

# Capstone Project: Battle of Neighborhoods

## Step 1: Business Problem

Clearly define a problem or an idea of your choice, where you would need to leverage the Foursquare location data to solve or execute. Remember that data science problems always target an audience and are meant to help a group of stakeholders solve a problem, so make sure that you explicitly describe your audience and why they would care about your problem.

I will be approaching this assignment from a restaurant business perspective. I'll need to use Foursquare to pull existing data on food establishments around the city of Bronx, New York and identify what variety of establishments there are in certain neighborhoods.

## Step 2: Data Pull

```
In [1]: 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 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 # transform JSON file into a pandas dataframe

# 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.')
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
<ipython-input-1-3a9c5a0811ae> in <module>
      8
      9 #!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
--> 10 from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
     11
     12 import requests # library to handle requests

ModuleNotFoundError: No module named 'geopy'
```

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

Data downloaded!

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

```
In [4]: newyork_data
```

```
Out[4]: {'type': 'FeatureCollection',
  'totalFeatures': 306,
  'features': [{'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]}},
    {'type': 'Feature',
      'id': 'nyu_2451_34572.2',
      'geometry': {'type': 'Point',
        'coordinates': [-73.82993910812398, 40.87429419303012]},
      'geometry_name': 'geom',
      'properties': {'name': 'Co-op City',
        'stacked': 2,
        'annoline1': 'Co-op',
        'annoline2': 'City',
        'annoline3': None,
        'annoangle': 0.0,
        'borough': 'Bronx',
        'bbox': [-73.82993910812398,
          40.87429419303012,
          -73.82993910812398,
          40.87429419303012]}},
    {'type': 'Feature',
      'id': 'nyu_2451_34572.3',
      'geometry': {'type': 'Point',
        'coordinates': [-73.82780644716412, 40.887555677350775]},
      'geometry_name': 'geom',
      'properties': {'name': 'Eastchester',
        'stacked': 1,
        'annoline1': 'Eastchester',
        'annoline2': None,
        'annoline3': None,
        'annoangle': 0.0,
        'borough': 'Bronx',
        'bbox': [-73.82780644716412,
          40.887555677350775,
          -73.82780644716412,
          40.887555677350775]}},
    {'type': 'Feature',
      'id': 'nyu_2451_34572.4',
      'geometry': {'type': 'Point',
        'coordinates': [-73.90564259591682, 40.89543742690383]},
      'geometry_name': 'geom',
      'properties': {'name': 'Fieldston',
        'stacked': 1,
        'annoline1': 'Fieldston',
        'annoline2': None,
        'annoline3': None,
        'annoangle': 0.0,
        'borough': 'Bronx',
        'bbox': [-73.90564259591682,
          40.89543742690383,
```

```
In [5]: neighborhoods_data = newyork_data['features']
```

```
In [6]: neighborhoods_data[0]
```

```
Out[6]: {'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]}}
```

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

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

```
In [8]: neighborhoods
```

```
Out[8]:
```

| Borough | Neighborhood | Latitude | Longitude |
|---------|--------------|----------|-----------|
|---------|--------------|----------|-----------|

## Step 2a: Data Processing

In this section the data will be processed into data frames separating boroughs, neighborhoods and lat longs for easier visual digestion.

```
In [9]: 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)
```

```
In [10]: neighborhoods.head()
```

```
Out[10]:
```

|   | 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 |

```
In [11]: print('The dataframe has {} boroughs and {} neighborhoods.'.format(  
            len(neighborhoods['Borough'].unique()),  
            neighborhoods.shape[0]  
        )  
    )
```

The dataframe has 5 boroughs and 306 neighborhoods.

```
In [12]: address = 'Brox, NY'
```

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

The geograpical coordinate of the Bronx are 41.3800936, -74.6923852.

```
In [13]: bronx_data = neighborhoods[neighborhoods['Borough'] == 'Bronx'].reset_index(drop=True)  
        bronx_data.head()
```

```
Out[13]:
```

|   | 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 |

## Step 2b: Data Visualization

Now that the data is in data tables, we can use the information to make a map for visualization to assist in validating the information pulled.

```

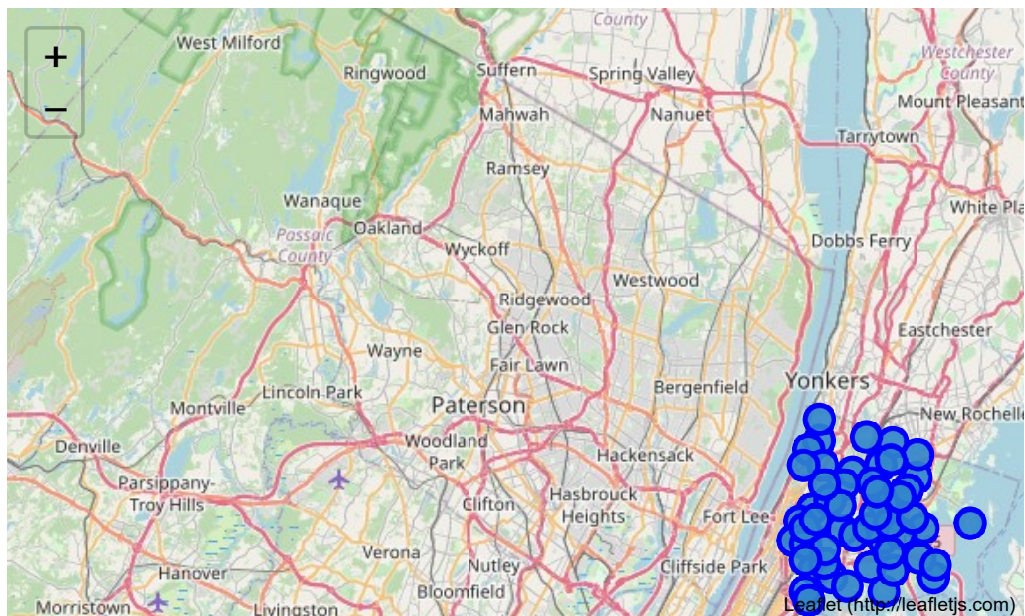
In [26]: # create map of Bronx using latitude and longitude values
map_bronx = folium.Map(location=[40.8448, -73.8648], zoom_start=10)

# add markers to map
for lat, lng, label in zip(bronx_data['Latitude'], bronx_data['Longitude'], bronx_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=8,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.9,
        parse_html=False).add_to(map_bronx)

map_bronx

```

Out [26]:



## Step 2c: Retrieve Information from Foursquare

In this step we are going to obtain venue information for Bronx, New York from Foursquare to identify the top varieties of venues in the city.

```
In [25]: CLIENT_ID = 'KKVBXH3EBDJEPSYKPA5PECJO1F3DXCDEAJ0XYTEM2PWXNZ' # your Foursquare ID
CLIENT_SECRET = 'MVX4FXGCHORFJPWCD2DFEALOT3XYJZOWT3QPILSWZZBU12AU' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

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

```
Your credentials:
CLIENT_ID: KKVBXH3EBDJEPSYKPA5PECJO1F3DXCDEAJ0XYTEM2PWXNZ
CLIENT_SECRET: MVX4FXGCHORFJPWCD2DFEALOT3XYJZOWT3QPILSWZZBU12AU
```

```
In [27]: # define URL
url = 'https://api.foursquare.com/v2/venues/explore?client_id=KKVBXH3EBDJEPSYKPA5PECJO1F3DXCDEAJ0XYTEM2PWXNZ&client_secret=MVX4FXGCHORFJPWCD2DFEALOT3XYJZOWT3QPILSWZZBU12AU&ll=40.8448,-73.8648&v=20181206'
```

```
In [28]: bronx_data.loc[0, 'Neighborhood']
```

```
Out[28]: 'Wakefield'
```

```
In [30]: neighborhood_latitude = bronx_data.loc[0, 'Latitude'] # neighborhood latitude value
neighborhood_longitude = bronx_data.loc[0, 'Longitude'] # neighborhood longitude value

neighborhood_name = bronx_data.loc[0, 'Neighborhood'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}'.format(neighborhood_name,
                                                                neighborhood_latitude,
                                                                neighborhood_longitude))
```

```
Latitude and longitude values of Wakefield are 40.89470517661, -73.84720052054902.
```



```
In [31]: results = requests.get(url).json()  
results
```

```

Out[31]: {'meta': {'code': 200, 'requestId': '5ea35a0c9da7ee001b7b8cda'},
  'response': {'suggestedFilters': {'header': 'Tap to show:',
    'filters': [{'name': '$-$$$$', 'key': 'price'},
      {'name': 'Open now', 'key': 'openNow'}]},
    'suggestedRadius': 1488,
    'headerLocation': 'Van Nest',
    'headerFullLocation': 'Van Nest, Bronx',
    'headerLocationGranularity': 'neighborhood',
    'totalResults': 98,
    'suggestedBounds': {'ne': {'lat': 40.85720122150215,
      'lng': -73.85048499236716},
      'sw': {'lat': 40.833572059264085, 'lng': -73.87933488298252}},
    'groups': [{'type': 'Recommended Places',
      'name': 'recommended',
      'items': [{'reasons': {'count': 0,
        'items': [{'summary': 'This spot is popular',
          'type': 'general',
          'reasonName': 'globalInteractionReason'}]}],
      'venue': {'id': '4c1c5630e9c4ef3b4ccd45aa',
        'name': "Conti's Pastry Shoppe",
        'location': {'address': '786 Morris Park Ave',
          'crossStreet': 'btw Barnes & Wallace',
          'lat': 40.845905639607956,
          'lng': -73.86283608706798,
          'labeledLatLngs': [{'label': 'display',
            'lat': 40.845905639607956,
            'lng': -73.86283608706798},
          {'label': 'entrance', 'lat': 40.84576, 'lng': -73.862837}],
          'distance': 206,
          'postalCode': '10462',
          'cc': 'US',
          'city': 'Bronx',
          'state': 'NY',
          'country': 'United States',
          'formattedAddress': ['786 Morris Park Ave (btw Barnes & Wallace)',
            'Bronx, NY 10462',
            'United States']},
        'categories': [{'id': '4bf58dd8d48988d1e0931735',
          'name': 'Coffee Shop',
          'pluralName': 'Coffee Shops',
          'shortName': 'Coffee Shop',
          'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/coffees
hop_',
            'suffix': '.png'},
          'primary': True}],
        'photos': {'count': 0, 'groups': []}},
      'referralId': 'e-0-4c1c5630e9c4ef3b4ccd45aa-0'},
    {'reasons': {'count': 0,
      'items': [{'summary': 'This spot is popular',
        'type': 'general',
        'reasonName': 'globalInteractionReason'}]}],
      'venue': {'id': '4be7f0b988ed2d7f0038cb1d',
        'name': 'New Morris Deli',
        'location': {'address': '744 Morris Park Ave',
          'crossStreet': 'Holland Ave',
          'lat': 40.84652921587026,
          'lng': -73.86387374550078,
          'labeledLatLngs': [{'label': 'display',
            'lat': 40.84652921587026,
            'lng': -73.86387374550078},
          {'label': 'entrance', 'lat': 40.845438, 'lng': -73.864408}],
          'distance': 207,
          'postalCode': '10462',
          'cc': 'US',

```

```
In [32]: # function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

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

```
In [33]: venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.
location.lng']
nearby_venues = nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

Out[33]:

|   | name                            | categories         | lat       | lng        |
|---|---------------------------------|--------------------|-----------|------------|
| 0 | Conti's Pastry Shoppe           | Coffee Shop        | 40.845906 | -73.862836 |
| 1 | New Morris Deli                 | Deli / Bodega      | 40.846529 | -73.863874 |
| 2 | Morris Park Pizza               | Pizza Place        | 40.844962 | -73.867606 |
| 3 | Primavera Pizzeria & Restaurant | Pizza Place        | 40.845761 | -73.863848 |
| 4 | F & J Pine Tavern               | Italian Restaurant | 40.848766 | -73.862242 |

```
In [34]: print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))

30 venues were returned by Foursquare.
```

```
In [40]: def getNearbyVenues(names, latitudes, longitudes, radius=500, limit=100):

    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=KKVBXH3EBDJEP
        SYEKPA5PECJO1F3DXCDEAJ0XYTEM2PWXNZ&client_secret=MVX4FXGCHORFJPWCD2DFEALOT3XYJZOWT
        3QPILSWZZBU12AU&ll=40.8448,-73.8648&v=20181206&radius=500&limit=100'.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 ve
nue_list])
    nearby_venues.columns = ['Neighborhood',
                            'Neighborhood Latitude',
                            'Neighborhood Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)
```

```
In [41]: bronx_venues = getNearbyVenues(names=bronx_data['Neighborhood'],
                                         latitudes=bronx_data['Latitude'],
                                         longitudes=bronx_data['Longitude']
                                         )
```

Wakefield  
Co-op City  
Eastchester  
Fieldston  
Riverdale  
Kingsbridge  
Woodlawn  
Norwood  
Williamsbridge  
Baychester  
Pelham Parkway  
City Island  
Bedford Park  
University Heights  
Morris Heights  
Fordham  
East Tremont  
West Farms  
High Bridge  
Melrose  
Mott Haven  
Port Morris  
Longwood  
Hunts Point  
Morrisania  
Soundview  
Clason Point  
Throgs Neck  
Country Club  
Parkchester  
Westchester Square  
Van Nest  
Morris Park  
Belmont  
Spuyten Duyvil  
North Riverdale  
Pelham Bay  
Schuylerville  
Edgewater Park  
Castle Hill  
Olinville  
Pelham Gardens  
Concourse  
Unionport  
Edenwald  
Claremont Village  
Concourse Village  
Mount Eden  
Mount Hope  
Bronxdale  
Allerton  
Kingsbridge Heights

```
In [42]: print(bronx_venues.shape)
bronx_venues.head()
```

```
(1768, 7)
```

```
Out[42]:
```

|   | Neighborhood | Neighborhood<br>Latitude | Neighborhood<br>Longitude | Venue                                 | Venue<br>Latitude | Venue<br>Longitude | Venue<br>Category            |
|---|--------------|--------------------------|---------------------------|---------------------------------------|-------------------|--------------------|------------------------------|
| 0 | Wakefield    | 40.894705                | -73.847201                | Conti's Pastry<br>Shoppe              | 40.845906         | -73.862836         | Coffee Shop                  |
| 1 | Wakefield    | 40.894705                | -73.847201                | New Morris Deli                       | 40.846529         | -73.863874         | Deli / Bodega                |
| 2 | Wakefield    | 40.894705                | -73.847201                | Morris Park<br>Pizza                  | 40.844962         | -73.867606         | Pizza Place                  |
| 3 | Wakefield    | 40.894705                | -73.847201                | Primavera<br>Pizzeria &<br>Restaurant | 40.845761         | -73.863848         | Pizza Place                  |
| 4 | Wakefield    | 40.894705                | -73.847201                | Arth Aljanathain                      | 40.847338         | -73.866632         | Middle Eastern<br>Restaurant |

```
In [43]: bronx_venues.groupby('Neighborhood').count()
```

Out [43]:

|                        | Neighborhood<br>Latitude | Neighborhood<br>Longitude | Venue | Venue<br>Latitude | Venue<br>Longitude | Venue<br>Category |
|------------------------|--------------------------|---------------------------|-------|-------------------|--------------------|-------------------|
| Neighborhood           |                          |                           |       |                   |                    |                   |
| Allerton               | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Baychester             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Bedford Park           | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Belmont                | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Bronxdale              | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Castle Hill            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| City Island            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Claremont Village      | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Clason Point           | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Co-op City             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Concourse              | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Concourse<br>Village   | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Country Club           | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| East Tremont           | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Eastchester            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Edenwald               | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Edgewater Park         | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Fieldston              | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Fordham                | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| High Bridge            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Hunts Point            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Kingsbridge            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Kingsbridge<br>Heights | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Longwood               | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Melrose                | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Morris Heights         | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Morris Park            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Morrisania             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Mott Haven             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Mount Eden             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Mount Hope             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| North Riverdale        | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Norwood                | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Olinville              | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Parkchester            | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Pelham Bay             | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Pelham Gardens         | 34                       | 34                        | 34    | 34                | 34                 | 34                |
| Pelham Parkway         | 34                       | 34                        | 34    | 34                | 34                 | 34                |



```
In [45]: print('There are {} uniques categories.'.format(len(bronx_venues['Venue Category'].unique())))
```

There are 22 uniques categories.

```
In [46]: # one hot encoding
bronx_onehot = pd.get_dummies(bronx_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
bronx_onehot['Neighborhood'] = bronx_venues['Neighborhood']

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

bronx_onehot.head()
```

Out[46]:

|   | Neighborhood | BBQ Joint | Bakery | Bus Station | Café | Chinese Restaurant | Chocolate Shop | Coffee Shop | Cosmetics Shop | Deli / Bodega | Diner | Dis |
|---|--------------|-----------|--------|-------------|------|--------------------|----------------|-------------|----------------|---------------|-------|-----|
| 0 | Wakefield    | 0         | 0      | 0           | 0    | 0                  | 0              | 1           | 0              | 0             | 0     |     |
| 1 | Wakefield    | 0         | 0      | 0           | 0    | 0                  | 0              | 0           | 0              | 1             | 0     |     |
| 2 | Wakefield    | 0         | 0      | 0           | 0    | 0                  | 0              | 0           | 0              | 0             | 0     |     |
| 3 | Wakefield    | 0         | 0      | 0           | 0    | 0                  | 0              | 0           | 0              | 0             | 0     |     |
| 4 | Wakefield    | 0         | 0      | 0           | 0    | 0                  | 0              | 0           | 0              | 0             | 0     |     |

```
In [48]: bronx_onehot.shape
```

Out[48]: (1768, 23)

```
In [49]: bronx_grouped = bronx_onehot.groupby('Neighborhood').mean().reset_index()  
         bronx_grouped
```

Out [49]:

|    | Neighborhood           | BBQ<br>Joint | Bakery   | Bus<br>Station | Café     | Chinese<br>Restaurant | Chocolate<br>Shop | Coffee<br>Shop | Cosmetics<br>Shop | De<br>Bode |
|----|------------------------|--------------|----------|----------------|----------|-----------------------|-------------------|----------------|-------------------|------------|
| 0  | Allerton               | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 1  | Baychester             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 2  | Bedford Park           | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 3  | Belmont                | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 4  | Bronxdale              | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 5  | Castle Hill            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 6  | City Island            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 7  | Claremont<br>Village   | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 8  | Clason Point           | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 9  | Co-op City             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 10 | Concourse              | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 11 | Concourse<br>Village   | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 12 | Country Club           | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 13 | East Tremont           | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 14 | Eastchester            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 15 | Edenwald               | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 16 | Edgewater<br>Park      | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 17 | Fieldston              | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 18 | Fordham                | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 19 | High Bridge            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 20 | Hunts Point            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 21 | Kingsbridge            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 22 | Kingsbridge<br>Heights | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 23 | Longwood               | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 24 | Melrose                | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 25 | Morris Heights         | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 26 | Morris Park            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 27 | Morrisania             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 28 | Mott Haven             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 29 | Mount Eden             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 30 | Mount Hope             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 31 | North<br>Riverdale     | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 32 | Norwood                | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 33 | Olinville              | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 34 | Parkchester            | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| 35 | Pelham Bay             | 0.029412     | 0.029412 | 0.058824       | 0.029412 | 0.088235              | 0.029412          | 0.029412       | 0.029412          | 0.0882     |
| -- | Pelham                 | -----        | -----    | -----          | -----    | -----                 | -----             | -----          | -----             | -----      |

```
In [50]: num_top_venues = 5

for hood in bronx_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = bronx_grouped[bronx_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue', 'freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

## ----Allerton----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----Baychester----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----Bedford Park----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----Belmont----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----Bronxdale----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----Castle Hill----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----City Island----

|   | venue              | freq |
|---|--------------------|------|
| 0 | Pizza Place        | 0.15 |
| 1 | Chinese Restaurant | 0.09 |
| 2 | Deli / Bodega      | 0.09 |
| 3 | Bus Station        | 0.06 |
| 4 | Spanish Restaurant | 0.06 |

## ----Claremont Village----

```
In [51]: def return_most_common_venues(row, num_top_venues):
row_categories = row.iloc[1:]
row_categories_sorted = row_categories.sort_values(ascending=False)

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

```
In [53]: 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'] = bronx_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

Out[53]:

|   | Neighborhood | 1st Most<br>Common<br>Venue | 2nd<br>Most<br>Common<br>Venue | 3rd Most<br>Common<br>Venue | 4th Most<br>Common<br>Venue | 5th Most<br>Common<br>Venue | 6th Most<br>Common<br>Venue | 7th Most<br>Common<br>Venue | 8th Most<br>Common<br>Venue | 9th Most<br>Common<br>Venue |
|---|--------------|-----------------------------|--------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| 0 | Allerton     | Pizza Place                 | Deli / Bodega                  | Chinese Restaurant          | Spanish Restaurant          | Restaurant                  | Bus Station                 | Italian Restaurant          | Video Store                 | Ba                          |
| 1 | Baychester   | Pizza Place                 | Deli / Bodega                  | Chinese Restaurant          | Spanish Restaurant          | Restaurant                  | Bus Station                 | Italian Restaurant          | Video Store                 | Ba                          |
| 2 | Bedford Park | Pizza Place                 | Deli / Bodega                  | Chinese Restaurant          | Spanish Restaurant          | Restaurant                  | Bus Station                 | Italian Restaurant          | Video Store                 | Ba                          |
| 3 | Belmont      | Pizza Place                 | Deli / Bodega                  | Chinese Restaurant          | Spanish Restaurant          | Restaurant                  | Bus Station                 | Italian Restaurant          | Video Store                 | Ba                          |
| 4 | Bronxdale    | Pizza Place                 | Deli / Bodega                  | Chinese Restaurant          | Spanish Restaurant          | Restaurant                  | Bus Station                 | Italian Restaurant          | Video Store                 | Ba                          |

### Step 3: Clustering Neighborhoods

Now that we have the top venues from Foursquare we can use this information to cluster neighborhoods in Bronx, New York.

```
In [63]: # set number of clusters
kclusters = 3

bronx_grouped_clustering = bronx_grouped.drop('Neighborhood', 1)

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

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

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/cluster/k_means_.p
y:971: ConvergenceWarning: Number of distinct clusters (1) found smaller than n_
clusters (3). Possibly due to duplicate points in X.
    return_n_iter=True)
```

```
Out[63]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int32)
```

```
In [64]: # add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

bronx_merged = bronx_data

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

bronx_merged.head() # check the last columns!
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-64-5557b8a8c730> in <module>
      1 # add clustering labels
----> 2 neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
      3
      4 bronx_merged = bronx_data
      5

/opt/conda/envs/Python36/lib/python3.6/site-packages/pandas/core/frame.py in ins
ert(self, loc, column, value, allow_duplicates)
    3471     value = self._sanitize_column(column, value, broadcast=False)
    3472     self._data.insert(loc, column, value,
-> 3473                     allow_duplicates=allow_duplicates)
    3474
    3475     def assign(self, **kwargs):

/opt/conda/envs/Python36/lib/python3.6/site-packages/pandas/core/internals/manag
ers.py in insert(self, loc, item, value, allow_duplicates)
    1147     if not allow_duplicates and item in self.items:
    1148         # Should this be a different kind of error??
-> 1149         raise ValueError('cannot insert {}, already exists'.format(i
tem))
    1150
    1151     if not isinstance(loc, int):

ValueError: cannot insert Cluster Labels, already exists
```

```

In [65]: # create map
map_clusters = folium.Map(location=[40.8448, -73.8648], zoom_start=11)

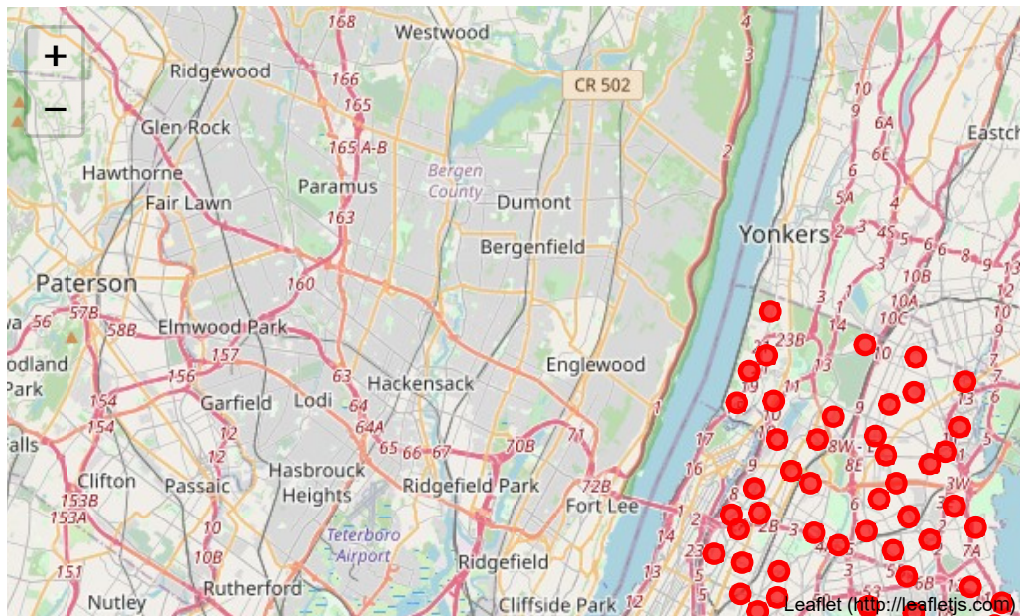
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(bronx_merged['Latitude'], bronx_merged['Longitude'],
    bronx_merged['Neighborhood'], bronx_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 [65]:



## Step 4: Analyze Clusters

Now that neighborhoods in Bronx, New York have been clustered you can analyze the clusters to identify the most common venues in the city.



```
In [58]: bronx_merged.loc[bronx_merged['Cluster Labels'] == 0, bronx_merged.columns[[1] + list(range(5, bronx_merged.shape[1]))]]
```

Out [58]:

|    | 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 | 9th Con V |
|----|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
| 0  | Wakefield          | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 1  | Co-op City         | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 2  | Eastchester        | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 3  | Fieldston          | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 4  | Riverdale          | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 5  | Kingsbridge        | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 6  | Woodlawn           | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 7  | Norwood            | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 8  | Williamsbridge     | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 9  | Baychester         | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 10 | Pelham Parkway     | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 11 | City Island        | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 12 | Bedford Park       | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 13 | University Heights | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 14 | Morris Heights     | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 15 | Fordham            | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 16 | East Tremont       | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 17 | West Farms         | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 18 | High Bridge        | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 19 | Melrose            | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 20 | Mott Haven         | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 21 | Port Morris        | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 22 | Longwood           | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 23 | Hunts Point        | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |
| 24 | Morrisania         | Pizza Place           | Deli / Bodega         | Chinese Restaurant    | Spanish Restaurant    | Restaurant            | Bus Station           | Italian Restaurant    | Video Store           | B         |

```
In [59]: bronx_merged.loc[bronx_merged['Cluster Labels'] == 1, bronx_merged.columns[[1] + list(range(5, bronx_merged.shape[1]))]]
```

Out[59]:

| 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 | 9th Most Common Venue |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|

```
In [60]: bronx_merged.loc[bronx_merged['Cluster Labels'] == 2, bronx_merged.columns[[1] + list(range(5, bronx_merged.shape[1]))]]
```

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 | 9th Most Common Venue |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|

```
In [61]: bronx_merged.loc[bronx_merged['Cluster Labels'] == 3, bronx_merged.columns[[1] + list(range(5, bronx_merged.shape[1]))]]
```

Out[61]:

| 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 | 9th Most Common Venue |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|

```
In [62]: bronx_merged.loc[bronx_merged['Cluster Labels'] == 4, bronx_merged.columns[[1] + list(range(5, bronx_merged.shape[1]))]]
```

Out[62]:

| 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 | 9th Most Common Venue |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|

## Results/Discussion:

The top three results that were provided for the city of Bronx, NY, were pizza place, delis, and chinese restaraunts. Outcome is dependent of the type of business venture, whether it be opening up a competitive food establishment to compete in the top three results or to establish a new food establishment that may not have a strong presense in the borough or neighborhood sought out to open in.

## Conclusion:

In conclusion, any sort of business venture comes with risks, in the mind set of deciding an area to open an establishment, whether it be new, existing and opening more loacations, or a competeing variety in an area with a dense culture of fewer varities, it is good to look at the overall demographics of the area. These methods are developed and used in order to process a lot of information in a smaller amount of time than it would take to read all of the documents. And having the ability to call out to data providing websites to provide locations and types of venues, but also have the ability to collect trending data as well, can help lower some of the risk of deciding to open in certain neighborhoods by providing a lot of information at once.

In [ ]: