Analysing neighbourhoods of London!

Introduction:

We intend to analyse the neighbourhoods in City of London and will try to understand and explore neighbourhoods. Our intention is to get the most common venue categories in each neighbourhood, and then use this feature to group the neighbourhoods into clusters.

We will use the Foursquare API to explore neighbourhoods and get the relevant data for each neighbourhood.

We will use the k-means clustering algorithm to complete this task. Finally, we will use the Folium library to visualize the neighbourhoods in London City and their emerging clusters. This project will be useful for people coming in the City of London, which will help them with an idea of how similar and diverse different neighbourhoods in the City of London are. It would help them choose/pick the places of their choice easily, for the different activities they would like to do in the City of London.

Before we get the data and start exploring it, we download all the dependencies that we will need:

```
!conda install -c conda-forge lxml --yes
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
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
print('Libraries imported.')
```

```
Solving environment: done

# All requested packages already installed.

Libraries imported.
```

1. Download and Explore Dataset

Here we are creating a DataFrame from the London data available on : "https://en.wikipedia.org/wiki/List_of_London_boroughs"

```
tables = pd.read_html("https://en.wikipedia.org/wiki/List_of_London_boroughs")
df1 = pd.DataFrame(tables[0])
df1.head()
```

Here is now a sample of Data Frame that we just created:

	Borough	Inner	Status	Local authority	Political control	Headquarters	Area (sq mi)	Population (2013 est)[1]	Co-ordinates	Nr. in map
0	Barking and Dagenham [note 1]	NaN	NaN	Barking and Dagenham London Borough Council	Labour	Town Hall, 1 Town Square	13.93	194352	51°33′39″N 0°09′21″E / 51.5607°N 0.1557°E	25
1	Barnet	NaN	NaN	Barnet London Borough Council	Conservative	North London Business Park, Oakleigh Road South	33.49	369088	51°37′31″N 0°09′06″W / 51.6252°N 0.1517°W	31
2	Bexley	NaN	NaN	Bexley London Borough Council	Conservative	Civic Offices, 2 Watling Street	23.38	236687	51°27′18″N 0°09′02″E / 51.4549°N 0.1505°E	23
3	Brent	NaN	NaN	Brent London Borough Council	Labour	Brent Civic Centre, Engineers Way	16.70	317264	51°33′32″N 0°16′54″W / 51.5588°N 0.2817°W	12
4	Bromley	NaN	NaN	Bromley London Borough Council	Conservative	Civic Centre, Stockwell Close	57.97	317899	51°24′14″N 0°01′11″E / 51.4039°N 0.0198°E	20

Now we have (done all kind of Data wrangling and manipulations) and cleaned all relevant data in Data Frame .Now we have our final desired Data Frame on which we will perform our further analysis.

	Borough	Local authority	Political control	Headquarters	Area (sq mi)	Population	Longitude	Latitude
0	Barking and Dagenham	Barking and Dagenham London Borough Council	Labour	Town Hall, 1 Town Square	13.93	194352	0.1557	51.5607
1	Barnet	Barnet London Borough Council	Conservative	North London Business Park, Oakleigh Road South	33.49	369088	0.1517	51.6252
2	Bexley	Bexley London Borough Council	Conservative	Civic Offices, 2 Watling Street	23.38	236687	0.1505	51.4549
3	Brent	Brent London Borough Council	Labour	Brent Civic Centre, Engineers Way	16.70	317264	0.2817	51.5588
4	Bromley	Bromley London Borough Council	Conservative	Civic Centre, Stockwell Close	57.97	317899	0.0198	51.4039

Create a map of London using Folium with neighbourhoods superimposed on top:

Explore Neighbourhoods in London:

Next, we are going to start utilizing the Foursquare API to explore the neighbourhoods and segment them.

	Neighborhood	Neighborhood Latitude	Neighborhood 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	Beacontree Heath Leisure Centre	51.560997	0.148932	Gym / Fitness Center
4	Barking and Dagenham	51.5607	0.1557	Morrisons Becontree Heath	51.559774	0.148752	Supermarket

Let's check how many venues were returned for each neighborhood

```
df_venues.groupby('Neighborhood').count()
```

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Barking and Dagenham	7	7	7	7	7	7
Bexley	27	27	27	27	27	27
Brent	2	2	2	2	2	2
Bromley	40	40	40	40	40	40
Camden	4	4	4	4	4	4
Croydon	7	7	7	7	7	7
Ealing	2	2	2	2	2	2
Enfield	4	4	4	4	4	4
Greenwich	42	42	42	42	42	42
Hackney	7	7	7	7	7	7
Hammersmith and Fulham	3	3	3	3	3	3
Haringey	2	2	2	2	2	2
Havering	38	38	38	38	38	38
Hillingdon	1	1	1	1	1	1
Hounslow	1	1	1	1	1	1
Islington	4	4	4	4	4	4
Kingston upon Thames	1	1	1	1	1	1
Lambeth	9	9	9	9	9	9

Analyse Each Neighbourhood: (One-hot Encoding)

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

```
df_grouped = df_onehot.groupby('Neighborhood').mean().reset_index()
df_grouped
```

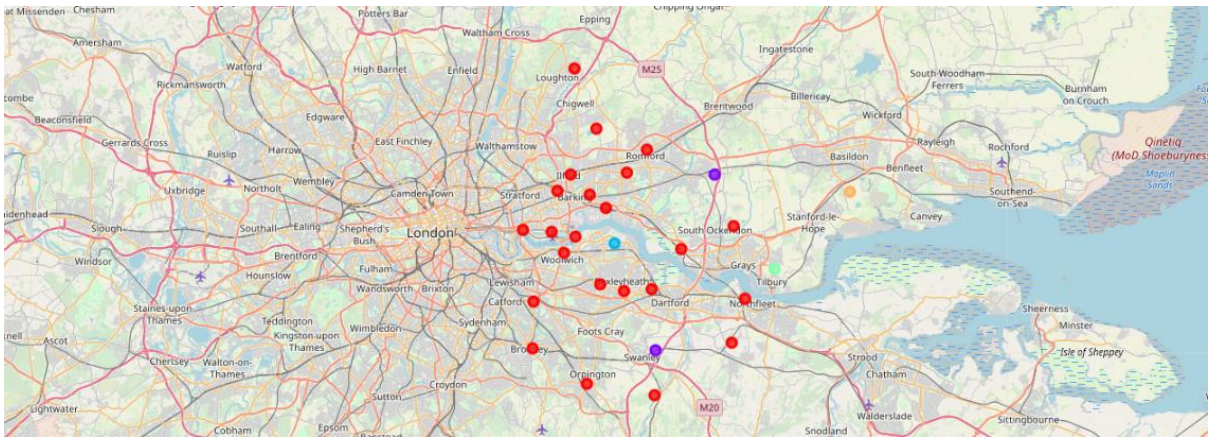
	Neighborhood	African Restaurant	Airport	Airport Lounge	Airport Service	American Restaurant	Asian Restaurant	Bakery	Bar	Bookstore	Boutique	Breakfast Spot	Buffet	Burger Joint
0	Barking and Dagenham	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
1	Bexley	0.00000	0.000000	0.000000	0.000000	0.037037	0.00000	0.037037	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
2	Brent	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
3	Bromley	0.00000	0.000000	0.000000	0.000000	0.000000	0.02500	0.025000	0.050000	0.025000	0.025000	0.00000	0.000000	0.050000
4	Camden	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
5	Croydon	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.142857	0.000000	0.000000	0.00000	0.000000	0.000000
6	Ealing	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
7	Enfield	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
8	Greenwich	0.02381	0.000000	0.000000	0.000000	0.000000	0.02381	0.023810	0.000000	0.023810	0.000000	0.02381	0.000000	0.023810
9	Hackney	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
10	Hammersmith and Fulham	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
11	Haringey	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
12	Havering	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.052632	0.026316	0.052632	0.026316	0.00000	0.000000	0.000000
13	Hillingdon	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
14	Hounslow	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
15	Islington	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000

Cluster Neighbourhoods: (Using K- means clustering)

Run *k*-means to cluster the neighbourhood into five clusters.

	Neighborhood	Local authority	Political control	Headquarters	Area (sq mi)	Population	Longitude	Latitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	
0	Barking and Dagenham	Barking and Dagenham London Borough Council	Labour	Town Hall, 1 Town Square	13.93	194352	0.1557	51.5607	0	Pool	Gym / Fitness Center	Golf Course	Park	S
1	Bexley	Bexley London Borough Council	Conservative	Civic Offices, 2 Watling Street	23.38	236687	0.1505	51.4549	0	Pub	Clothing Store	Italian Restaurant	Fast Food Restaurant	C
2	Brent	Brent London Borough Council	Labour	Brent Civic Centre, Engineers Way	16.70	317264	0.2817	51.5588	1	Pub	Golf Course	Warehouse Store	Electronics Store	
3	Bromley	Bromley London Borough Council	Conservative	Civic Centre, Stockwell Close	57.97	317899	0.0198	51.4039	0	Coffee Shop	Clothing Store	Gym / Fitness Center	Bar	F
4	Camden	Camden London Borough Council	Labour	Camden Town Hall, Judd Street	8.40	229719	0.1255	51.5290	0	Home Service	Gym	Rugby Pitch	Skate Park	

Finally, let us visualize the resulting cluster using Folium map:



Examine Clusters:

Now, we can examine each cluster and determine the discriminating venue categories that distinguish each cluster. Based on the defining categories, we can then assign a name to each cluster. Here we can clearly see below how distinct and clearly similar/dissimilar the different clusters are among themselves.

Cluster 1 : Seems like a place for Cafe , Pubs , Pools and Supermarkets

```
df_merged.loc[df_merged['Cluster Labels'] == 0, df_merged.columns[[1] + list(range(5, df_merged.shape[1]))]]
```

	Local authority	Population	Longitude	Latitude	Cluster Labels	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 London Borough Council	194352	0.1557	51.5607	0	Pool	Gym / Fitness Center	Golf Course	Park	Supermarket	Martial Arts Dojo	Bus Station	Duty-free Shop
1	Bexley London Borough Council	236687	0.1505	51.4549	0	Pub	Clothing Store	Italian Restaurant	Fast Food Restaurant	Coffee Shop	Supermarket	Chinese Restaurant	Bakery
3	Bromley London Borough Council	317899	0.0198	51.4039	0	Coffee Shop	Clothing Store	Gym / Fitness Center	Bar	Pizza Place	Burger Joint	Electronics Store	Cosmetics Shop
4	Camden London Borough Council	229719	0.1255	51.5290	0	Home Service	Gym	Rugby Pitch	Skate Park	Duty-free Shop	Cosmetics Shop	Department Store	Dessert Shop
5	Croydon London Borough Council	372752	0.0977	51.3714	0	Pizza Place	Coffee Shop	Pub	Italian Restaurant	Chinese Restaurant	Supermarket	Bar	Duty-free Shop
6	Ealing London Borough Council	342494	0.3089	51.5130	0	Home Service	Business Service	Fast Food Restaurant	Department Store	Dessert Shop	Diner	Discount Store	Dog Run
7	Enfield London Borough	320524	0.0799	51.6538	0	Park	Dog Run	Shopping Plaza	Warehouse Store	Electronics Store	Department Store	Dessert Shop	Diner

Cluster 2 Seems like a place for Pubs ,Golf Courses and Electronics and Warehouses

```
df_merged.loc[df_merged['Cluster Labels'] == 1, df_merged.columns[[1] + list(range(5, df_merged.shape[1]))]]
```

	Local authority	Population	Longitude	Latitude	Cluster Labels	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	9th Most Common Venue	Co
2	Brent London Borough Council	317264	0.2817	51.5588	1	Pub	Golf Course	Warehouse Store	Electronics Store	Cosmetics Shop	Department Store	Dessert Shop	Diner	Discount Store	D
19	Merton London Borough Council	203223	0.1958	51.4014	1	Pub	Warehouse Store	Electronics Store	Cosmetics Shop	Department Store	Dessert Shop	Diner	Discount Store	Dog Run	

Cluster 3 Seems like a place for Breakfast spots and Small shops and stores

```
df_merged.loc[df_merged['Cluster Labels'] == 2, df_merged.columns[[1] + list(range(5, df_merged.shape[1]))]]
```

	Local authority	Population	Longitude	Latitude	Cluster Labels	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	9th Most Common Venue	10th Comm Ve
27	Westminster City Council	226841	0.1372	51.4973	2	Breakfast Spot	Fish & Chips Shop	Department Store	Dessert Shop	Diner	Discount Store	Dog Run	Donut Shop	Duty-free Shop	Electro S

Cluster 4 Seems like a place for Fast Foods , eateries and some construction/ landscape along with warehousing places

```
df_merged.loc[df_merged['Cluster Labels'] == 3, df_merged.columns[[1] + list(range(5, df_merged.shape[1]))]]
```

	Local authority	Population	Longitude	Latitude	Cluster Labels	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	9th Most Common Venue	Co
14	Hounslow London Borough Council	262407	0.368	51.4746	3	Fast Food Restaurant	Warehouse Store	Construction & Landscaping	Department Store	Dessert Shop	Diner	Discount Store	Dog Run	Donut Shop	Du

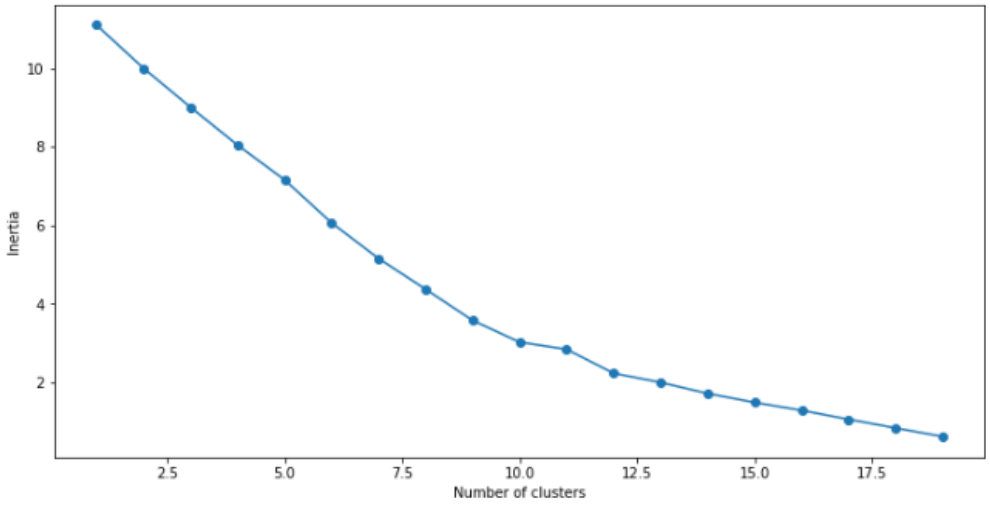
Cluster 5 Seems like a place for Distinct with Stables , Cosmetics and Departmental stores

```
df_merged.loc[df_merged['Cluster Labels'] == 4, df_merged.columns[[1] + list(range(5, df_merged.shape[1]))]]
```

	Local authority	Population	Longitude	Latitude	Cluster Labels	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	9th Most Common Venue	10 Most Common Ven
13	Hillingdon London Borough Council	286806	0.476	51.5441	4	Stables	Fast Food Restaurant	Cosmetics Shop	Department Store	Dessert Shop	Diner	Discount Store	Dog Run	Donut Shop	Duty-fr Sh

Possible K values for optimal analysis of clusters {Elbow point Observation}:

We can also go ahead and try to analyse what K values we can choose to define different types of Clusters. There is a popular method known as elbow method, which is used to determine the optimal value of K to perform the K-Means Clustering Algorithm. The basic idea behind this method is that it plots the various values of cost with changing k. As the value of K increases, there will be fewer elements in the cluster. So average distortion will decrease. The lesser number of elements means closer to the centroid. Therefore, the point where this distortion declines the most is the elbow point.



Conclusion on K- means , K values : Here we can clearly see that a possible range of values from 5-8 seems optimum for K - means clustering anaysis here

Results: Conclusion on K- means, K values:

Here we can clearly see that, a possible range of values, from 5-8 seems optimum for K - means clustering analysis used here.

Discussion:

Our conclusion is two folds here:

We have identified that we can go for K=8 for optimum results on K-means clustering. We can then easily identify the clusters and their different categories for (Cafe , Pubs , Pools and Supermarkets) or (Pubs ,Golf Courses and Electronics and Warehouses) or (Breakfast spots and Small shops and stores) or (Fast Foods , eateries and some construction/ landscape along with warehousing places) or (Distinct with Stables , Cosmetics and Departmental store).

Another point is that, we had used above, value of LIMIT = 100 # limit of number of venues returned by Foursquare API. If we increase this number to a larger value, then we would have more data and venues to cluster and which would refine our data and analysis, both. This would result in crisp and much detailed findings.

Conclusion:

Our conclusion is two folds here. We have identified that we can go for K= 8 for optimum results on K-means clustering. We can then easily identify the clusters and their different categories for (Cafe, Pubs, Pools and Supermarkets) or (Pubs, Golf Courses and Electronics and Warehouses) or (Breakfast spots and Small shops and stores) or (Fast Foods, eateries and some construction/ landscape along with warehousing places) or (Distinct with Stables, Cosmetics and Departmental store)

Another point is that we had used above LIMIT = 100 # limit of number of venues returned by Foursquare API. If we increase this number to a larger value, then we would have more data and venues to cluster, which would refine our data, and analysis both. This would result in crisp and much detailed findings.