Chinese Restaurants Neighbourhoods of London

Final Project

Applied Capstone

Ester Naranjo Rodrigo

28-01-2020

Index

1. Introduction
2. Business problem
3. Data
   Wikipedia
   Foursquare API
   ArcGIS geolocations
4. Methodology
   Tables and data obtained from Wikipedia and Foursquare
5. Results
6. Discussion
7. Conclusion

Introduction

This is the final project of the IBM Data Science Course on Coursera. In this project, I have analyzed the scenario of Chinese Restaurants in London. The results and data obtained might be used such as information for tourisms guides, neighbourhoods where it is possible to create a new Chinese restaurant or areas to avoid in the city for this news Chinese restaurants. Whit these ideas in mind, I have developed this project to get the correct information for tourists or the best locations for new restaurant. The project will be developed with python, Foursquare API and all tools and mechanism that I have learned in this IBM Course.

Business problem

The aim of this project is to help to find the best location for any person that wants establish his Chinese Restaurant in London. In addition, this project can help to elaborate a tourist guide with the better information about Chinese Restaurants in the capital of United Kingdom.

Data

It is necessary data about the boroughs (or Neighbourhoods), geolocation data of each borough and all the venues in each borough visited by the people in London. When we have obtained this information will be relevant the correct union of valid information in order to cluster the Neighbourhoods and obtain valuables results. We are going to use the following apps or websites pages in order to obtain the primary information:

1. Wikipedia
2. Foursquare API
3. ArcGIS geolocations

Wikipedia

To obtain the data of London's Borough, we have to scrape data from: https://en.wikipedia.org/wiki/List_of_areas_of_London

On this website we will obtain: Borough, Town and Postal code after clean the dataframes.

1. Borough: Name of Neighbourhood
2. Town: Name of Borough
3. Post-code: Postal codes in London

Foursquare API

We need credentials in order to obtain the information, so first of all we have to register in Foursquare Developer API https://foursquare.com/. All the information about venues location in London will be provided by Foursquare API. This information will be the cornerstone to elaborate this project.

On this website we will obtain: Neighbourhoods, latitude, longitude, venues and venues category.

1. Neighbourhood: Name of Neighbourhood in London
2. Latitude Neighbourhood: Latitude coordinate of each Neighbourhood in London
3. Longitude Neighbourhood: Longitude coordinate of each Neihgbourhood in London
4. Venue: Name of venue in London
5. Venue Category: Category assigned of each venue in London

ArcGIS geolocations

Arcgis is a System of Geographic Information (GIS) that provide to the project the coordinates of each neighbourhood and the city of London in order to obtain the maps and make the cluster with Folium.

1. Latitude Neighbourhood: Data of latitude coordinates
2. Longitude Neighbourhood: Data of longitude coordinates

1. Methodology

First, we need to get the list of Boroughs in London. So, we are extracting the list of borough from Wikipedia: https://en.wikipedia.org/wiki/List_of_areas_of_London.

We have to do web scraping by utilizing pandas with read_html to pull tabular directly from Wikipedia page into a data frame.

	Borough	Town	Post-code
0	Bexley, Greenwich	LONDON	SE2
1	Ealing, Hammersmith and Fulham	LONDON	W3, W4
2	Croydon	CROYDON	CR0
3	Croydon	CROYDON	CR0
4	Bexley	BEXLEY, SIDCUP	DA5, DA14
...	...	...	...
526	Greenwich	LONDON	SE18
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4
528	Hammersmith and Fulham	LONDON	W12
529	Hillingdon	HAYES	UB4
530	Hillingdon	WEST DRAYTON	UB7

531 rows × 3 columns

Obtain the shape and info dataframe

```
df.shape
```

```
(531, 3)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 531 entries, 0 to 530
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Borough    531 non-null    object
 1   Town       531 non-null    object
 2   Post-code  531 non-null    object
dtypes: object(3)
memory usage: 12.6+ KB
```

We must concatenate this list of borough names and postal codes of London with its coordinates. For this reason, it is necessary to use ArcGIS in order to get the coordinates of each neighbour in London. First, we have to get the coordinates in function of postal code and then we have to concatenate this results with the table obtained from Wikipedia.

```
coordinates_london = post_code_london.apply(lambda x: get_x_y_london(x))
coordinates_london
```

```
0       51.492450000000076,0.12127000000003818
1        51.51324000000005,-0.2674599999999714
2       51.38475500000004,-0.05149847299992416
3       51.38475500000004,-0.05149847299992416
4        51.50642000000005,-0.1272099999999341
                          ...
526     51.48207000000008,0.07143000000002075
527      51.50642000000005,-0.1272099999999341
528      51.50645000000003,-0.2369099999999662
529      51.50642000000005,-0.1272099999999341
530      51.50642000000005,-0.1272099999999341
Name: Post-code, Length: 531, dtype: object
```

Concat the two dataframes (wikipedia dataframe and arcgis dataframe)

```
london_merged = pd.concat([df,lat_uk.astype(float), long_uk.astype(float)], axis=1)
london_merged.columns= ['Borough','Town','Post-code','Latitude','Longitude']
london_merged
```

	Borough	Town	Post-code	Latitude	Longitude
0	Bexley, Greenwich	LONDON	SE2	51.492450	0.121270
1	Ealing, Hammersmith and Fulham	LONDON	W3, W4	51.513240	-0.267460
2	Croydon	CROYDON	CR0	51.384755	-0.051498
3	Croydon	CROYDON	CR0	51.384755	-0.051498
4	Bexley	BEXLEY, SIDCUP	DA5, DA14	51.506420	-0.127210
...	...	...	...	...	...
526	Greenwich	LONDON	SE18	51.482070	0.071430
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4	51.506420	-0.127210
528	Hammersmith and Fulham	LONDON	W12	51.506450	-0.236910
529	Hillingdon	HAYES	UB4	51.506420	-0.127210
530	Hillingdon	WEST DRAYTON	UB7	51.506420	-0.127210

531 rows × 5 columns

Finally, to finish this geolocation part we have to import the coordinates of the city of London in order to make a map and locate the neighbour on this map.

```
london_long_coords = london['location']['x']
london_lat_coords = london['location']['y']
print('The coordinates of London are {}, {}.'.format(london_lat_coords, london_long_coords))
```

```
The coordinates of London are 51.50642000000005, -0.12720999999999341.
```

Create and visualize London's map with Folium

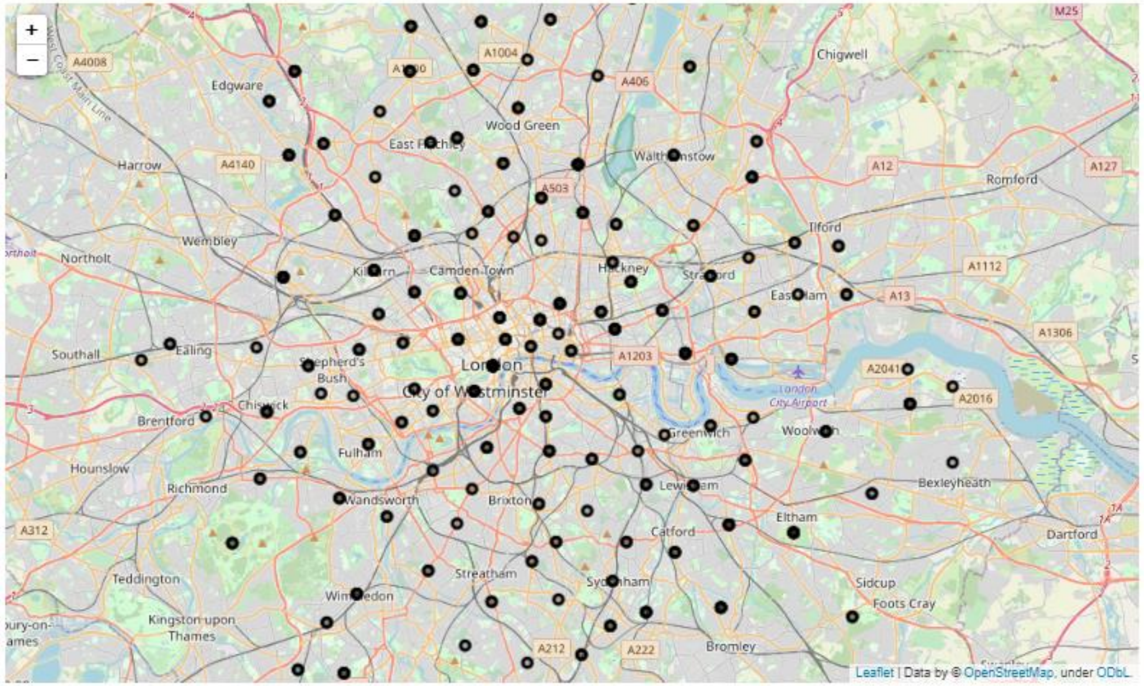
```
import folium
```

```
map_London = folium.Map(location=[london_lat_coords, london_long_coords], zoom_start=11)
```

Adding markers neighbourhoods to map

```
for latitude, longitude, borough, town in zip(london_merged['Latitude'], london_merged['Longitude'], london_merged['Borough'], lo
    label = '{}, {}'.format(town, borough)
    label = folium.Popup(label, parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius = 4,
        popup = label,
        color = 'black',
        fill = True
        ).add_to(map_London)

map_London
```



Next, it will be used Foursquare API to obtain the list of venues in a 500 meters radius from each Neighbourhood. First of all, I had created an account on Foursquare API developers to obtain the credentials that allow me to obtain these calls to Foursquare API. The result of these calls is the following data frame

```
venues_in_london.shape
```

```
(21560, 5)
```

```
venues_in_london.head()
```

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
0	Bexley, Greenwich	51.49245	0.12127	Lesnes Abbey	Historic Site
1	Bexley, Greenwich	51.49245	0.12127	Sainsbury's	Supermarket
2	Bexley, Greenwich	51.49245	0.12127	Lidl	Supermarket
3	Bexley, Greenwich	51.49245	0.12127	Abbey Wood Railway Station (ABW)	Train Station
4	Bexley, Greenwich	51.49245	0.12127	Bean @ Work	Coffee Shop

Tables and data obtained from Wikipedia and Foursquare

- Number of venues on each Neighbourhood

```
venues_in_london['Neighbourhood'].value_counts()
```

```
Westminster                                      1570
Havering                                         1386
Bromley                                          1378
Bexley                                           1224
Hillingdon                                       1122
Richmond upon Thames                              885
Hounslow                                          817
Camden                                            804
Harrow                                            792
Kingston upon Thames                              706
Islington                                         694
Barnet                                            642
Croydon                                           620
Hackney                                           600
Redbridge                                         586
Kensington and Chelsea                            567
Southwark                                         525
Brent                                             472
Tower Hamlets                                     469
Ealing                                            417
Barking and Dagenham                              410
Hammersmith and Fulham                            410
Lewisham                                          400
Wandsworth                                        370
Newham                                            356
Sutton                                            331
Haringey                                          317
Merton                                            304
Lambeth                                           291
Waltham Forest                                    257
City                                              235
Enfield                                           224
Greenwich                                         151
Bexley, Bromley                                   132
City, Westminster                                 100
Kensington and Chelsea, Hammersmith and Fulham     96
Redbridge, Waltham Forest                          94
Islington, City                                    79
Lambeth, Wandsworth                                73
Brent, Harrow                                      66
Brent, Ealing, Harrow                              66
Barnet, Enfield                                    66
Redbridge, Barking and Dagenham                    66
Sutton, Kingston upon Thames                       66
Haringey, Barnet                                   45
Lambeth, Southwark                                 41
Camden, Islington                                  41
Hounslow, Ealing, Hammersmith and Fulham           40
Brent, Camden                                      34
Islington, Camden                                  31
Haringey, Islington                                26
Lewisham, Bromley                                  13
Greenwich, Lewisham                                 9
Brent, Ealing                                       8
Bexley, Greenwich                                   8
Lewisham, Southwark                                 8
Ealing, Hammersmith and Fulham                      7
Barnet, Brent, Camden                               5
Bexley, Greenwich                                   5
Harrow, Brent                                       3
Name: Neighbourhood, dtype: int64
```

- Number of Chinese Restaurants in London

```
g = Gb_venues_category
g['Venue'].loc['Chinese Restaurant']
```

```
93
```

- Number of Neighbourhood with at least one Chinese Restaurant

```
len(Gb_neighbourhood_london[Gb_neighbourhood_london['Chinese Restaurant'] > 0])
```

```
18
```

Tables and data obtained from Wikipedia and Foursquare

- Venues in London group by Neighbourhood

```
Gb_neighbourhood = venues_in_london.groupby('Neighbourhood').head()
Gb_neighbourhood
```

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
0	Bexley, Greenwich	51.49245	0.12127	Lesnes Abbey	Historic Site
1	Bexley, Greenwich	51.49245	0.12127	Sainsbury's	Supermarket
2	Bexley, Greenwich	51.49245	0.12127	Lidl	Supermarket
3	Bexley, Greenwich	51.49245	0.12127	Abbey Wood Railway Station (ABW)	Train Station
4	Bexley, Greenwich	51.49245	0.12127	Bean @ Work	Coffee Shop
...	...	...	...	...	...
21333	Sutton, Kingston upon Thames	51.50642	-0.12721	Corinthia Hotel	Hotel
21334	Sutton, Kingston upon Thames	51.50642	-0.12721	Trafalgar Square	Plaza
21335	Sutton, Kingston upon Thames	51.50642	-0.12721	East Trafalgar Square Fountain	Fountain
21336	Sutton, Kingston upon Thames	51.50642	-0.12721	Horse Guards Parade	Plaza
21337	Sutton, Kingston upon Thames	51.50642	-0.12721	Trafalgar Square Lions	Outdoor Sculpture

298 rows × 5 columns

The following step on the methodology is the One Hot Encoding, we need to encode our venue categories to get a better result for our clustering.

One Hot encoding

```
london_venue_category = pd.get_dummies(venues_in_london[['Venue Category']], prefix = "", prefix_sep = "")
london_venue_category
```

	Accessories Store	Adult Boutique	African Restaurant	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	...	Vietnamese Restaurant	Warehouse Store	Whisky Bar	Wine Ba
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
21555	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
21556	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
21557	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
21558	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
21559	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0

21560 rows × 308 columns

To the last data frame we have to add the column Neighbourhood and group this data by each Neighbourhood.

```
Gb_neighbourhood_london = london_venue_category.groupby('Neighbourhood').mean().reset_index().round(8)
Gb_neighbourhood_london.head()
```

	Neighbourhood	Accessories Store	Adult Boutique	African Restaurant	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	...	Vietnamese Restaurant	Warehouse Store	Whisky Bar
0	Barking and Dagenham	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.029268	...	0.000000	0.0	0.0
1	Barnet	0.0	0.0	0.0	0.001558	0.0	0.0	0.0	0.006231	0.003115	...	0.001558	0.0	0.0
2	Barnet, Brent, Camden	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.000000	0.0	0.0
3	Barnet, Enfield	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.030303	...	0.000000	0.0	0.0
4	Bexley	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.029412	...	0.000000	0.0	0.0

5 rows × 309 columns

The next step is obtaining a data frame with the neighbourhood and its hot encoding of ‘Chinese Restaurant’ that is the aim of this project. With this data frame, we can perform the clustering method by using k-means.

```
london_chinese_resta = Gb_neighbourhood_london[['Neighbourhood','Chinese Restaurant']]
london_chinese_resta
```

	Neighbourhood	Chinese Restaurant
0	Barking and Dagenham	0.000000
1	Barnet	0.012461
2	Barnet, Brent, Camden	0.000000
3	Barnet, Enfield	0.000000
4	Bexley	0.000000
5	Bexley, Bromley	0.000000
6	Bexley, Greenwich	0.000000
7	Bexley, Greenwich	0.000000
8	Brent	0.014831
9	Brent, Camden	0.000000
10	Brent, Ealing	0.125000
11	Brent, Ealing, Harrow	0.000000
12	Brent, Harrow	0.000000
13	Bromley	0.000000
14	Camden	0.002488
15	Camden, Islington	0.000000
16	City	0.004255
17	City, Westminster	0.000000
18	Croydon	0.000000
19	Ealing	0.000000
20	Ealing, Hammersmith and Fulham	0.000000
21	Enfield	0.000000
22	Greenwich	0.026490
23	Greenwich, Lewisham	0.000000
24	Hackney	0.001667
25	Hammersmith and Fulham	0.012195
26	Haringey	0.000000
27	Haringey, Barnet	0.000000
28	Haringey, Islington	0.000000
29	Harrow	0.000000
30	Harrow, Brent	0.000000
31	Havering	0.000000
32	Hillingdon	0.000000
33	Hounslow	0.000000
34	Hounslow, Ealing, Hammersmith and Fulham	0.000000
35	Islington	0.005764
36	Islington, Camden	0.000000
37	Islington, City	0.012658
38	Kensington and Chelsea	0.003527
39	Kensington and Chelsea, Hammersmith and Fulham	0.000000
40	Kingston upon Thames	0.000000
41	Lambeth	0.006873
42	Lambeth, Southwark	0.000000
43	Lambeth, Wandsworth	0.000000
44	Lewisham	0.010000
45	Lewisham, Bromley	0.000000
46	Lewisham, Southwark	0.000000
47	Merton	0.000000
48	Newham	0.000000
49	Redbridge	0.003413
50	Redbridge, Barking and Dagenham	0.000000
51	Redbridge, Waltham Forest	0.000000
52	Richmond upon Thames	0.000000
53	Southwark	0.007619
54	Sutton	0.000000
55	Sutton, Kingston upon Thames	0.000000
56	Tower Hamlets	0.046908
57	Waltham Forest	0.003891
58	Wandsworth	0.000000
59	Westminster	0.014013

We can analysed more the data in order to obtain the best clustering so I find the most common venues of each neighbourhood and I look at if any Chinese Restaurant is among these common venues.

Finding the most common venues

```
def 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]
```

```
import numpy as np
```

Decision of number of common venues due to there are many venues in order to evaluate

```
number_common_venues = 12

ord_number = ['st', 'nd', 'rd']

columns = ['Neighbourhood']
for ordinal in np.arange(number_common_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ordinal+1, ord_number[ordinal]))
    except:
        columns.append('{}th Most Common Venue'.format(ordinal+1))


neighbourhoods_venues_sorted = pd.DataFrame(columns = columns)
neighbourhoods_venues_sorted['Neighbourhood'] = Gb_neighbourhood_london['Neighbourhood']

for i in np.arange(Gb_neighbourhood_london.shape[0]):
    neighbourhoods_venues_sorted.iloc[i, 1:] = most_common_venues(Gb_neighbourhood_london.iloc[i, :], number_common_venues)

neighbourhoods_venues_sorted
```

	Neighbourhood	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 Most Common Venue	1
0	Barking and Dagenham	Hotel	Plaza	Theater	Garden	Monument / Landmark	Cocktail Bar	Boutique	Ramen Restaurant	Burger Joint	Wine Bar	A
1	Barnet	Coffee Shop	Café	Grocery Store	Pub	Bus Stop	Italian Restaurant	Supermarket	Pharmacy	Sushi Restaurant	Turkish Restaurant	
2	Barnet, Brent, Camden	Hardware Store	Supermarket	Bakery	Clothing Store	Gym / Fitness Center	Office	Noodle House	Okonomiyaki Restaurant	Opera House	Optical Shop	P A
3	Barnet, Enfield	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	Japanese Restaurant	Sandwich Place	A
4	Bexley	Hotel	Garden	Theater	Plaza	Monument / Landmark	Boutique	Cocktail Bar	Sandwich Place	Ramen Restaurant	Burger Joint	St
5	Bexley, Bromley	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	Japanese Restaurant	Sandwich Place	A

```
neighbourhoods_venues_sorted['Neighbourhood']

numbers = list(range(1,13))
numbers_str = list(map(str, numbers))
ordinal_numbers = ["st", "nd", "rd", "th", "th", "th", "th", "th", "th", "th", "th", "th"]


for i in range(1, 13):
    a = neighbourhoods_venues_sorted.loc[neighbourhoods_venues_sorted[str(i) + ordinal_numbers[i-1] +' Most Common Venue'] == "Ch
    if ("Chinese Restaurant" in neighbourhoods_venues_sorted[str(i) + ordinal_numbers[i-1] + " Most Common Venue"].unique()):
        print("A Chinese Restaurant is the", i, ordinal_numbers[i-1], "Most Common Venue in\n", a)
```

```
A Chinese Restaurant is the 3 rd Most Common Venue in
 56    Tower Hamlets
Name: Neighbourhood, dtype: object
A Chinese Restaurant is the 5 th Most Common Venue in
 10    Brent, Ealing
Name: Neighbourhood, dtype: object
```

The following step of the methodology is a important step, k-means clustering algorithm identifies k number of centroids an then allocates every data point to the nearest cluster while keeping the centroids as small as possible. It is an unsupervised machine learning. I have decided clustering the neighbourhoods in London into 5 clusters based on their frequency of occurrence for ‘Chinese Restaurant’.

```
k_num = 5

london_clustering = london_chinese_resta.drop(['Neighbourhood'], 1)

k_means = KMeans(n_clusters = k_num, random_state=0).fit(london_clustering)
k_means
```

```
KMeans(n_clusters=5, random_state=0)
```

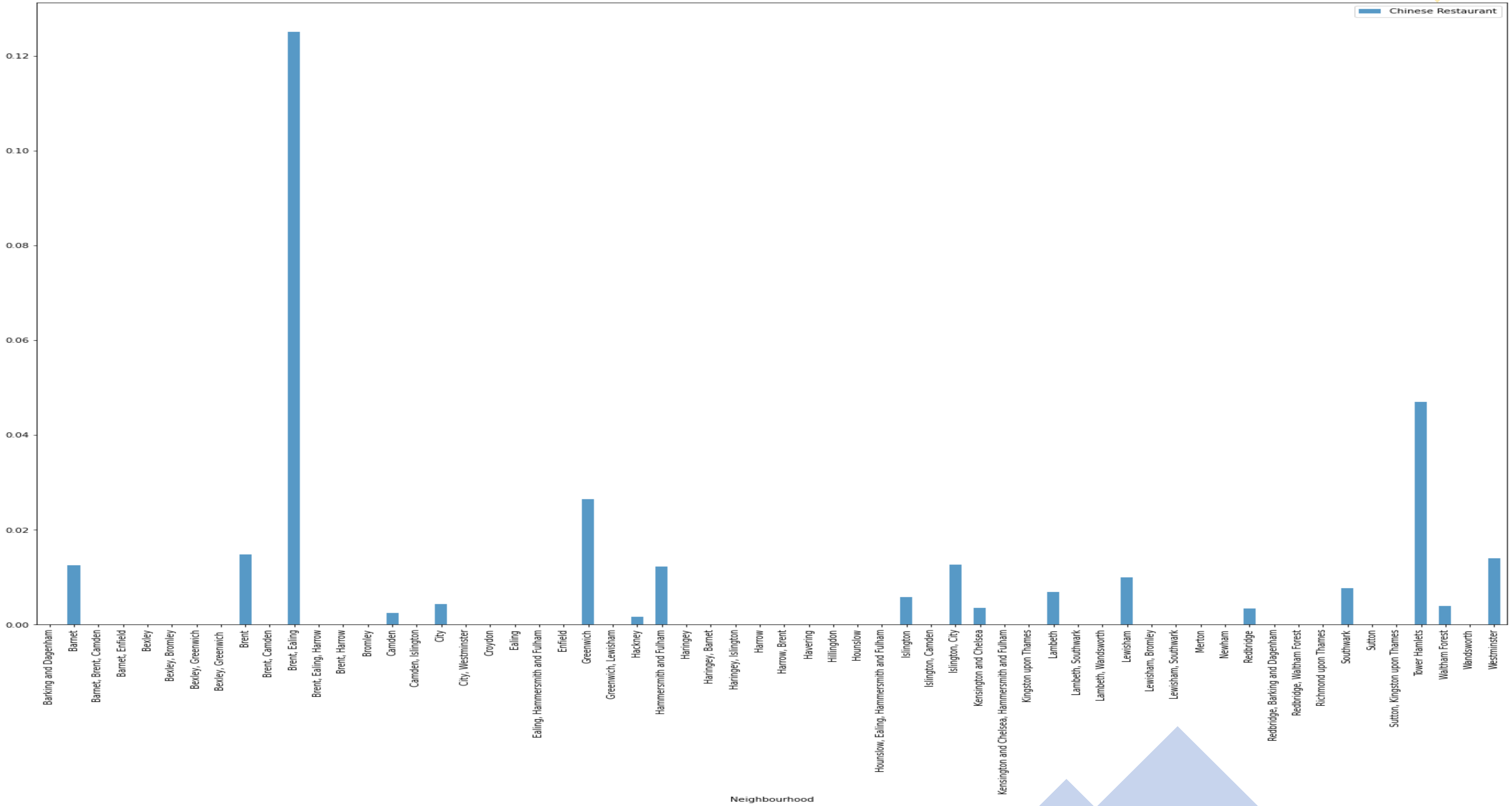
```
k_means.labels_[0:]
```

```
array([2, 0, 2, 2, 2, 2, 2, 2, 0, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       4, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 2, 2, 2, 0, 2, 2,
       0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 3, 2, 2, 0], dtype=int32)
```

	Neighbourhood	Chinese Restaurant	Cluster Labels
0	Barking and Dagenham	0.000000	2
1	Barnet	0.012461	0
2	Barnet, Brent, Camden	0.000000	2
3	Barnet, Enfield	0.000000	2
4	Bexley	0.000000	2
5	Bexley, Bromley	0.000000	2
6	Bexley, Greenwich	0.000000	2
7	Bexley, Greenwich	0.000000	2
8	Brent	0.014831	0
9	Brent, Camden	0.000000	2
10	Brent, Ealing	0.125000	1
11	Brent, Ealing, Harrow	0.000000	2
12	Brent, Harrow	0.000000	2
13	Bromley	0.000000	2
14	Camden	0.002488	2
15	Camden, Islington	0.000000	2
16	City	0.004255	2
17	City, Westminster	0.000000	2
18	Croydon	0.000000	2
19	Ealing	0.000000	2
20	Ealing, Hammersmith and Fulham	0.000000	2
21	Enfield	0.000000	2
22	Greenwich	0.026490	4
23	Greenwich, Lewisham	0.000000	2
24	Hackney	0.001667	2
25	Hammersmith and Fulham	0.012195	0
26	Haringey	0.000000	2
27	Haringey, Barnet	0.000000	2
28	Haringey, Islington	0.000000	2
29	Harrow	0.000000	2
30	Harrow, Brent	0.000000	2
31	Havering	0.000000	2
32	Hillingdon	0.000000	2
33	Hounslow	0.000000	2
34	Hounslow, Ealing, Hammersmith and Fulham	0.000000	2
35	Islington	0.005764	0
36	Islington, Camden	0.000000	2
37	Islington, City	0.012658	0
38	Kensington and Chelsea	0.003527	2
39	Kensington and Chelsea, Hammersmith and Fulham	0.000000	2
40	Kingston upon Thames	0.000000	2
41	Lambeth	0.006873	0
42	Lambeth, Southwark	0.000000	2
43	Lambeth, Wandsworth	0.000000	2
44	Lewisham	0.010000	0
45	Lewisham, Bromley	0.000000	2
46	Lewisham, Southwark	0.000000	2
47	Merton	0.000000	2
48	Newham	0.000000	2
49	Redbridge	0.003413	2
50	Redbridge, Barking and Dagenham	0.000000	2
51	Redbridge, Waltham Forest	0.000000	2
52	Richmond upon Thames	0.000000	2
53	Southwark	0.007619	0
54	Sutton	0.000000	2
55	Sutton, Kingston upon Thames	0.000000	2
56	Tower Hamlets	0.046908	3
57	Waltham Forest	0.003891	2
58	Wandsworth	0.000000	2
59	Westminster	0.014013	0

Results

- Hot encoding graphic

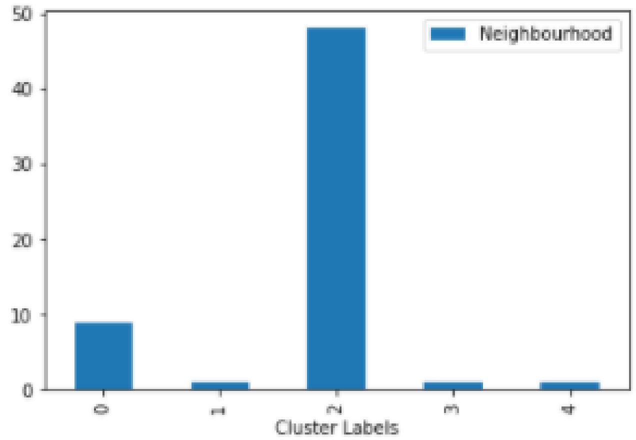


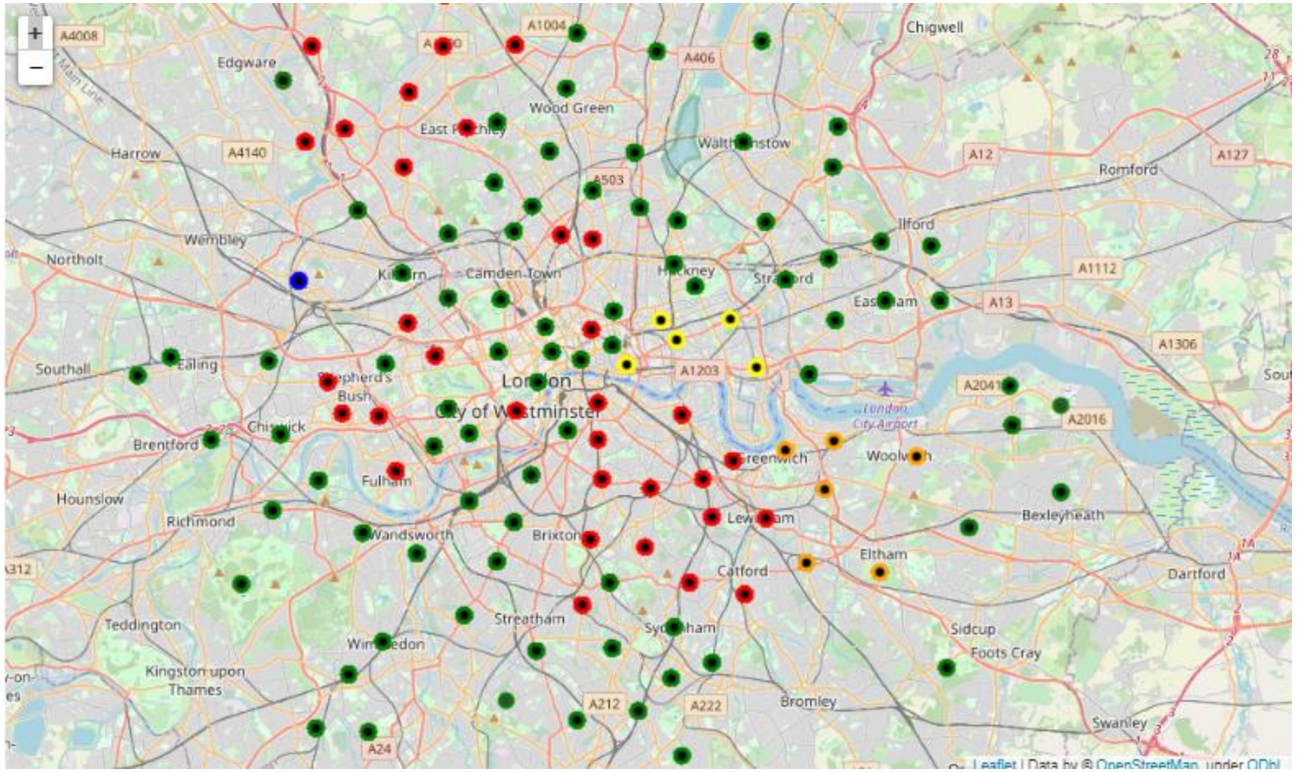
- Data frame merged with all data frames obtained.

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
59	Westminster	0.014013	0	51.49713	-0.13829	Loco Mexicano	Mexican Restaurant
35	Islington	0.005764	0	51.56393	-0.12945	Starbucks	Coffee Shop
35	Islington	0.005764	0	51.56393	-0.12945	Costa Coffee	Coffee Shop
35	Islington	0.005764	0	51.56393	-0.12945	Il Mio Mosaic	Italian Restaurant
35	Islington	0.005764	0	51.56393	-0.12945	The Landseer	Pub
...	...	...	...	...	...	...	...
22	Greenwich	0.026490	4	51.48454	0.00275	Star Express	Café
22	Greenwich	0.026490	4	51.48454	0.00275	Gurkha's Inn	Indian Restaurant
22	Greenwich	0.026490	4	51.48454	0.00275	Co-op Food	Grocery Store
22	Greenwich	0.026490	4	51.48454	0.00275	Tyler Street Bus Stop	Bus Stop
22	Greenwich	0.026490	4	51.48454	0.00275	Maze Hill Railway Station (MZH)	Train Station

21560 rows × 7 columns

- Number of neighbourhoods in each cluster.





*Cluster 1 Neighbourhoods with few Chinese restaurants in London

*Cluster 2 Neighbourhoods whit the greatest number of Chinese restaurants in London

*Cluster 3 Neighbourhoods with 0 or low number of Chinese restaurants in London

*Cluster 4 Neighbourhoods with high number of Chinese restaurants in London

*Cluster 5 Neighbourhoods with considerable number of Chinese restaurants in London

Cluster 1 (Red)

```
london_chinese_rest_cluster_0 = final_merged.loc[(final_merged['Cluster Labels'] == 0) & (final_merged['Venue Category'] == 'Chi
london_chinese_rest_cluster_0.drop_duplicates()
```

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
35	Islington	0.005764	0	51.52361	-0.09877	New East House	Chinese Restaurant
8	Brent	0.014831	0	51.53938	-0.25205	Good Taste	Chinese Restaurant
53	Southwark	0.007619	0	51.47480	-0.09313	Tasty House	Chinese Restaurant
53	Southwark	0.007619	0	51.47480	-0.09313	Lamoon	Chinese Restaurant
25	Hammersmith and Fulham	0.012195	0	51.53938	-0.25205	Good Taste	Chinese Restaurant
25	Hammersmith and Fulham	0.012195	0	51.47772	-0.20145	Royal China	Chinese Restaurant
44	Lewisham	0.010000	0	51.46268	-0.03558	Bamboo Garden	Chinese Restaurant
25	Hammersmith and Fulham	0.012195	0	51.49617	-0.22935	Steam Restaurant	Chinese Restaurant
41	Lambeth	0.006873	0	51.47480	-0.09313	Lamoon	Chinese Restaurant
41	Lambeth	0.006873	0	51.47480	-0.09313	Tasty House	Chinese Restaurant
37	Islington, City	0.012658	0	51.52361	-0.09877	New East House	Chinese Restaurant
44	Lewisham	0.010000	0	51.47489	-0.04038	Yao Kee	Chinese Restaurant
59	Westminster	0.014013	0	51.49713	-0.13829	A Wong	Chinese Restaurant
59	Westminster	0.014013	0	51.49713	-0.13829	Dragon Inn Club	Chinese Restaurant
59	Westminster	0.014013	0	51.51651	-0.11968	Kam Fung	Chinese Restaurant
59	Westminster	0.014013	0	51.51651	-0.11968	Canton Element	Chinese Restaurant
59	Westminster	0.014013	0	51.52587	-0.19526	Mayflower	Chinese Restaurant
59	Westminster	0.014013	0	51.52587	-0.19526	Gourmet Oriental	Chinese Restaurant
1	Barnet	0.012461	0	51.61568	-0.24511	The Good Earth	Chinese Restaurant
1	Barnet	0.012461	0	51.58918	-0.22805	Jun Peking Chinese Restaurant - 皇上皇	Chinese Restaurant
1	Barnet	0.012461	0	51.60104	-0.19401	Man Chui	Chinese Restaurant

Cluster 2 (Blue)

```
london_chinese_rest_cluster_1 = final_merged.loc[(final_merged['Cluster Labels'] == 1) & (final_merged['Venue Category'] == 'Chi
london_chinese_rest_cluster_1.drop_duplicates()
```

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
10	Brent, Ealing	0.125	1	51.53938	-0.25205	Good Taste	Chinese Restaurant

Cluster 3 (Green)

```
london_chinese_rest_cluster_2 = final_merged.loc[(final_merged['Cluster Labels'] == 2) & (final_merged['Venue Category'] == 'Chi
london_chinese_rest_cluster_2.drop_duplicates()
```

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
38	Kensington and Chelsea	0.003527	2	51.48563	-0.18144	New Culture Revolution, SW3	Chinese Restaurant
57	Waltham Forest	0.003891	2	51.55885	-0.00733	Fortune House Chinese Takeaways	Chinese Restaurant
49	Redbridge	0.003413	2	51.58977	0.03052	Wing Sing	Chinese Restaurant
16	City	0.004255	2	51.51841	-0.08815	Yauatcha	Chinese Restaurant
14	Camden	0.002488	2	51.51651	-0.11968	Canton Element	Chinese Restaurant
14	Camden	0.002488	2	51.51651	-0.11968	Kam Fung	Chinese Restaurant
24	Hackney	0.001667	2	51.55885	-0.00733	Fortune House Chinese Takeaways	Chinese Restaurant

Cluster 4 (Yellow)

```
london_chinese_rest_cluster_3 = final_merged.loc[(final_merged['Cluster Labels'] == 3) & (final_merged['Venue Category'] == 'Chi
london_chinese_rest_cluster_3.drop_duplicates()
```

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
56	Tower Hamlets	0.046908	3	51.51122	-0.01264	Sichuan Kitchen	Chinese Restaurant
56	Tower Hamlets	0.046908	3	51.52022	-0.05431	Sinh Le	Chinese Restaurant
56	Tower Hamlets	0.046908	3	51.52022	-0.05431	Tian Tian	Chinese Restaurant

Cluster 5 (Orange)

```
london_chinese_rest_cluster_4 = final_merged.loc[(final_merged['Cluster Labels'] == 4) & (final_merged['Venue Category'] == 'Chi
london_chinese_rest_cluster_4.drop_duplicates()
```

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
22	Greenwich	0.02649	4	51.48747	0.02795	Dragon & Phoenix	Chinese Restaurant
22	Greenwich	0.02649	4	51.48207	0.07143	Capital Noodle Bar	Chinese Restaurant

Discussion

In the cluster 2 that is the neighbourhood with the highest number of Chinese restaurants in London we can observe that a Chinese restaurant is the 5^{th} most common venue to be visited in the neighbourhood.

```
A Chinese Restaurant is the 5 th Most Common Venue in
 10    Brent, Ealing
```

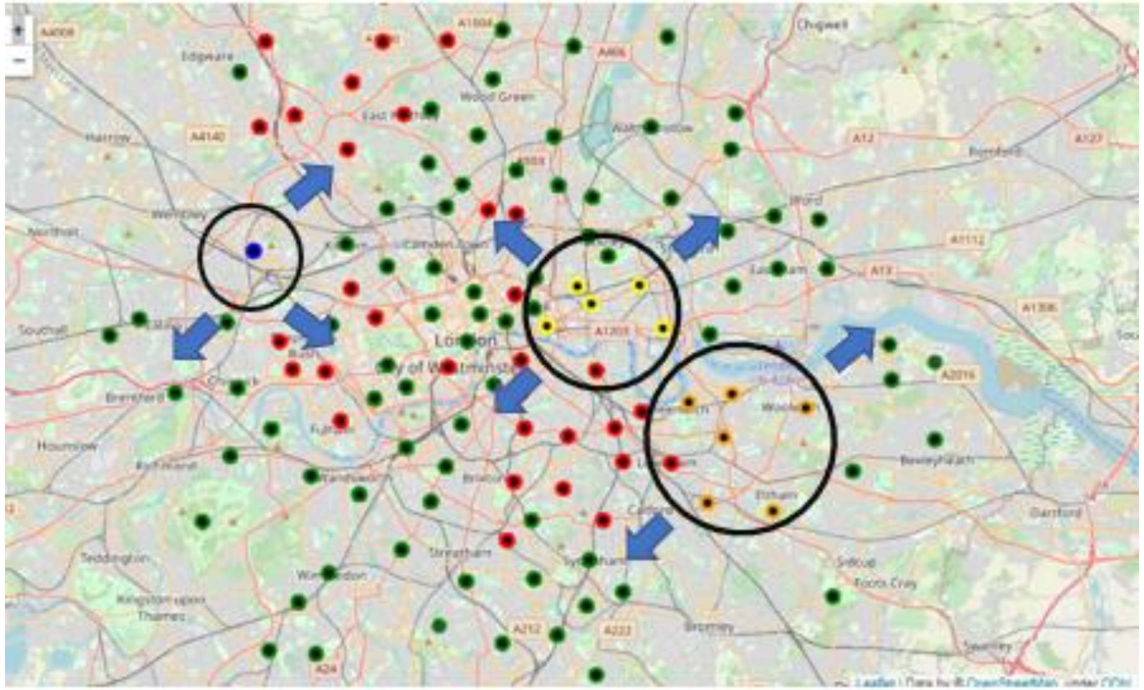
In the cluster 4 that is the neighbourhood with the highest number of Chinese restaurant in London we can observe that a Chinese restaurant is the 3^{rd} most common venue to be visited in the neighbourhood.

```
A Chinese Restaurant is the 3 rd Most Common Venue in
 56    Tower Hamlets
```

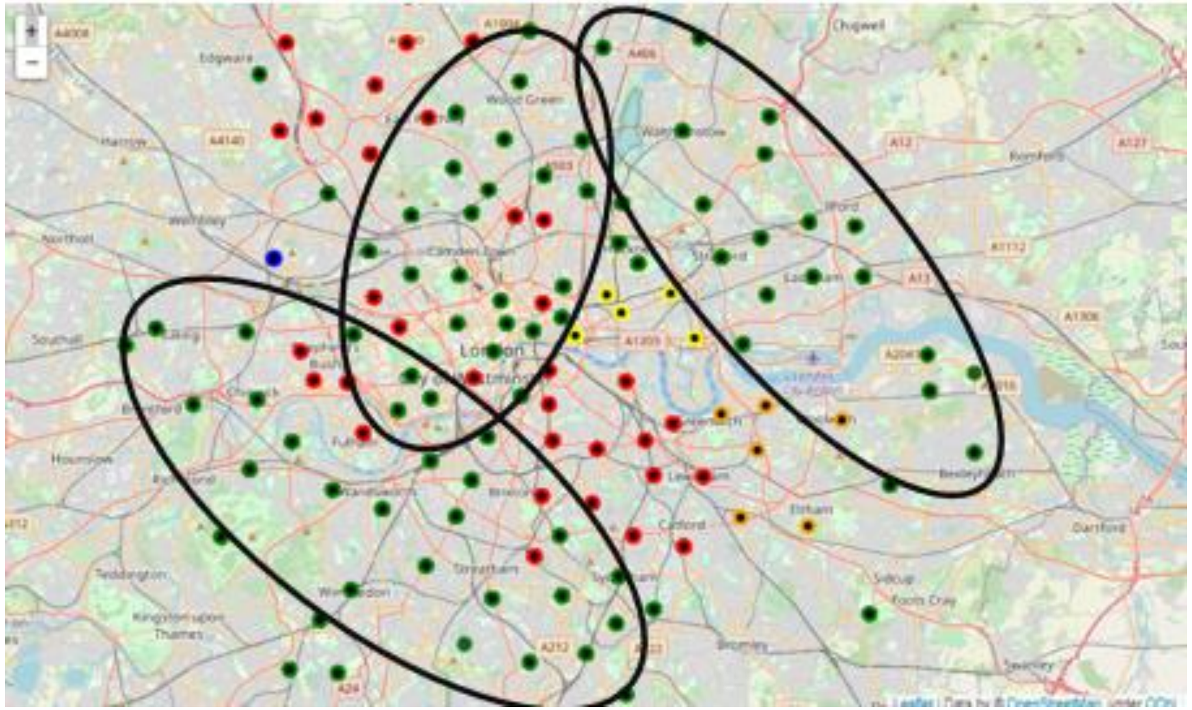
This information confirms the success of Chinese restaurant in these areas so this information can be very useful for tourists that are looking for a Chinese restaurants for his trip.

Evaluating the results of the clustering neighbourhoods in London depending on the number of Chinese restaurants in each neighbourhood we can observed that it exists a zone at southeast of London with two clusters that they having a high quantity of Chinese Restaurants (Cluster 4 and 5) and in the west of city there is the neighbourhood with the greatest number of Chinese restaurants.

The neighbourhoods near this neighbourhood (Brent (Ealing), Tower Hamlets and Greenwich) could be good opportunities to open a new Chinese restaurant.



If we carefully observe the London map with the colour clusters, we can explain another insight for someone that wants to open a new restaurant in London. There are three big zones which belong to the cluster number 3 where the number of Chinese restaurants is low or none.



Conclusion

After evaluating the results, we can conclude:

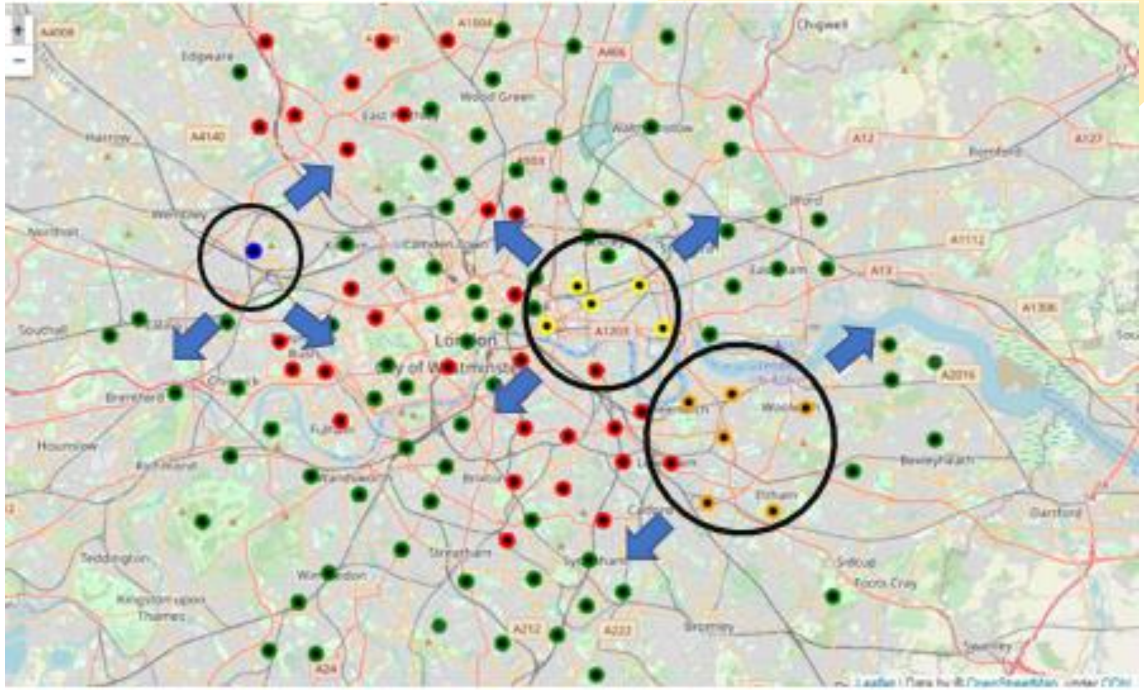
- The most number of Chinese restaurants are shared in the center of London.

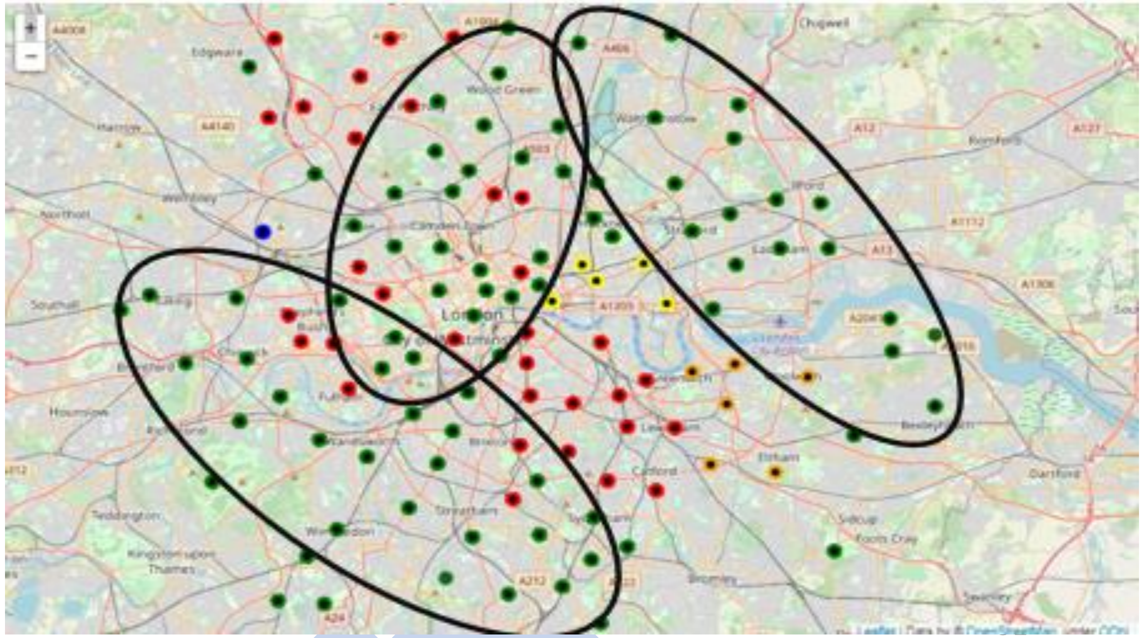
- The result map of London is useful to use by a tourist that want eat in a Chinese restaurant. There are three zones in London with a high number of Chinese restaurants Brent (Ealing), Tower Hamlets and Greenwich.

- For a person who want to open a new Chinese restaurant he have to consider two branches:

  1. Open a new restaurant near to Brent (Ealing), Tower Hamlets and Greenwich. Due to the success of Chinese restaurants in these neighbourhoods a new restaurant can attract a lot of people.

  2. If a fast-food chain want to open some Chinese restaurants it could be interested in the other model that is the following: Open restaurants where the number of restaurants are low or zero (cluster 3). People who lived here or tourist who have been accommodated on these zones will be the people interested in new Chinese restaurants in these zones.





Thanks for view my assignment