Looking for similarities neighborhood for a Bank opening in New York City

March 15, 2019

1. Introduction

In this final project we will help to an Bank to evaluate where can open in New York. Recently the bank is operating in Toronto. For the Executive Directory it's important know how the City of New York is clustering, looking for the principal businnes are exits in Manhattan.

Based on the New York Neighborhoods, we have to explore and find some similarys Neighbourhoods, compared with Toronto.

2. About the Data

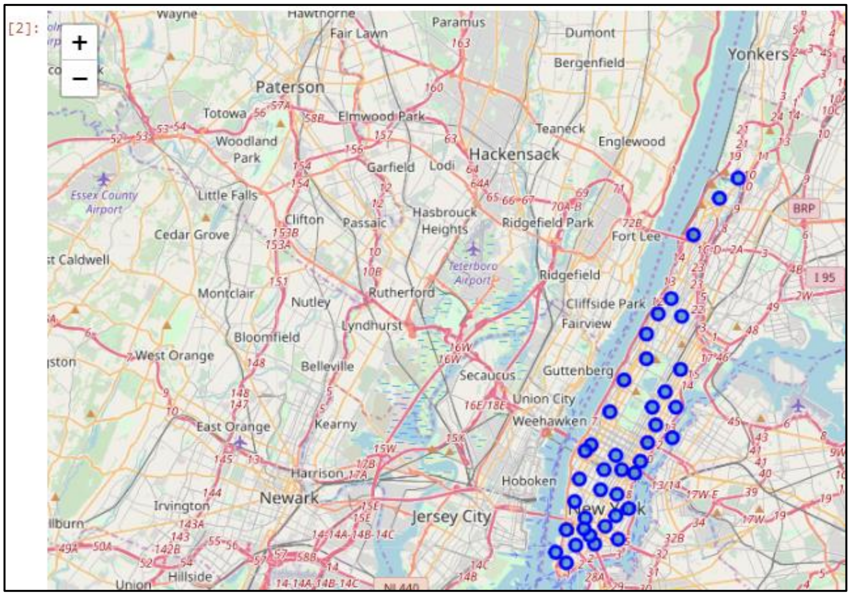
a. **New York Dataset**: First, we will download the Neighborhoods dataset from New York: A City of Neighborhoods. To handle this, we open as JSON file.

```
[1]: import json
     import pandas as pd

     !wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset

     with open('newyork_data.json') as json_data:
         newyork_data = json.load(json_data)

     neighborhoods_data = newyork_data['features']
     df_n = pd.DataFrame()

     for data in neighborhoods_data:
         borough = neighborhood_name = data['properties']['borough']
         neighborhood_name = data['properties']['name']

         neighborhood_latlon = data['geometry']['coordinates']
         neighborhood_lat = neighborhood_latlon[1]
         neighborhood_lon = neighborhood_latlon[0]

         df_n = df_n.append({'Borough': borough,
                             'Neighborhood': neighborhood_name,
                             'Latitude': neighborhood_lat,
                             'Longitude': neighborhood_lon,
                             'City': 'New York'
                            }, ignore_index=True)

     df_n = df_n[['Borough','Neighborhood','Latitude','Longitude','City']]
     df_n.head()
```

[1]:

	Borough	Neighborhood	Latitude	Longitude	City
0	Bronx	Wakefield	40.894705	-73.847201	New York
1	Bronx	Co-op City	40.874294	-73.829939	New York
2	Bronx	Eastchester	40.887556	-73.827806	New York
3	Bronx	Fieldston	40.895437	-73.905643	New York
4	Bronx	Riverdale	40.890834	-73.912585	New York

Then we put the data on a Folium Map to visualize the New York Neighborhoods.



b. Toronto Dataset: In second place, we need the neighbourhoods from Toronto, so for that we are going to download the postal code of each neighbourhood from Wikipedia through Web Scraping using BeautifulSoup libraries.

[3]:
```
import requests
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup

postal_codes = []

req = requests.get('https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M')
soup = BeautifulSoup(req.text, "html.parser")

postal_table = soup.find('table',{"class":"wikitable sortable"})

for row in postal_table.find_all('tr'):
    cols = row.find_all('td')
    if len(cols) == 3:
        postal_codes.append((cols[0].text.strip(), cols[1].text.strip(), cols[2].text.strip(),'Toronto'))

df_t = pd.DataFrame(postal_codes)
df_t.columns = ['Postcode', 'Borough','Neighbourhood','City']
df_t = df_t[df_t.Borough != 'Not assigned']
df_t = df_t.rename(columns={'Postcode': 'PostalCode'})
df_t.loc[df_t['Neighbourhood'] == "Not assigned", 'Neighbourhood'] = df_t['Borough']
df_t.head()
```

Then we JOIN the Toronto dataset with Geospatial data obtained from Cognitive Class.

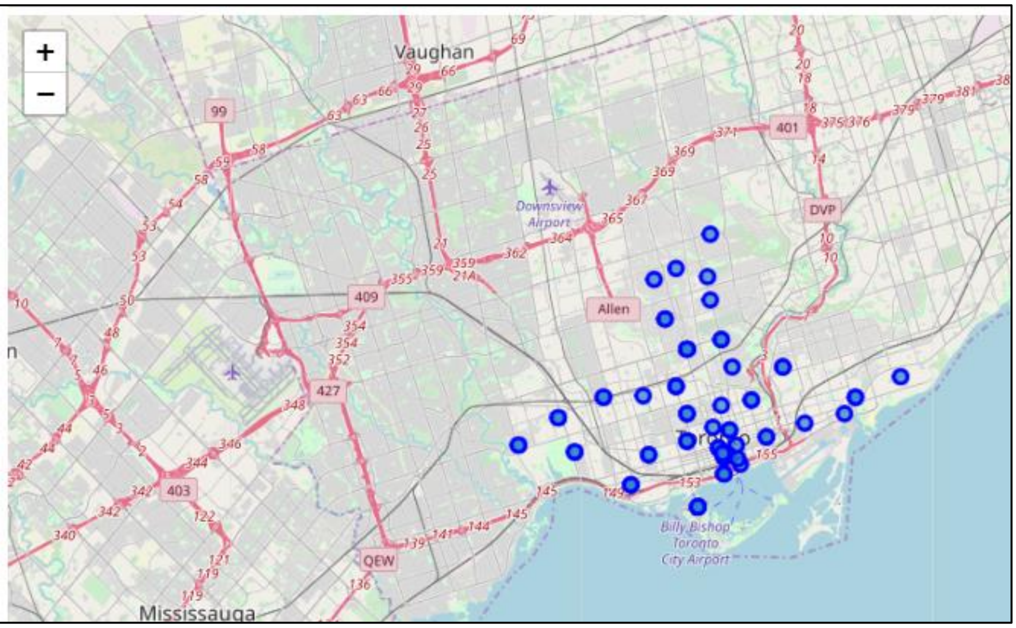
[4]:
```python
!wget -q -O 'canada_data.csv' https://cocl.us/Geospatial_data

df_coordinates = pd.read_csv("canada_data.csv")
df_coordinates = df_coordinates.rename(columns={'Postal Code':'PostalCode'})
df_coordinates = pd.merge(df_t, df_coordinates, on='PostalCode', how='inner')
df_coordinates = df_coordinates[['Borough','Neighbourhood','Latitude','Longitude','City']]
df_coordinates = df_coordinates.rename(columns={'Neighbourhood':'Neighborhood'})

toronto_data = df_coordinates[df_coordinates['Borough'].str.contains('Toronto')]
df_t  = toronto_data
df_t = df_t.reset_index(drop=True)
df_t.head()
```

[4]:

	Borough	Neighborhood	Latitude	Longitude	City
0	Downtown Toronto	Harbourfront	43.654260	-79.360636	Toronto
1	Downtown Toronto	Regent Park	43.654260	-79.360636	Toronto
2	Downtown Toronto	Ryerson	43.657162	-79.378937	Toronto
3	Downtown Toronto	Garden District	43.657162	-79.378937	Toronto
4	Downtown Toronto	St. James Town	43.651494	-79.375418	Toronto

And same as New York, we put the data on a Folium Map to visualize the Neighborhoods in Toronto.

[5]:



Below a brief summary from both data frames used to distinguish New York and Toronto dataset.

```
[6]: print('The data set from New York has {} rows.'.format(df_n.count().unique()))
     print('The data set from Toronto has {} rows.'.format(df_t.count().unique()))

     print('New York has {} unique Borough.'.format(len(df_n['Borough'].unique())))
     print('Toronto has {} unique Borough.'.format(len(df_t['Borough'].unique())))

     The data set from New York has [306] rows.
     The data set from Toronto has [74] rows.
     New York has 5 unique Borough.
     Toronto has 4 unique Borough.
```

Then we mixed both to handle for the analyze.

```
[7]: neighborhoods = pd.concat([df_n, df_t])
     neighborhoods = neighborhoods.reset_index(drop=True)
     neighborhoods.head()
```

[7]:

	Borough	Neighborhood	Latitude	Longitude	City
0	Bronx	Wakefield	40.894705	-73.847201	New York
1	Bronx	Co-op City	40.874294	-73.829939	New York
2	Bronx	Eastchester	40.887556	-73.827806	New York
3	Bronx	Fieldston	40.895437	-73.905643	New York
4	Bronx	Riverdale	40.890834	-73.912585	New York

3. Methodology

a. **Connecting to API Foursquare:** We will use the Foursquare API to get the nearby venues and k-means clustering algorithm to analyze the Neighbourhood.

```
[10]: neighborhoods_venues = getNearbyVenues(names=neighborhoods['Neighborhood'],
                                        latitudes=neighborhoods['Latitude'],
                                        longitudes=neighborhoods['Longitude'],
                                        city=neighborhoods['City'],
                                        radius=500
                                       )
```

Below show the existing venues from each Neighborhood with a Radius of 500.

```
[11]: neighborhoods_venues.groupby('Neighborhood').count().head()
```

[11]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Neighborhood City	Venue	Venue Latitude	Venue Longitude	Venue Category
Adelaide	100	100	100	100	100	100	100
Allerton	30	30	30	30	30	30	30
Annadale	11	11	11	11	11	11	11
Arden Heights	5	5	5	5	5	5	5
Arlington	8	8	8	8	8	8	8

4. Analyze Toronto and New York Neighborhoods

```
[13]: # one hot encoding
neighborhoods_onehot = pd.get_dummies(neighborhoods_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
neighborhoods_onehot['Neighborhood'] = neighborhoods_venues['Neighborhood']

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

neighborhoods_onehot.head()
```

[13]:

	Yoga Studio	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	...	Warehouse Store	Waste Facility	Watch Shop	Waterfront	Weight Loss Center	Whisky Bar	Wine Bar
0	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	0
2	0	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	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

5 rows × 455 columns

Analyze the mean of the frequency of occurrence of each category grouped by Neighborhood.

```
[15]: neighborhoods_grouped = neighborhoods_onehot.groupby('Neighborhood').mean().reset_index()
neighborhoods_grouped.head()
```

[15]:

	Neighborhood	Yoga Studio	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	...	Warehouse Store	Waste Facility	Watch Shop	Waterfront	Weight Loss Center	Whisky Bar
0	Adelaide	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
1	Allerton	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2	Annadale	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
3	Arden Heights	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
4	Arlington	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 455 columns

Search for Top 5 venues in each neighborhood.

```
[16]: num_top_venues = 5

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

5. Cluster Neighborhoods

Then we run k-means to cluster into 10 clusters the neighborhood.

```
[19]: # import k-means from clustering stage
      from sklearn.cluster import KMeans

      # set number of clusters
      kclusters = 10

      neighborhoods_grouped_clustering = neighborhoods_grouped.drop('Neighborhood', 1)

      # run k-means clustering
      kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(neighborhoods_grouped_clustering)

      # check cluster labels generated for each row in the dataframe
      kmeans.labels_[0:10]

[19]: array([0, 4, 3, 3, 0, 3, 3, 0, 4, 0], dtype=int32)
```

Now create an dataframe that includes the cluster with the top 10 venues for each neighborhood.

```
[20]: # add clustering labels
      neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

We check the new dataframe.

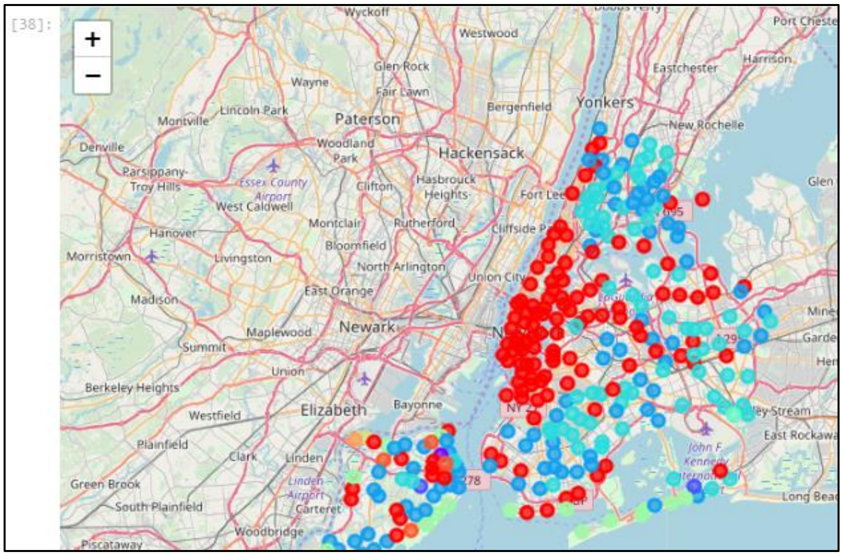
```
[21]: neighborhoods_merged = neighborhoods
      neighborhoods_merged.head()
```

[21]:

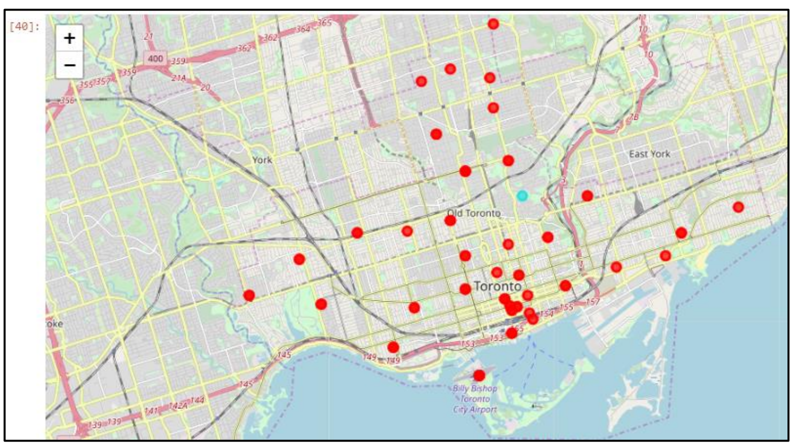
	Borough	Neighborhood	Latitude	Longitude	City
0	Bronx	Wakefield	40.894705	-73.847201	New York
1	Bronx	Co-op City	40.874294	-73.829939	New York
2	Bronx	Eastchester	40.887556	-73.827806	New York
3	Bronx	Fieldston	40.895437	-73.905643	New York
4	Bronx	Riverdale	40.890834	-73.912585	New York

a. Results

Cluster for New York:



Cluster for Toronto:



b. 5.2 Discussion

- We need more data about Toronto because in comparison with New York is too small.
- In the analyze can we add another factors like a rating from each venue to make more complex relations.
- Another important factor would be the estimated sales of each venue.
-

c. 5.3 Conclusions

- The cluster number #0 has similarity on their neighborhoods.
- The best place for the bank and make business with local markets are the Neighborhood in cluster #0 (points red color).