

Table of Contents

Introduction.....	2
Data.....	2
Approach	3
Queries that can be answered using above dataset.....	3
Analysis	3
• Required Libraries.....	3
• Geo-analysis of City of New York.	4
• Neighborhood analysis for Indian restaurant	7
• folium to visualize the results on a map	15
Conclusion	15

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

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
- 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 Restaurants using FourSquare API.
- Using rating for each restaurant, we will sort that data.
- Visualize the Ranking of neighborhoods using folium library(python)

Queries that can be answered using above dataset

- 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

- Required Libraries

```
In [1]: import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import requests
import sys
from bs4 import BeautifulSoup
import geocoder
import os
import folium # map rendering library
from geopy.geocoders import Nominatim
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.

- Geo-analysis of City of New York.

```
In [2]: 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 [22]: def get_venues(lat,lng):

#set variables
radius=1000
LIMIT=100
CLIENT_ID = '#####' # changed my Foursquare ID
CLIENT_SECRET = '#####' # changed 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)
print("done")
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 [23]: def get_venue_details(venue_id):

CLIENT_ID = '#####' # i have changed the id with ##
CLIENT_SECRET = '#####' # i have changed the secret with ##
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 [5]: 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 [9]: # get new york data
new_york_data=get_new_york_data()
new_york_data.head()
```

```
Out[9]:
```

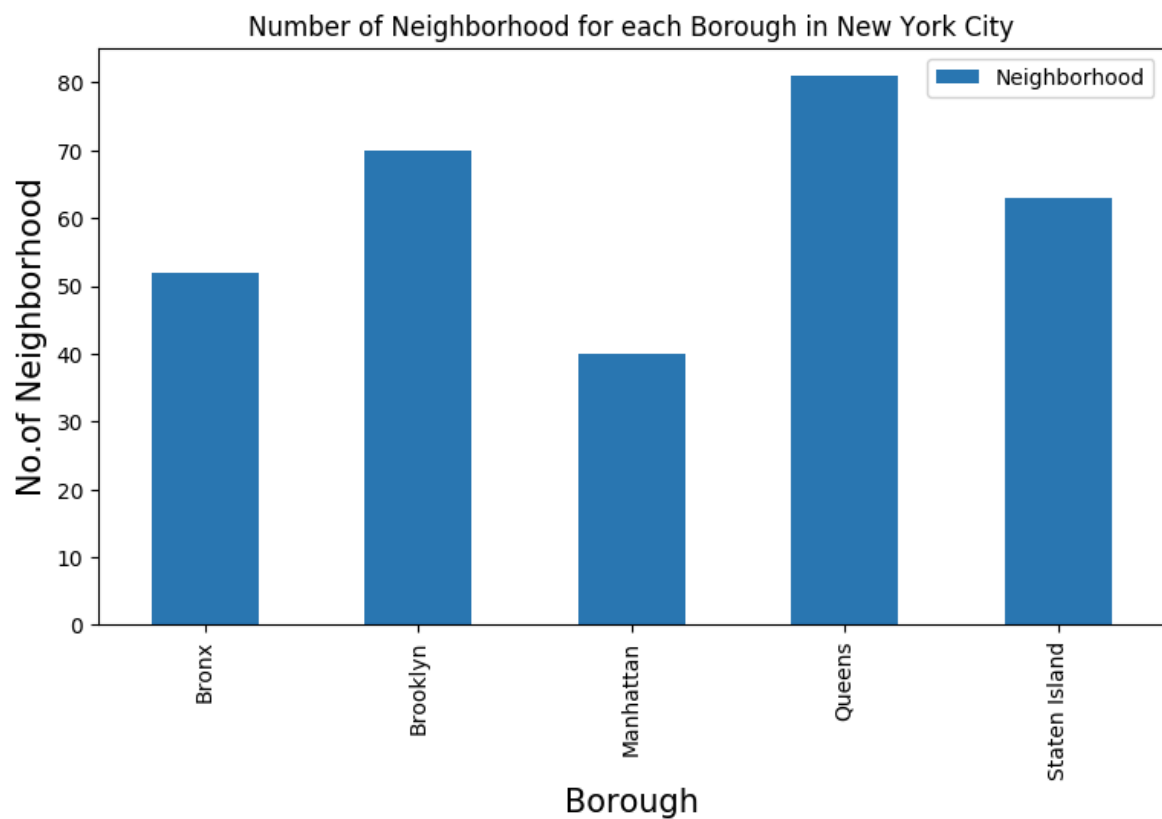
	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

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

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

So there are total of 306 different Neighborhoods in New York

```
In [10]: 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()
```



Conclusion: We see that Queens has highest number of neighbourhoods

- Neighbourhood analysis for Indian restaurant

```
In [11]: # prepare neighborhood list that contains indian restaurants
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_restaurants=venues[venues['Category']=='Indian Restaurant']
    print(' ',count,'/',len(new_york_data),' ', 'Indian Restaurants in '+Neighborhood+', '+Borough+':'+str(len(indian_restaurants)))
    for restaurant_detail in indian_restaurants.values.tolist():
        id, name , category=restaurant_detail
        indian_rest_ny = indian_rest_ny.append({'Borough': Borough,
                                                'Neighborhood': Neighborhood,
                                                'ID': id,
                                                'Name' : name
                                                }, ignore_index=True)
    count+=1

done
( 1 / 306 ) Indian Restaurants in Wakefield, Bronx:0
done
( 2 / 306 ) Indian Restaurants in Co-op City, Bronx:0
done
( 3 / 306 ) Indian Restaurants in Eastchester, Bronx:0
done
( 4 / 306 ) Indian Restaurants in Fieldston, Bronx:0
done
( 5 / 306 ) Indian Restaurants in Riverdale, Bronx:0
done
( 6 / 306 ) Indian Restaurants in Kingsbridge, Bronx:0
done
```

```
In [12]: indian_rest_ny.head()
```

```
Out[12]:
```

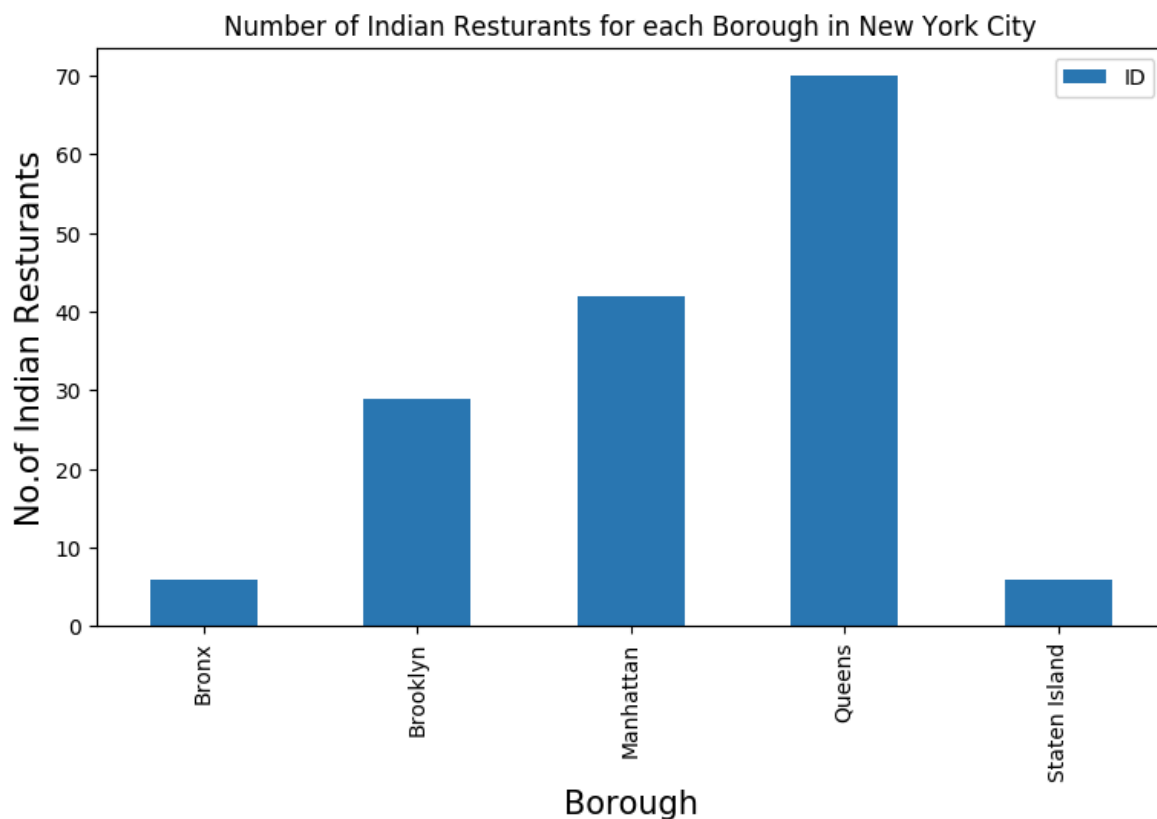
	Borough	Neighborhood	ID	Name
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill
2	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine
3	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird
4	Bronx	Unionport	4c194631838020a13e78e561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

```
In [13]: indian_rest_ny.shape
```

```
Out[13]: (153, 4)
```

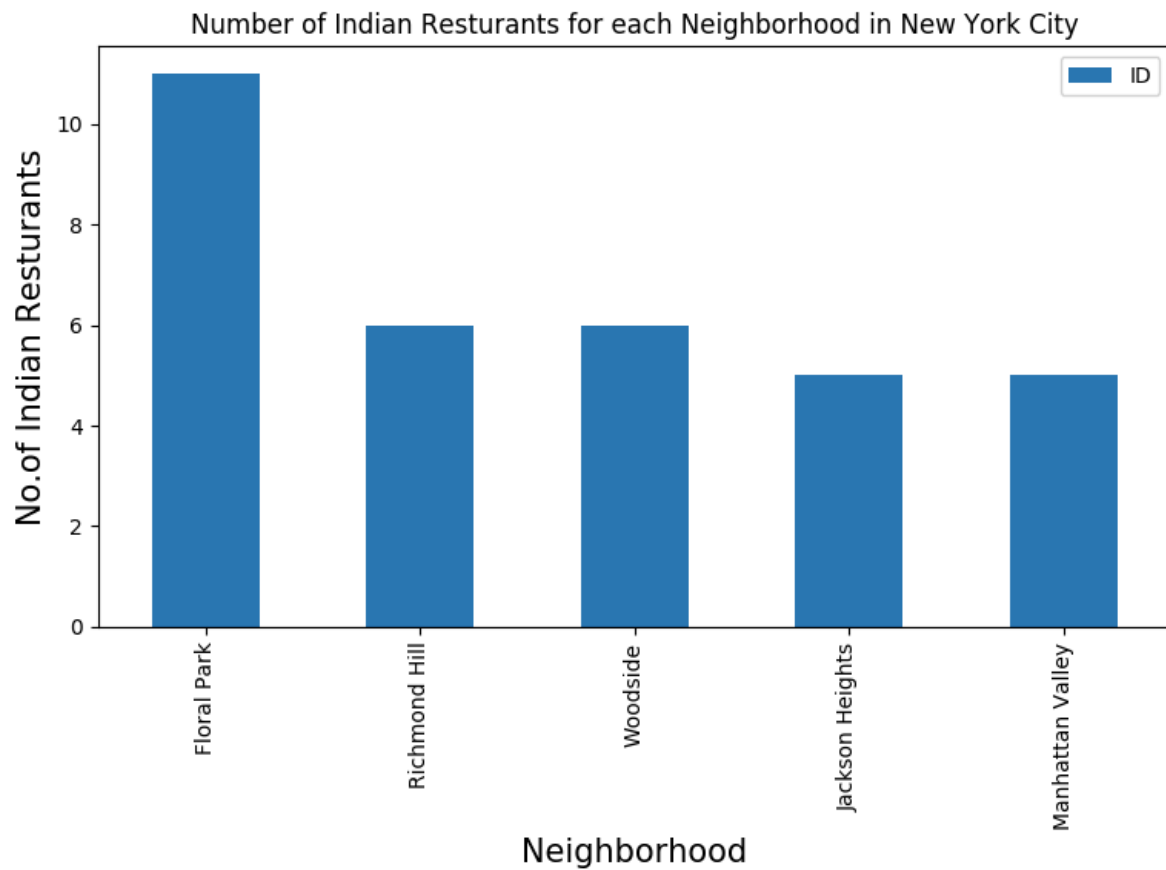
We got 153 Indian Restaurants across New York City

```
In [14]: 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()
```



Conclusion: We see that Queens has the largest number of indian resturants

```
In [15]: 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 [16]: indian_rest_ny[indian_rest_ny['Neighborhood']=='Floral Park']
```

Out[16]:

	Borough	Neighborhood	ID	Name
103	Queens	Floral Park	4e4e3e22bd4101d0d7a5c2d1	Kerala Kitchen
104	Queens	Floral Park	4b647b56f964a520c4b62ae3	Usha Foods & Usha Sweets
105	Queens	Floral Park	527ffc0811d2d329d5e49abd	Jackson Diner
106	Queens	Floral Park	4b787c49f964a5209cd12ee3	Santoor Indian Restaurant
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	55d68c1b498ecf05fa196fe1	Namaste Restaurant and Cafe
112	Queens	Floral Park	4c3e17f2ca012d7f82022fbe	Mushin's Halal Food [Gyro Cart]
113	Queens	Floral Park	4e6bfe1c7d8b2c711b17bbe5	Surya sweets and snacks

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

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

```
In [24]: # 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
```

```
In [25]: indian_rest_stats_ny.head()
```

```
Out[25]:
```

	Borough	Neighborhood	ID	Name	Likes	Rating	Tips
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot	4	8.1	11
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.0	2
2	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine	13	6.1	9
3	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird	8	6.9	3
4	Bronx	Unionport	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.0	2

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

```
Out[26]: (50, 7)
```

```
In [27]: indian_rest_ny.shape
```

```
Out[27]: (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 [28]: indian_rest_stats_ny.to_csv('indian_rest_stats_ny.csv', index=False)
```

Lets verify the data from saved csv file

```
In [29]: indian_rest_stats_ny_csv=pd.read_csv('indian_rest_stats_ny.csv')
```

```
In [30]: indian_rest_stats_ny_csv.shape
```

```
Out[30]: (50, 7)
```

```
In [31]: indian_rest_stats_ny_csv.head()
```

```
Out[31]:
```

	Borough	Neighborhood	ID	Name	Likes	Rating	Tips
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot	4	8.1	11
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.0	2
2	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine	13	6.1	9
3	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird	8	6.9	3
4	Bronx	Unionport	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	6.0	2

```
In [32]: indian_rest_stats_ny.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
Borough      50 non-null object
Neighborhood  50 non-null object
ID           50 non-null object
Name         50 non-null object
Likes        50 non-null object
Rating       50 non-null float64
Tips         50 non-null object
dtypes: float64(1), object(6)
memory usage: 2.8+ KB
```

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

```
In [33]: indian_rest_stats_ny['Likes']=indian_rest_stats_ny['Likes'].astype('float64')
```

```
In [34]: indian_rest_stats_ny['Tips']=indian_rest_stats_ny['Tips'].astype('float64')
```

```
In [35]: indian_rest_stats_ny.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
Borough      50 non-null object
Neighborhood  50 non-null object
ID           50 non-null object
Name         50 non-null object
Likes        50 non-null float64
Rating       50 non-null float64
Tips         50 non-null float64
dtypes: float64(3), object(4)
memory usage: 2.8+ KB
```

Now the data types looks correct

```
In [36]: # Resturant with maximum Likes
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Likes'].idxmax()]
```

```
Out[36]: Borough      Manhattan
Neighborhood  Midtown
ID            49d91c12f964a520015e1fe3
Name          The Kati Roll Company
Likes         836
Rating        8.8
Tips          259
Name: 41, dtype: object
```

```
In [37]: # Resturant with maximum Rating
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Rating'].idxmax()]
```

```
Out[37]: Borough          Manhattan
Neighborhood          Tribeca
ID          4bbb9dbded7776b0elad3e51
Name          Tamarind TriBeCa
Likes          586
Rating          9
Tips          145
Name: 45, dtype: object
```

```
In [38]: # Resturant with maximum Tips
indian_rest_stats_ny.iloc[indian_rest_stats_ny['Tips'].idxmax()]
```

```
Out[38]: Borough          Manhattan
Neighborhood          Midtown
ID          49d91c12f964a520015elfe3
Name          The Kati Roll Company
Likes          836
Rating          8.8
Tips          259
Name: 41, dtype: object
```

```
In [39]: ny_neighborhood_stats=indian_rest_stats_ny.groupby('Neighborhood',as_index=False).mean()[['Neighborhood','Rating']]
ny_neighborhood_stats.columns=['Neighborhood','Average Rating']
```

```
In [40]: ny_neighborhood_stats.sort_values(['Average Rating'],ascending=False).head(10)
```

```
Out[40]:
```

	Neighborhood	Average Rating
28	Tribeca	9.00
13	Greenwich Village	8.90
32	West Village	8.85
18	Midtown	8.80
3	Chelsea	8.80
19	Murray Hill	8.70
10	Fort Greene	8.60
20	North Side	8.60
26	South Side	8.60
25	Roosevelt Island	8.40

```
In [41]: ny_borough_stats=indian_rest_stats_ny.groupby('Borough',as_index=False).mean()[['Borough','Rating']]
ny_borough_stats.columns=['Borough','Average Rating']
```

```
In [42]: ny_borough_stats.sort_values(['Average Rating'],ascending=False).head()
```

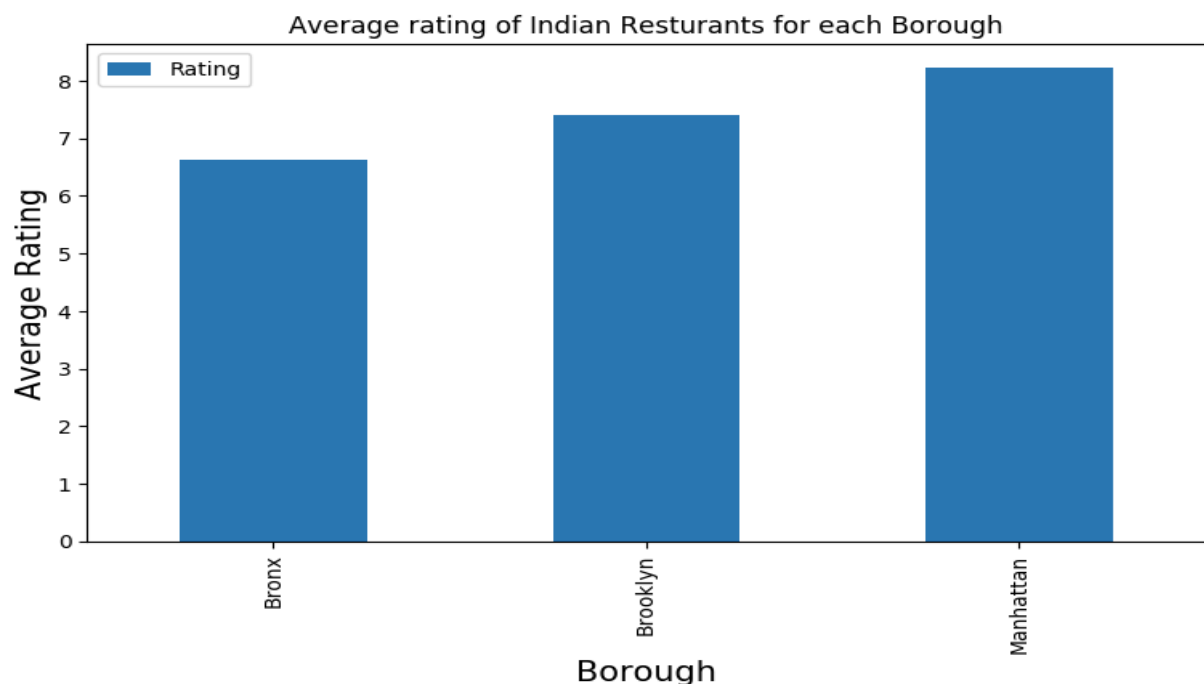
```
Out[42]:
```

	Borough	Average Rating
2	Manhattan	8.227273
1	Brooklyn	7.408696
0	Bronx	6.620000

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

Lets visualize it

```
In [43]: 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()
```



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

```
In [44]: ny_neighborhood_stats=ny_neighborhood_stats[ny_neighborhood_stats['Average Rating']>=9.0]
```

```
In [45]: ny_neighborhood_stats
```

```
Out[45]:
```

	Neighborhood	Average Rating
28	Tribeca	9.0

```
In [46]: ny_neighborhood_stats=pd.merge(ny_neighborhood_stats,new_york_data, on='Neighborhood')
```

```
In [47]: ny_neighborhood_stats=ny_neighborhood_stats[['Borough', 'Neighborhood', 'Latitude', 'Longitude', 'Average Rating']]
```

```
In [48]: ny_neighborhood_stats
```

```
Out[48]:
```

	Borough	Neighborhood	Latitude	Longitude	Average Rating
0	Manhattan	Tribeca	40.721522	-74.010683	9.0

Now we will show this data on a map

```
In [49]: # create map and display it
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)
```

```
In [50]: # 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 [51]: ny_neighborhood_stats['Label']=ny_neighborhood_stats['Neighborhood']+', '+ny_neighborhood_stats['Borough']+', ('+ny_neighborhood_stats['Average Rating'].map(str)+')
```

```
In [54]: # 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[54]:
```

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

- folium to visualize the results on a map

```
In [66]: 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

Limitations

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