BATTLE OF NEIGHBORHOODS

New York City

By Yadan Tang

Problem Statement

- The Catalysis Society of Metropolitan New York (CSMNY), a local chapter of North American Catalysis Society(NACS), need to find a location to host their annual meeting in Manhattan, New York City (NYC) in 2020. The number of attendees for this event is estimated to ~1000 people and of various cultural background. The attendees are coming for professional exchange as well as sightseeing and leisure.
- The host needs a recommendation of a suitable neighborhood in Manhattan NYC with the capacity of hotels, and various options of restaurants and activities for such an event.

Outlines

- Data acquisition and process the Neighborhood data in Manhattan, New York City.
- Explore the venues of all the neighborhoods in Manhattan, New York City.
- Using K-mean Clustering to segment the neighborhoods
- Conclusions based on the K-mean Clustering results

Data acquisition and Processing

- Data sources: the json data that contains all the information of NYC neighborhoods is available online: https://cocl.us/new_york_dataset
- The neighborhood data is available under the list [‘features’], and it was converted into dataframe for further processing
- Create a dataframe contains the ‘Borough’, ‘Neighborhood’, ’Latitude’, ‘Longitude’ columns.

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

```
[7]: 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}, ignore_index=True)
```

- Double check that the dataframe contains 306 neighborhoods and 5 unique Boroughs, which is accurate.

Data acquisition and Processing con’t

- Use ‘groupby’ function and ‘Borough’ as the argument and count() to see how many neighborhoods are in each Borough. The result showed Manhattan has 40 neighborhoods.

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

[12]:

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

- Create a dataframe that contains all the neighborhood information of Manhattan.

```
[13]: mht_data=neighborhoods[neighborhoods['Borough']=='Manhattan'].reset_index(drop=True)
mht_data.head()
```

[13]:

	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

Connect to Foursquare API and explore the nearby venues

- Connect to Foursquare API using personal Foursquare ID and secrets
- Define a function to get all the nearby venues in Manhattan, NYC. Return only the relevant information of each nearby venue and save it into a dataframe called “nearby_venues” which includes the name and the coordinates of the neighborhoods and its nearby venues.
- Group the dataframe by venue category, the count show there are 331 unique venue categories.
- To create columns of venues for each neighborhood by one hot encoding and then group it by neighborhood and show the mean counts of each venue in each neighborhood.

Connect t0 Foursquare API and explore the nearby venues

- Show the top 10 venues in each Neighborhood

[34]:
```
import numpy as np
top_venue_number = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(top_venue_number):
    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'] = mht_grouped['Neighborhood']

for ind in np.arange(mht_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = topVenues(mht_grouped.iloc[ind, :], top_venue_number)

neighborhoods_venues_sorted
```

[34]:

	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	10th Most Common Venue
0	Battery Park City	Park	Hotel	Coffee Shop	Gym	Memorial Site	Playground	Gourmet Shop	Food Court	Mexican Restaurant	Shopping Mall

K-means Clustering: segment the neighborhood of Manhattan

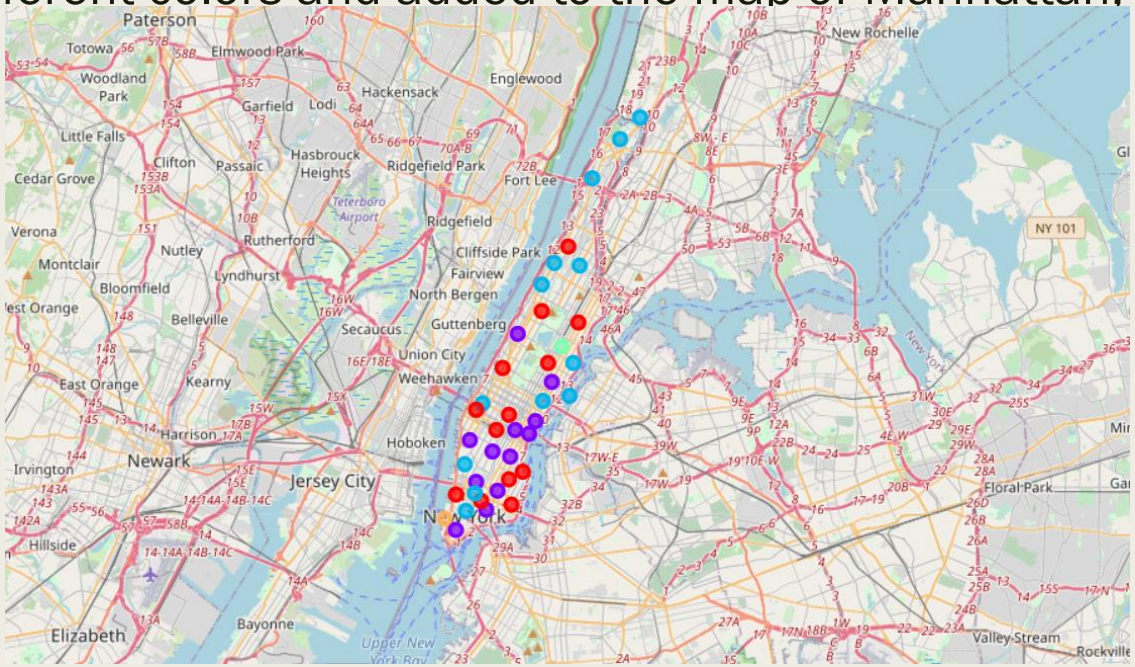
- The initial K-mean clustering is set to 5.

```
[42]: mht_merged=mht_data
mht_merged['Labels']=kMeans.labels_
mht_merged=mht_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')
mht_merged.head()
```

[42]:

	Borough	Neighborhood	Latitude	Longitude	Labels	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
0	Manhattan	Marble Hill	40.876551	-73.910660	2	Sandwich Place	Gym	Coffee Shop	Yoga Studio	Pharmacy	Supplement Shop	Steakhouse	Seafood Restaurant	Pizza Place
1	Manhattan	Chinatown	40.715618	-73.994279	1	Chinese Restaurant	Bakery	Cocktail Bar	Bubble Tea Shop	Coffee Shop	Optical Shop	Bar	American Restaurant	Spa

- Visualize the clustering result using folium and the clusters were classified and labelled in different colors and added to the map of Manhattan, NYC.



Explore the information in each cluster and find out the best options for the event

- Create dataframe for each cluster labels, showing one as an example:

```
[44]: cluster_0=mht_merged.loc[mht_merged['Labels']==0, mht_merged.columns[[1] + list(range(4, mht_merged.shape[1]))]]
      cluster_0.head()
```

[44]:

	Neighborhood	Labels	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	10th Most Common Venue
4	Hamilton Heights	0	Pizza Place	Deli / Bodega	Coffee Shop	Mexican Restaurant	Café	Yoga Studio	Sandwich Place	Sushi Restaurant	Bakery	Caribbean Restaurant
7	East Harlem	0	Bakery	Mexican Restaurant	Thai Restaurant	Sandwich Place	Latin American Restaurant	Deli / Bodega	Gas Station	Liquor Store	Steakhouse	Seafood Restaurant
8	Upper East Side	0	Italian Restaurant	Coffee Shop	Gym / Fitness Center	Bakery	French Restaurant	Spa	Yoga Studio	Juice Bar	American Restaurant	Hotel
13	Lincoln Square	0	Plaza	Café	Italian Restaurant	Gym / Fitness Center	Concert Hall	Theater	Performing Arts Venue	Gym	French Restaurant	Coffee Shop
15	Midtown	0	Hotel	Coffee Shop	Bakery	Theater	Pizza Place	Sushi Restaurant	Japanese Restaurant	Cuban Restaurant	Clothing Store	Cosmetics Shop

- Cluster_0 and Cluster_1 showed neighborhoods that has a good variety of restaurants/deli, gym/yoga studio, and hotels, plaza.
- Especially Midtown from Cluster_0 and Murray Hill from Cluster_1 are good options as Hotel is the most common venue for those Neighborhoods, and the nearby neighborhood has good combination of multicultural restaurants, and gym and plazas.

Conclusions

- Based on the exercise, we have successfully explore the neighborhood of Manhattan, NYC and the nearby venues of each neighborhood. K-mean clustering helps to identify the neighborhoods in Cluster_0 and 1 are excellent options for the event and especially Midtown and Murray Hill will have Hotels as the 1st common venue and still have a good combination of multicultural restaurants, deli/café, gyms and plaza.