Week_3

October 27, 2021

#First Part

0.1 Use the Notebook to build the code to scrape the following Wikipedia page

[0]:

```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

Scrape the List of postal codes of Canada

[0]:

```
List_url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
source = requests.get(List_url).text
```

[0]:

```
soup = BeautifulSoup(source, 'xml')
table=soup.find('table')
```

[0]:

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

# Search all the postcode, borough, neighborhood
for tr_cell in table.find_all('tr'):
    row_data=[]
    for td_cell in tr_cell.find_all('td'):
        row_data.append(td_cell.text.strip())
    if len(row_data)==3:
        df.loc[len(df)] = row_data
```

[5]:

```
df.head()
```

[5]:

```
  Postalcode           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
```

Now, let us do some data cleaning as required

[0]:

```
# remove rows where Borough is 'Not assigned'

df = df.groupby(['Postalcode', 'Borough'])['Neighborhood'].apply(list).
 ↪apply(lambda x:', '.join(x)).to_frame().reset_index()
```

[0]:

```
temp_df=df.groupby('Postalcode')['Neighborhood'].apply(lambda x: "%s" % ', '.
 ↪join(x))
temp_df=temp_df.reset_index(drop=False)
temp_df.rename(columns={'Neighborhood':'Neighborhood_joined'},inplace=True)
```

[20]:

```
df_merge = pd.merge(df, temp_df, on='Postalcode')
df_merge.drop(['Neighborhood'],axis=1,inplace=True)
df_merge.drop_duplicates(inplace=True)
df_merge.rename(columns={'Neighborhood_joined':'Neighborhood'},inplace=True)
df_merge.head()
```

[20]:

```
  Postalcode      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
```

use the .shape method to print the number of rows of your dataframe

[21]:

```
df.shape
```

[21]:

```
(103, 3)
```

1 Second Part

[0]:

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

Now that you 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.

[0]:

```
def get_geocode(postal_code):
    # initialize your variable to None
    lat_lng_coords = None
    while(lat_lng_coords is None):
        g = geocoder.google('{}, Toronto, Ontario'.format(postal_code))
        lat_lng_coords = g.latlng
    latitude = lat_lng_coords[0]
    longitude = lat_lng_coords[1]
    return latitude,longitude
```

[28]:

```
geo_df=pd.read_csv('http://cocl.us/Geospatial_data')
geo_df.head()
```

[28]:

```
  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
```

[0]:

```
geo_df.rename(columns={'Postal Code':'Postalcode'},inplace=True)
geo_merged = pd.merge(geo_df, df_merge, on='Postalcode')
geo_data=geo_merged[['Postalcode','Borough','Neighborhood','Latitude','Longitude']]
```

[31]:

```
geo_data.head()
```

[31]:

```
  Postalcode      Borough  …   Latitude  Longitude
0        M1B  Scarborough  …  43.806686 -79.194353
1        M1C  Scarborough  …  43.784535 -79.160497
2        M1E  Scarborough  …  43.763573 -79.188711
3        M1G  Scarborough  …  43.770992 -79.216917
4        M1H  Scarborough  …  43.773136 -79.239476

[5 rows x 5 columns]
```

2 Third part

- Explore and cluster the neighborhoods in Toronto.
- Generate maps to visualize your neighborhoods and how they cluster together

[0]:

```
from sklearn.cluster import KMeans
import folium
from geopy.geocoders import Nominatim
import matplotlib.cm as cm
import matplotlib.colors as colors
import geopandas as gpd
import numpy as np
```

[0]:

```
# Getting all the rows from the data frame which contains Toronto in their␣
 ↪Borough
df4 = geo_data[geo_data['Borough'].str.contains('Toronto',regex=False)]
```

[53]:

```
map_toronto = folium.Map(location=[43.651070,-79.347015],zoom_start=10)

for lat,lng,borough,neighbourhood in␣
 ↪zip(df4['Latitude'],df4['Longitude'],df4['Borough'],df4['Neighborhood']):
    label = '{}, {}'.format(neighbourhood, borough)
    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_toronto)
map_toronto
```

[53]:
```
<folium.folium.Map at 0x7ff632610048>
```

Using KMeans to cluster the neighbourhoods

[0]:
```
k=5
toronto_clustering = df4.drop(['Postalcode','Borough','Neighborhood'],1)
kmeans = KMeans(n_clusters = k,random_state=0).fit(toronto_clustering)
kmeans.labels_
df4.insert(0, 'Cluster Labels', kmeans.labels_)
```

Now visulaizing the clustering map

[59]:
```
map_clusters = folium.Map(location=[43.651070,-79.347015],zoom_start=10)

# set color scheme for the clusters
x = np.arange(k)
ys = [i + x + (i*x)**2 for i in range(k)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, neighbourhood, cluster in zip(df4['Latitude'], df4['Longitude'],␣
 ↪df4['Neighborhood'], df4['Cluster Labels']):
    label = folium.Popup(' 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
```

```
[59]: <folium.folium.Map at 0x7ff632589dd8>
```