The Battle of Neighborhoods

Predicting the Area to Open a French Restaurant in San Francisco

```
In [1]: import pandas as pd
        import numpy as np  # useful for many scientific computing in Python
```

In [2]:

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

print('Folium installed and imported!')
```

```
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
    altair-3.2.0               |           py36_0         770 KB  conda-forge
    ca-certificates-2019.6.16  |       hecc5488_0         145 KB  conda-forge
    branca-0.3.1               |             py_0          25 KB  conda-forge
    certifi-2019.6.16          |           py36_1         149 KB  conda-forge
    openssl-1.1.1c             |       h516909a_0         2.1 MB  conda-forge
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         3.3 MB

The following NEW packages will be INSTALLED:

    altair:          3.2.0-py36_0      conda-forge
    branca:          0.3.1-py_0        conda-forge
    folium:          0.5.0-py_0        conda-forge
    vincent:         0.4.4-py_1        conda-forge

The following packages will be UPDATED:

    ca-certificates: 2019.5.15-0                   --> 2019.6.16-hecc5488_0 c
onda-forge
    certifi:         2019.6.16-py36_1              --> 2019.6.16-py36_1     c
onda-forge

The following packages will be DOWNGRADED:

    openssl:         1.1.1c-h7b6447c_1             --> 1.1.1c-h516909a_0    c
onda-forge


Downloading and Extracting Packages
folium-0.5.0         | 45 KB     | ##################################### | 10
0%
altair-3.2.0         | 770 KB    | ##################################### | 10
0%
ca-certificates-2019 | 145 KB    | ##################################### | 10
0%
branca-0.3.1         | 25 KB     | ##################################### | 10
0%
certifi-2019.6.16    | 149 KB    | ##################################### | 10
0%
openssl-1.1.1c       | 2.1 MB    | ##################################### | 10
0%
```

```
vincent-0.4.4        | 28 KB     | ##################################### | 10
0%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Folium installed and imported!
```

In [3]:

```
df_incidents = pd.read_csv('https://s3-api.us-geo.objectstorage.softlayer.net/
cf-courses-data/CognitiveClass/DV0101EN/labs/Data_Files/Police_Department_Inci
dents_-_Previous_Year__2016_.csv')

print('Dataset downloaded and read into a pandas dataframe!')
```

```
Dataset downloaded and read into a pandas dataframe!
```

In [4]:

```
df_incidents.head()
```

Out[4]:

	IncidntNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolut
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	ARRE BOOK
1	120058272	WEAPON LAWS	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	ARRE BOOK
2	141059263	WARRANTS	WARRANT ARREST	Monday	04/25/2016 12:00:00 AM	14:59	BAYVIEW	ARRE BOOK
3	160013662	NON-CRIMINAL	LOST PROPERTY	Tuesday	01/05/2016 12:00:00 AM	23:50	TENDERLOIN	NO
4	160002740	NON-CRIMINAL	LOST PROPERTY	Friday	01/01/2016 12:00:00 AM	00:30	MISSION	NO

In [4]:

```
df_incidents.shape
```

Out[4]:

```
(150500, 13)
```

So each row consists of 13 features:

> 1. **IncidntNum**: Incident Number
> 2. **Category**: Category of crime or incident
> 3. **Descript**: Description of the crime or incident
> 4. **DayOfWeek**: The day of week on which the incident occurred
> 5. **Date**: The Date on which the incident occurred
> 6. **Time**: The time of day on which the incident occurred
> 7. **PdDistrict**: The police department district
> 8. **Resolution**: The resolution of the crime in terms whether the perpetrator was arrested or not
> 9. **Address**: The closest address to where the incident took place
> 10. **X**: The longitude value of the crime location
> 11. **Y**: The latitude value of the crime location
> 12. **Location**: A tuple of the latitude and the longitude values
> 13. **PdId**: The police department ID

In [5]:

```
# get the first 100 crimes in the df_incidents dataframe
limit = 1000
df_incidents = df_incidents.iloc[0:limit, :]
```

In [7]:

```
df_incidents.shape
```

Out[7]:

```
(1000, 13)
```

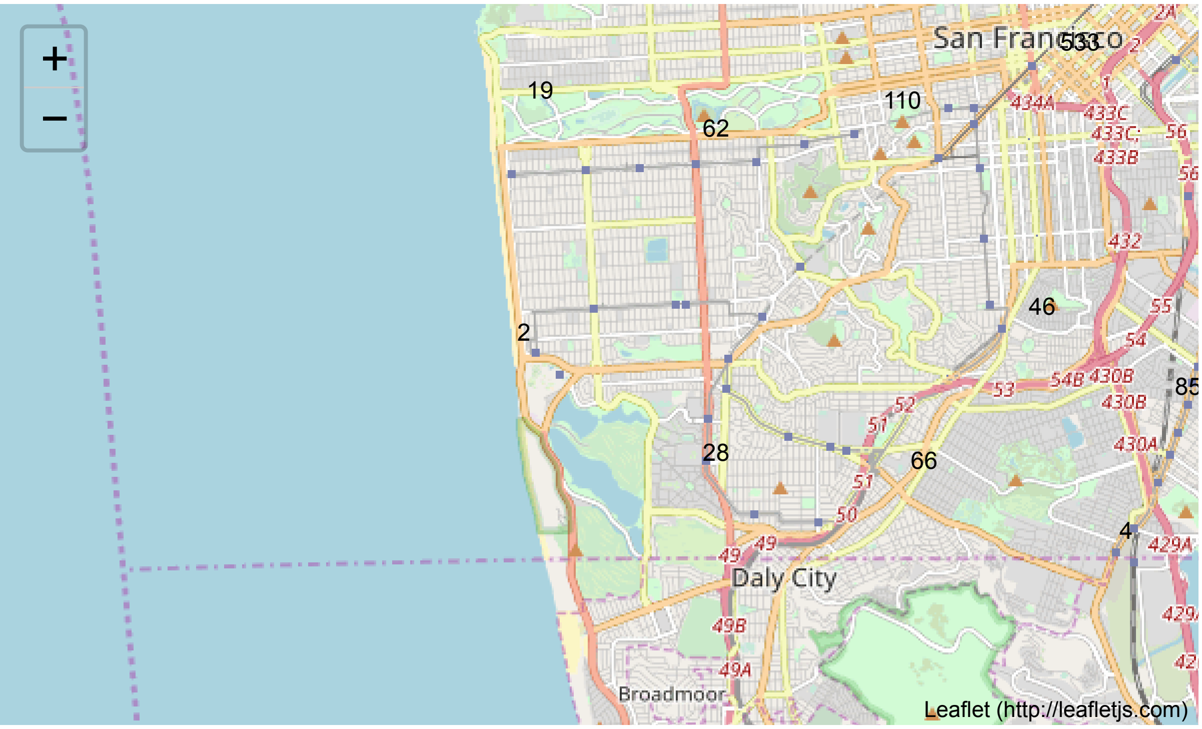
In [8]:

```
# San Francisco latitude and longitude values
latitude = 37.77
longitude = -122.42
```

In [9]:

```
from folium import plugins

# let's start again with a clean copy of the map of San Francisco
sanfran_map = folium.Map(location = [latitude, longitude], zoom_start = 12)

# instantiate a mark cluster object for the incidents in the dataframe
incidents = plugins.MarkerCluster().add_to(sanfran_map)

# loop through the dataframe and add each data point to the mark cluster
for lat, lng, label, in zip(df_incidents.Y, df_incidents.X, df_incidents.Locat
ion):
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(incidents)

# display map
sanfran_map
```

Out[9]:



Based on the map: There are two locations with very low criminal incidents:

2 incidents: (37.7388739744874, -122.50648456409)

4 incidents: (37.7119353727691, -122.405634443479)

Right now, we need to find number of French restaurants inside these two areas.

In [10]:

```python
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation

!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library

print('Folium installed')
print('Libraries imported.')
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - geopy


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    geopy-1.20.0               |             py_0          57 KB  conda-forge
    geographiclib-1.49         |             py_0          32 KB  conda-forge
    ------------------------------------------------------------
                                           Total:          90 KB

The following NEW packages will be INSTALLED:

    geographiclib: 1.49-py_0   conda-forge
    geopy:         1.20.0-py_0 conda-forge


Downloading and Extracting Packages
geopy-1.20.0         | 57 KB     | ##################################### | 10
0%
geographiclib-1.49   | 32 KB     | ##################################### | 10
0%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done

# All requested packages already installed.

Folium installed
Libraries imported.
```

In [11]:

```
CLIENT_ID = 'your Foursquare ID' # your Foursquare ID
CLIENT_SECRET = 'your Foursquare Secret' # your Foursquare Secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: 1ZONQHIMFSGFHLO5PF5XE0FWDVAUG3GLONGIQQJJR4UHT5MR
CLIENT_SECRET:P3LETF0FZNE4DJYIQGZNKSDD0MBU00LXIUN0I4THDHVQLI23
```

In [12]:

```
latitude = 37.7388739744874
longitude = -122.50648456409
```

In [13]:

```
search_query = 'French'
radius = 3000
print(search_query + ' .... OK!')
```

```
French .... OK!
```

In [14]:

```
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret=
{}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET,
latitude, longitude, VERSION, search_query, radius, LIMIT)
url
```

Out[14]:

```
'https://api.foursquare.com/v2/venues/search?client_id=1ZONQHIMFSGFHLO5PF5XE0
FWDVAUG3GLONGIQQJJR4UHT5MR&client_secret=P3LETF0FZNE4DJYIQGZNKSDD0MBU00LXIUN0
I4THDHVQLI23&ll=37.7388739744874,-122.50648456409&v=20180604&query=French&rad
ius=3000&limit=30'
```

In [16]:

```
results = requests.get(url).json()
results
```

```
Out[16]: {'meta': {'code': 200, 'requestId': '5d64d2d4e0c0c9002cd45d2e'},
          'response': {'venues': [{'id': '4a9ec3ecf964a520163b20e3',
             'name': 'French Bakery',
             'location': {'address': '1101 Taraval St',
              'lat': 37.74290296507471,
              'lng': -122.47802176889526,
              'labeledLatLngs': [{'label': 'display',
                'lat': 37.74290296507471,
                'lng': -122.47802176889526}],
              'distance': 2545,
              'postalCode': '94116',
              'cc': 'US',
              'city': 'San Francisco',
              'state': 'CA',
              'country': 'United States',
              'formattedAddress': ['1101 Taraval St',
               'San Francisco, CA 94116',
               'United States']},
             'categories': [{'id': '4bf58dd8d48988d16a941735',
               'name': 'Bakery',
               'pluralName': 'Bakeries',
               'shortName': 'Bakery',
               'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/bakery
         _',
                'suffix': '.png'},
               'primary': True}],
             'referralId': 'v-1566888660',
             'hasPerk': False},
            {'id': '4bb556e02f70c9b67c238430',
             'name': 'La Pelouse French School',
             'location': {'lat': 37.751352,
              'lng': -122.47601876666667,
              'labeledLatLngs': [{'label': 'display',
                'lat': 37.751352,
                'lng': -122.47601876666667}],
              'distance': 3020,
              'postalCode': '94116',
              'cc': 'US',
              'city': 'San Francisco',
              'state': 'CA',
              'country': 'United States',
              'formattedAddress': ['San Francisco, CA 94116', 'United States']},
             'categories': [{'id': '4bf58dd8d48988d13d941735',
               'name': 'High School',
               'pluralName': 'High Schools',
               'shortName': 'High School',
               'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/building/sch
         ool_',
                'suffix': '.png'},
               'primary': True}],
             'referralId': 'v-1566888660',
             'hasPerk': False},
            {'id': '4bca562c3740b713a0aa6065',
             'name': 'Mercury French Cleaners',
             'location': {'address': '4150 Irving St',
              'lat': 37.76252197548304,
              'lng': -122.50055003064311,
     'labeledLatLngs': [{'label': 'display',
       'lat': 37.76252197548304,
       'lng': -122.50055003064311}],
     'distance': 2683,
     'postalCode': '94122',
     'cc': 'US',
     'city': 'San Francisco',
     'state': 'CA',
     'country': 'United States',
     'formattedAddress': ['4150 Irving St',
      'San Francisco, CA 94122',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d1fc941735',
      'name': 'Laundry Service',
      'pluralName': 'Laundry Services',
      'shortName': 'Laundry',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/laundr
y_',
       'suffix': '.png'},
      'primary': True}],
    'referralId': 'v-1566888660',
    'hasPerk': False},
   {'id': '4c49322193249c747f0fef4c',
    'name': "ANNABELLE'S FRENCH CLEANERS",
    'location': {'address': '363 West Portal Ave',
     'crossStreet': 'btw 14th Ave & 15th Ave',
     'lat': 37.737266979170215,
     'lng': -122.46969766665704,
     'labeledLatLngs': [{'label': 'display',
       'lat': 37.737266979170215,
       'lng': -122.46969766665704}],
     'distance': 3243,
     'postalCode': '94127',
     'cc': 'US',
     'city': 'San Francisco',
     'state': 'CA',
     'country': 'United States',
     'formattedAddress': ['363 West Portal Ave (btw 14th Ave & 15th Ave)',
      'San Francisco, CA 94127',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d1fc941735',
      'name': 'Laundry Service',
      'pluralName': 'Laundry Services',
      'shortName': 'Laundry',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/laundr
y_',
       'suffix': '.png'},
      'primary': True}],
    'referralId': 'v-1566888660',
    'hasPerk': False},
   {'id': '4b803111f964a520e95a30e3',
    'name': 'Lakeside French Cleaners',
    'location': {'address': '2660 Ocean Ave',
     'lat': 37.732421821993846,
     'lng': -122.47479429669892,
     'labeledLatLngs': [{'label': 'display',
       'lat': 37.732421821993846,
       'lng': -122.47479429669892}],
     'distance': 2880,
     'postalCode': '94132',
     'cc': 'US',
     'city': 'San Francisco',
     'state': 'CA',
     'country': 'United States',
     'formattedAddress': ['2660 Ocean Ave',
      'San Francisco, CA 94132',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d1fc941735',
      'name': 'Laundry Service',
      'pluralName': 'Laundry Services',
      'shortName': 'Laundry',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/laundr
y_',
       'suffix': '.png'},
      'primary': True}],
    'referralId': 'v-1566888660',
    'hasPerk': False}]}}
```

2 incidents: Location(37.7388739744874, -122.50648456409)

In [17]:

```
# assign relevant part of JSON to venues
venues = results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.shape
```

Out[17]:

```
(5, 17)
```

4 incidents: Location(37.7119353727691, -122.405634443479)

In [18]:

```
latitude = 37.7119353727691
longitude = -122.405634443479


search_query = 'French'
radius = 3000

url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
results = requests.get(url).json()

# assign relevant part of JSON to venues
venues = results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.shape
```

Out[18]:

```
(3, 16)
```

In [ ]:

```
```