

▼ Capstone Project – The Battle of Neighbourhoods

Introduction

Colombo is the commercial capital and largest city of Sri Lanka by population. According to the Brookings Institution, Colombo metropolitan area has a population of 5.6 million and 752,993 in the city proper. It is the financial centre of the island and a tourist destination. It is located on the west coast of the island and adjacent to the Greater Colombo area which includes Sri Jayawardenepura Kotte, the legislative capital of Sri Lanka and Dehiwala-Mount Lavinia. Colombo is often referred to as the capital since Sri Jayawardenepura Kotte is itself within the urban/suburban area of Colombo. It is also the administrative capital of the Western Province and the district capital of Colombo District. Colombo is a busy and vibrant city with a mixture of modern life, colonial buildings and monuments.

With its diverse culture, comes diverse food items. There are many restaurants in Colombo City, each belonging to different categories like Chinese, Italian, French etc.

This project will find out the best Chinese restaurants in Colombo and analyze the entire Colombo city to find out best venues.

Questions that can be asked using the above mentioned datasets

- What is best location in Colombo City for Chinese Cuisines ?
- Which areas have large number of Chinese Restaurants ?
- Which all areas have less number of restaurants ?
- Which is the best place to stay if you prefer Chinese Cuisines ?
- What places are have best restaurants in Colombo?

Data

For this project we need the following data :

- Colombo Restaurants data that contains list Locality, Restaurant name, Rating along with their latitude and longitude.
 - Data source : [Zomato kaggle dataset](#)
 - Description : This data set contains the required information. And we will use this data set to explore various locality of Colombo city.
- Nearby places in each locality of Colombo city.
 - Data source : [Foursquare API](#)
 - Description : By using this api we will get all the venues in each neighborhood.

▼ Approach

- Collect the Colombo city data from [Zomato kaggel dataset](#)
- Using FourSquare API we will find all venues for each neighborhood.
- Filter out all venues that are nearby by locality.
- Using aggregative rating for each restaurant to find the best places.
- Visualize the Ranking of neighborhoods using folium library(python)

```
import pandas as pd
import numpy as np
import requests
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
import folium
import geocoder
```

▼ Read the zomato restaurant data from csv file

```
# Read cvs file
df = pd.read_csv('https://raw.githubusercontent.com/salitha10/Coursera_Capstone/main/zomat
df.head()
#df2 = df[df['City'].str.contains("Colombo")]
#df2
```

	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 Avenue...	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...

```
# Get all restaurants in colombo
df_SL = df[df['Country Code'] == 191]
df_res = df_SL[df_SL['City'] == 'Colombo']
df_res.reset_index(drop=True, inplace=True)
df_res.head()
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose
0	5800557	Chinese Dragon Cafe	191	Colombo	11, Milagiriya Avenue, Bambalapitiya, Colombo 04	Bambalapitiya, Colombo 04	Bambalapitiya Colombo 04 Colombo
1	5800634	Elite Indian Restaurant	191	Colombo	124, New Bullers Road, Bambalapitiya, Colombo 04	Bambalapitiya, Colombo 04	Bambalapitiya Colombo 04 Colombo
2	5800567	CIOCONAT Lounge	191	Colombo	107, Barnes Place, Cinnamon Gardens, Colombo 07	Cinnamon Gardens, Colombo 07	Cinnamon Gardens Colombo 07 Colombo
3	5800891	The Paddington	191	Colombo	36, Barnes Place, Cinnamon Gardens, Colombo 07	Cinnamon Gardens, Colombo 07	Cinnamon Gardens Colombo 07 Colombo
4	5800590	The Commons	191	Colombo	39 A, Flower Road, Cinnamon Gardens, Colombo 07	Cinnamon Gardens, Colombo 07	Cinnamon Gardens Colombo 07 Colombo

▼ Data Cleaning

Remove unnecessary columns and rows from the dataset

```
df_all= df_res[df_res.Longitude !=0.000000][['Restaurant Name','Locality','Longitude','Lat
```

```
df_all = df_all[df_all['Aggregate rating'] !=0.0]
df_all.head()
```

	Restaurant Name	Locality	Longitude	Latitude	Cuisines	Aggregate rating	Rating text	Vot
0	Chinese Dragon Cafe	Bambalapitiya, Colombo 04	79.856678	6.886341	Chinese	3.4	Average	1
1	Elite Indian Restaurant	Bambalapitiya, Colombo 04	79.857830	6.896084	North Indian, Chinese, Sri Lankan	2.4	Poor	2

▼ Map showing restaurants

```
# Get map
restaurants = folium.Map(location=[6.9271, 79.8612], zoom_start=10)

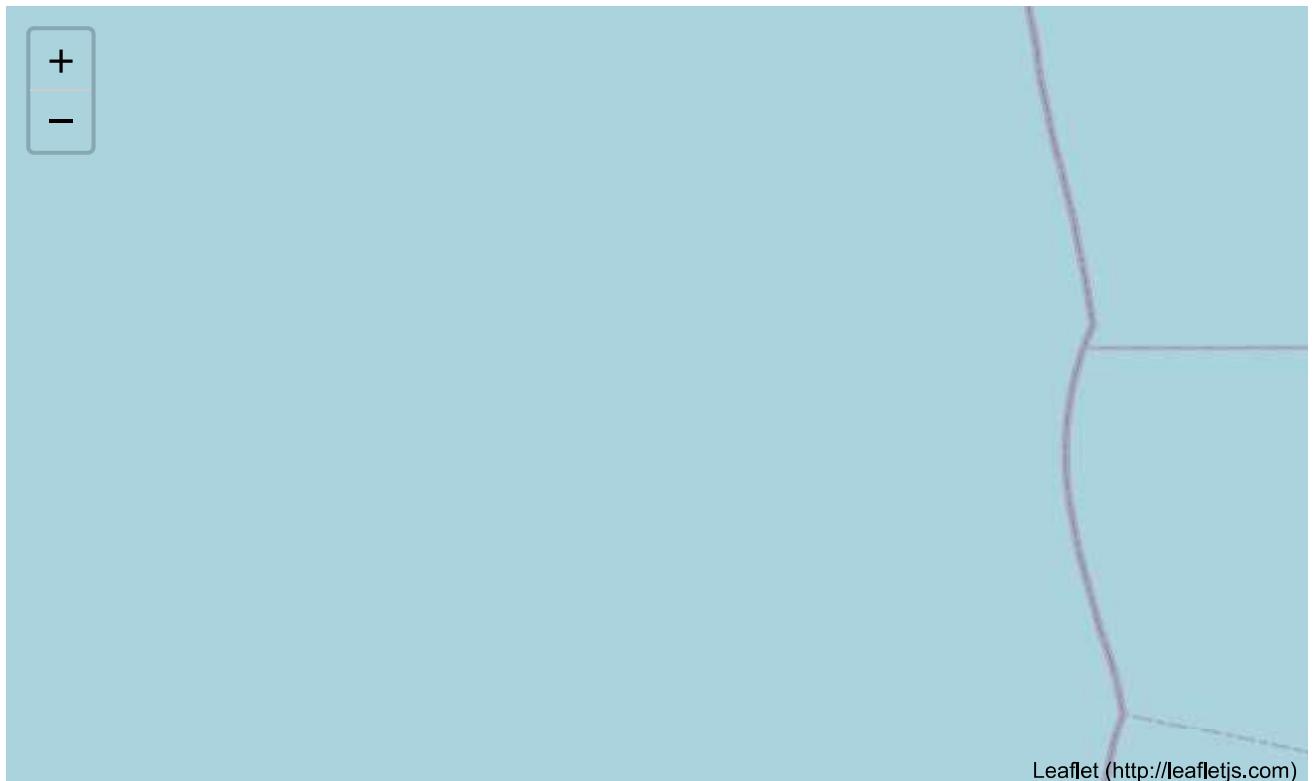
X = df_all['Latitude']
Y = df_all['Longitude']
Z = np.stack((X, Y), axis=1)

# clusture
kmeans = KMeans(n_clusters=5, random_state=0).fit(Z)

clusters = kmeans.labels_
colors = ['red', 'green', 'blue', 'yellow', 'orange']
df_all ['Cluster'] = clusters

for latitude, longitude, Locality, cluster in zip(df_all['Latitude'], df_all['Longitude'],
label = folium.Popup(Locality, parse_html=True)
folium.CircleMarker(
    [latitude, longitude],
    radius=5,
    popup=label,
    color='black',
    fill=True,
    fill_color=colors[cluster],
    fill_opacity=0.7).add_to(restaurants)

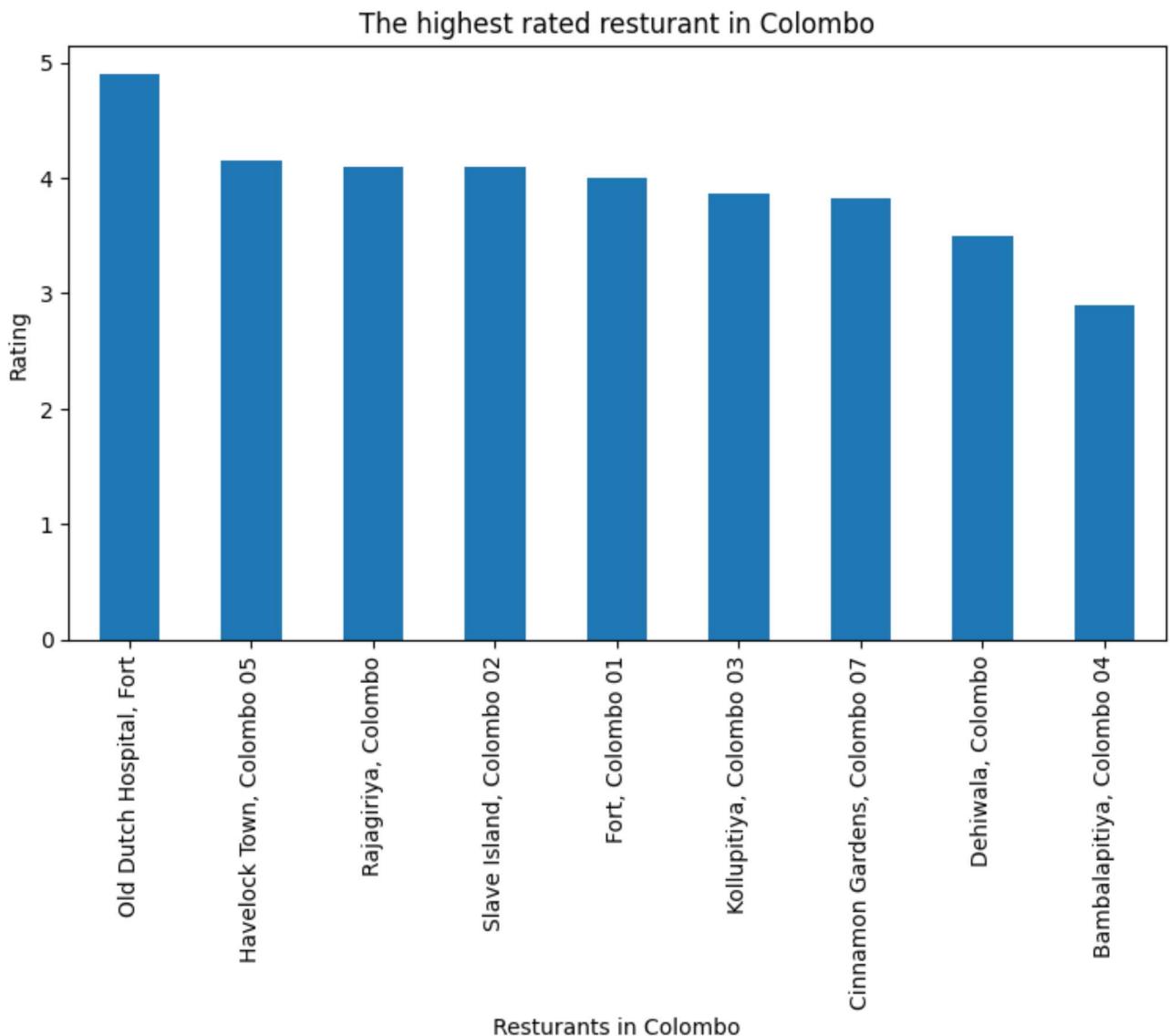
restaurants
```



▼ Locations with best restaurants in Colombo?

```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The highest rated restaurant in Colombo')
```

```
# display bar chart
df_all.groupby('Locality')['Aggregate rating'].mean().nlargest(10).plot(kind='bar')
plt.xlabel('Resturants in Colombo')
plt.ylabel('Rating')
plt.show()
```



df_all

	Restaurant Name	Locality	Longitude	Latitude	Cuisines	Aggregate rating	Rating text	V
0	Chinese Dragon Cafe	Bambalapitiya, Colombo 04	79.856678	6.886341	Chinese	3.4	Average	-
1	Elite Indian Restaurant	Bambalapitiya, Colombo 04	79.857830	6.896084	North Indian, Chinese, Sri Lankan	2.4	Poor	-
2	CIOCONAT Lounge	Cinnamon Gardens, Colombo 07	79.875114	6.912806	Italian, Cafe, Desserts	3.7	Good	-
3	The Paddington	Cinnamon Gardens, Colombo 07	79.868492	6.913292	Cafe, Italian	3.6	Good	-
4	The Commons	Cinnamon Gardens, Colombo 07	79.858105	6.908536	Cafe, Sri Lankan, Continental, American	4.0	Very Good	-
5	Upali's	Cinnamon Gardens, -	79.864727	6.912529	Sri Lankan	4.0	Very Good	-

The best restaurants are available in **Old Dutch Hospital, Fort**

Malav Dehiwala North

▼ Locations with worst restaurants in Colombo?

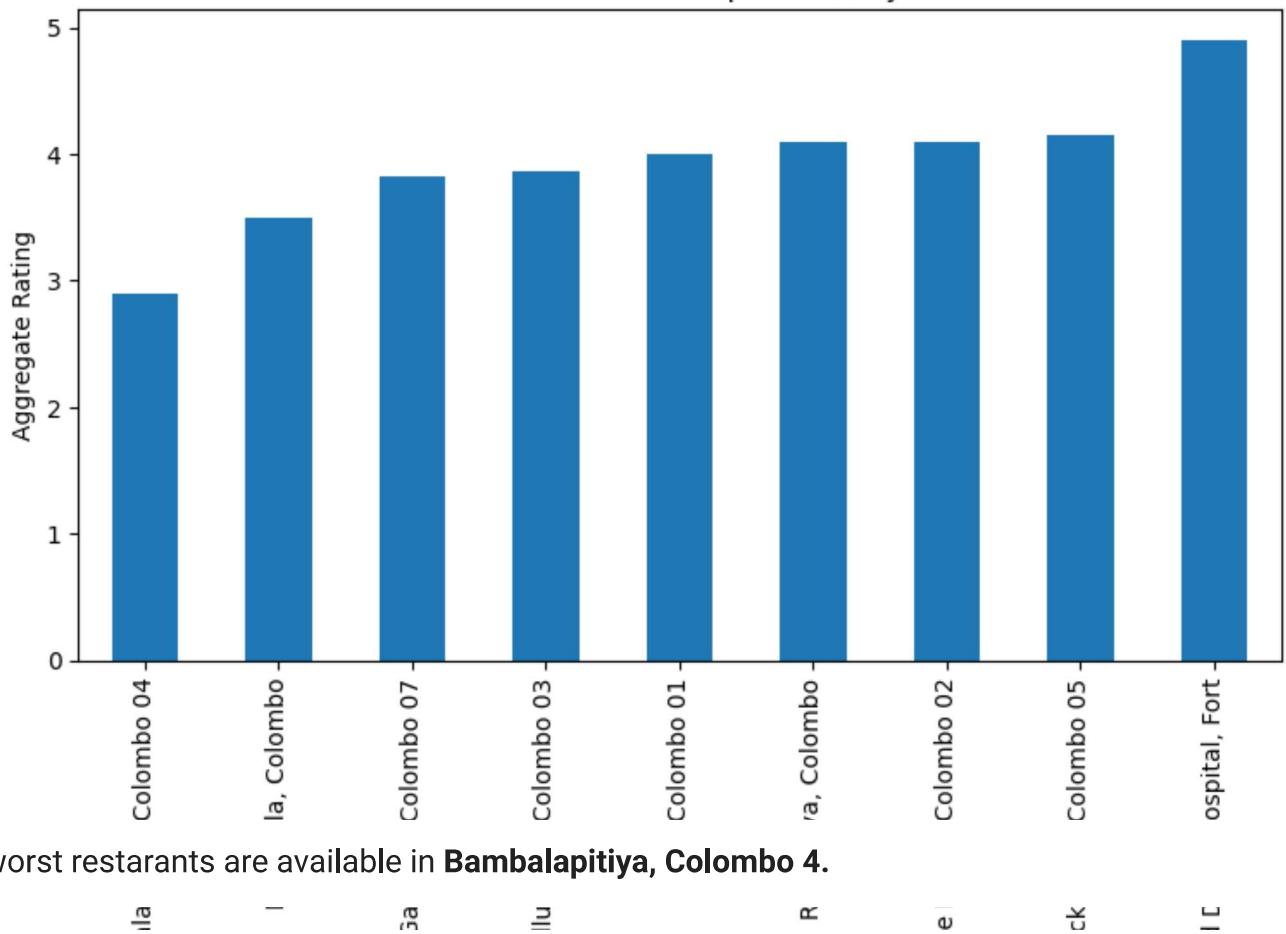
7 - - - - - 79.843575 6.932547 American, - - - - - 4.0 Very - - - - -

```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)

plt.title('The Worst rated restaurant in top 10 locality of Colombo')
df_all.groupby('Locality')['Aggregate rating'].mean().nlargest(10).plot(kind='bar')

plt.xlabel('Restaurant Locality in Colombo')
plt.ylabel('Aggregate Rating')
plt.show()
```

The Worst rated restaurant in top 10 locality of Colombo



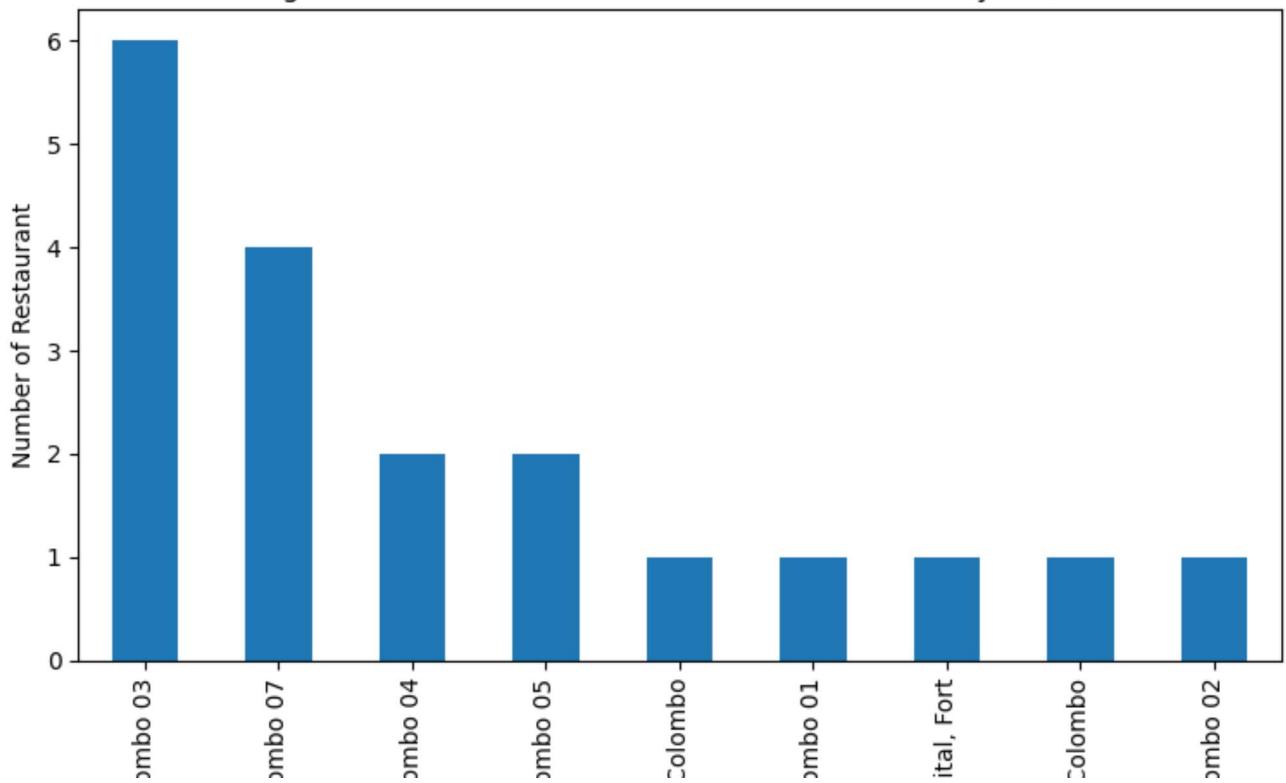
The worst restaurants are available in **Bambalapitiya, Colombo 4.**

▼ Best places to dine in Colombo

```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The highest number of Restaurants available in Locality of Colombo')

df_all.groupby('Locality')['Restaurant Name'].count().nlargest(10).plot(kind='bar')
plt.xlabel('Restaurant Locality in Colombo')
plt.ylabel('Number of Restaurant')
plt.show()
```

The highest number of Restaurants available in Locality of Colombo



Kollupitiya, Colombo 3 is the best place to dine in Colombo.

- ▼ Which locations are not suitable for dine in Colombo?

```

import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The lowest number of Restaurant available in Locality of Colombo')
#On x-axis

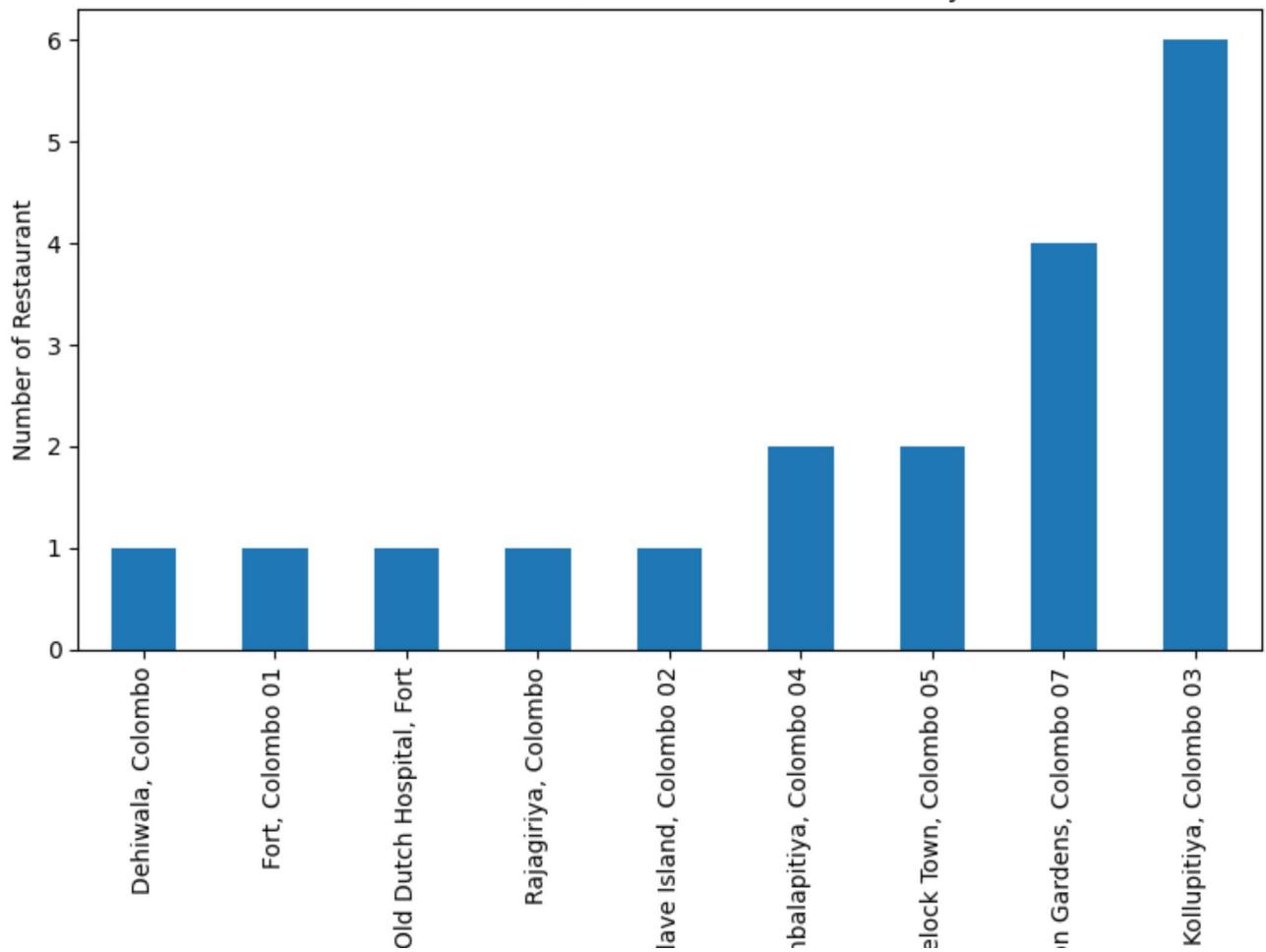
#giving a bar plot
df_all.groupby('Locality')['Restaurant Name'].count().nsmallest(10).plot(kind='bar')

plt.xlabel('Restaurant Locality in Colombo')
#On y-axis
plt.ylabel('Number of Restaurant')

#displays the plot
plt.show()

```

The lowest number of Restaurant available in Locality of Colombo



- ▼ The best places for chinese restaurants in Colombo
-

```

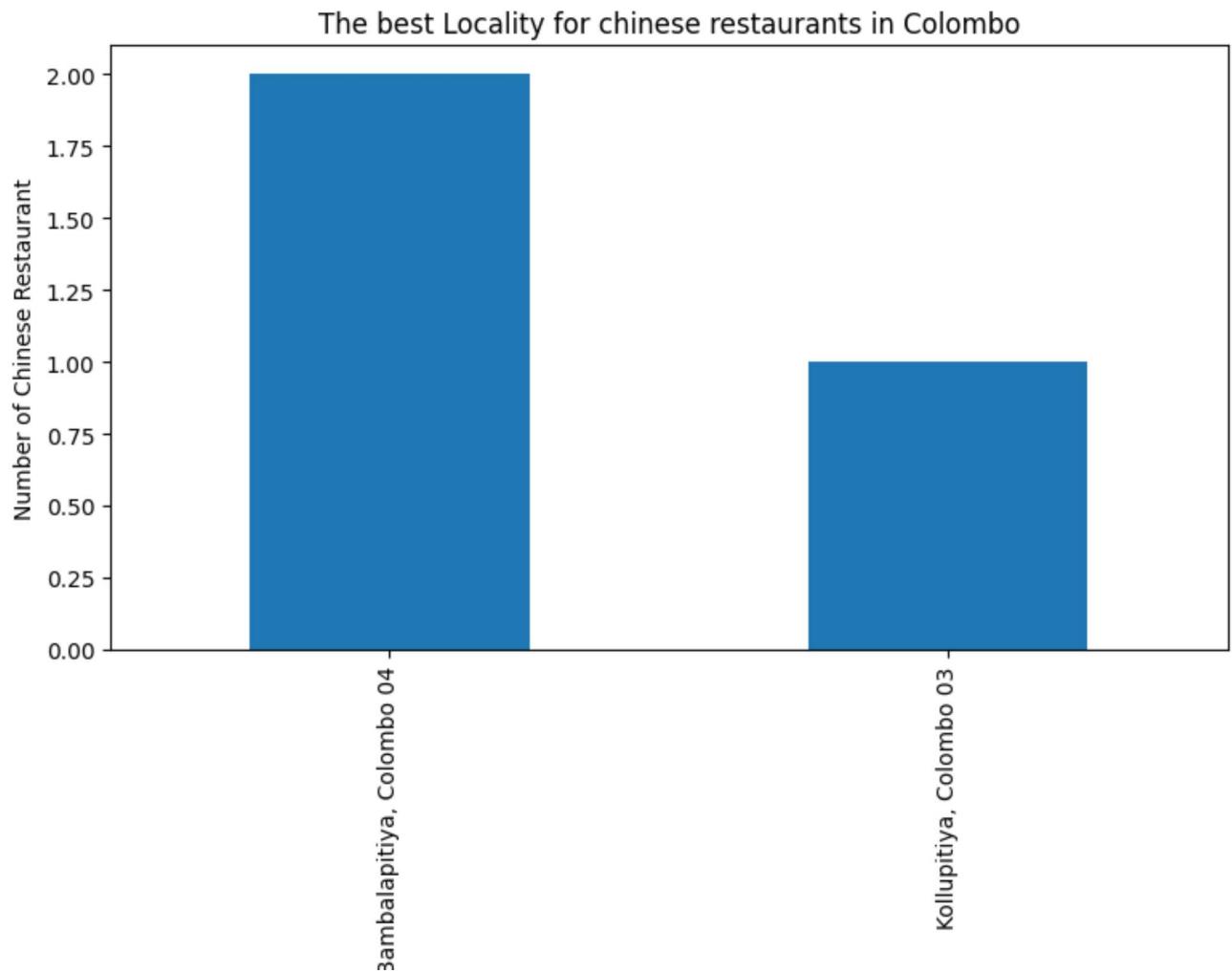
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The best Locality for chinese restaurants in Colombo')
#On x-axis

#giving a bar plot
df_all[df_all['Cuisines'].str.contains('Chinese')].groupby('Locality')['Restaurant Name'].

plt.xlabel('Resturant Locality in Colombo')
#On y-axis
plt.ylabel('Number of Chinese Restaurant')

#displays the plot
plt.show()

```



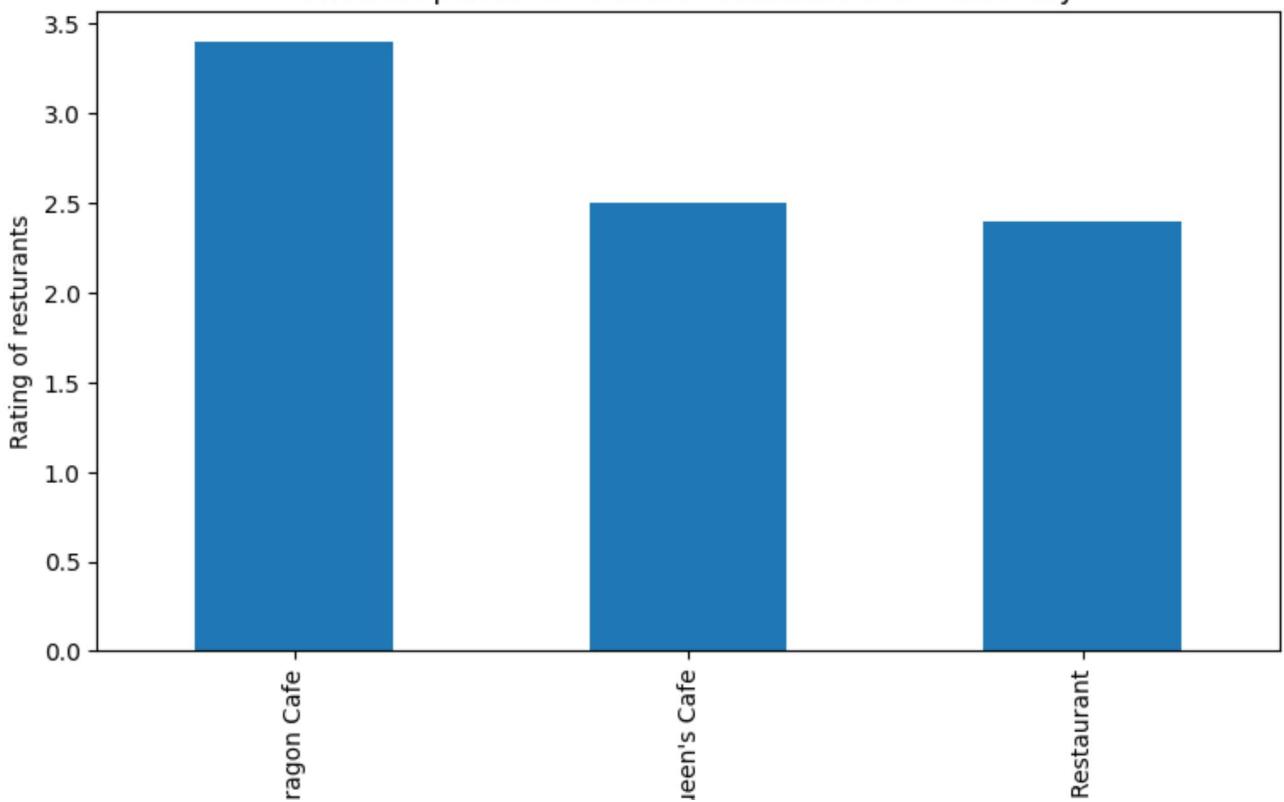
Bambalapitiya is the best place for Chinese restaurants.

▼ Best chinese restaurents in Colombo

```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The best places for Chinese restaurant in Colombo city')

df_all[df_all['Cuisines'].str.contains('Chinese')].groupby('Restaurant Name')[['Aggregate r
plt.xlabel('Restaurant Locality in Colombo')
plt.ylabel('Rating of resturants')
plt.show()
```

The best places for Chinese restaurant in Colombo city



Chinese Dragon Cafe is the best Chinese restaurant in Colombo.

▼ Data transformation

Based on Locality grouping the data

```
df_all_Loc = df_all.groupby('Locality').count()['Restaurant Name'].to_frame()
df_all_rating= df_all.groupby('Locality')['Aggregate rating'].mean().to_frame()

df_Cuisines = df_all.groupby(['Locality'])['Cuisines'].agg(', '.join).reset_index()
df_R = df_all.groupby(['Locality'])['Rating text'].unique().agg(', '.join).reset_index()
df_V = df_all.groupby(['Locality'])['Votes'].sum().to_frame()
df_Lat = df_all.groupby('Locality').mean()['Latitude'].to_frame()
df_Lng = df_all.groupby('Locality').mean()['Longitude'].to_frame()

# Merge
df_trans = pd.merge(df_Lat,df_Lng, on='Locality').merge(df_all_Loc, on='Locality').merge(df_all_rating, on='Locality')

df_trans = df_trans[df_trans['Aggregate rating'] != 0.000000]
df_trans.columns =[ 'Locality', 'Lat', 'Lng', 'No_of_Restaurant', 'Cusines', 'Agg_Rating', 'Comments']
df_trans.head()
```

Locality	Lat	Lng	No_of_Restaurant	Cusines	Agg_Rating	Comment
				Chinese, Korean		

▼ Foursquare Credentials

```
# Define Foursquare Credentials and Version
CLIENT_ID = 'VUOXS0IJKYVTB5P40Q0DOMHFZBRC1BRDYWUKSQHLUKQAAYB1'
CLIENT_SECRET = 'E10LOME2QLEUVIHRLGJSDLAAJLK2P3DK4PUDQSFARXJDYL5P'
VERSION = '20210701' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)

Your credentails:
CLIENT_ID: VUOXS0IJKYVTB5P40Q0DOMHFZBRC1BRDYWUKSQHLUKQAAYB1
CLIENT_SECRET:E10LOME2QLEUVIHRLGJSDLAAJLK2P3DK4PUDQSFARXJDYL5P
```

▼ Find venues

```
def findVenues(names, latitudes, longitudes, radius=500,LIMIT = 100):

    venues=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}'
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues.append([
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])
```

```

nearby_venues = pd.DataFrame([item for venue_list in venues for item in venue_list])
nearby_venues.columns = ['Locality',
                        'Latitude',
                        'Longitude',
                        'Restaurent',
                        'Restaurent Latitude',
                        'Restaurent Longitude',
                        'Category']

return(nearby_venues)

```

Find the venues in Colombo

```

colombo_ven = findVenues(names=df_trans['Locality'],
                         latitudes=df_trans['Lat'],
                         longitudes=df_trans['Lng']
                         )

```

colombo_ven.head()

Bambalapitiya, Colombo 04
Cinnamon Gardens, Colombo 07
Dehiwala, Colombo
Fort, Colombo 01
Havelock Town, Colombo 05
Kollupitiya, Colombo 03
Old Dutch Hospital, Fort
Rajagiriya, Colombo
Slave Island, Colombo 02

	Locality	Latitude	Longitude	Restaurent	Restaurent Latitude	Restaurent Longitude	Category
0	Bambalapitiya, Colombo 04	6.891212	79.857254	Raja Jewellers	6.889743	79.856922	Jewelry Store
1	Bambalapitiya, Colombo 04	6.891212	79.857254	CASA Colombo	6.888822	79.857499	Asian Restaurant
2	Bambalapitiya, Colombo 04	6.891212	79.857254	Dindigul Thalappakatti	6.888474	79.858556	South Indian Restaurant

colombo_ven.groupby('Locality').count()

Latitude	Longitude	Restaurent	Restaurent	Restaurent	Category
		Latitude	Longitude		

Locality

Analyze Each Locality

```

cmb = pd.get_dummies(colombo_ven[['Category']], prefix="", prefix_sep="")

# add Locality column back to dataframe
cmb['Locality'] = colombo_ven['Locality']
column_list = cmb.columns.tolist()
column_number = int(column_list.index('Locality'))
column_list = [column_list[column_number]] + column_list[:column_number] + column_list[column_number+1:]
cmb = cmb[column_list]

cmb.head()

```

	Locality	American Restaurant	Art Gallery	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auto Garage	Bakery
0	Bambalapitiya, Colombo 04	0	0	0	0	0	0	0
1	Bambalapitiya, Colombo 04	0	0	0	1	0	0	0
2	Bambalapitiya, Colombo 04	0	0	0	0	0	0	0
3	Bambalapitiya, Colombo 04	0	0	0	0	0	0	0
4	Bambalapitiya, Colombo 04	0	0	0	0	0	0	0

5 rows × 87 columns

```

cmb_grouped = cmb.groupby('Locality').mean().reset_index()
cmb_grouped

```

	Locality	American Restaurant	Art Gallery	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auto Garage	Ba
0	Bambalapitiya, Colombo 04	0.035714	0.000000	0.000000	0.107143	0.000000	0.000000	0.035714
1	Cinnamon Gardens, Colombo 07	0.000000	0.055556	0.000000	0.000000	0.027778	0.000000	0.000000
2	Dehiwala, Colombo	0.000000	0.000000	0.000000	0.058824	0.000000	0.000000	0.058824
3	Fort, Colombo 01	0.000000	0.000000	0.021277	0.000000	0.000000	0.000000	0.000000
4	Havelock Town, Colombo 05	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.111111

```
## print each Locality along with the top 5 most common venues
```

```
num_top_venues = 5
```

```
for hood in cmb_grouped['Locality']:
    print("----"+hood+"----")
    temp = cmb_grouped[cmb_grouped['Locality'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_ve
    print('\n')
```

```
----Bambalapitiya, Colombo 04----
      venue freq
0      Asian Restaurant 0.11
1 Vegetarian / Vegan Restaurant 0.11
2          Clothing Store 0.11
3          Jewelry Store 0.07
4          Bookstore 0.07
```

```
----Cinnamon Gardens, Colombo 07----
      venue freq
0     Coffee Shop 0.08
1          Pub 0.06
2 Breakfast Spot 0.06
3        Café 0.06
4 Art Gallery 0.06
```

```
----Dehiwala, Colombo----
      venue freq
0   Women's Store 0.06
1 Candy Store 0.06
2 Malay Restaurant 0.06
3        Gym 0.06
4 Falafel Restaurant 0.06
```

----Fort, Colombo 01----

	venue	freq
0	Hotel	0.11
1	Café	0.09
2	Lounge	0.09
3	Seafood Restaurant	0.09
4	Restaurant	0.04

----Havelock Town, Colombo 05----

	venue	freq
0	Clothing Store	0.22
1	Hotel	0.11
2	South Indian Restaurant	0.11
3	Thai Restaurant	0.11
4	Bakery	0.11

----Kollupitiya, Colombo 03----

	venue	freq
0	Chinese Restaurant	0.10
1	Coffee Shop	0.07
2	Hotel	0.05
3	Bakery	0.05
4	Casino	0.05

----Old Dutch Hospital, Fort----

	venue	freq
0	Hotel	0.09
1	Café	0.09
2	Lounge	0.09

Find all popular venues

```
def popular_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Top 10 venues for each Locality.

num_top_venues = 10

indicators = ['st', 'nd', 'rd']

```
# create columns
columns = ['Locality']
for x in np.arange(num_top_venues):
    try:
        columns.append('{}){} Most Common Venue'.format(x+1, indicators[x]))
    except:
        columns.append('{})th Most Common Venue'.format(x+1))
```

Locality_venues_sorted = pd.DataFrame(columns=columns)

```

Locality_venues_sorted['Locality'] = cmb_grouped['Locality']

for x in np.arange(cmb_grouped.shape[0]):
    Locality_venues_sorted.iloc[x, 1:] = popular_venues(cmb_grouped.iloc[x, :], num_top_ve

Locality_venues_sorted

```

	Locality	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
0	Bambalapitiya, Colombo 04	Vegetarian / Vegan Restaurant	Clothing Store	Asian Restaurant	Jewelry Store	Pizza Place	Bookstore
1	Cinnamon Gardens, Colombo 07	Coffee Shop	Breakfast Spot	Café	Pub	Art Gallery	Bar
2	Dehiwala, Colombo	Women's Store	Sandwich Place	Falafel Restaurant	Cosmetics Shop	Convenience Store	Malay Restaurant
3	Fort, Colombo 01	Hotel	Seafood Restaurant	Lounge	Café	Nightclub	Restaurant
4	Havelock Town, Colombo 05	Clothing Store	South Indian Restaurant	Hotel	Thai Restaurant	Café	Bakery
5	Kollupitiya, Colombo 03	Chinese Restaurant	Coffee Shop	Bakery	Dessert Shop	Casino	Hotel
6	Old Dutch Hospital, Fort	Café	Seafood Restaurant	Hotel	Lounge	Nightclub	Steakhouse
	Rajagiriwa	Chinese	Clothing		Asian		Fast Food

```
# Run k-means to cluster the Locality into 5 clusters.
```

```

kclusters = 5
cmb_clustering = cmb_grouped.drop('Locality', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(cmb_clustering)
kmeans.labels_[0:10]
kmeans.labels_.shape

(9,)

# add clustering labels
cmb_merged = df_trans.head(240)
cmb_merged['Cluster Labels'] = kmeans.labels_

# merge cmb_grouped with df_Chinese to add latitude/longitude for each Locality
cmb_merged = cmb_merged.join(Locality_venues_sorted.set_index('Locality'), on='Locality')

cmb_merged.head()

```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#inplace-mutation-with-loc-and-iloc
 This is separate from the ipykernel package so we can avoid doing imports until

	Locality	Lat	Lng	No_of_Restaurant	Cusines	Agg_Rating	Comments
0	Bambalapitiya, Colombo 04	6.891212	79.857254	2	Chinese, North Indian, Chinese, Sri Lankan	2.900	Average Po
1	Cinnamon Gardens, Colombo 07	6.911791	79.866609	4	Italian, Cafe, Desserts, Cafe, Italian, Cafe, ...	3.825	Good Ve Go
2	Dehiwala, Colombo	6.850283	79.870890	1	Malaysian, North Indian, Sri Lankan	3.500	Go
3	Fort, Colombo 01	6.932547	79.843575	1	American, Steak	4.000	Ve Go
4	Havelock Town, Colombo 05	6.887322	79.872536	2	Juices, Desserts, Cafe, Fast Food, Beverages	4.150	Excellent Go

```
# Map
clusters = folium.Map(location=[latitude, longitude], zoom_start=10)

x = np.arange(kclusters)
ys = [i+x+(i*x)**2 for i in range(kclusters)]
colors = ['red', 'green', 'blue', 'yellow', 'orange']

# add markers
markers_colors = []
for lat, lon, poi, cluster in zip(cmb_merged['Lat'], cmb_merged['Lng'], cmb_merged['Locality'], cmb_merged['Cluster']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color='black',
        fill=True,
        fill_color=colors[cluster],
        fill_opacity=0.7).add_to(clusters)

clusters
```



Leaflet (<http://leafletjs.com>)

```
# Cluster 1
cmb_merged.loc[cmb_merged['Cluster Labels'] == 0, cmb_merged.columns[[1]] + list(range(5, c
```

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common	2th Most Common	3th Most Common	C
# Cluster 2									
cmb_merged.loc[cmb_merged['Cluster Labels'] == 1, cmb_merged.columns[[1] + list(range(5, c									

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
1	6.911791	3.825000	Good, Very Good	563	1	Coffee Shop	Breakfast Spot	Ca	
2	6.850283	3.500000	Good	80	1	Women's Store	Sandwich Place	Fala	Restaura
5	6.905393	3.866667	Average, Very Good	904	1	Chinese Restaurant	Coffee Shop	Bake	

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
# Cluster 3									
cmb_merged.loc[cmb_merged['Cluster Labels'] == 2, cmb_merged.columns[[1] + list(range(5, c									

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
4	6.887322	4.15	Excellent, Good	227	2	Clothing Store	South Indian	Hotel	

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
# Cluster 4									
cmb_merged.loc[cmb_merged['Cluster Labels'] == 3, cmb_merged.columns[[1] + list(range(5, c									

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
8	6.923933	4.1	Very Good	199	3	Hotel	General Travel	Bar	Gov

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
# Cluster 5									
cmb_merged.loc[cmb_merged['Cluster Labels'] == 4, cmb_merged.columns[[1] + list(range(5, c									

	Lat	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1th Most Common Venue	2th Most Common Venue	3th Most Common Venue	C
0	6.891212	2.9	Average, Poor	358	4	Vegetarian / Vegan Restaurant	Clothing Store	Asian Restaurant	

Conclusions

- The best restaurants are available in Old Dutch Hospital, Fort.
- The worst restaurants are available in Bambalapitiya, Colombo 4.
- Kollupitiya, Colombo 3 is the best place to dine in Colombo.
- Bambalapitiya is the best place for Chinese restaurants.
- Chinese Dragon Cafe is the best Chinese restaurant in Colombo.

Cluster 1,4:

It is most recommended for Hotels.

Cluster 2:

It is most recommended for Chinese Restaurants.

Cluster 3:

It is most recommended for Clothing Stores.

Cluster 5:

It is most recommended for Vegetarian / Vegan Restaurants.