**Predicting the Ideal Neighborhood for Relocation**

**Gennaro Maida**

**September 3rd, 2019**[**Full Jupyter Notebook**](https://github.com/gtm1235/IBM_DS_Capstone/blob/master/neighborhood_relocation_recommendation.ipynb)

**1. Introduction**

**1.1 Background**:

Often it is difficult to choose neighborhoods when moving to a new city. Given certain criteria, can more informed decisions be made? The idea arose as my fiancée and I discussed where we would like to move in order to accelerate her medical career and my tech career. We hoped to find a lively neighborhood that fostered growth in both fields, as well as, be suitable to raise children. We found it is a daunting task.

**1.2 Problem:**

Most couples choosing to move have a list of cities they are considering and are stuck at that stage. These couples may have an idea what type of cities they prefer, or several cities they have enjoyed while visiting. It would be useful to provide customers with a select group of neighborhoods that offer the same atmosphere as the ideal locations. Additionally, customers may have specific demographic preferences in these neighborhoods such as cost of living, average age, or % of professionals within the community.

**1.3 Interest:**

I propose to save time by guiding customers in the search for a new home. Ideally, travel costs are reduced by offering a more focused search. I believe much of the stress in relocating can be reduced by ensuring a comfortable, familiar location as a destination.

**2. Data Collection and Acquisition:**

**2.1 Cities to Be Analyzed:**

1) We chose five cities as potential destinations:

* Dallas, TX
* Austin, TX
* Chicago, IL
* Jacksonville, FL
* San Diego, CA

2) We chose four our favorite locations around the US based on venues and culture:

* West Palm Beach, FL
* La Jolla, CA
* Portland, ME
* Downtown Chicago, IL

**2.2 Sources of Data:**

In order to solve the problem, we must acquire a variety of data sets:

1) Neighborhood Lists

A. Wikipedia.com

i. [List of communities and neighborhoods of San Diego](https://en.wikipedia.org/wiki/List_of_communities_and_neighborhoods_of_San_Diego)

ii. [List of Austin Neighborhoods](https://en.wikipedia.org/wiki/List_of_Austin_neighborhoods)

iii. [List of neighborhoods in Dallas](https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Dallas)

iv. [List of neighborhoods in Chicago](https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Chicago)

B. Conservopedia.com

i.  [List of Neighborhoods of Jacksonville, Florida](https://www.conservapedia.com/Neighborhoods_of_Jacksonville,_Florida)

2) Latitude, Longitude and zip code geolocation data:

This data will be obtained from Google’s geolocation API using the requests module.

3) Location Venue Data:

[Foursquare.com](https://developer.foursquare.com) API to explore local location venue data.

4) Zip Code Demographics:

[ZipWho.com](http://www.zipwho.com). A website that offers a variety of data such as median income, housing costs, cost of living index, housing, education, and more.

5) Additional Data:

Additional zip code data including average commute time and location-based industry income:

1. [Commute CSV Link](https://drive.google.com/file/d/1FCEbir7x__bhfdrufJ63xgCdPCnaZv6z/view?usp=sharing)
2. [Industry Salary CSV Link](https://drive.google.com/file/d/1-4HlSaGHyX07qRfo9sdw5AM1YaFb0Vmk/view?usp=sharing)

**2.3 Python Data Acquisition Tools:**

A) pd.read\_html -- table extraction tool for html-based data

B) BeautifulSoup4 -- a Python html parsing tool

C) Requests-- http request module

D) Numpy -- matrix manipulation tool

E) Pandas – Python Based Data Analytics Tool

**2.4 Data Cleaning:**

I acquired a list of 512 neighborhoods, of which, many were in the same zip code. I was able to reduce my data set to a more reasonable size of 152 neighborhoods by combining all relevant neighborhoods into a single zip code. I was also able to remove any neighborhoods for which no zip code was found through the Google API.

While using the Google API, several longitude and latitude values were erroneous as they appeared in different states; these were removed after noting them on observation. The API also returned more then one longitude and latitude value for multiple locations. I chose to append only the first returned to the data frame.

Table : Example Neighborhood Data Frame after cleaning – index 2-4 will be represented by zip code 32210

A screenshot of a cell phone

Description automatically generated

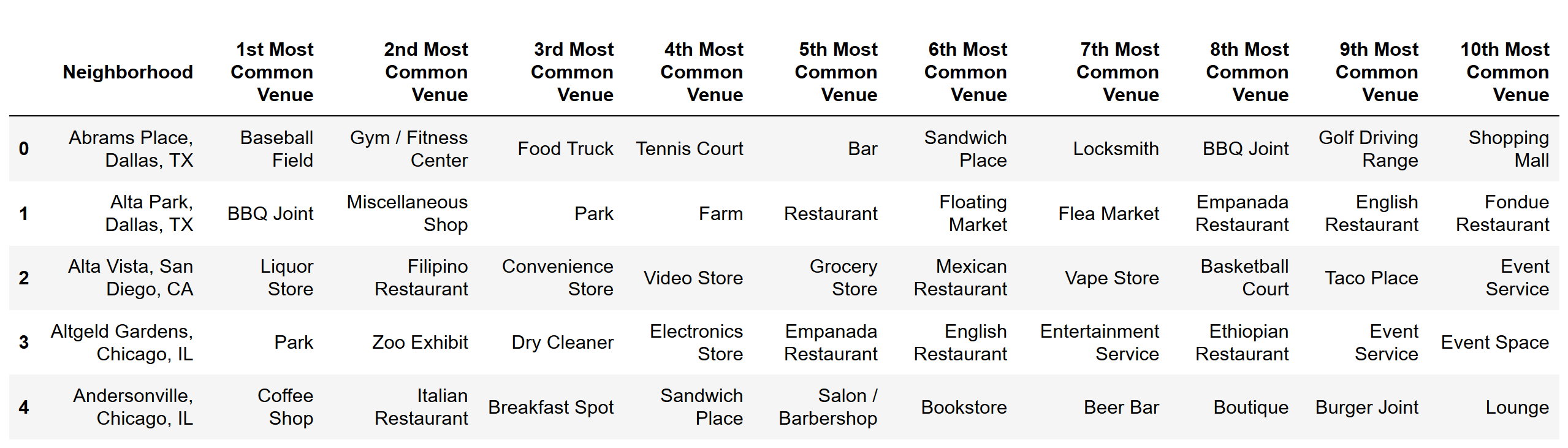
After acquiring the Demographic data which had 34 different categories, I eliminated the rank of each absolute value, for example Median Income and Median Income Rank. The absolute value was more useful then the relative value due to the small subset of data used. I eliminated any neighborhood for which there was no demographic data found. I retained all the remaining demographic attributed for potential use while sorting and filtering data in the final stage.

**2.5 Foursquare API:**

The Foursquare API allows one to fine tune the explore request while selecting venues. First, I varied the radius of the search depending on the location or city searched. For example, Jacksonville has very large neighborhoods on the order of miles. I chose to increase the search radius to 7000M. Chicago on the other hand has many tightly packed neighborhoods so the value was 500M for a narrower search. This allows one to keep venues specific to a neighborhood.

I limited the search to a maximum of 120 venues. I felt that this would return the most appropriate values for a neighborhood without saturating the data frame.

Table : Example of Top Ten Venues Data Frame After Sorting Venue Data



3. Exploratory Data Analysis:

**3.1 One-Hot Transform of Foursquare Data:**

In order to differentiate neighborhoods based on type and amount of venues present, I chose k-means clustering as an unsupervised machine learning tool. In order to prepare the data for the k-means cluster, I applied one-hot data transformation to the pandas data frame returned by Foursquare. Specifically, I grouped all one hot venues by neighborhood and calculated the mean. This showed not only the venue present but the weight or prevalence of that venue within a neighborhood. In the table below (which shows only 14 of the 422 potential venues) one can see that Andersonville has the most venues in the selection.

Table : Mean One-Hot Transformed Foursquare Data

A screenshot of a cell phone

Description automatically generated

**3.2 Ideal Cities One Hot Transform:**

In order to mimic my ideal city averaged across the four chosen cites, I first applied One Hot Transform again for each of the four cities favorite cities. I then took the mean across the columns to arrive at the “ideal-average” city:

Table : Ideal City Average One-Hot

A screenshot of a cell phone

Description automatically generated

There were only 92 venues present versus 412 in the larger Dataset. In order to match them I filled in zeros for the appropriate empty columns. The venues were weighted more heavily because only four cities were taken into account while calculating the mean; however, this helps to ensure a favoritism towards our ideal venues.

**3.3 K-Means Clustering Analysis:**

I employed the scikit-learn module to import the k-means cluster algorithm into my Jupyter notebook. K-means uses a “distance” based algorithm. A centroid or several centroids, depending on the number of clusters, are initiated randomly or more deterministically. Then they are slowly moved by gradient descent to minimize the distance of the input values from the various centroids. The function allows one to choose how the centroids are initially placed, how many iterations are performed for cluster calculations, how many clusters are calculated and averaged for best outcome, and the k-means initialization algorithm. I used these variations to best pick my overall matches.

I first ran the cluster one time on all the neighborhoods to get a feel for spread of the neighborhoods (examples in Folium Mapping). Then I ran 108 iterations of the k-means algorithm and varied each of the hyperparameters. I predicted the cluster for my ideal city and created a list of all cities in this cluster. After all iterations, I counted how many times each city appeared and retained the ones that appeared over 85% of the time over all iterations.

Table : k-means iterative process – **For Loop written Explicitly to show all layers of iteration**

A screenshot of a social media post

Description automatically generated

**3.4 Folium Mapping:**

I chose Folium maps due to the ease with which I can create gorgeous maps. I used Folium to show the neighborhoods and various clusters among them across all cities. In the analysis you will see the maps of matching clusters. Below is an example of the initial cluster of all neighborhoods.

A picture containing text, map

Description automatically generated A picture containing text, map

Description automatically generated

*Table 7: Dallas Neighborhood Clusters Table 8: Chicago Neighborhood Clusters*

**3.5 Pandas Analysis of Demographics:**

Using the appended data from the ZipWho website, I manipulated and refined the data using Pandas. I chose to include cities with a median income greater than $45,000. I further refined the data by choosing locations with a cost of living index less then 400 to remove cities that were outside my budgetary constraints. I kept the manipulations to a minimum as there are many choices based on housing, age, family size, job type etc... that are personal to the selections and final choices. I performed several manipulations to show the simplicity with which choices, filtering, and sorting can be made.

4. Analysis of Results:

**4.1 K-means Analysis:**

I chose novel methods for the k-means analysis. First by iterating the k-means algorithm through various centroid initializations, the clusters varied in size and distribution. This held true for the number of iterations performed on the data as well. I realize there were chances of both underfitting and over fitting, so I ran 108 iterations over which I varied the following hyperparameters:

1. max\_iter: 50, 100, 300
2. kclusters: 5, 10, 20 – Number of clusters
3. init: Random, k-means++ -- K-means centroid initialization
4. random-state: None, 10 – Random, deterministic.
5. n\_init: 5, 10, 20 – Number of times the k-means algorithm will be run with different centroid seeds

For further description see [scikit-learn k-means documentation](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html)

As stated in the previous section, I tallied the number of times each city appeared within the ideal cluster at each iteration resulting in the following plot:

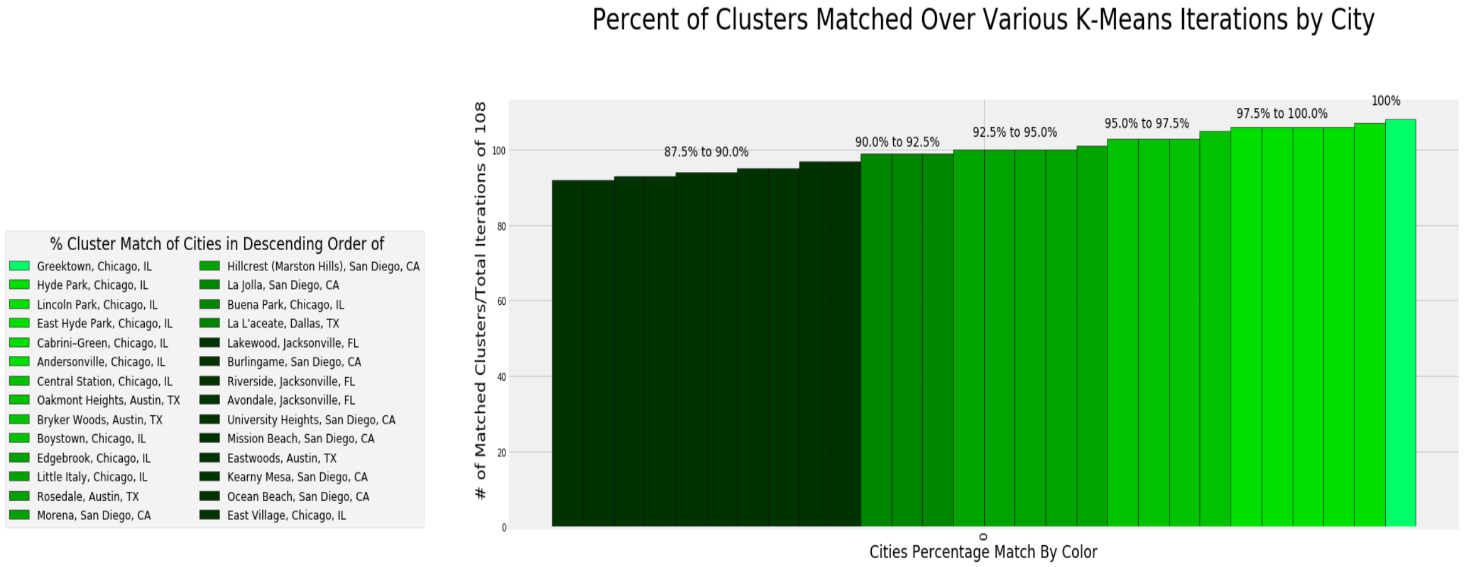


Table 9-1: Cities Most Represented During Iterative K-means Cluster Analyses

A screenshot of a cell phone

Description automatically generated

Table 9-2: Top 14 Cities Count

As the data shows only one neighborhood/zip code appeared in all 108 iterations, 28 cities appeared greater than 85% of the time. I feel comfortable that these are strong candidates in regard to venue breakdown within the location. However, I felt further verification was necessary.

**4.3 MSE Analysis:**

Because k-means clustering is based on distance from a centroid within each cluster, I felt that Mean Squared Error is an appropriate evaluation to measure distance error. The distance terror between ideal venue one-hot data and each neighborhood one hot data was calculated. I calculated the mean squared error for all 158 cities, the average mean squared error, standard deviation, and the number of standard deviations form the mean. Table one shows the distribution of MSE’s present in the data:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

*Table 10-1: Distribution MSE Plot with Outliers Table 10-2: Distribution MSE Plot without Outliers:*

I chose to use the data, after removing the outliers. As you can see there is a peak in distribution near 0.0001 with a skew to the right. There is a wide variation in the magnitude of the error above the mean. I analyzed the MSE and Standard Deviation Distances from the mean for each of the final cities (see below – chosen after pandas manipulation) and created data frame Table 11.

As can be seen in the table below. All the ideal neighborhoods have a value of standard deviation below the mean MSE. These vary from ~.2 to 1 standard deviation below the mean. I feel comfortable with both the values of the MSE (Average = .0000903) and an average of -.78 standard deviations below the mean. These show all predicted venues are most closely related to our ideal city choice. I also for fun included a column for:

A screenshot of a cell phone

Description automatically generated

Table 11: MSE Data for the Ideal Cities

**4.5 Demographic Analysis:**

I kept this step short as there will be variation based on the preferences of the customer. For the sake of this analysis, I first chose cities with a minimum Median Income of $40,000 and a maximum Cost of Living Index of 400. The many column choices to filter and sort include:

'Neighborhood', 'Latitude', 'Longitude', 'Zip Code', 'Median Income', 'Cost Of Living Index', 'Median Mortgage To Income Ratio', 'Owner Occupied Homes Percent', 'Median Rooms In Home', 'College Degree Percent', 'Professional Percent', 'Population', 'Average Household Size', ' Median Age', 'Male To Female Ratio', 'Married Percent', 'Divorced Percent', 'White Percent', 'Black Percent', 'Asian Percent', 'Hispanic Ethnicity Percent'

For the sake of brevity, I also refrained from adding commute times by zip code and compensation by profession by zip code. The data is linked in the above sections for further analysis if the reader wishes. This analysis resulted in a final data frame including 14 ideal cities:

A close up of a logo

Description automatically generated

*Table 12: Final Pandas Data frame post Pandas manipulation.*

**4.5 Visual Analysis with Folium:**

I plotted the resulting neighborhoods using folium to see their locations. The results did not include any neighborhoods in Dallas or Jacksonville. These were eliminated during the pandas manipulation due to the constraints on median income and cost of living.

A picture containing text, map

Description automatically generated A picture containing text, map

Description automatically generated

*Table 13.1: Austin Ideal Neighborhoods Table 13.2: San Diego Ideal Neighborhoods:*

*Tables Continued Below:*

*A close up of a map

Description automatically generated*

*Table 13.3: Chicago Ideal Neighborhoods*

These maps allow for a visual representation of the cities location which is important in decision making. Personally, I am attracted to the water, as well as, downtown areas which is reflected in the analysis visually.

**4.6: Ideal City Venue Analysis for Comparison:**

Lastly, A list of top ten venues present at the ideal locations was built to compare the results. I intend to compare this with table 12 to see the correlations in neighborhood venue types.

A screenshot of a cell phone

Description automatically generated

*Table 14: Top Ten Venues in Favorite Cities:*

Additionally, I created a data frame for the top venues from the ideal cities mean value one hot df to show the most common venues averaged over four cities:

A screenshot of a cell phone

Description automatically generated

*Table 15: Top Ten Venues in Favorite Cities Average:*

5 Conclusion:

I feel that this analysis was able to produce a list of cities with many characteristics that I was searching for. My ideal cities contained many venues that I like such as a variety of restaurants and cultural venues. There were also a variety of outdoor and\or athletic venues. Additionally, on visual inspection of the locations, they were indeed located by the water and/or the city center. Of course, many of our ideal city preferences were downtown or waterfront locations. These results seem to be consistent with the expected type of results.

Further pandas analysis allowed me to refine the search by using specifics that allowed me to choose neighborhoods where connections can be made, and businesses can be built. I wanted to find exciting venues with outdoor activities, combined with a professional yet family-oriented neighborhood. I feel that the final results reflect these qualities.

Sadly, I would have hoped to find neighborhoods with a more attractive cost of living index. There are also discrepancies due to the demographics. For example, the Median income is based on all ages, not just those of working age, and was significantly reduced. Certain neighborhoods were eliminated due to low median income; however, actual numbers appear to be significantly higher.

There is much room for improvement, and I have many ideas for the future:

6. Future Potential:

There are many ways to improve or build upon the original idea. I would like to try different clustering algorithms such as Gaussian Mixture to help clear up some of the cities on the borders between clusters.

I would like to run a deeper iterative cluster as well. I would like to find what the maximum number of chosen ideal locations would be before there are too many venues and the ideal location becomes less specific.

Of course, I would also like to further analyze and manipulate the final clustered city using available demographics and neighborhood data. Given time, I would also like to locate more reliable demographics data sets to improve the final sorting and filtering.

The options seem unlimited. Now that the basic work is done it seems there are many ways to improve and hone my Relocation Recommendation engine!