Capstone Project: The Battle of the Neighborhoods
Final Report

By: Abhay Saini; June 21st 2020

Contents

1. The Problem

1.1 Introduction/Background

London is a huge metropolis, probably the one of the oldest cities in continued existence. And since it is the capital of the United Kingdom, one of the first nations which stepped into the Modern Age, it is extensively mapped out and detailed.
There are two locations with the name London, one is the tiny City of London, the heart of the metropolis and the center of the administrative governance. The other is called “Greater London”, the bigger, wider area where London’s growing development snaked its tentacles to. We would be talking about “Greater London” in this project whenever we speak of London.
Its population is around 8.9 million and area is around 1,572 Square Kilometers. The official language of the city is English but because of England’s colonial history, an extremely diverse colonial diaspora has settled inside the city and consequently giving their own flavor to the beautiful city.

1.2 Problem to be Solved

The Problem that we will be going to solve in this Project will be “If someone were to open a new restaurant in London, which area or neighborhood, should he or she choose. The Stakeholders interested in the solution of this problem would be Prospective Entrepreneurs, City Planners and Franchise Owners wanting to Extend a Foreign Franchise into London.

---

2. Data

2.1 Data Sources

- I took data from primarily one source only. It was London Post Code Data from Doogal.co.uk https://www.doogal.co.uk/london_postcodes.php.
- It consisted of postcodes and other relevant data (Which I have trimmed to only relevant columns, the list of columns which will be used are mentioned below in image 1.0).
- The Analysis will be done at a Ward level, (a collection of postcodes). Since the total number of wards is around ~ 630 and the total area of Greater London is 1,572 sq.-km, average ward size came out to be ~2.2 sq.-km, a perfect size for a neighborhood)
- Secondly, I will use Foursquare Data to get Venue Categories, likes and comments. (The list of columns is mentioned below in image 1.1)

Postcode
Latitude
Longitude
District
Ward
District Code
Ward Code
Region
Index of Multiple Deprivation
Average Income
Distance to station

Postcode data

Venue Category
Likes
Comments
Users Reviews

Foursquare data

2.2 Description of the Postcard Data; -

1. Postcode: - Smallest and most basic unit of house identification.
2. Latitude: - Average Latitude of the postcodes of the Ward
3. Longitude: - Average Longitude of the postcodes of the Ward
4. District: - Consists of several Wards
5. Ward: - Collection of postcodes, clubbed together on some administrative commonality
6. Ward Code: - Unique Code of each Ward
7. Region: - East or North or West or North West etc.
8. Index of Multiple Deprivation: - Higher the index, more deprived that post code is
9. Average Income: - Income of several households averaged over the postcodes in one ward
10. Distance to Station (Tentative): - Distance to the nearest train station

---

3. Methodology

3.1 Approach

- Initially, the London postcode data was downloaded in the form of CSV files and some preliminary cleaning was done, to convert it into a Ward Level Data. Then, the file was read into the Python Notebook for processing.
- Then, some sanitary checks and EDA was done on the data to extract the relevant variables that will help in figuring out the best location for opening a new restaurant.
- After that, the foursquare location data was extracted using namely “explore” and “venue-id” commands and merged with the data frame above.
- Finally, the entire data was one hot encoded and aggregated, before clustering was done to create location clusters.

3.2 Analysis

3.2.1 Data Cleaning

Data cleaning was done at pre-analysis level while the data was in the CSV file. In that process: -

a. A total of ~150,000 non-functional post codes were removed.
b. Entire data was aggregated at Ward & District level separately through pivot tables and kept in two separate CSV files.
c. Except for population (which was summed), rest of the columns were averaged.

	Ward	Sum of Population	Sum of Households	Sum of Altitude	Average of Index of Multiple Deprivation	Average of Distance to station	Average of Latitude	Average of Longitude
0	Abbey	23016	9115	5623	15351.470320	0.359000	51.476602	-0.058257
1	Abbey Road	11091	5174	15292	24245.784620	0.348642	51.534391	-0.178132
2	Abbey Wood	15684	5958	2126	8847.297398	0.660415	51.490936	0.113496
3	Abingdon	9644	4672	4488	22257.260610	0.467441	51.497355	-0.196444
4	Acton Central	15424	6266	8417	12478.505450	0.383761	51.513191	-0.269339

Data Snapshot After Cleaning (Ward Level Only)

Postcode	In Use?	Latitude	Longitude	County	District	Ward	Ward Code	Population	Households	Rural/urban	Altitude	London zone	Index of Multiple Deprivation	Nearest station	Distance to stati
IG11 7AR	Yes	51.534399	0.07708	Greater London	Barking and Dagenham	Abbey	E05000026			Urban major conurbation	7	4	6900	Barking	0.6246
IG11 7FD	Yes	51.53561	0.080307	Greater London	Barking and Dagenham	Abbey	E05000026	72	29	Urban major conurbation	9	4	6900	Barking	0.4334
IG11 7FE	Yes	51.535481	0.08046	Greater London	Barking and Dagenham	Abbey	E05000026	39	18	Urban major conurbation	9	4	6900	Barking	0.4468
IG11 7FF	Yes	51.535297	0.080206	Greater London	Barking and Dagenham	Abbey	E05000026	19	7	Urban major conurbation	9	4	6900	Barking	0.4687
IG11 7FG	Yes	51.535428	0.079881	Greater London	Barking and Dagenham	Abbey	E05000026	41	18	Urban major conurbation	9	4	6900	Barking	0.4571
IG11 7FH	Yes	51.535882	0.080694	Greater London	Barking and Dagenham	Abbey	E05000026	15	7	Urban major conurbation	9	4	6900	Barking	0.4014
IG11 7FJ	Yes	51.535586	0.080104	Greater London	Barking and Dagenham	Abbey	E05000026	51	13	Urban major conurbation	9	4	6900	Barking	0.4376
IG11 7FL	Yes	51.536298	0.08005	Greater London	Barking and Dagenham	Abbey	E05000026	36	16	Urban major conurbation	9	4	6900	Barking	0.359

Data Snapshot before Cleaning

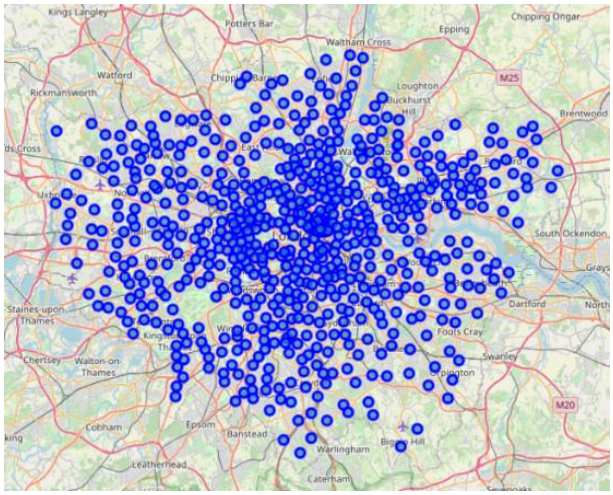
3.2.2 EDA (Exploratory Data Analysis)

Initially, the coordinates of London in our file were compared with geopy, with only a little difference in both. Then, the coordinates of all wards were plotted onto a folium map.

Averaged Coordinates : 51.504,-0.1215

Geopy Coordinates : 51.507, -0.1276

London Folium Map ->

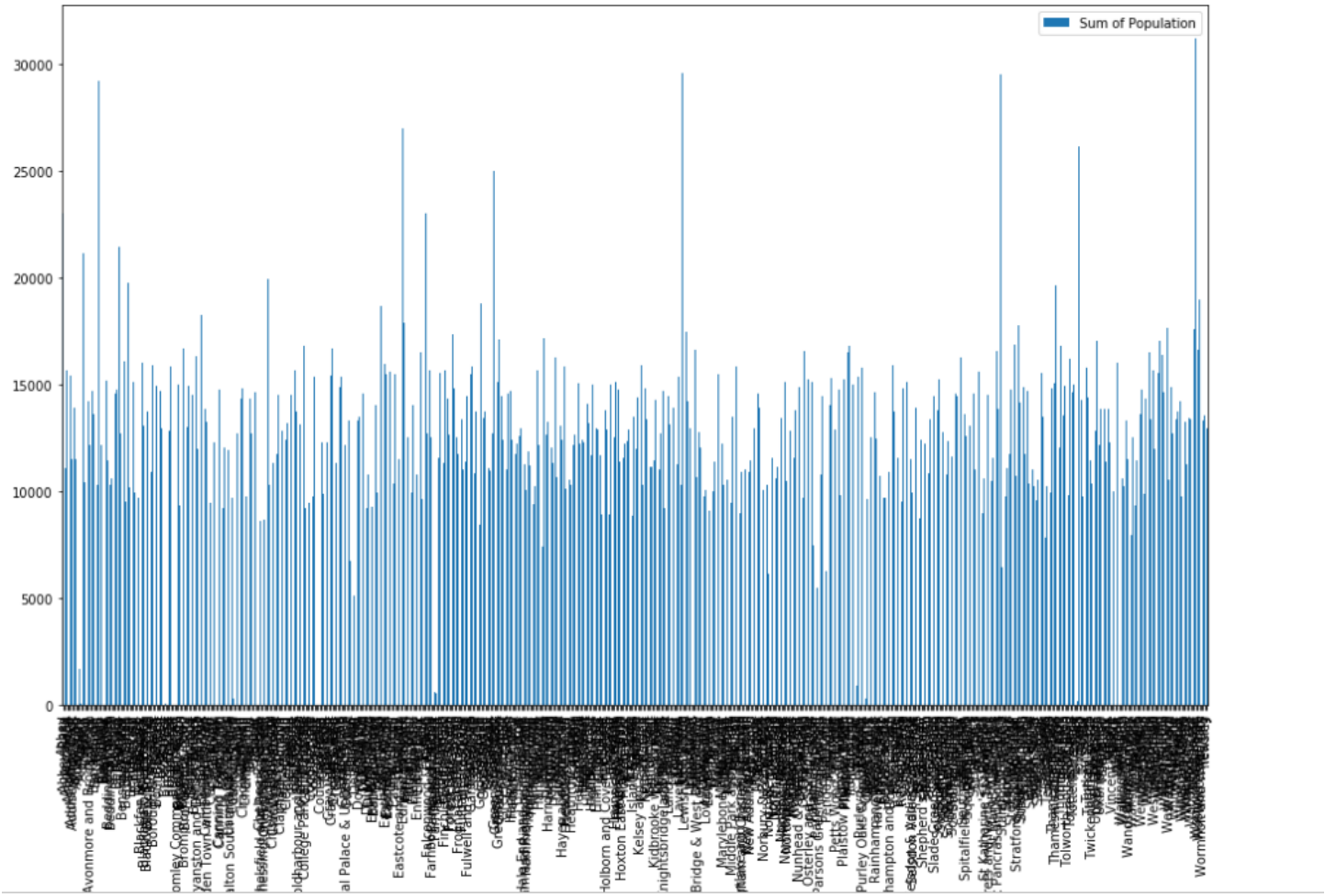


Even the population of London through our data came pretty close to the number in Wikipedia.

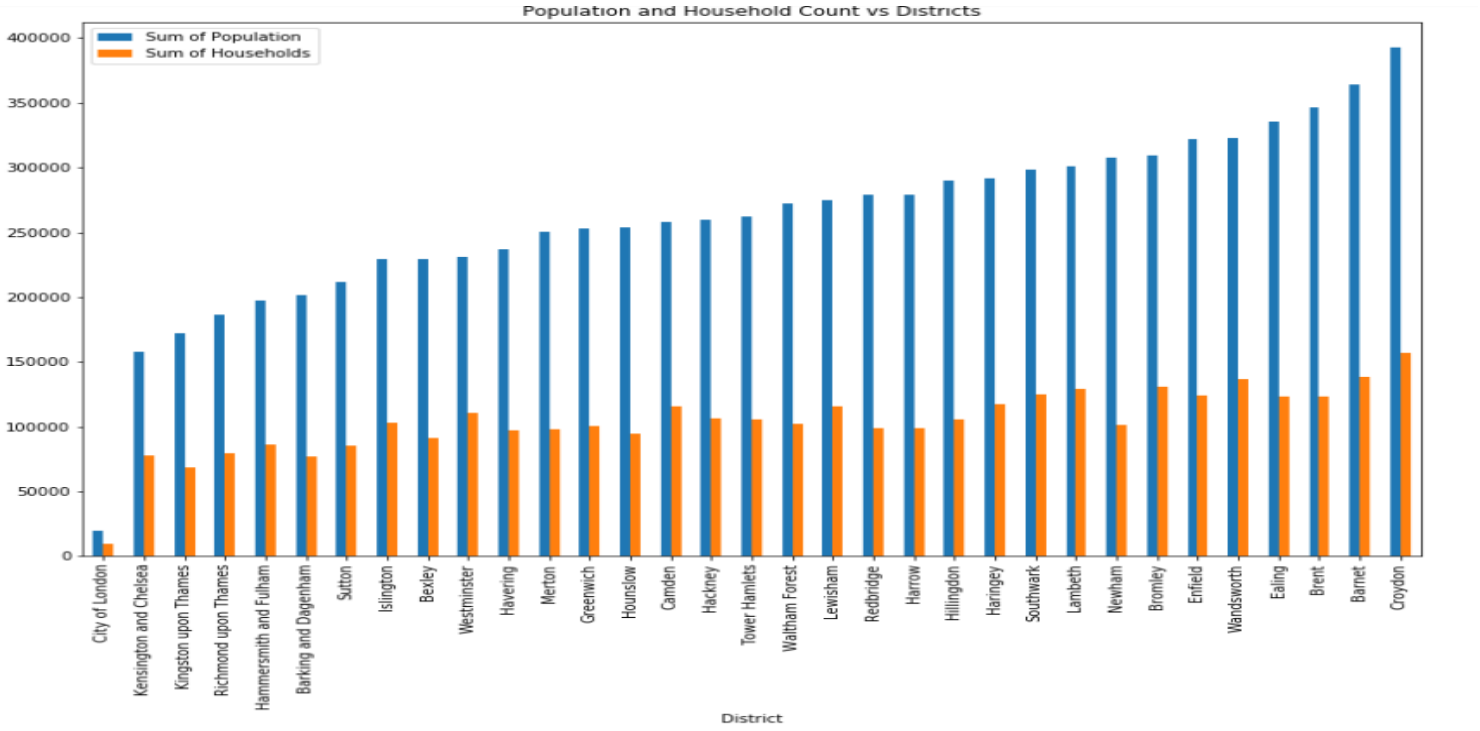
Population in Our Data: - 8,135,544

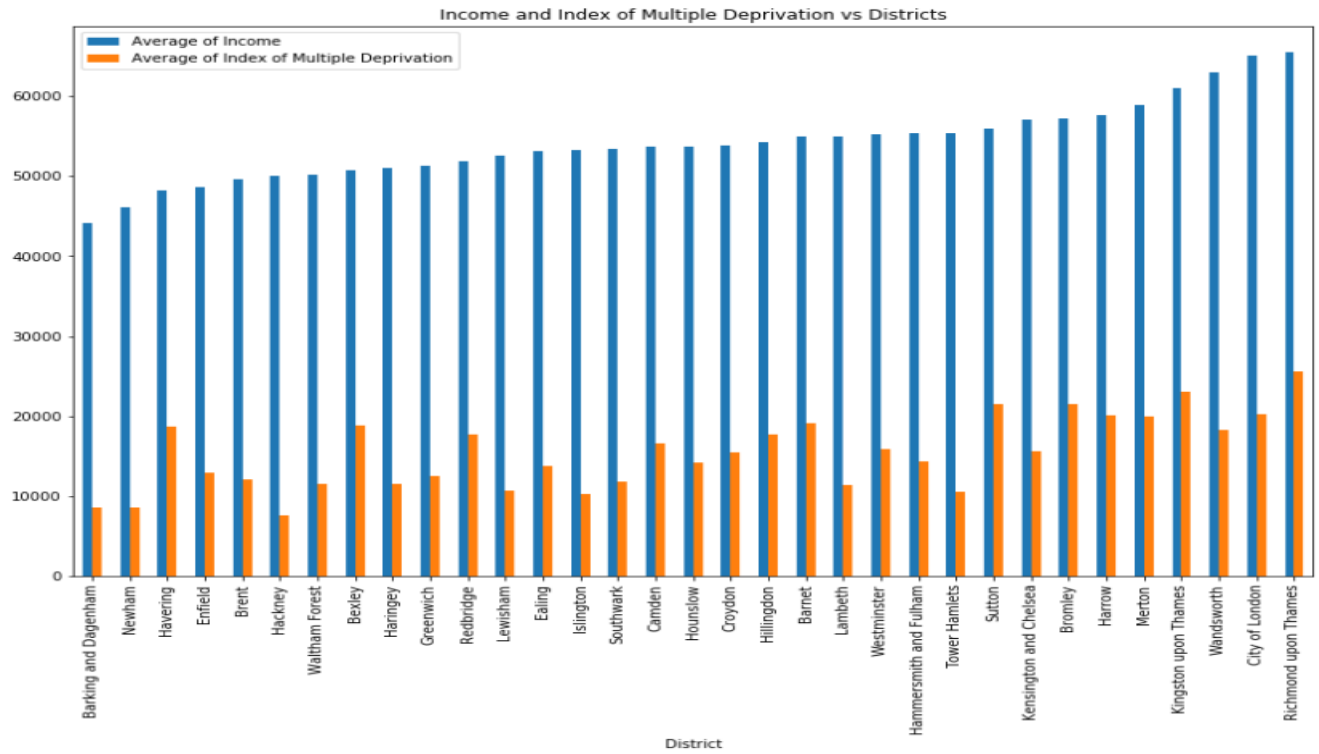
• City of London	1.12 sq mi (2.90 km^2)
• Greater London	606 sq mi (1,569 km^2)
Elevation[2]	36 ft (11 m)
Population (2018)[4]	
• Total[A]	8,908,081
• Density	14,670/sq mi (5,666/km^2)
• Urban	9,787,426
• Metro	14,257,962[3] (1st)
• City of London	8,706 (67th)
• Greater London	8,899,375

We, then analyzed the variables at a ward level, (Sum of Population is in the chart below)



However, it looked too crowded to be of any real help (except for the fact that most of wards have similar populations), so we simply changed direction and took a look at District level graphs.





Through the above charts, the decision was taken to include ‘Sum of Population’ and ‘Average Income’ in our analysis and exclude ‘Index of Multiple Deprivation’ as well as ‘Sum of Household Count’. We also decided to include ‘Distance to station’ as it was a completely separate variable altogether.
The correlation between the excluded variables and the included ones was calculated as well to confirm the hypothesis.

```
#Getting correlation between income and index of multiple deprivation as well as population and h
import numpy as np
print(np.corrcoef(df101['Sum of Population'], df101['Sum of Households']))
print(np.corrcoef(df101['Sum of Population'], df101['Count of Postcodes']))

print(np.corrcoef(df101['Average of Income'],df101['Average of Index of Multiple Deprivation']))
print(np.corrcoef(df101['Average of Income'],df101['Average of Distance to station']))
print(np.corrcoef(df101['Sum of Population'], df101['Average of Distance to station']))
```

```
[[1.         0.93531015]
 [0.93531015 1.        ]]
[[1.        0.3204964]
 [0.3204964 1.       ]]
[[1.         0.79266367]
 [0.79266367 1.        ]]
[[ 1.          -0.20129868]
 [-0.20129868  1.        ]]
[[1.         0.07777354]
 [0.07777354 1.        ]]
```

As we can see, ~ 94% and 79% correlation is found between population/households as well as income and index of multiple deprivation.

3.2.3 Data Preparation

This step involves three major steps, getting foursquare data, extracting the relevant parts of the data and then aggregating the with the original data frame prepared above.
For Foursquare data, two major queries were used, 1) EXPLORE 2) VENUE_ID/LIKES
In the first query, Foursquare credentials, Ward Latitude/longitude were added as parameters to get VENUE, VENUE_ID and VENUE_CATEGORY.
In the second query, only the unique VENUE_ID gotten above is incorporated as a parameter to get the number of likes a venue has gotten.

```
abhaysinghsaini@gmail.com:
CLIENT_ID: MAAYVXJ52QE1JYSBW
CLIENT_SECRET:1V4MMEQNBSIJHF
```

Credentials

```
# create the API request URL
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&clien
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    lat,
    lng,
    radius,
    LIMIT)

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

First Query

```
#practice
list_likes = []
for i in range(3176):
    venue_id = pd.Series(df31['Venue Name'].unique())[i]
    print(i)
     # create the API request URL
    url = 'https://api.foursquare.com/v2/venues/{}/likes?&client_id={}
    venue_id,
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    radius,
    LIMIT)
    # make the GET request
    results = requests.get(url).json()["response"]['likes']['count']
```

Second Query

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	Abbey	51.476602	-0.058257	Peckham Beer Rebellion	53f090cc498efa32861162a0	51.473612	-0.058636	Pub
1	Abbey	51.476602	-0.058257	Asylum Chapel	4e33163a227111ae768a86e0	51.479295	-0.060841	Art Gallery
2	Abbey	51.476602	-0.058257	Kudu	5a63310ca92d9807679ec024	51.473889	-0.059369	African Restaurant
3	Abbey	51.476602	-0.058257	Blackbird Bakery	53ea3ffd498e4b67a5a58878	51.473496	-0.057418	Bakery
4	Abbey	51.476602	-0.058257	Mamma Dough	573e14fc498e9c65d3550854	51.473456	-0.056323	Pizza Place
...	...	...	...	...	...	...	...	...
13265	Yeading	51.524316	-0.390686	B&Q	4bceead6ef109521d4bf8486	51.522648	-0.387927	Hardware Store
13266	Yeading	51.524316	-0.390686	Tesco Extra	4ba49d59f964a52081a738e3	51.524666	-0.385670	Supermarket
13267	Yeading	51.524316	-0.390686	Atlantis Fish & Chips	4e1c9a63b0fb543b8ff59212	51.525862	-0.384997	Seafood Restaurant
13268	Yiewsley	51.516236	-0.466221	Sea Life	502ac93de4b0b4f96235626d	51.516240	-0.472691	Fish & Chips Shop
13269	Yiewsley	51.516236	-0.466221	Candy's Cafe	5221e77a11d2f18f88a91fa9	51.513909	-0.472243	Café

13270 rows × 8 columns

Data Snapshot (After Running the First Query)

After data extraction, only a few venue categories were chosen (those which were relevant to restaurants) and they were all classified into broad super categories for ease of understanding. Finally, the data was one hot encoded (presence of a venue category in a neighborhood) and the counts were averaged.

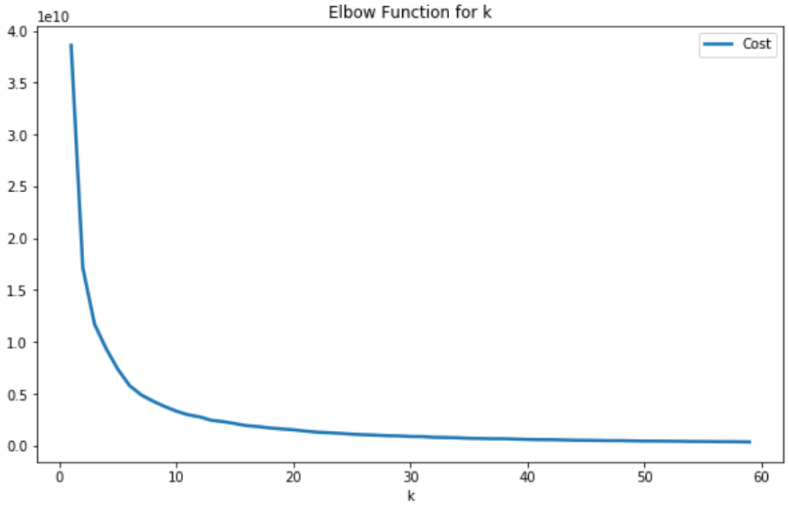
	Neighborhood	African	Anglo American	Asian	Continental European	Latin American	Middle Eastern	South Asian	Sum of Population	Average of Distance to station	Average of income	Count of Likes
0	Abbey	0.4	0.400000	0.200000	0.000000	0.0	0.000000	0.000000	23016	0.359000	58431.05023	20.400000
1	Abbey Road	0.0	0.333333	0.000000	0.166667	0.0	0.500000	0.000000	11091	0.348642	58690.76923	13.333333
2	Abbey Wood	0.0	0.000000	0.000000	0.000000	0.0	0.000000	1.000000	15684	0.660415	44023.42007	2.000000
3	Abingdon	0.0	0.346154	0.230769	0.230769	0.0	0.153846	0.038462	9644	0.467441	66320.00000	54.115385
4	Acton Central	0.0	0.666667	0.000000	0.000000	0.0	0.000000	0.333333	15424	0.383761	51656.00000	3.333333
...	...	...	...	...	...	...	...	...	...	...	...	...
494	Woolwich Common	0.0	0.000000	0.333333	0.000000	0.0	0.333333	0.333333	16641	0.853287	42761.16505	1.666667
495	Woolwich Riverside	0.0	1.000000	0.000000	0.000000	0.0	0.000000	0.000000	18970	0.395885	49334.09091	6.000000
496	Worcester Park	0.0	1.000000	0.000000	0.000000	0.0	0.000000	0.000000	11605	0.914041	59000.00000	0.000000
497	Wormholt and White City	0.0	0.000000	0.000000	1.000000	0.0	0.000000	0.000000	13294	0.672493	48223.56322	10.000000
498	Yeading	0.0	1.000000	0.000000	0.000000	0.0	0.000000	0.000000	13589	2.193932	50399.14163	2.000000

499 rows × 12 columns

Data Snapshot (After One Hot Encoding and Adding Super Category)

3.2.4 Clustering

Finally, we began to cluster the data based on the variables shown above. In clustering, the first step was to figure out the ideal 'k' through the elbow method.
Although, the curve flattens at k=40, it still is almost flat at k=20. However, k=20 is too large a number for clustering as it will be hard to understand from a business sense.
That's why 3 different Ks (5,10 and 20) were tried and the results compiled (discussed in the Results section)



```
# set number of clusters
from sklearn.cluster import KMeans
kclusters = 10
london_grouped_clustering = london_grouped_3.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(london_grouped_clustering)
# run k-means clustering
# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

Clustering code

4. Results

Clustering was done with 3 separate Ks (5,10,20) but the result that will be taken as final will be k=10 for understandability.

K = 10

	Lower Income		Middle Income					High Income		
Cluster Label	5	0	7	4	9	1	6	8	3	2
African	0%	1%	0%	1%	0%	2%	0%	0%	0%	0%
Anglo American	53%	46%	38%	45%	45%	38%	29%	37%	34%	31%
Asian	12%	18%	17%	13%	16%	14%	22%	25%	21%	28%
Continental European	11%	12%	13%	14%	19%	24%	22%	25%	27%	29%
Latin American	2%	1%	4%	1%	1%	3%	2%	1%	7%	2%
Middle Eastern	10%	7%	10%	9%	8%	9%	9%	4%	6%	3%
South Asian	12%	14%	18%	17%	11%	11%	15%	6%	4%	7%
Sum of Population	756,451	1,375,447	241,595	1,152,104	710,904	666,318	678,667	535,323	7,195	231,749
Average of Distance to statior	0.76	0.70	0.52	0.65	0.62	0.63	0.56	0.56	0.20	0.51
Average of income	$42,670	$ 48,176	$51,144	$ 52,762	$53,717	$59,002	$59,274	$65,061	$65,400	$71,594
Count of Likes	14	15	53	15	36	15	30	30	82	36
Average of Latitude	51.55	51.52	51.52	51.51	51.50	51.48	51.51	51.49	51.51	51.45
Average of Longitude	-0.02	-0.09	-0.17	-0.14	-0.16	-0.09	-0.20	-0.21	-0.09	-0.20

K = 5

Cluster Label	0	1	2	3	4
African	1%	0%	1%	1%	0%
Anglo American	44%	34%	50%	34%	36%
Asian	15%	21%	14%	20%	26%
Continental European	15%	27%	11%	23%	26%
Latin American	1%	7%	2%	3%	1%
Middle Eastern	8%	6%	10%	8%	4%
South Asian	16%	4%	13%	12%	6%
Sum of Population	2,412,367	7,195	1,662,771	1,554,275	719,145
Average of Distance to station	0.65	0.20	0.74	0.58	0.53
Average of income	$ 52,074	$ 65,400	$ 44,970	$ 58,946	$67,547
Count of Likes	22	82	14	29	31
Average of Latitude	51.51	51.51	51.53	51.50	51.47
Average of Longitude	-0.14	-0.09	-0.05	-0.16	-0.21

K = 20

Cluster Label	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
African	0%	0%	0%	2%	0%	0%	0%	0%	0%	0%	0%	1%	0%	0%	2%	1%	1%	6%	0%	2%
Anglo American	44%	40%	70%	52%	34%	34%	29%	34%	67%	32%	56%	38%	29%	27%	37%	41%	50%	52%	39%	41%
Asian	16%	28%	8%	17%	32%	21%	26%	17%	6%	23%	7%	15%	18%	30%	19%	14%	17%	13%	14%	16%
Continental European	19%	23%	1%	12%	14%	27%	27%	21%	4%	33%	6%	19%	14%	28%	21%	16%	10%	16%	22%	12%
Latin American	1%	0%	9%	1%	4%	7%	1%	3%	3%	2%	0%	1%	6%	1%	6%	1%	1%	0%	3%	1%
Middle Eastern	9%	4%	5%	10%	7%	6%	6%	8%	10%	3%	28%	8%	12%	5%	6%	7%	7%	6%	10%	8%
South Asian	11%	4%	7%	6%	10%	4%	11%	17%	10%	8%	3%	19%	21%	9%	7%	21%	14%	7%	11%	21%
Sum of Population	494,016	378,064	110,461	450,031	56,904	7,195	341,609	422,085	276,970	174,613	176,431	784,300	168,092	123,863	104,102	478,134	622,122	162,212	448,993	575,556
Average of Distance to station	0.69	0.53	0.79	0.72	0.62	0.20	0.59	0.58	0.71	0.51	0.75	0.66	0.49	0.51	0.46	0.75	0.68	0.60	0.61	0.64
Average of income	$53,632	$65,153	$41,145	$48,242	$56,606	$65,400	$61,349	$57,675	$50,354	$69,385	$40,300	$51,048	$49,913	$73,244	$56,938	$44,035	$47,444	$60,180	$59,483	$54,626
Count of Likes	32	29	26	19	33	82	23	33	12	38	7	16	75	25	76	16	10	14	16	14
Average of Latitude	51.50	51.48	51.55	51.52	51.53	51.51	51.49	51.52	51.52	51.47	51.60	51.52	51.55	51.46	51.49	51.53	51.52	51.49	51.49	51.49
Average of Longitude	-0.17	-0.19	-0.03	-0.10	-0.10	-0.09	-0.24	-0.17	-0.13	-0.25	-0.02	-0.13	-0.18	-0.15	-0.13	-0.03	-0.09	-0.16	-0.09	-0.11

5. Discussion

For k=10, the results could be divided into 3 separate collections of clusters.

Low Income Neighborhood

- High Traditional Anglo-American Places
- Less Ethnic Eating Places (Except for Middle Eastern)
- High Distance to Station
- Not very liked on Social Media either

Medium Income Neighborhood

- Comparatively higher number of Ethnic Eateries
- Cluster 7 and 4 are highest in South Asian cuisine and are well liked on Social Media (reflecting highly active South Asian population on Social Media). Cluster 7 is high in Latin American too
- 9 is almost like 4 with the exception of South Asian and Asian-European exchanging places
- 1 is a higher end Middle Income neighborhood with a drastic increase in Continent European
- 6 shows a drop in traditional Anglo American and a steep rise in Asian, along with Continental European

High Income Neighborhood

- Among the high income, 3 is unique in the sense that it has a low population and the highest Latin American % (7%)
- 2 has the highest Asian and Continental European as well as the highest income and is highest liked neighborhood in terms of likes)
- 8 has less ethnic (south Asian and Latin) varieties and fair number of others

Other Potential Variables/Approaches

- Police Coverage/Crime Rate
- Density of Restaurants
- Uber/Ola Service in the Neighborhood
- Age of the Residents
- Instead of Clustering, we could run a linear regression and predict the sales of a new restaurant in a particular neighborhood.

6. Conclusion

For opening a Working-Class Restaurant: -

- If it's Asian/South Asian, choose the 101 Neighborhoods in Cluster 0
- If it's generic/Anglo-American, choose the 55 Neighborhoods of Cluster 5

For opening Middle Class Restaurant: -

- If it's South Asian, choose the 9 Neighborhoods of Cluster 7th or the 76 Neighborhoods of Cluster 4
- If it's Continental European, choose the 41 Neighborhoods of Cluster 1
- If it's Asian, choose the 63 Neighborhoods of Cluster 6
- If it's Anglo-American, choose the 64 Neighborhoods of Cluster 9

For opening High End Restaurant: -

- If it's Asian or Continental European, Choose the 20 Neighborhoods of Cluster 2
- If it's Latin American, choose the 24 Neighborhoods of Cluster 3
- If it's a Generic, choose the 46 Neighborhoods of Cluster 8