Capstone Project: The Battle of Neighbourhoods

Table of Contents

Introduction

Description & Discussing of the Background

My Capstone Project will be using the city of London to show that when driven by venue and location data from FourSquare, it is possible to present potential house renters and buying with a list of attractive areas to rent or buy based the number/type of restaurant located in the different boroughs of London city.
London City demographic show that it is a large and ethnically diverse metropolis, It's on of the top 20 largest city in the continent of Europe and the world by population.
The population of London has grown considerably over the last decade. London is very diverse and this represent what is known as the British Empire.

Target Audience
Considering the diversity of London, a food lover would prefer renting a buying in areas where they have access to a variety of international cuisine at an affordable price. As such , this project will help in pointing our location in London the provides that need

Data Description¶

In order to achieve the goal of this project, we will be using the below data source
London City data that contains list Boroughs, Neighbourhoods along with their latitude and longitude. Data source: `https://en.wikipedia.org/wiki/List_of_areas_of_London`

- Description: This data set contains the required information. And we will use this data set to explore various neighbourhoods of London city.
- Social amenities in each neighbourhood of new London city. Data source: Fousquare API Description: By using this API we will get all the venues in each neighbourhood.
- We can filter these venues to get the type and number of social amenities like schools and hospital in each Boroughs.

Neighbourhood has a total of 34 borough. In order to segment the neighbourhoods and explore them, we will essentially need a dataset that contains the 34 boroughs and the neighbourhoods that exist in each borough as well as the the latitude and longitude coordinates of each neighbourhood.

Dataset 1:
In this project, London will used as synonymous to the "Greater London Area" in the project. Within the Greater London Area, the are areas that are within the London Area Postcode. The focus of this project will be the neighbourhoods are that are within the London Post Code area

The we scrapped of the Wikipedia page of the Greater London Area data is provided below:

IMPORT DEPENDENCIES

[1]:

```
!pip -q install folium
print('folium installed...')
import folium
print('folium imported...')

#library to handle data in a vectorized manner
import numpy as np
print('numpy imported...')

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

#library to handle JSON file
import json
print('json imported...')

!pip -q install geopy
# conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Four
square API lab
print('geopy installed...')
# convert an address into latitude and longitude values
from geopy.geocoders import Nominatim
print('Nominatim imported...')

# library to handle request
import requests
print('requests imported...')

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors
import matplotlib.pyplot as plt
%matplotlib inline
print('matplotlib imported...')

# transform JSON file into a pandas dataframe
from pandas.io.json import json_normalize
print('json_normalize imported...')

# library used for data visualization
import seaborn as sns
print('seaborn imported...')

# import k-means from clustering stage
from sklearn.cluster import KMeans
print('k-means imported...')

#import time
import time
print('time imported...')

# library for pulling data out of HTML and XML files
from bs4 import BeautifulSoup
print('BeautifulSoup imported...')

# install Geocoder
!pip -q install geocoder
print('geocoder installed...')
import geocoder
print('geocoder imported...')

print('DONE - All Dependencies imported...')
```

Retrieve Data from Data source

```
ı [2]: wiki_data_source_link = 'https://en.wikipedia.org/wiki/List_of_areas_of_London'
       headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefo
       x/64.0'}
       wikpedia_page = requests.get(wiki_data_source_link, headers = headers)
       wikpedia_page
```

```
ıt[2]: <Response [200]>
```

```
ı [3]: # Clean html file
       soup = BeautifulSoup(wikpedia_page.content, 'html.parser')

       # This extract the 'tbody' within the table where class is 'Wikitable sortable'
       table = soup.find('table', {'class':'wikitable sortable'}).tbody

       #Extraact all "tr" within the table above
       rows = table.find_all('tr')

       #Extracts the column headers, remove and replaces possible '\n' with space f or the "th" tag
       columns = [i.text.replace('\n', '')
                  for i in rows[0].find_all('th')]

       #Converts columns to pd dataframe
       df = pd.DataFrame(columns = columns)
       df
```

```
: # Remove [] from borough name extracted from wikipidia
  df['Borough'] = df['Borough'].map(lambda x: x.rstrip(']').rstrip('123456789').rstrip('['))
  df.head(5)
```

:

	Location	Borough	Town	Postcode	Dial Code	OSGridRe
0	Abbey Wood	Bexley, Greenwich	LONDON	SE2	020	TQ465785
1	Acton	Ealing, Hammersmith and Fulham	LONDON	W3, W4	020	TQ205805
2	Addington	Croydon	CROYDON	CR0	020	TQ375645
3	Addiscombe	Croydon	CROYDON	CR0	020	TQ345665
4	Albany Park	Bexley	BEXLEY, SIDCUP	DA5, DA14	020	TQ478728

```
: df.shape
```

```
: (533, 6)
```

```
: # To identify distint location, we'll need to assign individual location to each post code
  df_unique_postcode = df.drop('Postcode', axis=1).join(df['Postcode'].str.split(',', expand=True).s
  tack().reset_index(level=1, drop=True).rename('Postcode'))
```

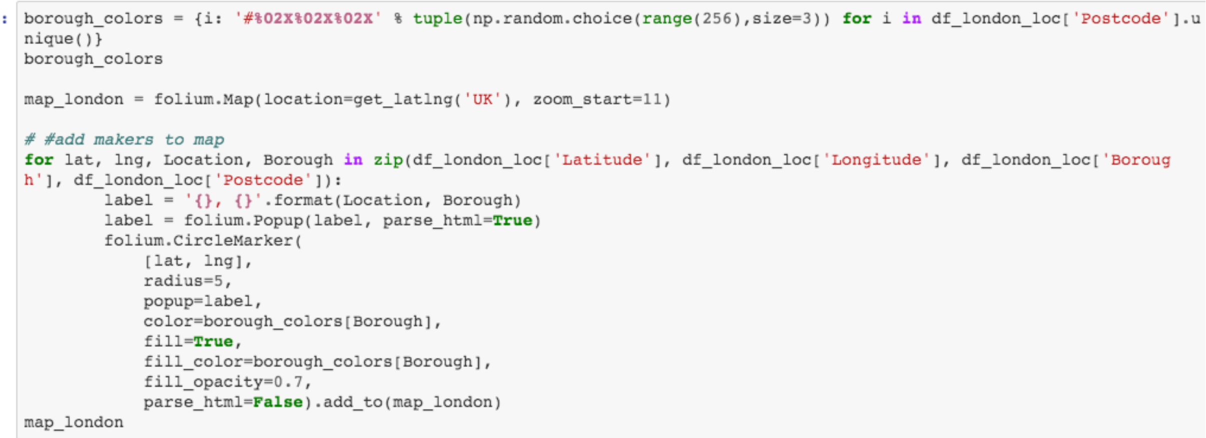
```
: df_unique_postcode.head()
```

:

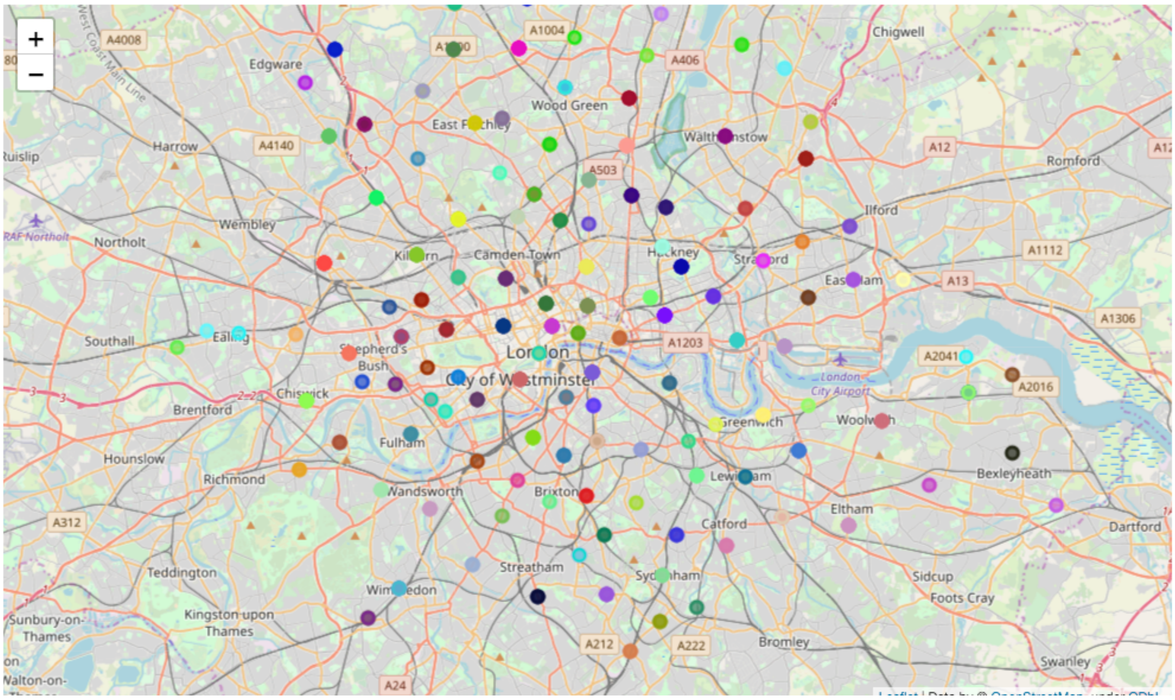
	Location	Borough	Town	Dial Code	OSGridRe	Postcode
0	Abbey Wood	Bexley, Greenwich	LONDON	020	TQ465785	SE2
1	Acton	Ealing, Hammersmith and Fulham	LONDON	020	TQ205805	W3
1	Acton	Ealing, Hammersmith and Fulham	LONDON	020	TQ205805	W4
10	Angel	Islington	LONDON	020	TQ345665	EC1
10	Angel	Islington	LONDON	020	TQ345665	N1

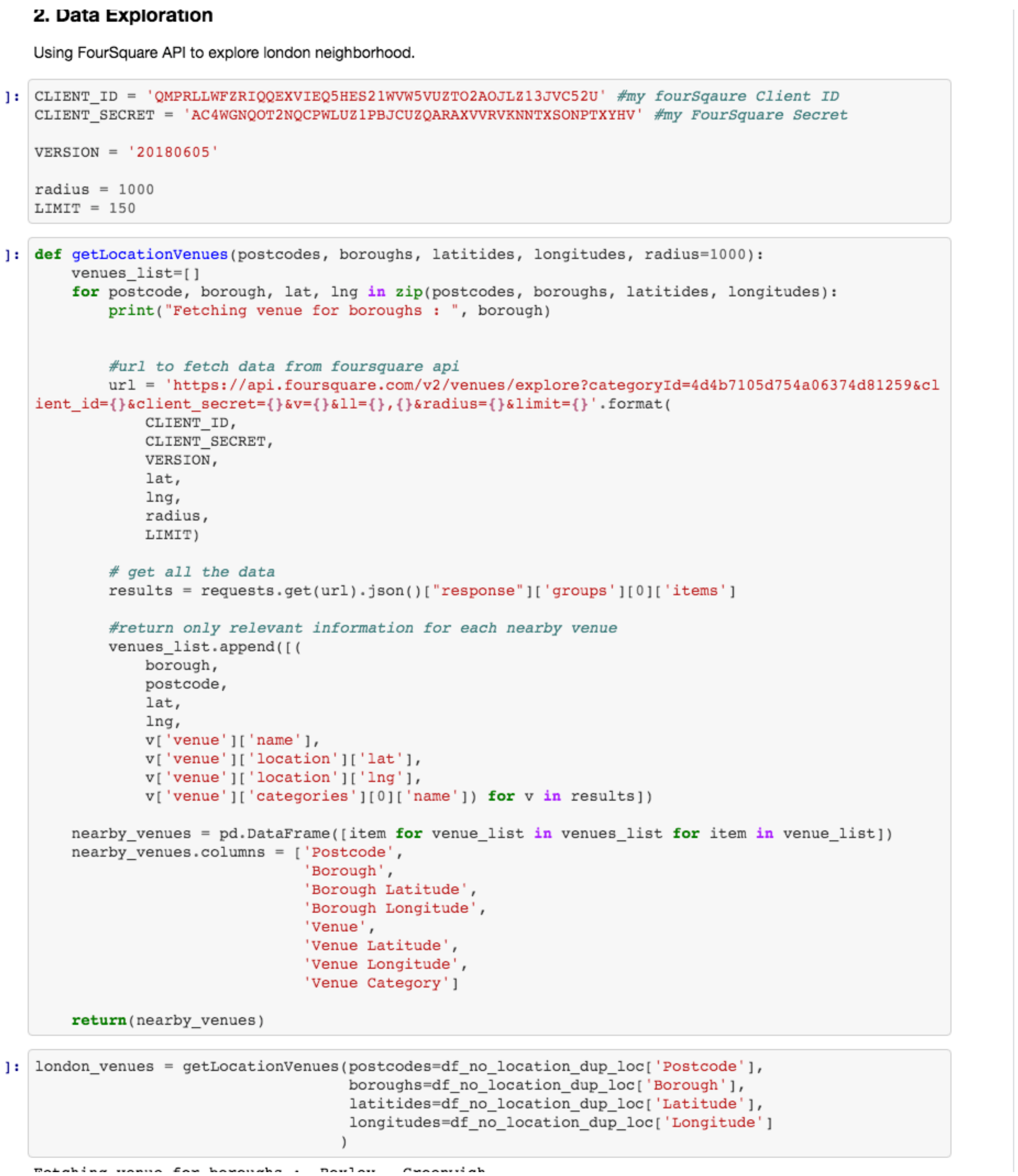
```
: df_unique_postcode.shape
```

```
: (637, 6)
```



```
borough_colors = {i: '#%02X%02X%02X' % tuple(np.random.choice(range(256),size=3)) for i in df_london_loc['Postcode'].u
nique()}
borough_colors

map_london = folium.Map(location=get_latlng('UK'), zoom_start=11)

# #add makers to map
for lat, lng, Location, Borough in zip(df_london_loc['Latitude'], df_london_loc['Longitude'], df_london_loc['Boroug
h'], df_london_loc['Postcode']):
        label = '{}, {}'.format(Location, Borough)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color=borough_colors[Borough],
            fill=True,
            fill_color=borough_colors[Borough],
            fill_opacity=0.7,
            parse_html=False).add_to(map_london)
map_london
```





2. Data Exploration

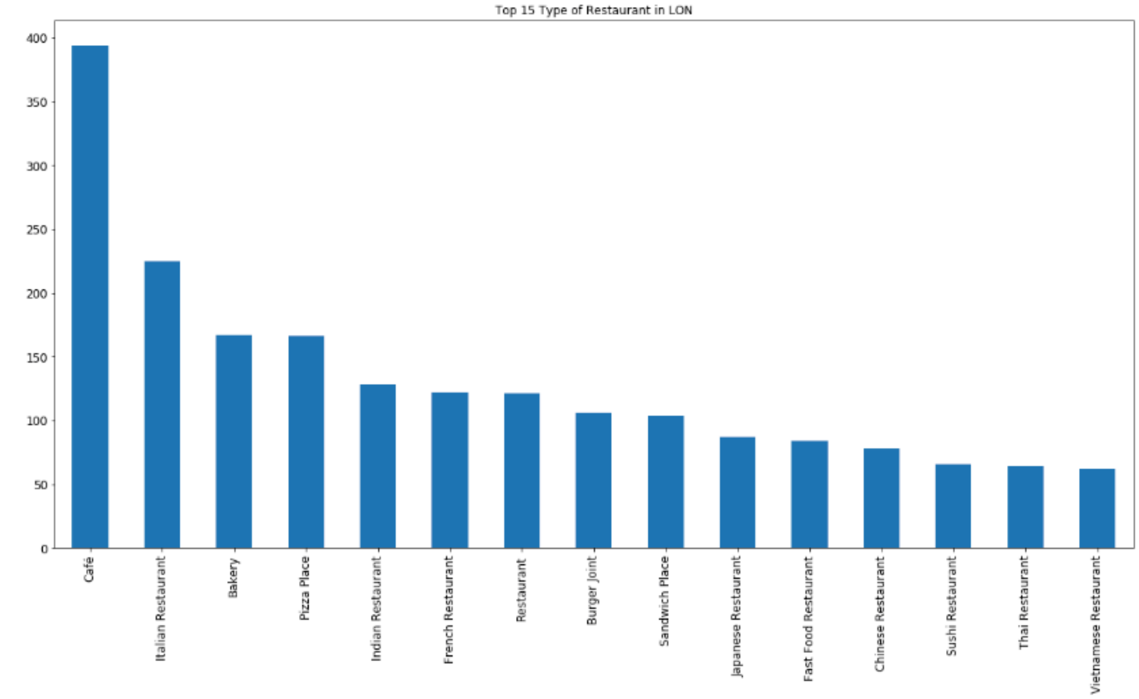
Using FourSquare API to explore london neighborhood.

```
]: CLIENT_ID = 'QMPRLLWFZRIQQEXVIEQ5HES21WVW5VUZTO2AOJLZ13JVC52U' #my fourSqaure Client ID
   CLIENT_SECRET = 'AC4WGNQOT2NQCPWLUZ1PBJCUZQARAXVVRVKNNTXSONPTXYHV' #my FourSquare Secret

   VERSION = '20180605'

   radius = 1000
   LIMIT = 150
```

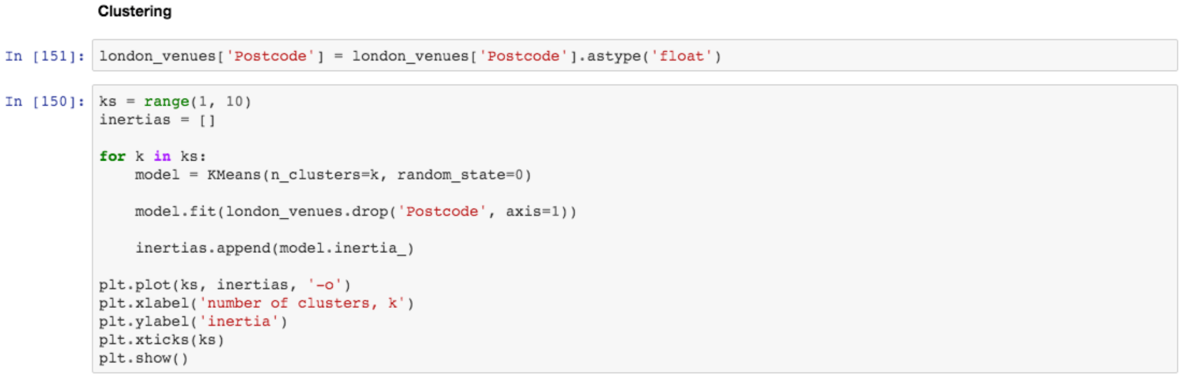
```
]: def getLocationVenues(postcodes, boroughs, latitides, longitudes, radius=1000):
       venues_list=[]
       for postcode, borough, lat, lng in zip(postcodes, boroughs, latitides, longitudes):
           print("Fetching venue for boroughs : ", borough)


           #url to fetch data from foursquare api
           url = 'https://api.foursquare.com/v2/venues/explore?categoryId=4d4b7105d754a06374d81259&cl
   ient_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
               CLIENT_ID,
               CLIENT_SECRET,
               VERSION,
               lat,
               lng,
               radius,
               LIMIT)

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

           #return only relevant information for each nearby venue
           venues_list.append([(
               borough,
               postcode,
               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 = ['Postcode',
                                'Borough',
                                'Borough Latitude',
                                'Borough Longitude',
                                'Venue',
                                'Venue Latitude',
                                'Venue Longitude',
                                'Venue Category']

       return(nearby_venues)
```

```
]: london_venues = getLocationVenues(postcodes=df_no_location_dup_loc['Postcode'],
                                     boroughs=df_no_location_dup_loc['Borough'],
                                     latitides=df_no_location_dup_loc['Latitude'],
                                     longitudes=df_no_location_dup_loc['Longitude']
                                    )
```

```
Fetching venue for boroughs :  Bexley , Greenwich
```



K-Cluster



Clustering

```
In [151]: london_venues['Postcode'] = london_venues['Postcode'].astype('float')
```

```
In [150]: ks = range(1, 10)
          inertias = []

          for k in ks:
              model = KMeans(n_clusters=k, random_state=0)

              model.fit(london_venues.drop('Postcode', axis=1))

              inertias.append(model.inertia_)

          plt.plot(ks, inertias, '-o')
          plt.xlabel('number of clusters, k')
          plt.ylabel('inertia')
          plt.xticks(ks)
          plt.show()
```

Conclusion

Berent and Harrow, seems have a high cluster of café, restaurant and some high end fast food restaurants. This is good as further analysis on shows that Brent is just outside the main city of London and which is overall less expensive then inner London Locations