Report

Mustafa Ozan Alpay

18/05/2021

1 Part 1: K-Nearest Neighbor

1.1 K-fold Cross-validation

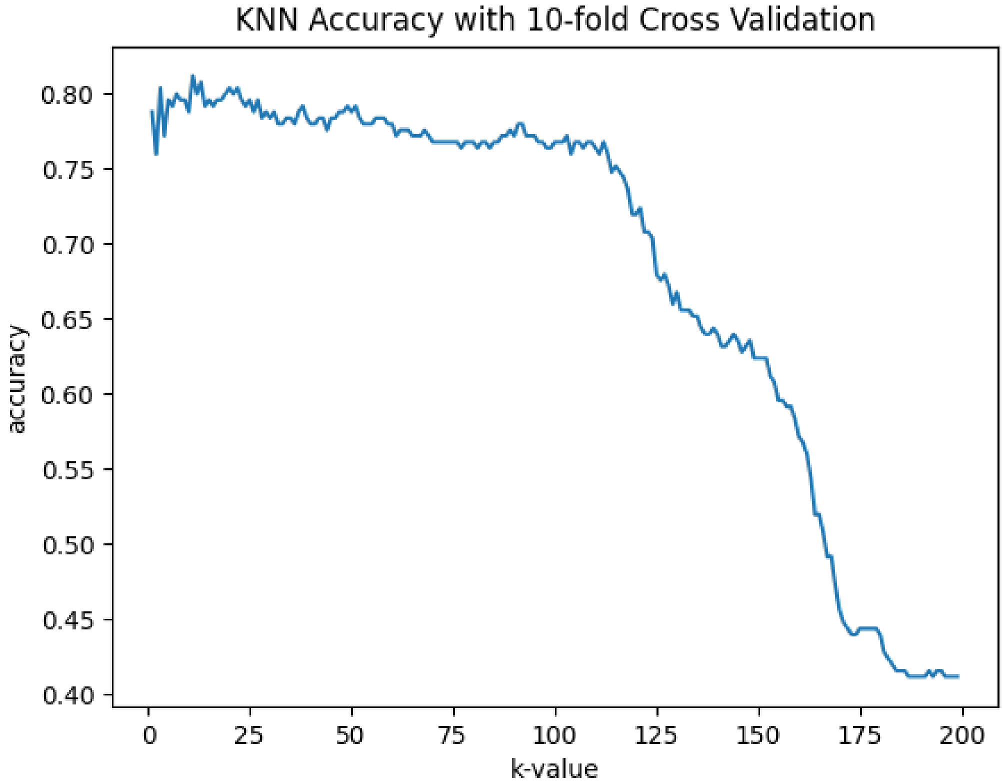


Figure 1: KNN K-fold Cross-Validation

1.2 Accuracy drops with very large k values

When we start increasing the k values, we increase the number of neighbours that we check. In the beginning, that provides a benefit, since we can detect more neighbours and get information about the surrounding data. However, as we increase the neighbour number, we eventually end up comparing too many data points with our test data, and if we continue this process we will end up with checking all data points in the model, which would not give any useful information at all. We need to find a sweet spot where we can get useful data that is enough for our model without adding extra useless neighbours.

1.3 Accuracy on test set with the best k

With using a K-value of `11`, I was able to achieve the best accuracy score, which is `0.8119999999999999`.

2 Part 2: K-means Clustering

2.1 Elbow method

2.1.1 Clustering 1

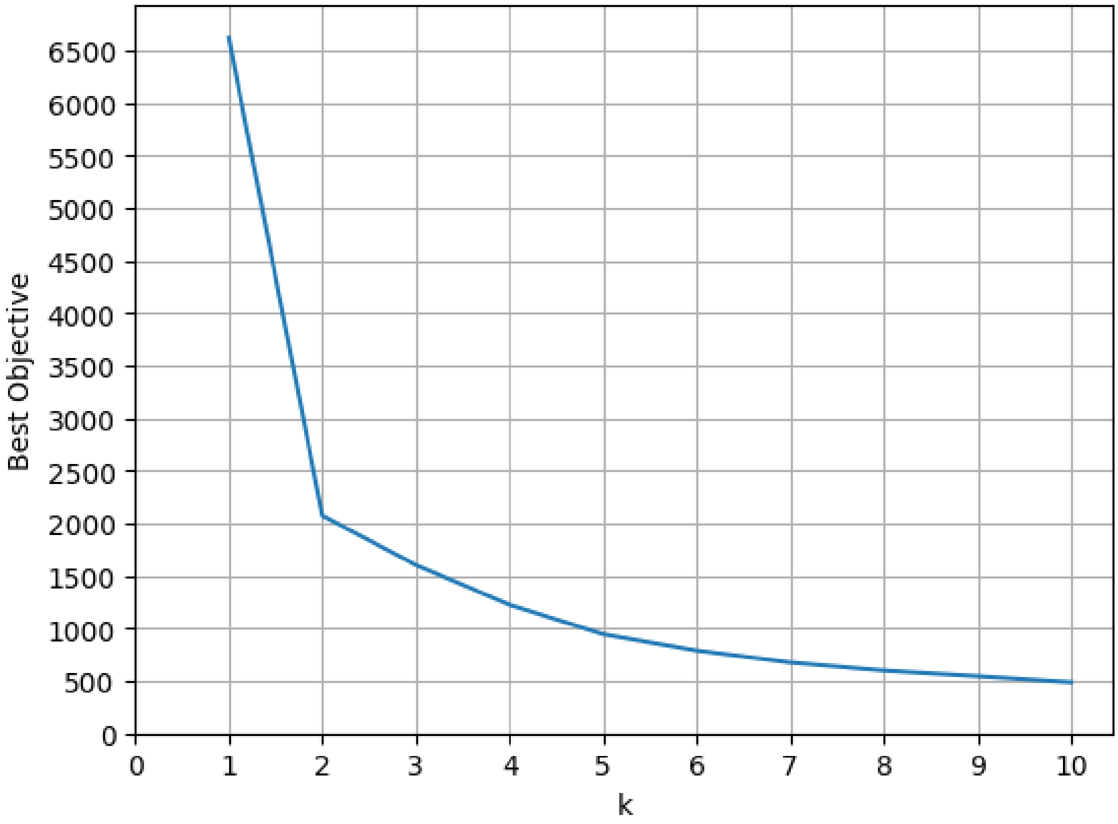


Figure 2: Elbow Plot for Clustering 1

According to the Figure 2, the suitable `k` value for this data is `2`.

2.1.2 Clustering 2

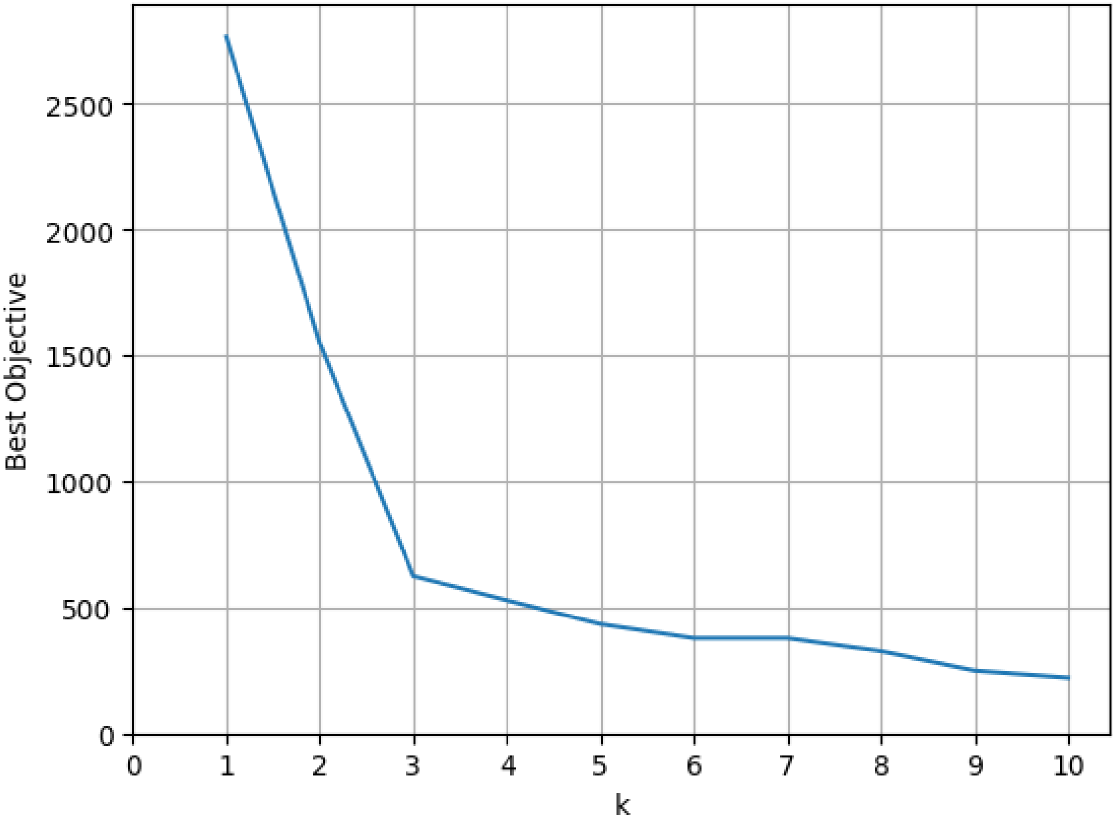


Figure 3: Elbow Plot for Clustering 2

According to the Figure 3, the suitable `k` value for this data is `3`.

2.1.3 Clustering 3

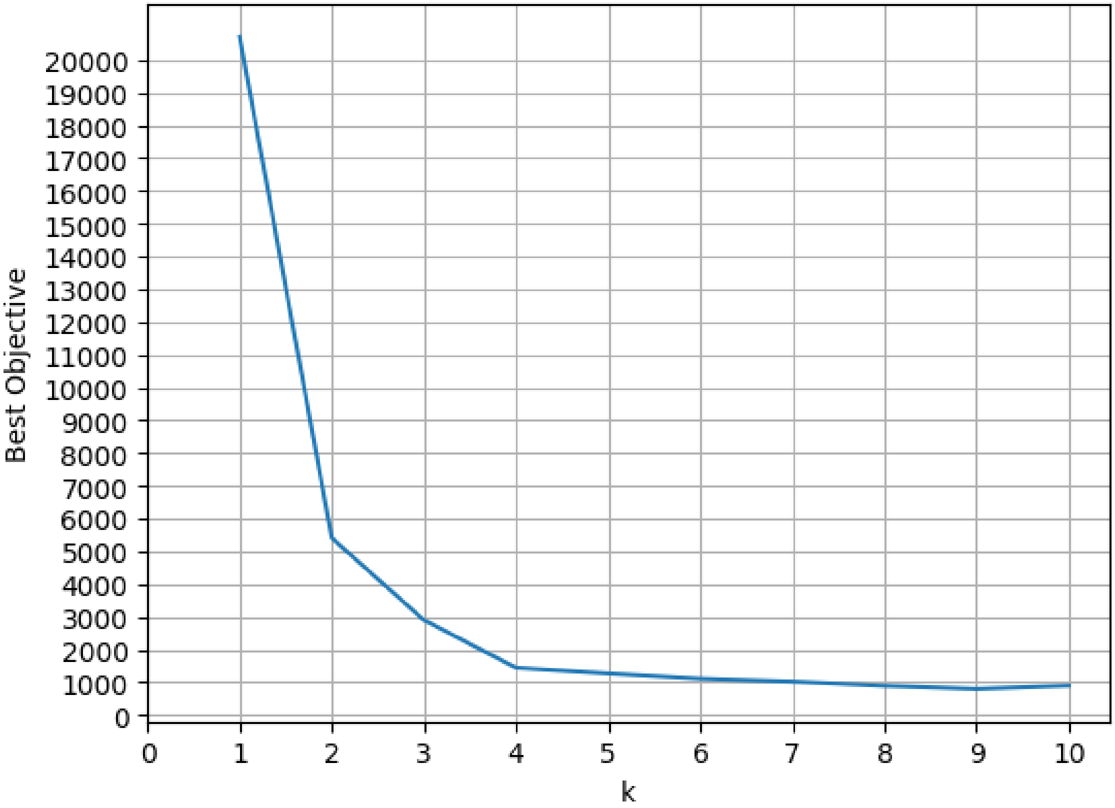


Figure 4: Elbow Plot for Clustering 3

According to the Figure 4, the suitable `k` value for this data is `4`.

2.1.4 Clustering 4

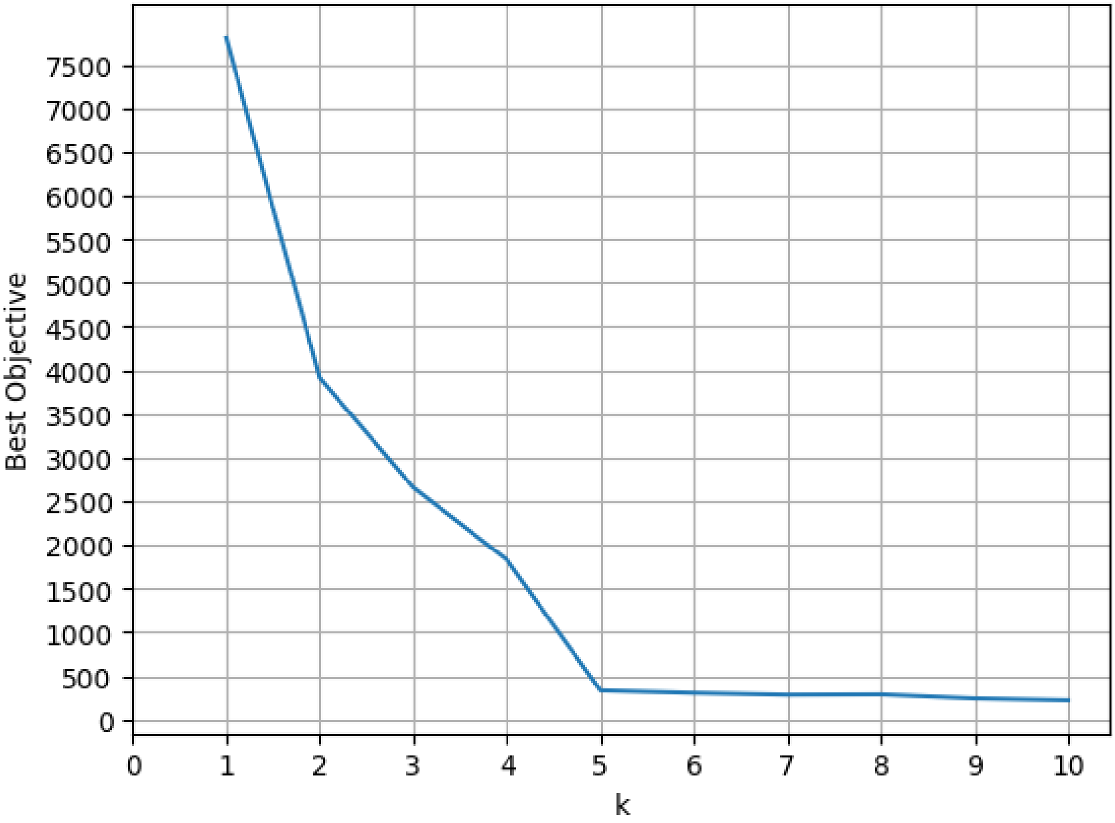


Figure 5: Elbow Plot for Clustering 4

According to the Figure 5, the suitable `k` value for this data is `5`.

2.2 Resultant Clusters

2.2.1 Clustering 1

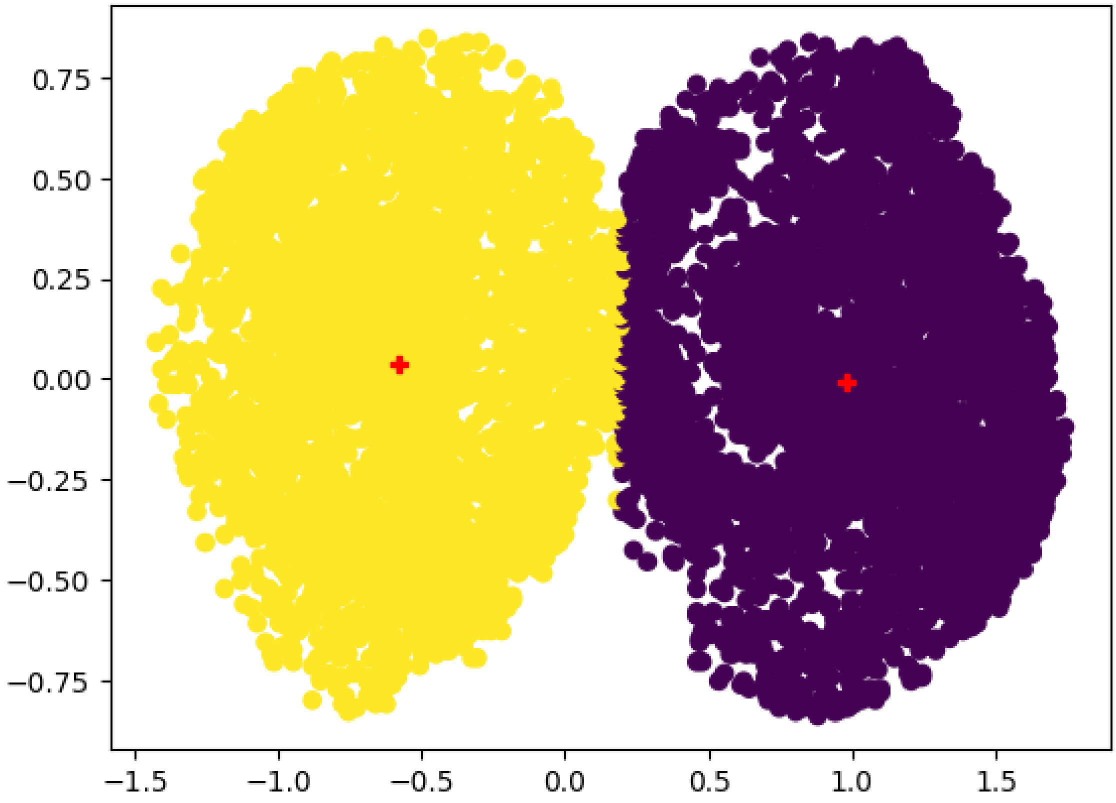


Figure 6: Final Clusters for Clustering 1

2.2.2 Clustering 2

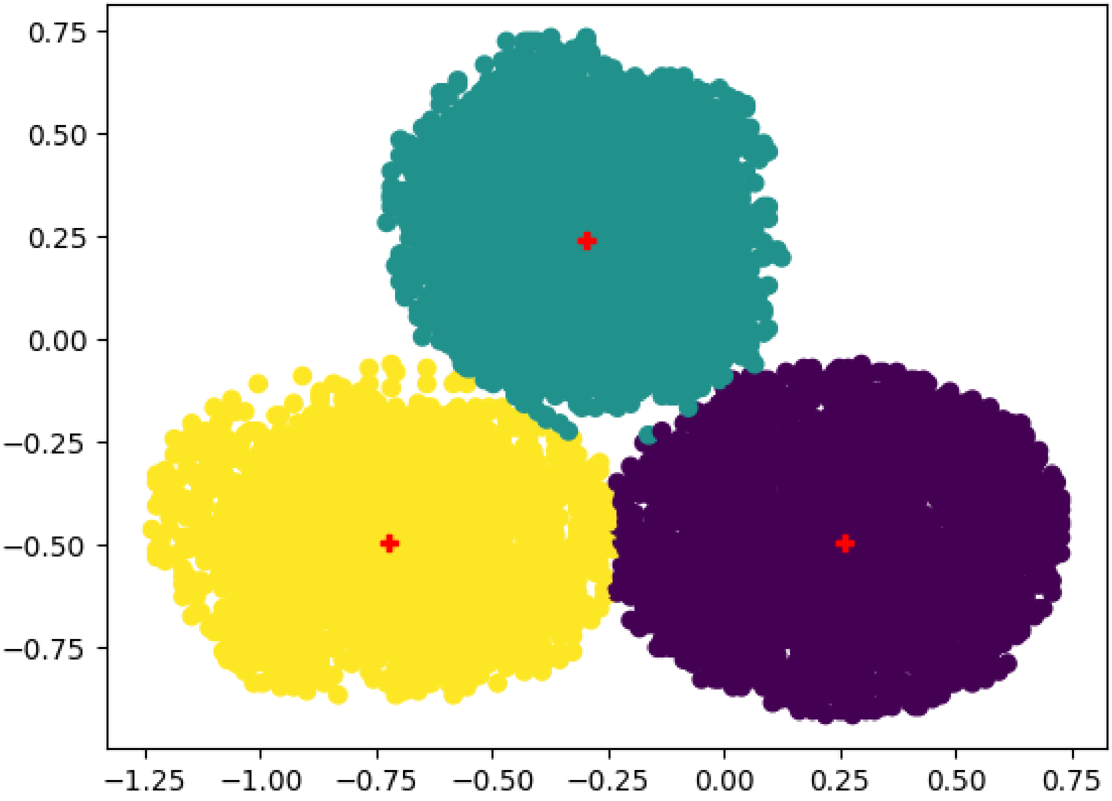


Figure 7: Final Clusters for Clustering 2

2.2.3 Clustering 3

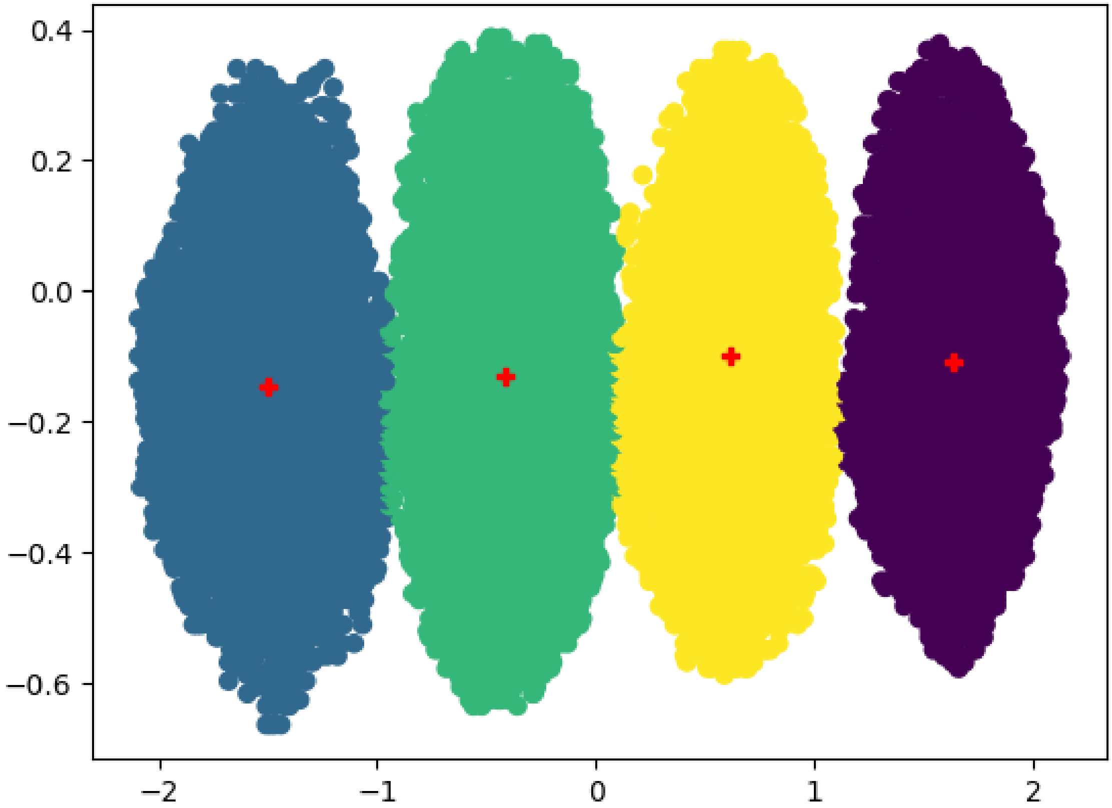


Figure 8: Final Clusters for Clustering 3

2.2.4 Clustering 4

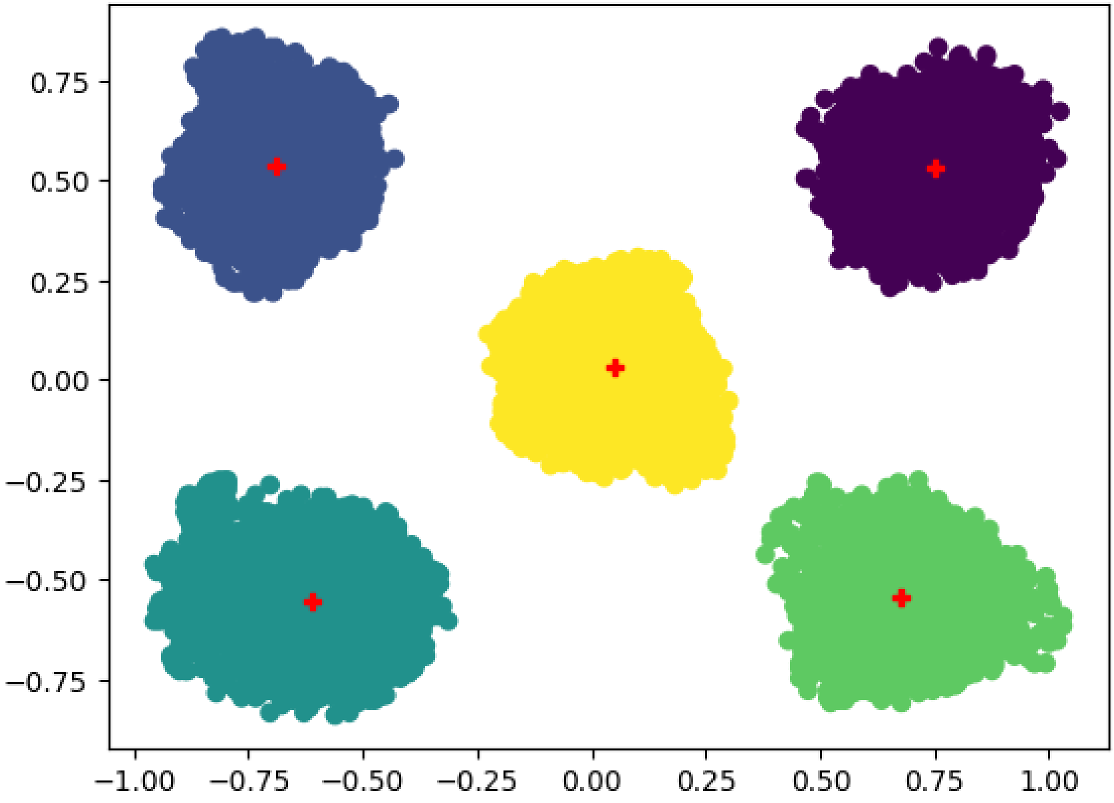
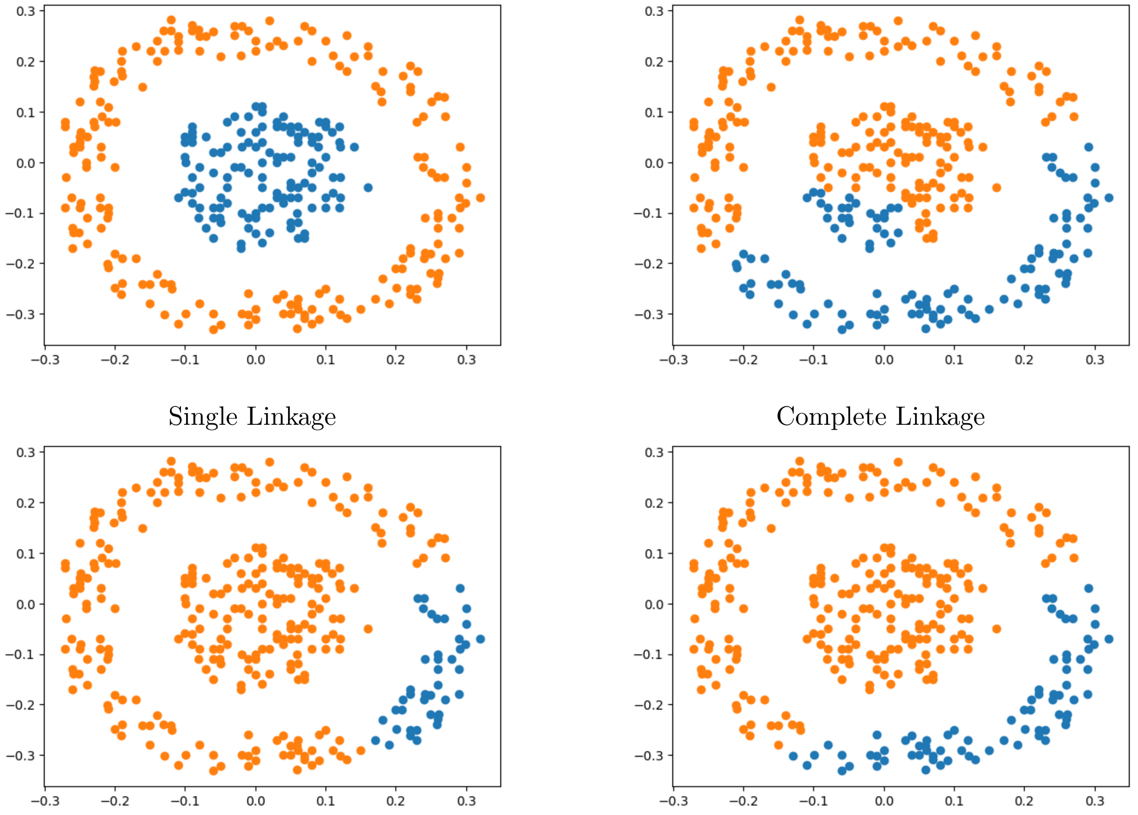


Figure 9: Final Clusters for Clustering 4

3 Part 3: Hierarchical Agglomerative Clustering

3.1 data1



Single Linkage

Complete Linkage

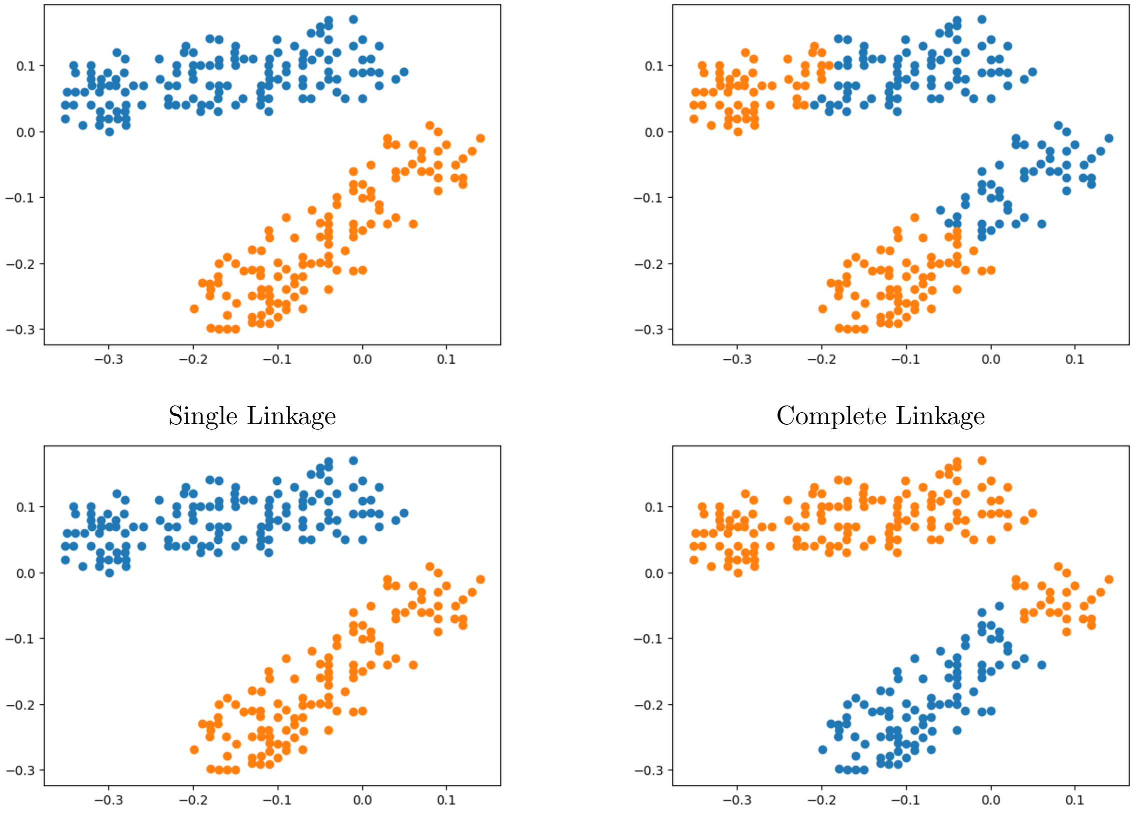
Average Linkage

Centroid Linkage

Figure 10: HAC Plots for data1

In data1, we can see that **Single Linkage** works the best. Other linkage functions appear to be functioning poorly, and there might be several reasons behind that. When we use single linkage, we cluster with minimum distance and trying to find the minimum distance while merging clusters of this dataset seems to work just fine. However with complete linkage, we try to find the maximum distance and I belive that causes the algorithm to pick points from the east and west sides of the data, hence the miscoloring. Average linkage seems to favor clustering a smaller portion of the outer disk, since it is a compromise between single linkage and complete linkage, taking the average of both plots would in fact gives us average linkage plot for this dataset. The problem with centroid linkage is probably due to not having sensible average data items for this dataset.

3.2 data2



Single Linkage

Complete Linkage

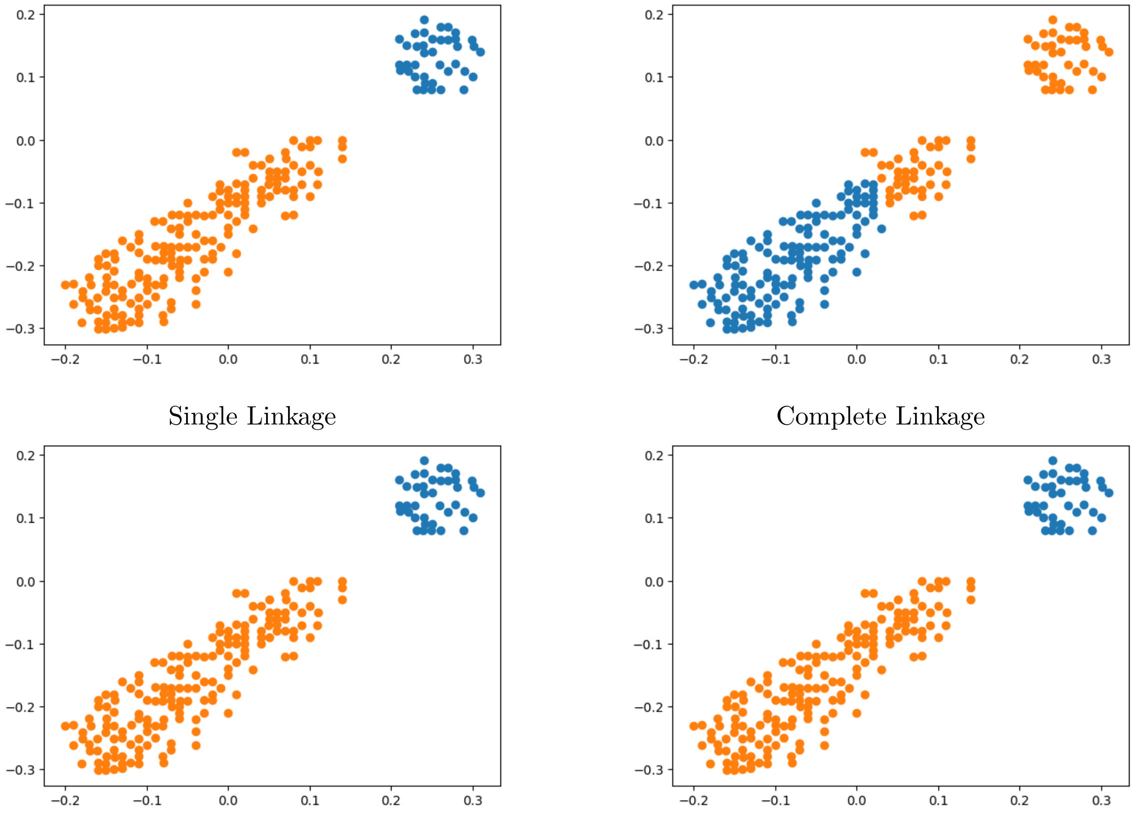
Average Linkage

Centroid Linkage

Figure 11: HAC Plots for data2

For data2, we can see that **Single Linkage** and **Average Linkage** work as expected, however Complete Linkage and Centroid Linkage seems to perform poorly. Single Linkage works as expected because it is trying to use the smallest distances, which falls within the same cluster due to the gap between two clusters. Average Linkage works because averaging the distances probably falls within the boundaries of the same clustered shape. If the distance was greater, it would probably appear more like the complete linkage plot. Complete Linkage fails because it tries to find the maximum distance, and it causes the furthest clusters to merge. Centroid Linkage fails probably due to not having sensible average data items for this dataset.

3.3 data3



Single Linkage

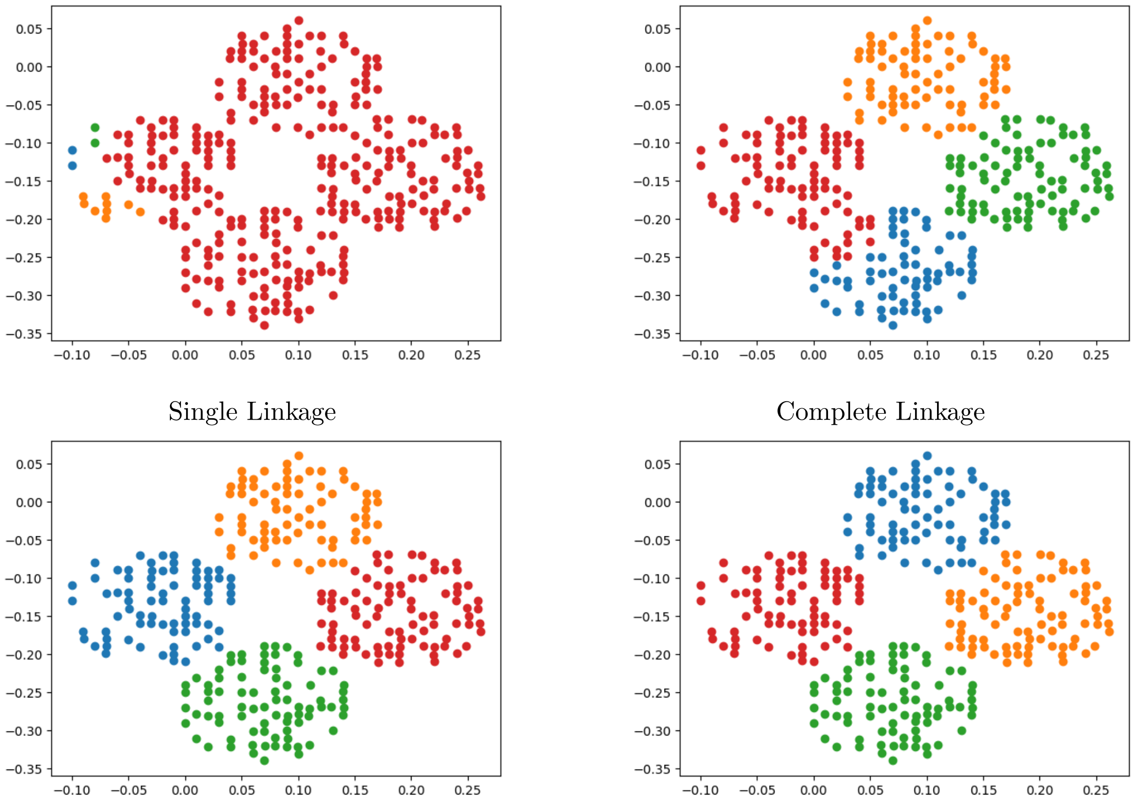
Complete Linkage

Average Linkage

Centroid Linkage

Figure 12: HAC Plots for data3

For data3, we can see that **Single Linkage**, **Average Linkage** and **Centroid Linkage** work as expected, however Complete Linkage fails. As similar to the previous datapoints, single linkage works because it tries to use the smallest distances possible, which allows merging the closest clusters, which fall within the same shape. Average Linkage works probably due to the gap between the two clusters that we see on the plot. Centroid Linkage works probably due to the huge gap between two clusters, which is caused by having sensible average of data items. On the other hand, Complete Linkage fails because it favors the maximum distance, which causes merging further clusters.



Single Linkage

Complete Linkage

Average Linkage

Centroid Linkage

Figure 13: HAC Plots for data4

3.4 data4

For data4, we can see that **Average Linkage** and **Centroid Linkage** work as expected, **Complete Linkage** works somewhat okay, but Single Linkage fails. This is due to Single Linkage favoring the smallest distances, it might cause premature merging, which can be seen from us having 2 clusters with 2 data points-each only. On the other hand, Complete Linkage behaves almost acceptable due to the well-distributed shape of the data. Having 4 data clusters that are far from each other with the same distances probably cause the maximum distances to be calculated within acceptable limits. Average linkage and centroid linkage probably provides the same output for this dataset because each cluster is a circle-shaped cluster, and their centers are also their average values.