The Battle of Neighborhoods

Gustavo C. Bicalho

February 2019

1 Introduction

When you are moving to a new city, you need to choose a neighborhood to live in. But that a hard task when you have no information about any place in your new town. In this project, we will create an prototype of a recommendation system application that can select the best neighborhood for the user, based on his favorite venues. And with information about the neighborhoods that best fit his preferences, he'll be able to make a better choice. It's important to notice that we'll only take into account the users preferences and not the whole quality of life the neighborhood offers

2 Data

For the data in this prototype, we will use Toronto as an example city, using a data base with all boroughs, postal code and name each neighborhood. Also we will use the Foursquare API to obtain the data on what venues are located at each neighborhood. In this data, we will have the latitude, longitude for each venue, as well as the category and name for them. For the user data, we will create a random user, with a random number of preferences(from 1 to 10) and select randomly from the list of categories available in the city(obtained through the Foursquare API) that number of categories.

3 Methodology

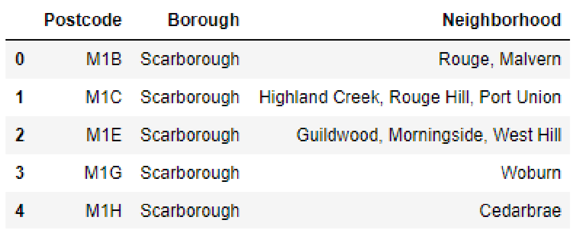
In this project, we will create a content based recommendation system that allows an user to choose the neighborhood that best fit his interests. In this prototype, we'll use Toronto as an example and generate a random user to test the system.

3.1 Preparing and Obtaining the Data

First thing we have to do is obtain the information from the boroughs and neighborhoods in the city. For that we'll use that table available at:

https://en.wikipedia.org/wiki/List_of_postal_codes$_of\_Canada : \_M$

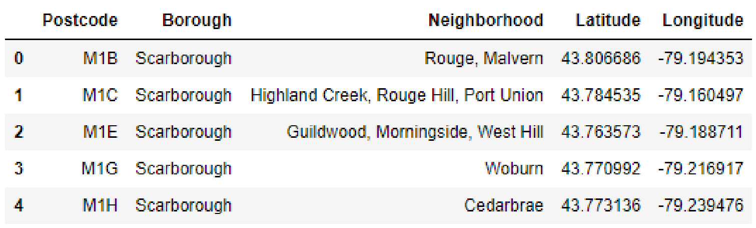
To extract the table, we'll use the Beautiful Soup library for python, allowing us to parse the HTML and extract the information needed, in this case we'll look for the 'th' tag for the headers and 'td' tags for the content. In this part we also need to remove any postal code that has no borough assigned to it, and save the borough name as the neighborhood if no neighborhood is assigned. After that, we have to group the table by the postal code, this will result in a table like shown in Figure 1.



	Postcode	Borough	Neighborhood
0	M1B	Scarborough	Rouge, Malvern
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

Figure 1: Figure showing the table with the postal code, borough and the grouped neighborhoods.

The next step is to obtain the latitude and longitude for each neighborhood in the table. To obatin that information, we can use the geocoder library or download it from http://cocl.us/Geospatial_data. Now we can join both tables, obtaning the table shown in figure 2



	Postcode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Rouge, Malvern	43.806686	-79.194353
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union	43.784535	-79.160497
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

Figure 2: Figure showing the table with the postal code, borough and the grouped neighborhoods and the longitude and latitude for them.

Now, we can use the Foursquare API to get the venues for each neighborhood. In this project, we'll limit the amount of venues per neighborhood to 100 and the range from the center of the neighborhood to 500 meters. With this API we will have all the venues for each neighborhood and be able to group them for each neighborhood. This will result in a table with all neighborhoods as the indexes and the venues and the categories for each of them. Applying the OneHot encode in the categories and applying the mean for the amount of venues with each category, we'll obtain a new table with the neighborhood as the index and the percentage of each category available in that neighborhood.

3.2 Random User

To generate a random user, we first need the list of all categories available in the city. To do that all we have to do is get the header values from that table with neighborhoods and the mean for each category and remove the header that represents the neighborhood.

Now, we select a random number from 1 to 10 that will represent the amount of categories selected by the user. Then, from the list of categories we'll sample that same amount, obtaining the list of categories that our user will have interest in. Now we create a table with the categories as the columns and one row, where the values are 1 if the user has that category in his list or 0 otherwise. This will result in a user profile that will be used in the recommendation system.

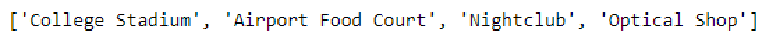
3.3 Recommendation System

Now, to make a recommendation system, all we have to do is compare our user profile to the table with the neighborhoods and the mean value for the amount of venues of each category in it. To do that, we multiply both matrix and apply a sum for each row. This will result in a new matrix with the neighborhoods and a score for each one of them. The higher the score the better the neighborhood matches the user interests. If we order this new table, we'll be able to see which neighborhoods have the highest scores. And now if we merge this table with the scores to the one with the geospatial table presented in Figure 2, we'll be able to print in a map where are the better neighborhoods for our user.

4 Results

In this prototype, we used a random user. The results will be based on one iteration of the algorithm.

In this test, the user had the categories shown in Figure 3.



```
['College Stadium', 'Airport Food Court', 'Nightclub', 'Optical Shop']
```

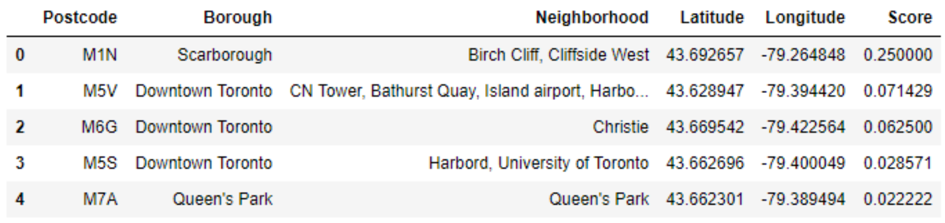
Figure 3: Categories random selected for the user.

With this user, we obtained the table with the scores shown in Figure 4.

And with that, we can generate a map with the best 5 fits for the user, as shown in Figure 5

5 Discussion

From this result, we can see that for our user, the best neighborhood based on his interests would be "Birch Cliff, Cliffside West". And also, from table in Figure 4, we can see that the difference between the first and the other results obtained are large. This means that the first result is by far the best place



	Postcode	Borough	Neighborhood	Latitude	Longitude	Score
0	M1N	Scarborough	Birch Cliff, Cliffside West	43.692657	-79.264848	0.250000
1	M5V	Downtown Toronto	CN Tower, Bathurst Quay, Island airport, Harbo...	43.628947	-79.394420	0.071429
2	M6G	Downtown Toronto	Christie	43.669542	-79.422564	0.062500
3	M5S	Downtown Toronto	Harbord, University of Toronto	43.662696	-79.400049	0.028571
4	M7A	Queen's Park	Queen's Park	43.662301	-79.389494	0.022222

Figure 4: Score table obtained for the random user.

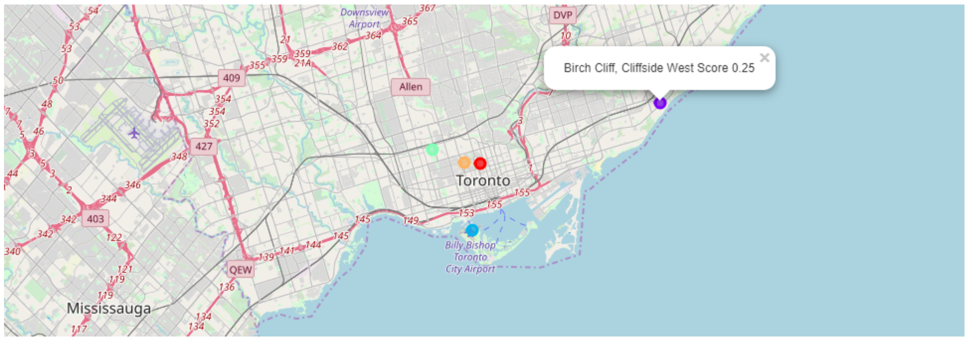

Birch Cliff, Cliffside West Score 0.25
Toronto
Mississauga
Downsview Airport
Billy Bishop Toronto City Airport
Allen
DVP
QEW

Figure 5: Map with the 5 best neighborhoods for our user.

based on his interests. One possibility for this happening is that our list has some unusual categories such as 'Airport Food Court' and 'College Stadium', and not having any or both of these categories will really hurt the score of a neighborhood. Analyzing the data, we can see that "Birch Cliff, Cliffside West" is the one where College Stadium has the highest weight. And because of that this will be the best fit.

6 Conclusion

This is a simple content based recommendation system that still need improvements. If a region has few venues assigned to it, the weight of single venues will be huge resulting and a probably best fit. To fix that it's possible to consider only neighborhoods with a certain number of venues found. Also, we are not considering other possible interests such as how close the place is to the users work, the GDP of the neighborhood and other possible features that could allow the user to make the best choice.

Also, this is only a prototype, the idea is that the application can be used for different cities and allow the user to enter the categories they prefer.