Indian Institute of Technology, Madras

Vedant Saboo - CS19B074, K V Vikram - CS19B021

CS5691 - Pattern Recognition and Machine Learning
Assignment 3

7th April, 2022

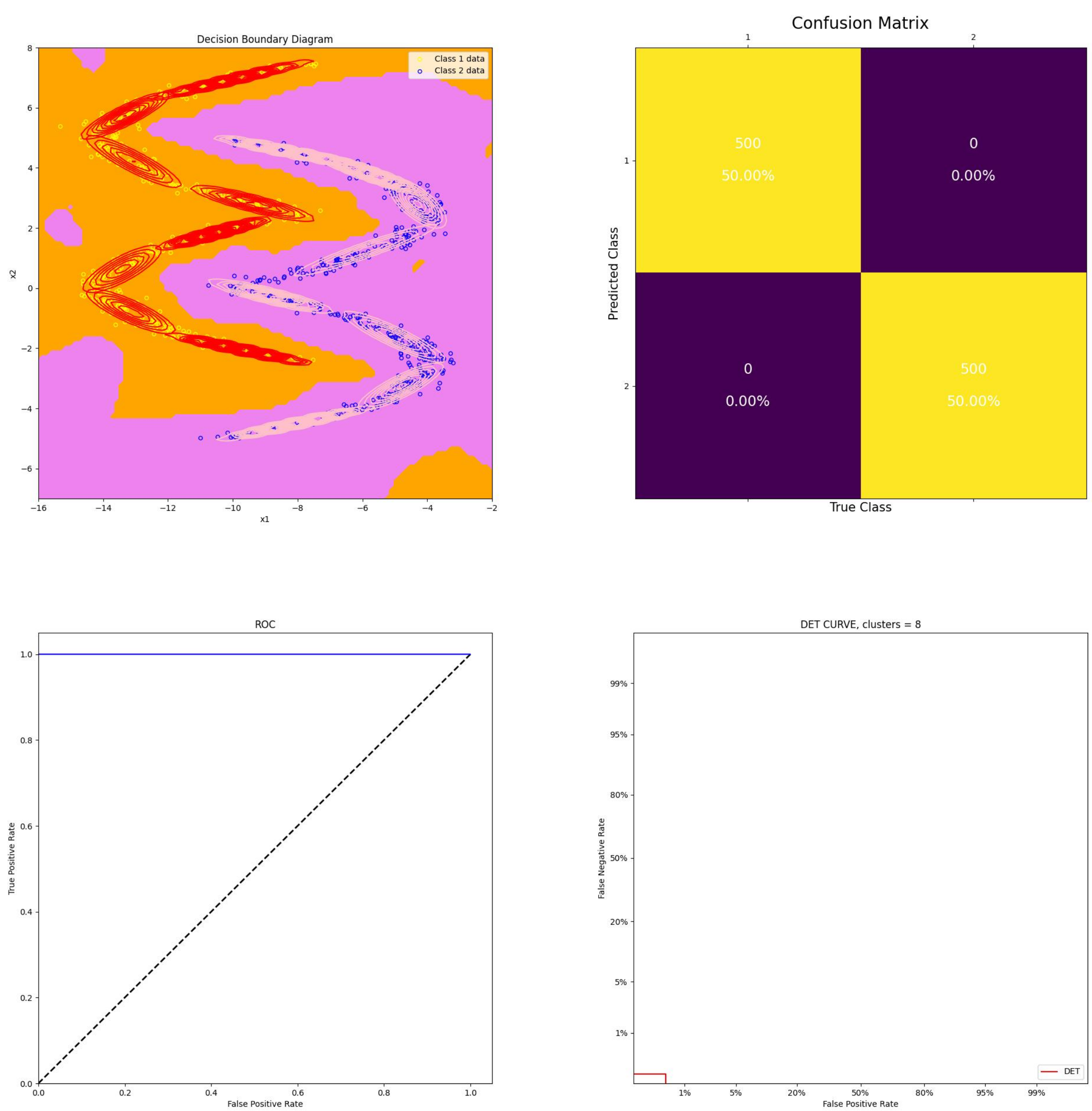
Contents

1 Task A : K Means, GMM

a Synthetic Dataset

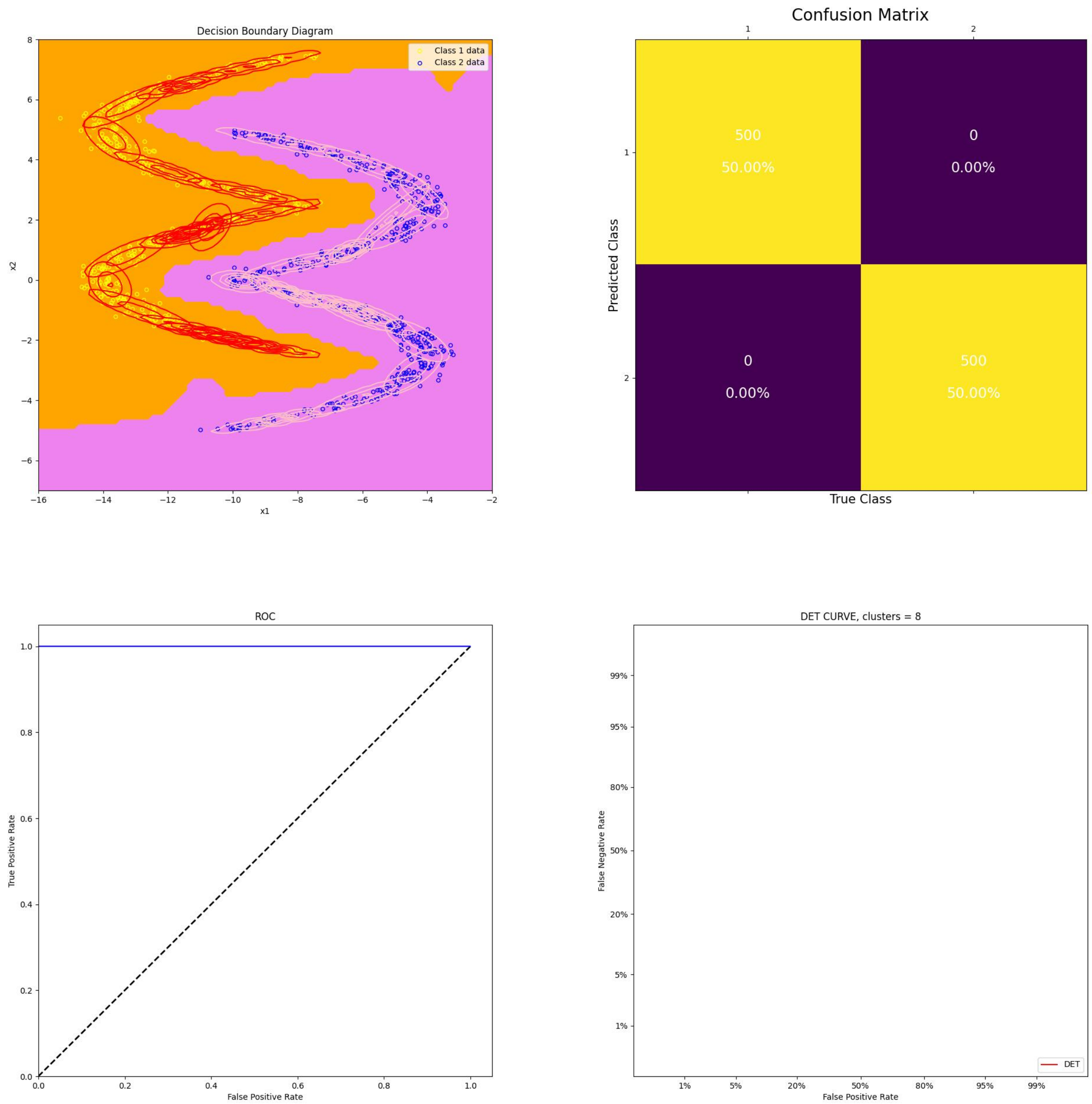
There are two classes. We modeled each class as a mixture of Gaussians. The number of clusters, k, and the number of EM iterations were treated as hyper parameters. After trying out and reasoning, we found that the classes would be best modeled with 14 clusters for each class and 20 EM iterations. We used K Means algorithm to initialise the mixture. For K Means clustering, we used EM algorithm as well. We fixed the iterations to be 30. To decide on this, we noted that 1. K Means algorithm always gets better on each iteration from any starting configuration. 2. 30 iterations were rather quick to execute.

For k = 8, iterations = 30, following results were obtained.



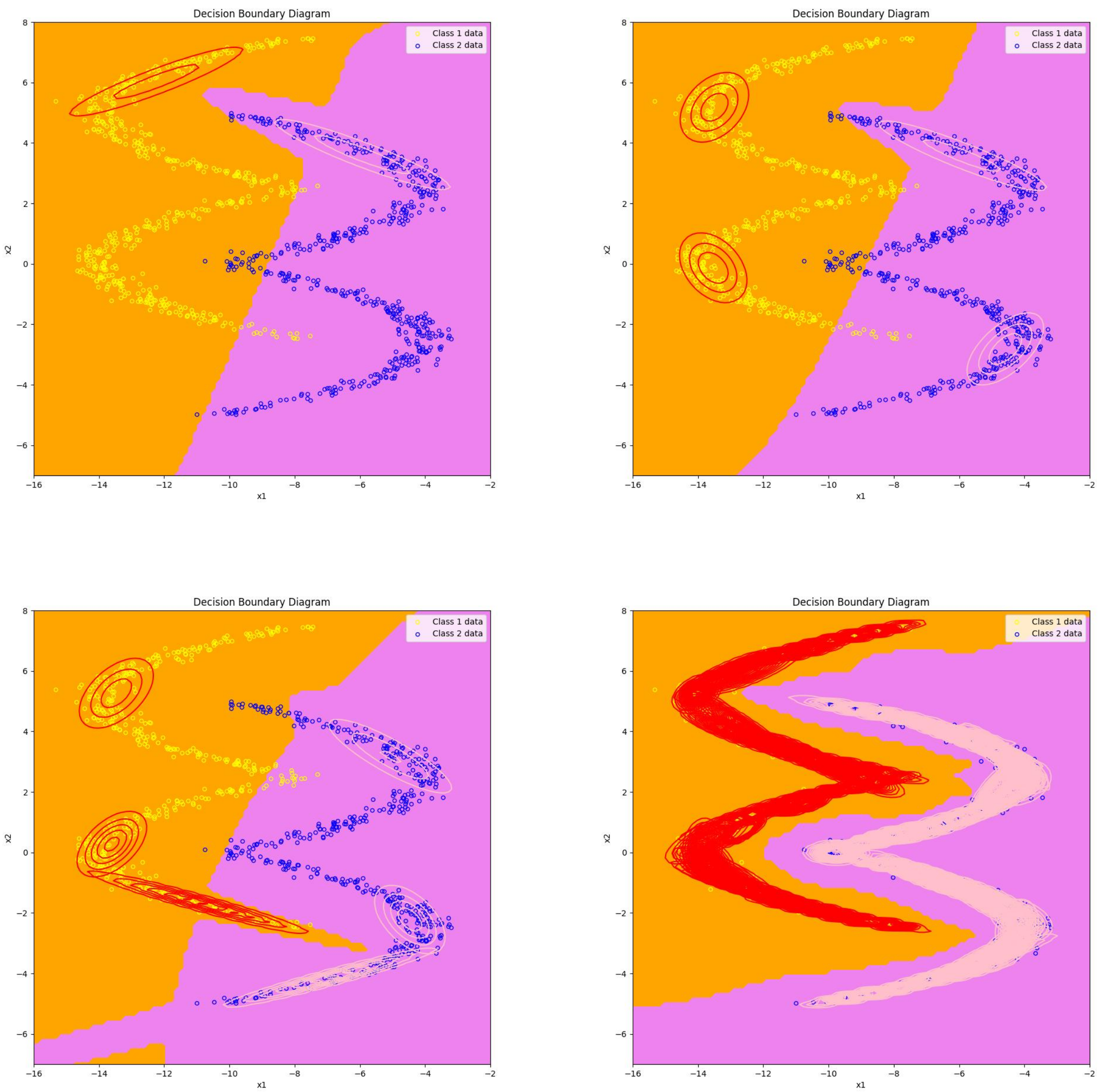
(a) Decision Boundary (b) Confusion Matrix (c) ROC Curve (d) DET Curve

For k = 16, iterations = 30, following results were obtained.



(a) Decision Boundary (b) Confusion Matrix (c) ROC Curve (d) DET Curve

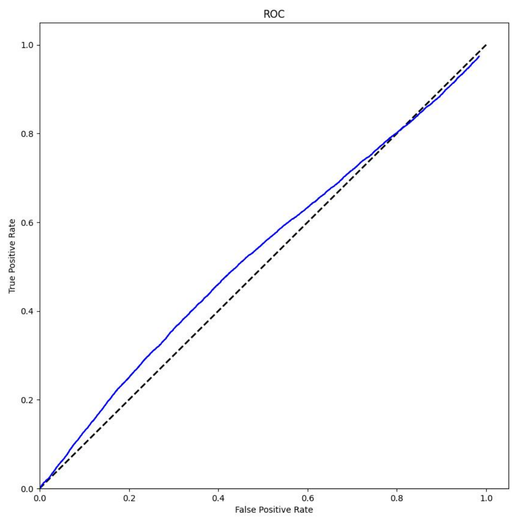
It is interesting to see how the clusters change with various k. We have shown the results for k = 8 and 16. Below are results for k = 2,3,4 and 32.

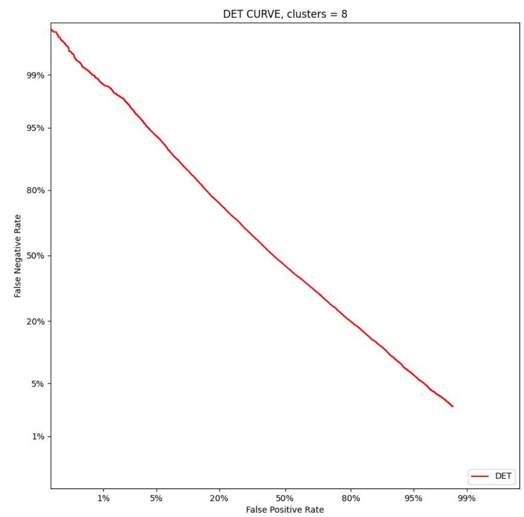


(a) k = 2 (b) k = 3 (c) k = 4 (d) k = 32

b Image Dataset

We were given 1200+ images belonging to **five** classes, each image divided in 36 blocks and a 23 dimension feature vector extracted from each image.
We used **10 cluster GMM** to model the data, obtaining an effeciency of 27.8 percent.
We ran several runs of the clustering for various values of the hyper parameters, mainly number of clusters and initialisation of the clusters. We found that the best accuracy is possible for the case when initialisation is done using **K Means** algorithm.
Here, we present the ROC and DET curves as we obtained them.





(a) ROC (b) DET

The results are not very good and the computation time was also very high. The confusion matrix was as follows.
2370 258 0 0 0
2117 259 0 0 0
1634 238 0 0 0
2284 416 0 0 0
2626 326 0 0 0

This shows that the model classifies most the images as the 'coast'. This is because scores given by all the other models are evenly spread, thus first class exceeds them. Thus most of the time, model predicts class 1.

We also note that GMM is heavily dependent on initialisation. For different instances, we could achieved strikingly different results. In fact, one of the runs we achieved an accuracy of 60 percent! However, the next run with the same parameters, the accuracy dropped to as low as 18 percent, worse than the expected accuracy of a random predictor!

The **synthetic data**, however, gave exemplary results on our system, touching **100 percent accuracy** for each run!

2 Task B - DTW and HMM

a Dynamic Time Warping on Isolated Spoken Digits(ISD)

The results for ISD data were really excellent without any normalization. This suggests that the provided mfcc data has been highly processed and tailor-made for ML classification. We performed DTW with/without windows and the results are given below. We consider the lowest $20\% = 8$ of the dtw errors against a class templates for averaging, in order to find the score of a test series towards a class.

i Without windows

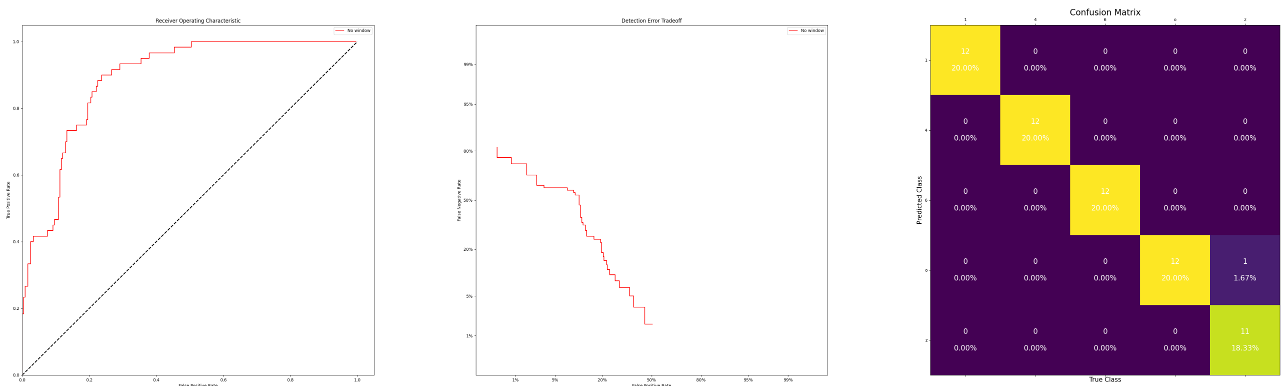
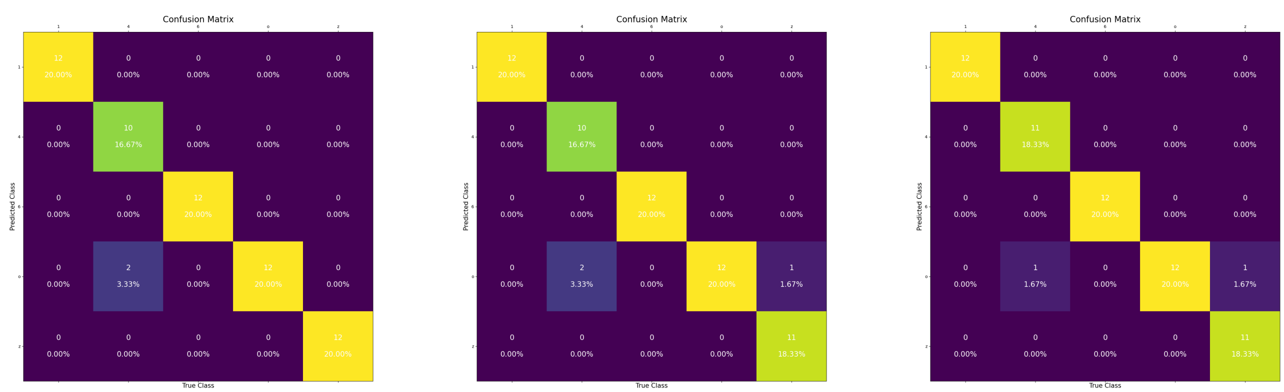


Figure 1: Confusion Matrix and ROCDET plots

ii With windows of sizes = 2, 4, 8, 12 and 16



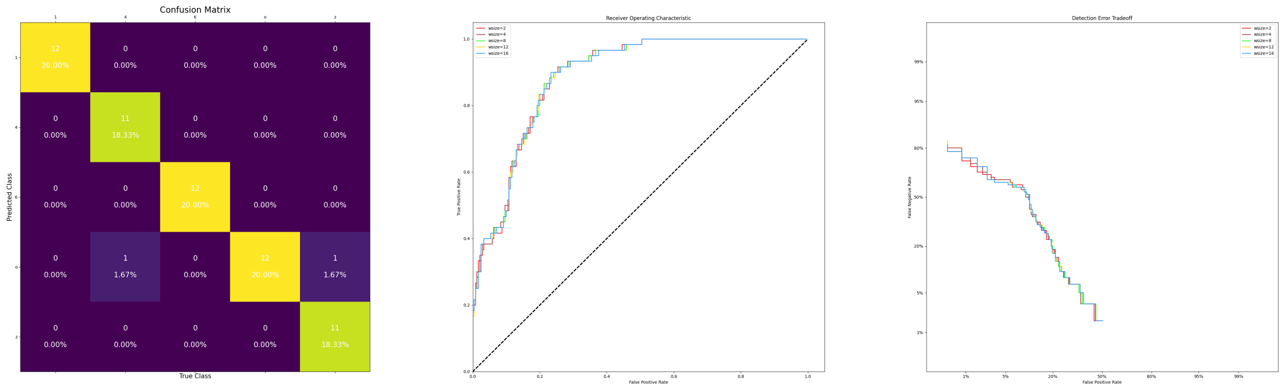


Figure 2: Confusion Matrices for wsize = 2, 4, 8, 16 and ROCDET plots

Analysis: We note that the ROC and DET curves are almost the same for all window sizes considered and from the confusion matrices, there is no big variation in accuracy when window size is increased from 2 to 16. The best confusion matrix we got was for DTW without windows but when windows were introduced, run-time improved.

b Dynamic Time Warping on Handwritten Characters Data(HCD)

i Without normalization

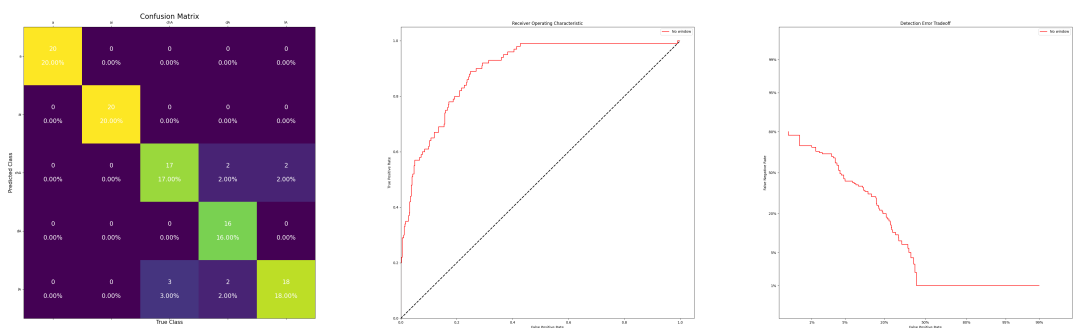


Figure 3: Confusion Matrix and ROCDET plots for DTW without windows

ii With normalization

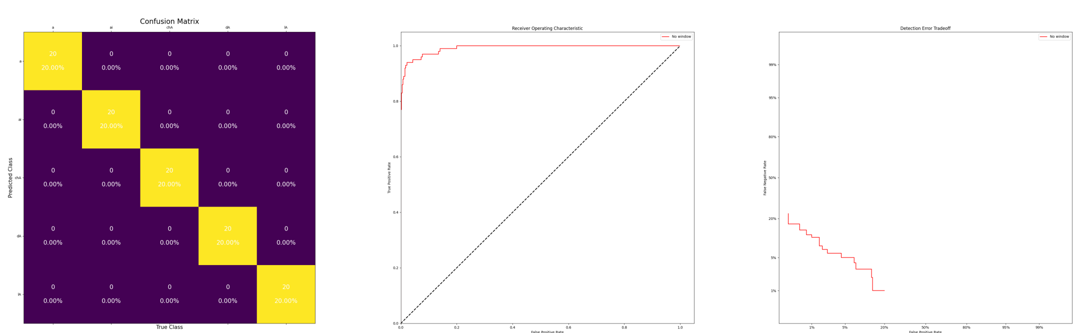


Figure 4: Confusion Matrix and ROCDET plots for DTW without windows

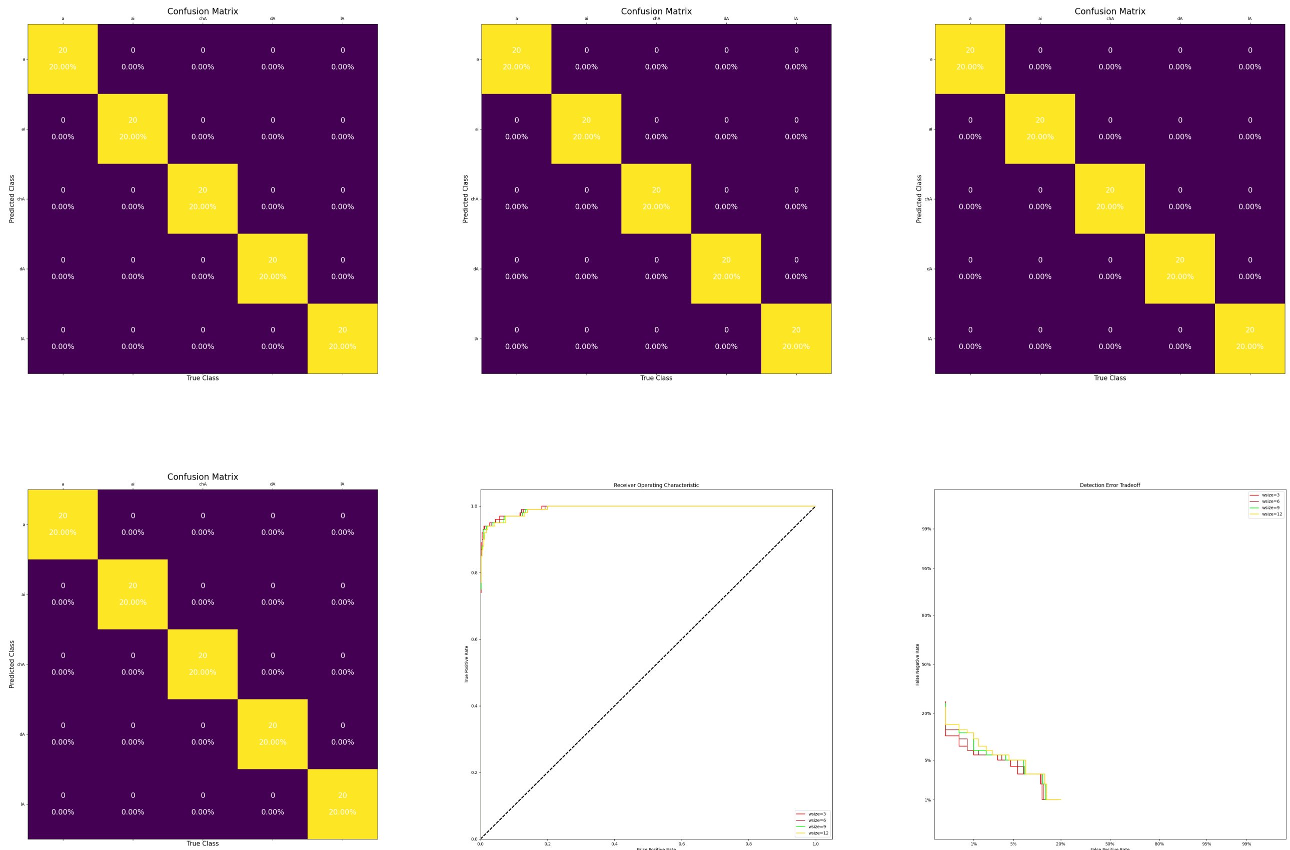
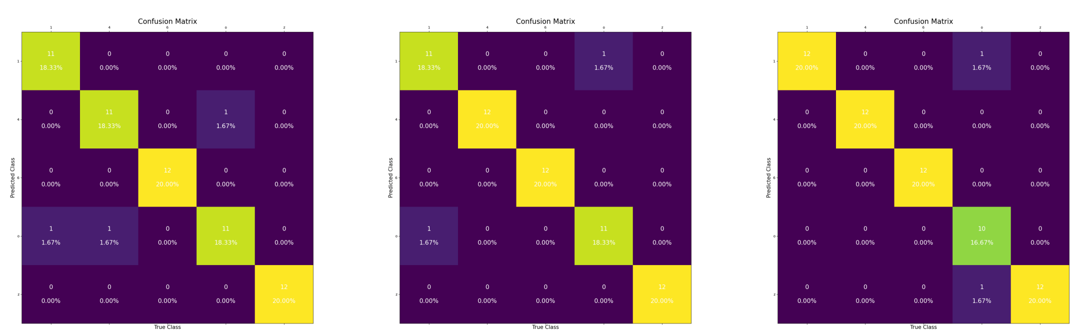


Figure 5: Confusion Matrices for wsize = 3, 6, 9, 12 and ROCDET plots

Analysis: For handwritten character data, the performance was really bad without normalization. However, when we did re-centering and re-scaling, there was a great improvement as shown above. Again, we used 20% = 15 lowest scores for score calculation. We can see that, with or without windows, the confusion matrices are all diagonal in nature and the accuracy is 100% after we do normalization.

c Hidden Markov Model on Isolated Spoken Digits(ISD)



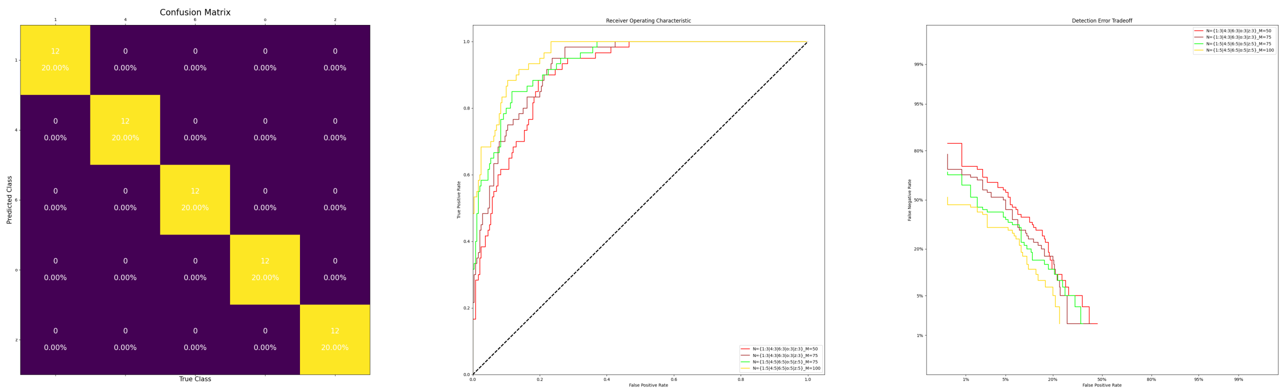
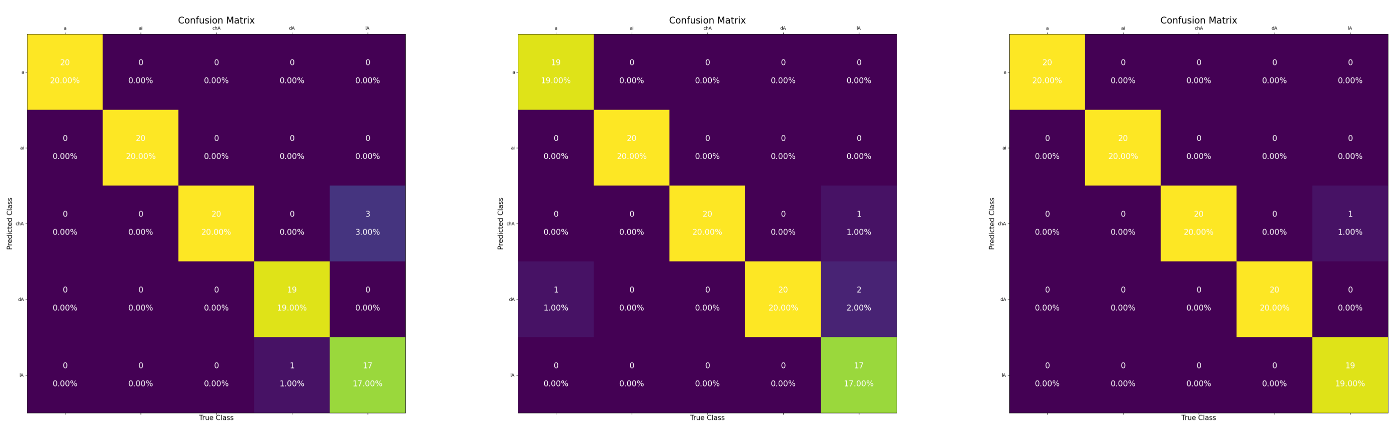


Figure 6: Confusion Matrices for N,M = 3,50; 3,75; 5,75; 5,100 and ROCDET plots

Analysis: We can see that by increasing the number of states from 3 to 5 (for all class HMMs) and by increasing the number of symbols from 50 to 75 (for N=3) or from 75 to 100 (for N = 5), the confusion matrix has improved and we got 100% accuracy when we took 5 states and 100 emission symbols. However, in other experiments we did, when we made N or M very large, the results were unpredictable. Hence, the determination of N and M is largely empirical in nature. A large M may result in no vectors being assigned to a cluster during KMeans and essentially adds an useless symbol in the codebook.

d Hidden Markov Model on Handwritten Character Data(HCD)

Given the exceptionally bad results obtained without normalization in DTW, we do normalization before building the HMMs. The results are given below.



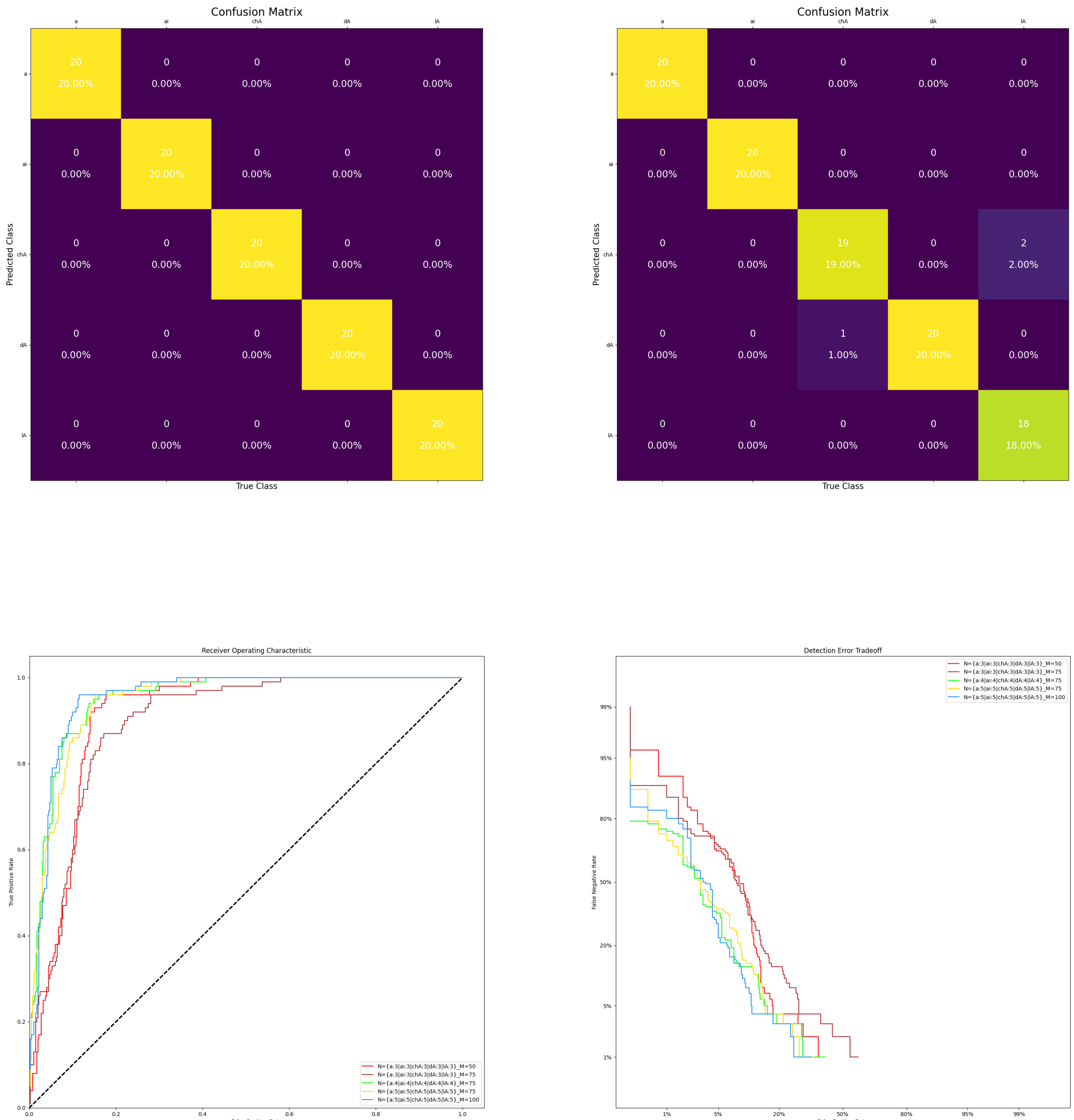


Figure 7: Confusion Matrices for N,M = 3,50; 3,75; 4,75; 5,75; 5,100 and ROCDET plots

Analysis: We can see that with an increase in the number of states from 3 to 4 and 4 to 5, the accuracy has generally improved. However, when we increased the number of symbols from 75 to 100 when the number of states was 5, the accuracy dropped (with respect to the confusion matrices). All these results, establish the empirical nature of N and M in the HMM framework.

e Concatenated HMMs to recognise Connected Spoken Digits

Method Used: We took the HMM models we generated for each class 1,4,6,o,z with number of states N = 5 and number of symbols M = 80 and concatenated them. The intra-HMM transition probabilities were already determined and we only needed to find the probability of transition from the last state of a digit's HMM to the first state of the next digit's HMM. We took this probability as **0.5** and built the concatenated HMM model file. We also need to save the codebook with the centroid co-ordinates in order to quantize the dev data and blind test data. We essentially built **125 HMMs for each possible 3-digit combination of the 5 digits and also the 25 HMMs for each possible 2-digit combination of the 5 digits**

Results :

1. Our model has an accuracy of **29.54 %** which indicates there is a tremendous scope for improvement. This also suggests that there might be a better technique for estimation of inter-digit transition probability.

2. There were some interesting predictions made as well. Our model made the following classifications - (61 -> 66), (1o -> 1o1), (46 -> 646), (zzz -> z6z) and many more. Here, (a -> b) denotes a vector series of true class a being classified as belonging to class b.

3. We can see that there is some similarity (prefix, suffix, same digits in different positions, different digits at a single position ...) between the true classes and predicted classes in most predictions.

4. Execution of our code will give predictions vs true class for all dev data and also the predictions for the blind data.

Blind Test Data Classification:-
The blind test data files were classified as shown below.

1. 1.mfcc -> z66

2. 2.mfcc -> z44

3. 3.mfcc -> oo4

4. 4.mfcc -> zz6

5. 5.mfcc -> z46