zomato-data-analysis

July 21, 2023

1 Zomato Data Analysis

1.1 About Dataset

This dataset provides a comprehensive view of the restaurant scene in the 13 metropolitan areas of India(900 restaurants) . Researchers, analysts, and food enthusiasts can use this dataset to gain insights into various aspects such as dining and delivery ratings, customer reviews and preferences, popular cuisines, best-selling items, and pricing information across different cities. It enables the exploration of dining patterns, the comparison of restaurants and cuisines between cities, and the identification of trends in the food industry. This dataset serves as a valuable resource for understanding the culinary landscape and making data-driven decisions related to the restaurant business, customer satisfaction, and food choices in these metropolitan areas of India. In this dataset, we have more than 127000 rows and 12 columns, a fairly large dataset. You will be able to get hands-on experience while performing the following tasks and will be able to understand how real-world problem statement analysis is done. In Data Analysis what all things we do

Handling Missing Values Explore numerical features. Explore categorical features. Finding relations between features. You have to perform the following tasks:

Explore the Data read the dataset understand each feature and write down the details. explore the dataset info, describe and find columns with categories, and numeric columns as well. Data Cleaning:

Deleting redundant columns. Renaming the columns. Dropping duplicates. Cleaning individual columns. Remove the NaN values from the dataset Check for some more Transformations Data Visualization:

1.1.1 find out few insights from dataset

What is the average dining rating across all restaurants in the dataset Which metropolitan area has the highest average delivery rating What is the total number of dining votes received by all restaurants in each city How many unique cuisines are represented in the dataset Which restaurant has the highest average dining rating in each city

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

[2]:
```
df = pd.read_csv('zomato_dataset.csv')
df.head()
```

[2]:

	Restaurant Name	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Cuisine	Place Name	City	Item Name	Best Seller	Votes	Prices
0	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Platter Kebab Combo	BESTSELLER	84	249.0
1	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Chicken Rumali Shawarma	BESTSELLER	45	129.0
2	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Chicken Tandoori Salad	NaN	39	189.0
3	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Chicken BBQ Salad	BESTSELLER	43	189.0
4	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Special Doner Wrap Combo	MUST TRY	31	205.0

[3]:
```
df.shape
```

[3]:
```
(123657, 12)
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 123657 entries, 0 to 123656
Data columns (total 12 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Restaurant Name  123657 non-null  object
 1   Dining Rating    91421 non-null   float64
 2   Delivery Rating  122377 non-null  float64
 3   Dining Votes     123657 non-null  int64
 4   Delivery Votes   123657 non-null  int64
 5   Cuisine          123657 non-null  object
 6   Place Name       123657 non-null  object
 7   City             123657 non-null  object
 8   Item Name        123657 non-null  object
 9   Best Seller      27942 non-null   object
 10  Votes            123657 non-null  int64
 11  Prices           123657 non-null  float64
```

```
dtypes: float64(3), int64(3), object(6)
memory usage: 11.3+ MB
```

[5]:
```
df.isnull().sum()
```

[5]:
```
Restaurant Name        0
Dining Rating      32236
Delivery Rating     1280
Dining Votes           0
Delivery Votes         0
Cuisine                0
Place Name             0
City                   0
Item Name              0
Best Seller        95715
Votes                  0
Prices                 0
dtype: int64
```

[6]:
```
df.drop(['Best Seller'],inplace= True, axis=1)
```

[7]:
```
mean = np.mean(df['Dining Rating'])
mean
```

[7]:
```
3.822264031240087
```

[8]:
```
df['Dining Rating']= df['Dining Rating'].replace(np.nan,3.8)
```

[9]:
```
df.isnull().sum()
```

[9]:
```
Restaurant Name       0
Dining Rating         0
Delivery Rating    1280
Dining Votes          0
Delivery Votes        0
Cuisine               0
Place Name            0
City                  0
Item Name             0
Votes                 0
Prices                0
dtype: int64
```

[10]:
```
print(np.mean(df['Delivery Rating']))
```

```
3.9631842584799433
```

[11]:
```
df['Delivery Rating']= df['Delivery Rating'].replace(np.nan,3.9)
```

[12]:
```
df.describe()
```

[12]:

	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	\
count	123657.000000	123657.000000	123657.000000	123657.000000	
mean	3.816460	3.962530	152.729858	115.763725	
std	0.351543	0.244708	232.214061	243.970828	
min	2.500000	2.500000	0.000000	0.000000	
25%	3.700000	3.800000	0.000000	0.000000	
50%	3.800000	4.000000	30.000000	0.000000	
75%	4.000000	4.100000	217.000000	23.000000	
max	4.800000	4.600000	997.000000	983.000000	

	Votes	Prices
count	123657.000000	123657.000000
mean	24.666772	241.378399
std	125.236009	192.830713
min	0.000000	0.950000
25%	0.000000	130.000000
50%	0.000000	208.570000
75%	15.000000	299.000000
max	9750.000000	12024.000000

[13]:
```
df.duplicated().sum()
```

[13]:
```
26322
```

[14]:
```
df=df.drop_duplicates()
```

[15]:
```
df.duplicated().sum()
```

[15]:
```
0
```

[16]:
```
df.describe()
```

[16]:

	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	\
count	97335.000000	97335.000000	97335.000000	97335.000000	
mean	3.815949	3.958819	152.631345	115.579771	
std	0.350789	0.244124	231.127900	242.644336	
min	2.500000	2.500000	0.000000	0.000000	
25%	3.700000	3.800000	0.000000	0.000000	
50%	3.800000	4.000000	30.000000	0.000000	
75%	4.000000	4.100000	221.000000	32.000000	
max	4.800000	4.600000	997.000000	983.000000	

	Votes	Prices
count	97335.000000	97335.000000
mean	16.701998	244.016323

```
std       98.271749    198.468133
min        0.000000      0.950000
25%        0.000000    130.000000
50%        0.000000    209.000000
75%        9.000000    299.000000
max     9750.000000  12024.000000
```

[17]:
```
df.corr()
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_16736\1134722465.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  df.corr()
```

[17]:

	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	\
Dining Rating	1.000000	0.262485	0.235939	-0.112168	
Delivery Rating	0.262485	1.000000	0.143883	-0.063411	
Dining Votes	0.235939	0.143883	1.000000	-0.246941	
Delivery Votes	-0.112168	-0.063411	-0.246941	1.000000	
Votes	0.034723	0.043759	0.004984	-0.054184	
Prices	0.058239	0.053642	0.016352	0.012276	

	Votes	Prices
Dining Rating	0.034723	0.058239
Delivery Rating	0.043759	0.053642
Dining Votes	0.004984	0.016352
Delivery Votes	-0.054184	0.012276
Votes	1.000000	-0.053287
Prices	-0.053287	1.000000

[18]:
```
df.columns
```

[18]:
```
Index(['Restaurant Name', 'Dining Rating', 'Delivery Rating', 'Dining Votes',
       'Delivery Votes', 'Cuisine ', 'Place Name', 'City', 'Item Name',
       'Votes', 'Prices'],
      dtype='object')
```

[19]:
```
df.nunique()
```

[19]:
```
Restaurant Name      826
Dining Rating         24
Delivery Rating       18
Dining Votes         294
Delivery Votes       263
Cuisine               48
Place Name           324
```

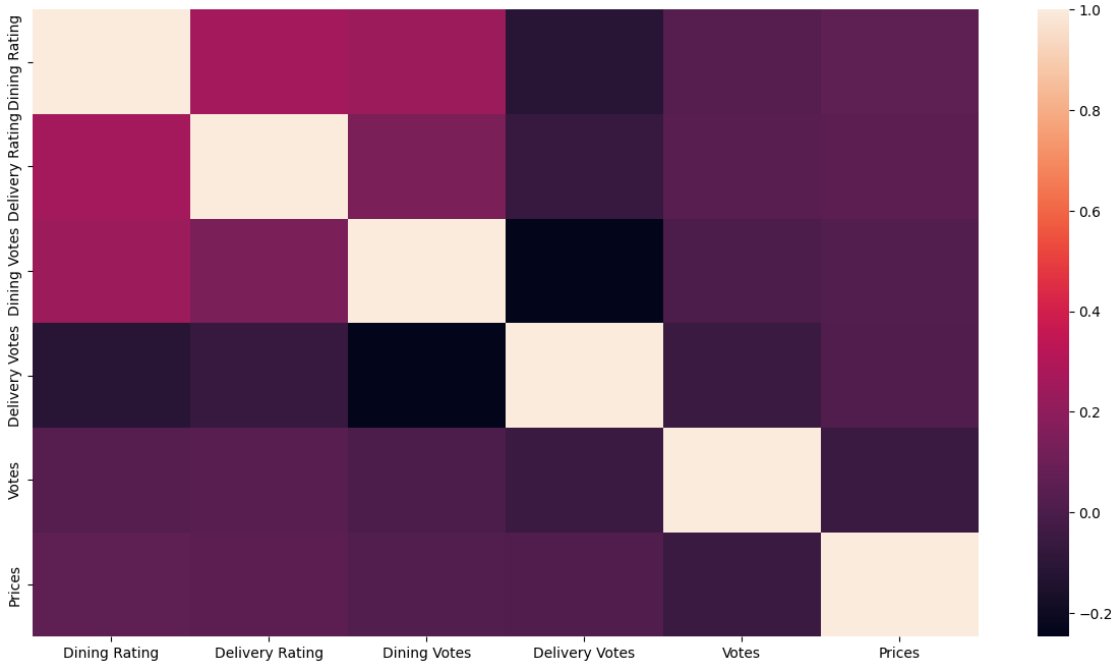
```
City                 17
Item Name         55693
Votes               760
Prices             2710
dtype: int64
```

[20]:
```
df['City'].unique()
```

[20]:
```
array([' Hyderabad', ' Mumbai', ' Chennai', ' Pune', ' Jaipur', ' Kochi',
       ' Goa', ' Bangalore', ' Kolkata', ' Ahmedabad', ' Banaswadi',
       ' Ulsoor', ' Malleshwaram', ' Magrath Road', ' Lucknow',
       ' New Delhi', ' Raipur'], dtype=object)
```

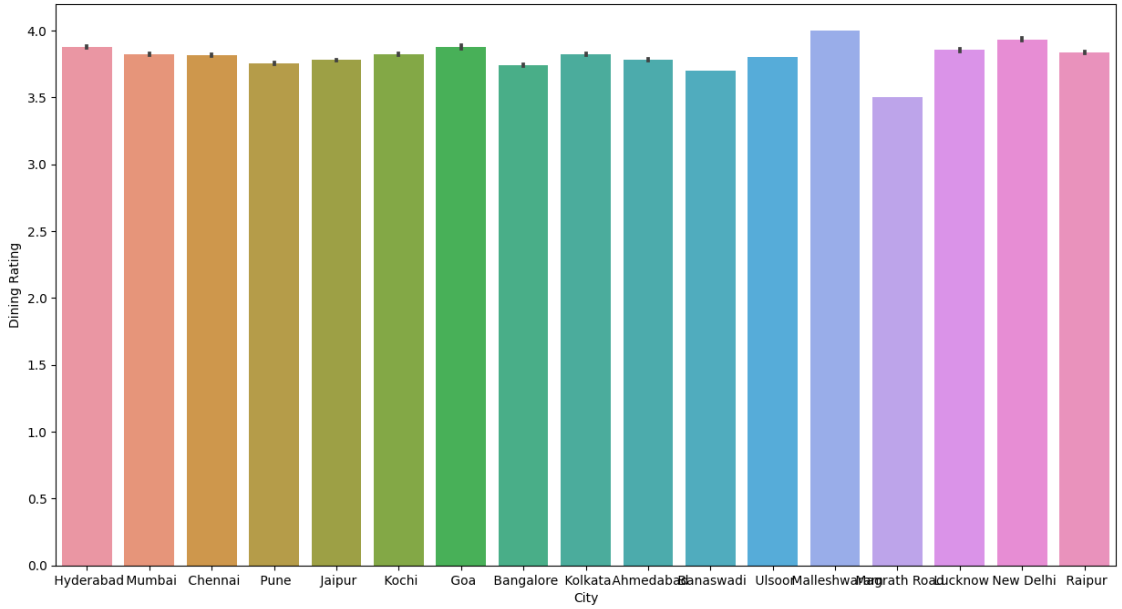
[21]:
```
plt.figure(figsize=(15,8))
sns.heatmap(df.corr())
plt.show()
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_16736\1966855796.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  sns.heatmap(df.corr())
```

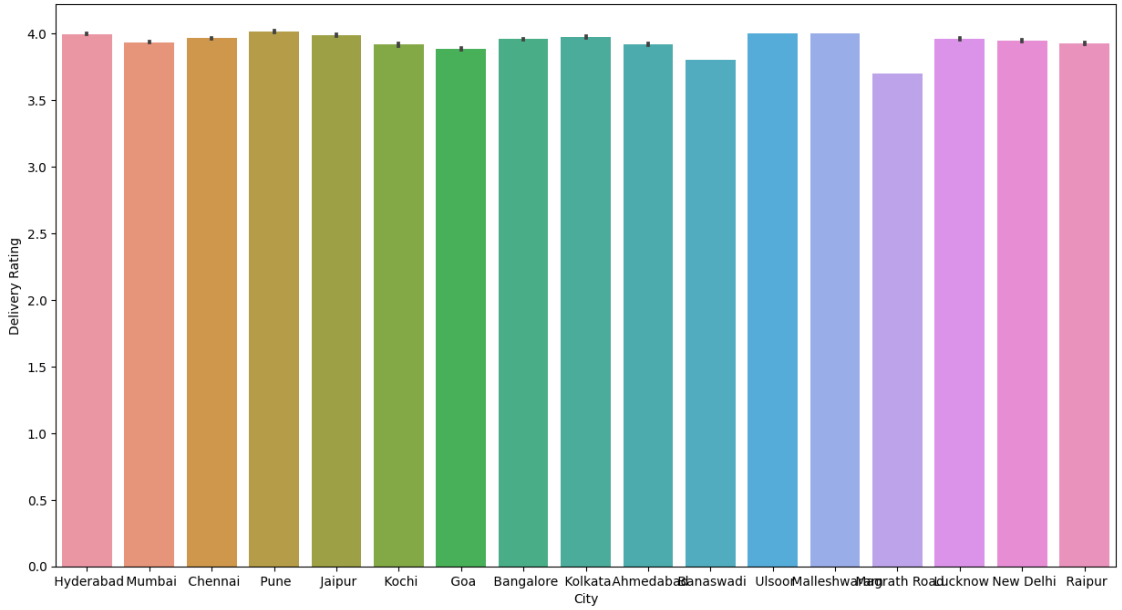


[22]:
```
plt.figure(figsize=(15,8))
sns.barplot(x='City',y='Dining Rating',data=df)
```

```
plt.show()
```



[23]:
```
plt.figure(figsize=(15,8))
sns.barplot(x='City',y= 'Delivery Rating', data=df)
plt.show()
```

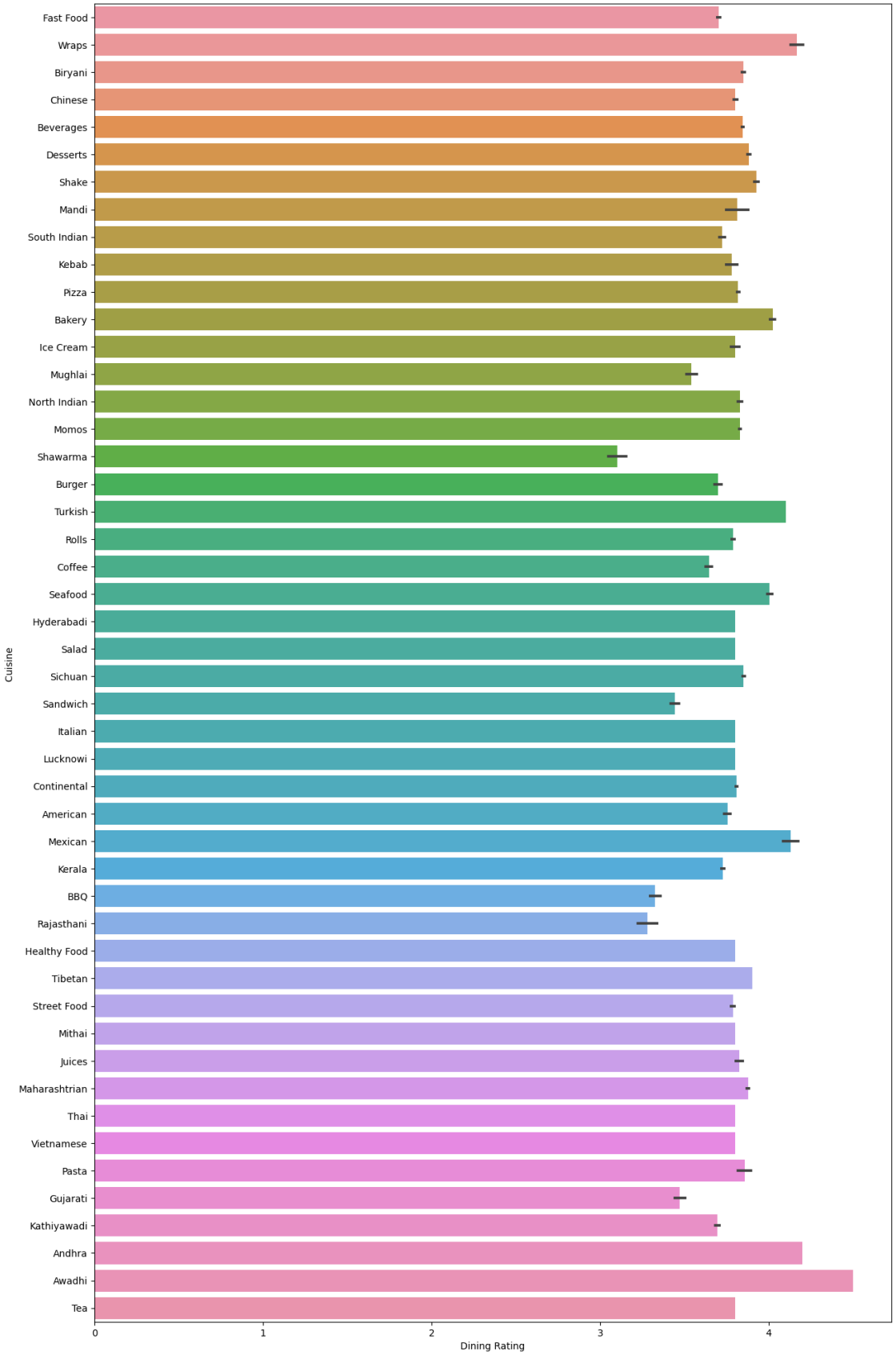


1.1.2 How many unique cuisines are represented in the dataset

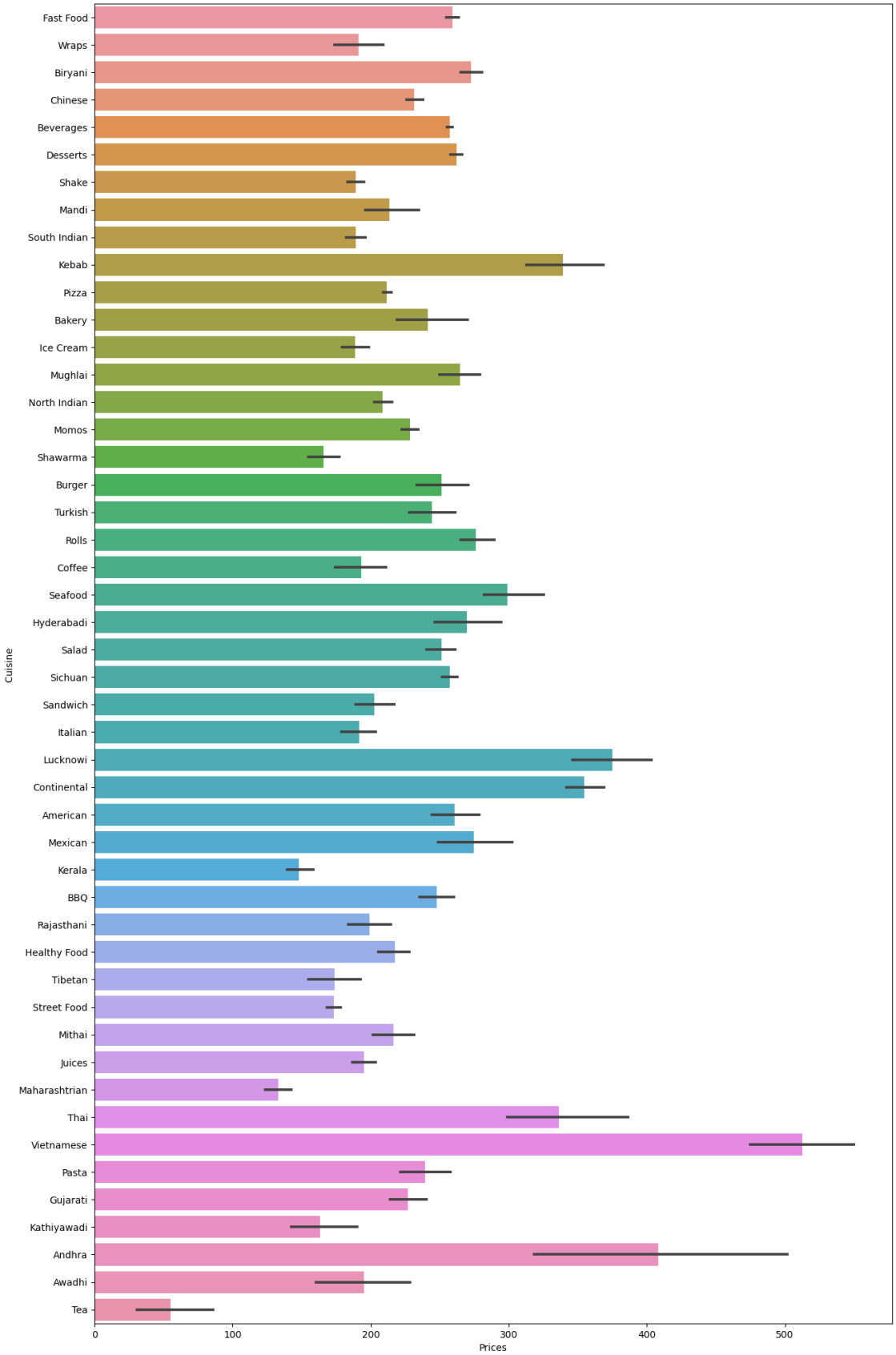
[24]:
```
df['Cuisine '].unique()
```

[24]:
```
array(['Fast Food', 'Wraps', 'Biryani', 'Chinese', 'Beverages',
       'Desserts', 'Shake', 'Mandi', 'South Indian', 'Kebab', 'Pizza',
       'Bakery', 'Ice Cream', 'Mughlai', 'North Indian', 'Momos',
       'Shawarma', 'Burger', 'Turkish', 'Rolls', 'Coffee', 'Seafood',
       'Hyderabadi', 'Salad', 'Sichuan', 'Sandwich', 'Italian',
       'Lucknowi', 'Continental', 'American', 'Mexican', 'Kerala', 'BBQ',
       'Rajasthani', 'Healthy Food', 'Tibetan', 'Street Food', 'Mithai',
       'Juices', 'Maharashtrian', 'Thai', 'Vietnamese', 'Pasta',
       'Gujarati', 'Kathiyawadi', 'Andhra', 'Awadhi', 'Tea'], dtype=object)
```

[25]:
```
plt.figure(figsize=(15,25))
sns.barplot(x='Dining Rating',y= 'Cuisine ', data=df)
plt.show()
```


Fast Food
Wraps
Biryani
Chinese
Beverages
Desserts
Shake
Mandi
South Indian
Kebab
Pizza
Bakery
Ice Cream
Mughlai
North Indian
Momos
Shawarma
Burger
Turkish
Rolls
Coffee
Seafood
Hyderabadi
Salad
Sichuan
Sandwich
Italian
Lucknowi
Continental
American
Mexican
Kerala
BBQ
Rajasthani
Healthy Food
Tibetan
Street Food
Mithai
Juices
Maharashtrian
Thai
Vietnamese
Pasta
Gujarati
Kathiyawadi
Andhra
Awadhi
Tea
Cuisine
0
1
2
3
4
Dining Rating

[26]:
```
plt.figure(figsize=(15,25))
sns.barplot(x='Prices',y= 'Cuisine ', data=df)
plt.show()
```


Fast Food
Wraps
Biryani
Chinese
Beverages
Desserts
Shake
Mandi
South Indian
Kebab
Pizza
Bakery
Ice Cream
Mughlai
North Indian
Momos
Shawarma
Burger
Turkish
Rolls
Coffee
Seafood
Hyderabadi
Salad
Sichuan
Sandwich
Italian
Lucknowi
Continental
American
Mexican
Kerala
BBQ
Rajasthani
Healthy Food
Tibetan
Street Food
Mithai
Juices
Maharashtrian
Thai
Vietnamese
Pasta
Gujarati
Kathiyawadi
Andhra
Awadhi
Tea
Cuisine
0
100
200
300
400
500
Prices

1.1.3 What is the average dining rating across all restaurants in the dataset

[27]:
```
print(np.mean(df['Dining Rating']))
```

```
3.815949041968459
```

1.1.4 Which metropolitan area has the highest average delivery rating

[28]:
```
gb = df.groupby('City')
```

[33]:
```
gb.mean()
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_16736\553916201.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  gb.mean()
```

[33]:

City	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Votes	Prices
Ahmedabad	3.782216	3.917175	142.973428	114.925402	12.531936	225.248306
Banaswadi	3.700000	3.800000	47.000000	139.000000	0.976471	349.466471
Bangalore	3.740401	3.957338	118.559854	94.002549	15.908335	231.867958
Chennai	3.818042	3.964381	206.660974	126.735592	11.900325	259.707505
Goa	3.877447	3.881912	20.161028	215.647722	0.693730	223.341547
Hyderabad	3.880092	3.996788	136.075731	99.663508		
Jaipur	3.779186	3.988339	192.196016	96.865439		
Kochi	3.823380	3.915401	159.499411	117.783538		
Kolkata	3.822085	3.974234	200.036808	52.443657		
Lucknow	3.855960	3.958623	244.714375	108.563969		
Magrath Road	3.500000	3.700000	0.000000	112.000000		
Malleshwaram	4.000000	4.000000	746.000000	0.000000		
Mumbai	3.825207	3.935348	143.072082	150.904364		
New Delhi	3.934645	3.945792	198.145719	52.298361		
Pune	3.752234	4.015382	87.548809	142.068504		
Raipur	3.834241	3.924095	68.001328	166.759880		
Ulsoor	3.800000	4.000000	0.000000	7.000000		

```
Hyderabad     34.876486  248.855075
Jaipur        20.038057  223.468007
Kochi         13.861134  228.144257
Kolkata       21.644592  237.867345
Lucknow       23.239026  238.958427
Magrath Road   1.800000  240.314000
Malleshwaram  17.161290  189.354839
Mumbai         8.189139  306.032386
New Delhi     15.345355  242.091027
Pune          10.840262  245.982877
Raipur         9.956161  197.693584
Ulsoor         0.000000  704.388889
```

From the above table we can see City Pune has highest average delivery rating.

1.1.5 What is the total number of dining votes received by all restaurants in each city

[49]:
```
df.groupby(['City','Restaurant Name'])['Dining Votes'].sum()
```

[49]:
```
City        Restaurant Name
 Ahmedabad  1944 -The HOCCO Kitchen            0
            A-One Bombay Biryani             450
            Al Baik Fast Food               2340
            Alinea Restaurant & Banquet    45738
            Alpine Restaurant & Banquet    56760
                                           …
 Raipur     Ustaad's Kitchen                5112
            Veggiies                        9548
            Wafflez                         5472
            Xero Degrees                       0
 Ulsoor     Dum Safar Biryani                  0
Name: Dining Votes, Length: 907, dtype: int64
```

Above table has total number of dining votes of each restaurant of the city

[46]:
```
gb.sum()
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_16736\624691301.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  gb.sum()
```

[46]:
```
              Dining Rating  Delivery Rating  Dining Votes  Delivery Votes  \
City
 Ahmedabad          31029.3          32136.5       1172954          942848
 Banaswadi            314.5            323.0          3995           11815
```

Bangalore	33745.9	35703.1	1069647	848091
Chennai	41139.4	42716.2	2226772	1365576
Goa	8596.3	8606.2	44697	478091
Hyderabad	48314.9	49768.0	1694415	1241010
Jaipur	40218.1	42443.9	2045350	1030842
Kochi	22714.7	23261.4	947586	699752
Kolkata	25336.6	26345.2	1326044	347649
Lucknow	18535.6	19029.1	1176342	521867
Magrath Road	122.5	129.5	0	3920
Malleshwaram	124.0	124.0	23126	0
Mumbai	43037.4	44276.6	1609704	1697825
New Delhi	10800.6	10831.2	543910	143559
Pune	24100.6	25790.8	562326	912506
Raipur	23089.8	23630.9	409504	1004228
Ulsoor	205.2	216.0	0	378

City	Votes	Prices
Ahmedabad	102812	1847937.10
Banaswadi	83	29704.65
Bangalore	143525	2091912.72
Chennai	128226	2798348.37
Goa	1538	495148.21
Hyderabad	434282	3098743.39
Jaipur	213245	2378146.53
Kochi	82349	1355405.03
Kolkata	143482	1576822.63
Lucknow	111710	1148673.16
Magrath Road	63	8410.99
Malleshwaram	532	5870.00
Mumbai	92136	3443170.37
New Delhi	42123	664539.87
Pune	69627	1579948.02
Raipur	59956	1190510.76
Ulsoor	0	38037.00

Above table has total number of dining votes for all restaurants in the city.

1.1.6 Which restaurant has the highest average dining rating in each city

[64]:
```
idx = gb['Dining Rating'].idxmax()
rate = df.loc[idx]
```

[65]:
```
print(rate[['Restaurant Name','City','Dining Rating']])
```

	Restaurant Name	City	Dining Rating
89172	Urban Khichdi	Ahmedabad	4.6
94473	GOPIZZA	Banaswadi	3.7

65625	Truffles	Bangalore	4.6
32723	AB's - Absolute Barbecues	Chennai	4.7
63090	Ritz Classic	Goa	4.4
15354	Exotica	Hyderabad	4.6
76069	Thali and More	Jaipur	4.7
56766	Cafe 17	Kochi	4.6
72232	Chowman	Kolkata	4.4
100331	Dastarkhwan	Lucknow	4.5
99791	Keventers Ice Cream	Magrath Road	3.5
96031	Rajdhani	Malleshwaram	4.0
17436	Chaitanya	Mumbai	4.5
106300	Natural Ice Cream	New Delhi	4.8
43248	Sukanta	Pune	4.2
118057	Creams N Caffeine	Raipur	4.3
94757	Dum Safar Biryani	Ulsoor	3.8

[ ]: