Capstone Project: Prime coffee shop location in London

1. Introduction

The Problem: *Where should I open a new coffee shop in London?*

If I were interested in opening a new coffee shop in London - a bustling city with plenty of independent cafes and restaurants scattered over many neighbourhoods - where would the best location be? In this project, I'll answer this question using location data from the Foursquare API.

In order to find the perfect spot, we need to consider criteria that might be markers of a "good" location. One clear marker could be locations that have few existing coffee shops, which would minimise competition. However, locations might not have coffee shops for good reasons (lack of demand/passing trade, etc.). So we should probably consider some other factors too. Another marker of a good location could be proximity to universities — students like caffeine! Finally, we would want a good amount of passing trade, so proximity to retail and other attractions, such as museums and theaters, should also be considered. All in all, the following criteria should help prospective coffee shop owners to choose a good location:

1. Few existing coffee shops (low competition)
2. Near students (caffeine demand)
3. Near retail, museums or other attractions (passing trade)

In this project, I will highlight the best possible neighbourhoods by exploring and clustering locations within the central London area, using the above criteria to select a group of candidate locations.

2. Data

The data used for this project comes from Wikipedia (London locations) and the Foursquare API (information on surrounding venues). Together, the data include London neighbourhoods, with their latitude and longitude, as well as information about the number of coffee shops, universities, shops, and attractions for each location collected using Foursquare.

- London locations are scraped from this Wikipedia page: https://en.wikipedia.org/wiki/List_of_areas_of_London. The data will be filtered to find all of the neighbourhoods within the central London postal area.
- I'll use the Foursquare API to run a series of searches for each neighbourhood, collecting the number of venues within different search categories (coffee, universities, retail, entertainment) per location.

The Foursquare data will be used to cluster my London locations according to their venue distribution (i.e. the frequency of coffee shops, universities, retail, and entertainment venues that exist in each area).

3. Methodology

The following packages were used to format and display the data:

- Pandas
- Numpy
- Scipy
- Geopy and Geocoder
- Folium
- Matplotlib

3.1. London neighbourhoods

First, I retrieved the London locations I'm interested in as well as their longitude and latitude. I used pandas to import the London areas from Wikipedia (see **Data**). Next, I cleaned up the location data by restricting locations to those within the central London postal area. In total, this resulted in 56 central London locations.

Now that I have my neighbourhoods, I fetched the longitude and latitude of each London location using the geocoder package. Here is the function to get coordinates from an address:

```
def get_latlng(loc,pc):

    lat_lng_coords = None

    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, London, {}, United Kingdom'.format(loc,pc))
        lat_lng_coords = g.latlng
        lat_lng_coords.append(loc) #add location to output
    return lat_lng_coords
```

I checked that the function returns the correct coordinates by using reverse coding. So, if provided with the coordinates returned from the above function, does geocoder give me back the correct address? Yes!

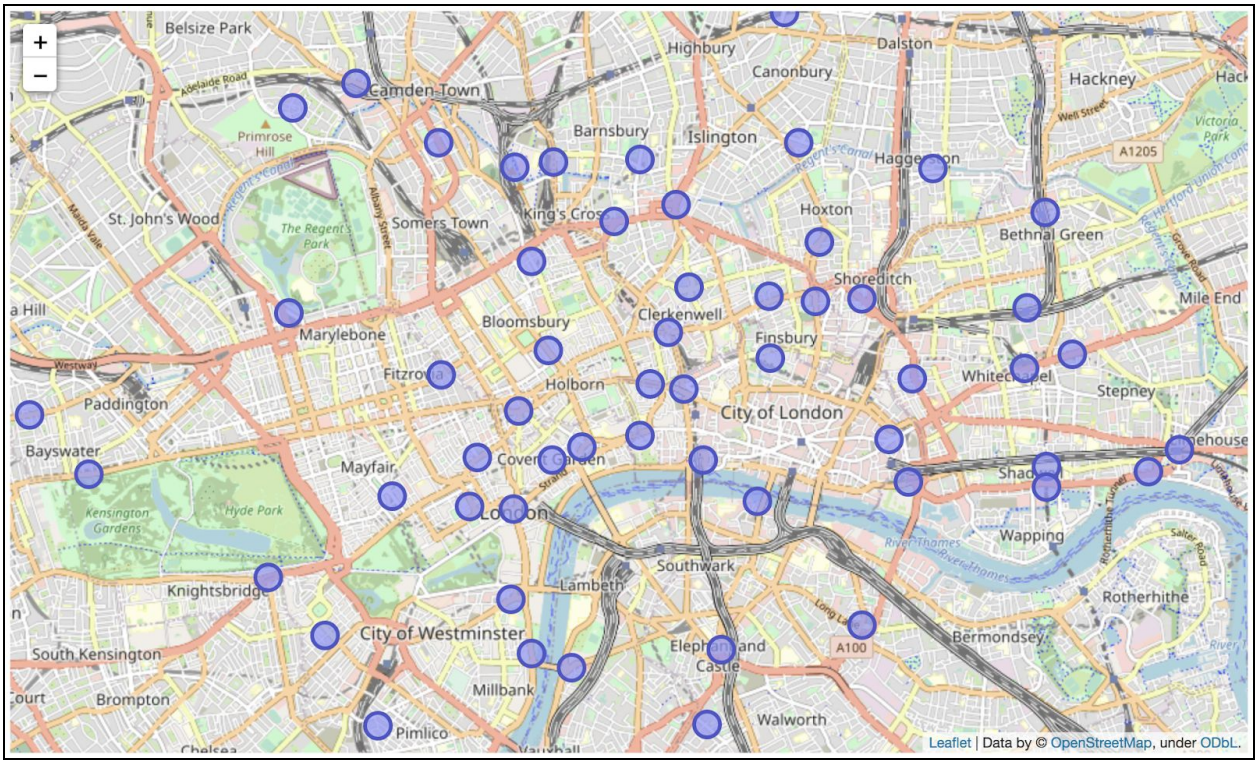
```
Reverse coding: Aldgate, EC3; coordinates = [51.51331168804641, -0.07776472785585485]

<[OK] Geocodefarm - Reverse [Leadenhall Street, Aldgate, EC3A 3DE, United Kingdom]>
```

Next, I applied this function to all of my London locations, ending up with a dataframe. Each row contains a postcode with its longitude and latitude values. Here are the top 10 rows of the resulting dataframe:

	Latitude	Longitude	Location
0	51.513312	-0.077765	Aldgate
1	51.512653	-0.118607	Aldwych
2	51.524730	-0.087540	Angel
3	51.508164	-0.095216	Bankside
4	51.520070	-0.093530	Barbican
5	51.536489	-0.110907	Barnsbury
6	51.510480	-0.184260	Bayswater
7	51.497050	-0.152750	Belgravia
8	51.497900	-0.081440	Bermondsey
9	51.524190	-0.059440	Bethnal Green

And here's a map of all of my locations, generated using folium.



3.2. Foursquare data

Now that I have my London locations, I set up the Foursquare API to obtain data about venues within each neighbourhood. To do this, I loaded in my credentials and specified relevant search terms and parameters. I limited each search result to a maximum of 50 venues to limit API calls, and I'm using a small radius of 500m for coffee shops, due to the high number, but a radius of 1km for other venue categories. For privacy, my Foursquare credentials were loaded in from a file called *foursquare_id.py*.

Here are the search terms that I used within Foursquare to gather information about my London locations:

["Coffee", "Colleges", "Shopping", "Fun"]

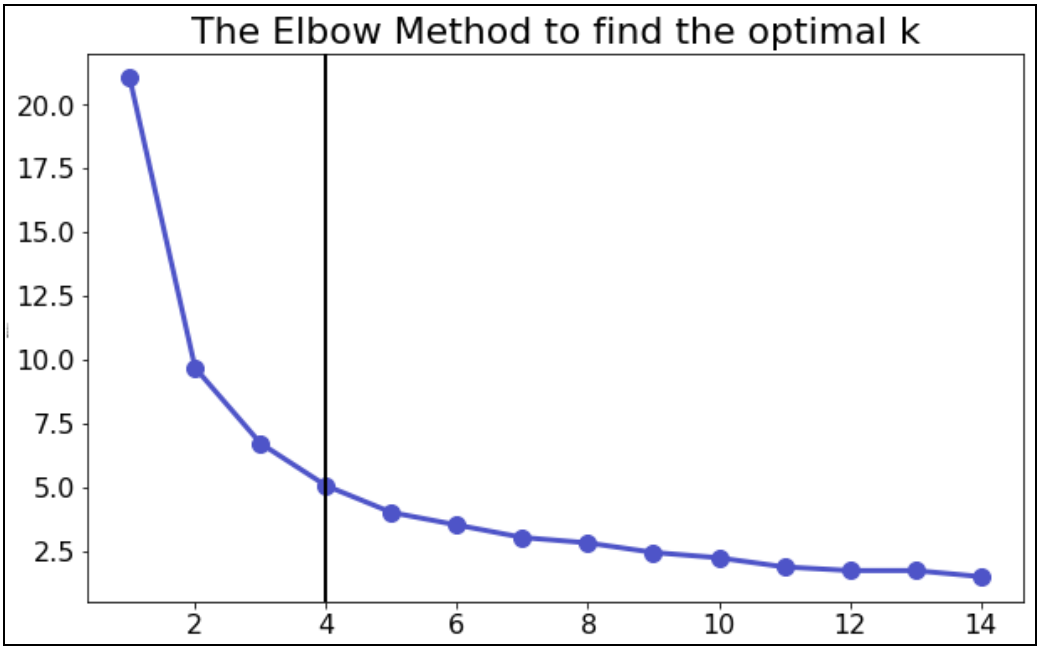
I looped over these query terms, saving out the number of venues returned for each term and each location. After returning the number of each venue category for each location, I normalized each category across locations to be between 0 and 1 so that the scales were comparable for later clustering. Here are the first 15 locations in the resulting data frame:

Category	Location	Latitude	Longitude	Coffee	Colleges	Fun	Shopping
0	Aldgate	51.513312	-0.077765	0.714286	0.000000	0.461538	0.000
1	Aldwych	51.512653	-0.118607	0.795918	1.000000	0.961538	0.625
2	Angel	51.524730	-0.087540	0.693878	0.166667	0.769231	0.125
3	Bankside	51.508164	-0.095216	0.530612	0.333333	0.346154	0.250
4	Barbican	51.520070	-0.093530	0.530612	0.166667	0.653846	0.375
5	Barnsbury	51.536489	-0.110907	0.102041	0.000000	0.307692	0.500
6	Bayswater	51.510480	-0.184260	0.061224	0.166667	0.115385	0.000
7	Belgravia	51.497050	-0.152750	0.142857	0.000000	0.269231	0.500
8	Bermondsey	51.497900	-0.081440	0.142857	0.166667	0.076923	0.000
9	Bethnal Green	51.524190	-0.059440	0.183673	0.000000	0.153846	0.125
10	Blackfriars	51.511600	-0.102408	0.387755	0.500000	0.500000	0.375
11	Bloomsbury	51.520740	-0.123100	0.714286	1.000000	1.000000	0.375
12	Cambridge Heath	51.532180	-0.056950	0.183673	0.000000	0.115385	0.000
13	Camden Town	51.537899	-0.137591	0.306122	0.833333	0.384615	0.250
14	Canonbury	51.548680	-0.091750	0.020408	0.000000	0.038462	0.125

3.3 Clustering

Finally, now that I have data about 56 central London locations and their venues from the Foursquare API, I clustered them. To do this, I used KMeans clustering from sklearn to group the locations based on their frequency of coffee shops, attractions, shopping, and universities.

I chose the number of clusters - *k* - based on the **Elbow Method**:



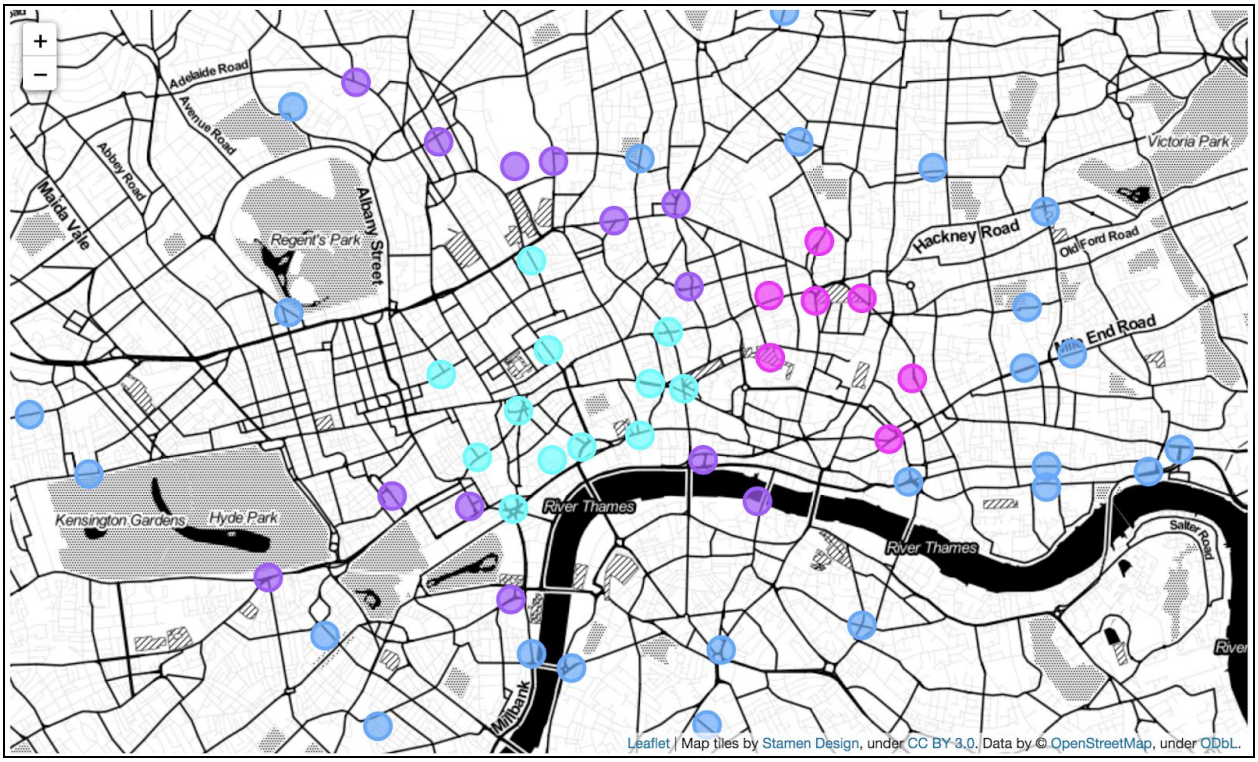
The above plot suggests that 4 clusters (at the elbow) is best for our data. So, I ran KMeans clustering searching for 4 clusters of central London locations, labelling each location according to its assigned cluster ID:

Category	Cluster	Location	Latitude	Longitude	Coffee	Colleges	Fun	Shopping
0	3	Aldgate	51.513312	-0.077765	0.714286	0.000000	0.461538	0.000
1	0	Aldwych	51.512653	-0.118607	0.795918	1.000000	0.961538	0.625
2	3	Angel	51.524730	-0.087540	0.693878	0.166667	0.769231	0.125
3	2	Bankside	51.508164	-0.095216	0.530612	0.333333	0.346154	0.250
4	3	Barbican	51.520070	-0.093530	0.530612	0.166667	0.653846	0.375
5	1	Barnsbury	51.536489	-0.110907	0.102041	0.000000	0.307692	0.500
6	1	Bayswater	51.510480	-0.184260	0.061224	0.166667	0.115385	0.000
7	1	Belgravia	51.497050	-0.152750	0.142857	0.000000	0.269231	0.500
8	1	Bermondsey	51.497900	-0.081440	0.142857	0.166667	0.076923	0.000
9	1	Bethnal Green	51.524190	-0.059440	0.183673	0.000000	0.153846	0.125

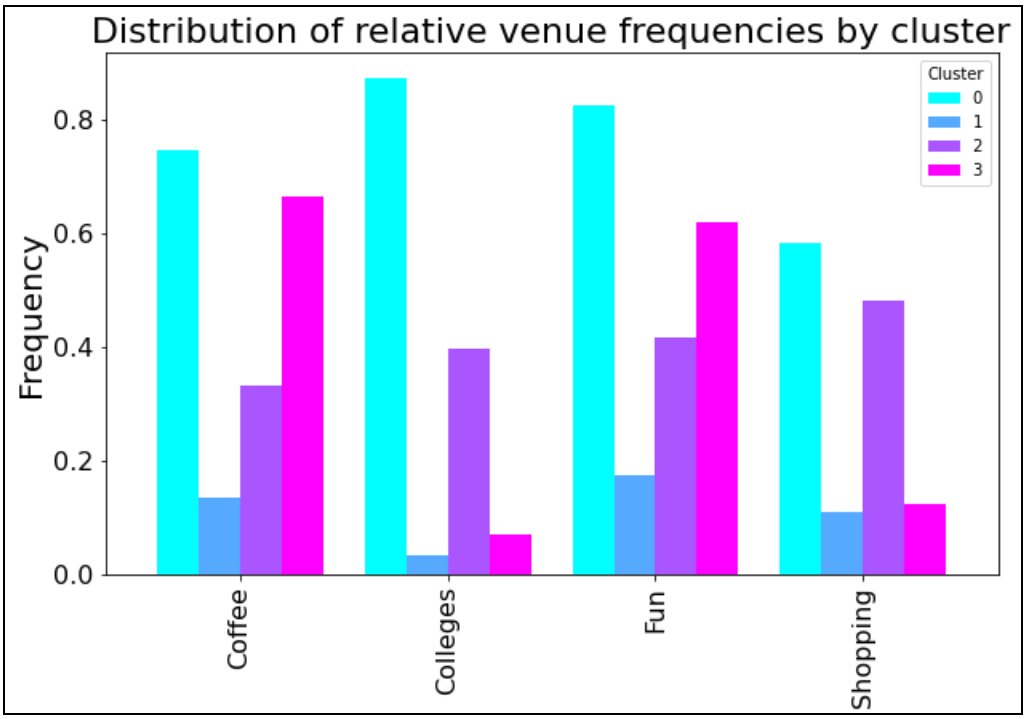
4. Results

The above Methodology describes how data was obtained for 56 central London locations and their nearby venues, using Wikipedia and the Foursquare API, and how those locations were clustered based on the type of nearby venues. Here, I'm presenting the results of the location clustering.

First, I re-plotted the London locations, but with each location colour-coded according to its cluster number. I used the Stamen Toner map for ease of seeing the cluster colours:

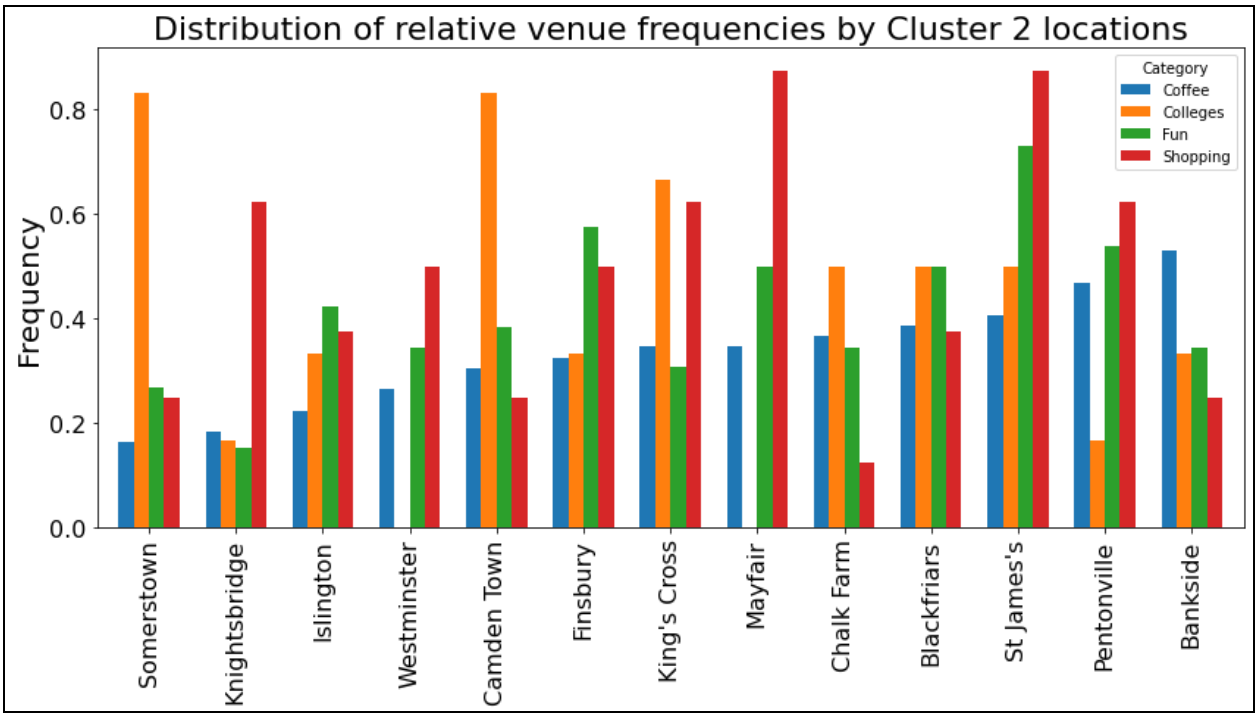


Next, I grouped the data by cluster and calculated the mean frequency of each venue category per cluster in order to compare the distributions:



From the above bar graph, it looks like cluster '2' is the best cluster to further explore — there aren't too many coffee shops already, and there's also a lot of attractions/shopping and reasonable proximity to

student populations, so perhaps there's a good location that we can find here? To answer this question, I plotted the frequency of venues within the individual cluster 2 locations:



A few possible locations jump out from the above bar graph:

- Camden Town - not too many coffee shops, some attractions, lots of students
- Kings Cross - not too many coffee shops, near students and shopping
- St. James's - students, and lots of attractions and shopping close by, but 3rd highest coffee shop count in cluster

5. Conclusion

Overall, this project aimed to find optimal locations within central London for opening a new coffee shop. To do so, I considered a potentially "good" location as one that had relatively few coffee shops already, but had proximity to a student population as well as attractions and retail.

Using clustering with Foursquare venue data, I found 4 clusters of London locations, with one particular cluster containing locations that, on average, had relatively few coffee shops already but reasonable proximity to students and entertainment. Within this cluster, 3 possible locations stand out to me as being the best: Camden Town, due to lack of existing competition and lots of students; King's Cross, with limited coffee shops and near to students and shopping; St James's, which has some students and lots of attractions close by.

Of course, other factors would need to be considered in order to fully specify the best location (e.g. unit availability, rent, and other overheads). Also, understanding the nearby residential population would be helpful.