Clustering Assignment

Requirements

For this assignment, you will be required to explore and cluster the neighborhoods in Toronto.

Start by creating a new Notebook for this assignment. Use the Notebook to build the code to scrape the following Wikipedia page, https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M), in order to obtain the data that is in the table of postal codes and to transform the data into a pandas dataframe like the one shown below:

Preprocessing

In [2]:
```
# importing necessary libraries
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import requests
```

In [3]:
```
# getting data from internet
wikipedia_link='https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
raw_wikipedia_page= requests.get(wikipedia_link).text

# using beautiful soup library to parse the HTML/XML codes.
soup = BeautifulSoup(raw_wikipedia_page,'xml')
#print(soup.prettify())
```

Processing-part-1: extracting raw table (from webpage)

In [4]:
```
# extracting the raw data into table inside that webpage
table = soup.find('table')

Postcode      = []
Borough       = []
Neighbourhood = []


#cleaning and extracting a clean form of the table
for tr_cell in table.find_all('tr'):

    counter = 1
    Postcode_var      = -1
    Borough_var       = -1
    Neighbourhood_var = -1

    for td_cell in tr_cell.find_all('td'):
        if counter == 1:
            Postcode_var = td_cell.text
        if counter == 2:
            Borough_var = td_cell.text
            tag_a_Borough = td_cell.find('a')

        if counter == 3:
            Neighbourhood_var = str(td_cell.text).strip()
            tag_a_Neighbourhood = td_cell.find('a')

        counter +=1

        if (Postcode_var == 'Not assigned' or Borough_var == 'Not assigned' or
Neighbourhood_var == 'Not assigned'):

            continue


    try:
        if ((tag_a_Borough is None) or (tag_a_Neighbourhood is None)):

            continue


    except:

        pass

    if(Postcode_var == -1 or Borough_var == -1 or Neighbourhood_var == -1):

        continue


    Postcode.append(Postcode_var)
    Borough.append(Borough_var)
    Neighbourhood.append(Neighbourhood_var)
```

Processing-part-2: integrating Postal codes with more than 1 neighbour

In [5]:
```
# Processing the data for unique postal codes
unique_p = set(Postcode)
print('num of unique Postal codes:', len(unique_p))
Postcode_u      = []
Borough_u       = []
Neighbourhood_u = []


for postcode_unique_element in unique_p:
    p_var = ''; b_var = ''; n_var = '';
    for postcode_idx, postcode_element in enumerate(Postcode):
        if postcode_unique_element == postcode_element:
            p_var = postcode_element;
            b_var = Borough[postcode_idx]
            if n_var == '':
                n_var = Neighbourhood[postcode_idx]
            else:
                n_var = n_var + ', ' + Neighbourhood[postcode_idx]
    Postcode_u.append(p_var)
    Borough_u.append(b_var)
    Neighbourhood_u.append(n_var)
```

```
num of unique Postal codes: 77
```

Post-processing: creating an appropriate Pandas Dataframe

In [6]:
```
#printing the Toronto district postal codes
toronto_dict = {'Postcode':Postcode_u, 'Borough':Borough_u, 'Neighbourhood':Ne
ighbourhood_u}
df_toronto = pd.DataFrame.from_dict(toronto_dict)
df_toronto.to_csv('toronto_part1.csv')
df_toronto.head(14)
```

Out[6]:

	Postcode	Borough	Neighbourhood
0	M9M	North York	Emery, Humberlea
1	M2R	North York	Willowdale West
2	M4N	Central Toronto	Lawrence Park
3	M6J	West Toronto	Little Portugal, Trinity
4	M4V	Central Toronto	Deer Park, Rathnelly, South Hill
5	M4B	East York	Woodbine Gardens, Parkview Hill
6	M5A	Downtown Toronto	Harbourfront, Regent Park
7	M5S	Downtown Toronto	University of Toronto
8	M8Z	Etobicoke	Mimico NW, The Queensway West
9	M5J	Downtown Toronto	Toronto Islands, Union Station
10	M2M	North York	Newtonbrook, Willowdale
11	M6L	North York	Downsview
12	M1S	Scarborough	Agincourt
13	M4C	East York	Woodbine Heights

In [7]:
```
# looking at the shape of Toronto
df_toronto.shape
```

Out[7]: (77, 3)

In [ ]:

In [ ]: