Indian Institute of Technology, Madras

Submitted by: **Vedant Saboo (CS19B074), K V Vikram (CS19B021)**

CS 5691 Pattern Recognition and Machine Learning

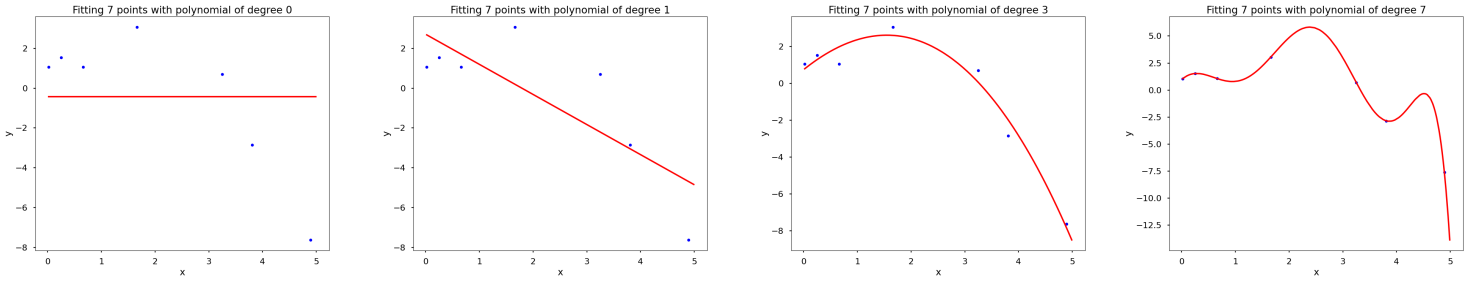
Assignment 2

March 3, 2022

(A) Regression

Experiment 1

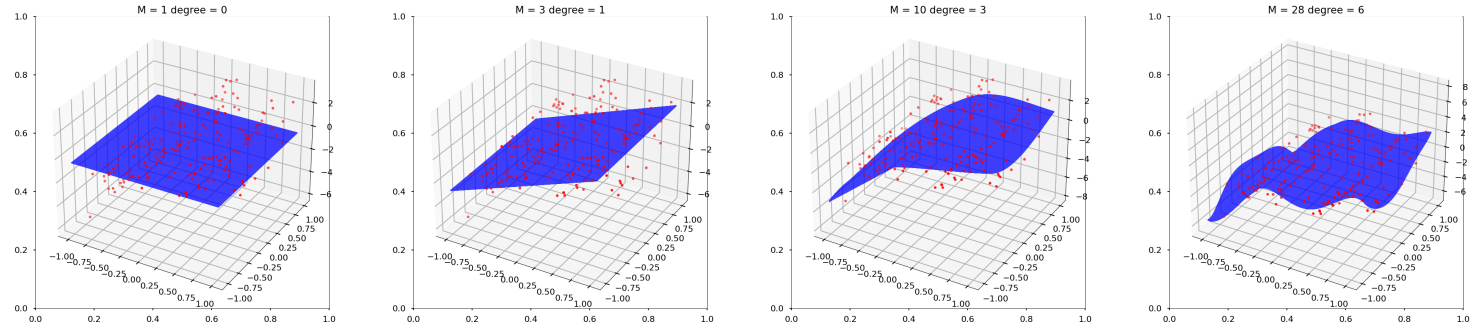
We performed regression with seven training points and with various values of hyperparameter M (size of the weight vector). In figure 1, we describe the some of the under-fitted, over-fitted and properly-fitted results for 1D data.



degree = 0 degree = 1 degree = 3 degree = 7

Figure 1: 1D - Least Squares - Different degree polynomial fitted with 7 data points, cases of under-fitting and over-fitting.

Figure 2 shows the results in case of 2D dataset.



degree = 0 degree = 1 degree = 3 degree = 6

Figure 2: 2D - Least Squares - Different degree polynomial fitted with 200 data points, cases of underfitting and overfitting.

Figure 3 shows the RMS Error trend in the training and development sets, as the parameter M varies. This is for 1D dataset.

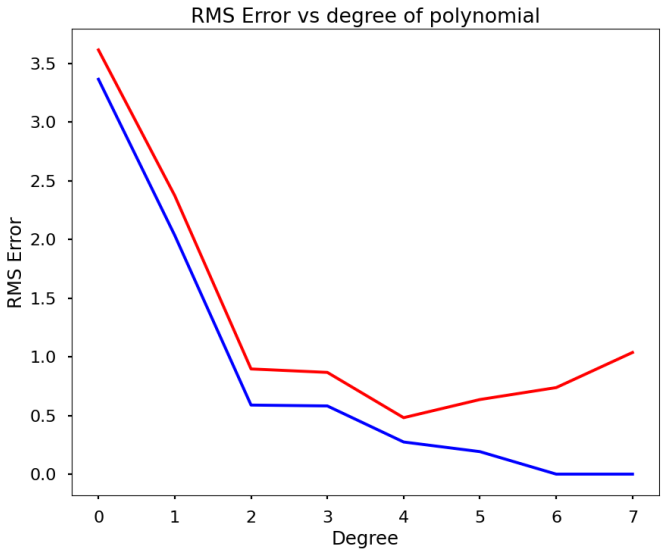


Figure 3: RMS Error vs degree plots for training (blue) and development (red) data (1D dataset)

Figure 4 shows the RMS Error trend in the training and development sets, as the parameter M varies. This is for 2D dataset.

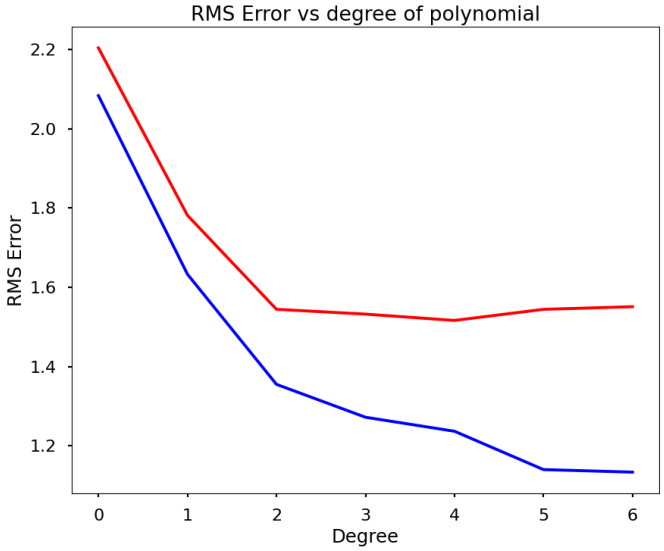
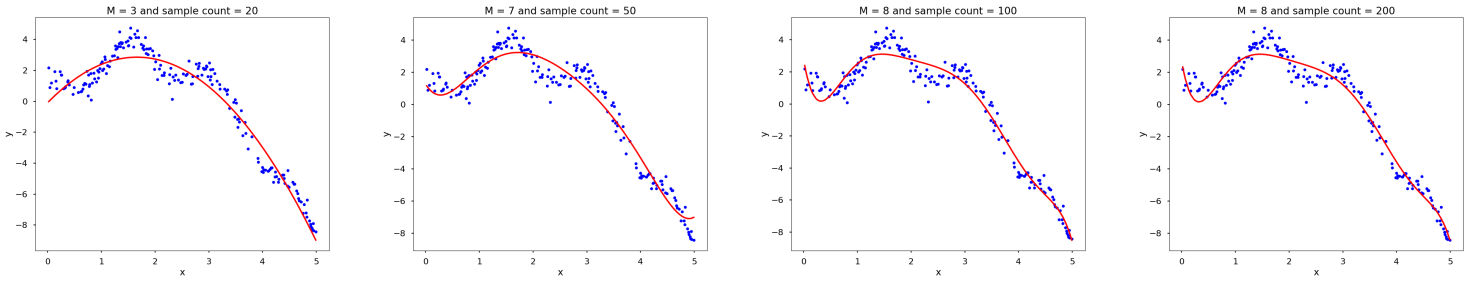


Figure 4: RMS Error vs degree plots for training (blue) and development (red) data (2D dataset)

Experiment 2

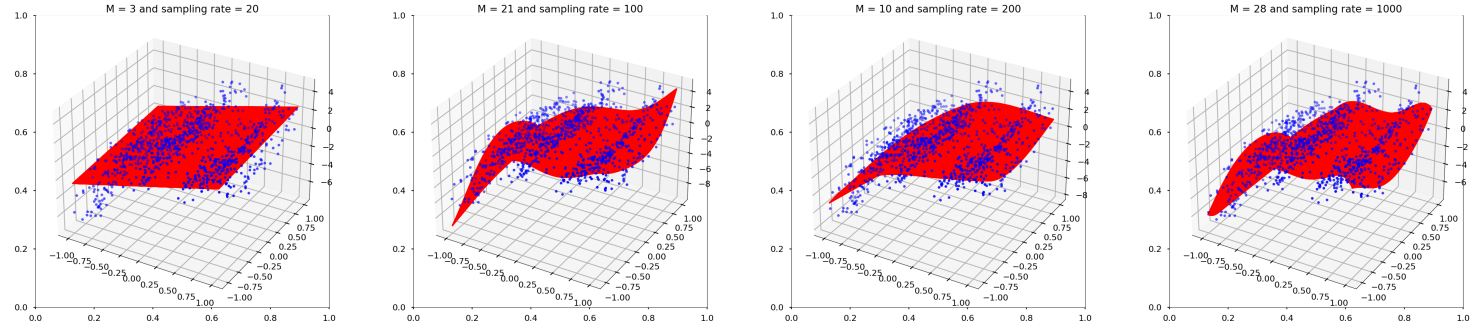
We now expand the training set, i.e, vary the number of training samples, and we choose the M with the lowest RMS Error. Figure 5 shows the results for 1D dataset.



sample size = 20 sample size = 50 sample size = 100 sample size = 200

Figure 5: 1D - Least Squares - Different sample sizes of training data

Figure 6 shows the results for the same experiment with 2D dataset.



sample size = 20 sample size = 100 sample size = 200 sample size = 1000

Figure 6: 1D - Least Squares - Different sample sizes of training data

Figure 7 shows the RMS value affected on the development set by the varying sample size.

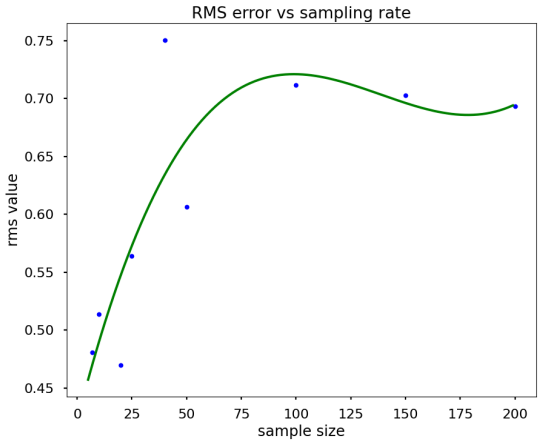


Figure 7: RMS Error vs sample size for 1D dataset

This plot is not the kind we expected for real data. This is because this is not real data, and the noise is not completely random in this dataset. To produce good results in this case, we divided the dataset into several batches of mentioned sizes, and took the average of corresponding coefficients over all the batches. As a result, even smaller sample sizes are performing better, even better than large ones.

Figure 8 shows the same plot for 2D dataset.

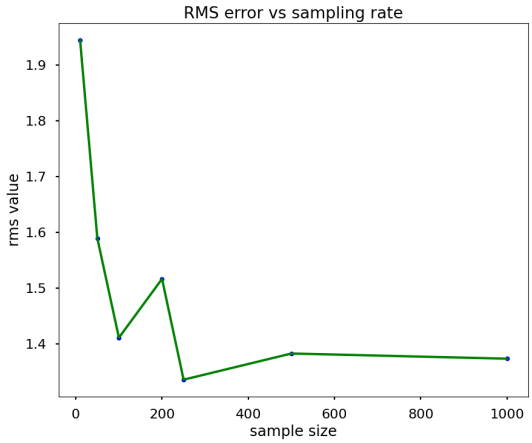


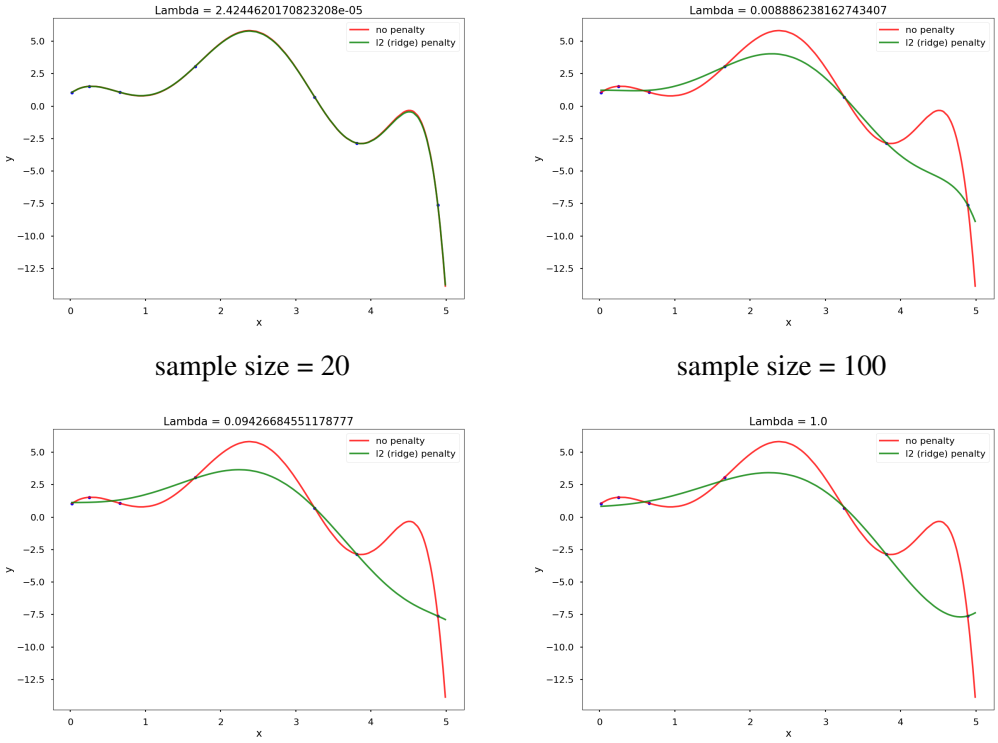
Figure 8: RMS Error vs sample size for 2D dataset

It can be seen that the above 2D dataset diagram represents expected behaviour more closely than the one for 1D dataset. This, however, is purely due to the nature of the data, and we don't expect this in general.

Experiment 3

We introduced regularization with quadratic error (ridge regression). Using several values of regularization parameter λ, we observed the changes it made in the fit polynomial of different degrees. As results, we present here with 4 values of λ used on different degrees of polynomial.

Figure 9 shows 1D dataset polynomial fit with 7 points and degree 6.

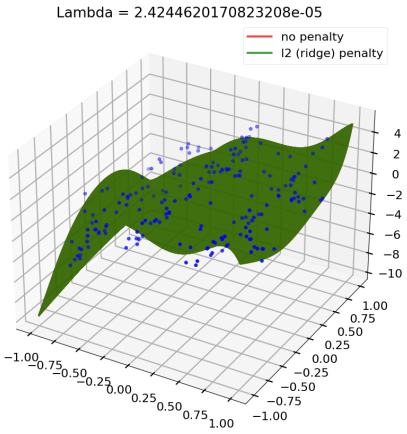


sample size = 20 sample size = 100

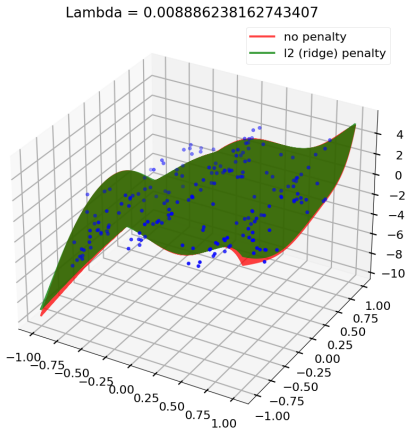
sample size = 200 sample size = 1000

Figure 9: 1D - Difference in the least squares approach and ridge regression.

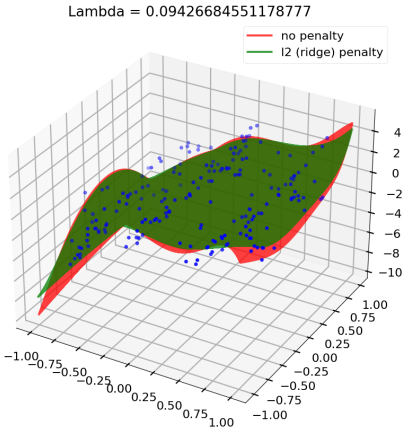
Similarly, we computed the regression fit for 2D dataset. Figure 10 shows polynomial fit on 2D dataset with 200 points and degree 6.



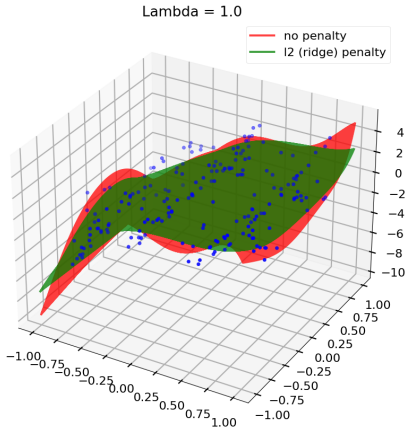
sample size = 20



sample size = 100



sample size = 200



sample size = 1000

Figure 10: 2D - Difference in the least squares approach and ridge regression.

From above figures, it is clearly evident how ridge regression uses l2 penalty to avoid the over-fitting that we observed in plain least square regresssion.

Figure 11 shows how RMS error reduces the error on development data for different values of lambda on 1D data. It reached its lowest when log lambda was somewhere between -2.5 and 0.0.

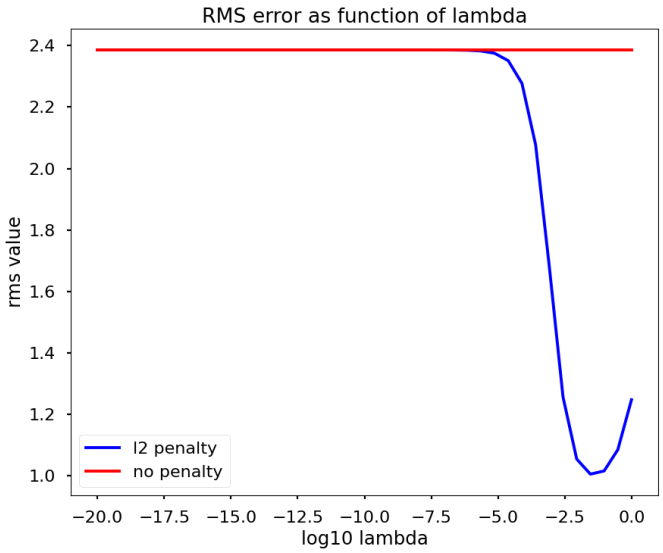


Figure 11: RMS Error vs log lambda for 1D dataset

Figure 12 shows the same statistic for 2D dataset.

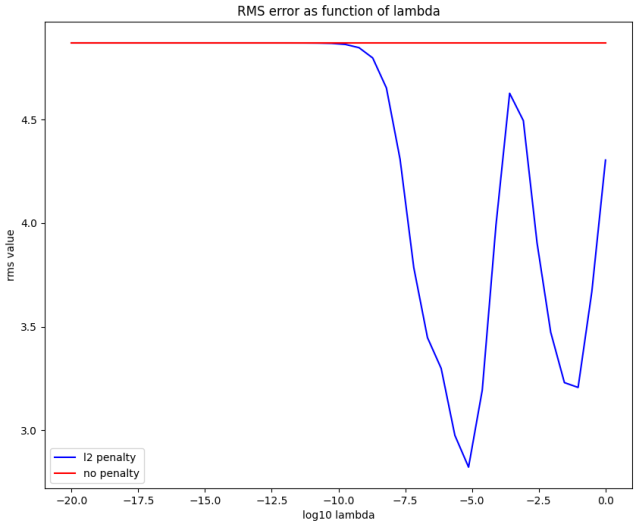


Figure 12: RMS Error vs log lambda for 2D dataset

In this paper we looked at linear regression, and the various results obtained by changing

1. the hyperparameter M, which is the dimension of the weight vector
2. the hyperparameter λ, which is the regularization constant
3. the sample sizes for training the model

We also noted that the noise in the data wasn't completely random, in which case, we decided to split the training data randomly into several smaller samples, and then find the polynomial fit for each of them, and finally average the corresponding coefficients of the weight vector.

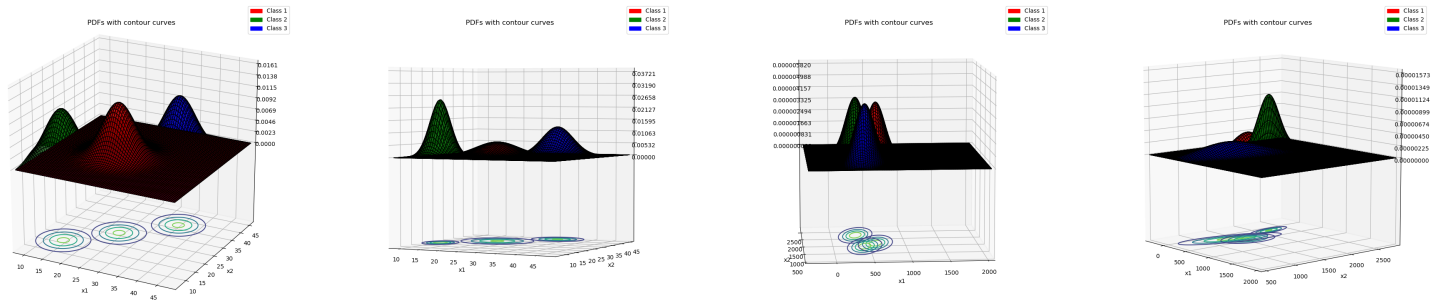
After that, we fine-tuned the hyperparameters by evaluating it on development data. Note that in all observations in this paper, the seed of the randomizer was fixed (value 6), to maintain consistency of the results. In practise, we prefer to not specify the seed.

(B) Bayesian Classifier

We use the abbreviations - LS for Linearly Separable Data, NLS - Non Linearly Separable Data, and RE for Real Data. We also use the Ci, as a data type suffix, for indicating that the plot belongs is for Case i as in the assignment statement. So, NLSC1 indicates Case 1 of NLS data. The most exciting plots were generated by Real Data and hence, we will focus on it. From training data, we see that the prior is an uniform distribution on the classes.

Probability Density Functions:-

We will first look at some sample PDFs from what we plotted for the 5 cases and 3 types of data.

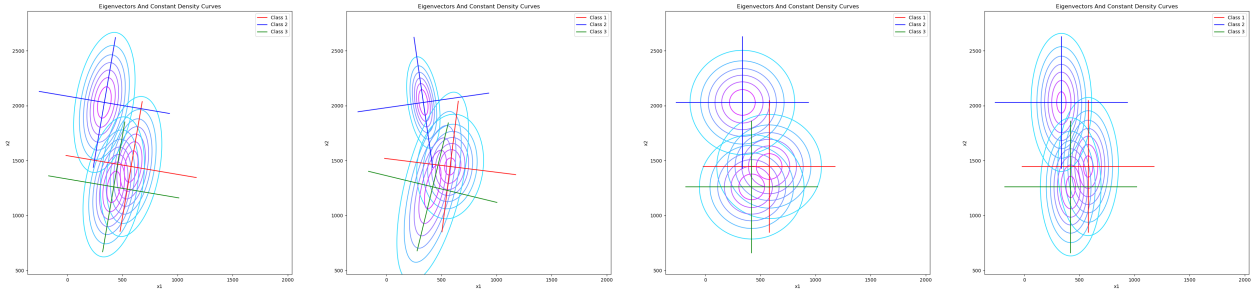


NLSC3 NLSC5 REC1 REC2

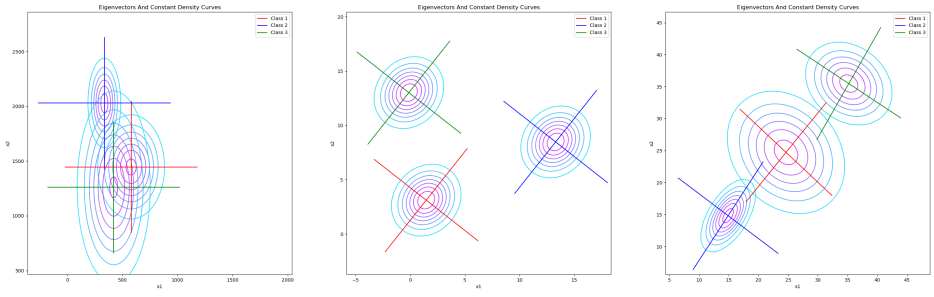
Figure 13: Probability Density Function Plots

From figure 13, we can see that a) for NLSC3 and REC1, all 3 class pdfs share the same shape (they are just shifted forms of each other), b) from NLSC5 and REC2, we can see that a large covariance determinant leads to a flatter PDF.

Eigenvector and Contour Plots:-



REC1 REC2 REC3 REC4

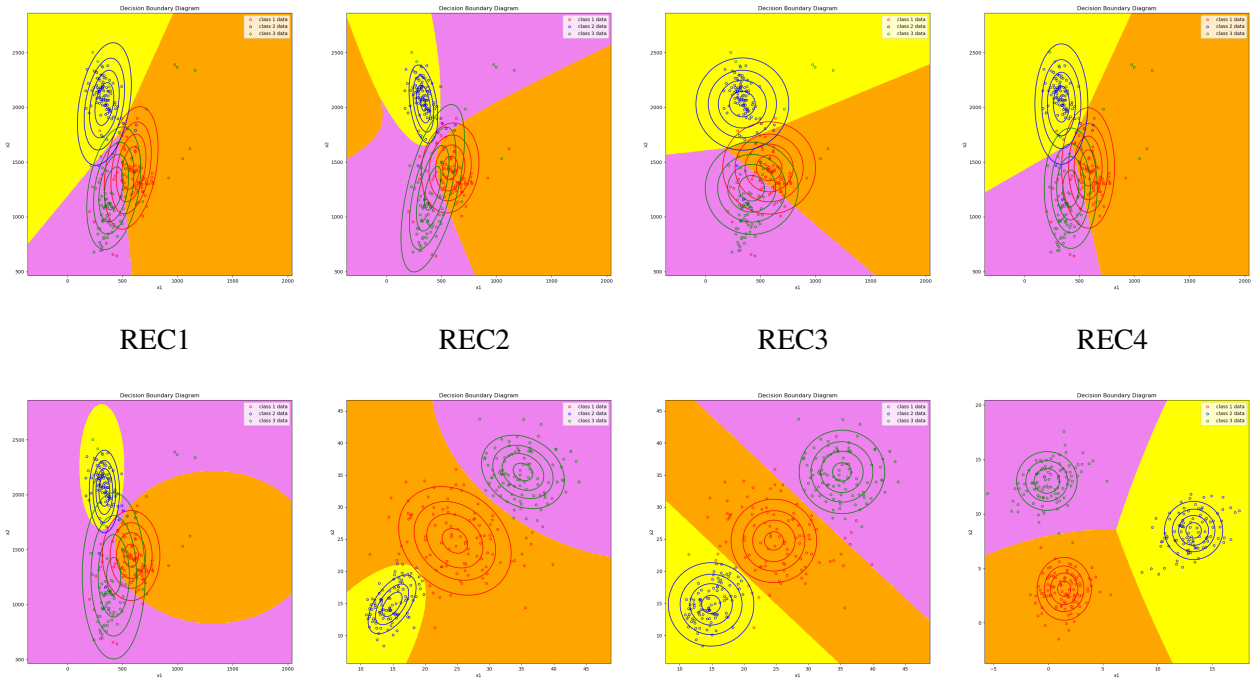


REC5 LSC1 NLSC2

Figure 14: Eigenvector and Contour Curves Plots

From figure 14, we note the following. For cases 1, 3, 4, where all classes share a covariance matrix, the contour ellipses of each class are the same (just shifted) and the eigenvectors of the covariance matrix are parallel. In cases 3, 4, 5, where we have a diagonal covariance matrix, the eigenvectors of every class is parallel to the co-ordinate axes. Also, a well-spread (large covariance determinant) indicates contours are more separated and enclose a larger area.

Decision Boundary Diagrams:-



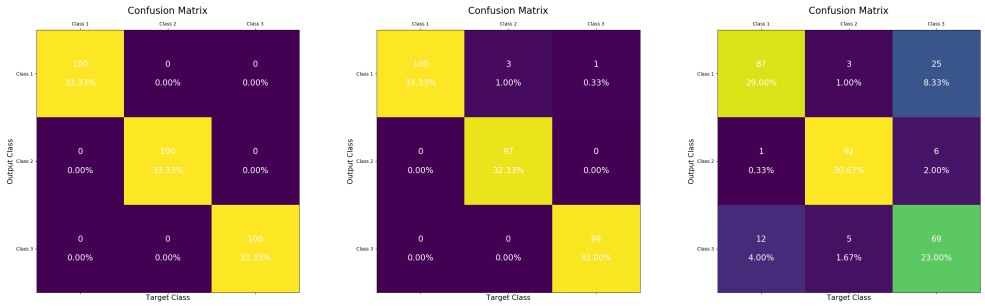
REC1 REC2 REC3 REC4

REC5 NLSC2 NLSC4 LSC5

Figure 15: Decision Boundary Diagrams

From figure 15, we can make several observations. We can see that quadratic decision boundaries will occur only when all classes are allowed to have different covariance matrices (cases 2 and 5). In cases 1, 3, 4, all decision boundaries are linear. In case 3, as the common covariance matrix is a multiple of the identity matrix, the contour curves of each class are concentric circles, the decision boundary between any 2 adjacent classes passes through the midpoint of their means and it is also perpendicular to the line joining the means. In cases 1, 4, where the classes just have a common covariance matrix, the decision boundary of 2 adjacent classes only passes through the midpoint. Classification using this diagram is also trivial - any development data point will belong to the class in whose shaded-region it lies.

Confusion Matrices:-

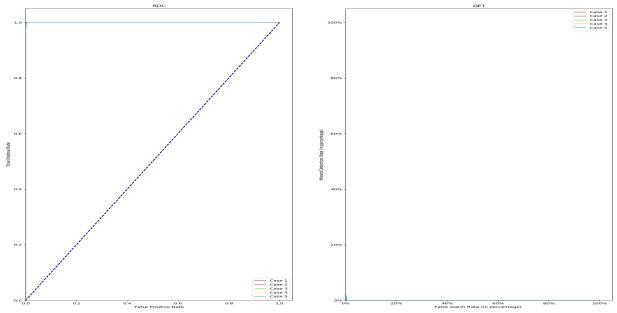


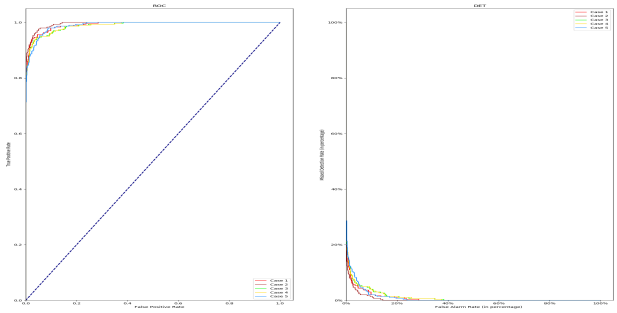
LSC5 NLSC2 REC5

Figure 16: Confusion Matrices

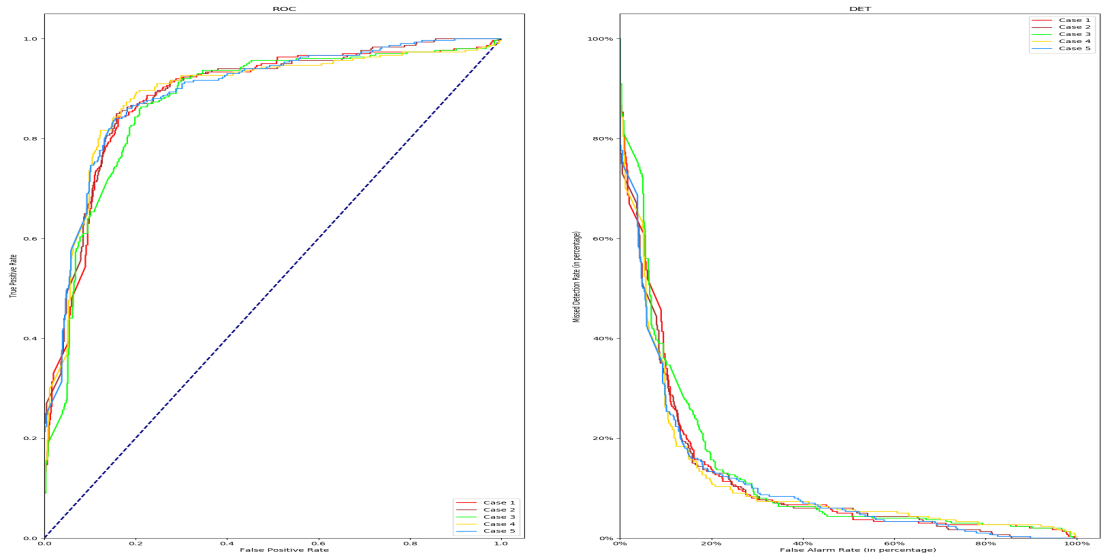
For LS data, the confusion matrices were all the same as LSC5. No matter which technique we use, all classifiers get a 100% accuracy. For NLS and RE data, the misclassification rates are quite high (especially for RE) which is indicated by the off-diagonal entries being non-zero (quite large numbers for RE).

ROC-DET curves:-





LSROCDET NLSROCDET



REROCDET

Figure 17: ROC-DET Plots

For LS data, all the 5 cases have a good ROC. Excellent operating points, which are close to $(0, 1)$, can be picked for each of the 5 cases. This is because of the fact that the data is linearly separable.

For NLS data, the ROC curves for all cases are now less than ideal. From the ROC curve, it can be seen that case 2 (the most relaxed assumption case) is a better choice than the other cases. A good operating point, determined by looking at the DET curve (as this provides a better look at the error rates), can be taken around $(4\%, 4\%)$.

For RE data, the ROC curves of all cases are far from ideal. No case performs comfortably better than the others. This is primarily because of the fact that this is actual data extracted from real sources. This data exhibits the general trend of the ROC curve in real-life classification scenarios, i.e, at large thresholds - both TPR and FPR are close to 0 (as no data is classified as Positive), with a decrease in threshold - both TPR and FPR increase and at small thresholds - both TPR and FPR are close to 1.