Coursera Capstone Project

IBM Data Science Specialization

March, 2019, Ming

In [1]:
```
# import libraries

import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json
from geopy.geocoders import Nominatim
import requests
from pandas.io.json import json_normalize
from sklearn.cluster import KMeans

import folium
import matplotlib.cm as cm
import matplotlib.colors as colors

print('Libraries imported.')
```

```
Libraries imported.
```

In [2]:
```
CLIENT_ID = '' # your Foursquare ID
CLIENT_SECRET = '' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
radius = 500
LIMIT = 100
```

```
Your credentails:
CLIENT_ID: 1RXZTD5EH50RBI3THQEE2NOJZH0QACXD4DUM30KV1M1KLTVS
CLIENT_SECRET:TBALVRU1HPKFVXNRTBN3AQFK12QDEMFEYAR4ISK1HWINXM4C
```

1. Introduction

In this project, we will use clustering to compare communities of Manhattan, New York and Toronto. Utilizing Four Square venues data in the two cities, we will 1) know which neighborhoods are similar between the two cities, 2) visualize how similar neighborhoods locate in the two cities, 3) picture similarities and differences of lifestypes bwteen Manhattan and Toronto.

People interested in this project would be residents in either city who are interested in the other, and people who are interested in moving to one of the cities.

2. Data

We will use Four Square API as the main data source. Neighborhood data of Manhattan is from the Coursera class data file (https://cocl.us/new_york_dataset (https://cocl.us/new_york_dataset)). And the Neighborhood data of Toronto is from the Wikipedia Page we used for Week 3 (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M)). We use Four Square API calls to get venues within 500m from the neighborhood coordinates, and do clustering based on frequency of venue categories within the neighborhood. Neighborhoods of Manhattan and Toronto are put together to do the clustering, so that similar neighborhoods will end up within the same cluster.

Let's load our data here

First Manhattan venues data

In [3]:

```
# open downloaded New York Data
with open('nyu_geojson.json') as json_data:
    newyork_data = json.load(json_data)

# create dataframe
ny_data = newyork_data['features']
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']
mht_neighborhoods = pd.DataFrame(columns=column_names)
for data in ny_data:
    borough = neighborhood_name = data['properties']['borough']
    if borough != 'Manhattan':
        continue
    neighborhood_name = data['properties']['name']
    neighborhood_latlon = data['geometry']['coordinates']
    neighborhood_lat = neighborhood_latlon[1]
    neighborhood_lon = neighborhood_latlon[0]

    mht_neighborhoods = mht_neighborhoods.append({'Borough': borough,
                                                  'Neighborhood': neighborhood_n
ame,
                                                  'Latitude': neighborhood_lat,
                                                  'Longitude': neighborhood_lon
}, ignore_index=True)
mht_neighborhoods.head()
```

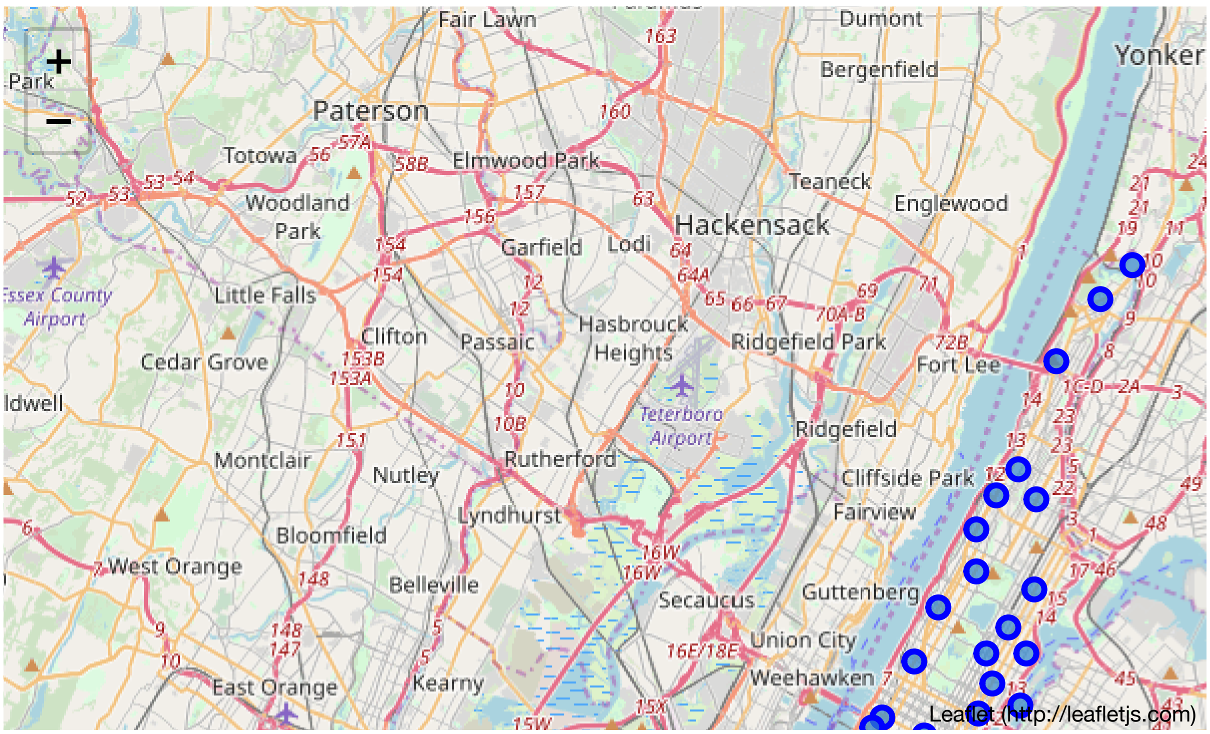
Out[3]:

	Borough	Neighborhood	Latitude	Longitude
0	Manhattan	Marble Hill	40.876551	-73.910660
1	Manhattan	Chinatown	40.715618	-73.994279
2	Manhattan	Washington Heights	40.851903	-73.936900
3	Manhattan	Inwood	40.867684	-73.921210
4	Manhattan	Hamilton Heights	40.823604	-73.949688

In [4]:
```
# visually verify that we indeed load neighborhoods in Manhattan
address = 'Manhattan, NY'
geolocator = Nominatim(user_agent="mht_explorer")
location = geolocator.geocode(address)
latitude1 = location.latitude
longitude1 = location.longitude

map_manhattan = folium.Map(location=[latitude1, longitude1], zoom_start=
11)
for lat, lng, label in zip(mht_neighborhoods['Latitude'], mht_neighborho
ods['Longitude'], mht_neighborhoods['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan)
map_manhattan
```

Out[4]:



In [5]:

```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']

# function to repeat the same process to all the neighborhoods in the da
taframe
def getNearbyVenues(names, latitudes, longitudes, radius=500):
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id=
{}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)
        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['ite
ms']
        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])
    nearby_venues = pd.DataFrame([item for venue_list in venues_list for
item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']
    return(nearby_venues)
```

In [6]:
```
# request venues information using API
mht_venues = getNearbyVenues(names=mht_neighborhoods['Neighborhood'],
                                   latitudes=mht_neighborhoods['Latitud
e'],
                                   longitudes=mht_neighborhoods['Longitu
de'])
mht_venues.shape
```

Out[6]: (3307, 7)

Then Toronto venues data

In [7]:
```
# define the dataframe columns
column_names = ['PostalCode', 'Borough', 'Neighborhood']
# instantiate the dataframe
trt_data = pd.DataFrame(columns=column_names)
```

In [8]:
```
# use BeautifulSoup to scrape the data table from wikipedia
from bs4 import BeautifulSoup
from urllib.request import urlopen
url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
soup = BeautifulSoup(urlopen(url))
table = soup.find('table', class_="wikitable")
```

In [9]:
```
# fill in dataframe
table_rows = table.find_all('tr')
for tr in table_rows:
    td = tr.find_all('td')
    row = [x.text.strip() for x in td]
    if len(row) != 3:
        continue
    postcode, borough, neighborhood = row
    if borough == 'Not assigned':
        continue
    if 'Toronto' not in borough:
        continue
    if neighborhood == 'Not assigned':
        neighborhood = borough
    trt_data = trt_data.append({'PostalCode': postcode,
                                'Borough': borough,
                                'Neighborhood': neighborhood}, ignore_in
dex=True)
```

In [10]:

```
# clean up data by combining rows with same PostalCode and Borough into
 the same row, and join their Neighborhood
trt_data = trt_data.groupby(['PostalCode','Borough'])['Neighborhood'].ap
ply(lambda x: "%s" % ', '.join(x)).reset_index()
trt_data.head()
```

Out[10]:

	PostalCode	Borough	Neighborhood
0	M4E	East Toronto	The Beaches
1	M4K	East Toronto	The Danforth West, Riverdale
2	M4L	East Toronto	The Beaches West, India Bazaar
3	M4M	East Toronto	Studio District
4	M4N	Central Toronto	Lawrence Park

In [11]:

```
# use the downloaded csv file provided by the classs site for Toronto ne
ighborhood coordinates
coords = pd.read_csv('Geospatial_Coordinates.csv')
coords.rename(columns = {'Postal Code': 'PostalCode'}, inplace = True)
trt_data = pd.merge(trt_data, coords, on = 'PostalCode')
trt_data.head()
```

Out[11]:

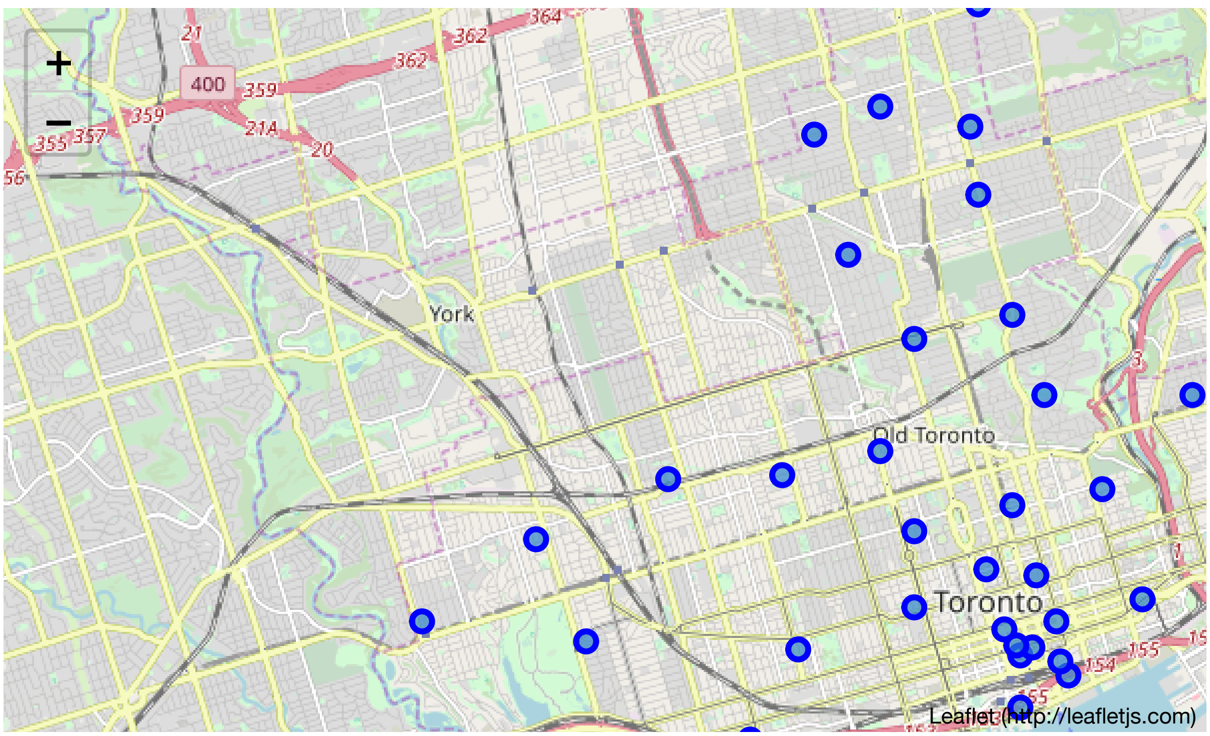
	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M4E	East Toronto	The Beaches	43.676357	-79.293031
1	M4K	East Toronto	The Danforth West, Riverdale	43.679557	-79.352188
2	M4L	East Toronto	The Beaches West, India Bazaar	43.668999	-79.315572
3	M4M	East Toronto	Studio District	43.659526	-79.340923
4	M4N	Central Toronto	Lawrence Park	43.728020	-79.388790

In [12]:
```
# visually verify that we indeed load neighborhoods in Toronto
address = 'Toronto, ON'

geolocator = Nominatim(user_agent="trt_explorer")
location = geolocator.geocode(address)
latitude2 = location.latitude
longitude2 = location.longitude

map_trt = folium.Map(location=[latitude2, longitude2], zoom_start=12)
for lat, lng, borough, neighborhood in zip(trt_data['Latitude'], trt_dat
a['Longitude'], trt_data['Borough'], trt_data['Neighborhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_trt)
map_trt
```

Out[12]:



In [13]:
```
# request venues information using API
trt_venues = getNearbyVenues(names=trt_data['Neighborhood'],
                                   latitudes=trt_data['Latitude'],
                                   longitudes=trt_data['Longitude'])
trt_venues.shape
```

Out[13]:
```
(1693, 7)
```

Methodology section

Exploratory data analysis

In [59]:
```
print('There are {} unique categories in Manhattan.'.format(len(mht_venu
es['Venue Category'].unique())))
print('There are {} unique categories in Toronto.'.format(len(trt_venues
['Venue Category'].unique())))
```

```
There are 329 unique categories in Manhattan.
There are 235 unique categories in Toronto.
```

In [60]:
```
# one hot encoding
manhattan_onehot = pd.get_dummies(mht_venues[['Venue Category']], prefix
="", prefix_sep="")
# add neighborhood column back to dataframe
manhattan_onehot['Neighborhood'] = mht_venues['Neighborhood']
# move neighborhood column to the first column
fixed_columns = [manhattan_onehot.columns[-1]] + list(manhattan_onehot.c
olumns[:-1])
manhattan_onehot = manhattan_onehot[fixed_columns]
manhattan_onehot.head()
```

Out[60]:

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	American Restaurant	Animal Shelter	Antique Shop
0	Marble Hill	0	0	0	0	0	0	0
1	Marble Hill	0	0	0	0	0	0	0
2	Marble Hill	0	0	0	0	0	0	0
3	Marble Hill	0	0	0	0	0	0	0
4	Marble Hill	0	0	0	0	0	0	0

In [61]:
```
manhattan_onehot.shape
```

Out[61]: (3307, 330)

In [62]:
```
# one hot encoding
trt_onehot = pd.get_dummies(trt_venues[['Venue Category']], prefix="", p
refix_sep="")
# add neighborhood column back to dataframe
trt_onehot['Neighborhood'] = trt_venues['Neighborhood']
# move neighborhood column to the first column
fixed_columns = [trt_onehot.columns[-1]] + list(trt_onehot.columns[:-1])
trt_onehot = trt_onehot[fixed_columns]
trt_onehot.head()
```

Out[62]:

	Yoga Studio	Adult Boutique	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0

In [18]:
```
trt_onehot.shape
```

Out[18]: (1693, 235)

Next, group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

In [94]:
```
manhattan_grouped = manhattan_onehot.groupby('Neighborhood').mean().rese
t_index()
manhattan_grouped.shape
```

Out[94]: (40, 330)

In [95]:
```
trt_grouped = trt_onehot.groupby('Neighborhood').mean().reset_index()
trt_grouped.shape
```

Out[95]: (38, 235)

Put the two cities in one single dataframe

In [96]:
```
mht = manhattan_grouped
trt = trt_grouped
mht.insert(1,'City', 'NY')
trt.insert(1, 'City', 'TRT')
both = pd.concat([mht, trt], axis = 0, join = 'inner')
both.head()
```

Out[96]:

	Neighborhood	City	Adult Boutique	Afghan Restaurant	American Restaurant	Antique Shop	Art Gallery	Art Museum	Arts & Crafts Store	R
0	Battery Park City	NY	0.0	0.0	0.010309	0.0	0.000000	0.00	0.0	
1	Carnegie Hill	NY	0.0	0.0	0.010000	0.0	0.000000	0.01	0.0	
2	Central Harlem	NY	0.0	0.0	0.046512	0.0	0.023256	0.00	0.0	
3	Chelsea	NY	0.0	0.0	0.040000	0.0	0.020000	0.00	0.0	
4	Chinatown	NY	0.0	0.0	0.040000	0.0	0.000000	0.00	0.0	

Clustering!

In [98]:
```
# set number of clusters
kclusters = 8
both_clustering = both.drop(['Neighborhood','City'], 1)
both_clustering.head()
```

Out[98]:

	Adult Boutique	Afghan Restaurant	American Restaurant	Antique Shop	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports
0	0.0	0.0	0.010309	0.0	0.000000	0.00	0.0	0.00	0.010309
1	0.0	0.0	0.010000	0.0	0.000000	0.01	0.0	0.00	0.000000
2	0.0	0.0	0.046512	0.0	0.023256	0.00	0.0	0.00	0.000000
3	0.0	0.0	0.040000	0.0	0.020000	0.00	0.0	0.02	0.000000
4	0.0	0.0	0.040000	0.0	0.000000	0.00	0.0	0.02	0.000000

In [101]:
```
# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=9).fit(both_clusterin
g)
# check cluster labels generated for each row in the dataframe
kmeans.labels_[:]
```

Out[101]:
```
array([7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1,
       7, 1, 1, 7, 1, 1, 7, 1, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
1,
       1, 0, 0, 1, 0, 1, 0, 7, 7, 0, 0, 1, 0, 5, 0, 0, 0, 1, 2, 1, 4,
1,
       1, 6, 3, 0, 1, 0, 0, 0, 7, 7, 7, 1], dtype=int32)
```

Results section

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

In [103]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

In [105]:

```
top_venues = both.drop(['City'],1)
num_top_venues = 10
indicators = ['st', 'nd', 'rd']
# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators
[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = top_venues['Neighborhood']

for ind in np.arange(top_venues.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venue
s(top_venues.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.insert(1, 'City', both['City'])
neighborhoods_venues_sorted.head()
```

Out[105]:

	Neighborhood	City	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 Cor
0	Battery Park City	NY	Park	Coffee Shop	Hotel	Wine Shop	Italian Restaurant	Gym	I
1	Carnegie Hill	NY	Pizza Place	Café	Coffee Shop	Cosmetics Shop	Bar	Japanese Restaurant	Boo
2	Central Harlem	NY	American Restaurant	Seafood Restaurant	French Restaurant	Chinese Restaurant	Cosmetics Shop	Pizza Place	D
3	Chelsea	NY	Coffee Shop	Italian Restaurant	Ice Cream Shop	American Restaurant	Bakery	Nightclub	
4	Chinatown	NY	Chinese Restaurant	Dim Sum Restaurant	American Restaurant	Vietnamese Restaurant	Cocktail Bar	Bubble Tea Shop	N I

Put neighborhood location and top venues into one dataframe

In [106]:
```
both_coords = pd.concat([mht_neighborhoods, trt_data.drop(['PostalCode'
], 1)])
# add clustering labels
both_coords.insert(0, 'Cluster Labels', kmeans.labels_)

# merge toronto_grouped with toronto_data to add latitude/longitude for
 each neighborhood
merged = both_coords.join(neighborhoods_venues_sorted.set_index('Neighbo
rhood'), on='Neighborhood')

merged.head() # check the last columns!
```

Out[106]:

	Cluster Labels	Borough	Neighborhood	Latitude	Longitude	City	1st Most Common Venue	2nd Most Common Venue	3rd Mo Comm Ven
0	7	Manhattan	Marble Hill	40.876551	-73.910660	NY	Discount Store	Coffee Shop	Ice Crea Sh
1	1	Manhattan	Chinatown	40.715618	-73.994279	NY	Chinese Restaurant	Dim Sum Restaurant	Americ Restaura
2	1	Manhattan	Washington Heights	40.851903	-73.936900	NY	Café	Bakery	De Bode
3	1	Manhattan	Inwood	40.867684	-73.921210	NY	Café	Mexican Restaurant	De Bode
4	1	Manhattan	Hamilton Heights	40.823604	-73.949688	NY	Mexican Restaurant	Coffee Shop	Ca

Now seperate the two cities and show labels on maps

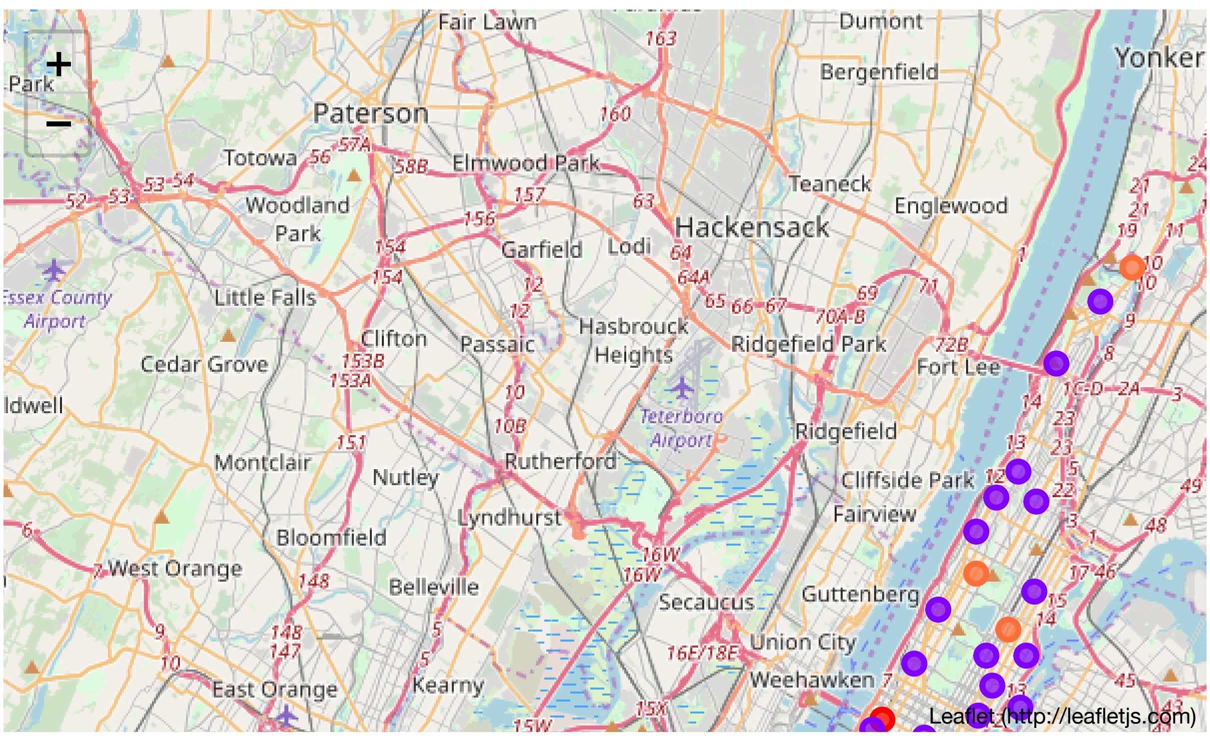
In [108]:
```
ny = merged[merged['City'] == 'NY']
trt = merged[merged['City'] == 'TRT']
```

In [110]:
```
# create NY map
nymap_clusters = folium.Map(location=[latitude1, longitude1], zoom_start
=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(ny['Latitude'], ny['Longitude'], ny['N
eighborhood'], ny['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_ht
ml=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(nymap_clusters)
nymap_clusters
```

Out[110]:

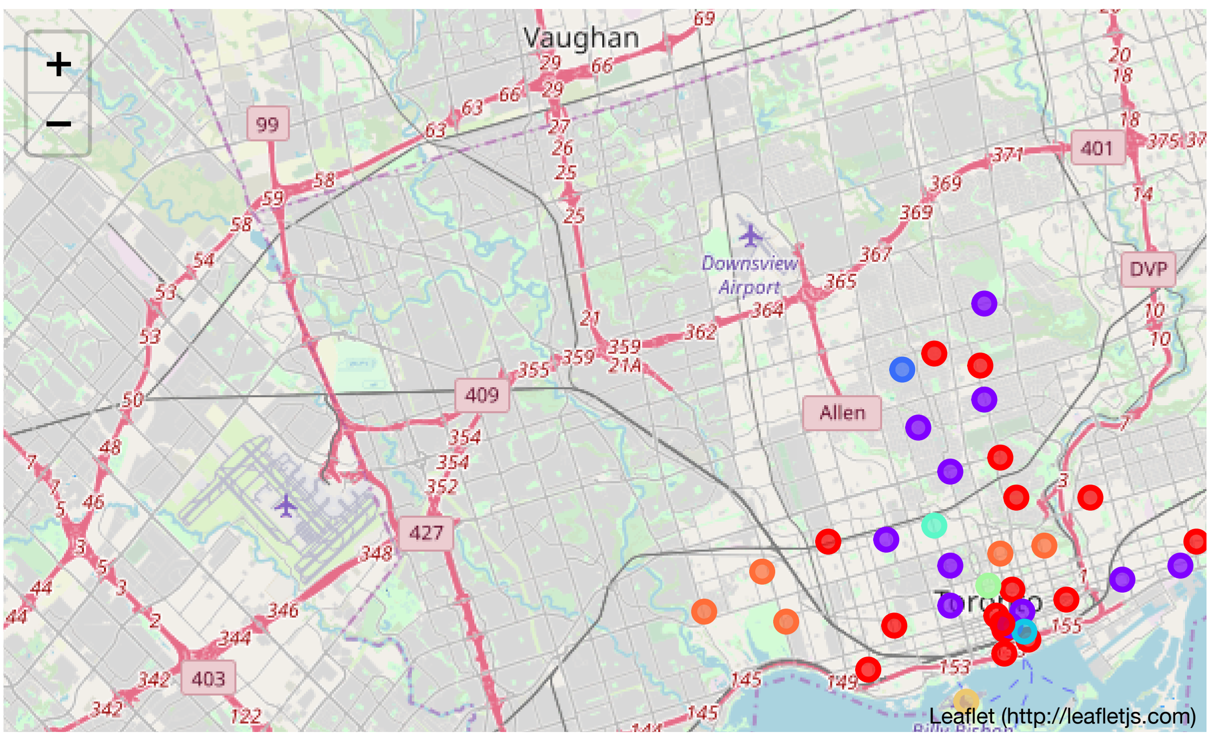


In [111]:
```
# create Toronto map
trtmap_clusters = folium.Map(location=[latitude2, longitude2], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(trt['Latitude'], trt['Longitude'], trt['Neighborhood'], trt['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(trtmap_clusters)
trtmap_clusters
```

Out[111]:



Now lets look at top venues for each cluster

In [112]:
```
merged.loc[merged['Cluster Labels'] == 0, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[112]:

	Borough	City	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 Con V
14	Manhattan	NY	Theater	American Restaurant	Gym / Fitness Center	Italian Restaurant	Coffee Shop	Spa	Wine
1	East Toronto	TRT	Greek Restaurant	Coffee Shop	Ice Cream Shop	Bookstore	Italian Restaurant	Yoga Studio	De
2	East Toronto	TRT	Sandwich Place	Pizza Place	Pub	Burger Joint	Park	Liquor Store	F
5	Central Toronto	TRT	Breakfast Spot	Gym	Dance Studio	Food & Drink Shop	Sandwich Place	Restaurant	E
6	Central Toronto	TRT	Sporting Goods Shop	Coffee Shop	Clothing Store	Yoga Studio	Cosmetics Shop	Dessert Shop	
8	Central Toronto	TRT	Playground	Trail	Eastern European Restaurant	Flower Shop	Flea Market	Fish Market	Fi Resta
10	Downtown Toronto	TRT	Park	Playground	Trail	Eastern European Restaurant	Flea Market	Fish Market	Fi Resta
13	Downtown Toronto	TRT	Coffee Shop	Park	Pub	Bakery	Café	Theater	Brea
14	Downtown Toronto	TRT	Clothing Store	Coffee Shop	Café	Cosmetics Shop	Middle Eastern Restaurant	Diner	B Tea
16	Downtown Toronto	TRT	Coffee Shop	Cocktail Bar	Seafood Restaurant	Café	Restaurant	Cheese Shop	I Resta
18	Downtown Toronto	TRT	Coffee Shop	Café	Steakhouse	Thai Restaurant	Bar	Burger Joint	Cosn
19	Downtown Toronto	TRT	Coffee Shop	Hotel	Pizza Place	Café	Italian Restaurant	Restaurant	B
20	Downtown Toronto	TRT	Coffee Shop	Café	Hotel	American Restaurant	Restaurant	Gastropub	Bo
29	Downtown Toronto	TRT	Café	Coffee Shop	Hotel	Restaurant	American Restaurant	Gastropub	Se Resta
31	West Toronto	TRT	Discount Store	Pharmacy	Supermarket	Bakery	Pizza Place	Café	M Ea Resta
32	West Toronto	TRT	Bar	Coffee Shop	Asian Restaurant	Café	Pizza Place	Cocktail Bar	B
33	West Toronto	TRT	Café	Coffee Shop	Breakfast Spot	Restaurant	Climbing Gym	Bar	B

In [113]:
```
merged.loc[merged['Cluster Labels'] == 1, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[113]:

	Borough	City	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
1	Manhattan	NY	Chinese Restaurant	Dim Sum Restaurant	American Restaurant	Vietnamese Restaurant	Cocktail Bar	Bubble Tea Shop
2	Manhattan	NY	Café	Bakery	Deli / Bodega	Donut Shop	Latin American Restaurant	Tapas Restaurant
3	Manhattan	NY	Café	Mexican Restaurant	Deli / Bodega	Lounge	Pizza Place	Restaurant
4	Manhattan	NY	Mexican Restaurant	Coffee Shop	Café	Pizza Place	Yoga Studio	Liquor Store
5	Manhattan	NY	Mexican Restaurant	Seafood Restaurant	Italian Restaurant	Coffee Shop	Bar	Chinese Restaurant
6	Manhattan	NY	American Restaurant	Seafood Restaurant	French Restaurant	Chinese Restaurant	Cosmetics Shop	Pizza Place
7	Manhattan	NY	Mexican Restaurant	Bakery	Latin American Restaurant	Deli / Bodega	Coffee Shop	Thai Restaurant
8	Manhattan	NY	Italian Restaurant	Coffee Shop	Juice Bar	Bakery	Boutique	Art Gallery
9	Manhattan	NY	Italian Restaurant	Coffee Shop	Bar	Gym	Pizza Place	Mexican Restaurant
10	Manhattan	NY	Italian Restaurant	Sushi Restaurant	Coffee Shop	Pizza Place	Gym / Fitness Center	Burger Joint
11	Manhattan	NY	Coffee Shop	Sandwich Place	Park	Farmers Market	Deli / Bodega	Café
12	Manhattan	NY	Italian Restaurant	Bar	Burger Joint	Wine Bar	Vegetarian / Vegan Restaurant	Indian Restaurant
13	Manhattan	NY	Gym / Fitness Center	Theater	Italian Restaurant	Plaza	Concert Hall	Café
15	Manhattan	NY	Clothing Store	Hotel	Steakhouse	Theater	Bakery	American Restaurant
16	Manhattan	NY	Coffee Shop	Hotel	Japanese Restaurant	Sandwich Place	Italian Restaurant	Bar
17	Manhattan	NY	Coffee Shop	Italian Restaurant	Ice Cream Shop	American Restaurant	Bakery	Nightclub
18	Manhattan	NY	Italian Restaurant	French Restaurant	Clothing Store	Sushi Restaurant	Café	Indian Restaurant
19	Manhattan	NY	Bar	Ice Cream Shop	Wine Bar	Mexican Restaurant	Chinese Restaurant	Ramen Restaurant
20	Manhattan	NY	Café	Ramen Restaurant	Chinese Restaurant	Art Gallery	Coffee Shop	Shoe Store

	Borough	City	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	
21	Manhattan	NY	Italian Restaurant	Park	Café	American Restaurant	Spa	Gym	
23	Manhattan	NY	Clothing Store	Boutique	Women's Store	Coffee Shop	Mediterranean Restaurant	Shoe Store	
24	Manhattan	NY	Italian Restaurant	New American Restaurant	Wine Bar	Gastropub	Cosmetics Shop	Jazz Club	
26	Manhattan	NY	Coffee Shop	American Restaurant	Bookstore	Park	Burger Joint	Café	
27	Manhattan	NY	Italian Restaurant	Cocktail Bar	American Restaurant	Bagel Shop	Coffee Shop	Pizza Place	
29	Manhattan	NY	Coffee Shop	Hotel	Wine Shop	Steakhouse	Bar	Gym	
31	Manhattan	NY	Italian Restaurant	French Restaurant	Cocktail Bar	Coffee Shop	American Restaurant	Grocery Store	
32	Manhattan	NY	Gym / Fitness Center	Bakery	Italian Restaurant	French Restaurant	Cocktail Bar	Yoga Studio	
33	Manhattan	NY	Korean Restaurant	Japanese Restaurant	Hotel	Hotel Bar	Cocktail Bar	Gym / Fitness Center	
34	Manhattan	NY	Gym / Fitness Center	Italian Restaurant	Indian Restaurant	Furniture / Home Store	Juice Bar	American Restaurant	
35	Manhattan	NY	Italian Restaurant	Sushi Restaurant	Coffee Shop	Wine Bar	Hotel	Indian Restaurant	S
36	Manhattan	NY	Park	Mexican Restaurant	Café	Asian Restaurant	Pizza Place	Deli / Bodega	
37	Manhattan	NY	Bar	Playground	Park	Harbor / Marina	Farmers Market	Boat or Ferry	
38	Manhattan	NY	Yoga Studio	Gym	American Restaurant	New American Restaurant	Gym / Fitness Center	Japanese Restaurant	
39	Manhattan	NY	Coffee Shop	Café	Hotel	Italian Restaurant	Restaurant	American Restaurant	
0	East Toronto	TRT	Health Food Store	Pub	Park	Coffee Shop	Yoga Studio	Eastern European Restaurant	F
3	East Toronto	TRT	Café	Coffee Shop	American Restaurant	Gastropub	Italian Restaurant	Bakery	
4	Central Toronto	TRT	Park	Bus Line	Yoga Studio	Ethiopian Restaurant	Flower Shop	Flea Market	F
7	Central Toronto	TRT	Dessert Shop	Sandwich Place	Coffee Shop	Sushi Restaurant	Café	Pizza Place	
9	Central Toronto	TRT	Coffee Shop	Pub	Convenience Store	Sushi Restaurant	Sports Bar	Bagel Shop	

	Borough	City	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	
15	Downtown Toronto	TRT	Coffee Shop	Restaurant	Café	Hotel	Park	Italian Restaurant	
21	Downtown Toronto	TRT	Coffee Shop	Café	Restaurant	Hotel	American Restaurant	Seafood Restaurant	
23	Central Toronto	TRT	Mexican Restaurant	Trail	Sushi Restaurant	Jewelry Store	Yoga Studio	Ethiopian Restaurant	F
25	Downtown Toronto	TRT	Café	Bookstore	Coffee Shop	Bakery	Restaurant	Japanese Restaurant	
26	Downtown Toronto	TRT	Café	Bar	Vegetarian / Vegan Restaurant	Bakery	Coffee Shop	Chinese Restaurant	\
30	Downtown Toronto	TRT	Grocery Store	Café	Park	Italian Restaurant	Coffee Shop	Restaurant	
37	East Toronto	TRT	Yoga Studio	Garden Center	Pizza Place	Restaurant	Farmers Market	Burrito Place	

In [114]:

```
merged.loc[merged['Cluster Labels'] == 2, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[114]:

	Borough	City	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	8t C
22	Central Toronto	TRT	Garden	Yoga Studio	Food Court	Flower Shop	Flea Market	Fish Market	Filipino Restaurant	Fa Res

In [115]:

```
merged.loc[merged['Cluster Labels'] == 3, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[115]:

	Borough	City	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
28	Downtown Toronto	TRT	Coffee Shop	Restaurant	Café	Seafood Restaurant	Italian Restaurant	Beer Bar	Hotel

In [116]:
```
merged.loc[merged['Cluster Labels'] == 4, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[116]:

	Borough	City	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 Con V
24	Central Toronto	TRT	Coffee Shop	Sandwich Place	Café	Pizza Place	Park	History Museum	BBQ Joint	Cl

In [117]:
```
merged.loc[merged['Cluster Labels'] == 5, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[117]:

	Borough	City	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	(
17	Downtown Toronto	TRT	Coffee Shop	Italian Restaurant	Café	Burger Joint	Bubble Tea Shop	Bar	Spa	R

In [118]:
```
merged.loc[merged['Cluster Labels'] == 6, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[118]:

	Borough	City	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	8tl Co
27	Downtown Toronto	TRT	Boutique	Harbor / Marina	Boat or Ferry	Sculpture Garden	Yoga Studio	Event Space	Flower Shop	

In [119]:
```
merged.loc[merged['Cluster Labels'] == 7, merged.columns[[1] + list(range(5, merged.shape[1]))]]
```

Out[119]:

	Borough	City	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 Mo Comm Ven
0	Manhattan	NY	Discount Store	Coffee Shop	Ice Cream Shop	Seafood Restaurant	Sandwich Place	Steakhouse	Do Sh
22	Manhattan	NY	Bakery	Café	Chinese Restaurant	Sandwich Place	Seafood Restaurant	Salon / Barbershop	Bub Tea Sh
25	Manhattan	NY	Coffee Shop	Pizza Place	Yoga Studio	Playground	Indian Restaurant	Deli / Bodega	Mexic Restaur
28	Manhattan	NY	Park	Coffee Shop	Hotel	Wine Shop	Italian Restaurant	Gym	Bur Jo
30	Manhattan	NY	Pizza Place	Café	Coffee Shop	Cosmetics Shop	Bar	Japanese Restaurant	Bookst
11	Downtown Toronto	TRT	Coffee Shop	Restaurant	Park	Flower Shop	Bakery	Café	F
12	Downtown Toronto	TRT	Coffee Shop	Japanese Restaurant	Gay Bar	Sushi Restaurant	Restaurant	Bubble Tea Shop	Bur Jo
34	West Toronto	TRT	Café	Mexican Restaurant	Bakery	Flea Market	Sandwich Place	Fast Food Restaurant	P
35	West Toronto	TRT	Breakfast Spot	Gift Shop	Cuban Restaurant	Bank	Dog Run	Eastern European Restaurant	Cof Sh
36	West Toronto	TRT	Pizza Place	Coffee Shop	Café	Sushi Restaurant	Italian Restaurant	Gourmet Shop	Sandw Pla

Discussion section

As we can see, Manhattan neighborhoods are mostly Cluster 1 while similar neighborhoods in Toronto are to the North of downtown, and alongside the coast line. If someone is looking for a smooth transition of lifestyle between the two cities, those are the neighborhoods to keep in mind.

Also, Toronto neighborhoods appear to be more diverse than Manhattan. Someone from New York looking for exotic experience may try to check out Toronto areas labeled cluster 2, 3, 4, 5 and 6.

Conclusion section

In this project, we reviewed and explored clustering of Manhattan and Toronto Neighborhoods. Using clustering of all neighborhoods together, we discovered that some communities in Manhattan and Toronto are quite similar in terms of popular venues. There are also more styles of neighborhoods in Toronto than in Manhattan.

In [ ]: