IBM Coursera Capstone

Toronto Neighborhood Analysis

IBM Applied Data Science Capstone

By

Belachew Ayele

January 2021

Table of Content **Page**

1. Introduction

This is a project to conclude the IBM Data science courses. The project is facilitated with the data from the wiki, and Foursquare API data though a personal account. Python machine learning and use of several steps-by-step process is applied.

The data that involved in this project is focusing on Toronto area neighborhood. It includes many attributes including Toronto Neighborhoods, borough, geographic coordinates and other attributes that at revealed thought Foursquare API data acquire.

In this process, data is download, cleaned, processes, mapped and clustered.

2. Business Problem

The business problem is to get data and classify them into cluster using k-means. How data are acquired, processed and clustered, and will answer the type and classes of the foursquare data of Toronto and surrounding area. Data users may lean and use the method and results of this process.

3. Objective / Target Audience of the project

With the given set of objectives, this neighborhood has been implemented for the exploration of data using segmenting and clustering techniques applied to the neighborhoods data in Toronto. The objective of Applied Data Science Capstone is given as follows:

1. To learn about clustering and k-means clustering in particular.
2. To showcase this project in the form of the public repository using the GitHub platform.
3. To learn how to use the Foursquare API and clustering to segment and cluster the neighborhoods in Toronto City.
4. To learn how to use different Python package to scrape websites and parse HTML code.
5. To apply the skills acquired so far in this course to segment and cluster neighborhoods in the city of Toronto.

Target Audience of the project

This project is the project that can benefit several in the process,

- Any busines that will start work in Toronto and surrounding
- Personnel that are interested in studies and application of python processing such as students, practitioners, researches, professors, entrepreneurs and so on.
- Personal benefit in increasing data sciences knowledge in honing our knowledge.
- General practice in data science application

4. Data

Based on the problem statement, data sources are of three sources,

These are:

4.1. **Data from wiki**. The following data source of data will be needed to extract/generate to require information **https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M**
This data is on the wiki with Toronto Postal code, Boroughs and Neighborhoods. As this was the data in the wiki, it is not having all the records with complete information. Some of the lines do not have Borough assigned. In such case the recorded gets neglected, however when the borough is assigned and Neighborhood not assigned, the Borough name was given as the Neighborhood. It was captured and cleaned using different of python libraries.

The following is the snip of the extracting script

```
In [11]:
source = requests.get("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M").text
soup = BeautifulSoup(source, 'lxml')

table = soup.find("table")
table_rows = table.tbody.find_all("tr")

res = []
for tr in table_rows:
    td = tr.find_all("td")
    row = [tr.text for tr in td]

    # Only process the cells that have an assigned borough. Ignore cells with a borough that is Not assigned.
    if row != [] and row[1] != "Not assigned\n":
        # If a cell has a borough but a "Not assigned" neighborhood, then the neighborhood will be the same as the borough.
        if "Not assigned\n" in row[2]:
            row[2] = row[1]
        res.append(row)

# Dataframe with 3 columns
toronto_data = pd.DataFrame(res, columns = ["PostalCode", "Borough", "Neighbourhood"])

# Remove "\n" at the end of each string in the columns
toronto_data["PostalCode"] = toronto_data["PostalCode"].str.replace("\n","")
toronto_data["Borough"] = toronto_data["Borough"].str.replace("\n","")
toronto_data["Neighbourhood"] = toronto_data["Neighbourhood"].str.replace("\n","")

toronto_data.head()
```

Here is the partial data that is produced for the above script,

Out[11]:

	PostalCode	Borough	Neighbourhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Regent Park, Harbourfront
3	M6A	North York	Lawrence Manor, Lawrence Heights
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

In the above, 103 records were collected

4.2. **Geo Data**: Centers of candidate areas will be generated algorithmically and approximate addresses of centers of those areas' coordinates will be obtained using **Geocoder.** https://cocl.us/Geospatial_data
Postal code coordinates are available in the geocoder site and were grabbed using the python command.

```
In [12]:
## Get the coordinates of the data

geo_data = pd.read_csv("https://cocl.us/Geospatial_data")

geo_data.head()
#print(geo_data.shape)
```

Here are the patial coodinate data of the Postal codes

Out[12]:

	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

However, the Postal code, Borough and Neighborhood data should get the coordinates from the geocoder data, by joining the two data sets using the postal code names, and drop the additional postal code.
This is the script used to join the two files and drop the additional postal code,

In [15]:
```
#We need to couple 2 dataframes "df" and "df_geo_coor" into one dataframe.


df_toronto = pd.merge(toronto_data, geo_data, how='left', left_on = 'PostalCode', right_on = 'Postal Code')
# remove the "Postal Code" column
df_toronto.drop("Postal Code", axis=1, inplace=True)
df_toronto.head()
```

Here are the patial view of the Poctal code, Boough, Negibourhoud and the latitude and longitude coordiates

Out[15]:

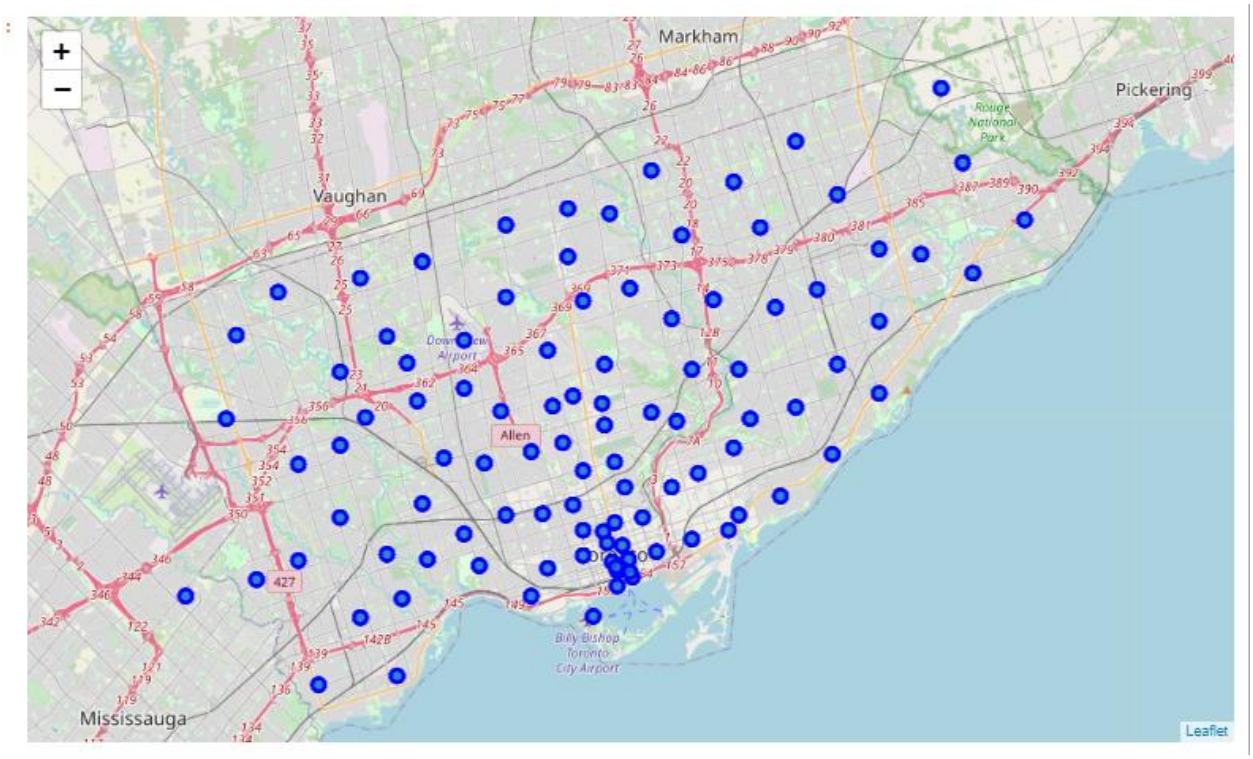
	PostalCode	Borough	Neighbourhood	Latitude	Longitude
0	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

As it is called, seeing is believing, above are going to be see which areas they cover on the maps, a folium app command was used to create a map.

This is the command script for the mapping of the joined result

In [32]:
```
# Mapping the torinto are and add the locaiaons

# get Toronto cooridntase
address = "Toronto, ON"

geolocator = Nominatim(user_agent="toronto_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Toronto city are {}, {}.'.format(latitude, longitude))


# map data on Tototo area
for lat, lng, borough, neighbourhood in zip(
        df_toronto['Latitude'],
        df_toronto['Longitude'],
        df_toronto['Borough'],
        df_toronto['Neighbourhood']):
    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
```

Map of the joined files show all the record of 103, with postal code, borough, Neighborhood, longitude and latitude.



4.3. **Foursquare Data**. Number of restaurants and their type and location in every neighborhood will be obtained using **Foursquare API**

One of the methods of getting the venue data is getting on the Foursquare option and get data by the type needed. Here in this project, we use the venue to get data surrounding the above mention table records. But to begin with the data extraction through foursquare, one has to get register and get a development permission. Then getting data form the foursquare API is possible.

After registering, the following credential were obtained.

5.1. Define Foursquare Credentials and Version

```
CLIENT_ID = 'XXXXXXXXXXXXXXXX' # hiding for secutity , you can add your oun credentials
CLIENT_SECRET= 'XXXXXXXXXXXXX' # hiding for secutity , you can add your oun credentials
VERSION = 20202612
radius = 500
LIMIT = 100
```

This is a script of extract data from the Foursquare API

```
# Write a python function

#Part 1 creating the API request  URL

def getNearbyVenues (names, latitudes, longitudes):
    venues_list =[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)


#part 2: making the Request

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)
        results = requests.get(url).json()["response"]['groups'][0]['items']

#part 3 returning only relevenat  information for each nearby venueand append to the list

        venues_list.append ([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    return (venues_list)
```

The foursquare process extracts 2129 data points with the covers that radius of 500 and limit of a 100. After steps of renaming the field names, below is the sample of a table. This table used in the analysis process to determine the next steps

Out[16]:

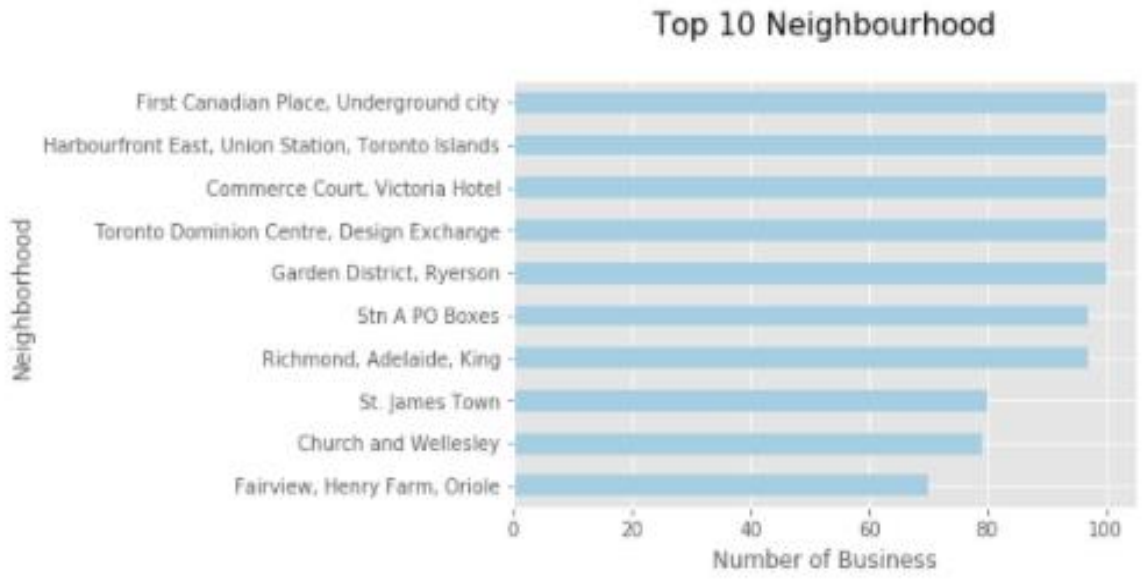
	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venuse	Venue Latitude	Venue Longitude	Venue Category
0	Malvern, Rouge	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurant
1	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	Royal Canadian Legion	43.782533	-79.163085	Bar
2	Guildwood, Morningside, West Hill	43.763573	-79.188711	RBC Royal Bank	43.766790	-79.191151	Bank
3	Guildwood, Morningside, West Hill	43.763573	-79.188711	G & G Electronics	43.765309	-79.191537	Electronics Store
4	Guildwood, Morningside, West Hill	43.763573	-79.188711	Sail Sushi	43.765951	-79.191275	Restaurant
...	...	...	...	...	...	...	...
2124	South Steeles, Silverstone, Humbergate, Jamest...	43.739416	-79.588437	Pizza Nova	43.736761	-79.589817	Pizza Place
2125	South Steeles, Silverstone, Humbergate, Jamest...	43.739416	-79.588437	Rogers Plus	43.741312	-79.585263	Video Store
2126	Northwest, West Humber - Clairville	43.706748	-79.594054	Economy Rent A Car	43.708471	-79.589943	Rental Car Location
2127	Northwest, West Humber - Clairville	43.706748	-79.594054	Saand Rexdale	43.705072	-79.598725	Drugstore
2128	Northwest, West Humber - Clairville	43.706748	-79.594054	Vectra Heavy Haulers	43.704891	-79.599410	Truck Stop

In data exploration analysis is one step needed to be accomplished. Here down are the top and bottom to neighborhood with the number of businesses in them.

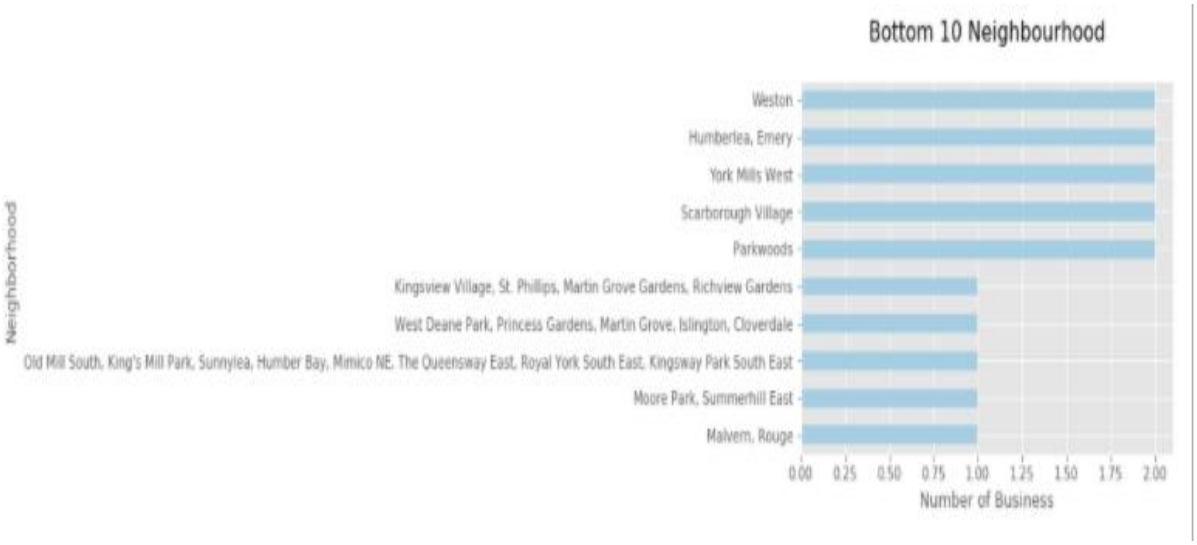
Counting the number of businesses (venues) in neighborhood that are in the top ten.

1. First Canadian Place, Underground City
2. Harbourfront East, Union Station, Toronto Islands
3. Commerce Court, Victoria Hotel
4. Toronto Dominion Centre
5. Garden District, Ryerson

Top 10 Neighborhood with number of business



Bottom 10 Neighborhood with number of business



5. Integrating data

This is the portion where the data from the wiki and the data acquired through four square API get integrated. Data should be integrated based on common characteristics or feature to join to one file. The coordinate will be shared/transferred from wiki data acquired through the API.

6. Analysis of Data

6.1. **Pre-Clustering** – this is preparing the data for clustering
Data acquired through the Foursquare method should go through a step-by-step process to be analyzed. These steps include

- Creating one hot file,
- Grouping the data
- Get the top five most common venues for teach neighborhood

Create onehot file - this is a process that creates a data frame that is testing for each neighbor and assign a zero or 1 based on the exitance of the venue in each Neighborhood with "1" = "exist" and "0" = "does not exist". It is needed for the clustering of the file/data fame in to cluster categories. The script and the result table are in the snippets below. The one hot table is consisted of 2129 records and 274 features (a feature in here is the number of cases such as Adult Boutique, Airport, Airport food court, etc.)

```
18]:
# one hot encoding
toronto_onehot = pd.get_dummies(Totonto_nearby_v[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
toronto_onehot['Neighbourhood'] = Totonto_nearby_v['Neighbourhood']

# move neighborhood column to the first column
fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]

toronto_onehot.head()
```

This is partial one hot table

318]:

	Neighbourhood	Adult Boutique	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	...	Theme Restaurant	R
0	The Beaches	0	0	0	0	0	0	0	0	0	...	0	
1	The Beaches	0	0	0	0	0	0	0	0	0	...	0	
2	The Beaches	0	0	0	0	0	0	0	0	0	...	0	
3	The Beaches	0	0	0	0	0	0	0	0	0	...	0	
4	The Beaches	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 233 columns

Grouping the data - the above data which is the product on the one hot was grouped by neighborhood and by the mean of the frequency of occurrence of each category. This produced 96 Neighborhoods (as rows) and 274 columns. The simple scrip and the table as in the snippets below

```
3]:
toronto_grouped = toronto_onehot.groupby('Neighbourhood').mean().reset_index()
toronto_grouped
```

This is the partial table result using the group by Neighborhood and get means of occurrents

353]:

	Neighbourhood	Adult Boutique	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	...	Rest
0	Berczy Park	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
1	Brockton, Parkdale Village, Exhibition Place	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
2	Business reply mail Processing Centre, South C...	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
3	CN Tower, King and Spadina, Railway Lands, Har...	0.0000	0.058824	0.058824	0.058824	0.117647	0.176471	0.117647	0.000000	0.000000	...	

Get the top five most common venues for teach neighborhood- this process extracts the top 10 common venues for each neighborhood and lists them in decreasing order. So, the 1st to 10th common venues table is in the table below

```
44]:  # Getting the first 10 common venues
      num_top_venues = 10

      indicators = ['st', 'nd', 'rd']

      # create columns according to number of top venues
      columns = ['Neighbourhood']
      for ind in np.arange(num_top_venues):
          try:
              columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
          except:
              columns.append('{}th Most Common Venue'.format(ind+1))

      # create a new dataframe
      neighbourhoods_venues_sorted = pd.DataFrame(columns=columns)
      neighbourhoods_venues_sorted['Neighbourhood'] = toronto_grouped['Neighbourhood']

      for ind in np.arange(toronto_grouped.shape[0]):
          neighbourhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(toronto_grouped.iloc[ind, :], num_top_venues)

      neighbourhoods_venues_sorted.head()
```

This is the partial top 10 common venue table of Neighborhoods

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Berczy Park	Coffee Shop	Café	Park	Hotel	Restaurant	Japanese Restaurant	Gastropub	Art Gallery	Creperie	Bakery
1	Brockton, Parkdale Village, Exhibition Place	Café	Restaurant	Coffee Shop	Bar	Bakery	Furniture / Home Store	Tibetan Restaurant	Gift Shop	Supermarket	Sandwich Place
2	Business reply mail Processing Centre, South C...	Park	Coffee Shop	Pizza Place	Brewery	Pet Store	Fast Food Restaurant	Sushi Restaurant	Italian Restaurant	Electronics Store	Burrito Place
3	CN Tower, King and Spadina, Railway Lands, Har...	Café	Harbor / Marina	Coffee Shop	Scenic Lookout	Dog Run	Airport	Airport Lounge	Sculpture Garden	Dance Studio	Garden
4	Central Bay Street	Coffee Shop	Ramen Restaurant	Café	Park	Hotel	Diner	Burrito Place	Juice Bar	Mexican Restaurant	Japanese Restaurant

6.2. Clustering

This is classifying the data into clusters.

The ultimate and main goal of this process int to get the clustered output of all the cases into different categories. However, after a thorough process of download, cleaning, join and other critical data processing, now is the time to go through running clustering algorithm or machine learning. To do that, it is usually preceded by determining how many clusters should be assigned? To do this a K-means clustering should be run.

Therefore, the following script is run to figure out how many classes were needed to cluster the data file

This is the scrip for K-means determination

```
# import k-means from clustering stage
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors
import matplotlib.pyplot as plt
```

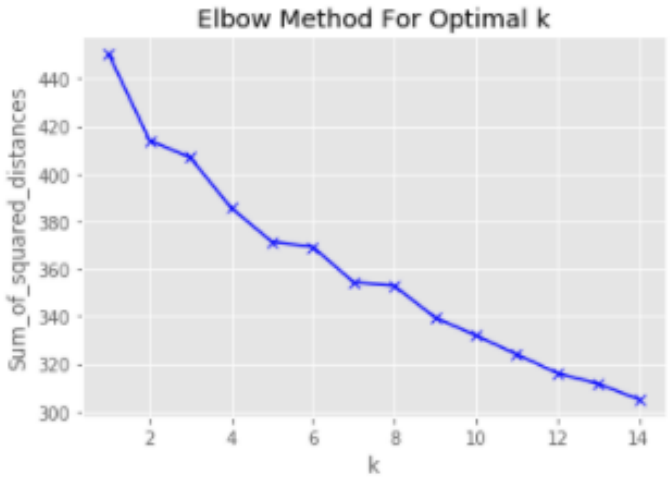
```
# scale the continuous features

toronto_grouped_clustering = toronto_grouped.drop('Neighbourhood', 1)

mms = MinMaxScaler()
mms.fit(toronto_grouped_clustering)
toronto_grouped_clustering_transformed = mms.transform(toronto_grouped_clustering)
```

```
#toronto_grouped_clustering
```

```
Sum_of_squared_distances = []
K = range(1,15)
for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(toronto_grouped_clustering_transformed)
    Sum_of_squared_distances.append(km.inertia_)
```

```
plt.plot(K, Sum_of_squared_distances, 'bx-')
plt.xlabel('k')
plt.ylabel('Sum_of_squared_distances')
plt.title('Elbow Method For Optimal k')
plt.show()
```

This is the K-means, number of K for clustering



The K- means, roughly was determine where the elbow is curing up a lot (in this case it is hard to determine that but had been taken to be are 7). Note that every run gives a different curve. So, rerunning test again may differ the graph and the result.

- Cluster running

  Based on the number estimated in the k-means, 7 clusters were used in creating the cluster by tunning the script below.

```
In [97]:  # add clustering labels

          neighbourhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

          toronto_merged = df_toronto

          # merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
          toronto_merged = toronto_merged.join(neighbourhoods_venues_sorted.set_index('Neighbourhood'), on='Neighbourhood')

          neighbourhoods_venues_sorted.head() # check the last columns!
```

Running the above, produced the following clusters:
(Cluster 0. Custer 1, Cluster 2, Cluster 3, Cluster 4, Cluster 5, Cluster 6

Cluster 0

	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Com V
0	M3A	Parkwoods	43.753259	-79.329656	0.0	Food & Drink Shop	Park	Yoga Studio	Diner	Discount Store	Distribution Center	Dog Run	D Resta
21	M6E	Caledonia-Fairbanks	43.689026	-79.453512	0.0	Park	Women's Store	Drugstore	Diner	Discount Store	Distribution Center	Dog Run	D Resta
35	M4J	East Toronto, Broadview North (Old East York)	43.685347	-79.338106	0.0	Park	Convenience Store	Drugstore	Diner	Discount Store	Distribution Center	Dog Run	D Resta
64	M9N	Weston	43.706876	-79.518188	0.0	Park	Jewelry Store	Yoga Studio	Drugstore	Discount Store	Distribution Center	Dog Run	D Resta
66	M2P	York Mills West	43.752758	-79.400049	0.0	Convenience Store	Park	Drugstore	Diner	Discount Store	Distribution Center	Dog Run	D Resta
85	M1V	Milliken, Agincourt North, Steeles East, L'Amo...	43.815252	-79.284577	0.0	Playground	Park	Intersection	Yoga Studio	Dog Run	Dessert Shop	Dim Sum Restaurant	
91	M4W	Rosedale	43.679563	-79.377529	0.0	Park	Playground	Trail	Yoga Studio	Doner Restaurant	Dim Sum Restaurant	Diner	Dis

Cluster 1

	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8 C
1	M4A	Victoria Village	43.725882	-79.315572	1.0	French Restaurant	Coffee Shop	Portuguese Restaurant	Hockey Arena	Yoga Studio	Donut Shop	Diner	
2	M5A	Regent Park, Harbourfront	43.654260	-79.360636	1.0	Coffee Shop	Park	Bakery	Café	Breakfast Spot	Pub	Theater	Yog
3	M6A	Lawrence Manor, Lawrence Heights	43.718518	-79.464763	1.0	Clothing Store	Accessories Store	Vietnamese Restaurant	Boutique	Furniture / Home Store	Coffee Shop	Event Space	Con Lan
4	M7A	Queen's Park, Ontario Provincial Government	43.662301	-79.389494	1.0	Coffee Shop	Sushi Restaurant	Yoga Studio	Smoothie Shop	Burrito Place	Sandwich Place	Café	Po Re
6	M1B	Malvern, Rouge	43.806686	-79.194353	1.0	Fast Food Restaurant	Donut Shop	Dim Sum Restaurant	Diner	Discount Store	Distribution Center	Dog Run	Re
...	...	...	...	...	...	...	...	...	...	...	...	...	
96	M4X	St. James Town, Cabbagetown	43.667967	-79.367675	1.0	Restaurant	Pizza Place	Coffee Shop	Bakery	Park	Café	Pet Store	Re
97	M5X	First Canadian Place, Underground city	43.648429	-79.382280	1.0	Coffee Shop	Café	Hotel	Japanese Restaurant	Restaurant	Gym	Seafood Restaurant	Re
99	M4Y	Church and Wellesley	43.665860	-79.383160	1.0	Coffee Shop	Sushi Restaurant	Japanese Restaurant	Gay Bar	Restaurant	Fast Food Restaurant	Yoga Studio	
100	M7Y	Business reply mail Processing Centre, South C...	43.662744	-79.321558	1.0	Light Rail Station	Gym / Fitness Center	Garden	Comic Shop	Pizza Place	Restaurant	Burrito Place	Sk
102	M8Z	Mimico NW, The Queensway West, South of Bloor,...	43.628841	-79.520999	1.0	Grocery Store	Convenience Store	Fast Food Restaurant	Kids Store	Burger Joint	Sandwich Place	Supplement Shop	

Cluster 2

	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
57	M9M	Humberlea, Emery	43.724766	-79.532242	2.0	Construction & Landscaping	Baseball Field	Yoga Studio	Eastern European Restaurant	Distribution Center	Dog Run	Doner Restaurant	Donut Shop
101	M8Y	Old Mill South, King's Mill Park, Sunnylea, Hu...	43.636258	-79.498509	2.0	Baseball Field	Yoga Studio	Drugstore	Discount Store	Distribution Center	Dog Run	Doner Restaurant	Donut Shop

Cluster 3

	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Cor
83	M4T	Moore Park, Summerhill East	43.689574	-79.38316	3.0	Summer Camp	Yoga Studio	Drugstore	Diner	Discount Store	Distribution Center	Dog Run	Doner Restaurant	

Cluster 4

	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
57	M9M	Humberlea, Emery	43.724766	-79.532242	4.0	Construction & Landscaping	Baseball Field	Yoga Studio	Eastern European Restaurant	Distribution Center	Dog Run	Doner Restaurant	Donut Shop
101	M8Y	Old Mill South, King's Mill Park, Sunnylea, Hu...	43.636258	-79.498509	4.0	Baseball Field	Yoga Studio	Drugstore	Discount Store	Distribution Center	Dog Run	Doner Restaurant	Donut Shop

Cluster 5

	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
77	M9R	Kingsview Village, St. Phillips, Martin Grove ...	43.688905	-79.554724	5.0	Sandwich Place	Yoga Studio	Dessert Shop	Event Space	Ethiopian Restaurant	Escape Room	Electronics Store	Eastern European Restaurant

Cluster 6

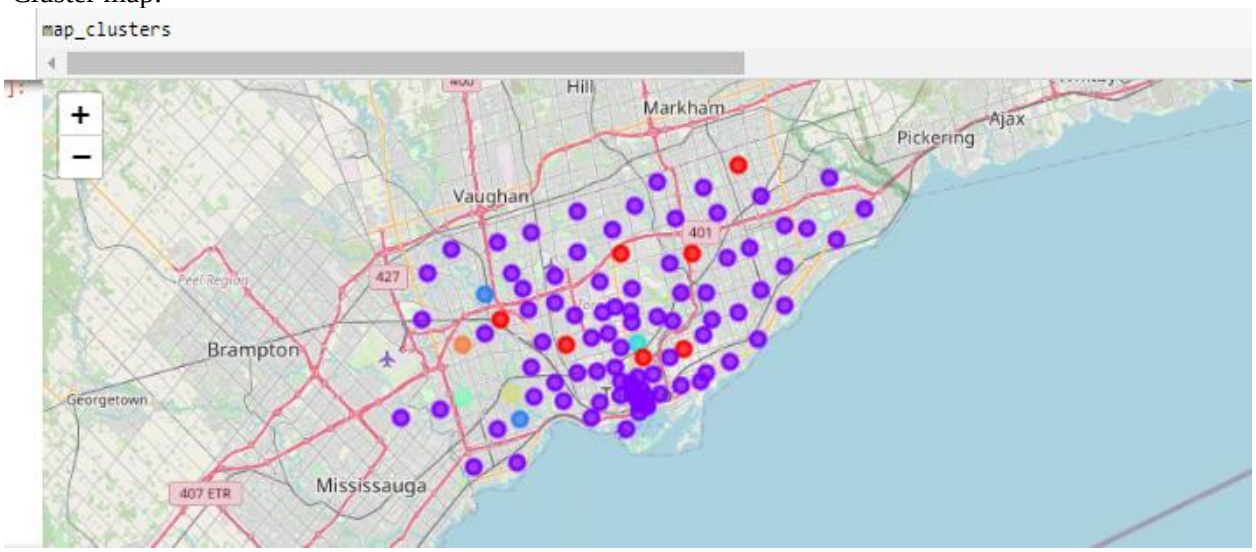
	PostalCode	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9 C
11	M9B	West Deane Park, Princess Gardens, Martin Grov...	43.650943	-79.554724	6.0	Filipino Restaurant	Yoga Studio	Drugstore	Diner	Discount Store	Distribution Center	Dog Run	Doner Restaurant	

7. Result and findings

Cluster classification was run using the data acquired and through Foursquare API and processed for one hot and extracted the top 10 venues in each neighborhood. A test for the K- means value and cluster was performed, followed by clustering the data. A clustering to 7 class as run and the following result was extracted.

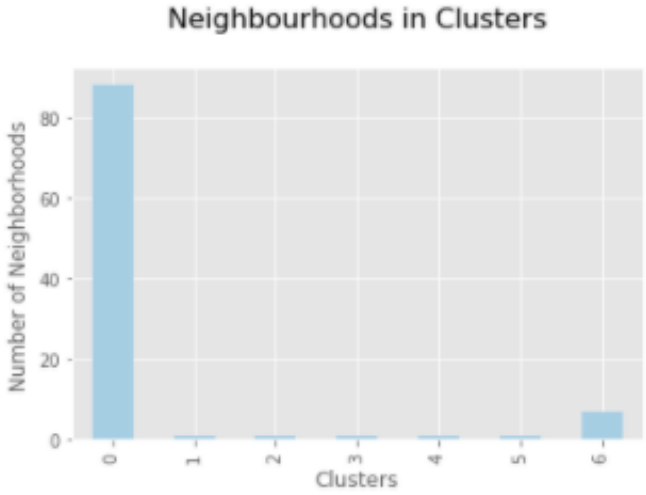
1. Map of the custard business (folium)
2. Major classes in cluster
3. Count of Neighbors in each cluster
4. Graph of counts

Cluster map:



Number of Neighborhood in Cluster

Cluster	Number of Neigborhoods	Some of the Common Venues in cluster
0	88	Bars, Coffee Shopes, Pizza Places
1	1	Fast Food Restaurant, Yoga Studio
2	1	Martial Arts School, Electronic Store
3	1	Baseball Field, Dog Run
4	1	Filipino Restaurant, Eastern European Restaurant
5	1	Pizza Places
6	7	Playground, Parks,



8. Conclusion

The purpose of this project was to get location data in the Toronto area and classify them into classes/cluster. On the way, it is to learn about the uses of Foursquare API method of downloading data, process and analyze it, and present it to users. This has been demonstrated all along in the above processes. Moreover, anyone who needs to see the classified business in Toronto or who tries to replicate the process to get similar or better results, the foundation is built thought this process. Many can benefit out this process, that incudes but is not limited to researchers, students, teacher, business entrepreneurs, local government official and so forth.

Data was classified into 7 clusters. Although the k-means are not clear because it changes in every k-means run. However, the general principle remains the same.

Reference

1. List of postal codes of Canada: M
   en.wikipidia.org ---- (visited on December 2020)
   https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

2. List of postal code from online ---- (visited on December 2020)
   https://cocl.us/Geospatial_data

3. Several IBM data science notes and lectures from the IBM Coursera class