Final Capstone Project

Introduction

This problem seems very interesting as we need to take care of multiple parameters into account for selecting any neighborhood region. As a person I would prefer the region where I would go is more familiar to me, or within a week it should be second homeplace to me. There are many primary things to consider here like, demographics of region like weather conditions, clock time difference, food and culture, some government data about how favored it was in past. But these things are just for so called secondary thing which is business idea.

Usually, citizens prefer places with moderate climate situation, having historical sites, city which is more similar to New York city (considering American Citizen), city having variety of restaurants like continental, sub-continental, language preferably English. So, for a guy living in New York city, if they want some outing so preferable locations might be Bangkok, Tokyo.

Some important points:

• People prefer language and culture most according to me.
• Worldwide reputation of the city also important, say ease of doing business ranking.
• How similar the location is when compared to their native place.
• Different tourists' attractions and sites to visit in that place.
• Budget.

Data Gathering

I used data of all the above-mentioned cities, and used various techniques as listed in project itself to explore venues at each place. Also, data cleaning and selection is performed to narrow down the search space and get more accurate result. So, I listed out proposed steps below:

-• I will be using Dataset which contain all the required geographical data about New York City.
-• To be more specific I am using 'Borough', 'Neighborhood', 'Latitude', and 'Longitude' kind of fields in dataset. The dataset is already gathered in week 3 of this course but need few fine tunings.
-• Geo-coordinates of districts will be obtained with the help of the geo-coder tool in the notebook.
-• Counting the occurrence of venues in each city and collected them all in one dataset so that we can see frequency of each place. Then we calculate the probability of each venue and make clustering to find the similar cities.

Getting Data Required for analysis

In [24]:
```
df=[]
cities = ['Bangkok','Tokyo','New York City']
for city in cities:
    address = city
    geolocator = Nominatim(user_agent="foursquare_agent")
    location = geolocator.geocode(address)
```

```
    latitude = location.latitude
    longitude = location.longitude
    print('For {}, The latitude is: {} and Longitude is: {}'.format(city, latitude,
    url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={
    print('The url for {}: is {}'.format(city,url))
    results = requests.get(url).json()
    results.keys()
    results['response'].keys()
    items = results['response']['groups'][0]['items']
    print('There are {} observations and {} columns for each item around {}'.format(
    df.append(pd.json_normalize(items))
```

```
For Bangkok, The latitude is: 13.7544238 and Longitude is: 100.4930399
The url for Bangkok: is https://api.foursquare.com/v2/venues/explore?client_id=H2NZQ
N05FD0V0DFDN1FO5VR2ZKCDKCKUJPSYEALGQJCLIUQJ&client_secret=CNTMAUZVWIPTZXHE3ZGEQ1AAFH
HRP5YY1GIWRPOS2EEIF1YS&ll=13.7544238,100.4930399&v=20180604&radius=500&limit=30
There are 26 observations and 21 columns for each item around Bangkok
For Tokyo, The latitude is: 35.6828387 and Longitude is: 139.7594549
The url for Tokyo: is https://api.foursquare.com/v2/venues/explore?client_id=H2NZQN0
5FD0V0DFDN1FO5VR2ZKCDKCKUJPSYEALGQJCLIUQJ&client_secret=CNTMAUZVWIPTZXHE3ZGEQ1AAFHHR
P5YY1GIWRPOS2EEIF1YS&ll=35.6828387,139.7594549&v=20180604&radius=500&limit=30
There are 30 observations and 22 columns for each item around Tokyo
For New York City, The latitude is: 40.7127281 and Longitude is: -74.0060152
The url for New York City: is https://api.foursquare.com/v2/venues/explore?client_id
=H2NZQN05FD0V0DFDN1FO5VR2ZKCDKCKUJPSYEALGQJCLIUQJ&client_secret=CNTMAUZVWIPTZXHE3ZGE
Q1AAFHHRP5YY1GIWRPOS2EEIF1YS&ll=40.7127281,-74.0060152&v=20180604&radius=500&limit=3
0
There are 30 observations and 28 columns for each item around New York City
```

Data Exploration

In [31]:

```
df_Bangkok.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   referralId                      26 non-null     object
 1   reasons.count                   26 non-null     int64
 2   reasons.items                   26 non-null     object
 3   venue.id                        26 non-null     object
 4   venue.name                      26 non-null     object
 5   venue.location.address          25 non-null     object
 6   venue.location.lat              26 non-null     float64
 7   venue.location.lng              26 non-null     float64
 8   venue.location.labeledLatLngs   25 non-null     object
 9   venue.location.distance         26 non-null     int64
 10  venue.location.postalCode       24 non-null     object
 11  venue.location.cc               26 non-null     object
 12  venue.location.city             25 non-null     object
 13  venue.location.state            26 non-null     object
 14  venue.location.country          26 non-null     object
 15  venue.location.formattedAddress 26 non-null     object
 16  venue.categories                26 non-null     object
 17  venue.photos.count              26 non-null     int64
 18  venue.photos.groups             26 non-null     object
 19  venue.location.crossStreet      13 non-null     object
 20  venue.location.neighborhood     7 non-null      object
dtypes: float64(2), int64(3), object(16)
memory usage: 4.4+ KB
```

Tokyo Dataset Structure

In [34]:

```
df_tokyo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   referralId                       30 non-null     object
 1   reasons.count                    30 non-null     int64
 2   reasons.items                    30 non-null     object
 3   venue.id                         30 non-null     object
 4   venue.name                       30 non-null     object
 5   venue.location.address           30 non-null     object
 6   venue.location.crossStreet       23 non-null     object
 7   venue.location.lat               30 non-null     float64
 8   venue.location.lng               30 non-null     float64
 9   venue.location.labeledLatLngs    30 non-null     object
 10  venue.location.distance          30 non-null     int64
 11  venue.location.postalCode        27 non-null     object
 12  venue.location.cc                30 non-null     object
 13  venue.location.city              30 non-null     object
 14  venue.location.state             30 non-null     object
 15  venue.location.country           30 non-null     object
 16  venue.location.formattedAddress  30 non-null     object
 17  venue.categories                 30 non-null     object
 18  venue.photos.count               30 non-null     int64
 19  venue.photos.groups              30 non-null     object
 20  venue.location.neighborhood      2 non-null      object
 21  venue.venuePage.id               1 non-null      object
dtypes: float64(2), int64(3), object(17)
memory usage: 5.3+ KB
```

New York Dataset Structure

In [37]:

```
df_New_York.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 28 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   referralId                            30 non-null     object
 1   reasons.count                         30 non-null     int64
 2   reasons.items                         30 non-null     object
 3   venue.id                              30 non-null     object
 4   venue.name                            30 non-null     object
 5   venue.location.address                30 non-null     object
 6   venue.location.lat                    30 non-null     float64
 7   venue.location.lng                    30 non-null     float64
 8   venue.location.labeledLatLngs         29 non-null     object
 9   venue.location.distance               30 non-null     int64
 10  venue.location.postalCode             30 non-null     object
 11  venue.location.cc                     30 non-null     object
 12  venue.location.neighborhood           3 non-null      object
 13  venue.location.city                   30 non-null     object
 14  venue.location.state                  30 non-null     object
 15  venue.location.country                30 non-null     object
 16  venue.location.formattedAddress       30 non-null     object
 17  venue.categories                      30 non-null     object
 18  venue.photos.count                    30 non-null     int64
 19  venue.photos.groups                   30 non-null     object
 20  venue.location.crossStreet            21 non-null     object
 21  venue.delivery.id                     12 non-null     object
 22  venue.delivery.url                    12 non-null     object
 23  venue.delivery.provider.name          12 non-null     object
 24  venue.delivery.provider.icon.prefix   12 non-null     object
 25  venue.delivery.provider.icon.sizes    12 non-null     object
 26  venue.delivery.provider.icon.name     12 non-null     object
 27  venue.venuePage.id                    9 non-null      object
```

```
dtypes: float64(2), int64(3), object(23)
memory usage: 6.7+ KB
```

Data Cleaning, Preprocessing and Feature Selection

Intersecting dataset

In [39]:
```
intersect_cols = list(set.intersection(*(set(city.columns) for city in dataset)))
```

Selecting name, categories, latitude and longitude from above

In [1]:
```
filtered_dataset = []
columns = ['venue.name', 'venue.categories', 'venue.location.lng','venue.location.la
for city in dataset:
    temp = city.loc[:,columns]
    temp.columns = [col.split('.')[-1] for col in temp.columns]
    temp['categories'] = temp.apply(category_type, axis=1)
    filtered_dataset.append(temp)
```

Exploration of Cleaned Datset

In [48]:
```
df_cleaned_Bangkok.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   name        26 non-null     object 
 1   categories  26 non-null     object 
 2   lng         26 non-null     float64
 3   lat         26 non-null     float64
dtypes: float64(2), object(2)
memory usage: 960.0+ bytes
```

In [50]:
```
df_cleaned_Tokyo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   name        30 non-null     object 
 1   categories  30 non-null     object 
 2   lng         30 non-null     float64
 3   lat         30 non-null     float64
dtypes: float64(2), object(2)
memory usage: 1.1+ KB
```

In [52]:
```
df_cleaned_New_York.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   name        30 non-null     object 
 1   categories  30 non-null     object 
 2   lng         30 non-null     float64
 3   lat         30 non-null     float64
```

```
dtypes: float64(2), object(2)
memory usage: 1.1+ KB
```

Merging Dataset

In [56]:
```
df_cleaned_Bangkok ['city'] = 'bangkok'
df_cleaned_Tokyo['city'] = 'tokyo'
df_cleaned_New_York['city'] = 'New_York'
df_final = pd.concat([df_cleaned_New_York,df_cleaned_Bangkok,df_cleaned_Tokyo])
df_final.head()
```

Out[56]:

	name	categories	lng	lat	city
0	The Bar Room at Temple Court	Hotel Bar	-74.006802	40.711448	New_York
1	The Beekman, A Thompson Hotel	Hotel	-74.006702	40.711173	New_York
2	Alba Dry Cleaner & Tailor	Laundry Service	-74.006272	40.711434	New_York
3	City Hall Park	Park	-74.007792	40.711893	New_York
4	Gibney Dance Center Downtown	Dance Studio	-74.005661	40.713923	New_York

In [57]:
```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 86 entries, 0 to 29
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   name        86 non-null     object 
 1   categories  86 non-null     object 
 2   lng         86 non-null     float64
 3   lat         86 non-null     float64
 4   city        86 non-null     object 
dtypes: float64(2), object(3)
memory usage: 4.0+ KB
```

Data Analysis For Project

MOST COMMON LOCATIONS TO VISIT IN BANGKOK, TOKYO, NEW YORK RESPECTIVELY

In [53]:
```
df_cleaned_Bangkok.categories.value_counts()
```

Out[53]:
```
History Museum         2
Café                   2
Dessert Shop           2
Palace                 2
Buddhist Temple        2
Spiritual Center       1
Neighborhood           1
Bakery                 1
Noodle House           1
Japanese Restaurant    1
Theater                1
Chinese Restaurant     1
Coffee Shop            1
Shopping Mall          1
Record Shop            1
College Bookstore      1
Wings Joint            1
```

```
Art Museum              1
Museum                  1
Historic Site           1
Soup Place              1
Name: categories, dtype: int64
```

In [54]:
```
df_cleaned_Tokyo.categories.value_counts()
```

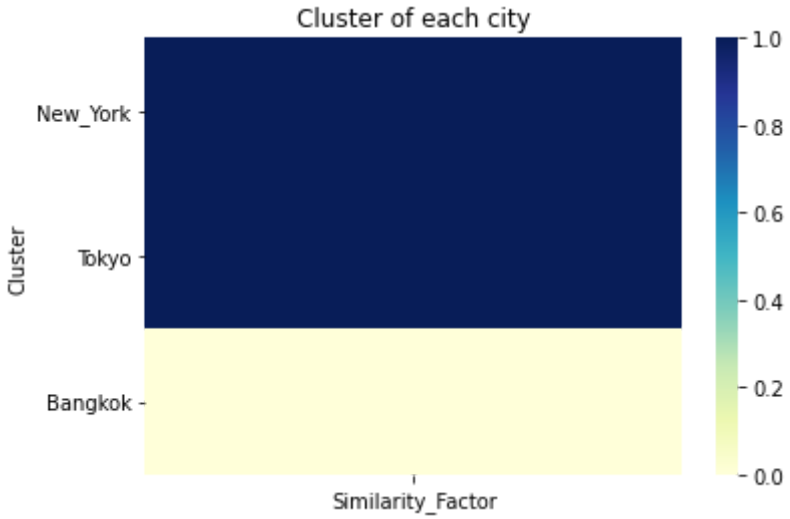
Out[54]:
```
Historic Site                   4
Park                            2
Sushi Restaurant                1
Japanese Restaurant             1
American Restaurant             1
Sake Bar                        1
Hotel Bar                       1
Garden                          1
Steakhouse                      1
Tempura Restaurant              1
Hotel                           1
Italian Restaurant              1
Supermarket                     1
Sukiyaki Restaurant             1
Mediterranean Restaurant        1
Clothing Store                  1
Plaza                           1
French Restaurant               1
Chinese Restaurant              1
Bar                             1
Brazilian Restaurant            1
Dessert Shop                    1
Wine Bar                        1
Paper / Office Supplies Store   1
Electronics Store               1
Lounge                          1
Name: categories, dtype: int64
```

In [55]:
```
df_cleaned_New_York.categories.value_counts()
```

Out[55]:
```
Hotel                   2
Gym                     2
Falafel Restaurant      2
Italian Restaurant      2
Coffee Shop             2
Hotel Bar               1
Gym / Fitness Center    1
Sandwich Place          1
American Restaurant     1
Boxing Gym              1
Laundry Service         1
Furniture / Home Store  1
Liquor Store            1
Park                    1
Cuban Restaurant        1
Indian Restaurant       1
Burger Joint            1
Pizza Place             1
Building                1
French Restaurant       1
Monument / Landmark     1
Yoga Studio             1
Taco Place              1
Dance Studio            1
Bakery                  1
Name: categories, dtype: int64
```

History locations are really popular, while in NY hotels are popular :)

In [87]:
```
import seaborn as sns
sns.heatmap(similarity.sort_values(by=['Similarity_Factor'], ascending=False), cmap=
plt.ylabel('Cluster')
plt.title('Cluster of each city')
plt.xticks(rotation = 0)
plt.yticks(rotation = 0);
```



Conclusions

Based on what we learned about the clusters and above maps, we can advise the prospective restaurant owner that their consideration of chosing Tokyo and Bangkok in the neighborhoods from USA is great. These are the neighborhoods where gastronomy is well represented and also hotels are frequent. These satisfy the criteria that the location should be in a gastronomical centre and in a location that is easily accessible for tourists and for wealthier local citizens as well. Usually, citizens prefer places with moderate climate situation, having historical sites(this also shown in above analysis), city which is more similar to New York city (considering American Citizen), city having variety of restaurants like continental, sub-continental, language preferably English. So, for a guy living in New York city, if they want some outing so preferable locations might be Bangkok, Tokyo.

In [ ]:
```
```