Capstone Project: Britties Yoga Snack

Cesar Garcia

Content

Introduction

Problem Description

"Britties" is a new snack, created and developed for the yoga community. It was created by three students who were studying business at New York University (NYU) and wanted to start their entrepreneurial life. One of them was also a yoga professor, who saw a need of a new snack for the yoga community. The product they created has the best taste and amount of protein per snack and they claimed that it would be a hit their product. In order validate their idea , the want to launch a pilot in the NY area, but due the limitation in their budget, they must focus in just one area of NY to do the pilot, so they want to apply their data science knowledge in order to find the proper area .

Stakeholders Definition

The main stakeholders would be the NY citizens, the yoga community and the three fellow students

Data Description

The data that would be used for the project will be the NY Map with the neighbor's name, latitude, longitude, among other things given by the IBM Specialization, and the data of the yoga places that will be segmented in Foursquare by the category ID:4bf58dd8d48988d102941735

Methodology

The methodology that was used was the same as the data science methodology course with the following steps in order to solve the problem

1.Business Understanding
2.Analytic Approach
3. Data Requirement
4.Data Collection
5.Data Understanding
6.Data Preparation
7.Modeling
8.Evaluation
9.Deployment

Results

The first thing that was done was to import all the data in order to start transforming it.

2-Importing the data

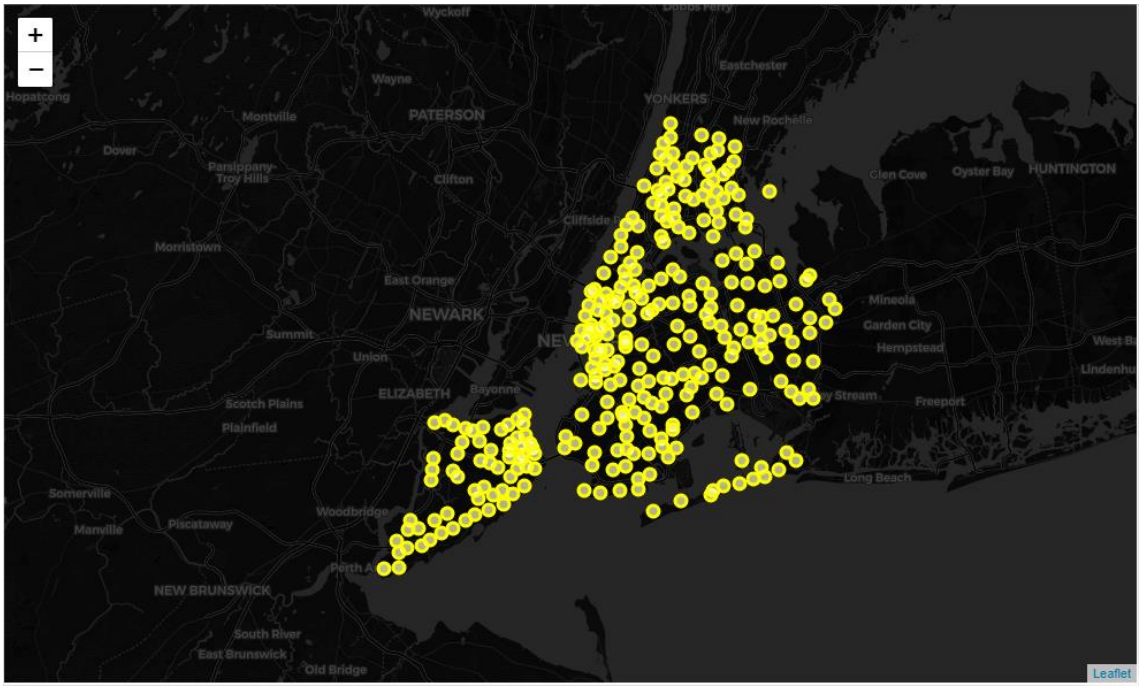
In [2]:
```
import urllib.request
url = 'https://cocl.us/new_york_dataset'
filename = 'newyork_data.json'
urllib.request.urlretrieve(url, filename)
print('Data downloaded!')
```

```
Data downloaded!
```

In [3]:
```
with open('newyork_data.json') as NYDF:
    NYDF = json.load(NYDF)
```

	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

After importing the first data set, we can visualize the NY distribution that has 5 boroughs and different neighborhoods. Now what we would do is to connect this location with the category data in Foursquare.



After connecting the data with the foursquare API, limiting the result to 500 venue (maximum venues with the current account), we got the following results for the yoga places (category id:4bf58dd8d48988d102941735) in NY.

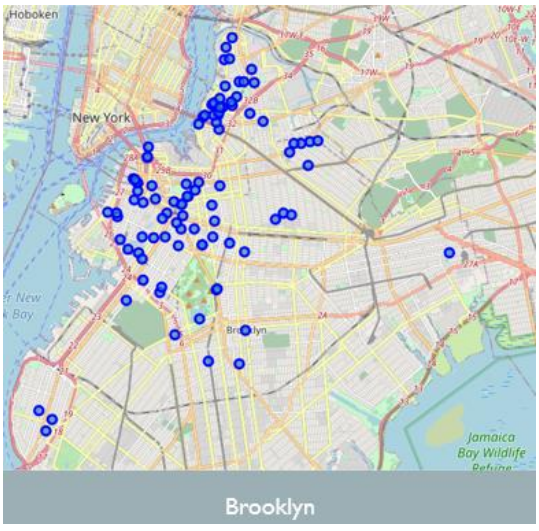
With this data let's calculate the distribution of the yoga places

```
maps = {}
for borough in boroughs:
    borough_lat = np.mean([results[borough]['response']['geocode']['geometry']['bounds']['ne']['lat'],
                           results[borough]['response']['geocode']['geometry']['bounds']['sw']['lat']])
    borough_lng = np.mean([results[borough]['response']['geocode']['geometry']['bounds']['ne']['lng'],
                           results[borough]['response']['geocode']['geometry']['bounds']['sw']['lng']])
    maps[borough] = folium.Map(location=[borough_lat, borough_lng], zoom_start=11)

    # add markers to map
    for lat, lng, label in zip(df_yoga[borough]['Lat'], df_yoga[borough]['Lng'], df_yoga[borough]['Name']):
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='#3186cc',
            fill_opacity=0.7,
            parse_html=False).add_to(maps[borough])
    print(f"Numbers of Yoga Studio in {borough} = ", results[borough]['response']['totalResults'])
    print("Showing Top 100")
```

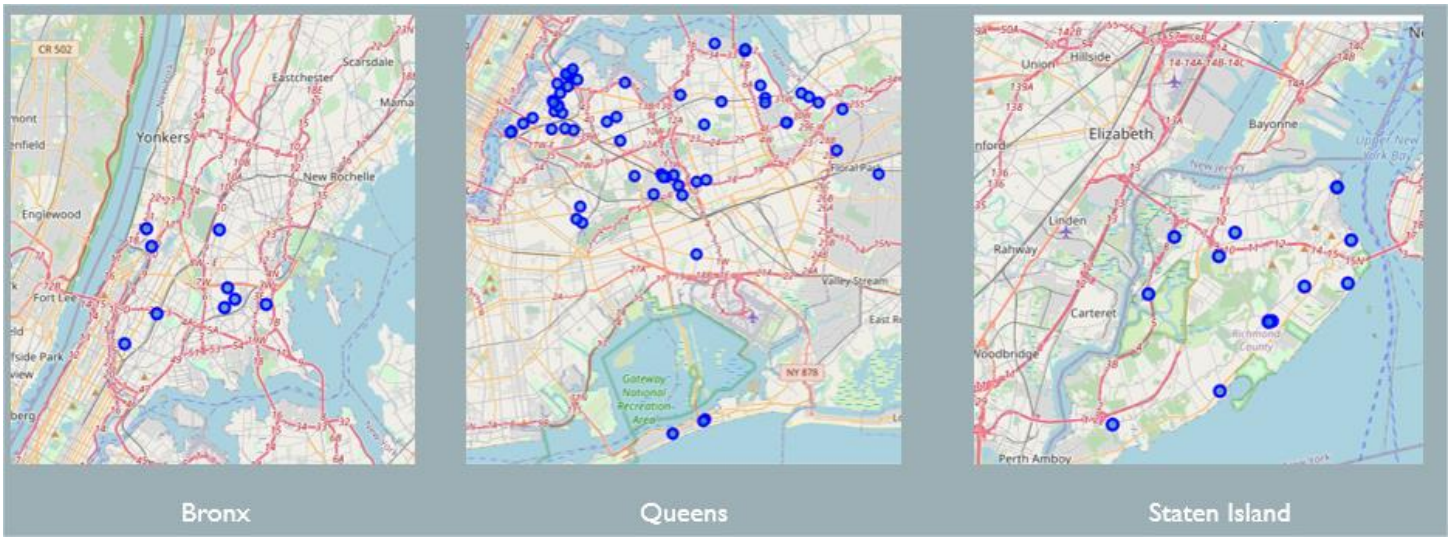
- Numbers of Yoga Studio in Bronx = 9
- Numbers of Yoga Studio in Manhattan = 227
- Numbers of Yoga Studio in Brooklyn = 189
- Numbers of Yoga Studio in Queens = 64
- Numbers of Yoga Studio in Staten Island = 14

We can visualize that the two most important places for the yoga studio are Manhattan and Brooklyn. Now let's visualize the results:



Brooklyn

Manhattan



Now that we have all the points and we already know that Brooklyn and Manhattan are the boroughs who have more yoga places, is necessary to calculate the density of the points, because , the young students want to focus just in one area, so we calculate the mean distance obtaining the following results:

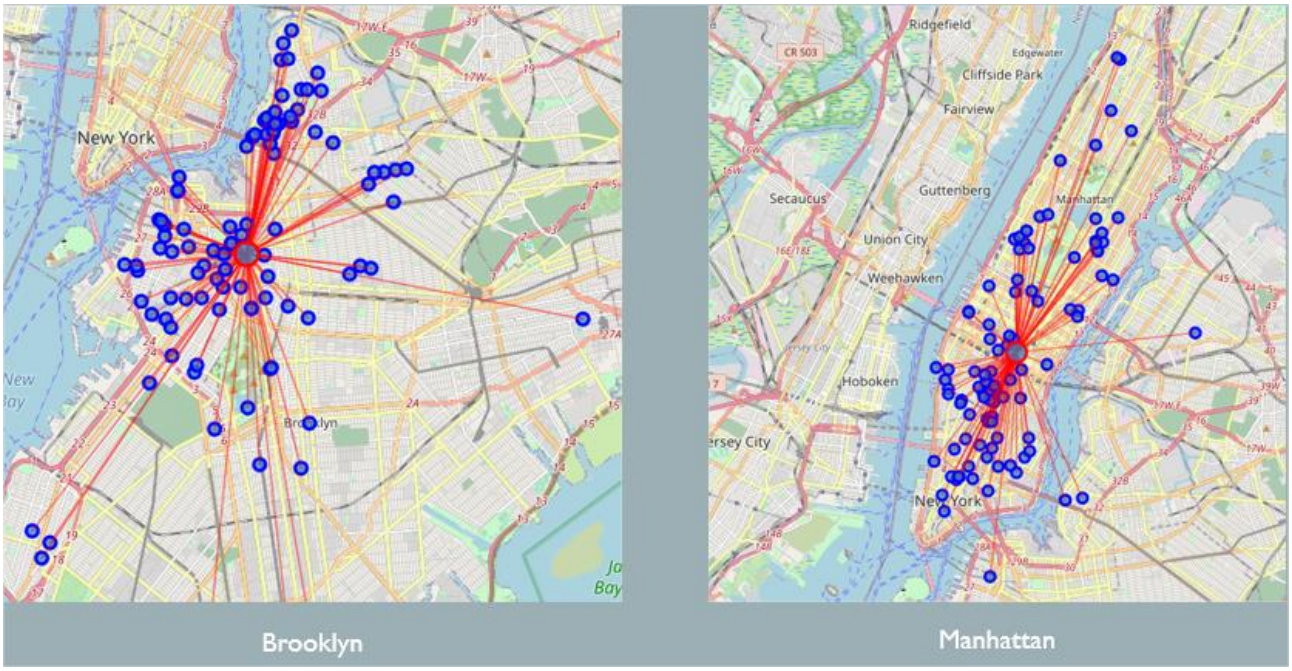
```
maps2 = {}
for borough in boroughs:
    borough_lat = np.mean([results[borough]['response']['geocode']['geometry']['bounds']['ne']['lat'],
                           results[borough]['response']['geocode']['geometry']['bounds']['sw']['lat']])
    borough_lng = np.mean([results[borough]['response']['geocode']['geometry']['bounds']['ne']['lng'],
                           results[borough]['response']['geocode']['geometry']['bounds']['sw']['lng']])
    maps2[borough] = folium.Map(location=[borough_lat, borough_lng], zoom_start=10)
    venues_mean_coor = [df_yoga[borough]['Lat'].mean(), df_yoga[borough]['Lng'].mean()]
    # add markers to map
    for lat, lng, label in zip(df_yoga[borough]['Lat'], df_yoga[borough]['Lng'], df_yoga[borough]['Name']):
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='#3186cc',
            fill_opacity=0.7,
            legend_name='borough:'+ borough,
            parse_html=False).add_to(maps2[borough])
        folium.PolyLine([venues_mean_coor, [lat, lng]], color="red", weight=1.5, opacity=0.5).add_to(maps2[borough])

    label = folium.Popup("Mean Co-ordinate", parse_html=True)
    folium.CircleMarker(
        venues_mean_coor,
        radius=10,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        caption=''+borough,
        parse_html=False).add_to(maps2[borough])

    print(borough)
    print("Mean Distance from Mean coordinates")
    print(np.mean(np.apply_along_axis(lambda x: np.linalg.norm(x - venues_mean_coor),1,df_yoga[borough][['Lat','Lng']].values)))
```

- Bronx Mean Distance from Mean coordinates: 0.03473164580176437
- Manhattan Mean Distance from Mean coordinates: 0.027892266753226636
- Brooklyn Mean Distance from Mean coordinates: 0.031715273837678366
- Queens Mean Distance from Mean coordinates: 0.07369320949347496
- Staten Island Mean Distance from Mean coordinates: 0.050034817407765574

We can see that Manhattan is the borough with more density of places, following in second place by Brooklyn



Discussion

After evaluating the yoga places distribution in NY, me can highly recommend trying the launch pilot in Manhattan, because it is the place with more yoga studios and with the highest density. However, if the budget of the students is really limited Brooklyn can be also an optimal solution according with the enterprise, because it has a good number of yoga places and a high density within the area

Conclusion

After making this project we have learned about the infinities of things that can be make with data science and with such a powerful tool as it I python with their package.

Also, we learn that not always the most obvious answer will be the right one. It is necessary to evaluate the need of the business and fitting it, for that reason I believe that the best proper solution to launching the pilot will be in Brooklyn due to the cost. Manhattan is one of the most expensive places in the world, so even though that it has more places to look , will be to expensive for the students, for that reason I highly recommend to launch the pilot in Brooklyn in order to succeed with the objective that was to validate the idea