IBM Data Science Specialization: The Battle of Neighborhood

Project Report

Best Place to Stay in Toronto

Date: 26 Apr 2020

Musfiqul Azad

Table of Contents

1. Introduction

a. Scenario and Background

Toronto is an international centre of business, finance, arts, and culture, and is recognized as one of the most multicultural and cosmopolitan cities in the world. The number of immigrants has been increased for last couple of decades. This is very difficult to choose a place of stay and the place of work without visiting Canada. This project will help the immigrants as well as local people to find optimum location with all the facilities available nearby which matches with previous place of stay.

b. Problem Statement

Finding a rental apartment in a good location in Toronto which has similar venues in Bangalore with reasonable cost. Let's quantify the necessities.

- The venues should be similar to the place I stay in Bangalore
- The rent should be between CAD 1500 and CAD 2200
- The number of bedrooms and bathroom should be 2
- There should be at least one Indian Restaurant nearby

c. Target Audience

This project will help any person or family moving to any major city in the world. As we use Foursquare data and mapping techniques, this can be replicated for similar cases for which data is available. People reviewing this project can be included in this group. This is a good start for this data science project and great learning experience for me as well.

2. Data Collection and Exploration

a. Data required

- https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M, The postal codes of Canada are available in the above link. The method web-scraping is used to collect the data from the webpage to csv using Beautifulsoup package in Python.

- http://cocl.us/Geospatial_data The link to a csv file that has the geographical coordinates of each postal code.

- https://www.kaggle.com/rmenon1998/bangalore-neighborhoods/data# This has Bangalore neighborhood data.

- https://finkode.com/ka/bangalore.html Pin code data
- https://www.kaggle.com/rajacsp/toronto-apartment-price Apartments available in Toronto with coordinates and price

b. Usage of data

i. Using Foursquare and geopy data find the current location in Bangalore neighborhood and top 10 venues nearby.
ii. Plot the data in a map.
iii. Map top 10 venues for all Toronto neighborhood and cluster in groups.
iv. Find out the coordinates of the rental apartments using geopy data.
v. Find out the location of the available apartments using apartment location and price data.
vi. Find out Indian Restaurants nearby using Foursquare with search query = 'Indian'
vii. Analyze the cluster groups that matches with the Bangalore location.

viii. Finalize the place of stay by plotting all the data on a map.

c. Visualizations

Multiple maps to be created to visualize the locations accurately. The clusters, nearby venues, Indian Restaurants and rental apartments are marked in Toronto map. The packages like Folium, Matplotlib are used to plot data to understand and analyze data.

3. Methodology

a. Business Understanding

The objective is to find out best location of stay and choose suitable apartment from the list. The apartment should have similar venues nearby as with Bangalore neighborhood. For this we have to analyze the venues nearby my house in Bangalore and venues in Toronto neighborhood.

b. Execution

Importing all required packages

In [208]:
```
import requests
import lxml.html as lh
import pandas as pd
from bs4 import BeautifulSoup
import geocoder # import geocoder
import numpy as np
from geopy.geocoders import Nominatim
import folium
import os
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

Exploring my Location

In [37]:
```
df_bang = pd.read_csv('Geospatial_Coordinates_Bangalore.csv')
```

In [44]:
```
df_bang[df_bang['Pincode']==560066]
```

Out[44]:

	Pincode	Neighbourhood	Borough	Latitude	Longitude
36	560066	Whitefield	Bangalore South	12.969807	77.749963

In [70]:
```
my_lat = df_bang[df_bang['Pincode']==560066]['Latitude'].values[0]
my_long = df_bang[df_bang['Pincode']==560066]['Longitude'].values[0]
print('The geograpical coordinate of my location are {}, {}.'.format(my_lat, my_long))
```

```
The geograpical coordinate of my location are 12.969806599999998, 77.7499632.
```

In [22]:
```
CLIENT_ID = os.environ['CLIENT_ID'] # your Foursquare ID
CLIENT_SECRET = os.environ['CLIENT_SECRET'] # your Foursquare Secret
VERSION = '20200411' # Foursquare API version
radius = 500
LIMIT = 50
```

In [315]:
```
my_url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(CL
```

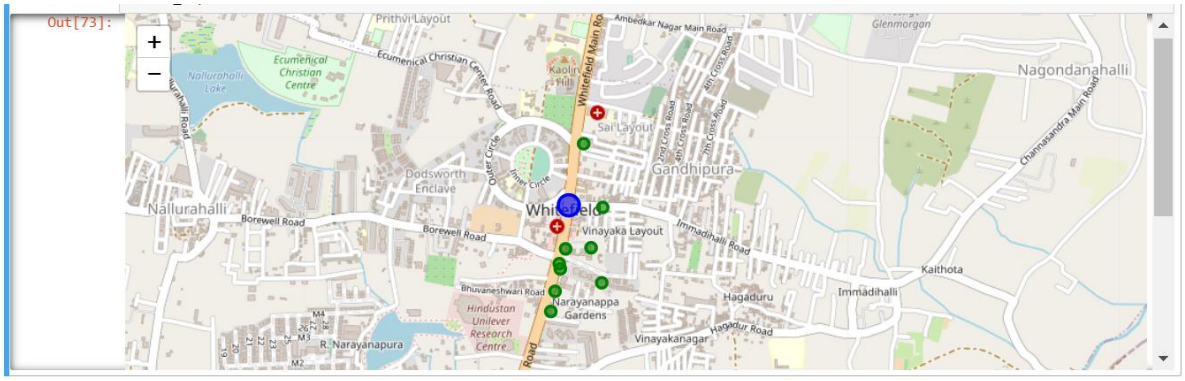
In [60]:
```
result = requests.get(my_url).json()
```

In [74]:
```
items = result['response']['groups'][0]['items']
len(items)
```

Out[74]: 9

The top 10 venues are found out from Foursquare and plotted on the map with current location.

Out[73]:



The required neighborhood data is collected from the URL https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M, using beautifulsoup package.

Collecting required data for Toronto

In [2]:
```
url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
result = requests.get(url).text
```

In [3]:
```
soup = BeautifulSoup(result, 'lxml')
col = []
values = []
table= soup.find('table', class_ = 'wikitable')

for match in table.find_all('th'):
    head = match.text.rstrip('\n')
    col.append((head,[]))

for item in table.find_all('td'):
    data = item.text.rstrip('\n')
    values.append(data)

for i in range(len(values)):
    col[i%3][1].append(values[i])
    i+=1

Dict={title:column for (title,column) in col}
df=pd.DataFrame(Dict)
df.tail()
```

Out[3]:

	Postal code	Borough	Neighborhood
175	M5Z	Not assigned	
176	M6Z	Not assigned	
177	M7Z	Not assigned	
178	M8Z	Etobicoke	Mimico NW / The Queensway West / South of Bloo...
179	M9Z	Not assigned	

The geospatial data is downloaded from URL http://cocl.us/Geospatial_data and all the co-ordinates are assigned to the postal codes of the neighborhoods of Toronto.

Uploading geospatial data and merging with original dataframe

In [13]:
```
geo_df = pd.read_csv('Geospatial_Coordinates.csv') #Creating geo_df with geospatial data
geo_df.rename(columns={"Postal Code":"Postal code"}, inplace = True) # Updating column names so that it will be easy for merge
geo_df.head()
```

Out[13]:

	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

In [14]:
```
df = pd.merge(df, geo_df, on ='Postal code', how ='left') # Merging two datafrmaes
```

In [15]:
```
df.head()
```

Out[15]:

	Postal code	Borough	Neighborhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494

In [111]:
```
df.describe()
```

Out[111]:

	Latitude	Longitude
count	103.000000	103.000000

The neighborhood data is mapped in Toronto geographical map.

Finding all the nearby venues of the neighborhoods of Toronto using Foursquare credentials.

Finding venues nearby for all the neighborhoods

In [27]:
```
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(CLIENT_
#url
```

In [28]:
```
results = requests.get(url).json()['response']['venues']
#a = []
#a.append([b['name'] for b in results])
#a
#results
```

In [29]:
```
venues_list = []
for items in results:
    try:
        v_name = items['name']
    except:
        v_name = ''
    try:
        v_lat = items['location']['lat']
    except:
        v_lat = ''
    try:
        v_lng = items['location']['lng'],
    except:
        v_lng = ''
    try:
        v_code = items['location']['postalCode'][:4]
    except:
        v_code = ''
    try:
        v_c_name = items['categories'][0]['name']
    except:
        v_c_name = ''
    venues_list.append([(v_name, v_lat, v_lng, v_code, v_c_name)])
```

In [148]:
```
nearby_venues.head()
```

Out[148]:

	Venue_Name	Venue_Latitude	Venue_Longitude	Postal_Code	Venue_Category
0	City Hall Council Chambers	43.651827	(-79.38394893163043,)		City Hall
1	Toronto City Hall	43.653140	(-79.38396692276001,)	M5H	City Hall
2	City of Toronto Civic Innovation Office	43.653454	(-79.383952,)	M5H	City Hall
3	City Hall Podium Green Roof	43.653504	(-79.38386645704847,)		Garden
4	the Archer / Three-Way Piece No. 2	43.652622	(-79.383923,)		Other Great Outdoors

In [151]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
50 venues were returned by Foursquare.
```

The venues are counted by grouping with column Neighborhood.

In [155]:
```
toronto_venues.groupby('Neighborhood').count()
```

Out[155]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Berczy Park	50	50	50	50	50	50
Brockton, Parkdale Village, Exhibition Place	22	22	22	22	22	22
Business reply mail Processing CentrE	18	18	18	18	18	18
CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport	18	18	18	18	18	18
Central Bay Street	50	50	50	50	50	50
Christie	18	18	18	18	18	18
Church and Wellesley	50	50	50	50	50	50
Commerce Court, Victoria Hotel	50	50	50	50	50	50
Davisville	33	33	33	33	33	33
Davisville North	6	6	6	6	6	6
Dufferin, Dovercourt Village	16	16	16	16	16	16
First Canadian Place, Underground city	50	50	50	50	50	50
Forest Hill North & West	5	5	5	5	5	5
Garden District, Ryerson	50	50	50	50	50	50
Harbourfront East, Union Station, Toronto Islands	50	50	50	50	50	50
High Park, The Junction South	22	22	22	22	22	22
India Bazaar, The Beaches West	18	18	18	18	18	18
Kensington Market, Chinatown, Grange Park	50	50	50	50	50	50
Lawrence Park	4	4	4	4	4	4
Little Portugal, Trinity	41	41	41	41	41	41
Moore Park, Summerhill East	2	2	2	2	2	2

One hot encoding is performed using get_dummies of Pandas.

In [293]:
```
# one hot encoding
toronto_onehot = pd.get_dummies(toronto_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
toronto_onehot['Neighborhood'] = toronto_venues['Neighborhood']

# 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(100)
```

Out[293]:

	Yoga Studio	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	Aquarium	...	Theater	Theme Restaurant	Toy / Game Store	Trail	Train Station	Vegetarian / Vegan Restaurant	Vi Ga St
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
96	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
97	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
98	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
99	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	

100 rows × 213 columns

Top 5 venues are shown by neighborhood.

In [162]:
```
num_top_venues = 5

for hood in toronto_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = toronto_grouped[toronto_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Berczy Park----
               venue  freq
0        Coffee Shop  0.08
1       Cocktail Bar  0.06
2  Italian Restaurant  0.04
3           Beer Bar  0.04
4        Cheese Shop  0.04


----Brockton, Parkdale Village, Exhibition Place----
            venue  freq
0           Café  0.14
1    Coffee Shop  0.09
2  Breakfast Spot  0.09
3   Grocery Store  0.05
4    Intersection  0.05


----Business reply mail Processing CentrE----
                venue  freq
```

Top 10 venues are assigned to the neighborhoods and columns are arranged in order from 1st to 10th.

In [185]:
```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
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
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = toronto_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

Out[185]:

	Neighborhood	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	Cocktail Bar	Cheese Shop	Beer Bar	Seafood Restaurant	Restaurant	Italian Restaurant	Farmers Market	Café	Bakery
1	Brockton, Parkdale Village, Exhibition Place	Café	Coffee Shop	Breakfast Spot	Stadium	Convenience Store	Performing Arts Venue	Pet Store	Climbing Gym	Burrito Place	Restaurant
2	Business reply mail Processing CentrE	Light Rail Station	Yoga Studio	Auto Workshop	Fast Food Restaurant	Farmers Market	Comic Shop	Garden Center	Park	Pizza Place	Burrito Place
3	CN Tower, King and Spadina, Railway Lands, Har...	Airport Service	Airport Lounge	Airport Terminal	Coffee Shop	Harbor / Marina	Plane	Rental Car Location	Boutique	Sculpture Garden	Bar
4	Central Bay Street	Coffee Shop	Sandwich Place	Café	Italian Restaurant	Bubble Tea Shop	Burger Joint	Ice Cream Shop	Department Store	Salad Place	Ramen Restaurant

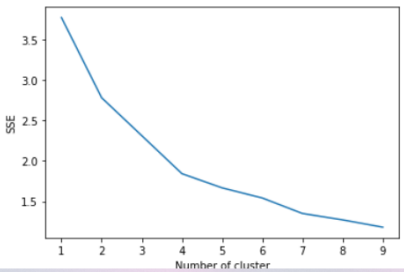
The neighborhoods are grouped in to clusters and the number of clusters to be decided by plotting SSE against no of clusters. So k is incremented from 1 to 10 and SSE is calculated for each step. In the graph mentioned below SSE is on Y axis and no of clusters is plotted on X axis. This process is called as Elbow method.

Cluster Neighborhoods

In [209]:
```
# set number of clusters
toronto_grouped_clustering = toronto_grouped.drop('Neighborhood', 1)
sse = {}
for kclusters in range(1, 10):

    # run k-means clustering
    kmeans = KMeans(n_clusters=kclusters, random_state=0, max_iter=1000).fit(toronto_grouped_clustering)
    sse[kclusters] = kmeans.inertia_

    # check cluster labels generated for each row in the dataframe
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel("Number of cluster")
plt.ylabel("SSE")
plt.show()
```



From the above graph we finalized the no of clusters should be 7. With k=7 the model is fitted and each row is assigned with a cluster number from 1 to 7.

In [214]:
```
k = 7
kmeans = KMeans(n_clusters=7, random_state=0).fit(toronto_grouped_clustering)
kmeans.labels_[0:100]
```

Out[214]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 5, 0, 1, 0,
       0, 0, 0, 0, 4, 2, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0])
```

In [212]:
```
neighborhoods_venues_sorted.drop(['Cluster Labels'], axis=1, inplace =True)
neighborhoods_venues_sorted.head()
```

Out[212]:

	Neighborhood	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	Cocktail Bar	Cheese Shop	Beer Bar	Seafood Restaurant	Restaurant	Italian Restaurant	Farmers Market	Café	Bakery
1	Brockton, Parkdale Village, Exhibition Place	Café	Coffee Shop	Breakfast Spot	Stadium	Convenience Store	Performing Arts Venue	Pet Store	Climbing Gym	Burrito Place	Restaurant
2	Business reply mail Processing CentrE	Light Rail Station	Yoga Studio	Auto Workshop	Fast Food Restaurant	Farmers Market	Comic Shop	Garden Center	Park	Pizza Place	Burrito Place
3	CN Tower, King and Spadina, Railway Lands, Har...	Airport Service	Airport Lounge	Airport Terminal	Coffee Shop	Harbor / Marina	Plane	Rental Car Location	Boutique	Sculpture Garden	Bar
4	Central Bay Street	Coffee Shop	Sandwich Place	Café	Italian Restaurant	Bubble Tea Shop	Burger Joint	Ice Cream Shop	Department Store	Salad Place	Ramen Restaurant

In [213]:
```
# add clustering labels

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

toronto_merged = toronto_df

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

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

Out[213]:

	Postal code	Borough	Neighborhood	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
0	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636	0	Coffee Shop	Pub	Park	Bakery	Theater	Restaurant	Café	Breakfast Spot	
1	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494	0	Coffee Shop	Sushi Restaurant	Diner	Mexican Restaurant	Beer Bar	Sandwich Place	Burger Joint	Burrito Place	
2	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937	0	Coffee Shop	Café	Tea Room	Middle Eastern Restaurant	Ramen Restaurant	Bookstore	Theater	Restaurant	
3	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418	0	Café	Gastropub	Coffee Shop	Creperie	Seafood Restaurant	Hotel	Farmers Market	Cosmetics Shop	
4	M4E	East Toronto	The Beaches	43.676357	-79.293031	6	Trail	Pub	Health Food Store	Wine Shop	Cupcake Shop	Doner Restaurant	Dog Run	Distribution Center	

The clusters are plotted on a map as shown below.

Red dots: Cluster 0

Purple dots: Cluster 1

Navy blue dots: Cluster 2

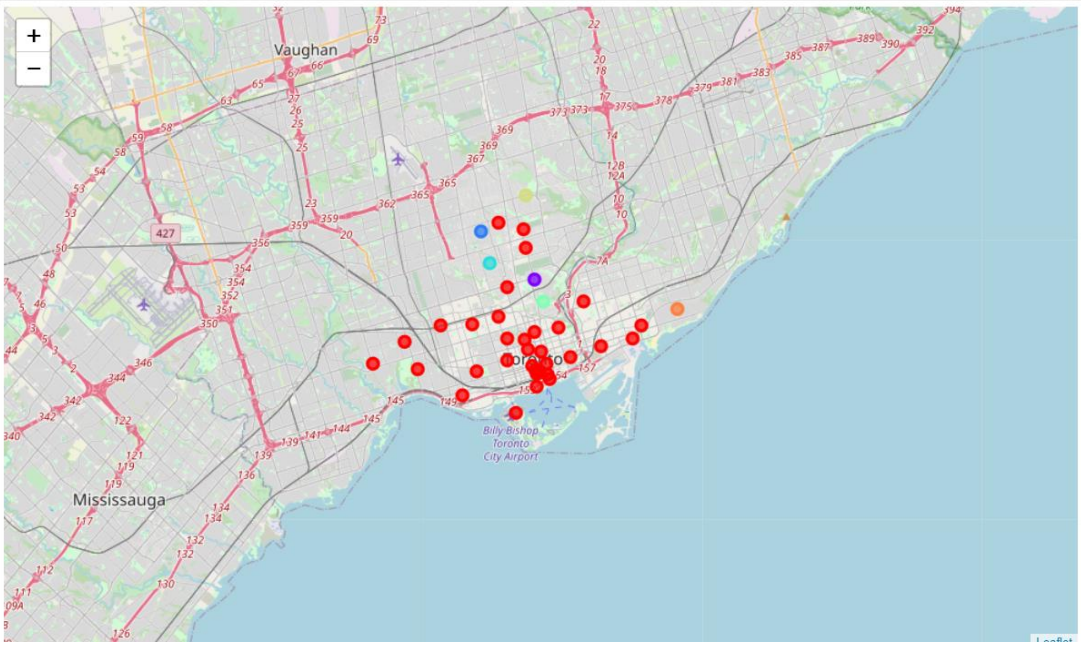
Light blue dots: Cluster 3

Aqua green dots: Cluster 4

Brown dots: Cluster 5

Orange dots: Cluster 6

Out[93]:



Apartment data with co-ordinates and prices are loaded to dataframe.

Loading Apartment data to dataframe

In [285]:
```
app_df = pd.read_csv('Toronto_apartment_rentals.csv')
app_df.shape
```

Out[285]: (1124, 7)

In [286]:
```
app_df[app_df.columns[6]] = app_df[app_df.columns[6]].replace('[\$,]', '', regex=True).astype(float)
```

In [287]:
```
app_df.head()
```

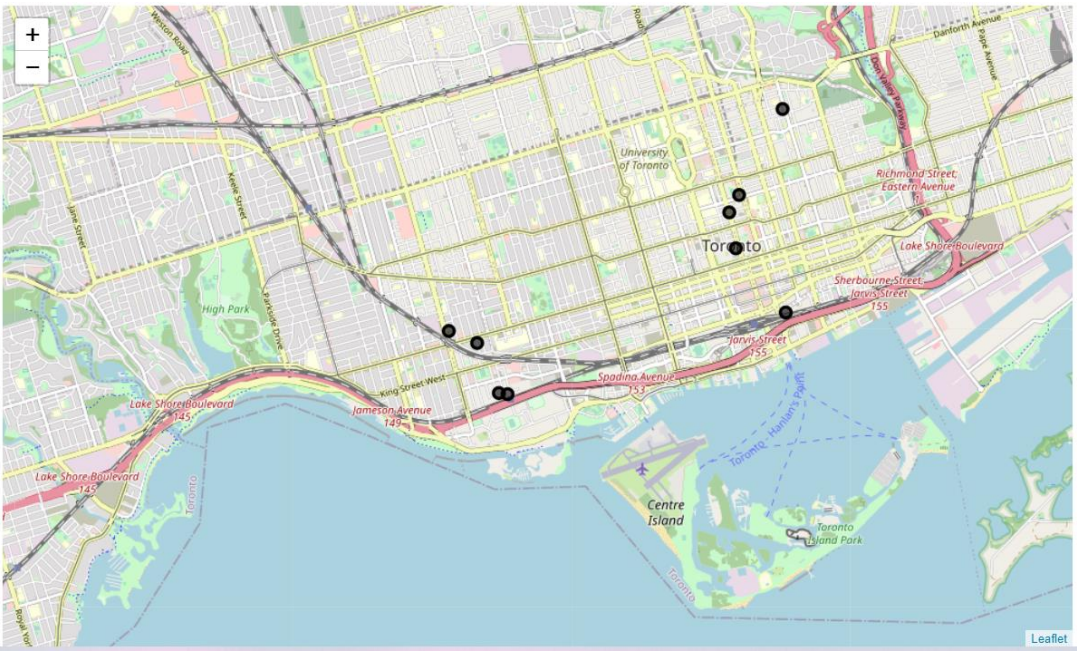
Out[287]:

	Bedroom	Bathroom	Den	Address	Lat	Long	Price
0	2	2.0	0	3985 Grand Park Drive, 3985 Grand Park Dr, Mis...	43.581639	-79.648193	2450.0
1	1	1.0	1	361 Front St W, Toronto, ON M5V 3R5, Canada	43.643051	-79.391643	2150.0
2	1	1.0	0	89 McGill Street, Toronto, ON, M5B 0B1	43.660605	-79.378635	1950.0
3	2	2.0	0	10 York Street, Toronto, ON, M5J 0E1	43.641087	-79.381405	2900.0
4	1	1.0	0	80 St Patrick St, Toronto, ON M5T 2X6, Canada	43.652487	-79.389622	1800.0

Refining the apartment data with required below conditions and plotting on Toronto geographical map.

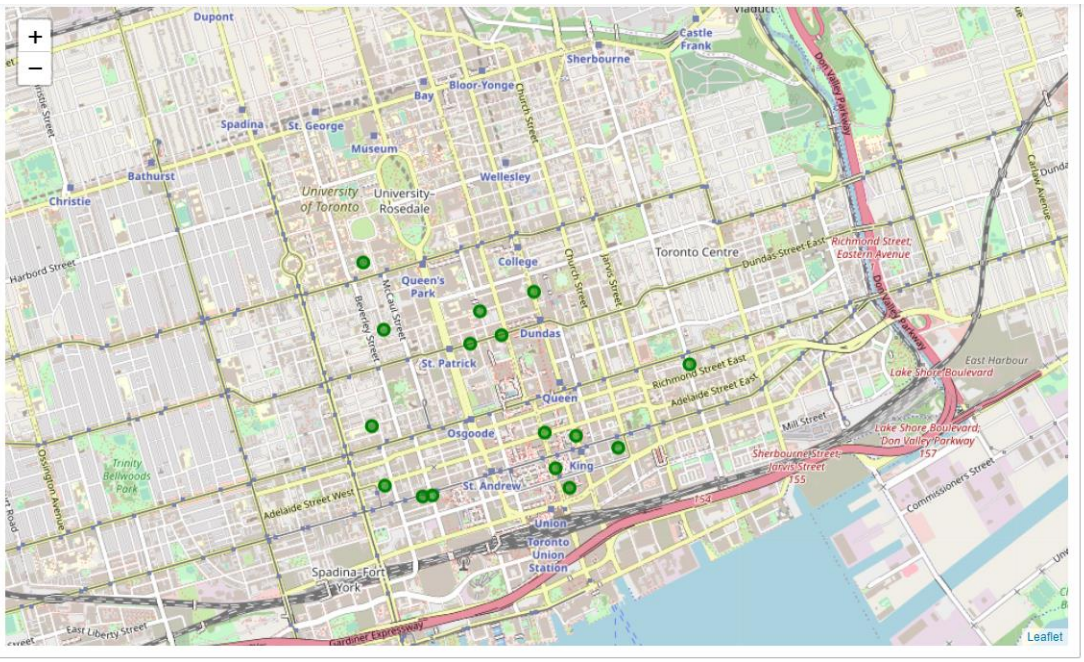
- The rent should be between CAD 1500 and CAD 2200
- The number of bedrooms and bathroom should be 2

Out[229]:



Finding Indian restaurants in Toronto using Foursquare and search query as ‘Indian'. Plotting the restaurant locations on the map.

Out[81]:



Combining all the plots on one map.

Red dots: Cluster 0

Purple dots: Cluster 1

Navy blue dots: Cluster 2

Light blue dots: Cluster 3

Aqua green dots: Cluster 4

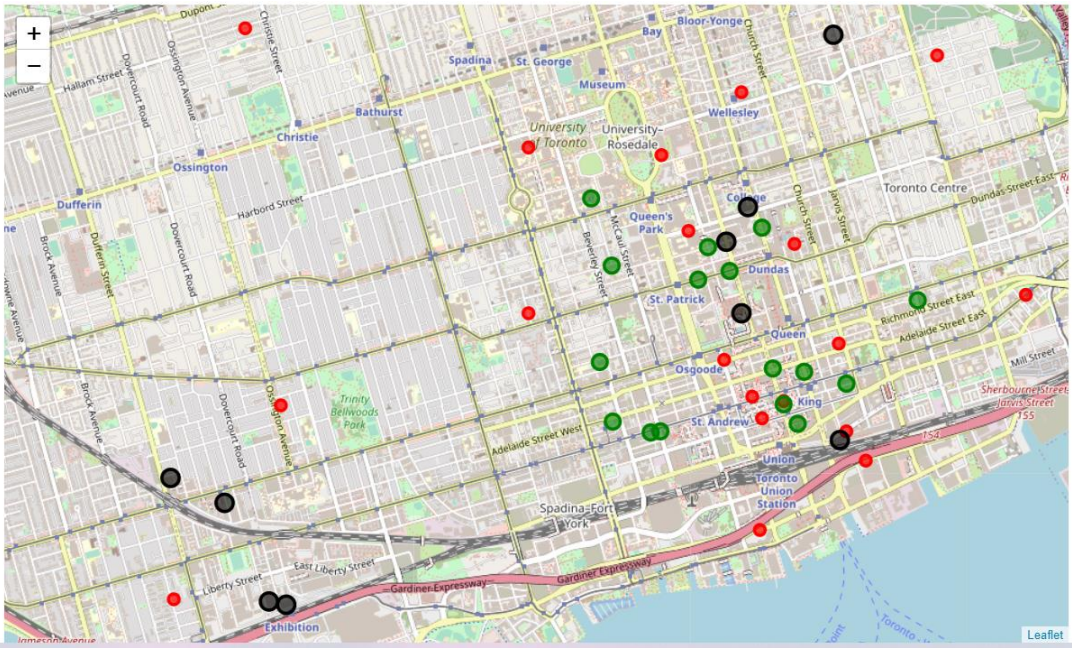
Brown dots: Cluster 5

Orange dots: Cluster 6

Black dots: Apartment locations

Green dots: Indian restaurant locations

Out[230]:



c. Analysis

Cluster 0:

Out[231]:

	Borough	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	Downtown Toronto	0	Coffee Shop	Pub	Park	Bakery	Theater	Restaurant	Café	Breakfast Spot	Farmers Market	Hotel
1	Downtown Toronto	0	Coffee Shop	Sushi Restaurant	Diner	Mexican Restaurant	Beer Bar	Sandwich Place	Burger Joint	Burrito Place	Café	Park
2	Downtown Toronto	0	Coffee Shop	Café	Tea Room	Middle Eastern Restaurant	Ramen Restaurant	Bookstore	Theater	Restaurant	Clothing Store	Sandwich Place
3	Downtown Toronto	0	Café	Gastropub	Coffee Shop	Creperie	Seafood Restaurant	Hotel	Farmers Market	Cosmetics Shop	Optical Shop	Ice Cream Shop
5	Downtown Toronto	0	Coffee Shop	Cocktail Bar	Cheese Shop	Beer Bar	Seafood Restaurant	Restaurant	Italian Restaurant	Farmers Market	Café	Bakery
6	Downtown Toronto	0	Coffee Shop	Sandwich Place	Café	Italian Restaurant	Bubble Tea Shop	Burger Joint	Ice Cream Shop	Department Store	Salad Place	Ramen Restaurant
7	Downtown Toronto	0	Grocery Store	Café	Park	Nightclub	Diner	Italian Restaurant	Restaurant	Baby Store	Candy Store	Athletics & Sports
8	Downtown Toronto	0	Coffee Shop	American Restaurant	Café	Restaurant	Concert Hall	Pizza Place	Seafood Restaurant	Steakhouse	Mediterranean Restaurant	Opera House
9	West Toronto	0	Pharmacy	Bakery	Wine Shop	Gym / Fitness Center	Park	Pizza Place	Middle Eastern Restaurant	Café	Brewery	Bar
10	Downtown Toronto	0	Coffee Shop	Aquarium	Hotel	Brewery	Plaza	Café	Park	Chinese Restaurant	Monument / Landmark	Beer Bar
11	West Toronto	0	Bar	Restaurant	Men's Store	Vietnamese Restaurant	Vegetarian / Vegan Restaurant	Café	Asian Restaurant	Yoga Studio	Brewery	Record Shop
12	East Toronto	0	Greek Restaurant	Coffee Shop	Italian Restaurant	Ice Cream Shop	Furniture / Home Store	Bookstore	Indian Restaurant	Grocery Store	Brewery	Restaurant
13	Downtown Toronto	0	Café	Coffee Shop	Seafood Restaurant	Hotel	Japanese Restaurant	Restaurant	Gym / Fitness Center	Gym	Deli / Bodega	Bakery
14	West Toronto	0	Café	Coffee Shop	Breakfast Spot	Stadium	Convenience Store	Performing Arts Venue	Pet Store	Climbing Gym	Burrito Place	Restaurant

Cluster 1:

```
In [232]: toronto_merged.loc[toronto_merged['Cluster Labels'] == 1, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
```

Out[232]:

	Borough	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
29	Central Toronto	1	Playground	Restaurant	Wine Shop	Cupcake Shop	Doner Restaurant	Dog Run	Distribution Center	Discount Store	Diner	Dim Sum Restaurant

Cluster 2:

```
In [233]: toronto_merged.loc[toronto_merged['Cluster Labels'] == 2, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
```

Out[233]:

	Borough	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
19	Central Toronto	2	Garden	Wine Shop	Cupcake Shop	Donut Shop	Doner Restaurant	Dog Run	Distribution Center	Discount Store	Diner	Dim Sum Restaurant

Cluster 3:

```
In [234]: toronto_merged.loc[toronto_merged['Cluster Labels'] == 3, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
```

Out[234]:

	Borough	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
21	Central Toronto	3	Sushi Restaurant	Park	Trail	Jewelry Store	Bus Line	Wine Shop	Doner Restaurant	Dog Run	Distribution Center	Discount Store

Cluster 4:

```
In [235]: toronto_merged.loc[toronto_merged['Cluster Labels'] == 4, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
```

Out[235]:

	Borough	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
33	Downtown Toronto	4	Park	Trail	Playground	Cuban Restaurant	Dog Run	Distribution Center	Discount Store	Diner	Dim Sum Restaurant	Dessert Shop

Cluster 5:

```
In [236]: toronto_merged.loc[toronto_merged['Cluster Labels'] == 5, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
```

Out[236]:

	Borough	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
18	Central Toronto	5	Park	Bus Line	Dim Sum Restaurant	Swim School	Doner Restaurant	Dog Run	Distribution Center	Discount Store	Diner	Dessert Shop

Cluster 6:

```
In [237]: toronto_merged.loc[toronto_merged['Cluster Labels'] == 6, toronto_merged.columns[[1] + list(range(5, toronto_merged.shape[1]))]]
```

Out[237]:

	Borough	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
4	East Toronto	6	Trail	Pub	Health Food Store	Wine Shop	Cupcake Shop	Doner Restaurant	Dog Run	Distribution Center	Discount Store	Diner

As the 7 clusters are not well diversified and Bangalore location venues do not match with cluster 1 to 6. Let us assume that Bangalore current location venues will be similar to cluster 0 in Toronto so that we have to look for apartments in cluster 0. Cluster 0 has covered most of the areas in Toronto so it won't be much problem to find place to stay.

The apartments available in cluster 0 with Indian restaurants nearby are mentioned below.

1. Bay St, Toronto, ON, Canada
2. 386 Yonge St, Toronto, ON M5G 2K2, Canada
3. , toronto m1m4j3 ON, Canada

```
In [221]: app_df[app_df['Address']=='Bay St, Toronto, ON, Canada']
```

Out[221]:

	Bedroom	Bathroom	Den	Address	Lat	Long	Price
931	2	2.0	0	Bay St, Toronto, ON, Canada	43.657298	-79.384365	1500.0

```
In [222]: app_df[app_df['Address']=='386 Yonge St, Toronto, ON M5G 2K2, Canada']
```

Out[222]:

	Bedroom	Bathroom	Den	Address	Lat	Long	Price
830	2	2.0	0	386 Yonge St, Toronto, ON M5G 2K2, Canada	43.659329	-79.382675	2100.0

```
In [226]: app_df[app_df['Address']==', toronto m1m4j3 ON, Canada']
```

Out[226]:

	Bedroom	Bathroom	Den	Address	Lat	Long	Price
1089	2	2.0	0	, toronto m1m4j3 ON, Canada	43.653226	-79.383184	1750.0

```
In [ ]:
```

As apartment 1 is located in cluster 0 has 4 Indian Restaurants are nearby and the rent is CAD 1500 which is affordable. Hence this can be selected.

4. Results

As per the choice of cluster, apartment location and nearby Indian restaurant apartment no 1 is selected. The selected option satisfies all our required criteria.

- The apartment is in a similar cluster as in Bangalore
- The rent is CAD 1500
- There are 2 bedrooms and 2 bathrooms
- There are multiple Indian Restaurant nearby

5. Discussion

Below are the assumptions made while working on this project.

- The venues in the neighborhoods in Bangalore and Toronto may not match as per the clustering groups as the culture and lifestyle are different.
- The clusters/groups are created for the neighborhoods of Toronto with available data. These can be improved or enhanced using refining the data.
- The graph used for selection of k in Elbow method is not quite significant.
- Collection of real time data is difficult so sample data from 2018 is used for rental apartment price and location.
- This project can be replicated for analysis of any locations having Foursquare data.
- The quality of the house and furniture can vary as per Price
- Facilities available and quality of the apartments are considered as equivalent

6. Conclusion

This project can be very helpful for people migrating to other city or country. This will help choosing best location of stay and finding great and interesting venues nearby. The process of learning and implementing in a real-time project has been a great experience for me. The tools used in the project are very powerful and can help anyone solving real time problems.