Capstone project

For

IBM Data Science Specialization

Urban Happiness

Ayushi Sharma

sharmayu20@gmail.com

Introduction/Business Problem

Relocating from one place to another is in itself a very difficult task for a person, he/she has to decide whether the neighborhood has the required venues that he/she likes and on top of that whether the neighborhood is safe.

What if a family man wants to move and is unaware of the environment around his selected neighborhood? This is where the project Urban Happiness comes in.

Urban Happiness is a project written in Python that can help a user decide his/her favorable neighborhood based on the venue that he requires and keeping in mind the crime rate of the neighborhood in city of San Francisco.

Urban Happiness presents a map to the user with the crime ranges of the locality and the clustered neighborhood based on the venues. It also outputs the closest neighborhood from the venue given by the user.

The map marks two neighborhoods that are close to the venue given by the user.

The project clusters the neighborhoods based on all the venues and the crime rate in that neighborhood into three clusters and shows it to the user via a map.

Data

The data that will be required for the project would be the crimes rate in San Francisco based on the neighborhoods, the json file that contains the coordinates of the neighborhoods in the form of a polygon which can then be used by folium to mark the neighborhoods, then the postal code or the pincode of the neighborhoods which is then used to gather the exact latitude and the longitude of the neighborhood, then the venue list that is present in the given locality which can be fetched using the foursquare API.

The following section has a detailed description of all the data that will be used for the completion of the project.

1. Crime rate in San Francisco
   This data file is a csv file that contains the arrests or crimes that were committed in different neighborhood of San Francisco.
   The important columns that are considered for the project are:
   - District
   - Count of crimes in each district

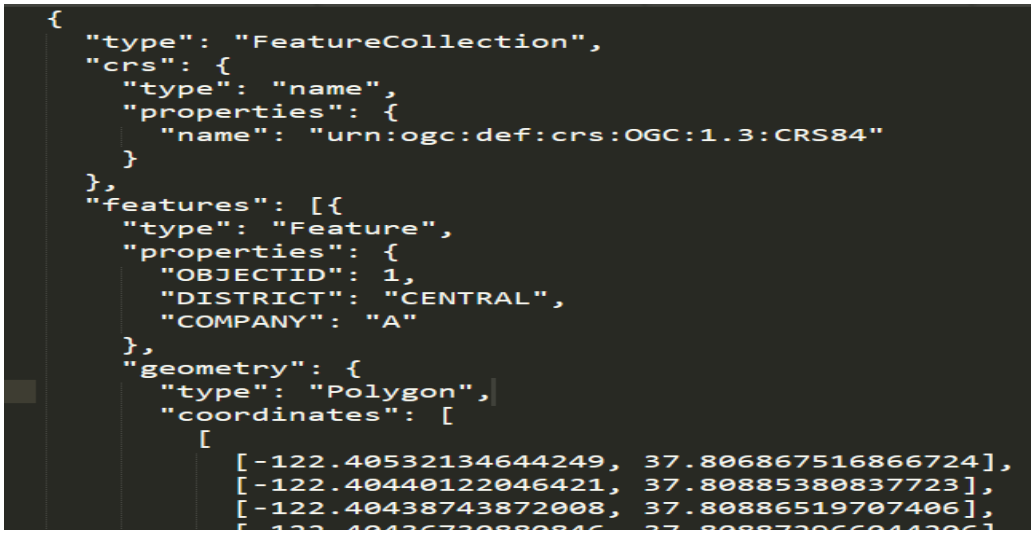
   Snapshot of the csv file used.

IncidntNur	Category	Descript	DayOfWee	Date	Time	PdDistrict	Resolution	Address	X	Y	Location	PdId
1.2E+08	WEAPON I	POSS OF P	Friday	01/29/201	11:00	SOUTHERI	ARREST, B	800 Block	-122.403	37.77542	(37.77542(	1.2E+13
1.2E+08	WEAPON I	FIREARM,	Friday	01/29/201	11:00	SOUTHERI	ARREST, B	800 Block	-122.403	37.77542	(37.77542(	1.2E+13
1.41E+08	WARRANT	WARRANT	Monday	04/25/201	14:59	BAYVIEW	ARREST, B	KEITH ST /	-122.389	37.72998	(37.72998(	1.41E+13
1.6E+08	NON-CRIM	LOST PROI	Tuesday	########	23:50	TENDERLC	NONE	JONES ST /	-122.413	37.78579	(37.78578	1.6E+13
1.6E+08	NON-CRIM	LOST PROI	Friday	########	00:30	MISSION	NONE	16TH ST /	-122.42	37.76505	(37.76505(	1.6E+13
1.6E+08	ASSAULT	BATTERY	Friday	########	21:35	NORTHERI	NONE	1700 Block	-122.426	37.78802	(37.78801	1.6E+13
1.6E+08	OTHER OF	PAROLE VI	Saturday	########	00:04	SOUTHERI	ARREST, B	MARY ST /	-122.406	37.78088	(37.78087	1.6E+13
1.6E+08	NON-CRIM	FIRE REPO	Saturday	########	01:02	TENDERLC	NONE	200 Block	-122.412	37.78398	(37.78398(	1.6E+13
1.6E+08	WARRANT	WARRANT	Saturday	########	12:21	SOUTHERI	ARREST, B	4TH ST / B	-122.393	37.77579	(37.77578	1.6E+13
1.6E+08	MISSING P	FOUND PE	Friday	########	10:06	BAYVIEW	NONE	100 Block	-122.387	37.72097	(37.72096(	1.6E+13
1.6E+08	LARCENY/	ATTEMPTE	Friday	01/29/201	22:30	TARAVAL	NONE	1200 Block	-122.477	37.76448	(37.76447	1.6E+13
1.6E+08	NON-CRIM	AIDED CAS	Saturday	########	13:30	TARAVAL	NONE	2200 Block	-122.478	37.74574	(37.74573	1.6E+13
1.6E+08	OTHER OF	RESISTING	Monday	01/25/201	23:20	BAYVIEW	ARREST, B	200 Block	-122.377	37.7357	(37.73569	1.6E+13
1.41E+08	ASSAULT	AGGRAVA	Thursday	09/15/201	07:40	INGLESIDE	ARREST, B	SILVER AV	-122.432	37.72927	(37.72927(	1.41E+13

2. GeoJson file that contains information on San Francisco

   This GeoJson file contains the information on San Francisco in the form of key value pairs. The file contains the coordinates of different neighborhoods in San Francisco. These coordinates are given to the folium's geo_data attribute of the Chloropleth class.
   This geo_data is responsible for marking the districts and the colors for the folium map.

Snapshot of the geoJson file



3. Postal codes

Since there was no website that provided the postal codes of the neighborhood, the postal codes were manually added to the data frame by adding another column to the data frame. Postal codes were manually searched from the in internet and added to the data frame.

Snapshot of the modified data frame.

```
df3['Pincode']=pincode
```

Pincodes of the neighbourhood added to the dataframe

```
df3
```

5]:

	Neighbourhood	Count	Pincode
0	BAYVIEW	14303	94124
1	CENTRAL	17666	94104
2	INGLESIDE	11594	94112
3	MISSION	19503	94114
4	NORTHERN	20100	94109
5	PARK	8699	94117
6	RICHMOND	8922	94121
7	SOUTHERN	28445	94105
8	TARAVAL	11325	94116
9	TENDERLOIN	9942	94102

4. GeoCoder:

Geocoder package is used to fetch the latitudes and the longitudes of place by passing the postal code to it.
This package has a function nomi.query_postal_code() which takes postal code as a input and outputs the longitude and the latitude of the place

Snapshot of the function used to access the longitude and the latitude and the result.

```
def get_geocoder(post):

    nomi = pgeocode.Nominatim('us')
    x=nomi.query_postal_code('{}'.format(post))

    lat=x.latitude
    long=x.longitude
    #print(lat)

    return lat,long

df3['Latitude'], df3['Longitude'] = zip(*df3['Pincode'].apply(get_geocoder))
df3=df3[['Neighbourhood','Count','Pincode','Latitude','Longitude']]

df3
```

]:

	Neighbourhood	Count	Pincode	Latitude	Longitude
0	BAYVIEW	14303	94124	37.7309	-122.3886
1	CENTRAL	17666	94104	37.7915	-122.4018
2	INGLESIDE	11594	94112	37.7195	-122.4411
3	MISSION	19503	94114	37.7587	-122.4330
4	NORTHERN	20100	94109	37.7917	-122.4186
5	PARK	8699	94117	37.7712	-122.4413
6	RICHMOND	8922	94121	37.7786	-122.4892

5. FourSquare API:

The foursquare API is used to fetch the list of venues that are close to the given latitude and the longitude. The API uses the client ID and the Client Secret to fetch the details.
The url is then used on a get request method to the API, the url contains the client id, client secret, version of the Foursquare, latitude and longitude of the location, radius to be considered around the location and the limit as to fetch how many venues around the location.
The response is then stored in the form of json object. The response can contain details of the venue such as name, latitude and longitude of the venue, category of the venue, or rating or tip of the venues.
The response can then be converted to a pandas data frame and then be used for further operations.

Snapshot of the result after converting to pandas data frame:

```
#venues dataframe for each location
print(s_venues.shape)
s_venues.head()
```

(177, 7)

27]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	BAYVIEW	37.7309	-122.3886	Bayview Hunters Point YMCA	37.731851	-122.389733	Gym
1	BAYVIEW	37.7309	-122.3886	Foodway Liquors	37.730519	-122.388617	Liquor Store
2	BAYVIEW	37.7309	-122.3886	Palou And Lane 23 44 Bus Stop	37.732858	-122.388903	Bus Station
3	CENTRAL	37.7915	-122.4018	Pushkin	37.790943	-122.403877	Russian Restaurant
4	CENTRAL	37.7915	-122.4018	Blue Bottle Coffee	37.791320	-122.400983	Coffee Shop

Methodology

The project aims at [giving the user a map that will provide the user with the details of the venues around a location and clusters neighborhood in San Francisco based on the crime rates and venues around it. The project also presents the user with a map that will show two nearest location to the venue selected by the user. This done in the project in the following steps:

Data Acquisition:

Data is acquired for the project as described in the previous section Data.

Data Preprocessing:

In this step the data from the csv file is processed so that the data is stored in the data frame with neighborhood, count as the columns. This is done by using the group by method provided by the pandas library.

The data frame after initial preprocessing looks like this:

	Neighbourhood	Count
0	BAYVIEW	14303
1	CENTRAL	17666
2	INGLESIDE	11594
3	MISSION	19503
4	NORTHERN	20100
5	PARK	8699

To this data frame then the postal codes of San Francisco is added as a column as shown:

```
df3['Pincode']=pincode
```

Pincodes of the neighbourhood added to the dataframe

```
df3
```

5]:

	Neighbourhood	Count	Pincode
0	BAYVIEW	14303	94124
1	CENTRAL	17666	94104
2	INGLESIDE	11594	94112
3	MISSION	19503	94114
4	NORTHERN	20100	94109
5	PARK	8699	94117
6	RICHMOND	8922	94121
7	SOUTHERN	28445	94105
8	TARAVAL	11325	94116
9	TENDERLOIN	9942	94102

Then, the latitude and the longitude is added to the data frame above as mentioned in the previous section Data under Geocoder.

After using the Foursquare API to fetch venue details near the neighborhoods, the response data is then processed to show Venue, Venue category, Venue Latitude and Longitude to the data frame alongside the Neighborhood it is.

The data frame is then converted to a new data frame with the one hot encoding techniques for Venue categories, that shows whether a category is present at a location or not.

	Neighbourhood	Adult Boutique	American Restaurant	Art Gallery	Art Museum	Asian Restaurant	Bakery	Bar	Beer Bar	Boxing Gym	Breakfast Spot	Bubl T Sh
0	BAYVIEW	0	0	0	0	0	0	0	0	0	0	
1	BAYVIEW	0	0	0	0	0	0	0	0	0	0	
2	BAYVIEW	0	0	0	0	0	0	0	0	0	0	
3	CENTRAL	0	0	0	0	0	0	0	0	0	0	
4	CENTRAL	0	0	0	0	0	0	0	0	0	0	

Then the given dataset is scaled using the mean() function, this can be used to get the frequency of a category at a given location.

The crime rate is also scaled using the minmax_scale library from preprocessing module.

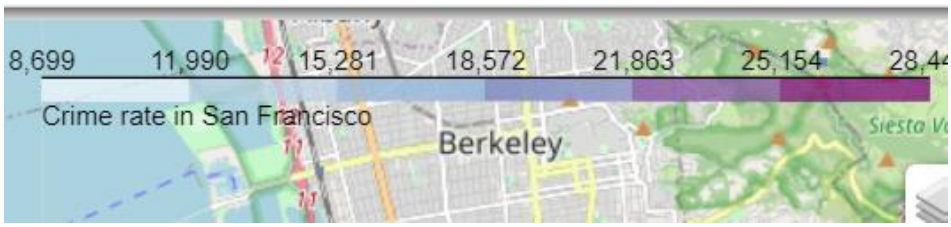
Below figure shows the result of the given step

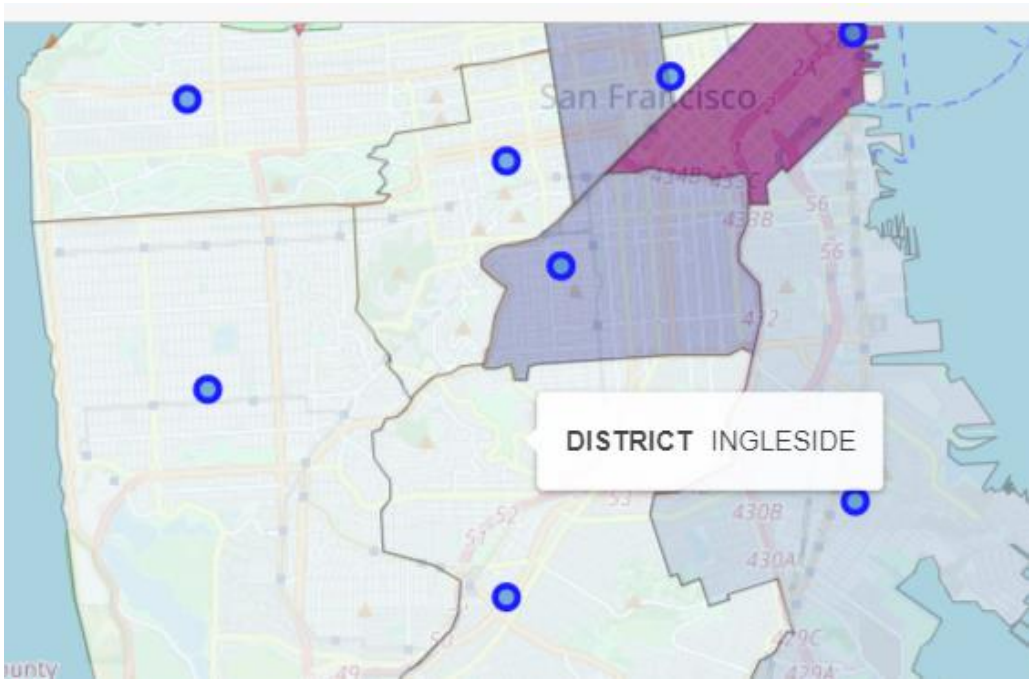
	Neighbourhood	Crime rate	Adult Boutique	American Restaurant	Art Gallery	Art Museum	Asian Restaurant	Bakery	Bar	Beer Bar
0	BAYVIEW	0.283804	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000
1	CENTRAL	0.454117	0.000000	0.000000	0.00	0.000000	0.000000	0.021739	0.000000	0.000000
2	INGLESIDE	0.146612	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000
3	MISSION	0.547149	0.000000	0.000000	0.05	0.000000	0.000000	0.000000	0.000000	0.000000
4	NORTHERN	0.577383	0.027778	0.027778	0.00	0.000000	0.000000	0.027778	0.111111	0.027778
5	PARK	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000
6	RICHMOND	0.011293	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.200000	0.000000
7	SOUTHERN	1.000000	0.000000	0.071429	0.00	0.000000	0.000000	0.000000	0.000000	0.000000

This data is then used to perform machine learning algorithm to find similar neighborhoods.

Data Visualization:

The project makes use of folium package to present to the user maps that can give better insights to the user.





The map shows the crime rate and with the various neighborhoods marked.

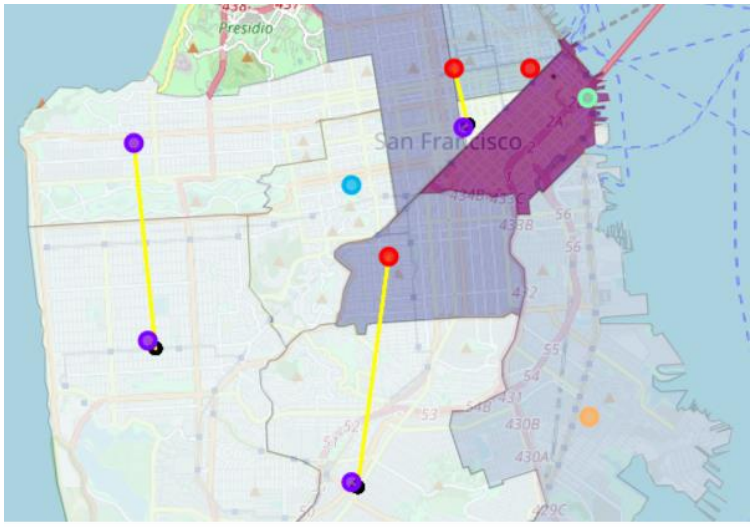
The user are also presented with the common venues for all the neighborhoods

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
0	BAYVIEW	Gym	Liquor Store	Bus Station	Deli / Bodega	Diner	Dive Bar
1	CENTRAL	Coffee Shop	Gym	Food Truck	Juice Bar	Japanese Restaurant	Restaurant
2	INGLESIDE	Grocery Store	Thai Restaurant	Cosmetics Shop	Chinese Restaurant	Furniture / Home Store	Filipino Restaurant
3	MISSION	Convenience Store	Coffee Shop	Thai Restaurant	Clothing Store	Scenic Lookout	Grocery Store
4	NORTHERN	Bar	Gym	Diner	Vietnamese Restaurant	Massage Studio	Mediterranean Restaurant

The user are also presented with the venue distance to the closest neighborhood, where the venue is entered by the user himself.

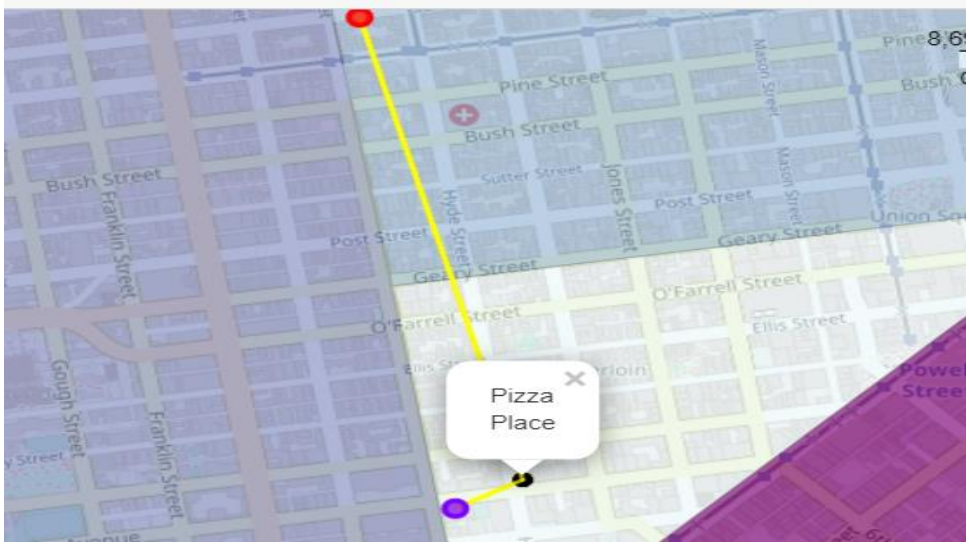
	Neighbourhood	N latitude	N longitude	Venue	V Latitude	V Longitude	Distance to the venue
29	TENDERLOIN	37.7813	-122.4167	Himalayan Pizza and Momo	37.781925	-122.415360	0.136784
2	INGLESIDE	37.7195	-122.4411	Little Joe's Pizza	37.718478	-122.439856	0.157805
18	TARAVAL	37.7441	-122.4863	Eagle Pizzeria	37.742719	-122.484575	0.215905
24	NORTHERN	37.7917	-122.4186	Himalayan Pizza and Momo	37.781925	-122.415360	1.123956
21	CENTRAL	37.7915	-122.4018	Himalayan Pizza and Momo	37.781925	-122.415360	1.598472
27	SOUTHERN	37.7864	-122.3892	Himalayan Pizza and Momo	37.781925	-122.415360	2.352916
25	PARK	37.7712	-122.4413	Himalayan Pizza and Momo	37.781925	-122.415360	2.573724
23	MISSION	37.7587	-122.4330	Himalayan Pizza and Momo	37.781925	-122.415360	3.013146
16	RICHMOND	37.7786	-122.4892	Eagle Pizzeria	37.742719	-122.484575	4.011685
3	MISSION	37.7587	122.4330	Little Joe's Pizza	37.718478	122.439856	4.514349

The latitudes and the longitudes of the neighborhoods and the venues are used to mark the distance between them on the map.



The different color marks used are for the clusters into which the neighborhoods are clustered using the venue categories and the crime rate.

A zoomed version of the above image shows that the map marks two nearest neighborhood to the venue along with the venue marked.



Machine Learning:

K-Means clustering is used to cluster the neighborhoods. Since the label in which the neighborhoods are not known classification algorithm cannot be used. K-Means clustering algorithm makes clusters of data points which are similar to each other. The aim of the proposed system also is to give users the similar neighborhoods based on crime rates and venues.

Therefore, a data frame that contains the one hot encoding of the venue categories and the crime rates along with the neighborhood is given to the K-Mean clustering algorithm, which then clusters the data into five clusters.

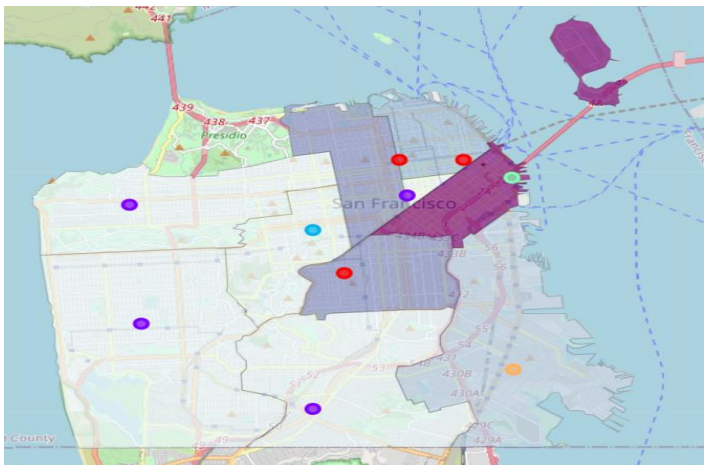
Below figure is the input provided to the algorithm.

	Neighbourhood	Crime rate	Adult Boutique	American Restaurant	Art Gallery	Art Museum	Asian Restaurant	Bakery	Bar	Beer Bar	Boxing Gym	Breakfas Spo
0	BAYVIEW	0.283804	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
1	CENTRAL	0.454117	0.000000	0.000000	0.00	0.000000	0.000000	0.021739	0.000000	0.000000	0.021739	0.02173
2	INGLESIDE	0.146612	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
3	MISSION	0.547149	0.000000	0.000000	0.05	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
4	NORTHERN	0.577383	0.027778	0.027778	0.00	0.000000	0.000000	0.027778	0.111111	0.027778	0.000000	0.00000
5	PARK	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
6	RICHMOND	0.011293	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.200000	0.000000	0.000000	0.00000
7	SOUTHERN	1.000000	0.000000	0.071429	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
8	TARAVAL	0.132989	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
9	TENDERLOIN	0.062949	0.000000	0.064516	0.00	0.032258	0.032258	0.000000	0.032258	0.032258	0.000000	0.00000

The output of the algorithm gives cluster label to each neighborhood which is then added to the data frame.

	Neighbourhood	Count	Pincode	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	
0	BAYVIEW	14303	94124	37.7309	-122.3886	1	Gym	Liquor Store	Bus Station	Deli / Bodega	Diner	
1	CENTRAL	17666	94104	37.7915	-122.4018	0	Coffee Shop	Gym	Food Truck	Juice Bar	Japanese Restaurant	F
2	INGLESIDE	11594	94112	37.7195	-122.4411	1	Grocery Store	Thai Restaurant	Cosmetics Shop	Chinese Restaurant	Furniture / Home Store	F
3	MISSION	19502	94114	37.7587	-122.4330	0	Convenience	Coffee Shop	Thai	Clothing Store	Scenic	Gro

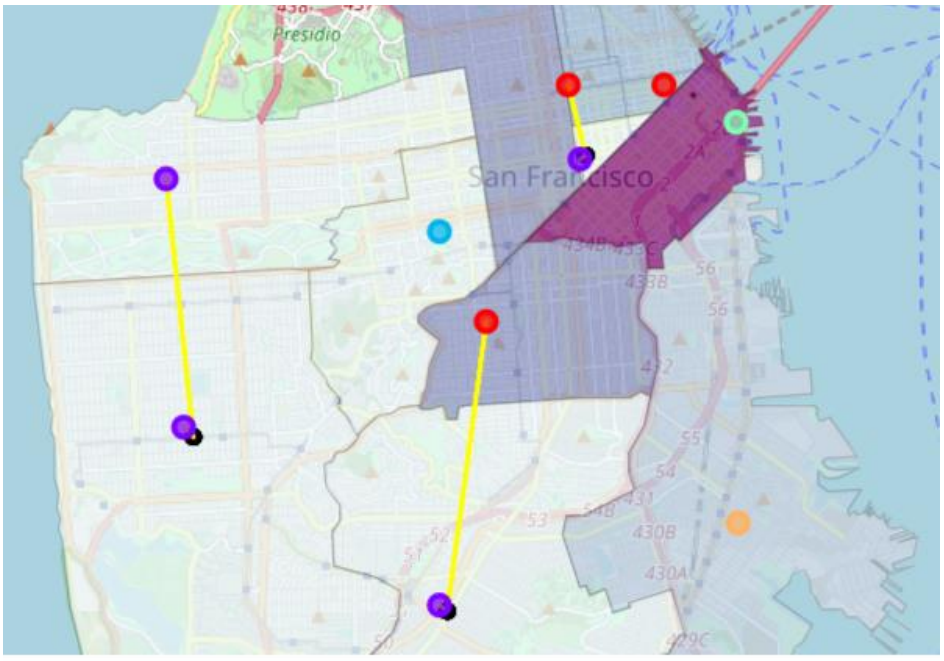
These clusters are then marked with different colors on the map.

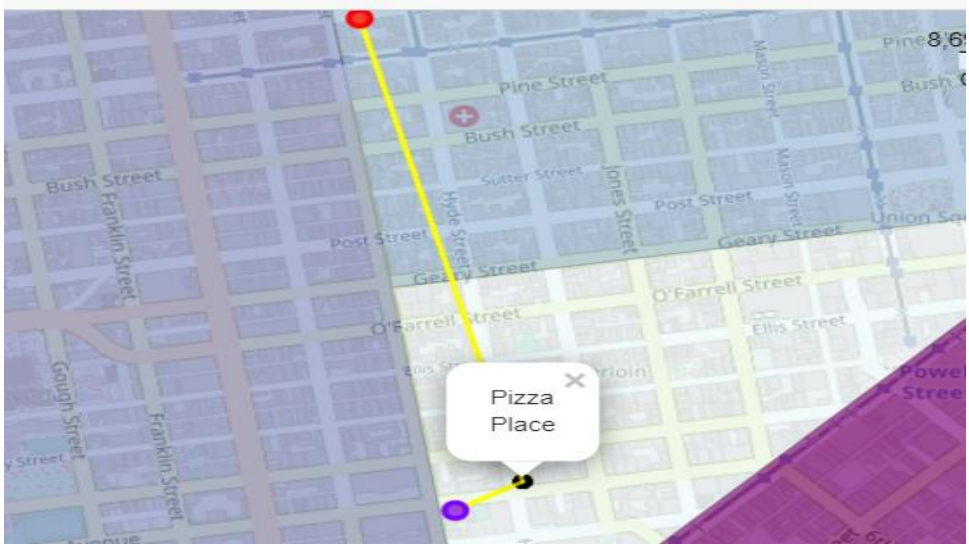


Results

The results that are shown to the user can then used by him/her to make a decision.

The venue was selected as Pizza Place by the user.





All the pizza Places are marked on the map and two of the nearest neighborhood(including the one in which it is located) are also marked.

Discussion

The results show the two nearest neighborhood of the venue that was entered by the user. The common venues are also listed for each of the of the neighborhood.

For, example if the user is more into fitness, then Northern region of San Francisco is apt for the user since gyms and fitness centers are quite common in that region, if the user is into cooking or is a home-maker then Ingleside and Mission are the region that have grocery stores and convenience stores as the most common venues.

Although, the figuring out the best neighborhood is left to the user, in the future scope there can be a section ahead which can directly state neighborhoods in the most favorable order.

Conclusion

Thus, the proposed system successfully clustered the neighborhoods of San Francisco based on attributes such as crime rate, venues around it and shows it to the user in the form of a map. The map can then be used by the user to select the neighborhood in which to move in.