A Simple Recommender System for Pseudo Neighborhoods

Kevin Rayco

May 18, 2020

I. Introduction

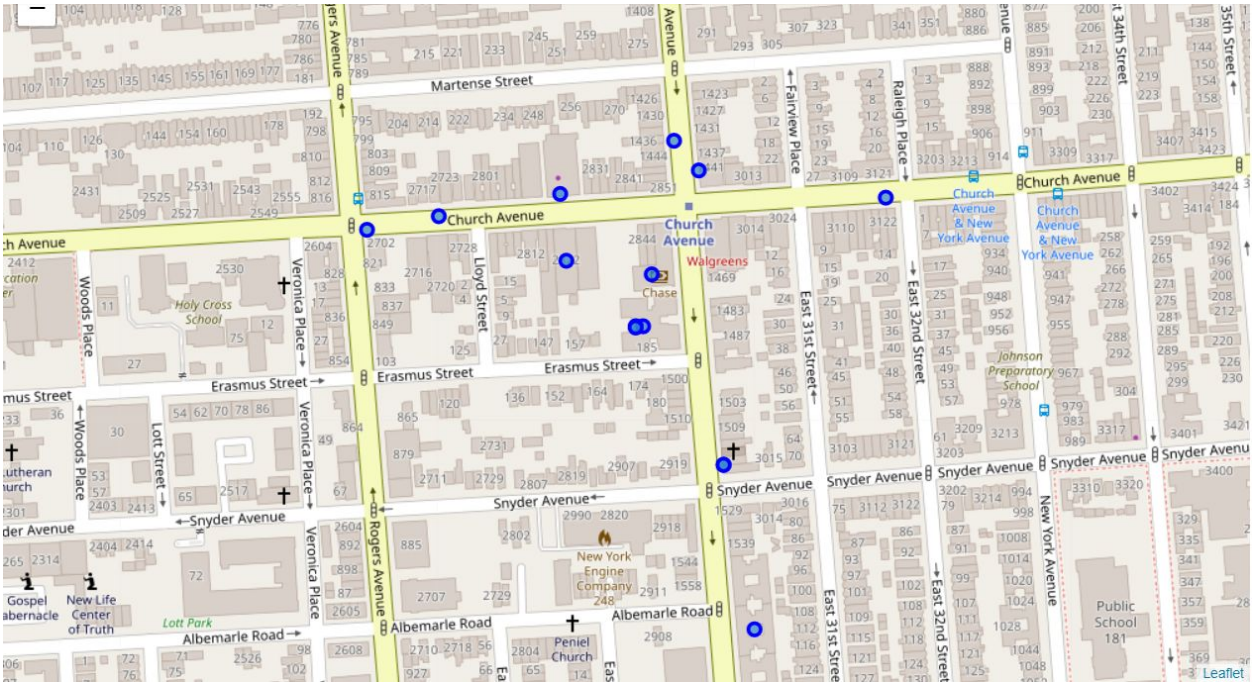
Suppose that a person plans to move from their own country to an entirely different country for a new job. Moving to a new place can sometimes be difficult, especially if it is an unfamiliar and strange place. Let's assume that one way to ease the moving process is to make sure that the new place is somehow familiar or similar in terms of the neighborhood.

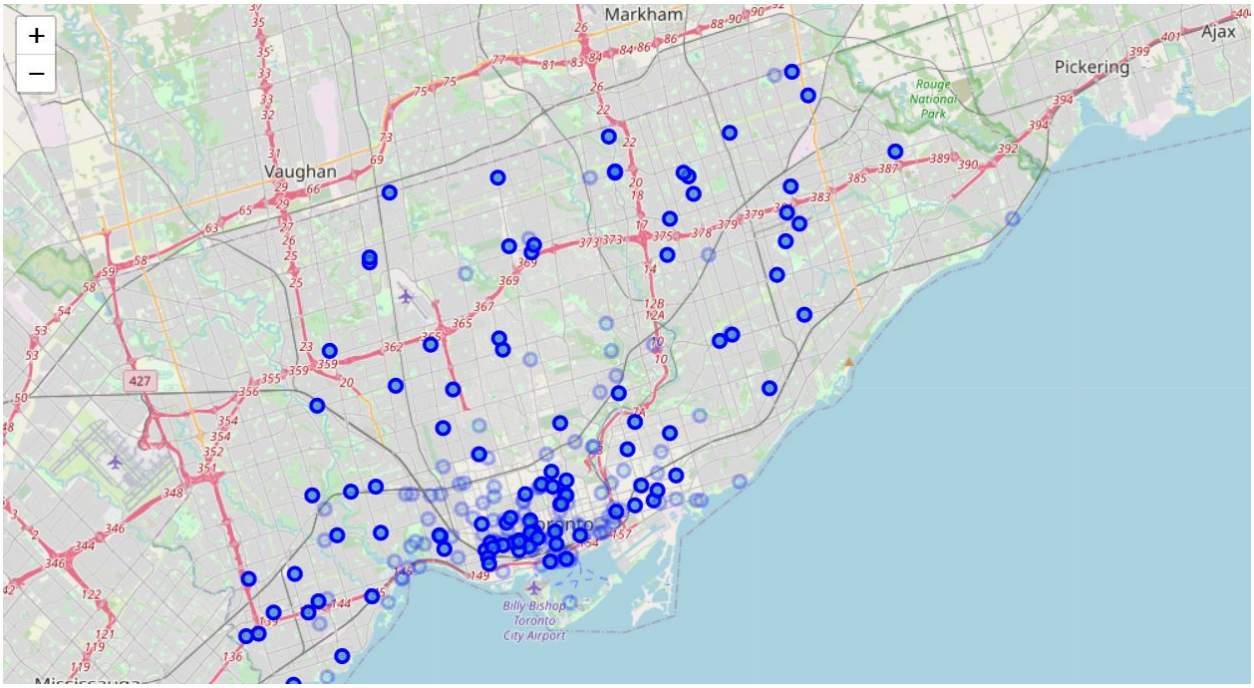
This project is a simple recommendation system that gives a ranked list of pseudo neighborhoods within a specified target location based on the similarity of venues in the neighborhood with an origin location.

II. Data

Similarity of neighborhood will be influenced by the presence of venues with the same category between the original neighborhood and each of the target neighborhoods. With that, we only need few types of data: the coordinates of origin and target location, and venue information such as venue ID, venue name, geographical coordinates, venue category, and category ID. Data will be extracted from and provided by GeoPy for the coordinates and Foursquare API for the venue information.

For the sake of discussion, let’s use Brooklyn, New York and Toronto, Canada as the origin location and target location respectively. We obtained the 15 closest venues from the center of Brooklyn and all 222 results within Toronto. Let’s have a look at the map and venues of Brooklyn and Toronto respectively.





III. Methodology

To build the recommender system, we want to define the multiple classes or neighborhood profiles, and then classify probabilistically the origin neighborhood profile against those.

We want a recommender system for neighborhoods without relying on physical boundaries. A set of venues of interest might be in between of two or more neighborhoods, so a pseudo neighborhood might be a better choice than a physical neighborhood for this system.

Two clustering algorithms were considered for this system, DBSCAN and k-means. We want the target neighborhood to have nearly the same area as the original neighborhood, so we chose to use k-means over DBSCAN as the latter has a tendency to include nearby points or venues that might have belonged to a different cluster.

For the number of clusters, we set it to the ratio of the standard deviation of latitude and longitude of all venues within the target neighborhood over that of the origin neighborhood, which is approximately equivalent to 42 as shown below.

```
Number of clusters: 42
```

	Neighborhood ID	index	Venue ID	Name	Latitude	Longitude	Category ID	Category
216	5	227	4bd444f5462cb71393ebdf07	Francesca Bakery	43.787716	-79.256852	4bf58dd8d48988d16a941735	Bakery
217	11	228	5675fb22498e98ce8baff885	Crown Pizza	43.824307	-79.247393	4bf58dd8d48988d1ca941735	Pizza Place
218	11	229	4aef94c7f964a52060d921e3	Walmart	43.833671	-79.256036	4bf58dd8d48988d10f951735	Pharmacy
219	8	230	4e0b137722713e13018e7117	Tim Hortons / Esso	43.801863	-79.199296	4bf58dd8d48988d148941735	Donut Shop
220	28	231	53669066498eb02e34d50dcd	Yoga Grove - Small Classes. Big Difference.	43.649227	-79.506812	4bf58dd8d48988d102941735	Yoga Studio

A neighborhood profile in this system is defined by the presence of venue of interest or venues categories that are present in the original neighborhood. The target neighborhood profiles look like the figure below.

Finally, we will use Gaussian Naive Bayes to classify probabilistically the origin neighborhood profile against multiple target neighborhood profiles.

```
Target Neighborhood Profiles:
```

	Neighborhood ID	Playground	Gym / Fitness Center	Bank	Yoga Studio	Furniture / Home Store	Donut Shop	Juice Bar	Caribbean Restaurant	Pharmacy	Bakery	Pizza Place
0	0	0	1	0	1	0	0	0	1	1	1	0
1	1	0	0	0	1	0	0	1	0	0	1	1
2	2	0	0	1	1	0	0	0	0	1	1	0
3	3	0	0	1	0	0	0	0	0	1	1	1
4	4	0	0	0	0	1	0	0	0	1	1	1

IV. Analysis

Neighborhood profile similarities are calculated by presence of venues of interest using Gaussian Naive Bayes, which is used for multiclass probabilistic classification. The predicted probability in this set up tends to result in extreme values of one and zeros, so we will rank each neighborhood profile according to log-probability instead. We should observe values from negative infinity (zero probability) up to 0 (equivalent to 100% probability).

	Log Probability	Neighborhood ID	Mean Latitude	Mean Longitude
0	0.000000e+00	0	43.669676	-79.388641
1	-2.075294e+09	1	43.645195	-79.418322
2	-2.075294e+09	2	43.648460	-79.399714
3	-2.075294e+09	3	43.776172	-79.256715
4	-4.150588e+09	4	43.658614	-79.408111

Since K-Means Clustering does not guarantee convergence to the global optimum, each run of k-means will produce differing mean coordinates of the high ranking neighborhoods, but these neighborhoods will always place higher than the rest.

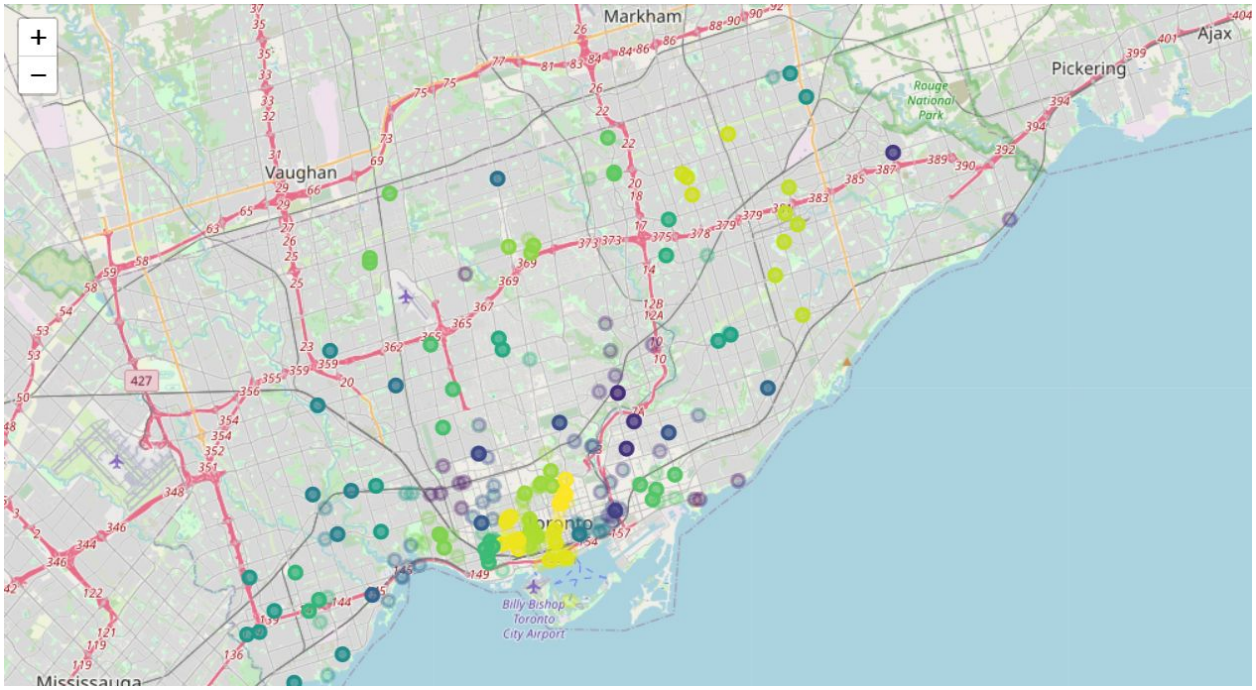
Based on the log-probability estimates, let's look at the rearranged profiles by rank. We should observe more ones at the higher rows compared to lower rows.

Rearranged Target Neighborhood Profiles:

	Neighborhood ID	Playground	Gym / Fitness Center	Bank	Yoga Studio	Furniture / Home Store	Donut Shop	Juice Bar	Caribbean Restaurant	Pharmacy	Bakery	Pizza Place
0	0	0	1	1	1	0	0	0	1	1	1	0
1	1	1	0	0	1	0	0	0	0	1	1	1
2	2	0	0	0	1	0	0	1	0	0	1	1
3	3	1	0	0	1	1	0	0	0	0	1	0
4	4	0	0	0	0	1	0	0	0	1	1	1

V. Results & Discussion

Let's view the map to see the neighborhood ranking colored by rank of log probability. We use viridis colormap to color each neighborhood. Higher ranking neighborhoods tend to be more yellow, while lower ranks tend to be more blue. Venues that are not of interest are still shown but in lower opacity.



From here, we can observe the high ranking neighborhoods from this as desired.

VI. Conclusion

The interested users and stakeholders can already utilize this system as is and gain desirable recommendations. Improvements can be done on neighborhood clustering by finding global maximum or consistent clustering, and on map visualization by adding neighborhood boundaries. More modification can still be done on the definition of a neighborhood profile, depending on stakeholders' preference, such as adding more features aside from venues and utilizing other APIs aside from Foursquare.

Nevertheless, this system is working as intended to recommend similar neighborhoods within a given location based on a smaller input location. We were able to achieve that inconsistently due to the local maximum convergence tendency of k-means clustering, but the ranking of neighborhoods are consistently correct, informative, and hopefully useful for the target market.