QUEEN MARY, UNIVERSITY OF LONDON

SCHOOL OF ELECTRONIC ENGINEERING AND COMPUTER SCIENCE

ECS708P: Machine Learning

Assignment 2 - Unsupervised Learning

Renee Mendonca | 221040908

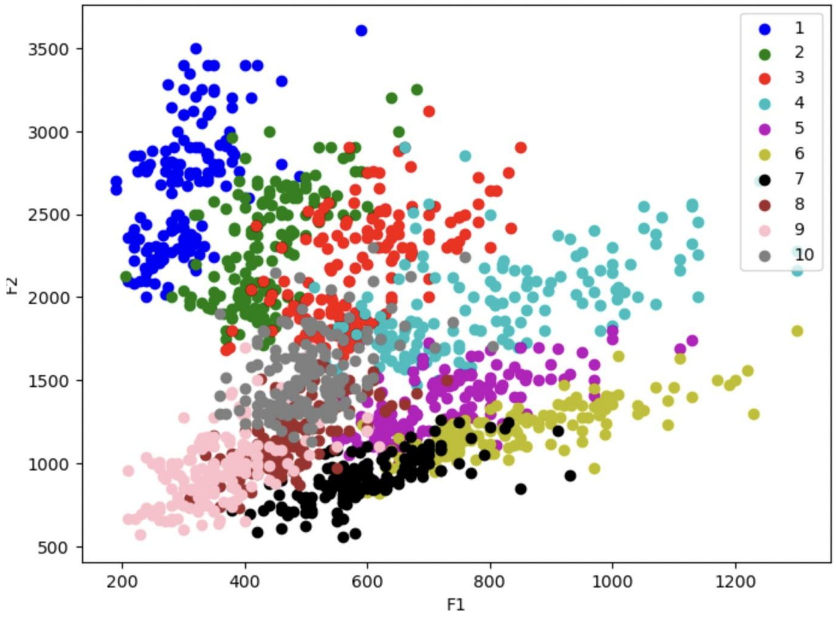
Due: 8th December, 2023

***Note: outputs in notebook are different due to last moment rerun of files.**

Q1. Produce a plot of F1 against F2. (You should be able to spot some clusters already in this scatter plot.). Comment on the figure and the visible clusters [2 marks]

Ans- We can see from the graph that clusters are formed when the values of F1 and F2, shown with various colors according to the phoneme ID.

For phoneme class 1, for instance, most F1 takes values between 200 and 400, whereas most F2 takes values between 2000 and 3500. dividing from the class as a whole at the plot's upper left corner. All classes are also capable of bifurcating and splitting into separate areas. As a result, the maximum duration of F1 and F2 takes value in a set range that is associated with a particular phoneme class.



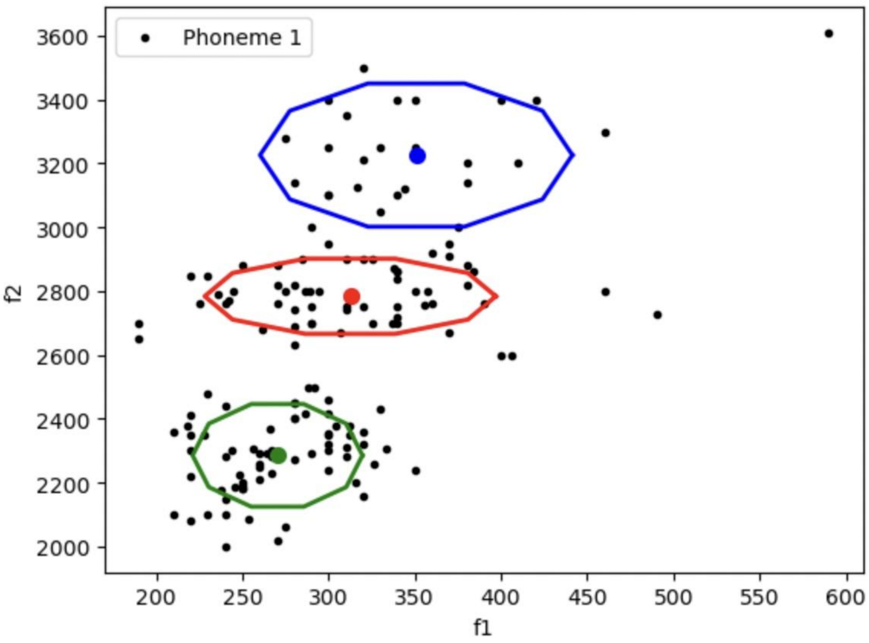
1. MoG Using the EM algorithm

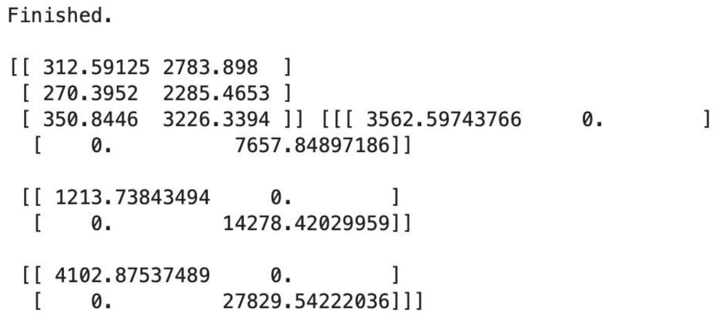
Q2. Run the code multiple times for K=3, what do you observe? Use figures and the printed MoG parameters to support your arguments [2 mark]

Ans: The mean and covariance matrix barely change after the model is trained for three clusters and phoneme class 1. All that is altered is the printing sequence in accordance with the allocation of clusters. However, the actual clusters stay the same, with about the same covariance and mean values across all runs.

This is due to the algorithm's initialization at several arbitrary setup points and its generally non-uniform conclusion in a variety of configurations.

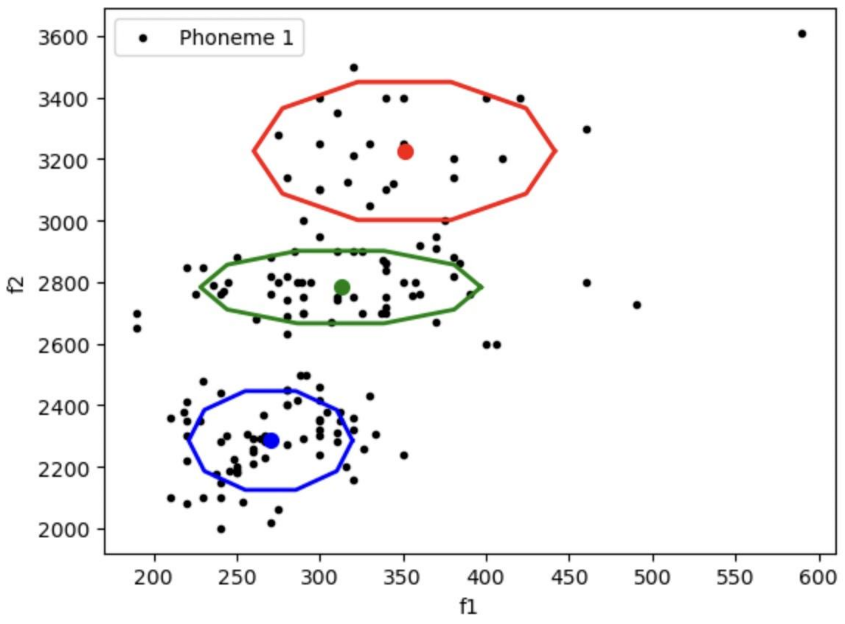
Run 1 when k=3





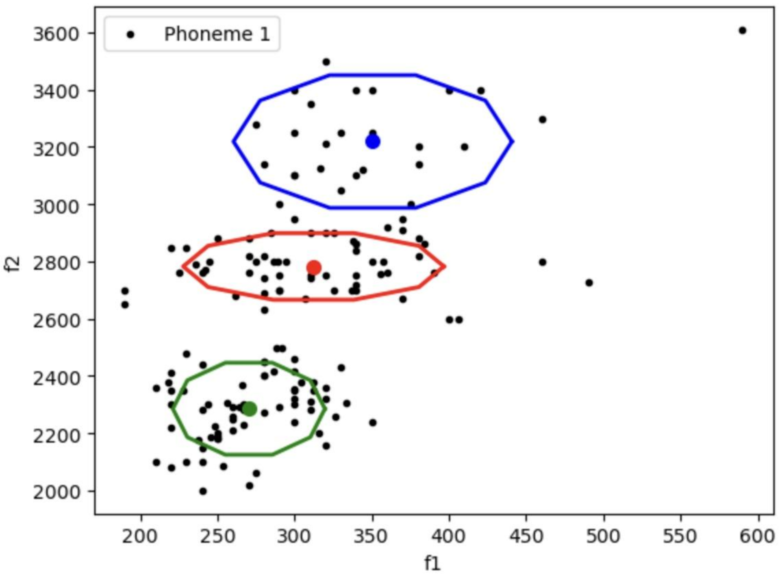
```
Finished.

[[ 312.59125 2783.898  ]
 [ 270.3952  2285.4653 ]
 [ 350.8446  3226.3394 ]] [[[ 3562.59743766     0.        ]
  [    0.          7657.84897186]]

 [[ 1213.73843494     0.        ]
  [    0.         14278.42029959]]

 [[ 4102.87537489     0.        ]
  [    0.         27829.54222036]]]
```

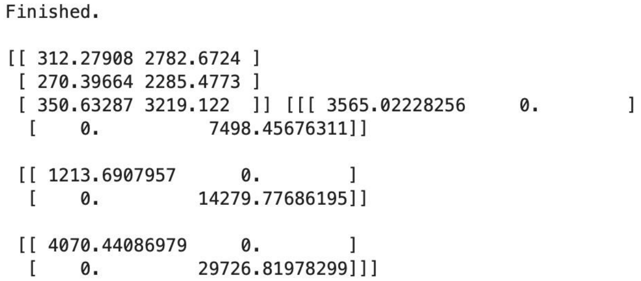
Run 2 when k=3



```
Finished.

[[ 350.8446  3226.3394 ]
 [ 312.59125 2783.898  ]
 [ 270.3952  2285.4653 ]] [[[ 4102.875375       0.        ]
  [    0.         27829.54221439]]

 [[ 3562.59743765     0.        ]
  [    0.          7657.84897244]]

 [[ 1213.73843494     0.        ]
  [    0.         14278.4202995 ]]]
```

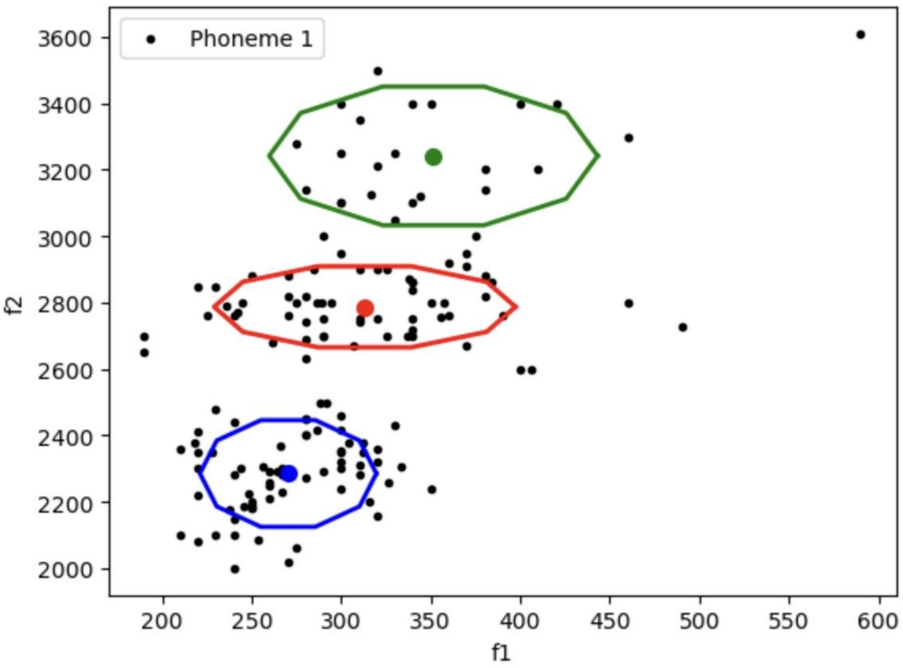
Run 3 when k=3

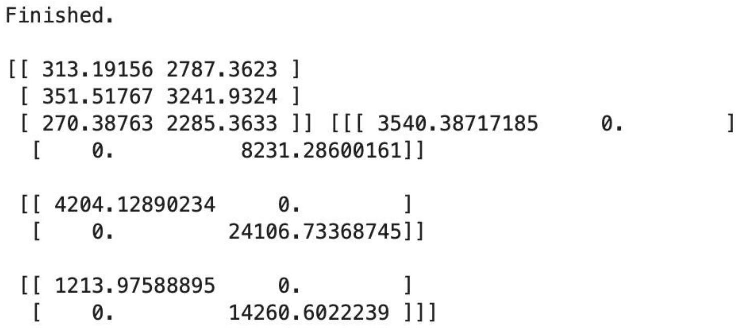




```
Finished.

[[ 312.27908 2782.6724 ]
 [ 270.39664 2285.4773 ]
 [ 350.63287 3219.122  ]] [[[ 3565.02228256     0.        ]
  [    0.          7498.45676311]]

 [[ 1213.6907957      0.        ]
  [    0.         14279.77686195]]

 [[ 4070.44086979     0.        ]
  [    0.         29726.81978299]]]
```

Run 4 when k=3





```
Finished.

[[ 313.19156 2787.3623 ]
 [ 351.51767 3241.9324 ]
 [ 270.38763 2285.3633 ]] [[[ 3540.38717185     0.        ]
  [    0.          8231.28600161]]

 [[ 4204.12890234     0.        ]
  [    0.         24106.73368745]]

 [[ 1213.97588895     0.        ]
  [    0.         14260.6022239 ]]]
```

Q5. Use the 2 MoGs (K=3) learnt in tasks 2 & 3 to build a classifier to discriminate between phonemes 1 and 2, and explain the process in the report [4 marks]

Ans: 1. In our example, we load the trained models with a predetermined number of clusters (k=3/k=6).

2. Extracting the mixture coefficient, mean matrix, and covariance matrix values.

3. Pass the data to the get_prediction function to obtain the prediction using the factors extracted earlier for both the phoneme 1 and phoneme 2 models.
4. The mean, variance, and mixing coefficient for the specific sample are returned by get_prediction.
5. To determine which particular class it belongs to, W adds all these factors together to create one.
6. We compare phoneme 1 and phoneme 2 to see which value (summed mean, variance, and coefficient) is higher and designate that as the anticipated value.
7. To determine the accuracy, we use the sk-learn accuracy_score function to compare the projected value with the ground truth.
8. The accuracy score for Miss Classification is 100, as shown.

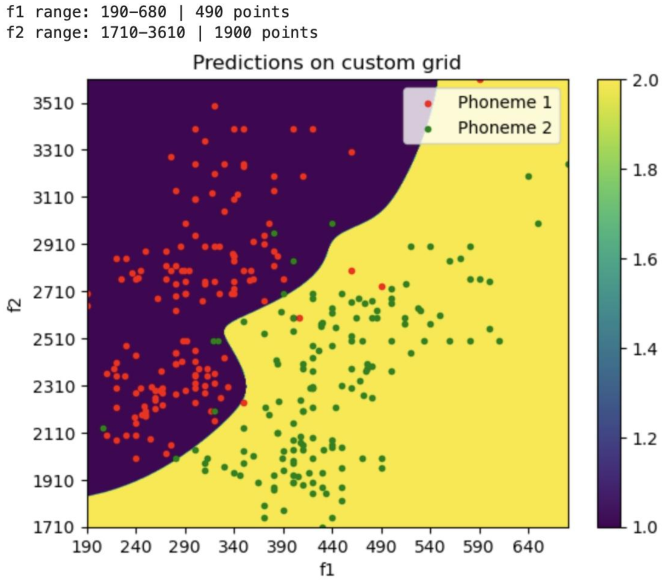
Q6. Repeat for K=6 and compare the results in terms of accuracy. [2 mark]

Ans: K=3 has an accuracy score of 0.9506578947368421.
K=6 has an accuracy score of 0.9605263157894737.
Here, we discover that the accuracy of 6 clusters is slightly higher than that of 3.
Sometimes retraining the models produced nearly identical accuracy, indicating that increasing the number of clusters had no effect on accuracy.

Q7. Display a "classification matrix" assigning labels to a grid of all combinations of the F1 and F2 features for the K=3 classifiers from above. Next, repeat this step for K=6 and compare the two. [3 marks]

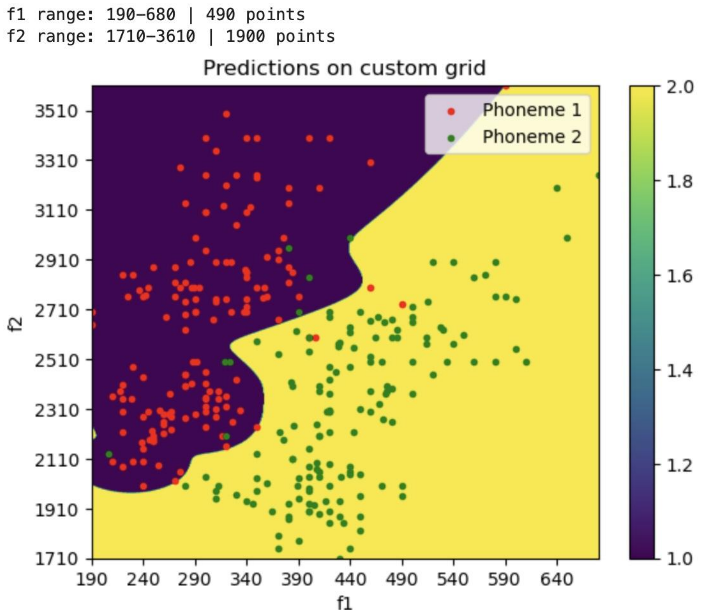
Ans:

K=3



```
f1 range: 190–680 | 490 points
f2 range: 1710–3610 | 1900 points
```

K=6



```
f1 range: 190–680 | 490 points
f2 range: 1710–3610 | 1900 points
```

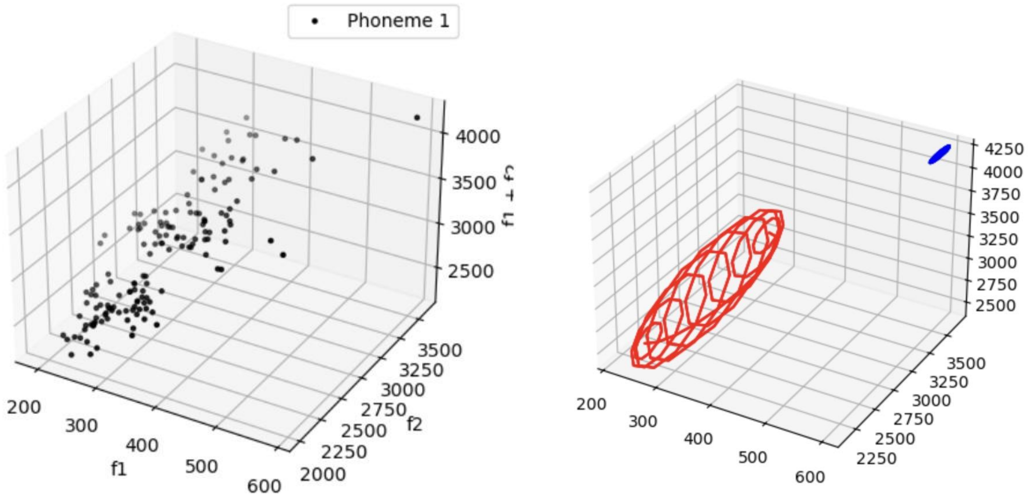
We see that the point classifications in the two scenarios are nearly identical. The line dividing the two phonemes differs, which is what makes them different.

In model 2 (k=6), certain samples that were correctly identified in model 1 (k=3) are misclassified, and the opposite is also true.

The optimal model configuration takes into account both the smallest GMM distance, which indicates the stability of the fitting process, and the largest number of clusters, or the amount of information contained.

Q8. Try to fit a MoG model to the new data. What is the problem that you observe? Explain why it occurs [2 marks]

Ans:



A very spiky Gaussian that "collapses" to a single data point is what we get when we fit a Gaussian to it using maximum likelihood. In the multivariate Gaussian case, the singularity problem is the state in which there is only one point and the variance is zero, leading to a singular covariance matrix.

$$\mathcal{N}(\mathbf{x}_n|\mathbf{x}_n, \sigma_j^2\mathbf{I}) = \frac{1}{(2\pi)^{1/2}}\frac{1}{\sigma_j}$$

When the variance hits zero, the model is overfitted and the likelihood of the Gaussian component increases to infinity. This does not occur when we fit a single Gaussian to a set of points since the variance cannot be zero. But when there is a mixing of Gaussians, it can happen.

Observations:

- The MoG model is attempting to use two Gaussian components to match the data.
- Nevertheless, one of the covariance matrices becomes unique and non-invertible as a result of the data's singularity issue.
- Numerical instability and problems with model convergence result from this.

The second feature in the synthetic dataset is a perfect linear function of the first feature, which gives rise to the singularity problem.
When components in a Gaussian Mixture Model contain singular covariance matrices, it is impossible to estimate the parameters with sufficient accuracy, which causes instability in the optimization process.
The red ellipses on the image, which stand in for the covariance matrices of the Gaussian components, clearly show the singularity problem.

In real-world settings, the singularity problem may arise from strongly correlated or duplicated information. To solve this, regularization or dimensionality reduction—discussed earlier—are frequently used as solutions.

Q9. Suggest ways of overcoming the singularity problem and implement one of them. Show any training outputs in the report and discuss. [3 marks]

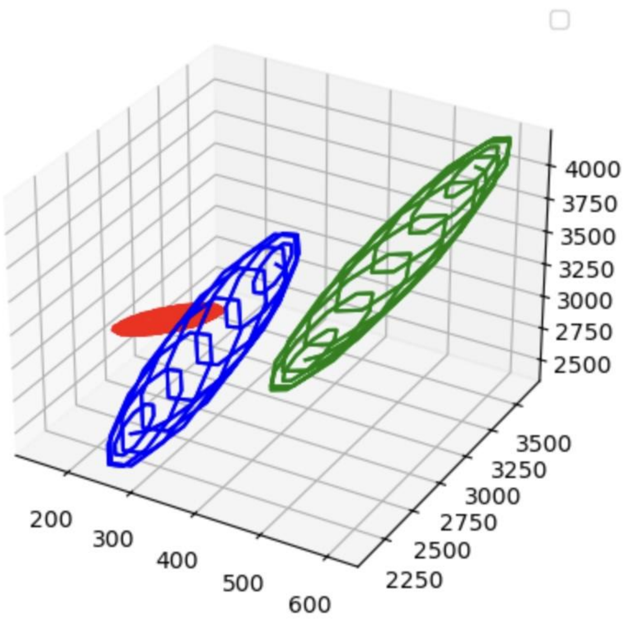
Ans: **Methods**:-
1. Regularization: To make the covariance matrix positive definite, add a little constant to its diagonal. This can avoid singularities and is also referred to as covariance matrix regularization.
2. Changing from MLE to MAP and using a previous. It suggests that the likelihood is weighed by the prior in MAP.
3. Preprocessing: Make the dataset's features more uniform or standardized.
To lessen multicollinearity, eliminate features that are superfluous or strongly associated.
Reduce the amount of features by using dimensionality reduction techniques such as Principal Component Analysis (PCA).

Technique (covariance-regularization) put into practise:-
Singularity can be avoided by simply keeping the covariance matrix from becoming a 0 matrix. This is accomplished by adding a very small value to the covariance matrix diagonal. The value we added was

If the Multivariate Gaussian collapses into a single point during the iteration between the E and M step, we obtain the 0 covariance matrix shown above. This might occur, for example, if we try to fit three gaussians to a dataset that really only has two classes (clusters). In this case, each of the three gaussians, roughly speaking, captures its own cluster, while the final gaussian only captures the point on which it sits.

$$\sigma_j{}^2 = \frac{1}{N}\sum_n \left(x_n(j) - \hat{\mu}(j)\right)^2$$

(Square root of given formula in photo i.e only variance)


4000
3750
3500
3250
3000
2750
2500
3500
3250
3000
2750
2500
2250
200
300
400
500
600