Foursquare Data, A Poor Substitute for Identifying Similar Neighborhoods

By Shahrad Ardaghi

March 15, 2019

I. Table of Contents

II. Introduction

a. Background

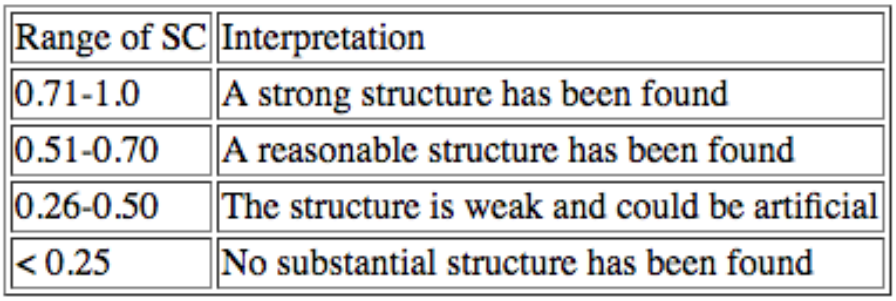
Initially, I intended to replicate similar strategies/methodology from the course's New York City example to Toronto, Downtown LA, and LA city. Depending on the results, I wished to then compare the neighborhoods between each of the respective cities to see which neighborhoods could between cities are similar. My initial motivation was tied to my experiences living in LA and NYC and noticing similarities between certain neighborhoods in each of them. I also believed that such information could prove valuable for people moving from one big city to another by relating giving someone an idea of what kind of neighborhood they would want to move to by comparing it to neighborhoods they might be familiar with.

After settling on a problem and the data that I wanted to acquire I began looking for appropriate sources of data. While this itself proved somewhat challenging, the true surprise came after I had acquired and prepared the data and proceeded to use K-means to cluster it. The result of the clustering proved, repeatedly, to be inaccurate. At first, I began by assuming that there was a problem with the data and proceeded to attempt to try different data sets, different ways of cleaning the data, etc. These efforts also proved fruitless, however, before proceeding any further

I decided to examine the initial example of Manhattan by checking how reliable the clustering was for the given example.

b. Problem

While using the 'inertia' or sum squared error does a good job for determining the ideal number of clusters needed for K-means; it was an unreliable method for comparing the different datasets or even in assisting me in determining how reliable the clustering was. I determined that using the Silhouette Coefficient would be the best method for accomplishing this task. The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample [(b-a) / max(a,b)]. This returns a value between -1 and 1, with negative values indicating a sample that has been assigned to the wrong cluster and values near zero indicating overlapping cluster.[1] Positive values can indicate that a data structure has been found based on the following range:[2]



Range of SC	Interpretation
0.71-1.0	A strong structure has been found
0.51-0.70	A reasonable structure has been found
0.26-0.50	The structure is weak and could be artificial
< 0.25	No substantial structure has been found

I was determined to use the Silhouette Coefficient on the various datasets to determine if any of them seemed to create a strong data structure. The Manhattan dataset was particularly valuable as this dataset and the proceeding clustering were determined/created step-by-step as a tutorial. Thus, if the Silhouette Coefficient found for the data created by the tutorial proved to not create a reasonable structure there would be good reason to believe that we cannot rely on any of the produced clusterings.

c. Interest

This problem should be of interest to anyone who wished to utilize the Foursquare data to categorize a neighborhood. More importantly, it should give pause to anyone who wished to make any sort of business decision based on a neighborhood clustering of a city based only on Foursquare data.

III. Methodology (Data Acquisition & Cleaning)

a. Introduction

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

[2] https://www.stat.berkeley.edu/~spector/s133/Clus.html

As already briefly mentioned, I will be utilizing the data from the New York City example and the Toronto training example. I will be utilizing the saved data frames for each of these cities as they were before we ran the K-Means algorithm. As for my 'original' data, I initially looked for data that would allow me to create a data frame with the zip-codes for neighborhoods in the City of Los Angeles. This process went through several iterations. First, I looked to see if there was an already existing csv file that contained the needed data (similarly to the data that we were provided for New York City). Locating such a file proved fruitless. I then attempted to locate a website that included a list of LA City neighborhoods and their respective zip codes. This also proved fruitless for various reasons. The closest webpage I found with this data proved to contain a myriad of duplicitous data that was not easily removed. The biggest problem was simply a lack of a reliable website that contained LA City neighborhoods and their zip codes.

b. Data Sources

Ultimately, I 'scraped' a Wikipedia webpage that listed all of the notable districts/neighborhoods in LA City. I then attempted to combine this data frame with various other data frames I was playing with to add in the zip-code information. However, this proved fruitless (while I could create data frames I was not sure how reliable they were). Examining the Wikipedia webpage again, I realized that each element of the list of notable neighborhoods/districts had a hyperlink to their respective Wikipedia webpages. Furthermore, for the most part, each Wikipedia webpage contained the longitude and latitude for their respective neighborhood.

Since the Foursquare api requires longitude and latitude this data frame of coordinates and neighborhoods would work perfectly. I ‘scraped’ the Wikipedia page with the list of notable neighborhoods/districts using Beautiful Soup. I captured the name of each neighborhood/district and their corresponding html link. I then proceeded to go through the datagram, using the hyperlink element to scrape through each element’s corresponding webpage and get their respective longitude and latitude.

I used the same method for getting the data frame for LA City and Downtown LA.

c. Data Cleaning

First, the coordinates were not provided in the right format, so I had to convert them into the appropriate format. Second, I examined the neighborhoods that did not provide any coordinates and verified that all such elements could appropriately be excluded from the data frame. Lastly, I created a method to remove 'duplicitous' neighborhoods by checking each coordinate with the other ones and eliminating the ones that were too close to each other (I chose to eliminate the ones that had a distance of 500 meters or less between them). This would ensure that any neighborhoods that shared over 50% of the same area with another neighborhood would be excluded (the method did not discriminate between which neighborhood would get eliminated, keeping the first neighborhood in the data frame and removing the second one).

Using this data frame, I proceeded to get the first 100 venues within 650 meters of each neighborhood. I used the foursquare api, using the coordinates as the input and then capturing the venues' longitude, latitude, name, and category. This will be the data frame that is used for attempting to cluster the neighborhoods.

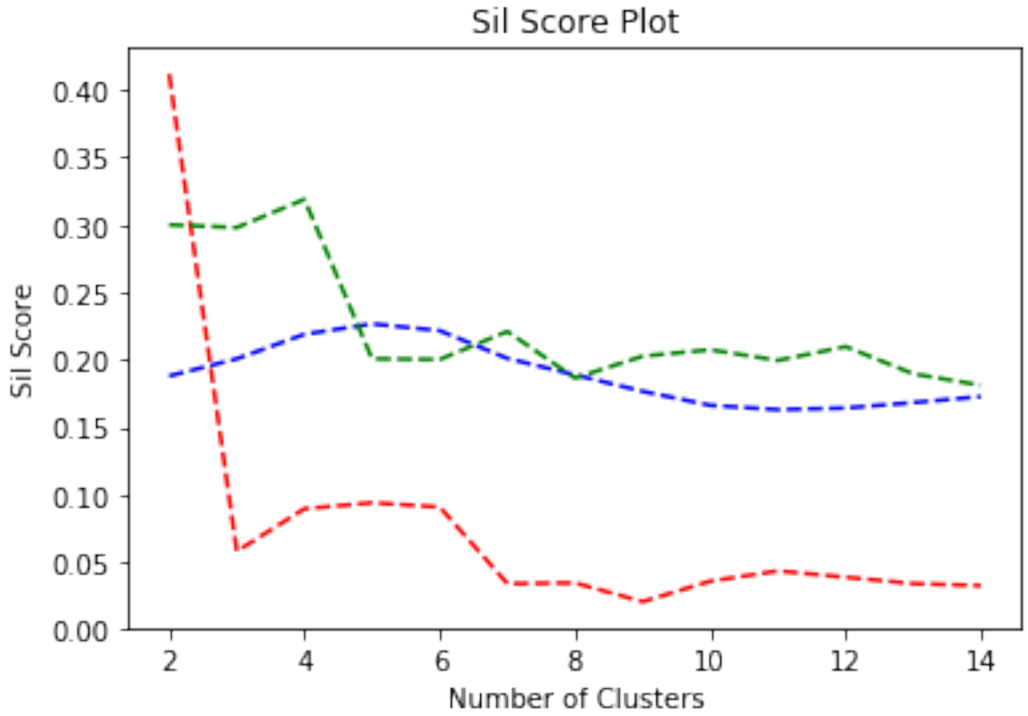
d. Feature Selection/Reduction

I wanted to check if improving the features/categories for the data could improve the results. Thus, I created a method that went through the category hierarchy provided by the Foursquare api and used it to create a mapping of all the sub-categories to their main categories. I was then able to use this method to reduce the features of the data automatically. Aside from automatically changing the categories, I decided to manually go through the Manhattan categories and map them to more general categories as well.

IV. Results

a. NYC Data as Baseline without PCA Reduction

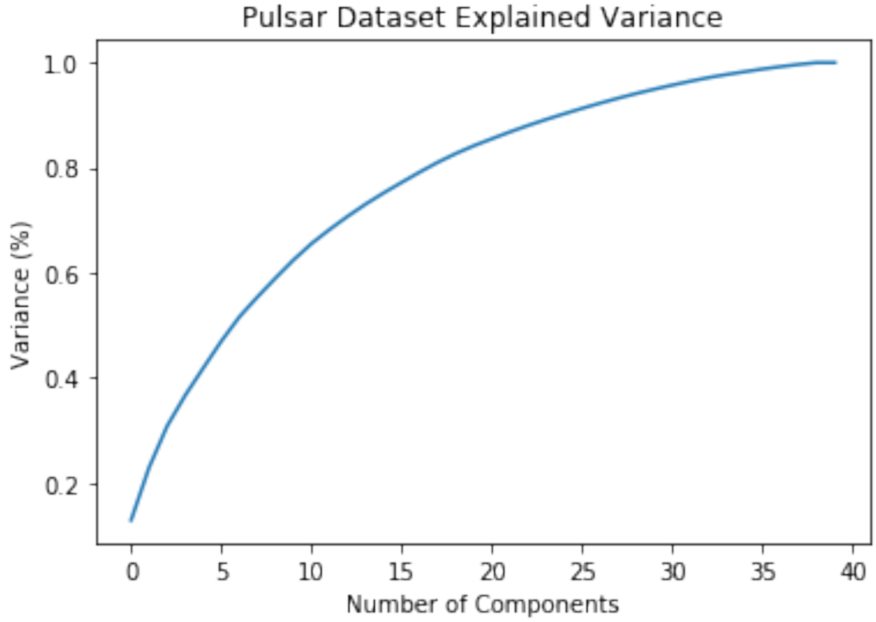
Checking the validity of the Manhattan data was of the paramount importance as it would indicate how valid our attempts at clustering might be. I graphed the Silhouette Coefficient based on the number of clusters for three different datasets: (1) Red for the Manhattan dataframe as provided by the tutorial, (2) Blue for the Manhattan dataframe with my customized categories, and (3) Green for the Manhattan dataframe with the automatically reduced categories.



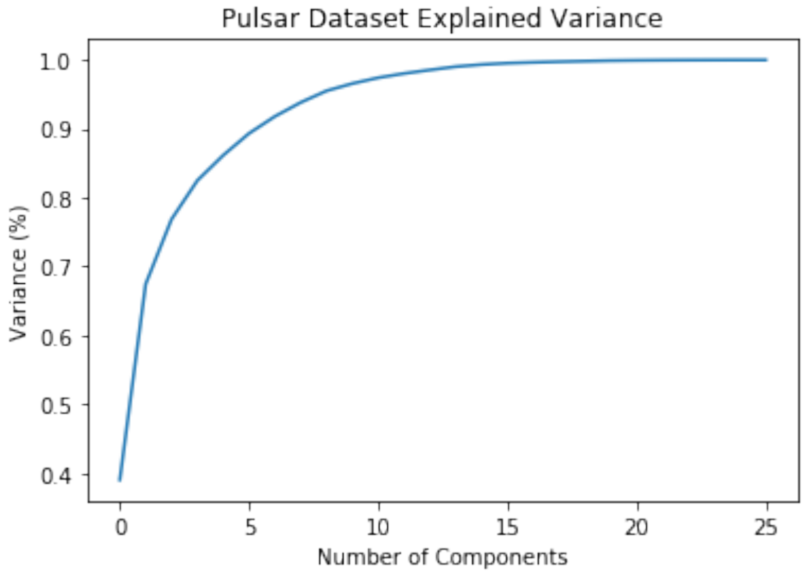
b. NYC Data as Baseline with PCA Reduction

To further attempt to increase the validity/Silhouette Coefficient of the Manhattan data I further reduced the dimensions of each dataframe by using PCA reduction. I determined how many dimensions to reduce each dataframe to by first graphing the variance percentage based on the number of dimensions.

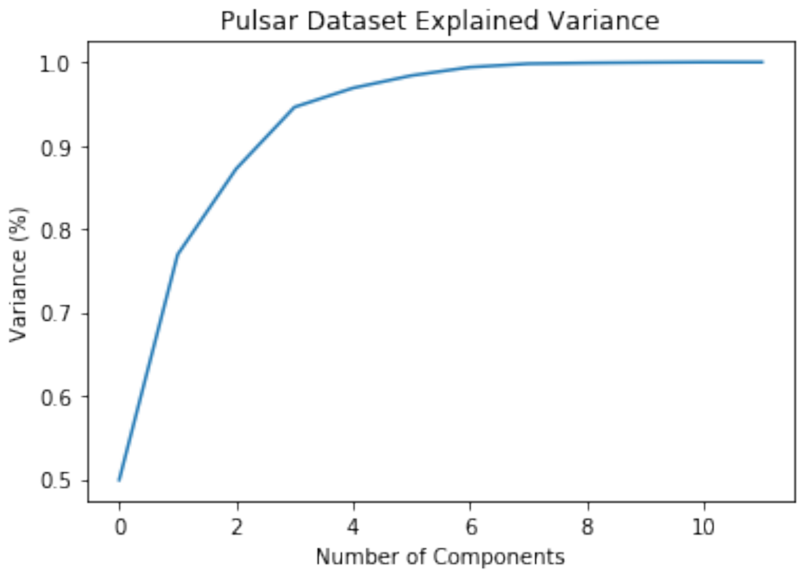
(1) Manhattan dataframe as provided by the tutorial:



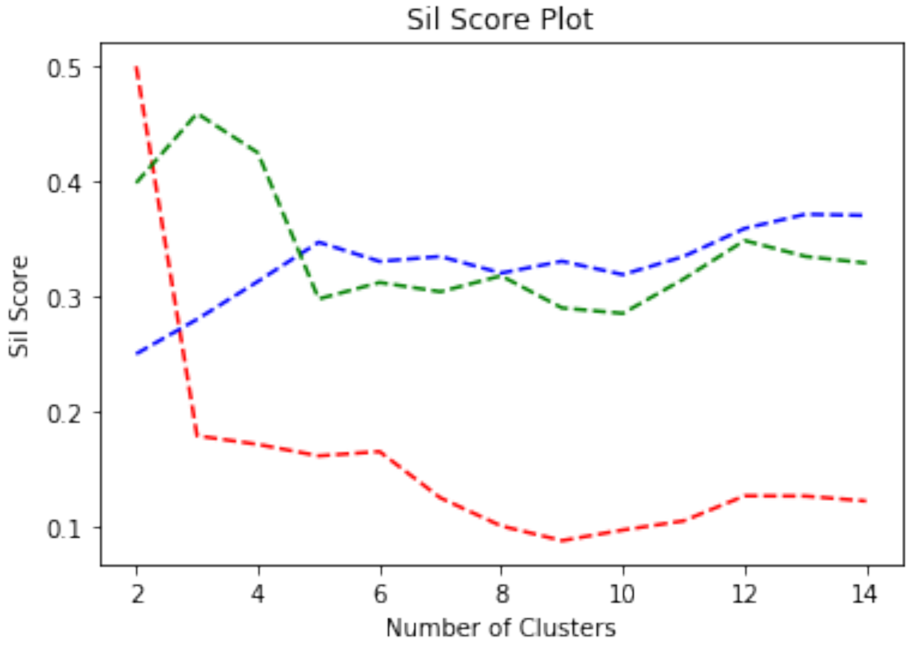
2) Manhattan dataframe with my customized categories:



3) Manhattan dataframe with the automatically reduced categories



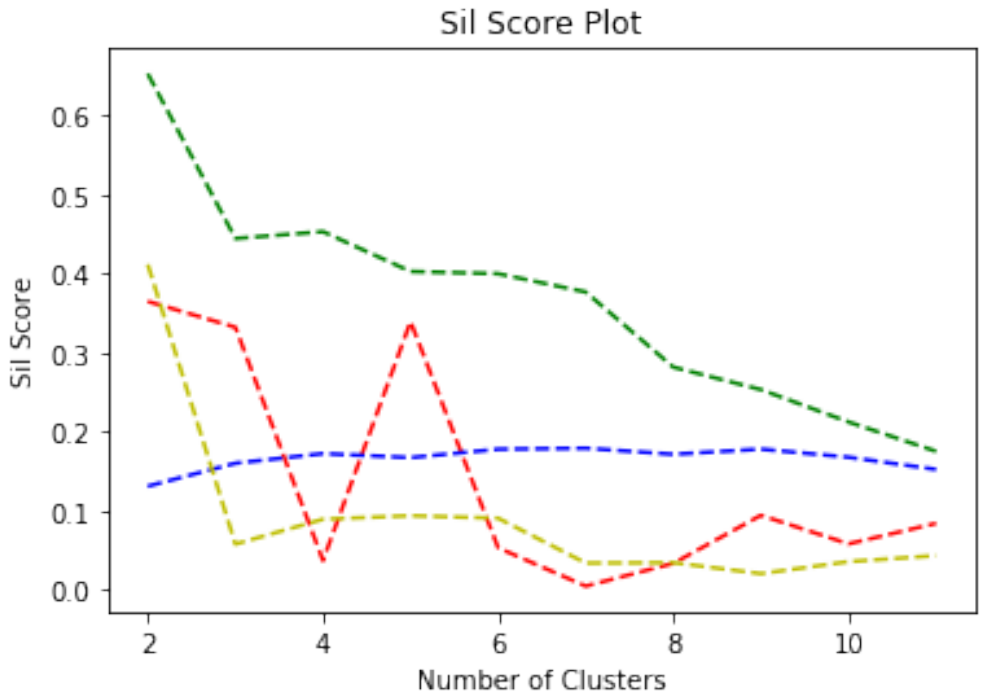
4) With PCA Reduction: Red for the Manhattan dataframe as provided by the tutorial, Blue for the Manhattan dataframe with my customized categories, and Green for the Manhattan dataframe with the automatically reduced categories.



c. No Feature Modification All Data Sets

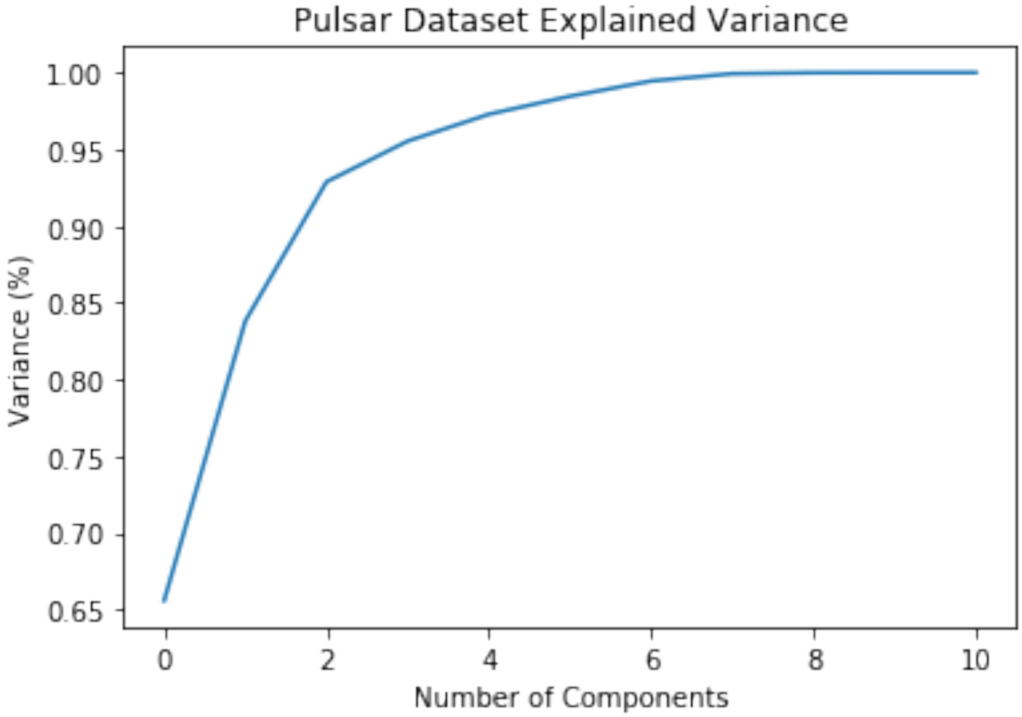
Following these tests I then graphed the Silhouette Coefficient for each of the four dataframes with the original categories in place and no PCA reduction.

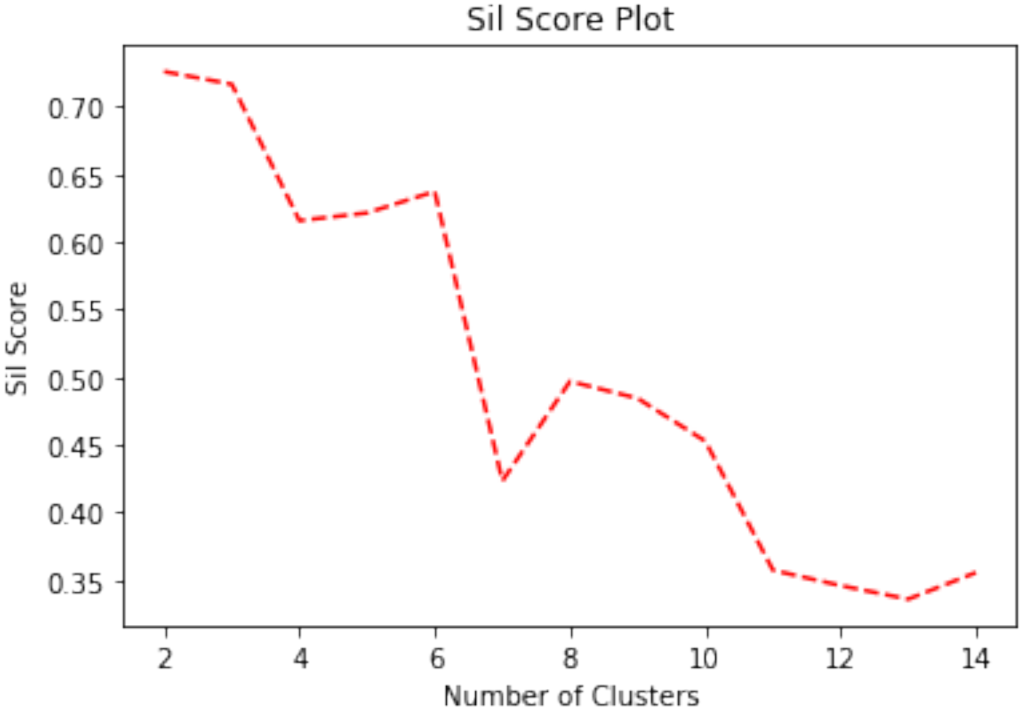
R: LA City --- B: Downtown LA --- G: Toronto --- Y: NYC



d. Toronto Data

Because the Toronto data appeared to be the most promising I decided to run PCA Reduction on the Toronto dataframe with the automatic categories.





V. Discussion

Beginning with the Manhattan dataframe we can see that the clustering as shown to us in the tutorial does not provide us with a clustering that we can have any confidence in. The given dataframe does not exceed a Silhouette Coefficient (SC) of .25, which indicates that no substantial structure has been found.[3] Furthermore, the SS of the customized Manhattan categories also does not exceed .25, while the SC of the automatically created Manhattan categories appears to peak at .32 which only indicates that an artificial structure may exist.

After using PCA Reduction the SC of the provided Manhattan Data still remains under .25, while the SC of the Manhattan Data with automatic categories and customized categories peaks at .45, which still only indicates that an artificial structure may exist

Lastly checking the SC of all the data with their given categories (NYC, Toronto, LA City, LA Downtown), only the Toronto data shows any kind of promise with its SC score achieving a peak of approximately .6. Further examining the Toronto data, I used the automatic categories to reduce the categories and then used PCA reduction to further reduce the dimensions of the dataframe. This produced a SC of approximately .63 for four clusters.

Ultimately, the testing indicates that we cannot use these methods to cluster similar neighborhoods. I believe that these results are due to three main reasons (1) a neighborhood cannot be grouped solely based on the 'venues' located within a neighborhood, (2) the categories provided by Foursquare are inadequate for our purposes, and (3) the 'curse of dimensionality' renders our attempts at analysis to be impracticable.

Several of the different tests were made specifically to examine reasons (2) and (3). Generally speaking, it appears that reducing the number of dimensions (PCA Reduction) increases the Silhouette Coefficient. Furthermore, changing the Foursquare categories into more general categories also does a good job of increasing the Silhouette Coefficient. While changing the categories to more general ones majorly reduces the dimensions (~300 to ~40), it does not appear that the improvement to the Silhouette Coefficient is solely based on dimension reduction but also due to a better selection of dimensions/features, which can also be improved by then doing a PCA Reduction.

VI. Conclusion

Using only Foursquare venue data to try and cluster similar neighborhoods together does not appear to be a valid use of the data or clustering algorithms. It should be possible to create such a cluster if one used various techniques to refine the data. Perhaps the most important technique would be a better breakdown of categories. As it stands the Foursquare categories seem ill-suited for the purposes of clustering similar neighborhoods. Furthermore, neighborhood similarity based solely on the venue information provided by Foursquare seems to be very inadequate. One way to fix this might be to rely more heavily on demographic data instead of 'venue' data.

[3] This is with the exception of two clusters, although, it seems plainly obvious to anyone who has lived in Manhattan that the approximately 40 neighborhoods in the data cannot be split up into only two 'clusters' while providing us with any kind of helpful delineation.

Furthermore, utilizing PCA reduction on a more refined dataset would most likely further assist in creating clustering's with more validity.