Coursera Capstone Project — The Battle of Neighbourhoods

As part of the courser capstone final project , I have chosen to explore Indian restaurants in the city of New York.

Introduction

New York City's demographics show that it is a large and ethnically diverse metropolis. New York City comprises 5 boroughs sitting where the Hudson River meets the Atlantic Ocean. At its core is Manhattan, a densely populated borough that's among the world's major commercial, financial and cultural centers. Its iconic sites include skyscrapers such as the Empire State Building and sprawling Central Park. Broadway theater is staged in neon-lit Times Square.

New York City is the largest city in the United States with a long history of international immigration. It has been a major point of entry for immigrants; the term "melting pot" was coined to describe densely populated immigrant neighbourhoods 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 with English being the most widely spoken language.

With its diverse culture , comes diverse food culture. There are many restaurants in New York City, each belonging to different categories like African, Chinese , Indian , French etc. This project will list and visualize all major parts of New York City that has topnotch 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 (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 : Fousquare 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
  - Datasource:https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm (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 Resturants.
- Find rating , tips and like count for each Indian Resturants using FourSquare API.
- Using rating for each resturant , 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 Resturant Market ?
- Which of the areas lack indian Resturants

- 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 [7]:

```
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
! pip install  geocoder
! pip install folium
import geocoder
import os
import folium # map rendering library
from geopy.geocoders import Nominatim # convert an address into latitude and longitude v
alues
# 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.')
```

In [8]:

```
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 intract 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 [9]:

```
def get_venues(lat,lng):

    #set variables
    radius=1000
    LIMIT=100
    CLIENT_ID = '1EN1UFPHIZBZ1LWLCUDOLL5KW5ZYU0WBYGAARMIZCS320JF3' # your Foursquare ID
    CLIENT_SECRET = 'E1EAO23H0DI3FHYRV4L0OC1HGXLJLI5J0PRU2ZOT2SOUHOJT' # 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={}&v={}&ll={},{}&radius={}&limit={}'.format(
            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 [10]:
```
def get_venue_details(venue_id):

    CLIENT_ID = 'R01LINGO2WC45KLRLKT3ZHU2QENAO2IPRK2N2ELOHRNK4P3K' # your Foursquare ID
    CLIENT_SECRET = '4JT1TWRMXMPLX5IOKNBAFU3L3ARXK4D5JJDPFK1CLRZM2ZVW' # 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={}'.
format(
            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 funtion to get the new york city data such as Boroughs, Neighborhoods along with their latitude and longitude.

In [11]:
```
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_index=True)

    return new_york_data
```

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

In [12]:
```
# get new york data
new_york_data=get_new_york_data()
```

In [13]:
```
new_york_data.head()
```

Out[13]:

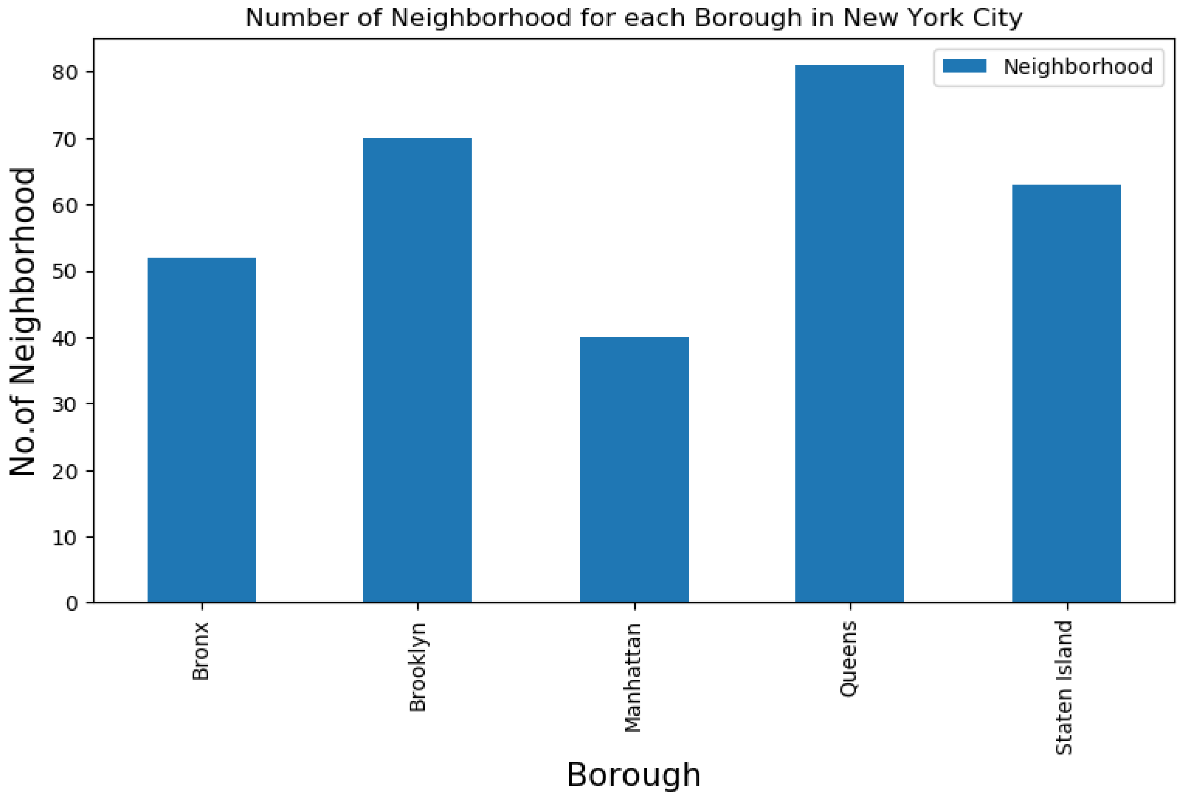
	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

In [14]:
```
new_york_data.shape
```

Out[14]:
```
(306, 4)
```

So there are total of 306 different Neighborhoods in New York

In [15]:
```
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 [16]:

```
# 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 Resturants in '+Neighborhood+', '
+Borough+':'+str(len(indian_resturants)))
    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
```

```
( 256 / 306 ) Indian Resturants in Emerson Hill, Staten Island:0
( 257 / 306 ) Indian Resturants in Randall Manor, Staten Island:0    (
258 / 306 ) Indian Resturants in Howland Hook, Staten Island:0
( 259 / 306 ) Indian Resturants in Elm Park, Staten Island:0
( 260 / 306 ) Indian Resturants in Remsen Village, Brooklyn:0
( 261 / 306 ) Indian Resturants in New Lots, Brooklyn:0
( 262 / 306 ) Indian Resturants in Paerdegat Basin, Brooklyn:0
( 263 / 306 ) Indian Resturants in Mill Basin, Brooklyn:0
( 264 / 306 ) Indian Resturants in Jamaica Hills, Queens:4
( 265 / 306 ) Indian Resturants in Utopia, Queens:0
( 266 / 306 ) Indian Resturants in Pomonok, Queens:0
( 267 / 306 ) Indian Resturants in Astoria Heights, Queens:0
( 268 / 306 ) Indian Resturants in Claremont Village, Bronx:0
( 269 / 306 ) Indian Resturants in Concourse Village, Bronx:1
( 270 / 306 ) Indian Resturants in Mount Eden, Bronx:0
( 271 / 306 ) Indian Resturants in Mount Hope, Bronx:0
( 272 / 306 ) Indian Resturants in Sutton Place, Manhattan:3
( 273 / 306 ) Indian Resturants in Hunters Point, Queens:0
( 274 / 306 ) Indian Resturants in Turtle Bay, Manhattan:2
( 275 / 306 ) Indian Resturants in Tudor City, Manhattan:2
( 276 / 306 ) Indian Resturants in Stuyvesant Town, Manhattan:0
( 277 / 306 ) Indian Resturants in Flatiron, Manhattan:0
( 278 / 306 ) Indian Resturants in Sunnyside Gardens, Queens:1
( 279 / 306 ) Indian Resturants in Blissville, Queens:1
( 280 / 306 ) Indian Resturants in Fulton Ferry, Brooklyn:0
( 281 / 306 ) Indian Resturants in Vinegar Hill, Brooklyn:0
```

In [24]:
```
indian_rest_ny.head()
```

Out[24]:

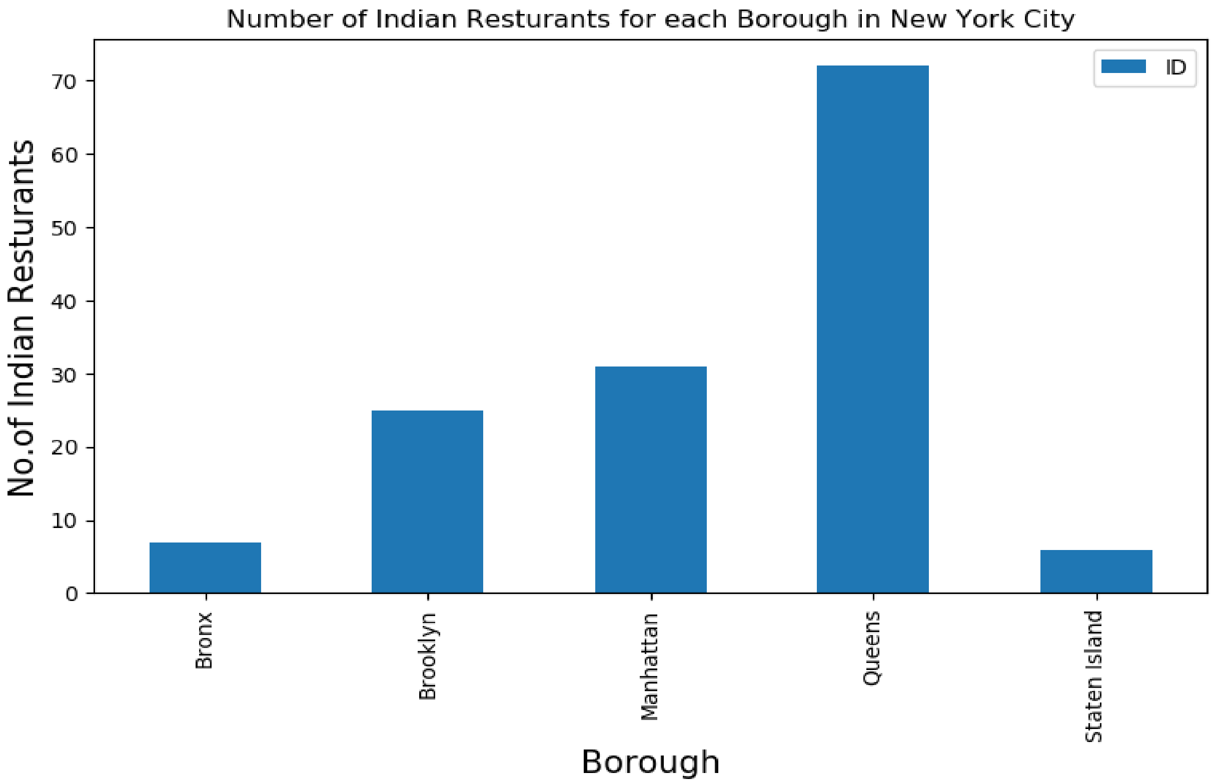
	Borough	Neighborhood	ID	Name
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill
2	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine
3	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird
4	Bronx	Unionport	4c194631838020a13e78e561	Melanies Roti Bar And Grill

In [25]:
```
indian_rest_ny.shape
```

Out[25]:
```
(141, 4)
```

We got 141 Indian Resturants across New York City

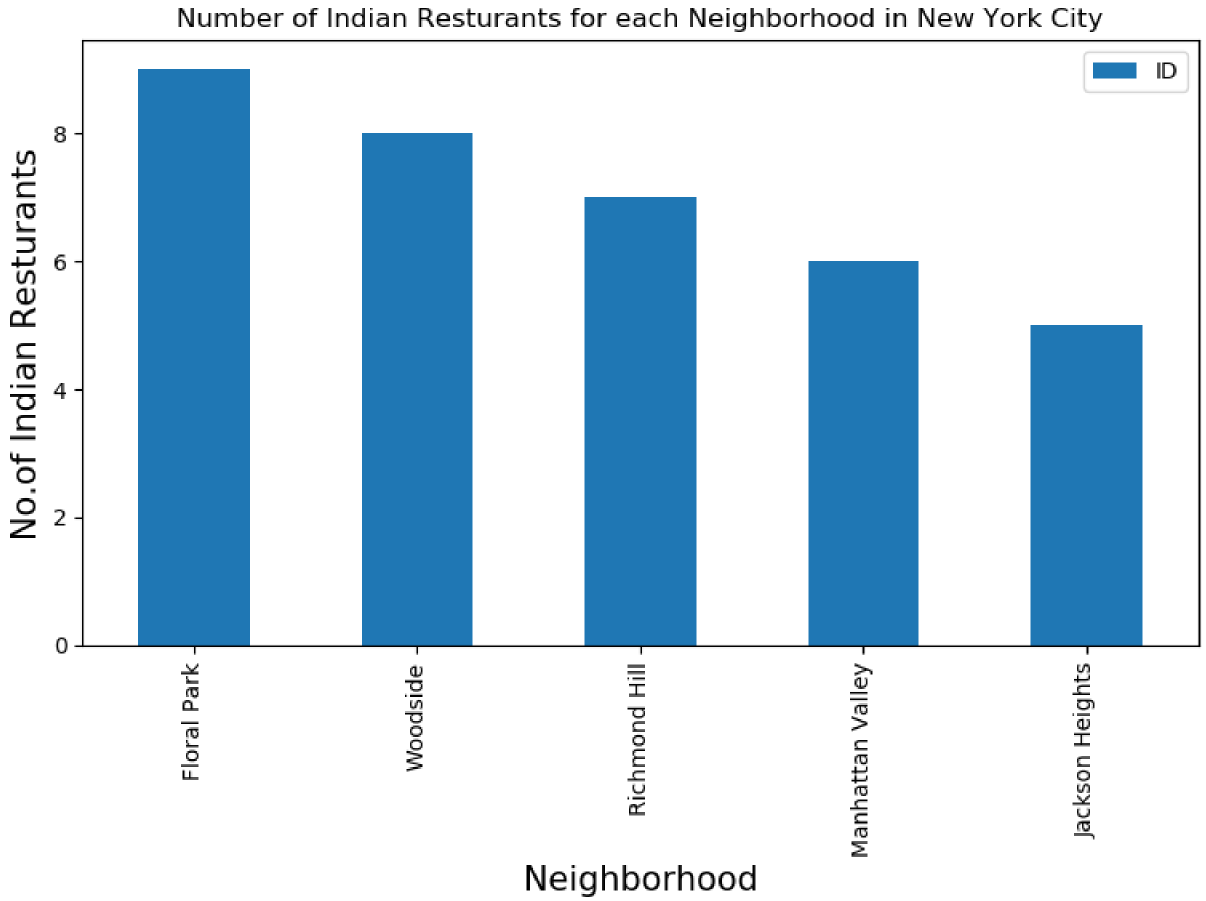
In [26]:
```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Indian Resturants for each Borough in New York  City')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('No.of Indian Resturants', fontsize=15)
#giving a bar plot
indian_rest_ny.groupby('Borough')['ID'].count().plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```



We see that Queens has the largest number of indian resturants

In [28]:

```
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 [29]:

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

Out[29]:

	Borough	Neighborhood	ID	Name
96	Queens	Floral Park	527ffc0811d2d329d5e49abd	Jackson Diner
97	Queens	Floral Park	4e4e3e22bd4101d0d7a5c2d1	Kerala Kitchen
98	Queens	Floral Park	4b647b56f964a520c4b62ae3	Usha Foods & Usha Sweets
99	Queens	Floral Park	4b787c49f964a5209cd12ee3	Santoor Indian Restaurant
100	Queens	Floral Park	4c0c01e0bbc676b00d6b4cd5	Mumbai Xpress
101	Queens	Floral Park	4c76ff35a5676dcb72671721	Flavor Of India
102	Queens	Floral Park	4df0f39dd4c04d0392c853ea	Sagar Chinese
103	Queens	Floral Park	55d68c1b498ecf05fa196fe1	Namaste Restaurant and Cafe
104	Queens	Floral Park	4e6bfe1c7d8b2c711b17bbe5	Surya sweets and snacks

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 [20]:

```
# 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  4c194631838020a13e78e561  Melanies Roti Bar And Grill      3     6.3     2
( 2 / 141 ) processed
                         ID                  Name  Likes  Rating  Tips
0  4c04544df423a593ac83d116  Cumin Indian Cuisine     13     5.9     9
( 3 / 141 ) processed
                         ID         Name  Likes  Rating  Tips
0  551b7f75498e86c00a0ed2e1  Hungry Bird      8     6.7     3
( 4 / 141 ) processed
                         ID                             Name  Likes  Rating  Tips
0  4c194631838020a13e78e561  Melanies Roti Bar And Grill      3     6.3     2
( 5 / 141 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 55dfa36a498e164ef19bef7b
( 6 / 141 ) processed
                         ID       Name  Likes  Rating  Tips
0  4b5a4dc8f964a520a2bb28e3  Taj Mahal     38     8.3   26
( 7 / 141 ) processed
                         ID          Name  Likes  Rating  Tips
0  4af0d31bf964a5207ddf21e3  Pak Nasheman      9     7.5     4
( 8 / 141 ) processed
                         ID          Name  Likes  Rating  Tips
0  52213c4211d295d4c57a607c  Ashoka Grill      8     7.2    14
( 9 / 141 ) processed
```

In [21]:
```
indian_rest_stats_ny.head()
```

Out[21]:

	Borough	Neighborhood	ID	Name	Likes	Rating	Tips
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot	5	8.0	10
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.3	2
2	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine	13	5.9	9
3	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird	8	6.7	3
4	Bronx	Unionport	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.3	2

In [22]:
```
indian_rest_stats_ny.shape
```

Out[22]:
```
(141, 7)
```

In [23]:
```
indian_rest_ny.shape
```

Out[23]:
```
(141, 4)
```

Now that we have 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 [24]:
```
indian_rest_stats_ny.to_csv('indian_rest_stats_ny.csv', index=False)
```

Verify that the data was saved to csv file

In [25]:
```
indian_rest_stats_ny_csv=pd.read_csv('indian_rest_stats_ny.csv')
```

In [26]:
```
indian_rest_stats_ny_csv.shape
```

Out[26]:
```
(141, 7)
```

In [39]:
```
indian_rest_stats_ny_csv.head()
```

Out[39]:

	Borough	Neighborhood	ID	Name	Likes	Rating	Tips
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot	5	8.0	10
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.3	2
2	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine	13	5.9	9
3	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird	8	6.7	3
4	Bronx	Unionport	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.3	2

In [27]:
```
indian_rest_stats_ny.info()
```

```
<class  'pandas.core.frame.DataFrame'>
RangeIndex: 141 entries, 0 to 140
Data columns (total 7 columns):
Borough          141  non-null  object
Neighborhood     141  non-null  object
ID               141  non-null  object
Name             141  non-null  object
Likes            141  non-null  object
Rating           141 non-null float64
Tips             141 non-null object
dtypes: float64(1),  object(6)
memory usage: 7.8+ KB
```

In [28]:
```
indian_rest_stats_ny['Likes']=indian_rest_stats_ny['Likes'].astype('float64')
```

In [29]:
```
indian_rest_stats_ny['Tips']=indian_rest_stats_ny['Tips'].astype('float64')
```

In [30]:
```
indian_rest_stats_ny.info()
```

```
<class  'pandas.core.frame.DataFrame'>
RangeIndex: 141 entries, 0 to 140
Data columns (total 7 columns):
Borough          141  non-null  object
Neighborhood     141  non-null  object
ID               141  non-null  object
Name             141  non-null  object
Likes            141  non-null  float64
Rating           141  non-null float64
Tips             141 non-null float64
dtypes: float64(3),  object(4)
memory usage: 7.8+ KB
```

In [31]:
```
# Resturant with maximum Likes
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Likes'].idxmax()]
```

Out[31]:
```
Borough                              Manhattan
Neighborhood                           Tribeca
ID                    4bbb9dbded7776b0e1ad3e51
Name                         Tamarind  TriBeCa
Likes                                      590
Rating                                     9.1
Tips                                       148
Name: 37, dtype: object
```

In [32]:
```
# Resturant with maximum Rating
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Rating'].idxmax()]
```

Out[32]:
```
Borough                              Manhattan
Neighborhood                           Tribeca
ID                    4bbb9dbded7776b0e1ad3e51
Name                         Tamarind  TriBeCa
Likes                                      590
Rating                                     9.1
Tips                                       148
Name: 37, dtype: object
```

In [33]:

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

Out[33]:

```
Borough                          Manhattan
 Neighborhood                     Gramercy
 ID                    4a70a75bf964a52016d81fe3
 Name                  Bhatti   Indian   Grill
 Likes                                   422
 Rating                                  8.7
 Tips                                    161
 Name: 45, dtype: object
```

Now lets visualize neighborhood with maximum average rating of resturants

In [34]:

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

In [35]:

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

Out[35]:

	Neighborhood	Average Rating
0	Astoria	9.1
64	Tribeca	9.1
60	Sunnyside	9.1
4	Blissville	9.1
11	Civic Center	9.1
40	Murray Hill	8.8
12	Clinton Hill	8.7
26	Gramercy	8.7
22	Fort Greene	8.7
59	Steinway	8.6

In [36]:

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

In [37]:

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

Out[37]:

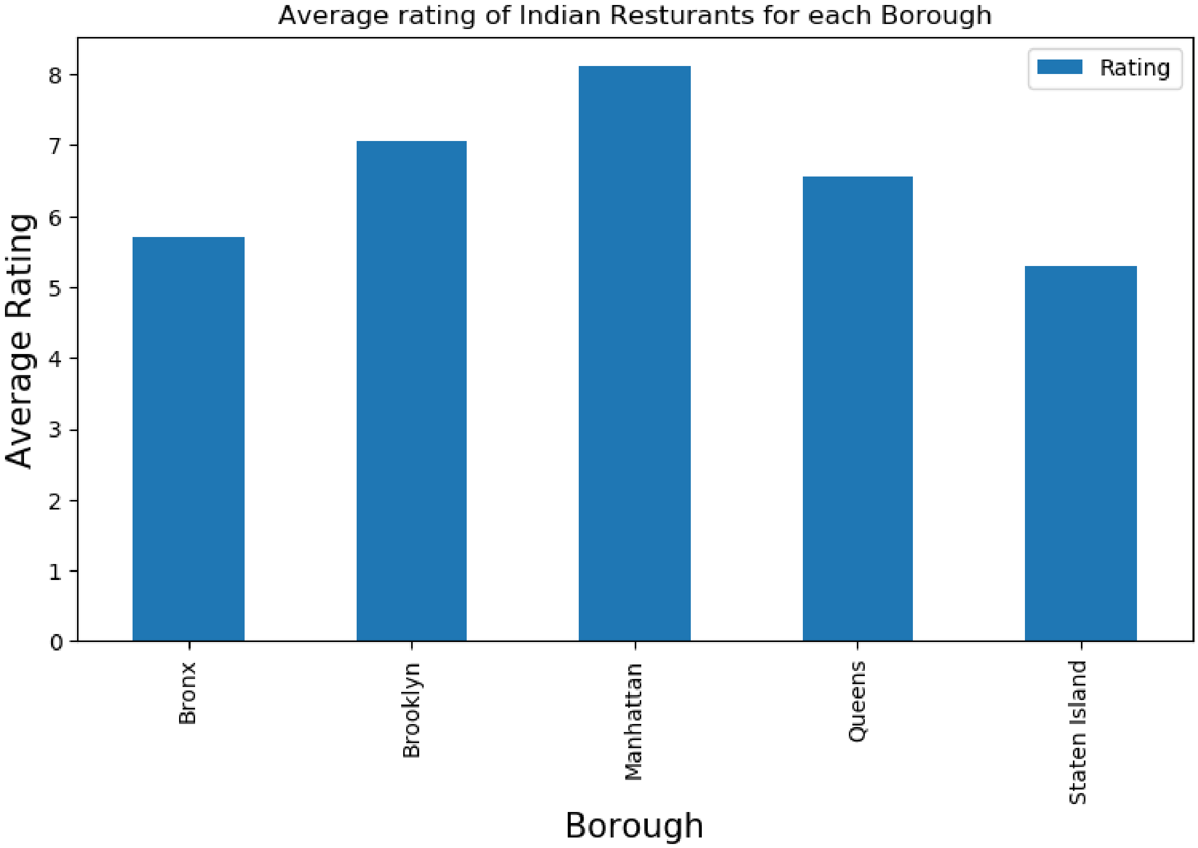
	Borough	Average Rating
2	Manhattan	8.116129
1	Brooklyn	7.048000
3	Queens	6.556944
0	Bronx	5.700000
4	Staten Island	5.300000

Above are the average rating of Indian Resturants for each Borough

Now Lets visualize it

In [38]:

```
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 8.9 to visualize on map

In [39]:
```
ny_neighborhood_stats=ny_neighborhood_stats[ny_neighborhood_stats['Average Rating']>=8.9
]
```

In [40]:
```
ny_neighborhood_stats
```

Out[40]:

	Neighborhood	Average Rating
0	Astoria	9.1
4	Blissville	9.1
11	Civic Center	9.1
60	Sunnyside	9.1
64	Tribeca	9.1

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

In [41]:
```
ny_neighborhood_stats=pd.merge(ny_neighborhood_stats,new_york_data, on='Neighborhood')
```

In [42]:
```
ny_neighborhood_stats=ny_neighborhood_stats[['Borough','Neighborhood','Latitude','Longit
ude','Average Rating']]
```

In [43]:
```
ny_neighborhood_stats
```

Out[43]:

	Borough	Neighborhood	Latitude	Longitude	Average Rating
0	Queens	Astoria	40.768509	-73.915654	9.1
1	Queens	Blissville	40.737251	-73.932442	9.1
2	Manhattan	Civic Center	40.715229	-74.005415	9.1
3	Queens	Sunnyside	40.740176	-73.926916	9.1
4	Staten Island	Sunnyside	40.612760	-74.097126	9.1
5	Manhattan	Tribeca	40.721522	-74.010683	9.1

In [44]:
```
# create map and display it
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)
```

In [45]:
```
# 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
        )
    )
```

Add a new field to dataframe for labeling purpose

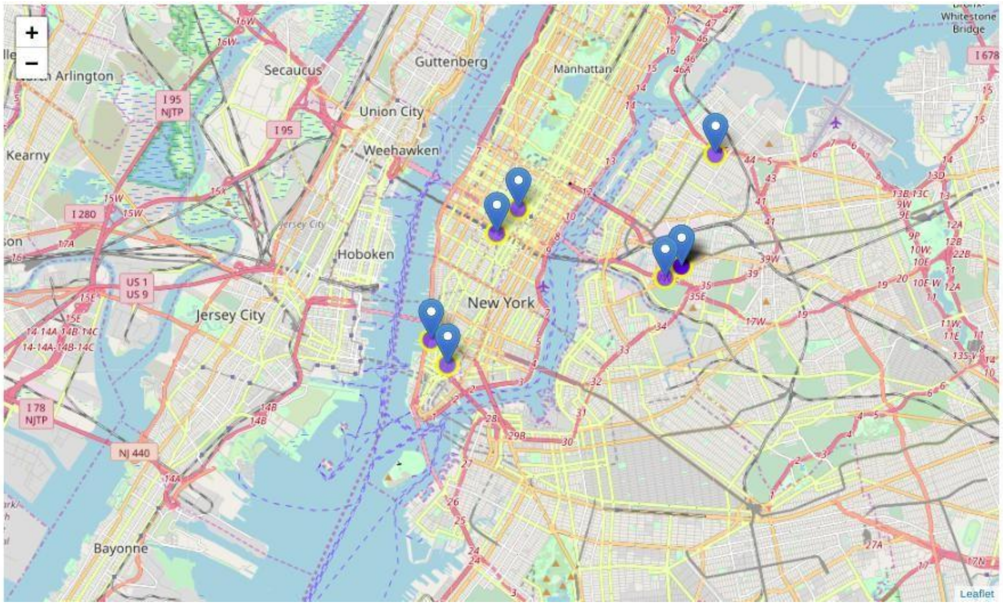
In [46]:

```
ny_neighborhood_stats['Label']=ny_neighborhood_stats['Neighborhood']+', '+ny_neighborhood_stats['Borough']+'('+ny_neighborhood_stats['Average Rating'].map(str)+')'
```

In [47]:

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

Out[47]:



Best neighbourhoods in New York that has highest average rating for Indian Restaurants.

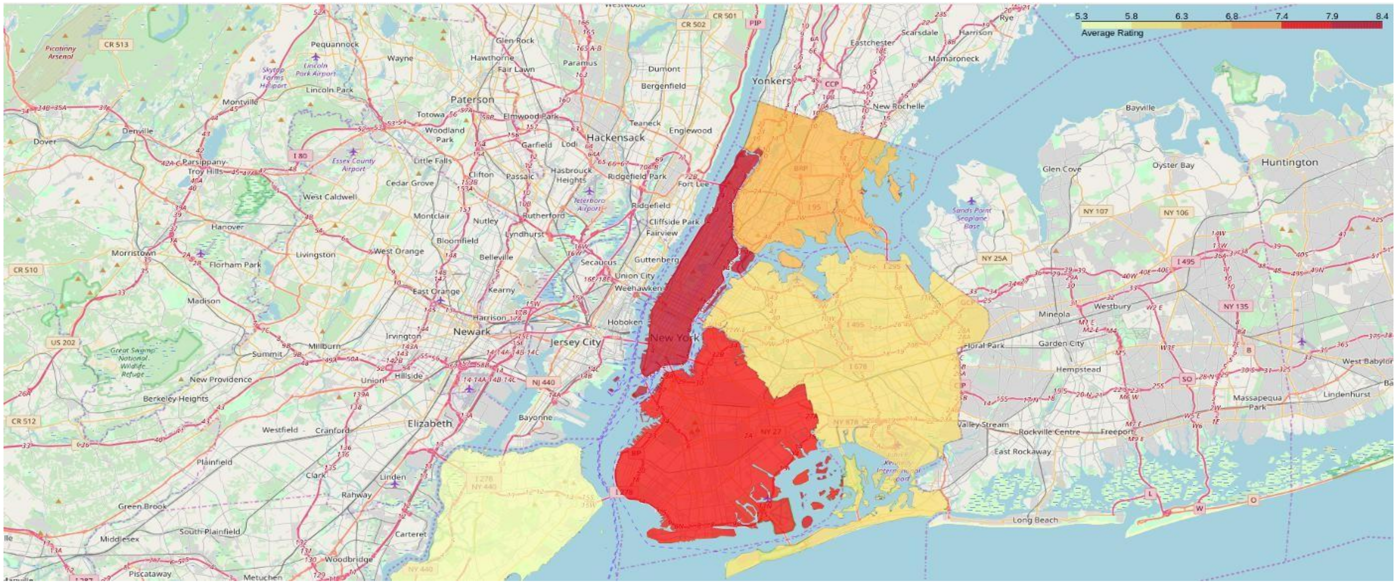
Visualizing the Boroughs based on average Rating

In [48]:

```
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)
ny_geo = r'https://data.cityofnewyork.us/api/geospatial/tqmj-j8zm?method=export&format=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
display(ny_map)
```



Borough Choropleth map based on average rating of Indian Restaurants.

Conclusion

- Manhattan is the best place to stay if you prefer Indian Cuisine.
- 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.

Limitations

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

```
In [ ]:
```