

# Capstone Project-The Battle of Neighbourhoods

## 1.1 Introduction

Gurgaon is the capital city of Haryana. Gurgaon is a city just southwest of New Delhi in northern India.

It's known as a financial and technology hub. The city itself has a population of 8.77 lakhs.

The official language of Gurgaon and the one that is most widely spoken is Hindi.

Gurgaon would be incomplete without a taste of its delicious street food

With its diverse culture , comes diverse food items.

There are many restaurants in Gurgaon City, each belonging to different categories like Indian , Chinese , Italian etc.

Questions That can be raised

What is best location in Gurgaon City for Chinese food ?

Which areas have large number of Chinese Restaurant Market ?

Which all areas have less number of restaurant ?

Which is the best place to stay if I prefer Chinese Cuisine ?

What places are have best restaurant in Gurgaon?

## 1.2 Data

For this project we need the following data :

Gurgaon Restaurants data that contains list Locality, Restaurant name, Rating along with their latitude and longitude.

Data source : Zomato kaggle dataset : "<https://www.kaggle.com/shrutimehta/zomato-restaurants-data>"

Description : This data set contains the required information. And we will use this data set to explore various locality of gurgaon city.

Nearby places in each locality of gurgaon city.

Data source : Foursquare API : "<https://developer.foursquare.com/>"

Description : By using this api we will get all the venues in each neighborhood.

## 1.3 Methodology

- Collect the Gurgaon city data from [Zomato kaggle 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)

### 1.3.1 Import Library

```
import pandas as pd
import numpy as np
import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors
# import k-means from clustering stage
from sklearn.cluster import KMeans

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you have not completed the Foursquare API lab
import folium # map rendering library
! pip install geocoder
import geocoder
```

### 1.3.2 Read Data

```
df = pd.read_csv('https://raw.githubusercontent.com/haanjiankur/Capstone-Project---The-Battle-of-Neighborhoods/master/zomato.csv',encoding='ISO-8859-1')
df.head()
```

### 1.3.3 Segregate Data For Gurgaon

```
df_india = df[df['Country Code'] == 1]
df_Gurgaon = df_india[df_india['City'] == 'Gurgaon']
df_Gurgaon.reset_index(drop=True, inplace=True)
df_Gurgaon.head()
```

### 1.3.4 Remove the inconsistent Data

```
df_Res= df_Gurgaon[df_Gurgaon.Longitude !=0.000000][['Restaurant Name','Locality','Longitude','Latitude','Cuisines','Aggregate rating','Rating text','Votes']]

df_Res = df_Res[df_Res['Aggregate rating'] !=0.0]
```

### 1.3.5 Clustering of Restaurants

```
Gurgaon_Rest = folium.Map(location=[28.45, 77.02], zoom_start=12)
```

```
X = df_Res['Latitude']
```

```
Y = df_Res['Longitude']
```

```
Z = np.stack((X, Y), axis=1)
```

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

```
clusters = kmeans.labels_
```

```
colors = ['red', 'green', 'blue', 'yellow', 'orange']
```

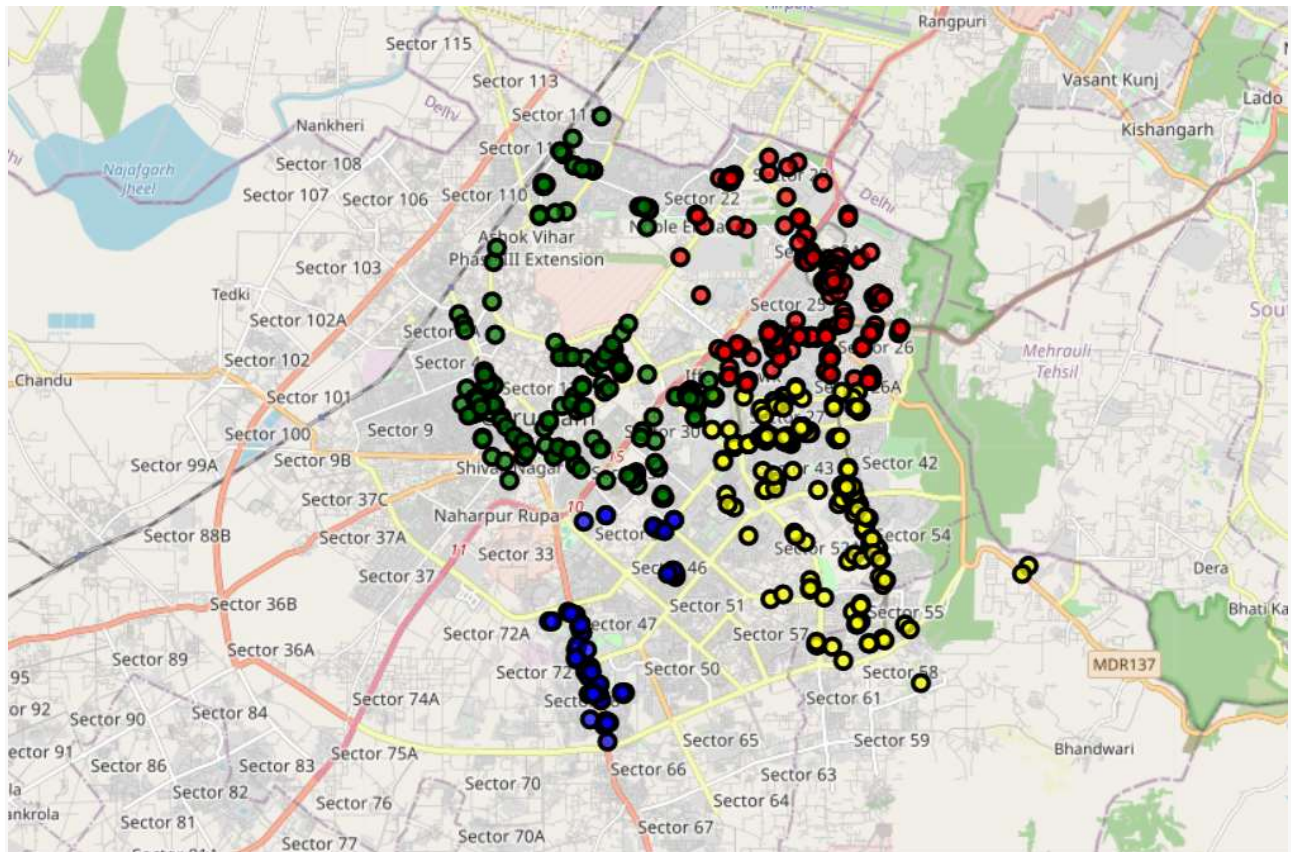
```
df_Res['Cluster'] = clusters
```

```
for latitude, longitude, Locality, cluster in zip(df_Res['Latitude'], df_Res['Longitude'], df_Res['Locality'], df_Res['Cluster']):
```

```
    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(Gurgaon_Rest)
```

```
Gurgaon_Rest
```

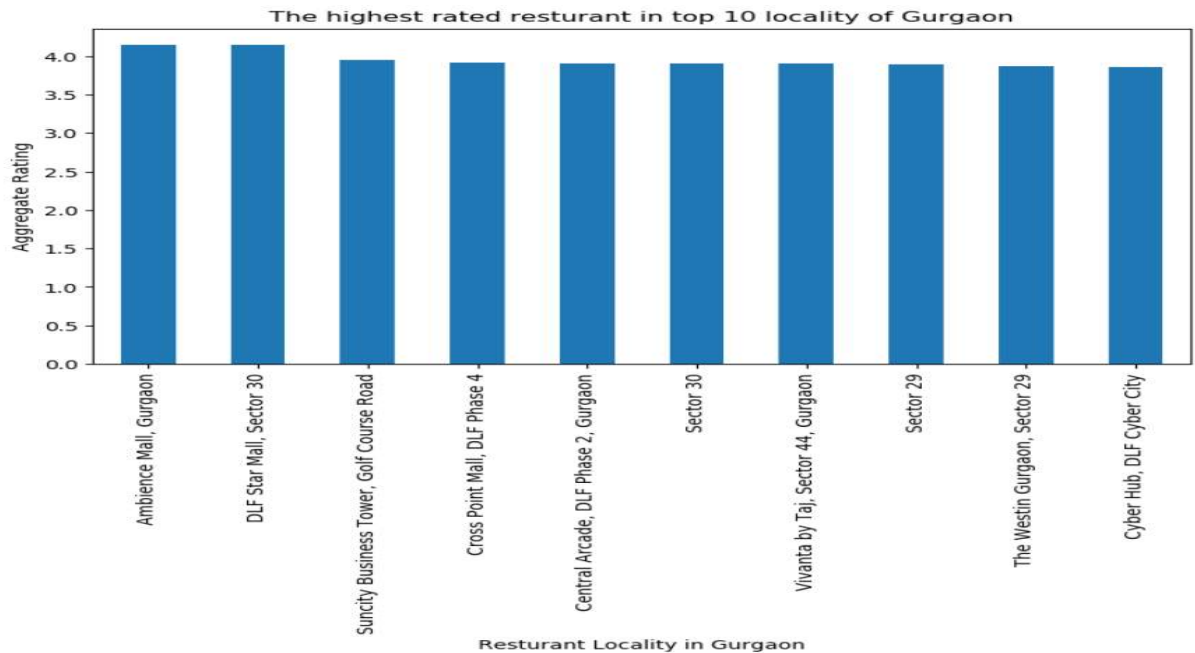


### 1.3.6 List of best Restaurants in Gurgaon

```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The highest rated resturant in top 10 locality of Gurgaon')
#On x-axis

#giving a bar plot
df_Res.groupby('Locality')['Aggregate rating'].mean().nlargest(10).plot(kind='bar')

plt.xlabel('Resturant Locality in Gurgaon')
#On y-axis
plt.ylabel('Aggregate Rating')
#displays the plot
plt.show()
```



### 1.3.7 what places are have worst restaurants in Gurgaon?

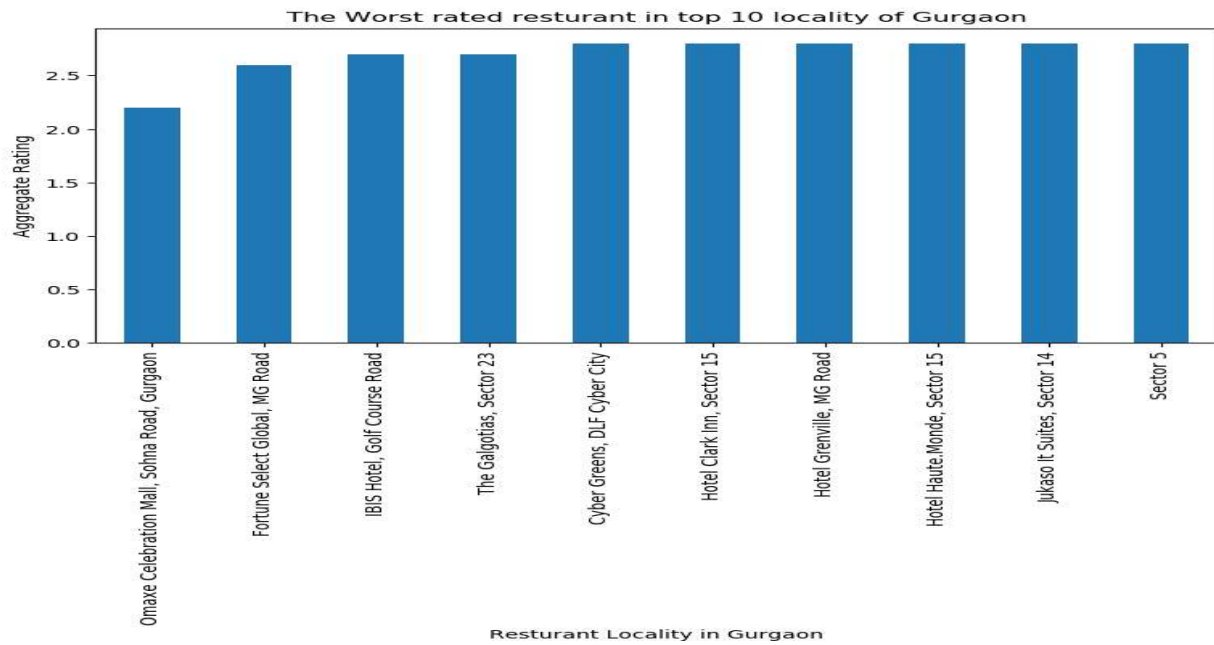
```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The Worst rated resturant in top 10 locality of Gurgaon')
#On x-axis

#giving a bar plot

df_Res.groupby('Locality')['Aggregate rating'].mean().nsmallest(10).plot(kind='bar')

plt.xlabel('Resturant Locality in Gurgaon')
#On y-axis
plt.ylabel('Aggregate Rating')

#displays the plot
plt.show()
```



### 1.3.8 Which place are suitable for edible person in Gurgaon city?

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

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

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

#displays the plot
plt.show()
```



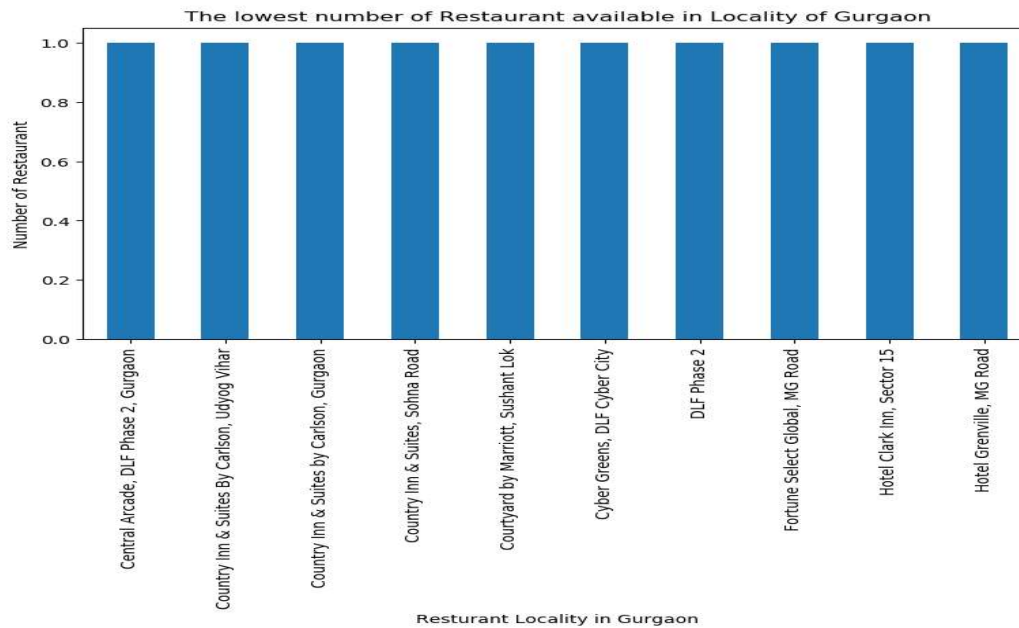
### 1.3.9 Which place are not suitable for edible person in Gurgaon city?

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

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

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

#displays the plot
plt.show()
```



### 1.3.10 What are the best places for Chinese restaurant in Gurgaon city

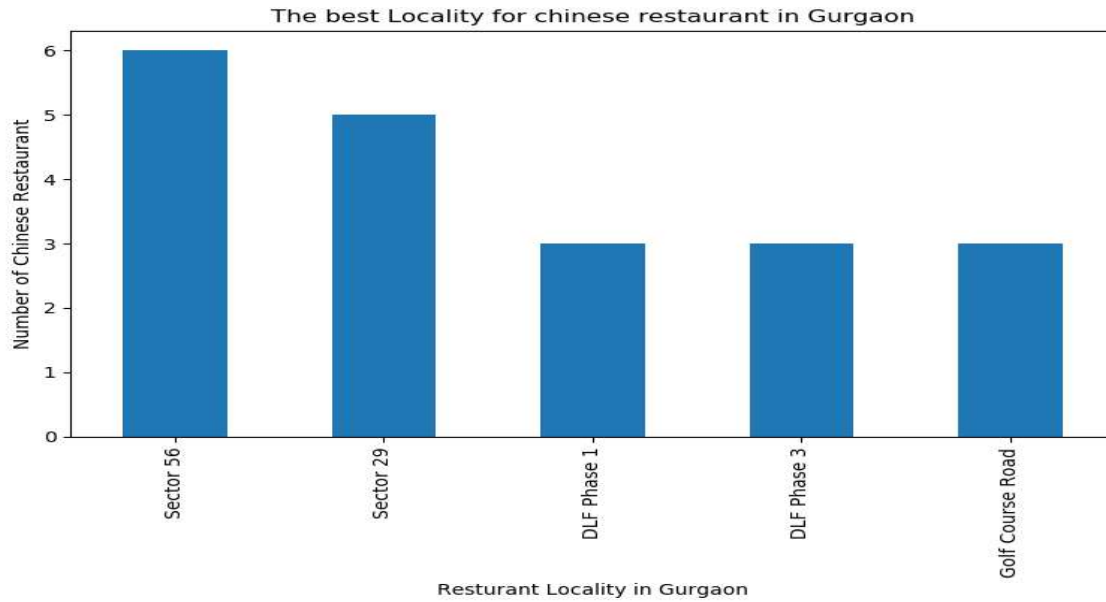
```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The best Locality for chinese restaurant in Gurgaon')
#On x-axis

#giving a bar plot
df_Res[df_Res['Cuisines'].str.startswith('Chinese')].groupby('Locality')['Restau
rant Name'].count().nlargest(5).plot(kind='bar')

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

#displays the plot
plt.show()
```





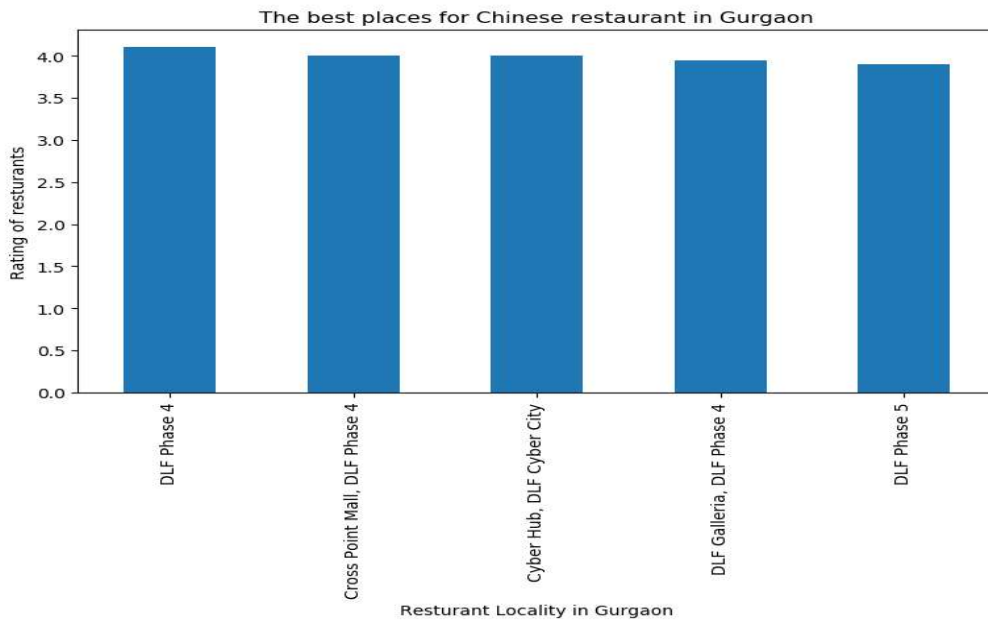
### 1.3.11 which places are the best chinese resturants in Gurgaon?

```
import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The best places for Chinese restaurant in Gurgaon')
#On x-axis

#giving a bar plot
df_Res[df_Res['Cuisines'].str.startswith('Chinese')].groupby('Locality')['Aggregate rating'].mean().nlargest(5).plot(kind='bar')

plt.xlabel('Resturant Locality in Gurgaon')
#On y-axis
plt.ylabel('Rating of resturants')

#displays the plot
plt.show()
```



### 1.3.12 Data transformation Based on Locality grouping the data

```
df_Res_Loc = df_Res.groupby('Locality').count()['Restaurant Name'].to_frame()
df_Res_rating = df_Res.groupby('Locality')['Aggregate rating'].mean().to_frame()
d_Cuisines = df_Res.groupby(['Locality'])['Cuisines'].agg(', '.join).reset_index()
d_R = df_Res.groupby(['Locality'])['Rating text'].unique().agg(', '.join).reset_index()
d_V = df_Res.groupby(['Locality'])['Votes'].sum().to_frame()
d_Lat = df_Res.groupby('Locality').mean()['Latitude'].to_frame()
d_Lng = df_Res.groupby('Locality').mean()['Longitude'].to_frame()
df_final = pd.merge(d_Lat, d_Lng, on='Locality').merge(df_Res_Loc, on='Locality').merge(d_Cuisines, on='Locality').merge(df_Res_rating, on='Locality').merge(d_R, on='Locality').merge(d_V, on='Locality')

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

### 1.3.13 Define Foursquare Credentials and Version

```
CLIENT_ID = '5PD3GVV1S5VLT2BCR4QQC3DWNJPEVQWCLKQ23A2OG2RBJXXJ' # Foursquare ID
CLIENT_SECRET = 'XP51SYC2LY4ZMYGPRNJ5RXXBOULXLBK0NBDS3CIWTAXWVZLD' # Foursquare Secret
VERSION = '20200527' # Foursquare API version

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

### 1.3.14 create a function to repeat the same process to all the Locality in Gurgaon

```
## create a function to repeat the same process to all the Locality in Gurgaon

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

    venues_list=[]
    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={}&v={}&ll={},{}&radius={}&limit={}'.format(
            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_list.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_list for item in venue_list])
    nearby_venues.columns = ['Locality',
                            'Locality Latitude',
                            'Locality Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)
```

### 1.3.15 find the venues in all Gurgaon Locality

```
Gurgaon_venues = getNearbyVenues(names=df_final['Locality'],
                                  latitudes=df_final['Lat'],
                                  longitudes=df_final['Lng']
                                  )
Gurgaon_venues.groupby('Locality').count()
```

```

print('There are {} uniques categories.'.format(len(Gurgaon_venues['Venue Category'].unique())))

## Analyze Each Locality

# one hot encoding
Gurgaon_onehot = pd.get_dummies(Gurgaon_venues[['Venue Category']], prefix="", prefix_sep="")

# add Locality column back to dataframe
Gurgaon_onehot['Locality'] = Gurgaon_venues['Locality']

# move Locality column to the first column
column_list = Gurgaon_onehot.columns.tolist()
column_number = int(column_list.index('Locality'))
column_list = [column_list[column_number]] + column_list[:column_number] + column_list[column_number+1:]
Gurgaon_onehot = Gurgaon_onehot[column_list]

Gurgaon_onehot.head(2000)

Gurgaon_grouped = Gurgaon_onehot.groupby('Locality').mean().reset_index()
Gurgaon_grouped

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

num_top_venues = 5

for hood in Gurgaon_grouped['Locality']:
    print("----"+hood+"----")
    temp = Gurgaon_grouped[Gurgaon_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_venues))
    print('\n')

```

### 1.3.16 Create the new dataframe and display the top 10 venues for the each Locality

```

num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Locality']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
Locality_venues_sorted = pd.DataFrame(columns=columns)

```

```

Locality_venues_sorted['Locality'] = Gurgaon_grouped['Locality']

for ind in np.arange(Gurgaon_grouped.shape[0]):
    Locality_venues_sorted.iloc[ind, 1:] = return_most_common_venues(Gurgaon_grouped.iloc[ind, :], num_top_venues)

Locality_venues_sorted

```

### 1.3.17 Cluster Localities

```

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

# set number of clusters
kclusters = 5

Gurgaon_clustering = Gurgaon_grouped.drop('Locality', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(Gurgaon_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
kmeans.labels_.shape

# add clustering labels
Gurgaon_merged = df_final.head(109)
Gurgaon_merged['Cluster Labels'] = kmeans.labels_

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

Gurgaon_merged.head()

```

### 1.3.18 Create Final Map

```

# create final map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=12)

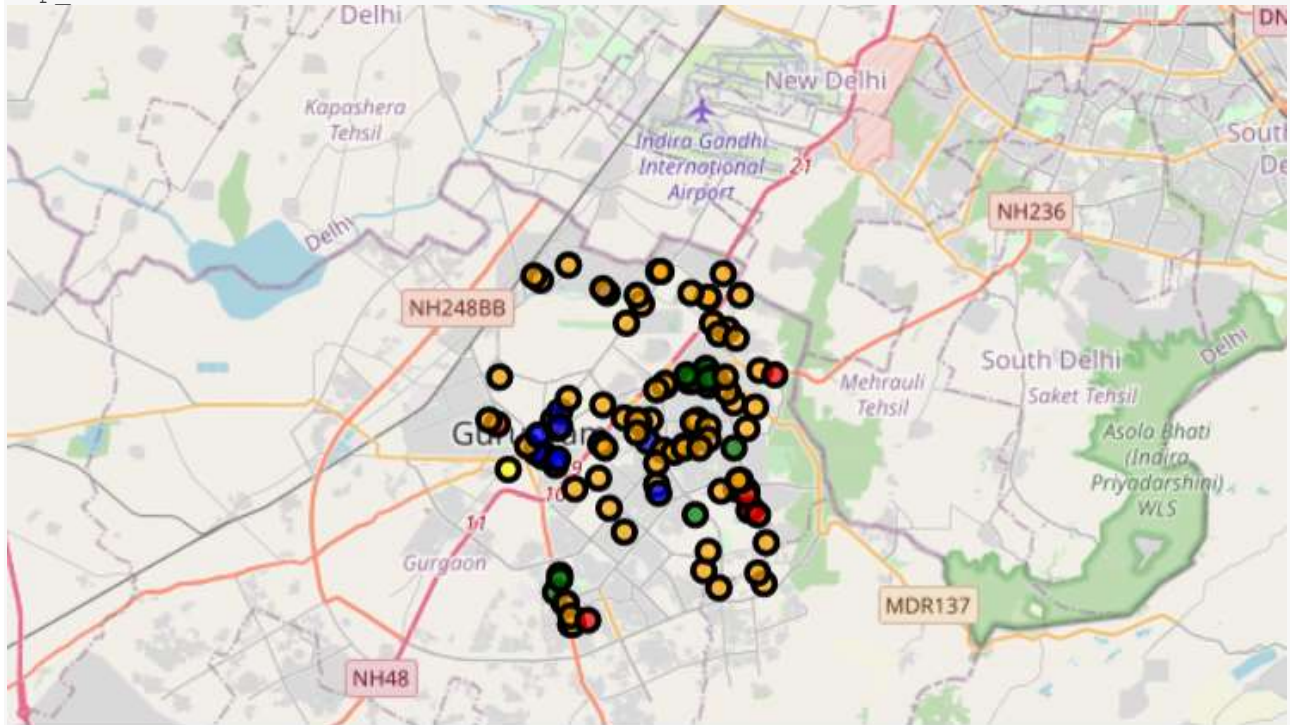
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i+x+(i*x)**2 for i in range(kclusters)]
#colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
#rainbow = [colors.rgb2hex(i) for i in colors_array]
colors = ['red', 'green', 'blue', 'yellow', 'orange']

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(Gurgaon_merged['Lat'], Gurgaon_merged['Lng'],
Gurgaon_merged['Locality'], Gurgaon_merged['Cluster Labels']):
    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(map_clusters)
```

map\_clusters



## 1.4 Results

- Cluster 1 - The Chinese restaurants are most recommended venues nearby the locations.
- Cluster 2 - It is recommended for the Indian Restaurant venues
- Cluster 3 - It is recommended for the Hotel venues areas
- Cluster 4 - It seems to be electronics Market
- Cluster 5 - Fast food, cafes are the most recommended venues

## 1.5 Conclusion

- Sector 56 and Sector 29 are some of the best neighborhoods for Chinese cuisine.
- DLF phase 4, Cross point mall , Cyber Hub have the best Chinese Restaurant.
- DLF phase 3 , Cyber hub , Sector 14 , Sohna Road are the best places for edible person.
- Ambience Mall , DLF Star Mall , Suncity Business Tower have best rated restaurants in Gurgaon.