**Student Name:** Ayman Yasin

**Student ID:** 301193273

**Repository URL:**

**Written Report**

**Question 1:**

In this question, I started by loading the dataset of Olivetti faces. There are 400 64x64 pixel photos in this collection. For every 40 distinct people, there are ten photos. The next step involves using a classifier and a train-val-test split. Next, we utilized K-Means to decrease the dimensionality, split the data into train, val, and test, and then we employed a classifier once again. In order to cluster related photos together based on their density, we lastly used DBSCAN on the dataset.

**Images and/or Rationale:**

**Step 1:**

**A collage of a person's face

Description automatically generated**

The above image just shows what our dataset looks like. There are 40 different people.

**Step 2:**

**A graph of blue and white lines

Description automatically generated**

A graph with blue lines

Description automatically generated

A graph of a number of samples for each classes

Description automatically generated

For the second step, I decided to split the dataset into:

* 80% train
* 10% validation
* 10% test

The images above are from the respective sets and shows the number of samples per person. Since there are only 400 rows in the dataset, I decided to split it because I want to have more data for training while still having the ability to validate and test. My initial notion was to use 80-10-10 instead of 70-15-15, however since there are 10 images of a single person, 80-10-10 allows us to divide the samples uniformly into 8-1-1 images per person as opposed to the uneven 7-1.5-1.5 images. Another option would be 60-20-20, but I don't think there would be enough training data for that.

**Step 3:**

The image below shows the result of the prediction on the validation set (97.5%).

**A black text on a white background

Description automatically generated**

**Step 4:**

**A graph with blue lines

Description automatically generated**

The outcome of the silhouette score method is displayed in the image above. First off, since there are 40 distinct people in the dataset, 40 is the theoretically "correct" number of clusters. I thus experimented with the range of 20 to 100, and I found that 98 was the ideal K. Nevertheless, it seems that if K is greater, the score may increase much more. I so attempted to set the top bound at 150. The second iteration—trying it from 20 to 150—is seen in the graphic above. Additionally, 125 appears to be the ideal number. Since it appears like the scores start to decline once K is about 140, I didn't try to go any higher.

To carry out the clustering, Euclidean distance is the similarity metric that we are utilizing. Since the values in our data are the pixel values of photographs, which are integers, I think this metric should be suitable. We can use Euclidean distance to calculate the distance between two photos based on their pixel values as they are merely integers.

Since 125 was the ideal K value, we next lower the dimensionality to that number. The number of samples for each class following the dimensionality reduction is shown once again in the following image:

A graph of a number of simple lines

Description automatically generated with medium confidence

Remember that the labels on the data are now the cluster id rather than the person id. To put it another way, each of the 125 clusters that the data belongs to is identified by its label.

**Step 5:**

The image below is the result of the prediction on the validation set after dimensionality has been reduced to 125 (85%).

A screenshot of a computer

Description automatically generated

**Step 6:**

**A screenshot of a computer

Description automatically generated**

I used DBSCAN on the previously reduced dataset (which was reduced to 125D using K-Means) to get the image you see above. I experimented with several eps and min\_samples values, and this appears to get the best Silhouette score. But there are a lot of noise points and the samples aren't evenly distributed around the clusters (for example, cluster 1 contains 90 samples whereas cluster 8 has just 4 samples), so these aren't the ideal conditions.   
Apart from 125D, I also attempted to use PCA to decrease the dimensionality to 2D so that I could plot it using the tool from the class's DBSCAN lab.

**(F2024\_COMP257\_\_Lab\_2\_b\_dbscan. ipynb). Here are the outcomes:**

**A screenshot of a computer screen

Description automatically generated**

As we can see, despite having fewer clusters, we still receive a higher silhouette score in this instance. This might be because there is less distinct divisions and a lot of overlap in the larger dimensionality of 125, making clustering simpler due to the lower dimensionality.   
Similar to Step 4, Euclidean distance is the similarity metric utilized here with DBSCAN (both 125D and 2D data). Once more, the reason this similarity metric is employed in this case is that the values in our data are pixels, and since those values are merely integers, we can calculate their Euclidean distance to determine how similar they are. Pixel values will be similar, for instance, in pictures of the same subject or in pictures with comparable lighting or expression.

**Learning point from question 1:**

* I learned that it could take some time to get the ideal K for K-Means, and even then, it might not have a high silhouette score.
* I learned that in this instance, decreasing dimensionality might negatively impact a classifier's performance (turning its previously 97.5% accuracy into 85%). However, it can be beneficial for quicker processing or visualization.
* I learned that since pixel values in photos are integers that can be used to gauge similarity, the distance measured by Euclid is a suitable similarity metric for face photographs.
* I gained better knowledge about analyzing DBSCAN clustering results without creating a graph. This is due to the fact that the reduced dataset is 125-D, which makes it impossible for us to plot. Instead, we must look at other factors like the number of clusters, noise, and sample distribution between clusters.

**References**

Purdue University. (2024). Cross Validation: training, validation and testing splits: The

Examples Book. The Examples Book. <https://the-examples-book.com/starter-guides/data-science/data-modeling/resampling-methods/cross-validation/train-valid-test>