CSE 574 Machine Learning

Assignment 2: Classification and Regression

April 12, 2017

Group 64

Debanjan Paul - 50208716
Amaan Modak - 50206525
Jay Bakshi - 50206954

Report 1: Experiment with Gaussian Discriminators

LDA and QDA are two classic classifiers. They have closed-form solutions that can be easily computed, are inherently multi-class, have proven to work well in practice and have no hyper parameters to tune. As can be seen from the plot below, LDA has a decision boundary which is linear and QDA has decision boundary which is quadratic. Both, LDA and QDA use similar approach for training, the only difference being, unlike QDA, LDA assumes a single generalized covariance matrix, while QDA trains individual covariance matrices for each of the

The accuracy for LDA and QDA for the sample data is as follows:
LDA: 97.00%
QDA: 96.00%

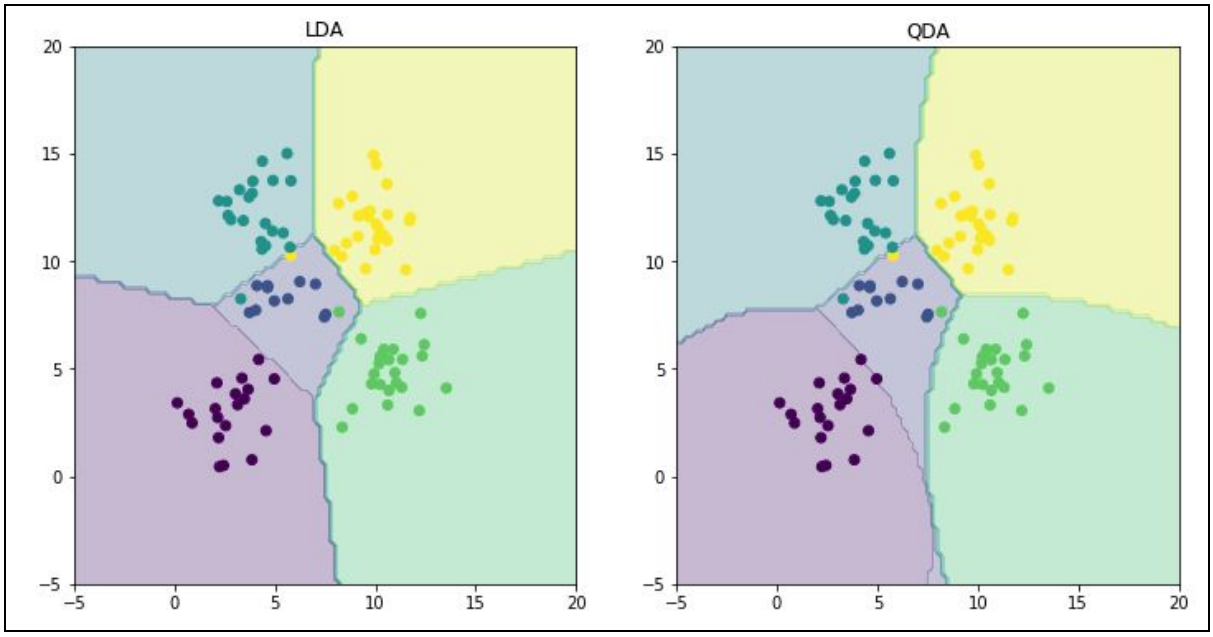

Fig.1 LDA and QDA boundaries.

In the case of LDA, the Gaussians for each class share the same covariance matrix. This leads to linear decision surfaces between the data. In the case of QDA, there are no assumptions that the covariance matrices for each of the classes are identical, leading to quadratic decision surfaces. Thus, the differences between the two boundaries are literally, that LDA draws straight line boundaries, and quadratic draws curved lines(which are obviously quadratic).

Report 2: Experiment with Linear Regression

Train Data is provided in “diabetes.pickle”, variables **X (training data)** and **y (training labels)**
Test Data is provided in “diabetes.pickle”, variables **Xtest (testing data)** and **ytest (testing labels)**

The first thing we calculate is the value of parameter **w** - weight matrix,

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$ **[1]**

Then we proceed to calculate the MSE,

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 \quad = \quad \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$ **[2]**

This gives us the following results, without using an intercept and with using an intercept:

Data	Without intercept	With intercept
Training	19099.4468446	2187.16029493
Testing	106775.361424	3707.84018038

Table.1 MSE Values for Training Data and Testing Data, each with and without intercept.

Our observations -

1. MSE values with intercept improve significantly when comparing it for the same data-set i.e. for training data there is a decrease of about 88% in error and for testing data there is decrease of about 96%!
2. MSE values when compared across the data sets shows the error rate when without intercept for testing data is about 5.6% increase over training data. This however, changes for with intercept across the data sets as error observed in testing data is only 1.7% increase over training data.

Based on the above observations, we learn that error is reduced significantly when an intercept is used. The overall accuracy is much higher with intercept as compared to without intercept. This behavior is because with intercept we get a better fit for linear regression. Hence error is less for data with intercept.

Report 3: Experiment with Ridge Regression

mses3_train	mses3	lambdas	mses3_train	mses3	lambdas	mses3_train	mses3	lambdas
2187.16	3707.84	0.00	2888.46	3127.25	0.46	3238.85	3471.30	0.92
2306.83	2982.45	0.01	2897.35	3135.21	0.47	3245.35	3478.16	0.93
2354.07	2900.97	0.02	2906.18	3143.16	0.48	3251.81	3484.99	0.94
2386.78	2870.94	0.03	2914.94	3151.09	0.49	3258.23	3491.80	0.95
2412.12	2858.00	0.04	2923.63	3159.01	0.50	3264.61	3498.57	0.96
2433.17	2852.67	0.05	2932.26	3166.92	0.51	3270.96	3505.32	0.97
2451.53	2851.33	0.06	2940.83	3174.81	0.52	3277.26	3512.04	0.98
2468.08	2852.35	0.07	2949.33	3182.69	0.53	3283.53	3518.73	0.99
2483.37	2854.88	0.08	2957.77	3190.55	0.54	3289.76	3525.39	1.00
2497.74	2858.44	0.09	2966.15	3198.39	0.55			
2511.43	2862.76	0.10	2974.47	3206.21	0.56			
2524.60	2867.64	0.11	2982.73	3214.01	0.57			
2537.35	2872.96	0.12	2990.93	3221.79	0.58			
2549.78	2878.65	0.13	2999.07	3229.55	0.59			
2561.92	2884.63	0.14	3007.16	3237.29	0.60			
2573.84	2890.86	0.15	3015.18	3245.00	0.61			
2585.56	2897.31	0.16	3023.15	3252.70	0.62			
2597.11	2903.94	0.17	3031.07	3260.36	0.63			
2608.50	2910.74	0.18	3038.92	3268.01	0.64			
2619.75	2917.68	0.19	3046.73	3275.63	0.65			
2630.87	2924.75	0.20	3054.48	3283.23	0.66			
2641.88	2931.94	0.21	3062.17	3290.80	0.67			
2652.77	2939.23	0.22	3069.82	3298.34	0.68			
2663.56	2946.60	0.23	3077.41	3305.86	0.69			
2674.25	2954.07	0.24	3084.95	3313.35	0.70			
2684.85	2961.60	0.25	3092.43	3320.82	0.71			
2695.35	2969.20	0.26	3099.87	3328.26	0.72			
2705.76	2976.86	0.27	3107.26	3335.68	0.73			
2716.08	2984.56	0.28	3114.60	3343.06	0.74			
2726.32	2992.32	0.29	3121.89	3350.42	0.75			
2736.47	3000.12	0.30	3129.13	3357.76	0.76			
2746.54	3007.95	0.31	3136.32	3365.06	0.77			
2756.53	3015.81	0.32	3143.47	3372.34	0.78			
2766.44	3023.70	0.33	3150.57	3379.59	0.79			
2776.27	3031.61	0.34	3157.62	3386.81	0.80			
2786.03	3039.55	0.35	3164.63	3394.01	0.81			
2795.70	3047.49	0.36	3171.59	3401.17	0.82			
2805.30	3055.45	0.37	3178.51	3408.31	0.83			
2814.83	3063.42	0.38	3185.39	3415.42	0.84			
2824.28	3071.40	0.39	3192.22	3422.51	0.85			
2833.66	3079.39	0.40	3199.01	3429.56	0.86			
2842.97	3087.37	0.41	3205.75	3436.59	0.87			
2852.21	3095.35	0.42	3212.45	3443.59	0.88			
2861.37	3103.34	0.43	3219.12	3450.56	0.89			
2870.47	3111.32	0.44	3225.74	3457.50	0.90			
2879.50	3119.29	0.45	3232.31	3464.42	0.91			

Table.2 λ, MSE for Training Data and MSE for Testing Data.

For Training Data MSE value is lowest at $\lambda = 0$ and for Testing data MSE value is lowest at $\lambda = 0.06$. These values are the optimal λ values for respective datasets.

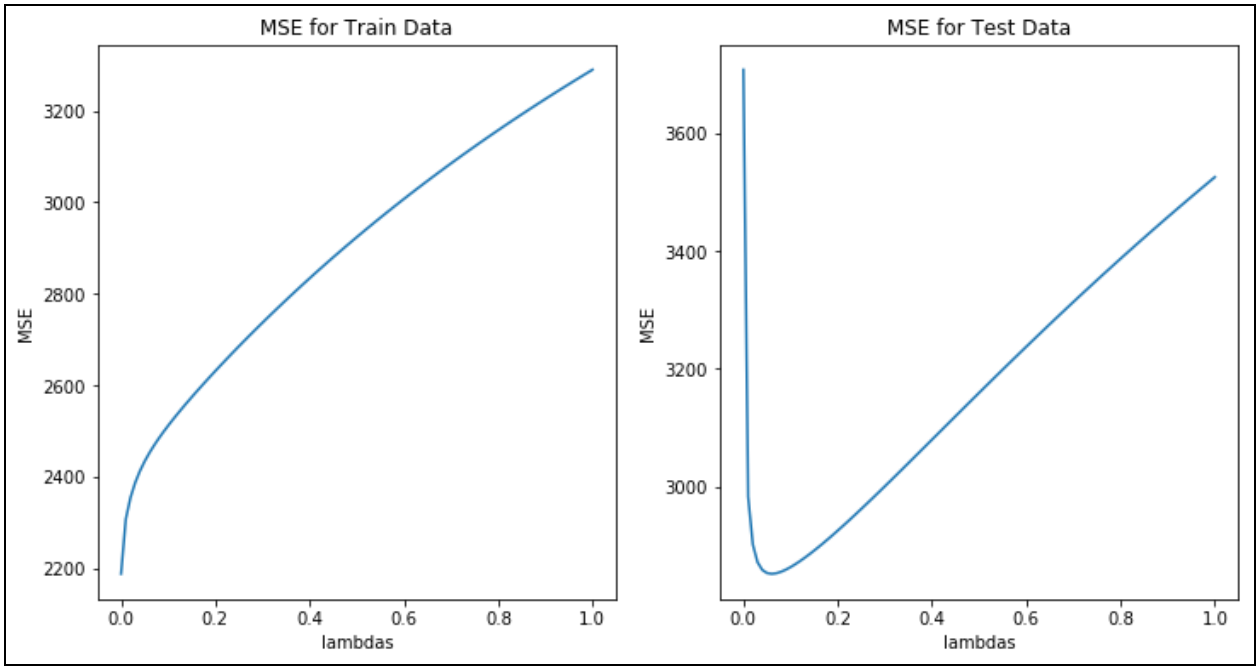

Fig.2 MSE for different values of λ plotted side-by-side.

We are varying λ in step values of 0.01 within the range of [0,1] inclusive of the boundary values.

Observations -

1. MSE values (with intercept) calculated with OLE Regression are the same as MSE values calculated with Ridge Regression for $\lambda = 0$.
2. But with $\lambda = 0.06$ we get the optimum value of MSE value calculated with Ridge Regression, since it is significantly lower than the MSE value for OLE.

This leads to our conclusion that Ridge Regression is more efficient algorithm than OLE, on the dataset we're using for this project.

Report 4: Using Gradient Descent for Ridge Regression Learning

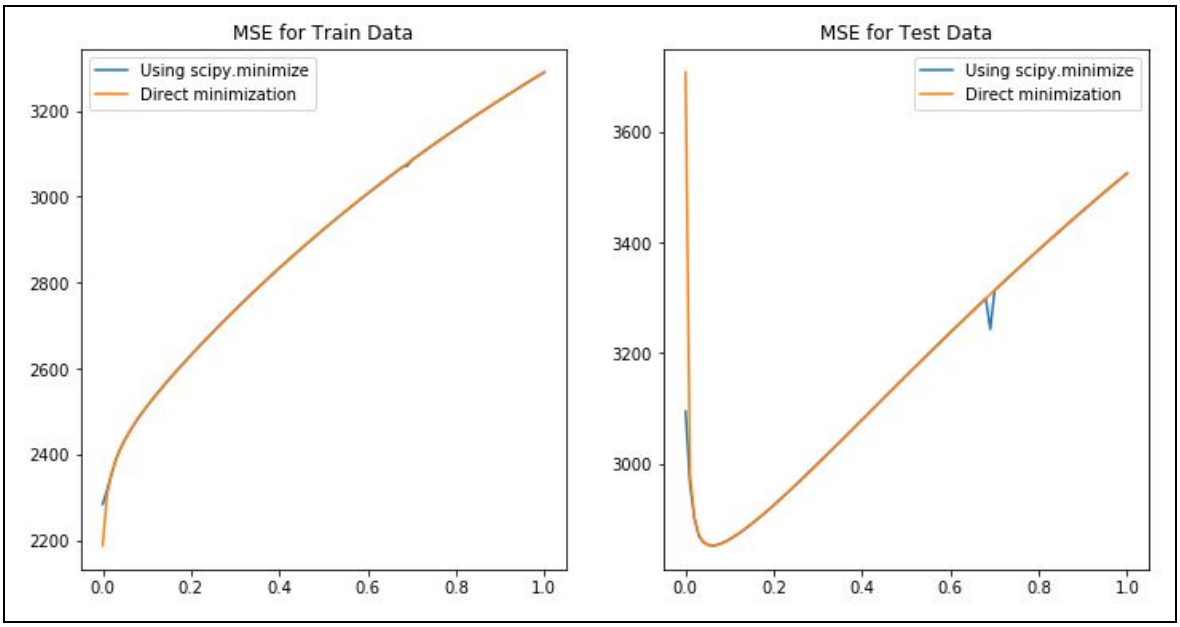

Fig.3 MSE for Training Data vs. λ and MSE for Testing Data vs. λ, both using Regular and Gradient Descent Algorithm.

Comparing the MSE values for Regular Ridge Regression and Ridge Regression using Gradient Descent Algorithm we can clearly see that the plot is very much identical. Though there are 2 key differences noticeable:

1. There are a few outliers visible in result using Gradient Descent Algorithm, while the curve is smooth for the Regular Ridge Regression. So Regular Ridge Regression basically handles the case of a few outliers in the data better, by ignoring it when applicable.
2. There is a trade-off between the size of the dataset and the algorithm of choice. For dataset in our project consideration both are working almost equal. However, for larger matrices, inverse operation in Regular Ridge Regression is not the computationally the favored choice when compared with iterative step function in Gradient Descent Algorithm.

The error function used is given below:

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2}\lambda\mathbf{w}^\top\mathbf{w}$$ **[3]**

mses4_train	mses4	lambdas	mses4_train	mses4	lambdas	mses4_train	mses4	lambdas
2258.98	3236.87	0.00	2888.46	3127.25	0.46	3238.85	3471.30	0.92
2316.70	2964.71	0.01	2897.35	3135.21	0.47	3245.35	3478.16	0.93
2354.61	2900.36	0.02	2906.18	3143.16	0.48	3251.81	3484.99	0.94
2387.14	2869.88	0.03	2914.94	3151.09	0.49	3258.23	3491.80	0.95
2412.80	2857.38	0.04	2923.63	3159.01	0.50	3264.61	3498.57	0.96
2433.76	2852.40	0.05	2932.26	3166.92	0.51	3270.96	3505.32	0.97
2451.54	2851.30	0.06	2940.83	3174.81	0.52	3277.26	3512.04	0.98
2502.35	2874.36	0.07	2949.33	3182.69	0.53	3283.53	3518.73	0.99
2483.33	2854.91	0.08	2957.77	3190.55	0.54	3289.76	3525.39	1.00
2497.75	2858.45	0.09	2966.15	3198.39	0.55			
2511.45	2862.73	0.10	2974.47	3206.21	0.56			
2524.60	2867.64	0.11	2982.73	3214.01	0.57			
2537.34	2872.95	0.12	2990.93	3221.79	0.58			
2549.78	2878.65	0.13	2999.07	3229.55	0.59			
2561.92	2884.63	0.14	3007.16	3237.29	0.60			
2573.84	2890.86	0.15	3015.18	3245.00	0.61			
2585.56	2897.31	0.16	3023.15	3252.70	0.62			
2597.11	2903.94	0.17	3031.07	3260.36	0.63			
2608.49	2910.74	0.18	3038.92	3268.01	0.64			
2625.26	2912.21	0.19	3046.73	3275.63	0.65			
2630.87	2924.75	0.20	3054.48	3283.23	0.66			
2641.88	2931.94	0.21	3062.17	3290.80	0.67			
2652.77	2939.23	0.22	3069.82	3298.34	0.68			
2663.56	2946.62	0.23	3071.95	3242.90	0.69			
2674.25	2954.07	0.24	3084.95	3313.35	0.70			
2684.85	2961.60	0.25	3092.43	3320.82	0.71			
2695.35	2969.20	0.26	3099.87	3328.26	0.72			
2705.76	2976.85	0.27	3107.26	3335.68	0.73			
2716.08	2984.56	0.28	3114.60	3343.06	0.74			
2726.32	2992.32	0.29	3121.96	3350.45	0.75			
2736.47	3000.12	0.30	3129.13	3357.76	0.76			
2746.54	3007.95	0.31	3136.32	3365.06	0.77			
2756.53	3015.81	0.32	3143.47	3372.34	0.78			
2766.44	3023.70	0.33	3150.57	3379.59	0.79			
2776.27	3031.61	0.34	3157.62	3386.81	0.80			
2786.23	3039.78	0.35	3164.63	3394.01	0.81			
2795.70	3047.49	0.36	3171.59	3401.17	0.82			
2805.30	3055.45	0.37	3178.51	3408.31	0.83			
2814.83	3063.42	0.38	3185.39	3415.42	0.84			
2824.28	3071.40	0.39	3192.22	3422.51	0.85			
2833.66	3079.39	0.40	3199.01	3429.56	0.86			
2842.97	3087.37	0.41	3205.75	3436.59	0.87			
2852.21	3095.35	0.42	3212.45	3443.59	0.88			
2861.37	3103.34	0.43	3219.12	3450.56	0.89			
2870.47	3111.32	0.44	3225.74	3457.50	0.90			
2879.50	3119.29	0.45	3232.31	3464.42	0.91			

Table.4 λ, MSE for Training Data and MSE for Testing Data for Ridge Regression, using Gradient Descent Algorithm.

For Training Data MSE value is lowest at $\lambda = 0$ and for Testing data MSE value is lowest at $\lambda = 0.06$. These values are the optimal λ values for respective datasets.

Report 5: Non-Linear Regression

In computational modeling, often times data is encountered which on the first look may not imply a linear relationship between features and targets. In such cases, it may be a viable option to transform the feature set with a function to represent the target concept.

In a simplified way, it can be said, in linear regression,

$$y = X\beta + \varepsilon \quad [4]$$

But, in some cases it is easier to represent this as,

$$y = \varphi(x) \quad [5]$$

Where, $\varphi(x)$ can be any algebraic function which fits with the training set appropriately.

Here, we are trying to model the data with one feature and iterating over multiple φ values, which in this case is essentially X raised exponentially as:

$$\varphi(x) = 1 + x + x^2 + \ldots + x^p$$

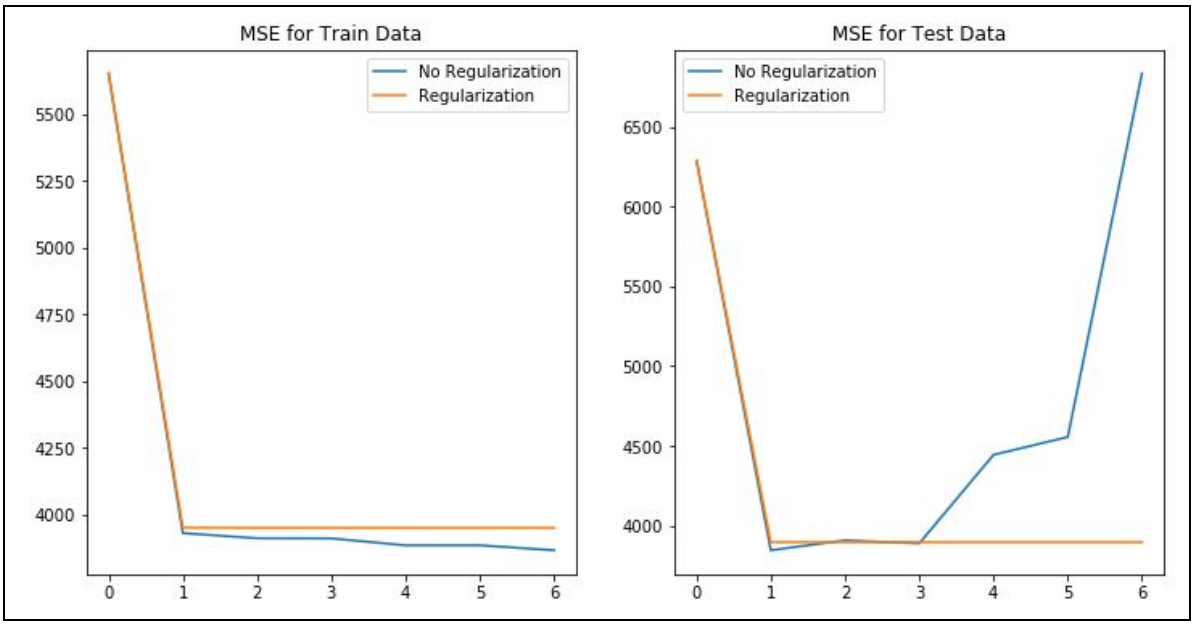

Fig. 4 Degree of polynomial vs. MSE for Training Data and MSE for Testing Data, both using Non-Linear Regression.

The degree of polynomial varies from 0 to 6 and plot also shows lines for error values with regularization and without regularization for these different degree of polynomial. The optimal value $\lambda = 0.06$

From the above two figure plots we can clearly infer that for Training dataset, as the degree of polynomial increases the values for MSE is decreasing. This behavior is observed for both, with and without regularization.

However, that is not the case for Testing data. In fact, after degree = 3 the values of error suddenly shoots up, for the case of without regularization. The reason for this is that with higher degree the curve tries to better fit with the data points, but starts overfitting since it is highly bound to the Training data, so with a different Testing data the error shoots up erratically. To arrest such behavior, we use perform Non-Linear Regression with regularization. This can be seen in the second plot in the above figure.

p	wr	wor
0	5650.71054	5650.71191
1	3930.91541	3951.83912
2	3911.83967	3950.68731
3	3911.18866	3950.68253
4	3885.47307	3950.68234
5	3885.40716	3950.68234
6	3866.88345	3950.68234

Table. 4 Error values with regularization (wr) and without regularization (wor) for different values of p, for training data.

p	wr	wor
0	6286.40479	6286.88197
1	3845.03473	3895.85646
2	3907.12810	3895.58406
3	3887.97554	3895.58272
4	4443.32789	3895.58267
5	4554.83038	3895.58267
6	6833.45915	3895.58267

Table. 5 Error values with regularization (wr) and without regularization (wor) for different values of p, for testing data.

The optimal value is p = 1 for the testing data with regularization and p = 4 without regularization. The error values are the least at these value of polynomial degree in respective case.

Report 6: Interpreting Results

Approach	MSE For Training Data	MSE For Test Data	Remarks
Linear Regression (Without Intercept)	19099.45	106775.36	The most basic linear regression model, passing through the origin, but its use is limited without using a intercept, which can be thought of as a bias or a prior.
Linear Regression (With Intercept)	2187.16	3707.84	A very popular learning model used widely in a lot of different applications
Ridge Regression (With Regularization λ=0.06)	2451.53	2851.33	Essentially same as Linear Regression but with the addition of a regularization parameter to minimize the issue of overfitting. Implementation here is using **[1]**, which is infeasible for large feature set and training set.
Ridge Regression With Gradient Descent	2258.98	3236.87	A very efficient method for modeling, which minimizes the error **[3]** in the model weights with each pass. These values represent the model with no regularization.
Ridge Regression With Gradient Descent (With Regularization λ=0.06)	2451.54	2851.30	Gradient Descent approach for Ridge Regression with optimal regularization.
Non-linear Regression (Without Regularization)	3930.92	3845.03	Useful in cases of non-linear relationship between features. But susceptible to overfitting in peculiar cases.
Non-linear Regression (With Regularization)	3951.84	3895.86	An improvement over above regression, that can avoid overfitting to give a smooth fitting curve bound well to the data points.

The best metric that we can use to choose the best setting from the different approaches is the accuracy value obtained for each of the approaches. We can calculate the accuracy of each approach by comparing the MSE value obtained from the execution of the code.

According to the table given above, the MSE value for Linear Regression with intercept is extremely high and will lead to a lot of errors, so it is an infeasible approach if we consider accuracy as our metric. We see that the Ridge Regression approach provides us with much lower values for MSE which denotes that the error obtained from this approach is the lower. Also, the MSE values for Ridge Regression with Gradient Descent are almost identical to the Ridge Regression values.

However, if we perform regularization using an optimal value for λ, we see that the MSE value is at its lowest, which means that in this case we achieve optimal accuracy results. Similarly for the Non-linear Regression approach, both for with regularization and without regularization, we see that the MSE values are slightly higher than those of the Ridge Regression approach, so we will not choose this approach either.

Therefore, if we were to consider accuracy as the only metric, we will choose the Ridge Regression with Gradient Descent approach, with regularization using optimal value for λ, as it produces the highest accuracy.