Regression and Classification

Programming Assignment 2 Report

CS5691 - Pattern Recognition and Machine Learning

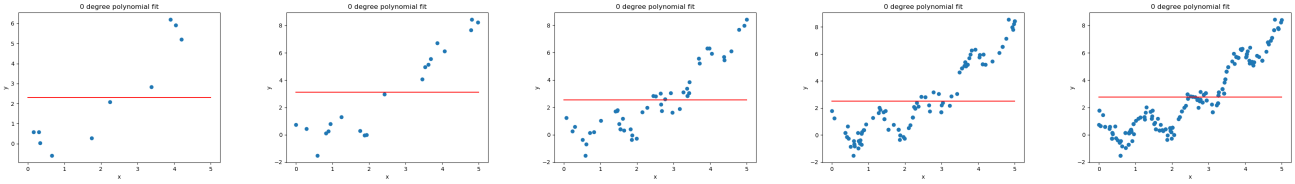
Team 29 - Abhigyan Chattopadhyay (EE19B146) & Nihal John George (EE19B131)

March 3, 2022

1 Part A – Regression

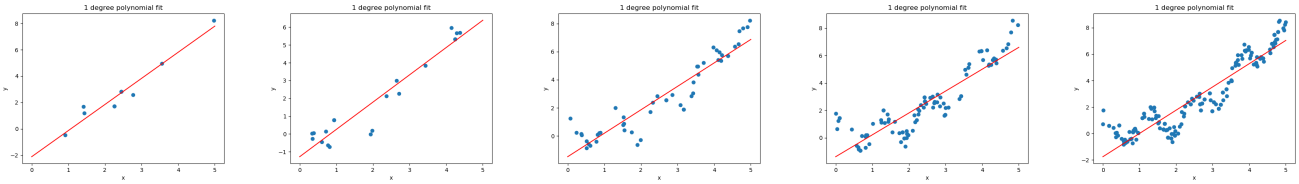
1.1 1D Dataset Experiments

First, the dataset size was varied against the order of the polynomial, resulting in the following plots:



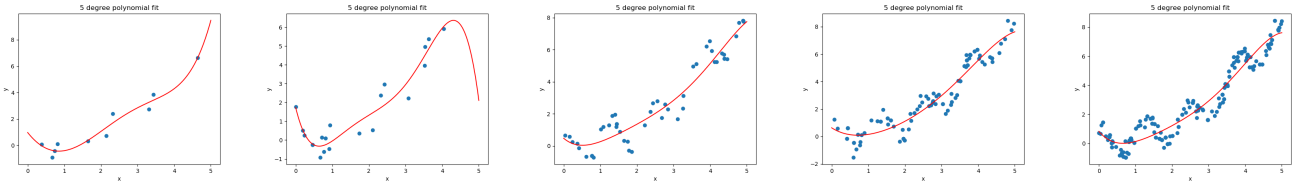
(a) 10 data points (b) 20 data points (c) 50 data points (d) 100 data points (e) 200 data points

Figure 1: Order 0



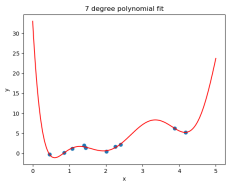
(a) 10 data points (b) 20 data points (c) 50 data points (d) 100 data points (e) 200 data points

Figure 2: Order 1

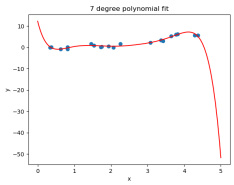


(a) 10 data points (b) 20 data points (c) 50 data points (d) 100 data points (e) 200 data points

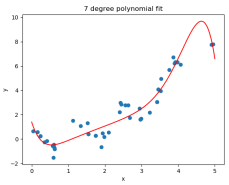
Figure 3: Order 5



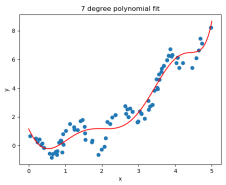
(a) 10 data points



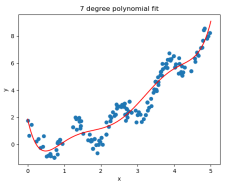
(b) 20 data points



(c) 50 data points

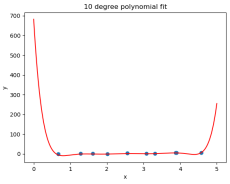


(d) 100 data points

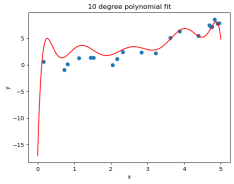


(e) 200 data points

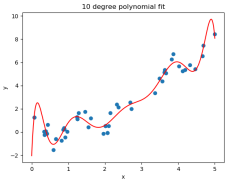
Figure 4: Order 7



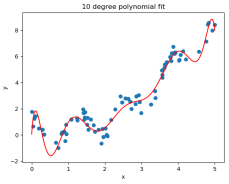
(a) 10 data points



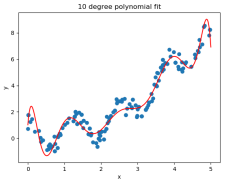
(b) 20 data points



(c) 50 data points



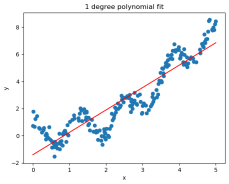
(d) 100 data points



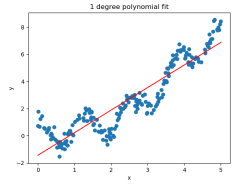
(e) 200 data points

Figure 5: Order 10

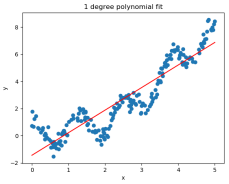
Next, we vary the Regularization Parameter against the Polynomial Order:



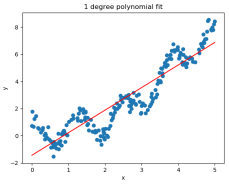
(a) $\lambda = 1$



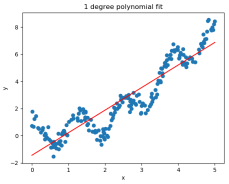
(b) $\lambda = 0.01$



(c) $\lambda = 0.0001$

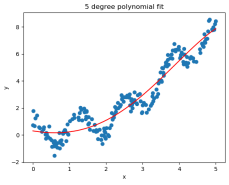


(d) $\lambda = 0.000001$

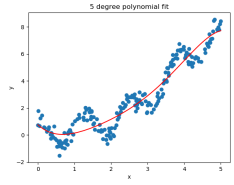


(e) $\lambda = 0.000000001$

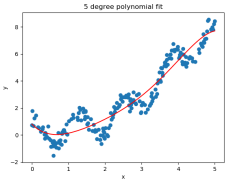
Figure 6: Order 1



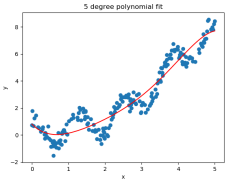
(a) $\lambda = 1$



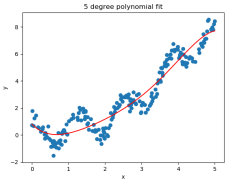
(b) $\lambda = 0.01$



(c) $\lambda = 0.0001$

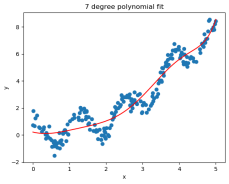


(d) $\lambda = 0.000001$

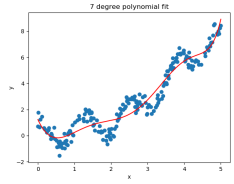


(e) $\lambda = 0.00000001$

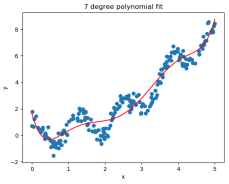
Figure 7: Order 5



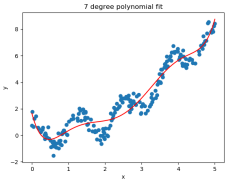
(a) $\lambda = 1$



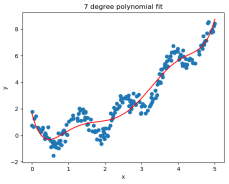
(b) $\lambda = 0.01$



(c) $\lambda = 0.0001$

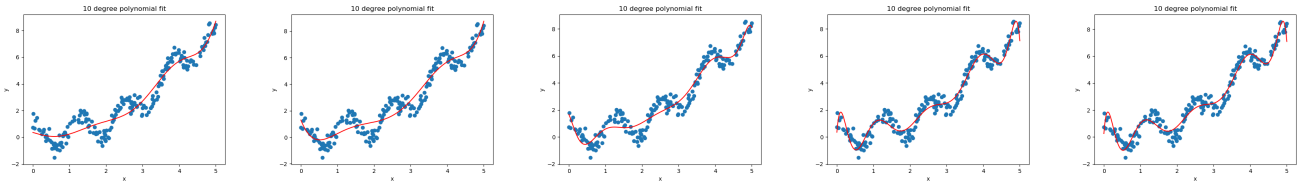


(d) $\lambda = 0.000001$



(e) $\lambda = 0.000000001$

Figure 8: Order 7



(a) $\lambda = 1$ (b) $\lambda = 0.01$ (c) $\lambda = 0.0001$ (d) $\lambda = 0.000001$ (e) $\lambda = 0.000000001$

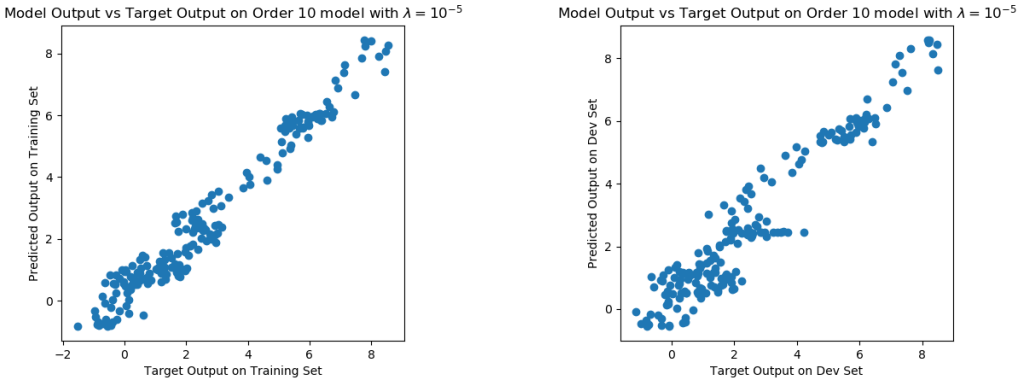
Figure 9: Order 10

We weren't able to go over a maximum polynomial order of 10 because of the large condition numbers in the matrices, which would result in large inverse calculations that would result in inaccurate graphs.

Read more here: Condition Number Post on StackExchange

1.1.1 Best Performing Model

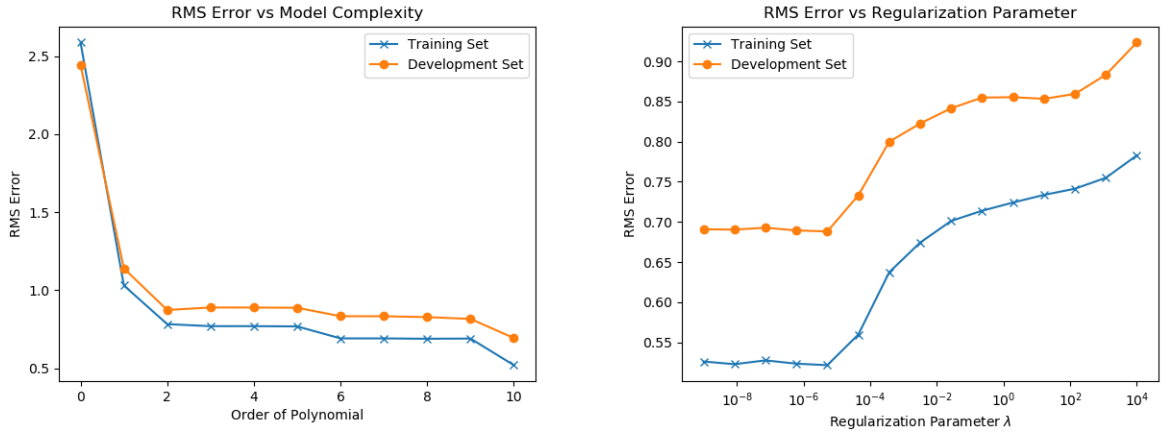
The lowest error was found to be on the Model of Order 10 for regression parameter around 10^{-5} (regularization parameter was chosen from observation, polynomial order from error graph shown afterwards)



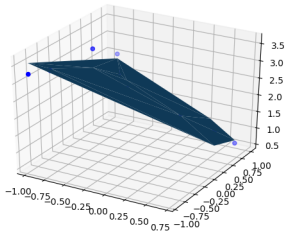
(a) Training Set (b) Development Set

Figure 10: Best Performing Model's Scatter Plot

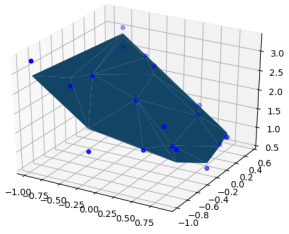
1.1.2 RMS Error vs Model Complexity



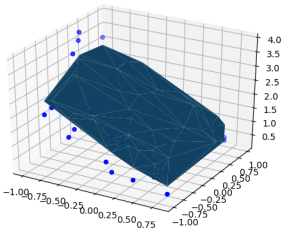
1.2 2D Dataset Experiments



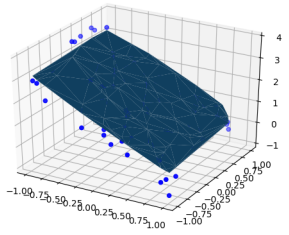
(a) 10 data points



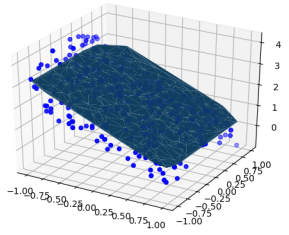
(b) 20 data points



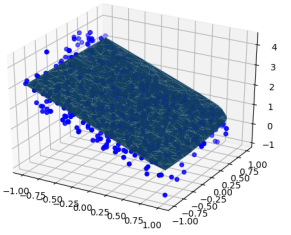
(c) 50 data points



(d) 100 data points

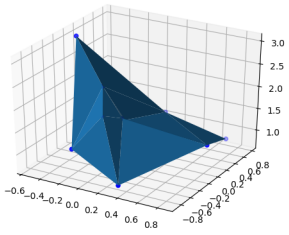


(e) 500 data points

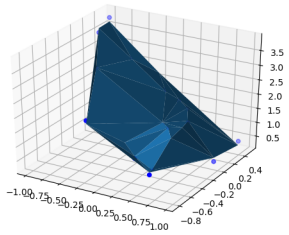


(f) 1000 data points

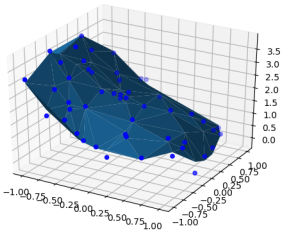
Figure 11: Order 1



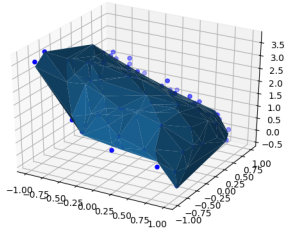
(a) 10 data points



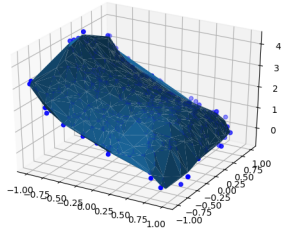
(b) 20 data points



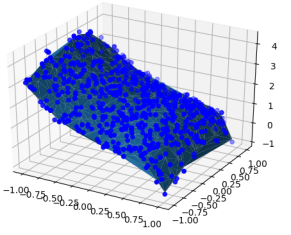
(c) 50 data points



(d) 100 data points

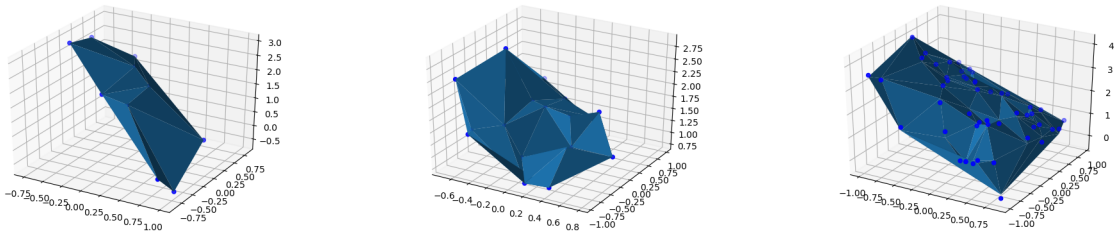


(e) 500 data points

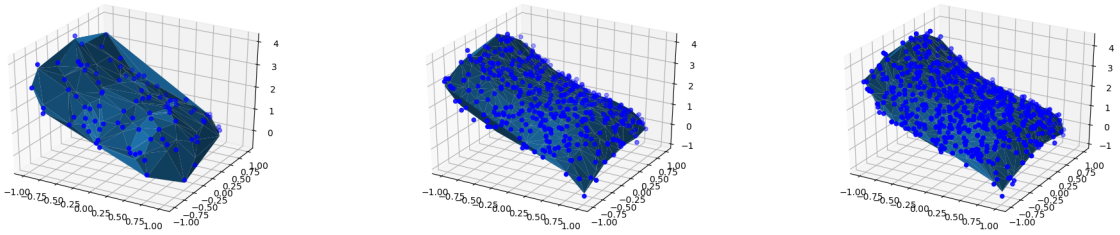


(f) 1000 data points

Figure 12: Order 5



(a) 10 data points (b) 20 data points (c) 50 data points

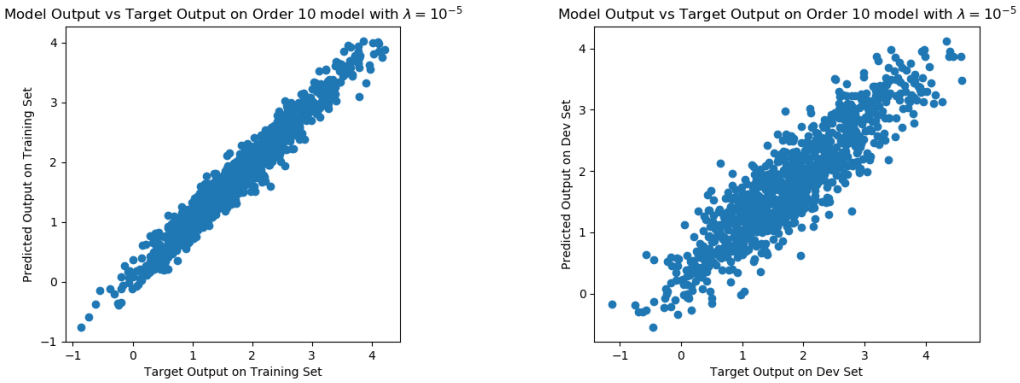


(d) 100 data points (e) 500 data points (f) 1000 data points

Figure 13: Order 10

1.2.1 Best Performing Model

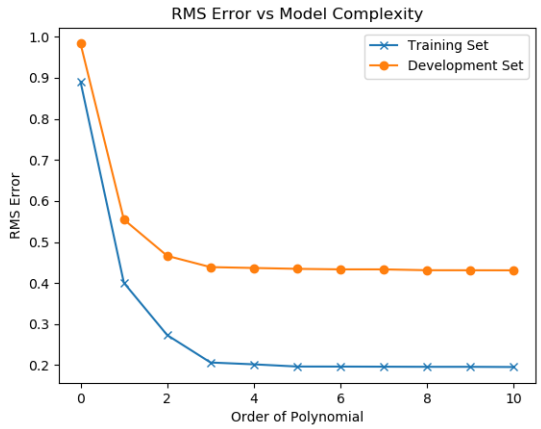
The lowest error was found to be on the Model of Order 10 for regression parameter around 10^{-5} (regularization parameter was chosen from observation, polynomial order from error graph shown afterwards)

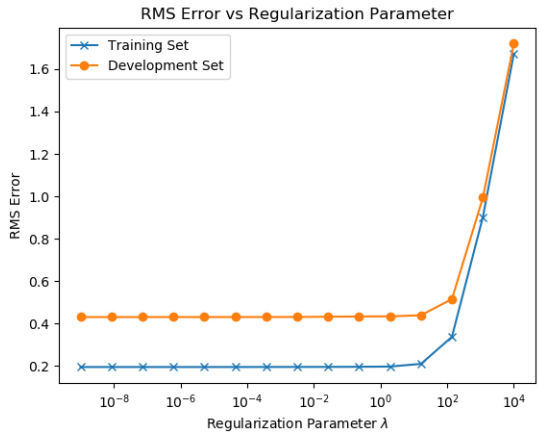


(a) Training Set (b) Development Set

Figure 14: Best Performing Model's Scatter Plot

1.2.2 RMS Error





1.3 Observations

- As regularization parameter λ is increased, the fit initially improves until around 10^{-5} and then the model begins to underfit the data
- As model complexity increases, the fit improves, but begins to react too aggressively to sudden changes in the data values
- With polynomial orders larger than 10, due to large condition numbers, small errors in inverse calculations result in large variations in the model and hence a poor fit is obtained.
- Similar results are seen with 2D as well as 1D regression, although 2D data is possible to be represented by lesser complex models with higher accuracy.
- As the condition number problem did not allow us to increase the model order above 10, we were unable to show an overfit, but as our 1D data plot had 7 turning points, it should be possible to represent with an order 8 polynomial without any overfit, and with larger than 8 it would overfit and would need to be rectified by regularization, which is why we are able to see a visible difference in the plots for order 10 with different values of λ.
- We also notice that the coefficients of the equations reduce in magnitude when we increase the regularization coefficient

2 Part B – Bayesian Classification

Aim: To perform 3-class classification by modeling data as coming from unimodal Gaussian distributions. Evaluate the performance of the model using confusion matrices, ROC and DET plots, and a plot of data with the decision boundaries.

2.1 Experiments

Five experiments were conducted. In each, a particular assumption was made about the underlying distribution of data. This had an effect on the developed models and their performance.

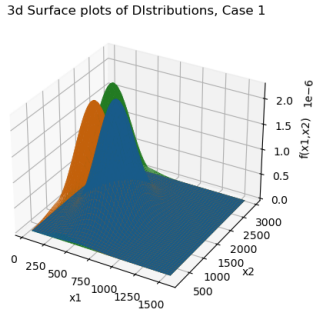
After loading the data, the mean feature vector for each class in all cases was computed (Take mean of values in particular class along each feature column). The following procedures were applied for calculating covariance in each case (we use notation B = Bayes, NB = Naive Bayes, C = covariance matrix, x_1, x_2 = feature columns) –

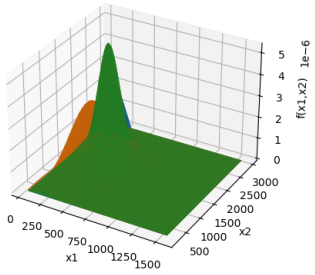
- Case 1 (B, same C) – Covariance using x_1^2, x_2^2, x_1x_2 and squares of means, all classes together. No Bessel correction (we divided by N instead of N-1, since the former gives the MLE of covariance)
- Case 2 (B, diff C) – Same as above, but for each class separately, giving 3 different matrices.
- Case 3 (NB, $C = \sigma^2 I$) – Naive assumes each feature column can be modelled as a 1D-distribution independent of other columns. Since this case assumes same variance for both columns, we can find the variance of all values x_1 and x_2, instead of doing x_1 separately and x_2 separately. This variance gives σ^2
- Case 4 (NB, same diag C) – Same as above, but find variance of each feature column separately.
- Case 5 (NB, diff diag C) – Same as above, but done for each class separately.

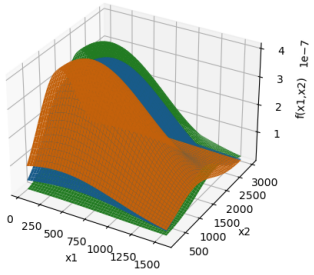
Then, a Bayes classifer was made, and points from the Dev Set were classified, and results plotted.

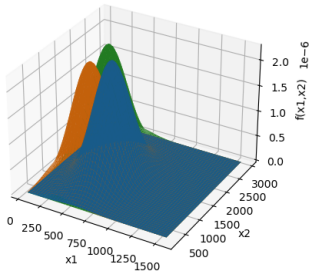
2.2 Observations

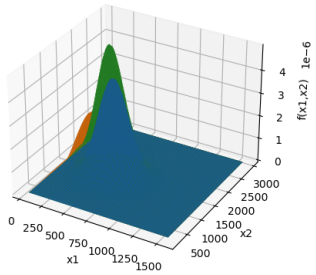
Multiple Plots Below. All are from the 'real' dataset, except the ROC and DET, which are for all 3 datasets. Plots for linsep and nonlinsep can be generated by running the code

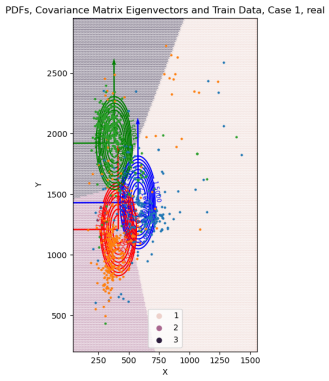


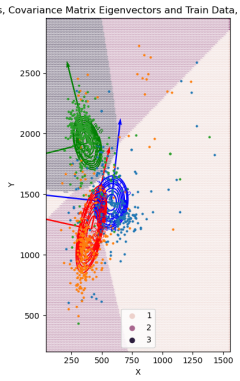


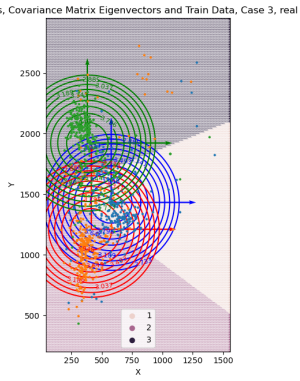


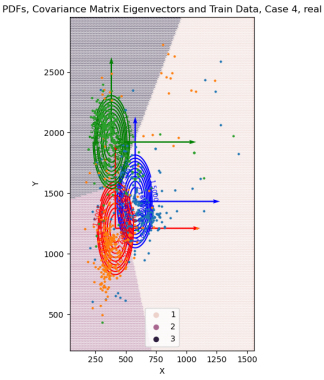


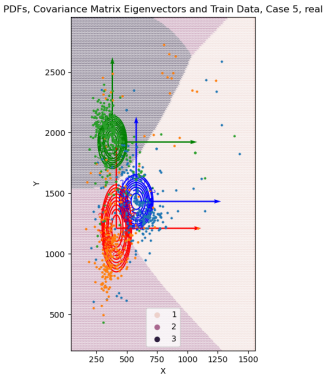


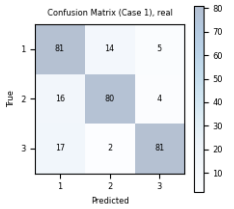

PDFs, Covariance Matrix Eigenvectors and Train Data, Case 1, real
X
Y
1
2
3


PDFs, Covariance Matrix Eigenvectors and Train Data, Case 2, real
X
Y
1
2
3

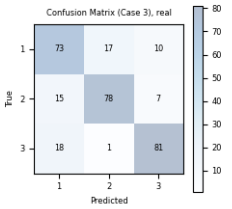

PDFs, Covariance Matrix Eigenvectors and Train Data, Case 3, real
X
Y
1
2
3

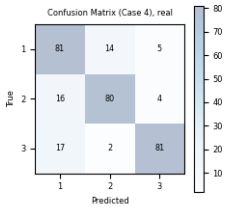

PDFs, Covariance Matrix Eigenvectors and Train Data, Case 4, real
X
Y
1
2
3

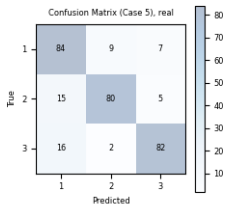

PDFs, Covariance Matrix Eigenvectors and Train Data, Case 5, real
X
Y
1
2
3

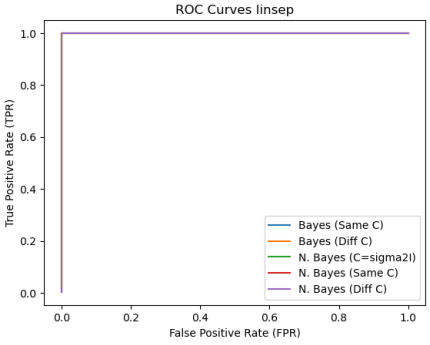

Confusion Matrix (Case 1), real
True
Predicted
81 14 5
16 80 4
17 2 81

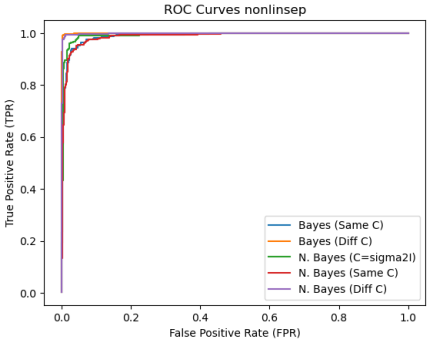

Confusion Matrix (Case 2), real
True
Predicted
82 10 8
14 82 4
16 2 82

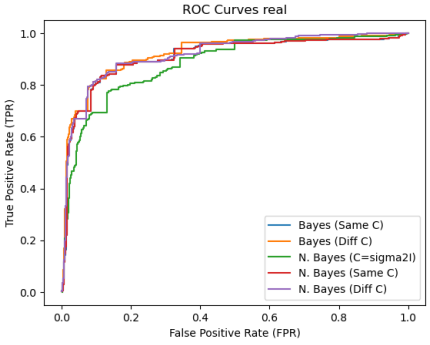

Confusion Matrix (Case 3), real
True
Predicted
73 17 10
15 78 7
18 1 81

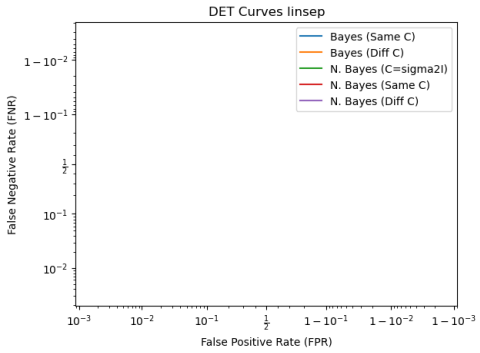

Confusion Matrix (Case 4), real
True
Predicted
81 14 5
16 80 4
17 2 81

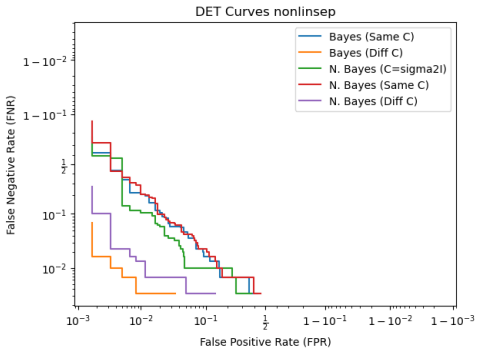

Confusion Matrix (Case 5), real
True
Predicted
84 9 7
15 80 5
16 2 82

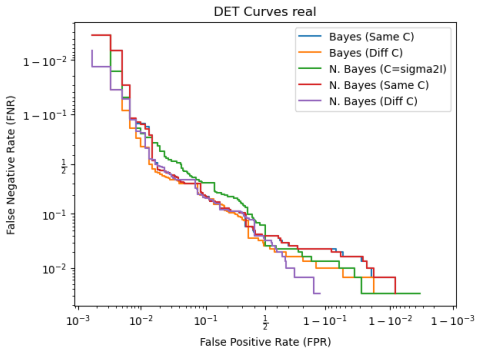

ROC Curves linsep
True Positive Rate (TPR)
False Positive Rate (FPR)
Bayes (Same C)
Bayes (Diff C)
N. Bayes (C=sigma2I)
N. Bayes (Same C)
N. Bayes (Diff C)


ROC Curves nonlinsep
True Positive Rate (TPR)
False Positive Rate (FPR)
Bayes (Same C)
Bayes (Diff C)
N. Bayes (C=sigma2I)
N. Bayes (Same C)
N. Bayes (Diff C)


ROC Curves real
True Positive Rate (TPR)
False Positive Rate (FPR)
Bayes (Same C)
Bayes (Diff C)
N. Bayes (C=sigma2I)
N. Bayes (Same C)
N. Bayes (Diff C)







From these plots, we notice that –

- Contours are ellipses, and the eigenvectors of covariance matrix form the axes. Their values give their relative magnitude of major vs minor axis
- Case 1 and 2 have tilted ellipses, rest have axis-aligned ellipses. Case 3 in particular has circular contours. This is because of the non-zero off-diagonal values in covariance matrix, and equal variances for Case 3.
- The decision boundaries are piecewise linear (Case 1,3,4) or hyperquadratic (Case 2,5), as derived in class due to equal and unequal covariance matrices across classes respectively. The regions are sometimes disjoint for the same class, but this is a result of 3-class classification.
- The ROC and DET curves tend to show that Bayes with diff cov mats for each class is the best, and Naive-Bayes with diff cov mats trailing behind. Considering class wise information and covariance of feature columns with each other has helped in better classification.
- It can be proved that the Bayes classifier developed here can always classify linearly separable data with 100% accuracy. The definition of linearly separable implies 3 lines forming the decision boundaries, and given those lines, one can calculate Gaussians that cause such a decision boundary.
- For non-linearly separable and real data, this method need not give 100% accuracy since the best order of the decision boundary achievable by the unimodal Gaussian model is hyperquadratic.