

# Capstone Project | Tha Battle of Neighborhoods | Finding the better place in Scarborough, Toronto

## 1. Installing and importing all required libraries

```
In [1]: !pip install geocoder
```

...

```
In [3]: !pip install geopy
```

Collecting geopy

Downloading <https://files.pythonhosted.org/packages/07/e1/9c72de674d5c2b8fcb0738a5ceeb5424941feffa080bfe4e240d0bacb5a38/geopy-2.0.0-py3-none-any.whl> ([http s://files.pythonhosted.org/packages/07/e1/9c72de674d5c2b8fcb0738a5ceeb5424941feffa080bfe4e240d0bacb5a38/geopy-2.0.0-py3-none-any.whl](http://s://files.pythonhosted.org/packages/07/e1/9c72de674d5c2b8fcb0738a5ceeb5424941feffa080bfe4e240d0bacb5a38/geopy-2.0.0-py3-none-any.whl)) (111kB)

Collecting geographiclib<2,>=1.49 (from geopy)

Downloading <https://files.pythonhosted.org/packages/8b/62/26ec95a98ba64299163199e95ad1b0e34ad3f4e176e221c40245f211e425/geographiclib-1.50-py3-none-any.whl> (<https://files.pythonhosted.org/packages/8b/62/26ec95a98ba64299163199e95ad1b0e34ad3f4e176e221c40245f211e425/geographiclib-1.50-py3-none-any.whl>)

Installing collected packages: geographiclib, geopy

Successfully installed geographiclib-1.50 geopy-2.0.0

In [3]: *#Importing all required libraries*

```
import pandas as pd
import numpy as np
import requests
import geocoder
import folium
import requests
import matplotlib.cm as cm
import matplotlib.colors as colors
import json
import xml
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

from pandas.io.json import json_normalize
from sklearn.cluster import KMeans
from geopy.geocoders import Nominatim
from bs4 import BeautifulSoup

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

## 2. Data Extraction and Cleaning¶

In [4]: *#Using BeautifulSoup Scraping List of Postal Codes of Given Wikipedia Page.*

```
url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
extracting_data = requests.get(url).text
wiki_data = BeautifulSoup(extracting_data, 'lxml')
```

In [5]: *#Converting content of PostalCode HTML table as dataframe*

```
column_names = ['Postalcode', 'Borough', 'Neighborhood']
toronto = pd.DataFrame(columns = column_names)

content = wiki_data.find('div', class_='mw-parser-output')
table = content.table.tbody
postcode = 0
borough = 0
neighborhood = 0

for tr in table.find_all('tr'):
    i = 0
    for td in tr.find_all('td'):
        if i == 0:
            postcode = td.text
            i = i + 1
        elif i == 1:
            borough = td.text
            i = i + 1
        elif i == 2:
            neighborhood = td.text.strip('\n').replace(']', '')
    toronto = toronto.append({'Postalcode': postcode, 'Borough': borough, 'Neighborhood': neighborhood})
```

In [6]: toronto.head()

Out[6]:

	Postalcode	Borough	Neighborhood
0	0	0	0
1	M1A\n	Not assigned\n	Not assigned
2	M2A\n	Not assigned\n	Not assigned
3	M3A\n	North York\n	Parkwoods
4	M4A\n	North York\n	Victoria Village

In [7]: toronto.replace(to\_replace=[r"\t|\\n|\\r", "\t|\\n|\\r"], value=["", ""], regex=True)  
toronto.head()

Out[7]:

	Postalcode	Borough	Neighborhood
0	0	0	0
1	M1A	Not assigned	Not assigned
2	M2A	Not assigned	Not assigned
3	M3A	North York	Parkwoods
4	M4A	North York	Victoria Village

```
In [8]: toronto = toronto[toronto.Borough != 'Not assigned']
toronto.head()
```

Out[8]:

	Postalcode	Borough	Neighborhood
0	0	0	0
3	M3A	North York	Parkwoods
4	M4A	North York	Victoria Village
5	M5A	Downtown Toronto	Regent Park, Harbourfront
6	M6A	North York	Lawrence Manor, Lawrence Heights

```
In [9]: toronto = toronto[toronto.Borough != 0]
toronto.head()
```

Out[9]:

	Postalcode	Borough	Neighborhood
3	M3A	North York	Parkwoods
4	M4A	North York	Victoria Village
5	M5A	Downtown Toronto	Regent Park, Harbourfront
6	M6A	North York	Lawrence Manor, Lawrence Heights
7	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

```
In [10]: toronto.reset_index(drop = True, inplace = True)
toronto.head()
```

Out[10]:

	Postalcode	Borough	Neighborhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Regent Park, Harbourfront
3	M6A	North York	Lawrence Manor, Lawrence Heights
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

```
In [11]: df = toronto.groupby(['Postalcode', 'Borough'])['Neighborhood'].apply(', '.join).reset_index()
df.head()
```

Out[11]:

	Postalcode	Borough	Neighborhood
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

```
In [12]: i = 0
for i in range(0,toronto.shape[0]):
    if toronto.iloc[i][2] == 'Not assigned':
        toronto.iloc[i][2] = toronto.iloc[i][1]
        i = i+1
```

```
In [13]: df = toronto.groupby(['Postalcode', 'Borough'])['Neighborhood'].apply(', '.join).reset_index()
df.head()
```

Out[13]:

	Postalcode	Borough	Neighborhood
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

```
In [14]: df.describe()
```

Out[14]:

	Postalcode	Borough	Neighborhood
count	103	103	103
unique	103	10	99
top	M1L	North York	Downsview
freq	1	24	4

```
In [15]: df = df.dropna()
empty = 'Not assigned'
df = df[(df.Postalcode != empty) & (df.Borough != empty) & (df.Neighborhood != empty)]
```

In [16]: `df.head()`

Out[16]:

	Postalcode	Borough	Neighborhood
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

```
In [17]: def neighborhood_list(grouped):
          return ', '.join(sorted(grouped['Neighborhood'].tolist()))

grp = df.groupby(['Postalcode', 'Borough'])
df_2 = grp.apply(neighborhood_list).reset_index(name='Neighborhood')
```

In [18]: `df_2.describe()`

Out[18]:

	Postalcode	Borough	Neighborhood
count	103	103	103
unique	103	10	99
top	M1L	North York	Downsview
freq	1	24	4

```
In [19]: print(df_2.shape)
df_2.head()
```

(103, 3)

Out[19]:

	Postalcode	Borough	Neighborhood
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

```
In [20]: def get_latilong(postal_code):
    lati_long_coords = None
    while(lati_long_coords is None):
        g = geocoder.arcgis('{}', Toronto, Ontario'.format(postal_code))
        lati_long_coords = g.latlng
    return lati_long_coords

get_latilong('M4G')
```

Out[20]: [43.709090000000006, -79.36409999999995]

```
In [21]: # Retrieving Postal Code Co-ordinates
postal_codes = df_2['Postalcode']
coords = [ get_latilong(postal_code) for postal_code in postal_codes.tolist() ]
```

```
In [22]: # Adding Columns Latitude & Longitude
df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
df_2['Latitude'] = df_coords['Latitude']
df_2['Longitude'] = df_coords['Longitude']
```

```
In [23]: df_2[df_2.Postalcode == 'M5G']
```

Out[23]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude
57	M5G	Downtown Toronto	Central Bay Street	43.65609	-79.38493

```
In [24]: df_2.head()
```

Out[24]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Malvern, Rouge	43.81153	-79.19552
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.78564	-79.15871
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.76575	-79.17520
3	M1G	Scarborough	Woburn	43.76820	-79.21761
4	M1H	Scarborough	Cedarbrae	43.76969	-79.23944

```
In [30]: geolocator = Nominatim(user_agent="geoapiExercises")
address = 'Scarborough,Toronto'

location = geolocator.geocode(address)
latitude_x = location.latitude
longitude_y = location.longitude
print('The Geographical Co-ordinate of Seattle,Washington are {}, {}'.format(lat:

The Geographical Co-ordinate of Seattle,Washington are 43.773077, -79.257774.
```

### 3. Map of Scarborough

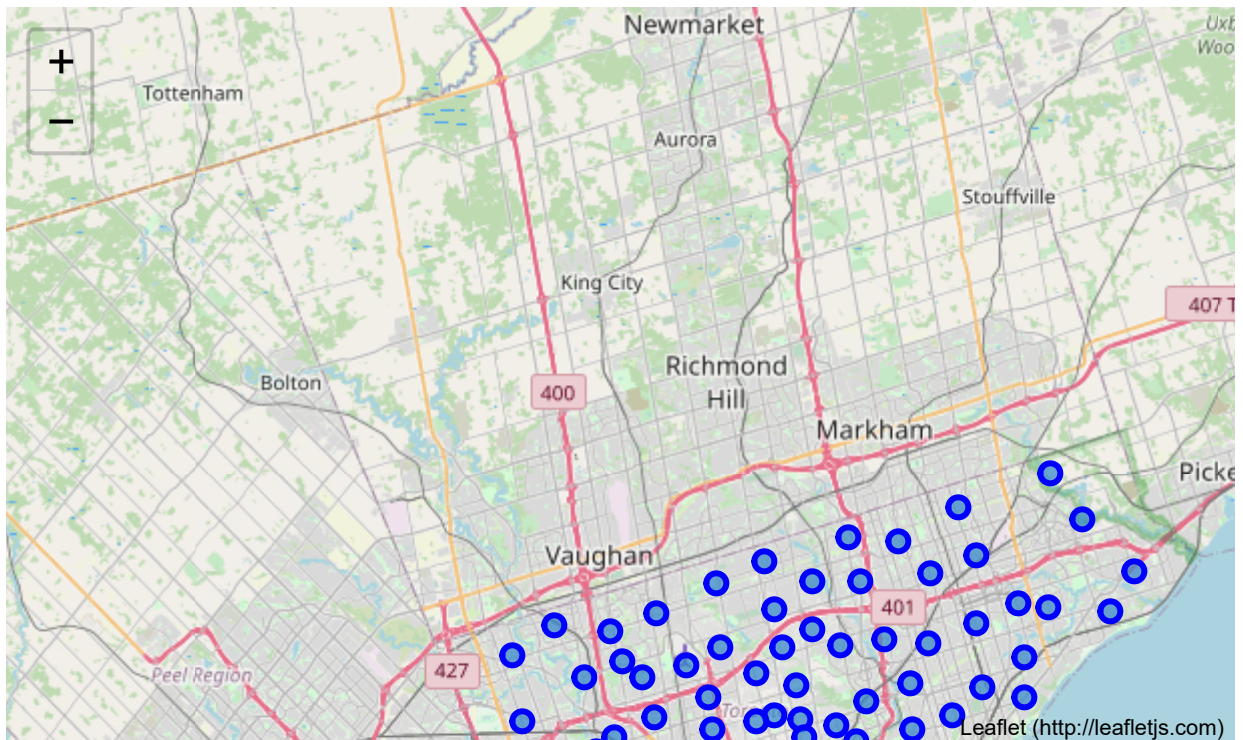
```
In [31]: map_Scarborough = folium.Map(location=[latitude_x, longitude_y], zoom_start=10)

for lat, lng, nei in zip(df_2['Latitude'], df_2['Longitude'], df_2['Neighborhood']):

    label = '{}'.format(nei)
    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_Scarborough)

map_Scarborough
```

Out[31]:



```
In [32]: address = 'Scarborough, Toronto'

geolocator = Nominatim(user_agent="geoapiExercises")
location = geolocator.geocode(address)
latitude_n1 = location.latitude
longitude_n1 = location.longitude
print('The Geographical Co-ordinate of Neighborhood_1 are {}, {}'.format(latitude_n1, longitude_n1))

The Geographical Co-ordinate of Neighborhood_1 are 43.773077, -79.257774.
```



```
In [5]: CLIENT_ID = '3WPOAAET0T5MLDOBVDAAAFVHGPODUP5XHDAYSDETPAXBXXUI' # my Foursquare ID
CLIENT_SECRET = 'LJFPQDQHVRMDPA2ZYKBNSP2BMZCGKXL2GAFEVGMTYFGPPFTL' # my Foursquare Secret
VERSION = '20180604'
LIMIT = 30
print('Got Client_ID and Client_Secret')
```

Got Client\_ID and Client\_Secret

```
In [35]: radius = 700
LIMIT = 100
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&version={}&limit={}&lat={}&lng={}&radius={}&v={}'
      CLIENT_ID,
      CLIENT_SECRET,
      VERSION,
      latitude_n1,
      longitude_n1,
      radius,
      LIMIT)
results = requests.get(url).json()
```

```
In [36]: venues=results['response']['groups'][0]['items']
nearby_venues = json_normalize(venues)
nearby_venues.columns
```

```
Out[36]: Index(['referralId', 'reasons.count', 'reasons.items', 'venue.id',
               'venue.name', 'venue.location.address', 'venue.location.crossStreet',
               'venue.location.lat', 'venue.location.lng',
               'venue.location.labeledLatLngs', 'venue.location.distance',
               'venue.location.postalCode', 'venue.location.cc', 'venue.location.city',
               'venue.location.state', 'venue.location.country',
               'venue.location.formattedAddress', 'venue.categories',
               'venue.photos.count', 'venue.photos.groups',
               'venue.location.neighborhood', 'venue.venuePage.id'],
              dtype='object')
```

```
In [37]: 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']
```

## 4. Nearby Venues/Locations

```
In [38]: filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues = nearby_venues.loc[:, filtered_columns]
nearby_venues.head()
```

Out[38]:

	venue.name	venue.categories	venue.location.lat	venue.location.lng
0	SEPHORA	[{'id': '4bf58dd8d48988d10c951735', 'name': 'C...'}]	43.775017	-79.258109
1	Disney Store	[{'id': '4bf58dd8d48988d1f3941735', 'name': 'T...'}]	43.775537	-79.256833
2	American Eagle Store	[{'id': '4bf58dd8d48988d103951735', 'name': 'C...'}]	43.776012	-79.258334
3	Shoppers Drug Mart	[{'id': '4bf58dd8d48988d10f951735', 'name': 'P...'}]	43.773305	-79.251662
4	Chipotle Mexican Grill	[{'id': '4bf58dd8d48988d1c1941735', 'name': 'M...'}]	43.776410	-79.258069

## 5. Categories of Nearby Venues/Locations

```
In [39]: nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[0] for col in nearby_venues.columns]

nearby_venues.head(5)
```

Out[39]:

	name	categories	lat	lng
0	SEPHORA	Cosmetics Shop	43.775017	-79.258109
1	Disney Store	Toy / Game Store	43.775537	-79.256833
2	American Eagle Store	Clothing Store	43.776012	-79.258334
3	Shoppers Drug Mart	Pharmacy	43.773305	-79.251662
4	Chipotle Mexican Grill	Mexican Restaurant	43.776410	-79.258069

```
In [40]: # Top 10 Categories
a=pd.Series(nearby_venues.categories)
a.value_counts()[:10]
```

```
Out[40]: Clothing Store      8
Coffee Shop      5
Restaurant      5
Sandwich Place   2
Gas Station     2
Pharmacy        2
Chocolate Shop   1
Cosmetics Shop   1
Gym             1
Mexican Restaurant 1
Name: categories, dtype: int64
```

```
In [41]: def getNearbyVenues(names, latitudes, longitudes, radius=700):

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

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

        # making GET request
        venue_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 venue_results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in
    nearby_venues.columns = ['Neighborhood',
                            'Neighborhood Latitude',
                            'Neighborhood Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)
```

```
In [42]: # Nearby Venues
Scarborough_venues = getNearbyVenues(names=df_2['Neighborhood'],
                                     latitudes=df_2['Latitude'],
                                     longitudes=df_2['Longitude']
                                     )
```

Malvern, Rouge  
Rouge Hill, Port Union, Highland Creek  
Guildwood, Morningside, West Hill  
Woburn  
Cedarbrae  
Scarborough Village  
Kennedy Park, Ionview, East Birchmount Park  
Golden Mile, Clairlea, Oakridge  
Cliffside, Cliffcrest, Scarborough Village West  
Birch Cliff, Cliffside West  
Dorset Park, Wexford Heights, Scarborough Town Centre  
Wexford, Maryvale  
Agincourt  
Clarks Corners, Tam O'Shanter, Sullivan  
Milliken, Agincourt North, Steeles East, L'Amoreaux East  
Steeles West, L'Amoreaux West  
Upper Rouge  
Hillcrest Village  
Fairview, Henry Farm, Oriole  
Bayview Village  
York Mills, Silver Hills  
Willowdale, Newtonbrook  
Willowdale, Willowdale East  
York Mills West  
Willowdale, Willowdale West  
Parkwoods  
Don Mills  
Don Mills  
Bathurst Manor, Wilson Heights, Downsview North  
Northwood Park, York University  
Downsview  
Downsview  
Downsview  
Downsview  
Victoria Village  
Parkview Hill, Woodbine Gardens  
Woodbine Heights  
The Beaches  
Leaside  
Thorncliffe Park  
East Toronto, Broadview North (Old East York)  
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  
Bedford Park, Lawrence Manor East  
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  
Lawrence Manor, Lawrence Heights  
Glencairn  
Humewood-Cedarvale  
Caledonia-Fairbanks  
Christie  
Dufferin, Dovercourt Village  
Little Portugal, Trinity  
Brockton, Parkdale Village, Exhibition Place  
North Park, Maple Leaf Park, Upwood Park  
Del Ray, Mount Dennis, Keelsdale and Silverthorn  
Runnymede, The Junction North  
High Park, The Junction South  
Parkdale, Roncesvalles  
Runnymede, Swansea  
Queen's Park, Ontario Provincial Government  
Canada Post Gateway Processing Centre  
Business reply mail Processing Centre, South Central Letter Processing Plant  
Toronto  
New Toronto, Mimico South, Humber Bay Shores  
Alderwood, Long Branch  
The Kingsway, Montgomery Road, Old Mill North  
Old Mill South, King's Mill Park, Sunnylea, Humber Bay, Mimico NE, The Queens  
way East, Royal York South East, Kingsway Park South East  
Mimico NW, The Queensway West, South of Bloor, Kingsway Park South West, Roya  
l York South West  
Islington Avenue, Humber Valley Village  
West Deane Park, Princess Gardens, Martin Grove, Islington, Cloverdale  
Eringate, Bloordale Gardens, Old Burnhamthorpe, Markland Wood  
Humber Summit  
Humberlea, Emery  
Weston  
Westmount  
Kingsview Village, St. Phillips, Martin Grove Gardens, Richview Gardens  
South Steeles, Silverstone, Humbergate, Jamestown, Mount Olive, Beaumont Heig

In [43]:

```
print('There are {} Uniques Categories.'.format(len(Scarborough_venues['Venue Category'])))
Scarborough_venues.groupby('Neighborhood').count().head()
```

There are 305 Uniques Categories.

Out[43]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Agincourt	33	33	33	33	33	33
Alderwood, Long Branch	8	8	8	8	8	8
Bathurst Manor, Wilson Heights, Downsview North	4	4	4	4	4	4
Bayview Village	4	4	4	4	4	4
Bedford Park, Lawrence Manor East	27	27	27	27	27	27

## One Hot Encoding of Features

```
In [44]: # one hot encoding
Scarborough_onehot = pd.get_dummies(Scarborough_venues[['Venue Category']], prefix=

# add neighborhood column back to dataframe
Scarborough_onehot['Neighborhood'] = Scarborough_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [Scarborough_onehot.columns[-1]] + list(Scarborough_onehot.columns[
Scarborough_onehot = Scarborough_onehot[fixed_columns]
Scarborough_grouped = Scarborough_onehot.groupby('Neighborhood').mean().reset_index()
Scarborough_onehot.head(5)
```

Out[44]:

	Zoo Exhibit	Accessories Store	Afghan Restaurant	African Restaurant	Airport	American Restaurant	Animal Shelter	Antique Shop	Aquarium
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	1	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 [45]: num_top_venues = 5
for hood in Scarborough_grouped['Neighborhood']:
    print("---- "+hood+" ----")
    temp = Scarborough_grouped[Scarborough_grouped['Neighborhood'] == hood].T.res
    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(
    print('\n')
```

```
---- Agincourt ----
```

	venue	freq
0	Chinese Restaurant	0.09
1	Shopping Mall	0.09
2	Pizza Place	0.06
3	Seafood Restaurant	0.03
4	Bakery	0.03

```
---- Alderwood, Long Branch ----
```

	venue	freq
0	Pool	0.12
1	Coffee Shop	0.12
2	Print Shop	0.12
3	Pub	0.12
4	Sandwich Place	0.12

```
---- Bathurst Manor, Wilson Heights, Downsview North ----
```

## Most Common venues in neighborhood

```
In [46]: 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 [47]: 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'] = Scarborough_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()

```

Out[47]:

	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
0	Agincourt	Chinese Restaurant	Shopping Mall	Pizza Place	Skating Rink	Pool Hall	Supermarket	Sushi Restaurant
1	Alderwood, Long Branch	Pizza Place	Print Shop	Gas Station	Coffee Shop	Gym	Pub	Sandwich Place
2	Bathurst Manor, Wilson Heights, Downsview North	Park	Convenience Store	Other Great Outdoors	Doner Restaurant	Dumpling Restaurant	Eastern European Restaurant	Electronic Store
3	Bayview Village	Dog Run	Asian Restaurant	Trail	Park	Ethiopian Restaurant	Dumpling Restaurant	Eastern European Restaurant
4	Bedford Park, Lawrence Manor East	Coffee Shop	Pizza Place	Italian Restaurant	Sandwich Place	Intersection	Butcher	Sports Club

## K-Means Clustering

```
In [48]: # Using K-Means to cluster neighborhood into 3 clusters
Scarborough_grouped_clustering = Scarborough_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=3, random_state=0).fit(Scarborough_grouped_clustering)
kmeans.labels_
```

```
Out[48]: array([2, 2, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 0, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2,
        0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0,
        2, 2, 2, 2, 2, 2, 0, 2, 2, 1])
```

```
In [49]: neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

Scarborough_merged = df_2.iloc[:16,:]

# merge toronto_grouped with toronto_data to add Latitude/Longitude for each neighborhood
Scarborough_merged = Scarborough_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'))

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

```
Out[49]:
```

	Postalcode	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue
0	M1B	Scarborough	Malvern, Rouge	43.81153	-79.19552	2	Zoo Exhibit	Financial or Legal Service
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.78564	-79.15871	2	Construction & Landscaping	Home Service
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.76575	-79.17520	0	Park	Athletics & Sports
3	M1G	Scarborough	Woburn	43.76820	-79.21761	0	Coffee Shop	Chinese Restaurant
4	M1H	Scarborough	Cedarbrae	43.76969	-79.23944	2	Thai Restaurant	Indian Restaurant

## Clusters Map

```
In [50]: kclusters = 10
```

```

In [51]: # create map
map_clusters = folium.Map(location=[latitude_x, longitude_y], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
colors_array = cm.rainbow(np.linspace(0, 1, kclusters))
rainbow = [colors.rgb2hex(i) for i in colors_array]
print(rainbow)
# add markers to the map

markers_colors = []
for lat, lon, nei, cluster in zip(Scarborough_merged['Latitude'],
                                   Scarborough_merged['Longitude'],
                                   Scarborough_merged['Neighborhood'],
                                   Scarborough_merged['Cluster Labels']):
    label = folium.Popup(str(nei) + ' 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

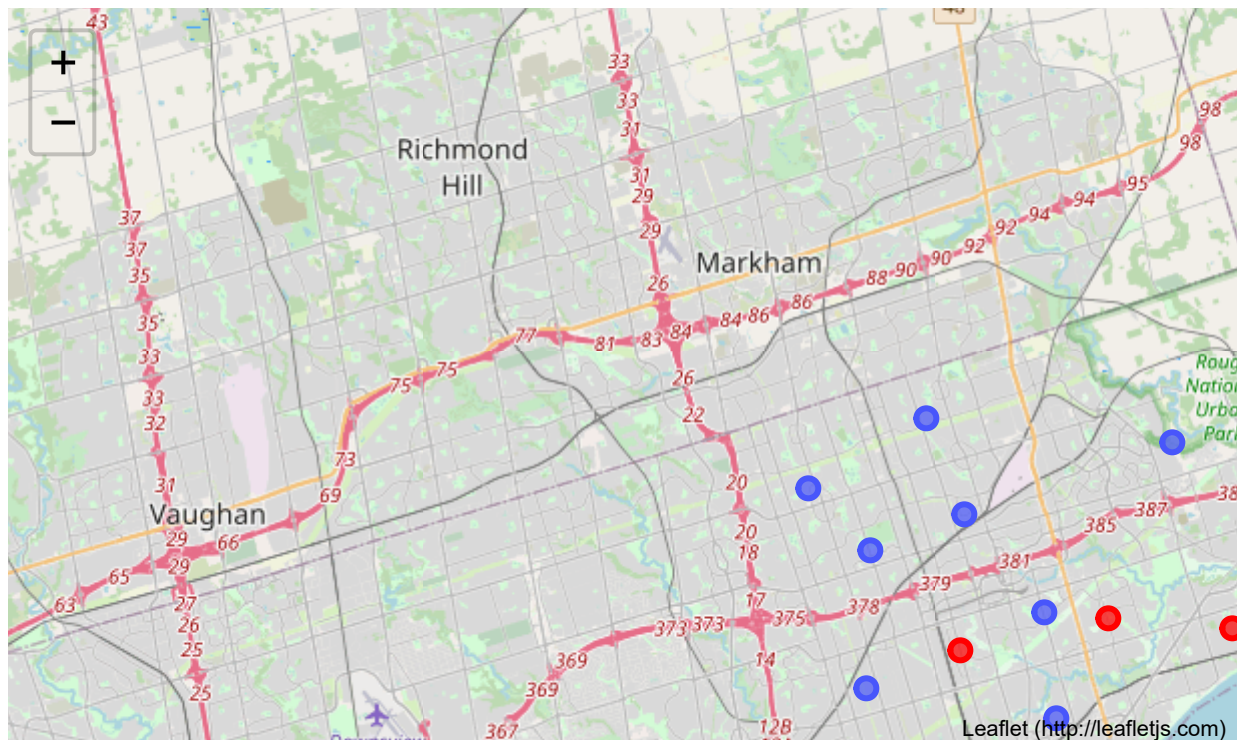
```

```

['#8000ff', '#4856fb', '#10a2f0', '#2adddd', '#62fbc4', '#9cfba4', '#d4dd80',
 '#ffa256', '#ff562c', '#ff0000']

```

Out[51]:



```
In [52]: df1=Scarborough_merged.loc[Scarborough_merged['Cluster Labels'] == 0,Scarborough_
df2=Scarborough_merged.loc[Scarborough_merged['Cluster Labels'] == 1,Scarborough_
df3=Scarborough_merged.loc[Scarborough_merged['Cluster Labels'] == 2,Scarborough_
```

```
In [53]: Scarborough_Avg_HousingPrice=pd.DataFrame({"Neighborhood":df_2["Neighborhood"],
                                                    "Average_Housing_Price":[335000.0,286600.0
                                                    573900.0,225000.0
                                                    279200.0,225000.0
                                                    435000.0,289500.0
                                                    589900.0,295000.0
                                                    229200.0,365000.0
                                                    573900.0,367000.0
                                                    279200.0,271000.0
                                                    573900.0,329000.0
                                                    279200.0,375000.0
                                                    573900.0,425000.0
                                                    279200.0,195000.0
                                                    573900.0,325000.0
                                                    279200.0,289000.0
                                                    ]})
```

```
In [54]: Scarborough_Avg_HousingPrice.set_index('Neighborhood',inplace=True,drop=True)
```

```
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x258256fe588>
```



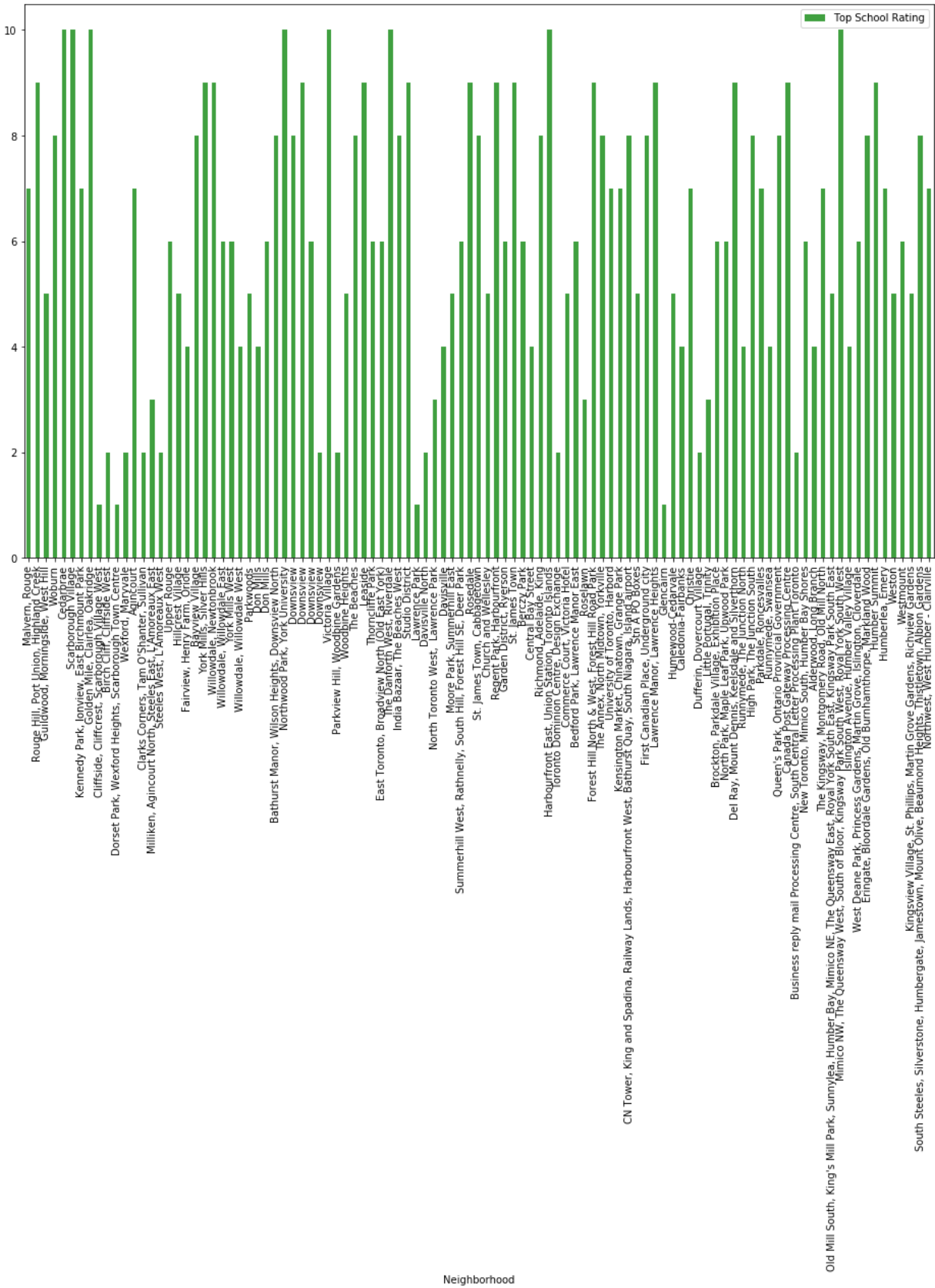
## School Rating by Clusters

```
In [56]: clusters=pd.DataFrame({"Cluster1":df1["Neighborhood"],
                                "Cluster2":df2["Neighborhood"],
                                "Cluster4":df3["Neighborhood"]})
clusters = clusters.replace(np.nan, '', regex=True)
```

```
In [57]: new_Scarborough=Scarborough_merged.set_index("Neighborhood",drop=True)
#Source:https://www.greatschools.org
Scarborough_school_ratings=pd.DataFrame({"Neighborhood":df["Neighborhood"],
                                          "Top School Rating":[7,9,5,8,10,10,7,10,1,
                                                                5,4,8,9,9,6,6,4,5,4,6,
                                                                10,2,5,8,9,6,6,10,8,9,
                                                                8,5,9,6,9,6,4,8,10,2,
                                                                7,8,5,8,9,1,5,4,7,2,3,
                                                                4,8,9,2,6,4,7,5,10,4,
                                                                ]})
```

```
In [58]: Scarborough_school_ratings.set_index('Neighborhood',inplace=True,drop=True)
```

```
In [59]: Scarborough_school_ratings.plot(kind='bar',figsize=(16,10),color='green',alpha=0.5)
```



In [ ]: