Final Project

May 18, 2020

1 Capstone Project

1.1 Introduction

In this project, I will explore the relationship between location data and number of COVID-19 cases. Using San Francisco as an example, I will first explore the types of different neighborhoods in San Francisco. Then, I will investigate whether different types of neighborhoods have any connection with the number of COVID-19 cases. As the COVID-19 pandemic escalates in the whole world, the results from this project might be potentially helpful to medical researchers, policy makers, and the general public. On the one hand, people need to understand the means by which the virus has been spreading. On the other hand, it provides information about which areas might have high exposure risk to the virus that people should avoid.

I will use data from three sources in this project. First, the rate of COVID-19 cases by census zip code in San Francisco. This dataset gives me the number of confirmed cases in San Francisco by zip code and normalized by 2017 American Community Survey (ACS) 5-year estimates for population data to calculate rate per 10,000 residents. Second, the latitude and longitude of US zip codes. With information from this zipcode, I will pull data from the FourSquare API for the zip codes. Then the venues from the zip codes will be analyzed and clustered into different types of neighborhoods. Finally, I will look into the association between clusters of neighborhoods and number of COVID-19 cases.

1.2 Data

I will use data from three sources in this project. First, I got the rate of COVID-19 cases by census zip code in San Francisco from the DataSF website. This dataset gives me the number of confirmed cases in San Francisco by zip code and normalized by 2017 American Community Survey (ACS) 5-year estimates for population data to calculate rate per 10,000 residents. Second, the latitude and longitude of US zip codes. With information from this zipcode, I will pull data from the FourSquare API for the zip codes. Then the venues from the zip codes will be analyzed and clustered into different types of neighborhoods. Finally, I will look into the association between clusters of neighborhoods and number of COVID-19 cases. The two datasets can be found below:

- COVID-19 cases by zip code in San Francisco
- Geo-coordinates for US zip codes

1.3 Methodology

In this section, I will describe the methods that are used to analyze the data and achieve the goal of this project. First, I load a few packages that will be useful in exploring the data:

[50]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import folium
import matplotlib.cm as cm
import matplotlib.colors as colors
import requests
from scipy.spatial.distance import cdist
import matplotlib.pyplot as plt
```

Then import the two datasets I will be using for this project:

[16]:
```
covid = pd.read_csv('Rate_of_COVID-19_Cases_by_Census_ZIP_Code_Tabulation_Area.
↪csv')
covid.rename(columns={'ZIP Code': 'Zip'}, inplace = True)
covid.head(5)
```

[16]:

	Data as of	OBJECTID	Zip	Count of Confirmed Cases	Estimated 2017 ACS Population	Estimated Rate of Cases per 10k	Rate Groups	Count of San Francisco Confirmed Cases	Estimated 2017 ACS San Francisco Population	Estimated Rate of San Francisco Cases per 10k
0	2020/05/08	2	94121	34.0	43638	7.79	5-10	1891	864263	21.88
1	2020/05/08	4	94123	28.0	25461	11.00	10-15	1891	864263	
2	2020/05/08	13	94158	18.0	6547	27.49	25-30	1891	864263	
3	2020/05/08	18	94107	115.0	29920	38.44	35-40	1891	864263	
4	2020/05/08	1	94118	31.0	41417	7.48	5-10	1891	864263	

```
1                                          21.88
2                                          21.88
3                                          21.88
4                                          21.88

   Case Rate Difference from San Francisco  \
0                                  -14.09
1                                  -10.88
2                                    5.61
3                                   16.56
4                                  -14.40

                                        multipolygon
0  MULTIPOLYGON (((-122.48542599984555 37.7898249…
1  MULTIPOLYGON (((-122.45005999994794 37.8024729…
2  MULTIPOLYGON (((-122.3836959998312 37.75470099…
3  MULTIPOLYGON (((-122.38530302568738 37.7898378…
4  MULTIPOLYGON (((-122.44767900001601 37.7917029…
```

[13]:
```
coords = pd.read_csv('us-zip-code-latitude-and-longitude.csv')
coords.head(5)
```

[13]:
```
     Zip      City State   Latitude  Longitude  Timezone  \
0  71937      Cove    AR  34.398483  -94.39398        -6
1  72044  Edgemont    AR  35.624351  -92.16056        -6
2  56171  Sherburn    MN  43.660847  -94.74357        -6
3  49430    Lamont    MI  43.010337  -85.89754        -5
4  52585  Richland    IA  41.194129  -91.98027        -6

   Daylight savings time flag   geopoint
0                           1  34.398483
1                           1  35.624351
2                           1  43.660847
3                           1  43.010337
4                           1  41.194129
```

Then, I merge the two datasets and keep only the zip codes in both datasets, and removed zip codes with blank confirmed cases:

[34]:
```
df = pd.merge(left = covid, right = coords)
df = df[np.isnan(df['Count of Confirmed Cases']) == False]
df.head(5)
```

[34]:
```
   Data as of  OBJECTID    Zip  Count of Confirmed Cases  \
0  2020/05/08         2  94121                      34.0
1  2020/05/08         4  94123                      28.0
2  2020/05/08        18  94107                     115.0
```

```
3  2020/05/08         1  94118                      31.0
4  2020/05/08        27  94117                      48.0

   Estimated 2017 ACS Population  Estimated Rate of Cases per 10k Rate Groups  \
0                          43638                             7.79        5-10
1                          25461                            11.00       10-15
2                          29920                            38.44       35-40
3                          41417                             7.48        5-10
4                          43610                            11.01       10-15

   Count of San Francisco Confirmed Cases  \
0                                    1891
1                                    1891
2                                    1891
3                                    1891
4                                    1891

   Estimated 2017 ACS San Francisco Population  \
0                                       864263
1                                       864263
2                                       864263
3                                       864263
4                                       864263

   Estimated Rate of San Francisco Cases per 10k  \
0                                          21.88
1                                          21.88
2                                          21.88
3                                          21.88
4                                          21.88

   Case Rate Difference from San Francisco  \
0                                   -14.09
1                                   -10.88
2                                    16.56
3                                   -14.40
4                                   -10.87

                                        multipolygon           City State  \
0  MULTIPOLYGON (((-122.48542599984555 37.7898249…  San Francisco    CA
1  MULTIPOLYGON (((-122.45005999994794 37.8024729…  San Francisco    CA
2  MULTIPOLYGON (((-122.38530302568738 37.7898378…  San Francisco    CA
3  MULTIPOLYGON (((-122.44767900001601 37.7917029…  San Francisco    CA
4  MULTIPOLYGON (((-122.42992899967089 37.7779089…  San Francisco    CA

    Latitude  Longitude  Timezone  Daylight savings time flag   geopoint
0  37.778729 -122.49265        -8                           1  37.778729
```

```
1  37.801028 -122.43836        -8                          1  37.801028
2  37.766529 -122.39577        -8                          1  37.766529
3  37.782029 -122.46158        -8                          1  37.782029
4  37.770937 -122.44276        -8                          1  37.770937
```

Then, I use FourSquare to get venues near these zip codes:

[56]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?
↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item␣
↪in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']
```

```
    return(nearby_venues)
```

[58]:
```
CLIENT_ID = '2PATX5OVWCHMEOOHRZ3OXSTVJVVTUV4EVEHZGFN2KKYPGHQI' # your␣
 ↪Foursquare ID
CLIENT_SECRET = 'UNIJA3LTM5DYJMQGD4ZD3BLJUTPDMJ2KTGWDY1DP5ZNEJ1TD' # your␣
 ↪Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100
radius = 50

df_venues = getNearbyVenues(names=df['Zip'],
                                    latitudes=df['Latitude'],
                                    longitudes=df['Longitude']
                                   )
```

```
94121
94123
94107
94118
94117
94116
94115
94109
94114
94112
94110
94105
94103
94132
94102
94133
94134
94122
94124
94127
94131
```

[ ]:
```
Then, I perform one hot coding to the venues:
```

[59]:
```
# one hot encoding
df_onehot = pd.get_dummies(df_venues[['Venue Category']], prefix="",␣
 ↪prefix_sep="")

# add neighborhood column back to dataframe
df_onehot['Neighborhood'] = df_venues['Neighborhood']

# move neighborhood column to the first column
```

```
fixed_columns = [df_onehot.columns[-1]] + list(df_onehot.columns[:-1])
df_onehot = df_onehot[fixed_columns]

df_grouped = df_onehot.groupby('Neighborhood').mean().reset_index()
df_grouped
```

[59]:

	Neighborhood	ATM	Accessories Store	Adult Boutique	\
0	94102	0.000000	0.000000	0.000000	
1	94103	0.000000	0.000000	0.000000	
2	94105	0.000000	0.000000	0.000000	
3	94107	0.000000	0.000000	0.000000	
4	94109	0.000000	0.000000	0.000000	
5	94110	0.000000	0.000000	0.000000	
6	94112	0.000000	0.000000	0.000000	
7	94114	0.000000	0.000000	0.013333	
8	94115	0.000000	0.000000	0.000000	
9	94116	0.000000	0.000000	0.000000	
10	94117	0.000000	0.026316	0.000000	
11	94118	0.016949	0.000000	0.000000	
12	94121	0.000000	0.000000	0.000000	
13	94122	0.000000	0.000000	0.000000	
14	94123	0.000000	0.000000	0.000000	
15	94124	0.000000	0.000000	0.000000	
16	94127	0.000000	0.000000	0.000000	
17	94131	0.000000	0.000000	0.000000	
18	94132	0.000000	0.018519	0.000000	
19	94133	0.000000	0.012048	0.000000	
20	94134	0.000000	0.000000	0.000000	

	African Restaurant	Alternative Healer	American Restaurant	Antique Shop	\
0	0.000000	0.000000	0.025974	0.000000	
1	0.000000	0.000000	0.000000	0.000000	
2	0.000000	0.013333	0.013333	0.000000	
3	0.000000	0.000000	0.033333	0.033333	
4	0.012500	0.000000	0.025000	0.000000	
5	0.000000	0.000000	0.020408	0.000000	
6	0.000000	0.000000	0.000000	0.000000	
7	0.000000	0.000000	0.013333	0.000000	
8	0.000000	0.000000	0.020202	0.000000	
9	0.000000	0.000000	0.000000	0.000000	
10	0.000000	0.000000	0.000000	0.000000	
11	0.000000	0.000000	0.000000	0.000000	
12	0.000000	0.000000	0.046512	0.023256	
13	0.000000	0.000000	0.000000	0.000000	
14	0.000000	0.014286	0.028571	0.000000	
15	0.058824	0.000000	0.000000	0.000000	
16	0.000000	0.000000	0.000000	0.000000	

```
17            0.000000            0.000000            0.000000      0.000000
18            0.000000            0.000000            0.000000      0.000000
19            0.000000            0.000000            0.000000      0.000000
20            0.000000            0.000000            0.000000      0.000000

    Art Gallery  Art Museum  …     Trail  Trattoria/Osteria  \
0      0.000000    0.012987  …  0.000000           0.000000
1      0.026667    0.000000  …  0.000000           0.000000
2      0.040000    0.000000  …  0.000000           0.000000
3      0.000000    0.000000  …  0.000000           0.000000
4      0.000000    0.000000  …  0.000000           0.000000
5      0.020408    0.000000  …  0.000000           0.000000
6      0.000000    0.000000  …  0.000000           0.000000
7      0.013333    0.000000  …  0.013333           0.000000
8      0.000000    0.000000  …  0.000000           0.000000
9      0.000000    0.000000  …  0.000000           0.000000
10     0.000000    0.000000  …  0.000000           0.000000
11     0.016949    0.000000  …  0.000000           0.000000
12     0.000000    0.000000  …  0.000000           0.000000
13     0.000000    0.000000  …  0.000000           0.000000
14     0.000000    0.000000  …  0.000000           0.000000
15     0.000000    0.000000  …  0.000000           0.000000
16     0.000000    0.000000  …  0.250000           0.000000
17     0.000000    0.000000  …  0.142857           0.000000
18     0.000000    0.000000  …  0.000000           0.000000
19     0.012048    0.000000  …  0.036145           0.012048
20     0.000000    0.000000  …  0.400000           0.000000

    Turkish Restaurant  Udon Restaurant  Vegetarian / Vegan Restaurant  \
0             0.000000         0.000000                       0.012987
1             0.000000         0.000000                       0.000000
2             0.000000         0.000000                       0.000000
3             0.000000         0.000000                       0.000000
4             0.000000         0.000000                       0.012500
5             0.000000         0.000000                       0.000000
6             0.000000         0.000000                       0.000000
7             0.000000         0.000000                       0.000000
8             0.000000         0.000000                       0.000000
9             0.000000         0.000000                       0.000000
10            0.000000         0.000000                       0.000000
11            0.016949         0.000000                       0.000000
12            0.000000         0.000000                       0.000000
13            0.000000         0.000000                       0.000000
14            0.000000         0.000000                       0.000000
15            0.000000         0.000000                       0.000000
16            0.000000         0.000000                       0.000000
17            0.000000         0.000000                       0.000000
```

```
18            0.000000         0.018519                      0.000000
19            0.000000         0.000000                      0.000000
20            0.000000         0.000000                      0.000000
```

```
    Vietnamese Restaurant  Wine Bar  Wine Shop  Wings Joint  Yoga Studio
0                0.000000  0.038961   0.012987       0.0000     0.000000
1                0.000000  0.026667   0.000000       0.0000     0.000000
2                0.013333  0.000000   0.000000       0.0000     0.013333
3                0.033333  0.000000   0.066667       0.0000     0.000000
4                0.037500  0.025000   0.012500       0.0125     0.012500
5                0.000000  0.000000   0.000000       0.0000     0.000000
6                0.058824  0.000000   0.000000       0.0000     0.000000
7                0.000000  0.026667   0.013333       0.0000     0.026667
8                0.020202  0.020202   0.000000       0.0000     0.020202
9                0.000000  0.000000   0.000000       0.0000     0.025000
10               0.000000  0.000000   0.000000       0.0000     0.000000
11               0.033898  0.016949   0.033898       0.0000     0.016949
12               0.023256  0.000000   0.000000       0.0000     0.000000
13               0.000000  0.000000   0.000000       0.0000     0.000000
14               0.000000  0.028571   0.000000       0.0000     0.000000
15               0.000000  0.000000   0.000000       0.0000     0.000000
16               0.000000  0.000000   0.000000       0.0000     0.000000
17               0.000000  0.000000   0.000000       0.0000     0.000000
18               0.000000  0.000000   0.000000       0.0000     0.000000
19               0.000000  0.000000   0.000000       0.0000     0.024096
20               0.000000  0.000000   0.000000       0.0000     0.000000

[21 rows x 220 columns]
```

Then the top 10 venues are stored into a data frame:

[43]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

[174]:
```
import numpy as np
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_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_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = df_grouped['Neighborhood']

for ind in np.arange(df_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] =␣
↪return_most_common_venues(df_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

[174]:

	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	94102	Coffee Shop	Hotel	Café	Theater	Wine Bar	Beer Bar	Cocktail Bar	Concert Hall	Thai Restaurant	Park
1	94103	Nightclub	Cocktail Bar	Gay Bar	Food Truck	Motorcycle Shop	Thai Restaurant	Cosmetics Shop	Mexican Restaurant	Bar	Sushi Restaurant
2	94105	Coffee Shop	Food Truck	Café	Art Gallery	Gym	Gym / Fitness Center	Juice Bar	Burger Joint	Japanese Restaurant	New American Restaurant
3	94107	Wine Shop	Park	Café	Breakfast Spot	Coffee Shop	Deli / Bodega	Distillery	Sandwich Place	Rock Club	French Restaurant
4	94109	Grocery Store	Steakhouse	Gym / Fitness Center	Deli / Bodega	Coffee Shop	Vietnamese Restaurant	Pet Store	Chinese Restaurant	Diner	Bar

In order to find the optimal number of clusters for the data, I use the elbow method:

[51]:
```
distortions = []
inertias = []
mapping1 = {}
mapping2 = {}
```

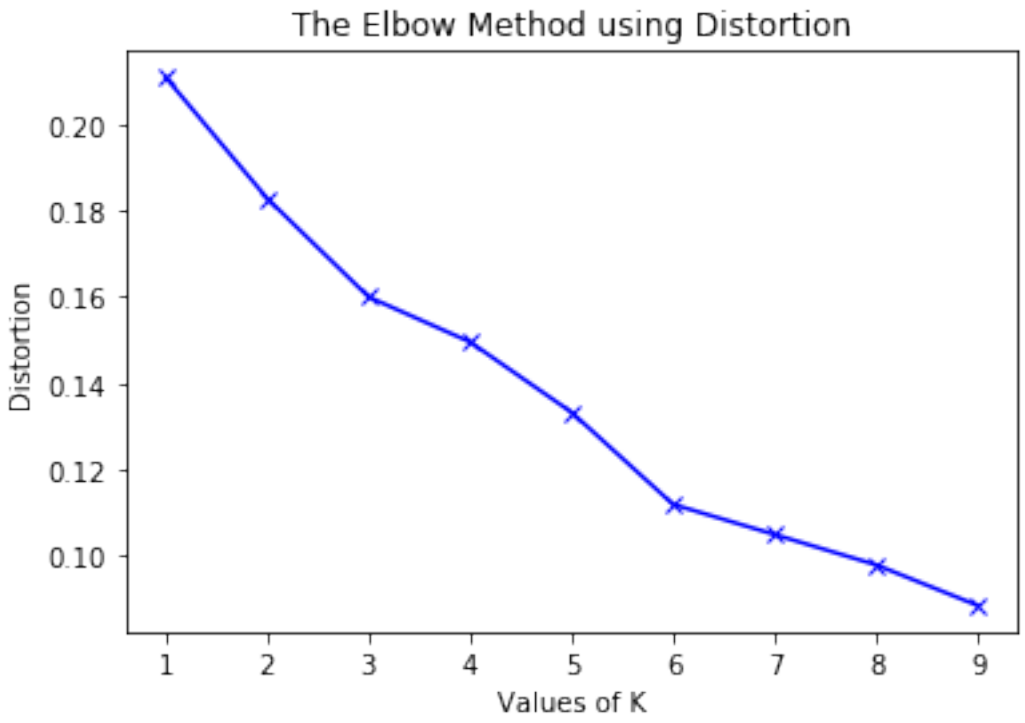
```
K = range(1,10)
X = df_grouped.drop('Neighborhood', 1)

for k in K:
    #Building and fitting the model
    kmeanModel = KMeans(n_clusters=k).fit(X)
    kmeanModel.fit(X)

    distortions.append(sum(np.min(cdist(X, kmeanModel.cluster_centers_,
                      'euclidean'),axis=1)) / X.shape[0])
    inertias.append(kmeanModel.inertia_)

    mapping1[k] = sum(np.min(cdist(X, kmeanModel.cluster_centers_,
                 'euclidean'),axis=1)) / X.shape[0]
    mapping2[k] = kmeanModel.inertia_

plt.plot(K, distortions, 'bx-')
plt.xlabel('Values of K')
plt.ylabel('Distortion')
plt.title('The Elbow Method using Distortion')
plt.show()
```



From the plot, it looks like the elbow point could be 3 or 6. To reserve the interpretability of the clusters, I choose to run k-means with 3 clusters:

[129]:
```
# set number of clusters
kclusters = 3

df_grouped_clustering = df_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(df_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
pd.Series(kmeans.labels_).value_counts()
```

[129]:
```
1    18
2     2
0     1
dtype: int64
```

1.4 Results

Now, the results for the clustering analysis are presented. I first merge the clustering results with the original dataset:

[175]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

[188]:
```
df_merged = df

# merge df_grouped with toronto_data to add latitude/longitude for each␣
 ↪neighborhood
df_merged.rename(columns={'ZIP': 'Neighborhood'}, inplace=True)
df_merged = df_merged.join(neighborhoods_venues_sorted.
 ↪set_index('Neighborhood'), on='Neighborhood')
df_merged['Neighborhood'] = df_merged['Neighborhood'].apply(str)
df_merged # check the last columns!
```

[188]:

	Data as of	OBJECTID	Neighborhood	Count of Confirmed Cases	\
0	2020/05/08	2	94121	34.0	
1	2020/05/08	4	94123	28.0	
2	2020/05/08	18	94107	115.0	
3	2020/05/08	1	94118	31.0	
4	2020/05/08	27	94117	48.0	
5	2020/05/08	26	94116	30.0	
6	2020/05/08	25	94115	117.0	
7	2020/05/08	20	94109	87.0	
8	2020/05/08	24	94114	38.0	
9	2020/05/08	23	94112	231.0	
11	2020/05/08	21	94110	283.0	

12	2020/05/08	17	94105	15.0
13	2020/05/08	15	94103	125.0
16	2020/05/08	10	94132	28.0
17	2020/05/08	14	94102	101.0
18	2020/05/08	11	94133	25.0
19	2020/05/08	12	94134	111.0
20	2020/05/08	3	94122	53.0
21	2020/05/08	5	94124	147.0
22	2020/05/08	6	94127	23.0
25	2020/05/08	9	94131	45.0

	Estimated 2017 ACS Population	Estimated Rate of Cases per 10k	\
0	43638	7.79	
1	25461	11.00	
2	29920	38.44	
3	41417	7.48	
4	43610	11.01	
5	47708	6.29	
6	35751	32.73	
7	56587	15.37	
8	34561	11.00	
9	85373	27.06	
11	73737	38.38	
12	7675	19.54	
13	26990	46.31	
16	31155	8.99	
17	30140	33.51	
18	26942	9.28	
19	43074	25.77	
20	62516	8.48	
21	35492	41.42	
22	21093	10.90	
25	29056	15.49	

	Rate Groups	Count of San Francisco Confirmed Cases	\
0	5-10	1891	
1	10-15	1891	
2	35-40	1891	
3	5-10	1891	
4	10-15	1891	
5	5-10	1891	
6	30-35	1891	
7	15-20	1891	
8	10-15	1891	
9	25-30	1891	
11	35-40	1891	
12	15-20	1891	

```
13       45-50                                        1891
16        5-10                                        1891
17       30-35                                        1891
18        5-10                                        1891
19       25-30                                        1891
20        5-10                                        1891
21       40-45                                        1891
22       10-15                                        1891
25       15-20                                        1891

    Estimated 2017 ACS San Francisco Population  \
0                                          864263
1                                          864263
2                                          864263
3                                          864263
4                                          864263
5                                          864263
6                                          864263
7                                          864263
8                                          864263
9                                          864263
11                                         864263
12                                         864263
13                                         864263
16                                         864263
17                                         864263
18                                         864263
19                                         864263
20                                         864263
21                                         864263
22                                         864263
25                                         864263

    Estimated Rate of San Francisco Cases per 10k  …  \
0                                           21.88  …
1                                           21.88  …
2                                           21.88  …
3                                           21.88  …
4                                           21.88  …
5                                           21.88  …
6                                           21.88  …
7                                           21.88  …
8                                           21.88  …
9                                           21.88  …
11                                          21.88  …
12                                          21.88  …
13                                          21.88  …
```

```
16                                      21.88  …
17                                      21.88  …
18                                      21.88  …
19                                      21.88  …
20                                      21.88  …
21                                      21.88  …
22                                      21.88  …
25                                      21.88  …

             1st Most Common Venue 2nd Most Common Venue  \
0                            Café    Chinese Restaurant
1                French Restaurant  Gym / Fitness Center
2                        Wine Shop                  Park
3                           Bakery   Japanese Restaurant
4                       Coffee Shop             Boutique
5                Chinese Restaurant   Dumpling Restaurant
6                           Bakery                   Spa
7                     Grocery Store            Steakhouse
8                           Gay Bar       Thai Restaurant
9                Mexican Restaurant           Pizza Place
11               Mexican Restaurant         Grocery Store
12                      Coffee Shop            Food Truck
13                        Nightclub         Cocktail Bar
16                        Juice Bar                 Café
17                      Coffee Shop                Hotel
18                      Coffee Shop           Pizza Place
19                            Trail                Garden
20               Chinese Restaurant            Playground
21  Southern / Soul Food Restaurant    Mexican Restaurant
22                         Bus Line                Garden
25                            Trail                  Park

   3rd Most Common Venue  4th Most Common Venue   5th Most Common Venue  \
0    American Restaurant            Pizza Place       Convenience Store
1         Sandwich Place                   Park                     Spa
2                  Café        Breakfast Spot             Coffee Shop
3     Burmese Restaurant     Chinese Restaurant         Thai Restaurant
4                   Park            Pizza Place  Thrift / Vintage Store
5                  Café      Korean Restaurant      Light Rail Station
6                  Café         Cosmetics Shop             Pizza Place
7   Gym / Fitness Center          Deli / Bodega             Coffee Shop
8           Coffee Shop         Scenic Lookout             Yoga Studio
9         Sandwich Place  Vietnamese Restaurant             Bus Station
11          Coffee Shop                   Park             Pizza Place
12                 Café            Art Gallery                     Gym
13              Gay Bar             Food Truck         Motorcycle Shop
16        Cosmetics Shop         Sandwich Place             Pizza Place
```

17	Café	Theater	Wine Bar
18	Café	Italian Restaurant	Park
19	Park	Baseball Field	Yoga Studio
20	Light Rail Station	Dessert Shop	Shoe Store
21	Bakery	Pharmacy	Theater
22	Pawn Shop	Trail	Yoga Studio
25	Shopping Mall	Scenic Lookout	Cantonese Restaurant

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	Japanese Restaurant	Music Store	Dessert Shop
1	Burger Joint	Salad Place	Taco Place
2	Deli / Bodega	Distillery	Sandwich Place
3	Wine Shop	Pizza Place	Vietnamese Restaurant
4	Liquor Store	Gastropub	Bookstore
5	Liquor Store	Sandwich Place	Shoe Store
6	Chinese Restaurant	Boutique	Salon / Barbershop
7	Vietnamese Restaurant	Pet Store	Chinese Restaurant
8	Clothing Store	Playground	Pet Store
9	Liquor Store	Fried Chicken Joint	Food Truck
11	Bookstore	Deli / Bodega	Massage Studio
12	Gym / Fitness Center	Juice Bar	Burger Joint
13	Thai Restaurant	Cosmetics Shop	Mexican Restaurant
16	Clothing Store	Bakery	Lingerie Store
17	Beer Bar	Cocktail Bar	Concert Hall
18	Chinese Restaurant	Deli / Bodega	Bakery
19	Ethiopian Restaurant	Food & Drink Shop	Flower Shop
20	Café	Pharmacy	Hill
21	Café	Dumpling Restaurant	Gym
22	Ethiopian Restaurant	Food & Drink Shop	Flower Shop
25	Dog Run	Coffee Shop	Grocery Store

	9th Most Common Venue	10th Most Common Venue
0	Antique Shop	Pharmacy
1	Coffee Shop	Thai Restaurant
2	Rock Club	French Restaurant
3	Pet Store	Yoga Studio
4	Breakfast Spot	Playground
5	Spa	Bubble Tea Shop
6	Bubble Tea Shop	New American Restaurant
7	Diner	Bar
8	Indian Restaurant	Deli / Bodega
9	Café	Cajun / Creole Restaurant
11	Fish Market	Cocktail Bar
12	Japanese Restaurant	New American Restaurant
13	Bar	Sushi Restaurant
16	Candy Store	Mexican Restaurant
17	Thai Restaurant	Park

```
18                  Trail             Sandwich Place
19            Fish Market        Filipino Restaurant
20     Electronics Store             History Museum
21            Bus Station                       Park
22            Fish Market        Filipino Restaurant
25    Salon / Barbershop         Dim Sum Restaurant

[21 rows x 30 columns]
```

Then, I check the frequency of the clusters:

[83]:
```
df_merged['Cluster Labels'].value_counts()
```

[83]:
```
1    18
2     2
0     1
Name: Cluster Labels, dtype: int64
```

It looks like most zip codes are in cluster 1, and very few clusters fall in the other 2 clusters. So I further check the centroids of the clusters. Looks like Cluster 0 is near Chinese restaurants, dessert shop, playground, pharmacy, etc. Cluster 1 has a little bit of everything, and Cluster 2 is close to trail, garden, bus line, etc.

[136]:
```
centroids = pd.DataFrame(kmeans.cluster_centers_).transpose()
centroids.index = df_grouped.columns[1:]
centroids.columns = ('Cluster 0', 'Cluster 1', 'Cluster 2')
print(centroids.sort_values(by = ['Cluster 0'], ascending = False).head(10))
print(centroids.sort_values(by = ['Cluster 1'], ascending = False).head(10))
print(centroids.sort_values(by = ['Cluster 2'], ascending = False).head(10))
```

```
                    Cluster 0  Cluster 1  Cluster 2
Chinese Restaurant      0.250   0.021823        0.0
Dessert Shop            0.125   0.009525        0.0
Playground              0.125   0.011592        0.0
Pharmacy                0.125   0.011493        0.0
Café                    0.125   0.036026        0.0
Light Rail Station      0.125   0.004412        0.0
Shoe Store              0.125   0.003592        0.0
Office                  0.000   0.001852        0.0
Motorcycle Shop         0.000   0.002222        0.0
Music School            0.000   0.000722        0.0
                    Cluster 0  Cluster 1  Cluster 2
Coffee Shop             0.000   0.047331        0.0
Café                    0.125   0.036026        0.0
Mexican Restaurant      0.000   0.031244        0.0
Park                    0.000   0.029974        0.1
Pizza Place             0.000   0.028122        0.0
Bakery                  0.000   0.025447        0.0
```

```
Chinese Restaurant      0.250   0.021823        0.0
Sandwich Place          0.000   0.021356        0.0
Thai Restaurant         0.000   0.017783        0.0
Grocery Store           0.000   0.015412        0.0
                     Cluster 0     Cluster 1  Cluster 2
Trail                      0.0  1.068528e-02      0.325
Garden                     0.0  1.481481e-03      0.225
Bus Line                   0.0  1.633987e-03      0.125
Pawn Shop                  0.0 -5.204170e-18      0.125
Baseball Field             0.0  1.633987e-03      0.100
Park                       0.0  2.997382e-02      0.100
Moroccan Restaurant        0.0  2.032730e-03      0.000
Motel                      0.0  7.936508e-04      0.000
Motorcycle Shop            0.0  2.222222e-03      0.000
Music School               0.0  7.215007e-04      0.000
```

Then, get ready for displaying clustering results on the map, with number of COVID-19 cases layered at the bottom:

[189]:
```
import folium
import matplotlib.cm as cm
import matplotlib.colors as colors
# create map
map_clusters = folium.Map(location=[37.773972, -122.431297], zoom_start=12)
```

[196]:
```
import urllib.request, json
with urllib.request.urlopen("https://data.sfgov.org/api/geospatial/favi-qct6?
 ↪method=export&format=GeoJSON") as url:
    data = json.loads(url.read().decode())
```

[190]:
```
map_clusters.choropleth(
    geo_data=data,
    data=df_merged,
    columns=['Neighborhood', 'Estimated Rate of Cases per 10k'],
    key_on='feature.properties.zip_code',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Estimated Rate of COVID-19 Cases per 10k in San Francisco'
)
```

[194]:
```
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
```

```
markers_colors = []
for lat, lon, poi, cluster in zip(df_merged['Latitude'],␣
 ↪df_merged['Longitude'], df_merged['Neighborhood'], df_merged['Cluster␣
 ↪Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

[194]: <folium.folium.Map at 0x7fe1ec03cc88>

1.5 Discussion

From the results I presented above, it can be seen that the majority of zip codes in San Francisco is homogeneous. The neighborhoods in San Francisco do not have many distinctive types. Most neighborhoods have all kinds of venues around them, which means San Francisco is in general a very convenient city to live in. It is also possible that the distinctive types of neighborhoods actually exist, but they are smaller than what a zip code covers. Exploring the zip codes and estimated rate of COVID-19 cases per 10k reveals that the relationship between the two might be very weak. From the map, it can be seen that there are more COVID-19 cases in the east side of San Francisco. However, there is no clear evidence that more cases are necessarily related to the zip code, or the venues around the area. It is likely that other factors might be associated with the high number of COVID-19 cases, such as population density. Further research is needed to provide evidence regarding other factors.

1.6 Conclusion

In this project, I used exploratory data analysis methods and machine learning methods, such as k-means, to explore the clustering of neighborhoods in San Francisco based on zip codes and the venues around the zip codes. Then, the relationship between the clusters and the number of COVID-19 cases per 10k population was investigated. No strong linkage was found between the clustering structure of the neighborhoods and the number of cases.