University at Buffalo
The State University of New York

CSE 574 Introduction to Machine Learning

Programming Assignment 2

Classification and Regression

Submitted by:

Debika Dutt: **50170009**

Mahesh Venkataramaiah: **50170332**

Vardhana Srinivas Kulkarni: **50169374**

Group# 7

Problem 1 Report 1: Experiment with Gaussian discriminators

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA)

Accuracy

a.	Linear Discriminant Analysis	97%
b.	Quadratic Discriminant Analysis	97 %

Discriminating boundary for Linear and Quadratic discriminators

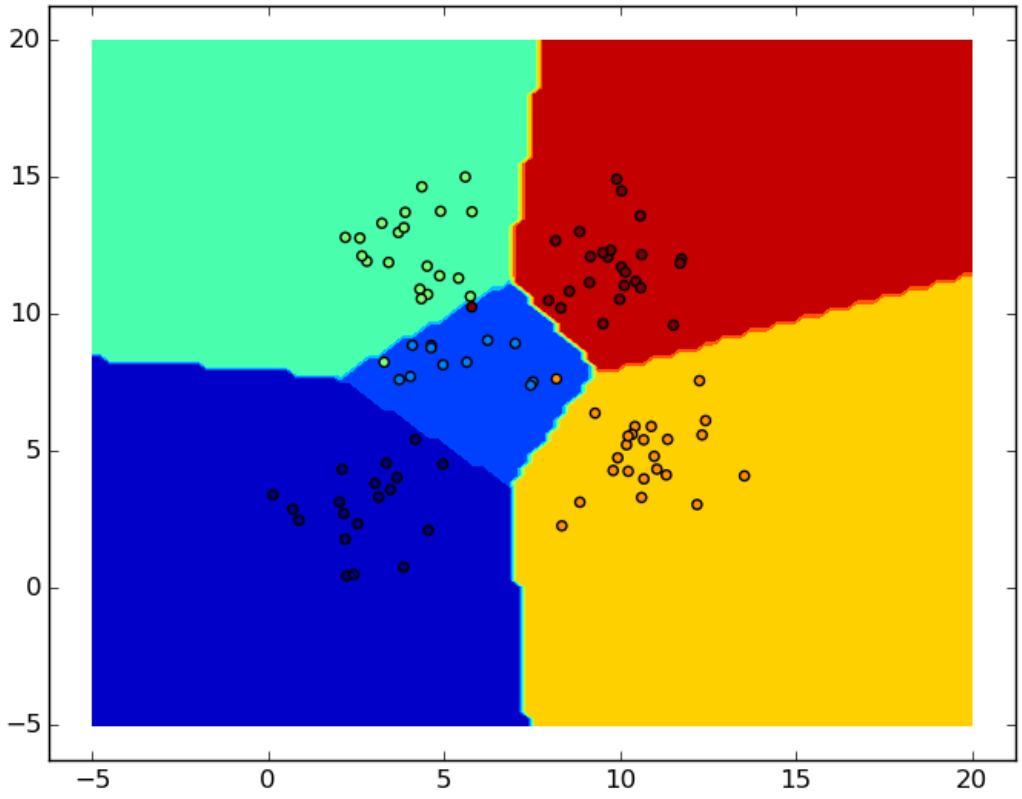

Fig 1.1 Linear Discriminant Analysis (LDA)

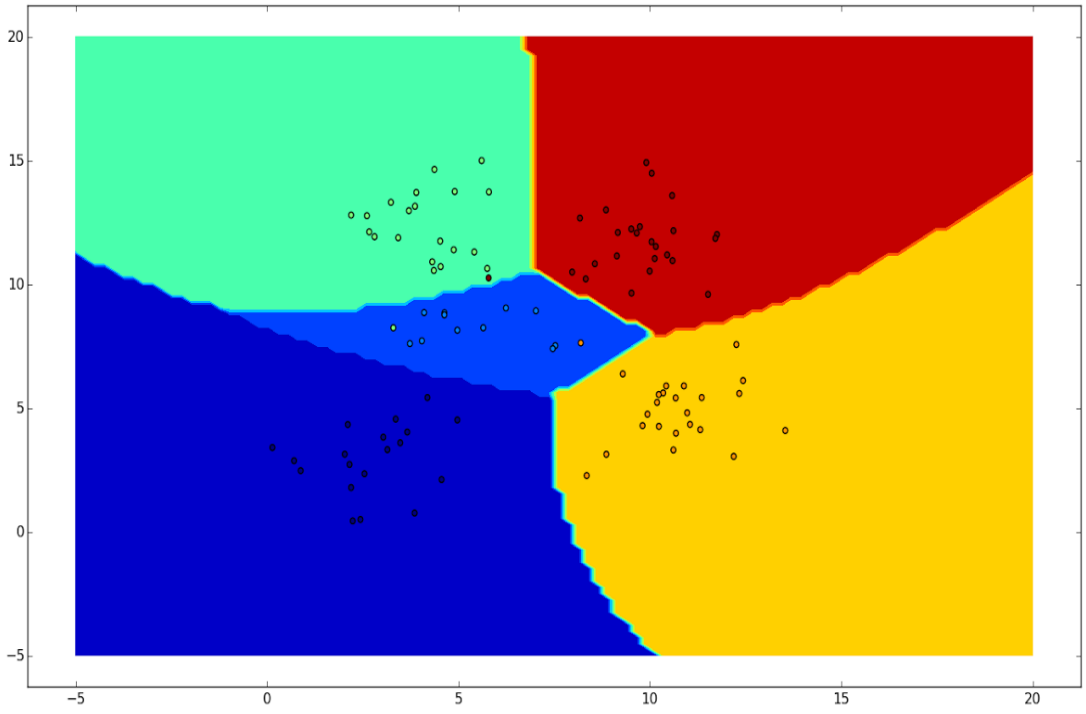

Fig 1.2 Quadratic Discriminant Analysis (QDA)

Difference between Boundaries

QDA decision boundary looks more flexible than LDA boundary in the above shown plots. LDA can detect only linear boundaries and QDA can detect even the quadratic boundaries. This is because the covariance matrix is calculated for entire training data in the case of LDA and covariance matrix is calculated for training data belonging to each class in the case of QDA. If the values for the same class is distributed in non-linear way, LDA fails to identify the classes effectively.

Implementation Details

Linear Discriminant Analysis(LDA)

ldaTest function implements the following equation to classify the values.

$$\mu_c^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c \text{ -}$$

Quadratic Discriminant Analysis(QDA)

qdaTest function implements the numerator part of the following equation to classify the data as the denominator is same for all the classes.

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{\pi_c |2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c)\right]}{\sum_{c'} \pi_{c'} |2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{c'})^T \boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{c'})\right]}$$

Problem 2 Report 2: Experiment with Linear Regression

Comparison of RMSE for training and test data with intercept and without intercept

RMSE for train data without intercept	138.2007
RMSE for train data with intercept	46.76709
RMSE for test data without intercept	326.765
RMSE for test data with intercept	60.89204

RMSE for the data with Intercept (or bias) is lesser than that of data without intercept as we can see from the above observations for both training and test data. Adding intercept reduces the error.

RMSE for the training data is less than that of corresponding test data as the weights are calculated for the training data. This holds true when

1. Train data with intercept is compared with the test data with the intercept
2. Train data without intercept is compared with the test data without the intercept.

Also, we can notice that RMSE for the test data with intercept is lesser than the RMSE calculated for training data containing no intercept.

From the above observations, we can notice that the data with intercept gives better results.

Implementation details

Weights in learnOLERegression are calculated using the following equation.

$$\hat{\mathbf{w}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Root Mean Squared Error (RMSE) in testOLERegression is calculated using the following equation

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top\mathbf{x}_i)^2}$$

The table below shows the comparisons of the Train and Test data of lambda values:

	Train Data		Test Data	
P	Lambda = 0	Lambda = 0.06	Lambda = 0	Lambda = 0.06
0	75.1712081777	75.1712172778	79.2868513165	79.2898604296
1	62.6970127463	62.8636550286	62.0083440367	62.416796333
2	62.5447013839	62.8544931757	62.5070243981	62.4146141215
3	62.5394968394	62.8544551446	62.3536329193	62.4146033867
4	62.3335629345	62.8544535956	66.6582919959	62.4146030051
5	62.3330342387	62.8544535827	67.4894834581	62.4146030085
6	62.1842701127	62.8544535824	82.6647394523	62.4146030086

Problem 3 Report 3: Experiment with Ridge Regression

Introduction

Ridge Regression is a technique for analyzing multiple regression data that suffer from multi-collinearity. When multi-collinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable. We implemented parameter estimation for ridge regression by regularizing the squared loss which can be written in matrix-vector notation. The function has been implemented in learnRidgeRegression.

The equation that we used is:

MAP Estimate of W:

$$\widehat{\mathbf{w}}_{MAP} = (\lambda \mathbf{I}_D + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Where:
w = [w1, . . . ,wM], the weight vector (to learn using the training data).
Inputs $\mathbf{x}_i$: D-dimensional vectors (R^D),
Responses $\mathbf{y}_i$: scalars (R)

Penalized linear regression is also known as ridge regression

Observation and Results for the Training and Test Data

After implementing the "*learnRidgeRegression*" method , and upon testing the test samples we get the following results. The graph of variation of Root Mean Square Error with lambda is as shown below.

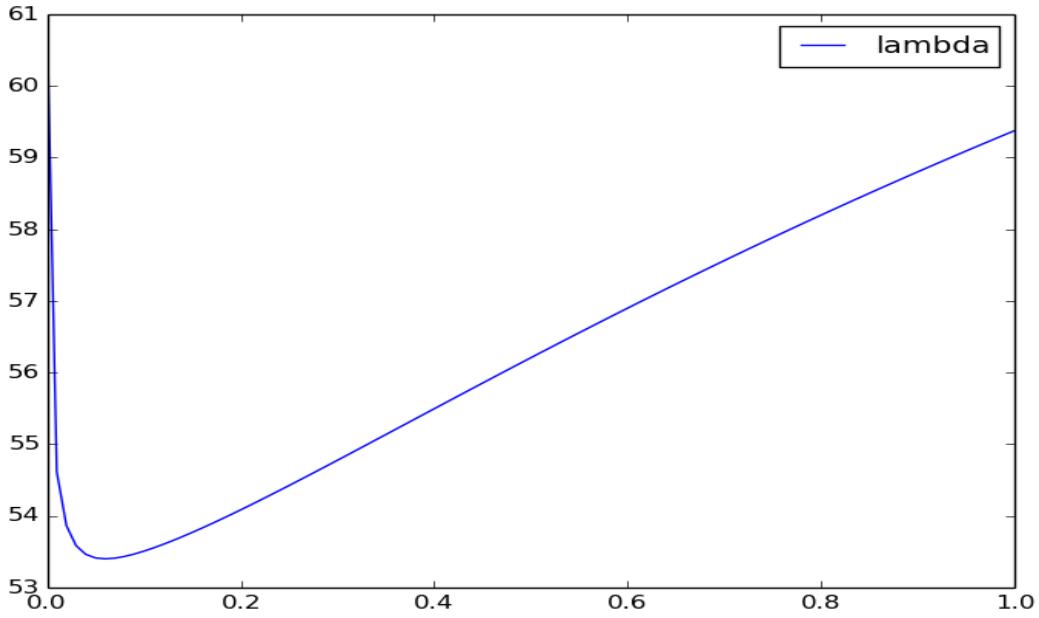

Fig.3.1 Plot of Test data of Lambda vs RMSE values

By observing the results of the ridge regression for the test data set, we observe that the root mean square error initially decreases until a certain value and then it increases. Some values of RMSE for some lambda values in the range 0-1 are shown below.

Lambda	RMSE
0	60.89
0.02	53.86
0.04	53.46
0.06	53.39
0.2	54.08
0.25	54.42
0.3	54.77
0.5	56.2
0.65	57.23
0.92	58.91
1	59.37

Table 3.1: RMSE values varies with Lambda values for Test Data

Upon observation, we find that the root mean square error is minimum for the value of lambda = 0.06.

The result of ridge regression for the training data set is as follows:

Lambda	RMSE
0	46.76
0.02	48.51
0.04	49.11
0.06	49.51
0.2	51.29
0.25	51.81
0.3	52.31
0.5	54.07
0.65	55.19
0.92	56.91
1	57.35

Table 3.2: RMSE values varies with Lambda values for Train Data

The variation of the lambda in the range [0-1] with the root mean square error is as follows.

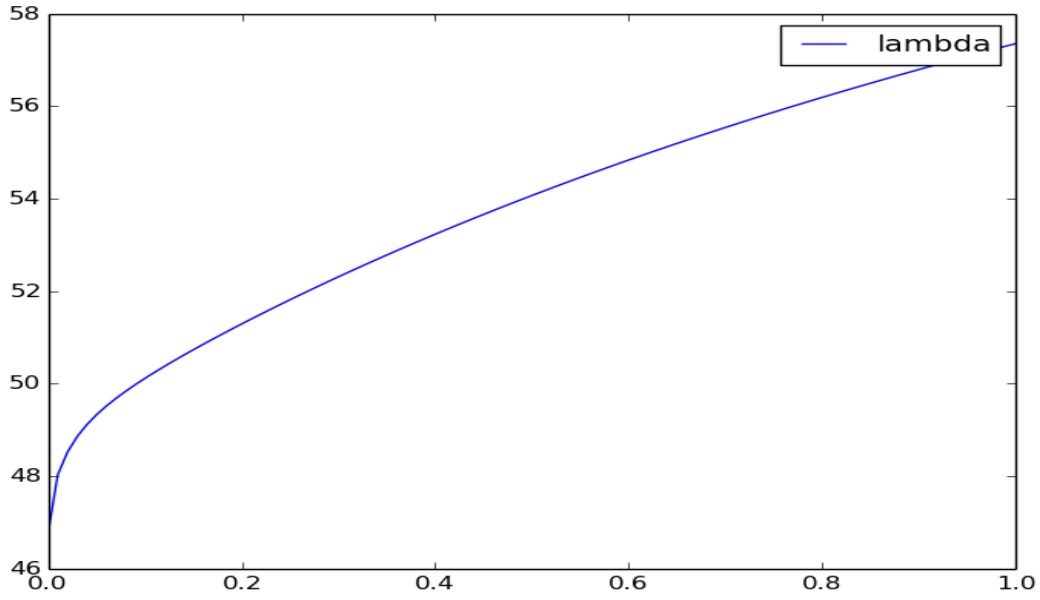

Fig.3.2 Plot of Train data of Lambda vs RMSE values

Comparison of weights learnt during OLE and Ridge regression

We observe that the weights of ridge regression increases with increase in the value of lambda until lambda equals 0.06 and then it decreases. With the value of lambda equal to zero, the weights obtained are same during the OLE as well as Ridge regression.

Due to the few points in each dimension and the straight line that linear regression uses to follow these points as well as it can, noise on the observations will cause great variance as shown in the first plot. Every line's slope can vary quite a bit for each prediction due to the noise induced in the observations.

Ridge regression is basically minimizing a penalized version of the least-squared function. The penalizing shrinks the value of the regression coefficients. Despite the few data points in each dimension, the slope of the prediction is much more stable and the variance in the line itself is greatly reduced, in comparison to that of the standard linear regression.

The Optimal value of lambda obtained is 0.06 as it gives the minimum error.

Comparison of errors during OLE and Ridge regression

We observe that for the test data, the error increases with the increase in the value of lambda as seen in the table 3.2 with the minimum error for lambda = 0.
On the other hand, for the training data, the error decreases up to the value of lambda = 0.06 and further increases with the increase in the value of lambda till 1.

Problem 4 Report 4: Using Gradient Descent for Ridge Regression Learning

Introduction

The regression parameters can be calculated by using gradient descent to minimize the loss function. The weights are estimated using gradient descent. The regularized square error is calculated by the following equation

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

The gradient of the error is calculated as follows

Gradient of error = $(X^TX)W - X^TY + \lambda W$

The variation of error with respect to lambda for the test data is as follows. We observe that the error is minimum at lambda = 0.06.
The error decreases till the value of lambda = 0.06 and further goes on increasing with increase in lambda.

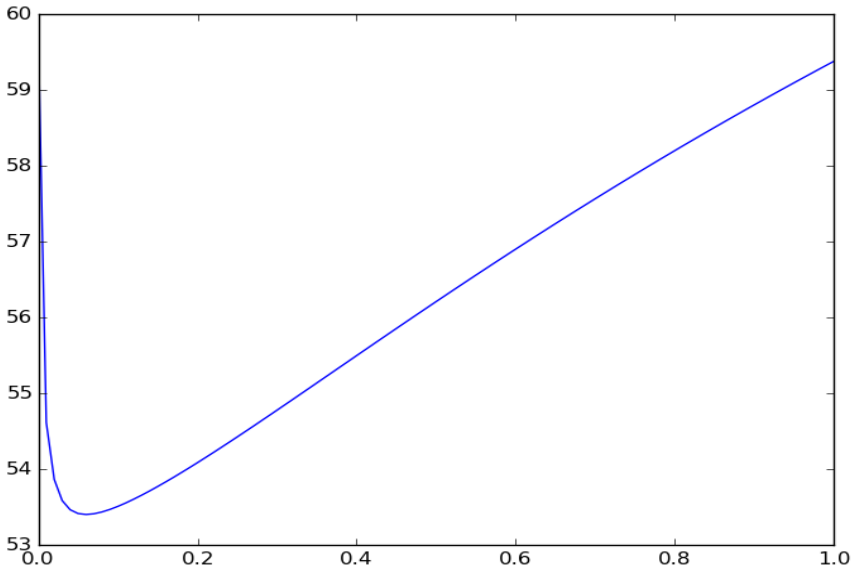

Fig 4.1- Variation of RMSE for Test data with variation of lambda

The variation of error with respect to lambda for the training data is as follows. We observe that the error is minimum at the value of lambda = 0 . Unlike the test data, here we do not observe initial decrease in the value of error with lambda. With the increase in lambda, the RMSE also increases.

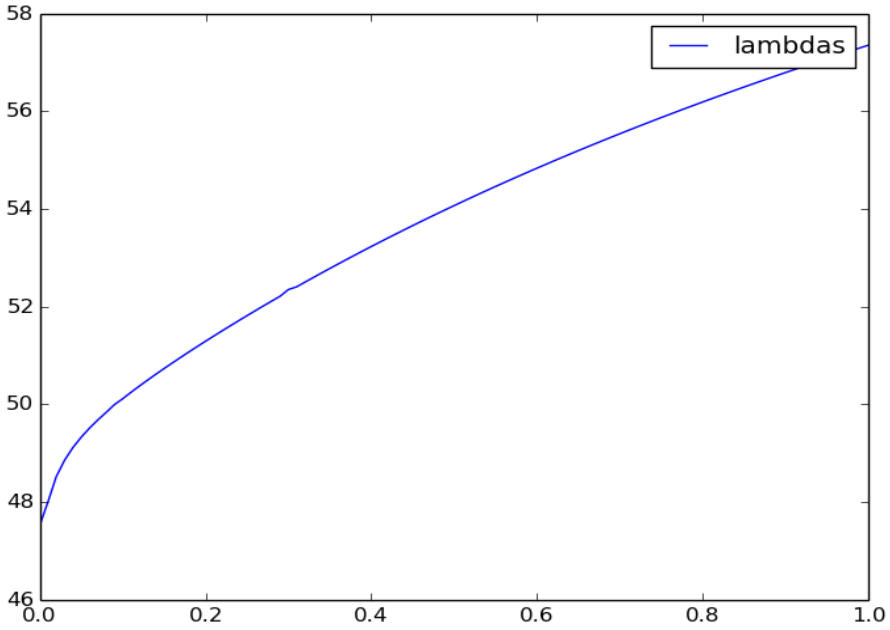

Fig 4.2 :Variation of RMSE for Training data with variation of lambda.

Comparison of RMSE for Ridge Regression using gradient descent & the one in Problem 3

For the training data, we observe that the variation in lambda increases the error in each of the two cases. It is minimum for lambda = 0 and as the value of lambda increases, the RMSE increases. Even for the test data, we observe that the value of RMSE vs lambda curse is almost same for both the cases, with the minimum RMSE value for lambda = 0.06.

Problem 5 Report 5: Non-Linear Regression

Error values for Test data for λ=0 and λ=0.06

In the problem 4, the optimal value for Lambda observed was 0.06. The following table shows the error values calculated for λ=0 and λ=0.06 for Test data.

Test Data		
P	**λ = 0**	**λ = 0.06**
0	79.2868513165	79.2898604296
1	62.0083440367	62.416796333
2	62.5070243981	62.4146141215
3	62.3536329193	62.4146033867
4	66.6582919959	62.4146030051
5	67.4894834581	62.4146030086
6	82.6647394523	62.4146030086

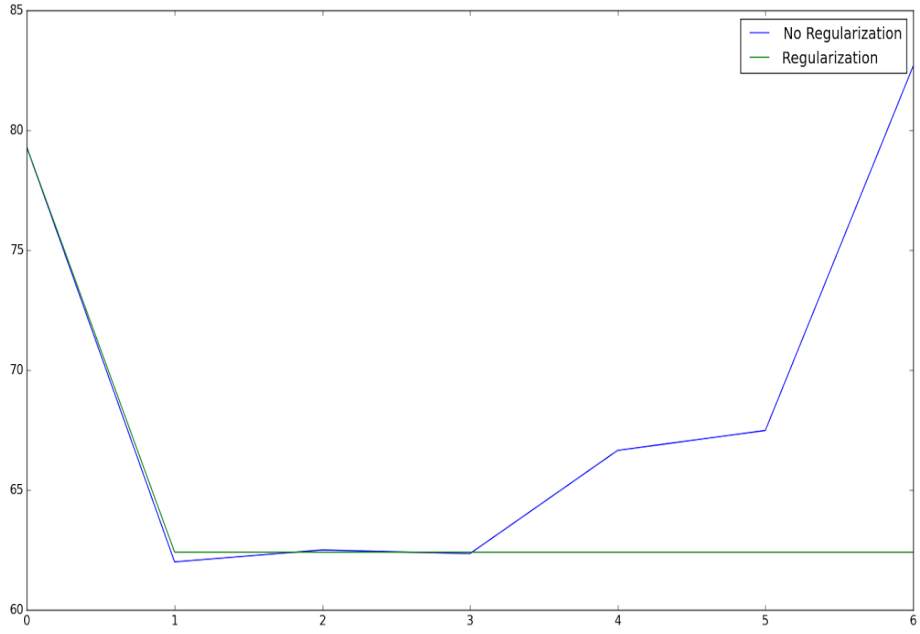

Fig 4.1 Plot of Regularization vs Non Regularization

The optimal values for P

For Lambda = 0:

The optimal value of P is 1. As we can see from the above table, P=1 has the lowest error when there is no regularization.

For Lambda = 0.06:

The error value becomes constant after p=1. As we can see from the table, the error values for p=2,3,4,5 and 6 is almost same. The optimal value for P can be any of 2, 3, 4, 5 and 6

Plot comparing RMSE for test data using λ=0 and λ=optimal

The following plot shows two lines which represent RMSE values with the regularization (λ=0.06) and RMSE values without regularization(λ=0**)** during the training.

Observation:
RMSE value with the regularization does not change after p = 1. It remains same for the remaining p values. The lowest error can be observed for multiple p values.

RMSE value with no-regularization component changes for different p. It can be noticed that RMSE remains constant for p=2 and 3. But, it increases for the remaining values of p.

Conclusion

RMSE value with regularization component does not change after p=1.

Problem 6 Report 6: Interpreting Results

Interpreting the results

Predicting diabetes level using the input features by using the observations made in the previous 4 problems.

Result from Problem 2:

RMSE for train data without intercept	138.20
RMSE for train data with intercept	46.76
RMSE for test data without intercept	326.76
RMSE for test data with intercept	60.89

Result from Problem 3:

RMSE for the Test data	53.39	λ=0.06
RMSE for the Train data	46.76	λ=0

Result from Problem 4:

RMSE for the Test data	53.39	λ=0.06
RMSE for the Train data	47.05	λ=0

.

Result from Problem 5:

RMSE for the Test data	62.41	λ=0.06 p = 1, 2,3,4,5,6
	62.00	λ=0, p = 1

To predict the diabetes level using the input features, need to combine the optimal settings from each of the four models.

Optimal setting from problem 2

Input data with the intercept (or bias) gives better results.

Optimal setting from problem 3

λ=0.06 gives the best results.

Optimal setting from problem 4

λ=0.06 gives the best results.

Optimal setting from problem 5

λ=0.06 and p = 1 gives best results. Any p value after p gives the same results as p=1.

Conclusion

From the above observations, it is recommended that anyone using regression should use input data with the intercept. The regression model can be ridge regression with or without using the Gradient descent with the parameter λ=0.06. For the Non-linear regression, the parameter λ should be 0.06 and p should be 1.