

Applied Data Science Capstone

Segmenting and Clustering Neighborhoods in Toronto

Importing Libraries

```
In [3]: import numpy as np # Library to handle data in a vectorized manner

import pandas as pd # Library for data analysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # Library to handle JSON files

#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API Lab
from geopy.geocoders import Nominatim # convert an address into Latitude and Longitude values

import requests # Library to handle requests
from pandas.io.json import json_normalize # transform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API Lab
import folium # map rendering library

from bs4 import BeautifulSoup

print('Libraries imported.')
```

Libraries imported.

2. Download and Explore Dataset

Creating DataFrame for Postal codes of Canada

```
In [4]: url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
```

```
req = requests.get(url)

soup = BeautifulSoup(req.text, 'lxml')
```

```
In [5]: canada_postal_data = soup.tbody.find_all('tr')

column_name = []
column_data = []

for i in canada_postal_data[0].find_all('th'):
    column_name.append(i.text.replace('\n', ' ').strip())

for i in canada_postal_data[1:]:
    row = []
    for td in i.find_all('td'):
        row.append(td.text.replace('\n', ' ').strip())
    column_data.append(row)

df = pd.DataFrame(column_data, columns = column_name)
```

```
In [6]: index_name = df[df['Borough'] == "Not assigned"].index
df.drop(index_name, inplace = True)

df.replace(to_replace = 'Not assigned', value = df.Borough)
df.shape
```

Out[6]: (103, 3)

```
In [7]: df.sort_values(by = ['Postal Code'], inplace = True)
df.reset_index(inplace = True, drop = True)
df.head()
```

| | Postal Code | Borough | Neighbourhood |
|----------|--------------------|----------------|--|
| 0 | M1B | Scarborough | Malvern, Rouge |
| 1 | M1C | Scarborough | Rouge Hill, Port Union, Highland Creek |
| 2 | M1E | Scarborough | Guildwood, Morningside, West Hill |
| 3 | M1G | Scarborough | Woburn |
| 4 | M1H | Scarborough | Cedarbrae |

Getting Lat and Long

```
In [8]: import geocoder # import geocoder
...
# initialize your variable to None
lat_lng_coords = None

# loop until you get the coordinates
while(lat_lng_coords is None):
    g = geocoder.google('{}, Toronto, Ontario'.format(df.Postal_Code))
    lat_lng_coords = g.latlng

latitude = lat_lng_coords[0]
longitude = lat_lng_coords[1]
...
```

```
Out[8]: "\n# initialize your variable to None\nlat_lng_coords = None\n# loop until you get the coordinates\nwhile(lat_lng_coords is None):\n    g = geocoder.google('{}, Toronto, Ontario'.format(df.Postal_Code))\n    lat_lng_coords = g.latlng\n\nlatitude = lat_lng_coords[0]\nlongitude = lat_lng_coords[1]\n"
```

Above Code is taking More time

Directly Downloading Lat Long file.

```
In [9]:latlong = pd.read_csv('https://cocl.us/Gespatial_data')
print('File Downloaded!')
```

File Downloaded!

```
In [10]:latlong.sort_values(by = ['Postal Code'], inplace = True)
latlong.head()
```

| | Postal Code | Latitude | Longitude |
|---|-------------|-----------|------------|
| 0 | M1B | 43.806686 | -79.194353 |
| 1 | M1C | 43.784535 | -79.160497 |
| 2 | M1E | 43.763573 | -79.188711 |
| 3 | M1G | 43.770992 | -79.216917 |
| 4 | M1H | 43.773136 | -79.239476 |

```
In [11]: neighborhoods = pd.merge(df, latlong, how = 'left', on = ['Postal Code'])
```

```
neighborhoods.head()
```

Out[11]:

| | Postal Code | Borough | Neighbourhood | Latitude | Longitude |
|----------|--------------------|----------------|--|-----------------|------------------|
| 0 | M1B | Scarborough | Malvern, Rouge | 43.806686 | -79.194353 |
| 1 | M1C | Scarborough | Rouge Hill, Port Union, Highland Creek | 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 |

```
In [12]: print('The dataframe has {} boroughs and {} neighborhoods.'.format(
    len(neighborhoods['Borough'].unique()),
    neighborhoods.shape[0]
)
)
```

The dataframe has 10 boroughs and 103 neighborhoods.

Use geopy library to get the latitude and longitude values of New York City.

In order to define an instance of the geocoder, we need to define a user_agent. We will name our agent ny_explorer, as shown below.

In [13]:

```
address = 'Toronto, Ontario'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geographical coordinate of New York City are {}, {}.'.format(latitude, longitude))
```

The geographical coordinate of New York City are 43.6534817, -79.3839347.

Create a map of New York with neighborhoods superimposed on top.

In [14]:

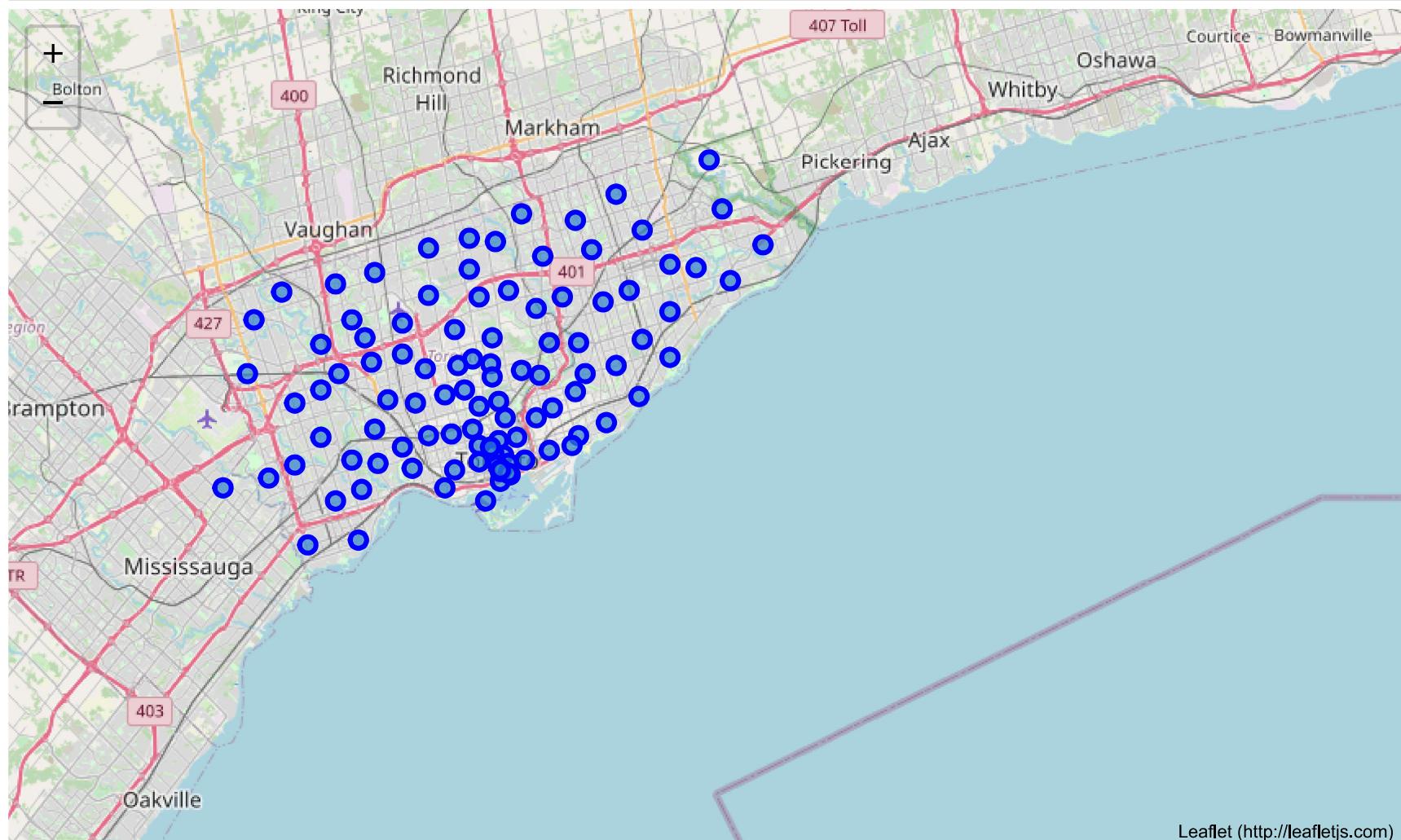
```
# create map of New York using Latitude and Longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], neighborhoods['Longitude'], neighborhoods['Borough'],
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_toronto)
```

```
map_toronto
```

Out[14]:



Leaflet (<http://leafletjs.com>)

Clustering only the neighborhoods in Toronto

```
In [15]: toronto_data = neighborhoods[neighborhoods['Borough'].str.contains('Toronto', regex=True)].reset_index(drop=True)
toronto_data.head()
```

Out[15]:

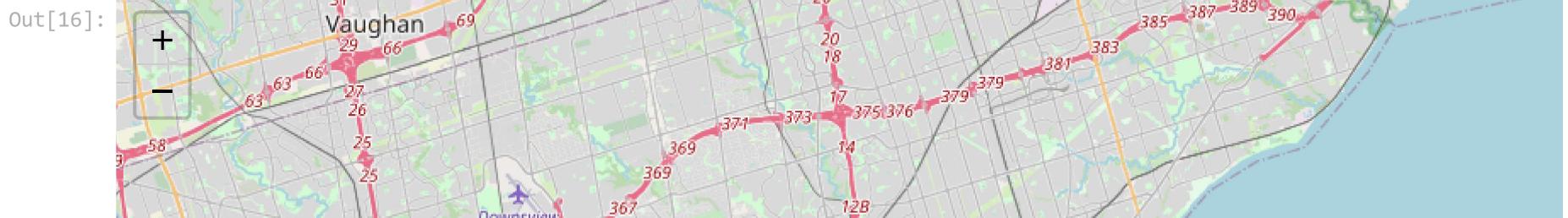
| | Postal Code | Borough | Neighbourhood | Latitude | Longitude |
|---|-------------|-----------------|--------------------------------|-----------|------------|
| 0 | M4E | East Toronto | The Beaches | 43.676357 | -79.293031 |
| 1 | M4K | East Toronto | The Danforth West, Riverdale | 43.679557 | -79.352188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

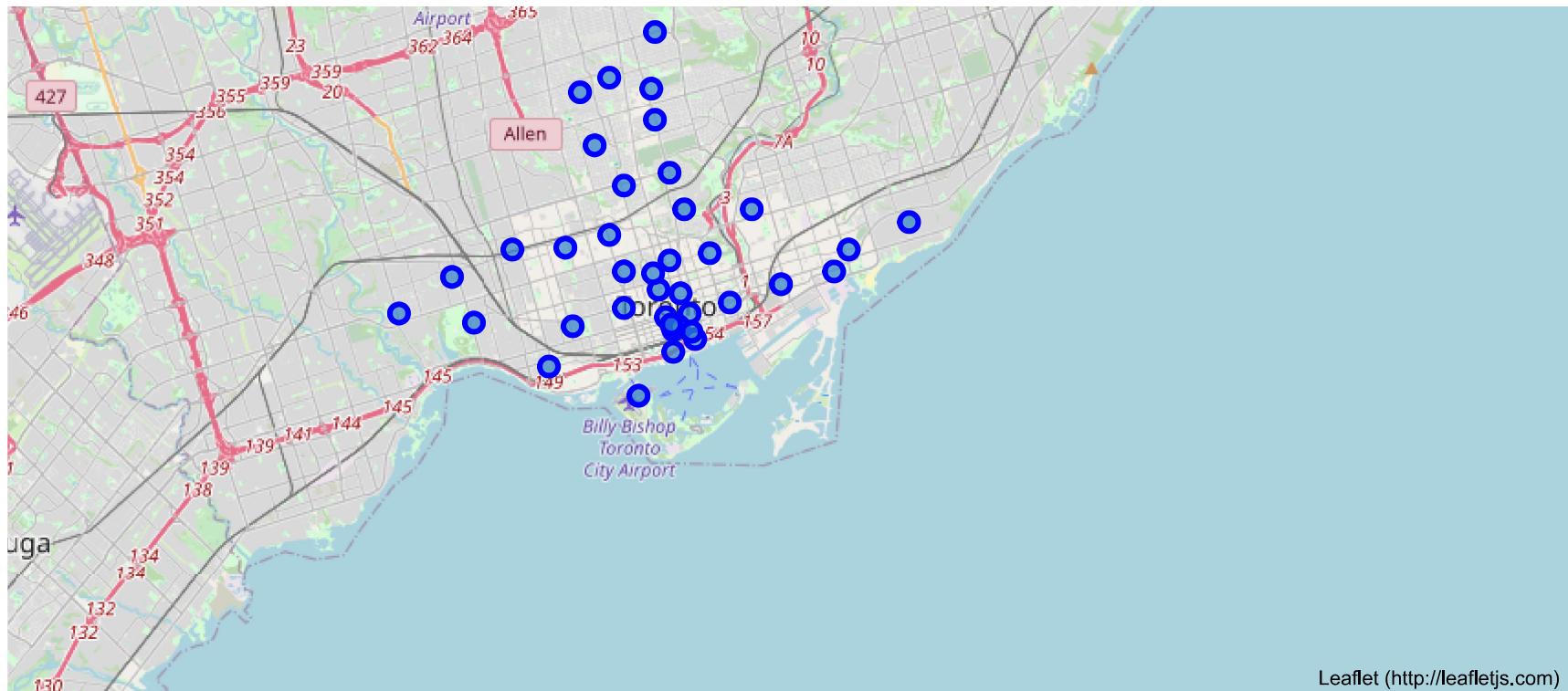
Visualize Toronto Neighbourhood in it.

```
In [16]: # create map of Toronto using Latitude and Longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(toronto_data['Latitude'], toronto_data['Longitude'], toronto_data['Neighbourhood']):
    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_toronto)

map_toronto
```





Define Foursquare Credentials and Version

```
In [17]: CLIENT_ID = 'SC3WT1HT53F1T0DBFDY0PACLF0GGSBKXA4T3005TGDLMBT5' # your Foursquare ID
CLIENT_SECRET = 'UPLFQ1W1ZL0ILTQVI0A1JV3H1JKLYFJA5G1HTPH55YB1NJ3X' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

print('Your credentials:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentials:
CLIENT_ID: SC3WT1HT53F1T0DBFDY0PACLF0GGSBKXA4T3005TGDLMBT5
CLIENT_SECRET: UPLFQ1W1ZL0ILTQVI0A1JV3H1JKLYFJA5G1HTPH55YB1NJ3X
```

2. Explore Neighborhoods in Manhattan

```
In [18]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
```

```

for name, lat, lng in zip(names, latitudes, longitudes):
    print(name)

# 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]["items"]

# 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 = ['Neighbourhood',
                        'Neighbourhood Latitude',
                        'Neighbourhood Longitude',
                        'Venue',
                        'Venue Latitude',
                        'Venue Longitude',
                        'Venue Category']

return(nearby_venues)

```

In [20]: # type your answer here

```

toronto_venues = getNearbyVenues(names=toronto_data['Neighbourhood'],
                                  latitudes=toronto_data['Latitude'],
                                  longitudes=toronto_data['Longitude']
                                 )

```

The Beaches
The Danforth West, Riverdale

India Bazaar, The Beaches West
 Studio District
 Lawrence Park
 Davisville North
 North Toronto West, Lawrence Park
 Davisville
 Moore Park, Summerhill East
 Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park
 Rosedale
 St. James Town, Cabbagetown
 Church and Wellesley
 Regent Park, Harbourfront
 Garden District, Ryerson
 St. James Town
 Berczy Park
 Central Bay Street
 Richmond, Adelaide, King
 Harbourfront East, Union Station, Toronto Islands
 Toronto Dominion Centre, Design Exchange
 Commerce Court, Victoria Hotel
 Roselawn
 Forest Hill North & West, Forest Hill Road Park
 The Annex, North Midtown, Yorkville
 University of Toronto, Harbord
 Kensington Market, Chinatown, Grange Park
 CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport
 Stn A PO Boxes
 First Canadian Place, Underground city
 Christie
 Dufferin, Dovercourt Village
 Little Portugal, Trinity
 Brockton, Parkdale Village, Exhibition Place
 High Park, The Junction South
 Parkdale, Roncesvalles
 Runnymede, Swansea
 Queen's Park, Ontario Provincial Government
 Business reply mail Processing Centre, South Central Letter Processing Plant Toronto

Let's check the size of the resulting dataframe

```
In [21]: print(toronto_venues.shape)
toronto_venues.head()
```

(1624, 7)

Out[21]:

| | Neighbourhood | Neighbourhood Latitude | Neighbourhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---------------|------------------------|-------------------------|-------------------|----------------|-----------------|----------------|
| 0 | The Beaches | 43.676357 | -79.293031 | Glen Manor Ravine | 43.676821 | -79.293942 | Trail |

| | Neighbourhood | Neighbourhood Latitude | Neighbourhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|------------------------------|------------------------|-------------------------|------------------------------------|----------------|-----------------|-------------------|
| 1 | The Beaches | 43.676357 | -79.293031 | The Big Carrot Natural Food Market | 43.678879 | -79.297734 | Health Food Store |
| 2 | The Beaches | 43.676357 | -79.293031 | Grover Pub and Grub | 43.679181 | -79.297215 | Pub |
| 3 | The Beaches | 43.676357 | -79.293031 | Upper Beaches | 43.680563 | -79.292869 | Neighborhood |
| 4 | The Danforth West, Riverdale | 43.679557 | -79.352188 | Pantheon | 43.677621 | -79.351434 | Greek Restaurant |

In [22]: `toronto_venues.groupby('Neighbourhood').count()`

Out[22]:

| Neighbourhood | Neighbourhood Latitude | Neighbourhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|--|------------------------|-------------------------|-------|----------------|-----------------|----------------|
| Berczy Park | 55 | 55 | 55 | 55 | 55 | 55 |
| Brockton, Parkdale Village, Exhibition Place | 23 | 23 | 23 | 23 | 23 | 23 |
| Business reply mail Processing Centre, South Central Letter Processing Plant Toronto | 16 | 16 | 16 | 16 | 16 | 16 |
| CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport | 16 | 16 | 16 | 16 | 16 | 16 |
| Central Bay Street | 68 | 68 | 68 | 68 | 68 | 68 |
| Christie | 16 | 16 | 16 | 16 | 16 | 16 |
| Church and Wellesley | 75 | 75 | 75 | 75 | 75 | 75 |
| Commerce Court, Victoria Hotel | 100 | 100 | 100 | 100 | 100 | 100 |
| Davisville | 33 | 33 | 33 | 33 | 33 | 33 |
| Davisville North | 9 | 9 | 9 | 9 | 9 | 9 |
| Dufferin, Dovercourt Village | 13 | 13 | 13 | 13 | 13 | 13 |
| First Canadian Place, Underground city | 100 | 100 | 100 | 100 | 100 | 100 |
| Forest Hill North & West, Forest Hill Road Park | 4 | 4 | 4 | 4 | 4 | 4 |

| Neighbourhood | Neighbourhood Latitude | Neighbourhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|--|------------------------|-------------------------|-------|----------------|-----------------|----------------|
| Neighbourhood | | | | | | |
| Garden District, Ryerson | 100 | 100 | 100 | 100 | 100 | 100 |
| Harbourfront East, Union Station, Toronto Islands | 100 | 100 | 100 | 100 | 100 | 100 |
| High Park, The Junction South | 25 | 25 | 25 | 25 | 25 | 25 |
| India Bazaar, The Beaches West | 19 | 19 | 19 | 19 | 19 | 19 |
| Kensington Market, Chinatown, Grange Park | 74 | 74 | 74 | 74 | 74 | 74 |
| Lawrence Park | 3 | 3 | 3 | 3 | 3 | 3 |
| Little Portugal, Trinity | 45 | 45 | 45 | 45 | 45 | 45 |
| Moore Park, Summerhill East | 2 | 2 | 2 | 2 | 2 | 2 |
| North Toronto West, Lawrence Park | 18 | 18 | 18 | 18 | 18 | 18 |
| Parkdale, Roncesvalles | 14 | 14 | 14 | 14 | 14 | 14 |
| Queen's Park, Ontario Provincial Government | 33 | 33 | 33 | 33 | 33 | 33 |
| Regent Park, Harbourfront | 44 | 44 | 44 | 44 | 44 | 44 |
| Richmond, Adelaide, King | 100 | 100 | 100 | 100 | 100 | 100 |
| Rosedale | 4 | 4 | 4 | 4 | 4 | 4 |
| Roselawn | 2 | 2 | 2 | 2 | 2 | 2 |
| Runnymede, Swansea | 33 | 33 | 33 | 33 | 33 | 33 |
| St. James Town | 85 | 85 | 85 | 85 | 85 | 85 |
| St. James Town, Cabbagetown | 48 | 48 | 48 | 48 | 48 | 48 |
| Stn A PO Boxes | 96 | 96 | 96 | 96 | 96 | 96 |
| Studio District | 37 | 37 | 37 | 37 | 37 | 37 |
| Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park | 14 | 14 | 14 | 14 | 14 | 14 |
| The Annex, North Midtown, Yorkville | 19 | 19 | 19 | 19 | 19 | 19 |
| The Beaches | 4 | 4 | 4 | 4 | 4 | 4 |

| | Neighbourhood | Neighbourhood Latitude | Neighbourhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue | Venue Category |
|--|--|---------------------------|----------------------------|-------|-------------------|--------------------|-------|-------------------|
| | Neighbourhood | | | | | | | |
| | The Danforth West, Riverdale | 43 | | | | | | |
| | Toronto Dominion Centre, Design Exchange | 100 | | | | | | |
| | University of Toronto, Harbord | 34 | | | | | | |

3. Analyze Each Neighborhood

In [23]:

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

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

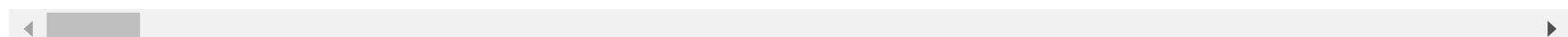
# move neighborhood column to the first column
fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]

print(toronto_onehot.shape)
toronto_onehot.head()
```

(1624, 237)

Out[23]:

| | Neighbourhood | Afghan Restaurant | Airport | Airport Food Court | Airport Gate | Airport Lounge | Airport Service | Airport Terminal | American Restaurant | Antique Shop | Aquarium | Art Gallery | Art Museum | Art & Craft Stor |
|---|------------------------------|-------------------|---------|--------------------|--------------|----------------|-----------------|------------------|---------------------|--------------|----------|-------------|------------|------------------|
| 0 | The Beaches | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | The Beaches | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | The Beaches | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | The Beaches | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | The Danforth West, Riverdale | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |



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

```
In [24]: toronto_grouped = toronto_onehot.groupby('Neighbourhood').mean().reset_index()
print(toronto_grouped.shape)
toronto_grouped.head()
```

(39, 237)

Out[24]:

| | Neighbourhood | Afghan Restaurant | Airport | Airport Food Court | Airport Gate | Airport Lounge | Airport Service | Airport Terminal | American Restaurant | Antique Shop | Aquarium | Art Gallery | Art Museum | Art Craft Sto |
|---|---|-------------------|---------|--------------------|--------------|----------------|-----------------|------------------|---------------------|--------------|----------|-------------|------------|---------------|
| 0 | Berczy Park | 0.0 | 0.0000 | 0.0000 | 0.0000 | 0.000 | 0.000 | 0.0000 | 0.0 | 0.0 | 0.0 | 0.018182 | 0.000000 | C |
| 1 | Brockton, Parkdale Village, Exhibition Place | 0.0 | 0.0000 | 0.0000 | 0.0000 | 0.000 | 0.000 | 0.0000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.000000 | C |
| 2 | Business reply mail Processing Centre, South C... | 0.0 | 0.0000 | 0.0000 | 0.0000 | 0.000 | 0.000 | 0.0000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.000000 | C |
| 3 | CN Tower, King and Spadina, Railway Lands, Har... | 0.0 | 0.0625 | 0.0625 | 0.0625 | 0.125 | 0.125 | 0.0625 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.000000 | C |
| 4 | Central Bay Street | 0.0 | 0.0000 | 0.0000 | 0.0000 | 0.000 | 0.000 | 0.0000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.014706 | C |

Let's print each neighborhood along with the top 5 most common venues

```
In [25]: num_top_venues = 5

for hood in toronto_grouped['Neighbourhood']:
    print("----"+hood+"----")
    temp = toronto_grouped[toronto_grouped['Neighbourhood'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
```

```
print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
print('\n')
```

----Berczy Park----

| | venue | freq |
|---|--------------------|------|
| 0 | Coffee Shop | 0.09 |
| 1 | Cheese Shop | 0.04 |
| 2 | Farmers Market | 0.04 |
| 3 | Bakery | 0.04 |
| 4 | Seafood Restaurant | 0.04 |

----Brockton, Parkdale Village, Exhibition Place----

| | venue | freq |
|---|-----------------------|------|
| 0 | Café | 0.13 |
| 1 | Breakfast Spot | 0.09 |
| 2 | Coffee Shop | 0.09 |
| 3 | Nightclub | 0.09 |
| 4 | Performing Arts Venue | 0.04 |

----Business reply mail Processing Centre, South Central Letter Processing Plant Toronto----

| | venue | freq |
|---|----------------------|------|
| 0 | Gym / Fitness Center | 0.06 |
| 1 | Auto Workshop | 0.06 |
| 2 | Garden Center | 0.06 |
| 3 | Garden | 0.06 |
| 4 | Light Rail Station | 0.06 |

----CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport----

| | venue | freq |
|---|---------------------|------|
| 0 | Airport Lounge | 0.12 |
| 1 | Airport Service | 0.12 |
| 2 | Rental Car Location | 0.06 |
| 3 | Harbor / Marina | 0.06 |
| 4 | Boutique | 0.06 |

----Central Bay Street----

| | venue | freq |
|---|--------------------|------|
| 0 | Coffee Shop | 0.18 |
| 1 | Café | 0.06 |
| 2 | Sandwich Place | 0.04 |
| 3 | Italian Restaurant | 0.04 |
| 4 | Thai Restaurant | 0.03 |

----Christie----

| | venue | freq |
|---|---------------|------|
| 0 | Grocery Store | 0.25 |
| 1 | Café | 0.19 |
| 2 | Park | 0.12 |
| 3 | Restaurant | 0.06 |
| 4 | Nightclub | 0.06 |

----Church and Wellesley----

| | venue | freq |
|---|---------------------|------|
| 0 | Coffee Shop | 0.09 |
| 1 | Japanese Restaurant | 0.05 |
| 2 | Sushi Restaurant | 0.05 |
| 3 | Gay Bar | 0.05 |
| 4 | Restaurant | 0.04 |

----Commerce Court, Victoria Hotel----

| | venue | freq |
|---|-------------|------|
| 0 | Coffee Shop | 0.13 |
| 1 | Restaurant | 0.07 |
| 2 | Café | 0.06 |
| 3 | Hotel | 0.06 |
| 4 | Gym | 0.04 |

----Davisville----

| | venue | freq |
|---|----------------|------|
| 0 | Sandwich Place | 0.09 |
| 1 | Pizza Place | 0.09 |
| 2 | Dessert Shop | 0.09 |
| 3 | Café | 0.06 |
| 4 | Gym | 0.06 |

----Davisville North----

| | venue | freq |
|---|----------------------|------|
| 0 | Gym / Fitness Center | 0.11 |
| 1 | Food & Drink Shop | 0.11 |
| 2 | Dance Studio | 0.11 |
| 3 | Department Store | 0.11 |
| 4 | Dog Run | 0.11 |

----Dufferin, Dovercourt Village----

| | venue | freq |
|---|----------|------|
| 0 | Pharmacy | 0.15 |
| 1 | Bakery | 0.15 |
| 2 | Brewery | 0.08 |

```
3      Bar  0.08
4  Supermarket  0.08
```

----First Canadian Place, Underground city----

| | venue | freq |
|---|-------------|------|
| 0 | Coffee Shop | 0.12 |
| 1 | Café | 0.07 |
| 2 | Restaurant | 0.04 |
| 3 | Hotel | 0.04 |
| 4 | Gym | 0.04 |

----Forest Hill North & West, Forest Hill Road Park----

| | venue | freq |
|---|---------------------|------|
| 0 | Park | 0.25 |
| 1 | Jewelry Store | 0.25 |
| 2 | Trail | 0.25 |
| 3 | Sushi Restaurant | 0.25 |
| 4 | Monument / Landmark | 0.00 |

----Garden District, Ryerson----

| | venue | freq |
|---|---------------------|------|
| 0 | Clothing Store | 0.09 |
| 1 | Coffee Shop | 0.09 |
| 2 | Café | 0.04 |
| 3 | Cosmetics Shop | 0.03 |
| 4 | Japanese Restaurant | 0.03 |

----Harbourfront East, Union Station, Toronto Islands----

| | venue | freq |
|---|---------------------|------|
| 0 | Coffee Shop | 0.13 |
| 1 | Aquarium | 0.05 |
| 2 | Hotel | 0.04 |
| 3 | Café | 0.04 |
| 4 | Fried Chicken Joint | 0.03 |

----High Park, The Junction South----

| | venue | freq |
|---|---------------------|------|
| 0 | Mexican Restaurant | 0.08 |
| 1 | Café | 0.08 |
| 2 | Thai Restaurant | 0.08 |
| 3 | Bar | 0.08 |
| 4 | Fried Chicken Joint | 0.04 |

----India Bazaar, The Beaches West----

| | venue | freq |
|---|--------------------|------|
| 0 | Park | 0.11 |
| 1 | Italian Restaurant | 0.05 |
| 2 | Pet Store | 0.05 |
| 3 | Coffee Shop | 0.05 |
| 4 | Pub | 0.05 |

----Kensington Market, Chinatown, Grange Park----

| | venue | freq |
|---|-------------------------------|------|
| 0 | Mexican Restaurant | 0.05 |
| 1 | Bar | 0.05 |
| 2 | Coffee Shop | 0.05 |
| 3 | Vegetarian / Vegan Restaurant | 0.05 |
| 4 | Café | 0.05 |

----Lawrence Park----

| | venue | freq |
|---|---------------------|------|
| 0 | Park | 0.33 |
| 1 | Bus Line | 0.33 |
| 2 | Swim School | 0.33 |
| 3 | Afghan Restaurant | 0.00 |
| 4 | Monument / Landmark | 0.00 |

----Little Portugal, Trinity----

| | venue | freq |
|---|-----------------------|------|
| 0 | Bar | 0.09 |
| 1 | Coffee Shop | 0.07 |
| 2 | Restaurant | 0.04 |
| 3 | Asian Restaurant | 0.04 |
| 4 | Vietnamese Restaurant | 0.04 |

----Moore Park, Summerhill East----

| | venue | freq |
|---|---------------------|------|
| 0 | Playground | 0.5 |
| 1 | Trail | 0.5 |
| 2 | Afghan Restaurant | 0.0 |
| 3 | Moroccan Restaurant | 0.0 |
| 4 | Malay Restaurant | 0.0 |

----North Toronto West, Lawrence Park----

| | venue | freq |
|---|----------------|------|
| 0 | Clothing Store | 0.11 |
| 1 | Coffee Shop | 0.11 |

```
2     Yoga Studio  0.06
3         Diner   0.06
4     Restaurant  0.06
```

----Parkdale, Roncesvalles----

| | venue | freq |
|---|-----------------------------|------|
| 0 | Breakfast Spot | 0.14 |
| 1 | Gift Shop | 0.14 |
| 2 | Eastern European Restaurant | 0.07 |
| 3 | Bar | 0.07 |
| 4 | Movie Theater | 0.07 |

----Queen's Park, Ontario Provincial Government----

| | venue | freq |
|---|-----------------------|------|
| 0 | Coffee Shop | 0.24 |
| 1 | Yoga Studio | 0.03 |
| 2 | Portuguese Restaurant | 0.03 |
| 3 | Smoothie Shop | 0.03 |
| 4 | Café | 0.03 |

----Regent Park, Harbourfront----

| | venue | freq |
|---|----------------|------|
| 0 | Coffee Shop | 0.18 |
| 1 | Pub | 0.07 |
| 2 | Bakery | 0.07 |
| 3 | Park | 0.07 |
| 4 | Breakfast Spot | 0.05 |

----Richmond, Adelaide, King----

| | venue | freq |
|---|-------------|------|
| 0 | Coffee Shop | 0.08 |
| 1 | Café | 0.05 |
| 2 | Hotel | 0.04 |
| 3 | Gym | 0.04 |
| 4 | Restaurant | 0.04 |

----Rosedale----

| | venue | freq |
|---|---------------------|------|
| 0 | Park | 0.50 |
| 1 | Playground | 0.25 |
| 2 | Trail | 0.25 |
| 3 | Moroccan Restaurant | 0.00 |
| 4 | Malay Restaurant | 0.00 |

----Roselawn----

| | venue | freq |
|---|---------------------|------|
| 0 | Music Venue | 0.5 |
| 1 | Garden | 0.5 |
| 2 | Afghan Restaurant | 0.0 |
| 3 | Moroccan Restaurant | 0.0 |
| 4 | Malay Restaurant | 0.0 |

----Runnymede, Swansea----

| | venue | freq |
|---|--------------------|------|
| 0 | Coffee Shop | 0.09 |
| 1 | Café | 0.09 |
| 2 | Sushi Restaurant | 0.06 |
| 3 | Pizza Place | 0.06 |
| 4 | Italian Restaurant | 0.06 |

----St. James Town----

| | venue | freq |
|---|--------------|------|
| 0 | Coffee Shop | 0.07 |
| 1 | Café | 0.06 |
| 2 | Cocktail Bar | 0.05 |
| 3 | Restaurant | 0.05 |
| 4 | Gastropub | 0.04 |

----St. James Town, Cabbagetown----

| | venue | freq |
|---|--------------------|------|
| 0 | Coffee Shop | 0.08 |
| 1 | Pizza Place | 0.06 |
| 2 | Restaurant | 0.06 |
| 3 | Café | 0.06 |
| 4 | Chinese Restaurant | 0.04 |

----Stn A PO Boxes----

| | venue | freq |
|---|--------------------|------|
| 0 | Coffee Shop | 0.10 |
| 1 | Italian Restaurant | 0.04 |
| 2 | Pub | 0.03 |
| 3 | Café | 0.03 |
| 4 | Seafood Restaurant | 0.03 |

----Studio District----

| | venue | freq |
|---|-------------|------|
| 0 | Coffee Shop | 0.08 |

```

1      Brewery  0.05
2      Café    0.05
3 American Restaurant 0.05
4      Bakery  0.05

```

----Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park----

| | venue | freq |
|---|---------------------|------|
| 0 | Coffee Shop | 0.14 |
| 1 | Light Rail Station | 0.07 |
| 2 | Liquor Store | 0.07 |
| 3 | Fried Chicken Joint | 0.07 |
| 4 | Restaurant | 0.07 |

----The Annex, North Midtown, Yorkville----

| | venue | freq |
|---|----------------|------|
| 0 | Café | 0.16 |
| 1 | Sandwich Place | 0.16 |
| 2 | Coffee Shop | 0.11 |
| 3 | Park | 0.05 |
| 4 | History Museum | 0.05 |

----The Beaches----

| | venue | freq |
|---|-------------------|------|
| 0 | Trail | 0.25 |
| 1 | Health Food Store | 0.25 |
| 2 | Neighborhood | 0.25 |
| 3 | Pub | 0.25 |
| 4 | Men's Store | 0.00 |

----The Danforth West, Riverdale----

| | venue | freq |
|---|------------------------|------|
| 0 | Greek Restaurant | 0.16 |
| 1 | Coffee Shop | 0.09 |
| 2 | Italian Restaurant | 0.07 |
| 3 | Furniture / Home Store | 0.05 |
| 4 | Ice Cream Shop | 0.05 |

----Toronto Dominion Centre, Design Exchange----

| | venue | freq |
|---|-------------|------|
| 0 | Coffee Shop | 0.14 |
| 1 | Hotel | 0.08 |
| 2 | Café | 0.05 |
| 3 | Restaurant | 0.05 |
| 4 | Salad Place | 0.03 |

```
----University of Toronto, Harbord----
      venue    freq
0        Café  0.15
1   Bookstore  0.09
2       Bar  0.06
3 Japanese Restaurant  0.06
4 Sandwich Place  0.06
```

Let's put that into a pandas dataframe

First, let's write a function to sort the venues in descending order.

```
In [26]: 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]
```

Now let's create the new dataframe and display the top 10 venues for each neighborhood.

```
In [27]: num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighbourhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{0}{1} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{0}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighbourhoods_venues_sorted = pd.DataFrame(columns=columns)
neighbourhoods_venues_sorted['Neighbourhood'] = toronto_grouped['Neighbourhood']

for ind in np.arange(toronto_grouped.shape[0]):
    neighbourhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(toronto_grouped.iloc[ind, :], num_top_venues)

neighbourhoods_venues_sorted.head()
```

Out[27]:

| | 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 | Berczy Park | Coffee Shop | Seafood Restaurant | Cocktail Bar | Farmers Market | Beer Bar | Restaurant | Cheese Shop | Bakery | Sandwich Place | Department Store |
| 1 | Brockton, Parkdale Village, Exhibition Place | Café | Breakfast Spot | Nightclub | Coffee Shop | Climbing Gym | Burrito Place | Restaurant | Italian Restaurant | Intersection | Bar |
| 2 | Business reply mail Processing Centre, South C... | Park | Pizza Place | Light Rail Station | Skate Park | Burrito Place | Farmers Market | Fast Food Restaurant | Butcher | Restaurant | Recording Studio |
| 3 | CN Tower, King and Spadina, Railway Lands, Har... | Airport Lounge | Airport Service | Boutique | Harbor / Marina | Boat or Ferry | Rental Car Location | Bar | Plane | Coffee Shop | Sculpture Garden |
| 4 | Central Bay Street | Coffee Shop | Café | Italian Restaurant | Sandwich Place | Salad Place | Bubble Tea Shop | Department Store | Burger Joint | Japanese Restaurant | Thai Restaurant |

4. Cluster Neighborhoods

Run k-means to cluster the neighborhood into 5 clusters.

In [28]:

```
# set number of clusters
kclusters = 5

toronto_grouped_clustering = toronto_grouped.drop('Neighbourhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(toronto_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[28]: array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

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

In [29]:

```
# add clustering Labels
neighbourhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

```
toronto_merged = toronto_data

# merge manhattan_grouped with manhattan_data to add Latitude/Longitude for each neighborhood
toronto_merged = toronto_merged.join(neighbourhoods_venues_sorted.set_index('Neighbourhood'), on='Neighbourhood')

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

Out[29]:

| | Postal Code | Borough | Neighbourhood | 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 C |
|---|-------------|-----------------|--------------------------------|-----------|------------|----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------------|--------|
| 0 | M4E | East Toronto | The Beaches | 43.676357 | -79.293031 | 3 | Neighborhood | Pub | Health Food Store | Trail | Yoga Studio | Dim Sum Restaurant | |
| 1 | M4K | East Toronto | The Danforth West, Riverdale | 43.679557 | -79.352188 | 2 | Greek Restaurant | Coffee Shop | Italian Restaurant | Restaurant | Ice Cream Shop | Bookstore | Fu |
| 2 | M4L | East Toronto | India Bazaar, The Beaches West | 43.668999 | -79.315572 | 2 | Park | Gym | Sushi Restaurant | Sandwich Place | Liquor Store | Burrito Place | Re: |
| 3 | M4M | East Toronto | Studio District | 43.659526 | -79.340923 | 2 | Coffee Shop | Café | Gastropub | American Restaurant | Brewery | Bakery | |
| 4 | M4N | Central Toronto | Lawrence Park | 43.728020 | -79.388790 | 0 | Park | Swim School | Bus Line | Escape Room | Electronics Store | Eastern European Restaurant | Di Re: |

Finally, let's visualize the resulting clusters

In [30]:

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

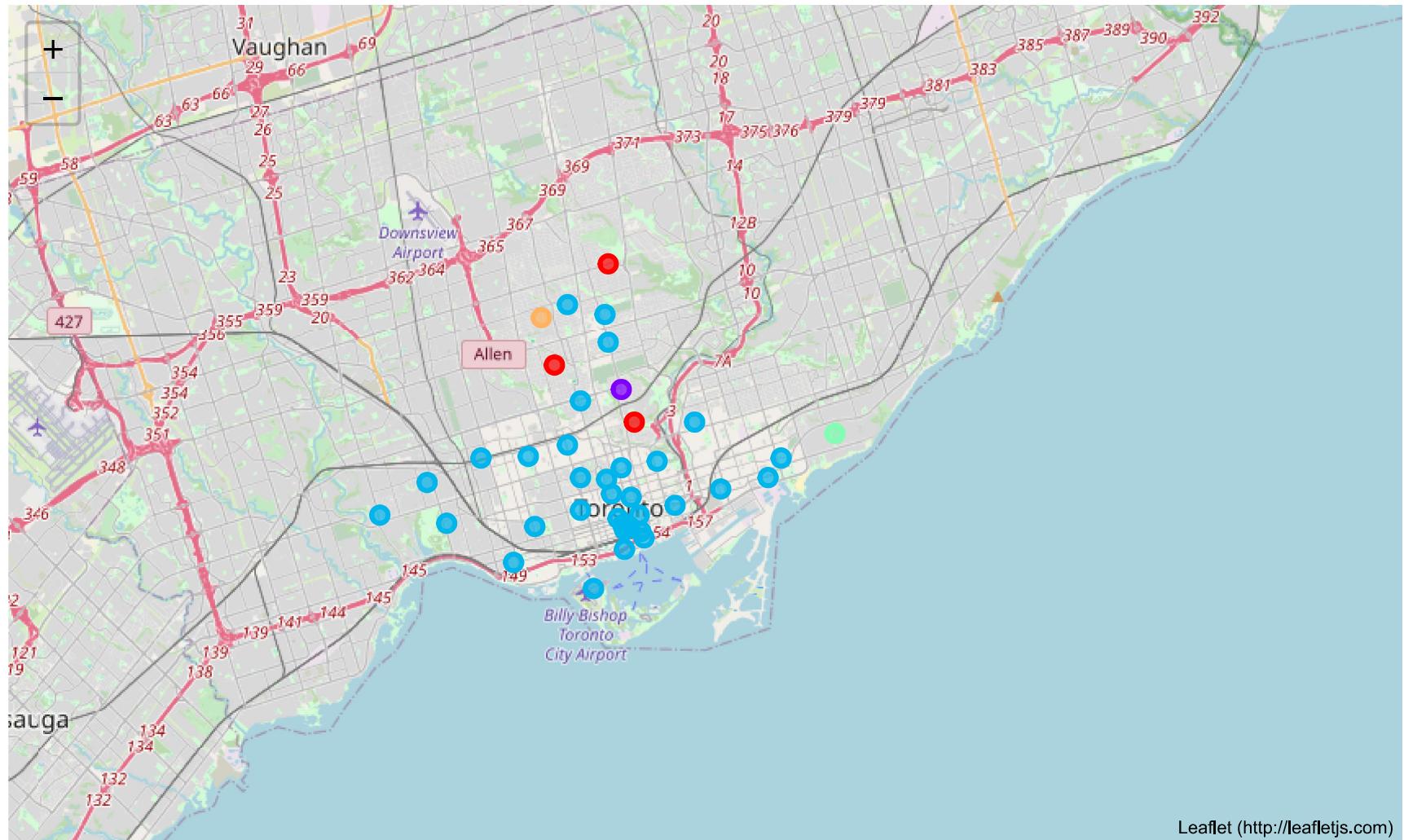
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
```

```
for lat, lon, poi, cluster in zip(toronto_merged['Latitude'], toronto_merged['Longitude'], toronto_merged['Neighbourhood']  
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(map_clusters)
```

map_clusters

Out[30]:

Leaflet (<http://leafletjs.com>)

In []: