

✓ Geospatial Analysis

Project Overview:

This project analyzes restaurant data using location (latitude & longitude), city, country, and rating. It visualizes restaurant distributions, explores correlations with ratings, and identifies hotspots for deeper insights.

Author: Abhijeet Sah

GitHub: <https://github.com/Abhijeet-sah>

Table of Contents

1. [Load Dataset](#)
2. [Visualize Restaurant Locations on Map](#)
3. [Heatmap of Restaurant Density](#)
4. [Distribution of Restaurants by City & Country](#)
5. [Correlation Between Location & Rating](#)
6. [Extra Insights]
 - [Top-Rated Restaurants per City](#)
 - [Rating Distribution per Country](#)
 - [Hotspot Detection with Clustering](#)

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium.plugins import HeatMap
```

Load Dataset

```
df = pd.read_csv("/content/Dataset .csv")
print("Dataset shape:", df.shape)
df.head()
```

Dataset shape: (9551, 21)

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose
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...
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, Ortigas, Mandaluyong City, Ma...
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...
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...

5 rows × 21 columns

Warning: Total number of columns (21) exceeds max_columns (20) limiting to first (20)

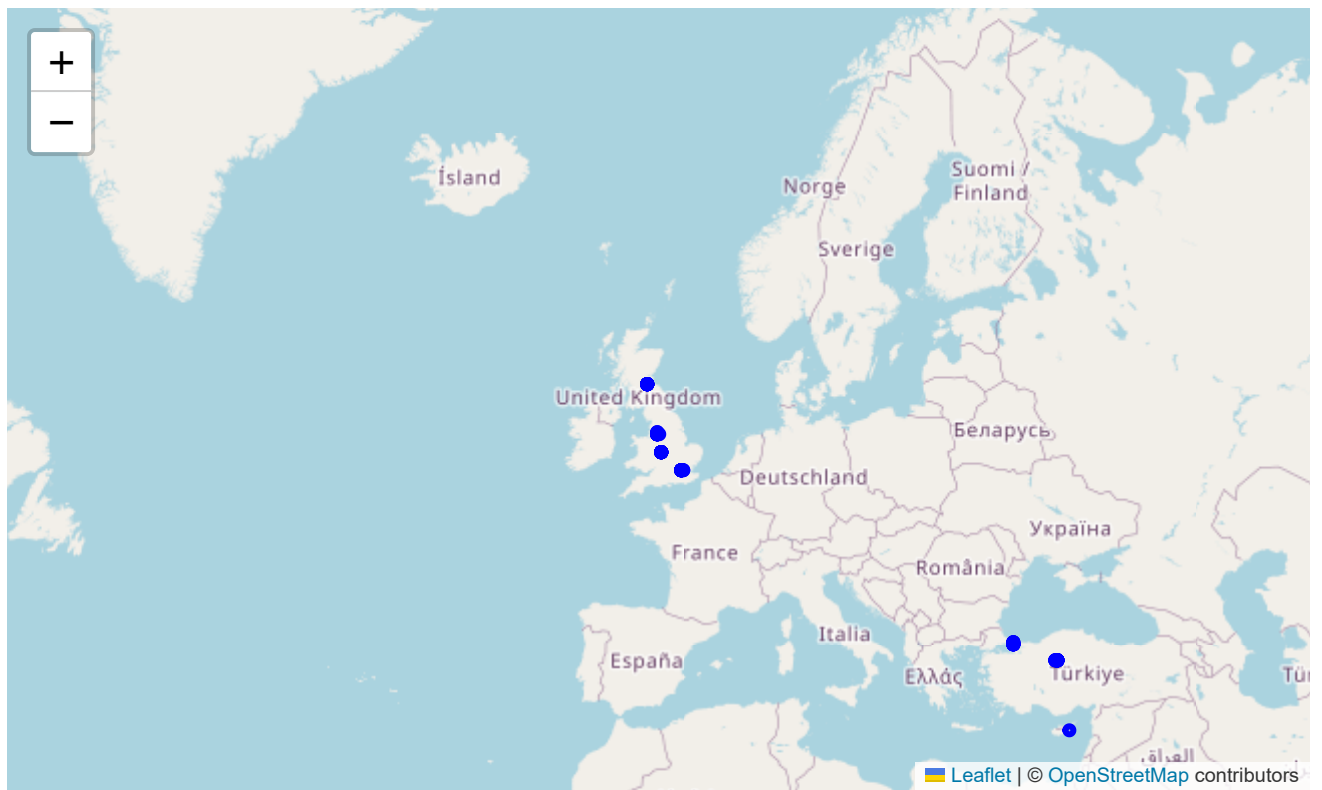
Visualize Restaurant Locations on Map

```
# Center map around average coordinates
map_center = [df['Latitude'].mean(), df['Longitude'].mean()]
restaurant_map = folium.Map(location=map_center, zoom_start=3)

# Add markers
for _, row in df.iterrows():
```

```
if not np.isnan(row['Latitude']) and not np.isnan(row['Longitude']):
    folium.CircleMarker(
        location=[row['Latitude'], row['Longitude']],
        radius=2,
        popup=f"{row['City']} | Rating: {row['Aggregate rating']}",
        color="blue",
        fill=True,
        fill_color="blue"
    ).add_to(restaurant_map)
```

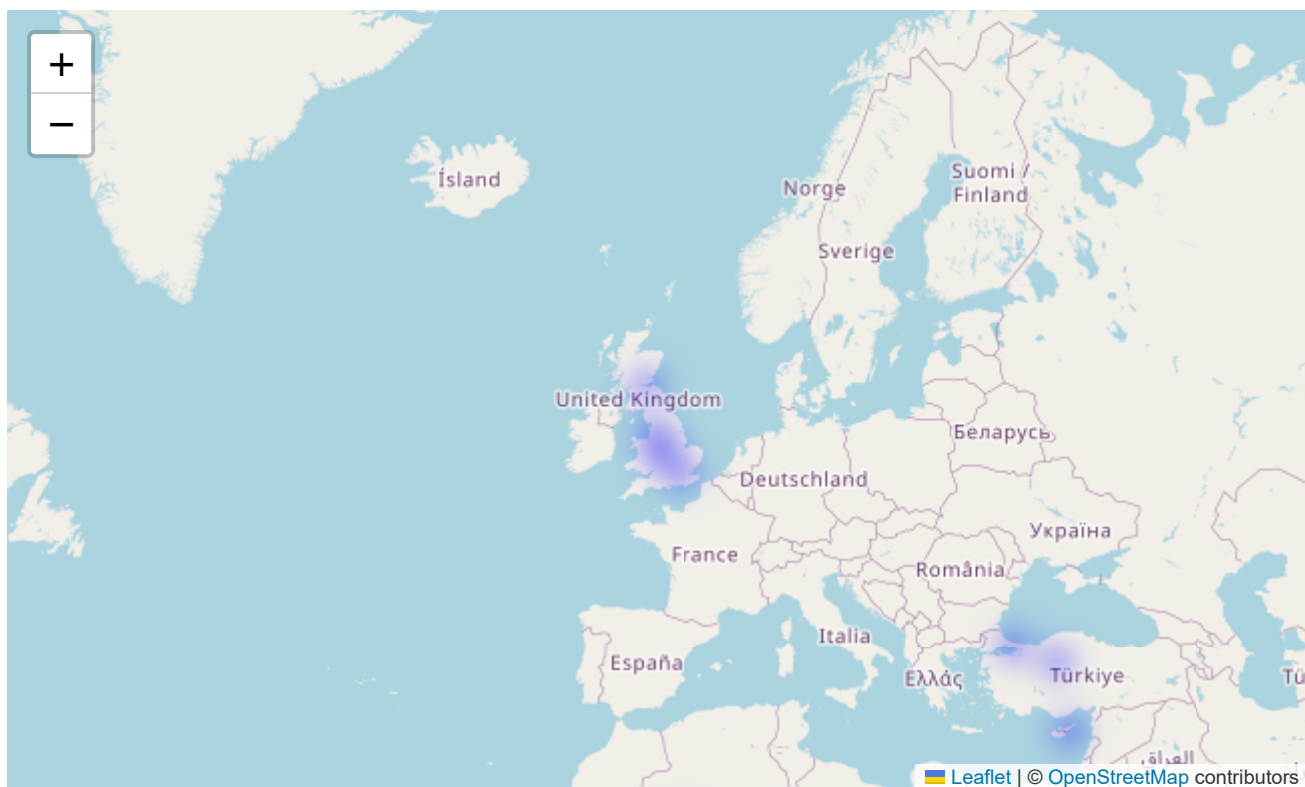
restaurant_map



Heatmap of Restaurant Density

```
heatmap_map = folium.Map(location=map_center, zoom_start=3)
heat_data = df[['Latitude', 'Longitude']].dropna().values.tolist()

HeatMap(heat_data, radius=8).add_to(heatmap_map)
heatmap_map
```

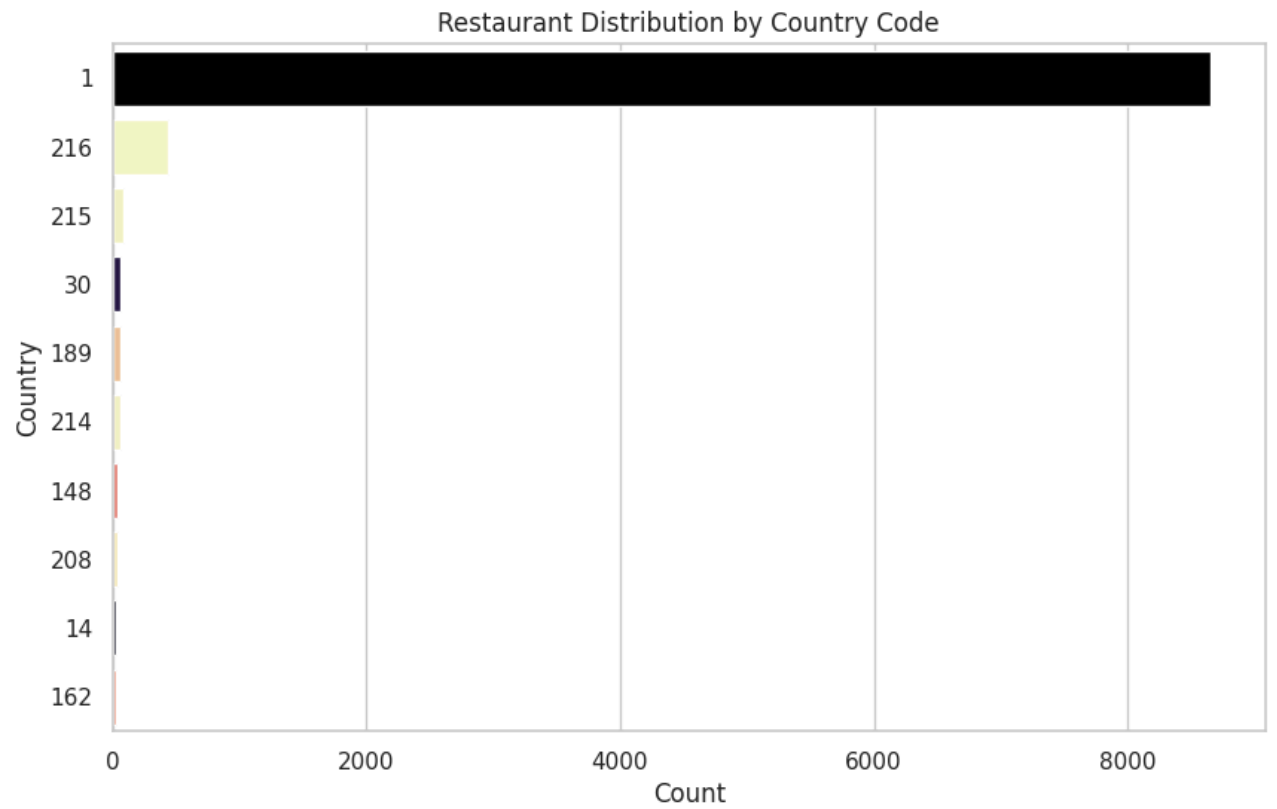
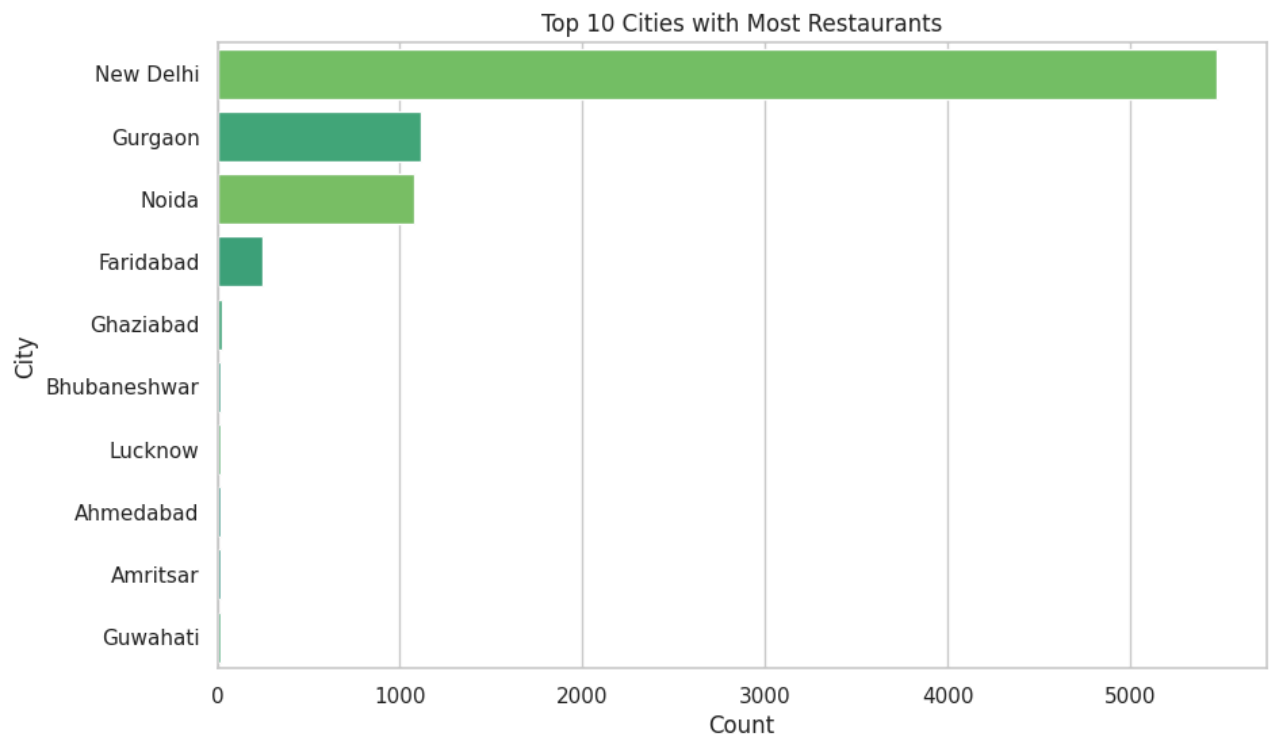


Distribution of Restaurants by City & Country

```
# Top 10 Cities
plt.figure(figsize=(10,6))
sns.countplot(
    data=df,
    y='City',
```

```
order=df['City'].value_counts().head(10).index,
hue='City',
dodge=False,
legend=False,
palette="viridis"
)
plt.title("Top 10 Cities with Most Restaurants")
plt.xlabel("Count")
plt.ylabel("City")
plt.show()

# Country Distribution
plt.figure(figsize=(10,6))
sns.countplot(
    data=df,
    y='Country Code',
    order=df['Country Code'].value_counts().head(10).index,
    hue='Country Code',
    dodge=False,
    legend=False,
    palette="magma"
)
plt.title("Restaurant Distribution by Country Code")
plt.xlabel("Count")
plt.ylabel("Country")
plt.show()
```

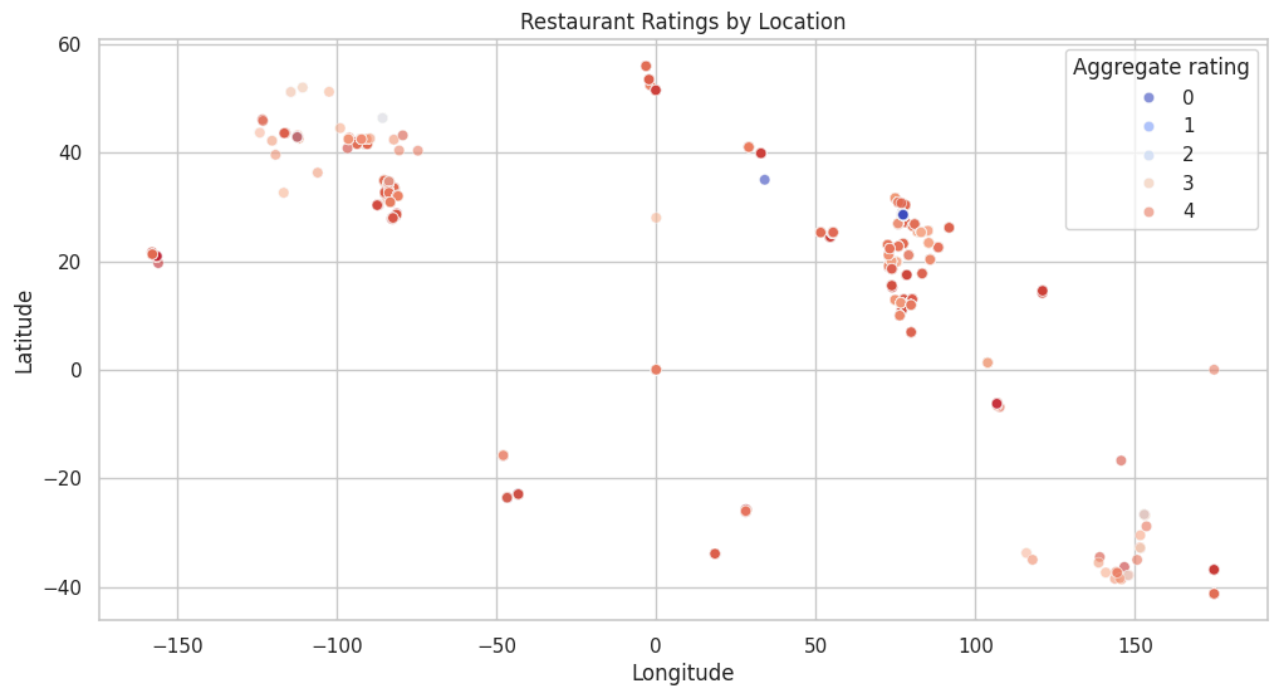


Correlation Between Location & Rating

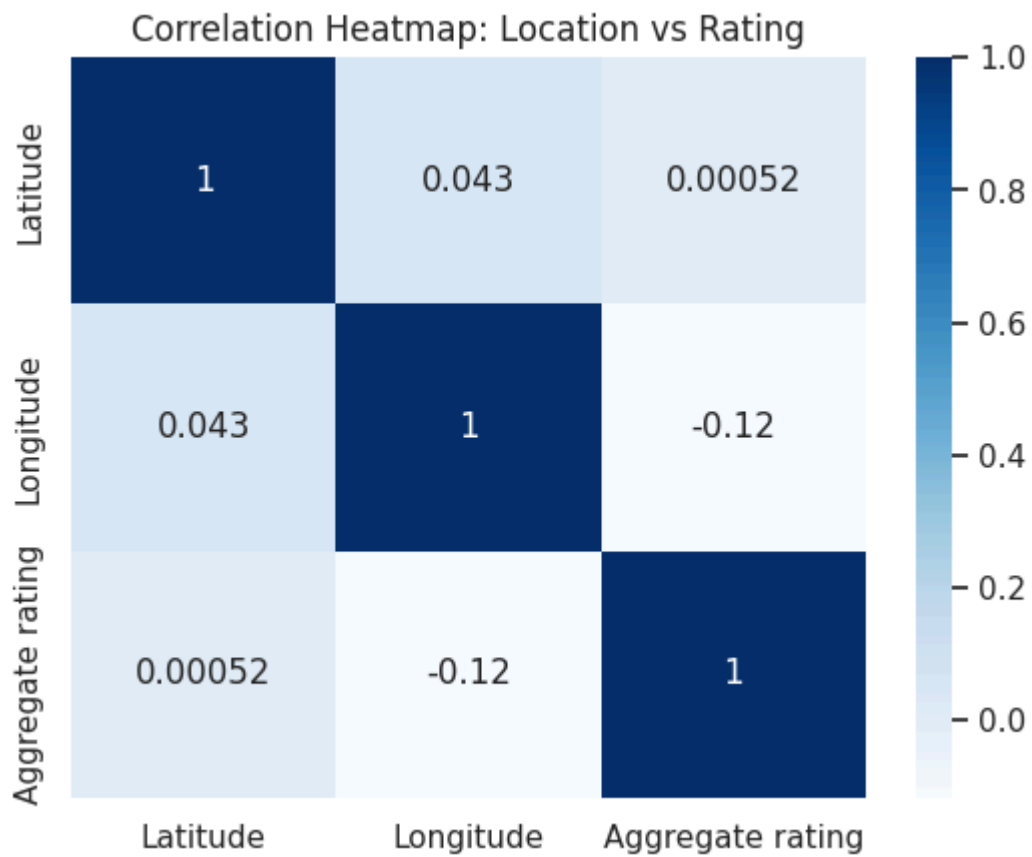
```
# Scatterplot of ratings by location
plt.figure(figsize=(12,6))
sns.scatterplot(
    data=df,
    x="Longitude", y="Latitude",
    hue="Aggregate rating",
    palette="coolwarm",
    alpha=0.6
)
plt.title("Restaurant Ratings by Location")
plt.show()

# Correlation heatmap
corr_data = df[['Latitude', 'Longitude', 'Aggregate rating']].corr()
print(corr_data)

sns.heatmap(corr_data, annot=True, cmap="Blues")
plt.title("Correlation Heatmap: Location vs Rating")
plt.show()
```

	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



Top-Rated Restaurants per City

```
# Drop rows with missing values in essential columns
df_clean = df.dropna(subset=['City', 'Aggregate rating', 'Restaurant Name'])

# Find the index of the top-rated restaurant in each city
top_indices = df_clean.groupby('City')['Aggregate rating'].idxmax()

# Select the top-rated restaurants
top_per_city = df_clean.loc[top_indices, ['City', 'Restaurant Name', 'Aggregate rating']]

# Display top 10 cities by restaurant rating
top_per_city_sorted = top_per_city.sort_values('Aggregate rating', ascending=False).head(
top_per_city_sorted
# Plot top 10 by rating
plt.figure(figsize=(10,5))
sns.barplot(
    x='Aggregate rating',
    y='City',
    data=top_per_city_sorted,
    hue='City', # Added hue
    legend=False, # Added legend=False
    palette="crest"
)
plt.title("Top-Rated Restaurants by City")
plt.xlabel("Rating")
plt.ylabel("City")
plt.show()
```