CSE574 Introduction to Machine Learning

Programming Assignment 2

Classification and Regression

Team-60

Himal Dwarakanath (himaldwa)
Manish Kasireddy (manishka)
MuthuPalaniappan Karuppayya (muthupal)

Introduction

In this assignment, we have performed classification and regression techniques on the given data (Diabetes). The results and their interpretation are discussed in this report

The following experiments were performed:

1. Gaussian Discriminators
2. Linear Regression
3. Ridge Regression
4. Ridge Regression using gradient descent
5. Non-Linear Regression

1. Experiment with Gaussian Discriminators

We attempt to compare the performance of Linear Discriminant Analysis (LDA) and Quadratic Discriminant (QDA) after training both of them over a sample dataset. In both the training phases we split the input data into partitions such that all entries in each partition of input maps to a common output. Post this, we calculate the local means for each of the partitions for the *k* classes of output data to obtain the means matrices (μ_0, μ_1, … μ_{k-1}). We then calculate the covariance (Σ) amongst the dimensions of the input.

In the case of LDA, the covariance matrix is calculated on the complete input dataset, by using a global mean of all the data. While in the case of QDA, we calculate a separate covariance matrix for each partitioned input data set using the data set's local mean. The covariance matrix plays a crucial role in summarizing the shape of the distribution. LDA learns linear boundaries between classes of the data points while QDA forms quadratic boundaries.

While QDA conceptually gives more flexible boundaries, it is often seen that LDA defines better boundaries in scenarios where the data is actually linearly separable or if the training data is a limited set which isn't very densely populated. We analyze this in the following discussion.

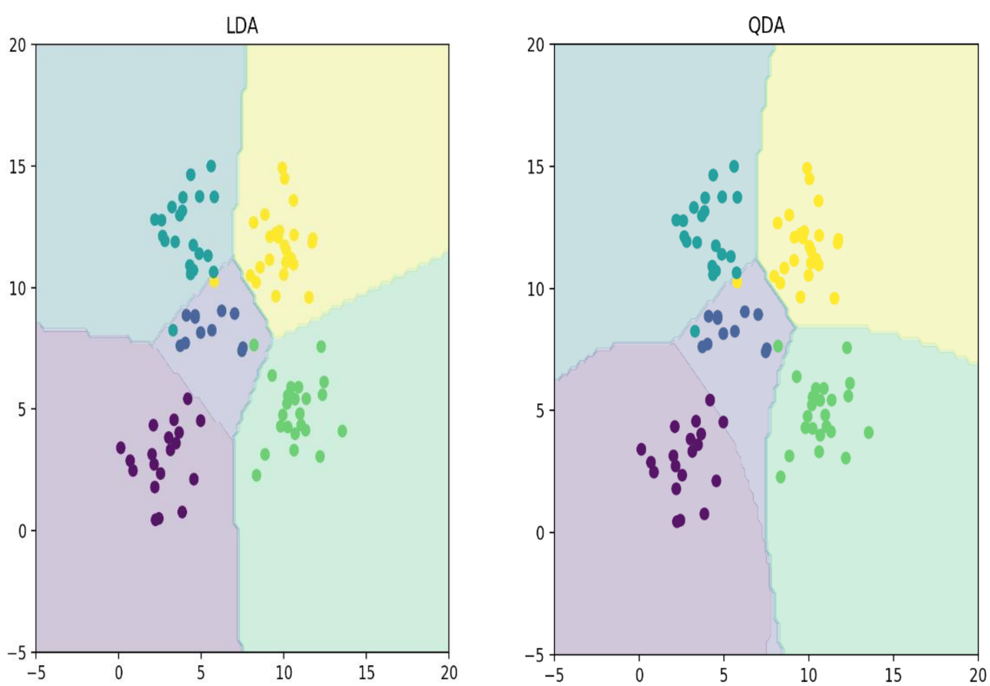
After training the mean and covariance matrices using the training data, we use them to predict the output values for a given set of test data by using the below equation:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k))$$

It is to be noted that while predicting using LDA, we may consider the determinant of covariance matrix in the denominator to be a constant, while in QDA, the covariance matrix changes for each class. This difference results in a slightly higher accuracy when we use LDA for prediction.

With the above process, we get accuracy of **97% for LDA and 96% for QDA.**

Below are the boundaries formed by LDA and QDA:



2. Experiment with Linear Regression

	MSE without Intercept	MSE with Intercept	% Improvement with Intercept
Training Data	19099.4468446	2187.16029493	88.55
Testing Data	106775.361555	3707.84018132	96.53

It can be seen the MSE value **with intercept** is better in both training data and test data. This is because, without an intercept, the linear regression line is forced to pass through the origin and thus, does not fit with well with the actual data. When we add the intercept, the linear regression line is aligned more closely with actual data. The training data error is lesser compared to test data because there are possibly lesser outliers. From the above table, we can see the 2 advantages of using an intercept: a significant error decrease when considering a single data set (either training or test), and an even more impressive reduction of the error committed on the test set (96.53%) compared to the training set (88.55%).

3. Experiment with Ridge Regression

The Ridge regression is similar to Linear Regression, with a minor modification in the implementation. Ridge Regression includes an additional parameter called Regularization parameter (**λ**). The **λ** will be fed back with a value that adjusts the weights in a way to reduce the Mean Square Error.

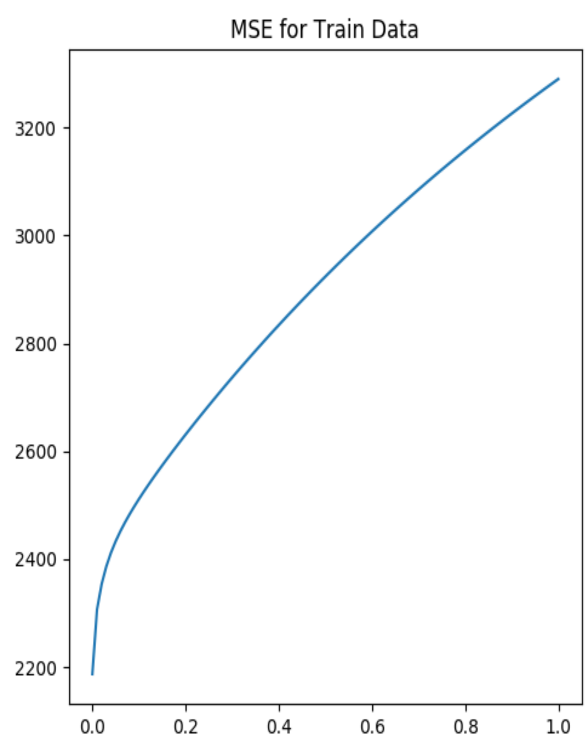
A cross-validation approach is used to select the best value for **λ**. A model is fitted to the training set with a specific value of **λ**. Once values for the co-efficient have been determined, the predictive accuracy of the model is determined by applying the model to test set data. This process is repeated for different values of **λ**. The model with the least MSE on the test set is then selected.

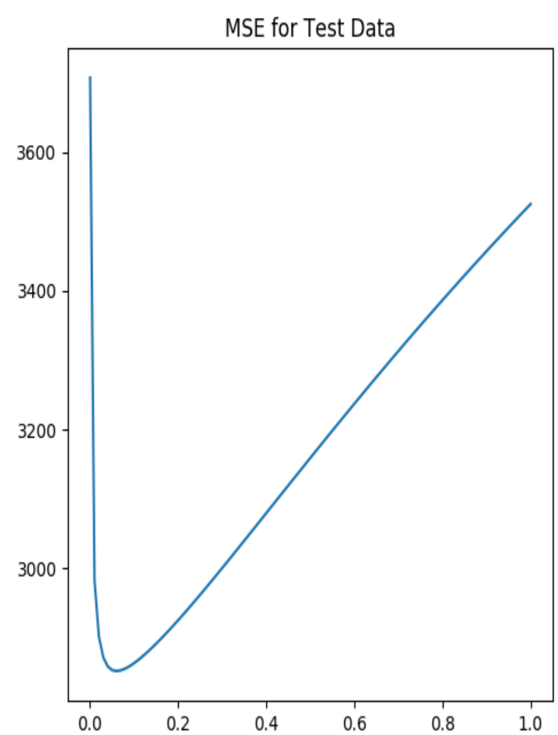
The below tables displays the MSE values calculated for Test and Training data set with **λ** values varying from 0 to 1 in the steps of 0.01.

λ	Test - MSE	Train - MSE		λ	Test - MSE	Train - MSE
0	3707.840181	2187.160295		0.51	3166.921324	2932.260444
0.01	2982.44612	2306.832218		0.52	3174.813291	2940.827193
0.02	2900.973587	2354.071344		0.53	3182.688908	2949.331065
0.03	2870.941589	2386.780163		0.54	3190.547215	2957.772777
0.04	2858.00041	2412.119043		0.55	3198.387318	2966.153041
0.05	2852.665735	2433.174437		0.56	3206.208382	2974.472563
0.06	**2851.330213**	2451.528491		0.57	3214.009633	2982.732039
0.07	2852.349994	2468.077553		0.58	3221.790346	2990.93216
0.08	2854.879739	2483.365647		0.59	3229.549851	2999.073611
0.09	2858.444421	2497.740259		0.6	3237.287523	3007.157067
0.1	2862.757941	2511.432282		0.61	3245.002781	3015.183199
0.11	2867.637909	2524.600039		0.62	3252.695087	3023.152668
0.12	2872.962283	2537.3549		0.63	3260.363943	3031.066127
0.13	2878.645869	2549.776887		0.64	3268.008886	3038.924224
0.14	2884.626914	2561.924528		0.65	3275.629488	3046.727598
0.15	2890.85911	2573.841288		0.66	3283.225355	3054.476879
0.16	2897.306659	2585.559875		0.67	3290.796124	3062.172691
0.17	2903.941126	2597.105192		0.68	3298.341459	3069.81565
0.18	2910.739372	2608.4964		0.69	3305.861052	3077.406362
0.19	2917.682164	2619.748386		0.7	3313.354623	3084.945428
0.2	2924.753222	2630.872823		0.71	3320.821913	3092.43344
0.21	2931.938544	2641.878946		0.72	3328.262686	3099.870981
0.22	2939.22593	2652.774126		0.73	3335.676731	3107.258627
0.23	2946.604624	2663.564301		0.74	3343.063853	3114.596946
0.24	2954.065056	2674.254297		0.75	3350.423878	3121.886499
0.25	2961.598643	2684.848078		0.76	3357.75665	3129.127838
0.26	2969.197637	2695.348935		0.77	3365.062031	3136.321508
0.27	2976.855001	2705.759629		0.78	3372.339896	3143.468045
0.28	2984.564321	2716.082507		0.79	3379.590137	3150.567979
0.29	2992.319722	2726.319587		0.8	3386.812661	3157.621831
0.3	3000.115809	2736.47263		0.81	3394.007386	3164.630117
0.31	3007.947616	2746.543191		0.82	3401.174246	3171.593342
0.32	3015.810555	2756.532665		0.83	3408.313184	3178.512005
0.33	3023.700386	2766.442316		0.84	3415.424154	3185.3866
0.34	3031.613181	2776.273307		0.85	3422.507124	3192.21761

0.35	3039.545297	2786.026719		0.86	3429.562069	3199.005514
0.36	3047.493351	2795.703568		0.87	3436.588973	3205.750782
0.37	3055.454198	2805.30482		0.88	3443.587832	3212.453878
0.38	3063.424913	2814.831398		0.89	3450.558648	3219.115258
0.39	3071.402772	2824.284191		0.9	3457.50143	3225.735372
0.4	3079.385238	2833.664063		0.91	3464.416198	3232.314665
0.41	3087.369947	2842.971855		0.92	3471.302975	3238.853573
0.42	3095.354694	2852.208389		0.93	3478.161794	3245.352525
0.43	3103.337424	2861.374474		0.94	3484.992692	3251.811947
0.44	3111.316218	2870.470905		0.95	3491.795713	3258.232255
0.45	3119.289287	2879.498467		0.96	3498.570906	3264.613861
0.46	3127.254961	2888.457936		0.97	3505.318324	3270.95717
0.47	3135.211679	2897.350077		0.98	3512.038029	3277.262582
0.48	3143.157988	2906.17565		0.99	3518.730082	3283.53049
0.49	3151.09253	2914.935407		1	3525.394553	3289.761281
0.5	3159.014036	2923.630092				

The below plots show the MSE values calculated for varying **λ** values. From the below plots and above table, we can infer the optimal **λ** value is **0.06**.





Comparing the two approaches, linear regression and Ridge regression, in terms of MSE,

MSE for training data with intercept using Linear regression: **2187.16029493**

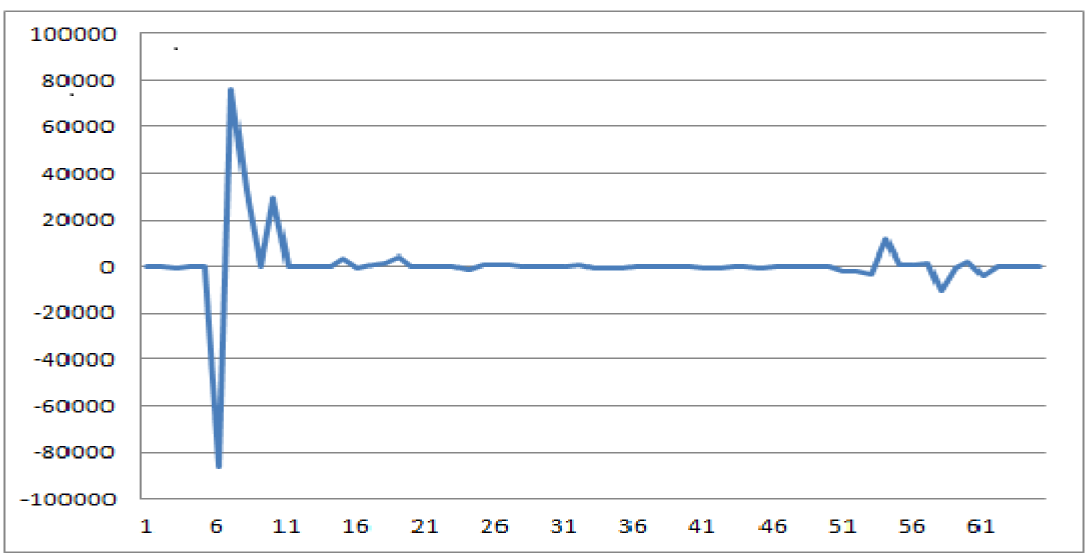
MSE for testing data with intercept using Linear regression: **3707.84018103**

MSE for training data using Ridge Regression (optimal, **λ** = 0): **2187.16029493**

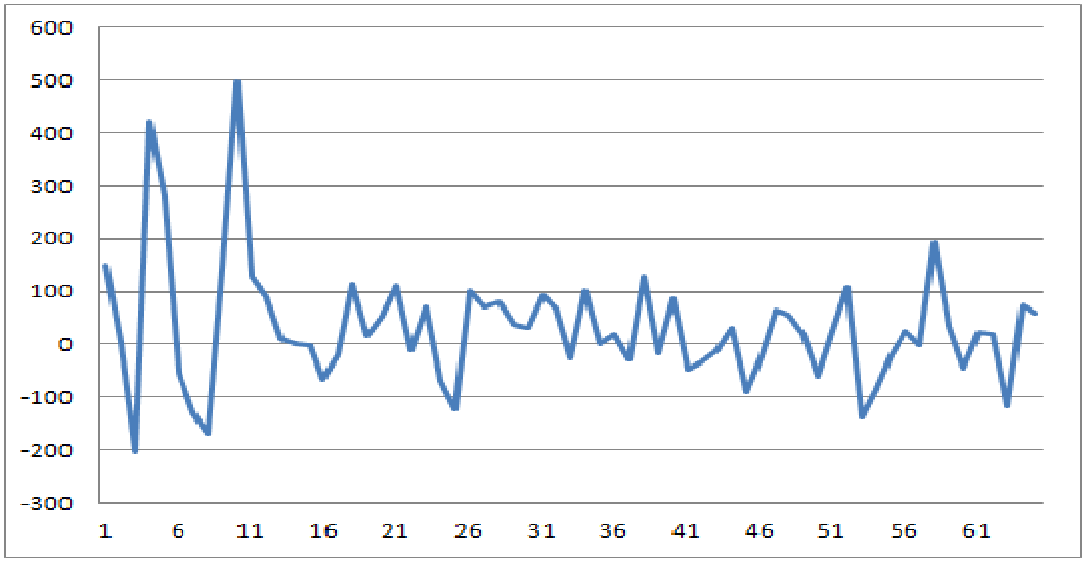
MSE for testing data using Ridge Regression (optimal, **λ =** 0.06): **2851.330213**

From the values of MSE for Linear regression and ridge regression it is clear that the error for ridge regression for test data is lower for ridge regression and it is a better approach.

Comparing the two approaches, in terms of weights, based on the below plots, we can see that weights learnt using Linear Regression have a much higher magnitude compared to weights learnt using Ridge Regression. This huge difference is due to the regularization parameter (**λ**) used in Ridge Regression.

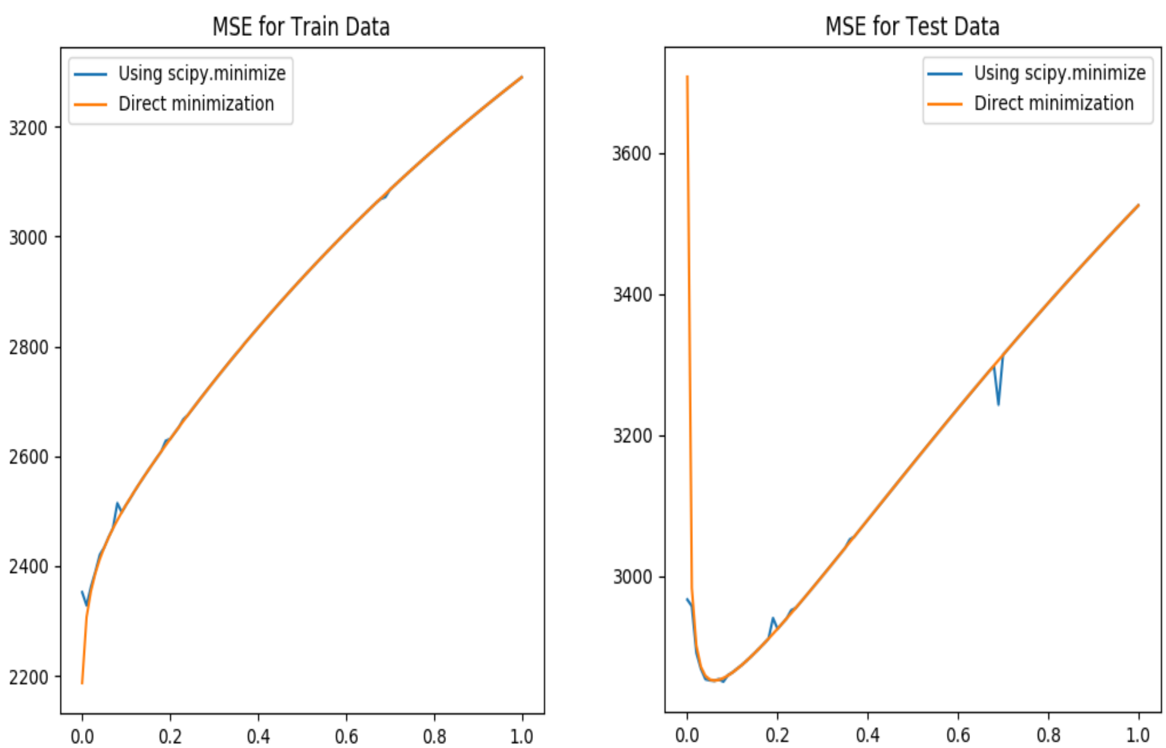


Weights learnt using Linear Regression with intercept



Weights learnt using Ridge Regression at optimal **λ** with intercept

4. Experiment using Gradient Descent for Ridge Regression Learning



In the plots above, the curves obtained using regular ridge regression and using gradient descent method is nearly identical. However, the lines produced using gradient descent method is not as smooth as those produced using regular ridge regression and has some outliers.

The optimal MSE using gradient descent for ridge regression is as follows:

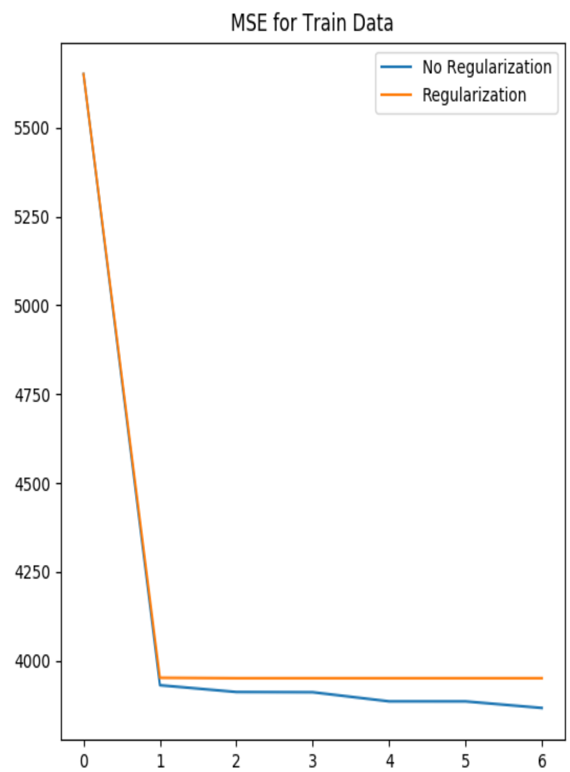
MSE for train data = **2313.35303649**

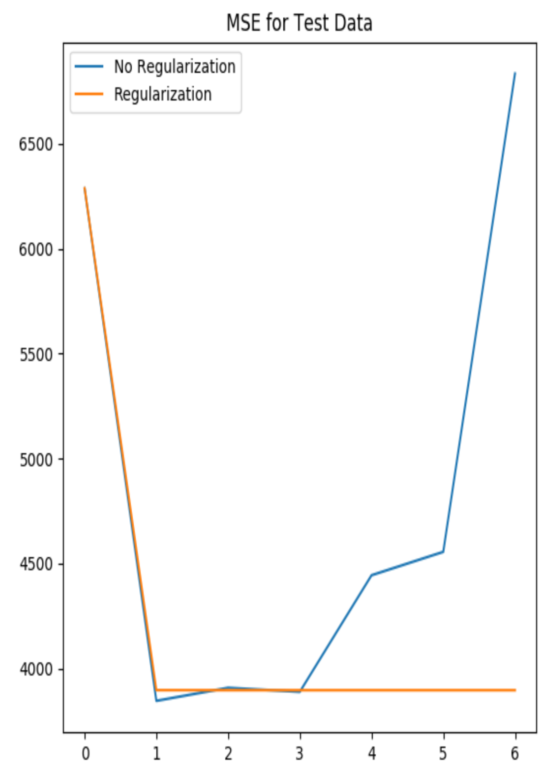
MSE for test data = **2850.64091846**

As seen above and from the data got from ridge regression, MSE is higher for gradient descent. This is because the minimize function in gradient descent takes a while to converge. Hence, the regular ridge regression is faster than gradient descent method and gives lesser MSE. However, when the matrices are bigger, the inversion of matrices is expensive. In such cases, gradient descent is a better option compared to regular ridge regression.

5. Experiment with Non-Linear Regression

The results of this problem show the correlation between the order of polynomial and mean squared error (MSE) for a given set of test and train data. The order of polynomial(p) is varied from 0 to 6. The below plots show MSE values calculated for different values of p and with and without regularization(**λ**).





It is observed that for the train data, we get minimal MSE when **p = 6** in both cases. The minimal values recorded are as follows:

MSE for training data without Regularization: **3866.89**

MSE for training data with Regularization: **3950.68**

For the test data, the minimal MSE without regularization is observed at **p = 1** while the minimum MSE with regularization is observed at **p = 4**. The minimal values recorded are as follows:

MSE for test data without Regularization: **3845.03**

MSE for test data with Regularization: **3895.58**

Conclusion

Problem	Train MSE	Test MSE
2 – LR with Intercept	2187.16029493	3707.84018132
2 – LR without intercept	19099.4468446	106775.361555
3 – Optimal RR	2187.16029493	2851.330213
4 – Optimal RR with GD	2313.35303649	2850.64091846
5 – Optimal NLR (No Regularization)	3866.89	3845.03
5 – Optimal NLR (Regularization)	3950.68	3895.58

Using the MSE values calculated for different models for the given data set, Ridge Regression and Ridge Regression with Gradient Descent, with optimal **λ,** are the best choices. Comparatively, Linear and Non-Linear Regression MLE values for this given data set are significantly much higher. Hence it is not recommended to use these 2 techniques for the given data set.

For small datasets (like given dataset), normal Ridge Regression performs slights better and much faster compared to Gradient Descent. But when we deal with huge datasets with higher order matrices, Gradient Descent will be much faster compared to Ridge Regression and is recommended.