NIGHTLIFE SPOT LOCATIONS & VANUES DATA ANALYSYS OF NEWYORK

Tuyen Truong – Capstone Project

Introduction: Business Problem

In this project I will try to find an optimal location for a Nightlife Spot for someone who is looking for open it in Newyork City, US. Especially, this project is targeted to stakeholders who want to open a Bar in Newyork City in US.

I will focus on the boroughs of Manhattan (40 neighborhoods) and Brooklyn (70)

DATA

With the definition of my problem, these are the factors that will affect my decision:

- The distance of neighborhood to city center
- Number of and distance to Nightlife Spots in the neighborhood
- Number of and distance to Bars in the neighborhood (any type of bar)

Following data sources will be needed to extract/generate the required information:

- Centers of candidate areas will be generated algorithmically and approximate addresses of centers of those areas will be obtained using Google Maps API reverse geocoding
- Number of coffee shops and their type and location in every neighborhood will be obtained using Foursquare API
- Coordinate of Newyork center will be obtained using Google Maps API geocoding of well known NewYork location (New York City, NY, USA)

DATA

Let's now place all this into a Pandas dataframe.

```
import pandas as pd

df_locations = pd.DataFrame({'Address': addresses,
                             'Latitude': latitudes,
                             'Longitude': longitudes,
                             'X': xs,
                             'Y': ys,
                             'Distance from center': distances_from_center})

df_locations.head(10)
```

	Address	Latitude	Longitude	X	Y	Distance from center
0	1626 Hooper St, Brooklyn, NY 11211	40.701263	-73.962112	-5.824082e+06	9.863884e+06	5992.495307
1	556 Bedford Ave, Brooklyn, NY 11249	40.704797	-73.961984	-5.823482e+06	9.863884e+06	5840.376700
2	166 S 9th St, Brooklyn, NY 11211	40.708330	-73.961856	-5.822882e+06	9.863884e+06	5747.173218
3	158 S 3rd St, Brooklyn, NY 11211	40.711864	-73.961727	-5.822282e+06	9.863884e+06	5715.767665
4	184 Metropolitan Ave, Brooklyn, NY 11249	40.715399	-73.961599	-5.821682e+06	9.863884e+06	5747.173218
5	76 N 6th St, Brooklyn, NY 11249	40.718933	-73.961470	-5.821082e+06	9.863884e+06	5840.376700
6	10 Nassau Ave, Brooklyn, NY 11222	40.722468	-73.961342	-5.820482e+06	9.863884e+06	5992.495307
7	299 Park Ave, Brooklyn, NY 10017	40.696048	-73.966325	-5.824982e+06	9.864403e+06	5855.766389
8	22 Flushing Ave, Brooklyn, NY 11205	40.699581	-73.966197	-5.824382e+06	9.864403e+06	5604.462508
9	595 Kent Ave, Brooklyn, NY 11249	40.703114	-73.966069	-5.823782e+06	9.864403e+06	5408.326913

Methodology : Analysis

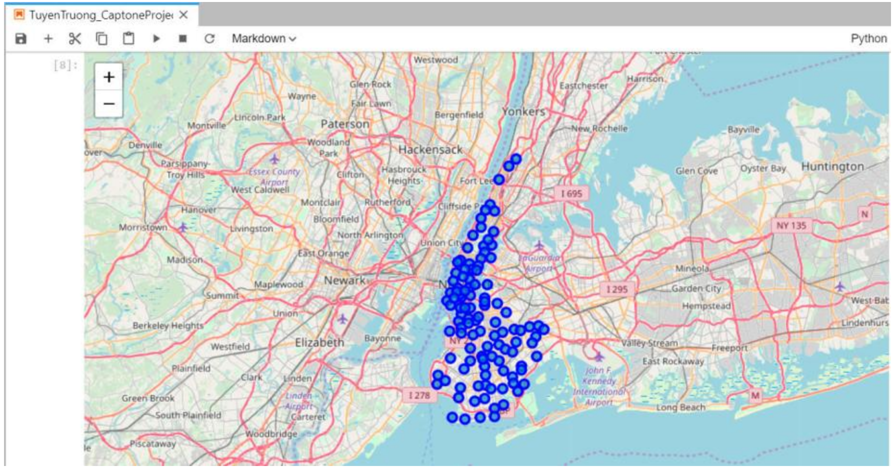
First let's count the number of neighborhoods in every area candidate:

```
neighborhood_count_manhattan = sum(neighborhoods['Borough'] == "Manhattan")
print(f"Manhattan has {neighborhood_count_manhattan} neighborhoods")

neighborhood_count_brooklyn = sum(neighborhoods['Borough'] == "Brooklyn")
print(f"Brooklyn has {neighborhood_count_brooklyn} neighborhoods")
```

```
Manhattan has 40 neighborhoods
Brooklyn has 70 neighborhoods
```

Visualization setup



- Get coordinates of Manhattan / Brooklyn

  The geograpical coordinate of Manhattan/Brooklyn are 40.72009534999994, -73.9547059.

- Populate raw Folium map

Request and compute Foursquare data

```
    nearby_venues = pd.DataFrame(venues_list)
    nearby_venues.columns = [
            'Borough',
            'Neighborhood',
            'Neighborhood Latitude',
            'Neighborhood Longitude',
            'Venue',
            'Venue Latitude',
            'Venue Longitude',
            'Venue Category']


    return(nearby_venues)

nightlifespot="4d4b7105d754a06376d81259"
bar_categories ="4bf58dd8d48988d116941735"
venues = getNearbyVenues(neighborhoods, f"{nightlifespot},{bar_categories}")
```

```
Manhattan // Marble Hill has 8 result
Brooklyn // Bay Ridge has 42 result
Brooklyn // Bensonhurst has 5 result
Brooklyn // Sunset Park has 8 result
Brooklyn // Greenpoint has 68 result
Brooklyn // Gravesend has 8 result
Brooklyn // Brighton Beach has 5 result
Brooklyn // Sheepshead Bay has 13 result
Brooklyn // Manhattan Terrace has 5 result
```

Request and compute Foursquare data

```
display(venues.head(10000))
```

	Borough	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Manhattan	Marble Hill	40.876551	-73.910660	Mr. McGoo's	40.879419	-73.904243	Bar
1	Manhattan	Marble Hill	40.876551	-73.910660	The Local	40.878553	-73.903462	Bar
2	Manhattan	Marble Hill	40.876551	-73.910660	Randal Og's	40.878544	-73.903435	Bar
3	Brooklyn	Bay Ridge	40.625801	-74.030621	The Kettle Black	40.622839	-74.031411	Bar
4	Brooklyn	Bay Ridge	40.625801	-74.030621	Bean Post Pub	40.628689	-74.022940	Bar
5	Brooklyn	Bay Ridge	40.625801	-74.030621	The Hideout	40.622400	-74.025502	Bar
6	Brooklyn	Bay Ridge	40.625801	-74.030621	Salty Dog Bar and Restaurant	40.631143	-74.027758	Bar
7	Brooklyn	Bay Ridge	40.625801	-74.030621	LoneStar Bar & Grill	40.620761	-74.026793	Bar

Methodology: Model Selection

I apply a KMeans(n=3): Clustering analysis. I tried various values (n=3..7) and noticed that there are really just one big group, namely...

```
from sklearn.cluster import KMeans

kclusters = 3

venues_filtered_grouped_clustering = venues_filtered_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(venues_filtered_grouped_clustering)   # run k-me
ans clustering
venues_filtered_grouped.insert(0, 'Cluster Labels', kmeans.labels_)   # add clustering labels

venues_filtered_merged = venues_filtered

# merge cluster label with list of venues
venues_filtered_merged = venues_filtered_merged.join(venues_filtered_grouped.set_index('Neighborhood'), on
='Neighborhood')
venues_filtered_merged.sort_values('Cluster Labels', inplace=True)
venues_filtered_merged.reset_index(inplace=True, drop=True)

# translate meaning of cluster labels
venues_filtered_grouped = venues_filtered_grouped.sort_values("Cluster Labels").reset_index(drop=True)
cluster_groups = venues_filtered_grouped.groupby('Cluster Labels').mean().reset_index()
display(cluster_groups)
display(venues_filtered_grouped)
```

```
/home/jupyterlab/conda/lib/python3.6/site-packages/sklearn/cluster/k_means_.py:971: ConvergenceWarning: Nu
mber of distinct clusters (1) found smaller than n_clusters (3). Possibly due to duplicate points in X.
  return_n_iter=True)
```

	Cluster Labels	Bar
0	0	1

	Cluster Labels	Neighborhood	Bar
0	0	Battery Park City	1
1	0	Park Slope	1
2	0	Ocean Hill	1
3	0	North Side	1

Results: Visualization & Conclusion

✓ The clustering shows a very clear pattern. We have all bars are kind of nightlife spots. So we just have one cluster.
✓ So we can suggest that the stakeholders can open any bar in any location in NewYork. They will have a successful business.