Capstone Project - Battle of the Neighbors

Clustering and Segmentation of New York City based on Cuisines

by Antony Sundar Singh

February 22, 2020

Table of Contents

Problem Background

New York City is the most populous city in the United States and it just might be the most diverse city on the planet, as it is home to over 8.6 million people and over 800 languages.

The idea of this project is to categorically segment the neighborhoods of New York City into major clusters and examine their cuisines. A desirable intention is to examine the neighborhood cluster's food habits and taste. Further examination might reveal if food has any relationship with the diversity of a neighborhood.

This project will help to understand the diversity of a neighborhood by leveraging venue data from Foursquare's 'Places API' and 'k-means clustering' machine learning algorithm. Exploratory Data Analysis (EDA) will help to discover further about the culture and diversity of the neighborhood.

Targeted Audience

Stakeholders would be the one who are interested to use this quantifiable analysis to understand the distribution of different cultures and cuisines over "the most diverse city on the planet - NYC". Also, this project can be utilized by a new food vendor who is willing to open his or her restaurant. Or by a government authority to examine and study their city's culture diversity better.

Data

To examine the above said, following data sources will be used:

New York City Dataset

Link: https://geo.nyu.edu/catalog/nyu_2451_34572

Description: This New York City Neighborhood Names point file was created as a guide to New York City's neighborhoods that appear on the web resource, "New York: A City of Neighborhoods." Best estimates of label centroids were established at a 1:1,000 scale, but are ideally viewed at a 1:50,000 scale. This dataset will provide the addresses of neighborhood of NYC in json format. An extract of the json is as follows:

{'type': 'Feature',
'id': 'nyu_2451_34572.306',
'geometry': {'type': 'Point',
'coordinates': [-74.08173992211962, 40.61731079252983]},
'geometry_name': 'geom',
'properties': {'name': 'Fox Hills',
'stacked': 2,

'annoline1': 'Fox',
'annoline2': 'Hills',
'annoline3': None,
'annoangle': 0.0,
'borough': 'Staten Island',
'bbox': [-74.08173992211962,
40.61731079252983,
-74.08173992211962,

- 40.61731079252983]}}

Foursquare API

Link: https://developer.foursquare.com/docs

Description: Foursquare API, a location data provider, will be used to make RESTful API calls to retrieve data about venues in different neighborhoods. This is the link to Foursquare Venue Category Hierarchy. Venues retrieved from all the neighborhoods are categorized broadly into "Arts & Entertainment", "College & University", "Event", "Food", "Nightlife Spot", "Outdoors & Recreation", etc. An extract of an API call is as follows:

'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}],
'verified': False,
'stats': {'tipCount': 17},
'url': 'http://eccorestaurantny.com',

- 'price': {'tier': 4, 'message': 'Very Expensive', 'currency'

Methodology

Download and Explore New York City Dataset

In order to segment the neighborhoods of New York City, a dataset is required that contains the 5 boroughs and the neighborhoods, that exist in each borough, with respective latitude and longitude coordinates. This dataset is downloaded using the mentioned URL.

Once the .json file is downloaded, it is analyzed to understand the structure of the file. A python dictionary is returned by the URL and all the relevant data is found to be in the features key, which is basically a list of the

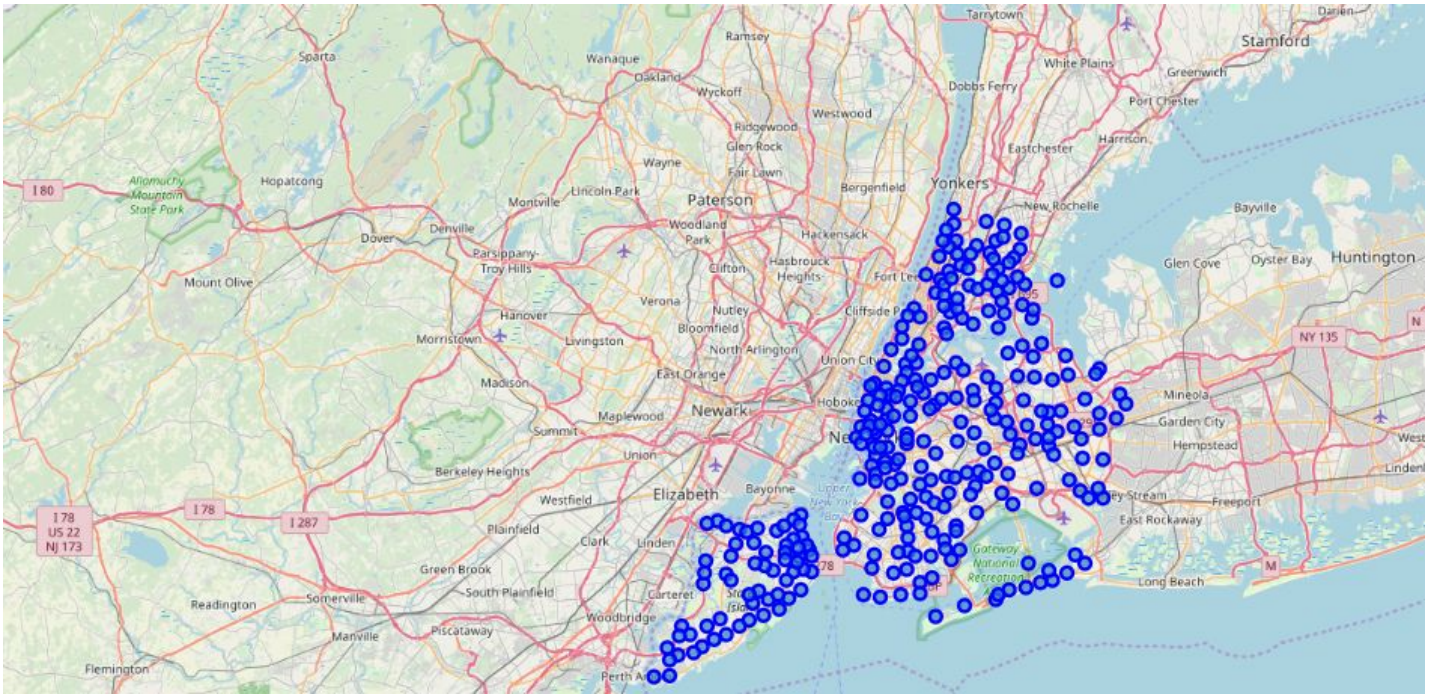
neighborhoods. The dictionary is transformed, into a pandas dataframe, by looping through the data and filling the data frame rows one at a time.

As a result, a data frame is created with Borough, Neighborhood, Latitude and Longitude details of the New York City's neighborhood.

	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

Upon analysis, it is found that the dataframe consists of 5 boroughs and 306 neighborhoods.

Further, 'geopy' library is used to get the latitude and longitude values of New York City, which was returned to be Latitude: 40.71, Longitude: -74.01. The curated dataframe is then used to visualize by creating a map of New York City with neighborhoods superimposed on top. The following depiction is a map generated using python 'folium' library.



RESTful API Calls to Foursquare

The Foursquare API is used to explore the neighborhoods and segment them. To access the API, 'CLIENT_ID', 'CLIENT_SECRET' and 'VERSION' is defined. There are many endpoints available on

Foursquare for various GET requests. But, to explore the cuisines, it is required that all the venues extracted are from ‘Food’ category. Foursquare Venue Category Hierarchy is retrieved using the following code block:

```
url = 'https://api.foursquare.com/v2/venues/categories?&client_id={}&client_secret={}&v={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION)
category_results = requests.get(url).json()
```

Upon analysis, it is found that there are 10 major or parent categories of venues, under which all the other sub-categories are included. Following depiction shows the ‘Category ID’ and ‘Category Name’ retrieved from API:

```
for data in category_list:
    print(data['id'], data['name'])
```

```
4d4b7104d754a06370d81259 Arts & Entertainment
4d4b7105d754a06372d81259 College & University
4d4b7105d754a06373d81259 Event
4d4b7105d754a06374d81259 Food
4d4b7105d754a06376d81259 Nightlife Spot
4d4b7105d754a06377d81259 Outdoors & Recreation
4d4b7105d754a06375d81259 Professional & Other Places
4e67e38e036454776db1fb3a Residence
4d4b7105d754a06378d81259 Shop & Service
4d4b7105d754a06379d81259 Travel & Transport
```

As said earlier, the ‘FOOD’ category in the above depiction is the matter of interest. A function is created to return a dictionary with ‘Category ID’ & ‘Category Name’ of ‘Food’ & it's sub-categories. To further understand the results of GET Request, the first neighborhood of the ‘New York City’ dataset is explored. The first neighborhood returned is ‘Wakefield’ with Latitude 40.89 and Longitude -73.85. Then, a GET request URL is created to search for Venue with ‘Category ID’ = '4d4b7105d754a06374d81259', which is the ‘Category ID’ for ‘Food’, and radius = 500 meters.

To overcome the redundancy of the process followed above, a function ‘getNearbyFood’ is created. This functions loop through all the neighborhoods of New York City and creates an API request URL with radius = 500, LIMIT = 100. By limit, it is defined that maximum 100 nearby venues should be returned.

Further, the GET request is made to Foursquare API and only relevant information for each nearby venue is extracted from it. The data is then appended to a python ‘list’. Lastly the python ‘list’ is unfolded or flattened to append it to dataframe being returned by the function.

It is inquisitive to know that Foursquare API returns all the sub-categories, if a toplevel category is specified in the GET Request.

Pickle

Pickle is a very important and easy-to-use library. It is used to serialize the information retrieved from GET requests, to make a persistent ‘.pkl’ file. This file can later be deserialized to retrieve an exact python object structure. This is a crucial step as it will counter any redundant requests to the Foursquare API, which is chargeable over the threshold limits. The returned ‘dataframe’ is as follows.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Wakefield	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop
1	Wakefield	40.894705	-73.847201	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop
2	Wakefield	40.894705	-73.847201	Cooler Runnings Jamaican Restaurant Inc	40.898083	-73.850259	Caribbean Restaurant
3	Wakefield	40.894705	-73.847201	E&L Bakery	40.893564	-73.856997	Bakery
4	Wakefield	40.894705	-73.847201	SUBWAY	40.890468	-73.849152	Sandwich Place

As of now, two python ‘dataframe’ are created:

1. ‘neighborhoods’ which contains the Borough, Neighborhood, Latitude and Longitude details of the New York City’s neighborhood, and

2. ‘nyc_venues’ which is a merger between ‘neighborhoods’ dataframe and its ‘Food’ category venues searched with ‘Radius’ = 500 meters and ‘Limit’ = 100. Also, each venue has its own Latitude, Longitude and Category.

Exploratory Data Analysis

The merged dataframe ‘nyc_venues’ has all the required information. The size of this dataframe is determined, and it is found that there are total 14,047 venues.

```
print(nyc_venues.shape)
nyc_venues.head()
```

```
(13908, 7)
```

Now, it is important to find out that how many unique categories can be curated from all the returned venues. There are 194 such categories, with most occurring venues as follows:

```
print('There are {} uniques categories.'.format(len(nyc_venues['Venue Category'].unique())))
nyc_venues.groupby('Venue Category')['Venue Category'].count().sort_values(ascending=False)
```

```
There are 202 uniques categories.

Venue Category
Deli / Bodega                1046
Pizza Place                  1036
Coffee Shop                   905
Donut Shop                    638
Fast Food Restaurant          620
Chinese Restaurant            607
Bakery                        548
Italian Restaurant            544
American Restaurant           443
Café                          426
Caribbean Restaurant          361
Mexican Restaurant            337
Bagel Shop                    335
Sandwich Place                325
Fried Chicken Joint           324
Diner                         289
```

Data Cleaning

It is crucial to understand that the point of interest in the project is to understand the cultural diversity of a neighborhood by clustering it categorically, using the venues' categories. Thus, it is important to remove all the venues from the 'dataframe' which have generalized categories. Here, by generalized, it means that these categorized venues are common across different cultures and food habits. Example of categories of this type of venues are Coffee Shop, Cafe, etc. So, firstly all the unique categories are fed into a python 'list'.

Then, manually the categories are determined to be 'general' (as explained above). This data pre-preparation totally depends upon the 'Data Analyst' discretion and can be modified as required. Following are the categories listed as 'general':

```
# manually create a list of generalized categories
general_categories = ['Dessert Shop','Food','Ice Cream Shop','Donut Shop','Bakery','Sandwich Place','Comfort Food Restaurant',
                      'Deli / Bodega','Food Truck','Bagel Shop','Burger Joint','Restaurant','Frozen Yogurt Shop','Coffee Shop',
                      'Diner','Wings Joint','Café','Juice Bar','Breakfast Spot','Grocery Store','Bar','Cupcake Shop',
                      'Pub','Fish & Chips Shop','Cafeteria','Other Nightlife','Arcade','Hot Dog Joint','Food Court',
                      'Health Food Store','Convenience Store','Food & Drink Shop','Cocktail Bar','Cheese Shop',
                      'Snack Place','Sports Bar','Lounge','Theme Restaurant','Buffet','Bubble Tea Shop','Building',
                      'Irish Pub','College Cafeteria','Tea Room','Supermarket','Hotpot Restaurant','Gastropub','Beer Garden',
                      'Fish Market','Beer Bar','Clothing Store','Music Venue','Bistro','Salad Place','Wine Bar','Gourmet Shop',
                      'Indie Movie Theater','Art Gallery','Gift Shop','Pie Shop','Fruit & Vegetable Store',
                      'Street Food Gathering','Dive Bar','Factory','Farmers Market','Mac & Cheese Joint','Creperie',
                      'Candy Store','Event Space','Skating Rink','Miscellaneous Shop','Gas Station','Organic Grocery',
                      'Pastry Shop','Club House','Flea Market','Hotel','Furniture / Home Store','Bookstore','Pet Café',
                      'Gym / Fitness Center','Flower Shop','Financial or Legal Service','Hotel Bar','Hookah Bar','Poke Place',
                      'Market','Gluten-free Restaurant','Smoothie Shop','Butcher','Food Stand','Beach Bar','Beach',
                      'Soup Place','Rock Club','Residential Building (Apartment / Condo)','Laundry Service',
                      'Government Building','Bowling Alley','Nightclub','Park','Moving Target']
```

The python 'list' curated above, is used to remove all the venues with categories not in 'food_categories', and the following dataframe is retrieved:

```
nyc_venues = nyc_venues[nyc_venues['Venue Category'].isin(food_categories)].reset_index()
nyc_venues.head(5)
```

:

	index	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	2	Wakefield	40.894705	-73.847201	Cooler Runnings Jamaican Restaurant Inc	40.898083	-73.850259	Caribbean Restaurant
1	5	Wakefield	40.894705	-73.847201	Burger King	40.895540	-73.856460	Fast Food Restaurant
2	7	Wakefield	40.894705	-73.847201	Golden Krust Caribbean Restaurant	40.903773	-73.850051	Caribbean Restaurant
3	9	Wakefield	40.894705	-73.847201	McDonald's	40.892779	-73.857473	Fast Food Restaurant
4	12	Wakefield	40.894705	-73.847201	McDonald's	40.902645	-73.849485	Fast Food Restaurant

Feature Engineering

Now, each neighborhood is analyzed individually to understand the most common cuisine being served within its 500 meters of vicinity. The above process is taken forth by using 'one hot encoding' function of python 'pandas' library. One hot encoding converts the categorical variables (which are 'Venue Category') into a form that could be provided to ML algorithms to do a better job in prediction. Number of venues of each category in each neighborhood are counted.

```
venue_counts = nyc_onehot.groupby('Neighborhood').sum()
venue_counts.head(5)
```

Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Australian Restaurant	Austrian Restaurant	BBQ Joint	Bath House	Brazilian Restaurant	Burrito Place	Cajun / Creole Restaurant	Cambodian Restaurant	Cantonese Restaurant	Caribbean Restaurant	Caucasian Restaurant	Chinese Restaurant	Chocolate Shop
Allerton	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	0
Annadale	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Arden Heights	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0
Arlington	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0
Arrochar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Next, the rows of the neighborhood are grouped together and the frequency of occurrence of each category is calculated by taking the mean.

```
nyc_grouped = nyc_onehot.groupby('Neighborhood').mean().reset_index()
nyc_grouped.head()
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Australian Restaurant	Austrian Restaurant	BBQ Joint	Bath House	Brazilian Restaurant	Burrito Place	Cajun / Creole Restaurant	Cambodian Restaurant	Cantonese Restaurant	Caribbean Restaurant	Caucasian Restaurant	Chinese Restaurant
0	Allerton	0.0	0.0	0.041667	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.041667	0.0	0.166667
1	Annadale	0.0	0.0	0.166667	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.055556
2	Arden Heights	0.0	0.0	0.157895	0.0	0.0	0.000000	0.0	0.0	0.052632	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.105263
3	Arlington	0.0	0.0	0.111111	0.0	0.0	0.055556	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.055556	0.0	0.055556
4	Arrochar	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.062500

As the limit is set to be 100, there will be many venues being returned by the Foursquare API. But a neighborhood food habit can be defined by the top 5 venues in its vicinity. The above created dataframe is fed with the top 5 most common venues categories in the respective neighborhood.

```
for ind in np.arange(nyc_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(nyc_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Allerton	Pizza Place	Chinese Restaurant	Mexican Restaurant	Fried Chicken Joint	Fast Food Restaurant
1	Annadale	Pizza Place	Italian Restaurant	American Restaurant	Sushi Restaurant	Japanese Restaurant
2	Arden Heights	Pizza Place	Italian Restaurant	American Restaurant	Sushi Restaurant	Mexican Restaurant
3	Arlington	Pizza Place	American Restaurant	Peruvian Restaurant	Fast Food Restaurant	Spanish Restaurant
4	Arrochar	Italian Restaurant	Pizza Place	Middle Eastern Restaurant	Mediterranean Restaurant	Polish Restaurant

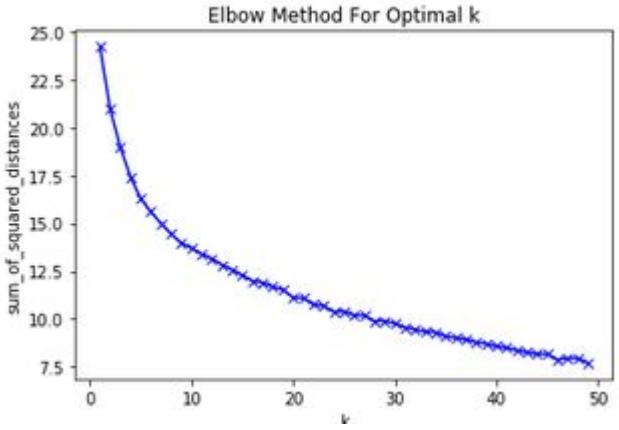
Machine Learning

‘k-means’ is an unsupervised machine learning algorithm which creates clusters of data points aggregated together because of certain similarities. This algorithm will be used to count neighborhoods for each cluster label for variable cluster size. To implement this algorithm, it is very important to determine the optimal number

of clusters (i.e. k). There are 2 most popular methods for the same, namely ‘The Elbow Method’ and ‘The Silhouette Method’.

The Elbow Method

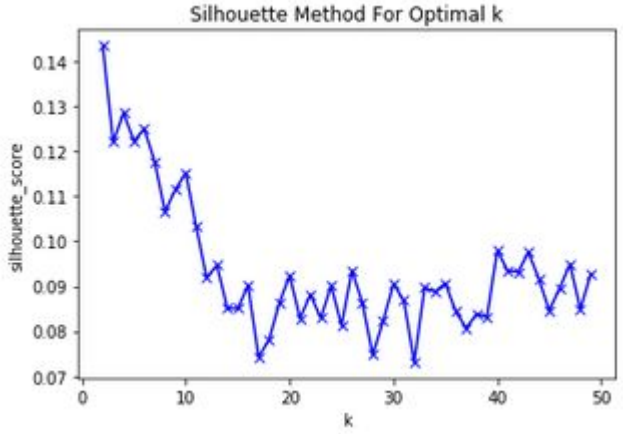
The Elbow Method calculates the sum of squared distances of samples to their closest cluster center for different values of ‘k’. The optimal number of clusters is the value after which there is no significant decrease in the sum of squared distances.



Sometimes, Elbow method does not give the required result, which happened in this case. As, there is a gradual decrease in the sum of squared distances, optimal number of clusters can not be determined. To counter this, another method can be implemented, as discussed below.

The Silhouette Method

As quoted in Wikipedia – “The Silhouette Method measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation).” It requires minimum 2 clusters to define dissimilarity number of clusters (i.e. ‘k’) will vary from 2 to 49.



k-Means

Following code block runs the k-Means algorithm with number of clusters = 8 and prints the counts of neighborhoods assigned to different clusters:

```
# set number of clusters
kclusters = 8

# run k-means clustering
kmeans = KMeans(init="k-means++", n_clusters=kclusters, n_init=50).fit(nyc_grouped_clustering)

print(Counter(kmeans.labels_))
```

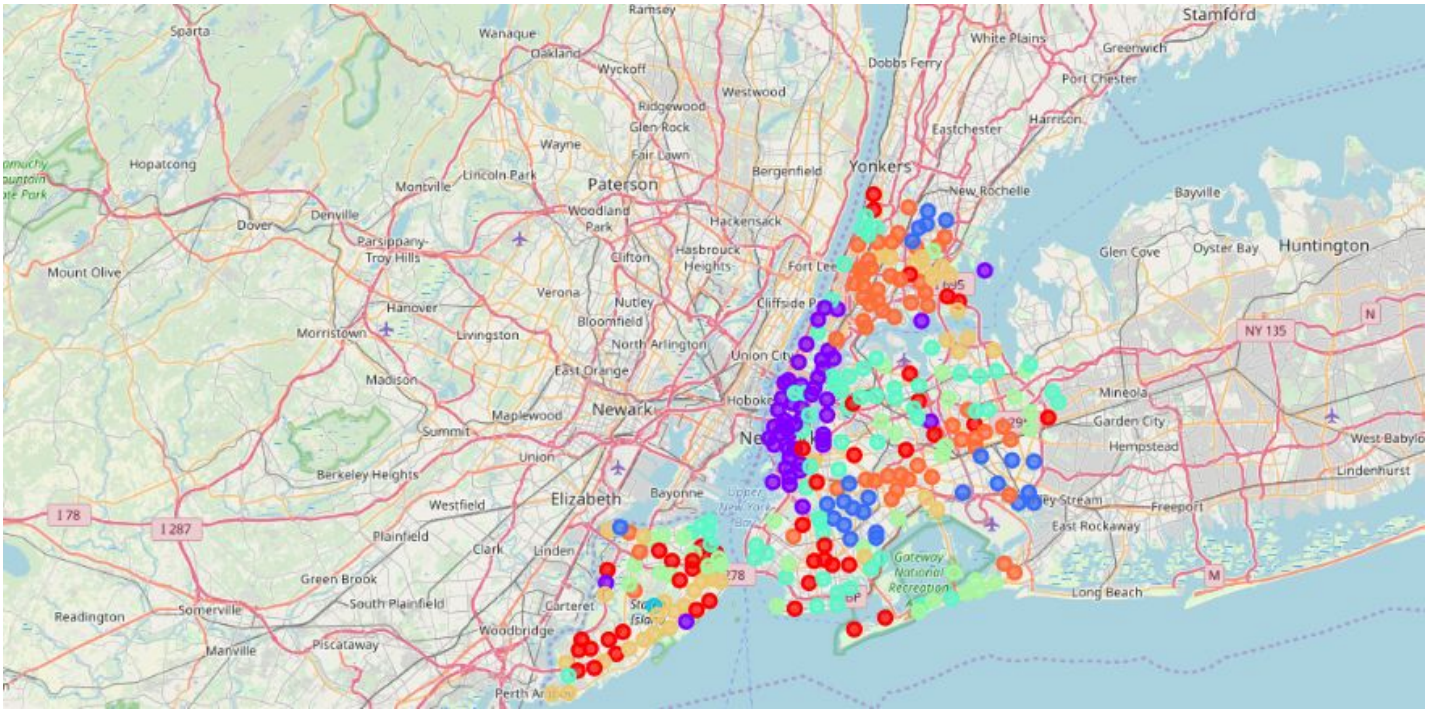
Further the cluster labels curated are added to the dataframe to get the desired results of segmenting the neighborhood based upon the most common venues in its vicinity:

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	7	Allerton	Pizza Place	Chinese Restaurant	Mexican Restaurant	Fried Chicken Joint	Fast Food Restaurant
1	0	Annadale	Pizza Place	Italian Restaurant	American Restaurant	Sushi Restaurant	Japanese Restaurant
2	0	Arden Heights	Pizza Place	Italian Restaurant	American Restaurant	Sushi Restaurant	Mexican Restaurant
3	7	Arlington	Pizza Place	American Restaurant	Peruvian Restaurant	Fast Food Restaurant	Spanish Restaurant
4	6	Arrochar	Italian Restaurant	Pizza Place	Middle Eastern Restaurant	Mediterranean Restaurant	Polish Restaurant

Now, ‘neighborhoods_venues_sorted’ is merged with ‘nyc_data’ to add the Borough, Latitude and Longitude for each neighborhood.

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
0	7	Allerton	Pizza Place	Chinese Restaurant	Mexican Restaurant	Fried Chicken Joint	Fast Food Restaurant	Bronx	40.865788	-73.859319
1	0	Annadale	Pizza Place	Italian Restaurant	American Restaurant	Sushi Restaurant	Japanese Restaurant	Staten Island	40.538114	-74.178549
2	0	Arden Heights	Pizza Place	Italian Restaurant	American Restaurant	Sushi Restaurant	Mexican Restaurant	Staten Island	40.549286	-74.185887
3	7	Arlington	Pizza Place	American Restaurant	Peruvian Restaurant	Fast Food Restaurant	Spanish Restaurant	Staten Island	40.635325	-74.165104
4	6	Arrochar	Italian Restaurant	Pizza Place	Middle Eastern Restaurant	Mediterranean Restaurant	Polish Restaurant	Staten Island	40.596313	-74.067124

Again, the New York City’s neighborhoods are visualized by using the code block as shown, which utilizes the python ‘folium’ library.


Sparta
Wanaque
Ramsey
Oakland
Wyckoff
Ridgewood
Glen Rock
Wayne
Fair Lawn
Paterson
Lincoln Park
Montville
Hopatcong
Denville
Dover
Parsippany-Troy Hills
Mount Olive
Woodland Park
Clifton
Hasbrouck Heights
Hackensack
Bergenfield
Westwood
Dobbs Ferry
White Plains
Greenwich
Stamford
Port Chester
Harrison
Eastchester
Yonkers
New Rochelle
Bayville
Glen Cove
Oyster Bay
Huntington
Fort Lee
Cliffside Park
Verona
Nutley
Bloomfield
Hanover
Morristown
Livingston
North Arlington
Union City
Hoboken
Madison
East Orange
Newark
Maplewood
Summit
Union
Berkeley Heights
Elizabeth
Bayonne
Westfield
Plainfield
Clark
Linden
Green Brook
South Plainfield
Somerville
Readington
Manville
Piscataway
Carteret
Woodbridge
Perth Amboy
Staten Island
Gateway National Recreation
Long Beach
East Rockaway
Freeport
Hempstead
Garden City
Mineola
Lindenhurst
West Babylon
NY 135
I 78
I 287
M
N

Results

We have created 8 clusters using k-Mean algorithm..

Cluster 0 :

```
for col in required_column:
    print(cluster_0[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Pizza Place            42
Taco Place              1
American Restaurant     1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Italian Restaurant      11
American Restaurant      8
Fast Food Restaurant     6
Mexican Restaurant       4
Sushi Restaurant         3
Japanese Restaurant      2
Pizza Place              2
BBQ Joint                2
Chinese Restaurant       2
Taco Place               1
Asian Restaurant         1
Thai Restaurant          1
Spanish Restaurant       1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
Staten Island    19
Brooklyn         11
Queens            9
Bronx             5
Name: Borough, dtype: int64
---------------------------------------------
```

Cluster 1 :

```
for col in required_column:
    print(cluster_1[col].value_counts(ascending = False))
    print("----------------------------------------------")
```

```
American Restaurant                25
Pizza Place                         6
Italian Restaurant                  6
French Restaurant                   3
New American Restaurant             3
Dim Sum Restaurant                  2
Korean Restaurant                   1
Southern / Soul Food Restaurant     1
Seafood Restaurant                  1
Fast Food Restaurant                1
Name: 1st Most Common Venue, dtype: int64
----------------------------------------------
American Restaurant         14
Italian Restaurant           9
Pizza Place                  8
Japanese Restaurant          2
Seafood Restaurant           2
Ramen Restaurant             2
Thai Restaurant              2
Mexican Restaurant           2
Chinese Restaurant           2
German Restaurant            1
Korean Restaurant            1
Mediterranean Restaurant     1
Dim Sum Restaurant           1
French Restaurant            1
BBQ Joint                    1
Name: 2nd Most Common Venue, dtype: int64
----------------------------------------------
Manhattan        29
Brooklyn         14
Queens            2
Staten Island     2
Bronx             2
Name: Borough, dtype: int64
```

Cluster 2 :

```
for col in required_column:
    print(cluster_2[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Caribbean Restaurant    21
Chinese Restaurant       2
Fried Chicken Joint      1
American Restaurant      1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
Fast Food Restaurant    7
Fried Chicken Joint     5
Pizza Place             5
Chinese Restaurant      4
Caribbean Restaurant    3
Seafood Restaurant      1
Name: 2nd Most Common Venue, dtype: int64
-----------------------------------------------
Brooklyn         11
Queens            8
Bronx             5
Staten Island     1
Name: Borough, dtype: int64
-----------------------------------------------
```

Cluster 3 :

```
for col in required_column:
    print(cluster_3[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Italian Restaurant    1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
Yemeni Restaurant    1
Name: 2nd Most Common Venue, dtype: int64
-----------------------------------------------
Staten Island    1
Name: Borough, dtype: int64
-----------------------------------------------
```

Cluster 4 :

```
for col in required_column:
    print(cluster_4[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Pizza Place                    16
Italian Restaurant              9
Fast Food Restaurant            8
Mexican Restaurant              5
Korean Restaurant               5
Sushi Restaurant                3
Indian Restaurant               2
Thai Restaurant                 2
Caribbean Restaurant            2
Ramen Restaurant                1
Greek Restaurant                1
Seafood Restaurant              1
Sri Lankan Restaurant           1
Filipino Restaurant             1
American Restaurant             1
Russian Restaurant              1
Eastern European Restaurant     1
Chinese Restaurant              1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
Pizza Place                   10
Italian Restaurant            10
Fast Food Restaurant           7
Chinese Restaurant             6
Mexican Restaurant             6
Latin American Restaurant      2
Caribbean Restaurant           2
Thai Restaurant                2
Vietnamese Restaurant          2
Greek Restaurant               2
American Restaurant            2
Russian Restaurant             2
Sushi Restaurant               1
Mediterranean Restaurant       1
Spanish Restaurant             1
```

Cluster 5 :

```
for col in required_column:
    print(cluster_5[col].value_counts(ascending = False))
    print("-------------------------------------------")
```

```
Chinese Restaurant    18
Pizza Place           13
Italian Restaurant     2
Indian Restaurant      1
Greek Restaurant       1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------------
Chinese Restaurant      12
Pizza Place             10
Italian Restaurant       4
Mexican Restaurant       2
Fried Chicken Joint      1
Asian Restaurant         1
American Restaurant      1
Japanese Restaurant      1
Cantonese Restaurant     1
Caribbean Restaurant     1
Korean Restaurant        1
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------------
Queens           18
Staten Island    11
Brooklyn          5
Bronx             1
Name: Borough, dtype: int64
-------------------------------------------
```

Cluster 6 :

```
for col in required_column:
    print(cluster_6[col].value_counts(ascending = False))
    print("-------------------------------------------")
```

```
Italian Restaurant     26
Pizza Place             9
American Restaurant     1
Asian Restaurant        1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------------
Pizza Place                14
Italian Restaurant         10
Fast Food Restaurant        4
American Restaurant         3
Asian Restaurant            2
Mexican Restaurant          2
Japanese Restaurant         1
New American Restaurant     1
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------------
Staten Island    20
Bronx             9
Queens            6
Brooklyn          2
Name: Borough, dtype: int64
-------------------------------------------
```

Cluster 7 :

```
for col in required_column:
    print(cluster_7[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Pizza Place                  20
Fast Food Restaurant         20
Chinese Restaurant            7
Fried Chicken Joint           3
Seafood Restaurant            1
Spanish Restaurant            1
Latin American Restaurant     1
Caribbean Restaurant          1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
Pizza Place                          14
Fast Food Restaurant                 12
Chinese Restaurant                   10
Fried Chicken Joint                   6
Caribbean Restaurant                  5
Mexican Restaurant                    3
Spanish Restaurant                    2
Southern / Soul Food Restaurant       1
American Restaurant                   1
Name: 2nd Most Common Venue, dtype: int64
-----------------------------------------------
Bronx            27
Queens           13
Brooklyn          9
Staten Island     3
Manhattan         2
Name: Borough, dtype: int64
-----------------------------------------------
```

Discussions

We analysed the different clusters by using the following metrics

- Cont of Borough
- Count of 1st Most Common Venue
- Count of 2nd Most Common Venue

From the analysis, it is obvious that Pizza Place is the most common venue across all the clusters. Following could be the name of the clusters segmented by K-Means unsupervised machine learning algorithm.

- Cluster 0 : Pizza
- Cluster 1 : American
- Cluster 2 : Caribbean
- Cluster 3 : Italian
- Cluster 4 : Mix of Cuisines
- Cluster 5 : Chinese
- Cluster 6 : Italian
- Cluster 7 : Fast Food

Conclusion

The Neighborhoods of New York City were very briefly segmented into eight clusters and upon analysis it was possible to rename the clusters based on the venues in and arond the neighborhood. Along with American cuisine, Italian & Chinese are very dominant in NYC.

The results of this project can be improved by using a current dataset along with API which is more interested in Food venues. The scope of the project can be expanded further to understand the dynamics of each neighborhood and suggest a new vendor a profitable venue to start his food place.

References

Notebook created by Alex Aklson and Polong Lin for the Applied Data Science Capstone project.