
Level 1

Task 1 : Data Exploration and Preprocessing

```
In [ ]: # importing the Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import folium

import warnings
warnings.filterwarnings('ignore')
```

```
In [ ]: # reading dataset

retail=pd.read_csv('Dataset .csv')
retail
```

Out[]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City
...
9546	5915730	Naml Gurme	208	İstanbul	Kemankeş Karamustafa Paşası Mahallesi, Rıhtım ...	Karaköy
9547	5908749	Ceviz Aca	208	İstanbul	Koşuyolu Mahallesi, Muhittin ... Cadd...	Koşuyolu
9548	5915807	Huqqa	208	İstanbul	Kuruçeşme Mahallesi, Muallim Naci Caddesi, N...	Kuruçeşme
9549	5916112	Ak Kahve	208	İstanbul	Kuruçeşme Mahallesi, Muallim Naci Caddesi, N...	Kuruçeşme
9550	5927402	Walter's Coffee Roastery	208	İstanbul	Cafea Mahallesi, Bademaltı Sokak, No 21/B, ...	Moda

9551 rows × 21 columns

```
In [ ]: # printing the first 5 rows of the dataset

retail.head()
```

Out[]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Localit Verbos
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century Cit Ma Poblacion Makati City Mak
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
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, Ortiga Mandaluyon City, Ma
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megama Ortiga Mandaluyon Cit, Mandal
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megama Ortiga Mandaluyon Cit, Mandal

5 rows × 21 columns



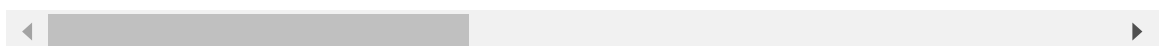
```
In [ ]: # printing sample rows of the dataset

retail.sample(5)
```

Out[]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose
6723	8771	Ashu Bhature Wala	1	New Delhi	B-6/150, Sector 8, Rohini, New Delhi	Rohini	Rohini, New Delhi
2447	15705	Kitchen At 95 - Hyatt Regency	1	Ludhiana	Hyatt Regency, Site 4, Ferozepur Road, Rajguru...	Hyatt Regency, Rajguru Nagar	Hyatt Regency, Rajguru Nagar, Ludhiana
6281	18350101	The Taste of Delhi	1	New Delhi	Shop 2, 4, 6, & 7, CD Block, Sagar Complex, Pi...	Pitampura	Pitampura, New Delhi
5032	311067	Chatori Zubaan 2	1	New Delhi	Near Nike Showroom, Rangpuri, NH-8, Mahipalpur...	Mahipalpur	Mahipalpur, New Delhi
7234	311364	Believe in Taste	1	New Delhi	G-19, BB Block, Vardhaman Complex, Shalimar Ba...	Shalimar Bagh	Shalimar Bagh, New Delhi

5 rows × 21 columns

In []: *# dimensions of the dataset*

retail.shape

Out[]: (9551, 21)

In []: *# returns the concise summary of the dataset*

retail.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 [ ]: # displaying the number of unique data in each column

retail.nunique()
```

```
Out[ ]: 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
```

```
In [ ]: # checking for duplicates in the dataset

retail.duplicated().sum()
```

Out[]: 0

```
In [ ]: # checking for missing data  
  
retail.isnull().sum()
```

```
Out[ ]: 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
```

From this we understood that the column "Cuisines" has 9 missing values

```
In [ ]: # Calculate the mode of the 'Cuisines' column  
mode_cuisines = retail['Cuisines'].mode()[0]  
mode_cuisines
```

Out[]: 'North Indian'

```
In [ ]: # Replace missing values in the 'Cuisines' column with the mode  
retail['Cuisines'].fillna(mode_cuisines, inplace=True)
```

```
In [ ]: retail
```

Out[]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City
...
9546	5915730	Naml Gurme	208	istanbul	Kemanke Karamustafa Pa Mahallesi, Rihlm ...	Karak_y
9547	5908749	Ceviz Aacl	208	istanbul	Ko uyolu Mahallesi, Muhittin st_nda Cadd...	Ko uyolu
9548	5915807	Huqqa	208	istanbul	Kuru_e me Mahallesi, Muallim Naci Caddesi, N...	Kuru_e me
9549	5916112	Ak Kahve	208	istanbul	Kuru_e me Mahallesi, Muallim Naci Caddesi, N...	Kuru_e me
9550	5927402	Walter's Coffee Roastery	208	istanbul	Cafea Mahallesi, Bademalt Sokak, No 21/B, ...	Moda

9551 rows × 21 columns

```
In [ ]: retail.isnull().sum()
```

```
Out[ ]: Restaurant ID      0
        Restaurant Name   0
        Country Code     0
        City              0
        Address           0
        Locality          0
        Locality Verbose  0
        Longitude         0
        Latitude          0
        Cuisines           0
        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
```

Now, all the missing values in the column "Cuisines" is replaced with the most frequent value in that same column.

```
In [ ]: # Check data types
        print(retail.dtypes)
```

```
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
dtype: object
```

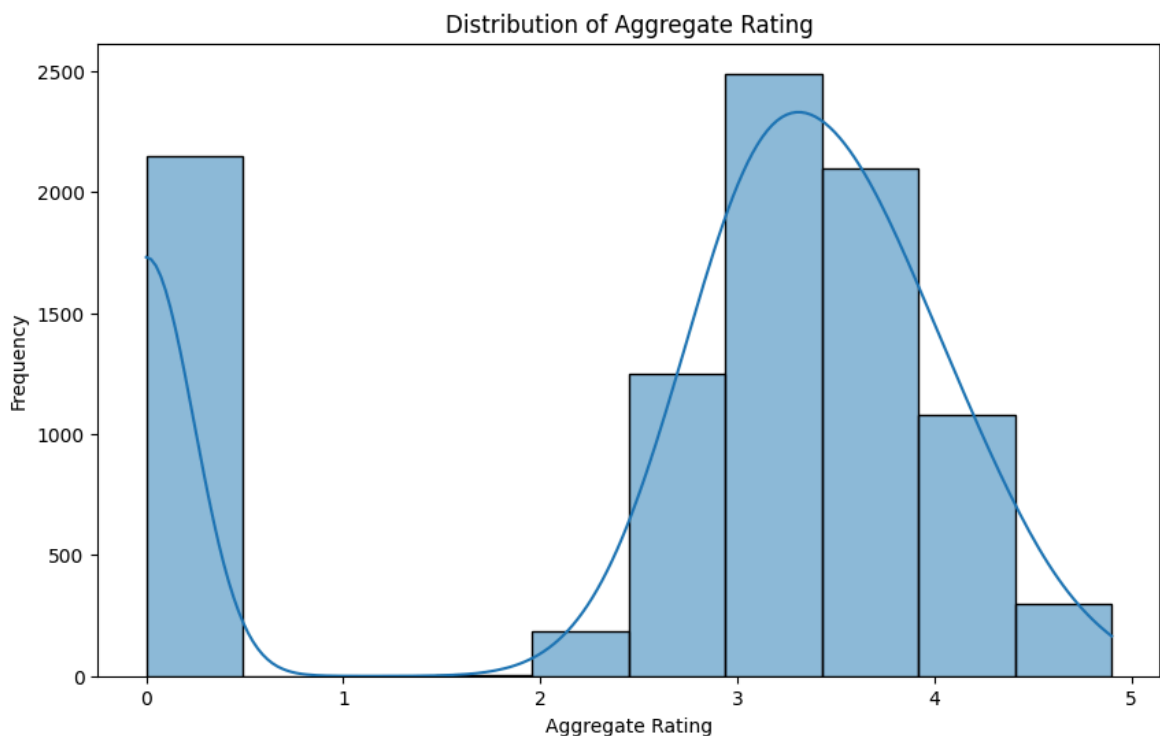
```
In [ ]: # Display summary statistics to further inspect the 'Aggregate rating' column
```



```
print(retail['Aggregate rating'].describe())
```

```
count    9551.000000
mean      2.666370
std       1.516378
min       0.000000
25%       2.500000
50%       3.200000
75%       3.700000
max       4.900000
Name: Aggregate rating, dtype: float64
```

```
In [ ]: # Analyze the distribution of the 'Aggregate rating'
plt.figure(figsize=(10, 6))
sns.histplot(retail['Aggregate rating'], bins=10, kde=True)
plt.title('Distribution of Aggregate Rating')
plt.xlabel('Aggregate Rating')
plt.ylabel('Frequency')
plt.show()
```



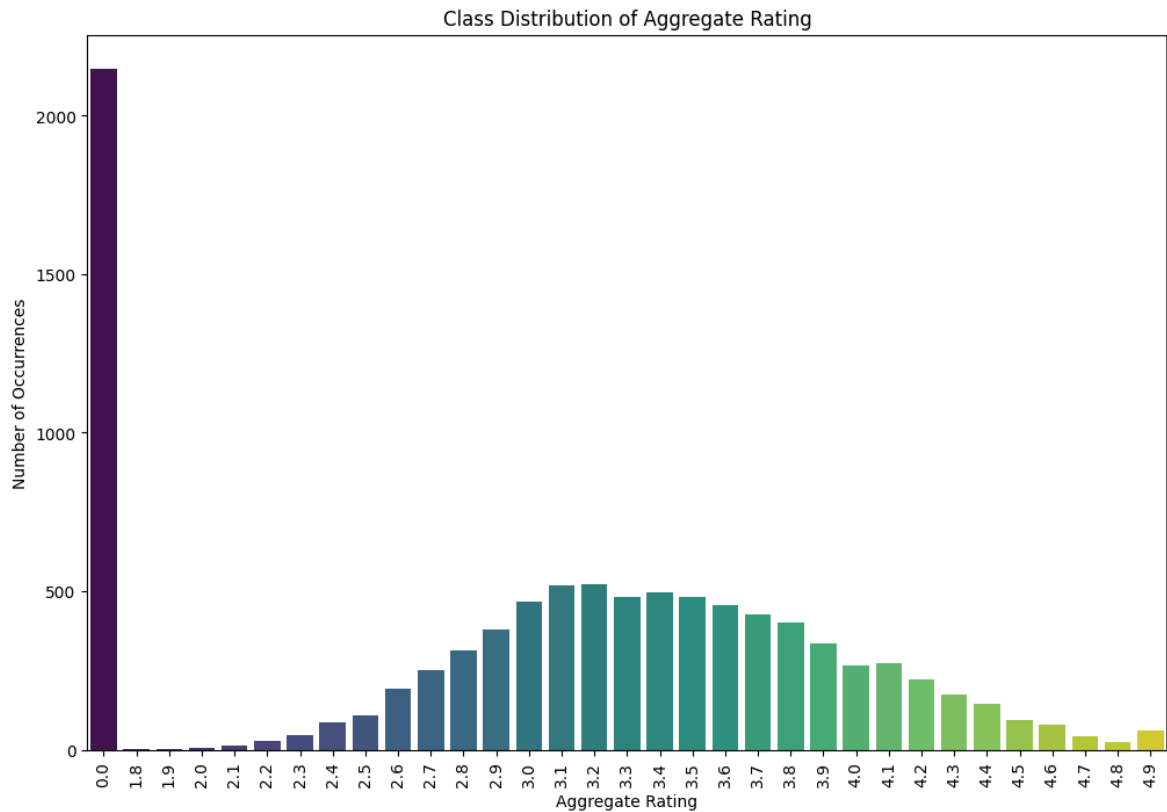
```
In [ ]: # Identify class imbalances
# We can use value_counts to see the distribution
rating_counts = retail['Aggregate rating'].value_counts().sort_index()
print(rating_counts)
```

Aggregate rating

0.0	2148
1.8	1
1.9	2
2.0	7
2.1	15
2.2	27
2.3	47
2.4	87
2.5	110
2.6	191
2.7	250
2.8	315
2.9	381
3.0	468
3.1	519
3.2	522
3.3	483
3.4	498
3.5	480
3.6	458
3.7	427
3.8	400
3.9	335
4.0	266
4.1	274
4.2	221
4.3	174
4.4	144
4.5	95
4.6	78
4.7	42
4.8	25
4.9	61

Name: count, dtype: int64

```
In [ ]: # Plot the distribution to visualize class imbalance
plt.figure(figsize=(12, 8))
sns.barplot(x=rating_counts.index, y=rating_counts.values, palette='viridis')
plt.title('Class Distribution of Aggregate Rating')
plt.xlabel('Aggregate Rating')
plt.ylabel('Number of Occurrences')
plt.xticks(rotation=90)
plt.show()
```



In []:

Task 2 : Descriptive Analysis

```
In [ ]: # returns statistical summary
retail.describe()
```

Out[]:

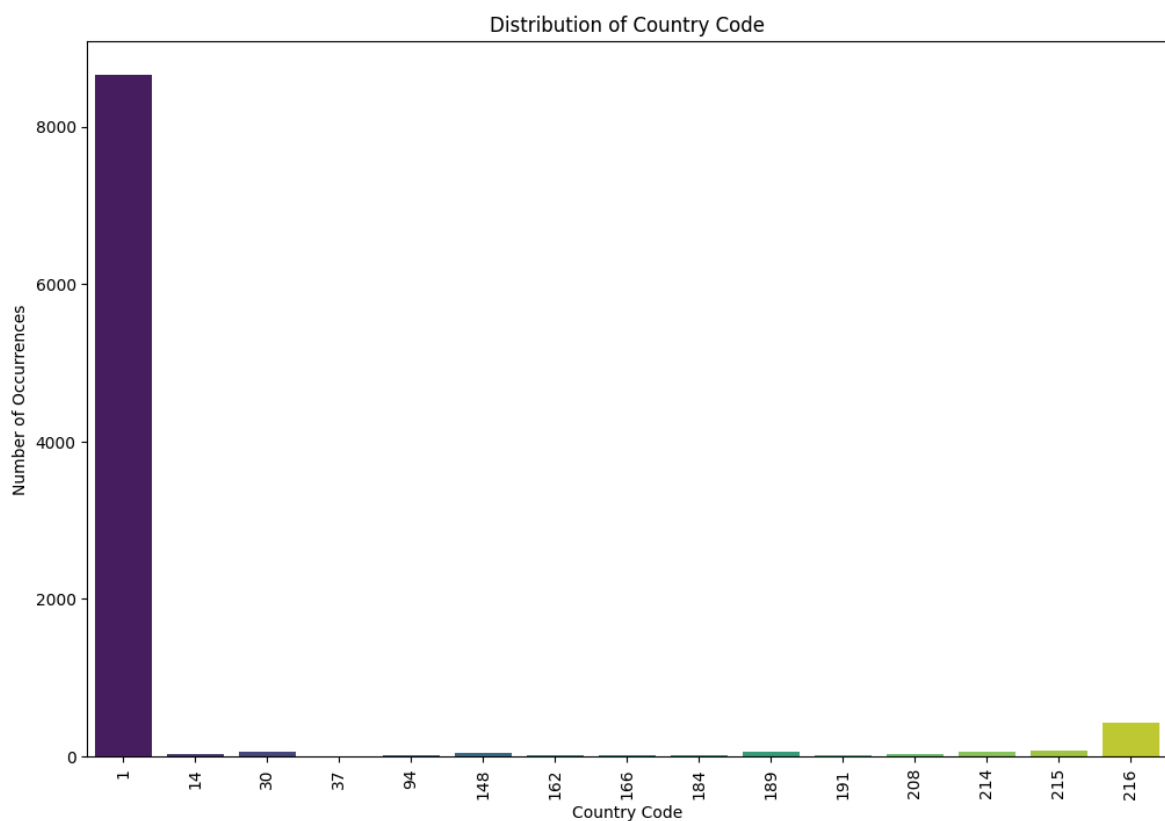
	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609
min	5.300000e+01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

```
In [ ]: # Function to plot the distribution of a categorical variable
def plot_categorical_distribution(column_name, dataset, top_n=None):
    plt.figure(figsize=(12, 8))

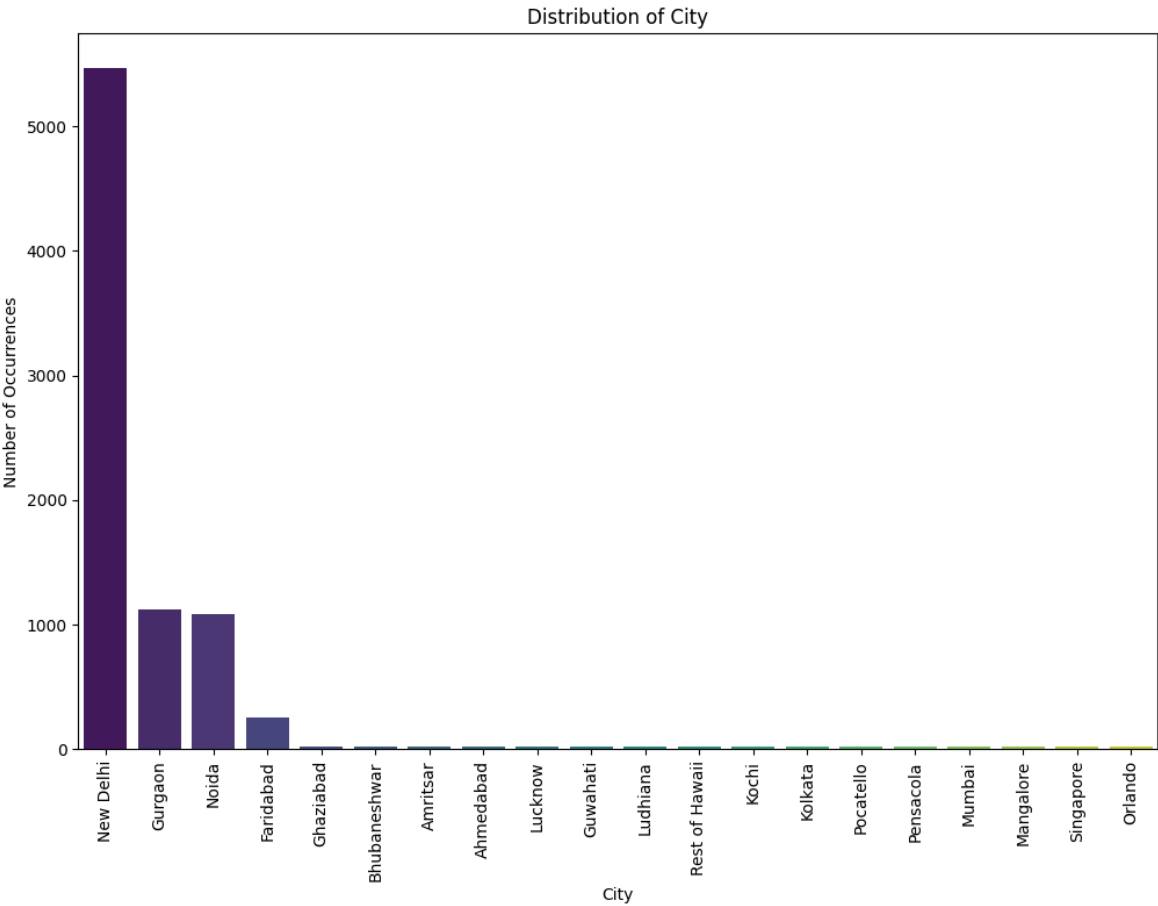
    if top_n:
        # Get the top N categories
        top_categories = dataset[column_name].value_counts().nlargest(top_n)
        sns.barplot(x=top_categories.index, y=top_categories.values, palette='vi
    else:
        category_counts = dataset[column_name].value_counts()
        sns.barplot(x=category_counts.index, y=category_counts.values, palette='

    plt.title(f'Distribution of {column_name}')
    plt.xlabel(column_name)
    plt.ylabel('Number of Occurrences')
    plt.xticks(rotation=90)
    plt.show()
```

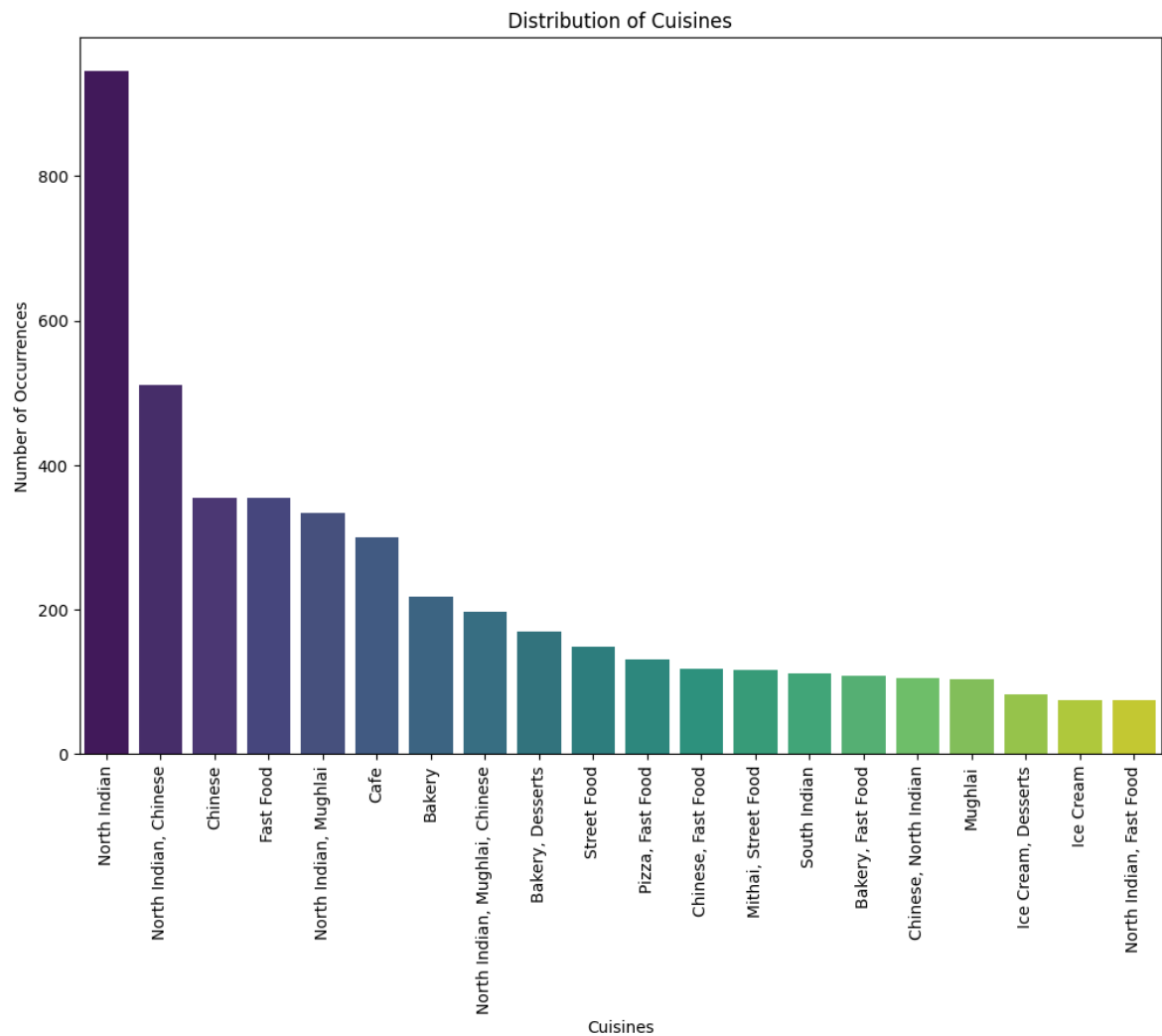
```
In [ ]: plot_categorical_distribution('Country Code', retail)
```



```
In [ ]: # Explore the distribution of "City" (limiting to top 20 for better visualization)
plot_categorical_distribution('City', retail, top_n=20)
```



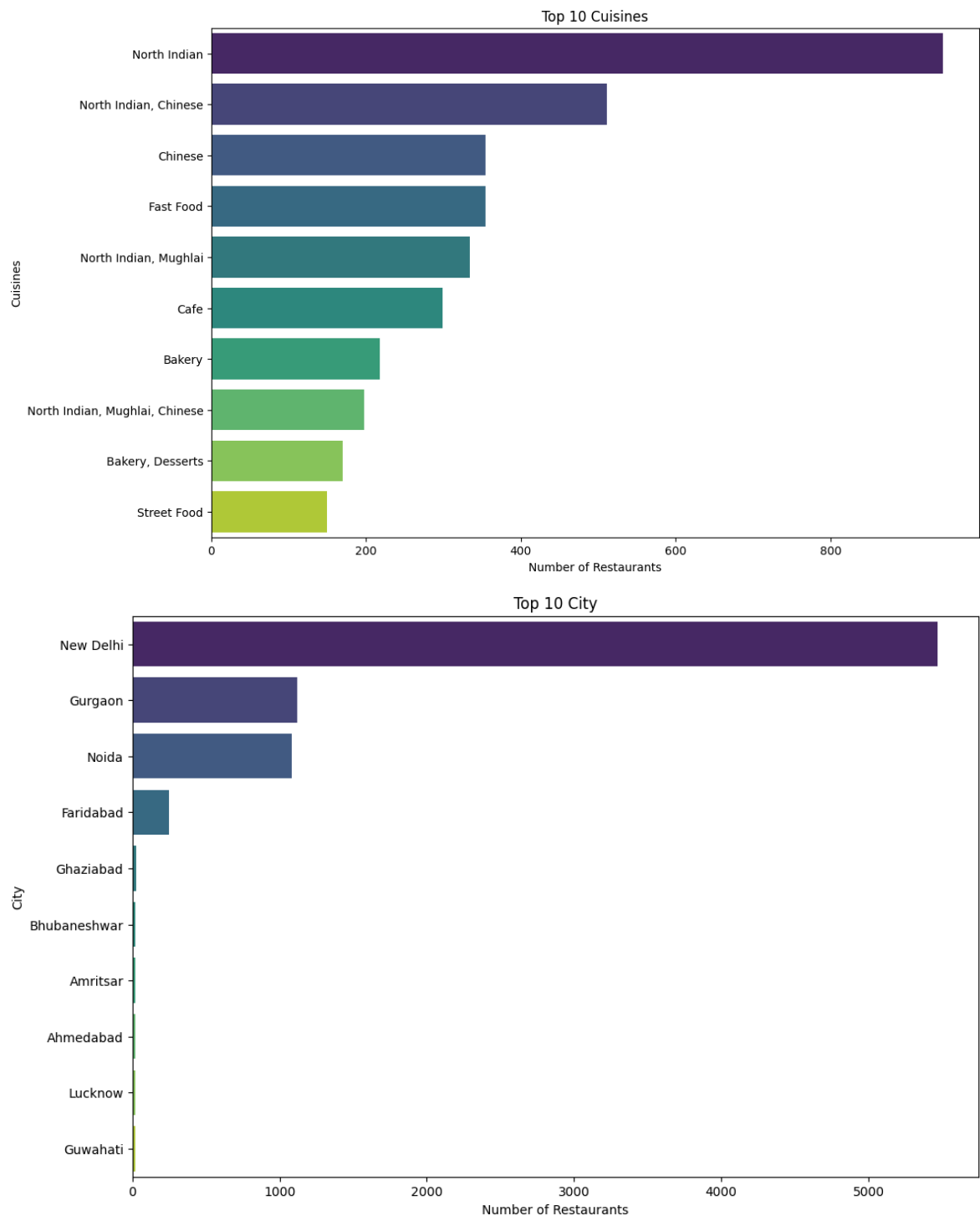
```
In [ ]: # Explore the distribution of "Cuisines" (limiting to top 20 for better visualiz
plot_categorical_distribution('Cuisines', retail, top_n=20)
```



```
In [ ]: # Function to plot top N categories
def plot_top_n_categories(column_name, dataset, n=10):
    top_categories = dataset[column_name].value_counts().nlargest(n)
    plt.figure(figsize=(12, 8))
    sns.barplot(x=top_categories.values, y=top_categories.index, palette='viridis')
    plt.title(f'Top {n} {column_name}')
    plt.xlabel('Number of Restaurants')
    plt.ylabel(column_name)
    plt.show()

# Identify and plot the top cuisines
plot_top_n_categories('Cuisines', retail)

# Identify and plot the top cities
plot_top_n_categories('City', retail)
```



In []:

Task 3 : Geospatial Analysis

In []:

```
# Initialize the map centered around a point (e.g., first restaurant location)
map_center = [retail['Latitude'].iloc[0], retail['Longitude'].iloc[0]]
mymap = folium.Map(location=map_center, zoom_start=12)

# Add markers for each restaurant location
for idx, row in retail.iterrows():
```

```

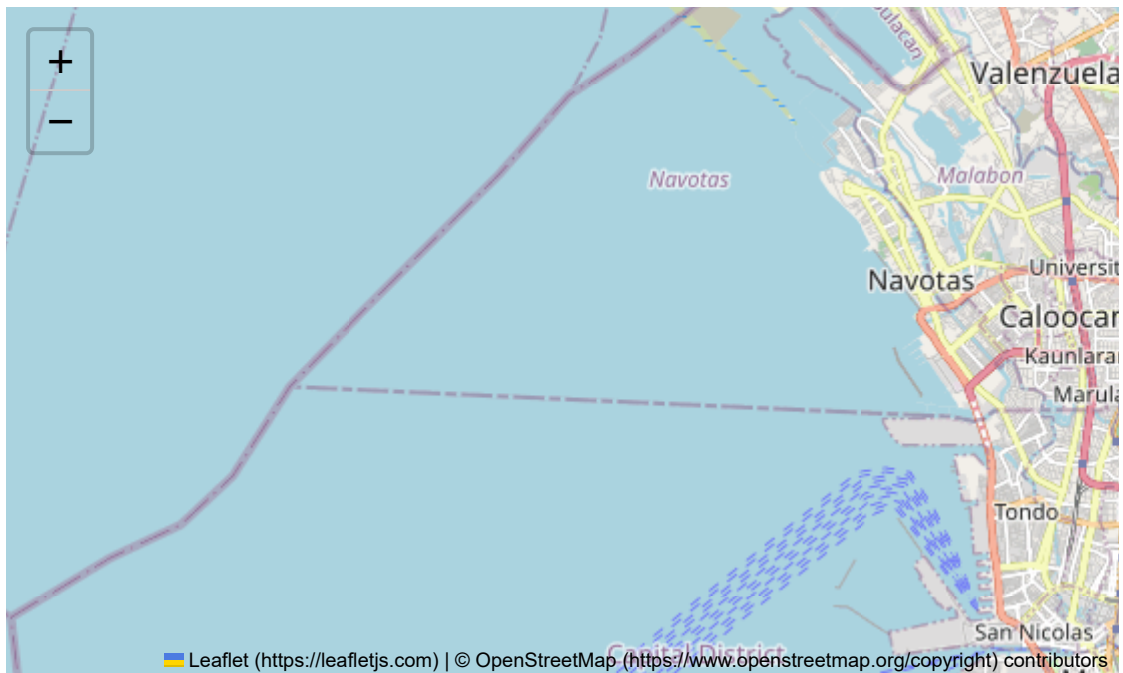
folium.Marker([row['Latitude'], row['Longitude']], popup=row['Restaurant Nam

# Save the map as an HTML file
mymap.save('restaurant_locations_map.html')

# Display the map
mymap

```

Out[]:



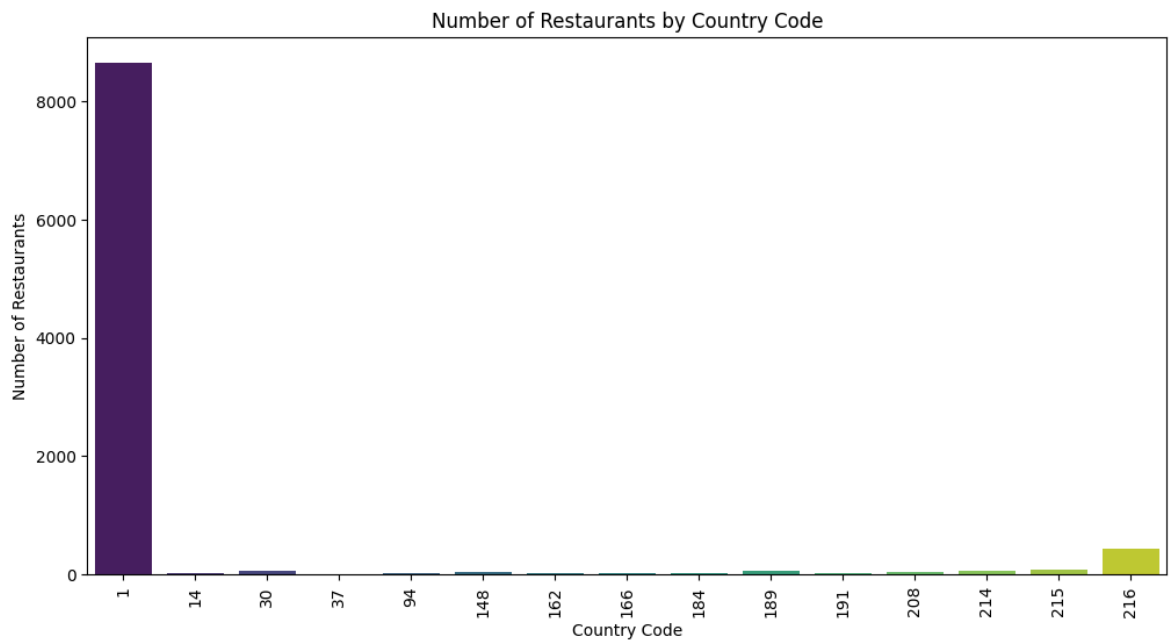
```

In [ ]: # Analyze distribution across cities
plt.figure(figsize=(24, 22))
city_counts = retail['City'].value_counts().sort_values(ascending=False)
sns.barplot(x=city_counts.index, y=city_counts.values, palette='viridis')
plt.title('Number of Restaurants by City')
plt.xlabel('City')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=90)
plt.show()

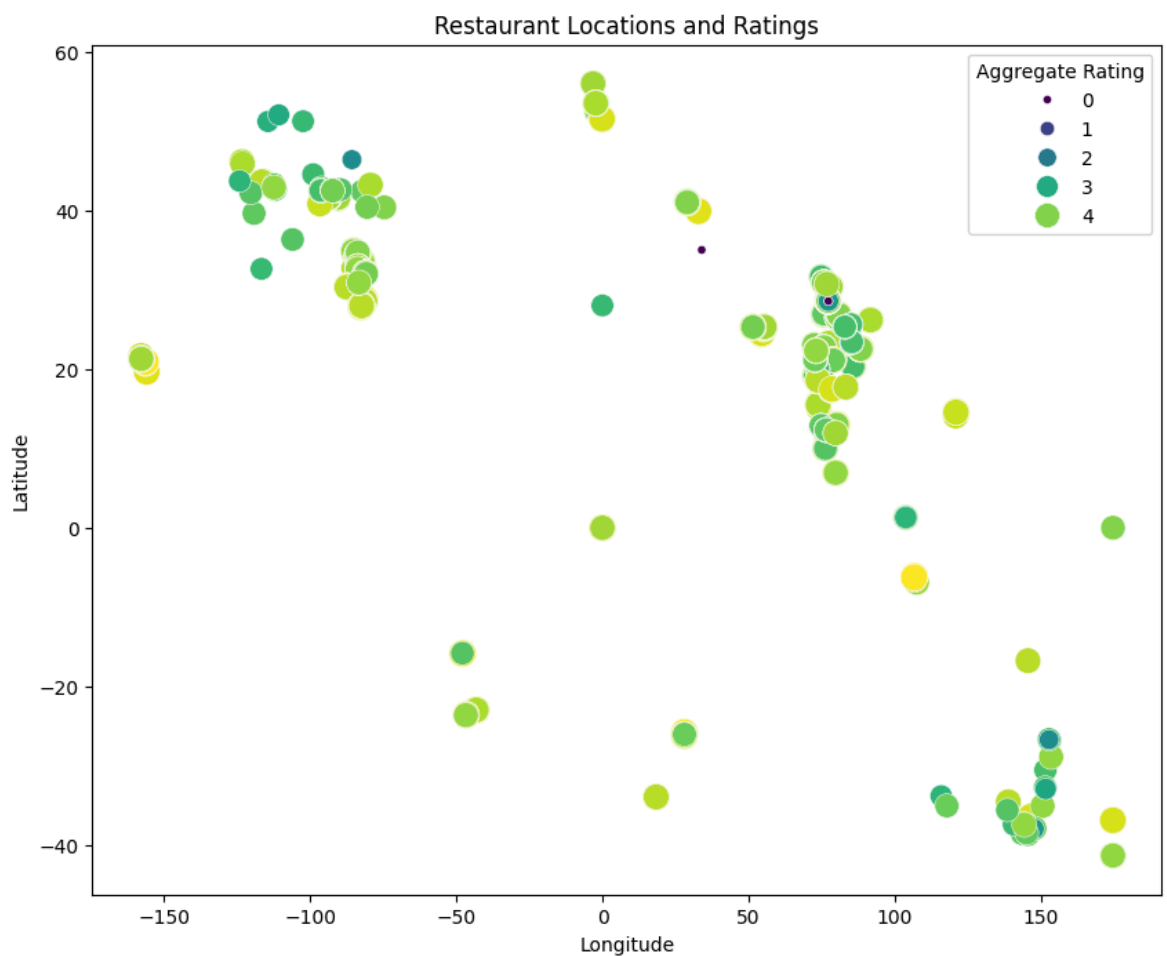
```




```
In [ ]: if 'Country Code' in retail.columns:
        plt.figure(figsize=(12, 6))
        country_counts = retail['Country Code'].value_counts().sort_values(ascending
        sns.barplot(x=country_counts.index, y=country_counts.values, palette='viridi
        plt.title('Number of Restaurants by Country Code')
        plt.xlabel('Country Code')
        plt.ylabel('Number of Restaurants')
        plt.xticks(rotation=90)
        plt.show()
```



```
In [ ]: # Explore correlation between Location (Latitude and Longitude) and rating
plt.figure(figsize=(10, 8))
sns.scatterplot(x='Longitude', y='Latitude', hue='Aggregate rating', data=retail)
plt.title('Restaurant Locations and Ratings')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend(title='Aggregate Rating')
plt.show()
```



```
In [ ]: # Calculate correlation coefficient between location and rating
correlation = retail[['Latitude', 'Longitude', 'Aggregate rating']].corr()
print("Correlation Matrix:")
print(correlation)
```

Correlation Matrix:

	Latitude	Longitude	Aggregate rating
Latitude	1.000000	0.043207	0.000516
Longitude	0.043207	1.000000	-0.116818
Aggregate rating	0.000516	-0.116818	1.000000

The analysis shows that the geographical location (latitude and longitude) of restaurants does not have a significant linear impact on their aggregate ratings