Untitled2

November 24, 2019

1 Segmenting and Clustering Neighborhoods in Toronto

1.0.1 Reading our data into a dataframe after storing it in a csv file

[3]:

```
import pandas as pd
import numpy as np
df=pd.read_csv('Classeur1.csv', sep=';')
df
```

[3]:

```
    Postcode       Borough   Neighbourhood
0        M9Z  Not assigned    Not assigned
1        M9Y  Not assigned    Not assigned
2        M9X  Not assigned    Not assigned
3        M9W     Etobicoke       Northwest
4        M9V     Etobicoke  Albion Gardens
..       …             …              …
282      M1C   Scarborough      Rouge Hill
283      M1C   Scarborough      Port Union
284      M1B   Scarborough           Rouge
285      M1B   Scarborough         Malvern
286      M1A  Not assigned    Not assigned

[287 rows x 3 columns]
```

1.0.2 Data Cleaning

[4]:

```
df['Borough'].replace('Not assigned', np.nan, inplace=True)
df.dropna(subset=['Borough'], inplace=True)

df.head()
```

[4]:

```
  Postcode    Borough     Neighbourhood
3      M9W  Etobicoke         Northwest
4      M9V  Etobicoke    Albion Gardens
5      M9V  Etobicoke  Beaumond Heights
6      M9V  Etobicoke        Humbergate
7      M9V  Etobicoke         Jamestown
```

[5]:
```
df = df.groupby(['Postcode', 'Borough'])['Neighbourhood'].apply(', '.join).
 ↪reset_index()
df.columns = ['Postcode', 'Borough', 'Neighbourhood']
df
```

[5]:
```
    Postcode      Borough                                      Neighbourhood
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
..       …            …                                                  …
98       M9N         York                                             Weston
99       M9P    Etobicoke                                          Westmount
100      M9R    Etobicoke  Kingsview Village, Martin Grove Gardens, Richv…
101      M9V    Etobicoke  Albion Gardens, Beaumond Heights, Humbergate, …
102      M9W    Etobicoke                                          Northwest

[103 rows x 3 columns]
```

[6]:
```
df['Neighbourhood'].replace('Not assigned', "Queen's Park", inplace=True)

df
```

[6]:
```
    Postcode      Borough                                      Neighbourhood
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
..       …            …                                                  …
98       M9N         York                                             Weston
99       M9P    Etobicoke                                          Westmount
100      M9R    Etobicoke  Kingsview Village, Martin Grove Gardens, Richv…
101      M9V    Etobicoke  Albion Gardens, Beaumond Heights, Humbergate, …
102      M9W    Etobicoke                                          Northwest

[103 rows x 3 columns]
```

1.0.3 Data shape

[7]:
```
df.shape
```

[7]:
```
(103, 3)
```

[ ]:
```

```

Now that we have built a dataframe of the postal code of each neighborhood along with the borough name and neighborhood name, in order to utilize the Foursquare location data, we need to get the latitude and the longitude coordinates of each neighborhood.

[8]:
```
df_geo = pd.read_csv('http://cocl.us/Geospatial_data')
df_geo.columns = ['Postcode', 'Latitude', 'Longitude']
```

[9]:
```
df_pos = pd.merge(df, df_geo, on=['Postcode'], how='inner')

df_tor = df_pos[['Borough', 'Neighbourhood', 'Postcode', 'Latitude',␣
 ↪'Longitude']].copy()

df_tor.head()
```

[9]:
```
       Borough                           Neighbourhood Postcode   Latitude  \
0  Scarborough                          Rouge, Malvern      M1B  43.806686
1  Scarborough  Highland Creek, Rouge Hill, Port Union      M1C  43.784535
2  Scarborough       Guildwood, Morningside, West Hill      M1E  43.763573
3  Scarborough                                  Woburn      M1G  43.770992
4  Scarborough                               Cedarbrae      M1H  43.773136

   Longitude
0 -79.194353
1 -79.160497
2 -79.188711
3 -79.216917
4 -79.239476
```

[ ]:
```

```

[11]:
```
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

#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't␣
 ↪completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and␣
 ↪longitude values

import requests # library to handle requests
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

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you␣
 ↪haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

[12]:

```
address = 'Toronto, Canada'

geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of the City of Toronto are {}, {}.'.
 ↪format(latitude, longitude))
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:3: DeprecationWarning: Using Nominatim with the
default "geopy/1.20.0" `user_agent` is strongly discouraged, as it violates
Nominatim's ToS https://operations.osmfoundation.org/policies/nominatim/ and may
possibly cause 403 and 429 HTTP errors. Please specify a custom `user_agent`
with `Nominatim(user_agent="my-application")` or by overriding the default
`user_agent`: `geopy.geocoders.options.default_user_agent = "my-application"`.
In geopy 2.0 this will become an exception.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

```
The geograpical coordinate of the City of Toronto are 43.653963, -79.387207.
```

[13]:

```
# create map of New York using latitude and longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(df_tor['Latitude'],␣
 ↪df_tor['Longitude'], df_tor['Borough'], df_tor['Neighbourhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=3,
        popup=label,
        color='green',
        fill=True,
        fill_color='#3199cc',
        fill_opacity=0.3,
        parse_html=False).add_to(map_toronto)

map_toronto
```

```
[13]: <folium.folium.Map at 0x7f24c1349518>
```

```
[22]: CLIENT_ID = 'xxxxxx' # your Foursquare ID
CLIENT_SECRET = 'xxxxxxxxxx' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: xxxxxx
CLIENT_SECRET:xxxxxxxxxx
```

```
[23]: df_t4 = df_tor[df_tor['Borough'].str.contains('Toronto')]

to_data = df_t4.reset_index(drop=True)
to_data
```

```
[23]:
             Borough                                      Neighbourhood  \
0       East Toronto                                        The Beaches
1       East Toronto                       The Danforth West, Riverdale
2       East Toronto                     The Beaches West, India Bazaar
3       East Toronto                                    Studio District
4    Central Toronto                                      Lawrence Park
5    Central Toronto                                   Davisville North
6    Central Toronto                                 North Toronto West
7    Central Toronto                                         Davisville
8    Central Toronto                        Moore Park, Summerhill East
9    Central Toronto  Deer Park, Forest Hill SE, Rathnelly, South Hi…
10  Downtown Toronto                                           Rosedale
11  Downtown Toronto                        Cabbagetown, St. James Town
12  Downtown Toronto                               Church and Wellesley
13  Downtown Toronto                                       Harbourfront
14  Downtown Toronto                           Ryerson, Garden District
15  Downtown Toronto                                     St. James Town
16  Downtown Toronto                                        Berczy Park
17  Downtown Toronto                                 Central Bay Street
18  Downtown Toronto                           Adelaide, King, Richmond
19  Downtown Toronto  Harbourfront East, Toronto Islands, Union Station
20  Downtown Toronto           Design Exchange, Toronto Dominion Centre
```

```
21  Downtown Toronto                     Commerce Court, Victoria Hotel
22   Central Toronto                                           Roselawn
23   Central Toronto                Forest Hill North, Forest Hill West
24   Central Toronto                The Annex, North Midtown, Yorkville
25  Downtown Toronto                     Harbord, University of Toronto
26  Downtown Toronto          Chinatown, Grange Park, Kensington Market
27  Downtown Toronto  CN Tower, Bathurst Quay, Island airport, Harbo… 
28  Downtown Toronto                    Stn A PO Boxes 25 The Esplanade
29  Downtown Toronto             First Canadian Place, Underground city
30  Downtown Toronto                                           Christie
31      West Toronto                       Dovercourt Village, Dufferin
32      West Toronto                           Little Portugal, Trinity
33      West Toronto       Brockton, Exhibition Place, Parkdale Village
34      West Toronto                      High Park, The Junction South
35      West Toronto                             Parkdale, Roncesvalles
36      West Toronto                                 Runnymede, Swansea
37      East Toronto  Business Reply Mail Processing Centre 969 Eastern

   Postcode   Latitude  Longitude
0       M4E  43.676357 -79.293031
1       M4K  43.679557 -79.352188
2       M4L  43.668999 -79.315572
3       M4M  43.659526 -79.340923
4       M4N  43.728020 -79.388790
5       M4P  43.712751 -79.390197
6       M4R  43.715383 -79.405678
7       M4S  43.704324 -79.388790
8       M4T  43.689574 -79.383160
9       M4V  43.686412 -79.400049
10      M4W  43.679563 -79.377529
11      M4X  43.667967 -79.367675
12      M4Y  43.665860 -79.383160
13      M5A  43.654260 -79.360636
14      M5B  43.657162 -79.378937
15      M5C  43.651494 -79.375418
16      M5E  43.644771 -79.373306
17      M5G  43.657952 -79.387383
18      M5H  43.650571 -79.384568
19      M5J  43.640816 -79.381752
20      M5K  43.647177 -79.381576
21      M5L  43.648198 -79.379817
22      M5N  43.711695 -79.416936
23      M5P  43.696948 -79.411307
24      M5R  43.672710 -79.405678
25      M5S  43.662696 -79.400049
26      M5T  43.653206 -79.400049
27      M5V  43.628947 -79.394420
```

```
28      M5W  43.646435 -79.374846
29      M5X  43.648429 -79.382280
30      M6G  43.669542 -79.422564
31      M6H  43.669005 -79.442259
32      M6J  43.647927 -79.419750
33      M6K  43.636847 -79.428191
34      M6P  43.661608 -79.464763
35      M6R  43.648960 -79.456325
36      M6S  43.651571 -79.484450
37      M7Y  43.662744 -79.321558
```

[24]:

```
# create map of Toronto using latitude and longitude values
map_tohood = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(to_data['Latitude'],␣
 ↪to_data['Longitude'], to_data['Borough'], to_data['Neighbourhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=3,
        popup=label,
        color='green',
        fill=True,
        fill_color='#3199cc',
        fill_opacity=0.3,
        parse_html=False).add_to(map_tohood)

map_tohood
```

[24]:

```
<folium.folium.Map at 0x7f24c10536a0>
```

[25]:

```
to_data.loc[0, 'Neighbourhood']
```

[25]:

```
'The Beaches'
```

[26]:

```
neighbourhood_latitude = to_data.loc[0, 'Latitude'] # neighbourhood latitude␣
 ↪value
neighbourhood_longitude = to_data.loc[0, 'Longitude'] # neighbourhood longitude␣
 ↪value

neighbourhood_name = to_data.loc[0, 'Neighbourhood'] # neighbourhood name

print('Latitude and longitude values of {} are {}, {}.'.
 ↪format(neighbourhood_name,
                                                                         ␣
 ↪neighbourhood_latitude,
                                                                         ␣
 ↪neighbourhood_longitude))
```

```
Latitude and longitude values of The Beaches are 43.67635739999999, -79.2930312.
```

[27]:

```
LIMIT = 100
radius = 500

url = 'https://api.foursquare.com/v2/venues/explore?
↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighbourhood_latitude,
    neighbourhood_longitude,
    radius,
    LIMIT)
url
```

[27]:

```
'https://api.foursquare.com/v2/venues/explore?&client_id=xxxxxx&client_secret=xx
xxxxxxxx&v=20180605&ll=43.67635739999999,-79.2930312&radius=500&limit=100'
```

[28]:

```
results = requests.get(url).json()
results
```

[28]:

```
{'meta': {'code': 400,
  'errorType': 'invalid_auth',
  'errorDetail': 'Missing access credentials. See
https://developer.foursquare.com/docs/api/configuration/authentication for
details.',
  'requestId': '5ddae26c47e0d60028e6f302'},
 'response': {}}
```

[29]:

```
# function that extracts the category of the venue
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']
```

[ ]:

```
```