Neighborhood recommendation for movers

Balázs Dedinszky

November 6, 2020

1. Introduction and Description of Business Problem

This is the Capstone project of IBM Data Science Professional Certificate. It was required to prepare a data science project that using geological data and *foursquare.com* information. As I live in Budapest (capital of Hungary) I wanted to include this city to the project. So I came up with the following idea: let's assume that someone in Budapest gets an opportunity to work in London, New York or Toronto. I want to recommend her/him similar neighborhoods to move with the family.

Just as in previous weeks I am going to calculate "similarity" by looking for similar facilities/venues in the neighborhood. I want to focus on facilities that are important to families with children, so I am going to give some facilities/venues (nurseries, schools, playgrounds, parks, medical centers) higher weight in this project. The final target is to prepare DataFrame that lists 5 most similar neighborhoods in London/New York/Toronto for each neighborhood in Budapest.

2. Description of Data

For this project I am going to use the following sources:

1. **Neighborhood names** that are geocoded with their centroid coordinates:

   - I am reusing the DataFrames for New York and Toronto from the previous weeks
   - For London neighborhoods I'm web scraping outer postal codes from this wiki page: https://en.wikipedia.org/wiki/London_postal_district
   - For Budapest neighborhoods it was not that simple. I found a Hungarian webpage from where geographical data can be downloaded (https://data2.openstreetmap.hu/hatarok/index.php?admin=10) I got an *.shp* file that must have been converted to neighborhood centroids and saved to a *.csv* file.

2. **Geocoding** London neighborhoods with *Google Maps API*. (The other cities are already geocoded in previous sessions) I'm using Google Maps as I realized that OpenStreetMap gives incorrect results if I put only postal codes as search keywords.

3. *Foursquare.com* searches for **facilities and venues** within the neighborhoods. The typical venues include restaurants, shops, bakeries, etc. I may use additional *Google Places* searches for special facilities (e.g. playgrounds, nurseries) if the original search

is not satisfactory. The disadvantage of Google Places search is that it provides only 20 results with one request so I will not fully replace Foursquare. Though I found Google more precise with non-profit (or governmental) facilities (e.g. clinics, libraries, playgrounds etc.)

With the list of the nearby facilities for each neighborhoods I'm going to calculate a similarity index among Budapest neighborhoods and other cities.

3. Methodology

As a preparatory step I had to collect the name and coordinates of London neighborhoods (Budapest, New York and Toronto neighborhoods with coordinates are already available from previous projects in .csv format).

Collecting the name of the neighborhoods

I found a useful neighborhood list on wiki page https://en.wikipedia.org/wiki/London_postal_district. The collection was done by Pandas library in-built web-scraping ability with read_html() function. Then the data went through some cleaning and analysis.
At the beginning I was wondering which approach is better for geocoding:

- using the name of the neighborhood as a keyword (as we did in New York), or
- using postal codes as search keywords (as in Toronto) During the analysis I realized that even though postal codes have more exactly defined area, they are not useful in this case. Unlike in Toronto, where one postal code covers one neighborhood most of the time, there is no such correlation in London:
- In some cases, neighborhoods have multiple postal codes
- Some postal codes span through multiple neighborhoods (even through Boroughs) Considering that it is more comprehensible to choose among neighborhoods than postal code areas I decided to use neighborhood names as search keyword even though it raised some ambiguity issues.

Geocoding neighborhood names

Geocoding was done with the names of the neighborhoods as keywords using Google Maps API. There are 2-2 neighborhoods in London that have the same name, but they are in different Boroughs (Church End in Brent and in Barnet, Grove Park in Hounslow and Lewisham). To differentiate them I added postal codes to these search keywords.

The latitude and longitude coordinates are stored in the DataFrame to use them in the next step.

Feature collection

To calculate similarity of the neighborhoods I prepared a set of features that can be compared. In this case (just like in previous lab sessions) I am interested in the number and types of venues/facilities (restaurants, shops, playgrounds, etc.) of the neighborhood. The main assumption of the project is that two neighborhoods are more similar to each other if there are similar types of facilities nearby with similar frequency. For example, a neighborhood with lots of bars and pubs are probably rather different from a neighborhood with lots of parks, playgrounds, and pizzerias only.

I used foursquare.com to collect the top 100 popular facilities for each neighborhood in Budapest, London, New York, and Toronto. The search was done in the center of the neighborhoods with 1000 m radius. The result is one-hot coded by type of venues and aggregated, so the output is a table that contains all the different type of facilities as columns and the number of that type in each neighborhood. I used these columns as features.

During previous similar assignments (e.g. clustering Toronto neighborhoods) I realized that there is one flaw of foursquare search that must be addressed. It returns the top 100 most popular venues in the area, that means that some important facilities of the area may be missed. Looking at the aims of foursquare application it is a reasonable assumption that pubs, shops, restaurants and hotels are over-represented in the results while some non-profit facilities (e.g. schools, playgrounds, etc.) get less focus, so they are more probable not on the top 100 list. As I described in the introduction my focus is on the needs of family with children, so this issue has to be addressed.

To reduce this effect, I added three additional features to the table that are: schools, playgrounds, medical facilities and pharmacies. I collected the number of these types in each area by using Google Places API. I found that it is easier to use than foursquare in keyword search.

Calculating similarity

After collecting the features for each neighborhood in every city I calculated similarity. I defined similarity as the Euclidian distance between two feature sets.

Scikit-learn library has a very convenient function pairwise_distances that simplifies the process. The function computes distances between two sets of entities provided they have the same feature sets. This way, the full city distance matrix can be calculated with one function call.
The prerequisite of the correct distance calculation is that two sets of entities have the same features. To achieve this, I used the intersection of the feature sets, i.e. I kept only those features (type of facilities) that can be found at least once in both cities. It has not reduced the number of features too much (Budapest-London: 298 common features, Budapest-New York: 314, Budapest-Toronto: 257).

As the original business problem was that someone wants to move from Budapest, 3 distance pairs matrices were calculated for further analysis: Budapest-London, Budapest-New York, Budapest-Toronto.

As Euclidian distance is a bit counter-intuitive (the lower the number the more similar two neighborhoods are, zero is the totally equal while there is no upper limit of dissimilarity) I created a "similarity index" for better comprehension, that is compares areas on a 1-100 scale where 100 means the most similar. It is done with a general normalization method.

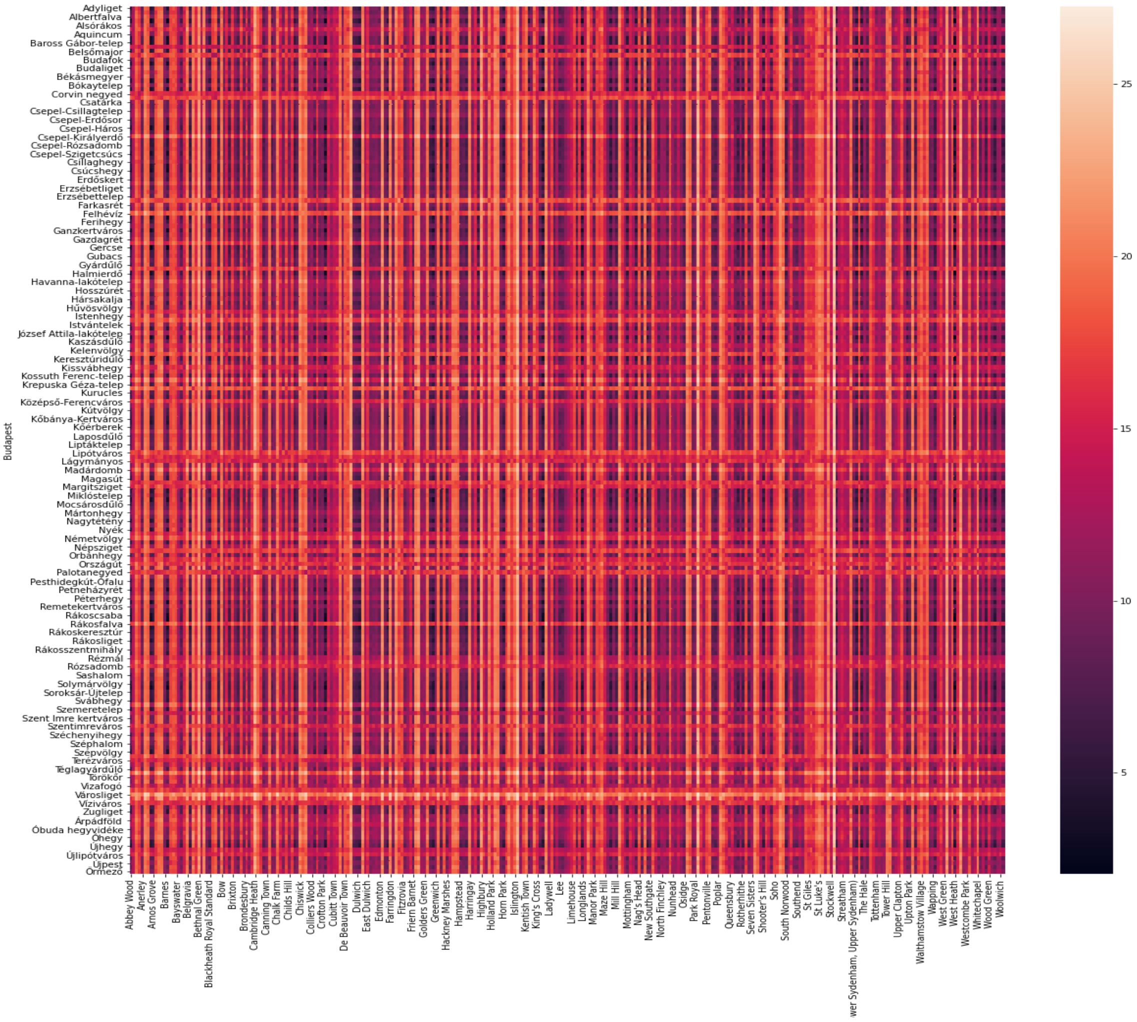
4. Results

As discussed in the previous section three matrices has been created and stored in DataFrames: Budapest-London, Budapest-New York and Budapest-Toronto. Basic data:

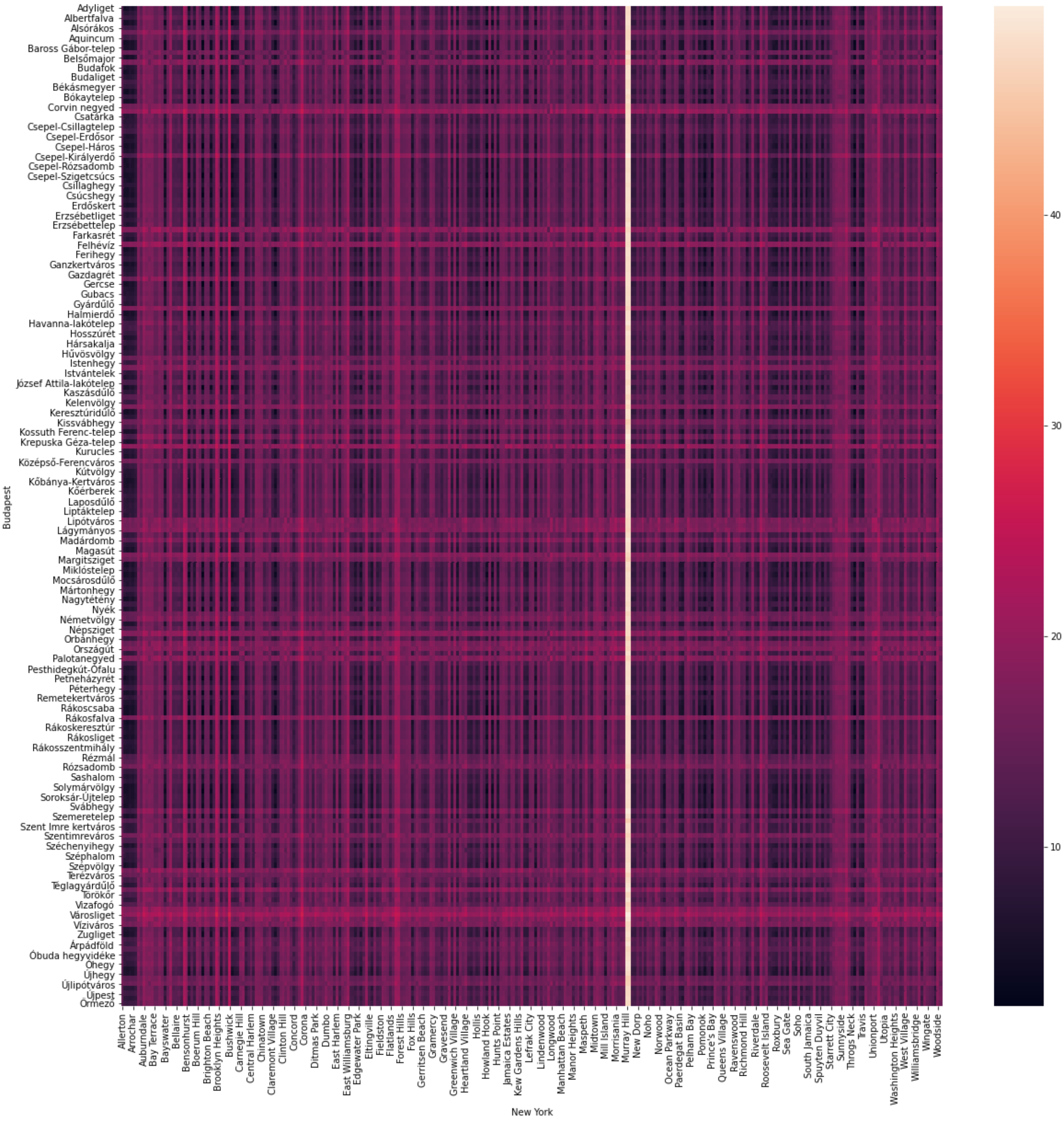
	Budapest-London	**Budapest-New York**	**Budapest-Toronto**
Average distance	14.53	14.62	12.87
Std deviation	4.52	4.73	4.3

I also created heatmaps from the distance matrices:

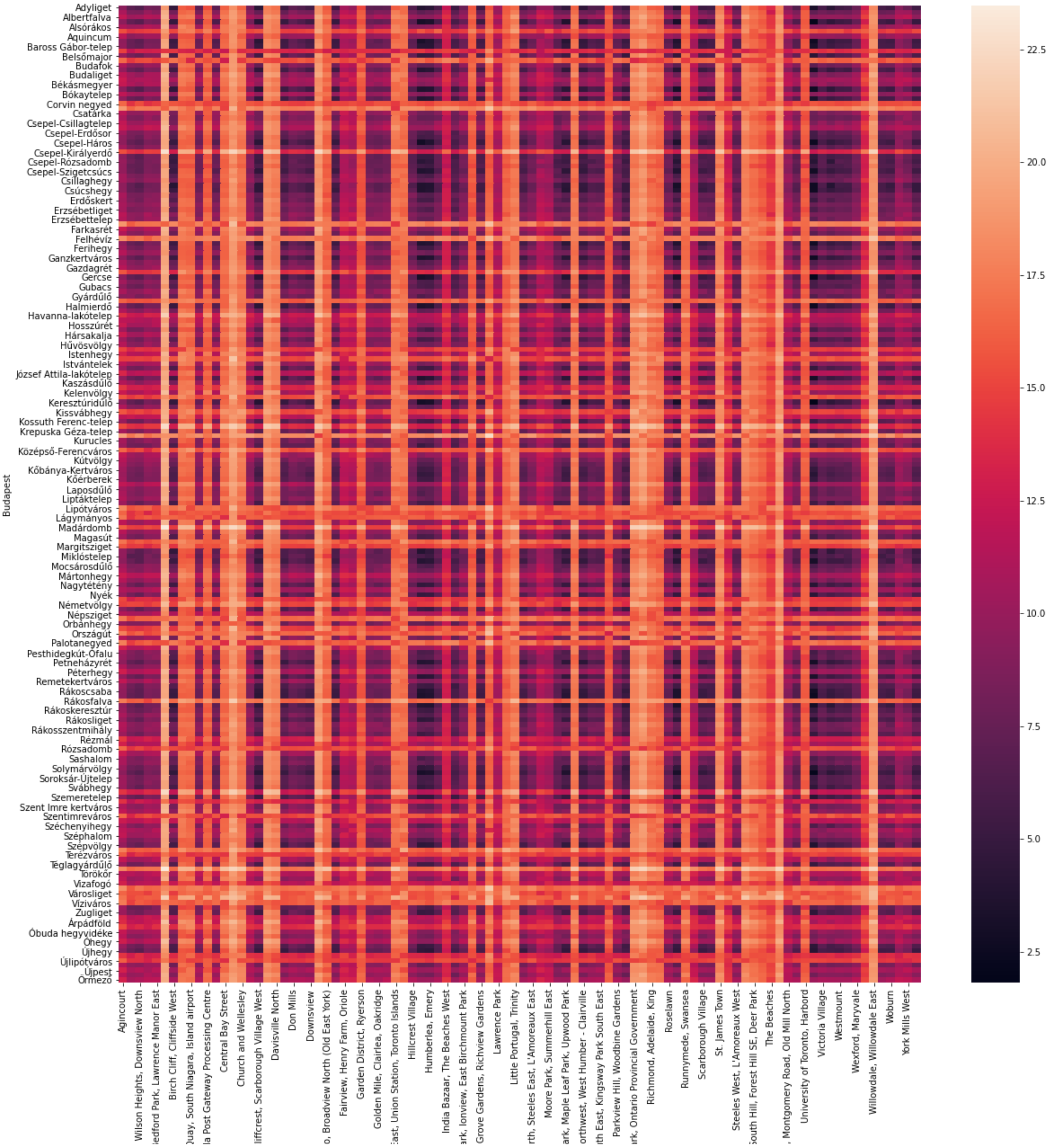
Budapest-London:



Budapest-New York:



Budapest-Toronto:



These heatmaps can be used for some quick check but they are not really used for practical purposes. To demonstrate the usability of the results I set up 3 scenarios:

- Someone wants to move from Budafok neighborhood (suburban area of Budapest)
- Someone wants to move from Terézváros neighborhood (inner city)
- Someone wants to move from Újpest neighborhood (in-between, mixed residential and industrial area) The question to be answered is which neighborhoods of London, New York and Toronto are the most similar to these areas.

With the distance matrices created it is easy to find the top 5 similar neighborhoods in each city, and also visualize similarity on map. (Please note that these are just pictures. You can find interactive maps with labeling in images folder of the project.)

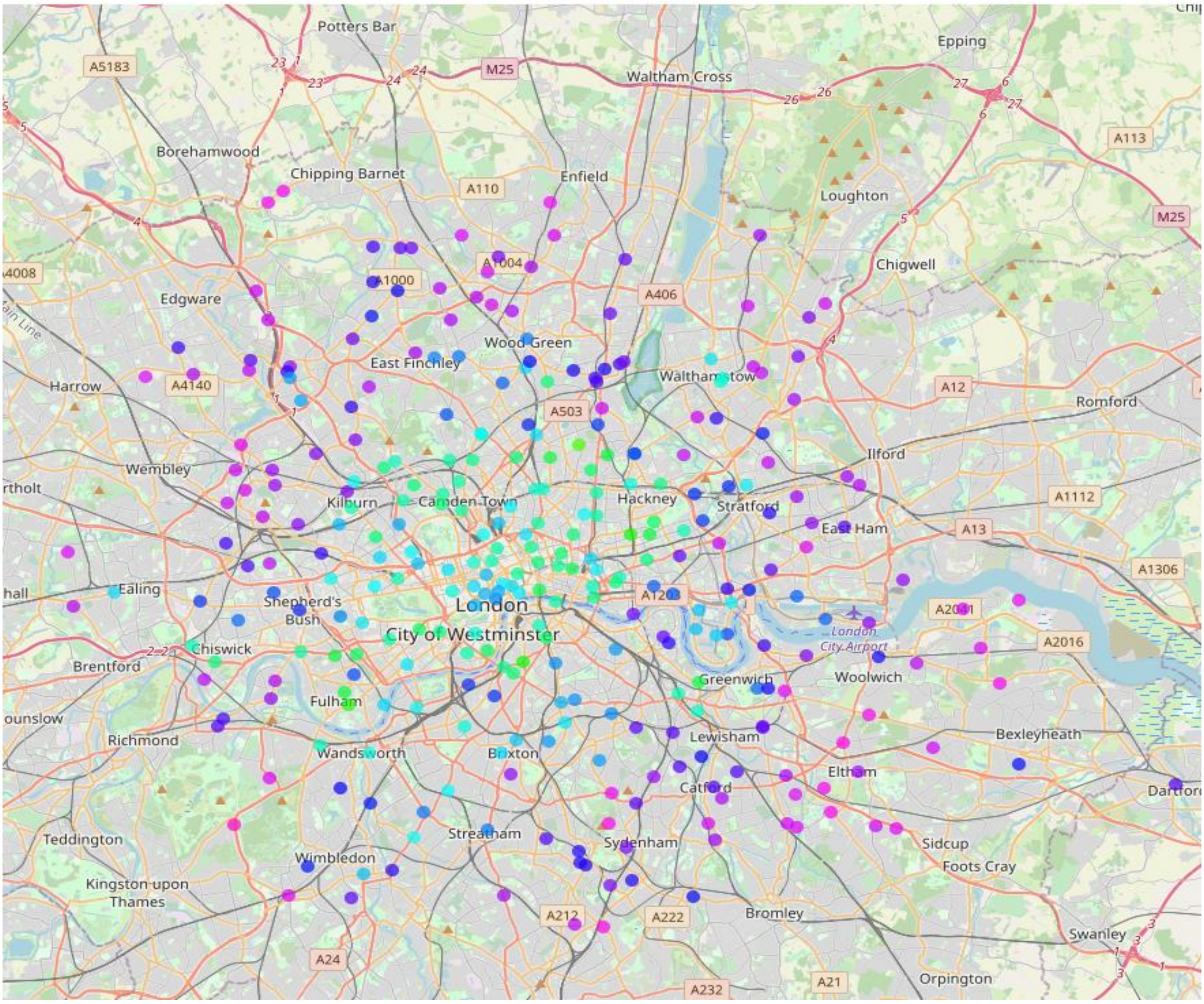
Budapest-London

Budafok Top five similar neighborhoods:

Neighborhood	Similarity Index
Kingston Vale	83.18
Sydenham Hill	80.29
Well Hall	79.72
Middle Park	79.36
Mottingham	79.18

Similarity map:

Budafok neighbourhood similarity index

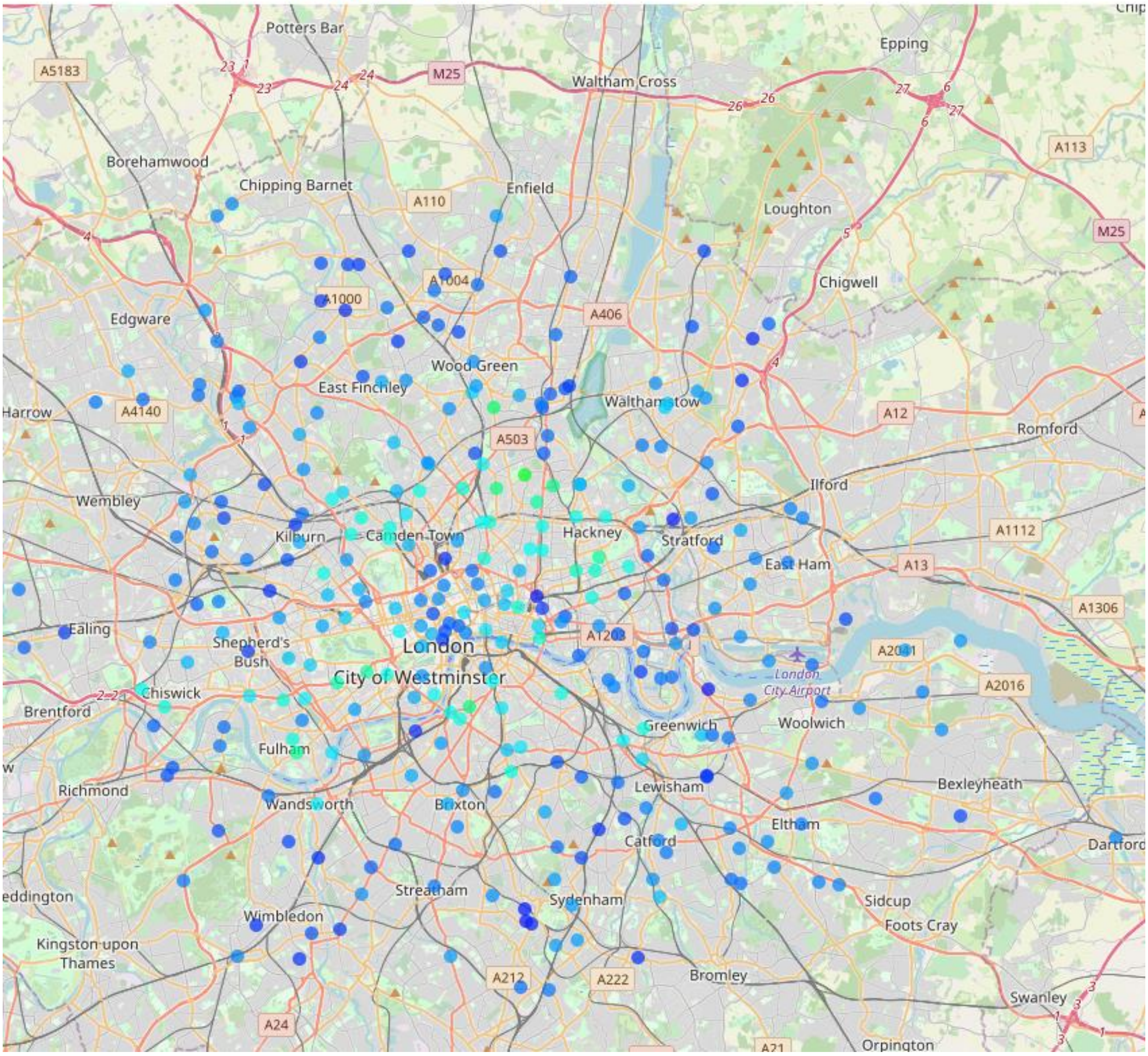


Terézváros Top five similar neighborhoods:

Neighbourhood	Similarity Index
Gipsy Hill	56.81
Shoreditch	56.63
Greenwich	56.37
Stratford	56.19
Crystal Palace	56.18

Similarity map:

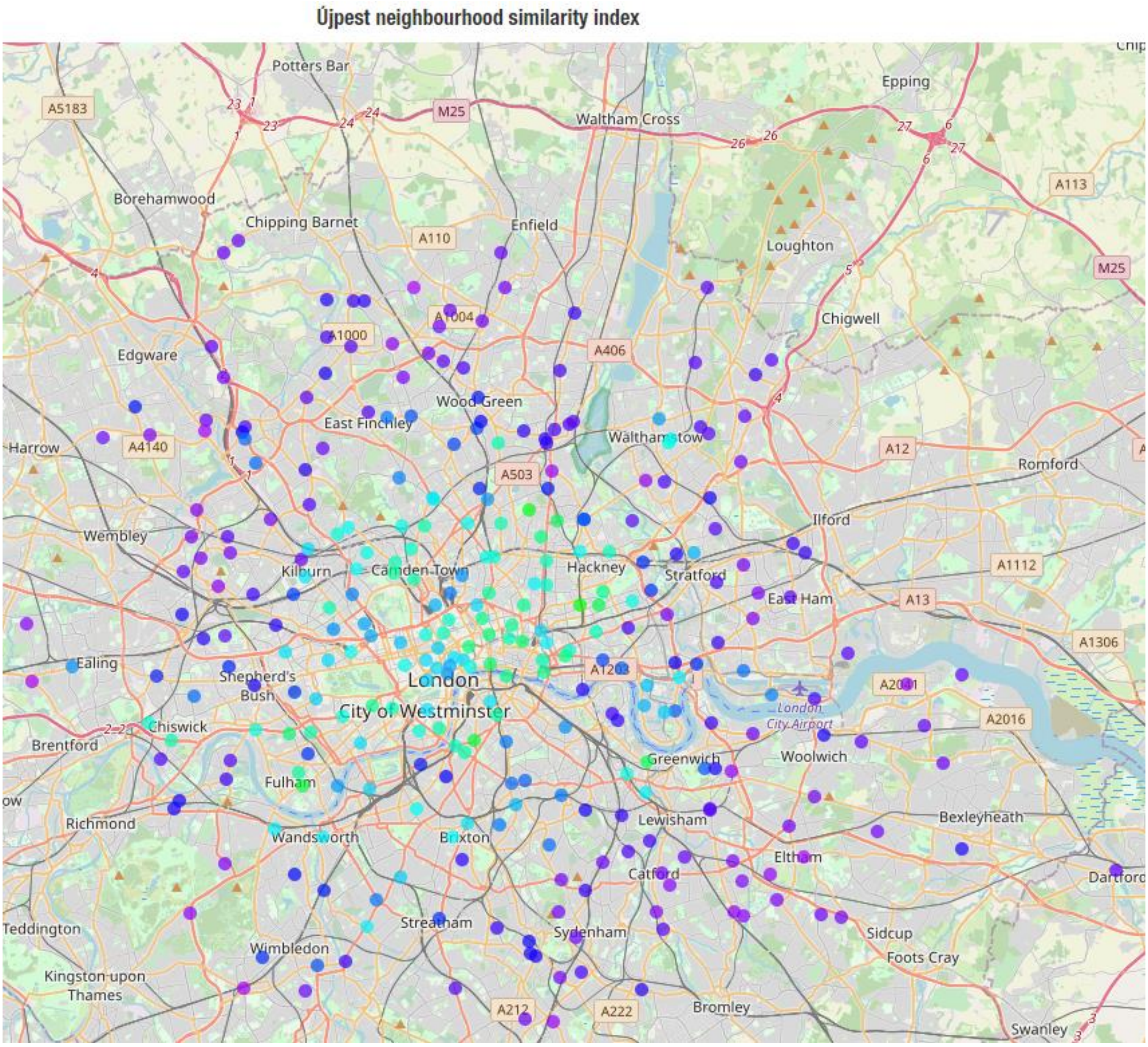
Terézváros neighbourhood similarity index



Újpest Top five similar neighborhoods:

Neighborhood	Similarity Index
Osidge	72.54
The Hyde	71.63
Hanwell	71.47
Raynes Park	71.40
Eltham	71.00

Similarity map:

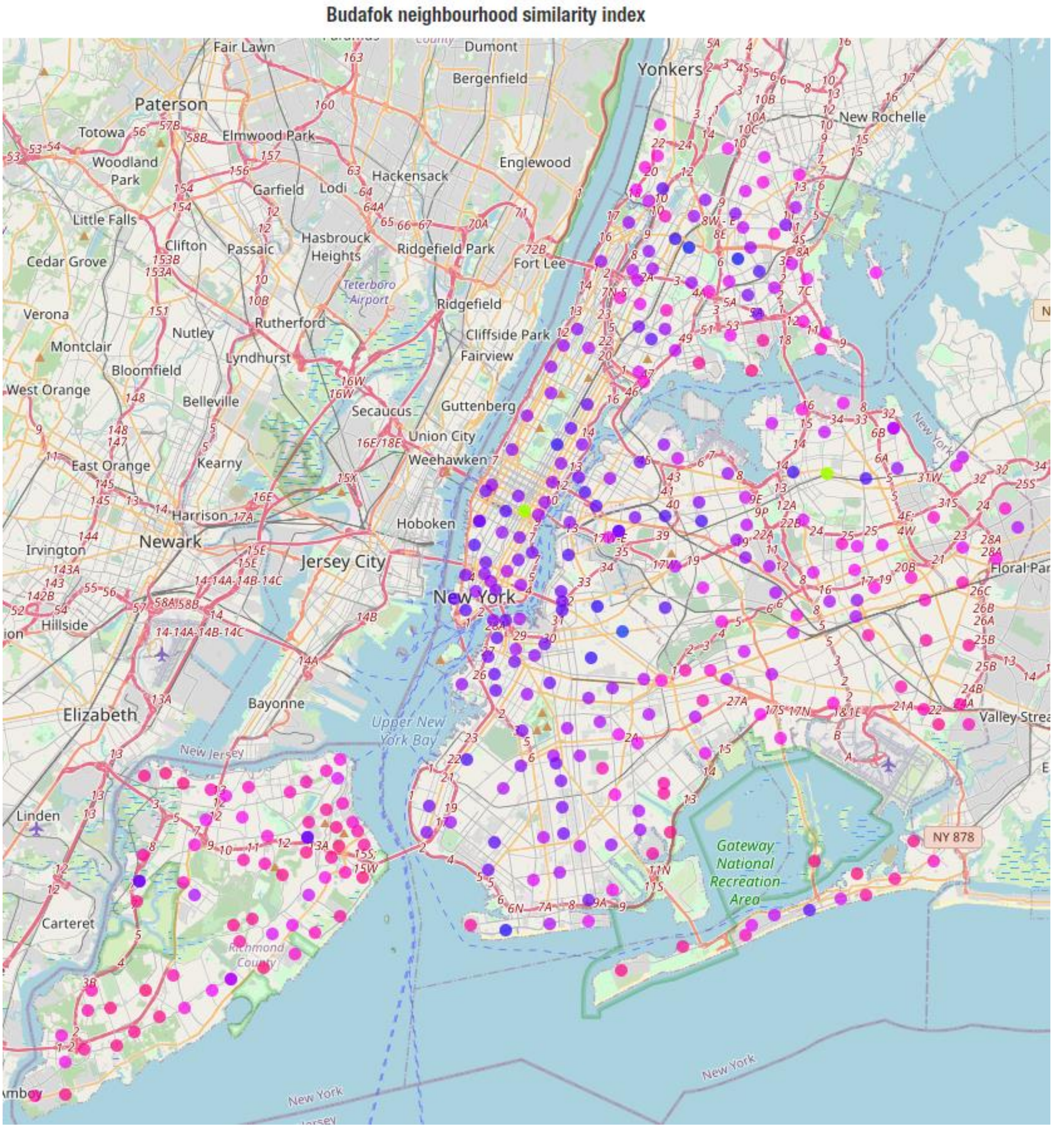


Budapest-New York

Budafok Top five similar neighborhoods:

Neighbourhood	Similarity Index
Park Hill	90.84
Arden Heights	90.22
Oakwood	90.13
Somerville	89.60
Arlington	89.43

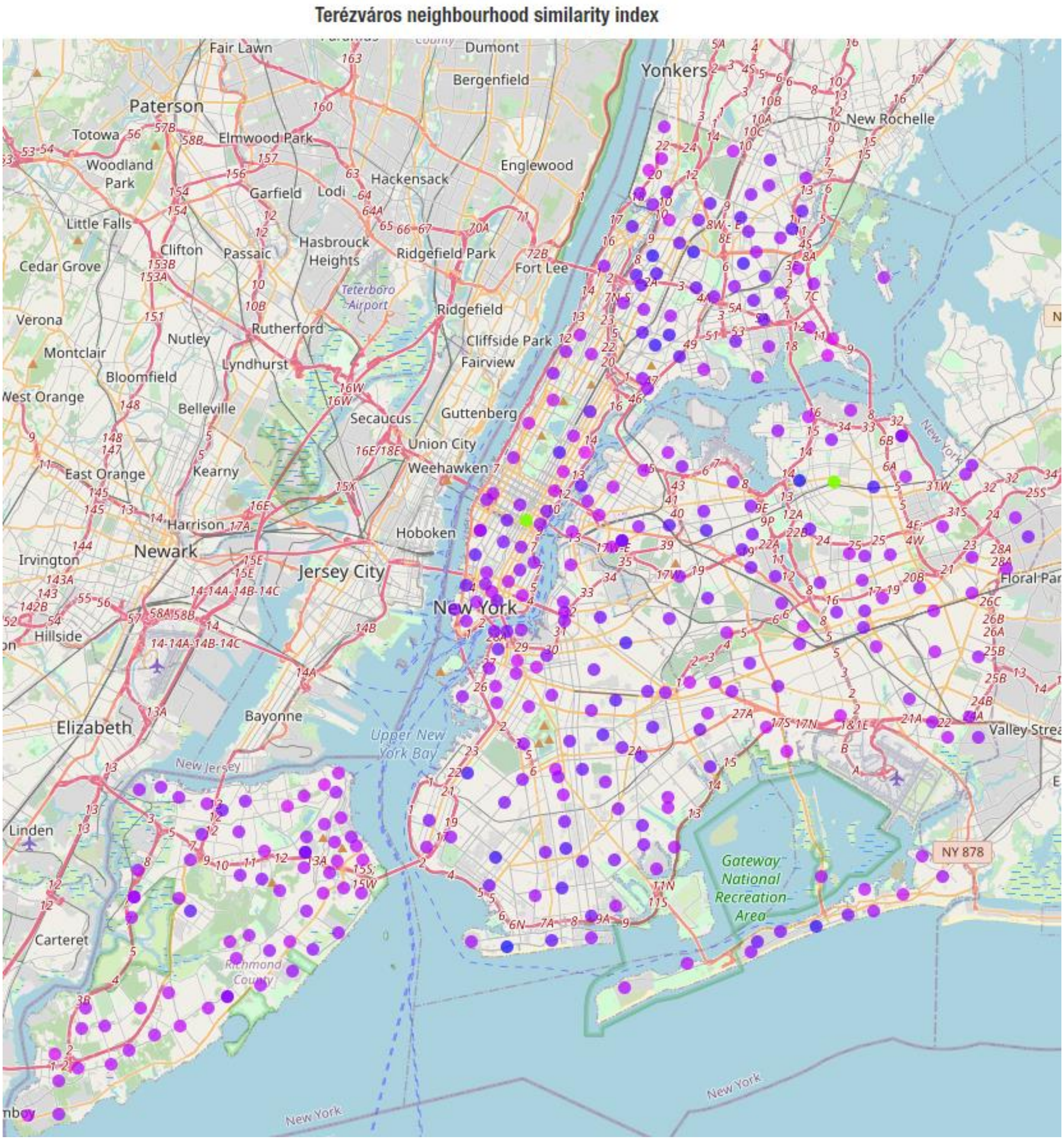
Similarity map:


Budafok neighbourhood similarity index

Terézváros Top five similar neighborhoods:

Neighbourhood	Similarity Index
Yorkville	76.67
Edgewater Park	75.91
Throgs Neck	75.73
Riverdale	75.66
New Brighton	75.48

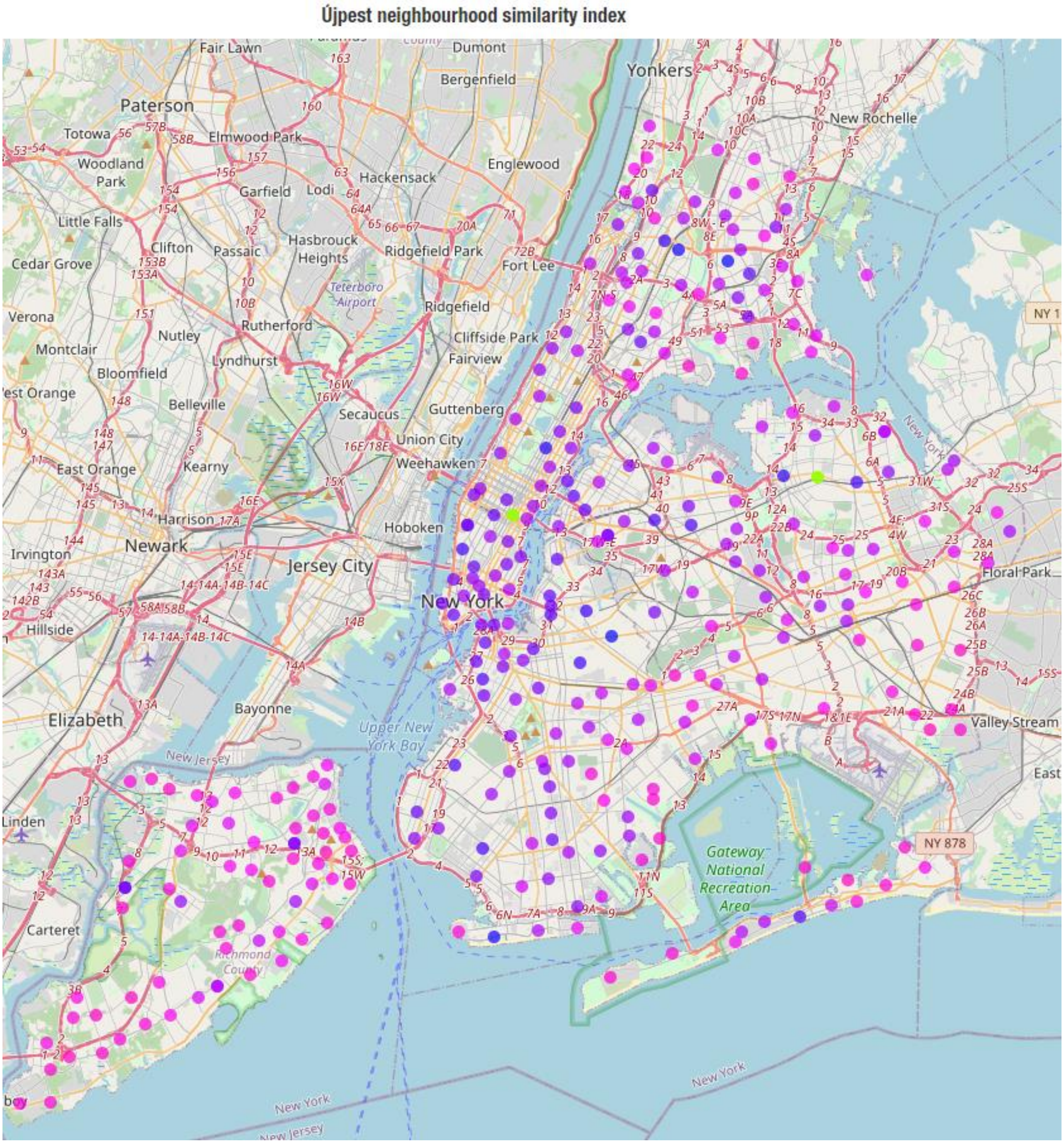
Similarity map:



Újpest Top five similar neighborhoods:

Neighbourhood	Similarity Index
Brookville	84.23
Hunts Point	83.95
Arlington	83.73
Somerville	83.65
Park Hill	83.65

Similarity map:



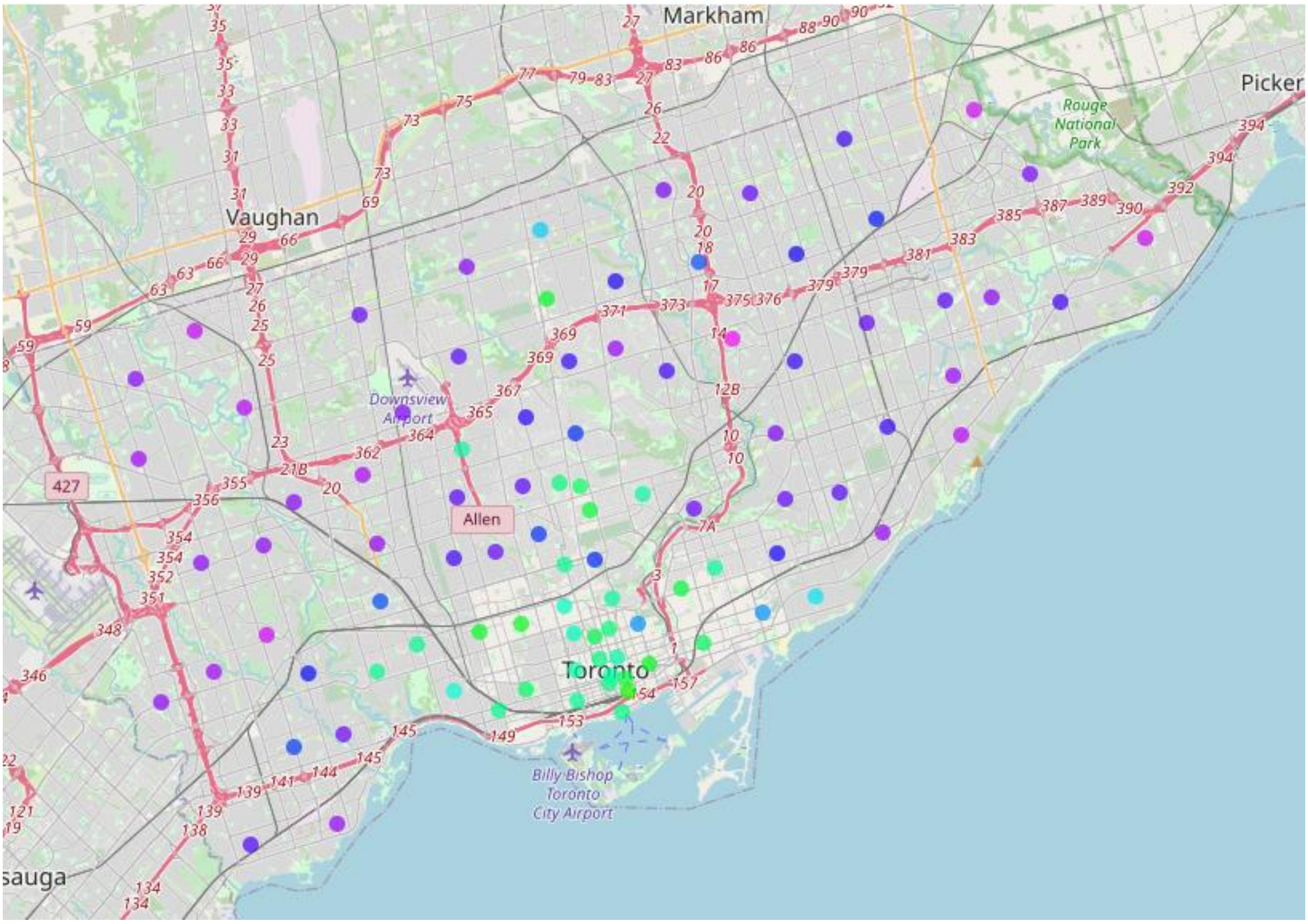
Budapest-Toronto

Budafok Top five similar neighborhoods:

Neighbourhood	Similarity Index
Parkwoods	78.61
Rouge Hill, Port Union, Highland Creek	75.29
Upper Rouge	75.27
Islington Avenue, Humber Valley Village	75.16
Humber Summit	74.13

Similarity map:

Budafok neighbourhood similarity index

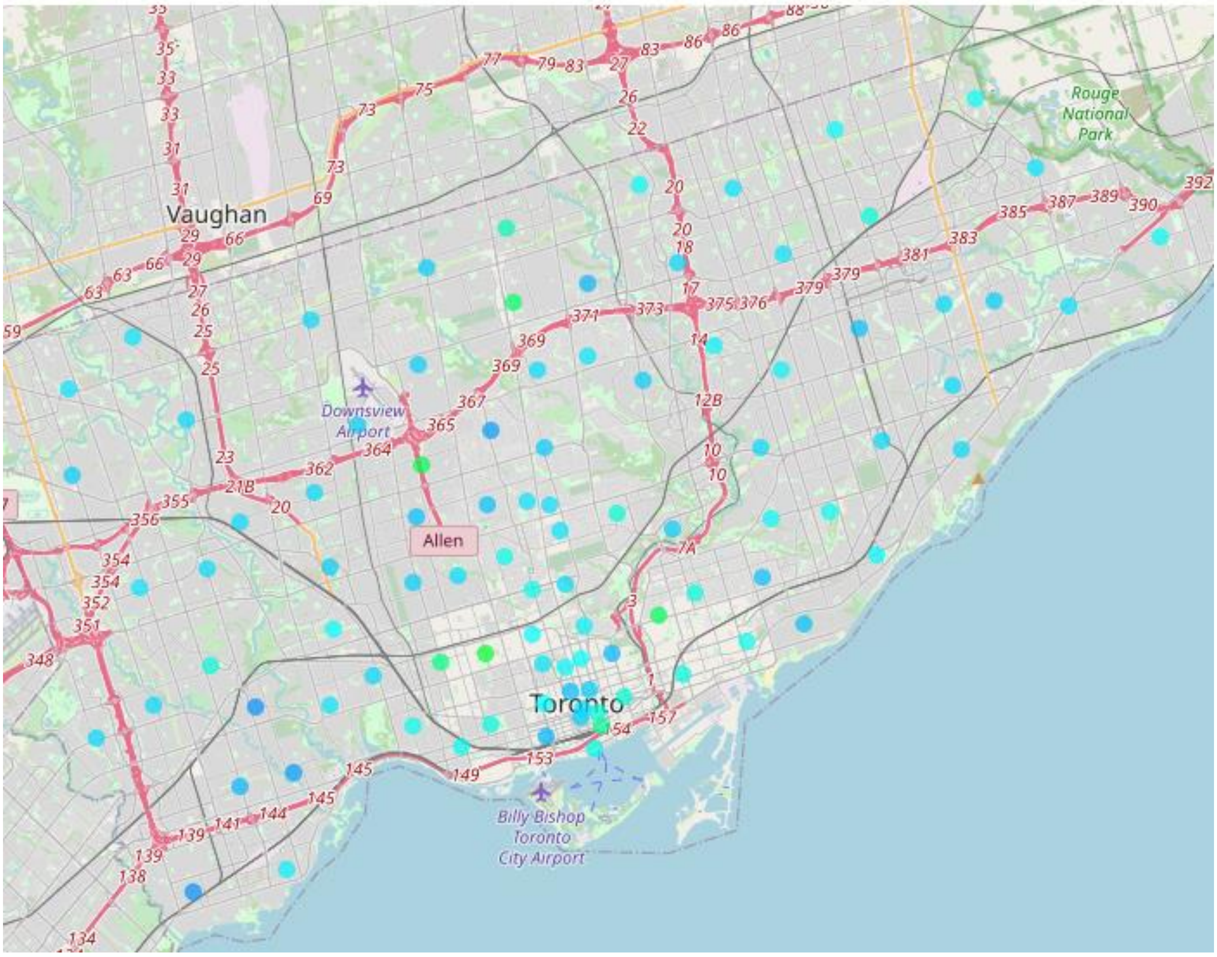


Terézváros Top five similar neighborhoods:

Neighbourhood	Similarity Index
The Kingsway, Montgomery Road, Old Mill North	46.86
Alderwood, Long Branch	46.72
Old Mill South, King's Mill Park, Sunnylea	45.69
Bedford Park, Lawrence Manor East	45.31
St. James Town, Cabbagetown	43.68

Similarity map:

Terézváros neighbourhood similarity index

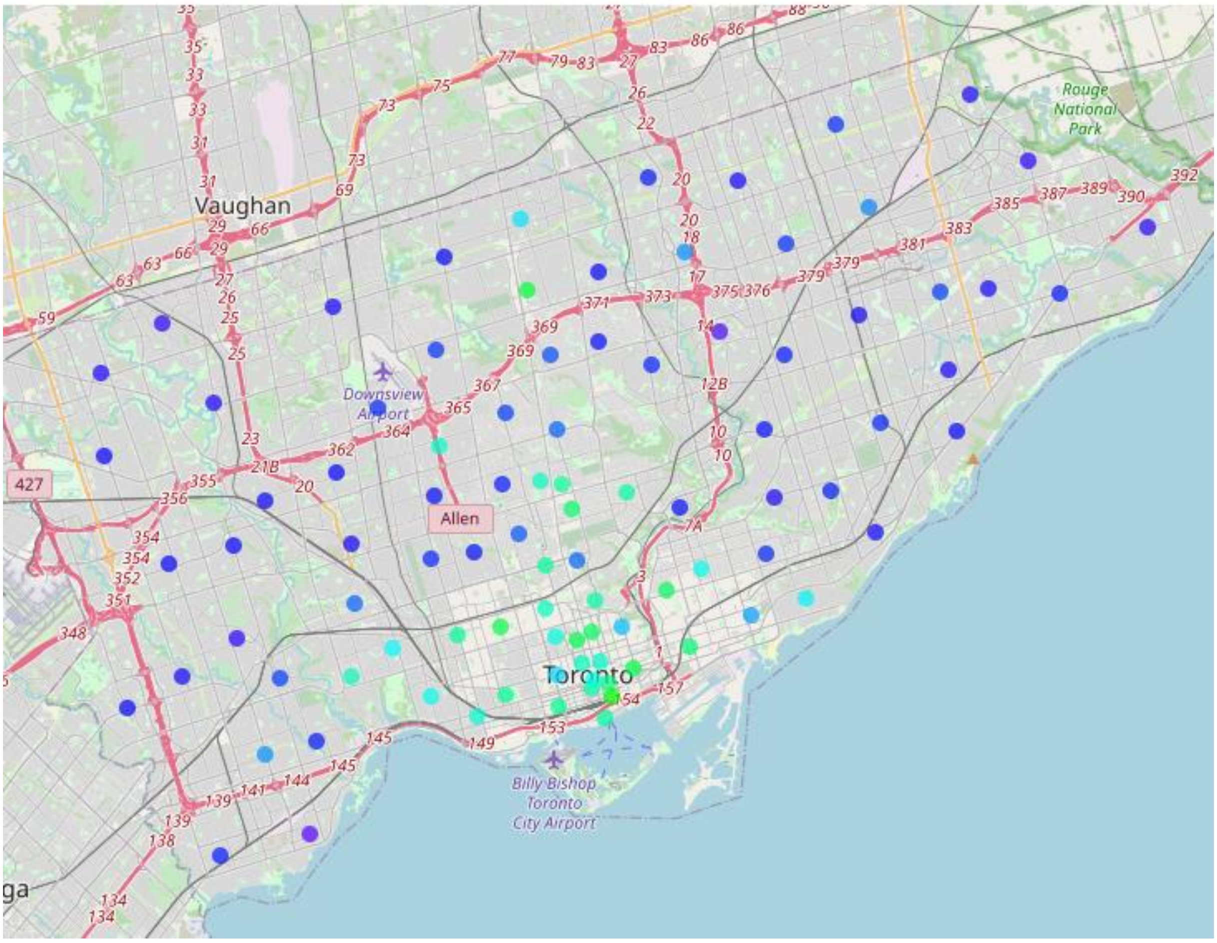


Újpest Top five similar neighborhoods:

Neighbourhood	Similarity Index
New Toronto, Mimico South, Humber Bay Shores	65.72
Parkwoods	64.34
Islington Avenue, Humber Valley Village	61.50
Malvern, Rouge	61.31
Humber Summit	61.24

Similarity map:

Újpest neighbourhood similarity index



5. Discussion

Overall findings

Based on the calculations and the visualizations I can see the following overall findings:

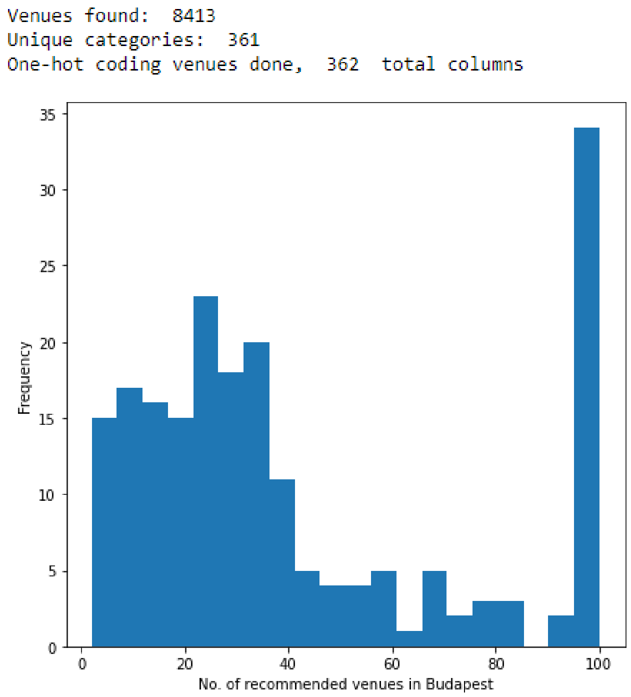
- The results are in line with expectations: a suburban area of Budapest is more similar to outer areas of the other cities, while inner city area is more similar to inner ones.
- It seems that Toronto in overall is more similar to Budapest than the other two cities. It is not surprising as Toronto is much closer to Budapest in terms of area and population than the other two megapolises.
- Even though the overall standard deviation is quite similar in each matrices, it is more difficult in New York maps to differentiate more similar and less similar areas in the three scenarios. See considerations below.

Some additional considerations

Based on the maps it seems that the model is more useful in London and Toronto than in New York, I'm trying to assess the reasons.

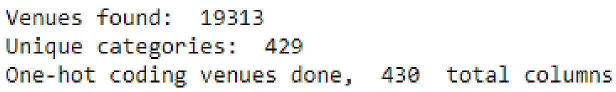
The first reason I think of is the limitation of the foursquare search. The search is limited to 100 results that means that if there are more than 100 facilities within an area some were lost. It is ambiguous which facilities are returned and which not so it is possible that these "lost" results have significant impact on results. I created histograms of returned results:

Budapest:

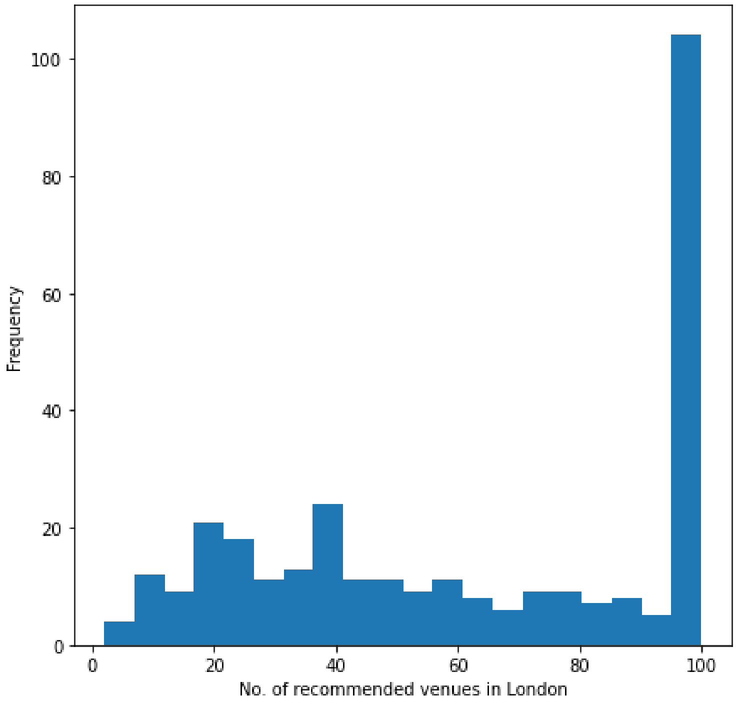


```
Venues found:  8413
Unique categories:  361
One-hot coding venues done,  362  total columns
```

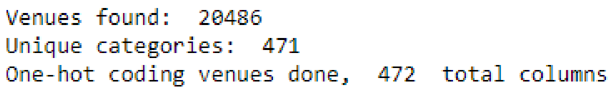
London:



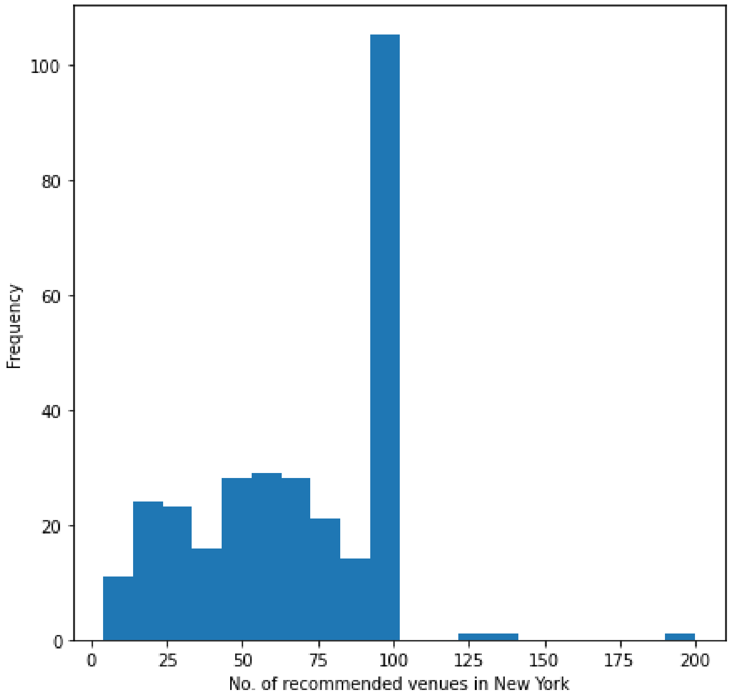
```
Venues found:  19313
Unique categories:  429
One-hot coding venues done,  430  total columns
```



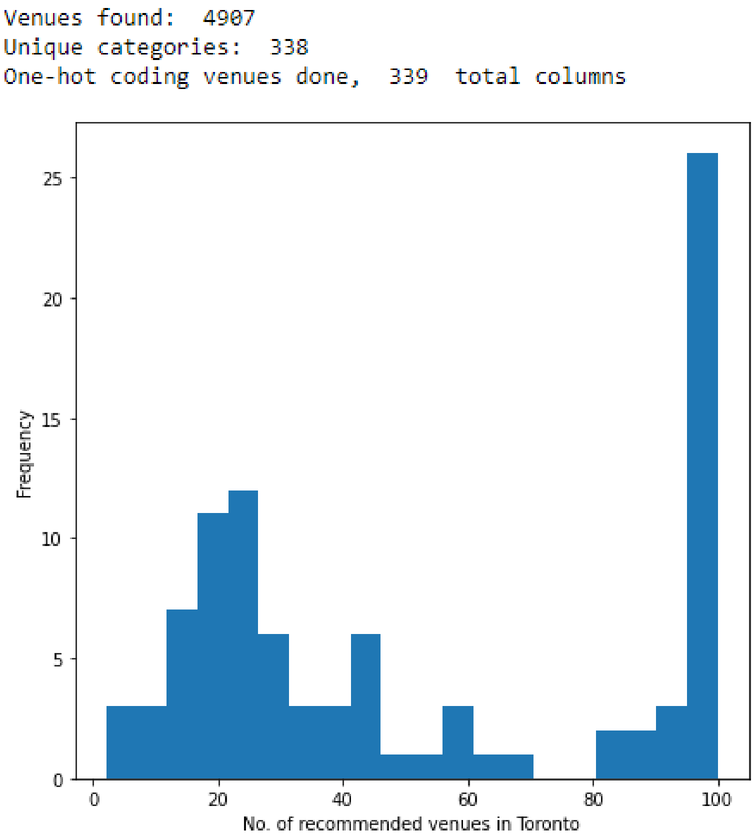
New York:



```
Venues found:  20486
Unique categories:  471
One-hot coding venues done,  472  total columns
```



Toronto:



```
Venues found:  4907
Unique categories:  338
One-hot coding venues done,  339  total columns
```

As you can see more than 20,000 venues was find in New York, while only 8,000 in Budapest, and 4,000 in Toronto. Also, only 17% of the searches capped at 100 results in Budapest, while it is 35% in New York. I tried to mitigate this issue by adding additional features (schools, etc. see in Methodology), but it cannot be fully negated this way.

Diversity of facilities could be another problem. Our method takes *“Restaurant”*, *”Italian Restaurant”* or *”Chinese Restaurant”* as different type of venues, which is at least questionable concerning our original business problem (i.e. where to move?). Bigger cities are more diverse so there may be more categories of similar type of facilities than in smaller cities.

Possible improvements

Results may be improved in some way, but the detailed analysis is out of this report’s scope:

- **Raising search limits**: By increasing limits of maximum results (100 in foursquare and 20 in Google) all relevant facilities can be collected. It would have been simple to do but I may have not fit into the free tier of the services.
- **Feature grouping**: As I mentioned above similar features (e.g. different kind of restaurants) should be treated as the same and grouped into one feature.
- **Additional features**: additional features can be collected from different sources, e.g. crime rate, pollution, population density, school results. The challenge with these

mixed features is standardization, how any of the features not to be overpriorized to others.

6. Conclusion

In this report I showed that it is feasible to find similar areas in different cities using coordinate-based searching services (Foursquare, Google Places). Similarity index seems reliable to build a recommendation system onto it. The method may be improved in several ways but a viable solution on our original business problem was successfully accomplished.