**Capstone project: Segmenting and Clustering Neighborhoods in Toronto City**

**Description of the problem:**

In this capstone project I will be working with Toronto Ontario city geo data and will be exploring the segmenting and Clustering function to get the most common venue categories in each neighborhood, and then use this feature to group the neighborhoods into clusters.

Throughout the course we learnt various techniques to convert addresses into their equivalent latitude and longitude values. Also, will use the Foursquare API to explore neighborhoods in Toronto Ontario City.

**Discussion of the background:**

In Module 3, we explored New York City and the city of Toronto and segmented and clustered their neighborhoods. Both cities are very diverse and are the financial capitals of their respective countries.

So, I would be interested in comparing the neighborhoods of the Toronto cities and determine how similar or dissimilar they are.

I will use the k-means clustering algorithm to complete this task. Finally, using the Folium library to visualize the neighborhoods in New York City and their emerging clusters.

**Description of the data:**

# Web scraping for Toronto neighborhood and build a clean data frame:

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>, in order to obtain the data that is in the table of postal codes and to transform the data into a pandas dataframe.

* The dataframe will consist of three columns: PostalCode, Borough, and Neighborhood
* Only process the cells that have an assigned borough. Ignore cells with a borough that is Not assigned.
* More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that M5A is listed twice and has two neighborhoods: Harbourfront and Regent Park. These two rows will be combined into one row with the neighborhoods separated with a comma.
* If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough. So for the 9th cell in the table on the Wikipedia page, the value of the Borough and the Neighborhood columns will be Queen's Park.
* Clean your Notebook and add Markdown cells to explain your work and any assumptions you are making.
* In the last cell of your notebook, use the .shape method to print the number of rows of your dataframe.
* **Note:** There are different website scraping libraries and packages in Python. One of the most common packages is BeautifulSoup. Here is the package's main documentation page: <http://beautiful-soup-4.readthedocs.io/en/latest/>
* The data we want is in a table, with 3 columns PostalCode, Borough and Neighborhood.
* The table contains a list of postal codes in Canada where the first letter is M. Postal codes beginning with M are located within the city of Toronto in the province of Ontario. Only the first three characters are listed, corresponding to the Forward Sortation Area.
* Group neighborhoods by postal and borough
* There are some neighborhoods that belongs to the same postal code and borough (Ex: M5A Downtown Toronto has 2 neighborhoods Harbourfront and Regent Park). We will concat them into same row, seperated by a colon.
* Deal with Not assigned Neighborhood
* For M7A Queen's Park, there is no neighborhood assigned. We will replace the 'Not assigned' with the value of the corresponding Borough

**How the data will be used to solve the problem:**

Part 2: Getting coordinates and add to the Toronto DataFrame

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.

The problem with this Package is you must be persistent sometimes in order to get the geographical coordinates of a given postal code. So you can make a call to get the latitude and longitude coordinates of a given postal code and the result would be None, and then make the call again and you would get the coordinates.

geolocator = Nominatim(user\_agent="tl-toronto-neigh")

postalList = toronto\_df\_cleaned['PostalCode'].values latList = [] longList = []

for post in postalList: location = None while(location is None): location = geolocator.geocode('{}, Toronto, Ontario'.format(post)) if(location != None): lat = location.latitude long = location.longitude print(post, lat, long) latList.append(lat) longList.append(long)

toronto\_coors = [('Postal Code', postalList), ('Latitude', latList), ('Longitude', longList)] coors = pd.DataFrame.from\_items(toronto\_coors)

**Note:** Given that this package can be very unreliable, in case you are not able to get the geographical coordinates of the neighborhoods using the Geocoder package, here is a link to a csv file that has the geographical coordinates of each postal code: <http://cocl.us/Geospatial_data>

The csv file format has:

3 columns: Postal Code, Latitude and Longitude

103 rows: corresponding to 103 postal codes in our toronto dataframe

**Methodology:**

* Explore and cluster the neighborhoods in Toronto Ontario city geo data and will be exploring the segmenting and Clustering function to get the most common venue categories in each neighborhood, and then use this feature to group the neighborhoods into clusters.
* Throughout the course we learnt various techniques to convert addresses into their equivalent latitude and longitude values. Also, will use the Foursquare API to explore neighborhoods in Toronto Ontario City.
* I will use the k-means clustering algorithm to complete this task. Finally, using the Folium library to visualize the neighborhoods in New York City and their emerging clusters.

**Results**:

* Using the Ontario latitude and longitude values the we can see that how many similar areas are there and then I reduce the number of Boroughs to explore To reduce the numbers of calls to Foursquare API, we will only explore boroughs that have Toronto in their names.
* With the use of Foursquare API, I explore the Boroughs. And filtered the data to have values only for ['East Toronto', 'Central Toronto', 'Downtown Toronto', 'West Toronto']
* Also checked the frequency of occurrence of each category in an area and then Get 10 most occurrence venue types in each area: just to check the most crowed places.
* Using K means clustering divided the central Toronto area in 4 cluster.

**Conclusion:**

* Cluster 0 (Red): Living area (with mostly park, trail, school, and some small businesses)
* Cluster 1 (Yellow): Roselawn - Central Toronto (nothing here except a garden)
* Cluster 2 (Purple): Business area (with lots of business venues)
* Cluster 3 (Blue): Oldest neighborhood with lot of nature park.