ZomatoEDA

January 4, 2025

0.1 Zomato Dataset Exploratory Data Analysis

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

[2]:
```
df=pd.read_csv('zomato.csv',encoding='latin-1')
df.head()
```

[2]:
```
   Restaurant ID         Restaurant Name  Country Code              City  \
0        6317637        Le Petit Souffle           162       Makati City
1        6304287        Izakaya Kikufuji           162       Makati City
2        6300002  Heat - Edsa Shangri-La           162  Mandaluyong City
3        6318506                    Ooma           162  Mandaluyong City
4        6314302             Sambo Kojin           162  Mandaluyong City

                                             Address  \
0  Third Floor, Century City Mall, Kalayaan Avenu…
1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi…
2  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…
3  Third Floor, Mega Fashion Hall, SM Megamall, O…
4  Third Floor, Mega Atrium, SM Megamall, Ortigas…

                                     Locality  \
0   Century City Mall, Poblacion, Makati City
1  Little Tokyo, Legaspi Village, Makati City
2  Edsa Shangri-La, Ortigas, Mandaluyong City
3      SM Megamall, Ortigas, Mandaluyong City
4      SM Megamall, Ortigas, Mandaluyong City

                                    Locality Verbose   Longitude   Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708
2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831  14.581404
3  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475  14.585318
4  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]
```

[3]:
```
df.columns
```

[3]:
```
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')
```

[4]:
```
df.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
```

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

[5]:
```
       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
```

0.2 In Data Analysis What All Things We Do

1. Missing Values
2. Explore About the Numerical Variables
3. Explore About categorical Variables
4. Finding Relationship between features

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

```
[6]: (9551, 21)
```

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

```
[7]: 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
```

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

```
[8]: 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
```

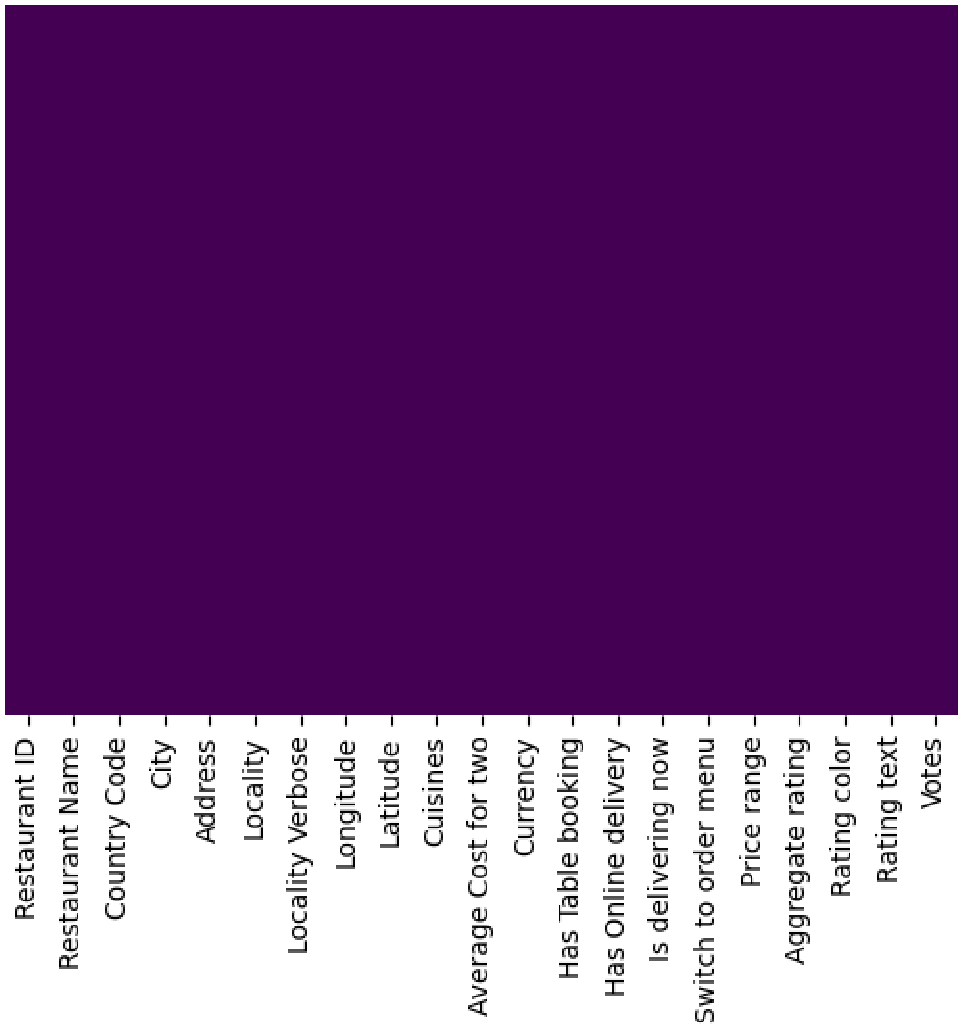
```
Rating color            0
Rating text             0
Votes                   0
dtype: int64
```

[9]:
```
[features for features in df.columns if df[features].isnull().sum()>0]
```

[9]:
```
['Cuisines']
```

[10]:
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

[10]:
```
<Axes: >
```



[11]:
```
df_country=pd.read_excel('Country-Code.xlsx')
df_country.head()
```

```
[11]:    Country Code    Country
      0             1      India
      1            14  Australia
      2            30     Brazil
      3            37     Canada
      4            94  Indonesia
```

```
[12]: df.columns
```

```
[12]: 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')
```

```
[13]: final_df=pd.merge(df,df_country,on='Country Code', how='left')
```

```
[14]: final_df.head(2)
```

```
[14]:    Restaurant ID   Restaurant Name  Country Code         City  \
      0        6317637  Le Petit Souffle           162  Makati City
      1        6304287  Izakaya Kikufuji           162  Makati City

                                                  Address  \
      0  Third Floor, Century City Mall, Kalayaan Avenu…
      1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi…

                                         Locality  \
      0   Century City Mall, Poblacion, Makati City
      1  Little Tokyo, Legaspi Village, Makati City

                                        Locality Verbose   Longitude   Latitude  \
      0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
      1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708

                               Cuisines  …  Has Table booking Has Online delivery  \
      0  French, Japanese, Desserts  …                Yes                  No
      1                    Japanese  …                Yes                  No

        Is delivering now Switch to order menu Price range Aggregate rating  \
      0              No                   No            3              4.8
      1              No                   No            3              4.5

         Rating color  Rating text Votes      Country
      0    Dark Green    Excellent   314  Phillipines
```

```
1    Dark Green    Excellent   591  Phillipines

[2 rows x 22 columns]
```

[15]:
```
##To check Data Types
final_df.dtypes
```

[15]:
```
Restaurant ID              int64
Restaurant Name           object
Country Code               int64
City                      object
Address                   object
Locality                  object
Locality Verbose          object
Longitude                float64
Latitude                 float64
Cuisines                  object
Average Cost for two       int64
Currency                  object
Has Table booking         object
Has Online delivery       object
Is delivering now         object
Switch to order menu      object
Price range                int64
Aggregate rating         float64
Rating color              object
Rating text               object
Votes                      int64
Country                   object
dtype: object
```

[16]:
```
final_df.columns
```

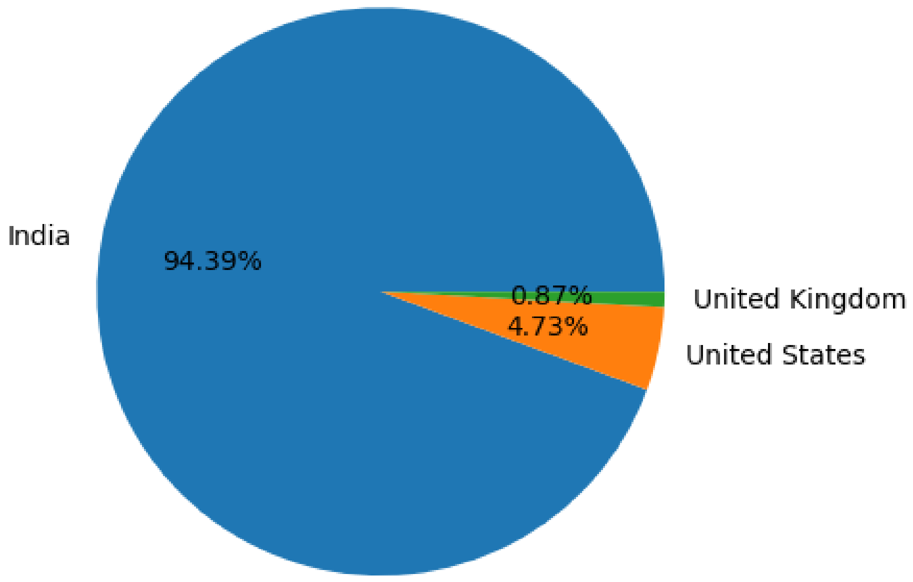
[16]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes', 'Country'],
      dtype='object')
```

[17]:
```
country_names=final_df.Country.value_counts().index
```

[18]:
```
country_val=final_df.Country.value_counts().values
```

[19]:
```
## Pie Chart- Top 3 countries that uses zomato
plt.pie(country_val[:3],labels=country_names[:3],autopct='%1.2f%%')
```

```
[19]: ([<matplotlib.patches.Wedge at 0x7a3cc189bca0>,
        <matplotlib.patches.Wedge at 0x7a3cc189bb80>,
        <matplotlib.patches.Wedge at 0x7a3cc16e4760>],
       [Text(-1.0829742700952103, 0.19278674827836725, 'India'),
        Text(1.077281715838356, -0.22240527134123297, 'United States'),
        Text(1.0995865153823035, -0.03015783794312073, 'United Kingdom')],
       [Text(-0.590713238233751, 0.10515640815183668, '94.39%'),
        Text(0.5876082086391032, -0.12131196618612707, '4.73%'),
        Text(0.5997744629358018, -0.01644972978715676, '0.87%')])
```



Observation:Zomato maximum records or transaction are from India After that USA and then United Kingdoms

```
[20]: final_df.columns
```

```
[20]: 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', 'Country'],
            dtype='object')
```

[21]:
```
ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).
↪size().reset_index().rename(columns={0:'Rating Count'})
```

[22]:
```
ratings
```

[22]:
```
    Aggregate rating Rating color Rating text  Rating Count
0                0.0        White   Not rated          2148
1                1.8          Red        Poor             1
2                1.9          Red        Poor             2
3                2.0          Red        Poor             7
4                2.1          Red        Poor            15
5                2.2          Red        Poor            27
6                2.3          Red        Poor            47
7                2.4          Red        Poor            87
8                2.5       Orange     Average           110
9                2.6       Orange     Average           191
10               2.7       Orange     Average           250
11               2.8       Orange     Average           315
12               2.9       Orange     Average           381
13               3.0       Orange     Average           468
14               3.1       Orange     Average           519
15               3.2       Orange     Average           522
16               3.3       Orange     Average           483
17               3.4       Orange     Average           498
18               3.5       Yellow        Good           480
19               3.6       Yellow        Good           458
20               3.7       Yellow        Good           427
21               3.8       Yellow        Good           400
22               3.9       Yellow        Good           335
23               4.0        Green   Very Good           266
24               4.1        Green   Very Good           274
25               4.2        Green   Very Good           221
26               4.3        Green   Very Good           174
27               4.4        Green   Very Good           144
28               4.5   Dark Green   Excellent            95
29               4.6   Dark Green   Excellent            78
30               4.7   Dark Green   Excellent            42
31               4.8   Dark Green   Excellent            25
32               4.9   Dark Green   Excellent            61
```

0.3 Observation

1. When Rating is between 4.5 to 4.9—-> Excellent
2. When Rating are between 4.0 to 3.4—->very good
3. when Rating is between 3.5 to 3.9—-> good
4. when Rating is between 3.0 to 3.4—-> average
5. when Rating is between 2.5 to 2.9—-> average

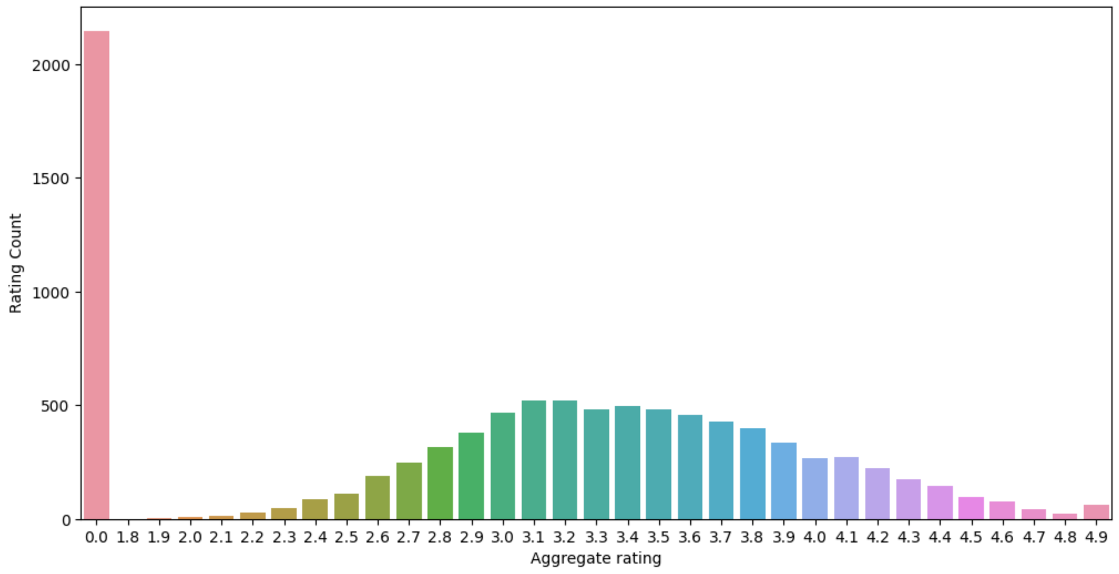
6. when Rating is between 2.0 to 2.4—-> Poor

[23]:
```
ratings.head()
```

[23]:
```
   Aggregate rating Rating color Rating text  Rating Count
0               0.0        White   Not rated          2148
1               1.8          Red        Poor             1
2               1.9          Red        Poor             2
3               2.0          Red        Poor             7
4               2.1          Red        Poor            15
```

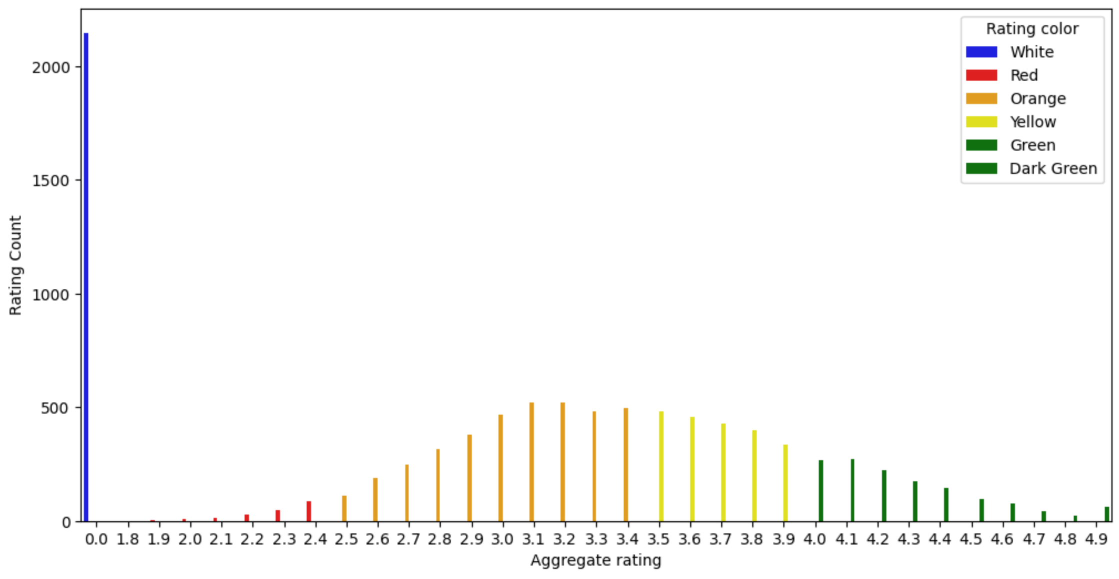
[24]:
```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (12, 6)
sns.barplot(x="Aggregate rating",y="Rating Count",data=ratings)
```

[24]:
```
<Axes: xlabel='Aggregate rating', ylabel='Rating Count'>
```



[25]:
```
sns.barplot(x="Aggregate rating",y="Rating Count",hue='Rating␣
↪color',data=ratings,palette=['blue','red','orange','yellow','green','green'])
```

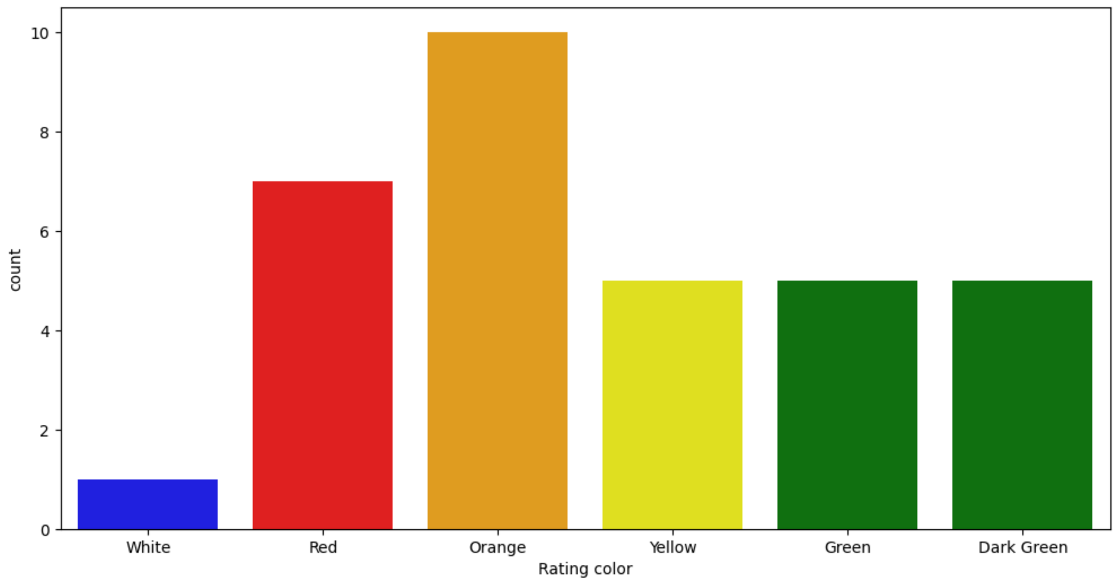
[25]:
```
<Axes: xlabel='Aggregate rating', ylabel='Rating Count'>
```



Observation: 1. Not Rated count is very high 2. Maximum number of rating are between 2.5 to 3.4

[26]:
```
## Count plot
sns.countplot(x="Rating␣
↪color",data=ratings,palette=['blue','red','orange','yellow','green','green'])
```

[26]:
```
<Axes: xlabel='Rating color', ylabel='count'>
```



[27]:
```
ratings
```

[27]:
```
    Aggregate rating Rating color Rating text  Rating Count
0                0.0        White   Not rated          2148
1                1.8          Red        Poor             1
2                1.9          Red        Poor             2
3                2.0          Red        Poor             7
4                2.1          Red        Poor            15
5                2.2          Red        Poor            27
6                2.3          Red        Poor            47
7                2.4          Red        Poor            87
8                2.5       Orange     Average           110
9                2.6       Orange     Average           191
10               2.7       Orange     Average           250
11               2.8       Orange     Average           315
12               2.9       Orange     Average           381
13               3.0       Orange     Average           468
14               3.1       Orange     Average           519
15               3.2       Orange     Average           522
16               3.3       Orange     Average           483
17               3.4       Orange     Average           498
18               3.5       Yellow        Good           480
19               3.6       Yellow        Good           458
20               3.7       Yellow        Good           427
21               3.8       Yellow        Good           400
22               3.9       Yellow        Good           335
23               4.0        Green   Very Good           266
24               4.1        Green   Very Good           274
25               4.2        Green   Very Good           221
26               4.3        Green   Very Good           174
27               4.4        Green   Very Good           144
28               4.5   Dark Green   Excellent            95
29               4.6   Dark Green   Excellent            78
30               4.7   Dark Green   Excellent            42
31               4.8   Dark Green   Excellent            25
32               4.9   Dark Green   Excellent            61
```

[28]:
```
### Find the countries name that has given 0 rating
final_df[final_df['Rating color']=='White'].groupby('Country').size().
 ↪reset_index()
```

[28]:
```
          Country     0
0          Brazil     5
1           India  2139
2  United Kingdom     1
3   United States     3
```

```
[29]: final_df.groupby(['Aggregate rating','Country']).size().reset_index().head(5)
```

```
[29]:    Aggregate rating         Country     0
      0               0.0          Brazil     5
      1               0.0           India  2139
      2               0.0  United Kingdom     1
      3               0.0   United States     3
      4               1.8           India     1
```

Observations Maximum number of 0 ratings are from Indian customers

```
[30]: ##find out which currency is used by which country?
      final_df.columns
```

```
[30]: 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', 'Country'],
            dtype='object')
```

```
[31]: final_df[['Country','Currency']].groupby(['Country','Currency']).size().
       ↪reset_index()
```

```
[31]:            Country                Currency     0
      0        Australia              Dollar($)    24
      1           Brazil      Brazilian Real(R$)    60
      2           Canada              Dollar($)     4
      3            India      Indian Rupees(Rs.)  8652
      4        Indonesia  Indonesian Rupiah(IDR)    21
      5      New Zealand           NewZealand($)    40
      6       Phillipines        Botswana Pula(P)    22
      7            Qatar          Qatari Rial(QR)    20
      8        Singapore              Dollar($)    20
      9     South Africa                 Rand(R)    60
      10       Sri Lanka   Sri Lankan Rupee(LKR)    20
      11          Turkey         Turkish Lira(TL)    34
      12             UAE      Emirati Diram(AED)    60
      13  United Kingdom              Pounds(£)    80
      14   United States              Dollar($)   434
```

```
[32]: ## Which Countries do have online deliveries option
```

```
[33]: final_df[final_df['Has Online delivery'] =="Yes"].Country.value_counts()
```

[33]:
```
India    2423
UAE        28
Name: Country, dtype: int64
```

[34]:
```
final_df[['Has Online delivery','Country']].groupby(['Has Online␣
 ↪delivery','Country']).size().reset_index()
```

[34]:
```
   Has Online delivery         Country     0
0                   No       Australia    24
1                   No          Brazil    60
2                   No          Canada     4
3                   No           India  6229
4                   No       Indonesia    21
5                   No     New Zealand    40
6                   No      Phillipines    22
7                   No           Qatar    20
8                   No       Singapore    20
9                   No    South Africa    60
10                  No       Sri Lanka    20
11                  No          Turkey    34
12                  No             UAE    32
13                  No  United Kingdom    80
14                  No   United States   434
15                 Yes           India  2423
16                 Yes             UAE    28
```

Observations: 1. Online Deliveries are available in India and UAE

[35]:
```
final_df.columns
```

[35]:
```
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', 'Country'],
      dtype='object')
```

[36]:
```
## Create a pie chart for top 5 cities distribution
```

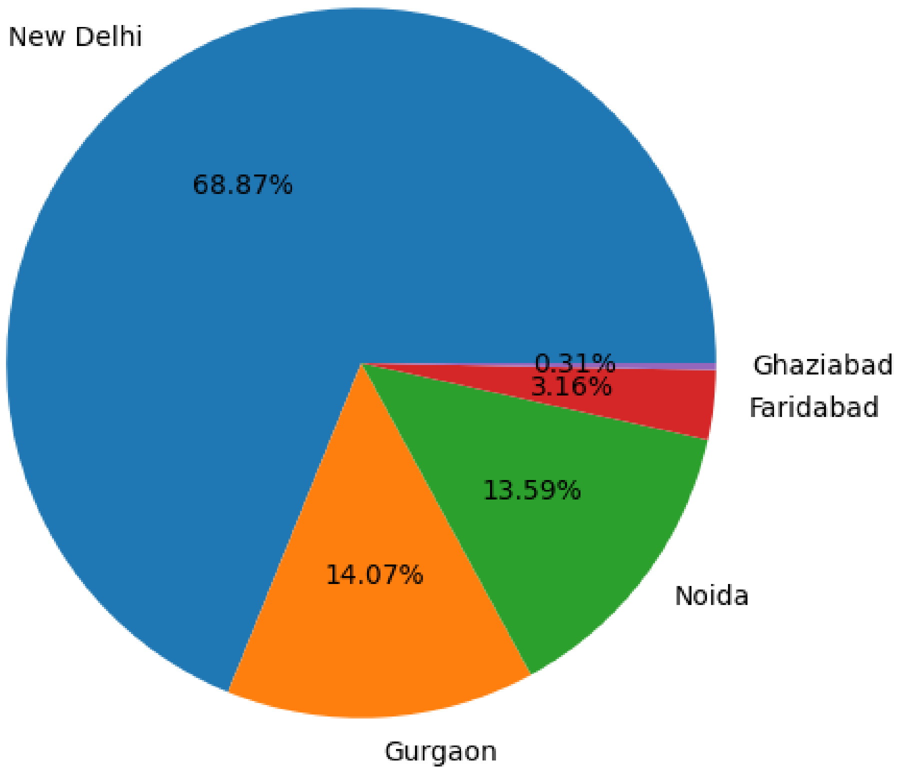
[37]:
```
final_df.City.value_counts().index
```

[37]:
```
Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad',
       'Bhubaneshwar', 'Amritsar', 'Ahmedabad', 'Lucknow', 'Guwahati',
       …
       'Ojo Caliente', 'Montville', 'Monroe', 'Miller', 'Middleton Beach',
       'Panchkula', 'Mc Millan', 'Mayfield', 'Macedon', 'Vineland Station'],
      dtype='object', length=141)
```

```
[38]: city_values=final_df.City.value_counts().values
      city_labels=final_df.City.value_counts().index
```

```
[39]: plt.pie(city_values[:5],labels=city_labels[:5],autopct='%1.2f%%')
```

```
[39]: ([<matplotlib.patches.Wedge at 0x7a3cc1007a90>,
        <matplotlib.patches.Wedge at 0x7a3cc1007970>,
        <matplotlib.patches.Wedge at 0x7a3cc1028550>,
        <matplotlib.patches.Wedge at 0x7a3cc1028be0>,
        <matplotlib.patches.Wedge at 0x7a3cc1029270>],
       [Text(-0.6145352824185932, 0.9123301960708633, 'New Delhi'),
        Text(0.0623675251198054, -1.0982305276263407, 'Gurgaon'),
        Text(0.8789045225625368, -0.6614581167535246, 'Noida'),
        Text(1.0922218418223437, -0.13058119407559224, 'Faridabad'),
        Text(1.099946280005612, -0.010871113182029924, 'Ghaziabad')],
       [Text(-0.3352010631374145, 0.497634652402289, '68.87%'),
        Text(0.0340186500653484, -0.5990348332507311, '14.07%'),
        Text(0.47940246685229276, -0.36079533641101336, '13.59%'),
        Text(0.5957573682667329, -0.07122610585941394, '3.16%'),
        Text(0.5999706981848791, -0.005929698099289049, '0.31%')])
```


New Delhi
Ghaziabad
Faridabad
Noida
Gurgaon