Machine Learning (CSE 574)
Classification and Regression
PA - 1
PROJECT GROUP - 36

Shubham Gulati
(sgulati3)

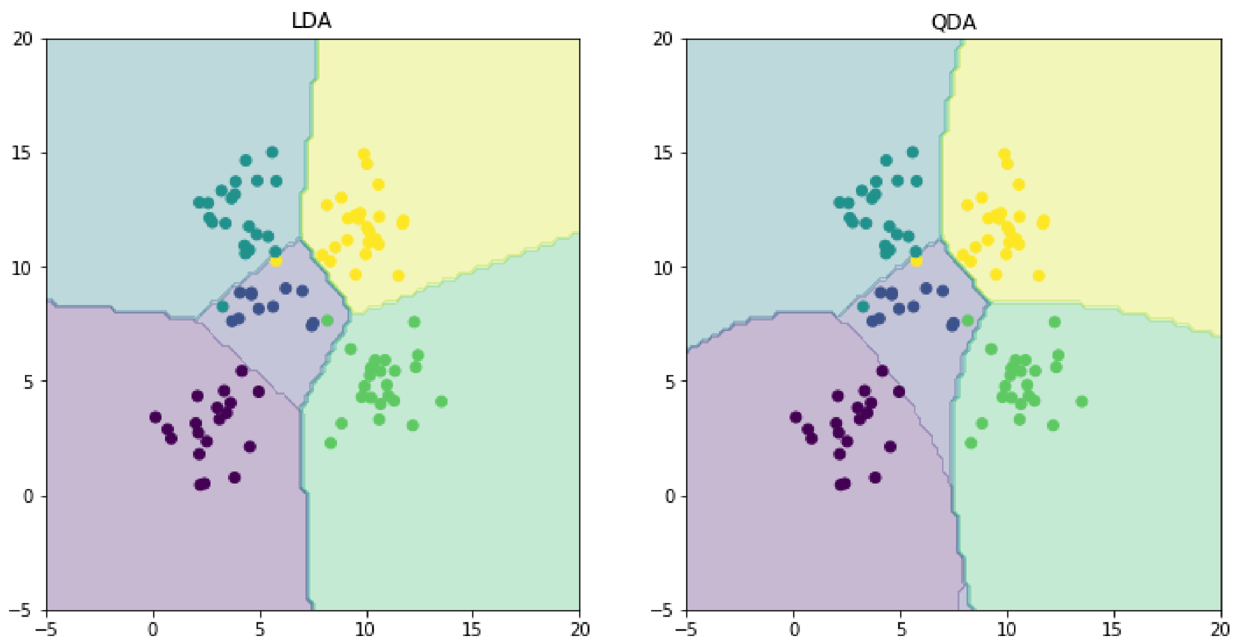
Mohammad Umair
(m39)

Yichen Wang
(yichenwa)

PROBLEM 1

Accuracy of ldaLearn = 97 %
Accuracy of qdaLearn = 97 %

LDA : Same covariance matrix is used for all the classes.
QDA : Different covariance matrix is used for each class.



The difference in boundaries comes from using same covariance matrix (Σ) in case of LDA and different covariance matrices in case of QDA. Because of this, the lines in case of QDA become quadratic in nature whereas they are linear in case of LDA. This quadratic nature induces a curve in the lines in case of QDA.

PROBLEM 2

Train Data:

$\text{MSE without intercept} = 19099.44684457$
$\text{MSE with intercept} \quad = 2187.16029493$

MSE with intercept, i.e. 2187, is better than MSE without intercept, i.e. 19099, by multiple orders of magnitude. This is because a line is able to fit better if it is not enforced to pass through the origin.

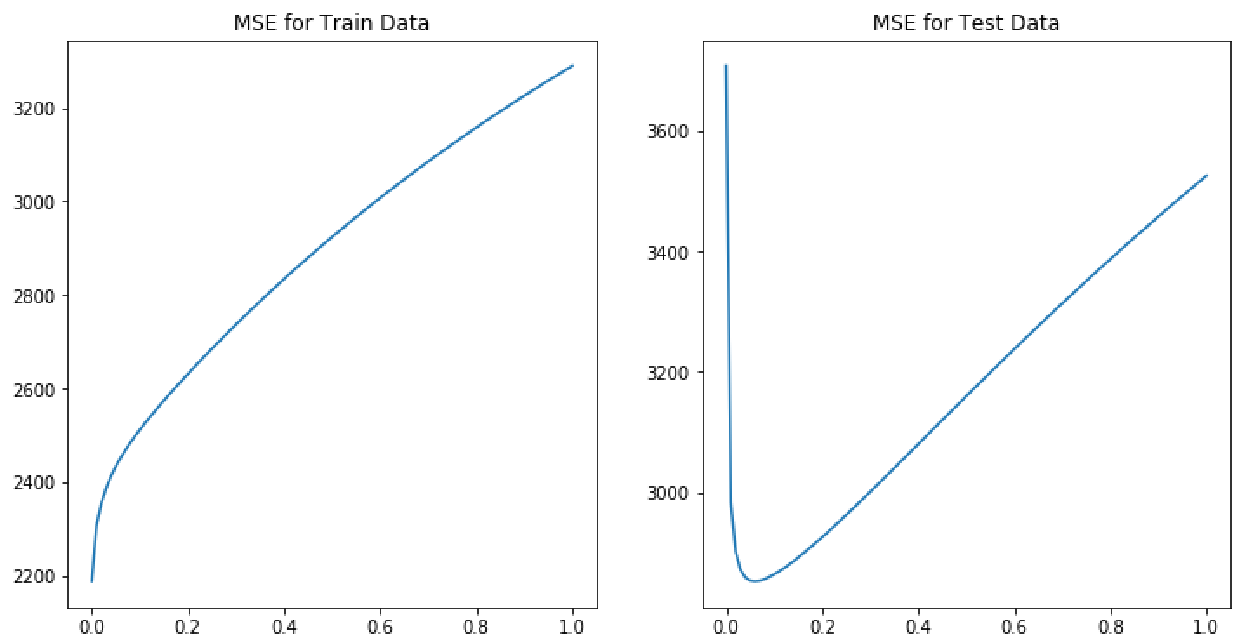
Test Data:

$\text{MSE without intercept} = 106775.36155952$
$\text{MSE with intercept} \quad = 3707.84018127$

MSE with intercept, i.e. 3707, is better than MSE without intercept, i.e. 106775, by multiple orders of magnitude.

MSE without intercept is way more worse for test data than it for training data because the prediction without an intercept is very poor. However, MSE with intercept in test data is comparable to MSE with intercept in train data.

PROBLEM 3



MSE for train data increases with the increase of lambda because increasing the value of lambda doesn't let the curve fit as closely as it could on the training data points. Setting lambda to 0 makes the curve to overfit the training data which gives us the least error on the training data but this curve (lambda = 0) performs very poor on the test data.

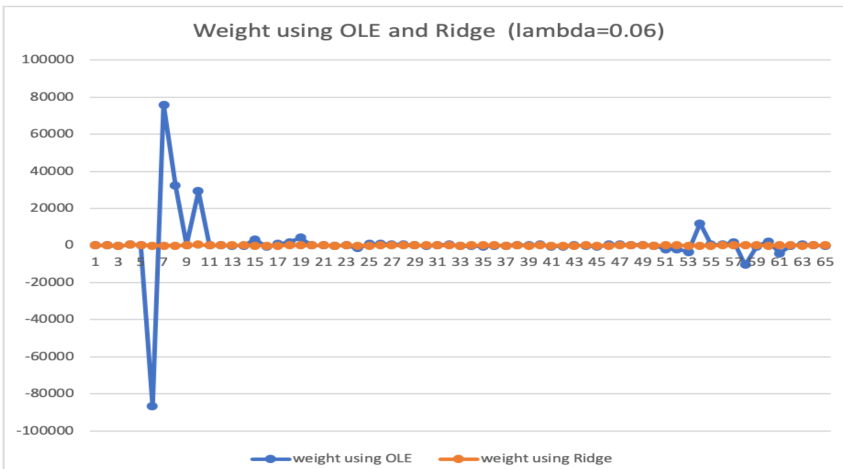
$\text{MSE for Linear Regression on train data} = 2187.16029493$
$\text{MSE for Linear Regression on test data} = 3707.84018127$
$\text{MSE for Ridge Regression on train data (optimal value of lambda)} = 2187.16029493$
$\text{MSE for Ridge Regression on test data (optimal value of lambda)} = 2851.33021344$

As expected, MSE for Linear Regression on train data is less than MSE for Ridge Regression on train data. This is because the regularisation term in Ridge Regression doesn't let the curve to overfit. Consequently, when the curve is not overfitting on train data, the MSE for Ridge Regression on test data is less than MSE for Linear Regression on test data.

Problem 3											
lambda	train	test	lambda	train	test	lambda	train	test	lambda	train	test
0	2187.16029	3707.84018	0.25	2684.84808	2961.59864	0.5	2923.63009	3159.01404	0.75	3121.8865	3350.42388
0.01	2306.83222	2982.44612	0.26	2695.34894	2969.19764	0.51	2932.26044	3166.92132	0.76	3129.12784	3357.75665
0.02	2354.07134	2900.97359	0.27	2705.75963	2976.855	0.52	2940.82719	3174.81329	0.77	3136.32151	3365.06203
0.03	2386.78016	2870.94159	0.28	2716.08251	2984.56432	0.53	2949.33106	3182.68891	0.78	3143.46804	3372.3399
0.04	2412.11904	2858.00041	0.29	2726.31959	2992.31972	0.54	2957.77278	3190.54722	0.79	3150.56798	3379.59014
0.05	2433.17444	2852.66574	0.3	2736.47263	3000.11581	0.55	2966.15304	3198.38732	0.8	3157.62183	3386.81266
0.06	2451.52849	2851.33021	0.31	2746.54319	3007.94762	0.56	2974.47256	3206.20838	0.81	3164.63012	3394.00739
0.07	2468.07755	2852.34999	0.32	2756.53266	3015.81055	0.57	2982.73204	3214.00963	0.82	3171.59334	3401.17425
0.08	2483.36565	2854.87974	0.33	2766.44232	3023.70039	0.58	2990.93216	3221.79035	0.83	3178.51201	3408.31318
0.09	2497.74026	2858.44442	0.34	2776.27331	3031.61318	0.59	2999.07361	3229.54985	0.84	3185.3866	3415.42415
0.1	2511.43228	2862.75794	0.35	2786.02672	3039.5453	0.6	3007.15707	3237.28752	0.85	3192.21761	3422.50712
0.11	2524.60004	2867.63791	0.36	2795.70357	3047.49335	0.61	3015.1832	3245.00278	0.86	3199.00551	3429.56207
0.12	2537.3549	2872.96228	0.37	2805.30482	3055.4542	0.62	3023.15267	3252.69509	0.87	3205.75078	3436.58897
0.13	2549.77689	2878.64587	0.38	2814.8314	3063.42491	0.63	3031.06613	3260.36394	0.88	3212.45388	3443.58783
0.14	2561.92453	2884.62691	0.39	2824.28419	3071.40277	0.64	3038.92422	3268.00889	0.89	3219.11526	3450.55865
0.15	2573.84129	2890.85911	0.4	2833.66406	3079.38524	0.65	3046.7276	3275.62949	0.9	3225.73537	3457.50143
0.16	2585.55987	2897.30666	0.41	2842.97185	3087.36995	0.66	3054.47688	3283.22536	0.91	3232.31467	3464.4162
0.17	2597.10519	2903.94113	0.42	2852.20839	3095.35469	0.67	3062.17269	3290.79612	0.92	3238.85357	3471.30298
0.18	2608.4964	2910.73937	0.43	2861.37447	3103.33742	0.68	3069.81565	3298.34146	0.93	3245.35253	3478.16179
0.19	2619.74839	2917.68216	0.44	2870.4709	3111.31622	0.69	3077.40636	3305.86105	0.94	3251.81195	3484.99269
0.2	2630.87282	2924.75322	0.45	2879.49847	3119.28929	0.7	3084.94543	3313.35462	0.95	3258.23225	3491.79571
0.21	2641.87895	2931.93854	0.46	2888.45794	3127.25496	0.71	3092.43344	3320.82191	0.96	3264.61386	3498.57091
0.22	2652.77413	2939.22593	0.47	2897.35008	3135.21168	0.72	3099.87098	3328.26269	0.97	3270.95717	3505.31832
0.23	2663.5643	2946.60462	0.48	2906.17565	3143.15799	0.73	3107.25863	3335.67673	0.98	3277.26258	3512.03803
0.24	2674.2543	2954.06506	0.49	2914.93541	3151.09253	0.74	3114.59695	3343.06385	0.99	3283.53049	3518.73008
									1	3289.76128	3525.39455

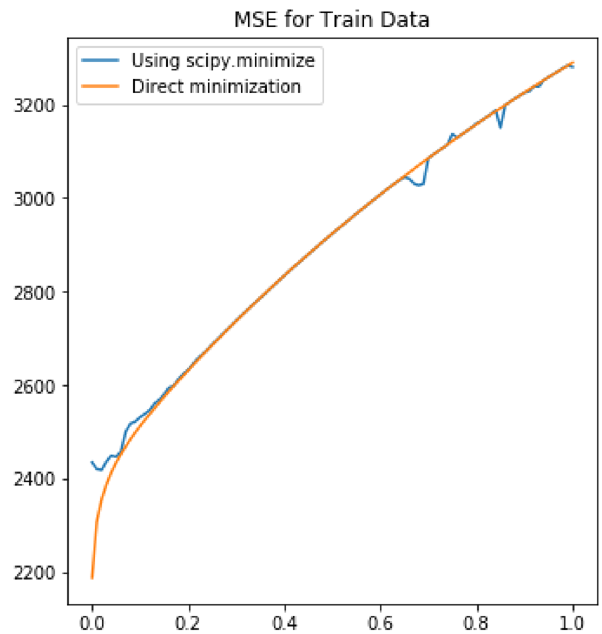
As can be seen in the above table, we get the least value of MSE for test data when lambda equals 0.06. Therefore, this is our optimal value of lambda.

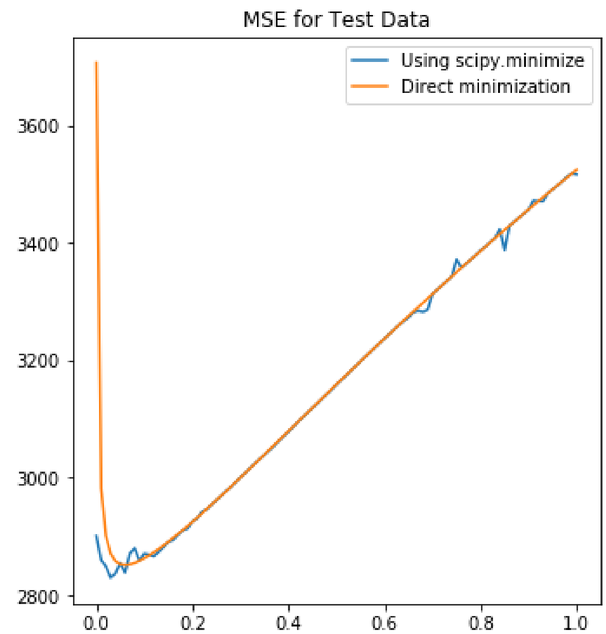
Serial Number	weight using OLE	weight using Ridge	Serial Number	weight using OLE	weight using Ridge	Serial Number	weight using OLE	weight using Ridge
1	148.154876	150.4595981	23	82.24245937	71.67974829	45	-729.6435311	-89.33525423
2	1.274852212	4.807768988	24	-1456.662084	-69.30906366	46	377.4083369	-22.73053674
3	-293.3835224	-202.9061147	25	827.3867027	-124.0343729	47	439.7942904	65.41116624
4	414.7254483	421.7194576	26	869.2909523	102.639818	48	308.5143733	55.11621318
5	272.0891343	279.4510729	27	586.2344952	72.64220588	49	189.8596788	19.14925041
6	-86639.45704	-52.29708233	28	427.0267266	79.24754013	50	-109.773797	-59.84315841
7	75914.46795	-128.5941891	29	90.246769	38.48319215	51	-1919.65699	26.64350735
8	32341.62279	-167.5005703	30	-17.88762242	32.98009446	52	-1924.633779	108.4050128
9	221.1012142	145.740681	31	141.6967738	92.09539122	53	-3489.795277	-137.6175697
10	29299.55117	496.3060412	32	582.8193844	68.97936154	54	11796.96872	-83.04383566
11	125.2303603	129.9484578	33	-234.0375107	-24.41700914	55	530.6744148	-20.40214777
12	94.41108333	88.30438076	34	-256.0714523	101.8538797	56	543.3059057	24.9726362
13	-93.86286322	11.29067689	35	-385.1774005	1.391226691	57	1821.07518	-0.924510929
14	-33.72827997	1.885325306	36	-33.4176738	20.85757155	58	-10463.98068	191.9130658
15	3353.197721	-2.583641569	37	-10.73500661	-29.65490134	59	-516.6276109	34.78309393
16	-621.0962997	-66.89445481	38	257.1071888	130.4111599	60	2064.359174	-43.90393505
17	791.7365341	-20.61939955	39	59.95545922	-16.75108796	61	-4199.413348	23.2002376
18	1767.760389	113.3930145	40	383.7280423	87.51340344	62	-140.4957052	20.8504118
19	4191.674054	17.99086827	41	-404.1583898	-45.64238362	63	374.15709	-117.853228
20	119.4381209	52.50235963	42	-514.2864343	-30.92288499	64	51.47574916	75.30611309
21	76.61034004	109.6876551	43	38.36366416	-10.07139781	65	-46.44927304	60.36839226
22	-15.2001293	-10.72779629	44	-44.6102889	31.13334896			



It is visible from the above graph that weights vary too much in case of Linear Regression. On the other hand, in case of Ridge Regression, weights do not vary as much because of the regularisation term which puts a high penalty on large values of w.

PROBLEM 4





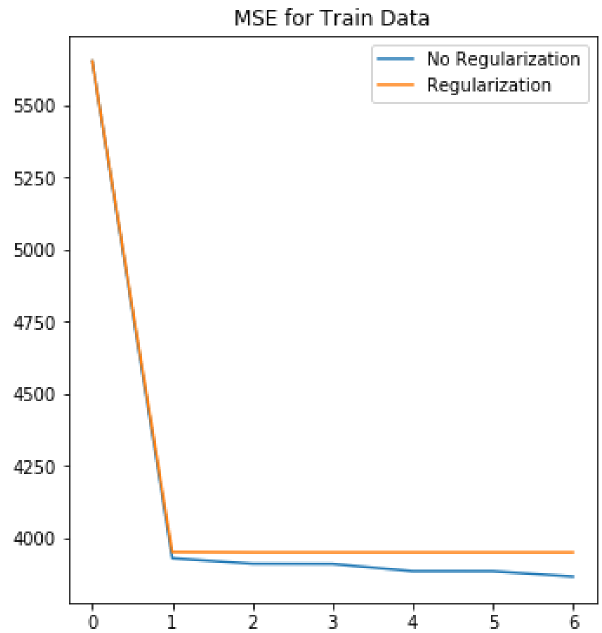
Problem 4											
lambda	train	test	lambda	train	test	lambda	train	test	lambda	train	test
0	2430.40177	2868.51463	0.25	2684.80373	2958.72159	0.5	2923.65124	3158.98885	0.75	3121.88648	3350.39548
0.01	2463.15035	2872.5033	0.26	2696.17138	2969.36465	0.51	2932.39292	3166.94223	0.76	3129.14677	3357.77193
0.02	2415.86184	2859.49914	0.27	2706.64311	2977.71768	0.52	2940.81101	3174.76339	0.77	3136.32715	3365.0716
0.03	2436.47613	2854.8753	0.28	2721.12528	2997.06367	0.53	2949.28775	3182.65811	0.78	3143.46756	3372.33719
0.04	2454.52606	2854.84104	0.29	2726.21432	2992.04625	0.54	2957.76863	3190.54642	0.79	3150.56797	3379.59025
0.05	2484.64114	2859.30896	0.3	2736.27152	3000.03443	0.55	2966.17264	3198.3806	0.8	3157.63152	3386.81288
0.06	2472.52284	2843.55761	0.31	2747.22289	3007.76021	0.56	2974.45136	3206.20686	0.81	3164.63011	3394.0073
0.07	2483.67033	2842.20233	0.32	2757.16794	3015.87678	0.57	2982.70926	3214.00461	0.82	3171.59333	3401.17425
0.08	2490.78417	2836.56773	0.33	2766.57534	3023.457	0.58	2990.9302	3221.80016	0.83	3178.51138	3408.31289
0.09	2520.05464	2860.47412	0.34	2776.52489	3031.24998	0.59	2999.04636	3229.54306	0.84	3185.38325	3415.42465
0.1	2510.63944	2858.28421	0.35	2786.01497	3039.01002	0.6	3007.16593	3237.27468	0.85	3192.21761	3422.50714
0.11	2525.63839	2866.23966	0.36	2795.80421	3047.59666	0.61	3015.16051	3244.97451	0.86	3199.00555	3429.56205
0.12	2538.23029	2871.76385	0.37	2806.15322	3055.85866	0.62	3023.16269	3252.69517	0.87	3205.75078	3436.58891
0.13	2552.39871	2878.91847	0.38	2814.93175	3063.48626	0.63	3031.06492	3260.3637	0.88	3212.4517	3443.58743
0.14	2563.49309	2884.37369	0.39	2824.4746	3071.02923	0.64	3038.92327	3268.01267	0.89	3219.11603	3450.55699
0.15	2575.39936	2890.61721	0.4	2833.86909	3079.20153	0.65	3046.7276	3275.63028	0.9	3225.73518	3457.5009
0.16	2586.43916	2895.65114	0.41	2842.95864	3087.39328	0.66	3054.47829	3283.22205	0.91	3232.31505	3464.41674
0.17	2597.86132	2902.92972	0.42	2852.1697	3095.29572	0.67	3062.17269	3290.79665	0.92	3238.85608	3471.30722
0.18	2607.65386	2908.53679	0.43	2861.02175	3102.10478	0.68	3069.81567	3298.34165	0.93	3245.35262	3478.16181
0.19	2622.62973	2917.5332	0.44	2870.55316	3109.40816	0.69	3077.43978	3305.86164	0.94	3251.81236	3484.9929
0.2	2631.01444	2924.0056	0.45	2879.26989	3119.09473	0.7	3084.95344	3313.36899	0.95	3258.23193	3491.7957
0.21	2642.67178	2932.4031	0.46	2888.14986	3126.82533	0.71	3092.44023	3320.82422	0.96	3264.61363	3498.57067
0.22	2653.65937	2938.15654	0.47	2897.3828	3135.188	0.72	3099.87098	3328.26287	0.97	3270.96191	3505.32284
0.23	2665.17679	2946.95942	0.48	2906.06235	3142.8537	0.73	3107.25785	3335.67129	0.98	3277.26258	3512.03806
0.24	2674.30358	2954.70945	0.49	2914.7059	3150.81502	0.74	3114.59695	3343.0639	0.99	3283.53058	3518.73008
									1	3289.76128	3525.39457

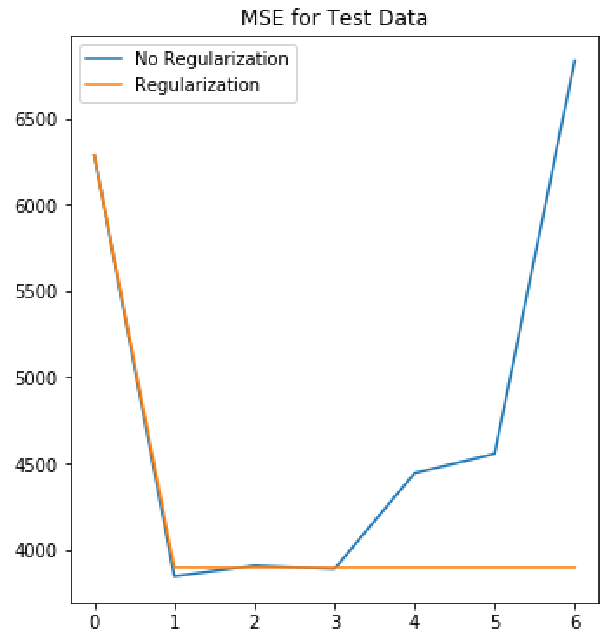
From the above graphs of MSE for train and test data, it is clear that applying gradient descent is pretty much same as as doing direct minimization by finding the inverse covariance matrix.

While gradient descent is slow in performance as it has to go through the iterations step by step, it is still better in the sense that there is no need to compute inverse of a matrix which is not possible a lot of times. Therefore, gradient always works whereas direct minimization technique is dependent on the existence of inverse of the covariance matrix.

We get the least value of MSE for test data when lambda equals 0.08. Therefore, 0.08 is our optimal value of lambda in case of gradient descent.

PROBLEM 5



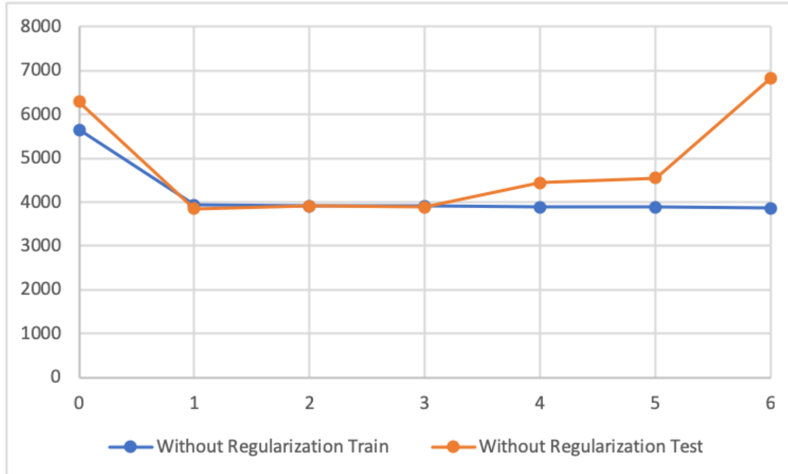


In the left graph, we can see that as p increases, the error keeps on decreasing for the train data. This is because a higher order polynomial curve is easily able to fit the training data than a lower order polynomial curve. The error for regularisation is more than no regularisation because regularisation stops the model from overfitting, and hence there is more error in the train data.

In the right graph, we can see that with regularization (lambda = 0.06), error for large values of p (p>3) in case of test data is small and more or less constant. However, error drastically increases without regularization in case of test data for large values of p because the model overfits without the presence of the regularization term.

Error vs p plot for MSE without regularization:

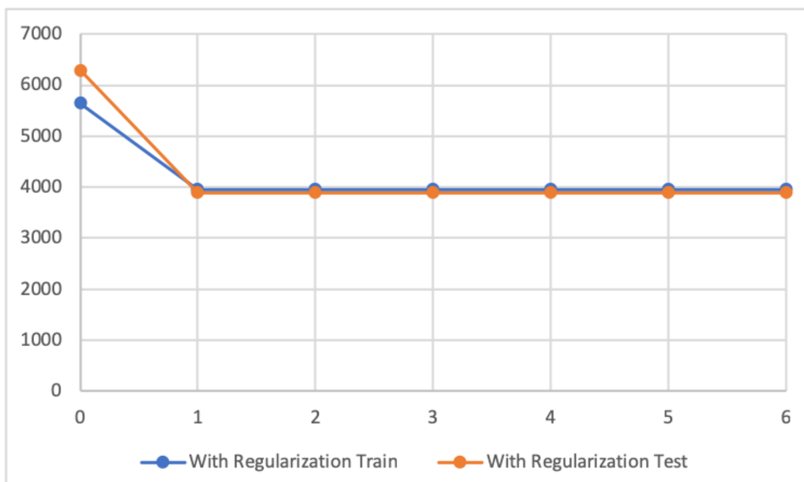
Without Regularization		
p	Train	Test
0	5650.71054	6286.40479
1	3930.91541	3845.03473
2	3911.83967	3907.1281
3	3911.18866	3887.97554
4	3885.47307	4443.32789
5	3885.40716	4554.83038
6	3866.88345	6833.45915



As can be seen in the above table, without regularization, the optimal value of p is 1. At p = 1, the error is least.

Error vs p plot for MSE with regularization:

With Regularization		
p	Train	Test
0	5650.71191	6286.88197
1	3951.83912	3895.85646
2	3950.68731	3895.58406
3	3950.68253	3895.58272
4	3950.68234	3895.58267
5	3950.68234	3895.58267
6	3950.68234	3895.58267



As can be seen in the above table, with regularization, the optimal value of p could be either one of 3, 4, 5 or 6. We would prefer a smaller value as it leads to lesser amount of computation. Therefore, the optimal value of p is 3.

PROBLEM 6

FINAL RESULTS

Metric used is MSE

In the below table we compare all MSE and find out the best solution.

Methods	Training Data	Testing Data
MSE without Intercept	19099.4468	106775.3615
MSE with Intercept	2187.1602	3707.8401
MSE with Ridge regression (Lambda = 0.06)	2187.1602	2851.3302
MSE with Ridge regression using Gradient Descent (Lambda = 0.08)	2415.8618	2836.5677
MSE with Non-Linear regression without regularization (p=1)	3866.8834	3845.0347
MSE with Non-Linear regression with Regularization (p=3)	3950.6823	3895.5826

In the above table we can see all errors and we use this table to choose the best solution among the given techniques for the provided datasets. From the table we can see for training data, **Linear Regression with Intercept** gives the least error which is 2187.1602. For testing data, **Ridge Regression using Gradient Descent** gives us the least error which is 2836.5677. We conclude the ridge regression with a lambda of 0.08 is the best technique for the given dataset.