CLICK THE LINK TO VIEW THE MAPS
Open Notebook (https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/e0282939-39bb-4d4f-af2e-ee4961c06467/view?access_token=5c845e2cb947e1b6a8e9dca7fb2858a77deed1866089c1ded976ca3c161a3538)

Location recommender system for opening a new restaurant

Business Problem

A chef wants to open his own restaurant in Brooklyn, New York City. He wants to know the best area to open a restaurant by analyzing different neighborhoods. Therefore, solution for this problem is to create a machine learning recommender system which will help the chef to make better and informed decision on best areas to open the restaurant

This Project would be helpful for many people to take a better decision on choosing the best neighborhood out of many neighborhoods to build/open their restaurants in Brooklyn, New York City based on the distribution of various different restaurants in and around that neighborhood. Therefore, this project would compare various restaurants and other locations in that neighborhoods and analyses the top 10 most common venues to open a restaurant. Also, this project uses K-mean clustering unsupervised machine learning algorithm to cluster the venues based on the place category such as restaurants and other locations.

As an example, this project would give a better understanding for the small business people to find out the best areas to start a business.

Data

This project would use Four-square API as its prime data gathering source to get details of the presense of similar restaurants in each neighborhood. Foursquare rest-based API provide details like Venue Name, Catagory, customer likes, etc.
The data from Foursquare will be fetched using the longitude and latiatude details of neighborhoods around Brooklyn.

This project also requires New York City GeoJson data which contains details like Borough, Neighborhood, Latitude, Longitude of each major locations in New York.

The idea is to use K-mean clustering unsupervised machine learning algorithm to cluster the venues based on the place category such as restaurants and other locations.

Methodology

Import all the necessary libraries

In [1]:
```
import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import numpy as np # library to handle data in a vectorized manner

# 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

import json # library to handle JSON files
!conda install -c conda-forge geopy --yes # uncomment this line if you haven
't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and
longitude values

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

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if yo
u haven't completed the Foursquare API lab
import folium # map rendering library

from tqdm import tqdm

print('All libraries imported !')
```

```
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

All libraries imported !
```

download New York City json file

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

```
Data downloaded!
```

Open json file

In [10]:

```
with open('newyork_data.json') as f:
    newyork_data = json.load(f)
neighbor_data = newyork_data['features']
neighbor_data[0] #display the data at index 0
```

Out[10]:

```
{'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]}}
```

Extract borough, neighborhood, latitude and longitude from json file to pandas dataframe

In [11]:

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

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

for data in neighbor_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}, ig
nore_index=True)

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

Analize the data by finding total no. of neighborhoods in each borough in New York City

In [12]:
```
neighborhoods.groupby('Borough').count()
```

Out[12]:

	Neighborhood	Latitude	Longitude
Borough			
Bronx	52	52	52
Brooklyn	70	70	70
Manhattan	40	40	40
Queens	81	81	81
Staten Island	63	63	63

From above data we can see that Brooklyn is in 2nd place which has highest Neighborhood

Get all the neighborhoods only in Brooklyn

In [13]:
```
brooklyn_data = neighborhoods[neighborhoods['Borough'] == 'Brooklyn'].reset_
index(drop=True)
brooklyn_data.head()
```

Out[13]:

	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

In [14]:
```
brooklyn_data.shape
```

Out[14]:
```
(70, 4)
```

To get the geograpical coordinate of Brooklyn

In [15]:
```
address = 'Brooklyn, New York, United States'
geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Brooklyn are {}, {}.'.format(latitude,
longitude))
```

```
/home/sam/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: Depr
ecationWarning: Using Nominatim with the default "geopy/1.18.1" `user_agent`
is strongly discouraged, as it violates Nominatim's ToS https://operations.os
mfoundation.org/policies/nominatim/ and may possibly cause 403 and 429 HTTP e
rrors. Please specify a custom `user_agent` with `Nominatim(user_agent="my-ap
plication")` or by overriding the default `user_agent`: `geopy.geocoders.opti
ons.default_user_agent = "my-application"`. In geopy 2.0 this will become an
exception.


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

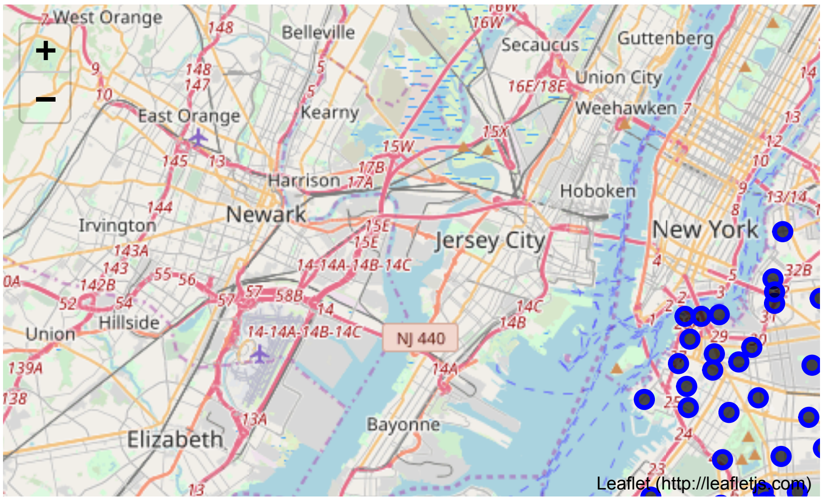
Using the geograpical coordinate create a map of Brooklyn and mark all the neighborhood locations

In [18]:
```
# create map of Brooklyn 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_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#000000',
        fill_opacity=0.7,
        parse_html=False).add_to(map_brooklyn)

map_brooklyn
```

Out[18]:



This map shows all the neighborhoods only in Brooklyn

To get the nearby locations like restaurants, parks and other areas which are in the neighborhoods of Brooklyn, we need to use Foursquare rest-based API.

In [19]:
```
CLIENT_ID = 'ZSQMX00DOTDY5KTFDEFVG1KYD35CNTZECIYXWPMTVJDTPFBE' # your Foursquare ID
CLIENT_SECRET = 'WNQ3UDARPAHP5GQDTOYLVL2SCTU1KXVWWSPVN23GI0CZKQYA' # your Foursquare Secret
VERSION = '20281216' # Foursquare API version
```

In [20]:

```python
def getNearbyVenues(names, latitudes, longitudes, radius=500, LIMIT=200):

    pbar = tqdm(total=36)
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        pbar.update(1)



        # 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])


    pbar.close()

    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 [22]: brooklyn_venues = getNearbyVenues(names=brooklyn_data['Neighborhood'],
                                           latitudes=brooklyn_data['Latitude'],
                                           longitudes=brooklyn_data['Longitude']
                                           )
```

```
  0%|          | 0/36 [00:00<?, ?it/s]
  6%|▌         | 2/36 [00:02<00:47,  1.40s/it]
  8%|▊         | 3/36 [00:04<00:51,  1.56s/it]
 11%|█         | 4/36 [00:06<00:52,  1.65s/it]
 14%|█▍        | 5/36 [00:10<01:08,  2.21s/it]
 17%|█▋        | 6/36 [00:12<01:10,  2.33s/it]
 19%|█▉        | 7/36 [00:15<01:10,  2.43s/it]
 22%|██▏       | 8/36 [00:17<01:08,  2.44s/it]
 25%|██▌       | 9/36 [00:20<01:05,  2.43s/it]
 28%|██▊       | 10/36 [00:22<01:00,  2.33s/it]
 31%|███       | 11/36 [00:24<00:57,  2.31s/it]
 33%|███▎      | 12/36 [00:26<00:50,  2.11s/it]
 36%|███▌      | 13/36 [00:28<00:49,  2.15s/it]
 39%|███▉      | 14/36 [00:31<00:50,  2.31s/it]
 42%|████▏     | 15/36 [00:34<00:53,  2.53s/it]
 44%|████▍     | 16/36 [00:36<00:50,  2.51s/it]
 47%|████▋     | 17/36 [00:39<00:48,  2.56s/it]
 50%|█████     | 18/36 [00:41<00:44,  2.47s/it]
 53%|█████▎    | 19/36 [00:44<00:42,  2.53s/it]
 56%|█████▌    | 20/36 [00:48<00:47,  3.00s/it]
 58%|█████▊    | 21/36 [00:51<00:44,  2.95s/it]
 61%|██████    | 22/36 [00:54<00:41,  2.93s/it]
 64%|██████▍   | 23/36 [00:56<00:37,  2.85s/it]
 67%|██████▋   | 24/36 [00:59<00:32,  2.67s/it]
 69%|██████▉   | 25/36 [01:01<00:29,  2.72s/it]
 72%|███████▏  | 26/36 [01:05<00:29,  2.92s/it]
 75%|███████▌  | 27/36 [01:07<00:24,  2.70s/it]
 78%|███████▊  | 28/36 [01:09<00:20,  2.56s/it]
 81%|████████  | 29/36 [01:12<00:18,  2.59s/it]
 83%|████████▎ | 30/36 [01:14<00:15,  2.59s/it]
 86%|████████▌ | 31/36 [01:17<00:13,  2.64s/it]
 89%|████████▉ | 32/36 [01:20<00:10,  2.64s/it]
 92%|█████████▏| 33/36 [01:22<00:07,  2.41s/it]
 94%|█████████▍| 34/36 [01:23<00:04,  2.20s/it]
 97%|█████████▋| 35/36 [01:25<00:02,  2.07s/it]
100%|██████████| 36/36 [01:28<00:00,  2.25s/it]
37it [01:30,  2.13s/it]
38it [01:32,  2.22s/it]
39it [01:34,  2.22s/it]
40it [01:37,  2.43s/it]
41it [01:40,  2.50s/it]
42it [01:44,  2.92s/it]
43it [01:46,  2.67s/it]
44it [01:48,  2.47s/it]
45it [01:50,  2.47s/it]
46it [01:53,  2.59s/it]
47it [01:56,  2.55s/it]
48it [01:58,  2.58s/it]
49it [02:01,  2.61s/it]
50it [02:03,  2.55s/it]
51it [02:07,  2.77s/it]
52it [02:11,  3.12s/it]
53it [02:13,  2.92s/it]
54it [02:15,  2.69s/it]
55it [02:18,  2.66s/it]
56it [02:20,  2.40s/it]
57it [02:22,  2.45s/it]
58it [02:24,  2.30s/it]
59it [02:26,  2.28s/it]
60it [02:28,  2.03s/it]
61it [02:30,  2.16s/it]
62it [02:33,  2.37s/it]
63it [02:36,  2.59s/it]
64it [02:38,  2.48s/it]
65it [02:41,  2.48s/it]
66it [02:43,  2.44s/it]
67it [02:45,  2.35s/it]
68it [02:48,  2.32s/it]
```

In [27]:
```
print("Total venues found in the neighborhoods brooklyn: ", brooklyn_venues.shape[0])
```

```
Total venues found in the neighborhoods brooklyn:  2822
```

In [28]:
```
brooklyn_venues.head()
```

Out[28]:

	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	Cocoa Grinder	40.623967	-74.030863	Juice Bar
2	Bay Ridge	40.625801	-74.030621	Bagel Boy	40.627896	-74.029335	Bagel Shop
3	Bay Ridge	40.625801	-74.030621	Pegasus Cafe	40.623168	-74.031186	Breakfast Spot
4	Bay Ridge	40.625801	-74.030621	Ho' Brah Taco Joint	40.622960	-74.031371	Taco Place

In [29]:
```
print('There are {} uniques venue categories.'.format(len(brooklyn_venues['Venue Category'].unique())))
```

```
There are 283 uniques venue categories.
```

In [32]:
```
brooklyn_onehot = pd.get_dummies(brooklyn_venues[['Venue Category']], prefix="", prefix_sep="")
brooklyn_onehot['Neighborhood'] = brooklyn_venues['Neighborhood']
fixed_columns = [brooklyn_onehot.columns[-1]] + list(brooklyn_onehot.columns[:-1])
brooklyn_onehot = brooklyn_onehot[fixed_columns]
brooklyn_onehot.head()
```

Out[32]:

	Yoga Studio	Adult Boutique	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Arts & Crafts Store	Arts & Entertainment	As Restau
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	

In [33]:
```
brooklyn_onehot.shape
```

Out[33]:
```
(2822, 283)
```

Extract all the places like restaurants to a new dataframe

In [34]:
```
brooklyn_restaurant = brooklyn_onehot.filter(regex='Neighborhood|Restaurant|Pizza|Coffee|Café|Food Court')
brooklyn_restaurant.shape
```

Out[34]:
```
(2822, 61)
```

In [36]:
```
brooklyn_restaurants = brooklyn_restaurant.groupby('Neighborhood').mean().re
set_index()
brooklyn_restaurants.head()
```

Out[36]:

	Neighborhood	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Café	Cajun / Creole Restaurant	Cantonese Restaurant	Car Rest
0	Bath Beach	0.000000	0.0	0.0	0.020000	0.000000	0.0	0.02	
1	Bay Ridge	0.033333	0.0	0.0	0.011111	0.011111	0.0	0.00	
2	Bedford Stuyvesant	0.000000	0.0	0.0	0.000000	0.071429	0.0	0.00	
3	Bensonhurst	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.00	
4	Bergen Beach	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.00	

In [37]:
```
brooklyn_restaurants.shape
```

Out[37]:
```
(70, 61)
```

Extract all the other places apart from places like restaurants into another new dataframe

In [35]:
```
other_cols = [col for col in brooklyn_onehot.columns if 'Restaurant' not in
col\
                                                       and 'Pizza' not in col\
                                                       and 'Coffee' not in col
\
                                                       and 'Café' not in col\
                                                       and 'Food Court' not in
col]

brooklyn_others = brooklyn_onehot[other_cols]
brooklyn_others.shape
```

Out[35]:
```
(2822, 223)
```

In [38]:
```
brooklyn_others_grouped = brooklyn_others.groupby('Neighborhood').mean().res
et_index()
brooklyn_others_grouped.head()
```

Out[38]:

	Neighborhood	Yoga Studio	Adult Boutique	Antique Shop	Art Gallery	Arts & Crafts Store	Arts & Entertainment	Athletics & Sports	Automotive Shop	I J
0	Bath Beach	0.0	0.0	0.0	0.000000	0.0	0.0	0.000	0.0	0.000
1	Bay Ridge	0.0	0.0	0.0	0.011111	0.0	0.0	0.000	0.0	0.000
2	Bedford Stuyvesant	0.0	0.0	0.0	0.000000	0.0	0.0	0.000	0.0	0.035
3	Bensonhurst	0.0	0.0	0.0	0.000000	0.0	0.0	0.000	0.0	0.000
4	Bergen Beach	0.0	0.0	0.0	0.000000	0.0	0.0	0.125	0.0	0.000

In [39]:
```
brooklyn_others_grouped.shape
```

Out[39]:
```
(70, 223)
```

Finding top 10 most common restaurants in every neighborhoods of Brooklyn

In [40]:
```
def 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 [41]:
```
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 Restaurant'.format(ind+1, indicator
s[ind]))
    except:
        columns.append('{}th Most Common Restaurant'.format(ind+1))

# create a new dataframe
neighborhoods_restaurant_sorted = pd.DataFrame(columns=columns)
neighborhoods_restaurant_sorted['Neighborhood'] = brooklyn_restaurants['Neig
hborhood']

for ind in np.arange(brooklyn_restaurants.shape[0]):
    neighborhoods_restaurant_sorted.iloc[ind, 1:] = most_common_venues(brook
lyn_restaurants.iloc[ind, :], num_top_venues)

neighborhoods_restaurant_sorted.head()
```

Out[41]:

	Neighborhood	1st Most Common Restaurant	2nd Most Common Restaurant	3rd Most Common Restaurant	4th Most Common Restaurant	5th Most Common Restaurant	6th Most Common Restaurant	7th Most Common Restaurant	(Re
0	Bath Beach	Pizza Place	Chinese Restaurant	Italian Restaurant	Sushi Restaurant	Fast Food Restaurant	Asian Restaurant	Cantonese Restaurant	
1	Bay Ridge	Pizza Place	Italian Restaurant	American Restaurant	Greek Restaurant	Thai Restaurant	Chinese Restaurant	Sushi Restaurant	R
2	Bedford Stuyvesant	Coffee Shop	Pizza Place	Café	Japanese Restaurant	New American Restaurant	Vietnamese Restaurant	Hawaiian Restaurant	R
3	Bensonhurst	Chinese Restaurant	Sushi Restaurant	Vietnamese Restaurant	Hotpot Restaurant	Pizza Place	Italian Restaurant	Shabu-Shabu Restaurant	
4	Bergen Beach	Vietnamese Restaurant	Vegetarian / Vegan Restaurant	Italian Restaurant	Israeli Restaurant	Indian Restaurant	Hotpot Restaurant	Hawaiian Restaurant	R

Finding top 10 most common places other than restaurants in every neighborhoods of Brooklyn

In [42]:
```
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_others_sorted = pd.DataFrame(columns=columns)
neighborhoods_others_sorted['Neighborhood'] = brooklyn_others_grouped['Neigh
borhood']

for ind in np.arange(brooklyn_others_grouped.shape[0]):
    neighborhoods_others_sorted.iloc[ind, 1:] = most_common_venues(brooklyn_
others_grouped.iloc[ind, :], num_top_venues)

neighborhoods_others_sorted.head()
```

Out[42]:

	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	Bath Beach	Donut Shop	Pharmacy	Kids Store	Smoke Shop	Sandwich Place	Rental Car Location	Playground	Park
1	Bay Ridge	Spa	Bar	Bagel Shop	Pharmacy	Hookah Bar	Sandwich Place	Playground	Ice Cream Shop
2	Bedford Stuyvesant	Bus Stop	Bar	Deli / Bodega	BBQ Joint	Bus Station	Boutique	Gourmet Shop	Basketball Court
3	Bensonhurst	Ice Cream Shop	Park	Grocery Store	Dessert Shop	Playground	Cosmetics Shop	Donut Shop	Noodle House
4	Bergen Beach	Harbor / Marina	Playground	Athletics & Sports	Baseball Field	Park	Donut Shop	Hockey Field	Diner

Using K-means clustering technique for clustering all the areas without restaurants

In [43]:
```
# set number of clusters
kclusters = 5

brooklyn_others_clustering = brooklyn_others_grouped.drop('Neighborhood', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

Out[43]:
```
array([4, 0, 0, 0, 0, 0, 2, 4, 2, 0, 4, 0, 4, 0, 4, 0, 0, 4, 4, 4, 4, 0,
       0, 4, 4, 2, 0, 0, 4, 4, 0, 4, 0, 4, 0, 0, 0, 0, 4, 4, 4, 4, 4, 4,
       2, 4, 4, 1, 4, 0, 4, 4, 4, 0, 0, 4, 4, 0, 4, 4, 3, 4, 0, 4, 4, 0,
       0, 0, 0, 4], dtype=int32)
```

In [44]:

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

brooklyn_others_merged = brooklyn_data

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

brooklyn_others_merged.head()
```

Out[44]:

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5tl Co
0	Brooklyn	Bay Ridge	40.625801	-74.030621	0	Spa	Bar	Bagel Shop	Pharmacy	H
1	Brooklyn	Bensonhurst	40.611009	-73.995180	0	Ice Cream Shop	Park	Grocery Store	Dessert Shop	Play
2	Brooklyn	Sunset Park	40.645103	-74.010316	4	Bakery	Bank	Gym	Mobile Phone Shop	Pha
3	Brooklyn	Greenpoint	40.730201	-73.954241	0	Bar	Cocktail Bar	Yoga Studio	Record Shop	
4	Brooklyn	Gravesend	40.595260	-73.973471	0	Lounge	Bus Station	Bakery	Spa	G

Locate all the places without restaurants

In [45]:

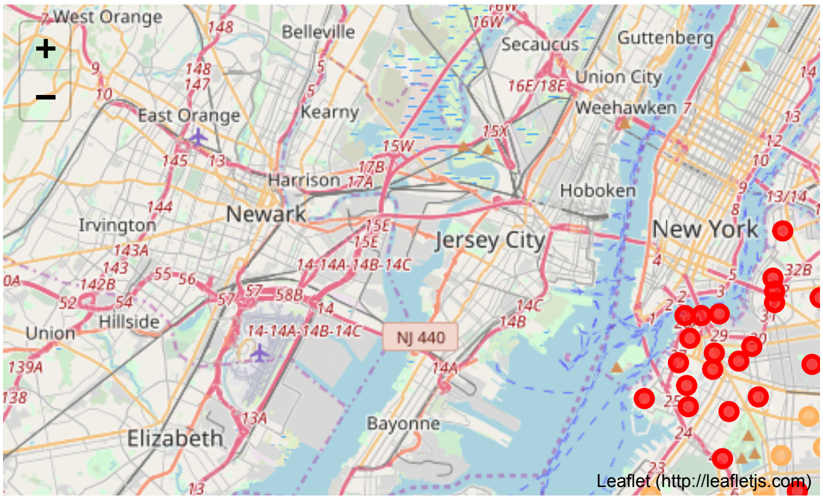
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# 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(brooklyn_others_merged['Latitude'],\
                                  brooklyn_others_merged['Longitude'],\
                                  brooklyn_others_merged['Neighborhood'],\
                                  brooklyn_others_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=T
rue)
    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[45]:



Display All the clusters created

Cluster 1

In [46]:

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

Out[46]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Bay Ridge	Spa	Bar	Bagel Shop	Pharmacy	Hookah Bar
1	Bensonhurst	Ice Cream Shop	Park	Grocery Store	Dessert Shop	Playground
3	Greenpoint	Bar	Cocktail Bar	Yoga Studio	Record Shop	Bakery
4	Gravesend	Lounge	Bus Station	Bakery	Spa	Grocery Store
12	Windsor Terrace	Plaza	Food Truck	Diner	Park	Beer Store
13	Prospect Heights	Bar	Cocktail Bar	Wine Shop	Gourmet Shop	Bakery
15	Williamsburg	Bar	Bagel Shop	Yoga Studio	Breakfast Spot	Steakhouse
16	Bushwick	Bar	Deli / Bodega	Discount Store	Thrift / Vintage Store	Bakery
17	Bedford Stuyvesant	Bus Stop	Bar	Deli / Bodega	BBQ Joint	Bus Station
18	Brooklyn Heights	Yoga Studio	Deli / Bodega	Park	Cosmetics Shop	Gym
19	Cobble Hill	Bar	Ice Cream Shop	Playground	Cocktail Bar	Deli / Bodega
20	Carroll Gardens	Bakery	Cocktail Bar	Wine Shop	Gym / Fitness Center	Bar
21	Red Hook	Art Gallery	Park	Ice Cream Shop	Bagel Shop	Wine Shop
22	Gowanus	Bar	Gym / Fitness Center	Furniture / Home Store	Food Truck	Nightclub
23	Fort Greene	Flower Shop	Yoga Studio	Opera House	Theater	Bar
24	Park Slope	Burger Joint	Bagel Shop	Bakery	Bookstore	Pub
36	Gerritsen Beach	Bar	Ice Cream Shop	Bagel Shop	Department Store	Convenience Store
38	Clinton Hill	Yoga Studio	Wine Shop	Grocery Store	Cocktail Bar	Bagel Shop
40	Downtown	Burger Joint	Sandwich Place	Spa	Bakery	Gift Shop
41	Boerum Hill	Dance Studio	Spa	Bar	Yoga Studio	Bakery
45	Bergen Beach	Harbor / Marina	Playground	Athletics & Sports	Baseball Field	Park
49	East Williamsburg	Bar	Deli / Bodega	Cocktail Bar	Concert Hall	Music Venue
50	North Side	Yoga Studio	Bar	Cocktail Bar	Jewelry Store	Burger Joint
51	South Side	Bar	Breakfast Spot	Burger Joint	BBQ Joint	Pub
61	Fulton Ferry	Park	Scenic Lookout	Ice Cream Shop	Bakery	Hotel Bar
62	Vinegar Hill	Food Truck	Wine Shop	Art Gallery	Bookstore	Ice Cream Shop
63	Weeksville	Discount Store	Deli / Bodega	Cocktail Bar	Check Cashing Service	Lounge
65	Dumbo	Park	Scenic Lookout	Bookstore	Wine Shop	Bakery
69	Erasmus	Bar	Yoga Studio	Bank	Convenience Store	Donut Shop

Cluster 2

In [47]:

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

Out[47]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
30	Mill Island	Pool	Food	Lake	Factory	Fried Chicken Joint

Cluster 3

In [48]:

```
brooklyn_others_merged.loc[brooklyn_others_merged['Cluster Labels'] == 2,\
                                 brooklyn_others_merged.columns[[1]\
                               + list(range(5, brooklyn_others_merged.shape[1
] -5))]]
```

Out[48]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
26	East New York	Deli / Bodega	Pharmacy	Metro Station	Gym	Women's Store
34	Borough Park	Deli / Bodega	Bank	Pharmacy	Farmers Market	Metro Station
37	Marine Park	Ice Cream Shop	Athletics & Sports	Soccer Field	Gym	Basketball Court
64	Broadway Junction	Deli / Bodega	Metro Station	Diner	Donut Shop	Recording Studio

Cluster 4

In [49]:

```
brooklyn_others_merged.loc[brooklyn_others_merged['Cluster Labels'] == 3,\
                                 brooklyn_others_merged.columns[[1]\
                               + list(range(5, brooklyn_others_merged.shape[1
] -5))]]
```

Out[49]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
39	Sea Gate	Optical Shop	Spa	Video Store	Clothing Store	Bus Station

Cluster 5

In [50]:

```
brooklyn_others_merged.loc[brooklyn_others_merged['Cluster Labels'] == 4,\
                                brooklyn_others_merged.columns[[1]\
                              + list(range(5, brooklyn_others_merged.shape[1
] -5))]]
```

Out[50]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
2	Sunset Park	Bakery	Bank	Gym	Mobile Phone Shop	Pharmacy
5	Brighton Beach	Beach	Bank	Pharmacy	Gourmet Shop	Mobile Phone Shop
6	Sheepshead Bay	Dessert Shop	Sandwich Place	Buffet	Yoga Studio	Boat or Ferry
7	Manhattan Terrace	Bakery	Cosmetics Shop	Donut Shop	Bagel Shop	Ice Cream Shop
8	Flatbush	Plaza	Juice Bar	Pharmacy	Donut Shop	Sandwich Place
9	Crown Heights	Museum	Metro Station	Liquor Store	Playground	Park
10	East Flatbush	Hardware Store	Wine Shop	Deli / Bodega	Pharmacy	Check Cashing Service
11	Kensington	Grocery Store	Ice Cream Shop	Sandwich Place	Donut Shop	Racetrack
14	Brownsville	Moving Target	Park	Donut Shop	Playground	Men's Store
25	Cypress Hills	Ice Cream Shop	Fried Chicken Joint	Donut Shop	Deli / Bodega	Liquor Store
27	Starrett City	Moving Target	Intersection	Donut Shop	Supermarket	Gym
28	Canarsie	Gym	Women's Store	Furniture / Home Store	Frozen Yogurt Shop	Fried Chicken Joint
29	Flatlands	Pharmacy	Deli / Bodega	Park	Lounge	Electronics Store
31	Manhattan Beach	Ice Cream Shop	Playground	Food	Bus Stop	Sandwich Place
32	Coney Island	Monument / Landmark	Pharmacy	Bakery	Gourmet Shop	Beach
33	Bath Beach	Donut Shop	Pharmacy	Kids Store	Smoke Shop	Sandwich Place
35	Dyker Heights	Dance Studio	Hot Dog Joint	Burger Joint	Bagel Shop	Golf Course
42	Prospect Lefferts Gardens	Bakery	Ice Cream Shop	Deli / Bodega	Supermarket	Wine Shop
43	Ocean Hill	Deli / Bodega	Playground	Bus Stop	Food	Bakery
44	City Line	Donut Shop	Fried Chicken Joint	Bus Stop	Liquor Store	Food Truck
46	Midwood	Ice Cream Shop	Metro Station	Bakery	Bagel Shop	Video Game Store
47	Prospect Park South	Women's Store	Grocery Store	Pharmacy	Mobile Phone Shop	Clothing Store
48	Georgetown	Breakfast Spot	Pharmacy	Bank	Donut Shop	Miscellaneous Shop
52	Ocean Parkway	Men's Store	Donut Shop	Bus Station	Bus Stop	Sake Bar
53	Fort Hamilton	Gym / Fitness Center	Sandwich Place	Pharmacy	Bagel Shop	Deli / Bodega
54	Ditmas Park	Deli / Bodega	Women's Store	Clothing Store	Metro Station	Donut Shop
55	Wingate	Deli / Bodega	Donut Shop	Juice Bar	Field	Liquor Store
56	Rugby	Supermarket	Ice Cream Shop	Bank	Pharmacy	Diner
57	Remsen Village	Breakfast Spot	Fried Chicken Joint	Supermarket	Donut Shop	Fish Market
58	New Lots	Metro Station	Breakfast Spot	Bank	Pharmacy	Discount Store
59	Paerdegat Basin	Photography Studio	Food	Women's Store	Factory	Fried Chicken Joint
60	Mill Basin	Bank	Burger Joint	Bagel Shop	Ice Cream Shop	Pharmacy
66	Homecrest	Bank	Donut Shop	Grocery Store	Sandwich Place	Shipping Store

To display all kinds of restaurants in Brooklyn

In [56]:
```
restaurants_list = list(brooklyn_restaurants)
restaurants_list.pop(0)
restaurants_list
```

Out[56]:
```
['American Restaurant',
 'Arepa Restaurant',
 'Argentinian Restaurant',
 'Asian Restaurant',
 'Café',
 'Cajun / Creole Restaurant',
 'Cantonese Restaurant',
 'Caribbean Restaurant',
 'Chinese Restaurant',
 'Coffee Shop',
 'Cuban Restaurant',
 'Dim Sum Restaurant',
 'Dumpling Restaurant',
 'Eastern European Restaurant',
 'Ethiopian Restaurant',
 'Falafel Restaurant',
 'Fast Food Restaurant',
 'Filipino Restaurant',
 'Food Court',
 'French Restaurant',
 'German Restaurant',
 'Greek Restaurant',
 'Halal Restaurant',
 'Hawaiian Restaurant',
 'Hotpot Restaurant',
 'Indian Restaurant',
 'Israeli Restaurant',
 'Italian Restaurant',
 'Japanese Restaurant',
 'Jewish Restaurant',
 'Kebab Restaurant',
 'Korean Restaurant',
 'Kosher Restaurant',
 'Latin American Restaurant',
 'Lebanese Restaurant',
 'Mediterranean Restaurant',
 'Mexican Restaurant',
 'Middle Eastern Restaurant',
 'New American Restaurant',
 'Pakistani Restaurant',
 'Peruvian Restaurant',
 'Pizza Place',
 'Polish Restaurant',
 'Ramen Restaurant',
 'Restaurant',
 'Russian Restaurant',
 'Seafood Restaurant',
 'Shabu-Shabu Restaurant',
 'Shanghai Restaurant',
 'South American Restaurant',
 'Southern / Soul Food Restaurant',
 'Spanish Restaurant',
 'Sushi Restaurant',
 'Taiwanese Restaurant',
 'Tapas Restaurant',
 'Thai Restaurant',
 'Tibetan Restaurant',
 'Turkish Restaurant',
 'Vegetarian / Vegan Restaurant',
 'Vietnamese Restaurant']
```

Enter what type of restaurant you want to open

In [57]:
```
restaurant_type = input("restaurant type.. ")
```

```
restaurant type.. Indian Restaurant
```

Extract the location of most common restaurants of this type which is located in the neighborhood

In [60]:
```
col_num = neighborhoods_restaurant_sorted.shape[1]

# define the dataframe columns
col_names = ['Neighborhood', 'Type', 'Most Common']

# instantiate the dataframe
is_common_restaurant = pd.DataFrame(columns=col_names)

for index, row in neighborhoods_restaurant_sorted.iterrows():
    for i in range (1, col_num):
        if restaurant_type in row[i]:
            is_common_restaurant = is_common_restaurant.append({'Neighborhood': row[0],
                                          'Type': row[i], 'Most Common': i}, ignore_index=True)
is_common_restaurant.shape
```

Out[60]:
```
(38, 3)
```

In [61]:

```
brooklyn_restaurant_filtered = is_common_restaurant

brooklyn_restaurant_filtered = brooklyn_restaurant_filtered.join(brooklyn_da
ta.set_index('Neighborhood'), on='Neighborhood')

brooklyn_restaurant_filtered.drop(['Borough'], axis=1, inplace=True)
brooklyn_restaurant_filtered
```

Out[61]:

	Neighborhood	Type	Most Common	Latitude	Longitude
0	Bergen Beach	Indian Restaurant	5	40.615150	-73.898556
1	Boerum Hill	Indian Restaurant	7	40.685683	-73.983748
2	Borough Park	Indian Restaurant	10	40.633131	-73.990498
3	Broadway Junction	Indian Restaurant	6	40.677861	-73.903317
4	Brooklyn Heights	Indian Restaurant	7	40.695864	-73.993782
5	Brownsville	Indian Restaurant	6	40.663950	-73.910235
6	Bushwick	Indian Restaurant	9	40.698116	-73.925258
7	Clinton Hill	Indian Restaurant	7	40.693229	-73.967843
8	Coney Island	Indian Restaurant	7	40.574293	-73.988683
9	Dyker Heights	Indian Restaurant	5	40.619219	-74.019314
10	East Flatbush	Indian Restaurant	7	40.641718	-73.936103
11	East New York	Indian Restaurant	8	40.669926	-73.880699
12	Erasmus	Indian Restaurant	6	40.646926	-73.948177
13	Flatlands	Indian Restaurant	7	40.630446	-73.929113
14	Fulton Ferry	Indian Restaurant	6	40.703281	-73.995508
15	Georgetown	Indian Restaurant	8	40.623845	-73.916075
16	Gerritsen Beach	Indian Restaurant	7	40.590848	-73.930102
17	Gowanus	Indian Restaurant	5	40.673931	-73.994441
18	Gravesend	Indian Restaurant	6	40.595260	-73.973471
19	Highland Park	Indian Restaurant	5	40.681999	-73.890346
20	Madison	Indian Restaurant	7	40.609378	-73.948415
21	Manhattan Beach	Indian Restaurant	6	40.577914	-73.943537
22	Marine Park	Indian Restaurant	5	40.609748	-73.931344
23	Midwood	Indian Restaurant	6	40.625596	-73.957595
24	Mill Island	Indian Restaurant	5	40.606336	-73.908186
25	New Lots	Indian Restaurant	7	40.662744	-73.885118
26	Ocean Hill	Indian Restaurant	7	40.678403	-73.913068
27	Ocean Parkway	Indian Restaurant	7	40.613060	-73.968367
28	Paerdegat Basin	Indian Restaurant	7	40.631318	-73.902335
29	Prospect Lefferts Gardens	Indian Restaurant	4	40.658420	-73.954899
30	Prospect Park South	Indian Restaurant	8	40.647009	-73.962613
31	Remsen Village	Indian Restaurant	8	40.652117	-73.916653
32	Sea Gate	Indian Restaurant	5	40.576375	-74.007873
33	Starrett City	Indian Restaurant	6	40.647589	-73.879370
34	Sunset Park	Indian Restaurant	7	40.645103	-74.010316
35	Vinegar Hill	Indian Restaurant	7	40.703321	-73.981116
36	Weeksville	Indian Restaurant	7	40.675040	-73.930531
37	Wingate	Indian Restaurant	7	40.660947	-73.937187

Plot a map to locate the neighborhood where specific restaurant is common

In [64]:

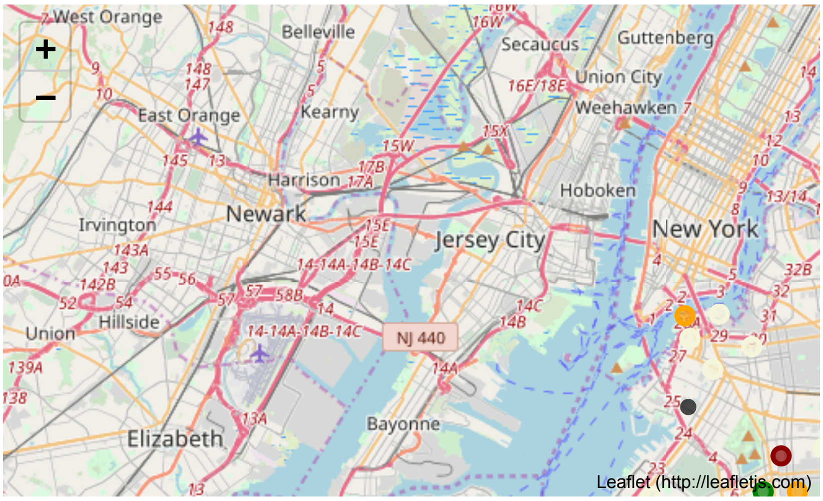
```
map_resturant = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
rainbow = colors = ['red', 'blue', 'gray', 'darkred', 'lightred', 'orange',
'beige',
    'green', 'darkgreen', 'lightgreen', 'darkblue', 'lightblue', 'purple', '
darkpurple',
    'pink', 'cadetblue', 'lightgray', 'black']

# add markers to the map
markers_colors = []
for lat, lon, poi, common in zip(brooklyn_restaurant_filtered['Latitude'],\
                                  brooklyn_restaurant_filtered['Longitude'],
\
                                  brooklyn_restaurant_filtered['Neighborhood
'],\
                                  brooklyn_restaurant_filtered['Most Common'
]):
    label = folium.Popup(str(poi) + ' Most Common: ' + str(common), parse_ht
ml=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[common-1],
        fill=True,
        fill_color=rainbow[common-1],
        fill_opacity=0.7).add_to(map_resturant)

map_resturant
```

Out[64]:



Extract neighborhood where the specific restaurant is not common (Not in top 10)

In [63]:

```
brooklyn_restaurant_suitable = brooklyn_data

brooklyn_restaurant_suitable = brooklyn_restaurant_suitable.join(is_common_r
estaurant.set_index('Neighborhood'), on='Neighborhood')

brooklyn_restaurant_suitable.drop(['Borough'], axis=1, inplace=True)

brooklyn_restaurant_neighborhood_recommendation = brooklyn_restaurant_suitab
le.loc[brooklyn_restaurant_suitable['Most Common'].isnull()]
brooklyn_restaurant_neighborhood_recommendation
```

Out[63]:

	Neighborhood	Latitude	Longitude	Type	Most Common
0	Bay Ridge	40.625801	-74.030621	NaN	NaN
1	Bensonhurst	40.611009	-73.995180	NaN	NaN
3	Greenpoint	40.730201	-73.954241	NaN	NaN
5	Brighton Beach	40.576825	-73.965094	NaN	NaN
6	Sheepshead Bay	40.586890	-73.943186	NaN	NaN
7	Manhattan Terrace	40.614433	-73.957438	NaN	NaN
8	Flatbush	40.636326	-73.958401	NaN	NaN
9	Crown Heights	40.670829	-73.943291	NaN	NaN
11	Kensington	40.642382	-73.980421	NaN	NaN
12	Windsor Terrace	40.656946	-73.980073	NaN	NaN
13	Prospect Heights	40.676822	-73.964859	NaN	NaN
15	Williamsburg	40.707144	-73.958115	NaN	NaN
17	Bedford Stuyvesant	40.687232	-73.941785	NaN	NaN
19	Cobble Hill	40.687920	-73.998561	NaN	NaN
20	Carroll Gardens	40.680540	-73.994654	NaN	NaN
21	Red Hook	40.676253	-74.012759	NaN	NaN
23	Fort Greene	40.688527	-73.972906	NaN	NaN
24	Park Slope	40.672321	-73.977050	NaN	NaN
25	Cypress Hills	40.682391	-73.876616	NaN	NaN
28	Canarsie	40.635564	-73.902093	NaN	NaN
33	Bath Beach	40.599519	-73.998752	NaN	NaN
40	Downtown	40.690844	-73.983463	NaN	NaN
44	City Line	40.678570	-73.867976	NaN	NaN
49	East Williamsburg	40.708492	-73.938858	NaN	NaN
50	North Side	40.714823	-73.958809	NaN	NaN
51	South Side	40.710861	-73.958001	NaN	NaN
53	Fort Hamilton	40.614768	-74.031979	NaN	NaN
54	Ditmas Park	40.643675	-73.961013	NaN	NaN
56	Rugby	40.655572	-73.926882	NaN	NaN
60	Mill Basin	40.615974	-73.915154	NaN	NaN
65	Dumbo	40.703176	-73.988753	NaN	NaN
66	Homecrest	40.598525	-73.959185	NaN	NaN

Result

Suggested neighborhoods to open the restaurant

```
In [66]: brooklyn_restaurant_neighborhood_recommendation[['Neighborhood']]
```

Out[66]:

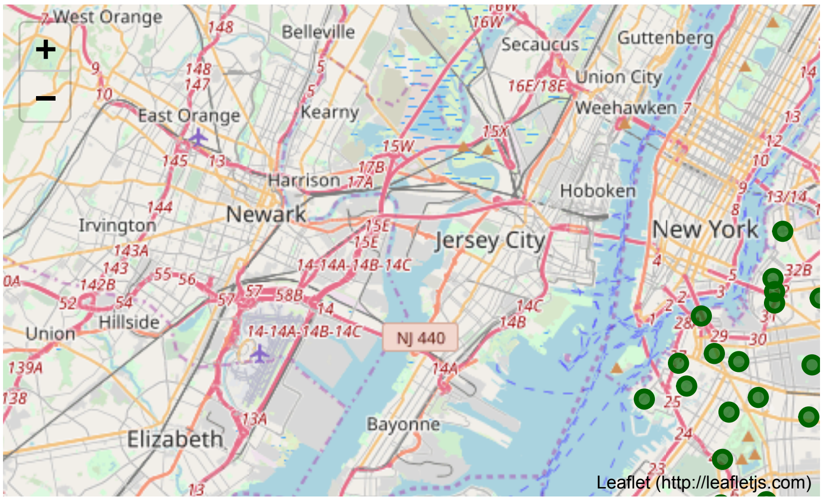
	Neighborhood
0	Bay Ridge
1	Bensonhurst
3	Greenpoint
5	Brighton Beach
6	Sheepshead Bay
7	Manhattan Terrace
8	Flatbush
9	Crown Heights
11	Kensington
12	Windsor Terrace
13	Prospect Heights
15	Williamsburg
17	Bedford Stuyvesant
19	Cobble Hill
20	Carroll Gardens
21	Red Hook
23	Fort Greene
24	Park Slope
25	Cypress Hills
28	Canarsie
33	Bath Beach
40	Downtown
44	City Line
49	East Williamsburg
50	North Side
51	South Side
53	Fort Hamilton
54	Ditmas Park
56	Rugby
60	Mill Basin
65	Dumbo
66	Homecrest

In [69]:
```
# create map
map_resturant = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to the map
markers_colors = []
for lat, lon, poi in zip(brooklyn_restaurant_neighborhood_recommendation['La
titude'],\
                                  brooklyn_restaurant_neighborhood_recommend
ation['Longitude'],\
                                 brooklyn_restaurant_neighborhood_recommenda
tion['Neighborhood']):
    label = folium.Popup(str(poi), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        fill=True,
        fill_opacity=0.7,
        fill_color="#006400",
        color="#006400").add_to(map_resturant)

map_resturant
```

Out[69]:



Discussion

The clustering provides an insight of the similarities in different neighborhoods in Brooklyn by analyzing different restaurants and other venues.

This project finds the best location only by comparing the restaurant type which are already present in Brooklyn. But if the people like our chef wants to open a totally different restaurant type which is not already present in Brooklyn then as a business tactic any neighborhood would be a best choice to open the restaurant

Conclusion

With this analysis can conclude that using the location data from Foresquare along with Machine Learning algorithms like K-means clustering we can design a system that will help to guide small business owners to make informative decision on which is the best neighborhood to a new restaurant.

```
In [ ]:
```