Assignment - 3
Report

Team Details:

1. Revanth Mopidevi 2017AAPS0280H
2. A. Shrugnin Reddy 2017B3A80864H
3. Anthareddy Pranay Reddy 2018AAPS0511H

Model:

- By randomly shuffling data 10 polynomials of degree 1 to 10 were built.
- GD and SGD models were built with and without regularization parameters for each degree polynomial and the results were compared.
- Sum of squared errors is taken as the key performance measure for the models without regularisation. $E(w) = \frac{1}{2}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2$ where, $\hat{y}_i (= x_i * w)$ is the predicted value and y_i is the actual value.
- For models with regularisation, the following error function has been used.

$$E(w) = \frac{1}{2}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=0}^{n}|w_i|$$

$$E(w) = \frac{1}{2}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda(\sum_{i=0}^{n}(w_i)^2)$$

- The first equation represents Lasso regularisation and the second equation represents Ridge regularisation.
- Data is splitted into 70:20:10 split for training, validation and testing respectively.

Implementation of algorithms:

- **Gradient Descent (without regularisation)**:
  - Error function is to be minimized and here, first we take random weights and calculate the gradient and decrease it until error change is too small i.e., until the gradient converges to minimum.
  - The weights are initialized to zero.

$$w = w - \tau * \frac{\partial E(w)}{\partial w}$$

- Then, sequentially the weights are updated using the above formula where τ is the learning rate (10^{-5} in our case) to get to the global minimum of the error function.
- These iterations are run until we reach a point where the error change is very less ($<10^{-4}$) and epochs are restricted to 25k.

- **Stochastic Gradient Descent (without regularisation):**
  - Same as GD but the weights are updated by calculating partial differential of error at only one point(batch size=1) whereas in GD we update using partial derivative sum of all points which becomes complex by the increment in dataset size.
  - The weights are initialized to zero before running the algorithm.

$$w = w - \tau * \frac{\partial E(w)}{\partial w}_{w=w_i}$$

  - Cost computation is similar to GD but the difference is, we update the weights at every data point.
  - Learning rate τ is taken as 10^{-7} and termination error change is 10^{-6} and epochs are restricted to 25k.

- **GD and SGD (with regularisation):**
  - These are almost similar to the implementation of GD and SGD as discussed above.
  - The only difference is the addition of the regularisation parameter. This parameter penalises the values of coefficients so that the model does not overfit our data.
  - Five λ values are chosen randomly.
  - For every λ, validation error is calculated and finally the least of these errors is selected.
  - Following regularisations are implemented.

$$E(w) = \frac{1}{2}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=0}^{n}|w_i|$$

$$E(w) = \frac{1}{2}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda(\sum_{i=0}^{n}(w_i)^2)$$

Results:

Method-Results	Minimum	Degree	λ
GD			
Training Error	14.967070	2	0
Testing Error	2.554342	2	0
GD with L1			
Training Error	15.041453	1	0.015396
Testing Error	2.688825	1	0.015396
GD with L2			
Training Error	15.059731	1	0.985359
Testing Error	2.680408	2	0.222962
SGD			
Training Error	36.715794	7	0
Testing Error	5.746477	7	0
SGD with L1			
Training Error	19.249783	2	0.229559
Testing Error	3.614329	2	0.229559
SGD with L2			
Training Error	21.271077	3	0.291099
Testing Error	3.527573	1	0.006887

Questions to Ponder

1. What happens to the training and testing error as polynomials of higher degree are used for prediction?

As the polynomials of higher degree are used, then the training error will decrease for a small sample size, this is because the model is able to fit according to a small number of samples in the training data and predict them perfectly. But, the same model has really high testing error, this is because it is trained to very specific data points and it cannot make generalised predictions for new data.

2.Does a single global minimum exist for Polynomial Regression as well? If yes, justify.

Yes, a single global minimum exists for polynomial regression as well. Any polynomial regression equation can be solved to obtain final weights in order to minimise the error by differentiating w.r.t. the weights w0,w1,....wNN.

3. Which form of regularization curbs overfitting better in your case? Can you think of a case when Lasso regularization works better than Ridge?

Ridge regularisation curbs slightly better overfitting than lasso regression.

Lasso regularisation works better than ridge regularisation when there are a small number of significant parameters and the others are close to zero I.e. only few predictors will properly influence the response.

Ridge regularisation works well if there are many large parameters of about the same value I.e. when most of the predictors impact the response.

4. How does the regularization parameter affect the regularization process and weights? What would happen if a higher value for λ (> 2) was chosen?

When the regularisation parameter is larger, more penalty will be assigned to larger weights for features, this implies that extent of overfitting is inversely proportional to regularisation parameter. This regularisation parameter helps to tackle overfitting.

If a higher lambda value is chosen, high penalties will be assigned to weights of features. As we increase the lambda value, the model starts to underfit the data and slowly the weights of some features will become insignificant I.e. they tend to become zero. For a much higher lambda (> 100), the regression line is almost parallel to the x-axis since only theta zero significantly contributes to the equation. In such a case, testing and training errors are extremely high.

5. Regularization is necessary when you have a large number of features but limited training instances. Do you agree with this statement?

Yes, we agree with this statement. When training samples are of low numbers, the regression will try to fit the data points perfectly and the coefficients of the features become large. Variance will be high, when the same weights are used to make predictions on testing data. Hence, regularisation term is used to penalise the large parameters which leads to overfitting.

6. If you are provided with D original features and are asked to generate new matured features of degree N, how many such new matured features will you be able to generate? Answer in terms of N and D.

Number of features => $(D,N)=(D+N)CN$

where D is the number of original features. N is degree of polynomial.

7. What is bias-variance trade off and how does it relate to overfitting and regularization.

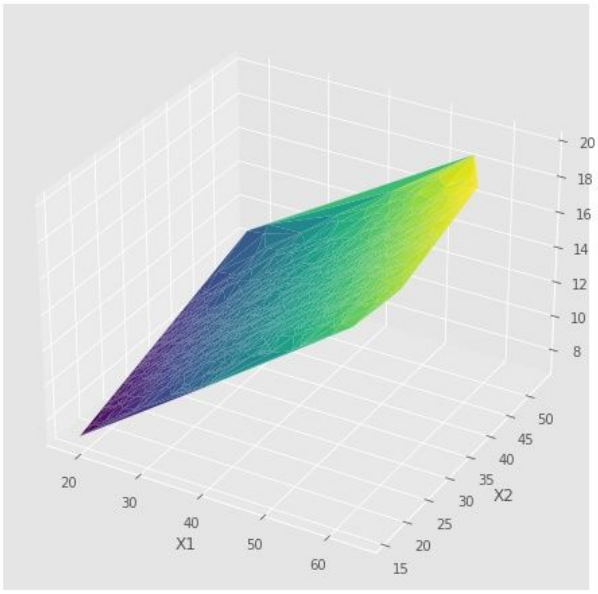
A model with high bias (I.e. training error) and high variance (I.e. testing error) will face the problem of under-fitting.

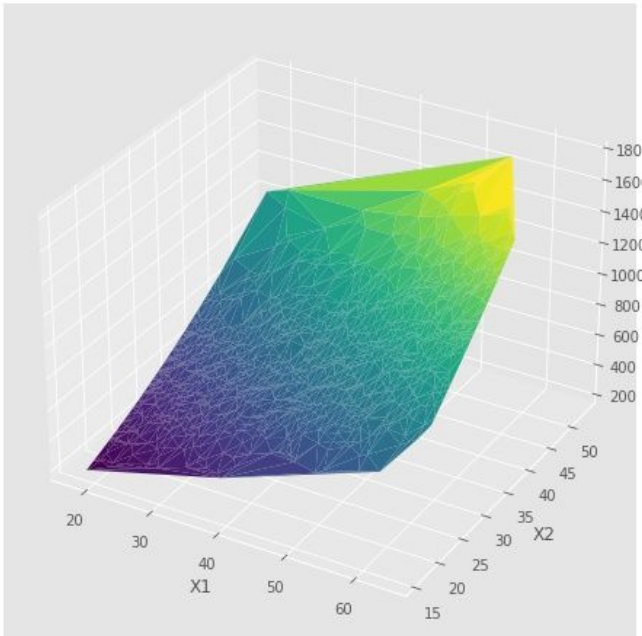
Similarly, the model with low bias and high variance will face the problems of Overfitting.

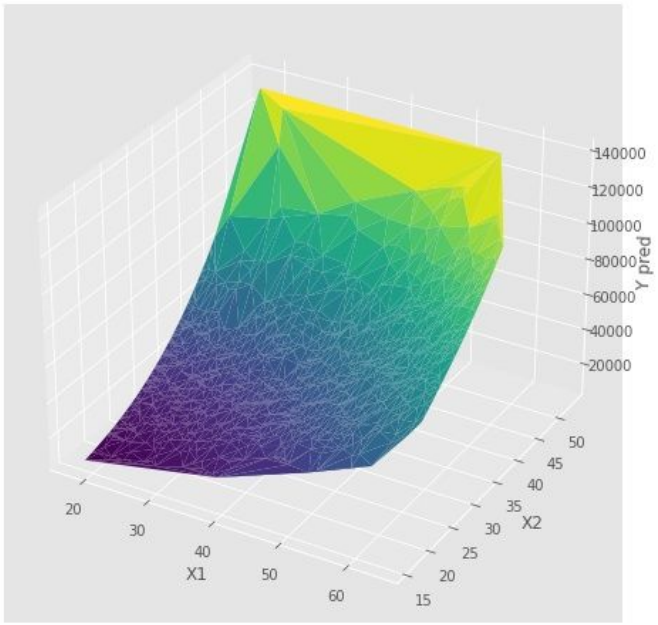
A model with low bias and low variance is always a good model. This is called Bias-Variance trade-off. In Order to arrive at an optimal Bias-Variance trade-off, the degree of polynomial and the regularisation parameter need to be tuned.

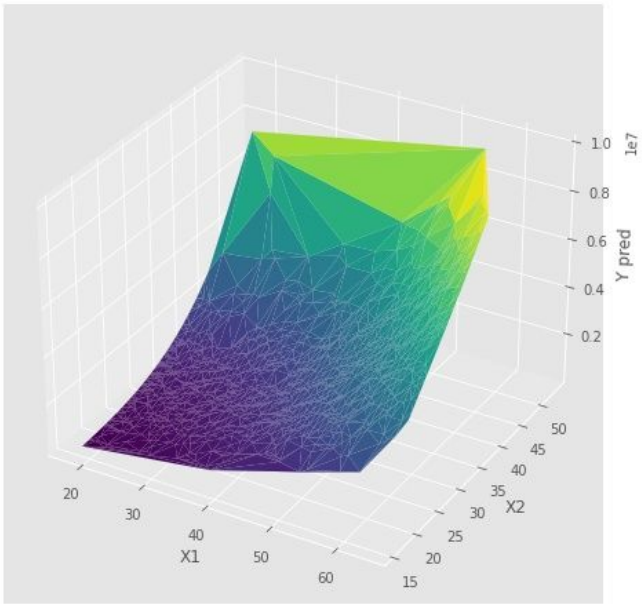
Surface Plots:

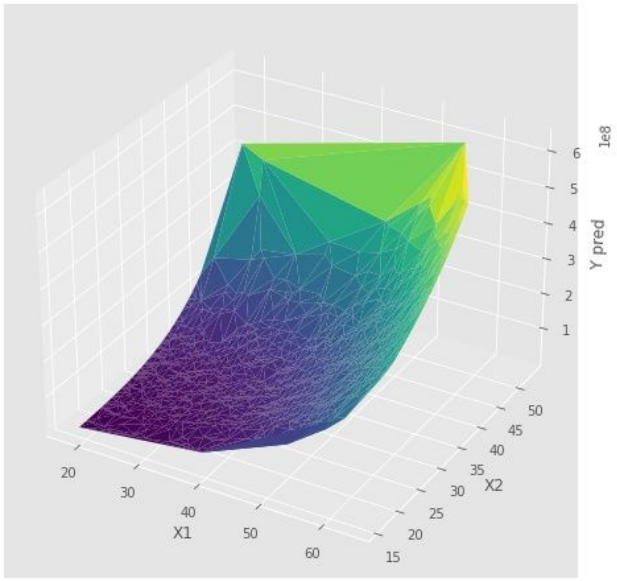
Surface plots of predictions against features are shown below. Axes are age, bmi and charges.These models try to fit our training data perfectly and thus the value of coefficients become very large or very small.

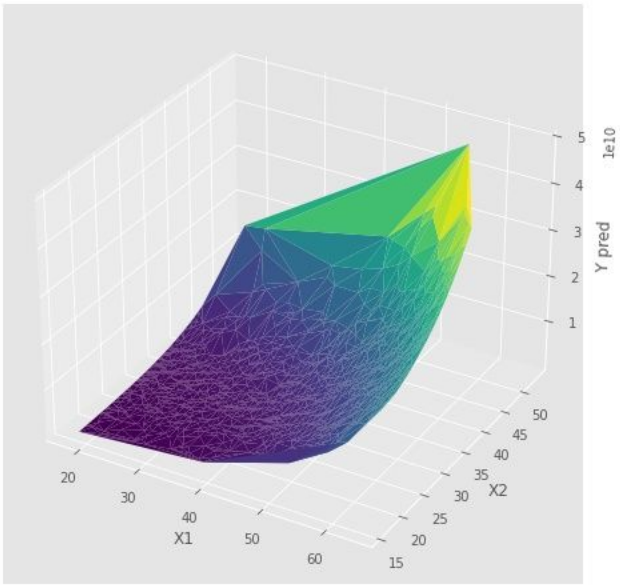

Y pred
X1
X2

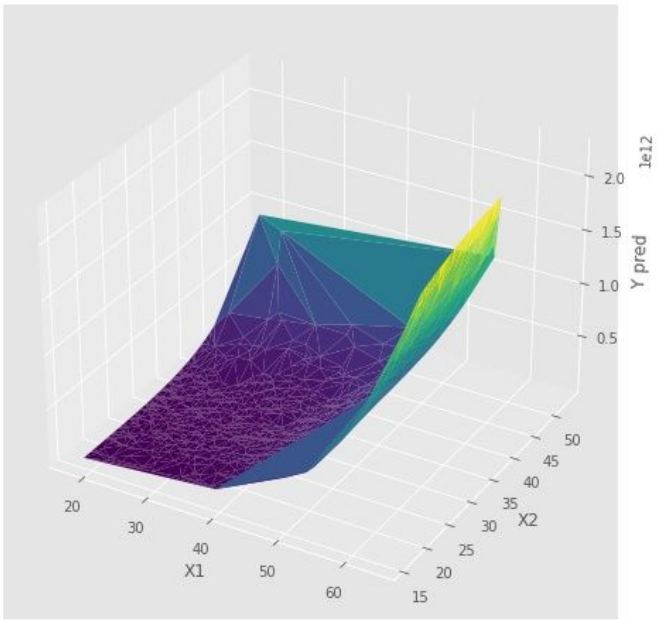

Y pred
X1
X2

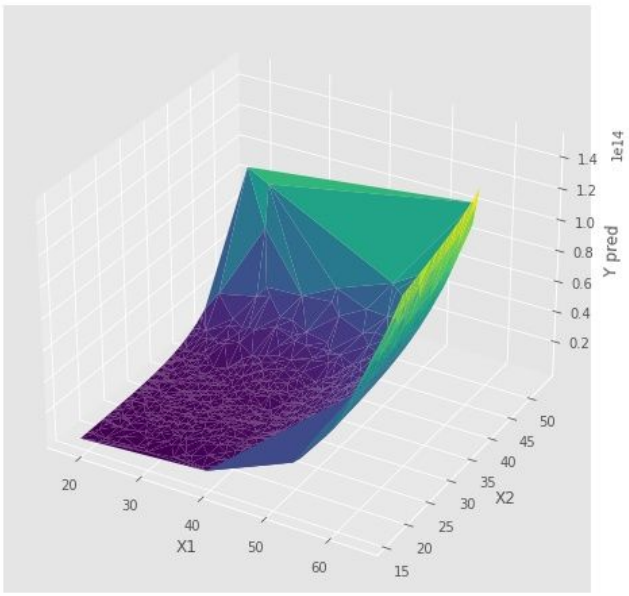

140000
120000
100000
80000
60000
40000
20000
Y pred
20
30
40
50
60
X1
15
20
25
30
35
40
45
50
X2

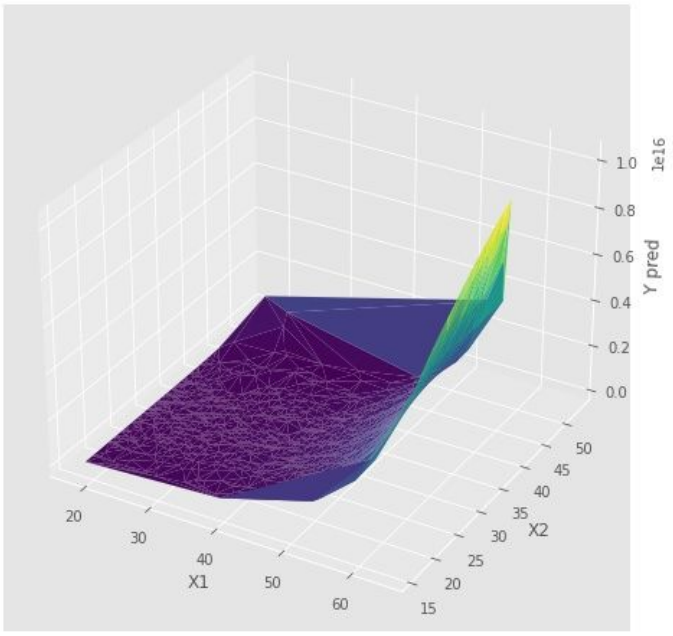

1.0
0.8
0.6
0.4
0.2
1e7
Y pred
20
30
40
50
60
X1
15
20
25
30
35
40
45
50
X2

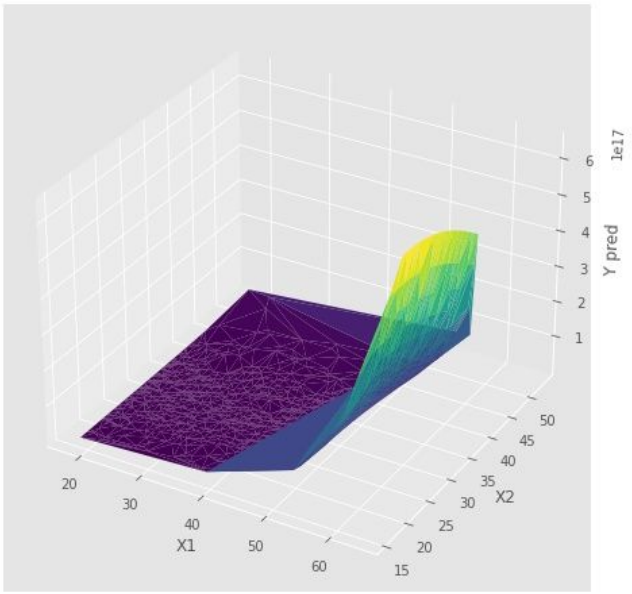

1e8
Y pred
6
5
4
3
2
1
X1
20
30
40
50
60
X2
15
20
25
30
35
40
45
50


1e10
Y pred
5
4
3
2
1
X1
20
30
40
50
60
X2
15
20
25
30
35
40
45
50


1e12
2.0
1.5
1.0
0.5
Y pred
X1
20
30
40
50
60
X2
15
20
25
30
35
40
45
50


1e14
1.4
1.2
1.0
0.8
0.6
0.4
0.2
Y pred
X1
20
30
40
50
60
X2
15
20
25
30
35
40
45
50


1e16
1.0
0.8
0.6
0.4
0.2
0.0
Y pred
X1
20
30
40
50
60
X2
15
20
25
30
35
40
45
50


1e17
6
5
4
3
2
1
Y pred
X1
20
30
40
50
60
X2
15
20
25
30
35
40
45
50

The above plots are for the various polynomial models of degree 1 – 10 in order respectively. Higher degree models were overfitting our data.