# THE TRAVELING ADVISOR

Finding similar locations in a new city that resemble your favorite places the most for visitors

JANUARY 19, 2019

## Introduction

When people visit other cities for some reasons (traveling, attending conference, visiting a friends, etc.), many of them are eager to explore the city they are not familiar with. It would be helpful to suggest some places for them to go based on their only interests. For example, if a person is enthusiastic about museum exhibitions, it would be good to recommend her to a neighborhood that many local museums are located; if someone loves French cuisine, it is preferable to suggest a place with many such restaurants for him to choose. A person who have being living in a city for years may be able to offer useful advices regarding above scenarios, however, identifying such a person is not easy and it could be time-consuming, especially for strangers of this place. This report is aimed to solve such a problem by leveraging the ability of machine learning algorithms, punctuated by the help of various python libraries and Foursquare API, to identify places of interests for people based on their needs. To be more specific, in the jupyter notebook of this report, a person could provide a place that she favors (i.e., a place she is already familiar with, such as an art block in her city, a beach park nearby, a place with some of her most favorite restaurants, etc.), and the program will find the cluster of places that the proposed place belongs to in Toronto (the city to be visited in this report) and recommend five most similar places in the city to explore. In short, the problem to be solved in this report is:

Find the closest cluster of the place provided by the user in Toronto and recommend five most similar places in the city based on the proposed location.

## Data

To solve the problem described in the introduction, several sources of data will be considered.

First, it is important to find a reference point of the user-proposed place in order to calculate relevant venues around the point and then generate a vector for cluster assignment and comparison. Geocoder, a python library to convert a manually input address into coordinates, is therefore used in this report. Figure 1 shows a screenshot of how a Geocoder module could be used to find the coordinates given a sample address.

```
address='104 Montgomery St, San Francisco, CA 94129'

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
37.80136265 -122.458720604517
```

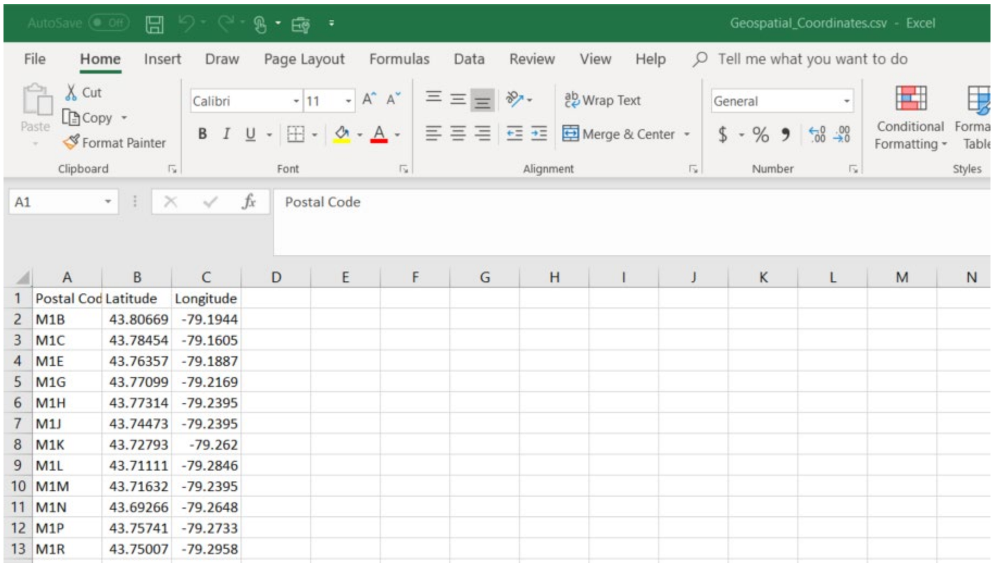
```
map_query = folium.Map(location=[latitude, longitude], zoom_start=16)

map_query
```


Figure 1. Code snippets of the application of Geocoder to find coordinates.

Second, since this report uses Toronto as the target city to perform the task, data regarding its neighborhoods is necessary in order to run clustering algorithm. Luckily, such kind of data is readily available from previous modules of the same course. To be specific, boroughs and neighborhoods names as well as postal codes could be found from the Wikipedia page and coordinates of each neighborhood could be provided by either the Geocoder library or the

"Geospatial Coordinates.csv" file prepared by the course instructor. Figure 2 shows a screenshot of part of the Geospatial Coordinates file which lists coordinates and their corresponding postal codes.



| Postal Cod | Latitude | Longitude |
|---|---|---|
| M1B | 43.80669 | -79.1944 |
| M1C | 43.78454 | -79.1605 |
| M1E | 43.76357 | -79.1887 |
| M1G | 43.77099 | -79.2169 |
| M1H | 43.77314 | -79.2395 |
| M1J | 43.74473 | -79.2395 |
| M1K | 43.72793 | -79.262 |
| M1L | 43.71111 | -79.2846 |
| M1M | 43.71632 | -79.2395 |
| M1N | 43.69266 | -79.2648 |
| M1P | 43.75741 | -79.2733 |
| M1R | 43.75007 | -79.2958 |

Figure 2. The Geospatial Coordinates file lists postal codes of Toronto neighborhoods and their coordinates.

Third, in order to find the venues and their respective categories of the favorite place proposed by the user and neighborhoods in the target city (Toronto), Foursquare API is applied. Foursquare is a third-party API that could be used to query specified venues around a given location. Figure 3 shows a search query using Foursquare to identify presumably French restaurants/bakeries within 1000 meters of a location in New York City by setting the query string to "French". The result is in JSON format which could be handled to extract useful information later.

```python
search_query = 'French'
radius = 1000
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_
results = requests.get(url).json()
results
```


```
{'meta': {'code': 200, 'requestId': '5c43b711dd57975fd5219e8c'},
 'response': {'venues': [{'id': '4e4e4a47bd4101d0d7a6f34b',
    'name': 'Le Croissant Shop French Bakery',
    'location': {'address': '10 Ave. of the Americas',
     'lat': 40.71799133506263,
     'lng': -74.00949587210722,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.71799133506263,
       'lng': -74.00949587210722}],
     'distance': 597,
     'postalCode': '10013',
     'cc': 'US',
     'city': 'New York',
     'state': 'NY',
     'country': 'United States',
     'formattedAddress': ['10 Ave. of the Americas',
      'New York, NY 10013',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d16a941735',
      'name': 'Bakery',
      'pluralName': 'Bakeries',
      'shortName': 'Bakery',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/bakery_',
```


Figure 3. A sample query using Foursquare and its returned results in JSON format.

## Methodology

The solution of the proposed problem in this report could be roughly separated into two consecutive steps: 1. Extract Toronto neighborhoods information and perform clustering algorithms to obtain group assignment; 2. Obtain surrounding venue information of the place proposed by the user and compare it to Toronto neighborhoods clusters and individual neighborhood.

During step 1, neighborhoods' names and postal codes were gleaned from Wikipedia, which was facilitated by Beautifulsoup, a python library that makes obtaining information from webpages an easy task. The postal codes were then provided to Geocoder or compared with the Geospatial Coordinates file (whichever is more convenient to get the results) in order to extract coordinates of each neighborhoods (Figure 4). Figure 5 shows the visualization of those neighborhoods locations on a map in which Toronto is located at the center.

| | 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 4. Example of Toronto neighborhoods and their corresponding information.

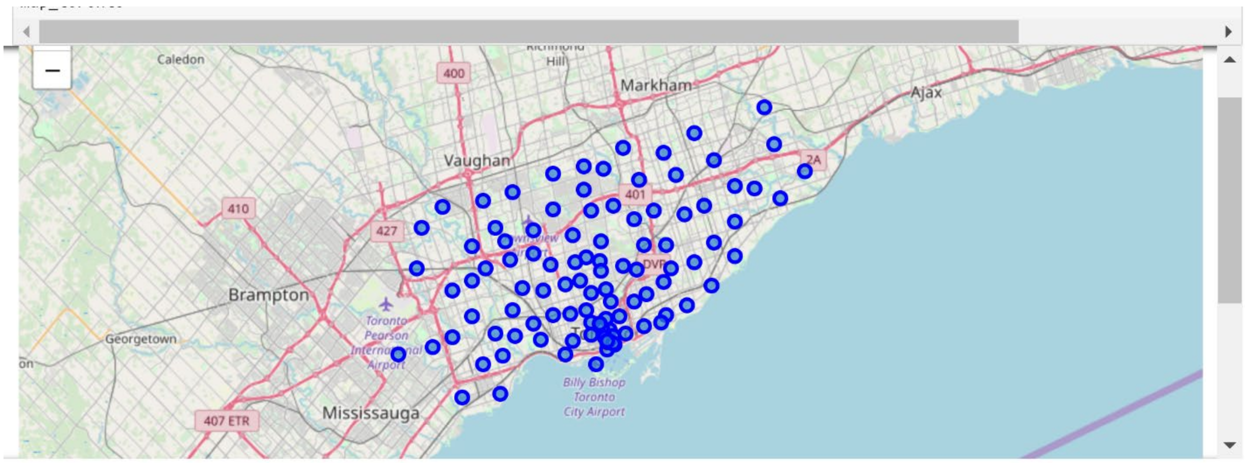



Figure 5. Visualization of Toronto neighborhoods based on their coordinates.

After the coordinates of each neighborhood were identified, Foursquare API was then used to extract venue information surrounds each neighborhood. I set a limit of 200 and the radius of 800 so that at most 200 venues within 800 meters of the location were yielded for each location. I tried several combinations of the two parameters and thought this could be a good representation of a particular neighborhood (just think of finding 200 places to represent a circular area whose radius is only 800 meters). For all the 103 neighborhoods in Toronto area, 3969 venues with their categories were then identified, and in total of 332 unique categories were summarized. Each venue was then one-hot coded against each unique category and the grouped mean of every neighborhood was calculated to represent itself and to feed into clustering algorithms. Figure 6 shows the grouped results.

| | Neighborhood | Yoga Studio | Accessories Store | Adult Boutique | Afghan Restaurant | African Restaurant | Airport | Airport Food Court | Airport Gate | Airport Lounge | ... | Vegetarian / Vegan Restaurant | Video Game Store | Video Store | Vietnamese Restaurant | Warehou St |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Adelaide, King, Richmond | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.01 | 0.0 | 0.0 | 0.0 | |
| 1 | Agincourt | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.00 | 0.0 | 0.0 | 0.0 | |
| 2 | Agincourt North, L'Amoreaux East, Milliken, St... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.00 | 0.0 | 0.0 | 0.0 | |
| 3 | Albion Gardens, Beaumond Heights, Humbergate, ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.00 | 0.0 | 0.0 | 0.0 | |
| 4 | Alderwood, Long Branch | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.00 | 0.0 | 0.0 | 0.0 | |

Figure 6. Grouped mean of venues within each neighborhood.

Based on the grouped mean of venues within each neighborhood (Figure 6), certain descriptive analyses of neighborhood could be performed. For example, Figure 7 shows the result of some most common venues within each neighborhood. From the figure we could see that each neighborhood is different in some ways from others in terms of venue category frequencies. And clustering algorithm could be used to differentiate them into different groups.

| | Neighborhood | 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 | Café | Coffee Shop | Hotel | American Restaurant | Thai Restaurant | Theater | Bar | Concert Hall | Steakhouse | Gastropub |
| 1 | Agincourt | Chinese Restaurant | Shopping Mall | Shanghai Restaurant | Motorcycle Shop | Supermarket | Sandwich Place | Discount Store | American Restaurant | Lounge | Malay Restaurant |
| 2 | Agincourt North, L'Amoreaux East, Milliken, St... | Fast Food Restaurant | Pizza Place | Park | Chinese Restaurant | BBQ Joint | Fried Chicken Joint | Coffee Shop | Pharmacy | Caribbean Restaurant | Noodle House |
| 3 | Albion Gardens, Beaumond Heights, Humbergate, ... | Pizza Place | Grocery Store | Pharmacy | Sandwich Place | Liquor Store | Beer Store | Fried Chicken Joint | Coffee Shop | Fast Food Restaurant | Hardware Store |
| 4 | Alderwood, Long Branch | Pizza Place | Coffee Shop | Gas Station | Pub | Pharmacy | Park | Convenience Store | Sandwich Place | Dance Studio | Skating Rink |

Figure 7. Common venues within each neighborhood.

K-means clustering algorithm was applied to cluster the 103 neighborhoods of Toronto into 5 different categories. Figure 8 shows the clustering results of Toronto neighborhoods, the column named “Cluster Labels” in the figure indicates group assignment. Figure 9 shows the visualization of those clusters on the map.

| | PostCode | Borough | Neighborhood | Latitude | Longitude | Cluster Labels | 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 I Com Ve |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | M1B | Scarborough | Rouge, Malvern | 43.806686 | -79.194353 | 1 | Fast Food Restaurant | Coffee Shop | Chinese Restaurant | Spa | Martial Arts Dojo | Paper / Office Supplies Store | Hobby Shop | Afi Restau |
| **1** | M1C | Scarborough | Highland Creek, Rouge Hill, Port Union | 43.784535 | -79.160497 | 4 | Breakfast Spot | Italian Restaurant | Burger Joint | Bar | Women's Store | Electronics Store | Dive Bar | Dog |
| **2** | M1E | Scarborough | Guildwood, Morningside, West Hill | 43.763573 | -79.188711 | 1 | Pizza Place | Fast Food Restaurant | Coffee Shop | Plaza | Fried Chicken Joint | Beer Store | Pharmacy | G Restau |
| **3** | M1G | Scarborough | Woburn | 43.770992 | -79.216917 | 3 | Coffee Shop | Park | Business Service | Dumpling Restaurant | Dim Sum Restaurant | Diner | Discount Store | Dive |
| **4** | M1H | Scarborough | Cedarbrae | 43.773136 | -79.239476 | 4 | Coffee Shop | Bakery | Indian Restaurant | Yoga Studio | Rental Car Location | Burger Joint | Bus Line | Fl S |

Figure 8. Clustering results of Toronto neighborhoods

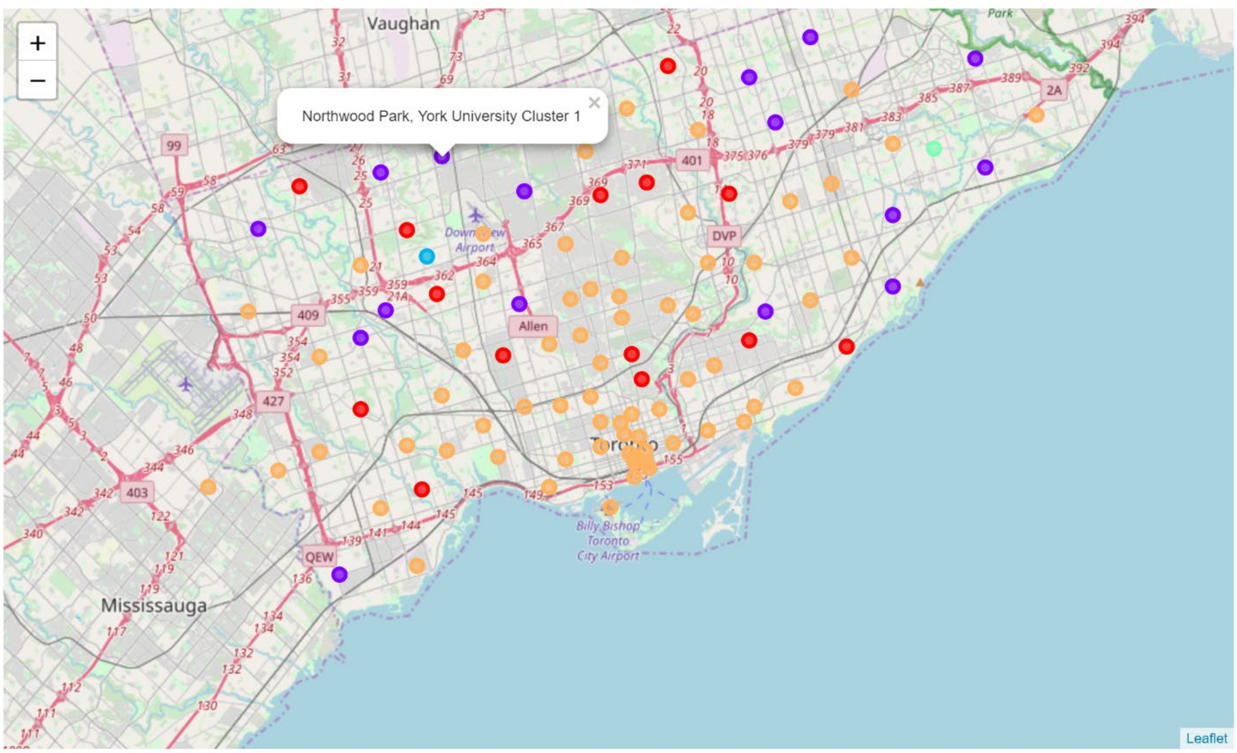



Figure 9. Clustering results as shown in the map (different colors of circles represent different clusters).

The second step of the analysis in this report is to read in a user input of location and find similar clusters as well as individual neighborhoods in Toronto. In order to achieve it, user-proposed place was first used to gain the coordinates. In this report, I used one of my favorite places in Boston area, that is, Little Italy for instance. Figure 10 shows the code to obtain the coordinates of a famous place within Little Italy and Figure 11 presents a visualization of this area.

```
address = '14 N Bennet St, Boston, MA 02113' #Little iItaly, Boston

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
42.365053 -71.053983
```

Figure 10. The python code of obtaining coordinates from an address string.


Figure 11. Visualization of the place for testing purpose in this report: Little Italy in Boston.

After coordinates were identified, Foursquare was used to load surrounding venues of the proposed place (Little Italy) and perform calculations. The results were presented in the Results section.

## Results

Figure 12 shows the extracted venue information of Little Italy, Boston. There were 100 venues in total and 40 unique categories within this area. And this table was then aggregated into a vectorized representation of Little Italy following the same format of the grouped mean data of Toronto neighborhoods for calculation purposes (Figure 13).

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| **0** | Place Query | 42.365053 | -71.053983 | Locale | 42.365055 | -71.053271 | Pizza Place |
| **1** | Place Query | 42.365053 | -71.053983 | The Old North Church | 42.366256 | -71.054386 | Church |
| **2** | Place Query | 42.365053 | -71.053983 | Monica's Mercato | 42.365077 | -71.055444 | Market |
| **3** | Place Query | 42.365053 | -71.053983 | Prezza | 42.364711 | -71.052687 | Italian Restaurant |
| **4** | Place Query | 42.365053 | -71.053983 | Little Italy | 42.363697 | -71.054672 | Neighborhood |

Figure 12. Extracted venue information of Little Italy

| | Neighborhood | Yoga Studio | Accessories Store | Adult Boutique | Afghan Restaurant | African Restaurant | Airport | Airport Food Court | Airport Gate | Airport Lounge | ... | Vegetarian / Vegan Restaurant | Video Game Store | Video Store | Vietnamese Restaurant | Warehou St |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | |

Figure 13. Grouped mean of venues of Little Italy.

Using the model trained to cluster Toronto neighborhoods, Little Italy was predicted to resemble cluster 4 (the orange circles as represented in Figure 9) the most. An inspection of cluster 4 revealed that it is the most common cluster combining food (such as restaurants, bakeries, coffee shops), markets, parks, etc. whose members distribute densely at downtown Toronto (Figure 9). This is also the case for Little Italy, since it is located at downtown Boston and it possesses rich history of early Italian settlement with a lot of restaurants, markets, and historic parks. Figure 14 lists the most common venues within Little Italy area and you can see the pattern.

| | Neighborhood | 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** | 0.0 | Italian Restaurant | Bakery | Pizza Place | Seafood Restaurant | Park | Café | Market | Sandwich Place | Wine Shop | Historic Site |

Figure 14. Most common venues of Little Italy.

We now know cluster 4 is likely to have places in Toronto that resemble Little Italy in Boston. Euclidean distance measure was then used to determine five neighborhoods within that cluster that resembles Little Italy the most. Figure 15 shows the results. According to this figure, all places listed here have “Italian Restaurants” as one of their most common venues, and venues such as Café, Pizza Place, Bakery, etc. that are common in Little Italy also appear frequently in those five neighborhoods.

| | PostCode | Borough | Neighborhood | Latitude | Longitude | Cluster Labels | 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 Mo Commo Venu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **82** | M6P | West Toronto | High Park, The Junction South | 43.661608 | -79.464763 | 4 | Café | Bar | Italian Restaurant | Bakery | Antique Shop | Coffee Shop | Deli / Bodega | Pa |
| **47** | M4S | Central Toronto | Davisville | 43.704324 | -79.388790 | 4 | Coffee Shop | Italian Restaurant | Sushi Restaurant | Sandwich Place | Café | Dessert Shop | Restaurant | Pizz Plac |
| **62** | M5M | North York | Bedford Park, Lawrence Manor East | 43.733283 | -79.419750 | 4 | Italian Restaurant | Coffee Shop | Sushi Restaurant | Juice Bar | Fast Food Restaurant | Skating Rink | Baby Store | Bag Sho |
| **45** | M4P | Central Toronto | Davisville North | 43.712751 | -79.390197 | 4 | Pizza Place | Café | Coffee Shop | Italian Restaurant | Park | Food & Drink Shop | Sporting Goods Shop | Fast Foo Restaura |
| **49** | M4V | Central Toronto | Deer Park, Forest Hill SE, Rathnelly, South Hi... | 43.686412 | -79.400049 | 4 | Coffee Shop | Sushi Restaurant | Italian Restaurant | Thai Restaurant | Café | Gym | Spa | Sandwic Plac |

Figure 15. Top 5 most similar places in Toronto as compared with Little Italy.

## Discussion

As you can see from the previous sections, the query of Little Italy resulted in a predicted cluster label 4 and five neighborhoods that resemble Little Italy the most in Toronto. This information is helpful to me, as a visitor, to explore Toronto in that I can easily identify blocks in a strange city that are similar to places I am familiar with in my own city in some way. Take Little Italy as an example, I personally very much enjoy strolling around that place because of its rich historic sites, good Italian food and bakery, and interesting local markets exhibiting interesting stuff. And yes, if I were not empowered by technology, I could still have found some interesting places in downtown Toronto to explore. However, if I really like places such as Little Italy and eager to find its counterpart in Toronto, all the top five options as shown in Figure 16 are not quite downtown at all!

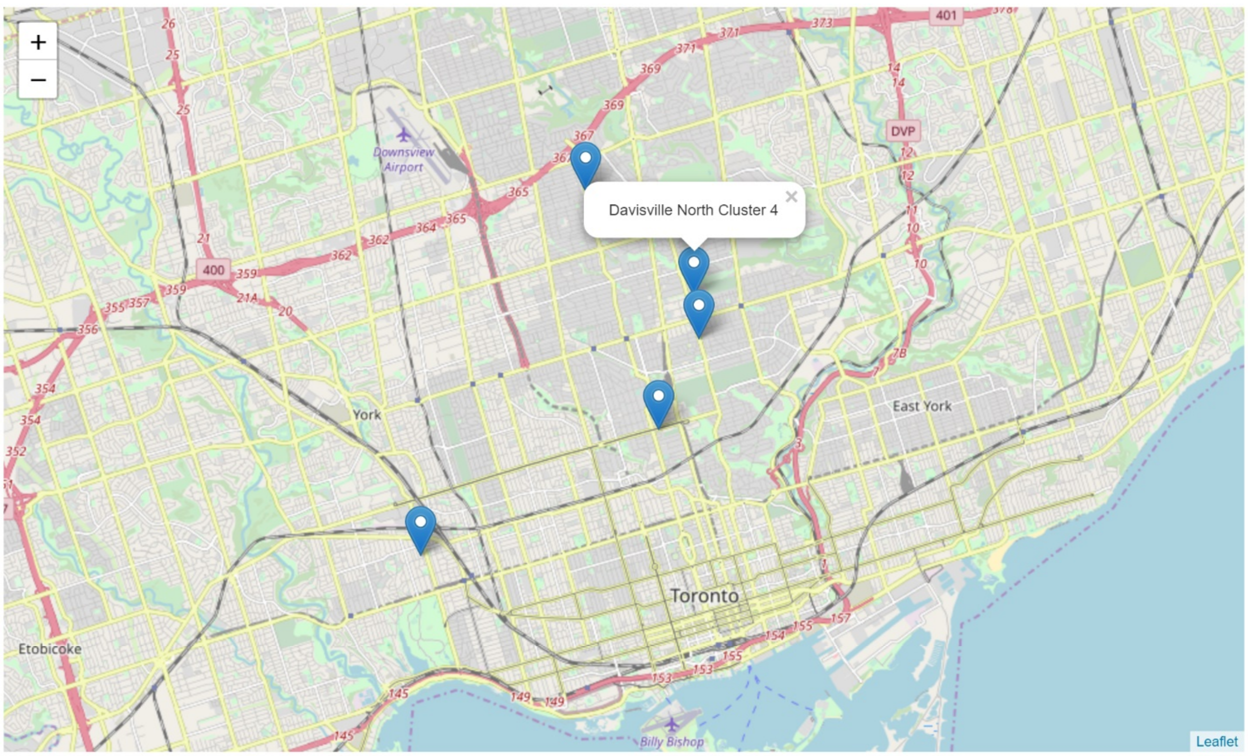



Figure 16. Locations of the top 5 places in Toronto that resemble Little Italy the most.

## Conclusion

This report solves a problem of utilizing a user-input location and then finding similar places in Toronto area and visualizing them. Taking Little Italy, Boston as an example, the results showed that the program worked well in identifying possible interesting location to go for new explorers of this city. The code could be easily adapted to search locations in other cities like Toronto, as long as the neighborhood information (neighborhood name and coordinates) is readily accessible from the web.