Luis Ferrufino

G#00997076

CS 484-002

3/25/20

For my project, I had to produce clusterings on two datasets, dataset1 and dataset2. I did this with a file called lferrufi\_hw3\_cs484.py. As I have done in previous assignments, I comment out different parts of the code, depending on which dataset I am using and which model I am utilizing. I have delineated the different sections of my code with comment headers of the format “#BEGIN \_\_\_ SECTION:” and “#END \_\_\_ SECTION.” The first section is the “KMEANS” section, where I test out each value of K (from 2 to 5) and validate the clusterings by calculating the SSE (sum of squared errors). I found that the best value for K was 4 for datatset1 and 2 for dataset2. The second section is the “DBSCAN” section, where I build a DBSCAN model and validate it with SSE as well. While tuning the Eps parameter for DBSCAN, I noticed that sub-optimal values would group everything as a single cluster, and in some cases label everything as noise; I found that the best values for Eps were 2 for dataset1 and 0.3 for dataset2. The third section is the “CORRELATION” section, where I compute the correlation between a distance matrix and incidence matrix for a given model and dataset. The fifth and final section is the “SILHOUETTE” section, where I compute the average silhouette coefficient for all points in a given clustering.

Here are the results of my cluster validation:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUM OF SQUARED ERRORS (SSE) | | | | | | | | | | | | | |
| DATASET 1 | | | | | | | | DATASET 2 | | | | | |
| Kmeans | | | | | DBSCAN | | Kmeans | | | | | | DBSCAN |
| K=2 | K=3 | K=4 | K=5 | Eps=2 | | K=2 | | | K=3 | K=4 | K=5 | Eps=0.3 | |
| 247825.774 | 2220.642 | 20151.7 | 24818.297 | 57.629 | | 128.91688 | | | 138.83502 | 151.71798 | 150.78416 | 279.5751 | |

As we can see, for dataset1, DBSCAN has the lowest SSE, 57.629; the second contender was Kmeans with K=4 and SSE=20151.7. For dataset2, Kmeans with K=2 had the smallest SSE, SSE=128.91688; DBSCAN had the largest SSE, SSE=279.5751.

|  |  |  |  |
| --- | --- | --- | --- |
| CORRELATION | | | |
| DATASET 1 | | DATASET 2 | |
| Kmeans, K=4 | DBSCAN, Eps=2 | Kmeans, K=2 | DBSCAN, Eps=0.3 |
| -0.714 | 0.01576 | -0.313 | -0.18347 |

For correlation, -1 means a perfect negative correlation, 0 means no correlation at all, and +1 means a perfect positive correlation. Thus, for both dataset1 and dataset2, Kmeans did better than DBSCAN.

|  |  |  |  |
| --- | --- | --- | --- |
| SILHOUETTE COEFFICIENTS | | | |
| DATASET 1 | | DATASET 2 | |
| Kmeans, K=4 | DBSCAN, Eps=2 | Kmeans, K=2 | DBSCAN, Eps=0.3 |
| 0.171 | -0.752 | -0.58454 | -0.64174 |

Considering how silhouette coefficients are on a scale from -1 to 1, with -1 being the worst and 1 the best, Kmeans K=4 for dataset1 did the best, although it could be much better; for dataset2, Kmeans=2 did better. The other silhouettte coefficients are negative, so this could suggest their corresponding clusterings have too few clusters.

In summary, since dataset1 had the best correlation score (when using Kmeans with K=4), and the best silhouette coeffiecient (when using Kmeans = 4), I would say that dataset2 was the noisy dataset and dataset1 was the dataset1 with actual structure. Furthermore, since Kmeans consistently had better correlation scores and silhouette coefficients than those of DBSCAN, I would say Kmeans preformed better for these datasets (especially for dataset1).

I am happy to report that, unlike previous assignments every time I needed to create a clustering and evaluate it, it never took more than half of a minute. The only libraries I used were numpy, sklearn, and math. I converted the datasets into csv files (taking advantage of the fact that spaces can be separators), and converted them into numpy arrays in my script. The corresponding files are dataset1.csv and dataset2.csv.