IBM Data Science Capstone Project: Week 4 & 5

Analyzing Lahore City to find best town w.r.t nearby facilities

Lahore is a city of Pakistan. There are few towns that are old, and few are built in recent past. People come here to seek jobs and to start their own business. For a person planning to settle in Lahore, he/ she will be having difficulty to choose the best town. More facilities surrounded by the town may be a perfect choice for someone. This project aims to provide a overview of all the town w.r.t nearby venues like restaurants, shopping malls etc.

Data Used in Project

- *Data Preparation Phase*: Data of Towns in Lahore is available in Excel Format.
- Geographical coordinates of the Town will be getting from online sources (OpenStreetMap or arcgis)
- Obtain the venue data for the Towns from **Foursquare API**
- Explore and cluster the Towns
- Select the best Town

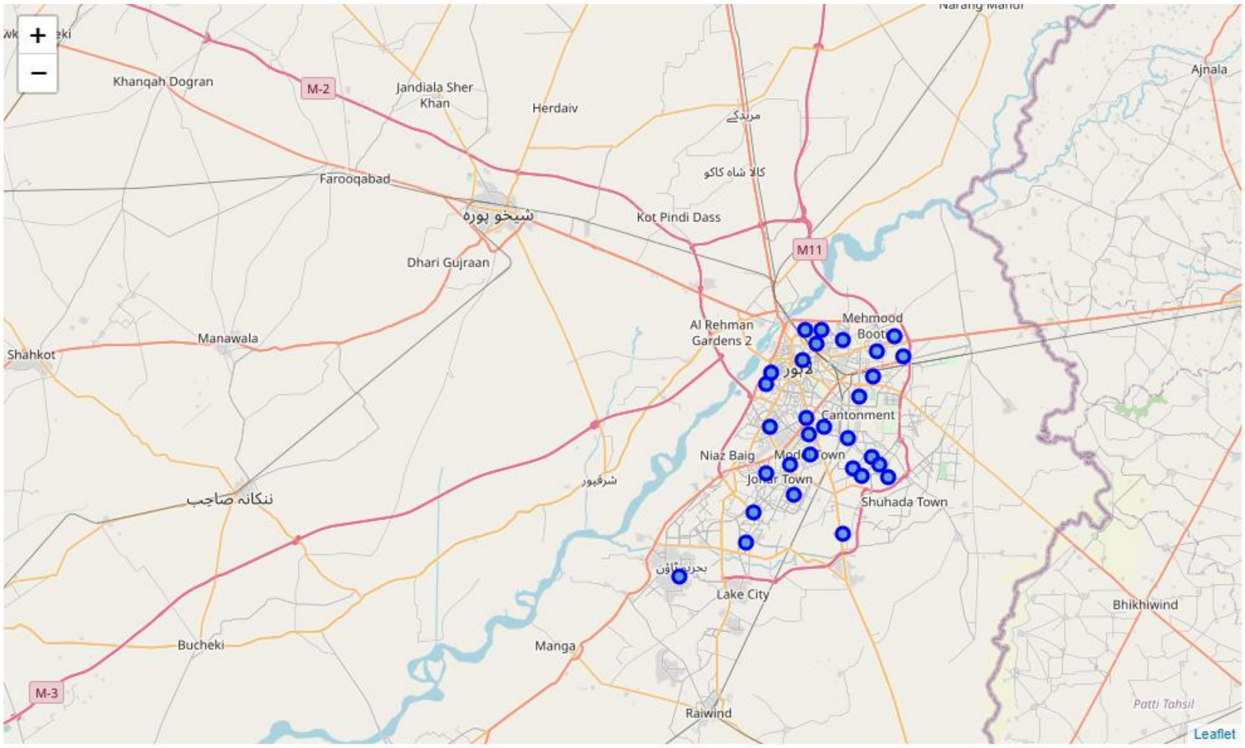
Importing All Necessary Libraries

```
import pandas as pd
import numpy as np
import json
from geopy.geocoders import Nominatim
import geocoder
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

print("Libraries imported.")
```

Loading Data

```
df_lhr = pd.read_excel("D:\\Documents\\Online Courses\\IBM Data Science\\Capstone Project\\Lahore_Towns.xlsx")

df_lhr.head()

df_lhr_loc = pd.read_excel("D:\\Documents\\Online Courses\\IBM Data Science\\Capstone Project\\Lahore_Towns_Location_Info.xlsx")

df_lhr_loc.head()
```

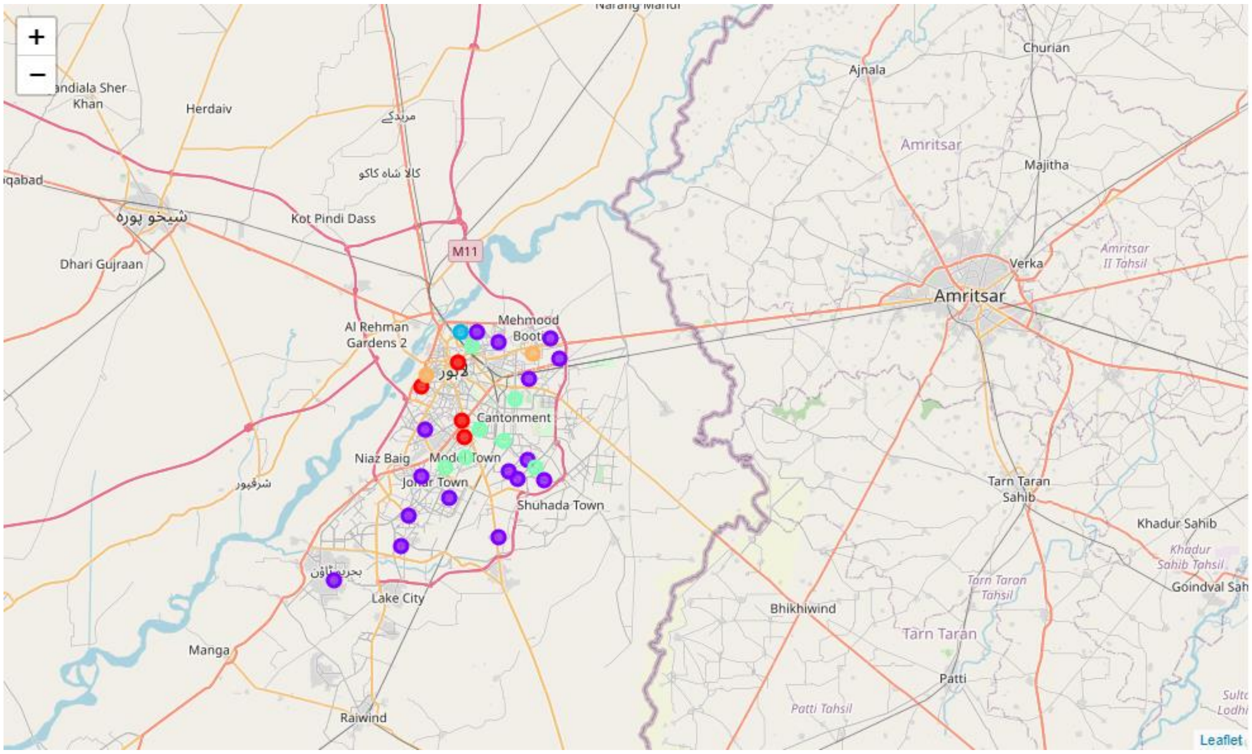
```
df_lhr = pd.merge(df_lhr,df_lhr_loc,how='left',on='Town Names')

df_lhr.head()

address = 'Lahore, Punjab, Pakistan'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Lahore City are {}, {}.'.format(latitude, longitude))

map_lhr = folium.Map(location=[latitude, longitude], zoom_start=10)

for lat, lng, neighborhood in zip(df_lhr['Latitude'], df_lhr['Longitude'], df_lhr['Town Names']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_lhr)

map_lhr
```



```
radius = 2000
```

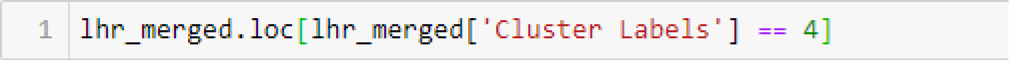
```
LIMIT = 100

venues = []

for lat, long, neighborhood in zip(df_lhr['Latitude'], df_lhr['Longitude'], df_lhr['Town Names']):


    url =
"https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{}&radius={}&
limit={}".format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        long,
        radius,
        LIMIT)


    results = requests.get(url).json()["response"]['groups'][0]['items']


    for venue in results:
        venues.append((
            neighborhood,
            lat,
            long,
            venue['venue']['name'],
            venue['venue']['location']['lat'],
            venue['venue']['location']['lng'],
            venue['venue']['categories'][0]['name']))

venues_df = pd.DataFrame(venues)

venues_df.columns = ['Neighborhood', 'Latitude', 'Longitude', 'VenueName', 'VenueLatitude',
'VenueLongitude', 'VenueCategory']

print(venues_df.shape)
venues_df.head()

venues_df.to_excel("D:\\Documents\\Online Courses\\IBM Data Science\\Capstone
Project\\LHR_Venue_Info.xlsx",index=False)

venues_df.groupby(["Neighborhood"]).count()

print('There are {} uniques categories.'.format(len(venues_df['VenueCategory'].unique())))

kl_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="", prefix_sep="")


kl_onehot['Neighborhoods'] = venues_df['Neighborhood']
```

```
fixed_columns = [kl_onehot.columns[-1]] + list(kl_onehot.columns[:-1])
kl_onehot = kl_onehot[fixed_columns]

print(kl_onehot.shape)
kl_onehot.head()

kl_grouped = kl_onehot.groupby(["Neighborhoods"]).mean().reset_index()

print(kl_grouped.shape)
kl_grouped

lhr_rest = kl_grouped[["Neighborhoods","Pakistani Restaurant"]]

lhr_rest.head()

kclusters = 5

lhr_clustering = lhr_rest.drop(["Neighborhoods"], 1)

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(lhr_clustering)

kmeans.labels_[0:10]

lhr_merged = lhr_rest.copy()

lhr_merged["Cluster Labels"] = kmeans.labels_

lhr_merged.rename(columns={"Neighborhood": "Town Names"}, inplace=True)
lhr_merged.head()

lhr_merged = pd.merge(lhr_merged,df_lhr,on="Town Names",how='left')

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_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

markers_colors = []
for lat, lon, poi, cluster in zip(lhr_merged['Latitude'], lhr_merged['Longitude'], lhr_merged['Town
Names'], lhr_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' - Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

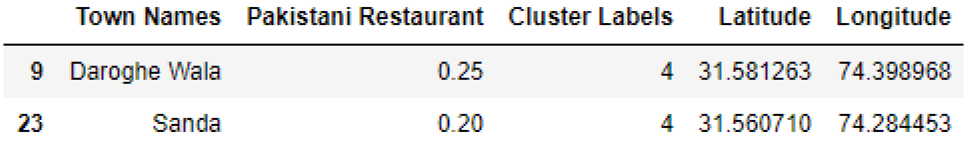


lhr_merged.loc[lhr_merged['Cluster Labels'] == 0]

lhr_merged.loc[lhr_merged['Cluster Labels'] == 1]



```
lhr_merged.loc[lhr_merged['Cluster Labels'] == 4]
```



	Town Names	Pakistani Restaurant	Cluster Labels	Latitude	Longitude
9	Daroghe Wala	0.25	4	31.581263	74.398968
23	Sanda	0.20	4	31.560710	74.284453

Final Findings

As can be seen above, cluster 5 [Index 4] is having high percentage of Pakistani Resturants around them. A foodie person moving into Lahore would prefer this location.