**Question 1**

**What we did:**

In this question, we firstly did similar things as in Assignment 2. Namely, we loaded the Olivetti dataset, split it into train-validation-test, and trained a classifier. After that, we used AHC to reduce the dimensionality of the set by using 3 different similarity measures. After that, we used the silhouette score to choose the optimal number of clusters for each of the similarity measure. Lastly, we used the reduced dataset to train a classifier like we did with the original dataset.

**Images and/or Discussion:**

Step 1:

A collage of different faces

Description automatically generated

The image above simply shows the Olivetti dataset.

Step 2:

A white rectangular object with a black border

Description automatically generated

The image shows the shapes after train-validation-test split on the original dataset.

Step 3:



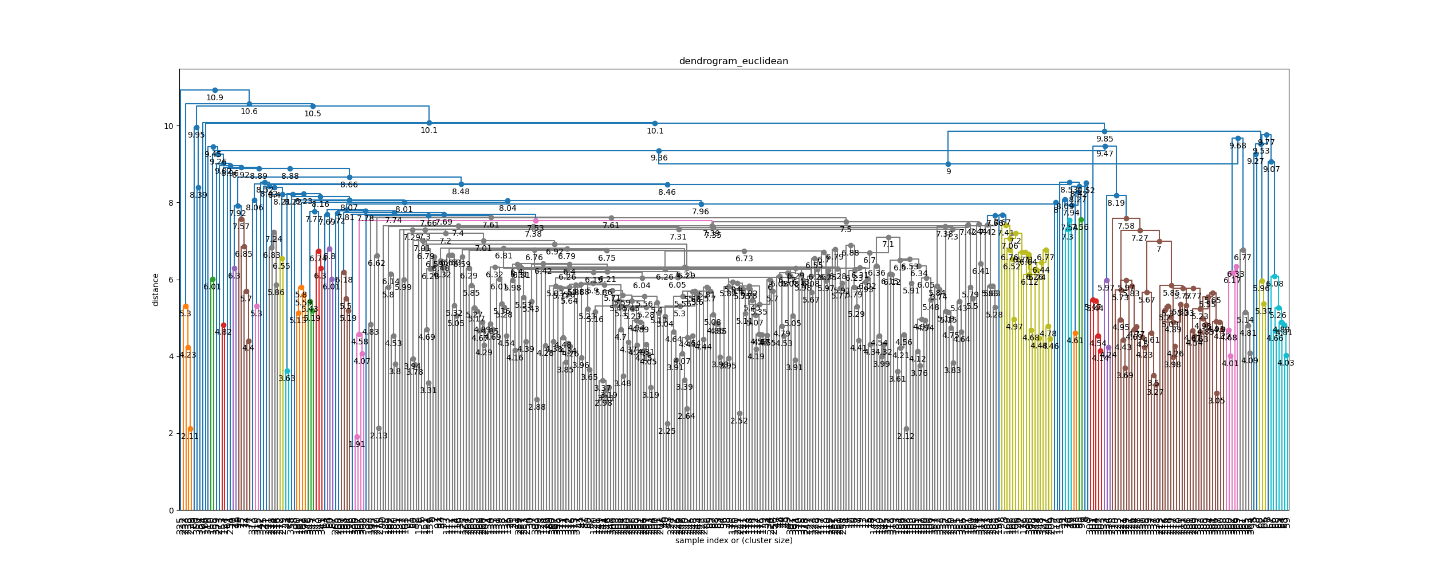
The image shows the accuracy of cross-validation with 4 folds. And the last line shows the accuracy of the classifier on the validation set.

Step 4:

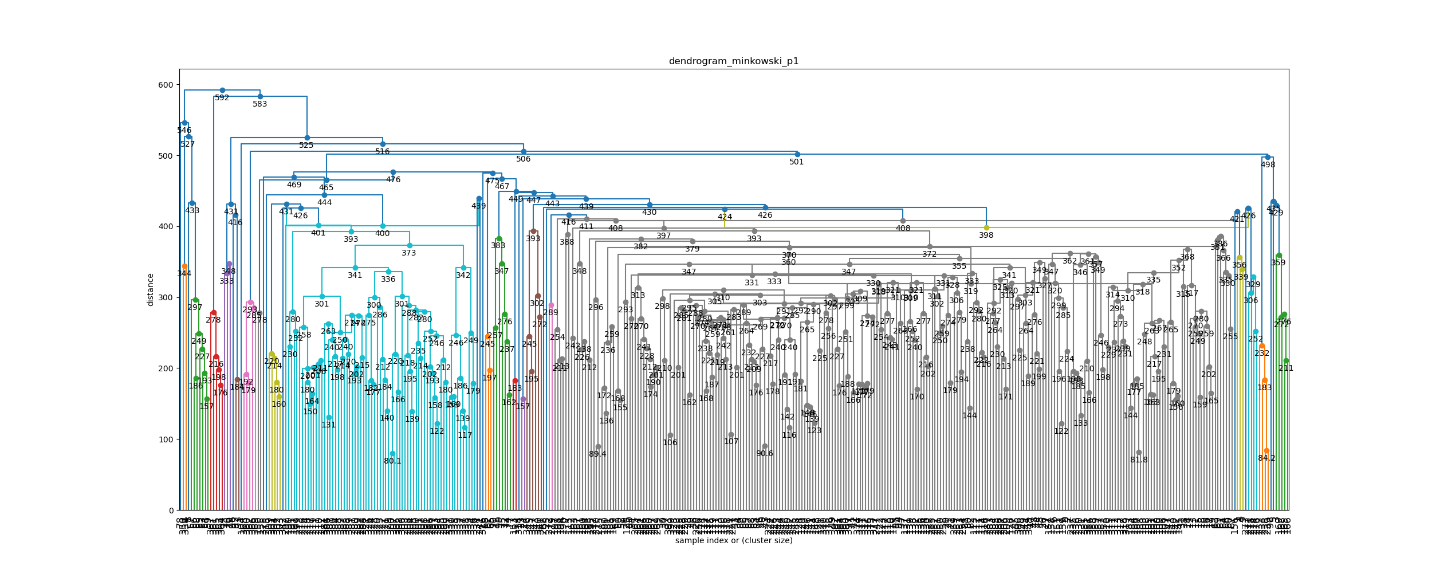
For this step, I originally used 40 as the number of clusters. This is because the Olivetti dataset has pictures of 40 unique people, so I was thinking that it would make sense to have 1 cluster for each person. And of course, this doesn’t mean that it’s a perfect number of clusters to choose because we could have maybe clusters based on other things like the shape of the nose, eyes, etc.

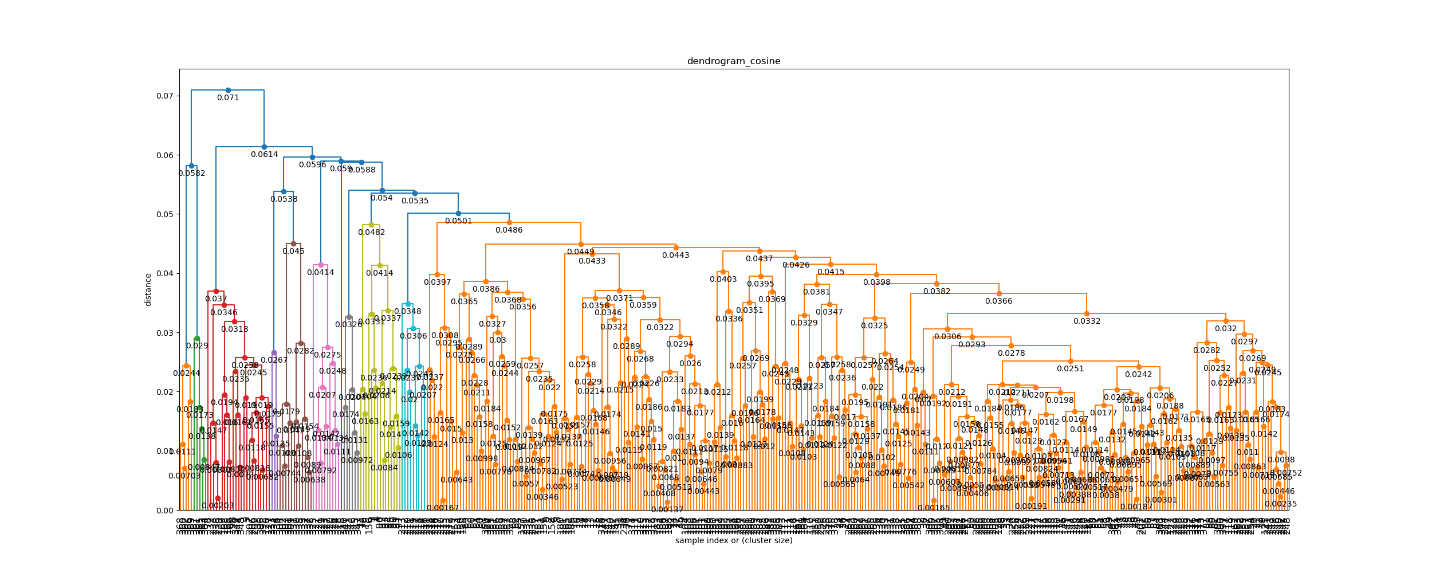
Step 5:

For this step, I will first discuss the discrepancies between the 3 similarity measures using the dendrogram for each of the similarity measure. And because the dendrogram might not be clear in this document, I will include them in the submission. And also, I used .savefig in the code, so even if the image is somehow not there, we can simply run my code.



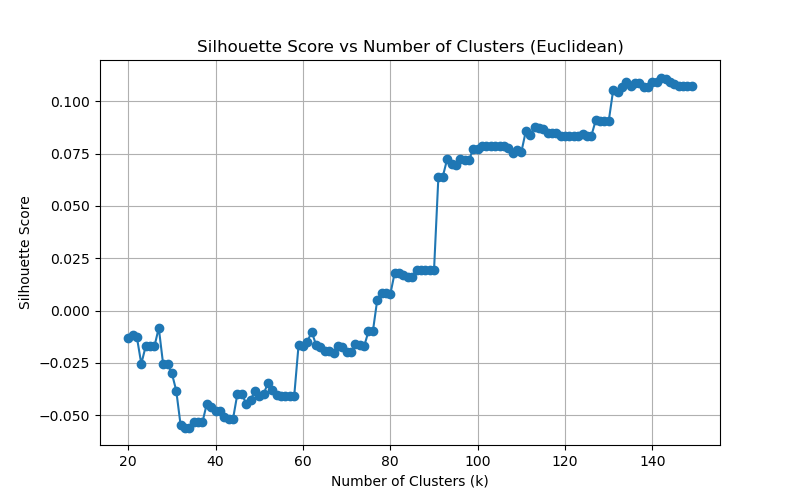
Firstly, we’ll look at the dendrogram for the clustering using Euclidean. The dots represent merges, and we can see that there are many dots at lower distance. This means that those data points that are merged has lower Euclidean distance. These lower merges might be images of the same person being clustered together, and higher merges might be clusters of different people being merged.

Secondly, we’ll look at the clustering using Minkowski (p=1), which is equivalent to the Manhattan distance. Here, we can see similar trend as the clustering previously. We can a lot of merges happening at low distance. But one thing that is different than Euclidean is that merges happen at a higher distance, e.g. the bright blue lines on the left of the dendrogram. This might be because the Manhattan distance calculates the sum of the absolute differences across all dimensions, which will result in higher distance compared to Euclidean, which is a straight-line distance. In terms of Olivetti dataset, this means that Manhattan distance captures more of the differences between each feature.

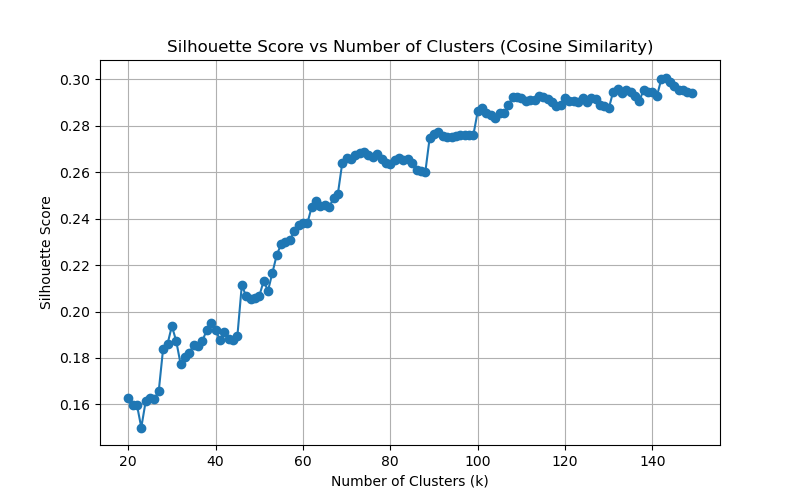


Lastly, it is the dendrogram for clustering using Cosine Similarity. This similarity measures the angle of the vectors of each data point. What this means is that it focuses on the pattern of the pixels in the Olivetti dataset. From the dendrogram, we can see that a lot of merges happen at lower distances, which makes sense because they are images of people, and those images have similar pattern where everyone similarities such as eyes, nose, and mouth.

After that, I used the silhouette score approach to find the optimal number of clusters for the clustering using 3 different similarity measures, and here is the result:

A graph with blue lines

Description automatically generated



A close up of black text

Description automatically generated

As we can see, the optimal number of clusters for all 3 similarity measures hover around 145, and it seems like it can go higher, but there is no reason to go too high because our objective is to reduce our dimensionality.

Step 6:

Here, I basically use the reduced dataset to train the classifier like previously. These are the shapes of the dataset after reducing them using the optimal number of clusters:

A screenshot of a computer code

Description automatically generated

After that, I did the same 80-10-10 training-validation-test split, and here are the shapes of the respective sets:

A screenshot of a computer code

Description automatically generated

After having those splits, I used k-fold cross validation, and here is the result:

A white background with black text

Description automatically generated

Finally, I evaluated the classifier using the validation set:

A group of black text

Description automatically generated

As we can see, the dataset that was reduced using Cosine similarity measure performs the best, and the reason might be that it’s able to capture the facial patterns better since it calculates similarity using the angle of the input vectors instead of their distances like Euclidean and Minkowski\_p1.

**Lesson learned:**

* I learned more about reading a dendrogram. On top of the course module, this video helps me understand it further: <https://www.youtube.com/watch?v=ijUMKMC4f9I&pp=ygUKZGVuZHJvZ3JhbQ%3D%3D>
* I learned that AHC is much easier to implement in Python compared to DHC (which is why I chose to do AHC for this assignment). This is because for AHC, we can use libraries like SciPy and scikit-learn, but for DHC, we have to write our own recursive functions like this:  
  <https://github.com/ronak-07/Divisive-Hierarchical-Clustering/blob/master/Divisive.py#L12-L47>
* I learned more about how to do clustering with DHC, especially using the SciPy library.
* I learned that reducing dimensionality doesn’t always mean reducing the number of features. It can also mean reducing the number of data points via clustering.
* I learned that to get the reduced X of the dataset, we can get the centroids of each cluster of data points.
* I learned that to get the reduced y of the dataset, we can get it using the most common labels of each cluster.
* I learned that in SciPy, when using Cosine similarity, the linkage with centroid method will produce incorrect results. This is because this similarity measure calculates the similarity of the angles of the vectors, and not the distances. The documentation also mentions that if we use centroid with Cosine similarity, it will produce incorrect result: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html>
* I learned further that reducing dimensionality of dataset means that we’re removing some details about the dataset, which can make the results of classification worse. However, using the reduced dataset, we can calculate things faster or visualize it better.