Deciding Where to Move Using Certificate Skills

Braden Smolko

9/9/20

Introduction

The stakeholders in this case are me and my girlfriend. We have both recently graduated from college and have been debating where to move. We want to live somewhere near family, but also somewhere that is like our current situation. We live in Downtown Frederick now and we really enjoy it. We must move for her graduate program soon though and we want to make the best decision that we can.

I thought that it would be interesting to apply the data science skills that I have picked up from the Coursera courses and the Foursquare location data to make a more informed decision. I talked with my girlfriend and she helped me create a list of possible places to live. I performed a cluster analysis on these cities to see which of them were like Downtown Frederick. With this knowledge, we can make a more informed decision that will lead to greater satisfaction with our move. If there are others in a similar situation, this method may help you find familiar living spaces.

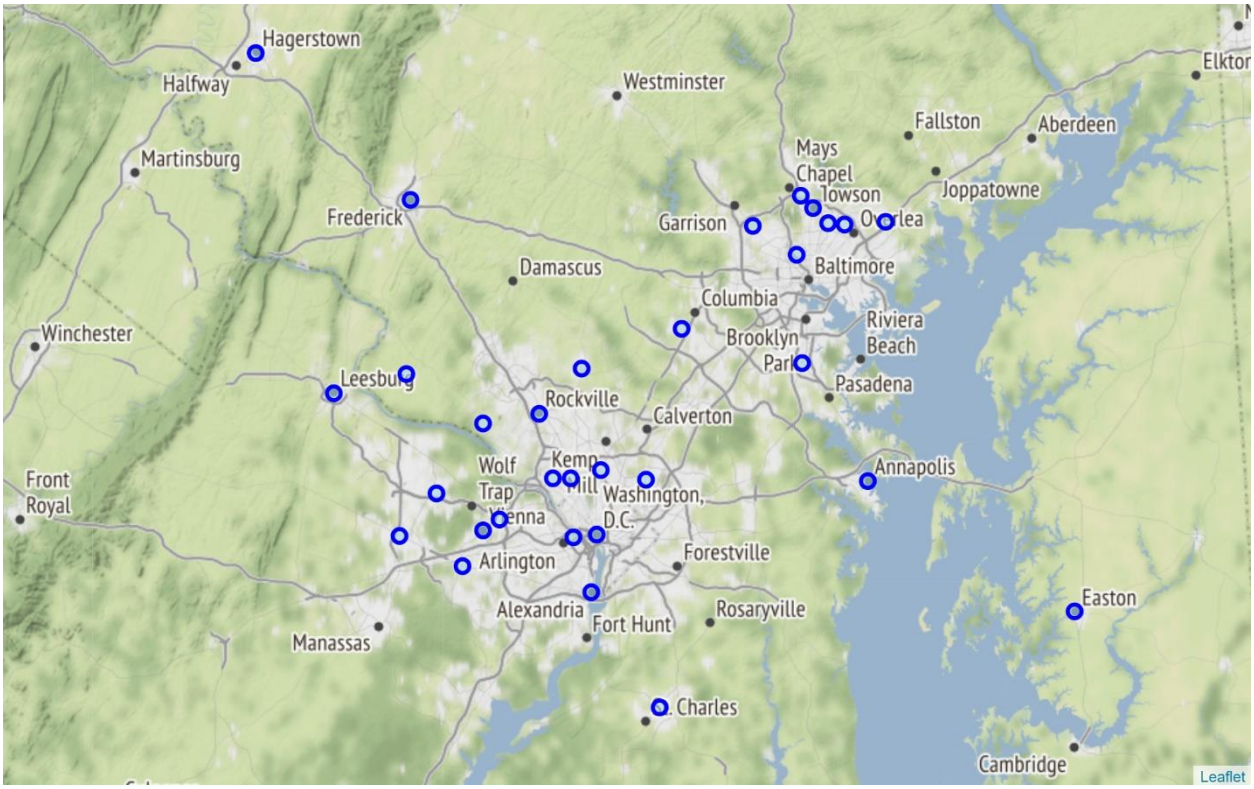
Data

This project requires that we use the Foursquare location data, which is perfect for the problem at hand. The Foursquare explore endpoint is of interest for this project. The explore endpoint allows for the user to specify where to search (latitude and longitude), how far around the given point to search (radius), how many venues to retrieve (limit), and how to choose those top values (relevance, popularity, or distance). This call to the API will return the venues and information that meet all the parameters. The information will include the location, category, and tips, among other things related to the venue. Category is the main feature that we are interested in.

I made a search for each city on my list to obtain the top 100 most popular venues within a 5-kilometer radius of the center. I then found the proportion of the categories for each city and used that as the feature set for the clustering algorithm. I used K-means clustering to determine similar groups of cities based on their venues and then examined the groups to see which cluster contained Frederick. Exploring the cluster gave me more information about what Frederick's venues look like and what similar cities tended to look like as well.

Methodology

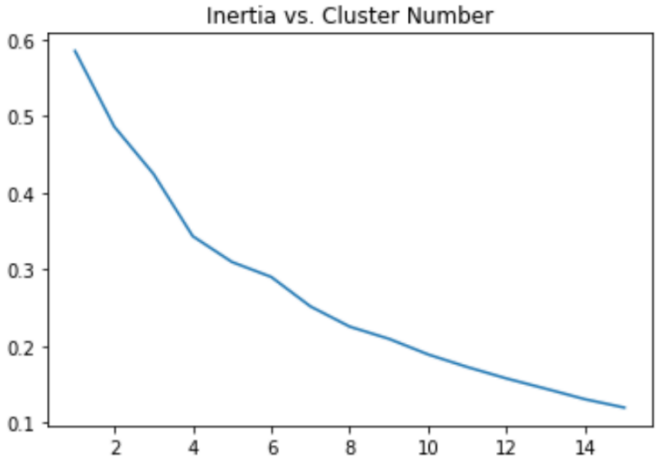
To start this problem, I needed a list of possible locations to move. To get this list, I asked my girlfriend what areas she would consider living. I also added other towns from the Baltimore-Washington Metropolitan Area to the list that were similar to Frederick in size. I ended up with a list of 31 possible cities to use in my analysis. In order to use these cities accurately with the Foursquare API, I used the Geopy library to geocode them. This means that I obtained latitude and longitude coordinates for each city. Geocoding the cities also allowed me to place them on a map (Map 1).

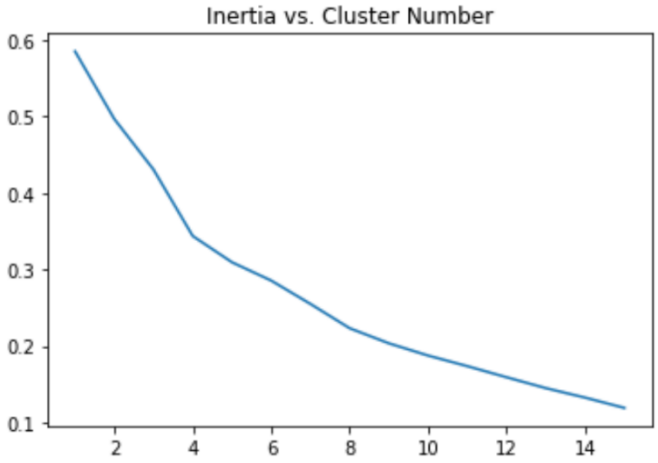


Map 1. Map of possible cities for our move.

Using the coordinates of the cities, I made calls to the Foursquare Places API. These calls returned the top 100 venues within a 5-kilometer radius of the coordinates for each city. From these returned venues I kept the categories and placed them in a DataFrame object with their corresponding city. I could then one-hot encode the categories and group by the city. This gave counts for venue categories in each city (i.e. 0 airports, 0 American restaurants, 3 coffee shops, etc.) I then turned the counts into percentages by dividing the number in that category by the number of venues returned for that city. For example, Annapolis had 0.07 in the "American Restaurant" category which can be interpreted as "7% of the venues returned for Annapolis are categorized as American Restaurants." Using the percentages instead of the counts helps to standardize the values for locations that might not have as many venues in the immediate area.

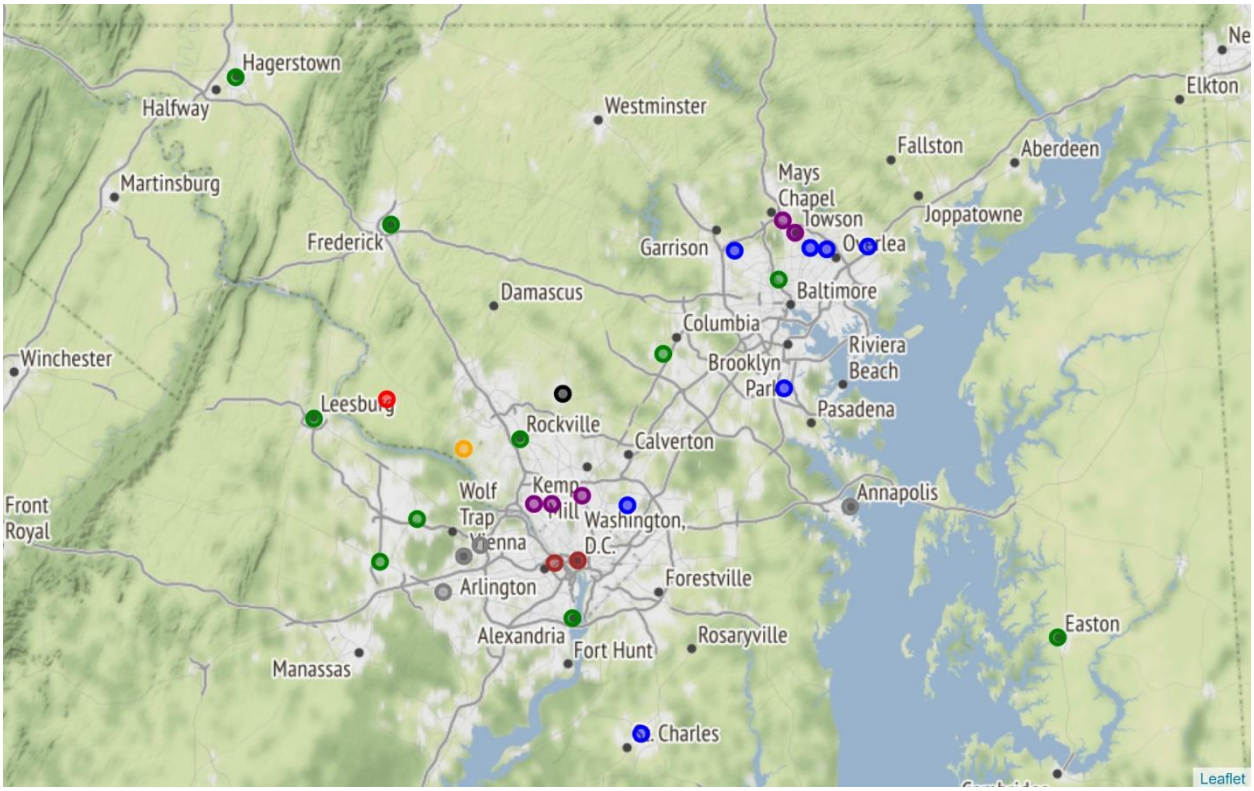
Once I had the category scores for each city, I could move onto some analysis. The data was unlabeled and perfect for the K-means cluster analysis technique. When using K-means, you must specify the number of clusters to use. To determine this number, I ran the K-means algorithm with varying numbers of clusters and varying random states. From each of these trials I stored the “inertia” value which is a measure of the within cluster variation. The goal of the algorithm is to create the lowest within cluster variance such that points in the same cluster are similar and points in different clusters are dissimilar (without just making each data point its own cluster). I used the elbow technique for determining the number clusters to use. To do this, I plotted the inertia graphs for varying numbers of clusters and random state values. Two example graphs can be seen below (random state values 3 and 4):





I looked at these graphs to find that the "elbow," the bend in the line before the inertia begins exponentially decreasing, was occurring at around n = 8 clusters most of the time. This led me to choose 8 as the number of clusters for my final model. I also chose a random state of 4 (represented in the second graph) because it had the clearest "elbow."

With the cluster number and random state values determined I was able to make my final model. This model placed all the cities into clusters based on their venue similarity. The different clusters can be visualized on a map with different colors representing different clusters.



Map 2. A map of the cities after clustering.

The clusters were not contained to location, meaning that cities with similar venues can be found in different areas. This is good news for me and my girlfriend because we are looking for a city that feels like Frederick but is closer to her graduate program. I examined the top categories for each cluster to see the breakdowns of venues. I then also looked at the number of parks in each city that showed up in the Frederick cluster. Number of parks and proximity to my girlfriend's graduate program helped me decide which city from the Frederick cluster we should consider.

Results

The clustering algorithm placed Frederick with 9 other cities to form a cluster. The cities are shown below in Table 1.

	City	State	Latitude	Longitude	Cluster
0	Frederick	MD	39.414219	-77.410927	7
1	Hagerstown	MD	39.641922	-77.720264	7
2	Hampden	MD	39.330940	-76.634969	7
3	Rockville	MD	39.084005	-77.152757	7
4	Columbia Town Center	MD	39.214397	-76.864589	7
5	Easton	MD	38.774495	-76.076307	7
6	Reston	VA	38.958374	-77.357980	7
7	Alexandria	VA	38.805110	-77.047023	7
8	Leesburg	VA	39.115450	-77.564545	7
9	Chantilly	VA	38.894154	-77.431151	7

Table 1. All cities in the Frederick cluster.

My girlfriend and I like to go on hikes and walks outdoors, so I used the Foursquare API to get the number of parks and trails within a 5-kilometer radius of each of these cities. The calls to the API for this kind of search were limited to 50 results, which ended up posing a slight problem. Most of the cities had over 45 parks nearby with Rockville, Hampden, and Alexandria all having a max score of 50. I was unable to use this to determine a clear winner for the best city, but it did show me that most of the cities in question would have ample places to explore. With this new information in mind, I decided it best to just pick the city that was closest to her school and to family. This option was Hampden! It is not the closest of our original list, but it is the closest from the Frederick cluster and I think it is the best option from this analysis.

I also looked a little further into each of the clusters to see what kinds of venues each had to offer. To explore the categories in each cluster, I combined all venue categories again and grouped by cluster instead of city like the first time. I then added up all the category counts and

divided by the total number of venues in the cluster before multiplying it by 100. This gave me the percentage of each category in the whole cluster. Exploring the top five categories shows the Frederick cluster to be diverse. It seems that the cluster was not decidedly similar because of a prominent category, but instead by the fact that the cities have a diverse collection of venues. The top category in the Frederick cluster was Coffee Shop, but interestingly this top category was less than 5% of the total venues. The Frederick cluster had low percentages for any one category, but it had a variety in the categories. The Frederick cluster's top 5 categories were: Coffee Shop, American Restaurant, Grocery Store, Pizza Place, and Park.

Discussion

The K-means algorithm has a few drawbacks that should be mentioned here. One drawback is the fact that the algorithm's result is based on the initial locations of the starting centroids. The algorithm then converges to the best option that it finds for the clustering, but this "best" may be a local optimum of clusters and could be different if the algorithm were run again with different initial centroids. This is the reason that I specified and tested multiple random states. Specifying a random state helps make the algorithm reproducible for the future by creating the same randomized starting centroids every time. I did not know this at first, so I became quite confused when my code was producing different results every time that I ran it. I settled on the cluster number and random state based on the elbow method, but sometimes the graphs were not so clear. I think that using a more decisive method for selecting the number of clusters would be beneficial for future projects.

I also feel like the clustering results were influenced by the number of venues returned for each city. The cities with fewer venues had higher category percentages for those venues simply because there were not as many. I think that clustering the cities based on these

percentages may have biased the results slightly. The places with fewer returned venues likely became more distant from the other venues and ended up in their own clusters. A similar bias would also likely be introduced without standardizing the results to a percentage. This biased result does not discourage my decision though. One thing that I like about Frederick is the fact that there are so many small places around, so having other cities in the cluster with lots of options seems like a good thing anyway.

Conclusion

Using cluster analysis and location data, I have decided that Hampden is the best choice for us. Hampden has a wide variety of venues available in the area, which may not be the most important factor during COVID right now but having the options in the future would be great. Hampden also has lots of places to explore with a higher number of parks and trails in the immediate vicinity. This choice also comes from the knowledge that it is the best balance of distance to my girlfriend's graduate program and proximity to family. This project has been very exciting and very insightful.