# Coursera Capstone Project

**The Battle of Neighborhoods**

*Presentation by Laima Marcin*

## ***Finding the best place to open an Italian Restaurant in London***

## TABLE OF CONTENTS:



**PART 1 - A description of the problem and a discussion of the background**

**1.1 Target audience**

This project is intended to help people who are planning to open an Italian restaurant in London how to choose the right location by providing data about each borough of London and finding out the top venues for each borough and checking if there are many other Italian restaurants already present in that area.

- The **goal** is to find the place with least competitors and highest population.

**1.2 Discussion of the Background**

- London is the capital and largest city of England and the United Kingdom.
- London's 2020 population is now estimated at 9,304,016 and it keeps growing.
- London is one of the most multicultural cities on the planet.
- One of the world's most visited cities, London has something for everyone: from history and culture to fine food. With such diversity, London's cultural dynamism makes it among the world's most international cities.
- London is a city where businesses thrive.
- **The Italian cuisine is at the top list of the Londoners' diet.**
- (M. Sannino, E. Robustelli, A. Biccario) – ***It is widely claimed that Londoners are obsessed with the Italian food,*** or at least, the majority of them.
- Such popularity of the 'Made in Italy' is due to the quality of their products but also due to the intense promotion made over the last few years.

**1.3 Description of the Problem**

Finding the best place to open an Italian restaurant in London requires some careful consideration, research and preparation. Since there are over 39,338 food service establishments in London (restaurants, coffee shops, food halls) we need deeper insight from available data in order to be able to make a decision where to establish the first Italian restaurant.

**PART 2 - Data Acquisition and Processing¶**

In this project, I will be using the following datasets to help solve my problem

- List of London Boroughs (from **Wikipedia** page), and **Foursquare API**.

Information on boroughs and their population & coordinates

- Population can be used to determine how big and how dense the specific borough is.
- Coordinates can be used to get neighborhood data from Foursquare and finding the most popular venues in each borough and then clustering them by using K-means and analysing each cluster and finding which cluster has least restaurants and least Italians restaurants and I will provide my observation which clusters are most suitable and I will create a map using Folium library to show these clusters on London map.

**2.1 - Data Source**

**Wikipedia** url: https://en.wikipedia.org/wiki/List_of_London_boroughs

**Foursquare API**

- List of top 50 popular places in the neighborhood
- source: Foursquare
- url: https://api.foursquare.com

**2.2 Data processing**

- Create a dataframe consisting of the columns BoroughName = [] Population = [] Coordinates = [] by using BS4, BeautifulSoup

- Clean and analyse the data and find how many unique boroughs there are in London and what are their coordinates (latitude and longitude)
- Create London map and show all the boroughs on the map from geopy.geocoders import Nominatim
- Create a function to explore all boroughs
- Get top 50 venues in 500m radius of the centre of each Borough
- Use One hot encoding before clustering
- Find top 10 venues for each neighbourhood and create pandas dataframe.
- Conduct K-means clustering to group the boroughs according to what convenience facilities they have using Foursquare data.
- Add clustering labels
- Merge london_grouped with london_data to add latitude/longitude for each neighbourhood
- Create a map showing all the clusters.
- Analyse each cluster individually and find 3 most suitable clusters for opening an Italian restaurant and I will create a map showing these 3 clusters.

In [3]:
```
# Extracting 3 columns from Wikipedia
BoroughName = []
Population = []
Coordinates = []

for row in soup.find('table').find_all('tr'):
    cells = row.find_all('td')
    if len(cells) > 0:
        BoroughName.append(cells[0].text.rstrip('\n'))
        Population.append(cells[7].text.rstrip('\n'))
        Coordinates.append(cells[8].text.rstrip('\n'))
```

In [4]:
```
# Creating a dataframe
dict = {'BoroughName' : BoroughName,
       'Population' : Population,
       'Coordinates': Coordinates}
info = pd.DataFrame.from_dict(dict)
info.head()
```

Out[4]:

| | BoroughName | Population | Coordinates |
|---|---|---|---|
| 0 | Barking and Dagenham [note 1] | 194,352 | 51°33′39″N 0°09′21″E / 51.5607°N 0.1557°E /... |
| 1 | Barnet | 369,088 | 51°37′31″N 0°09′06″W / 51.6252°N 0.1517°W /... |
| 2 | Bexley | 236,687 | 51°27′18″N 0°09′02″E / 51.4549°N 0.1505°E /... |
| 3 | Brent | 317,264 | 51°33′32″N 0°16′54″W / 51.5588°N 0.2817°W /... |
| 4 | Bromley | 317,899 | 51°24′14″N 0°01′11″E / 51.4039°N 0.0198°E /... |

In [7]:
```
# Spliting Latitude and Longitude in to separate columns
info.drop(labels=['Coordinates','Coordinates1','Coordinates2'], axis=1,inplace = True)
info[['Latitude','Longitude']] = info['Coordinates3'].str.split(';',expand=True)
info.head()
```

Out[7]:

| | BoroughName | Population | Coordinates3 | Latitude | Longitude |
|---|---|---|---|---|---|
| 0 | Barking and Dagenham | 194,352 | 51.5607; 0.1557 (Barking and Dagenham) | 51.5607 | 0.1557 (Barking and Dagenham) |
| 1 | Barnet | 369,088 | 51.6252; -0.1517 (Barnet) | 51.6252 | -0.1517 (Barnet) |
| 2 | Bexley | 236,687 | 51.4549; 0.1505 (Bexley) | 51.4549 | 0.1505 (Bexley) |
| 3 | Brent | 317,264 | 51.5588; -0.2817 (Brent) | 51.5588 | -0.2817 (Brent) |
| 4 | Bromley | 317,899 | 51.4039; 0.0198 (Bromley) | 51.4039 | 0.0198 (Bromley) |

In [8]:
```
info.drop(labels=['Coordinates3'], axis=1,inplace = True)
info['Latitude'] = info['Latitude'].map(lambda x: x.rstrip(u'\ufeff'))
info['Latitude'] = info['Latitude'].map(lambda x: x.lstrip())
info['Longitude'] = info['Longitude'].map(lambda x: x.rstrip(')'))
info['Longitude'] = info['Longitude'].map(lambda x: x.rstrip('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQR
STUVWXYZ '))
info['Longitude'] = info['Longitude'].map(lambda x: x.rstrip(' ('))
info['Longitude'] = info['Longitude'].map(lambda x: x.rstrip(u'\ufeff'))
info['Longitude'] = info['Longitude'].map(lambda x: x.lstrip())
info['Population'] = info['Population'].str.replace(',','')
info.head()
```

Out[8]:

| | BoroughName | Population | Latitude | Longitude |
|---|---|---|---|---|
| 0 | Barking and Dagenham | 194352 | 51.5607 | 0.1557 |
| 1 | Barnet | 369088 | 51.6252 | -0.1517 |
| 2 | Bexley | 236687 | 51.4549 | 0.1505 |
| 3 | Brent | 317264 | 51.5588 | -0.2817 |
| 4 | Bromley | 317899 | 51.4039 | 0.0198 |

In [9]:
```
# Finding the unique Boroughs
info['BoroughName'].unique()
```

Out[9]:
```
array(['Barking and Dagenham', 'Barnet', 'Bexley', 'Brent', 'Bromley',
       'Camden', 'Croydon', 'Ealing', 'Enfield', 'Greenwich', 'Hackney',
       'Hammersmith and Fulham', 'Haringey', 'Harrow', 'Havering',
       'Hillingdon', 'Hounslow', 'Islington', 'Kensington and Chelsea',
       'Kingston upon Thames', 'Lambeth', 'Lewisham', 'Merton', 'Newham',
       'Redbridge', 'Richmond upon Thames', 'Southwark', 'Sutton',
       'Tower Hamlets', 'Waltham Forest', 'Wandsworth', 'Westminster'],
      dtype=object)
```

In [10]:
```
info.shape
```

Out[10]: (32, 4)

We found that London has 32 boroughs.

We got geograpical coordinates of London:

In [11]:
```
from geopy.geocoders import Nominatim
```

In [12]:
```
# Getting geograpical coordinates of London
address = 'London, United Kingdom'

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

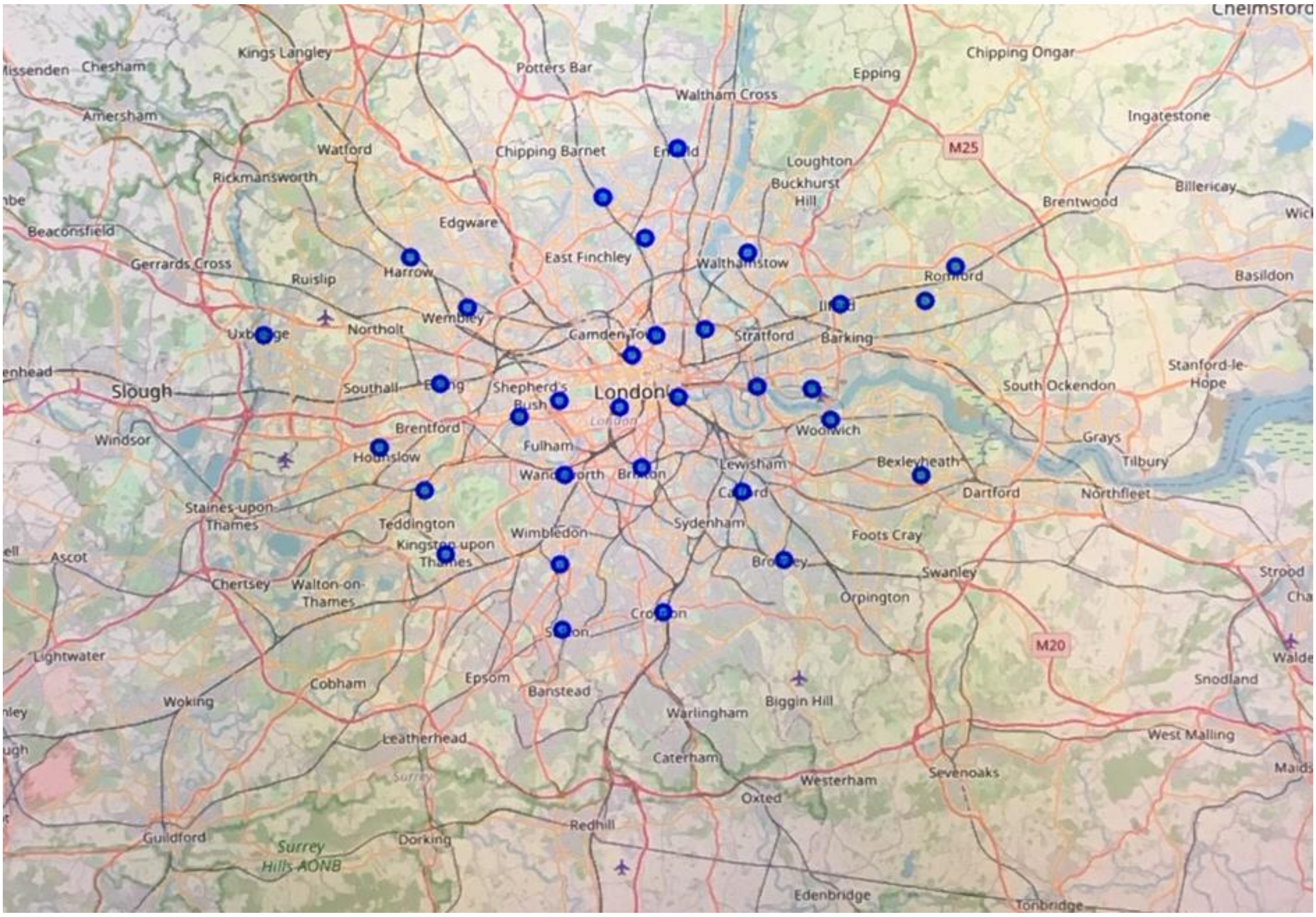
```
The geograpical coordinate of London are 51.5073219, -0.1276474.
```

We created a map of London showing all the London boroughs:

In [24]:
```python
# Creating a map of London and showing all the London boroughs on the map
map = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough in zip(info['Latitude'], info['Longitude'], info['BoroughName']):
    label = folium.Popup(borough, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map)

map
```



**PART 3 - Methodology**

In this section we will explore the data using visualization and we will conduct cluster analysis and identify 3 boroughs most suitable for opening an Italian restaurant.

In [25]:
```
# Foursquare credentials
CLIENT_ID = 'OM3MNXPFT2YBEIGZ4F0AA22HGIBQQTBBRWYHUSSQXPEUE03I'
CLIENT_SECRET = 'GIQ1VHUDT4MIJD1FZZXSSRPTMK0QJ1YFUOWXI4RDYKE1GCZS'
VERSION = '20180605'

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

```
Your credentails:
CLIENT_ID: OM3MNXPFT2YBEIGZ4F0AA22HGIBQQTBBRWYHUSSQXPEUE03I
CLIENT_SECRET:GIQ1VHUDT4MIJD1FZZXSSRPTMK0QJ1YFUOWXI4RDYKE1GCZS
```

We created a function for exploring all London boroughs:

In [26]:
```
# Creating a function for exploring all London boroughs
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # creating the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&r
adius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # making the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # returning 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 = ['BoroughName',
                  'Borough Latitude',
                  'Borough Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [27]:
```
# Getting top 50 venues in 500m radius of the center of each Borough
LIMIT = 50
venues = getNearbyVenues(names=info['BoroughName'],
                                   latitudes=info['Latitude'],
                                   longitudes=info['Longitude']
                                  )
```

```
Barking and Dagenham
Barnet
Bexley
Brent
Bromley
Camden
Croydon
Ealing
Enfield
Greenwich
Hackney
Hammersmith and Fulham
Haringey
Harrow
Havering
Hillingdon
Hounslow
Islington
Kensington and Chelsea
Kingston upon Thames
Lambeth
Lewisham
Merton
Newham
Redbridge
Richmond upon Thames
Southwark
Sutton
Tower Hamlets
Waltham Forest
Wandsworth
Westminster
```

In [28]:
```
print(venues.shape)
venues.head()
```

```
(1138, 7)
```

Out[28]:

| | BoroughName | Borough Latitude | Borough Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Barking and Dagenham | 51.5607 | 0.1557 | Central Park | 51.559560 | 0.161981 | Park |
| 1 | Barking and Dagenham | 51.5607 | 0.1557 | Crowlands Heath Golf Course | 51.562457 | 0.155818 | Golf Course |
| 2 | Barking and Dagenham | 51.5607 | 0.1557 | Robert Clack Leisure Centre | 51.560808 | 0.152704 | Martial Arts Dojo |
| 3 | Barking and Dagenham | 51.5607 | 0.1557 | Morrisons | 51.559774 | 0.148752 | Supermarket |
| 4 | Barking and Dagenham | 51.5607 | 0.1557 | Beacontree Heath Leisure Centre | 51.560997 | 0.148932 | Gym / Fitness Center |

In [29]:
```
venues.groupby('BoroughName').count()
```

Out[29]:

| | Borough Latitude | Borough Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|
| BoroughName | | | | | | |
| Barking and Dagenham | 7 | 7 | 7 | 7 | 7 | 7 |
| Barnet | 4 | 4 | 4 | 4 | 4 | 4 |
| Bexley | 30 | 30 | 30 | 30 | 30 | 30 |
| Brent | 50 | 50 | 50 | 50 | 50 | 50 |
| Bromley | 39 | 39 | 39 | 39 | 39 | 39 |
| Camden | 50 | 50 | 50 | 50 | 50 | 50 |

We found that there are 187 unique categories:

In [30]:
```
print('There are {} uniques categories.'.format(len(venues['Venue Category'].unique())))
```

```
There are 187 uniques categories.
```

We created One Hot encoding before clustering:

In [32]:
```
# Creating One Hot encoding before clustering
kut_onehot = pd.get_dummies(venues[['Venue Category']], prefix="", prefix_sep="")

# Adding BoroughName column back to dataframe
kut_onehot['BoroughName'] = venues['BoroughName']

# Moving BoroughName column to the first column
fixed_columns = [kut_onehot.columns[-1]] + list(kut_onehot.columns[:-1])
kut_onehot = kut_onehot[fixed_columns]

kut_onehot.head()
```

Out[32]:

| | BoroughName | African Restaurant | Airport | Airport Lounge | Airport Service | American Restaurant | Argentinian Restaurant | Art Gallery | Art Museum | Arts & Crafts Store | ... | Used Bookstore |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Barking and Dagenham | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 |
| 1 | Barking and Dagenham | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 |
| 2 | Barking and Dagenham | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 |
| 3 | Barking and Dagenham | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 |
| 4 | Barking and Dagenham | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 |

5 rows × 188 columns

We grouped rows by borough and taking the mean of frequency of each venue category:

In [33]:
```python
# Grouping rows by borough and taking the mean of frequency of each venue category
kut_grouped = kut_onehot.groupby('BoroughName').mean().reset_index()
kut_grouped.head()
```

Out[33]:

| | BoroughName | African Restaurant | Airport | Airport Lounge | Airport Service | American Restaurant | Argentinian Restaurant | Art Gallery | Art Museum | Arts & Crafts Store | ... | Used Bookstore |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Barking and Dagenham | 0.0 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 |
| 1 | Barnet | 0.0 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 |
| 2 | Bexley | 0.0 | 0.0 | 0.0 | 0.0 | 0.033333 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 |
| 3 | Brent | 0.0 | 0.0 | 0.0 | 0.0 | 0.040000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 |
| 4 | Bromley | 0.0 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 |

5 rows × 188 columns

In [34]:
```python
kut_grouped.shape
```

Out[34]: (32, 188)

We got top 10 venues for each borough:

In [35]:
```python
# Getting top 10 venues for each neighborhood
num_top_venues = 10

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

```
----Barking and Dagenham----
                   venue  freq
0    Gym / Fitness Center  0.14
1                    Pool  0.14
2             Bus Station  0.14
3             Supermarket  0.14
4             Golf Course  0.14
5       Martial Arts Dojo  0.14
6                    Park  0.14
7      African Restaurant  0.00
8  Okonomiyaki Restaurant  0.00
9            Optical Shop  0.00
```

We created a dataframe with top 10 venues for each borough:

In [36]:
```python
# Creating a dataframe with top 10 venues for each borough
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]
```

In [39]:
```python
num_top_venues = 10
indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['BoroughName']
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_sort = pd.DataFrame(columns=columns)
neighborhoods_venues_sort['BoroughName'] = kut_grouped['BoroughName']

for ind in np.arange(kut_grouped.shape[0]):
    neighborhoods_venues_sort.iloc[ind, 1:] = return_most_common_venues(kut_grouped.iloc[ind, :], num_top_
venues)

neighborhoods_venues_sort
```

Out[39]:

| | BoroughName | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Barking and Dagenham | Golf Course | Pool | Bus Station | Supermarket | Park | Gym / Fitness Center | Martial Arts Dojo | Yoga Studio |
| 1 | Barnet | Bus Stop | Business Service | Café | Salon / Barbershop | Yoga Studio | English Restaurant | Fish Market | Fish & Chips Shop |
| 2 | Bexley | Clothing Store | Coffee Shop | Pub | Italian Restaurant | Furniture / Home Store | Supermarket | Fast Food Restaurant | Pharmacy |
| 3 | Brent | Coffee Shop | Hotel | Clothing Store | Grocery Store | Bar | Sporting Goods Shop | American Restaurant | Sandwich Place |
| 4 | Bromley | Clothing Store | Coffee Shop | Pizza Place | Burger Joint | Bar | Bookstore | Café | Sandwich Place |

We clustered all boroughs in to 5 clusters and added clustering labels:

In [40]:
```
# Importing k-means for clustering
from sklearn.cluster import KMeans

# Setting the number of clusters to 5
kclusters = 5

kut_grouped_clustering = kut_grouped.drop('BoroughName', 1)

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

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

Out[40]:
```
array([2, 3, 2, 1, 2, 1, 1, 1, 2, 2], dtype=int32)
```

In [41]:
```
# Adding clustering labels
neighborhoods_venues_sort.insert(0, 'Cluster Labels', kmeans.labels_)

kut_merged = info

# Merging London_grouped with London_data to add latitude/longitude for each neighborhood
kut_merged = kut_merged.join(neighborhoods_venues_sort.set_index('BoroughName'), on='BoroughName')

kut_merged.head()
```

Out[41]:

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Mos Commo Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Barking and Dagenham | 194352 | 51.5607 | 0.1557 | 2 | Golf Course | Pool | Bus Station | Supermarket | Park | Gym / Fitness Center |
| 1 | Barnet | 369088 | 51.6252 | -0.1517 | 3 | Bus Stop | Business Service | Café | Salon / Barbershop | Yoga Studio | English Restaur |
| 2 | Bexley | 236687 | 51.4549 | 0.1505 | 2 | Clothing Store | Coffee Shop | Pub | Italian Restaurant | Furniture / Home Store | Superm |
| 3 | Brent | 317264 | 51.5588 | -0.2817 | 1 | Coffee Shop | Hotel | Clothing Store | Grocery Store | Bar | Sportin Goods |
| 4 | Bromley | 317899 | 51.4039 | 0.0198 | 2 | Clothing Store | Coffee Shop | Pizza Place | Burger Joint | Bar | Booksto |

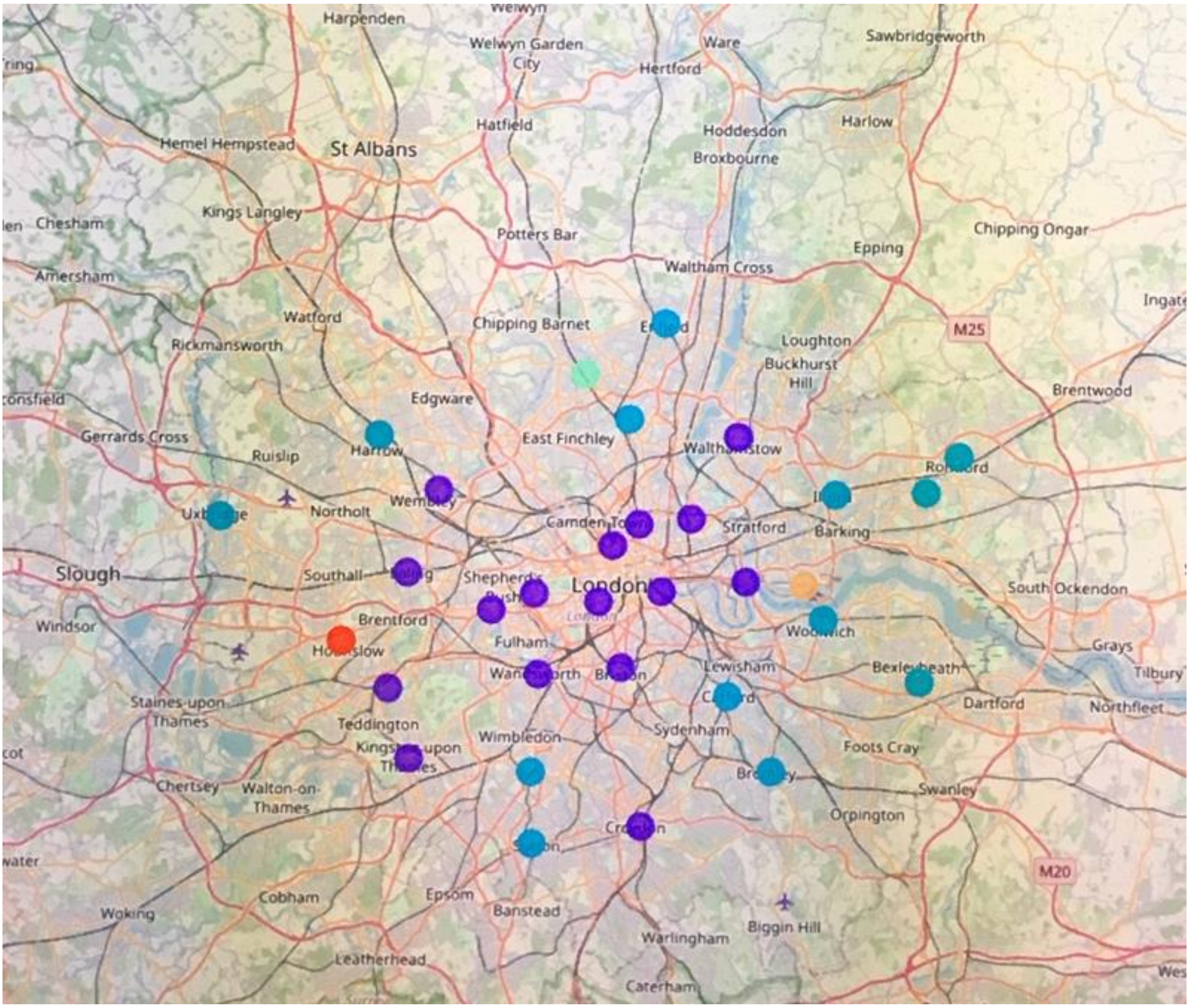
We created a map showing all the clusters:

```
In [43]: # Creating a map showing all the clusters
import matplotlib.cm as cm
import matplotlib.colors as colors
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=10)

# Setting the 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]

# Adding markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(kut_merged['Latitude'], kut_merged['Longitude'], kut_merged['BoroughNam
e'], kut_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=8,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```



We analysed each cluster:

In [44]:
```
# Cluster:0
kut_merged[kut_merged['Cluster Labels'] == 0]
```

Out[44]:

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Mos Commo Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | Hounslow | 262407 | 51.4746 | -0.368 | 0 | Café | Chinese Restaurant | Park | Bed & Breakfast | Yoga Studio | English Restaura |

In [45]:
```
# Cluster:1
kut_merged[kut_merged['Cluster Labels'] == 1]
```

Out[45]:

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | Brent | 317264 | 51.5588 | -0.2817 | 1 | Coffee Shop | Hotel | Clothing Store | Grocery Store | Bar |
| 5 | Camden | 229719 | 51.5290 | -0.1255 | 1 | Coffee Shop | Café | Burger Joint | Pub | Train Station |
| 6 | Croydon | 372752 | 51.3714 | -0.0977 | 1 | Pub | Coffee Shop | Portuguese Restaurant | Supermarket | Spanish Restaurant |
| 7 | Ealing | 342494 | 51.5130 | -0.3089 | 1 | Coffee Shop | Clothing Store | Park | Vietnamese Restaurant | Burger Joint |
| 10 | Hackney | 257379 | 51.5450 | -0.0553 | 1 | Pub | Coffee Shop | Café | Bakery | Clothing Store |
| 11 | Hammersmith and Fulham | 178685 | 51.4927 | -0.2339 | 1 | Pub | Italian Restaurant | Indian Restaurant | Café | Coffee Shop |
| 17 | Islington | 215667 | 51.5416 | -0.1022 | 1 | Pub | Burger Joint | Park | Ice Cream Shop | Cocktail Bar |
| 18 | Kensington and Chelsea | 155594 | 51.5020 | -0.1947 | 1 | Clothing Store | Café | Juice Bar | Bakery | Italian Restaurant |
| 19 | Kingston upon Thames | 166793 | 51.4085 | -0.3064 | 1 | Café | Coffee Shop | Italian Restaurant | Pub | Burger Joint |
| 20 | Lambeth | 314242 | 51.4607 | -0.1163 | 1 | Caribbean Restaurant | Market | Indian Restaurant | BBQ Joint | Beer Bar |
| 25 | Richmond upon Thames | 191365 | 51.4479 | -0.3260 | 1 | Pub | Coffee Shop | Italian Restaurant | Café | Indian Restaurant |
| 26 | Southwark | 298464 | 51.5035 | -0.0804 | 1 | Coffee Shop | Hotel | Bar | Theater | Hotel Bar |
| 28 | Tower Hamlets | 272890 | 51.5099 | -0.0059 | 1 | Light Rail Station | Hotel | Coffee Shop | Italian Restaurant | Convenience Store |
| 29 | Waltham Forest | 265797 | 51.5908 | -0.0134 | 1 | Pub | Concert Hall | Gym / Fitness Center | Pool | Coffee Shop |
| 30 | Wandsworth | 310516 | 51.4567 | -0.1910 | 1 | Coffee Shop | Pub | Clothing Store | Breakfast Spot | Supermarket |
| 31 | Westminster | 226841 | 51.4973 | -0.1372 | 1 | Coffee Shop | Hotel | Theater | Sushi Restaurant | Sporting Goods Shop |

In [46]:
```
# Cluster:2
kut_merged[kut_merged['Cluster Labels'] == 2]
```
Out[46]:

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Barking and Dagenham | 194352 | 51.5607 | 0.1557 | 2 | Golf Course | Pool | Bus Station | Supermarket | Park |
| 2 | Bexley | 236687 | 51.4549 | 0.1505 | 2 | Clothing Store | Coffee Shop | Pub | Italian Restaurant | Furniture / Home Sto |
| 4 | Bromley | 317899 | 51.4039 | 0.0198 | 2 | Clothing Store | Coffee Shop | Pizza Place | Burger Joint | Bar |
| 8 | Enfield | 320524 | 51.6538 | -0.0799 | 2 | Clothing Store | Coffee Shop | Pub | Department Store | Supermar |
| 9 | Greenwich | 264008 | 51.4892 | 0.0648 | 2 | Pub | Clothing Store | Fast Food Restaurant | Supermarket | Coffee Sh |
| 12 | Haringey | 263386 | 51.6000 | -0.1119 | 2 | Fast Food Restaurant | Café | Grocery Store | Supermarket | Mediterrai Restaurar |
| 13 | Harrow | 243372 | 51.5898 | -0.3346 | 2 | Indian Restaurant | Grocery Store | Coffee Shop | Fast Food Restaurant | Supermar |
| 14 | Havering | 242080 | 51.5812 | 0.1837 | 2 | Coffee Shop | Clothing Store | Fast Food Restaurant | Shopping Mall | Bakery |
| 15 | Hillingdon | 286806 | 51.5441 | -0.4760 | 2 | Coffee Shop | Italian Restaurant | Clothing Store | Pharmacy | Burger Jo |
| 21 | Lewisham | 286180 | 51.4452 | -0.0209 | 2 | Supermarket | Grocery Store | Coffee Shop | Platform | Italian Restaurar |
| 22 | Merton | 203223 | 51.4014 | -0.1958 | 2 | Park | Italian Restaurant | Café | Supermarket | Indian Restaurar |
| 24 | Redbridge | 288272 | 51.5590 | 0.0741 | 2 | Clothing Store | Supermarket | Fast Food Restaurant | Sandwich Place | Bakery |
| 27 | Sutton | 195914 | 51.3618 | -0.1945 | 2 | Pub | Clothing Store | Coffee Shop | Italian Restaurant | Departme Store |

```
In [47]: # Cluster:3
kut_merged[kut_merged['Cluster Labels'] == 3]
```

Out[47]:

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Barnet | 369088 | 51.6252 | -0.1517 | 3 | Bus Stop | Business Service | Café | Salon / Barbershop | Yoga Studio |

```
In [48]: # Cluster:4
kut_merged[kut_merged['Cluster Labels'] == 4]
```

Out[48]:

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|
| 23 | Newham | 318227 | 51.5077 | 0.0469 | 4 | Hotel | Airport Service | Light Rail Station | Chinese Restaurant | Rafting |

**PART 4 – Results & Recommendation**

After analysing each cluster individually we got the following results:

**CLUSTER 0:** - It is a good place for opening an Italian Restaurant because this Cluster has no Italian Restaurants but it has cafe, Chinese restaurant, English restaurant and fish & chips shop as most common venues.

**CLUSTER 1:** - Not recommended because of competition - it has many Italian Restaurants, coffee shops and fast food restaurants.

**CLUSTER 2:** - Not recommended because of competition - it has many Italian Restaurants.

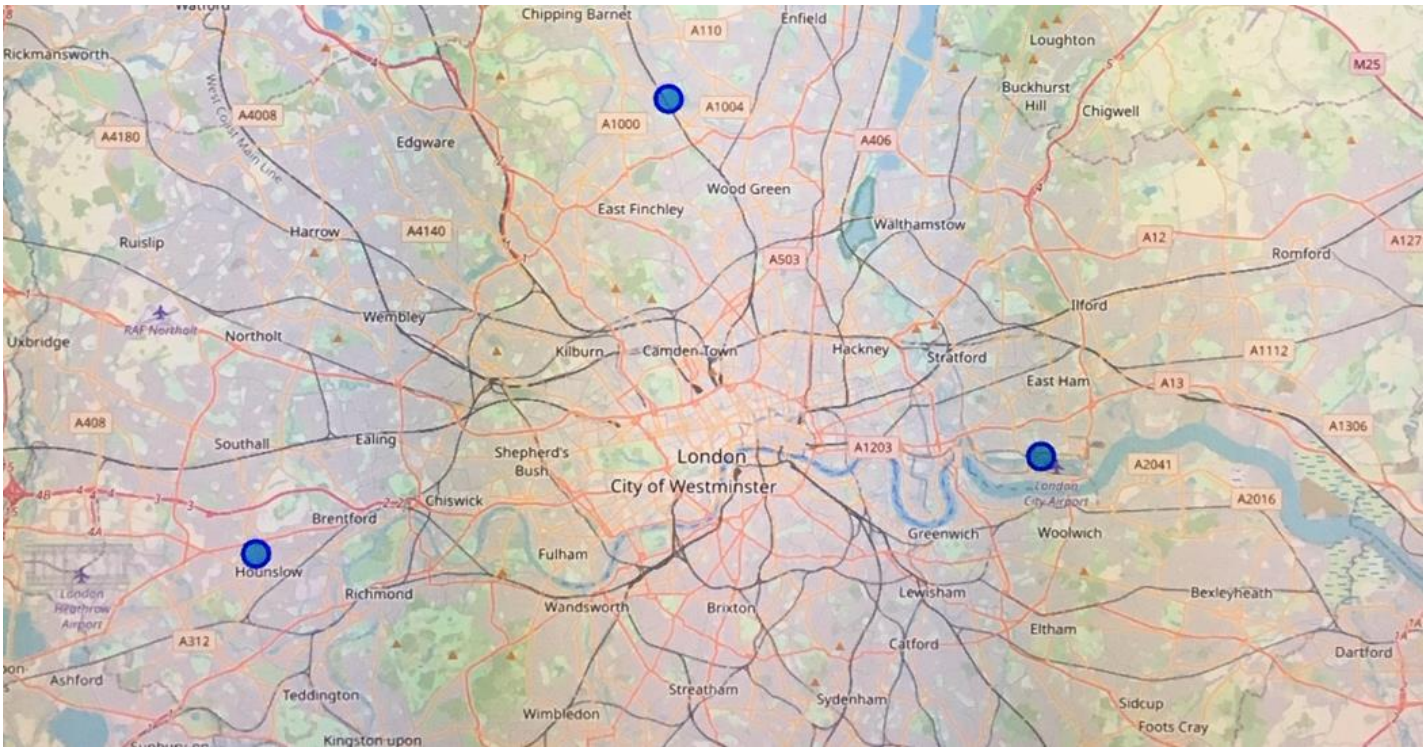
**CLUSTER 3:** - It is a good place for opening an Italian Restaurant because this Cluster has not Italian Restaurants, but it has high population of 369088 and it has business service and cafe and English restaurant and fast food restaurant.

**CLUSTER 4:** - Newham is a very good place for opening an Italian Restaurant because it has an airport and a rail station and no Italian Restaurants as most common venues. Also this borough has high population of 318227.

**PART 5 – Conclusion**

From our analysis, we have found that Barnet, Hounslow and Newham are the best 3 boroughs in London for opening an Italian Restaurant, based on the availability of Italian Restaurants and other type of restaurants available in the neighborhoods and high population.

| | BoroughName | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Mo Comn Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Barnet | 369088 | 51.6252 | -0.1517 | 3 | Bus Stop | Business Service | Café | Salon / Barbershop | Yoga Studio | Englis Restau |
| 16 | Hounslow | 262407 | 51.4746 | -0.3680 | 0 | Café | Chinese Restaurant | Park | Bed & Breakfast | Yoga Studio | Englis Restau |
| 23 | Newham | 318227 | 51.5077 | 0.0469 | 4 | Hotel | Airport Service | Light Rail Station | Chinese Restaurant | Rafting | Pharm |




*This project would benefit from a further research of these 3 chosen boroughs. We could find out and compare the commercial rent prices in these boroughs and average income. We could also research the best streets and buildings with would be most suitable for an Italian restaurant. Also it would be beneficial to have a food market or a supermarket nearby for buying fresh produce.*