**Student Name:** Ayman Yasin

**Student ID:** 301193273

**Repository URL:** <https://github.com/Ayman850/COMP257.git>

**Written Report**

**Question 1:**

As with the previous assignment (Assignment 3), we used the Olivetti dataset in this question as well. After splitting the training set into segments for training, validation, and testing, we employed PCA to minimize the training set's dimensionality while retaining 99% of its variance. After then, the training set was shrunk from 4096 features to 222 features. The best kind of covariance for the training set was then identified. After adjusting for 35 components, I discovered that the "diag" type performs best when utilizing both AIC and BIC. Subsequently, I determined the minimum number of clusters with both AIC and BIC using the "diag" type. According to AIC, there were three clusters, while BIC found two. In order to have the fewest possible clusters, I selected step 2. Following that, we showed the PCA results, the results of determining the optimal covariance type, and the results of determining the least number of clusters. Following that, we output for each instance in the smaller training set the hard clustering assignments and the soft clustering probabilities. Next, we created five new faces using the model and visualized them. The first five images in the training set were then altered by darkening, flipping them horizontally, and rotating them in random degrees. Next, we tested the model's ability to identify the anomalies (modified images) using the score\_samples() method.

**Images and Discussion:**

**Step 3:**

The reduced dataset is obtained by running PCA with the "n\_components" parameter set to 0.99 to maintain 99% of the variance. 222 features remain in this smaller dataset as opposed to the 4096 in the original.

**Step 4:**

I choose to use Gaussian Mixture’s "n\_components" parameter of 20 for this. Since there are 40 distinct persons, I reasoned that there should ideally be 40 clusters, thus I initially attempted to employ 40. But I encountered this problem :



Consequently, I choose to reduce the number of clusters rather than raise reg\_covar in order to maybe have fewer clusters with extremely little points.

"Spherical," "diag," "tied," and "full" are the four covariance types that I tried next. Lastly, the AIC and BIC outcome is as follows:

A white background with black numbers and letters

Description automatically generated

As we can see, "diag" appears to be the best covariance type based on both criteria because it has the lowest scores.

**Step 5:**

In this step, we calculated the bare minimum of clusters. Because of the prior step, I set the covariance type to "diag" and the range to try as 1 to 20, which includes the number of components I used in the previous phase. The outcome of AIC and BIC is as follows:

A screenshot of a computer

Description automatically generated

As we can see, the optimal number of clusters according to BIC is two, however according to AIC it is three. It is understandable why BIC favors less complex models with fewer parameters than AIC, as mentioned in the course modules.   
Wanting the smallest possible number of clusters, I decided that the number of clusters for the following phases should be 2, as calculated by BIC.

**Step 6:**

In this step, we plotted results from step 3, 4, and 5.

A graph with a curve

Description automatically generated

The outcome of step 3, which involves using PCA on the training set, is the image that you see above. As we can see, 222 characteristics are required to maintain 99% of the variance.

A graph of different covarvarian types

Description automatically generated

The output of step 4, in which we attempt to ascertain which covariance type to employ, is the graphic above. It is evident that when the covariance type is "diag," the AIC and BIC are at their lowest.

A graph of a number of clusters

Description automatically generated

Step 5 produced the image above, which shows the outcome of our attempt to use either AIC or BIC to find the smallest number of clusters. According to the graph, the optimal number of clusters for AIC is three, whereas for BIC it is two.

**Step 7:**

**A screenshot of a computer code

Description automatically generated**

We used the two components and "diag" covariance type—the optimal parameters that we had identified in the earlier steps—for this point. The outcome of.predict() was then printed out. Additionally, we are able to count the number of points that are part of each of the two clusters. It is evident that 169 points are associated with cluster 0 and 151 points are associated with cluster 1.

**Step 8:**

**A screen shot of a computer code

Description automatically generated**

We printed out each instance's probability in this step, and the results match those from step 7. The first case has an almost 100% chance of falling into cluster 1, as can be seen here, confirming that the first entry in the list in step 7 is, in fact, 1.

**Step 9:**

A screenshot of a computer

Description automatically generated

In this step, five new pictures and labels were created. We saw them in visual form as well. It looks like a number of faces have been blended together, which is rather unsettling.

**Step 10:**

A collage of a person's face

Description automatically generated

I made changes to the first five training set photographs in this step. The modification steps are as follows:   
1. A rotating picture with an arbitrary angle between -180 and 180   
2. Horizontally flip the image   
3. Make the picture darker.

**Step 11:**

In this step the adjusted image from the previous phase is used to test the model's ability to identify anomalies. The 222 features that our model will accept are the same because we employed the same PCA from the previously fitted stage.   
A screenshot of a computer code

Description automatically generated

The result of both modified and unmodified photos is seen in the image above. Based on the data, I think the model can identify abnormalities. This is a result of how widely apart the scores are. For example, in the second image, the changed image scores approximately -191.1, whereas the normal image scores approximately 39.8.

**Learning point from question 1:**

* I became aware that a number between 0 and 1 for the PCA "n\_components" option will represent the variance % to be preserved rather than the actual number of components.
* I became more knowledgeable about the four types of covariance for GaussianMixture, and I discovered that an excessive number of components can distort the cluster shapes and provide an error coded as "ill-defined empirical covariance.
* I also learned that the minimum size of the "bubbles" surrounding the clusters in GaussianMixture is defined by the "reg\_covar" option. To ensure that the bubbles and bounds don't get too small, I must raise this value if I want to employ a lot of components.
* I gained more knowledge about the.score\_samples() technique and how GaussianMixture can assist in identifying anomalies.