Capstone Project - The Battle of Neighborhoods

Analyzing Business Development

Areas in Toronto

Introduction/Business Problem

Toronto is the provincial capital of Ontario and the most populous city in Canada.The city is the anchor of the Golden Horseshoe, an urban agglomeration of 9,245,438 people (as of 2016) surrounding the western end of Lake Ontario. 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. Toronto is very attractive city for moving in or opening new business. Therefore, it is very important to understand business attractivness before opening new business or choosing the living area. So, in this project we will try to find attracctive neighborhoods which the lowest rate of fire incidents in Toronto.

The aim of this project is to find more attractive business development areas location for opening new business , especialy restaurant business, in Toronto, Canada. In this project , we will try to answer how to find the attractive neighborhood for the restaurant business, where restaurants are not amongst the most common venues.

Although, this project focused on restaurant business, it will be interested for individuals who try to find a attractive place with good neighborhoods to move in or establish a business.

Data

In this project, we will use real data from City of Toronto Open Data, Wikipedia and the Foursquare location data.

From City of Toronto Open Data we will use Business Improvement Areas from Wikipedia data about Neighborhood and finally, we wil use the Foursquare location data to fetch venues for the listed neigborhood.

The Business Improvement Areas dataset will need data wrangling and acquuistion, that's why we will merge it with data from Wikipedia and in combination with the Foursquare location data we will find the satisfying neigbhorhood for project's criteria.

Data Acquistion and Cleaning

Importing Libraries

In [1]:
```
# Importing Libraries

import requests
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
import random
import urllib
import json

#Command to install OpenCage Geocoder for fetching Lat and Lng of Neighborhood
!pip install opencage

#Importing OpenCage Geocoder
from opencage.geocoder import OpenCageGeocode

!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values

# use the inline backend to generate the plots within the browser
%matplotlib inline

#Importing Matplot lib and associated packages to perform Data Visualisation and Exploratory Data Analysis
import matplotlib as mpl
import matplotlib.pyplot as plt

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

#Importing folium to visualise Maps and plot based on Lat and Lng
import folium

#Requests to request web pages by making get requests to FourSquare REST Client
import requests

#To normalise data returned by FourSquare API
from pandas.io.json import json_normalize

#Importing KMeans from SciKit library to Classify neighborhoods into clusters
from sklearn.cluster import KMeans
```

Importing Dataframes from Toronto Open Data

In [2]:
```
import urllib
import json
import pandas as pd

# Get the dataset metadata by passing package_id to the package_search endpoint
# For example, to retrieve the metadata for this dataset:

url = "https://ckan0.cf.opendata.inter.prod-toronto.ca/api/3/action/package_show"
params = { "id": "9edb9628-1213-42bd-8352-5c4ed28e9e42"}
response = urllib.request.urlopen(url, data=bytes(json.dumps(params), encoding="utf-8"))
package = json.loads(response.read())
print(package["result"])

# Get the data by passing the resource_id to the datastore_search endpoint
# See https://docs.ckan.org/en/latest/maintaining/datastore.html for detailed parameters options
# For example, to retrieve the data content for the first resource in the datastore:

for idx, resource in enumerate(package["result"]["resources"]):
    if resource["datastore_active"]:
        url = "https://ckan0.cf.opendata.inter.prod-toronto.ca/api/3/action/datastore_search"
        p = { "id": resource["id"] }
        r = urllib.request.urlopen(url, data=bytes(json.dumps(p), encoding="utf-8"))
        data = json.loads(r.read())
        df = pd.DataFrame(data["result"]["records"])
        break
df
```

In [3]:
```
del df['X']#Deleting additional information that we do not need
del df['Y']
del df['OBJECTID']
del df['PARENT_AREA_ID']
```

In [4]:
```
# Importing Wikipedia data from "Segmenting and Clustering Neighborhoods in Toronto" Assignment
new_data=pd.read_csv('geo_data.csv', index_col=0) #Loading data
new_data.head() #printing data
```

Out[4]:

	PostalCode	Borough	Neighbourhood	Latitude	Longitude
0	M1B	Scarborough,Scarborough	Rouge,Malvern	43.806686	-79.194353
1	M1C	Scarborough,Scarborough,Scarborough	Highland Creek,Rouge Hill,Port Union	43.784535	-79.160497
2	M1E	Scarborough,Scarborough,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

In [5]:
```
new_data.columns = ['PostalCode','Borough','Neighbourhood', 'LATITUDE', 'LONGITUDE'] #Changing Columns Names
new_data.head()  #printing data
```

Out[5]:

	PostalCode	Borough	Neighbourhood	LATITUDE	LONGITUDE
0	M1B	Scarborough,Scarborough	Rouge,Malvern	43.806686	-79.194353
1	M1C	Scarborough,Scarborough,Scarborough	Highland Creek,Rouge Hill,Port Union	43.784535	-79.160497
2	M1E	Scarborough,Scarborough,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

Merging Data for starting Analyze

In [6]:
```
alldata= pd.merge(df,new_data, on=['LATITUDE','LONGITUDE'], how='outer') #merge dataframes to get new dataframe

alldata.head()
```

Out[6]:

	AREA_ATTR_ID	AREA_DESC	AREA_ID	AREA_LONG_CODE	AREA_NAME	AREA_SHORT_CODE	DATE_EFFECTIVE	LATITUDE	LONGITUDE	Shape__Area	Shape__Length	_id	geometry	PostalCode	Borough	N
0	26008975.0	Rogers Road	2481875.0	115-00	Rogers Road	115-00	2020-02-04T17:20:36	43.681791	-79.469890	351093.855469	5936.862796	2053.0	{"type": "Polygon", "coordinates": [[[-79.4682...	NaN	NaN	
1	26008974.0	Bloor-Yorkville	2481874.0	031-02	Bloor-Yorkville	031-02	2020-02-04T17:20:36	43.670401	-79.389159	918046.484375	6613.691633	2054.0	{"type": "Polygon", "coordinates": [[[-79.3872...	NaN	NaN	
2	26008973.0	Little Italy	2481873.0	020-01	Little Italy	020-01	2020-02-04T17:20:36	43.655397	-79.414394	232341.589844	3917.542802	2055.0	{"type": "Polygon", "coordinates": [[[-79.4205...	NaN	NaN	
3	26008972.0	Liberty Village	2481872.0	042-01	Liberty Village	042-01	2020-02-04T17:20:36	43.637870	-79.421265	797292.068406	4400.913504	2056.0	{"type": "Polygon", "coordinates": [[[-79.4246...	NaN	NaN	
4	26008971.0	Leslieville	2481871.0	093-01	Leslieville	093-01	2020-02-04T17:20:36	43.662460	-79.333555	351302.890625	6457.749078	2057.0	{"type": "Polygon", "coordinates": [[[-79.3240...	NaN	NaN	

In [7]:
```
alldata.shape
```

Out[7]: (186, 16)

In [8]:
```
# Analyzing Borough in the neighbourhoods in Toronto
alldata['Borough'].value_counts()
```

Out[8]:
```
North York                                                                                                13
Downtown Toronto                                                                                           9
North York,North York                                                                                      8
Scarborough,Scarborough,Scarborough                                                                        7
Downtown Toronto,Downtown Toronto                                                                          6
Scarborough                                                                                                6
West Toronto,West Toronto                                                                                  5
Central Toronto                                                                                            5
East York                                                                                                  4
North York,North York,North York                                                                           3
East Toronto                                                                                               3
York                                                                                                       3
Scarborough,Scarborough                                                                                    3
Downtown Toronto,Downtown Toronto,Downtown Toronto                                                         3
Etobicoke,Etobicoke,Etobicoke,Etobicoke,Etobicoke,Etobicoke,Etobicoke,Etobicoke                            2
East Toronto,East Toronto                                                                                  2
Etobicoke,Etobicoke,Etobicoke,Etobicoke,Etobicoke                                                          2
Etobicoke                                                                                                  2
Central Toronto,Central Toronto                                                                            2
Etobicoke,Etobicoke,Etobicoke,Etobicoke                                                                    2
Etobicoke,Etobicoke,Etobicoke                                                                              2
East York,East York                                                                                        1
Scarborough,Scarborough,Scarborough,Scarborough                                                            1
West Toronto,West Toronto,West Toronto                                                                     1
Downtown Toronto,Downtown Toronto,Downtown Toronto,Downtown Toronto,Downtown Toronto,Downtown Toronto       1
Central Toronto,Central Toronto,Central Toronto                                                            1
York,York,York,York                                                                                        1
Queen's Park                                                                                               1
York,York                                                                                                  1
Central Toronto,Central Toronto,Central Toronto,Central Toronto,Central Toronto                            1
Mississauga                                                                                                1
Etobicoke,Etobicoke                                                                                        1
Name: Borough, dtype: int64
```

In [9]:
```
print("Total Neighbourhood Count",len(alldata['Neighbourhood']),"Borough Count",len(alldata['Borough'].unique()))
```

```
Total Neighbourhood Count 186 Borough Count 33
```

Methodology

Starting our Analyze by using Pandas describe() to view some basic statistics

In [10]:
```
alldata.describe() #of course for our data this method is not correct
```

Out[10]:

	AREA_ATTR_ID	AREA_ID	LATITUDE	LONGITUDE	Shape__Area	Shape__Length	_id
count	8.300000e+01	8.300000e+01	186.000000	186.000000	8.300000e+01	83.000000	83.000000
mean	2.600693e+07	2.481834e+06	43.690943	-79.405603	8.887586e+05	6052.929883	2094.000000
std	2.410394e+01	2.410394e+01	0.048071	0.083829	2.580218e+06	6881.022836	24.103942
min	2.600689e+07	2.481793e+06	43.595984	-79.615819	5.455134e+04	1691.557634	2053.000000
25%	2.600691e+07	2.481814e+06	43.654339	-79.456510	1.653594e+05	3154.111534	2073.500000
50%	2.600693e+07	2.481834e+06	43.679560	-79.403966	2.793838e+05	4131.523729	2094.000000
75%	2.600695e+07	2.481854e+06	43.716083	-79.364417	5.186004e+05	6402.295632	2114.500000
max	2.600698e+07	2.481875e+06	43.836125	-79.160497	1.863314e+07	52454.576305	2135.000000

Setting Up Foursquare Credentials

In [13]:
```
#Four Square Credentials

CLIENT_ID = '14CXVCFRHS4UCISHLGXUBVNVAYFFTFEFZBAXM0UCMB2TAAFK'
CLIENT_SECRET = 'FFDYGV2FDTGL3HLBUGPG4DYTWHDXAVHYZ3OPVOEJFTEORUAP'
VERSION = '20200129'
LIMIT = 100

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

```
Your credentails:
CLIENT_ID: 14CXVCFRHS4UCISHLGXUBVNVAYFFTFEFZBAXM0UCMB2TAAFK
CLIENT_SECRET:FFDYGV2FDTGL3HLBUGPG4DYTWHDXAVHYZ3OPVOEJFTEORUAP
```

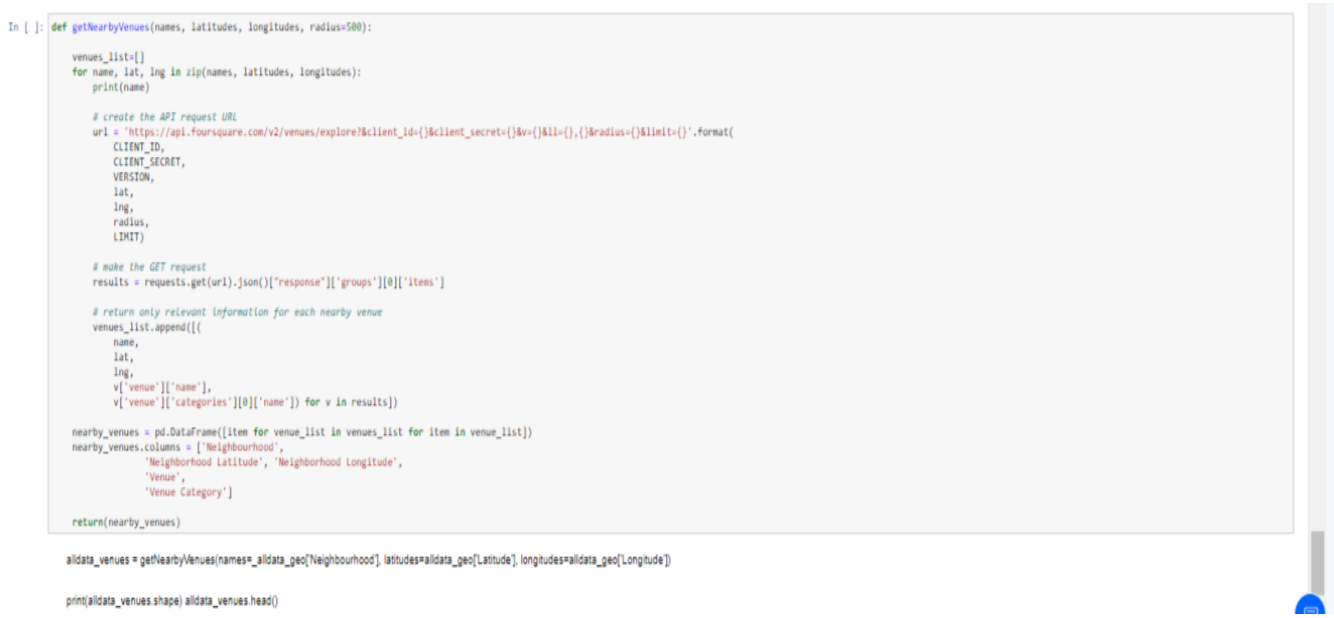
In [14]:
```
address = 'Toronto, Canada'

location = geocoder.geocode(address)
latitude = location[0]['geometry']['lat']
longitude = location[0]['geometry']['lng']

print('The geograpical coordinate of Toronto, Canada are {}, {}.'.format(latitude, longitude))
```

In [11]:
```
#Sorting the data by Borough as per neighborhood

alldata.sort_values(['Borough'], ascending = False, axis = 0, inplace = True )

alldata_top5 = alldata.iloc[1:6]
alldata_top5
```

Out[11]:

	AREA_ATTR_ID	AREA_DESC	AREA_ID	AREA_LONG_CODE	AREA_NAME	AREA_SHORT_CODE	DATE_EFFECTIVE	LATITUDE	LONGITUDE	Shape__Area	Shape__Length	_id	geometry	PostalCode
164	NaN	NaN	NaN	NaN	NaN	NaN	NaN	43.673185	-79.487262	NaN	NaN	NaN	NaN	M6N
156	NaN	NaN	NaN	NaN	NaN	NaN	NaN	43.693781	-79.428191	NaN	NaN	NaN	NaN	M6C
181	NaN	NaN	NaN	NaN	NaN	NaN	NaN	43.706876	-79.518188	NaN	NaN	NaN	NaN	M9N
157	NaN	NaN	NaN	NaN	NaN	NaN	NaN	43.689026	-79.453512	NaN	NaN	NaN	NaN	M6E
161	NaN	NaN	NaN	NaN	NaN	NaN	NaN	43.636847	-79.428191	NaN	NaN	NaN	NaN	M6K

In [12]:
```
alldata_low_Borough = alldata.tail(5)
alldata_low_Borough
```



```
In [ ]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

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

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighbourhood',
                  'Neighborhood Latitude', 'Neighborhood Longitude',
                  'Venue',
                  'Venue Category']

    return(nearby_venues)

alldata_venues = getNearbyVenues(names=_alldata_geo['Neighbourhood'], latitudes=alldata_geo['Latitude'], longitudes=alldata_geo['Longitude'])

print(alldata_venues.shape) alldata_venues.head()
```

Results and Discussion

The objective of this project was to help Individual one of the businessImprovement Areas in Toronto, and an appropriate neighborhood within the borough to set up a commercial establishment especially a restaurant business. TAs we see , I have choose not correct data, so analyze is fure and we understand that there are need to additional data.

Conclusion

This project tried to explore the most attractive business development areas to understand which neighborhood to choose in Toronto.