Principal Component Analysis, Fisher Discriminant Analysis, Perceptron and SVM classifier

Akash Agrawal, B15303, IIT Mandi

Group No: 2

Abstract

This report presents theoretical analysis of the results obtained on classification of various datasets using Gaussian Mixture Model after reducing the dimension of data using Principal Component Analysis, and Fisher Discriminant Analysis. This report then covers the results of classification of data using Perceptron and SVM classifier.

1 Introduction

Dimensionality Reduction

Principal Component Analysis(PCA):

In recognizing pattern in data and classify or predict the pattern in the data we need extract features of the data and represent them in some form to be feed into the model. These feature vectors are generally in the form of an array. As the dimensionality of the data increases the complexity in estimation of parameters for the pattern recognition model increases. It therefore generates the need of reduction of dimension of feature vector such that the information content lost is minimum. PCA is a well-known algorithm for dimensionality reduction.

PCA Algorithm:

Step 1:
Represent the data in mean subtraction form (i.e., subtract the mean vector from all the data points so that the mean shifts to the origin).

Step 2:
Calculate the covariance matrix of the original data (let it be Σ).

Step 3:
$\lambda_i q_i = \Sigma q_i$

Now do eigen analysis and get the corresponding eigen values and eigen vectors.

Step 4:
Sort the eigen values according to the values in decreasing order and take l highest values from it (where l is the dimension you want to reduce into).

Step 5:
Now dot product all the feature vectors one by one with the l eigen vectors. The l values we get represents the new l dimension feature vector for that data.

Fisher Linear Discriminant Analysis(FDA):

FDA is used to find a single direction of projection of the data such that the separability of the projected data is maximum. In FDA we try to find a direction in which the separation of the mean of the data is maximum and the variance of the data is minimum. It is a binary Discriminant Analysis technique.

FDA algorithm:

Step 1:

First we calculate,

$S_W = S_+ + S_-$ where S_+ , S_- are scatter matrix of two classes.

$S_B = (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T$ where μ_+ , μ_- are the means of two classes.

Step 2:

$S_w^{-1} S_B w = \lambda w$

Now perform eigen analysis and get the value λ_{max}. The eigen vector corresponding to λ_{max} is the direction of maximum separation.

Discriminative learning Technique

Perceptron Learning:

Perceptron is an algorithm for supervised learning of binary classifier. The data which is fed into the classifier should be linear. It is a linear classifier and will converge only when the data is linear.

Perceptron Learning Algorithm:

Consider the equation of line: $g(\overline{x}) = \overline{w}^T\overline{x} + w_o = 0$

Step 1:

Initialise the values of $\overline{w}$ and w_o η (learning rate).

Step 2:

Now repeat this step until the D_M (set of misclassified data) is empty.

$\overline{a}(k+1) = \overline{a}(k) + \eta \sum_{Xn \in Dm} Yn * Zn$

where $\overline{a}(k+1) = [w_0, w_1, w_2, \dots ,w_d]$ = line parameters in $(k+1)^{th}$ iteration,

X_n = data point $[x_1, x_2, \dots ,x_d]$,

Y_n = class label for X_n data point,

$Z_n = [1, x_1, x_2, \dots ,x_d]$,

η = learning rate,

D_m = set of misclassified examples

Limitations:

- Data should be linearly separable or otherwise it will not converge.
- No method for good initialization of parameter values.

Support Vector Machine (SVM):

SVM is a non-probabilistic binary classifier. It needs no information nor make any assumptions about how the data of a class is distributed. It constructs a maximum margin hyperplane unlike Perceptron Learning where we only get a separating plane without any notion of margin. SVM is also used for Non-linearly separable data.

SVM's work by transforming the data into a higher dimension space where it is linearly separable and then constructing a hyperplane in that dimension for classification, regression and other pattern classification tasks. This is achieved using various kernel functions. Using these Kernel functions we actually need not to go into that higher dimensions and by computing only pairwise-inner products from the original dimension we can classify data. This is called **Kernel Trick**.

There are various kernel functions which does this are:

- Linear Kernel:
  $K(X_m, X_n) = X_m^T X_n$
- Polynomial Kernel:
  $K(X_m, X_n) = (aX_m^T X_n + b)^P$
- Gaussian Kernel:
  $K(X_m, X_n) = \exp(-\frac{|Xm-Xn\ |**2}{\sigma**2})$

Assumptions

- As we are not interested in playing with the parameters of SVM Kernel, so I have assumed some constant values for the parameters of Kernel:
  a) Linear Kernel: C=1.0
  b) Polynomial Kernel: degree=3, C=1.0
  c) Gaussian Kernel: gamma=1.0, C=1.0

2 Results and Inferences for Principal Component Analysis

i) <u>GMM Mixtures: 1</u>

a) <u>Dimension</u> = 1

[[45, 33, 43], [3, 12, 5], [2, 5, 2]]
Accuracy: 0.39
Precision for class 0 : 0.37
Recall for class 0 : 0.9
F-measure for class 0 ; 0.53
Precision for class 1 : 0.6
Recall for class 1 : 0.24
F-measure for class 1 ; 0.34
Precision for class 2 : 0.22
Recall for class 2 : 0.04
F-measure for class 2 ; 0.06
Mean Precision: 0.39
Mean Recall: 0.39
Mean F-measure: 0.31

b) <u>Dimension</u> = 4

[[44, 31, 32], [5, 12, 7], [1, 7, 11]]
Accuracy: 0.44666666666666666
Precision for class 0 : 0.411214953271028
Recall for class 0 : 0.88
F-measure for class 0 ; 0.5605095541401274
Precision for class 1 : 0.5
Recall for class 1 : 0.24
F-measure for class 1 ; 0.32432432432432434
Precision for class 2 : 0.5789473684210527
Recall for class 2 : 0.22
F-measure for class 2 ; 0.3188405797101449
Mean Precision: 0.49672077389736025
Mean Recall: 0.4466666666666667
Mean F-measure: 0.4012248193915322

c) Dimension = 12
[[41, 26, 14], [8, 21, 5], [1, 3, 31]]
Accuracy: 0.62
Precision for class 0 : 0.5061728395061729
Recall for class 0 : 0.82
F-measure for class 0 ; 0.6259541984732825
Precision for class 1 : 0.6176470588235294
Recall for class 1 : 0.42
F-measure for class 1 ; 0.5
Precision for class 2 : 0.8857142857142857
Recall for class 2 : 0.62
F-measure for class 2 ; 0.7294117647058823
Mean Precision: 0.6698447280146627
Mean Recall: 0.62
Mean F-measure: 0.6184553210597216

d) Dimension = 24
[[45, 39, 13], [1, 4, 4], [4, 7, 33]]
Accuracy: 0.5466666666666666
Precision for class 0 : 0.4639175257731959
Recall for class 0 : 0.9
F-measure for class 0 ; 0.6122448979591837
Precision for class 1 : 0.4444444444444444
Recall for class 1 : 0.08
F-measure for class 1 ; 0.13559322033898308
Precision for class 2 : 0.75
Recall for class 2 : 0.66
F-measure for class 2 ; 0.702127659574468
Mean Precision: 0.5527873234058801
Mean Recall: 0.5466666666666667
Mean F-measure: 0.483321925957545

e) Dimension = 28
[[47, 43, 15], [0, 1, 1], [3, 6, 34]]
Accuracy: 0.5466666666666666
Precision for class 0 : 0.44761904761904764
Recall for class 0 : 0.94
F-measure for class 0 ; 0.6064516129032259
Precision for class 1 : 0.5
Recall for class 1 : 0.02
F-measure for class 1 ; 0.038461538461538464
Precision for class 2 : 0.7906976744186046
Recall for class 2 : 0.68
F-measure for class 2 ; 0.7311827956989247
Mean Precision: 0.579438907345884
Mean Recall: 0.5466666666666667
Mean F-measure: 0.4586986490212297

f) Dimension = 40
[[35, 35, 25], [3, 9, 0], [12, 6, 25]]
Accuracy: 0.46
Precision for class 0 : 0.3684210526315789
Recall for class 0 : 0.7
F-measure for class 0 ; 0.48275862068965514
Precision for class 1 : 0.75
Recall for class 1 : 0.18
F-measure for class 1 ; 0.2903225806451613
Precision for class 2 : 0.5813953488372093
Recall for class 2 : 0.5
F-measure for class 2 ; 0.5376344086021505
Mean Precision: 0.5666054671562627
Mean Recall: 0.45999999999999996
Mean F-measure: 0.43690520331232235

g) Dimension = 50
[[43, 36, 45], [3, 4, 0], [4, 10, 5]]
Accuracy: 0.3466666666666667
Precision for class 0 : 0.3467741935483871
Recall for class 0 : 0.86
F-measure for class 0 ; 0.4942528735632184
Precision for class 1 : 0.5714285714285714
Recall for class 1 : 0.08
F-measure for class 1 ; 0.14035087719298248
Precision for class 2 : 0.2631578947368421
Recall for class 2 : 0.1
F-measure for class 2 ; 0.14492753623188404
Mean Precision: 0.39378688657126687
Mean Recall: 0.3466666666666667
Mean F-measure: 0.2598437623293616

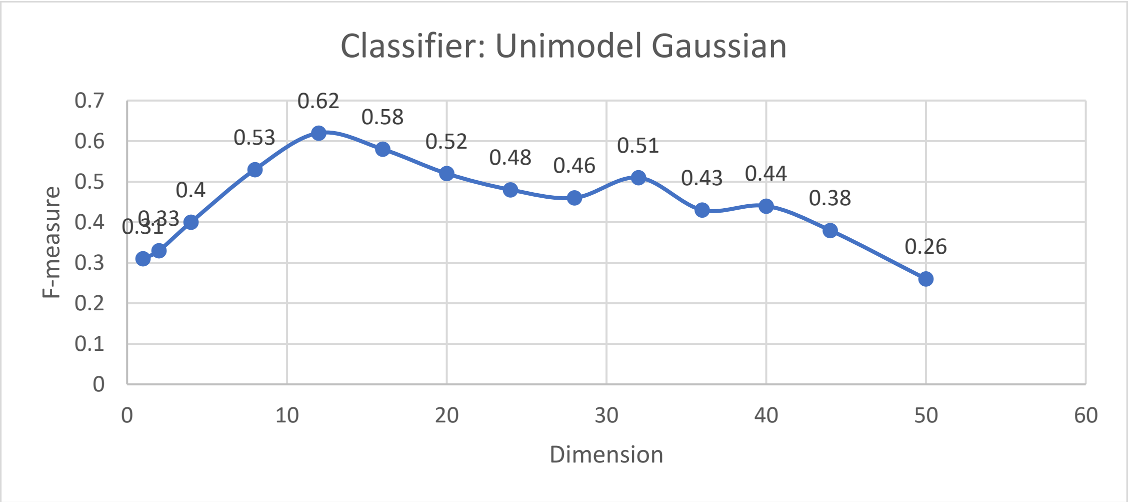

Figure: Variation of F-measure when data reduced to different dimensions using PCA

ii) GMM mixtures = 2

a) Dimension = 1

[[0, 3, 4], [47, 42, 32], [3, 5, 14]]
Accuracy: 0.37333333333333335
Precision for class 0 : 0.0
Recall for class 0 : 0.0
Precision for class 1 : 0.34710743801652894
Recall for class 1 : 0.84
F-measure for class 1 ; 0.4912280701754386
Precision for class 2 : 0.6363636363636364
Recall for class 2 : 0.28
F-measure for class 2 ; 0.3888888888888889
Mean Precision: 0.32782369146005513
Mean Recall: 0.37333333333333335
Mean F-measure: 0.2933723196881092

b) Dimension = 4

[[43, 29, 33], [3, 8, 7], [4, 13, 10]]
Accuracy: 0.4066666666666667
Precision for class 0 : 0.4095238095238095
Recall for class 0 : 0.86
F-measure for class 0 ; 0.5548387096774193
Precision for class 1 : 0.4444444444444444
Recall for class 1 : 0.16
F-measure for class 1 ; 0.23529411764705882
Precision for class 2 : 0.37037037037037035
Recall for class 2 : 0.2
F-measure for class 2 ; 0.2597402597402597
Mean Precision: 0.40811287477954145
Mean Recall: 0.4066666666666667
Mean F-measure: 0.34995769568824

c) Dimension = 12

[[9, 10, 13], [36, 26, 8], [5, 14, 29]]
Accuracy: 0.4266666666666667
Precision for class 0 : 0.28125
Recall for class 0 : 0.18
F-measure for class 0 ; 0.21951219512195122
Precision for class 1 : 0.37142857142857144
Recall for class 1 : 0.52
F-measure for class 1 ; 0.43333333333333335
Precision for class 2 : 0.6041666666666666
Recall for class 2 : 0.58
F-measure for class 2 ; 0.5918367346938774
Mean Precision: 0.41894841269841265
Mean Recall: 0.4266666666666666
Mean F-measure: 0.414894087716387

d) <u>Dimension</u> = 24
[[47, 44, 11], [1, 1, 0], [2, 5, 39]]
Accuracy: 0.58
Precision for class 0 : 0.46078431372549017
Recall for class 0 : 0.94
F-measure for class 0 ; 0.618421052631579
Precision for class 1 : 0.5
Recall for class 1 : 0.02
F-measure for class 1 ; 0.038461538461538464
Precision for class 2 : 0.8478260869565217
Recall for class 2 : 0.78
F-measure for class 2 ; 0.8125
Mean Precision: 0.6028701335606707
Mean Recall: 0.58
Mean F-measure: 0.48979419703103

e) <u>Dimension</u> = 28
[[40, 31, 14], [6, 14, 1], [4, 5, 35]]
Accuracy: 0.5933333333333334
Precision for class 0 : 0.47058823529411764
Recall for class 0 : 0.8
F-measure for class 0 ; 0.5925925925925927
Precision for class 1 : 0.6666666666666666
Recall for class 1 : 0.28
F-measure for class 1 ; 0.3943661971830986
Precision for class 2 : 0.7954545454545454
Recall for class 2 : 0.7
F-measure for class 2 ; 0.7446808510638298
Mean Precision: 0.6442364824717766
Mean Recall: 0.5933333333333334
Mean F-measure: 0.5772132136131

f) <u>Dimension</u> = 40
[[37, 33, 27], [5, 9, 1], [8, 8, 22]]
Accuracy: 0.4533333333333333
Precision for class 0 : 0.38144329896907214
Recall for class 0 : 0.74
F-measure for class 0 ; 0.5034013605442176
Precision for class 1 : 0.6
Recall for class 1 : 0.18
F-measure for class 1 ; 0.2769230769230769
Precision for class 2 : 0.5789473684210527
Recall for class 2 : 0.44
F-measure for class 2 ; 0.5
Mean Precision: 0.5201302224633749
Mean Recall: 0.4533333333333333
Mean F-measure: 0.42677481248909

g) Dimension = 50
[[46, 40, 50], [4, 10, 0], [0, 0, 0]]
Accuracy: 0.37333333333333335
Precision for class 0 : 0.3382352941176471
Recall for class 0 : 0.92
F-measure for class 0 ; 0.49462365591397855
Precision for class 1 : 0.7142857142857143
Recall for class 1 : 0.2
F-measure for class 1 ; 0.3125
Recall for class 2 : 0.0
F-measure for class 2 ; 0.0
Mean Precision: 0.3508403361344538
Mean Recall: 0.37333333333333335
Mean F-measure: 0.26904121863799285

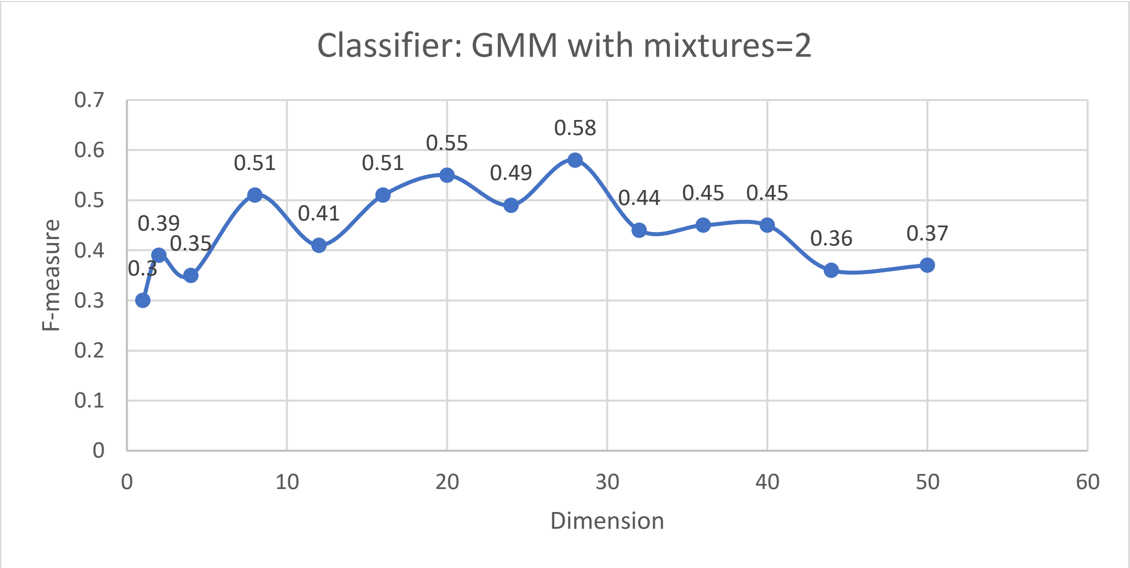

Figure: Variation of F-measure when data reduced to different dimensions using PCA

iii) <u>GMM mixtures = 4</u>

a) <u>Dimension</u> = 1

[[13, 10, 7], [28, 28, 15], [9, 12, 28]]
Accuracy: 0.46
Precision for class 0 : 0.43333333333333335
Recall for class 0 : 0.26
F-measure for class 0 ; 0.325
Precision for class 1 : 0.39436619718309857
Recall for class 1 : 0.56
F-measure for class 1 ; 0.4628099173553719
Precision for class 2 : 0.5714285714285714
Recall for class 2 : 0.56
F-measure for class 2 ; 0.5656565656565656
Mean Precision: 0.46637603398166777
Mean Recall: 0.46
Mean F-measure: 0.451155494337

b) <u>Dimension</u> = 4

[[42, 25, 13], [5, 17, 8], [3, 8, 29]]
Accuracy: 0.5866666666666667
Precision for class 0 : 0.525
Recall for class 0 : 0.84
F-measure for class 0 ; 0.6461538461538462
Precision for class 1 : 0.5666666666666667
Recall for class 1 : 0.34
F-measure for class 1 ; 0.425
Precision for class 2 : 0.725
Recall for class 2 : 0.58
F-measure for class 2 ; 0.6444444444444445
Mean Precision: 0.6055555555555556
Mean Recall: 0.5866666666666666
Mean F-measure: 0.57186609

c) <u>Dimension</u> = 12

[[40, 37, 11], [1, 1, 6], [9, 12, 33]]
Accuracy: 0.49333333333333335
Precision for class 0 : 0.45454545454545453
Recall for class 0 : 0.8
F-measure for class 0 ; 0.5797101449275363
Precision for class 1 : 0.125
Recall for class 1 : 0.02
F-measure for class 1 ; 0.03448275862068966
Precision for class 2 : 0.6111111111111112
Recall for class 2 : 0.66
F-measure for class 2 ; 0.6346153846153846
Mean Precision: 0.3968855218855219
Mean Recall: 0.49333333333333335
Mean F-measure: 0.4162694293

d) Dimension = 24
[[46, 43, 14], [2, 2, 2], [2, 5, 34]]
Accuracy: 0.5466666666666666
Precision for class 0 : 0.44660194174757284
Recall for class 0 : 0.92
F-measure for class 0 ; 0.6013071895424836
Precision for class 1 : 0.3333333333333333
Recall for class 1 : 0.04
F-measure for class 1 ; 0.07142857142857142
Precision for class 2 : 0.8292682926829268
Recall for class 2 : 0.68
F-measure for class 2 ; 0.7472527472527474
Mean Precision: 0.5364011892546109
Mean Recall: 0.5466666666666667
Mean F-measure: 0.47332950274126

e) Dimension = 28
[[45, 44, 20], [1, 1, 1], [4, 5, 29]]
Accuracy: 0.5
Precision for class 0 : 0.41284403669724773
Recall for class 0 : 0.9
F-measure for class 0 ; 0.5660377358490566
Precision for class 1 : 0.3333333333333333
Recall for class 1 : 0.02
F-measure for class 1 ; 0.03773584905660377
Precision for class 2 : 0.7631578947368421
Recall for class 2 : 0.58
F-measure for class 2 ; 0.6590909090909091
Mean Precision: 0.5031117549224744
Mean Recall: 0.5
Mean F-measure: 0.4209548

f) Dimension = 40
[[37, 29, 30], [4, 11, 0], [9, 10, 20]]
Accuracy: 0.4533333333333333
Precision for class 0 : 0.3854166666666667
Recall for class 0 : 0.74
F-measure for class 0 ; 0.5068493150684932
Precision for class 1 : 0.7333333333333333
Recall for class 1 : 0.22
F-measure for class 1 ; 0.3384615384615385
Precision for class 2 : 0.5128205128205128
Recall for class 2 : 0.4
F-measure for class 2 ; 0.449438202247191
Mean Precision: 0.5438568376068376
Mean Recall: 0.4533333333333333
Mean F-measure: 0.431583

g) Dimension = 50

[[48, 41, 50], [2, 9, 0], [0, 0, 0]]
Accuracy: 0.38
Precision for class 0 : 0.34532374100719426
Recall for class 0 : 0.96
F-measure for class 0 ; 0.5079365079365079
Precision for class 1 : 0.8181818181818182
Recall for class 1 : 0.18
F-measure for class 1 ; 0.29508196721311475
Recall for class 2 : 0.0
F-measure for class 2 ; 0.0
Mean Precision: 0.38783518639633746
Mean Recall: 0.37999999999999995
Mean F-measure: 0.267672825

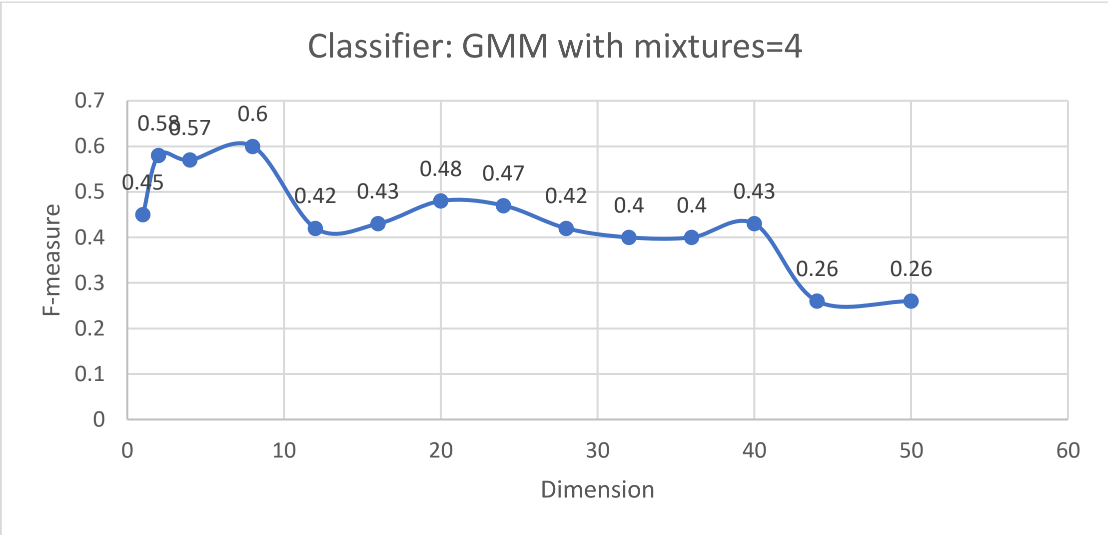

Figure: Variation of F-measure when data reduced to different dimensions using PCA

iv) GMM mixtures = 8

a) Dimension = 1

[[40, 28, 6], [6, 11, 18], [4, 11, 26]]
Accuracy: 0.5133333333333333
Precision for class 0 : 0.5405405405405406
Recall for class 0 : 0.8
F-measure for class 0 ; 0.6451612903225806
Precision for class 1 : 0.3142857142857143
Recall for class 1 : 0.22
F-measure for class 1 ; 0.25882352941176473
Precision for class 2 : 0.6341463414634146
Recall for class 2 : 0.52
F-measure for class 2 ; 0.5714285714285714
Mean Precision: 0.4963241987632232
Mean Recall: 0.5133333333333333
Mean F-measure: 0.491804463720

b) Dimension = 4

[[38, 28, 14], [11, 22, 18], [1, 0, 18]]
Accuracy: 0.52
Precision for class 0 : 0.475
Recall for class 0 : 0.76
F-measure for class 0 ; 0.5846153846153846
Precision for class 1 : 0.43137254901960786
Recall for class 1 : 0.44
F-measure for class 1 ; 0.4356435643564357
Precision for class 2 : 0.9473684210526315
Recall for class 2 : 0.36
F-measure for class 2 ; 0.5217391304347826
Mean Precision: 0.6179136566907465
Mean Recall: 0.52
Mean F-measure: 0.513

c) Dimension = 12

[[43, 40, 11], [2, 1, 3], [5, 9, 36]]
Accuracy: 0.5333333333333333
Precision for class 0 : 0.4574468085106383
Recall for class 0 : 0.86
F-measure for class 0 ; 0.5972222222222222
Precision for class 1 : 0.16666666666666666
Recall for class 1 : 0.02
F-measure for class 1 ; 0.03571428571428571
Precision for class 2 : 0.72
Recall for class 2 : 0.72
F-measure for class 2 ; 0.72
Mean Precision: 0.44803782505910167
Mean Recall: 0.5333333333333333
Mean F-measure: 0.45097883597

d) Dimension = 24
[[46, 44, 16], [2, 3, 0], [2, 3, 34]]
Accuracy: 0.5533333333333333
Precision for class 0 : 0.4339622641509434
Recall for class 0 : 0.92
F-measure for class 0 ; 0.5897435897435898
Precision for class 1 : 0.6
Recall for class 1 : 0.06
F-measure for class 1 ; 0.1090909090909091
Precision for class 2 : 0.8717948717948718
Recall for class 2 : 0.68
F-measure for class 2 ; 0.7640449438202247
Mean Precision: 0.6352523786486051
Mean Recall: 0.5533333333333333
Mean F-measure: 0.48762648088

e) Dimension = 28
[[46, 47, 20], [2, 1, 0], [2, 2, 30]]
Accuracy: 0.5133333333333333
Precision for class 0 : 0.40707964601769914
Recall for class 0 : 0.92
F-measure for class 0 ; 0.5644171779141105
Precision for class 1 : 0.3333333333333333
Recall for class 1 : 0.02
F-measure for class 1 ; 0.03773584905660377
Precision for class 2 : 0.8823529411764706
Recall for class 2 : 0.6
F-measure for class 2 ; 0.7142857142857143
Mean Precision: 0.5409219735091676
Mean Recall: 0.5133333333333333
Mean F-measure: 0.438812913752142

f) Dimension = 40
[[46, 41, 47], [3, 9, 0], [1, 0, 3]]
Accuracy: 0.38666666666666666
Precision for class 0 : 0.34328358208955223
Recall for class 0 : 0.92
F-measure for class 0 ; 0.5
Precision for class 1 : 0.75
Recall for class 1 : 0.18
F-measure for class 1 ; 0.2903225806451613
Precision for class 2 : 0.75
Recall for class 2 : 0.06
F-measure for class 2 ; 0.1111111111111111
Mean Precision: 0.6144278606965173
Mean Recall: 0.3866666666666667
Mean F-measure: 0.300477897252

g) <u>Dimension</u> = 50
[[50, 50, 50], [0, 0, 0], [0, 0, 0]]
Accuracy: 0.3333333333333333
Precision for class 0 : 0.3333333333333333
Recall for class 0 : 1.0
F-measure for class 0 ; 0.5
Recall for class 1 : 0.0
F-measure for class 1 ; 0.0
Recall for class 2 : 0.0
F-measure for class 2 ; 0.0
Mean Precision: 0.1111111111111111
Mean Recall: 0.3333333333333333
Mean F-measure: 0.1666666666666666

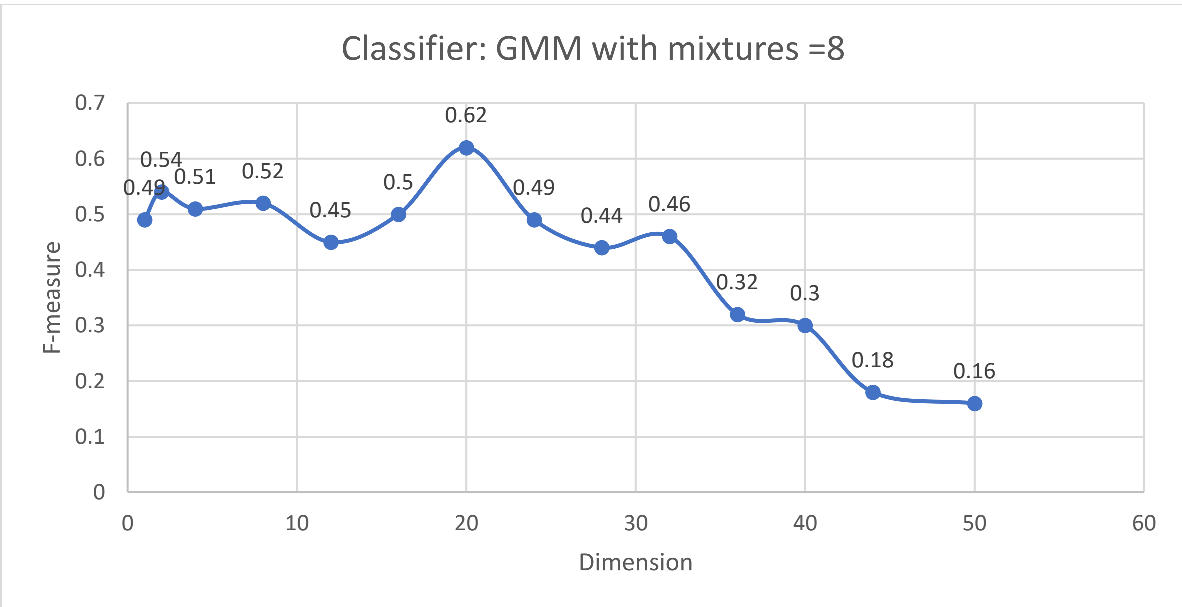

Figure: Variation of F-measure when data reduced to different dimensions using PCA

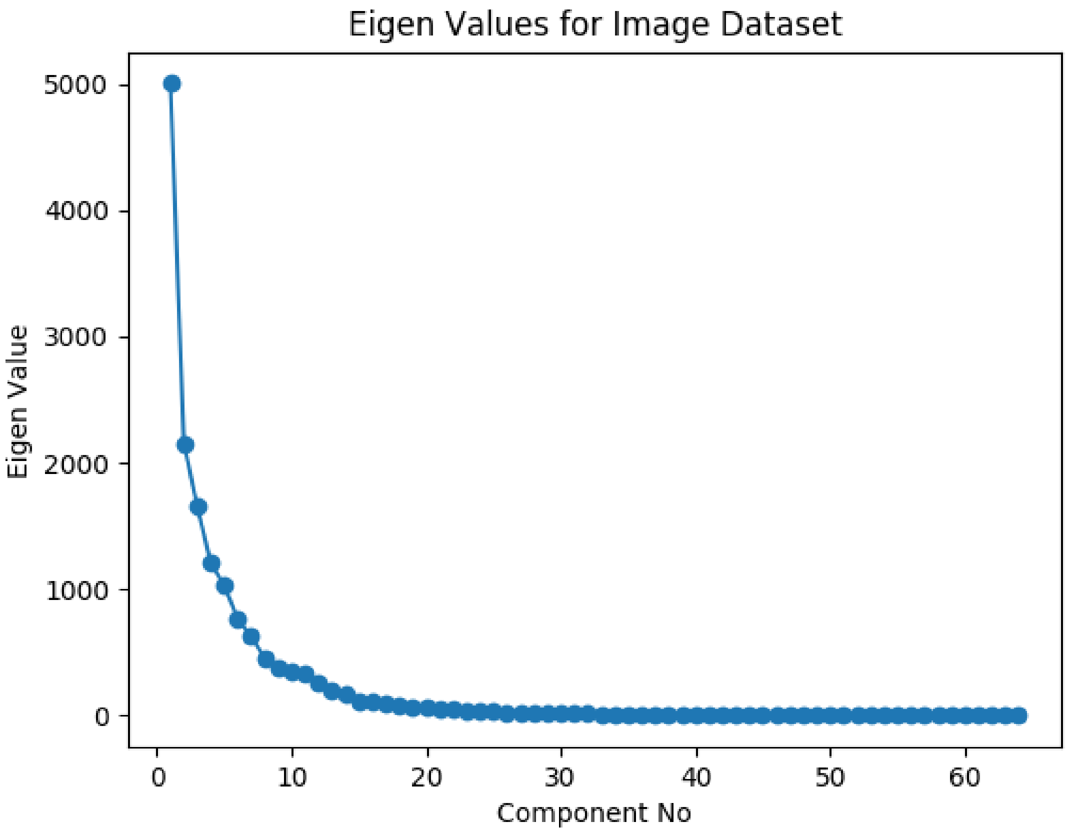

Figure: Plot showing eigen values in decreasing order

There is no particular pattern in which reducing the dimension of data affects the classification of the test examples. It is not necessary that on decreasing the dimension of the data the classification will become poorer and poorer. Instead the F-measure increases and decreases continuously.

Also we can see from the Eigen Values plot that there are only 25-30 significant Eigen Values. Other than these values all the values are very close to zero. So this shows that our data can be easily reduced to 30 dimension without any major loss in the features. Each dimension in Image Dataset represents a colour entity like sky, mountain etc. This means that we have considered more colour entities than what is actually there in the image. That's why we can reduce the error dimension of the feature vector and hence we are getting peak of F-measure below 30 dimension in all different GMM mixtures.

3 Results and Inferences for Fisher Discriminant Analysis

Dataset: Linearly Separable Dataset

a) GMM mixtures = 1,2,4,8

[[125, 0, 0], [0, 125, 0], [0, 0, 125]]

Accuracy: 1.0

Precision for class 0 : 1.0

Recall for class 0 : 1.0

F-measure for class 0 ; 1.0

Precision for class 1 : 1.0

Recall for class 1 : 1.0

F-measure for class 1 ; 1.0

Precision for class 2 : 1.0

Recall for class 2 : 1.0

F-measure for class 2 ; 1.0

Mean Precision: 1.0

Mean Recall: 1.0

Mean F-measure: 1.0

Maximum Separating Line:

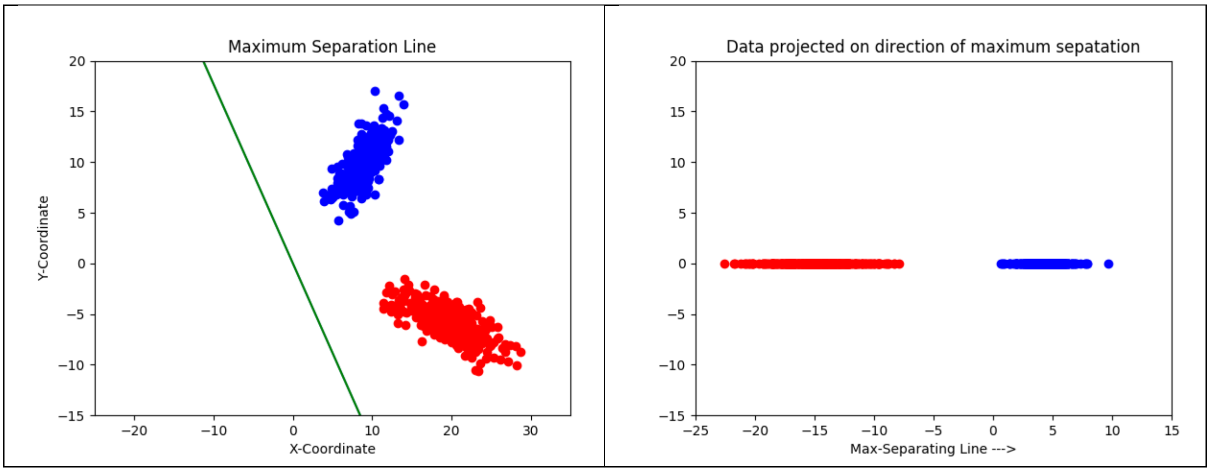

Figure: a)Maximum Separation Line for Class1 and Class2, b) Projection of data of both classes on the maximum separating line

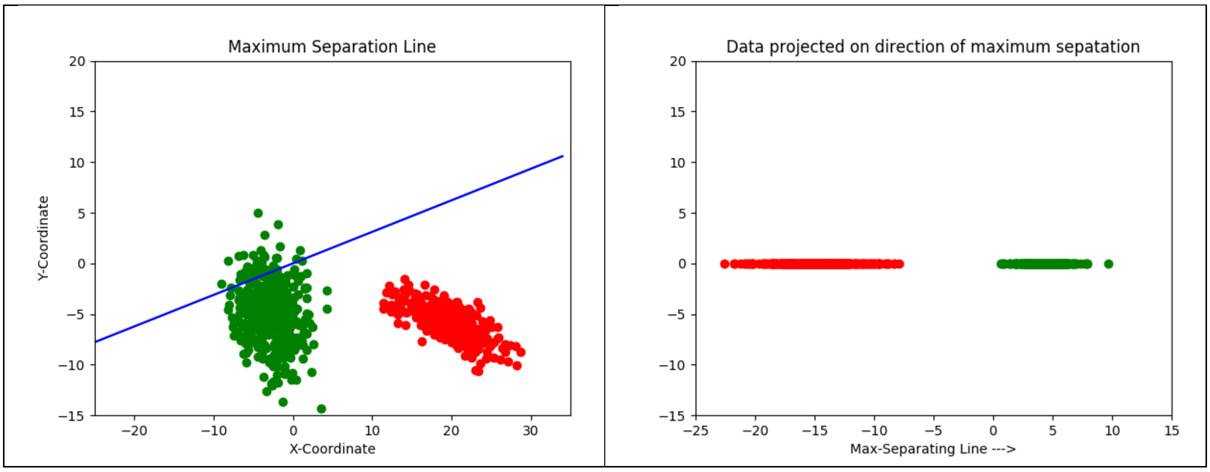

Figure: a) Maximum Separation Line for Class1 and Class3, b) Projection of data of both classes on the maximum separating line

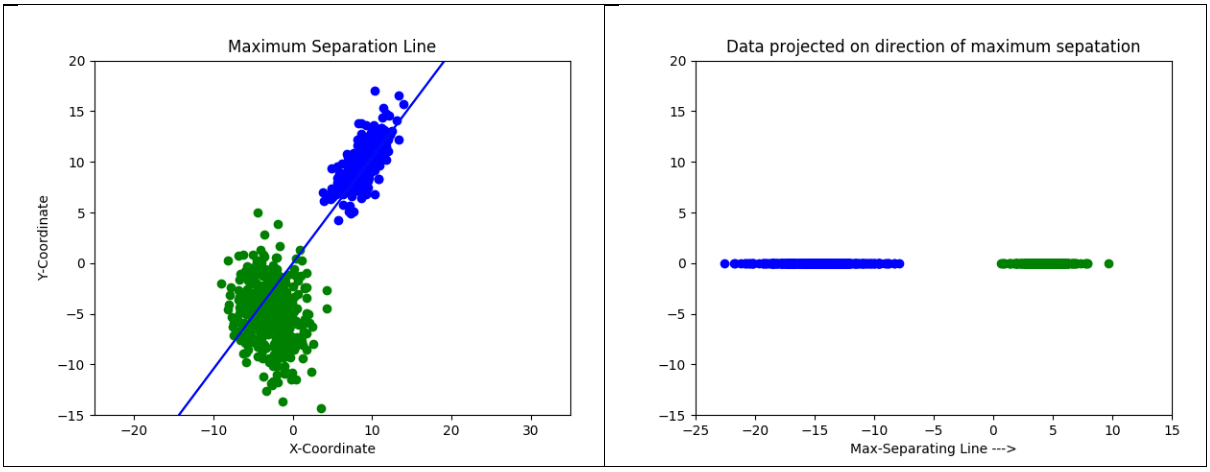

Figure: a) Maximum Separation Line for Class2 and Class3, b) Projection of data of both classes on the maximum separating line

We can clearly see from the plots that the data of all the pair of the classes is separated along the Maximum Separation Line. So when the test data will belong to Class1 then the probability of the test data in other classes will be very small and hence the test data is always classified correctly.

Dataset: Non-linearly Separable

a) GMM mixtures = 1

[[101, 18, 21], [11, 107, 0], [13, 0, 104]]
Accuracy: 0.832
Precision for class 0 : 0.7214285714285714
Recall for class 0 : 0.808
F-measure for class 0 ; 0.7622641509433963
Precision for class 1 : 0.9067796610169492
Recall for class 1 : 0.856
F-measure for class 1 ; 0.8806584362139918
Precision for class 2 : 0.8888888888888888
Recall for class 2 : 0.832
F-measure for class 2 ; 0.859504132231405
Mean Precision: 0.8390323737781364
Mean Recall: 0.832
Mean F-measure: 0.8341422397962642

b) GMM mixtures = 2

[[103, 44, 38], [13, 81, 0], [9, 0, 87]]
Accuracy: 0.7226666666666667
Precision for class 0 : 0.5567567567567567
Recall for class 0 : 0.824
F-measure for class 0 ; 0.6645161290322581
Precision for class 1 : 0.8617021276595744
Recall for class 1 : 0.648
F-measure for class 1 ; 0.7397260273972603
Precision for class 2 : 0.90625
Recall for class 2 : 0.696
F-measure for class 2 ; 0.7873303167420814
Mean Precision: 0.7749029614721104
Mean Recall: 0.7226666666666667
Mean F-measure: 0.730524157723866

c) GMM mixtures = 4

[[98, 18, 25], [6, 107, 0], [21, 0, 100]]
Accuracy: 0.8133333333333334
Precision for class 0 : 0.6950354609929078
Recall for class 0 : 0.784
F-measure for class 0 ; 0.7368421052631579
Precision for class 1 : 0.9469026548672567
Recall for class 1 : 0.856
F-measure for class 1 ; 0.8991596638655461
Precision for class 2 : 0.8264462809917356
Recall for class 2 : 0.8
F-measure for class 2 ; 0.8130081300813008
Mean Precision: 0.8227947989506333
Mean Recall: 0.8133333333333335
Mean F-measure: 0.8163366330700016

d) GMM mixtures = 8
[[111, 9, 22], [7, 116, 1], [7, 0, 102]]
Accuracy: 0.8773333333333333
Precision for class 0 : 0.7816901408450704
Recall for class 0 : 0.888
F-measure for class 0 ; 0.8314606741573034
Precision for class 1 : 0.9354838709677419
Recall for class 1 : 0.928
F-measure for class 1 ; 0.931726907630522
Precision for class 2 : 0.9357798165137615
Recall for class 2 : 0.816
F-measure for class 2 ; 0.8717948717948718
Mean Precision: 0.8843179427755246
Mean Recall: 0.8773333333333334
Mean F-measure: 0.8783274845275657

Maximum Separating Line:

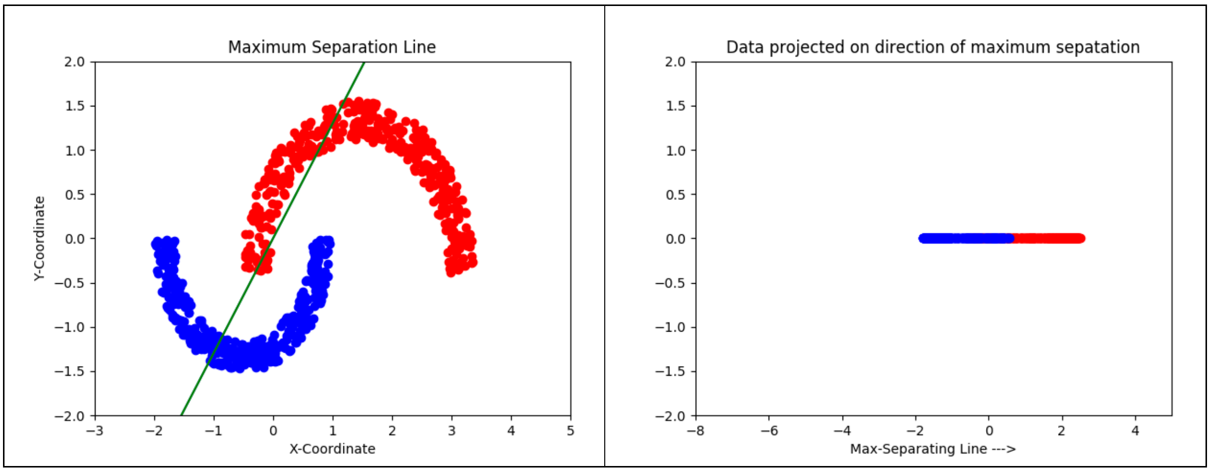

Figure: a)Maximum Separation Line for Class1 and Class2, b) Projection of data of both classes on the maximum separating line

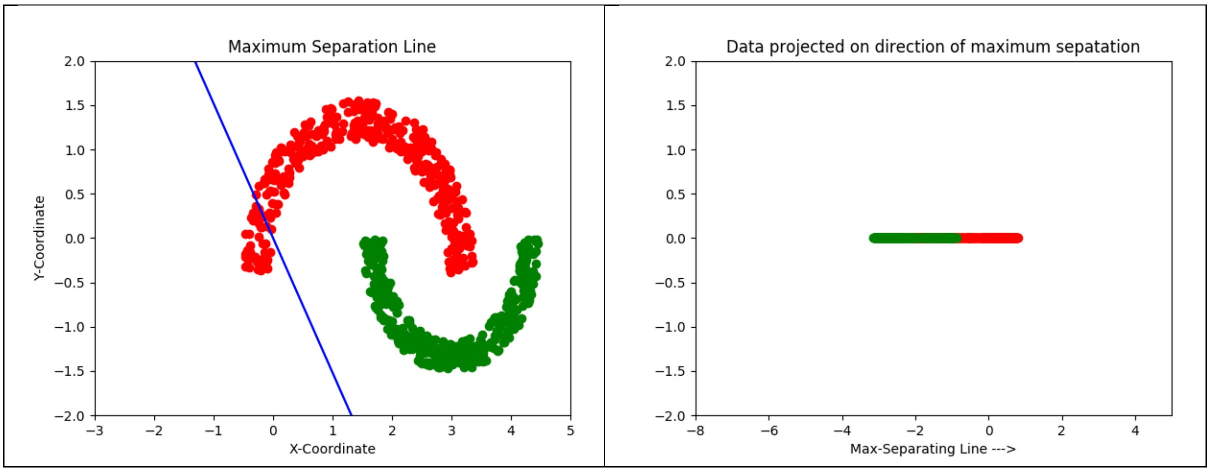

Figure: a)Maximum Separation Line for Class1 and Class2, b) Projection of data of both classes on the maximum separating line

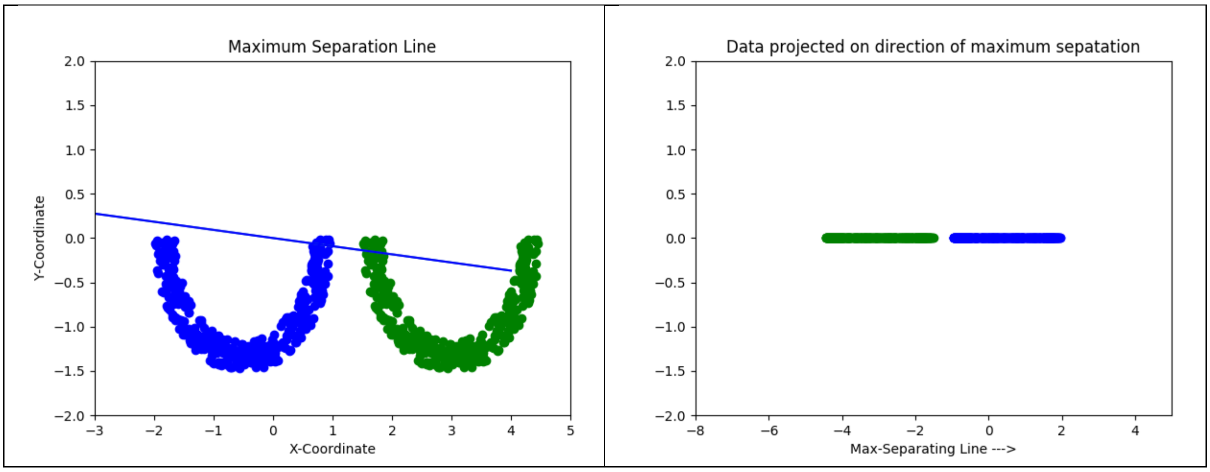

Figure: a)Maximum Separation Line for Class1 and Class2, b) Projection of data of both classes on the maximum separating line

As the data can't be linearly separated, therefore FDA gives the direction of maximum separation of the data. And since the data is overlapping after projection on the maximum separating line, therefore the classification is not accurate(even with higher number of Gaussian mixtures). Data was correctly classified using multi-model Gaussian distribution as we have seen in Assignment-2 but due to loss of information after FDA the data is no more classified 100% though accuracy is still around 85-90%.

Dataset: Image(coast, Kennel, Volleyball)

a) GMM mixtures = 1

[[49, 45, 43], [0, 0, 0], [1, 5, 7]]
Accuracy: 0.37333333333333335
Precision for class 0 : 0.35766423357664234
Recall for class 0 : 0.98
F-measure for class 0 ; 0.5240641711229946
Recall for class 1 : 0.0
F-measure for class 1 ; 0.0
Precision for class 2 : 0.5384615384615384
Recall for class 2 : 0.14
F-measure for class 2 ; 0.2222222222222222
Mean Precision: 0.2987085906793936
Mean Recall: 0.37333333333333335
Mean F-measure: 0.24876213111507228

b) GMM mixtures = 2

[[37, 21, 13], [7, 7, 8], [6, 22, 29]]
Accuracy: 0.4866666666666667
Precision for class 0 : 0.5211267605633803
Recall for class 0 : 0.74
F-measure for class 0 ; 0.6115702479338843
Precision for class 1 : 0.3181818181818182
Recall for class 1 : 0.14
F-measure for class 1 ; 0.19444444444444445
Precision for class 2 : 0.5087719298245614
Recall for class 2 : 0.58
F-measure for class 2 ; 0.5420560747663552
Mean Precision: 0.44936016952325325
Mean Recall: 0.48666666666666664
Mean F-measure: 0.4493569223815613

c) GMM mixtures = 4

[[26, 15, 7], [16, 24, 9], [8, 11, 34]]
Accuracy: 0.56
Precision for class 0 : 0.5416666666666666
Recall for class 0 : 0.52
F-measure for class 0 ; 0.5306122448979592
Precision for class 1 : 0.4897959183673469
Recall for class 1 : 0.48
F-measure for class 1 ; 0.48484848484848486
Precision for class 2 : 0.6415094339622641
Recall for class 2 : 0.68
F-measure for class 2 ; 0.6601941747572816
Mean Precision: 0.5576573396654259
Mean Recall: 0.56
Mean F-measure: 0.55855163483457

d) GMM mixtures = 8
[[18, 13, 7], [30, 34, 6], [2, 3, 37]]
Accuracy: 0.5933333333333334
Precision for class 0 : 0.47368421052631576
Recall for class 0 : 0.36
F-measure for class 0 ; 0.40909090909090906
Precision for class 1 : 0.4857142857142857
Recall for class 1 : 0.68
F-measure for class 1 ; 0.5666666666666667
Precision for class 2 : 0.8809523809523809
Recall for class 2 : 0.74
F-measure for class 2 ; 0.8043478260869565
Mean Precision: 0.6134502923976608
Mean Recall: 0.5933333333333334
Mean F-measure: 0.593368467281510

Comparing PCA and FDA:

	Taking only λ_{max} in PCA	FDA
GMM mix = 1	0.31	0.24
GMM mix = 2	0.29	0.44
GMM mix = 4	0.45	0.55
GMM mix = 8	0.49	0.59

Table: Table showing values of F-measure for case when we reduce the feature vector to Linear Space using only one component in PCA and using FDA

We can see from the above table that the direction of maximum variance doesn't guarantee that the data contains maximum information in that direction and hence it should be used for classification. As we can clearly see that the direction FDA gives contains much more information for discriminating the data and hence obtained better F-measure for FDA.

4 Results and Inference Perceptron Learning

Initial Values of parameters $\overline{w}$ = [1, … ,1] (d-dimension vector), w_o=-1

Dataset: Linearly Separable

[[125, 1, 0], [0, 124, 0], [0, 0, 125]]
Accuracy: 0.9973333333333333
Precision for class 0 : 0.9920634920634921
Recall for class 0 : 1.0
F-measure for class 0 ; 0.9960159362549801
Precision for class 1 : 1.0
Recall for class 1 : 0.992
F-measure for class 1 ; 0.9959839357429718
Precision for class 2 : 1.0
Recall for class 2 : 1.0
F-measure for class 2 ; 1.0
Mean Precision: 0.9973544973544973
Mean Recall: 0.9973333333333333
Mean F-measure: 0.997333290665984

Inference:

Since the dataset is linearly separable therefore the perceptron finds the separating hyperplane and converges.

One point in the second class got misclassified because the Perceptron gives the separating hyperplane in correspondence with train data. It might be that a test point comes and even which lies on the other side of the hyperplane but the data is still linearly separable. So if we will again find the separating hyperplane then it will give 100% accuracy after considering that test point.

5 Results and Inferences for SVM classifier

<u>Dataset: Linearly Separable</u>

a) <u>Kernel:</u> Linear, Polynomial

[[125, 0, 0], [0, 125, 0], [0, 0, 125]]
Accuracy: 1.0
Precision for class 0 : 1.0
Recall for class 0 : 1.0
F-measure for class 0 ; 1.0
Precision for class 1 : 1.0
Recall for class 1 : 1.0
F-measure for class 1 ; 1.0
Precision for class 2 : 1.0
Recall for class 2 : 1.0
F-measure for class 2 ; 1.0
Mean Precision: 1.0
Mean Recall: 1.0
Mean F-measure: 1.0

b) <u>Kernel:</u> Gaussian

[[124, 0, 0], [0, 124, 0], [1, 1, 125]]
Accuracy: 0.9946666666666667
Precision for class 0 : 1.0
Recall for class 0 : 0.992
F-measure for class 0 ; 0.9959839357429718
Precision for class 1 : 1.0
Recall for class 1 : 0.992
F-measure for class 1 ; 0.9959839357429718
Precision for class 2 : 0.984251968503937
Recall for class 2 : 1.0
F-measure for class 2 ; 0.9920634920634921
Mean Precision: 0.994750656167979
Mean Recall: 0.9946666666666667
Mean F-measure: 0.9946771211831452

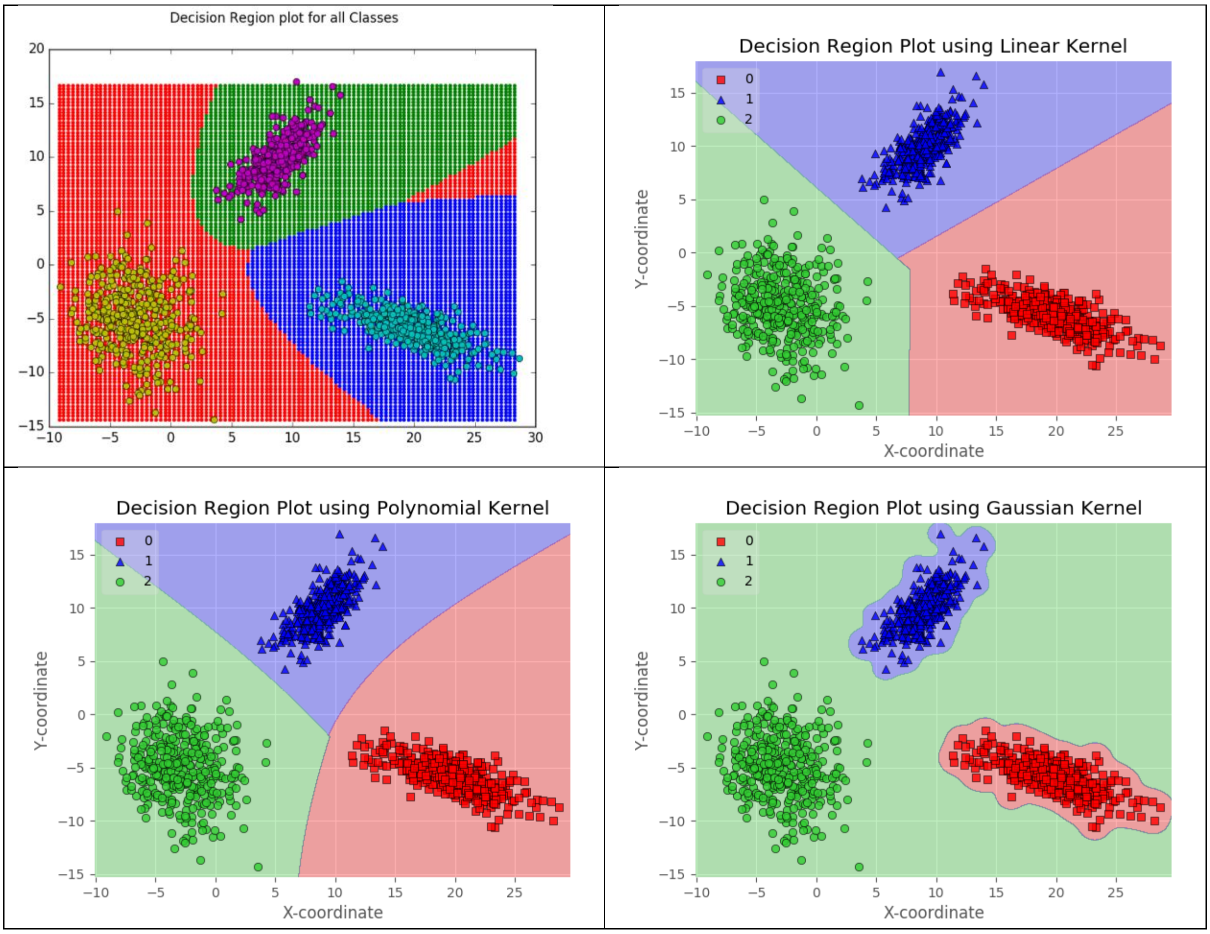

Figure: Decision Region plot for a) Bayes Classifier b) Linear Kernel SVM c) Polynomial Kernel SVM d) Gaussian Kernel SVM on Linearly Separable Dataset

Linearly Separable data is classified with 100% accuracy.

The decision boundary in the Bayes classifier resembles the distribution of the data i.e., in Bayes Classifier we assume the distribution of data and then according to the probability of data in that distribution we classify the data. Because of this the decision boundary clearly shows the distribution of data and how it is oriented. But in SVM we don't need to assume any underlying distribution of the data. We discriminate the data of the two classes and obtain the boundary and hence boundary shows no sign of how data of particular class is oriented.

Dataset: Non-linearly Separable

a) Kernel: Linear

[[110, 3, 11], [4, 122, 0], [11, 0, 114]]
Accuracy: 0.9226666666666666
Precision for class 0 : 0.8870967741935484
Recall for class 0 : 0.88
F-measure for class 0 ; 0.8835341365461847
Precision for class 1 : 0.9682539682539683
Recall for class 1 : 0.976
F-measure for class 1 ; 0.9721115537848605
Precision for class 2 : 0.912
Recall for class 2 : 0.912
F-measure for class 2 ; 0.912
Mean Precision: 0.9224502474825056
Mean Recall: 0.9226666666666666
Mean F-measure: 0.92254856344368

b) Kernel: Polynomial

[[115, 12, 12], [0, 113, 3], [10, 0, 110]]
Accuracy: 0.9013333333333333
Precision for class 0 : 0.8273381294964028
Recall for class 0 : 0.92
F-measure for class 0 ; 0.8712121212121212
Precision for class 1 : 0.9741379310344828
Recall for class 1 : 0.904
F-measure for class 1 ; 0.9377593360995852
Precision for class 2 : 0.9166666666666666
Recall for class 2 : 0.88
F-measure for class 2 ; 0.8979591836734694
Mean Precision: 0.9060475757325174
Mean Recall: 0.9013333333333334
Mean F-measure: 0.90231021366172

c) Kernel: Gaussian

[[115, 12, 12], [0, 113, 3], [10, 0, 110]]
Accuracy: 0.9013333333333333
Precision for class 0 : 0.8273381294964028
Recall for class 0 : 0.92
F-measure for class 0 ; 0.8712121212121212
Precision for class 1 : 0.9741379310344828
Recall for class 1 : 0.904
F-measure for class 1 ; 0.9377593360995852
Precision for class 2 : 0.9166666666666666
Recall for class 2 : 0.88
F-measure for class 2 ; 0.8979591836734694
Mean Precision: 0.9060475757325174
Mean Recall: 0.9013333333333334
Mean F-measure: 0.90231021366172

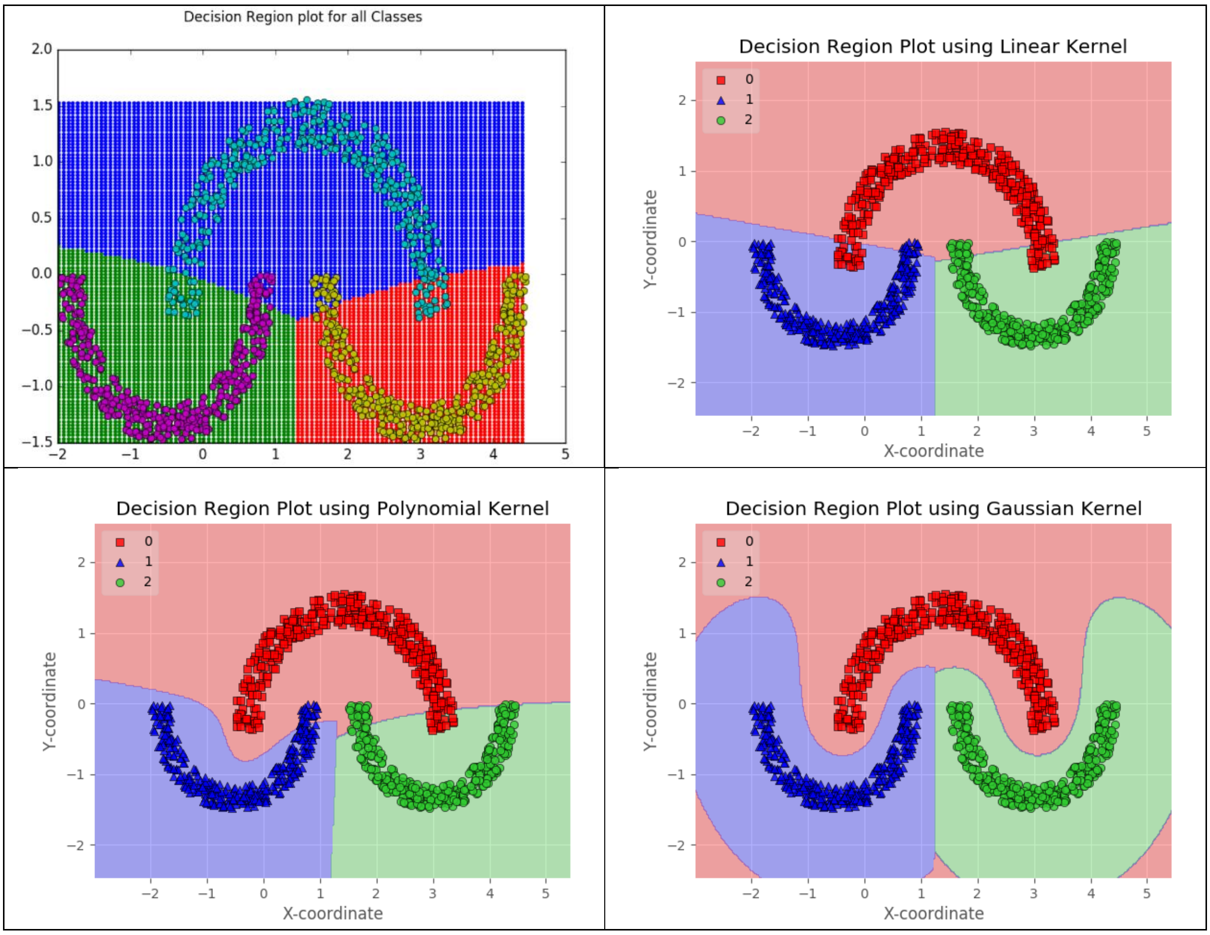

Figure: Decision Region plot for a) Bayes Classifier b) Linear Kernel SVM c) Polynomial Kernel SVM d) Gaussian Kernel SVM on Non-linearly Separable Dataset

We can see that the Bayes classifier has made similar boundary for lower class(U shaped) data because they both have same covariance matrix and they are symmetric hence they have a linear boundary between them. But SVM polynomial Kernel don't make same boundary as it only constructs a maximum margin hyperplane to separate the data.

Dataset: Image (coast, Kennel, volleyball)

a) Kernel: Linear

[[31, 15, 5], [16, 30, 8], [3, 5, 37]]
Accuracy: 0.6533333333333333
Precision for class 0 : 0.6078431372549019
Recall for class 0 : 0.62
F-measure for class 0 ; 0.6138613861386139
Precision for class 1 : 0.5555555555555556
Recall for class 1 : 0.6
F-measure for class 1 ; 0.576923076923077
Precision for class 2 : 0.8222222222222222
Recall for class 2 : 0.74
F-measure for class 2 ; 0.7789473684210526
Mean Precision: 0.6618736383442265
Mean Recall: 0.6533333333333333
Mean F-measure: 0.6565772771609145

b) Kernel: Polynomial

[[43, 23, 6], [2, 13, 2], [5, 14, 42]]
Accuracy: 0.6533333333333333
Precision for class 0 : 0.5972222222222222
Recall for class 0 : 0.86
F-measure for class 0 ; 0.7049180327868853
Precision for class 1 : 0.7647058823529411
Recall for class 1 : 0.26
F-measure for class 1 ; 0.3880597014925373
Precision for class 2 : 0.6885245901639344
Recall for class 2 : 0.84
F-measure for class 2 ; 0.7567567567567568
Mean Precision: 0.6834842315796993
Mean Recall: 0.6533333333333333
Mean F-measure: 0.6165781636787264

c) Kernel: Gaussian

[[0, 1, 0], [2, 2, 0], [48, 47, 50]]
Accuracy: 0.3466666666666667
Precision for class 0 : 0.0
Recall for class 0 : 0.0
Precision for class 1 : 0.5
Recall for class 1 : 0.04
F-measure for class 1 ; 0.07407407407407407
Precision for class 2 : 0.3448275862068966
Recall for class 2 : 1.0
F-measure for class 2 ; 0.5128205128205129
Mean Precision: 0.28160919540229884
Mean Recall: 0.3466666666666667
Mean F-measure: 0.195631528964862

6 Comparison with all classifiers for each dataset

1) Linearly Separable Dataset

S. No.	Classifier	Accuracy
1.	Bayes Classifier	100
2.	Bayes classifier using GMM	100
3.	Bayes classifier(after FDA)	100
4.	Perceptron based Classifier	100
5.	Support Vector Machines	100

2) Non-linearly Separable Dataset

S. No.	Classifier	Accuracy
1.	Bayes Classifier ($\Sigma = \sigma$**2 I)	86
2.	Bayes Classifier (Full cov. matrix same for all classes)	94
3.	Bayes Classifier (diagonal Covariance Matrix)	93
4.	Bayes Classifier (Full covariance matrix diff for all classes)	93
5.	GMM (after FDA)	87
6.	Support Vector Machine	92

3) Real World Dataset

S. No.	Classifier	Accuracy
1.	Bayes Classifier ($\Sigma = \sigma$**2 I)	98
2.	Bayes Classifier (Full cov. matrix same for all classes)	98
3.	Bayes Classifier (diagonal Covariance Matrix)	98
4.	Bayes Classifier (Full covariance matrix diff for all classes)	98

4) Image Dataset

S. No.	Classifier	Accuracy
1.	K- Nearest Neighbour	41
2.	Ergodic Hidden Markov Model	43
3.	GMM (after PCA)	62
4.	GMM (after FDA)	59
5.	Support Vector Machine	65

5) Speech Dataset

S. No.	Classifier	Accuracy
1.	K-Nearest Neighbour	60
2.	Non-Ergodic Hidden Markov Model	43

*****THE END*****