

Task 4: Location-Based Restaurant Analysis

Objective

Perform a geographical and statistical analysis of restaurant locations, cuisines, and ratings using city/locality-wise groupings.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import folium
from folium.plugins import MarkerCluster

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Load Dataset from Google Drive
file_path = '/content/drive/My Drive/ML_Internship/resturant_dataset.csv'
df = pd.read_csv(file_path)

# Preview first few rows
df.head()

# Basic view
df[['City', 'Latitude', 'Longitude', 'Aggregate rating', 'Cuisines']].head()
```

Mounted at /content/drive

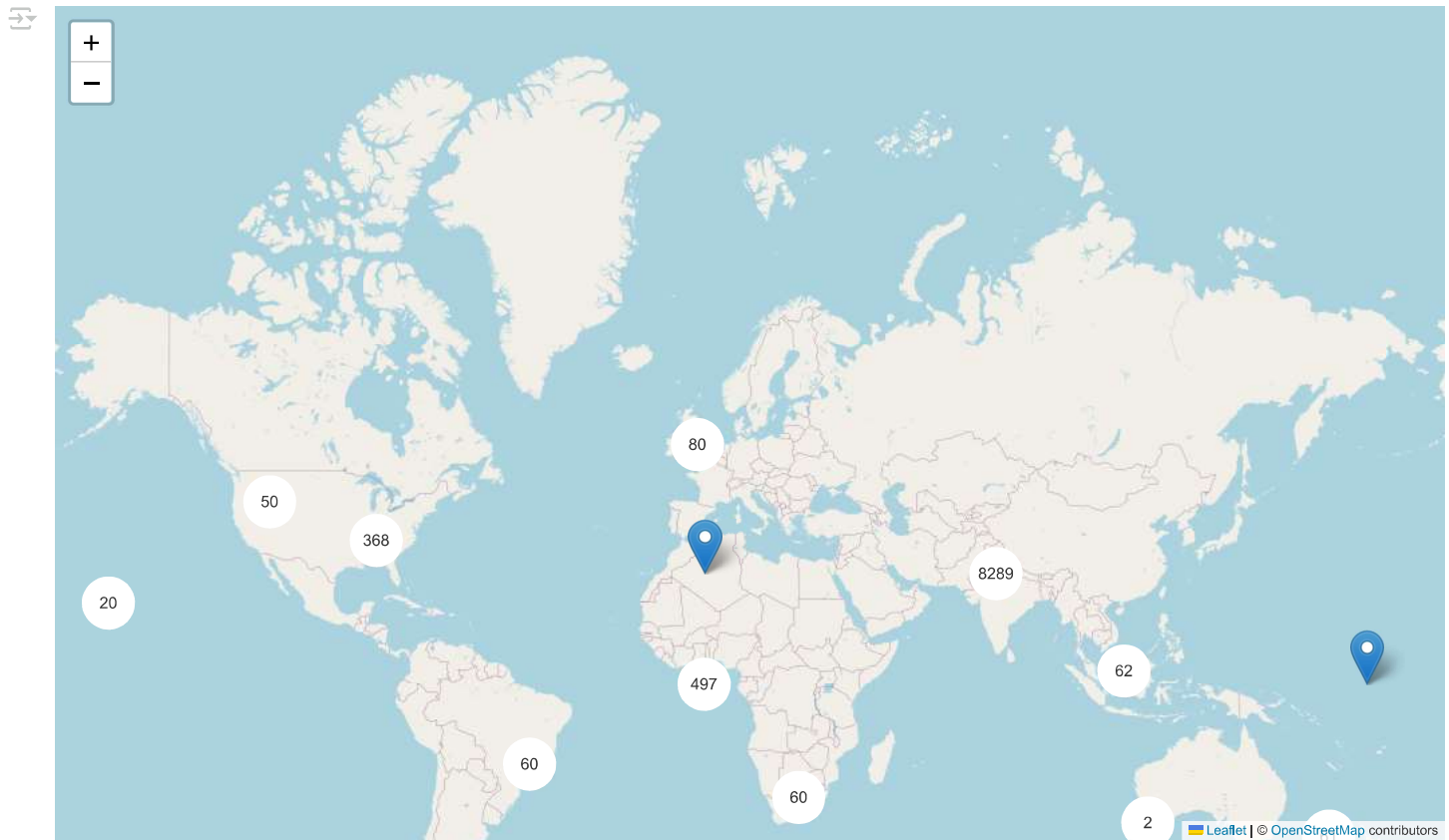
	City	Latitude	Longitude	Aggregate rating	Cuisines
0	Makati City	14.565443	121.027535	4.8	French, Japanese, Desserts
1	Makati City	14.553708	121.014101	4.5	Japanese
2	Mandaluyong City	14.581404	121.056831	4.4	Seafood, Asian, Filipino, Indian
3	Mandaluyong City	14.585318	121.056475	4.9	Japanese, Sushi
4	Mandaluyong City	14.584450	121.057508	4.8	Japanese, Korean

Step 1: Restaurant Location Mapping (Using Folium)

```
# Center at median coordinates
map_center = [df['Latitude'].median(), df['Longitude'].median()]
restaurant_map = folium.Map(location=map_center, zoom_start=2)
marker_cluster = MarkerCluster().add_to(restaurant_map)

for idx, row in df.iterrows():
    folium.Marker(location=[row['Latitude'], row['Longitude']],
                  popup=row['City']).add_to(marker_cluster)

restaurant_map
```



▼ Step 2: City-wise Analysis of Restaurant Count and Average Rating

```
# Top cities by number of restaurants
top_cities = df['City'].value_counts().nlargest(10)
plt.figure(figsize=(10,5))
sns.barplot(x=top_cities.values, y=top_cities.index)
plt.title('Top 10 Cities by Number of Restaurants')
plt.xlabel('Number of Restaurants')
plt.ylabel('City')
plt.show()
```



Top 10 Cities by Number of Restaurants



```
# Average rating by city
avg_ratings = df.groupby('City')['Aggregate rating'].mean().sort_values(ascending=False).head(10)
plt.figure(figsize=(10,5))
sns.barplot(x=avg_ratings.values, y=avg_ratings.index)
plt.title('Top 10 Cities by Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('City')
plt.show()
```



Top 10 Cities by Average Rating

