

Capstone Project - The Battle of Neighborhoods

Introduction

New York is one of the biggest cities in the world and has a huge number of companies based there. Every year a lot of Brazilians, like me, need to go there to work for some months or to live there. I will make a analysis to find out what is the best neighborhood for a Brazilian stay during some months in New York. I will start by looking for neighborhoods with Brazilian restaurants because I know that would be very hard for me not eating Brazilian food for more than one week. So I prefer to stay around some Brazilian foods. Also, going to Brazilian restaurants I can meet others Brazilians and chat a little bit in Portuguese. Than, I will look for others places that usually Brazilians love like supermarket, gym, bar and coffee. I will also list the best hotels to stay in those neighborhoods.

Data

For this project we need the following data :

- New York City data that contains list Boroughs, Neighborhoods along with their latitude and longitude. Data source : https://cocl.us/new_york_dataset (https://cocl.us/new_york_dataset) Description : This data set contains the required information. And we will use this data set to explore various neighborhoods of new york city.
- Brazilian restaurants in each neighborhood of new york city. Data source : Fousquare API Description : By using this api we will get all the venues in each neighborhood. We can filter these venues to get only brazilian restaurants.
- Hotel list with ratings Data source : Fousquare API Description : By using this api we will get all hotels around neighborhood with rate.

Methodology

The Methodology section will describe the main components of our analysis. All steps made during the analysis are described below.

1. Let's download all the dependencies that we will need for this analysis.

```
In [3]: import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analysis
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 haven't com
pleted the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longit
ude values

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

# 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 if you haven
't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

Solving environment: done

```
==> WARNING: A newer version of conda exists. <==
current version: 4.5.11
latest version: 4.7.12
```

Please update conda by running

```
$ conda update -n base -c defaults conda
```

Package Plan

environment location: /home/jupyterlab/conda/envs/python

added / updated specs:
- geopy

The following packages will be downloaded:

package	build		
geopy-1.20.0	py_0	57 KB	conda-forge
certifi-2019.9.11	py36_0	147 KB	conda-forge
geographiclib-1.49	py_0	32 KB	conda-forge
Total:		237 KB	

The following NEW packages will be INSTALLED:

```
geographiclib: 1.49-py_0      conda-forge
geopy:         1.20.0-py_0    conda-forge
```

The following packages will be UPDATED:

```
certifi:       2019.6.16-py36_1 conda-forge --> 2019.9.11-py36_0 conda-forge
```

Downloading and Extracting Packages

```
geopy-1.20.0      | 57 KB | ##### | 100%
certifi-2019.9.11 | 147 KB | ##### | 100%
geographiclib-1.49 | 32 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done
```

```
==> WARNING: A newer version of conda exists. <==
current version: 4.5.11
latest version: 4.7.12
```

Please update conda by running

```
$ conda update -n base -c defaults conda
```

All requested packages already installed.

Libraries imported.

1. Downloading and Exploring Dataset

We will use the New York dataset from this link: https://geo.nyu.edu/catalog/nyu_2451_34572 (https://geo.nyu.edu/catalog/nyu_2451_34572) to check which Boroughs and Neighborhoods have Brazilian restaurants.

We will use the same file that was downloaded previously for the lab.

```
In [4]: !wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
        print('Data downloaded!')
```

Data downloaded!

Loading the data now.

```
In [5]: with open('newyork_data.json') as json_data:
        newyork_data = json.load(json_data)
```

As all the relevant data is in the *features* key, which is basically a list of the neighborhoods we will define a new variable that includes this data.

```
In [6]: neighborhoods_data = newyork_data['features']
```

Let's transform this data of nested Python dictionaries into a *pandas* dataframe.

```
In [7]: # define the dataframe columns
        column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

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

```
In [8]: 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_in
dex=True)
```

Let's check the the first lines of dataframe created.

```
In [11]: neighborhoods.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

Checking the dataset size.

```
In [10]: print('The dataframe has {} boroughs and {} neighborhoods.'.format(  
            len(neighborhoods['Borough'].unique()),  
            neighborhoods.shape[0]  
        )  
    )
```

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

Now we will create a function to get top 100 venues from Foursquare api within a radius of 1000 metres for a given latitude and longitude. Below function will return us the venue id , venue name and category.

```

In [9]: def get_venues(lat,lng):

    #set variables
    radius=1000
    LIMIT=100
    CLIENT_ID = 'VD2VBP2YGIDOVBIH122JBCUN1YLN44EPVCGDUAWBXSB051'
    CLIENT_SECRET = 'X353ACX0FWHWV3ZWKKGDWO4KXKU4YM2DX5CJDU4EM12MBZBU' # your Fours
quare Secret
    VERSION = '20180605' # Foursquare API version

    #url to fetch data from foursquare api
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secre
t={}&v={}&ll={},{}&radius={}&limit={}'.format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        lng,
        radius,
        LIMIT)

    # get all the data
    results = requests.get(url).json()
    venue_data=results["response"]["groups"][0]["items"]
    venue_details=[]
    for row in venue_data:
        try:
            venue_id=row['venue']['id']
            venue_name=row['venue']['name']
            venue_category=row['venue']['categories'][0]['name']
            venue_details.append([venue_id,venue_name,venue_category])
        except KeyError:
            pass

    column_names=['ID','Name','Category']
    df = pd.DataFrame(venue_details,columns=column_names)
    return df

```

Let's create the neighborhoods list that contains Brazilian restaurants

```
In [15]: column_names=['Borough', 'Neighborhood', 'ID', 'Name']
br_rest_ny=pd.DataFrame(columns=column_names)
count=1
for row in neighborhoods.values.tolist():
    Borough, Neighborhood, Latitude, Longitude=row
    venues = get_venues(Latitude,Longitude)
    br_restaurants=venues[venues['Category']=='Brazilian Restaurant']
    print('(',count,'/',len(neighborhoods),'),','Brazilian Restaurants in '+Neighborhood+', '+Borough+':'+str(len(br_restaurants)))
    for restaurant_detail in br_restaurants.values.tolist():
        id, name , category=restaurant_detail
        br_rest_ny = br_rest_ny.append({'Borough': Borough,
                                         'Neighborhood': Neighborhood,
                                         'ID': id,
                                         'Name' : name
                                         }, ignore_index=True)

    count+=1
```

(1 / 306) Brazilian Resturants in Wakefield, Bronx:0
(2 / 306) Brazilian Resturants in Co-op City, Bronx:0
(3 / 306) Brazilian Resturants in Eastchester, Bronx:0
(4 / 306) Brazilian Resturants in Fieldston, Bronx:0
(5 / 306) Brazilian Resturants in Riverdale, Bronx:0
(6 / 306) Brazilian Resturants in Kingsbridge, Bronx:0
(7 / 306) Brazilian Resturants in Marble Hill, Manhattan:0
(8 / 306) Brazilian Resturants in Woodlawn, Bronx:0
(9 / 306) Brazilian Resturants in Norwood, Bronx:0
(10 / 306) Brazilian Resturants in Williamsbridge, Bronx:0
(11 / 306) Brazilian Resturants in Baychester, Bronx:0
(12 / 306) Brazilian Resturants in Pelham Parkway, Bronx:0
(13 / 306) Brazilian Resturants in City Island, Bronx:0
(14 / 306) Brazilian Resturants in Bedford Park, Bronx:0
(15 / 306) Brazilian Resturants in University Heights, Bronx:0
(16 / 306) Brazilian Resturants in Morris Heights, Bronx:0
(17 / 306) Brazilian Resturants in Fordham, Bronx:0
(18 / 306) Brazilian Resturants in East Tremont, Bronx:0
(19 / 306) Brazilian Resturants in West Farms, Bronx:0
(20 / 306) Brazilian Resturants in High Bridge, Bronx:0
(21 / 306) Brazilian Resturants in Melrose, Bronx:0
(22 / 306) Brazilian Resturants in Mott Haven, Bronx:0
(23 / 306) Brazilian Resturants in Port Morris, Bronx:0
(24 / 306) Brazilian Resturants in Longwood, Bronx:0
(25 / 306) Brazilian Resturants in Hunts Point, Bronx:0
(26 / 306) Brazilian Resturants in Morrisania, Bronx:0
(27 / 306) Brazilian Resturants in Soundview, Bronx:0
(28 / 306) Brazilian Resturants in Clason Point, Bronx:0
(29 / 306) Brazilian Resturants in Throgs Neck, Bronx:0
(30 / 306) Brazilian Resturants in Country Club, Bronx:0
(31 / 306) Brazilian Resturants in Parkchester, Bronx:0
(32 / 306) Brazilian Resturants in Westchester Square, Bronx:0
(33 / 306) Brazilian Resturants in Van Nest, Bronx:0
(34 / 306) Brazilian Resturants in Morris Park, Bronx:0
(35 / 306) Brazilian Resturants in Belmont, Bronx:0
(36 / 306) Brazilian Resturants in Spuyten Duyvil, Bronx:0
(37 / 306) Brazilian Resturants in North Riverdale, Bronx:0
(38 / 306) Brazilian Resturants in Pelham Bay, Bronx:0
(39 / 306) Brazilian Resturants in Schuylerville, Bronx:0
(40 / 306) Brazilian Resturants in Edgewater Park, Bronx:0
(41 / 306) Brazilian Resturants in Castle Hill, Bronx:0
(42 / 306) Brazilian Resturants in Olinville, Bronx:0
(43 / 306) Brazilian Resturants in Pelham Gardens, Bronx:0
(44 / 306) Brazilian Resturants in Concourse, Bronx:0
(45 / 306) Brazilian Resturants in Unionport, Bronx:0
(46 / 306) Brazilian Resturants in Edenwald, Bronx:0
(47 / 306) Brazilian Resturants in Bay Ridge, Brooklyn:0
(48 / 306) Brazilian Resturants in Bensonhurst, Brooklyn:0
(49 / 306) Brazilian Resturants in Sunset Park, Brooklyn:0
(50 / 306) Brazilian Resturants in Greenpoint, Brooklyn:0
(51 / 306) Brazilian Resturants in Gravesend, Brooklyn:0
(52 / 306) Brazilian Resturants in Brighton Beach, Brooklyn:0
(53 / 306) Brazilian Resturants in Sheepshead Bay, Brooklyn:0
(54 / 306) Brazilian Resturants in Manhattan Terrace, Brooklyn:0
(55 / 306) Brazilian Resturants in Flatbush, Brooklyn:0
(56 / 306) Brazilian Resturants in Crown Heights, Brooklyn:0
(57 / 306) Brazilian Resturants in East Flatbush, Brooklyn:0
(58 / 306) Brazilian Resturants in Kensington, Brooklyn:0
(59 / 306) Brazilian Resturants in Windsor Terrace, Brooklyn:0
(60 / 306) Brazilian Resturants in Prospect Heights, Brooklyn:0
(61 / 306) Brazilian Resturants in Brownsville, Brooklyn:0
(62 / 306) Brazilian Resturants in Williamsburg, Brooklyn:0
(63 / 306) Brazilian Resturants in Bushwick, Brooklyn:0
(64 / 306) Brazilian Resturants in Bedford Stuyvesant, Brooklyn:0


```
In [16]: #showing dataset result
br_rest_ny
```

Out[16]:

	Borough	Neighborhood	ID	Name
0	Manhattan	Clinton	57671bd4498e7856b7d79963	Samba Kitchen & Bar
1	Manhattan	West Village	4a5b5143f964a520fdb1fe3	Berimbau do Brasil
2	Queens	Astoria	4bdf502a89ca76b062b75d5e	Favela Grill
3	Queens	Long Island City	5338a897498e1b9bc410d5d1	Beija Flor
4	Queens	East Elmhurst	4f0b6fb3e4b07c79f8f42d61	Rainhas Churrascaria
5	Queens	Steinway	58d6aa0898f8aa0d67c21411	Kilo
6	Queens	Rockaway Beach	5554b037498e2fc87369fe8a	The Summer Shift by The MP Shift
7	Queens	Ravenswood	4b6dbc02f964a520c98a2ce3	New York Pão de Queijo
8	Queens	Ravenswood	4c4f33b824edc9b633ab4ebb	Copacabana Brazilian Restaurant
9	Queens	Ravenswood	5338a897498e1b9bc410d5d1	Beija Flor
10	Queens	North Corona	4f0b6fb3e4b07c79f8f42d61	Rainhas Churrascaria
11	Manhattan	Stuyvesant Town	458d06def964a520ff3f1fe3	Esperanto

We see that Queens has the most of Brazilian restaurants and we Manhattan has only 3.

So now I will explore Queens Borough and find out a good place to leave for some months.

```
In [12]: #creating a new dataframe filtering only Queens Borough.
queens_data = neighborhoods[neighborhoods['Borough'] == 'Queens'].reset_index(drop=
True)
queens_data.head()
```

Out[12]:

	Borough	Neighborhood	Latitude	Longitude
0	Queens	Astoria	40.768509	-73.915654
1	Queens	Woodside	40.746349	-73.901842
2	Queens	Jackson Heights	40.751981	-73.882821
3	Queens	Elmhurst	40.744049	-73.881656
4	Queens	Howard Beach	40.654225	-73.838138

Now we will use same function used on New York lab to get top 100 venues to all the neighborhoods in Queens in a radius of 500 meters.

```

In [13]: CLIENT_ID = 'VD2VBP2YGIDOVBIH122JBCUN1YLN144EPVCGDUAWBXSBO51'
CLIENT_SECRET = 'X353ACX0FWHWV3ZWKKGDWO4KXKU4YM2DX5CJDU4EM12MBZBU' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
RADIUS=500
LIMIT=100

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

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

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

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

        # return only relevant information for each nearby venue
        venues_list.append([
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['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 item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                            'Neighborhood Latitude',
                            'Neighborhood Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)

```

Here we will run the above function on each neighborhood and create a new dataframe called *queens_venues*.

```
In [14]: queens_venues = getNearbyVenues(names=queens_data['Neighborhood'],  
                                          latitudes=queens_data['Latitude'],  
                                          longitudes=queens_data['Longitude']  
                                          )
```

Astoria
Woodside
Jackson Heights
Elmhurst
Howard Beach
Corona
Forest Hills
Kew Gardens
Richmond Hill
Flushing
Long Island City
Sunnyside
East Elmhurst
Maspeth
Ridgewood
Glendale
Rego Park
Woodhaven
Ozone Park
South Ozone Park
College Point
Whitestone
Bayside
Auburndale
Little Neck
Douglaston
Glen Oaks
Bellerose
Kew Gardens Hills
Fresh Meadows
Briarwood
Jamaica Center
Oakland Gardens
Queens Village
Hollis
South Jamaica
St. Albans
Rochdale
Springfield Gardens
Cambria Heights
Rosedale
Far Rockaway
Broad Channel
Breezy Point
Steinway
Beechhurst
Bay Terrace
Edgemere
Arverne
Rockaway Beach
Neponsit
Murray Hill
Floral Park
Holliswood
Jamaica Estates
Queensboro Hill
Hillcrest
Ravenswood
Lindenwood
Laurelton
Lefrak City
Belle Harbor
Rockaway Park
Somerville

Checking queens_venues dataframe size and first lines:

```
In [15]: print(queens_venues.shape)
queens_venues.head()

(2155, 7)
```

Out[15]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Astoria	40.768509	-73.915654	Favela Grill	40.767348	-73.917897	Brazilian Restaurant
1	Astoria	40.768509	-73.915654	CrossFit Queens	40.769404	-73.918977	Gym
2	Astoria	40.768509	-73.915654	Titan Foods Inc.	40.769198	-73.919253	Gourmet Shop
3	Astoria	40.768509	-73.915654	Orange Blossom	40.769856	-73.917012	Gourmet Shop
4	Astoria	40.768509	-73.915654	Simply Fit Astoria	40.769114	-73.912403	Gym

1. Analyzing each Neighborhood in Queens Borough.

```
In [26]: # one hot encoding
queens_onehot = pd.get_dummies(queens_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
queens_onehot['Neighborhood'] = queens_venues['Neighborhood']

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

queens_onehot.head()
```

Out[26]:

	Yoga Studio	Accessories Store	Afghan Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Arts Entertainment
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	

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

```
In [27]: queens_grouped = queens_onehot.groupby('Neighborhood').mean().reset_index()  
         queens_grouped
```

Out [27]:

	Neighborhood	Yoga Studio	Accessories Store	Afghan Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum
0	Arverne	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	Astoria	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	Astoria Heights	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	Auburndale	0.000000	0.000000	0.000000	0.052632	0.000000	0.000000	0.000000	0.000000
4	Bay Terrace	0.000000	0.024390	0.000000	0.048780	0.000000	0.000000	0.000000	0.000000
5	Bayside	0.013514	0.000000	0.000000	0.040541	0.000000	0.000000	0.000000	0.000000
6	Bayswater	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	Beechhurst	0.062500	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	Bellaire	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	Belle Harbor	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	Bellerose	0.000000	0.000000	0.000000	0.055556	0.000000	0.000000	0.000000	0.000000
11	Blissville	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
12	Breezy Point	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
13	Briarwood	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
14	Broad Channel	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
15	Brookville	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
16	Cambria Heights	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
17	College Point	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
18	Corona	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
19	Douglaston	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
20	East Elmhurst	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
21	Edgemere	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
22	Elmhurst	0.000000	0.000000	0.000000	0.000000	0.000000	0.025000	0.000000	0.000000
23	Far Rockaway	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
24	Floral Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25	Flushing	0.016949	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
26	Forest Hills	0.050000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
27	Forest Hills Gardens	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
28	Fresh Meadows	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
29	Glen Oaks	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
30	Glendale	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
31	Hammels	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
32	Hillcrest	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
33	Hollis	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
34	Holliswood	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
35	Howard Beach	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
36	Hunters Point	0.013514	0.000000	0.000000	0.027027	0.000000	0.000000	0.000000	0.000000

So in dataframe above, we can see what venues, with frequency, we have for each neighborhood in a radius of 500 meters. We can check for example how many restaurants, churches, gyms and so on we have on Astoria or any other neighborhood in Queens.

Now let's create a function to sort the venues in descending order.

```
In [28]: 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]
```

And create a new dataframe with top 10 venues for each neighborhood.

```
In [60]: 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[ind]))
    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'] = queens_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

Out[60]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	Arverne	Surf Spot	Sandwich Place	Metro Station	Beach	Bed & Breakfast	Thai Restaurant	Donut Shop	Coffee Shop
1	Astoria	Bar	Middle Eastern Restaurant	Greek Restaurant	Hookah Bar	Seafood Restaurant	Mediterranean Restaurant	Bakery	Pizzeria
2	Astoria Heights	Playground	Italian Restaurant	Plaza	Bus Station	Bowling Alley	Supermarket	Bakery	Burgers
3	Auburndale	Hookah Bar	Gymnastics Gym	Korean Restaurant	Supermarket	Noodle House	Furniture / Home Store	Italian Restaurant	Discoteca
4	Bay Terrace	Clothing Store	Women's Store	Shoe Store	Cosmetics Shop	American Restaurant	Donut Shop	Mobile Phone Shop	Kiosk

1. Clustering neighborhoods

Now we will use k-means clustering model to cluster the neighborhoods into 3 clusters. I tried to set the number of clusters to 4 and 5, but it created 2 groups with one neighborhood on each, so I decided to keep only 3 clusters. We will check how the neighborhoods in each cluster are similar to each other in terms of the features included in the dataset.

```
In [61]: # set number of clusters
kclusters = 3

queens_grouped_clustering = queens_grouped.drop('Neighborhood', 1)

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

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

```
Out[61]: array([2, 2, 2, 2, 2, 2, 0, 2, 2, 1], dtype=int32)
```

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

```
In [62]: # add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

queens_merged = queens_data

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

queens_merged.head()
```

```
Out[62]:
```

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Queens	Astoria	40.768509	-73.915654	2	Bar	Middle Eastern Restaurant	Greek Restaurant	Hookah Bar	5th Most Common Venue
1	Queens	Woodside	40.746349	-73.901842	2	Grocery Store	Thai Restaurant	Latin American Restaurant	Filipino Restaurant	
2	Queens	Jackson Heights	40.751981	-73.882821	2	Latin American Restaurant	Peruvian Restaurant	Bakery	Mobile Phone Shop	Ar
3	Queens	Elmhurst	40.744049	-73.881656	2	Thai Restaurant	Mexican Restaurant	South American Restaurant	Chinese Restaurant	Vietn
4	Queens	Howard Beach	40.654225	-73.838138	2	Italian Restaurant	Fast Food Restaurant	Pharmacy	Bagel Shop	Cons Land

Let's get the geographical coordinates of Queens to create a map showing the 3 clusters that were created.

```
In [63]: address = 'Queens, NY'

geolocator = Nominatim(user_agent="ny_explorer")

location = geolocator.geocode(address)

latitude = location.latitude

longitude = location.longitude

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

The geograpical coordinate of Queens are 40.6524927, -73.7914214158161.

```
In [64]: # 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(queens_merged['Latitude'], queens_merged['Longitude'], queens_merged['Neighborhood'], queens_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    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 [64]:

You can see in the map that most of neighborhoods is on cluster 3 in green . Let's see each cluster below on dataframe.

```
In [65]: #Cluster 1 - In this cluster we have neighborhoods with first common venue Park and
others similar venues like Women's Store, Farmers Market, Eletronics Stores and oth
ers.

queens_merged.loc[queens_merged['Cluster Labels'] == 0, queens_merged.columns[[1] +
list(range(5, queens_merged.shape[1]))]]
```

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 Most Common Venue	8th Most Common Venue	
63	Somerville	Park	Women's Store	Farmers Market	Electronics Store	Empanada Restaurant	Event Space	Falafel Restaurant	Farm	I R
79	Bayswater	Park	Playground	Women's Store	Farmers Market	Electronics Store	Empanada Restaurant	Event Space	Falafel Restaurant	

```
In [66]: #Cluster 2 - This cluster we have most of neighborhoods with first common venue Bea
ch and others similar venues like Event Space, Bus Stop and Fast Food Restaurant.

queens_merged.loc[queens_merged['Cluster Labels'] == 1, queens_merged.columns[[1] +
list(range(5, queens_merged.shape[1]))]]
```

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	8th Mos Common Venue	
43	Breezy Point	Beach	Monument / Landmark	Board Shop	Trail	Women's Store	Fast Food Restaurant	Event Space	Falafe Restauran	
50	Neponsit	Beach	Bus Stop	Filipino Restaurant	Event Space	Falafel Restaurant	Farm	Farmers Market	Fast Foo Restauran	
61	Belle Harbor	Beach	Spa	Deli / Bodega	Chinese Restaurant	Ice Cream Shop	Pharmacy	Bus Stop	Italian Restauran	
62	Rockaway Park	Beach	Pizza Place	Deli / Bodega	Bank	Donut Shop	Bagel Shop	Mediterranean Restaurant	Bus Sto	
75	Roxbury	Beach	Fast Food Restaurant	Irish Pub	Trail	Hardware Store	Baseball Field	Deli / Bodega	Pul	
77	Malba	Tennis Court	Rest Area	Egyptian Restaurant	Electronics Store	Empanada Restaurant	Event Space	Falafel Restaurant	Farm	
78	Hammels	Beach	Bus Station	Building	Fast Food Restaurant	Gym / Fitness Center	Café	Shoe Store	Dog Rui	

```
In [67]: #Cluster 3 - - In this cluster we have neighborhoods with high number of restaurant  
s, bars, stores, gyms and supermarkets.  
  
queens_merged.loc[queens_merged['Cluster Labels'] == 2, queens_merged.columns[[1] +  
list(range(5, queens_merged.shape[1]))]]
```

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	Astoria	Bar	Middle Eastern Restaurant	Greek Restaurant	Hookah Bar	Seafood Restaurant	Mediterranean Restaurant	Bakery
1	Woodside	Grocery Store	Thai Restaurant	Latin American Restaurant	Filipino Restaurant	Bakery	American Restaurant	Pizza Place
2	Jackson Heights	Latin American Restaurant	Peruvian Restaurant	Bakery	Mobile Phone Shop	South American Restaurant	Thai Restaurant	Diner
3	Elmhurst	Thai Restaurant	Mexican Restaurant	South American Restaurant	Chinese Restaurant	Vietnamese Restaurant	Bubble Tea Shop	Hotpot Restaurant
4	Howard Beach	Italian Restaurant	Fast Food Restaurant	Pharmacy	Bagel Shop	Construction & Landscaping	Sandwich Place	Deli / Bodega
5	Corona	Mexican Restaurant	Convenience Store	Juice Bar	Deli / Bodega	Park	Donut Shop	Restaurant
6	Forest Hills	Gym	Gym / Fitness Center	Park	Food Truck	Thai Restaurant	Asian Restaurant	Yoga Studio
7	Kew Gardens	Chinese Restaurant	Deli / Bodega	Bar	Bank	Cosmetics Shop	Pizza Place	Donut Shop
8	Richmond Hill	Pizza Place	Bank	Latin American Restaurant	Lounge	Caribbean Restaurant	Spanish Restaurant	Supermarket
9	Flushing	Hotpot Restaurant	Chinese Restaurant	Korean Restaurant	Bubble Tea Shop	Karaoke Bar	Construction & Landscaping	Bakery
10	Long Island City	Coffee Shop	Hotel	Bar	Gym / Fitness Center	Café	Pizza Place	Mexican Restaurant
11	Sunnyside	Pizza Place	South American Restaurant	Italian Restaurant	Chinese Restaurant	Hotel	Discount Store	Coffee Shop
12	East Elmhurst	Donut Shop	Lake	Rental Car Location	Chinese Restaurant	Gas Station	Coffee Shop	Supermarket
13	Maspeth	Pizza Place	Diner	Mobile Phone Shop	Grocery Store	Bank	Pharmacy	Chinese Restaurant
14	Ridgewood	Bakery	Deli / Bodega	Bank	Pizza Place	Mobile Phone Shop	Greek Restaurant	Grocery Store
15	Glendale	Arts & Crafts Store	Food & Drink Shop	Pizza Place	Brewery	Food	Food Court	Flower Shop
16	Rego Park	Bakery	Sandwich Place	Donut Shop	Grocery Store	Restaurant	Sushi Restaurant	Pizza Place
17	Woodhaven	Deli / Bodega	Bank	Pharmacy	Hookah Bar	Park	Dive Bar	Donut Shop
18	Ozone Park	Diner	Pharmacy	Pizza Place	Fast Food Restaurant	Gym	Bank	Martial Arts Dojo
19	South Ozone Park	Park	Deli / Bodega	Fast Food Restaurant	Food	Food Truck	Hotel	Bar
20	College Point	Deli / Bodega	Bakery	Karaoke Bar	Latin American Restaurant	Asian Restaurant	Pizza Place	Bar

So now, Brazilians who needs to move to New York have a lot of information about Neighborhoods in Queens (where there are most of Brazilian restaurants) to choose where to leave. Analyzing all those clusters, I prefer to leave in one of neighborhoods on cluster 3. I chose 3 neighborhoods that looks very nice for me because they have good restaurants and gyms around: Long Island City, Forest Hills and Hunters Point. So now I will search for hotels on those places.

1. Search for a specific venue category

```
https://api.foursquare.com/v2/venues/ search ?client_id= CLIENT_ID &
client_secret= CLIENT_SECRET &ll= LATITUDE , LONGITUDE &v= VERSION &
query= QUERY &radius= RADIUS &limit= LIMIT
```

Let's check the latitude and longitude for 3 neighborhoods: Long Island City, Forest Hills and Hunters Point

```
In [50]: neigh_list = ['Long Island City', 'Forest Hills', 'Hunters Point']
possible_neigh_df = queens_data.loc[queens_data['Neighborhood'].isin(neigh_list)]
possible_neigh_df
```

Out[50]:

	Borough	Neighborhood	Latitude	Longitude
6	Queens	Forest Hills	40.725264	-73.844475
10	Queens	Long Island City	40.750217	-73.939202
72	Queens	Hunters Point	40.743414	-73.953868

Checking if there is any hotel around Forest Hills Neighborhood. I used a radius of 700 meters

```
In [63]: latitude=40.725264
longitude=-73.844475
search_query = 'hotel'
radius = 700
url = 'https://api.foursquare.com/v2/venues/search?client_id={} & client_secret={} & ll={} , {} & v={} & query={} & radius={} & limit={} '.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
url
```

```
Out[63]: 'https://api.foursquare.com/v2/venues/search?client_id=VD2VBP2YGIDOVBI DH122JBCUN
1YLN L44EPVCGDUAWBXSB051 & client_secret=X353ACX0FWHWV3ZWKKGDWO4KXKU4YM2DX5CJDU4EM1
2MBZBU & ll=40.725264, -73.844475 & v=20180605 & query=hotel & radius=700 & limit=100 '
```

```
In [56]: results = requests.get(url).json()
results
```

```
Out[56]: {'meta': {'code': 200, 'requestId': '5d7fedb9ad1789002cf71d0b'},
  'response': {'venues': [{'id': '4dd7f0821fc7d8d86645fad7',
    'name': 'Hotel Pennsylvania Preservation Society',
    'location': {'address': '99-22 67th Rd',
      'crossStreet': 'Austin Street & Booth Street',
      'lat': 40.72404943670689,
      'lng': -73.85355930152319,
      'labeledLatLngs': [{'label': 'display',
        'lat': 40.72404943670689,
        'lng': -73.85355930152319}],
      'distance': 778,
      'postalCode': '11375',
      'cc': 'US',
      'city': 'Forest Hills',
      'state': 'NY',
      'country': 'United States',
      'formattedAddress': ['99-22 67th Rd (Austin Street & Booth Street)',
        'Forest Hills, NY 11375',
        'United States']},
    'categories': [{'id': '50328a8e91d4c4b30a586d6c',
      'name': 'Non-Profit',
      'pluralName': 'Non-Profits',
      'shortName': 'Non-Profit',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/building/default_',
        'suffix': '.png'},
      'primary': True}],
    'venuePage': {'id': '49998039'},
    'referralId': 'v-1568665017',
    'hasPerk': False}]}}
```

Get relevant part of JSON and transform it into a pandas dataframe

```
In [60]: # assign relevant part of JSON to venues
venues = results['response']['venues']

# transform venues into a dataframe
dataframe = json_normalize(venues)
dataframe
```

```
Out[60]:
```

	id	name	categories	referralId	hasPerk	location.address:
0	4dd7f0821fc7d8d86645fad7	Hotel Pennsylvania Preservation Society	[{'id': '50328a8e91d4c4b30a586d6c', 'name': 'N...	v-1568665017	False	99-22 67th Rd

Lets see the rating for this hotel


```
In [68]: venue_id = '4dd7f0821fc7d8d86645fad7' # ID of Hotel Pennsylvania Preservation Society
url = 'https://api.foursquare.com/v2/venues/{}?client_id={} & client_secret={} & v={} '.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)
url
```

```
Out[68]: 'https://api.foursquare.com/v2/venues/4dd7f0821fc7d8d86645fad7?client_id=VD2VBP2YGIDOVBI DH122JBCUN1YLN L44EPVCGDUAWBXSB051 & client_secret=X353ACX0FWHHWV3ZWKKGDWO4KXKU4YM2DX5CJDU4EM12MBZBU & v=20180605'
```

```
In [69]: try:
          print(result['response']['venue']['rating'])
        except:
          print('This venue has not been rated yet.')
```

This venue has not been rated yet.

Unfortunately there is no rate for this hotel on Foursquare. So, let me check the hotels on others 2 neighborhoods

```
In [110]: #Long Island City
latitude=40.750217
longitude=-73.939202
search_query = 'hotel'
radius = 700
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&lat={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
results = requests.get(url).json()
# assign relevant part of JSON to venues
venues = results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe
```

Out[110]:

		id	name	categories	referralId	hasPerk	location.address
0	4dfa7f1c8877b30c3988120f		Z NYC Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	11-01 43rd Avenue
1	57f9887a498e90a780b039cc		The Vue Hotel, an Ascend Hotel Collection Member	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	40-47 22nd
2	4b494c3df964a520c36c26e3		Best Western Plaza Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	3934 21st
3	4bc4baedd57beee15cc3479f		Hotel Verve	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	4003 29th
4	4df142ca18386ecb4e26dad1		Z Hotel New York - Rooftop Bar/Lounge	'{"id": "4bf58dd8d48988d1d5941735", "name": "H...	v-1568669215	False	1101 43rd Avenue
5	59d21cf58a6f171bfe1100ef		Boro Hotel Fitness Center	'{"id": "4bf58dd8d48988d175941735", "name": "G...	v-1568669215	False	Level C, Boro Hotel, 38-38, 27th
6	5578639e498eb1d3badc495f		Boro Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	38-28 27th
7	4fe7048c754a0d91c0a9ef30		Q4 Hotel	'{"id": "4bf58dd8d48988d1ee931735", "name": "H...	v-1568669215	False	29-09 Queens Plaza North Lobby
8	57193525498e590d64281443		LIC Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	44 21th Street Long Island City
9	5d56b49827dd5a00082a76ba		Lic Plaza Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	40-36 27th Street
10	4f04ad3d6da1b97b167b2a86		Hotel Vetiver	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	2911 39th Avenue
11	59fa1f641de765135cbe403d		Q4 Hotel and Hostel	'{"id": "4bf58dd8d48988d1ee931735", "name": "H...	v-1568669215	False	29-09 Queens Plaza
12	5a91e6258b98fd465be61c71		Feather Factory Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	2632 Skillman Avenue
13	59f9cafdf193c01073424c15		Giorgio Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	38-60 1st Street, Queens
14	58e1af3b54386d495993cadb		Boro Hotel Lobby Bar	'{"id": "4bf58dd8d48988d116941735", "name": "B...	v-1568669215	False	Niagara Square
15	4ff0bd6fe4b0db68d4b32cc0		Nesva Hotel	'{"id": "4bf58dd8d48988d1fa931735", "name": "H...	v-1568669215	False	39-12 29th
16	5972c4d4db1d817cc8e462f4		Boro Hotel Rooftop Lounge	'{"id": "4bf58dd8d48988d1d5941735", "name": "H...	v-1568669215	False	Boro Hotel
17	51a2cd67498ed44656af0078		Henry Limousine Ltd Main Office At The Z	'{"id": "4bf58dd8d48988d130951735", "name": "T...	v-1568669215	False	1101 43rd Avenue

```
In [111]: # keep only columns that include venue name and ID.
filtered_columns = ['id', 'name']
dataframe_filtered = dataframe.loc[:, filtered_columns]
dataframe_filtered.rename(columns={'name': 'Name'}, inplace=True)
dataframe_filtered.rename(columns={'id': 'ID'}, inplace=True)
dataframe_filtered
```

Out[111]:

	ID	Name
0	4dfa7f1c8877b30c3988120f	Z NYC Hotel
1	57f9887a498e90a780b039cc	The Vue Hotel, an Ascend Hotel Collection Member
2	4b494c3df964a520c36c26e3	Best Western Plaza Hotel
3	4bc4baedd57beee15cc3479f	Hotel Verve
4	4df142ca18386ecb4e26dad1	Z Hotel New York - Rooftop Bar/Lounge
5	59d21cf58a6f171bfe1100ef	Boro Hotel Fitness Center
6	5578639e498eb1d3badc495f	Boro Hotel
7	4fe7048c754a0d91c0a9ef30	Q4 Hotel
8	57193525498e590d64281443	LIC Hotel
9	5d56b49827dd5a00082a76ba	Lic Plaza Hotel
10	4f04ad3d6da1b97b167b2a86	Hotel Vetiver
11	59fa1f641de765135cbe403d	Q4 Hotel and Hostel
12	5a91e6258b98fd465be61c71	Feather Factory Hotel
13	59f9cafd193c01073424c15	Giorgio Hotel
14	58e1af3b54386d495993cadb	Boro Hotel Lobby Bar
15	4ff0bd6fe4b0db68d4b32cc0	Nesva Hotel
16	5972c4d4db1d817cc8e462f4	Boro Hotel Rooftop Lounge
17	51a2cd67498ed44656af0078	Henry Limousine Ltd Main Office At The Z Hotel...
18	522d47cd498e8690612dabec	Z NYC Hotel @Room 903
19	4b1d1678f964a520b90b24e3	Holiday Inn L.I. City-Manhattan View
20	4b64aeef964a52036c82ae3	Ramada Long Island City
21	52f27359498e7163b8ef6279	Loews Hotels

There are a huge list of hotels on Long Island City. So let's create a function to check the rates, likes and tips for all hotels in the list.

```

In [106]: CLIENT_ID = 'VD2VBP2YGIDOVBI DH122JBCUN1YLN L44EPVCGDUAWBX SBO51 '
CLIENT_SECRET = 'X353ACX0FWHWV3ZWKKGDWO4KXKU4YM2DX5CJDU4EM12MBZBU' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

def get_venue_details(venue_id):

    #url to fetch data from foursquare api
    url = 'https://api.foursquare.com/v2/venues/{}?&client_id={}&client_secret={}&v={}'.format(
        venue_id,
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION)

    # get all the data
    results = requests.get(url).json()
    venue_data=results['response']['venue']
    venue_details=[]
    try:
        venue_id=venue_data['id']
        venue_name=venue_data['name']
        venue_likes=venue_data['likes']['count']
        venue_rating=venue_data['rating']
        venue_tips=venue_data['tips']['count']
        venue_details.append([venue_id,venue_name,venue_likes,venue_rating,venue_tips])
    except KeyError:
        pass

    column_names=['ID','Name','Likes','Rating','Tips']
    df = pd.DataFrame(venue_details,columns=column_names)
    return df

```

```
In [112]: # prepare neighborhood list that contains hotels
column_names=['ID','Name','Likes','Rating','Tips']
hotel_list=pd.DataFrame(columns=column_names)
count=1

for row in dataframe_filtered.values.tolist():
    ID,Name=row
    try:
        venue_details=get_venue_details(ID)
        print(venue_details)
        id,name,likes,rating,tips=venue_details.values.tolist()[0]
    except IndexError:
        print('No data available for id=',id)
        # we will assign 0 value for these restaurants as they may have been
        #recently opened or details does not exist in FourSquare Database
        id,name,likes,rating,tips=[0]*5
    print('(',count,'/',len(dataframe_filtered),')','processed')
    hotel_list = hotel_list.append({'ID': id,
                                    'Name' : name,
                                    'Likes' : likes,
                                    'Rating' : rating,
                                    'Tips' : tips
                                    }, ignore_index=True)

    count+=1
```

```

                                ID          Name  Likes  Rating  Tips
0  4dfa7f1c8877b30c3988120f  Z NYC Hotel      74      7.3    38
( 1 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 4dfa7f1c8877b30c3988120f
( 2 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  4b494c3df964a520c36c26e3  Best Western Plaza Hotel      56      6.7    24
( 3 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  4bc4baedd57beee15cc3479f  Hotel Verve      18      6.4    14
( 4 / 22 ) processed

                                ID          Name  Likes  \
0  4df142ca18386ecb4e26dad1  Z Hotel New York - Rooftop Bar/Lounge      45

      Rating  Tips
0         7.1    20
( 5 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 4df142ca18386ecb4e26dad1
( 6 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  5578639e498eb1d3badc495f  Boro Hotel      66      9.0    12
( 7 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  4fe7048c754a0d91c0a9ef30  Q4 Hotel      41      7.2    22
( 8 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  57193525498e590d64281443  LIC Hotel      17      7.1     6
( 9 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 57193525498e590d64281443
( 10 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  4f04ad3d6da1b97b167b2a86  Hotel Vetiver      42      6.6    12
( 11 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 4f04ad3d6da1b97b167b2a86
( 12 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 0
( 13 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 0
( 14 / 22 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 0
( 15 / 22 ) processed

                                ID          Name  Likes  Rating  Tips
0  4ff0bd6fe4b0db68d4b32cc0  Nesva Hotel      12      6.3     8

```

```
In [113]: hotel_list
```

```
Out[113]:
```

	ID	Name	Likes	Rating	Tips
0	4dfa7f1c8877b30c3988120f	Z NYC Hotel	74	7.3	38
1	0	0	0	0.0	0
2	4b494c3df964a520c36c26e3	Best Western Plaza Hotel	56	6.7	24
3	4bc4baedd57beee15cc3479f	Hotel Verve	18	6.4	14
4	4df142ca18386ecb4e26dad1	Z Hotel New York - Rooftop Bar/Lounge	45	7.1	20
5	0	0	0	0.0	0
6	5578639e498eb1d3badc495f	Boro Hotel	66	9.0	12
7	4fe7048c754a0d91c0a9ef30	Q4 Hotel	41	7.2	22
8	57193525498e590d64281443	LIC Hotel	17	7.1	6
9	0	0	0	0.0	0
10	4f04ad3d6da1b97b167b2a86	Hotel Vetiver	42	6.6	12
11	0	0	0	0.0	0
12	0	0	0	0.0	0
13	0	0	0	0.0	0
14	0	0	0	0.0	0
15	4ff0bd6fe4b0db68d4b32cc0	Nesva Hotel	12	6.3	8
16	0	0	0	0.0	0
17	0	0	0	0.0	0
18	0	0	0	0.0	0
19	4b1d1678f964a520b90b24e3	Holiday Inn L.I. City-Manhattan View	30	6.7	17
20	4b64aeef964a52036c82ae3	Ramada Long Island City	16	4.9	15
21	0	0	0	0.0	0

We see that the hotel with best rate is Boro Hotel. Now let check hotels in Hunters Point.


```

In [114]: #Hunters Point
latitude=40.743414
longitude=-73.95386
search_query = 'hotel'
radius = 700
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{ }&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
results = requests.get(url).json()
# assign relevant part of JSON to venues
venues = results['response']['venues']

# transform venues into a dataframe
dataframe = json_normalize(venues)
dataframe

```

Out[114]:

		id	name	categories	referralld	hasPerk	location.address	I
0	4e87b2e329c23b6afa01d71a		The Box House Hotel	[[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...'}]]	v-1568669495	False	77 Box St	
1	56dca6a6cd1056a1df5908de		Hotel Bennica Grand Café	[[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...'}]]	v-1568669495	False	NaN	
2	4fe76145e4b0a0c3b0b59f23		Desi' Hotel Lounge	[[{'id': '4d954b06a243a5684965b473', 'name': 'R...'}]]	v-1568669495	False	NaN	

```

In [115]: # keep only columns that include venue name, and anything that is associated with location
filtered_columns = ['id', 'name']
dataframe_filtered = dataframe.loc[:, filtered_columns]
dataframe_filtered.rename(columns={'name': 'Name'}, inplace=True)
dataframe_filtered.rename(columns={'id': 'ID'}, inplace=True)
dataframe_filtered

```

Out[115]:

	ID	Name
0	4e87b2e329c23b6afa01d71a	The Box House Hotel
1	56dca6a6cd1056a1df5908de	Hotel Bennica Grand Café
2	4fe76145e4b0a0c3b0b59f23	Desi' Hotel Lounge

```
In [116]: # prepare neighborhood list that contains hotels
column_names=['ID','Name','Likes','Rating','Tips']
hotel_list=pd.DataFrame(columns=column_names)
count=1

for row in dataframe_filtered.values.tolist():
    ID,Name=row
    try:
        venue_details=get_venue_details(ID)
        print(venue_details)
        id,name,likes,rating,tips=venue_details.values.tolist()[0]
    except IndexError:
        print('No data available for id=',id)
        # we will assign 0 value for these restaurants as they may have been
        # recently opened or details does not exist in FourSquare Database
        id,name,likes,rating,tips=[0]*5
    print('(',count,'/',len(dataframe_filtered),')','processed')
    hotel_list = hotel_list.append({'ID': id,
                                    'Name' : name,
                                    'Likes' : likes,
                                    'Rating' : rating,
                                    'Tips' : tips
                                    }, ignore_index=True)

    count+=1
```

```

              ID              Name  Likes  Rating  Tips
0  4e87b2e329c23b6afa01d71a  The Box House Hotel    102    8.4    22
( 1 / 3 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 4e87b2e329c23b6afa01d71a
( 2 / 3 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 0
( 3 / 3 ) processed
```

That is not a very good rating. Let's check the rating of the second closest Italian restaurant.

```
In [117]: hotel_list
```

```
Out[117]:
```

	ID	Name	Likes	Rating	Tips
0	4e87b2e329c23b6afa01d71a	The Box House Hotel	102	8.4	22
1	0	0	0	0.0	0
2	0	0	0	0.0	0

The Box House Hotel looks a very good option too. Let me check now if there is train or metro around the both hotels that is better rated: The Box House Hotel around Hunters Point or Boro Hotel in Long Island City.

```

In [122]: #Box House Hotel
latitude=40.737683
longitude=-73.953455
search_query = 'subway'
radius = 500
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&latitude={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
results = requests.get(url).json()
# assign relevant part of JSON to venues
venues = results['response']['venues']

# transform venues into a dataframe
dataframe = json_normalize(venues)
dataframe

```

Out[122]:

	id	name	categories	referralId	hasPerk	location.address
0	4b07380af964a520ebf922e3	MTA Subway - Greenpoint Ave (G)	[{'id': '4bf58dd8d48988d1fd931735', 'name': 'M...	v-1568670491	False	Greenpoint Ave
1	4f024eb48b81b0190d2d5192	SUBWAY	[{'id': '4bf58dd8d48988d1c5941735', 'name': 'S...	v-1568670491	False	10-46 Jackson Ave
2	4e24110122717a5245e4919e	Subway Parking	[{'id': '4d4b7105d754a06375d81259', 'name': 'P...	v-1568670491	False	10-40 Border Ave
3	4b526470f964a5200a7b27e3	MTA Bus - G Train Shuttle	[{'id': '4bf58dd8d48988d12b951735', 'name': 'B...	v-1568670491	False	NaN

```
In [121]: #Boro Hotel
latitude=40.754708
longitude=-73.935824
search_query = 'subway'
radius = 500
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&lat={}&lon={}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
results = requests.get(url).json()
# assign relevant part of JSON to venues
venues = results['response']['venues']

# transform venues into a dataframe
dataframe = json_normalize(venues)
dataframe
```

Out[121]:

		id	name	categories	referralId	hasPerk	location.address
0	4ad684abf964a520970721e3		MTA Subway - Queensboro Plaza (7/N/W)	{'id': '4bf58dd8d48988d1fd931735', 'name': 'M...	v-1568670458	False	Queensbor Plaz
1	4bebd433a9900f4799331840		MTA Subway - Queens Plaza (E/M/R)	{'id': '4bf58dd8d48988d1fd931735', 'name': 'M...	v-1568670459	False	Queens Plaza :
2	4b2c5b4ff964a5203ac624e3		MTA Subway - 39th Ave (N/W)	{'id': '4bf58dd8d48988d1fd931735', 'name': 'M...	v-1568670459	False	39th Av
3	50789d32e4b06da1414ea72c		SUBWAY	{'id': '4bf58dd8d48988d1c5941735', 'name': 'S...	v-1568670459	False	39-42 21st S

Results

Both hotels are near to subways stations. So I can stay on one of those two. Now I will need only check the price and availability of them to decide where will be my home for some months!

Discussion section

Doing this analysis we found out that Queens Borough have the most of Brazilian restaurants. So we analyzed each neighborhood in Queens to know if any of them would be a good place for Brazilians stay for some months or even leave in New York city. We found out that most of neighborhoods have a lot of restaurants and others places that Brazilians love like Gyms, Bars, Supermarkets, Parks, Coffees and there are public transportation available. We also clustered the neighborhoods on 7 clusters to check how similar were them.

Conclusion

Each person can check the list of venues for each neighborhood and decide what would be the best one to stay. I decided to stay in Hunters Point or Long Island City, because both neighborhoods are similiar and have good hotels.