CSE 574

Programming Assignment 2

A PROJECT REPORT ON

Classification and Regression

Group Number: 29

Gitanjali Palwe (50134107)

Sarthak Bhat (50134371)

Mohit Kothari (50134655)

Ankit Goyal (50133155)

Abstract:

The main objective of the project is to evaluate the performance of various classification and regression techniques.

Problem 1: Experiment with Gaussian discriminators

Implementation:

Linear discriminant analysis (**LDA**):

Both LDA and QDA are multivariate Gaussian but having a non-diagonal covariance. In case of LDA we consider same covariance matrix for each target class. Consider a set of observations $\vec{x}$ (also called features, attributes, variables or measurements) for each sample of an object or event with known class y. This set of samples is called the training set. The classification problem is then to find a good predictor for the class y of any sample of the same distribution (not necessarily from the training set) given only an observation $\vec{x}$.

LDA approaches the problem by assuming that the conditional probability density functions $p(\vec{x}|y=0)$ and $p(\vec{x}|y=1)$ are both normally distributed with mean and covariance parameters $(\vec{\mu}_0, \Sigma_0)$ and $(\vec{\mu}_1, \Sigma_1)$, respectively.

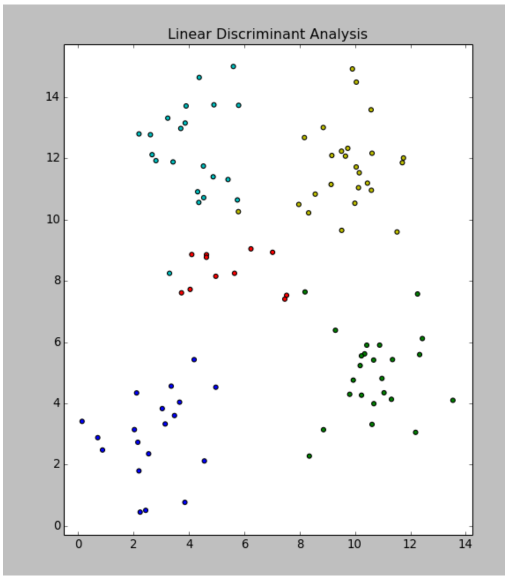
Quadratic discriminant analysis(QDA):

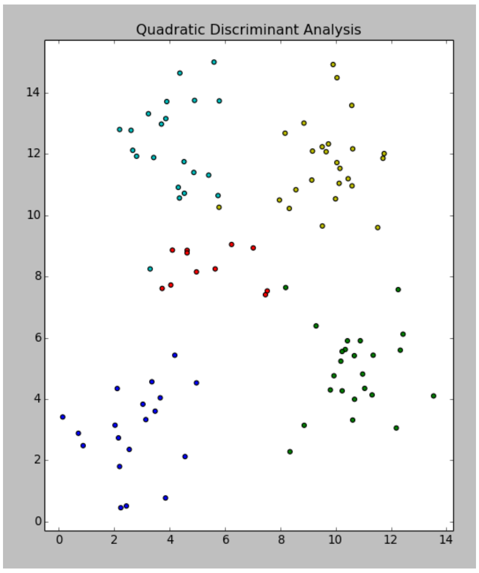
Quadratic discriminant analysis (QDA) is closely related to linear discriminant analysis (LDA), where it is assumed that the measurements from each class are normally distributed. Unlike LDA however, in QDA there is no assumption that the covariance of each of the classes is identical. When the normality assumption is true, the best possible test for the hypothesis that a given measurement is from a given class is the likelihood ratio test. Suppose there are only two groups, (so $y \in \{0,1\}$), and the means of each class are defined to be $\mu_{y=0}, \mu_{y=1}$ and the covariance's are defined as $\Sigma_{y=0}, \Sigma_{y=1}$.

$$\begin{aligned} p(y|\mathbf{x}) &= p(y)\prod_j p(x_j|y) = p(y)\prod_j \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_j-\mu_i)^2}{2\sigma_j^2}} \\ &= p(y)\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}{2}} \end{aligned}$$

Observations:

During this experiment we trained our system using a dataset with 150 samples and there true classes, there were 5 classes involved in this example. For LDA, using the training data we calculated mean of each attribute for each class and covariant matrix and passed the two matrix to testing functions where using these two matrices we predict the label of test data and then compare it with its true label.





As we can observe LDA is giving us 97% accuracy and QDA 95%. So this classifier is really good as we have trained the system with very less samples then also it is giving us this much accuracy, but the problem with this classifier is that as the no of attributes will increase so does the complexity and it will effect our accuracy also.

Problem 2: Experiment with Linear Regression

Implementation:

We have implemented the least squares method to estimate regression parameters by minimizing the squared loss. For learning the weights, we have used the following formula

$$\widehat{\mathbf{w}}_{MLE} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$$

Observations:

Following rmse values were observed by applying linear regression on training and testing data. We do comparison between rmse values which are once calculated with a bias term (intercept) and once without it.

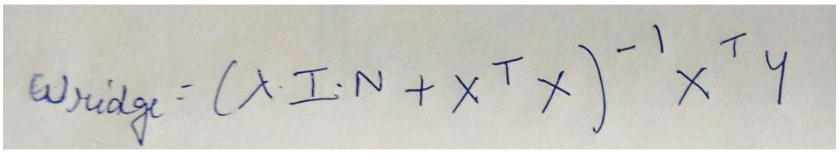
a. RMSE value without intercept: 23.10577434
b. RMSE value with intercept: 4.30571724

Thus, we see by introducing the bias term error value significantly decreases .This shows using a bias term in linear regression gives us more accurate classification with less error.

Problem 3: Experiment with Ridge Regression

Implementation:

For learning the weights using ridge regression, we have used the following formula



$$W_{ridge} = (\lambda \cdot I \cdot N + X^T X)^{-1} X^T Y$$

Observations:

Following graph was plotted for different values of rmse obtained by varying values of lambda from 0 to 0.004.i.e first with no regularization and then adding regularization term.

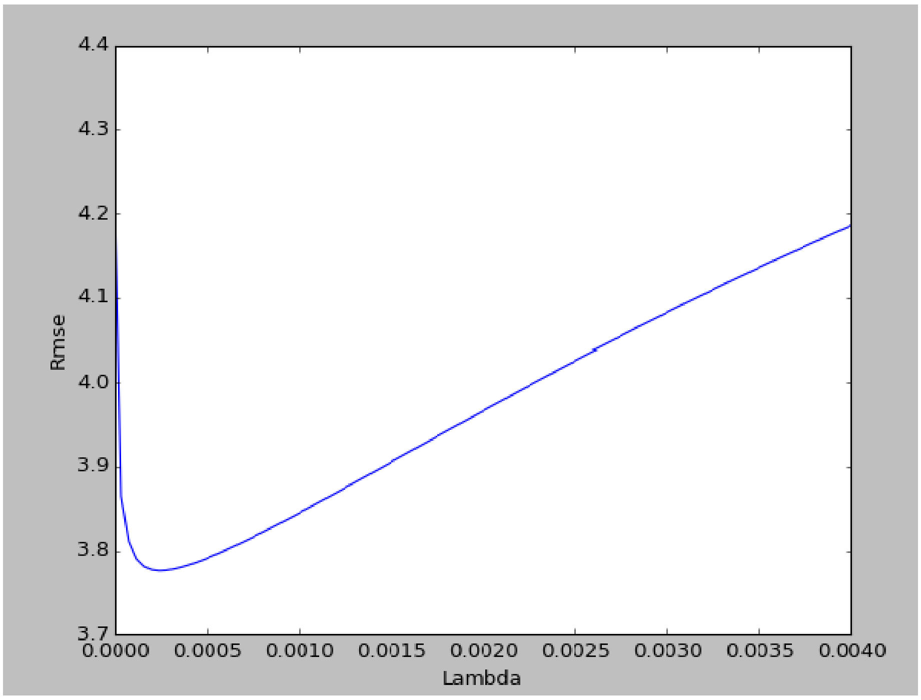


Fig: RMSE vs Lambda

As we see in the graph when there is no regularization rmse value is high around 4.2 but when we add regularization term in the equation rmse value goes on decreasing and is minimum at 0.0002 then it goes on increasing again. Thus optimal value of lambda can be said to be 0.0002 which gives least error.

Comparison of weights using OLE and ridge regression:

Compared to the linear regression weights, the ridge regression weights are slightly shifted toward zeros, which stabilises them. This helps in better estimation of regression parameters

OLE vs Ridge regression:

Ridge regression is basically minimizing a penalised version of the least-squared function. The penalising shrinks the value of the regression coefficients. 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. Linear regression basically follows a straight line in which the noise in observations will cause great variance thus not giving accurate classification. Thus ridge regression is a better method compared to linear regression

Problem 4: Using Gradient Descent for Ridge regression learning

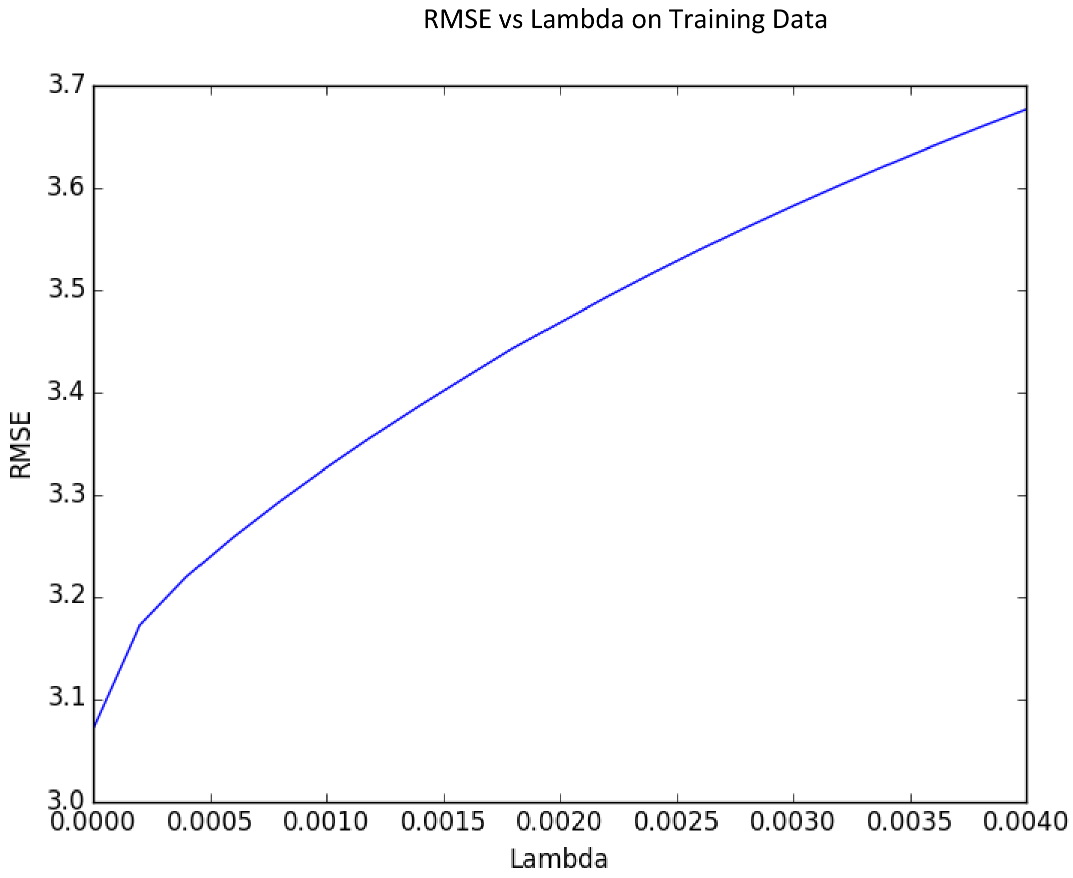
Implementation:

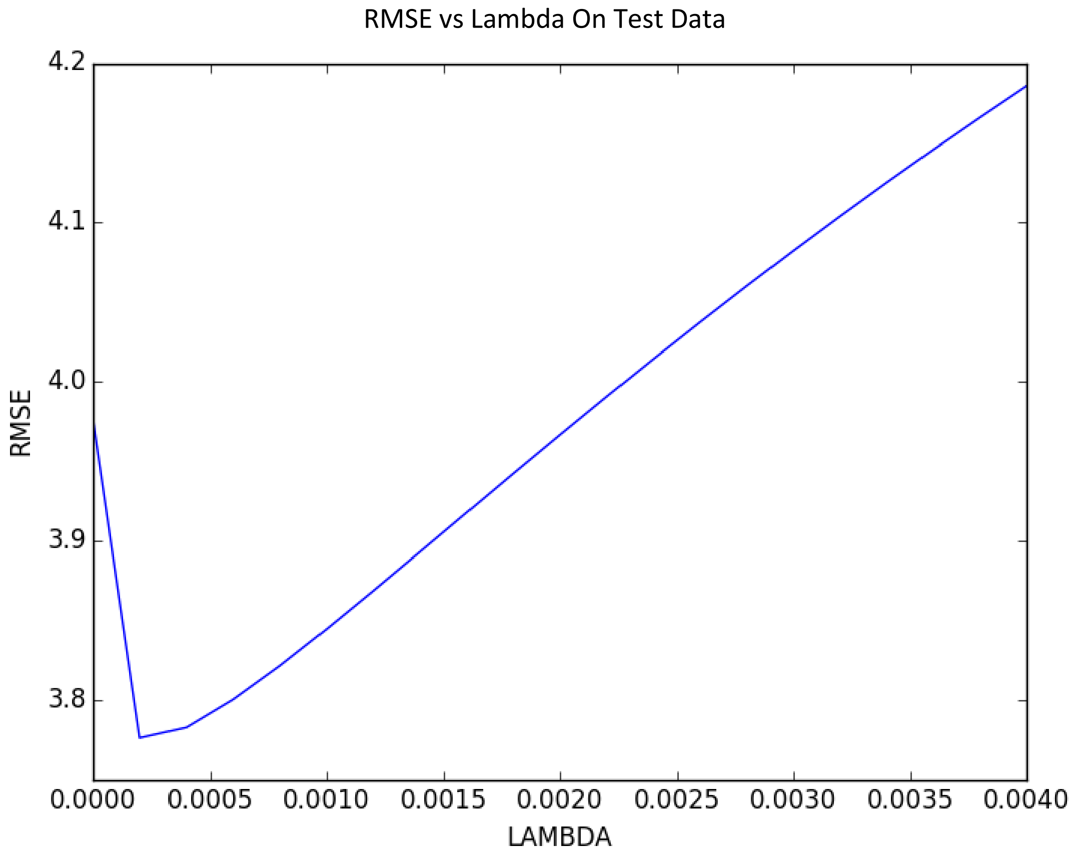
Here in this problem gradient decent is used to minimize the loss function. So to estimate the error via gradient decent use the following equation for error and error gradient function.

Error = $1_{/2N}$ $[(Y – X w)^T(Y – X w)] + 1/2\lambda w^Tw$

Error Gradient = $1_{/N}$ $[-Y^TX + w^T (X^T X)] + \lambda w$

Observations:





Optimal value for lambda: 0.0002

<u>Comparison with problem 3:</u>
Result of Gradient Descent and Ridge Regression learning for optimal lambda and minimum RMSE is the same i.e. lambda = 0.0002 and RMSE = 3.755

But Gradient descent approach uses minimizer that in turn calls the regressionObjVal() [100 times] to compute regularized square error, which is slow as compared to Ridge regression approach .

Ridge Regression runtime: 0.0351555206239
Gradient Descent runtime: 0.940770195496
So Ridge Regression is approximately 30 times faster than Gradient Descent method.

Problem 5: Non Linear Regression

Observations:

Case 1:

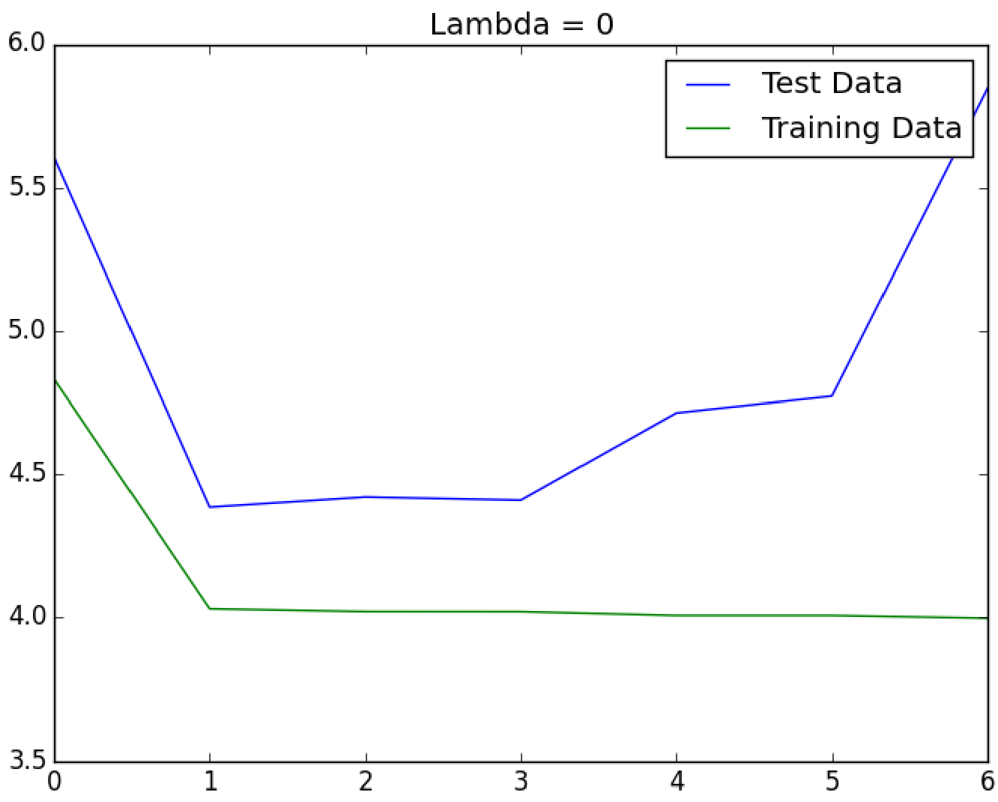
Lambda = 0:

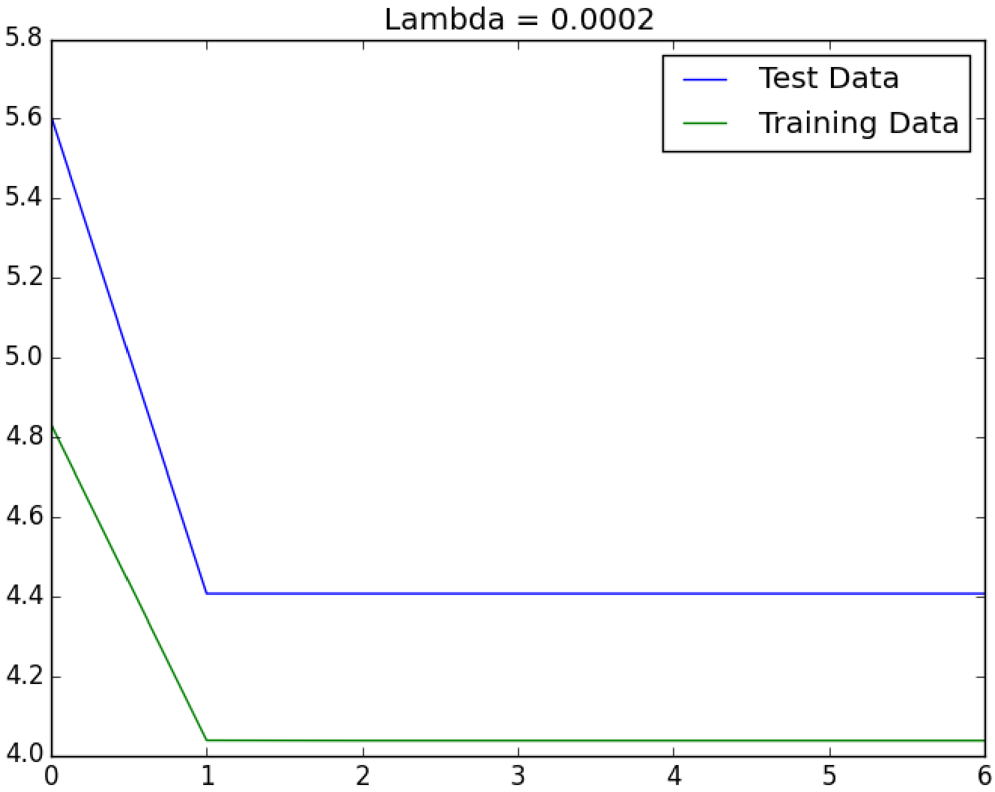
p	TEST DATA	Train Data
0	5.60642702248	4.8321882775
1	4.3846520558	4.03031662483
2	4.41991408237	4.0205256796
3	4.40906766689	4.02019111883
4	4.7134530292	4.00695318605
5	4.7722271412	4.00691920019
6	5.84527978318	3.99735627998

Case 2:

Lambda = 0.0002:

p	TEST DATA	TRAIN DATA
0	5.60659857586	4.83218865818
1	4.40623928425	4.03759717594
2	4.40616765889	4.03690643916
3	4.40616843176	4.03690407992
4	4.40616843275	4.03690396502
5	4.40616843354	4.03690396412
6	4.40616843356	4.0369039641





Minimum error:

	Lambda=0	Lambda=0.0002
Train Data	3.997	4.036
Test Data	4.384	4.406

So from the above figures we can conclude that the optimal value for d=1 for both the values of lambda. The complexity involved increases as the value of p increases. Selecting d=1 also implies that this is a linear ridge regression.

Problem 6: Interpreting results

<u>Comparison of training and testing error using different approach:</u>

Metric for the best setting:

The best metric for comparison of the previous 4 approaches for predicting diabetes level is – test error.

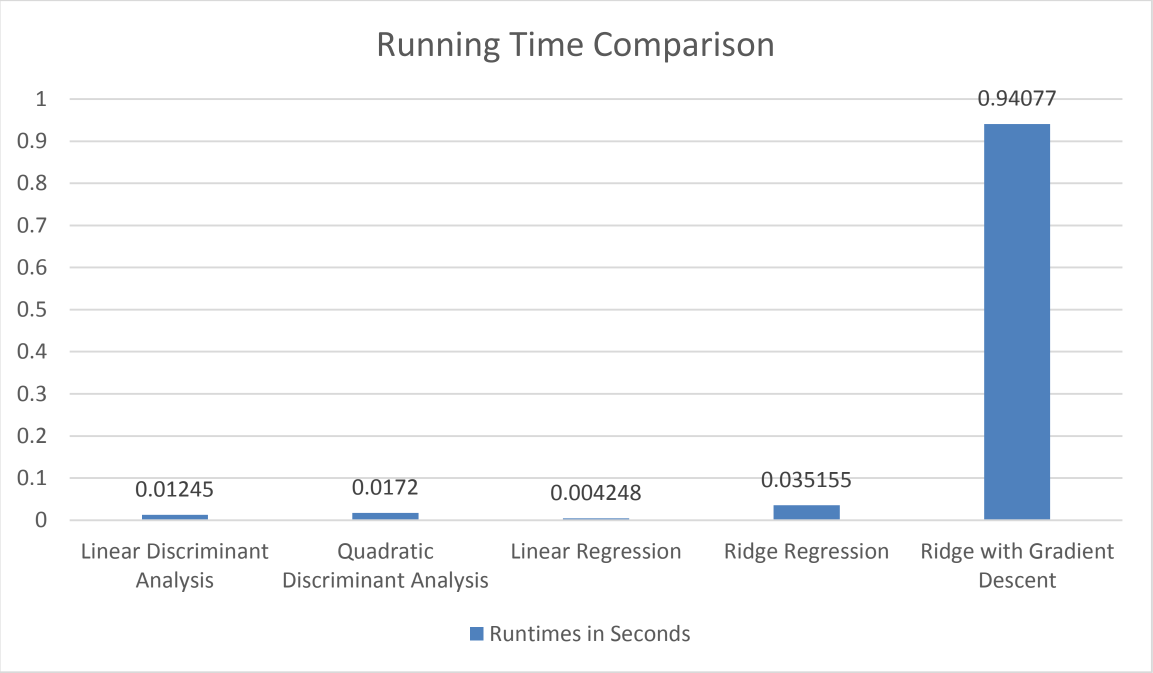
- Linear Discriminant Analysis and Quadratic Discriminant Analysis has the accuracy 97% and 95% respectively.
- Minimum RMSE for test data for Linear Regression (without intercept): 23.10577434

- Minimum RMSE for test data for Linear Regression (with intercept): 4.30571724
- Minimum RMSE for Test data for ridge Regression for Optimal λ= 0.0002 is: 3.755
- Minimum RMSE for Test data for ridge Regression (by Gradient Descent method) for Optimal λ= 0.0002: 3.755
- Minimum RMSE for Test data for non-linear Regression without regularization: 4.384
- Minimum RMSE for Test data for non-linear Regression with regularization (Optimal λ= 0.0002): 4.406

From the above error values, we can infer that Ridge regression and Gradient descent method gives the minimum test error compared to other approaches. But the running time in case of Gradient descent method is high and hence ridge regression method is the best one for test data.

Running Time Comparison:

- Runtime for Linear Discriminant Analysis: 0.0124530189274 s
- Runtime for Quadratic Discriminant Analysis: 0.0172019512072 s
- Running time for Linear regression: 0.00424837981668 s
- Running time for Ridge regression: 0.0351555206239 s
- Running time for Ridge regression with Gradient Descent method: 0.940770195496 s



From the above values, we can infer that the linear regression method takes the minimum running time for implementation.

References:

- http://en.wikipedia.org/wiki/Quadratic_classifier
- http://en.wikipedia.org/wiki/Linear_discriminant_analysis