Capstone Project – The Battle of Neighbourhoods in London

1. Introduction

1.1 Problem Statement

This research aims to analysis the popularity and geolocation information of restaurant in London, United Kingdom to facilitate the decision of new store location and category.

1.2 Background

In 2020, tremendous restaurants in London closed due to COVID-19 since it was discovered in China early January. It is believe the economy will be recovered in 2021 alongside with mass production of vaccine. Thus, it is a perfect opportunity to conduct a market research on existing restaurant in London although it is not the best time to start a new one.

So, how could we leverage Foursquare location data and machine learning to help us make decision and find appropriate neighbourhoods? This is the problem I would like to address in this capstone project taking Tokyo as an example. In this project, I am going to use Foursquare location data and clustering methods to group the districts to different group by their restaurant venues information.

2. Data Requirement

For this project we need following data:

- Tokyo data that contains list districts (Wards) along with their latitude and longitude.

  Datasource: https://en.wikipedia.org/wiki/London_boroughs#Former_authoritie

  Description: We will Scrap London Borough Table from Wikipedia and get the coordinates of these Borough using geocoder class of Geopy client.

- Restaurants in each neighbourhood of London:

  Data source: Foursquare APIs

  Description: By using this API we will get all the venues in each neighbourhood. We can filter these venues to get only restaurants.

3. Methodology

3.1 Data Preparation

3.1.1 Scraping London Borough information from Wikipedia

First thing first, let's retrieve the information of Borough in London from wiki and create a data-frame directly with pandas' read_html function to transfer the data in the table from Wikipedia into a data-frame containing borough name, designation and other info.

In [3]:
```
df = pd.read_html('https://en.wikipedia.org/wiki/London_boroughs#Former_authorities')[2]
df
```

Out[3]:

	London borough	Designation	Former areas	Former areas.1	Former areas.2	Former areas.3	Former areas.4
0	Camden	Inner	Hampstead (11a)	St Pancras (11b)	Holborn (11c)	NaN	NaN
1	Greenwich	Inner	Greenwich (22a)	Woolwich (part) (22b)	NaN	NaN	NaN
2	Hackney	Inner	Hackney (9a)	Shoreditch (9b)	Stoke Newington (9c)	NaN	NaN
3	Hammersmith[notes 2]	Inner	Hammersmith (4a)	Fulham (4b)	NaN	NaN	NaN
4	Islington	Inner	Islington (10a)	Finsbury (10b)	NaN	NaN	NaN
5	Kensington and Chelsea	Inner	Kensington (3a)	Chelsea (3b)	NaN	NaN	NaN
6	Lambeth	Inner	Lambeth (6a)	Wandsworth (part) (6b)	NaN	NaN	NaN
7	Lewisham	Inner	Lewisham (21a)	Deptford (21b)	NaN	NaN	NaN
8	Southwark	Inner	Bermondsey (7b)	Camberwell (7c)	Southwark (7a)	NaN	NaN
9	Tower Hamlets	Inner	Bethnal Green (8a)	Poplar (8c)	Stepney (8b)	NaN	NaN
10	Wandsworth	Inner	Battersea (5b)	Wandsworth (part) (5a)	NaN	NaN	NaN
11	Westminster	Inner	Paddington (2c)	St Marylebone (2b)	Westminster (2a)	NaN	NaN
12	Barking[notes 3]	Outer	Barking (part) (25a)	Dagenham (part) (25b)	NaN	NaN	NaN
13	Barnet	Outer	Barnet (31a)	East Barnet (31b)	Finchley (31d)	Hendon (31c)	Friern Barnet (31e)
14	Bexley	Outer	Bexley (23b)	Erith (23a)	Crayford (23c)	Chislehurst and Sidcup (part) (23d)	NaN
15	Brent	Outer	Wembley (12a)	Willesden (12b)	NaN	NaN	NaN
16	Bromley	Outer	Bromley (20c)	Beckenham (20b)	Orpington (20e)	Penge (20a)	Chislehurst and Sidcup (part) (20d)
17	Croydon	Outer	Croydon (19a)	Coulsdon and Purley (19b)	NaN	NaN	NaN

After manipulating the data, the data-frame is updated as below,

Out[4]:

	Borough	Designation
0	Camden, London	Inner
1	Greenwich, London	Inner
2	Hackney, London	Inner
3	Hammersmith, London	Inner
4	Islington, London	Inner
5	Kensington and Chelsea, London	Inner
6	Lambeth, London	Inner
7	Lewisham, London	Inner
8	Southwark, London	Inner
9	Tower Hamlets, London	Inner
10	Wandsworth, London	Inner
11	Westminster, London	Inner
12	Barking, London	Outer
13	Barnet, London	Outer
14	Bexley, London	Outer
15	Brent, London	Outer
16	Bromley, London	Outer
17	Croydon, London	Outer
18	Ealing, London	Outer
19	Enfield, London	Outer
20	Haringey, London	Outer

3.1.2 Retrieving Coordinates of London Borough

With the name of 31 Boroughs ready, we are going to obtain the coordinate information with gorcoder class of Geopy client,

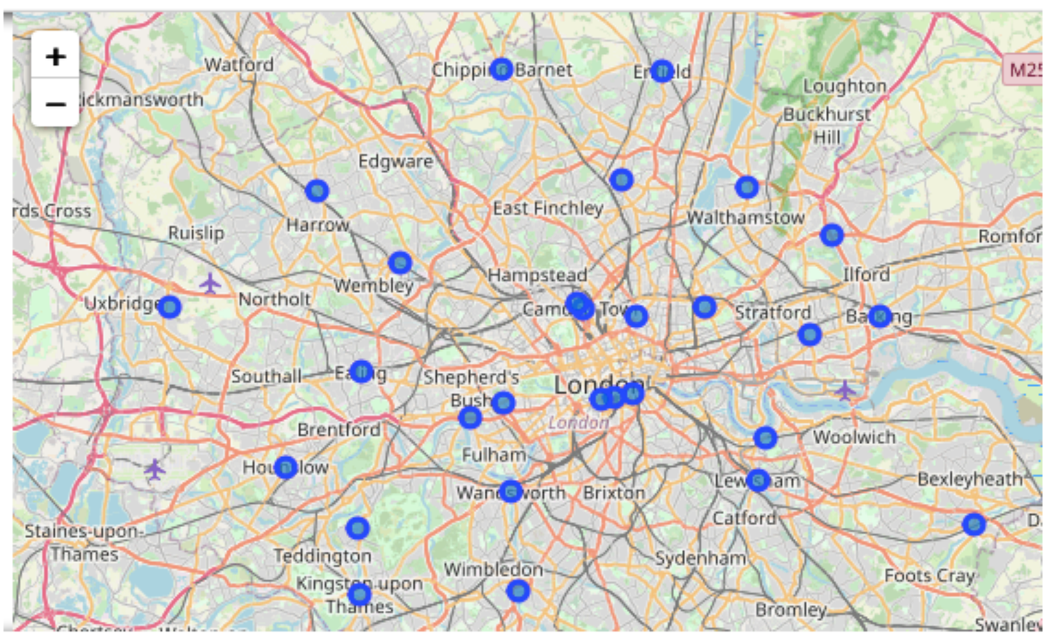
Retrieve Geospatial Data

```
]: geolocator = Nominatim(user_agent="London_Analysis")

df['location_details'] = df['Borough'].apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude))
df[['Latitude', 'Longitude']] = df['location_details'].apply(pd.Series)
df.drop(['location_details'], axis=1, inplace = True)
df['Borough'] = df['Borough'].str.replace(', London', '')

df
```

:[6]:

	Borough	Designation	Latitude	Longitude
0	Camden	Inner	51.542305	-0.139560
1	Greenwich	Inner	51.482084	-0.004542
2	Hackney	Inner	51.543240	-0.049362
3	Hammersmith	Inner	51.492038	-0.223640
4	Islington	Inner	51.538429	-0.099905
5	Kensington and Chelsea	Inner	51.498480	-0.199043
6	Lambeth	Inner	51.501301	-0.117287
7	Lewisham	Inner	51.462432	-0.010133
8	Southwark	Inner	51.502922	-0.103458
9	Tower Hamlets	Inner	51.525629	-0.033585
10	Wandsworth	Inner	51.457027	-0.193261
11	Westminster	Inner	51.500444	-0.126540
12	Barking	Outer	51.538992	0.080424

With coordinates of each borough, we can visualise their location in a map with latitude, longitude and folium library



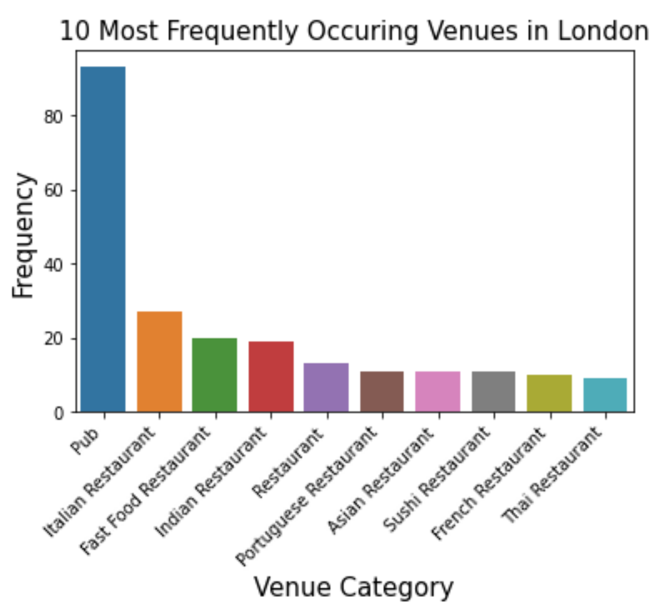
3.2 Exploratory Data Analysis

3.2.1 Applying Foursquare Location Data

Let’s make use of Foursquare API and retrieve the top 100 venue in every Borough and filter out the non-restaurant venue. Here we categorised “Pub” as one of the restaurant category as Pub in Britain sell food and beer.

```
London_restaurant = London_venues[London_venues['Venue Category'].str.contains('Restaurant')].reset_index(drop=True)
London_restaurant = London_restaurant.append(London_venues[London_venues['Venue Category'] == "Pub"].reset_index(drop=True))
London_restaurant.index = np.arange(1, len(London_restaurant )+1)

print (London_restaurant['Venue Category'].value_counts())
```

```
Pub                                93
Italian Restaurant                 27
Fast Food Restaurant               20
Indian Restaurant                  19
Restaurant                         13
Portuguese Restaurant              11
Asian Restaurant                   11
Sushi Restaurant                   11
French Restaurant                  10
Thai Restaurant                     9
Japanese Restaurant                 7
Mediterranean Restaurant            7
Turkish Restaurant                  7
Vietnamese Restaurant               7
Chinese Restaurant                  6
Vegetarian / Vegan Restaurant       6
English Restaurant                  6
Ramen Restaurant                    5
Korean Restaurant                   4
Modern European Restaurant          4
Caribbean Restaurant                4
Mexican Restaurant                  4
German Restaurant                   3
Spanish Restaurant                  3
Latin American Restaurant           3
Kebab Restaurant                    3
Greek Restaurant                    2
Argentinian Restaurant              2
```



3.3 Apply k-Mean clustering to Data

After preparing the restaurant data in 31 borough, we analyse the top 10 restaurant category for each borough.

1. Create an one hot encoding data-frame for restaurant

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Argentinian Restaurant	Asian Restaurant	Austrian Restaurant	Brazilian Restaurant	Cajun / Creole Restaurant	Caribbean Restaurant	...	Seafood Restaurant	South American Restaurant	Southern / Soul Food Restaurant	Spanish Restaurant	Sushi Restaurant	Tap Restaura
1	Camden	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	Camden	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
3	Camden	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	Camden	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	
5	Camden	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

2. Calculate the mean of occurrence of each restaurant category for each neighbourhood

3. Merge the restaurant summary of each borough with their location information

ighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Argentinian Restaurant	Asian Restaurant	Austrian Restaurant	Brazilian Restaurant	Cajun / Creole Restaurant	Caribbean Restaurant	...	Seafood Restaurant	South American Restaurant	Southern / Soul Food Restaurant	Spanish Restaurant	Sushi Restaurant	Tapas Restaurant
Barking	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Barnet	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Bexley	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Brent	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Bromley	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Camden	0.00000	0.037037	0.000000	0.000000	0.037037	0.00000	0.000000	0.00000	0.074074	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Croydon	0.00000	0.000000	0.000000	0.000000	0.071429	0.00000	0.000000	0.00000	0.071429	...	0.000000	0.00000	0.000000	0.071429	0.071429	0.000000
Ealing	0.00000	0.000000	0.033333	0.000000	0.033333	0.00000	0.000000	0.00000	0.033333	...	0.000000	0.00000	0.033333	0.033333	0.033333	0.000000
Enfield	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Greenwich	0.00000	0.000000	0.000000	0.076923	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.076923	0.000000
Hackney	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
ammersmith	0.00000	0.000000	0.000000	0.000000	0.043478	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.043478	0.043478
Haringey	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Harrow	0.50000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Havering	0.00000	0.000000	0.000000	0.000000	0.083333	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Hillingdon	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
Hounslow	0.00000	0.000000	0.000000	0.000000	0.083333	0.00000	0.000000	0.00000	0.000000	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000

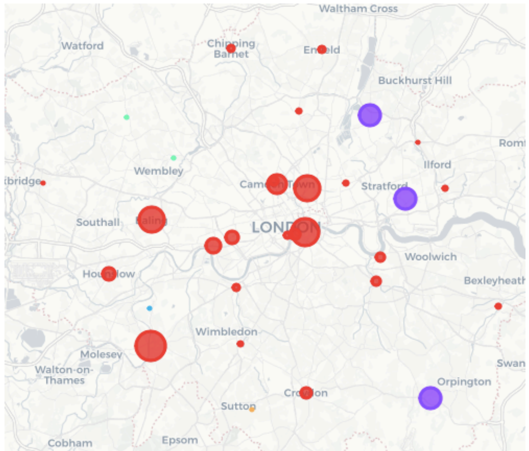
With the summary of restaurant category, we can apply k-Means clustering to cluster these 31 borough base on the restaurant categories, base on the similarities of venue categories.

```
# set number of clusters
kclusters = 5

London_grouped_clustering = London_grouped.drop('Neighborhood', 1)

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

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

```
]: array([0, 0, 0, 3, 1, 0, 0, 0, 0, 0], dtype=int32)
```

```
# add clustering labels

neighbuorhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

London_merged = df

London_merged.rename(columns={'Borough':'Neighborhood'}, inplace=True)

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
London_merged = London_merged.join(neighbuorhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

London_merged.head(23) # check the last columns!
```

We will also plot the clustered borough onto a map with folium library.



4. Discussion

Let's summarise our finding from the result of clustering to support the business decision on opening a new restaurant.

- Pub is the most common and popular category in London with 97 occurrence
- Kingston upon Thames, Southwark and Ealing has the most restaurant
- Bromley, Newham and Waltham Forest has the fewest restaurant
- Brent, Haringey, Harrow, Hounslow and Lambeth has no Pub within the borough
- 24 Borough have fallen into the same cluster with Pub as a popular category.

From the result of clustering, we can assume the customers in the borough within the same cluster expected or prefer a similar restaurant category. While we can observe that in cluster 1, Pub is not a top ten category in Lambeth, Hounslow and Haringey. Therefore we can assume if a Pub is opened in these borough, they will be as popular as other borough. In addition, Lambeth is the only inner designation borough that lacks of Pub. We can assume starting a Pub business in Lambeth will be relatively good, given that there is no competitor while providing popular restaurant choice to the resident and tourist.

However, we have only taken the general number of restaurant of each borough in to account while ignoring the actual location, marketing and promotion, award of the restaurant. Hence, the accuracy of this analysis as well as the business decision can be improved if we can obtain extra important data and take them in to account.

Last but not least, we can apply different clustering techniques like DBSCAN to obtain result from different aspect.

5. Conclusion

In 21st century while data is overflowing our daily life, they are very likely to be connected to some real life problem. If we can analysis these data properly, we are definitely going to obtain a solution to these problem.

We have seen some frequently used python libraries to retrieve, manipulate, visualise and analyse data like numpy, pandas, seaborn, matplotlib and folium in this project to provide a preferable location for new restaurant within London base on the data we retrieved.

Similarly, data can be used to solve other real life problems which we are encountering everyday. I believe there will are no unsolvable problem in the future if we can utilise the power of data and machine.