Analysis report for opening a New Restaurant in NY

Introduction

This is a Capstone project wherein am trying to find a s uitable place in Newyork City to open a ne restaurant. The city is filled with variety of people and it has all types of cuisines. The below link describes the various cuisines in New york and which neighborhood serves what cuisine.

https://en.wikipedia.org/wiki/Cuisine_of_New_York_City

Business Problem

Newyork city has a diverse population and serves as a Global hub for many industries. It has people from all over the world and have a different taste when comes to tasting a food. It already has a lot of restaurants which are famous and satisfied the people in Newyork. So starting a new restaurant in the area where we have established restaurants invovles lot of analysis.

Few of the critical things that we need to consider before opening up a restaurant is

NY demographics

Population

Cuisines served

Competitors in the area

Unexplored horizon

Nearby venues that favours the opening of new restaurant Nearby Markets

Target Audience

Anyone who wants to open a new restaurant in NY city

Data Section

Since we have taken New york city for our analysis. The following datasets will be used ot get the required data that will help us in completing the project.

https://cocl.us/new_york_dataset - NY neighborhood and Borough data

https://en.wikipedia.org/wiki/Cuisine_of_New_York_City - Cuisines of NY city

Coordinates from Foursquare API to explore the neighborhoods of NY city

Methodology

Data will be collected from https://cocl.us/new_york_dataset using bs4 webscraping utility.

The collected data will be cleaned and loaded into a data frame fro further analysis.

Foursquare API will be used to locate the venues with restaurants.

Data will be processed and sorted based on the number of restaurants in the neigborhood

Data will be visually assessed to make any decision using Folium

Execution of codes

Importing the necessary Libraries to perform our analysis

In [1]:

In [1]:

```
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
import os
!conda install -c conda-forge folium=0.5.0 --yes
import folium
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
%matplotlib inline
import seaborn as sns
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    branca-0.3.1               |             py_0          25 KB  conda-forge
    altair-3.2.0               |           py36_0         770 KB  conda-forge
    ca-certificates-2019.9.11  |       hecc5488_0         144 KB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    openssl-1.1.1d             |       h516909a_0         2.1 MB  conda-forge
    certifi-2019.9.11          |           py36_0         147 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         3.2 MB

The following NEW packages will be INSTALLED:

    altair:          3.2.0-py36_0      conda-forge
    branca:          0.3.1-py_0        conda-forge
    folium:          0.5.0-py_0        conda-forge
    vincent:         0.4.4-py_1        conda-forge

The following packages will be UPDATED:

    certifi:         2019.9.11-py36_0             --> 2019.9.11-py36_0     conda-forge

The following packages will be DOWNGRADED:

    ca-certificates: 2019.10.16-0                 --> 2019.9.11-hecc5488_0 conda-forge
    openssl:         1.1.1d-h7b6447c_3            --> 1.1.1d-h516909a_0    conda-forge


Downloading and Extracting Packages
vincent-0.4.4        | 28 KB     | ##################################### | 100% 
branca-0.3.1         | 25 KB     | ##################################### | 100% 
altair-3.2.0         | 770 KB    | ##################################### | 100% 
ca-certificates-2019 | 144 KB    | ##################################### | 100% 
folium-0.5.0         | 45 KB     | ##################################### | 100% 
openssl-1.1.1d       | 2.1 MB    | ##################################### | 100% 
certifi-2019.9.11    | 147 KB    | ##################################### | 100% 
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - geopy
```

```
The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    geopy-1.20.0               |             py_0          57 KB  conda-forge
    geographiclib-1.50         |             py_0          34 KB  conda-forge
    ------------------------------------------------------------
                                           Total:          91 KB

The following NEW packages will be INSTALLED:

    geographiclib: 1.50-py_0   conda-forge
    geopy:         1.20.0-py_0 conda-forge


Downloading and Extracting Packages
geopy-1.20.0         | 57 KB     | ##################################### | 100%
geographiclib-1.50   | 34 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Setting the variables for Foursquare API

Defining the following functions that will be used in our exercise

Function to return Longitude and Latitude values

Function to hit Foursquar API to get the venue and other details

Function to scrap NY data

In [9]:

```
def geo_location(address):
    geolocator = Nominatim(user_agent="foursquare_agent")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    return latitude,longitude


def get_venues(lat,lng):
    radius=400
    LIMIT=100
    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)
    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


def get_venue_details(venue_id):
    url = 'https://api.foursquare.com/v2/venues/{}?&client_id={}&client_secret={}&v={}'.format(
            venue_id,
            venue_id,
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION)
    results = requests.get(url).json()
    print(results)
    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


def get_new_york_data():
    url='https://cocl.us/new_york_dataset'
    resp=requests.get(url).json()
    features=resp['features']
    column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']
    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
```

Results and Conclusion

Going by the metrics, the number of restaurants in the Borough "Bronx" is less compared to other Boroughs which implies the competition is less to start a new restaurant.The average rating in "Bronx" is also very less compared to other Boroughs, which implies the people in Bronx are not happy with the restaurants that are currently functioning. A new restaurant satisfying the needs of the people in Bronx will be a right choice going by the metrics. Also please note, the analysis and the findings are purely based on the data sets and available ratings,likes in the Foursquare API. There might be more data which may not be captured or stored in Foursquare which will change the outcome. It is always better to re execute this analysis on top of various available datasets to arrive at the final outcome.

```
In [ ]:
```