**Selecting Neighbourhoods for Vacation based on Individual Preferences**

Anna Keshabyan

March 30, 2019

**Introduction**

**Background**

Planning vacations can be tricky. It's not a secret that different people look for different things when it comes to vacations. Culture, entertainment, food, sports, and natural sights can all be important for tourists. The uprising of accommodation sharing services like Airbnb and growing affordability of travel have led to increasing number of people planning their own trips without any help from tour agencies and travel operators. One of the most important parts of planning a self-guided vacation is selecting the place where one is about to stay. Choosing the neighborhood that suits one’s interests and travel needs is key to having a successful vacation. However, with different people having different preferences, it can be a daunting task.

**Problem**

When choosing a neighborhood for a vacation stay tourists face a number of problems. Not only do they have to research all the neighborhoods in the area, they also need to know the ins and outs of local transit, what are the main attractions in the area, whether there are any good restaurants and grocery stores next to their accommodations and what kinds of entertainment they can expect to find nearby. Doing all this research can take a lot of time and energy. Therefore, this project aims to build a base of a system that recommends the best neighborhood to stay at during a vacation based on the person’s preferences for what a good neighborhood should have.

**Interest**

People planning their vacations without external help will benefit from this project as it will save them hours of research of residential areas in their selected vacation spot. It could also help anyone who likes to explore new cities or even new parts of their own city, as the algorithm will suggest the places a person might want to check out based on the places that they said they liked or disliked.

**Project Scope**

For the purposes of the assignment I decided to use three cities in California, namely San Francisco, Los Angeles, and San Diego. Any other city or group of cities could be selected, I just chose these three for simplicity purposes. The base scenario for the project involves a person living in one of these cities, who is familiar with the neighborhoods in their hometown and has formed some preferences and who wants to travel to an unfamiliar city and wonders what part of it they should stay at. The goal of the project is to suggest neighborhoods to such person based on persons preference and neighborhoods’ similarity to one another in term of available venues.

**Data Acquisition and Cleaning**

**Data Sources**

The first step in this project was to find a list of neighborhoods in each of the selected cities. I used Google search to come across Wikipedia pages listing San Francisco, Los Angeles, and San Diego neighborhoods.

The three respective pages were:

1. San Francisco: <https://en.wikipedia.org/wiki/List_of_neighborhoods_in_San_Francisco>
2. Los Angeles: <https://en.wikipedia.org/wiki/List_of_districts_and_neighborhoods_of_Los_Angeles>
3. San Diego: <https://en.wikipedia.org/wiki/List_of_communities_and_neighborhoods_of_San_Diego>

**Web Scraping and Geocoding**

Following the first step, I used BeautifulSoup and Pandas libraries to scrape the web page and create the data frames with the lists of neighborhoods for each respective city. Geocoder tools were used to find the unique coordinates (latitude and longitude) for each individual neighborhood. There were several neighborhoods that the Geocoder could not locate coordinates for and in order not to complicate the task, I excluded those neighborhoods from my sample. The final sample included a total of 310 neighbourhoods: 124 in Los Angeles, 105 in San Diego, and 81 in San Francisco.

**Data Acquisition**

After creating an initial list of neighborhoods, I used FourSquare API to get the data on the type of venues available at each of these neighborhoods. Initially, I included all kinds of venues in my analysis for it to be as complete as possible. After this step, I had a dataframe containing 5,738 unique venues that belonged to 391 unique categories.

**Data Cleaning**

In the next step, I cleaned up the data. I first grouped the venue information by neighborhood and obtained a dataframe outlining what categories of venues are prevalent in each of the neighborhoods. There was a total of 310 neighborhoods in 3 cities with the venues in 391 categories.

**Feature Selection**

The problem with having so many venue categories in the data set is that some rare categories can severely skew the results of models like K-Means clustering by assigning the mean points closer towards outliers than they should be. For this reason, I decided to cut the number of categories I had. Through the exploration of the data, I determined that most of the venue categories could be grouped into the following categories:

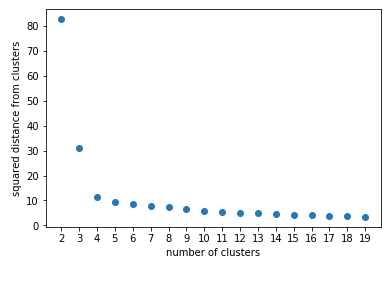
* Accommodation (hotels, motels, etc.)
* Food (restaurants, cafes, bars, etc.
* Cultural sights (museums, galleries, performance venues)
* Natural sights (parks, beaches)
* Entertainment (clubs, lounges, karaoke bars)
* Transit (bus stops, railway stations, airports)

Hence, I combined most columns into one of the above categories. I used the sum of columns with relevant values as aggregate measure for each category. For example, the values for venue categories like “Vietnamese Restaurant”, “Bakery”, “Coffee Shop” would be combined into Food variable, while the values for “Beach”, “Lake”, “Trail” categories would all contribute to the value of the Nature column. As such, at the end of this exercise, I had a dataframe with 310 neighbourhoods and 6 venue categories. For further exploration purposes, I also constructed a table displaying 10 most popular venues in each of the neighbourhoods.

**Data Analysis and Results**

**K-Means Clustering**

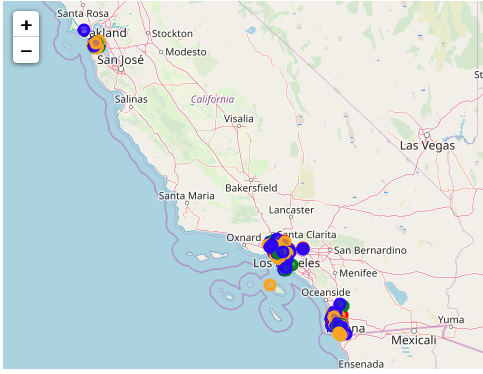
First, I decided to conduct K-Means clustering in order to explore the groups of neighbourhoods that are similar to each other. In order to determine how many clusters I need to use for my algorithm, I first constructed a scree plot to visualize the decrease in error each additional cluster provided (see below).



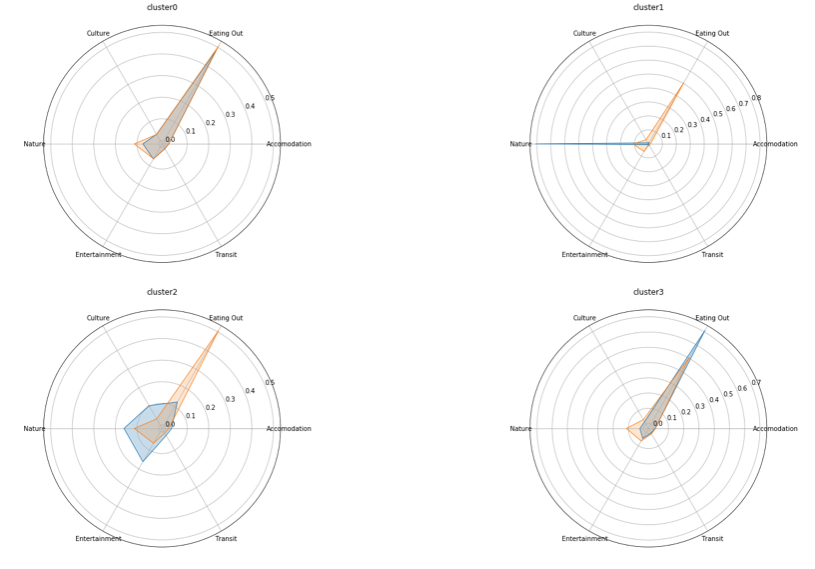
The scree plot demonstrates that the error decrease levels off significantly after the number of clusters exceeds 4. Thus, I used 4 clusters to train my model. After training the model, I obtained the cluster labels for each of the neighborhoods. As demonstrated in the table below, Cluster 3 was the most numerous cluster with 139 neighbourhoods, while Cluster 1 was the smallest with only 21 neighbourhoods.

|  |  |  |
| --- | --- | --- |
| **Cluster** | **# of Neighborhoods** | **% of Sample** |
| 0 | 112 | 36.1% |
| 1 | 21 | 6.8% |
| 2 | 38 | 12.3% |
| 3 | 139 | 44.8% |

My next step was creating an interactive map with each cluster specified by a separate color



I then moved to exploring each cluster in detail by viewing the most popular venues in each of them. The most popular venues in Cluster 0 were predominantly food venues like coffee shops, bars, and restaurants. In Cluster 1, most popular venues were all nature-related: it had lots of parks, trails, lookouts, and preserves. Cluster 2 neighbourhoods comprised a healthy mix of cultural sights and some nature spots. Finally, Cluster 3 had a lot of food venues as well as entertainment and cultural scene. In order to visualize the differences among clusters on these different categories, I went a step further and created radar plots using Seaborn library.



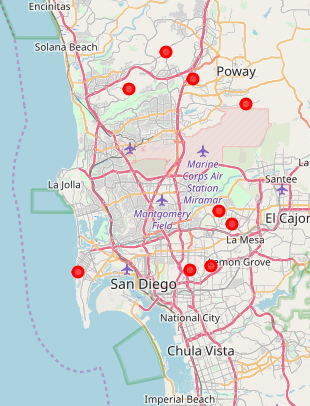
These radar plots supplement and confirm the results of my exploration. These radar plots are designed in such a way that the orange shape represents the sample average while the blue shape represents the average of a specific cluster thereby showing how this particular cluster is different from the overall population. Similar to previously discussed analysis, they show that clusters 0 and 3 are indeed pretty similar to each other, with cluster 3 having slightly less nature and slightly more food venues than cluster 0. Cluster 1 is confirmed to have lots of nature sights, while cluster 2 neighbourhoods look like they have busier cultural and entertainment scene with some nature mix-in.

**Discussion**

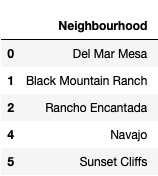
**Potential Applications**

After obtaining all the information and exploring the clusters of California neighbourhoods, I looked into the potential applications of this model. As this model is designed to assist with selecting accommodation for a short-term stay, one of the potential applications of this clustering involves identifying neighborhoods in a new city which are similar to the neighbourhoods one is familiar with.

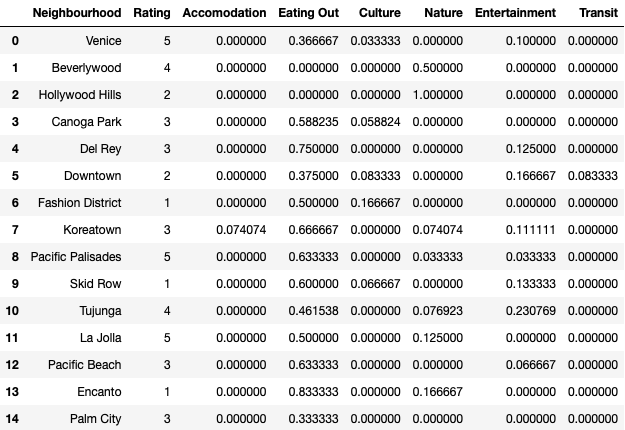
For my first use case, I used a relatively simple example of a person travelling to San Diego who wants to find the neighborhoods which are similar to her home neighborhood of Hollywood Hills in Los Angeles. I first found which cluster Hollywood Hills belonged to (Cluster 1) and then used this information to locate the neighborhoods in San Diego which belonged to the same cluster. This search returned 9 neighbourhoods in San Diego located in the suburbs close to parks, mountains, and beaches.

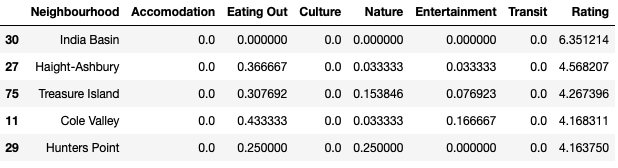


My next step was to identify the neighbourhoods in San Diego that would be exactly like Hollywood Hills. In order to do so, I used the distances between a vector with the features of Hollywood Hills neighbourhood and matrix containing cluster 1 neighbourhoods in San Diego. The minimum distance was 0 for 5 neighbourhoods in San Diego which means that these neighbourhoods were exactly like Hollywood Hills in terms of different categories of venues used in this analysis.



In my second use case, I decided to explore a slightly more complicated scenario in which a person who live in Los Angeles and had previously travelled to San Diego and ranked neighbourhoods in these two cities, wanted to find the best neighbourhoods to stay in San Francisco based on the previous ratings. I came up with those ratings arbitrary simply for the demonstration purposes. As I only used one person’s ratings for this scenario, I trained multiple linear regression on the existing ratings and used it to come up with ratings for neighbourhoods in San Francisco. Below you can see the input table with ranked neighbourhoods and their characteristics, and the output table with top 5 recommended neighbourhoods in San Francisco.





The scenarios discussed above are definitely not the only applications of the clustering model and recommender tools. With more people ranking different neighbourhoods it is possible to develop a more robust and accurate model that would allow to make better recommendations and suggest better neighborhoods for people looking for the short-term accommodations.

**Conclusion**

To summarize, the purpose of this assignment was to develop a tool that would allow tourists select neighborhoods at their vacation destination that best suit their preferences and resemble their favourite neighborhoods at home. I used both outside source like Wikipedia and FourSquare API to come up with neighborhoods, their coordinates, and types of venues available. Furthermore, I ran k-means clustering to identify unique categories of neighborhoods along the coast of California. I then used the outcomes of this model to demonstrate how a recommender system can be set up.