Capstone Project - The Battle of Neighborhoods

Yasser Bigdeli San Francisco, Ca

Location recommendation for a new vegetarian restaurant opening in San Francisco, California

1. Business Problem and Background:

Restaurants are one of the most important business in the world. In the San Francisco bay area , it is even more important to have a right restaurant in right place to have a successful business, because the cost in the San Francisco is very high for a new business like restaurant. The business plan is to open a new vegetarian restaurant in the City of San Francisco, California and the following are the very important questions to be answered properly before starting the business:

- The locations of vegetarian restaurants in san Francisco?
- Ratio of Vegetarian places to total number of restaurants?
- Where is the most dense area for restaurants?
- Where is the most dense neighborhood for vegetarian restaurants?
- What are the rating of the Vegetarian restaurants?
- What is the percentages of vegetarian population in the area? (if possible)

2. Data Source and Description:

San Francisco neighborhood data will be obtained via web scraping. The geographical longitude and latitudes of each neighborhoods will be obtained using ‘geopy.geocoders’ python library. The location data will be obtained for ‘foursquare.com’ data base and will be based on the geographical coordinates. The registered credentials of foursquare API will be used for obtaining the venues, user, rating, tips, etc. data. This section will be done through the following steps:

1. Find the neighborhoods
2. Obtain the geographical coordinates
3. Get location data from foursquare API
4. Clean the data and create data frame
5. Explore the neighborhoods
6. Analyze each neighborhood
7. Make a data frame for vegetarian restaurant
8. Group by neighborhood
9. Clustering Neighborhoods using K-means clustering
10. Visualizing the resulting clusters
11. Making recommendation for a vegetarian restaurant location

Importing Libraries

In [4]:
```
!pip install geopy
```

```
Collecting geopy
  Using cached https://files.pythonhosted.org/packages/07/e1/9c72de674d5c2b8f
cb0738a5ceeb5424941fefa080bfe4e240d0bacb5a38/geopy-2.0.0-py3-none-any.whl
Collecting geographiclib<2,>=1.49 (from geopy)
  Using cached https://files.pythonhosted.org/packages/8b/62/26ec95a98ba64299
163199e95ad1b0e34ad3f4e176e221c40245f211e425/geographiclib-1.50-py3-none-any.
whl
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-1.50 geopy-2.0.0
```

In [6]:
```
import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
from geopy.geocoders import Nominatim
from pandas.io.json import json_normalize
from sklearn.cluster import KMeans
import folium
```

List of San Farancisco neighborhood: source: https://localwiki.org/sf/Neighborhoods (https://localwiki.org/sf/Neighborhoods)

In [30]:
```
neigh='''Alamo Square
Bayview
Bernal Heights
Buena Vista
Butcher Town
Chinatown
Civic Center
Cole Valley
Cow Hollow
Crocker Amazon
Diamond Heights
Dogpatch
Duboce Triangle
Fillmore District
Financial District
Folsom
Forest Hill
Glen Park
Golden Gate Heights
Hayes Valley
Hunters Point
India Basin
Ingleside
Jackson Square
Japantown
Laurel Heights
Lower Haight
Miraloma Park
Mission Bay
Nob Hill
Noe Valley
NOPA
North Beach
Ocean View
Pacific Heights
Park Merced
Parkside
Parnassus Heights
Polk
Portola
Potrero Hill
Rincon Hills
Russian Hill
Saint Francis Wood
SoMa
South Beach
Stonestown
Telegraph Hill
Theater District
The Castro
The Dolores Valley
The Embarcadero
The Excelsior
The Marina
The Mission
The Presidio
```

```
The Sunset
The Tenderloin
Twin Peaks
Upper Market
Visitacion Valley
Western Addition
West Portal'''
```

Data Cleaning

In [31]:
```
sf_neigh=pd.DataFrame(neigh.split('\n'), columns=['Neighborhood'])
```

In [32]:
```
sf_neigh.tail()
```

Out[32]:

	Neighborhood
58	Twin Peaks
59	Upper Market
60	Visitacion Valley
61	Western Addition
62	West Portal

In [33]:
```
sf_neigh['latitude']=np.nan
sf_neigh['longitude']=np.nan
# sf_neigh.head()
```

San Francisco Geograpgical coordinate

In [34]:
```
address = 'San Francisco City, CA, USA'

geolocator = Nominatim(user_agent="ca_explorer")
location = geolocator.geocode(address)
latitude_sf = location.latitude
longitude_sf = location.longitude
print('The geograpical coordinate of San Francisco are {}, {}.'.format(latitud
e_sf, longitude_sf))
```

```
The geograpical coordinate of San Francisco are 37.7790262, -122.4199061.
```

Adding Neighborhood geographical coordinates to SF_neigh data frame

In [35]:
```
for i , neighbor in enumerate(sf_neigh['Neighborhood']):

    address = neighbor + ', ' + 'San Francisco City, CA, USA'
    try:
        geolocator = Nominatim(user_agent="California_explorer")
        location = geolocator.geocode(address, timeout=10000)
        latitude = location.latitude
        longitude = location.longitude
        sf_neigh.loc[i,'latitude']=latitude
        sf_neigh.loc[i,'longitude']=longitude
#         print(latitude, longitude)
    except:
        pass

sf_neigh.dropna(inplace=True)
```

In [36]:
```
sf_neigh.head(10)
```

Out[36]:

	Neighborhood	latitude	longitude
0	Alamo Square	37.776360	-122.434689
1	Bayview	40.772627	-124.183950
2	Bernal Heights	37.741001	-122.414214
3	Buena Vista	37.806532	-122.420648
5	Chinatown	37.794301	-122.406376
6	Civic Center	37.779026	-122.419906
7	Cole Valley	37.765813	-122.449962
8	Cow Hollow	37.797262	-122.436248
9	Crocker Amazon	37.709378	-122.438587
11	Dogpatch	37.760698	-122.389202

In [37]:
```
sf_neigh.shape
```

Out[37]:
```
(51, 3)
```

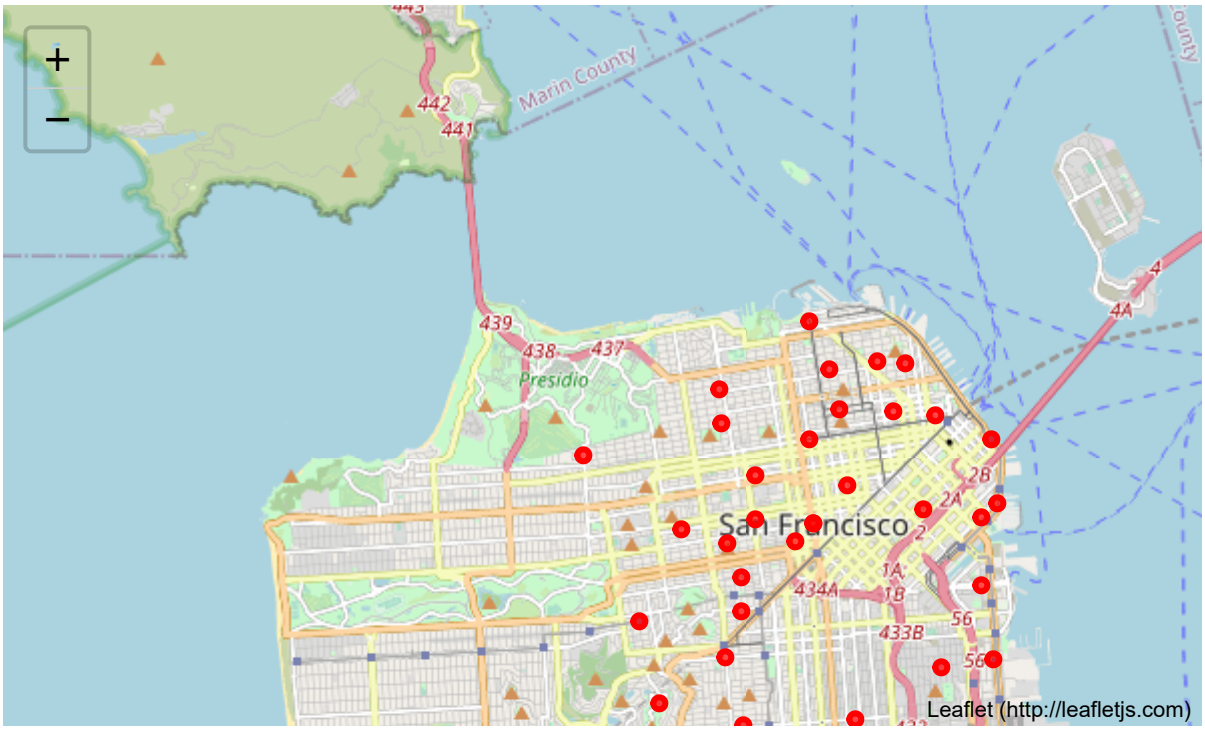
Create SF map and neighbors

In [38]:
```
# create map of SF using latitude and longitude values
map_SF = folium.Map(location=[latitude_sf, longitude_sf], zoom_start=12)

# add markers to map
for lat, lng, neighborhood in zip(sf_neigh['latitude'], sf_neigh['longitude'],
sf_neigh['Neighborhood']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=3,
        popup=label,
        color='red',
        fill=True,
#         fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_SF)

map_SF
```

Out[38]:



In [ ]:

In [ ]: