

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

```
In [2]: df = pd.read_csv("C://Users//sande//Downloads//archive//zomato.csv", encoding='latin-1')
```

```
In [3]: df.head()
```

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	I On deliv
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenue...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)	Yes	
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	
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	
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	
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	

5 rows × 21 columns

```
In [4]: df.columns
```

```
Out[4]: 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 [5]: df.shape
```

```
Out[5]: (9551, 21)
```

```
In [6]: 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   LocalityVerbose   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 [7]: `df.describe()`

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156.909748
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430.169145
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5.000000
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31.000000
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

What all things we do in data analysis

1. Checking the missing values
2. Explore about numerical variables
3. Explore about categorical variables
4. Finding relationship between features

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

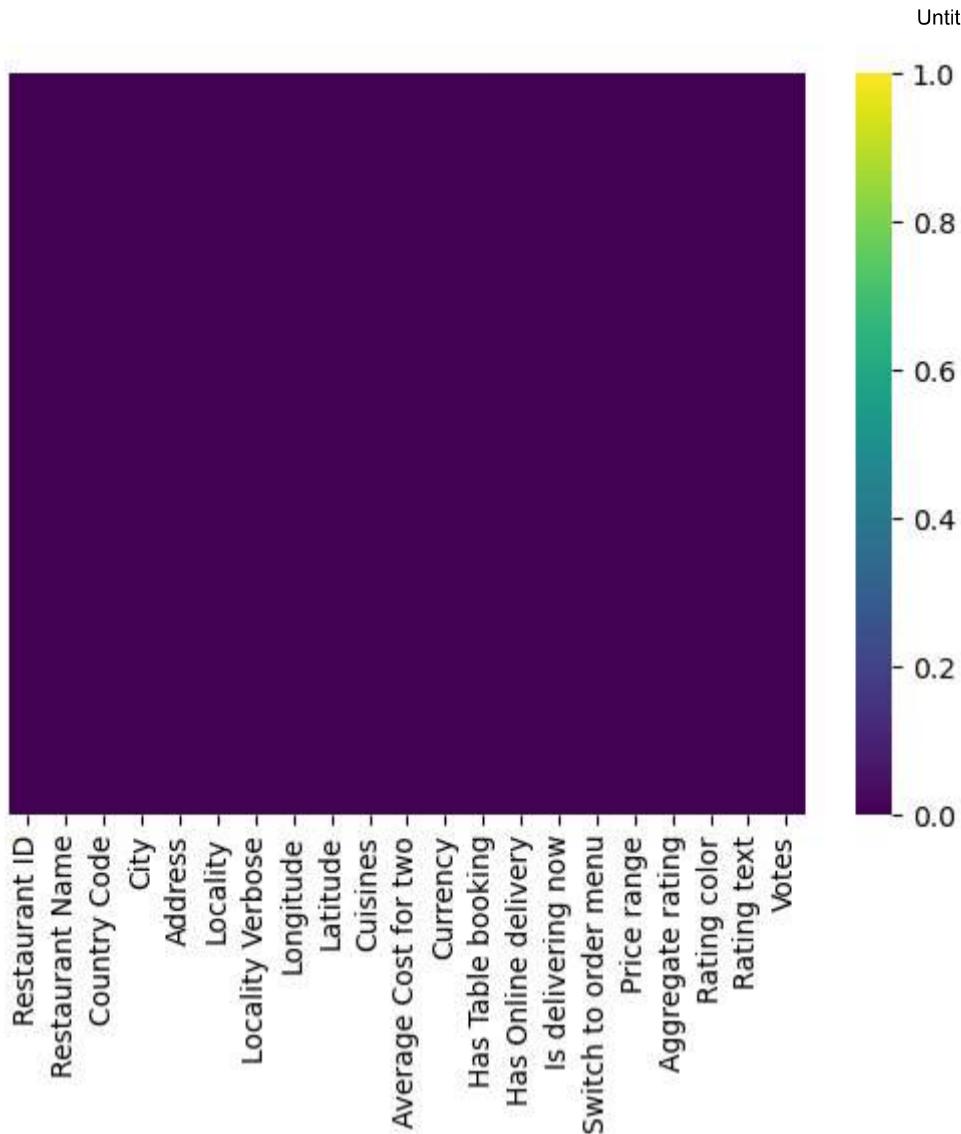
```
Restaurant ID      0
Restaurant Name    0
Country Code       0
City               0
Address            0
Locality           0
LocalityVerbose    0
Longitude          0
Latitude           0
Cuisines           9
Average Cost for two 0
Currency           0
Has Table booking  0
Has Online delivery 0
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 [9]: `[features for features in df.columns if df[features].isnull().sum() > 0]`

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

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

Out[10]: `<Axes: >`



```
In [13]: df_country= pd.read_excel("C://Users//sande//Downloads//archive//Country-Code.xlsx")
df_country.head
```

```
Out[13]: <bound method NDFrame.head of
   Country Code          Country
0           1            India
1          14        Australia
2          30         Brazil
3          37        Canada
4          94      Indonesia
5         148     New Zealand
6         162    Phillipines
7         166          Qatar
8         184      Singapore
9         189    South Africa
10        191      Sri Lanka
11        208          Turkey
12        214          UAE
13        215  United Kingdom
14        216  United States>
```

```
In [14]: df.columns
```

```
Out[14]: 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 [17]: #comining the both dataframes usint country code feature
Final_df=pd.merge(df,df_country, on = "Country Code", how="left")
```

```
In [18]: Final_df.head(2)
```

Out[18]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu
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	...	Yes	No	No	No
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	...	Yes	No	No	No

2 rows × 22 columns

In [21]: Final_df.dtypes

```
Out[21]:
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
```

In [22]: final_df.columns

```
Out[22]:
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 [23]: #Zomato is mostly available in India, Zomato is basically earning more in India
Final_df.Country.value_counts()

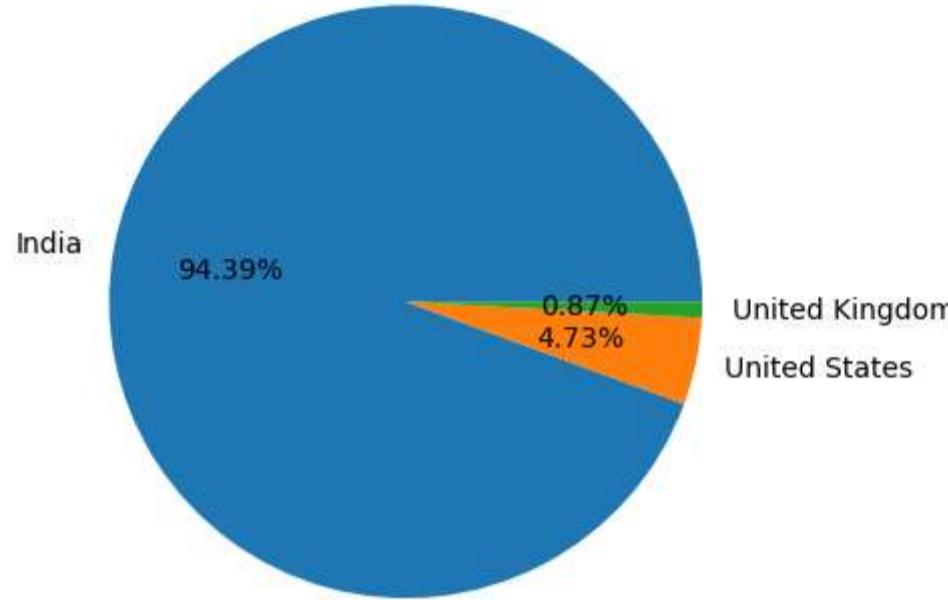
```
Out[23]:
India          8652
United States  434
United Kingdom 80
Brazil         60
UAE            60
South Africa   60
New Zealand    40
Turkey          34
Australia       24
Phillipines     22
Indonesia       21
Singapore       20
Qatar           20
Sri Lanka       20
Canada          4
Name: Country, dtype: int64
```

In [26]: country_names = Final_df.Country.value_counts().index

In [27]: country_val=Final_df.Country.value_counts().values

In [29]: #Pie chart- top 3 countries that uses zomato
plt.pie(country_val[:3], labels = country_names[:3], autopct= "%1.2f%%")

```
Out[29]: ([<matplotlib.patches.Wedge at 0x21a20055330>,
 <matplotlib.patches.Wedge at 0x21a20055240>,
 <matplotlib.patches.Wedge at 0x21a20055fc0>],
 [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: India has the maximum number of orders, India is the business hub for Zomato and then USA is on the 2nd position in terms of placing the orders on zomato and UK is on the 3rd position

```
In [30]: Final_df.columns
```

```
Out[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')
```

```
In [32]: ratings=Final_df.groupby(["Aggregate rating","Rating color","Rating text"]).size().reset_index().rename(columns={0:"Rating count"})
```

```
In [33]: ratings
```

Out[33]:

	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

Observations

1. Maximum number of people didn't give ratings
2. When Rating is between 4.5 to 4.9 --> Excellent
3. When Rating is between 4.0 to 4.4 --> Very Good
4. When Rating is between 3.5 to 3.9 --> Good
5. When Rating is between 2.5 to 3.4 --> Average
6. When Rating is between 1.8 to 2.4 --> Poor
7. When Rating is 0.0 --> Not Rated

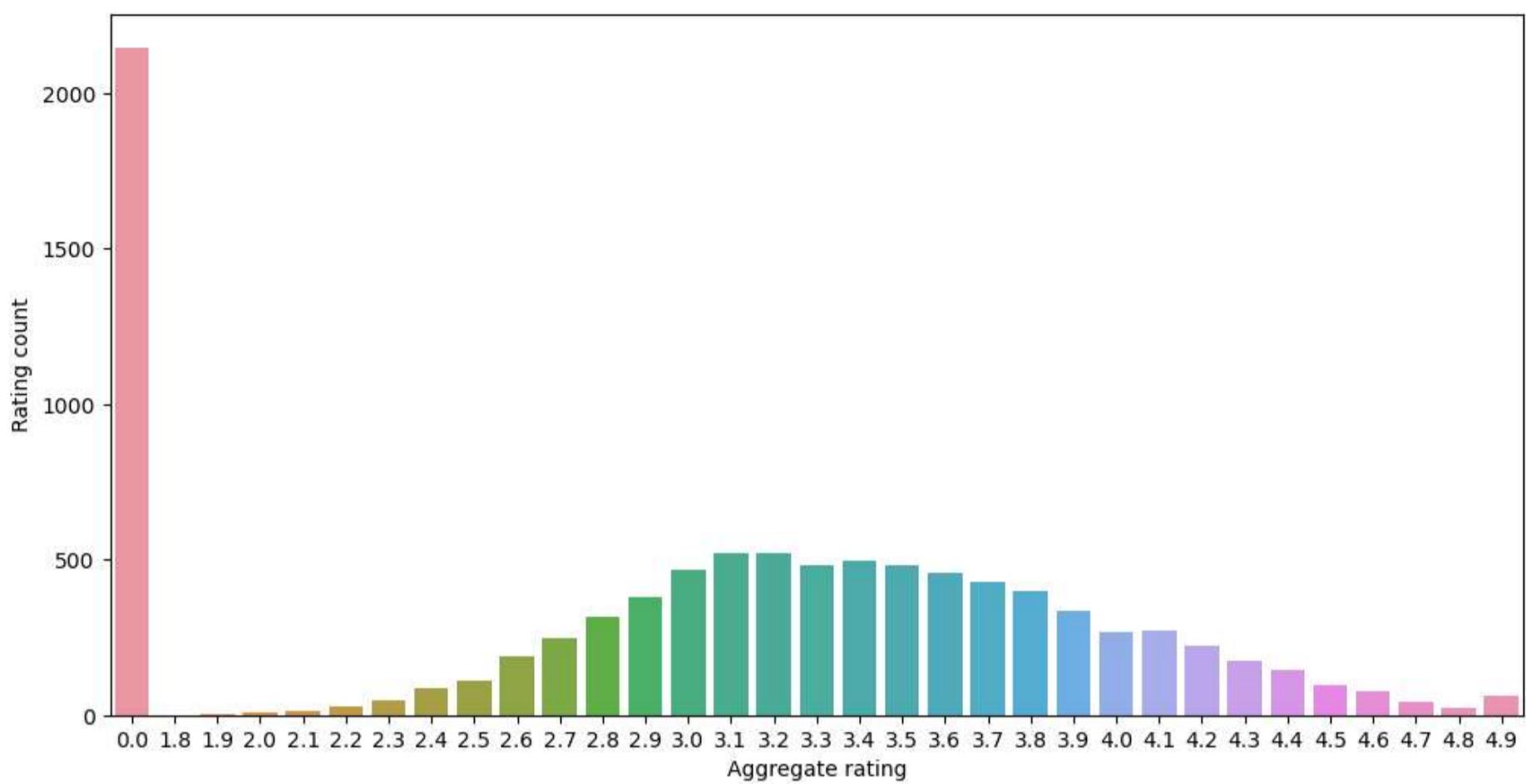
In [34]: `ratings.head()`

Out[34]:

	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

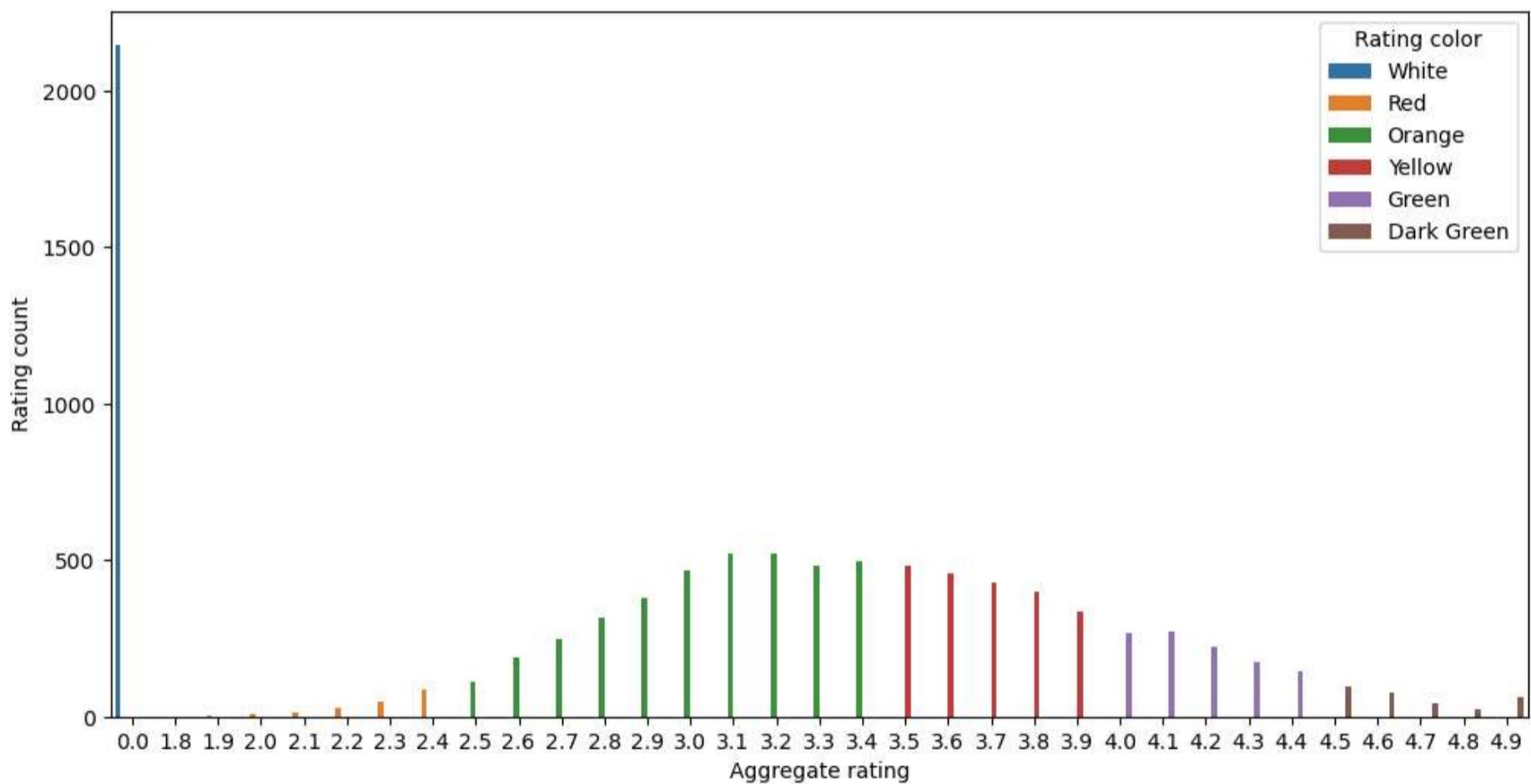
In [39]: `import matplotlib
matplotlib.rcParams["figure.figsize"] = (12, 6)
sns.barplot(x="Aggregate rating", y="Rating count", data=ratings)`

Out[39]: <Axes: xlabel='Aggregate rating', ylabel='Rating count'>



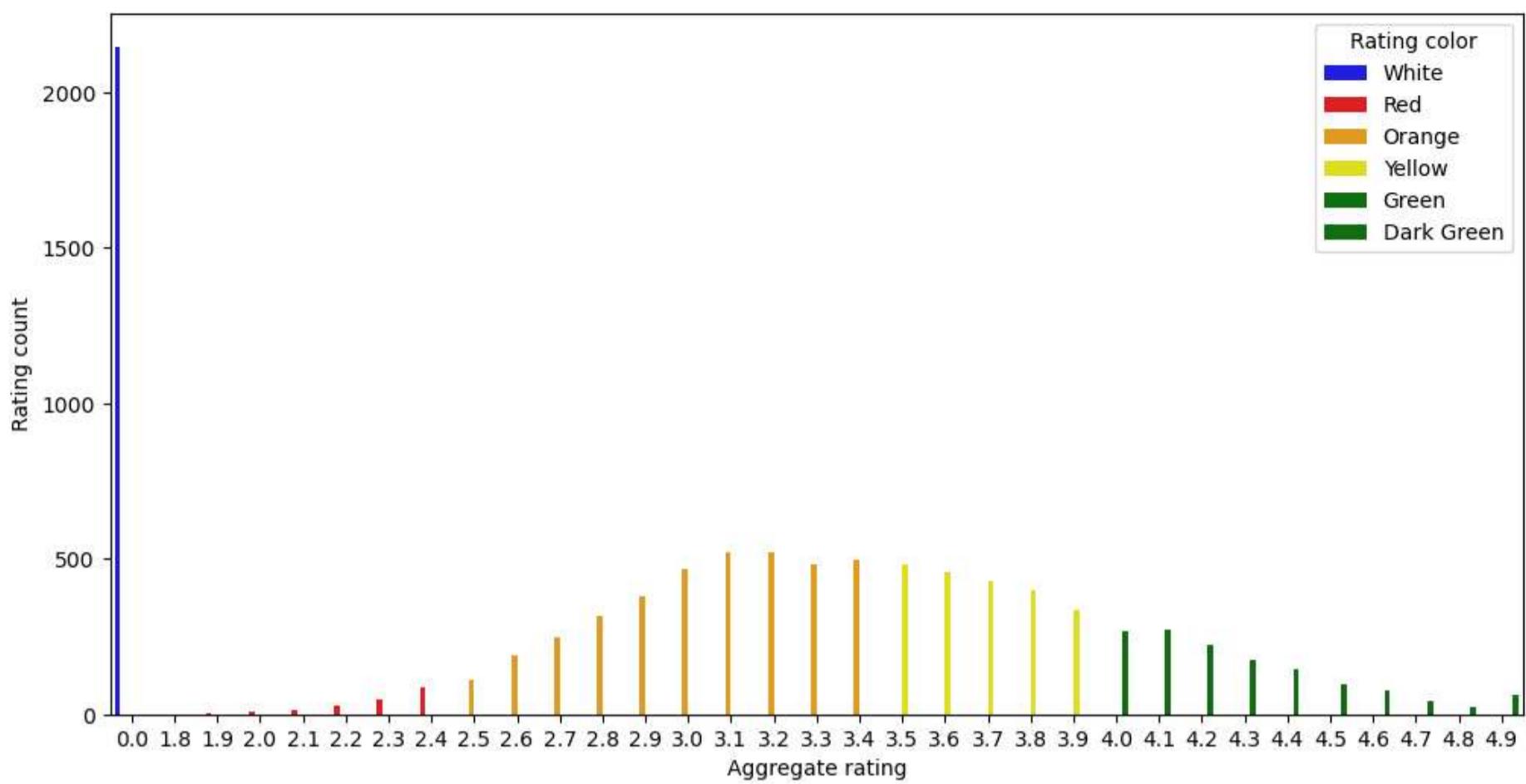
In [40]: `sns.barplot(x="Aggregate rating", y="Rating count", hue="Rating color", data=ratings)`

Out[40]: <Axes: xlabel='Aggregate rating', ylabel='Rating count'>



In [48]: `sns.barplot(x= "Aggregate rating", y= "Rating count", hue= "Rating color", data=ratings, palette = ["blue", "red", "orange", "yellow", "green", "darkgreen"])`

Out[48]: <Axes: xlabel='Aggregate rating', ylabel='Rating count'>

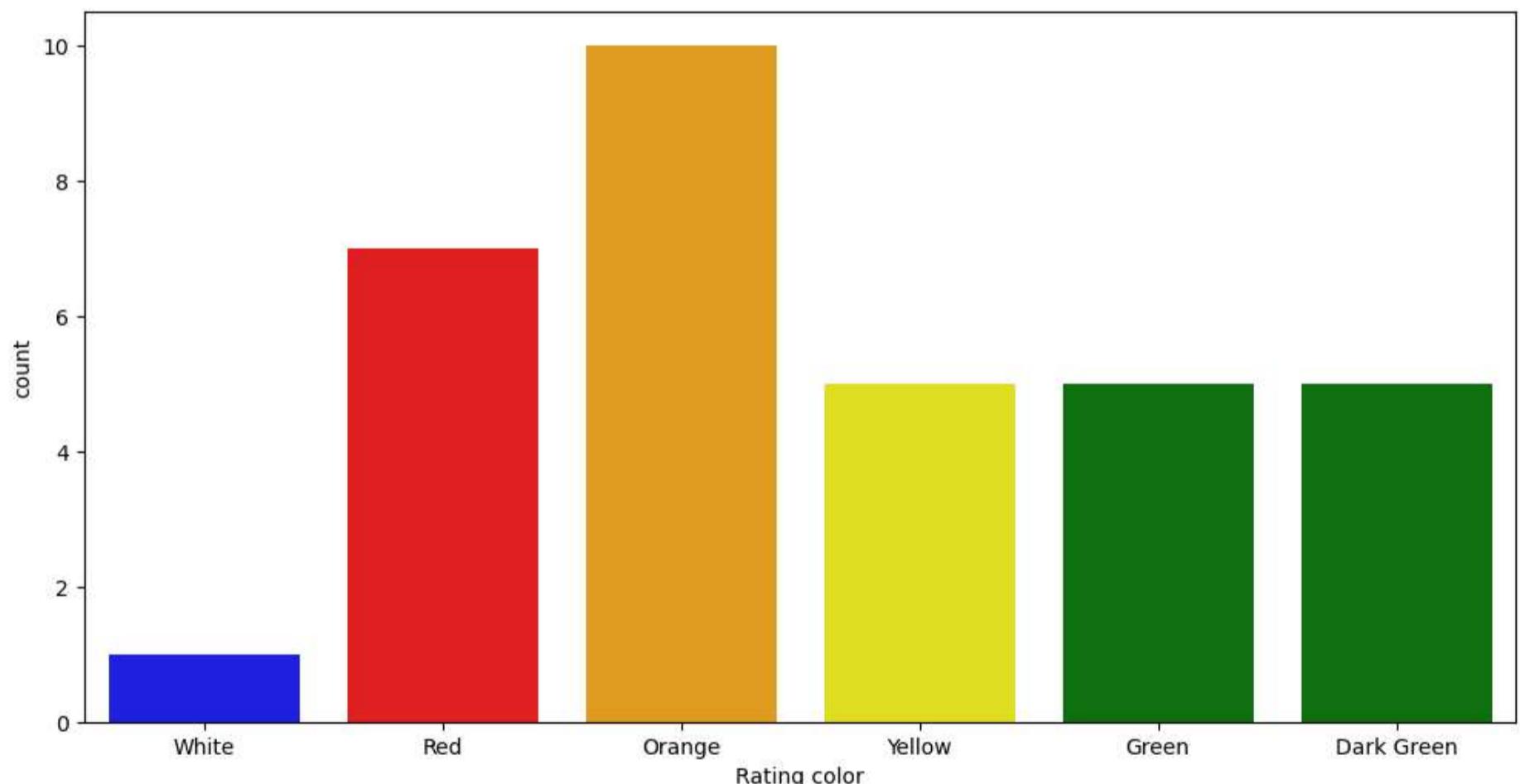


Observations:

1. Not Rated count is very high
2. Maximum numbers of ratings are between 2.5 and 3.4

```
In [49]: sns.countplot(x="Rating color", data = ratings, palette= ["blue","red", "orange", "yellow", "green","green"])
```

```
Out[49]: <Axes: xlabel='Rating color', ylabel='count'>
```



```
In [50]: ratings
```

Out[50]:

	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

In [52]: `## Find the countries that has given 0 rating
Final_df.groupby(["Aggregate rating", "Country"]).size().reset_index().head(7)`

Out[52]:

	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
5	1.9	India	2
6	2.0	India	7

Observations: Maximum number of 0 ratings are given by Indian customers

In [53]: `Final_df[final_df["Aggregate rating"]== 0.0].groupby("Country").size().reset_index()`

Out[53]:

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

Find out which currency is used by which country?

```
In [55]: Final_df.groupby(["Country", "Currency"]).size().reset_index()
```

	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	New Zealand(\$)	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

```
In [56]: Final_df[["Country", "Currency"]].groupby(["Country", "Currency"]).size().reset_index()
```

	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	New Zealand(\$)	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

Which country has online delivery?

```
In [57]: Final_df[["Country", "Has Online delivery"]].groupby(["Has Online delivery", "Country"]).size().reset_index()
```

	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

```
In [58]: Final_df[final_df["Has Online delivery"]=="Yes"].Country.value_counts()
```

```
Out[58]: India    2423
         UAE     28
         Name: Country, dtype: int64
```

```
In [59]: Final_df.columns
```

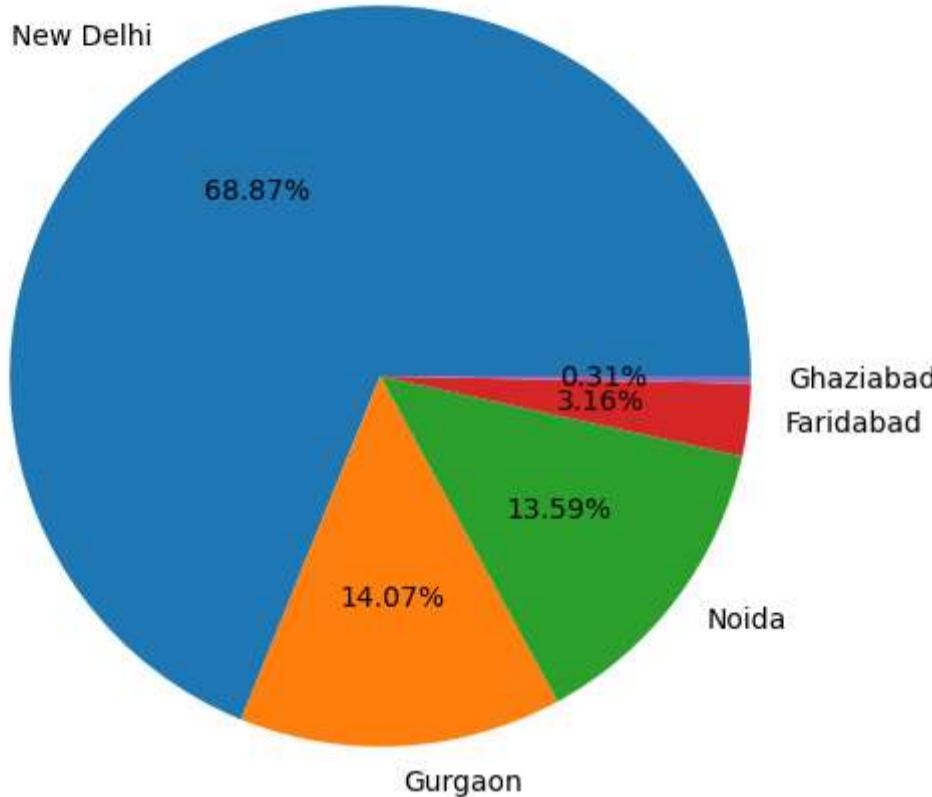
```
Out[59]: Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes', 'Country'],
      dtype='object')
```

```
In [60]: city_values= Final_df.City.value_counts().values
city_names= Final_df.City.value_counts().index
```

Pie chart- top 5 cities distribution

```
In [61]: plt.pie(city_values[:5], labels= city_names[:5], autopct= "%1.2f%%")
```

```
Out[61]: ([<matplotlib.patches.Wedge at 0x21a27cdcae60>,
 <matplotlib.patches.Wedge at 0x21a27cdad70>,
 <matplotlib.patches.Wedge at 0x21a27cdb9d0>,
 <matplotlib.patches.Wedge at 0x21a279340a0>,
 <matplotlib.patches.Wedge at 0x21a27934730>],
 [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%')])
```



Observations: Maximum number of orders are placed in New Delhi.

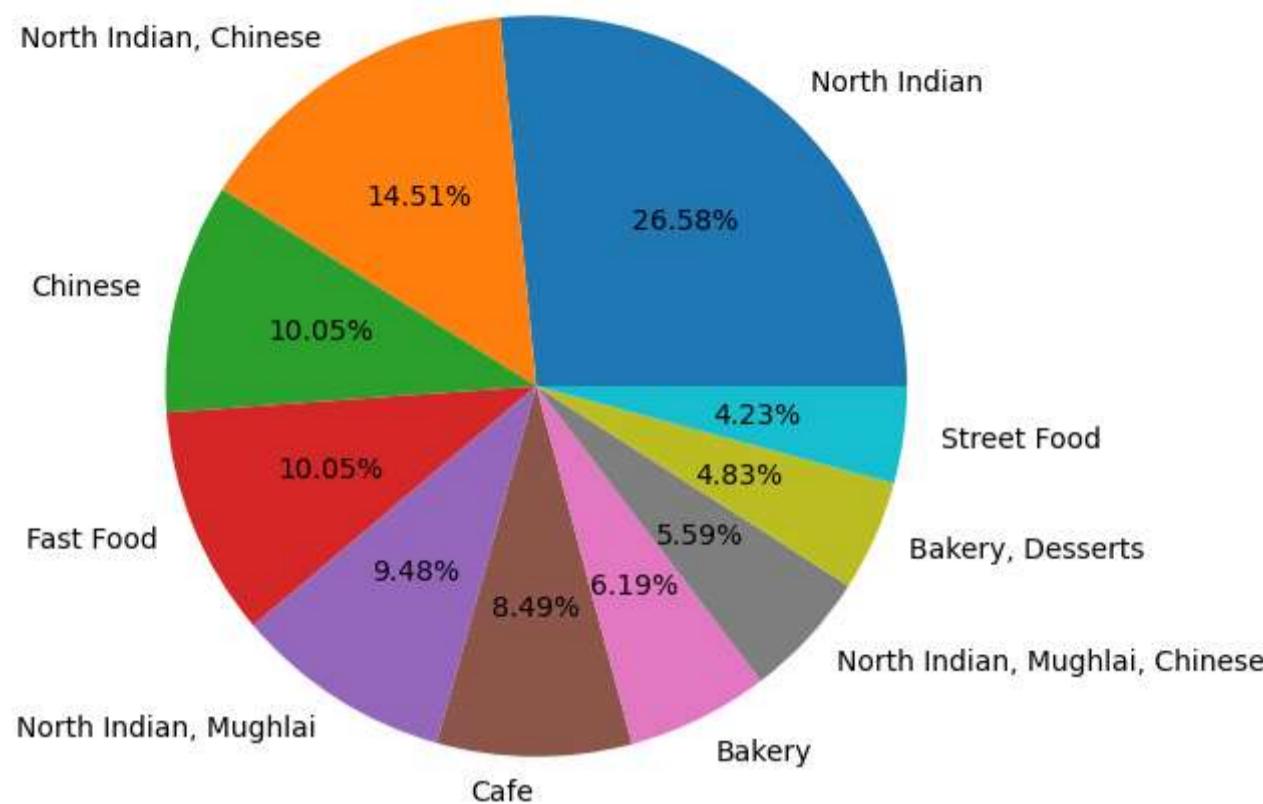
Find the top 10 cuisines

```
In [62]: cuisines_count = Final_df.Cuisines.value_counts().values
cuisines_names = Final_df.Cuisines.value_counts().index
print(cuisines_count)
print(cuisines_names)

[936 511 354 ... 1 1 1]
Index(['North Indian', 'North Indian, Chinese', 'Chinese', 'Fast Food',
       'North Indian, Mughlai', 'Cafe', 'Bakery',
       'North Indian, Mughlai, Chinese', 'Bakery, Desserts', 'Street Food',
       ...
       'Cafe, Pizza, Burger',
       'Healthy Food, Continental, Juices, Beverages, Italian, Salad, Lebanese',
       'Goan, American, Portuguese', 'South Indian, Desserts, Beverages',
       'Healthy Food, North Indian, Italian, Salad', 'Bengali, Fast Food',
       'North Indian, Rajasthani, Asian',
       'Chinese, Thai, Malaysian, Indonesian',
       'Bakery, Desserts, North Indian, Bengali, South Indian',
       'Italian, World Cuisine'],
      dtype='object', length=1825)
```

```
In [63]: plt.pie(cuisines_count[:10], labels=cuisines_names[:10], autopct="%1.2f%%")
```

```
Out[63]: ([<matplotlib.patches.Wedge at 0x21a2796b490>,
 <matplotlib.patches.Wedge at 0x21a2796b3a0>,
 <matplotlib.patches.Wedge at 0x21a27994040>,
 <matplotlib.patches.Wedge at 0x21a279946d0>,
 <matplotlib.patches.Wedge at 0x21a27994d60>,
 <matplotlib.patches.Wedge at 0x21a279953f0>,
 <matplotlib.patches.Wedge at 0x21a27995a80>,
 <matplotlib.patches.Wedge at 0x21a27996110>,
 <matplotlib.patches.Wedge at 0x21a279967a0>,
 <matplotlib.patches.Wedge at 0x21a27996e30>],
[Text(0.7383739846958008, 0.8153550507137645, 'North Indian'),
 Text(-0.5794679314239953, 0.9349956772366362, 'North Indian, Chinese'),
 Text(-1.067309479615702, 0.26617752482593154, 'Chinese'),
 Text(-1.0185984499802057, -0.4152796620326146, 'Fast Food'),
 Text(-0.5935788454809928, -0.9261015895664211, 'North Indian, Mughlai'),
 Text(-0.005887079599915552, -1.0999842463843672, 'Cafe'),
 Text(0.4842062514572988, -0.9876964645323336, 'Bakery'),
 Text(0.808736477166136, -0.7456174022251013, 'North Indian, Mughlai, Chinese'),
 Text(1.0055375294202338, -0.44597564611473206, 'Bakery, Desserts'),
 Text(1.090298995560443, -0.14576728123927227, 'Street Food')],
[Text(0.4027494461977095, 0.4447391185711442, '26.58%'),
 Text(-0.316073417140361, 0.5099976421290743, '14.51%'),
 Text(-0.5821688070631101, 0.14518774081414446, '10.05%'),
 Text(-0.5555991545346576, -0.22651617929051704, '10.05%'),
 Text(-0.32377027935326874, -0.5051463215816842, '9.48%'),
 Text(-0.003211134327226664, -0.5999914071187457, '8.49%'),
 Text(0.26411250079489024, -0.5387435261085456, '6.19%'),
 Text(0.441128987545165, -0.40670040121369155, '5.59%'),
 Text(0.5484750160474001, -0.24325944333530836, '4.83%'),
 Text(0.5947085430329688, -0.07950942613051214, '4.23%')])
```



Observations: People like the North Indian food the most.