CSCI4150U: Data Mining

K-Means and DBSCAN Clustering

Syed Naqvi
Student ID: 100590852

November 20, 2024

Abstract

This report analyzes the effectiveness of K-means clustering vs DBSCAN clustering in correctly identifying complex cluster shapes while using Sum of Squared Errors (SSE) as a measure of cluster compactness. As expected, traditional K-means performed decently in identifying more globular shaped clusters but struggled with non-globular shapes while DBSCAN was able to identify both globular and non-globular shapes quite well.

1 Introduction

1.1 Methodology

We perform the k-means and DBSCAN algorithms on 4 distinct datasets consisting of various shaped clusters, and then compare the effectiveness of each method. For K-means clustering, model k values range from 1 to 6 and cluster quality is measured using Sum of Squared Error (SSE) between points and the cluster centroids. Next, given that our data is 2-dimensional, we start with a *minPts* value of 4 for the DBSCAN algorithm and then use k-distance plots to estimate ideal **eps** values. The *minPts* and **eps** values are further adjusted using trial and error.

1.2 Preprocessing

To initialize our dataset for clustering, we perform feature standardization and remove the existing labels column giving the following initial plots:

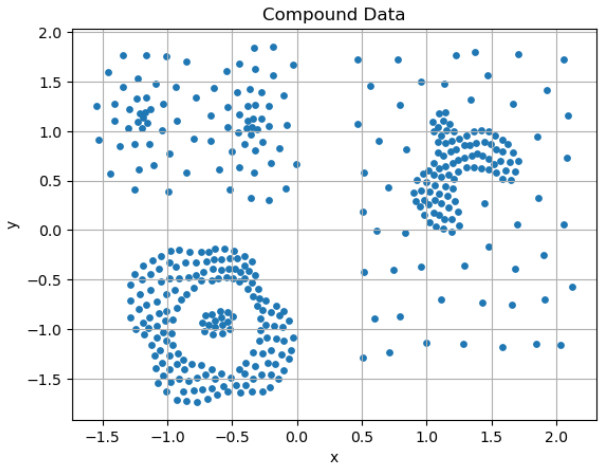

Figure 1: Compound Data

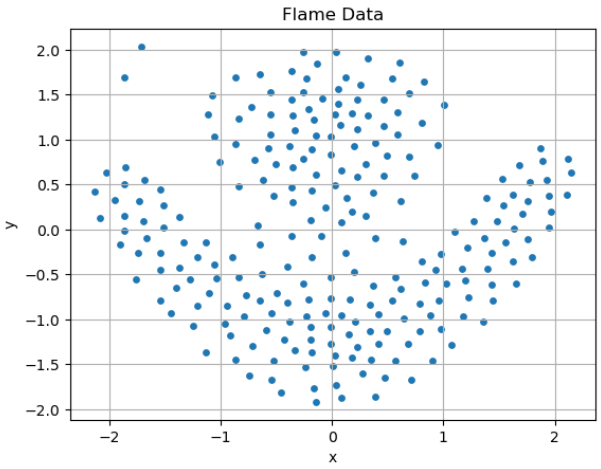

Figure 2: Flame Data

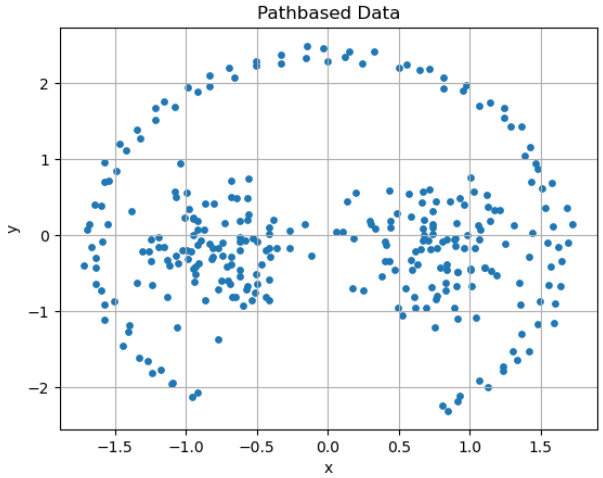

Figure 3: Pathbased Data

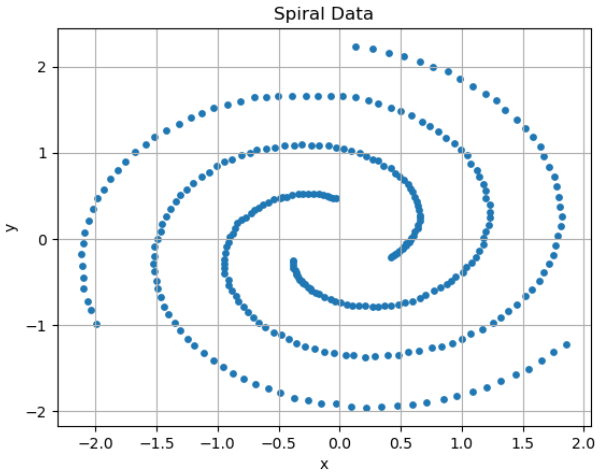

Figure 4: Spiral Data

2 Part I:

2.1 K-Means Clustering

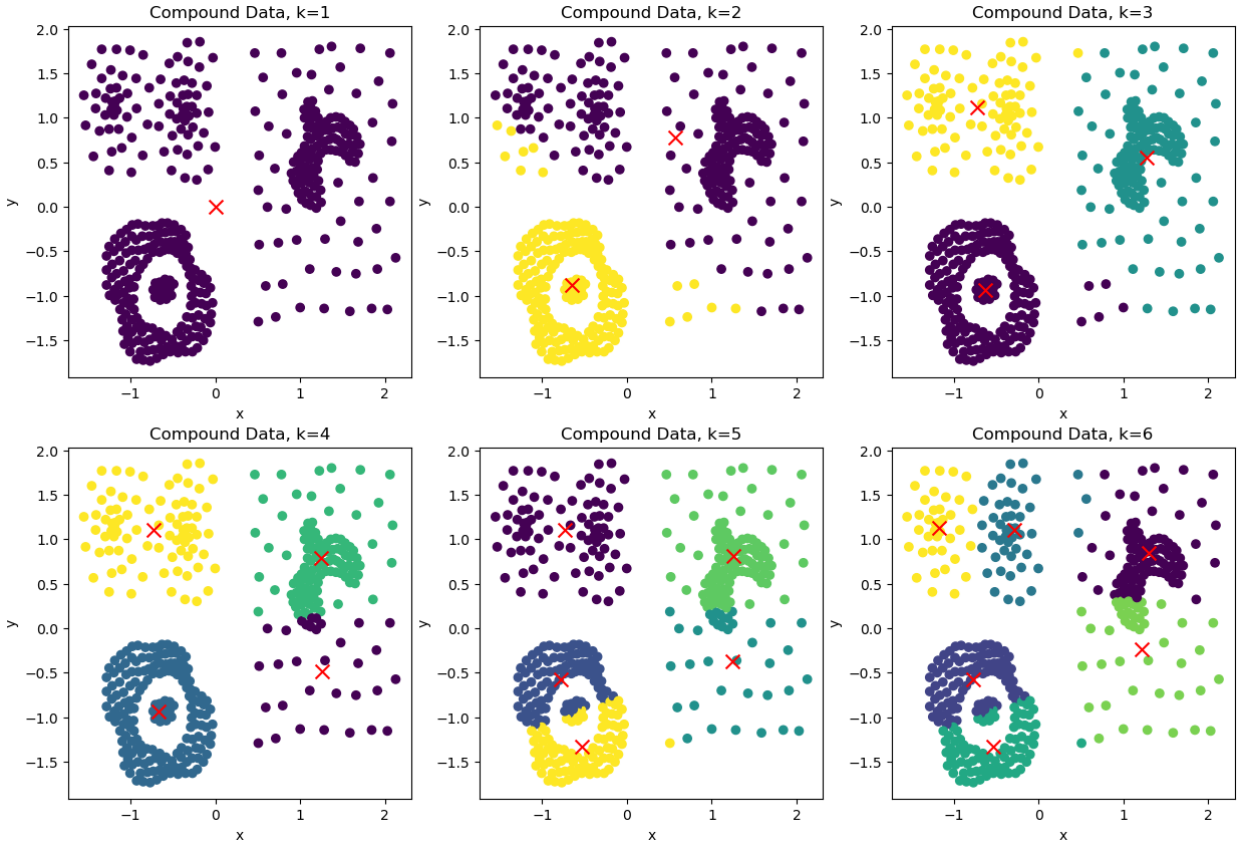

Figure 5: Compound Data Clustering

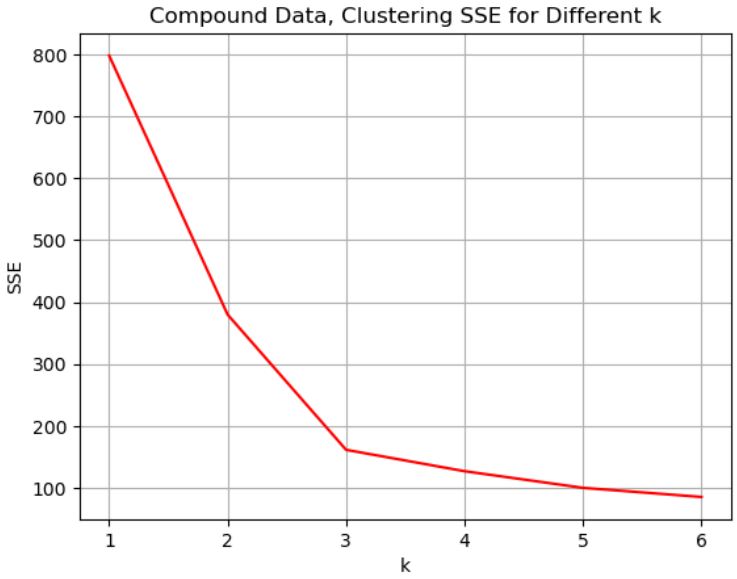

Figure 6: Compound Data SSE for Different k

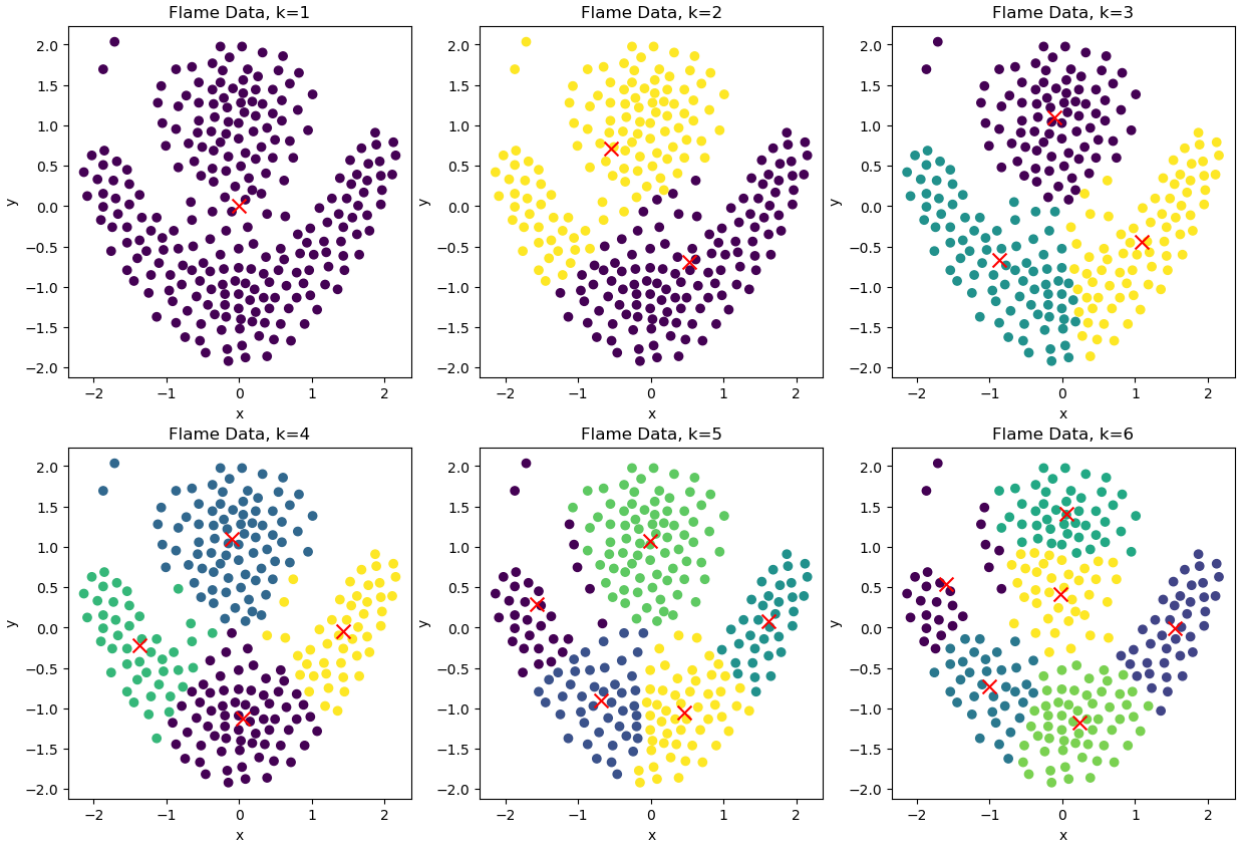

Figure 7: Flame Data Clustering

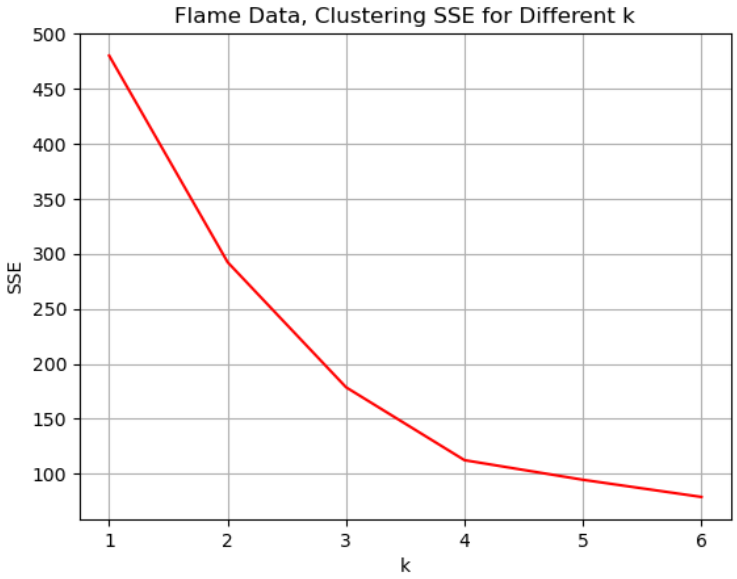

Figure 8: Flame Data SSE for Different k

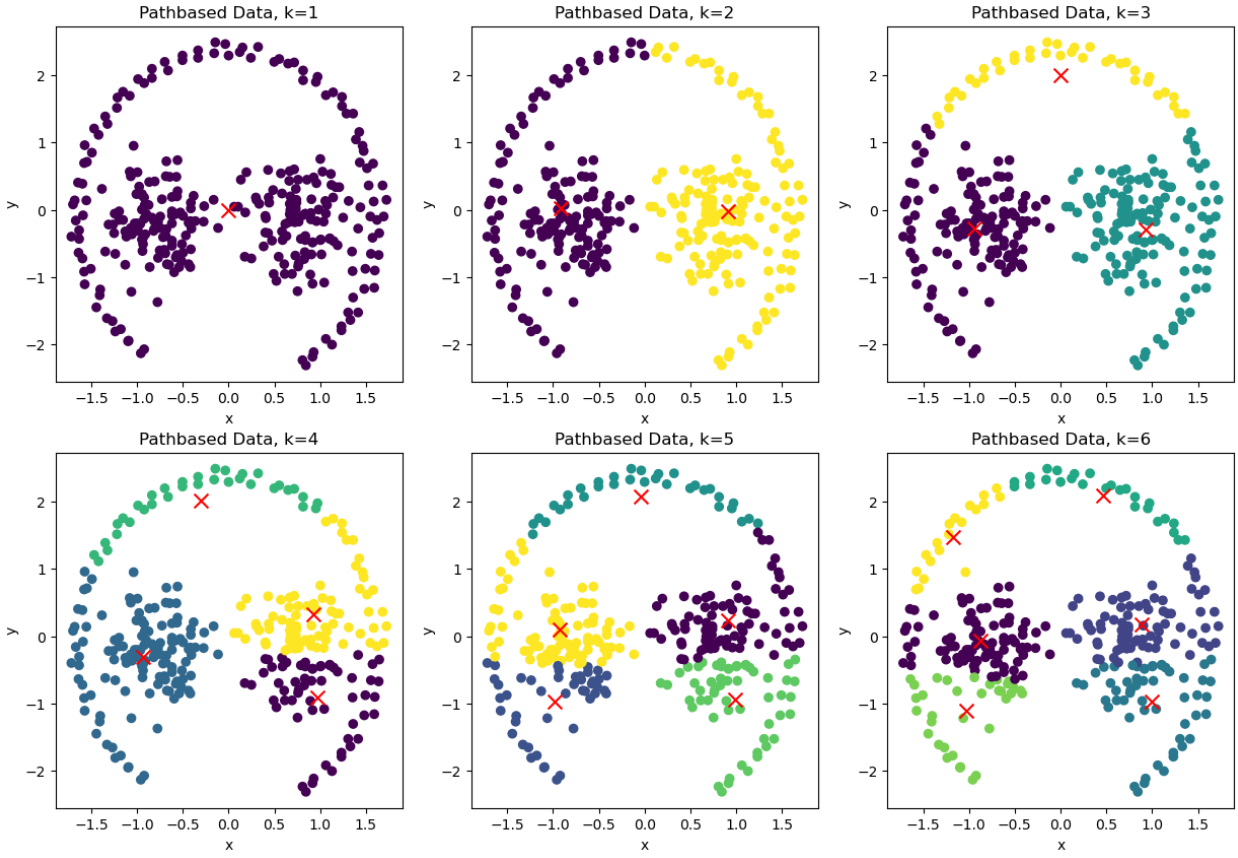

Figure 9: Pathbased Data Clustering

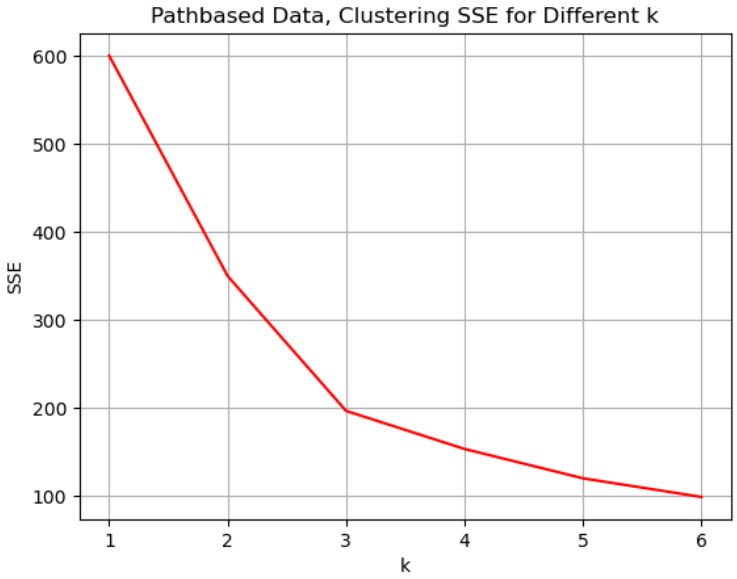

Figure 10: Pathbased Data SSE for Different k

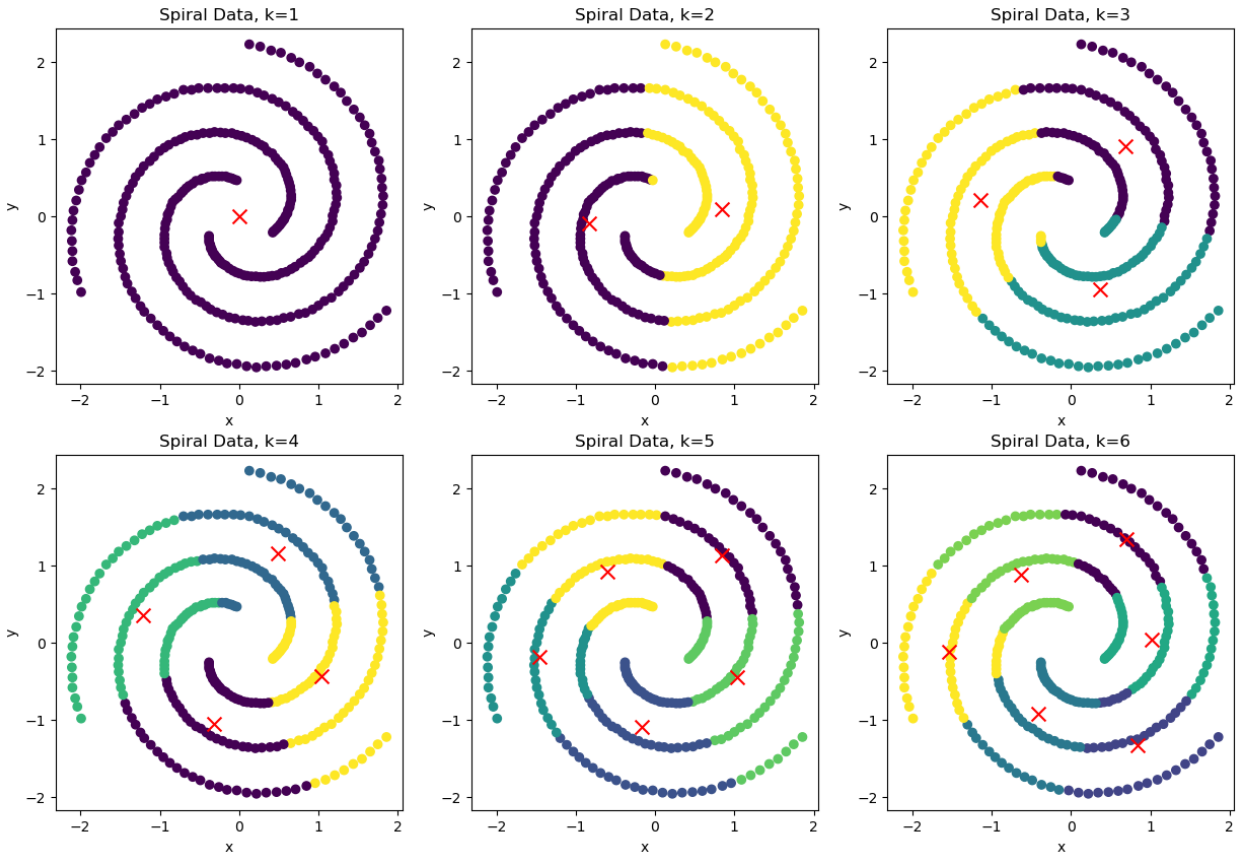

Figure 11: Compound Data Clustering

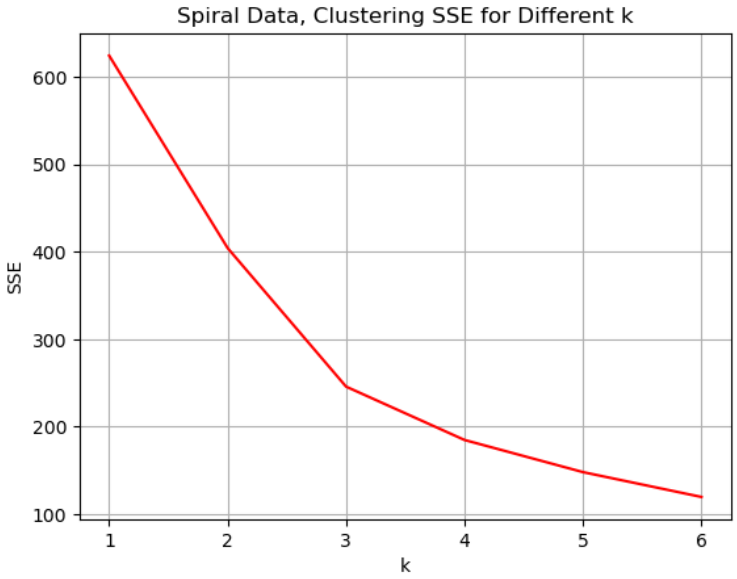

Figure 12: Spiral Data SSE for Different k

2.2 Conclusion

From the above figures it becomes clear that simply reducing the raw SSE score is not always the best way to determine the ideal number of clusters. This is because increasing the number of clusters, even if they are not accurately representing true groupings in the data will still reduce the SSE. This is most evident in the above plots where despite the SSE measures following the familiar 'elbow' curves, none of the corresponding clusters can accurately distinguish spiral arms in the Spiral dataset, the outer 'containing' cluster at the bottom left of the Compound Dataset or the border of the Pathbased dataset. Even though k-means cannot distinguish between complex cluster shapes, using centroids to approximate globular shaped clusters seems to work best when k=3 for all the datasets.

3 Part II: DBSCAN

First, we use a random value assignment for *minPts* and *eps* to get a sense of the performance of the DBSCAN algorithm:

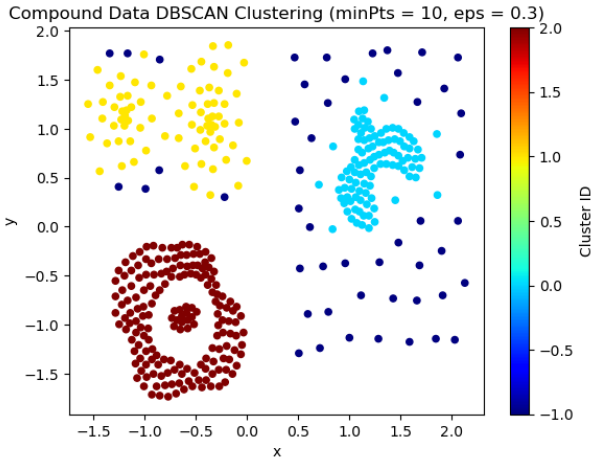

Figure 13: DBSCAN Compound Data

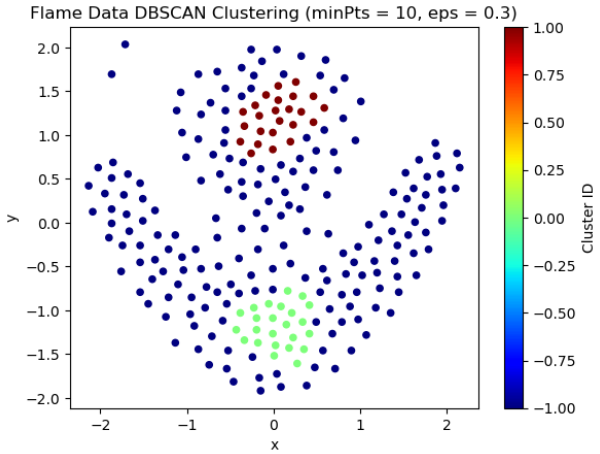

Figure 14: DBSCAN Flame Data

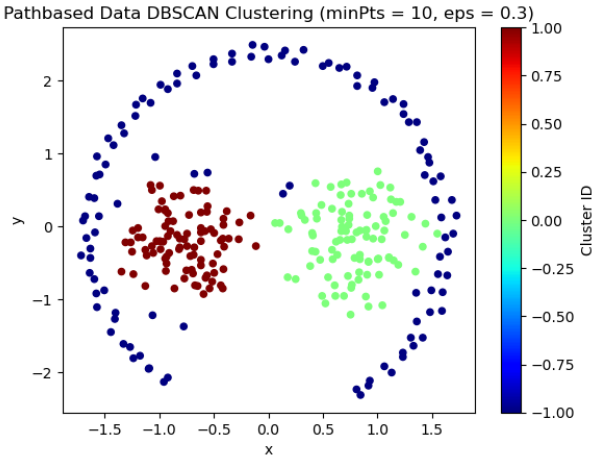

Figure 15: DBSCAN Pathbased Data

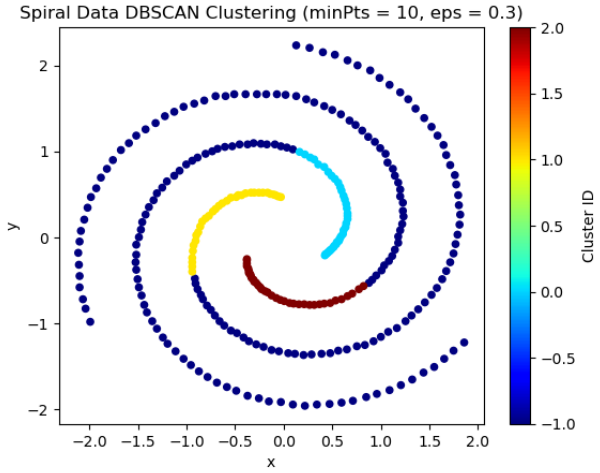

Figure 16: DBSCAN Spiral Data

Next, we can use k-distance plots and some trial and error to arrive at the ideal parameters values that result in the best clustering for all datasets:

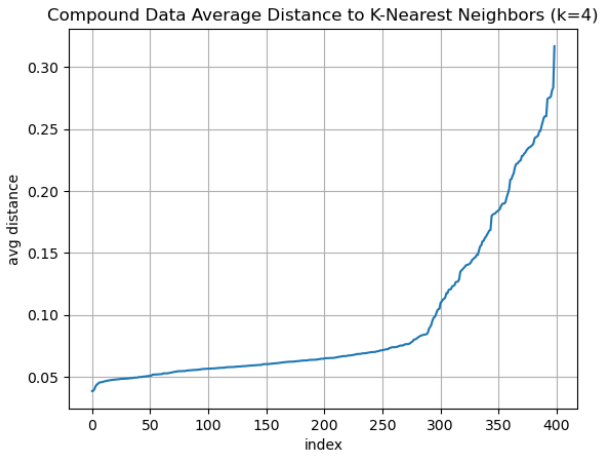

Figure 17: K-Distance Plot Compound Data

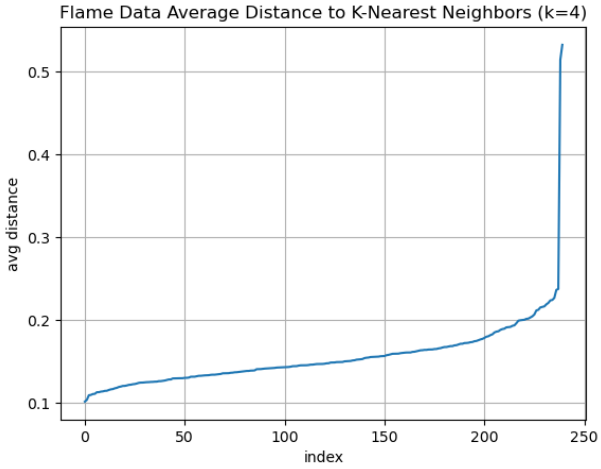

Figure 18: K-Distance Plot Flame Data

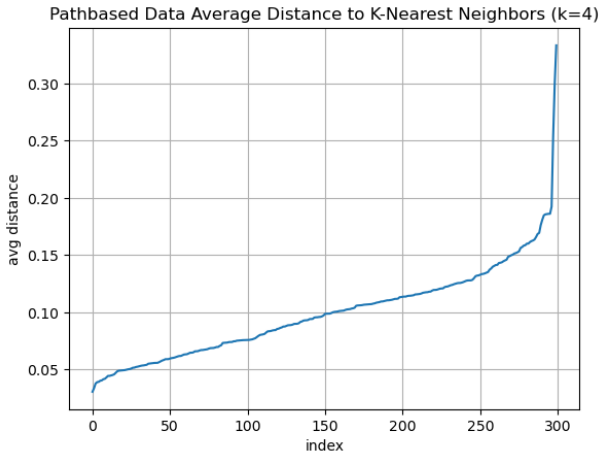

Figure 19: K-Distance Plot Pathbased Data

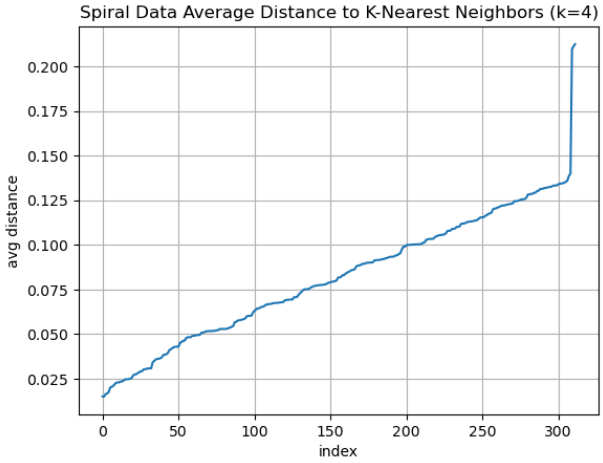

Figure 20: K-Distance Plot Spiral Data

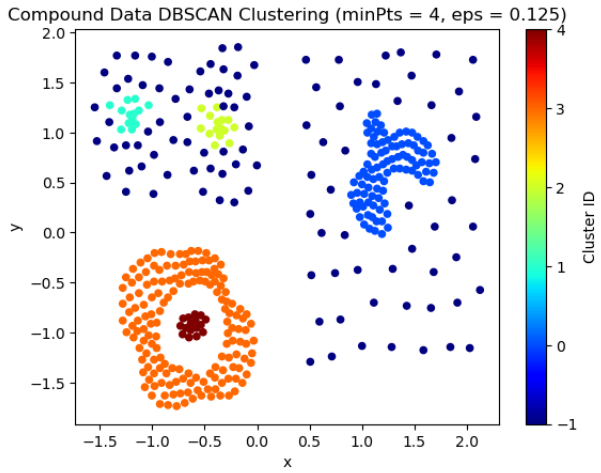

Figure 21: DBSCAN Final Clusters Compound Data

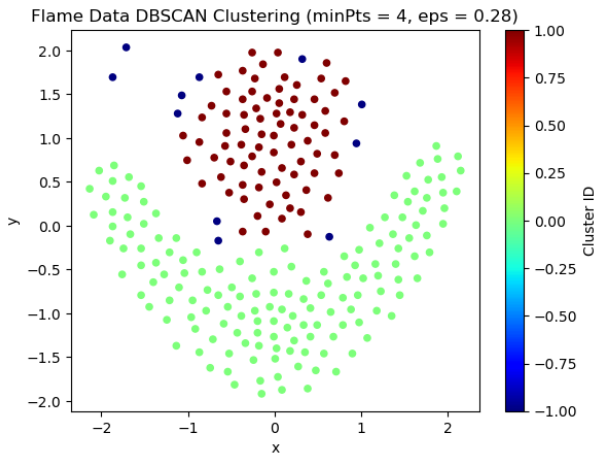

Figure 22: DBSCAN Final Clusters Flame Data

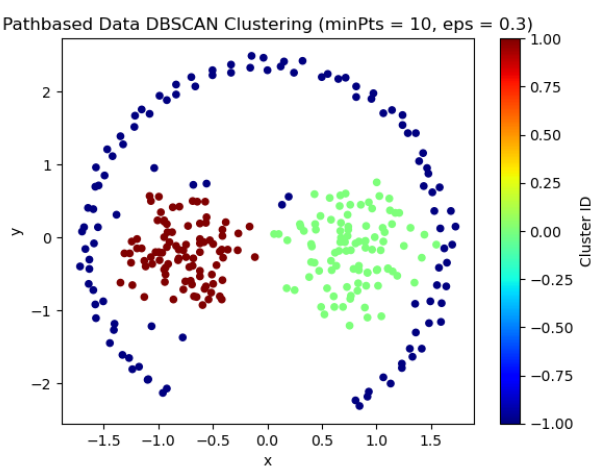

Figure 23: DBSCAN Final Clusters Pathbased Data

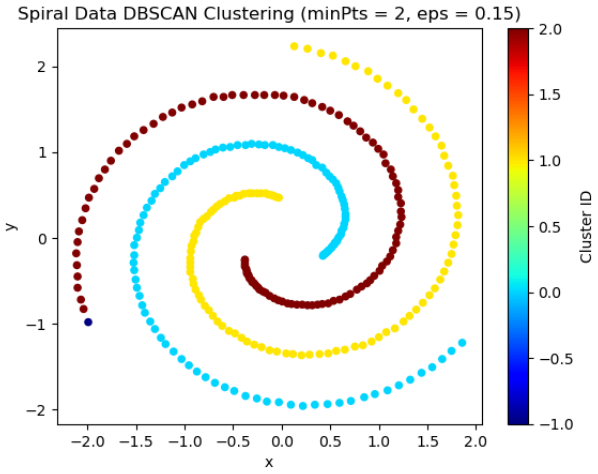

Figure 24: DBSCAN Final Clusters Spiral Data

3.1 Conclusion

We can see that DBSCAN, with some effort in parameter selection, performs very well in identifying clusters at a level most humans would find intuitive. The denisty based approach is much more flexible in finding complex cluster shapes as opposed to centroid based clustering which only works well in globular regional partitioning. Even if a particular region may contain multiple non-globular shaped clusters, k-means would try to partition the region to fit the globular mold.