In this IBM Capstone Project, I knew something about what data scientists go through in real life. Objectives of the final assignments were to define a business problem, look for data in the web and, use Foursquare location data to compare different neighborhoods of Boston to figure out which neighborhood is suitable for starting a restaurant business. As prepared for the assignment, I go through the problem designing, data preparation and final analysis section step by step.

1. Background of the Business Problem:

I'm living in Boston now. I really love this city due to its convenience and diversity, There is no doubt that starting a restaurant business in Boston is a great idea. So I try my best to figure out which is the most suitable neighborhood for this kind business.

Target Audience

1. Business personnel who wants to invest or open a restaurant. This analysis will be a comprehensive guide to start or expand restaurants targeting the large pool of office workers in Boston.
2. Freelancer who loves to have their own restaurant as a side business. This analysis will give an idea, how beneficial it is to open a restaurant and what are the pros and cons of this business.
3. New graduates, to find reasonable lunch/breakfast place close to office.
4. Budding Data Scientists, who want to implement some of the most used Exploratory Data Analysis techniques to obtain necessary data, analyze it, and, finally be able to tell a story out of it.

2. Data Preparation:

2.1. Scraping Boston Neighborhoods Table from Wikipedia

I first scrapped the Wikipedia page to get the name and information of neighborhoods in Boston. For this, I've used requests and Beautifulsoup4 library to create a data-frame containing name of the 22 districts of Boston.

First, Get The Names of Major Districts from Wikipedia

In [135]:
```
response_obj = requests.get('https://en.wikipedia.org/wiki/Neighborhoods_in_Boston').text
print (type (response_obj))
```

```
<class 'str'>
```

In [136]:
```
soup = BeautifulSoup(response_obj,'lxml')
```

In [137]:
```
name=soup.select('#mw-content-text > div > div.div-col.columns.column-width')
for name in name:
 name=name.text.strip()
```

In [138]:
```
print(name)
```

After little manipulation, the data-frame is obtained as below

	Name
1	Allston
2	Back Bay
3	Bay Village
4	Beacon Hill
5	Brighton
6	Charlestown
7	Chinatown/Leather District
8	Dorchester (divided for planning purposes into...
9	Downtown
10	East Boston
11	Fenway Kenmore (includes Longwood)
12	Hyde Park
13	Jamaica Plain
14	Mattapan
15	Mission Hill
16	North End
17	Roslindale
18	Roxbury
19	South Boston
20	South End
21	West End
22	West Roxbury

2.2. Getting Coordinates of Major Districts : Geopy Client

Next objective is to get the coordinates of these 22 major districts using geocoder class of Geopy client. Using the code snippet as below —

Get the Coordinates of the Major Districts

```
from geopy.geocoders import Nominatim
geolocator = Nominatim(timeout=50)
df['Neighborhoods_Coord']= df['Name'].apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude) if x else Nor
```

	Name	Neighborhoods_Coord
1	Allston	(42.3554344, -71.1321271)
2	Back Bay	(42.3507067, -71.0797297)
3	Bay Village	(41.4849875, -81.920832)
4	Beacon Hill	(42.3587085, -71.067829)
5	Brighton	(50.8220399, -0.1374061)
6	Charlestown	(43.2387, -72.424622)
7	Chinatown	(40.7164913, -73.9962504)
8	Dorchester	(50.7116772, -2.4422170980612727)
9	Downtown	(37.7875138, -122.407159)
10	East Boston	(42.3750973, -71.0392173)
11	Fenway Kenmore	(42.34422445, -71.09444515776886)
12	Hyde Park	(51.5074889, -0.1622053801825995)
13	Jamaica Plain	(42.3098201, -71.1203299)
14	Mattapan	(42.2675657, -71.0924273)
15	Mission Hill	(29.7147884, -98.1648442)
16	North End	(42.3650974, -71.0544954)
17	Roslindale	(42.2912093, -71.1244966)
18	Roxbury	(41.5568282, -73.3088922)
19	South Boston	(36.6987494, -78.9013987)
20	South End	(42.34131, -71.0772298)
21	West End	(42.3639186, -71.0638993)
22	West Roxbury	(42.2792649, -71.1494972)

As you can see some coordinates are completely wrong, which is due to the names of the districts are written little different than the way they are in this data-frame, so I had to replace these coordinates with values acquired from google search.

After little more playing around with pandas, I could get one well-arranged data-frame as below —

	Name	Latitude	Longitude
1	Allston	42.355434	-71.132127
2	Back Bay	42.350707	-71.079730
3	Bay Village	42.349000	-71.069800
4	Beacon Hill	42.358708	-71.067829
5	Brighton	42.346400	-71.162700
6	Charlestown	42.378200	-71.060200
7	Chinatown	42.350100	-71.062400
8	Dorchester	42.301600	-71.067600
9	Downtown	42.355700	-71.057200
10	East Boston	42.375097	-71.039217
11	Fenway Kenmore	42.344224	-71.094445
12	Hyde Park	42.256500	-71.124100
13	Jamaica Plain	42.309820	-71.120330
14	Mattapan	42.267566	-71.092427
15	Mission Hill	42.329600	-71.106200
16	North End	42.365097	-71.054495
17	Roslindale	42.291209	-71.124497
18	Roxbury	42.315200	-71.091400
19	South Boston	42.338100	-71.047600
20	South End	42.341310	-71.077230
21	West End	42.363919	-71.063899
22	West Roxbury	42.279265	-71.149497

2.3. Average Land Price in Major Districts of Tokyo: Web Scraping

Another factor that can guide us later for deciding which district would be best to open a restaurant is, the average land price of these districts. I'd like to mainly focus on 5 hottest neighborhoods. The data-frame looks as below

	Name	Latitude	Longitude	Median list prices/ft2(USD)
1	Back Bay	42.350707	-71.079730	1256
2	Beacon Hill	42.358708	-71.067829	1245
3	Chinatown	42.350100	-71.062400	1149
4	Downtown	42.355700	-71.057200	1197
5	South End	42.341310	-71.077230	1054

2.4. Using Foursquare Location Data:

Foursquare data is very comprehensive and it powers location data for Apple, Uber etc. For this business problem I have used, as a part of the assignment, the Foursquare API to retrieve information about the popular spots around these 5 Major Districts of Boston. The popular spots returned depends on the highest foot traffic and thus it depends on the time when the call is made. So we may get different popular venues depending upon different time of the day. The call returns a JSON file and we need to turn that into a data-frame. Here I've chosen 100 popular spots for each major districts within a radius of 1 km. Below is the data-frame obtained from the JSON file that was returned by Foursquare —

	District	Dist_Latitude	Dist_Longitude	Venue	Venue_Lat	Venue_Long	Venue_Category
495	South End	42.34131	-71.07723	Flour Bakery + Cafe	42.348289	-71.073542	Bakery
496	South End	42.34131	-71.07723	Sorellina	42.348718	-71.077984	Italian Restaurant
497	South End	42.34131	-71.07723	Thornton's Restaurant & Cafe	42.345288	-71.082010	Diner
498	South End	42.34131	-71.07723	Gaslight Brasserie	42.340974	-71.067347	French Restaurant
499	South End	42.34131	-71.07723	Kung Fu Tea	42.342588	-71.084086	Bubble Tea Shop

3. Visualization and Data Exploration:

3.1. Folium Library and Leaflet Map:

Folium is a python library that can create interactive leaflet map using coordinate data. Since I am interested in restaurants as popular spots first I create a data-frame where the 'Venue_Category' column in previous data-frame contains the word 'Restaurant'. I used the following snippet of code —

```
# Create a Data-Frame out of it to Concentrate Only on Restaurants

Boston_5_Dist_Venues_only_restaurant = Boston_5_Dist_Venues[Boston_5_Dist_Venues['Venue_Category']\
                                                            .str.contains('Restaurant')].reset_index(drop=True)
Boston_5_Dist_Venues_only_restaurant.index = np.arange(1, len(Boston_5_Dist_Venues_only_restaurant)+1)
print ("Shape of the Data-Frame with Venue Category only Restaurant: ", Boston_5_Dist_Venues_only_restaurant.shape)
Boston_5_Dist_Venues_only_restaurant.head()
```

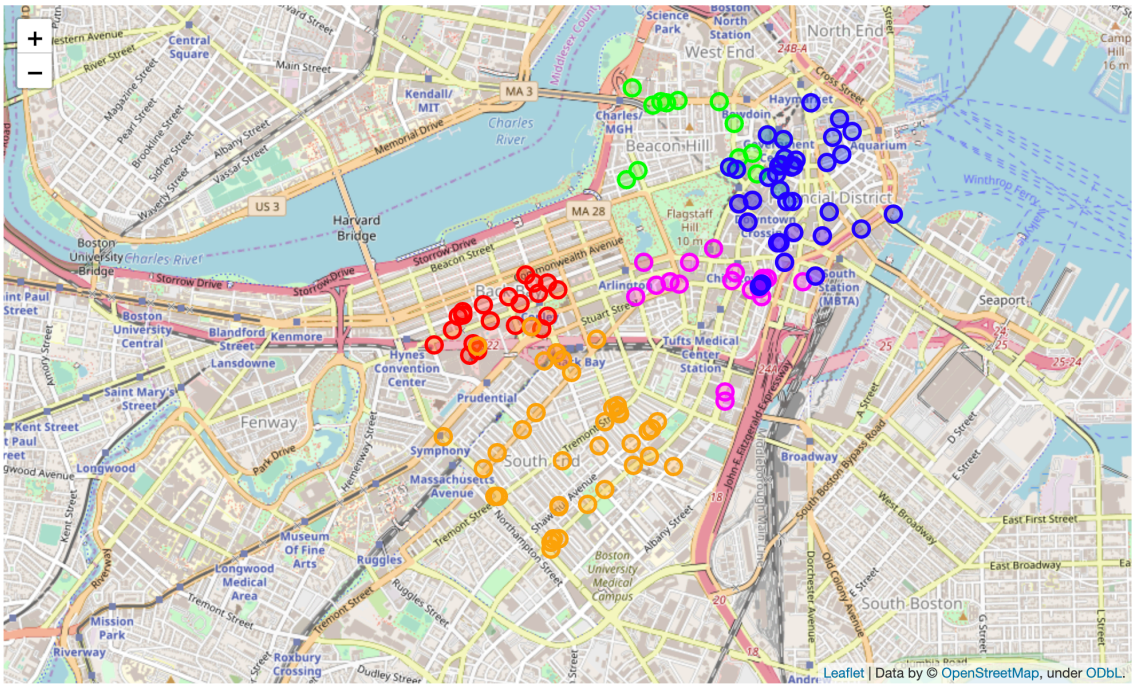
```
Shape of the Data-Frame with Venue Category only Restaurant:  (149, 7)
```

	District	Dist_Latitude	Dist_Longitude	Venue	Venue_Lat	Venue_Long	Venue_Category
1	Back Bay	42.350707	-71.07973	Gre.Co	42.349920	-71.081633	Greek Restaurant
2	Back Bay	42.350707	-71.07973	Lolita Cocina & Tequila Bar	42.350563	-71.077544	Mexican Restaurant
3	Back Bay	42.350707	-71.07973	Saltie Girl Seafood Bar	42.351111	-71.077811	Seafood Restaurant
4	Back Bay	42.350707	-71.07973	Atlantic Fish Company	42.349014	-71.081096	Seafood Restaurant
5	Back Bay	42.350707	-71.07973	Sorellina	42.348718	-71.077984	Italian Restaurant

Next step is to use this data-frame to create a leaflet map with Folium to see the distribution of the most visited restaurants in the 5 major districts.

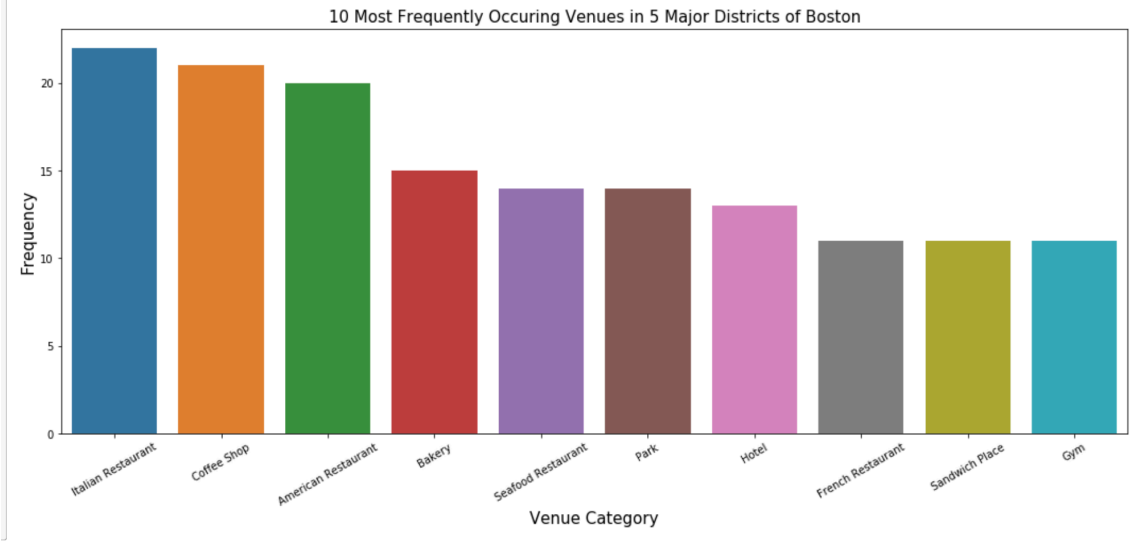
```
## Show in Map the Top Rated Restaruants in the Top 5 Districts

map_restaurants = folium.Map(location=[Boston_latitude, Boston_longitude], zoom_start=11, tiles="openstreetmap",
                             attr="<a href=https://github.com/python-visualization/folium/>Folium</a>")

# set color scheme for the Venues based on the Major Districts
Districts = ['Back Bay', 'Beacon Hill', 'Chinatown', 'Downtown', 'South End']

x = np.arange(len(Districts))

rainbow = ['#00ff00', '#ff00ff','#0000ff','#ffa500' ,'#ff0000']

# add markers to the map
# markers_colors = []
for lat, lon, poi, distr in zip(Boston_5_Dist_Venues_only_restaurant['Venue_Lat'],
                                Boston_5_Dist_Venues_only_restaurant['Venue_Long'],
                                Boston_5_Dist_Venues_only_restaurant['Venue_Category'],
                                Boston_5_Dist_Venues_only_restaurant['District']):
    label = folium.Popup(str(poi) + ' ' + str(distr), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=7,
        popup=label,
        color=rainbow[Districts.index(distr)-1],
        fill=True,
        fill_color=rainbow[Districts.index(distr)-1],
        fill_opacity=0.3).add_to(map_restaurants)

map_restaurants
```

With the code snippet above the leaflet map looks as below



3.2. Exploratory Data Analysis:

There are 133 unique venue categories and Italian Restaurants top the charts as we can see in the plot below —



To know about the top 5 venues of each district we proceed as follows

- Create a data-frame with pandas one hot encoding for the venue categories.
- Use pandas groupby on the District column and obtain the mean of the one-hot encoded venue categories.
- Transpose the data-frame at step 2 and arrange in descending order.

Let's see the code snippet below —

```
### Use One Hot Encoding to Get More Information about the Venue Categories
Boston_5_Dist_Venues_onehot = pd.get_dummies(Boston_5_Dist_Venues[['Venue_Category']], prefix="", prefix_sep="")

### add district column back to dataframe
Boston_5_Dist_Venues_onehot['District'] = Boston_5_Dist_Venues['District']
### move district column to the first column
fixed_columns = [Boston_5_Dist_Venues_onehot.columns[-1]] + list(Boston_5_Dist_Venues_onehot.columns[:-1])
Boston_5_Dist_Venues_onehot = Boston_5_Dist_Venues_onehot[fixed_columns]
###Boston_5_Dist_Venues_onehot.head(3)
```

```
Boston_5_Dist_Venues_Grouped = Boston_5_Dist_Venues_onehot.groupby('District').mean().reset_index()
Boston_5_Dist_Venues_Grouped.index = np.arange(1, len(Boston_5_Dist_Venues_Grouped)+1)
Boston_5_Dist_Venues_Grouped
```

	District	Accessories Store	American Restaurant	Arepa Restaurant	Art Museum	Asian Restaurant	Athletics & Sports	Bakery	Bank	Bar	...	Theater	Tiki Bar	Tourist Information Center	Trail	Vegetarian / Vegan Restaurant	Viet Re
1	Back Bay	0.01	0.05	0.00	0.00	0.00	0.01	0.01	0.01	0.00	...	0.00	0.00	0.00	0.01	0.00	
2	Beacon Hill	0.00	0.05	0.00	0.00	0.01	0.01	0.03	0.00	0.02	...	0.02	0.00	0.01	0.00	0.00	
3	Chinatown	0.00	0.01	0.00	0.00	0.05	0.00	0.05	0.00	0.00	...	0.04	0.01	0.01	0.00	0.01	
4	Downtown	0.00	0.03	0.00	0.01	0.04	0.01	0.02	0.00	0.02	...	0.00	0.00	0.00	0.00	0.01	
5	South End	0.01	0.06	0.01	0.00	0.01	0.00	0.04	0.00	0.02	...	0.02	0.00	0.00	0.01	0.00	

5 rows × 134 columns

```
num_top_venues = 5

for places in Boston_5_Dist_Venues_Grouped['District']:
    print("%%%%%%%%%"+places+"%%%%%%%%%")
    temp = Boston_5_Dist_Venues_Grouped[Boston_5_Dist_Venues_Grouped['District'] == places].T.reset_index()
    temp.columns = ['Venue','Freq']
    temp = temp.iloc[1:]
    temp['Freq'] = temp['Freq'].astype(float)
    temp = temp.round({'Freq': 2})
    print(temp.sort_values('Freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

The above code outputs top 5 venues of each district —

```
%%%%%%%%%Back Bay%%%%%%%%
                 Venue  Freq
0  American Restaurant  0.05
1          Coffee Shop  0.04
2   Seafood Restaurant  0.04
3       Clothing Store  0.04
4                  Spa  0.04
```

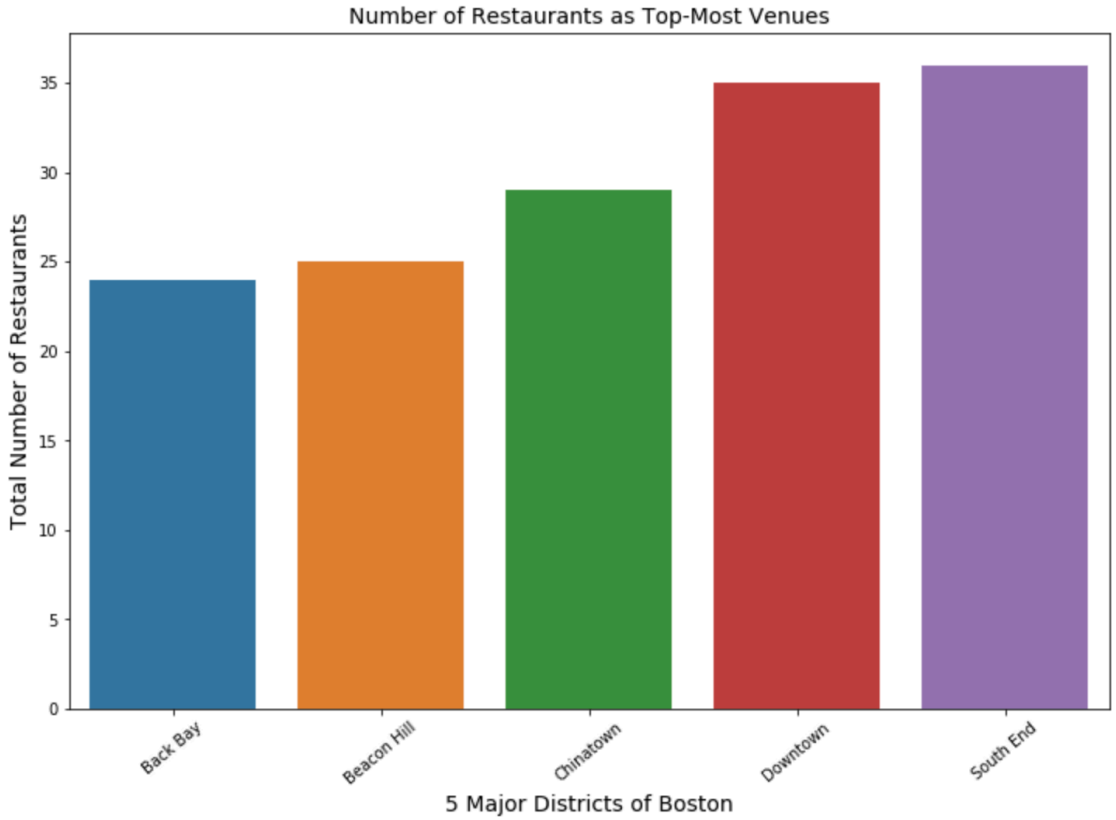
```
%%%%%%%%%Beacon Hill%%%%%%%%
                 Venue  Freq
0   Italian Restaurant  0.06
1  American Restaurant  0.05
2                Hotel  0.04
3          Pizza Place  0.04
4               Bakery  0.03
```

```
%%%%%%%%%Downtown%%%%%%%%
                       Venue  Freq
0                Coffee Shop  0.06
1              Historic Site  0.06
2         Seafood Restaurant  0.04
3  New American Restaurant  0.04
4             Sandwich Place  0.04
```

```
%%%%%%%%%Chinatown%%%%%%%%
                 Venue  Freq
0    Asian Restaurant  0.05
1  Chinese Restaurant  0.05
2              Bakery  0.05
3         Coffee Shop  0.04
4  Italian Restaurant  0.04
```

```
%%%%%%%%%South End%%%%%%%%
                   Venue  Freq
0   Italian Restaurant  0.06
1  American Restaurant  0.06
2          Coffee Shop  0.05
3             Wine Bar  0.04
4               Bakery  0.04
```

From the several data-frames that I had to create for exploratory data analysis, using one of them, I've plotted which district has restaurants among the most frequently visited places and, South End comes on top with 36 restaurants.



Once we get quite a broad overview of the different types of venues and specially restaurants around 5 major districts of Boston, it is time to use clustering the districts using K-Means.

4. Clustering the Districts

Finally, we try to cluster these 5 districts based on the venue categories and use K-Means clustering. So our expectation would be based on the similarities of venue categories, these districts will be clustered. I have used the code snippet below

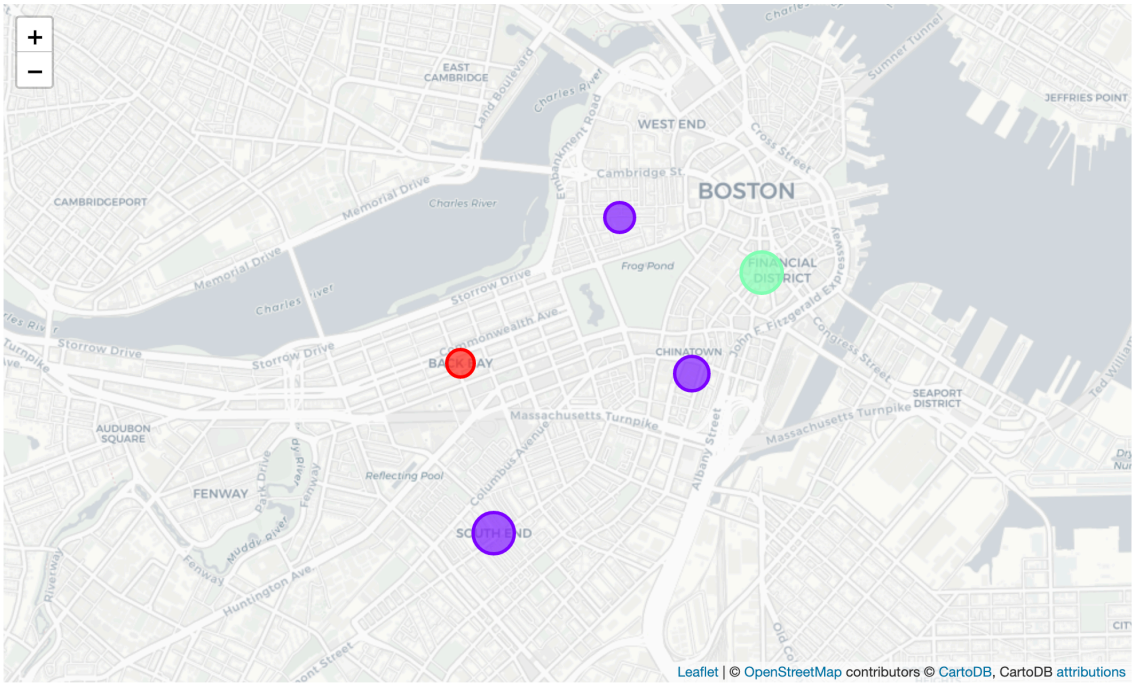
```
# set number of clusters
kclusters = 3

Boston_grouped_clustering = Boston_5_Dist_Venues_Grouped.drop('District', 1)

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

# check cluster labels generated for each row in the dataframe
print ("Check the 5 Cluster labels :",  kmeans.labels_[0:5])
```

```
Check the 5 Cluster labels : [0 1 1 2 1]
```

```
# add clustering labels

District_top10_venues_sorted.insert(0, 'Cluster Label', kmeans.labels_)
District_top10_venues_sorted
```

	Cluster Label	District	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
1	0	Back Bay	American Restaurant	Hotel	Spa	Seafood Restaurant	Clothing Store	Coffee Shop	Gym	Gym / Fitness Center	Ice Cream Shop	Restaurant
2	1	Beacon Hill	Italian Restaurant	American Restaurant	Hotel	Pizza Place	Park	Bakery	French Restaurant	Steakhouse	Coffee Shop	Gym / Fitness Center
3	1	Chinatown	Chinese Restaurant	Bakery	Asian Restaurant	Coffee Shop	Theater	Italian Restaurant	Performing Arts Venue	Seafood Restaurant	Spa	Sandwich Place
4	2	Downtown	Historic Site	Coffee Shop	Italian Restaurant	New American Restaurant	Sandwich Place	Seafood Restaurant	Asian Restaurant	Park	Gym / Fitness Center	Hotel
5	1	South End	American Restaurant	Italian Restaurant	Coffee Shop	Wine Bar	French Restaurant	Bakery	Park	Mexican Restaurant	Café	Gym

We can represent these 3 clusters in a leaflet map using Folium library as below



5. Results and Discussion:

We reached at the end of the analysis, where we got a sneak peak of the 5 major neighborhoods of Boston and, as the business problem started with benefits and drawbacks of opening a restaurant in one of the busiest districts, the data exploration was mostly concentrated on the restaurants. I have used data from web resources like Wikipedia, python libraries like Geopy, and Foursquare API, to set up a very realistic data-analysis scenario. We have found out that —

- Italian restaurants top the charts of most common venues in the 5 districts.
- Back Bay, Beacon Hill, Chinatown and South End districts are dominated by restaurants as the most common venue whereas Downtown area is dominated by historic sites and coffee shops as most common venues.
- South End has maximum number of restaurants as the most common venue whereas Back Bay area has the least.
- Since the clustering was based only on the most common venues of each district, Beacon Hill, Chinatown and South End are under the same cluster. Back Bay and Downtown are separate cluster as, Historic sites, coffee shop and hotels stand out as the common venue (with a very high frequency).

According to this analysis, Downtown area will provide least competition for an upcoming restaurant as historic sites and coffee shops are the most common venue in this area and, the frequency of restaurants as common venue are very low compared to the remaining districts. Also seen from the web-scrapped data, the average land price in and around Downtown is relatively cheaper compared to the districts close to central Boston. So, definitely this region could potentially be a target for starting quality restaurants. Some drawbacks of this analysis are — the clustering is completely based on the most common venues obtained from Foursquare data. Since land price, distance of the venues from closest stations and number of potential customers could all play a major role and thus, this analysis is

definitely far from being conclusory. However, it certainly gives us some very important preliminary information on possibilities of opening restaurants around the major districts of Boston. Also, another pitfall of this analysis could be consideration of only 5 major districts of Boston, taking into account of all the areas in Boston would give us an even more realistic picture. Furthermore, this results also could potentially vary if we use some other clustering techniques like DBSCAN.

6. Conclusion

To conclude this project, We have got a small glimpse of how real life data-science projects look like. I've made use of some frequently used python libraries to scrap web-data, use Foursquare API to explore the major districts of Boston and saw the results of segmentation of districts using Folium leaflet map. Potential for this kind of analysis in a real-life business problem is discussed in great detail. Also, some of the drawbacks and chance for improvements to represent even more realistic pictures are mentioned. Finally, since my analysis were mostly concentrated on the possibilities of opening a restaurant targeting the huge pool of office workers, some of the results obtained are surprisingly exactly what I have expected after staying 2 years in Boston. Hopefully, this kind of analysis will provide you initial guidance to take more real-life challenges using data-science.