# Clustering and Comparing the Neighborhoods of San Francisco and Chicago cities

**Abdalrhman Abdalla**

July 13, 2020

IBM Data Science Professional Certificate

(Applied Capstone Project)

## 1. Introduction and Problem Statement

The problem to be addressed in this project is for my friend who has a job offer in the United State. Upon his request, I will not mention his real name let's, call him Boss. Boss lives in the Middle East and he works in the tech sector and he got a job offer from the same company in the US. Boss is still confused which city he is going to reside on since he can work remotely, he has two major cities in his mind, they are San Francisco, CA and Chicago, IL and he would like to know which neighborhoods are better and meet his living style, Boss habits and requirements are below :

- He goes to the gym daily (GYM is a big factor).
- He is addicted to coffee (Café is also a big factor).
- He likes different types of food but Indian and Italian are his favorite.
- He wants to be near to parks.
- He wants a clean and well-maintained neighborhood.

*Figure 1 (Chicago, IL)*

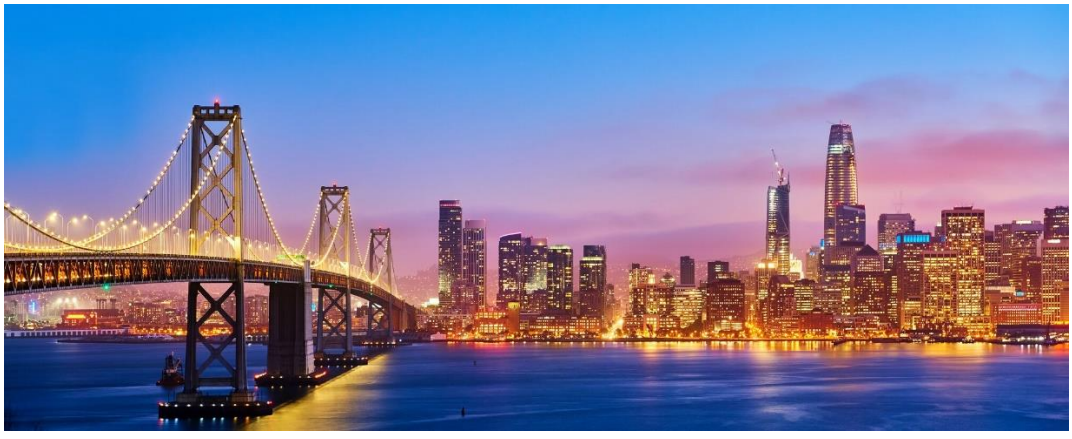

*Figure 2 (San Francisco, CA)*

## 2. Data acquisition and cleaning

- **Data Source**

A Neighborhood list of the two cities San Francisco and Chicago is available online in different website and can be found in the following links:

**San Francisco** http://www.healthysf.org/bdi/outcomes/zipmap.htm

**Chicago** http://www.lizshomes.com/resource_zips.html

**Foursquare API's** has a database of over 105 million places worldwide and will be consulted for this project. We'll use the Venue Recommendation to explore the two cities, which returns a list of recommended venues near a specific location. To make the query in the Foursquare API we need the coordinates given for a given neighborhood in Latitude and Longitude. So, the first thing I need to do is to get coordinates in each of the cities for each neighborhood. To get the coordinates of given a city name and neighborhoods I used Geocoder. Geocoder is a geocoding library very easy to use written in python that finds locations (latitude, longitude) of addresses, or zip codes.

- **Cleansing**

The data for San Francisco was scraped from the healthysf website with the list of neighborhoods and their cross-bonding zip codes for each neighborhood. On the other hand, the data for Chicago neighborhoods were copied from lizshomes website to an excel file and then converted to dataframe after altering a few the neighborhood names which have multiple zip codes, and to consult nearby venues in each neighborhood, the coordinates were added to the two dataframes. So, in the end, we end up with a dataframes like those below:

| | Zip Code | Neighborhood | Latitude | Longitude |
|---|---|---|---|---|
| 1 | 94102 | Hayes Valley/Tenderloin/North of Market | 37.777015 | -122.421875 |
| 2 | 94103 | South of Market | 37.772000 | -122.408735 |
| 3 | 94107 | Potrero Hill | 37.760651 | -122.394064 |
| 4 | 94108 | Chinatown | 37.791775 | -122.407440 |
| 5 | 94109 | Polk/Russian Hill (Nob Hill) | 37.790105 | -122.420590 |

*Figure 3 (San Francisco)*

| | Zip Code | Neighborhood | Latitude | Longitude |
|---|---|---|---|---|
| 0 | 60601 | Loop1 | 41.886255 | -87.622310 |
| 1 | 60602 | Loop2 | 41.883250 | -87.630795 |
| 2 | 60603 | Loop3 | 41.880890 | -87.621270 |
| 3 | 60604 | Loop | 41.878160 | -87.631010 |
| 4 | 60605 | Loop, Near South Side | 41.869255 | -87.626255 |

*Figure 4 (Chicago)*

Data that describes neighborhoods and the categories of those venues are needed for each city. Data on the venues will be obtained from Foursquare, a common source of venues and locations data. Foursquare API can be used for viewing and downloading data. To retrieve data from

Foursquare using their API, a URL should be prepared and used to request data related to a specific location. An example URL is the following:

https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}

where search indicates the API endpoint used, client_id and client_secret are credentials used to access the API service and are obtained when registering a Foursquare developer account, v indicates the API version to use, ll indicates the latitude and longitude of the desired location, radius is the maximum distance in meters between the specified location and the retrieved venues, and limit is used to limit the number of returned results if necessary.

Figure 5 shows the code used to design a function that uses the names, latitudes, and longitudes of the neighborhoods as inputs, and returns a dataframe with information about each neighborhood and its venues. Creates an API URL for each neighborhood and retrieves data for each neighborhood.

```
def getNearbyVenues(names, latitudes, longitudes, radius=500,LIMIT=100):

    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={}'.forr
            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)
```

*Figure 5*

Using the function in Figure 5 with the two cities' neighborhood data, for each venue, venue name, category, latitude, and longitude were retrieved.

The data frame head returned by the function shown in Figure 5. We can see that every row in the data frame contains data on one venue: the venue name, coordinates (latitude and longitude), and category, in addition to the neighborhood where the venue is located and the neighborhood coordinates.

Figure 6 & 7 shows the head of the two dataframes for San Francisco and Chicago respectively after the retrieval

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Hayes Valley/Tenderloin/North of Market | 37.777015 | -122.421875 | SFJazz Center | 37.776350 | -122.421539 | Jazz Club |
| 1 | Hayes Valley/Tenderloin/North of Market | 37.777015 | -122.421875 | Blue Bottle Coffee | 37.776430 | -122.423224 | Coffee Shop |
| 2 | Hayes Valley/Tenderloin/North of Market | 37.777015 | -122.421875 | Louise M. Davies Symphony Hall | 37.777976 | -122.420157 | Concert Hall |
| 3 | Hayes Valley/Tenderloin/North of Market | 37.777015 | -122.421875 | Linden Alley | 37.776329 | -122.423594 | Pedestrian Plaza |
| 4 | Hayes Valley/Tenderloin/North of Market | 37.777015 | -122.421875 | Fatted Calf | 37.775935 | -122.423146 | Butcher |

*Figure 6*

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Loop1 | 41.886255 | -87.62231 | Chicago Architecture Center | 41.887720 | -87.623650 | Tour Provider |
| 1 | Loop1 | 41.886255 | -87.62231 | Roti Modern Mediterranean | 41.886048 | -87.624948 | Mediterranean Restaurant |
| 2 | Loop1 | 41.886255 | -87.62231 | Wildberry Pancakes & Cafe | 41.884412 | -87.623047 | Breakfast Spot |
| 3 | Loop1 | 41.886255 | -87.62231 | Giordano's | 41.885130 | -87.623761 | Pizza Place |
| 4 | Loop1 | 41.886255 | -87.62231 | St. Jane Chicago | 41.886573 | -87.624902 | Hotel |

*Figure 7*

## 3. Exploratory Data Analysis

Throughout this section, the datasets generated in the previous section will be discussed through effective visualizations to better understand the data.

- **Most Common Venue Categories**

What are the categories that have more venues than the others in both cities? To answer this question, the number of occurrences for each category of the venue is shown in Figure 8 (San Francisco) and Figure 9 (Chicago).

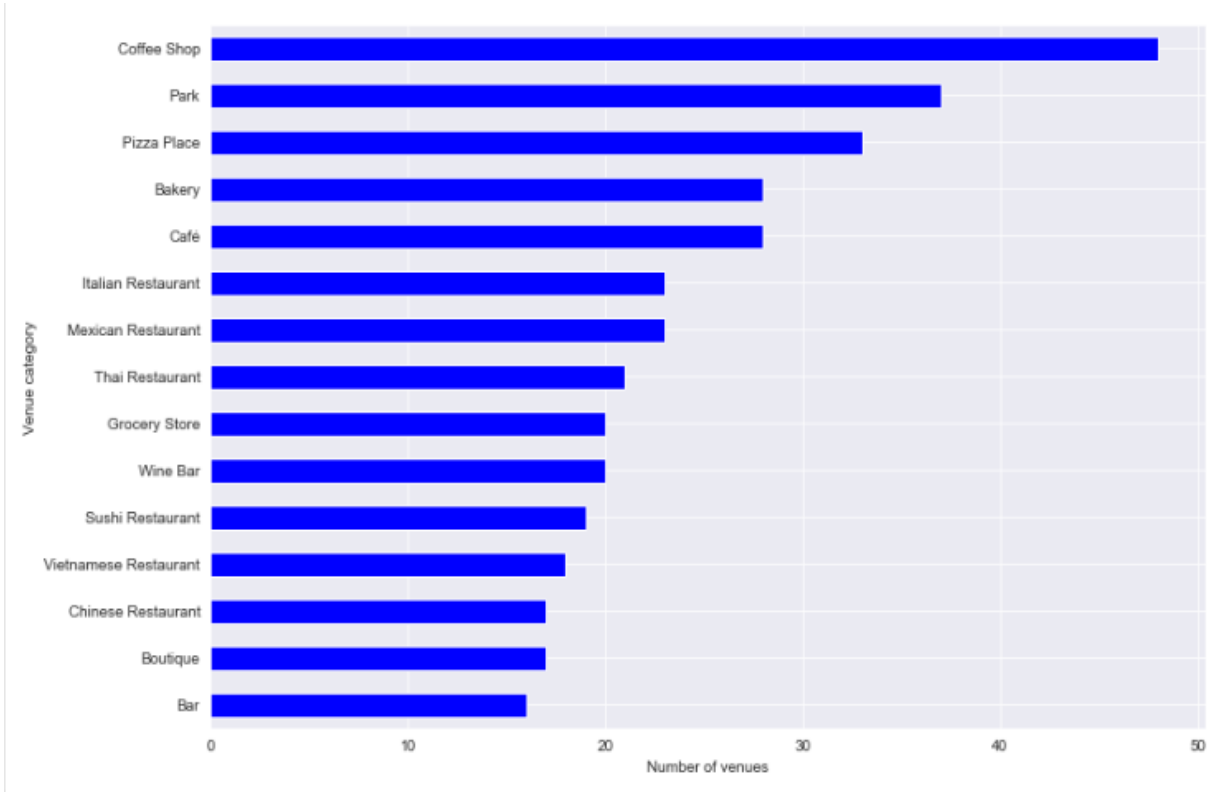


*Figure 8*

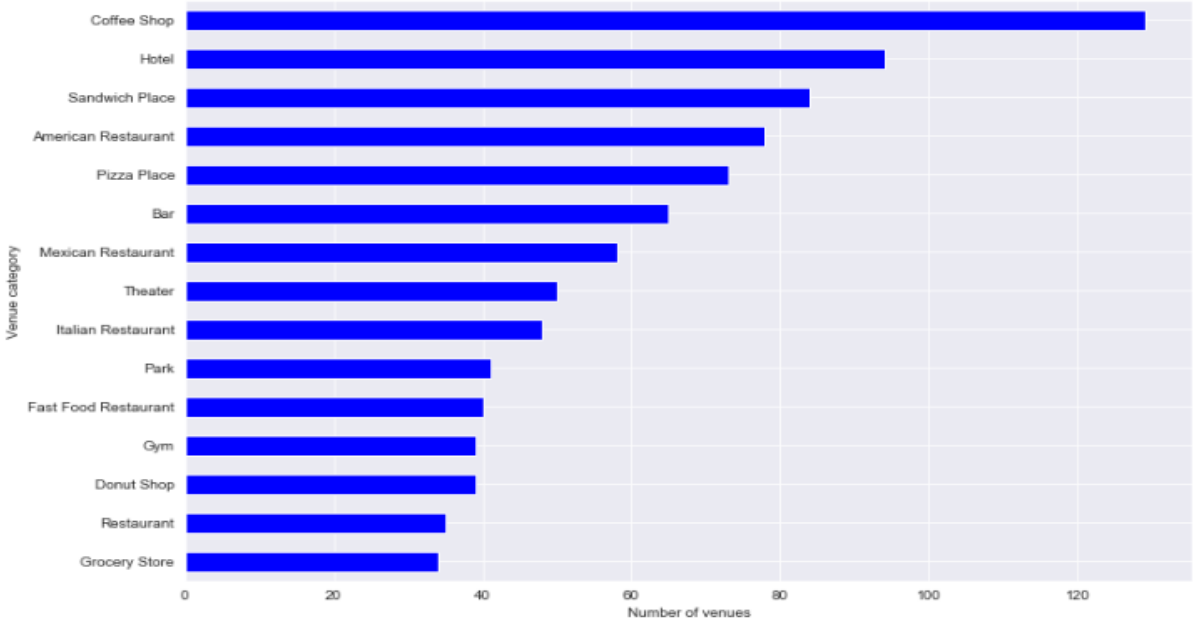


*Figure 9*

Figure 8 displays a bar plot of the most common venues in San Francisco. We can see that the most common category is "Coffee Shop" with ~50 venues. The category "Park" appears in the second rank with ~40 venues. The "Pizza Place" category falls in third place.

Figure 9 displays a bar plot of the most common venues in Chicago. We can see that the most common category is also "Coffee Shop" with ~130 venues. The category "Hotel" appears in the second rank with ~100 venues. The "Sandwich Place" category falls in the third row.

- **Most Widespread Venue Categories**

Now another question needs to be answered: What are the venue categories that occur in certain neighborhoods? This situation is different from the one stated earlier. To illustrate the disparity with an example, suppose that there are 20 venues in the category "Italian Restaurant" and that these venues exist in only 10 neighborhoods out of 100 neighborhoods; also suppose that there are 15 venues in the category "Indian Restaurant" and that these venues exist in 15 neighborhoods each in a different neighborhood. It can then be argued that the category "Italian Restaurant" is more common than the category "Indian Restaurant" because there are more venues in this category, and it can be assumed that the category "Indian Restaurant" is more widespread than the category "Italian Restaurant" because the venues in this category operate in more neighborhoods than the other category.

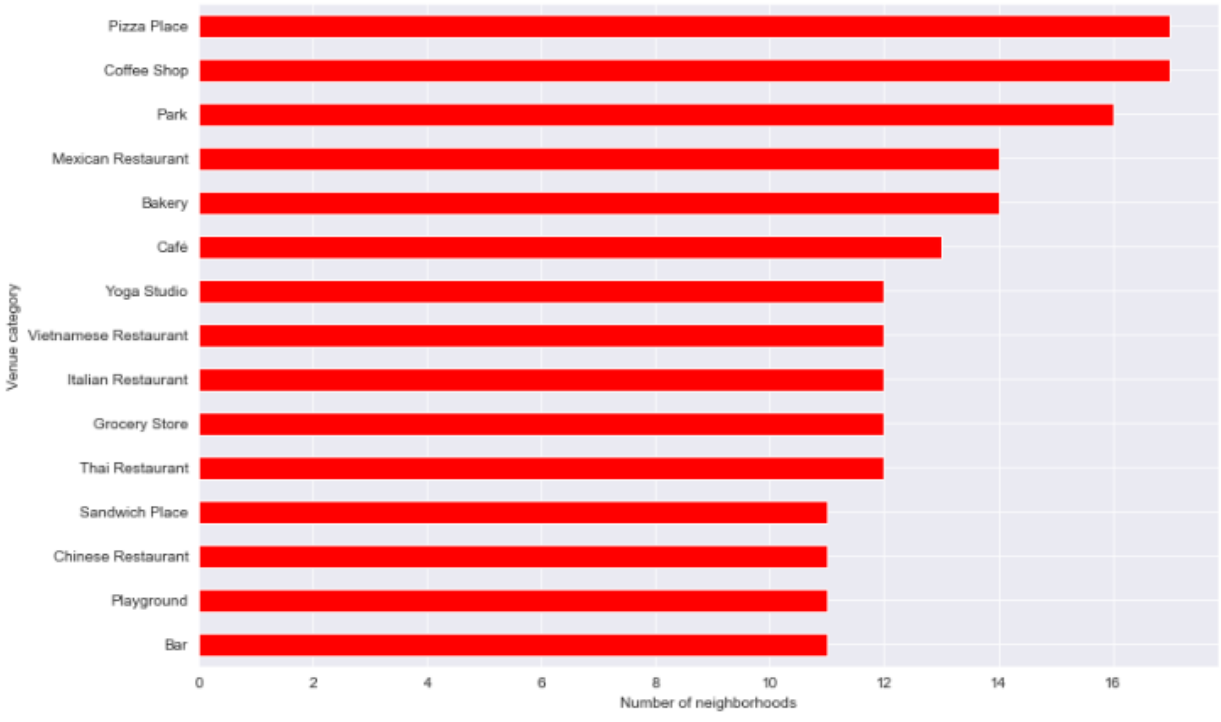


*Figure 10*

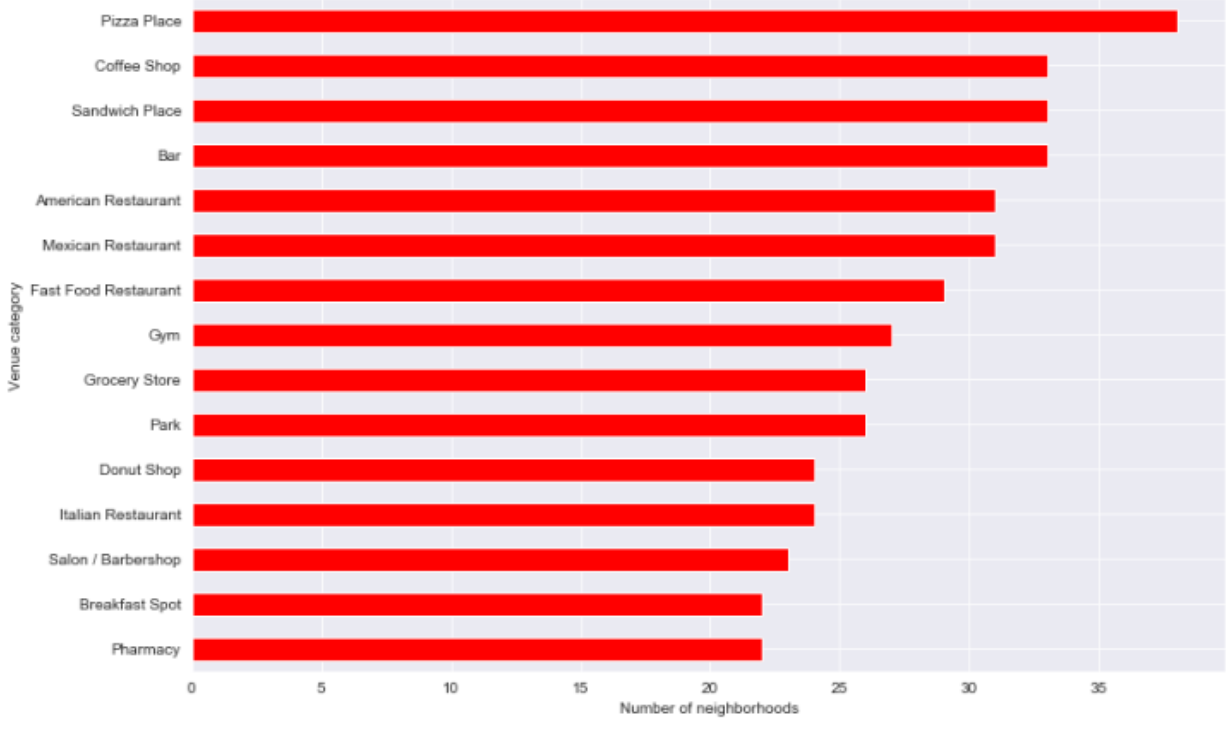


*Figure 11*

Figure 10 shows the most widespread categories of venues in **San Francisco**. It can be seen that this time the order of categories is different from that of the most common categories (Figure 8). The most widespread category is "Pizza Place" which exists in ~18 neighborhoods out of 21 neighborhoods. After that comes the group "Coffee Shop" with venues in ~18 neighborhoods. In the third place the category "Park" with venues in ~16 neighborhoods.

Figure 11 shows the most widespread categories of venues in **Chicago**. It can be seen that this time the order of categories is different from that of the most common categories (Figure 9). The most widespread category is "Pizza Place" which exists in ~38 neighborhoods out of 59 neighborhoods. After that come the categories "Coffee Shop", "Sandwich Place" and "Bar" with venues in ~33 neighborhoods for each.

## 4. Methodology

- **Clustering of Neighborhoods**

In this section, clustering in the San Francisco and Chicago neighborhoods will be used to identify similar neighborhoods in both cities. Clustering is the method of finding related objects in a data set based on the characteristics (features) of the dataset objects. K-means the Scikit-learn Python library clustering algorithm will be used. A dataset suitable for clustering is required to be able to perform clustering; the datasets presented in Figure 6 and Figure 7 are not ready to be used with clustering algorithms.

- **Features Selection**

The purpose of clustering is to group neighborhoods based on the similarity of venue categories in the neighborhoods. It means that the neighborhood and the venue categories in the neighborhood are the two factors of importance here. The following two features will, therefore,

be selected from the data frames of Figure 6 and Figure 7: "Neighborhood" and "Venue Category." But even then, the data is not ready for the clustering algorithm because the algorithm works with numerical features.

To do this, **one-hot encoding** will be applied to the "Venue Category" feature and the encoding result will be used for clustering. One-hot encoding will be applied to data from Sand Francisco and Chicago, and then, as explained later, the data from both cities will be merged.

After applying one-hot encoding to the two dataframes, the resulting dataframes look like the two shown in Figure 12 (San Francisco), Figure 13 (Chicago).

| | Neighborhood | ATM | Accessories Store | Adult Boutique | African Restaurant | Alternative Healer | American Restaurant | Arcade | Argentinian Restaurant | Art Gallery | ... | Tuscan Restaurant | Udon Restaurant | Vegetarian / Vegan Restauran |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Hayes Valley/Tenderloin/North of Market | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | |
| 1 | Hayes Valley/Tenderloin/North of Market | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | |
| 2 | Hayes Valley/Tenderloin/North of Market | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | |
| 3 | Hayes Valley/Tenderloin/North of Market | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | |
| 4 | Hayes Valley/Tenderloin/North of Market | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | |

5 rows × 237 columns

*Figure 12*

| | Neighborhood | ATM | Accessories Store | Adult Boutique | African Restaurant | American Restaurant | Amphitheater | Antique Shop | Arcade | Arepa Restaurant | ... | Vietnamese Restaurant | Vineyard | Waterfront | Whi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Loop1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | |
| 1 | Loop1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | |
| 2 | Loop1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | |
| 3 | Loop1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | |
| 4 | Loop1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | |

5 rows × 275 columns

*Figure 13*

The next step is to aggregate the values for each neighborhood so that there is just one row in each neighborhood. The aggregation will be achieved by grouping rows by neighborhood and taking the mean frequency of occurrence of each category.

Figure 14 shows how the aggregated dataframe for San Francisco looks like and Figure 15 for Chicago.

| | Neighborhood | ATM | Accessories Store | Adult Boutique | African Restaurant | Alternative Healer | American Restaurant | Arcade | Argentinian Restaurant | Art Gallery | ... | Tuscan Restaurant | Udon Restaurant | Vegeta / Ve Restau |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Bayview-Hunters Point | 0.0 | 0.000000 | 0.000000 | 0.047619 | 0.0 | 0.000000 | 0.000000 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.000 |
| 1 | Castro/Noe Valley | 0.0 | 0.000000 | 0.012658 | 0.000000 | 0.0 | 0.012658 | 0.000000 | 0.0 | 0.012658 | ... | 0.0 | 0.0 | 0.000 |
| 2 | Chinatown | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.034884 | 0.000000 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.011 |
| 3 | Haight-Ashbury | 0.0 | 0.024691 | 0.000000 | 0.000000 | 0.0 | 0.012346 | 0.012346 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.012 |
| 4 | Hayes Valley/Tenderloin/North of Market | 0.0 | 0.010000 | 0.000000 | 0.000000 | 0.0 | 0.010000 | 0.000000 | 0.0 | 0.000000 | ... | 0.0 | 0.0 | 0.000 |

5 rows × 237 columns

*Figure 14*

| | Neighborhood | ATM | Accessories Store | Adult Boutique | African Restaurant | American Restaurant | Amphitheater | Antique Shop | Arcade | Arepa Restaurant | ... | Vietnamese Restaurant | Vineyard | Waterfront | Whi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Albany Park, Forest Glen, Irving Park, Jeffers... | 0.0 | 0.0 | 0.0 | 0.0 | 0.043478 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | |
| 1 | Albany Park, Lincoln Square, North Park | 0.0 | 0.0 | 0.0 | 0.0 | 0.021739 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | |
| 2 | Archer Heights, Brighton Park, Gage Park, Garf... | 0.0 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | |
| 3 | Armour Square, Bridgeport, Douglas, Fuller Par... | 0.0 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | |
| 4 | Armour Square, Bridgeport, Douglas,Lower West ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.013333 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | |

5 rows × 275 columns

*Figure 15*

Upon generating aggregated dataframes for both San Francisco and Chicago, these dataframes will be merged before implementing the clustering algorithm. However, to differentiate San Francisco neighborhoods from Chicago neighborhoods in the new dataframe, a text string is inserted at the end of each neighborhood name before combining the dataframes: for San Francisco, the string to be inserted is "_SF" and "_CH" for Chicago.

Both, San Francisco and Chicago do not have the same venue categories (i.e. some columns in the data frame of Figure 14 do not appear in the data frame of Figure 15 and vice versa). To address this problem when merging dataframes, the columns of both dataframes are made the same by adding columns that occur only in the San Francisco dataframe to the Chicago dataframe and vice versa; the newly introduced columns have a value of 0 for all rows.

Figure 16 displays a part of the data frame resulting from the integration of San Francisco and Chicago aggregated data frames. This dataframe contains data on 80 neighborhoods in both San Francisco and Chicago.

| | Neighborhood | ATM | Accessories Store | Adult Boutique | African Restaurant | Alternative Healer | American Restaurant | Arcade | Argentinian Restaurant | Art Gallery | ... | Hobby Shop | Candy Store | Turkish Restaurant |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Bayview-Hunters Point_SF | 0.0 | 0.000000 | 0.000000 | 0.047619 | 0.0 | 0.000000 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| 1 | Castro/Noe Valley_SF | 0.0 | 0.000000 | 0.012658 | 0.000000 | 0.0 | 0.012658 | 0.000000 | 0.0 | 0.012658 | ... | 0.000000 | 0.0 | 0.0 |
| 2 | Chinatown_SF | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.034884 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| 3 | Haight-Ashbury_SF | 0.0 | 0.024691 | 0.000000 | 0.000000 | 0.0 | 0.012346 | 0.012346 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| 4 | Hayes Valley/Tenderloin/North of Market_SF | 0.0 | 0.010000 | 0.000000 | 0.000000 | 0.0 | 0.010000 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 75 | Pullman, Roseland, Washington Heights, West Pu... | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| 76 | Riverdale Chicago_CH | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.050505 | 0.010101 | 0.0 | 0.000000 | ... | 0.010101 | 0.0 | 0.0 |
| 77 | Rogers Park_CH | 0.0 | 0.000000 | 0.000000 | 0.021277 | 0.0 | 0.042553 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| 78 | South Shore_CH | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.041667 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |
| 79 | West Ridge_CH | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.111111 | 0.000000 | 0.0 | 0.000000 | ... | 0.000000 | 0.0 | 0.0 |

80 rows × 323 columns

*Figure 16*

Using the dataframe in Figure 16, another dataframe is generated to define the 10 most common categories for each neighborhood in San Francisco and Chicago. This data frame is generated by extracting the 10 categories with the highest values for each neighborhood in Figure 16. This data frame is seen in Figure 17.

| | 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_SF | Park | Southern / Soul Food Restaurant | Bakery | Fried Chicken Joint | Grocery Store | Thrift / Vintage Store | Theater | Latin American Restaurant | Market | Bus Station |
| 1 | Castro/Noe Valley_SF | Gay Bar | Coffee Shop | Park | Thai Restaurant | Scenic Lookout | Wine Bar | Playground | Cosmetics Shop | Seafood Restaurant | Indian Restaurant |
| 2 | Chinatown_SF | Hotel | Coffee Shop | Bakery | Bubble Tea Shop | Boutique | American Restaurant | Gym | Italian Restaurant | Sushi Restaurant | Restaurant |
| 3 | Haight-Ashbury_SF | Boutique | Coffee Shop | Clothing Store | Breakfast Spot | Shoe Store | Bookstore | Bus Station | Ice Cream Shop | Lingerie Store | Park |
| 4 | Hayes Valley/Tenderloin/North of Market_SF | Wine Bar | Boutique | New American Restaurant | Pizza Place | Clothing Store | Coffee Shop | Dessert Shop | Bakery | Food & Drink Shop | Burger Joint |

*Figure 17*

- **Clustering and Results**

The clustering algorithm can now be applied by obtaining the dataframe in Figure 16. Figure 18 displays the code used to execute clustering using the Scikit-learn Library K-means algorithm. The variable called sf_ch_grouped includes the dataframe in Figure 16. Note that the "Neighborhood" column was removed before implementing the clustering algorithm (i.e. the clustering algorithm was implemented to all columns except that column); this was achieved as the clustering algorithm does not allow non-numeric columns as stated above. However, this column will be re-added as will be explained soon.

```
kclusters = 8 #set the number of clusters
sf_ch_grouped_clustering = sf_ch_grouped.drop('Neighborhood', axis=1)
# run k-means clustering
sf_ch_kmeans = KMeans(n_clusters=kclusters, random_state=1).fit(sf_ch_grouped_clustering)
```

*Figure 18*

The clustering algorithm produced cluster-labels; these labels denote the cluster of each record (i.e. each neighborhood) in the data. Using these labels and the data frame of Figure 17, a data frame is built to display the neighborhoods of San Francisco and Chicago, the cluster to which each neighborhood corresponds, and the most frequent venue categories in each neighborhood. The data frame can be seen in Figure 19.

| Neighborhood | 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 Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bayview-Hunters Point_SF | 0 | Park | Southern / Soul Food Restaurant | Bakery | Fried Chicken Joint | Grocery Store | Thrift / Vintage Store | Theater | Latin American Restaurant | Market | Bus Station |
| Castro/Noe Valley_SF | 6 | Gay Bar | Coffee Shop | Park | Thai Restaurant | Scenic Lookout | Wine Bar | Playground | Cosmetics Shop | Seafood Restaurant | Indian Restaurant |
| Chinatown_SF | 3 | Hotel | Coffee Shop | Bakery | Bubble Tea Shop | Boutique | American Restaurant | Gym | Italian Restaurant | Sushi Restaurant | Restaurant |
| Haight-Ashbury_SF | 6 | Boutique | Coffee Shop | Clothing Store | Breakfast Spot | Shoe Store | Bookstore | Bus Station | Ice Cream Shop | Lingerie Store | Park |
| Hayes Valley/Tenderloin/North of Market_SF | 6 | Wine Bar | Boutique | New American Restaurant | Pizza Place | Clothing Store | Coffee Shop | Dessert Shop | Bakery | Food & Drink Shop | Burger Joint |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Pullman, Roseland, Washington Heights, West Pullman_CH | 7 | Fast Food Restaurant | Grocery Store | Sandwich Place | Electronics Store | Video Game Store | Bank | Chinese Restaurant | Gift Shop | Gourmet Shop | Indian Restaurant |
| Riverdale Chicago_CH | 3 | Coffee Shop | Hotel | Sandwich Place | Theater | American Restaurant | Donut Shop | Italian Restaurant | Restaurant | Department Store | Vegetarian / Vegan Restaurant |
| Rogers Park_CH | 6 | Mexican Restaurant | Pizza Place | Theater | Bus Station | American Restaurant | Bar | Mediterranean Restaurant | Bakery | Asian Restaurant | Dive Bar |
| South Shore_CH | 6 | Grocery Store | Pizza Place | Fried Chicken Joint | Cosmetics Shop | Seafood Restaurant | Bus Station | Food | Rental Car Location | Train Station | Record Shop |
| West Ridge_CH | 6 | Bar | American Restaurant | Bus Station | Asian Restaurant | Park | Gym | Currency Exchange | Mediterranean Restaurant | Salon / Barbershop | Supermarket |

80 rows × 11 columns

*Figure 19*

The output of the clustering operation is 8 clusters with 0, 1, 2, 3, 5,6, and 7 cluster labels. Each cluster is expected to contain a group of similar neighborhoods based on the categories of venues in each neighborhood. The clustering algorithm was used in 80 neighborhoods in San Francisco and Chicago. Figure 20 shows the number of neighborhoods within each cluster. Note that the labels from 0-7, so the first cluster labeled with 0 but I will refer to it as cluster1, and the same rule applied to the reset.

```
sf_ch_merged['Cluster Labels'].value_counts()
```

```
6    41
3    13
2    11
0     9
7     3
5     1
4     1
1     1
Name: Cluster Labels, dtype: int64
```

*Figure 20*

- **Clustering and Analysis**

The clustering algorithm divided neighborhoods of San Francisco and Chicago into 8 clusters based on the similarities of their positions, but cluster 2,8,5 and 6 have just one neighborhood for each except cluster 8 which has 3 neighborhoods and all of them don't have GYM, and because GYM is a major consideration for us, and they will be skipped. Now, the remaining clusters 1,3,4 and 7 will be examined to see the most suitable clusters for Boss to make his decision.

1. **Cluster1:** In this cluster, there are 9 neighborhoods, Figure 21 shows the top 10 venues categories and their distribution throughout the cluster1, so in this cluster, we can tell that Park category comes first with 0.9 which means there are 9 Parks in this cluster followed by Italian Restaurant category with 0.3 and same for Grocery Store, Fried Chicken Joint and Indie Theater categories

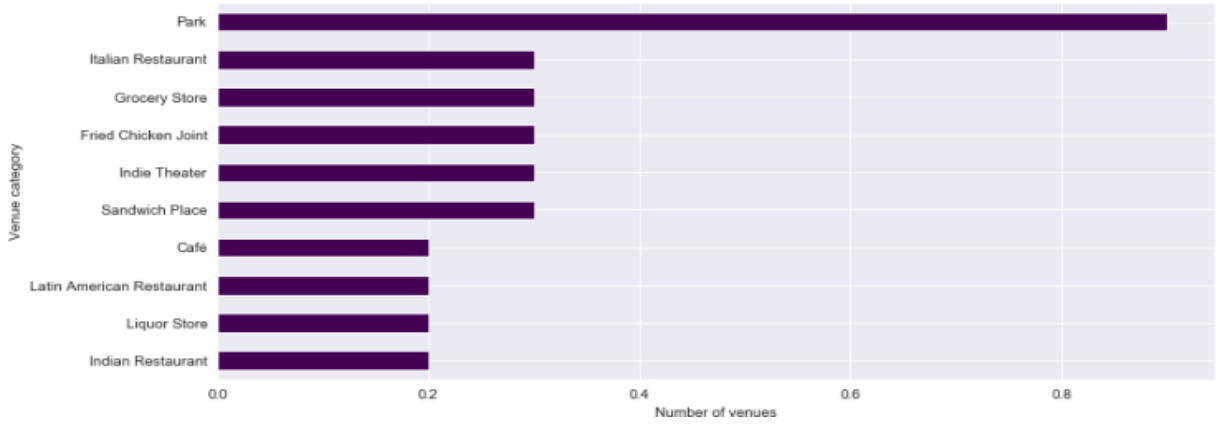


*Figure 21*

2. **Cluster3:** In this cluster, there are 11 neighborhoods, Figure 22 shows the top 10 venues categories and their distribution throughout the cluster1, so in this cluster, we can tell that Pizza Place category comes first with 0.9 followed by Mexican Restaurant category with 0.8 and for Liquor Store, Sandwich place categories 0.5

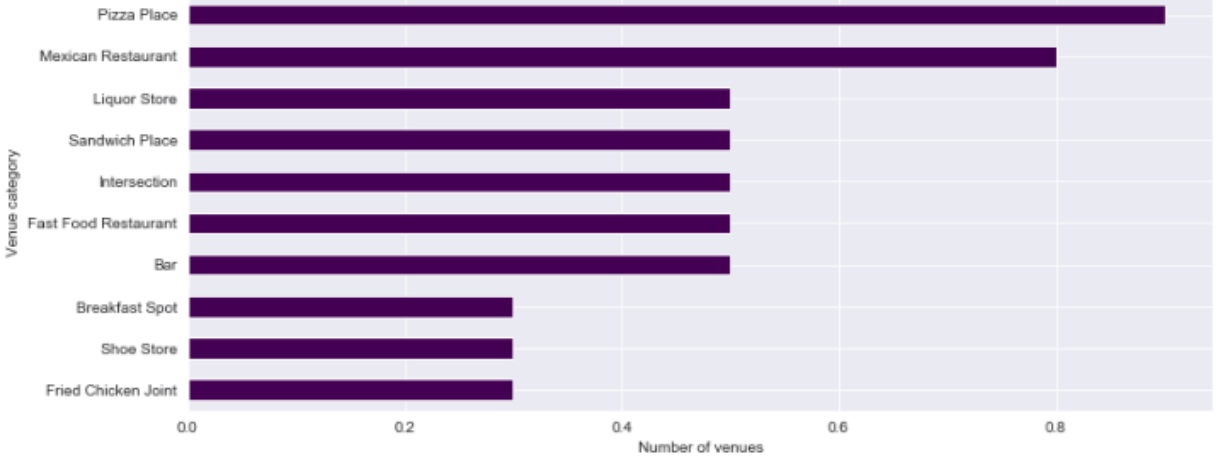


*Figure 22*

3. **Cluster4:** In this cluster, there are 13 neighborhoods, Figure 23 shows the top 10 venue categories and their distribution throughout the cluster1, so in this cluster, we can tell that American Restaurant category and Hotel category come first with 1.2 followed by Coffee Shop category with 1.1.

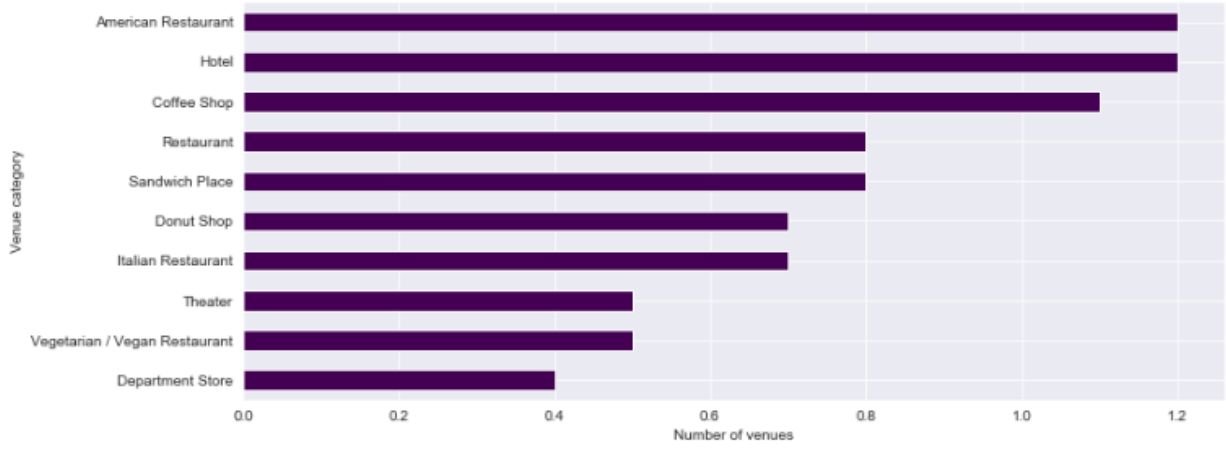


*Figure 23*

4. **Cluster7:** In this cluster, there are 41 neighborhoods, Figure 24 shows the top 10 venue categories and their distribution throughout cluster1, so in this cluster, we can tell that the Coffee Shop category comes first with 2.5 followed by Pizza Place category with 1.9 and Bar categories 1.8.

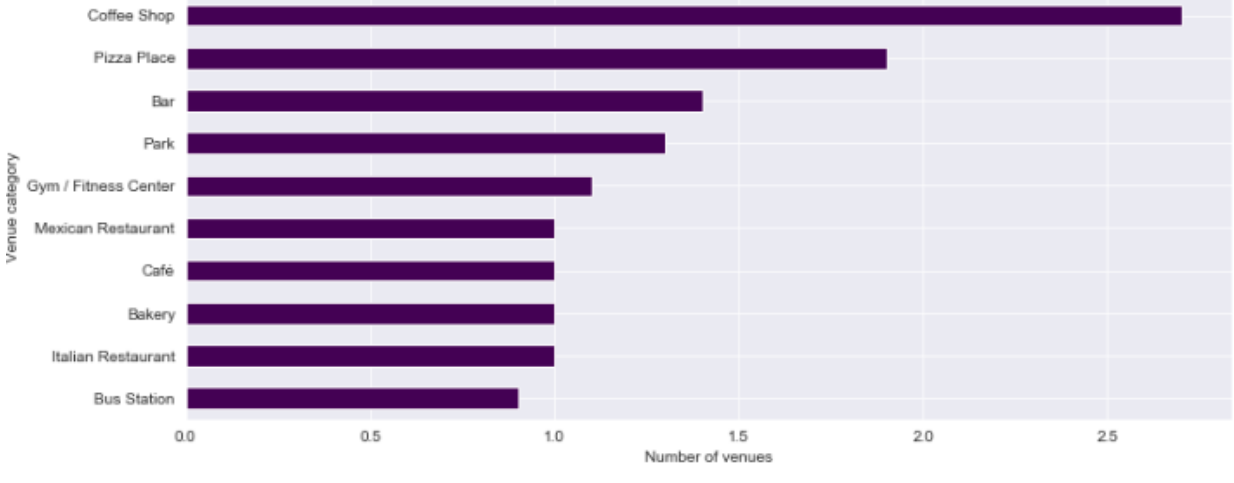


*Figure 24*

With a close look at all the clusters, cluster 7 contains most of the requirements that Boss asked for followed by cluster 4 considering it has Hotels and many hotels have gym especially when the area does not have, and then cluster 1,3 respectively.

## Conclusion

For this project, the neighborhoods of San Francisco City and Chicago City were grouped into multiple clusters based on the categories (types) of venues in these neighborhoods. The findings revealed that there are venue categories that are more common in some clusters than others; the most common venue categories vary from cluster to cluster. A deeper analysis — taking into account further aspects — has been carried out, and the result has been the discovery of different styles in each cluster depending on the criteria we were looking for.