Neighborhood Recommender: New York vs Toronto

Introduction

The goal of this project is to build a neighborhood recommendation system. In this project, we will use New York and Toronto data sets to show how the system works. This project presents a solution to those who want to move from one city to another either for personal or business reasons. The system will compare neighborhoods of Toronto city and New York city by exploring the venues of individual neighborhoods. The project would establish a methodology to individuals and businesses to choose what neighborhood best suits their living or business style on the target city based on their preferred neighborhood in their current city. The comparison is accomplished by calculating the similarity index between each neighborhood of both cities. This will help listing the neighborhoods from the most similar to the most dissimilar to a specific neighborhood. Even though the project is focused only on Toronto and New York, the idea can be applied on any set of cities.

Data

Previously in the course, we studied two neighborhood data sets: Toronto and New York. These two data sets will be used in this project beside the venues data that will be obtained from the Foursquare API. Foursquare venues data will be used to create two data sets: one for New York neighborhoods and another for Toronto. These two data sets will be further transformed based on the category of the venues so that we have categories as columns as shown below.

	Neighborhood	Accessories Store	Adult Boutique	African Restaurant	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Audi
0	Battery Park City	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.015385	0.0
1	Carnegie Hill	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.011628	0.000000	0.011628	0.000000	0.000000	0.000000	0.0
2	Central Harlem	0.000000	0.00	0.066667	0.044444	0.00	0.000000	0.000000	0.022222	0.000000	0.000000	0.000000	0.000000	0.0
3	Chelsea	0.000000	0.00	0.000000	0.040000	0.00	0.010000	0.000000	0.040000	0.000000	0.000000	0.010000	0.000000	0.0
4	Chinatown	0.000000	0.00	0.000000	0.040000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.020000	0.000000	0.0
5	Civic Center	0.000000	0.00	0.000000	0.020000	0.01	0.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.0
6	Clinton	0.000000	0.00	0.000000	0.050000	0.00	0.000000	0.000000	0.010000	0.000000	0.000000	0.000000	0.000000	0.0
7	East Harlem	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0

The categories columns will be used as vectors in order to calculate the similarity index, so both data sets must have the same number of categories. The similarity index is simply the Pearson coefficient between the preferred neighborhood and each neighborhood in the target city. The comparison result will be saved into a new data set where we list each neighborhood in the target city along with the similarity index to the preferred neighborhood in the home city. Below is a screenshot of what the result data frame would look like.

Toronto_Neighborhood	*Toronto_n_Longitude*	*Toronto_n_Latitude*	*NYC_Borough*	*NYC_Neighborhood*	*NYC_n_Longitude*	*NYC_n_Latitude*	*Similarity*
Bathurst Quay	*43.63579*	*-79.398329*	*Manhattan*	*Carnegie Hill*	*40.782683*	*-73.953256*	*0.536265*
Bathurst Quay	*43.63579*	*-79.398329*	*Manhattan*	*Financial District*	*40.707107*	*-74.010665*	*0.533281*
Bathurst Quay	*43.63579*	*-79.398329*	*Manhattan*	*Turtle Bay*	*40.752042*	*-73.967708*	*0.524912*
Bathurst Quay	*43.63579*	*-79.398329*	*Manhattan*	*Tudor City*	*40.746917*	*-73.971219*	*0.505284*
Bathurst Quay	*43.63579*	*-79.398329*	*Manhattan*	*Morningside Heights*	*40.808000*	*-73.963896*	*0.477027*

Methodology

We're going to find out how similar each neighborhood is to the another through the **Pearson Correlation Coefficient**. It is used to measure the strength of a linear association between two variables. The formula for finding this coefficient between sets X and Y with N values can be seen in the image below.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Why Pearson Correlation?

Pearson correlation is invariant to scaling, i.e. multiplying all elements by a nonzero constant or adding any constant to all elements. For example, if you have two vectors X and Y, then, pearson(X, Y) == pearson(X, 2 * Y + 3).

The values given by the formula vary from r = -1 to r = 1, where 1 forms a direct correlation between the two entities (it means a perfect positive correlation) and -1 forms a perfect negative correlation.

In our case, a 1 means that the two neighborhoods are similar while a -1 means the opposite.

Instead of writing a code to calculate Pearson Coefficient, SciPy provides a function to easily accomplish that. All we have to do is to pass the two vectors (the two neighborhoods) to the SciPy function pearsonr(neighborhood_1, neighborhood_2). Each vector in this case represents the values of the categories for each neighborhood. Therefore, we need to modify both cities data so that both data frames have the same number of categories and also in the same order.

```
nyc_set = set(nyc_categories)
toronto_set = set(toronto_categories)
common_categories = nyc_set & toronto_set   # intersection of both sets
all_categories = (nyc_set | toronto_set) - {"Neighborhood"}
print("There are {} common categories between nyc_grouped and toronto_grouped data frames.".format(len(common_categories)))
print("The total number of categories in each data frame should be {}.".format(len(all_categories)))
```

```
There are 291 common categories between nyc_grouped and toronto_grouped data frames.
The total number of categories in each data frame should be 475.
```

As shown above, there are a total of 475 unique categories. Our goal is to make sure that each city data frame includes these categories. If a category does not exist in a city, we make it equal to zero. Also, we need to sort the categories in both data frames so that the vectors have corresponding values.

$$v1 = [x1, x2, \ldots, xn] \; and \; v2 = [x1, x2, \ldots, xn]$$

The screenshots below show how to add and sort categories.

```
nyc_missing_cat = all_categories - nyc_set
print("There are {} missing categories from nyc_grouped data frame.".format(len(nyc_missing_cat)))
print(nyc_missing_cat)
```

```
There are 46 missing categories from nyc_grouped data frame.
{'Theme Restaurant', 'Marijuana Dispensary', 'College Theater', 'Hakka Restaurant', 'Aquarium', 'Distribution Center', 'Escape
Room', 'Airport Service', 'Fabric Shop', 'Night Market', 'Elementary School', 'Tunnel', 'Soccer Stadium', 'Housing Developmen
t', 'Hong Kong Restaurant', 'General Travel', 'Costume Shop', 'Tree', 'Food Service', 'Basketball Stadium', 'Poutine Place', 'C
ollege Rec Center', 'College Auditorium', 'University', 'Print Shop', 'College Gym', 'Shoe Repair', 'Outdoor Supply Store', 'In
dian Chinese Restaurant', 'Laser Tag', 'Doner Restaurant', 'Chiropractor', 'Luggage Store', 'Castle', 'Road', 'Indonesian Resta
urant', 'Portuguese Restaurant', 'Auto Dealership', 'Belgian Restaurant', 'Syrian Restaurant', 'Light Rail Station', 'Hospita
l', 'Airport', 'Hockey Arena', 'Golf Driving Range', 'College Stadium'}
```

Adding missing categories to nyc_grouped

```
for category in nyc_missing_cat:
    nyc_grouped[category] = 0
nyc_grouped.head()
```

	Borough	Neighborhood	Latitude	Longitude	Pie Shop	Eastern European Restaurant	Bistro	Hotel Pool	Outdoor Gym	Filipino Restaurant	Farm	Chocolate Shop	Hobby Shop	Other Great Outdoors	Beer Bar	Music Store	Ni
0	Bronx	Wakefield	40.894705	-73.847201	0	0	0	0	0	0	0	0	0	0	0	0	
1	Bronx	Co-op City	40.874294	-73.829939	0	0	0	0	0	0	0	0	0	0	0	0	
2	Bronx	Eastchester	40.887556	-73.827806	0	0	0	0	0	0	0	0	0	0	0	0	
3	Bronx	Fieldston	40.895437	-73.905643	0	0	0	0	0	0	0	0	0	0	0	0	
4	Bronx	Riverdale	40.890834	-73.912585	0	0	0	0	0	0	0	0	0	0	0	0	

Sorting nyc_grouped categories columns

```
main_cols = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']
nyc_grouped = nyc_grouped[main_cols + sorted(list(all_categories))]
nyc_grouped.head()
```

	Borough	Neighborhood	Latitude	Longitude	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Service	Airport Terminal	American Restaurant	Animal Shelter	Antique Shop	Aq
0	Bronx	Wakefield	40.894705	-73.847201	0	0	0	0	0	0	0	0	0	0	
1	Bronx	Co-op City	40.874294	-73.829939	0	0	0	0	0	0	0	0	0	0	
2	Bronx	Eastchester	40.887556	-73.827806	0	0	0	0	0	0	0	0	0	0	
3	Bronx	Fieldston	40.895437	-73.905643	0	0	0	0	0	0	0	0	0	0	
4	Bronx	Riverdale	40.890834	-73.912585	0	0	0	0	0	0	0	0	0	0	

Results

In the project Jupyter Notebook, a function was written to return a data frame with similarity index between one neighborhood from one city and each neighborhood in the target city. Below are two examples that illustrates the results.

Example 1: recommending neighborhoods in Toronto

Let's say that an individual is interested in moving from New York to Toronto. Currently, they live in Bensonhurst, Brooklyn, NY. The goal is to find similar neighborhood/s in Toronto to that of New York. The way this works is similar to a recommender system.

Let's call the similarity function, passing "nyc", "Brooklyn", and "Bensonhurst" as parameters. Below is the head of the sorted data frame returned by the function. Note that these are the five

most similar neighborhoods as the data frame is sorted from the most similar to the most dissimilar. The "Similarity" column represents the similarity index, which ranges from -1 (most dissimilar) to 1 (most similar).

```
df = similarity(city='nyc', borough='Brooklyn', neighborhood='Bensonhurst')
df.head()
```

_Borough	NYC_Neighborhood	NYC_n_Longitude	NYC_n_Latitude	Toronto_Borough	Toronto_Neighborhood	Toronto_n_Longitude	Toronto_n_Latitude	Similarity
Brooklyn	Bensonhurst	-73.99518	40.611009	Scarborough	Milliken	-79.301763	43.823174	0.391332
Brooklyn	Bensonhurst	-73.99518	40.611009	Central Toronto	Davisville	-79.397291	43.697936	0.388051
Brooklyn	Bensonhurst	-73.99518	40.611009	Central Toronto	Davisville North	-79.397291	43.697936	0.388051
Brooklyn	Bensonhurst	-73.99518	40.611009	Central Toronto	Lawrence Park	-79.403252	43.729199	0.375662
Brooklyn	Bensonhurst	-73.99518	40.611009	Scarborough	Agincourt North	-79.266439	43.808038	0.355401

Let's check the most common venues in Bensonhurst and the most similar neighborhood in Toronto.

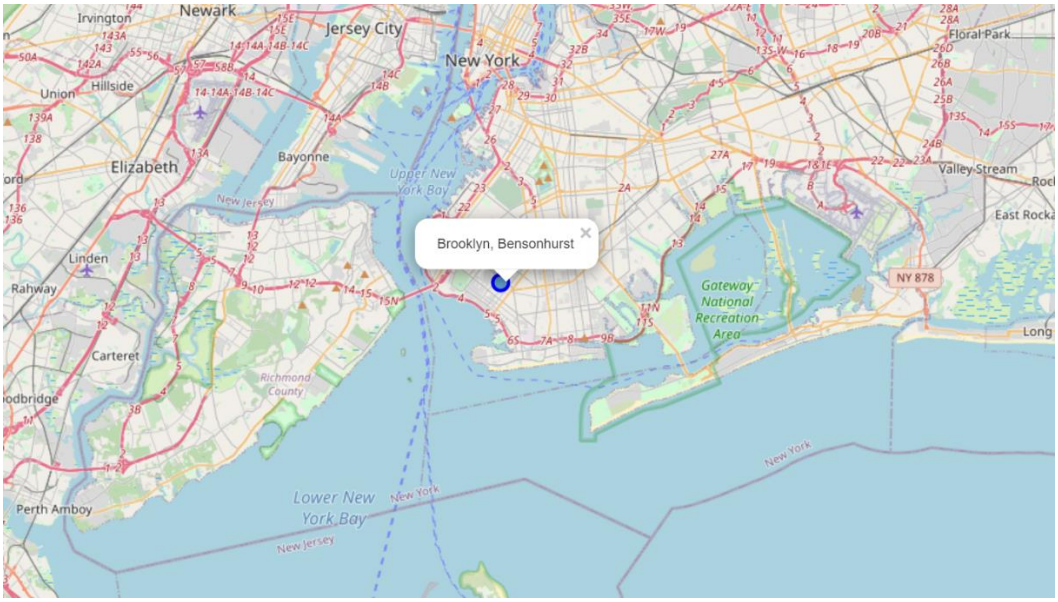
```
nyc_venues_sorted[nyc_venues_sorted.Neighborhood=='Bensonhurst']
```

	Borough	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
47	Brooklyn	Bensonhurst	Chinese Restaurant	Donut Shop	Sushi Restaurant	Ice Cream Shop	Italian Restaurant	Park	Hotpot Restaurant	Noodle House	Cha Chaan Teng	Spa

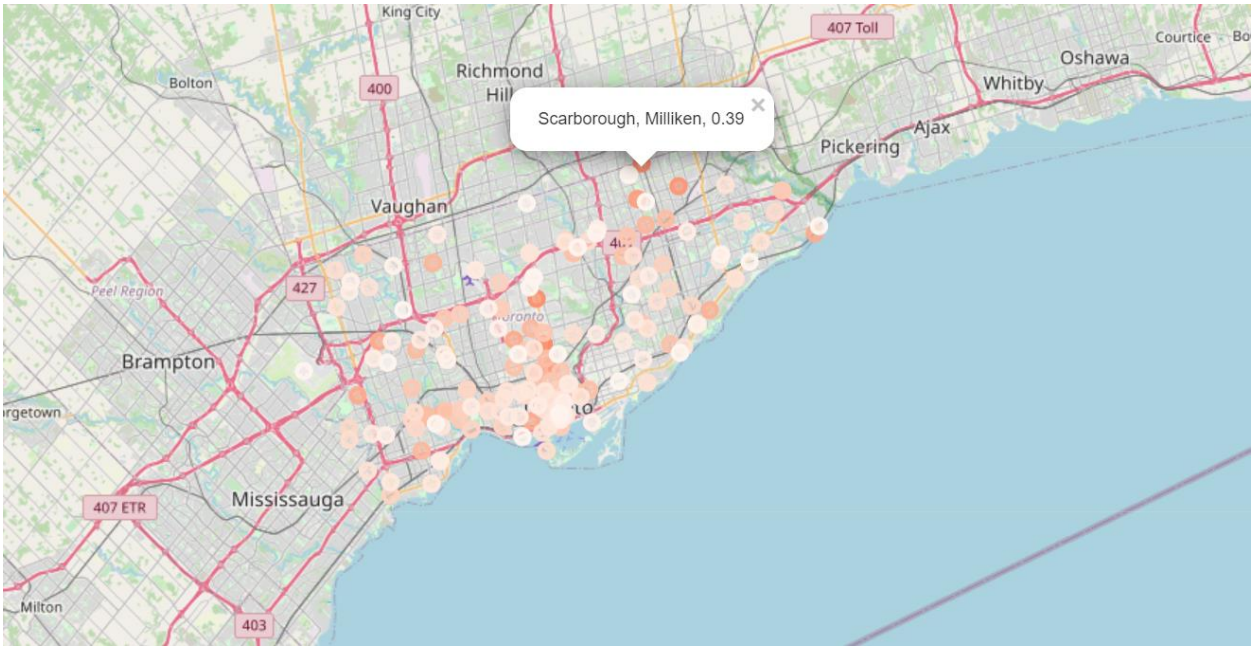
```
toronto_venues_sorted[toronto_venues_sorted.Neighborhood=='Milliken']
```

	Borough	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
144	Scarborough	Milliken	Chinese Restaurant	Bubble Tea Shop	Bakery	Japanese Restaurant	Intersection	Noodle House	Asian Restaurant	Pet Store	Hong Kong Restaurant	Korean Restaurant

As you can see, both neighborhoods feature Asian venues: Sushi, Chinese, and Japanese Restaurants. Also, they are both kind of suburban neighborhoods as shown in the two maps below.



Below is a map for Toronto neighborhoods. The darker the color, the more similar the neighborhood to Bensonhurst neighborhood.



Example 2: recommending neighborhoods in New York City

In this example, we would like to find the New York neighborhoods that are most similar to a Toronto neighborhood. The target Toronto neighborhood is Downtown Toronto, Bathurst Quay.

```
df2 = similarity(city='toronto', borough='Downtown Toronto', neighborhood='Bathurst Quay')
df2.head()
```

nto_Borough	Toronto_Neighborhood	Toronto_n_Longitude	Toronto_n_Latitude	NYC_Borough	NYC_Neighborhood	NYC_n_Longitude	NYC_n_Latitude	Similarity
ntown Toronto	Bathurst Quay	-79.398329	43.63579	Manhattan	Carnegie Hill	-73.953256	40.782683	0.536265
ntown Toronto	Bathurst Quay	-79.398329	43.63579	Manhattan	Financial District	-74.010665	40.707107	0.533281
ntown Toronto	Bathurst Quay	-79.398329	43.63579	Manhattan	Turtle Bay	-73.967708	40.752042	0.524912
ntown Toronto	Bathurst Quay	-79.398329	43.63579	Manhattan	Tudor City	-73.971219	40.746917	0.505284
ntown Toronto	Bathurst Quay	-79.398329	43.63579	Manhattan	Morningside Heights	-73.963896	40.808000	0.477027

Let's check the most common venues in Bathhurst Quay and the most similar neighborhoods in New York City.

```
toronto_venues_sorted[toronto_venues_sorted.Neighborhood=='Bathurst Quay']
```

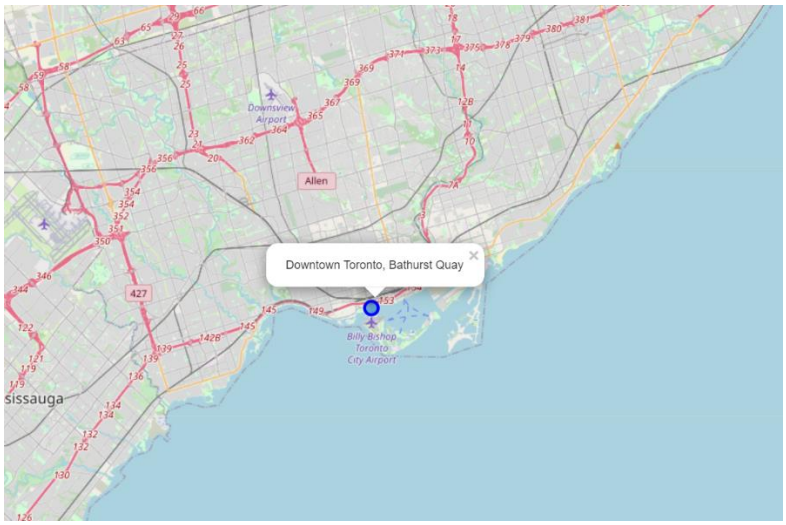
	Borough	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
157	Downtown Toronto	Bathurst Quay	Coffee Shop	Café	Grocery Store	Park	Sculpture Garden	Garden	Ramen Restaurant	Sushi Restaurant	Airport Service	Tunnel

```
nyc_venues_sorted[nyc_venues_sorted.Neighborhood=='Carnegie Hill']
```

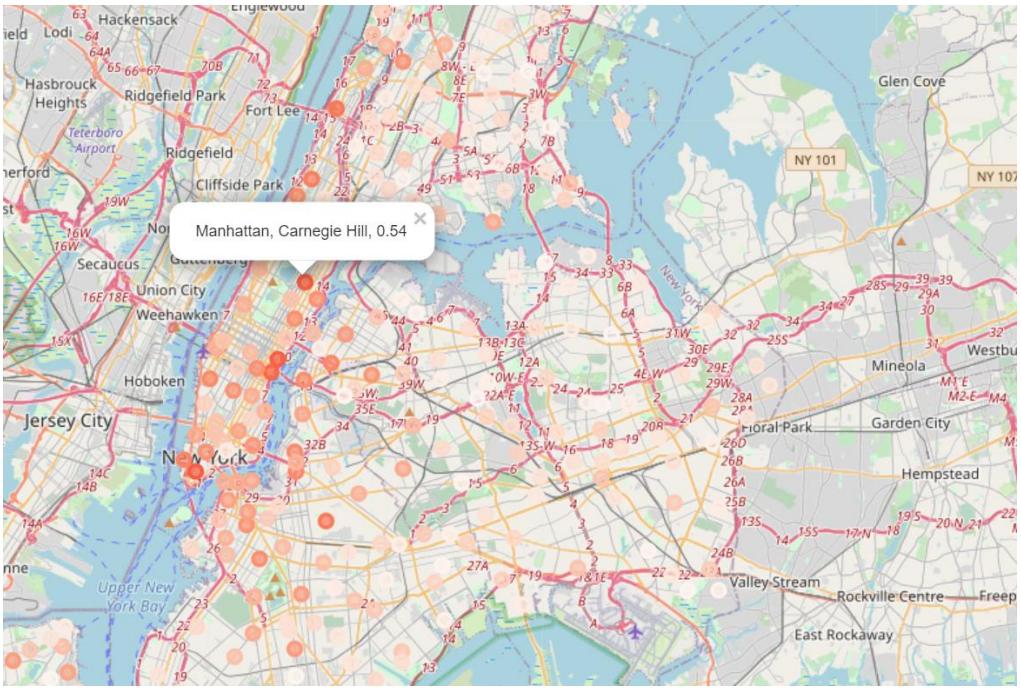
	Borough	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
246	Manhattan	Carnegie Hill	Coffee Shop	Café	Pizza Place	Gym	Gym / Fitness Center	Yoga Studio	Bar	Wine Shop	Bookstore	French Restaurant

As you can see, both neighborhoods have a lot in common either in terms of venue categories or their locations in the city; both are downtown neighborhoods as shown in the maps below. Also, if you look closely at these two maps, both neighborhoods have parks nearby.

Bathurst Quay Map



New York Neighborhoods Similarity Map



Discussion

As you might have noticed in the two examples, the similarity index is slightly in the low range (<60%) even for the most similar neighborhood. There are many reasons for that including the large number of categories, 475. Also, more testing is needed to check which similarity measure best fit this problem, such as Euclidean distance and Jaccard Coefficient. Nevertheless, the two examples also show that Pearson coefficient is still a solid approach.

Conclusion

This project presented a simple and yet efficient approach to building a neighborhood recommender system. Even though the system was based only on venue categories, more data can be added to make the system more efficient such as using demographic data of each neighborhoods. Also, the system can help businesses that are looking to scale up and open new branches by recommending neighborhoods based on the neighborhoods of the current successful branches.