Peergraded Assignment Capstone Project - The Battle of Neighborhoods (report)

January 17, 2020

0.1 Peer-graded Assignment: Capstone Project - The Battle of Neighborhoods (Report)

0.2 Introduction/Business Problem

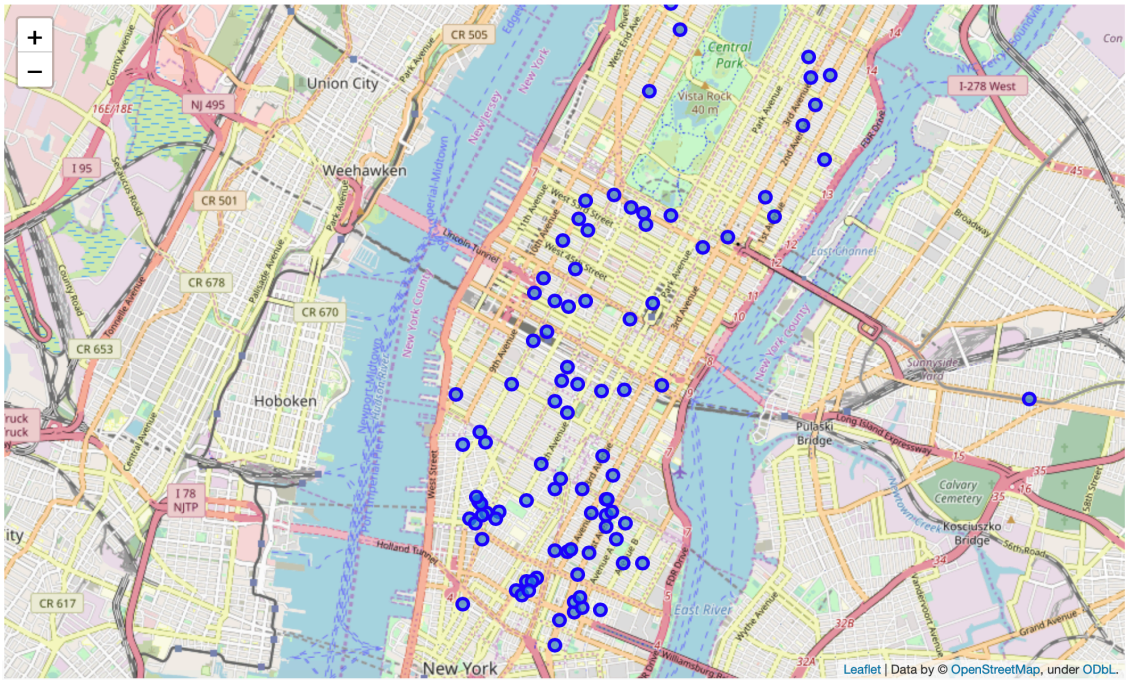
Many tourists look for fast and sometimes cheaper options for food on their trips. Being in unknown cities brings the challenge of finding the best options and technology can be a great ally. In this experiment, let's consider a tourist visiting the U.S. and looking for the highest concentration of pizza places in some cities.

1 Data

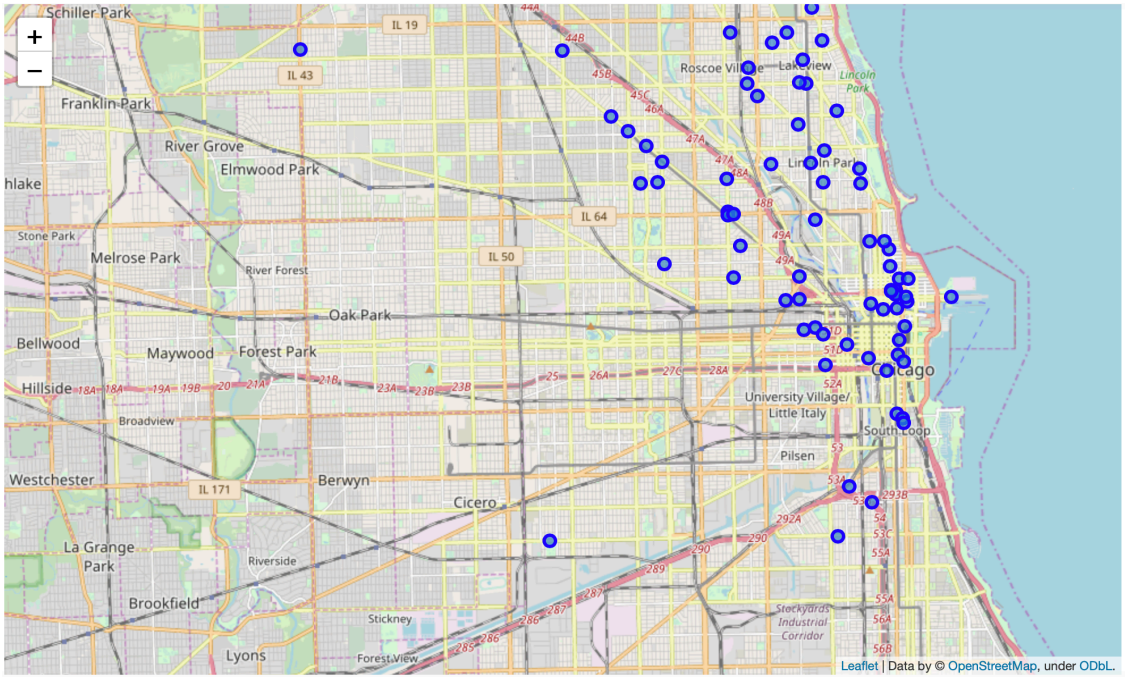
We will use Foursquare to gather information about the most concentrated pizza places in five major cities in the US: Chicago, NY, San Francisco, Jersey and Boston

```
In [15]: from IPython.display import display, Image
```

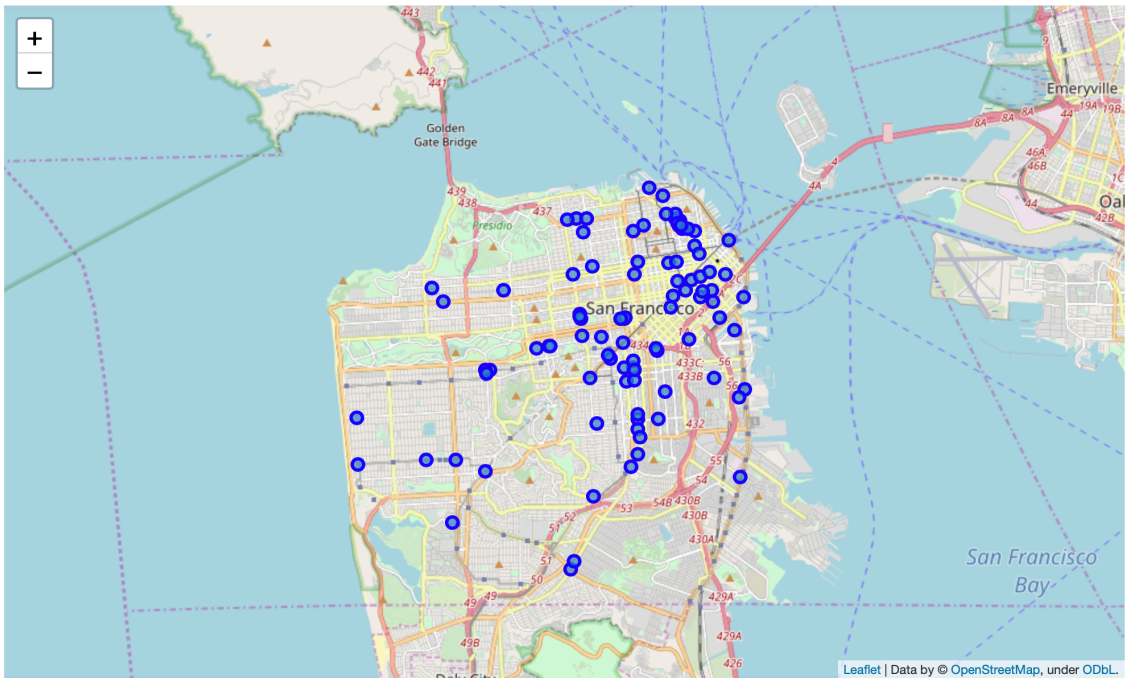
```
In [4]: display(Image(filename='img1.png'))
```



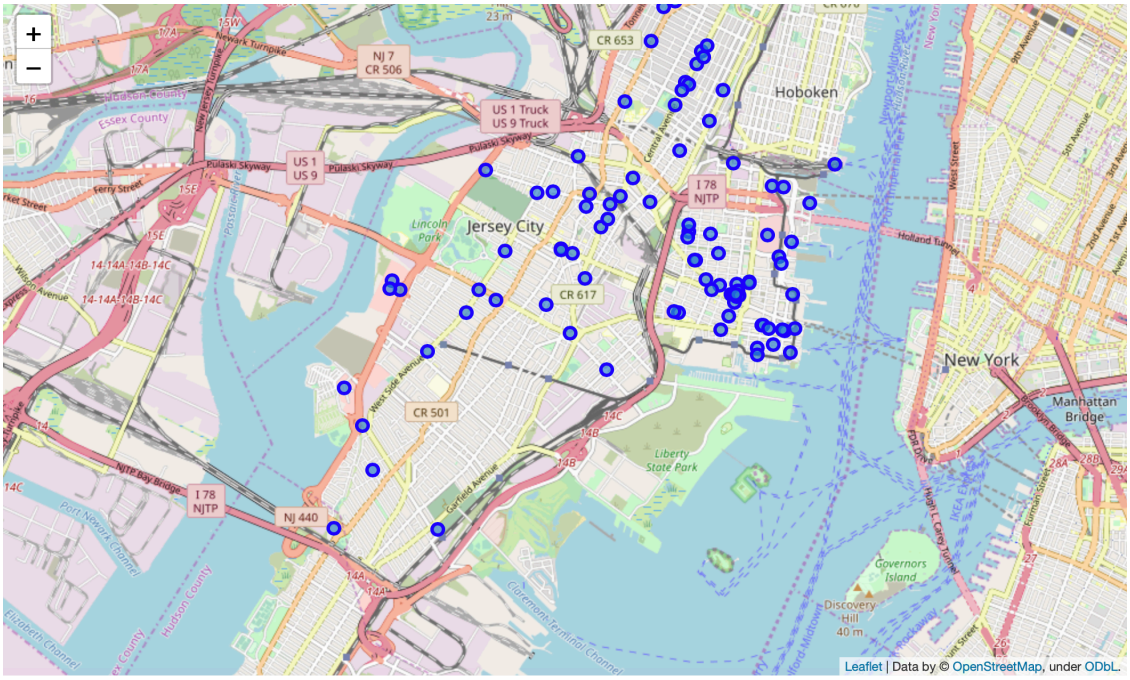
In [5]: display(Image(filename='img2.png'))



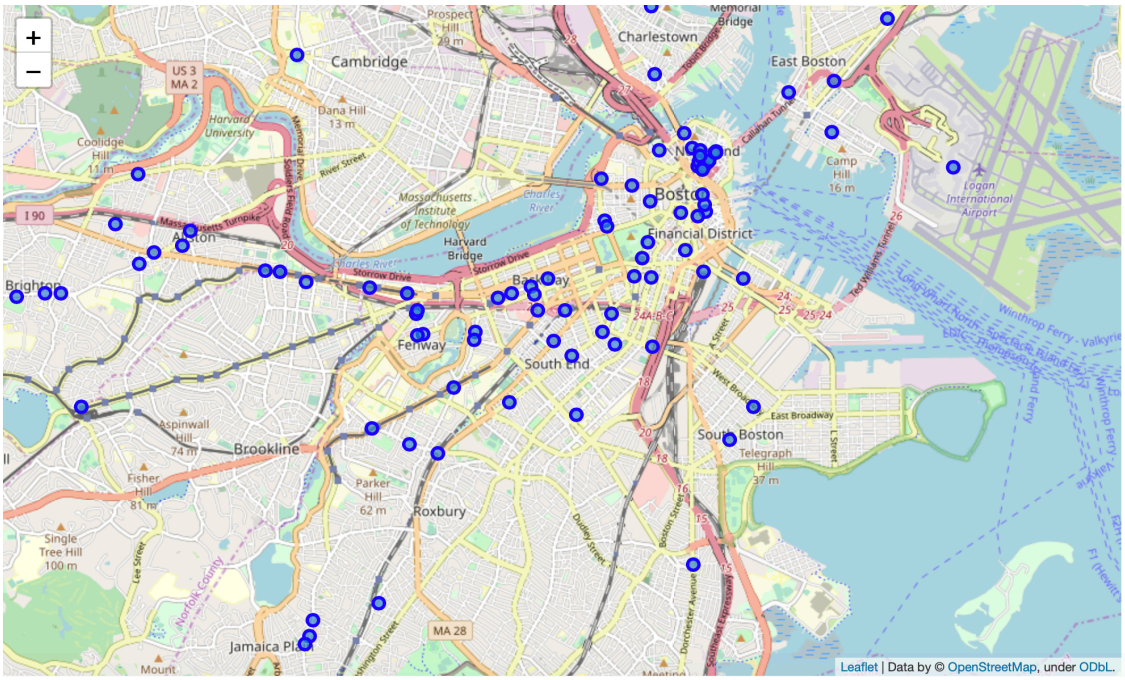
In [6]: display(Image(filename='img3.png'))



In [8]: display(Image(filename='img4.png'))



In [9]: display(Image(filename='img5.png'))



1.1 Methodology

Using foursquare to get data for analysis, to making a API call through the venues channel by categoryID to show Pizza Places, limited to max of 100 registers per query. (one query per city).

After get the data, we can plot the register on the map to visualize the pizza place concentration by city. In the sequence, we can use the parameters of distance and coordinate to calculate the mean and reach out the density concentration os pizza places.

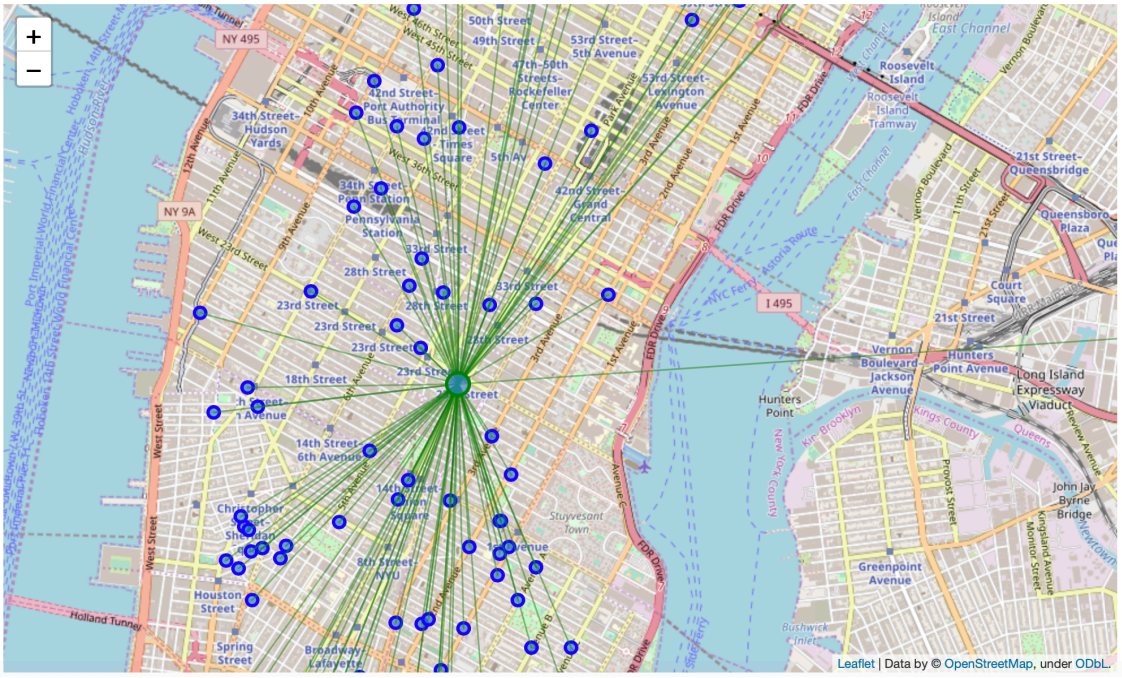
```
In [11]: maps = {}
         for city in cities:
             city_lat = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne
                                 results[city]['response']['geocode']['geometry']['bounds']['su
             city_lng = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne
                                 results[city]['response']['geocode']['geometry']['bounds']['su
             maps[city] = folium.Map(location=[city_lat, city_lng], zoom_start=11)
             venues_mean_coor = [df_venues[city]['Lat'].mean(), df_venues[city]['Lng'].mean()]
             # add markers to map
             for lat, lng, label in zip(df_venues[city]['Lat'], df_venues[city]['Lng'], df_venu
                 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[city])
                 folium.PolyLine([venues_mean_coor, [lat, lng]], color="green", weight=1.5, opa

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

             print(city)
             print("Mean Distance from Mean coordinates")
             print(np.mean(np.apply_along_axis(lambda x: np.linalg.norm(x - venues_mean_coor),1
```

```
New York, NY
Mean Distance from Mean coordinates
0.022123441105883532
Chicago, IL
Mean Distance from Mean coordinates
0.0593422082245271
San Francisco, CA
Mean Distance from Mean coordinates
0.02825277263662251
Jersey City, NJ
Mean Distance from Mean coordinates
0.01915517535258975
Boston, MA
Mean Distance from Mean coordinates
0.03557142430375053
```

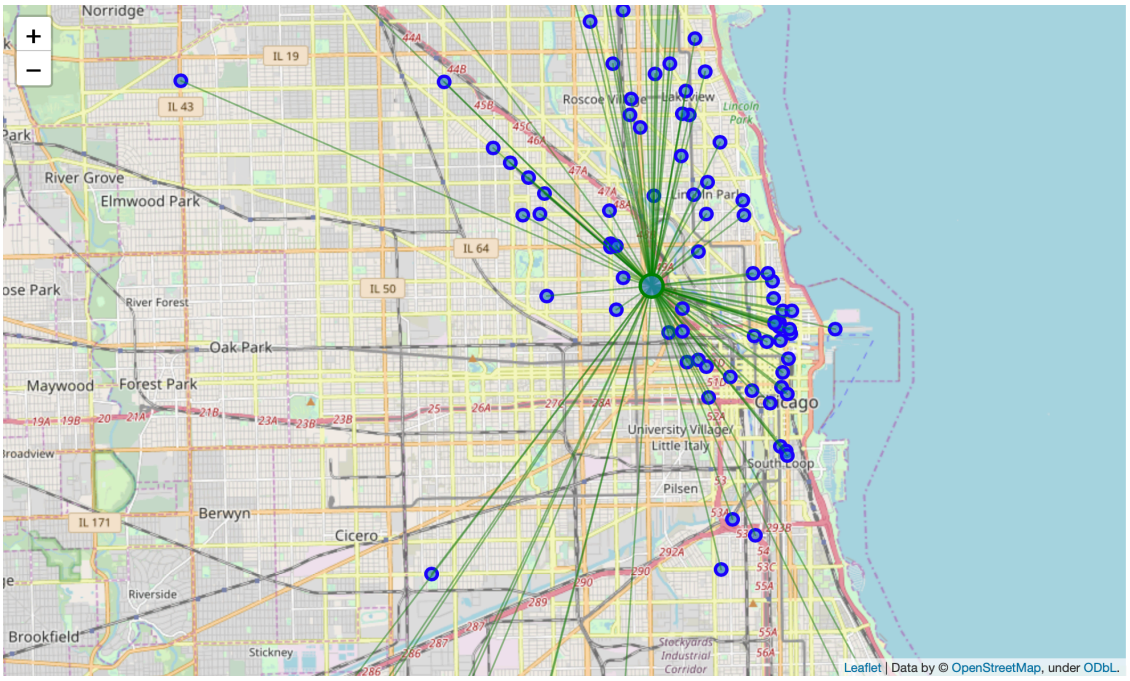
1.2 Results

The geoplots generated bellow show us we have several pizza places as a option.

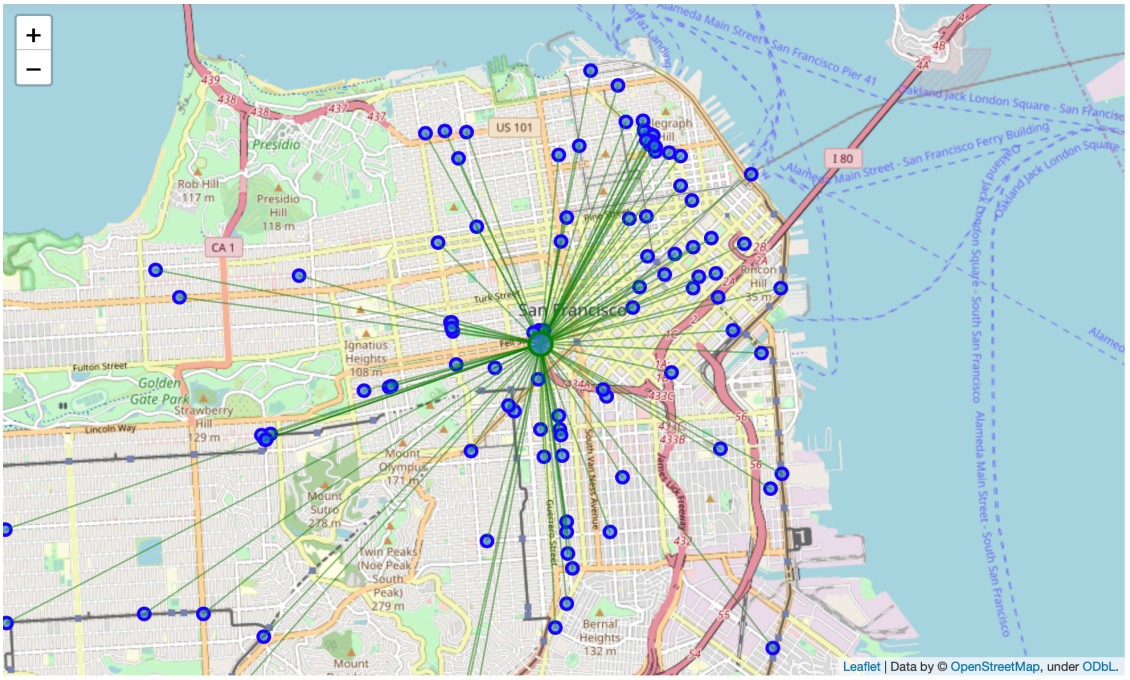
```
In [10]: display(Image(filename='img6.png'))
```



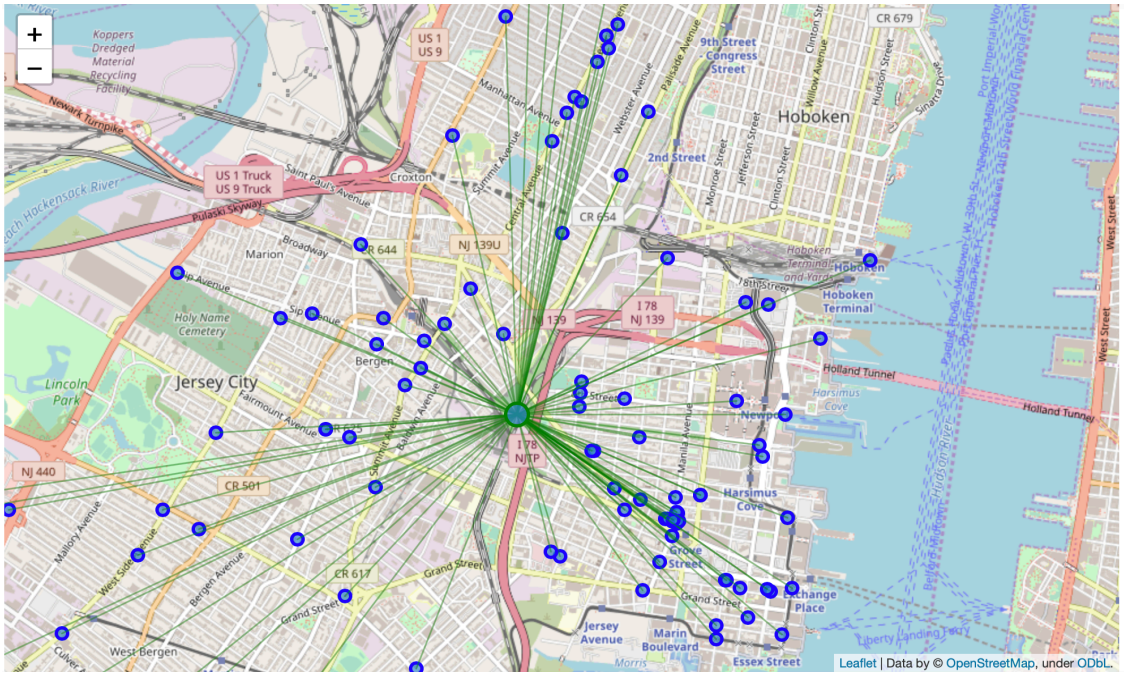

```
In [11]: display(Image(filename='img7.png'))
```



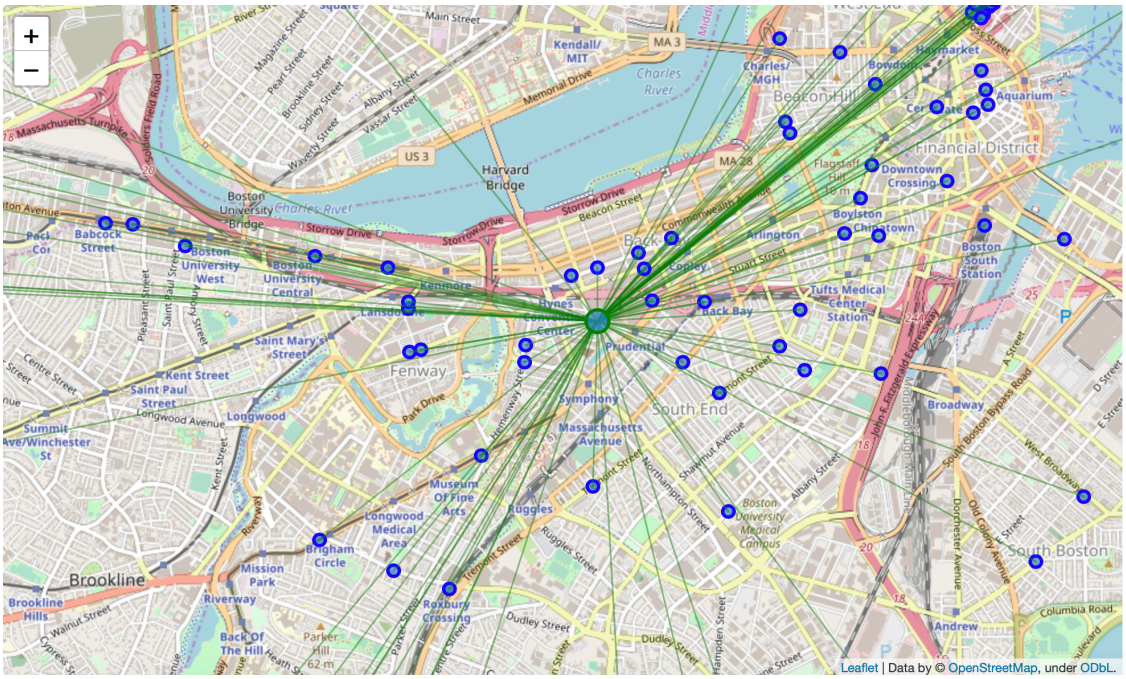
In [12]: display(Image(filename='img8.png'))



In [13]: display(Image(filename='img9.png'))



In [14]: display(Image(filename='img10.png'))



In [17]: city = 'Jersey City, NJ'
venues_mean_coor = [df_venues[city]['Lat'].mean(), df_venues[city]['Lng'].mean()]

```
        print(city)
        print("Mean Distance from Mean coordinates")
        dists = np.apply_along_axis(lambda x: np.linalg.norm(x - venues_mean_coor),1,df_venues
        dists.sort()
        print(np.mean(dists[:-1]))# Ignore the biggest distance
```

```
Jersey City, NJ
Mean Distance from Mean coordinates
0.018736968844191782
```

1.3 Discussion

Jersey has the best mean distance from mean coordinate The mean distance from mean coordinate show us a very nice pizza place concentration overview per city, it would be nice consider a wide range of city on our experiement and repeat it with a lot of more data.

1.4 Conclusion

As expected, NY is no doubt the best option to have some pizza with many pizza places options and we got Jersey as a great nearby option.