Design Document-

Made By – Anusha Agrawal (2018A3PS0032H), Kriti Jethlia (2018A7PS0223H), Tanay Gupta (2018AAPS0343H)

Pre-Processing-

- Splitting Data – The data is randomly split into 70-20-10 train validation test dataset.

- Standardization – Data is standardized to scale it so that gradient descent will work better. We only standardize the input features as standardizing the output features will only scale the loss value and affect the shape of the loss vs parameters curve. Also mean and variance of train data is calculated and this mean and variance is used for the test data.

- Features can be generated using sklearn's PolynomialFeatures function.

Model-

- We are using polynomial linear regression to fit the data. The equation used by a 2nd degree model is :

  - $Y = w0 + w1x1 + w2x2 + w3x1^2 + w4x1x2 + w5x2^2$

- Degrees from 1 to 10 have been tried out, with and without regression.

Implementation-

- Gradient Descent –
  - Features are generated based on the degree of the function.
  - Sklearn's function adds an extra column of one for the bias term.
  - Y_pred = X.Theta and loss = Y_pred-Y
  - We calculate average d_theta = 1/m*(XT.loss)
  - Theta = Theta – learning_rate*d_theta
  - These calculations are repeated for many epochs.

- Gradient Descent with Regularization –

  - Cost function has an extra term for regularization. For L1 regularization, absolute value of weights, multiplied by a constant lambda is added to the weight. Various

values are tried out for lambda in the range of 0 to 1 and the best performing model on the validation set is picked.

☐ For L2 regularization, squared value of weights, multiplied by a constant lambda is added to the weight. Various values are tried out for lambda in the range of 0 to 1 and the best performing model on the validation set is picked.

- Stochastic Gradient Descent –
  - ☐ An extra column is added to X for the bias term.
  - ☐ y_pred = X.Theta here the difference is that we use only one training example(x) instead of all the training examples(X) and subsequently the loss and d_theta is also for it.
  - ☐ d_theta =   (XT.loss)
  - ☐ Theta = Theta – learning_rate*d_theta

  - ☐ These calculations are repeated for many epochs.

- Stochastic Gradient Descent with Regularization –

☐ Cost function has an extra term for regularization. For L1 regularization, absolute value of weights, multiplied by a constant lambda is added to the weight. Various values are tried out for lambda in the range of 0 to 1 and the best performing model on the validation set is picked.

☐ For L2 regularization, squared value of weights, multiplied by a constant lambda is added to the weight. Various values are tried out for lambda in the range of 0 to 1 and the best performing model on the validation set is picked.

Minimum Errors for different degrees –

Gradient Descent –

Degree	Training rmse (* 10000)	Testing rmse (* 10000)
1	1.579400760236879	1.4827584193152368
2	1.2495471002878265	1.2559843542154494
3	2.478605195828301	2.352711526479722

4	2.327262929760502	2.213166971203762
5	2.1551075043112564	2.1168932176686535
6	1.5367855702529012	1.608558935532338
7	1.3316897760191766	1.3023257918605544
8	1.4988177747443923	1.5956758175210135
9	1.231115951150244	1.2985144981314558
10	1.01373893628671	0.9323523820129203

With L1 regularization

Degree	Optimal lambda	Average test error (* 10000)	Minimum Training error (* 10000)	Minimum validation error (* 10000)
1	0.5	0.6119092	0.66406157	0.593619
2	0.3	0.003209	0.0034791	0.003521
3	0.3	0.00277	0.0026029	0.00252
4	0.1	0.00129652	0.0018644	0.00204
5	0.9	0.00066132	0.000902	0.00078
6	0.1	0.00305	0.00242	0.00245
7	0.5	0.0011111	0.00102892	0.00111507
8	0.3	0.00235876	0.00109	0.0013
9	0.9	0.00438685	0.004789	0.004856
10	0.3	0.00761693	0.00618966	0.00580445

With L2 regularization

Degree	Optimal lambda	Average test error (* 10000)	Minimum Training error (* 10000)	Minimum validation error (* 10000)

1	0.1	0.64796525	0.6645054	0.59373
2	0.1	0.00215809	0.003022	0.00303966
3	0.3	0.00246034	0.002686	0.0022737
4	0.9	0.00344785	0.0033282	0.00343153
5	0.7	0.00132709	0.00167299	0.0016888
6	0.3	0.001607	0.00139827	0.001727
7	0.3	0.00198353	0.001713	0.00138107
8	0.5	0.0126305	0.00290352	0.0031299
9	0.5	0.00285058	0.00233888	0.002359
10	0.1	0.0048733	0.00447597	0.00410723

Stochastic Gradient Descent -

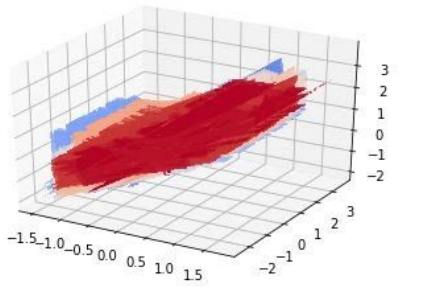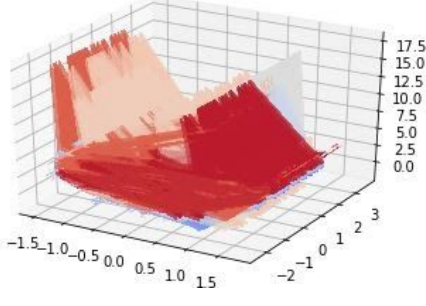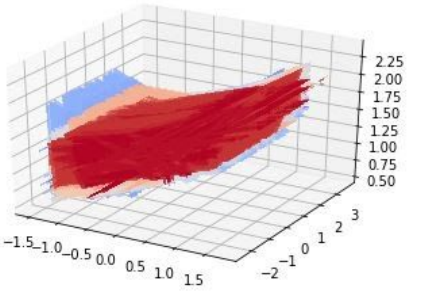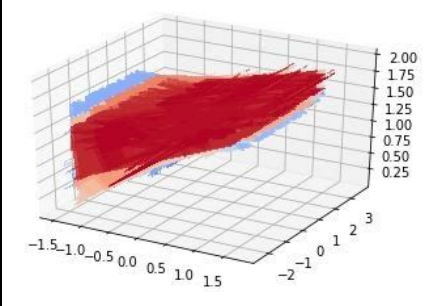| Degree	Training rmse (* 10000)	Testing rmse (* 10000)
1	1.1183214938853379	1.1819166653541942
2	1.1450466890765711	1.1202880059225857
3	1.1454761723993732	1.116604744770453
4	1.1514088310582482	1.1277214579402575
5	1.1439460780457227	1.1688818168719841
6	1.1932012451746632	1.1731170169958423
7	1.1146703622242022	1.225660699831726
8	1.1420582140864457	1.158067760883009
9	1.1125792734843556	1.4813955006505357
10	1.1094809373685215	1.6429280156168853
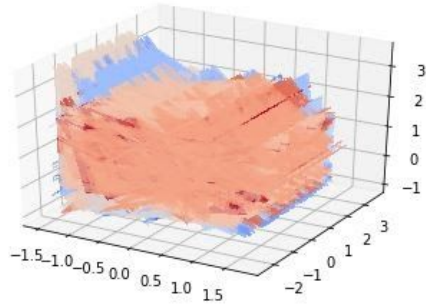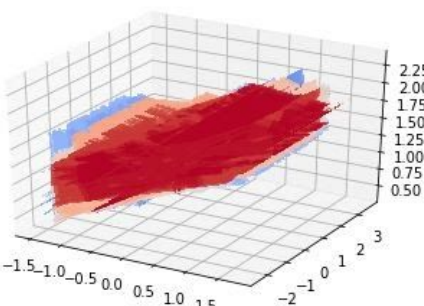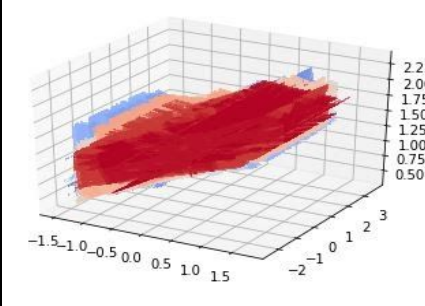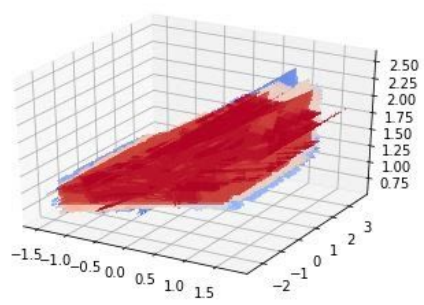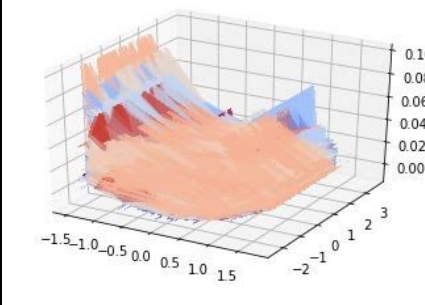
With L1 regularization

Degree	Optimal lambda	Average test error (* 10000)	Rmse (* 10000)	Minimum validation error (* 10000)
1	0.1	1.5943753360	1.56097920157	1.5695234998
2	0.1	1.5937699446	1.56092427698	1.56916522262
3	0.1	1.5917908173	1.55991473555	1.56756218145
4	0.1	1.5902493113	1.55915767102	1.56614864895
5	0.1	1.5875507812	1.55737939767	1.56365014947
6	0.3	1.6217025286	1.52519335589	1.54224036961
7	0.1	1.7386834700	1.52957771816	1.60664637707
8	0.1	1.4923200643	1.55876615790	1.58901079569
9	0.1	1.4891047535	1.55605413032	1.58565332575
10	0.2	1.4863206346	1.55351247983	1.58320537368

With L2 regularization

| Degree	Optimal lambda	Average test error (* 10000)	Rmse (* 10000)	Minimum validation error (* 10000)
1	0.1	1.34503234195	1.3275194785	1.33424570861
2	0.1	1.34511193822	1.3265147509	1.33812076751
3	0.1	1.33739433806	1.3215714820	1.33093499581
4	0.1	1.334130341425	1.32231322894	1.33921892423
5	0.325	1.338126416280	1.32569756649	1.36590404795
6	0.1	1.399431674487	1.35998240885	1.34164806889
7	0.1	1.515771777806	1.29902210929	1.371860939871
8	0.1	1.251033557404	1.31572901279	1.36415789304
9	0.1	1.299992310709	1.31053883536	1.43029389798
10	0.325	1.403497609499	1.31189144810	1.531135651737

Plots –

Degree	Gradient Descent	L1 regularization	L2 regularization
1			
2			
3			

4			
5			
6			
7			

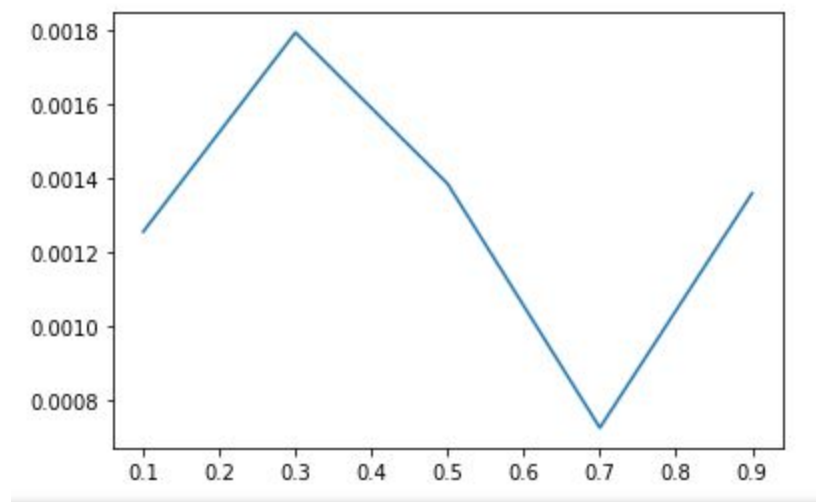| 8			
9			
10			
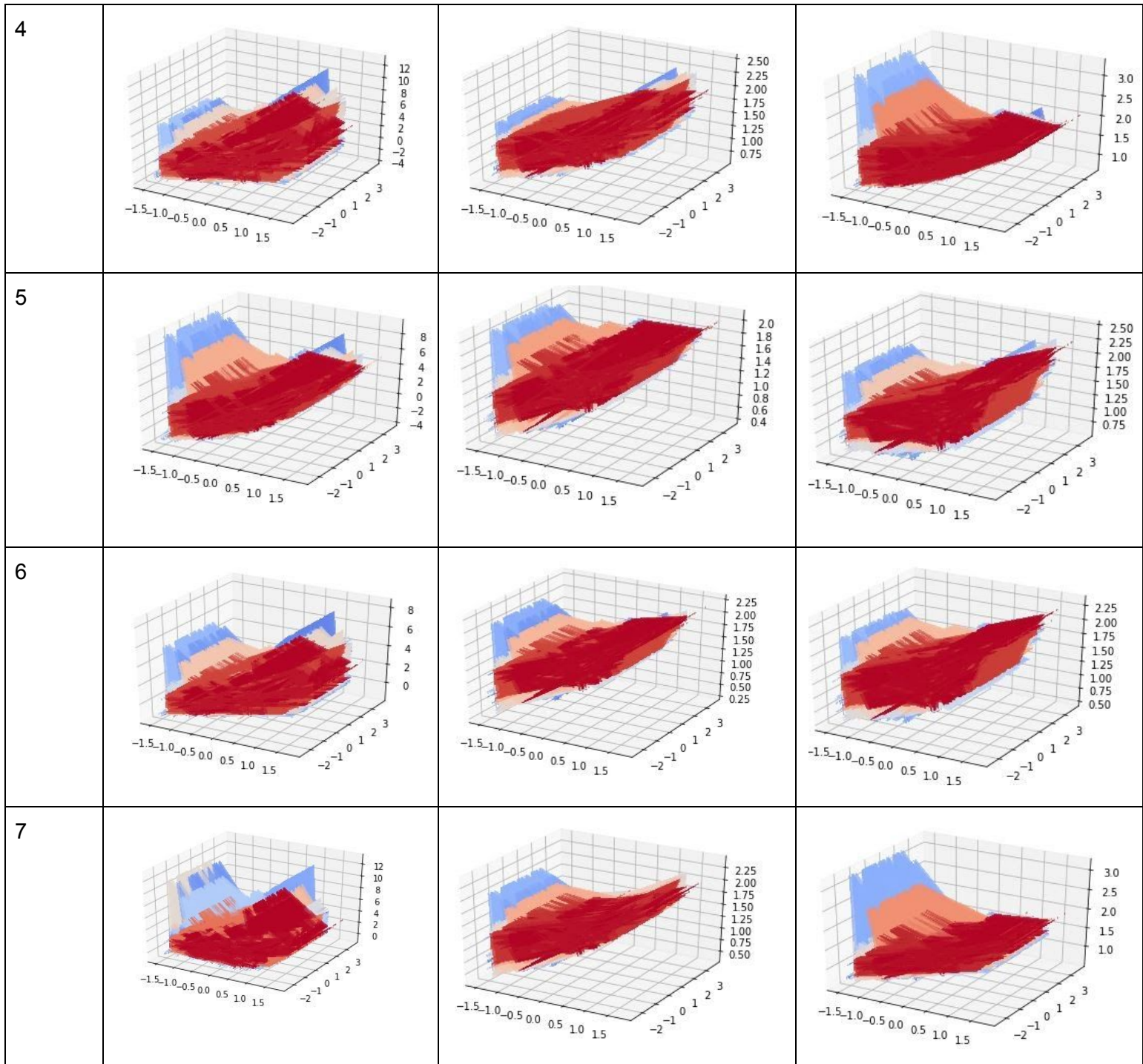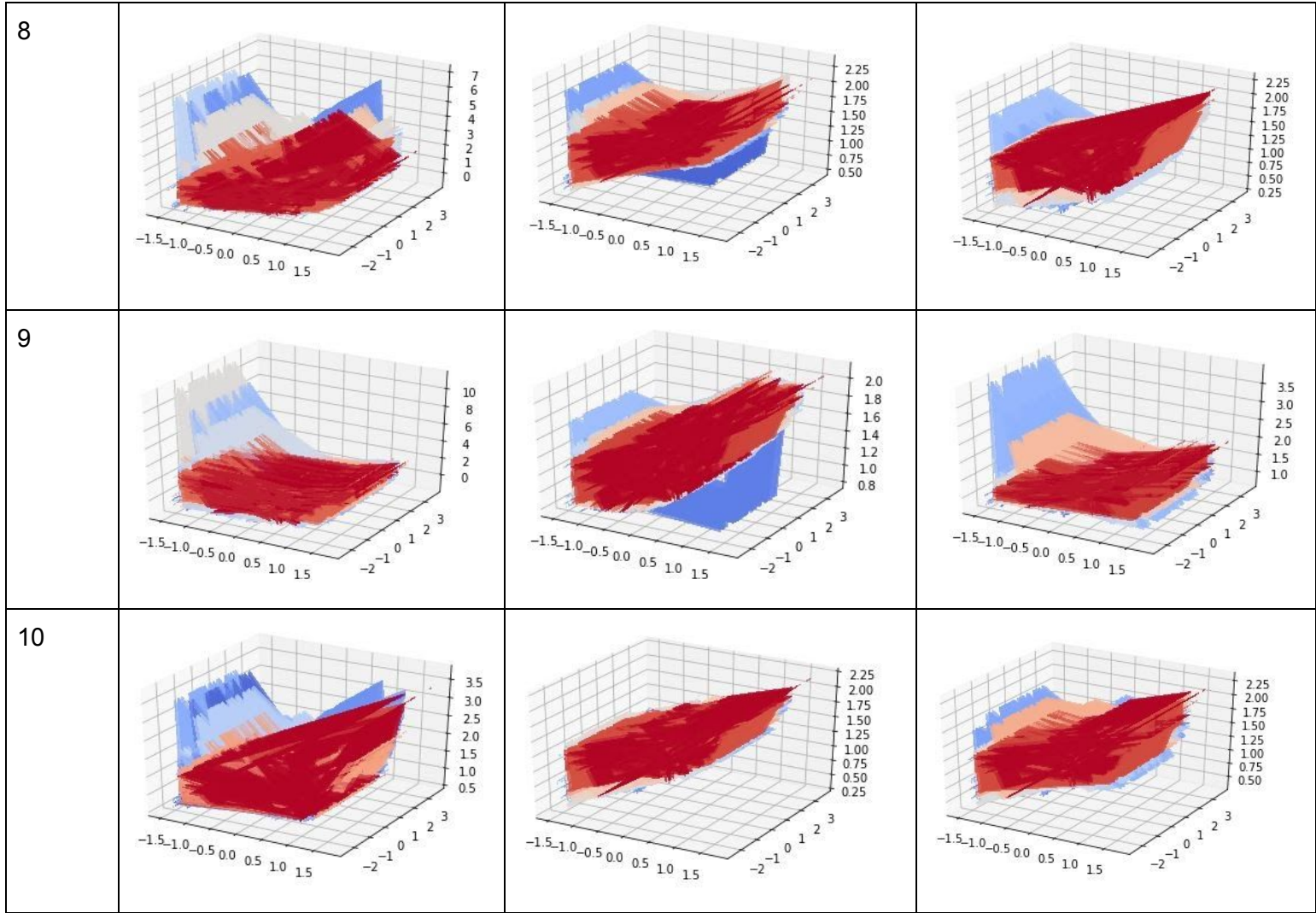
Errors vs Lambdas
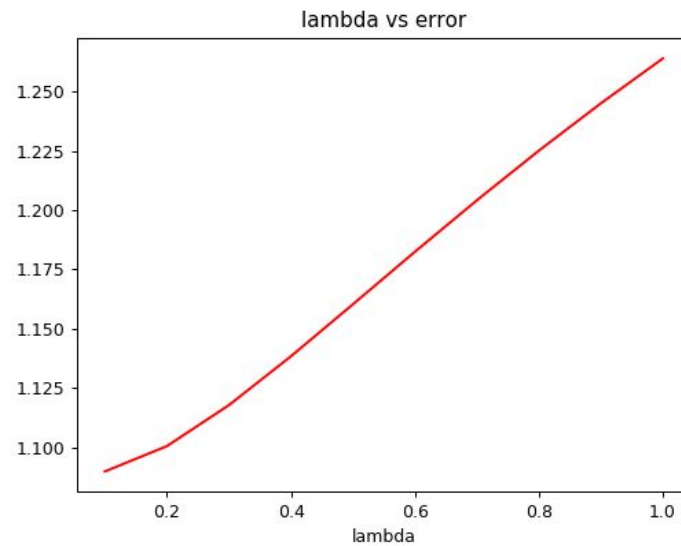
For ridge regression stochastic gradient descent polynomial degree 1

Questions –

Q1 .What happens to the training and testing error as polynomials of higher degree are used for prediction?

 Ans :As the degree of polynomial increases the training error generally decreases for a small sample size since the model is able to fit according to the small number of samples in the training data and perfectly predict them. But the trained model might have high testing error

because it is now tested for very different data points and might not be able to make predictions correctly for testing data. It might be an overfit model.

Q2. Does a single global minimum exist for Polynomial Regression as well? If yes, justify.

Ans : Yes a single global minimum exists for Polynomial Regression as well. Any polynomial regression equation can be solved to obtain the final weights to minimize the error by differentiating w.r.t the weights $w_0, w_1, ... w_{11}, w_{12},,,, w_{NN}$. We are generating new features when we increase the degree and hence this problem is the same as linear regression just with a greater number of features.

Q3 .Which form of regularization curbs overfitting better in your case? Can you think of a case when Lasso regularization works better than Ridge?

Ans : Ridge regularization had slightly better results as compared to lasso regularization. Lasso tends to do well if there are a small number of significant parameters and the others are close to zero (when only a few predictors actually influence the response). Ridge works well if there are many large parameters of about the same value (when most predictors impact the response). Lasso regularization is used for feature selection.

Q4.How does the regularization parameter affect the regularization process and weights? What would happen if a higher value for λ (> 2) was chosen?

Ans: λ is the balancing factor between the weightage given to sum of squares and weights' growth. Small values of λ give more freedom to weights to grow large. Larger the λ, more penalty is assigned to larger weights for features, hence, the extent of overfitting is inversely related to the value of regularization parameter. This regularization parameter thus, helps to tackle overfitting.
For a high lambda value, high penalties are assigned to the weights for the features. As lambda is increased, the regression model starts to underfit the data and slowly the weights of some features become insignificant . For a really high λ (>100), the regression line is almost parallel to the x-axis since only theta zero significantly contributes to the equation. Testing and training errors are extremely high in such a case.

Q5. Regularization is necessary when you have a large number of features but limited training instances. Do you agree with this statement?
Ans : Yes,regularization is necessary when you have a large number of features but limited training instances because our regression model tries to fit perfectly to these data points and thus the coefficients of the features also become large. When the same weights are used to make predictions on   testing data, variance is very high. Hence, a regularization term is

introduced to penalize the large parameters that lead to overfitting.

Q6. 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.
 Ans :Number of features for (D,N) is $^{(D+N)}C_{(D)}$ where D is the number of the original features and N is the degree of the polynomial.

Q7 . What is bias-variance trade off and how does it relate to overfitting and regularization.
 Ans : Models with low bias and high Variance are usually Overfitting models.The model with a high bias (training error) and low variance (testing error) is Under-fitting. A good model would always have low Bias and low Variance. If bias increases, variance decreases.This is called the Bias-Variance trade-off. The degree of the polynomial and the regularization parameters need to be tuned to arrive at the optimal bias-variance trade-off. Regularization helps in overcoming overfitting by restricting the values of parameters.