Recommendation to Open New Indian Hyderabadi Restaurant in New York

IBM-COURSERA DATA SCIENCE CAPSTONE PROJECT

Syed Javeed Pasha | IBM Data Science Capstone Project | August 23, 2020

Capstone Project - The Competing Neighborhoods

(WEEK 2/ WEEK 5)

APPLIED DATA SCIENCE CAPSTONE BY IBM/COURSERA

TABLE OF CONTENTS

I INTRODUCTION: BUSINESS PROBLEM

In this capstone project a new restaurant's suitable/profitable/optimal location will be found. Particularly, the stakeholders are interested in **Indian Hyderabdi restaurant** in **New York**, **USA**.
There are so many restaurants in New York we will choose such **a place/location where there are less restaurants**. Also, we also needs to choose such a **location where there are no Indian Hyderabadi restaurants nearby**. Furthermore, we also take care that the **prefer places/locations should be as close as possible to the city center**, along with the first two criteria are met.

We will recommend the most suitable palces/neighborhoods based above mentioned criteria using the skills we learn during this long journey of Data Science Course. So stakeholders can choose best, suitable, possible area/location as we explain pros and cons each location in clear and easy way.

II DATA/DATASET

As we defined our problem and based on this, Our deicsion is influenced by following criteria/factors:

- The number of already existing restaurants(can be of any type) in the area/neighborhood
- If any, how many number of and far to Hyderabadi resturants
- From the heart of the city, the neighborhood is how much far away

```
In [1]: !wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
        print('Data downloaded!')

Data downloaded!
```

Load and explore the data

```
In [4]: import json # library to handle JSON files

        with open('newyork_data.json') as json_data:
            newyork_data = json.load(json_data)
```

```
In [5]: newyork_data
```

To define our neighborhood/area, we will use regularly spaced grid of locations, centered aoround heart of the city.

To extract/generate the required information, below data sources will be required:

- **Foursquare API** to get venue data related to these neighborhoods, number of restaurants & their type and location in every area/neighborhood will be obtained
- To solve this problem, we will need below data:
- List of neighborhoods in New York, USA
- Latitude and Longitude of these neighborhoods
- Venue data related to Indian Hyderabadi restaurants. This will help us find neighborhoods that are more suitable to open an Indian Hyderabadi Restaurant.
- EXTRACTING THE DATA
- The scrapping of New York neighborhoods via Wikipedia
- Getting Latitude and Longitude data of these neighborhoods via Geocoder package

Neighborhood Candidates

For centroids of our candidate neighborhoods, let's create latitude and longitude coordinates. Around the heart of the New York City, we will create a grid of cells covering our area of interest which is approximately 11x11 kilometers.

Using specific, well known address and Google Maps geocoding API, let's first find the latitude & longitude of heart of the New York City.

III METHODOLOGY

First, I need to get the list of neighborhoods in New York, USA. This is possible by extracting the list of neighborhoods from url already available in one of the labs in this course. I did the web scraping by utilizing pandas HTML table scraping method as it is easier and more convenient to pull tabular data directly from a web page into the data frame. However, it is only a list of neighborhood names and postal codes. I need to get their coordinates to utilize Foursquare to pull the list of venues near these neighborhoods. To get the coordinates, I tried using Geocoder Package

but it was not working so I used the CSV file provided by IBM team to match the coordinates of New York neighborhoods.

In [57]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

After gathering these coordinates, I visualize the map of New York using Folium package to verify whether these are correct coordinates. Next, I use Foursquare API to pull the list of top 100 venues within 500 meters radius. I have created a Foursquare developer account in order to obtain account ID and API key to pull the data. From Foursquare, I am able to pull the names, categories, latitude, and longitude of the venues. With this data, I can also check how many unique categories that I can get from these venues. Then, I analyze each neighborhood by grouping the rows by neighborhood and taking the mean on the frequency of occurrence of each venue category. This is to prepare clustering to be done later. Here, I made a justification to specifically look for "Indian restaurants".

In [13]:
```
neighborhoods.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]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(neighborhoods['Borough'].unique()),
        neighborhoods.shape[0]
    )
)
```

```
The dataframe has 5 boroughs and 306 neighborhoods.
```

In [16]:
```
neighborhoods.rename(columns={'Neighbourhood': 'Neighborhood'}, inplace=True)
```

In [17]:
```
neighborhoods.groupby('Borough').count()['Neighborhood']
```

Out[17]:
```
Borough
Bronx            52
Brooklyn         70
Manhattan        40
Queens           81
Staten Island    63
Name: Neighborhood, dtype: int64
```

In [32]:
```
CLIENT_ID = '0EFN0AWTYNZUBCTPZQM1ZHTYMGIS2WJGSOFH1USNGXL2ZJVG' # your Foursquare ID
CLIENT_SECRET = 'VQFQJKYGIOK2AMFI0JAQW0C3TTIIAPVFLRF04V5TJKPFJYDO' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # Limit of number of venues returned by Foursquare API
radius = 500 # define radius

def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        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)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
```

Lastly, I performed the clustering method by using k-means clustering. K-means clustering algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster while keeping the centroids as small as possible. It is one of the simplest and popular unsupervised machine learning algorithms and it is highly suited for this project as well.

IV ANALYSIS

```
In [33]: newyork_venues.shape
Out[33]: (10087, 7)
```

```
In [34]: newyork_venues.groupby('Neighborhood').count()
```

Out[34]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Allerton	31	31	31	31	31	31
Annadale	12	12	12	12	12	12
Arden Heights	4	4	4	4	4	4
Arlington	6	6	6	6	6	6
Arrochar	24	24	24	24	24	24
Arverne	21	21	21	21	21	21
Astoria	99	99	99	99	99	99
Astoria Heights	11	11	11	11	11	11
Auburndale	20	20	20	20	20	20
Bath Beach	48	48	48	48	48	48

```
In [42]: len(to_grouped[to_grouped["Indian Restaurant"] > 0])
Out[42]: 43

In [43]: to_indian = to_grouped[["Neighborhoods","Indian Restaurant"]]

In [44]: to_indian.head(10)
```

Out[44]:

	Neighborhoods	Indian Restaurant
0	Allerton	0.000000
1	Annadale	0.000000
2	Arden Heights	0.000000
3	Arlington	0.000000
4	Arrochar	0.000000
5	Arverne	0.000000
6	Astoria	0.040404
7	Astoria Heights	0.000000
8	Auburndale	0.000000
9	Bath Beach	0.000000

I have clustered the neighborhoods in New York into 3 clusters based on their frequency of occurrence for “Indian food”. Based on the results (the concentration of clusters), I will be able to recommend the ideal location to open the restaurant.

V RESULTS AND DISCUSSION

In [45]:
```
from sklearn.cluster import KMeans
toclusters = 3

to_clustering = to_indian.drop(["Neighborhoods"], 1)

# run k-means clustering
kmeans = KMeans(n_clusters=toclusters, random_state=1)
kmeans.fit_transform(to_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:20]
```

Out[45]:
```
array([0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0],
      dtype=int32)
```

In [46]:
```
to_merged = to_indian.copy()

# add clustering labels
to_merged["Cluster Labels"] = kmeans.labels_
```

In [49]:
```
to_merged.rename(columns={"Neighborhoods": "Neighborhood"}, inplace=True)
to_merged.head(20)
```

Out[49]:

	Neighborhood	Indian Restaurant	Cluster Labels
0	Allerton	0.000000	0
1	Annadale	0.000000	0
2	Arden Heights	0.000000	0
3	Arlington	0.000000	0

In [52]:
```
map_clusters = folium.Map(location=[lat_newyork, lon_newyork],zoom_start=14)

# set color scheme for the clusters


# add markers to the map
markers_colors={}
markers_colors[0] = 'red'
markers_colors[1] = 'blue'
markers_colors[2] = 'green'
markers_colors[3] = 'yellow'
markers_colors[4] = 'cyan'
markers_colors[5] = 'black'
for lat, lon, cluster in zip(to_merged['Neighborhood Latitude'], to_merged['Neighborhood Longitude'], to_merged['Cluster Labels']):


    folium.features.CircleMarker(
        [lat, lon],
        radius=5,

        color =markers_colors[cluster],
        fill_color=markers_colors[cluster],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

In [54]:
```
#Cluster 1
to_merged.loc[(to_merged['Cluster Labels'] ==1) & (to_merged['Venue Category'] == 'Indian Restaurant') ]
```

Out[54]:

	Neighborhood	Indian Restaurant	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
99	Floral Park	0.375000	1	40.741378	-73.708847	Flavor Of India	40.737425	-73.708540	Indian Restaurant
99	Floral Park	0.375000	1	40.741378	-73.708847	Shahi Darbar	40.737488	-73.710022	Indian Restaurant
141	Jamaica Estates	0.333333	1	40.716805	-73.787227	Dhaka Hajir Biryani	40.720989	-73.786781	Indian Restaurant
99	Floral Park	0.375000	1	40.741378	-73.708847	Namaste Restaurant and Cafe	40.737173	-73.709756	Indian Restaurant

Most of the Indian restaurants are in cluster 2 which is around Bayside, Astoria, Greenwich Village. And lowest in cluster 0 and 3 areas which are in midtown, noho, and north side areas.also, there are good opportunities to open new "Indian Hyderabad restaurant" in this area. So stakeholders can choose this area.

In [55]:
```
#Cluster 2

to_merged.loc[(to_merged['Cluster Labels'] ==2) & (to_merged['Venue Category'] == 'Indian Restaurant') ]
```

Out[55]:

	Neighborhood	Indian Restaurant	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
14	Bayside	0.054054	2	40.766041	-73.774274	Masala Box	40.762674	-73.770682	Indian Restaurant
14	Bayside	0.054054	2	40.766041	-73.774274	masalabox	40.762670	-73.770690	Indian Restaurant
14	Bayside	0.054054	2	40.766041	-73.774274	Ayna Agra Indian Restaurant	40.765478	-73.771737	Indian Restaurant
14	Bayside	0.054054	2	40.766041	-73.774274	Agra Indian Cuisine	40.765396	-73.771535	Indian Restaurant
164	Manhattanville	0.022727	2	40.816934	-73.957385	Chapati House - NYC	40.814572	-73.959154	Indian Restaurant
277	Upper West Side	0.034884	2	40.787658	-73.977059	Swagat	40.783573	-73.978030	Indian Restaurant
144	Kew Gardens	0.043478	2	40.705179	-73.829819	Mehak Mughlai Cuisine	40.709164	-73.829509	Indian Restaurant

In [56]:
```
#Cluster 3

to_merged.loc[(to_merged['Cluster Labels'] ==3) & (to_merged['Venue Category'] == 'Indian Restaurant') ]
```

Out[56]:

	Neighborhood	Indian Restaurant	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category

Insights

VI CONCLUSION

Most of the Indian restaurants are in cluster 2 which is around Bayside, Astoria, Greenwich Village. And lowest in cluster 0 and 3 areas which are in midtown, noho, and north side areas.also, there are good opportunities to open new "Indian Hyderabad restaurant" in this area. So stakeholders can choose this area.