The Battle of Neighborhoods

Finding a job in Italian Restaurant of New York City

Introduction:

This project is for those who seek for a job and need to find a temporary hotel to stay until he gets a job. To simplify, here the person seeks for a chef position in restaurants and needs to find a hotel to stay in New York city.

Business Problem:

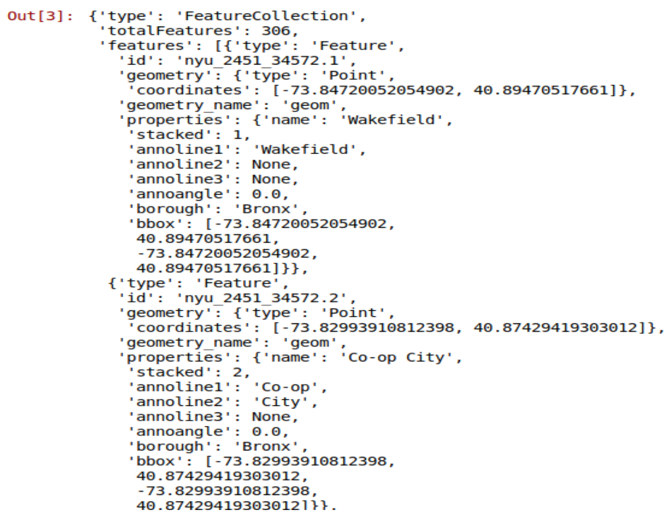
To recommend a list of hotels for a person in the neighborhood which has more restaurant venues is our business problem. Basically, if a person is having more experience in any field, he will look for a better career opportunity. So, he will look for company which is having highest ratings. Similarly, here for the person who looks for a job in a restaurant, he needs to apply for a restaurant with high ratings. Also, for a temporary stay he may want a hotel in same neighbourhood which has high ratings. New York City is the most populous city in the united states and more happening city which has more opportunities. Because of the city's immigrant history, there are plenty of restaurants with variety of cuisine are there. Hence for a job seeker, based on his cooking style like Chinese, Italian, Indian etc., we need to find neighborhoods cluster accordingly. After finding the neighborhoods list, we need to find list to best hotels in those neighborhoods cluster for the client to stay. In this project, let us say that our client Michael looks for a chef position in Italian restaurant in the city of New York.

Data Section:

For New York City's Boroughs, Neighborhoods and their coordinate values, we use the below JSON file.

`https://ibm.box.com/shared/static/fbpwbovar7lf8p5sgddm06cgipa2rxpe.json`

The JSON file looks like below.



```
Out[3]: {'type': 'FeatureCollection',
          'totalFeatures': 306,
          'features': [{'type': 'Feature',
            'id': 'nyu_2451_34572.1',
            'geometry': {'type': 'Point',
             'coordinates': [-73.84720052054902, 40.89470517661]},
            'geometry_name': 'geom',
            'properties': {'name': 'Wakefield',
             'stacked': 1,
             'annoline1': 'Wakefield',
             'annoline2': None,
             'annoline3': None,
             'annoangle': 0.0,
             'borough': 'Bronx',
             'bbox': [-73.84720052054902,
              40.89470517661,
              -73.84720052054902,
              40.89470517661]}},
           {'type': 'Feature',
            'id': 'nyu_2451_34572.2',
            'geometry': {'type': 'Point',
             'coordinates': [-73.82993910812398, 40.87429419303012]},
            'geometry_name': 'geom',
            'properties': {'name': 'Co-op City',
             'stacked': 2,
             'annoline1': 'Co-op',
             'annoline2': 'City',
             'annoline3': None,
             'annoangle': 0.0,
             'borough': 'Bronx',
             'bbox': [-73.82993910812398,
              40.87429419303012,
              -73.82993910812398,
              40.87429419303012]}},
```

From the above JSON file, we extract only the Bronx Borough's Neighborhoods and store them as below dataframe.

```
In [11]: Bronx_data = neighborhoods[neighborhoods['Borough'] == 'Bronx'].reset_index(drop=True)
         Bronx_data.head()
```

Out[11]:

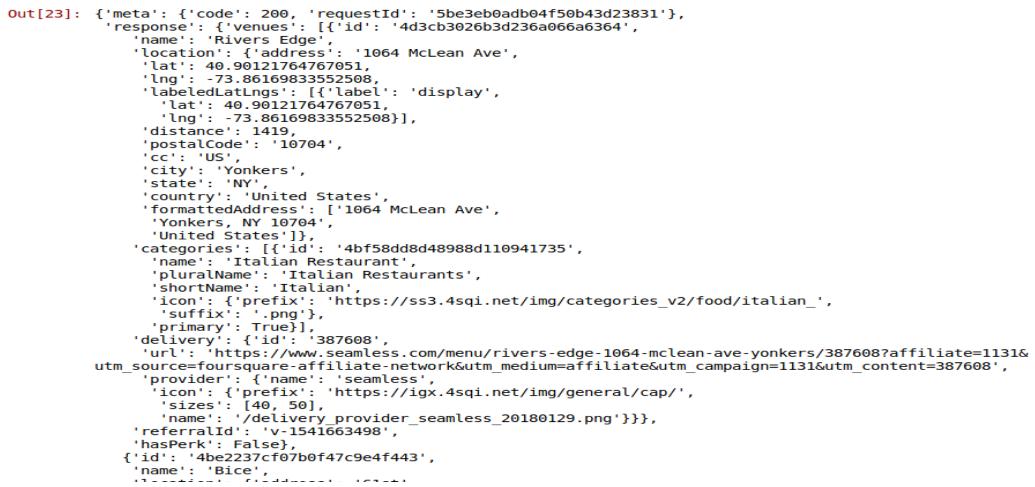
	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

Foursquare is used for getting the venues available in a particular neighborhood and their venue details such as rating. In our case, Foursquare API is used to get the italian restaurants available in Bronx Borough's Neighborhoods and their ratings.

The ID for Italian restaurants in Foursquare API is 4bf58dd8d48988d110941735. Below sample Foursquare API call will return the list of italian restaurants available in a Neighborhood.

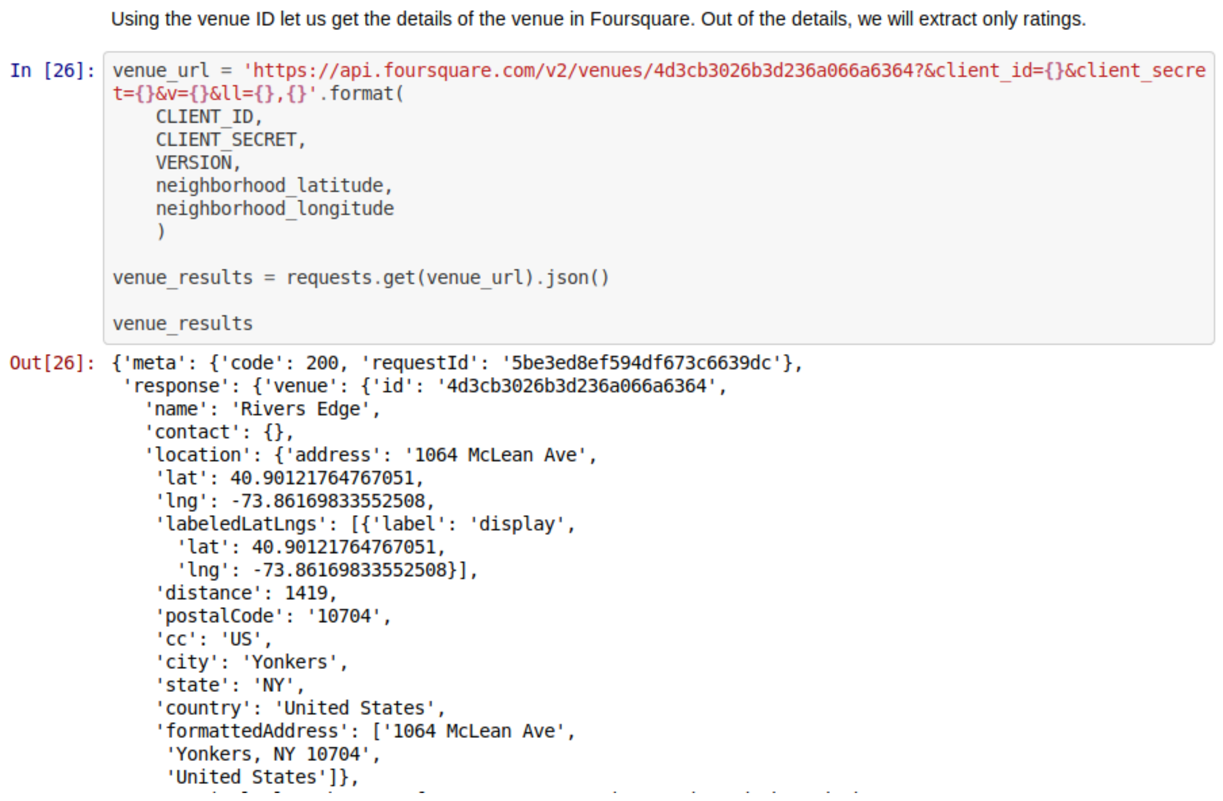
```
Latitude and longitude values of Wakefield are 40.89470517661, -73.84720052054902.

In [22]: LIMIT = 100 # limit of number of venues returned by Foursquare API

         radius = 2000 # define radius
         # create URL
         url = 'https://api.foursquare.com/v2/venues/search?categoryId=4bf58dd8d48988d110941735&client_id={}&c
         lient_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
             CLIENT_ID,
             CLIENT_SECRET,
             VERSION,
             neighborhood_latitude,
             neighborhood_longitude,
             radius,
             LIMIT)
         url

Out[22]: 'https://api.foursquare.com/v2/venues/search?categoryId=4bf58dd8d48988d110941735&client_id=JJGGL0HAFM
         BKDVKLPEMMURPPL2SSAAG0OYDNAB4PFHDYQ4NL&client_secret=4XAV5XBRUSFD00FSB5XEZ5Q4SCNKXXU0PCV0DOUJETW3NLDF
         &v=20180605&ll=40.89470517661,-73.84720052054902&radius=2000&limit=100'
```

Below JSON format is returned from Foursquare.



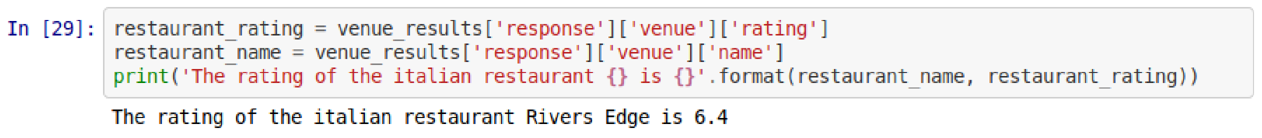
```
Out[23]: {'meta': {'code': 200, 'requestId': '5be3eb0adb04f50b43d23831'},
          'response': {'venues': [{'id': '4d3cb3026b3d236a066a6364',
             'name': 'Rivers Edge',
             'location': {'address': '1064 McLean Ave',
              'lat': 40.90121764767051,
              'lng': -73.86169833552508,
              'labeledLatLngs': [{'label': 'display',
                'lat': 40.90121764767051,
                'lng': -73.86169833552508}],
              'distance': 1419,
              'postalCode': '10704',
              'cc': 'US',
              'city': 'Yonkers',
              'state': 'NY',
              'country': 'United States',
              'formattedAddress': ['1064 McLean Ave',
               'Yonkers, NY 10704',
               'United States']},
             'categories': [{'id': '4bf58dd8d48988d110941735',
               'name': 'Italian Restaurant',
               'pluralName': 'Italian Restaurants',
               'shortName': 'Italian',
               'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
                'suffix': '.png'},
               'primary': True}],
             'delivery': {'id': '387608',
              'url': 'https://www.seamless.com/menu/rivers-edge-1064-mclean-ave-yonkers/387608?affiliate=1131&
         utm_source=foursquare-affiliate-network&utm_medium=affiliate&utm_campaign=1131&utm_content=387608',
              'provider': {'name': 'seamless',
               'icon': {'prefix': 'https://igx.4sqi.net/img/general/cap/',
                'sizes': [40, 50],
                'name': '/delivery_provider_seamless_20180129.png'}}},
             'referralId': 'v-1541663498',
             'hasPerk': False},
            {'id': '4be2237cf07b0f47c9e4f443',
             'name': 'Bice',
```

From the JSON format, the list of restaurants are extracted and using their venue ID, the rating can be obtained like below.


Using the venue ID let us get the details of the venue in Foursquare. Out of the details, we will extract only ratings.

```
In [26]: venue_url = 'https://api.foursquare.com/v2/venues/4d3cb3026b3d236a066a6364?&client_id={}&client_secre
         t={}&v={}&ll={},{}'.format(
             CLIENT_ID,
             CLIENT_SECRET,
             VERSION,
             neighborhood_latitude,
             neighborhood_longitude
             )

         venue_results = requests.get(venue_url).json()

         venue_results
```

```
Out[26]: {'meta': {'code': 200, 'requestId': '5be3ed8ef594df673c6639dc'},
          'response': {'venue': {'id': '4d3cb3026b3d236a066a6364',
            'name': 'Rivers Edge',
            'contact': {},
            'location': {'address': '1064 McLean Ave',
             'lat': 40.90121764767051,
             'lng': -73.86169833552508,
             'labeledLatLngs': [{'label': 'display',
               'lat': 40.90121764767051,
               'lng': -73.86169833552508}],
             'distance': 1419,
             'postalCode': '10704',
             'cc': 'US',
             'city': 'Yonkers',
             'state': 'NY',
             'country': 'United States',
             'formattedAddress': ['1064 McLean Ave',
              'Yonkers, NY 10704',
              'United States']},
```

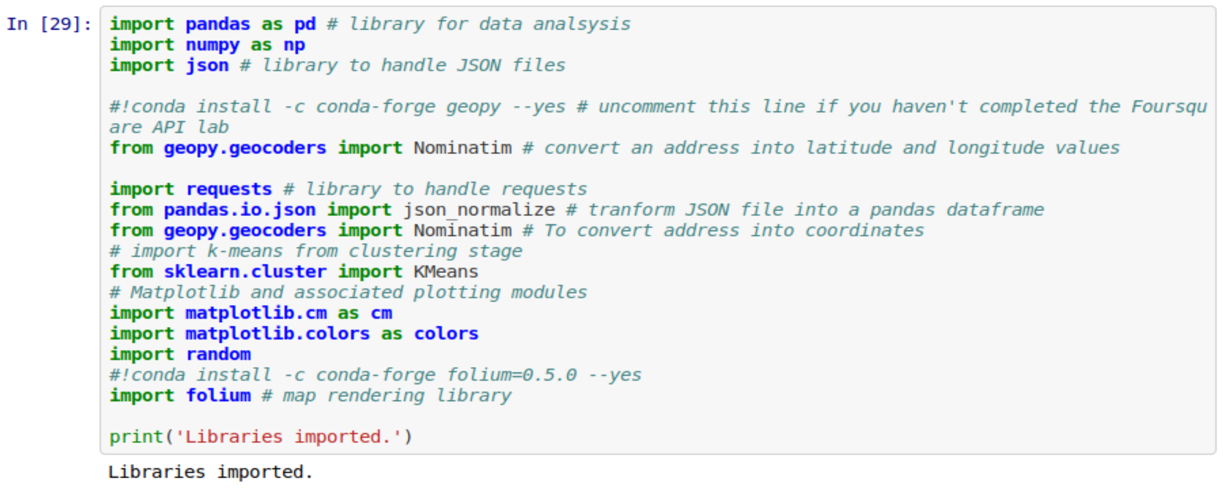


```
In [29]: restaurant_rating = venue_results['response']['venue']['rating']
         restaurant_name = venue_results['response']['venue']['name']
         print('The rating of the italian restaurant {} is {}'.format(restaurant_name, restaurant_rating))

         The rating of the italian restaurant Rivers Edge is 6.4
```

During the exact coding I could not retrieve the ratings of all the Italian restaurants and hence I had to randomly generate the ratings of italian restaurants based on their counts. This is because the Foursquare limits us getting the ratings of a venue by premium call. My quota has exceeded for premium call.

Methodology:

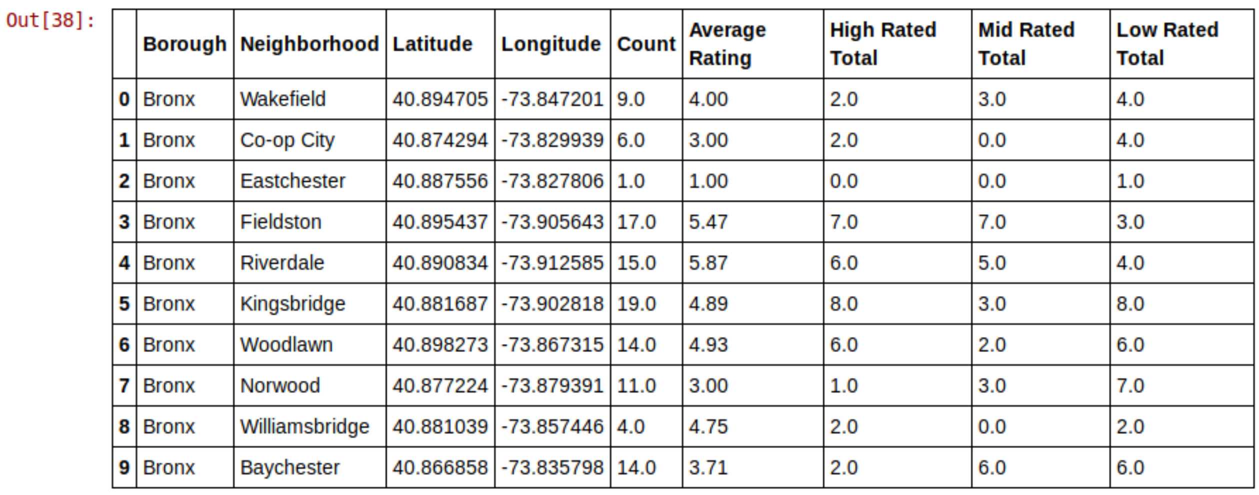
The below libraries were imported and used.



```
In [29]: import pandas as pd # library for data analsysis
         import numpy as np
         import json # library to handle JSON files

         #!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursqu
         are API lab
         from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

         import requests # library to handle requests
         from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
         from geopy.geocoders import Nominatim # To convert address into coordinates
         # import k-means from clustering stage
         from sklearn.cluster import KMeans
         # Matplotlib and associated plotting modules
         import matplotlib.cm as cm
         import matplotlib.colors as colors
         import random
         #!conda install -c conda-forge folium=0.5.0 --yes
         import folium # map rendering library

         print('Libraries imported.')

         Libraries imported.
```

After getting done with data preparation as mentioned in Data Section, the italian restaurants are grouped by high rating, mid rating and low rating counts. Also I have calculated the average rating of the available restaurants. The high rated restaurants will have ratings in range of from 7 to 10. The mid rated restaurants will have ratings from 4 to 6. The low rated restaurants will have below 3 as ratings.

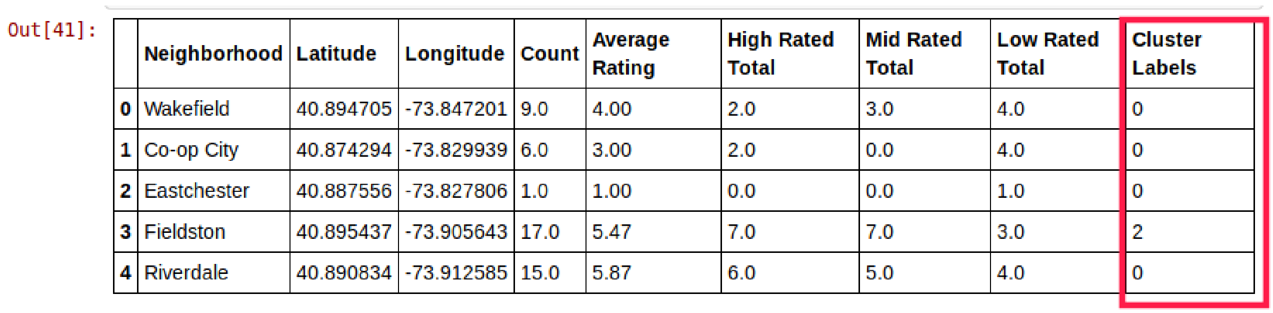
So, the dataframe looks like below.



Out[38]:

	Borough	Neighborhood	Latitude	Longitude	Count	Average Rating	High Rated Total	Mid Rated Total	Low Rated Total
0	Bronx	Wakefield	40.894705	-73.847201	9.0	4.00	2.0	3.0	4.0
1	Bronx	Co-op City	40.874294	-73.829939	6.0	3.00	2.0	0.0	4.0
2	Bronx	Eastchester	40.887556	-73.827806	1.0	1.00	0.0	0.0	1.0
3	Bronx	Fieldston	40.895437	-73.905643	17.0	5.47	7.0	7.0	3.0
4	Bronx	Riverdale	40.890834	-73.912585	15.0	5.87	6.0	5.0	4.0
5	Bronx	Kingsbridge	40.881687	-73.902818	19.0	4.89	8.0	3.0	8.0
6	Bronx	Woodlawn	40.898273	-73.867315	14.0	4.93	6.0	2.0	6.0
7	Bronx	Norwood	40.877224	-73.879391	11.0	3.00	1.0	3.0	7.0
8	Bronx	Williamsbridge	40.881039	-73.857446	4.0	4.75	2.0	0.0	2.0
9	Bronx	Baychester	40.866858	-73.835798	14.0	3.71	2.0	6.0	6.0

For this dataframe, I have applied the K Means clustering algorithm to cluster the neighborhoods based on their counts and average ratings. I wanted to have 3 clusters and I got below dataframe with clusters.

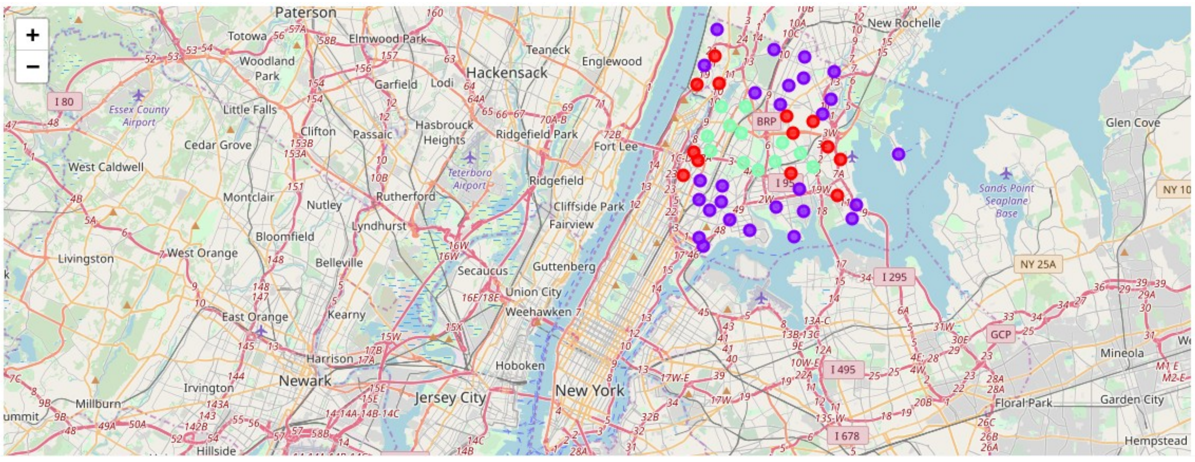


Out[41]:

	Neighborhood	Latitude	Longitude	Count	Average Rating	High Rated Total	Mid Rated Total	Low Rated Total	Cluster Labels
0	Wakefield	40.894705	-73.847201	9.0	4.00	2.0	3.0	4.0	0
1	Co-op City	40.874294	-73.829939	6.0	3.00	2.0	0.0	4.0	0
2	Eastchester	40.887556	-73.827806	1.0	1.00	0.0	0.0	1.0	0
3	Fieldston	40.895437	-73.905643	17.0	5.47	7.0	7.0	3.0	2
4	Riverdale	40.890834	-73.912585	15.0	5.87	6.0	5.0	4.0	0

Below is the visualization of clustered map with Bronx neighborhoods.

Out[18]:



By examining the clusters below were observed.

The cluster 1 has decent number of italian restaurants with more number of low rated italian restaurants.

The cluster 2 has lower number of italian restaurants but has decent average ratings.

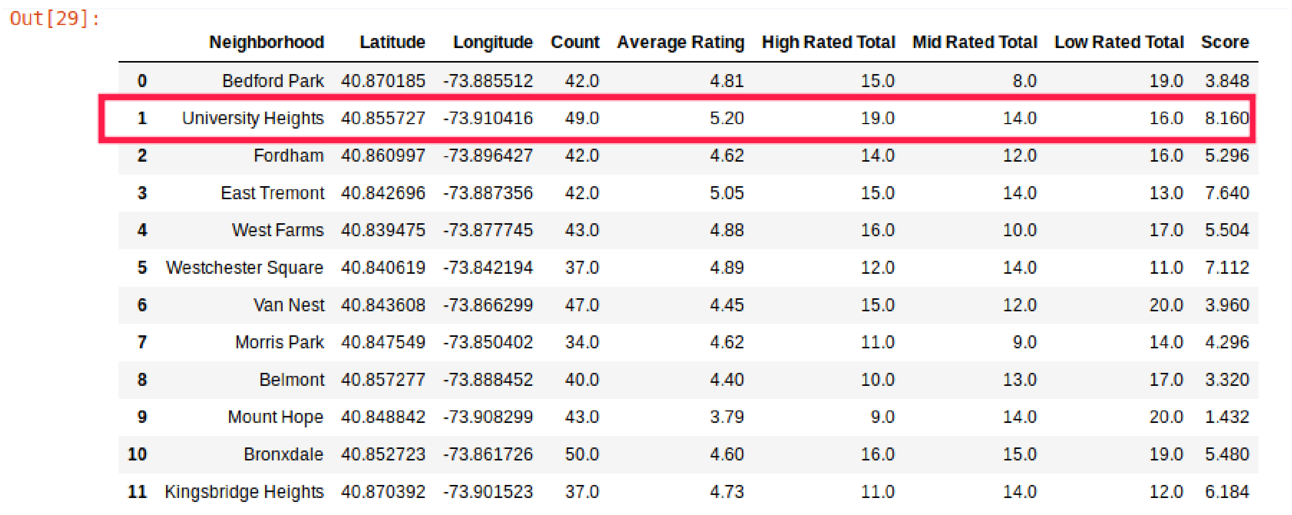
The cluster 3 has high number of italian restaurants and is having decent average ratings.

Easily, The cluster 3 will be recommended to proceed with since our client is looking for a job and it is easy for him to find one if there are plenty of options. Here, we can't only go by the total number of italian restaurants. Because, though the neighborhood has more number of restaurants, the ratings can be poor. So, I am going to find the best neighborhood in cluster 3 by means of following formula in such a way that count holds 20% weight and average rating holds 80% weight.

Restaurant_score = (((High_rated*2) + *Mid_rated* - (*Low_rated**2))**0.2)* + (*Average_rating**0.8)

The best neighborhood is the one which has highest restaurant_score.

Below is the dataframe of cluster 3 which has restaurant scores calculated for each Neighborhoods.

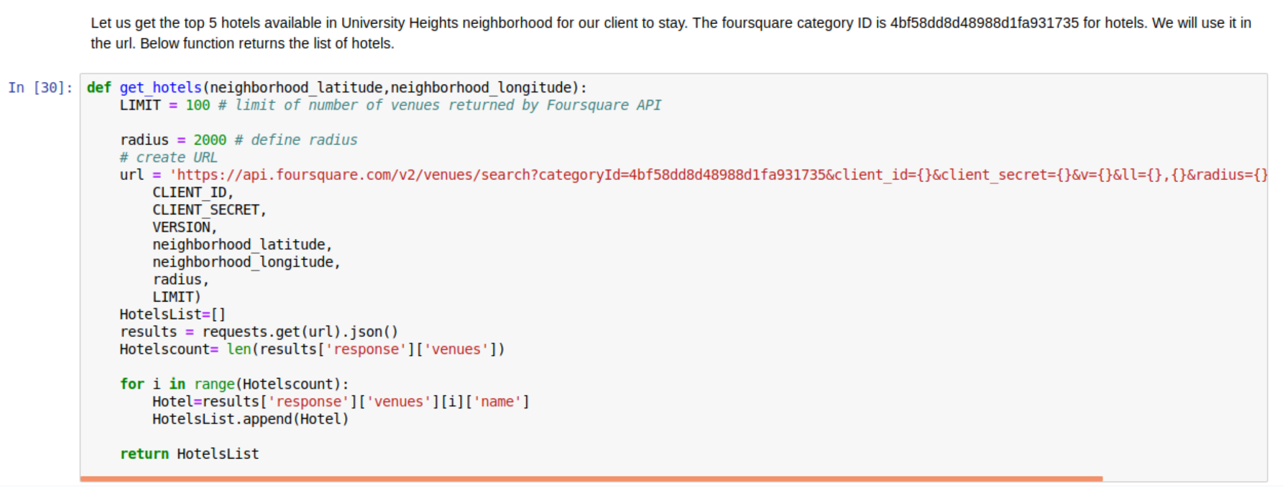


Out[29]:

	Neighborhood	Latitude	Longitude	Count	Average Rating	High Rated Total	Mid Rated Total	Low Rated Total	Score
0	Bedford Park	40.870185	-73.885512	42.0	4.81	15.0	8.0	19.0	3.848
1	University Heights	40.855727	-73.910416	49.0	5.20	19.0	14.0	16.0	8.160
2	Fordham	40.860997	-73.896427	42.0	4.62	14.0	12.0	16.0	5.296
3	East Tremont	40.842696	-73.887356	42.0	5.05	15.0	14.0	13.0	7.640
4	West Farms	40.839475	-73.877745	43.0	4.88	16.0	10.0	17.0	5.504
5	Westchester Square	40.840619	-73.842194	37.0	4.89	12.0	14.0	11.0	7.112
6	Van Nest	40.843608	-73.866299	47.0	4.45	15.0	12.0	20.0	3.960
7	Morris Park	40.847549	-73.850402	34.0	4.62	11.0	9.0	14.0	4.296
8	Belmont	40.857277	-73.888452	40.0	4.40	10.0	13.0	17.0	3.320
9	Mount Hope	40.848842	-73.908299	43.0	3.79	9.0	14.0	20.0	1.432
10	Bronxdale	40.852723	-73.861726	50.0	4.60	16.0	15.0	19.0	5.480
11	Kingsbridge Heights	40.870392	-73.901523	37.0	4.73	11.0	14.0	12.0	6.184

In that, the Neighborhood with the highest restaurant score is University of Heights having 8.16.

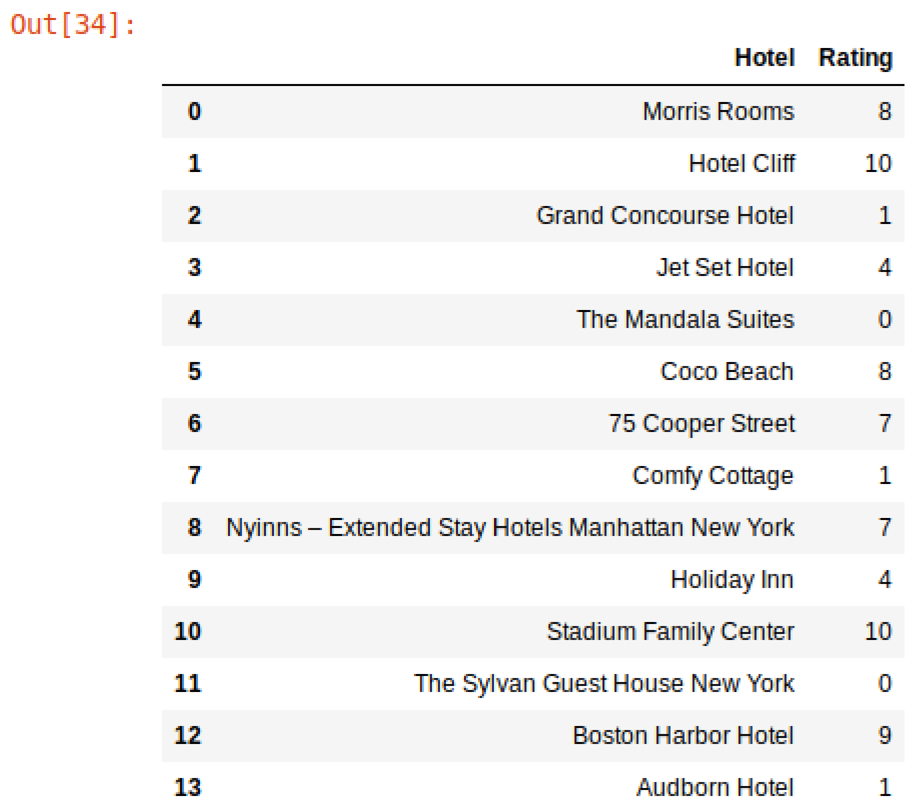
So, In University of Heights Neighborhood, the top 5 Hotels are obtained. For that, we again use Foursquare API to return the list of hotels available like below.



Let us get the top 5 hotels available in University Heights neighborhood for our client to stay. The foursquare category ID is 4bf58dd8d48988d1fa931735 for hotels. We will use it in the url. Below function returns the list of hotels.

```
In [30]: def get_hotels(neighborhood_latitude,neighborhood_longitude):
             LIMIT = 100 # limit of number of venues returned by Foursquare API

             radius = 2000 # define radius
             # create URL
             url = 'https://api.foursquare.com/v2/venues/search?categoryId=4bf58dd8d48988d1fa931735&client_id={}&client_secret={}&v={}&ll={},{}&radius={}
                 CLIENT_ID,
                 CLIENT_SECRET,
                 VERSION,
                 neighborhood_latitude,
                 neighborhood_longitude,
                 radius,
                 LIMIT)
             HotelsList=[]
             results = requests.get(url).json()
             Hotelscount= len(results['response']['venues'])

             for i in range(Hotelscount):
                 Hotel=results['response']['venues'][i]['name']
                 HotelsList.append(Hotel)

             return HotelsList
```

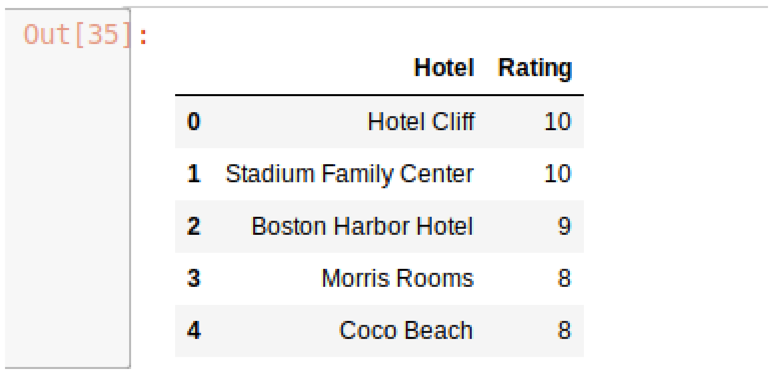
Since the venue details end point is premium call in Foursquare API, I have randomly generated the Hotel ratings and built the below dataframe.



Out[34]:

	Hotel	Rating
0	Morris Rooms	8
1	Hotel Cliff	10
2	Grand Concourse Hotel	1
3	Jet Set Hotel	4
4	The Mandala Suites	0
5	Coco Beach	8
6	75 Cooper Street	7
7	Comfy Cottage	1
8	Nyinns – Extended Stay Hotels Manhattan New York	7
9	Holiday Inn	4
10	Stadium Family Center	10
11	The Sylvan Guest House New York	0
12	Boston Harbor Hotel	9
13	Audborn Hotel	1

Then, the top 5 Hotels were found by sorting.



Out[35]:

	Hotel	Rating
0	Hotel Cliff	10
1	Stadium Family Center	10
2	Boston Harbor Hotel	9
3	Morris Rooms	8
4	Coco Beach	8

Results:

After studying the Bronx's neighborhoods for italian restaurants for our client Michael to get a job, we found that the University Heights is the best neighborhoods having more number of italian restaurants and good ratings. Also, in University Heights neighborhood, Hotel Cliff, Stadium Family Center, Boston Harbor Hotel, Morris Rooms and Coco Beach are the top 5 hotels to stay.

Discussion:

This project helps us to discover more venues about any neighborhood. This can be useful not only is finding a job which is our scope in this project but more ideas like, starting a bar in any neighborhood, to analyze which neighborhood cluster has more tourist attractions, etc. In our case, the venue details were limited since it is a premium call as per Foursquare. It would have been great if it was available as regular call and more lively to group the restaurants and find the hotels. However, the methodology is same either we generate in random or use Foursquare API. Furthermore recommendation will be not only the ratings we can take, it also has to consider the number of ratings were given. Because, if only one person gave a rating, this will be unsufficient amount of data to consider the restaurant as good or bad.

Conclusion:

Using this project, we can analyse whichever Borough and its neighborhoods across the world given we have it's location data and access to Foursquare API. It will be more useful for the people who seek job in any field. I believe it will be the smart way to start the job hunt.