Applied Data Science Capstone Project

1. INTRODUCTION

This is the capstone project for IBM Data Science Professional Certificate. In this one, I am creating a hypothetical scenario that there might not be enough Vietnamese restaurants in New York area. Therefore, it might be a great chance for an entrepreneur who is now looking for an opportunity to start a business in these neighborhoods. This entrepreneur might be thinking of opening his business in areas where Vietnamese community reside. With that purpose in mind, finding an appropriate location to open such a restaurant is one of the most important decisions for this entrepreneur and I am designing this project to help him find the most suitable location.

2. BUSINESS PROBLEM

The objective of this capstone project is to find the most suitable location for the entrepreneur to open a new Vietnamese restaurant in New York City, NY. By using machine learning methods and tools along with appropriate algorithms such as clustering, the goal of this project is to provide solutions to answer the business question: Where should an entrepreneur consider opening an Vietnamese restaurant in New York City?

3. TARGET AUDIENCE

Entrepreneurs those who are looking for locations to open an Vietnamese restaurant around New York.

4. DATA

To solve this problem, we will need below data:

- List of neighborhoods in New York City, NY
- Latitude and Longitude of these neighborhoods
- Venue data related to Vietnamese restaurants. This will help us find the neighborhoods that are more suitable to open an Vietnamese restaurant.

5. PROCESSING THE DATA

- Collecting New York neighborhoods via the file newyork_data.json
- Getting Latitude and Longitude data of these neighborhoods via Geocoder package.
- Using Foursquare API to get venue data related to these neighborhoods.

In [46]:

```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.
```

Out[46]:

In [2]:

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

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

import wget

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this lir
from geopy.geocoders import Nominatim # convert an address int

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

# 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
import folium # map rendering library

print('Libraries imported.')
```

```
usage: conda-script.py [-h] [-V] command ...
conda-script.py: error: unrecognized arguments:
# uncomment this line if you haven't completed
the Foursquare API lab

Libraries imported.
```

Download and Explore Dataset

Neighborhood has a total of 5 boroughs and 306 neighborhoods. In order to segement the neighborhoods and explore them, we will essentially need a dataset that contains the 5 boroughs and the neighborhoods that exist in each borough as well as the the latitude and logitude coordinates of each neighborhood.

Load and explore the data

In [3]:

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

Take a quick look at the 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, 40.8
9470517661]},
   '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,
```


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

In [5]:

```
neighborhoods_data = newyork_data['features']
```

Take a look at the first item in this list.

In [6]:

```
neighborhoods_data[0]
```

Out[6]:


```
{'type': 'Feature',
 'id': 'nyu_2451_34572.1',
 'geometry': {'type': 'Point',
  'coordinates': [-73.84720052054902, 40.894705
17661]},
 'geometry_name': 'geom',
 'properties': {'name': 'Wakefield',
  'stacked': 1,
  'annoline1': 'Wakefield',
  'annoline2': None,
  'annoline3': None,
  'annoangle': 0.0,
  'borough': 'Bronx',
  'bbox': [-73.84720052054902,
   40.89470517661,
   -73.84720052054902,
   40.89470517661]}}
```


Tranform the data into a *pandas* dataframe

In [47]:

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

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

Out[47]:

Borough	Neighborhood	Latitude	Longitude

Then let's loop through the data and fill the dataframe one row at a time.

In [48]:

```
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': neig
                                          'Latitude': neighbor
                                          'Longitude': neighbo
```

In [49]:

```
neighborhoods.head()
```

Out[49]:

	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

The dataset has all 5 boroughs and 306 neighborhoods.

In [50]:

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

```
The dataframe has 5 boroughs and 306 neighborho
ods.
```

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

In order to define an instance of the geocoder, we need to define a user_agent, named *ny_explorer*.

In [51]:

```
address = 'New York City, NY'

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

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

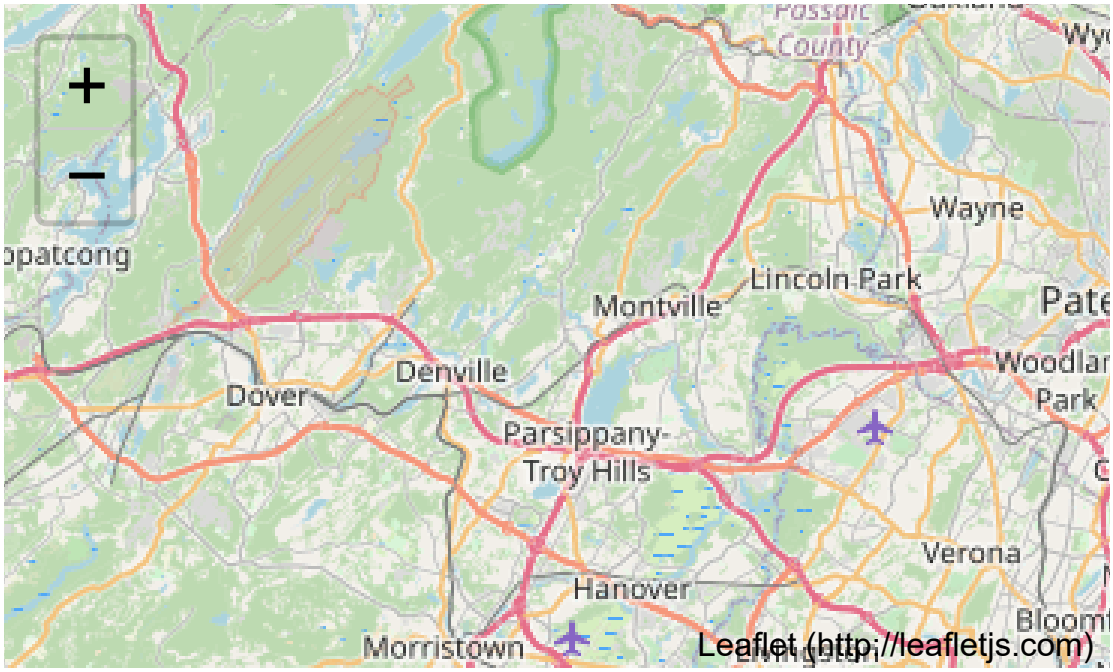
In [52]:

```
# create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[latitude, longitude], zoom_

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

map_newyork
```

Out[52]:



Simplify the above map and segment and cluster only the neighborhoods in Manhattan since this is the most bustling place in the city.

In [54]:

```
manhattan_data = neighborhoods[neighborhoods['Borough'] == 'Ma
manhattan_data
```

Out[54]:

	Borough	Neighborhood	Latitude	Longitude
0	Manhattan	Marble Hill	40.876551	-73.910660
1	Manhattan	Chinatown	40.715618	-73.994279
2	Manhattan	Washington Heights	40.851903	-73.936900
3	Manhattan	Inwood	40.867684	-73.921210
4	Manhattan	Hamilton Heights	40.823604	-73.949688
5	Manhattan	Manhattanville	40.816934	-73.957385
6	Manhattan	Central Harlem	40.815976	-73.943211
7	Manhattan	East Harlem	40.792249	-73.944182
8	Manhattan	Upper East Side	40.775639	-73.960508
9	Manhattan	Yorkville	40.775930	-73.947118
10	Manhattan	Lenox Hill	40.768113	-73.958860
11	Manhattan	Roosevelt Island	40.762160	-73.949168
12	Manhattan	Upper West Side	40.787658	-73.977059
13	Manhattan	Lincoln Square	40.773529	-73.985338
14	Manhattan	Clinton	40.759101	-73.996119
15	Manhattan	Midtown	40.754691	-73.981669
16	Manhattan	Murray Hill	40.748303	-73.978332
17	Manhattan	Chelsea	40.744035	-74.003116

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

Get the geographical coordinates of Manhattan.

In [55]:

```
address = 'Manhattan, NY'

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

```
The geograpical coordinate of Manhattan are 40.
7896239, -73.9598939.
```

As we did with all of New York City, let's visualizat Manhattan the neighborhoods in it.

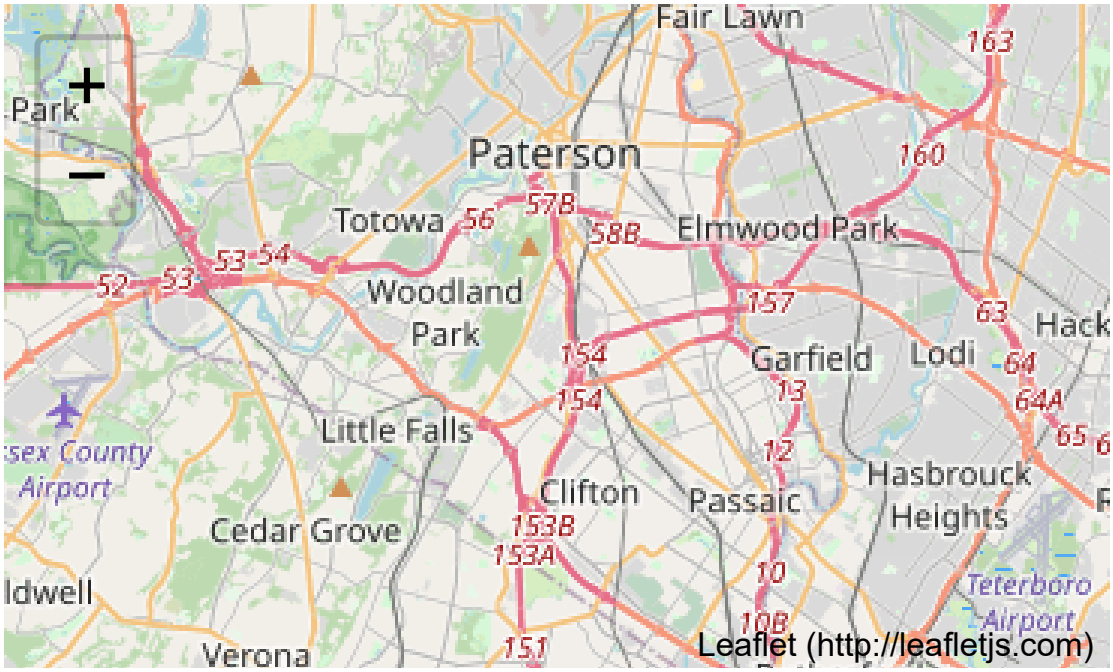
In [56]:

```
# create map of Manhattan using latitude and longitude values
map_manhattan = folium.Map(location=[latitude, longitude], zoc

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

map_manhattan
```

Out[56]:



Utilizing the Foursquare API to explore the neighborhoods and segment them.

Define Foursquare Credentials and Version

In [1]:

```
CLIENT_ID = 'my ID' # your Foursquare ID
CLIENT_SECRET = 'my secret' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

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

```
Your credentails:
CLIENT_ID: my ID
CLIENT_SECRET:my secret
```

Explore Neighborhoods in Manhattan

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

In [60]:

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

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&c
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

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

        # return only relevant information for each nearby ver
        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 resu

    nearby_venues = pd.DataFrame([item for venue_list in venue
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
```

```
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

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

In [61]:

```
# type your answer here

manhattan_venues = getNearbyVenues(names=manhattan_data['Neigh
                                   latitudes=manhattan_data['L
                                   longitudes=manhattan_data['
                                  )
```

```
Marble Hill
Chinatown
Washington Heights
Inwood
Hamilton Heights
Manhattanville
Central Harlem
East Harlem
Upper East Side
Yorkville
Lenox Hill
Roosevelt Island
Upper West Side
Lincoln Square
Clinton
Midtown
Murray Hill
Chelsea
Greenwich Village
```

Let's check the size of the resulting dataframe

In [63]:

```
print(manhattan_venues.shape)
manhattan_venues.head()
```

```
(3102, 7)
```

Out[63]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue
0	Marble Hill	40.876551	-73.91066	Arturo's
1	Marble Hill	40.876551	-73.91066	Bikram Yoga
2	Marble Hill	40.876551	-73.91066	Tibbett Diner
3	Marble Hill	40.876551	-73.91066	Starbucks
4	Marble Hill	40.876551	-73.91066	Dunkin'

Check how many venues were returned for each neighborhood

In [65]:

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

Out[65]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venu Latitu
Battery Park City	64	64	64	6
Carnegie Hill	84	84	84	8
Central Harlem	46	46	46	4
Chelsea	100	100	100	10
Chinatown	100	100	100	10
Civic Center	99	99	99	9
Clinton	100	100	100	10
East Harlem	42	42	42	4
East Village	100	100	100	10
Financial District	100	100	100	10
Flatiron	100	100	100	10
Gramercy	84	84	84	8
Greenwich Village	100	100	100	10
Hamilton Heights	60	60	60	6
Hudson Yards	54	54	54	5

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venu Latitu
Inwood	57	57	57	5
Lenox Hill	100	100	100	1(
Lincoln Square	98	98	98	9
Little Italy	100	100	100	1(
Lower East Side	49	49	49	4
Manhattan Valley	40	40	40	4
Manhattanville	44	44	44	4
Marble Hill	26	26	26	2
Midtown	100	100	100	1(
Midtown South	100	100	100	1(
Morningside Heights	41	41	41	4
Murray Hill	77	77	77	7
Noho	100	100	100	1(
Roosevelt Island	27	27	27	2
Soho	89	89	89	8
Stuyvesant Town	17	17	17	1
Sutton Place	100	100	100	1(
Tribeca	76	76	76	7

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venu Latitu
Tudor City	74	74	74	7
Turtle Bay	100	100	100	10
Upper East Side	88	88	88	8
Upper West Side	82	82	82	8
Washington Heights	84	84	84	8
West Village	100	100	100	10
Yorkville	100	100	100	10

In [66]:

```
print('There are {} uniques categories.'.format(len(manhattan_
```

```
There are 329 uniques categories.
```

Explore how many asian restaurants in the area.

In [108]:

```
manhattan_venues[manhattan_venues['Venue Category']=='Vietname
```

Out[108]:

```
Neighborhood              18
Neighborhood Latitude     18
Neighborhood Longitude    18
Venue                     18
Venue Latitude            18
Venue Longitude           18
Venue Category            18
dtype: int64
```

Examine Each Neighborhood

In [68]:

```
# one hot encoding
manhattan_onehot = pd.get_dummies(manhattan_venues[['Venue Cat

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

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

manhattan_onehot.head(10)
```

Out[68]:

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	R
0	Marble Hill	0	0	0	
1	Marble Hill	0	0	0	
2	Marble Hill	0	0	0	
3	Marble Hill	0	0	0	
4	Marble Hill	0	0	0	
5	Marble Hill	0	0	0	
6	Marble Hill	0	0	0	
7	Marble Hill	0	0	0	
8	Marble Hill	0	0	0	
9	Marble Hill	0	0	0	

In [32]:

```
manhattan_onehot.shape
```

Out[32]:

```
(3102, 330)
```

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

In [73]:

```
manhattan_grouped = manhattan_onehot.groupby('Neighborhood').m
manhattan_grouped
```

Out[73]:

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant
0	Battery Park City	0.000000	0.00	0.00
1	Carnegie Hill	0.000000	0.00	0.00
2	Central Harlem	0.000000	0.00	0.00
3	Chelsea	0.000000	0.00	0.00
4	Chinatown	0.000000	0.00	0.00
5	Civic Center	0.000000	0.00	0.00
6	Clinton	0.000000	0.00	0.00
7	East Harlem	0.000000	0.00	0.00
8	East Village	0.000000	0.00	0.00
9	Financial District	0.000000	0.00	0.00
10	Flatiron	0.000000	0.00	0.00
11	Gramercy	0.000000	0.00	0.00
12	Greenwich Village	0.000000	0.00	0.00
13	Hamilton Heights	0.000000	0.00	0.00

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant
14	Hudson Yards	0.000000	0.00	0.00
15	Inwood	0.000000	0.00	0.00
16	Lenox Hill	0.000000	0.00	0.01
17	Lincoln Square	0.000000	0.00	0.00
18	Little Italy	0.000000	0.00	0.00
19	Lower East Side	0.000000	0.00	0.00
20	Manhattan Valley	0.000000	0.00	0.00
21	Manhattanville	0.000000	0.00	0.00
22	Marble Hill	0.000000	0.00	0.00
23	Midtown	0.000000	0.00	0.00
24	Midtown South	0.000000	0.00	0.00
25	Morningside Heights	0.000000	0.00	0.00
26	Murray Hill	0.000000	0.00	0.00
27	Noho	0.000000	0.00	0.00
28	Roosevelt Island	0.000000	0.00	0.00
29	Soho	0.000000	0.00	0.00
30	Stuyvesant Town	0.000000	0.00	0.00
31	Sutton Place	0.000000	0.01	0.00
32	Tribeca	0.000000	0.00	0.00

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant
33	Tudor City	0.000000	0.00	0.00
34	Turtle Bay	0.000000	0.00	0.00
35	Upper East Side	0.000000	0.00	0.00
36	Upper West Side	0.000000	0.00	0.00
37	Washington Heights	0.011905	0.00	0.00
38	West Village	0.000000	0.00	0.00
39	Yorkville	0.000000	0.00	0.00

Examine the new size

In [74]:

```
manhattan_grouped.shape
```

Out[74]:

```
(40, 330)
```

In [109]:

```
len(manhattan_grouped[manhattan_grouped["Vietnamese Restaurant
```

Out[109]:

11

In [111]:

```
manhattan_vietnam = manhattan_grouped[["Neighborhood","Vietnam
manhattan_vietnam.head()
```

Out[111]:

	Neighborhood	Vietnamese Restaurant
0	Battery Park City	0.00000
1	Carnegie Hill	0.02381
2	Central Harlem	0.00000
3	Chelsea	0.00000
4	Chinatown	0.03000

In [112]:

```
from sklearn.cluster import KMeans
clusters = 3

clustering = manhattan_vietnam.drop(["Neighborhood"], 1)

# run k-means clustering
kmeans = KMeans(n_clusters=clusters, random_state=1)
kmeans.fit_transform(clustering)

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

Out[112]:

```
array([0, 1, 0, 0, 1, 2, 0, 0, 1, 0, 0, 0, 1,
0, 0, 0, 0, 0, 2, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 2, 0, 0, 1])
```

In [127]:

```
merged = manhattan_vietnam.copy()

# add clustering labels
merged["Cluster Labels"] = kmeans.labels_
merged.head()
```

Out[127]:

	Neighborhood	Vietnamese Restaurant	Cluster Labels
0	Battery Park City	0.00000	0
1	Carnegie Hill	0.02381	1
2	Central Harlem	0.00000	0
3	Chelsea	0.00000	0
4	Chinatown	0.03000	1

In [128]:

```
merged = merged.join(manhattan_venues.set_index("Neighborhood"
print(merged.shape)
merged.head()
```

```
(3102, 9)
```

Out[128]:

	Neighborhood	Vietnamese Restaurant	Cluster Labels	Neighborhood Latitude	Neigl L
0	Battery Park City	0.0	0	40.711932	-7
0	Battery Park City	0.0	0	40.711932	-7
0	Battery Park City	0.0	0	40.711932	-7
0	Battery Park City	0.0	0	40.711932	-7
0	Battery Park City	0.0	0	40.711932	-7

In [132]:

```
merged[merged['Cluster Labels'] == 0]['Neighborhood'].unique()
```

Out[132]:

```
array(['Battery Park City', 'Central Harlem',
'Chelsea', 'Clinton',
       'East Harlem', 'Financial District', 'Fl
atiron', 'Gramercy',
       'Hamilton Heights', 'Hudson Yards', 'Inw
ood', 'Lenox Hill',
       'Lincoln Square', 'Manhattanville', 'Mar
ble Hill', 'Midtown',
       'Midtown South', 'Morningside Heights',
'Murray Hill', 'Noho',
       'Roosevelt Island', 'Soho', 'Stuyvesant
Town', 'Sutton Place',
       'Tribeca', 'Turtle Bay', 'Upper East Sid
e', 'Washington Heights',
       'West Village'], dtype=object)
```

In [115]:

```
merged['Venue Category'].unique()
```

Out[115]:

```
array(['Park', 'Cooking School', 'Food Cour
t', 'Plaza', 'Gym',
       'Shopping Mall', 'Memorial Site', 'Am
erican Restaurant',
       'Playground', 'BBQ Joint', 'Buildin
g', 'Scenic Lookout',
       'Auditorium', 'Sandwich Place', 'Burr
ito Place',
       'Monument / Landmark', 'Gourmet Sho
p', 'Garden', 'Coffee Shop',
       'Movie Theater', 'Hotel', 'Steakhous
e', 'Burger Joint', 'Pub',
       'Hotel Bar', 'Electronics Store', 'At
hletics & Sports',
       'Mexican Restaurant', 'Convenience St
ore', 'Beer Garden',
       'Clothing Store', 'Pet Store', 'Cosme
tics Shop', 'Bistro',
       'Historic Site', 'Wine Shop', 'Chines
e Restaurant',
       'Mediterranean Restaurant', 'Pizza Pl
ace', "Women's Store",
       'Boat or Ferry', 'Bookstore', 'Wine B
ar', 'Dance Studio',
       'Community Center', 'Gym / Fitness Ce
nter', 'Italian Restaurant',
       'Café', 'Karaoke Bar', 'Bagel Shop',
'Bar', 'Ramen Restaurant',
       'French Restaurant', 'Restaurant', 'S
ushi Restaurant',
       'Yoga Studio', 'Art Museum', 'Shippin
```

```
g Store', 'Food Truck',
       'Salon / Barbershop', 'Museum', 'Viet
namese Restaurant', 'Spa',
       'Bakery', 'Kosher Restaurant', 'Groce
ry Store', 'Cocktail Bar',
       'Indian Restaurant', 'Japanese Restau
rant', 'Hot Dog Joint',
       'Gift Shop', 'Argentinian Restauran
t', 'Martial Arts Dojo',
       'Fast Food Restaurant', 'Supermarke
t', 'New American Restaurant',
       "Doctor's Office", 'Salad Place', 'Cy
cle Studio', 'Beer Bar',
       'Library', 'Music Venue', 'African Re
staurant', 'Jazz Club',
       'Ethiopian Restaurant', 'Dessert Sho
p', 'Juice Bar', 'Boutique',
       'Cafeteria', 'Art Gallery', 'Fried Ch
icken Joint',
       'Caribbean Restaurant', 'Seafood Rest
aurant', 'Public Art',
       'Event Space', 'Southern / Soul Food
Restaurant', 'Market',
       'Metro Station', 'Cupcake Shop', 'Spe
akeasy', 'Theater',
       'Fish Market', 'Taco Place', 'Ice Cre
am Shop', 'Office', 'Butcher',
       'Health & Beauty Service', 'Middle Ea
stern Restaurant',
       'College Theater', 'Cheese Shop', 'Ph
otography Studio', 'Pool',
       'Nightclub', 'Flea Market', 'Indie Th
eater', 'Smoothie Shop',
       'Paella Restaurant', 'Club House', 'G
eneral Entertainment',
       'Jewelry Store', 'Shoe Store', 'Pharm
acy', 'Harbor / Marina',
       'Greek Restaurant', 'English Restaura
```

```
nt', 'Tea Room',
       'Hotpot Restaurant', 'Indie Movie The
ater', 'Roof Deck',
       'Spanish Restaurant', 'Noodle House',
'Bike Shop',
       'Asian Restaurant', 'Thai Restauran
t', 'Organic Grocery',
       'Bubble Tea Shop', 'Malay Restauran
t', 'Snack Place', 'Diner',
       'Paper / Office Supplies Store', 'Dim
Sum Restaurant',
       'Shanghai Restaurant', 'Optical Sho
p', 'Sports Club',
       'Dumpling Restaurant', 'Korean Restau
rant', 'Austrian Restaurant',
       'Vegetarian / Vegan Restaurant', 'Tai
wanese Restaurant',
       'Cha Chaan Teng', 'Molecular Gastrono
my Restaurant',
       'Falafel Restaurant', 'Cuban Restaura
nt', 'Antique Shop',
       'Furniture / Home Store', 'Australian
Restaurant', 'Nail Salon',
       'Baby Store', 'Medical Center', 'Stri
p Club', "Men's Store",
       'Laundry Service', 'Sporting Goods Sh
op', 'Boxing Gym', 'Lounge',
       'Farmers Market', 'Modern European Re
staurant', 'Liquor Store',
       'Wings Joint', 'Whisky Bar', 'Peruvia
n Restaurant', 'Comedy Club',
       'Pie Shop', 'Sports Bar', 'Health Foo
d Store',
       'Residential Building (Apartment / Co
ndo)', 'Music School',
       'Dog Run', 'Caucasian Restaurant', 'D
eli / Bodega', 'Poke Place',
       'Brazilian Restaurant', 'Performing A
```

```
rts Venue',
       'Latin American Restaurant', 'Street
Art', 'Donut Shop',
       'Gas Station', 'Beer Store', 'Pet Caf
é', 'Scandinavian Restaurant',
       'Moroccan Restaurant', 'Swiss Restaur
ant', 'Filipino Restaurant',
       'Soup Place', 'Garden Center', 'Arepa
Restaurant',
       'Arts & Crafts Store', 'Record Shop',
'Tapas Restaurant',
       'Hookah Bar', 'Mac & Cheese Joint',
'Pet Service',
       'Lingerie Store', 'Breakfast Spot',
'Pedestrian Plaza',
       'Rental Car Location', 'Miscellaneous
Shop', 'Russian Restaurant',
       'Toy / Game Store', 'Chocolate Shop',
'Kebab Restaurant',
       'Tech Startup', 'Outdoor Sculpture',
'Bridal Shop', 'Camera Store',
       'Bike Rental / Bike Share', 'Irish Pu
b', 'Arcade',
       'Thrift / Vintage Store', 'Comfort Fo
od Restaurant',
       'Pilates Studio', 'South Indian Resta
urant', 'Mattress Store',
       'Udon Restaurant', 'Hobby Shop', 'Leb
anese Restaurant',
       'Gaming Cafe', 'Creperie', 'Eastern E
uropean Restaurant',
       'Gastropub', 'School', 'Financial or
Legal Service', 'Smoke Shop',
       'Bank', 'Department Store', 'Frozen Y
ogurt Shop', 'Veterinarian',
       'History Museum', 'Empanada Restauran
t', 'Bus Station',
       'Moving Target', 'College Academic Bu
```

```
ilding', 'Czech Restaurant',
       'Afghan Restaurant', 'Non-Profit', 'T
urkish Restaurant',
       'Kitchen Supply Store', 'Opera Hous
e', 'Concert Hall',
       'College Arts Building', 'Circus', 'F
ountain', 'College Bookstore',
       'Climbing Gym', 'Candy Store', "Denti
st's Office", 'River',
       'Newsstand', 'Egyptian Restaurant',
'Rock Club', 'Massage Studio',
       'Flower Shop', 'Food Stand', 'Track',
'Tennis Court', 'Bus Stop',
       'Hostel', 'Hawaiian Restaurant', 'Sze
chuan Restaurant',
       'Japanese Curry Restaurant', 'Bike Tr
ail', 'Food & Drink Shop',
       'Tennis Stadium', 'Supplement Shop',
'Discount Store',
       'Video Game Store', 'Kids Store', 'Ta
ilor Shop',
       'Other Great Outdoors', 'Train Statio
n',
       'South American Restaurant', 'Golf Co
urse', 'Tattoo Parlor',
       'Persian Restaurant', 'Basketball Sta
dium', 'Food',
       'College Cafeteria', 'Jewish Restaura
nt', 'Cultural Center',
       'Resort', 'Himalayan Restaurant', 'Ou
tdoors & Recreation',
       'Waterfront', 'Soccer Field', 'Bus Li
ne', 'Bridge',
       'Baseball Field', 'Heliport', 'Adult
Boutique',
       'Cambodian Restaurant', 'Hardware Sto
re', 'Volleyball Court',
       'Gym Pool', 'Mini Golf', 'Skate Par
```

```
k', 'Basketball Court',
       'High School', 'Trail', 'Duty-free Sh
op',
       'Tourist Information Center', 'Sake B
ar', 'Soba Restaurant',
       'German Restaurant', 'Exhibit', 'Scul
pture Garden',
       'North Indian Restaurant', 'Tiki Ba
r', 'Israeli Restaurant',
       'Drugstore', 'Mobile Phone Shop', 'Ac
cessories Store', 'Rest Area',
       'Board Shop', 'Temple', 'Pier', 'Gay
Bar', 'Video Store',
       'Daycare', 'Gymnastics Gym'], dtype=o
bject)
```

In [125]:

```
map_clusters = folium.Map(location=[latitude, longitude], zoom
# set color scheme for the clusters

# add markers to the map
markers_colors={}
markers_colors[0] = 'red'
markers_colors[1] = 'blue'
markers_colors[2] = 'green'

for lat, lon, cluster in zip(merged['Neighborhood Latitude'],

    folium.features.CircleMarker(
        [lat, lon],
        radius=5,
        color =markers_colors[cluster],
        fill_color=markers_colors[cluster],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[125]:

Cluster 0

In [121]:

```
merged.loc[(merged['Cluster Labels'] == 0) & (merged['Venue Ca
```

Out[121]:

	Neighborhood	Vietnamese Restaurant	Cluster Labels	Neighborhood Latitude	Neighl Lo

Cluster 1

In [122]:

```
merged.loc[(merged['Cluster Labels'] == 1) & (merged['Venue Ca
```

Out[122]:

	Neighborhood	Vietnamese Restaurant	Cluster Labels	Neighborhood Latitude	Neig
33	Tudor City	0.027027	1	40.746917	
33	Tudor City	0.027027	1	40.746917	
39	Yorkville	0.020000	1	40.775930	
39	Yorkville	0.020000	1	40.775930	
8	East Village	0.020000	1	40.727847	
8	East Village	0.020000	1	40.727847	
4	Chinatown	0.030000	1	40.715618	
4	Chinatown	0.030000	1	40.715618	
1	Carnegie Hill	0.023810	1	40.782683	
1	Carnegie Hill	0.023810	1	40.782683	
4	Chinatown	0.030000	1	40.715618	

	Neighborhood	Vietnamese Restaurant	Cluster Labels	Neighborhood Latitude	Neig
20	Manhattan Valley	0.025000	1	40.797307	
19	Lower East Side	0.020408	1	40.717807	
12	Greenwich Village	0.020000	1	40.726933	
12	Greenwich Village	0.020000	1	40.726933	

Cluster 2

In [123]:

```
merged.loc[(merged['Cluster Labels'] == 2) & (merged['Venue Ca
```

Out[123]:

	Neighborhood	Vietnamese Restaurant	Cluster Labels	Neighborhood Latitude	Neig
18	Little Italy	0.010000	2	40.719324	
36	Upper West Side	0.012195	2	40.787658	
5	Civic Center	0.010101	2	40.715229	

Conclusion

We can clearly see that most of Vietnamese restaurants in Manhattan located at cluster 1 around Chinatown, Greenwich Village, East Village, Tudor City, Yorkville, Carnegie Hill. There are 3 Vietnamese restaurants located at cluster 2 around Little Italy, Upper West Side, Civic Center. So this cluster might be a good place to open a Vietnamese restaurant. However, there is no Vietnamese restaurant at cluster 0 around Financial District, Upper East Side, Manhattanville, Chelsea neighborhoods. Obviously, it is a good opportunity for an entrepreneur to open an authentic Vietnamese restaurant here since there is little or no competition in this area.

Thank you!