Capstone Project - Relocation from New York City to Toronto

For Applied Data Science Capstone Course by IBM/Coursera

Table of contents

Introduction: Business Problem Description

This project is designed to resolve the following assumed business case:

A US business manager who is working for a large financial company and also living in Manhatten, New York city. Recently his company has asked him to relocate to Toronto Canada to lead the efforts to set up a branch business there. He needs to decide where to set up the business office in Toronto. He also needs to select a new residential location for his family, preferrablely in a similar favorable location to his home in Manhatten in New York.

This business manager approached a specialized service company to help him to evaluate the location options in Toronto for both business office and his family home in Toronto. After signing the service contract, this service company starts to work on this project. This project could involve the following aspects:

1. First to evaluate and to understand the preferred characteristics from this US business manager on his current business office and family home in Manhatten in New York.
2. Second to collect and to evaluate information related to Toronto.
3. Compare the similarities and differences between New York and Toronto.
4. Based on US manager's preferred selection criterias to recommend location options on new business office and residential home in Toronto.

For each of above project aspects, we will conduct relevent data and analyze the data to provide quantitative asscessment. We will work the following data science project steps:

1. Collect neighborhood information between New York and Toronto.
2. Collect relevant venue information that could be required to compare between New York and Toronto.
3. Analyze the collected data to compare the similarities and differences between New York and Toronto.
4. Based on US business manager's proposed location selection preference criterias to recommend location options for both busines office and family home in Toronto.

Data Collection

Based on the above busienss problem descriptinon, we will collect the following data:

1. Neighborhood and venue information for New York city, including:
   (1) Neighborhood information for New York city.
   (2) Neighborhood information for Manhatten, New York.
   (3) Venues around existing business office and family home locations in Manhatten, New York city.

2. Neighborhood and venue information for Toronto, including:
   (1) Neighborhood information for Toronto.
   (2) Neighborhood information for Toronto financial central location.
   (3) Venue information for Toronto financial central location.

1. Collect neighborhood and venue Information for New York city

(1) Collect neighborhood Information for New York city

To get New York neighborhood information, we first download existing dataset newyork_data online. We convert the online dataset into pandas dataframe. The neighborhood information is then visualized on New York map.

In [1]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!conda install -c conda-forge geopy --yes # uncomment this line if you
from geopy.geocoders import Nominatim # convert an address into latitude

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pa

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

# import k-means from clustering stage
from sklearn.cluster import KMeans

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line i
import folium # map rendering library

print('Libraries imported.')
```

```
Collecting package metadata (repodata.json): done
Solving environment: done

# All requested packages already installed.

Libraries imported.
```

In [2]:
```
!wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
```

In [3]:
```
with open('newyork_data.json') as json_data:
    newyork_data = json.load(json_data)
```

In [4]:
```
neighborhoods_data = newyork_data['features']
```

The next is to transform the data into a pandas dataframe.

In [5]:
```
# define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
```

In [6]:
```
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]

    neighborhoods = neighborhoods.append({'Borough': borough,
                                          'Neighborhood': neighborhood_n
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}
```

In [7]:
```
# neighborhoods.head()
```

In [8]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(neighborhoods['Borough'].unique()),
        neighborhoods.shape[0]
    )
)
```

```
The dataframe has 5 boroughs and 306 neighborhoods.
```

Use geopy library to get the latitude and longitude values of New York City

In [9]:
```
address = 'New York City, NY'

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

```
The geograpical coordinate of New York City are 40.7127281, -74.006015
2.
```

In [10]:
```
# create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], ne
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_newyork)

map_newyork
```

Out[10]:

In [11]:
```
df_newyork = neighborhoods
df_newyork.head()
```

Out[11]:

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

(2) Collect neighborhood Information for Manhatten, New York

The Manhatten neighborhood data is a subset from New York neighborhood dataset. The data is visualized on a map.

In [12]:
```
manhattan_data = neighborhoods[neighborhoods['Borough'] == 'Manhattan'].
manhattan_data.head(40)
```

Out[12]:

	Borough	Neighborhood	Latitude	Longitude
0	Manhattan	Marble Hill	40.876551	-73.910660
1	Manhattan	Chinatown	40.715618	-73.994279
2	Manhattan	Washington Heights	40.851903	-73.936900
3	Manhattan	Inwood	40.867684	-73.921210
4	Manhattan	Hamilton Heights	40.823604	-73.949688
5	Manhattan	Manhattanville	40.816934	-73.957385
6	Manhattan	Central Harlem	40.815976	-73.943211
7	Manhattan	East Harlem	40.792249	-73.944182
8	Manhattan	Upper East Side	40.775639	-73.960508
9	Manhattan	Yorkville	40.775930	-73.947118
10	Manhattan	Lenox Hill	40.768113	-73.958860
11	Manhattan	Roosevelt Island	40.762160	-73.949168
12	Manhattan	Upper West Side	40.787658	-73.977059
13	Manhattan	Lincoln Square	40.773529	-73.985338
14	Manhattan	Clinton	40.759101	-73.996119

15	Manhattan	Midtown	40.754691	-73.981669
16	Manhattan	Murray Hill	40.748303	-73.978332
17	Manhattan	Chelsea	40.744035	-74.003116
18	Manhattan	Greenwich Village	40.726933	-73.999914
19	Manhattan	East Village	40.727847	-73.982226
20	Manhattan	Lower East Side	40.717807	-73.980890
21	Manhattan	Tribeca	40.721522	-74.010683
22	Manhattan	Little Italy	40.719324	-73.997305
23	Manhattan	Soho	40.722184	-74.000657
24	Manhattan	West Village	40.734434	-74.006180
25	Manhattan	Manhattan Valley	40.797307	-73.964286
26	Manhattan	Morningside Heights	40.808000	-73.963896
27	Manhattan	Gramercy	40.737210	-73.981376
28	Manhattan	Battery Park City	40.711932	-74.016869
29	Manhattan	Financial District	40.707107	-74.010665
30	Manhattan	Carnegie Hill	40.782683	-73.953256
31	Manhattan	Noho	40.723259	-73.988434
32	Manhattan	Civic Center	40.715229	-74.005415
33	Manhattan	Midtown South	40.748510	-73.988713
34	Manhattan	Sutton Place	40.760280	-73.963556
35	Manhattan	Turtle Bay	40.752042	-73.967708
36	Manhattan	Tudor City	40.746917	-73.971219
37	Manhattan	Stuyvesant Town	40.731000	-73.974052
38	Manhattan	Flatiron	40.739673	-73.990947
39	Manhattan	Hudson Yards	40.756658	-74.000111

In [13]:
```
address = 'Manhattan, NY'

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

```
The geograpical coordinate of Manhattan are 40.7900869, -73.9598295.
```

In [14]:
```
# create map of Manhattan using latitude and longitude values
map_manhattan = folium.Map(location=[latitude, longitude], zoom_start=11

# add markers to map
for lat, lng, label in zip(manhattan_data['Latitude'], manhattan_data['L
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan)

map_manhattan
```

Out[14]:

(3) Collect venues around existing business office and family home locations in Manhattan, New York city

We assume that the business office address is 100 Wall St New York, near New York Stock Exchange location.

Collect venues around existing business office in Manhattan, New

York city

We are using Foursqure API to collect venue information

In [15]:
```
CLIENT_ID = '0D33T0AVFXWBBXCZDHJ2IA32T5GG5IK1J5GYXBAIP14Z4KRF' # your Fo
CLIENT_SECRET = '0L2NKXGCDXVV3HLBJDOJDSNIOPCOODUIZOQQECZ4DC5F2ZIQ' # you
VERSION = '20180604'
```

In [16]:
```
address = '100 Wall St, New York, NY'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
40.7052203 -74.006799602293
```

In [17]:
```
LIMIT = 100 # limit of number of venues returned by Foursquare API
radius = 500 # define radius

url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    latitude,
    longitude,
    radius,
    LIMIT)
#url
```

In [18]:
```
results = requests.get(url).json()
#results
```

In [19]:
```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

In [20]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues)
```

In [21]:
```
# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.la
nearby_venues = nearby_venues.loc[:, filtered_columns]
# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_typ
# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.col

Manhattan_venues = nearby_venues
Manhattan_venues.head()
```

Out[21]:

	name	categories	lat	lng
0	Crown Shy	Restaurant	40.706187	-74.007490
1	sweetgreen	Salad Place	40.705586	-74.008382
2	Black Fox Coffee Co.	Coffee Shop	40.706573	-74.008155
3	La Colombe Torrefaction	Coffee Shop	40.705899	-74.008421
4	East River Esplanade	Pedestrian Plaza	40.704847	-74.004593

In [22]:
```
Manhattan_venues.shape
```

Out[22]:
```
(100, 4)
```

In [23]:
```
print('Around Manhattan Office, there are over {} venues were returned b
```

```
Around Manhattan Office, there are over 100 venues were returned by Fo
ursquare.
```

Explore Neighborhoods in Manhattan

In [24]:
```
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_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

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

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

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

    return(nearby_venues)
```

In [25]:
```
manhattan_venues = getNearbyVenues(names=manhattan_data['Neighborhood'],
                                   latitudes=manhattan_data['Latitude'],
                                   longitudes=manhattan_data['Longitude'
                                  )
```

```
Marble Hill
Chinatown
Washington Heights
Inwood
Hamilton Heights
Manhattanville
Central Harlem
East Harlem
Upper East Side
Yorkville
Lenox Hill
Roosevelt Island
Upper West Side
Lincoln Square
Clinton
Midtown
Murray Hill
Chelsea
Greenwich Village
East Village
Lower East Side
Tribeca
Little Italy
Soho
West Village
Manhattan Valley
Morningside Heights
Gramercy
Battery Park City
Financial District
Carnegie Hill
Noho
Civic Center
Midtown South
Sutton Place
Turtle Bay
Tudor City
Stuyvesant Town
Flatiron
Hudson Yards
```

In [26]:
```
print(manhattan_venues.shape)
manhattan_venues.head()
```

```
(3329, 7)
```

Out[26]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Marble Hill	40.876551	-73.91066	Arturo's	40.874412	-73.910271	Pizza Place
1	Marble Hill	40.876551	-73.91066	Bikram Yoga	40.876844	-73.906204	Yoga Studio
2	Marble Hill	40.876551	-73.91066	Tibbett Diner	40.880404	-73.908937	Diner
3	Marble Hill	40.876551	-73.91066	Starbucks	40.877531	-73.905582	Coffee Shop
4	Marble Hill	40.876551	-73.91066	Dunkin'	40.877136	-73.906666	Donut Shop

Output collected New York datasets as local data files

In [27]:
```
neighborhoods.to_csv("New_York_neighborhood_data.csv")
manhattan_data.to_csv("Manhattan_neighborhood_data.csv")
Manhattan_venues.to_csv("ManhattanOffice_venues_data.csv")
```

Collect neighborhood and venue information for Toronto

We will collect the following neighborhood and venue information for Toronto:
(1) Neighborhood information for Toronto.
(2) Neighborhood information for Toronto financial central location. (3) Venue information for Toronto financial central location.

(1) Collect neighborhood information for Toronto

The Toronto neighborhood data is scraped from a Wikipedia page. The data was structured into pandas dataframe

In [28]:
```
import lxml.html as lh
```

In [29]:
```
url='https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
page = requests.get(url)
doc = lh.fromstring(page.content)
tr_elements = doc.xpath('//tr')
```

In [30]:
```
col=[]
i=0
for t in tr_elements[0]:
    i+=1
    name=t.text_content()
    col.append((name,[]))
```

In [31]:
```
for j in range(1,len(tr_elements)):
    T=tr_elements[j]

    if len(T)!=3:
        break
    i=0
    for t in T.iterchildren():
        data=t.text_content()
        if i>0:
            try:
                data=int(data)
            except:
                pass
        col[i][1].append(data)
        i+=1
```

In [32]:
```
Dict={title:column for (title,column) in col}
df=pd.DataFrame(Dict)
```

In [33]:
```
df.columns = df.columns.str.strip()
df['Neighbourhood'] = df.Neighbourhood.str.replace('\n','')
df.drop(df.loc[df['Borough']=="Not assigned"].index, inplace=True)
```

In [34]:
```
df_group=df.groupby(['Postcode','Borough'],as_index=False)['Neighbourhoo
```

In [35]:
```
df_coordinate=pd.read_csv('Geospatial_Coordinates.csv')
df_coordinate.rename(columns={'Postal Code': 'Postcode'}, inplace=True)
df_toronto = df_group.merge(df_coordinate,on='Postcode')
```

In [36]:
```
df_toronto.rename(columns={'Neighbourhood': 'Neighborhood'}, inplace=Tru
```

In [37]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(df_toronto['Borough'].unique()),
        df_toronto.shape[0]
    )
)
```

```
The dataframe has 11 boroughs and 103 neighborhoods.
```

In [38]:
```
address = 'Toronto City, Canada'

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

```
The geograpical coordinate of Toronto City are 43.7189883, -79.44157.
```

In [39]:
```
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=10)
# add markers to map
for lat, lng, borough, neighborhood in zip(df_toronto['Latitude'], df_to
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toronto)

map_toronto
```

Out[39]:

In [40]:
```
df_toronto = df_toronto.drop(['Postcode'], axis=1)
```

In [41]:
```
df_toronto.head()
```

Out[41]:

	Borough	Neighborhood	Latitude	Longitude
0	Scarborough	Rouge,Malvern	43.806686	-79.194353
1	Scarborough	Highland Creek,Rouge Hill,Port Union	43.784535	-79.160497
2	Scarborough	Guildwood,Morningside,West Hill	43.763573	-79.188711
3	Scarborough	Woburn	43.770992	-79.216917
4	Scarborough	Cedarbrae	43.773136	-79.239476

In [42]:
```
df_toronto.shape
```

Out[42]:
```
(103, 4)
```

(2) Collect neighborhood information for Central Toronto, the financial central location

The Toronto financial center is located in Central Toronto borough. The Central Toronto neighborhood data is a subset from Toronto neighborhood dataset. It is visually shown on a map.

In [43]:
```
TorontoCentral_data = df_toronto[df_toronto['Borough'] == 'Central Toron
TorontoCentral_data.head()
```

Out[43]:

	Borough	Neighborhood	Latitude	Longitude
0	Central Toronto	Lawrence Park	43.728020	-79.388790
1	Central Toronto	Davisville North	43.712751	-79.390197
2	Central Toronto	North Toronto West	43.715383	-79.405678
3	Central Toronto	Davisville	43.704324	-79.388790
4	Central Toronto	Moore Park,Summerhill East	43.689574	-79.383160

In [44]:
```
address = 'Central Toronto, Toronto'

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

```
The geograpical coordinate of Central Toronto are 43.653963, -79.38720
7.
```

In [45]:
```
# create map of Toronto Central using latitude and longitude values
map_TorontoCentral = folium.Map(location=[latitude, longitude], zoom_sta

# add markers to map
for lat, lng, label in zip(TorontoCentral_data['Latitude'], TorontoCentr
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_TorontoCentral)

map_TorontoCentral
```

Out[45]:

(3) Collect venue information for Central Toronto, the financial center

In [46]:
```
print('The geograpical coordinate of Central Toronto are {}, {}.'.format
```

```
The geograpical coordinate of Central Toronto are 43.653963, -79.38720
7.
```

Let's create a function to repeat the same process to all the neighborhoods in Central Toronto

In [47]:
```
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_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

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

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

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

    return(nearby_venues)
```

Now write the code to run the above function on each neighborhood and create a new dataframe called *manhattan_venues*.

In [48]:
```
TorontoCentral_venues = getNearbyVenues(names=TorontoCentral_data['Neigh
                                        latitudes=TorontoCentral_data['Latitu
                                        longitudes=TorontoCentral_data['Longi
                                       )
```

```
Lawrence Park
Davisville North
North Toronto West
Davisville
Moore Park,Summerhill East
Deer Park,Forest Hill SE,Rathnelly,South Hill,Summerhill West
Roselawn
Forest Hill North,Forest Hill West
The Annex,North Midtown,Yorkville
```

In [49]:
```
print(TorontoCentral_venues.shape)
TorontoCentral_venues.head()
```

```
(112, 7)
```

Out[49]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Lawrence Park	43.728020	-79.388790	Lawrence Park Ravine	43.726963	-79.394382	Park
1	Lawrence Park	43.728020	-79.388790	Zodiac Swim School	43.728532	-79.382860	Swim School
2	Lawrence Park	43.728020	-79.388790	TTC Bus #162 - Lawrence-Donway	43.728026	-79.382805	Bus Line
3	Davisville North	43.712751	-79.390197	Sherwood Park	43.716551	-79.387776	Park
4	Davisville North	43.712751	-79.390197	Summerhill Market North	43.715499	-79.392881	Food & Drink Shop

In [50]:
```
TorontoCentral_venues.groupby('Neighborhood').count()
```

Out[50]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Davisville	35	35	35	35	35	35
Davisville North	7	7	7	7	7	7
Deer Park,Forest Hill SE,Rathnelly,South Hill,Summerhill West	16	16	16	16	16	16
Forest Hill North,Forest Hill West	4	4	4	4	4	4
Lawrence Park	3	3	3	3	3	3
Moore Park,Summerhill East	4	4	4	4	4	4
North Toronto West	18	18	18	18	18	18
Roselawn	1	1	1	1	1	1
The Annex,North Midtown,Yorkville	24	24	24	24	24	24

In [51]:
```
print('There are {} uniques categories.'.format(len(TorontoCentral_venue
```

```
There are 59 uniques categories.
```

In [ ]:
```

```

Output collected Toronto datasets as local data files

In [52]:
```
df_toronto.to_csv("Toronto_neighborhood_data.csv")
TorontoCentral_data.to_csv("TorontoCentral_neighborhood_data.csv")
TorontoCentral_data.to_json("TorontoCentral_neighborhood_data.json")
TorontoCentral_venues.to_csv("TorontoCentral_venues_data.csv")
TorontoCentral_venues.to_json("TorontoCentral_venues_data.json")
```

Methodology

In this project, we will use the following methods to analyze the data:

1. First, we will compare the total number of venues between Manhattan and Centrol Toronto. Such total venue number comparison could show the similarities and differences in terms of overall venue availability and distribution.
2. Second, for each neighborhood, we will rank top venues based on venue grouping. Such ranking information could show what venues are more available for a specific neighborhood. This local venue availability and concentration information will be valuable to match preferred venues with neighborhood options.
3. Third, we will apply k-means clustering to analyze the distribution of venue categories within different neighborhoods for both Manhattan and Central Toronto. This statistic significance based results could give good venue clustering distribution information for location selection.

To further refine the location selection criteria, we will specify the following US business manager's selection preferences for potential location selection options:

1. For new business office in Toronto, prefer location options at central financial district that exhibits similar venue clustering and vailability as those in Manhattan finncial district.
2. For new family home in Toronto, prefer location options that have good availability in recreation and social venues, such as park, musium and restaurants.

After proposing potential location options, we will also identify areas for further study to refine the location search by incorporating feedbacks and improvement suggestions from US business manager and his company decison makers after presenting this project report.

Analysis and Results

Analyze Manhatten Neighborhood and Results

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

In [53]:
```
# one hot encoding
manhattan_onehot = pd.get_dummies(manhattan_venues[['Venue Category']], 

# add neighborhood column back to dataframe
manhattan_onehot['Neighborhood'] = manhattan_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [manhattan_onehot.columns[-1]] + list(manhattan_onehot.c
manhattan_onehot = manhattan_onehot[fixed_columns]

manhattan_onehot.head()
```

Out[53]:

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Arcade	I
0	Marble Hill	0	0	0	0	0	0	0	
1	Marble Hill	0	0	0	0	0	0	0	
2	Marble Hill	0	0	0	0	0	0	0	
3	Marble Hill	0	0	0	0	0	0	0	
4	Marble Hill	0	0	0	0	0	0	0	

In [54]:
```
manhattan_grouped = manhattan_onehot.groupby('Neighborhood').mean().rese
#manhattan_grouped
```

In [55]:
```
manhattan_grouped.shape
```

Out[55]:
```
(40, 340)
```

First, let's write a function to sort the venues in descending order.

In [56]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

In [57]:
```
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
    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'] = manhattan_grouped['Neighbo

for ind in np.arange(manhattan_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venue

neighborhoods_venues_sorted.head(10)
```

Out[57]:

	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
0	Battery Park City	Park	Coffee Shop	Hotel	Gym	Memorial Site	Boat or Ferry	Italian Restaurant
1	Carnegie Hill	Coffee Shop	Pizza Place	Bar	Yoga Studio	Spa	Bakery	Bookstore
2	Central Harlem	African Restaurant	French Restaurant	Chinese Restaurant	Public Art	Cosmetics Shop	Seafood Restaurant	Bar
3	Chelsea	Coffee Shop	Italian Restaurant	Ice Cream Shop	Bakery	Nightclub	American Restaurant	Seafood Restaurant
4	Chinatown	Chinese Restaurant	Cocktail Bar	American Restaurant	Salon / Barbershop	Vietnamese Restaurant	Bubble Tea Shop	Spa
5	Civic Center	Gym / Fitness Center	Italian Restaurant	Coffee Shop	French Restaurant	Hotel	Sandwich Place	Sporting Goods Shop

Run *k*-means to cluster the Manhattan neighborhood into 5 clusters

In [58]:
```
kclusters = 5

manhattan_grouped_clustering = manhattan_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(manhattan_grou

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

Out[58]:
```
array([1, 1, 1, 1, 1, 1, 1, 0, 1, 1], dtype=int32)
```

In [59]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

manhattan_merged = manhattan_data

# merge toronto_grouped with toronto_data to add latitude/longitude for
manhattan_merged = manhattan_merged.join(neighborhoods_venues_sorted.set

#manhattan_merged = manhattan_merged.drop(['Unnamed:0'], axis=1)
manhattan_merged.head(20) # check the last columns!
```

Out[59]:

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue
0	Manhattan	Marble Hill	40.876551	-73.910660	4	Sandwich Place	Discount Store	Coffee Shop
1	Manhattan	Chinatown	40.715618	-73.994279	1	Chinese Restaurant	Cocktail Bar	American Restaurant
2	Manhattan	Washington Heights	40.851903	-73.936900	2	Café	Bakery	Mobile Phone Shop
3	Manhattan	Inwood	40.867684	-73.921210	2	Mexican Restaurant	Lounge	Café
4	Manhattan	Hamilton Heights	40.823604	-73.949688	0	Pizza Place	Coffee Shop	Mexican Restaurant
5	Manhattan	Manhattanville	40.816934	-73.957385	2	Italian Restaurant	Mexican Restaurant	Seafood Restaurant
6	Manhattan	Central Harlem	40.815976	-73.943211	1	African Restaurant	French Restaurant	Chinese Restaurant
7	Manhattan	East Harlem	40.792249	-73.944182	0	Mexican Restaurant	Bakery	Deli / Bodega
8	Manhattan	Upper East Side	40.775639	-73.960508	1	Italian Restaurant	Exhibit	Art Gallery

9	Manhattan	Yorkville	40.775930	-73.947118	1	Italian Restaurant	Gym	Coffee Shop
10	Manhattan	Lenox Hill	40.768113	-73.958860	1	Coffee Shop	Italian Restaurant	Pizza Place
11	Manhattan	Roosevelt Island	40.762160	-73.949168	1	Coffee Shop	Sandwich Place	Park
12	Manhattan	Upper West Side	40.787658	-73.977059	1	Italian Restaurant	Wine Bar	Bar
13	Manhattan	Lincoln Square	40.773529	-73.985338	1	Gym / Fitness Center	Theater	Café
14	Manhattan	Clinton	40.759101	-73.996119	1	Theater	Gym / Fitness Center	Hotel
15	Manhattan	Midtown	40.754691	-73.981669	1	Hotel	Coffee Shop	Cocktail Bar
16	Manhattan	Murray Hill	40.748303	-73.978332	1	Coffee Shop	Hotel	Sandwich Place
17	Manhattan	Chelsea	40.744035	-74.003116	1	Coffee Shop	Italian Restaurant	Ice Cream Shop
18	Manhattan	Greenwich Village	40.726933	-73.999914	1	Italian Restaurant	Clothing Store	Sushi Restaurant
19	Manhattan	East Village	40.727847	-73.982226	1	Bar	Wine Bar	Chinese Restaurant

Finally, let's visualize the resulting clusters

In [61]:
```
address = 'Manhattan, New York'

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

```
The geograpical coordinate of Manhattan are 40.7900869, -73.9598295.
```

In [62]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(manhattan_merged['Latitude'], manhatta
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_ht
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[62]:

Examine Manhattan venue clusters

Examine each cluster and determine the discriminating venue categories that distinguish each cluster. Based on the defining categories, you can then assign a name to each cluster. I will leave this exercise to you.

Cluster 1

In [63]:
```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 0, manhattan_
```

Out[63]:

	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
4	Hamilton Heights	Pizza Place	Coffee Shop	Mexican Restaurant	Café	Yoga Studio	Indian Restaurant	Sushi Restaurant
7	East Harlem	Mexican Restaurant	Bakery	Deli / Bodega	Pizza Place	Latin American Restaurant	Thai Restaurant	Spa
25	Manhattan Valley	Indian Restaurant	Pizza Place	Coffee Shop	Yoga Studio	Playground	Bar	Café

Cluster 2

In [64]:
```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 1, manhattan_
```

Out[64]:

	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	
1	Chinatown	Chinese Restaurant	Cocktail Bar	American Restaurant	Salon / Barbershop	Vietnamese Restaurant	Bubble Tea Shop	
6	Central Harlem	African Restaurant	French Restaurant	Chinese Restaurant	Public Art	Cosmetics Shop	Seafood Restaurant	
8	Upper East Side	Italian Restaurant	Exhibit	Art Gallery	Bakery	Gym / Fitness Center	Juice Bar	C
9	Yorkville	Italian Restaurant	Gym	Coffee Shop	Bar	Pizza Place	Sushi Restaurant	
10	Lenox Hill	Coffee Shop	Italian Restaurant	Pizza Place	Sushi Restaurant	Gym / Fitness Center	Sporting Goods Shop	B
11	Roosevelt Island	Coffee Shop	Sandwich Place	Park	Indie Theater	Dry Cleaner	Bus Stop	
12	Upper West Side	Italian Restaurant	Wine Bar	Bar	Vegetarian / Vegan	Mediterranean Restaurant	Bakery	

					Restaurant			
13	Lincoln Square	Gym / Fitness Center	Theater	Café	Concert Hall	Plaza	Italian Restaurant	
14	Clinton	Theater	Gym / Fitness Center	Hotel	American Restaurant	Italian Restaurant	Wine Shop	
15	Midtown	Hotel	Coffee Shop	Cocktail Bar	Theater	Clothing Store	American Restaurant	
16	Murray Hill	Coffee Shop	Hotel	Sandwich Place	Japanese Restaurant	Italian Restaurant	Gym / Fitness Center	
17	Chelsea	Coffee Shop	Italian Restaurant	Ice Cream Shop	Bakery	Nightclub	American Restaurant	
18	Greenwich Village	Italian Restaurant	Clothing Store	Sushi Restaurant	Café	Seafood Restaurant	Indian Restaurant	
19	East Village	Bar	Wine Bar	Chinese Restaurant	Mexican Restaurant	Ice Cream Shop	Pizza Place	
21	Tribeca	Italian Restaurant	Spa	Park	Café	American Restaurant	Boutique	
22	Little Italy	Bakery	Café	Bubble Tea Shop	Clothing Store	Sandwich Place	Salon / Barbershop	Mec
23	Soho	Clothing Store	Boutique	Art Gallery	Shoe Store	Women's Store	Italian Restaurant	G
24	West Village	Italian Restaurant	New American Restaurant	American Restaurant	Cosmetics Shop	Wine Bar	Jazz Club	
27	Gramercy	Bar	Italian Restaurant	Cocktail Bar	American Restaurant	Pizza Place	Bagel Shop	
28	Battery Park City	Park	Coffee Shop	Hotel	Gym	Memorial Site	Boat or Ferry	
29	Financial District	Coffee Shop	Hotel	Wine Shop	Steakhouse	Gym	Cocktail Bar	F
30	Carnegie Hill	Coffee Shop	Pizza Place	Bar	Yoga Studio	Spa	Bakery	
31	Noho	Italian Restaurant	French Restaurant	Sushi Restaurant	Cocktail Bar	Bookstore	Grocery Store	
32	Civic Center	Gym / Fitness Center	Italian Restaurant	Coffee Shop	French Restaurant	Hotel	Sandwich Place	G
	Midtown	Korean		Dessert	American	Japanese		

33	South	Restaurant	Hotel	Shop	Restaurant	Restaurant	Hotel Bar	
34	Sutton Place	Gym / Fitness Center	Italian Restaurant	Furniture / Home Store	Indian Restaurant	Juice Bar	Gym	F
35	Turtle Bay	Italian Restaurant	Steakhouse	Sushi Restaurant	Coffee Shop	Wine Bar	Ramen Restaurant	
38	Flatiron	Gym	Yoga Studio	American Restaurant	Japanese Restaurant	Clothing Store	Gym / Fitness Center	Cy
39	Hudson Yards	American Restaurant	Italian Restaurant	Gym / Fitness Center	Café	Spanish Restaurant	Restaurant	

Cluster 3

In [65]:
```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 2, manhattan_
```

Out[65]:

	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 Mos Commo Venu
2	Washington Heights	Café	Bakery	Mobile Phone Shop	Grocery Store	Spanish Restaurant	Coffee Shop	Tapa Restaura
3	Inwood	Mexican Restaurant	Lounge	Café	Pizza Place	Deli / Bodega	Wine Bar	America Restaura
5	Manhattanville	Italian Restaurant	Mexican Restaurant	Seafood Restaurant	Park	Coffee Shop	Deli / Bodega	Supermark
20	Lower East Side	Coffee Shop	Ramen Restaurant	Pizza Place	Café	Japanese Restaurant	Art Gallery	Bake
26	Morningside Heights	Coffee Shop	American Restaurant	Park	Bookstore	Burger Joint	Food Truck	Del Bodeg
36	Tudor City	Park	Mexican Restaurant	Café	Greek Restaurant	Hotel	Dog Run	Sus Restaura

Cluster 4

In [66]:
```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 3, manhattan_
```

Out[66]:

	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	8 C
37	Stuyvesant Town	Bar	Park	Playground	German Restaurant	Basketball Court	Baseball Field	Harbor / Marina	

Cluster 5

In [67]:
```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 4, manhattan_
```

Out[67]:

	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	
0	Marble Hill	Sandwich Place	Discount Store	Coffee Shop	Yoga Studio	Steakhouse	Supplement Shop	Shopping Mall	R

Analyze Central Toronto Neighborhood and Results

In [68]:
```
TorontoCentral_venues.head()
```

Out[68]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Lawrence Park	43.728020	-79.388790	Lawrence Park Ravine	43.726963	-79.394382	Park
1	Lawrence Park	43.728020	-79.388790	Zodiac Swim School	43.728532	-79.382860	Swim School
2	Lawrence Park	43.728020	-79.388790	TTC Bus #162 - Lawrence-Donway	43.728026	-79.382805	Bus Line
3	Davisville North	43.712751	-79.390197	Sherwood Park	43.716551	-79.387776	Park
4	Davisville North	43.712751	-79.390197	Summerhill Market North	43.715499	-79.392881	Food & Drink Shop

In [69]:
```
# one hot encoding
Toronto_onehot = pd.get_dummies(TorontoCentral_venues[['Venue Category']

# add neighborhood column back to dataframe
Toronto_onehot['Neighborhood'] = TorontoCentral_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [Toronto_onehot.columns[-1]] + list(Toronto_onehot.colum
Toronto_onehot = Toronto_onehot[fixed_columns]

Toronto_onehot.head()
```

Out[69]:

	Neighborhood	American Restaurant	BBQ Joint	Bagel Shop	Breakfast Spot	Brewery	Burger Joint	Bus Line	Café	Cheese Shop	Re
0	Lawrence Park	0	0	0	0	0	0	0	0	0	
1	Lawrence Park	0	0	0	0	0	0	0	0	0	
2	Lawrence Park	0	0	0	0	0	0	1	0	0	
3	Davisville North	0	0	0	0	0	0	0	0	0	
4	Davisville North	0	0	0	0	0	0	0	0	0	

In [70]:
```
Toronto_grouped = Toronto_onehot.groupby('Neighborhood').mean().reset_in
Toronto_grouped
```

Out[70]:

	Neighborhood	American Restaurant	BBQ Joint	Bagel Shop	Breakfast Spot	Brewery	Burger Joint	Bus Line	
0	Davisville	0.000000	0.000000	0.000000	0.000000	0.028571	0.000000	0.000000	0.
1	Davisville North	0.000000	0.000000	0.000000	0.142857	0.000000	0.000000	0.000000	0.
2	Deer Park,Forest Hill SE,Rathnelly,South Hill,...	0.062500	0.000000	0.062500	0.000000	0.000000	0.000000	0.000000	0.
3	Forest Hill North,Forest Hill West	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
4	Lawrence Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.333333	0.
5	Moore Park,Summerhill East	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
6	North Toronto West	0.000000	0.000000	0.055556	0.000000	0.000000	0.000000	0.000000	0.
7	Roselawn	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
8	The Annex,North Midtown,Yorkville	0.041667	0.041667	0.000000	0.000000	0.000000	0.041667	0.000000	0.

In [71]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

In [72]:
```
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
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
TorontoCentral_venues_sorted = pd.DataFrame(columns=columns)
```

```
TorontoCentral_venues_sorted = pd.DataFrame(columns=columns)
TorontoCentral_venues_sorted['Neighborhood'] = Toronto_grouped['Neighbor

for ind in np.arange(Toronto_grouped.shape[0]):
    TorontoCentral_venues_sorted.iloc[ind, 1:] = return_most_common_venu

TorontoCentral_venues_sorted.head(10)
```

Out[72]:

	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 Mo Comm Ven
0	Davisville	Coffee Shop	Sandwich Place	Dessert Shop	Italian Restaurant	Pizza Place	Café	Su Restaur
1	Davisville North	Sandwich Place	Hotel	Breakfast Spot	Gym	Park	Clothing Store	Food Dr Sh
2	Deer Park,Forest Hill SE,Rathnelly,South Hill,...	Coffee Shop	Pub	American Restaurant	Sports Bar	Vietnamese Restaurant	Fried Chicken Joint	Light F Stat
3	Forest Hill North,Forest Hill West	Trail	Jewelry Store	Sushi Restaurant	Mexican Restaurant	Yoga Studio	Dessert Shop	Hist Muse
4	Lawrence Park	Swim School	Bus Line	Park	Yoga Studio	Dessert Shop	Hotel	Hist Muse
5	Moore Park,Summerhill East	Tennis Court	Park	Summer Camp	Playground	Yoga Studio	Dessert Shop	Hist Muse
6	North Toronto West	Clothing Store	Coffee Shop	Yoga Studio	Salon / Barbershop	Bagel Shop	Chinese Restaurant	Dess Sh
7	Roselawn	Garden	Yoga Studio	Dessert Shop	Ice Cream Shop	Hotel	History Museum	G
8	The Annex,North Midtown,Yorkville	Sandwich Place	Coffee Shop	Café	Pizza Place	American Restaurant	Park	P

Run *k*-means to cluster the Central Toronto neighborhood into 5 clusters

In [73]:
```
kclusters = 5

Toronto_grouped_clustering = Toronto_grouped.drop('Neighborhood', 1)

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

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

Out[73]:
```
array([0, 0, 0, 4, 3, 2, 0, 1, 0], dtype=int32)
```

In [74]:
```
# add clustering labels
TorontoCentral_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

Toronto_merged = TorontoCentral_data

# merge toronto_grouped with toronto_data to add latitude/longitude for
Toronto_merged = Toronto_merged.join(TorontoCentral_venues_sorted.set_in

#manhattan_merged = manhattan_merged.drop(['Unnamed:0'], axis=1)
Toronto_merged.head() # check the last columns!
```

Out[74]:

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4tl Co
0	Central Toronto	Lawrence Park	43.728020	-79.388790	3	Swim School	Bus Line	Park	
1	Central Toronto	Davisville North	43.712751	-79.390197	0	Sandwich Place	Hotel	Breakfast Spot	
2	Central Toronto	North Toronto West	43.715383	-79.405678	0	Clothing Store	Coffee Shop	Yoga Studio	Barb
3	Central Toronto	Davisville	43.704324	-79.388790	0	Coffee Shop	Sandwich Place	Dessert Shop	Res
4	Central Toronto	Moore Park,Summerhill East	43.689574	-79.383160	2	Tennis Court	Park	Summer Camp	Play

In [75]:
```
address = 'Central Toronto, Toronto'

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

```
The geograpical coordinate of Central Toronto are 43.653963, -79.38720
7.
```

In [76]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(Toronto_merged['Latitude'], Toronto_me
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_ht
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[76]:

In [ ]:

Examine Central Toronto venue clusters

Examine each cluster and determine the discriminating venue categories that distinguish each cluster. Based on the defining categories, you can then assign a name to each cluster. I will leave this exercise to you.

Cluster 1

In [77]:
```
Toronto_merged.loc[Toronto_merged['Cluster Labels'] == 0, Toronto_merged
```

Out[77]:

	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 Mo Comm Ven
1	Davisville North	Sandwich Place	Hotel	Breakfast Spot	Gym	Park	Clothing Store	Food Dr Sh
2	North Toronto West	Clothing Store	Coffee Shop	Yoga Studio	Salon / Barbershop	Bagel Shop	Chinese Restaurant	Dess Sh
3	Davisville	Coffee Shop	Sandwich Place	Dessert Shop	Italian Restaurant	Pizza Place	Café	Su Restaur
5	Deer Park,Forest Hill SE,Rathnelly,South Hill,...	Coffee Shop	Pub	American Restaurant	Sports Bar	Vietnamese Restaurant	Fried Chicken Joint	Light F Stat
8	The Annex,North Midtown,Yorkville	Sandwich Place	Coffee Shop	Café	Pizza Place	American Restaurant	Park	P

Cluster 2

In [78]:
```
Toronto_merged.loc[Toronto_merged['Cluster Labels'] == 1, Toronto_merged
```

Out[78]:

	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 I Com V
6	Roselawn	Garden	Yoga Studio	Dessert Shop	Ice Cream Shop	Hotel	History Museum	Gym	G Resta

Cluster 3

In [79]:
```
Toronto_merged.loc[Toronto_merged['Cluster Labels'] == 2, Toronto_merged
```

Out[79]:

	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 Co
4	Moore Park,Summerhill East	Tennis Court	Park	Summer Camp	Playground	Yoga Studio	Dessert Shop	History Museum	

Cluster 4

In [80]:
```
Toronto_merged.loc[Toronto_merged['Cluster Labels'] == 3, Toronto_merged
```

Out[80]:

	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 M Comm Ve
0	Lawrence Park	Swim School	Bus Line	Park	Yoga Studio	Dessert Shop	Hotel	History Museum	C

Cluster 5

In [81]:
```
Toronto_merged.loc[Toronto_merged['Cluster Labels'] == 4, Toronto_merged
```

Out[81]:

	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 Cor
7	Forest Hill North,Forest Hill West	Trail	Jewelry Store	Sushi Restaurant	Mexican Restaurant	Yoga Studio	Dessert Shop	History Museum	

In [ ]:
```

```

Result Discussion and Recommendation

Result Discussion

Based on the outlined analysis methodology, we will discuss the results as follows:

1. Total number of venue comparison between Manhattan and Central Toronto:

(1) Both Manhattan and Central Toronto show similarities in large numbers of diverse venue in their neighborhoods, reflecting their leading business and cultural center status in their respective countries.
(2) However, based on individual neighborhood venue data comparison, Manhattan has order of magnitude more of total number of venue, reflecting that Manhattan has much more number of neighborhood (40 for Manhattan and 9 for Central Toronto) and much higher working and residential population than Central Toronto.
(3) Based on the above comparison, there should be good location options for new business office and family home in Central Toronto. However, the selection options in Central Toronto could still be much less than Manhattan since the total number of venue options in Central Toronto is much less than the total number of venue in Manhattan.

2. Venue ranking information and their implications:

(1) For Financial district in Manhattan, the top ranking venues include coffee shop, hotel, and various restaurants, reflecting those venues serving working and traveling populations. Will select new busienss office location options in Central Toronto with the similar venue ranking.

(2) For high-end residential neighborhood in Manhattan, such as Upper East Side next to famious New York Central Park, the top venues include restaurant, exhibit and art gallery, and fitness facilities. These venues are also preferred by the busienss manager for their new family home location optons in Toronto.

(3) To match the venue preference, the new business office location will clearly be at relatively small area of financial district in Central Toronto where there are high concentration of office buildings, hotels, restaurants and coffee shop.

(4) For family home location options in Central Toronto, the choices could be relatively hard to match Upper East Side neighborhood in Manhattan that are close to Park, musium and restaurants. Although downtown area in Central Toronto could offer good restaurant and musium venue, but downtown could be far away fro large park venue comparable to Manhattan Central Park. On the other hand, the large park venue in Toronto could be more available in a far distance areas from Toronto downtown and from restaurant and musium venues.

3. K-means venue clustering results and their implications:

(1) K-means venue clustering analysis provides consistent results with those from venue ranking analysis. It provides convincing confirmation regarding observations from the venue ranking analysis.

(2) In addition, the clustering results reveal that the neighborhood of Davisville North offers good venues of Park, grovery shopping and dancing and fitness facilities. Such venue availability could be suitable for new family home location in Central Toronto.

Recommendations:

Based on the above result discussions, we will propose the following recommendations regarding potential location options in Central Toronto for US business manager to consider:
(1) For new business office in Central Toronto, the clear option is at the financial district of Central Toronto, with plenty venues of office building, hotel, and restaurants.
(2) For new family home, the potential location options could be in the neighborhood of Davisville North or Davisville, where there are revnues of park, grocery shopping, restaurants and fitness facilities.

Suggestions for further study:

```
(1) After presenting the project, need to solocit feedbacks from
the US business manager and his company's decision makers.
(2) To verify and to evaluate location options, suggest to arran
ge visits to Central Toronto by the US business manager and his
family.
(3) May need to expand this project to inlcude other important f
actors in location selection, such as cost and transportation co
nvininces.
```

Conclusion

Based on data collection and analysis, this project provides detail evaluation of relocation options for US business manager for office and family home location selection.

Although Toronto also offers large number of venue for business and family life, comparison to Manhattan in New York, there are still some limitations, especially for family home location considerations. There coudl be tradeoff considerations to balance the preference for various venues.

While the report presents recommendations, it also suggests to further improve this study by getting feedbacks, arranging visits to Toronto, and to consider other important factors.

With further iterative data analysis and evaluation, we believe that Toronto could offer very satisfactory options for both new business office and family home selection.