Capstone Project - The Battle of Neighborhoods

Finding the most populated US cities with highest density of Peruvian restaurants

1-Description of the problem and discussion of the background

Scenario

A businessman in the importing and delivery industry has noticed how difficult it is for international cuisine restaurant owners to get specific ingredients, that are only available in their native countries; those groceries and fresh produce are essential to provide customers with authentic recipes and dishes.

Business Problem

The businessman is a Peruvian food aficionado and he is interested in importing those special ingredients to Peruvian restaurants in US; however, he needs some help in deciding what city is the best option for his business.

He is interested in high density of restaurants within the selected city, so he can get the most revenue.

I have been given the exciting task of assisting him to make data-driven decisions on what cities are suitable for his needs. This will be a major part of his decision-making process.

2-A description of the data and how it will be used to solve the problem

Why using data?

Without leveraging data to make decisions about this new enterprise, my customer could spend countless hours walking around spending precious time and efforts and ending choosing a city that is not the best option.

Data will provide better answers and better solutions to this task at hand.

How the data will be used to solve the problem?

I will concentrate in finding the top 5 ranked cities in US by population, and using an API to get the number of Peruvian restaurants in those cities; then analyze that data to get the mean coordinates and the mean distances to mean coordinate(MDMC) for its restaurants, to calculate density and display the findings in map charts.

The best city for the goods delivery business will be the one with a combination of highest number of restaurants and at the same time the lowest mean distances to mean coordinate average of the restaurants.

Data sources

Top 5 populated cities in US will be retrieved from Wikipedia:

- Source: https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population

2018 rank	City	State[c]	2018 estimate	2010 Census	Change	2016 land area		2016 population density		Location
1	New York[d]	New York	8,398,748	8,175,133	+2.74%	301.5 sq mi	780.9 km^2	28,317/sq mi	10,933/km^2	40.6635°N 73.9387°W
2	Los Angeles	California	3,990,456	3,792,621	+5.22%	468.7 sq mi	1,213.9 km^2	8,484/sq mi	3,276/km^2	34.0194°N 118.4108°W
3	Chicago	Illinois	2,705,994	2,695,598	+0.39%	227.3 sq mi	588.7 km^2	11,900/sq mi	4,600/km^2	41.8376°N 87.6818°W
4	Houston[3]	Texas	2,325,502	2,100,263	+10.72%	637.5 sq mi	1,651.1 km^2	3,613/sq mi	1,395/km^2	29.7866°N 95.3909°W
5	**Phoenix**	Arizona	1,660,272	1,445,632	+14.85%	517.6 sq mi	1,340.6 km^2	3,120/sq mi	1,200/km^2	33.5722°N 112.0901°W
6	Philadelphia[e]	Pennsylvania	1,584,138	1,526,006	+3.81%	134.2 sq mi	347.6 km^2	11,683/sq mi	4,511/km^2	40.0094°N 75.1333°W
7	San Antonio	Texas	1,532,233	1,327,407	+15.43%	461.0 sq mi	1,194.0 km^2	3,238/sq mi	1,250/km^2	29.4724°N 98.5251°W
8	San Diego	California	1,425,976	1,307,402	+9.07%	325.2 sq mi	842.3 km^2	4,325/sq mi	1,670/km^2	32.8153°N 117.1350°W
9	Dallas	Texas	1,345,047	1,197,816	+12.29%	340.9 sq mi	882.9 km^2	3,866/sq mi	1,493/km^2	32.7933°N 96.7665°W
10	San Jose	California	1,030,119	945,942	+8.90%	177.5 sq mi	459.7 km^2	5,777/sq mi	2,231/km^2	37.2967°N 121.8189°W

Foursquare API will be used to get restaurant data within those cities:

- Source: https://foursquare.com/developers/

Importing Libraries and Required Data

Importing the required libraries for the project

```
import numpy as np # library to handle data in a vectorized manner
import pandas as pd # library for data analsysis and data manipulation
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

Installing Folium to display map charts

```
!conda install -c conda-forge folium=0.5.0
```

```
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
    ---------------------------|-----------------
    folium-0.5.0               |             py_0          45 KB  conda-forge
    python_abi-3.6             |          1_cp36m           4 KB  conda-forge
    altair-4.0.1               |             py_0         575 KB  conda-forge
    certifi-2019.11.28         |   py36h9f0ad1d_1         149 KB  conda-forge
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    branca-0.4.0               |             py_0          26 KB  conda-forge
    openssl-1.1.1e             |       h516909a_0         2.1 MB  conda-forge
    ca-certificates-2019.11.28 |       hecc5488_0         145 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         3.1 MB

The following NEW packages will be INSTALLED:

    altair:          4.0.1-py_0        conda-forge
    branca:          0.4.0-py_0        conda-forge
    folium:          0.5.0-py_0        conda-forge
    python_abi:      3.6-1_cp36m       conda-forge
    vincent:         0.4.4-py_1        conda-forge

The following packages will be UPDATED:

    certifi:         2019.11.28-py36_0           --> 2019.11.28-py36h9f0ad1d_1 conda-forge
    openssl:         1.1.1e-h7b6447c_0           --> 1.1.1e-h516909a_0         conda-forge

The following packages will be DOWNGRADED:

    ca-certificates: 2020.1.1-0                  --> 2019.11.28-hecc5488_0     conda-forge


Downloading and Extracting Packages
folium-0.5.0         | 45 KB     | ##################################### | 100%
python_abi-3.6       | 4 KB      | ##################################### | 100%
altair-4.0.1         | 575 KB    | ##################################### | 100%
certifi-2019.11.28   | 149 KB    | ##################################### | 100%
vincent-0.4.4        | 28 KB     | ##################################### | 100%
branca-0.4.0         | 26 KB     | ##################################### | 100%
openssl-1.1.1e       | 2.1 MB    | ##################################### | 100%
ca-certificates-2019 | 145 KB    | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Importing the list of United States by Population:

```
df = pd.read_html('https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population')[4]
```

```
df.head()
```

	2018rank	City	State[c]	2018estimate	2010Census	Change	2016 land area	2016 land area.1	2016 population density	2016 population density.1	Location
0	1	New York[d]	New York	8398748	8175133	+2.74%	301.5 sq mi	780.9 km2	28,317/sq mi	10,933/km2	40°39′49″N 73°56′19″W / 40.6635°N 73.9387°W
1	2	Los Angeles	California	3990456	3792621	+5.22%	468.7 sq mi	1,213.9 km2	8,484/sq mi	3,276/km2	34°01′10″N 118°24′39″W / 34.0194°N 118.4108°W
2	3	Chicago	Illinois	2705994	2695598	+0.39%	227.3 sq mi	588.7 km2	11,900/sq mi	4,600/km2	41°50′15″N 87°40′54″W / 41.8376°N 87.6818°W
3	4	Houston[3]	Texas	2325502	2100263	+10.72%	637.5 sq mi	1,651.1 km2	3,613/sq mi	1,395/km2	29°47′12″N 95°23′27″W / 29.7866°N 95.3909°W
4	5	Phoenix	Arizona	1660272	1445632	+14.85%	517.6 sq mi	1,340.6 km2	3,120/sq mi	1,200/km2	33°34′20″N 112°05′24″W / 33.5722°N 112.0901°W

Replacing column name from 2018estimate to Population

```
df.rename(columns = {'2018estimate':'Population'}, inplace = True)
```

Removing unnecessary columns

```
df.drop(['2010Census', 'Change', 'Change', '2016 land area', '2016 land area.1', '2016 population density', '2016 population den
sity.1', 'Location'], axis=1)
```

	2018rank	City	State[c]	Population
0	1	New York[d]	New York	8398748
1	2	Los Angeles	California	3990456
2	3	Chicago	Illinois	2705994
3	4	Houston[3]	Texas	2325502
4	5	Phoenix	Arizona	1660272
5	6	Philadelphia[e]	Pennsylvania	1584138
6	7	San Antonio	Texas	1532233
7	8	San Diego	California	1425976
8	9	Dallas	Texas	1345047
9	10	San Jose	California	1030119
10	11	Austin	Texas	964254

Adding my Foursquare's Client ID + Client secret + today's date

```
CLIENT_ID = 'EXSEDHK41WBAR3KEAMJPMU2M3XF4IEFP4YY2ZGTZVMFPOMWC' # Foursquare ID
CLIENT_SECRET = 'PXAAOPATK1VC3XCUY05AKXS0W3ENP1DDNDDA0CXUM1AG4NI1' # Foursquare Secret
VERSION = '20200329' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
```

```
Your credentails:
CLIENT_ID: EXSEDHK41WBAR3KEAMJPMU2M3XF4IEFP4YY2ZGTZVMFPOMWC
```

Adding US top 5 cities by population, and Peruvian restaurant category ID

```
# foursquare data:
LIMIT = 1000 # Maximum is 100
cities = ['"New York, NY"', 'Los Angeles, CA', 'Chicago, IL', 'Houston, TX','Phoenix, AZ', ]
results = {}
for city in cities:
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&near={}&limit={}&categoryId={}'.form
at(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        city,
        LIMIT,
        "4eb1bfa43b7b52c0e1adc2e8") # Peruvian restaurant category ID
    results[city] = requests.get(url).json()
```

Getting the results

```
results
```


```
{'"New York, NY"': {'meta': {'code': 200,
    'requestId': '5e81593d60ba08001b63a896'},
   'response': {'suggestedFilters': {'header': 'Tap to show:',
     'filters': [{'name': 'Open now', 'key': 'openNow'},
      {'name': '$-$$$$', 'key': 'price'}]},
    'geocode': {'what': '',
     'where': 'new york ny',
     'center': {'lat': 40.742185, 'lng': -73.992602},
     'displayString': 'New York, NY, United States',
     'cc': 'US',
     'geometry': {'bounds': {'ne': {'lat': 40.882214, 'lng': -73.907},
       'sw': {'lat': 40.679548, 'lng': -74.047285}}},
     'slug': 'new-york-city-new-york',
     'longId': '72057594043056517'},
    'headerLocation': 'New York',
    'headerFullLocation': 'New York',
    'headerLocationGranularity': 'city',
    'query': 'peruvian',
    'totalResults': 44,
    'suggestedBounds': {'ne': {'lat': 40.841693907115,
      'lng': -73.87397987594642},
     'sw': {'lat': 40.66261360093418, 'lng': -74.0149807323931}},
    'groups': [{'type': 'Recommended Places',
      'name': 'recommended',
      'items': [{'reasons': {'count': 0,
         'items': [{'summary': 'This spot is popular',
           'type': 'general',
           'reasonName': 'globalInteractionReason'}]},
        'venue': {'id': '4b1b1f52f964a52074f823e3',
         'name': 'Pio Pio',
         'location': {'address': '604 10th Ave',
          'crossStreet': 'btwn W 43rd & W 44th St',
          'lat': 40.76063594478618,
          'lng': -73.99471374607128,
          'labeledLatLngs': [{'label': 'display',
            'lat': 40.76063594478618,
            'lng': -73.99471374607128}],
          'postalCode': '10036',
```

Getting restaurant attributes

```
df_venues={}
for city in cities:
    venues = json_normalize(results[city]['response']['groups'][0]['items'])
    df_venues[city] = venues[['venue.name', 'venue.location.address', 'venue.location.lat', 'venue.location.lng']]
    df_venues[city].columns = ['Name', 'Address', 'Lat', 'Lng']
```

Displaying restaurant attributes

```
df_venues
```

```
{'"New York, NY"':                              Name
 0                              Pio Pio            604 10th Ave
 1                         Flor de Mayo        484 Amsterdam Ave
 2                       Pio Pio Salon        702 Amsterdam Ave
 3                              Pio Pio            210 E 34th St
 4                              Pio Pio            1746 1st Ave
 5                            Llama-San  359 Avenue of the Americas
 6                                Chirp            369 W 34th St
 7                              Llamita            80 Carmine St
 8                                Panca               92 7th Ave
 9                 Riko Peruvian Cuisine              409 8th Ave
 10               Morocho Peruvian Fusion           West 52nd St.
 11                    Panca Cebiche Bar             92 7th Ave S
 12                         Le Bernardin           155 W 51st St
 13                           Baby Brasa            173 7th Ave S
 14                             Coppelia           207 W 14th St
 15                         Flor De Mayo            2651 Broadway
 16                  Chino's Rotisserie               23 Pell St
 17                           Sen Sakana             28 W 44th St
 18                     Mary's Fish Camp           64 Charles St
 19                              Desnuda             122 E 7th St
 20  El Sol Peruvian Cuisine Bar & Grill   8707 Northern Boulevard
 21    Mancora Peruvian Restaurant & Bar              99 1st Ave
 22                  Tutuma Social Club           164 E 56th St
 23                      Mission Ceviche             1400 2nd Ave
 24                 Sticky's Finger Joint              31 W 8th St
 25                     Nobu Fifty Seven             40 W 57th St
 26               Cascolate Latin Bistro             2126 2nd Ave
 27                                 Inti            820 10th Ave
 28                   Tina's Cuban Cuisine         179 Madison Ave
 29                 The Little Beet Table          333 Park Ave S
 30             Stumptown Coffee Roasters            18 W 29th St
 31                              Bubby's           120 Hudson St
 32                           The Tippler           425 W 15th St
 33                             Coco Roco             392 5th Ave
 34                         Lantern's Keep            49 W 44th St
 35                       Rosa's Empanadas                     NaN
 36           Irving Farm Coffee Roasters            1424 3rd Ave
 37                     Buddha Bodai 佛菩提素菜                 5 Mott St
 38               Restaurante La Libertad             3764 Broadway
 39                       Quechua Nostra       1634 Lexington Ave
 40                     Mama Pio Kitchen         53-05 65th Place
 41                          Guantanamera             939 8th Ave
 42                 Maison Saigon Tacu Tacu            134 N 6th St
 43                           la liberdad                 Broadway
```

Getting total restaurants per city and preparing venue data for plotting

```python
maps = {}
for city in cities:
    city_lat = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne']['lat'],
                        results[city]['response']['geocode']['geometry']['bounds']['sw']['lat']])
    city_lng = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne']['lng'],
                        results[city]['response']['geocode']['geometry']['bounds']['sw']['lng']])
    maps[city] = folium.Map(location=[city_lat, city_lng], zoom_start=11)

    # add markers to map
    for lat, lng, label in zip(df_venues[city]['Lat'], df_venues[city]['Lng'], df_venues[city]['Name']):
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='green',
            fill=True,
            fill_color='#86cc31',
            fill_opacity=0.8,
            parse_html=False).add_to(maps[city])
    print(f"Total number of Peruvian restuarants in {city} = ", results[city]['response']['totalResults'])
    print("Showing Top 100")
```

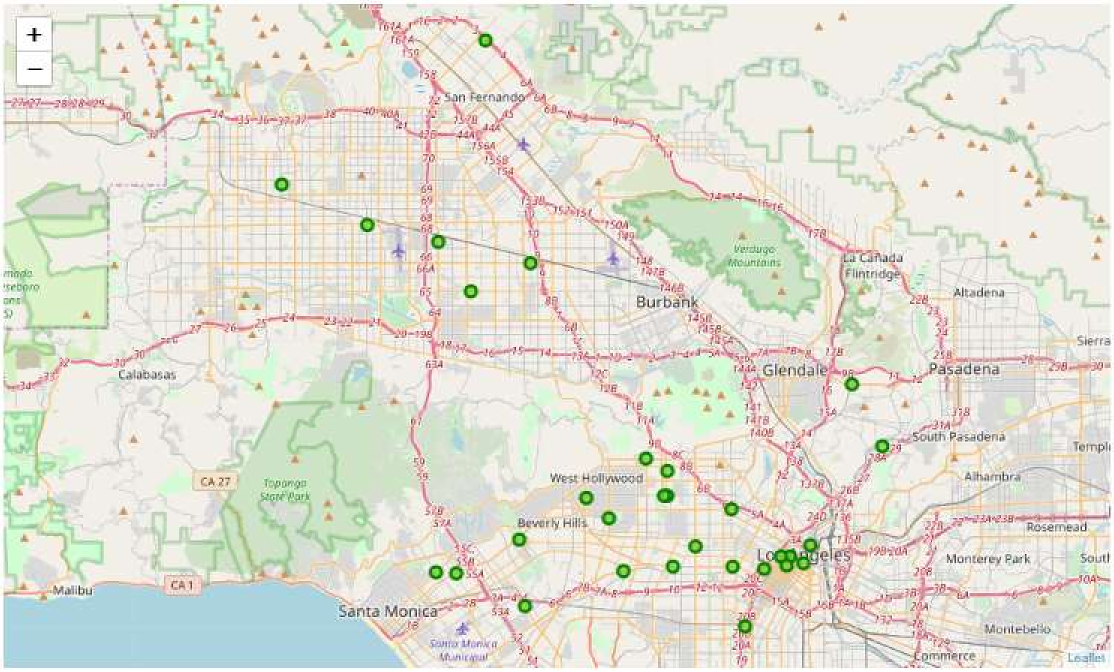
Results

```
Total number of Peruvian restuarants in "New York, NY" =  44
Showing Top 100
Total number of Peruvian restuarants in Los Angeles, CA =  34
Showing Top 100
Total number of Peruvian restuarants in Chicago, IL =  17
Showing Top 100
Total number of Peruvian restuarants in Houston, TX =  17
Showing Top 100
Total number of Peruvian restuarants in Phoenix, AZ =  7
Showing Top 100
```

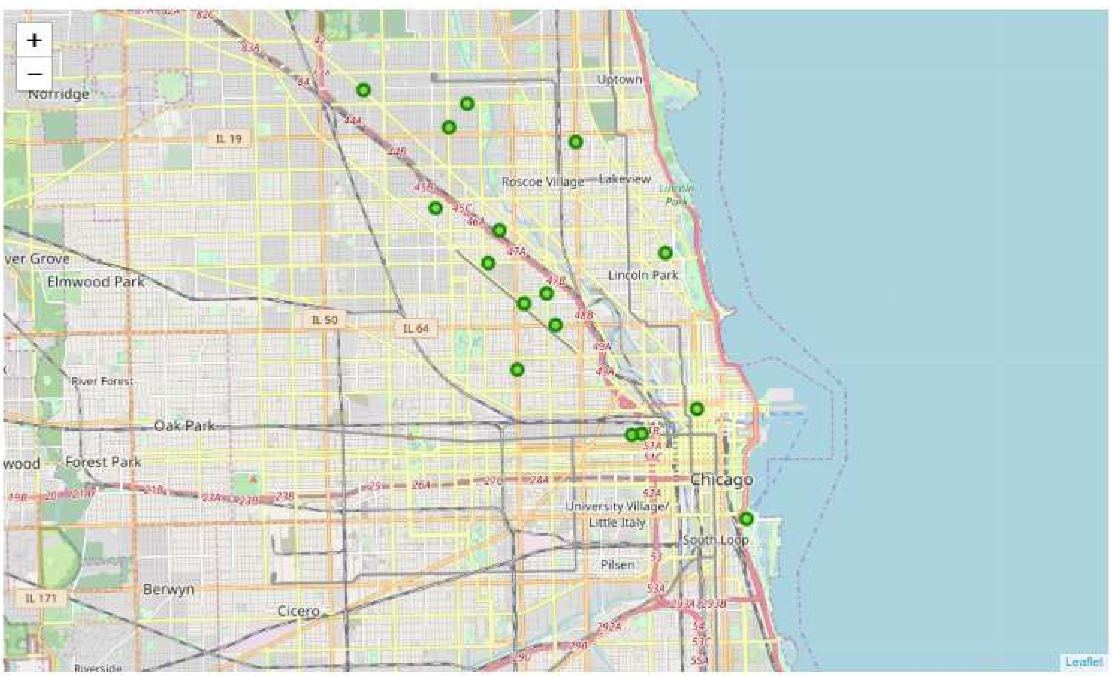
Plotting the results for each city

New York:

Los Angeles:

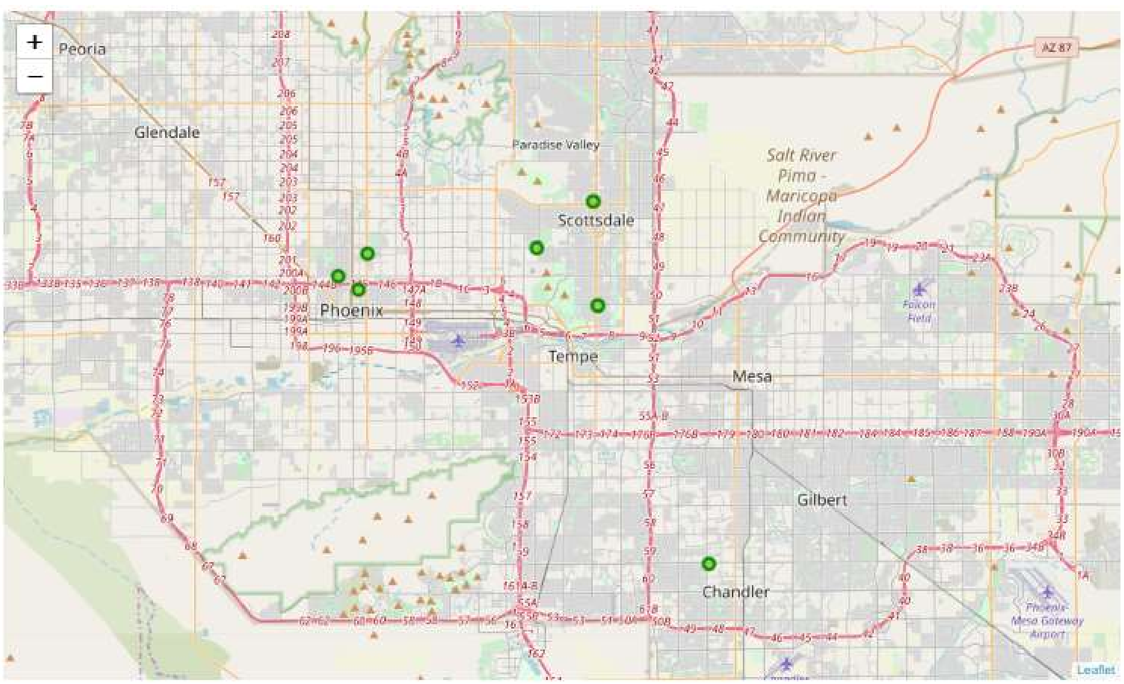


Chicago:



Houston:

Phoenix:



Calculating the mean location of Peruvian restaurants, and getting their average distance from the mean coordinates

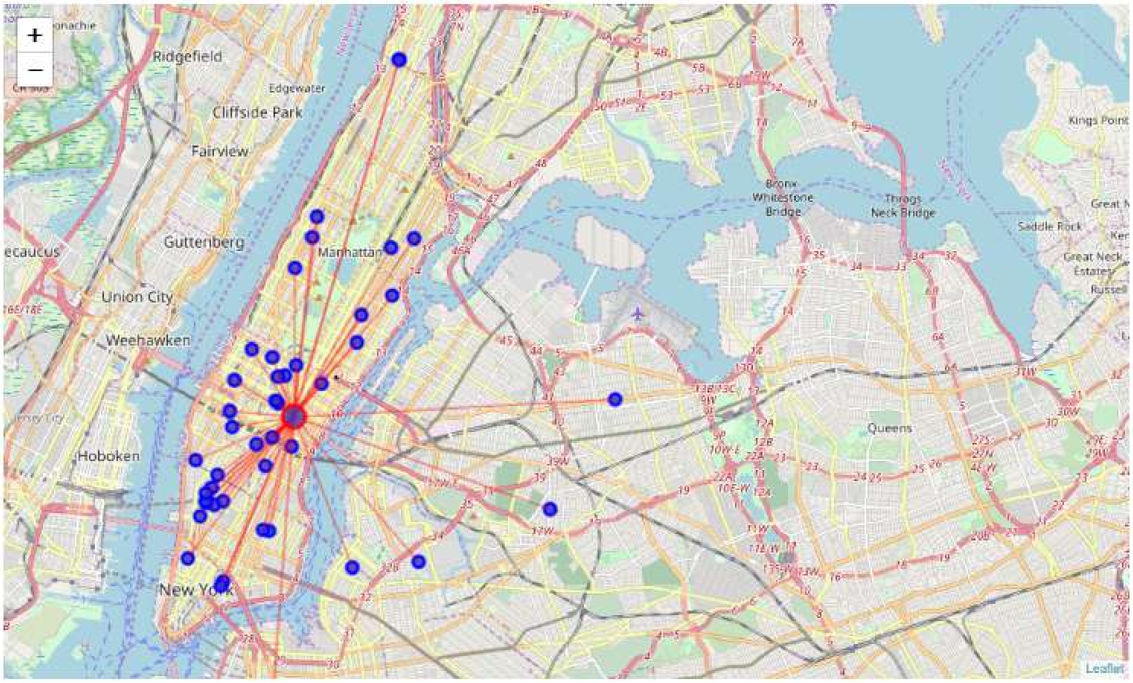
```
maps = {}
for city in cities:
    city_lat = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne']['lat'],
                        results[city]['response']['geocode']['geometry']['bounds']['sw']['lat']])
    city_lng = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne']['lng'],
                        results[city]['response']['geocode']['geometry']['bounds']['sw']['lng']])
    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_venues[city]['Name']):
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='#3138cc',
            fill_opacity=0.8,
            parse_html=False).add_to(maps[city])
        folium.PolyLine([venues_mean_coor, [lat, lng]], color="red", weight=1.5, opacity=0.5).add_to(maps[city])

    label = folium.Popup("Mean Co-ordinate", parse_html=True)
    folium.CircleMarker(
        venues_mean_coor,
        radius=10,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3138cc',
        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,df_venues[city][['Lat','Lng']].values)))
```

Results

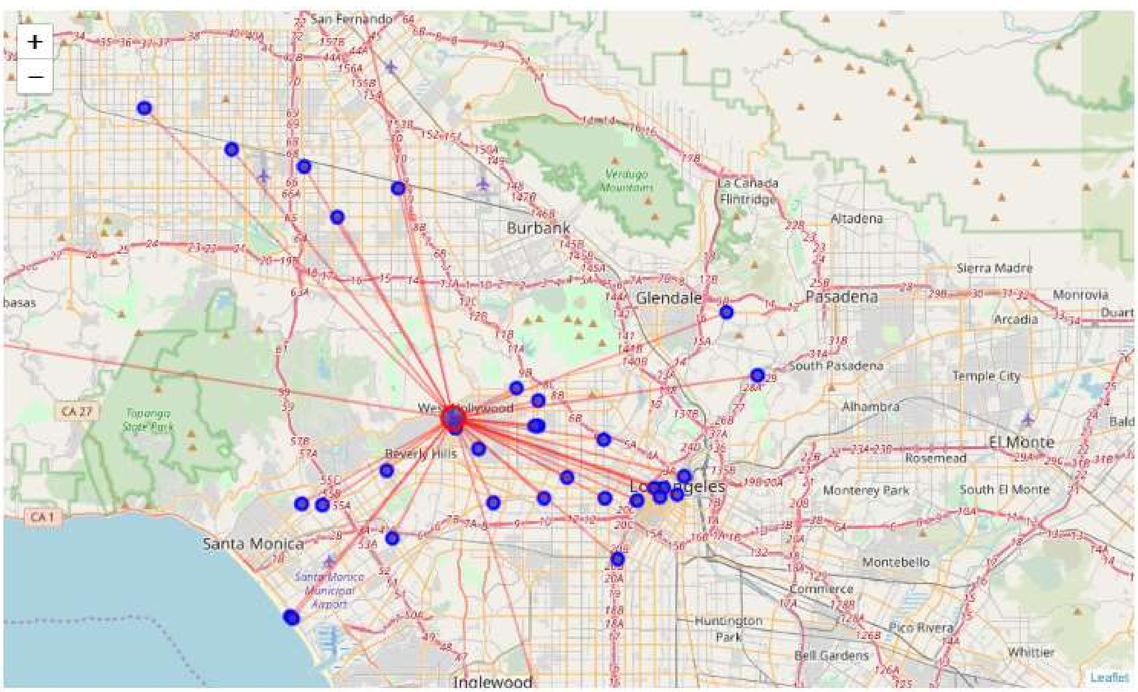
```
"New York, NY"
Mean Distance from Mean coordinates
0.03403361834108446
Los Angeles, CA
Mean Distance from Mean coordinates
0.1333208138726697
Chicago, IL
Mean Distance from Mean coordinates
0.03921846798477392
Houston, TX
Mean Distance from Mean coordinates
0.10194325397112565
Phoenix, AZ
Mean Distance from Mean coordinates
0.08928591991231158
```

Plotting the mean location of Peruvian restaurants, along with distances from the mean coordinates

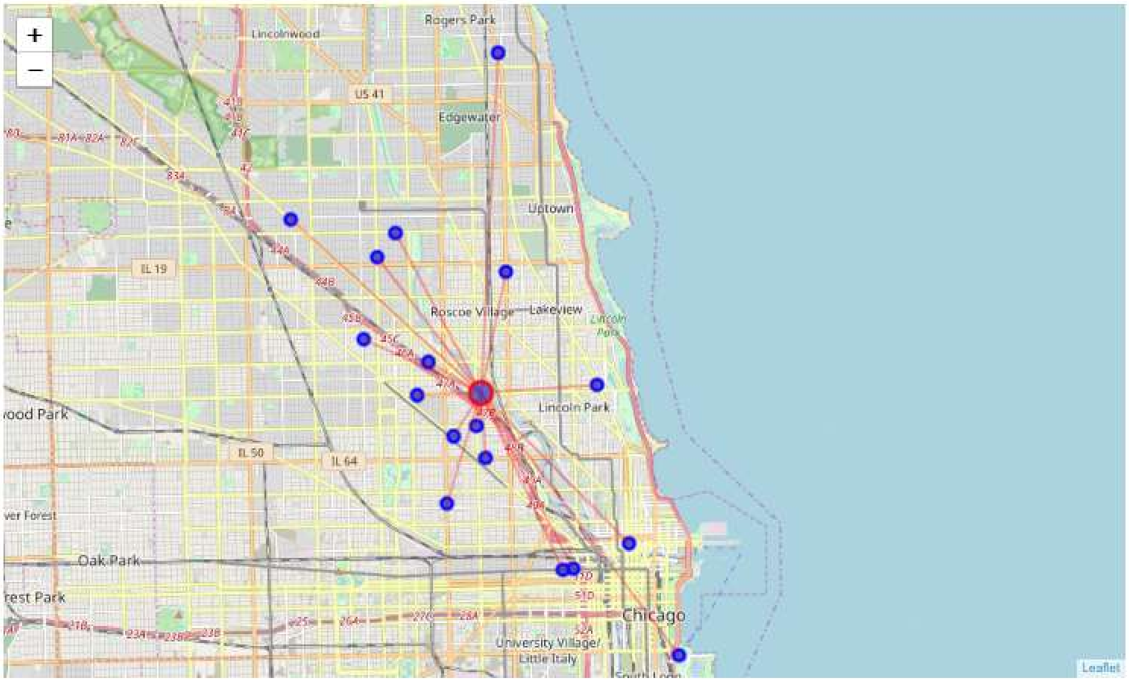
New York:



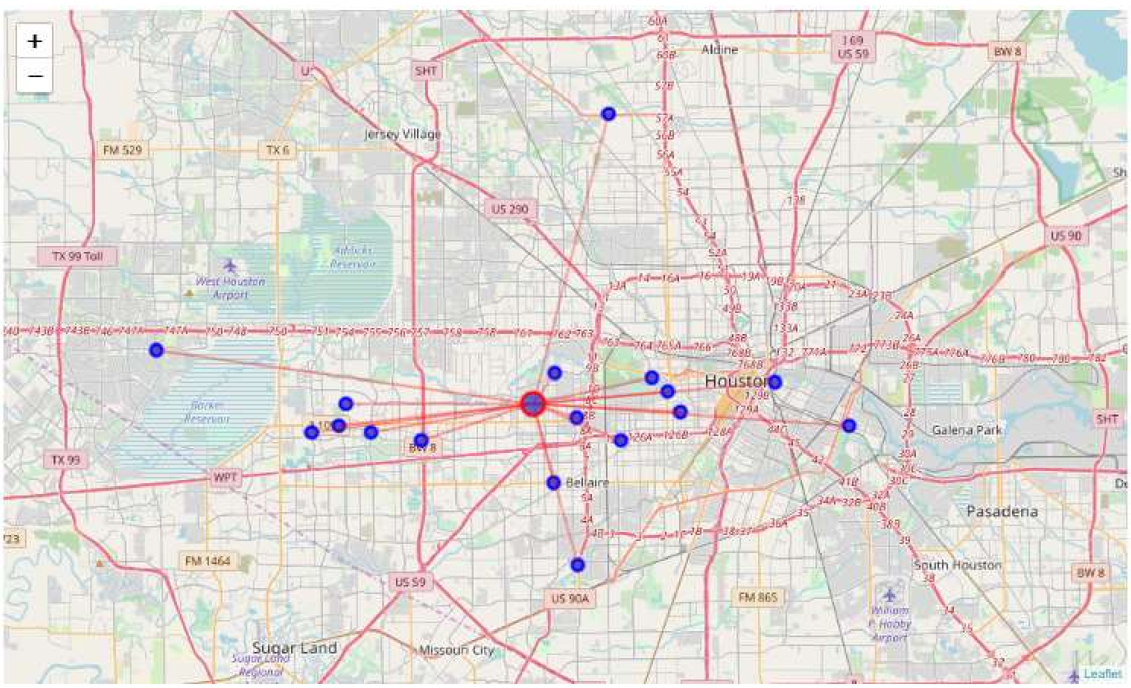
Los Angeles:



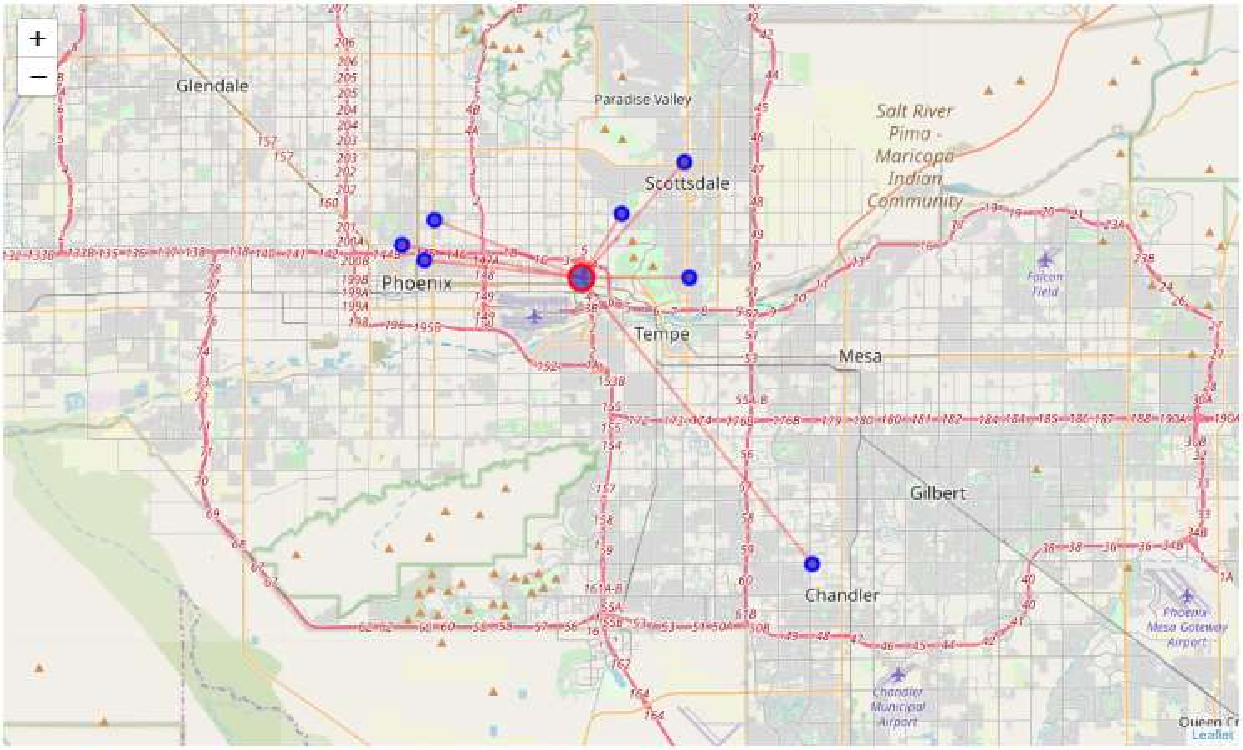
Chicago:



Houston:



Phoenix:



Glendale
Paradise Valley
Salt River
Pima -
Maricopa
Indian
Community
Scottsdale
Phoenix
Tempe
Mesa
Gilbert
Chandler
Leaflet

It was determined that New York city has the lowest mean distance (0.034) compared to the other cities, and therefore the highest restaurant density among all cities.

Another advantage New York has is the total amount of Peruvian restaurants (44), which is the highest and combined the lowest mean distance, makes it the best option for goods delivery, as restaurants are closer together, maximizing delivery routes and delivery times; and increasing potential sales, therefore this is the option recommended to the customer.

```
"New York, NY"
Mean Distance from Mean coordinates
0.03403361834108446
Los Angeles, CA
Mean Distance from Mean coordinates
0.1333208138726697
Chicago, IL
Mean Distance from Mean coordinates
0.03921846798477392
Houston, TX
Mean Distance from Mean coordinates
0.10194325397112565
Phoenix, AZ
Mean Distance from Mean coordinates
0.08928591991231158
```

Discussion

Even though customer request did not mention the number of restaurants per city as a factor to decide the best option, it is a very important factor.

By analyzing the best (lowest) mean distances, it was shown that Chicago was the second-best option; however the total amount of restaurants (170) was just half of Los Angeles amount (34), and you might think that the most restaurants you have available to deliver your goods, the more chance of getting more profits; therefore this outcome has to be mentioned to the customer, so he can take into consideration and make a more educated decision.

Data also showed one city (Phoenix) could be ruled out right away, as it showed considerable bigger mean distances and its restaurant numbers were very low (Just 7).

Finally, considering the highest populated cities does not mean they have the biggest amount of (Peruvian restaurants) in this case, and that is something we -as future data scientists- should take into consideration, to understand the business needs and anticipate any variables to deliver a better result than just what the customer requested.

Conclusion

There are many real-life situations where data along with technology can be used to find solutions to most problems; either by just analysis or by training existing data to predict future outcomes.

Having the right tools and the knowledge can make a big difference for the future of a business; for instance, the fictional problem developed in this project got great insights from using Foursquare API and data science tools to determine the best options, therefore decision could be made based in facts, and even though, they were just a small sample of all the variables to take in consideration, they showed how important data is.

This project was just a small taste of what can be accomplished by implementing data science to the decision-making process.

Appendix

Notebook links:

Github

https://github.com/ReinierAraya/Capstone_Project/blob/master/Capstone_The%20Battle%20of%20Neighborhoods%20(Week%202).ipynb

IBM Watson Studio

https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/395137d0-f8fb-4952-b36e-68a3ab8b8c81/view?access_token=842b4fa485f56fc6d9c62601e7d6bdc33bfd7c7e896175ebadeca3f7814b084d

End