Cluster-based tool for comparing neighborhoods in familiar and unfamiliar cities

Liam Spoletini

IBM Digital Certificate in Data Science

Introduction

Whatever the particulars of your situation, if you're planning on moving to a new city, it can be overwhelming trying to understand the reputations of that city's neighborhoods. You peruse lists, glance at reviews of local restaurants, and might even hop on google street view to take a virtual tour, but, understandably, you're still lost. The previous activity in this online module, where we were asked to cluster neighborhoods in Toronto into groups using information about local venues, gave me an idea on how to answer the following question: how can I compare neighborhoods in a new city to ones I have intimate knowledge of? The answer, of course, is to put neighborhoods from different cities in the same clustering algorithm. Hypothetically, if I wanted to move to a place that was similar to an area of my hometown (let's say the Belle Meade area in Nashville), I could cluster my hometown's neighborhoods with neighborhoods from the city I plan on moving to and see what neighborhoods from the new city are clustered along with Belle Meade. For my capstone project, I will be conducting this analysis using two cities I might move to (Washington, DC, and Seattle, WA) abd the city in which I currently reside (Nashville, TN). Before clustering them together, I will cluster the cities individually in order to explore the data further. The goal of this analysis is to link my understanding of the place I live to places I might live next year, but this analysis would be helpful for anyone who's trying to make a decision about where to move.

Data

The first step in this analysis will be to generate a list of neighborhoods for each city and corresponding GPS coordinates for each neighborhood. To do so, I will download lists of neighborhoods and their corresponding zip codes from these three links:

DC:

https://www.cccarto.com/dc/index.html

Seattle:

http://seattlearea.com/zip-codes/

Nashville:

https://www.nashvillesmls.com/nashville-area-zip-codes.php

There are websites with all zip codes in the country listed for different cities, but there was no information on the neighborhoods associated with the zip codes. I decided to use the above links because although they may be more difficult to download from, the neighborhood name is an important variable that can help potential movers do additional research following the clustering analysis. Once the neighborhood names and zip codes are in dataframes, I will use the pgeocode python library to generate latitude and longitude coordinates for each neighborhood, adding them to the dataframe. Foursquare data to generate nearby venues for each neighborhood to be used as features in a subsequent clustering analysis.

Methodology

Data cleaning and organization

Once I manually loaded the data from the links above into Excel and saved them as csv files, I noticed that some of the neighborhoods listed had the same zip code. I chose to combine these duplicate zipcode neighborhoods into one row, and I did so in a loop. Afterwards, I used the pgeocode package from python to extract latitude and longitude values for each zip code, storing each in the data frame along with the neighborhood(s) names and zip codes as columns. At the end of this process, I had a data frame for each city, with columns for the names of neighborhoods, the corresponding zip codes, and the latitude and longitude values of the zip code regions.

Visualization of neighborhoods in each city

The folium library was used to generate visualizations of the regions in each city. Shown below are the maps of Nashville, Seattle, and Washington, D.C., respectively

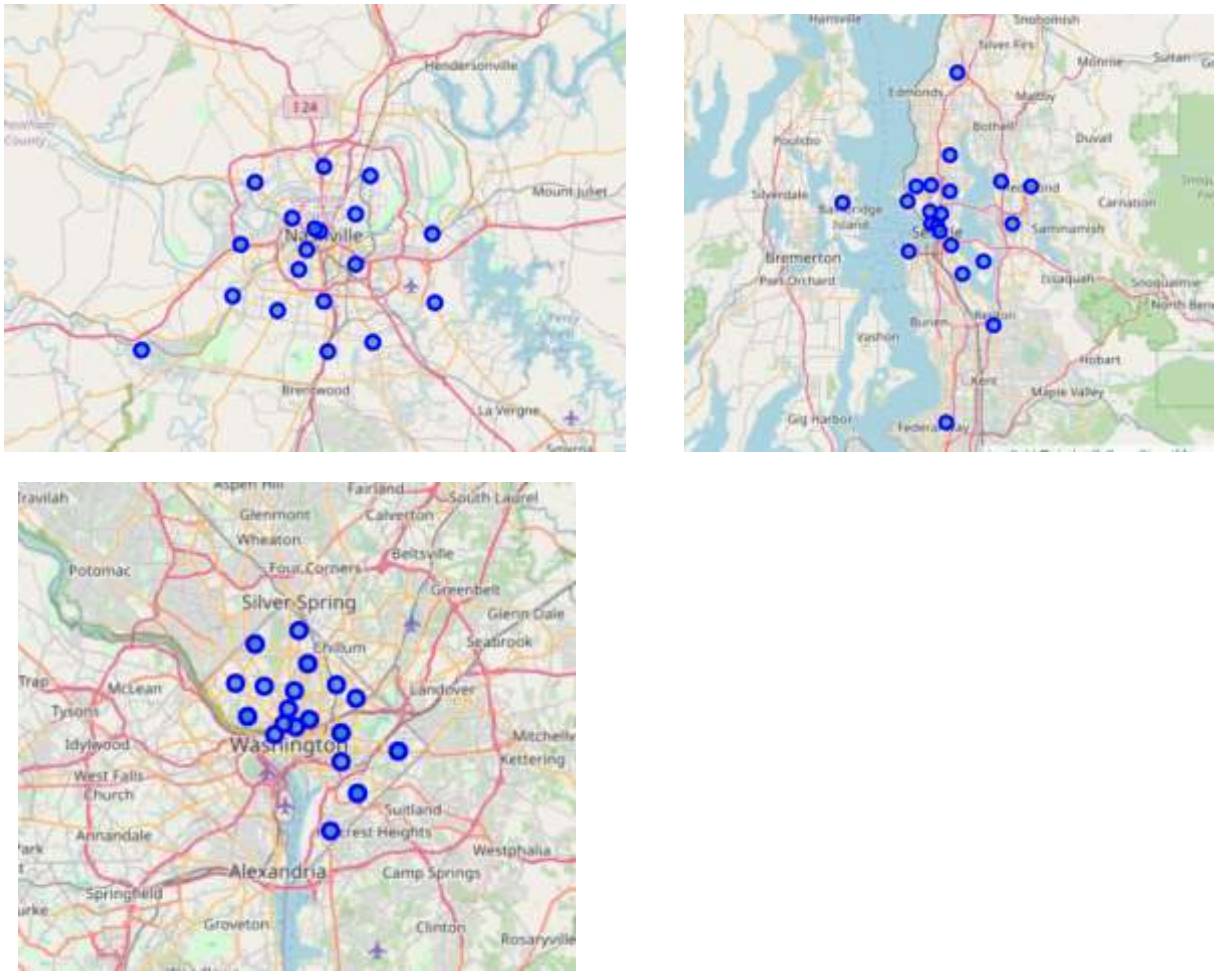


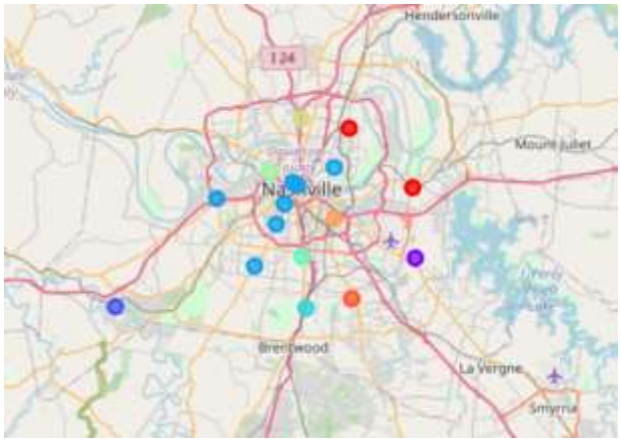
Figure 1: *Visualization of neighborhoods in each city*

Accessing venue details and preparing data for clustering

Using the getnearbyvenues function from earlier in the course, I used each gps coordinate in my data frames to extract information about venues nearby from foursquare. Unfortunately for Nashville, there were two zip code regions that had zero results when querying venues nearby, so they were excluded from subsequent analysis. For each city, I now had a table with rows representing each venue instance within a threshold distance from the gps coordinates of the zip codes provided. The columns of each table each had a name indicating the venue type, and the entries in those columns indicated whether the row instance belonged to that venue type. When each table was grouped by the neighborhood for each entry, each row of the resulting table represented a neighborhood, and the entries under the venue type columns indicated the mean occurrence of this venue type in near the zip code gps coordinates. Now that rows were instances and columns were features, each of the tables were ready to train unsupervised clustering algorithms.

Training clustering algorithms

As an exploratory step, clusters of neighborhoods were generated for each city, then plotted using the folium maps. I chose to use 10 clusters per city to see how the algorithm grouped neighborhoods I understood well in my hometown. The resulting clustering maps are shown below for Nashville, Seattle, and DC, respectively:



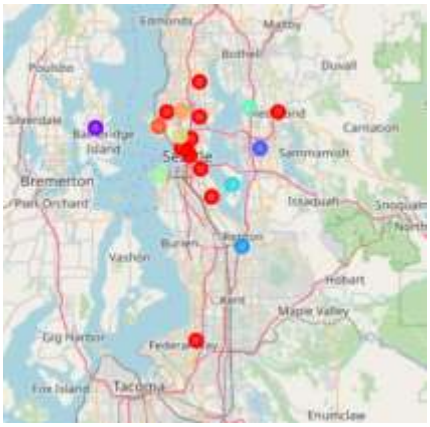


Figure 2: *Visualization of individual clustering of neighborhoods in cities*

Combining neighborhoods from different cities

For the final analysis, neighborhood names and gps information from all three cities were combined, and the venue information queries, clustering, and visualization methods were repeated.

Evaluation of model performance

On the combined run of the city clustering, the cluster number parameter was evaluated using the silhouette score of trained models and the elbow curve technique. Silhouette score was plotted as a dependent variable with cluster number as an independent variable, and the cluster number where the silhouette score stops decreasing rapidly was identified to be the optimal number of clusters.

Results

When data from all three cities were combined into one table, cluster numbers from 3 to 20 were evaluated using an elbow curve. Shown below is the resulting plot, from which I determined that 10 clusters was a good choice for separating the neighborhoods.

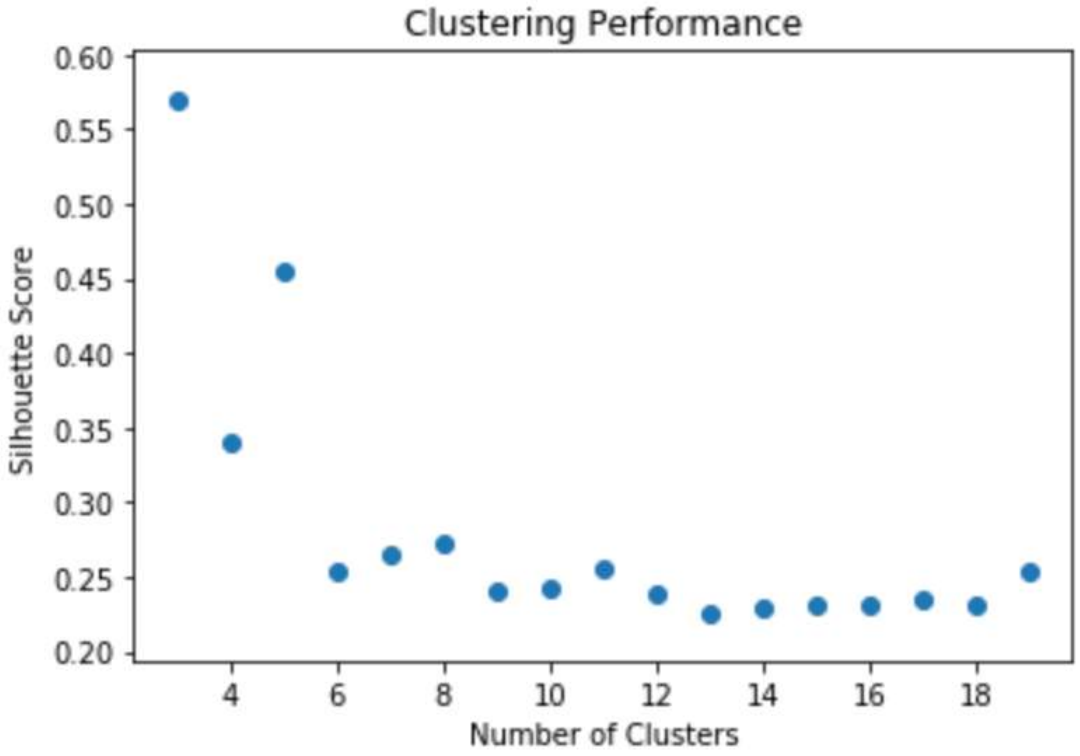


Figure 3: *Elbow curve of clustering model performance.*

After clustering the neighborhoods into 10 groups, I generated a map for each city where neighborhood marker colors were shared between cities and represented the cluster they belonged to.

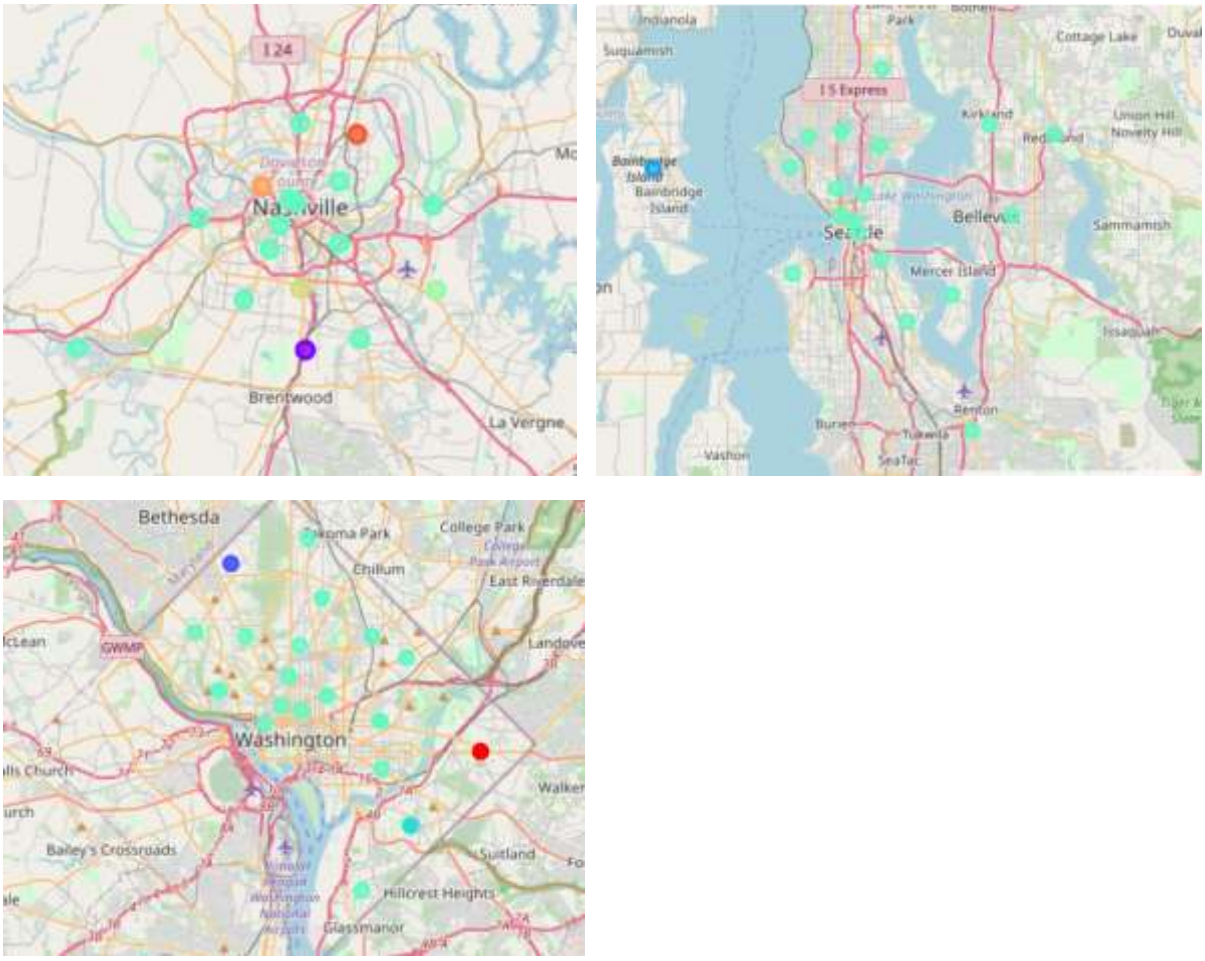


Figure 4: *Visualization of combined cluster neighborhoods in each city, n_clusters = 10.*

Many of the neighborhoods across cities were grouped into cluster five, and the rest of the clusters each only had one neighborhood each. Even though 10 clusters was selected from the elbow curve, I did not find it subjectively helpful to the problem I was answering. I increased the number of clusters to 20, and first looked at my hometown of Nashville to see if the clusters had any subjective meaning to me.

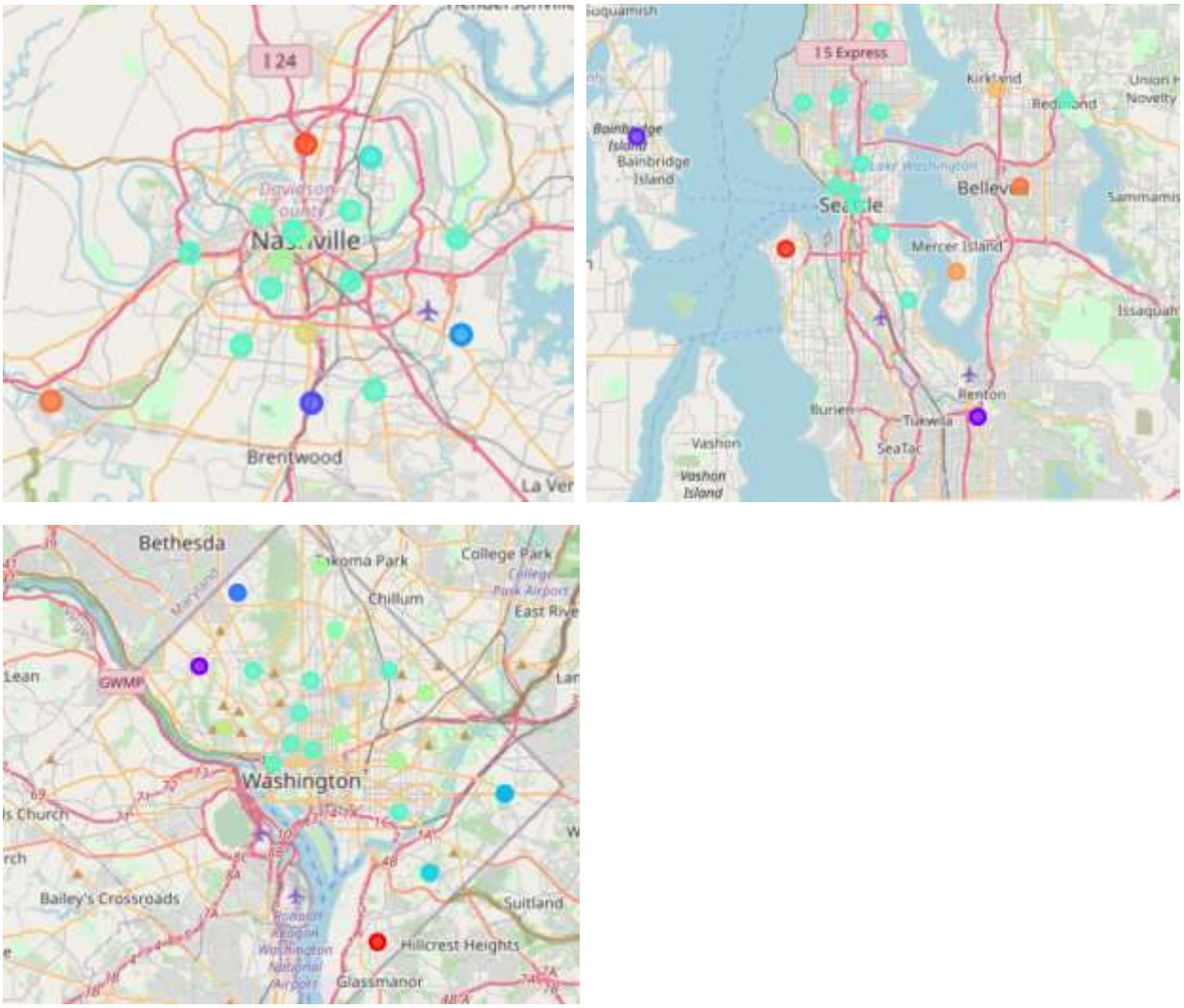


Figure 5: *Visualization of combined cluster neighborhoods in each city, n_clusters = 20.*

Upon examination of the n=20 cluster maps, I was much happier with the results for my hometown. More meaningful separations were made between neighborhoods, and the new relationships made sense to me as someone familiar with the area. The region of "The Gulch" was clustered with "Downtown Nashville," and I find these areas to be similar in real life. Additionally, the teal dots or Cluster #14 (see Table 1) in Nashville were all areas I understood to be suburban in nature, sharing many key characteristics. With a clustering model that better matched my place-knowledge, I decided to apply it to Seattle and D.C. I found repeated patterns of Cluster #14, or the suburbs, throughout Seattle and D.C., as well as areas that matched other regions of Nashville.

As can be seen in the table below, running the clustering algorithm with 20 clusters produced greater separation between neighborhoods and produced more clusters that shared neighborhoods between cities instead of one barely-useful mega-cluster in the n=10 analysis that had over 60 neighborhoods in it.

Cluster Number	City	Number of Neighborhoods in Cluster
0	DC	1
1	DC	1
	Seattle	1
2	Seattle	1
3	Nashville	1
4	DC	2
5	Nashville	1
6	DC	6
7	DC	4
8	Nashville	1
9	DC	8
	Nashville	8
	Seattle	12
10	Nashville	1
11	DC	2
12	DC	7
	Nashville	2
	Seattle	2
13	DC	3
14	Nashville	1
15	Seattle	1
16	Seattle	1
17	Nashville	1
	Seattle	1
18	Nashville	1
19	Seattle	1

Table 1: *Summarization of results with n_clusters=20.*

Discussion

This analysis managed to accomplish some separation between neighborhoods in individual cities and find common threads between neighborhoods across the nation. Although it did not accomplish separation like I was hoping, in which there would be a clear divide between hipster neighborhoods and historic ones, the algorithm did seem to do well identifying what I would consider to be suburbs in Nashville, and separating them from lively food spots like Downtown Nashville and "The Gulch." Noticing similar trends in the other two cities in the analysis, I would feel comfortable drawing parallels between the neighborhoods in Nashville to the ones in DC and Seattle. In this case, I felt the elbow rule did not guide the analysis well. As this tool was meant to help inform a subjective process, I'm placing a lot of stock into my subjective analysis of the results, as prone to confirmation bias as they may be. Though the n=10 clusters analysis may have produced more quantitatively sound separation, I found the n=20 cluster analysis to be more useful.

To demonstrate the subjective validity of cross-city clustering as a way to compare neighborhoods in different cities, below I've added pictures from a neighborhood in each city that belongs to cluster 12. All these regions seems to have newer shops and restaurants, similar cars on the street, well-maintained trees, and similar housing developments.

Figure 6: *The Gulch, Nashville, TN (Cluster 12); Courtesy of Google Street View*

Figure 7: *Area near Judiciary Square/ Howard University, Washington, D.C. (Cluster 12); Courtesy of Google Street View*

Figure 8: *Queen Anne, Seattle, WA (Cluster 12); Courtesy of Google Street View*

Conclusion

In this project, I set out to compare neighborhoods from an unfamiliar place to one I knew by using unsupervised machine learning techniques. Using information about nearby venues from Foursquare, I managed to cluster neighborhoods in different cities together in a way that was subjectively valuable to me. Though this project likely took more work than it would to read articles about these places and gradually update your understandings of these neighborhoods, the data-driven approach appeals to me and has limitless potential. In the future, more training information could be used like climate, air quality index, median income, etc. to improve the clustering algorithm's performance. For now, the results are solely based on venues stored on Foursquare, and I question whether Foursquare is missing a significant number of venues due to low usage. The fact that there were no reported venues within the threshold distance of Belle Meade in Nashville, a place I know to be packed full of coffee shops and the like, makes me seriously doubt how up to date Foursquare's database is. For now, the algorithm can be used to compare any city or group of cities that the user cares to collect data for. With more neighborhoods in the database, the clustering could become more meaningful. It's possible that the algorithm didn't pick up on a cluster as nuanced as hipster areas because it hasn't seen enough examples of these sorts of neighborhoods in the training data to form a statistically meaningful distinction.