Segmenting and Clustering Neighborhoods in Toronto Part 1 and 2

April 6, 2020

1 IBM Applied Data Science Capstone Course by Coursera

1.0.1 Week 3 Part 1 and 2

.Build a dataframe of the postal code of each neighborhood along with the borough name and neighborhood name in Toronto.

.Get the geographical coordinates of the neighborhoods in Toronto.

[32]:
```
!conda install -c conda-forge geopy --yes
```

```
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.5.11
  latest version: 4.8.3

Please update conda by running

    $ conda update -n base -c defaults conda



# All requested packages already installed.
```

[33]:
```
get_ipython().system(u' pip install beautifulsoup4')
```

```
Requirement already satisfied: beautifulsoup4 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (4.8.2)
Requirement already satisfied: soupsieve>=1.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
beautifulsoup4) (2.0)
```

1. Import Libraries

[34]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
```

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

import json # library to handle JSON files

from geopy.geocoders import Nominatim # convert an address into latitude and␣
 ↪longitude values

import requests # library to handle requests
from bs4 import BeautifulSoup # library to parse HTML and XML documents

from pandas.io.json import json_normalize # tranform JSON file into a pandas␣
 ↪dataframe

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

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

import folium # map rendering library

print("Libraries imported.")
```

```
Libraries imported.
```

2. Scrap data from Wikipedia page into dataframe

[36]:
```
# send the GET request
data = requests.get('https://en.wikipedia.org/wiki/
 ↪List_of_postal_codes_of_Canada:_M').text
```

[37]:
```
# parse data from the html into a beautifulsoup object
soup = BeautifulSoup(data, 'html.parser')
```

[38]:
```
# create three lists to store table data
postalCodeList = []
boroughList = []
neighborhoodList = []
```

Using BeautifulSoup

[40]:
```
# find the table
soup.find('table').find_all('tr')

# find all the rows of the table
soup.find('table').find_all('tr')
```

```
# for each row of the table, find all the table data
for row in soup.find('table').find_all('tr'):
    cells = row.find_all('td')
```

[41]:
```
# append the data into the respective lists
for row in soup.find('table').find_all('tr'):
    cells = row.find_all('td')
    if(len(cells) > 0):
        postalCodeList.append(cells[0].text)
        boroughList.append(cells[1].text)
        neighborhoodList.append(cells[2].text.rstrip('\n')) # avoid new lines␣
 ↪in neighborhood cell
```

[42]:
```
# create a new DataFrame from the three lists
toronto_df = pd.DataFrame({"PostalCode": postalCodeList,
                           "Borough": boroughList,
                           "Neighborhood": neighborhoodList})
toronto_df['PostalCode'] = toronto_df['PostalCode'].replace('\n','', regex=True)
toronto_df['Borough'] = toronto_df['Borough'].replace('\n','', regex=True)
toronto_df.head()
```

[42]:

	PostalCode	Borough	Neighborhood
0	M1A	Not assigned	
1	M2A	Not assigned	
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park / Harbourfront

3. Drop cells with a Borough that is "Not assigned"

[43]:
```
# drop cells with a borough that is Not assigned
toronto_df_dropna = toronto_df[toronto_df.Borough != "Not assigned"].
 ↪reset_index(drop=True)
toronto_df_dropna.head()
```

[43]:

	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

4. Get Neighborhood in the same borough

[45]:
```
# group neighborhoods in the same borough
toronto_df_grouped = toronto_df_dropna.groupby(["PostalCode", "Borough"],␣
↪as_index=False).agg(lambda x: ", ".join(x))
toronto_df_grouped.head()
```

[45]:

	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

5. For the Neighborhood which is "Not assigned" make the value same as Borough

[46]:
```
# for Neighborhood="Not assigned", make the value the same as Borough
for index, row in toronto_df_grouped.iterrows():
    if row["Neighborhood"] == "Not assigned":
        row["Neighborhood"] = row["Borough"]

toronto_df_grouped.head()
```

[46]:

	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

6.Check whether it is the same as required by the question

[47]:
```
# create a new test dataframe
column_names = ["PostalCode", "Borough", "Neighborhood"]
test_df = pd.DataFrame(columns=column_names)

test_list = ["M5G", "M2H", "M4B", "M1J", "M4G", "M4M", "M1R", "M9V", "M9L",␣
↪"M5V", "M1B", "M5A"]

for postcode in test_list:
    test_df = test_df.
↪append(toronto_df_grouped[toronto_df_grouped["PostalCode"]==postcode],␣
↪ignore_index=True)

test_df
```

[47]:

	PostalCode	Borough	\
0	M5G	Downtown Toronto	
1	M2H	North York	

```
2          M4B         East York
3          M1J       Scarborough
4          M4G         East York
5          M4M      East Toronto
6          M1R       Scarborough
7          M9V         Etobicoke
8          M9L        North York
9          M5V  Downtown Toronto
10         M1B       Scarborough
11         M5A  Downtown Toronto
```

```
                                        Neighborhood
0                                  Central Bay Street
1                                   Hillcrest Village
2                    Parkview Hill / Woodbine Gardens
3                                Scarborough Village
4                                           Leaside
5                                   Studio District
6                                  Wexford / Maryvale
7   South Steeles / Silverstone / Humbergate / Jam…
8                                        Humber Summit
9   CN Tower / King and Spadina / Railway Lands / …
10                                   Malvern / Rouge
11                       Regent Park / Harbourfront
```

7. Finally, print the number of the rows of cleaned dataframe

[48]:
```
# print the number of rows of the cleaned dataframe
toronto_df_grouped.shape
```

[48]:
```
(103, 3)
```

8. Load the coordinate from CSV file

[49]:
```
# load the coordinates from the csv file on Coursera
coordinates = pd.read_csv("http://cocl.us/Geospatial_data")
coordinates.head()
```

[49]:
```
  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
```

[50]:
```
# rename the column "PostalCode"
coordinates.rename(columns={"Postal Code": "PostalCode"}, inplace=True)
coordinates.head()
```

[50]:
```
  PostalCode   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
```

9. Merge two tables to get the coordinates

[51]:
```
# merge two table on the column "PostalCode"
toronto_df_new = toronto_df_grouped.merge(coordinates, on="PostalCode",␣
↪how="left")
toronto_df_new.head()
```

[51]:
```
  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

   Latitude  Longitude
0  43.806686 -79.194353
1  43.784535 -79.160497
2  43.763573 -79.188711
3  43.770992 -79.216917
4  43.773136 -79.239476
```

10. Finally check to make sure that coordinates are added as required bu the question

[52]:
```
# create a new test dataframe
column_names = ["PostalCode", "Borough", "Neighborhood", "Latitude",␣
↪"Longitude"]
test_df = pd.DataFrame(columns=column_names)

test_list = ["M5G", "M2H", "M4B", "M1J", "M4G", "M4M", "M1R", "M9V", "M9L",␣
↪"M5V", "M1B", "M5A"]

for postcode in test_list:
    test_df = test_df.
↪append(toronto_df_new[toronto_df_new["PostalCode"]==postcode],␣
↪ignore_index=True)

test_df
```

[52]:
```
   PostalCode           Borough  \
0         M5G  Downtown Toronto
```

```
1          M2H        North York
2          M4B         East York
3          M1J       Scarborough
4          M4G         East York
5          M4M      East Toronto
6          M1R       Scarborough
7          M9V         Etobicoke
8          M9L        North York
9          M5V  Downtown Toronto
10         M1B       Scarborough
11         M5A  Downtown Toronto
```

	Neighborhood	Latitude	Longitude
0	Central Bay Street	43.657952	-79.387383
1	Hillcrest Village	43.803762	-79.363452
2	Parkview Hill / Woodbine Gardens	43.706397	-79.309937
3	Scarborough Village	43.744734	-79.239476
4	Leaside	43.709060	-79.363452
5	Studio District	43.659526	-79.340923
6	Wexford / Maryvale	43.750072	-79.295849
7	South Steeles / Silverstone / Humbergate / Jam…	43.739416	-79.588437
8	Humber Summit	43.756303	-79.565963
9	CN Tower / King and Spadina / Railway Lands / …	43.628947	-79.394420
10	Malvern / Rouge	43.806686	-79.194353
11	Regent Park / Harbourfront	43.654260	-79.360636

[ ]: