

Capstone Project - The Battle of Neighborhoods

Introduction

New York City's demographics show that it is a large and ethnically diverse metropolis. It is the largest city in the United States with a long history of international immigration. New York City was home to nearly 8.5 million people in 2014, accounting for over 40% of the population of New York State and a slightly lower percentage of the New York metropolitan area, home to approximately 23.6 million. Over the last decade the city has been growing faster than the region. The New York region continues to be by far the leading metropolitan gateway for legal immigrants admitted into the United States.

Throughout its history, New York City has been a major point of entry for immigrants; the term "melting pot" was coined to describe densely populated immigrant neighborhoods on the Lower East Side. As many as 800 languages are spoken in New York, making it the most linguistically diverse city in the world. English remains the most widely spoken language, although there are areas in the outer boroughs in which up to 25% of people speak English as an alternate language, and/or have limited or no English language fluency. English is least spoken in neighborhoods such as Flushing, Sunset Park, and Corona.

With its diverse culture, comes diverse food items. There are many restaurants in New York City, each belonging to different categories like Chinese, Indian, French etc.

So as part of this project, we will list and visualize all major parts of New York City that has great Indian restaurants.

Data ¶

For this project we need the following data :

- New York City data that contains list Boroughs, Neighborhoods along with their latitude and longitude.
 - Data source : https://cocl.us/new_york_dataset
 - Description : This data set contains the required information. And we will use this data set to explore various neighborhoods of New York City.
- Indian restaurants in each neighborhood of New York City.
 - Data source : Foursquare API
 - Description : By using this API we will get all the venues in each neighborhood. We can filter these venues to get only Indian restaurants.
- GeoSpace data
 - Data source : <https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm>
 - Description : By using this geo space data we will get the New York Borough boundaries that will help us visualize choropleth map.

Approach

- Collect the New York City data from https://cocl.us/new_york_dataset (https://cocl.us/new_york_dataset)
- Using Foursquare API we will find all venues for each neighborhood.
- Filter out all venues that are Indian Restaurants.
- Find rating, tips and like count for each Indian Restaurant using Foursquare API.
- Using rating for each restaurant, we will sort that data.

- Visualize the Ranking of neighborhoods using folium library(python)

Questions that can be asked using the above mentioned datasets

- What is best location in New York City for Indian Cuisine ?
- Which areas have potential Indian Restaurant Market ?
- Which all areas lack Indian Restaurants ?
- Which is the best place to stay if I prefer Indian Cuisine ?

Analysis

We will import the required libraries for python.

- pandas and numpy for handling data.
- request module for using FourSquare API.
- geopy to get co-ordinates of City of New York.
- folium to visualize the results on a map

In [206]:

```
import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import requests
from bs4 import BeautifulSoup
import geocoder
import os
import folium # map rendering library
from geopy.geocoders import Nominatim # convert an address into latitude and longitude
# Matplotlib and associated plotting modules
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
%matplotlib inline

print('Libraries imported.')
```

Libraries imported.

Now we define a function to get the geocodes i.e latitude and longitude of a given location using geopy.

In [207]:

```
def geo_location(address):
    # get geo location of address
    geolocator = Nominatim(user_agent="ny_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    return latitude, longitude
```

We define a function to interact with FourSquare API and get top 100 venues within a radius of 1000 metres for a given latitude and longitude. Below function will return us the venue id , venue name and category.

In [208]:

```
def get_venues(lat,lng):

    #set variables
    radius=1000
    LIMIT=100
    CLIENT_ID = os.environ['CLIENT_ID'] # your Foursquare ID
    CLIENT_SECRET = os.environ['CLIENT_SECRET'] # your Foursquare Secret
    VERSION = '20180605' # Foursquare API version

    #url to fetch data from foursquare api
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&client_secret={}&version={}&lat={}&lng={}&radius={}&limit={}'
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        lng,
        radius,
        LIMIT)

    # get all the data
    results = requests.get(url).json()
    venue_data=results["response"]["groups"][0]['items']
    venue_details=[]
    for row in venue_data:
        try:
            venue_id=row['venue']['id']
            venue_name=row['venue']['name']
            venue_category=row['venue']['categories'][0]['name']
            venue_details.append([venue_id,venue_name,venue_category])
        except KeyError:
            pass

    column_names=['ID','Name','Category']
    df = pd.DataFrame(venue_details,columns=column_names)
    return df
```

Now we will define a function to get venue details like like count , rating , tip counts for a given venue id. This will be used for ranking.

In [209]:

```
def get_venue_details(venue_id):

    CLIENT_ID = os.environ['CLIENT_ID'] # your Foursquare ID
    CLIENT_SECRET = os.environ['CLIENT_SECRET'] # your Foursquare Secret
    VERSION = '20180605' # Foursquare API version

    #url to fetch data from foursquare api
    url = 'https://api.foursquare.com/v2/venues/{}?&client_id={}&client_secret={}&v
        venue_id,
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION)

    # get all the data
    results = requests.get(url).json()
    venue_data=results['response']['venue']
    venue_details=[]
    try:
        venue_id=venue_data['id']
        venue_name=venue_data['name']
        venue_likes=venue_data['likes']['count']
        venue_rating=venue_data['rating']
        venue_tips=venue_data['tips']['count']
        venue_details.append([venue_id,venue_name,venue_likes,venue_rating,venue_tips])
    except KeyError:
        pass

    column_names=['ID','Name','Likes','Rating','Tips']
    df = pd.DataFrame(venue_details,columns=column_names)
    return df
```

Now we define a function to get the New York City data such as Boroughs, Neighborhoods along with their latitude and longitude.

In [210]:

```
def get_new_york_data():
    url='https://cocl.us/new_york_dataset'
    resp=requests.get(url).json()
    # all data is present in features label
    features=resp['features']

    # define the dataframe columns
    column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']
    # instantiate the dataframe
    new_york_data = pd.DataFrame(columns=column_names)

    for data in features:
        borough = data['properties']['borough']
        neighborhood_name = data['properties']['name']

        neighborhood_latlon = data['geometry']['coordinates']
        neighborhood_lat = neighborhood_latlon[1]
        neighborhood_lon = neighborhood_latlon[0]

        new_york_data = new_york_data.append({'Borough': borough,
                                              'Neighborhood': neighborhood_name,
                                              'Latitude': neighborhood_lat,
                                              'Longitude': neighborhood_lon}, ignore_in

    return new_york_data
```

We will call the above funtion to get the new york city data.

In [211]:

```
# get new york data
new_york_data=get_new_york_data()
```

In [212]:

```
new_york_data.head()
```

Out[212]:

| | 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 |

In [213]:

```
new_york_data.shape
```

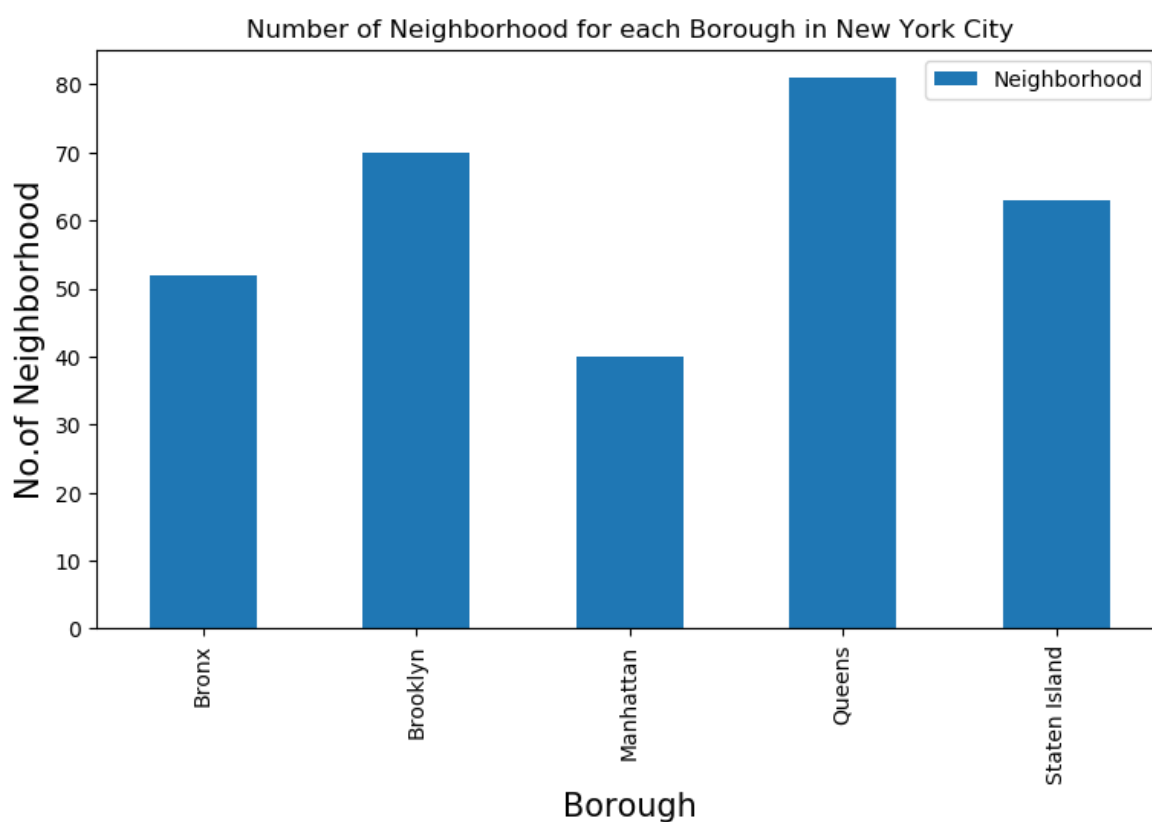
Out[213]:

(306, 4)

So there are total of 306 different Neighborhoods in New York

In [219]:

```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Neighborhood for each Borough in New York City')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('No.of Neighborhood', fontsize=15)
#giving a bar plot
new_york_data.groupby('Borough')['Neighborhood'].count().plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```



We see that Queens has highest number of neighborhoods

Now we will collect Indian restaurants for each Neighborhood

In [230]:

```
# prepare neighborhood list that contains indian resturants
column_names=['Borough', 'Neighborhood', 'ID', 'Name']
indian_rest_ny=pd.DataFrame(columns=column_names)
count=1
for row in new_york_data.values.tolist():
    Borough, Neighborhood, Latitude, Longitude=row
    venues = get_venues(Latitude,Longitude)
    indian_resturants=venues[venues['Category']=='Indian Restaurant']
    print('(',count, '/', len(new_york_data), ')', 'Indian Restaurants in '+Neighborhood)
    for resturant_detail in indian_resturants.values.tolist():
        id, name , category=resturant_detail
        indian_rest_ny = indian_rest_ny.append({'Borough': Borough,
                                                'Neighborhood': Neighborhood,
                                                'ID': id,
                                                'Name' : name
                                                }, ignore_index=True)

    count+=1
```

```
( 287 / 306 ) Indian Restaurants in Sandy Ground, Staten Island:0
( 288 / 306 ) Indian Restaurants in Egbertville, Staten Island:0
( 289 / 306 ) Indian Restaurants in Roxbury, Queens:0
( 290 / 306 ) Indian Restaurants in Homecrest, Brooklyn:0
( 291 / 306 ) Indian Restaurants in Middle Village, Queens:0
( 292 / 306 ) Indian Restaurants in Prince's Bay, Staten Island:0
( 293 / 306 ) Indian Restaurants in Lighthouse Hill, Staten Island:0
( 294 / 306 ) Indian Restaurants in Richmond Valley, Staten Island:0
( 295 / 306 ) Indian Restaurants in Malba, Queens:0
( 296 / 306 ) Indian Restaurants in Highland Park, Brooklyn:0
( 297 / 306 ) Indian Restaurants in Madison, Brooklyn:0
( 298 / 306 ) Indian Restaurants in Bronxdale, Bronx:0
( 299 / 306 ) Indian Restaurants in Allerton, Bronx:0
( 300 / 306 ) Indian Restaurants in Kingsbridge Heights, Bronx:0
( 301 / 306 ) Indian Restaurants in Erasmus, Brooklyn:1
( 302 / 306 ) Indian Restaurants in Hudson Yards, Manhattan:0
( 303 / 306 ) Indian Restaurants in Hammels, Queens:0
( 304 / 306 ) Indian Restaurants in Bayswater, Queens:0
( 305 / 306 ) Indian Restaurants in Queensbridge, Queens:2
( 306 / 306 ) Indian Restaurants in Fox Hills, Staten Island:1
```

Now that we have got all the indian resturants in new york city , we will analyze it

In [232]:

```
indian_rest_ny.head()
```

Out[232]:

| | Borough | Neighborhood | ID | Name |
|---|---------|----------------|--------------------------|-----------------------------|
| 0 | Bronx | Riverdale | 4c04544df423a593ac83d116 | Cumin Indian Cuisine |
| 1 | Bronx | Kingsbridge | 4c04544df423a593ac83d116 | Cumin Indian Cuisine |
| 2 | Bronx | Woodlawn | 4c0448d9310fc9b6bf1dc761 | Curry Spot |
| 3 | Bronx | Parkchester | 4c194631838020a13e78e561 | Melanies Roti Bar And Grill |
| 4 | Bronx | Spuyten Duyvil | 4c04544df423a593ac83d116 | Cumin Indian Cuisine |

In [233]:

```
indian_rest_ny.shape
```

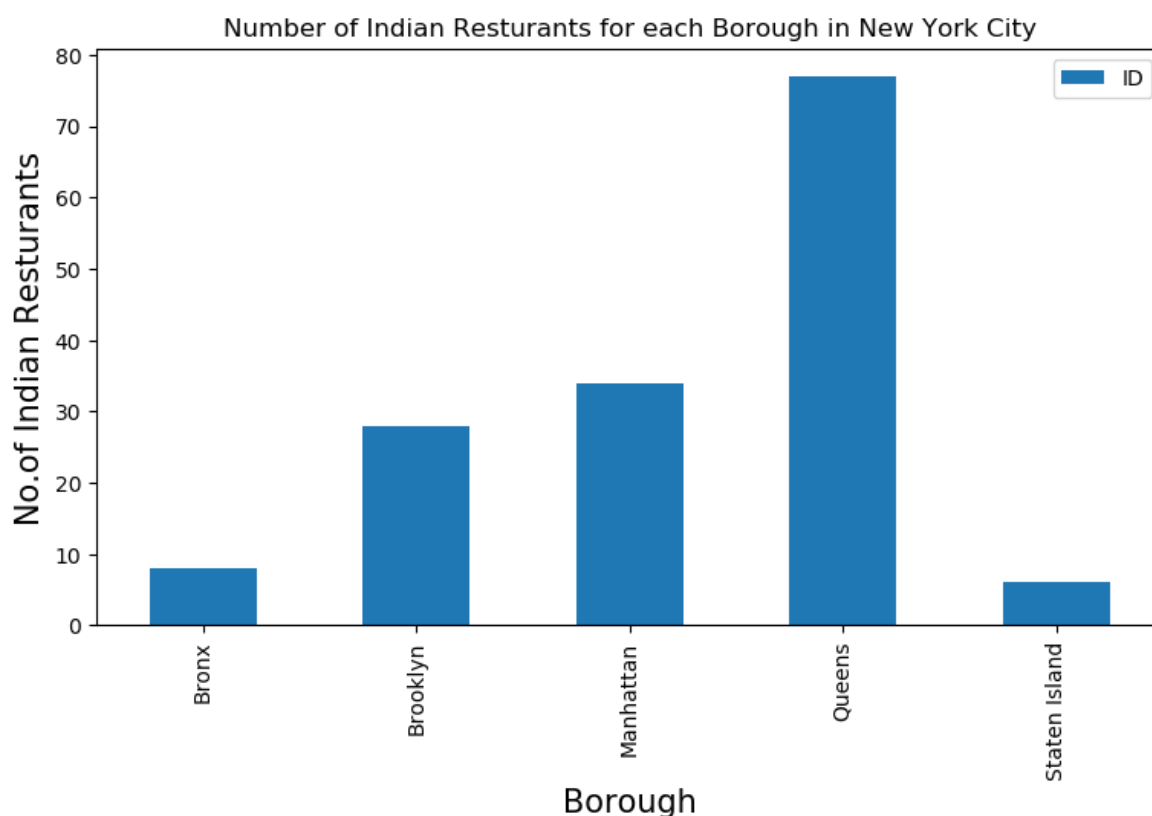
Out[233]:

(153, 4)

We got 153 Indian Resturants across New York City

In [234]:

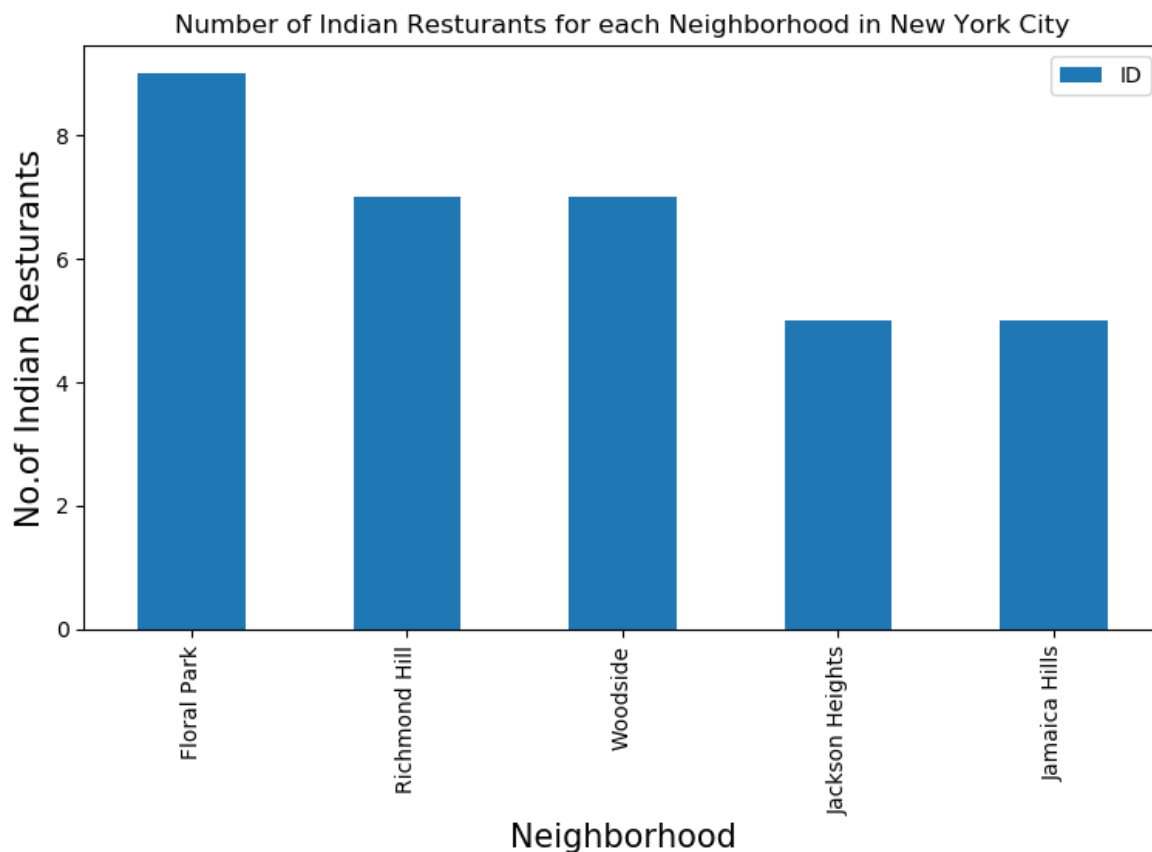
```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Indian Resturants for each Borough in New York City')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('No.of Indian Resturants', fontsize=15)
#giving a bar plot
indian_rest_ny.groupby('Borough')['ID'].count().plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```



We see that Queens has the largest number of indian resturants

In [236]:

```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Indian Resturants for each Neighborhood in New York City')
#On x-axis
plt.xlabel('Neighborhood', fontsize = 15)
#On y-axis
plt.ylabel('No.of Indian Resturants', fontsize=15)
#giving a bar plot
indian_rest_ny.groupby('Neighborhood')['ID'].count().nlargest(5).plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```



In [238]:

```
indian_rest_ny[indian_rest_ny['Neighborhood']=='Floral Park']
```

Out[238]:

| | Borough | Neighborhood | ID | Name |
|-----|---------|--------------|--------------------------|---------------------------------|
| 103 | Queens | Floral Park | 527ffc0811d2d329d5e49abd | Jackson Diner |
| 104 | Queens | Floral Park | 4b647b56f964a520c4b62ae3 | Usha Foods & Usha Sweets |
| 105 | Queens | Floral Park | 4b787c49f964a5209cd12ee3 | Santoor Indian Restaurant |
| 106 | Queens | Floral Park | 4e4e3e22bd4101d0d7a5c2d1 | Kerala Kitchen |
| 107 | Queens | Floral Park | 4c0c01e0bbc676b00d6b4cd5 | Mumbai Xpress |
| 108 | Queens | Floral Park | 4c76ff35a5676dcb72671721 | Flavor Of India |
| 109 | Queens | Floral Park | 4df0f39dd4c04d0392c853ea | Sagar Chinese |
| 110 | Queens | Floral Park | 571af96a498e9e392d8d3786 | Namaste Authenic Indian Cuisine |
| 111 | Queens | Floral Park | 51d84192498ea979a3c4f13d | Sunshine Grill & Restaurant |

So Floral Park in Queens has the highest number of Indian Resturants with a total count of 9.

Now we will get the ranking of each resturant for further analysis.

In [260]:

```
# prepare neighborhood list that contains indian resturants
column_names=['Borough', 'Neighborhood', 'ID', 'Name', 'Likes', 'Rating', 'Tips']
indian_rest_stats_ny=pd.DataFrame(columns=column_names)
count=1

for row in indian_rest_ny.values.tolist():
    Borough,Neighborhood,ID,Name=row
    try:
        venue_details=get_venue_details(ID)
        print(venue_details)
        id,name,likes,rating,tips=venue_details.values.tolist()[0]
    except IndexError:
        print('No data available for id=',ID)
        # we will assign 0 value for these resturants as they may have been
        #recently opened or details does not exist in FourSquare Database
        id,name,likes,rating,tips=[0]*5
    print('(',count,',',len(indian_rest_ny),')', 'processed')
    indian_rest_stats_ny = indian_rest_stats_ny.append({'Borough': Borough,
                                                         'Neighborhood': Neighborhood,
                                                         'ID': id,
                                                         'Name' : name,
                                                         'Likes' : likes,
                                                         'Rating' : rating,
                                                         'Tips' : tips
                                                         }, ignore_index=True)

    count+=1
```

```

          ID  Name  Likes  Rating  Tips
0  5b931ea69d7468002c3b1382  Adda      71      9.2      20
( 149 / 153 ) processed
          ID  Name  Likes  Rating  Tips
0  564d283d498e6e851df79d87  Great Indian Curry      3      6.7      2
( 150 / 153 ) processed
          ID  Name  Likes  Rating  Tips
0  4b1b341bf964a5208af923e3  Five Star Banquet     29      7.4     31
( 151 / 153 ) processed
          ID  Name  Likes  Rating  Ti
ps
0  50a287a7e4b0033f830f06db  Raj's Indian Kitchen     21      7.2
9
( 152 / 153 ) processed

Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 4b65f2e3f964a5206e0a2be3
( 153 / 153 ) processed
```

In [261]:

```
indian_rest_stats_ny.head()
```

Out[261]:

| | Borough | Neighborhood | ID | Name | Likes | Rating | Tips |
|---|---------|----------------|--------------------------|-----------------------------|-------|--------|------|
| 0 | Bronx | Riverdale | 4c04544df423a593ac83d116 | Cumin Indian Cuisine | 13 | 6.6 | 9 |
| 1 | Bronx | Kingsbridge | 4c04544df423a593ac83d116 | Cumin Indian Cuisine | 13 | 6.6 | 9 |
| 2 | Bronx | Woodlawn | 4c0448d9310fc9b6bf1dc761 | Curry Spot | 4 | 7.7 | 10 |
| 3 | Bronx | Parkchester | 4c194631838020a13e78e561 | Melanies Roti Bar And Grill | 3 | 6.1 | 2 |
| 4 | Bronx | Spuyten Duyvil | 4c04544df423a593ac83d116 | Cumin Indian Cuisine | 13 | 6.6 | 9 |

In [265]:

```
indian_rest_stats_ny.shape
```

Out[265]:

(153, 7)

In [266]:

```
indian_rest_ny.shape
```

Out[266]:

(153, 4)

So we got data for all resturants Now lets save this data to a csv sheet. In case we by mistake modify it. As the number of calls to get details for venue are premium call and have limit of 500 per day, we will refer to saved data sheet csv if required

In [267]:

```
indian_rest_stats_ny.to_csv('indian_rest_stats_ny.csv', index=False)
```

Lets verify the data from saved csv file

In [268]:

```
indian_rest_stats_ny_csv=pd.read_csv('indian_rest_stats_ny.csv')
```

In [269]:

```
indian_rest_stats_ny_csv.shape
```

Out[269]:

(153, 7)

In [270]:

```
indian_rest_stats_ny_csv.head()
```

Out[270]:

| | Borough | Neighborhood | ID | Name | Likes | Rating | Tips |
|---|---------|----------------|--------------------------|-----------------------------|-------|--------|------|
| 0 | Bronx | Riverdale | 4c04544df423a593ac83d116 | Cumin Indian Cuisine | 13 | 6.6 | 9 |
| 1 | Bronx | Kingsbridge | 4c04544df423a593ac83d116 | Cumin Indian Cuisine | 13 | 6.6 | 9 |
| 2 | Bronx | Woodlawn | 4c0448d9310fc9b6bf1dc761 | Curry Spot | 4 | 7.7 | 10 |
| 3 | Bronx | Parkchester | 4c194631838020a13e78e561 | Melanies Roti Bar And Grill | 3 | 6.1 | 2 |
| 4 | Bronx | Spuyten Duyvil | 4c04544df423a593ac83d116 | Cumin Indian Cuisine | 13 | 6.6 | 9 |

In [277]:

```
indian_rest_stats_ny.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 153 entries, 0 to 152
Data columns (total 7 columns):
Borough      153 non-null object
Neighborhood  153 non-null object
ID           153 non-null object
Name         153 non-null object
Likes        153 non-null object
Rating       153 non-null float64
Tips         153 non-null object
dtypes: float64(1), object(6)
memory usage: 8.4+ KB
```

We see that values like Likes, Tips are strig values. We would need to convert them into float for further analysis

In [279]:

```
indian_rest_stats_ny['Likes']=indian_rest_stats_ny['Likes'].astype('float64')
```

In [280]:

```
indian_rest_stats_ny['Tips']=indian_rest_stats_ny['Tips'].astype('float64')
```

In [283]:

```
indian_rest_stats_ny.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 153 entries, 0 to 152
Data columns (total 7 columns):
Borough      153 non-null object
Neighborhood  153 non-null object
ID           153 non-null object
Name         153 non-null object
Likes        153 non-null float64
Rating       153 non-null float64
Tips         153 non-null float64
dtypes: float64(3), object(4)
memory usage: 8.4+ KB
```

Now the data types looks correct

In [286]:

```
# Resturant with maximum Likes
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Likes'].idxmax()]
```

Out[286]:

```
Borough      Manhattan
Neighborhood  Midtown
ID           49d91c12f964a520015e1fe3
Name         The Kati Roll Company
Likes        819
Rating       9
Tips         257
Name: 43, dtype: object
```

In [287]:

```
# Resturant with maximum Rating
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Rating'].idxmax()]
```

Out[287]:

```
Borough      Manhattan
Neighborhood  Tribeca
ID           4bbb9dbded7776b0e1ad3e51
Name         Tamarind TriBeCa
Likes        566
Rating       9.2
Tips         141
Name: 45, dtype: object
```

In [288]:

```
# Resturant with maximum Tips
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Tips'].idxmax()]
```

Out[288]:

```
Borough                Manhattan
Neighborhood           Midtown
ID                    49d91c12f964a520015e1fe3
Name                  The Kati Roll Company
Likes                  819
Rating                 9
Tips                  257
Name: 43, dtype: object
```

Now lets visualize neighborhood with maximum average rating of resturants

In [374]:

```
ny_neighborhood_stats=indian_rest_stats_ny.groupby('Neighborhood',as_index=False).mean()
ny_neighborhood_stats.columns=['Neighborhood','Average Rating']
```

In [375]:

```
ny_neighborhood_stats.sort_values(['Average Rating'],ascending=False).head(10)
```

Out[375]:

| | Neighborhood | Average Rating |
|----|------------------|----------------|
| 0 | Astoria | 9.200000 |
| 71 | Sunnyside | 9.200000 |
| 75 | Tribeca | 9.200000 |
| 5 | Blissville | 9.200000 |
| 11 | Civic Center | 9.200000 |
| 47 | Midtown | 9.000000 |
| 48 | Midtown South | 9.000000 |
| 30 | Gramercy | 8.866667 |
| 66 | Roosevelt Island | 8.700000 |
| 53 | North Side | 8.700000 |

Above are the top neighborhoods with top average rating of Indian resturants

In [376]:

```
ny_borough_stats=indian_rest_stats_ny.groupby('Borough',as_index=False).mean()[['Bo
ny_borough_stats.columns=['Borough','Average Rating']
```

In [377]:

```
ny_borough_stats.sort_values(['Average Rating'],ascending=False).head()
```

Out[377]:

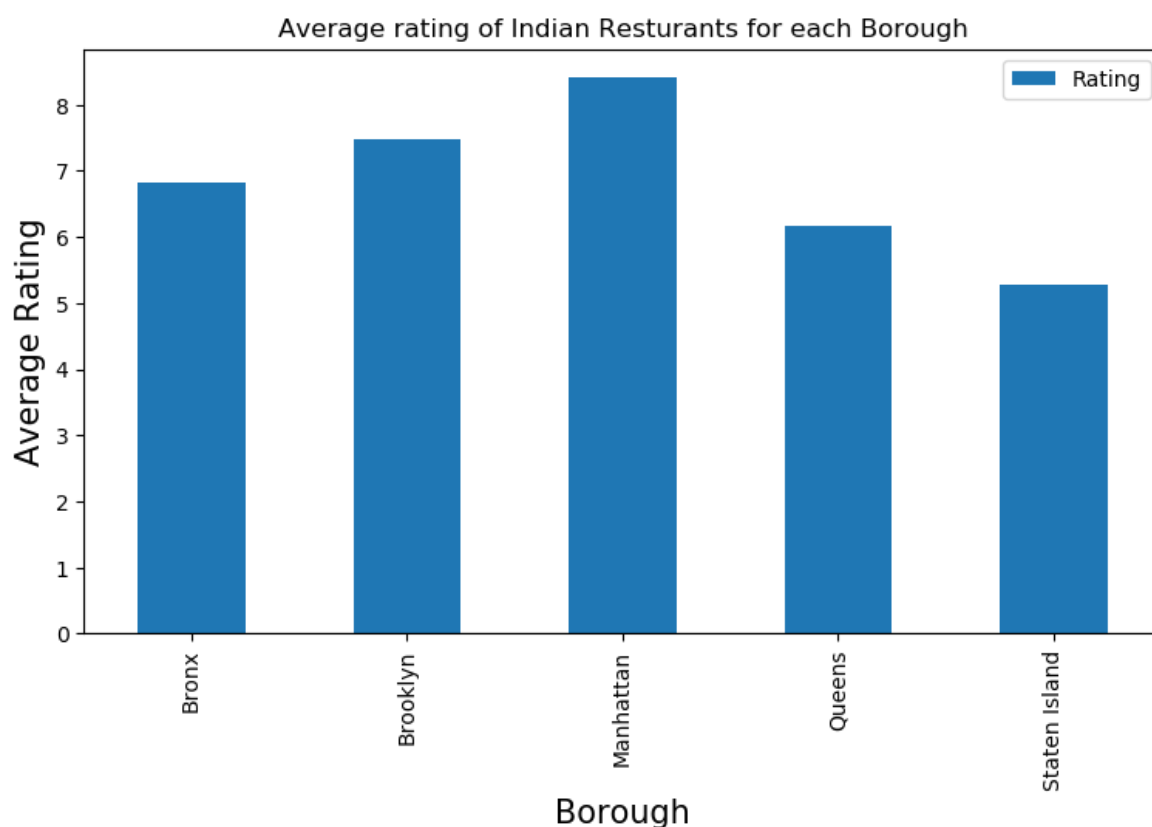
| | Borough | Average Rating |
|---|---------------|----------------|
| 2 | Manhattan | 8.414706 |
| 1 | Brooklyn | 7.478571 |
| 0 | Bronx | 6.812500 |
| 3 | Queens | 6.155844 |
| 4 | Staten Island | 5.266667 |

Similarly these are the average rating of Indian Resturants for each Borough

Lets visualize it

In [466]:

```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Average rating of Indian Resturants for each Borough')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('Average Rating', fontsize=15)
#giving a bar plot
indian_rest_stats_ny.groupby('Borough').mean()['Rating'].plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```



We will consider all the neighborhoods with average rating greater or equal 9.0 to visualize on map

In [472]:

```
ny_neighborhood_stats=ny_neighborhood_stats[ny_neighborhood_stats['Average Rating']
```

In [473]:

```
ny_neighborhood_stats
```

Out[473]:

| | Borough_x | Neighborhood | Latitude_x | Longitude_x | Average Rating | Label | Borough_y | La |
|---|---------------|---------------|------------|-------------|----------------|-------------------------------|---------------|----|
| 0 | Queens | Astoria | 40.768509 | -73.915654 | 9.2 | Astoria, Queens(9.2) | Queens | 40 |
| 1 | Queens | Blissville | 40.737251 | -73.932442 | 9.2 | Blissville, Queens(9.2) | Queens | 40 |
| 2 | Manhattan | Civic Center | 40.715229 | -74.005415 | 9.2 | Civic Center, Manhattan(9.2) | Manhattan | 40 |
| 3 | Manhattan | Midtown | 40.754691 | -73.981669 | 9.0 | Midtown, Manhattan(9.0) | Manhattan | 40 |
| 4 | Manhattan | Midtown South | 40.748510 | -73.988713 | 9.0 | Midtown South, Manhattan(9.0) | Manhattan | 40 |
| 5 | Queens | Sunnyside | 40.740176 | -73.926916 | 9.2 | Sunnyside, Queens(9.2) | Queens | 40 |
| 6 | Queens | Sunnyside | 40.740176 | -73.926916 | 9.2 | Sunnyside, Queens(9.2) | Staten Island | 40 |
| 7 | Staten Island | Sunnyside | 40.612760 | -74.097126 | 9.2 | Sunnyside, Staten Island(9.2) | Queens | 40 |
| 8 | Staten Island | Sunnyside | 40.612760 | -74.097126 | 9.2 | Sunnyside, Staten Island(9.2) | Staten Island | 40 |
| 9 | Manhattan | Tribeca | 40.721522 | -74.010683 | 9.2 | Tribeca, Manhattan(9.2) | Manhattan | 40 |

We will join this dataset to original new york data to get lonitude and latitude

In [474]:

```
ny_neighborhood_stats=pd.merge(ny_neighborhood_stats,new_york_data, on='Neighborhood')
```

In [475]:

```
ny_neighborhood_stats=ny_neighborhood_stats[['Borough','Neighborhood','Latitude','Longitude']]
```

In [476]:

```
ny_neighborhood_stats
```

Out[476]:

| | Borough | Neighborhood | Latitude | Longitude | Average Rating |
|----|---------------|---------------|-----------|------------|----------------|
| 0 | Queens | Astoria | 40.768509 | -73.915654 | 9.2 |
| 1 | Queens | Blissville | 40.737251 | -73.932442 | 9.2 |
| 2 | Manhattan | Civic Center | 40.715229 | -74.005415 | 9.2 |
| 3 | Manhattan | Midtown | 40.754691 | -73.981669 | 9.0 |
| 4 | Manhattan | Midtown South | 40.748510 | -73.988713 | 9.0 |
| 5 | Queens | Sunnyside | 40.740176 | -73.926916 | 9.2 |
| 6 | Staten Island | Sunnyside | 40.612760 | -74.097126 | 9.2 |
| 7 | Queens | Sunnyside | 40.740176 | -73.926916 | 9.2 |
| 8 | Staten Island | Sunnyside | 40.612760 | -74.097126 | 9.2 |
| 9 | Queens | Sunnyside | 40.740176 | -73.926916 | 9.2 |
| 10 | Staten Island | Sunnyside | 40.612760 | -74.097126 | 9.2 |
| 11 | Queens | Sunnyside | 40.740176 | -73.926916 | 9.2 |
| 12 | Staten Island | Sunnyside | 40.612760 | -74.097126 | 9.2 |
| 13 | Manhattan | Tribeca | 40.721522 | -74.010683 | 9.2 |

Now we will show this data on a map

In [477]:

```
# create map and display it
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)
```

In [478]:

```
# instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in ny_neighborhood_stats[['Latitude', 'Longitude']].values:
    incidents.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=10, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )
```

Lets add a new field to dataframe for labeling purpose

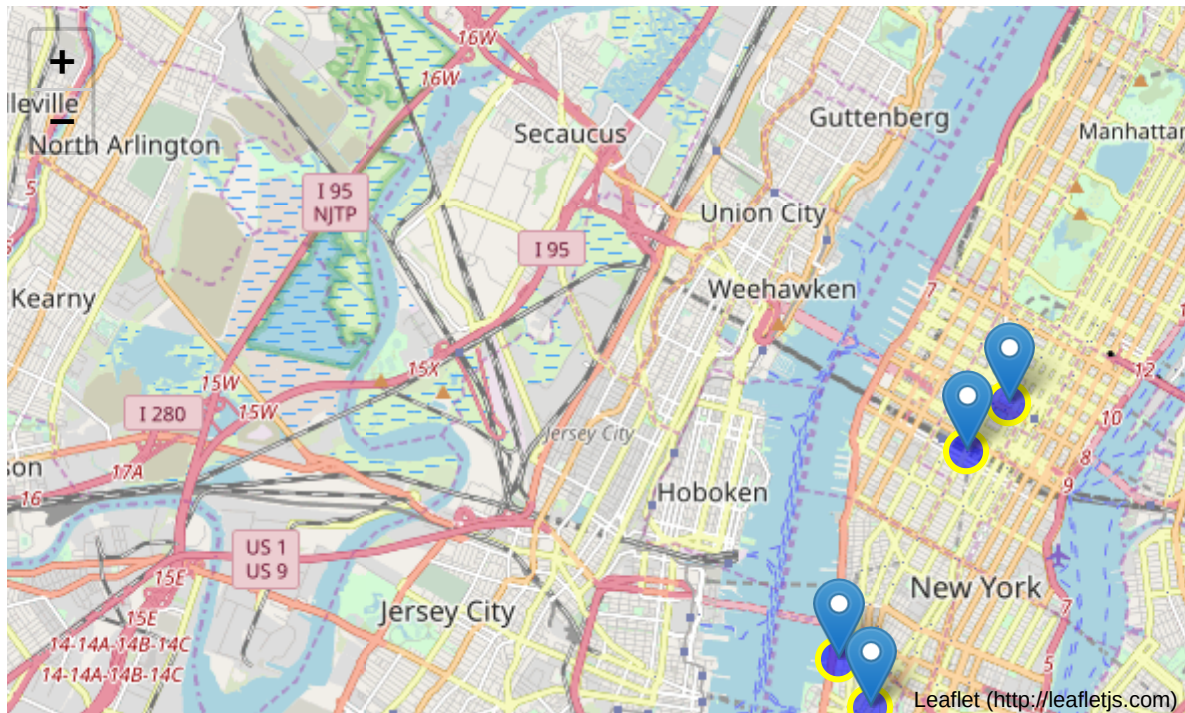
In [479]:

```
ny_neighborhood_stats['Label']=ny_neighborhood_stats['Neighborhood']+', '+ny_neighb
```

In [480]:

```
# add pop-up text to each marker on the map
for lat, lng, label in ny_neighborhood_stats[['Latitude', 'Longitude', 'Label']].valu
    folium.Marker([lat, lng], popup=label).add_to(ny_map)
# add incidents to map
ny_map.add_child(incidents)
```

Out[480]:



Now that we have visualized the Neighborhoods.
Lets Visualize Boroughs based on average Rating

In [482]:

```
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)
ny_geo = r'Borough Boundaries.geojson'

ny_map.choropleth(
    geo_data=ny_geo,
    data=ny_borough_stats,
    columns=['Borough', 'Average Rating'],
    key_on='feature.properties.boro_name',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Average Rating'
)

# display map
# as this is huge map data , we will save it to a file
ny_map.save('borough_rating.html')
```

The saved image can be downloaded at : https://github.com/kunal-chhabra/Coursera_Capstone/blob/master/borough_rating.html (https://github.com/kunal-chhabra/Coursera_Capstone/blob/master/borough_rating.html)

Conclusion

- Astoria(Queens), Blissville(Queens), Civic Center(Manhattan) are some of the best neighborhoods for indian cuisine.
- Manhattan have potential Indian Resturant Market/
- Staten Island ranks last in average rating of Indian Resturants.
- Manhattan is the best place to stay if you prefer Indian Cuisine.

Limitations

- The ranking is purely on basis of rating of resturants
- The accuracy of data depends purely depends on the data provided by FourSquare

In []: