Table of contents

Introduction

The strength and vitality of the many neighbourhoods that make up Toronto, Ontario, Canada has earned the city its unofficial nickname of "the city of neighbourhoods." There are over 140 neighbourhoods officially recognized by the City of Toronto and upwards of 240 official and unofficial neighbourhoods within city limits. Before 1998, Toronto was a much smaller municipality and formed the core of Metropolitan Toronto. When the city amalgamated that year, Toronto grew to encompass the former municipalities of York, East York, North York, Etobicoke, and Scarborough. Each of these former municipalities still maintains, to a certain degree, its own distinct identity, and the names of these municipalities are still used by their residents, sometimes for disambiguation purposes as amalgamation resulted in duplicated street names. The area known as Toronto before the amalgamation is sometimes called the "old" City of Toronto, the Central District or simply "Downtown".

The "inner ring" suburbs of York and East York are older, predominantly middle-income areas, and ethnically diverse. Much of the housing stock in these areas consists of pre-World War II single-family houses and do not (obviously) post-war high-rises. Many of the neighbourhoods in these areas were built up as streetcar suburbs and contain many dense and mixed-use streets, some of which are one-way. They share many characteristics with sections of the "old" city outside the downtown core.

Map of Toronto including the former municipalities

The "outer ring" suburbs of Etobicoke, Scarborough, and North York are much more suburban in nature (although these boroughs are developing urban centres of their own, such as North York City Centre around Mel Lastman Square).

Problem Statement

The idea is to find neighborhood in Toronto city of Canada that has all the basic necessity shops within kilometers of the living place. People who are new to the city or shifting from another city to Toronto may require a place to live in. It might be difficult for them to find the neighborhood with all their necessities. The aim of the project is to divide the city neighborhoods in different categories according to shops and facilities available in the neighborhoods. The Foursquare API will be used to find all the nearby venues in neighborhoods and retrieve categories and count of shops in each category for each neighborhood.

Methodology

The aim of the project is to get the best neighborhood according to places near it. For that, the longitude and latitude of each neighborhood were required. The GeoSpatial data contains Postal codes, Longitude and Latitude data for all the 103 Boroughs of the Toronto city. The two datasets were combined and the new dataset with boroughs, neighborhoods and latitude and longitude was prepared.

The next step was to retrieve nearby places of each neighborhood. The Foursquare API was used for this purpose. The explore request was used to get nearby venues. Limit of 100 was set for each neighborhood nearby venues. The Foursquare API returned a JSON response of the explore query for all the neighborhoods. The information needed from the JSON response was name, longitude, latitude and category of each venue retrieved. The new data frame containing Neighborhood name, longitude, latitude, Venue name, Venue Category, Venue latitude and Venue longitude. As a cleaning step neighborhood with less than 5 nearby venues were removed from the dataset. The reason behind the step was to provide neighborhoods that has possibility of covering all the facilities and so neighborhoods with less than 5 venues were not perfect fit for the solution.

The category data was to be converted to numerical data for modeling the data. Categories data was one-hot encoded using pandas get_dummies function. Now, Data has Neighborhoods and each numerical category data. In the dataset, some neighborhoods were repeated as they had multiple venues and to compare neighborhoods, we have to combine all the same neighborhoods data into one row. For that purpose, mean of each neighborhood for each category was retrieved.

Import of all necessary libraries

```
from bs4 import BeautifulSoup
import requests
import pandas as pd
import numpy as np
```

BeautifulSoup Object to make request to website

```
URL = 'http://en.turkcewiki.org/wiki/List_of_postal_codes_of_Canada:_M'
page = requests.get(URL)

soup = BeautifulSoup(page.content, 'lxml')
```

```
tble = soup.find('table')
print(len(tble))
```

```
2
```

```
postal_codes=[]
boroughs = []
neighs = []
```


Web Scraping from the table

```
for neigh in tble.find_all('td'):
  sp = neigh.find('span')
  if(sp.text != 'Not assigned'):
    postal_codes.append(neigh.find('b').text)
    data = sp.text
    split_both = data.split("(")
    hoods = split_both[1].split(")")[0]
    hoods_data = hoods.replace("/",",")
    boroughs.append(split_both[0])
    neighs.append(hoods_data)
```

```
print(len(postal_codes))
print(len(boroughs))
print(len(neighs))
```

```
103
103
103
```

```
df = pd.DataFrame(
columns=['PostalCode','Borough','Neighbourhood'])
df
```



```
df = pd.DataFrame(
columns=['PostalCode','Borough','Neighbourhood'])
df
```

PostalCode	Borough	Neighbourhood

```
df['PostalCode'] =postal_codes
df['Borough'] = boroughs
df['Neighbourhood'] = neighs

df
```

	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	Queen's Park	Ontario Provincial Government
...	...	...	...
98	M8X	Etobicoke	The Kingsway , Montgomery Road , Old Mill North
99	M4Y	Downtown Toronto	Church and Wellesley
100	M7Y	East TorontoBusiness reply mail Processing Cen...	Enclave of M4L

GeoSpatial Data of Latitude and Longitude

```
from io import StringIO
url = 'http://cocl.us/Geospatial_data'
s=requests.get(url).content
c=pd.read_csv(StringIO(s.decode('utf-8')))
c
```

	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
...	...	...	...
98	M9N	43.706876	-79.518188
99	M9P	43.696319	-79.532242

Merging tables

```
df2= df.merge(c, left_on='PostalCode',right_on = 'Postal Code', how='left')
df2.drop(columns=['Postal Code'],inplace=True)
df2
```

	PostalCode	Borough	Neighbourhood	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	Queen's Park	Ontario Provincial Government	43.662301	-79.389494
...	...	...	...	...	...
98	M8X	Etobicoke	The Kingsway , Montgomery Road , Old Mill North	43.653654	-79.506944
99	M4Y	Downtown Toronto	Church and Wellesley	43.665860	-79.383160
100	M7Y	East TorontoBusiness reply mail Processing Cen...	Enclave of M4L	43.662744	-79.321558
101	M8Y	Etobicoke	Old Mill South , King's Mill Park , Sunnylea ,...	43.636258	-79.498509
102	M8Z	Etobicoke	Mimico NW , The Queensway West , South of Bloo...	43.628841	-79.520999

103 rows × 5 columns

Examine the North York Borough of Toronto

```
df3 = df2[df2['Borough']=='North York']
df3
```

	PostalCode	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
3	M6A	North York	Lawrence Manor , Lawrence Heights	43.718518	-79.464763
7	M3B	North York	Don Mills	43.745906	-79.352188
10	M6B	North York	Glencairn	43.709577	-79.445073
13	M3C	North York	Don Mills	43.725900	-79.340923
27	M2H	North York	Hillcrest Village	43.803762	-79.363452
28	M3H	North York	Bathurst Manor , Wilson Heights , Downsview North	43.754328	-79.442259
33	M2J	North York	Fairview , Henry Farm , Oriole	43.778517	-79.346556
34	M3J	North York	Northwood Park , York University	43.767980	-79.487262
39	M2K	North York	Bayview Village	43.786947	-79.385975
40	M3K	North York	Downsview	43.737473	-79.464763

import libaries

```
from geopy.geocoders import Nominatim
import folium
import json
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors

from sklearn.cluster import KMeans
```

Getting Longitude and latitude of Toronto City

```
address = 'Toronto'

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

```
The geograpical coordinate of Manhattan are 43.6534817, -79.3839347.
```

Clustering

K-means clustering with 5 clusters were used on the dataset. The features of clustering were those 7 categories retrieved on previous step. The frequency of occurrence of each category determined clusters of neighborhoods. The cluster which has high frequency of occurrence of these categories are better. These clusters will help in recognizing neighborhoods with needed category shop

Data

Data of boroughs and neighborhoods of the Toronto City would be retrieved from Wikipedia (https://en.wikipedia.org/wiki/List_of_neighbourhoods_in_Toronto). The data is there in form of tables with postal codes and names of neighborhoods in each of the Borough. The Geospatial data would be used to retrieve Longitude and Latitude of each neighborhood. Then, Foursquare API would be used to retrieved nearby venues of each neighborhood.

- **Wikipedia Data:** Columns Retrieved: Borough, Postal Code, Neighborhoods
- **Foursquare Data:** Latitude, Longitude, Venues, Category

Libraries used in the Project

- **Pandas:** For creating and manipulating dataframes.
- **Folium:** Python visualization library would be used to visualize the neighborhoods cluster distribution of using interactive leaflet map.
- **Scikit Learn:** For importing k-means clustering.
- **JSON:** Library to handle JSON files.
- **XML:** To separate data from presentation and XML stores data in plain text format.
- **Geocoder:** To retrieve Location Data.
- **Beautiful Soup and Requests:** To scrap and library to handle http requests.
- **Matplotlib:** Python Plotting Module.

Adding foursquare credential

```python
CLIENT_ID = 'K5BR4LYCYZRKQGKKR1UEXCFBB4BNZVO4XJ3NG3TZYYC3NXF2'
CLIENT_SECRET = 'N3HVKBN2421EIBC3NYGHQVOBKDRHAVYQYJENGMBUQ0ERMF5W'
VERSION = '20210515'
LIMIT = 100

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: K5BR4LYCYZRKQGKKR1UEXCFBB4BNZVO4XJ3NG3TZYYC3NXF2
CLIENT_SECRET:N3HVKBN2421EIBC3NYGHQVOBKDRHAVYQYJENGMBUQ0ERMF5W
```

longitude and lattitude of the first neighbourhood

```python
neighborhood_latitude = df3.loc[0, 'Latitude']
neighborhood_longitude = df3.loc[0, 'Longitude']

neighborhood_name = df3.loc[0, 'Neighbourhood']

print('Latitude and longitude values of {} are {}, {}.'.format(neighborhood_name,
                                                               neighborhood_latitude,
                                                               neighborhood_longitude))
```

request to foursquare API to explore venues near Parkwood

```python
LIMIT =100
radius =500
AUTH = 'IKEMNNFA2KMSR1AEWHKN2HUZEV3HCQB3GNZHGQKJ4EZ3XL3O'
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&oauth_token={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    AUTH,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
```

```python
results = requests.get(url).json()
results
```


```
{'meta': {'code': 200, 'requestId': '609e95c2112cbf088820e65e'},
 'notifications': [{'item': {'unreadCount': 0}, 'type': 'notificationTray'}],
 'response': {'groups': [{'items': [{'reasons': {'count': 0,
      'items': [{'reasonName': 'globalInteractionReason',
        'summary': 'This spot is popular',
        'type': 'general'}]},
```


Getting venues near all the neighbourhoods in north york borough

```python
def getNearbyVenues(names, latitudes, longitudes, radius=500):
    AUTH = 'IKEMNNFA2KMSR1AEWHKN2HUZEV3HCQB3GNZHGQKJ4EZ3XL3O'
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        url = 'https://api.foursquare.com/v2/venues/explore?&oauth_token={}&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            AUTH,
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        results = requests.get(url).json()["response"]['groups'][0]['items']

        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
```

Generating dataframe of all neighbourhoods

```python
north_york_venues = getNearbyVenues(names=df3['Neighbourhood'],
                                    latitudes=df3['Latitude'],
                                    longitudes=df3['Longitude'])
north_york_venues
```

```
Parkwoods
Victoria Village
Regent Park , Harbourfront
Lawrence Manor , Lawrence Heights
Ontario Provincial Government
Islington Avenue
Malvern , Rouge
Don Mills
Parkview Hill , Woodbine Gardens
Garden District, Ryerson
Glencairn
West Deane Park , Princess Gardens , Martin Grove , Islington , Cloverdale
Rouge Hill , Port Union , Highland Creek
Don Mills
Woodbine Heights
St. James Town
Humewood-Cedarvale
Eringate , Bloordale Gardens , Old Burnhamthorpe , Markland Wood
```

Getting count of venues near neighbourhoods

```
north_york_venues.groupby('Neighborhood').count()
```

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Agincourt	7	7	7	7	7	7
Alderwood , Long Branch	12	12	12	12	12	12
Bathurst Manor , Wilson Heights , Downsview North	34	34	34	34	34	34
Bayview Village	6	6	6	6	6	6
Bedford Park , Lawrence Manor East	54	54	54	54	54	54
...	...	...	...	...	...	...
Willowdale , Newtonbrook	3	3	3	3	3	3
Woburn	4	4	4	4	4	4
Woodbine Heights	16	16	16	16	16	16
York Mills , Silver Hills	3	3	3	3	3	3
York Mills West	5	5	5	5	5	5

97 rows × 6 columns

Getting top 5 venues of each neighbourhood and their frequencies

```
num_top_venues = 5

for hood in north_york_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = north_york_grouped[north_york_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')
```

```
----Agincourt----
            venue  freq
0  Breakfast Spot  0.14
1          Lounge  0.14
2  Fireworks Store  0.14
```

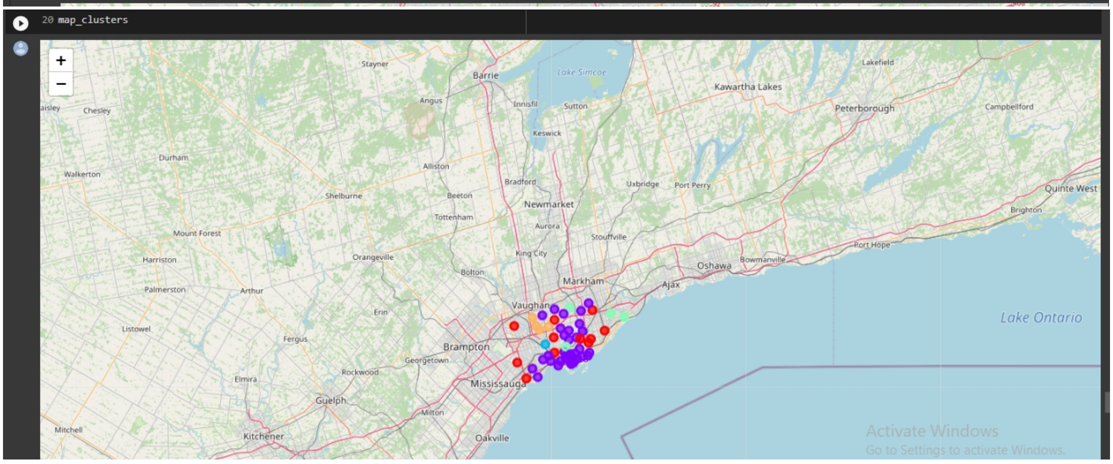
Neighbourhoods with their most common 10 venues

```
num_top_venues = 10

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

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))

neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = north_york_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

	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	Agincourt	Hardware Store	Breakfast Spot	Skating Rink	Fireworks Store	Lounge	Latin American Restaurant	Clothing Store	Empanada Restaurant	Drugstore	Dry Cleaner
1	Alderwood , Long Branch	Pizza Place	Skating Rink	Gym	Pharmacy	Sandwich Place	Athletics & Sports	Playground	Coffee Shop	Pub	Pool

Results and Discussion

There are 5 different clusters of neighborhoods. Red and Purple clusters have more neighborhoods compared to other clusters. There are basically 5 different types. The red clusters are mostly on the airport side of the City which seems less populated. Purple neighborhoods are near University of Toronto and beach side. This side is more dense than other sides. The yellow cluster is of neighborhoods which are very far from main city area. The sea blue cluster has only one neighborhood in it which is inside city region but it is only one neighborhood in the area. The Cyan clusters are nearly on the border of the city.



The results include 5 clusters and are of different properties and characteristics. The sea blue cluster has only one neighborhood and it is very deserted area. This area does not all the necessary facilities which makes it very weak candidate for the selection of this neighborhood. The Cyan cluster is at very end of the city which makes it very obvious for having less amenities so it is also not good for selection. The yellow cluster has very similar properties as Cyan s it is also a very bad candidate. There are two clusters remaining for the selection Red and Purple. The red cluster has no ATMs. The purple has few ATMs but is scarce in terms of Gyms and Shopping Malls. The red cluster is very scattered and purple is very dense in the area. The decision of choosing neighborhood now depends on distance, area of choice and which facilities are more important than others. For example, if Gyms and Shopping malls are more important and more frequently visited than ATMs and the person like to live in scattered area with some free space then neighborhoods from Red clusters will be more good choice over purple clusters. Then, to choose a neighborhood from the selected cluster would consist of consideration of proximity of work place. The one thing that was not considered in the discussion was number of restaurants. The reason was that there were many categories of restaurants in the City so it would clearly depend on the person to choose type of restaurant with his/her favorite food types. Here, I have considered generic restaurant category for clustering.

Limitation

The one limitation I can identify of this approach is that some small shops in small cities may not be registered on Foursquare and it would become difficult to take them into consideration while finding best fit of neighborhood.

Conclusion

The project overall helps person select best neighborhood to live in. The other aspect of the project may help shop owners and businessmen to determine what kind of shops would be required in the area. If a person could identify basic needs of people living in the neighborhood than one place with all those facilities can be built and would give guaranteed business. Finally, this project would help all the stakeholders to solve the problem and get the best solution.