**Introduction**

For people planning big moves, it would be extremely useful to compare their current location with the potential one.

They can evaluate the accessibility to grocery stores, gyms, doctors, and more. Utilizing the four square to compare neighborhoods would allow a user to either match a cluster’s characteristics to their current one, or find a different kind of neighborhood to live in. Say a couple is planning a move and is planning to have children in the next few years. They make be looking to move from a Downtown vibe with lots of restaurants, coffee shops, etc, to a neighborhood with parks, libraries and schools. They are probably cognizant of appropriate areas in their location and can compare clusters that represent that lifestyle in their new city.

Target User:

Professionals and families planning a move out of their current town into another within the United States.

**Data Cleaning**

Data:

The source data for this project is a combination of neighborhood indicators from City-Data.com in addition to venue data provided through the Four Square API Developer Portal.

Using the python package, Beautiful Soup, specific html tags are scraped, providing the Neighborhood name, its Median Income, Population Density, and Median Rent.

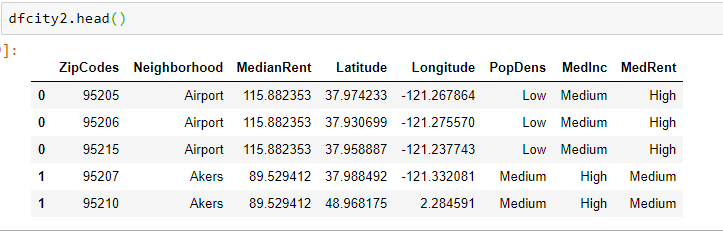
Population Density is directly comparable in the two cities. However, the assumption must be made that the cost of living is different in the compared cities. To combat this, Median Income and Median Rent are adjusted in proportion to where they compare to the whole cities value.

The equations:

The Population Density, Median Rent and Median Income are then converted to categorical variables utilizing the follow boundaries:

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Low | Medium | High |
| Population Density | <=4000 people/sq. mile | >4000 people/sq. mile & <8000 | >10000 people/sq. mile |
| Median Rent | <=75% of City Median | 75% to 100% of City Median | Over 100% of City Median |
| Median Income | <=75% of City Median | 75% to 100% of City Median | Over 100% of City Median |

After scraping the relevant data and storing them in lists to gather the data for each neighborhood. A zipped tuple of the lists is used to create a Pandas data frame. GeoPy Geocoding is used to obtain the neighborhood latitudes and longitudes in batch, then added to the data frame. The resulting data frame is as below:



For direct viewing of the code used to create the above table, see the Capstone Notebook.

This process is completed for city one and city 2. In combination with the foursquare most common venues, the cluster will be created based on the tale above.

Categorical values are given dummy values to provide one hot encoded values for processing. The 10 most common attributes are retrieved.

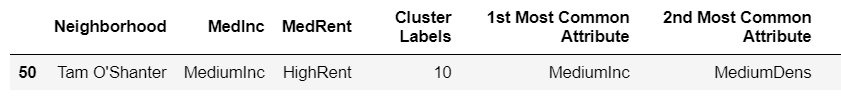
**Clustering**

Using the KMeans package from the sklearn.cluster, clusters will be created according to the most common venues in a neighborhood and the ranking of Population Density, Median Income, and Housing prices.

For comparison purposes, clusters were created and observed using only the Foursquare venue features. Using a city where attributes of a neighborhood were well understood allows an idea of the appropriateness of the created clusters. For the data only based on venues, neighborhoods that are extremely different in attributes and walkability were paired based only on the prevalence of venues. For example, using Richmond, VA as an example, the outer neighborhood mostly comprising industrial stops and high ways was clustered with an area known most for its hotel due to the prevalence of gas stations. Adding the population, income, and rent information gives KMeans more important features to work on.

Various values of ‘k’ [8, 12, 15], will be trialed to produce the best result.

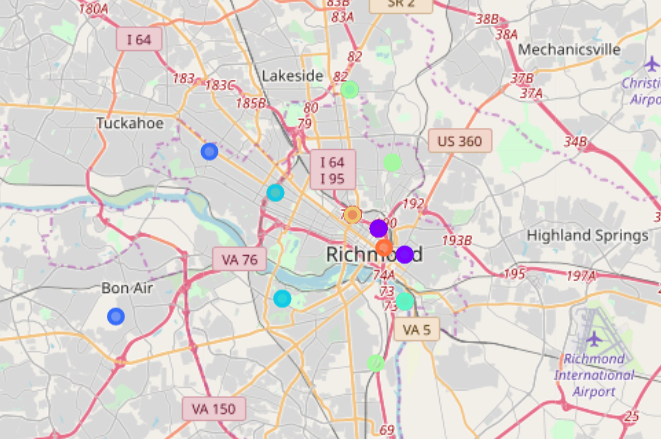
A value of 12 for K, produces fairly well developed clusters, depending on economic factors and then common venues. However, there are a few clusters with very few neighborhoods included that could easily be grouped together. For example, the 10th cluster in city 2 has only one neighborhood.

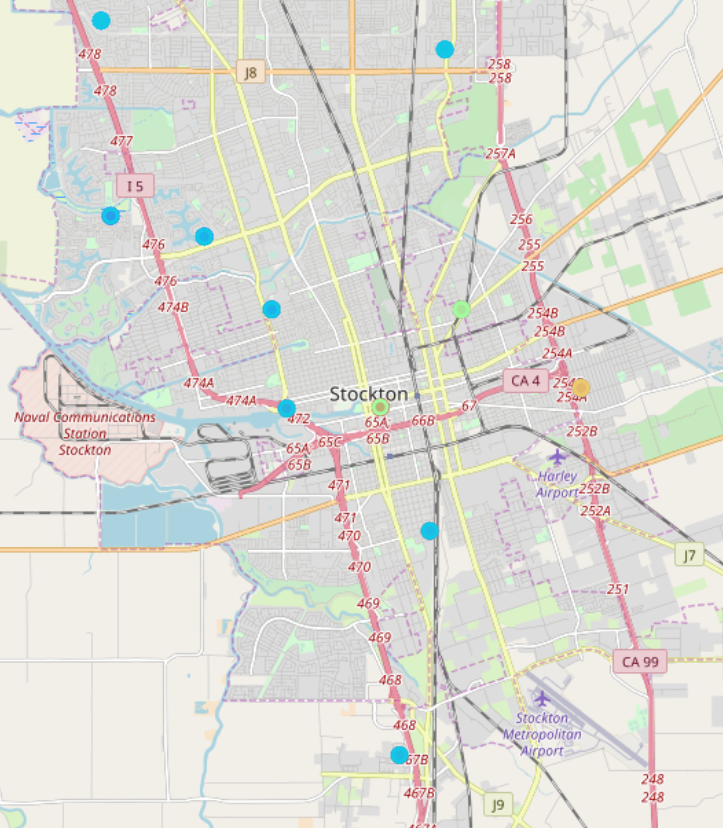


To achieve this, a value of 15 for k was processed, to observe whether some of the bigger clusters would split, and then any such small clusters as the example above would be absorbed. In fact, more 1 neighborhood clusters appeared. So a value of 8 for k was implemented.

All one neighborhood clusters we’re eliminated, and the clusters are more similarly shaped than in previous trials. Again, heavy dependence on the economic and population factors, as expected.

Clusters Mapped (k=8)



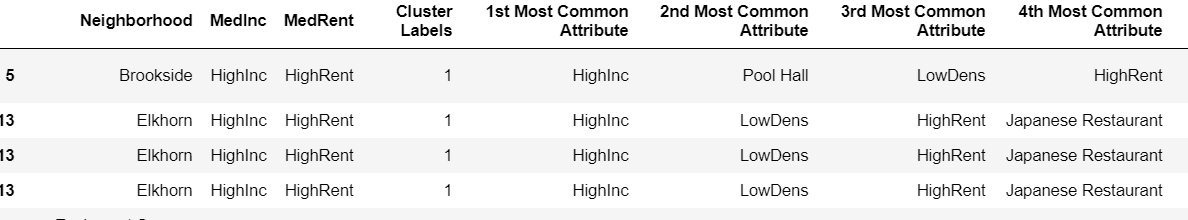


**Conclusion**

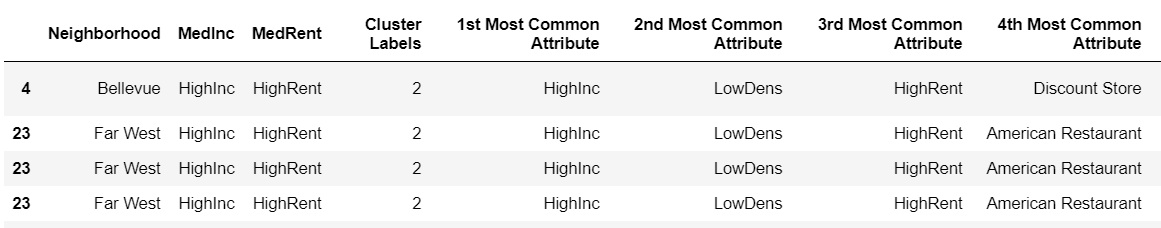
The value of 8 for k creates the most appropriate clusters for the two cities in this test case, Richmond, VA and Stockton, CA.

Cluster 1 in city 2 compared to cluster 2 in city 1 both show neighborhoods with High Median Incomes, Low Population Density and High Median Rents. Knowledge of the first city affirms that that City 1 Cluster 2 represents affluent suburbs, desirable for families and those seeking to invest in owning their own home.

City 2 Cluster 1



City 1 Cluster 2



Likewise, Cluster 5 in City 1 and Cluster 2 in City 2 share many attributes pointing to middle income, residential areas. Both clusters have Medium Income, Medium Rent and Medium Population Density, with restaurants and services catering to residents (banks and gyms).

**Future Directions**

It may be of interest to scrape and cluster with the added feature of Housing Prices to give more information to the users who will be looking to purchase real estate. The feature was not included in this application so far because many neighborhoods on City-Data.com do not have explicit information. In such a case, perhaps data could be drawn from government sites from public tax information.

The ideal end state of the above processes would be a user interface that allows users to input their two cities and select what kind of attributes they value most. For example, proximity to restaurants, or housing within a specific price band? As it stands, a user of the above processes is required to have a FourSquare Developer account as well as manually changing the cities targeted.