MACHINE LEARNING PROJECT REPORT

CSE574

Group42

Anuj Rastogi

Sagar Dhamija

Nalin Kumar

Problem 1:

We did the linear and quadratic discriminant analysis in this problem. The data was given from 5 classes. Based on the data we identified the Gaussian distribution that generated the data.

In case of LDA, the covariance matrix was same for all the classes and used a clustered mean for all the classes. In QDA, the covariance matrix was separate for all the classes and hence, the covariance for a particular class was calculated related to the mean for that class. The difference is evident from the accuracy learnt. In case of LDA the accuracy reported on the provided Test data was:

```
LDA Accuracy = (97.0,
```

On the other hand QDA accuracy on the same Test Data was:

```
QDA Accuracy = (95.0,
```

The plots for QDA and LDA are as follows:

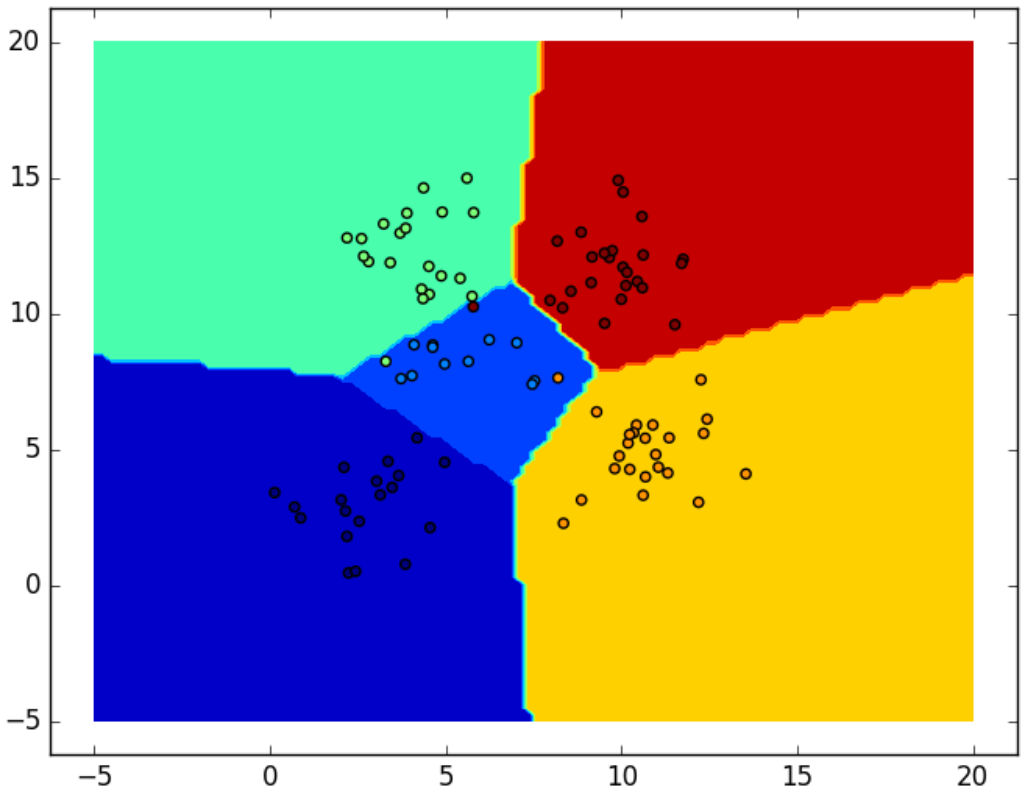

Figure 1. LDA Plot

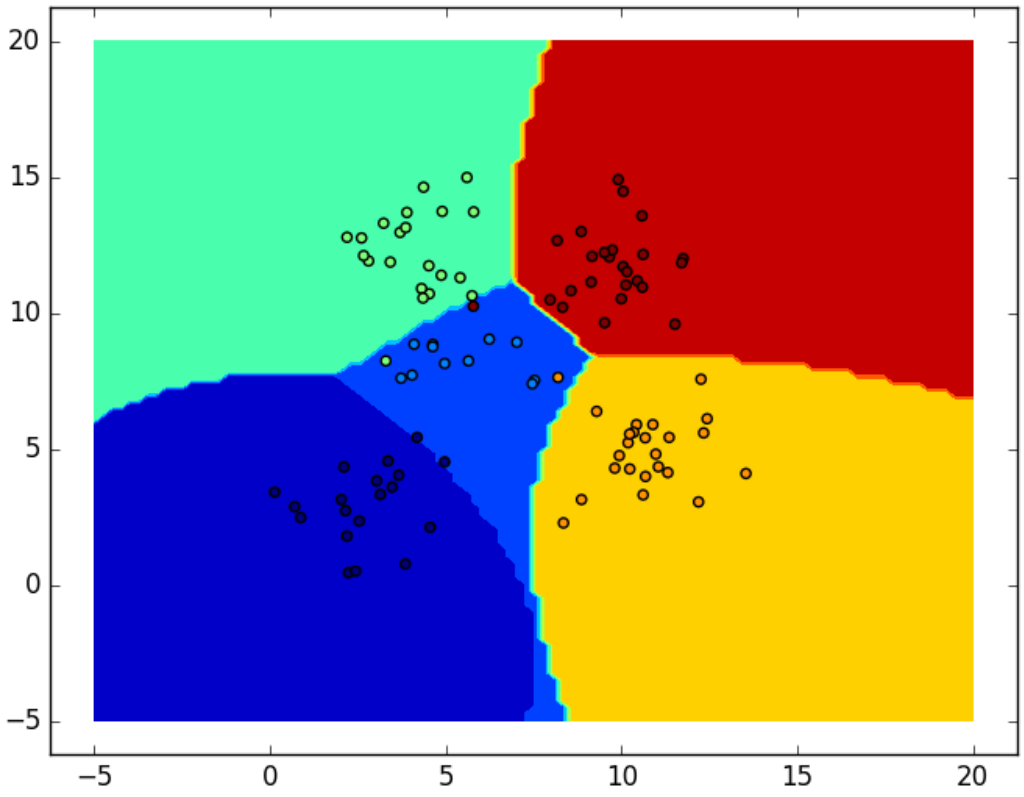

Figure 2. QDA Plot

EXPLANATION OF BOUNDARIES: The LDA boundaries as evident are linear in nature. In fact LDA only learns linear boundaries no matter what the data points are. Hence, it is less flexible compared to QDA, which can work fairly well on complex data set where linear separation is not possible.

The difference in the accuracy is due to the point that QDA treats 5 classes as separate Gaussian distributions with separate co-variance matrix. Due to this, the boundaries are complex and hence, this can cause over fitting on a test data set compared to the LDA where boundaries are just simple lines. But that does not mean that, LDA will always be better than QDA. In fact where the data is fairly complex and it is hard to separate it with a line, QDA will give higher accuracy. In our case as visible from the figure, the data is fairly separated and hence, a simpler model like LDA works well.

Problem 2:

In this problem we were supposed to work with simple linear regression. This is a prediction model. Here we computed the linear line for prediction. The line was computed with and without intercept.

Without Intercept:

In the first case we find the RMSE for training and test data without intercept. Then we find the RMSE for training and test data with intercept. The following are the results:

```
RMSE without intercept training[[ 138.20074835]]
RMSE without intercept [[ 326.76499439]]
RMSE with intercept training[[ 46.76708559]]
RMSE with intercept [[ 60.8920371]]
```

Explanation of the above results:

In case of RMSE both the models work well in case of training data. This is due to the fact that the weights have been learnt from this data. Now, we also see that with intercept the model performs better on test data. This is due to the fact that, without intercept the line can only rotate for points which are far off. A suitable explanation can be given with the help of the below figure:

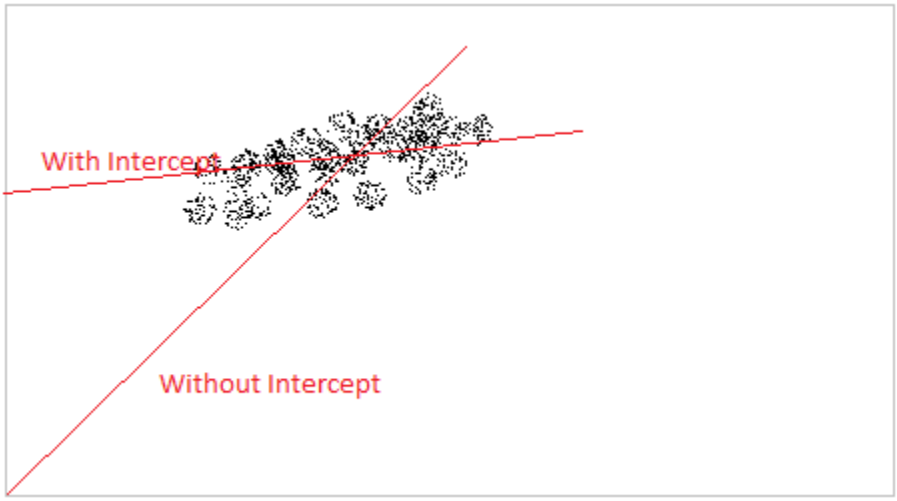

Figure 3. Example

For these data points a line with intercept will be a better fit as it can pass through the middle of their location. This is the reason why in the above case the RMSE error is reduced in case of intercept.

Problem 3:

The following is the training and the test RMSE error in case of ridge regression:

```
RMSE for ridge regression on Training data
46.7670855937
best lambda : 0.0

RMSE for ridge regression on test data
53.3978483971
best lambda : 0.06
```

The below is the effect of variation of lambda on training and test data:

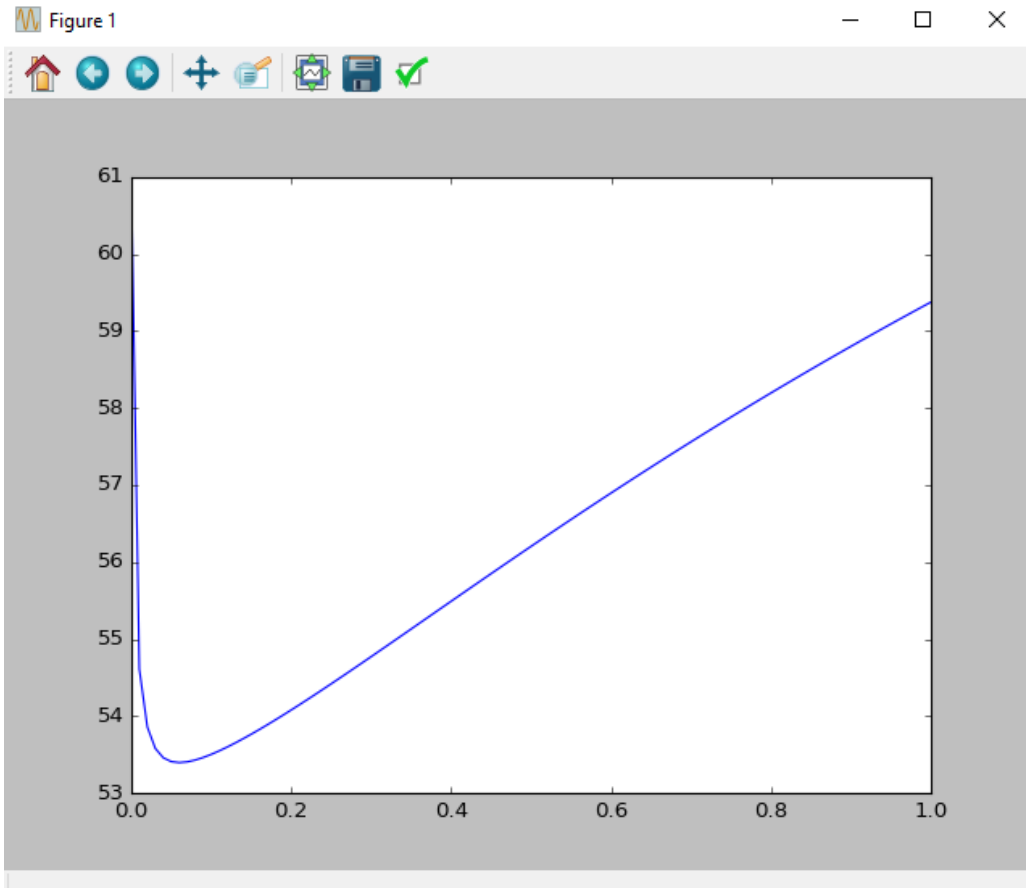

Figure 4. Training data

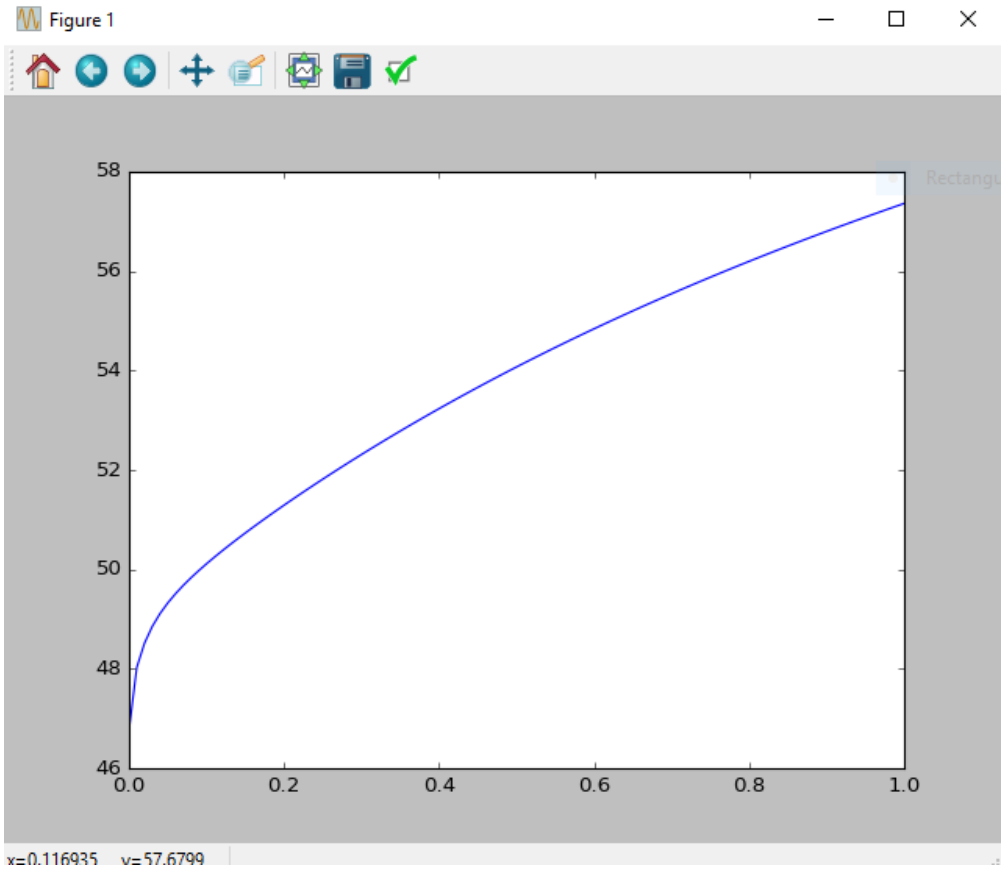

Figure 5. Test Data

Explanation of Figure:

With the variation in lambda, at lambda = 0.06 the error in test data is minimum. What happens when we model the parameters from the training data is that, the model becomes more and more complicated as per the training data. Due to this we add regularization so that the fluctuation in weights is reduced and the model becomes a bit simple. But as we increase lambda to a larger value, the model becomes simpler and hence, the RMSE starts increasing on test data as well.

Therefore, we have to choose the value of lambda so that, the model is neither too complex to avoid overfitting nor too simple to avoid under fit.

The main reason behind using ridge regression is the fact that, in case of correlated coefficients linear regression gets confused and hence give different weights on different runs for the same data. Therefore, once the ridge parameter is added the weights stabilize and do not give fluctuating results.

Problem 4:

In this part we used gradient descent to compute the weights from training data. Gradient descent is used when it is not possible to calculate the inverse of and [X (transpose) * X] matrix. It is also based upon the notion of minimizing the least squared error. On learning the weights through gradient descent and plotting the graph for error varying the value of lambda we get the following graphs for training and test data.

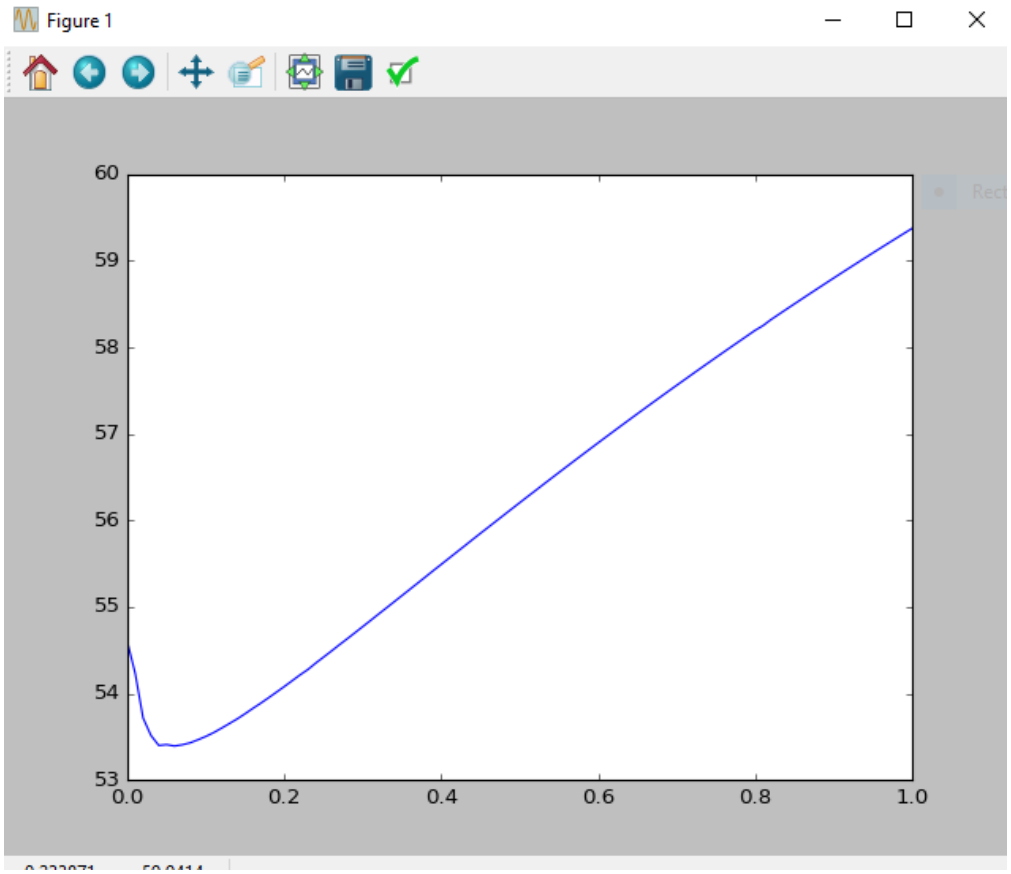

Figure 6. Test Data

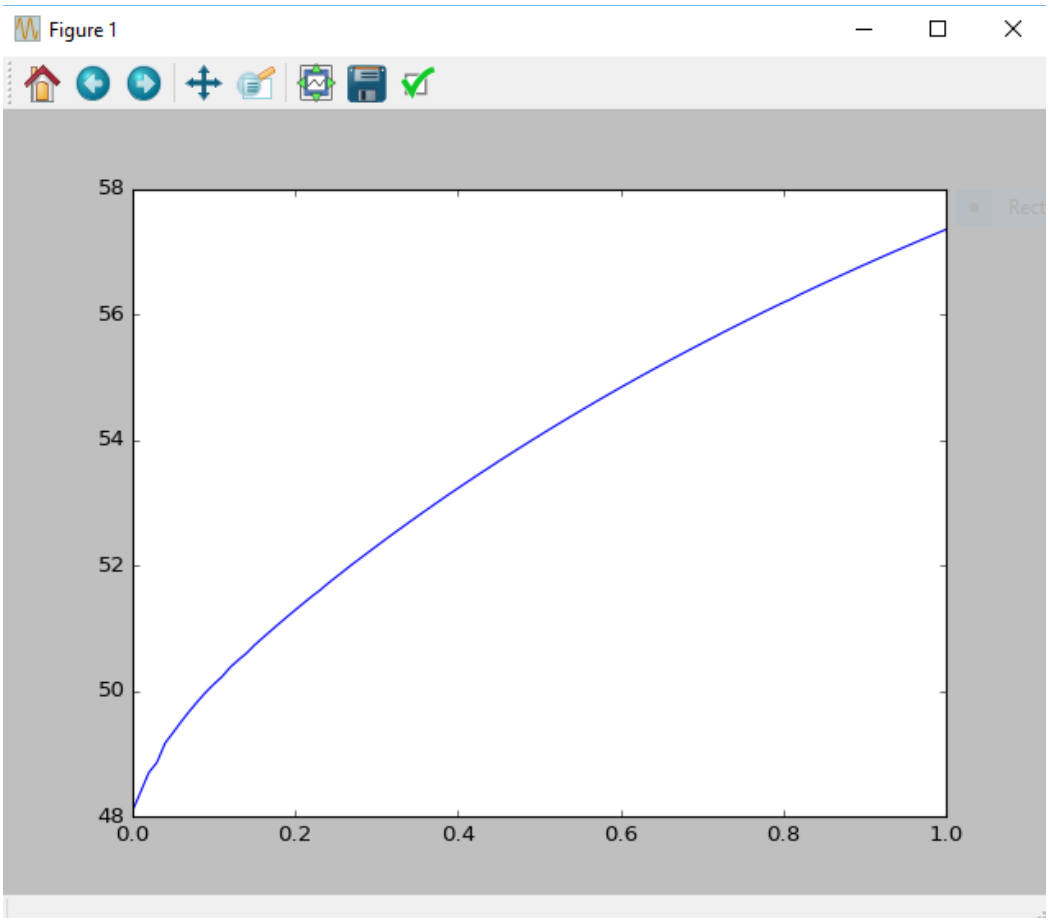

Figure 7. Training Data

Comparison with Problem 3:

If we compare the results obtained from Gradient descent with the results obtained from Ridge Regression we find that the same graphs are obtained which is obvious. This is due to the reason that, when we do Gradient descent using least square error method and do regularization the curve is convex and hence, has only one peak. Therefore, the Gradient descent reaches the appropriate answer.

Also, the ridge parameter in question 3 is equivalent to the regularization of Gradient descent. This also proves that the formulae obtained by maximizing the likelihood, in case of ridge regression, is the same as that obtained from minimizing the error, in case of Gradient descent.

Problem 5:

The following graphs show error on training and test data corresponding to different values of "p" and values of lambda as "0" and "0.06". 0.06 is the optimum value of lambda obtained from problem 3. The graphs are shown for training and test data.

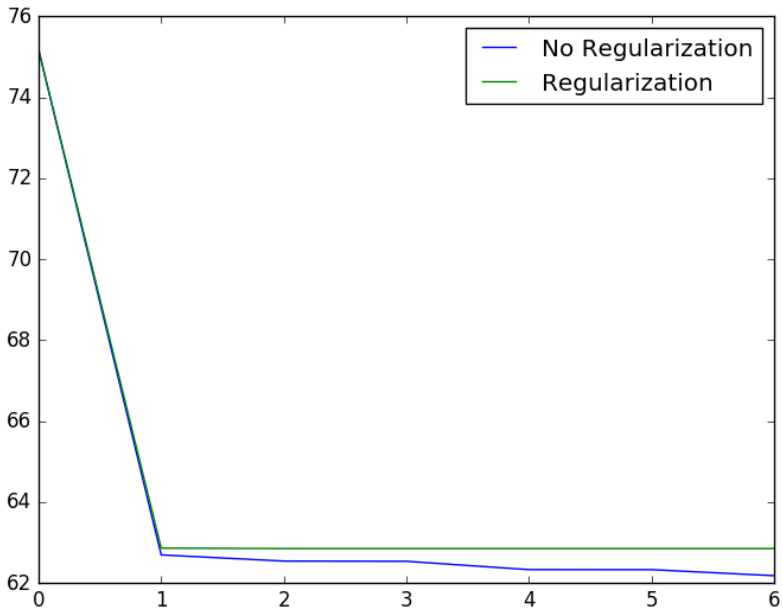

Figure 8. Training Data

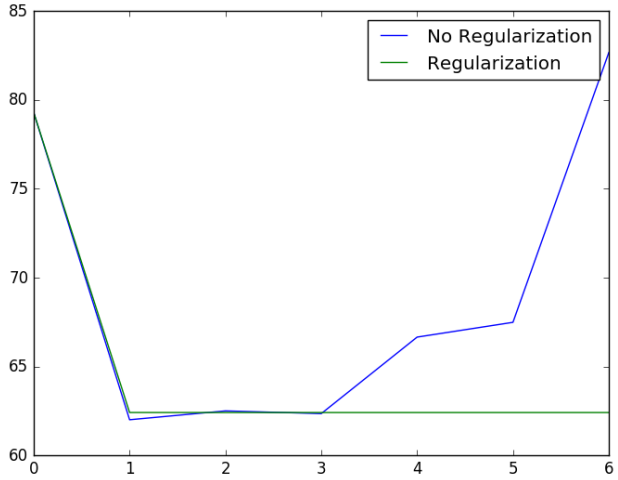

Figure 9. Test Data

Explanation of Figure:

We see that in case of training data, because the parameter or “Ws” are obtained from it so the more complex the model is the better result it gives. Hence, with no regularization and increased value of “p” the model gains complexity and therefore gives the best result in case of training data. This is evident from Figure 8.

In case of Test data though a more complex model will be over fitted and hence, will increase the RMSE. Therefore, the best result will be obtained if the model is simple and not over fitted and at the same time not under fitted as well. Hence, if we see Figure 9 where the test data is tested, the regularized model with less value of “p” gives a better result.

Also, it is worth noting that, due to ridge regression, which prevents the weights from fluctuation in case of co related coefficients, the weights do not fluctuate even at high value of “p”. The model thus, gives more or less the same RMSE and hence “Ws” in case of all “p”.

Also from the below data we can get the optimum value of “p” which gives a good result on test data for value of lambda as 0.06.

```
[[ 79.28685132  79.28986043]
 [ 62.00834404  62.41679633]
 [ 62.5070244   62.41461412]
 [ 62.35363292  62.41460339]
 [ 66.658292    62.41460301]
 [ 67.48948346  62.41460301]
 [ 82.66473945  62.41460301]]
```

From the above it is clear that the best value of p is **p = 1**. This is also visible from graph.

Problem 6:

For anyone working with diabetic data, one should use a linear model. This is proved by the nonlinear regression which shows that the best outcome comes with the value of p as 1.

Also, one should use ridge regression, which minimizes the error by preventing overfitting and the value of lambda as 0.06.

One may also opt for linear regression using gradient descent to avoid issues with invertible matrices.