**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 resturants.

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)](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 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 longit # 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 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 [208]:

**def** get\_venues(lat,lng):

*#set variables* radius**=**1000 LIMIT**=**100

CLIENT\_ID **= ‘**X2WDJKQQM5CE5CTLAOC0NWHBZ30JTT5AECHKGEWQKQXPDHMC’CLIENT\_SECRET **=** 'VF4B1PY4KQYX3AIELJFJAARSTF4I11Z4SM0PZK5Y0YJMM4IG'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\_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.

**def** get\_venue\_details(venue\_id):

CLIENT\_ID **= ‘**X2WDJKQQM5CE5CTLAOC0NWHBZ30JTT5AECHKGEWQKQXPDHMC’CLIENT\_SECRET **=** 'VF4B1PY4KQYX3AIELJFJAARSTF4I11Z4SM0PZK5Y0YJMM4IG'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\_ti

**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.

**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*

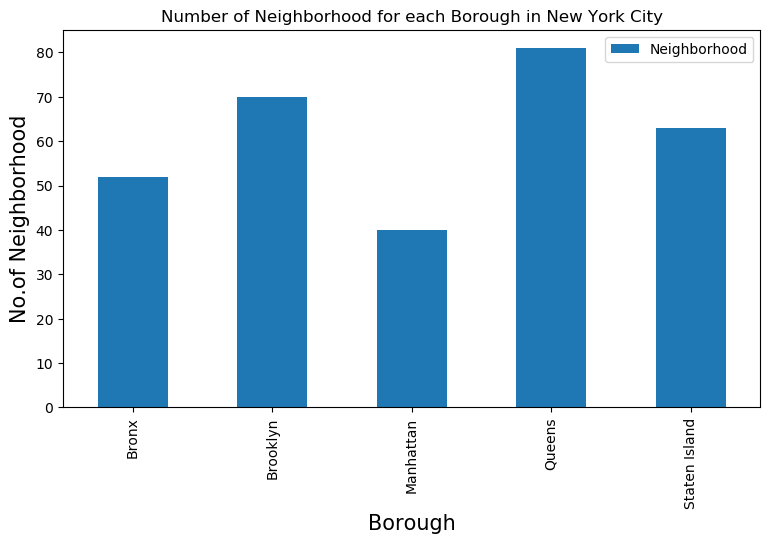
plt.xlabel('Borough', fontsize **=** 15)

*#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 resturants for each Neighborhood

*# 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 **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

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

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*

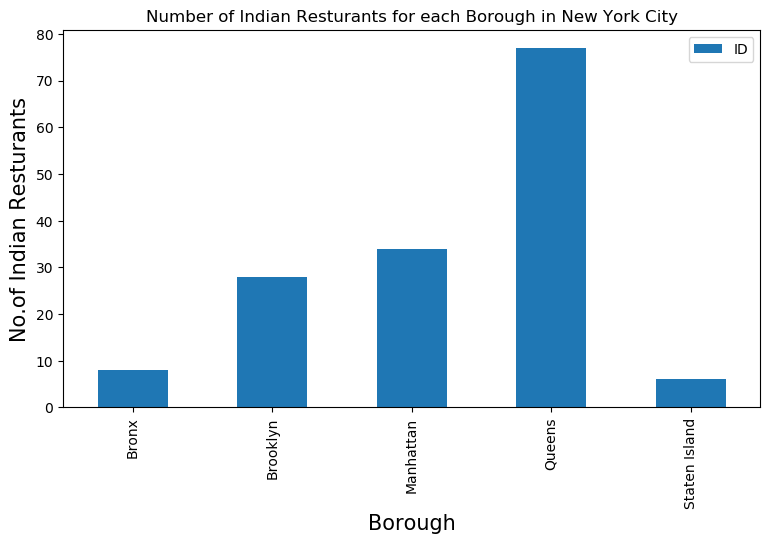
plt.xlabel('Borough', fontsize **=** 15)

*#On y-axis*

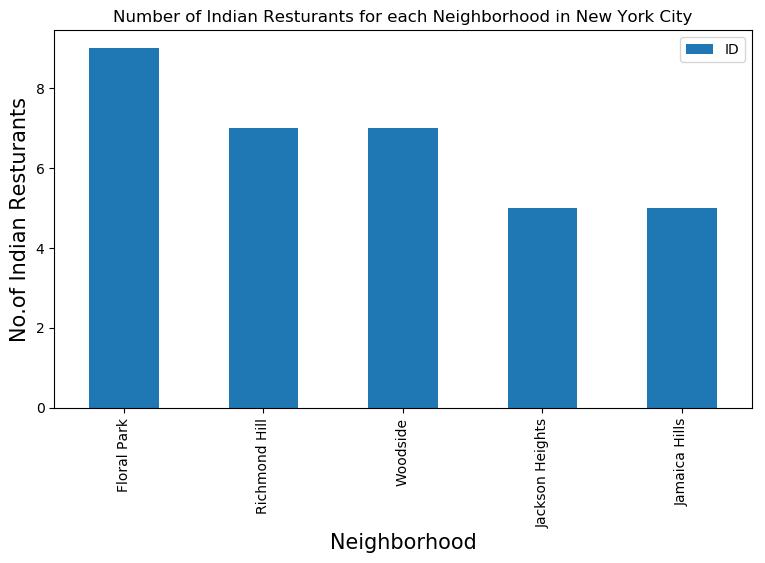
plt.ylabel('No.of Indian Resturants', fontsize**=**15)

*#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



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()

indian\_rest\_ny[indian\_rest\_ny['Neighborhood']**==**'Floral Park']

Out[238]:

**Borough Neighborhood ID Name**

1. Queens Floral Park 527ffc0811d2d329d5e49abd Jackson Diner
2. Queens Floral Park 4b647b56f964a520c4b62ae3 Usha Foods & Usha Sweets
3. Queens Floral Park 4b787c49f964a5209cd12ee3 Santoor Indian Restaurant
4. Queens Floral Park 4e4e3e22bd4101d0d7a5c2d1 Kerala Kitchen
5. Queens Floral Park 4c0c01e0bbc676b00d6b4cd5 Mumbai Xpress
6. Queens Floral Park 4c76ff35a5676dcb72671721 Flavor Of India
7. Queens Floral Park 4df0f39dd4c04d0392c853ea Sagar Chinese
8. Queens Floral Park 571af96a498e9e392d8d3786 Namaste Authenic Indian Cuisine
9. 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.

*# 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

indian\_rest\_stats\_ny.head()

Out[261]:

**Name Likes Rating Tips**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Borough** | **Neighborhood** | **ID** |
| **0** | Bronx | Riverdale | 4c04544df423a593ac83d116 |
| **1** | Bronx | Kingsbridge | 4c04544df423a593ac83d116 |
| **2** | Bronx | Woodlawn | 4c0448d9310fc9b6bf1dc761 |
| **3** | Bronx | Parkchester | 4c194631838020a13e78e561 |
| **4** | Bronx | Spuyten Duyvil | 4c04544df423a593ac83d116 |
| In | [265]: |  |  |

Cumin Indian

Cuisine

Cumin Indian

Cuisine

13 6.6 9

13 6.6 9

Curry Spot 4 7.7 10

Out[265]:

indian\_rest\_stats\_ny.shape

(153, 7)

Melanies Roti Bar

And Grill

Cumin Indian

Cuisine

3 6.1 2

13 6.6 9

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)

indian\_rest\_stats\_ny\_csv.head()

Out[270]:

**Name Likes Rating Tips**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Borough** | **Neighborhood** | **ID** |
| **0** | Bronx | Riverdale | 4c04544df423a593ac83d116 |
| **1** | Bronx | Kingsbridge | 4c04544df423a593ac83d116 |
| **2** | Bronx | Woodlawn | 4c0448d9310fc9b6bf1dc761 |
| **3** | Bronx | Parkchester | 4c194631838020a13e78e561 |
| **4** | Bronx | Spuyten Duyvil | 4c04544df423a593ac83d116 |
| In | [277]: |  |  |

Cumin Indian

Cuisine

Cumin Indian

Cuisine

13 6.6 9

13 6.6 9

Curry Spot 4 7.7 10

<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

indian\_rest\_stats\_ny.info()

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

Melanies Roti Bar

And Grill

Cumin Indian

Cuisine

3 6.1 2

13 6.6 9

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')

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

*# 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**).m 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']

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

plt.figure(figsize**=**(9,5), dpi **=** 100)

*# title*

plt.title('Average rating of Indian Resturants for each Borough')

*#On x-axis*

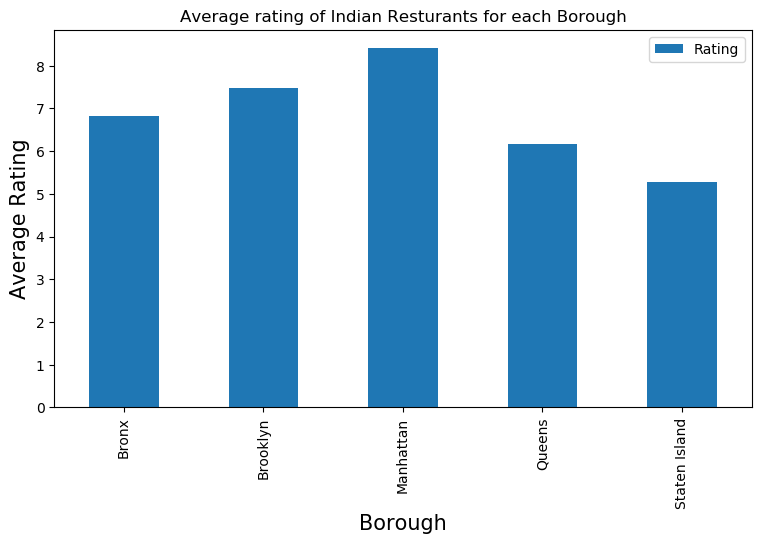
plt.xlabel('Borough', fontsize **=** 15)

*#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']

**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)

1. Queens Blissville 40.737251 -73.932442 9.2 Blissville,

Queens(9.2)

1. Manhattan Civic Center 40.715229 -74.005415 9.2 Civic Center,

Manhattan(9.2)

1. Manhattan Midtown 40.754691 -73.981669 9.0 Midtown,

Manhattan(9.0)

Midtown

Queens 40

Queens 40

Manhattan 40

Manhattan 40

**4** Manhattan Midtown South 40.748510 -73.988713 9.0

South, Manhattan(9.0)

Manhattan 40

1. Queens Sunnyside 40.740176 -73.926916 9.2 Sunnyside,

Queens(9.2)

Queens 40

1. Queens Sunnyside 40.740176 -73.926916 9.2 Sunnyside,

Queens(9.2)

|  |  |  |  |
| --- | --- | --- | --- |
| Sunnyside | 40.612760 | -74.097126 | 9.2 |
| Sunnyside | 40.612760 | -74.097126 | 9.2 |
| Tribeca | 40.721522 | -74.010683 | 9.2 |

Staten Island 40

Staten

1. Island
2. Staten Island

Sunnyside,

Staten Island(9.2)

Sunnyside,

Staten Island(9.2)

Queens 40

Staten 40 Island

1. Manhattan

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**=**'Neighborhoo

In [475]:

ny\_neighborhood\_stats**=**ny\_neighborhood\_stats[['Borough','Neighborhood','Latitude','L

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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



ny\_neighborhood\_stats['Label']**=**ny\_neighborhood\_stats['Neighborhood']**+**', '**+**ny\_neighb

In [480]:

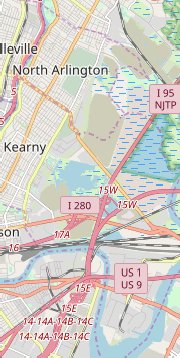
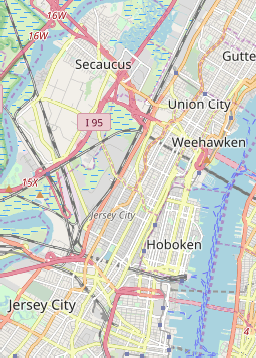
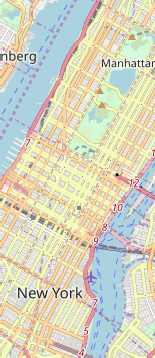
*# add pop-up text to each marker on the map*

**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]:



**+**

**−**

[Leaflet (http://leafletjs.com)](http://leafletjs.com/)

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')

# 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 [ ]: