Introduction

People move frequently from one city to another, perhaps one they have never been to before, be it for work or any other reason, and a lot of us would a place that most resembles the one they moved from, as it would be easier to adapt and live in. For this purpose, this project proposes an algorithm that measures the similarity between boroughs of different cities, based on their venues, and lists the most similar five boroughs in another city. Such an algorithm could be very interesting for those who move a lot because of their work, as well as adventurous individuals who love dynamic life style and move frequently between cities.

Data

Description of the datasets

This project is an illustration of the idea and how it works, and therefore it won't contain all the major cities in the world, however, two cities, London and Paris, are selected to demonstrate how the model works. To that end, the boroughs of city of London will be compared with those of Paris considering the type of venues each lie in each borough. The data required for this tasks is a list of London boroughs, a list of Paris boroughs, and lists of the venues in every borough. The data of the borough of both cities is extracted from Wikipedia pages, while the venues data is obtained using the Foursquare API.

Loading and preprocessing

The data of the boroughs is extracted using the Pandas workframe, once the tables are extracted, they are stored in a dataframe for preprocessing to take place. The preprocessing includes correcting the columns headers, dropping the unnecessary columns, and cleaning the cells that contain alphanumeric values that are irrelevant.

To load the venues data of each borough, the geographical coordinates of the borough are necessary. For that purpose, the geopy library which could obtain the coordinates data based on the borough name is used. A function is defined (coordinates_finder) which takes the address as an input and spits out the latitude and the longitude of the address. This function is then used to find the coordinates of each borough, and the results are stored in a list, which is then joined with the dataframe that contains the boroughs names.

A second function is defined (getNearbyVenues) to extract a specific number of venues that are closest to the specified coordinates, as the retrieval of all the venues is not very practical. This function’s parameters are the boroughs name, its latitude, longitude, and the radius which the extracted venues should be in. The results are then stored in the same dataframe that contains the boroughs data.

Data visualization

To visualize the boroughs locations on the map and how disperse they are, folium library of python is used, which enables drawing on maps. The results are as follows:

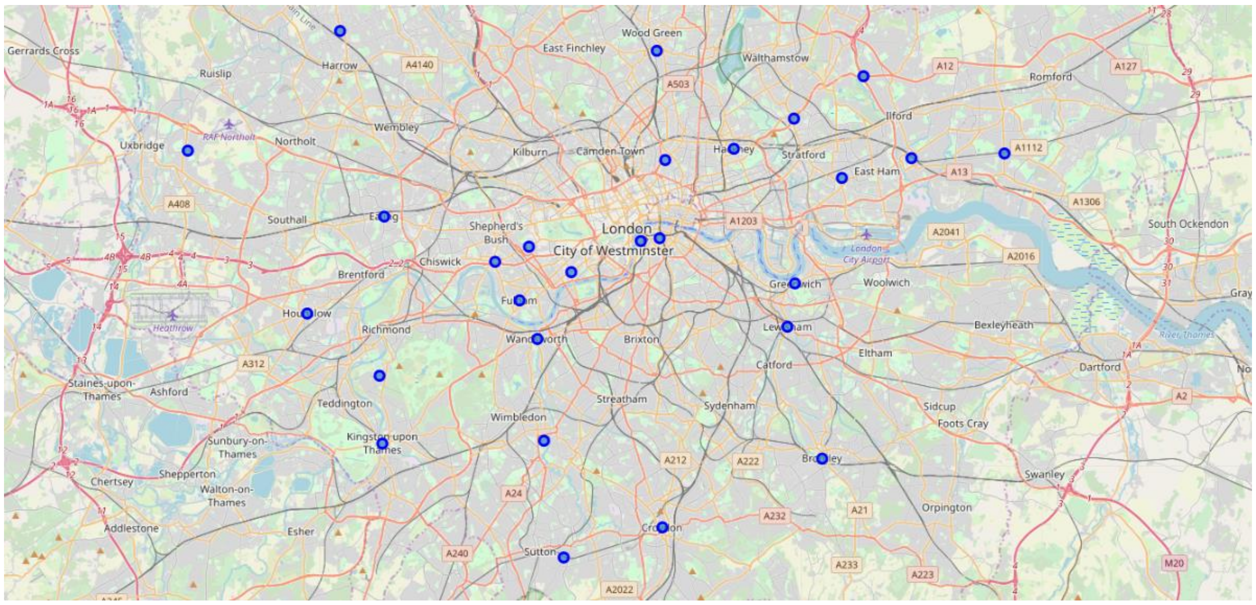


Figure 1: London map

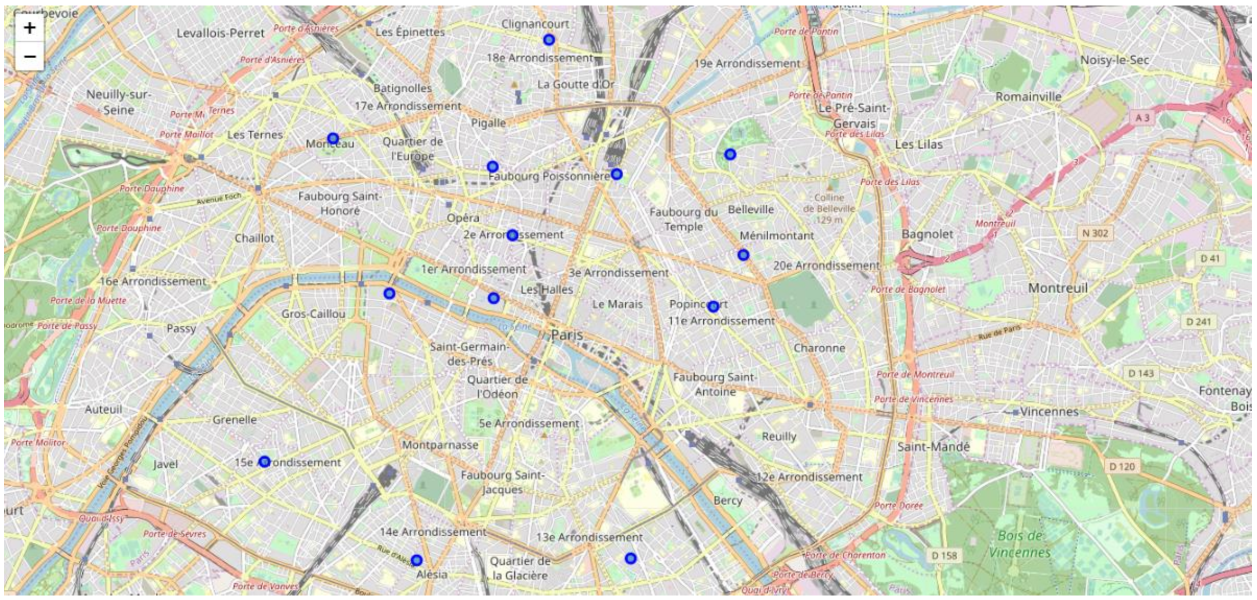


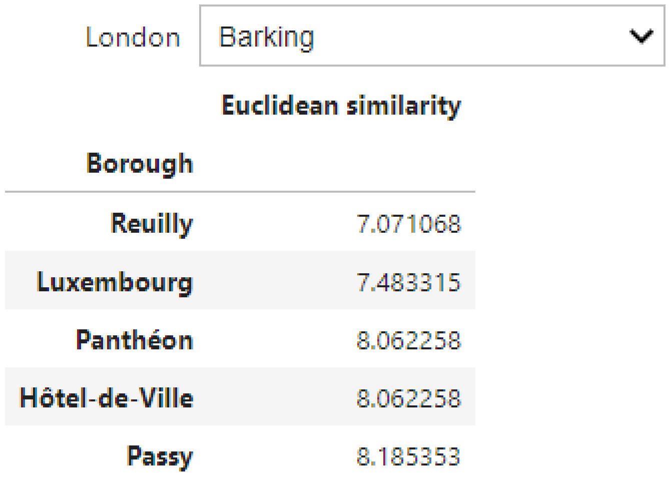
Figure 2: Paris map

Methodology

One-hot encoding method is used to convert the categorical values of the venues column in the dataframe to a numerical value. A column is created for each venue, if such a venue lies in the borough, a value of 1 is awarded, otherwise a value of 0 is given. Later on, the venues that are missing from London borough and exist in Paris boroughs are added to the dataframe and parsed with zeros, and the same is done for Paris venues. Finally, a similarity measuring algorithms, that employs the Euclidean distance and Cosine similarity, are used to compute the similarity between a given borough of London and all the boroughs of Paris, the results sorted by the similarity index, and the top five are selected and displayed to the user. The results of the two algorithms are discussed in the next sections.

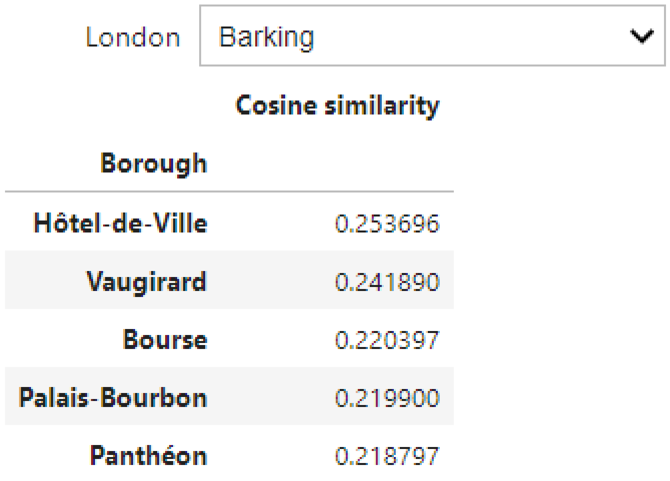
Results

The two algorithms used compute the similarity index, a sample result of each, selecting London's Barking borough as an input, are shown below. The results of the first algorithm are sorted ascendingly because the Euclidean similarity algorithm measures, in essence, dissimilarity, and not similarity, while the results of the Cosine similarity algorithm are sorted in a descending manner as the algorithm measures similarity, and hence, the larger the output the more similar the result is to the input.


London: Barking

Borough	Euclidean similarity
Reuilly	7.071068
Luxembourg	7.483315
Panthéon	8.062258
Hôtel-de-Ville	8.062258
Passy	8.185353

Figure 3: Euclidean similarity algorithm sample result


London: Barking

Borough	Cosine similarity
Hôtel-de-Ville	0.253696
Vaugirard	0.241890
Bourse	0.220397
Palais-Bourbon	0.219900
Panthéon	0.218797

Figure 4: Cosine similarity algorithm sample result

Discussion

It’s noticeable that the results of the two algorithms are somewhat significantly different, as only two of the boroughs suggested are common between the two algorithms’ results. In most cases less than three boroughs are common between the suggestions of the two algorithms, and many times, there will be only one borough or perhaps none they both recommend. This is due to the fact that Euclidean algorithm measures the distance between the vectors that represent the boroughs to be compared, while the Cosine algorithm measures the angle between the two vectors.

Therefore, if the two vectors were to be in the same direction, but with different magnitudes the first algorithm would still capture the difference between the two, while the latter would consider them completely similar. In other words, the first algorithm takes magnitude of the vector into account while latter doesn’t, and since the magnitude of the vectors vary from one to another, the similarity index should account for it. This leads to prioritizing the results yielded by the Euclidean distance, as it is more suitable to the type of data we are dealing with here.

Conclusion

In conclusion, the approach described here might useful in quantifying the similarity between cities or boroughs, however, it should be noted that only the venues in a specific borough are used to characterize the similarity, which might not be ideal as there are other parameters that should be considered such as the weather, how polluted it is, the language spoken and culture practiced by the locals, and others, and that could be the scope of the future works that would like to contribute to this problem.