IBM Coursera Data Science Professional Certificate
Capstone Project: The Battle of the Neighborhoods
March 5, 2020
Marcelo Guimarães

Comparing Neighbourhoods in New York City and Toronto

Introduction

When you have to move from your home, it is always difficult to find the right neighbourhood to live. Throughout my life I have moved among different neighbourhoods in the same city, from one city to another inside the same state, from one state to another, and even from one country to another, including countries in different continents. Every time I was moving from one place to another, the same question arises: where in this new city will I find the right place to live?
This problem could be minimized if we were able to compare the neighbourhoods in different cities and make a list of the best candidates, or at least the neighbourhoods that are similar to the one we like.

What if we could create a recommendation system for neighbourhoods? We will try to create such system by gathering information about the neighbourhoods using the Foursquare API. The recommendation system will be based on our preferred venues and their ratings, its output will be a list of possible candidates. It is not a complete solution, but it is a start.

In this project we will consider a client that lives in Toronto, specifically in the neighbourhood called Little Portugal. The client will move to New York City and would like to know which neighbourhoods would be similar to the current one.

All the details of the process are well documented and described in the Jupyter Notebook associated with this final report.

Data

We will collect data from different sources in order to understand the distribution of venues in New York City and Toronto, and start to search for good areas to live.

For New York City, we will collect information about each neighbourhood and borough from the website: https://cocl.us/new_york_dataset
It returns a JSON file, which will be open using Pandas and read into a Dataframe. The first 5 rows for this dataframe is displayed in Table 1.

Table 1: New York City Neighbourhood Information.

Borough	Neighbourhood	Latitude	Longitude
Bronx	Wakefield	40.89	-73.85
Bronx	Co-op City	40.87	-73.83
Bronx	Eastchester	40.89	-73.83
Bronx	Fieldston	40.9	-73.91
Bronx	Riverdale	40.89	-73.91

The localization of each neighbourhood can be seen in the map below.

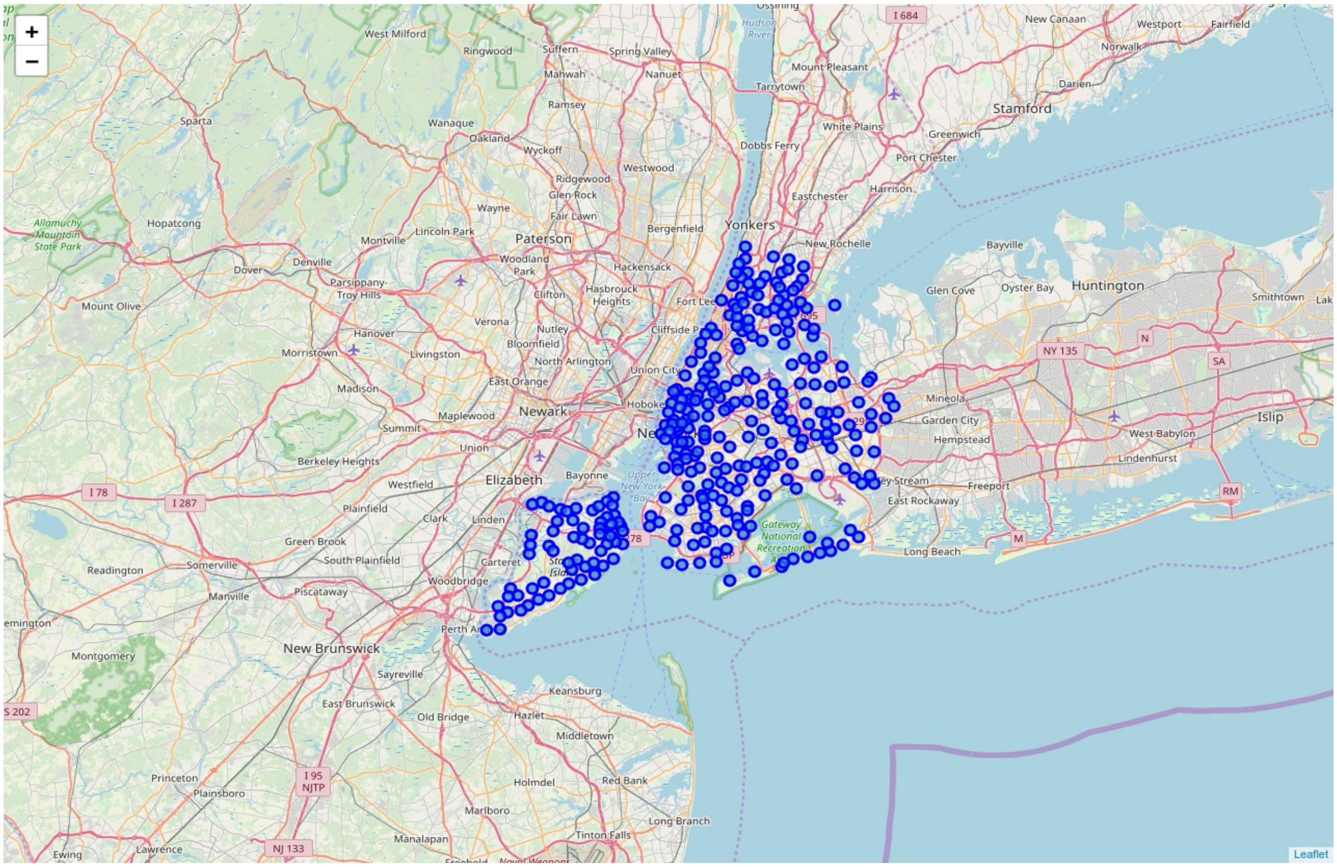


Figure 1: Map of New York City and its neighbourhoods.

For Toronto we will use the Wikipedia page containing information about the postcodes and neighbourhoods of the city. We will use the Wikipedia library for Python in order to extract the table with the information important to us. The latitude and longitude for each neighbourhood will be extracted from the CSV file: Geospatial_Coordinates.csv. We can see the first 5 rows of the final dataframe below.

Table 2: Toronto Neighbourhood Information

Borough	Neighbourhood	Latitude	Longitude
North York	Parkwoods	43.75	-79.33
North York	Victoria Village	43.73	-79.32
Downtown Toronto	Harbourfront	43.65	-79.36
North York	Lawrence Heights, Lawrence Manor	43.72	-79.46
Downtown Toronto	Queen's Park	43.66	-79.39

The map of Toronto with its neighbourhoods follows below.

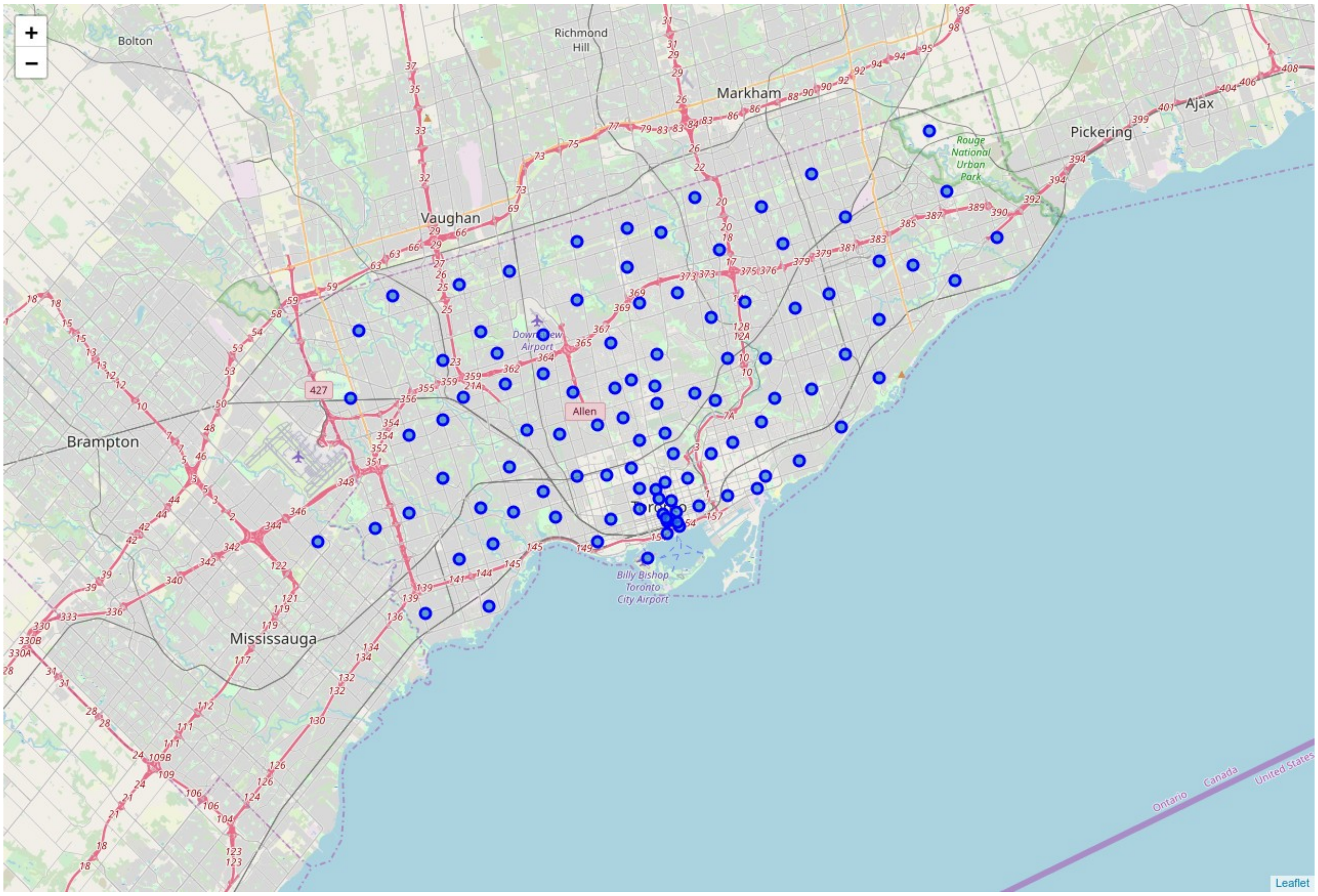


Figure 2: Map of Toronto and its neighbourhoods.

We will use the Foursquare API to retrieve relevant data for New York City and Toronto and organize it into pandas Dataframes.

We limited our search in 100 venues/neighbourhood and a search radius of 500 meters, centred in the latitude and longitude of the neighbourhood.

For New York City we have 306 neighbourhoods, distributed in 5 boroughs and our query using the Foursquare API returned 10278 venues, with 429 unique categories.
The first 5 rows of the dataframe with the venues data are shown in Table 3.

Table 3: Example of venues returned by the Foursquare API. New York Venues Dataframe.

Neighbourhood	Neigh. Latitude	Neigh. Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Wakefield	40.89	-73.85	Lollipops Gelato	40.89	-73.85	Dessert Shop
Wakefield	40.89	-73.85	Rite Aid	40.9	-73.84	Pharmacy
Wakefield	40.89	-73.85	Carvel Ice Cream	40.89	-73.85	Ice Cream Shop
Wakefield	40.89	-73.85	Walgreens	40.9	-73.84	Pharmacy
Wakefield	40.89	-73.85	Dunkin'	40.89	-73.85	Donut Shop

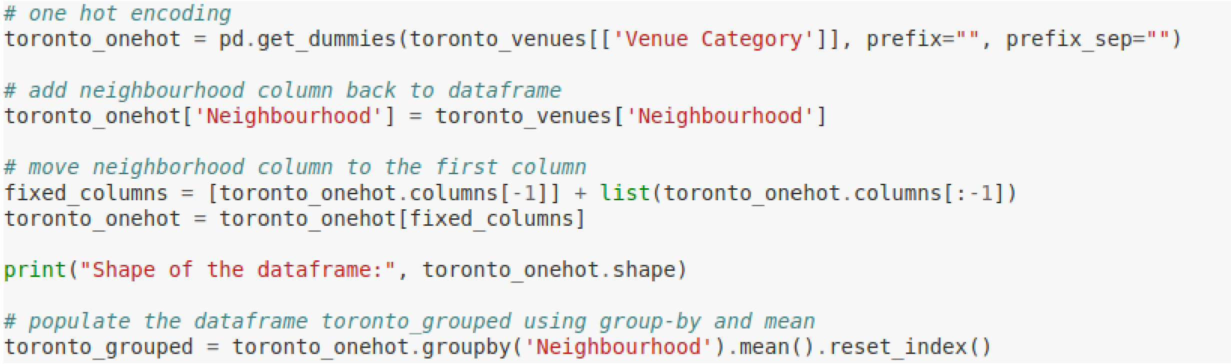
For Toronto we have 103 neighbourhoods, distributed in 10 boroughs and a total of 2228 venues, as shown in the table below. These 2228 venues are distributed in 267 unique categories. An example of the Dataframe is shown in Table 4.

Table 4: Example of venues returned by the Foursquare API. Toronto Venues Dataframe.

Neighbourhood	Neigh. Latitude	Neigh. Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Parkwoods	43.75	-79.33	Brookbanks Park	43.75	-79.33	Park
Parkwoods	43.75	-79.33	Variety Store	43.75	-79.33	Food & Drink Shop
Victoria Village	43.73	-79.32	Victoria Village Arena	43.72	-79.32	Hockey Arena
Victoria Village	43.73	-79.32	Tim Hortons	43.73	-79.31	Coffee Shop
Victoria Village	43.73	-79.32	Portugril	43.73	-79.31	Portuguese Restaurant

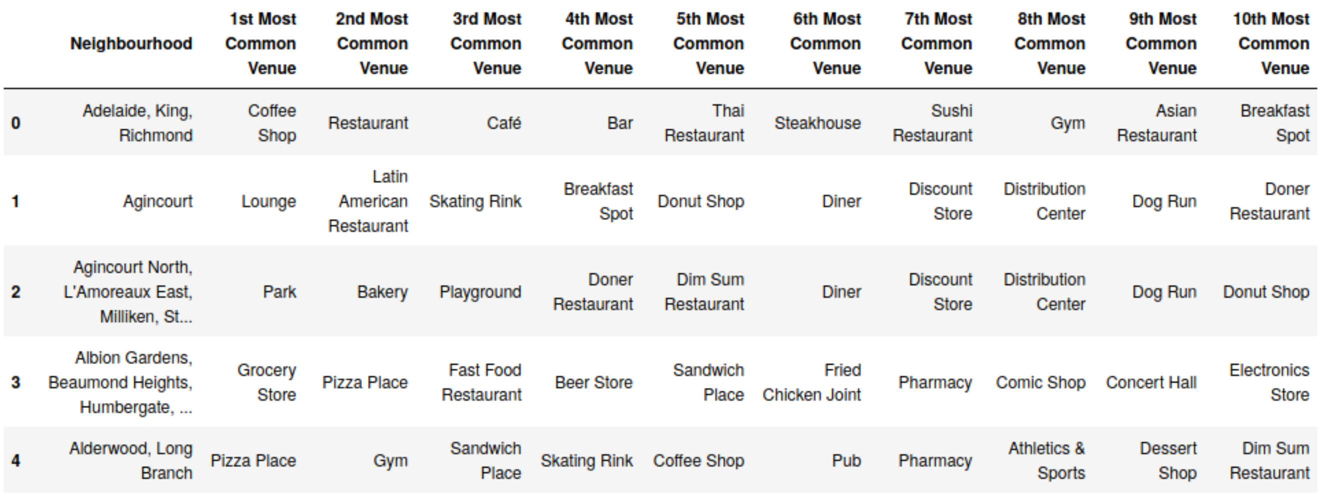
Methodology

Like said in the Introduction, our client lives currently in the neighbourhood Little Portugal in Toronto but wants to move to a similar neighbourhood in New York City. We will start by analyzing our client's current neighbourhood, specifically we will list the most common venues in Little Portugal. We will create a new dataframe, listing all the unique categories for each neighbourhood. Our intention is to obtain a list of most frequent venues per neighbourhood. We will then use this information to characterize the neighbourhoods. This task is accomplished using the one-hot encoding, details can be seen in the Jupyter Notebook made available together with this report.



```python
# one hot encoding
toronto_onehot = pd.get_dummies(toronto_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighbourhood column back to dataframe
toronto_onehot['Neighbourhood'] = toronto_venues['Neighbourhood']

# move neighborhood column to the first column
fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]

print("Shape of the dataframe:", toronto_onehot.shape)

# populate the dataframe toronto_grouped using group-by and mean
toronto_grouped = toronto_onehot.groupby('Neighbourhood').mean().reset_index()
```

After the one-hot step we proceed to classify the most common venues and an example of the Dataframe produced in this last step is shown below, where we can seethe 10 most common venues for 5 neighbourhoods in Toronto.



	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Adelaide, King, Richmond	Coffee Shop	Restaurant	Café	Bar	Thai Restaurant	Steakhouse	Sushi Restaurant	Gym	Asian Restaurant	Breakfast Spot
1	Agincourt	Lounge	Latin American Restaurant	Skating Rink	Breakfast Spot	Donut Shop	Diner	Discount Store	Distribution Center	Dog Run	Doner Restaurant
2	Agincourt North, L'Amoreaux East, Milliken, St...	Park	Bakery	Playground	Doner Restaurant	Dim Sum Restaurant	Diner	Discount Store	Distribution Center	Dog Run	Donut Shop
3	Albion Gardens, Beaumond Heights, Humbergate, ...	Grocery Store	Pizza Place	Fast Food Restaurant	Beer Store	Sandwich Place	Fried Chicken Joint	Pharmacy	Comic Shop	Concert Hall	Electronics Store
4	Alderwood, Long Branch	Pizza Place	Gym	Sandwich Place	Skating Rink	Coffee Shop	Pub	Pharmacy	Athletics & Sports	Dessert Shop	Dim Sum Restaurant

We will increase the number of venues to be analyzed in our specific case, Little Portugal, to 15. Our client provided ratings for these 15 most common venues in Little Portugal and we used this information to create a rating vector. We can see these details below.

Table 5: The 15 most common venues for Little Portugal, Toronto, and the rating provided by the client.

Ranking Most Common Venue	Venue Category	Client Rating
1	Bar	9
2	Coffee Shop	9.5
3	Asian Restaurant	9.5
4	Restaurant	9
5	Café	10
6	Pizza Place	7
7	Bakery	10
8	Men's Store	4.5
9	Wine Bar	5
10	Vietnamese Restaurant	8.5
11	Italian Restaurant	7.5
12	Japanese Restaurant	9.5
13	Bistro	7
14	Brewery	6.5
15	Gift Shop	6.5

The next step is to filter the venues for the New York neighbourhoods and select only the same type of venue category present in our rating vector. This will create a new Dataframe with all the neighbourhoods in New York with these 15 types of venue. We then multiply these Dataframe by the rating vector, using a dot product between matrices. Each neighbourhood will have a final score associated with it.

Results

Using our recommendation system we have the final scores for the neighbourhoods in New York City. Table 5 presents the top 10 highest scores. We can submit this result to our client, together with a map of the localization of each neighbourd.

Table 6: Top 10 Highest Scores for New York City Neighbourhoods.

Neighbourhood	Score
Belmont	304.5
Carroll Gardens	290
Yorkville	288.5
Greenpoint	283.5
Carnegie Hill	271
Upper West Side	265
Murray Hill	255
South Side	242
Prospect Heights	234
East Village	226.5

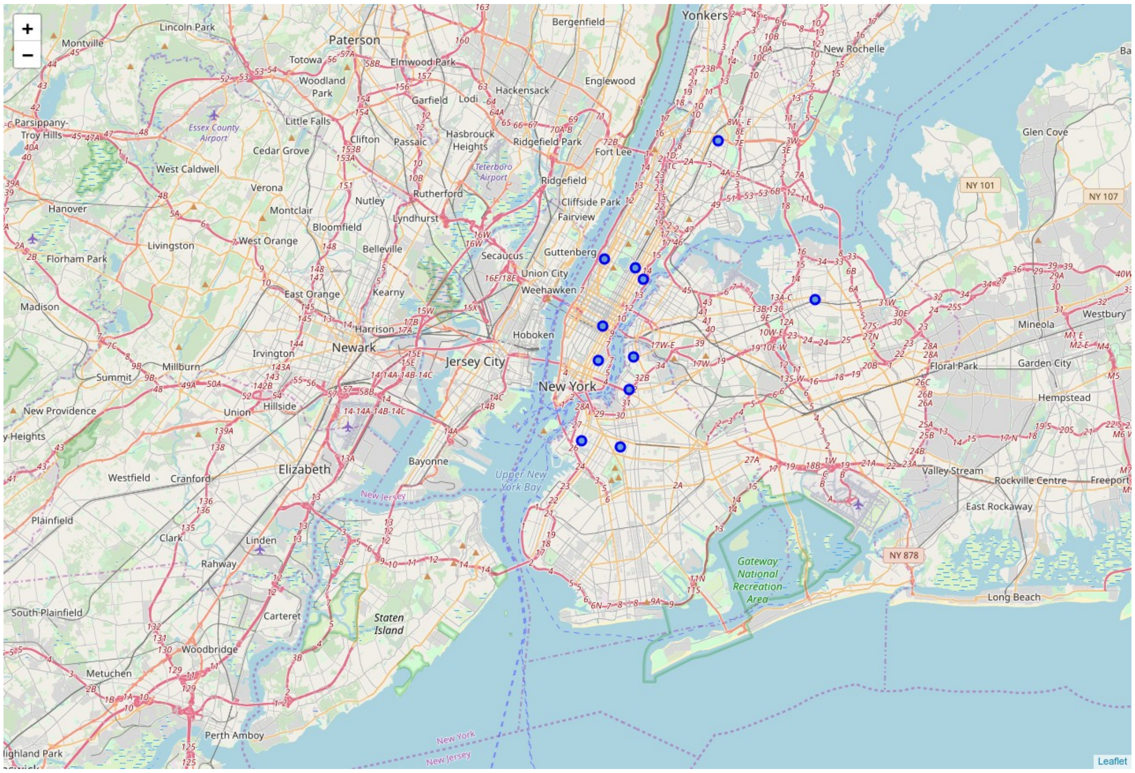


Figure 3: Localization of the Top 10 neighbourhoods with highest scores.

Discussion

The methodology applied here is very simple, compared to what is really necessary to select a new neighbourhood in a different city.
However, it is a start. We would need more information, like rental or saling prices, public transportation, schools, etc.
Unfortunately we don't have that information with Foursquare.
This project can be improved with time, allowing for more constrains to be used in order to select similar neighbourhoods to live.

Conclusion

In conclusion, the Foursquare API is a powerful machine to help us solve problems regarding selection of venues in different locations.

A simple recommendation system worked fine and we are able to provide our client with a list of similar neighbourhoods in a different city, together with a map showing their localization.

There is room for improvement. The combination of such a recommendation system with an API that could retrieve real state data about sales and rental prices would be very interesting.