CSE474/574 Introduction to Machine Learning
Programming Assignment 2
Classification and Regression

Group 35

TARIQ SIDDIQUI **KAUSHIK RAMASUBRAMANIAN** **JUNAID SHAIKH**

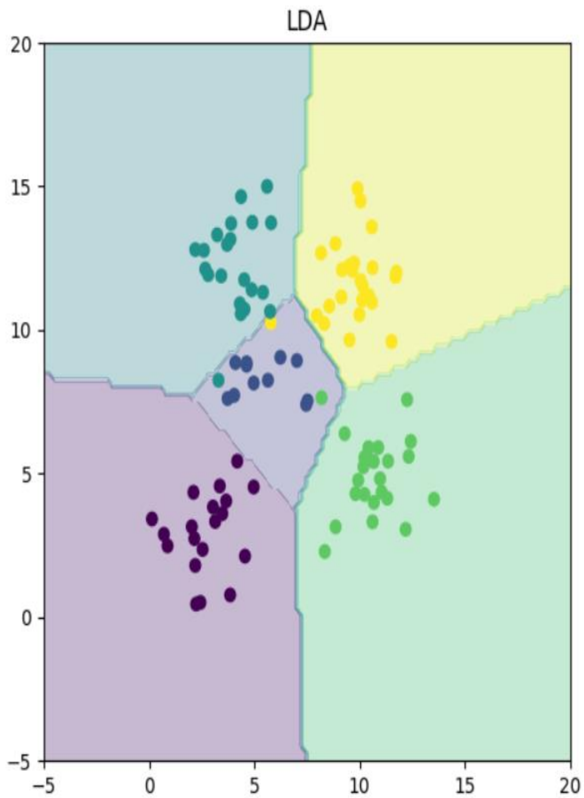
Problem 1: Experiment with Gaussian discriminators

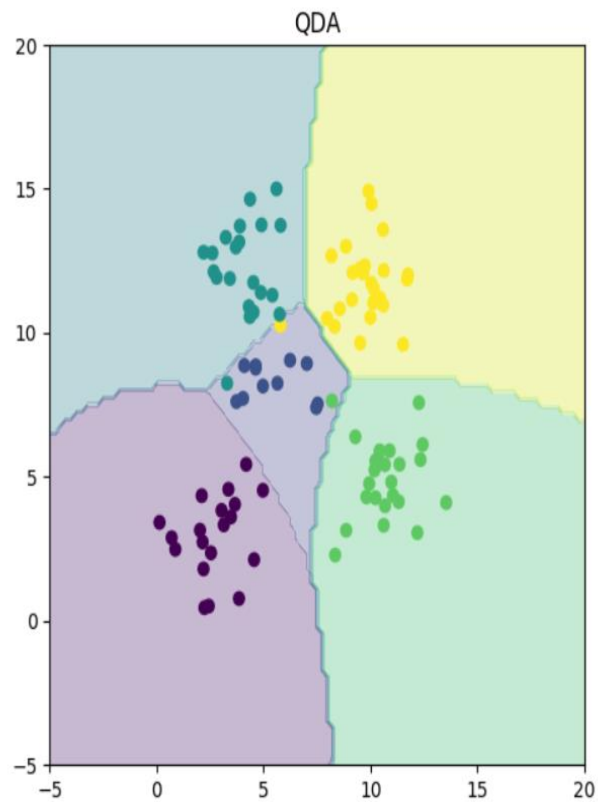
We got the following accuracies for LDA and QDA:

Accuracy for LDA – 97%

Accuracy for QDA – 96%

The corresponding plots for LDA and QDA are as shown below:





As seen from the plots above, the decision boundaries are straight lines in case of LDA and curves in case of QDA. As we know, for LDA we use the entire data set and do MLE for Sigma. That is, LDA uses the covariance of the complete data set provided, however QDA computes the covariance differently for each class wher**as LDA relaxes criteria of large amount of data.** Since the covariance is computed separately, QDA gives a non linear boundaries which give better classification when the data is complicated.

Observation1:

In case we are required to perform classification of complex data set, then QDA is preferred as it takes care of considering boundaries for varying data, using curves. LDA should be preferred if the results are to be obtained **faster**, but QDA gives **better** results.

Problem 2: Experiment with Linear Regression

The following Regression Mean Squared error (MSE) results were obtained for train and test data respectively:

MSE for train data:

1. MSE without intercept: `19099.44684457`
2. MSE with intercept: `2187.16029493`

MSE for test data:

1. MSE without intercept: `106775.36155355`
2. MSE with intercept: `3707.84018128`

From the above results we can see that the MSE without intercept is much greater than when using intercept. This is what is expected as a line without intercept will pass through the origin allowing it to only rotate. We can see that the error without an intercept for a test data is almost 30 times greater than that with an intercept. Since a trained model should give us less error, that is what we observe here that as the MSE for train data is much greater than test data for without intercept.

Observation2:

Adding Intercept(Bias term) : results in considerable drop in error for both test and training data

Observation3:

Error reduction on test data is quite higher when compared to training data

Problem 3: Experiment with Ridge Regression

Lambda	Train MSE	Test MSE
0	2187.16	3707.84
0.01	2306.832	2982.446
0.02	2354.071	2900.974
0.03	2386.78	2870.942
0.04	2412.119	2858
0.05	2433.174	2852.666
0.06	2451.528	2851.33
0.07	2468.078	2852.35
0.08	2483.366	2854.88
0.09	2497.74	2858.444
0.1	2511.432	2862.758
0.11	2524.6	2867.638
0.12	2537.355	2872.962
0.13	2549.777	2878.646
0.14	2561.925	2884.627
0.15	2573.841	2890.859
0.16	2585.56	2897.307
0.17	2597.105	2903.941
0.18	2608.496	2910.739
0.19	2619.748	2917.682
0.2	2630.873	2924.753
0.21	2641.879	2931.939
0.22	2652.774	2939.226
0.23	2663.564	2946.605
0.24	2674.254	2954.065
0.25	2684.848	2961.599
0.26	2695.349	2969.198
0.27	2705.76	2976.855
0.28	2716.083	2984.564
0.29	2726.32	2992.32
0.3	2736.473	3000.116
0.31	2746.543	3007.948
0.32	2756.533	3015.811
0.33	2766.442	3023.7
0.34	2776.273	3031.613
0.35	2786.027	3039.545
0.36	2795.704	3047.493
0.37	2805.305	3055.454

0.38	2814.831	3063.425
0.39	2824.284	3071.403
0.4	2833.664	3079.385
0.41	2842.972	3087.37
0.42	2852.208	3095.355
0.43	2861.374	3103.337
0.44	2870.471	3111.316
0.45	2879.498	3119.289
0.46	2888.458	3127.255
0.47	2897.35	3135.212
0.48	2906.176	3143.158
0.49	2914.935	3151.093
0.5	2923.63	3159.014
0.51	2932.26	3166.921
0.52	2940.827	3174.813
0.53	2949.331	3182.689
0.54	2957.773	3190.547
0.55	2966.153	3198.387
0.56	2974.473	3206.208
0.57	2982.732	3214.01
0.58	2990.932	3221.79
0.59	2999.074	3229.55
0.6	3007.157	3237.288
0.61	3015.183	3245.003
0.62	3023.153	3252.695
0.63	3031.066	3260.364
0.64	3038.924	3268.009
0.65	3046.728	3275.629
0.66	3054.477	3283.225
0.67	3062.173	3290.796
0.68	3069.816	3298.341
0.69	3077.406	3305.861
0.7	3084.945	3313.355
0.71	3092.433	3320.822
0.72	3099.871	3328.263
0.73	3107.259	3335.677
0.74	3114.597	3343.064
0.75	3121.886	3350.424
0.76	3129.128	3357.757
0.77	3136.322	3365.062

0.78	3143.468	3372.34
0.79	3150.568	3379.59
0.8	3157.622	3386.813
0.81	3164.63	3394.007
0.82	3171.593	3401.174
0.83	3178.512	3408.313
0.84	3185.387	3415.424
0.85	3192.218	3422.507
0.86	3199.006	3429.562
0.87	3205.751	3436.589
0.88	3212.454	3443.588
0.89	3219.115	3450.559
0.9	3225.735	3457.501
0.91	3232.315	3464.416
0.92	3238.854	3471.303
0.93	3245.353	3478.162
0.94	3251.812	3484.993
0.95	3258.232	3491.796
0.96	3264.614	3498.571
0.97	3270.957	3505.318
0.98	3277.263	3512.038
0.99	3283.53	3518.73
1	3289.761	3525.395

Table 1: MSE for Train and Test Data

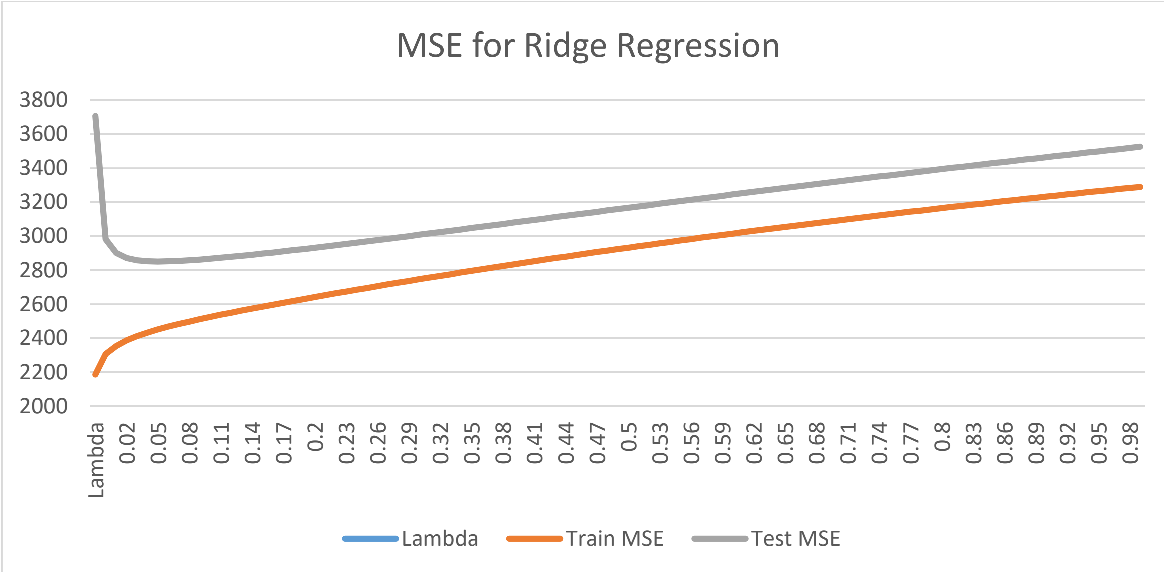


Figure 1.1 MSE For Ridge Regression

From the Table 1 above we can see that the **optimum value of lambda observed is 0.06**. It is observed that at this value of lambda , MSE value obtained is the lowest 2187.160295.

From the above figure it is evident that the MSE for test data is higher than that of the train data.

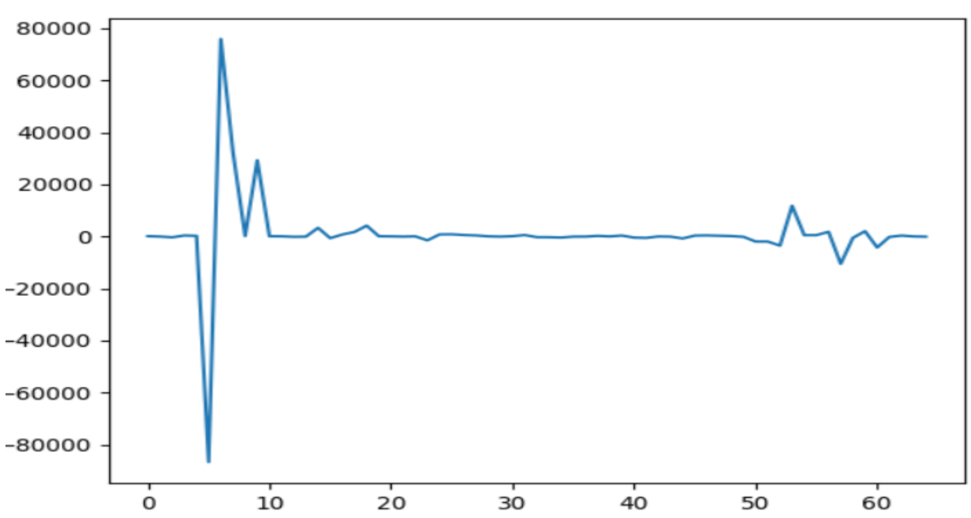


Fig 3.1 – Weights for OLE Regression with Intercept

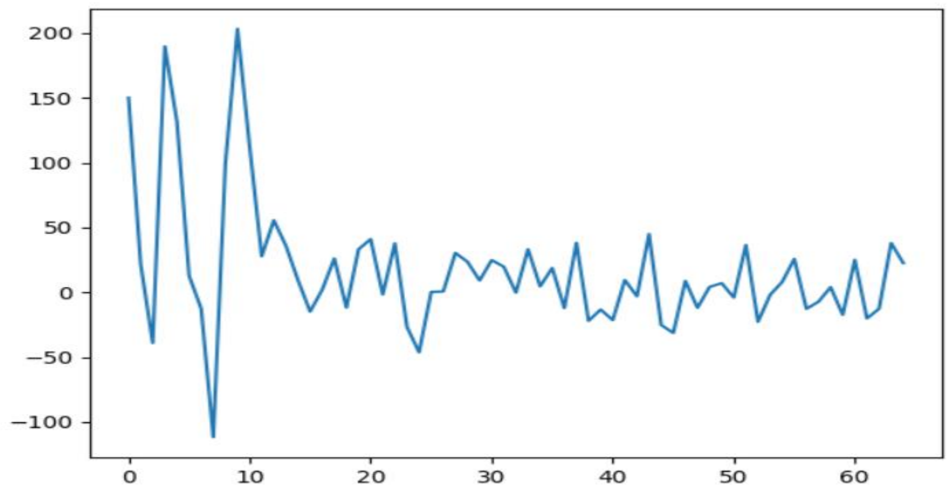


Fig 3.2 - Weights for Ridge Regression with Intercept

Looking at the Fig 3.1 and Fig 3.2 we can conclude that there is a huge gap between the weights learnt using OLE and Ridge Regression. OLE Regression has large values of weights in thousands however Ridge Regression has lesser when lambda is gradually increased from 0.

Observation4:

OLE Regression has large values of weights(magnitude) in thousands however Ridge Regression has lesser when lambda is gradually increased from 0. This is expected as due to randomization factor, Ridge preferences smaller weights

Problem 4: Using gradient descent for Ridge Regression learning

The following dataset represents mean squared error values for train and test data using various values of Lambda using Gradient descent approach.

Lambda	MSE (Train Data)	MSE(test data)
0	2433.664633	2900.546222
0.01	2396.442109	2861.287296
0.02	2415.244166	2826.952565
0.03	2457.264186	2849.186679
0.04	2431.807677	2846.754051
0.05	2467.326056	2842.919057
0.06	2459.069359	2837.61201

0.07	2482.84389	2851.198389
0.08	2512.75605	2856.413252
0.09	2517.630624	2845.55429
0.1	2531.709668	2864.540342
0.11	2541.946671	2865.011016
0.12	2548.353299	2871.165799
0.13	2557.073031	2881.239504
0.14	2573.580576	2891.40362
0.15	2580.470856	2894.285206
0.16	2592.3115	2904.760641
0.17	2601.826348	2900.650472
0.18	2612.184981	2908.228078
0.19	2621.594681	2920.415794
0.2	2634.914822	2925.509623
0.21	2643.744775	2933.663093
0.22	2655.204101	2938.881452
0.23	2664.683163	2946.687203
0.24	2675.891651	2954.033945
0.25	2686.061299	2961.536926
0.26	2696.793584	2969.517944
0.27	2706.660223	2977.130397
0.28	2716.202937	2984.866092
0.29	2726.982513	2992.103726
0.3	2736.331393	2998.209833
0.31	2746.193488	3006.462233
0.32	2757.332029	3015.810321
0.33	2766.899863	3023.814546
0.34	2776.179015	3031.510213
0.35	2786.331515	3039.36525
0.36	2795.921349	3047.564365
0.37	2805.131629	3055.219838
0.38	2815.05263	3063.527387
0.39	2824.131331	3070.33751
0.4	2833.325263	3079.059594
0.41	2842.16964	3086.191386
0.42	2851.628798	3094.740449
0.43	2861.374332	3103.115495
0.44	2870.508253	3111.138336
0.45	2878.875595	3118.449866
0.46	2888.560743	3127.168702
0.47	2897.325434	3135.22987
0.48	2906.238055	3143.090619
0.49	2915.0297	3151.155892

0.5	2923.669583	3159.030314
0.51	2932.085565	3166.87972
0.52	2940.587239	3174.54463
0.53	2949.375165	3182.687332
0.54	2957.689459	3190.387835
0.55	2966.158381	3198.363042
0.56	2974.474768	3206.210603
0.57	2982.733814	3214.011674
0.58	2990.930438	3221.78473
0.59	2999.126702	3229.574379
0.6	3007.139073	3237.29706
0.61	3015.337706	3245.311011
0.62	3023.178933	3252.654437
0.63	3031.081054	3260.342996
0.64	3038.332616	3267.03008
0.65	3044.324949	3273.084071
0.66	3041.330581	3281.621963
0.67	3031.027044	3284.811341
0.68	3027.107411	3282.717607
0.69	3071.950081	3242.898486
0.7	3084.944007	3313.350492
0.71	3092.432226	3320.818197
0.72	3099.876004	3328.255186
0.73	3107.28567	3335.672723
0.74	3114.602636	3343.068315
0.75	3136.868704	3372.830077
0.76	3129.054478	3357.776073
0.77	3136.298023	3365.067985
0.78	3143.425918	3372.304797
0.79	3150.571252	3379.63576
0.8	3159.42958	3387.873214
0.81	3164.61874	3393.966734
0.82	3171.993796	3401.689756
0.83	3178.698752	3408.462173
0.84	3188.134569	3423.456322
0.85	3150.503695	3387.956776
0.86	3199.356781	3429.972613
0.87	3205.97434	3436.64809
0.88	3212.933698	3443.918686
0.89	3218.877293	3450.056164
0.9	3225.375834	3457.161743
0.91	3228.207441	3472.57303
0.92	3238.854037	3471.276599

0.93	3238.348832	3471.208421
0.94	3251.905525	3485.120448
0.95	3258.263432	3491.836218
0.96	3264.615109	3498.088107
0.97	3270.968285	3505.311353
0.98	3278.399723	3513.510914
0.99	3283.699879	3518.783514
1	3280.783437	3517.191683

Table 2: RMSE for train and Test Data using Gradient descent approach

From the above table we can see it was found that for Lambda value of 0.06 has the lowest MSE for test data. Thus, this is our optimum value of Lambda.

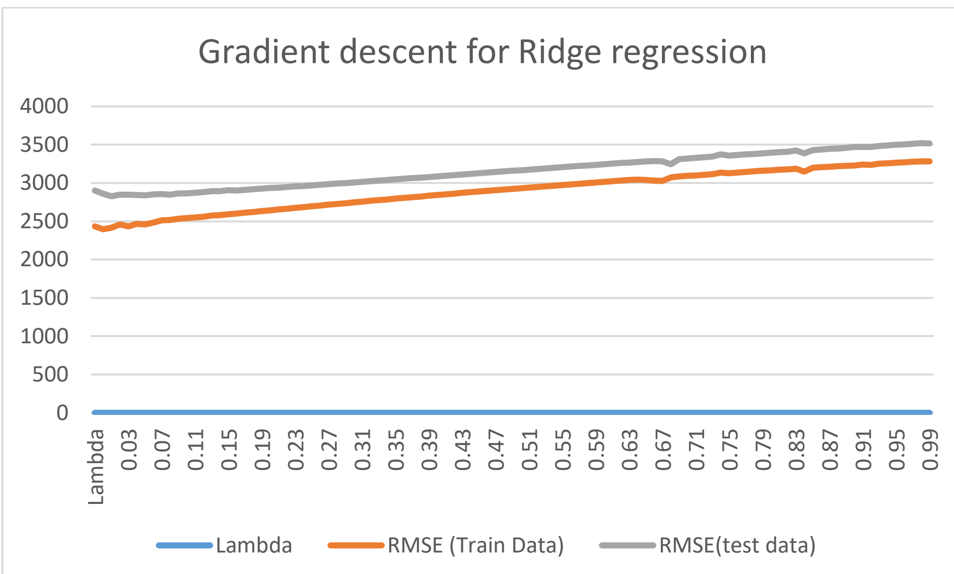


Figure 4 1 MSE vs Lambda using Gradient Descent

In the above figure we found that the MSE for test data is greater than train data. This is expected as the test data is more error prone.

When we compare the Fig 4.1 with Fig 1.1 we found that for lower values of Lambda the Ridge regression is more prone to errors in comparison to the gradient descent.

However, as the lambda value increases the MSE for both train and test data in both the cases of Ridge regression and gradient descent using Ridge Regression are almost the same.

Observation5:

Smoothness of output using Gradient Descent depends on maximum iterations over which GD is performed. Smoothness of curve increases with the iterations.

Problem 5: Non-linear regression

For Train Data –

From the below figure we get to know that the error decreases steeply from regularization of 0 to 1, and further reduces by a very less amount.

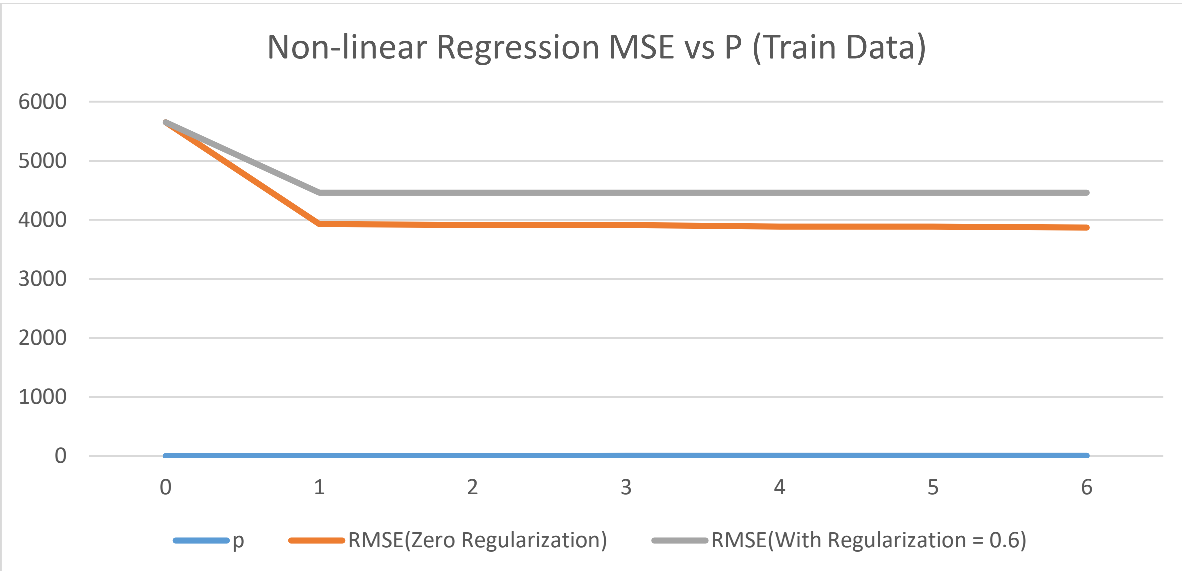


Figure 5 1 Non-linear Regression MSE vs P (Train Data)

Values for Train Data:

p	MSE(Zero Regularization)	MSE(With Regularization = 0.6)
0	5650.710539	5650.710539
1	3930.915407	4458.163111
2	3911.839671	4457.781931
3	3911.188665	4457.766894
4	3885.473068	4457.766813
5	3885.407157	4457.766702
6	3866.883449	4457.766691

For Test Data –

From the below figure we can see that for test data, the error is high initially but then eventually decreases. At higher p values the error again increases

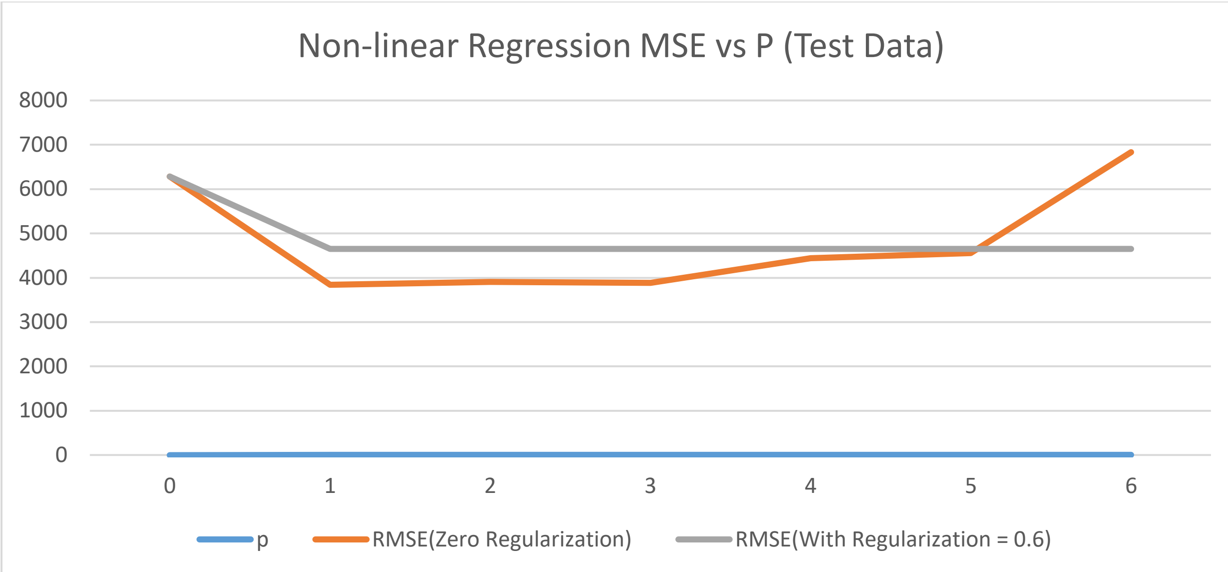


Figure 5-2 : Non-linear Regression RMSE vs P (Test Data)

Values for test data:

p	MSE(Zero Regularization)	MSE(With Regularization = 0.6)
0	6286.404792	6286.404792
1	3845.03473	4649.387802
2	3907.128099	4648.484397
3	3887.975538	4648.448478
4	4443.327892	4648.448209
5	4554.830377	4648.448324
6	6833.459149	4648.448316

From the training data we found that the optimum values of p only decrease beyond a certain point. That optimum value for their corresponding regularization as below:

For Regularization = 0, optimum value of p is 3.

For Regularization = 0.6, optimum value of p is 2.

Problem 6: Interpreting Results

Below is a comparison of the errors in the training and testing phase from the above 4 problems:

Type of Problem	Training MSE	Testing MSE
OLE with intercept	2187.16029493	3707.84018128

OLE without intercept	19099.44684457	106775.36155355
Ridge Regression	3707.840181	2187.160295
Gradient Descent	2861.287296	2396.442109
Non-linear Regression with regularization	4457.766691	4648.448209
Non-linear Regression without regularization	3885.407157	3845.03473

From the above table we observed that, the gradient descent and Ridge regression seems to provide very low errors on the test data. On the other hand linear regression without intercept gives very high MSE. Compared to this, linear regression with intercept gives very low MSE value.

Summary:

Observation1:

In case we are required to perform classification of complex data set, then QDA is preferred as it takes care of considering boundaries for varying data, using curves. LDA should be preferred if the results are to be obtained **faster**, but QDA gives **better** results.

Observation2:

Adding Intercept(Bias term) : results in considerable drop in error for both test and training data.

Observation3:

OLE Regression has large values of weights(magnitude) in thousands however Ridge Regression has lesser when lambda is gradually increased from 0. This is expected as due to randomization factor, Ridge preferences smaller weights

Observation4:

Smoothness of output using Gradient Descent depends on maximum iterations over which GD is performed. Smoothness of curve increases with the iterations.