Exploratory Data Analysis on Zomato Dataset

In this notebook we are exploring about

1. Missing values
2. Explore about the numerical variable
3. Explore about the categorical variable
4. Finding Relationship between features

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

In [2]:
```
# read the dataset

df = pd.read_csv('data/Zomatodataset/zomato.csv', encoding='latin-1')
df.head(3)
```

Out[2]:

	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 rows × 21 columns

In [3]:
```
# get the columns
df.columns
```

Out[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')
```

In [4]:
```
#finding shape of the dataframe
df.shape
```

Out[4]:
```
(9551, 21)
```

In [5]:
```
# get the datatypes and columns names
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
```

In [6]:
```
# get the statistical info about the dataset - numerical features
df.describe()
```

Out[6]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.0
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156.9
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430.1
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.0
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5.0
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31.0
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131.0
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.0

In [7]:
```
# Checking missing values
df.isnull().sum()
```

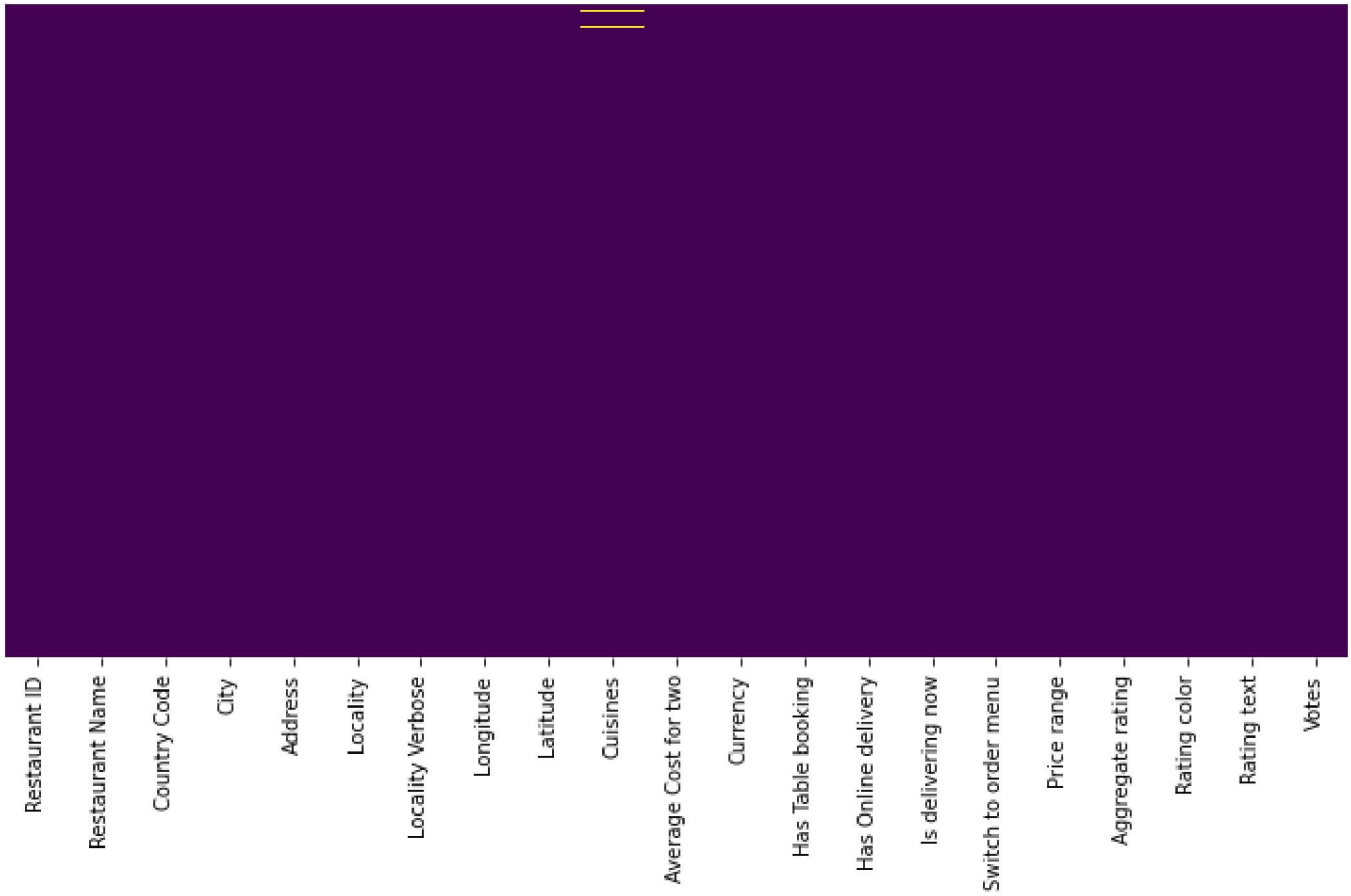
Out[7]:
```
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                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
```

In [8]:
```
# finding which column has missing values - other options
[features for features in df.columns if df[features].isnull().sum()>0]
```

Out[8]:
```
['Cuisines']
```

In [9]:
```
# null values using seaborn's heatmap
# our dataframe size is (9551, 21) out of this we have only 9 missing value so its hard to
matplotlib.rcParams['figure.figsize'] = (12,6)
sns.heatmap(df.isnull(), yticklabels=False, cbar=False, cmap='viridis')
```

Out[9]: <AxesSubplot:>



In [10]:
```
#read country code dataset
df_country = pd.read_excel("data/Zomatodataset/Country-Code.xlsx")
```

In [11]:
```
df_country.head()
```

Out[11]:

	Country Code	Country
0	1	India
1	14	Australia
2	30	Brazil
3	37	Canada
4	94	Indonesia

In [12]:
```
#merge country_code from df_country dataset with zomato dataset's cuntry_code
# on - on which column, how = how to join - inner, left, right, full
#our dataframe has full data so joining on left join
df_final = pd.merge(df, df_country, on='Country Code', how='left')
```

In [13]:
```
df_final.head()
```

Out[13]:

	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

5 rows × 22 columns

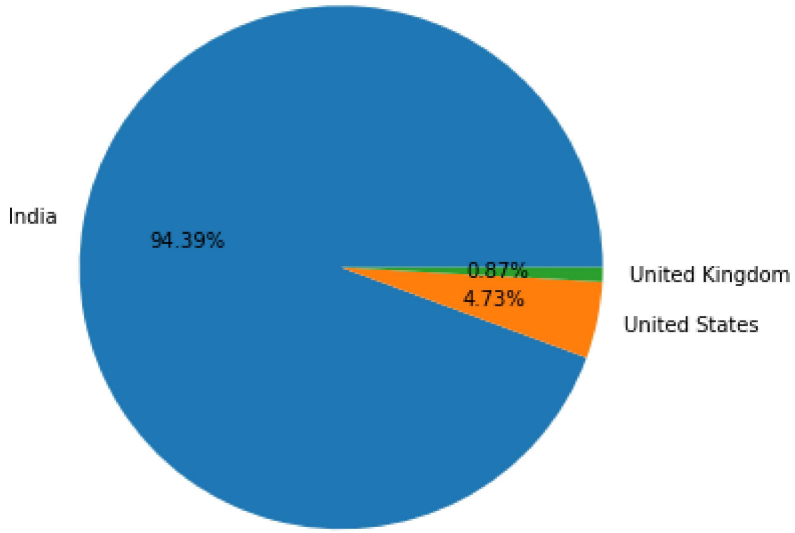
In [14]:
```
# checking unique records

df_final.Country.value_counts()
```

Out[14]:
```
India             8652
United States      434
United Kingdom      80
Brazil              60
UAE                 60
South Africa        60
New Zealand         40
Turkey              34
Australia           24
Phillipines         22
Indonesia           21
Singapore           20
Qatar               20
Sri Lanka           20
Canada               4
Name: Country, dtype: int64
```

In [15]:
```
#getting all the unique country name
country_names = df_final.Country.value_counts().index
```

In [16]:
```
country_values = df_final.Country.value_counts().values
```

In [17]:
```
#plot pie chart for top three countries
fig, ax = plt.subplots(figsize=(12, 6))
plt.pie(country_values[:3], labels = country_names[:3],autopct='%1.2f%%')
plt.show()
```



Observation

- Zomato's top transactions are from India after that USA and UK

In [18]:
```
df_final.groupby(['Aggregate rating', 'Rating color','Rating text']).size().reset_index()
```

Out[18]:

	Aggregate rating	Rating color	Rating text	0
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

In [19]:
```
rating = df_final.groupby(['Aggregate rating', 'Rating color','Rating text']).size().reset_
rating
```

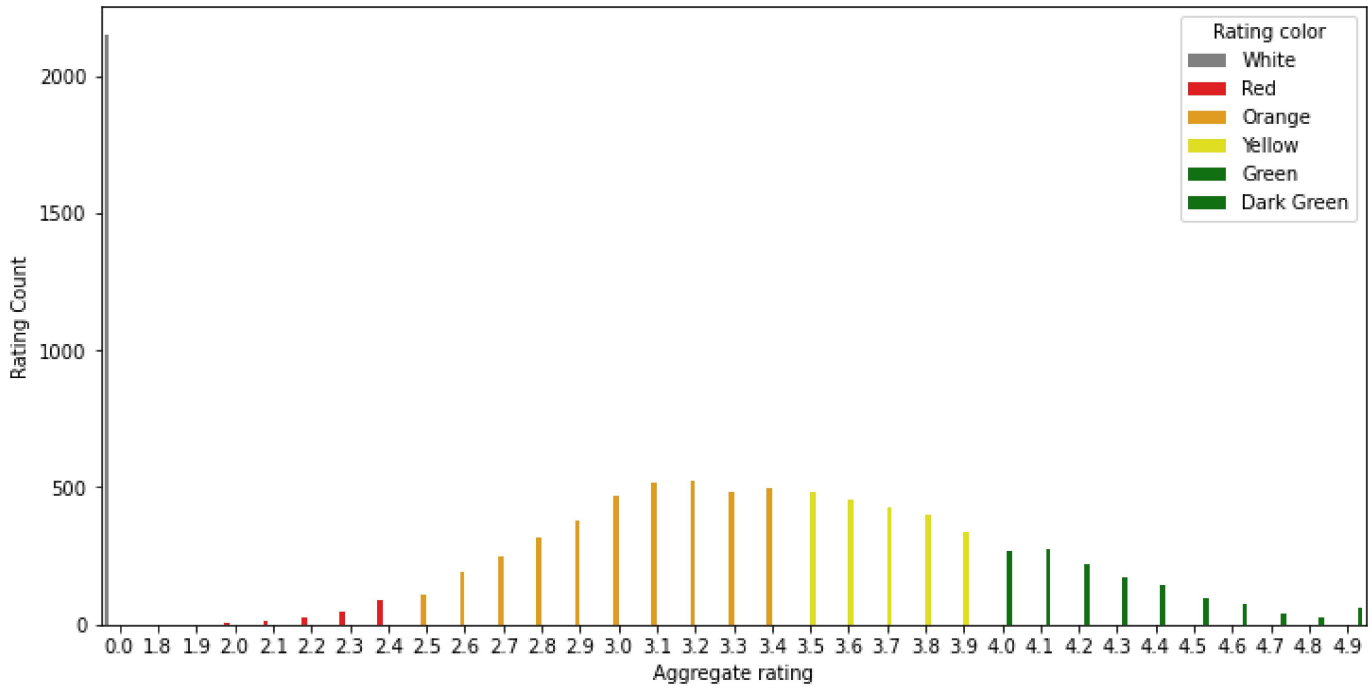
Out[19]:

	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

Observation

- When the ratings are between 4.5 to 4.9 --> Excellent
- When the ratings are between 4.0 to 4.4 --> very good
- When the ratings are between 3.5 to 3.9 --> good
- When the ratings are between 2.5 to 3.4 --> average
- When the ratings are between 1.8 to 2.4 --> poor
- When the rating is 0 --> not rated

In [20]:
```
# import matplotlib
# matplotlib.rcParams['figure.figsize'] = (12,8)

fig, ax = plt.subplots(figsize=(12, 6))
sns.barplot(x = "Aggregate rating", y="Rating Count", data = rating, hue = 'Rating color',
plt.show()
```

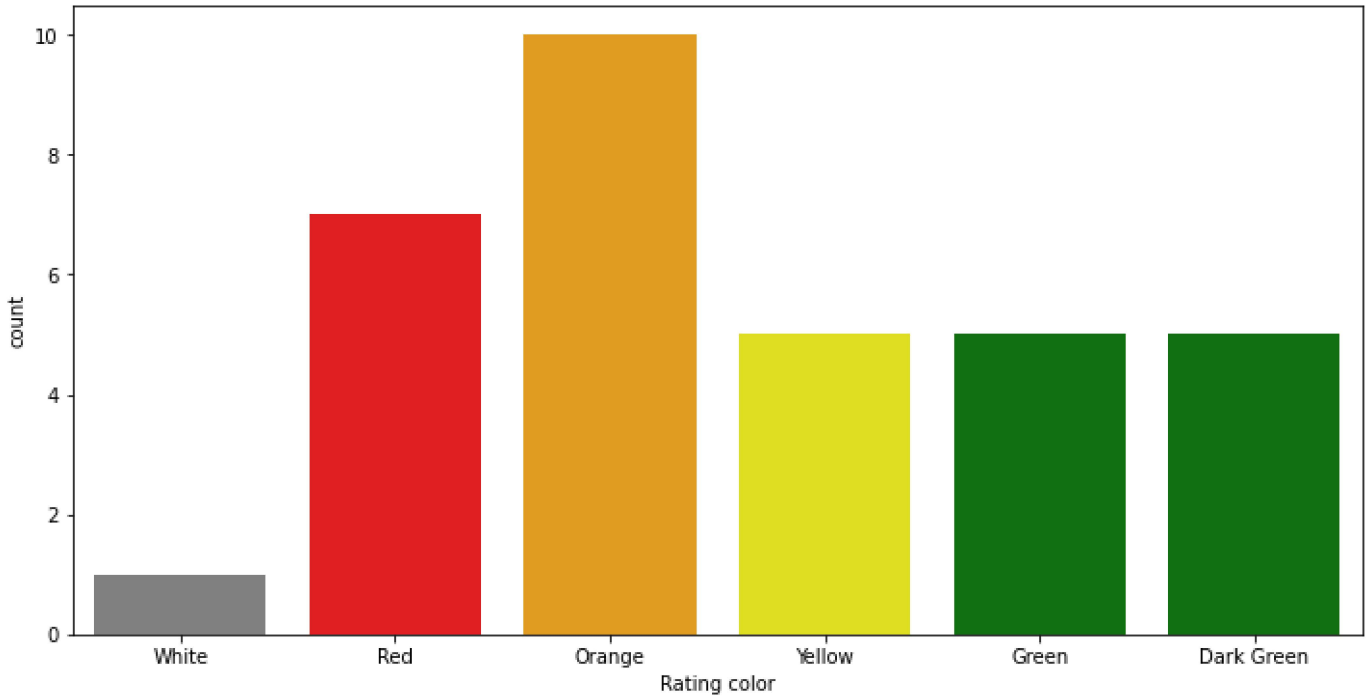


Observation

- Not rated count is very high
- Maximum number of rating are between 2.6 to 3.9

In [21]:
```
# count plot

sns.countplot(x="Rating color",data = rating, palette=['grey','red','orange','yellow','gree
```

Out[21]:
```
<AxesSubplot:xlabel='Rating color', ylabel='count'>
```



In [22]:
```
df_final.head()
```

Out[22]:

	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

5 rows × 22 columns

In [23]:
```
df_final.columns
```

Out[23]:
```
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')
```

In [24]:
```
# find the countries which has given zero rating

df_final[df_final["Aggregate rating"]==0].groupby("Country").size().reset_index()
```

Out[24]:

	Country	0
0	Brazil	5
1	India	2139
2	United Kingdom	1
3	United States	3

In [25]:
```
df_final.groupby(["Aggregate rating","Country"]).size().reset_index().head(4)
```

Out[25]:

	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

Observatoin

- Maximum number of zero rating are from Indian customers

In [26]:
```
# Find out country wise currency
df_final.groupby(["Currency","Country"]).size().reset_index()
```

Out[26]:

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

In [27]:
```
df_final[["Currency","Country"]].groupby(["Currency","Country"]).count().reset_index()
```

Out[27]:

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

In [28]:
```
#find which country do have online delivery option

df_final[df_final['Has Online delivery']=='Yes'].groupby('Country').size().reset_index()
```

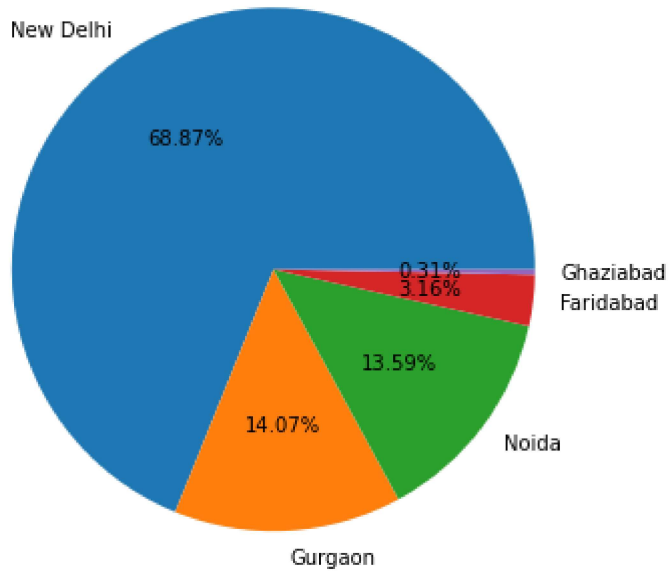
Out[28]:

	Country	0
0	India	2423
1	UAE	28

Observation

- Online deliveries are available in India and UAE

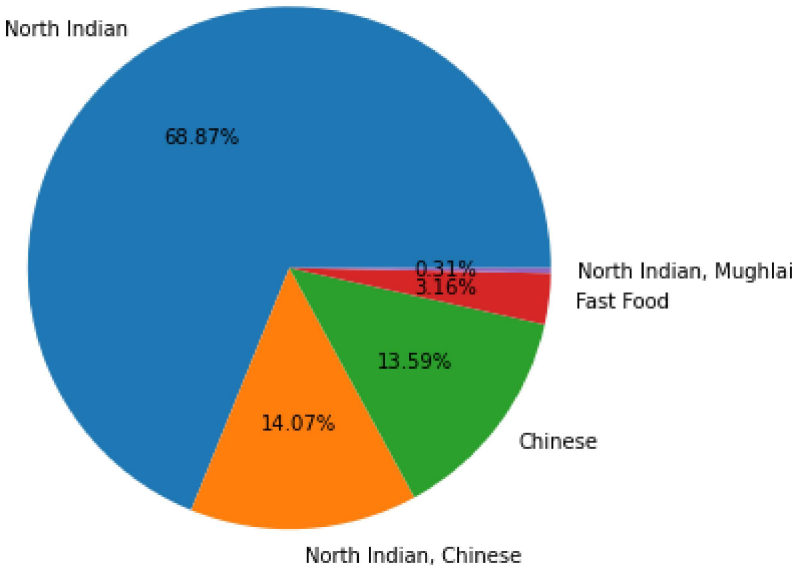
In [29]:
```
## Create a pie chart for top 5 cities distribution

city_names = df_final.City.value_counts().index
city_values = df_final.City.value_counts().values

fig, ax = plt.subplots(figsize=(12, 6))
plt.pie(x=city_values[:5], labels=city_names[:5], autopct='%1.2f%%')
plt.show()
```



In [30]:
```
# Find top 10 Cuisines
df_final.Cuisines.value_counts().head(10)
```

Out[30]:
```
North Indian                        936
North Indian, Chinese               511
Chinese                             354
Fast Food                           354
North Indian, Mughlai               334
Cafe                                299
Bakery                              218
North Indian, Mughlai, Chinese      197
Bakery, Desserts                    170
Street Food                         149
Name: Cuisines, dtype: int64
```

In [31]:
```
#top 5 cuisines

Cuisine_names = df_final.Cuisines.value_counts().head(10).index
Cuisine_values = df_final.City.value_counts().values

fig, ax = plt.subplots(figsize=(12, 6))
plt.pie(x=Cuisine_values[:5], labels=Cuisine_names[:5], autopct='%1.2f%%')
plt.show()
```



In [ ]: