CS 573 – Homework 5

Priyank Jain
jain206@purdue.edu

April 28, 2017

A. Exploration

1. Visualization of digits

Figures 1-10 show the ten digits, one from each class, visualized as a 28 x 28 grayscale matrix.

2. Visualization of 1000 randomly selected examples in 2D

Figure 11 shows the scatter plot of 1000 randomly selected examples in 2D.

B. Analysis of k-means

1. WCSSD and SC as a function of K

Figures 12-14 show the within-cluster sum of squared distances (WCSSD) as a function of K for the three different datasets. Figures 15-17 show the silhouette coefficient (SC) as a function of K for the three different datasets.

2.

For dataset with 10 classes, since WCSSD keeps decreasing with increasing value of K and SC achieves it maximum at K=8 and then drops off. On either side of K=8, SC has a lower value.

For dataset with 4 classes, WCSSD keeps decreasing with increasing value of K and SC achieves it maximum at K=4 and then drops off. Judging by the slope of the plot around K=4, we see that SC would decrease on either side of 4, so I select K=4 to be the best K.

For dataset with 2 classes, WCSSD keeps decreasing with increasing value of K and SC too keeps decreasing with increasing value of K. Since the maximum occurs at K=2 for SC, I select K=2 as the best value of K.

I am basing my decisions on SC only, since WCSSD would always decreases with increase in K as the number of points in a cluster drops as K increases. A higher value of SC indicates that all examples on average are well matched to their own clusters.

How the results compare: For all the three datasets WCSSD keeps decreasing with increase in value of K. For dataset with 2 classes SC keeps decreasing with increase in K. For dataset with

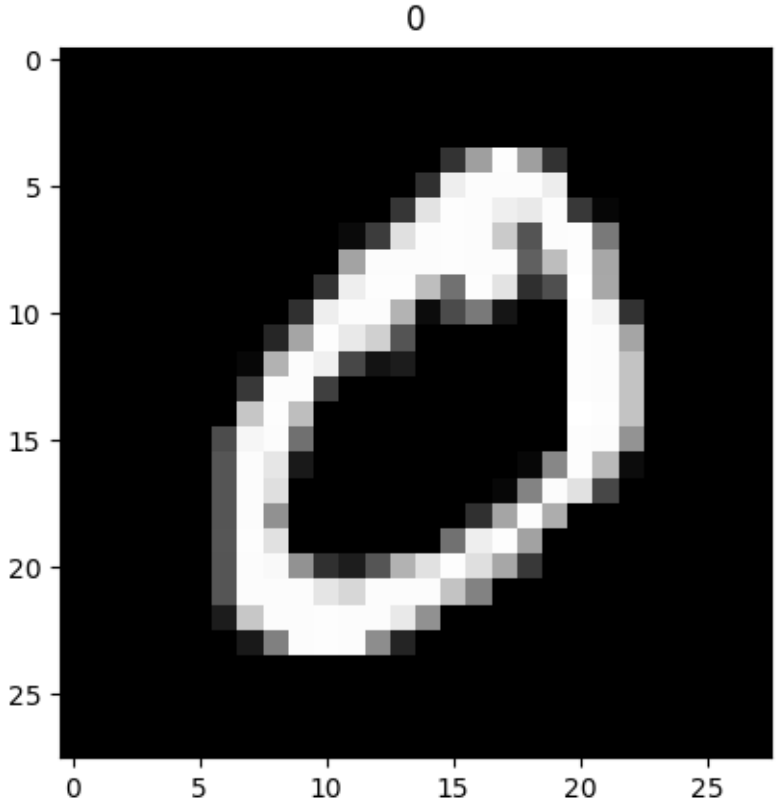

Figure 1: Example of class 0

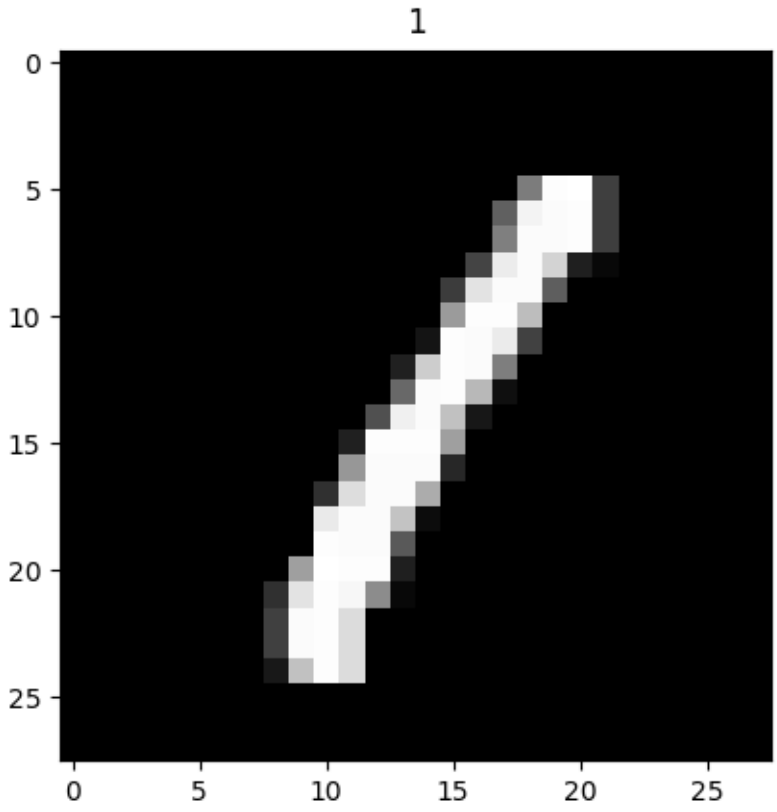

Figure 2: Example of class 1

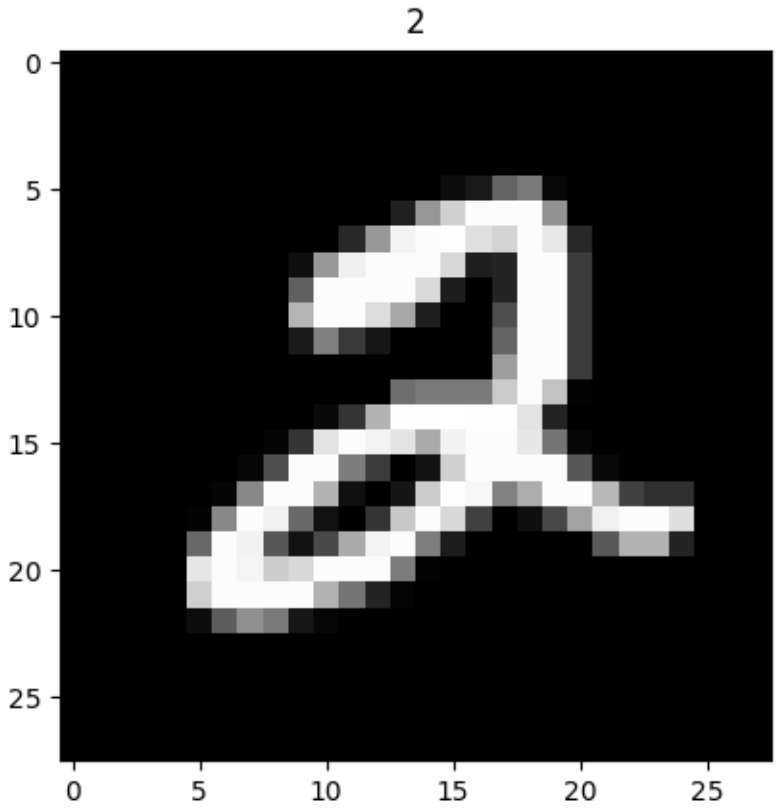

Figure 3: Example of class 2

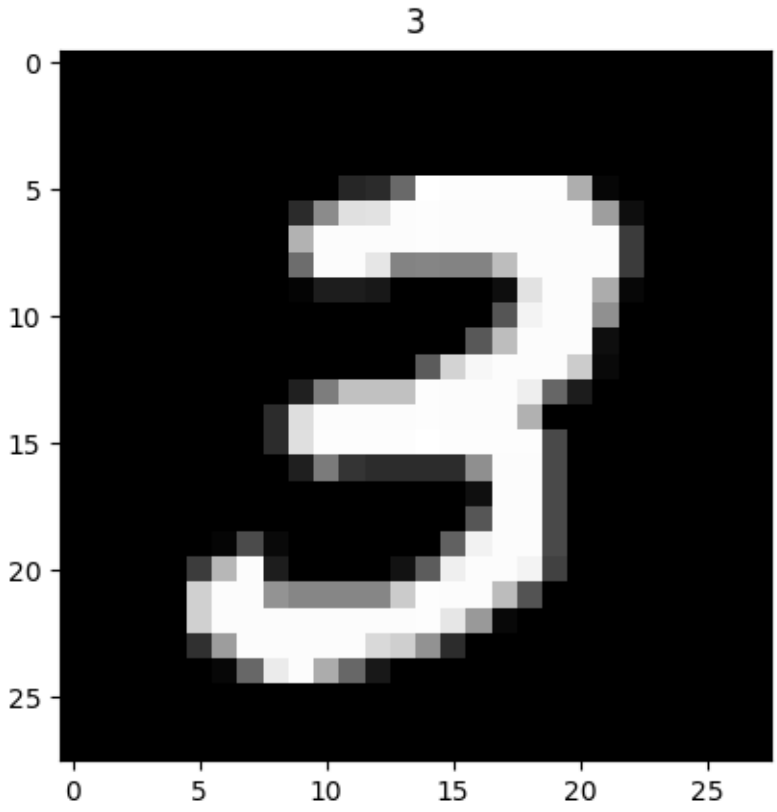

Figure 4: Example of class 3

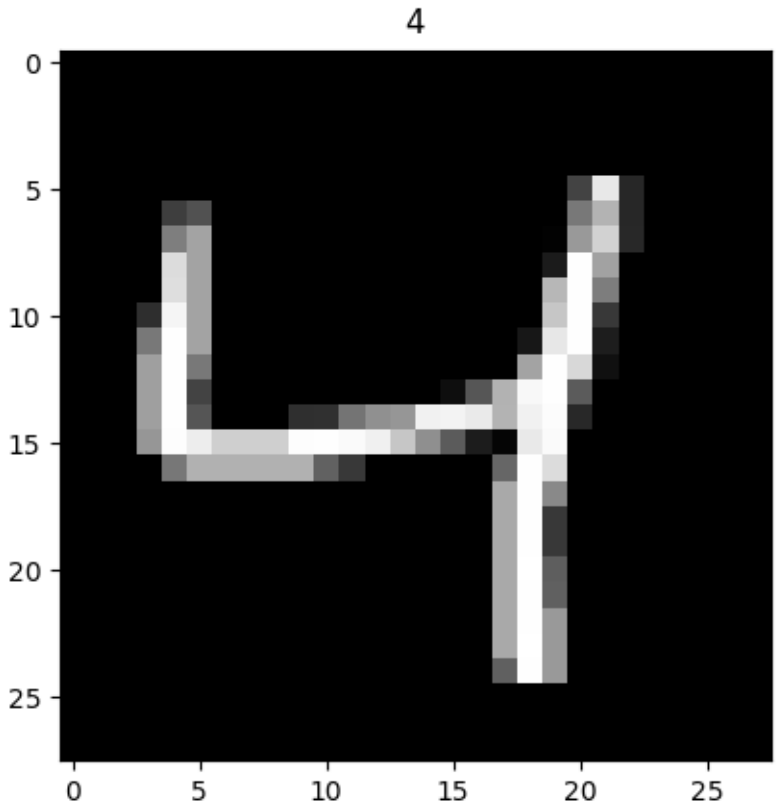

Figure 5: Example of class 4

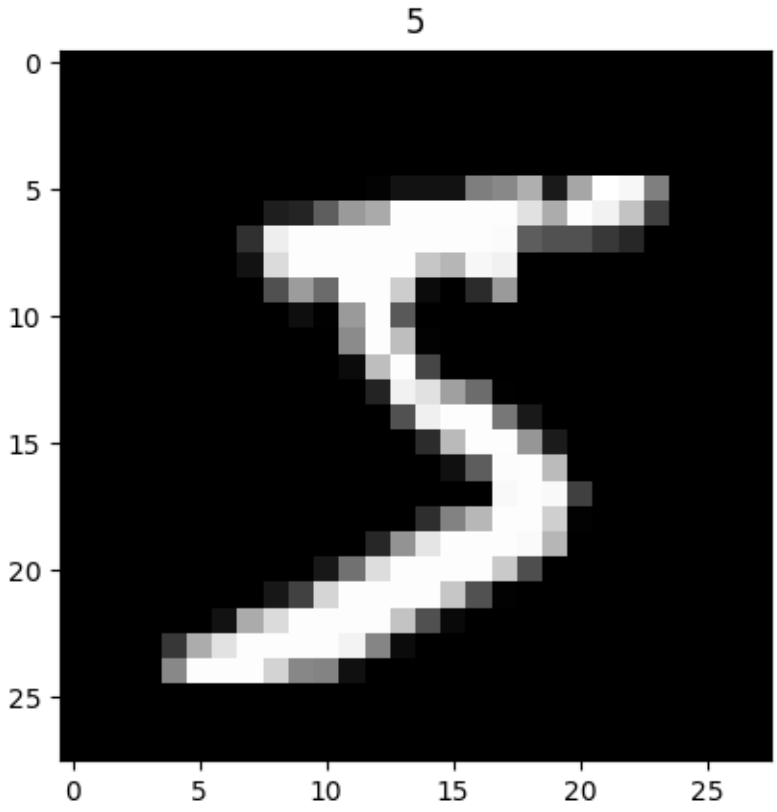

Figure 6: Example of class 5

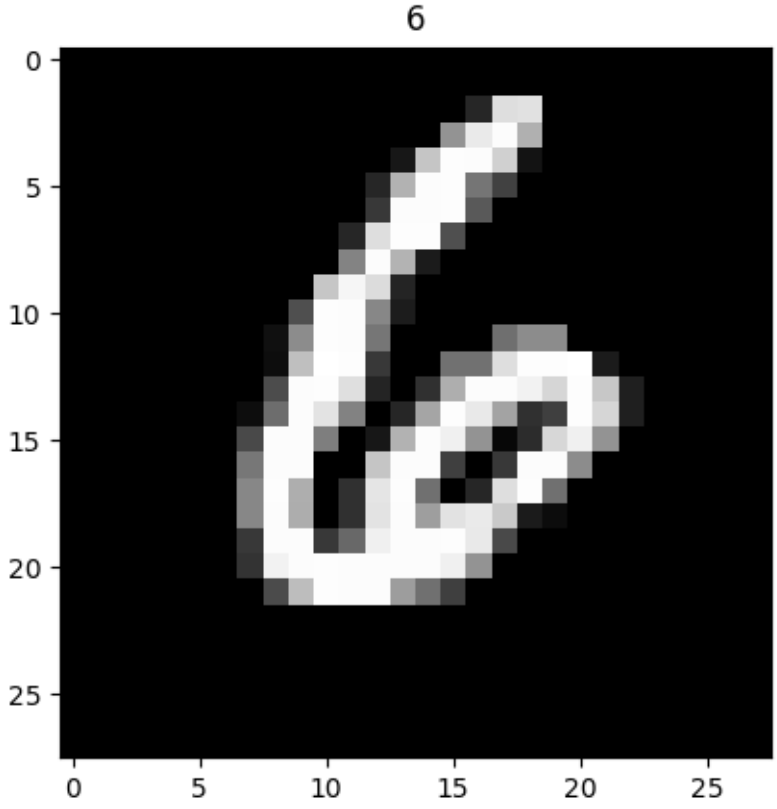

Figure 7: Example of class 6

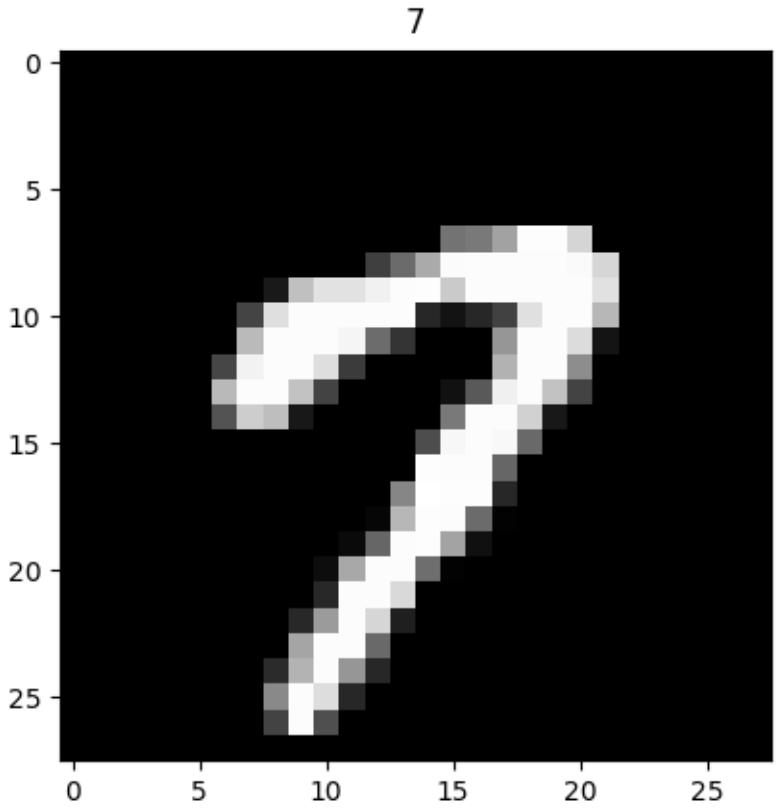

Figure 8: Example of class 7

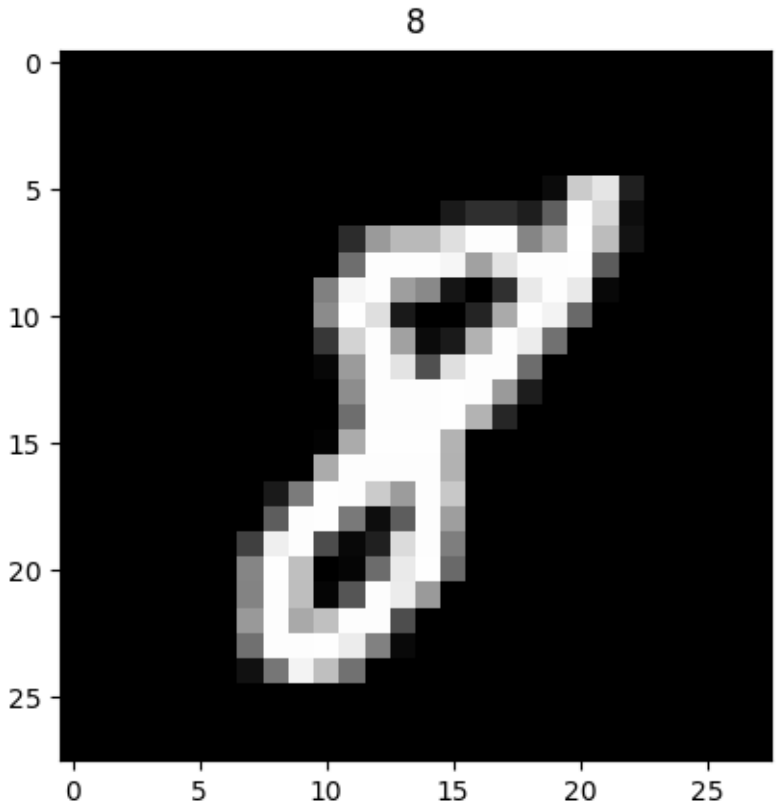

Figure 9: Example of class 8

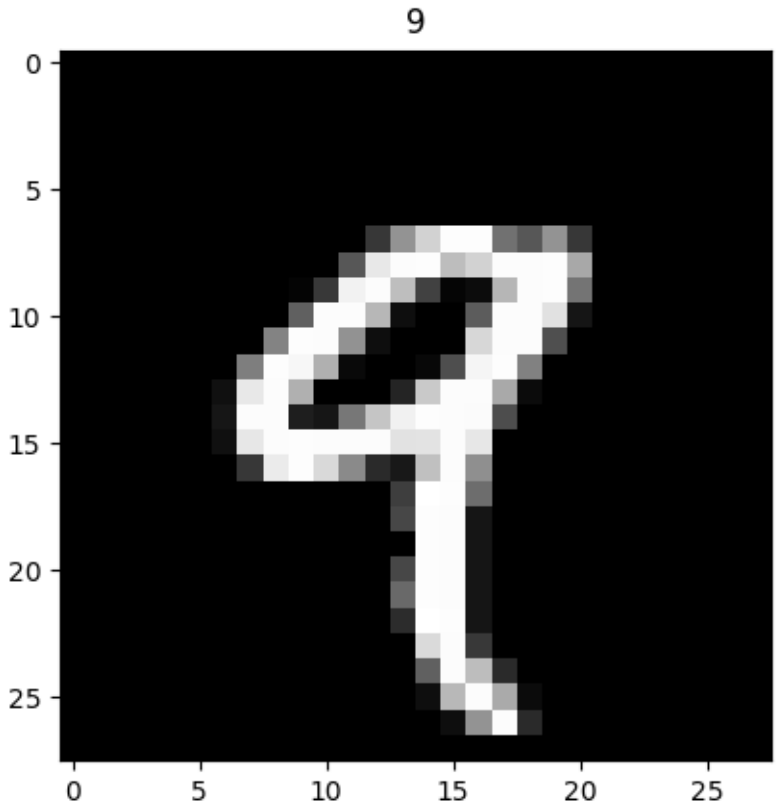

Figure 10: Example of class 9

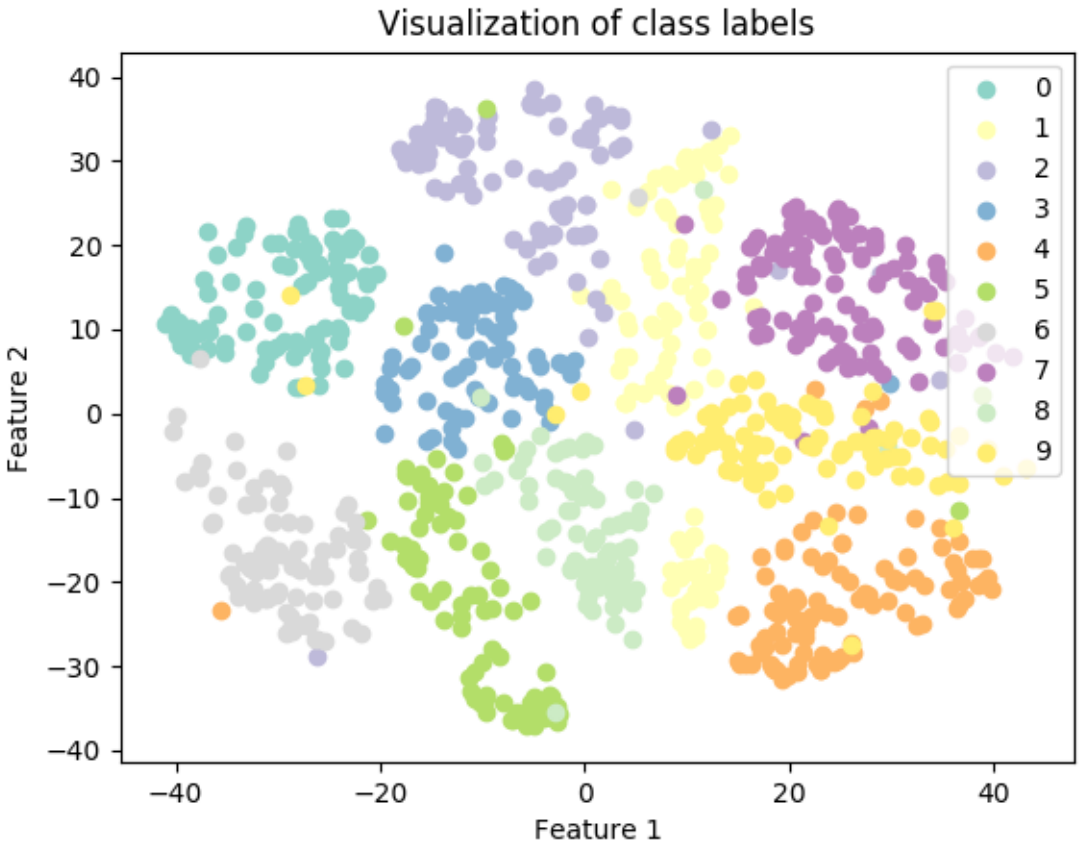

Figure 11: Visualization of class labels

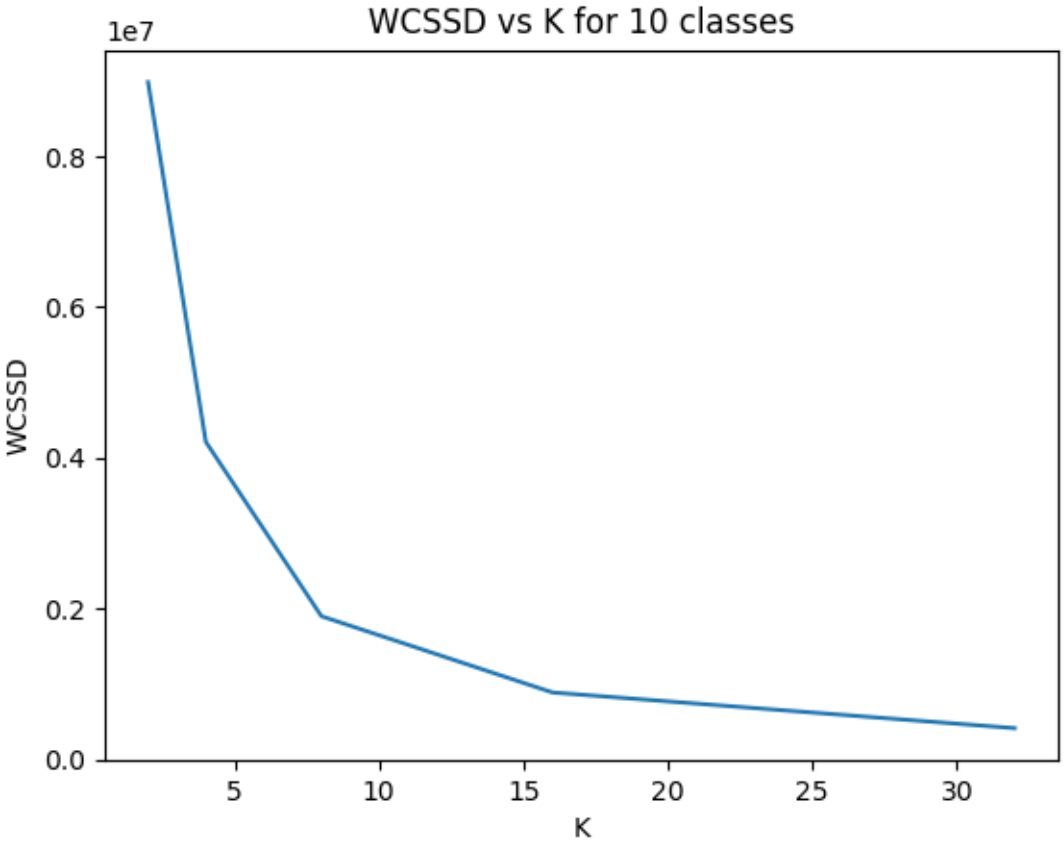

Figure 12: WCSSD vs K for 10 classes

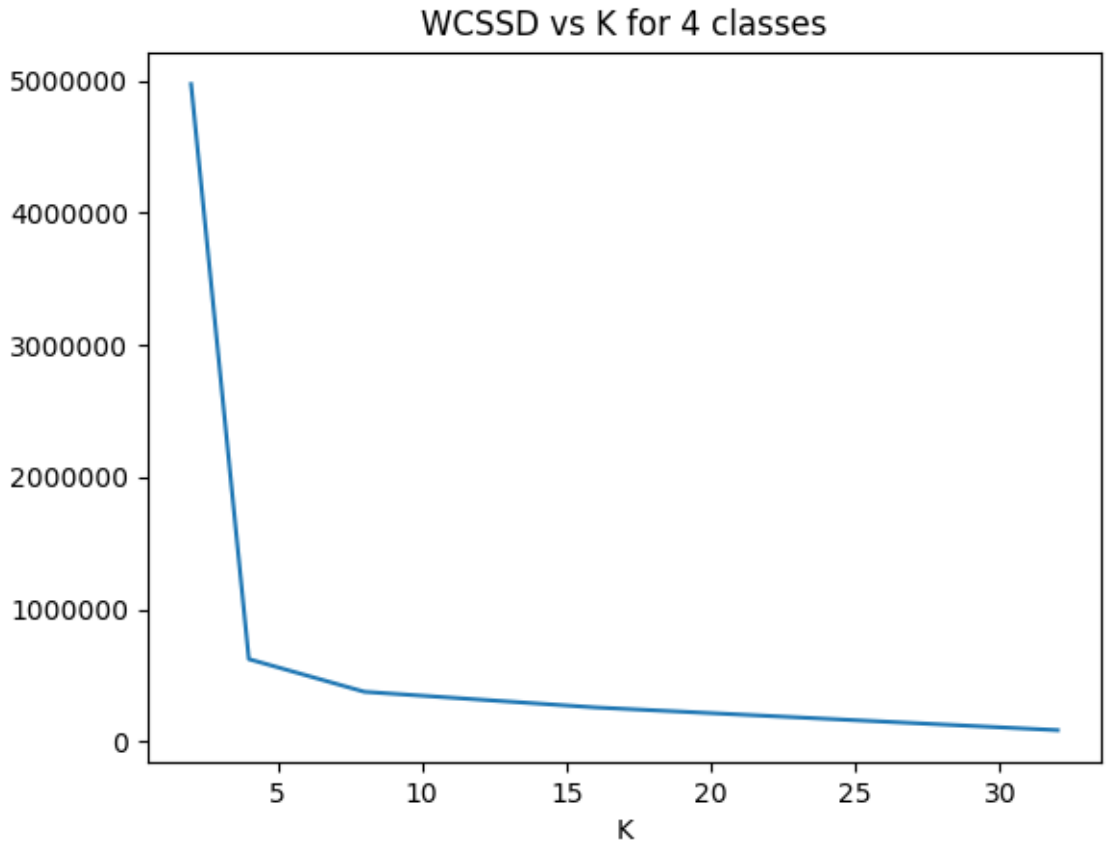

Figure 13: WCSSD vs K for 4 classes

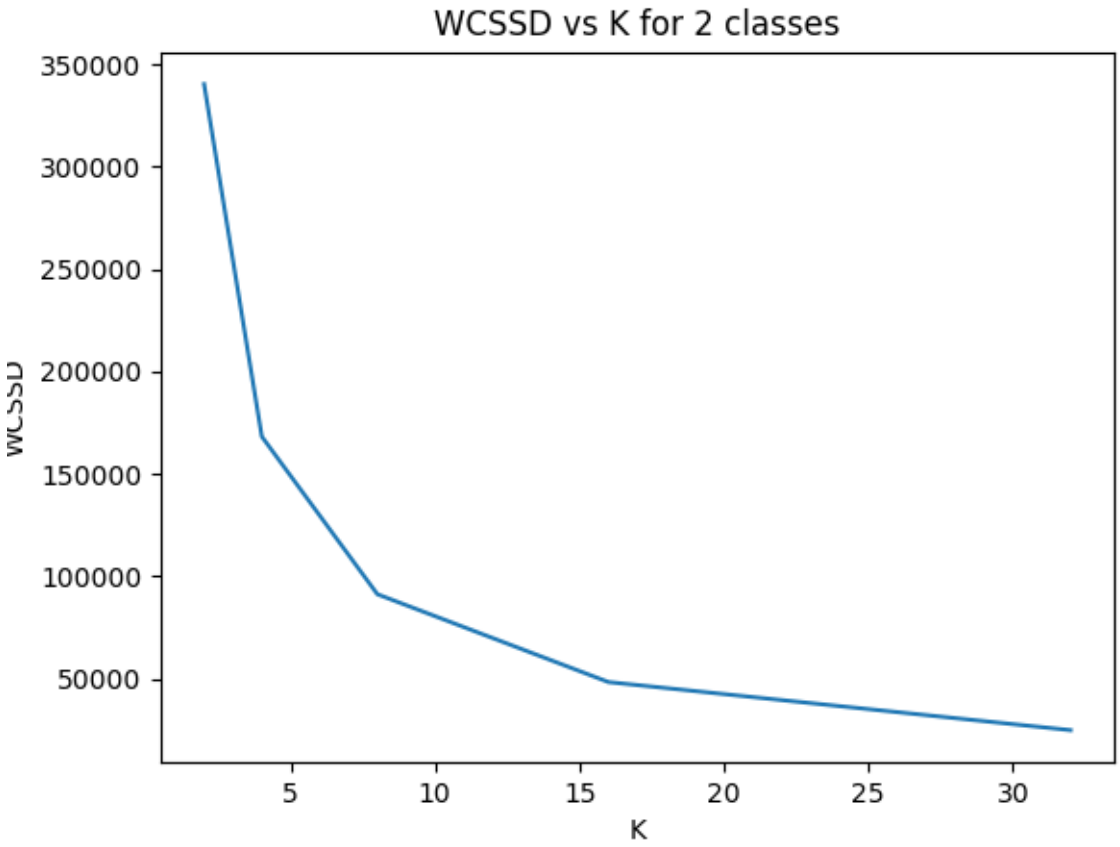

Figure 14: WCSSD vs K for 2 classes

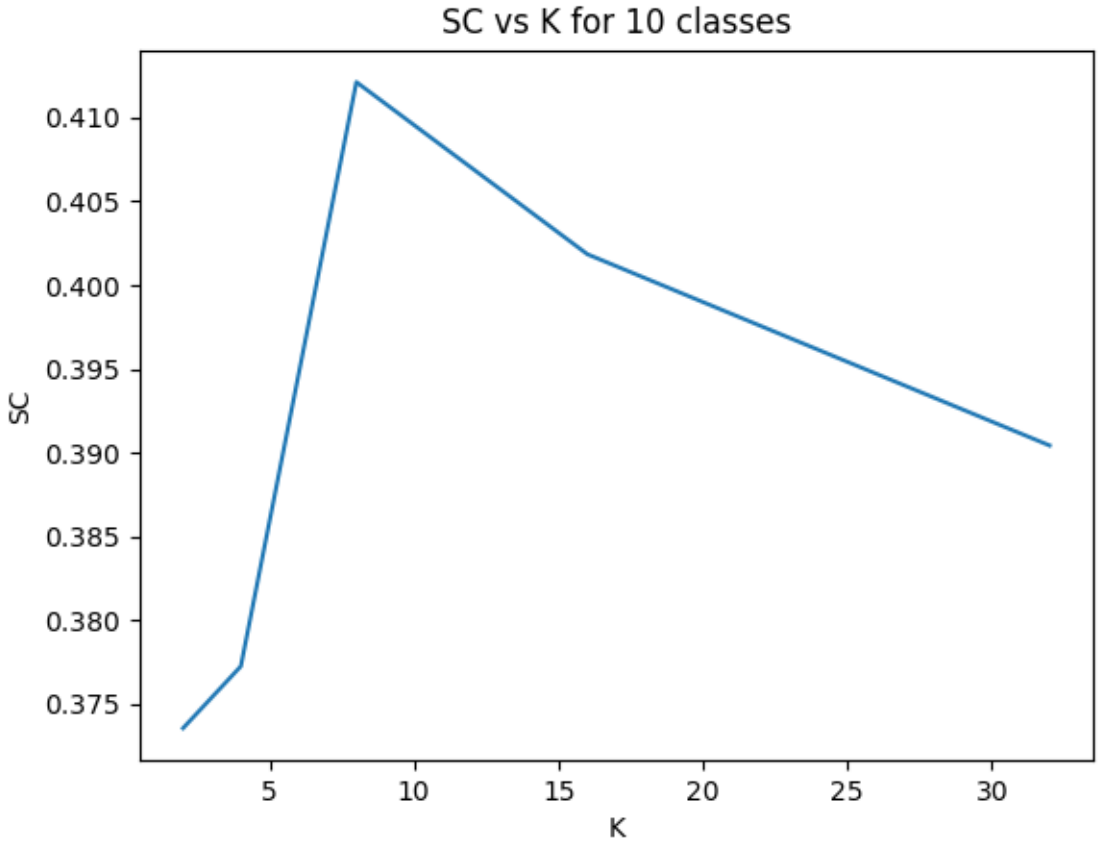

Figure 15: SC vs K for 10 classes

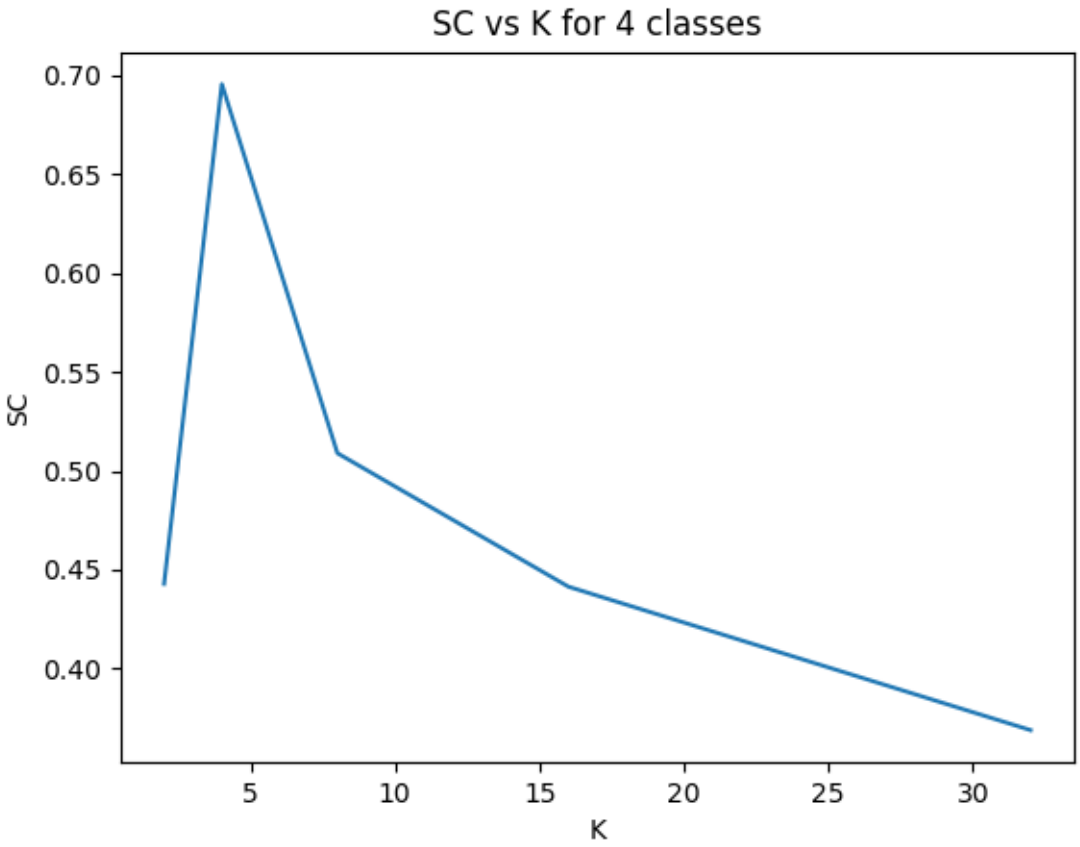

Figure 16: SC vs K for 4 classes

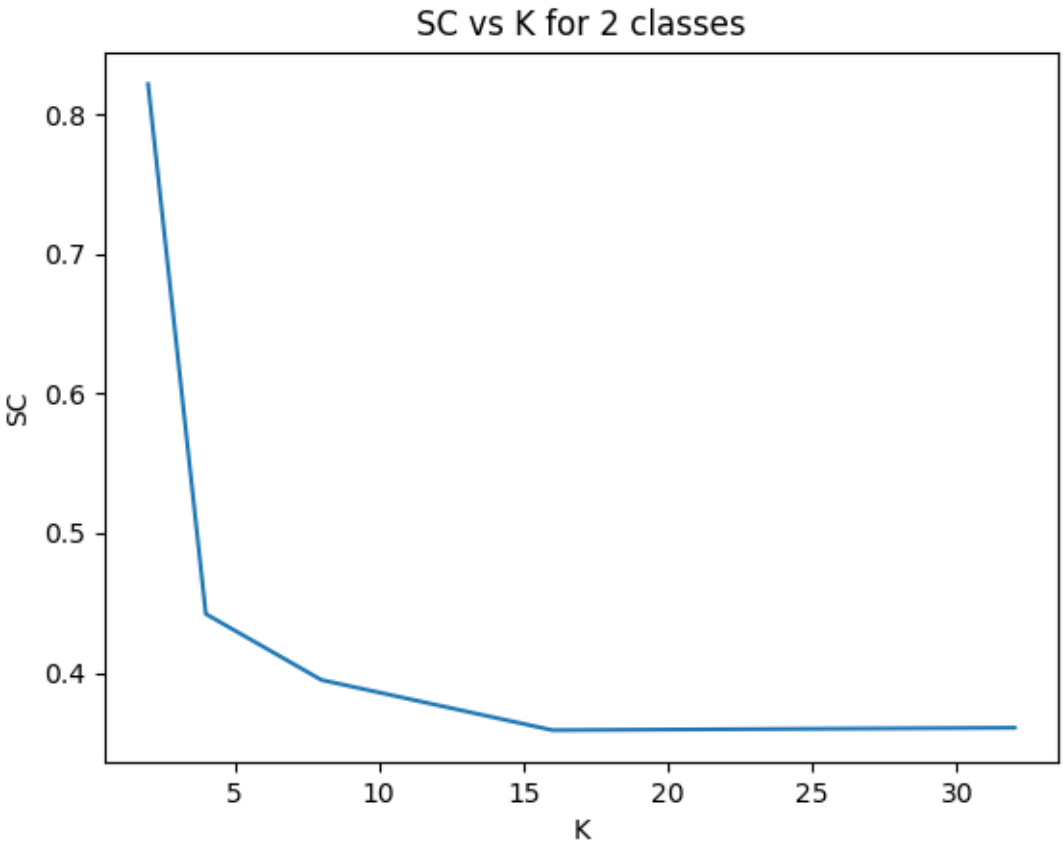

Figure 17: SC vs K for 2 classes

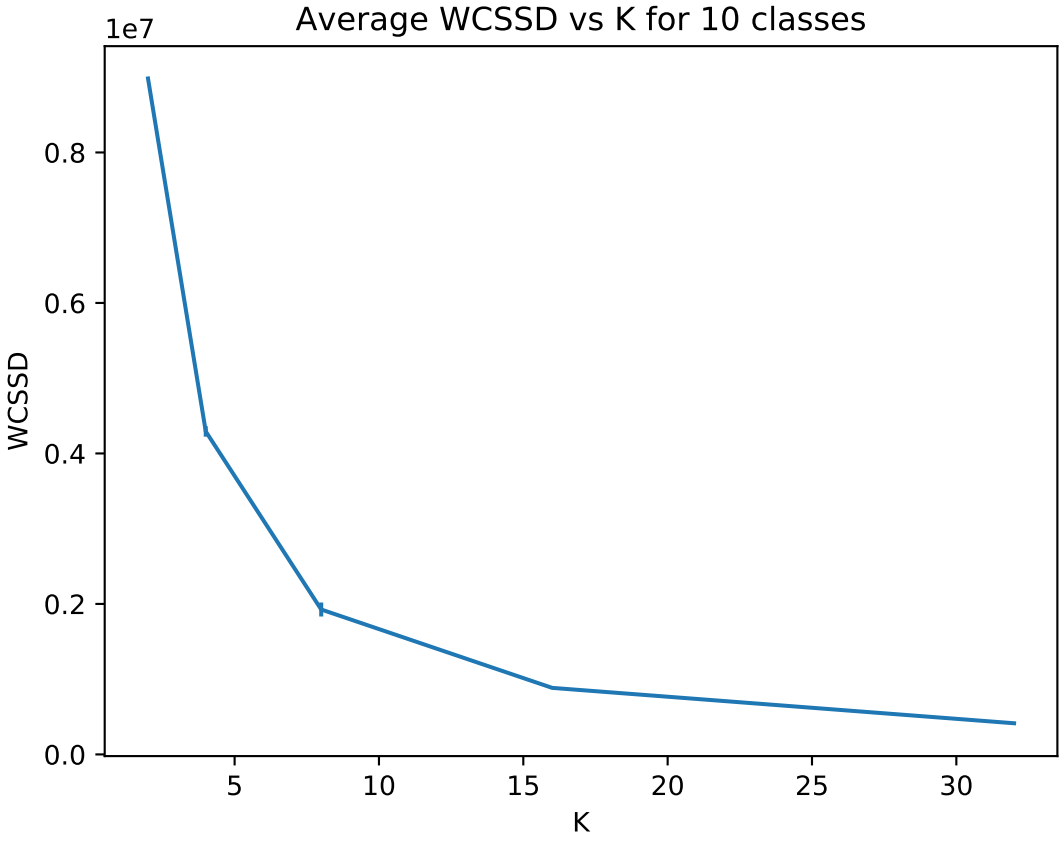

Figure 18: Average WCSSD vs K for 10 classes

4 classes, SC increases till K=4 and then drops off. For dataset with 10 classes, SC increases till K=8, and then drops off till K=32.

3.

Figures 18-20 show the average and standard deviation of WCSSD for the three different datasets across the ten runs. Figures 21-23 show the average and standard deviation of SC for the three different datasets across the ten runs.
What these results say about the sensitivity of K-means to initialization of centroids: Since SC scores have a large standard deviation, the clustering obtained with K-means is highly sensitive to the chosen initial centroids.

4. NMI Computation

NMI for 10 classes with K=8 is 0.34
NMI for 4 classes with K=4 is 0.36
NMI for 2 classes with K=2 is 0.49.
We see that as the number of classes decreases, the NMI score increases towards its maximum threshold 0.5. This explains that as the number of classes decreases, the cluster labels are more in agreement with the class labels.
I used the formula given in the slides for NMI computation.

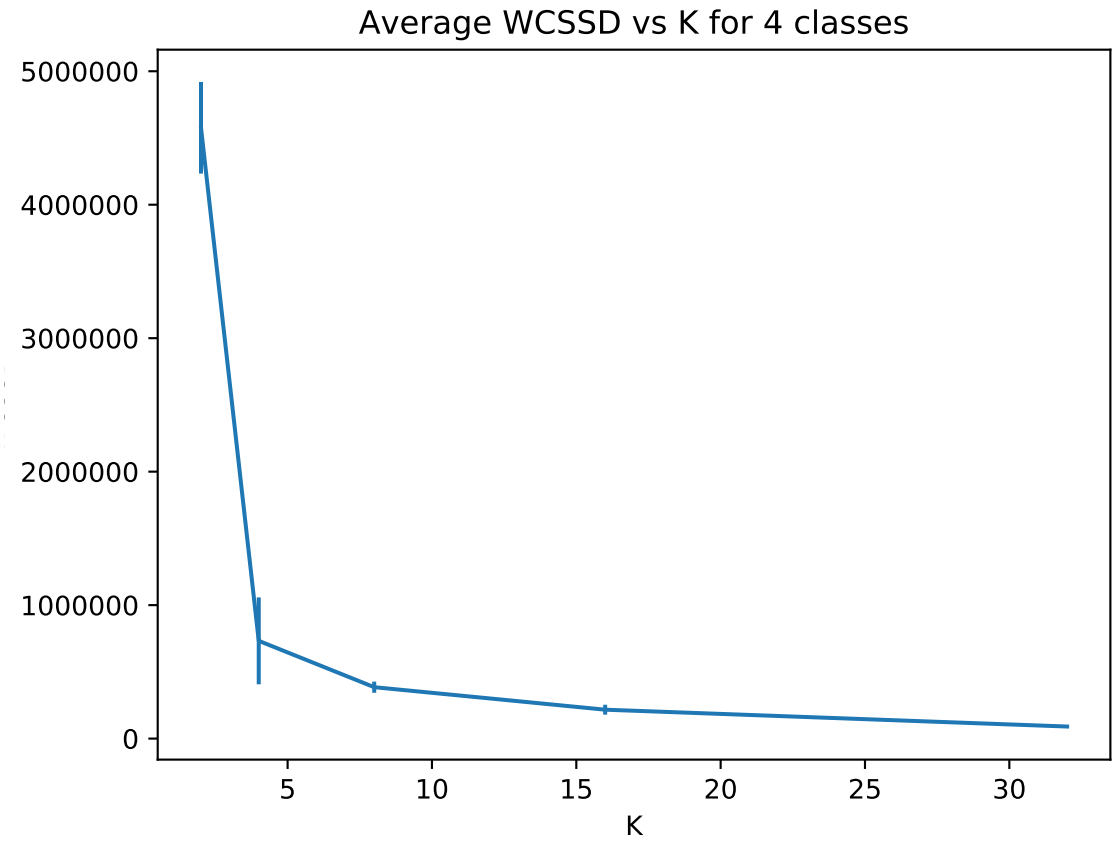

Figure 19: Average WCSSD vs K for 4 classes

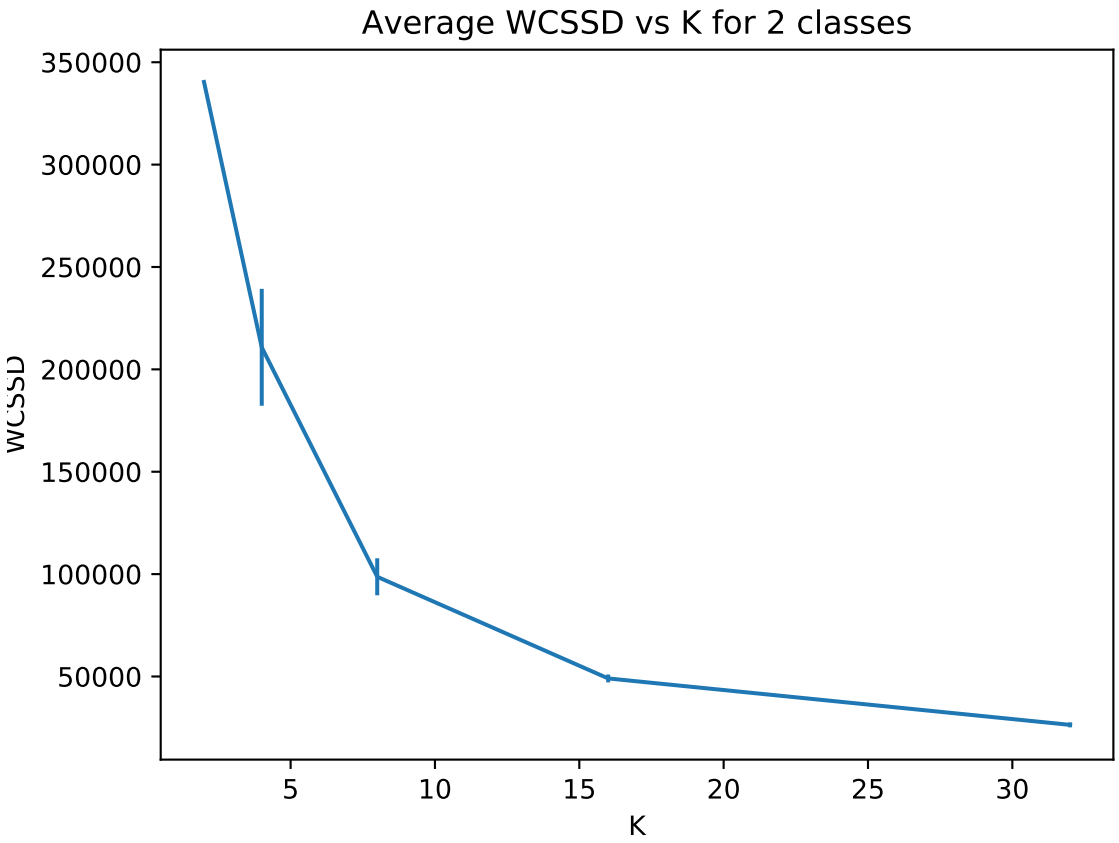

Figure 20: Average WCSSD vs K for 2 classes

Figure 21: Average SC vs K for 10 classes

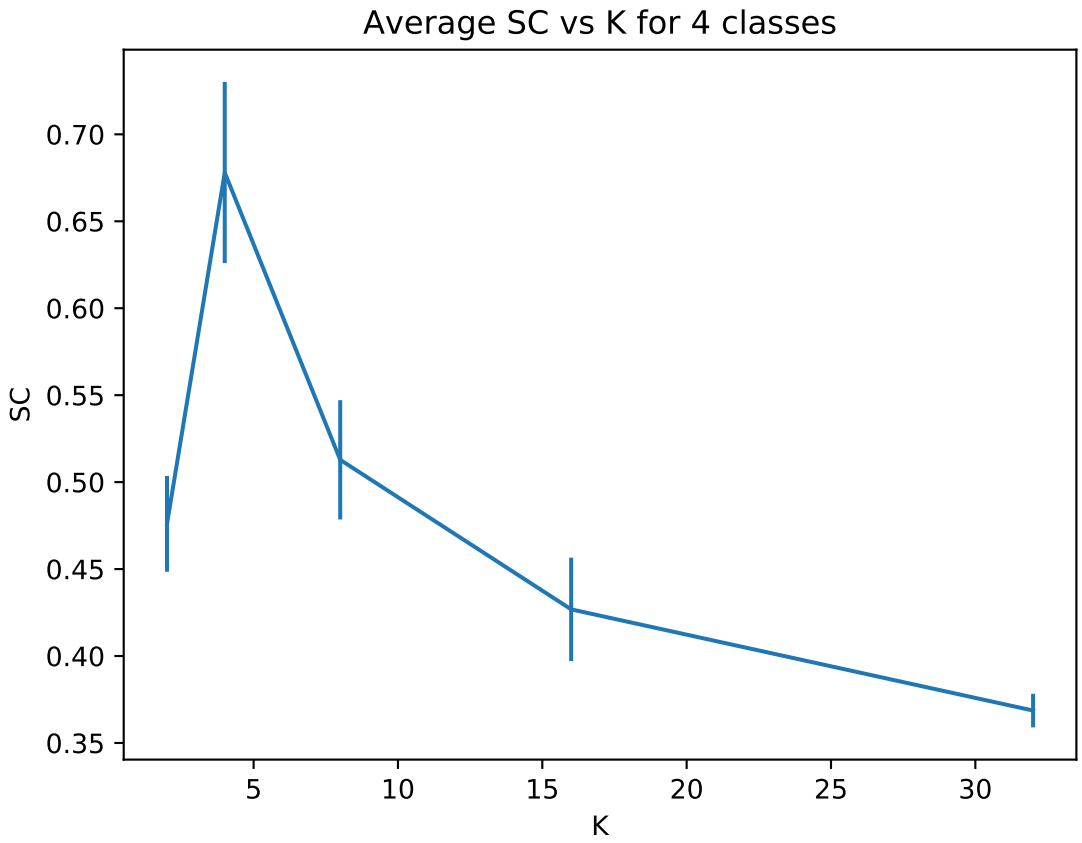

Figure 22: Average SC vs K for 4 classes

Figure 23: Average SC vs K for 2 classes

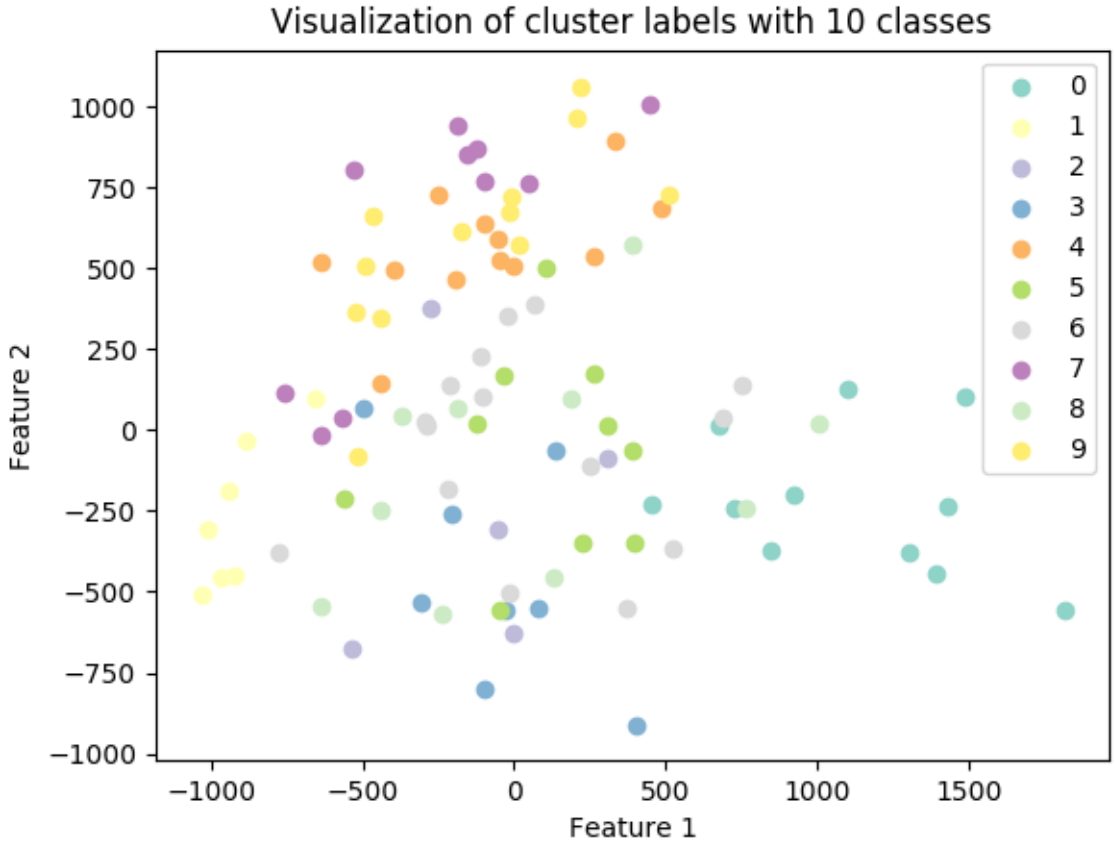

Figure 24: Visualization of cluster labels with 10 classes

Figures 24-26 show the visualization of cluster labels of 1000 randomly selected examples in 2D. From the plots, it is evident that the data of 10 classes is too crowded and not well seperated. The data of 4 classes are better seperated, there are however few violations. The data of 2 classes is best seperated as the two clusters are far away and the number of clusters is less.

C. Comparison to hierarchical clustering

1. Single Linkage

Figures 27-29 show the dendrograms corresponding to agglomerative hierarchical clustering using single linkage for different datasets.

2. Complete and Average Linkage

Figure 30-32 show the dendrograms corresponding to agglomerative hierarchical clustering using complete linkage for the three different datasets. Figure 33-35 show the dendrograms corresponding to agglomerative hierarchical clustering using average linkage for the three different datasets.

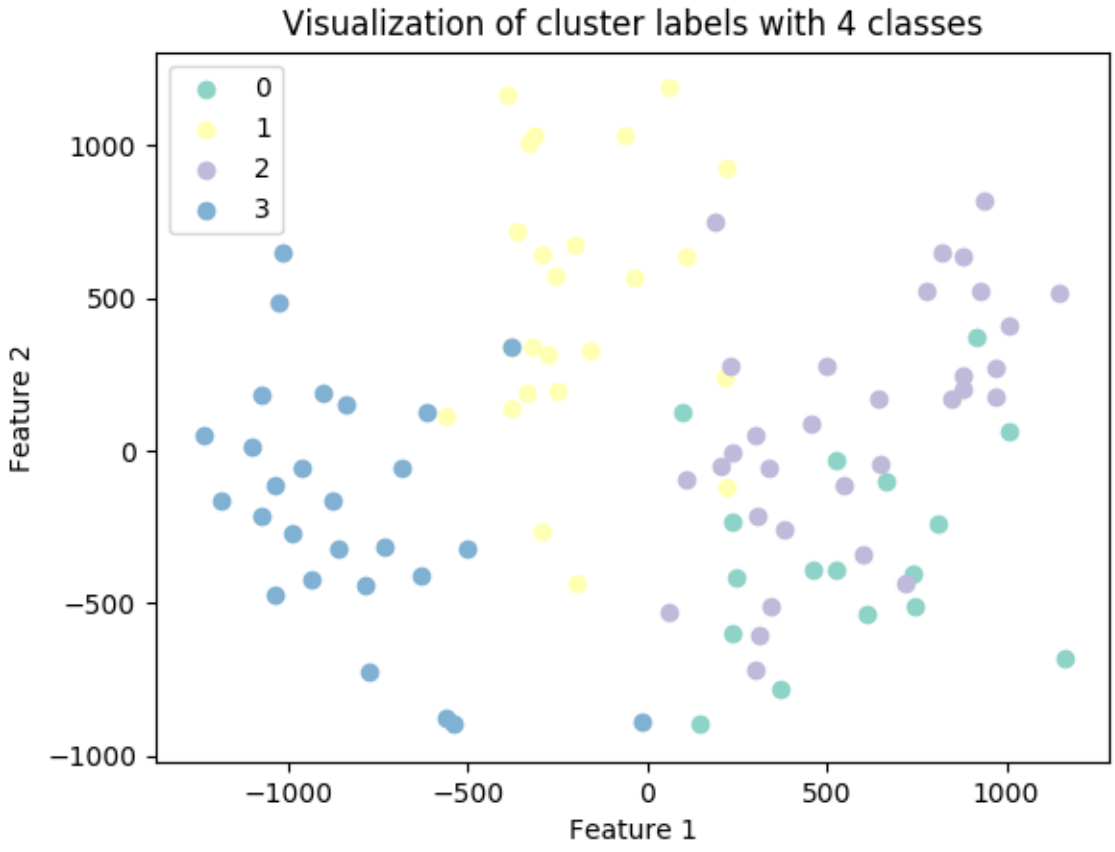

Figure 25: Visualization of cluster labels with 4 classes

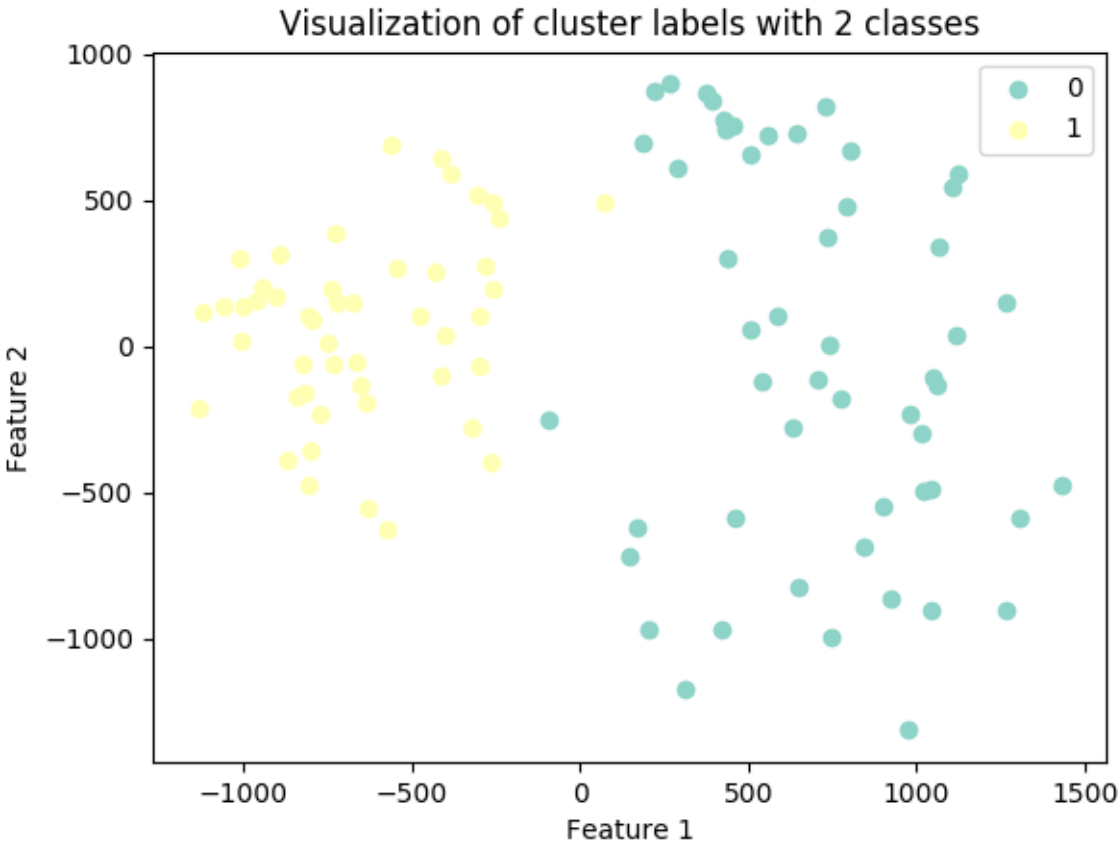

Figure 26: Visualization of cluster labels with 2 classes

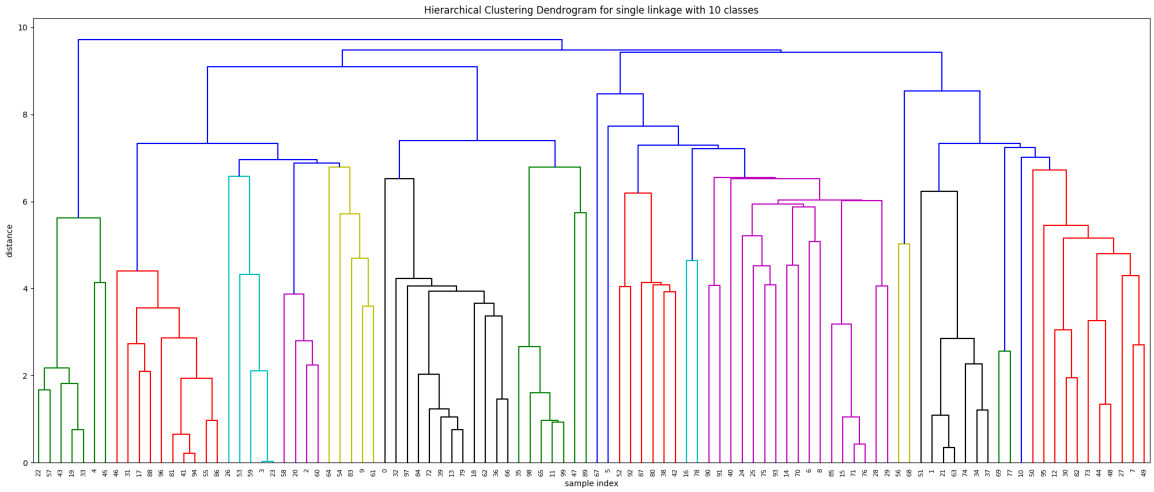

Figure 27: Hierarchical Clustering Dendrogram for single linkage with 10 classes

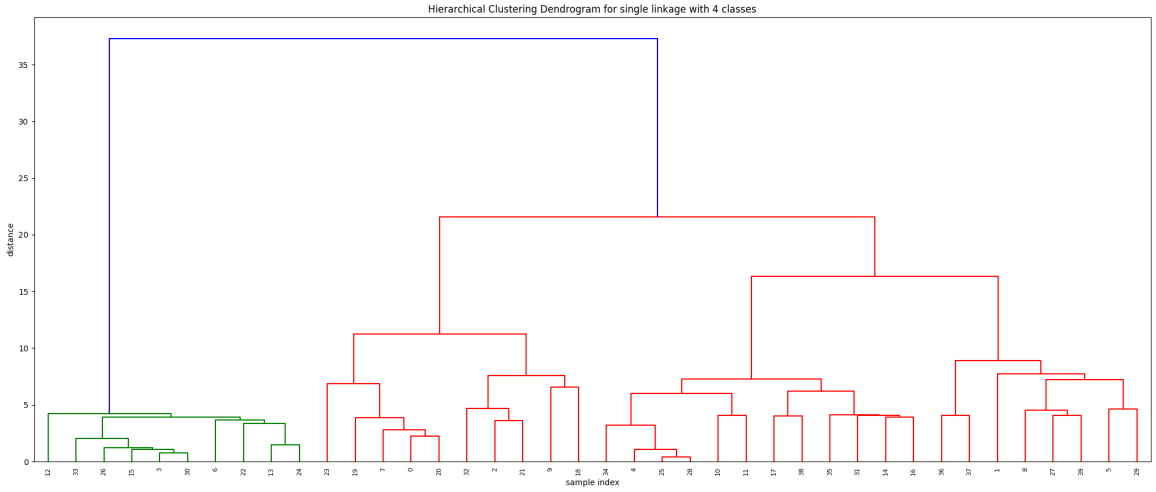

Figure 28: Hierarchical Clustering Dendrogram for single linkage with 4 classes

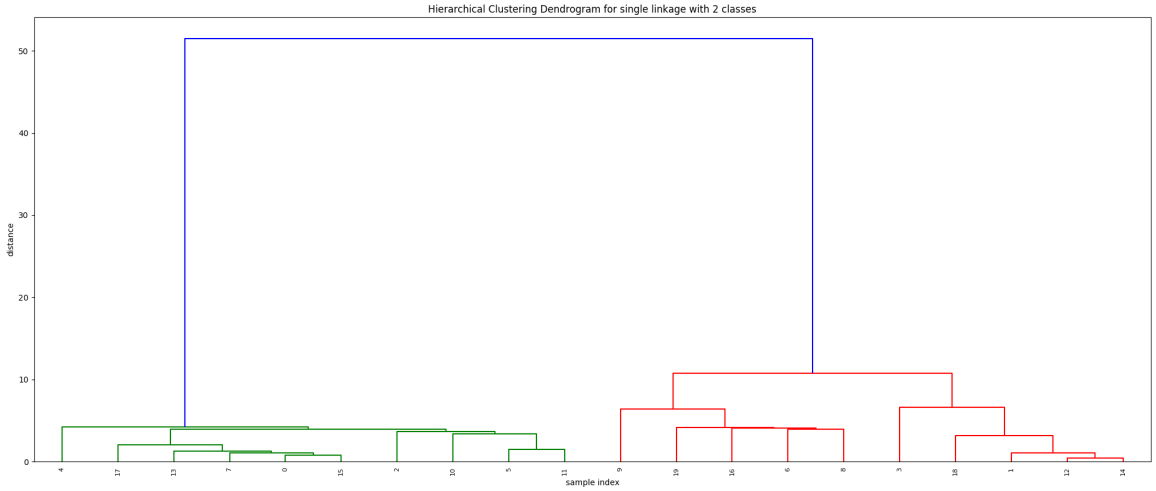

Figure 29: Hierarchical Clustering Dendrogram for single linkage with 2 classes

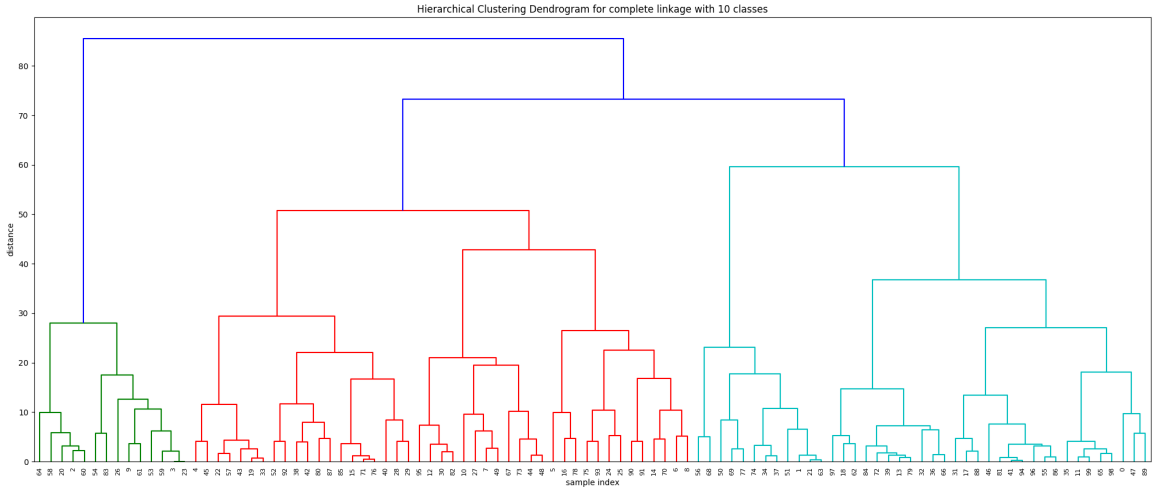

Figure 30: Hierarchical Clustering Dendrogram for complete linkage with 10 classes

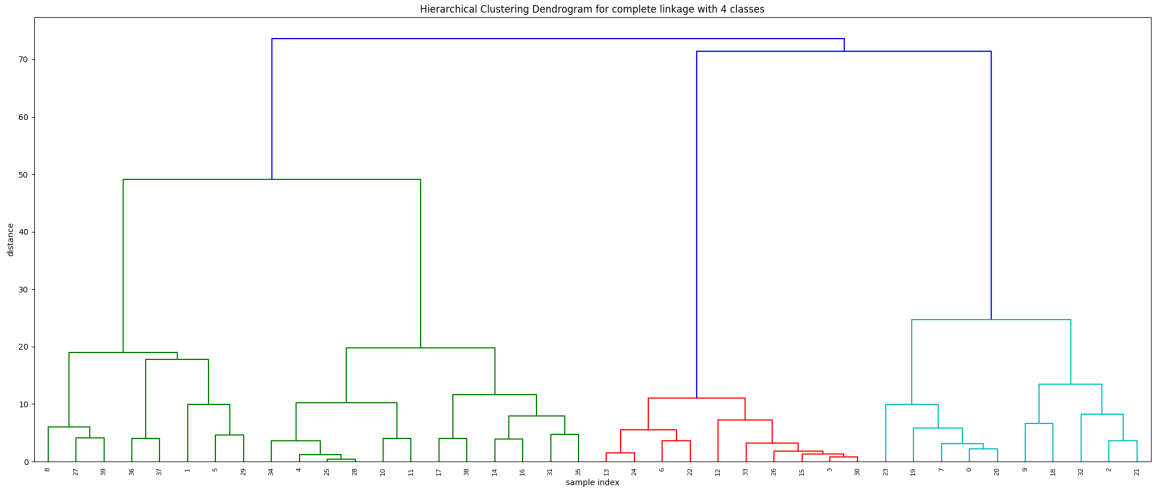

Figure 31: Hierarchical Clustering Dendrogram for complete linkage with 4 classes

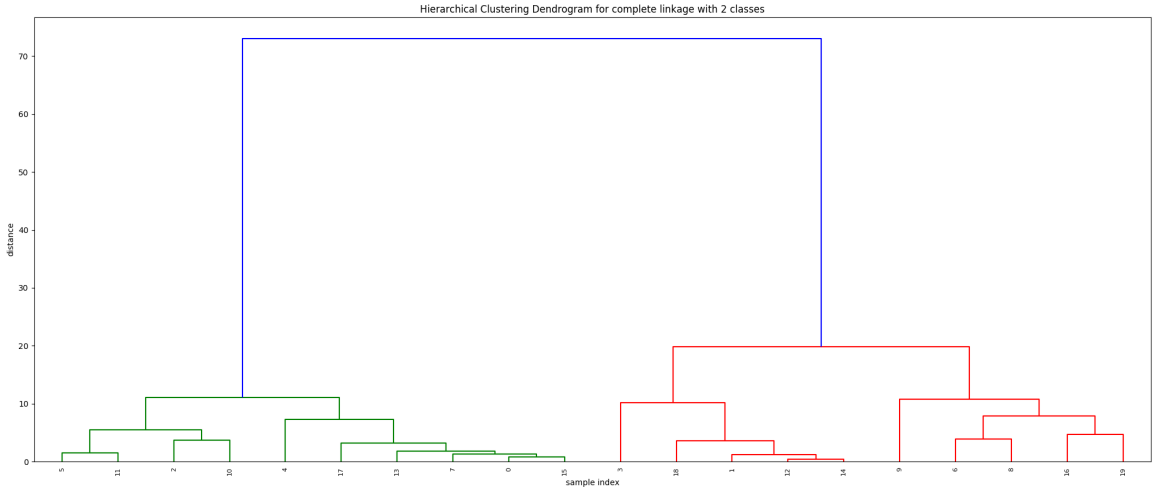

Figure 32: Hierarchical Clustering Dendrogram for complete linkage with 2 classes

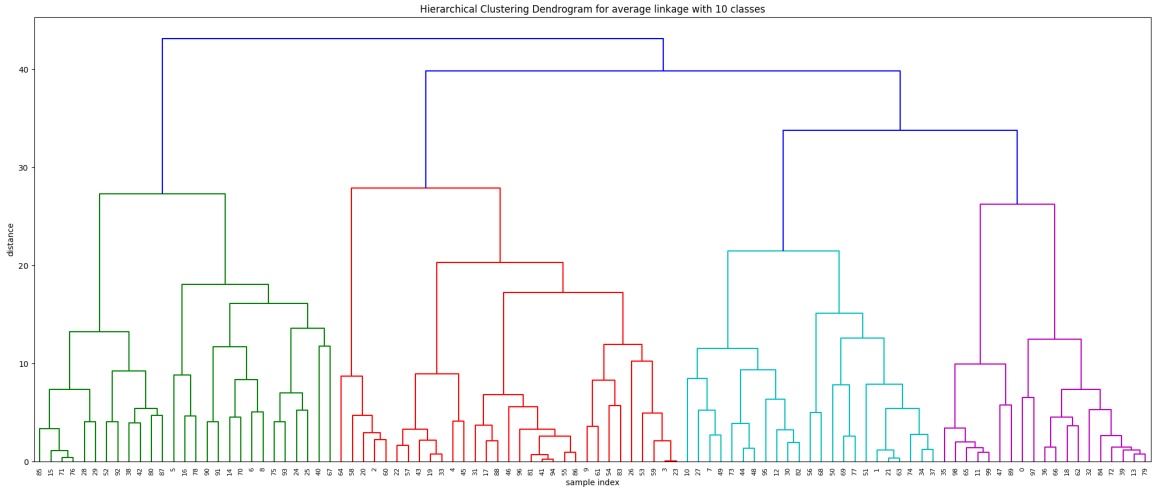

Figure 33: Hierarchical Clustering Dendrogram for average linkage with 10 classes

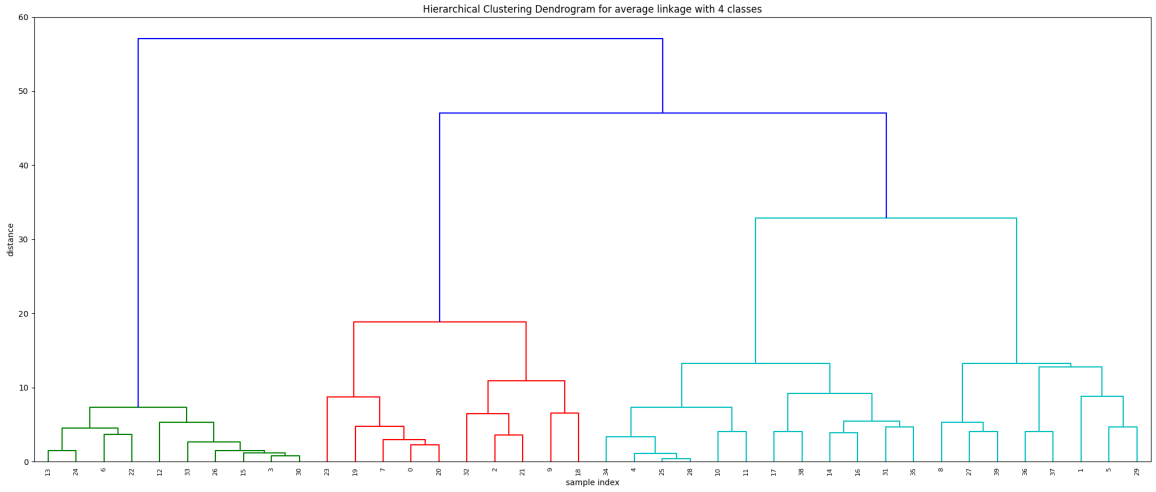

Figure 34: Hierarchical Clustering Dendrogram for average linkage with 4 classes

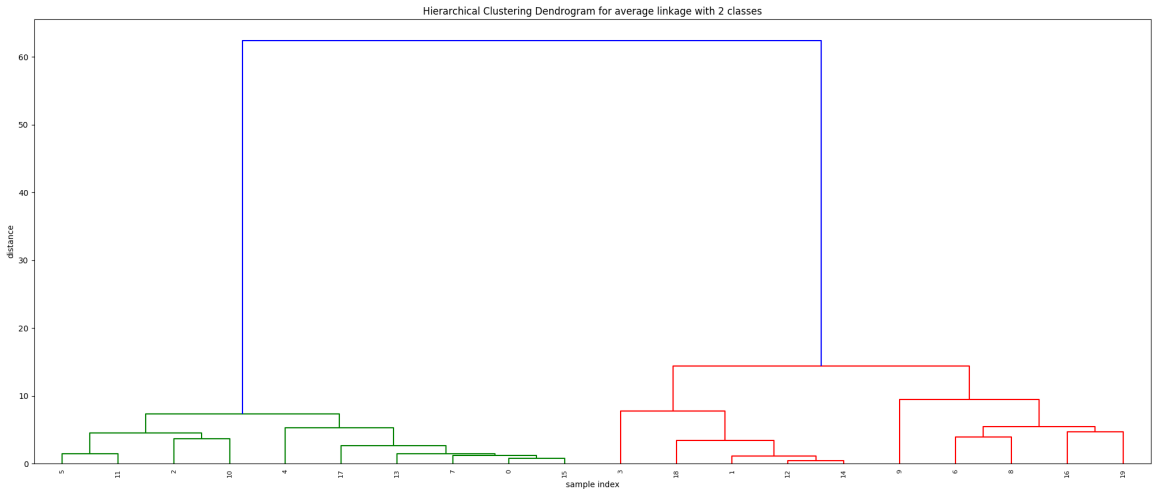

Figure 35: Hierarchical Clustering Dendrogram for average linkage with 2 classes

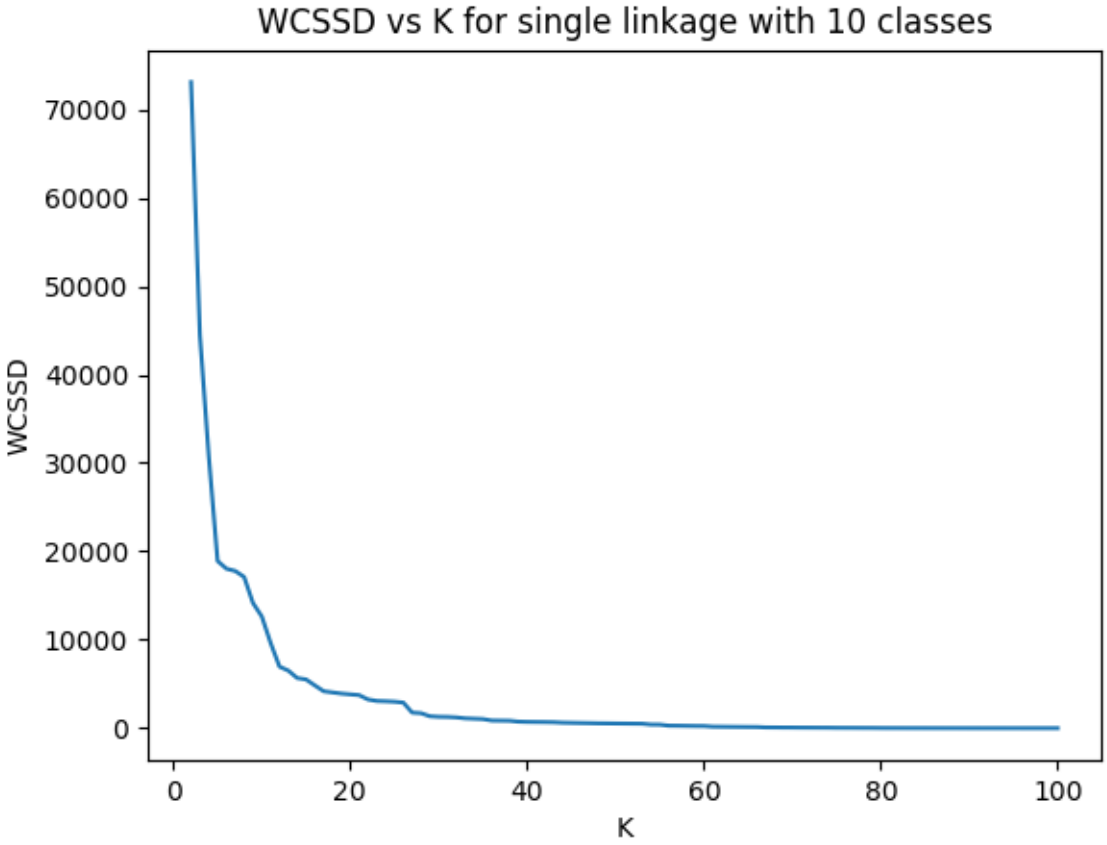

Figure 36: WCSSD vs K for single linkage with 10 classes

3. WCSSD and SC as a function of K

Figures 36-38 show the within-cluster sum of squared distances (WCSSD) as a function of K using single linkage for the three different datasets. Figures 39-41 show the silhouette coefficient (SC) as a function of K using single linkage for the three datasets. Figures 42-44 show the within-cluster sum of squared distances (WCSSD) as a function of K using complete linkage for the three different datasets. Figures 45-47 show the silhouette coefficient (SC) as a function of K using complete linkage for the three datasets. Figures 48-50 show the within-cluster sum of squared distances (WCSSD) as a function of K using average linkage for the three different datasets. Figures 51-53 show the silhouette coefficient (SC) as a function of K using average linkage for the three datasets.

4. Choose K

I choose K with highest value of SC.
For 10 classes with single linkage, the best K is 40 with SC 0.42.
For 10 classes with complete linkage, the best K is 10 with SC 0.51.
For 10 classes with average linkage, the best K is 12 with SC 0.51.
For 4 classes with single linkage, the best K is 4 with SC 0.71.
For 4 classes with complete linkage, the best K is 4 with SC 0.71.
For 4 classes with average linkage, the best K is 4 with SC 0.71.

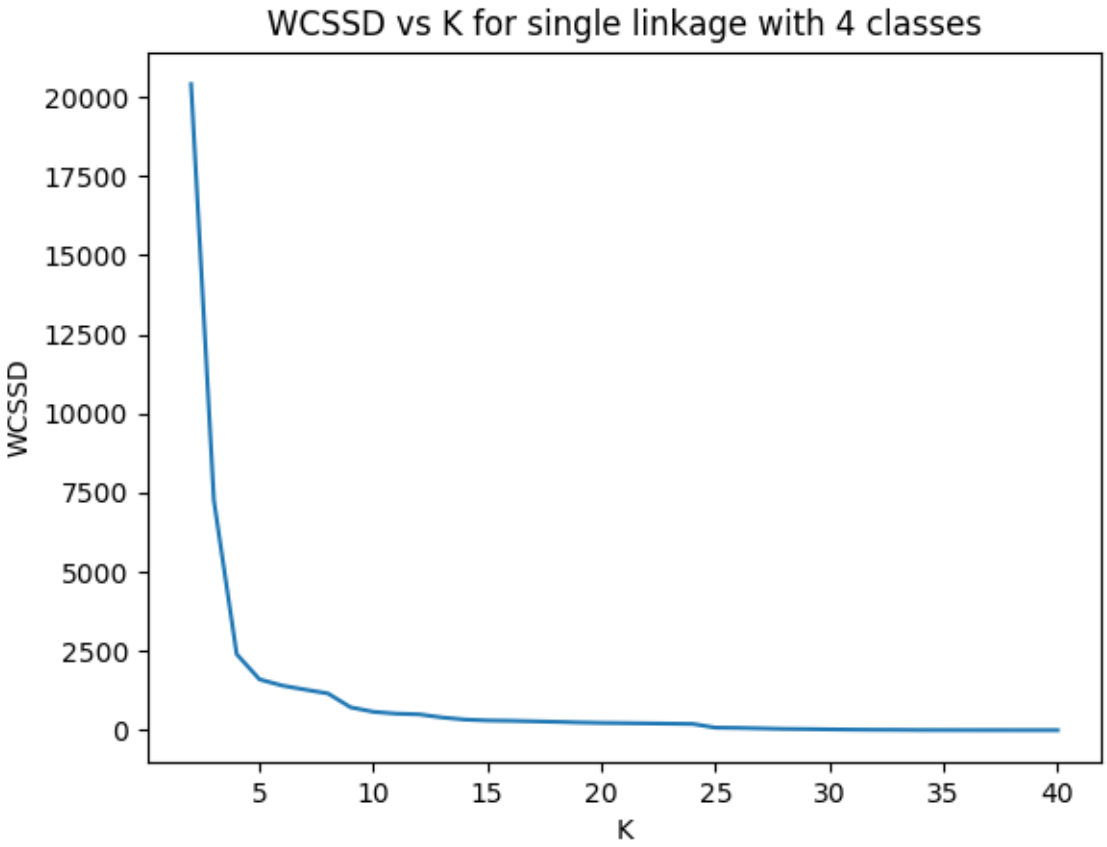

Figure 37: WCSSD vs K for single linkage with 4 classes

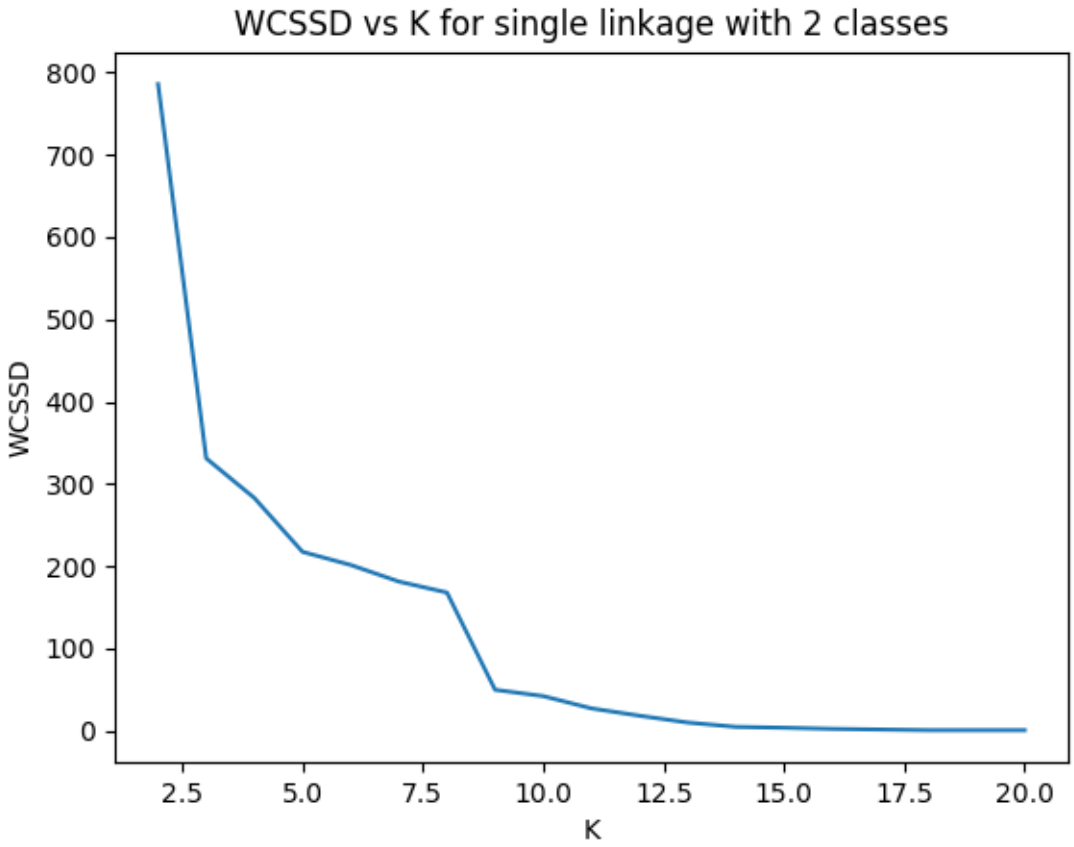

Figure 38: WCSSD vs K for single linkage with 2 classes

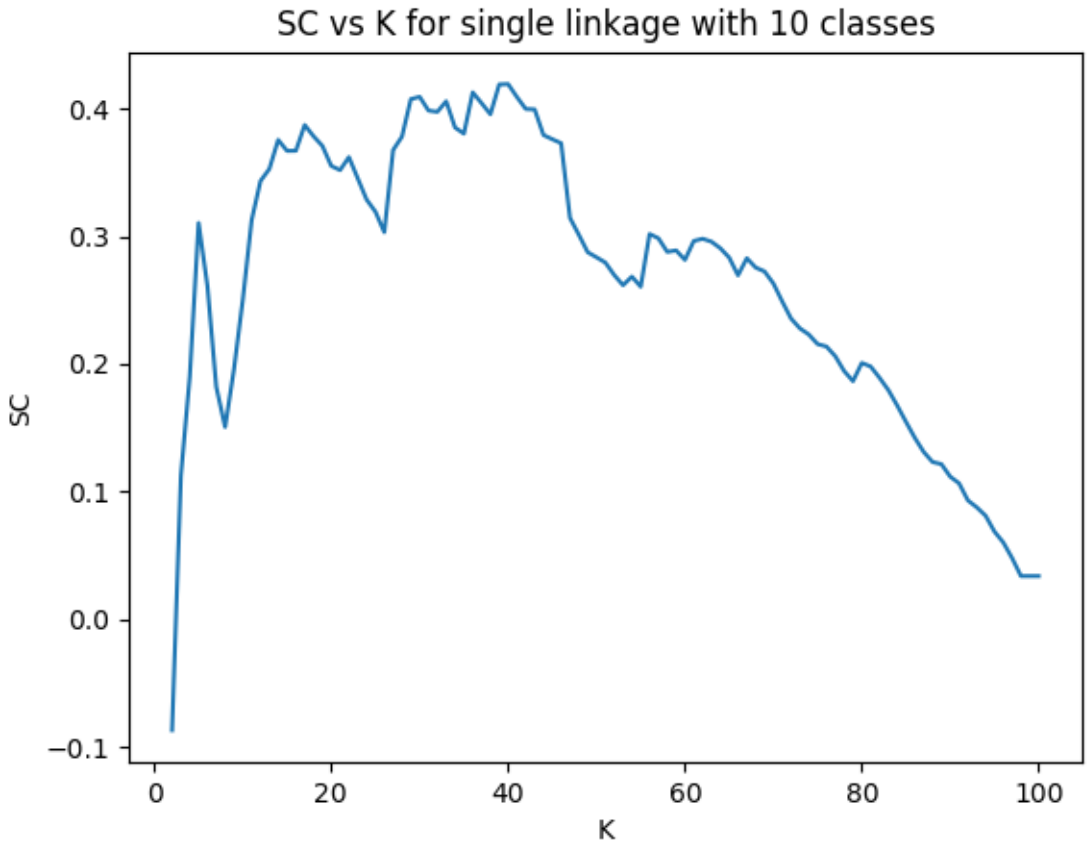

Figure 39: SC vs K for single linkage with 10 classes

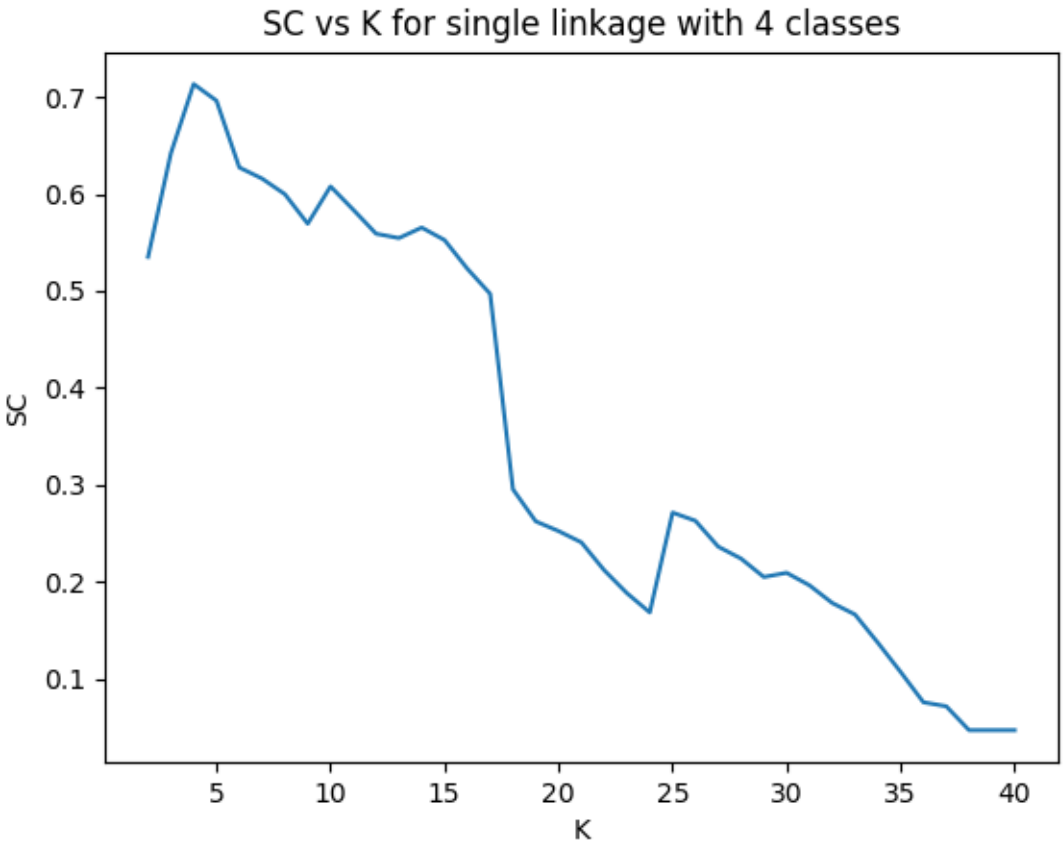

Figure 40: SC vs K for single linkage with 4 classes

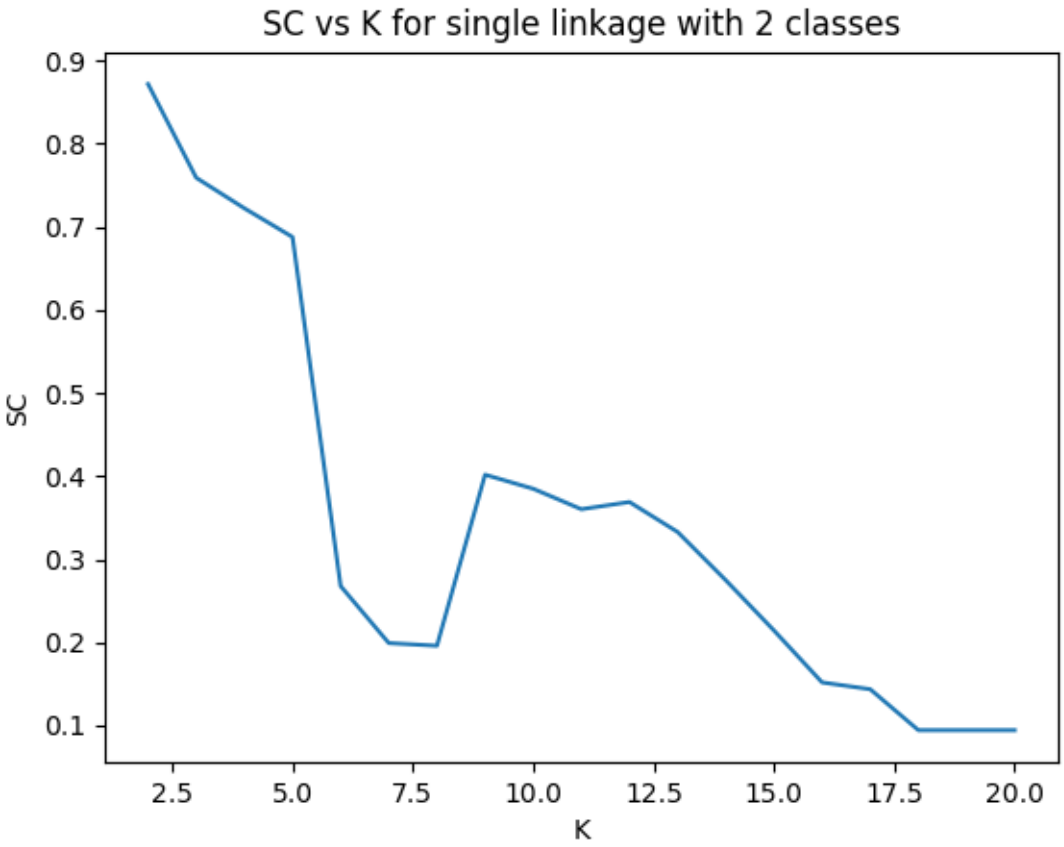

Figure 41: SC vs K for single linkage with 2 classes

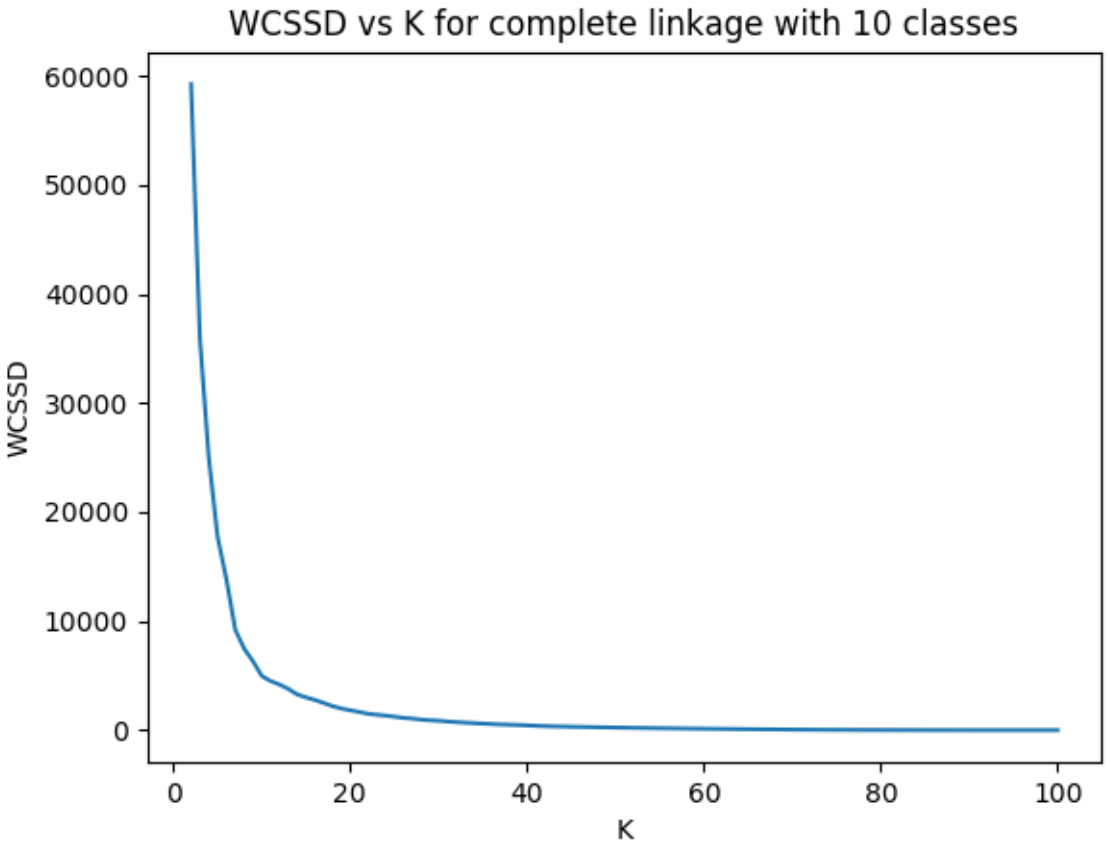

Figure 42: WCSSD vs K for complete linkage with 10 classes

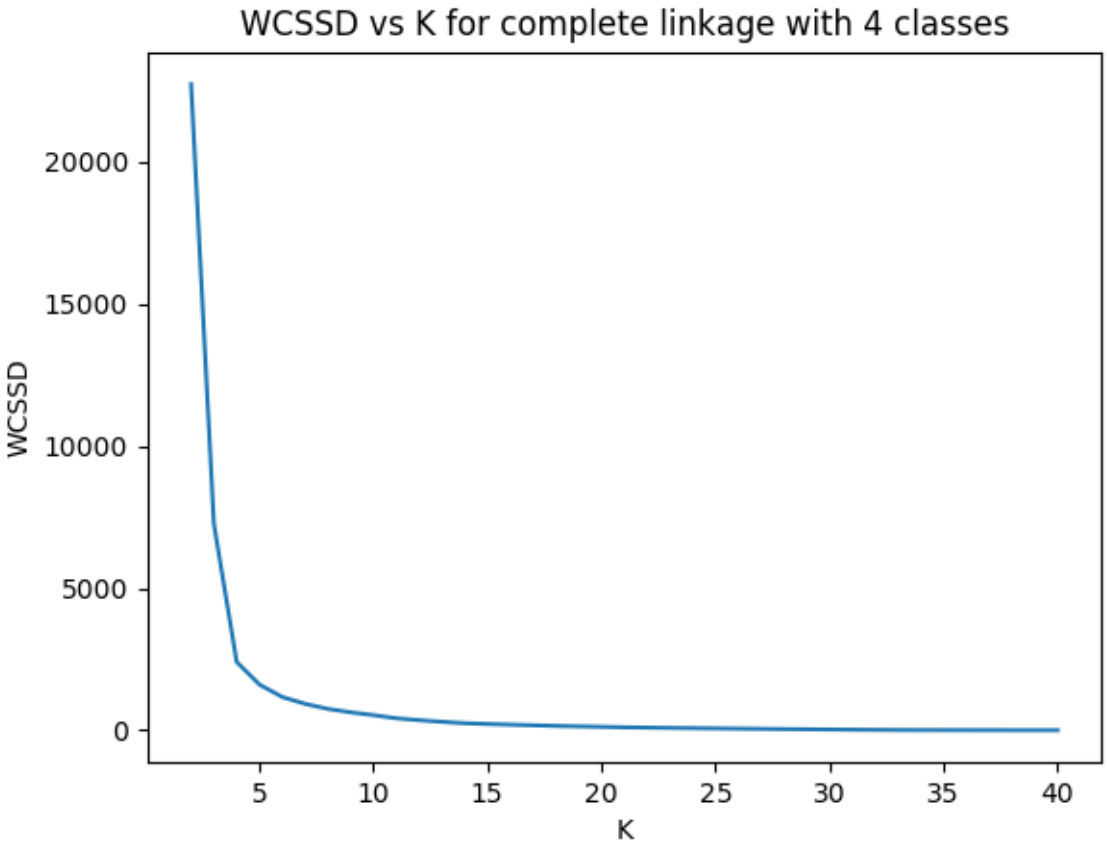

Figure 43: WCSSD vs K for complete linkage with 4 classes

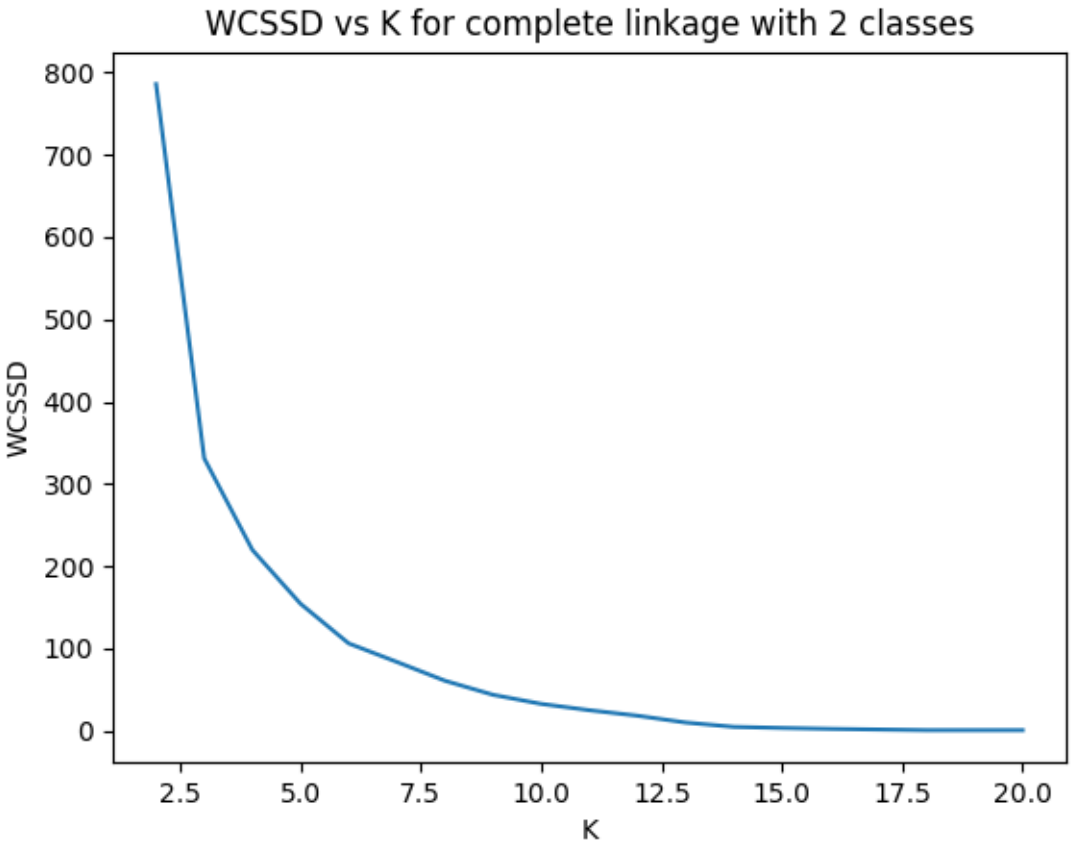

Figure 44: WCSSD vs K for complete linkage with 2 classes

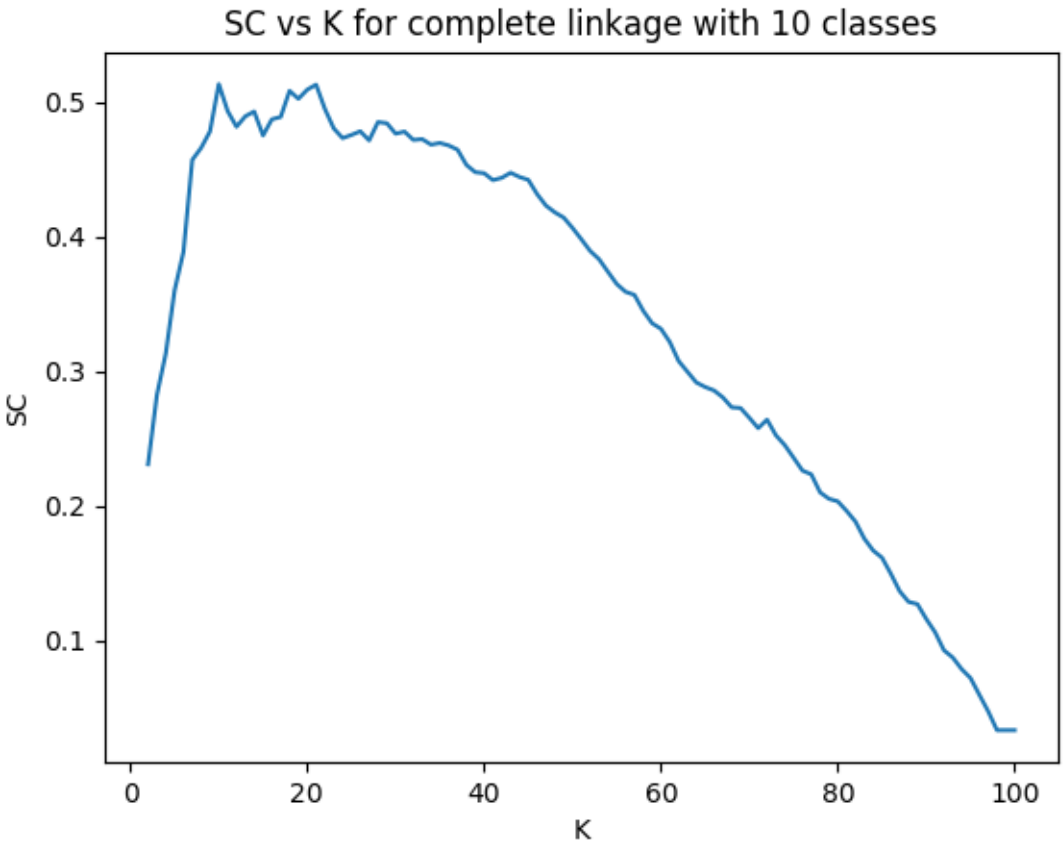

Figure 45: SC vs K for complete linkage with 10 classes

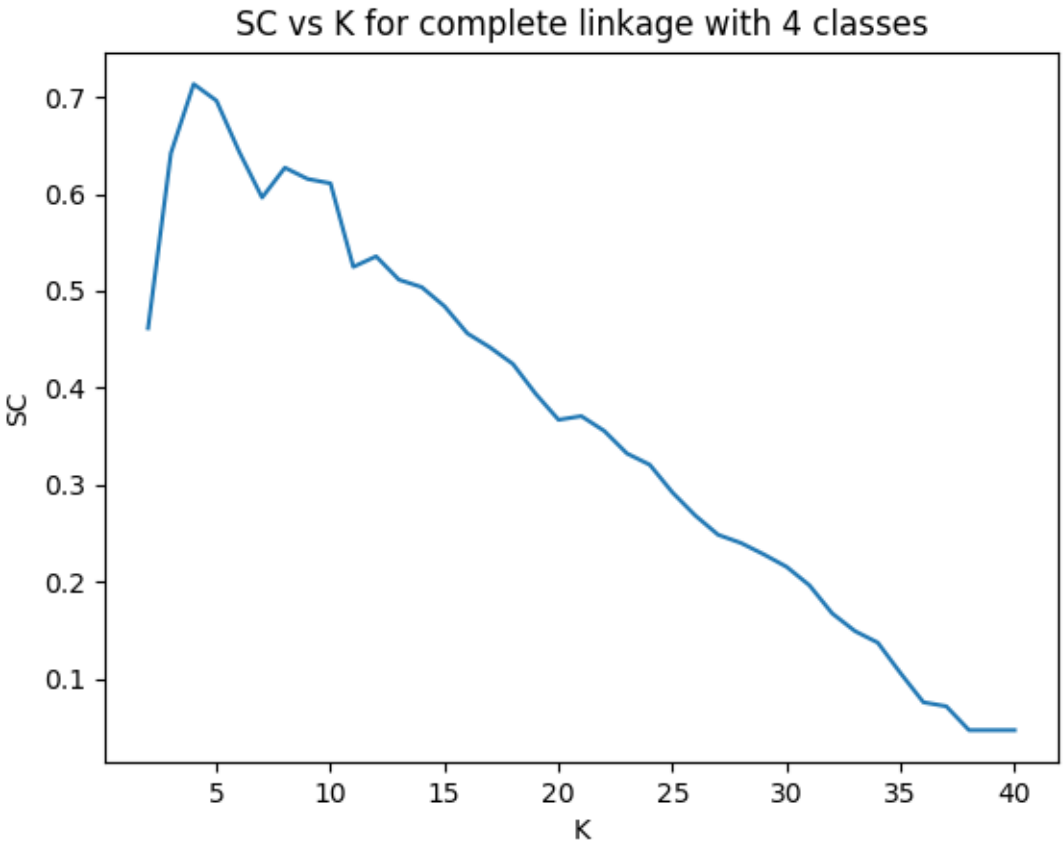

Figure 46: SC vs K for complete linkage with 4 classes

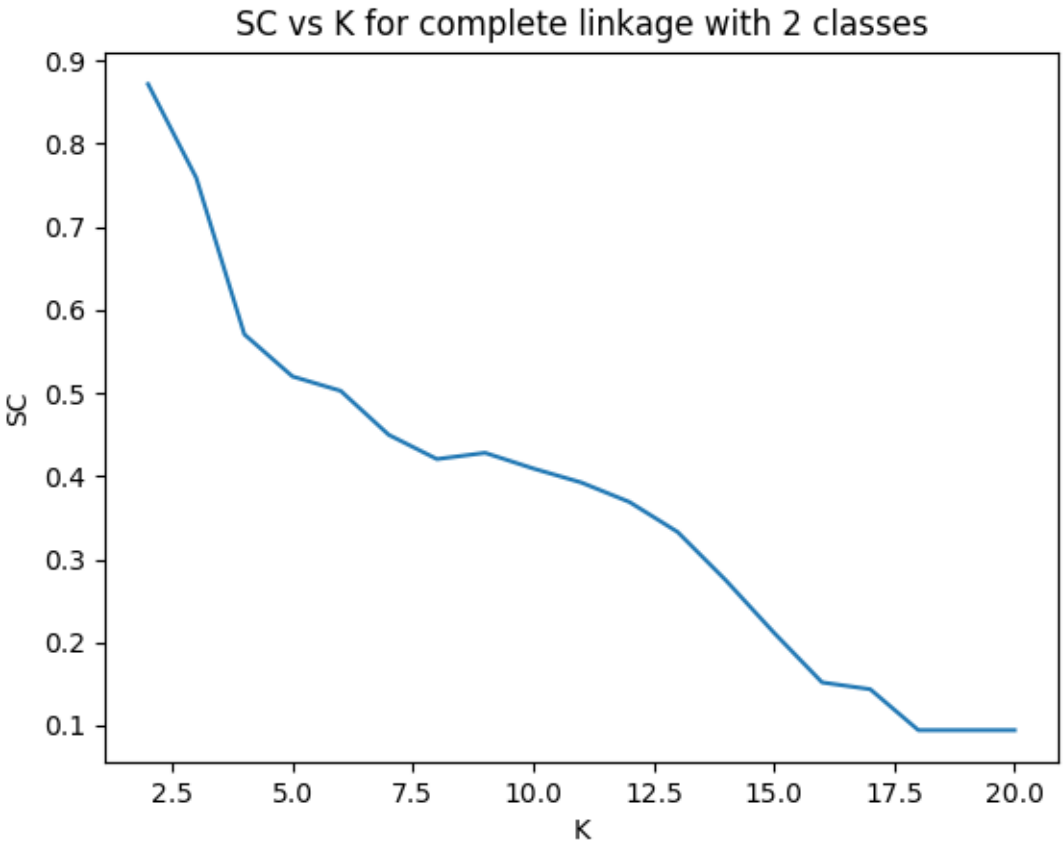

Figure 47: SC vs K for complete linkage with 2 classes

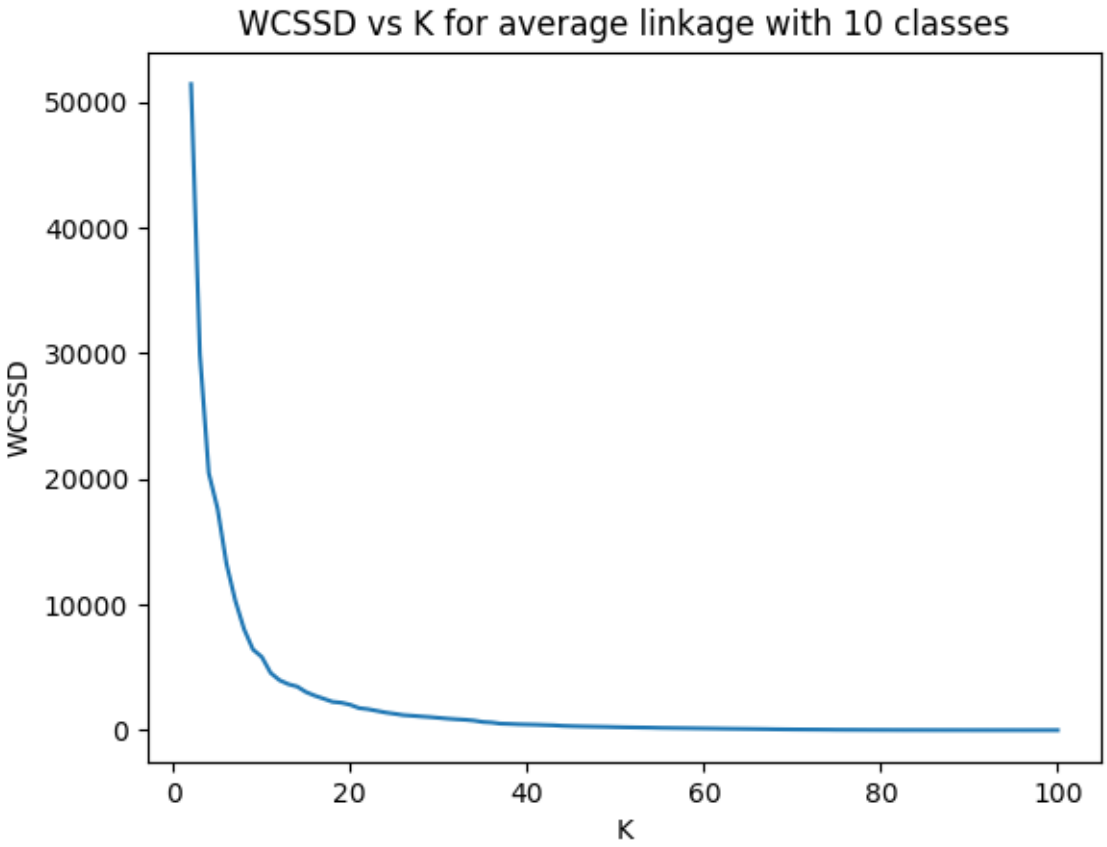

Figure 48: WCSSD vs K for average linkage with 10 classes

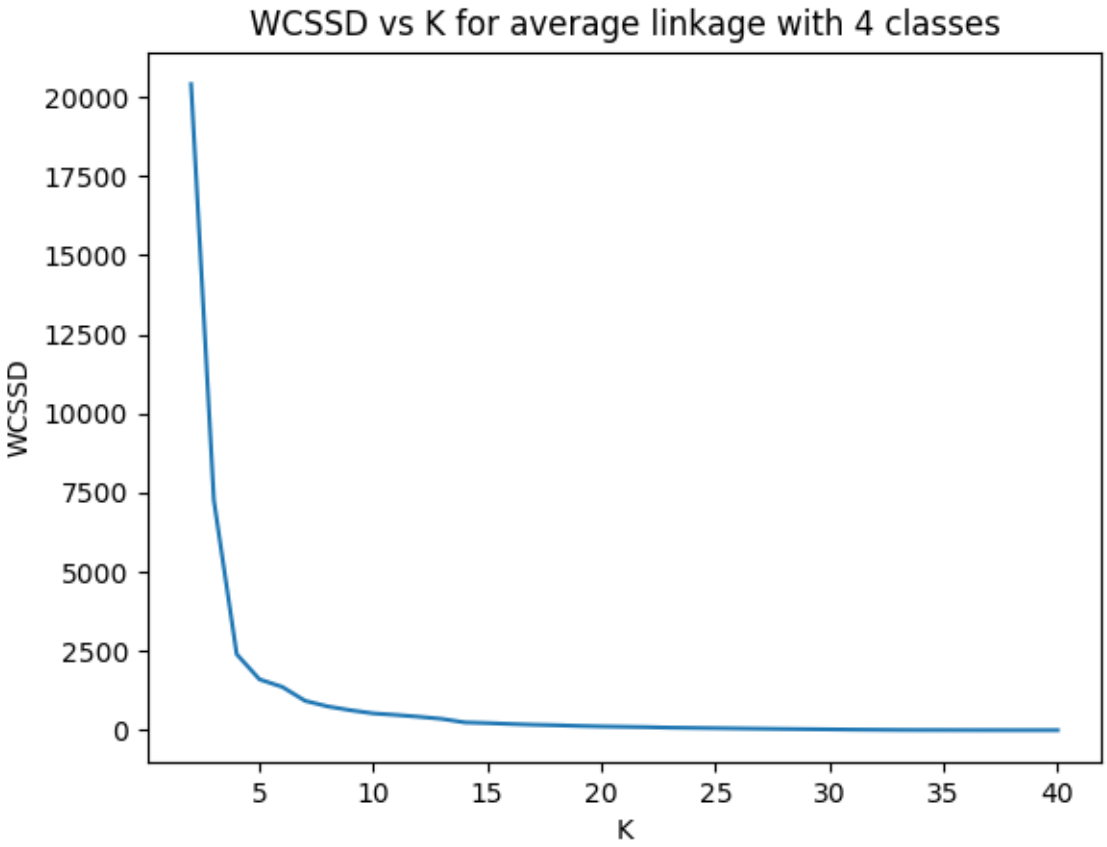

Figure 49: WCSSD vs K for average linkage with 4 classes

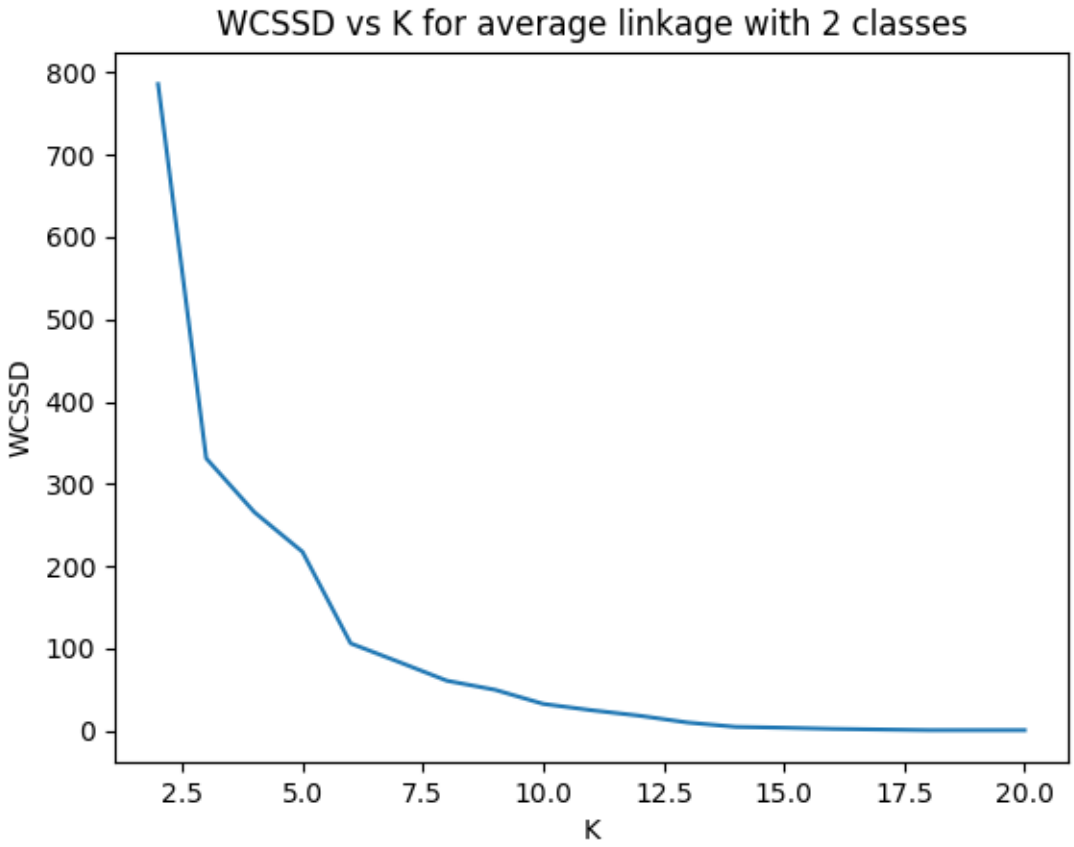

Figure 50: WCSSD vs K for average linkage with 2 classes

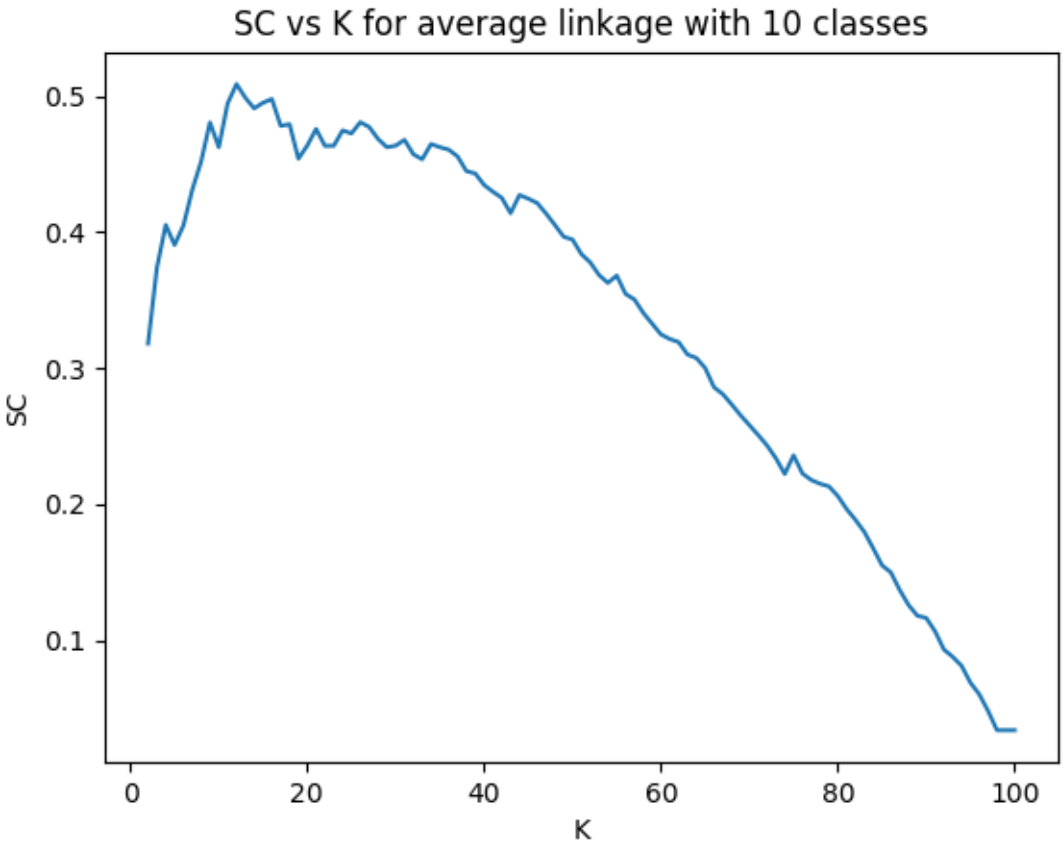

Figure 51: SC vs K for average linkage with 10 classes

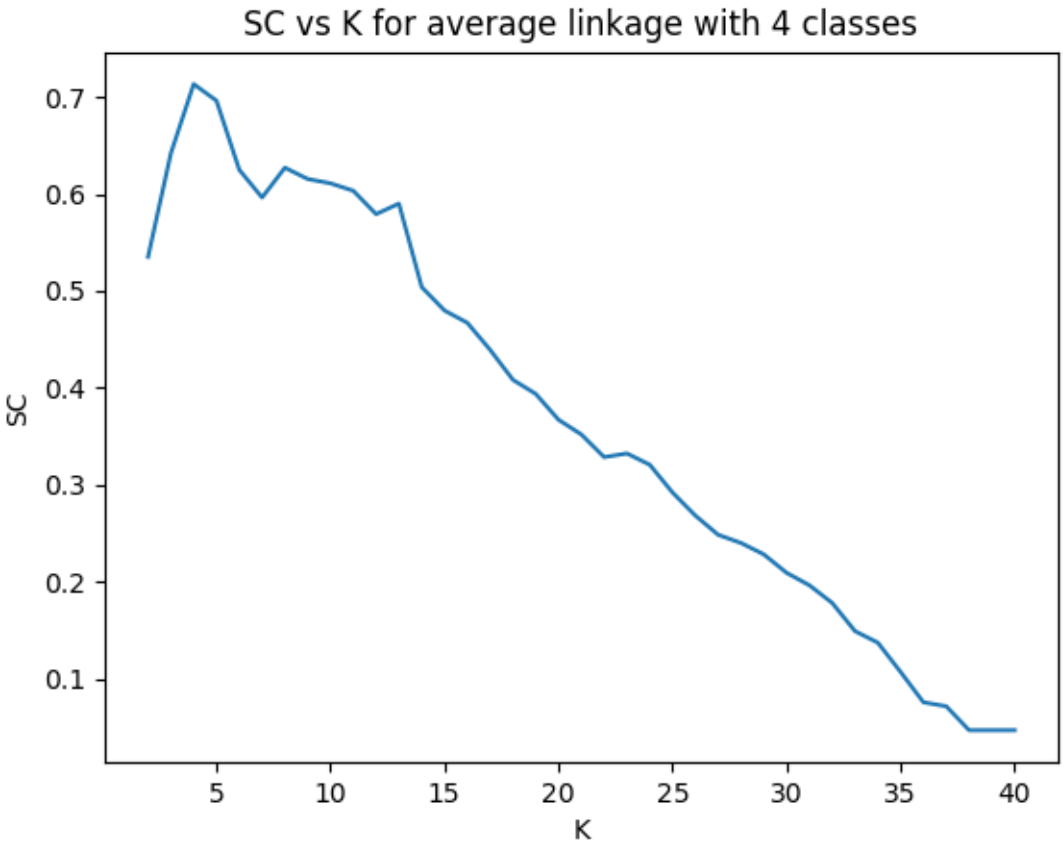

Figure 52: SC vs K for average linkage with 4 classes

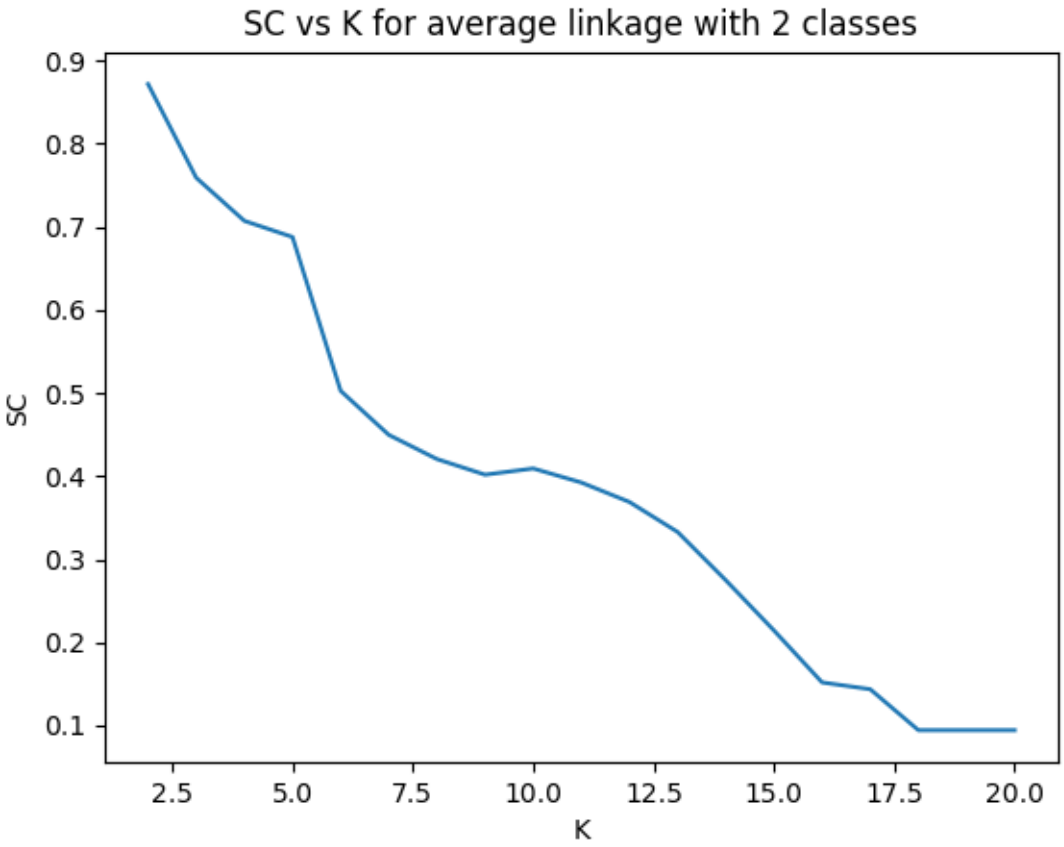

Figure 53: SC vs K for average linkage with 2 classes

For 2 classes with single linkage, the best K is 2 with SC 0.87.
For 2 classes with complete linkage, the best K is 2 with SC 0.87.
For 2 classes with average linkage, the best K is 2 with SC 0.87.

Comparing this to section B.1: For 2 and 4 classes dataset, the choice of K is same. For 10 classes dataset, the choice of K is 10 for complete linkage, 12 for average linkage and 40 for single linkage. Whereas in section B.1, the chosen K was 8. K=10 and 12 seem to be good choices as well as they are close to the number of classes (10). For single linkage, as can be seen from the graph, the SC is comparable for K=10 to K=40, the minor difference can be attributed to random selection of 100 images.

5. NMI Score

NMI for single linkage with 10 classes, K=40 is 0.385.
NMI for complete linkage with 10 classes, K=10 is 0.403
NMI for average linkage with 10 classes, K=12 0.402
NMI for single linkage with 4 classes 0.453
NMI for complete linkage with 4 classes 0.453
NMI for average linkage with 4 classes 0.453
NMI for single linkage with 2 classes 0.5
NMI for complete linkage with 2 classes 0.5
NMI for average linkage with 2 classes 0.5
The NMI scores of hierarchical clustering are higher than those of K-means for single, complete and average linkage across all the datasets. The NMI scores for 2-classes and 4-classes are similar/comparable for complete, single and average linkage. Also, for 2 classes, we achieve a perfect NMI score of 0.5 for all three different types of linkages. The NMI scores for 10 classes for single linkage are marginally lower than those of average and complete linkage, wherease those of average and complete linkage are comparable.

Bonus

2. Eigenvectors

Figures 54-63 show the eigenvectors corresponding to the top 10 principal components, as 28 x 28 grayscale matrices. (For dataset with 10 classes)

3. Visualization of class labels of 1000 randomly selected points

Figure 64 shows the visualization of class labels of 1000 randomly selected examples using the first two principle components.
Compared to the tSNE embedding, the clusters formed by the first two principle components are less clearly seperated and less compact and there is more overlap between the clusters. tSNE embedding has clusters which are representative of their class labels, whereas since it's difficult to visually cluster points as per classes for the PCA case, this implies the mapping between cluster and class labels for PCA is not relatively good.

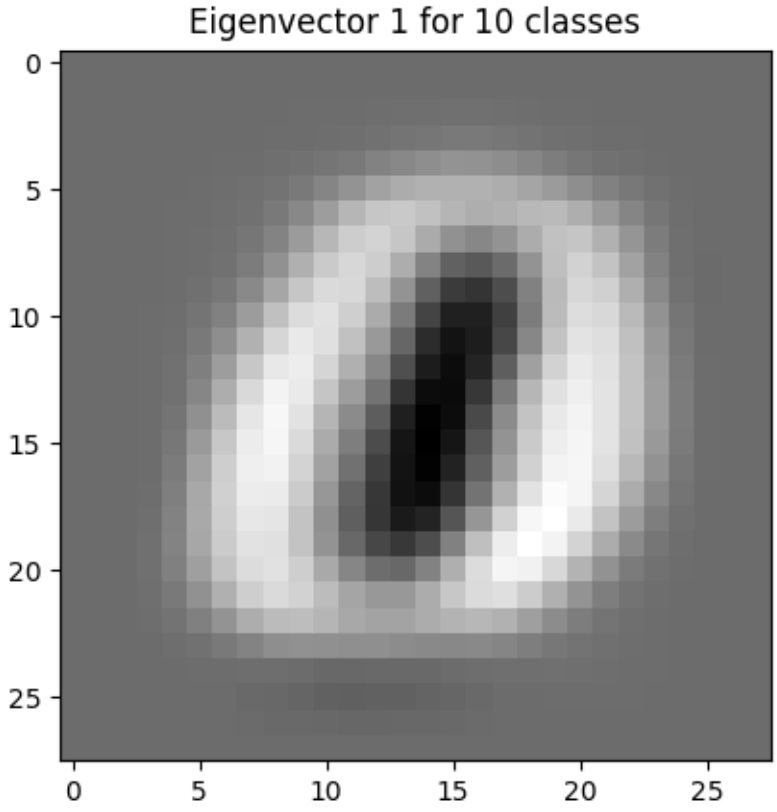

Figure 54: Eigenvector 1

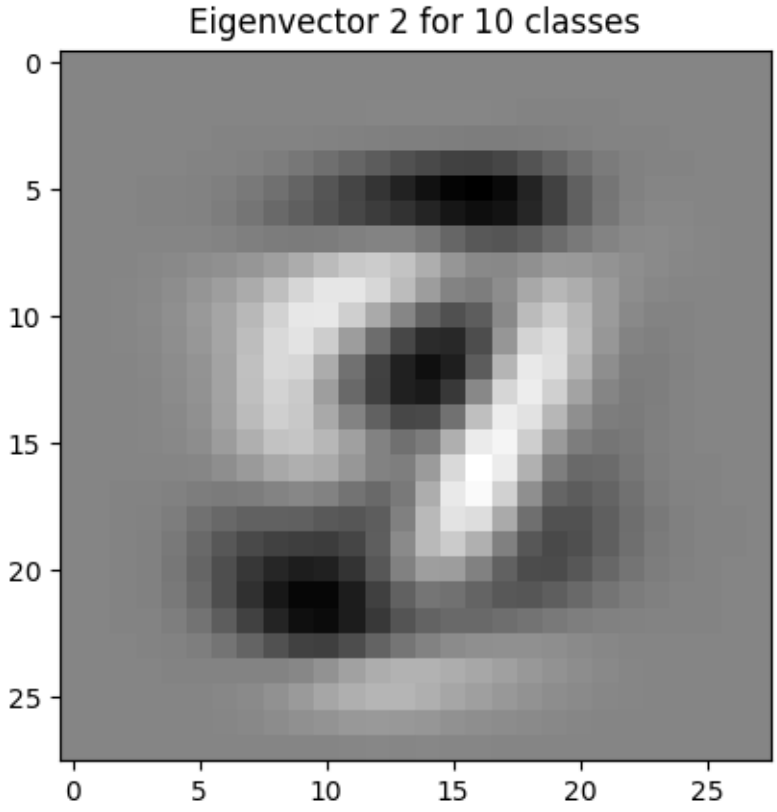

Figure 55: Eigenvector 2

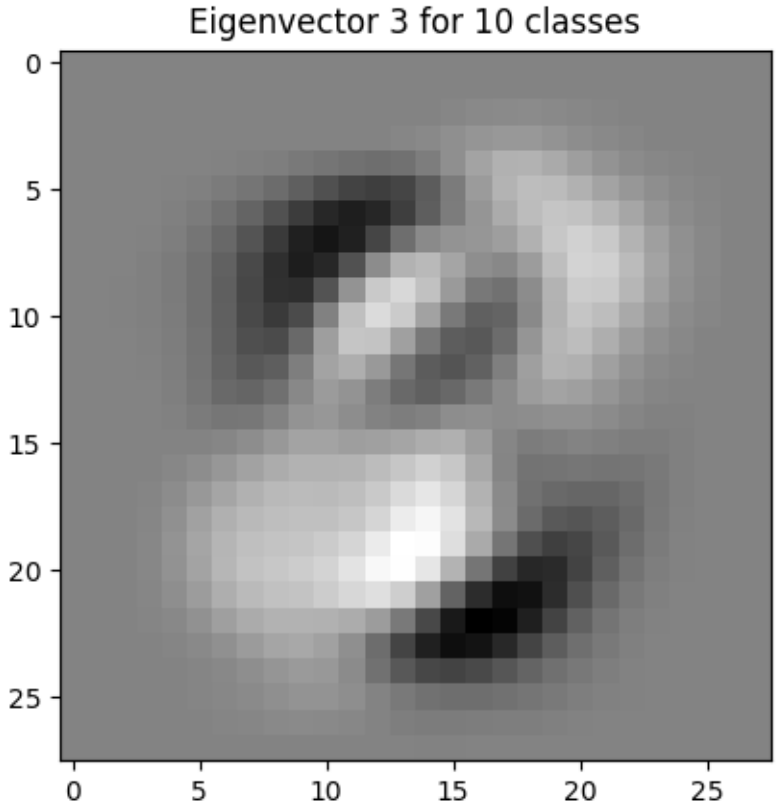

Figure 56: Eigenvector 3

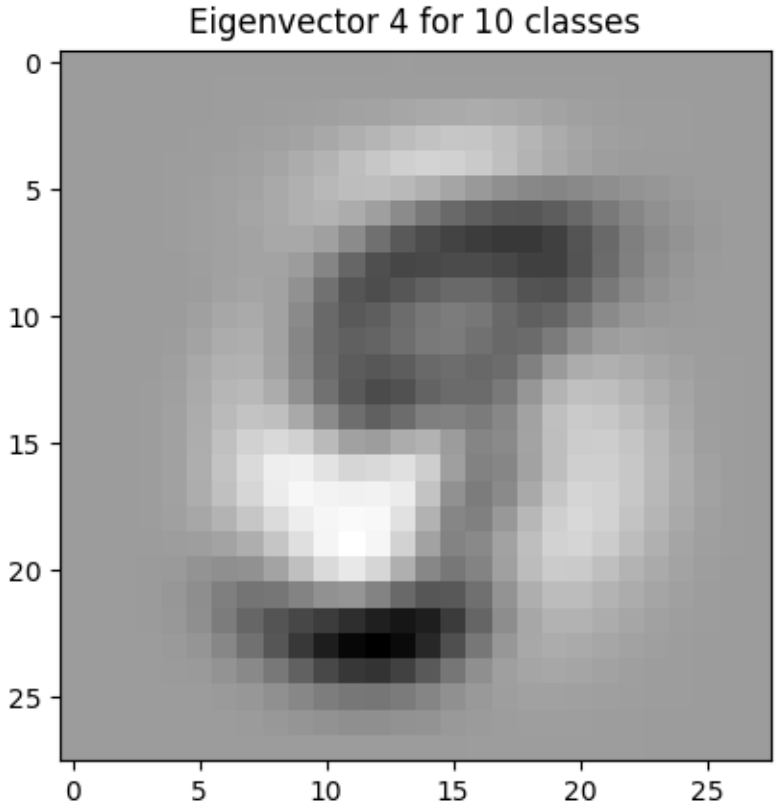

Figure 57: Eigenvector 4

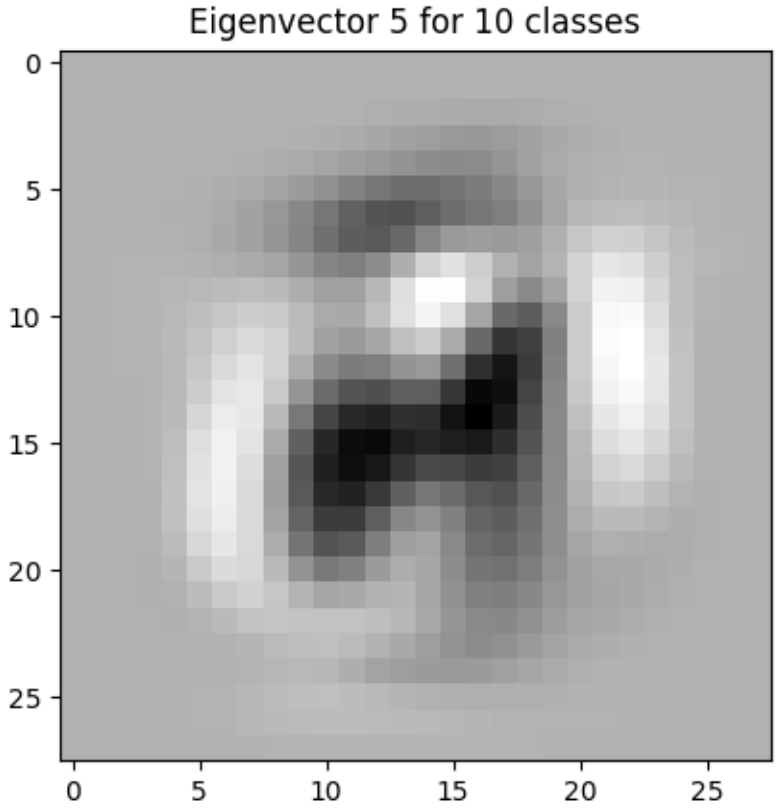

Figure 58: Eigenvector 5

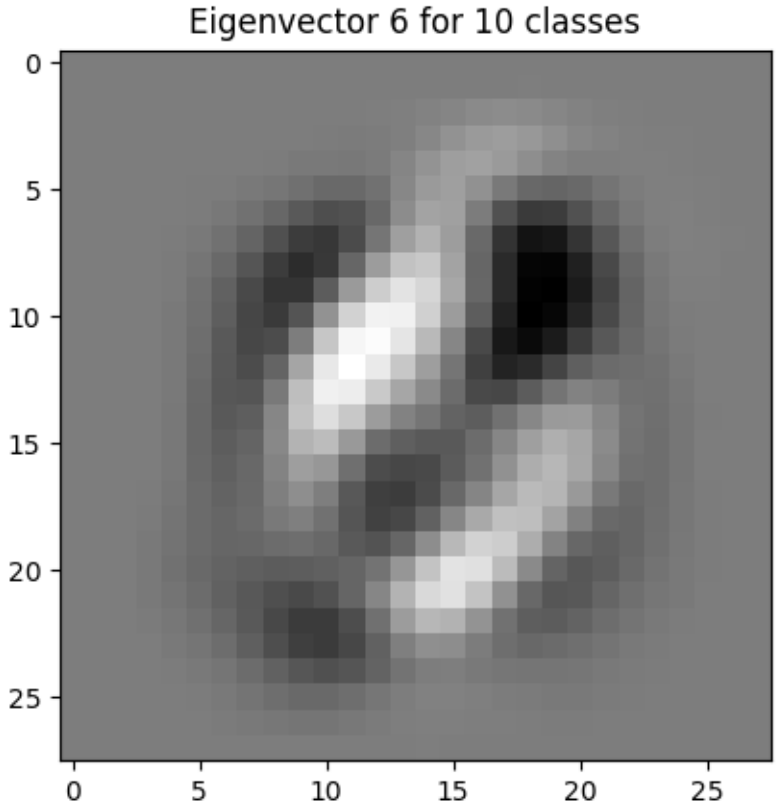

Figure 59: Eigenvector 6

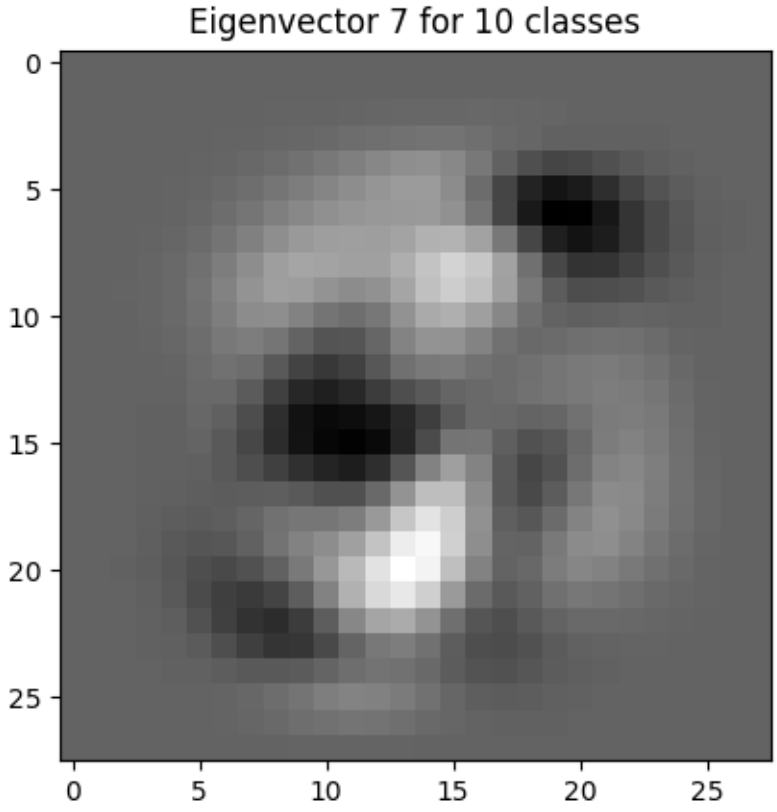

Figure 60: Eigenvector 7

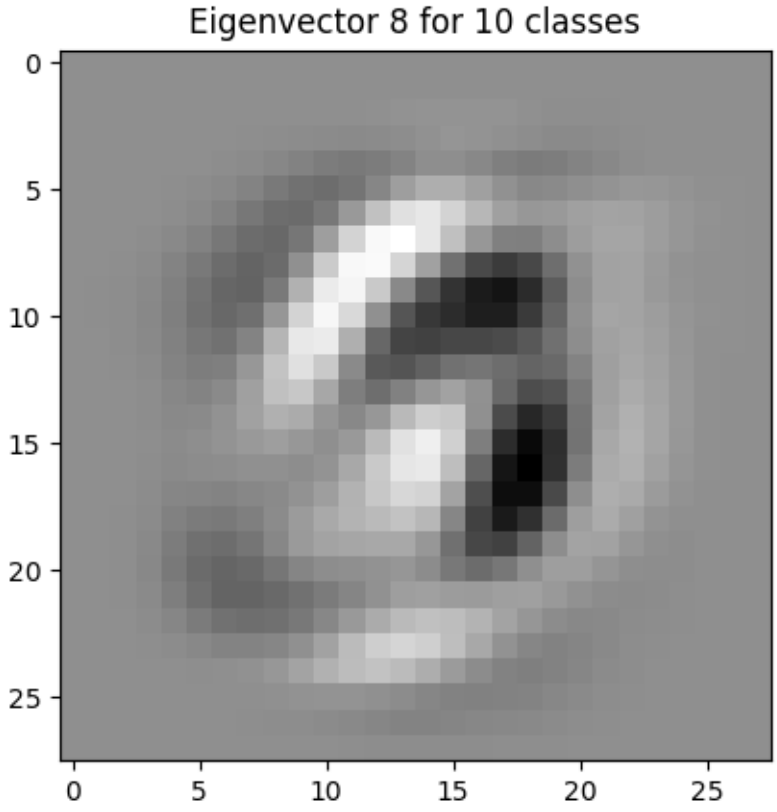

Figure 61: Eigenvector 8

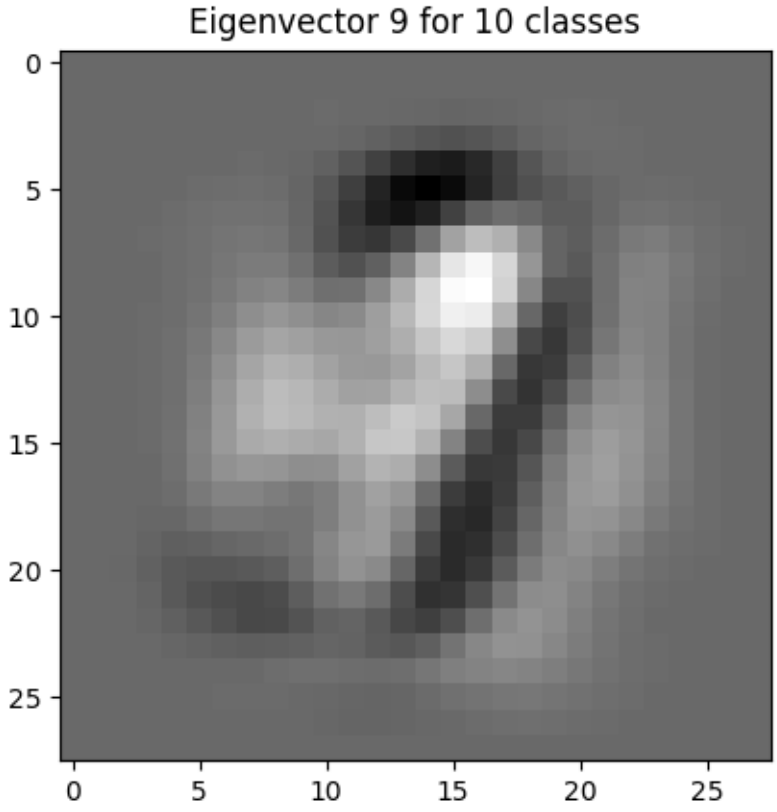

Figure 62: Eigenvector 9

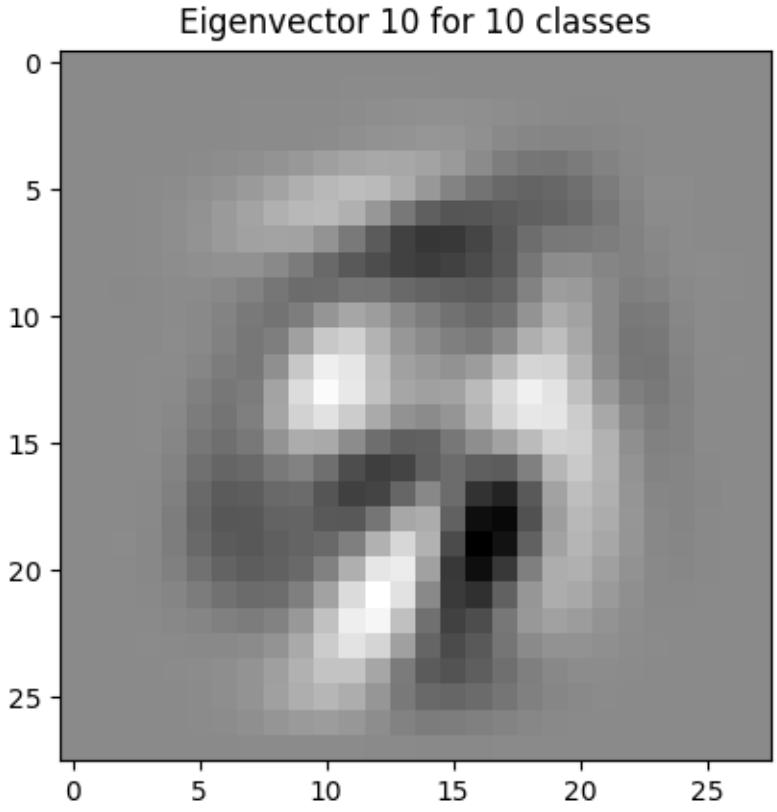

Figure 63: Eigenvector 10

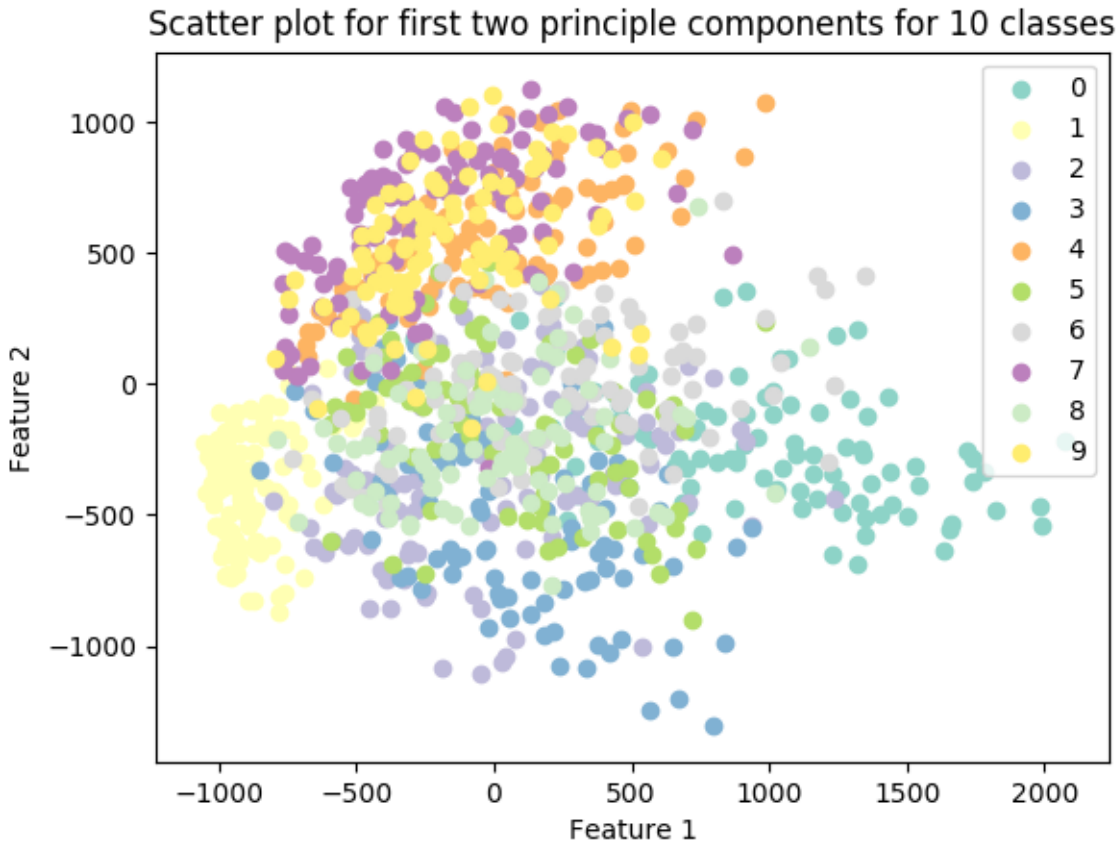

Figure 64: Scatter plot for first two principle components using 10 classes dataset

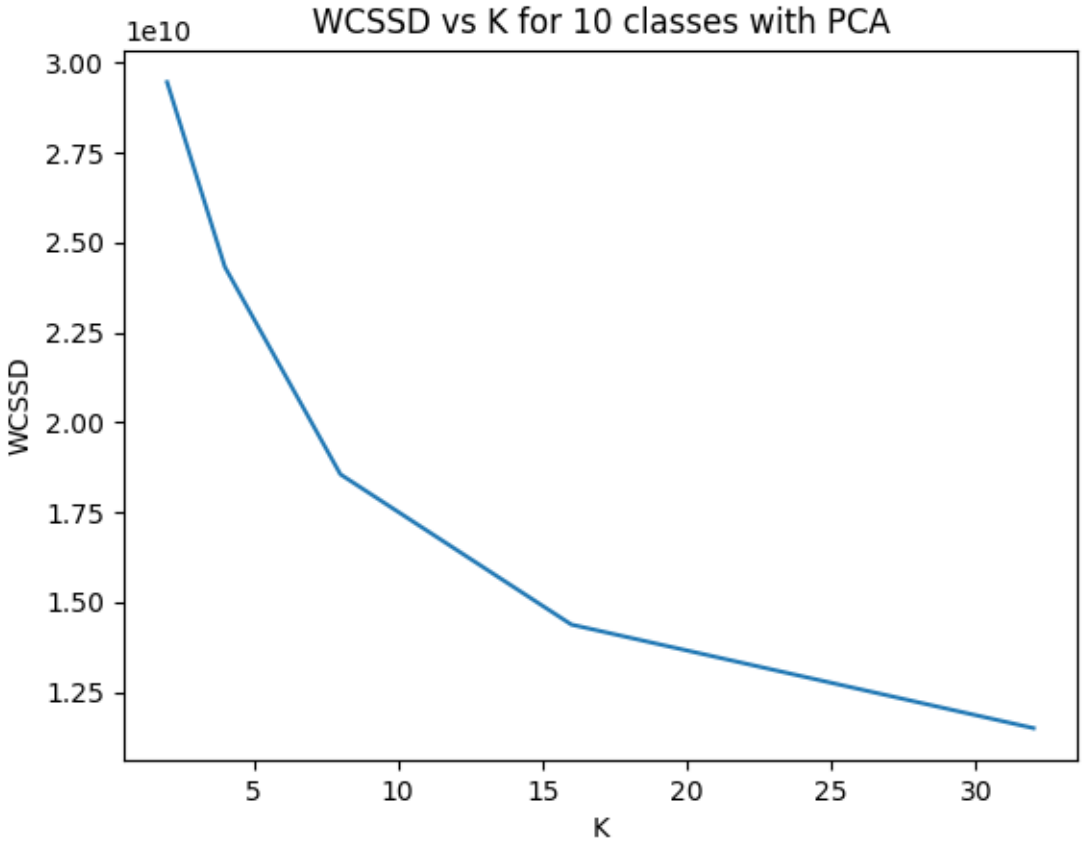

Figure 65: WCSSD vs K for 10 classes with PCA

4. and 5. Using PCA embedding for clustering for three different datasets

Figures 65-67 show the within-cluster sum of squared distances (WCSSD) as a function of K for the three different datasets using PCA.

Figures 68-70 show the silhouette coefficient (SC) as a function of K for the three different datasets using PCA.

For dataset with 10 classes, since WCSSD keeps decreasing with increasing value of K and SC achieves it maximum at K=8 and then drops off. On either side of K=8, SC has a lower value.

For dataset with 4 classes, WCSSD keeps decreasing with increasing value of K and SC achieves it maximum at K=4 and then drops off. Judging by the slope of the plot around K=4, we see that SC would decrease on either side of 4, so I select K=4 to be the best K.

For dataset with 2 classes, WCSSD keeps decreasing with increasing value of K and SC too keeps decreasing with increasing value of K. Since the maximum occurs at K=2 for SC, I select K=2 as the best value of K.

I am basing my decisions on SC only, since WCSSD would always decreases with increase in K as the number of points in a cluster drops as K increases. A higher value of SC indicates

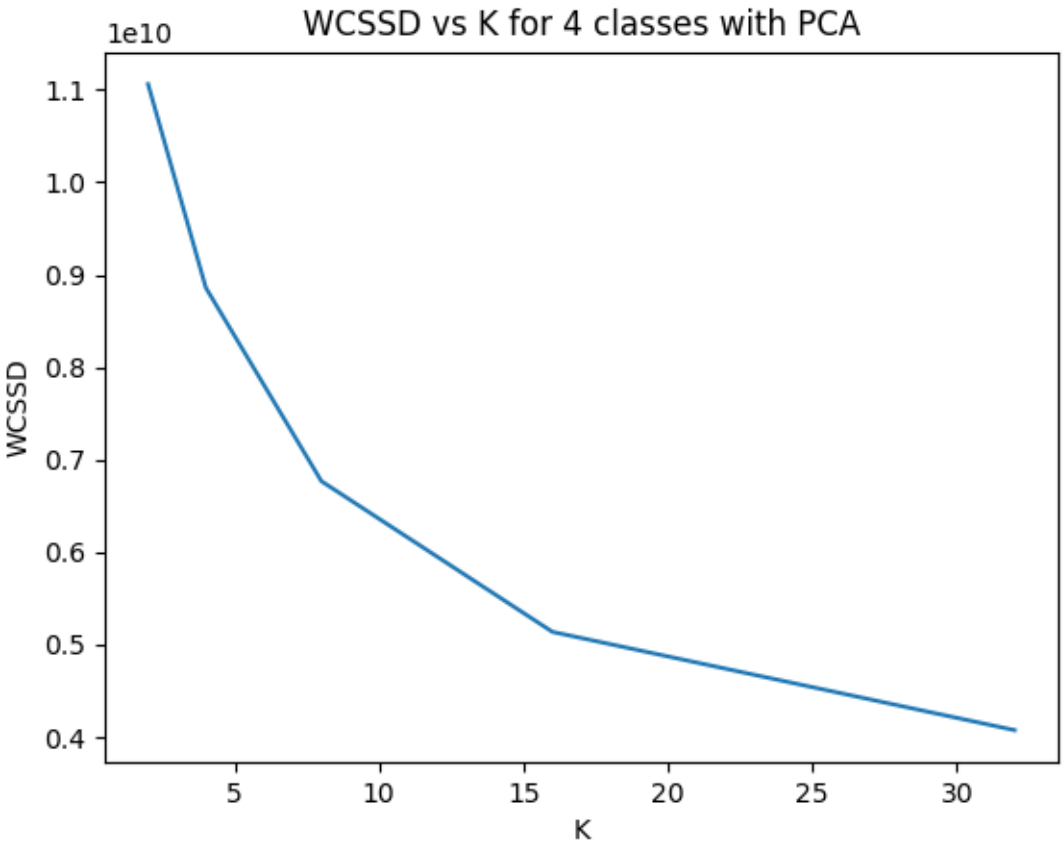

Figure 66: WCSSD vs K for 4 classes with PCA

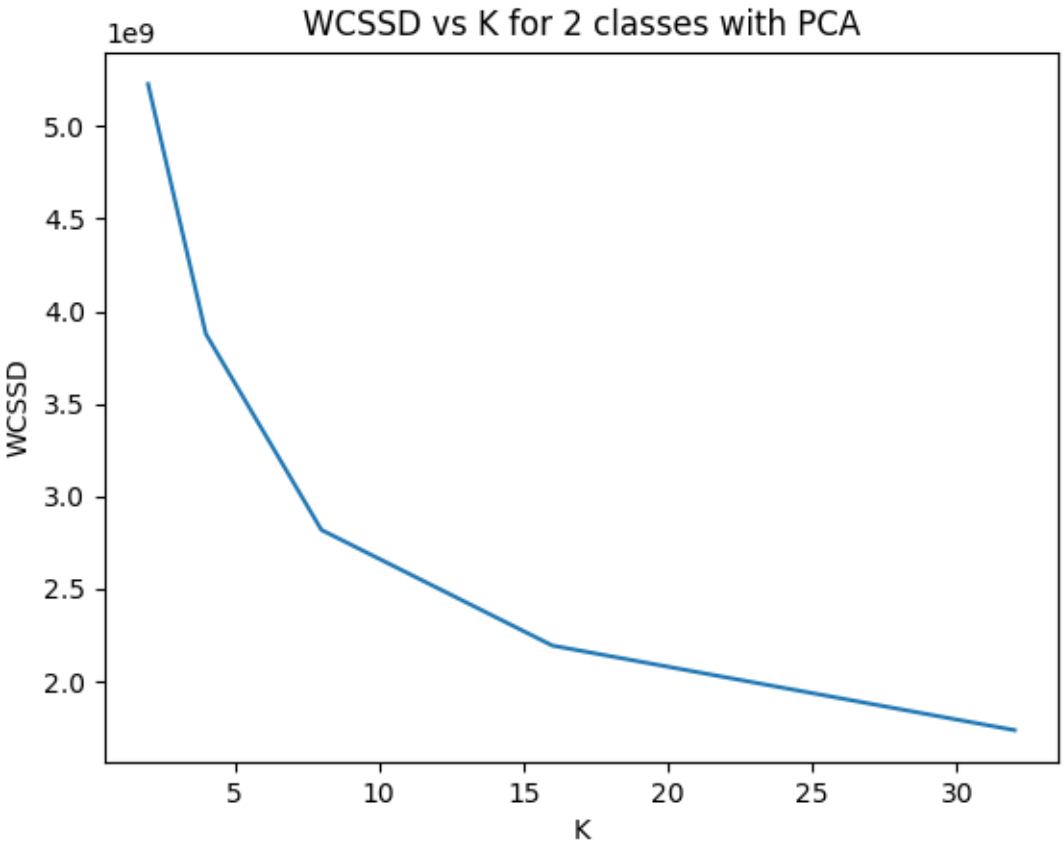

Figure 67: WCSSD vs K for 2 classes with PCA

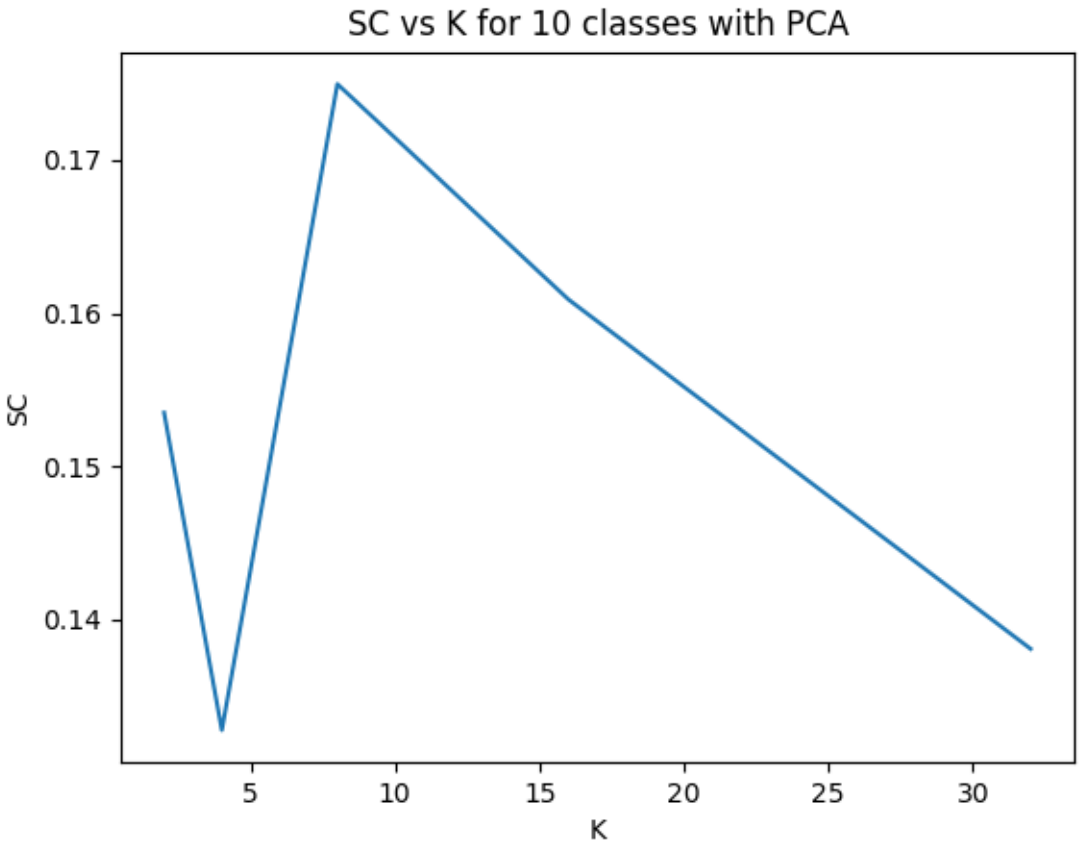

Figure 68: SC vs K for 10 classes with PCA

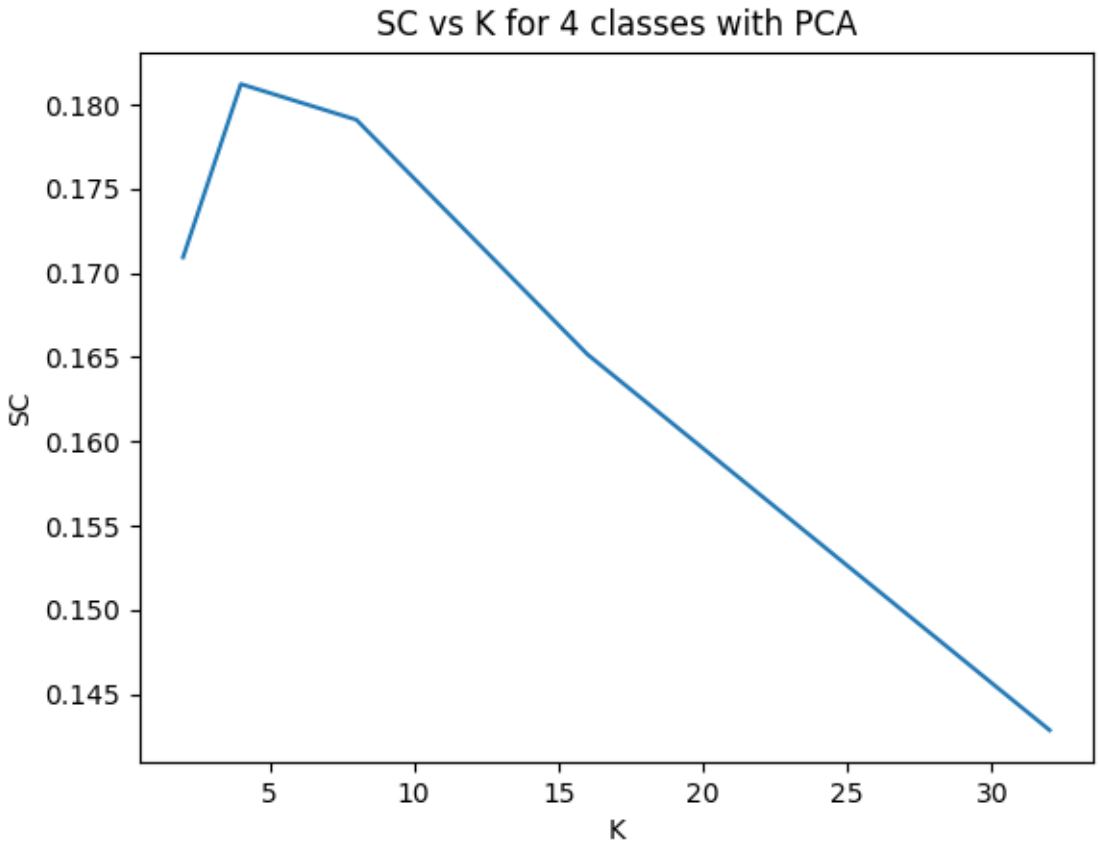

Figure 69: SC vs K for 4 classes with PCA

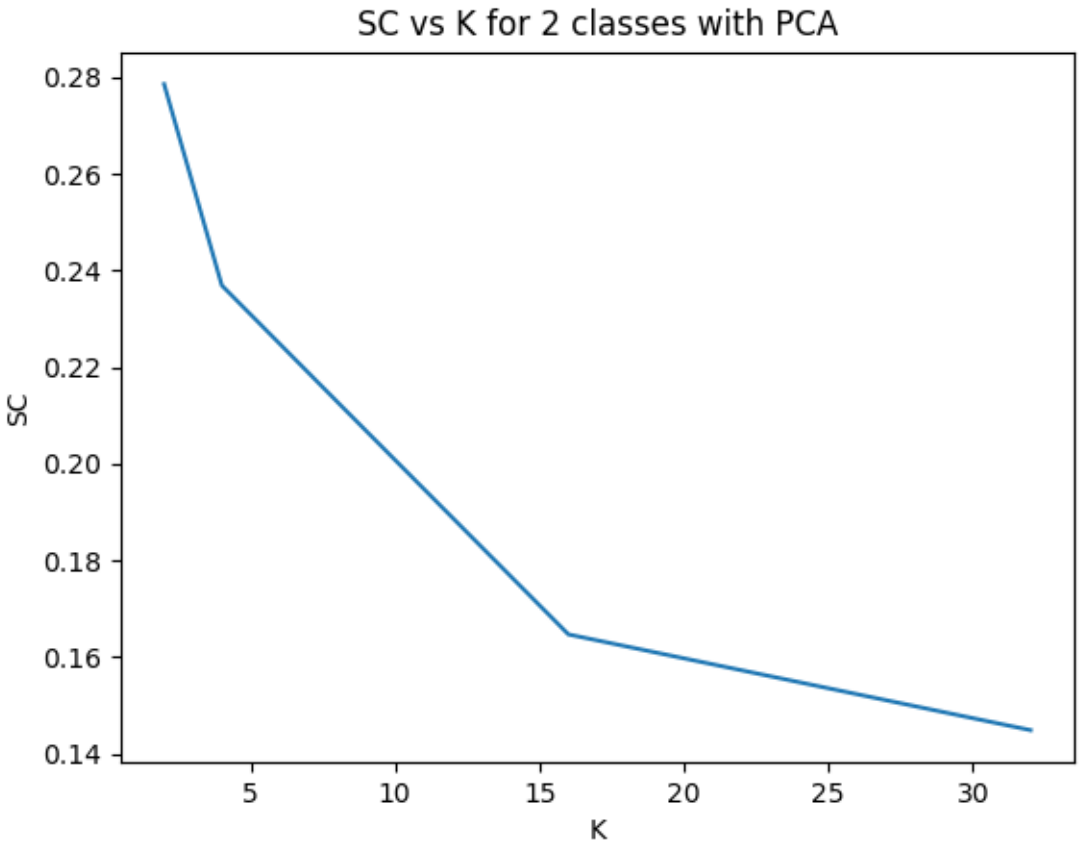

Figure 70: SC vs K for 2 classes with PCA

that all examples on average are well matched to their respective cluster.

How the results compare: For all the three datasets WCSSD keeps decreasing with increase in value of K. For dataset with 2 classes SC keeps decreasing with increasing in K. For dataset with 4 classes, SC increases till K=4 and then drops off. For dataset with 10 classes, SC drops for K=4 as compared to K=2, then increases till K=16 and then drops off.

How the results compare with tSNE embedding: For dataset with 2 classes, the nature of the plot is similar with PCA versus with tSNE, since SC decreases with increase in K. However SC values are on average better with tSNE as compared to the SC values with PCA. This tells us that we are able to get better clustering with tSNE embedding. The SC values for tSNE start with approximately 0.8 for K=2 and then drop to approximately 0.4 for K=32. The SC values for PCA start with approximately 0.28 for K=2 and drop to approximately 0.14 for K=32.

For dataset with 4 classes, the natures of the plot of SC vs K with tSNE and of SC vs K with PCA are similar, since they increase from K=2 till a maximum and then drop till K=32. For tSNE embedding, the SC values are higher on average than those with PCA. For tSNE embedding, SC values increase from approximately 0.45 for K=2 to approximately 0.7 for K=4 and then keeps decreasing to approximately 0.37 for K=32. For PCA, SC values increase from approximately 0.17 for K=2 to approximately 0.182 for K=4 and then keeps decreasing to approximately 0.143 for K=32. This tells us that again we have well seperated and compact clusters with tSNE embedding.

For dataset with 10 classes, SC values with tSNE increase from K=2 (approximately 0.375) till maximum SC at K=8 (approximately 0.415) and then keep decreasing till K=32 (approximately 0.39). For PCA, SC values decrease from approximately 0.154 for K=2 to approximately 0.132 for K=4, then increases to approximately 0.175 for K=8 and then keeps decreasing to approximately 0.14 for K=32. This tells us that again we have relatively well behaved clusters with tSNE embedding, but there is more noise and relatively less well seperate/compact clusters which can be attributed to the fact that there are 10 classes with overlaps.

For WCSSD vs K plots, the nature of the plots are same for both tSNE embedding and PCA across all the three different datasets, since WCSSD decreases with increase in K. However, PCA clusters have more WCSSD overall compared to clusters with tSNE embedding. For 10 classes, WCSSD is of the order of 10^{10} with PCA, whereas they are of the order of 10^6 with tSNE embedding. For 4 classes, WCSSD scores are of the order of 10^9 to 10^{10} with PCA, whereas they are of the order of 10^6 with tSNE embedding. For 2 classes, WCSSD scores are of the order of 10^9 with PCA, whereas they are of the order of 10^5 with tSNE embedding. This shows that the clusters with PCA are less compact compared to the clusters formed with tSNE embedding.
For PCA, NMI for 10 classes with K=8 is 0.22
NMI for 4 classes with K=4 is 0.34
NMI for 2 classes with K=2 is 0.46
We see that as the number of classes decreases, the NMI score increases. This explains that as the number of classes decreases, the cluster labels are more in agreement with the class labels.

Comparison to tSNE embedding: We see that the NMI scores with PCA are less than those with tSNE embedding across all the three datasets. This can be attributed to the fact that tSNE embedding is able to capture more variance in the original data as compared to PCA and hence tSNE has clusters which are more in alignment with the class labels. I used the formula

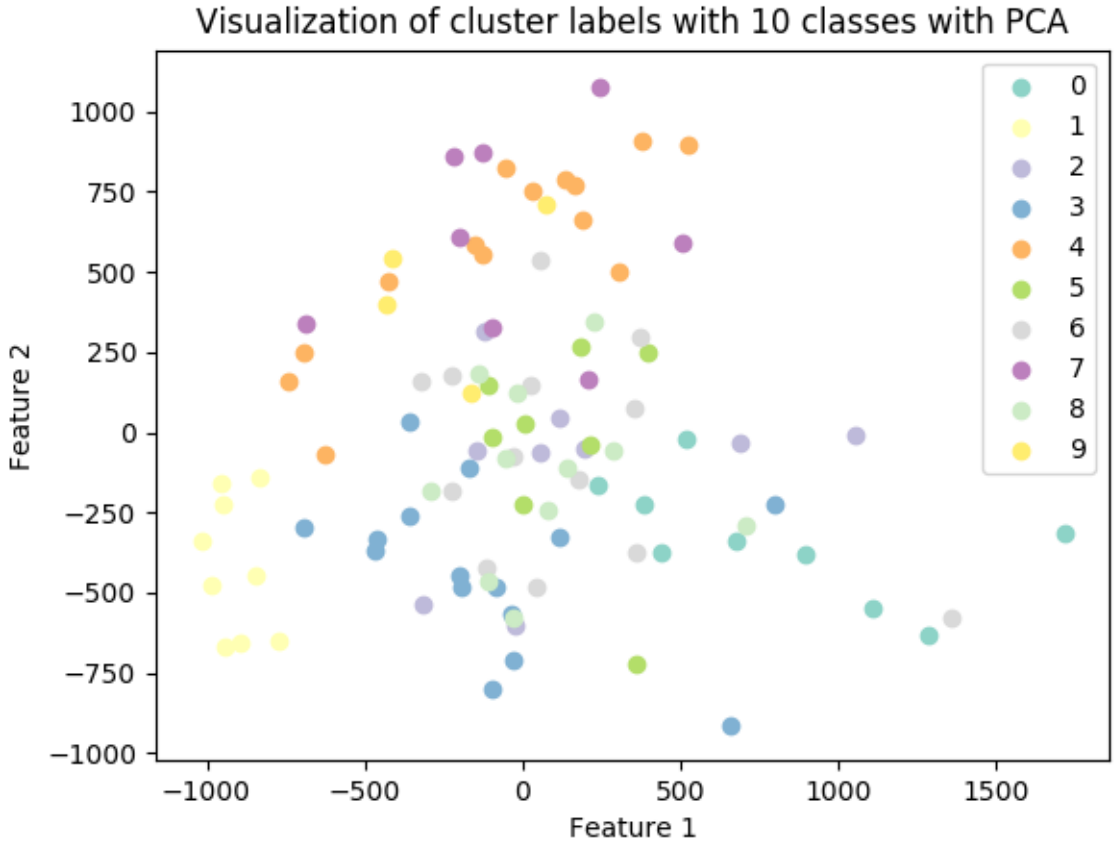

Figure 71: Visualization of cluster labels with 10 classes with PCA

given in the slides for NMI computation.
Figures 71-73 show the visualization of cluster labels of 1000 randomly selected examples in 2D. From the plots, it is evident that the data of 10 classes is too much crowded, and the clusters are less compact and not well seperated. The clusters are much crowded when compared to the clusters formed with tSNE embedding. The data of 4 classes with PCA are better seperated, there are however considerable number of violations. However, the clusters are more spread out, less compact and less well seperated when compared to the clusters formed with tSNE embedding. The data of 2 classes is best seperated as the two clusters are far away and the number of clusters is less. Again, compared to the clusters formed with tSNE embedding, the clusters are more spread out and less compact.

Code

Figure 74 shows the output of the code part when run on the given dataset with $K = 10$

Figure 72: Visualization of cluster labels with 4 classes with PCA

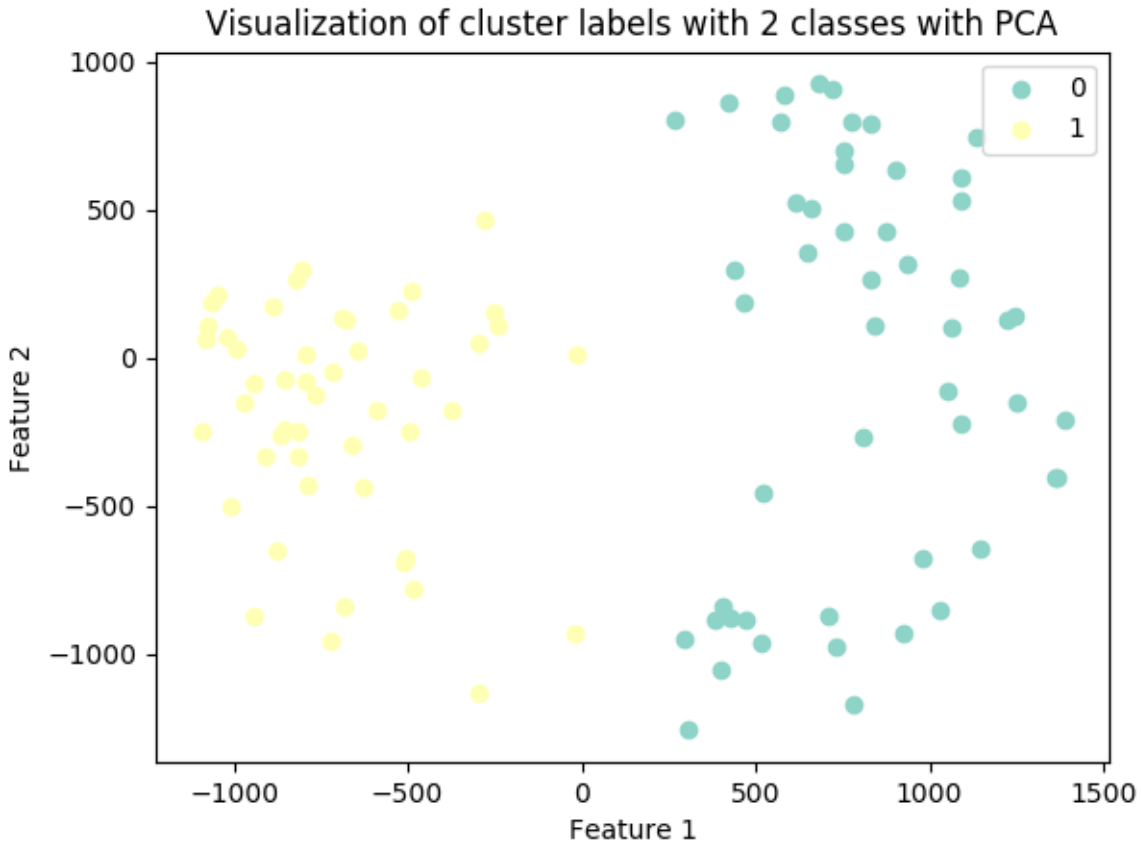

Figure 73: Visualization of cluster labels with 2 classes with PCA

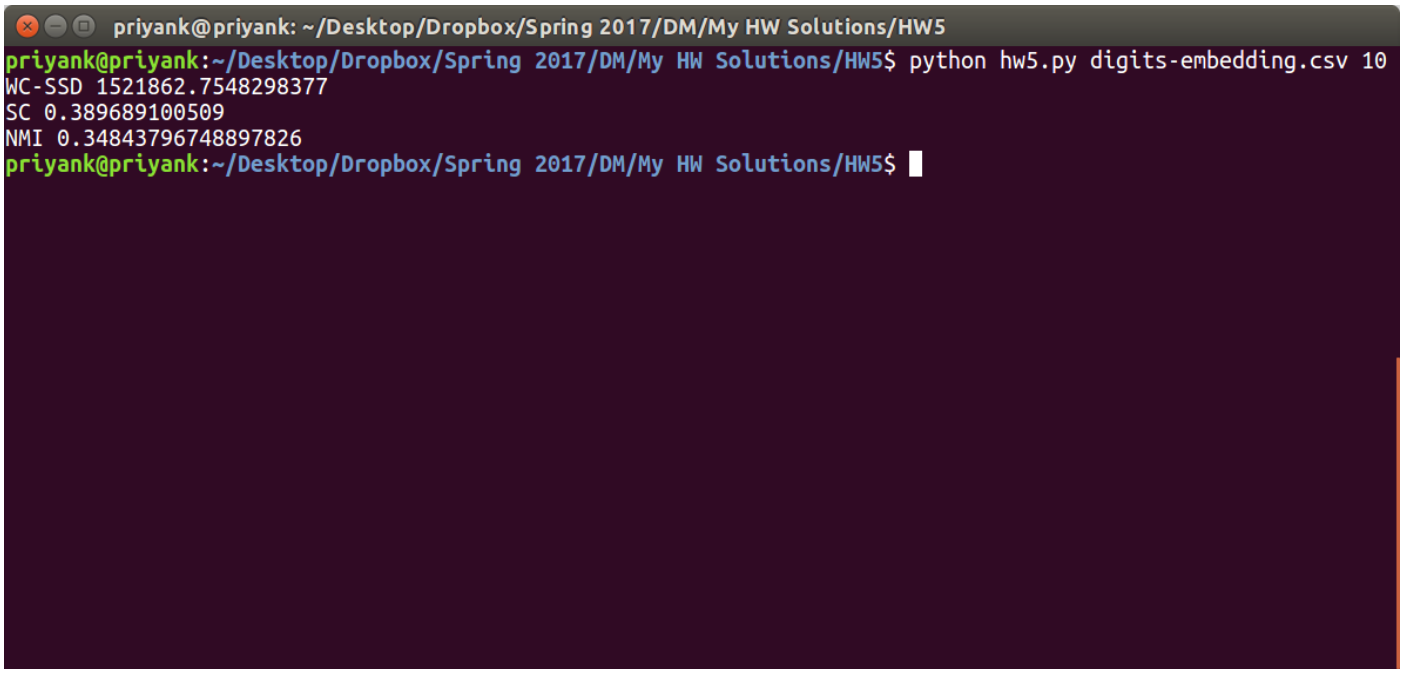

Figure 74: Output of Python script