Finding a Suitable Neighborhood using Clustering & Foursquare API

1. Introduction

1.1 Problem

The project is based on a hypothetical personal scenario, but can be understood and tuned by others to help them out if they are in a similar situation.

I used to live in the city of Manchester. I used to love the place so much, but now due to family needs, we are moving to London.

I wonder how my lifestyle will change? Is there a way I can make myself feel at home even after changing the place? Maybe I can find a similar flat and residence. But what if we take a step further, and try to find a neighborhood that has all the right favorite places at similar proximity as the past residence?

The favorite places at my previous residence were:

- University (3.2 Miles)
- Indian Restaurant (2 Miles)
- Chinese Restaurant (1 Mile)
- Library (5.8 Miles)
- Martial Arts Dojo (6 Miles)
- Hospital (5 Miles)
- Garden (3 Miles)

If I could get all of these at the right proximity from the new home, maybe that would make it very easy to adapt to the new lifestyle. Perfect!

So the problem is :

"Can we find neighborhoods that are most similar to another one, in terms of proximity and availability of favorite venues?"

1.2 Audience

The audience of this experiment is any individual who is looking for moving to a new place, and needs all his favorite places available in a similar fashion as the old residence.

We have chosen the particular case of various Boroughs in London, and the above mentioned venues at particular distances. But the reader can use similar methods to implement a solution for their problem.

2. Data

2.1 Boroughs in London & Their Coordinates

We definitely need to enlist all the Boroughs out there in the City of London, in order to proceed with any step in the experiment.

I found the right thing available at the Wikipedia page https://en.wikipedia.org/wiki/London_boroughs

On which we will use some Web Scraping methods to extract the data.

We will also need the latitude and longitudes of the neighborhoods in order to use the Foursquare API. For the coordinates we will use a free geocoding API named LocationIQ (https://locationiq.com/).

	borough	latitude	longitude
0	London Borough of Camden	51.542855	-0.162526
1	Royal Borough of Greenwich	51.468629	0.048838
2	London Borough of Hackney	51.548882	-0.047669
3	London Borough of Hammersmith and Fulham	51.498314	-0.227878
4	London Borough of Islington	51.547035	-0.101658

2.2 Category-wise Popular Venues in each Borough

We will need to find for each Neighborhood the most popular venues in each category (University, Indian Restaurant, Garden, etc.)

We will average the distance of top venues for a category instead to choosing the minimum. This is because we cannot rely on just one venue to choose the new

neighborhoods, in case it is not good enough.

The Foursquare API provides the **search** endpoint to find popular venues near a location given by particular latitude and longitude. It also allows to filter the results by category.

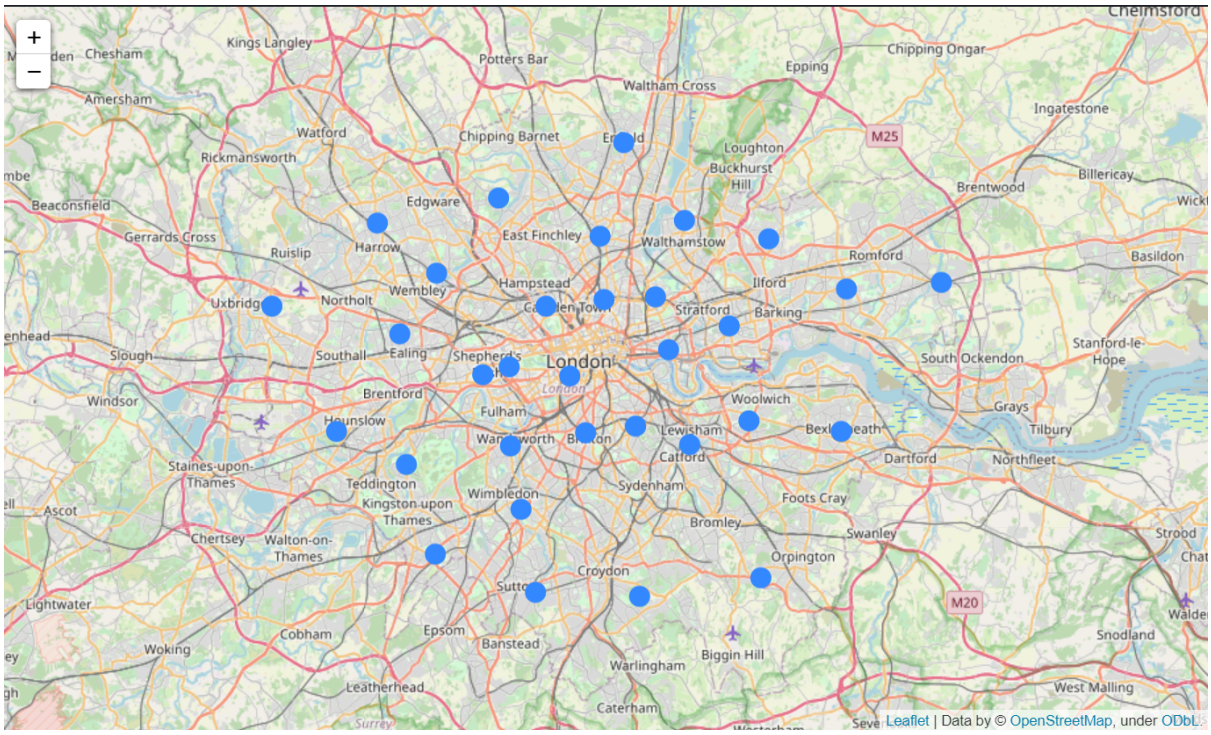
https://api.foursquare.com/v2/venues/search

	borough	University	Indian Restaurant	Chinese Restaurant	Library	Martial Arts Dojo	Hospital	Garden
0	London Borough of Camden	6716.8	3995.2	4209.8	9926.4	11932.4	11979.2	5227.6
1	Royal Borough of Greenwich	11186.6	13420.6	14567.8	17561.4	15217.6	13605.0	16145.2
2	London Borough of Hackney	6158.6	6342.8	8690.6	10365.6	12699.4	14846.8	8908.4
3	London Borough of Hammersmith and Fulham	9884.6	7581.0	5310.6	12851.8	13591.6	12141.8	7796.6
4	London Borough of Islington	5384.2	3829.2	5773.6	9345.8	11479.8	12614.4	6299.4

3. Methodology

3.1 Visualizing the neighborhoods on map

We first need to make sure if all the location coordinates obtained are correct. This can be done by plotting the location markers on a map using folium and observing if the location is wrong or any extreme outliers are seen.



All the points observed were at right place and no outliers were found.

3.2 Adding the venue distances of previous residence

The distances of favorite venues from my previous residence at Manchester were as follows:

```
previous_place = {
'borough' : 'Previous Place',
'University'           : 5199.0,
'Indian Restaurant'    : 3087.6,
'Chinese Restaurant'   : 1585.6,
'Library'             : 9345.8,
'Martial Arts Dojo'    : 9806.2,
'Hospital'            : 8068.0,
'Garden'               : 4879.6,
}
```

We will add an extra datapoint for this location in the Dataset, this will help us to cluster similar places.

3.3 Clustering

After clearing up the dataset, these are our final features:

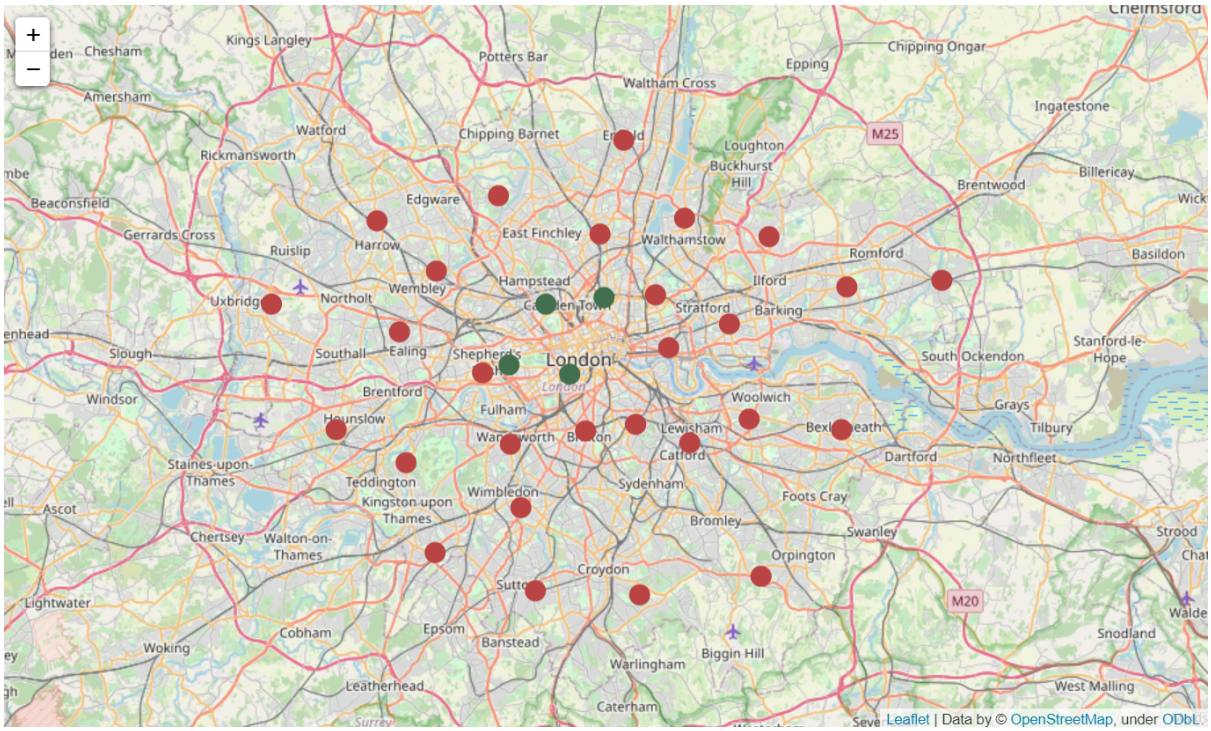
	University	Indian Restaurant	Chinese Restaurant	Library	Martial Arts Dojo	Hospital	Garden
0	6716.8	3995.2	4209.8	9926.4	11932.4	11979.2	5227.6
1	11186.6	13420.6	14567.8	17561.4	15217.6	13605.0	16145.2
2	6158.6	6342.8	8690.6	10365.6	12699.4	14846.8	8908.4
3	9884.6	7581.0	5310.6	12851.8	13591.6	12141.8	7796.6
4	5384.2	3829.2	5773.6	9345.8	11479.8	12614.4	6299.4

We will not apply any type of normalization on the dataset. This is due to the reason that we want to preserve the importance of some venues over other.

We will tune the number of clusters (k) parameter for the KMeans algorithm such that we end up having only 5 datapoints in the same group as of the ideal point.

```
array([0, 6, 5, 5, 0, 0, 5, 6, 5, 5, 5, 0, 4, 3, 4, 6, 4, 4, 6, 3, 5, 2,
       1, 4, 4, 4, 6, 3, 2, 2, 4, 3, 0])
```

4-5. Results & Discussion



After successfully clustering the neighborhoods, we find that the four neighborhoods most similar to the previous residence are Boroughs of Camden, Islington, Kensington, and city of Westminster.

It is quite intuitive that all those neighborhoods are located at the center of London. Most venues might be located near the center naturally.

	borough	labels	latitude	longitude
0	London Borough of Camden	0	51.542855	-0.162526
4	London Borough of Islington	0	51.547035	-0.101658
5	Royal Borough of Kensington and Chelsea	0	51.503795	-0.200789
11	City of Westminster	0	51.497321	-0.137149

6. Conclusion

Finally, we have executed a data science project using Foursquare API, python libraries such as scikit-learn, numpy, pandas. We also used concepts of REST APIs. We put machine learning algorithms such as K-means Clustering to use. Used various libraries such as matplotlib and folium to visualize data. Succesfully solved the problem at hand using all the wonderful tools, libraries and techniques available to us.