Brian Koehler
UFID 7551-8931

CAP4770 Assignment 5

K-Means

Parameters

Provided transposed normalized data set and set number of clusters to 12. The number of clusters was determined by finding the score of clustering with values from 2 to 40 and taking that which produced the highest score (in this case it was 12, but generally varies from 10-30).

Best Score

Producing 12 clusters provided a silhouette score of roughly 0.24405. This score was determined by utilizing the *silhouettes* function of the Clustering package.

Running Time

According to Jupyter, it took 8.4 seconds to produce and score 2 clusters to 40 clusters.

Interpretation

Due to the data being in 27 dimensions, it is difficult to visualize and draw a clear conclusion from the clusters that were produced. One important note was that there were around 4 or so centers where many of the features had values of either 0.0 or 1.0, indicating the polarity of some of the data.

Outliers

Outliers was identified in clusters 4, 6, and 10. These outliers were identified by finding the 3 smallest silhouette scores and matching them to the assignments vector produced by *kmeans*. This assignment vectors denotes which cluster each data point was assigned to.

K-Medoids

Parameters

Provided the distances matrix and set the number of clusters to 20. The distances matrix was produced using the *pairwise* function and providing it the transposed normalized data set. It provided the Squared Euclidean distances. Because the transposed data was provided, the dimensions of it were set to 2. The number of clusters was determined by finding the score of clustering with values from 2 to 40 and taking that which produced the highest score (in this case it was 20).

Best Score

Producing 20 clusters provided a silhouette score of roughly 0.18727. This score was determined by utilizing the *silhouettes* function of the Clustering package.

Running Time

According to Jupyter, it took 17.9 seconds to produce and score 2 clusters to 40 clusters.

Interpretation

K-Medoids involves finding some number of data points such that the total distance between each point to the closest "medoid" is minimal. In this case, 20 clusters were produced, so that demonstrates that there were an average amount of cancer scenarios that could be said to be similar to each other, as 20 is mid-range in the spectrum of possible clustering.

Outliers
Outliers was identified in clusters 1, 1, and 10. These outliers were identified by finding the 3 smallest silhouette scores and matching them to the assignments vector produced by *kmedoids*. This assignment vectors denotes which cluster each data point was assigned to.

Affinity Propagation

Parameters
Provided the distances matrix. The distances matrix was produced using the *pairwise* function and providing it the transposed normalized data set. It provided the Squared Euclidean distances. Because the transposed data was provided, the dimensions of it were set to 2.

Best Score
The algorithm resulted in 90 clusters with a silhouette score of only 0.025714. This score was determined by utilizing the *silhouettes* function of the Clustering package.

Running Time
According to Jupyter, it took 52.8 seconds.

Interpretation
This clustering method is like K-Medoids in that it examines similarities but produced significantly different clusters. Basing upon this clustering algorithm, it can be stated that the cancer scenarios were very different from one another, which could lead to much more individualized work in the hospitals.

Outliers
Outliers was identified in clusters 7, 9, 52. These outliers were identified by finding the 3 smallest silhouette scores and matching them to the assignments vector produced by *affinityprop*. This assignment vectors denotes which cluster each data point was assigned to.

Hierarchical Clustering

Parameters
Provided *hlcust* the distances matrix. The distances matrix was produced using the *pairwise* function and providing it the transposed normalized data set. It provided the Squared Euclidean distances. Because the transposed data was provided, the dimensions of it were set to 2.

When cutting the tree, the *cutree* function was given the tree produced by *hlcust* and a cluster amount of 2, because that produced the highest score.

Best Score
The algorithm resulted in 2 clusters with a silhouette score of 0.22585. This score was determined by utilizing the *silhouettes* function of the Clustering package.

Running Time
According to Jupyter, it took 10.4 seconds to produce and score 2 clusters to 40 clusters.

Interpretation
This clustering algorithm did a very poor job of splitting the cancer patients into smaller groups, as it essentially only halved them. In using this clustering method, hospitals would not have to perform as much individualized work, because many of the cases are viewed as in the same class.

Outliers
Outliers was identified in clusters 1, 1, and 2. These outliers were identified by finding the 3 smallest silhouette scores and matching them to the assignments vector produced by *cutree*. This assignment vectors denotes which cluster each data point was assigned to.

GMM

Parameters
The two-parameter constructor was utilized. The first parameter was set to 35, because that number of clusters resulted in the highest score. The second parameter was set to 27, because that was the number of features in the data set. When updating the model, the model itself was passed and the normalized data that was constructed using the *minmaxscaler* from Sci-kit Learn.

Best Score
The algorithm resulted in 35 clusters with a log-likelihood of 0.11334. This score was determined using the *avll* function of the GaussianMixtures package.

Running Time
According to Jupyter, it took 3.3 seconds to produce and score 2 clusters to 40 clusters.

Interpretation
This clustering algorithm performed well in that it produced a good number of clusters with a reasonable score. This algorithm would provide a good balance for hospitals to cluster cancer cases, as there is a good amount of individualism, but not too much to the point that it would be overwhelming.

Outliers
Outliers was identified at indices 21, 22, and 35. These outliers were identified by finding the 3 smallest log-likelihoods.