Machine Learning- Assignment 1
Report (15CS10053)

For all the parts, we had to first normalise the features to bring them to the same scale. For column X, we used: (X-mean(X))/stddev(X) as the feature. The theta reported here are obtained after transforming back to denormalized form.

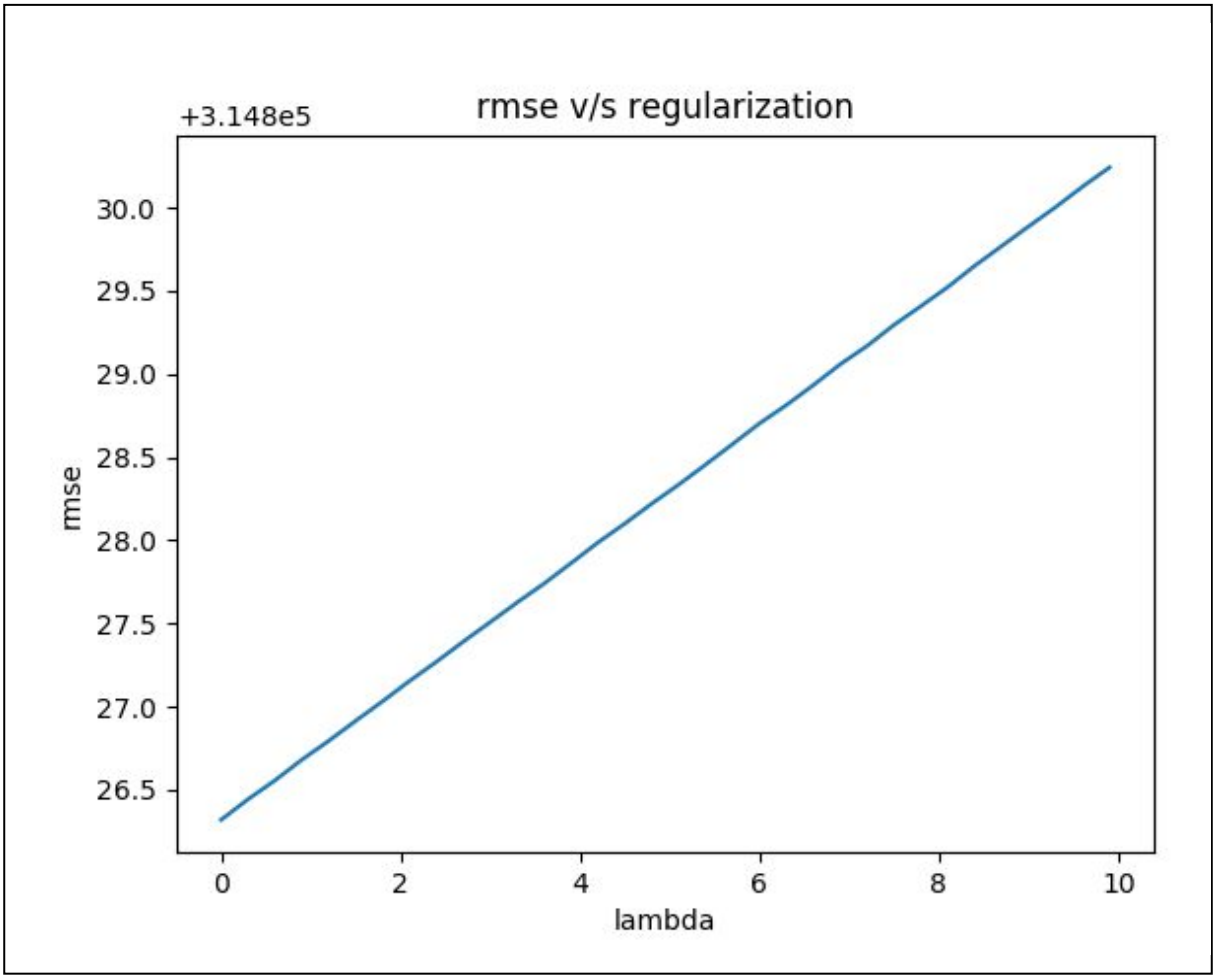
Part (a)

The final learned values for model parameters are:
Without regularization:
Theta = [-3.97954130e+04 3.77378156e-01 2.01371613e+04 1.74726689e+04
  2.32404280e+05]

RMSE = 314733.8019

With regularization (*lambda = 0.9*)
Theta = [-5.99811209e+04 4.20979927e-01 3.30319664e+04 2.87114280e+04
  2.12561547e+05]

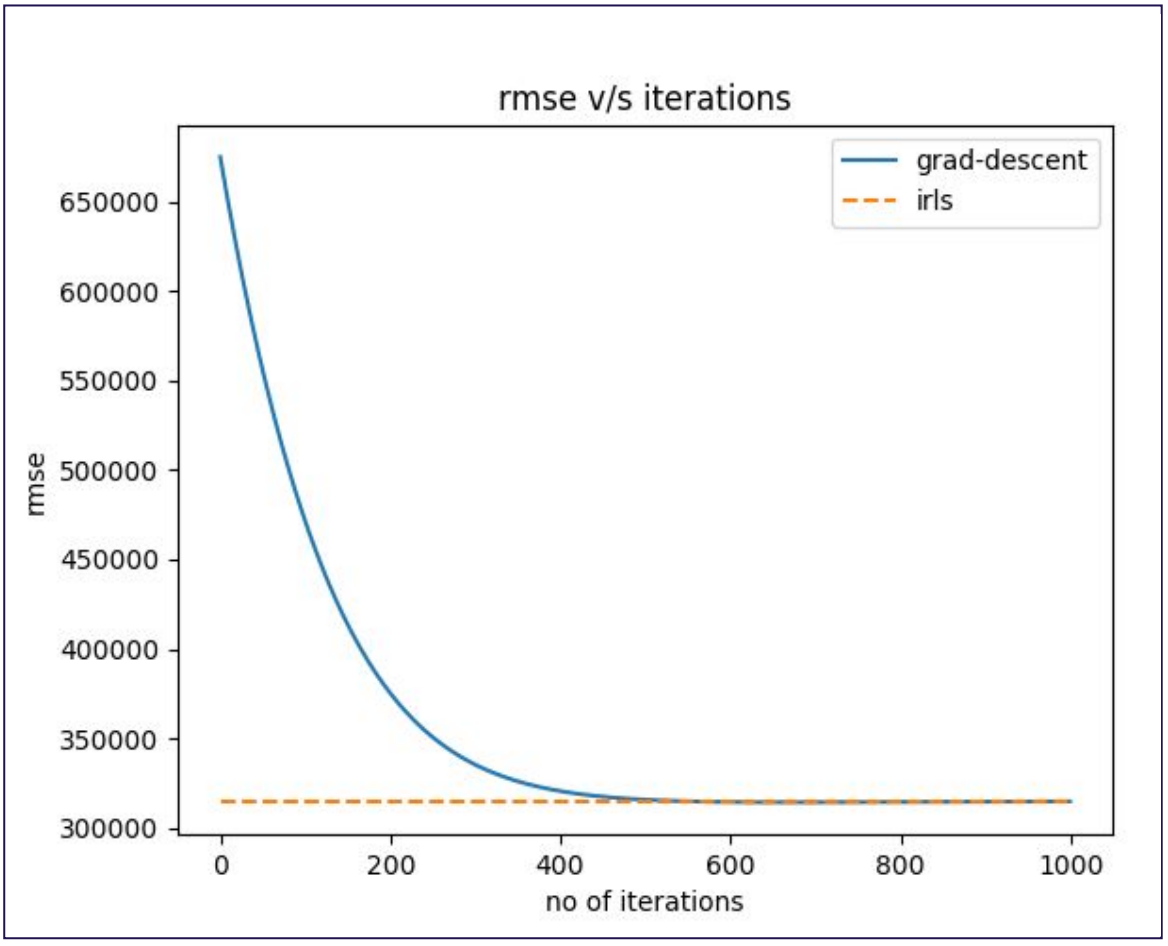
RMSE = 314826.679



Part (b)

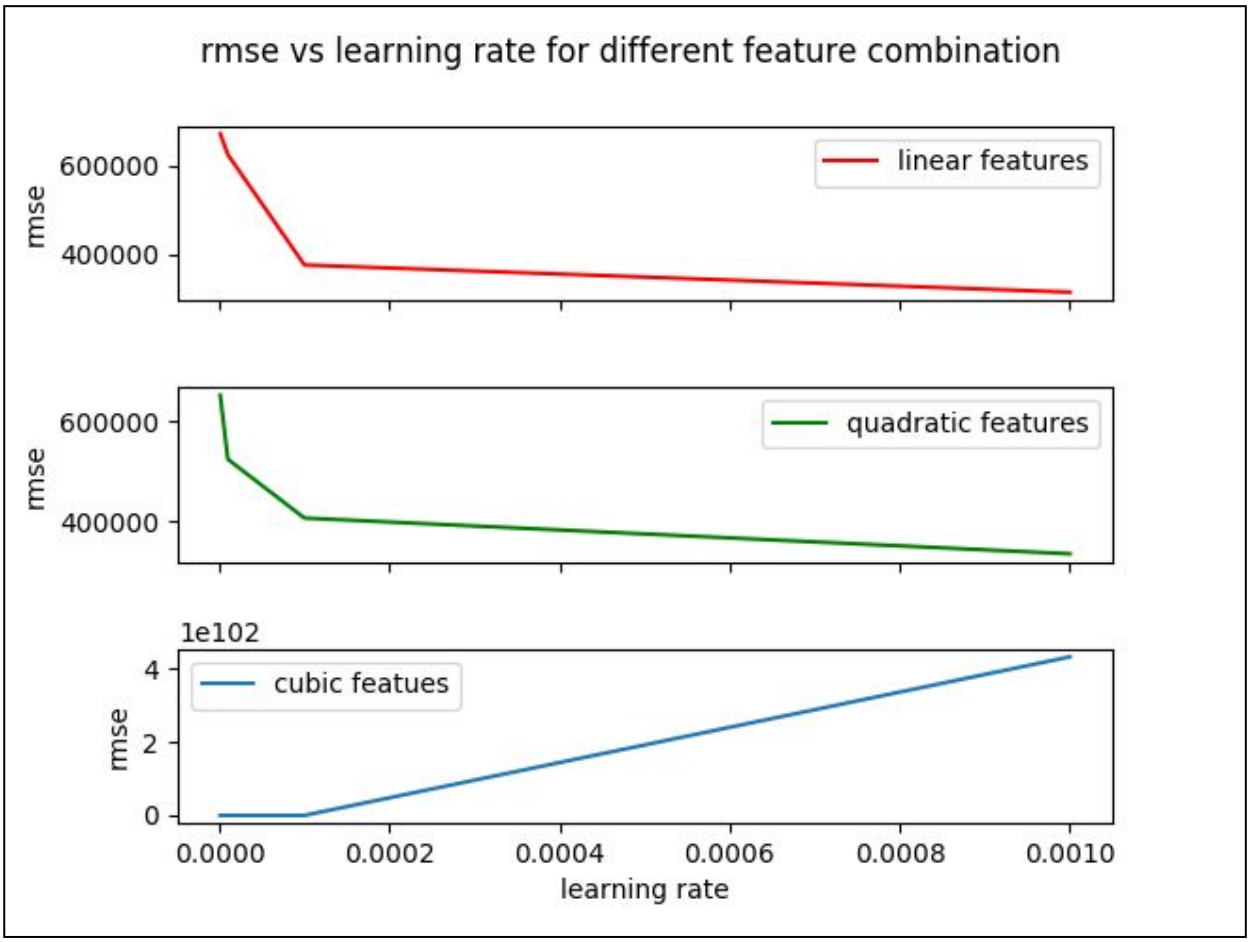
Theta obtained using ***gradient descent*** is [-5.99923379e+04 4.20985968e-01 3.30267952e+04 2.87091982e+04 2.12574161e+05]
RMSE = 314826.321

Theta obtained using ***IRLS*** is [-3.97919650e+04 3.77371497e-01 2.01336367e+04 1.74708748e+04 2.32408029e+05]
RMSE = 314733.654



We see that Iterative reweighted least square method (IRLS) gives optimum parameter values in a single step. It is a *closed form solution for the convex optimization problem*. Gradient descent algorithm also converges to optimum value after ~500 iterations. We would **prefer IRLS** in this case because it is possible to obtain the optimum model parameter values in a single step.

Part (c)



After running 10000 iterations for linear and quadratic combination of features and 40 iterations (as it diverges) for cubic combination of features.

Model parameter values *which give least rmse* obtained for:

(i) linear combination of features

[-4.28166392e+04 3.83279735e-01 2.30173433e+04 1.90888626e+04
2.29168074e+05]

for alpha = 0.001 , RMSE = 314847.2276

(ii) quadratic combination of features

[1.61183164e+05 1.31864487e-02 4.16259792e+04 -3.25119452e+01
9.98179247e+04]

for alpha = 0.001, RMSE = 352232306.409

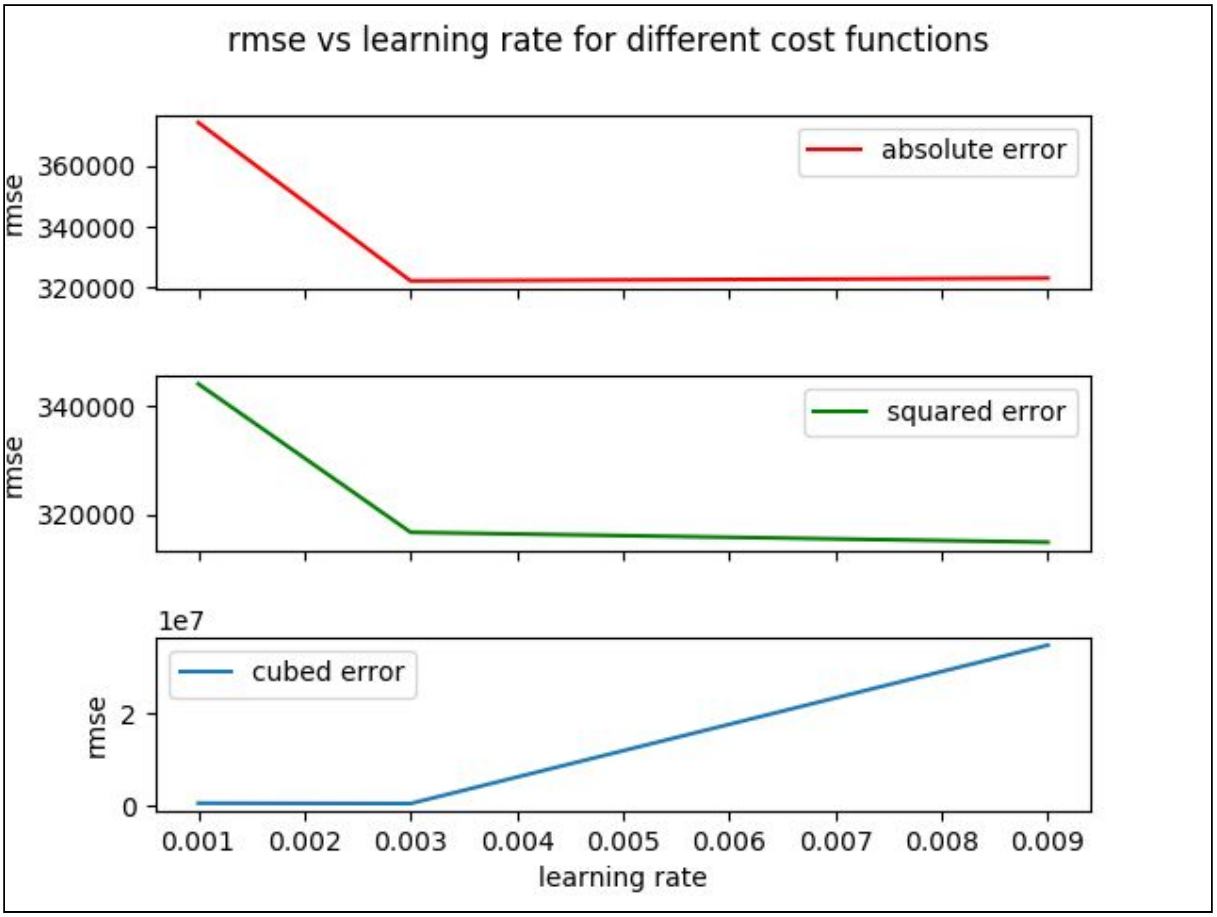
(iii) cubic combination of features

[-2.91657203e+02 9.07040502e-04 2.16224498e+01 3.03913197e+01
7.79961339e+01]

for alpha = 0.000001 , RMSE = 24166087508089.004

We would **prefer linear combination of features** because the cost reduced most rapidly with rate in this case. With quadratic combination also the cost reduced but the drop wasn't as quick. Cubic features lead to divergence with higher learning rate.

Part (d)



After running 1000 iterations for mean absolute error and mean squared error and 110 iterations (as it diverges) for mean cubed error.

Model parameter values which give least rmse obtained for:

(i) mean absolute error
[4.06933970e+04 4.18719126e-01 4.23879179e+04 2.18497563e+04 1.42560942e+05]
for alpha = 0.009, RMSE = 323657.0638

(ii) mean squared error
[-4.77068315e+04 3.92598663e-01 2.83021694e+04 2.15763148e+04
2.23782163e+05]
for alpha = 0.009, RMSE = 315129.4690

(iii) mean cubed error
[-1.29615841e+06 1.30802368e+00 4.14516860e+05 2.39699032e+05 2.21115025e+05]
for alpha = 0.001, RMSE = 493414.4367

We would **prefer mean squared error** because it is *differentiable* at all points unlike mean absolute error. It is also *convex error function* unlike mean cubed error, so, there exists a global minimum cost value. As we can see from the plot, mean cubed error function doesn't converge which is not surprising considering the nature of a cubic function.