UNIVERSITY AT BUFFALO

A Project Report

On

<u>CSE 574 (Machine Learning)</u>

<u>Classification and Regression Project</u>

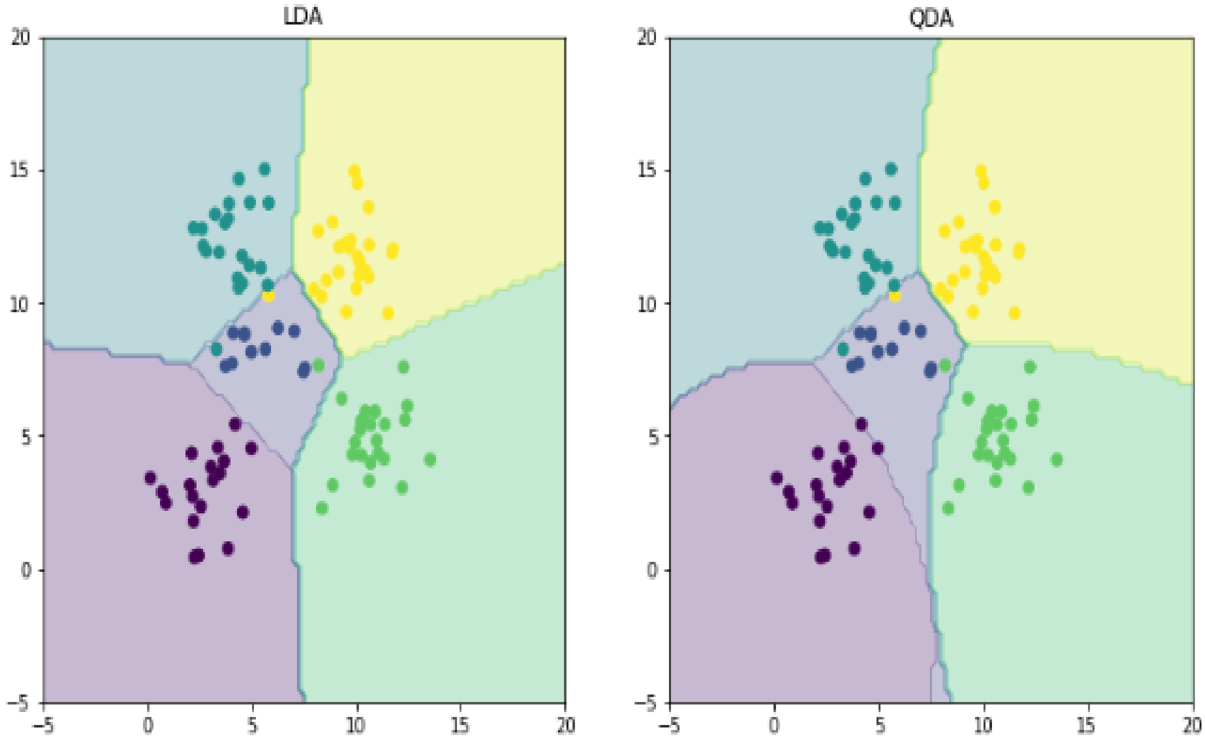
by

(Group 24)
Nishant Kapoor
Manasa Challa
Mahek Sangwan

Problem 1: Experiment with Gaussian Discriminators

The accuracy of obtained Linear Discriminant Analysis is **97%**

The accuracy of obtained Quadratic Discriminant Analysis is **95%**



In LDA, we assume that

1. Each class has normal distribution
2. Each class has its own mean
3. Variance is common amongst classes

Covariance matrix remains fixed in case of LDA analysis

In QDA, we assume that

1. Each class has its own mean factor
2. Generally, a class in QDA analysis carries its own covariance matrix

QDA is better for large datasets having low bias and higher variance

To get curved lines, we assign different covariance matrix in QDA.

Due to difference in Covariance matrix in LDA and QDA, we get different boundaries in LDA and QDA.

Problem 2: Experiment with Linear Regression

```
MSE for training data without intercept is [19097.4468447]
MSE for testing data without intercept is [106775.36145124]
MSE for training data with intercept is [2183.16029493]
MSE for testing data with intercept is[3707.84018096]
```

a) Training Data

- It is observed that there is a significant drop in the error reported in training data.
- Error_training = ((19097.4468447-2183.16029493)/19097.4468447)*100
- Error_training = 88.56%

b) Testing Data

- It is observed that there is a significant drop in the error reported in testing data.
- Error_testing = ((106775.36145-3707.8401)/ 106775.36145)*100
- Error_testing = 96.54%
- Thus, Using intercept showed better results than not using intercept.

Problem 3: Experiment with Ridge Regression

Varying values of λ from 0 (no regularization) to 1 in steps of 0.01

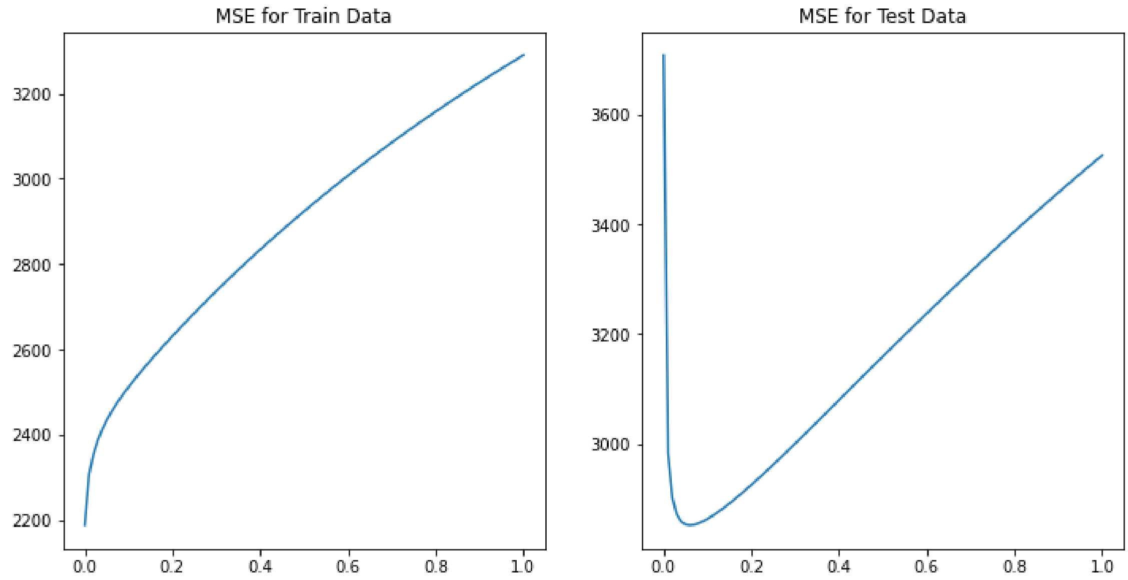
LAMBDAS	TRAINING DATA	TESTING DATA
0.0	array([2187.16029493])	array([3707.84018177]))
0.01	array([2306.83221793])	array([2982.44611971]))
0.02	array([2354.07134393])	array([2900.97358708]))
0.03	array([2386.7801631])	array([2870.94158888]))
0.04	array([2412.119043])	array([2858.00040957]))
0.05	array([2433.1744367])	array([2852.66573517]))
0.06	array([2451.52849064])	array([2851.33021344]))
0.07	array([2468.07755253])	array([2852.34999406]))
0.08	array([2483.36564653])	array([2854.87973918]))
0.09	array([2497.74025857])	array([2858.44442115]))
0.1	array([2511.43228199])	array([2862.75794143]))
0.11	array([2524.60003852])	array([2867.63790917]))
0.12	array([2537.35489985])	array([2872.96228271]))
0.13	array([2549.77688678])	array([2878.64586939]))
0.14	array([2561.92452773])	array([2884.62691417]))
0.15	array([2573.84128774])	array([2890.85910969]))
0.16	array([2585.55987497])	array([2897.30665895]))
0.17	array([2597.10519217])	array([2903.94112629]))
0.18	array([2608.49640025])	array([2910.73937213]))
0.19	array([2619.74838623])	array([2917.68216413]))
0.2	array([2630.8728232])	array([2924.75322165]))
0.21	array([2641.87894616])	array([2931.93854417]))
0.22	array([2652.77412633])	array([2939.22592987]))
0.23	array([2663.56430077])	array([2946.60462378]))
0.24	array([2674.25429667])	array([2954.06505602]))
0.25	array([2684.84807809])	array([2961.59864341]))
0.26	array([2695.34893502])	array([2969.19763677]))
0.27	array([2705.75962912])	array([2976.85500119]))
0.28	array([2716.0825067])	array([2984.56432079]))
0.29	array([2726.31958674])	array([2992.31972181]))
0.3	array([2736.4726296])	array([3000.11580946]))
0.31	array([2746.54319109])	array([3007.94761559]))
0.32	array([2756.53266482])	array([3015.81055453]))
0.33	array([2766.44231574])	array([3023.70038563]))
0.34	array([2776.27330654])	array([3031.61318093]))
0.35000000000000003	array([2786.02671854])	array([3039.54529713]))
0.36	array([2795.70356824])	array([3047.49335111]))
0.37	array([2805.30482034])	array([3055.45419817]))
0.38	array([2814.83139806])	array([3063.42491285]))
0.39	array([2824.28419133])	array([3071.40277169]))
0.4	array([2833.66406312])	array([3079.38523776]))
0.41000000000000003	array([2842.97185452])	array([3087.36994673]))
0.42	array([2852.2083886])	array([3095.35469418]))
0.43	array([2861.3744735])	array([3103.33742413]))
0.44	array([2870.47090474])	array([3111.31621849]))
0.45	array([2879.49846701])	array([3119.28928746]))
0.46	array([2888.45793552])	array([3127.25496075]))
0.47000000000000003	array([2897.35007697])	array([3135.21167941]))

0.48	array([2906.17565032])	array([3143.15798839]))
0.49	array([2914.93540723])	array([3151.09252966]))
0.5	array([2923.63009243])	array([3159.01403582]))
0.51	array([2932.26044392])	array([3166.92132421]))
0.52	array([2940.82719309])	array([3174.81329145]))
0.53	array([2949.33106473])	array([3182.68890838]))
0.54	array([2957.77277699])	array([3190.54721533]))
0.55	array([2966.15304137])	array([3198.38731777]))
0.56	array([2974.47256259])	array([3206.20838225]))
0.5700000000000001	array([2982.73203851])	array([3214.00963255]))
0.58	array([2990.93215999])	array([3221.79034621]))
0.59	array([2999.07361078])	array([3229.5498512]))
0.6	array([3007.15706742])	array([3237.28752288]))
0.61	array([3015.1831991])	array([3245.00278108]))
0.62	array([3023.15266757])	array([3252.69508746]))
0.63	array([3031.06612707])	array([3260.36394297]))
0.64	array([3038.92422416])	array([3268.00888553]))
0.65	array([3046.72759776])	array([3275.6294878]))
0.66	array([3054.47687898])	array([3283.22535516]))
0.67	array([3062.17269114])	array([3290.79612376]))
0.68	array([3069.81564971])	array([3298.34145873]))
0.6900000000000001	array([3077.40636224])	array([3305.86105245]))
0.7000000000000001	array([3084.94542842])	array([3313.354623]))
0.71	array([3092.43344001])	array([3320.82191265]))
0.72	array([3099.87098085])	array([3328.26268646]))
0.73	array([3107.25862691])	array([3335.67673095]))
0.74	array([3114.59694628])	array([3343.06385289]))
0.75	array([3121.88649919])	array([3350.42387813]))
0.76	array([3129.12783807])	array([3357.75665047]))
0.77	array([3136.3215076])	array([3365.0620307]))
0.78	array([3143.46804472])	array([3372.33989556]))
0.79	array([3150.56797875])	array([3379.59013686]))
0.8	array([3157.62183137])	array([3386.81266063]))
0.81	array([3164.63011677])	array([3394.00738631]))
0.8200000000000001	array([3171.59334168])	array([3401.17424594]))
0.8300000000000001	array([3178.51200544])	array([3408.31318353]))
0.84	array([3185.38660008])	array([3415.42415428]))
0.85	array([3192.21761044])	array([3422.50712403]))
0.86	array([3199.0055142])	array([3429.56206859]))
0.87	array([3205.75078202])	array([3436.58897321]))
0.88	array([3212.45387757])	array([3443.58783202]))
0.89	array([3219.11525768])	array([3450.55864755]))
0.9	array([3225.73537241])	array([3457.50143021]))
0.91	array([3232.31466512])	array([3464.41619786]))
0.92	array([3238.8535726])	array([3471.30297539]))
0.93	array([3245.35252514])	array([3478.16179431]))
0.9400000000000001	array([3251.81194665])	array([3484.99269234]))
0.9500000000000001	array([3258.23225474])	array([3491.79571308]))
0.96	array([3264.61386081])	array([3498.57090566]))
0.97	array([3270.95717015])	array([3505.3183244]))
0.98	array([3277.26258207])	array([3512.03802854]))
0.99	array([3283.53048993])	array([3518.7300819]))
(1.0	array([3289.7612813])	array([3525.39455263]))

According to the findings we observed that the value of MSE in testing data is lowest for λ =0.06. So, this is our solution

Referring to the MSE values from problem 2, it can be seen that MSE for ridge regression is lower compared to OLE regression, by this we can conclude that Ridge regression is more efficient than OLE regression.

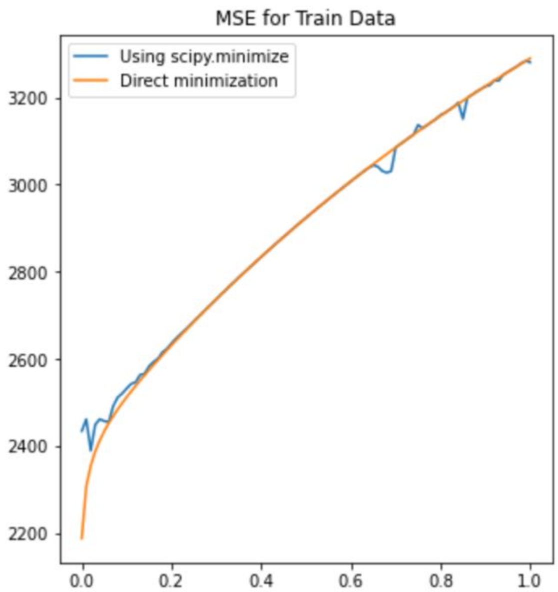
<u>Errors observed on training and testing data for different values of Lambda</u>

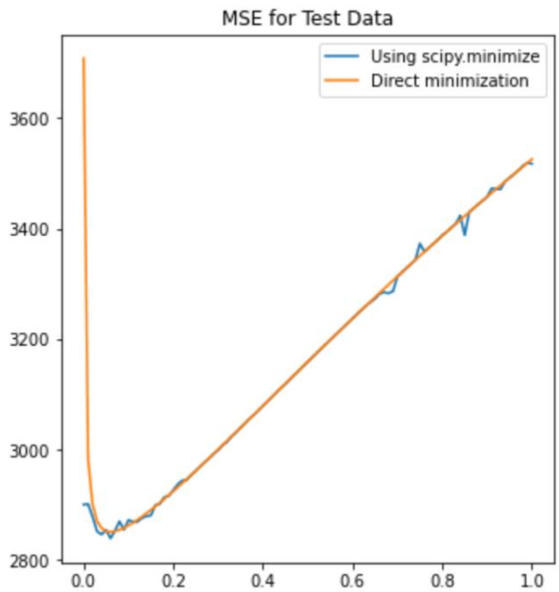


MSE findings for training data increases at a decreasing rate while in case of testing data, findings of MSE values decreases and then starts to escalate. Thus, there is dissimilarity in trends. At 0.2, the graph starts increasing at a constant rate for test data.

Problem 4: Using Gradient Descent for Ridge Regression Learning

Error on training and testing data (Using gradient descent based learning) by varying the regularization parameter





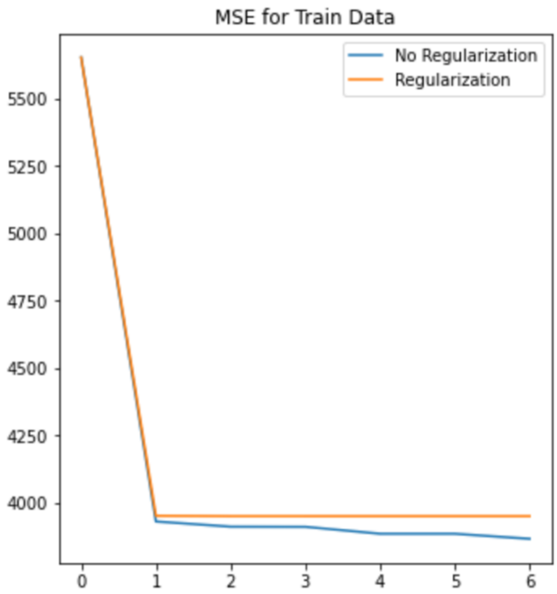
Comparing the obtained two graphs to the graphs in problem 3, we can see that both the graphs are very similar (almost same). Whenever the lambda value is very low or very high there are a lot of irregularities in the plot of gradient decent approach. The intermediate values don't have this issue.

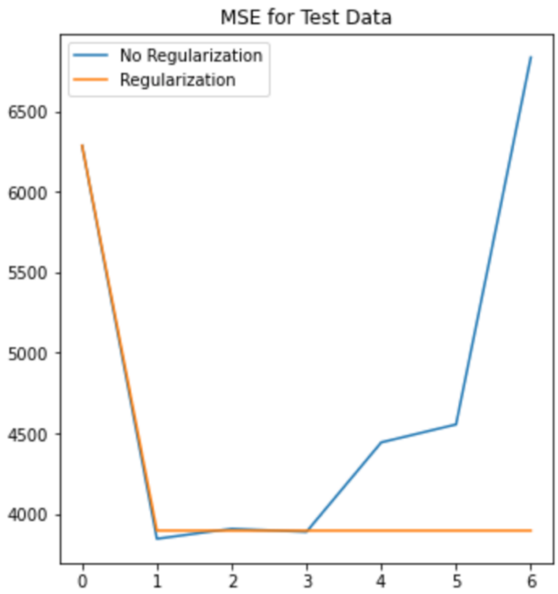
MSE data for Problem 4

```
(0.0, array([2433.67645892]), array([2900.54228854]))
(0.01, array([2375.768565]), array([2907.67057247]))
(0.02, array([2390.76136299]), array([2876.6310522]))
(0.03, array([2448.62539852]), array([2852.13981571]))
(0.04, array([2432.09206213]), array([2847.006124]))
(0.05, array([2451.80707236]), array([2845.54228913]))
(0.06, array([2458.10099225]), array([2865.60302752]))
(0.07, array([2482.27276248]), array([2851.93464119]))
(0.08, array([2515.66361551]), array([2862.72188406]))
(0.09, array([2509.43642924]), array([2847.76383296]))
(0.1, array([2526.1755762]), array([2862.34652875]))
(0.11, array([2542.14824896]), array([2869.809547]))
(0.12, array([2553.23099472]), array([2883.4363304]))
(0.13, array([2562.57868494]), array([2869.55235518]))
(0.14, array([2573.39886503]), array([2882.3580064]))
(0.15, array([2582.42416649]), array([2880.96733716]))
(0.16, array([2592.48403295]), array([2899.48073315]))
(0.17, array([2606.2984893]), array([2921.20646577]))
(0.18, array([2613.46510623]), array([2912.53777835]))
```

Problem 5: Non-linear Regression

Errors on training and testing data





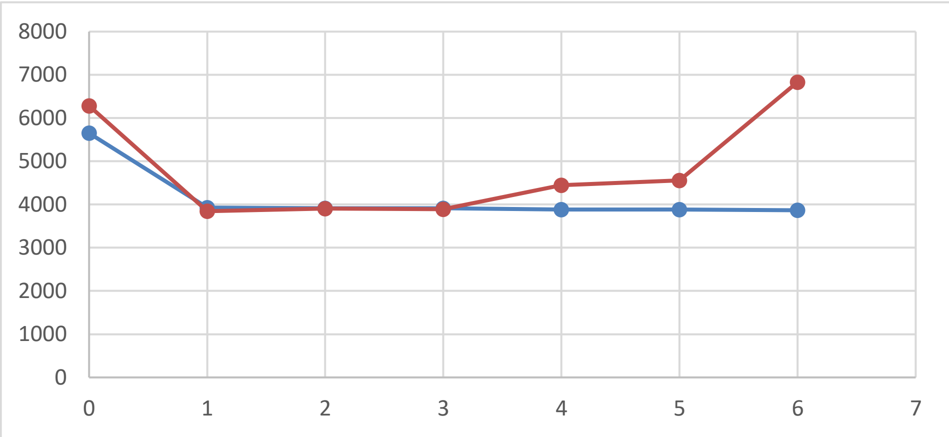
From problem 3, where there is no regularization, we choose **λ** to be 0.06 .

Graph 1 i.e, MSE of training data clearly shows that there is a declination in the error. The error estimated at p=1.0 in both the colour lines is nearly equal. When p is greater than 1.0, a linear graph was observed along x-axis in the case of regularization. Thus, a decrease in the error is observed.

Graph 2 i.e, MSE of testing data clearly shows that for no regularization. When p is greater than 1.0, the error increases and at p = 3, growth rate becomes fast and starts escalating more. With respect to regularization similar trend was observed on training data. It gets the minimum error at p = 1. Relying more on training data leads to increment in error in case of test data.

Curve for the value of p varying from 0 to 6 for λ = 0 (without regularization)

p	Train (Without regularization)	Test (Without regularization)
0	5650.7105389	6286.40479168
1	3930.91540732	3845.03473017
2	3911.8396712	3907.12809911
3	3911.18866493	3887.97553824
4	3885.47306811	4443.32789181
5	3885.4071574	4554.83037743
6	3866.88344945	6833.45914872



Red color indicates training without regularization

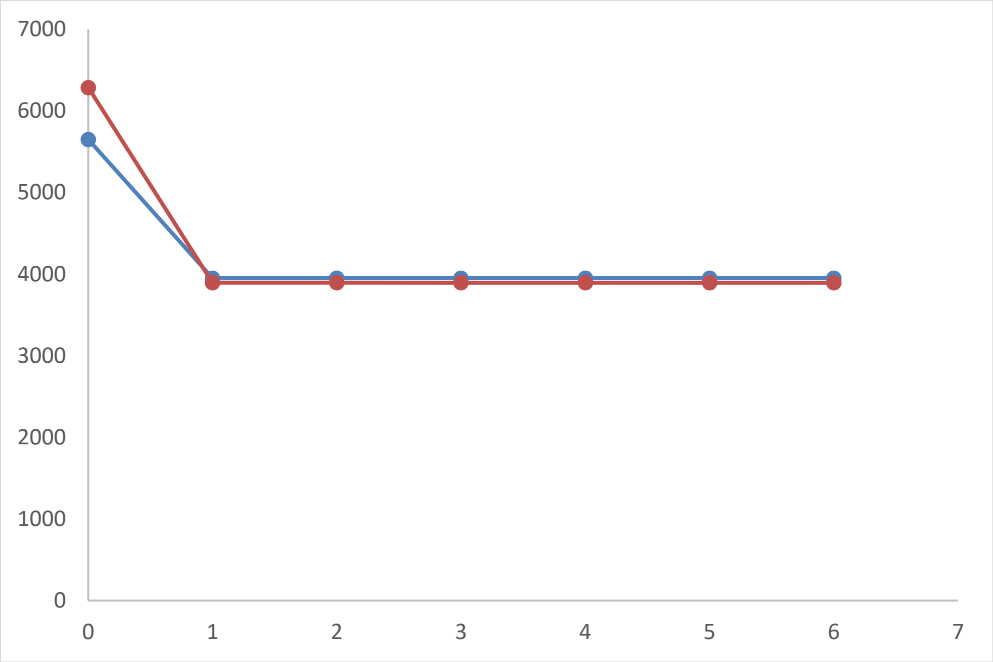
Blue color indicates testing without regularization

The optimal value for p in case of test error is 5 as curve is constant after that value.

The optimal value for p in case of train error is 1.

Curve for the optimal value of p varying from 0 to 6 for λ = 0.06 (with regularization)

p	Train (With regularization)	Test (With regularization)
0	5650.71190703	6286.88196694
1	3951.83912356	3895.85646447
2	3950.68731238	3895.58405594
3	3950.68253152	3895.58271592
4	3950.6823368	3895.58266828
5	3950.68233518	3895.5826687
6	3950.68233514	3895.58266872



The optimal value for p in case of test error is 4.

The optimal value for p in in case of train error is 4 or 5.

PROBLEM 6 – INTERPRETING RESULTS

Type of Regression	**Train MSE**	**Test MSE**
1. With intercept (MSE Findings)	2183.16029	3707.8401
2. Without intercept (MSE Findings)	19097.446847	106775.36145
3. Ridge regression (MSE Findings)	2453.4284	2853.3328
4. Ridge regression using Gradient Descent (MSE Findings)	2454.4148	2839.4193
5. Non-Linear regression with regularization (MSE Findings)	3950.6823	3895.5826

As error obtained is minimum in calculating MSE with intercept for training data and MSE with ridge regression in testing data, we choose these metrics as our optimal solutions.