# **Unsupervised classification of Iris Species**

**Summary Stats:**

count mean std min 25% 50% 75% max

sepal\_length 150.0 5.84 0.83 4.3 5.1 5.80 6.4 7.9

sepal\_width 150.0 3.05 0.43 2.0 2.8 3.00 3.3 4.4

petal\_length 150.0 3.76 1.76 1.0 1.6 4.35 5.1 6.9

petal\_width 150.0 1.20 0.76 0.1 0.3 1.30 1.8 2.5

sepal\_length has a mode of 5.0, which makes up 6.67% of non-NaN values

sepal\_width has a mode of 3.0, which makes up 17.33% of non-NaN values

petal\_length has a mode of 1.5, which makes up 9.33% of non-NaN values

petal\_width has a mode of 0.2, which makes up 18.67% of non-NaN values

species has mode(s) of: 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica', which makes up 33.33% of non-NaN values

**Number of Unique Values:**

sepal\_length has 35 unique values out of 150

sepal\_width has 23 unique values out of 150

petal\_length has 43 unique values out of 150

petal\_width has 22 unique values out of 150

species has 3 unique values out of 150

**Number of Null Values:**

sepal\_length has 0 null values out of 150: 0.0%

sepal\_width has 0 null values out of 150: 0.0%

petal\_length has 0 null values out of 150: 0.0%

petal\_width has 0 null values out of 150: 0.0%

species has 0 null values out of 150: 0.0%

**Feature Data Types:**

sepal\_length is float64 data

sepal\_width is float64 data

petal\_length is float64 data

petal\_width is float64 data

species is object data

A blue and white screen

Description automatically generated with medium confidenceA blue and white rectangular object

Description automatically generatedA blue and white rectangle

Description automatically generatedA blue and white rectangle

Description automatically generated

Figure 1 – Distribution of numeric features.

A line drawn on a white surface

Description automatically generatedA red line with blue dots

Description automatically generatedA blue dotted line with a red line

Description automatically generatedA blue and red dotted line

Description automatically generated

Figure 2 – The QQ plots of numeric features, the closer the blue points to the red line the closer the distribution is to normality.

All of the numeric features have |skewness| < 0.4, and this, coupled with visual inspection of distributions from figures 1 and 2, indicates that these are close to normally distributed.

**Collinearity**

Each of the features had a VIF (Variance Inflation Factor) over 50 and this indicates a high degree of collinearity.

A screenshot of a computer screen

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a graph

Description automatically generatedA screenshot of a computer screen

Description automatically generated

Figure 3 – Comparison of logarithmic Variance Inflation Factor with and without each numeric feature to determine extent of linear multicollinearity.

A blue rectangular object with a black background

Description automatically generatedA blue rectangle with black border

Description automatically generated

Figure 4 – the total number of outliers determined by different statistical methods with varying levels of robustness to outliers.

Based on both Z-score and MAD (Median Absolute Deviance), only one feature contained outliers, and as the number of outliers is very small (roughly 1.3% of the total data for that feature) the outliers were retained for clustering.

The first attempt at clustering was DBSCAN, which was problematic as determining the number of min samples and the magnitude of epsilon was time consuming. The data was clustered based both on the Euclidean distance and the Gower distance.

**DBSCAN – K-Neighbours plot**

For the purpose of determining the parameters for clustering, two plots were created. Firstly, a k-distance plot, this aids in determining the number of min\_samples and usually you would look to identify an elbow point; however, this was unclear in the below graphs and when attempting to visually ascertain the correct values the algorithm failed to cluster, so I calculated the mean and standard deviation of distances and any points that are over 2 standard deviations away from the mean are highlighted in red, whereas those in green indicate a number position that is within 90% of the range, and these were also investigated.

A line of dots and lines

Description automatically generated

Figure 5 – Example of a K-distance plot with outliers determined by MAD removed.

For the purpose of finding the number of min\_samples and eps those points identified as outliers were temporarily removed, as these may have affected the outcome.

**DBSCAN – Reachability**

Subsequently to identifying potential min\_samples a reachability plot was graphed for each potential min\_sample, below is an example but this same graph was plotted for each min\_sample. Note that that the number of neighbours is 7, however the final number of min\_samples from this graph is 8 as reachability is based on KNN and that does not include the initial point whereas DBSCAN is inclusive of the self-data point.

A line graph with different colored dots

Description automatically generated

Figure 6 – Example of a reachability to determine the magnitude of epsilon. The green values indicate those that are over a threshold of +0.5std over the mean and those in red are the 20 largest differences in distance.

**DBSCAN – Results**

As DBSCAN is also sensitive to the distance metric used to compute the neighbourhoods of data points it was performed using Gower distance and the Euclidean distance (default for DBSCAN) and these yielded very different results.

Euclidean Distance

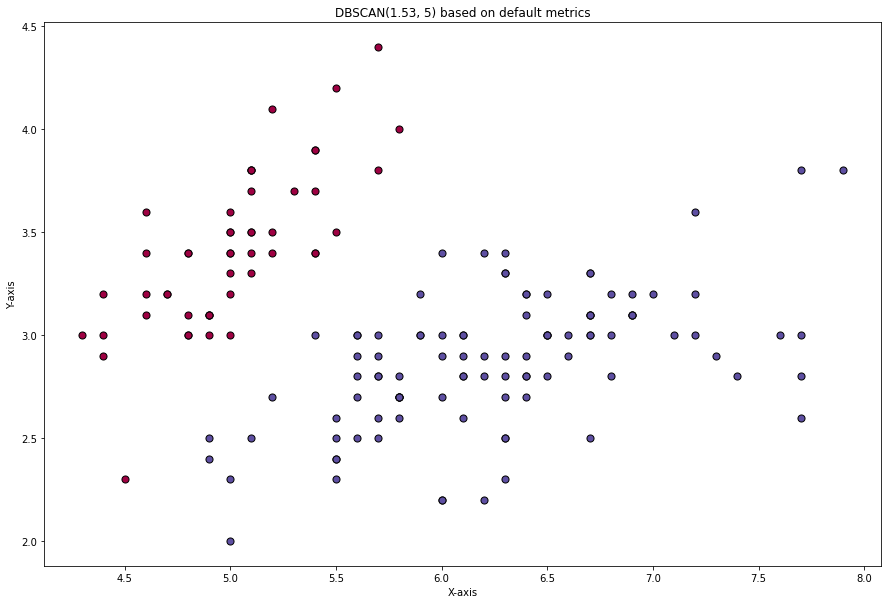
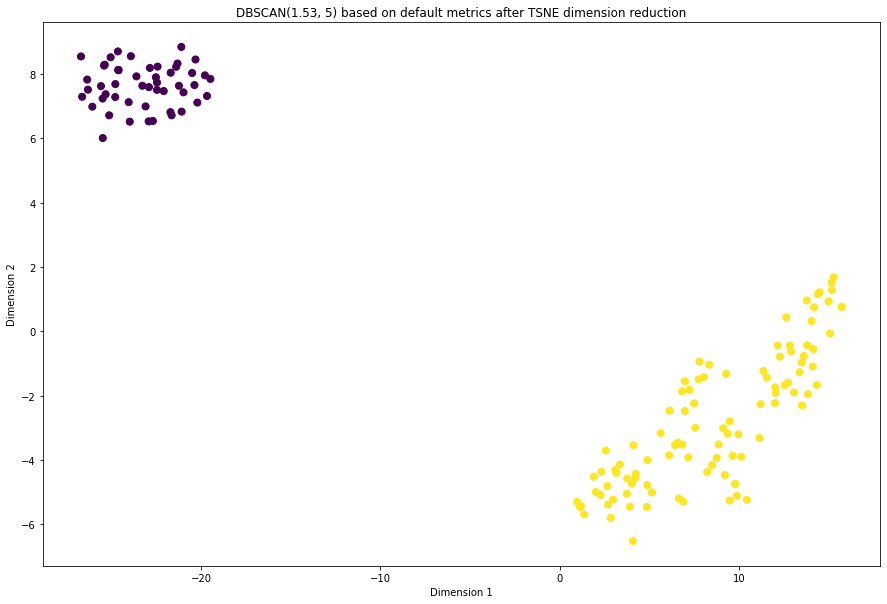


Figure 7 – The best result of DBSCAN clustering producing only 2 clusters when 3 were expected. The left is a simple projection of the results on a graph whilst the right is the data after undergoing TSNE dimension reduction.

Silhouette score = 0.69

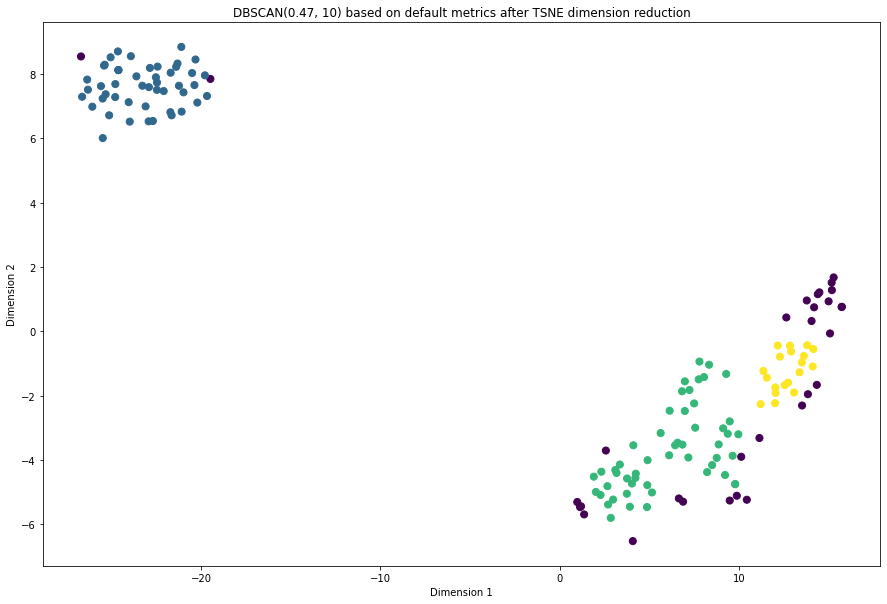
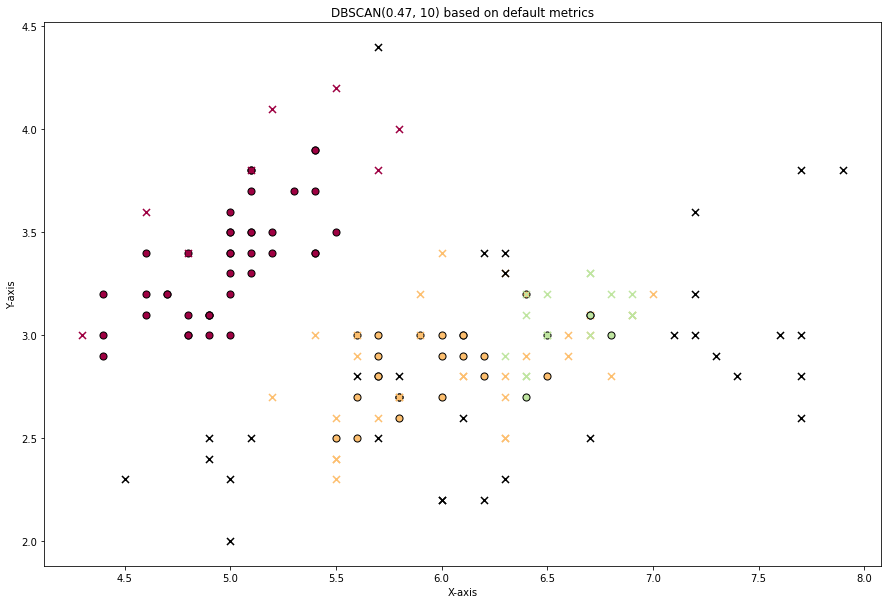


Figure 8 – The second-best attempt at clustering the data, the coloured crosses indicate edge points, and black crosses indicate outliers. Silhouette score = 0.36

Gower Distance

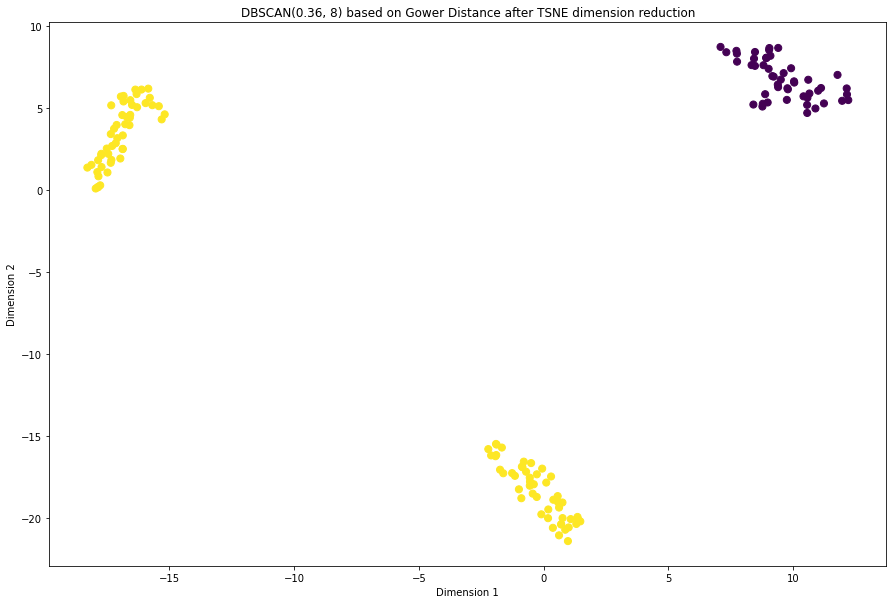
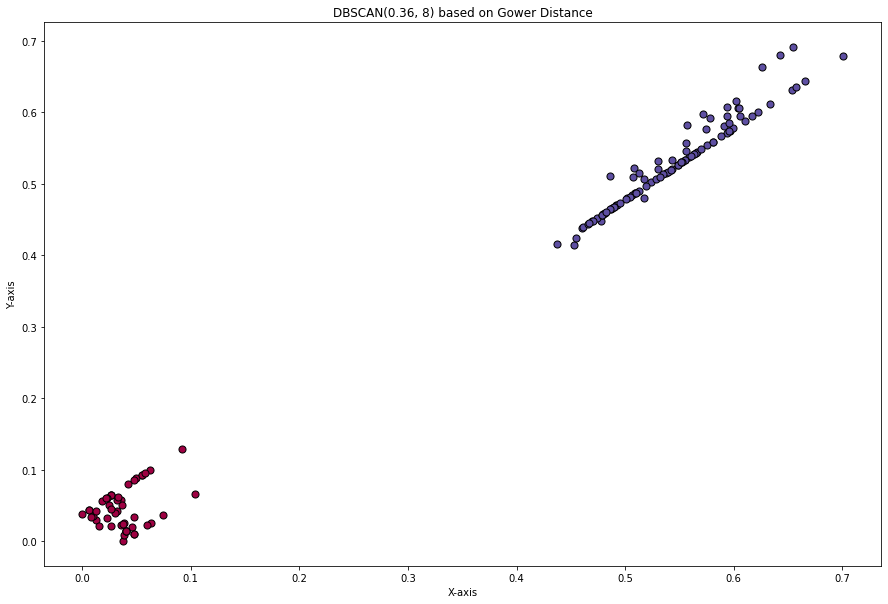


Figure 9 – The best cluster produced using Gower distance. Silhouette score = 0.67

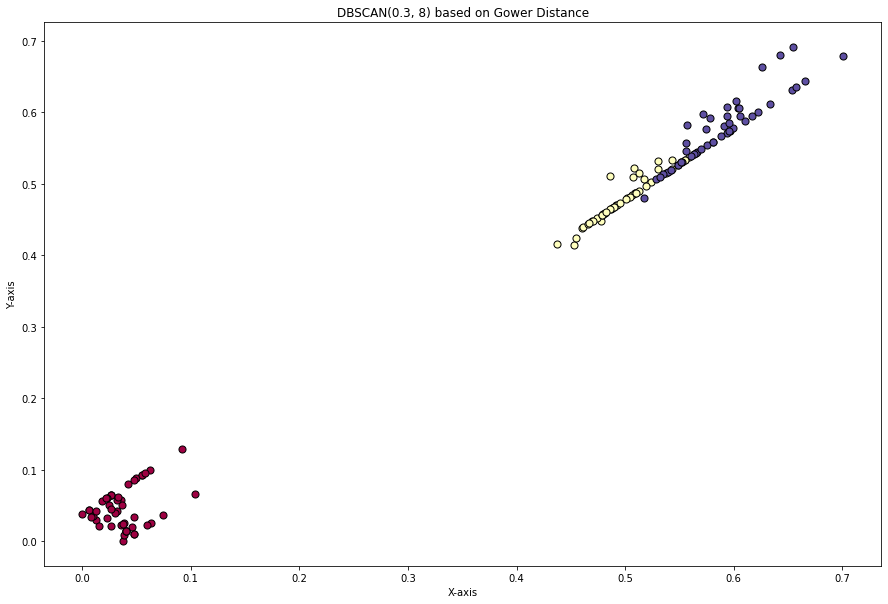


Figure 10 – The second-best cluster produced using Gower distance. Silhouette score = 0.62

**K-Means**

After trying multiple combinations, I found that the difference in initialisation type made very little difference to the silhouette score of the cluster. As such, for consistency, I have provided the best two clusters with the same initialisation type (k-means++).

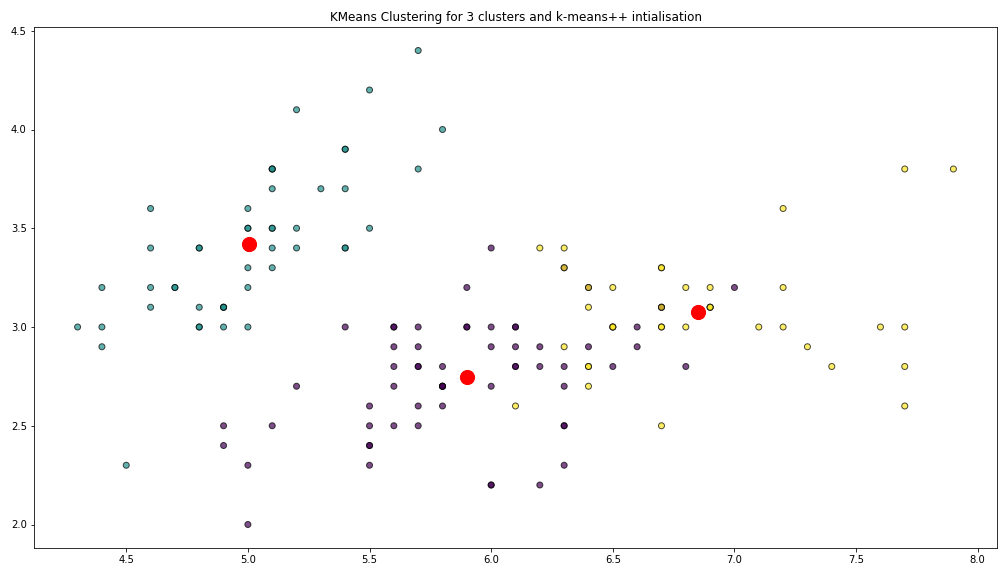
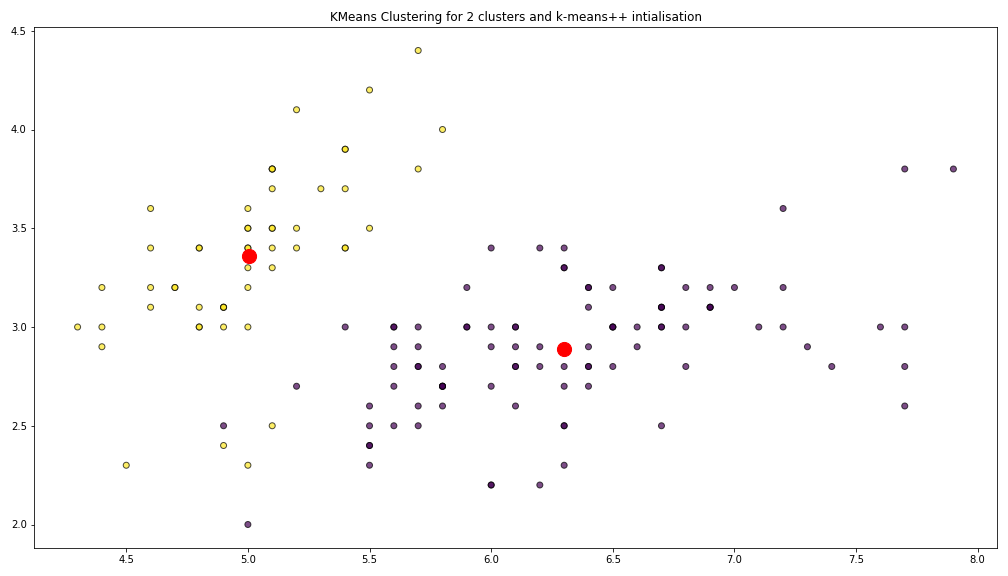


Figure 11 – On the left is the cluster with the highest silhouette score (0.68) and the right is the second highest (0.55).

**Hierarchical Agglomerative Clustering (HAC)**

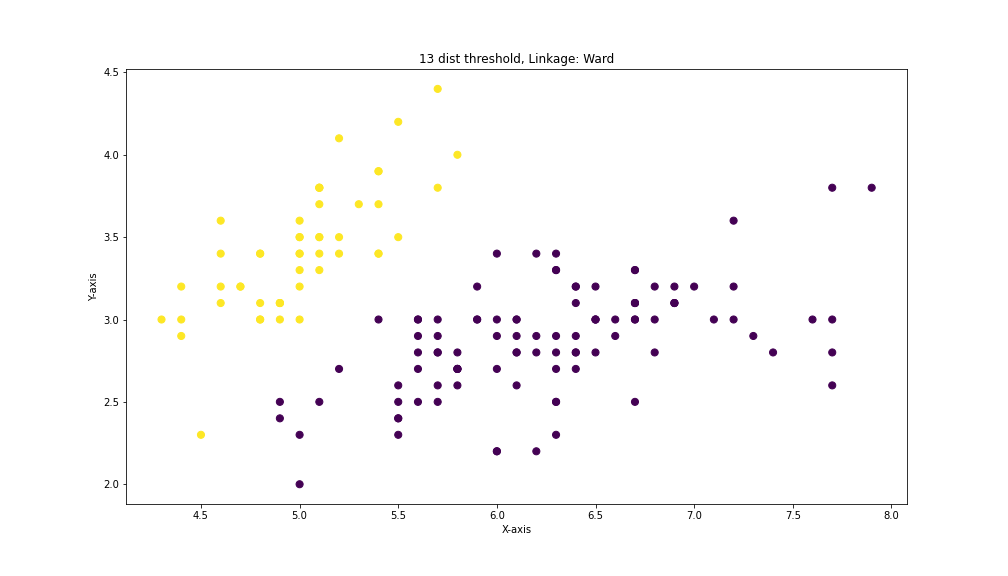


Figure 12 – The HAC cluster with the highest silhouette score (right = 0.69) and the second largest (left = 0.55).

Multiple combinations of ward linkage and different distance thresholds yielded the same silhouette score I chose these two to illustrate the common trend that the HAC algorithm tended to cluster the data into either two or three different groups.

**Conclusions**

All three of these clustering algorithms produced similar results, albeit with varying silhouette scores, I believe that out of these algorithms the best choice for clustering the Iris dataset is DBSCAN using the Gower distance. Multiple algorithms were able to successfully separate the dataset into three clusters as expected but only DBSCAN based on Gower distance was able to retain a silhouette score of over 0.6. Future work on this dataset would include performing dimensionality reduction like TSNE on the other graphs to better visualise the results of the algorithms and using different distance metrics, in particular Gower, to validate whether or not the DBSCAN was in fact intrinsically better for the data or if it was purely based on the distance metric.