CSE 574 Introduction to Machine Learning

Programming Assignment 1

Classification and Regression

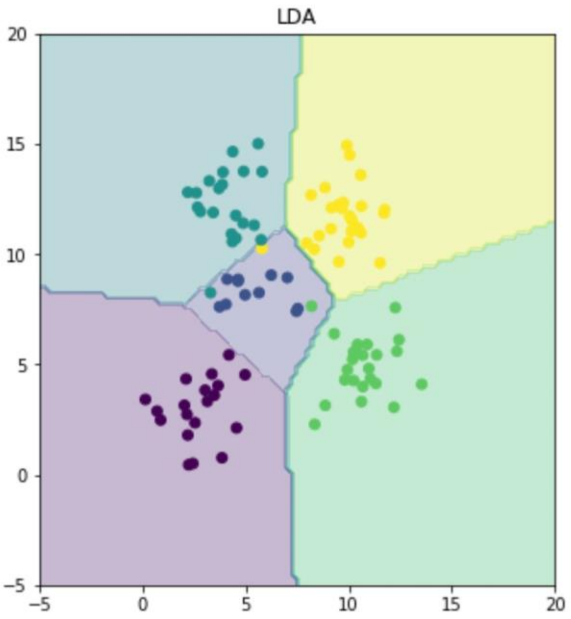
Group 150

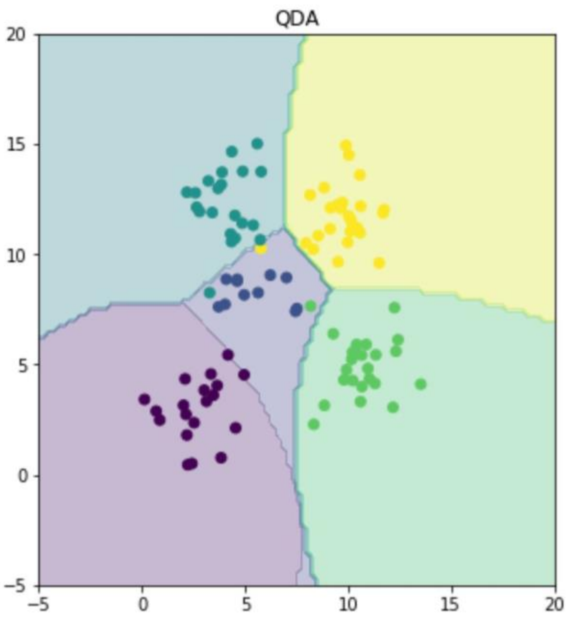
Ashutosh Ahmad Alexandar **Harish Ganesan**

Problem 1: Experiment with Gaussian Discriminators

Accuracy for LDA: 97

Accuracy for QDA: 96

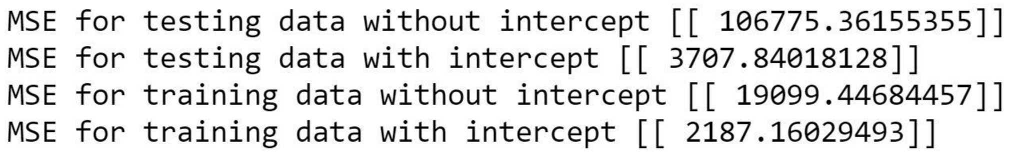




The boundaries for LDA and QDA are different because we can see that in LDA, all the decision boundaries are straight lines, this is because it’s linear. QDA has curved decision boundaries because it is quadratic in nature. This difference ultimately emerges due to LDA having only one covariance matrix for all 5 classes and QDA having a unique covariance matrix for each class present in the dataset.

Problem 2: Experiment with Linear Regression

The values of MSE for training and testing data with and without intercept are as follows -



```
MSE for testing data without intercept [[ 106775.36155355]]
MSE for testing data with intercept [[ 3707.84018128]]
MSE for training data without intercept [[ 19099.44684457]]
MSE for training data with intercept [[ 2187.16029493]]
```

As we can see from the above numbers, the Mean Squared Error is far greater without an intercept than with an intercept. This is because if we leave the intercepts in, the mean of the residuals will be 0. If we remove this intercept, it thus increases the error as the mean of the residuals is not minimized. The training data with intercept has much lesser than the testing data because it is learning from the training data itself and if we test the model with the training data itself, the error is bound to be lesser.

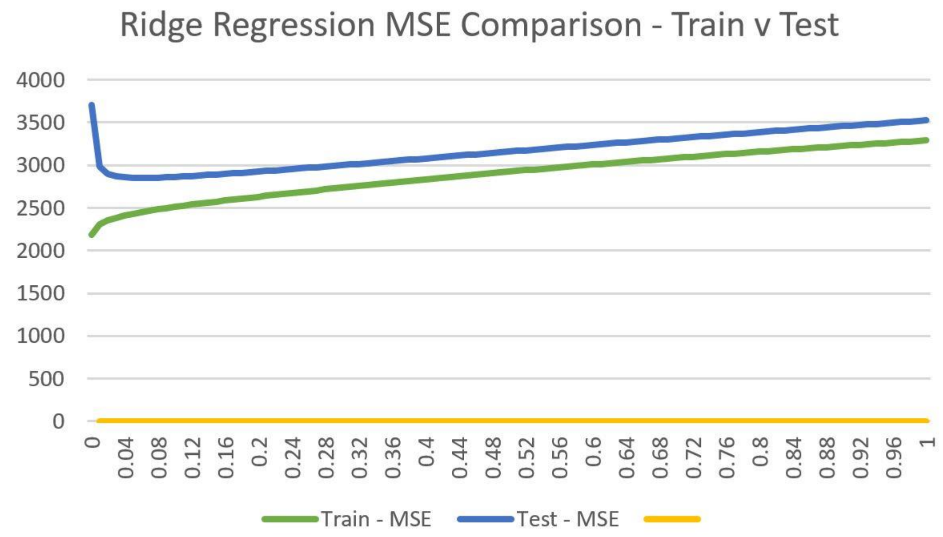
Problem 3: Experiment with Ridge Regression

Below we can find the numbers for the train and test MSE using ridge regression. We can see that the minimum value of MSE Test is at lambda = 0.06.

Lambda	Train - MSE	Test - MSE
0	2187.160295	3707.840181
0.01	2306.832218	2982.44612
0.02	2354.071344	2900.973587
0.03	2386.780163	2870.941589
0.04	2412.119043	2858.00041
0.05	2433.174437	2852.665735
0.06	**2451.528491**	**2851.330213**
0.07	2468.077553	2852.349994
0.08	2483.365647	2854.879739
0.09	2497.740259	2858.444421
0.1	2511.432282	2862.757941
0.11	2524.600039	2867.637909
0.12	2537.3549	2872.962283
0.13	2549.776887	2878.645869
0.14	2561.924528	2884.626914
0.15	2573.841288	2890.85911
0.16	2585.559875	2897.306659
0.17	2597.105192	2903.941126
0.18	2608.4964	2910.739372

0.19	2619.748386	2917.682164
0.2	2630.872823	2924.753222
0.21	2641.878946	2931.938544
0.22	2652.774126	2939.22593
0.23	2663.564301	2946.604624
0.24	2674.254297	2954.065056
0.25	2684.848078	2961.598643
0.26	2695.348935	2969.197637
0.27	2705.759629	2976.855001
0.28	2716.082507	2984.564321
0.29	2726.319587	2992.319722
0.3	2736.47263	3000.115809
0.31	2746.543191	3007.947616
0.32	2756.532665	3015.810555
0.33	2766.442316	3023.700386
0.34	2776.273307	3031.613181
0.35	2786.026719	3039.545297
0.36	2795.703568	3047.493351
0.37	2805.30482	3055.454198
0.38	2814.831398	3063.424913
0.39	2824.284191	3071.402772
0.4	2833.664063	3079.385238
0.41	2842.971855	3087.369947
0.42	2852.208389	3095.354694
0.43	2861.374474	3103.337424
0.44	2870.470905	3111.316218
0.45	2879.498467	3119.289287
0.46	2888.457936	3127.254961
0.47	2897.350077	3135.211679
0.48	2906.17565	3143.157988
0.49	2914.935407	3151.09253
0.5	2923.630092	3159.014036
0.51	2932.260444	3166.921324
0.52	2940.827193	3174.813291
0.53	2949.331065	3182.688908
0.54	2957.772777	3190.547215
0.55	2966.153041	3198.387318
0.56	2974.472563	3206.208382
0.57	2982.732039	3214.009633
0.58	2990.93216	3221.790346
0.59	2999.073611	3229.549851
0.6	3007.157067	3237.287523
0.61	3015.183199	3245.002781

0.62	3023.152668	3252.695087
0.63	3031.066127	3260.363943
0.64	3038.924224	3268.008886
0.65	3046.727598	3275.629488
0.66	3054.476879	3283.225355
0.67	3062.172691	3290.796124
0.68	3069.81565	3298.341459
0.69	3077.406362	3305.861052
0.7	3084.945428	3313.354623
0.71	3092.43344	3320.821913
0.72	3099.870981	3328.262686
0.73	3107.258627	3335.676731
0.74	3114.596946	3343.063853
0.75	3121.886499	3350.423878
0.76	3129.127838	3357.75665
0.77	3136.321508	3365.062031
0.78	3143.468045	3372.339896
0.79	3150.567979	3379.590137
0.8	3157.621831	3386.812661
0.81	3164.630117	3394.007386
0.82	3171.593342	3401.174246
0.83	3178.512005	3408.313184
0.84	3185.3866	3415.424154
0.85	3192.21761	3422.507124
0.86	3199.005514	3429.562069
0.87	3205.750782	3436.588973
0.88	3212.453878	3443.587832
0.89	3219.115258	3450.558648
0.9	3225.735372	3457.50143
0.91	3232.314665	3464.416198
0.92	3238.853573	3471.302975
0.93	3245.352525	3478.161794
0.94	3251.811947	3484.992692
0.95	3258.232255	3491.795713
0.96	3264.613861	3498.570906
0.97	3270.95717	3505.318324
0.98	3277.262582	3512.038029
0.99	3283.53049	3518.730082
1	3289.761281	3525.394553



Upon getting the weights in each of the two cases – OLER and Ridge, we calculate the magnitude of these two weight vectors using the following commands -

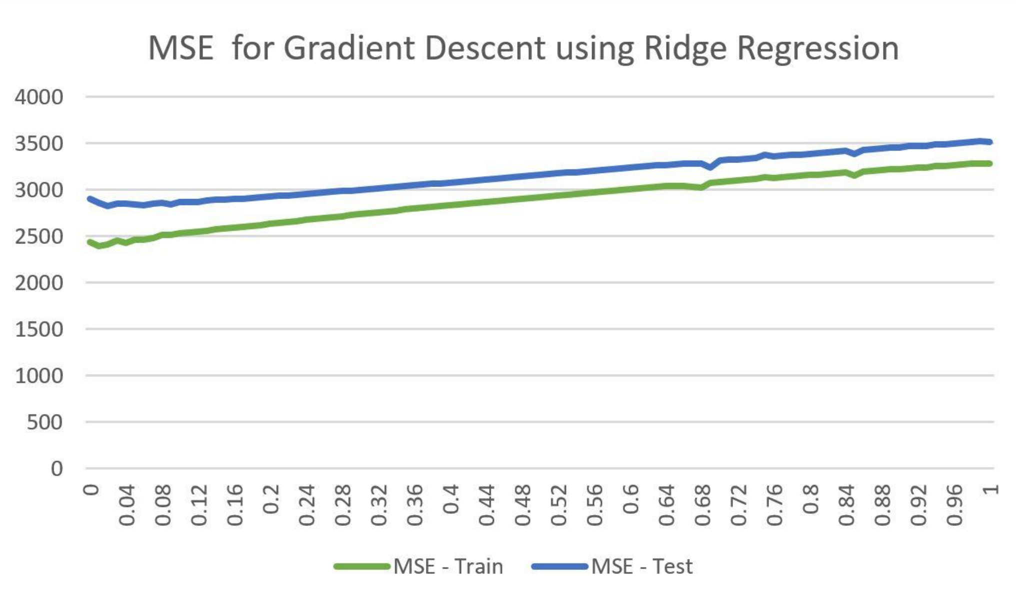
```
print(np.linalg.norm(w_l))
```

```
print(np.linalg.norm(w_i))
```

This gives us values of 124531.526527 for OLER and 959.312960893 for Ridge Regression. We can observe that the value is lesser in case of Ridge Regression and significantly greater for OLER. The optimal value of Lambda is 0.06 because this is the point where the value of the MSE is minimum.

Problem 4: Experiment with Ridge Regression

Here, we find that that the minimum value of MSE Test is 2826.953, and hence the optimal value of lambda is 0.02. Evidently, the MSE is greater for test data. The errors are much larger in the first few iterations of running (lesser values of lambda) the loop in Problem 3 as compared to Problem 4. In Problem 3, the error reaches a minimum point and then rises from there on. This is not the case for Problem 4, as the minimum is in the beginning itself, and from there the error keeps on steadily decreasing.

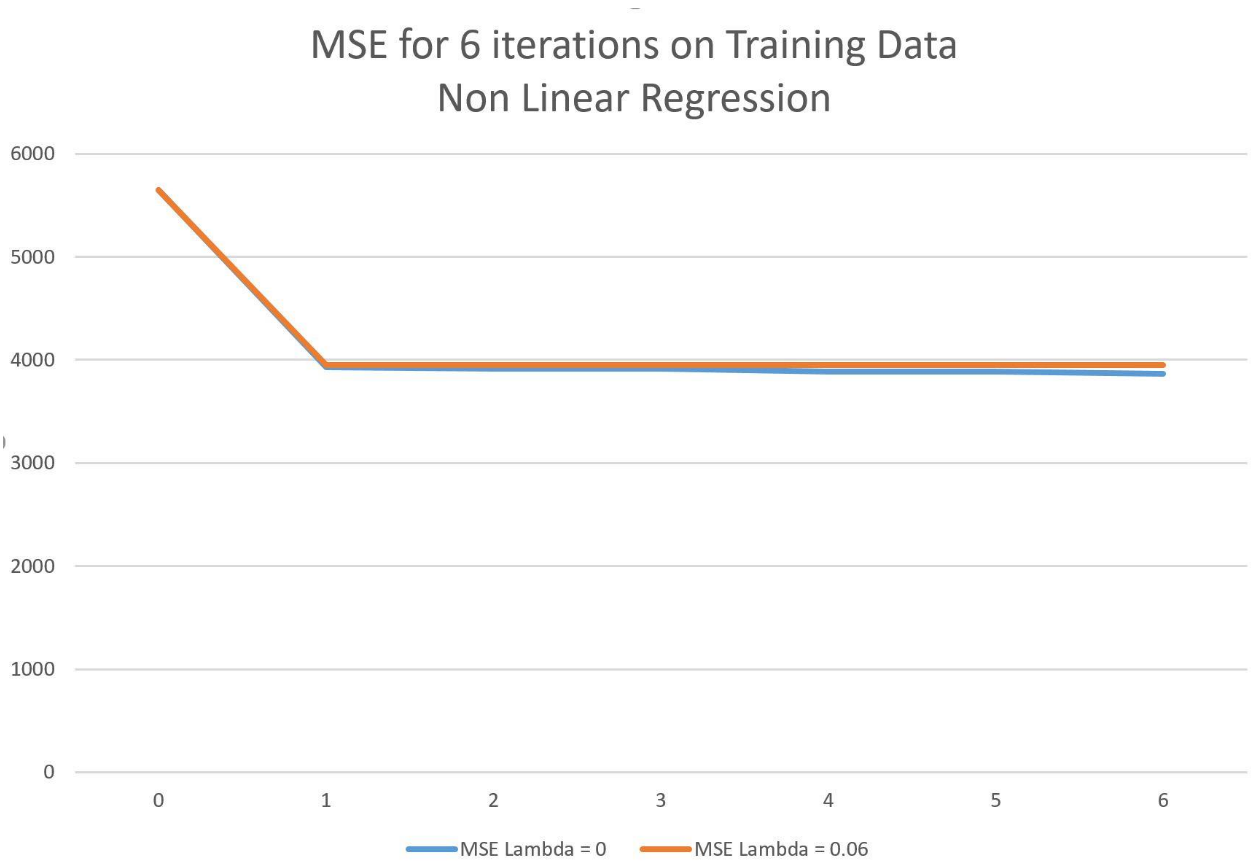


Lambda	MSE - Train	MSE - Test
0	2433.665	2900.546
0.01	2396.442	2861.287
0.02	2415.244	2826.953
0.03	2457.264	2849.187
0.04	2431.808	2846.754
0.05	2467.326	2842.919
0.06	2459.069	2837.612
0.07	2482.844	2851.198
0.08	2512.756	2856.413
0.09	2517.631	2845.554
0.1	2531.71	2864.54
0.11	2541.947	2865.011
0.12	2548.353	2871.166
0.13	2557.073	2881.24
0.14	2573.581	2891.404
0.15	2580.471	2894.285

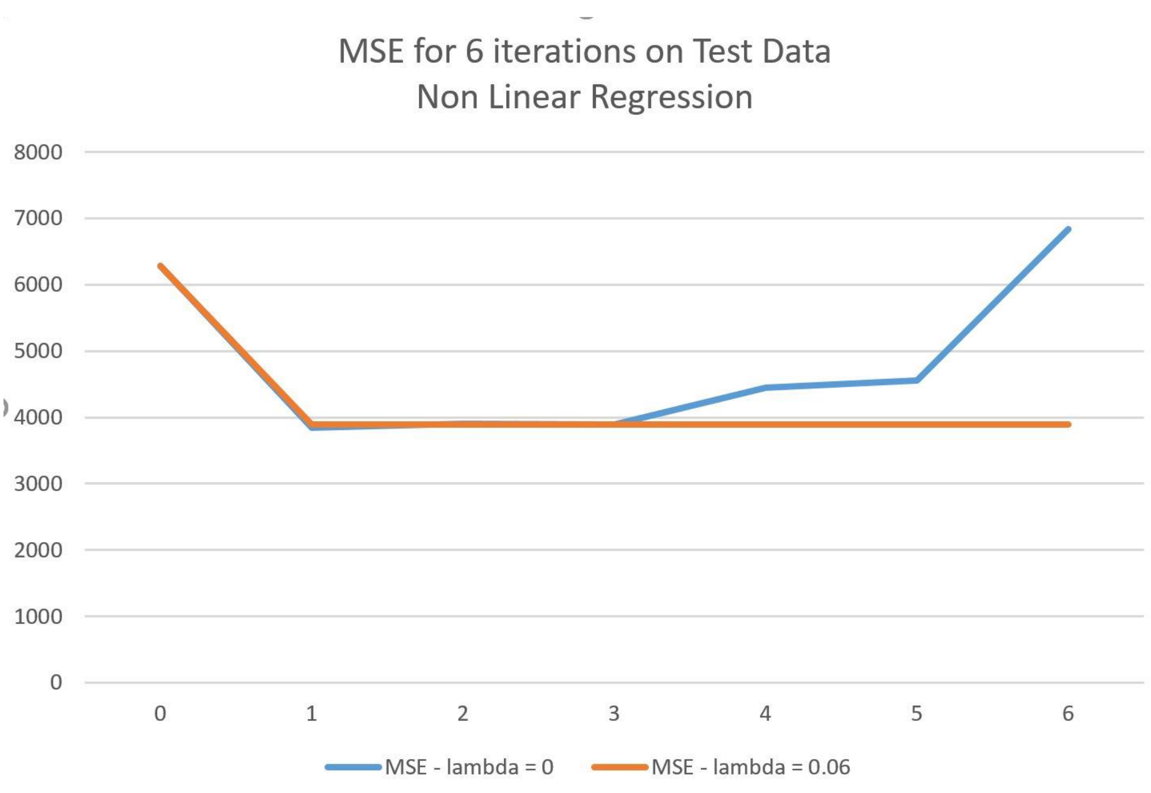
0.16	2592.311	2904.761
0.17	2601.826	2900.65
0.18	2612.185	2908.228
0.19	2621.595	2920.416
0.2	2634.915	2925.51
0.21	2643.745	2933.663
0.22	2655.204	2938.881
0.23	2664.683	2946.687
0.24	2675.892	2954.034
0.25	2686.061	2961.537
0.26	2696.794	2969.518
0.27	2706.66	2977.13
0.28	2716.203	2984.866
0.29	2726.983	2992.104
0.3	2736.331	2998.21
0.31	2746.193	3006.462
0.32	2757.332	3015.81
0.33	2766.9	3023.815
0.34	2776.179	3031.51
0.35	2786.332	3039.365
0.36	2795.921	3047.564
0.37	2805.132	3055.22
0.38	2815.053	3063.527
0.39	2824.131	3070.338
0.4	2833.325	3079.06
0.41	2842.17	3086.191
0.42	2851.629	3094.74
0.43	2861.374	3103.115
0.44	2870.508	3111.138
0.45	2878.876	3118.45
0.46	2888.561	3127.169
0.47	2897.325	3135.23
0.48	2906.238	3143.091
0.49	2915.03	3151.156
0.5	2923.67	3159.03
0.51	2932.086	3166.88
0.52	2940.587	3174.545
0.53	2949.375	3182.687
0.54	2957.689	3190.388
0.55	2966.158	3198.363
0.56	2974.475	3206.211
0.57	2982.734	3214.012
0.58	2990.93	3221.785

0.59	2999.127	3229.574
0.6	3007.139	3237.297
0.61	3015.338	3245.311
0.62	3023.179	3252.654
0.63	3031.081	3260.343
0.64	3038.333	3267.03
0.65	3044.325	3273.084
0.66	3041.331	3281.622
0.67	3031.027	3284.811
0.68	3027.107	3282.718
0.69	3071.95	3242.898
0.7	3084.944	3313.35
0.71	3092.432	3320.818
0.72	3099.876	3328.255
0.73	3107.286	3335.673
0.74	3114.603	3343.068
0.75	3136.869	3372.83
0.76	3129.054	3357.776
0.77	3136.298	3365.068
0.78	3143.426	3372.305
0.79	3150.571	3379.636
0.8	3159.43	3387.873
0.81	3164.619	3393.967
0.82	3171.994	3401.69
0.83	3178.699	3408.462
0.84	3188.135	3423.456
0.85	3150.504	3387.957
0.86	3199.357	3429.973
0.87	3205.974	3436.648
0.88	3212.934	3443.919
0.89	3218.877	3450.056
0.9	3225.376	3457.162
0.91	3228.207	3472.573
0.92	3238.854	3471.277
0.93	3238.349	3471.208
0.94	3251.906	3485.12
0.95	3258.263	3491.836
0.96	3264.615	3498.088
0.97	3270.968	3505.311
0.98	3278.4	3513.511
0.99	3283.7	3518.784
1	3280.783	3517.192

Problem 5: Non-Linear Regression



p	MSE Lambda = 0	MSE Lambda = 0.06
0	5650.710539	5650.711907
1	3930.915407	3951.839124
2	3911.839671	3950.687312
3	3911.188665	3950.682532
4	3885.473068	3950.682337
5	3885.407157	3950.682335
6	3866.883449	3950.682335



p	MSE - lambda = 0	MSE - lambda = 0.06
0	6286.404792	6286.881967
1	3845.03473	3895.856464
2	3907.128099	3895.584056
3	3887.975538	3895.582716
4	4443.327892	3895.582668
5	4554.830377	3895.582669
6	6833.459149	3895.582669

As we can see from the above two graphs and tables, the error in the test data is initially high, then reaches an optimal minimum, and then increases again for lambda = 0.

In case of lambda = 0.06, the error is high when p = 1, but then decreases further with greater values of p.

For Lambda = 0, the optimal p = 1, and for lambda = 0.06 the optimal p = 4.

Problem 6: Interpreting Results

Method	MSE- Training	MSE-Test
OLE with intercept	2187.16029493	3707.84018128
OLE without intercept	19099.44684457	106775.36155355
Ridge Regression	2451.528491	2851.330213
Gradient Descent	2415.244	2826.953
Non-linear (lambda = 0)	3866.883449	3950.682335
Non-linear(lambda = 0.06)	3845.03473	3895.582668

We must pick the best method depending on the value of the minimum of Mean Squared Error amongst all methods.

As we can see from the above table, the method that gives us the best performance on the test data (minimum error) is Ridge Regression. This is followed closely by Gradient Descent.