CS57300 Data Mining
Assignment 5

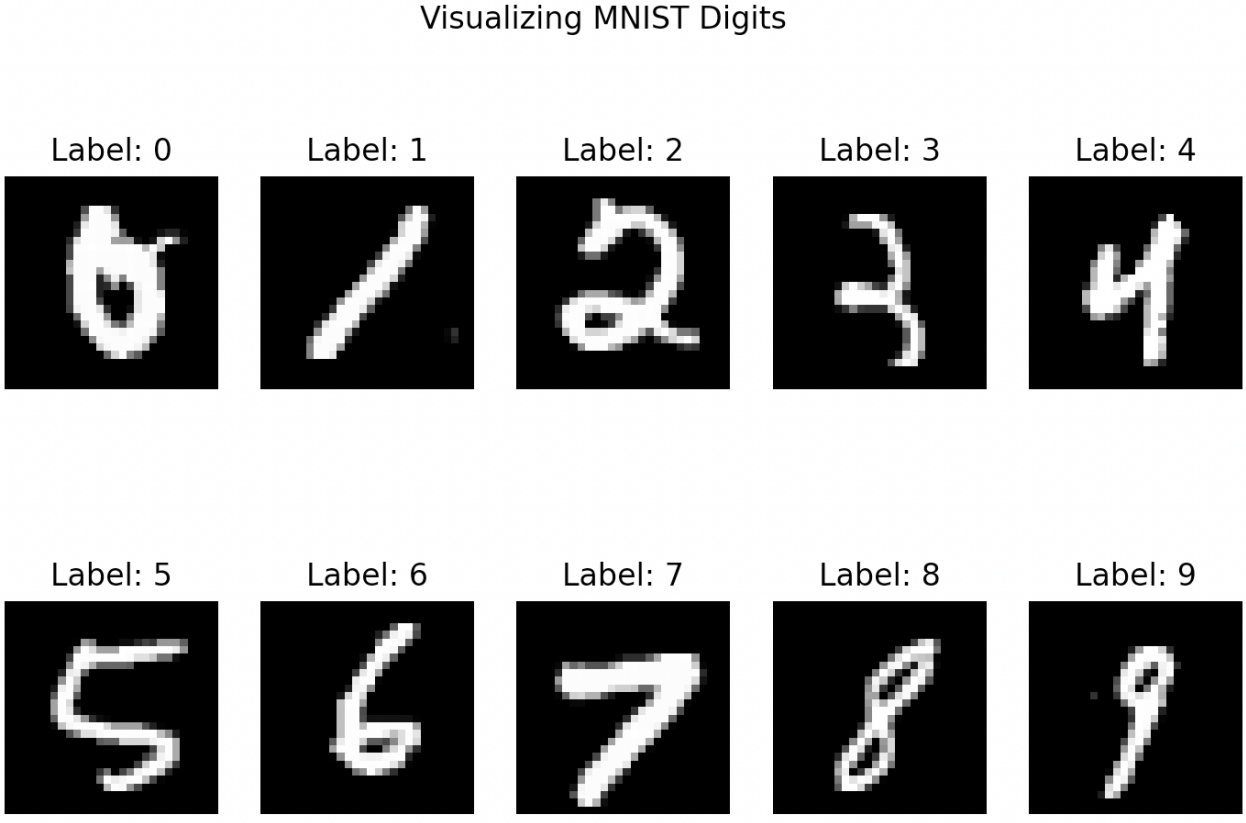
November 30, 2021

Vivek Gupta
gupta690@purdue.edu

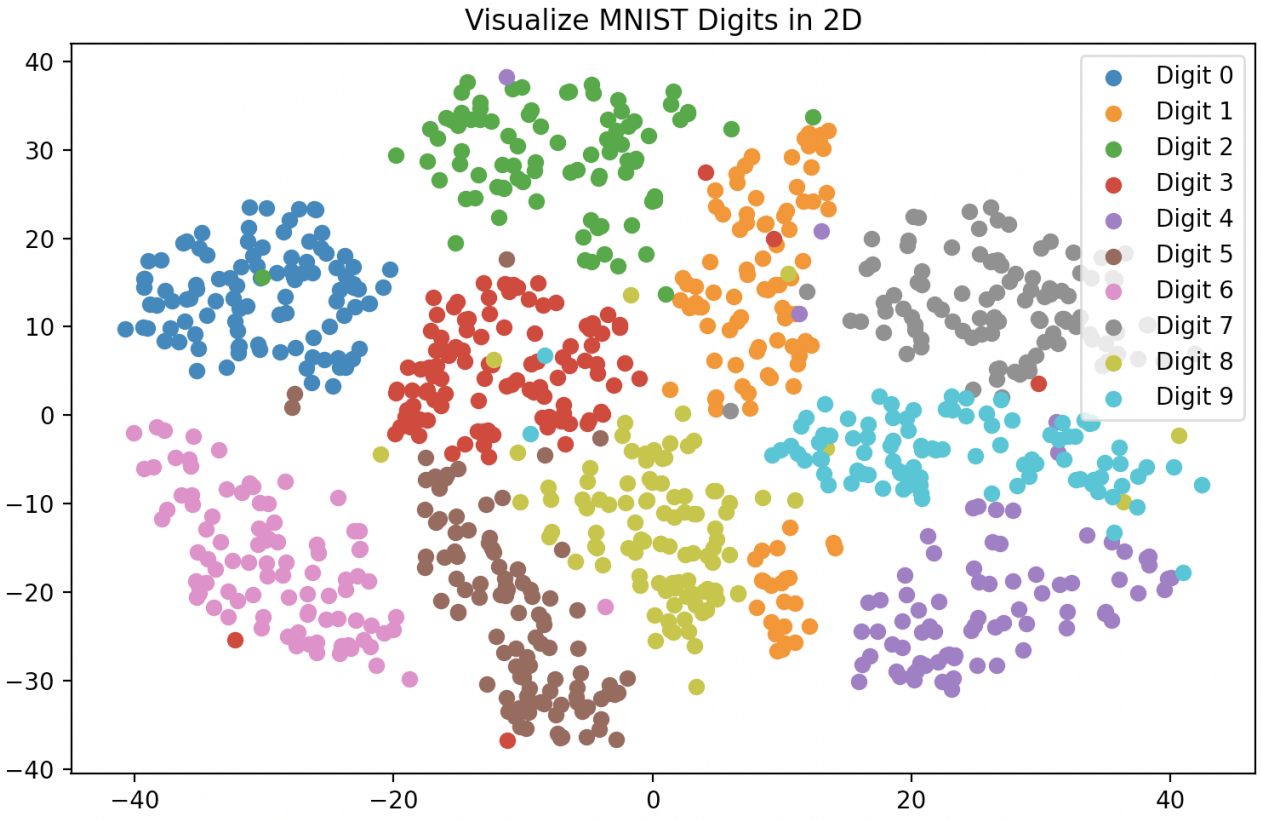
Environment:
Experiment done on Apple Silicon M1.
Python version: **3.9.7**
Numpy version: **1.21.2**
Pandas version: **1.3.3**
Number of Extension Days used: 2

1 Exploration

1.



2.

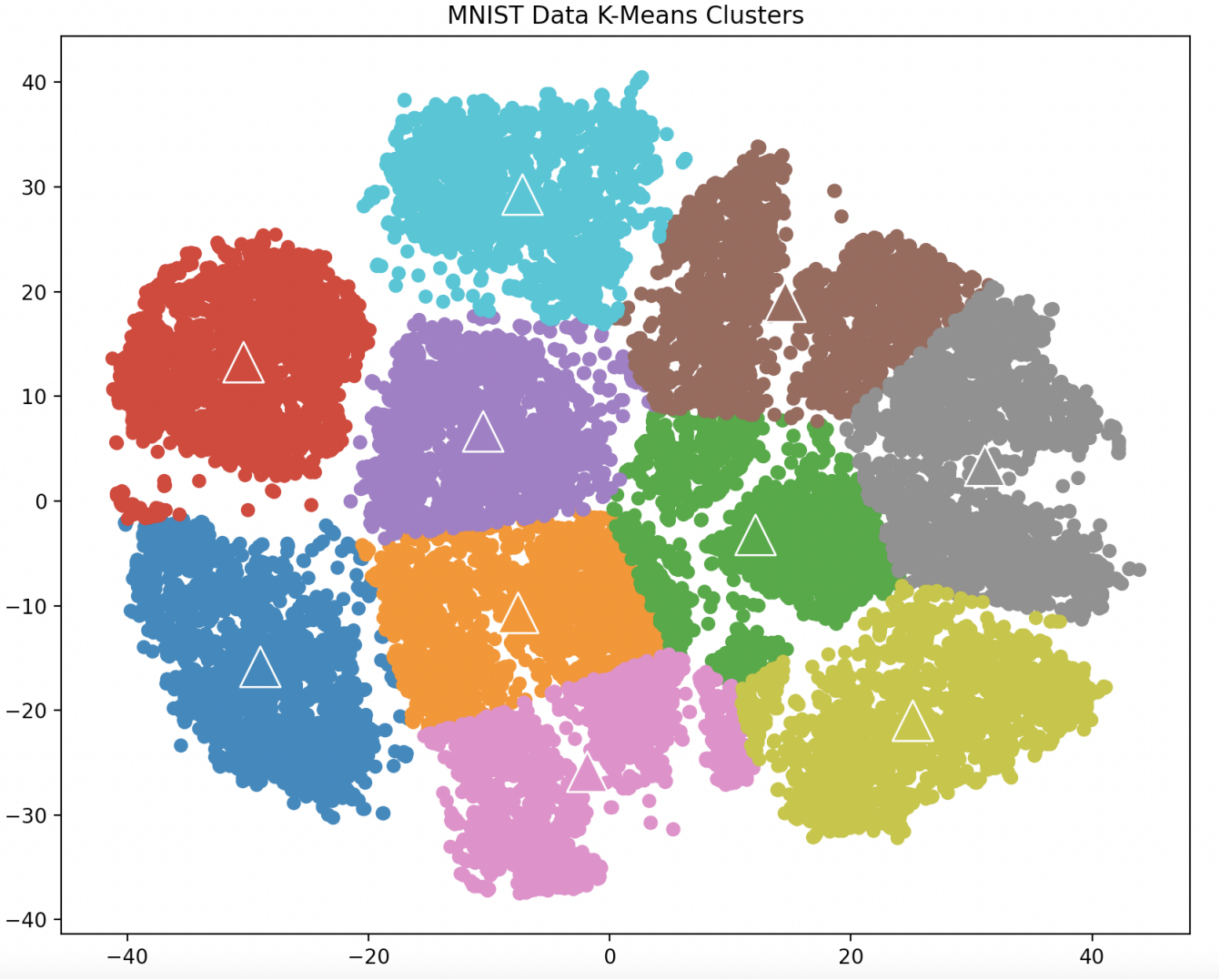

Visualize MNIST Digits in 2D
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
40
30
20
10
0
−10
−20
−30
−40
−40
−20
0
20
40

2 K-Means Clustering

2.1 Code

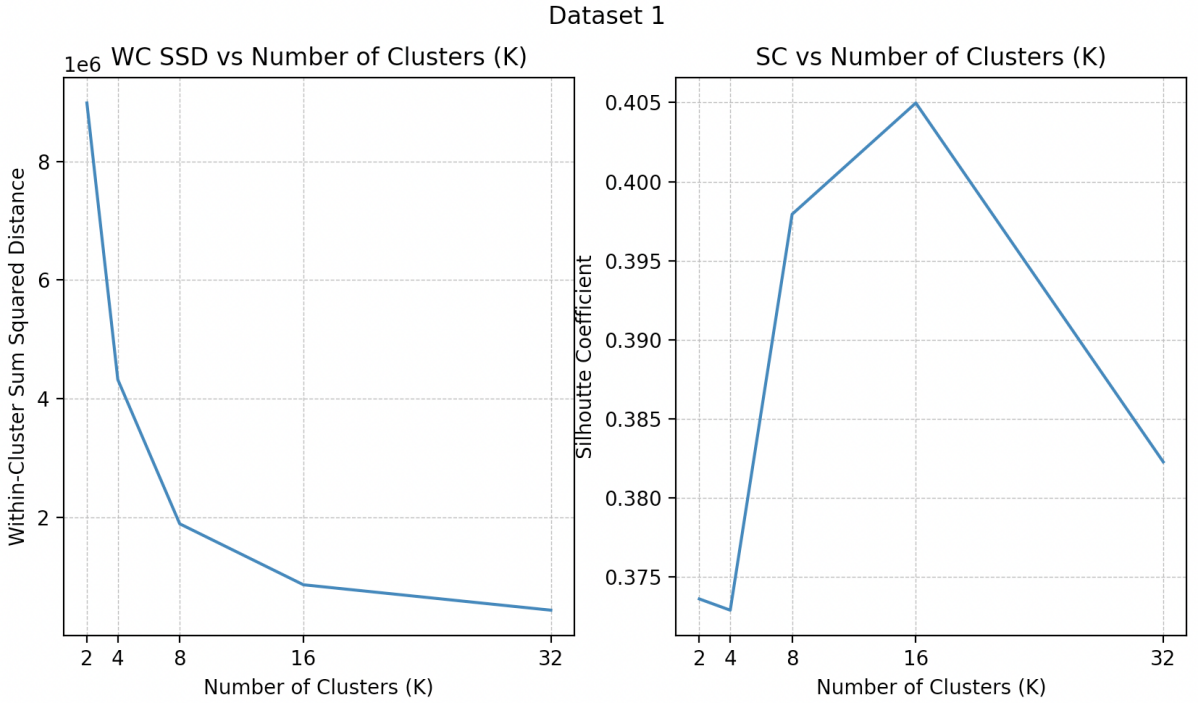
```
(dm) → CS 573 Assignment 5 python kmeans.py digits-embedding.csv 10
WC-SSD: 1489650.532
SC: 0.404
NMI: 0.359
```

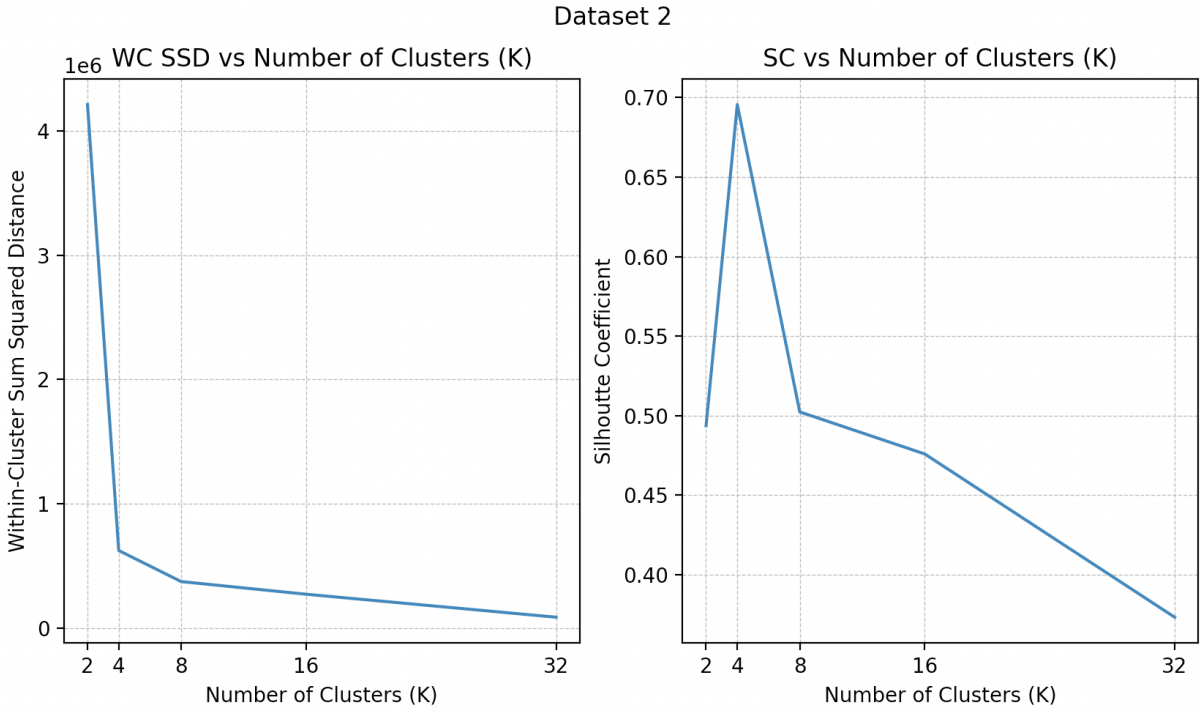
The 10 clusters after 50 iterations of kmeans look like this: Triangles here represent the centroid of the cluster.

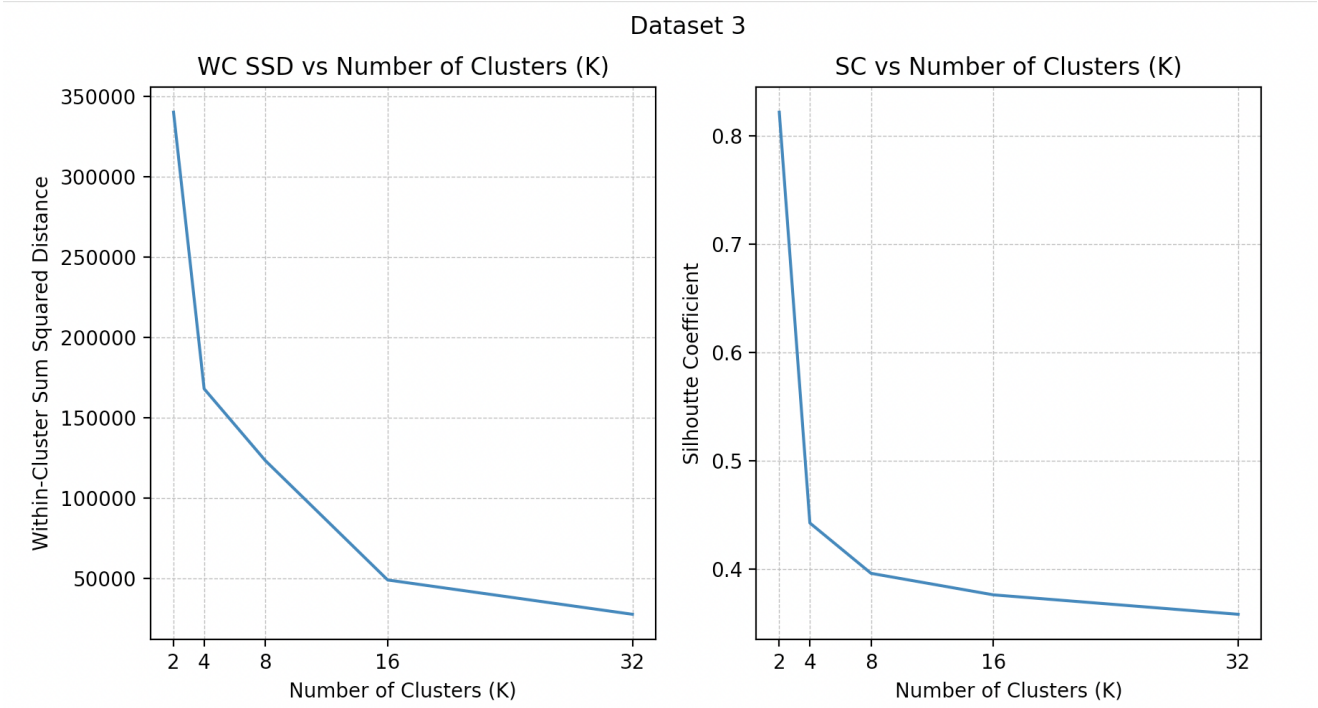


2.2 Analysis

1.





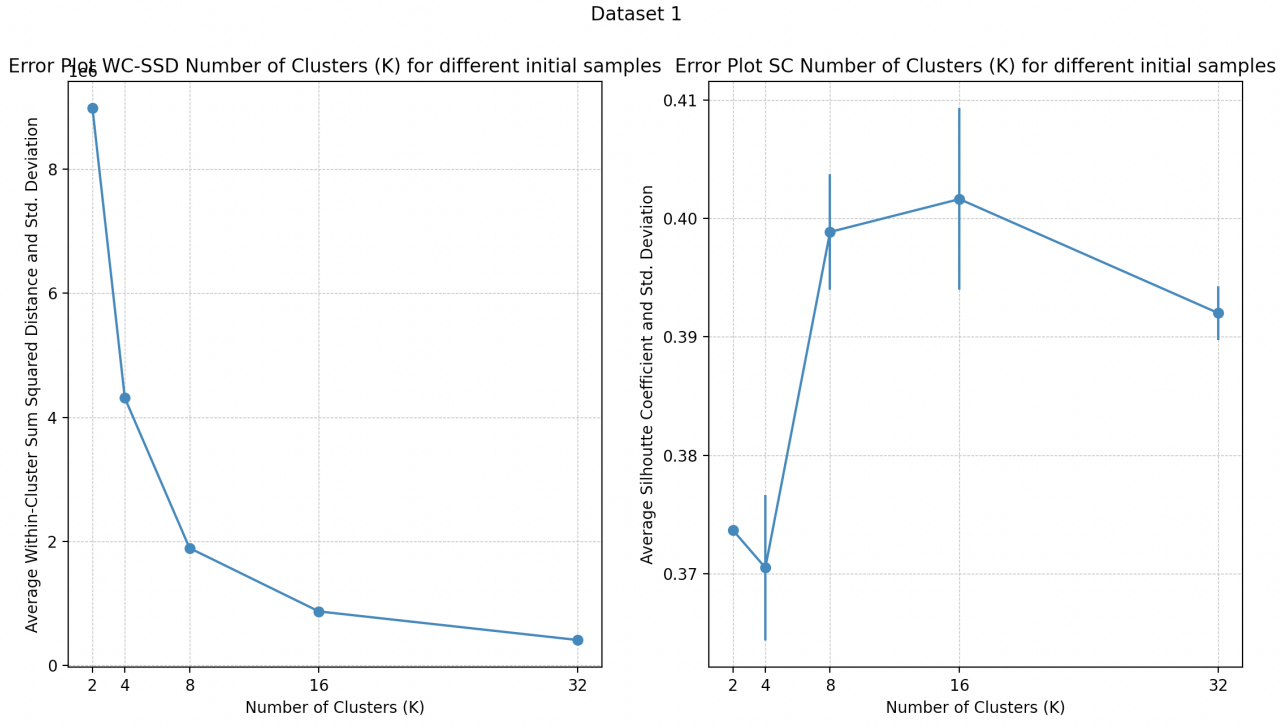


2. While looking for the optimal K, we look for the elbow in the WC-SSD plot and peak in SC plot. The optimal values of K according to the above plots for each of the datasets are:

   (a) Dataset 1: 8

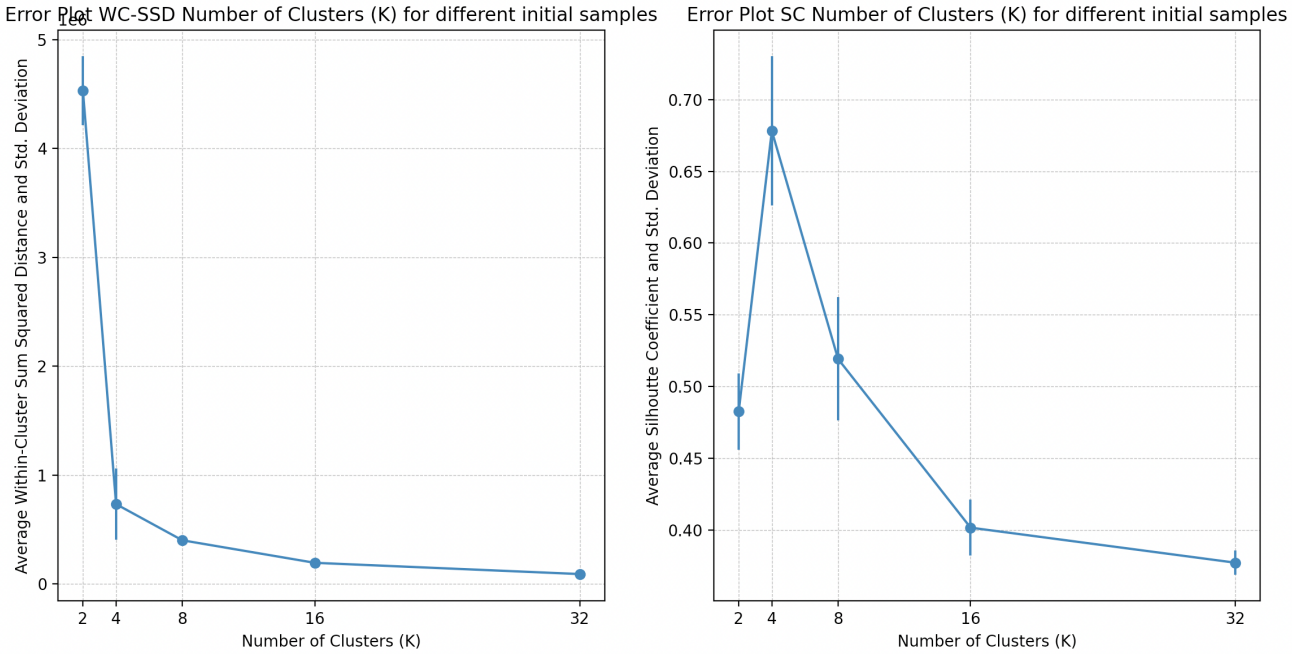
   (b) Dataset 2: 4

   (c) Dataset 3: 2

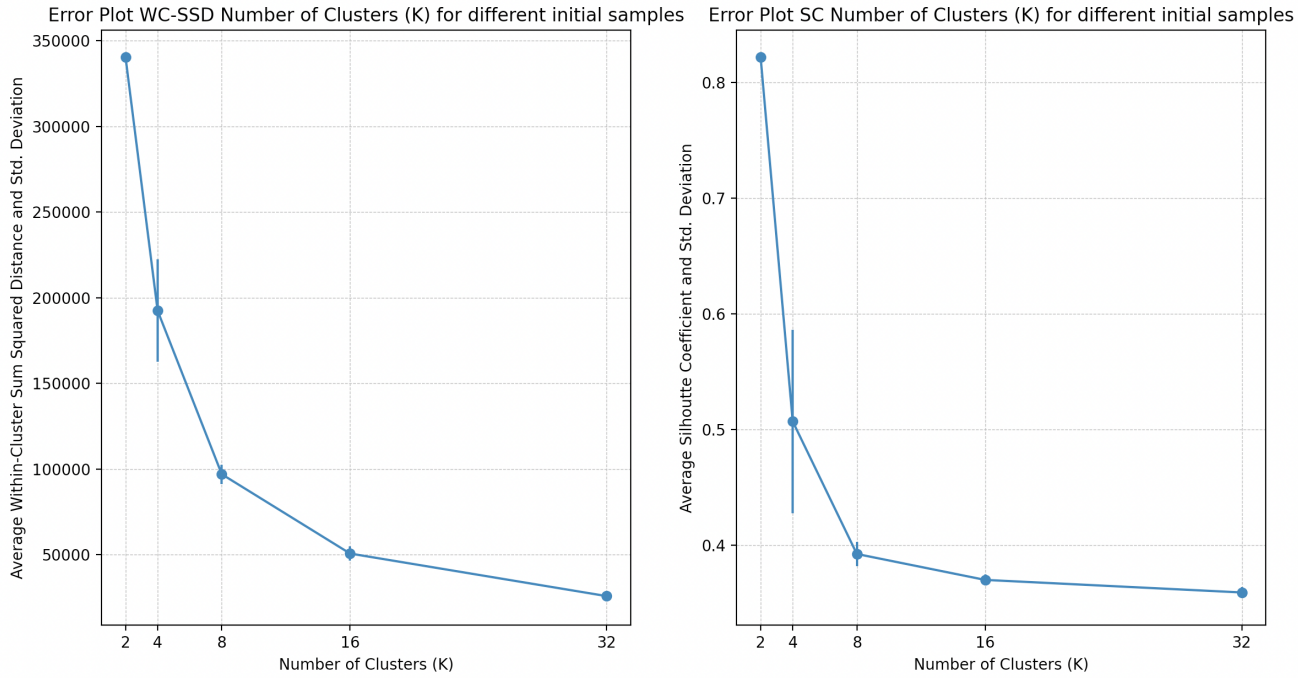


3.

Dataset 2

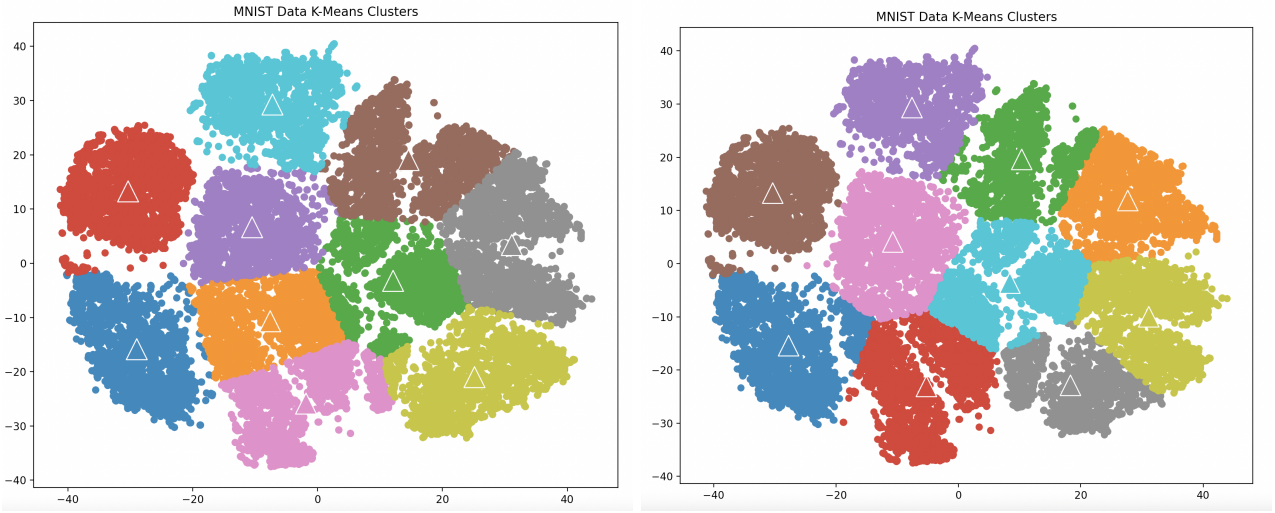


Dataset 3



The standard deviation shows that k-means is very sensitive to the initial choice of the centroids of the clusters, and the results of clustering vary greatly when the initial locations of centroids are changed. For the purpose of comparison: These are the results of clusering obtained on the same data, using the same kmeans algorithm after running for 50 epochs, but with different initial seeds:

4. The values for NMI for different versions of the data is given below. We observe that as we decrease the number of class labels in our data, the NMI increases i.e the knowledge about the classes increases more when the clusters are identfied, when the number of classes are less. As a result, as the number of classes are less, the clustering results are better.



(a) (b)

Figure 1: (a) K-Means Clusering using RandomSeed - 0 (b) K-Means Clusering using RandomSeed - 1

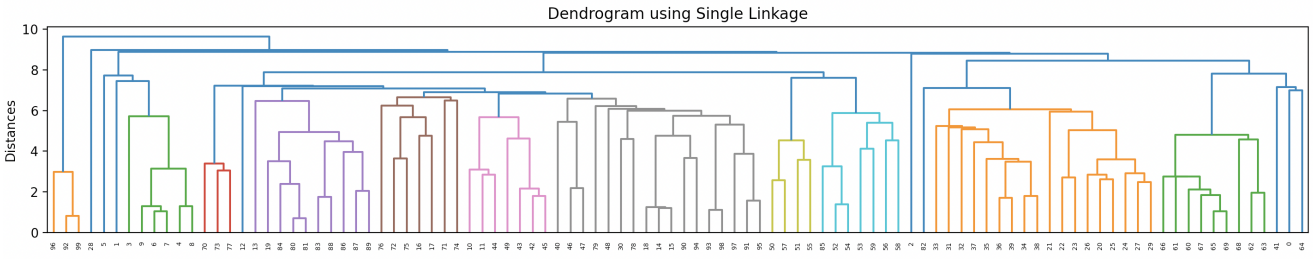
(a) (b) (c)

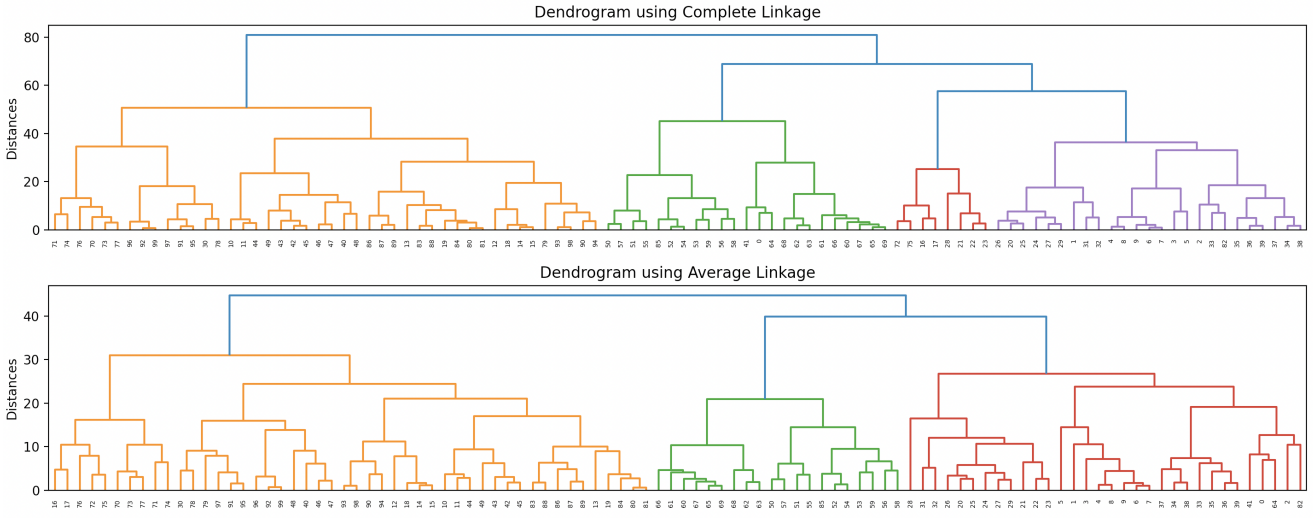
Figure 2: (a) Dataset 1 (b) Dataset 2 (c) Dataset 3

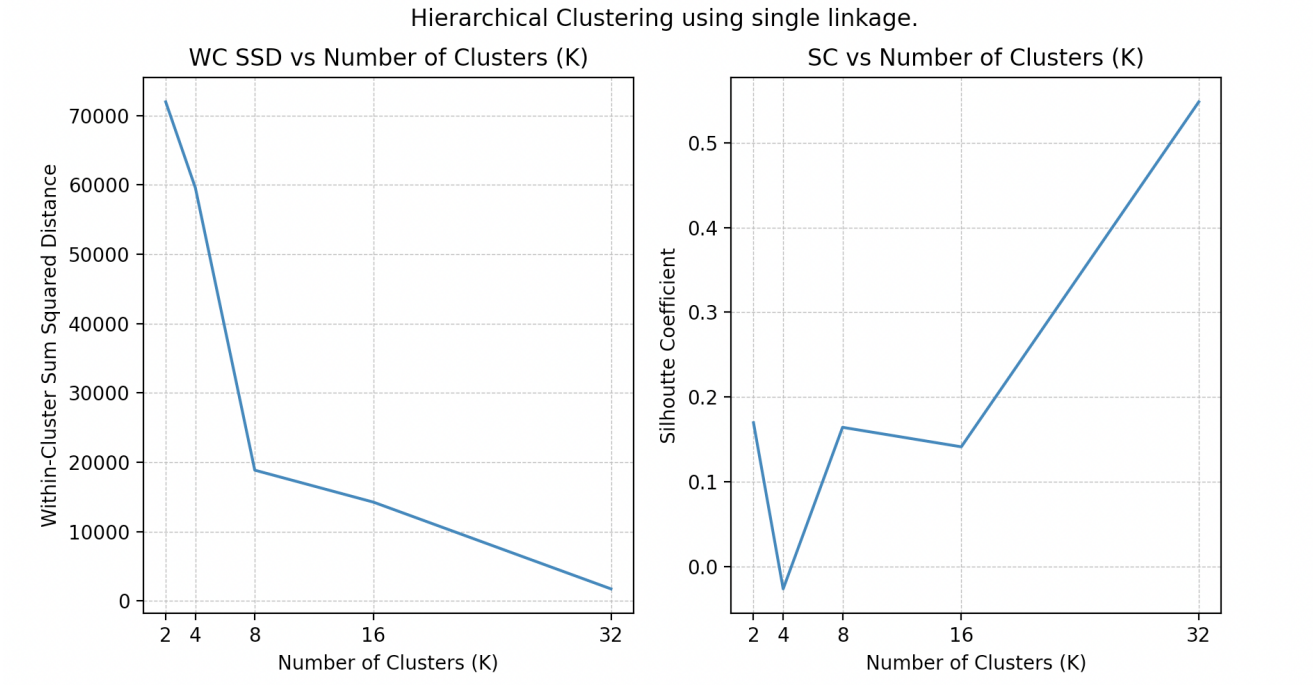
```
(dm) →  CS 573 Assignment 5 python kmeans_analysis.py
NMI: 0.3468 Dataset: 1
NMI: 0.4547 Dataset: 2
NMI: 0.4907 Dataset: 3
```

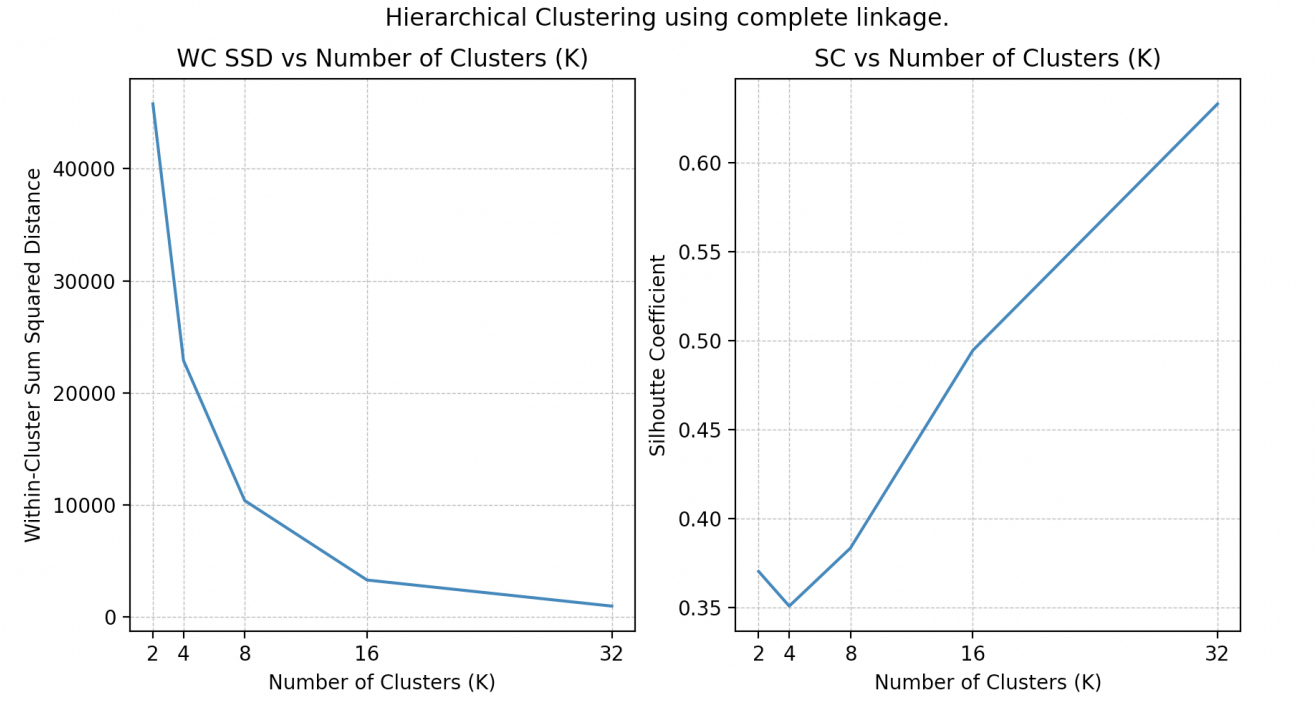
3 Hierarchical Clustering

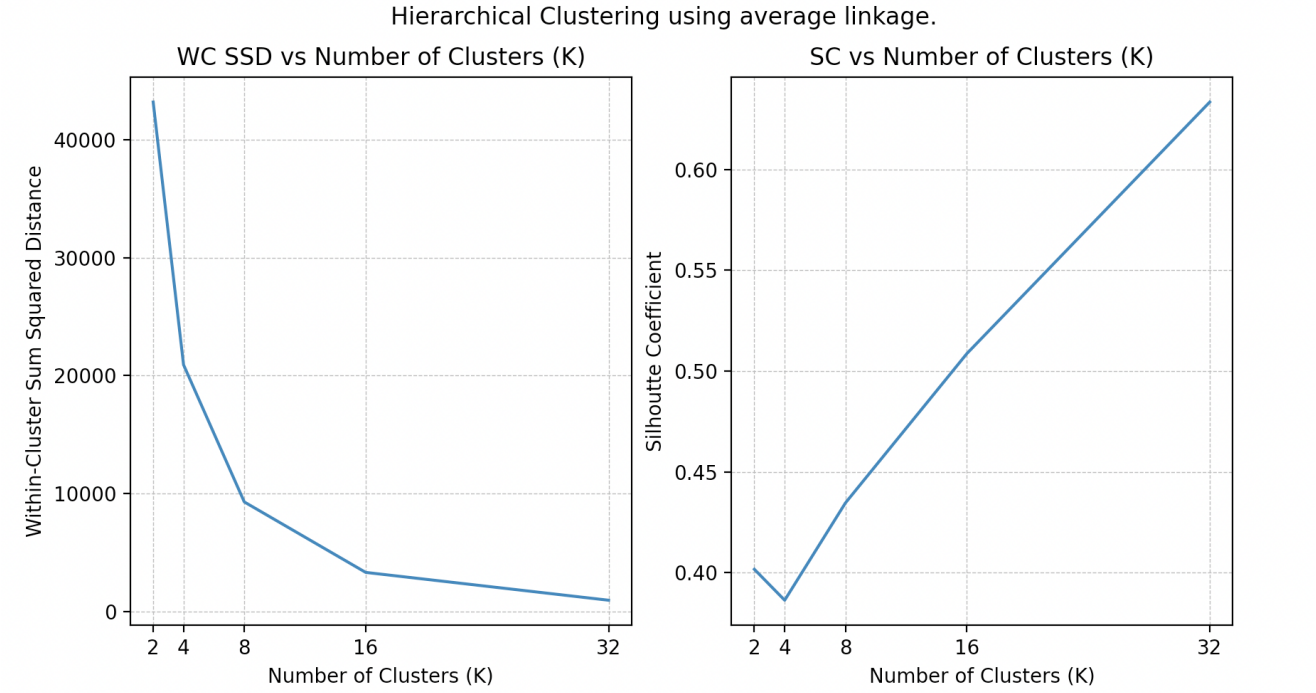
1. The code is present in **hierarchical.py**.

2. 

3. 

4. 





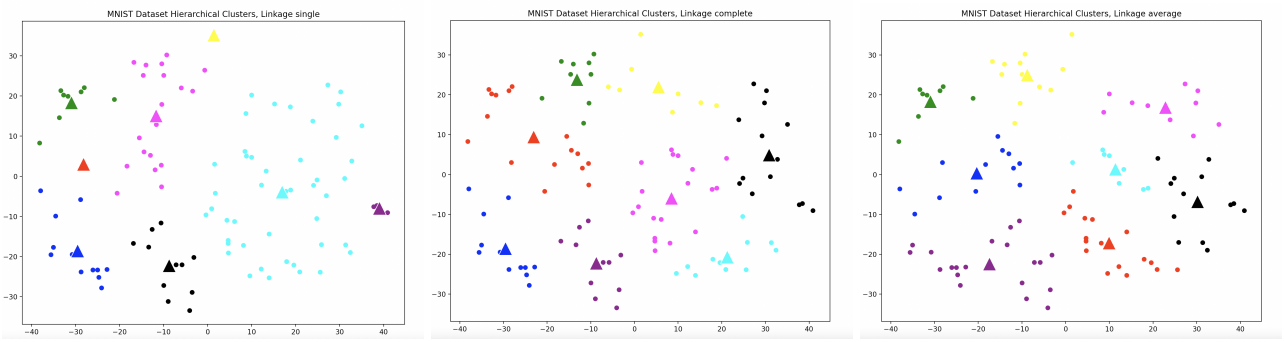
5. We can see that the SC plots do not give us a clear idea to pick an optimal K, as the value of SC increases as K increases. We must look for the elbow point in our WC-SCC plots. The values of 8 and 16 both look good for complete and average linkage, however the promising values of K for our dataset for different linkages are:

   (a) Single linkage: 8

   (b) Complete linkage: 8

   (c) Average linkage: 8

   These values are similar to the values we obtained for the same dataset using K-Means algorithm.



(a) (b) (c)

Figure 3: (a) Single Linkage (b) Complete Linkage 2 (c) Average Linkage

6.
```
(dm) → CS 573 Assignment 5 python hierarchical.py
Hierarchical Clustering NMI: 0.3317 Linkage: single
Hierarchical Clustering NMI: 0.3435 Linkage: complete
Hierarchical Clustering NMI: 0.3433 Linkage: average
```

The NMI for hierarchical clustering is slightly lower to the NMI we obtained from K-Means. Among the three linkages, the NMI for single linkage is the lowest.