Capstone Project – The Battle of Neighborhoods | Finding a Better Place in Scarborough, Toronto

1. Installing and Importing Python Libraries and Dependencies

In [1]:

```
!pip install geocoder
!pip install folium
```

```
Requirement already satisfied: geocoder in /home/untold/anaconda3/lib/python3
.8/site-packages (1.38.1)
Requirement already satisfied: requests in /home/untold/anaconda3/lib/python3
.8/site-packages (from geocoder) (2.24.0)
Requirement already satisfied: six in /home/untold/anaconda3/lib/python3.8/si
te-packages (from geocoder) (1.15.0)
Requirement already satisfied: click in /home/untold/anaconda3/lib/python3.8/
site-packages (from geocoder) (7.1.2)
Requirement already satisfied: ratelim in /home/untold/anaconda3/lib/python3.
8/site-packages (from geocoder) (0.1.6)
Requirement already satisfied: future in /home/untold/anaconda3/lib/python3.8
/site-packages (from geocoder) (0.18.2)
Requirement already satisfied: idna<3,>=2.5 in /home/untold/anaconda3/lib/pyt
hon3.8/site-packages (from requests->geocoder) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /home/untold/anaconda3/li
b/python3.8/site-packages (from requests->geocoder) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /ho
me/untold/anaconda3/lib/python3.8/site-packages (from requests->geocoder) (1.
25.9)
Requirement already satisfied: certifi>=2017.4.17 in /home/untold/anaconda3/l
ib/python3.8/site-packages (from requests->geocoder) (2020.6.20)
Requirement already satisfied: decorator in /home/untold/anaconda3/lib/python
3.8/site-packages (from ratelim->geocoder) (4.4.2)
Requirement already satisfied: folium in /home/untold/anaconda3/lib/python3.8
/site-packages (0.12.0)
Requirement already satisfied: requests in /home/untold/anaconda3/lib/python3
.8/site-packages (from folium) (2.24.0)
Requirement already satisfied: jinja2>=2.9 in /home/untold/anaconda3/lib/pyth
on3.8/site-packages (from folium) (2.11.2)
Requirement already satisfied: branca>=0.3.0 in /home/untold/anaconda3/lib/py
thon3.8/site-packages (from folium) (0.4.2)
Requirement already satisfied: numpy in /home/untold/anaconda3/lib/python3.8/
site-packages (from folium) (1.18.5)
Requirement already satisfied: idna<3,>=2.5 in /home/untold/anaconda3/lib/pyt
hon3.8/site-packages (from requests->folium) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /home/untold/anaconda3/li
b/python3.8/site-packages (from requests->folium) (3.0.4)
```

```
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /ho
me/untold/anaconda3/lib/python3.8/site-packages (from requests->folium) (1.25
.9)
Requirement already satisfied: certifi>=2017.4.17 in /home/untold/anaconda3/l
ib/python3.8/site-packages (from requests->folium) (2020.6.20)
Requirement already satisfied: MarkupSafe>=0.23 in /home/untold/anaconda3/lib
/python3.8/site-packages (from jinja2>=2.9->folium) (1.1.1)
```

Importing Libraries

In [2]:

```
import pandas as pd
import requests
import numpy as np
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)

print("All Required Libraries Imported!")
```

```
All Required Libraries Imported!
```

2. Data Extraction and Cleaning

Using BeautifulSoup Scraping List of Postal Codes of Given Wikipedia Page. Link: https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

In [3]:

```
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')
```

Converting content of PostalCode HTML table as dataframe

In [4]:

```
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,'Neig
hborhood': neighborhood},ignore_index=True)
```

In [5]:

```
# clean dataframe
toronto = toronto[toronto.Borough!='Not assigned']
toronto = toronto[toronto.Borough!= 0]
toronto.reset_index(drop = True, inplace = True)
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 [6]:

```
df = toronto.groupby(['Postalcode','Borough'])['Neighborhood'].apply(', '.joi
n).reset_index()
df.head()
```

Out[6]:

	Postalcode	Borough	Neighborhood
0	M1A\n	Not assigned\n	Not assigned\n
1	M1B\n	Scarborough\n	Malvern, Rouge
2	M1C\n	Scarborough\n	Rouge Hill, Port Union, Highland Creek
3	M1E\n	Scarborough\n	Guildwood, Morningside, West Hill
4	M1G\n	Scarborough\n	Woburn

In [7]:

```
df.describe()
```

Out[7]:

	Postalcode	Borough	Neighborhood
count	180	180	180
unique	180	11	100
top	M2K\n	Not assigned\n	Not assigned\n
freq	1	77	77

In [8]:

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

In [9]:

```
df.head()
```

Out[9]:

	Postalcode	Borough	Neighborhood
0	M1A\n	Not assigned\n	Not assigned\n
1	M1B\n	Scarborough\n	Malvern, Rouge
2	M1C\n	Scarborough\n	Rouge Hill, Port Union, Highland Creek
3	M1E\n	Scarborough\n	Guildwood, Morningside, West Hill
4	M1G\n	Scarborough\n	Woburn

In [10]:

```
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 [11]:

```
df_2.describe()
```

Out[11]:

	Postalcode	Borough	Neighborhood
count	180	180	180
unique	180	11	100
top	M2K\n	Not assigned\n	Not assigned\n
freq	1	77	77

In [12]:

```
print(df_2.shape)
df_2.head()
```

```
(180, 3)
```

Out[12]:

	Postalcode	Borough	Neighborhood
0	M1A\n	Not assigned\n	Not assigned\n
1	M1B\n	Scarborough\n	Malvern, Rouge
2	M1C\n	Scarborough\n	Rouge Hill, Port Union , Highland Creek
3	M1E\n	Scarborough\n	Guildwood, Morningside , West Hill
4	M1G\n	Scarborough\n	Woburn

In [13]:

```
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[13]:

```
[43.709020000000066, -79.36348999999996]
```

In [14]:

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

In [15]:

```
# 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 [16]:

```
df_2[df_2.Postalcode == 'M5G']
```

Out[16]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude

In [17]:

```
df_2.head(10)
```

Out[17]:

	Postalcode	Borough	Neighborhood	Latitude	Longitude
0	M1A\n	Not assigned\n	Not assigned\n	43.64869	-79.38544
1	M1B\n	Scarborough\n	Malvern, Rouge	43.81139	-79.19662

	Postalcode	Borough	Neighborhood	Latitude	Longitude
2	M1C\n	Scarborough\n	Rouge Hill, Port Union, Highland Cre	43.78574	-79.15875

	Postalcode	**Borough**	**Neighborhood**	**Latitude**	**Longitude**
			ek		
3	M1E\n	Scarborough\n	Guildwood,Morningside,WestH	43.765575	-79.177470

	Postalcode	**Borough**	**Neighborhood**	**Latitude**	**Longitude**
			ill		
4	M1G\n	Scarborough\n	Woburn	43.76812	-79.21761
5	M1H\n	Scarborough	Cedarbrae	43.76944	-79.23892

	Postalcode	Borough	Neighborhood	Latitude	Longitude
		\n			
6	M1J\n	Scarborough\n	ScarboroughVillage	43.744446	-79.231117
7	M1K\n	Scarbor	Kennedy	43.725	-79.264

	Postalcode	Borough	Neighborhood	Latitude	Longitude
		ough\n	Park,Ionview,EastBirchmountPark	82	61

	Postalcode	Borough	Neighborhood	Latitude	Longitude
8	M1L\n	Scarborough\n	Golden Mile, Clairlea, Oakridge	43.71289	-79.28506

	Postalcode	Borough	Neighborhood	Latitude	Longitude
9	M1M\n	Scarborough\n	Cliffside,Cliffcrest,Scarboroug	43.72360	-79.23496

	Postalcode	Borough	Neighborhood	Latitude	Longitude
			hVillageWest		

In [18]:

```
address = 'Scarborough,Toronto'

geolocator = Nominatim(user_agent="http")
location = geolocator.geocode(address)
latitude_x = location.latitude
longitude_y = location.longitude
print('The Geograpical Co-ordinate of Seattle,Washington are {}, {}.'.format(
latitude_x, longitude_y))
```

```
The Geograpical Co-ordinate of Seattle,Washington are 43.7729744, -79.2576479
.
```

3. Map of Scarborough

In [19]:

```
map_Scarborough = folium.Map(location=[latitude_x, longitude_y], zoom_start=1
0)
```

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

    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[19]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [20]:

```
address = 'Scarborough,Toronto'

geolocator = Nominatim(user_agent="http")
location = geolocator.geocode(address)
latitude_n1 = location.latitude
longitude_n1 = location.longitude
print('The Geograpical Co-ordinate of Neighborhood_1 are {}, {}.'.format(lati
tude_x, longitude_y))
```

```
The Geograpical Co-ordinate of Neighborhood_1 are 43.7729744, -79.2576479.
```

In [21]:

```
# @hiddel_cell
CLIENT_ID = 'DPBYY4JUY3DU20ALPSUV4ONY2K1GOJJKJ1NIHBB32XEMOVYY' # my Foursquar
e ID
CLIENT_SECRET = '1MV443TYEP4HUO0WDUW5NQ5W10L2Y4G05NWG11WIR3NUGC5B' # my Fours
quare Secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: '+CLIENT_ID)
print('CLIENT_SECRET: '+CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: DPBYY4JUY3DU20ALPSUV4ONY2K1GOJJKJ1NIHBB32XEMOVYY
CLIENT_SECRET: 1MV443TYEP4HUO0WDUW5NQ5W10L2Y4G05NWG11WIR3NUGC5B
```

In [22]:

```
radius = 700
LIMIT = 100
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secr
et={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    latitude_n1,
   longitude_n1,
    radius,
   LIMIT)
results = requests.get(url).json()
```

In [23]:

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

Out[23]:

```
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 [24]:

```
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 [25]:

```
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', '
venue.location.lng']
```

```
nearby_venues =nearby_venues.loc[:, filtered_columns]
nearby_venues.head()
```

Out[25]:

	venue.name	venue.categories	venue.location.lat	venue.location.lng
0	DisneyStore	[{'id': '4bf58dd8d48988d1f3941735', 'name': 'T...	43.775537	-79.256833
1	SEPHO	[{'id': '4bf58dd8d48988d10c95	43.7750	-79.258

	venue.name	venue.categories	venue.location.lat	venue.location.lng
	RA	1735', 'name': 'C...	17	109
2	ColiseumScaroboroug	[{'id': '4bf58dd8d48988d17f941735', 'name': 'M...	43.775995	-79.255649

	venue.name	venue.categories	venue.location.lat	venue.location.lng
	hCinemas			
3	TommyHilfiger	[{'id': '4bf58dd8d48988d103951735', 'name': 'C...	43.7760015	-79.2573669

	venue.name	venue.categories	venue.location.lat	venue.location.lng
4	ShoppersDrugMart	[{'id': '4bf58dd8d48988d10f951735', 'name': 'P...	43.773305	-79.251662

5. Categories of Nearby Venues/Locations

In [26]:

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

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]
```

```
nearby_venues.head(5)
```

Out[26]:

	name	categories	lat	lng
0	DisneyStore	Toy/GameStore	43.775537	-79.256833
1	SEPHORA	CosmeticsShop	43.775017	-79.258109

	name	categories	lat	lng
2	ColiseumScarboroughCinemas	MovieTheater	43.775995	-79.255649
3	Tommy	Cloth	43.77	-79.2

	name	categories	lat	lng
	Hilfiger	ingStore	6015	57369
4	ShoppersDrugMart	Pharmacy	43.773305	-79.251662

In [27]:

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

Out[27]:

```
Clothing Store            8
```

```
Restaurant                  5
Coffee Shop                 5
Sandwich Place              2
Furniture / Home Store      2
Gas Station                 2
Intersection                2
Department Store            2
Pharmacy                    2
Grocery Store               1
Name: categories, dtype: int64
```

In [28]:

```
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={}&cli
ent_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # making GET request
        venue_results = requests.get(url).json()["response"]['groups'][0]['it
ems']

        # 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
venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
```

```
                       'Neighborhood Longitude',
                       'Venue',
                       'Venue Latitude',
                       'Venue Longitude',
                       'Venue Category']

    return(nearby_venues)
```

In [29]:

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

```
Not assigned

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
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Hillcrest Village
Fairview, Henry Farm, Oriole
```

Bayview Village
York Mills, Silver Hills
Willowdale, Newtonbrook
Willowdale, Willowdale East
York Mills West
Willowdale, Willowdale West
Not assigned

Not assigned

Not assigned

Not assigned
Not assigned

Not assigned

Not assigned

Parkwoods
Don Mills
Don Mills
Not assigned

Not assigned

Bathurst Manor, Wilson Heights, Downsview North
Northwood Park, York University
Downsview
Downsview
Downsview
Downsview
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned
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
Not assigned

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 Qua
y, South Niagara, Island airport
Stn A PO Boxes
First Canadian Place, Underground city
Not assigned

Not assigned

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
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned
Queen's Park, Ontario Provincial Government
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Canada Post Gateway Processing Centre
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Business reply mail Processing Centre, South Central Letter Processing Pla
nt Toronto
Not assigned

Not assigned
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

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 Queensway East, Royal York South East, Kingsway Park South East
Mimico NW, The Queensway West, South of Bloor, Kingsway Park South West, Royal York South West
Islington Avenue, Humber Valley Village
West Deane Park, Princess Gardens, Martin Grove, Islington, Cloverdale
Eringate, Bloordale Gardens, Old Burnhamthorpe, Markland Wood
Not assigned

Not assigned

Not assigned

Not assigned

Not assigned

Humber Summit
Humberlea, Emery
Weston
Westmount
Kingsview Village, St. Phillips, Martin Grove Gardens, Richview Gardens
Not assigned

Not assigned

South Steeles, Silverstone, Humbergate, Jamestown, Mount Olive, Beaumond Heights, Thistletown, Albion Gardens

Northwest, West Humber - Clairville
Not assigned

Not assigned

Not assigned

In [30]:

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

There are 307 Uniques Categories.

Out[30]:

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
Agincou	20	20	20	20	20	20

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
rt						
Alde	7	7	7	7	7	7

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
rwood, Lo						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
ngBranch						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
Bathurst	13	13	13	13	13	13

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
Manor, Wi						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
lsonHeig						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
hts,Down						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
sviewNor						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
th						
Bayv	6	6	6	6	6	6

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
iewVilla						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
ge						
Bedf	24	24	24	24	24	24

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
ordPark,						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
Lawrence						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
ManorEa						

	NeighborhoodLatitude	NeighborhoodLongitude	Venue	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
st						

One Hot Encoding of Features

In [31]:

```
# one hot encoding
Scarborough_onehot = pd.get_dummies(Scarborough_venues[['Venue Category']], p
refix="", prefix_sep="")
```

```
# 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.co
lumns[:-1])
Scarborough_onehot = Scarborough_onehot[fixed_columns]
Scarborough_grouped = Scarborough_onehot.groupby('Neighborhood').mean().reset
_index()
Scarborough_onehot.head(5)
```

Out[31]:

In [32]:

```
num_top_venues = 5
for hood in Scarborough_grouped['Neighborhood']:
    print("---- "+hood+" ----")
    temp =Scarborough_grouped[Scarborough_grouped['Neighborhood'] == 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).he
ad(num_top_venues))
    print('\n')
```

```
---- Agincourt ----
                venue  freq
0       Shopping Mall  0.10
1      Breakfast Spot  0.05
2  Chinese Restaurant  0.05
3         Skating Rink  0.05
4                 Park  0.05


---- Alderwood, Long Branch ----
          venue  freq
0           Pub  0.14
1   Pizza Place  0.14
2           Gym  0.14
3   Gas Station  0.14
4      Pharmacy  0.14


---- Bathurst Manor, Wilson Heights, Downsview North ----
                     venue  freq
0              Coffee Shop  0.15
1  Mediterranean Restaurant  0.08
2              Men's Store  0.08
3                     Park  0.08
4        Sushi Restaurant  0.08


---- Bayview Village ----
               venue  freq
0               Park  0.17
1  Asian Restaurant  0.17
2        Flower Shop  0.17
3              Trail  0.17
4            Dog Run  0.17
---- Bedford Park, Lawrence Manor East ----
                venue  freq
0      Sandwich Place  0.08
1         Coffee Shop  0.08
```

```
2           Pet Store  0.08
3  Italian Restaurant  0.08
4     Thai Restaurant  0.04


---- Berczy Park ----
                 venue  freq
0          Coffee Shop  0.09
1  Japanese Restaurant  0.05
2                Hotel  0.05
3                 Café  0.04
4           Restaurant  0.03


---- Birch Cliff, Cliffside West ----
                   venue  freq
0                   Park  0.17
1                   Café  0.17
2                    Gym  0.17
3  General Entertainment  0.17
4           Skating Rink  0.17


---- Brockton, Parkdale Village, Exhibition Place ----
         venue  freq
0         Café  0.07
1  Coffee Shop  0.07
2          Bar  0.06
3       Bakery  0.05
4   Restaurant  0.05
---- Business reply mail Processing Centre, South Central Letter Processin
g Plant Toronto ----
                 venue  freq
0          Coffee Shop  0.11
1                 Café  0.07
2                Hotel  0.07
3           Restaurant  0.04
4  Japanese Restaurant  0.03


---- CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurs
t Quay, South Niagara, Island airport ----
           venue  freq
0    Coffee Shop  0.09
1            Gym  0.07
2           Café  0.06
3  Grocery Store  0.05
4           Park  0.05


---- Caledonia-Fairbanks ----
                venue  freq
0                Park  0.33
1              Bakery  0.17
```

```
2       Women's Store  0.17
3  Mexican Restaurant  0.17
4         Pizza Place  0.17
---- Canada Post Gateway Processing Centre ----
                venue  freq
0         Coffee Shop  0.11
1                Café  0.07
2               Hotel  0.07
3          Restaurant  0.04
4  Japanese Restaurant  0.03


---- Cedarbrae ----
              venue  freq
0            Bakery  0.12
1       Gas Station  0.12
2   Thai Restaurant  0.12
3  Hakka Restaurant  0.12
4            Lounge  0.12


---- Central Bay Street ----
            venue  freq
0     Coffee Shop  0.14
1  Clothing Store  0.06
2            Café  0.03
3     Art Gallery  0.03
4           Diner  0.02
---- Christie ----
               venue  freq
0   Korean Restaurant  0.16
1       Grocery Store  0.10
2         Pizza Place  0.04
3  Mexican Restaurant  0.04
4                Café  0.04


---- Church and Wellesley ----
                venue  freq
0         Coffee Shop  0.09
1  Japanese Restaurant  0.06
2     Sushi Restaurant  0.04
3           Restaurant  0.04
4                 Café  0.03


---- Clarks Corners, Tam O'Shanter, Sullivan ----
                  venue  freq
0   Fast Food Restaurant  0.09
1               Pharmacy  0.06
2  Vietnamese Restaurant  0.06
3            Pizza Place  0.06
4           Liquor Store  0.03
---- Cliffside, Cliffcrest, Scarborough Village West ----
```

```
            venue  freq
0  Ice Cream Shop  0.17
1  Hardware Store  0.08
2          Bistro  0.08
3            Bank  0.08
4        Pharmacy  0.08


---- Commerce Court, Victoria Hotel ----
                venue  freq
0         Coffee Shop  0.10
1                Café  0.06
2           Gastropub  0.04
3  Seafood Restaurant  0.04
4                 Gym  0.03


---- Davisville ----
            venue  freq
0     Pizza Place  0.09
1  Sandwich Place  0.07
2    Dessert Shop  0.07
3     Coffee Shop  0.07
4            Café  0.05


---- Davisville North ----
               venue  freq
0        Pizza Place  0.08
1   Department Store  0.04
2     Sandwich Place  0.04
3  Food & Drink Shop  0.04
4               Café  0.04
---- Del Ray, Mount Dennis, Keelsdale and Silverthorn ----
                        venue  freq
0                Home Service  0.33
1        Fast Food Restaurant  0.17
2  Construction & Landscaping  0.17
3                  Playground  0.17
4                 Coffee Shop  0.17


---- Don Mills ----
                 venue  freq
0          Coffee Shop  0.08
1           Restaurant  0.06
2  Japanese Restaurant  0.06
3       Sandwich Place  0.04
4                  Gym  0.04


---- Dorset Park, Wexford Heights, Scarborough Town Centre ----
            venue  freq
0         Brewery   0.2
```

```
1  Rental Service   0.2
2     Coffee Shop   0.2
3          Bakery   0.2
4            Park   0.2


---- Downsview ----
                   venue  freq
0            Coffee Shop  0.09
1          Grocery Store  0.06
2            Pizza Place  0.06
3         Discount Store  0.05
4  Vietnamese Restaurant  0.05
---- Dufferin, Dovercourt Village ----
           venue  freq
0            Bar  0.10
1    Coffee Shop  0.10
2         Bakery  0.08
3           Park  0.04
4  Grocery Store  0.04


---- East Toronto, Broadview North (Old East York) ----
                venue  freq
0      Farmers Market  0.17
1            Bus Line  0.17
2         Coffee Shop  0.17
3  Italian Restaurant  0.17
4                Park  0.17


---- Eringate, Bloordale Gardens, Old Burnhamthorpe, Markland Wood ----
               venue  freq
0               Park  0.18
1  Convenience Store  0.09
2        Pizza Place  0.09
3      Shopping Mall  0.09
4     Baseball Field  0.09


---- Fairview, Henry Farm, Oriole ----
                  venue  freq
0        Clothing Store  0.11
1           Coffee Shop  0.08
2  Fast Food Restaurant  0.05
3            Restaurant  0.03
4     Convenience Store  0.03
---- First Canadian Place, Underground city ----
              venue  freq
0             Hotel  0.10
1       Coffee Shop  0.08
2              Café  0.06
3        Restaurant  0.05
4  Asian Restaurant  0.03
```

```
---- Forest Hill North & West, Forest Hill Road Park ----
                      venue  freq
0                      Bank  0.15
1        Salon / Barbershop  0.08
2                      Café  0.08
3                 Bookstore  0.08
4  Mediterranean Restaurant  0.08


---- Garden District, Ryerson ----
                 venue  freq
0          Coffee Shop  0.10
1       Clothing Store  0.04
2            Gastropub  0.03
3  Japanese Restaurant  0.03
4   Italian Restaurant  0.03


---- Glencairn ----
               venue  freq
0    Grocery Store  0.15
1      Pizza Place  0.10
2   Ice Cream Shop  0.05
3  Photography Lab  0.05
4      Fish Market  0.05


---- Golden Mile, Clairlea, Oakridge ----
          venue  freq
0        Bakery  0.14
1  Intersection  0.14
2         Diner  0.07
3      Bus Line  0.07
4   Bus Station  0.07


---- Guildwood, Morningside, West Hill ----
                           venue  freq
0                           Park   0.4
1           Gym / Fitness Center   0.2
2                 Gymnastics Gym   0.2
3             Athletics & Sports   0.2
4  Paper / Office Supplies Store   0.0


---- Harbourfront East, Union Station, Toronto Islands ----
                 venue  freq
0          Coffee Shop  0.11
1                 Café  0.06
2                Hotel  0.06
3  Japanese Restaurant  0.03
4           Restaurant  0.03
```

```
---- High Park, The Junction South ----
               venue  freq
0  Convenience Store  0.10
1                Bar  0.08
2    Thai Restaurant  0.06
3        Pizza Place  0.04
4             Bakery  0.04
---- Hillcrest Village ----
                                      venue  freq
0                                      Park  0.29
1  Residential Building (Apartment / Condo)  0.14
2                                  Pharmacy  0.14
3                                    Bakery  0.14
4                        Chinese Restaurant  0.14


---- Humber Summit ----
          venue  freq
0  Home Service  0.33
1    Hobby Shop  0.33
2   Music Store  0.33
3   Zoo Exhibit  0.00
4     Nightclub  0.00


---- Humberlea, Emery ----
                       venue  freq
0                Coffee Shop  0.29
1  Latin American Restaurant  0.14
2                  Nightclub  0.14
3             Discount Store  0.14
4                       Café  0.14


---- Humewood-Cedarvale ----
               venue  freq
0      Grocery Store  0.22
1  Convenience Store  0.11
2       Hockey Arena  0.11
3     Soccer Stadium  0.11
4              Field  0.11
```

```
---- Hillcrest Village ----
                                      venue  freq
0                                      Park  0.29
1  Residential Building (Apartment / Condo)  0.14
2                                  Pharmacy  0.14
3                                    Bakery  0.14
4                        Chinese Restaurant  0.14


---- Humber Summit ----
          venue  freq
```

```
0  Home Service  0.33
1    Hobby Shop  0.33
2   Music Store  0.33
3   Zoo Exhibit  0.00
4     Nightclub  0.00


---- Humberlea, Emery ----
                      venue  freq
0               Coffee Shop  0.29
1  Latin American Restaurant  0.14
2                 Nightclub  0.14
3            Discount Store  0.14
4                      Café  0.14


---- Humewood-Cedarvale ----
               venue  freq
0      Grocery Store  0.22
1  Convenience Store  0.11
2       Hockey Arena  0.11
3     Soccer Stadium  0.11
4              Field  0.11
---- India Bazaar, The Beaches West ----
                  venue  freq
0  Fast Food Restaurant  0.09
1            Restaurant  0.09
2                Bakery  0.06
3               Brewery  0.06
4           Coffee Shop  0.06


---- Islington Avenue, Humber Valley Village ----
                 venue  freq
0             Pharmacy  0.18
1    Convenience Store  0.09
2  Japanese Restaurant  0.09
3                 Café  0.09
4         Skating Rink  0.09


---- Kennedy Park, Ionview, East Birchmount Park ----
                venue  freq
0      Discount Store  0.33
1  Chinese Restaurant  0.17
2    Department Store  0.17
3         Bus Station  0.17
4         Coffee Shop  0.17


---- Kensington Market, Chinatown, Grange Park ----
                           venue  freq
0                           Café  0.08
1                    Coffee Shop  0.06
```

```
2  Vegetarian / Vegan Restaurant  0.06
3                            Bar  0.04
4                   Yoga Studio  0.03
---- Kingsview Village, St. Phillips, Martin Grove Gardens, Richview Garde
ns ----
                venue  freq
0    Mobile Phone Shop  0.14
1  American Restaurant  0.14
2             Bus Line  0.14
3             Pharmacy  0.14
4     Business Service  0.14


---- Lawrence Manor, Lawrence Heights ----
              venue  freq
0    Clothing Store  0.17
1      Dessert Shop  0.05
2        Restaurant  0.05
3  Greek Restaurant  0.03
4         Bookstore  0.03


---- Lawrence Park ----
                 venue  freq
0          Coffee Shop  0.14
1  Gym / Fitness Center  0.14
2             Bus Line  0.14
3           Restaurant  0.14
4            Bookstore  0.14


---- Leaside ----
                venue  freq
0         Coffee Shop  0.08
1  Sporting Goods Shop  0.06
2          Restaurant  0.04
3     Department Store  0.04
4        Shopping Mall  0.04
---- York Mills West ----
            venue  freq
0      Restaurant  0.19
1     Coffee Shop  0.14
2             Gym  0.10
3  Sandwich Place  0.05
4   Metro Station  0.05


---- York Mills, Silver Hills ----
            venue  freq
0     Concert Hall   0.5
1             Park   0.5
```

```
2       Zoo Exhibit   0.0
3     Noodle House   0.0
4  Paintball Field   0.0
```

In [33]:

```
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]
```

Most Common venues near neighborhood

In [34]:

```
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['Neighborho
od']

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

neighborhoods_venues_sorted.head()
```

Out[34]:

K-Means Clustering Approach

In [35]:

```
# 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_cluster
ing)
kmeans.labels_
```

Out[35]:

```
array([0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 1, 0, 2, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0,
       0, 2, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       2, 0, 0, 0, 0, 0, 2, 2, 0, 0, 2], dtype=int32)
```

In [36]:

```
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
```

```python
Scarborough_merged = Scarborough_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

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

Out[36]:

Map of Clusters

In [37]:

```
kclusters = 10
```

In [38]:

```
# 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=Tr
ue)
    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[38]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [39]:

```
df1=Scarborough_merged.loc[Scarborough_merged['Cluster Labels'] == 0,Scarboro
ugh_merged.columns[[2] + list(range(5, Scarborough_merged.shape[1]))]]
df2=Scarborough_merged.loc[Scarborough_merged['Cluster Labels'] == 1,Scarboro
ugh_merged.columns[[2] + list(range(5, Scarborough_merged.shape[1]))]]
df3=Scarborough_merged.loc[Scarborough_merged['Cluster Labels'] == 2,Scarboro
ugh_merged.columns[[2] + list(range(5, Scarborough_merged.shape[1]))]]
```

In [46]:

```
Scarborough_Avg_HousingPrice=pd.DataFrame({"Neighborhood":df_2["Neighborhood"
],
                                          "Average_Housing_Price":[335000.0,28660
0.0,175000.0,225900.0,219400.0,
                                                                  573900.0,22500
0.0,370500.0,370500.0,433500.0,279200.0,
                                                                  279200.0,22500
0.0,370500.,255400.0,433500.0,433500.0,
                                                                  435000.0,28950
0.0,265000.0,285900.0,239400.0,
                                                                  589900.0,29500
0.0,380500.0,378500.0,438500.0,229200.0,
                                                                  229200.0,36500
0.0,388500.,285400.0,493500.0,477500.0,378000.0,316600.0,195000.0,225900.0,21
9400.0,
                                                                  573900.0,36700
0.0,370500.0,370500.0,363500.0,279200.0,
                                                                  279200.0,27100
0.0,370500.,255400.0,383500.0,433500.0,335000.0,286600.0,185000.0,225900.0,21
9400.0,
```

```
                                                     573900.0,32900
0.0,370500.0,370500.0,533500.0,279200.0,
                                                     279200.0,37500
0.0,370500.,255400.0,493500.0,433500.0,335000.0,286600.0,165000.0,225900.0,21
9400.0,
                                                     573900.0,42500
0.0,370500.0,370500.0,433500.0,279200.0,
                                                     279200.0,19500
0.0,370500.,255400.0,403500.0,433500.0,335000.0,286600.0,187000.0,225900.0,21
9400.0,
                                                     573900.0,32500
0.0,370500.0,370500.0,333500.0,279200.0,
                                                     279200.0,28900
0.0,370500.,255400.0,413500.0,433500.0,254800.0,
                                                     573900.0,32900
0.0,370500.0,370500.0,533500.0,279200.0,
                                                     279200.0,37500
0.0,370500.,255400.0,493500.0,433500.0,335000.0,286600.0,165
000.0,225900.0,219400.0,
                                                     573900.0,32900
0.0,370500.0,370500.0,533500.0,279200.0,
                                                     279200.0,37500
0.0,370500.,255400.0,493500.0,433500.0,335000.0,286600.0,165000.0,225900.0,21
9400.0,
                                                     573900.0,32900
0.0,370500.0,370500.0,533500.0,279200.0,
                                                     279200.0,37500
0.0,370500.,255400.0,493500.0,433500.0,335000.0,286600.0,165000.0,225900.0,21
9400.0,
                                                     573900.0,32900
0.0,370500.0,370500.0,533500.0,279200.0,
                                                     279200.0,37500
0.0,370500.,255400.0,493500.0,433500.0,335000.0,286600.0,165000.0,225900.0,21
9400.0,
                                                     573900.0,32900
0.0,370500.0,370500.0,533500.0,279200.0,
                                                     279200.0,37500
0.0,370500.


                                                    ]})
```

In [47]:

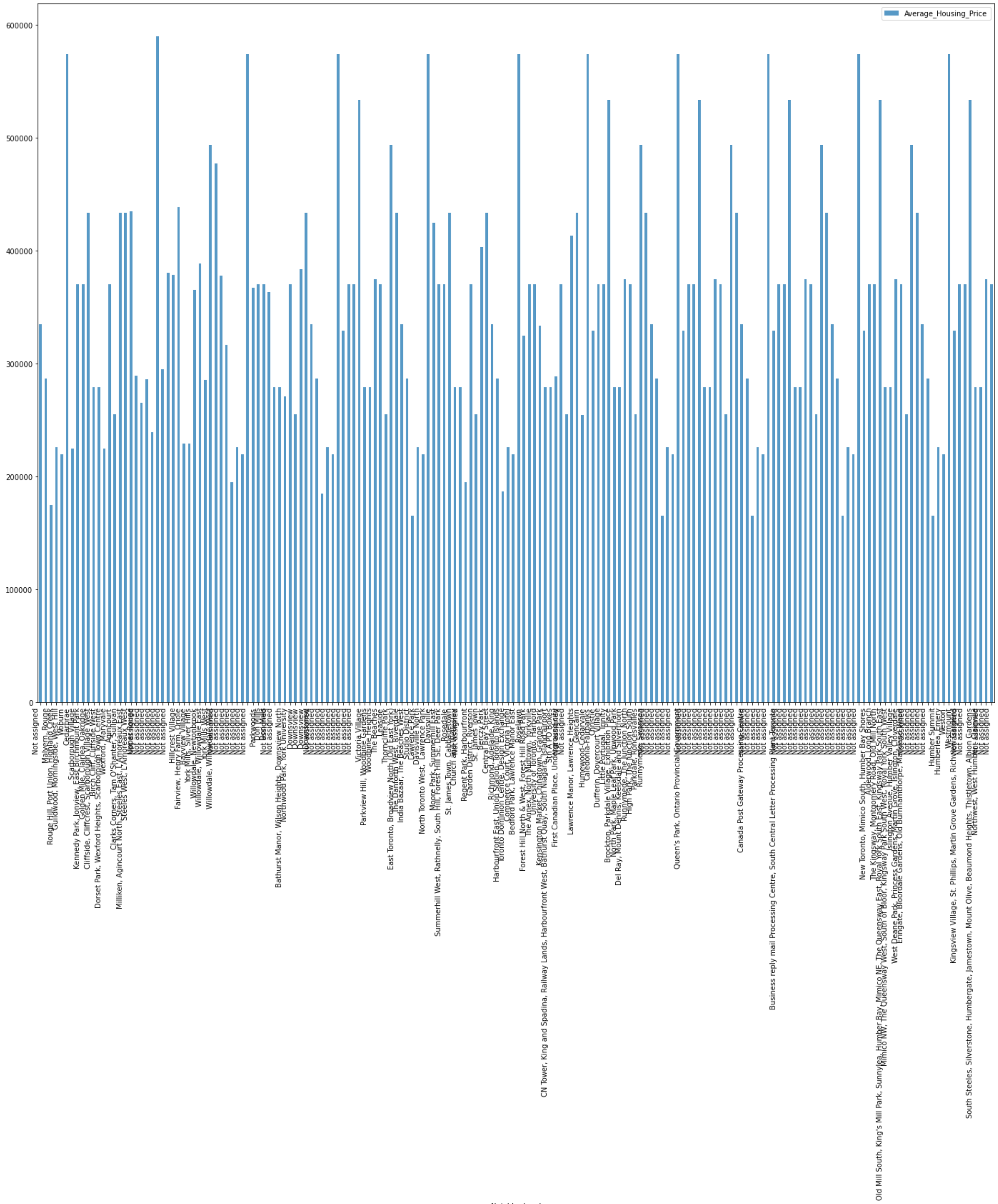
```
Scarborough_Avg_HousingPrice.set_index('Neighborhood',inplace=True,drop=True)
```

In [48]:

```
Scarborough_Avg_HousingPrice.plot(kind='bar',figsize=(24,18),alpha=0.75)
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb7580dee80>
```



School Rating by Clusters

In [49]:

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

In [54]:

```
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
,1,2,1,2,7,2,3,2,6,
                                                            5,4,8,9,9,6,6,4,5,
4,6,8,10,8,9,6,2,
                                                            10,2,5,8,9,6,6,10,
8,9,1,2,3,4,5,6,9,
                                                            8,5,9,6,9,6,4,8,10
,2,5,6,3,9,8,7,
                                                            7,8,5,8,9,1,5,4,7,
2,3,6,6,9,4,8,7,
                                                            4,8,9,2,6,4,7,5,10
,4,6,8,9,7,5,6,5,8,7,
                                                            5,9,6,9,6,4,8,10,2
,5,6,3,9,8,7,
                                                            7,8,5,8,9,1,5,4,7,
2,3,6,6,9,4,8,7,
                                                            4,8,9,2,6,4,7,5,10
,4,6,8,9,7,5,6,5,8,7,
                                                            5,9,6,9,6,4,8,10,2
,5,6,3,9,8,7,
                                                            7,8,5,8,9,1,5,4,7,
2,3

                                                            ]})
```

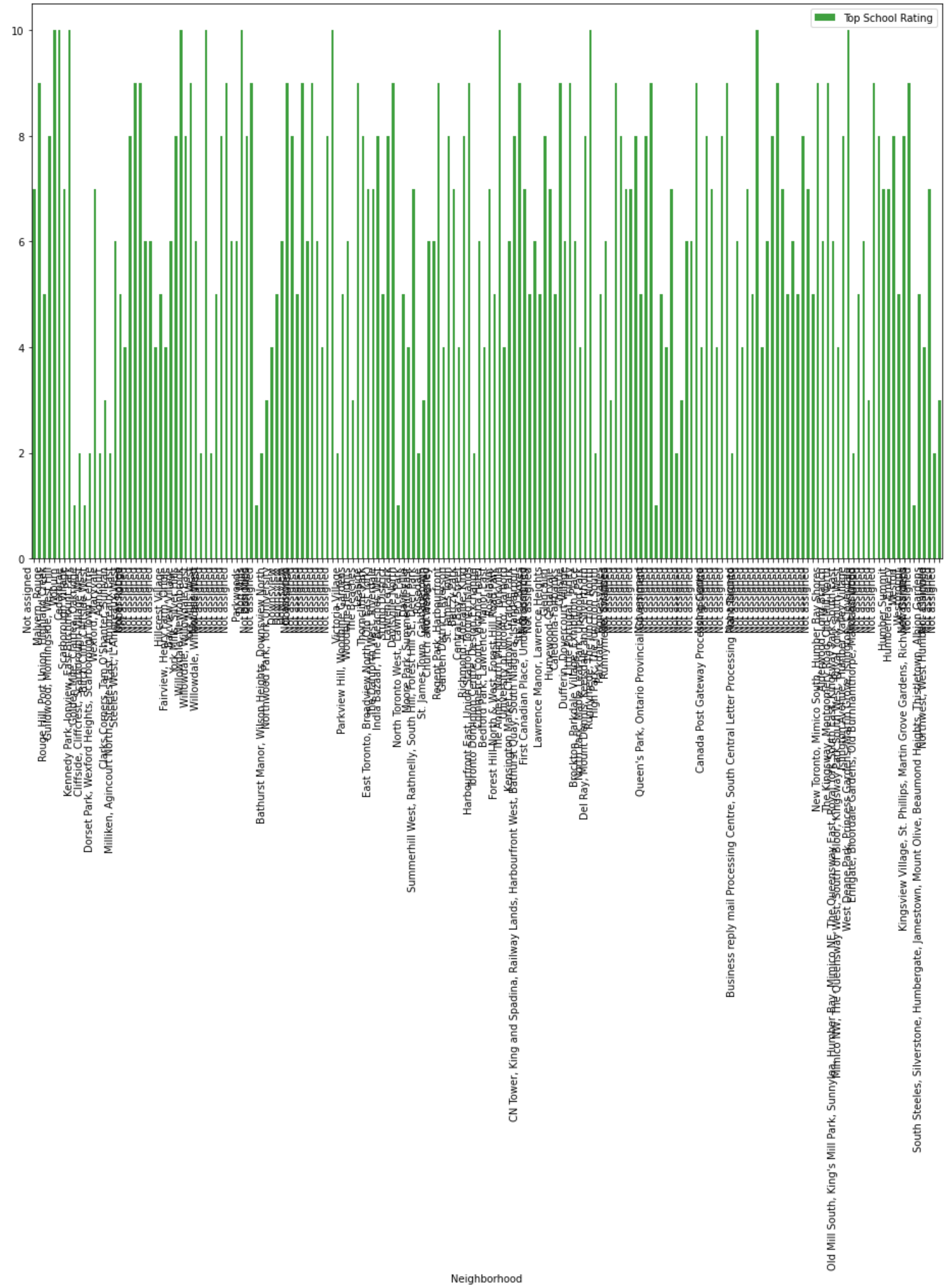
In [55]:

```
Scarborough_school_ratings.set_index('Neighborhood',inplace=True,drop=True)
```

In [56]:

```
Scarborough_school_ratings.plot(kind='bar',figsize=(16,10),color='green',alph
a=0.75);
```

Conclusion: In this project, using k-means cluster algorithm I separated the neighborhood into 10(Ten) different clusters and for 103 different lattitude and logitude from dataset, which have very-similar neighborhoods around them. Using the charts above results presented to a particular neighborhood based on average house prices and school rating have been made.



Conclusion: In this project, using k-means cluster algorithm I separated the neighborhood into 10(Ten) different clusters and for 103 different lattitude and logitude from dataset, which have very-similar neighborhoods around them. Using the charts above results presented to a particular neighborhood based on average house prices and school rating have been made.