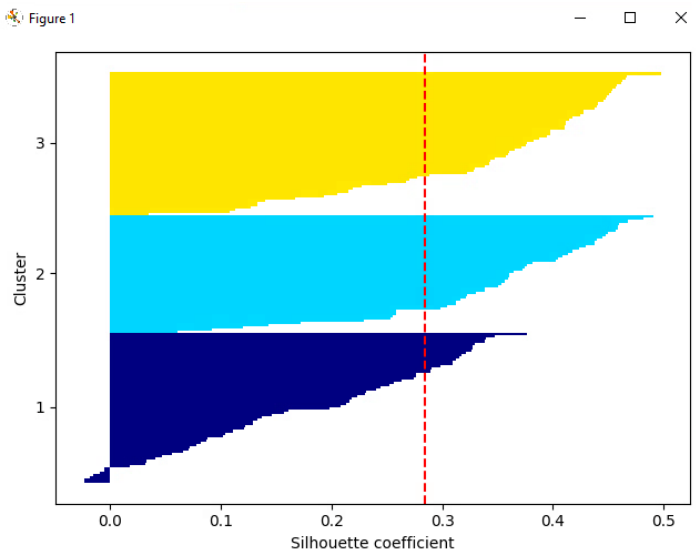
Thomas Lamont

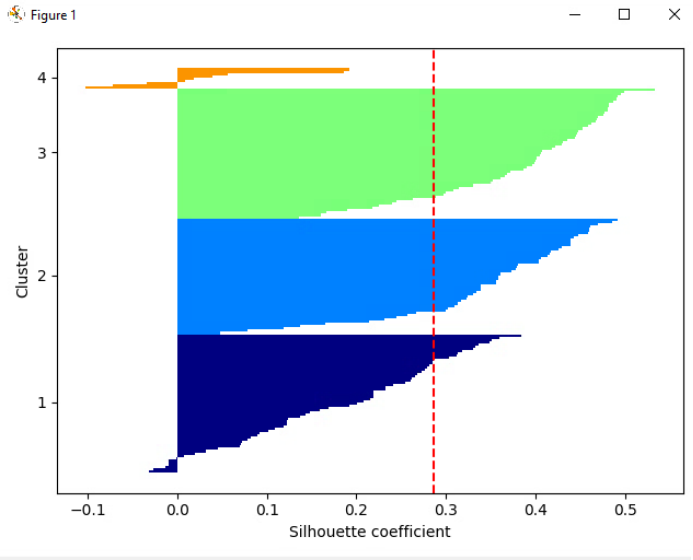
CS4275

Small Project 2

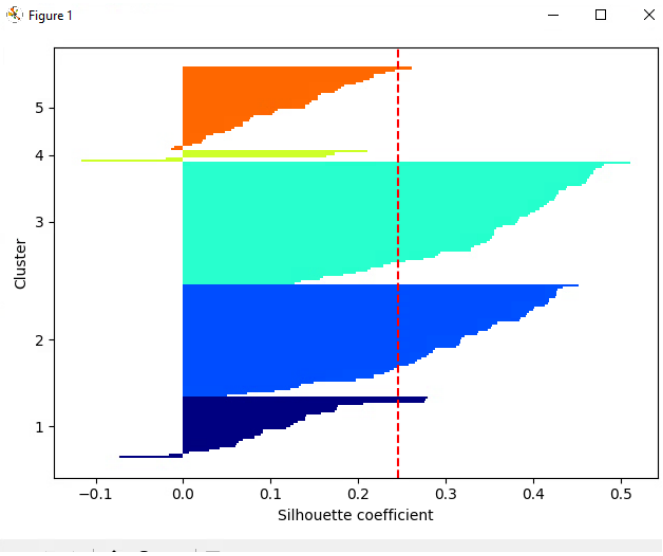
We are starting with wine data that is structured, but not labeled, this means we do not know how many classifications exist in the set. To handle this unsupervised learning problem, we can use a silhouette graph to help us determine the number of clusters in our data. We should first standardize so that our clustering algorithms that rely on distance are consistent.



In our first plot with 3 clusters, we see three relatively well separated clusters, by their coefficients the yellow and cyan clusters are well separated. The blue cluster has the lowest average coefficient, so it is less well defined than the other 2. Our average coefficient sits just below .3

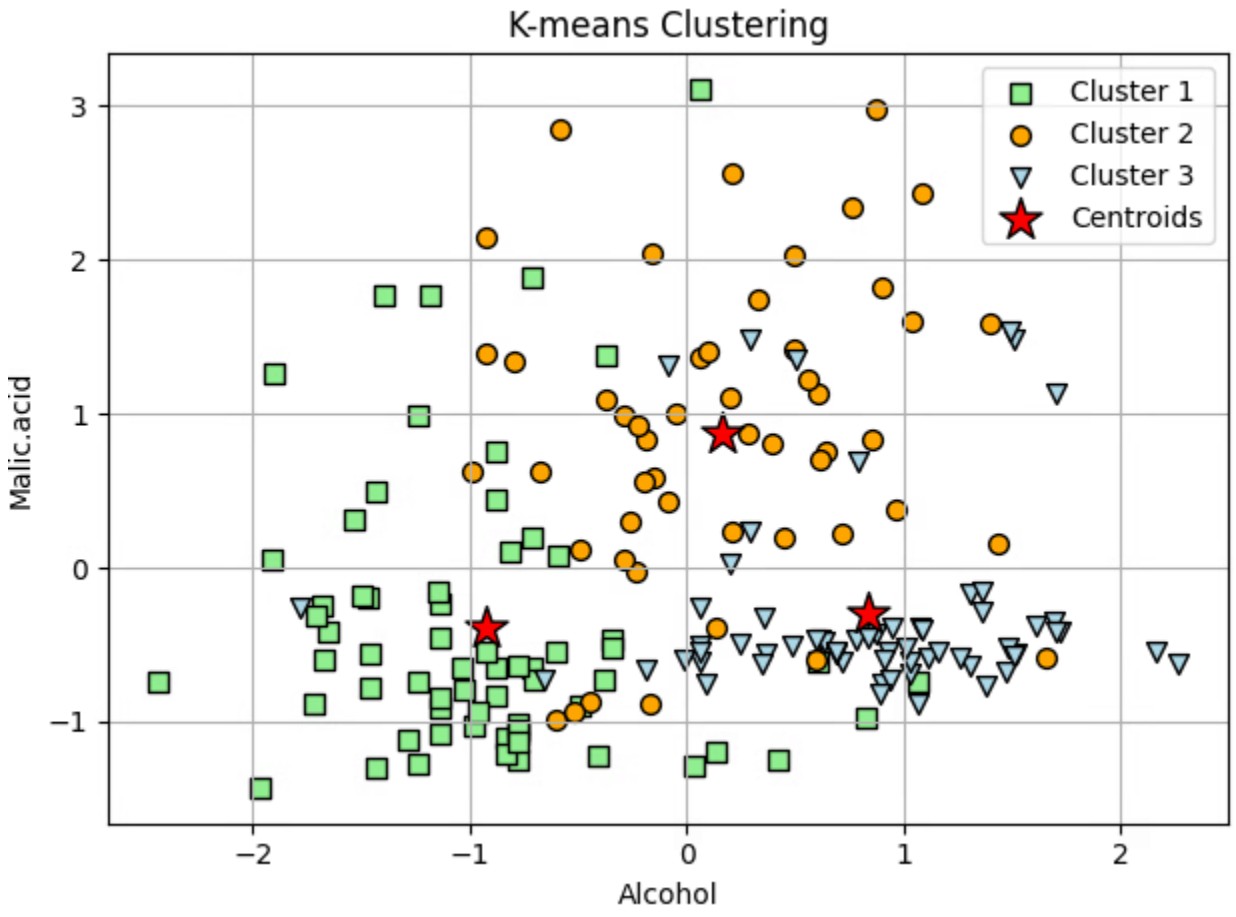


With 4 clusters, the average coefficient is in a similar spot. We again see two strong clusters and the slightly weaker third cluster, but there is also a 4th cluster that is small and quite weak, there is not good separation between the boundaries of the 4th cluster. The addition of this 4th weak cluster implies that 3 is the more likely correct number of classifications.

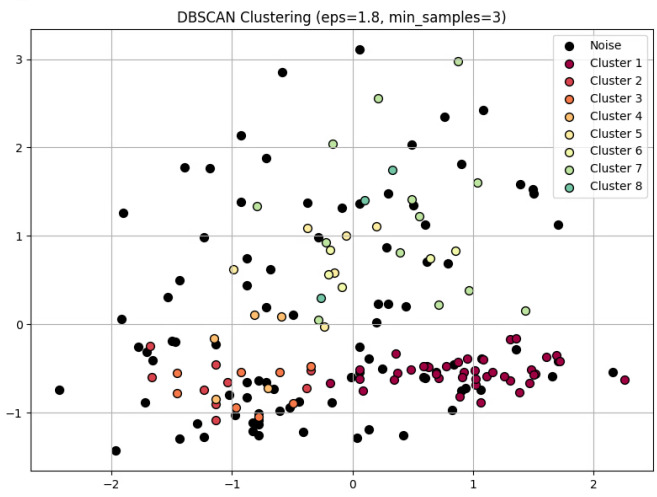


At 5 clusters the average coefficient drops significantly to just under .25. We see the same two well defined clusters from earlier, but the 3rd cluster has been split into two. There are several points with negative values in the small, less defined clusters. These graphs imply that 3 is probably the correct number of clusters and we can use this information going forward.

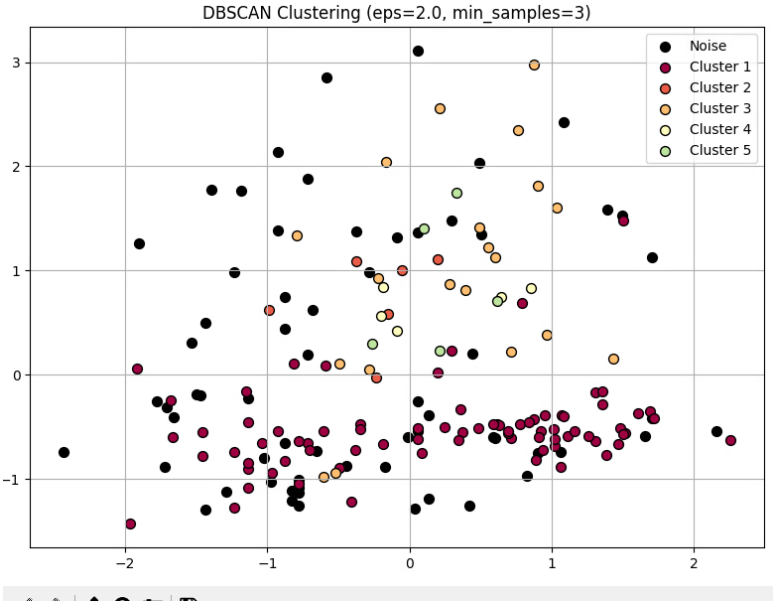
Starting with a K-means approach we can see three relatively well separated clusters. Each region is distinct with some overlap. Cluster one and two occupy a larger portion while cluster three is more tightly grouped. The centroids illustrate the separate regions. Overall it seems that the k-means strategy is working well for classifying our wines.



DB scan was a bit trickier to get working. With epsilon values under 1.8, the vast majority of points were shown as noise and as such not being clustered. Some noise is reasonable and expected, as not all clusters will be tight, the k-means model tries to classify every point which leads to overlap, and outliers being considered in the model. I knew from the silhouette charts earlier that there would ideally be 3 groups, to find decent hyperparameters for the db-scan model, I trained the data several times with different epsilons and min sample sizes.

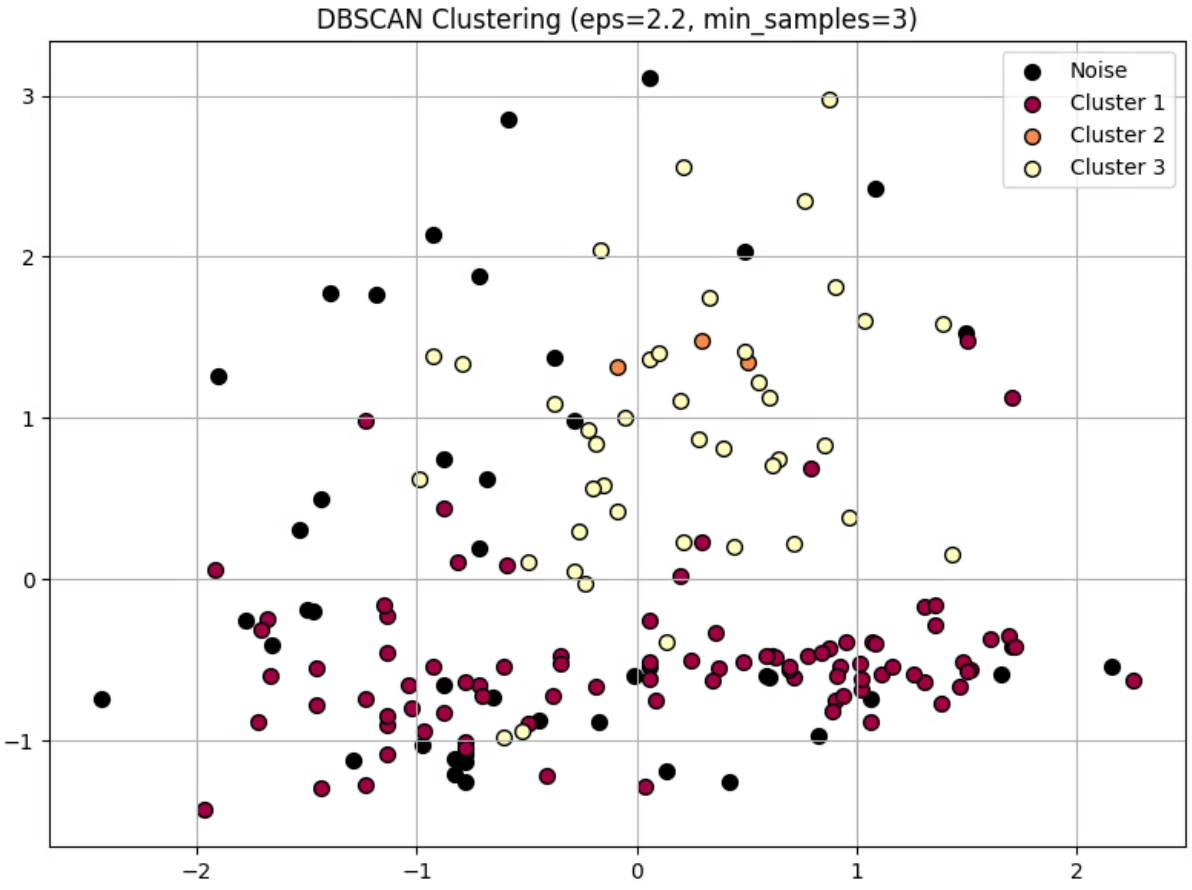


With an epsilon of 1.8, there was a lot of noise and a lot of small clusters.



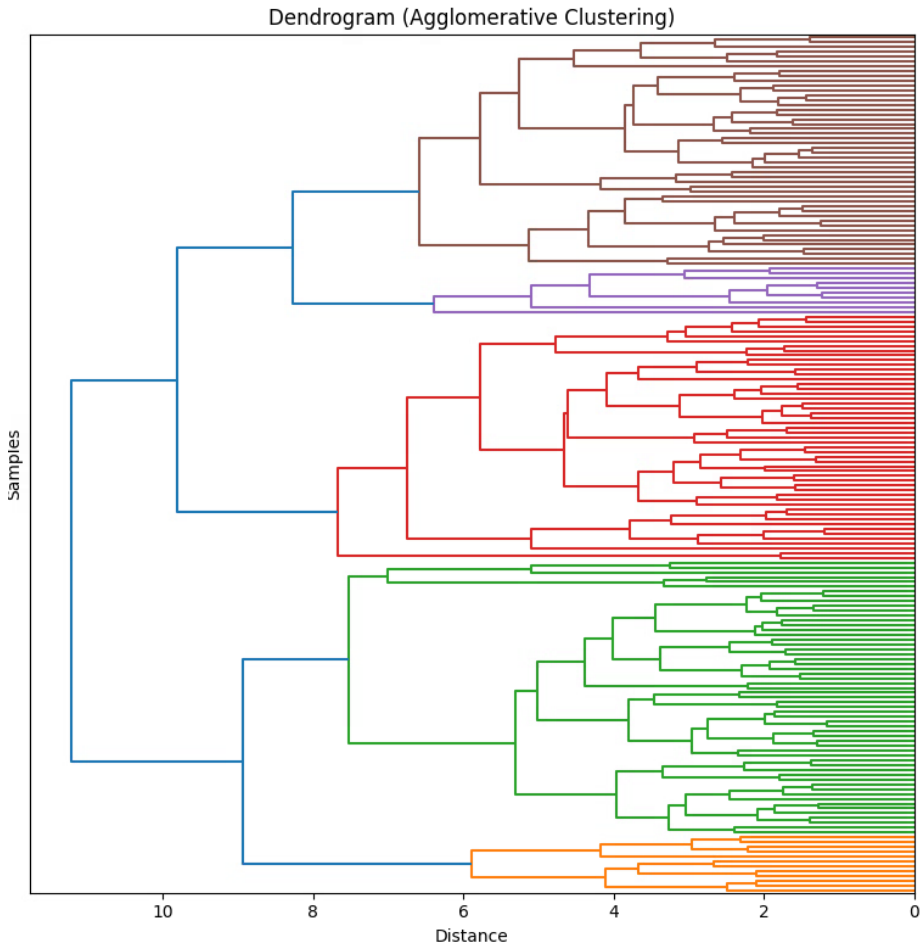
Increasing the epsilon to 2.0 helped a bit, there is a strong cluster 1 and less noise.

Further increasing the epsilon to 2.2 gave the sought-after 3 clusters, but we can see one two well defined clusters and one small third cluster in the middle. Increasing the epsilon further led to the elimination of the third small cluster, and eventually the assimilation of all points into a single cluster.

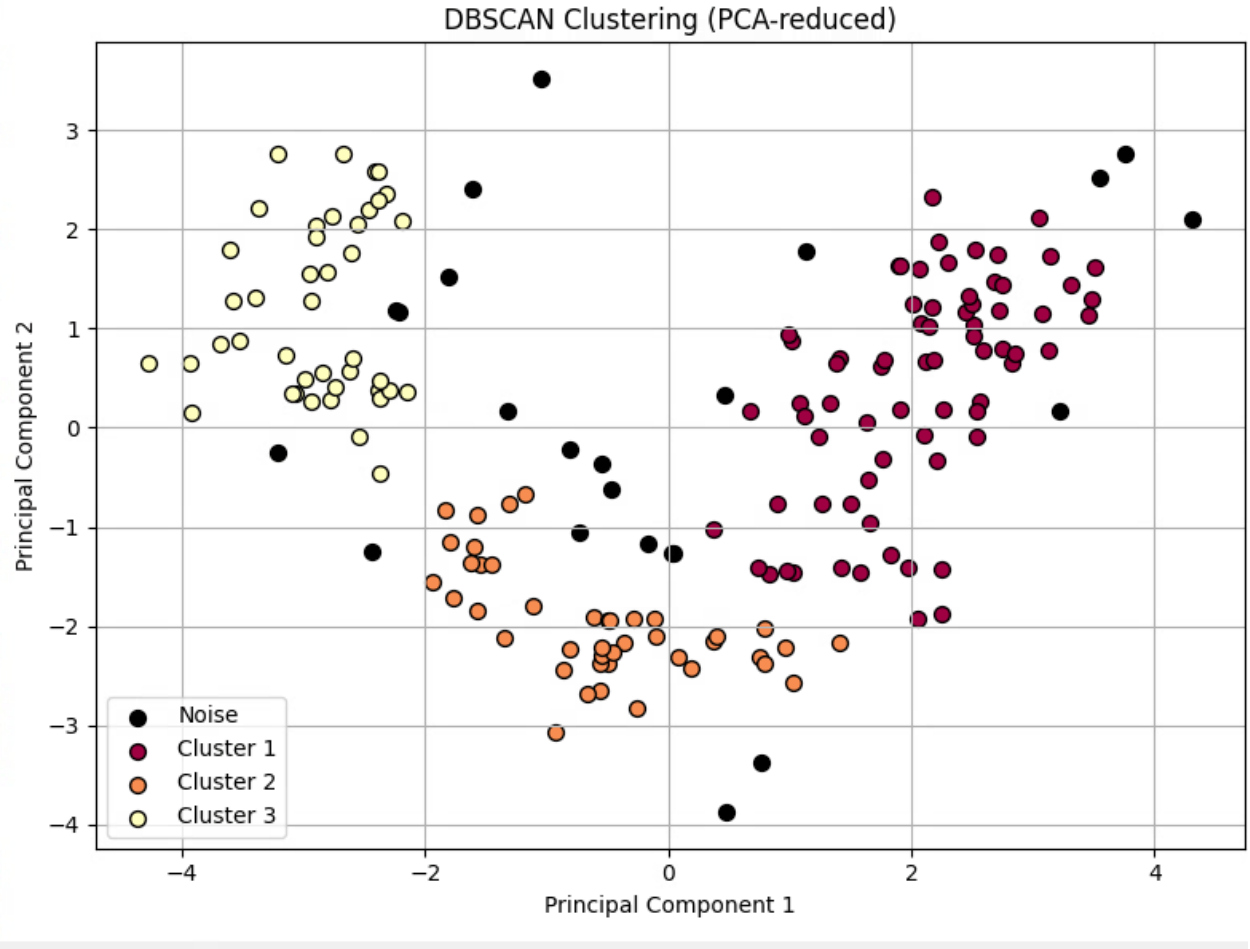


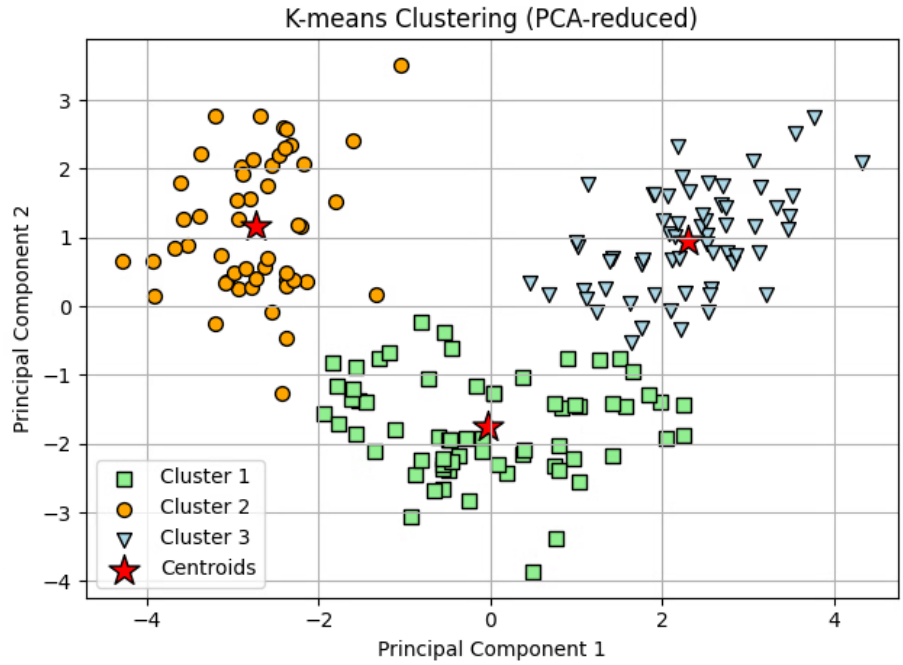
The high amount of noise, even with a high epsilon value, implies that there are a lot of outliers if our db-scan model is optimized.

Our agglomerative clustering dendrogram starts with each sample represented individually and slowly merging groups with their closest neighbors until eventually all groups merge at the highest distance. Starting at a distance of around 6 we see 5 distinct clusters, and at 9 distance we see 3. This lets us visualize how similar clusters are to one another as well as which clusters are most and least alike.

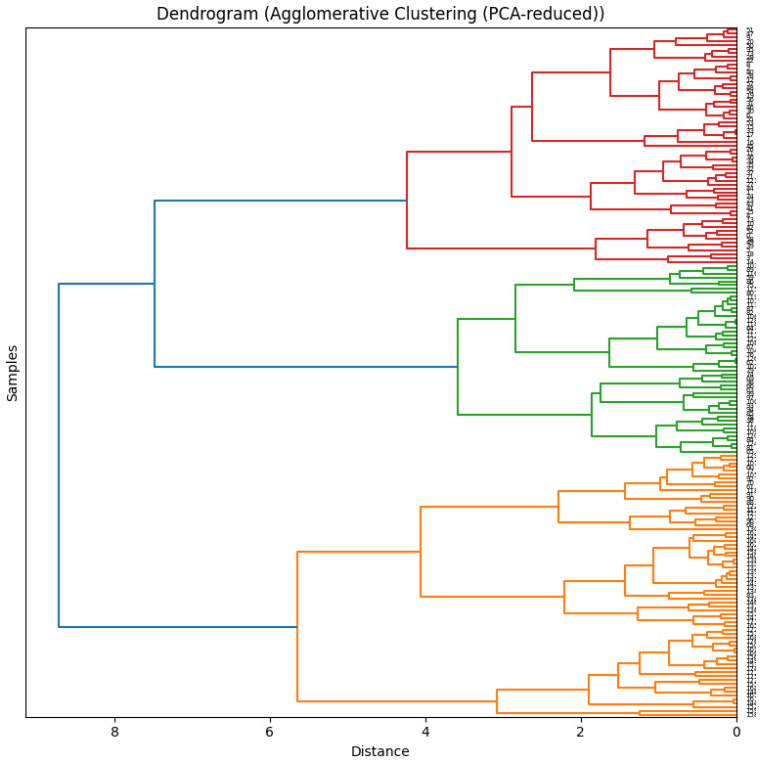


After the relative lack of success with the db-scan model, I wanted to try lowering the dimensionality from the 13 features in the wine-data set, to a more condensed two dimensions. This will cause points to be closer together in space and could help the db-scan model make a clearer prediction. With the initial high epsilon value from the original model, every point was homogenized into a single cluster. This was a good sign as it implied a lowering of the epsilon was called for. I reduced the epsilon to a more reasonable .55 with a minimum group size of 5 and the results were immediately improved. We can see 3 distinct clusters with only a few outliers.





After the dimensionality reduction, the k-means and agglomerative clustering models also improved their predictions. We can see much less overlap in the k-means model and the three clusters are visible by a distance of 6 in the dendrogram.



Overall, the reduction of dimensionality seemed like a success. This makes sense for clustering models, with the curse of dimensionality, points get exponentially further apart as the number of dimensions increases. The original set has 13 features, and if we try to build a model based on Euclidean distance and all the features, the points are too far apart.