# level-1

June 28, 2024

[75]:
```
import pandas as pd # for the data manupulation
import numpy as np # for computations
import matplotlib.pyplot as plt # for visual graphs
import seaborn as sns
```

[38]:
```
df1 = pd.read_csv("Dataset .csv")
```

[39]:
```
df1.head()
```

[39]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Longitude | Latitude |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalayaan Avenu… | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City, Mak… | 121.027535 | 14.565443 |
| 1 | 6304287 | Izakaya Kikufuji | 162 | Makati City | Little Tokyo, 2277 Chino Roces Avenue, Legaspi… | Little Tokyo, Legaspi Village, Makati City | Little Tokyo, Legaspi Village, Makati City, Ma… | 121.014101 | 14.553708 |
| 2 | 6300002 | Heat - Edsa Shangri-La | 162 | Mandaluyong City | Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal… | Edsa Shangri-La, Ortigas, Mandaluyong City | Edsa Shangri-La, Ortigas, Mandaluyong City, Ma… | 121.056831 | 14.581404 |
| 3 | 6318506 | Ooma | 162 | Mandaluyong City | Third Floor, Mega Fashion Hall, SM Megamall, O… | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City, Mandal… | 121.056475 | 14.585318 |
| 4 | 6314302 | Sambo Kojin | 162 | Mandaluyong City | Third Floor, Mega Atrium, SM Megamall, Ortigas… | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City, Mandal… | 121.057508 | 14.584450 |

```
                          Cuisines  …          Currency Has Table booking  \
0        French, Japanese, Desserts  …  Botswana Pula(P)               Yes
1                          Japanese  …  Botswana Pula(P)               Yes
2  Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)               Yes
3                   Japanese, Sushi  …  Botswana Pula(P)                No
4                  Japanese, Korean  …  Botswana Pula(P)               Yes

  Has Online delivery Is delivering now Switch to order menu Price range  \
0                  No                No                    No           3
1                  No                No                    No           3
2                  No                No                    No           4
3                  No                No                    No           4
4                  No                No                    No           4

   Aggregate rating  Rating color Rating text Votes
0               4.8    Dark Green   Excellent   314
1               4.5    Dark Green   Excellent   591
2               4.4         Green   Very Good   270
3               4.9    Dark Green   Excellent   365
4               4.8    Dark Green   Excellent   229

[5 rows x 21 columns]
```

[40]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Restaurant ID         9551 non-null   int64
 1   Restaurant Name       9551 non-null   object
 2   Country Code          9551 non-null   int64
 3   City                  9551 non-null   object
 4   Address               9551 non-null   object
 5   Locality              9551 non-null   object
 6   Locality Verbose      9551 non-null   object
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object
 10  Average Cost for two  9551 non-null   int64
 11  Currency              9551 non-null   object
 12  Has Table booking     9551 non-null   object
 13  Has Online delivery   9551 non-null   object
 14  Is delivering now     9551 non-null   object
 15  Switch to order menu  9551 non-null   object
 16  Price range           9551 non-null   int64
```

```
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object
 19  Rating text           9551 non-null   object
 20  Votes                 9551 non-null   int64
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

[41]:
```
df1.describe() #descriptive stat
```

[41]:
```
       Restaurant ID  Country Code    Longitude     Latitude  \
count   9.551000e+03   9551.000000  9551.000000  9551.000000
mean    9.051128e+06     18.365616    64.126574    25.854381
std     8.791521e+06     56.750546    41.467058    11.007935
min     5.300000e+01      1.000000  -157.948486   -41.330428
25%     3.019625e+05      1.000000    77.081343    28.478713
50%     6.004089e+06      1.000000    77.191964    28.570469
75%     1.835229e+07      1.000000    77.282006    28.642758
max     1.850065e+07    216.000000   174.832089    55.976980

       Average Cost for two  Price range  Aggregate rating         Votes
count           9551.000000  9551.000000       9551.000000   9551.000000
mean            1199.210763     1.804837          2.666370    156.909748
std            16121.183073     0.905609          1.516378    430.169145
min                0.000000     1.000000          0.000000      0.000000
25%              250.000000     1.000000          2.500000      5.000000
50%              400.000000     2.000000          3.200000     31.000000
75%              700.000000     2.000000          3.700000    131.000000
max           800000.000000     4.000000          4.900000  10934.000000
```

[42]:
```
df1.isnull().sum() #says if any null values exist
```

[42]:
```
Restaurant ID         0
Restaurant Name       0
Country Code          0
City                  0
Address               0
Locality              0
Locality Verbose      0
Longitude             0
Latitude              0
Cuisines              9
Average Cost for two  0
Currency              0
Has Table booking     0
Has Online delivery   0
Is delivering now     0
Switch to order menu  0
```

```
Price range             0
Aggregate rating        0
Rating color            0
Rating text             0
Votes                   0
dtype: int64
```

[43]:
```
df1.shape
```

[43]:
```
(9551, 21)
```

[44]:
```
df1.columns
```

[44]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

[45]:
```
df1.nunique() # for unique values in the dataset
```

[45]:
```
Restaurant ID          9551
Restaurant Name        7446
Country Code             15
City                    141
Address                8918
Locality               1208
Locality Verbose       1265
Longitude              8120
Latitude               8677
Cuisines               1825
Average Cost for two    140
Currency                 12
Has Table booking         2
Has Online delivery       2
Is delivering now         2
Switch to order menu      1
Price range               4
Aggregate rating         33
Rating color              6
Rating text               6
Votes                  1012
dtype: int64
```

[46]:
```
df1['Cuisines'].unique()
```

[46]:
```
array(['French, Japanese, Desserts', 'Japanese',
       'Seafood, Asian, Filipino, Indian', …, 'Burger, Izgara',
       'World Cuisine, Patisserie, Cafe', 'Italian, World Cuisine'],
      dtype=object)
```

[47]:
```
df1.tail()
```

[47]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Longitude | Latitude | Cuisines | … | Currency | Has Table booking | Has Online delivery | Is delivering now | Switch to order menu | Price range | Aggregate rating | Rating color |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9546 | 5915730 | Naml Gurme | 208 | stanbul | Kemanke Karamustafa Pa a Mahallesi, R ht m … | Karak _y | Karak _y, stanbul | 28.977392 | 41.022793 | Turkish | … | Turkish Lira(TL) | No | No | No | No | 3 | 4.1 | Green |
| 9547 | 5908749 | Ceviz A ac | 208 | stanbul | Ko uyolu Mahallesi, Muhittin st _nda Cadd… | Ko uyolu | Ko uyolu, stanbul | 29.041297 | 41.009847 | World Cuisine, Patisserie, Cafe | … | Turkish Lira(TL) | No | No | No | No | 3 | 4.2 | Green |
| 9548 | 5915807 | Huqqa | 208 | stanbul | Kuru _e me Mahallesi, Muallim Naci Caddesi, N… | Kuru _e me | Kuru _e me, stanbul | 29.034640 | 41.055817 | Italian, World Cuisine | … | Turkish Lira(TL) | No | No | No | No | 4 | 3.7 | Yellow |
| 9549 | 5916112 | A k Kahve | 208 | stanbul | Kuru _e me Mahallesi, Muallim Naci Caddesi, N… | Kuru _e me | Kuru _e me, stanbul | 29.036019 | 41.057979 | Restaurant Cafe | … | Turkish Lira(TL) | No | No | No | No | 4 | 4.0 | Green |
| 9550 | 5927402 | Walter's Coffee Roastery | 208 | stanbul | Cafea a Mahallesi, Bademalt Sokak, No 21/B, … | Moda | Moda, stanbul | 29.026016 | 40.984776 | Cafe | … | Turkish Lira(TL) | No | No | No | No | 3 | 4.1 | Green |

```
9550                  No           2               4.0         Green

     Rating text Votes
9546   Very Good   788
9547   Very Good  1034
9548        Good   661
9549   Very Good   901
9550   Very Good   591

[5 rows x 21 columns]
```

[48]:
```
df1['City'].unique()
```

[48]:
```
array(['Makati City', 'Mandaluyong City', 'Pasay City', 'Pasig City',
       'Quezon City', 'San Juan City', 'Santa Rosa', 'Tagaytay City',
       'Taguig City', 'Bras _lia', 'Rio de Janeiro', 'S o Paulo',
       'Albany', 'Armidale', 'Athens', 'Augusta', 'Balingup',
       'Beechworth', 'Boise', 'Cedar Rapids/Iowa City', 'Chatham-Kent',
       'Clatskanie', 'Cochrane', 'Columbus', 'Consort', 'Dalton',
       'Davenport', 'Des Moines', 'Dicky Beach', 'Dubuque',
       'East Ballina', 'Fernley', 'Flaxton', 'Forrest', 'Gainesville',
       'Hepburn Springs', 'Huskisson', 'Inverloch', 'Lakes Entrance',
       'Lakeview', 'Lincoln', 'Lorn', 'Macedon', 'Macon', 'Mayfield',
       'Mc Millan', 'Middleton Beach', 'Miller', 'Monroe', 'Montville',
       'Ojo Caliente', 'Orlando', 'Palm Cove', 'Paynesville', 'Penola',
       'Pensacola', 'Phillip Island', 'Pocatello', 'Potrero', 'Princeton',
       'Rest of Hawaii', 'Savannah', 'Singapore', 'Sioux City',
       'Tampa Bay', 'Tanunda', 'Trentham East', 'Valdosta', 'Vernonia',
       'Victor Harbor', 'Vineland Station', 'Waterloo', 'Weirton',
       'Winchester Bay', 'Yorkton', 'Abu Dhabi', 'Dubai', 'Sharjah',
       'Agra', 'Ahmedabad', 'Allahabad', 'Amritsar', 'Aurangabad',
       'Bangalore', 'Bhopal', 'Bhubaneshwar', 'Chandigarh', 'Chennai',
       'Coimbatore', 'Dehradun', 'Faridabad', 'Ghaziabad', 'Goa',
       'Gurgaon', 'Guwahati', 'Hyderabad', 'Indore', 'Jaipur', 'Kanpur',
       'Kochi', 'Kolkata', 'Lucknow', 'Ludhiana', 'Mangalore', 'Mohali',
       'Mumbai', 'Mysore', 'Nagpur', 'Nashik', 'New Delhi', 'Noida',
       'Panchkula', 'Patna', 'Puducherry', 'Pune', 'Ranchi',
       'Secunderabad', 'Surat', 'Vadodara', 'Varanasi', 'Vizag',
       'Bandung', 'Bogor', 'Jakarta', 'Tangerang', 'Auckland',
       'Wellington City', 'Birmingham', 'Edinburgh', 'London',
       'Manchester', 'Doha', 'Cape Town', 'Inner City', 'Johannesburg',
       'Pretoria', 'Randburg', 'Sandton', 'Colombo', 'Ankara',
       ' stanbul'], dtype=object)
```

[49]:
```
df1.duplicated()
```

[49]:
```
0       False
1       False
2       False
3       False
4       False
        …
9546    False
9547    False
9548    False
9549    False
9550    False
Length: 9551, dtype: bool
```

[50]:
```
df1["Restaurant ID"].duplicated().sum()
```

[50]:
```
0
```

As the data is clean we can proceed for the analysis.

# 1 Task 1

Find the top three cusines and the percentage that serve them.

[51]:
```
#calculate the cusine types
Cuisines_counts = df1['Cuisines'].value_counts()
print(Cuisines_counts)
```

```
Cuisines
North Indian                                              936
North Indian, Chinese                                     511
Chinese                                                   354
Fast Food                                                 354
North Indian, Mughlai                                     334
                                                         …
Bengali, Fast Food                                          1
North Indian, Rajasthani, Asian                             1
Chinese, Thai, Malaysian, Indonesian                        1
Bakery, Desserts, North Indian, Bengali, South Indian       1
Italian, World Cuisine                                      1
Name: count, Length: 1825, dtype: int64
```

[52]:
```
Top_three = Cuisines_counts.head(3)
```

[53]:
```
print(Top_three)
```

```
Cuisines
North Indian             936
North Indian, Chinese    511
```

```
Chinese                  354
Name: count, dtype: int64
```

The North Indian, North Indian, Cheinese and Chinese are the three Cuisines which servered most

To find how many of Restaurants serve them need to find how many resturants are their

[54]:
```
restaurants = len(df1['Restaurant Name'])
print(restaurants)
```

```
9551
```

[55]:
```
# as we have total number of restaurants we can calculate the percentage

Each_restaurant_percent = (Top_three/restaurants)*100
print(Each_restaurant_percent)
```

```
Cuisines
North Indian             9.800021
North Indian, Chinese    5.350225
Chinese                  3.706418
Name: count, dtype: float64
```

## 2 Task 2

City Analysis, . Identify the city with the highest number of restaurants in the dataset. . Calculate the average rating for restaurants in each city. . Determine the city with the highest average rating.

[56]:
```
#First one
city_to_restaurants = df1['City'].value_counts()
print(city_to_restaurants)
```

```
City
New Delhi           5473
Gurgaon             1118
Noida               1080
Faridabad            251
Ghaziabad             25
                    ...
Panchkula              1
Mc Millan              1
Mayfield               1
Macedon                1
Vineland Station       1
Name: count, Length: 141, dtype: int64
```

[73]:
```
# As we known the new delhi has highest restaurants we can do
print("The City with highest restaurants is :", city_to_restaurants.idxmax())
print("which is with the restaurant count as:", city_to_restaurants.max() )
```

```
The City with highest restaurants is : New Delhi
which is with the restaurant count as: 5473
```

In the above task the idxmax() is used because The pandas.DataFrame.idxmax() function returns the index of the first occurrence of the maximum value over the requested axis.

As the next task we have to find the average Rating first look at the columns

[64]:
```
df1.columns
```

[64]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

As we want the Avg Rating of the restaurants in each city we will use the group by and mean functions togeother.

[68]:
```
Avg_rating_each_city = df1.groupby("City")["Aggregate rating"].mean()
print(Avg_rating_each_city)
```

```
City
Abu Dhabi          4.300000
Agra               3.965000
Ahmedabad          4.161905
Albany             3.555000
Allahabad          3.395000
                       …
Weirton            3.900000
Wellington City    4.250000
Winchester Bay     3.200000
Yorkton            3.300000
 stanbul          4.292857
Name: Aggregate rating, Length: 141, dtype: float64
```

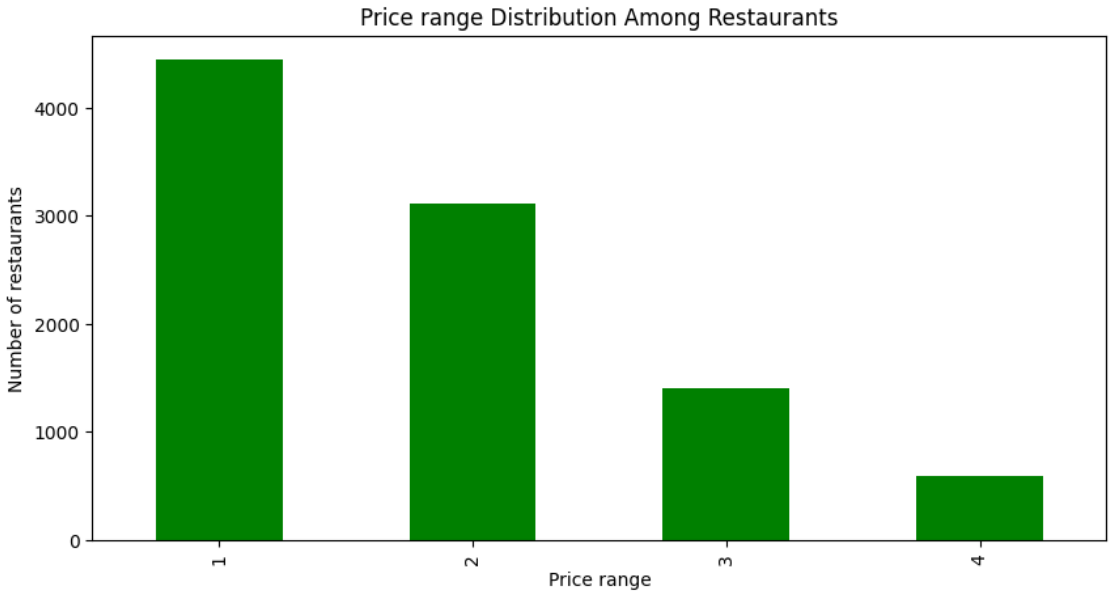
[71]:
```
# So as we want to known highest avg rating we can do that by
print("City with Highest Avg Rating for the restaurants is :",␣
 ↪Avg_rating_each_city.idxmax())
print("Which is with the rating :", Avg_rating_each_city.max())
```

```
City with Highest Avg Rating for the restaurants is : Inner City
Which is with the rating : 4.9
```

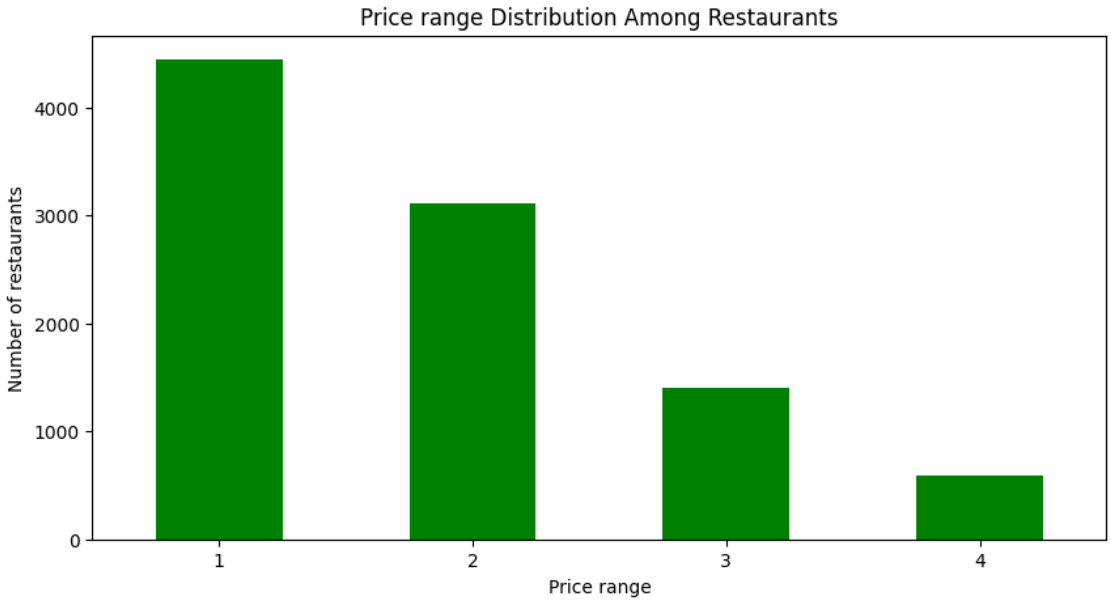
## 3 Task 3

Price Range Distribution. A.Create a histogram or bar chart to visualize the distribution of price ranges among the restaurants. B. Calculate the percentage of restaurants in each price range category.

[80]:
```
plt.figure(figsize=(10, 5))
df1['Price range'].value_counts().plot(kind='bar', color='green')
plt.title('Price range Distribution Among Restaurants')
plt.xlabel('Price range')
plt.ylabel('Number of restaurants')
plt.show()
```




[81]:
```
# As we can see the x axis has the labels rounded we can arrange them correctly
plt.figure(figsize=(10, 5))
df1['Price range'].value_counts().plot(kind='bar', color='green')
plt.title('Price range Distribution Among Restaurants')
plt.xlabel('Price range')
plt.ylabel('Number of restaurants')
plt.xticks(rotation=0)
plt.show()
```




[82]:
```
# each price range category
Price_counts = df1['Price range'].value_counts()
print(Price_counts)
```

```
Price range
1    4444
2    3113
3    1408
4     586
Name: count, dtype: int64
```

[83]:
```
# so we want to known the price range percentage
price_percent = (Price_counts/len(df1['Price range'])*100)
print(price_percent)
```

```
Price range
1    46.529159
2    32.593446
3    14.741912
4     6.135483
Name: count, dtype: float64
```

# 4 Task 4

A. Determine the percentage of restaurants that offer online delivery. B. Compare the average ratings of restaurants with and without online delivery.

[84]:
```
df1.columns
```

[84]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

[85]:
```
df1.head()
```

[85]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Longitude | Latitude | Cuisines | … | Currency | Has Table booking | Has Online delivery | Is delivering now | Switch to order menu | Price range | Aggregate rating | Rating color | Rating text | Votes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalayaan Avenu… | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City, Mak… | 121.027535 | 14.565443 | French, Japanese, Desserts | … | Botswana Pula(P) | Yes | No | No | No | 3 | 4.8 | Dark Green | Excellent | 314 |
| 1 | 6304287 | Izakaya Kikufuji | 162 | Makati City | Little Tokyo, 2277 Chino Roces Avenue, Legaspi… | Little Tokyo, Legaspi Village, Makati City | Little Tokyo, Legaspi Village, Makati City, Ma… | 121.014101 | 14.553708 | Japanese | … | Botswana Pula(P) | Yes | No | No | No | 3 | 4.5 | Dark Green | Excellent | 591 |
| 2 | 6300002 | Heat - Edsa Shangri-La | 162 | Mandaluyong City | Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal… | Edsa Shangri-La, Ortigas, Mandaluyong City | Edsa Shangri-La, Ortigas, Mandaluyong City, Ma… | 121.056831 | 14.581404 | Seafood, Asian, Filipino, Indian | … | Botswana Pula(P) | Yes | No | No | No | 4 | 4.4 | Green | Very Good | 270 |
| 3 | 6318506 | Ooma | 162 | Mandaluyong City | Third Floor, Mega Fashion Hall, SM Megamall, O… | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City, Mandal… | 121.056475 | 14.585318 | Japanese, Sushi | … | Botswana Pula(P) | No | No | No | No | 4 | 0.0 | White | Not rated | 365 |
| 4 | 6314302 | Sambo Kojin | 162 | Mandaluyong City | Third Floor, Mega Atrium, SM Megamall, Ortigas… | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City, Mandal… | 121.057508 | 14.584450 | Japanese, Korean | … | Botswana Pula(P) | Yes | No | No | No | 4 | 4.9 | Dark Green | Excellent | 365 |

```
  Has Online delivery Is delivering now Switch to order menu Price range  \
0                  No                 No                    No           3
1                  No                 No                    No           3
2                  No                 No                    No           4
3                  No                 No                    No           4
4                  No                 No                    No           4

   Aggregate rating  Rating color Rating text Votes
0               4.8    Dark Green   Excellent   314
1               4.5    Dark Green   Excellent   591
2               4.4         Green   Very Good   270
3               4.9    Dark Green   Excellent   365
4               4.8    Dark Green   Excellent   229

[5 rows x 21 columns]
```

[90]:
```
# as the Has online Delivery is catogorial in nature
online_delivery_avaliable = (df1['Has Online delivery'] == 'Yes').sum()
print("Online Delivery is avaliable in the restaurants :"
      ,online_delivery_avaliable,", out of",len(df1['Has Online delivery']))
# as We want percentage
percent_online = (online_delivery_avaliable/len(df1['Has Online delivery'])*100)
print("The percentyage for the above numbers is :",round(percent_online, 2),"%")
```

```
Online Delivery is avaliable in the restaurants : 2451 , out of 9551
The percentyage for the above numbers is : 25.66 %
```

[91]:
```
offline_delivery_avaliable = (df1['Has Online delivery'] == 'No').sum()
print(online_delivery_avaliable)
print(offline_delivery_avaliable)
```

```
2451
7100
```

[93]:
```
# So we will use the group by along with aggreate rating
Avg_rating_online = df1.groupby('Has Online delivery')['Aggregate rating'].
 ↪mean()
print(Avg_rating_online)
```

```
Has Online delivery
No     2.465296
Yes    3.248837
Name: Aggregate rating, dtype: float64
```

[97]:
```
print("The Avg Rating for the restaurants with online delivery is :",␣
 ↪round(Avg_rating_online.max(),2))
```

```
print("The Avg Rating without online delivery is :", round(Avg_rating_online.
  ↪min(), 2))
```

```
The Avg Rating for the restaurants with online delivery is : 3.25
The Avg Rating without online delivery is : 2.47
```

[ ]: