Coursera Peer Graded Assignment - Capstone Project

Introduction

Bangalore, officially Bengaluru, is the capital of the Indian state of Karnataka. It has a population of over ten million, making it a megacity and the third-most populous city and fifth-most populous urban agglomeration in India.

Bengaluru is widely regarded as the "Silicon Valley of India" (or "IT capital of India") because of its role as the nation's leading information technology (IT) exporter. Indian technological organisations such as ISRO, Infosys, Wipro and HAL are headquartered in the city. A demographically diverse city, Bangalore is the second fastest-growing major metropolis in India. Recent estimates of the metro economy of its urban area have ranked Bangalore either the fourth or fifth-most productive metro area of India. It is home to many educational and research institutions in India, such as Indian Institute of Science (IISc), Indian Institute of Management (Bangalore) (IIMB), International Institute of Information Technology, Bangalore (IIITB), National Institute of Fashion Technology, Bangalore, National Institute of Design, Bangalore (NID R&D Campus), National Law School of India University (NLSIU) and National Institute of Mental Health and Neurosciences (NIMHANS). Numerous state-owned aerospace and defence organisations, such as Bharat Electronics, Hindustan Aeronautics and National Aerospace Laboratories are located in the city. The city also houses the Kannada film industry.

Business Problem

With huge development in the city, the population is expected to increase in the coming years. Bangalore's 2020 population is now estimated at 12,326,532. The aim of this project is to find suitable places in the city with best dining experiences.

Target Audience

The target audience of this report is for anyone who wants to open up a new restaurant in Bangalore.

1. Importing required libraries

We start this project by importing all the necessary libraries required.

```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

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
import folium # map rendering library

print('Libraries imported.')
```

2. Download and explore the Dataset

In this step, we download our data from the source and modify the data to form a dataset. This includes cleaning the data, joining the data from different locations to form one complete table, renaming the columns to make it easier to understand etc.

```
url = "https://en.wikipedia.org/wiki/List_of_neighbourhoods_in_Bangalore"
html = requests.get(url).content
df = pd.read_html(html)
print(df)
```

```
df[0]["Region"] = "Central"
df[1]["Region"] = "Eastern"
df[2]["Region"] = "North Eastern"
df[3]["Region"] = "Northern"
df[4]["Region"] = "South Eastern"
df[5]["Region"] = "Southern"
df[6]["Region"] = "Southern suburbs"
df[7]["Region"] = "Western"
```

```
bangalore_data = pd.concat([df[0] , df[1] , df[2] , df[3] , df[4] , df[5] , df[6] , df[7]] , ignore_index = True)
```

```
bangalore_data = bangalore_data.drop(columns = ["Image" , "Summary"])
print(bangalore_data)
```

3. Adding latitudes and longitudes for the neighborhood names

The file contents from the webpage is retrieved into a Pandas dataframe named bangalore_data. The latitudes and longitudes are retrieved using the geocode package. These values are then added as separate columns in the existing dataframe.

```
locator = Nominatim(user_agent="myGeocoder")

from geopy.extra.rate_limiter import RateLimiter

# 1 - conveneint function to delay between geocoding calls
geocode = RateLimiter(locator.geocode, min_delay_seconds=1)

# 2- - create location column
bangalore_data['location'] = bangalore_data['Name'].apply(geocode)

# 3 - create longitude, laatitude and altitude from location column (returns tuple)
bangalore_data['point'] = bangalore_data['location'].apply(lambda loc: tuple(loc.point) if loc else None)

print(bangalore_data)
```

```
bangalore_data["latitudes"] = [item[0] for item in bangalore_data["point"]]
bangalore_data["longitudes"] = [item[1] for item in bangalore_data["point"]]
print(bangalore_data)
```

4. Obtaining Venue Data from FourSquare API

We use FourSquare API to obtain the venue data by providing all the required parameters. The result is stored as a separate dataframe which contains venue data of all the neighborhoods in our first dataframe.

```
CLIENT_ID = 'SHSKPYGRDE1TC11LSGJQ0RBRS4ENYMIKMFUBWO5EJVF5WF1N'
CLIENT_SECRET = 'URS3VNDHHSX2AECQWMUA3SGYSCSQFBNZFGAYTGDFJ1D0CNP3'
VERSION = '20180605'

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

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

    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 = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

```
LIMIT = 20

bangalore_venues = getNearbyVenues(names = bangalore_data['Name'],
                                   latitudes = bangalore_data['latitudes'],
                                   longitudes = bangalore_data['longitudes'])
```

5. Analysing Each Neighborhood - One Hot encoding

One hot encoding is a process in which categorical values are converted into a form that can be provided to ML algorithms to do a better job in prediction. For the K-means Cluster Algorithm, all unique items under Venue Category are one-hot encoded.

```
# one hot encoding
bangalore_onehot = pd.get_dummies(bangalore_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
bangalore_onehot['Neighborhood'] = bangalore_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [bangalore_onehot.columns[-1]] + list(bangalore_onehot.columns[:-1])
bangalore_onehot = bangalore_onehot[fixed_columns]

bangalore_onehot.head()
```

```
bangalore_grouped = bangalore_onehot.groupby('Neighborhood').mean().reset_index()
bangalore_grouped
```

```
num_top_venues = 5

for hood in bangalore_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = bangalore_grouped[bangalore_grouped['Neighborhood'] == 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')
```

```
def return_most_common_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]
```

Now let's create the new dataframe and display the top 10 venues for each neighborhood.

```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
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
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = bangalore_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

6. Clustering Neighborhood

The Venue data is then trained using K-Means Clustering.

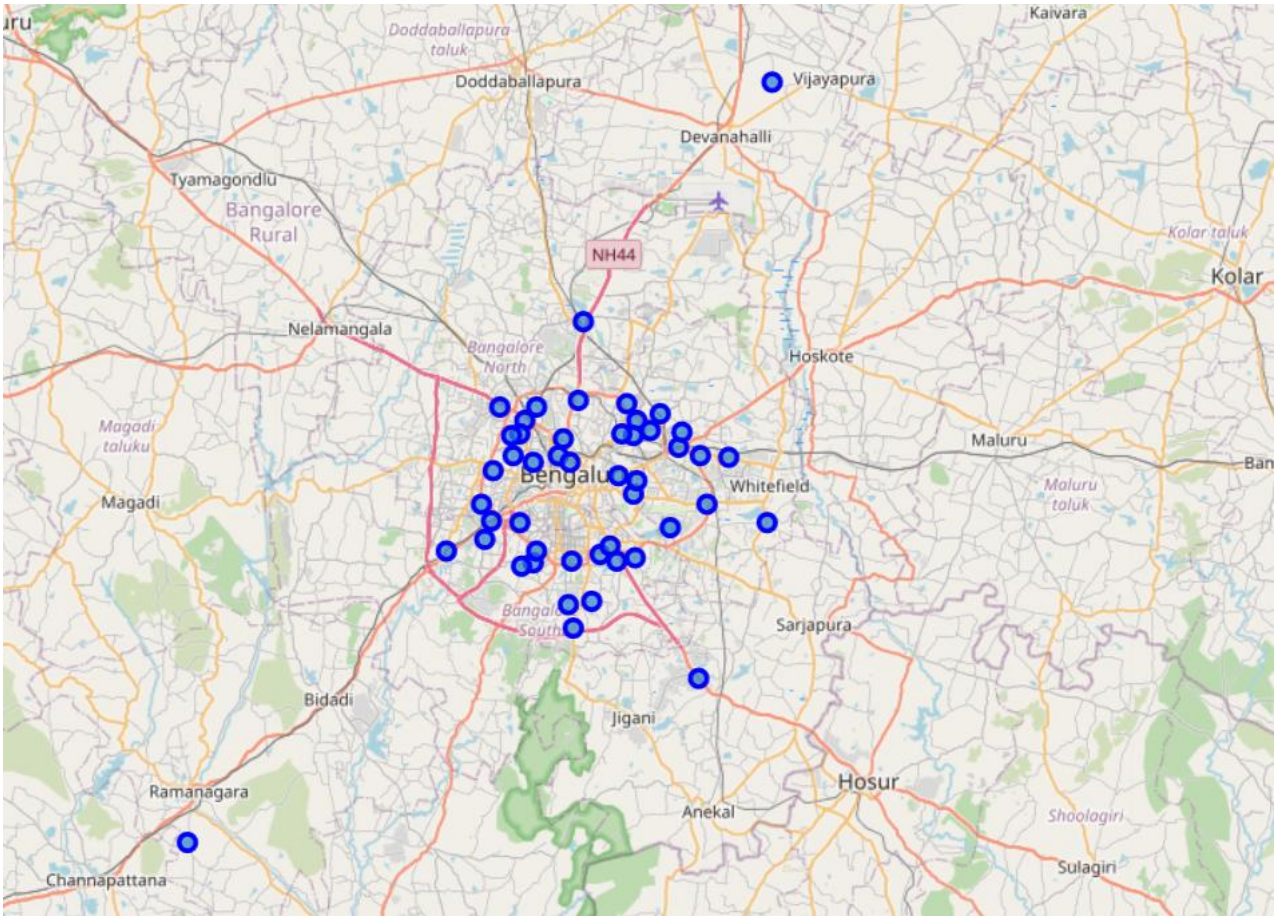
```
# set number of clusters
kclusters = 8

bangalore_grouped_clustering = bangalore_grouped.drop('Neighborhood', 1)

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

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

```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

bangalore_data = bangalore_data.rename(columns = {"Name" : "Neighborhood"})

bangalore_merged = bangalore_data

bangalore_merged = bangalore_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

bangalore_merged = bangalore_merged.dropna(axis = 0)

bangalore_merged.head()
```

7. Folium Map

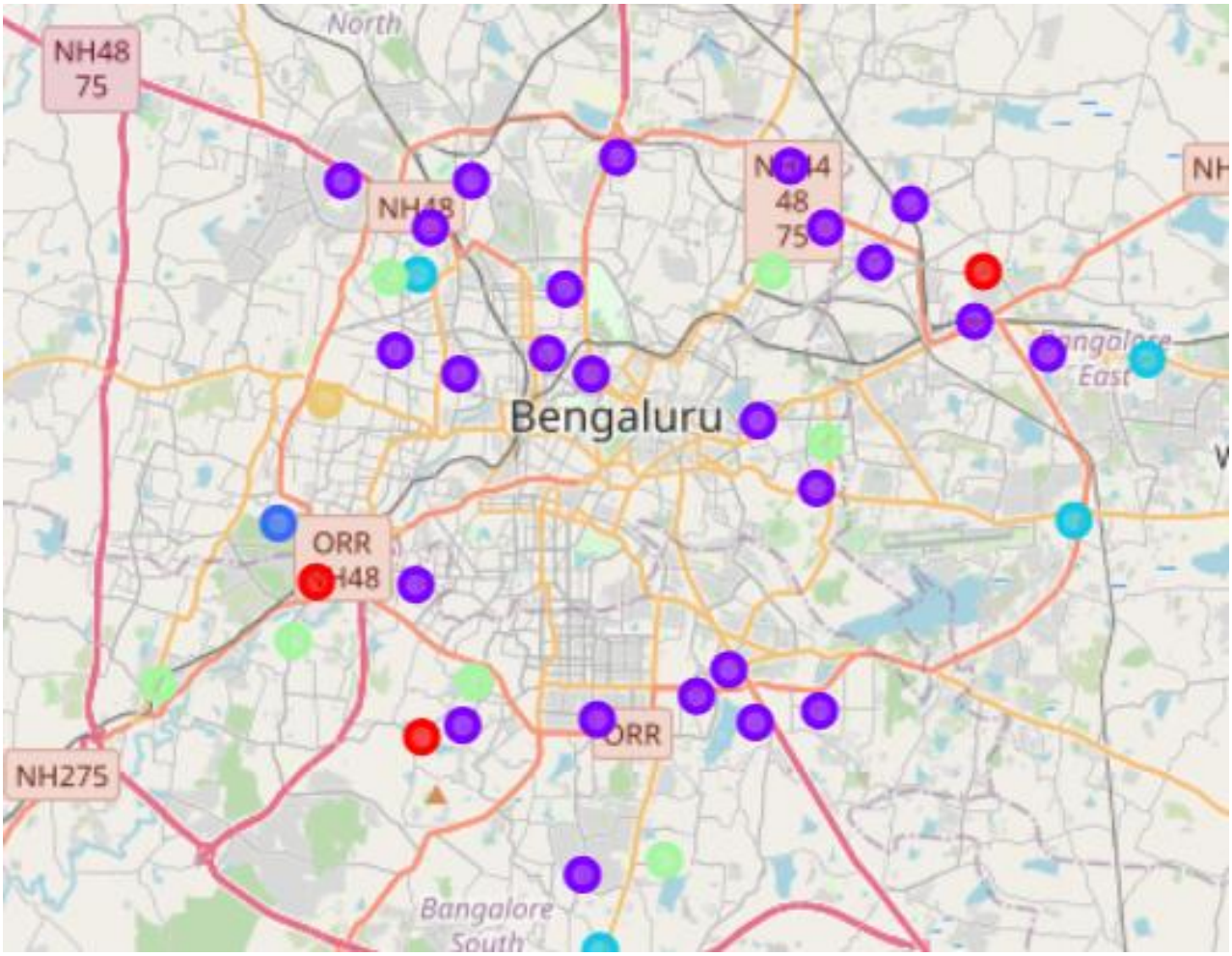
Folium builds on the data wrangling strengths of Python ecosystem and the mapping strengths of leaflet.js.library. All cluster visualisation are done with the help of Folium which in turn generates a Leaflet map made using OpenStreetMap technology.

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# 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]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(bangalore_merged['latitudes'], bangalore_merged['longitudes'], bangalore_merged['Neighborhood'], bangalore_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color= rainbow[int(cluster)-1],
        fill=True,
        fill_color= rainbow[int(cluster)-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```



8. Results

The neighborhoods are divided into 8 clusters. The clustered neighborhoods are visualised using different colours so as to make them more distinguishable.



9. Discussion

After analysing the above cluster, we find that the fourth cluster fits best to answer our question. The top neighborhood places with Restaurants as the most common venue are Shivaji Nagar, C V Raman Nagar, Hoodi, Marathahalli and R T Nagar.

These five places have restaurants of different cuisines as their top most common venue. The different cuisines being Indian, Chinese, Vegetarian and Korean. So depending upon the choice of cuisine, the client can easily choose the areas in which he/she would want to open a restaurant.

11. Conclusion

According to the report "The Rise of Restaurant Industry in India", published by AIMS, (link: "https://theaims.ac.in/resources/rise-of-the-restaurant-industry-in-india.html"), the food industry in India is rapidly expanding. The changing lifestyle, rise of the nuclear family, more women stepping out of their traditional roles to go out and work, rapid urbanization are some of the factors responsible for the growth of the restaurant industry in India. Added to

that is the increased exposure to international lifestyles and cuisines. More and more Indians are demonstrating a growing appetite for a variety of cuisines ranging from Chinese and Italian to Mexican and Middle Eastern.

Greater awareness of global cuisines combined with a larger disposable income is leading many Indian consumers to seek experiential eating or fine dining. Fine dining is not just about going out and eating. Fine dining is about elevating the dining experience of consumers through ambience, décor, presentation of the food, quality of service,use of gourmet ingredients, etc. High-end or fine dining is slowly coming of age in India.

So, the people who are interested in opening a restaurant should not only prioritize the area, but also need to emphasize on the type of cuisine, quality of service, ambience of the restaurant. All these factors builds to providing a great dining experience.