Capstone Project:
The Battle of Neighborhoods
Opening a Japanese restaurant in New York

Author: Elizaveta Kuzevanova

March 15, 2021

1. Introduction

- New York City is perhaps the most difficult market in which to open a restaurant. Businesses must distinguish themselves from countless others — both new and old — diners' tastes are constantly evolving and the rent is, well, too damn high. One trend that has picked up steam in the past year or two in the Big Apple is the opening of Japanese chain (or mini-chain) restaurants. These restaurants have all seized on growth opportunities in New York and diners have been quick to gravitate towards them (Embricos, 2018).
- This Capstone project explores different neighborhoods in New York, and attempts to answer the following business problem: "*If investor is looking to open a new Japanese restaurant, where would you recommend that they open it?*".
- This project might be of interest to potential *business investors* specializing in Japanese cuisine/ restaurant chains, as well as to aspiring *Data Scientists*, who want to learn how to implement some of the most common Exploratory Data Analysis techniques to obtain necessary data, analyze it and, finally to be able to explain your insights in a compelling story.

2. Data

- For this project we need the following data:
- New York City data that contains all Boroughs and Neighborhoods along with their latitudes and longitudes.
  Data Source: https://cocl.us/new_york_dataset
- Median price per square foot for each Neighborhood in NYC.
  Data Source: https://www.zumper.com/blog/nyc-by-square-foot-see-which-neighborhood-gets-you-the-most-space-for-your-money/
- DatData related to locations and ratings of Japanese restaurants in NYC. a
  Source: **Foursquare API**

3. Methodology

- import all required libraries
- obtain information about NYC boroughs/neighborhoods along with their *coordinates* (using **requests** library) from https://cocl.us/new_york_dataset and load it into a data frame
- obtain information about *medium rental Price per Sq Foot* for each NYC neighborhood from https://www.zumper.com/blog/nyc-by-square-foot-see-which-neighborhood-gets-you-the-most-space-for-your-money/ and load it into a data frame (using **BeautifulSoup** package)
- merge the above data frames on their Neighborhood value (note that not all neighborhoods had info on the rental price in the area, so there will be some data cleansing steps required along the way)
- next, we are going to start utilizing the **Foursquare API** to explore the neighborhoods and segment them:
  - define Foursquare credentials and version
  - define a function **get_venues** that returns top 100 venues for a Neighborhood within a radius of 500 meters (using url to fetch data from Foursquare API)
  - analyze how many Japanese restaurants are there in each Neighborhood and borough
    - prepare a list that contains Japanese restaurants using **get_venues** function
    - calculate how many Japanese restaurants are there in each Neighborhood
    - merge the new data with the data frame from the previous step on Neighborhoods to link the number of Japanese restaurants in the Neighborhood to its coordinates and median rental price per sq foot
    - data cleansing and pre-processing to prepare data frame for clustering
- run clustering on the final data frame containing Neighborhoods, the medium rental price per sq foot and total number of Japanese restaurants in the area using **K-means algorithm** (set number of clusters = 5)
- add clustering labels to the final data frame
- visualize the results
  - use **geopy** library to get the latitude and longitude values of New York City
  - create map to visualize clusters
  - examine the clusters

Now let's visualize the data:

As not all of NYC neighbourhoods had data on Medium Price per Sq Foot in our data source, there are some missing values in the initial **ny_data** data set. Let's proceed to final data cleansing before Cluster Visualization.

In [47]:
```
# simply drop whole row with NaN in "Cluster" column
ny_merged.dropna(subset=["Cluster Labels"], axis=0, inplace=True)

# reset index, because we droped two rows
ny_merged.reset_index(drop=True, inplace=True)

ny_merged
```

Out[47]:

	Borough	Neighbourhood	Latitude	Longitude	Cluster Labels	Median Price Per Sq Foot	# of Japanese restaurants
0	Bronx	Wakefield	40.894705	-73.847201	1.0	2.08	0.0
1	Bronx	Eastchester	40.887556	-73.827806	1.0	2.07	0.0
2	Bronx	Riverdale	40.890834	-73.912585	1.0	2.52	0.0
3	Bronx	Kingsbridge	40.881687	-73.902818	1.0	2.51	0.0
4	Bronx	Norwood	40.877224	-73.879391	1.0	2.13	0.0
5	Bronx	Williamsbridge	40.881039	-73.857446	2.0	1.80	0.0
6	Bronx	Pelham Parkway	40.857413	-73.854756	1.0	2.17	0.0
7	Bronx	City Island	40.847247	-73.786488	2.0	1.88	0.0
8	Bronx	Bedford Park	40.870185	-73.885512	1.0	2.07	0.0
9	Bronx	University Heights	40.855727	-73.910416	1.0	2.07	0.0
10	Bronx	Morris Heights	40.847898	-73.919672	2.0	1.12	0.0

Final data frame (after data cleansing and pre-processing):

In [50]:
```
ny_merged['Cluster Labels'] = ny_merged['Cluster Labels'].astype('int64')
ny_merged['# of Japanese restaurants'] = ny_merged['# of Japanese restaurants'].astype('int64')
ny_merged.head()
```

Out[50]:

	Borough	Neighbourhood	Latitude	Longitude	Cluster Labels	Median Price Per Sq Foot	# of Japanese restaurants
0	Bronx	Wakefield	40.894705	-73.847201	1	2.08	0
1	Bronx	Eastchester	40.887556	-73.827806	1	2.07	0
2	Bronx	Riverdale	40.890834	-73.912585	1	2.52	0
3	Bronx	Kingsbridge	40.881687	-73.902818	1	2.51	0
4	Bronx	Norwood	40.877224	-73.879391	1	2.13	0

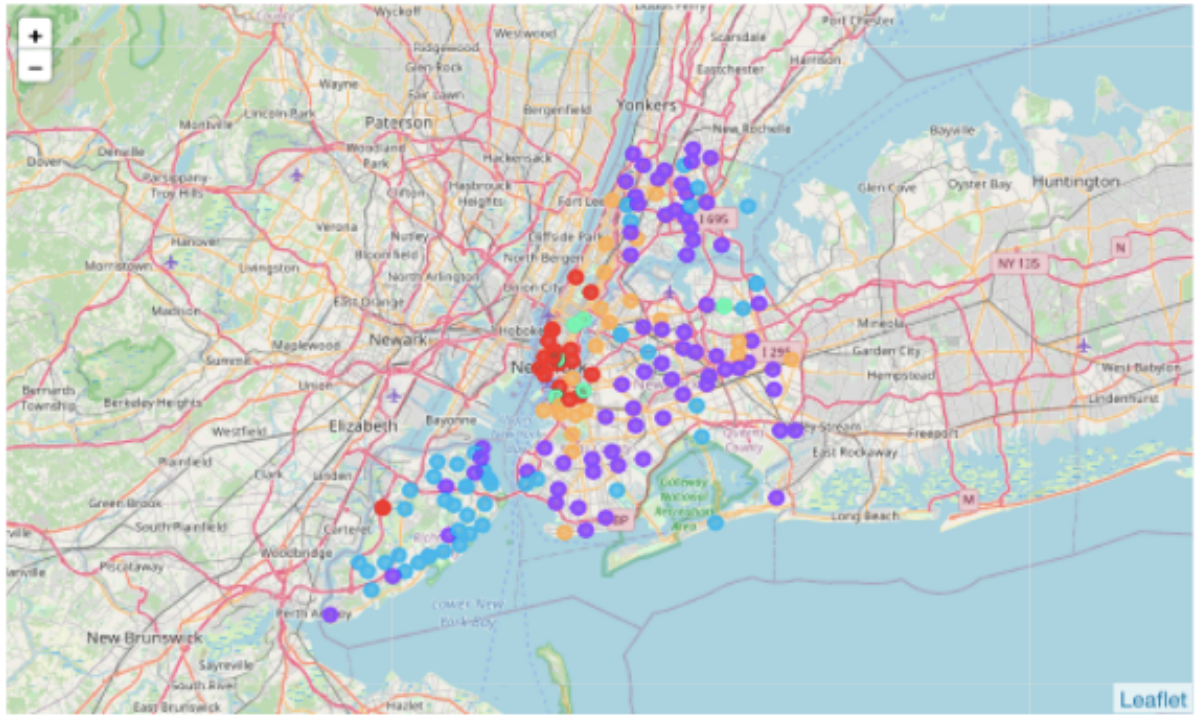
In [51]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# 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 markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(ny_merged['Latitude'], ny_merged['Longitude'], ny_merged['Neighbourhood'], ny_merged['Cluster 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)

map_clusters
```

Out[51]:



4. Results

Now, we can examine each cluster and determine the discriminating features that distinguish each cluster.

> We can see that Neighbourhoods in **Cluster 1** have *high* Median Price per Sq Foot and *medium* number of Japanese restaurants in the area.

> We can see that Neighbourhoods in **Cluster 2** have *low* Median Price per Sq Foot and almost *no Japanese restaurants in the area*.

> We can see that Neighbourhoods in **Cluster 3** have *low* Median Price per Sq Foot and *low* number of Japanese restaurants in the area.

> We can see that Neighbourhoods in **Cluster 4** have *high* Median Price per Sq Foot and *high* number of Japanese restaurants in the area.

> We can see that Neighbourhoods in **Cluster 5** have *medium* Median Price per Sq Foot and *medium* number of Japanese restaurants in the area.

5. Discussion

- There are several options for potential improvement of the analysis results:
  - to add an extra step of finding an **optimal value of K** for clustering purposes (e.g. using the Elbow method).
  - to add other features for analysis (e.g. **average rating** of the restaurants in the neighborhood etc.)

6. Conclusion

- **Cluster 1:** high rental price & medium competition/demand
- **Cluster 2:** low rental price & no competition/demand
- **Cluster 3:** low rental price & low competition/demand
- **Cluster 4:** high rental price & high competition/demand
- **Cluster 5:** medium rental price & medium competition/demand

Out of the Cluster descriptions outlined above, **any neighborhood in Cluster 5** that doesn't have a Japanese restaurant yet seems to be the most attractive option for investment, although this choice highly depends on the investor's budget.