Applied data science capstone project

Week 1

- ***Introduction :***

In this project we will try to find the best place to start our business in coffee shop in New York city(N/Y)

To do that our data that we need will be as follow:

1. Boroughs, Neighborhoods list of N/Y including their latitude and longitude from https://cocl.us/new_york_dataset
2. GeoSpace data to cleary define the boundaries of each boroughs from https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm
3. Coffee shops in each neighbourhood in N/Y using Foursqaure , a data location provider

- ***Methodology :***

By Foursquare we will find all venues for each neighbourhood using N/Y dataset as mentioned in introduction section .

Then we clean , wrangle and filter our data related to coffee shops to be in a suitable form to be processed

After that we will make use of customers rating to reach our goals

Collect the new york city data from https://cocl.us/new_york_dataset

- **Analysis**

We will import the required libraries for python.

1-pandas and numpy for handling data.

2-request module for using FourSquare API.

3-geopy to get co-ordinates of City of New York.

4-folium to visualize the results on a map

[1]:
```
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 # uncomment this line if you haven't completed the Foursquare API lab
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.pyplot as plt
import matplotlib.colors as colors
import os
# 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 haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.5.11
  latest version: 4.7.12

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##
```

[2]:
```
url = 'https://cocl.us/new_york_dataset'
urlRead = requests.get(url).json()
```

[3]:
```
def location(address):
    geolocator = Nominatim(user_agent="ny_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    print('The geograpical coordinate of New York City are {}, {}.'.format(latitude, longitude))
    return  latitude, longitude
```

[4]:
```
def read_data():

    features = urlRead['features']

    # define the dataframe columns
    column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
    NY_data = pd.DataFrame(columns=column_names)

    for data in features:
        borough = data['properties']['borough']
        neighborhood_name = data['properties']['name']

        neighborhood_latlon = data['geometry']['coordinates']
        neighborhood_lat = neighborhood_latlon[1]
        neighborhood_lon = neighborhood_latlon[0]

        NY_data = NY_data.append({'Borough': borough,
                                  'Neighborhood': neighborhood_name,
                                  'Latitude': neighborhood_lat,
                                  'Longitude': neighborhood_lon}, ignore_index=True)
    return NY_data
```

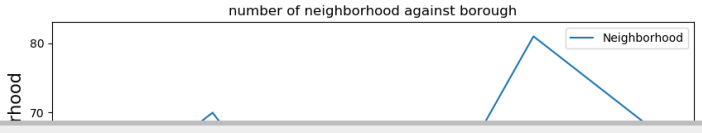
[5]:
```
newyork_data = read_data()
```

[6]:
```
newyork_data.head(10)
```

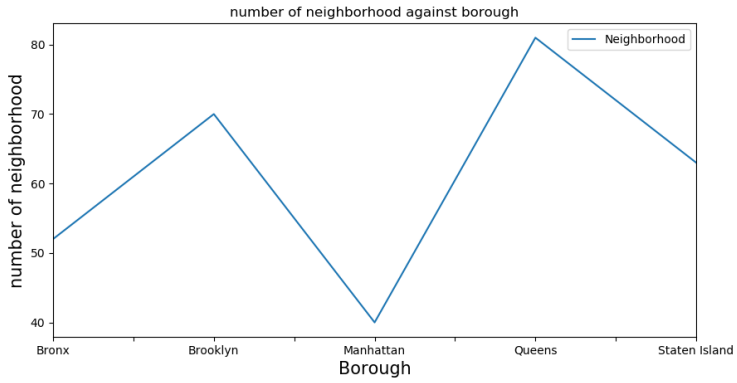
[6]:

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585
5	Bronx	Kingsbridge	40.881687	-73.902818
6	Manhattan	Marble Hill	40.876551	-73.910660
7	Bronx	Woodlawn	40.898273	-73.867315
8	Bronx	Norwood	40.877224	-73.879391
9	Bronx	Williamsbridge	40.881039	-73.857446

[7]:
```
plt.figure(figsize = (10,5) , dpi =100)
plt.title('number of neighborhood against borough')
plt.xlabel('borough name',fontsize = 15 )
plt.ylabel('number of neighborhood',fontsize = 15 )
newyork_data.groupby('Borough')['Neighborhood'].count().plot(kind= 'line')
plt.legend()
plt.show()
```



```
plt.legend()
plt.show()
```



[13]:
```
def venue(lat,lng):

    radius = 1000
    LIMIT = 100
    CLIENT_ID = '40ORBIYY0FUZGQUJE2UU3Z2VCSJ5ORNOYTUJESUI1PYAQMVD'
    CLIENT_SECRET ='AGLN2ZXROGTF5AAHQ5NSPLLGRC5OG1I3A0LTAJUBFBYHBKM1'
    VERSION = '20180605'
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        neighborhood_latitude,
        neighborhood_longitude,
        radius,
        LIMIT)
```

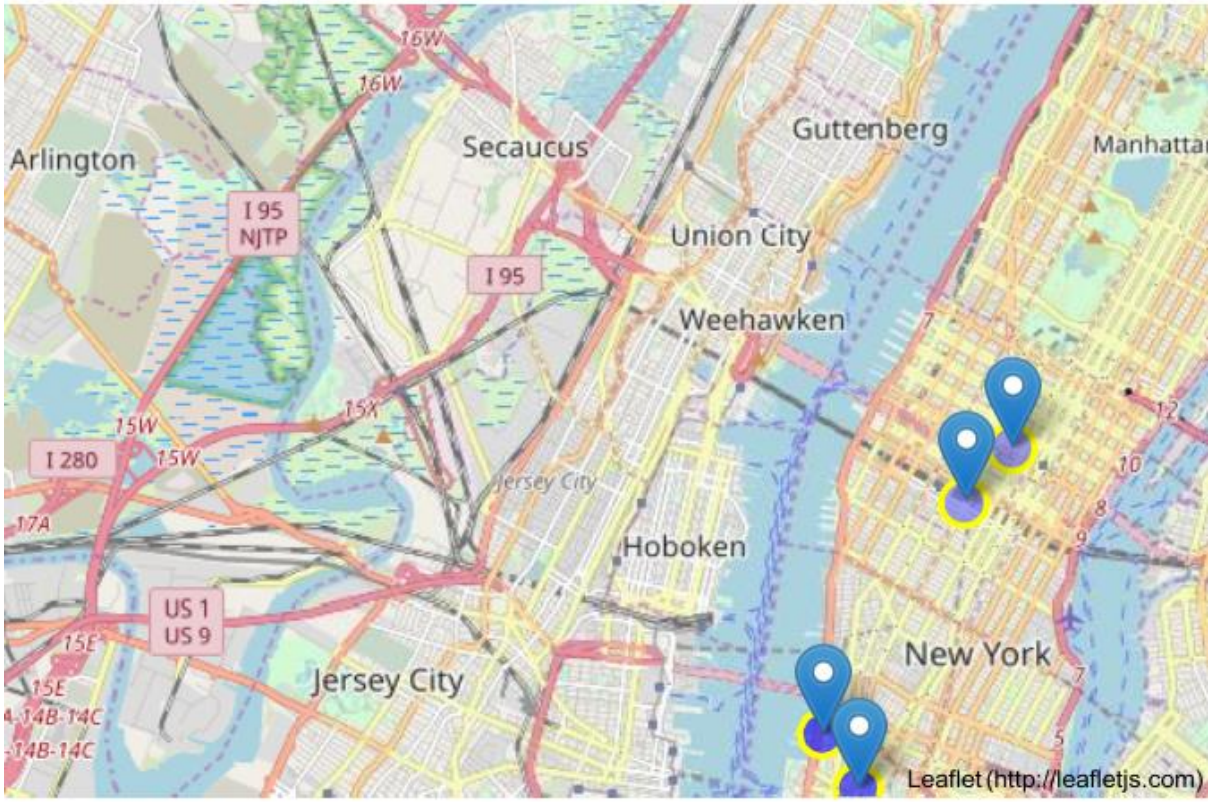
[13]:
```
def venue(lat,lng):

    radius = 1000
    LIMIT = 100
    CLIENT_ID = '40ORBIYY0FUZGQUJE2UU3Z2VCSJ5ORNOYTUJESUI1PYAQMVD'
    CLIENT_SECRET ='AGLN2ZXROGTF5AAHQ5NSPLLGRC5OG1I3A0LTAJUBFBYHBKM1'
    VERSION = '20180605'
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        neighborhood_latitude,
        neighborhood_longitude,
        radius,
        LIMIT)
    venues_details = []


    results = requests.get(url).json()
    venues_data = results['response']['groups'][0]['items']
    for row in venues_data:
        try:
            venue_id = row['venue']['id']
            venue_name=row['venue']['name']
            venue_category=row['venue']['categories'][0]['name']
            venues_details.append([venue_id,venue_name,venue_category])
        except KeyError:
            pass
    column_names = ['ID', 'Name', 'Category']
    df = pd.DataFranme(venue_details,columns = column_names)
    return df
```

[15]:
```
def get_venues_details(venur_id):
    CLIENT_ID = '40ORBIYY0FUZGQUJE2UU3Z2VCSJ5ORNOYTUJESUI1PYAQMVD'
    CLIENT_SECRET ='AGLN2ZXROGTF5AAHQ5NSPLLGRC5OG1I3A0LTAJUBFBYHBKM1'
    VERSION = '20180605'
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(

    results = requests.get(url).json()
    venues_data = results['response']['venue']
    for row in venues_data:
        try:
            venue_id = venue_data['id']
            venue_name=venue_data['name']
            venue_like=venue_data['likes']['count']
        except KeyError:
```



- ***Conclusion*** :

1- **Astoria**(**Queens**), **Blissville**(Queens), **Civic Center(**Manhattan) are some of the best neighborhoods for coffee shops

2- **Manhattan** have potential coffee shops

3- **Staten Island** ranks last in average rating coffee shops

4- **Manhattan** is the best place for coffee shops