Finding familiar (similar) Neighborhoods

Mohamed Nadhir DAOUD

04/08/2019

Introduction

As I was searching for ideas about this project, I encountered some websites that compares neighborhoods. with a closer look I found that these websites use general information about the cost of life or crime rates, but almost none of them tries to find similarities between the neighborhoods venues.

Let's now imagine someone who is moving to a new city, among other criteria, he may be interested in finding the most similar (familiar) neighborhood to his actual one. This project is about finding, in a new city, the neighborhoods most similar to the user actual neighborhood, based on the venue categories search.

It could be a web service for neighborhood comparison websites, as long as the website provide the list of neighborhood and related information (latitude, longitude).

Data

As one of the requirements of this assignment is to use Foursquare data, I decided to use the venues information from foursquare to compare neighborhoods.

But first I need the data of the neighborhoods of the targeted city and at least the data of the user actual neighborhood.

In the Notebook, and in ordered to illustrate how the app would work, I used New York city and Toronto neighborhood data. Both data could be found as csv files on my github repo:

https://github.com/Nadhir10/The_Battle_of_Neighborhoods/tree/master/Data

Methodology

After loading the cities neighborhoods data (neighborhood, Latitude, longitude), I look for the most frequent/common venue categories for each neighborhood.

I would have to explore/compare the categories of venues for both New York and Toronto.

Let's imagine that the user lives in a particular neighborhood in New York (chosen randomly), he would want to find the neighborhood in Toronto that has the most similar venues.

Now how would I compare the user actual neighborhood with the neighborhoods of Toronto. Actually, two main approaches came to my mind:

Approach 1: Euclidian Distance

Having tables of the frequency of each venue category for a given neighborhood can be seen as a multidimensional vector representing the neighborhood where every dimension is a venue category.

From that point of view, we could compute the distance between each neighborhood and the user actual neighborhood.

The most similar neighborhoods would be those with the lowest distance to the user neighborhood.

Approach 2: Clustering

In this approach we would add the user's neighborhood to the list of the other city neighborhoods. After performing a clustering, The most similar neighborhoods would be those with the same cluster label as the actual neighborhood.

Results

Comparing venue categories:

I find that almost 90% of Toronto venue categories appear in New York venue categories list.

However, and because NY has a much higher number of categories, only 57% of New York venue categories appear in Toronto venue categories list.

In other words, it would be easier for someone in Toronto to find a similar neighborhood in NY than the opposite. However, the app has to work for everyone, so I had to consider only the shared venues of the two cities.

It would be better to work on the most challenging case, which is someone in NY looking for a similar neighborhood in Toronto.

I chose randomly a neighborhood in NY, which happened to be **Prospect Park South**.

Approach 1:

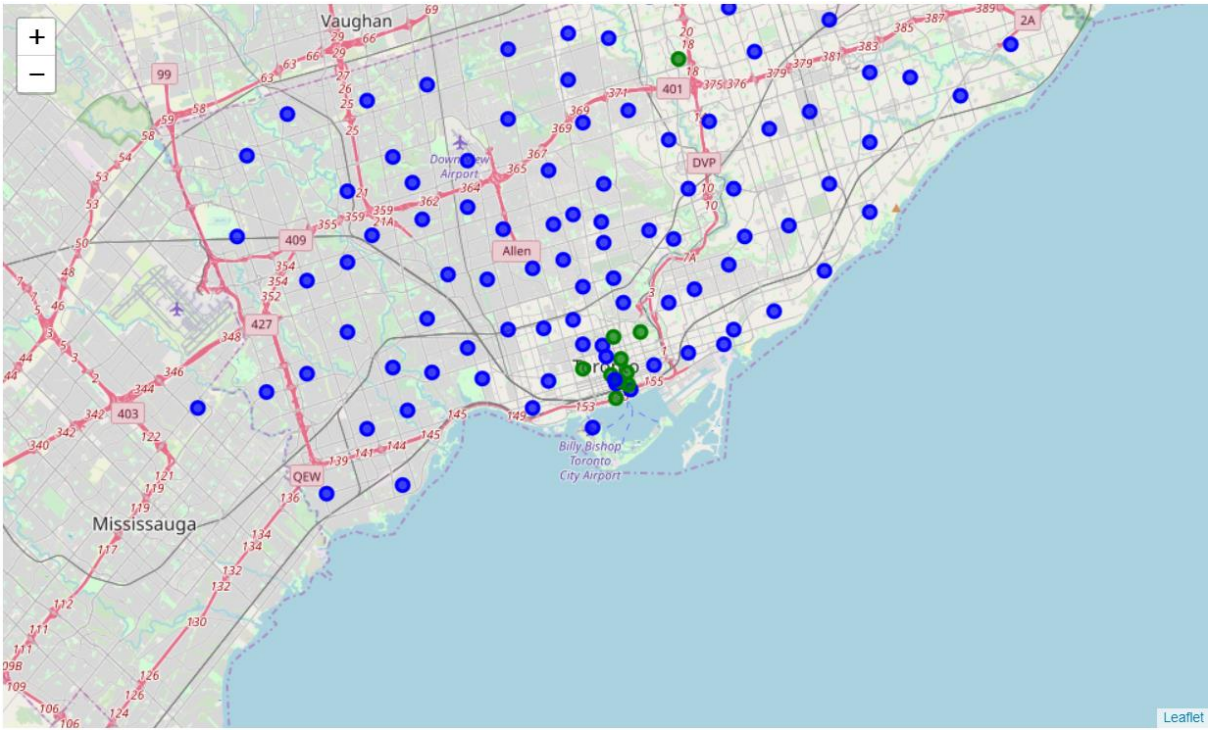
Now considering a neighborhood as a multidimensional vector where every dimension is the frequency of the venue category, we can compute easily the Euclidian distance, between two neighborhoods.

Later sort the neighborhoods of Toronto where we have the lowest distance (let's take the first 10):

Neighborhood	similarity	Nb of similar categories	Similar categories
Ryerson, Garden District	0.229408	2	[Pizza Place, Fast Food Restaurant]
Chinatown, Grange Park, Kensington Market	0.229459	0	[]
Church and Wellesley	0.236928	0	[]
St. James Town	0.242747	0	[]
Cabbagetown, St. James Town	0.243074	1	[Pizza Place]
Adelaide, King, Richmond	0.243694	0	[]
Harbourfront East, Toronto Islands, Union Station	0.244862	1	[Pizza Place]
Stn A PO Boxes 25 The Esplanade	0.245605	1	[Fast Food Restaurant]
Commerce Court, Victoria Hotel	0.254140	0	[]
Fairview, Henry Farm, Oriole	0.254142	1	[Fast Food Restaurant]

As we can see in the table above, having the lowest Euclidian distance doesn't necessarily mean that the neighborhoods share some of their top 10 common venue categories.

The following map shows in green the neighborhoods with the minimal distance to Prospect Park South, NY:



Approach 2:

First, we have to add the chosen neighborhood (Prospect Park South, NY) to the list of neighborhoods of Toronto.

Then, we perform a clustering and look at the cluster that contains the chosen neighborhood. The other neighborhood in that cluster are the most similar ones.

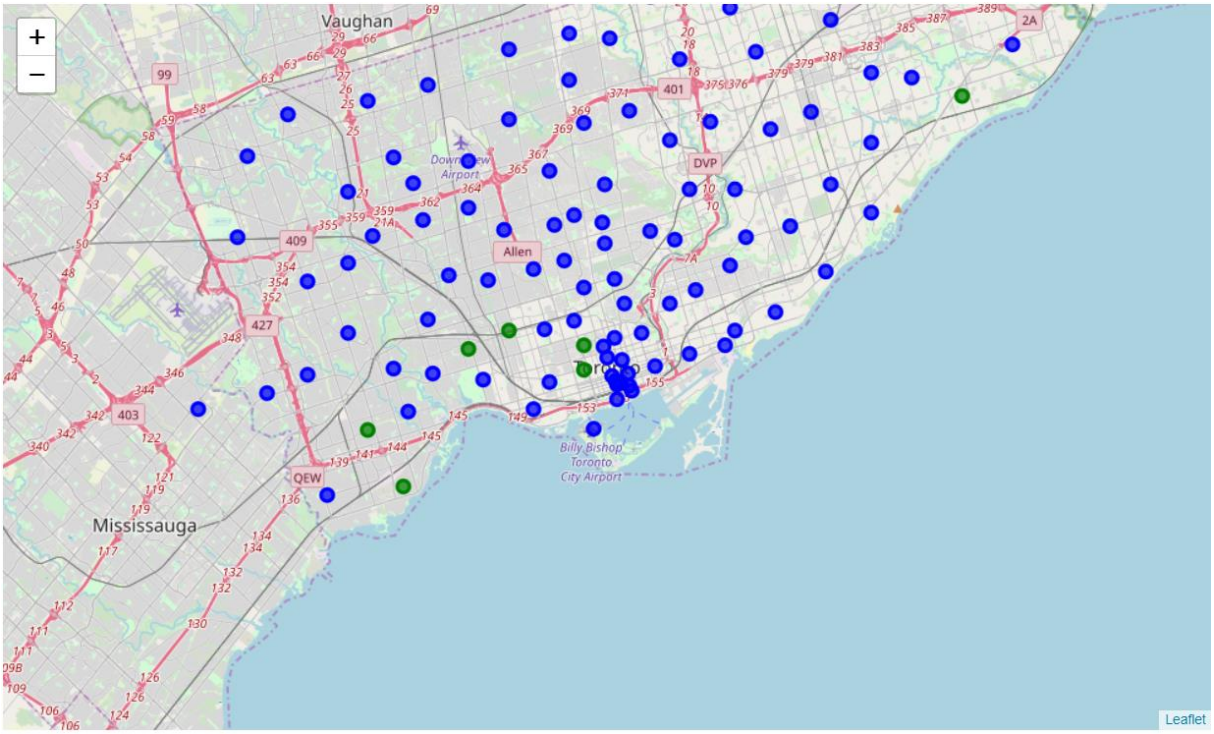
I choose to work with 48 clusters for the clustering, this number will be explained later.

After clustering, the cluster that contains ***Prospect Park South, NY*** is cluster N°2 and it contains 7 neighborhoods from Toronto.

Neighborhood	nb similar categories	Similar categories
Chinatown, Grange Park, Kensington Market	0	[]
Dovercourt Village, Dufferin	2	[Fast Food Restaurant, Pharmacy]
Guildwood, Morningside, West Hill	1	[Pizza Place]
Harbord, University of Toronto	0	[]
High Park, The Junction South	2	[Fried Chicken Joint, Fast Food Restaurant]
Humber Bay Shores, Mimico South, New Toronto	4	[Fried Chicken Joint, Fast Food Restaurant, Pi...
Kingsway Park South West, Mimico NW, The Queen...	1	[Fast Food Restaurant]

With the clustering, most of the results share some of their top 10 common venues. Still 2 neighborhoods, "Harbord, University of Toronto" and "Chinatown, Grange Park, Kensington Market", doesn't share any of their most common venue categories with

The results are shown in green in the map below:



Discussion

Surprisingly the two approaches show different results, and there is only one neighborhood that appears in the results of both approaches (Chinatown, Grange Park, Kensington Market).

While in the first approcah, the similar neighborhoods seems focused on one region, in the second approach the similar neighborhoods are spread through the city.

Now the question is: ***what is the best approach ?***

Well! it is hard to say, since every approach has its pro and cons,

In **approach 1**, as we consider each neighborhood as a vector of its most common venue categories, computing the euclidian distance between the vectors and finding those with the lowest distance seems appropriate.

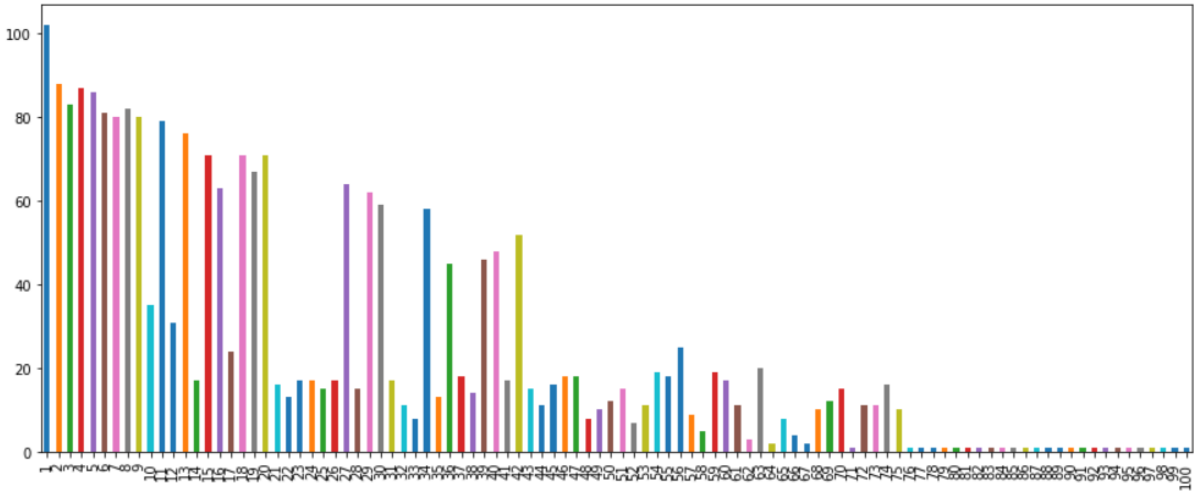
It's as if our selected neighborhood is the centroid of a general cluster and we are looking to its closest neighbors.

Now the good thing here is that we will always find similar neighborhoods, even though, when changing the selected neighborhood, we will have different distances, and it is hard to say when the results are good or bad.

In **approach 2**, we are building clusters with the hope that at least one cluster contains our selected neighborhood and enough other neighborhoods (not too much though), in that sense our neighborhood must be close the cluster centroid.

As Always with the clustering, the initial choice of the number of centroids is very important.

Depending on the number of clusters we choose, the number of elements of the cluster containing the randomly selected neighborhood, will change:



As we can see, the number of 48 clusters used earlier wasn't random, but I carefully choose it to have the lowest number of cluster elements while not increasing too much the number of clusters (below 50).

And as we have seen when we were discovering the most common venues of the other elements of the cluster, they can be quite similar.

However, with every new random neighborhood, we have to pick manually the number of clusters to be considered for the remaining part of the analysis.

Still, with this approach we were able to find a neighborhood that share 4 of its most common categories with the selected neighborhoods.

Conclusion

The clustering approach was able to find a more similar neighborhood sharing 4 of its 10 most common venue categories with the selected NY neighborhood.

Now the problem of the clustering is the initial number of clusters to use, we have seen i the previous example that it has to be picked manually for each new neighborhood.

the first approach (euclidain distance) has the advantage of being more systematic but fails to find as good results as the clustering.

Both approaches give some un-relaiable results.

In conclusion, despite the initial number of clusters issue, I prefer using the clustering technique.