Introduction

Business Problem:

A Michelin star chef wants to open an upscale restaurant in Manhattan. The menu for the proposed establishment has a high fixed price, the chef wants to ensure this is not a deterrent to his local population.

Background:

The location of a restaurant often contributes to many of the features that define it, such as its predominant clientele base, cost of food, and ambiance (Pillsbury, 1987). It is therefore important to define your target clientele prior to purchasing a location for a potential establishment. Koo *et al*. (1999) has found that high food costs at restaurants are less bothersome to individuals whose primary purpose is that of business, rather than family meal outings. Additionally, Kunst *et al*. (2019) found that nearly 57% of individuals travelling on business trips selected a hotel based on its proximity to restaurants and bars.

As the menu for the client has been set at a high cost, we have therefore determined the most optimal location for the restaurant to be in an area with a high density of offices and hotels. This study however can be expanded beyond the scope of this single client, and can be of interest to any individual seeking to open a high-cost restaurant.

Data

Source:

The two relevant variable we chose to utilize to approach this problem is frequency of **hotels** and **offices** in each neighborhood.

Neighborhood boundaries: This will be obtained from the NYU Spatial Data Repository[1]. It contains details of neighborhood names, location (latitude and longitude), and the bureau each neighborhood resides in. This will then be narrowed down to focus specifically on the borough of Manhattan. The neighborhood locations will be utilized when creating cluster locations in Foursquare API.

Neighborhood venues: Data for each neighborhood will be obtained from the foursquare database. Data will be "called" in the notebook using foursquare credentials, and the location of interest latitude and longitude.

By using the selected sources, we will find neighborhoods utilizing a K-means clustering analysis with the *Foursquare* venues data. Following our analysis, we will propose the best neighborhoods to open the new restaurant.

[1] https://geo.nyu.edu/catalog/nyu_2451_34572

Methodology

Download Necessary packages

- Import numpy as np: The library to handle data in a vectorized manner
- Import pandas as pd: The library for data analsysis
- Import json: The library to handle JSON files
- Install geopy from geopy.geocoders import Nominatim: Will convert an address into latitude and longitude values
- Import json_normalize: Will transform JSON file into a pandas dataframe
- Import Matplot lib and associated plotting modules
- import k-means from clustering stage from sklearn.cluster import KMeans
- Install folium: The map rendering library

Download the Data Source

The file we used from the NYU Spatial data repository was a json file. The data will be imported and transformed into a pandas data frame. This data contains the information regarding location, such as the name of the borough, neighborhood, and the latitude and longitude.

The bureau of Manhattan will be isolated from the Dataframe through filtering on the 'Borough' Column

Visualize the Initial Map

Using the geopy function, the latitude and longitude of Manhattan was extracted. Once extracted, folium was used to initially visualize the data.

Connecting Foursquare Credentials

Foursquare was connected using the *Client ID, Client Secret,* and *Version*.

Testing one neighborhood in the data frame

To ensure accuracy, one neighborhood in Manhattan was extracted from the data frame previously extracted from the json file. We defined a function that extracted the categories of each venue i.e "Office", "Restaurant", "Hotel", etc. We set a Limit of the number of venues returned to 200 and the radius of the data set was set to 500. The API was called using the defined test neighborhood information and connected foursquare credentials. We used the GET request URL to examine the results. Following the import of the dataset from four square, we structured it into a pandas data frame and defined the columns 'Venue Name', 'Venue Category', 'Venue Location Latitude', 'Venue Location Longitude.'

Repeat analysis with all boroughs of Manhattan

After testing the data, we defined a function to repeat the same process for all of Manhattan, calling the four square API via a URL, and completing the GET request. We visualized the data set by organizing the columns according to the frequency of different venues in each Neighborhood. The data was translated into a pandas dataframe.

Creating Clusters

We then clustered the neighborhoods into 5 distinct clusters. The cluster labels were inserted into the data set.

```
kclusters = 5

manhattan_grouped_clustering = manhattan_grouped.drop('Neighborhood', 1)

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

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

# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

manhattan_merged=Manhattan_df

manhattan_merged = manhattan_merged.join(neighborhoods_venues_sorted.set_i
ndex('Neighborhood'), on='Neighborhood')

manhattan_merged.head(20)
```

Define the list of Venue Categories of Interest

As we were specifically looking at the frequency of *Hotels* and *Offices* separate data frames were created for each and then merged.

In [234]:
```
Hotels_List = manhattan_venues.loc[manhattan_venues['Venue Cate
gory'].isin(['Hotel'])]
Hotels_List

Hotels_merged=Manhattan_df

Hotels_merged = Hotels_merged.join(Hotels_List.set_index('Neigh
borhood'), on='Neighborhood')

Hotels_merged.head(100)

Hotels_merged=Hotels_merged[Hotels_merged.Venue !='NaN']

Hotels_merged.head(50)

#remove nan values

Hotels_merged.dropna()
```

In [235]:
```
Office_List = manhattan_venues.loc[manhattan_venues['Venue Cate
gory'].isin(['Office'])]
Office_List

Office_merged=Manhattan_df

Office_merged = Office_merged.join(Office_List.set_index('Neigh
borhood'), on='Neighborhood')

Office_merged.head(100)

Office_merged=Office_merged[Office_merged.Venue !='NaN']

Office_merged.head(50)

#remove nan values
Office_merged.dropna()
```

```
In [236]: Office_Hotel_List = manhattan_venues.loc[manhattan_venues['Venue Category'].isin(['Office', 'Hotel'])]

          Office_Hotel_List = Office_Hotel_List.join(Manhattan_df.set_index('Neighborhood'), on='Neighborhood')

          Office_Hotel_List

Out[236]:
```

Finding the frequency

We then found the frequency of hotels and offices in each neighborhood.

```
In [262]: print('There are {} hotels and offices on record in New York.'.format(len(Office_Hotel_List['Venue'].unique())))

          There are 62 hotels and offices on record in New York.

In [261]: Office_Hotel_List['Neighborhood'].value_counts()

Out[261]: Midtown                7
          Midtown South          5
          Battery Park City      5
          Financial District     4
          Civic Center           4
          Chelsea                4
          Hudson Yards           4
          Upper East Side        3
          Noho                   3
          Gramercy               3
          Murray Hill            3
          Clinton                3
          Tribeca                2
          Little Italy           2
          Turtle Bay             2
          Soho                   2
          Lincoln Square         2
          Tudor City             2
          Carnegie Hill          1
          Upper West Side        1
          Sutton Place           1
          Flatiron               1
          Greenwich Village      1
          Chinatown              1
          Name: Neighborhood, dtype: int64
```

The highest frequency of hotels and offices are in Midtown and Midtown South

Visualize the Clusters using kMeans

We then created a visual using folium. We added points of interest (Hotels and Offices) to the map.

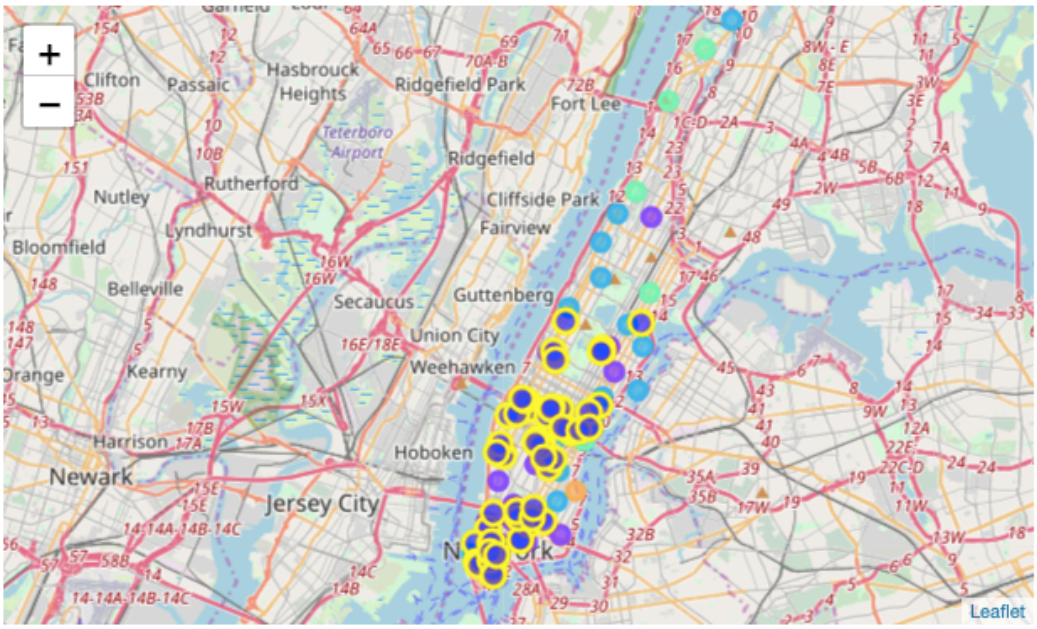
In [240]:
```
# create map
map_clusters = folium.Map(location=[40.7896239, -73.9598939], zoom_start=1
1)

# 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 clusters to the map
markers_colors = []
for lat, lon, poi, cluster in zip(manhattan_merged['Latitude'], manhattan_
merged['Longitude'], manhattan_merged['Neighborhood'], manhattan_merged['C
luster 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)

#add points of interest markers to the map (Hotels & Offices)
for lat, long, points, cat in zip(Office_Hotel_List['Venue Latitude'], Off
ice_Hotel_List['Venue Longitude'],
                                  Office_Hotel_List['Venue'], Office_Hotel_
List['Venue Category']):
        label = '{}, {}'.format(points, cat)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, long],
            radius = 7,
            popup = label,
            color = 'yellow',
            fill = True,
            fill_color = 'blue',
```

Out[240]:



Analyze the Clusters

We then analyzed each cluster to assess the most common venues in each neighborhood. This was repeated for 5 clusters.

In [241]:
```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 0, manhattan_me
rged.columns[[1] + list(range(5, manhattan_merged.shape[1]))]]
```

Out[241]:

	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 M Comn Ve
33	Midtown South	Korean Restaurant	Hotel	Hotel Bar	Japanese Restaurant	Dessert Shop	Coffee Shop	Cosme Shop

Results

We found that the highest cluster of hotels and offices were concentrated in Midtown South and Midtown. This was observed in both the frequency analysis and the k-means cluster analysis.

```
In [261]: Office_Hotel_List['Neighborhood'].value_counts()

Out[261]: Midtown               7
          Midtown South         5
          Battery Park City     5
          Financial District    4
          Civic Center          4
          Chelsea               4
          Hudson Yards          4
          Upper East Side       3
          Noho                  3
          Gramercy              3
          Murray Hill           3
          Clinton               3
          Tribeca               2
          Little Italy          2
          Turtle Bay            2
          Soho                  2
          Lincoln Square        2
          Tudor City            2
          Carnegie Hill         1
          Upper West Side       1
          Sutton Place          1
          Flatiron              1
          Greenwich Village     1
          Chinatown             1
          Name: Neighborhood, dtype: int64
```

Discussion

Our data demonstrated the highest frequency of Offices and Hotels in Midtown South and Midtown Neighborhoods. This would suggest that due to the nature of our clients proposed menu pricing, these would be the most optimal neighborhoods. Koo *et al*. (1999) demonstrated that meals with a primary purpose of business are less sensitive to cost. Additionally, Kunst *et al*. (2019) showed that individuals traveling on business are more likely to select a hotel in close vicinity to a restaurant.

Limitations and Future Research

We found data relating to Office locations were limited, and likely limited the generalizability of these results. Future studies should look to incorporate a larger data set to determine the frequency of office locations in the borough of Manhattan.

Additionally, other factors contribute to the overall success of a restaurant's outcome. While proximity to a hotel and office are factors in determining the success of a restaurant, other factors include vicinity to competition (Parsa *et al*., 2011). Future studies should incorporate other factors which can contribute to the failure of a restaurant on the basis of location.

Conclusion

Based on the above analysis, we would propose the most ideal location for an upscale restaurant is either in Midtown South or Midtown. As there is a lower concentration of restaurants in Midtown South than in Midtown, we would suggest the most ideal location for the new restaurant location is in Midtown.

References

H. G. Parsa, John Self, Sandra Sydnor-Busso & Hae Jin Yoon (2011) Why Restaurants Fail? Part II - The Impact of Affiliation, Location, and Size on Restaurant Failures: Results from a Survival Analysis, Journal of Foodservice Business Research, 14:4, 360-379, DOI: 10.1080/15378020.2011.625824

Pillsbury, R. (1987) FROM HAMBURGER ALLEY TO HEDGEROSE HEIGHTS: TOWARD A MODEL OF RESTAURANT LOCATION DYNAMICS, The Professional Geographer, 39:3, 326-344, DOI: 10.1111/j.0033-0124.1987.00326.x

Koo, L.C., Tao, F.K. and Yeung, J.H., 1999. Preferential segmentation of restaurant attributes through conjoint analysis. *international Journal of Contemporary Hospitality management*.

Kunst, A. (2017) How important are nearby restaurants and bars when choosing a hotel for a business trip? *Statista*. https://www.statista.com/statistics/719823/importance-of-restaurant-proximity-to-hotels-to-business-travelers-us/