Segmenting and Clustering Neighborhoods in Toronto part 2

March 22, 2019

In [27]:
```
import numpy as np # data in a vectorized manner manipulation
import pandas as pd # data analsysis
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
import json # JSON files manipulation
import requests # HTTP library
from bs4 import BeautifulSoup # scraping library

from sklearn.cluster import KMeans # clustering algorithm

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

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

# !conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

In [28]:
```
url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"

text_result = requests.get(url).text #get the entire html of the article as a str
html_parsed_result = BeautifulSoup(text_result, 'html.parser') #transform the text to h

neightborhood_info_table = html_parsed_result.find('table', class_ = 'wikitable')
neightborhood_rows = neightborhood_info_table.find_all('tr')

# extract the info ('Postcode', 'Borough', 'Neighbourhood') from the table
neightborhood_info = []
for row in neightborhood_rows:
    info = row.text.split('\n')[1:-1] # remove empty str (first and last items)
    neightborhood_info.append(info)
```

```
neightborhood_info[0:10]
```

Out[28]:
```
[['Postcode', 'Borough', 'Neighbourhood'],
 ['M1A', 'Not assigned', 'Not assigned'],
 ['M2A', 'Not assigned', 'Not assigned'],
 ['M3A', 'North York', 'Parkwoods'],
 ['M4A', 'North York', 'Victoria Village'],
 ['M5A', 'Downtown Toronto', 'Harbourfront'],
 ['M5A', 'Downtown Toronto', 'Regent Park'],
 ['M6A', 'North York', 'Lawrence Heights'],
 ['M6A', 'North York', 'Lawrence Manor'],
 ['M7A', "Queen's Park", 'Not assigned']]
```

In [29]:
```
#create a Neighborhoods dataframe
neightborhood_info[0][-1] = 'Neighborhood' # change to american spelling
neighborhood_df = pd.DataFrame(neightborhood_info[1:], columns=neightborhood_info[0])

neighborhood_df.head(10)
```

Out[29]:

	Postcode	Borough	Neighborhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront
5	M5A	Downtown Toronto	Regent Park
6	M6A	North York	Lawrence Heights
7	M6A	North York	Lawrence Manor
8	M7A	Queen's Park	Not assigned
9	M8A	Not assigned	Not assigned

In [30]:
```
not_assigned_boroughs = neighborhood_df.index[neighborhood_df['Borough'] == 'Not assign
not_assigned_neighborhoods = neighborhood_df.index[neighborhood_df['Neighborhood'] == '
not_assigned_neighborhoods_and_borough = not_assigned_boroughs & not_assigned_neighborh

print('The DataFrame shape is {}'.format(neighborhood_df.shape),'\n')
print('There are:')
print('  {} Postal codes'.format(neighborhood_df['Postcode'].unique().shape[0]))
print('  {} Boroughs'.format(neighborhood_df['Borough'].unique().shape[0] - 1)) # subst
print('  {} Neighborhoods'.format(neighborhood_df['Neighborhood'].unique().shape[0] - 1
print('  {} rows with Not assigned Borough'.format(not_assigned_boroughs.shape[0]))
print('  {} rows with Not assigned Neighborhood'.format(not_assigned_neighborhoods.shap
print('  {} rows with Not assigned Neighborhood and Borough'.format(not_assigned_neighb
```

```
The DataFrame shape is (289, 3)

There are:
  180 Postal codes
```

```
11 Boroughs
209 Neighborhoods
77 rows with Not assigned Borough
78 rows with Not assigned Neighborhood
77 rows with Not assigned Neighborhood and Borough
```

In [31]:
```
neighborhood_df.drop(neighborhood_df.index[not_assigned_boroughs], inplace=True)
neighborhood_df.reset_index(drop=True, inplace=True)

neighborhood_df.head(10)
```

Out[31]:

	Postcode	Borough	Neighborhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Harbourfront
3	M5A	Downtown Toronto	Regent Park
4	M6A	North York	Lawrence Heights
5	M6A	North York	Lawrence Manor
6	M7A	Queen's Park	Not assigned
7	M9A	Etobicoke	Islington Avenue
8	M1B	Scarborough	Rouge
9	M1B	Scarborough	Malvern

In [32]:
```
not_assigned_neighborhoods = neighborhood_df.index[neighborhood_df['Neighborhood'] == '

for idx in not_assigned_neighborhoods:
    neighborhood_df['Neighborhood'][idx] = neighborhood_df['Borough'][idx]

neighborhood_df.head(10)
```

Out[32]:

	Postcode	Borough	Neighborhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Harbourfront
3	M5A	Downtown Toronto	Regent Park
4	M6A	North York	Lawrence Heights
5	M6A	North York	Lawrence Manor
6	M7A	Queen's Park	Queen's Park
7	M9A	Etobicoke	Islington Avenue
8	M1B	Scarborough	Rouge
9	M1B	Scarborough	Malvern

In [33]:
```
print('After cleaning the DataFrame, its new shape is {}'.format(neighborhood_df.shape)
print('There are:')
print('  {} Postal codes'.format(neighborhood_df['Postcode'].unique().shape[0]))
print('  {} Boroughs'.format(neighborhood_df['Borough'].unique().shape[0]))
print('  {} Neighborhoods'.format(neighborhood_df['Neighborhood'].unique().shape[0]))
```

```
After cleaning the DataFrame, its new shape is (212, 3)

There are:
  103 Postal codes
  11 Boroughs
  210 Neighborhoods
```

```
In [34]: group = neighborhood_df.groupby('Postcode')
         grouped_neighborhoods = group['Neighborhood'].apply(lambda x: "%s" % ', '.join(x))
         grouped_boroughs = group['Borough'].apply(lambda x: set(x).pop())
         grouped_df = pd.DataFrame(list(zip(grouped_boroughs.index, grouped_boroughs, grouped_ne
         grouped_df.columns = ['Postcode', 'Borough', 'Neighborhood']

         grouped_df.head(10)
```

Out[34]:

	Postcode	Borough	Neighborhood
0	M1B	Scarborough	Rouge, Malvern
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae
5	M1J	Scarborough	Scarborough Village
6	M1K	Scarborough	East Birchmount Park, Ionview, Kennedy Park
7	M1L	Scarborough	Clairlea, Golden Mile, Oakridge
8	M1M	Scarborough	Cliffcrest, Cliffside, Scarborough Village West
9	M1N	Scarborough	Birch Cliff, Cliffside West

```
In [38]: print('The DataFrame shape is', grouped_df.shape)
```

```
The DataFrame shape is (103, 3)
```

```
In [39]: coordinates_df = pd.read_csv('Geospatial_Coordinates.csv') # transform the csv file int

         print('The coordinates dataframe shape is', coordinates_df.shape)
         coordinates_df.head()
```

```
The coordinates dataframe shape is (103, 3)
```

Out[39]:

	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 [41]: postcodes_with_coordinates_df = grouped_df.join(coordinates_df.set_index('Postal Code')

         postcodes_with_coordinates_df.head(16)
```

Out[41]:

	Postcode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Rouge, Malvern	43.806686	-79.194353
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union	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
5	M1J	Scarborough	Scarborough Village	43.744734	-79.239476
6	M1K	Scarborough	East Birchmount Park, Ionview, Kennedy Park	43.727929	-79.262029
7	M1L	Scarborough	Clairlea, Golden Mile, Oakridge	43.711112	-79.284577
8	M1M	Scarborough	Cliffcrest, Cliffside, Scarborough Village West	43.716316	-79.239476
9	M1N	Scarborough	Birch Cliff, Cliffside West	43.692657	-79.264848
10	M1P	Scarborough	Dorset Park, Scarborough Town Centre, Wexford ...	43.757410	-79.273304
11	M1R	Scarborough	Maryvale, Wexford	43.750072	-79.295849
12	M1S	Scarborough	Agincourt	43.794200	-79.262029
13	M1T	Scarborough	Clarks Corners, Sullivan, Tam O'Shanter	43.781638	-79.304302
14	M1V	Scarborough	Agincourt North, L'Amoreaux East, Milliken, St...	43.815252	-79.284577
15	M1W	Scarborough	L'Amoreaux West, Steeles West	43.799525	-79.318389

In [46]:
```
map = folium.Map(location=[43.6532,-79.3832], zoom_start=11)

for location in postcodes_with_coordinates_df.itertuples(): #iterate each row of the da
    label = 'Postal Code: {};  Borough: {};  Neighborhoods: {}'.format(location[1], loc
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [location[-2], location[-1]],
        radius=1,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
```

```
                parse_html=False).add_to(map)
            folium.Circle(
                radius=500,
                popup=label,
                location=[location[-2], location[-1]],
                color='#3186cc',
                fill=True,
                fill_color='#3186cc'
            ).add_to(map)

        map
```

Out[46]: <folium.folium.Map at 0x7f2fc1288b38>

In [ ]: