# Capstone Project Report: Battle of Neighborhoods

## Applied Data Science by Coursera/ IBM

IBM Data Science Professional Certificate

## Table of contents

# Introduction

Moving to a new city, state or country - is no easy task and one probably will feel the effect of change. Sometimes the effect of change could gradually lead us to culture shock! This might result to pack up and head back to home. Before doing so, let's have a look how data science (tools and methodology) can help us to find some similarities/ dissimilarities in neighborhoods.

Considering relocation, people usually tend to explore the places before moving and that involves so many aspects including **neighborhood analysis**. This requires a search algorithm that usually returns the requested features such as population rate, median housing price, school ratings, crime rates, weather conditions, recreational facilities and many more. Wouldn't it be nice to have an application (one platform) that could spit out an extensive analysis of all these features for a neighborhood or a comparative analysis between neighborhoods by just sending out the names of the neighborhoods?

In this project the above mentioned user need will be taken as main idea to develop the model. I will be focusing on **neighborhood analysis** in the city of our choice.

This Project aims to help the stakeholders take a better decision on choosing the best neighborhood out of many neighborhoods in the city of **Toronto** based on the distribution of various facilities in and around that neighborhood. For example, this project would compare 2 or more randomly chosen neighborhoods and analyze the top 10 most common venues in each of those two neighborhoods based on the number of visits by people in each of those places. I will use K-means clustering unsupervised machine learning algorithm to cluster the venues based on the place category such as restaurants, park, coffee shop, gym etc. This would help to understand better, the similarities and dissimilarities between/ among the chosen neighborhoods to retrieve more insights and to conclude which neighborhood wins over other.

# Data

According to this problem, I will need to acquire data about the city of Toronto, specifically the boroughs and neighborhoods of the city. Geospatial data of the city, its boroughs and all the neighborhoods. Following this it also requires to gather data about each neighborhoods such as what are the top venues and most common venues, which categories these venues belong to.

To obtain the best datasets to achieve our aim, I will use **Foursquare API** (*) as my prime data gathering source.

* *Foursquare API has a database of more than 105 million places, especially their places API which provides the ability to perform location search, location sharing and details about a business. Photos, tips and reviews jolted by Foursquare users can also be used in many productive ways to add value to the results.*

* *Please note: Due to limitations (on API call Quota) the number of places per neighborhood parameter would reasonably be set to 100 and the radius parameter would be set to 700.*

# Methodology

## ➢ Data Collection

This project depends on publicly available data, mainly from Wikipedia [1-5]. I will use **Web Scraping** with **Beautiful Soup** to retrieve data from different Web pages. Acquired data will then be cleaned and sorted according to the requirements.

```
#importing all libraries

import numpy as np
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import json
from pandas.io.json import json_normalize
from geopy.geocoders import Nominatim
from bs4 import BeautifulSoup
import lxml.html as lh
import requests
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
!conda install -c conda-forge folium=0.5.0 --yes
import folium
print('Folium installed')
print('Libraries imported.')
```

```
#Scraping Wikipedia

url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
r = requests.get(url)

soup = BeautifulSoup(r.content, 'html.parser')
table = soup.find('table')
trows = table.find_all('tr')
rows = []
for tr in trows:
    i = tr.find_all('td')
    if i:
        rows.append(i)
lst = []
for row in rows:
    postalcode = row[0].text.rstrip()
    borough = row[1].text.rstrip()
    neighborhood = row[2].text.rstrip()
    if borough != 'Not assigned':
        if neighborhood == 'Not assigned':
            neighborhood = borough
        lst.append([postalcode, borough, neighborhood])

cols = ['PostalCode', 'Borough', 'Neighborhood']
df = pd.DataFrame(lst, columns=cols)
print(df.shape)
```

```
(211, 3)
```

check – (.head) the scraped data:

| | PostalCode | Borough | Neighborhood |
|---|---|---|---|
| **0** | M3A | North York | Parkwoods |
| **1** | M4A | North York | Victoria Village |
| **2** | M5A | Downtown Toronto | Harbourfront |
| **3** | M5A | Downtown Toronto | Regent Park |
| **4** | M6A | North York | Lawrence Heights |

## ➢ Data Wrangle

I encountered that in the first data frame (df) there are several boroughs sharing the same postal code but recorded in separate rows. So I would group those by postal code to reduce the size of the data frame.

```
#Cleaning data

#groupby PostalCode, keep the first Borough and join() Neighborhoods
df = df.groupby('PostalCode').agg(
    {
        'Borough':'first',
        'Neighborhood': ', '.join,}
    ).reset_index()

df.shape
```

```
(103, 3)
```

The size now looks fine. Let's run a test to see if we have the dataset how we wanted it to be.

```
df.head()
```

| | 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 |

Yes! that's fine.

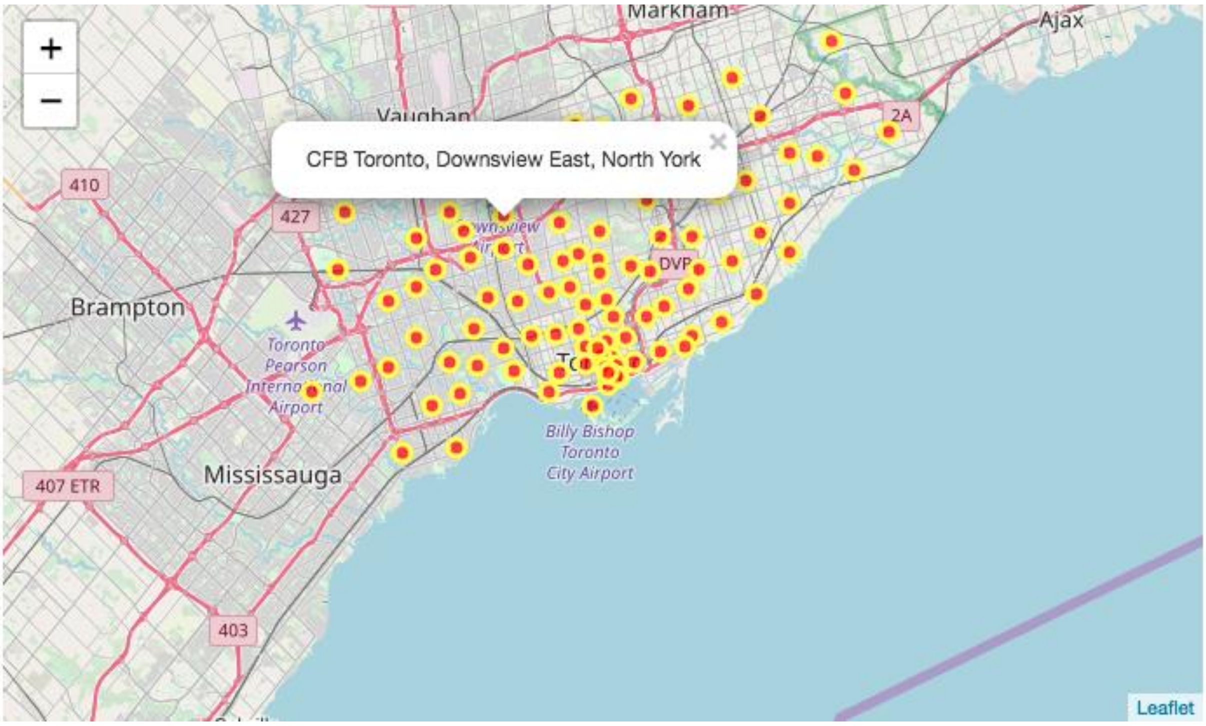
Another requirement of the dataset is to obtain the geospatial data for each neighborhood. I have collected geospatial data various sources [1-4]. I initially put all the collected data into .csv file and then read (dfgeo) in python and converted into pandas data frame (df2).

```
#Clean and sort
#read the csv and put it into pandas data frame
dfgeo = pd.read_csv("Geospatial.csv")
dfgeo.rename(columns={'Postal Code': 'PostalCode'}, inplace=True)
df2 = pd.merge(df, dfgeo, on="PostalCode", how='left')
df2.head()
```

| | PostalCode | Borough | Neighborhood | Latitude | Longitude | Population |
|---|---|---|---|---|---|---|
| **0** | M1B | Scarborough | Rouge, Malvern | 43.806686 | -79.194353 | 66108 |
| **1** | M1C | Scarborough | Highland Creek, Rouge Hill, Port Union | 43.784535 | -79.160497 | 35626 |
| **2** | M1E | Scarborough | Guildwood, Morningside, West Hill | 43.763573 | -79.188711 | 46943 |
| **3** | M1G | Scarborough | Woburn | 43.770992 | -79.216917 | 29690 |
| **4** | M1H | Scarborough | Cedarbrae | 43.773136 | -79.239476 | 24383 |

The city of Toronto has 11 Boroughs and 103 Neighborhoods.

After finding the geographical coordinate of Toronto, I used **Folium** (python visualization library) to visualize the neighborhoods and the cluster distribution of the city of Toronto over an interactive leaflet map.




Then I used **Foursquare** to collect the venue data of each neighborhood. As mentioned earlier I have set the limit for API calls for venue parameter to 100 and the radius parameter to 700.

| | name | categories | lat | lng |
|---|---|---|---|---|
| 0 | Downtown Toronto | Neighborhood | 43.653232 | -79.385296 |
| 1 | Textile Museum of Canada | Art Museum | 43.654396 | -79.386500 |
| 2 | Sansotei Ramen 三草亭 | Ramen Restaurant | 43.655157 | -79.386501 |
| 3 | Japango | Sushi Restaurant | 43.655268 | -79.385165 |
| 4 | Tsujiri | Tea Room | 43.655374 | -79.385354 |

Then I created a function to have the same process repeated for all 103 neighborhoods of Toronto. After processing the dataset I found that there are 319 unique venue categories exist in Toronto. However the size of the dataset is quite large and I reduced it from (3461, 319) to (102, 319) by “one hot encoding” and “.groupby()” method.

In addition, I looked at the neighborhoods with the top 10 most common venue and their frequencies.

```
----Adelaide, King, Richmond----
                venue  freq
0         Coffee Shop  0.07
1                Café  0.06
2          Steakhouse  0.04
3     Sushi Restaurant  0.04
4  American Restaurant  0.04
5            Gastropub  0.03
6           Restaurant  0.03
7                  Bar  0.03
8      Thai Restaurant  0.03
9              Theater  0.03


----Agincourt----
                venue  freq
0  Shanghai Restaurant   0.1
1            Pool Hall   0.1
2      Badminton Court   0.1
3       Breakfast Spot   0.1
4          Coffee Shop   0.1
5       Sandwich Place   0.1
6       Clothing Store   0.1
7      Motorcycle Shop   0.1
8               Lounge   0.1
9         Skating Rink   0.1


----Agincourt North, L'Amoreaux East, Milliken, Steeles Ea
```

I then put the collected data into pandas data frame and merged with the df2.

| stalCode | Borough | Neighborhood | Latitude | Longitude | Population | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M1B | Scarborough | Rouge, Malvern | 43.806686 | -79.194353 | 66108 | 4.0 | Fast Food Restaurant | Coffee Shop | Spa | Bus Station | Hobby Shop | Construction & Landscaping | Women's Store | Donut Shop |
| M1C | Scarborough | Highland Creek, Rouge Hill, Port Union | 43.784535 | -79.160497 | 35626 | 4.0 | Breakfast Spot | Bar | Burger Joint | Dumpling Restaurant | Discount Store | Dive Bar | Dog Run | Doner Restaurant |
| M1E | Scarborough | Guildwood, Morningside, West Hill | 43.763573 | -79.188711 | 46943 | 0.0 | Pizza Place | Fast Food Restaurant | Grocery Store | Breakfast Spot | Moving Target | Electronics Store | Fried Chicken Joint | Rental Car Location |
| M1G | Scarborough | Woburn | 43.770992 | -79.216917 | 29690 | 1.0 | Park | Coffee Shop | Convenience Store | Business Service | Event Space | Ethiopian Restaurant | Dessert Shop | Dim Sum Restaurant |
| M1H | Scarborough | Cedarbrae | 43.773136 | -79.239476 | 24383 | 4.0 | Coffee Shop | Indian Restaurant | Bakery | Thai Restaurant | Gym / Fitness Center | Fried Chicken Joint | Flower Shop | Chinese Restaurant |

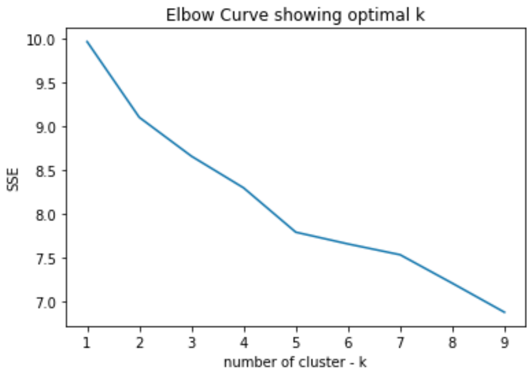
Before I proceeded to the next steps, I have done a small test on a single neighborhood as a part of the data exploration and to check the workability of our resources. (*see the supporting notebook*). The results look fine and I am ready to step forward.

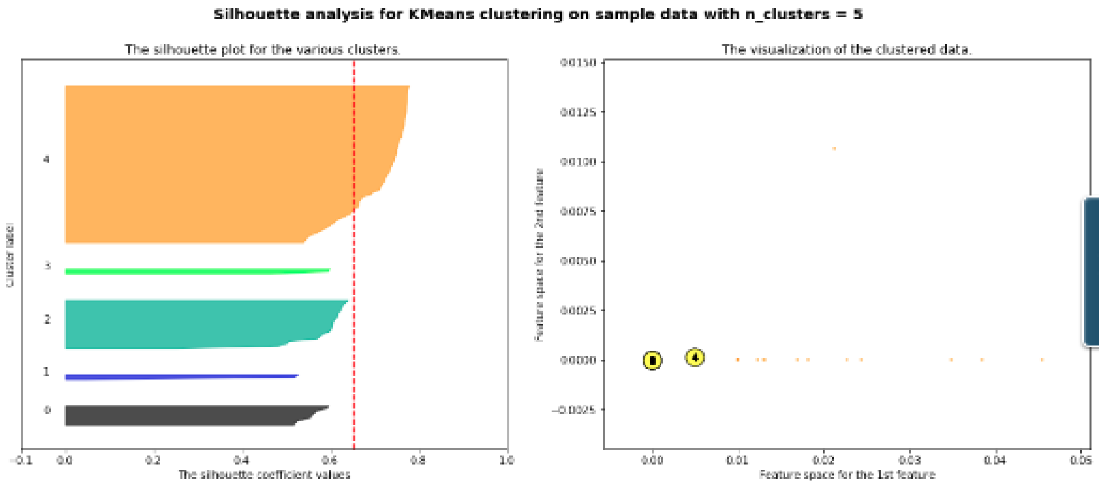
## ➢ Clustering

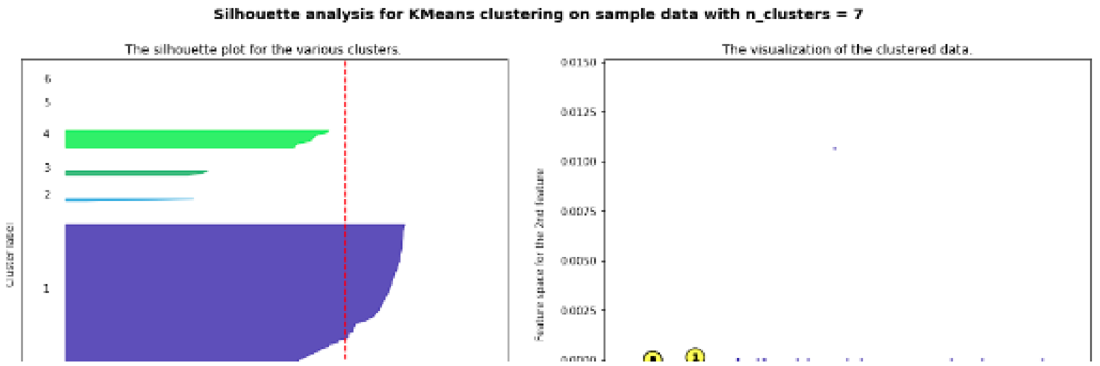
In order to carry out the extensive comparative analysis of randomly chosen neighborhoods, I have used K-means clustering; an unsupervised machine learning algorithm; to form the clusters of different categories of places residing in and around the neighborhoods.

However determining the optimal number of clusters in a data set is a fundamental issue in partitioning clustering, such as K-means clustering in our case, which requires the user to specify the number of clusters k to be generated. There are many methods to determine the optimal number. These methods include **direct methods** and **statistical testing methods**.

In this project, I have considered the direct methods. I used **Elbow method** and **Average Silhouette method** to calculate the optimal number of k.



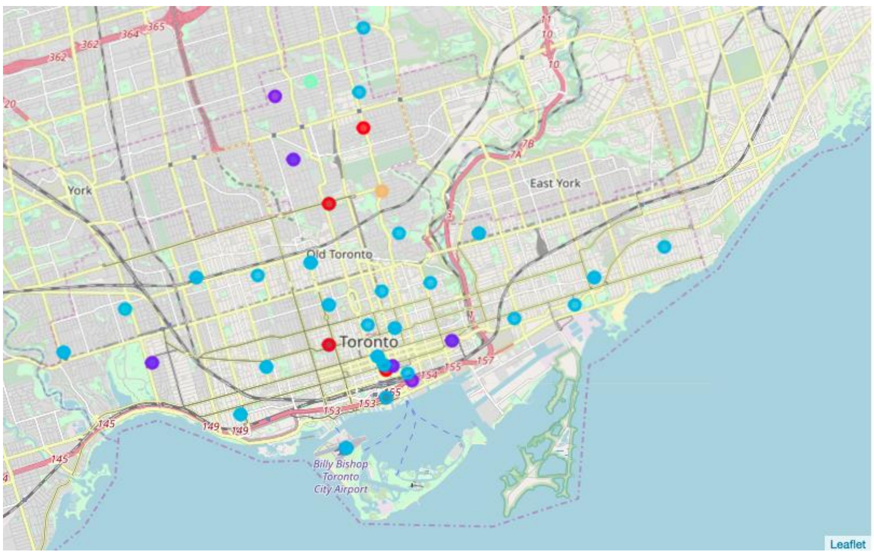



I set k = 5 and added the cluster labels to each group of neighborhoods accordingly.

```
new_toronto=toronto_merged.set_index("Neighborhood",drop=True)
new_toronto.head(10)
```

| Neighborhood | PostalCode | Borough | Latitude | Longitude | Population | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rouge, Malvern | M1B | Scarborough | 43.806686 | -79.194353 | 66108 | 4 | Fast Food Restaurant | Coffee Shop | Spa | Bus Station | Hobby Shop | Construction & Landscaping | Women's Store |
| Highland Creek, Rouge Hill, Port Union | M1C | Scarborough | 43.784535 | -79.160497 | 35626 | 4 | Breakfast Spot | Bar | Burger Joint | Dumpling Restaurant | Discount Store | Dive Bar | Dog Run |
| Guildwood, Morningside, West Hill | M1E | Scarborough | 43.763573 | -79.188711 | 46943 | 0 | Pizza Place | Fast Food Restaurant | Grocery Store | Breakfast Spot | Moving Target | Electronics Store | Fried Chicken Joint |
| Woburn | M1G | Scarborough | 43.770992 | -79.216917 | 29690 | 1 | Park | Coffee Shop | Convenience Store | Business Service | Event Space | Ethiopian Restaurant | Dessert Shop |
| Cedarbrae | M1H | Scarborough | 43.773136 | -79.239476 | 24383 | 4 | Coffee Shop | Indian Restaurant | Bakery | Thai Restaurant | Gym / Fitness | Fried Chicken Joint | Flower Shop |

**Visualization:** I have visualized this dataset (new_toronto), neighborhoods along with cluster labels by using Folium (as mentioned before).

# Analysis

In this section the clusters from each of the chosen neighborhoods would be analyzed individually, collectively and comparatively to derive the conclusions.

To begin with analysis, I have examined all the clusters. Individual outcomes are as follows:

```
#Examine
#Cluster 1

df3=toronto_merged.loc[toronto_merged['Cluster Labels'] == 0, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
df3
```

| | Borough | Population | 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 Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Scarborough | 46943 | 0 | Pizza Place | Fast Food Restaurant | Grocery Store | Breakfast Spot | Moving Target | Electronics Store | Fried Chicken Joint | Rental Car Location | Thrift / Vintage Store | Greek Restaurant |
| 5 | Scarborough | 36699 | 0 | Fast Food Restaurant | Women's Store | Convenience Store | Coffee Shop | Pizza Place | Dim Sum Restaurant | Diner | Discount Store | Dive Bar | Dog Run |
| 8 | Scarborough | 22913 | 0 | Furniture / Home Store | Chinese Restaurant | Wings Joint | Burger Joint | Dim Sum Restaurant | Diner | Discount Store | Dive Bar | Dog Run | Doner Restaurant |
| 11 | Scarborough | 29858 | 0 | Pizza Place | Burger Joint | Coffee Shop | Middle Eastern Restaurant | Seafood Restaurant | Bakery | Korean Restaurant | Fish Market | Intersection | Convenience Store |
| 13 | Scarborough | 34588 | 0 | Pharmacy | Shopping Mall | Pizza Place | Chinese Restaurant | Italian Restaurant | Sandwich Place | Bus Stop | Thai Restaurant | Fried Chicken Joint | Seafood Restaurant |
| 14 | Scarborough | 54680 | 0 | Chinese Restaurant | Pharmacy | BBQ Joint | Pizza Place | Park | Noodle House | Caribbean Restaurant | Shop & Service | Fast Food Restaurant | Bakery |
| 15 | Scarborough | 48471 | 0 | Fast Food Restaurant | Grocery Store | Chinese Restaurant | Pharmacy | Indian Restaurant | Burger Joint | Cosmetics Shop | American Restaurant | Other Great Outdoors | Sandwich Place |

```
#Cluster 2

df4=toronto_merged.loc[toronto_merged['Cluster Labels'] == 1, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
df4
```

| | Borough | Population | 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 Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | Scarborough | 29690 | 1 | Park | Coffee Shop | Convenience Store | Business Service | Event Space | Ethiopian Restaurant | Dessert Shop | Dim Sum Restaurant | Diner | Discount Store |
| 21 | North York | 32320 | 1 | Park | Coffee Shop | Bus Line | Trail | Women's Store | Donut Shop | Dim Sum Restaurant | Diner | Discount Store | Dive Bar |
| 23 | North York | 7843 | 1 | Park | Tennis Court | Intersection | Pet Store | Bank | Donut Shop | Dim Sum Restaurant | Diner | Discount Store | Dive Bar |
| 25 | North York | 34615 | 1 | Park | Fast Food Restaurant | Pet Store | Burger Joint | Food & Drink Shop | Women's Store | Donut Shop | Diner | Discount Store | Dive Bar |
| 30 | North York | 5997 | 1 | Sandwich Place | Coffee Shop | Airport | Park | Donut Shop | Dessert Shop | Dim Sum Restaurant | Diner | Discount Store | Dive Bar |
| 36 | East York | 46866 | 1 | Park | Pharmacy | Skating Rink | Asian Restaurant | Bus Stop | Bus Line | Curling Ice | Cosmetics Shop | Athletics & Sports | Video Store |
| 44 | Central Toronto | 15330 | 1 | Bus Line | Park | Business Service | Swim School | Women's Store | Donut Shop | Dim Sum Restaurant | Diner | Discount Store | Dive Bar |
| 48 | Central Toronto | 10463 | 1 | Park | Thai Restaurant | Gym / Fitness Center | Gym | Grocery Store | Playground | Bank | Women's Store | Dive Bar | Design Studio |
| 50 | Downtown Toronto | 14561 | 1 | Park | Playground | Gym / Fitness | Trail | Doner Restaurant | Design Studio | Dessert Shop | Dim Sum Restaurant | Diner | Discount Store |

```
#Examine
#Cluster 3

df5=toronto_merged.loc[toronto_merged['Cluster Labels'] == 2, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
df5
```

| | Borough | Population | 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 Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | North York | 11717 | 2 | Martial Arts Dojo | Cafeteria | Falafel Restaurant | Exhibit | Dessert Shop | Dim Sum Restaurant | Diner | Discount Store | Dive Bar | Dog Run |

```
#Examine
#Cluster 4

df6=toronto_merged.loc[toronto_merged['Cluster Labels'] == 3, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
df6
```

| | Borough | Population | 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 Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 96 | North York | 11950 | 3 | Bakery | Pizza Place | Empanada Restaurant | Women's Store | Donut Shop | Dim Sum Restaurant | Diner | Discount Store | Dive Bar | Dog Run |

```python
#Examine
#Cluster 5

df7=toronto_merged.loc[toronto_merged['Cluster Labels'] == 4, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
df7
```

| | Borough | Population | 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 Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Scarborough | 66108 | 4 | Fast Food Restaurant | Coffee Shop | Spa | Bus Station | Hobby Shop | Construction & Landscaping | Women's Store | Donut Shop | Diner | Discount Store |
| 1 | Scarborough | 35626 | 4 | Breakfast Spot | Bar | Burger Joint | Dumpling Restaurant | Discount Store | Dive Bar | Dog Run | Doner Restaurant | Donut Shop | Drugstore |
| 4 | Scarborough | 24383 | 4 | Coffee Shop | Indian Restaurant | Bakery | Thai Restaurant | Gym / Fitness Center | Fried Chicken Joint | Flower Shop | Chinese Restaurant | Rental Car Location | Caribbean Restaurant |
| 6 | Scarborough | 48434 | 4 | Discount Store | Coffee Shop | Sandwich Place | Light Rail Station | Department Store | Convenience Store | Chinese Restaurant | Intersection | Grocery Store | Metro Station |
| 7 | Scarborough | 35081 | 4 | Intersection | Diner | Coffee Shop | Bus Line | Bakery | Bus Station | Fast Food Restaurant | Park | Metro Station | Soccer Field |
| 9 | Scarborough | 22136 | 4 | Skating Rink | Bank | Café | Diner | Discount Store | General Entertainment | Thai Restaurant | Park | College Stadium | Eastern European Restaurant |
| 10 | Scarborough | 45571 | 4 | Electronics Store | Indian Restaurant | Fast Food Restaurant | Wings Joint | Gym / Fitness Center | Vietnamese Restaurant | Coffee Shop | Pet Store | Bakery | Chinese Restaurant |

From this I could observe that cluster 5 (with Cluster Label 4) has the highest number of neighborhood with the first most common venue "Fast Food Restaurant" and the second largest is cluster 2 (with Cluster Label 1) with the first most common venue "Park".

The project aims to compare randomly chosen neighborhoods and to do so I have created a table that contains all the data frames (df3,df4,df5,df6 and df7) according to the cluster labels.

```python
#creating cluster table
cluster_t=pd.DataFrame({"Cluster1":df3["Borough"],
                        "Cluster2":df4["Borough"],
                        "Cluster3":df5["Borough"],
                        "Cluster4":df6["Borough"],
                        "Cluster5":df7["Borough"]
                        })
```

```python
cluster_t = cluster_t.replace(np.nan, '', regex=True)
cluster_t
```

| | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 |
|---|---|---|---|---|---|
| 0 | | | | | Scarborough |
| 1 | | | | | Scarborough |
| 2 | Scarborough | | | | |
| 3 | | Scarborough | | | |
| 4 | | | | | Scarborough |
| 5 | Scarborough | | | | |
| 6 | | | | | Scarborough |
| 7 | | | | | Scarborough |
| 8 | Scarborough | | | | |
| 9 | | | | | Scarborough |

# Results and Discussion

In this step I take 2 random neighborhood names as input and run the comparison.

## Compare the neighborhoods

```
Nbd1=input("Enter the Neighborhood: ")
```

```
Enter the Neighborhood:  Northwest
```

```
Nbd2=input("Enter the Neighborhood: ")
```

```
Enter the Neighborhood:  Weston
```

```
venue_comparison=new_toronto.loc[[Nbd1,Nbd2]].T
venue_comparison
```

| Neighborhood | Northwest | Weston |
|---|---|---|
| **PostalCode** | M9W | M9N |
| **Borough** | Etobicoke | York |
| **Latitude** | 43.7067 | 43.7069 |
| **Longitude** | -79.5941 | -79.5182 |
| **Population** | 40684 | 25074 |
| **Cluster Labels** | 0 | 4 |
| **1st Most Common Venue** | Rental Car Location | Diner |
| **2nd Most Common Venue** | Home Service | Fried Chicken Joint |
| **3rd Most Common Venue** | Drugstore | Pharmacy |
| **4th Most Common Venue** | Hotel | Breakfast Spot |
| **5th Most Common Venue** | Donut Shop | Women's Store |
| **6th Most Common Venue** | Dessert Shop | Donut Shop |
| **7th Most Common Venue** | Dim Sum Restaurant | Dim Sum Restaurant |
| **8th Most Common Venue** | Diner | Discount Store |
| **9th Most Common Venue** | Discount Store | Dive Bar |
| **10th Most Common Venue** | Dive Bar | Dog Run |

From the comparison between "Northwest" and "Weston" I could retrieve the geospatial data of the neighborhoods, population count, the top 10 most common venue categories.

This comparison model also works for more than 2 randomly chosen neighborhoods. I have tested with 3 random neighborhood names as follows:

```
Nbd1=input("Enter the Neighborhood: ")
```

```
Enter the Neighborhood:  Rosedale
```

```
Nbd2=input("Enter the Neighborhood: ")
```

```
Enter the Neighborhood:  Woburn
```

```
Nbd3=input("Enter the Neighborhood: ")
```

```
Enter the Neighborhood:  Humber Summit
```

```
venue_comparison=new_toronto.loc[[Nbd1,Nbd2,Nbd3]].T
venue_comparison
```

| Neighborhood | Rosedale | Woburn | Humber Summit |
|---|---|---|---|
| PostalCode | M4W | M1G | M9L |
| Borough | Downtown Toronto | Scarborough | North York |
| Latitude | 43.6796 | 43.771 | 43.7563 |
| Longitude | -79.3775 | -79.2169 | -79.566 |
| Population | 14561 | 29690 | 11950 |
| Cluster Labels | 1 | 1 | 3 |
| 1st Most Common Venue | Park | Park | Bakery |
| 2nd Most Common Venue | Playground | Coffee Shop | Pizza Place |

From the comparison among "Rosedale", "Woburn" and "Humber Summit" I could retrieve the postal code, name of the boroughs, geospatial data of the neighborhoods, population count, cluster labels and the top 10 most common venue categories.

This comparison model clearly shows the expected outcomes. I aimed to build up a model that can compare two or more randomly chosen neighborhoods of the city of Toronto. The comparison is carried out using K-means clustering algorithm to cluster the neighborhoods based on its venue categories. This model could be helpful for stakeholders to gain more insights about individual neighborhood or to compare chosen neighborhoods.

# Conclusion

In this project, I have taken into account the need of an application or one platform that would help stakeholders to understand a country, state, city or its neighborhoods better. I have also mentioned that this would require a search algorithm that usually would return the requested features such as population rate, median housing price, school ratings, crime rates, weather conditions, recreational facilities etc. But I specifically focused only the neighborhood analysis in means of simple comparison of geospatial data, population counts and the top most common venues based on venue categories. This leaves us to an open end to elaborate the search algorithms by adding more features in the future and nonetheless refining and improving the algorithm further.

# References


1. *http://cocl.us/Geospatial_data*
2. *https://www.statcan.gc.ca*
3. *https://en.wikipedia.org/wiki/Demographics_of_Toronto_neighbourhoods*
4. *https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Demographics_of_Toronto.html*
5. *https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M*