In [1]:

```
!pip install geopy
!pip install folium
print("Installed!")
```

```
Requirement already satisfied: geopy in c:\users\dell\anaconda3\lib\site-pac
kages (1.22.0)
Requirement already satisfied: geographiclib<2,>=1.49 in c:\users\dell\anaco
nda3\lib\site-packages (from geopy) (1.50)
Requirement already satisfied: folium in c:\users\dell\anaconda3\lib\site-pa
ckages (0.11.0)
Requirement already satisfied: jinja2>=2.9 in c:\users\dell\anaconda3\lib\si
te-packages (from folium) (2.11.1)
Requirement already satisfied: branca>=0.3.0 in c:\users\dell\anaconda3\lib
\site-packages (from folium) (0.4.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-pac
kages (from folium) (1.18.1)
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-
packages (from folium) (2.22.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\dell\anaconda3\l
ib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3
\lib\site-packages (from requests->folium) (2019.11.28)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\dell\anacon
da3\lib\site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\dell\anaconda3\lib\site-packages (from requests->folium) (1.25.8)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\dell\anaconda3\lib
\site-packages (from requests->folium) (2.8)
Installed!
```

In [2]:

```
import folium
import requests
import json
import matplotlib.cm as cm
import matplotlib.colors as colors
import pandas as pd

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

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

print("Imported!")
```

```
Imported!
```

In [5]:

```
df = pd.read_csv('toronto_part2.csv')
print(df.shape)
df.head()
```

```
(180, 5)
```

Out[5]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude
0	M1A\n	Not assigned\n	NaN	43.648690	-79.385440
1	M1B\n	Scarborough\n	Malvern, Rouge	43.808626	-79.189913
2	M1C\n	Scarborough\n	Rouge Hill, Port Union, Highland Creek	43.785779	-79.157368
3	M1E\n	Scarborough\n	Guildwood, Morningside, West Hill	43.765806	-79.185284
4	M1G\n	Scarborough\n	Woburn	43.771545	-79.218135

In [7]:

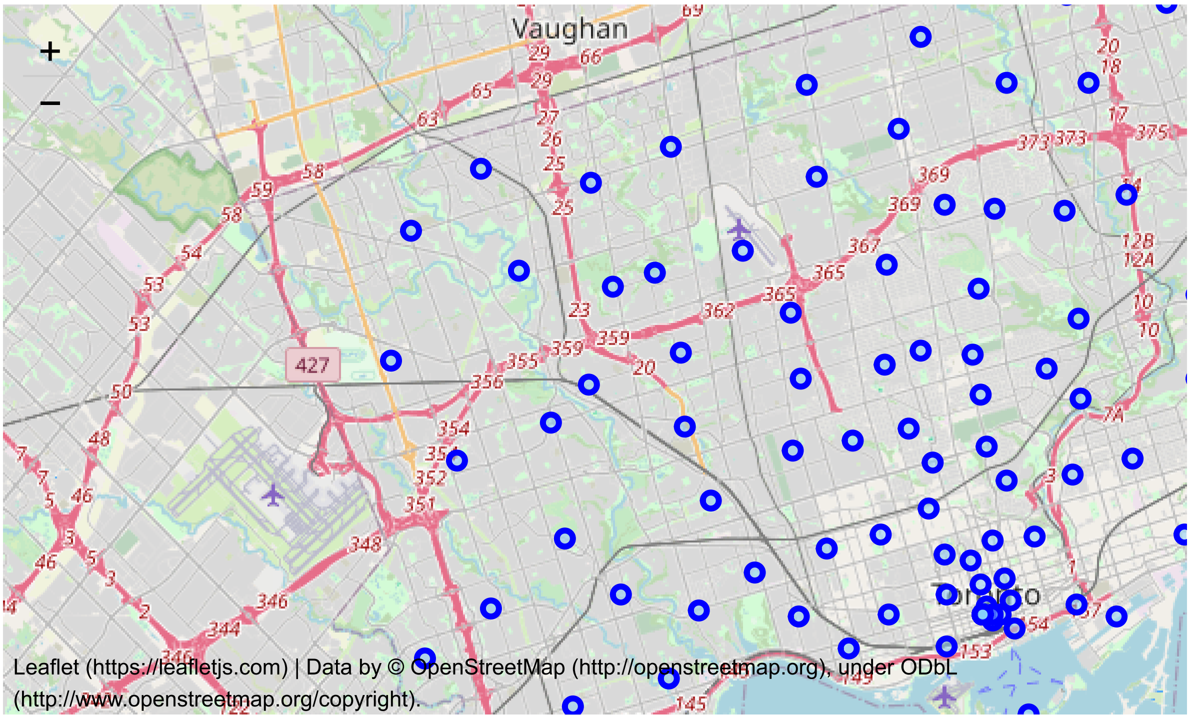
```
address = 'Toronto, Ontario Canada'
geolocator = Nominatim(user_agent="my-application")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Toronto Canada are {}, {}.'.format(latitude, longitude
```

```
The geograpical coordinate of Toronto Canada are 43.6534817, -79.3839347.
```

In [8]:

```
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=11)
for lat, lng, borough, neighborhood in zip(df['Latitude'], df['Longitude'], df['Borough'],
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=4,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#87cefa',
        fill_opacity=0.5,
        parse_html=False).add_to(map_toronto)
map_toronto
```

Out[8]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [9]:

```
toronto_data = df[df['Borough'].str.contains("Toronto")].reset_index(drop=True)
print(toronto_data.shape)
toronto_data.head()
```

```
(39, 5)
```

Out[9]:

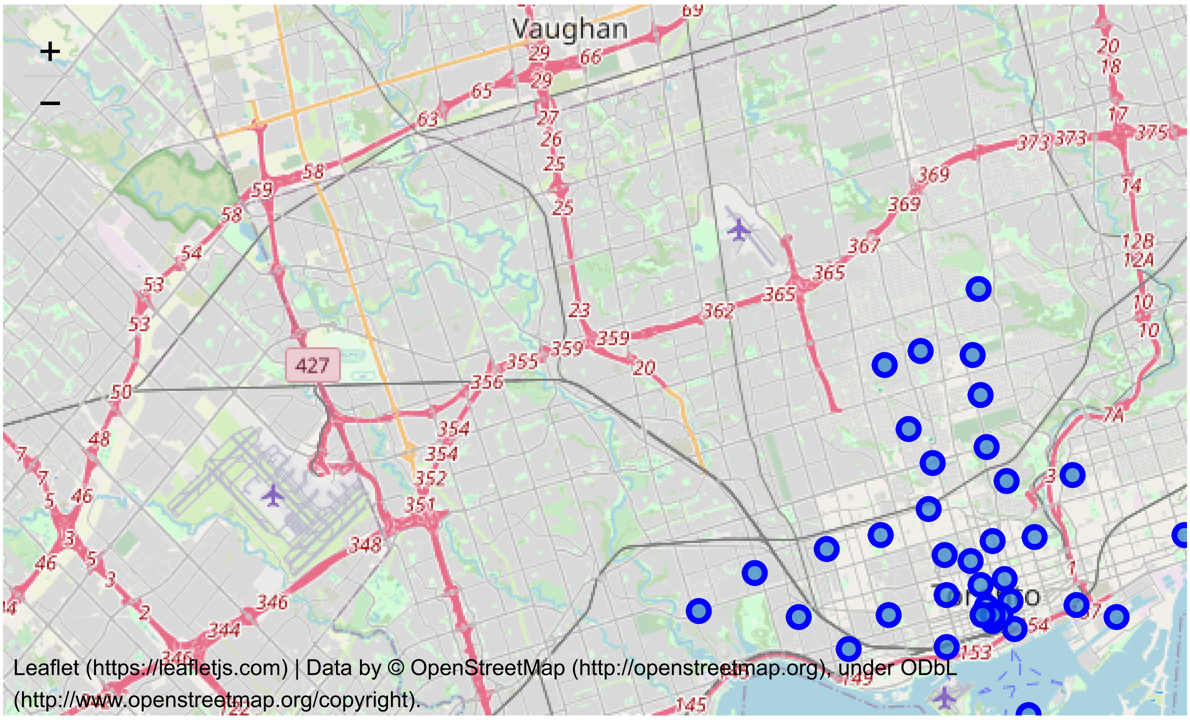
	Postalcode	Borough	Neighborhood	Latitude	Longitude
0	M4E\n	East Toronto\n	The Beaches	43.678148	-79.295349
1	M4K\n	East Toronto\n	The Danforth West, Riverdale	43.683424	-79.354564
2	M4L\n	East Toronto\n	India Bazaar, The Beaches West	43.668291	-79.315578
3	M4M\n	East Toronto\n	Studio District	43.648000	-79.339260
4	M4N\n	Central Toronto\n	Lawrence Park	43.729455	-79.386415

In [10]:

```
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=11)

for lat, lng, label in zip(toronto_data['Latitude'], toronto_data['Longitude'], toronto_dat
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker([lat, lng], radius=5, popup=label, color='blue', fill=True, fill_co
map_toronto
```

Out[10]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [11]:

```
CLIENT_ID = 'QY0AM0GF0OPQO214WT5YYRLIDUR4VMCLCIGUQ50ZWZQO0TUW' # Put Your Client Id
CLIENT_SECRET = '0E33UGPZ2IISAXO1SQIDRUQKZXKESNX0RB4RAUHUCXWYQOEP' # Put You Client Secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: Hidden')
print('CLIENT_SECRET: Hidden')
```

```
Your credentails:
CLIENT_ID: Hidden
CLIENT_SECRET: Hidden
```

In [12]:

```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&
            CLIENT_ID, CLIENT_SECRET, VERSION, lat, lng, radius, LIMIT)

        results = requests.get(url).json()["response"]['groups'][0]['items']

        venues_list.append([( name, lat, lng, v['venue']['name'], v['venue']['location']['l

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list
    nearby_venues.columns = ['Neighborhood', 'Neighborhood Latitude', 'Neighborhood Longitu

    return(nearby_venues)
```

In [13]:

```
df = toronto_data
toronto_venues = getNearbyVenues(names=df['Neighborhood'], latitudes=df['Latitude'],longitu
```

```
The Beaches
The Danforth West, Riverdale
India Bazaar, The Beaches West
Studio District
Lawrence Park
Davisville North
North Toronto West
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
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
```

In [14]:

```
print(toronto_venues.shape)
toronto_venues.head()
```

```
(789, 7)
```

Out[14]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	The Beaches	43.678148	-79.295349	Glen Manor Ravine	43.676821	-79.293942	Trail
1	The Beaches	43.678148	-79.295349	The Big Carrot Natural Food Market	43.678879	-79.297734	Health Food Store
2	The Beaches	43.678148	-79.295349	Grover Pub and Grub	43.679181	-79.297215	Pub
3	The Beaches	43.678148	-79.295349	Upper Beaches	43.680563	-79.292869	Neighborhood
4	The Beaches	43.678148	-79.295349	Calvary Baptist Church	43.681059	-79.299246	Church

In [15]:

```
toronto_venues.groupby('Neighborhood').count()
```

Out[15]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Berczy Park	30	30	30	30	30	30
Brockton, Parkdale Village, Exhibition Place	30	30	30	30	30	30
Business reply mail Processing Centre	30	30	30	30	30	30
CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport	30	30	30	30	30	30
Central Bay Street	30	30	30	30	30	30
Christie	12	12	12	12	12	12
Church and Wellesley	30	30	30	30	30	30
Commerce Court, Victoria Hotel	30	30	30	30	30	30
Davisville	25	25	25	25	25	25
Davisville North	7	7	7	7	7	7
Dufferin, Dovercourt Village	16	16	16	16	16	16
First Canadian Place, Underground city	30	30	30	30	30	30
Forest Hill North & West	3	3	3	3	3	3
Garden District, Ryerson	30	30	30	30	30	30
Harbourfront East, Union Station, Toronto Islands	4	4	4	4	4	4
High Park, The Junction South	6	6	6	6	6	6
India Bazaar, The Beaches West	22	22	22	22	22	22
Kensington Market, Chinatown, Grange Park	30	30	30	30	30	30
Lawrence Park	2	2	2	2	2	2
Little Portugal, Trinity	30	30	30	30	30	30
Moore Park, Summerhill East	4	4	4	4	4	4
North Toronto West	4	4	4	4	4	4
Parkdale, Roncesvalles	30	30	30	30	30	30
Queen's Park, Ontario Provincial Government	30	30	30	30	30	30

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Regent Park, Harbourfront	26	26	26	26	26	26
Richmond, Adelaide, King	30	30	30	30	30	30
Rosedale	4	4	4	4	4	4
Roselawn	3	3	3	3	3	3
Runnymede, Swansea	30	30	30	30	30	30
St. James Town	30	30	30	30	30	30
St. James Town, Cabbagetown	30	30	30	30	30	30
Stn A PO Boxes	30	30	30	30	30	30
Studio District	4	4	4	4	4	4
Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park	8	8	8	8	8	8
The Annex, North Midtown, Yorkville	29	29	29	29	29	29
The Beaches	5	5	5	5	5	5
The Danforth West, Riverdale	5	5	5	5	5	5
Toronto Dominion Centre, Design Exchange	30	30	30	30	30	30
University of Toronto, Harbord	30	30	30	30	30	30

In [16]:

```
print('There are {} uniques categories.'.format(len(toronto_venues['Venue Category'].unique
```

```
There are 191 uniques categories.
```

In [17]:

```
toronto_onehot = pd.get_dummies(toronto_venues[['Venue Category']], prefix="", prefix_sep="

toronto_onehot['Neighborhood'] = toronto_venues['Neighborhood']

fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]

toronto_onehot.head()
```

Out[17]:

	Yoga Studio	Accessories Store	American Restaurant	Antique Shop	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	B J
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	

In [18]:

```
toronto_onehot.shape
```

Out[18]:

```
(789, 191)
```

In [19]:

```
toronto_grouped = toronto_onehot.groupby('Neighborhood').mean().reset_index()
toronto_grouped
```

Out[19]:

	Neighborhood	Yoga Studio	Accessories Store	American Restaurant	Antique Shop	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	A &
0	Berczy Park	0.000000	0.000000	0.000000	0.000000	0.033333	0.000000	0.000000	0.000000	0
1	Brockton, Parkdale Village, Exhibition Place	0.000000	0.033333	0.000000	0.000000	0.000000	0.000000	0.033333	0.000000	0
2	Business reply mail Processing Centre	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0
	CN Tower, King and									

In [20]:

```
toronto_grouped.shape
```

Out[20]:

```
(39, 191)
```

In [21]:

```
num_top_venues = 5
for neigh in toronto_grouped['Neighborhood']:
    print("----"+neigh+"----")
    temp = toronto_grouped[toronto_grouped['Neighborhood'] == neigh].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_ven
    print('\n')
```

```
----Berczy Park----
                venue  freq
0            Beer Bar  0.07
1        Cocktail Bar  0.07
2  Seafood Restaurant  0.07
3            Creperie  0.03
4            Fountain  0.03


----Brockton, Parkdale Village, Exhibition Place----
                    venue  freq
0                    Café  0.10
1             Coffee Shop  0.07
2               Gift Shop  0.07
3  Thrift / Vintage Store  0.07
4             Supermarket  0.03


----Business reply mail Processing Centre----
```

In [22]:

```
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 [23]:

```
import numpy as np
num_top_venues = 10
indicators = ['st', 'nd', 'rd']

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))

neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = toronto_grouped['Neighborhood']

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

neighborhoods_venues_sorted.shape
```

Out[23]:

```
(39, 11)
```

In [26]:

```
from sklearn.cluster import KMeans
from sklearn.cluster import k_means_
km = KMeans(n_clusters=3, init='k-means++', max_iter=100, n_init=1,
  verbose=True)
```

In [27]:

```
kclusters = 10
toronto_grouped_clustering = toronto_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=1).fit(toronto_grouped_clustering)
print(kmeans.labels_[0:10])
print(len(kmeans.labels_))
```

```
[0 0 0 0 0 4 0 0 0 4]
39
```

In [28]:

```
df.head()
```

Out[28]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude
0	M4E\n	East Toronto\n	The Beaches	43.678148	-79.295349
1	M4K\n	East Toronto\n	The Danforth West, Riverdale	43.683424	-79.354564
2	M4L\n	East Toronto\n	India Bazaar, The Beaches West	43.668291	-79.315578
3	M4M\n	East Toronto\n	Studio District	43.648000	-79.339260
4	M4N\n	Central Toronto\n	Lawrence Park	43.729455	-79.386415

In [29]:

```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
toronto_merged = df
toronto_merged = toronto_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'),

toronto_merged.head()
```

Out[29]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue
0	M4E\n	East Toronto\n	The Beaches	43.678148	-79.295349	0	Trail	Health Food Store
1	M4K\n	East Toronto\n	The Danforth West, Riverdale	43.683424	-79.354564	4	Park	Grocery Store
2	M4L\n	East Toronto\n	India Bazaar, The Beaches West	43.668291	-79.315578	0	Park	Sandwich Place
3	M4M\n	East Toronto\n	Studio District	43.648000	-79.339260	5	Government Building	Night Market
4	M4N\n	Central Toronto\n	Lawrence Park	43.729455	-79.386415	1	Swim School	Bus Line

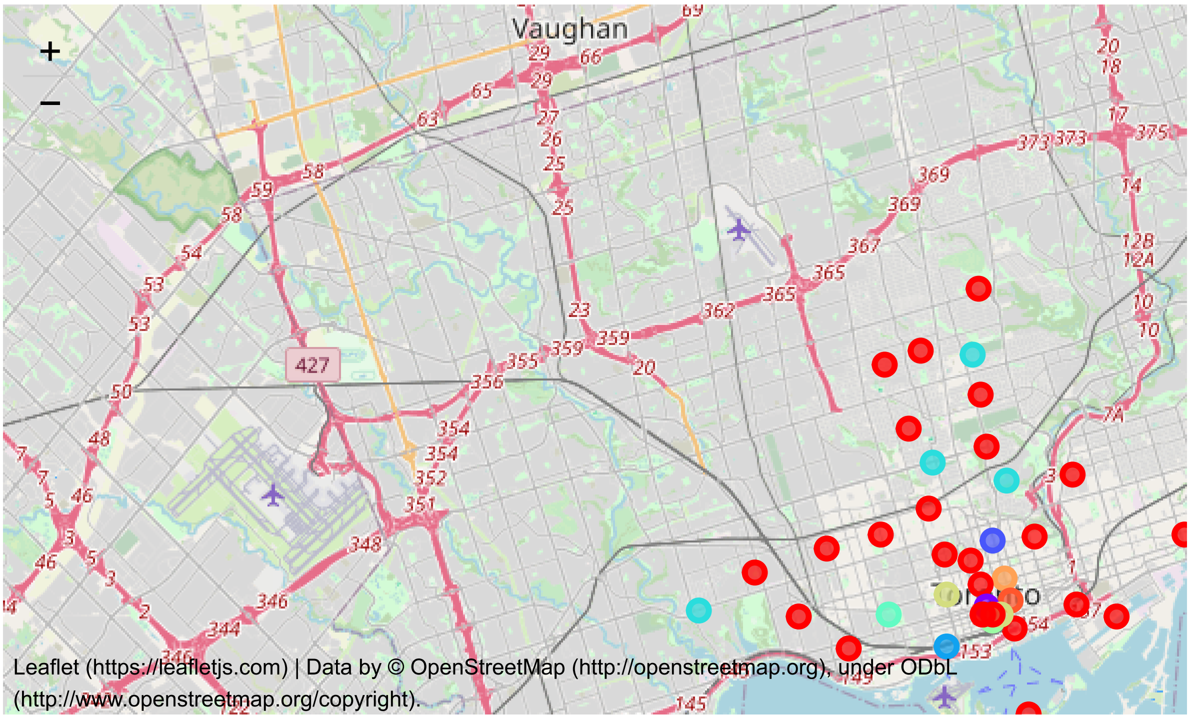
In [30]:

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

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]

markers_colors = []
for lat, lon, poi, cluster in zip(toronto_merged['Latitude'], toronto_merged['Longitude'],
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker([lat, lon], radius=5, popup=label, color=rainbow[cluster-1], fill=T
map_clusters
```

Out[30]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [ ]: