CSL 603 - Machine Learning : Assignment 2

By - Komal Chugh
2016csb1124

Part I

Objective : The objective of this assignment was to implement linear ridge regression and to predict the age of Abalone. Following are the results of different experiments.

Representation : Various attributes such as gender were represented in matrix form so that linear regression could be performed. Also 1 was appended to every example for the W_0 coefficient.

Standardisation : All the independent variables except the first 4 columns were standardised because the first column is for the intercept term, and the other 3 columns encode the male, female and infant information. The features were scaled because having all the features on the same scale improves gradient descent.

Linear Regression : The function mylinridgereg(X, Y, lambda) calculates the linear least squares solution with lambda as the penalty parameter and returns the best weights. The weights were learned using gradient descent whose update equation is as follows :

$$W^{new} = W^{old} - \alpha\,(X^T\,(FX - Y) + 2\,\lambda\,W^{old})$$

The stopping criteria is the number of iterations which in this case is fixed to 1000. Also the function mylinridgeregeval(X, W) was implemented which calculates the predicted target value as XW .

Experiment 1 : Different training set fractions and lambda values were taken and for each fraction and lambda pair, 100 iterations of linear regression were done and the average error over the 100 iterations was calculated.

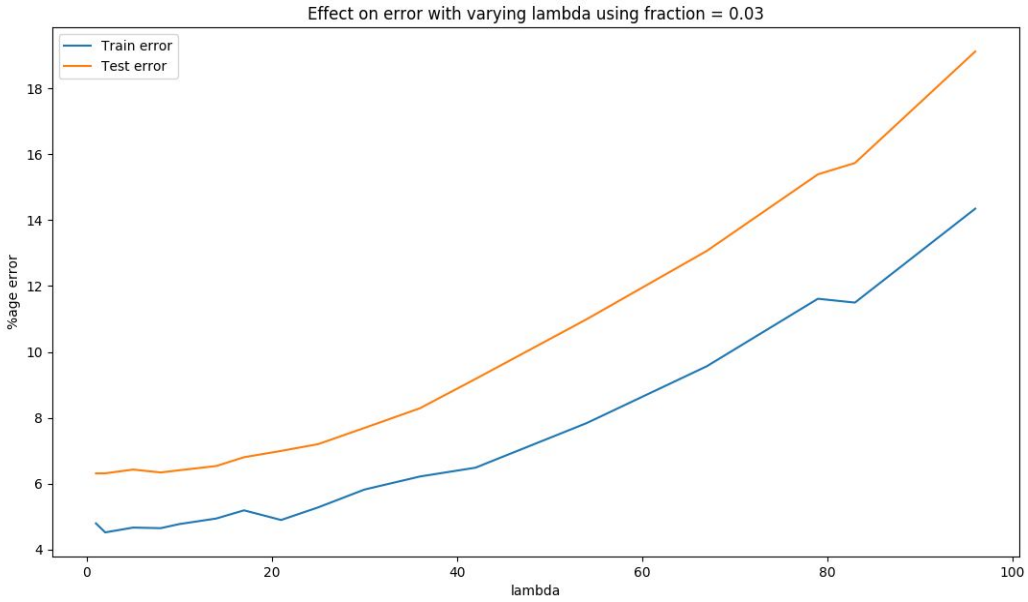
Ques 1 : Does the effect of λ on error change for size of the training set?
Ans : Yes, the effect of lambda on error changes for different partitions of the data

into training and test set. As the training set fraction increases, effect of lambda decreases on error. It can be seen from the below given plots.

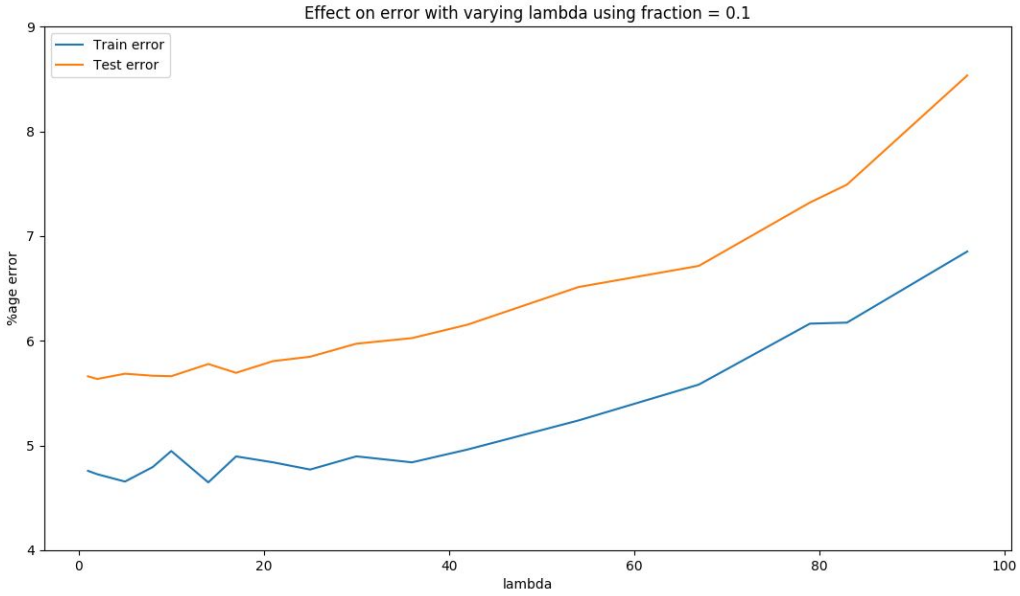
Ques 2 : How do we know if we have learned a good model?
Ans : We can check the goodness of a model by checking the accuracy on a test dataset. We feed examples to the model which it has not seen till now. If the model is able to classify them well, then we know that we have learned a good model. So, a low mean square error on test dataset is an indication of a good model.

Experiment 2 : Plotting graphs for the above experiment.

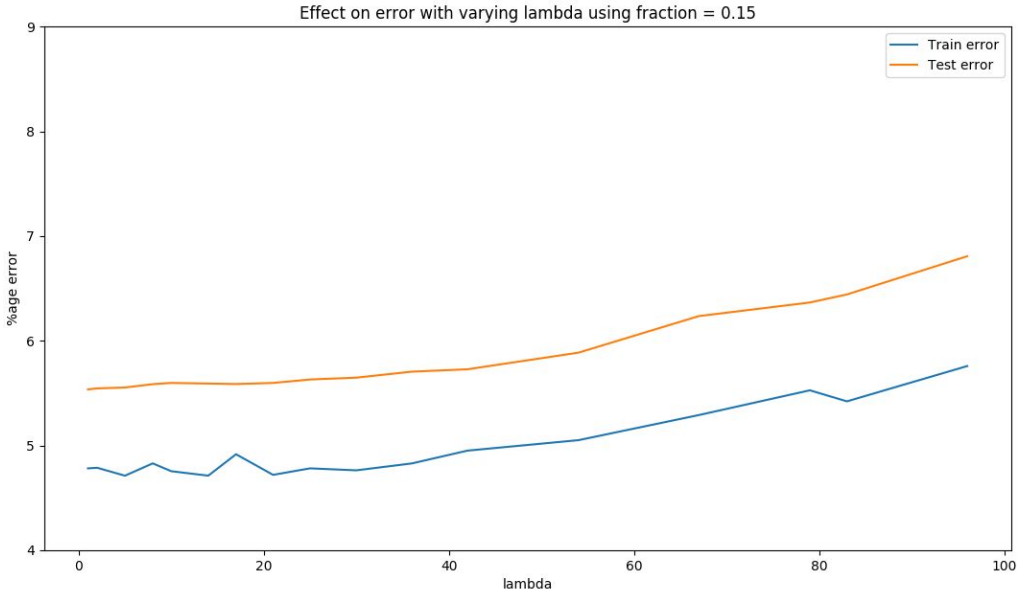
a) Fraction = 0.03



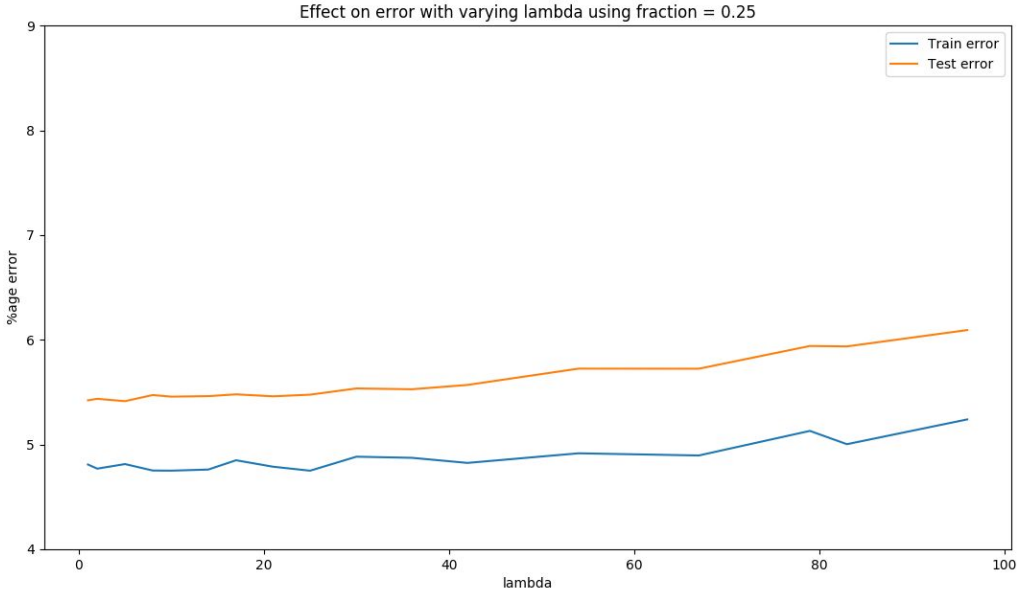
b) Fraction = 0.1



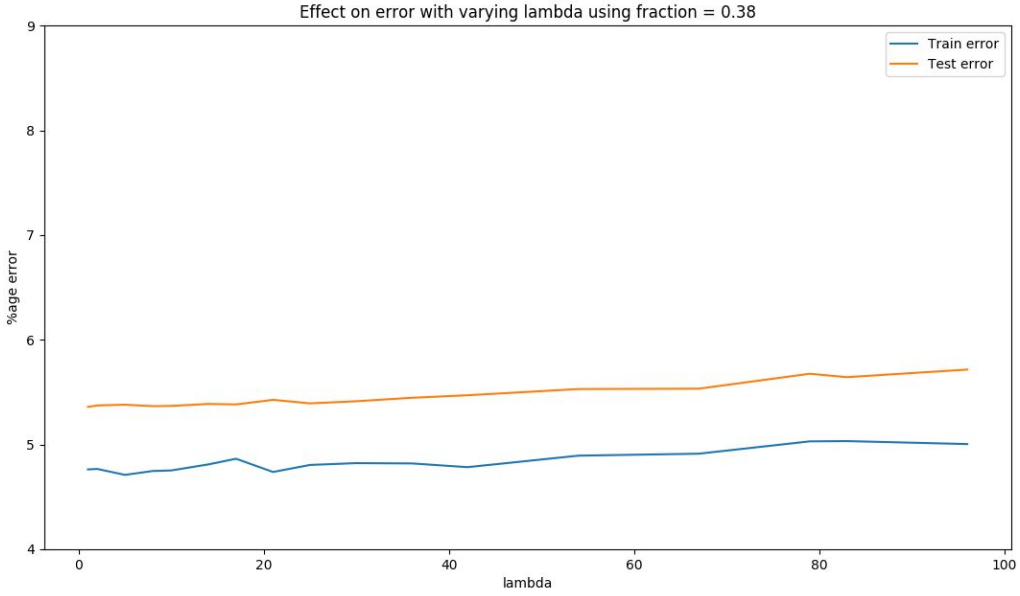
c) Fraction = 0.15



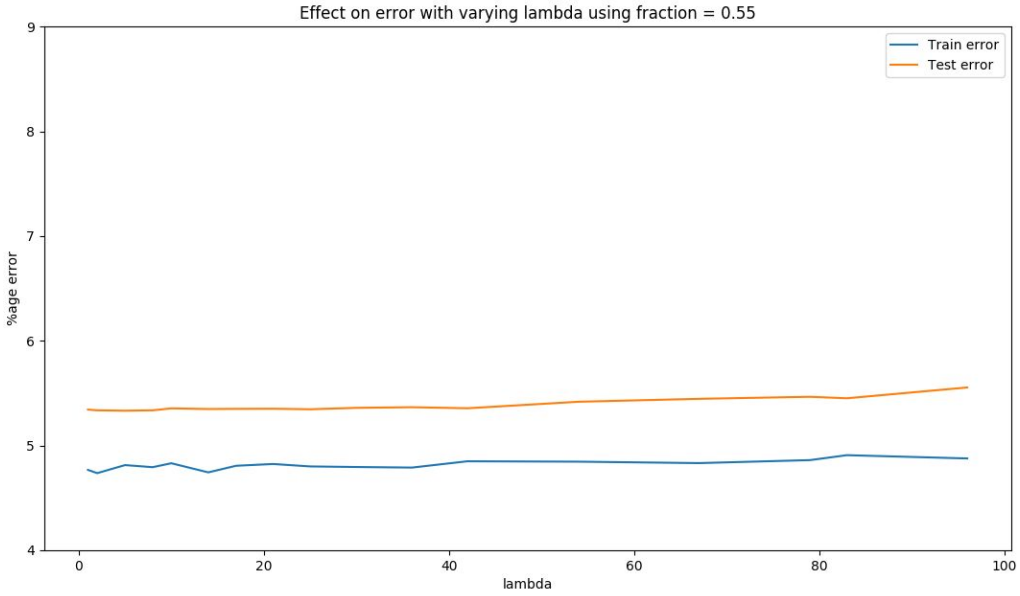
d) Fraction = 0.25



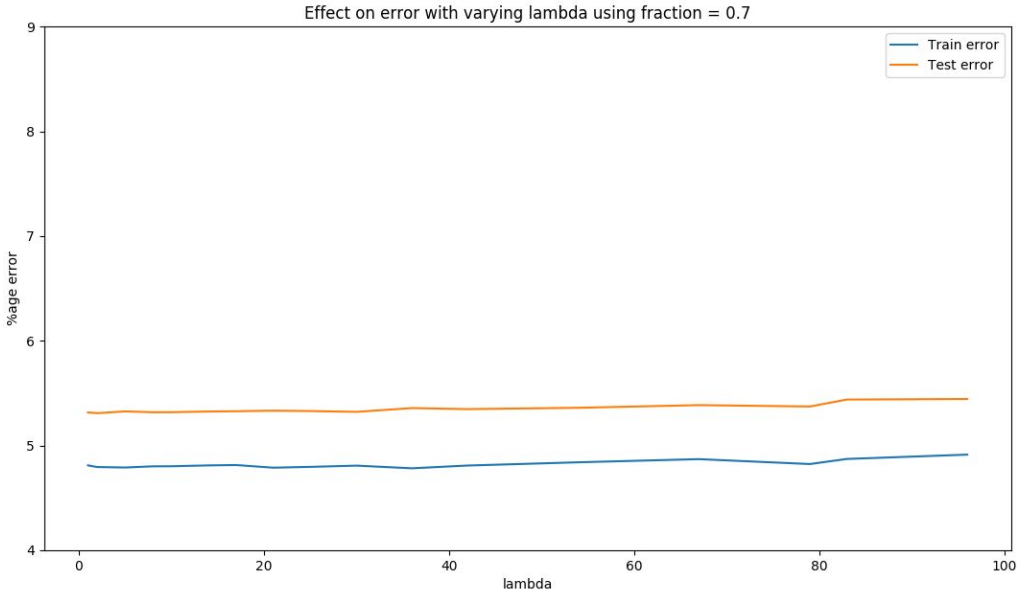
e) Fraction = 0.38



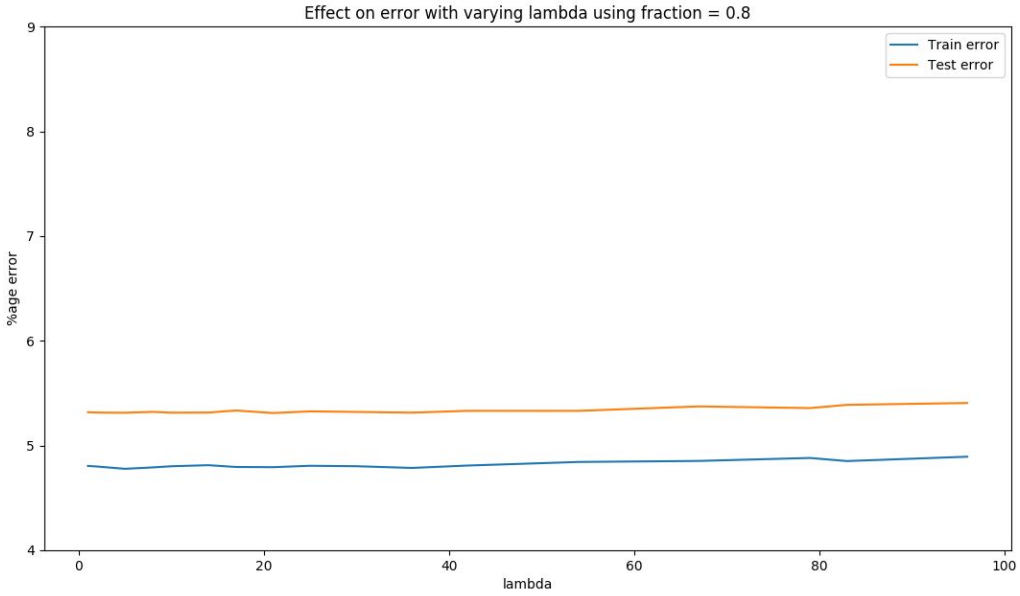
f) Fraction = 0.55



g) Fraction = 0.7



h) Fraction = 0.8

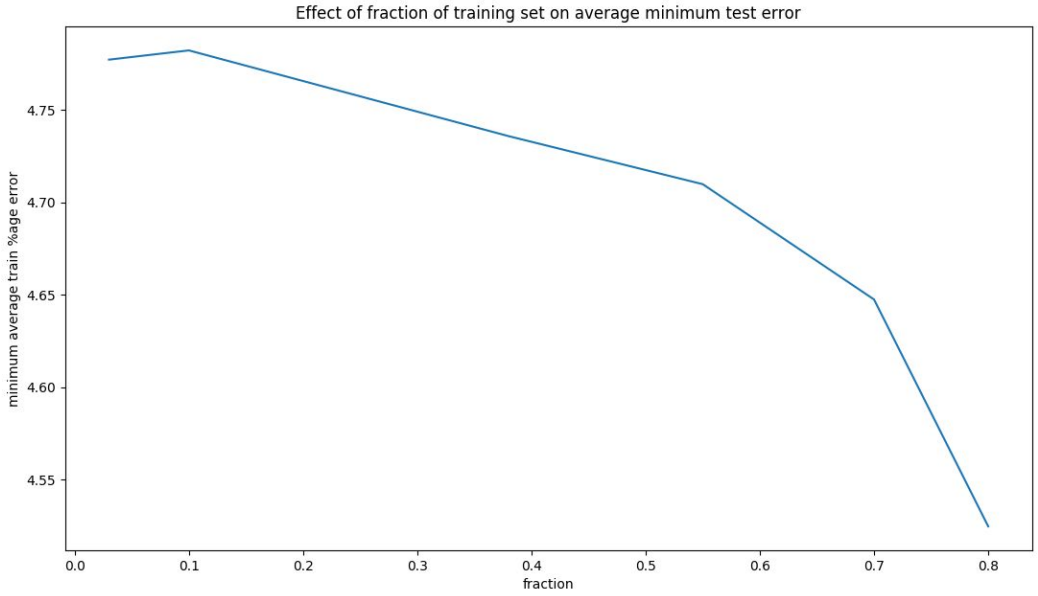


As evident from the above graphs, on increasing the training set fraction, effect of lambda decreases on the error. When we have less training data, then the change in lambda changes the training error significantly. The change is significant because we have less training points, whereas when we have large number of training examples, the change in lambda can't influence the training error because the error is being averaged
over large number of points. The same explanation applies to the test error also.

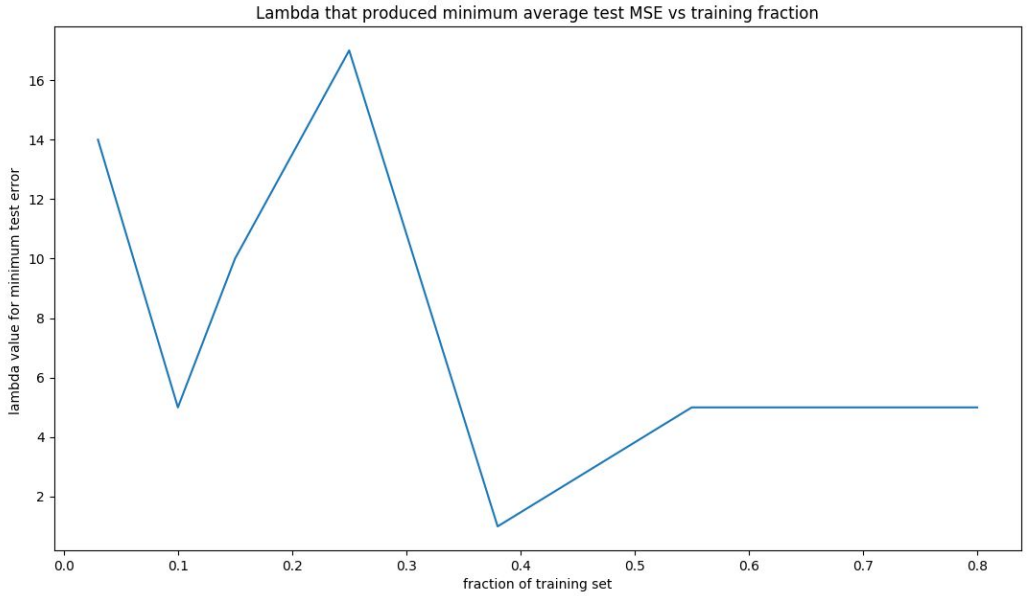
Experiment 3 :

a) Minimum average Mean Squared Error vs training fraction

As the training fraction increases, the minimum mean square error decreases. This can be explained as : Increasing fraction means increases the number of training examples and hence we have more and more data to learn from due to which the error decreases. It can be seen from the below given graph as well.



b) Lambda that produced minimum average test MSE vs training fraction



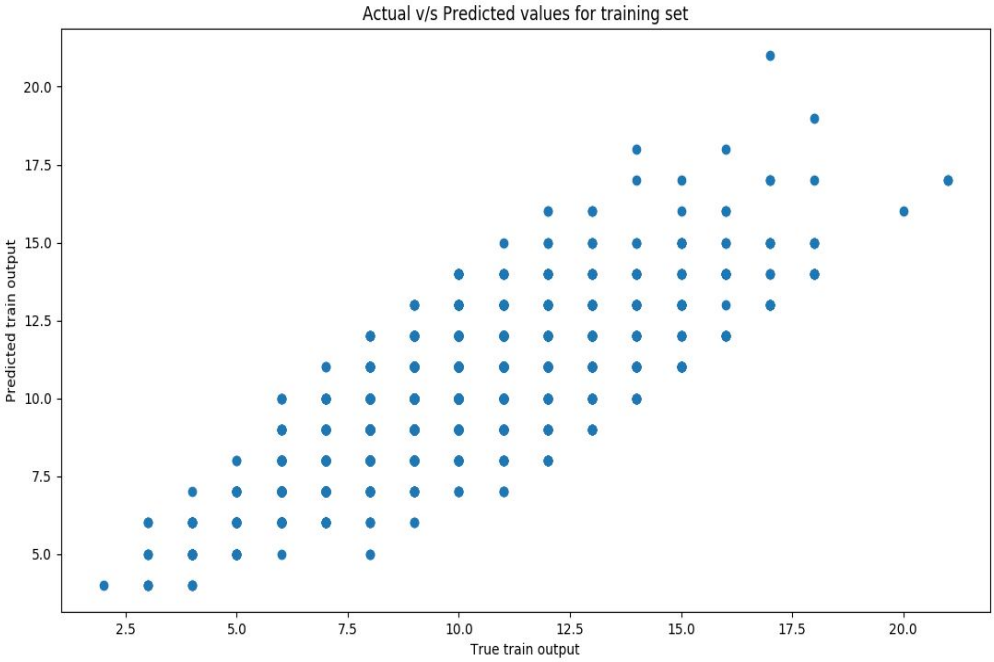
We can see from the graph that there is no direct trend as to which lambda

gives the best fit ie minimum error and hence the determination of lambda is empirical. Though for increasing fraction set, low values of lambda like 4 seem to fit well.

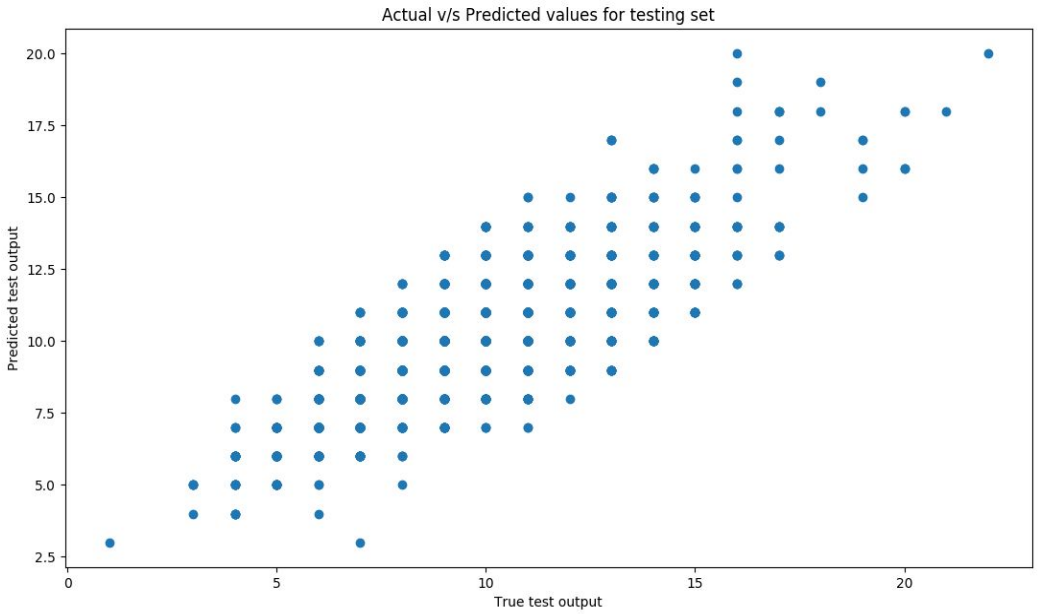
Experiment 4 : Actual v/s Predicted output plot

To see the contribution of each example in the error term, we plot its predicted target value against the actual target value. It can be observed that most of the points are close to the 45 degree line. It suggests that the learned model is good.

a) For training set



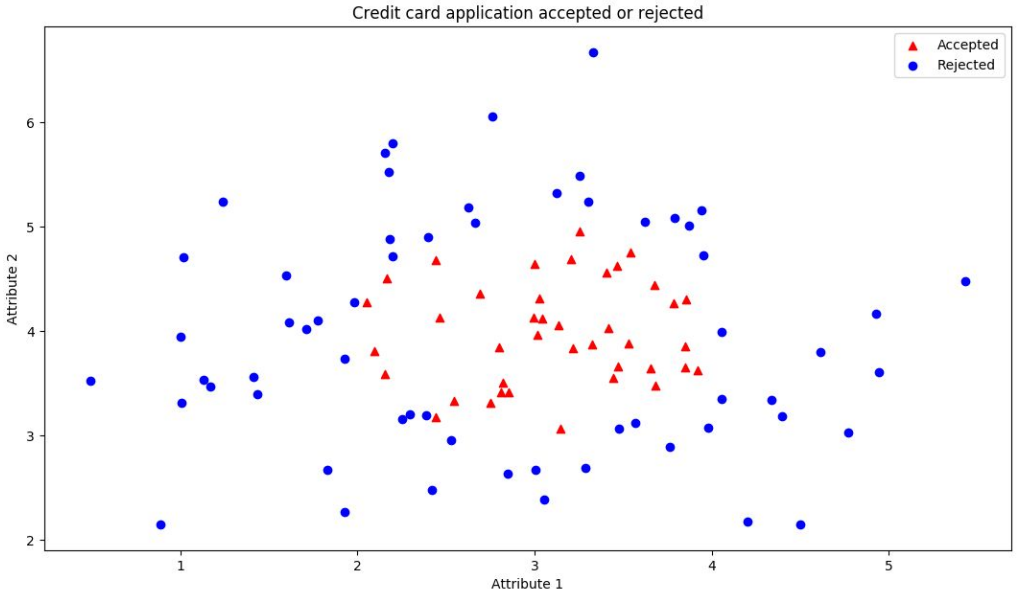
b) For test set



Part II

Objective : The objective of this assignment was to implement regularized Logistic Regression to predict whether a credit card can be issued to an individual or not on the basis of 2 given attributes. Here, Regularized Logistic Regression is implemented using Gradient Descent and Newton Raphson method. To get a better model, feature transformation was also implemented so that higher order polynomials could be learned.

Plotting data : The plot of training dataset is given below. Triangle indicates that the credit card was issued whereas circle indicates that the credit card was not issued.



Logistic Regression : Given the training matrix X and weights W, the output of logistic regression is given by :

$$FX = \frac{1}{1+e^{-XW}}$$

And the error function is defined as :

$$E = -\sum_{i=1}^{N} y^i \, log(fx^i) + (1-y^i) \, log(1-fx^i)$$

For the **gradient descent**, the weight update equation becomes :

$$W^{new} = W^{old} - \alpha \, (X^T (FX - Y) + 2 \lambda W^{old})$$

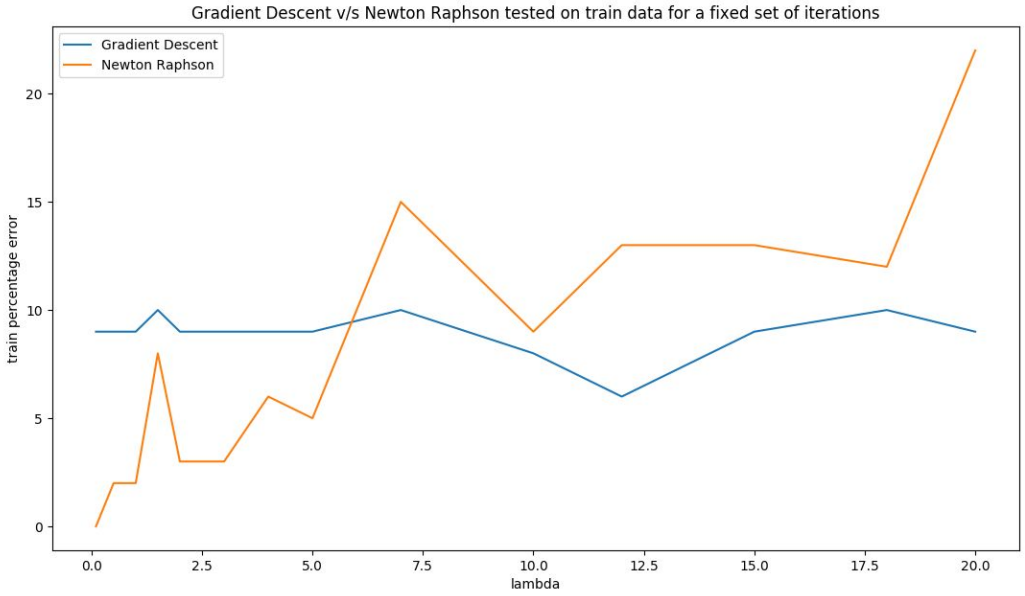
And the analytical solution for **Newton Raphson** method is :

$$W^{new} = W^{old} - (X^T R X + 2\lambda I)^{-1} (X^T (FX - Y) + 2 \lambda W^{old})$$

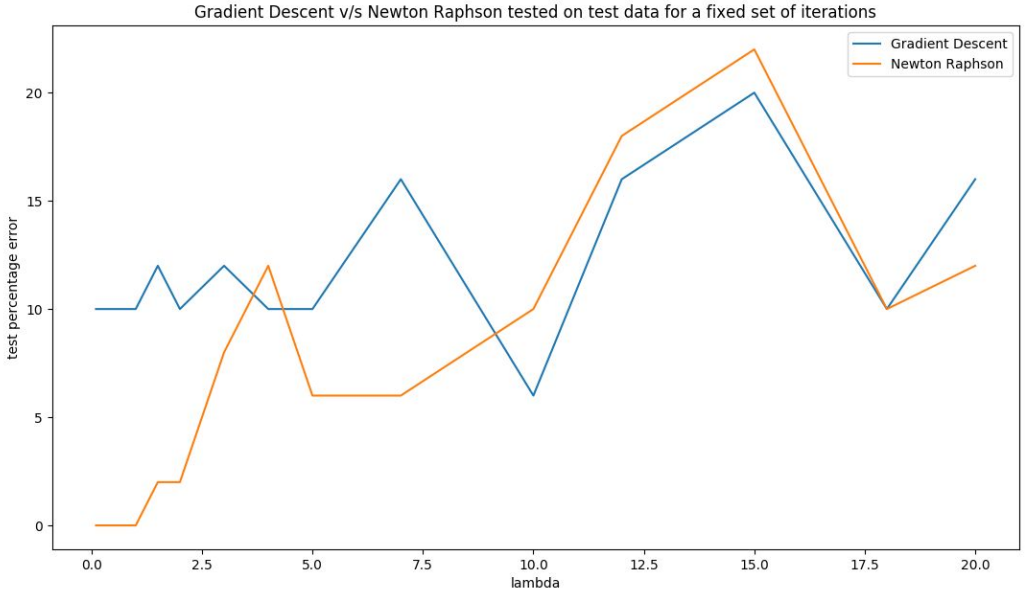
Where R is a diagonal matrix having ith entry as $fx^i (1 - fx^i)$

Comparison of performance : The performance of gradient descent and Newton raphson was compared using fixed set of iterations. In this case, the iterations were fixed to 1000.

a) Performance comparison on train data



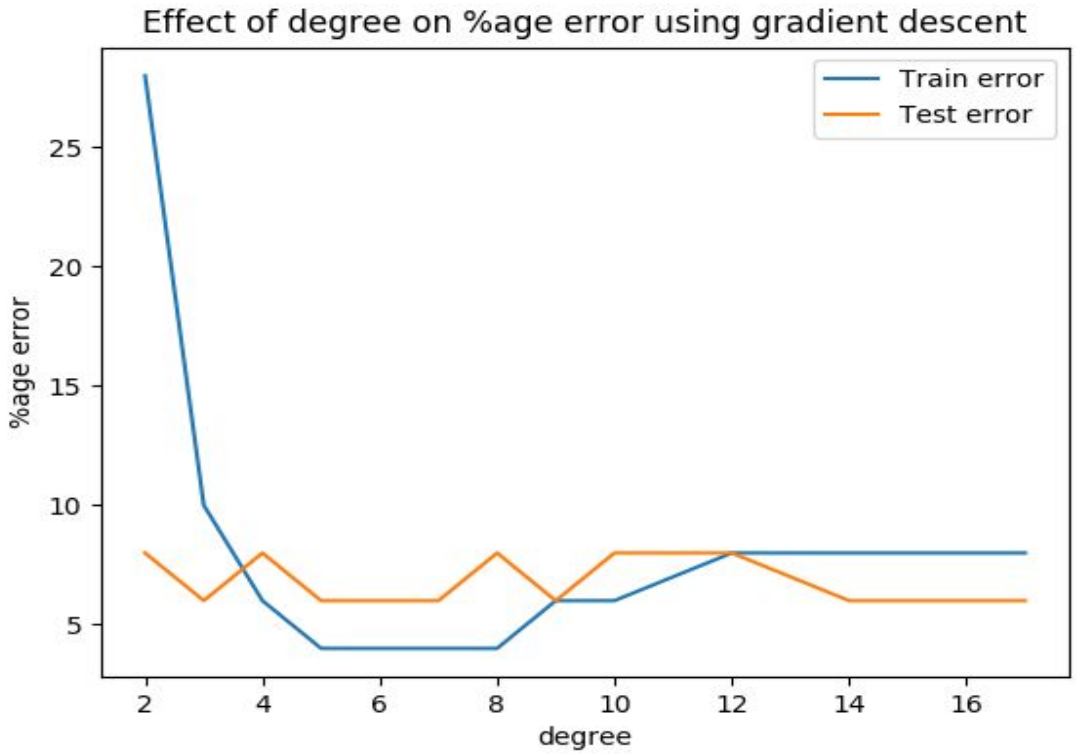
b) Performance comparison on test data



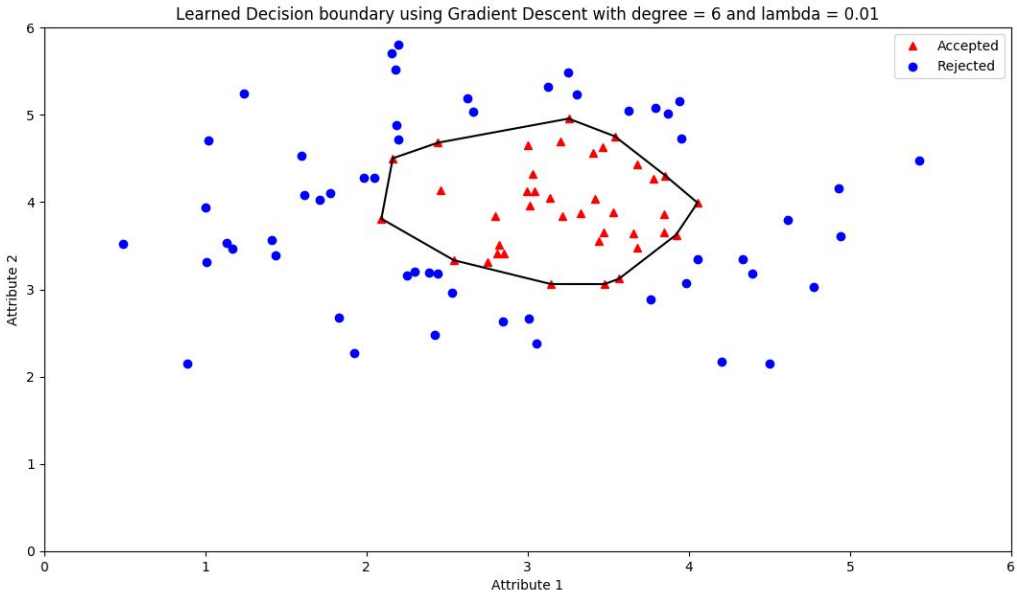
For the training set, Newton Raphson performed better than gradient descent and for the test set, gradient descent performed better for lower values of lambda whereas newton raphson performed better for higher values of lambda. Since there were only 20 examples in test set while training set contained 80 examples. Hence we can conclude that Newton Raphson performed better.

Ques : Is the data linearly separable?
Ans : No, the data is not linearly separable as it can be seen from the data plot. And since logistic regression can only model linearly separable data, hence to resolve this problem we use feature transformation.

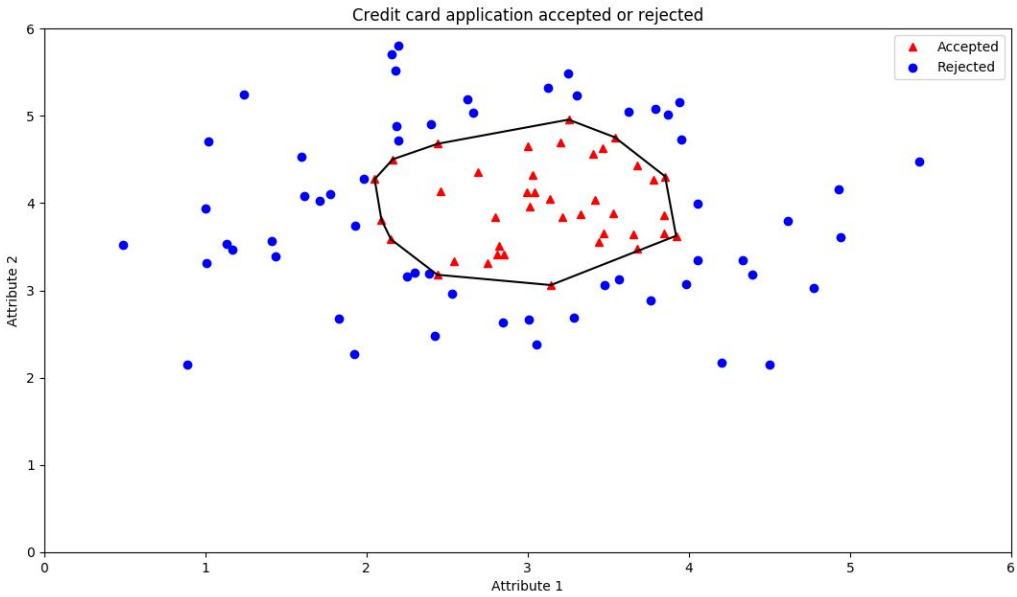
Feature Transformation : Various degrees for feature transformation were taken and the test and train error was computed. Following is the plot which shows that error decreases with increasing degree but after a certain point the decrease in error is not significant, hence we can conclude that **degree 6 is the most optimal degree**.



Decision boundary learned by feature transformation : Following is the plot of decision boundary learned by feature transformation using degree = 6 and lambda = 0.01



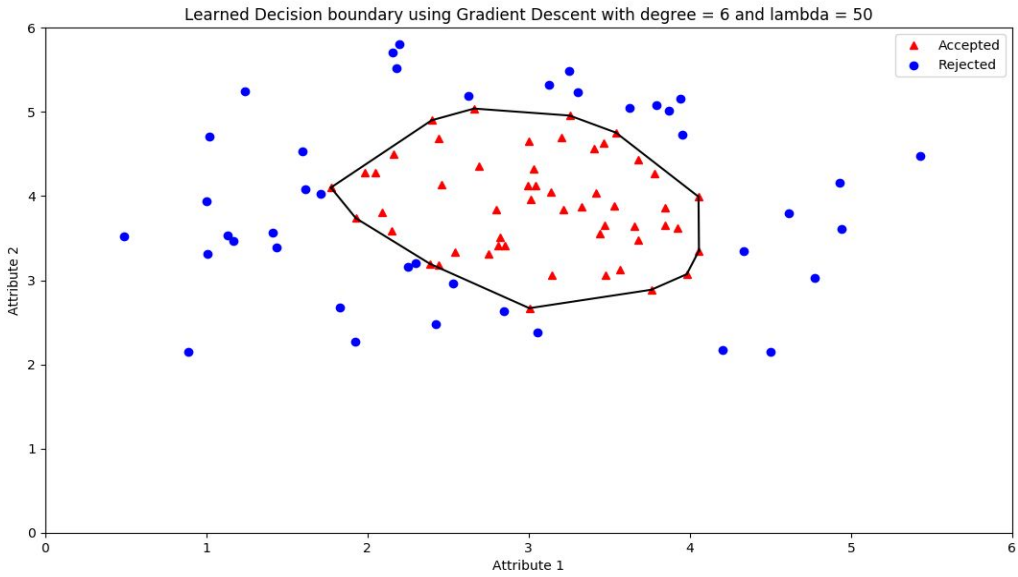
Whereas the ideal decision boundary (error = 0) is given below :



As can be seen from the plot, feature transformation learns a better model than logistic regression without feature transformation.

Varying Regularization Parameter : Values of lambda were varied to obtain the best fit model. Some of the lambdas resulted in under fitting and over fitting. Given below are the plots.

a) **Over fitting** : For lambda = 50



b) **Under fitting** : For lambda = 1