THE BATTLE OF NEIGHBOURHOODS

Luciano Guerra Domínguez

July - 2019

Contents

Introduction

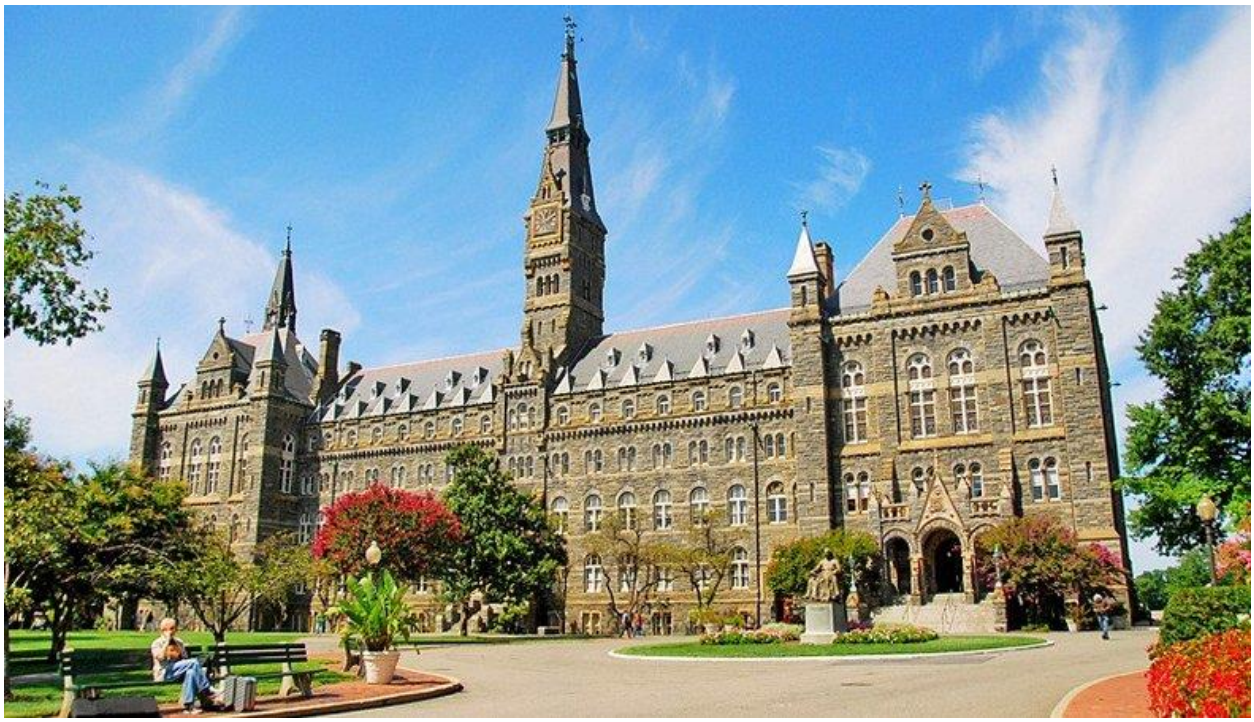
The realization of this project is carried out to obtain a tool that helps people who move to a new city. Depending on the profile of the neighborhood in which you would like to live, according to the known neighborhoods of your current city, similar neighborhoods are identified in the destination city. Similarly, identifying a work point or geographical location that serves as a starting point, will locate the neighborhood closest to this point for greater comfort.

The company I am working for has decided to bet on me like executive manager. Because of this, they are going to pay me an Executive Master in Business Administration, at the Georgetown University (Washington DC).

For this reason, my family and me will have to move to Washington. So, we have to decide which neighborhood is the best according with our needs and likes.

Identifying and understanding the similarities and differences between two chosen neighborhoods to retrieve more insights and to conclude with ease which neighborhood wins over other.

Understanding the similarities and differences between the neighborhoods using Unsupervised K-Mean Clustering Algorithm.



Actually, we have a very successful store in London. As a new business option, we want to open a new store in Washington. We (my wife and me) believe the typology of our actual neighborhood in London takes part of the success of our store, so we want to find all neighborhoods similar to it in Washington. Once we identify this type of neighborhood in Washington, we shall check for the closest to neighborhood to the University buildings. This could be our new neighborhood.

In our actual neighborhood we have a lot of venues, being most relevant:

1. Pub
2. Restaurant
3. Grocery Store
4. Coffee Shop
5. Park

6. Café
7. Gym / Fitness Center
8. Gym
9. Sandwich Place

Data Overview.

We decide to use similar data from Washington DC and London, as well the longitude and latitude of each neighborhood to check distances between them.

The data acquired from different data sources can give us information in *.csv* format.

Washington Data

From Open Data DC, the District of Columbia government shares hundreds of datasets. DC realizes that data's greatest value comes from having it freely shared among agencies, federal and regional governments and with the public to the extent possible when considering safety, privacy and security. The District invites you to browse the data, download it as a file, analyze it with your tools, or build apps using our APIs.

We shall download the Neighborhoods data. This dataset was created by the DC Office of Planning and provides a simplified representation of the neighborhoods of the District of Columbia. These boundaries are used by the Office of Planning to determine appropriate locations for placement of neighborhood names on maps. They do not reflect detailed boundary information, do not necessarily include all commonly-used neighborhood designations, do not match planimetric centerlines, and do not necessarily match Neighborhood Cluster boundaries. There is no formal set of standards that describes which neighborhoods are represented or where boundaries are placed. These informal boundaries are not appropriate for display, calculation, or reporting. Their only appropriate use is to guide the placement of text labels for DC's neighborhoods. This is an informal product used for internal mapping purposes only. It should be considered draft, will be subject to change on an irregular basis, and is not intended for publication

	X	Y	OBJECTID	GIS_ID	NAME	WEB_URL	LABEL_NAME	DATELASTMODIFIED
0	-76.980348	38.855658	1	nhood_050	Fort Stanton	http://NeighborhoodAction.dc.gov	Fort Stanton	2003-04-10T00:00:00.000Z
1	-76.997950	38.841077	2	nhood_031	Congress Heights	http://NeighborhoodAction.dc.gov	Congress Heights	2003-04-10T00:00:00.000Z
2	-76.995636	38.830237	3	nhood_123	Washington Highlands	http://NeighborhoodAction.dc.gov	Washington Highlands	2003-04-10T00:00:00.000Z
3	-77.009271	38.826952	4	nhood_008	Bellevue	http://NeighborhoodAction.dc.gov	Bellevue	2003-04-10T00:00:00.000Z
4	-76.967660	38.853688	5	nhood_073	Knox Hill/Buena Vista	http://NeighborhoodAction.dc.gov	Knox Hill/Buena Vista	2003-04-10T00:00:00.000Z

- X: Longitude
- Y: Latitude
- OBJECTID: Object identifier
- GIS_ID: Geographic Information System Identifier
- NAME: Neighborhood Name
- WEB_URL: http://NeighborhoodAction.dc.gov
- LABEL_NAME: Label of the Neighborhood Name
- DATELASTMODIFIED: Last date of modification.

London Data

We get the information from the London Datastore. The London Datastore has been created by the Greater London Authority (GLA) as a first step towards freeing London's data. We want everyone to be able access the data that the GLA and other public sector organizations hold, and to use that data however they see fit – for free. The GLA is committed to using its connections and influence to request other public sector organizations into releasing their data here too, and it's an objective backed strongly by Sadiq Khan, Mayor of London.

Releasing data though is just half the battle. Raw data often doesn't tell you anything until it has been presented in a meaningful way and most people don't have the tools to do this. That's why we're keen for you to visualize or build apps from the data available on the site.

In this dataset we have all this information: These profiles include data relating to: Population, Households (census), Demographics, Migrant population, Ethnicity, Language, Employment, NEET, DWP Benefits (client group), Housing Benefit, Qualifications, Earnings, Volunteering, Jobs density, Business Survival, Crime, Fires, House prices, New homes, Tenure, Greenspace, Recycling, Carbon Emissions, Cars, Public Transport Accessibility (PTAL), Indices of Multiple Deprivation, GCSE results, Children looked after, Children in out-of-work families, Life Expectancy, Teenage conceptions, Happiness levels, Political control, and Election turnout. But finally, just shall use the name of the neighborhood after filtering it.

	Code	Area_name	Inner/_Outer_London	GLA_Population_Estimate_2017	GLA_Household_Estimate_2017	Inland_Area_(Hectares)	Population_density_(per_hectare)_2017	Average_Age,_2017	Proportion_of_population_aged_0-15,_2015	Proportion_of_population_of_working-age,_2015	...
0	E09000001	City of London	Inner London	8800	5326	290	30.3	43.2	11.4	73.1	...
1	E09000002	Barking and Dagenham	Outer London	209000	78188	3,611	57.9	32.9	27.2	63.1	...
2	E09000003	Barnet	Outer London	389600	151423	8,675	44.9	37.3	21.1	64.9	...
3	E09000004	Bexley	Outer London	244300	97736	6,058	40.3	39.0	20.6	62.9	...
4	E09000005	Brent	Outer London	332100	121048	4,323	76.8	35.6	20.9	67.8	...

After filtering…

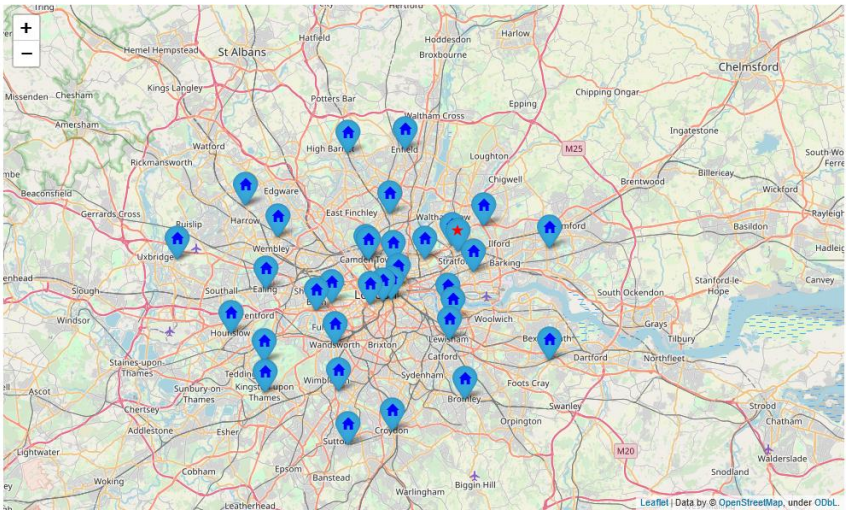
	Code	Neighborhood
0	E09000001	City of London
1	E09000002	Barking and Dagenham
2	E09000003	Barnet
3	E09000004	Bexley
4	E09000005	Brent

Methodology.

Store neighborhood.

We have to know which of the neighborhoods around our store has more influence in the store success. So, based on the location data of the store I am going to check the distance of all neighborhoods in London and our store. We shall decide the most relevant is the closest neighborhood.

We can check in a map all neighborhoods related to our store position.



We use the *folium* library to represent the map and calculate the distance between them using the *geopy* library.

After this, we decide the most relevant neighborhood in London is **Waltham Forest.**

	City	Latitude	Longitude	Neighborhood	Distance
30	London	51.556999	-0.005835	Waltham Forest	0.993647

Target position.

We have decided to build our new store close to Georgetown University and with a similar profile as our London neighborhood. So, we must understand the typology of neighborhoods in Washington.

We prepare a dataset with all this information, the name of the neighborhood and its position. Once the dataset is ready, we could append our London neighborhood and check similarities to other Washington neighborhoods.

After this we shall compare all neighborhoods including our London neighborhood.

```
In [142]: newAreaDF.tail()
```

Out[142]:

	City	Distance	Latitude	Longitude	Neighborhood
128	Washington	NaN	38.863453	-76.951630	Fairfax Village
129	Washington	NaN	38.861794	-76.960688	Hillcrest
130	Washington	NaN	38.943327	-77.041097	Crestwood
131	Washington	NaN	38.904340	-77.023313	Mount Vernon Square
132	London	0.993647	51.556999	-0.005835	Waltham Forest

Neighborhood clustering.

Once we have all data in the same dataset, we can check the typology of each neighborhood according with most representative venues.

We shall use foursquare to check this information and will prepare a new dataset with all venues (with a limit of 100 and less than 1km). So, we get a new dataframe with 6983 rows, including a row for each neighborhood and venue.

If we group by neighborhood, we can see some neighborhoods have the maximum number of venues we have limited.

```
In [88]: newAreavenues.groupby('Neighborhood').count()
```

Out[88]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
16th Street Heights	35	35	35	35	35	35
Adams Morgan	100	100	100	100	100	100
American University Park	64	64	64	64	64	64
Arboretum	41	41	41	41	41	41
Barnaby Woods	7	7	7	7	7	7
Barry Farm	11	11	11	11	11	11
Bellevue	14	14	14	14	14	14
Benning	29	29	29	29	29	29
Benning Ridge	24	24	24	24	24	24
Bloomingdale	85	85	85	85	85	85
Brentwood	65	65	65	65	65	65
Brightwood	30	30	30	30	30	30
Brightwood Park	21	21	21	21	21	21
Brookland	74	74	74	74	74	74
Burleith/Hillandale	100	100	100	100	100	100
Burrville	11	11	11	11	11	11
Buzzard Point	64	64	64	64	64	64
Capitol Hill	100	100	100	100	100	100
Capitol View	16	16	16	16	16	16
Cardozo/Shaw	100	100	100	100	100	100
Carver	73	73	73	73	73	73
Cathedral Heights	70	70	70	70	70	70
Central NE	29	29	29	29	29	29
Chevy Chase	29	29	29	29	29	29

Now we use the one hot encoding representation for each venue.

A one hot encoding is a representation of categorical variables as binary vectors. This first requires that the categorical values be mapped to integer values. Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

A one hot encoding allows the representation of categorical data to be more expressive. Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.

```
In [143]: newAreavenues_grouped = newAreavenues_onehot.groupby('Neighborhood').mean().reset_index()
          newAreavenues_grouped.head()
```

Out[143]:

	Neighborhood	Zoo Exhibit	Accessories Store	Afghan Restaurant	African Restaurant	Airport Lounge	Airport Terminal	American Restaurant	Antique Shop	Arcade	...	Weight Loss Center	Whisky Bar	Wine Bar	Wine Shop	Winery	Wings Joint	Women's Store	Xinjiang Restaurant	Yoga Studio	Zoo
0	16th Street Heights	0.0	0.0	0.00	0.00000	0.0	0.0	0.0	0.00000	0.0	...	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.0
1	Adams Morgan	0.0	0.0	0.01	0.00000	0.0	0.0	0.0	0.00000	0.0	...	0.0	0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0200	0.0
2	American University Park	0.0	0.0	0.00	0.00000	0.0	0.0	0.0	0.00000	0.0	...	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0625	0.0
3	Arboretum	0.0	0.0	0.00	0.02439	0.0	0.0	0.0	0.02439	0.0	...	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.0
4	Barnaby Woods	0.0	0.0	0.00	0.00000	0.0	0.0	0.0	0.00000	0.0	...	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.0

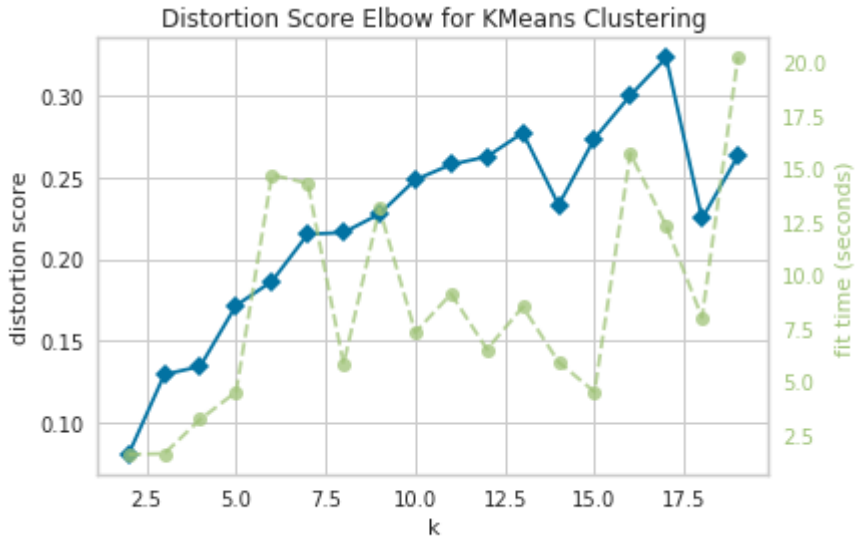
5 rows × 352 columns

```
In [92]: #rechecking the new size of the dataframe...
         newAreavenues_grouped.shape
```

Out[92]: (133, 352)

Once we have most relevant venues in all neighborhoods inside our Dataframe, we must create groups of types of neighborhood. This is the most important part on our agues to justify our store in Washington shall have same success as our store in London already has.

I shall use the **k-means** method for clustering neighborhoods. Before anything we shall check how many clusters give to us the best performance in the clustering. To do this, we will use the **Elbow method** with all previous information.



So, best performance is reached to k = 16. It means we shall have sixteen different clusters and our London neighborhood shall be on cluster 5.

```
In [99]: newAreaDF_merged[newAreaDF_merged.City == 'London']
```

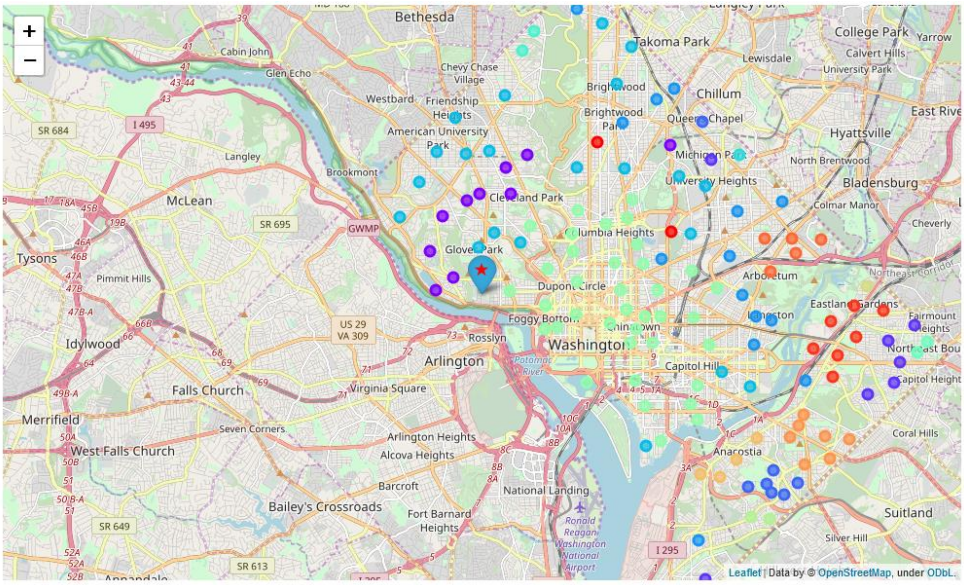
Out[99]:

	City	Distance	Latitude	Longitude	Neighborhood	Cluster	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
132	London	0.993647	51.556999	-0.005835	Waltham Forest	5	Pub	Restaurant	Grocery Store	Coffee Shop	Park	Gym	Gym / Fitness Center

Neighborhood selection.

Once we have all Washington neighborhoods clustered and linked to our London neighborhood, we can show this information over a map.

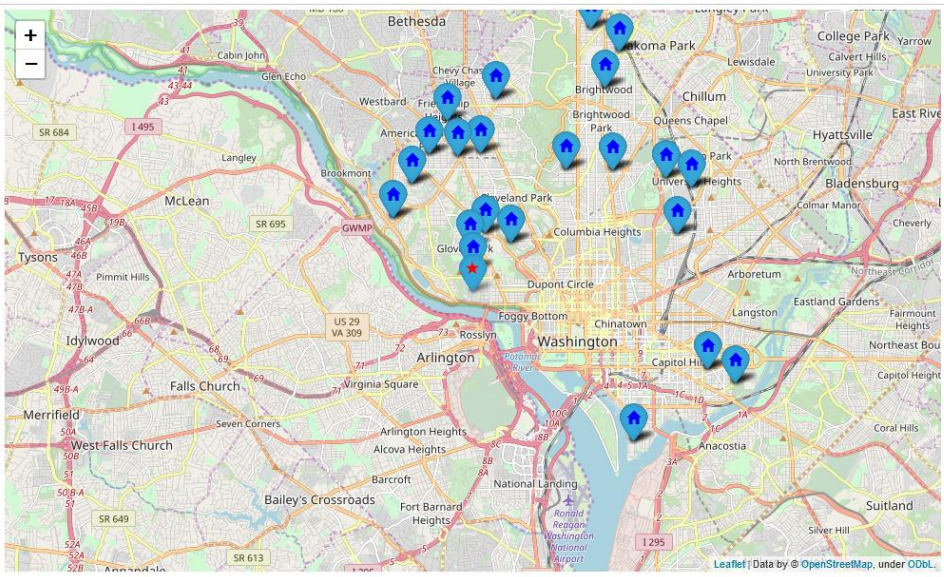
Moreover, we include the location of the Georgetown University to check the relationship with others neighborhoods.



Result.

According with this clustering and the location of the University, we can decide which is the best neighborhood to place our new store with options of success.

To obtain this, we use the *geopy library* again, to calculate distance between the university and the neighborhoods within the same cluster. As que can see in the next map.

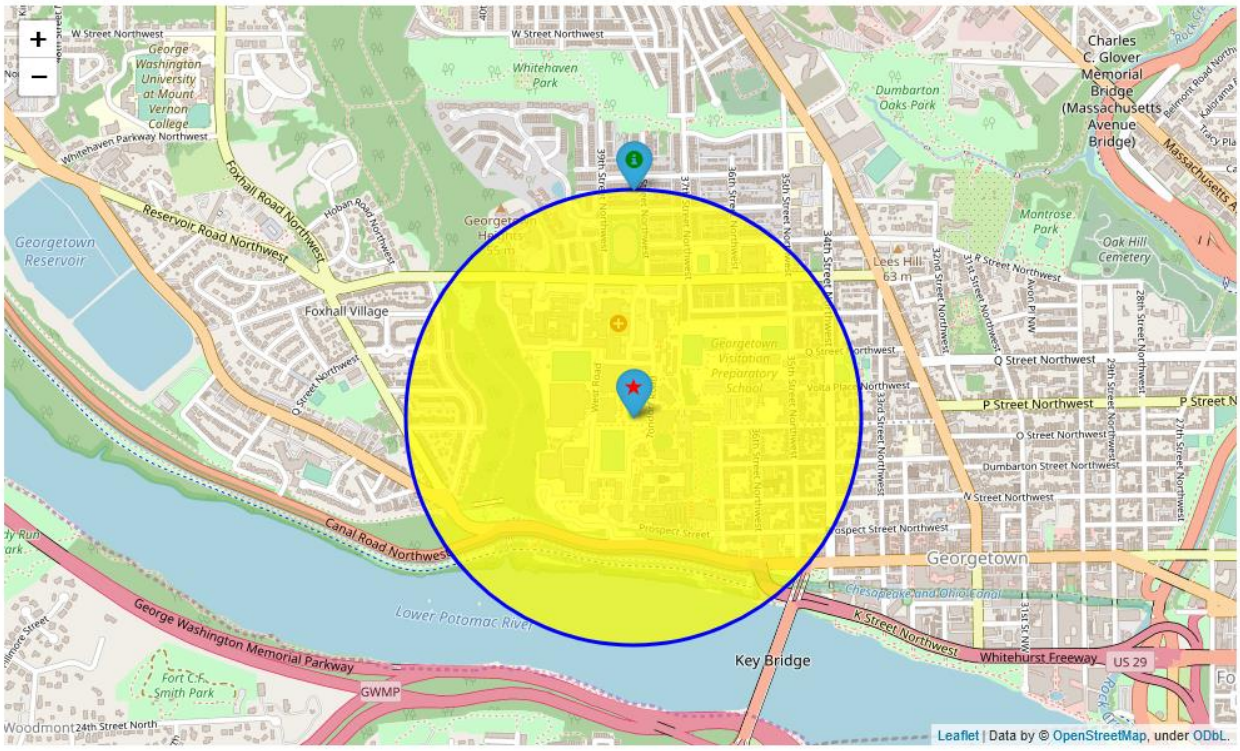


According this, the best neighborhood to place our store in Washington shall be: **Burleith/Hillandale.**

```
In [116]: clusterNeigh[clusterNeigh.Distance == clusterNeigh['Distance'].min()].Neighborhood

Out[116]: 85    Burleith/Hillandale
          Name: Neighborhood, dtype: object
```

Out[141]:



In [118]:
```
target.Distance
```

Out[118]:
```
85    0.67316
Name: Distance, dtype: float64
```

Discussion

After this, we have selected our new neighborhood in Washington with great expectations of success based on the information we get from our neighborhood in London.

Of course, we could include even more information in our dataset like income rate, restaurant types, crime rates… to make this decision even safer. But procedure should be the same as we did.

So, this methodology could help somebody to select where to move to a new town according with a known neighborhood in any city of the world.

Conclusion

This analysis is performed on limited data but procedure is clear. If we increase the amount of data collected from different sources, we could improve the result.

Finally, we have decided where to open our new store in Washington according with some trends in London that we know works fine. The typology of a neighborhood implies a way of life and if we can select clusters of people with similar behaviors of our neighborhood in London we shall hit in the correct place.

People try to group in accordance with their likes, salaries, needs, preferences… at the end, it is the same procedure we did searching for clusters as the data we got in the beginning.