Capstone Project - The Battle of the Neighborhoods

Applied Data Science Capstone by IBM/Coursera

1.Introduction

In the course of Applied Data Science Capstone,through weekly assignments and hands-on we explored New York City and the city of Toronto and segmented and clustered their neighborhoods.In the final project, our target is to compare the neighborhoods of the two cities and determine how similar or dissimilar they are. Both cities are very diverse and are the financial capitals of their respective countries. We have analyzed and compared the neighbourhoods of Toronto and Brooklyn, a Borough of NY for best 10 most common Venues.

In this project, we will implement the basic analysis and comparison and try to find the most optimal neighbourhood/city people often like to visit or for a stakeholder which city/neighbourhood is most likely to open a restaurant or beer bar.

2. DATA

2.1 Data description

For my analysis I have used the following datalinks to download the data:

1. for NY, source: https://geo.nyu.edu/catalog/nyu_2451_34572 (https://geo.nyu.edu/catalog/nyu_2451_34572)
2. for Toronto, source: https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M)

2.2. Download and prepare dataset for Canada

In [3]:

```
import pandas as pd
import json
```

In [4]:

```
#Read the table from wikipedia page and store it in data frame
url = r"https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
```

In [30]:

```
df = pd.read_html(url, header = 0)
df_new = df[0]
```

In [31]:

```
#Get rid of 'Not assigned' Boroughs
df_new = df_new[(df_new.Borough != 'Not assigned')]
df_new
```

Out[31]:

	Postcode	Borough	Neighbourhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront
5	M6A	North York	Lawrence Heights
6	M6A	North York	Lawrence Manor
...	...	...	...
281	M8Z	Etobicoke	Kingsway Park South West
282	M8Z	Etobicoke	Mimico NW
283	M8Z	Etobicoke	The Queensway West
284	M8Z	Etobicoke	Royal York South West
285	M8Z	Etobicoke	South of Bloor

210 rows × 3 columns

In [32]:

```
#Group by Postcode and combine the Neighbourhoods with identical Postcode
df_new['Neighbourhood'] = df_new.groupby(['Postcode', 'Borough'])['Neighbourhood'].transfor
df_new = df_new.drop_duplicates()
df_new.reset_index(drop= True, inplace = True)
```

```
C:\Users\dey65\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel
_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```

Edit the names of Not assigned Neighbourhoods

In [33]:

```
df_new.Neighbourhood[df_new['Neighbourhood']=='Not assigned'] = df_new.Borough[df_new['Neig
```

```
C:\Users\dey65\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\co
re\generic.py:8767: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  self._update_inplace(new_data)
```

In [34]:

```
df_new.rename(index=str, columns={"Postcode": "PostalCode", "Neighbourhood":"Neighborhood"}
```

```
C:\Users\dey65\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\co
re\frame.py:4133: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  errors=errors,
```

In [35]:

```
df_new.shape
```

Out[35]:

```
(103, 3)
```

In [36]:

```
df_new.head()
```

Out[36]:

	PostalCode	Borough	Neighborhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Harbourfront
3	M6A	North York	Lawrence Heights, Lawrence Manor
4	M7A	Downtown Toronto	Queen's Park

2.3. Import data with coordinates for Canada

In [39]:

```
df_geosp = pd.read_csv(r"https://cocl.us/Geospatial_data")
df_geosp.head(10)
```

Out[39]:

	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
5	M1J	43.744734	-79.239476
6	M1K	43.727929	-79.262029
7	M1L	43.711112	-79.284577
8	M1M	43.716316	-79.239476
9	M1N	43.692657	-79.264848

In [45]:

```
df_geosp.rename(index=str, columns={"Postal Code": "PostalCode"}, inplace = True)
df_geosp
```

Out[45]:

	PostalCode	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
100	M9R	43.688905	-79.554724
101	M9V	43.739416	-79.588437
102	M9W	43.706748	-79.594054

103 rows × 3 columns

In [46]:

```
df_toronto = df_new.join(df_geosp.set_index('PostalCode'), on = "PostalCode")
```

In [47]:

```
df_toronto.head(10)
```

Out[47]:

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Heights, Lawrence Manor	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park	43.662301	-79.389494
5	M9A	Etobicoke	Islington Avenue	43.667856	-79.532242
6	M1B	Scarborough	Rouge, Malvern	43.806686	-79.194353
7	M3B	North York	Don Mills North	43.745906	-79.352188
8	M4B	East York	Woodbine Gardens, Parkview Hill	43.706397	-79.309937
9	M5B	Downtown Toronto	Ryerson, Garden District	43.657162	-79.378937

In [48]:

```
print('The dataframe has {} Boroughs'.format(len(df_toronto['Borough'].unique())))
```

```
The dataframe has 10 Boroughs
```

2.4. Download and prepare dataset for New York

In [5]:

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

Let's take a quick look at the data.

In [6]:

```
newyork_data
```

```
     40.89470517661,
     -73.84720052054902,
     40.89470517661]}},
  {'type': 'Feature',
   'id': 'nyu_2451_34572.2',
   'geometry': {'type': 'Point',
    'coordinates': [-73.82993910812398, 40.87429419303012]},
   'geometry_name': 'geom',
   'properties': {'name': 'Co-op City',
    'stacked': 2,
    'annoline1': 'Co-op',
    'annoline2': 'City',
    'annoline3': None,
    'annoangle': 0.0,
    'borough': 'Bronx',
    'bbox': [-73.82993910812398,
     40.87429419303012,
     -73.82993910812398,
     40.87429419303012]}},
  {'type': 'Feature',
```

The relevant data is in the features key, which is basically a list of the neighborhoods. So, let's define a new variable that includes this data.

In [7]:

```
neighborhoods_data = newyork_data['features']
neighborhoods_data[0] # first entry
```

Out[7]:

```
{'type': 'Feature',
 'id': 'nyu_2451_34572.1',
 'geometry': {'type': 'Point',
  'coordinates': [-73.84720052054902, 40.89470517661]},
 'geometry_name': 'geom',
 'properties': {'name': 'Wakefield',
  'stacked': 1,
  'annoline1': 'Wakefield',
  'annoline2': None,
  'annoline3': None,
  'annoangle': 0.0,
  'borough': 'Bronx',
  'bbox': [-73.84720052054902,
   40.89470517661,
   -73.84720052054902,
   40.89470517661]}}
```

Tranform the data into a pandas dataframe

In [8]:

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

In [9]:

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

In [10]:

```
neighborhoods #Empty DataFrame
```

Out[10]:

Borough	Neighborhood	Latitude	Longitude

In [11]:

```
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_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore_index=True
```

In [12]:

```
neighborhoods.head(10)
```

Out[12]:

	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
5	Bronx	Kingsbridge	40.881687	-73.902818
6	Manhattan	Marble Hill	40.876551	-73.910660
7	Bronx	Woodlawn	40.898273	-73.867315
8	Bronx	Norwood	40.877224	-73.879391
9	Bronx	Williamsbridge	40.881039	-73.857446

In [13]:

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

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

3. Explore neighbourhoods in Toronto and New York

In [14]:

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

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

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

#!conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library
```

In [15]:

```
CLIENT_ID = 'FXZZ2JU4H4GDX1BI01043OSKMUEPGB4O5EOI31RD2CPV5E0C' # your Foursquare ID
CLIENT_SECRET = 'AWTDPEZJ5MBPS1LFHKQLULC2OHMKHNAVNTG0253L1GNJXLIK' # your Foursquare Secret
VERSION = '20180604' # Foursquare API version
LIMIT = 100
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: FXZZ2JU4H4GDX1BI01043OSKMUEPGB4O5EOI31RD2CPV5E0C
CLIENT_SECRET:AWTDPEZJ5MBPS1LFHKQLULC2OHMKHNAVNTG0253L1GNJXLIK
```

In [74]:

```
#Create a function to repeat the process of exploring the venues for all the neighborhoods
```

In [24]:

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

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            latitude,
            longitude,
            radius,
            LIMIT)

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

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

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

    return(nearby_venues)
```

In [87]:

```
LIMIT = 30
radius = 500
df_toronto_venues = getNearbyVenues(names=df_toronto['Neighborhood'],latitudes=df_toronto['
```

```
Parkwoods
Victoria Village
Harbourfront
Lawrence Heights, Lawrence Manor
Queen's Park
Islington Avenue
Rouge, Malvern
Don Mills North
Woodbine Gardens, Parkview Hill
Ryerson, Garden District
Glencairn
Cloverdale, Islington, Martin Grove, Princess Gardens, West Deane Park
Highland Creek, Rouge Hill, Port Union
Flemingdon Park, Don Mills South
Woodbine Heights
St. James Town
Humewood-Cedarvale
Bloordale Gardens, Eringate, Markland Wood, Old Burnhamthorpe
Guildwood, Morningside, West Hill
The Beaches
Berczy Park
Caledonia-Fairbanks
Woburn
Leaside
Central Bay Street
Christie
Cedarbrae
Hillcrest Village
Bathurst Manor, Downsview North, Wilson Heights
Thorncliffe Park
Adelaide, King, Richmond
Dovercourt Village, Dufferin
Scarborough Village
Fairview, Henry Farm, Oriole
Northwood Park, York University
East Toronto
Harbourfront East, Toronto Islands, Union Station
Little Portugal, Trinity
East Birchmount Park, Ionview, Kennedy Park
Bayview Village
CFB Toronto, Downsview East
The Danforth West, Riverdale
Design Exchange, Toronto Dominion Centre
Brockton, Exhibition Place, Parkdale Village
Clairlea, Golden Mile, Oakridge
Silver Hills, York Mills
Downsview West
The Beaches West, India Bazaar
Commerce Court, Victoria Hotel
Downsview, North Park, Upwood Park
Humber Summit
Cliffcrest, Cliffside, Scarborough Village West
Newtonbrook, Willowdale
```

Downsview Central
Studio District
Bedford Park, Lawrence Manor East
Del Ray, Keelesdale, Mount Dennis, Silverthorn
Emery, Humberlea
Birch Cliff, Cliffside West
Willowdale South
Downsview Northwest
Lawrence Park
Roselawn
The Junction North, Runnymede
Weston
Dorset Park, Scarborough Town Centre, Wexford Heights
York Mills West
Davisville North
Forest Hill North, Forest Hill West
High Park, The Junction South
Westmount
Maryvale, Wexford
Willowdale West
North Toronto West
The Annex, North Midtown, Yorkville
Parkdale, Roncesvalles
Canada Post Gateway Processing Centre
Kingsview Village, Martin Grove Gardens, Richview Gardens, St. Phillips
Agincourt
Davisville
Harbord, University of Toronto
Runnymede, Swansea
Clarks Corners, Sullivan, Tam O'Shanter
Moore Park, Summerhill East
Chinatown, Grange Park, Kensington Market
Agincourt North, L'Amoreaux East, Milliken, Steeles East
Deer Park, Forest Hill SE, Rathnelly, South Hill, Summerhill West
CN Tower, Bathurst Quay, Island airport, Harbourfront West, King and Spadin
a, Railway Lands, South Niagara
Humber Bay Shores, Mimico South, New Toronto
Albion Gardens, Beaumond Heights, Humbergate, Jamestown, Mount Olive, Silver
stone, South Steeles, Thistletown
L'Amoreaux West
Rosedale
Stn A PO Boxes 25 The Esplanade
Alderwood, Long Branch
Northwest
Upper Rouge
Cabbagetown, St. James Town
First Canadian Place, Underground city
The Kingsway, Montgomery Road, Old Mill North
Church and Wellesley
Business Reply Mail Processing Centre 969 Eastern
Humber Bay, King's Mill Park, Kingsway Park South East, Mimico NE, Old Mill
South, The Queensway East, Royal York South East, Sunnylea
Kingsway Park South West, Mimico NW, The Queensway West, Royal York South We
st, South of Bloor

In [89]:

```
print(df_toronto_venues.shape)
df_toronto_venues.head(10)
```

(1340, 7)

Out[89]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Ver Categ
0	Parkwoods	43.753259	-79.329656	Brookbanks Park	43.751976	-79.332140	F
1	Parkwoods	43.753259	-79.329656	Variety Store	43.751974	-79.333114	Foo Drink Sl
2	Victoria Village	43.725882	-79.315572	Victoria Village Arena	43.723481	-79.315635	Hoc Ar
3	Victoria Village	43.725882	-79.315572	Tim Hortons	43.725517	-79.313103	Co Sl
4	Victoria Village	43.725882	-79.315572	Portugril	43.725819	-79.312785	Portugu Restaur
5	Victoria Village	43.725882	-79.315572	Eglinton Ave E & Sloane Ave/Bermondsey Rd	43.726086	-79.313620	Intersec
6	Victoria Village	43.725882	-79.315572	Pizza Nova	43.725824	-79.312860	Pizza Pl
7	Harbourfront	43.654260	-79.360636	Roselle Desserts	43.653447	-79.362017	Bak
8	Harbourfront	43.654260	-79.360636	Tandem Coffee	43.653559	-79.361809	Co Sl
9	Harbourfront	43.654260	-79.360636	Cooper Koo Family YMCA	43.653249	-79.358008	Distribu Cer

In [90]:

```
#Check how many venues were returned for each neighborhood
```

In [91]:

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

Out[91]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Adelaide, King, Richmond	30	30	30	30	30	30
Agincourt	4	4	4	4	4	4
Agincourt North, L'Amoreaux East, Milliken, Steeles East	3	3	3	3	3	3
Albion Gardens, Beaumond Heights, Humbergate, Jamestown, Mount Olive, Silverstone, South Steeles, Thistletown	11	11	11	11	11	11
Alderwood, Long Branch	10	10	10	10	10	10
...	...	...	...	...	...	...
Willowdale West	6	6	6	6	6	6
Woburn	4	4	4	4	4	4
Woodbine Gardens, Parkview Hill	13	13	13	13	13	13
Woodbine Heights	9	9	9	9	9	9
York Mills West	4	4	4	4	4	4

99 rows × 6 columns

In [92]:

```
#Check unique Venue Categories
```

In [95]:

```
print(len(df_toronto_venues['Venue Category'].unique()))
```

234

In [97]:

```
# one hot encoding
df_toronto_onehot = pd.get_dummies(df_toronto_venues[['Venue Category']], prefix = "", pref
df_toronto_onehot
```

Out[97]:

	Accessories Store	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Art Gallery	Cr S
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	
1335	0	0	0	0	0	0	0	0	0	
1336	0	0	0	0	0	0	0	0	0	
1337	0	0	0	0	0	0	0	0	0	
1338	0	0	0	0	0	0	0	0	0	
1339	0	0	0	0	0	0	0	0	0	

1340 rows × 234 columns

In [105]:

```
df_toronto_onehot["Neighborhood"] = df_toronto_venues["Neighborhood"]#add the dataframe bac
df_toronto_onehot
```

Out[105]:

	Yoga Studio	Accessories Store	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Ga
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	
1335	0	0	0	0	0	0	0	0	0	
1336	0	0	0	0	0	0	0	0	0	
1337	0	0	0	0	0	0	0	0	0	
1338	0	0	0	0	0	0	0	0	0	
1339	0	0	0	0	0	0	0	0	0	

1340 rows × 234 columns

In [106]:

```
# move neighborhood column to the first column
rearranged_columns = [df_toronto_onehot.columns[-1]] + list(df_toronto_onehot.columns[:-1])
df_toronto_onehot = df_toronto_onehot[rearranged_columns]

df_toronto_onehot.head(10)
```

Out[106]:

	Women's Store	Yoga Studio	Accessories Store	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	Americ Restaur
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	

10 rows × 234 columns

In [107]:

```
#Group rows by neighborhood and by taking the mean of the frequency of occurrence of each c
```

In [108]:

```
grouped_toronto_neigh_cat= df_toronto_onehot.groupby('Neighborhood').mean().reset_index()
```

In [109]:

```
grouped_toronto_neigh_cat
```

Out[109]:

cessories Store	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	...	Trail	Train Station	Vegetarian / Vegan Restaurant	Vide Gan Sto
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.033333	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
...	...	...	...	...	...	...	...	...	...	...	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0

In [110]:

```
#Print each neighborhood along with the top 10 most common venues
```

In [111]:

```
num_top_venues = 10

for neigh in grouped_toronto_neigh_cat['Neighborhood']:
    print("----"+ neigh +"----")
    temp = grouped_toronto_neigh_cat[grouped_toronto_neigh_cat['Neighborhood'] == neigh].T.
    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_ven
    print('\n')
```

```
9           Restaurant  0.03


----Agincourt----
                     venue  freq
0             Skating Rink  0.25
1           Breakfast Spot  0.25
2                   Lounge  0.25
3  Latin American Restaurant  0.25
4            Women's Store  0.00
5  Middle Eastern Restaurant  0.00
6             Liquor Store  0.00
7        Mac & Cheese Joint  0.00
8                   Market  0.00
9           Massage Studio  0.00


----Agincourt North, L'Amoreaux East, Milliken, Steeles East----
              venue  freq
0              Park  0.33
```

Create a Pandas dataframe with 10 top_most_venues

In [112]:

```
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 [114]:

```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']

for ind in range(num_top_venues):

    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_df = pd.DataFrame(columns=columns)
neighborhoods_venues_df['Neighborhood'] = grouped_toronto_neigh_cat['Neighborhood']

for ind in np.arange(grouped_toronto_neigh_cat.shape[0]):
    neighborhoods_venues_df.iloc[ind, 1:] = return_most_common_venues(grouped_toronto_neigh

neighborhoods_venues_df.head(10)
```

Out[114]:

	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	
0	Adelaide, King, Richmond	Seafood Restaurant	Hotel	Café	Sushi Restaurant	Asian Restaurant	Coffee Shop	
1	Agincourt	Skating Rink	Latin American Restaurant	Breakfast Spot	Lounge	Wings Joint	College Arts Building	
2	Agincourt North, L'Amoreaux East, Milliken, St...	Park	Playground	Coffee Shop	Gastropub	Gas Station	Donut Shop	
3	Albion Gardens, Beaumond Heights, Humbergate, ...	Grocery Store	Pharmacy	Fast Food Restaurant	Japanese Restaurant	Video Store	Sandwich Place	E
4	Alderwood, Long Branch	Pizza Place	Coffee Shop	Pharmacy	Pub	Skating Rink	Pool	
5	Bathurst Manor, Downsview North, Wilson Heights	Coffee Shop	Middle Eastern Restaurant	Bank	Sushi Restaurant	Deli / Bodega	Sandwich Place	I
6	Bayview Village	Japanese Restaurant	Chinese Restaurant	Bank	Café	Wings Joint	Department Store	F

	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	
7	Bedford Park, Lawrence Manor East	Coffee Shop	Italian Restaurant	Restaurant	Sushi Restaurant	Sandwich Place	Grocery Store	F
8	Berczy Park	Coffee Shop	Beer Bar	Farmers Market	Café	Bakery	Cocktail Bar	F
9	Birch Cliff, Cliffside West	General Entertainment	Skating Rink	College Stadium	Café	Wings Joint	Deli / Bodega	F

4.a. Clustering Neighborhoods of Toronto by K-Means

In [115]:

```
# set number of clusters
kclusters = 7

toronto_grouped_clustering = grouped_toronto_neigh_cat.drop('Neighborhood', axis = 1)

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

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

Out[115]:

```
array([2, 2, 4, 2, 2, 2, 2, 2, 2, 2])
```

In [116]:

```
# add clustering labels
#del neighborhoods_venues_sorted
neighborhoods_venues_df.insert(0, 'Cluster Labels', kmeans.labels_)

toronto_merged = df_toronto

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

toronto_merged.head(10) # check the last columns!
```

Out[116]:

	PostalCode	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Mc Comm Ven
0	M3A	North York	Parkwoods	43.753259	-79.329656	4	Park	Fooc Dr Sh
1	M4A	North York	Victoria Village	43.725882	-79.315572	1	Coffee Shop	Piz Pla
2	M5A	Downtown Toronto	Harbourfront	43.654260	-79.360636	2	Coffee Shop	Pa
3	M6A	North York	Lawrence Heights, Lawrence Manor	43.718518	-79.464763	2	Clothing Store	Furnitur Hor Stc
4	M7A	Downtown Toronto	Queen's Park	43.662301	-79.389494	2	Coffee Shop	Pa
6	M1B	Scarborough	Rouge, Malvern	43.806686	-79.194353	2	Fast Food Restaurant	Wir Jo
7	M3B	North York	Don Mills North	43.745906	-79.352188	2	Café	Japane Restaura
8	M4B	East York	Woodbine Gardens, Parkview Hill	43.706397	-79.309937	2	Pizza Place	Pharma
9	M5B	Downtown Toronto	Ryerson, Garden District	43.657162	-79.378937	2	Coffee Shop	Ca
10	M6B	North York	Glencairn	43.709577	-79.445073	2	Park	Japane Restaura

In [17]:

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

In [118]:

```
address = 'Toronto'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
43.653963 -79.387207
```

In [119]:

```
# create map
toronto_map_clusters = folium.Map(location=[latitude, longitude], zoom_start=15)
```

In [120]:

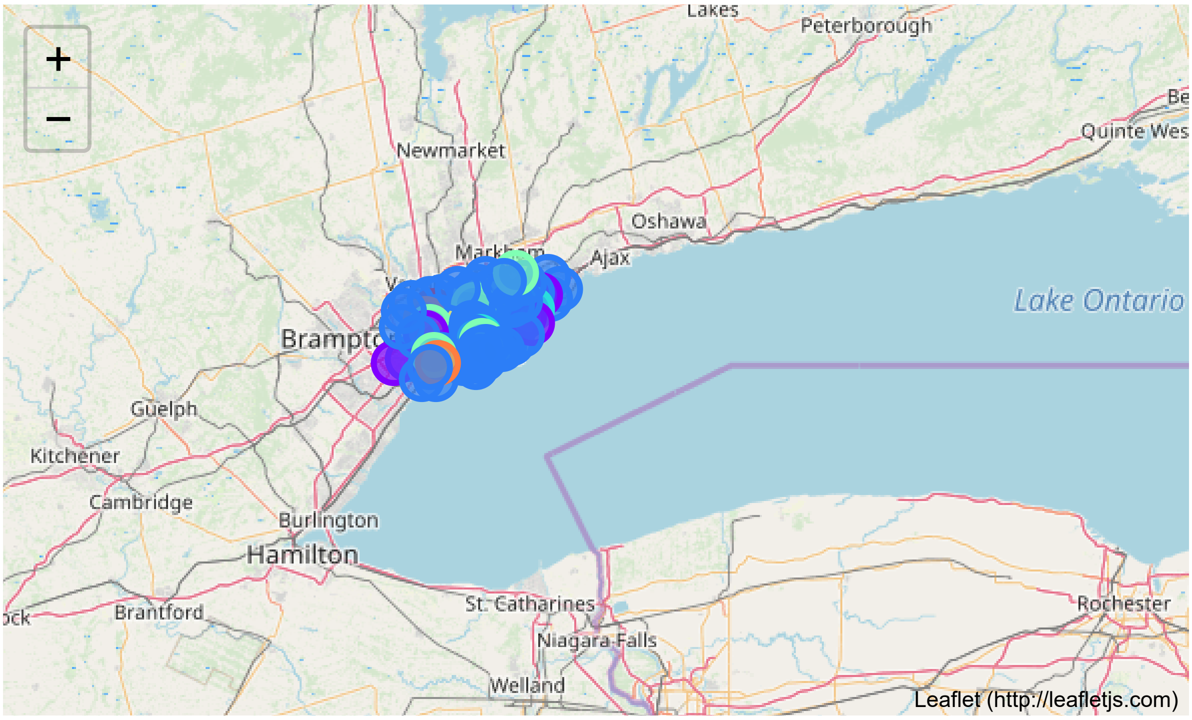
```
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]
```

In [121]:

```
# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(toronto_merged['Latitude'], toronto_merged['Longitude'],
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=10,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(toronto_map_clusters)

toronto_map_clusters
```

Out[121]:



In [18]:

```
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(latitude, longitude)
```

```
The geograpical coordinate of New York City are 40.7127281, -74.0060152.
```

Create a map of New York with neighborhoods superimposed on top.

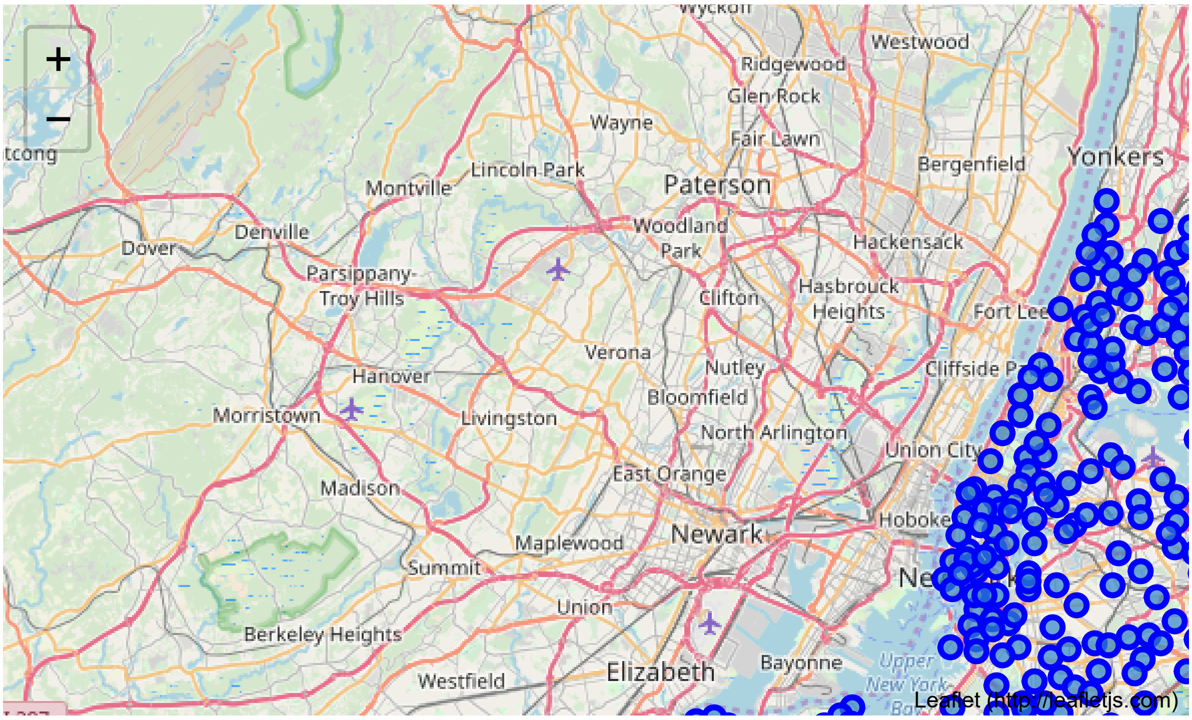
In [19]:

```
# 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'], neighborhoods['Longit
    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[19]:



However, for illustration purposes, let's simplify the above map and segment and cluster only the neighborhoods in Brooklyn. So let's slice the original dataframe and create a new dataframe of the borough Brooklyn data.

In [20]:

```
brooklyn_data = neighborhoods[neighborhoods['Borough'] == 'Brooklyn'].reset_index(drop=True
brooklyn_data.head()
```

Out[20]:

	Borough	Neighborhood	Latitude	Longitude
0	Brooklyn	Bay Ridge	40.625801	-74.030621
1	Brooklyn	Bensonhurst	40.611009	-73.995180
2	Brooklyn	Sunset Park	40.645103	-74.010316
3	Brooklyn	Greenpoint	40.730201	-73.954241
4	Brooklyn	Gravesend	40.595260	-73.973471

Let's get the geographical coordinates of Brooklyn.

In [21]:

```
address = 'Brooklyn, NY'

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

```
The geograpical coordinate of Brooklyn are 40.6501038, -73.9495823.
```

In [22]:

```
# create map of Manhattan using latitude and longitude values
map_brooklyn = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(brooklyn_data['Latitude'], brooklyn_data['Longitude'], brooklyn_
    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_brooklyn)

map_brooklyn
```

Out[22]:

In [25]:

```
LIMIT = 30
radius = 500
df_brooklyn_venues = getNearbyVenues(names=brooklyn_data['Neighborhood'],latitudes=brooklyn
```

```
Bay Ridge
Bensonhurst
Sunset Park
Greenpoint
Gravesend
Brighton Beach
Sheepshead Bay
Manhattan Terrace
Flatbush
Crown Heights
East Flatbush
Kensington
Windsor Terrace
Prospect Heights
Brownsville
Williamsburg
Bushwick
Bedford Stuyvesant
Brooklyn Heights
Cobble Hill
Carroll Gardens
Red Hook
Gowanus
Fort Greene
Park Slope
Cypress Hills
East New York
Starrett City
Canarsie
Flatlands
Mill Island
Manhattan Beach
Coney Island
Bath Beach
Borough Park
Dyker Heights
Gerritsen Beach
Marine Park
Clinton Hill
Sea Gate
Downtown
Boerum Hill
Prospect Lefferts Gardens
Ocean Hill
City Line
Bergen Beach
Midwood
Prospect Park South
Georgetown
East Williamsburg
North Side
South Side
Ocean Parkway
```

```
Fort Hamilton
Ditmas Park
Wingate
Rugby
Remsen Village
New Lots
Paerdegat Basin
Mill Basin
Fulton Ferry
Vinegar Hill
Weeksville
Broadway Junction
Dumbo
Homecrest
Highland Park
Madison
Erasmus
```

In [26]:

```
print(df_brooklyn_venues.shape)
df_brooklyn_venues.head(10)
```

```
(1619, 7)
```

Out[26]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Bay Ridge	40.625801	-74.030621	Pilo Arts Day Spa and Salon	40.624748	-74.030591	Spa
1	Bay Ridge	40.625801	-74.030621	Bagel Boy	40.627896	-74.029335	Bagel Shop
2	Bay Ridge	40.625801	-74.030621	Leo's Casa Calamari	40.624200	-74.030931	Pizza Place
3	Bay Ridge	40.625801	-74.030621	Cocoa Grinder	40.623967	-74.030863	Juice Bar
4	Bay Ridge	40.625801	-74.030621	Pegasus Cafe	40.623168	-74.031186	Breakfast Spot
5	Bay Ridge	40.625801	-74.030621	Ho' Brah Taco Joint	40.622960	-74.031371	Taco Place
6	Bay Ridge	40.625801	-74.030621	A.L.C. Italian Grocery	40.623051	-74.031224	Grocery Store
7	Bay Ridge	40.625801	-74.030621	Karam	40.622931	-74.028316	Middle Eastern Restaurant
8	Bay Ridge	40.625801	-74.030621	Georgian Dream Cafe and Bakery	40.625586	-74.030196	Caucasian Restaurant
9	Bay Ridge	40.625801	-74.030621	Windy City Ale House	40.628117	-74.029128	Sports Bar

Check how many venues were returned for each neighborhood

In [27]:

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

Out[27]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Bath Beach	30	30	30	30	30	30
Bay Ridge	30	30	30	30	30	30
Bedford Stuyvesant	27	27	27	27	27	27
Bensonhurst	30	30	30	30	30	30
Bergen Beach	8	8	8	8	8	8
...	...	...	...	...	...	...
Vinegar Hill	28	28	28	28	28	28
Weeksville	16	16	16	16	16	16
Williamsburg	30	30	30	30	30	30
Windsor Terrace	27	27	27	27	27	27
Wingate	19	19	19	19	19	19

70 rows × 6 columns

In [28]:

```
#Check unique Venue Categories
```

In [29]:

```
print(len(df_brooklyn_venues['Venue Category'].unique()))
```

```
248
```

In [30]:

```python
# one hot encoding
df_brooklyn_onehot = pd.get_dummies(df_brooklyn_venues[['Venue Category']], prefix = "", pr
df_brooklyn_onehot
```

Out[30]:

	Accessories Store	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Arts & Crafts Store	Asian Restaurant	A &
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
1614	0	0	0	0	0	0	0	0	
1615	0	0	0	0	0	0	0	0	
1616	0	0	0	0	0	0	0	0	
1617	0	0	0	0	0	0	0	0	
1618	0	0	0	0	0	0	0	0	

1619 rows × 248 columns

In [31]:

```
df_brooklyn_onehot["Neighborhood"] = df_brooklyn_venues["Neighborhood"]#add the dataframe b
df_brooklyn_onehot
```

Out[31]:

	Accessories Store	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Arts & Crafts Store	Asian Restaurant	A &
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
1614	0	0	0	0	0	0	0	0	
1615	0	0	0	0	0	0	0	0	
1616	0	0	0	0	0	0	0	0	
1617	0	0	0	0	0	0	0	0	
1618	0	0	0	0	0	0	0	0	

1619 rows × 248 columns

In [32]:

```
# move neighborhood column to the first column
rearranged_columns = [df_brooklyn_onehot.columns[-1]] + list(df_brooklyn_onehot.columns[:-1
df_brooklyn_onehot = df_brooklyn_onehot[rearranged_columns]

df_brooklyn_onehot.head(10)
```

Out[32]:

	Yoga Studio	Accessories Store	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Arts & Crafts Store	Asia Restaura
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	

10 rows × 248 columns

In [33]:

```
#Group rows by neighborhood and by taking the mean of the frequency of occurrence of each c
```

In [34]:

```
grouped_brooklyn_neigh_cat= df_brooklyn_onehot.groupby('Neighborhood').mean().reset_index()
```

In [35]:

```
grouped_brooklyn_neigh_cat
```

Out[35]:

	Neighborhood	Yoga Studio	Accessories Store	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Gall
0	Bath Beach	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.000
1	Bay Ridge	0.000000	0.0	0.033333	0.000000	0.0	0.0	0.000
2	Bedford Stuyvesant	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.000
3	Bensonhurst	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.000
4	Bergen Beach	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.000
...	...	...	...	...	...	...	...	
65	Vinegar Hill	0.000000	0.0	0.035714	0.035714	0.0	0.0	0.071
66	Weeksville	0.000000	0.0	0.062500	0.000000	0.0	0.0	0.000
67	Williamsburg	0.033333	0.0	0.000000	0.000000	0.0	0.0	0.000
68	Windsor Terrace	0.000000	0.0	0.037037	0.037037	0.0	0.0	0.000
69	Wingate	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.000

70 rows × 248 columns

In [36]:

```
#Print each neighborhood along with the top 10 most common venues
```

In [37]:

```
num_top_venues = 10

for neigh in grouped_brooklyn_neigh_cat['Neighborhood']:
    print("----"+ neigh +"----")
    temp = grouped_brooklyn_neigh_cat[grouped_brooklyn_neigh_cat['Neighborhood'] == neigh].
    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_ven
    print('\n')
```

```
                venue  freq
0     Sushi Restaurant  0.07
1   Russian Restaurant  0.07
2           Restaurant  0.07
3           Non-Profit  0.03
4           Taco Place  0.03
5        Grocery Store  0.03
6     Food & Drink Shop  0.03
7            Bookstore  0.03
8       Supplement Shop  0.03
9          Supermarket  0.03


----Broadway Junction----
                venue  freq
0           Donut Shop  0.11
1                Diner  0.11
2          Gas Station  0.11
3        Metro Station  0.05
```

Create a Pandas dataframe with 10 top_most_venues

In [38]:

```
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 [39]:

```python
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']

for ind in range(num_top_venues):

    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_df2 = pd.DataFrame(columns=columns)
neighborhoods_venues_df2['Neighborhood'] = grouped_brooklyn_neigh_cat['Neighborhood']

for ind in np.arange(grouped_brooklyn_neigh_cat.shape[0]):
    neighborhoods_venues_df2.iloc[ind, 1:] = return_most_common_venues(grouped_brooklyn_nei

neighborhoods_venues_df2.head(10)
```

Out[39]:

	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 Com V
0	Bath Beach	Bubble Tea Shop	Pharmacy	Fast Food Restaurant	Chinese Restaurant	Italian Restaurant	Bank	Canto Resta
1	Bay Ridge	Spa	Pizza Place	Grocery Store	Greek Restaurant	Caucasian Restaurant	Chinese Restaurant	Lo
2	Bedford Stuyvesant	Deli / Bodega	Pizza Place	Coffee Shop	Café	Bar	Japanese Restaurant	
3	Bensonhurst	Italian Restaurant	Chinese Restaurant	Donut Shop	Sushi Restaurant	Ice Cream Shop	Grocery Store	E
4	Bergen Beach	Harbor / Marina	Baseball Field	Playground	Hockey Field	Park	Donut Shop	Athlet S
5	Boerum Hill	Bar	Yoga Studio	Furniture / Home Store	Coffee Shop	Spa	Bakery	Japa Resta
6	Borough Park	Bank	Pizza Place	Pharmacy	Fast Food Restaurant	Deli / Bodega	Restaurant	C
7	Brighton Beach	Restaurant	Sushi Restaurant	Russian Restaurant	Other Great Outdoors	Food & Drink Shop	Mediterranean Restaurant	Lo
8	Broadway Junction	Diner	Donut Shop	Gas Station	Bus Stop	Burger Joint	Fried Chicken Joint	Nigh
9	Brooklyn Heights	Yoga Studio	Pet Store	Diner	Scenic Lookout	Coffee Shop	Cosmetics Shop	Hi Mus

4.b. Clustering Neighborhoods of Brooklyn by K-Means

In [40]:

```
# set number of clusters
kclusters = 7

brooklyn_grouped_clustering = grouped_brooklyn_neigh_cat.drop('Neighborhood', axis = 1)

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

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

Out[40]:

```
array([6, 0, 0, 6, 4, 0, 6, 0, 3, 0])
```

In [41]:

```
# add clustering labels
#del neighborhoods_venues_sorted
neighborhoods_venues_df2.insert(0, 'Cluster Labels', kmeans.labels_)

brooklyn_merged = brooklyn_data

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
brooklyn_merged = brooklyn_merged.join(neighborhoods_venues_df2.set_index('Neighborhood'),

brooklyn_merged.head(10) # check the last columns!
```

Out[41]:

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	
0	Brooklyn	Bay Ridge	40.625801	-74.030621	0	Spa	Pizza Place	Grocery Store	R
1	Brooklyn	Bensonhurst	40.611009	-73.995180	6	Italian Restaurant	Chinese Restaurant	Donut Shop	R
2	Brooklyn	Sunset Park	40.645103	-74.010316	6	Mexican Restaurant	Bank	Latin American Restaurant	
3	Brooklyn	Greenpoint	40.730201	-73.954241	0	Bar	Cocktail Bar	Café	R
4	Brooklyn	Gravesend	40.595260	-73.973471	6	Pizza Place	Bakery	Lounge	R
5	Brooklyn	Brighton Beach	40.576825	-73.965094	0	Restaurant	Sushi Restaurant	Russian Restaurant	
6	Brooklyn	Sheepshead Bay	40.586890	-73.943186	0	Dessert Shop	Turkish Restaurant	Sandwich Place	
7	Brooklyn	Manhattan Terrace	40.614433	-73.957438	6	Pizza Place	Donut Shop	Ice Cream Shop	
8	Brooklyn	Flatbush	40.636326	-73.958401	3	Chinese Restaurant	Coffee Shop	Caribbean Restaurant	R
9	Brooklyn	Crown Heights	40.670829	-73.943291	6	Pizza Place	Bagel Shop	Café	

In [42]:

```
from geopy.geocoders import Nominatim # convert an address into latitude and longitude valu
```

In [43]:

```
address = 'Brooklyn'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
40.6501038 -73.9495823
```

In [51]:

```
# create map
brooklyn_map_clusters = folium.Map(location=[latitude, longitude], zoom_start=15)
```

In [52]:

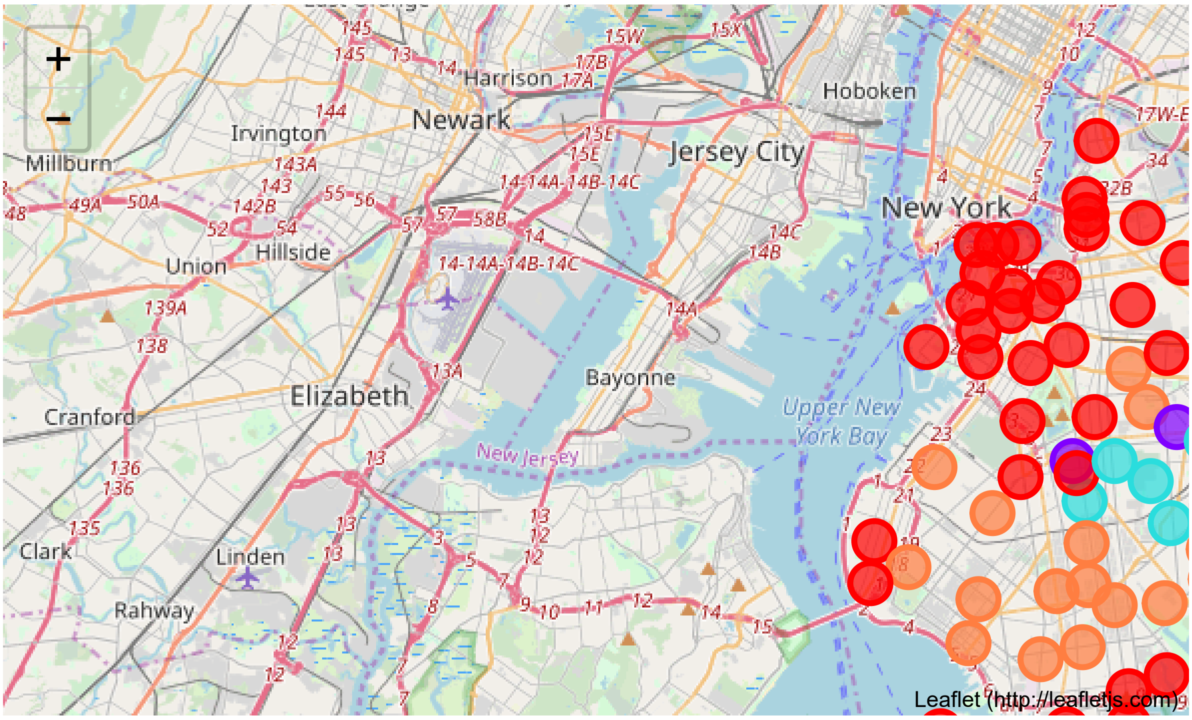
```
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]
```

In [53]:

```
# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(brooklyn_merged['Latitude'], brooklyn_merged['Longitude']
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=10,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(brooklyn_map_clusters)

brooklyn_map_clusters
```

Out[53]:



5. RESULTS AND DISCUSSIONS:

During this analysis 7 clusters were defined. We have analyzed Toronto and Brooklyn, NY for best 10 most common Venues. Both cities are very diverse and are the financial capitals of their respective countries. The most common places for both the cities are restaurants and/or coffee shops. Therefore, for any Stakeholder due to the high number of competitors, the placement of any new restaurant in that area is too risky venture.

5. CONCLUSION:

To conclude, the basic data analysis was performed to compare the neighborhoods of the two cities Toronto and Brooklyn, a borough of NY, USA and determine how similar or dissimilar they are. During the analysis, several important statistical features of the boroughs were explored and visualized. Furthermore, clustering helped to highlight the group of optimal areas.Though, both cities are very diverse and are the financial capitals of their respective countries but most common place that people often visit is restaurants and coffee shops. We can see the availability of diverse cuisines in both cities.

In [ ]:

```

```