# Battle of the Neighborhoods (Week 2) - An Analysis of Fitness in San Francisco Final Report

## 1 Introduction:

According to a poll conducted by US News in 2019, San Francisco was considered the 4th fittest city in the United States. With the new trend of healthy living and fitness it has become now more than ever a good time to invest in fitness centers. Also, with the accessibility to public transportation it has become easier to access different parts of the city.

As a new fitness center startup in the San Francisco area, we need to take an analysis of the city's neighborhoods to find the most opportunistic area to set up a new fitness center. The aim of this proposal is to analyze the neighborhoods by leveraging Four Square's API to see which areas have the right conditions to create a chain of fitness centers.

This will help to guide our company into making key strategic business decisions.

## 2 Business Problem:

The purpose of this proposal is to find the most optimal location to open a new fitness center. To do this, we need to answer to following key questions:

1. Using FourSquare API, can we get a visual map of different locations with the nearest venues?
2. From those venues, how many gyms or fitness centers are the most common in those neighborhoods?
3. Can we conduct an analysis where we can isolate specific neighborhoods that can be targeted (i.e., Clustering)?

## 3 Data:

To complete this analysis, we need to collect data regarding San Francisco's different neighborhoods. We also need to collect data on the different venues surrounding the neighborhoods.

Here is the following data being used to conduct this analysis:

1. For San Francisco postal code and Neighborhood data we will be using the following dataset: Data Source: http://www.healthysf.org/bdi/outcomes/zipmap.htm (http://www.healthysf.org/bdi/outcomes/zipmap.htm)
2. To get the latitude and longitude of the locations we will be using the pgeocode package.
3. We will use the Foursquare API to gather venue data for each San Francisco Neighborhood. It will help to isolate the nearest venues per location.

The following data points will be collected for this analysis:

1. Zip Code
2. Neighborhood
3. Neighborhood Latitude
4. Neighborhood Longitude
5. Name of Venue
6. Venue Latitude
7. Venue Longitude
8. Venue Category

# 4 Methodology:

## Data Collection and Cleaning

To begin our analysis, we first need to scrap the data from the following url: http://www.healthysf.org/bdi/outcomes/zipmap.htm (http://www.healthysf.org/bdi/outcomes/zipmap.htm). This url contains zip codes and Neighborhood data needed for this analysis.

To scrap this data, we need to import the requests and BeautifulSoup packages. From there we can load them into a pandas data frame and then clean up the data frame to only use zip code and neighborhood data.

In [90]:
```
# scrap the data from the url
url = "http://www.healthysf.org/bdi/outcomes/zipmap.htm"
san_fran_url = requests.get(url).text
soup = BeautifulSoup(san_fran_url,'lxml')
table = soup.find_all("table")

# move the data into a dataframe
san_fran_df = pd.read_html(str(table))

# clean the dataframe to fit
san_fran_df = pd.DataFrame(san_fran_df[4])
san_fran_df.columns = san_fran_df.iloc[0]
san_fran_df = san_fran_df.iloc[1:]
san_fran_df.drop(index = san_fran_df.index[21],axis = 0, inplace = True)
san_fran_df = san_fran_df.iloc[:,0:2]
san_fran_df.head()
```

Out[90]:

| | Zip Code | Neighborhood |
|---|---|---|
| 1 | 94102 | Hayes Valley/Tenderloin/North of Market |
| 2 | 94103 | South of Market |
| 3 | 94107 | Potrero Hill |
| 4 | 94108 | Chinatown |
| 5 | 94109 | Polk/Russian Hill (Nob Hill) |

Now that the data neighborhood data has been collected, we can begin collecting the latitude and longitude data from the pgeocode library. By using the Nominatim function, we can query the longitude and latitude data by using just the neighborhood zip code. The data frame will look like the following:

In [91]:
```
# Will use the pgeocode library to get the latitude and longitude coordinates for each neighborhood
nomi_object = pgeocode.Nominatim('us')
latitude = []
longitude = []

for index,row in san_fran_df.iterrows():
    zipcode = nomi_object.query_postal_code(row["Zip Code"])
    latitude.append(zipcode.latitude)
    longitude.append(zipcode.longitude)

san_fran_df["Latitude"] = latitude
san_fran_df["Longitude"] = longitude

san_fran_df.head()
```

Out[91]:

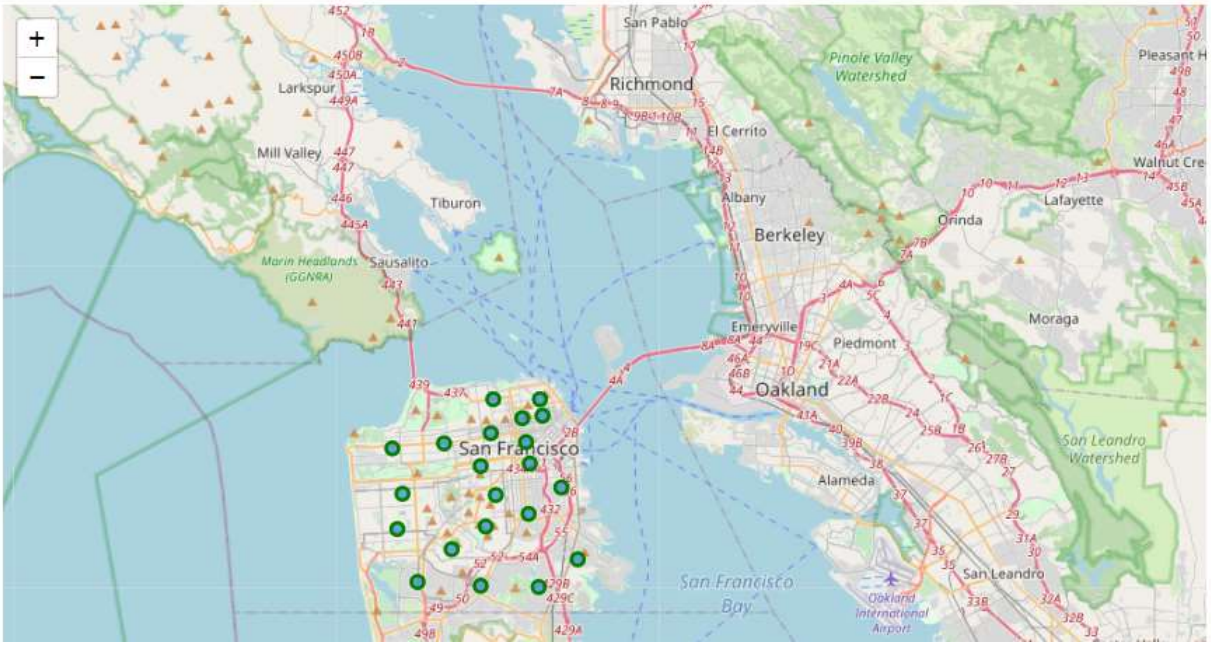
| | Zip Code | Neighborhood | Latitude | Longitude |
|---|---|---|---|---|
| 1 | 94102 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 |
| 2 | 94103 | South of Market | 37.7725 | -122.4147 |
| 3 | 94107 | Potrero Hill | 37.7621 | -122.3971 |
| 4 | 94108 | Chinatown | 37.7929 | -122.4079 |
| 5 | 94109 | Polk/Russian Hill (Nob Hill) | 37.7917 | -122.4186 |

## Visualizing the Data

Before we start adding additional venue data, let us take a visual map of San Francisco and display where each neighborhood is located. For this we will use the Folium library to build a visual map of San Francisco with the latitude and longitude data.

In [109]:
```
address = 'San Francisco, California'

geolocator = Nominatim(user_agent="san_francisco")
location = geolocator.geocode(address)
san_latitude = location.latitude
san_longitude = location.longitude

map_san_fran = folium.Map(location=[san_latitude,san_longitude], zoom_start = 10)

for lat,lng, neighborhood in zip(san_fran_df['Latitude'],san_fran_df['Longitude'],san_fran_df['Neighborhood']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label,parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius = 5,
        popup = label,
        color = 'green',
        fill = True,
        fill_color = '#3186cc',
        fill_opacity = 0.7,
        parse_html = False).add_to(map_san_fran)

map_san_fran
```

Out[109]:




## Foursquare API

The Foursquare API is a library that is used to get venue data from specified latitude and longitude data. In this case we will be using the san_fran_df data frame to create a new data frame with additional Venue data. We will be looking for the top 100 venues with a radius of 600 meters for each of the selected neighborhoods. This will create the nearby_venues data frame below:

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue Latitude | Venue Longitude | Venue Name | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 | 37.780178 | -122.416505 | Asian Art Museum | Art Museum |
| 1 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 | 37.782751 | -122.415656 | Ales Unlimited: Beer Basement | Beer Bar |
| 2 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 | 37.783084 | -122.417650 | Saigon Sandwich | Sandwich Place |
| 3 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 | 37.781266 | -122.416901 | Philz Coffee | Coffee Shop |
| 4 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 | 37.782896 | -122.418897 | Brenda's French Soul Food | Southern / Soul Food Restaurant |

## EDA

Before we conduct our cluster analysis, it is good idea to conduct some exploratory data analysis. This will ensure that any data cleaning or feature engineering will need to be done before we build our model. First, let us look at the number of venues by Neighborhood:

```
In [95]: # To get a count of the number of venues per neighborhood
         nearby_venues.groupby('Neighborhood')["Venue Name"].count()

Out[95]: Neighborhood
         Bayview-Hunters Point                     19
         Castro/Noe Valley                         75
         Chinatown                                 88
         Haight-Ashbury                            35
         Hayes Valley/Tenderloin/North of Market   67
         Ingelside-Excelsior/Crocker-Amazon        45
         Inner Mission/Bernal Heights              58
         Inner Richmond                            66
         Lake Merced                               14
         Marina                                   100
         North Beach/Chinatown                     72
         Outer Richmond                            34
         Parkside/Forest Hill                      38
         Polk/Russian Hill (Nob Hill)             100
         Potrero Hill                              33
         South of Market                          100
         St. Francis Wood/Miraloma/West Portal      5
         Sunset                                    31
         Twin Peaks-Glen Park                      16
         Visitacion Valley/Sunnydale                8
         Western Addition/Japantown               100
         Name: Venue Name, dtype: int64
```

By looking at the data, we need a way to convert the categorical data into a numeric value. To do this, we need to use one hot encoding by using the pandas get_dummies function. This will help to construct our K-means clustering model.

```
# This will encode the Venue Category column
san_fran_hot = pd.get_dummies(nearby_venues[['Venue Category']], prefix = "", prefix_sep="")

# Add zipcode and Neighborhood into the dataframe
san_fran_hot['Neighborhood'] = nearby_venues['Neighborhood']

# move zip code and Neighboorhood to the front of the data set
fixed_columns = list(san_fran_hot.columns[-1:]) + list(san_fran_hot.columns[:-1])
san_fran_hot = san_fran_hot[fixed_columns]

san_fran_hot
```

For the last step before we build our Kmeans clustering model, we need to build a data frame that takes top 10 most common venues within a neighborhood. To do this we need to first get the frequency of each of the occurrences per neighborhood.

In [98]:
```
# This will get the frequency of each of the occurances
san_fran_freq = san_fran_hot.groupby(['Neighborhood']).mean().reset_index()
san_fran_freq.head()
```

Out[98]:

| | Neighborhood | ATM | Accessories Store | Adult Boutique | African Restaurant | American Restaurant | Antique Shop | Arcade | Argentinian Restaurant | Art Gallery | ... | Video Game Store | Video Store | Vietnamese Restaurant |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Bayview-Hunters Point | 0.000000 | 0.000000 | 0.000000 | 0.052632 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.000000 |
| 1 | Castro/Noe Valley | 0.013333 | 0.013333 | 0.013333 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.013333 | ... | 0.0 | 0.0 | 0.000000 |
| 2 | Chinatown | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.022727 | 0.0 | 0.0 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.034091 |
| 3 | Haight-Ashbury | 0.000000 | 0.028571 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.000000 |
| 4 | Hayes Valley/Tenderloin/North of Market | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.104478 |

From there we will build a new neighborhood_venues data frame that will list the most common venues within a particular neighborhood. This data frame will be used to conduct our clustering analysis.

In [99]:
```
top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))


# create a new dataframe
neighborhoods_venues = pd.DataFrame(columns=columns)
neighborhoods_venues['Neighborhood'] = san_fran_freq['Neighborhood']

for ind in np.arange(san_fran_freq.shape[0]):
    row_categories = san_fran_freq.iloc[ind,:].iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)
    neighborhoods_venues.iloc[ind, 1:] = row_categories_sorted.index.values[0:top_venues]

neighborhoods_venues
```

Out[99]:

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Bayview-Hunters Point | Park | Southern / Soul Food Restaurant | Light Rail Station | Chinese Restaurant | Pharmacy | Theater | Grocery Store | BBQ Joint | Mexican Restaurant | Gym |
| 1 | Castro/Noe Valley | Gay Bar | Thai Restaurant | Coffee Shop | Yoga Studio | Pharmacy | Flower Shop | Mediterranean Restaurant | New American Restaurant | Clothing Store | Convenience Store |
| 2 | Chinatown | Chinese Restaurant | Bakery | Hotel | Coffee Shop | Vietnamese Restaurant | Dim Sum Restaurant | Tea Room | Cocktail Bar | Szechuan Restaurant | Bank |
| 3 | Haight-Ashbury | Coffee Shop | Boutique | Park | Ice Cream Shop | Bakery | Bus Stop | Breakfast Spot | Bubble Tea Shop | Mexican Restaurant | Burrito Place |
| 4 | Hayes Valley/Tenderloin/North of Market | Vietnamese Restaurant | Sandwich Place | Hotel | Thai Restaurant | Theater | Hotel Bar | Coffee Shop | Beer Bar | Concert Hall | Bakery |

## Kmeans Cluster

After our venue data frame has been completed, we can now build our clustering model. We are using the sklearn package we are importing the kmeans clustering algorithm to help to build our model. For our number of k clusters, we decided to 2 was the most optimal number because of the small number of neighborhoods. After the algorithm was modeled, we created a new data frame sf_new_df to show the Cluster labels.

In [101]:
```
kclusters = 2
san_fran_clusters = san_fran_freq.drop(['Neighborhood'],1)
kmeans = KMeans(n_clusters = kclusters, random_state = 0).fit(san_fran_clusters)
neighborhoods_venues.insert(0,'Cluster Labels', kmeans.labels_)
sf_new_df = san_fran_df
sf_new_df = sf_new_df.merge(neighborhoods_venues, on = 'Neighborhood')
```

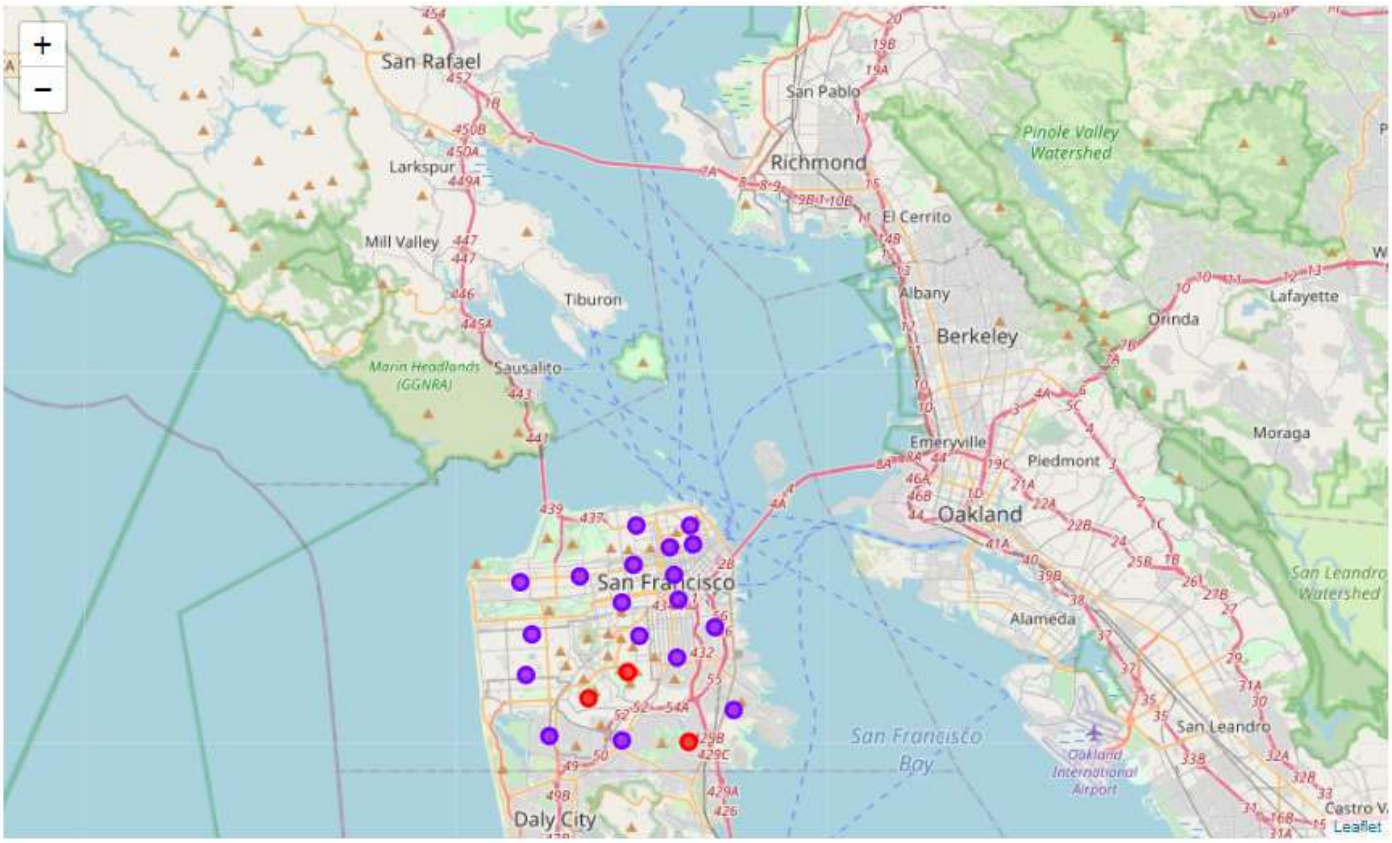
In [102]:
```
sf_new_df
```

Out[102]:

| | Zip Code | Neighborhood | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Co |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 94102 | Hayes Valley/Tenderloin/North of Market | 37.7813 | -122.4167 | 1 | Vietnamese Restaurant | Sandwich Place | Hotel | Thai Restaurant | Theater | Hotel Bar | Coffee Shop | Be |
| 1 | 94103 | South of Market | 37.7725 | -122.4147 | 1 | Nightclub | Coffee Shop | Food Truck | Gay Bar | Cocktail Bar | Sushi Restaurant | Pizza Place | Rent Lo |
| 2 | 94107 | Potrero Hill | 37.7621 | -122.3971 | 1 | Breakfast Spot | Coffee Shop | Café | Brewery | Cosmetics Shop | Office | Grocery Store | Rest |
| 3 | 94108 | Chinatown | 37.7929 | -122.4079 | 1 | Chinese Restaurant | Bakery | Hotel | Coffee Shop | Vietnamese Restaurant | Dim Sum Restaurant | Tea Room | C |
| 4 | 94109 | Polk/Russian Hill (Nob Hill) | 37.7917 | -122.4186 | 1 | Grocery Store | Thai Restaurant | Italian Restaurant | Massage Studio | Vietnamese Restaurant | Bakery | French Restaurant | |

# 5 Results:

Now that the model has been build, we can take a visual look at where these current clusters are located. Using the Folium library, we were able to build a map using the new data frame with the cluster labels. Cluster 1 is indicated by the Red and Cluster 2 is indicated by the Purple:



# 6 Discussion:

Now that we have looked at our results, we can dive deeper into the characteristics of each cluster.

Looking at cluster # 1 below, we can see that this is a much smaller cluster the all the characteristics are very similar. This cluster seems to form around different parks, trails, and playgrounds in the neighborhood. This cluster seems to focus on outdoor activities.

**Cluster #1:**

```
In [112]: sf_new_df.loc[sf_new_df['Cluster Labels'] == 0, sf_new_df.columns[[1] + list(range(5, sf_new_df.shape[1]))]]
```

Out[112]:

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | St. Francis Wood/Miraloma/West Portal | Park | Bus Line | Trail | Monument / Landmark | Tree | Food Court | Flower Shop | Food | Food & Drink Shop | Yoga Studio |
| 17 | Twin Peaks-Glen Park | Playground | Trail | Scenic Lookout | Salon / Barbershop | Dim Sum Restaurant | Shopping Mall | Food | Grocery Store | Bakery | Coffee Shop |
| 20 | Visitacion Valley/Sunnydale | BBQ Joint | Playground | Music Venue | Park | Performing Arts Venue | Trail | Scenic Lookout | Garden | Ethiopian Restaurant | Event Space |

By further examining cluster #2, this cluster seems to be the area where most of the indoor activity takes place. This cluster has more restaurant, theater and even gym locations. And by looking at this data frame we see the number of gym or Fitness Center is rather limited neighborhoods. These locations would be the most ideal locations to open a new fitness center.

**Cluster #2:**

```
In [113]: sf_new_df.loc[sf_new_df['Cluster Labels'] == 1, sf_new_df.columns[[1] + list(range(5, sf_new_df.shape[1]))]]
```

Out[113]:

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Hayes Valley/Tenderloin/North of Market | Vietnamese Restaurant | Sandwich Place | Hotel | Thai Restaurant | Theater | Hotel Bar | Coffee Shop | Beer Bar | Concert Hall | Bakery |
| 1 | South of Market | Nightclub | Coffee Shop | Food Truck | Gay Bar | Cocktail Bar | Sushi Restaurant | Pizza Place | Rental Car Location | Clothing Store | Thai Restaurant |
| 2 | Potrero Hill | Breakfast Spot | Coffee Shop | Café | Brewery | Cosmetics Shop | Office | Grocery Store | Sushi Restaurant | Bookstore | Sandwich Place |
| 3 | Chinatown | Chinese Restaurant | Bakery | Hotel | Coffee Shop | Vietnamese Restaurant | Dim Sum Restaurant | Tea Room | Cocktail Bar | Szechuan Restaurant | Bank |
| 4 | Polk/Russian Hill (Nob Hill) | Grocery Store | Thai Restaurant | Italian Restaurant | Massage Studio | Vietnamese Restaurant | Bakery | French Restaurant | Café | Wine Bar | Gym / Fitness Center |

# 6 Conclusion:

When reviewing the data of each of the San Francisco Neighborhoods, we can see that the neighborhoods in Cluster #2 are the ideal candidates to open a new fitness center. This can be explained by the venue types that are associated with that cluster against what we see in cluster #1. In Cluster #1 we see mainly outdoor venues and locations while Cluster #1 has more indoor venues. Also, public transport also seems more accessible within the 2nd cluster which may mean that the populations may be higher at these locations which can also mean more foot traffic past our fitness location.

With San Francisco being one of the fittest places in the United States, this would be the best time to invest in the creation of this new fitness center.