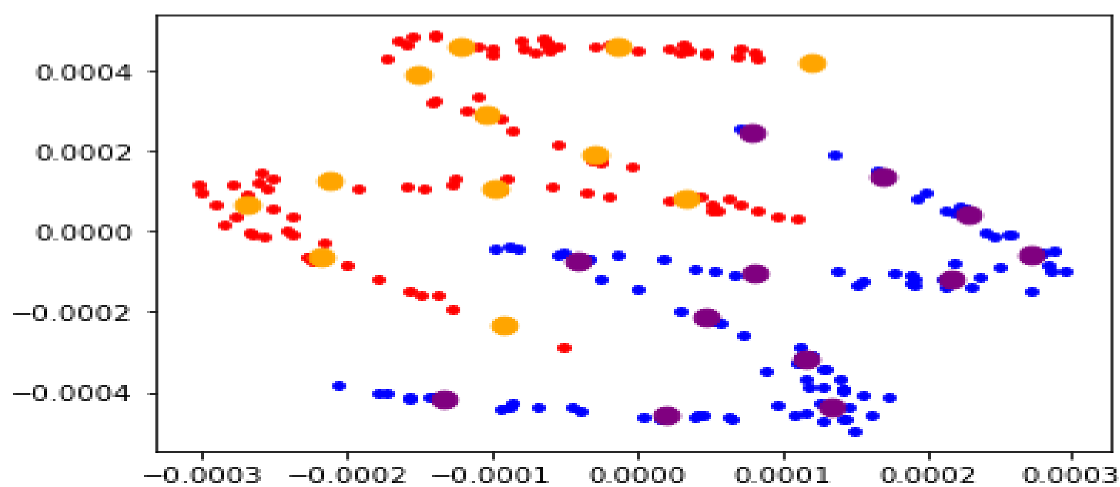
**PRML ASSIGNMENT - 3 REPORT**

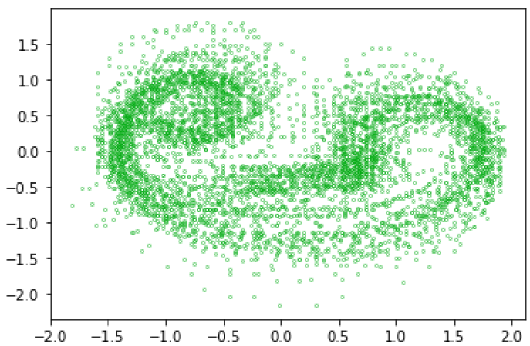
**Team 36, Jaya Kamal Maddula (CS19B028), Vedaant Alok Arya (CS19B046)**

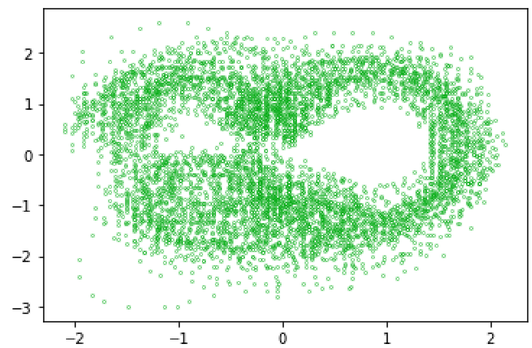
**GMM (Gaussian Mixture Models):** We couldn't get the algorithm to work due to numerical imprecisions in NumPy. To mitigate this, we attempted a complete switch to SymPy (which uses mpmath which has arbitrary precision), however the solution was too slow to be feasible.
We then attempted to use a mixed solution with NumPy handling the matrices and SymPy being used during the calculation of the probabilities, however the values tend to be so low that the trip back to NumPy still results in NaNs and logs-of-zero.
We also attempted to normalize the data, use extra factors in the initial covariance matrices, use only a particular region of the 36 regions of an image (hence reducing down to 23-dimensional data), etc but it still kept crashing. The normalization did help a little (training progressed further before crashing) but did not fix the issue.

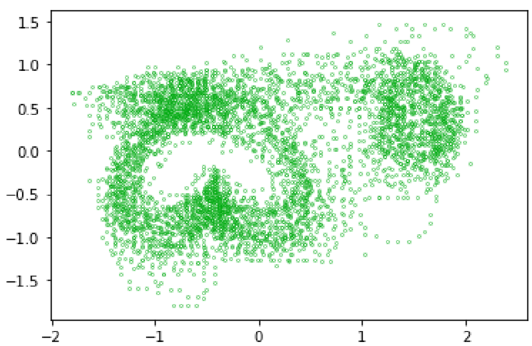


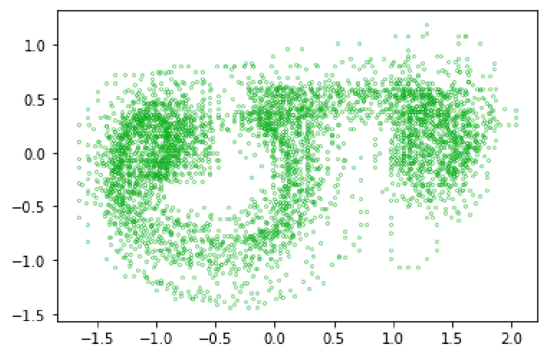
**DTW (Dynamic Time Warping) :** We used this algorithm to compare similarities between two temporal sequences. So for audio files after the mfcc files with feature vectors were given, we take the training data as a standard of reference and any new test data will be compared with the existing data to find optimal match with a minimum cost, the algorithm we use is almost similar to that of longest common subsequence.

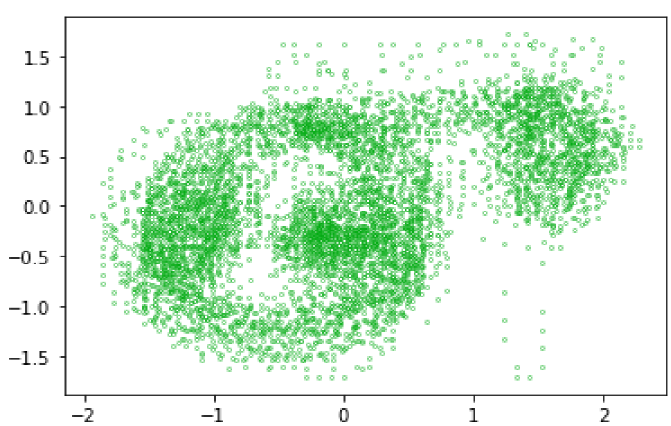
For the audio files no additional normalization was required as they were highly preprocessed. But the letter coordinates needed to be scaled and shifted accordingly so every dataset can be put in the same region. Both Z-score normalization and Min-Max normalization gave good results for the DTW model of the letters. Given below are the plots of the letter datasets after properly normalizing them with the Z-score normalization.



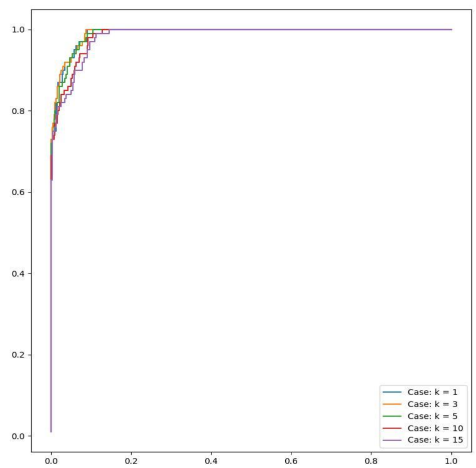



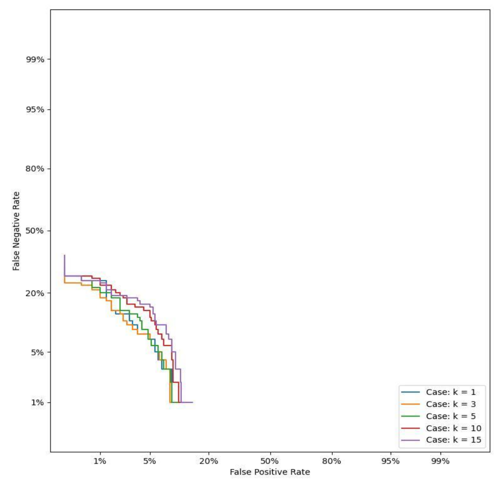



It is hard to plot the graphs for the audio files because of its multiple dimensions. The DTW model had 100% accuracy for the handwritten letter models but predicted 2 cases wrong in the audio file because of the distinctly different pronunciation from the rest of the pronunciations. ( a case of 9 resembles close to 1 and a case of 5 is incomplete.)

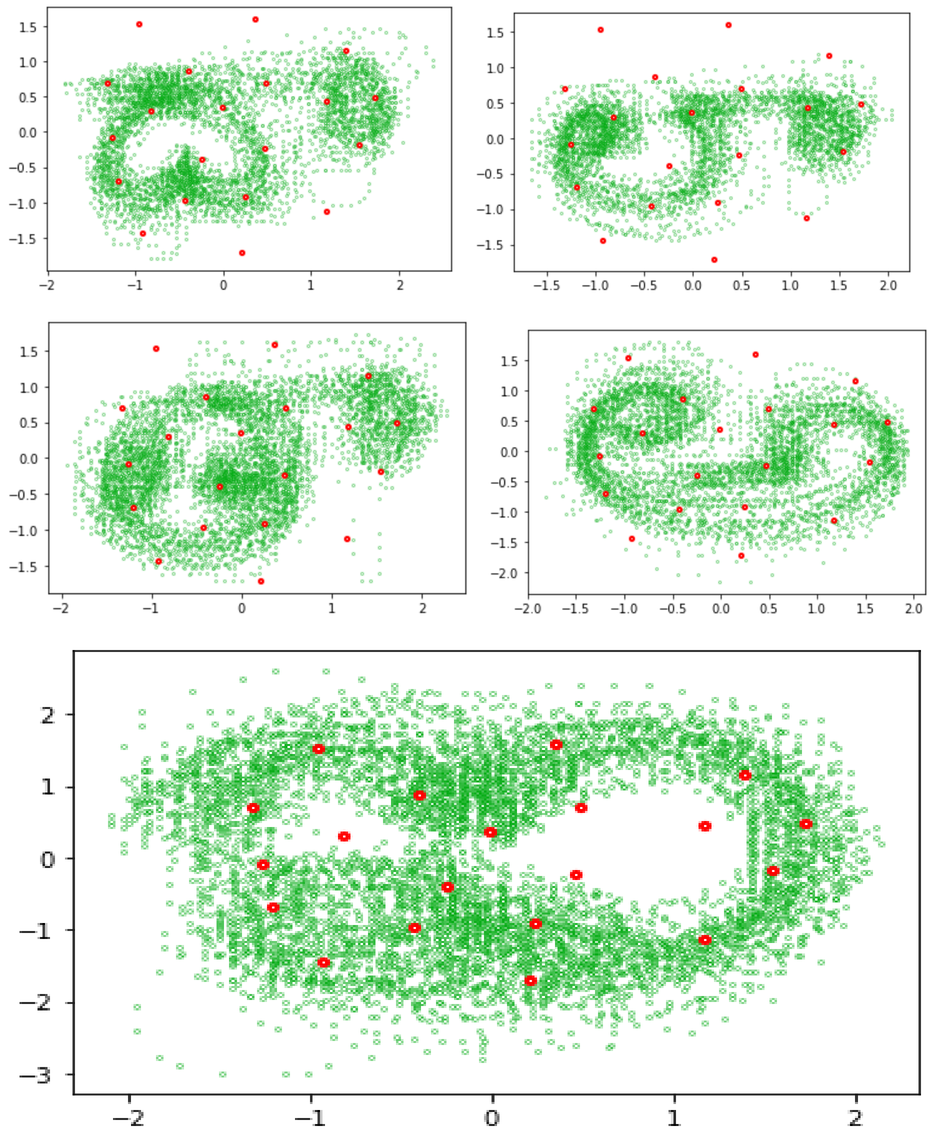
The DTW models are highly accurate and will have a great practical application. Here are the Det&Roc curves and confusi for the handwritten letter models.

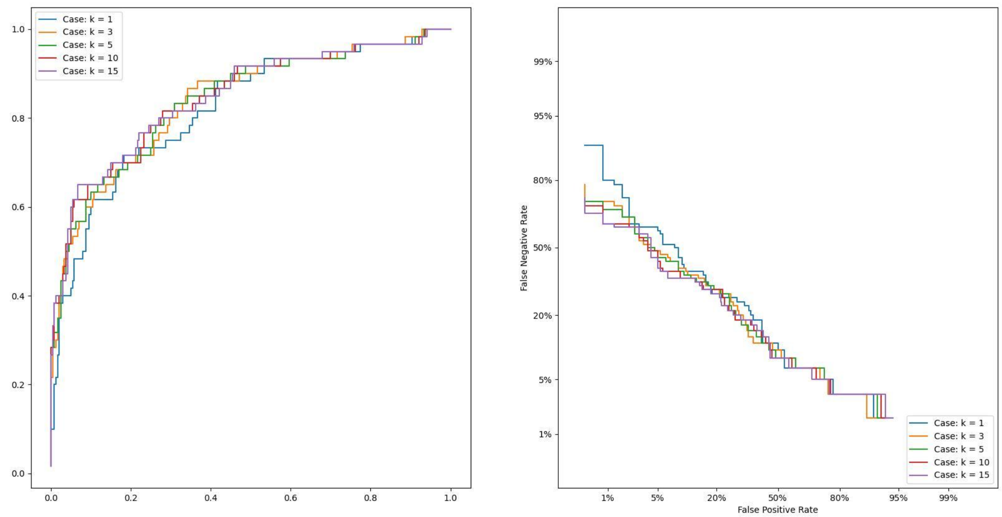


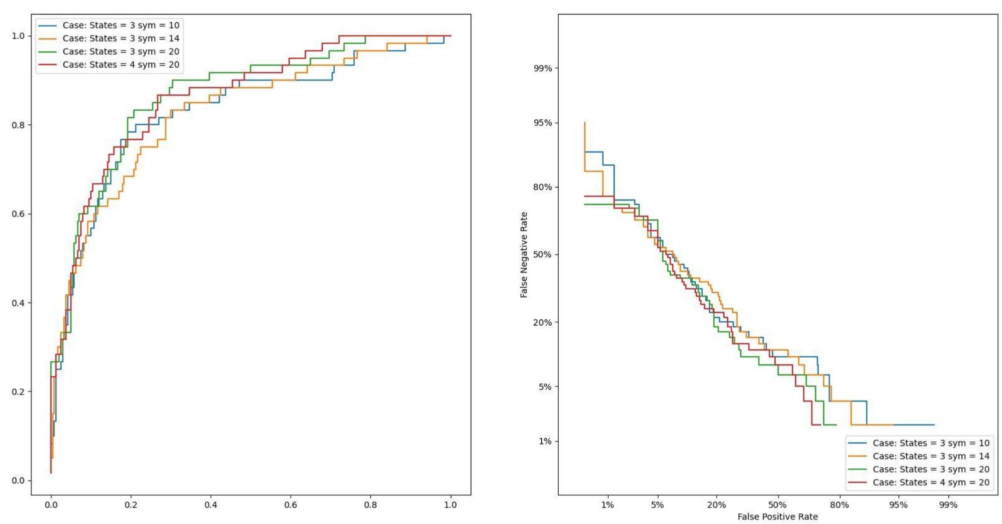




| | 1 | 2 | 3 | 4 | 5 | total % correct |
|---|---|---|---|---|---|---|
| 1 | 20.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 2 | 0.0 | 20.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 3 | 0.0 | 0.0 | 20.0 | 0.0 | 0.0 | 100.0 |
| 4 | 0.0 | 0.0 | 0.0 | 20.0 | 0.0 | 100.0 |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 20.0 | 100.0 |
| total % correct | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

And for the HMM models, they are faster than the DTW models, but they are less accurate than the DTW models and also need a lot of hypertuning unlike the DTW models. First an optimal K must be chosen for the clustering and then optimal number of states for better accuracy. So it requires a lot of hypertuning. Given below are the images of the letter mappings with cluster centers on them.






The DET and ROC curves for the HMM models are given above. The value of k increase until we reach the least error and then if we increase k anymore it would cause more error.