State University of New York at Buffalo

CSE 574 – Introduction to Machine Learning

Spring 2017

Programming Assignment -2

Classification and Regression Project Report

Group 32

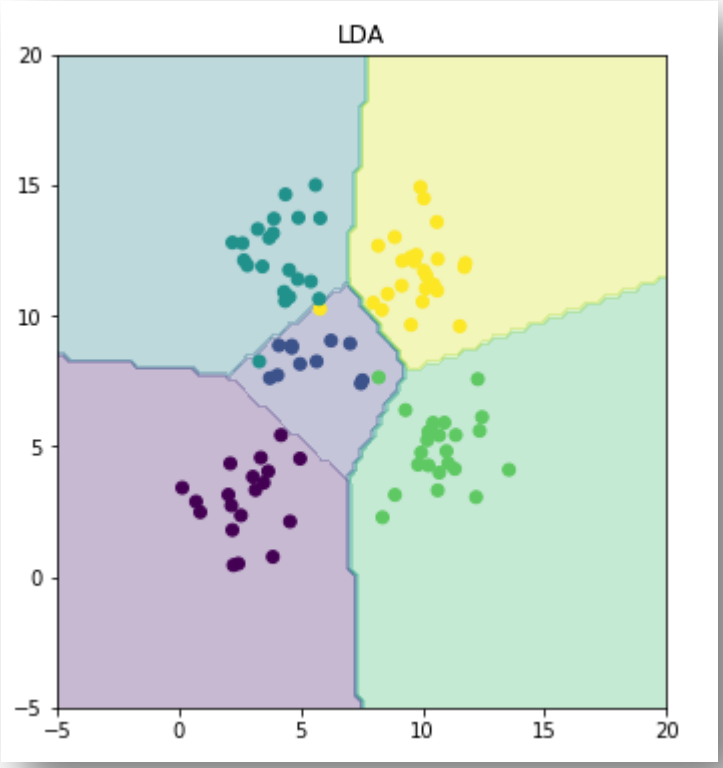
Ajay Gandhi – 50207403 – agandhi3@buffalo.edu

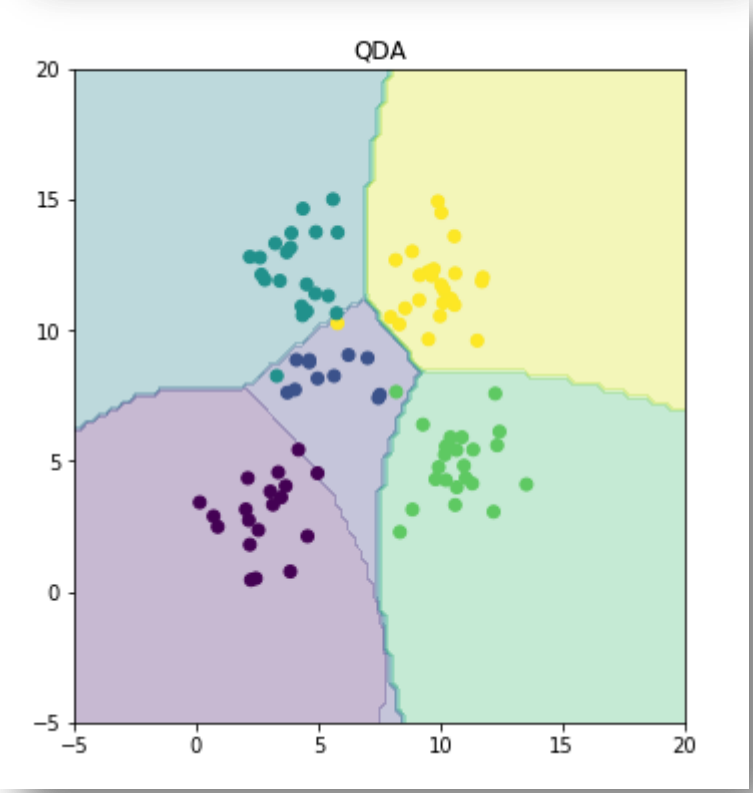
Apoorva Hejib – 50206516 – apoorvah@buffalo.edu

Siddharth Pateriya – 50206348 – spateriy@buffalo.edu

Problem 1: Experiment with Gaussian Discriminators

LDA Accuracy = 97.0 %
QDA Accuracy = 96.0 %





We can see the difference in the boundaries of LDA and QDA in the figures above. LDA boundaries form fairly straight lines, whereas the QDA boundaries depict curves rather than straight lines. This can be attributed to the nature of LDA and QDA; LDA takes into account the covariance of the entire data, whereas QDA utilizes covariance of each class of the training data.

Problem 2: Experiment with Linear Regression

Results based on Training Data

MSE without Intercept: 19099.4468446

MSE with Intercept: 2187.16029493

Results based on Test Data

MSE without Intercept: 106775.361558

MSE with Intercept: 3707.84018179

We can see from the above results that for both the training and test data, the MSE taken with the intercept is much lower than compared to what it is taken without intercept. **Thus, we can conclude that for the purpose of regression, calculating the MSE with the Intercept is the better approach.**

Problem 3: Experiment with Ridge Regression

MSE from OLE for Training Data (with Intercept) : 2187.16029493

MSE from OLE for Test Data (with Intercept) : 3707.84018179

MSE values for Ridge Regression (Varying Lambda from 0 to 1)

Lambda	Training Error	Test Error
0	2187.16	3707.840182
0.01	2306.83	2982.44612
0.02	2354.07	2900.973587
0.03	2386.78	2870.941589
0.04	2412.12	2858.00041
0.05	2433.17	2852.665735
0.06	2451.53	2851.330213
0.07	2468.08	2852.349994
0.08	2483.37	2854.879739
0.09	2497.74	2858.444421
0.1	2511.43	2862.757941
0.11	2524.60	2867.637909
0.12	2537.35	2872.962283
0.13	2549.78	2878.645869
0.14	2561.92	2884.626914
0.15	2573.84	2890.85911
0.16	2585.56	2897.306659
0.17	2597.11	2903.941126
0.18	2608.50	2910.739372
0.19	2619.75	2917.682164
0.2	2630.87	2924.753222
0.21	2641.88	2931.938544
0.22	2652.77	2939.22593
0.23	2663.56	2946.604624

Lambda	Training Error	Test Error
0.24	2674.25	2954.065056
0.25	2684.85	2961.598643
0.26	2695.35	2969.197637
0.27	2705.76	2976.855001
0.28	2716.08	2984.564321
0.29	2726.32	2992.319722
0.3	2736.47	3000.115809
0.31	2746.54	3007.947616
0.32	2756.53	3015.810555
0.33	2766.44	3023.700386
0.34	2776.27	3031.613181
0.35	2786.03	3039.545297
0.36	2795.70	3047.493351
0.37	2805.30	3055.454198
0.38	2814.83	3063.424913
0.39	2824.28	3071.402772
0.4	2833.66	3079.385238
0.41	2842.97	3087.369947
0.42	2852.21	3095.354694
0.43	2861.37	3103.337424
0.44	2870.47	3111.316218
0.45	2879.50	3119.289287
0.46	2888.46	3127.254961
0.47	2897.35	3135.211679

Lambda	Training Error	Test Error
0.48	2906.18	3143.157988
0.49	2914.94	3151.09253
0.5	2923.63	3159.014036
0.51	2932.26	3166.921324
0.52	2940.83	3174.813291
0.53	2949.33	3182.688908
0.54	2957.77	3190.547215
0.55	2966.15	3198.387318
0.56	2974.47	3206.208382
0.57	2982.73	3214.009633
0.58	2990.93	3221.790346
0.59	2999.07	3229.549851
0.6	3007.16	3237.287523
0.61	3015.18	3245.002781
0.62	3023.15	3252.695087
0.63	3031.07	3260.363943
0.64	3038.92	3268.008886

Lambda	Training Error	Test Error
0.65	3046.73	3275.629488
0.66	3054.48	3283.225355
0.67	3062.17	3290.796124
0.68	3069.82	3298.341459
0.69	3077.41	3305.861052
0.7	3084.95	3313.354623
0.71	3092.43	3320.821913
0.72	3099.87	3328.262686
0.73	3107.26	3335.676731
0.74	3114.60	3343.063853
0.75	3121.89	3350.423878
0.76	3129.13	3357.75665
0.77	3136.32	3365.062031
0.78	3143.47	3372.339896
0.79	3150.57	3379.590137
0.8	3157.62	3386.812661
0.81	3164.63	3394.007386

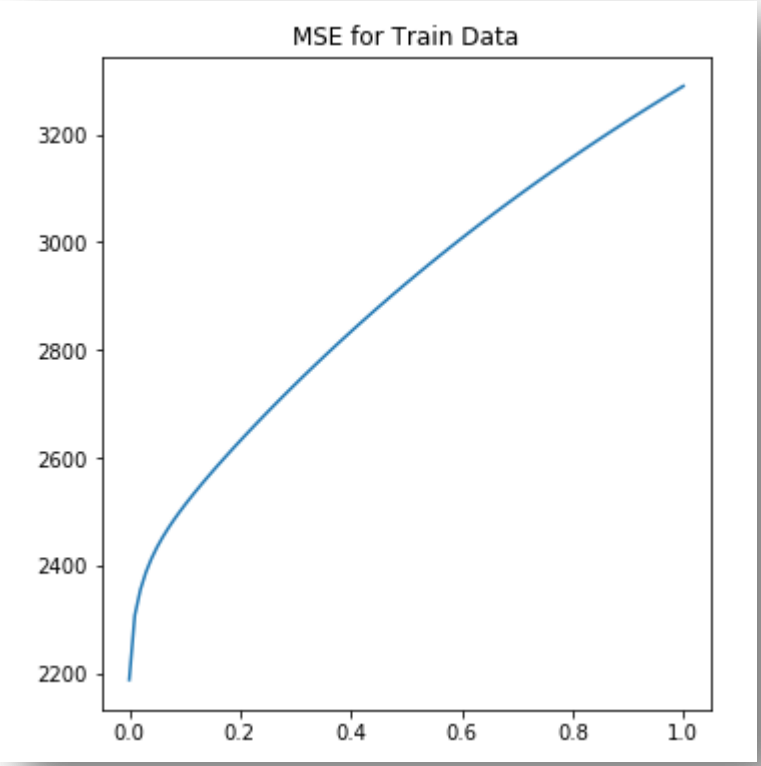
Lambda	Training Error	Test Error
0.82	3171.59	3401.174246
0.83	3178.51	3408.313184
0.84	3185.39	3415.424154
0.85	3192.22	3422.507124
0.86	3199.01	3429.562069
0.87	3205.75	3436.588973
0.88	3212.45	3443.587832
0.89	3219.12	3450.558648
0.9	3225.74	3457.50143
0.91	3232.31	3464.416198
0.92	3238.85	3471.302975
0.93	3245.35	3478.161794
0.94	3251.81	3484.992692
0.95	3258.23	3491.795713
0.96	3264.61	3498.570906
0.97	3270.96	3505.318324
0.98	3277.26	3512.038029
0.99	3283.53	3518.730082
1	3289.76	3525.394553

On arranging the above table in ascending order of Test Error, **the top 10 rows** are:

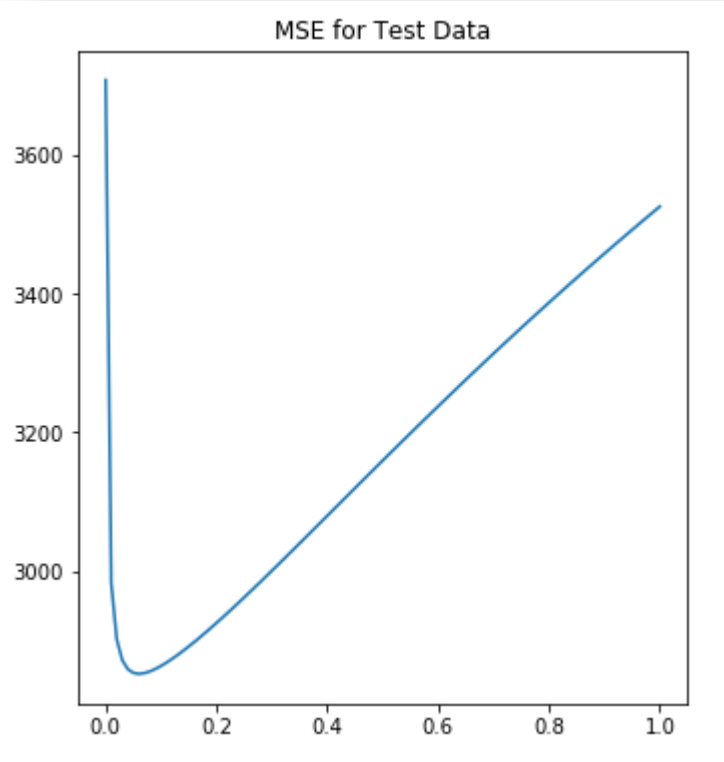
Lambda	Training Error	Test Error
0.06	2451.53	2851.330213
0.07	2468.08	2852.349994
0.05	2433.17	2852.665735
0.08	2483.37	2854.879739
0.04	2412.12	2858.00041
0.09	2497.74	2858.444421
0.1	2511.43	2862.757941
0.11	2524.60	2867.637909
0.03	2386.78	2870.941589
0.12	2537.35	2872.962283

Thus we see that MSE on test data is least when Lambda is 0.06, and hence is the optimal value.

Error in Ridge Regression (varying with lambda) for test and training Data:



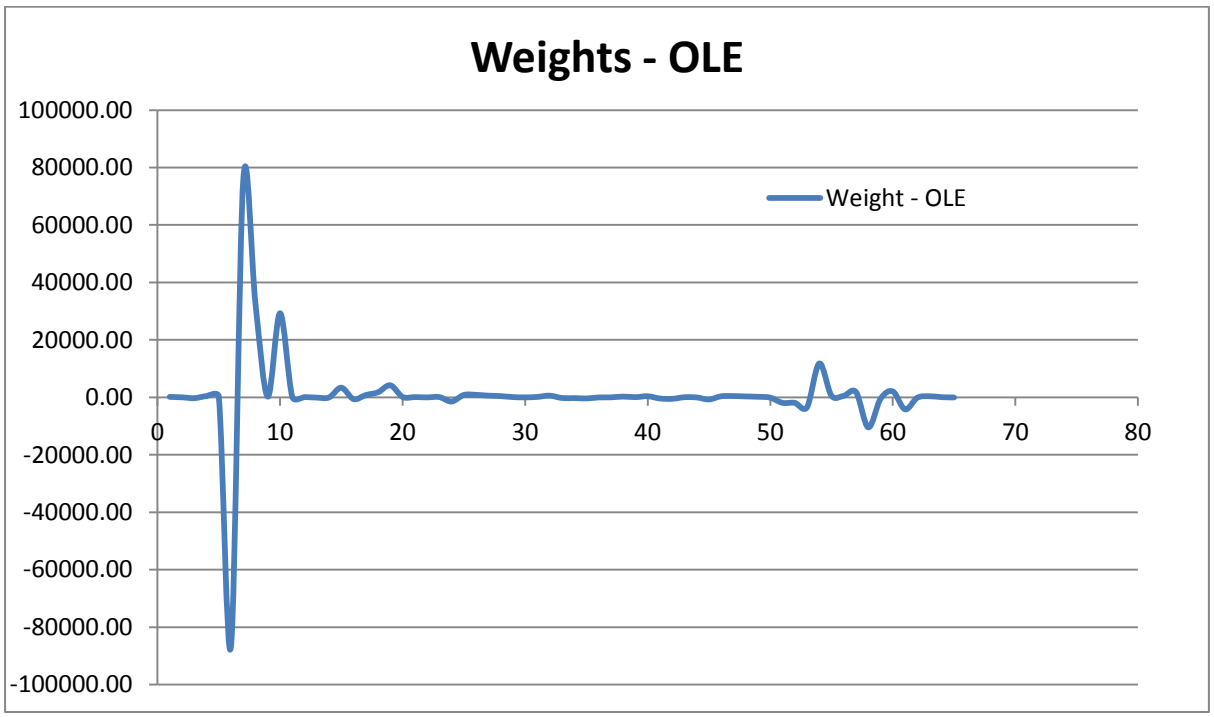
MSE for Training Data varying with Lambda

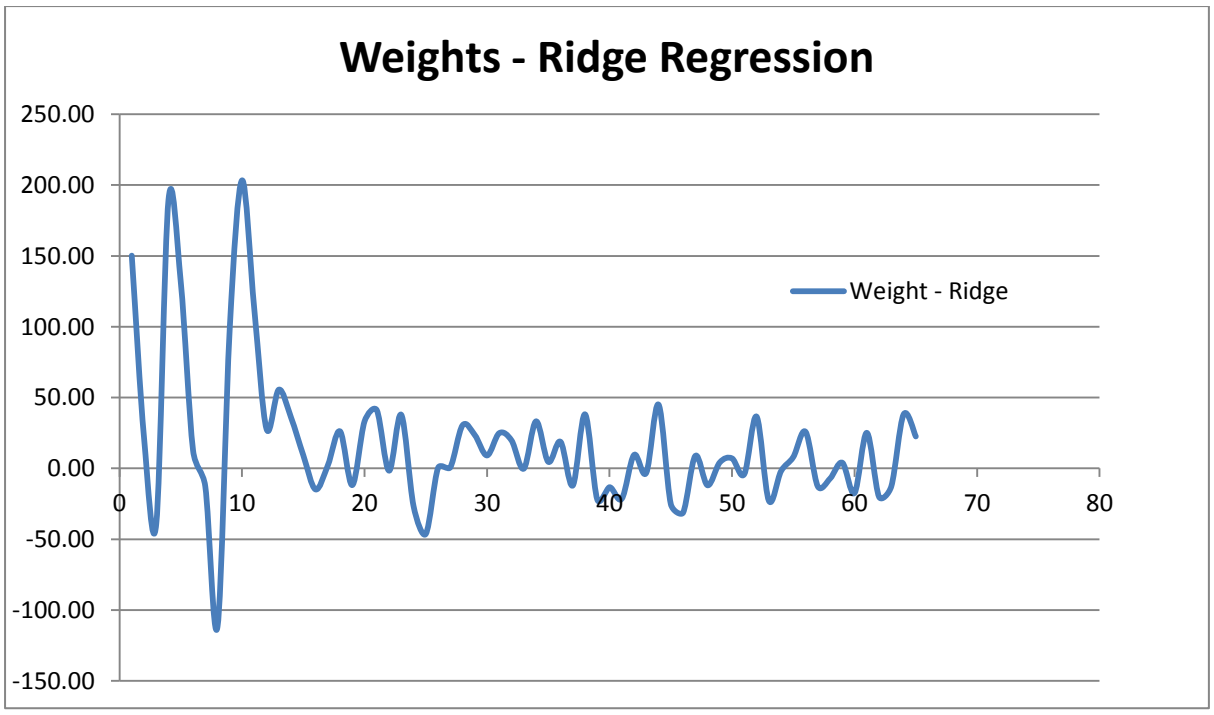


MSE for Test Data varying with Lambda

Comparison of relative magnitudes of weights from OLE and Ridge Regression

We have taken the learned weights (65 values each) from Problem 2 and Problem 3, and plotted the values separately to get a sense of the magnitude of values.





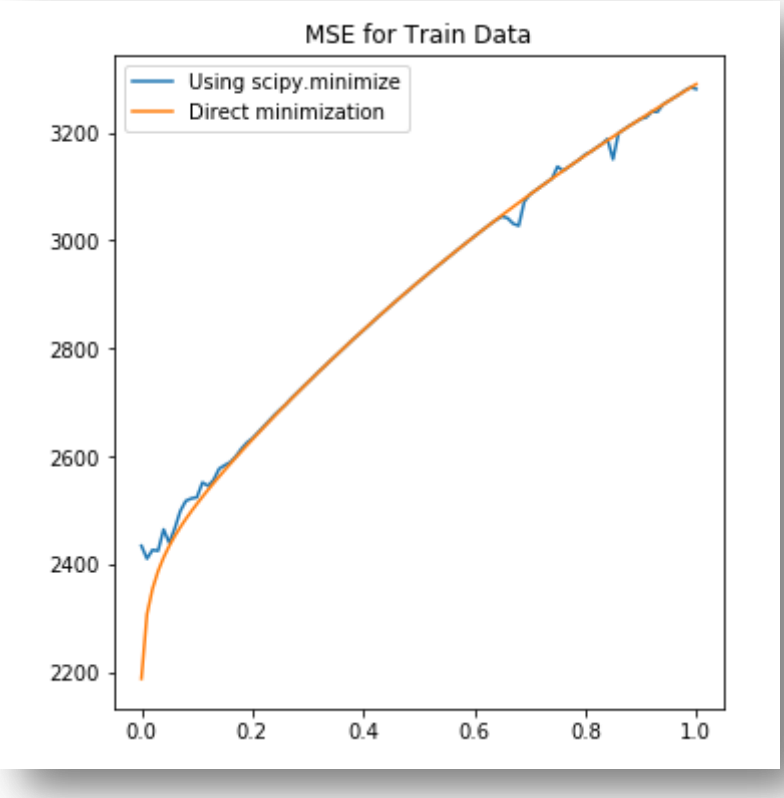
We can see above that the magnitude of weights from Linear Regression (Problem 2) is many orders of magnitude higher the weights obtained for Ridge Regression.

Comparison with Linear Regression

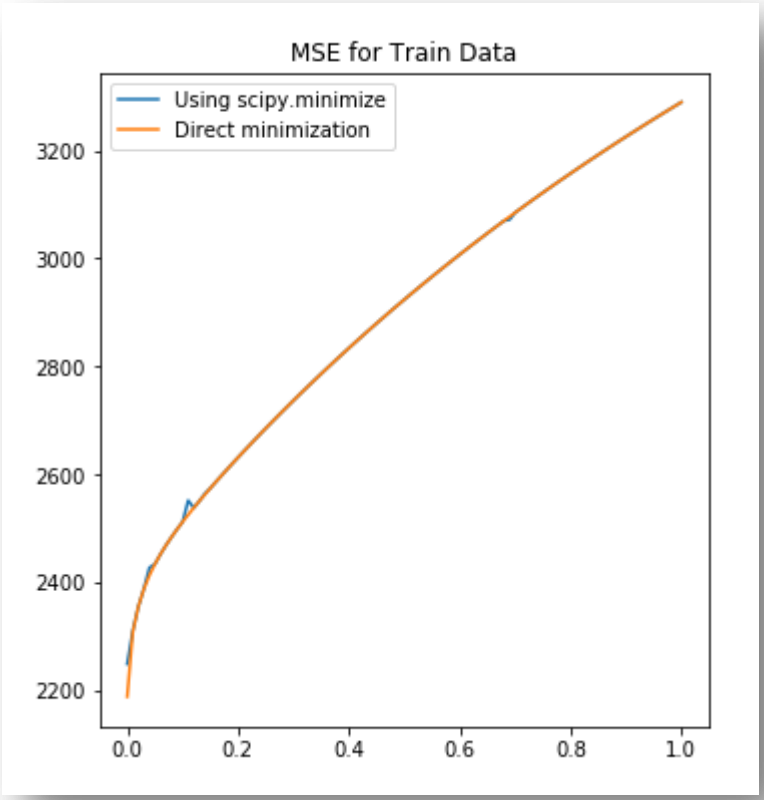
We can see the MSE on test data is much lower with Ridge Regression, comparing results with Linear Regression. In Ridge Regression, the optimal value of Lambda at 0.06 gives us MSE as 2851.33, whereas Linear Regression with intercept gives us a minimum test MSE of 3707.84, which is 30% higher than what we achieve with Ridge Regression. Another factor in favor of Ridge Regression is that the magnitude of weights is many orders lower than weights from Linear Regression, thus enabling faster computations. Overall, we believe Ridge Regression to be a better approach for estimating error, compared to Linear Regression.

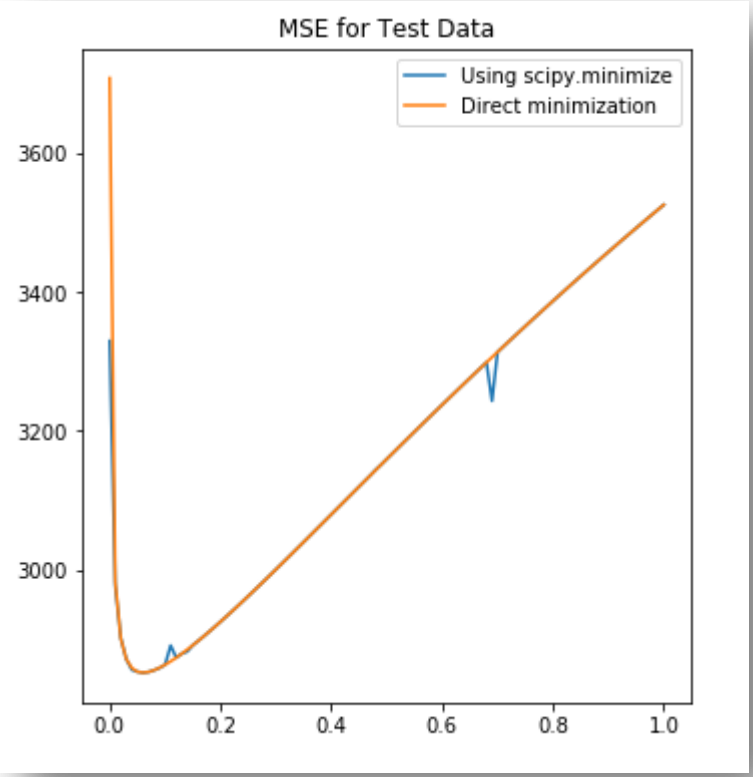
Problem 4: Using Gradient Descent for Ridge Regression Learning

For 20 Iterations, we have the following plots:



For 100 Iterations, the resultant plot for Gradient Descent is much smoother than for 20 iterations:

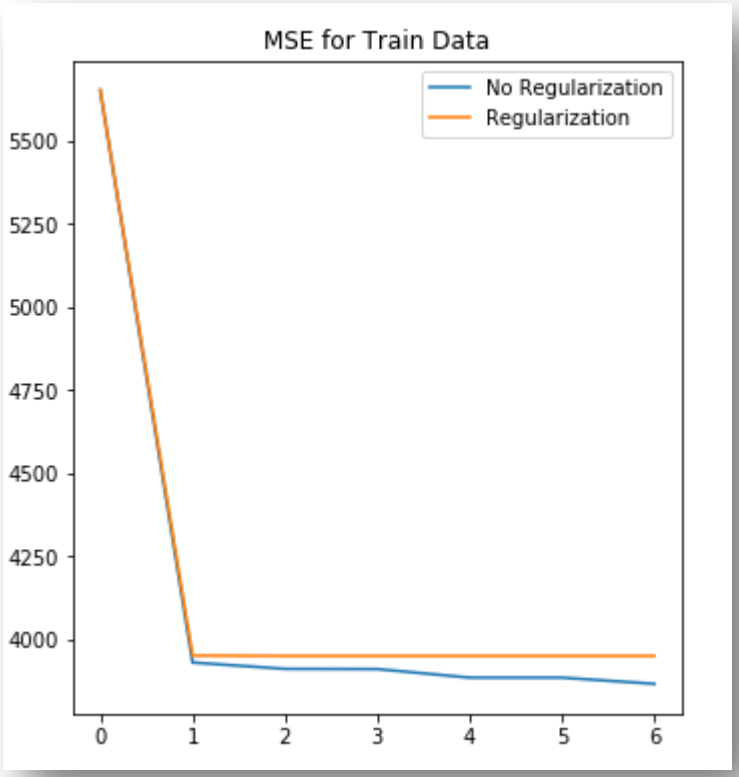


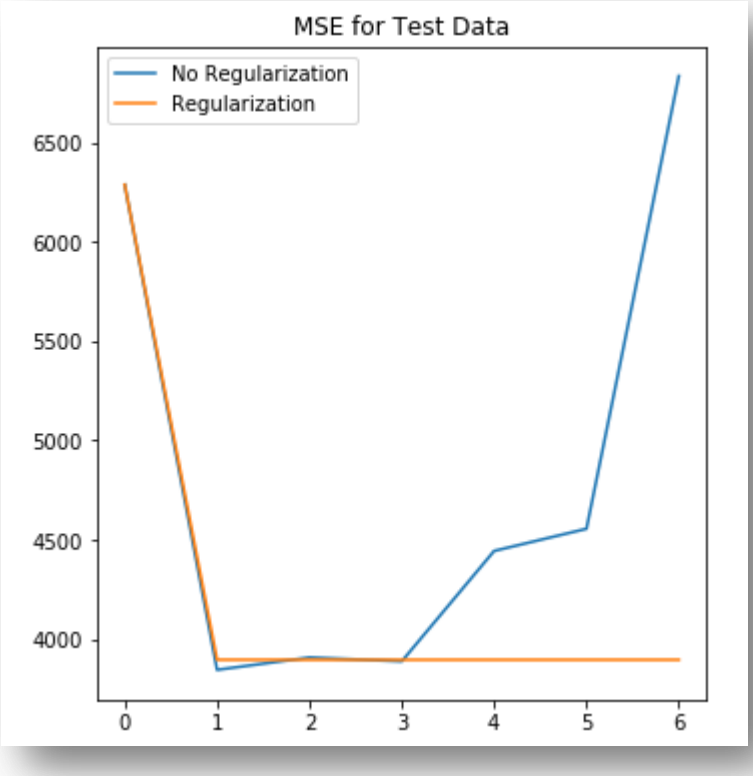


We see that when using Gradient Descent with Ridge Regression and not using Gradient Descent for Ridge Regression results in almost similar plots. However, when using Gradient Descent, we see that the line is not smooth in some parts, as can be seen from the spikes at the lower and higher values of lambda, but these are only few areas, and the rest of the lambda values have the two lines overlapping.

Problem 5: Non-linear Regression

With regularization (taking Lambda = 0.06) vs no regularization





Comparison of Error Values for Lambda = 0 and Lambda = Optimal (0.06 from Problem 3)

When Lambda = 0		
P	Train Error	Test Error
0	5650.71	6286.40
1	3930.92	3845.03
2	3911.84	3907.13
3	3911.19	3887.98
4	3885.47	4443.33
5	3885.41	4554.83
6	3866.88	6833.46

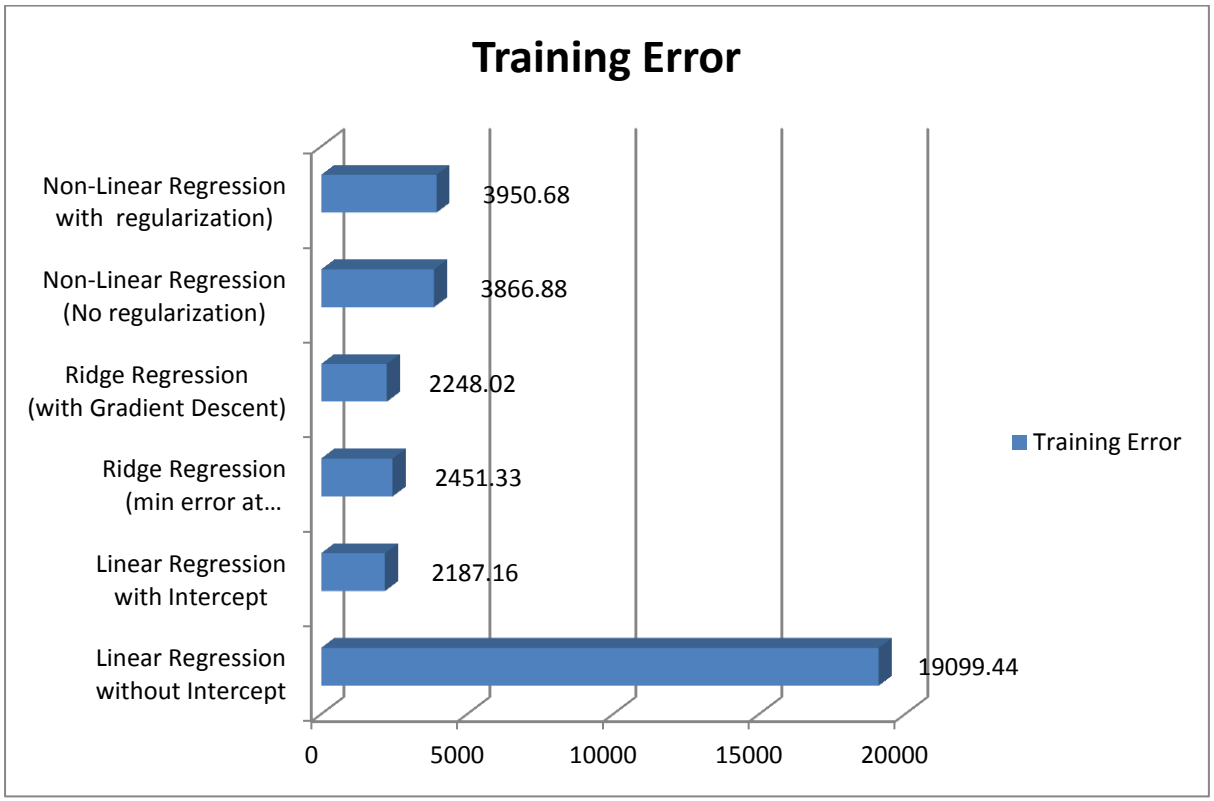
When Lambda = Optimal		
P	Train Error	Test Error
0	5650.71	6286.88
1	3951.84	3895.86
2	3950.69	3895.58
3	3950.68	3895.58
4	3950.68	3895.58
5	3950.68	3895.58
6	3950.68	3895.58

We see that with **no regularization**, the optimal value of **p is 1**, which results in the lowest test error. For **optimal Lambda**, all the values of **p >= 1** result in more or less the same minimum test error, and can be termed as constant for values of p greater than equal to 1.

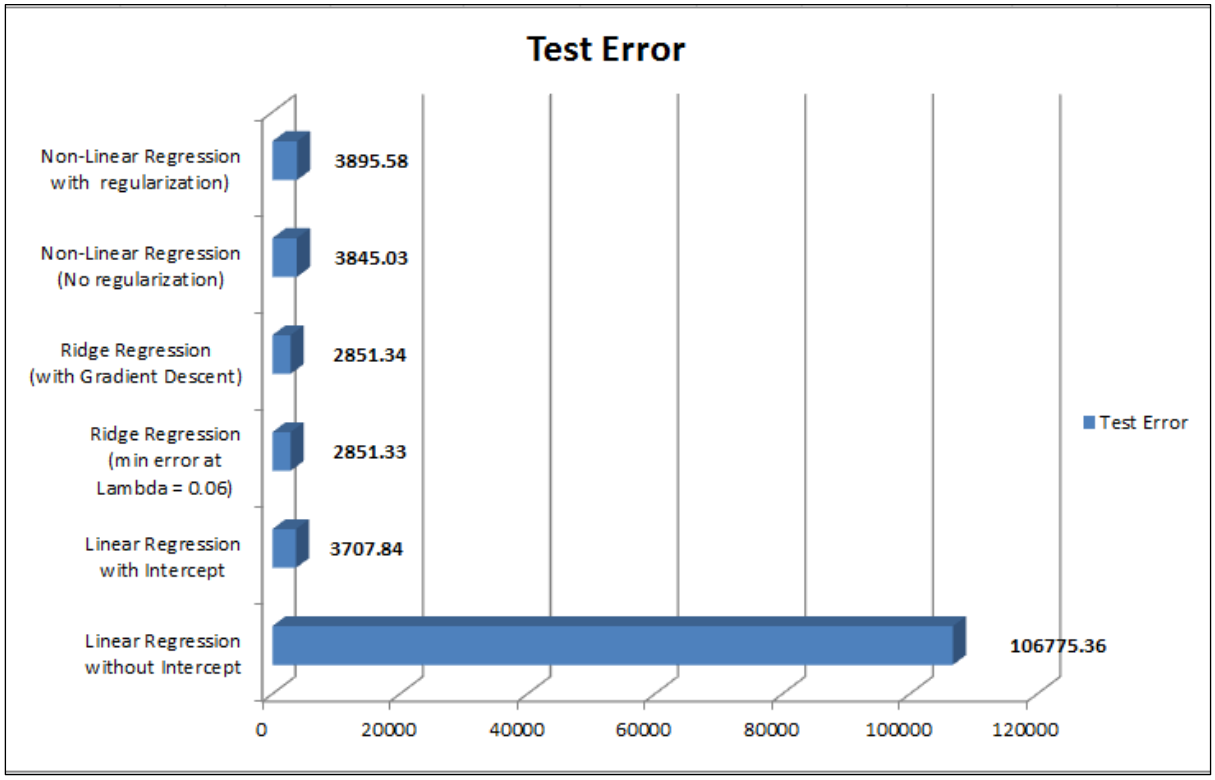
Problem 6: Interpreting Results

Comparison of results from the above methods, based on training and test error:

Method Used	Training Error
Linear Regression without Intercept	19099.44684
Linear Regression with Intercept	2187.160295
Ridge Regression (min error at Lambda = 0.06)	2451.33
Ridge Regression (with Gradient Descent)	2248.024022
Non-Linear Regression (No regularization)	3866.88
Non-Linear Regression with regularization)	3950.68



Method Used	Test Error
Linear Regression without Intercept	106775.3616
Linear Regression with Intercept	3707.840182
Ridge Regression (min error at Lambda = 0.06)	2851.33
Ridge Regression (with Gradient Descent)	2851.340758
Non-Linear Regression (No regularization)	3845.03
Non-Linear Regression with regularization)	3895.58



Comparing training and test errors, we see that on training data, Linear Regression with intercept has the lowest MSE, and Ridge Regression with Gradient Descent is a close second. However, real world results would be better represented using the test error, for which the results show a different picture.

As we can interpret from the results shown above, the least MSE on test data is achieved when using Ridge Regression, and the results are same whether using Gradient Descent or using normal Ridge Regression (at optimal Lambda). The other methods have considerably higher MSE when compared to Ridge Regression. Also **note that the results above with Ridge Regression are the ones which include the intercept.** Without including the intercept, the error would be many times higher. **Thus, the optimal metric for a similar data set would be MSE calculated through Ridge Regression, taken with the intercept.** Both Gradient Descent and Non-gradient descent Ridge Regression have almost similar results, and thus either can be used.