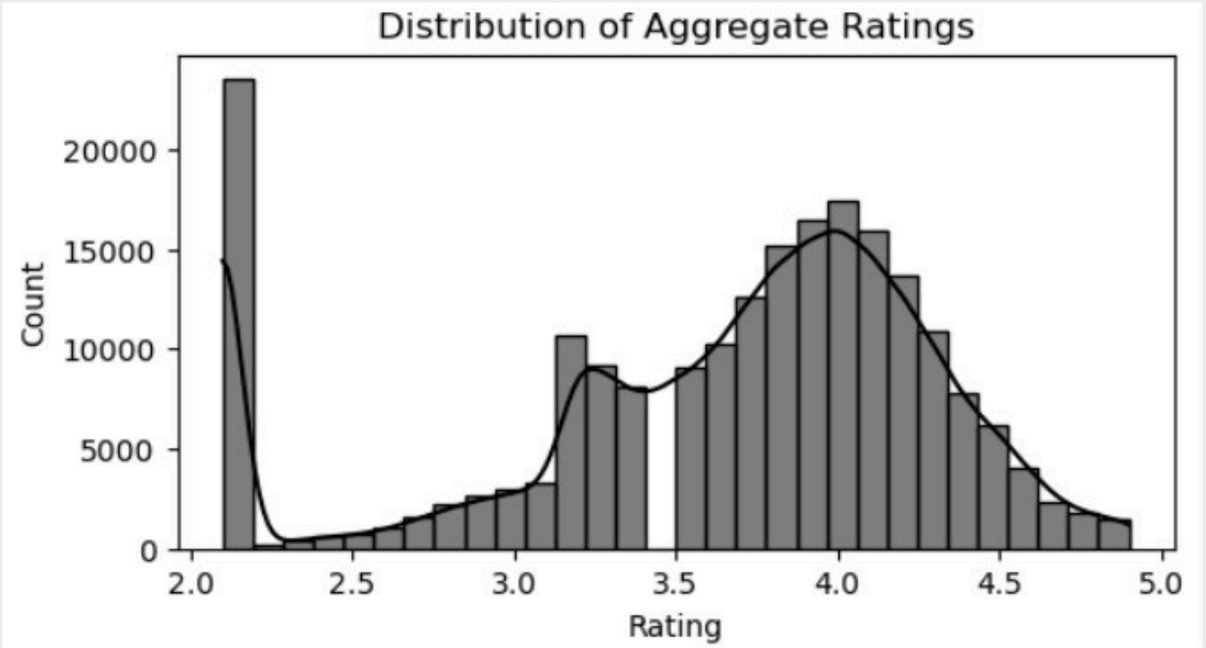
HYPOTHESIS

HYPOTHESIS- 1

1. Calculate and visualize the average rating of restaurant.
2. Analyze and distribution of restaurant rating and understand the overall rating

```python
## Calculating the average rating of restaurants
average_rating = zomato["aggregate_rating"].mean()

## Showing the distribution of rating through the restaurants
plt.figure(figsize = (8,4))
plt.title("Distribution of restaurant rating")
plt.xlabel("Aggregate Rating")
plt.ylabel("Restaurants")
sns.histplot(zomato, x = "aggregate_rating", bins = 25, kde = True, color = "orange")
plt.axvline(average_rating, color='Green', label = f"Average Rating is {average_rating:.2f}")
plt.grid(visible = True)
plt.legend()
plt.show()
```

```python
## Analyze the distribution of restaurant ratings to understand the overall rating
plt.figure(figsize = (6,3))
sns.histplot(zomato['aggregate_rating'], bins=30,  kde=True, color = "black")
plt.title('Distribution of Aggregate Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```



HYPOTHESIS- 2

1. Identify the city with the highest concentration of restaurants.

- To calculate the highest concentration we have to choose the city column and simply count the number of appearances of city.

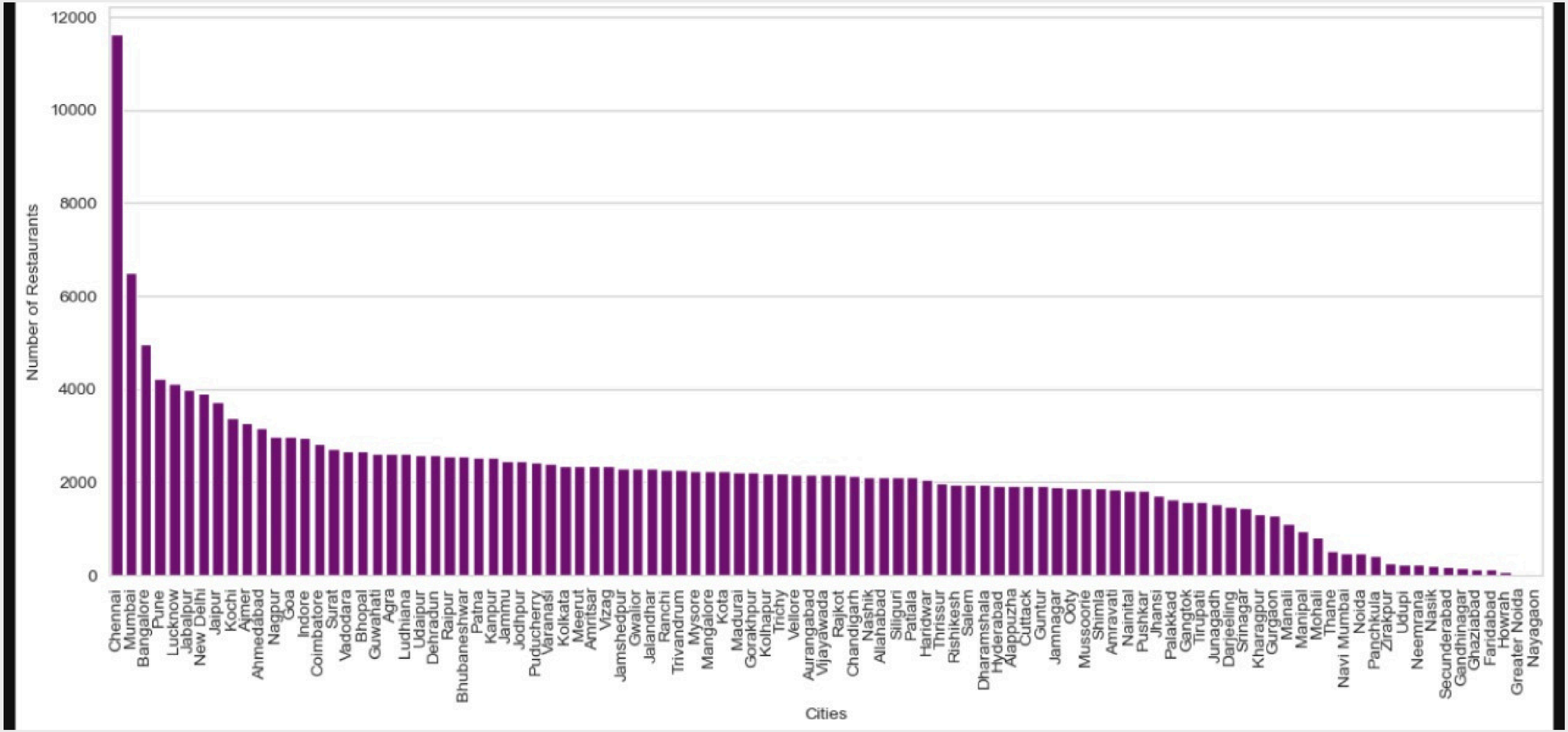
```
plt.figure(figsize = (12,8))

## convert series to dataframe
city_count = zomato["city"].value_counts().reset_index()
city_count.columns = ["city", "restaurant_count"]


## visualization of restaurant count
sns.barplot(data = city_count, x = "city", y = "restaurant_count", color = "purple")
plt.xticks(rotation = 90)
plt.xlabel("Cities")
plt.ylabel("Number of Restaurants")
plt.show()
```

- These are top 5 cities with highest concentration of restaurant and lowest on the other side.

city	count
Chennai	11630
Mumbai	6497
Bangalore	4971
Pune	4217
Lucknow	4121

Ghaziabad	132
Faridabad	124
Howrah	66
Greater Noida	33
Nayagaon	17

- Here is the BAR GRAPH of cities with the highest concentration of restaurants.



2. Distribution of Restaurants Ratings Across Cities.

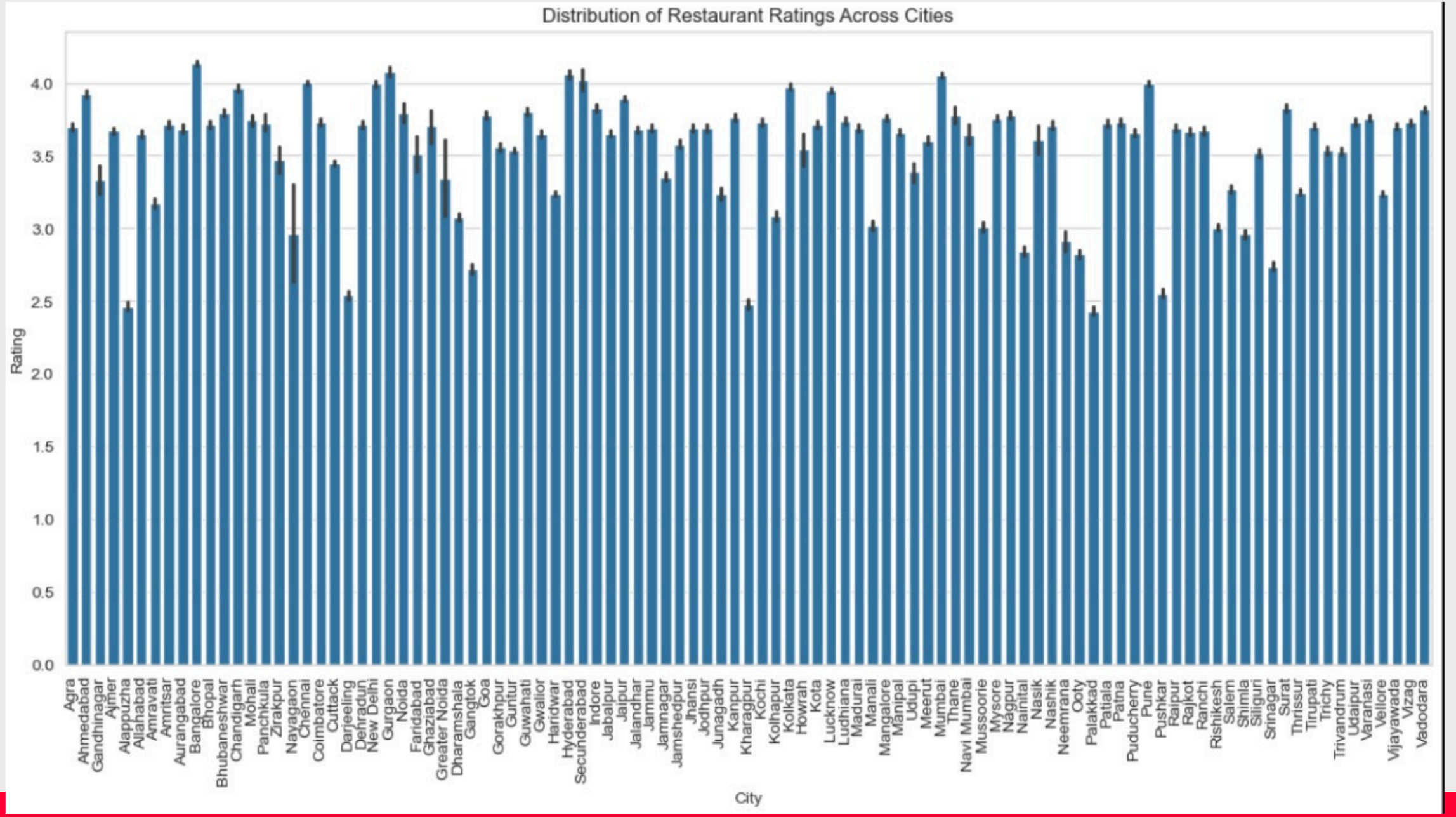
- For Rating vs city we have to work with two data column which are city and aggregate_rating

```python
# Create the plot
plt.figure(figsize=(15, 7))
sns.barplot(x='city', y='aggregate_rating', data=zomato)
plt.xticks(rotation=90)
plt.title('Distribution of Restaurant Ratings Across Cities')
plt.xlabel('City')
plt.ylabel('Rating')
plt.show()
```

T
aggregate_rating
4.4
4.4
4.2
4.3
4.9
4
4.2
3.8

F
city
Agra
Agra
Agra
Agra
Agra
Agra
Agra

- Here is the BAR GRAPH of cities with ratings



HYPOTHESIS- 3

1. Price range vs Restaurant rating

```
## Analyzing the relationship between price range and restaurant ratings.
grouped_series = zomato.groupby("price_range")["aggregate_rating"].mean().reset_index()
grouped_series
grouped_df = pd.DataFrame(grouped_series)
grouped_df
```

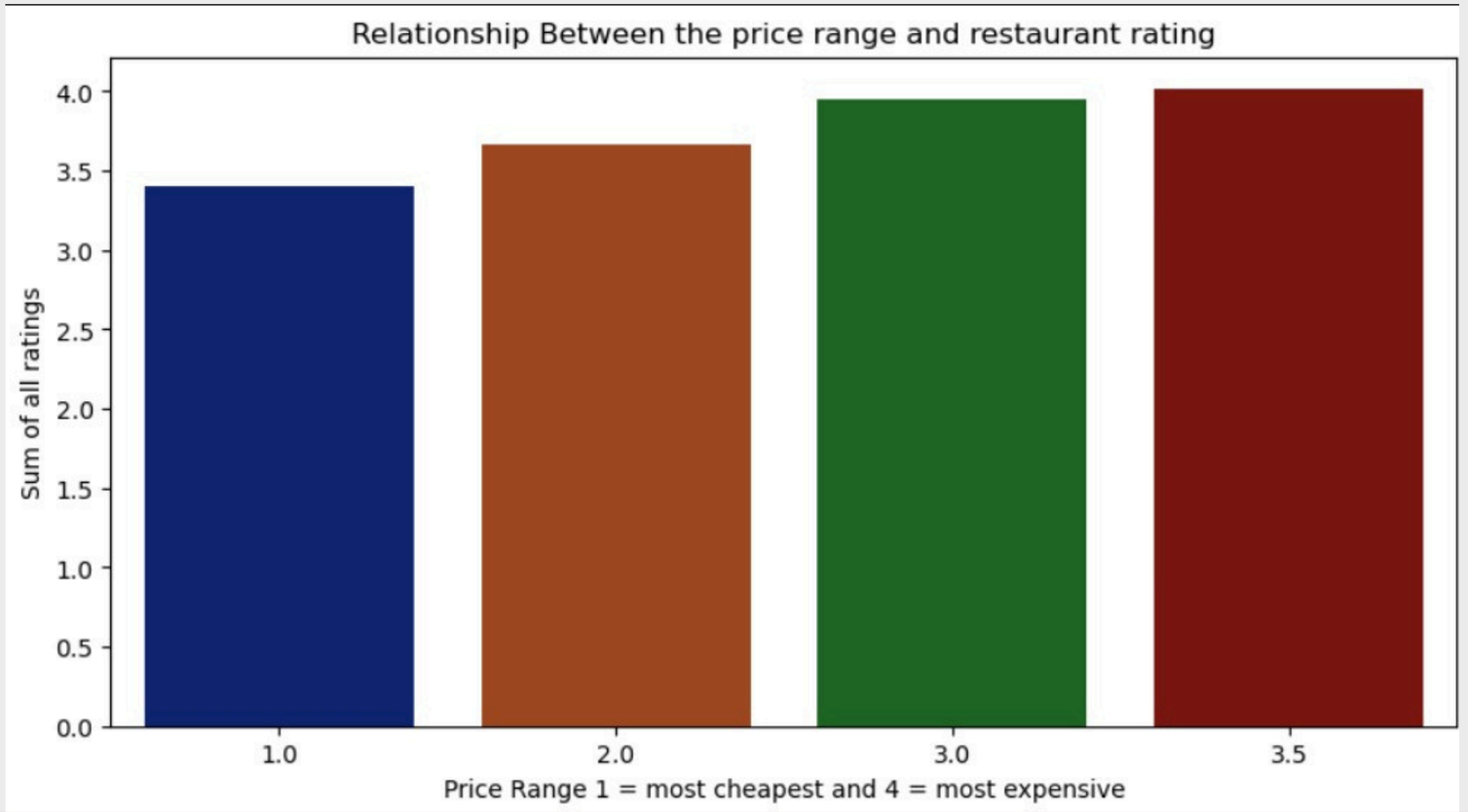
- Restaurant rating from 1 being the least and 5 being the highest.

- Price Range from 1 being the cheapest and 5 being the most expensive.

	price_range	aggregate_rating
0	1.0	3.405808
1	2.0	3.663985
2	3.0	3.950096
3	3.5	4.015623

2. Visualize the average cost for two people in different price categories.

```
plt.figure(figsize = (10,5))
sns.barplot(data= grouped_df, x="price_range",y="aggregate_rating", palette='dark')
plt.xlabel("Price Range 1 = most cheapest and 4 = most expensive")
plt.ylabel("Sum of all ratings")
plt.title("Relationship Between the price range and restaurant rating")
plt.show()
```



HYPOTHESIS- 4

1. Identify and visualize the top restaurant chains based on number of outlets.

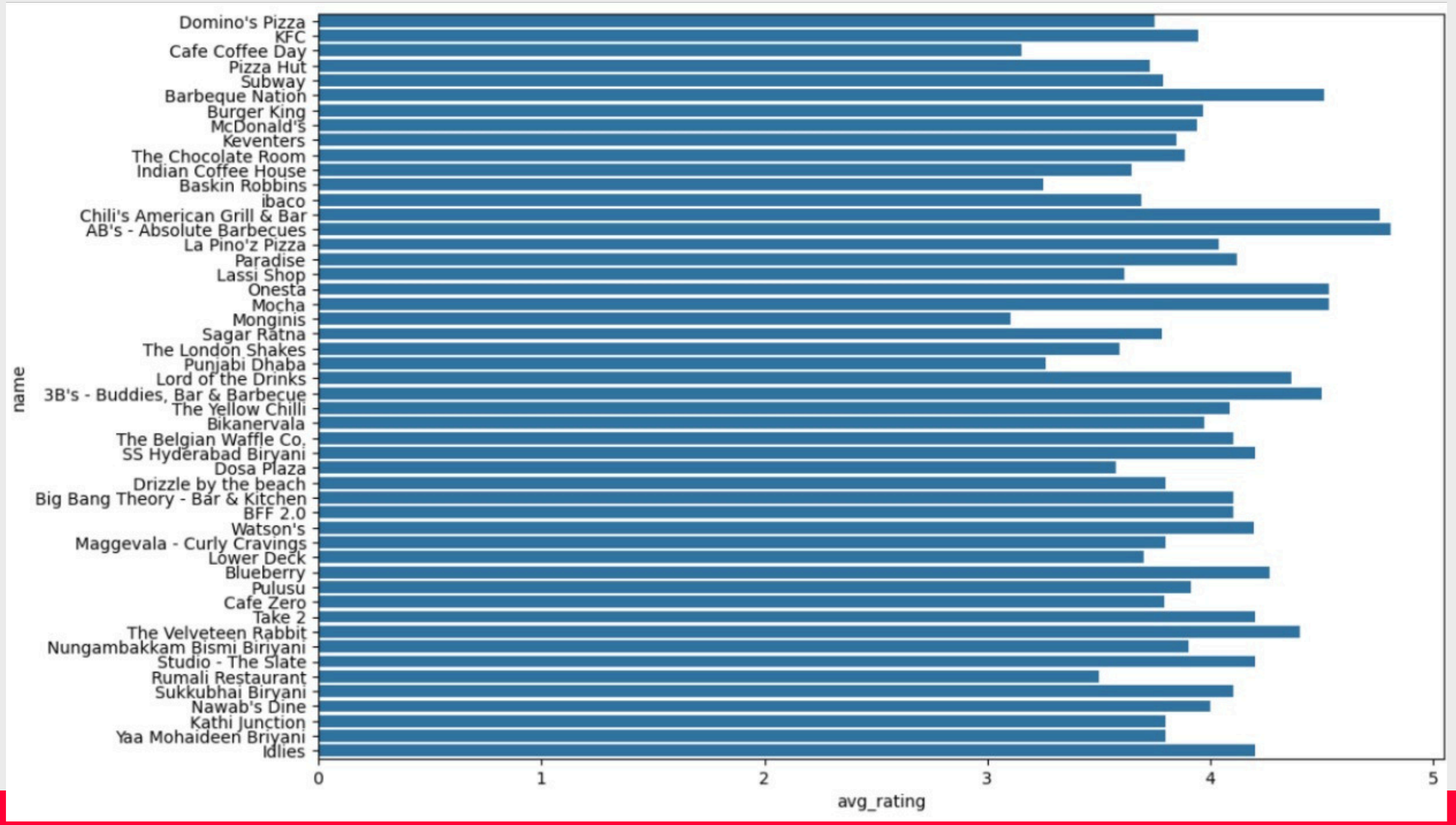
- These are highest outlets and then lowest in the top 50 next to it.

```
## Identify and visualize the top restaurant chains based on the number of outlets

chain_count = zomato["name"].value_counts().head(50)

chain_count_df = pd.DataFrame(chain_count)
chain_count_df
```

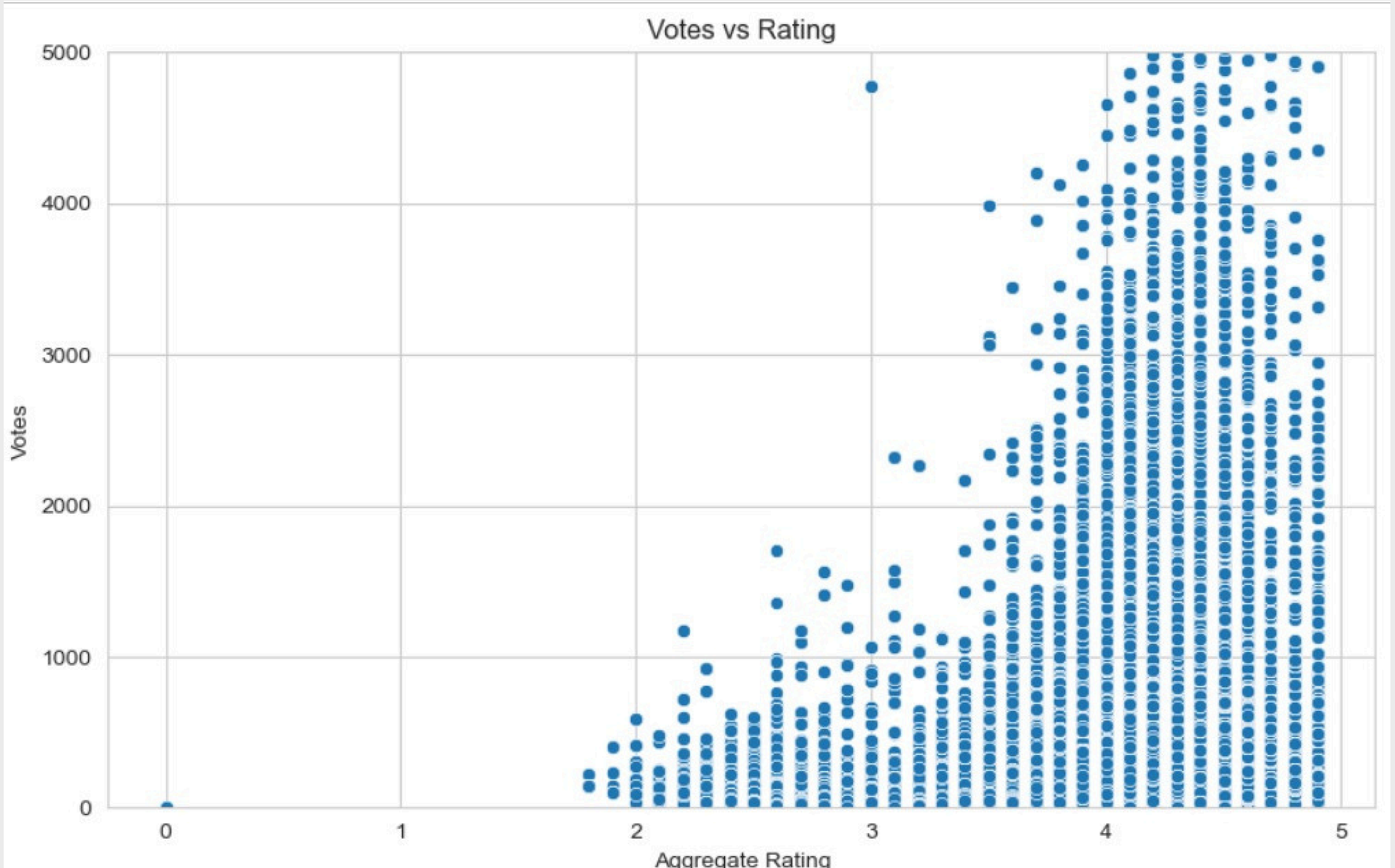
name	count
Domino's Pizza	3108
KFC	1343
Cafe Coffee Day	1068
Pizza Hut	936
Subway	766
Barbeque Nation	725
Burger King	658
McDonald's	578
Keventers	512

Nungambakkam Bismi Biriyani	149
Studio - The Slate	148
Kathi Junction	147
Rumali Restaurant	147
Sukkubhai Biryani	147
Nawab's Dine	147
Yaa Mohaideen Briyani	147
Loretta's Shakes & Screams	145

- Here is the visualization of top 50 chain restaurant.



HYPOTHESIS- 5 Votes vs Rating



OBSERVATION:

- As per the data the high price cuisine or restaurant is highly rated between (3.5 to 4.5) and for low price cuisine or restaurant is low rated between (2 to 3).
- Takeaway have a lot of missing information which means that the people are not very interested in takeaway orders.
- Most of the restaurants have the rating between 3 to 4.5.
- There are very few restaurants who are offering free Wifi

zomato

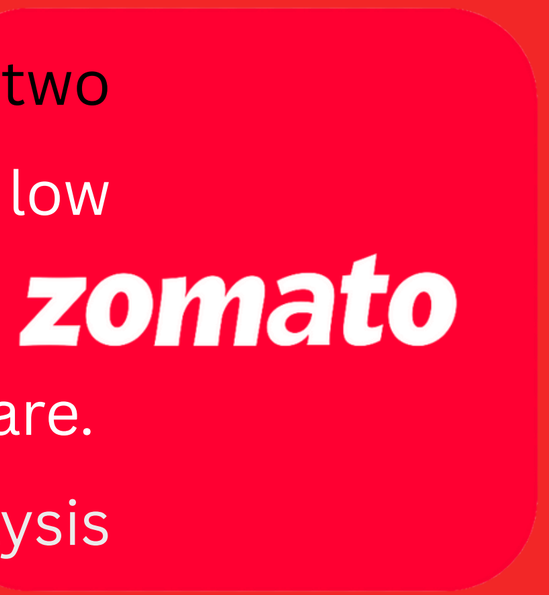
RECOMMENDATION:

- Restaurants should focus on the medium price range with high number of cuisines.
- Restaurant owners should change the cuisine menu according to the city or state.
- As per the data the people are more interested in online ordering the food and get it delivered to their home but as per the data takeaways have many missing values as which needs to be filled with either 0 or 1.

zomato

Conclusion:

- The analysis shows that customers have a mix opinion about the services like for takeaways people are less responsive but on the other hand people for delivery are very much satisfied.
- The other thing which effect the sales will be the price for two section on which some restuarants are doing great with their low pricing.
- There was also a huge impact of cities in which the restaurants are.
- There should be a good preprocessing and very precise analysis which make things easier for future reference.



Thank you