CSE 574 - Introduction to Machine Learning

Programming Assignment 2
Classification and Regression

Submitted by :

1) Madhur Gupta (UBIT Name : madhurgu)
2) Mohit Arora (UBIT Name : marora3)
3) Abhijit Pattanaik (UBIT Name : abhijitp)

Report 1.

1) **Linear Discriminative Analysis (LDA)** - It is a generalization of Fisher's linear discriminant which is used to classify given points into respective classes defined by linear boundaries between them. It is a generative classifier which tries to maximize the distance between the discriminating classes using linear method. Here in this assignment we are given points divided into 5 classes and we have to classify them into their respective classes. So when we use LDA we can see that there is line separating different points into their respective classes. We get linear decision boundary separating classes because in LDA we assume a fixed covariance for Gaussian distribution among all the classes. This can be seen in figure below:-

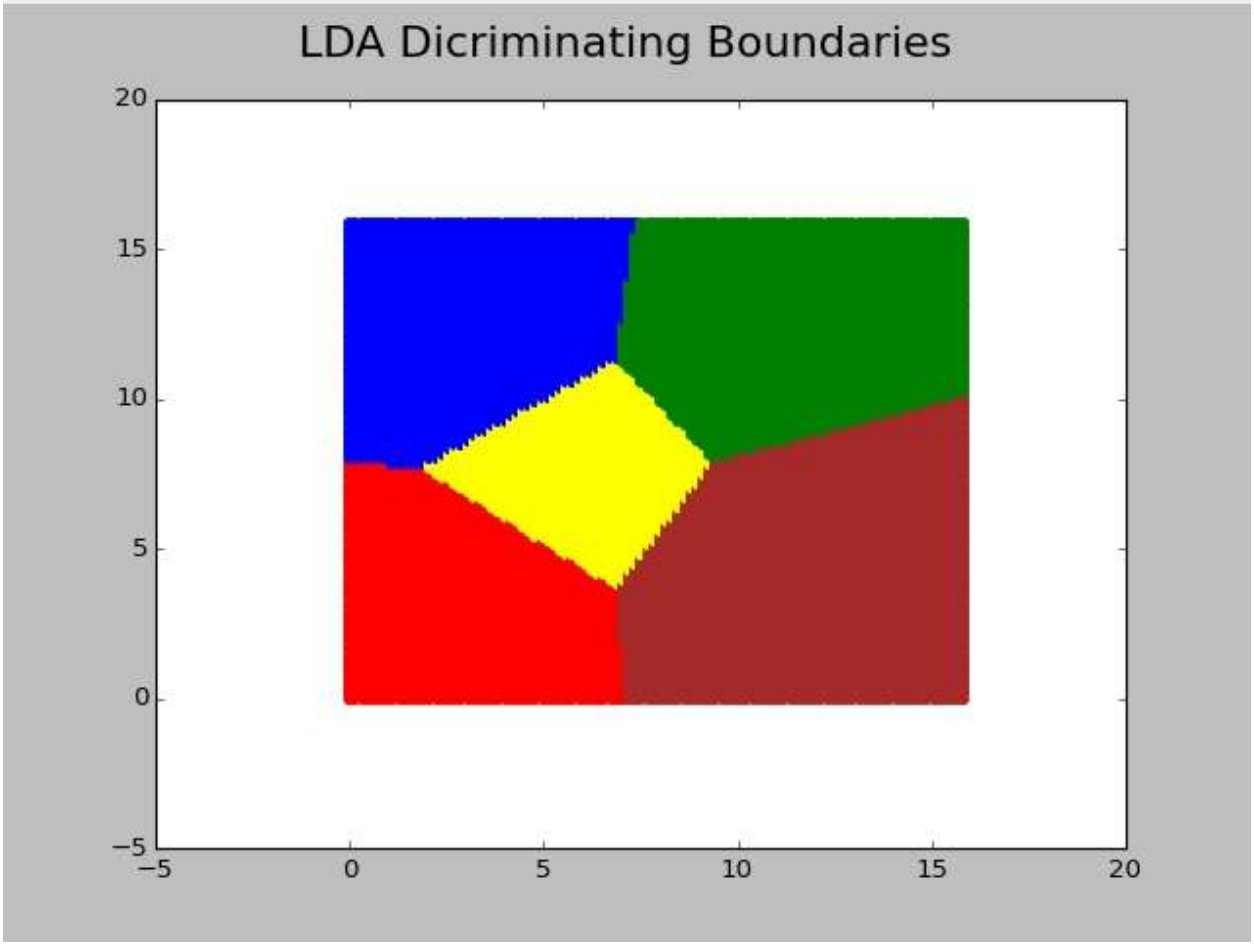

Figure-1 Linear Discriminative Analysis

Accuracy: 97%

2) **Quadratic Discriminative Analysis (QDA)**: - This is also a generative classifier which is used to separate out the given data into separate classes. Here instead of getting a linear boundary between the classes we get a curve differentiating different classes and this happens because we are not assuming that there is a fixed covariance for all the classes present in the data set. With different covariance for different class we get different Gaussian spread which leads to curved boundary separation. As it can be seen from the figure below:-

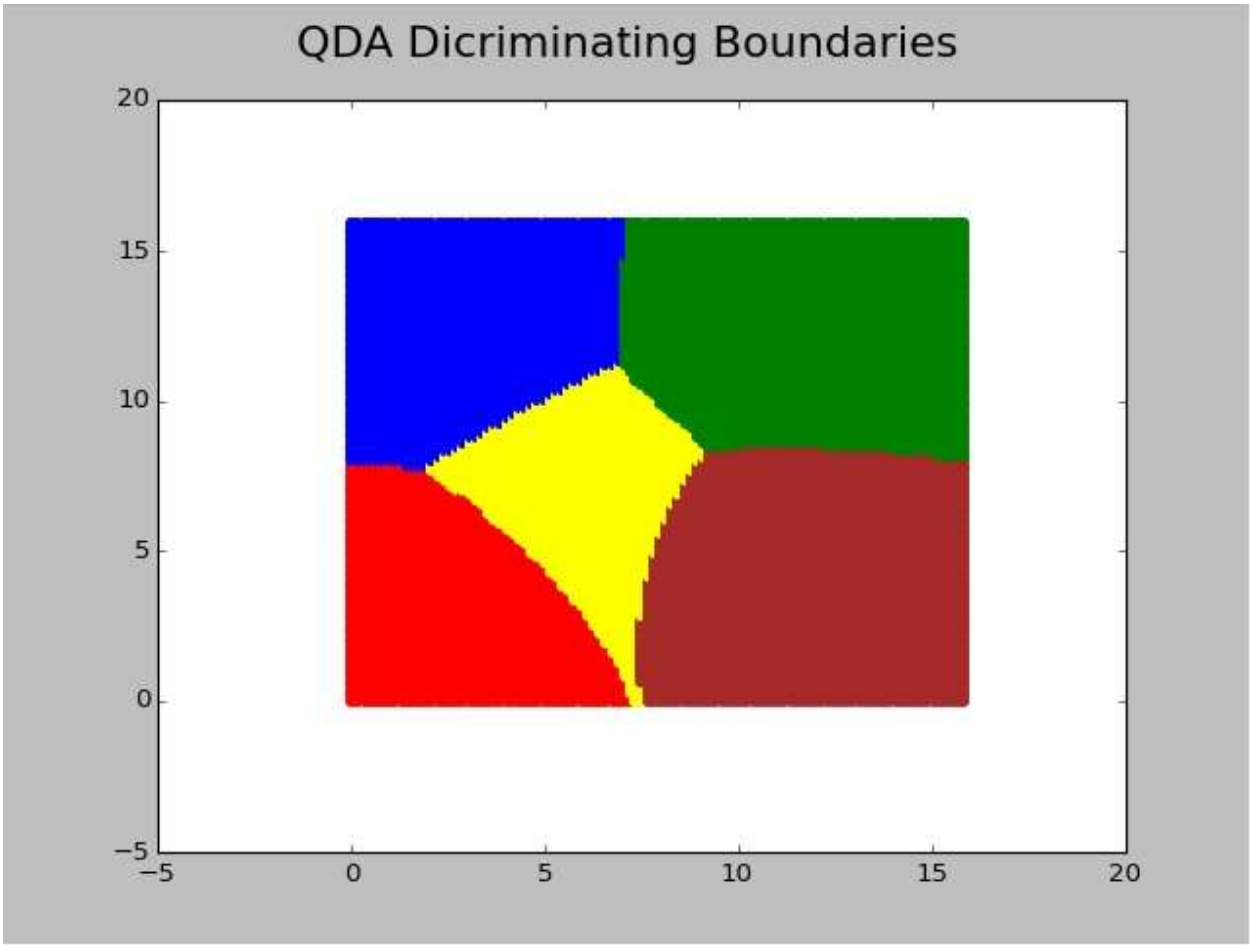

Figure 2:- Quadratic Discriminating Boundary

Accuracy: 96%

Report 2.

The following is the result of the experiment with linear regression:

For Training Data:
RMSE without intercept 8.88388057487
RMSE with intercept 3.0063021236

For Test Data:
RMSE without intercept 23.1057743391
RMSE with intercept 4.30571723503

It is evident from the report generated above, that the RMSE with intercept is better since it incorporates the bias.

Report 3.

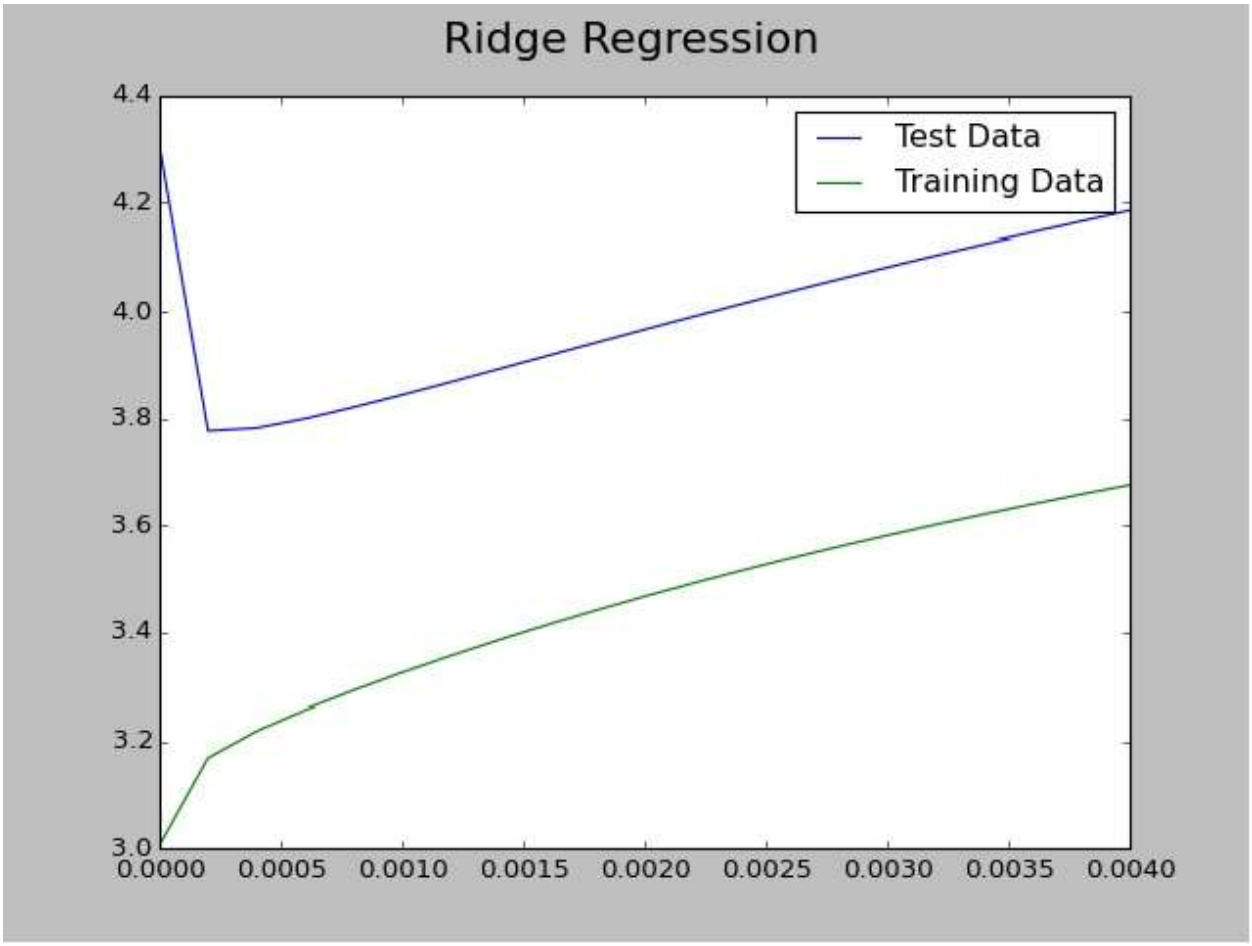

Figure 3:- Ridge Regression

Figure 3 shows the graph of lambda vs. RMSE on x-axis and y-axis respectively. As it can be seen, when the value of lambda is zero, there is no regularization, and hence the outliers in the data result in over fitting problem. And as the value of lambda increases, the lambda term becomes part of the objective function J (w) which is being minimized during the learning process, so lambda helps restrict the growth of the weight vector during learning and hence negates the over fitting problem. Now, the RMSE value increases once again when lambda goes beyond the value of 0.00020, and this happens because as lambda grows higher, it starts putting a tighter bound on the growth of the weights and hence resulting in an under fitting problem. So the optimum value of lambda for this experiment is 0.00020.

Minimum value of weight for OLE: -1358916.1222826946
Maximum value of weight for OLE: 1194622.6321152381
Maximum value of weight for Ridge Regression: -113.0259459832773
Maximum value of weight for Ridge regression: 206.7906174207713

Report 4.

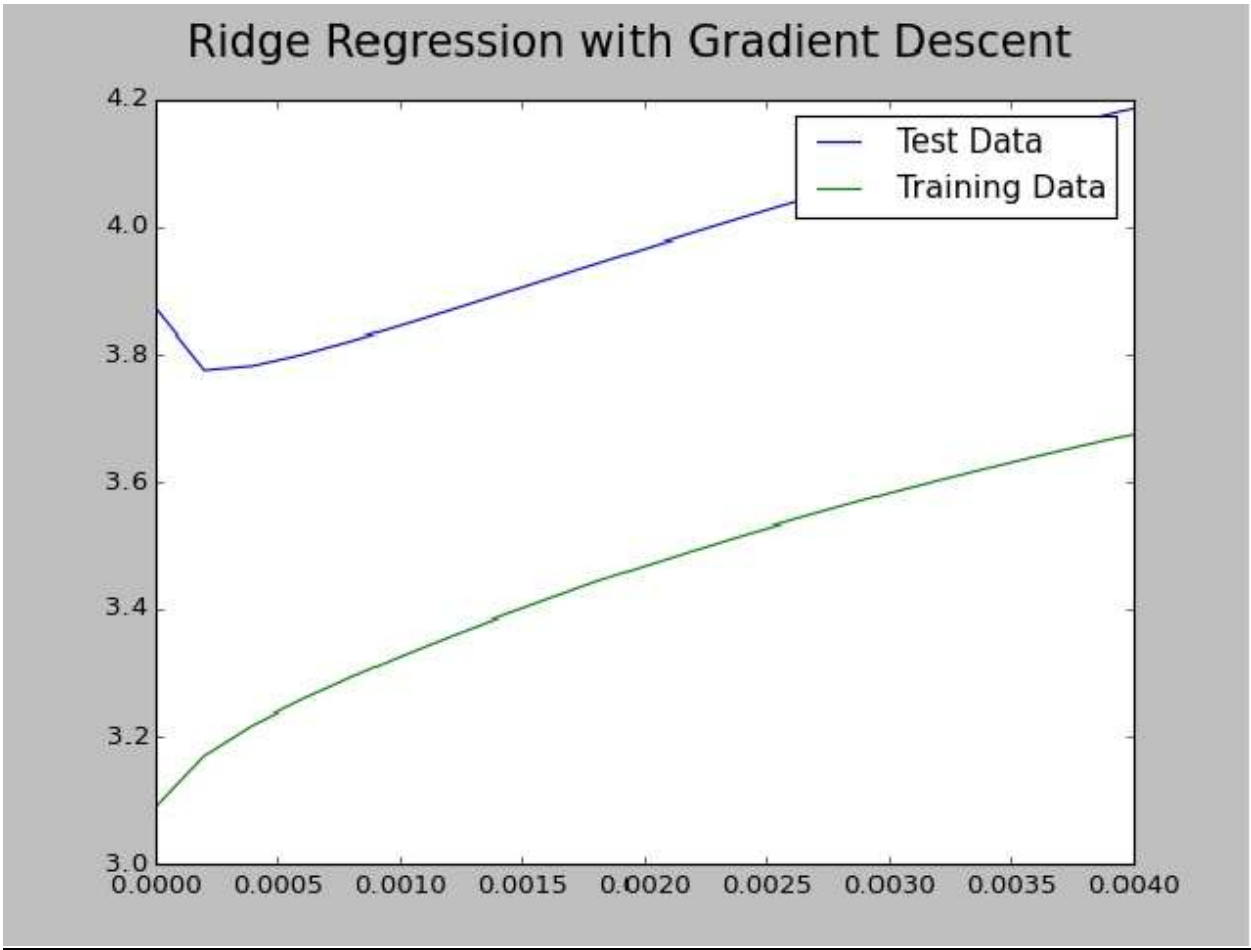

Figure 4:- Ridge Regression using Gradient Descent

Figure 4 shows the graph of lambda vs. RMSE on x-axis and y-axis respectively. As it can be seen, when the value of lambda is zero, there is no regularization, and hence the outliers in the data result in over fitting problem. And as the value of lambda increases, the lambda term becomes part of the objective function J (w) which is being minimized during the learning process, so lambda helps restrict the growth of the weight vector during learning and hence negates the over fitting problem. Now, the RMSE value increases once again when lambda goes beyond the value of 0.00020, and this happens because as lambda grows higher, it starts putting a tighter bound on the growth of the weights and hence resulting in an under fitting problem. So the optimum value of lambda for this experiment is 0.00019.

Results of 3 and 4^{th} are same. Using gradient descent is better approach as it requires less computation for calculating weights.

Report 5.

For various values of p and lambda, root mean square error (rmse) on test data can be found from the below table.

p	λ = 0	λ = 0.0002
0	5.60642702	5.60659858
1	4.38465206	4.40623928
2	4.41991408	4.40616766
3	4.40906767	4.40616843
4	4.71345303	4.40616843
5	4.77222714	4.40616843
6	5.84527978	4.40616843

Root Mean Square Error for various values of λ and p

From above table we can see that for λ=0 optimal value of p comes p=1 and for λ=0.0002 optimal value of p comes p=. For λ = 0, root mean square error decreases till p=1 and then it starts increasing. For λ=0.0002, root mean square error decreases till p=2 and then it becomes steady from p=3 to p=6.

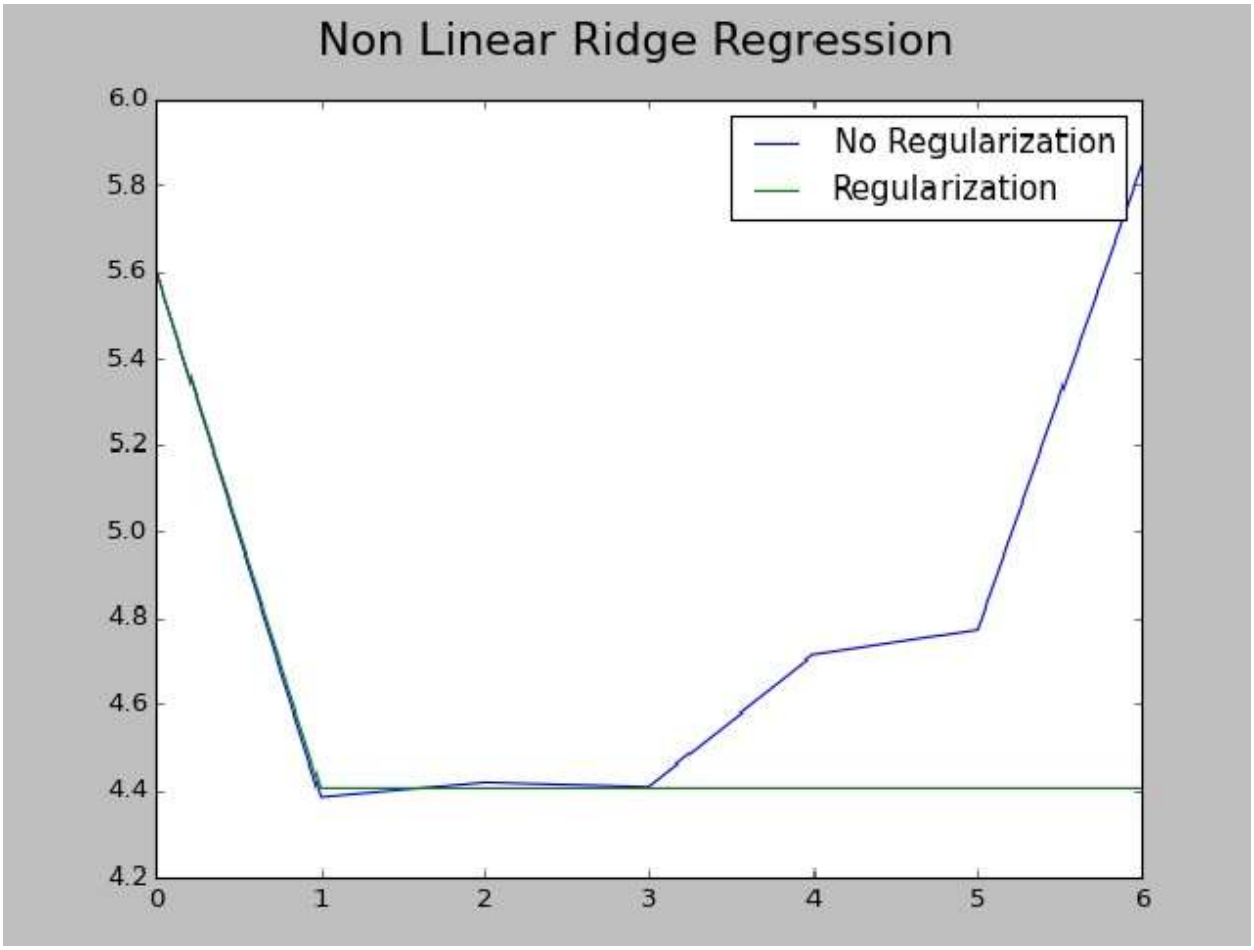

Figure 5:- Root Mean Square Error vs P for λ=0 and λ=0.0002

Report 6

Comparison of above four regressions can be done based on following categories.

a) **Computational Complexities:**

Performing linear regression, Ridge Regression involves highly computational tasks of calculating matrix inverses for learning the weights. As size of the data grows, calculating inverse becomes more and more expensive.
When we are minimizing the error function using gradient descent, learning weights does not involve the calculation of matrix inverses. So this is not a computationally expensive task.

Similarly non-linear regression uses ridge regression for learning weights and involves matrix inverses for calculating weights. So this is also highly expensive computational task.

b) **Error on Test data vs Train data**

Linear regression, ridge regression and gradient descent have less error compared to non-linear regression on test data. This is the case because our non-linear regression is performed based on single attribute of an input. If non-linear regression would have been performed based on all attributes of each input, non-linear map would have least error as compared to other regression.

When using non-linear regression, weights are learned in such a way that error should be minimum on train data. So when no-regularization is used, non-linear regression will have tendency towards over fitting the input data. This might cause more error on train data.
But when we use regularization in non-linear regression, it overcomes the problem of over fitting by penalizing the loss function.