Ease Relocation to New Places With Machine Learning

Prepared for: Coursera Applied Data Science Capstone Project

Prepared by: Stefano Cantù

3 July 2020

Introduction

Every year thousands of people relocate to new cities and **finding the right place to live is not an easy task.**
Each city area/zip code has its own characteristic and finding the right one to suit a person needs and preferences can be a challenge.
This project seeks to find a solution to a this common problem:
Define a recommending system that will find the best suitable area/zip code in a city based on user input.
The recommending system can cater to both individuals and/or relocation agencies.
For this project I will use Toronto as relocation city.

Data

In order to accomplish the goal multiple data sets are required:

- Toronto's list of area/zip codes - it can be scraped from this Wikipedia page
- Latitudes and Longitudes of Toronto area/zip code which can be extracted using geocoder.
- The number of recommended venues in a specific venue category for each area/zip code - this can be obtained using the Forsquare API explore endpoint.
- A user input that includes:
    - Importance/rating for each venue category.
    - Workplace address.
    - Importance/rating for distance from Workplace.

Methodology

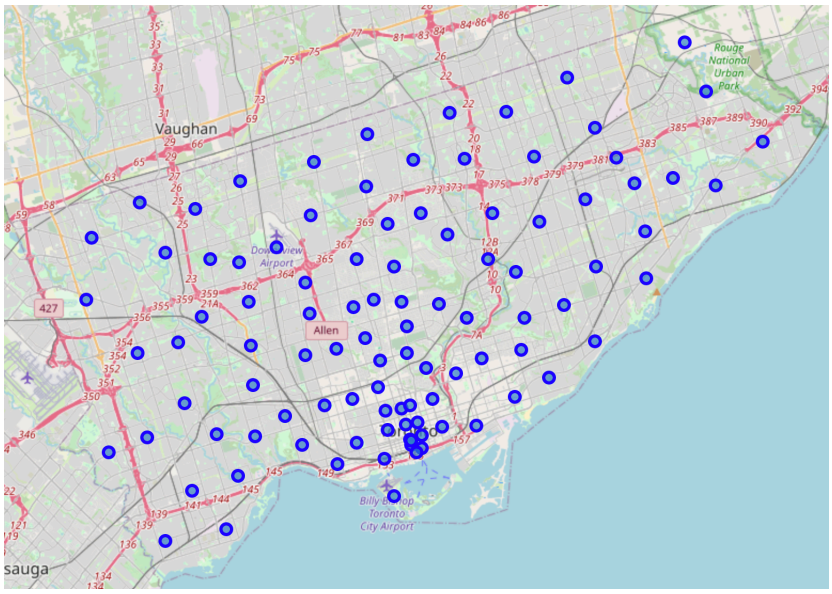
First step is to scrape the Wikipedia webpage to obtain the list of zip codes, borough and neighbourhood in Toronto. This can be done in multiple ways, I've leveraged Pandas libraries.

	Postal Code	Borough	Neighborhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
5	M6A	North York	Lawrence Manor, Lawrence Heights
6	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government
7	M8A	Not assigned	Not assigned
8	M9A	Etobicoke	Islington Avenue, Humber Valley Village
9	M1B	Scarborough	Malvern, Rouge

Next, I needed to assign geo coordinates to each zip code.
Also in this case, there are multiple ways to achieve this and for this project I've leveraged the pgeocode library.

Below the updated data frame and its map visualisation using Folium library.
The data frame has been cleaned by removing all the NaN values and the “Neighbourhood” column as it’s not relevant for this project.

	Postal Code	Borough	Latitude	Longitude
0	M3A	North York	43.7545	-79.3300
1	M4A	North York	43.7276	-79.3148
2	M5A	Downtown Toronto	43.6555	-79.3626
3	M6A	North York	43.7223	-79.4504
4	M7A	Downtown Toronto	43.6641	-79.3889
5	M9A	Etobicoke	43.6662	-79.5282



Next, I can use the Foursquare explore API the number of recommended venues of each category in a zip code.
To reduce the complexity of the data set I have applied the following constraints:

- Limit the venue search to a radius of 1000m, this was chosen because 1000m is a reasonable walking distance.
- Categorise venues using Foursquare high-level venue categories:
  - Arts & Entertainment (4d4b7104d754a06370d81259)
  - College & University (4d4b7105d754a06372d81259)
  - Event (4d4b7105d754a06373d81259)
  - Food (4d4b7105d754a06374d81259)
  - Nightlife Spot (4d4b7105d754a06376d81259)

- Outdoors & Recreation (4d4b7105d754a06377d81259)
- Professional & Other Places (4d4b7105d754a06375d81259)
- Residence (4e67e38e036454776db1fb3a)
- Shop & Service (4d4b7105d754a06378d81259)
- Travel & Transport (4d4b7105d754a06379d81259)

Below the API call made:

GET https://api.foursquare.com/v2/venues/explore?client_id={{client_id}}&client_secret={{client_secret}}&v={{v}}&ll=**{{lat}},{{lng}}&**radius=**1000**&categoryId={{cat_id}}

After json response cleaning and data transformation the result is the desired data frame which includes the count of Foursquare recommended venues by venue category for each zip code.

	Postal Code	Borough	Latitude	Longitude	Arts & Entertainment	College & University	Event	Food	Nightlife Spot	Outdoors & Recreation	Professional & Other Places	Residence	Shop & Service	Travel & Transport
0	M3A	North York	43.7545	-79.3300	1	2	0	5	1	3	10	5	7	8
1	M4A	North York	43.7276	-79.3148	3	2	0	5	1	0	12	2	8	1
2	M5A	Downtown Toronto	43.6555	-79.3626	27	22	0	95	28	38	0	16	58	27
3	M6A	North York	43.7223	-79.4504	10	5	0	51	4	8	21	0	115	13

The generated data frame gives me a good idea of the distribution of venues in each zip code and it can be used as the starting point to build the recommendation engine.

To build the recommendation engine, we first start by collecting the input from an hypothetical person that wants to relocate to Toronto.
The user input is collected by asking two questions:

1. What is your new workplace address
2. On a scale from 1 to 10 what kind of aspects & venue categories are important to you when thinking of an area/zip code to relocate?
    - Arts & Entertainment
    - College & University
    - Event Venues
    - Food
    - Nightlife Spot
    - Outdoors & Recreation
    - Professional & Other Places
    - Residence
    - Shop & Service
    - Travel & Transport
    - Distance from Workplace

The resulting user input is:

Workplace address:
”120 Bremner Blvd #1600, Toronto”
Area/Zip codes preferences:

Category	Rating
Arts & Entertainment	7
College & University	1
Event	7
Food	8
Nightlife Spot	6
Outdoors & Recreation	6
Professional & Other Places	3
Residence	5
Shop & Service	5
Travel & Transport	4
Distance from Workplace	9

Next, I can process the user input data.
First I extract the longitude and latitude of the workplace address using geocoder library.
Then I calculate the distance between the workplace and the zip codes using geopy library and append the results to the main data frame.

	Postal Code	Borough	Latitude	Longitude	Arts & Entertainment	College & University	Event	Food	Nightlife Spot	Outdoors & Recreation	Professional & Other Places	Residence	Shop & Service	Travel & Transport	Distance from Workplace
0	M3A	North York	43.7545	-79.3300	1	2	0	5	1	3	10	5	7	8	13103.463106
1	M4A	North York	43.7276	-79.3148	3	2	0	5	1	0	12	2	8	1	10888.055182
2	M5A	Downtown Toronto	43.6555	-79.3626	27	22	0	95	28	38	0	16	58	27	2150.233892
3	M6A	North York	43.7223	-79.4504	10	5	0	51	4	8	21	0	115	13	10355.151986
4	M7A	Downtown Toronto	43.6641	-79.3889	38	112	5	239	78	67	84	41	119	79	2393.823444

To ensure the different scales for each feature don’t skew the final result, I proceed with data normalisation using min-max scaler.
I also calculate the reciprocal value for the feature “Distance from Workplace” to ensure that the lower distances have a positive impact on the model.

Now, the data frame is finally complete ready to be used for the recommendation engine.

To build the recommendation engine, firstly I normalise the ratings provided by the hypothetical person who wants to relocate to Toronto.

Category	Rating
Arts & Entertainment	0.750
College & University	0.000
Event	0.750
Food	0.875
Nightlife Spot	0.625
Outdoors & Recreation	0.625
Professional & Other Places	0.250
Residence	0.500
Shop & Service	0.500
Travel & Transport	0.375
Distance from Workplace	1.000

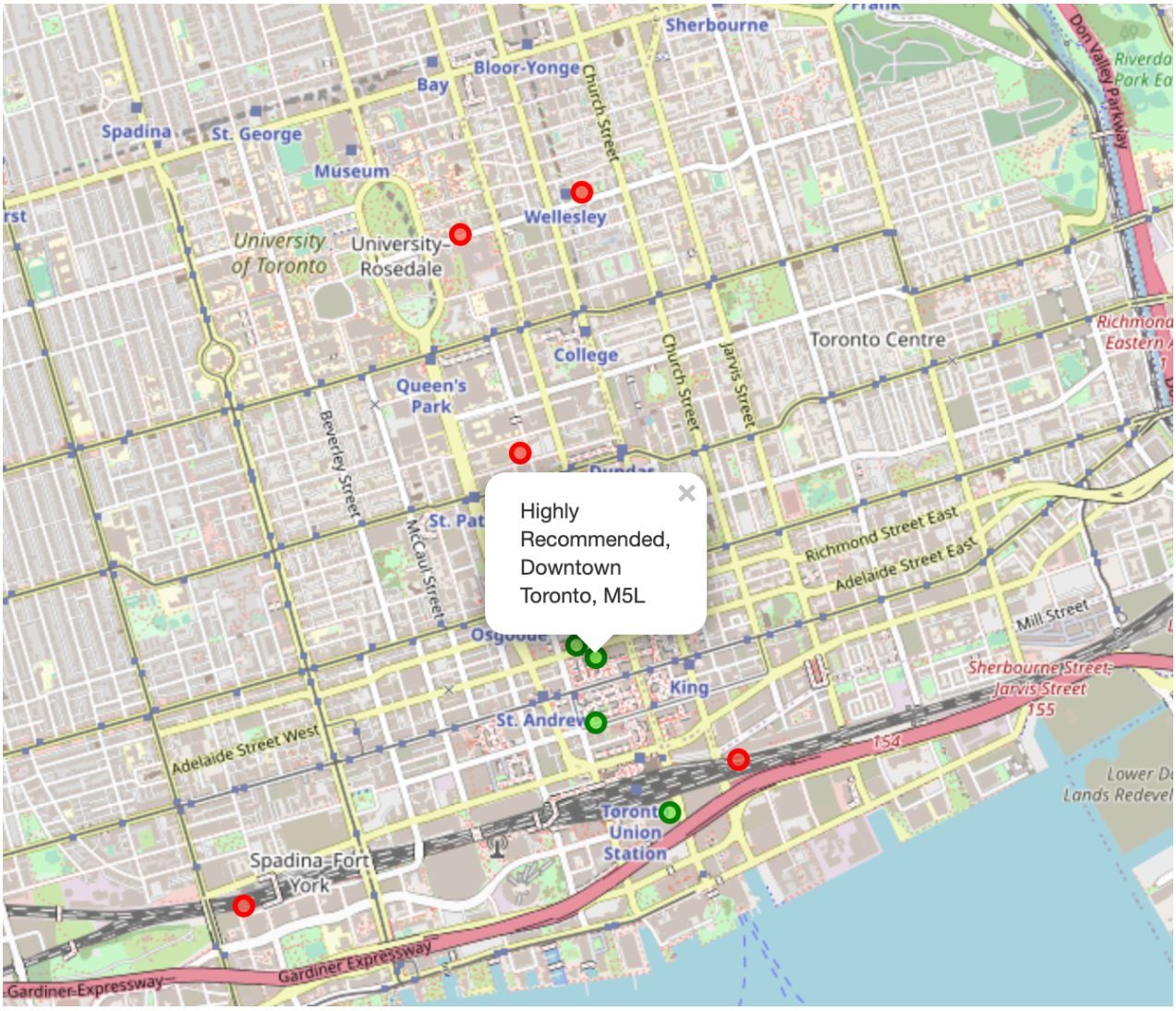
Then, I can calculate the **Recommendation Score** for each zip code by multiplying the main data frame features by the user ratings and summing each row.

	Postal Code	Arts & Entertainment	College & University	Event	Food	Nightlife Spot	Outdoors & Recreation	Professional & Other Places	Residence	Shop & Service	Travel & Transport	Distance from Workplace	Recommendation Score
0	M3A	0.010870	0.0	0.00	0.018305	0.004281	0.017523	0.023148	0.052083	0.025735	0.030612	0.506587	0.689146
1	M4A	0.032609	0.0	0.00	0.018305	0.004281	0.000000	0.027778	0.020833	0.029412	0.003827	0.418581	0.555626
2	M5A	0.293478	0.0	0.00	0.347803	0.119863	0.221963	0.000000	0.166667	0.213235	0.103316	0.071475	1.537801
3	M6A	0.108696	0.0	0.00	0.186715	0.017123	0.046729	0.048611	0.000000	0.422794	0.049745	0.397412	1.277825

I can now use the **Recommendation Score** to build a **Recommendation Table** and identify the Top 10 recommended zip codes for the hypothetical user relocating to Toronto.
I'm also setting a threshold of 4 that split the zip codes between "Recommended" (Recommendation Score below 4) and "Highly Recommended" (Recommendation Score above 4).

Postal Code	Borough	Latitude	Longitude	Recommendation Score	Recommendation
M5H	Downtown Toronto	43.6496	-79.3833	4.875008	Highly Recommended
M5X	Downtown Toronto	43.6492	-79.3823	4.854897	Highly Recommended
M5L	Downtown Toronto	43.6492	-79.3823	4.854897	Highly Recommended
M5K	Downtown Toronto	43.6469	-79.3823	4.836827	Highly Recommended
M5W	Downtown Toronto	43.6437	-79.3787	4.430335	Highly Recommended
M5E	Downtown Toronto	43.6456	-79.3754	3.784004	Recommended
M7A	Downtown Toronto	43.6641	-79.3889	3.705778	Recommended
M4Y	Downtown Toronto	43.6656	-79.3830	3.478949	Recommended
M5G	Downtown Toronto	43.6564	-79.3860	3.202910	Recommended
M5V	Downtown Toronto	43.6404	-79.3995	3.197743	Recommended

Then I can visualise the output of the recommendation engine on a map using Folium, highlighting with different color coding the different degree of recommendation.



Results

Based on the preferences and workplace location of the hypothetical person willing to move to Toronto the Downtown Toronto area is the most suitable to relocate.
Different zip codes in this area have different degree of preference for the user.

Discussion

Foursquare unfortunately doesn't offer enough data to provide a universal recommendation engine but just enough to build a very basic one.
The highest number of venues are in the Food and Shop & Service categories and it is not clear how “recommended venues” are identified.
Assuming this is defined based on the activity on Foursquare owned social media platform (Swarm), it is likely that the list of recommended venue might not be very reliable.

Conclusion and Possible Improvements

While Foursquare dataset is limited the recommendation engine still delivers on the goal of recommending an array of areas for relocation.
Improvements in the recommendation engine could be achieved by:

- Use a wider array of data points to generate the user profile - e.g. lower level venue category preference, Housing accommodation preferences, etc...
- Add a weighting system based on ratings, tips and likes for the scores of each category.
- Combine other sources - e.g. city data on number of residents, Events data, traffic data to calculate commuting time.
- Embed ratings from past living experiences - e.g. Checking similarities with area/zip codes someone have lived previously within the same city or other cities.
- Embed ratings from other individuals and develop a collaborative filtering recommendation engine.