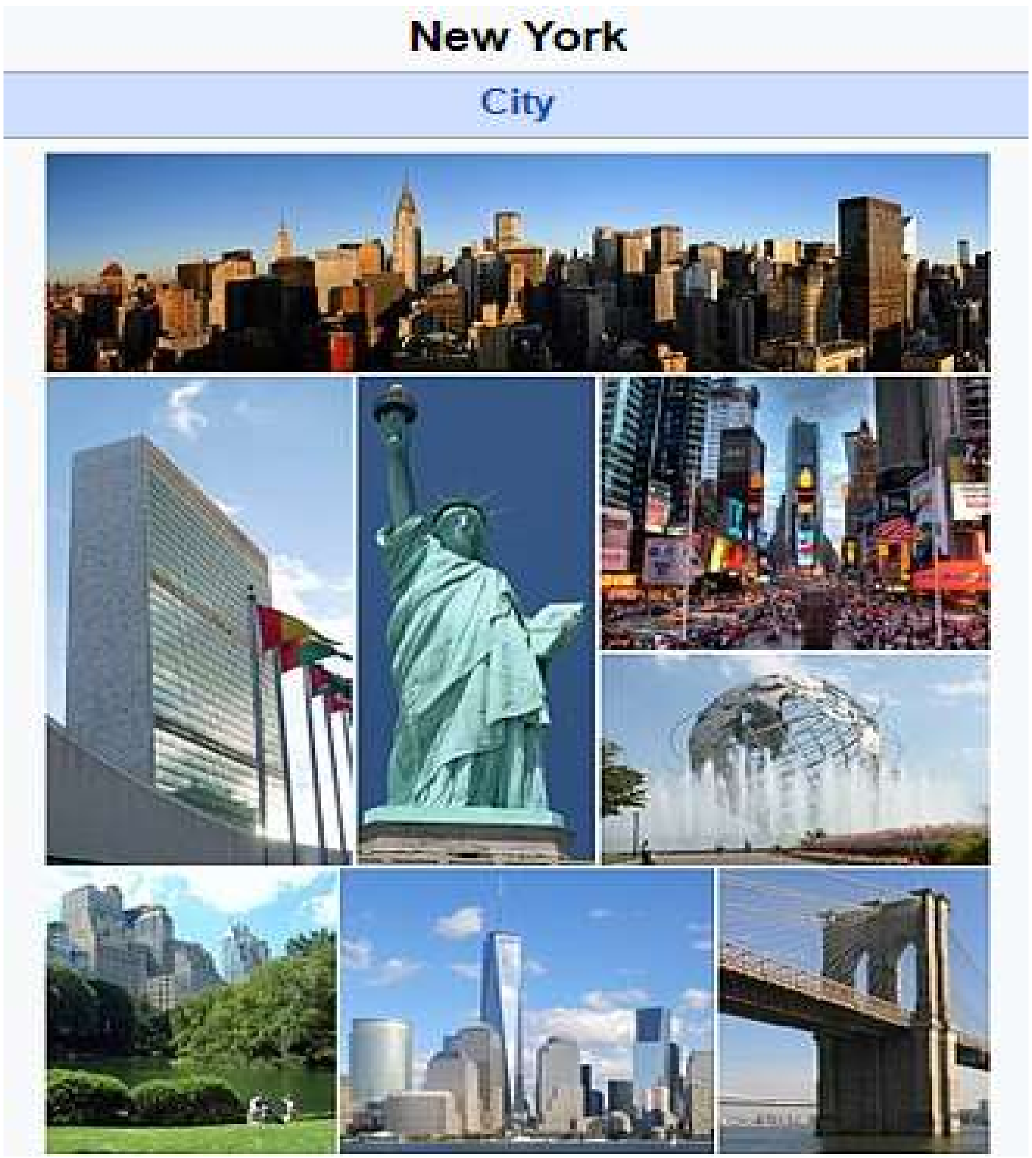
IBM – COURSERA
DATA SCIENCE SPECIALIZATION

CAPSTONE PROJECT – FINAL REPORT
The Battle of the Neighborhoods



SAJITH M P – Sept 2019

Table of content:

Contents

INTRODUCTION

The City of New York, usually called either New York City (NYC) or simply New York (NY), is the most populous city in the United States. With an estimated 2018 population of 8,398,748 distributed over a land area of about 302.6 square miles (784 km^2),

It is diverse and is the financial capital of USA. It is multicultural. It provides lot of business opportunities and business friendly environment. It has attracted many different players into the market. It is a global hub of business and commerce. The city is a major center for banking and finance, retailing, world trade, transportation, tourism, real estate, new media, traditional media, advertising, legal services, accountancy, insurance, theater, fashion, and the arts in the United States. This also means that the market is highly competitive. As it is highly developed city so cost of doing business is also one of the highest. Thus, any new business venture or expansion needs to be analyzed carefully. The insights derived from analysis will give good understanding of the business environment, which help in strategically targeting the market. This will help in reduction of risk and better control on the Return on Investment

New York is also the most densely populated major city in the United States. Located at the southern tip of the state of New York.A global power city, New York City has been described as the cultural, financial, and media capital of the world, and exerts a significant impact upon commerce, entertainment, research, technology, education, politics, tourism, art, fashion, and sports.

NY is split up into five boroughs: the Bronx, Brooklyn, Manhattan, Queens, and Staten Island. Each borough has the same boundaries as a county of the state.

The five boroughs: 1: Manhattan, 2: Brooklyn, 3: Queens, 4: The Bronx, 5: Staten Island

BUSINESS PROBLEM

The City of New York is famous for it's excellent cuisine. It's food culture includes an array of international cuisines influenced by the city's immigrant history. Italian & Indian restaurants have become so popular in the United States now it seems that there is one on every corner, not only in major cities but also in smaller cities. One of my friends who is thinking of starting a restaurant in the NY neighborhood, consulted with me to get some analysis done with the all-possible data available. Manhattan being the costliest place, it was decided to compare rest of the boroughs and pick one of the most suitable neighborhoods with in the shortlisted boroughs. Based on the data analysis, it is expected to logically conclude which restaurant type (Italian Or Indian) and its recommended location. All the choices to be rationalized with the data analysis & it helps to distinguish the selections, securing long-term success.

Overall Problem Statement can be broken into the following

- Exploring the Boroughs in NY and narrow down to one.
- Explore the Venues in the neighborhoods across that specific Borough
- Narrow down to handful of neighborhoods and then deep dive into the current Restaurants & Hotels landscape across those.
- Venue clustering by filtered neighborhoods and analyze the best choice of the restaurant and the best fit location.

TARGET AUDIENCE

Any Business Entrepreneurs or Companies who would like to start a Restaurant business in NewYork. The objective is to narrow down to best possible, affordable neighborhood to start a restaurant. The model also look at picking a type of restaurants from multiple choices like Italian Vs Indian. The Solution is expected to rationalize the choices baked up with data and its analysis. For this project, all boroughs except Manhattan being considered due to high cost.

SOLUTION DESIGN APPROACH

Solution is approached in seven steps as listed below

STEP 1: Pull all the boroughs & the respective neighborhood details of the New York data using newyork_data.json.['newyork_data.json' - https://cocl.us/new_york_dataset]

STEP 2: Narrowing down to one of the Boroughs – Basis of Population/Density analysis-

on the data available in Web.

https://en.wikipedia.org/wiki/Demographics_of_New_York_City"

STEP 3: Deep Dive into the shortlisted Borough from Step 2 Using **FourSquare APIs**

STEP 4: Explore Venues across the neighborhoods in that Borough & Narrow down to handful of it based on larger number of Venues Vs less number of Restaurants +Hotels

STEP 5: Deep Dive into the shortlisted neighborhoods using, **Word Cloud**, **Means of frequency** of each category of Restaurants & identifying the **Top5 Common** Restaurants/Hotels

STEP 6: Clustering the neighborhood using **K-means** & identifying the locations on the Map.

STEP 7: Concluding the Choices of Restaurants & Locations basis of the data analysis in Step

SUCCESS CRITERIA

The success criteria of this project will be a good recommendation of borough/neighborhood for the choice of a restaurant, to the Stakeholder from the Target Audience. All choices and recommendations should be rationalized with the data analysis and inferences made.

DATA

One City will be analyzed in this project : NewYork USA .

Data sources that's been analyzed in the projects are

Data1 : NewYork has a total of 5 boroughs and 306 neighborhoods. In order to segment the neighborhoods and explore them, we will essentially need a dataset that contains the 5 boroughs and the neighborhoods that exist in each borough as well as the latitude and longitude coordinates of each neighborhood.

Data Source : newyork_data.json' https://cocl.us/new_york_dataset

Data 2 : To Narrow down to one of the boroughs , basis of population /density analysis of the data available in Wikipedia

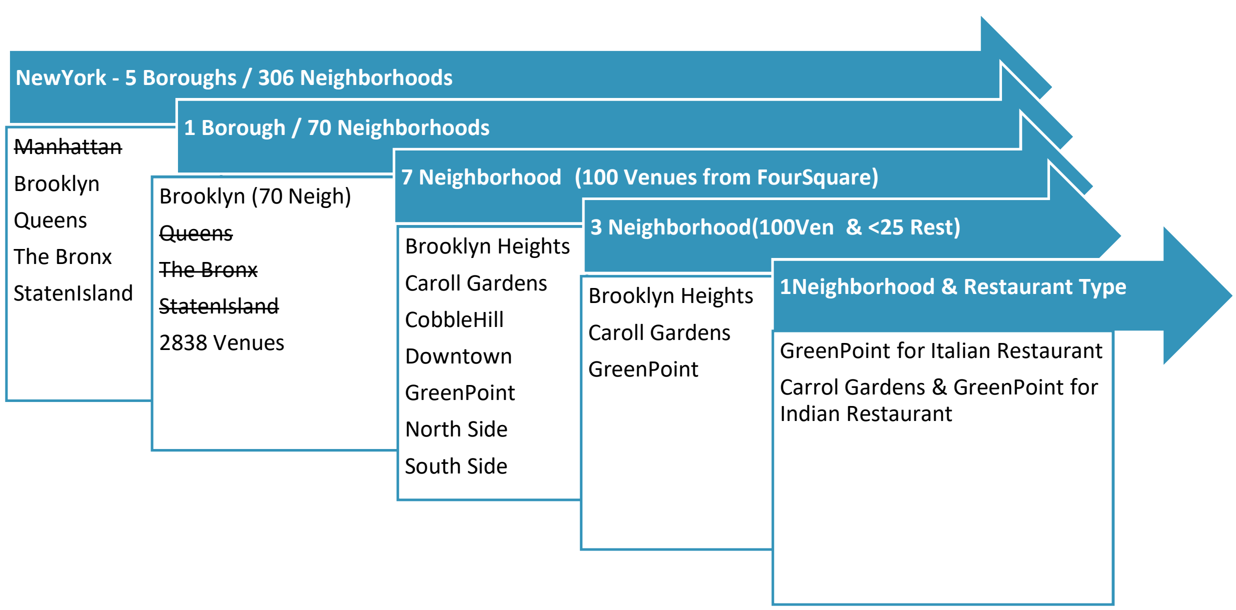
Data Source : https://en.wikipedia.org/wiki/Demographics_of_New_York_City

Data3 : Exploring the neighborhoods in one of the shortlisted boroughs using FourSquare APIS

METHODOLOGY

ANALYTIC APPROACH

New York city neighborhood has a total of 5 boroughs and 306 neighborhoods. In this project we excluded Manhattan due to high cost and focus only on the rest of the 4 boroughs. From 300 + Neighborhoods across all the boroughs, we have applied the following analytic approach to narrow down to 3 Neighborhood in Brooklyn through multiple data exploratory analysis as explained below.



DATA EXPLORATORY ANALYSIS

Solution is approached in seven-step data exploratory analysis as explained below

STEP 1: Pull all the boroughs & the respective neighborhood details of the New York data using newyork_data.json.['newyork_data.json' - https://cocl.us/new_york_dataset]

```
In [2]: !wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
        print('Data downloaded!')

        Data downloaded!

In [3]: with open('newyork_data.json') as json_data:
            newyork_data = json.load(json_data)

In [4]: NYnghhood_data = newyork_data['features']
        NYnghhood_data[0]

Out[4]: {'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
```

```
In [7]: NYneighborhoods.head()
        print('The dataframe has {} boroughs and {} neighborhoods.'.format(
                len(NYneighborhoods['Borough'].unique()),
                NYneighborhoods.shape[0]
            )
        )
        NYneighborhoods.head()
        # STEP 1 Completes

        The dataframe has 5 boroughs and 306 neighborhoods.
```

Out[7]:

	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

STEP 2: Narrowing down to one of the Boroughs – Basis of Population/Density analysis- on the data available in Web.

https://en.wikipedia.org/wiki/Demographics_of_New_York_City"

```
Table_array1string = StringIO(Table_array1string)
df = pd.read_csv(Table_array1string, sep="\n")
df.drop([45,46,47],axis=0,inplace=True)
df = pd.DataFrame(df.Heading.values.reshape(-1, 9), columns=['Borough', 'County', 'Population Est(2017)',"GDP-USD-Billions","Per-Capita-US
df.shape
df
```

Out[9]:

	Borough	County	Population Est(2017)	GDP-USD-Billions	Per-Capita-USD	LandArea-SqMile	LandArea-SqKM	Density-SqMiles	Density-SqMiles
0	The Bronx	Bronx	1,471,160	28.787	19,570	42.10	109.04	34,653	13,231
1	Brooklyn	Kings	2,648,771	63.303	23,900	70.82	183.42	37,137	14,649
2	Manhattan	New York	1,664,727	629.682	378,250	22.83	59.13	72,033	27,826
3	Queens	Queens	2,358,582	73.842	31,310	108.53	281.09	21,460	8,354
4	Staten Island	Richmond	479,458	11.249	23,460	58.37	151.18	8,112	3,132

STEP 2 - Narrowing down to One of the Boroughs - Brooklyn Basis of Population/Density

```
In [8]: import pandas as pd
        import requests
        from bs4 import BeautifulSoup
        from io import StringIO
        # Webscrapping the URL
        url = "https://en.wikipedia.org/wiki/Demographics_of_New_York_City"
        page = requests.get(url)
        print(page.status_code)
        soup = BeautifulSoup(page.text,"html.parser")

        200

In [9]: # READ Table
        Table_array = []
        Table_text_element = soup.find_all( class_= "wikitable sortable")
        #print (Table_text_element[0])
        Table_text_element=Table_text_element[0]
        for row in Table_text_element.find_all('tr'):
```

STEP 3: Deep Dive into the shortlisted Borough from Step 2 Using **FourSquare APIs**

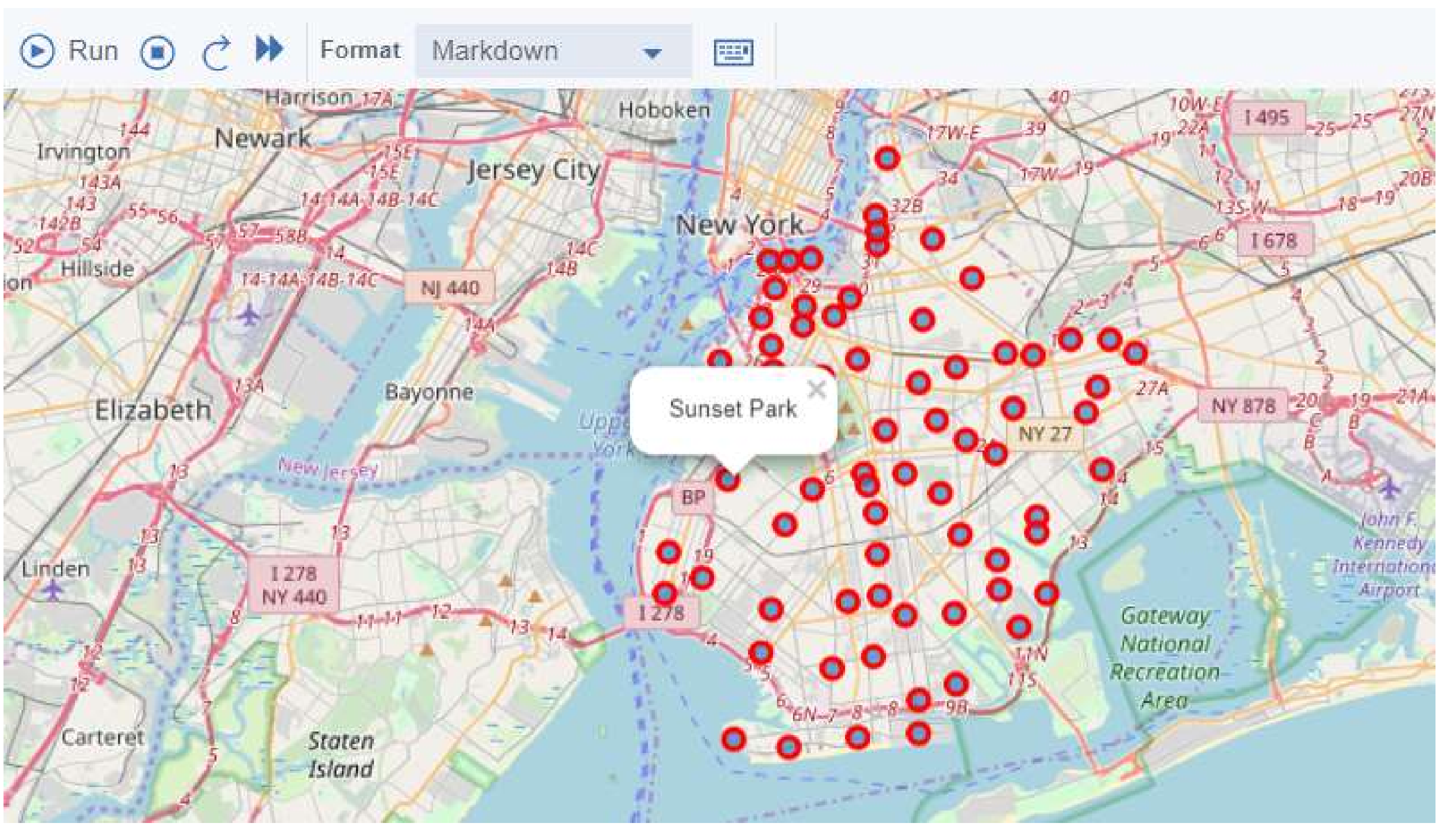
In [10]:
```python
# STEP 3 - STARTS
brooklyn_data = NYneighborhoods[NYneighborhoods['Borough'] == 'Brooklyn'].reset_index(drop=True)
brooklyn_data.head()
```

Out[10]:

	Borough	Neighborhood	Latitude	Longitude
0	Brooklyn	Bay Ridge	40.625801	-74.030621
1	Brooklyn	Bensonhurst	40.611009	-73.995180
2	Brooklyn	Sunset Park	40.645103	-74.010316
3	Brooklyn	Greenpoint	40.730201	-73.954241
4	Brooklyn	Gravesend	40.595260	-73.973471

In [11]:
```python
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(brooklyn_data['Borough'].unique()),
        brooklyn_data.shape[0]
    )
)
```

```
The dataframe has 1 boroughs and 70 neighborhoods.
```



STEP 4: Explore Venues across the neighborhoods in that Borough & Narrow down to handful of it based on larger number of Venues Vs less number of Restaurants +Hotels

In [18]:
```
LIMIT = 250 # limit of number of venues returned by Foursquare API
radius = 500 # define radius
# create URL
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url # display URL
```

Out[18]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=AV2RXHWVXPVA2W4UAFKRNVMEINKR3U2RAQYF2XBVARV3U0PG&client_s
LIXKSCVWNOQ2HM130004DB0KOQX5MHHXEB&v=20180605&ll=40.62580106501065 6,-74.03062069353813&radius=500&limit=250'
```

In [19]:
```
brooklynresults = requests.get(url).json()
brooklynresults
```

Out[19]:

```
{'meta': {'code': 200, 'requestId': '5d90499ca87921002ccf0921'},
 'response': {'suggestedFilters': {'header': 'Tap to show:',
   'filters': [{'name': 'Open now', 'key': 'openNow'},
    {'name': '$-$$$$', 'key': 'price'}]},
  'headerLocation': 'Bay Ridge',
  'headerFullLocation': 'Bay Ridge, Brooklyn',
```


```
nearby_venues.insert(0, 'neighborhood', 'Bay Ridge')

nearby_venues.head(50)
```

Out[20]:

	neighborhood	name	categories	lat	lng
0	Bay Ridge	Pilo Arts Day Spa and Salon	Spa	40.624748	-74.030591
1	Bay Ridge	Bagel Boy	Bagel Shop	40.627896	-74.029335
2	Bay Ridge	Cocoa Grinder	Juice Bar	40.623967	-74.030863
3	Bay Ridge	Pegasus Cafe	Breakfast Spot	40.623168	-74.031186
4	Bay Ridge	Ho' Brah Taco Joint	Taco Place	40.622960	-74.031371
5	Bay Ridge	Brooklyn Market	Grocery Store	40.626939	-74.029948
6	Bay Ridge	Georgian Dream Cafe and Bakery	Caucasian Restaurant	40.625586	-74.030196
7	Bay Ridge	The Bookmark Shoppe	Bookstore	40.624577	-74.030562
8	Bay Ridge	Karam	Middle Eastern Restaurant	40.622931	-74.028316
9	Bay Ridge	Mimi Nails	Spa	40.622571	-74.031477
10	Bay Ridge	A.L.C. Italian Grocery	Grocery Store	40.623051	-74.031224
11	Bay Ridge	RED OAK Restaurant & Bar & Hookah Lounge	Hookah Bar	40.625447	-74.030246

In [24]:
```
print(brooklyn_venues.shape)
brooklyn_venues.head()
```

(2838, 7)

Out[24]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Bay Ridge	40.625801	-74.030621	Pilo Arts Day Spa and Salon	40.624748	-74.030591	Spa
1	Bay Ridge	40.625801	-74.030621	Bagel Boy	40.627896	-74.029335	Bagel Shop
2	Bay Ridge	40.625801	-74.030621	Cocoa Grinder	40.623967	-74.030863	Juice Bar
3	Bay Ridge	40.625801	-74.030621	Pegasus Cafe	40.623168	-74.031186	Breakfast Spot
4	Bay Ridge	40.625801	-74.030621	Ho' Brah Taco Joint	40.622960	-74.031371	Taco Place

FILTERING NEIGHBORHOODS HAVING 100 VENUES

In [25]:
```
brooklyn_venues_grt100 = brooklyn_venues.groupby('Neighborhood').count()
brooklyn_Neigh_grt100 = brooklyn_venues_grt100.loc[brooklyn_venues_grt100["Venue"] == 100].reset_index()
#brooklyn_Neigh_grt100
brooklyn_venues = brooklyn_venues.loc[brooklyn_venues["Neighborhood"].isin(brooklyn_Neigh_grt100["Neighborhood"])]
#df.loc[df['column_name'] == some_value]
```

In [26]:
```
print('There are {} uniques categories.'.format(len(brooklyn_venues['Venue Category'].unique())))
brooklyn_venues
```

There are 180 uniques categories.

Out[26]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
155	Greenpoint	40.730201	-73.954241	Karczma	40.730102	-73.955092	Polish Restaurant
156	Greenpoint	40.730201	-73.954241	Oxomoco	40.729981	-73.955460	Mexican Restaurant
157	Greenpoint	40.730201	-73.954241	goodyoga	40.730010	-73.956167	Yoga Studio
158	Greenpoint	40.730201	-73.954241	Sunshine Laundry & Pinball Emporium	40.729318	-73.953564	Laundry Service
159	Greenpoint	40.730201	-73.954241	Early	40.732069	-73.954721	Café
160	Greenpoint	40.730201	-73.954241	Friducha	40.731512	-73.954281	Mexican Restaurant
161	Greenpoint	40.730201	-73.954241	Brooklyn Craft Company	40.730357	-73.953139	Arts & Crafts Store

FOCUSSING ON THE "RESTAURANTS & HOTELS" IN THE VENUE CATEGORY

```
brooklyn_venues_final.head()
brooklyn_venues_final_filter=brooklyn_venues_final.drop(["Neighborhood Latitude","Neighborhood Longitude","Venue
brooklyn_venues_final_filter
#brooklyn_venues_final_4Kmeans=brooklyn_venues_final.drop(["Venue Latitude","Venue Longitude","count"],axis=1)
#brooklyn_venues_final_4Kmeans.head()
```

Out[27]:

	Neighborhood	Venue	Venue Category	count	Venue Type
0	Greenpoint	Karczma	Polish Restaurant	1	Restaurant
1	Greenpoint	Oxomoco	Mexican Restaurant	1	Restaurant
2	Greenpoint	Friducha	Mexican Restaurant	1	Restaurant
3	Greenpoint	Citroën	French Restaurant	1	Restaurant
4	Greenpoint	Chiko	Sushi Restaurant	1	Restaurant
5	Greenpoint	Archestratus Books & Foods	Restaurant	1	Restaurant
6	Greenpoint	Jungle Cafe	Vegetarian / Vegan Restaurant	1	Restaurant
7	Greenpoint	Adelina's	Italian Restaurant	1	Restaurant
8	Greenpoint	Đi ăn Đi	Vietnamese Restaurant	1	Restaurant
9	Greenpoint	Esme	New American Restaurant	1	Restaurant
10	Greenpoint	Sakura 6	Sushi Restaurant	1	Restaurant

STEP 5: Deep Dive into the shortlisted neighborhoods using, **Word Cloud**, **Means of frequency** of each category of Restaurants & identifying the **Top5 Common** Restaurants/Hotels

a) WORD CLOUD to look at the Restaurant Types among the Seven Neighborhoods

```
wordcloud = WordCloud(max_font_size=50, max_words=100,stop
print("\n"+ color.RED + " Analyziing {} Neighborhood ".for
# display the cloud
fig = plt.figure()
fig.set_figwidth(7)
fig.set_figheight(9)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

--------- Analyziing Brooklyn Heights Neighborhood ---------


```
wordcloud = WordCloud(max_font_size=50, max_words=100,stopwords=s
print("\n"+ color.RED + " Analyziing {} Neighborhood ".format(wor
# display the cloud
fig = plt.figure()
fig.set_figwidth(7)
fig.set_figheight(9)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

--------- Analyziing Carroll Gardens Neighborhood ---------

b) PIVOT to Look at the Less Restaurants/Hotels Venues with in the shortlisted 7 Neighborhoods

In [150]:
```
pivot = pd.pivot_table(brooklyn_venues_final_filter,index=["Neighborhood","Venue Type"], values=["count"],aggfunc=np.sum)
pivot
```

Out[150]:

Neighborhood	Venue Type	count
Brooklyn Heights	Restaurant	22
Carroll Gardens	Restaurant	24
Cobble Hill	Restaurant	25
Downtown	Hotel	2
	Restaurant	28
Greenpoint	Hotel	1
	Restaurant	23
North Side	Hotel	1
	Restaurant	24
South Side	Restaurant	31

c) Grouping the Neighborhood Using Means of Frequency of each Category

Grouping the Neighbourghood using means of Frequency of each category

[77]:
```
brooklyn_grouped = brooklyn_onehot.groupby('Neighborhood').mean().reset_index()
brooklyn_grouped.head(10)
```

Out[77]:

	Neighborhood	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Caribbean Restaurant	Chinese Restaurant	Cuban Restaurant	Dumpling Restaurant	Eastern European Restaurant	Ethiopian Restaurant	Falafel Restaurant	Fast Food Restaurant
0	Brooklyn Heights	0.090909	0.000000	0.000000	0.090909	0.000000	0.045455	0.000000	0.000000	0.045455	0.000000	0.045455	0.045455
1	Carroll Gardens	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.041667	0.041667	0.000000	0.000000	0.000000	0.000000
2	Cobble Hill	0.038462	0.000000	0.038462	0.000000	0.000000	0.038462	0.000000	0.038462	0.000000	0.038462	0.038462	0.000000
3	Downtown	0.000000	0.000000	0.000000	0.066667	0.033333	0.066667	0.033333	0.000000	0.000000	0.000000	0.000000	0.000000
4	Greenpoint	0.041667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.041667	0.000000
5	North Side	0.115385	0.038462	0.038462	0.038462	0.000000	0.076923	0.000000	0.038462	0.000000	0.000000	0.000000	0.000000
6	South Side	0.125000	0.031250	0.000000	0.000000	0.000000	0.093750	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

d) Exploring Each Neighborhood along with top 5 Common Restaurants/Hotels

Exploring each Neighbourhood along with the top 5 Common Restaurants/Hotels

n [85]:
```
num_top_RestHtl = 10

for Nghhood in brooklyn_grouped['Neighborhood']:
    print("----"+Nghhood+"----")
    temp = brooklyn_grouped[brooklyn_grouped['Neighborhood'] == Nghhood].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_RestHtl))
    print('\n')
```

```
----Brooklyn Heights----
                  venue  freq
0        Italian Restaurant  0.14
1       American Restaurant  0.09
2         Indian Restaurant  0.09
3           Thai Restaurant  0.09
4          Asian Restaurant  0.09
5          Sushi Restaurant  0.05
```

e) Sorting the Venues in the Descending Order

```
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = brooklyn_grouped['Neighborhood']

for ind in np.arange(brooklyn_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(brooklyn_grouped.iloc[ind, :], num_top_RestHtl)

neighborhoods_venues_sorted
```

Out[81]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Brooklyn Heights	Italian Restaurant	American Restaurant	Thai Restaurant	Asian Restaurant	Indian Restaurant
1	Carroll Gardens	Italian Restaurant	Thai Restaurant	Cuban Restaurant	Restaurant	French Restaurant
2	Cobble Hill	Italian Restaurant	Japanese Restaurant	Thai Restaurant	French Restaurant	Mediterranean Restaurant
3	Downtown	French Restaurant	Thai Restaurant	Asian Restaurant	Chinese Restaurant	Shanghai Restaurant
4	Greenpoint	French Restaurant	Mexican Restaurant	New American Restaurant	Sushi Restaurant	Italian Restaurant
5	North Side	American Restaurant	Vegetarian / Vegan Restaurant	Mediterranean Restaurant	Chinese Restaurant	South American Restaurant
6	South Side	American Restaurant	Chinese Restaurant	Seafood Restaurant	Vegetarian / Vegan Restaurant	Korean Restaurant

STEP 6: Clustering the neighborhood using **K-means** & identifying the locations on the Map.

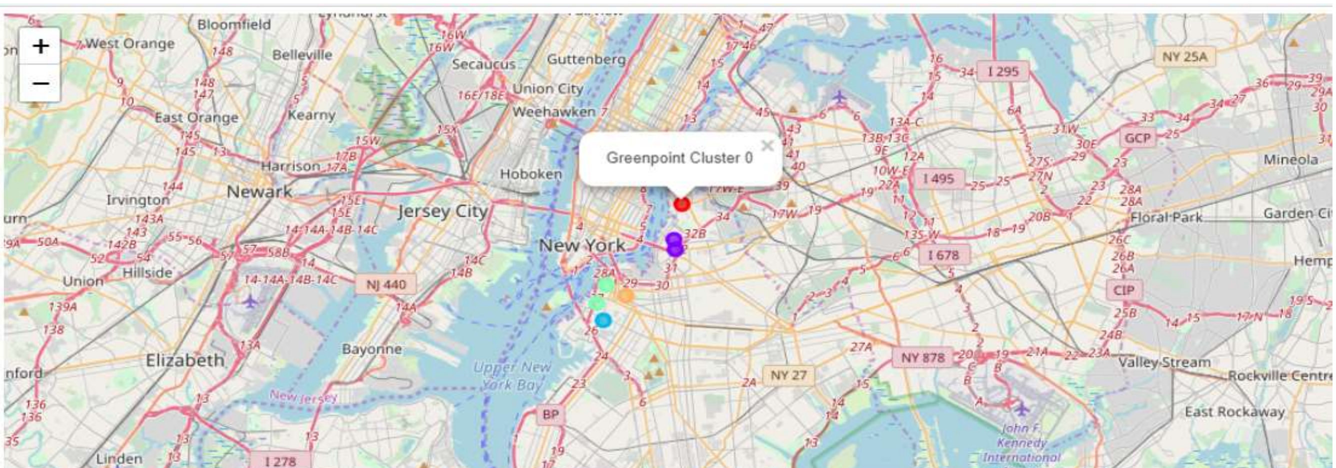
```
brooklyn_grouped_clustering = brooklyn_grouped.drop('Neighborhood', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
#brooklyn_venues_final_4Kmeans=brooklyn_venues_final.drop(["Venue Latitude","Venue Longitude","count"],axis=1)
#brooklyn_venues_final_4Kmeans
#neighborhoods_venues_sorted
neighborhoods_venues_sorted=neighborhoods_venues_sorted.drop(["Cluster Labels"],axis=1)
neighborhoods_venues_sorted
```

Out[119]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Brooklyn Heights	Italian Restaurant	American Restaurant	Thai Restaurant	Asian Restaurant	Indian Restaurant
1	Carroll Gardens	Italian Restaurant	Thai Restaurant	Cuban Restaurant	Restaurant	French Restaurant
2	Cobble Hill	Italian Restaurant	Japanese Restaurant	Thai Restaurant	French Restaurant	Mediterranean Restaurant
3	Downtown	French Restaurant	Thai Restaurant	Asian Restaurant	Chinese Restaurant	Shanghai Restaurant
4	Greenpoint	French Restaurant	Mexican Restaurant	New American Restaurant	Sushi Restaurant	Italian Restaurant

CLUSTER MAP

Out[47]:



STEP 7: Concluding the Choices of Restaurants & Locations basis of the data analysis in Step

a) **Examining the Cluster -**0 - Green Point

In [122]:
```
# Examining the Clusters
# Cluster =
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 0, brooklyn_merged.columns[[1] + list(range(5, brooklyn_merged.shape[1]))]]
```

Out[122]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
3	Greenpoint	French Restaurant	Mexican Restaurant	New American Restaurant	Sushi Restaurant	Italian Restaurant

b) **Examining the Cluster -2** – Carrol Gardens

In [50]:
```
# Examining the Clusters
# Cluster = 2
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 2, brooklyn_merged.columns[[1] + list(range(5, brooklyn_merged.shape[1]))]]
```

Out[50]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
20	Carroll Gardens	Italian Restaurant	Thai Restaurant	Cuban Restaurant	Restaurant	French Restaurant

c) **Examining the Cluster -3** – Brooklyn Heights

In [51]:
```
# Examining the Clusters
# Cluster = 3
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 3, brooklyn_merged.columns[[1] + list(range(5, brooklyn_merged.shape[1]))]]
```

Out[51]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
18	Brooklyn Heights	Italian Restaurant	American Restaurant	Thai Restaurant	Asian Restaurant	Indian Restaurant

RESULTS

Out of those shortlisted three Neighborhoods, Asian & Indian Restaurants are not that common in Cluster 0 or in Cluster 2, whereas it's quite common in Brooklyn Heights. So Indian Restaurant would be preferred in Carrol Gardens or GreenPoint. If It's Italian Restaurant, best bet would be @ GreenPoint.

DISCUSSION

- When combining data from multiple sources, inconsistent can happen. And lots of efforts are required to check, research and change the data before merge.
- For data obtained through API calls, different results are returned with different set of parameters and different point of time. Multiple trial and error runs are required to get the optimal result.

- Even after the dataset has been constructed, lots of research and analysis are required to decide if the data should be kept as is or be transform by normalization or standardization.

It can be considered the most important process in the whole data science pipeline. Which can affect the most on the result.

On the other hand, choosing the suitable technique to construct the model is also a worthwhile process. As this report shows that, by applying a different method, the result can be improved.

CONCLUSION

It's an attempt to explore the different possible analysis we could do in the available data and rationalize the decision. Although all of the goals of this project were met there is definitely room for further improvement by analyzing few more supplementary data points like demographic information, Average Spent of the population, Proximity of other crowd pulling venues like Malls, shopping complex, Cinema halls etc. However, this project could definitely be handy to narrow down a Neighborhood and a type of Restaurant as a first step.