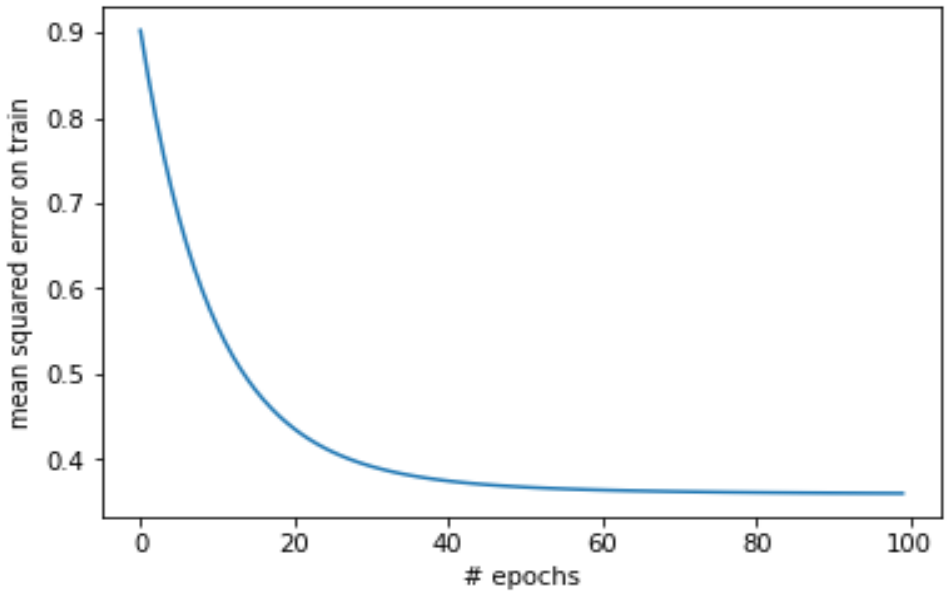
Machine Learning (CS60050): Assignment 1

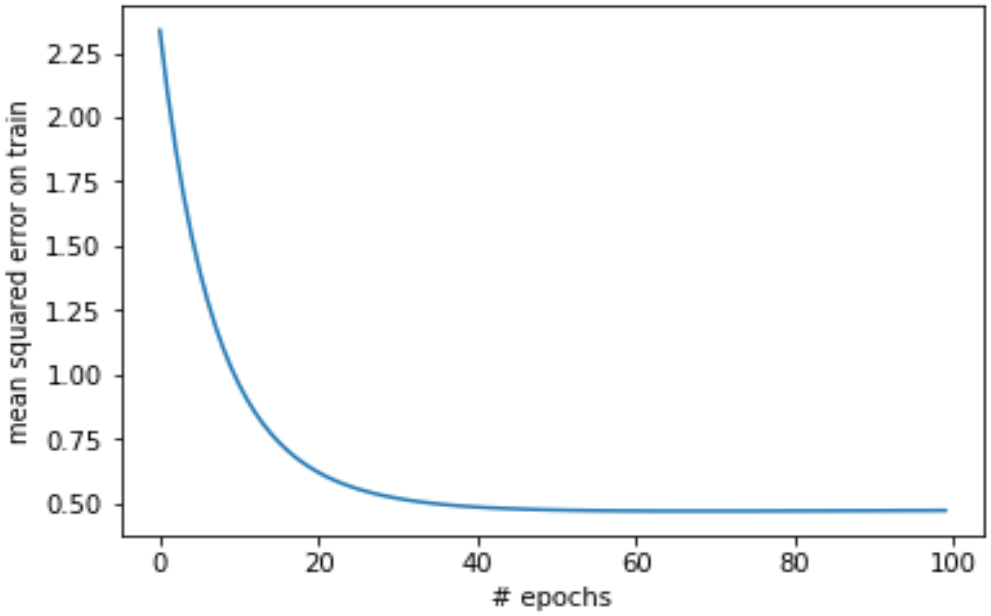
(a): implementing linear regression

We implement linear regression with learning rate = 0.05, with & without regularization, and after training for 100 epochs, these are the following results that have been obtained :

Mean squared error vs. No. of iterations with no regularization



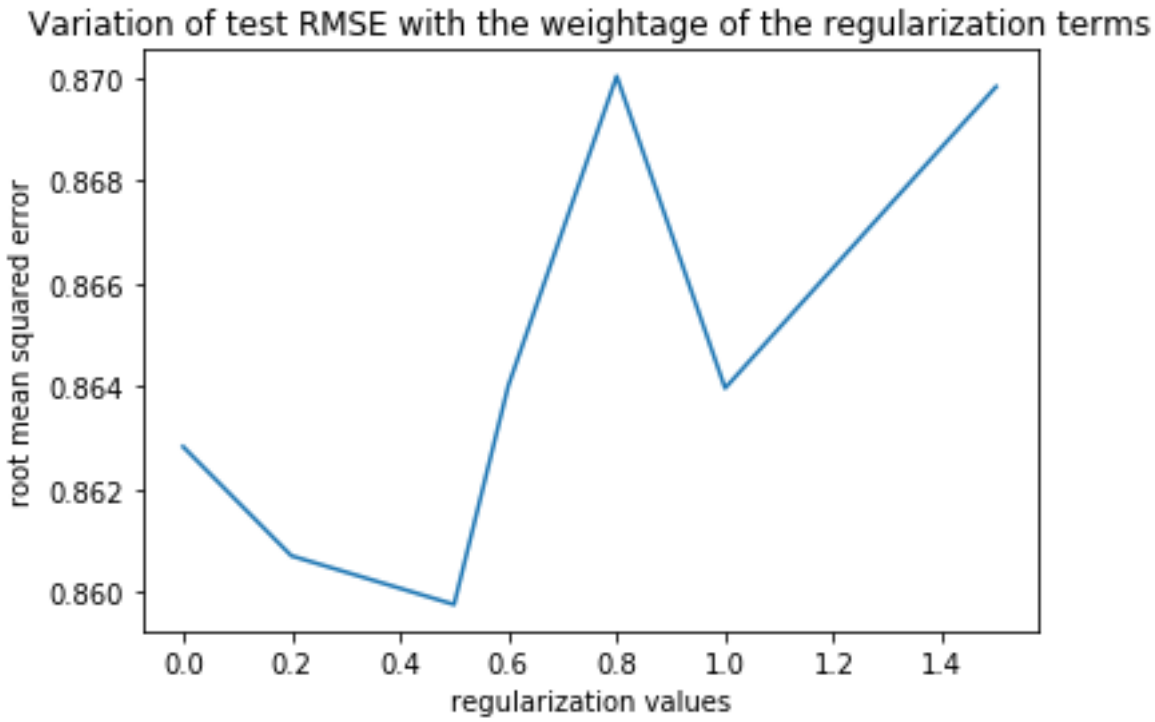
Mean squared error vs. No. of iterations with regularization=0.5



The above graphs indicate that with regularization, linear regression converges faster. The following test root mean square values with and without regularization show that regularization indeed reduces error (as it reduces over fitting).

For Regularization value = 0,
learnt parameters = [0.01932303 0.05278265 0.07852283 0.09533942 0.40614033]
test RMSE = 0.862126198718

For Regularization value=0.5
learnt parameters = [0.02355376 0.04307771 0.03389674 0.04489703 0.48497923]
test RMSE = 0.859471534236

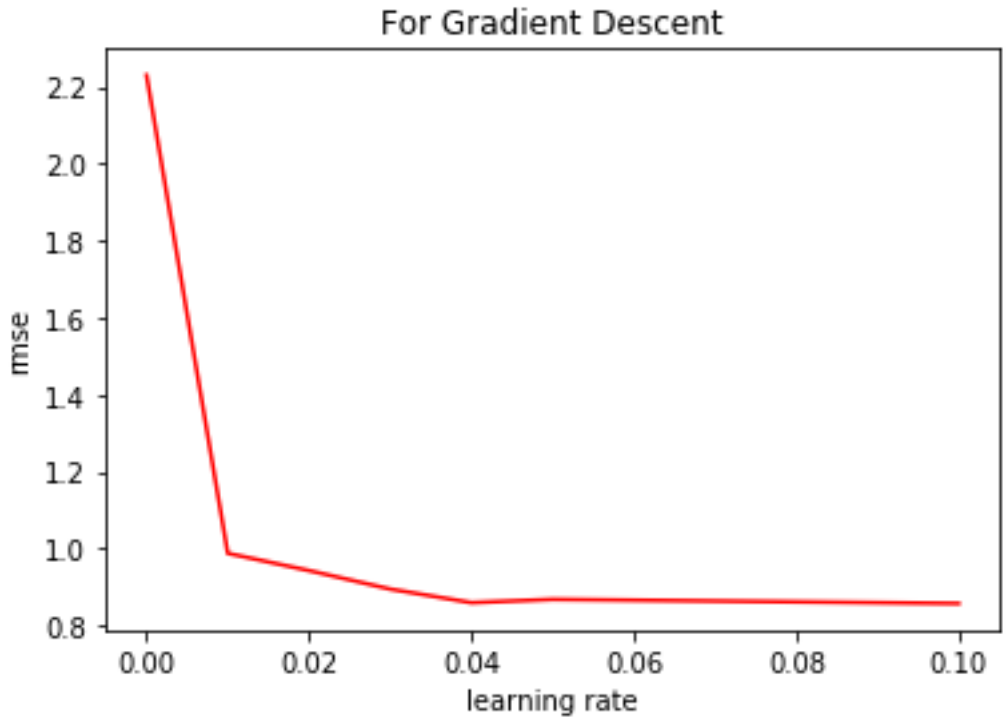


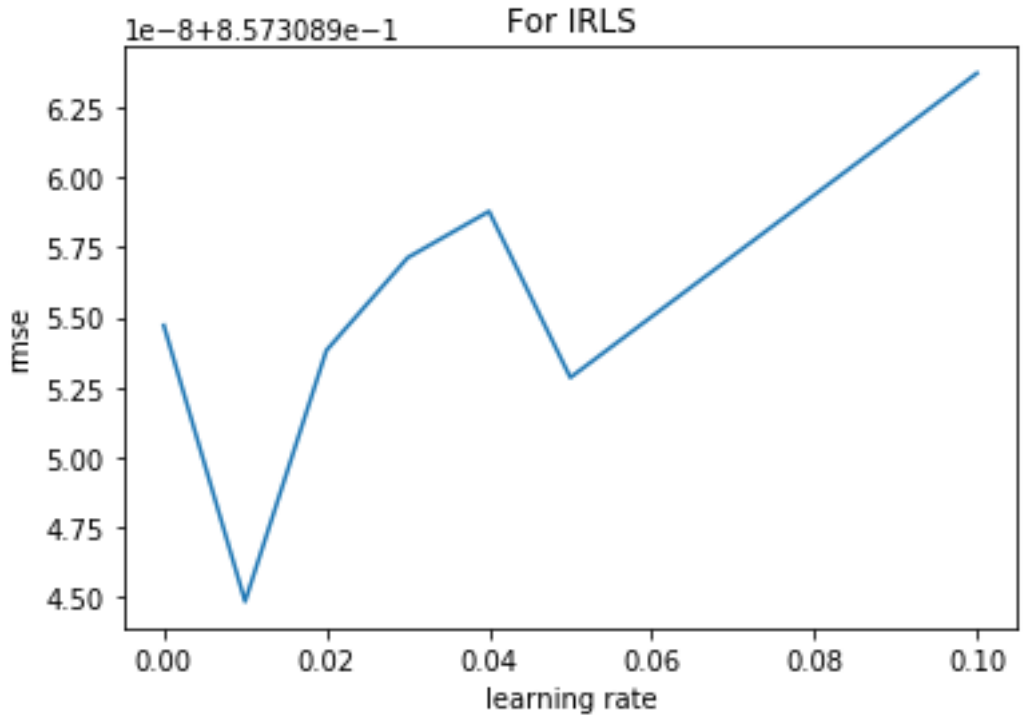
Finally we experiment with different regularization values and the above graph is the result that I have obtained. Here we can see, that the regularization value for which the model learns the best is 0.5.

(b): experimenting with optimization algorithms

When we experimented with *Gradient Descent* & *Iterative Reweighted Least Square* optimization algorithms we found the following results :

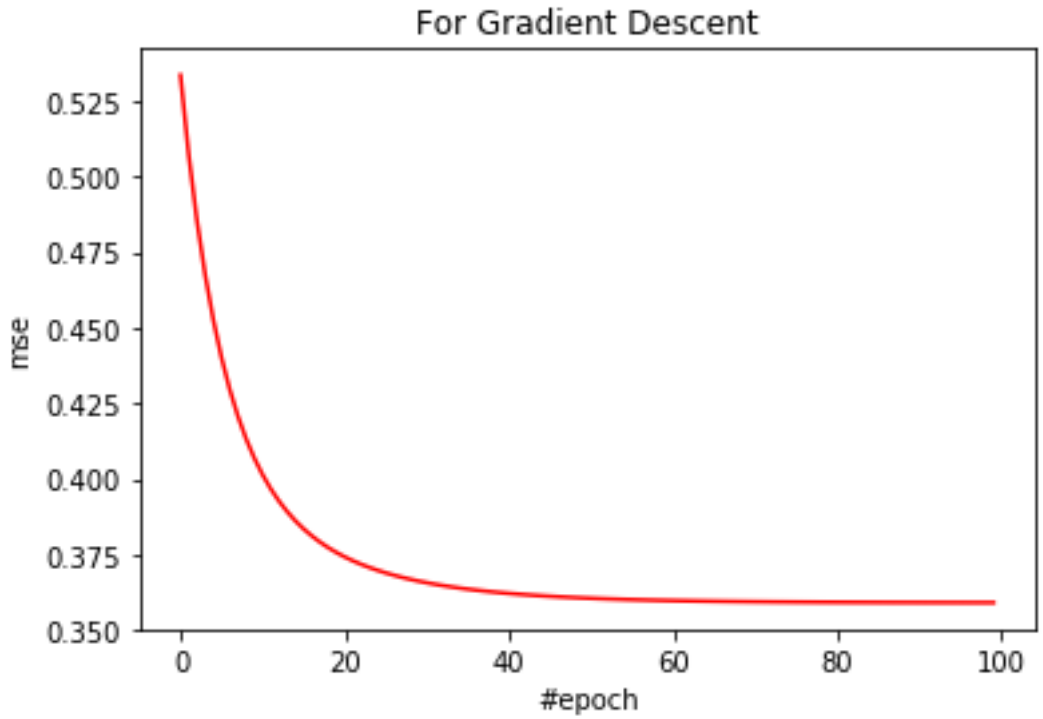
Variation of RMSE Vs. Learning Rate for both the algorithms.

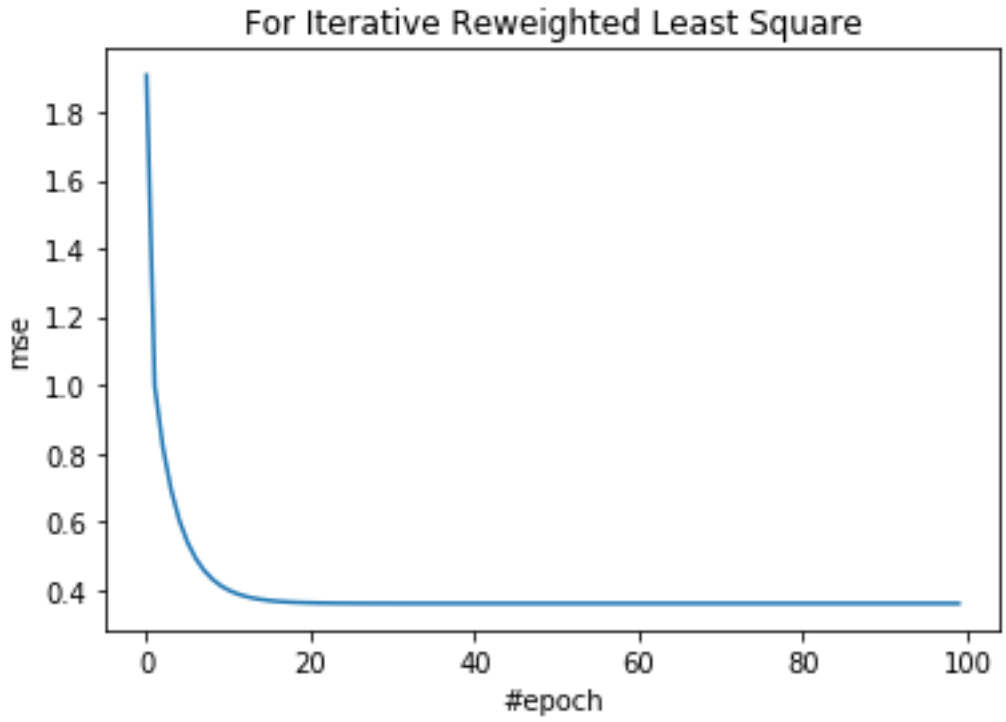




For both gradient descent & IRLS, we see, the learning rate for which convergence occurs best is almost 0.01.

To have a clearer perception of the results obtained by the two algorithms and compare, I tried plotting the mean squared error (mse) vs. the no. of epochs required to train the model (=100, in this case).





The above graph clearly shows that **for IRLS, the convergence is way faster.**

Also, the parameters learnt & test RMSE values for both the algorithms are as follows :

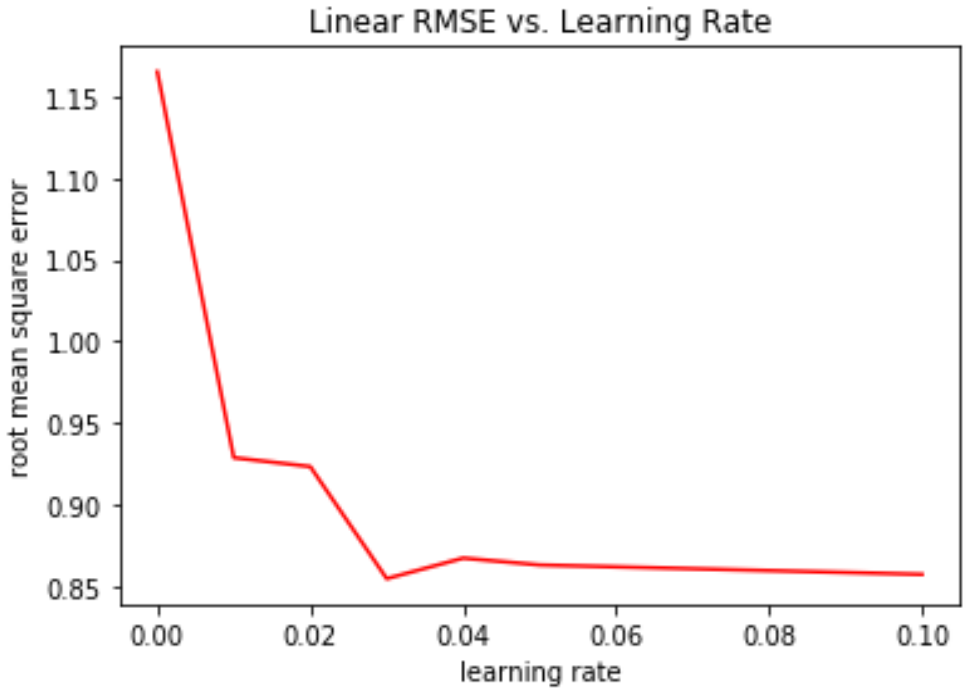
Gradient descent :
learnt parameters = [0.01725256 0.04365408 0.036922 0.05080936 0.47634543]
test RMSE = 0.857912540967

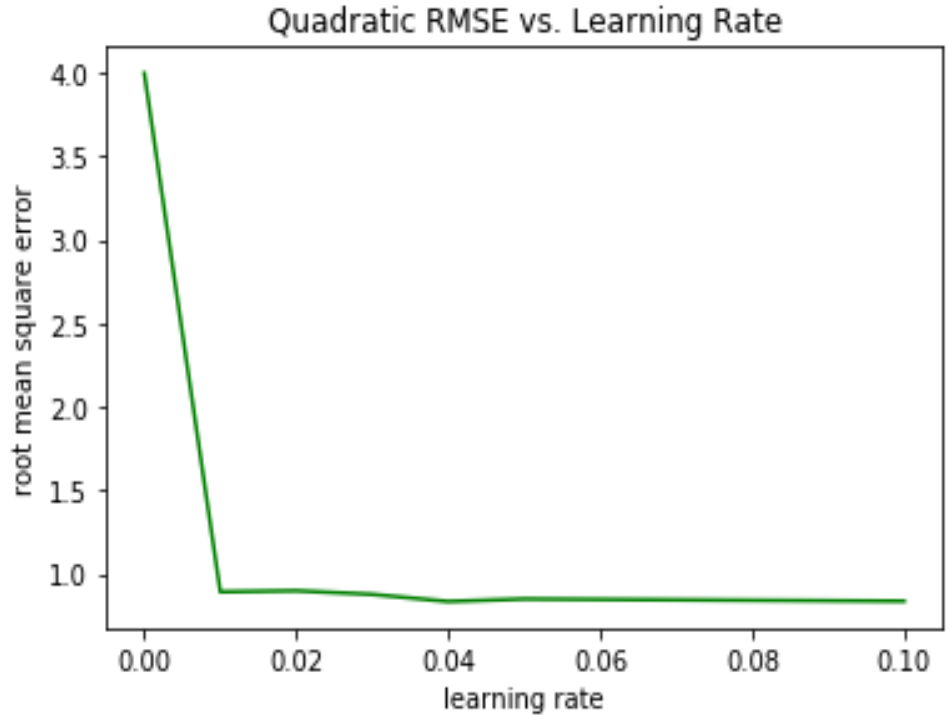
IRLS :
learnt parameters = [0.01713954 0.042573 0.02962148 0.04425472 0.48756371]
test RMSE = 0.857308963704

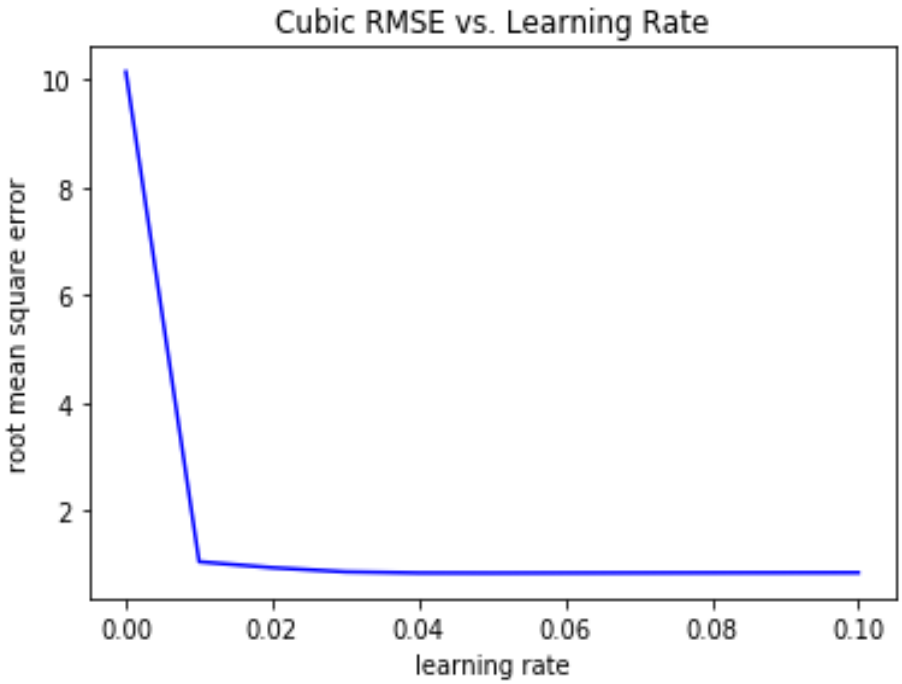
Athough by a very small margin here, IRLS fares better than Gradient descent and hence the optimization algorithm *preferred for this problem would be IRLS*.

(c): experimenting with combinations of features

Here, on experimenting with linear, quadratic & cubic combination of features these are the following results that has been obtained :







The learnt parameters & test RMSE value in each case are as follows :

Linear Combination of features :

learnt parameters = [0.01716213 0.04290721 0.03183295 0.04635981 0.48406008]

 # parameters = 5

test RMSE = 0.857473752012

Quadratic Combination of features :

learnt parameters = [0.01481525 0.3213335 0.26599639 -0.14832613 -0.2618323 -0.0178304
0.12278574 -0.33342723 -0.06470961 -0.32378776 0.13923249 -0.03030624
-0.00454273 0.31693901 0.59403991]

 # parameters = 15

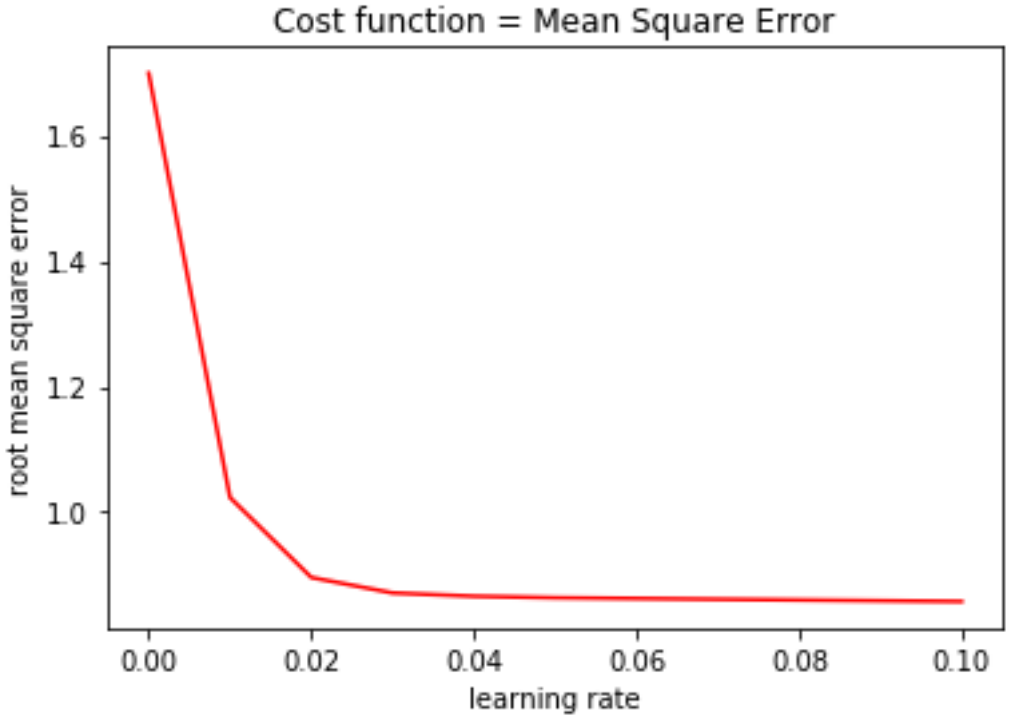
test RMSE = 0.837148068502

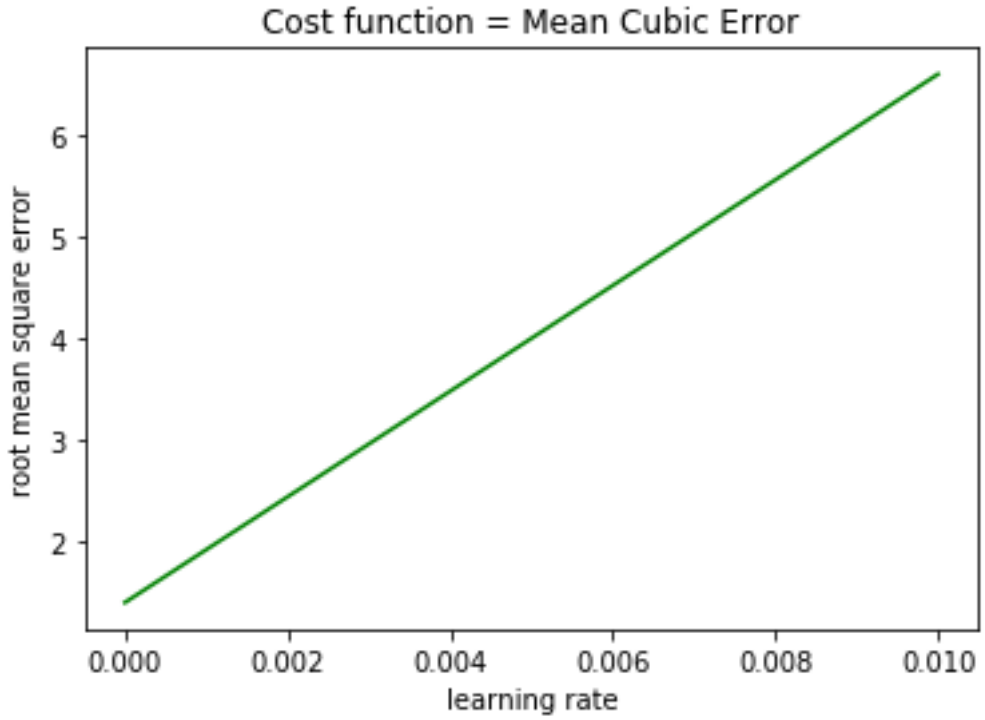
Cubic Combination of features :
learnt parameters = [0.01889859 0.39586497 -0.03439636 0.3095037 0.25797467 0.12390052
-0.07840686 -0.18056645 -0.30953812 -0.09634454 -0.03855116 -0.32707398
0.10535533 -0.27585519 -0.0469034 -0.00659081 -0.25612711 0.02629165
0.09618168 0.22215336 -0.04645921 0.00370582 -0.16060993 0.28889197
-0.07273581 0.07690124 0.19953808 0.24911361 -0.41033273 0.24759882
0.04782645 0.08724439 -0.02529049 0.03447117 0.45712765]
 # parameters = 35
test RMSE = 0.852431631623

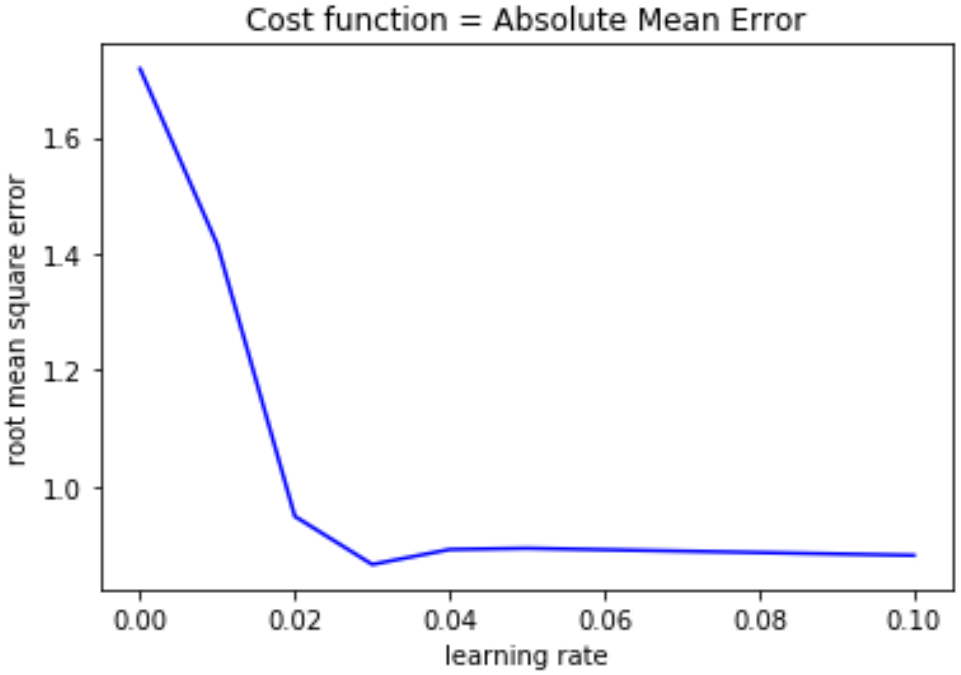
We can see that quadratic combination of features incurs the least test RMSE. Although here they differ by small margins, but for this problem the *preferred combination of features would be quadratic.*

(d): experimenting with cost functions

When experimenting with different cost functions, the following results are obtained :







For mean cube error, the result diverges after a certain number of epochs. That is probably because it gives more weight to outliers. On the other hand, the squared error is differentiable and simple to work with without being too sensitive to outliers. The learnt parameters and respective test RMSE values are :

Using mse:
learnt parameters =[0.0173574 0.04410394 0.04019295 0.0535138 0.47153349]
test RMSE=0.858232026637
Using mce :
learnt parameters=[nan nan nan nan nan]
test RMSE=nan
Using ame :
learnt parameters=[-0.14836316 0.04879275 0.05952685 0.05333781 0.30285835]
test RMSE =0.880962363419

We can see that for mean squared error, the RMSE obtained is least. Thus, keeping all the discussed factors in my mind, *Mean Squared Error is the preferred cost function.*