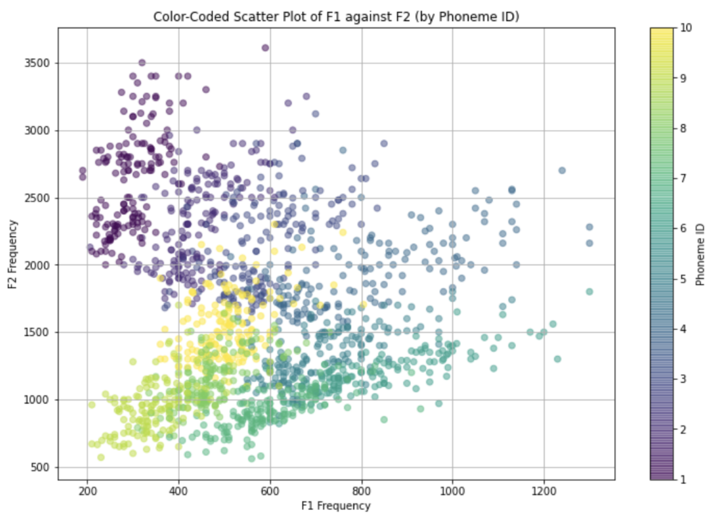
Q1. Produce a plot of $F1$ against $F2$ (You should be able to spot some clusters already in this scatter plot.). Comment on the figure and the visible clusters [2 marks]



The figure above shows distinct clusters of data points. These clusters represent different phonemes, as each phoneme has its characteristic formant frequencies. The distribution of frequencies in both F1 and F2 appears to cover a wide range, indicating variability in the phoneme production across different speakers. The horizontal and vertical dispersion suggests that F1 and F2 capture different characteristics of the phonemes. For instance, F1 could be more related to the height of the tongue during pronunciation, whereas F2 might be more influenced by the front-back position of the tongue. The visible separation between some clusters suggests that these features (F1 and F2) could be useful for phoneme classification tasks.

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]

I observe that the three circles (representing the Gaussian components in red, blue, and green) begin closely positioned and joined together and then separate from each other after running the code multiple times for K=3. As the algorithm progresses through iterations, it refines the parameters (mean and covariance) of each Gaussian, leading to a better separation and a more accurate representation of the underlying clusters in the data. The covariance matrices, which are diagonal in this case, indicate the spread or variance of each Gaussian component along each feature axis. The variances along the diagonal suggest the extent of spread in the f1 and f2 frequency dimensions. This output demonstrates how the EM algorithm has converged to these parameters, reflecting the clustering of the data in the two-dimensional feature space.

```
Finished.

[[ 270.3952  2285.4653 ]
 [ 350.84372 3226.3118 ]
 [ 312.5901  2783.8928 ]] [[[ 1213.73821191     0.        ]
  [    0.         14278.42697584]]

 [[ 4102.73665808     0.        ]
  [    0.         27836.65061886]]

 [[ 3562.61533753     0.        ]
  [    0.          7657.16360165]]]
```

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]

The process began with data preparation, where I extracted and combined data for these specific phonemes from a larger dataset. This step was crucial to focus my analysis on the phonemes of interest. Next, I loaded pre-trained GMM parameters for each phoneme. These parameters, including means, covariances, and mixture probabilities, were essential for determining the likelihood of data points belonging to each phoneme's distribution. I then employed the get_predictions function to calculate these likelihoods. This step was pivotal in assessing how well each data point fit into the Gaussian distributions of phoneme 1 and phoneme 2. For classification, I adopted the Maximum Likelihood approach. Each data point was assigned to the phoneme whose GMM yielded the higher likelihood. This method ensured a data-driven, probabilistic classification grounded in my pre-trained models. Finally, I evaluated my classifier by calculating its accuracy and the miss-classification error rate. These metrics were important in assessing the effectiveness of my classifier. They provided insight into how well the GMMs could differentiate between the two phonemes.

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

With K=6, my classifier achieved an accuracy of approximately 95.72% and a miss-classification error rate of about 4.28%. Comparing this to the results from K=3, where the accuracy was approximately 96.38%, it seems that increasing the number of Gaussian components to 6 did not improve the classifier's performance. In fact, there's a slight decrease in accuracy.

For K=3:

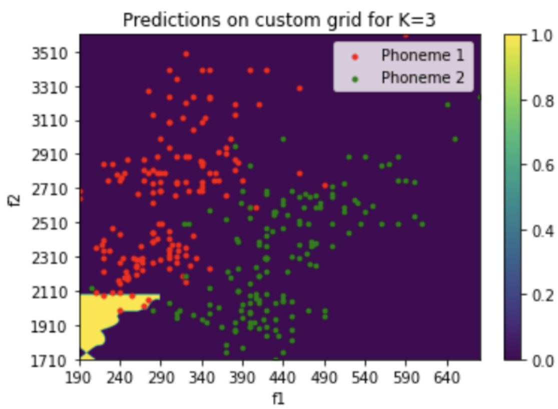
```
Predicted Classifications: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
 1 2 2 2 2 2 2 2]
Accuracy: 0.9638157894736842
Miss-classification Error: 0.03618421052631582
```

For K=6:

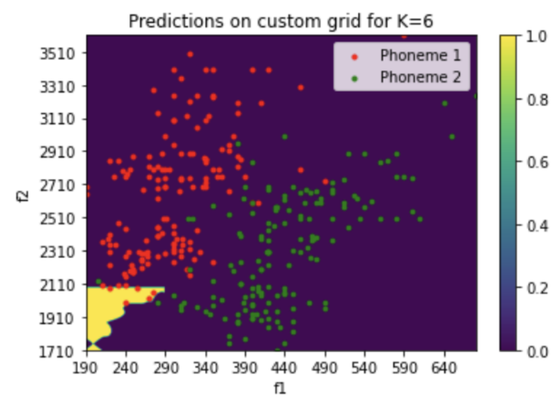
```
Predicted Classifications: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
 1 2 2 2 2 2 2 2]
Accuracy: 0.9572368421052632
Miss-classification 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. [3 marks]

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

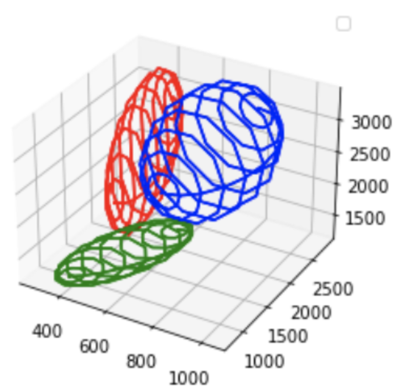


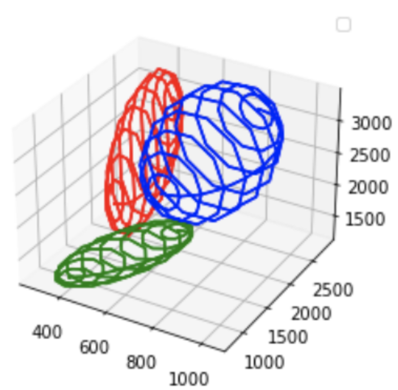

```
f1 range: 190–680 | 490 points
f2 range: 1710–3610 | 1900 points
```



Both classification matrices for K=3 and K=6 have the same f1 and f2 range and their clusters are spread out the same way.

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





Overlapping Gaussians: The Gaussian components appear to be overlapping significantly. This might suggest that the model is having difficulty distinguishing between the different clusters in the data. Overlapping mainly occurs if the data does not have clear or well-separated clusters that the model can identify.

Covariance Estimation: The shapes of the Gaussians suggest that the covariance matrices are not well-estimated. If the covariance matrices are too similar, the model may not capture the true spread and orientation of the data points within each cluster.

Q9. Suggest ways of overcoming the singularity problem and implement one of them.

Adding a small value to the covariance matrices diagonal during the EM algorithm's M-step. This prevents the determinants of the covariance matrices from becoming zero.

Restricting the model to use diagonal covariance matrices, reduces the complexity of the model and can help prevent singularities. If the dataset is too small, it may provide enough information to estimate the parameters of the model accurately. Increasing the dataset size can help overcome this.