Table of Contents

Problem to resolved

The challenge to resolve is being able to find an apartment unit in Manhattan NY that offers similar characteristics and benefits to my current situation. Therefore, in order to set a basis for comparison, I want to place subject to the following conditions:

•Top ammenities in the selected neighborhood shall be similar to current residence
•Desirable to have venues such as coffee shops, restaurants Asian Thai, wine stores, gym and food shops
•As a reference, I have included a map of venues near current residence in Mumbai
•

Interested Audience

I believe this is a relevant project for a person or entity considering moving to a major city in Europe, US or Asia, since the approach and methodologies used here are applicable in all cases. The use of FourSquare data and mapping techniques combined with data analysis will help resolve the key questions arisen. Lastly, this project is a good practical case toward the development of Data Science skills

Data Information

The data for this project has been retrieved and processed through multiple sources, giving careful considerations to the accuracy of the methods used. 3.1 Neighbourhoods
The data of the neighbourhoods in Mumbai can be extracted out by web scraping using BeautifulSoup library for Python.

- **Geocoding -** The file contents from kolkata.csv is retrieved into a Pandas DataFrame. The latitude and longitude of the neighbourhoods are retrieved using Google Maps Geocoding API. The geometric location values are then stored into the intial dataframe.

- **Venue Data** - From the location data obtained after Web Scraping and Geocoding, the venue data is found out by passing in the required parameters to the FourSquare API, and creating another DataFrame to contain all the venue details along with the respective neighbourhoods.

- **Data Requirements**
  - Geodata for current areas in Mumbai with venues established using Foursquare.
  - List of Mumbai neighborhoods with clustered venues established via Foursquare (as in Course Lab for Toronto and Manhattan).
  - List of venues in Mumbai with addresses and geo data (latitude,longitude).
  - List of **offices** that may need our services in Mumbai area with information on neighborhood location, address and complemented with geo data via Nominatim.

- **Sources -** The list of Manhattan neighborhoods is worked out during LAB exercise during the course. A csv file was created which will be read in order to create a dataframe and its mapping. The csv file 'mh_neigh_data.csv' has the following below data structure. The file will be directly read to the Jupiter Notebook for convenience and space savings. The clustering of neighborhoods and mapping will be shown however. An algorythm was used to determine the geodata from Nominatim

Methodology

This section represents the main component of the report where the data is gathered, prepared for analysis. The tools described are used here and the Notebook cells indicates the execution of steps.

The analysis and the stragegy:

The strategy is based on mapping the above described data in order to facilitate the choice of a candidate places for accomodation. The choice is made based on the demands imposed : similar venues to Mumbai. This visual approach and maps with popups labels allow quick identification of location, thus making the selection very easy.

The procesing of these DATA and its mapping will allow to answer the key questions to make a decision. Find all the venues of the places and how the venues are distributed aming the neighbourhoods

Sample of a code

Use Foursquare Client ID and Secret

In [4]:
```
# your Foursquare ID
CLIENT_ID = 'YURAW5R5XNEOB3HBPIL1TKN4MB2HM2VBXGPFEZATUZ2K23K1'
# your Foursquare Secret
CLIENT_SECRET = 'BG0ZCMUVHPLYEZZCOF4A3YBG0HERLP5UYLOJDJNUZRZA2SBZ'
VERSION = '20202204'
LIMIT = 30

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

```
Your credentails:
CLIENT_ID: YURAW5R5XNEOB3HBPIL1TKN4MB2HM2VBXGPFEZATUZ2K23K1
CLIENT_SECRET:BG0ZCMUVHPLYEZZCOF4A3YBG0HERLP5UYLOJDJNUZRZA2SBZ
```

Install geopy and import Nominatim to convert address into latitude & longitude

In [2]:
```
!conda install -c conda-forge geopy --yes

# convert an address into latitude and longitude values
from geopy.geocoders import Nominatim
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - geopy


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    geopy-1.21.0               |             py_0          58 KB  conda-forge
    geographiclib-1.50         |             py_0          34 KB  conda-forge
    ------------------------------------------------------------
                                           Total:          92 KB

The following NEW packages will be INSTALLED:

    geographiclib: 1.50-py_0   conda-forge
    geopy:         1.21.0-py_0 conda-forge


Downloading and Extracting Packages
geopy-1.21.0         | 58 KB     | ##################################### | 100%
geographiclib-1.50   | 34 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Give the city name and use Nominatim to find the latitude & longitude

In [20]:
```
# define the city and get its latitude & longitude

address = 'Bandra, Mumbai'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate : {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate : 19.0549792, 72.8402203.
```

In [21]:
```
neighborhood_latitude = 19.0549792
neighborhood_longitude = 72.8402203
```

In [48]:
```
LIMIT = 100 # limit of number of venues returned by Foursquare API
radius = 1000 # define radius
#search_query = 'Hotel'
# create URL
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}
&radius={}&limit={}'.format(
    CLIENT_ID,
```

Accuracy of the Geocoding API : In the initial development phase with OpenCage Geocoder API, the number of erroneous results were of an appreciable amount, which led to the development of an algorithm to analyze the accuracy of the Geocoding API used. In the algorithm developed, Geocoding API from various providers were tested, and in the end, Google Maps Geocoder API turned out to have the least number of errors in the analysis.

Folium : It builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library. All cluster visualization are done with help of Folium which in turn generates a Leaflet map made using OpenStreetMap technology.

Venues

Due to high variety in the venues, only the top common venues are selected and a new DataFrame is made, which is used to train the K-means Clustering Algorithm.

Venues in Mumbai (Table and Map)

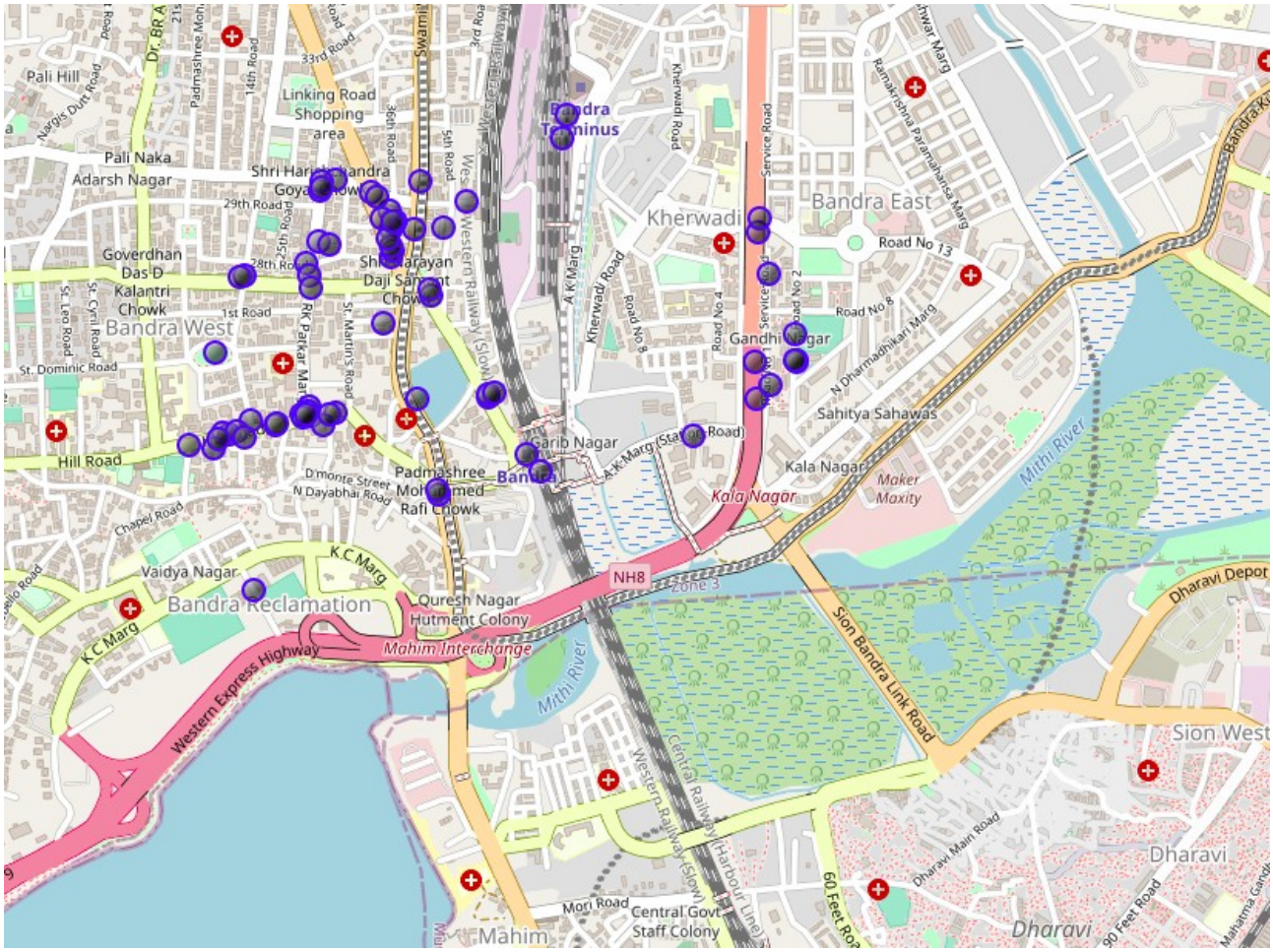
```
venues = results['response']['groups'][0]['items']
nearby_venues = json_normalize(venues) # flatten JSON
# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]
# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)
# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]
# Venues near current Singapore residence place
nearby_venues.head(10)
```

Out[55]:

	name	categories	lat	lng
0	Coalz	Café	19.056468	72.839154
1	National Restaurant	Indian Restaurant	19.056546	72.839141
2	Escobar	Lounge	19.060177	72.836446
3	Golconda Bowl	Indian Restaurant	19.055977	72.834189
4	Heng Bok	Korean Restaurant	19.059163	72.837547
5	Chocolateria San Churro	Dessert Shop	19.061918	72.834480
6	MIG Cricket Club	Sports Club	19.057461	72.847706
7	Godrej Nature's Basket	Gourmet Shop	19.055509	72.831935
8	Persian Darbar	Indian Restaurant	19.060323	72.836481
9	Nature's Basket	Gourmet Shop	19.055533	72.831778

```
nearby_venues.shape
```

Out[56]: (71, 4)



Result

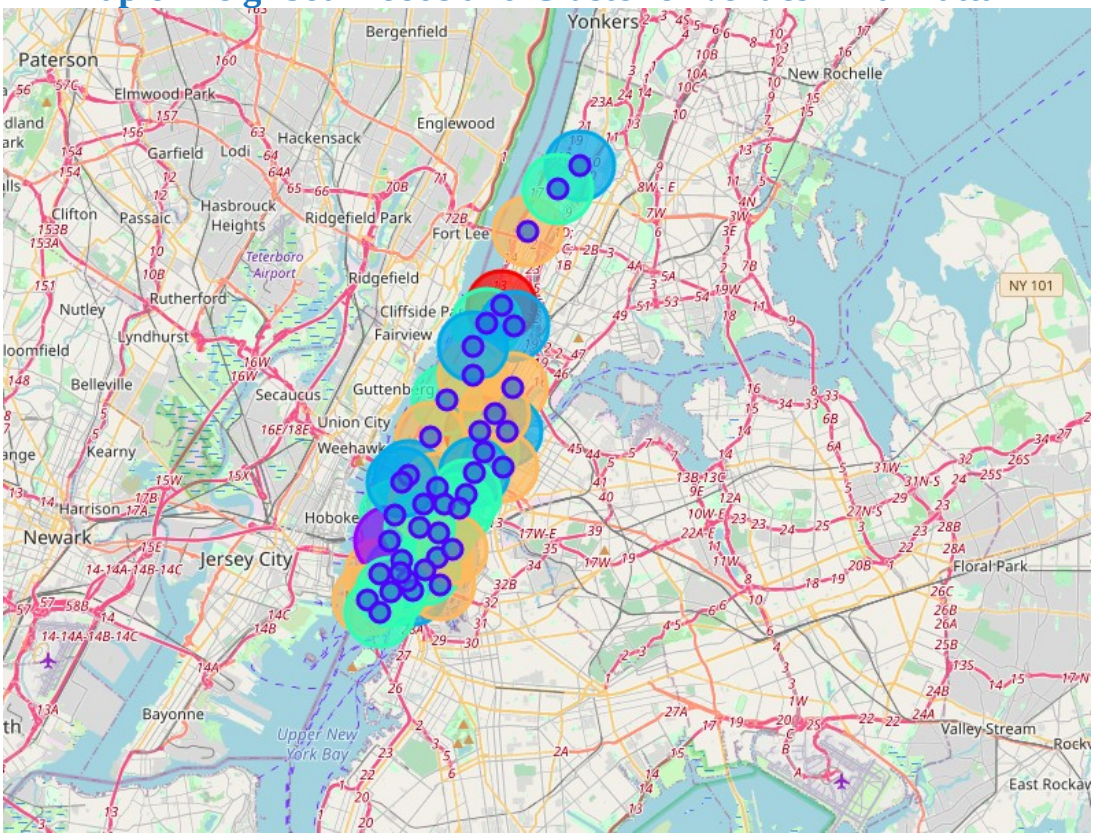
The neighbourhoods are divided into n clusters where n is the number of clusters found using the optimal approach. The clustered neighbourhoods are visualized using different colours so as to make them distinguishable.

```
body = client_09aac778b57d42f2a3f89dd160133d1c.get_object(Bucket='datascience-donotdelete-pr-ku7h6sj7wlfz9e',Key='manhattan_merged.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df_data_2 = pd.read_csv(body)
df_data_2.head()
```

	Borough	Neighborhood	Latitude	Longitude	Cluster 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
0	Manhattan	Marble Hill	40.876551	-73.910660	2	Coffee Shop	Discount Store	Yoga Studio	Steakhouse	Supplement Shop	Tennis Stadium	Shoe Store	Gym	Bank	Seafood Restaurant
1	Manhattan	Chinatown	40.715618	-73.994279	2	Chinese Restaurant	Cocktail Bar	Dim Sum Restaurant	American Restaurant	Vietnamese Restaurant	Salon / Barbershop	Noodle House	Bakery	Bubble Tea Shop	Ice Cream Shop
2	Manhattan	Washington Heights	40.851903	-73.936900	4	Café	Bakery	Mobile Phone Shop	Pizza Place	Sandwich Place	Park	Gym	Latin American Restaurant	Tapas Restaurant	Mexican Restaurant
3	Manhattan	Inwood	40.867684	-73.921210	3	Mexican Restaurant	Lounge	Pizza Place	Café	Wine Bar	Bakery	American Restaurant	Park	Frozen Yogurt Shop	Spanish Restaurant
4	Manhattan	Hamilton Heights	40.823604	-73.949688	0	Mexican Restaurant	Coffee Shop	Café	Deli / Bodega	Pizza Place	Liquor Store	Indian Restaurant	Sushi Restaurant	Sandwich Place	Yoga Studio

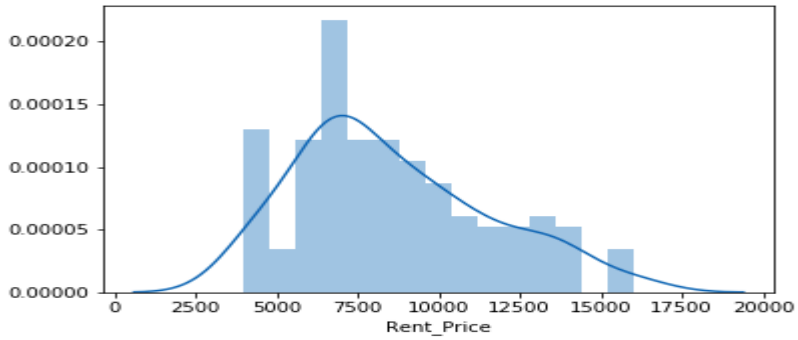
Map of Neighbourhoods and Cluster of Venues - Manhattan



Analyse (Graph – The price of the places in Manhattan)

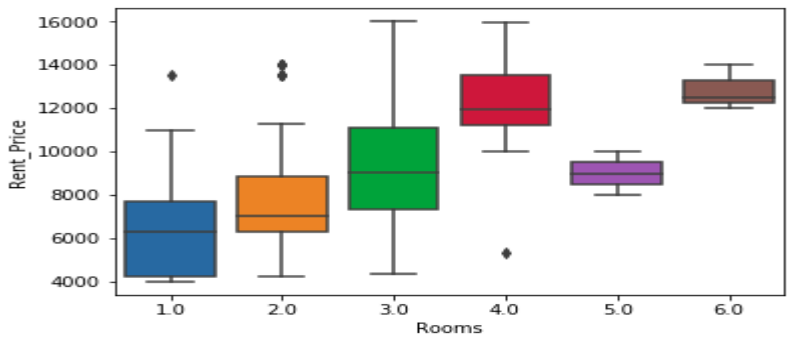
```
import seaborn as sns
sns.distplot(df_data_4['Rent_Price'],bins=15)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa5c9a239b0>
```

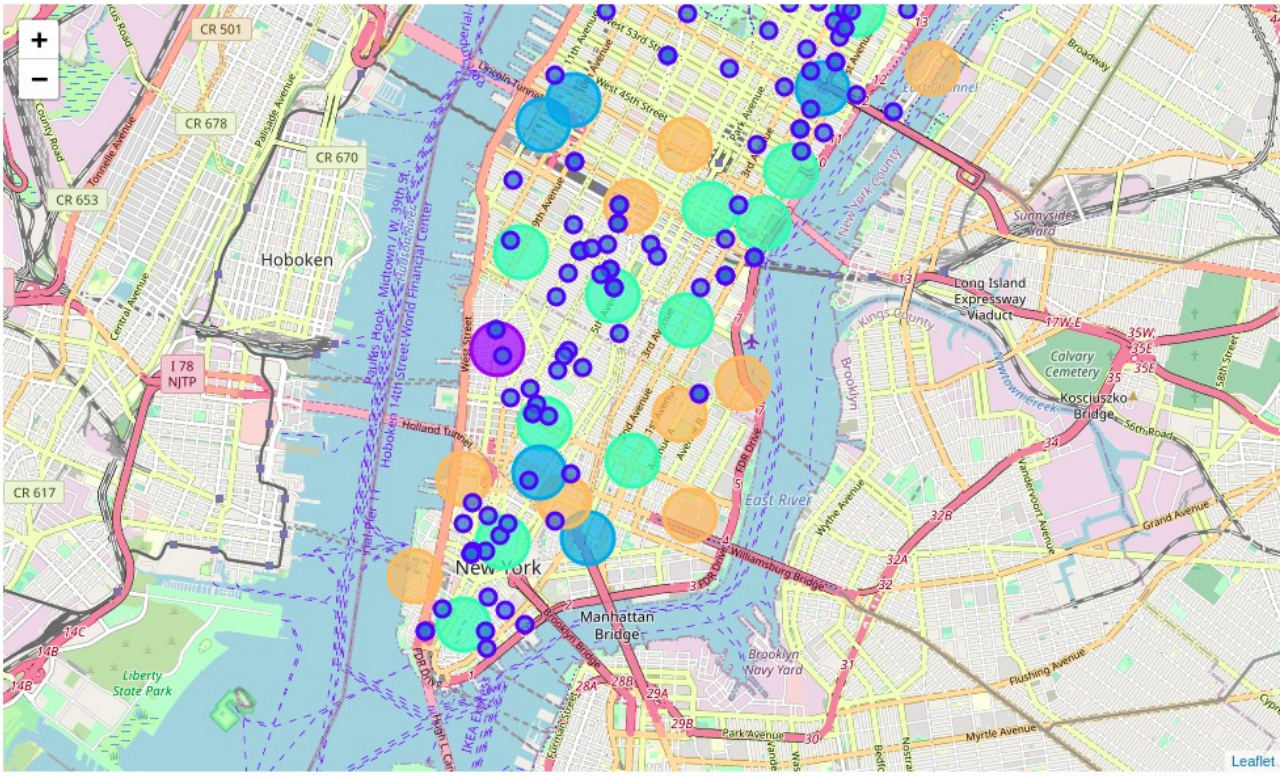



```
sns.boxplot(x='Rooms', y= 'Rent_Price', data=df_data_4)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa5b94e8208>
```



Map of Manhattan showing the places for rent and the cluster of venues. One can point to a rental place for price and address location information while knowing the cluster venues around it.



Discussion

- After analyzing the various clusters produced by the Machine learning algorithm, cluster no.2, is a prime fit to solving the problem of finding a cluster with common venue as a place for rent. After examining several cluster data , I concluded that cluster 2 resembles closer the Mumbai place, therefore providing guidance as to where to look for the future apartment.
- This analysis is performed on limited data. This may be right or may be wrong. But if good amount of data is available there is scope to come up with better results.
- It can be done more detailed analysis by adding other factors such as transportation, demographics of inhabitants.

Conclusion

Although all of the goals of this project were met there is definitely room for further improvement and development as noted below. However, the goals of the project were met and, with some more work, could easily be devleoped into a fully phledged application that could support the opening a business idea in an unknown location.
