IBM Coursera Capstone Project

Exploring Neighborhoods in NYC

Identifying an ideal neighborhood for Opening an Indian Restaurant

Introduction

New York is one of the famous places in the world. It is diverse in many ways. It is multicultural as well as the financial hub of the USA. Today Tourism is one of the pillars of the economy and the people most often visit NYC which is rich in heritage and developed enough from a foreign perspective, like a friendly environment. Every city is unique in it's own way and gives something new. And now the information is so common regarding the location of every place around the world on your fingertips which make it easier to explore.

This data science capstone project deals with the process of leveraging location data acquired from data providers such as Foursquare to explore the neighborhoods within a targeted city and create clustering models. Using K-means clusters, similar locations with minimum distance shall be grouped into clusters. It is the simplest form of unsupervised machine learning algorithm and it helps in grouping similar data points. In this project we will go through a step by step process to make a decision whether it is a good idea to open an Indian restaurant. We analyze the neighborhoods in NYC to identify the most profitable area since the success of the restaurant depends on the people and ambience. Since we already know that NYC shelters a greater number of Indians it is a good idea to start the restaurant here, but we just need to make sure whether it is a profitable idea or not. If so, where we can place it, so it yields more profit to the owner.

Data Exploration

The data for this project will be extracted, processed and analysed by integrating the borough information for New York City extracted from the web and venue related information acquired through Foursquare API.The data extraction from web shall be done using the web scraping libraries for python such as Beautiful Soup. After extracting the html page the information shall be converted into a data frame using the pandas python library. Using the pandas library, the data will be cleaned and processed to prepare a final data frame for analysis. In order to render my data onto a map, I will

be using the Folium library. Also, to create clusters of similar regions of interest, I will be using k-means clustering technique.

Data Sources

a) I am using the "https://cocl.us/new_york_dataset" page to get all the information about the neighborhoods present in New York City. This page has the borough & the name of all the neighborhoods, Latitude and Longitude present in New York city.

b) Then "https://en.wikipedia.org/wiki/Neighborhoods_in_New_York_City" page to get all the population details of the neighborhoods.

c) To get location and other information about various venues in New York City I'm using Foursquare's explore API. Using the Foursquare's explore API (which gives venues recommendations), I'm fetching details about the venues present in New York City and collecting their names, categories and locations (latitude and longitude). From Foursquare API (https://developer.foursquare.com/docs), I retrieved the following for each venue:

Name: The name of the venue.

Id: Venue Id

Category: The category type as defined by the API.

Latitude: The latitude value of the venue.

Longitude: The longitude value of the venue.

Data Cleaning:

a) Scraping New York City Neighborhoods:

Scraped the following Wikipedia page, "https://cocl.us/new_york_dataset" in order to obtain the data about the New York City & the Neighborhoods in it.

Dataframe will consist of four columns: Borough, Neighborhood, Latitude and Longitude.

wikipedia - package is used to scrape the data from wiki.

```
NY_df = get_new_york_data()
NY_df.head()
```

	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

b) Population Data from Wikipedia:

Once we get the NewYork Geospatial data, population data is added to the respective Neighborhoods

```
[ ] NY_df.set_index('Neighborhood')
    NY_population_df.set_index('Neighborhood')
    NYC_df = pd.merge(NY_df, NY_population_df, how="inner", on=["Borough", "Neighborhood"])
    NYC_df.head()
```

	Borough	Neighborhood	Latitude	Longitude	Population
0	Bronx	Wakefield	40.894705	-73.847201	29158
1	Bronx	Co-op City	40.874294	-73.829939	43752
2	Bronx	Fieldston	40.895437	-73.905643	3292
3	Bronx	Riverdale	40.890834	-73.912585	48049
4	Bronx	Kingsbridge	40.881687	-73.902818	10669

c) Get location data using Foursquare

Foursquare API is a very useful online application used by many developers & other applications like Uber etc. In this project it is used to retrieve information about the places present in the neighborhoods of New York City. The API returns a JSON file and we need to turn that into a data-frame. Here 100 popular venues with a radius of 1000 mts have been chosen.

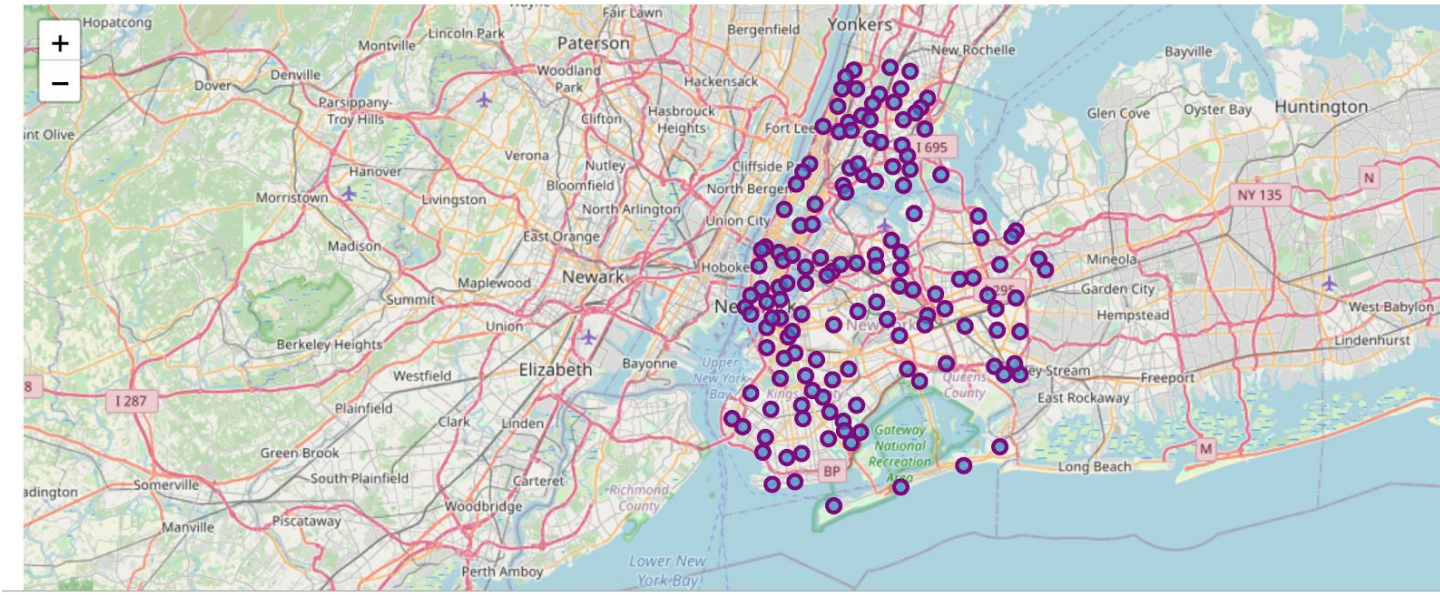
	id	name	categories	referralId	hasPerk	location.address	location.crossStreet	location.lat	location.lng
0	5984eba61499463427cadb0e	BoltBus 429 11th Ave	[{'id': '52f2ab2ebcbc57f1066b8b4f', 'name': 'B...	v-1593544624	False	429 11th Ave.	north of 35th St.	40.756702	-74.001120
1	5c9686fa95d986002c8df4e9	Finn's Bagels	[{'id': '4bf58dd8d48988d179941735', 'name': 'B...	v-1593544624	False	477 10th Ave	36th Street	40.756218	-73.997867
2	4283ee00f964a5209d221fe3	Jacob K. Javits Convention Center	[{'id': '4bf58dd8d48988d1ff931735', 'name': 'C...	v-1593544624	False	655 W 34th St	at 11th Ave	40.757140	-74.002404
3	4c071e5e8b4520a1e98c8697	Lobby @ 505 West 37	[{'id': '4bf58dd8d48988d130941735', 'name': 'B...	v-1593544624	False	505 W 37th St	10th ave	40.757220	-73.998260
4	4dc433ced4c0ad9c0f6880d4	Bolt Bus - W 34 St & 8 Av (Philadelphia/Boston)	[{'id': '52f2ab2ebcbc57f1066b8b4f', 'name': 'B...	v-1593544624	False	W 34th St	btwn 8th & 9th Ave	40.755688	-74.003738

Visualization using Maps and Plots:

For the below visualization done by using coordinates, Python's Folium library has been used.

```
[ ]  NYC_map = folium.Map(location=[latitude, longitude], zoom_start=10)
     for lat, lng, borough, neighborhood in zip(NYC_df['Latitude'], NYC_df['Longitude'], NYC_df['Borough'], NYC_df['Neighborhood']):
         label = '{},{}'.format(neighborhood, borough)
         label = folium.Popup(label, parse_html=True)
         folium.CircleMarker(
             [lat, lng],
             radius=5,
             popup=label,
             color='Purple',
             fill=True,
             fill_color='#3186cc',
             fill_opacity=0.7,
             parse_html=False).add_to(NYC_map)

     NYC_map
```



One hot encoding will be performed for knowing the venue categories, then that dataframe will be merged with the NYC DataFrame with latitude & longitude information on neighborhood. Finally extract just the Indian restaurant values along with neighborhood information. Code snippets are provided below for the same:

```
[ ] onehot_NYC = pd.get_dummies(NYC_venues[['Venue Category']], prefix="", prefix_sep="")

    onehot_NYC['Neighborhood'] = NYC_venues['Neighborhood']

    fixed_columns = [onehot_NYC.columns[-1]] + list(onehot_NYC.columns[:-1])
    onehot_NYC = onehot_NYC[fixed_columns]
    Grouped_NYC = onehot_NYC.groupby('Neighborhood').mean().reset_index()
    Grouped_NYC
```

	Neighborhood	Yoga Studio	Accessories Store	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment
0	Arverne	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	Bath Beach	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	Battery Park City	0.000000	0.000000	0.000000	0.000000	0.015625	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	Bay Ridge	0.000000	0.000000	0.000000	0.000000	0.036145	0.000000	0.000000	0.000000	0.000000	0.012048	0.000000	0.000000	0.000000
4	Bay Terrace	0.000000	0.025641	0.000000	0.000000	0.051282	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

```
[ ] NYC_area = Grouped_NYC[['Neighborhood', 'Indian Restaurant']]
    NYC_area
```

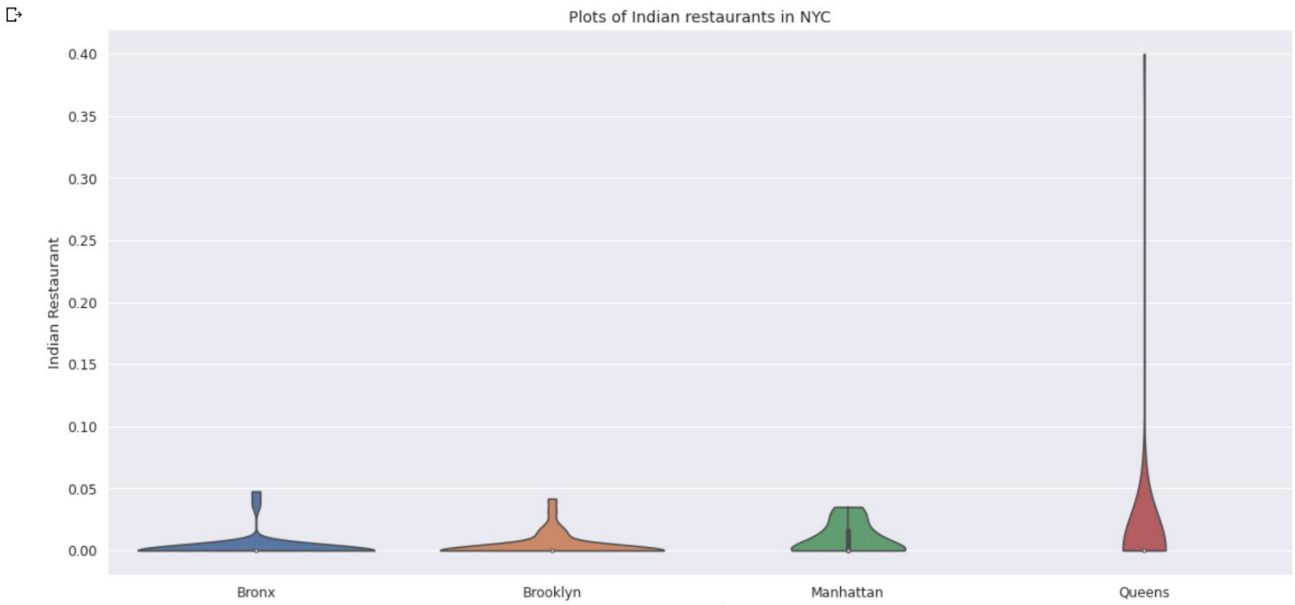
	Neighborhood	Indian Restaurant
0	Arverne	0.000000
1	Bath Beach	0.000000
2	Battery Park City	0.000000
3	Bay Ridge	0.012048
4	Bay Terrace	0.000000
5	Baychester	0.000000
6	Bayside	0.039474
7	Bedford Park	0.000000
8	Belmont	0.000000
9	Bensonhurst	0.000000

```
[ ] Merged_NYC = pd.merge(NYC_df, NYC_area, on='Neighborhood')
    Merged_NYC
```

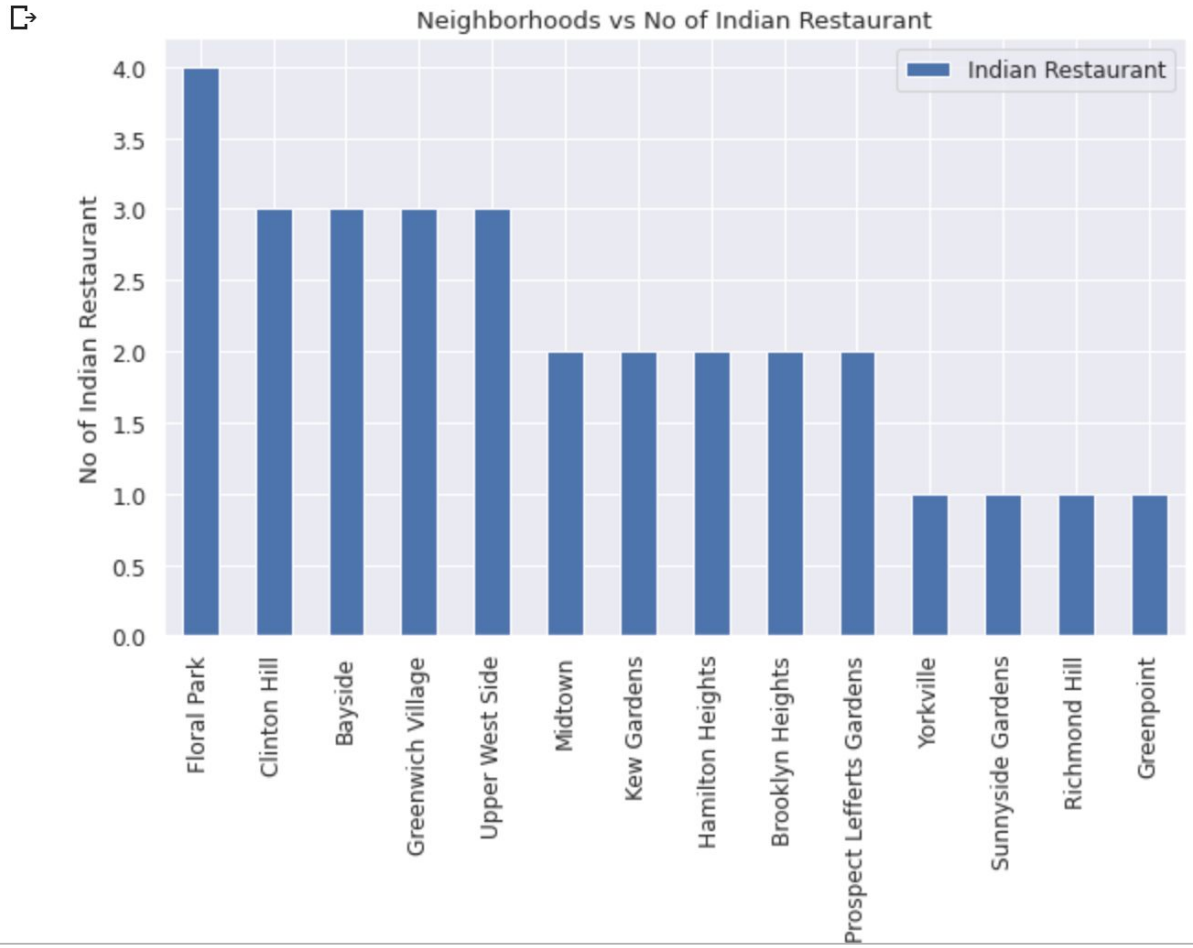
	Borough	Neighborhood	Latitude	Longitude	Population	Indian Restaurant
0	Bronx	Wakefield	40.894705	-73.847201	29158	0.000000
1	Bronx	Co-op City	40.874294	-73.829939	43752	0.000000
2	Bronx	Fieldston	40.895437	-73.905643	3292	0.000000
3	Bronx	Riverdale	40.890834	-73.912585	48049	0.000000
4	Bronx	Kingsbridge	40.881687	-73.902818	10669	0.000000
5	Bronx	Woodlawn	40.898273	-73.867315	42483	0.037037
6	Bronx	Norwood	40.877224	-73.879391	40494	0.000000
7	Bronx	Williamsbridge	40.881039	-73.857446	61321	0.000000
8	Bronx	Baychester	40.866858	-73.835798	63345	0.000000
9	Bronx	Pelham Parkway	40.857413	-73.854756	30073	0.000000
10	Bronx	Bedford Park	40.870185	-73.885512	37344	0.000000

Using Plots:

With the help of the violin plot we can identify the boroughs with densely populated Indian restaurants. It is drawn using a seaborn library to show the distribution of Indian restaurants in different boroughs.



Using Barplot Indian Restaurants available in each neighborhood can be explored.



Examining the Clusters:

Even before examining the clusters, the neighborhoods without any Indian restaurants have also been explored.

There are 115 such neighborhoods without any Indian restaurant. Below is the code snippet for the same.

```
[ ] df_rest_counts =  neighborhood_restaurants.groupby(['Neighborhood']).count().rename(columns={"Venue Category": "RestaurantCount"})[['RestaurantCount']]

    #find neighborhoods that does not have any restaurant
    noRestList = list(set(NYC_venues['Neighborhood']) - set(neighborhood_restaurants['Neighborhood']))

    #if exists , append neighborhoods without any restaurant to df_rest_counts
    if noRestList != []:
        df_rest_counts = df_rest_counts.append (pd.DataFrame( {'Neighborhood' : noRestList  , 'RestaurantCount': [0] * len(noRestList) } ).set_index('Neighborho

    df_rest_counts.reset_index(inplace=True)


    df_Ind_rest_counts =  Indian_restaurants.groupby(['Neighborhood']).count().rename(columns={"Venue Category": "IndRestaurantCount"})[['IndRestaurantCount']]

    #find neighborhoods that does not have any restaurant
    noRestList = list(set(NYC_venues['Neighborhood']) - set(Indian_restaurants['Neighborhood']))

    #if exists , append neighborhoods without any restaurant
    if noRestList != []:
        df_Ind_rest_counts = df_Ind_rest_counts.append (pd.DataFrame( {'Neighborhood' : noRestList  , 'IndRestaurantCount': [0] * len(noRestList) } ).set_index(

    df_Ind_rest_counts.reset_index(inplace=True)
    df_rest_counts= df_rest_counts.merge(df_Ind_rest_counts).set_index('Neighborhood')
    df_rest_counts= df_rest_counts.sort_values(by=['RestaurantCount'],ascending =False)

    print('{} neighborhoods do not have any Indian restaurant'.format(len(noRestList)))

115 neighborhoods do not have any Indian restaurant
```

A new dataframe has been created with top 10 venues for each neighborhood.

	Neighborhood	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
0	Arverne	Surf Spot	Sandwich Place	Metro Station	Donut Shop	Beach	Coffee Shop	Board Shop	Thai Restaurant	Café	Restaurant
1	Bath Beach	Pizza Place	Pharmacy	Chinese Restaurant	Gas Station	Bubble Tea Shop	Donut Shop	Italian Restaurant	Cantonese Restaurant	Fast Food Restaurant	Surf Spot
2	Battery Park City	Park	Hotel	Coffee Shop	Gym	Memorial Site	Sandwich Place	Shopping Mall	Food Court	Playground	Mexican Restaurant
3	Bay Ridge	Spa	Pizza Place	Italian Restaurant	Greek Restaurant	Pharmacy	Bar	Bagel Shop	American Restaurant	Hookah Bar	Thai Restaurant
4	Bay Terrace	Clothing Store	Women's Store	Donut Shop	Kids Store	Lingerie Store	Cosmetics Shop	Mobile Phone Shop	American Restaurant	Shoe Store	Gym
5	Baychester	Donut Shop	Mattress Store	Cosmetics Shop	Electronics Store	Bank	Mexican Restaurant	Shopping Mall	Fried Chicken Joint	Pizza Place	Convenience Store
6	Bayside	Bar	Chinese Restaurant	Pizza Place	American Restaurant	Indian Restaurant	Greek Restaurant	Sushi Restaurant	Spa	Bakery	Italian Restaurant

With these venues, 5 clusters have been formed 0,1,2,3,4 and each cluster has been examined.

Cluster 1:

Cluster 1 especially likes Fast Food, burger Joints, this part may not like Indian tastes .

Cluster 1

```
[ ] Merged_NYC.loc[Merged_NYC['Cluster Labels'] == 0, Merged_NYC.columns[[1] + list(range(5, Merged_NYC.shape[1]))]]
```

	Neighborhood	Cluster Labels	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	Co-op City	0	Fast Food Restaurant	Baseball Field	Restaurant	Bagel Shop	Park	Pizza Place	Supermarket	Pharmacy	Grocery Store	Bus Station
4	Kingsbridge	0	Pizza Place	Bar	Supermarket	Mexican Restaurant	Sandwich Place	Latin American Restaurant	Bakery	Spanish Restaurant	Donut Shop	Pharmacy
5	Woodlawn	0	Pizza Place	Food & Drink Shop	Playground	Deli / Bodega	Pub	Rental Car Location	Grocery Store	Liquor Store	Bar	Bakery
7	Williamsbridge	0	Bar	Soup Place	Nightclub	Caribbean Restaurant	Farmers Market	Fast Food Restaurant	Filipino Restaurant	Fish & Chips Shop	Fish Market	Flea Market
8	Baychester	0	Donut Shop	Mattress Store	Cosmetics Shop	Electronics Store	Bank	Mexican Restaurant	Shopping Mall	Fried Chicken Joint	Pizza Place	Convenience Store

Cluster 2:

Cluster 2 has an amalgam of restaurants.

Cluster 2

```
[ ] Merged_NYC.loc[Merged_NYC['Cluster Labels'] == 1, Merged_NYC.columns[[1] + list(range(5, Merged_NYC.shape[1]))]]
```

	Neighborhood	Cluster Labels	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
11	University Heights	1	Latin American Restaurant	Pizza Place	Sandwich Place	Convenience Store	Fried Chicken Joint	Supermarket	Fast Food Restaurant	Grocery Store	Shoe Store	Pharmacy
14	East Tremont	1	Pizza Place	Women's Store	Supermarket	Breakfast Spot	Café	Mobile Phone Shop	Food	Flea Market	Fast Food Restaurant	Paella Restaurant
17	Mott Haven	1	Gym	Spanish Restaurant	Pizza Place	Donut Shop	Mobile Phone Shop	Peruvian Restaurant	Latin American Restaurant	Bakery	Chinese Restaurant	Burger Joint
20	Hunts Point	1	Pizza Place	Juice Bar	BBQ Joint	Grocery Store	Café	Restaurant	Bank	Waste Facility	Farmers Market	Spanish Restaurant
21	Morrisania	1	Bus Station	Discount Store	Fast Food Restaurant	Liquor Store	Donut Shop	Seafood Restaurant	Fish Market	Pharmacy	Latin American Restaurant	Bowling Alley
29	Castle Hill	1	Market	Latin American Restaurant	Diner	Pharmacy	Bank	Cosmetics Shop	Pizza Place	Falafel Restaurant	Farmers Market	Fast Food Restaurant

Cluster 3:

Cluster 3 is a center full of restaurants.

Cluster 3

```
[ ] Merged_NYC.loc[Merged_NYC['Cluster Labels'] == 2, Merged_NYC.columns[[1] + list(range(5, Merged_NYC.shape[1]))]]
```

	Neighborhood	Cluster Labels	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
86	Elmhurst	2	Thai Restaurant	Mexican Restaurant	Chinese Restaurant	Bubble Tea Shop	Vietnamese Restaurant	Sushi Restaurant	Malay Restaurant	Park	Bar	Bank

Cluster 4:

Cluster 4 is also full of restaurants, just like cluster 3.

Cluster 4

```
[ ] Merged_NYC.loc[Merged_NYC['Cluster Labels'] == 3, Merged_NYC.columns[[1] + list(range(5, Merged_NYC.shape[1]))]]
```

	Neighborhood	Cluster Labels	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
0	Wakefield	3	Pharmacy	Food	Laundromat	Dessert Shop	Gas Station	Ice Cream Shop	Donut Shop	Sandwich Place	Falafel Restaurant	Farmers Market
2	Fieldston	3	River	Bus Station	Business Service	Plaza	Women's Store	Food	Falafel Restaurant	Farmers Market	Fast Food Restaurant	Filipino Restaurant
3	Riverdale	3	Park	Bus Station	Baseball Field	Home Service	Bank	Gym	Plaza	Food Truck	Food Stand	Farmers Market
6	Norwood	3	Park	Pizza Place	Chinese Restaurant	Bank	Deli / Bodega	Pharmacy	Bus Station	Grocery Store	Fast Food Restaurant	Coffee Shop

Cluster 5:

Cluster 5 suffers from restaurants. Especially there is no Indian restaurant.

Cluster 5

```
Merged_NYC.loc[Merged_NYC['Cluster Labels'] == 4, Merged_NYC.columns[[1] + list(range(5, Merged_NYC.shape[1]))]]
```

	Neighborhood	Cluster Labels	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
93	Sunnyside	4	Pizza Place	Chinese Restaurant	Italian Restaurant	Discount Store	Bakery	Deli / Bodega	Mexican Restaurant	Coffee Shop	South American Restaurant	Fast Food Restaurant

Conclusion

We have reached the end of the analysis, in this section we will document all the findings from above clustering & visualization of the dataset. In this project, we started off with the business problem of identifying a good neighborhood to open a new Indian restaurant. To achieve that we looked into all the neighborhoods in New York City, analysed each neighborhood & number of Indian restaurants in those neighborhoods to come to conclusion about which neighborhood would be a better spot. We

have used a variety of data sources to set up a very realistic data-analysis scenario. Below are the findings:

- It has been identified that only Queens, Manhattan, Brooklyn and Bronx have a high amount of Indian restaurants with the help of Violin plots between Number of Indian restaurants in the Borough of New York City.
- With the help of clusters examining & violin plots looks like Downtown Toronto, Central Toronto, East York are already densely populated with Indian restaurants. So it is better idea to leave those boroughs out and consider only Scarborough, East Toronto & North York for the new restaurant's location.
- After careful consideration it is a good idea to open a new Indian restaurant in the Sunnyside neighborhood since it has no Indian restaurants in the neighborhoods and also suffers from other restaurants too, which gives a higher number of customers and possibly lower competition.

According to this analysis, the Sunnyside neighborhood will provide the least competition for the new upcoming Indian restaurant as there is very little Indian restaurants spread or no Indian restaurants in few neighborhoods. So, definitely this region could potentially be a perfect place for starting a quality Indian restaurant. Some of the drawbacks of this analysis are — the clustering is completely based only on data obtained from Foursquare API.Even Though there are lots of areas where it can be improved yet this analysis has certainly provided us with some good insights, preliminary information on possibilities & a head start into this business problem by setting the step stones properly.

Finally to conclude this project, We have got a chance to win a business problem like how a real like data scientists would do. We have used many python libraries to fetch the data , to manipulate the contents & to analyze and visualize those datasets. We have made use of Foursquare API to explore the venues in neighborhoods of New York City, then got a good amount of data from Wikipedia which we scraped with help of Wikipedia python library and visualized using various plots present in seaborn & matplotlib. We also applied machine learning techniques to predict the output given the data and used Folium to visualize it on a map.