The Battle of Neighborhoods

Applied Data Science Capstone Project

Akshay Kumar S

Introduction:

This project aims to mimic the expertise of a local property agency situated in any part of the world. Utilizing the ample number of tools available to a Data Scientist, we will be providing expert advice on property rent rates and locality descriptions for would-be clients and match them with their ideal apartment.

For the sake of simplicity, This project deals with neighborhoods in London, UK. However, the analytical techniques used here can be utilized to replicate the same project in any locality.

Our primary targets, when analyzing neighborhoods in London are the following:

- Rent distribution across neighborhoods,
- Details of venues in each neighborhood.

Data Requirements:

The data on available apartments are collected by scraping a local website with apartment listings (https://www.your-move.co.uk). We clean up the values and add a column for numerical data for Rent(ie.Price) The data is then processed to get our first dataframe:

	Postal Code	Area	Street	Price	Raw_Price
0	DA14	Sidcup	Frognal Avenue	£3,593 pcm	3593
1	DA14	Sidcup	Frognal Avenue	£3,176 pcm	3176
2	SE10	London	Cutter Lane Chandlers Avenue	£2,990 pcm	2990
3	SE10	London	Telcon Way	£2,400 pcm	2400
4	SE5	London	Lettsom Street	£2,150 pcm	2150
5	BR1	Bromley	Plaistow Lane	£2,100 pcm	2100
6	SE9	London	Grove Place	£2,000 pcm	2000
7	TW20	Egham	The Hub Stoneylands Road	£2,000 pcm	2000
8	TW20	Egham	Egham Hill	£1,950 pcm	1950

It is further cleaned by removing Price values deviating from a monthly format and by removing duplicate values for Postal Code. The data contained information about places outside London as well, so a regex capture group was used to selectively extract Areas within London.

Next step was to obtain latitude and longitude for each Neighborhood in Dataset. Although geocoder could be used to obtain Latitude and Longitude values for each postal code, upon a rudimentary web search, all necessary information was readily available on doogal.co.uk.

Data Requirements:

The final data set which was used for analysis:

	Postal Code	City	Neighborhood	Price	Raw_Price	Latitude	Longitude
0	DA14	Sidcup	Frognal Avenue	£3,593 pcm	3593	51.426481	0.105038
1	SE10	London	Cutter Lane Chandlers Avenue	£2,990 pcm	2990	51.487122	0.003284
2	SE5	London	Lettsom Street	£2,150 pcm	2150	51.481796	-0.090377
3	BR1	Bromley	Plaistow Lane	£2,100 pcm	2100	51.401546	0.015415
4	SE9	London	Grove Place	£2,000 pcm	2000	51.450352	0.062336
5	SE14	London	Childeric Road	£1,900 pcm	1900	51.474867	-0.046933
6	E11	London	New Wanstead	£1,800 pcm	1800	51.564032	0.008813
7	SW15	London	Sherfield Gardens	£1,600 pcm	1600	51.464123	-0.222848
8	SE4	London	Adelaide Avenue	£1,450 pcm	1450	51.453268	-0.032498
9	E7	London	Disraeli Road	£1,400 pcm	1400	51.554412	0.019655
10	IG11	Barking	East Street	£1,400 pcm	1400	51.524785	0.078646
11	SE16	London	Jamaica Road	£1,395 pcm	1395	51.491797	-0.042421
12	SE26	London	Sydenham Road	£1,375 pcm	1375	51.429347	-0.044847
13	BR2	Bromley	Sandford Road	£1,350 pcm	1350	51.407775	-0.003321
14	SE21	London	Kingswood Estate	£1,350 pcm	1350	51.434204	-0.085226
15	SE15	London	Shurland Gardens Willowbrook Estate	£1,350 pcm	1350	51.476810	-0.068163
16	SE28	London	Anson Place	£1,300 pcm	1300	51.492037	0.087067
17	SE3	London	Coleraine Road	£1,300 pcm	1300	51.467676	0.018169
18	E6	London	High Street South	£1,250 pcm	1250	51.537580	0.040295
19	RM1	Romford	South Street	£1,200 pcm	1200	51.484461	0.084733
20	SE18	London	Elmley Street	£1,200 pcm	1200	51.577600	0.178261
21	E17	London	Hoe Street	£1,100 pcm	1100	51.585503	-0.018014
22	RM2	Romford	Kidman Close	£950 pcm	950	51.577913	0.190919

Now with all necessary data accumulated, we can begin our analysis.

Methodology:

Our process to achieve our targets in this project is as follows:

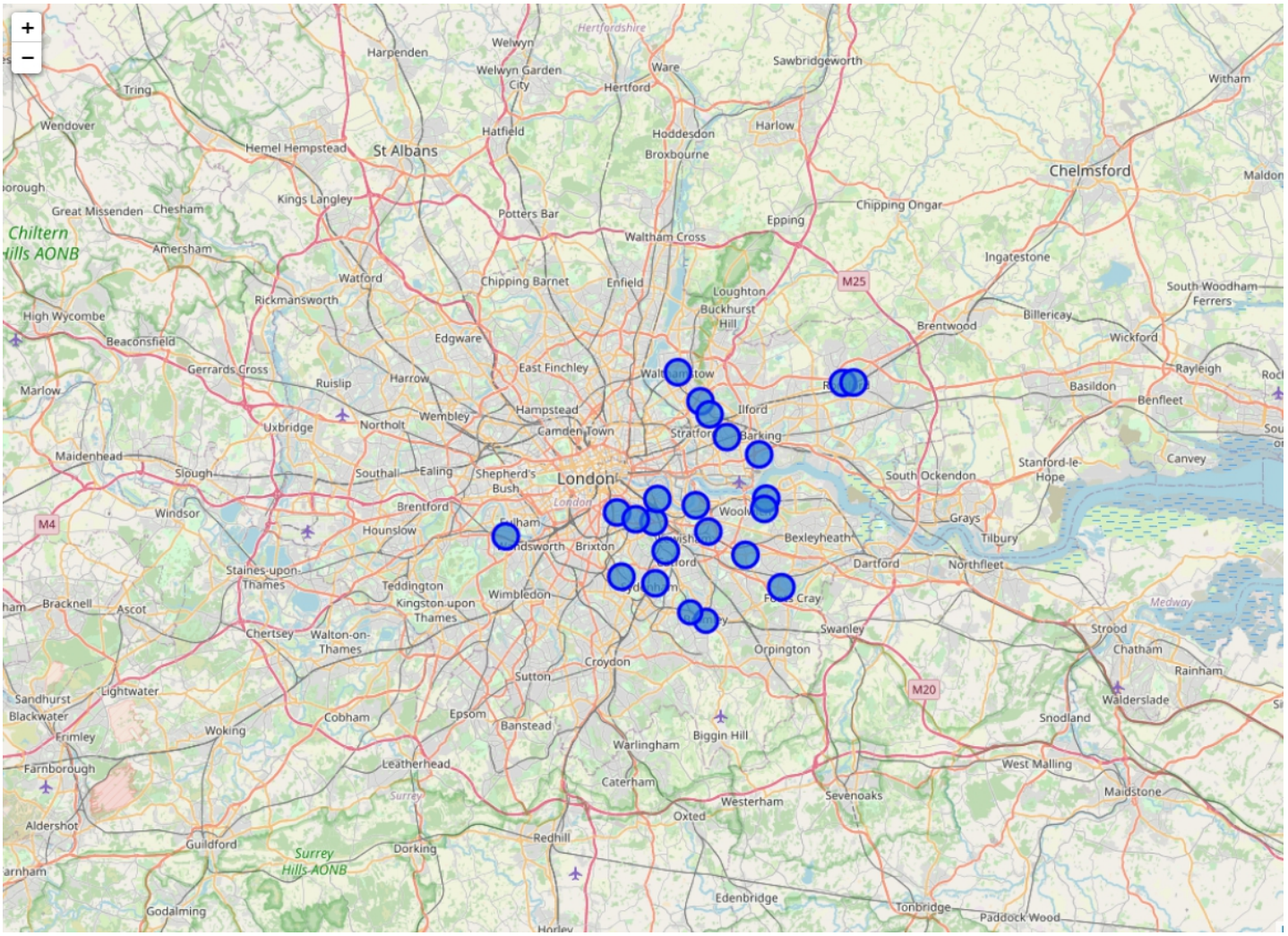
1. Identify neighborhoods on a leaflet map of London.
2. Collect data of all venues in each neighborhood
3. Segment the data into meaningful clusters
4. Conduct detailed cluster analysis
5. Conduct detailed Neighborhood analysis
6. Showcase the data using Folium,Bar and Bubble Charts
7. Formulate a conclusion using discovered relationship between location, rent and venue count within the neighborhood.

Analysis:

As mentioned before, a data scientist has access to multiple tools to facilitate his analysis.

One such tool is **Folium** (Folium — Folium 0.12.1 documentation (python-visualization.github.io)**.**

Folium let's us plot our neighborhood locations on an interactive leaflet map. We simply feed relevant information from our data set into our customized Folium code to get the following plot:



The blue markers cover a 1km radius around our chosen neighborhoods. These regions cover the neighborhoods that pop up most frequently on their website.

Now the first problem statement we will tackle is the following:

What venues are available within each of these zones ?

Analysis:

The task of obtaining real world data of each and every venue located within each of the marked zones is made possible with the use of **Foursquare API** (www.Foursquare.com).

Foursquare API can be called with our credentials by feeding it the latitude and longitude data of each neighborhood which we had collected beforehand. The API will return information pertaining to all the venues in that particular location in JSON format.

After cleaning, rearranging, grouping and classifying our data (All of which can be found in the Python Notebook for this project). We can find the **Top 10 Venue Categories For Each Neighborhood** based on their number of occurrences in each zone.

	Neighborhood	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
0	Adelaide Avenue	Coffee Shop	Fried Chicken Joint	Chinese Restaurant	Café	Bus Stop	Concert Hall	Malay Restaurant	Pub	Convenience Store	Theater
1	Anson Place	Train Station	Asian Restaurant	Breakfast Spot	Bus Station	Fish & Chips Shop	Furniture / Home Store	Fried Chicken Joint	Food Truck	Flower Shop	Yoga Studio
2	Childeric Road	Bus Stop	Platform	Convenience Store	Pub	Chinese Restaurant	Supermarket	Kebab Restaurant	Furniture / Home Store	Fried Chicken Joint	Electronics Store
3	Cutter Lane Chandlers Avenue	Grocery Store	Pier	Pub	Turkish Restaurant	Coffee Shop	Indian Restaurant	Pizza Place	Café	Japanese Restaurant	Mediterranean Restaurant
4	Disraeli Road	Bakery	Bar	Pub	Restaurant	Flower Shop	Food Truck	Yoga Studio	Fried Chicken Joint	Donut Shop	Eastern European Restaurant
5	East Street	Diner	Eastern European Restaurant	Soccer Field	Multiplex	Dim Sum Restaurant	Discount Store	Donut Shop	Electronics Store	English Restaurant	Ethiopian Restaurant
6	Elmley Street	Coffee Shop	Clothing Store	Café	Pub	Grocery Store	Fast Food Restaurant	Shopping Mall	American Restaurant	Bookstore	Italian Restaurant
7	Frognal Avenue	Pharmacy	Grocery Store	Coffee Shop	Fast Food Restaurant	Pizza Place	Café	Italian Restaurant	Supermarket	Hotel	Bakery
8	Grove Place	Mediterranean Restaurant	Department Store	Italian Restaurant	Portuguese Restaurant	Pub	Clothing Store	Movie Theater	Fast Food Restaurant	Pharmacy	Supermarket
9	High Street South	Grocery Store	Market	Metro Station	Park	Pub	Sandwich Place	Fast Food Restaurant	Soccer Field	Pharmacy	Bakery
10	Hoe Street	Coffee Shop	Pub	Sandwich Place	Restaurant	Clothing Store	Pharmacy	Bakery	Pizza Place	Café	Portuguese Restaurant
11	Jamaica Road	Pizza Place	Pharmacy	Supermarket	Pub	Vietnamese Restaurant	Platform	Gym	Clothing Store	Coffee Shop	American Restaurant
12	Kidman Close	Clothing Store	Pub	Multiplex	Sandwich Place	Park	Supermarket	Furniture / Home Store	Coffee Shop	Optical Shop	Video Game Store
13	Kingswood Estate	Garden Center	Bakery	Train Station	Gift Shop	Furniture / Home Store	Donut Shop	Eastern European Restaurant	Electronics Store	English Restaurant	Ethiopian Restaurant
14	Lettsom Street	Park	Middle Eastern Restaurant	Coffee Shop	Café	Pub	Ethiopian Restaurant	Building	Garden	Italian Restaurant	Fish & Chips Shop
15	New Wanstead	Grocery Store	Coffee Shop	Café	Pub	Pizza Place	Fast Food Restaurant	Fish & Chips Shop	Dim Sum Restaurant	Metro Station	Thai Restaurant
16	Plaistow Lane	Clothing Store	Coffee Shop	Gym / Fitness	Supermarket	Café	Chocolate Shop	Pub	Portuguese	Pharmacy	Park

So if a client kept proximity to a pub as his requirement, we could direct him to Neighborhoods ‘Kidman Close’ and ‘Hoe Street’ but what if he wanted availability of a gym nearby as well ? or if he has a vehicle and can easily move between zones, then which neighborhood should he stay in to maximize venue count ?

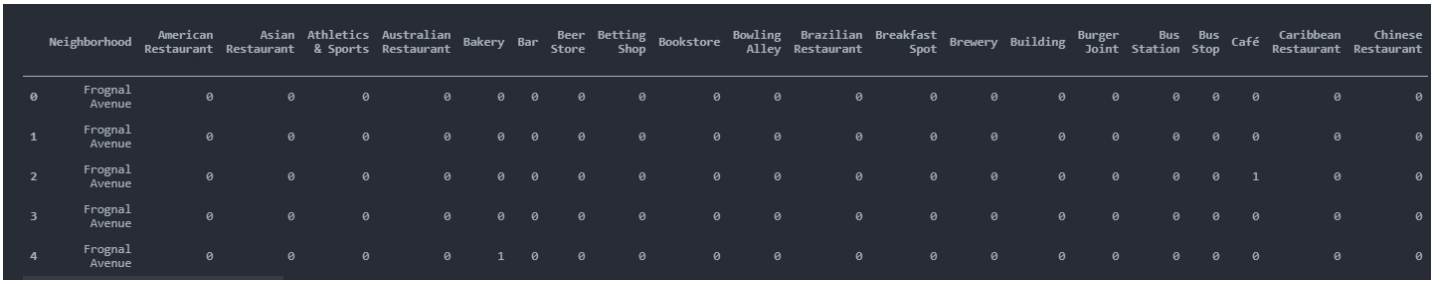
We can see that not only is the table difficult to read, it is also not giving us much inferences to work with either.

One reason for that would be the localized nature of the values in the table. A work-around for such an issue would be to cluster the neighborhoods based on similarities. Since the data we have is unlabeled for such clustering operations we will utilize a machine learning technique knows as **K-Means Clustering.**

Analysis:

Before we can conduct K-Means Clustering we have to perform some operations on our datasets.

- The first requirement is to convert our categorical variables such as Venue Category into a machine readable format. This is done by using **One-Hot Encoding.**



	Neighborhood	American Restaurant	Asian Restaurant	Athletics & Sports	Australian Restaurant	Bakery	Bar	Beer Store	Betting Shop	Bookstore	Bowling Alley	Brazilian Restaurant	Breakfast Spot	Brewery	Building	Burger Joint	Bus Station	Bus Stop	Café	Caribbean Restaurant	Chinese Restaurant
0	Frognal Avenue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	Frognal Avenue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Frognal Avenue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	Frognal Avenue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Frognal Avenue	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

One hot encoding transforms the DataFrame into one which provides unique identification for each categorical variable

- The second requirement is to Standardize all numerical data. This is done so that large numerical values (Such as that from our Raw_Price variable) do not cause the algorithm to be skewed in their favor. This scaling is done using **StandardScaler** (sklearn.preprocessing.StandardScaler — scikit-learn 0.24.2 documentation).

Result of scaling our data is shown below:

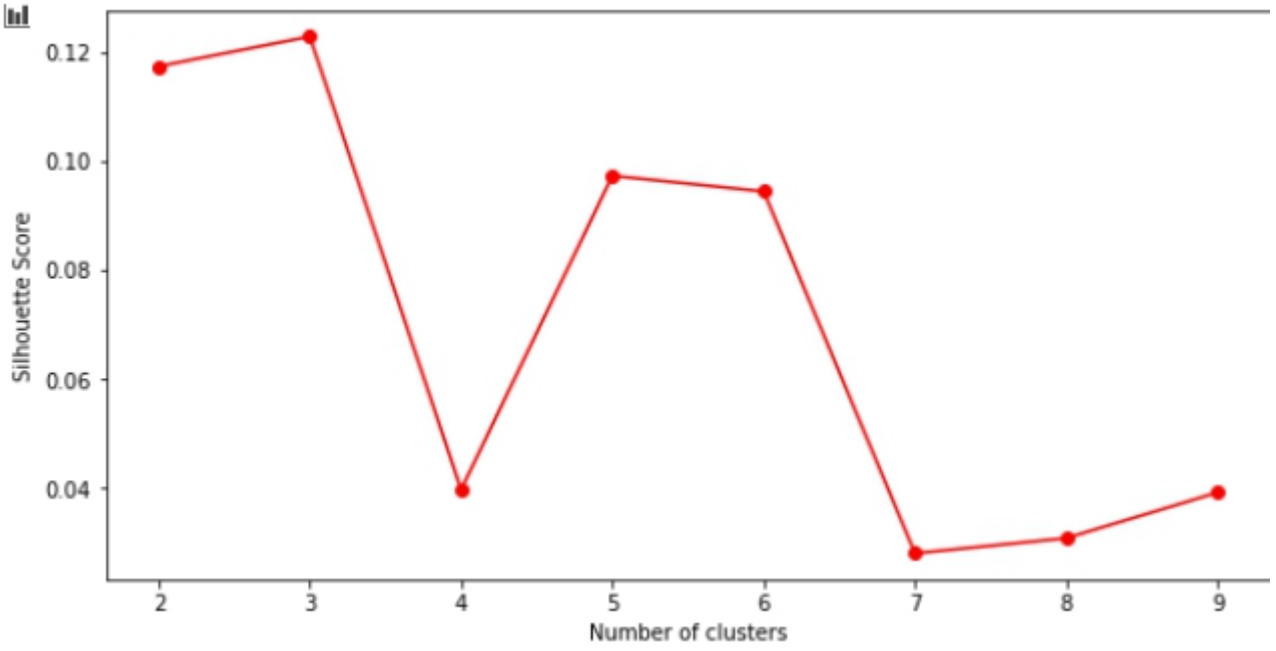
```
array([[-0.3159668 , -0.24225079, -0.30465356, ..., -0.21821789,
        -0.21821789, -0.31865458],
       [-0.3159668 ,  4.55431488, -0.30465356, ..., -0.21821789,
        -0.21821789, -0.56303295],
       [-0.3159668 , -0.24225079, -0.30465356, ..., -0.21821789,
        -0.21821789,  0.41448052],
       ...,
       [-0.3159668 , -0.24225079, -0.30465356, ..., -0.21821789,
        -0.21821789, -0.48157349],
       [-0.3159668 , -0.24225079, -0.30465356, ..., -0.21821789,
        -0.21821789, -0.72595186],
       [-0.3159668 , -0.24225079, -0.30465356, ..., -0.21821789,
        -0.21821789, -0.44084377]])
```

Now with both conditions satisfied we can apply K-Means algorithm to fit our data into clusters.

Analysis:

K-Means is an unsupervised machine learning algorithm which can divide our data set into clusters based on data fed into it.However, before processing the data we have to figure out the ideal number of clusters for the given data set.

This can be done either through trial and error or by using the **Silhouette Method**, which calculates a silhouette score for each cluster.



Here although k=3 provides the highest score, we chose k=5 as three clusters would not provide us with enough data to make a decision.

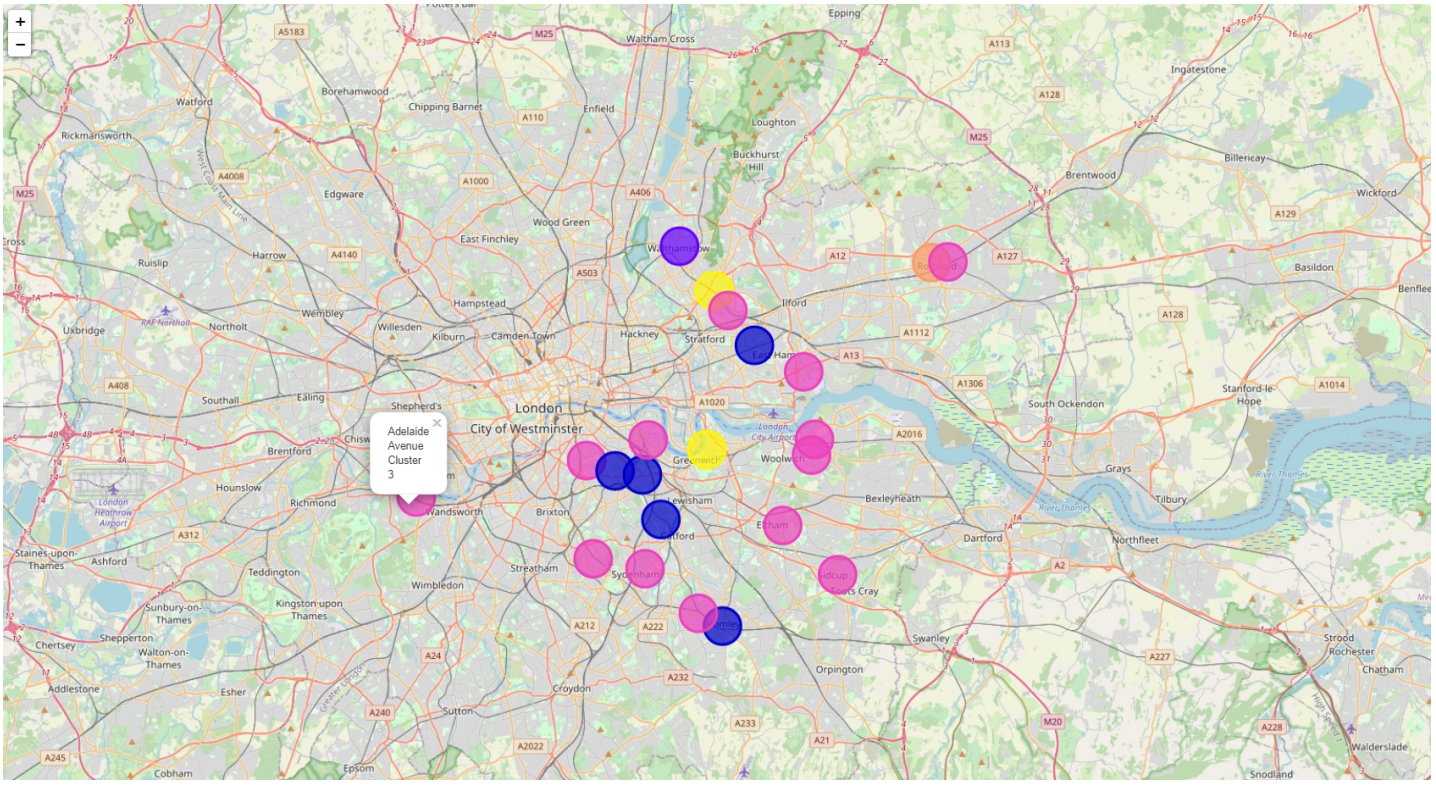
Now keeping k ie.Number of Clusters as 6, we perform K-Means on our Dataset.

The Cluster Labels generated are added to our DataFrame

	Postal Code	City	Neighborhood	Price	Raw_Price	Latitude	Longitude	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 Comm
0	DA14	Sidcup	Frognal Avenue	£3,593 pcm	3593	51.426481	0.105038	3	Pharmacy	Grocery Store	Coffee Shop	Fast Food Restaurant	Pizza Place	Café	Italian Restaurant	Supermarket	Hotel	
1	SE10	London	Cutter Lane Chandlers Avenue	£2,990 pcm	2990	51.487122	0.003284	3	Grocery Store	Pier	Pub	Turkish Restaurant	Coffee Shop	Indian Restaurant	Pizza Place	Café	Japanese Restaurant	Medit Re
2	SE5	London	Lettsom Street	£2,150 pcm	2150	51.481796	-0.090377	3	Park	Middle Eastern Restaurant	Coffee Shop	Café	Pub	Ethiopian Restaurant	Building	Garden	Italian Restaurant	Fish
3	BR1	Bromley	Plaistow Lane	£2,100 pcm	2100	51.401546	0.015415	3	Clothing Store	Coffee Shop	Gym / Fitness Center	Supermarket	Café	Chocolate Shop	Pub	Portuguese Restaurant	Pharmacy	
4	SE9	London	Grove Place	£2,000 pcm	2000	51.450352	0.062336	3	Mediterranean Restaurant	Department Store	Italian Restaurant	Portuguese Restaurant	Pub	Clothing Store	Movie Theater	Fast Food Restaurant	Pharmacy	Sup
5	SE14	London	Childeric Road	£1,900 pcm	1900	51.474867	-0.046933	0	Bus Stop	Platform	Convenience Store	Pub	Chinese Restaurant	Supermarket	Kebab Restaurant	Furniture / Home Store	Fried Chicken Joint	Ele
6	E11	London	New Wanstead	£1,800 pcm	1800	51.564032	0.008813	3	Grocery Store	Coffee Shop	Café	Pub	Pizza Place	Fast Food Restaurant	Fish & Chips Shop	Dim Sum Restaurant	Metro Station	Re
7	SW15	London	Sherfield Gardens	£1,600 pcm	1600	51.464123	-0.222848	2	Coffee Shop	Pub	Clothing Store	Café	Japanese Restaurant	Bakery	Grocery Store	Indian Restaurant	Sushi Restaurant	Re
8	SE4	London	Adelaide Avenue	£1,450 pcm	1450	51.453268	-0.032498	0	Coffee Shop	Fried Chicken Joint	Chinese Restaurant	Café	Bus Stop	Concert Hall	Malay Restaurant	Pub	Convenience Store	

Analysis:

Using this information we can plot another Folium map, but this time showcasing distribution of our Cluster Labels:



Here, the same neighborhoods are shown but this time under different Clusters

Yellow - Cluster 0
Blue - Cluster 1
Purple - Cluster 2
Pink -.Cluster 3
Orange - Cluster 4

Now with this data, Let us look at the Venues in each cluster.

Analysis:

Using the same concept we used before, We can run the Foursquare API to get our required venues, but this time separated based on Clusters.

Type and number of venues available in Cluster 0

```
[698] cluster0_venues['Venue Category'].value_counts()
```

Grocery Store	9
Pub	5
Coffee Shop	4
Pier	3
Café	3
Wine Shop	2
Turkish Restaurant	2
Thai Restaurant	2
Indian Restaurant	2
Pizza Place	2
Mediterranean Restaurant	1
South Indian Restaurant	1
Japanese Restaurant	1
English Restaurant	1
Brewery	1
Metro Station	1
Irish Pub	1
Pharmacy	1
Warehouse Store	1

Type and number of venues available in Cluster 1

```
[699] cluster1_venues['Venue Category'].value_counts()
```

Bus Stop	11
Coffee Shop	8
Clothing Store	7
Pub	7
Grocery Store	6
Supermarket	4
Park	4
Convenience Store	3
Sandwich Place	3
Gym / Fitness Center	3
Chinese Restaurant	3
Café	3
Bakery	3
Fast Food Restaurant	3
Fried Chicken Joint	3
Pharmacy	2
Bar	2
Platform	2

Type and number of venues available in Cluster 2

```
[700] cluster2_venues['Venue Category'].value_counts()
```

Coffee Shop	5
Pub	3
Clothing Store	2
Pizza Place	2
Café	2
Bakery	2
Sandwich Place	2
Pharmacy	2
Restaurant	2
Convenience Store	1
Gym / Fitness Center	1
Bookstore	1
Gift Shop	1
Museum	1
Indian Restaurant	1
Deli / Bodega	1

Type and number of venues available in Cluster 3

```
[701] cluster3_venues['Venue Category'].value_counts()
```

Pub	15
Coffee Shop	12
Café	9
Bakery	8
Clothing Store	7
Supermarket	7
Grocery Store	7
Italian Restaurant	6
Pharmacy	6
Park	4
Train Station	4
Pizza Place	4
Japanese Restaurant	3
Fast Food Restaurant	3
Hotel	3
Fish & Chips Shop	3
Shopping Mall	2
Vietnamese Restaurant	2
Multiplex	2
Video Game Store	2

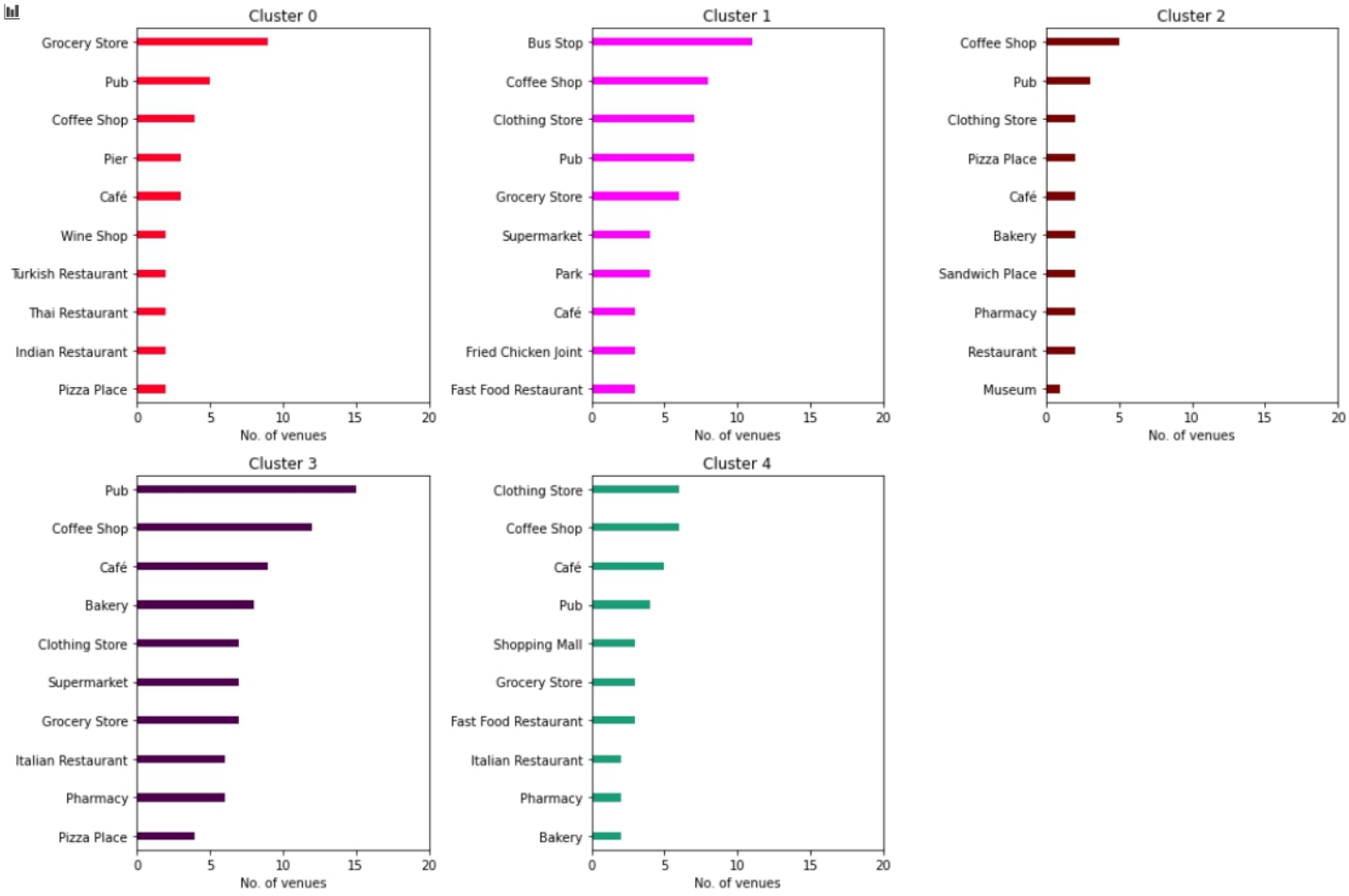
Type and number of venues available in Cluster 4

```
[702] cluster4_venues['Venue Category'].value_counts()
```

Clothing Store	6
Coffee Shop	6
Café	5
Pub	4
Shopping Mall	3
Grocery Store	3
Fast Food Restaurant	3
Supermarket	2
Bookstore	2
American Restaurant	2
Italian Restaurant	2
Bar	2
Bakery	2
Pharmacy	2
Multiplex	1
Park	1
English Restaurant	1
Hotel	1
Chinese Restaurant	1
Department Store	1
Bowling Alley	1

Analysis:

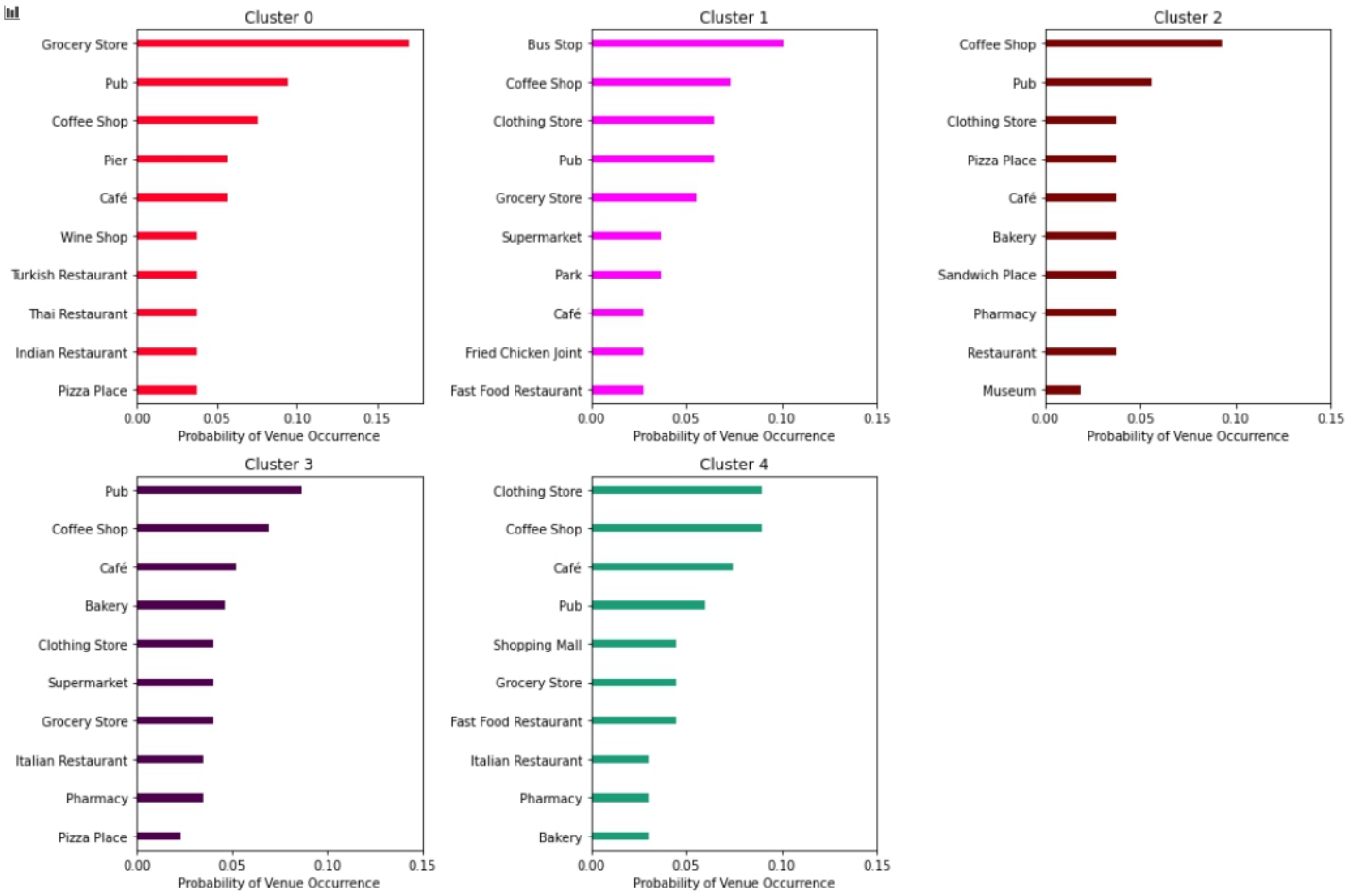
We then plot this data as Bar Charts:



These graphs give us the top 10 Venue Type in each cluster and their total count within the clusters. This can give would-be clients a clear picture of what to expect when moving into these clusters.

Analysis:

However we can notice from our Folium Map that some of these clusters are very large and so if a would-be client had a issue of not availability but proximity of these venues, then we can provide them the probability of occurrence of these venues within 1km of any Apartment in these clusters as:

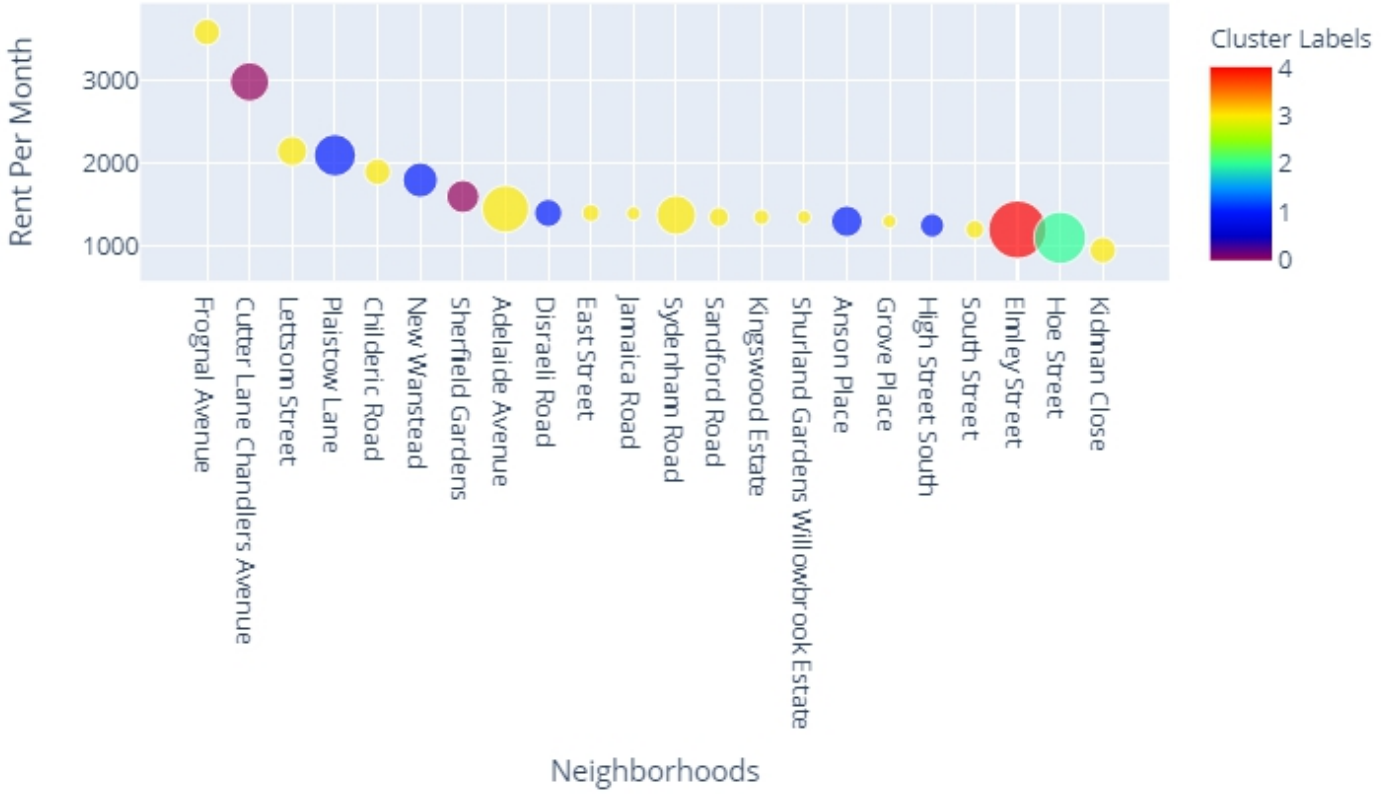


Combining these two graphs, our client will be able to choose a neighborhood that has a high chance of fulfilling all of his personal requirements.

But we still have not answered another important question. The question of whether the client can afford the place.

Analysis:

Since we already have Rent Data for each of the neighborhoods in our Dataset. We can combine all the data which we have utilized thus far and present it in an easy to understand **Bubble Chart:**



In this Bubble Chart, Rent per month for each neighborhood is plotted against each other with size of the Bubble representing the number of venues in that neighborhood.

Here we can observe the relationship between these three factors and combining with data about venue details within each cluster, Our client can finally choose which location to relocate to based on his personal needs and his budget.

Results:

For this section we can go through a number of every day cases a local property agent will have to solve and attempt to solve it using our code.

Case 1:
An international exchange student is moving to London. The student is trying to lessen her expenses as much as possible during her stay. This also means that close proximity to public travel is mandatory as the student will not have a vehicle. Which location should she be shown ?

Sol: Based on our data, Properties in Cluster 1 have high chances of being close to a Bus Stop. Disraeli Road, Anson Place and High Street South all fall in Cluster 1 and also has the lowest rent rates in the cluster.Therefore, the student can be shown properties in these three locations.

Case 2:
A Japanese Businessman is shifting his business to UK and wishes to stay in London. He has amassed a fortune and wants to be located in a very affluent location. Which place should he be shown ?

Sol: Here we can see that, based on our data and contrary to popular belief, Cluster 4 which has one of the lowest rent rates should be recommended to the person. This is because Cluster 4 has several high quality venues such as Shopping Malls, Italian Restaurants, Pubs, Clothing Stores etc.

Case 3:
A London resident wishes to relocate to a better neighborhood. She currently stays near Sydenham Road and wants to shift to a place that's more suited for raising her kids who have just started going to school. Which place should she see ?

Sol: We can see that our client currently resides in Cluster 3. From looking at our venue data for Clusters, we can see that if she shifted to an area in Cluster 1, her children will have close proximity to a park. However, the only place in cluster 1 that has access to larger number of venues than Sydenham Road is Plaistow Lane. We can show her the place and let her decide if the location is worth the increased rent she has to cover.

Discussion & Conclusion

From the example cases in the Results section, we can see that our model has been able to accurately mimic and in some cases even out-do the services of a local property agency. The same model can be reconfigured for locations around the world.

Businesses that offer property listings, for rent or purchase, can also consider adding such a modelling approach to their repertoire. Since most modern property listing websites give limited information regarding the neighborhood of their listing, any company that can incorporate such detailed neighborhood analysis can not only automate the process for all listings in their website but also gain a significant advantage over their competitors.