Battle of the Neighbourhoods

City of San Francisco

Elvis Tony   *Link to Python Notebook - (Here)*

Problem Selected:

Choosing suitable locations for building Apartments

Introduction

This is a project for the course
IBM Professional Certificate   (Coursera)

The Project aims to Help choose the best Location to begin an Apartments Project.

Below are some of the Problems we need to answer;

- Where is the best spot to start an Apartments Project ?

- What are the factors on which customers choose Apartments?

- What are the proximities of the basic Necessities available?

Business Aspect

Using results of this project I hope to provide a set of suitable locations for building an Apartment Complex that will Hypothetically be successful based on the above factors

We choose a popular city "San Francisco" for simulating the problem and testing the solution.
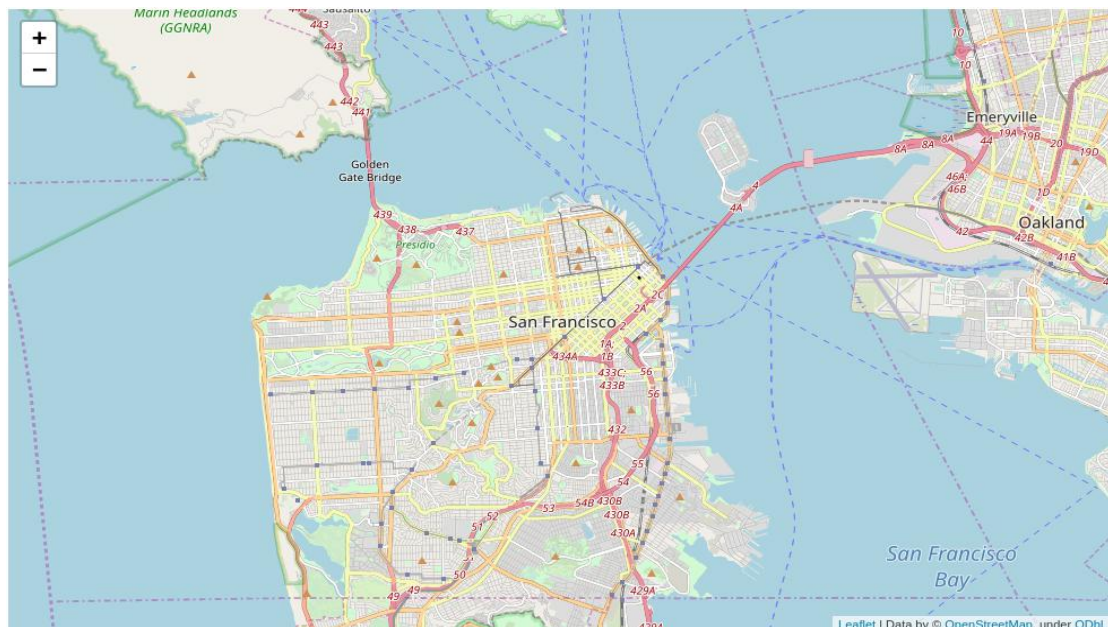
Introduction

Factors on Choosing an Apartment

It is almost impossible to accurately predict which apartment, a customer will select, but we can use a set of dependant factors on which that decision will be made.

We classify the factors to two categories

Building Factors:
> (Ignoring, these are left to the builder)

- Well Built and Sturdy.
- Electricity and Plumbing.
- Parking Space.
- Rent

Location Factors:
> (We'll be focusing on this Section)

- Nearest Hospital (distance)
- Nearest School/College(distance)
- Lifestyle / Gym Nearby
- Restaurants and Eateries

Data Source and Overview

We will be using Foursquare API as a reliable source of Location Data of the City,
San Francisco.

all rights to logo are
 reserved to FOURSQUARE.COM

We aim to make a suitable Scatter Plot of the Nearby Facilities and hence choose an appropriate location for Building the Apartment.

And Hence using points located using the API we will run the K-Means Algorithm to find the suitable locations for the problem.

Data Introduction

We first plan to map the points of all the
Hospitals,
Restaurants,
Gyms and
Schools
onto a map of San Francisco.

We will then convert those points to form a scatter plot (hopefully) marking each type of facility in a different colour.
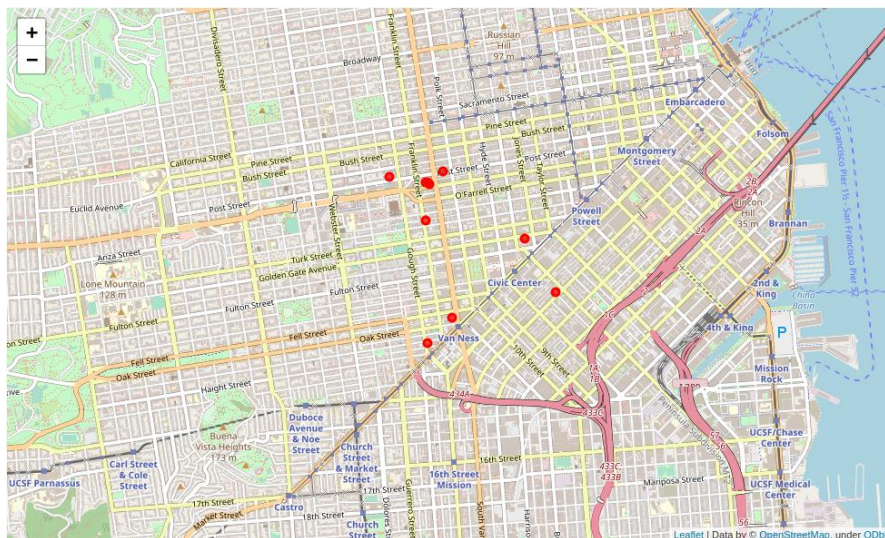
We will then perform   k-means algorithm on the Data and find Suitable Centroids that will hold as Locatons for building the apartment
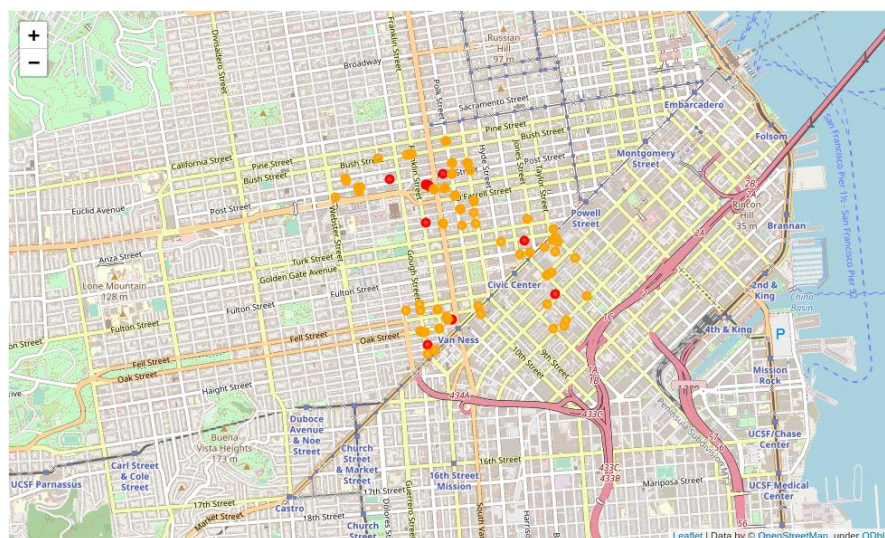
Initial Point Selection

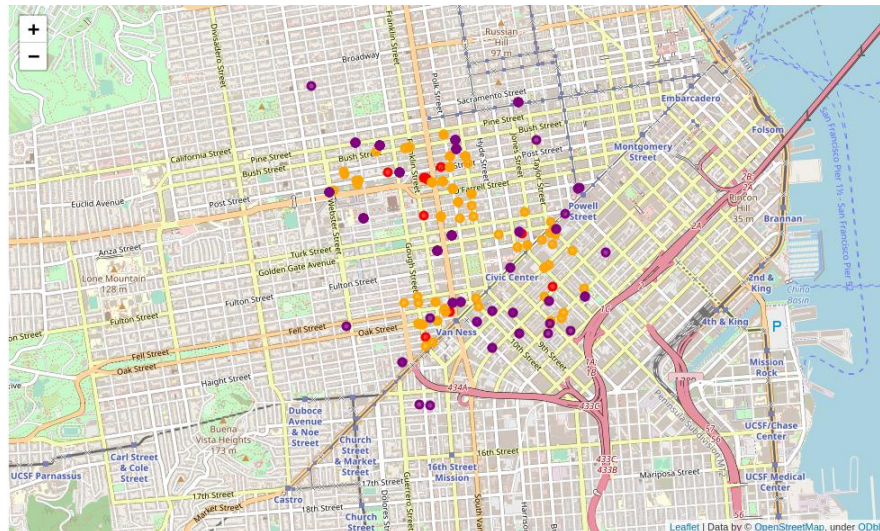Placing Hospitals as a mandatory necessity, we Began by marking the Hospitals on the Map.

Mapping of Nearby Eateries

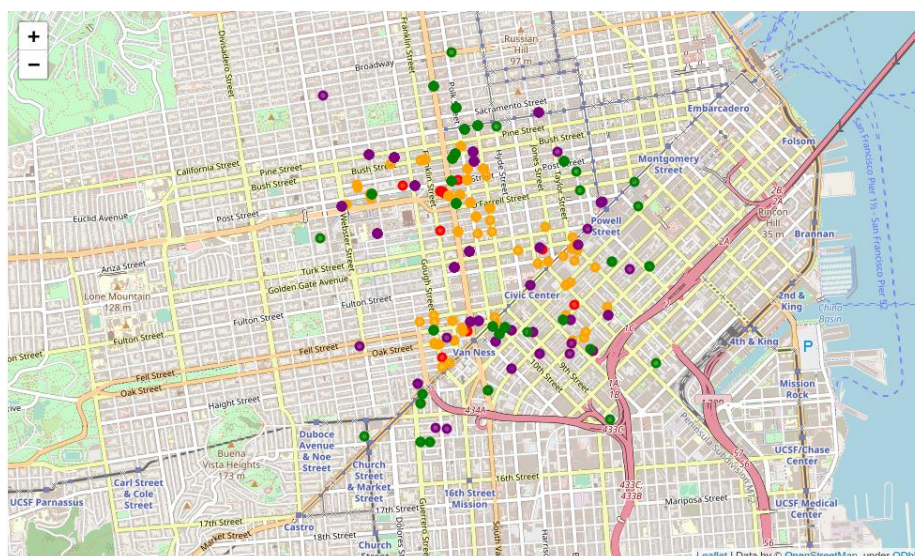We Find the nearby Eateries using Explore of Foursquare maps API.

Mapping of Nearby Schools

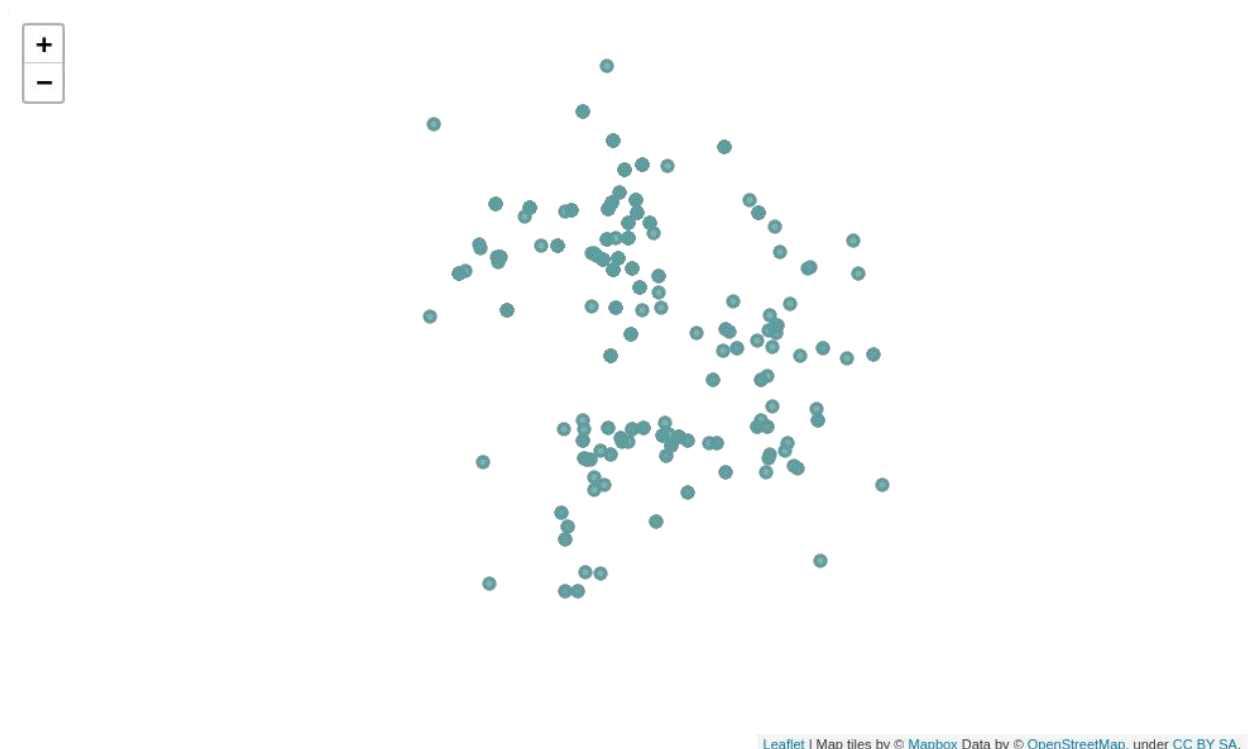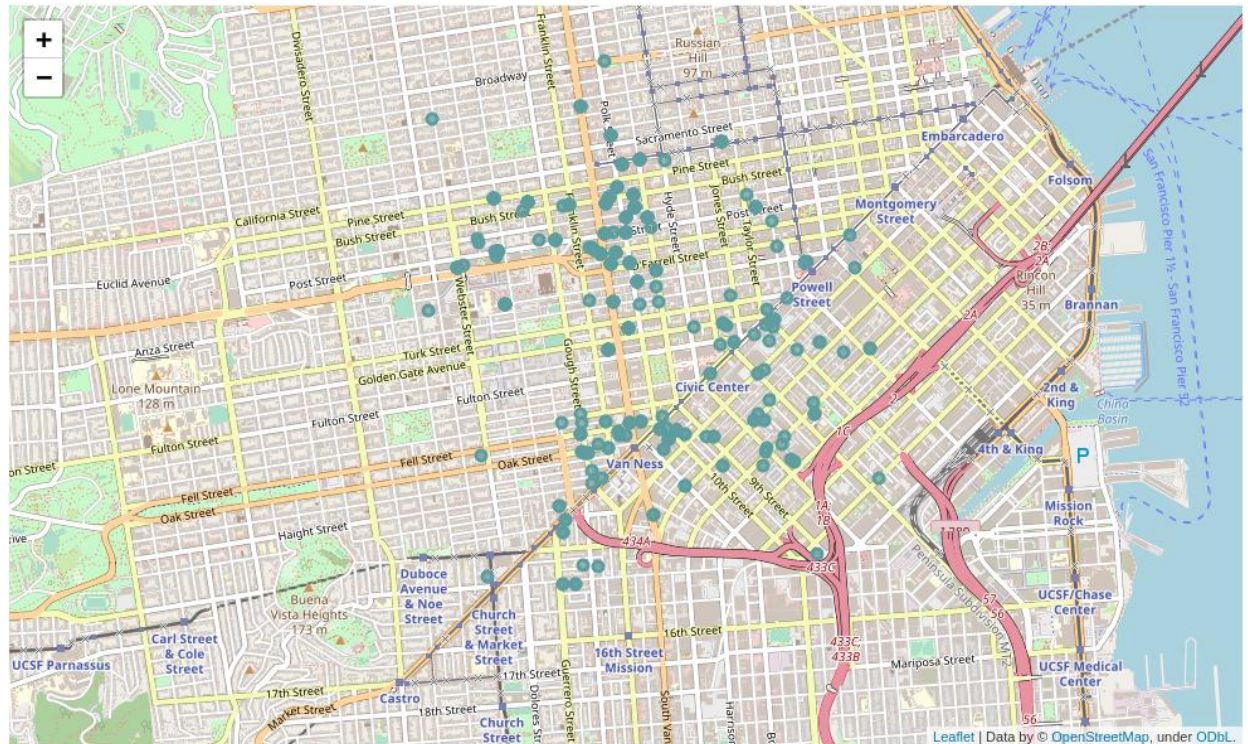Similarly,We Find the nearby Schools using Explore of Foursquare maps API

Mapping of Nearby Gyms

Lastly, We Find the nearby Gyms using Explore of Foursquare maps API.

Complete Map of Facilities

Prepping Data
for K-Means Algorithm

We prepare the data for performing K-Means Clustering.

Since the Data we have is Lattitude and Longitude, we take the following steps to convert it to a simpler form

    - Subtract all values by mean of the dataset
    - Multiply all values by 1000 to make the value Easier to plot
    - store the Mean of the dataset into 2 variables for later resubsitution

Performing K-Means Clustering

K-Means Clustering

We aim to make a suitable number of clusters and Ensure a proper output by this algorithm

```
Clusters = 4 or above
Source Data = Lattitude, Longitude
```

```
In [15]: k_means = KMeans(init="k-means++", n_clusters=4, n_init=12,max_iter=10)
         k_means
```

```
Out[15]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=10,
             n_clusters=4, n_init=12, n_jobs=None, precompute_distances='auto',
             random_state=None, tol=0.0001, verbose=0)
```

```
In [16]: X=backdf[['lat','lng']].to_numpy()
         k_means.fit(X)
```

```
Out[16]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=10,
             n_clusters=4, n_init=12, n_jobs=None, precompute_distances='auto',
             random_state=None, tol=0.0001, verbose=0)
```

```
In [17]: k_means_labels = k_means.labels_
         k_means_cluster_centers = k_means.cluster_centers_
```
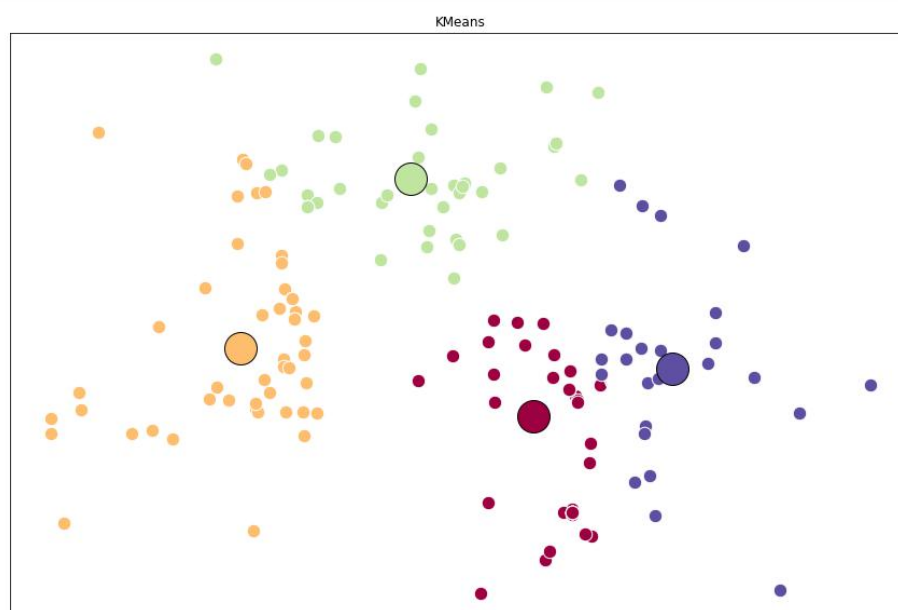
Final Result

Gathering results from Cluster, We now locate these points on the Map and Label them as Potential Locations

We recall, that we modified the dataset to suit the algorithm.

To revert back to the original form we perform the following steps,
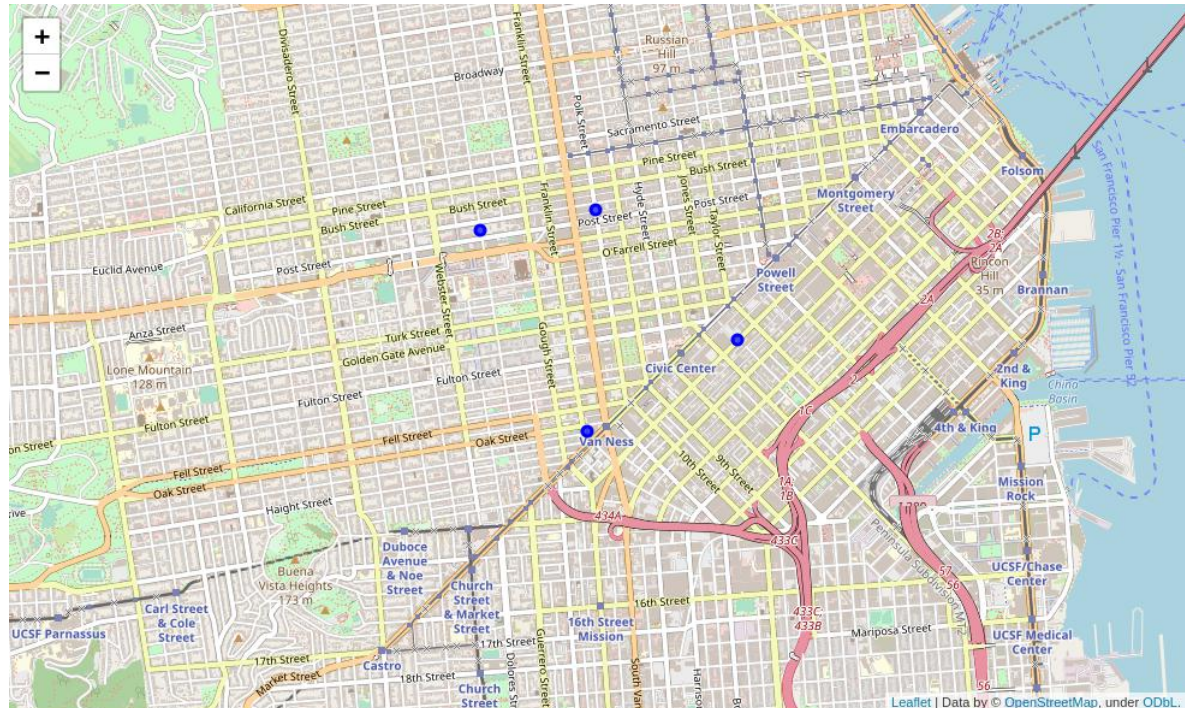
- Each value is divided by 1000
- Each value is added with the previously stored Mean of Lattitude and Longitude

We the following calculated points on the Map

```
lat              lng
37.780095        -122.409962
37.786258        -122.428280
37.774954        -122.420631
37.787381        -122.420065
```

Locating the Points on the Map

For an example we take a point and Display its nearby Facilities

Nearby Facilities and their Distances

	name	categories	lat	lng	distances_metres
0	California Skin Institute - San Francisco - No...	Hospital	37.789348	-122.417545	140.155565
1	Saint Francis Hospital Outpatient Registration	Hospital	37.789608	-122.416921	138.452522
2	SFMH Radiology	Hospital	37.789637	-122.416571	157.433378
0	Soko Gakuen Japanese Language School	Language School	37.788453	-122.426564	795.872803
1	Grace Cathedral	Church	37.791666	-122.413348	419.168993
2	Sips n Sews	School	37.788127	-122.419243	296.135904
0	Mymy Coffee Shop	Diner	37.790833	-122.419118	109.606637
1	Crostini And Java	Café	37.789111	-122.417099	177.781797
2	Swan Oyster Depot	Seafood Restaurant	37.790931	-122.420759	253.080523
0	Uforia Studios	Gym / Fitness Center	37.790441	-122.420148	196.878076
1	Core 40	Gym / Fitness Center	37.790700	-122.418967	93.578391
2	Krav Maga San Francisco	Gym / Fitness Center	37.788365	-122.421242	381.844228

Conclusion

I now conclude my Project, This project has shown how the Foursquare API
is used to solve Problems based on Locations and Topology.

The Project presented 4 viable locations for building an Apartment Complex by satisfying the criteria previously set in the Introduction.

Below are the Websites that helped me along the way,

https://https://www.coursera.org/

https://stackoverflow.com/

https://pandas.pydata.org/

https://deparkes.co.uk/2016/06/10/folium-map-tiles/

https://labs.cognitiveclass.ai/