CS5691 Assignment 3

Amogh Patil (EE19B134), Rishabh Adiga (EE19B135)

April 2022

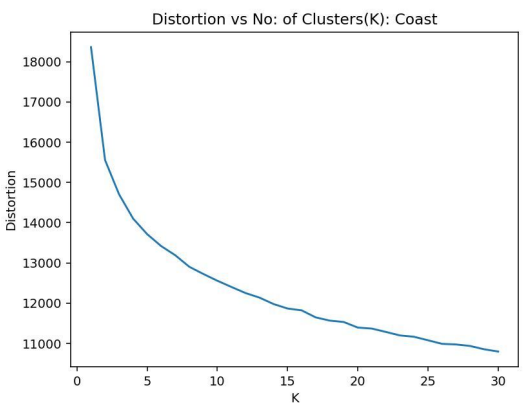
1 K-Means and GMM

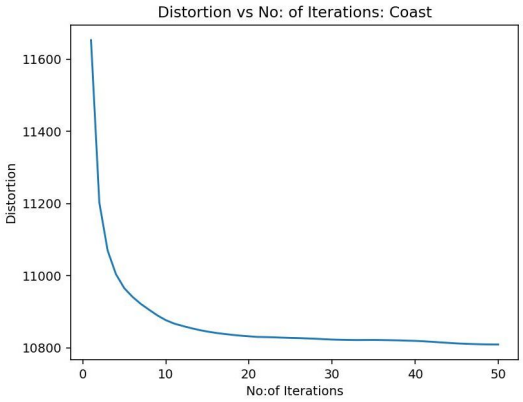
1.1 Introduction

We are given 2 datasets, the first dataset contains a set of images of coasts, forests, highways, mountains and open country. Each image is divided into 6*6(36) blocks. A 23 dimensional feature vector is extracted from each block. Features include colour histogram, edge-directed histograms, and Entropy of wavelet coefficients. The second dataset consists of two spirals wound around each other, with each spiral belonging to a class. The features here are the two dimensional coordinates of the spiral. The test example is a 2-D co-ordinate which needs to be classified as whether it belongs to class1 or class2.
We aim to implement a Gaussian Mixture Model(means initialised using K means algorithm) to classify given dev data. Various Extrinsic parameters are altered and their corresponding effects are observed.

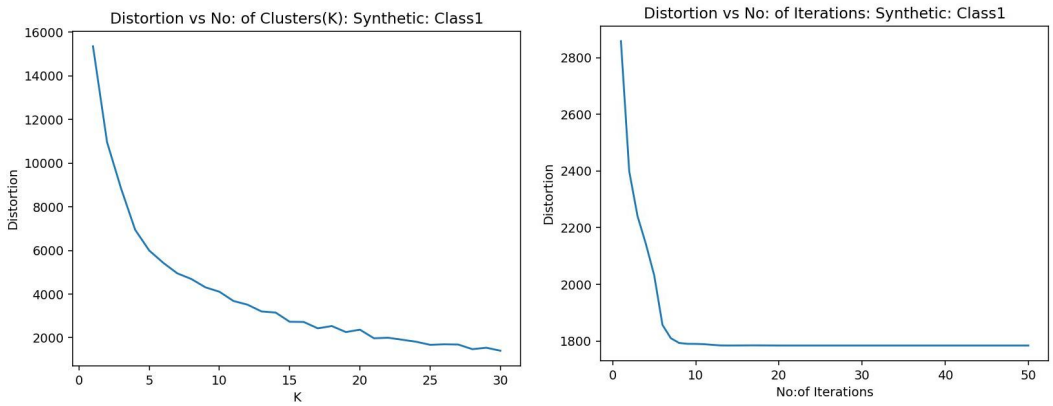
1.2 K-Means Algorithm

K-means algorithm is an unsupervised technique which aims to cluster the set of feature vectors into K groups. The means of the K groups are randomly initialised. Then each data point is assigned to the cluster of the nearest mean. The mean is again recomputed for each cluster and these steps are repeated for n iterations.
The distortion vs K and distortion vs number of E-M iterations plots are shown below for both datasets(plots for 1 class of both datasets are shown here):





Based on this plot, we take **K as 15** and **number of iterations as 30**



Based on this plot, we take **K as 25** and **number of iterations as 13**
These means and cluster assignments are used as the initializations for Gaussian mixture models.

1.3 Gaussian Mixture Model

This is an algorithm that is similar to the K means, except that it is a soft clustering algorithm. Each mixture has some contribution in describing each and every data point. The algorithm is divided into Expectation and Maximisation Steps. In the Expectation step the r_nk values are computed, i.e the contribution that the n data point belongs to the mixture k. In the Maximisation Step, the mean, covariance and weight of each mixture is computed. These steps are run for n no of iterations.
Also, we have **normalised all the data** by subtracting means of the features and then dividing by the range of the corresponding features.Additionally,we have applied **PCA** which preserves 99% variance directions and reduces the dimensions to 11 hence improving speed and numerical accuracy.

1.4 Image Classification

A test image is given which is decomposed into 36 blocks, with each block forming a 23 dimensional feature vector. The probability of a feature vector belonging to a class is computed by taking the weighted contribution of the probability from each Gaussian mixture of the class.

$$P(x/W_i) = \sum_{i=1}^{k} \pi_i * N(x/u_i, \epsilon_i) \tag{1}$$

$$P(Image/W_i) = (\Pi_{j=1}^{23} P(x_j/W_i)) * P(w_i) \tag{2}$$

We don't directly multiply the probabilities, but rather take log and add the values along with the priors.

$$log(P(Image/W_i)) = \sum_{j=1}^{23} log(P(x_j/W_i) + log(P(w_i) \tag{3}$$

1.4.1 Final models for image dataset

```
Accuracy for Non-Diagonal-Matrix = 68.79310%
Accuracy for Diagonal Matrix = 74.6671%
```

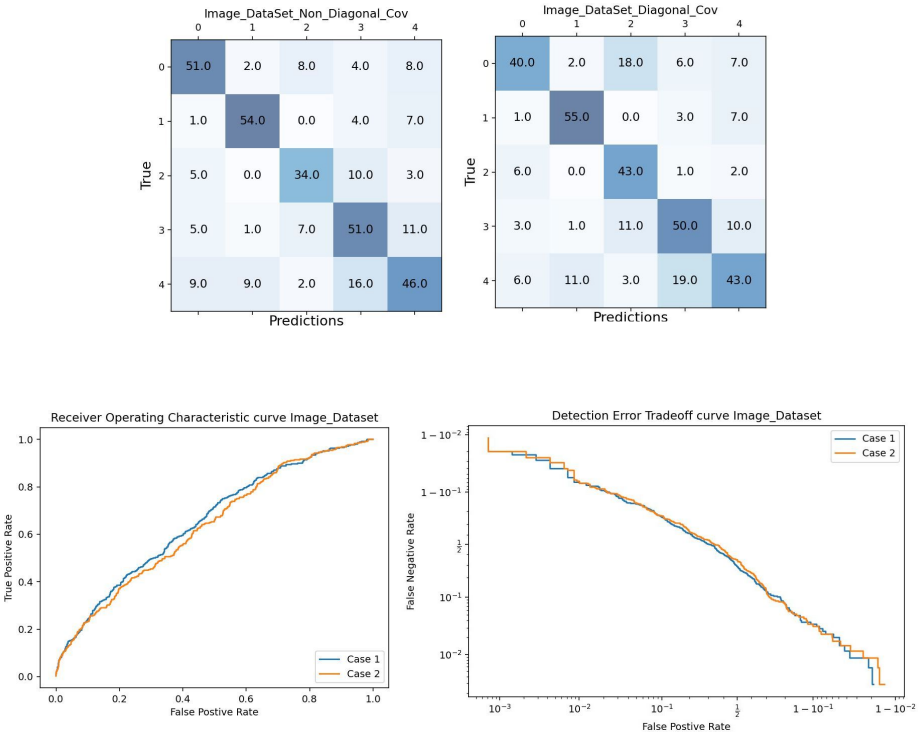


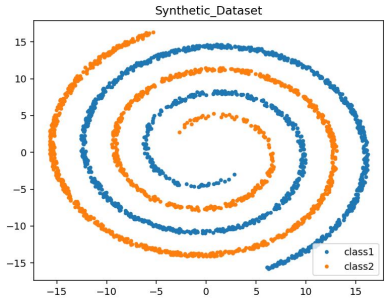
Figure 1: Case1 : Non_Diagonal Matrix, Case 2 : Diagonal Matrix

From the ROC and DET curves it is clear that the Diagonal Matrix does better at classification. Since it has lower number of parameters which prevents overfitting of data.

We take K = 15 and not K>20, as this prevents overfitting on the training data set, K-Means iterations = 30.

1.5 Synthetic Data

The provided dataset consists of 2 classes with 2 features which are both spirals and the plot of the data is shown below:



1.5.1 Diagonal and Non Diagonal Covariance Matrices for different K

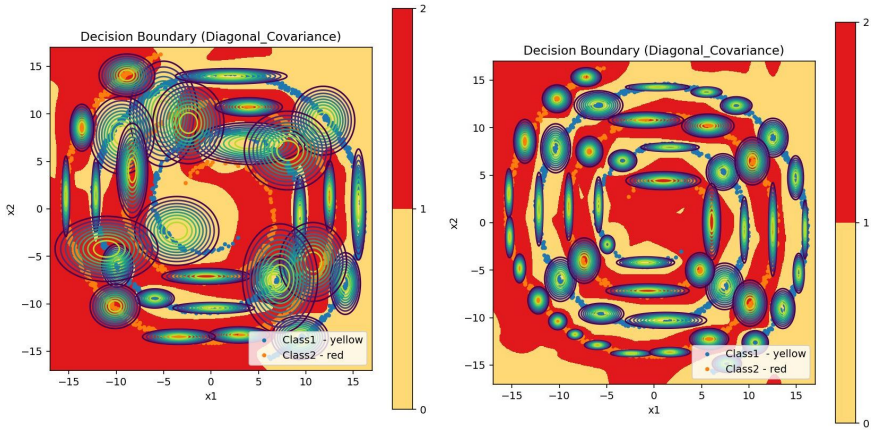


Figure 2: K = 15 and K=25 (Diagonal cov)

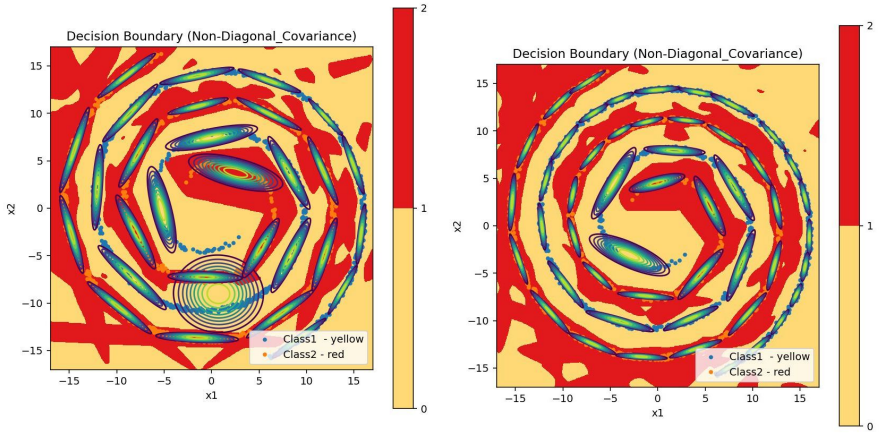


Figure 3: K = 15 and K=25(Non Diagonal cov)

- When small values of K(for example K=15) are used ,due to small number of clusters the model finds it hard to cover the entire spiral ,resulting in large gaps between 2 clusters and this results in the means of the outer loops of the spirals shifting inward or vice versa in order to account for these gaps.This is bad as the clusters are entering the regions of the other class.This is evident in the plots on the left side.
- For larger values of K(for example K=30) we get gaussians that are covering the entire spiral in sequence without shifting inward or outward and very high accuracy in classification.
- The decision boundaries show that the regions were the train data is present has been classified perfectly but other regions are classified based on the alignments of the gaussians in the plane.
- We can also see that the diagonal covariances results in gaussians being aligned parallel to the axes.For K>=20, both diagonal and non diagonal perform extremely well.

1.5.2 Final Models for synthetic dataset

The final model for synthetic dataset used K = 25 for both diagonal and non diagonal covariance matrices.The metrics are shown below:

```
Accuracy for Non-Diagonal-Matrix = 100%
Accuracy for Diagonal Matrix = 98.8%
```

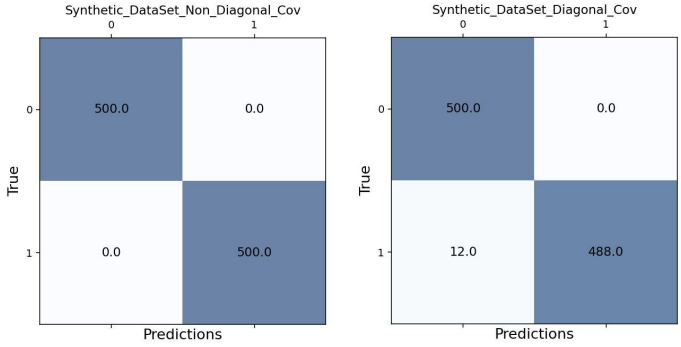


Figure 4

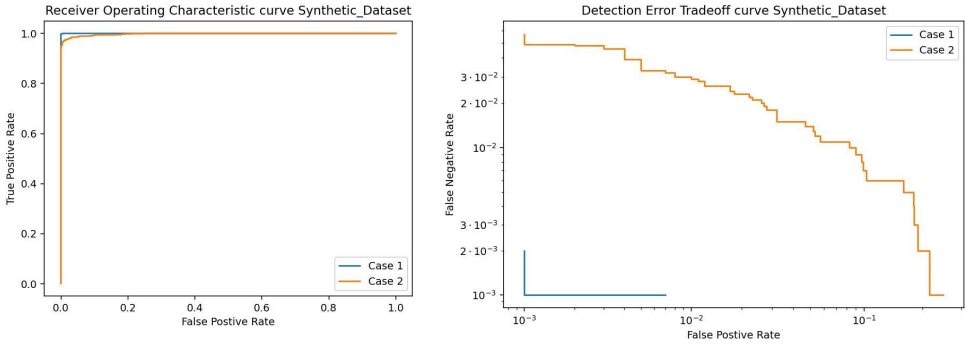


Figure 5

From the ROC and DET we see that the Non diagonal matrices performs better. We take K = 25, K-Means iterations = 13

2 DTW and HMM

2.1 Introduction

Given 2 datasets where one dataset consists of spoken utterances in the form of **MFCC features** and the other dataset consists of features of **handwritten Telugu characters** , our aim was to perform Dynamic Time Warping and implement a Hidden Markov Model for classification of this data.We observe the effect of various hyperparameters and compare the performance metrics of the models and their runtimes too.

2.2 Experiments and Inferences

DTW finds distance values between a given test sample and all train samples of all classes and can use either averages of distances or top K voting or min values from each class to classify the given test example(smaller the distance, the more it matches with that train sample).HMM code trains doubly probabilistic state and symbol based models for each class and using K-Means on all data to create the codebook and hence the symbols for our model.The probability of a given test sequence being generated from each of these models(1 for each class) is calculated and the one with highest probability is used for classification.
Some info about the files for HMM:

- **HMM_Symbol_Generation.py** is the python file used to generate symbols(**no need to run** this since the symbols are already generated and the required files are already present in the submission).
- **train_and_test_model.py** is the python file that **uses all the cpp files** for training the HMMs using terminal commands that are passed through this py file itself **using**

subprocess library.After training the models, this file does testing on dev data too and outputs accuracy,predictions,ROC,DET,Confusion matrix,etc.(This is the only file that needs to be run for checking code)

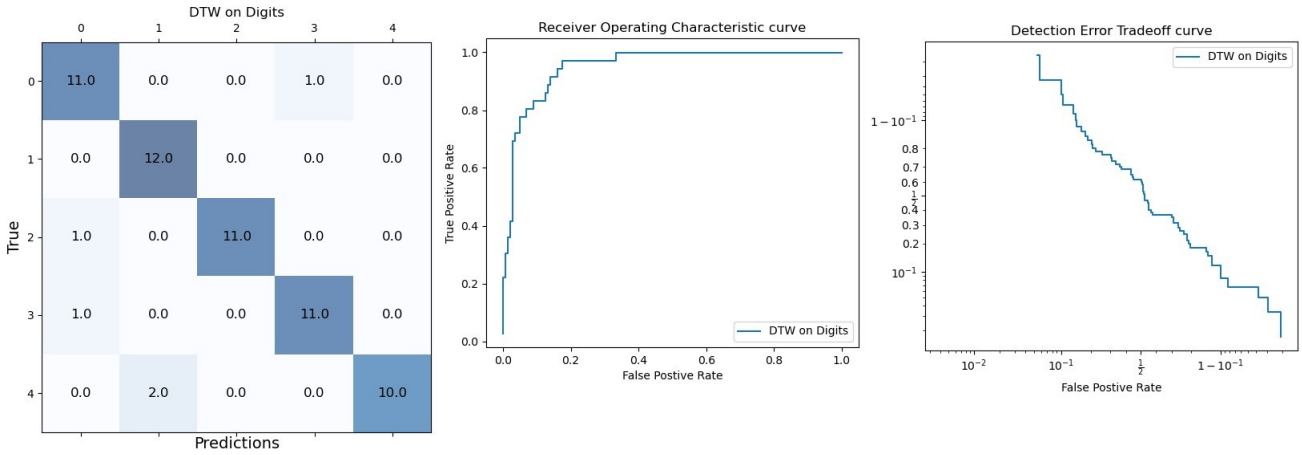
2.2.1 Isolated Spoken-Digit Dataset

The given 38 MFCC features for each audio clip of variable frame lengths is used by our code for DTW and HMM with no additional modifications.

a)**DTW**
We use euclidean distance as the distance based similarity measure between 2 feature vectors.Accuracy on dev data,Predictions,Confusion matrix,ROC and DET plots are calculated and this has been shown below:

```
Accuracy = 91.66 %
Predicted_classes = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0], [1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1], [2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2], [3, 0, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3], [4, 4, 4, 4,, 2, 2], [3, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3], [4, 4, 4, 4, 4, 1, 4, 4, 4, 4, 1, 4]]

Runtime without numba library for efficient machine operations = 15mins (approx)
Runtime with numba library for efficient machine operations =  63.048 seconds
```

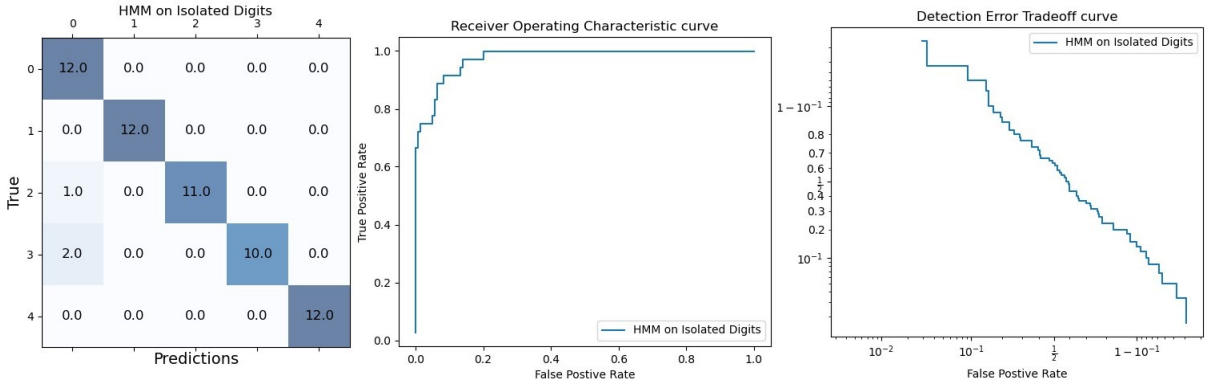


b)**HMM**
For running the given cpp code and training a HMM for a given class, we need to provide the symbol sequence of all the train examples of that class.These symbol sequence are obtained by applying K-Means clustering on all the feature vectors of all train examples of all classes and hence forming a code book which results in K symbols.

- The value of K (Number of symbols) chosen was 44 and this was chosen because there are **44 phonemes**.
- Phoneme refers to any of the **perceptually distinct units of sound** in a specified language that distinguish one word from another and hence to represent any of these , we need quite a few symbols.

Accuracy on dev data,Predictions,Confusion matrix,ROC and DET plots are calculated and this has been shown below:

```
Accuracy = 95.0%
Predicted_classes = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1], [2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2], [3, 3, 3, 3, 3,
0, 0, 3, 3, 3, 3, 3], [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]]
```



2.2.2 Online Handwritten-Character Dataset

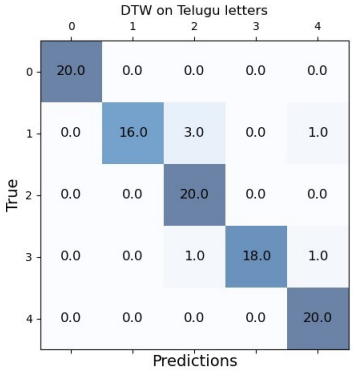
Here, we are given the features in the form of x,y coordinates of the character sequence.Training can be done directly on this data but we have added the additional feature of slope information of these characters.

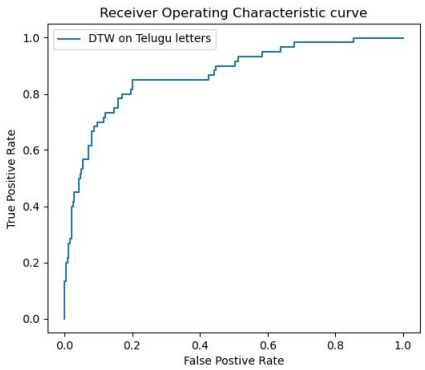
- The **3rd feature added is slope information** in the form of $tan^{-1}(slope)$ which is basically the angle between the tangent at the given point and x axis in radians.
- The reason this angle was used over the actual slope values is that slope values range from anywhere in between -∞ and +∞ and this results in extremely large values for the last feature compared to the first 2 features.
- Since slope feature results in large values,the main feature that is optimized is the slope itself giving very less weight to x,y coordinates.For example, in DTW since we are taking eucledian distances,a huge fraction of this distance value will come from the difference in slopes and this results in bad results in classification.
- This was experimented and using slopes directly resulted in **worse classification of dev data** compared to not using it as a feature.Instead of slope, as explained earlier , when the angle in radians($tan^{-1}(slope)$) is used, it results in **much better classification**(Accuracy values shown later in the report)

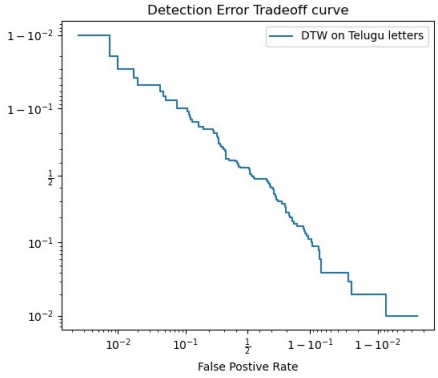
a)**DTW**
Accuracy on dev data,Predictions,Confusion matrix,ROC and DET plots are calculated and this has been shown below:

```
Classification Accuracy without angle(radians) feature = 77.0%
Classification Accuracy with angle(radians) feature = 94.0%
Predicted_classes = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [2, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1], [2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], [3, 4, 3, 3, 3, 3, 3,
3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3], [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4]]

Runtime without numba library for efficient machine operations = 20mins (approx)
Runtime with numba library for efficient machine operations =  80.393 seconds
```





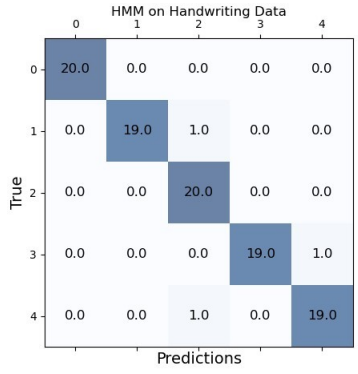


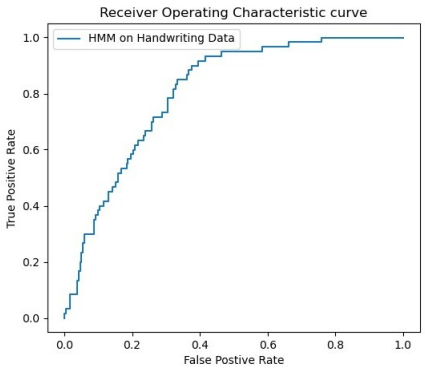
b)**HMM**

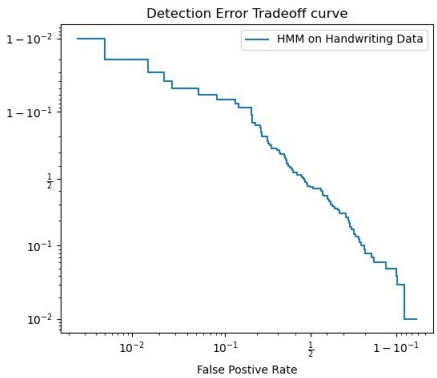
- We apply K-Means on all the features for this dataset too and obtain K symbols.
- The value of K (Number of symbols) chosen was**10** and this was chosen by seeing the number of **different symbols or patterns across all the letters** we were given to train on.
- The number of states were picked by looking at the relative complexities of writing these letters.

Accuracy on dev data,Predictions,Confusion matrix,ROC and DET plots are calculated and this has been shown below:

```
Classification Accuracy without angle(radians) feature = 86%
Classification Accuracy with angle(radians) feature = 97%
Predicted_classes = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [1, 1, 1, 1, 2, 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, 2, 2, 2, 2, 2, 2, 2], [3, 3, 3, 4, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3], [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 2, 4, 4, 4, 4, 4]]
```







2.2.3 Inferences

- DTW has a longer runtime than HMM for both datasets if we do not use libraries like numba for improved speeds.This makes HMM a better choice for classification of this type of data.
- Hidden Markov models are giving better dev data accuracy(percentage correctly classified) than Dynamic Time Warping based classification for both datasets.
- One major disadvantage of HMMs is the number of emperical parameters that we need to tune in order to get a good model but for DTW this is avoided.

- Although DTW is worse than HMM in accuracy, it has better ROC and DET curves and this is because, HMMs are more sensitive to the parameters and have optimal scores at very specific parameter values.
- The slope information in the form of angle between the tangent at that point and x axis is a very feature in classifying letters in a language and results in around 10 to 15% increase in classification accuracy.