

Week Three Peer Review Assignment

November 18, 2019

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

import pandas as pd # library for data analysis
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.')
```

Solving environment: done

```
==> WARNING: A newer version of conda exists. <==
current version: 4.5.11
latest version: 4.7.12
```

Please update conda by running

```
$ conda update -n base -c defaults conda
```

```
# All requested packages already installed.
```

```
Libraries imported.
```

```
[4]: pip install lxml
```

```
Collecting lxml
```

```
  Downloading https://files.pythonhosted.org/packages/ec/be/5ab8abdd8663c0386ec2dd595a5bc0e23330a0549b8a91e32f38c20845b6/lxml-4.4.1-cp36-cp36m-manylinux1_x86_64.whl (5.8MB)
```

```
|                                     | 5.8MB 25.6MB/s eta 0:00:01
```

```
Installing collected packages: lxml
```

```
Successfully installed lxml-4.4.1
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
[19]: #Fetching the Data from the webpage
url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
toronto_data = pd.read_html(url, header=0)
```

```
[20]: #Converting the data into a pandas dataframe
toronto_data = toronto_data[0]
toronto_data.head()
```

```
[20]:
```

	Postcode	Borough	Neighbourhood
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

```
[25]: #Dropping the Boroughs which are 'Not Assigned'
neighborhood_data = toronto_data[toronto_data.Borough != 'Not assigned']
neighborhood_data.reset_index(drop = True, inplace = True)
neighborhood_data.head()
```

```
[25]:
```

	Postcode	Borough	Neighbourhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Harbourfront
3	M6A	North York	Lawrence Heights
4	M6A	North York	Lawrence Manor

In place of the above code, we can also use the following `neighborhood_data = toronto_data.set_index("Borough") neighborhood_data.drop("Not assigned") neighborhood_data.reset_index(drop = True, inplace = True) neighborhood_data.head()` #to view the first five observations

```
[33]: #Combining the neighbourhoods of different coastal areas in one row separated
      ↪by comma
neighborhood_data = neighborhood_data.
      ↪groupby(['Postcode', 'Borough'])['Neighbourhood'].apply(', '.join)
neighborhood_data = neighborhood_data.reset_index(level =
      ↪['Postcode', 'Borough'])
neighborhood_data.head()
```

```
[33]:   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
```

```
[39]: #Here, we want to check the Neighbourhoods with the value 'Not assigned'
neighborhood_data[neighborhood_data.Neighbourhood == 'Not assigned']
```

```
[39]:   Postcode      Borough Neighbourhood
85      M7A  Queen's Park  Not assigned
```

```
[41]: #For the Neighbourhood with the value 'Not Assigned', we name the Neighbourhood
      ↪with it's 'Borough' name
#and then check is there is still any Neighbourhood with a 'Not assigned'
      ↪observation.
#None exist anymore
neighborhood_data.loc[(neighborhood_data.Neighbourhood == 'Not assigned'),
      ↪'Neighbourhood'] = neighborhood_data.Borough
neighborhood_data[neighborhood_data.Neighbourhood == 'Not assigned']
```

```
[41]: Empty DataFrame
Columns: [Postcode, Borough, Neighbourhood]
Index: []
```

```
[42]: #This code the verifies if the Neighbourhood observation of 'Not assigned' on
      ↪Postcode 'M7A' has been
#Replaced by the Borough name 'Queen's Park'
neighborhood_data[neighborhood_data.Postcode == 'M7A']
```

```
[42]:   Postcode      Borough Neighbourhood
85      M7A  Queen's Park  Queen's Park
```

```
[44]: #The shape of the data is 103 observations and 3 Variables  
neighborhood_data.shape
```

```
[44]: (103, 3)
```

```
[ ]:
```