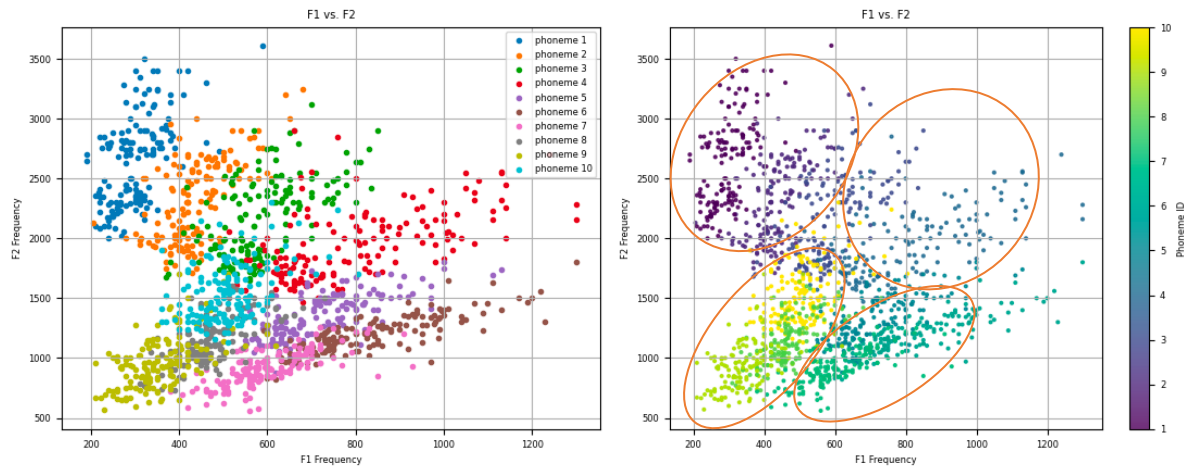


Name: Ruthwik Ganesh  
Student number: 230702930

## Machine Learning Assignment 2

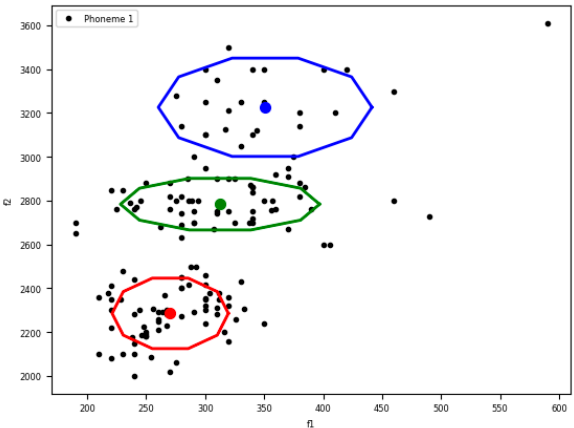
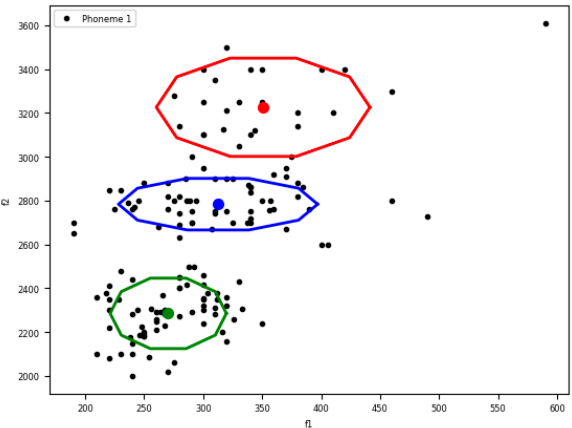
Q1. Produce a plot of F1 against F2. (You should be able to spot some clusters already in this scatter plot.). Comment on the figure and the visible clusters



The graphic above depicts a cluster of F1 vs F2 frequencies for each phoneme. There are ten Phoneme IDs in all, with each Id represented by a distinct colour. The figure should ideally show 10 distinct clusters. However, because we're only looking at two frequencies, F1 and F2, the phonemes for these two frequencies overlap and produce mixed clusters. As a result, the above map with mixed Phonemes shows four major clusters. Phoneme IDs ranging from 8 to 10 appear to form a single cluster at the plot's bottom left. We also observe a cluster of Phoneme IDs 6–8 at the bottom of the figure, another cluster appears to be formed at the top left side of the plot with Phoneme IDs ranging from 1 to 3. Finally, the last cluster can be seen towards the right side of the plot with Phoneme IDs ranging from 4 to 6.

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

The data is randomly sampled for each run and thus the central measure here, mean changes for every run thus we see a minor differences in the shapes of clusters. The mean and covariance do change yet they are comparable.

First run	Second run
	
<pre>mean: [   [270.3952 2285.4653 ]   [312.59122 2783.8977]   [350.84457 3226.338] ], cov: [   [     [ 1213.73842901  0. ]     [  0.      14278.42020411]   ]   [     [ 3562.5980625  0. ]     [  0.      7657.82809859]   ]   [     [ 4102.8710438  0. ]     [  0.      27829.75249001]   ] ]</pre>	<pre>mean: [   [ 350.8446 3226.3394]   [ 270.3952 2285.4653]   [312.59125 2783.898] ], cov: [   [     [4102.8753597  0. ]     [  0.      27829.54311454]   ]   [     [1213.73843491  0. ]     [  0.      14278.42031314]   ]   [     [3562.5974391  0. ]     [  0.      7657.84888533]   ] ]</pre>

The combination of the mean and variance tells us about the nature of each cluster. A cluster with a small variance is more tightly packed around its mean, whereas a cluster with a large variance is more spread out. The centre of each Gaussian component shows where each cluster is most dense in the  $f_1$  vs  $f_2$  space. For example, the first Gaussian (mean [270.3952 2285.4653]) is centred around a lower  $f_1$  and a higher  $f_2$  frequency compared to the others. The variances indicate the spread of the data in each cluster. A higher variance means the data points in that cluster are more spread out. For example, the first Gaussian has a higher variance in  $f_2$  (14278.42020411) compared to the other components, suggesting more spread in the  $f_2$  frequency within this cluster.

**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**

The algorithm computes the accuracy and miss-classification error on samples from phonemes 1 and 2 that have been pre-trained on those phonemes, respectively.

The logic flow is explained below:

- a) Load the two saved MoGs of phoneme 1 and 2 from Q2. and Q3. depending on the k value.
- b) Use Maximum likelihood estimation with the loaded MoGs to compute predictions for phonemes 1 and 2.
- c) Evaluate the predictions and save the results in classifications variable.
- d) Determine the accuracy by dividing the total number of accurately predicted data by the entire data.
- e) Use the formula **1- accuracy** to get the misclassification error.

Results:

- a) Accuracy: 0.9539473684210527
- b) Misclassification Error: 0.046052631578947345

**Q6. Repeat for K=6 and compare the results in terms of accuracy**

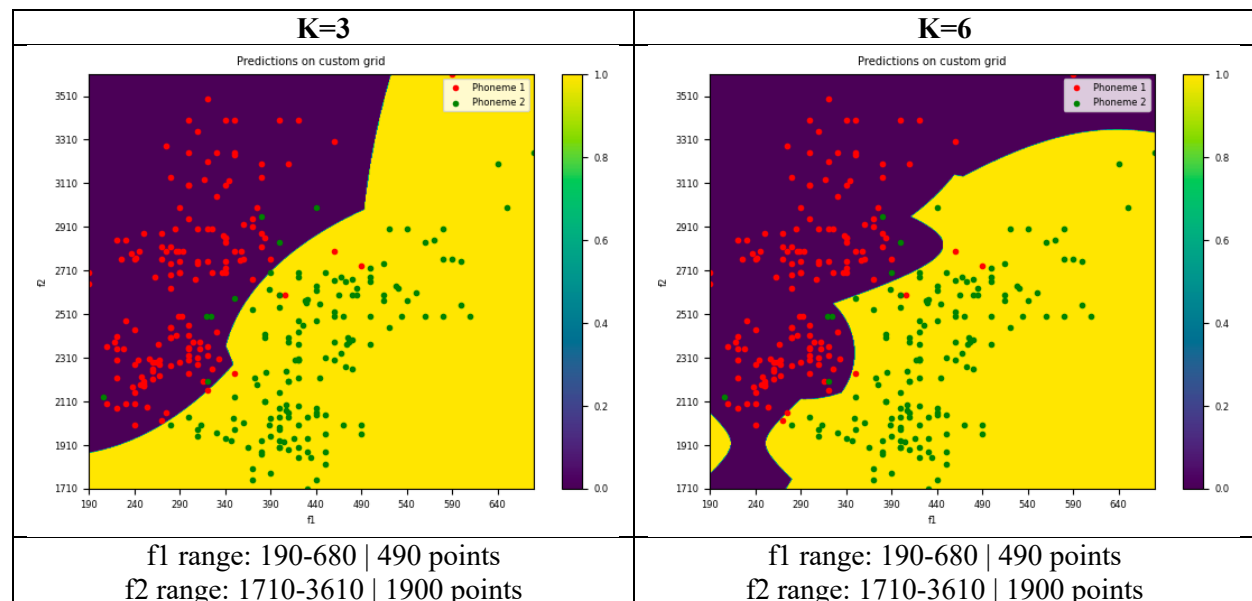
We see a slight increase in the accuracy with 6 clusters. By increasing k to 6, the MoG model can capture more nuanced aspects of the frequency distribution for a given phoneme, potentially leading to a more detailed understanding of the variations within this phoneme. Each Gaussian component offers a different perspective on the data, revealing the complexity and diversity of the phonetic characteristics represented in the dataset.

Results:

- a) Accuracy: 0.9572368421052632
- b) Misclassification Error: 0.042763157894736836

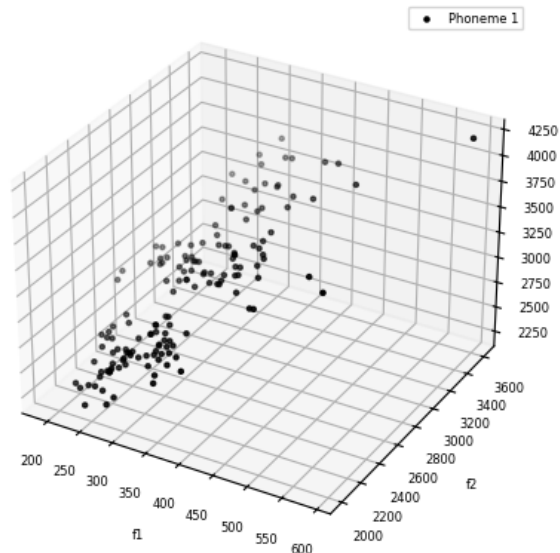
Q7. Display a "classification matrix" assigning labels to a grid of all combinations of the F1 and F2 features for the K=3 classifiers from above. Next, repeat this step for K=6 and compare the two.

Looking into the graphs below we see that for 6 clusters the decision boundary extends to capture outliers resulting to an overfit, this is not the case for 3 clusters which has generalised the two phonemes well. It's worth noting that increasing the number of clusters in Gaussian Mixture Models (GMMs) does not necessarily guarantee a better fit. The optimal number of clusters depends on the characteristics of the data. Adding too many clusters can lead to overfitting, where the model becomes too specific to the training data and may not generalize well to new, unseen data. On the other hand, too few clusters may result in underfitting, where the model fails to capture the underlying structure of the data.



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

The code will return a **ValueError**: Input contains infinity or a value too large for dtype('float64'). This is due the singularity problem of the covariance matrix. This issue is resolved in the Q9.



The singularity problem in Gaussian Mixture Models (GMMs) arises when one or more components of the mixture become too similar or collapse, leading to a singular (non-invertible) covariance matrix. In the context of GMMs, each component is represented by a multivariate Gaussian distribution with its mean vector and covariance matrix.

The covariance matrix is a crucial part of the GMM as it captures the shape and orientation of the Gaussian distribution in each component. A singular covariance matrix implies that the variables are not linearly independent, which can cause numerical instability and affect the performance of algorithms that rely on the inversion of the covariance matrix, such as the Expectation-Maximization (EM) algorithm.

The singularity problem often occurs when one of the components of the GMM has too few data points or when the data points are highly correlated. In such cases, the covariance matrix becomes ill-conditioned, making it difficult to estimate accurately. This can lead to issues in the parameter estimation process of the EM algorithm and result in poor model performance.

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

The singularity problem in Gaussian Mixture Models (GMMs) can be mitigated through various techniques. Here are some common approaches:

1. **Regularization:** Applying regularization techniques to the covariance matrix during the estimation process. Regularization adds a small constant to the diagonal of the covariance matrix, ensuring that it remains well-conditioned and preventing singularity. One common form of regularization is adding a small multiple of the identity matrix to the covariance matrix. This is often referred to as "diagonal loading."
2. **Covariance Clipping:** Clipping or truncate the eigenvalues of the covariance matrix to prevent them from becoming too small. This helps maintain numerical stability and avoids singularity.
3. **Initialization:** Initializing the parameters of the GMM. Random initialization can sometimes lead to components with very few data points, causing singularity issues. K-means clustering, or hierarchical clustering can be used to initialize the means and covariances more effectively.
4. **Dimensionality Reduction:** If the dimensionality of the data is high, consider applying dimensionality reduction techniques before fitting the GMM. This can help reduce the risk of singularity and improve the efficiency of the algorithm.
5. **Model Selection:** Choosing an appropriate number of components for the GMM. Having too many components relative to the amount of data can lead to singular covariance matrices. Model selection techniques, such as cross-validation or information criteria (e.g., AIC, BIC), can help determine the optimal number of components.
6. **Data Pre-processing:** Data should be pre-processed properly. Standardize or normalize the features to have zero mean and unit variance. This can improve the conditioning of the covariance matrix.

By applying a combination of these techniques, practitioners can mitigate the singularity problem in Gaussian Mixture Models and improve the overall performance and stability of the model. It's important to carefully consider the characteristics of the data and experiment with different approaches to find the most suitable solution for a given problem.

The sum of f1, f2 was normalised for k=3 and phoneme\_id = 1 and a small (0.001) regularization value to the covariance matrix for k=3 and phoneme\_id = 2, which gave better results as the clusters covered most of the points leaving out the outliers which is a sign of a good fit.

