# IBM Coursera Capstone Project

***By Abe Louw***

***This report contains 6 sections:***

# 1. Introduction

Before COVID-19 threw all other business strategies or plans into complete disarray, Europe was still getting to grips on how Brexit was going to affect the region.

Now that Britain has chosen to leave the EU, many businesses will relocate their head offices outside of London, the world's former financial center. With this relocation, hundreds of thousands (if not millions) of people will have to uplift their lives and move to another city.

Based on this forbes article, Amsterdam could be the 'new London'. If employees are forced to move from the one city to the other, the big question is: which neighborhoods should these soon-to-be ex-Londoners choose to move to in Amsterdam?

Based on Foursquare data, I aim to cluster Amsterdam and London neighborhoods based on similarity. So if you love your neighborhood in London and want something similar in Amsterdam, this is your go-to guide.

# 2. Data

I gathered data from three sources: Wikipedia, the Google Maps API and the Foursquare API.

- Wikipedia: I scraped Wikipedia pages for *Borough* and *Neighborhood* data for London and Amsterdam.
- Google Maps: I accessed the google maps API through GeoPy to get *Latitude* and *Longitude* data to correspond with the *Neighborhood* and *Borough* fields, gleaned off Wikipedia.

- Foursquare API: I used the geo-location data from Wikipedia and the Google Maps API to search the Foursquare API using the explore function to get the 10 most common venue categories for each neighborhood in Amsterdam and London respectively.

# 3. Methodology

As this project relies heavily on API's (Google Maps and Foursquare) and publicly available Wikipedia data, it required a lot of data cleaning and data wrangling. Once the data was in the right format, a *k nearest neighbor's* algorithm was used to cluster the neighborhoods

Broadly, the methodology is as below:

Data Gathering, Exploratory Data Analysis and Visualization

- Gather Wikipedia Neighborhood data for London and Amsterdam
- Use the Google Maps API and GeoPy to get corresponding latitude and longitude values per neighborhood
- Visualizing the Neighborhood breakdown by city
- Use the obtained latitude and longitude values to find the 10 most common venue categories in each neighborhood
- Visualizing the most common venue categories in London and Amsterdam

Clustering Approach

- Explaining why a k nearest neighbor's algorithm was used as well as the selection of k.

**Data Gathering**

<u>Wikipedia Scraping</u>

To get the London and Amsterdam data I scraped <u>this</u> and <u>that</u> Wikipedia page using pd.read_html to get the neighborhood information.

*Figure 1 Initial Wikipedia Data frame for London*

| | Location | London borough | Post town | Postcode district | Dial code | OS grid ref |
|---|---|---|---|---|---|---|
| 0 | Abbey Wood | Bexley, Greenwich [7] | LONDON | SE2 | 020 | TQ465785 |
| 1 | Acton | Ealing, Hammersmith and Fulham[8] | LONDON | W3, W4 | 020 | TQ205805 |
| 2 | Addington | Croydon[8] | CROYDON | CR0 | 020 | TQ375645 |
| 3 | Addiscombe | Croydon[8] | CROYDON | CR0 | 020 | TQ345665 |
| 4 | Albany Park | Bexley | BEXLEY, SIDCUP | DA5, DA14 | 020 | TQ478728 |

*Figure 2 Wikipedia Data frame for Amsterdam*



| | Borough | Area | Population | Population density | Location (in green) | Neighbourhoods |
|---|---|---|---|---|---|---|
| 0 | Centrum (Centre) | 8.04 km² | 86422 | 13,748/km² | NaN | Binnenstad, Grachtengordel, Haarlemmerbuurt, J... |
| 1 | Noord (North) | 49.01 km² | 94766 | 2,269/km² | NaN | Banne Buiksloot, Buiksloot, Buikslotermeer, Fl... |
| 2 | Nieuw-West(New West) | 32.38 km² | 151677 | 4,476/km² | NaN | Geuzenveld, Nieuw Sloten, Oostoever, Osdorp, O... |
| 3 | Oost (East) | 30.56 km² | 135767 | 7,635/km² | NaN | IJburg, Indische Buurt, Eastern Docklands, Oud... |
| 4 | West | 9.89 km² | 143842 | 15,252/km² | NaN | Frederik Hendrikbuurt, Houthaven, Spaarndammer... |

Both of the above data frames were cleaned, tidied and reshaped so that they were ready to be used to get latitude and longitude values. For those folks who know python, I broadly did the below:

- Drop unnecessary fields: 'Post Town', 'Dial Code', OS grid ref', 'Area', 'Population', 'Population Density', 'location (in green)
- Use regex to remove unnecessary reference brackets, 'English translations' of Amsterdam Boroughs and other strings
- Use split and stack on the Amsterdam data frame to parse out the neighborhood field using a comma as the separator

Sourcing Coordinates Using Google Maps API

Next, I had to use the Google Maps API to get the latitude and longitude data given each neighborhood name. Note that Google now charges for its API so I had to enter in my credit card details. The resulting data frame, after joining it back with the original data frame is as below:

*Figure 3: Google Maps API Data*



| | neighborhood | borough | latitude | longitude | city |
|---|---|---|---|---|---|
| 0 | Abbey Wood | Bexley | 51.486042 | 0.120259 | london |
| 1 | Acton | Ealing | 51.512075 | -0.267572 | london |
| 2 | Aldgate | City | 51.515987 | -0.078379 | london |
| 3 | Aldwych | Westminster | 51.510583 | -0.126768 | london |
| 4 | Anerley | Bromley | 51.412571 | -0.061397 | london |

We now have a final geo-location data frame with 368 neighborhoods across both cities with a rough 80/20 split where London has 299 Neighborhoods and Amsterdam only has 69. Given the size of London compared to Amsterdam, it's no surprise that London has over 3 times as many neighborhoods.

In total, there are 41 unique Boroughs between the two cities. The Borough with the most neighborhoods in Amsterdam is Centrum with 14 Neighborhoods. The Borough with the most neighborhoods in London is Barnet with 26 Neighborhoods.

Based on the neighborhood data, I then used folium to plot the neighborhoods onto each city. You can see that there are far more neighborhoods in London than Amsterdam- which makes sense because London covers a much larger area.

*Figure 4: London Neighborhoods*



*Figure 5: Amsterdam Neighborhoods*



Fetching Venue Information from Foursquare

To get our Foursquare Data in a workable format to do our clustering any cleaning and wrangling steps were followed:

- Fetching the JSON file information for every venue within a given radius of our neighborhood coordinates
- Flattening that JSON and parsing out the category for each venue. A category, for example could be a 'Pub', 'Park', or 'Soccer Stadium'.
- Finding the number of venues per category within a neighborhood
- Only selecting the top 10 categories by count per neighborhood

The resulting data frame was as below. You can see that the "Venue Category" has been added as the last field.

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Abbey Wood | 51.486042 | 0.120259 | Greggs | 51.490164 | 0.121305 | Bakery |
| 1 | Abbey Wood | 51.486042 | 0.120259 | Abbey Wood Caravan Club | 51.485502 | 0.120014 | Campground |
| 2 | Abbey Wood | 51.486042 | 0.120259 | Abbey Cafe | 51.489754 | 0.120822 | Café |
| 3 | Acton | 51.512075 | -0.267572 | Bake Me | 51.508452 | -0.268543 | Creperie |
| 4 | Acton | 51.512075 | -0.267572 | The Station House | 51.508877 | -0.263076 | Pub |

Now that we have the category for each venue, we need to count the number of venues per category and then only take the top 10 biggest categories per neighborhood. To do this we have to:

- Use 'onehot' encoding on our category field.
- Group by Neighborhood, taking the mean of the frequency of occurrence
- Write a function to sort the venues in descending order
- Create a new data frame to display the top venues for each neighborhood

The resulting cleaned data frame is as below:

*Figure 7: Top 10 Venue Categories per Neighborhood*

| | 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 | Bos en Lommer | Tram Station | Bar | Vegetarian / Vegan Restaurant | Bakery | Indonesian Restaurant | Coffee Shop | Gastropub | Snack Place | Thai Restaurant | Grocery Store |
| 1 | Buiksloot | Supermarket | Bus Stop | Park | Shopping Mall | Turkish Restaurant | Restaurant | Bakery | Fish & Chips Shop | Fish Market | English Restaurant |
| 2 | Buitenveldert | Supermarket | Clothing Store | Drugstore | Convenience Store | Electronics Store | Grocery Store | Bakery | Sandwich Place | Plaza | Restaurant |
| 3 | Buitenveldert | Drugstore | Hotel | Gym / Fitness Center | Snack Place | French Restaurant | Bookstore | Furniture / Home Store | Supermarket | Sushi Restaurant | Liquor Store |
| 4 | De Baarsjes | Bar | Restaurant | Falafel Restaurant | Greek Restaurant | Supermarket | Italian Restaurant | Pub | Deli / Bodega | Farm | Bistro |

Having a look at the most common venues across all neighborhoods we find that people from Amsterdam and London love Pubs, Café's and Coffee shops. I find it very interesting that there are more than twice as many pubs as there are Coffee Shops

*Figure 8: Most Common Venue Categories: Amsterdam and London*
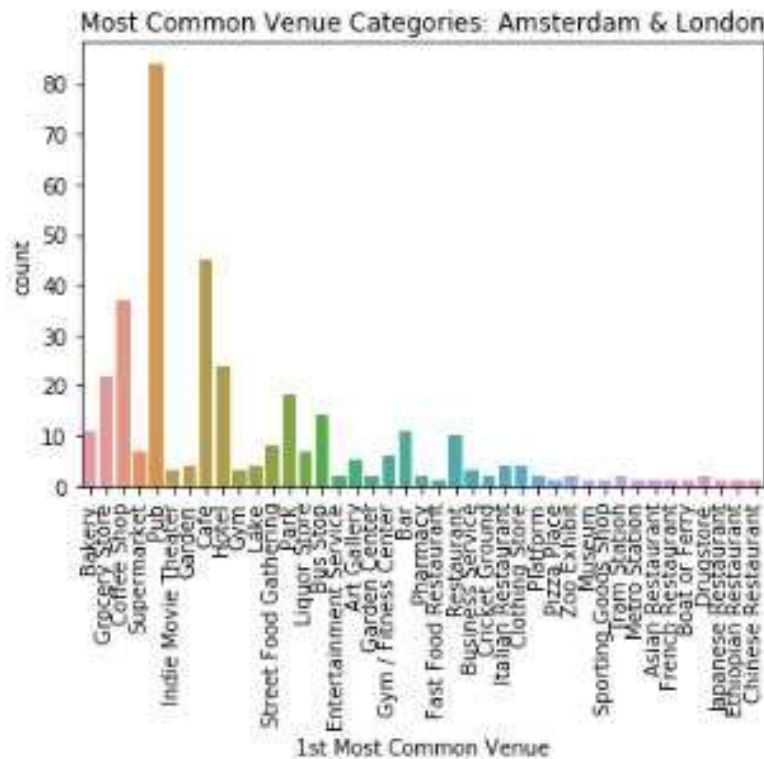
**Clustering Approach**

I chose to use the *'K Nearest Neighbors'* algorithm for clustering due to its ease of use, very simple implementation and given that the classes don't have to be linearly separable.

Based on previous experience, setting the number of clusters to be too high in this kind of analysis can end up making the final clusters hard to make sense of and the clustering hard to interpret. In the same breath if we use too few clusters, then we have homogeneity. Given the above I decided to set the number of clusters used for the analysis to be 5 (not too big and not too small).

# 4. Results

Once the clustering was complete, we have a final data frame ready for analysis. See below:

| | Cluster Labels | 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 | 0 | Bos en Lommer | Tram Station | Bar | Vegetarian / Vegan Restaurant | Bakery | Indonesian Restaurant | Coffee Shop | Gastropub | Snack Place | Thai Restaurant | Grocery Store |
| 1 | 0 | Buiksloot | Supermarket | Bus Stop | Park | Shopping Mall | Turkish Restaurant | Restaurant | Bakery | Fish & Chips Shop | Fish Market | English Restaurant |
| 2 | 0 | Buikslotermeer | Supermarket | Clothing Store | Drugstore | Convenience Store | Electronics Store | Grocery Store | Bakery | Sandwich Place | Plaza | Restaurant |
| 3 | 0 | Buitenveldert | Drugstore | Hotel | Gym / Fitness Center | Snack Place | French Restaurant | Bookstore | Furniture / Home Store | Supermarket | Sushi Restaurant | Liquor Store |
| 4 | 0 | De Baarsjes | Bar | Restaurant | Falafel Restaurant | Greek Restaurant | Supermarket | Italian Restaurant | Pub | Deli / Bodega | Farm | Bistro |

Analyzing the clusters using Figure 10 below, there are a few interesting observations:

- The largest cluster is cluster 0 with over 200 neighborhoods. Cluster 0 has 162 London neighborhoods and 67 Amsterdam Neighborhoods
- The second largest cluster is cluster 2 with 109 Neighborhoods- all from London.
- The third largest cluster is cluster 1 with 11 neighborhoods. 9 London Neighborhoods and 2 Amsterdam Neighborhoods
- Cluster 3 and 4 respectively are the smallest with 7 and 5 neighborhoods respectively- all in London

*Figure 10: Number of Clusters per City*
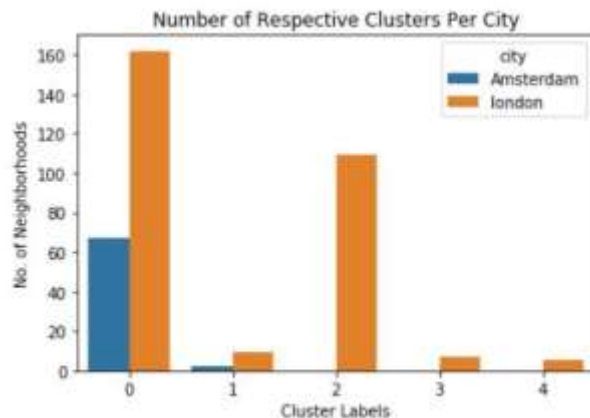


Unfortunately Cluster 2, 3 and 4 don't include any Amsterdam neighborhoods. This could mean that there truly is "no place like home" for Londoners who live in those areas.

Taking a deeper dive into the Clusters, I visualized the top most frequent venue categories. This also helped to get a better naming convention.

**Cluster 0: Family Friendly**

As you can see below, there are many venue categories in these neighborhoods. There are a high number of Pubs, Hotels, Cafe's, Grocery Stores and Bus Stops. I'd call this neighborhood "Family Friendly".
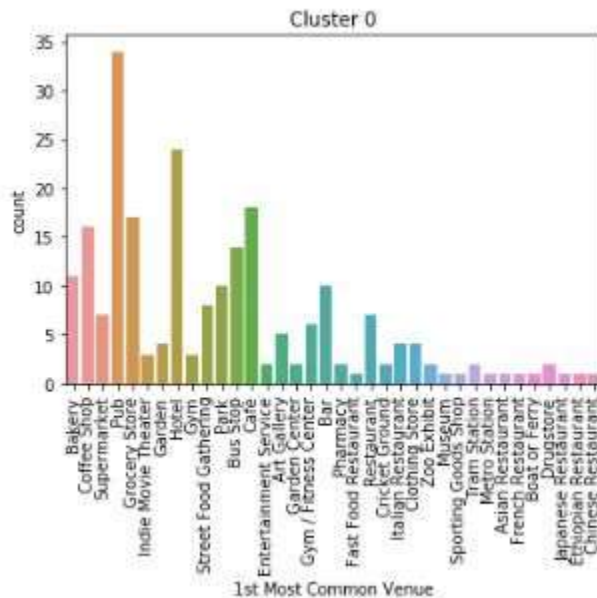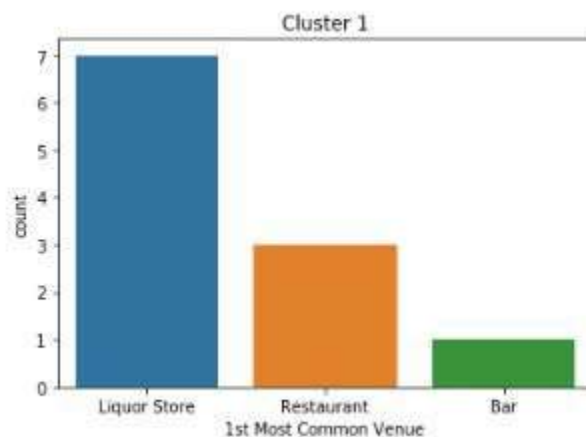
## Cluster 1: Party Place

These neighborhoods seem to be pretty homogenous with only 3 categories in the top pick. If you like to party, this could be the place for you as there are many liquor stores, bars and restaurants. I'd call these neighborhoods "Party Place".
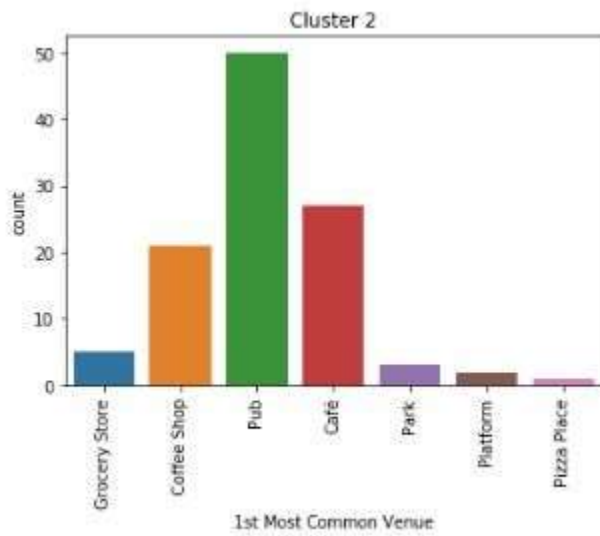
*Figure 12: Most Common Venue Categories in Cluster 1*



## Cluster 2: Coffee Corner

These neighborhoods seem to have more cafe's and coffee shops than other clusters. If you're tired of coffee you could also head over to a pub for a pint. I'd call these neighborhoods "Coffee Corner".
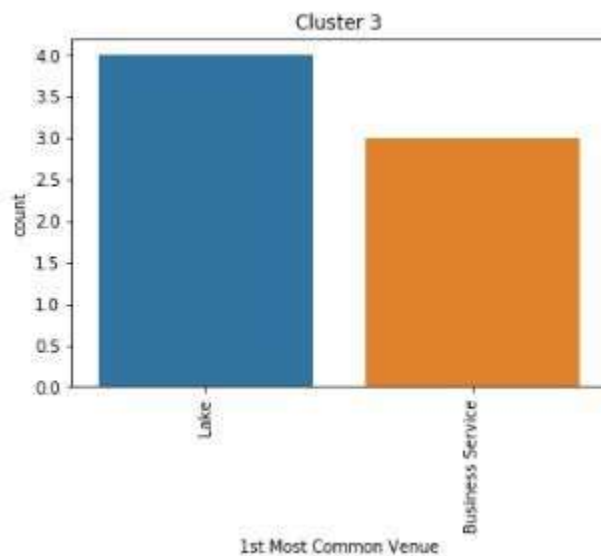
## Cluster 3: Lake Views

These neighborhoods have two things that they have in leaps and bounds: lakes and businesses. These could be a "Central Business District" areas with lake-side views. I'd call these neighborhoods "Lake Views".
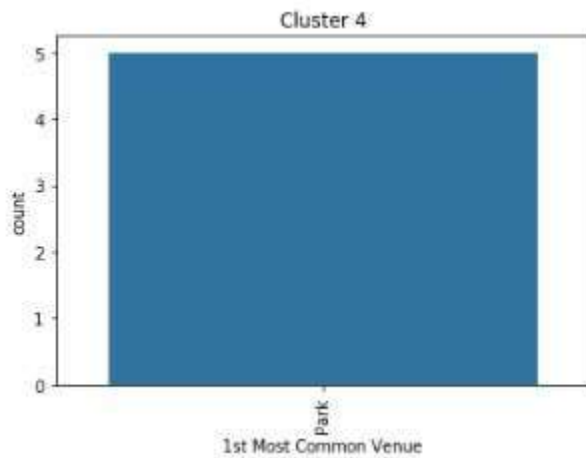
*Figure 14: Most Common Venue Categories in Cluster 3*



## Cluster 4: Park Pozzy

These neighborhoods have one thing in common: parks! I'd call these neighbors "Park Pozzy"

In summary, we have 5 heterogonous clusters:

- Family Friendly (Cluster 0)
- Party Place (Cluster 1)
- Coffee Corner (Cluster 2)
- Lake Views (Cluster 3)
- Park Pozzy" (Custer 4)

Using the below *Figure 11* and *Figure 12* folium maps to visualize the clusters, Londoners can use the below map to find out which cluster they currently live in and which cluster they could move into in Amsterdam.

*Figure 16: London Neighborhoods, Colour-coded by Cluster*

*Figure 17: Amsterdam Neighborhoods, Colour-coded by Cluster*
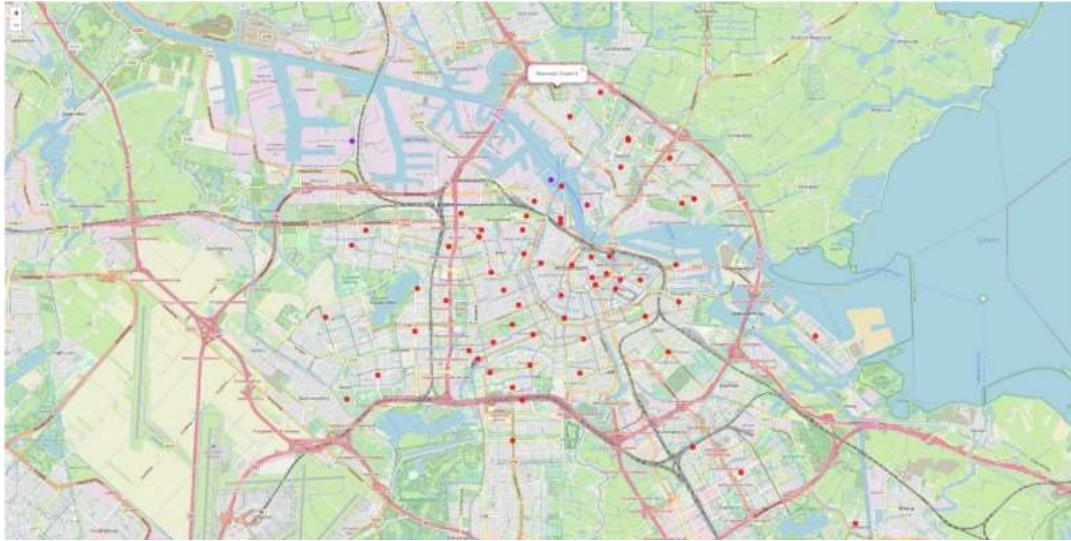


# 5. Discussion

The above tool is a great start although it isn't perfect. Further iterations of this project could be done using different clustering algorithms and numbers of clusters. This could help to solve the issue of not having any Amsterdam neighborhoods in Clusters 2, 3 and 4.

Furthermore, a web-based application would be useful to improve the user experience of this tool. An application where a user can enter in their address in London and then the corresponding Amsterdam neighborhood recommendation pops up, would be preferable to visualize and serve the results.

# 6. Conclusion

In conclusion, this report presents the steps taken to cluster London and Amsterdam neighborhoods based on similarity of venue category frequency. By using a KNN algorithm 5 neighborhood clusters were presented that were heterogeneous: Family Friendly, Party Place, Coffee Corner, Lake Views and Park Pozzy. As the last 3 clusters only include London neighborhoods, it could be inferred that there truly is "no place like home" for those neighborhoods outside of London.