

Foundations of Data Science (CS F320)

Assignment – 1

BITS Pilani- Hyderabad Campus

Members:

- Kshitij Upadhyay - 2019A7PS0105H
- Akshat Agrawal - 2019AAPS0264H
- Umesh Kumar A- 2019A4PS0868H

Brief Description Of The Model

Model Used:

The model used for our project is Polynomial Regression, and the optimization algorithms used are Stochastic Gradient descent and Gradient descent

Polynomial Regression:

Polynomial regression is a form of linear regression in which the data is fitted with a polynomial equation that has a curved relationship between the target and independent variables. The value of the target variable fluctuates in a non-uniform manner with regard to the predictor in a curvilinear connection (s). In a Linear Regression, We have the following equation with a single predictor:

$$Y = \theta_0 + \theta_1 x$$

where,

Y is the target,

x is the predictor,

θ_0 is the bias,

and θ_1 is the weight in the regression equation

This linear equation can be used to represent a linear relationship. But, in polynomial regression, we have a polynomial equation of degree n represented as:

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \dots + \theta_n x^n$$

Here:

θ_0 is the bias,

$\theta_1, \theta_2, \dots, \theta_n$ are the weights in the equation of the polynomial regression,

and n is the degree of the polynomial

The number of higher-order terms increases with the increasing value of n , and hence the equation becomes more complicated.

Regularization

We have regularized our model with both LASSO and RIDGE regression.

RIDGE Regularization:

L2-norm:

$$J(w) = \frac{1}{2} \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2} \sum_{j=0}^n w_j^2$$

One instance:

$$J(w) = \frac{1}{2} (h_w(x) - y)^2 + \frac{\lambda}{2} \sum_{j=0}^n w_j^2$$

$$\frac{\partial J(w)}{\partial w_j} = (h_w(x) - y) \frac{\partial}{\partial w_j} \left(\sum_{j=0}^n w_j x_j \right) + \lambda w_j$$

$$= (h_w(x) - y) x_j + \lambda w_j$$

All instances

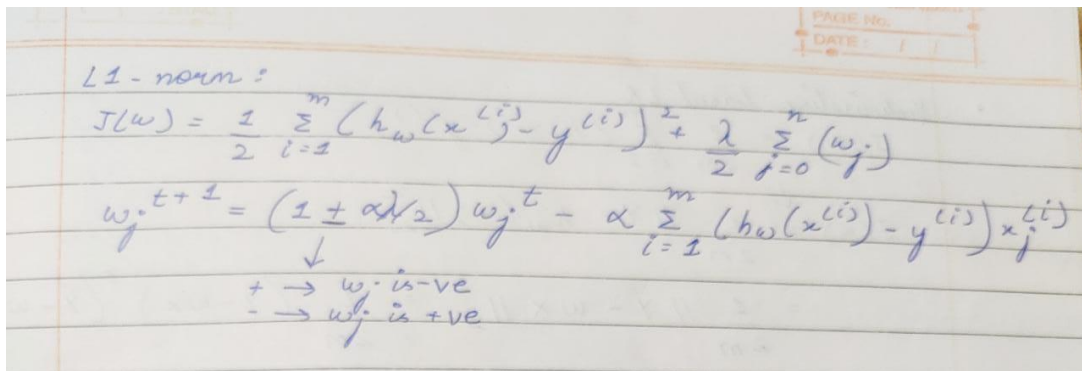
$$\frac{\partial J}{\partial w_j} = \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)}) x_j^{(i)} + \lambda w_j$$

$$w_j^{t+1} = w_j^t - \alpha \frac{\partial J}{\partial w_j}$$

$$w_j^{t+1} = w_j^t - \alpha \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)}) x_j^{(i)} - \alpha \lambda w_j^t$$

$$w_j^{t+1} = (1 - \alpha \lambda) w_j^t - \alpha \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

LASSO Regularization:



Handwritten notes on a piece of lined paper. At the top right, there is a small box with 'PAGE No.' and 'DATE: / /'. The text 'L1 - norm:' is written in blue ink. Below it, the cost function $J(w) = \frac{1}{2} \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2} \sum_{j=0}^n (w_j)$ is written. Below the cost function, the update rule $w_j^{t+1} = (1 \pm \alpha/2) w_j^t - \alpha \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)}) x_j^{(i)}$ is written. Under the term $(1 \pm \alpha/2)$, there is a downward arrow pointing to two lines: '+ $\rightarrow w_j$ is -ve' and '- $\rightarrow w_j$ is +ve'.

L1 - norm:

$$J(w) = \frac{1}{2} \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2} \sum_{j=0}^n (w_j)$$
$$w_j^{t+1} = (1 \pm \alpha/2) w_j^t - \alpha \sum_{i=1}^m (h_w(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

↓

+ $\rightarrow w_j$ is -ve

- $\rightarrow w_j$ is +ve

Part A

Below we have tabulated the minimum training and testing error achieved by our model by using polynomials of degrees 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 to predict the output. We have further visualized the surface plots of our predictions (using matplotlib and Axes3D) that you obtained by using polynomials of varying degrees and have commented on how overfitting actually works.

Stochastic gradient descent

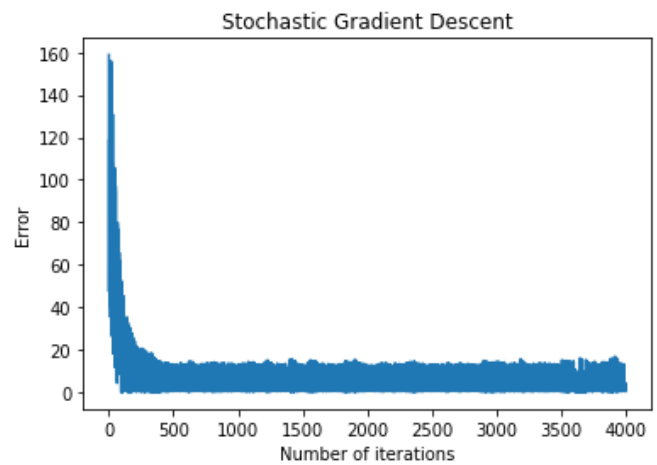
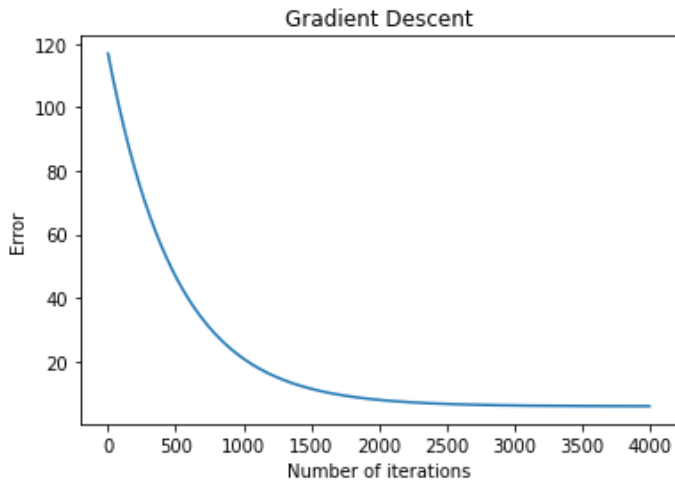
Degree	Evaluation Metrics on Training set			Evaluation Metrics on Testing set		
	MSE	MAE	correlation_ coefficient	MSE	MAE	correlation_ coefficient
0	11.6028	2.9887	0	11.9643	3.0374	0
1	2.3885	1.2408	0.8908	2.6489	1.3034	0.8873
2	2.3940165	1.225379	0.891145	2.737782	1.31065	0.88590
3	2.8290613	1.346392	0.87350	2.944453	1.38792	0.8797429
4	1.5870398	6.320259	0.428787	1.175053	5.3923954	0.4857684
5	203.96018	13.846758	0.200702	211.14527	14.098379	0.189436
6	184.17260	12.76490	0.074184	189.55900	12.974577	0.0574921
7	185.44665	12.80868	0.038776	191.09682	13.04020	0.010920
8	177.93720	12.27902	-0.040369	181.66585	12.401633	-0.026286
9	261900767 9314.8364	357239.74 921837135	-0.26970	157459326 2852.2366	294506.95 87426601	-0.289816

Gradient descent

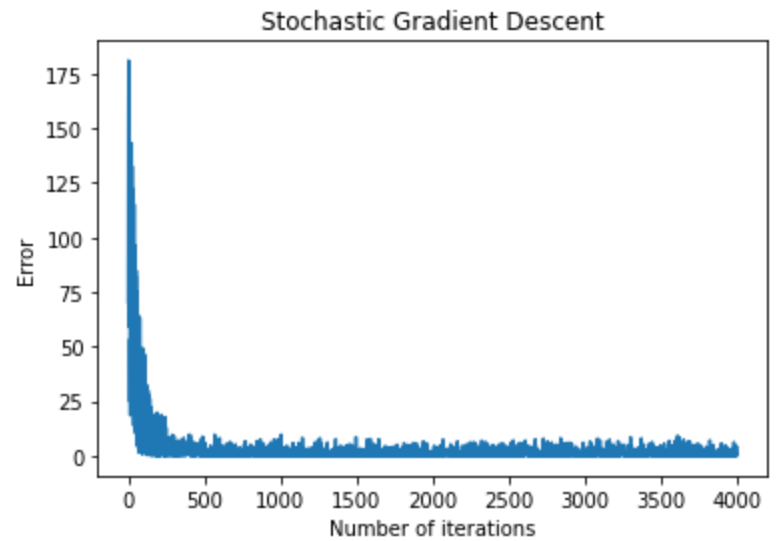
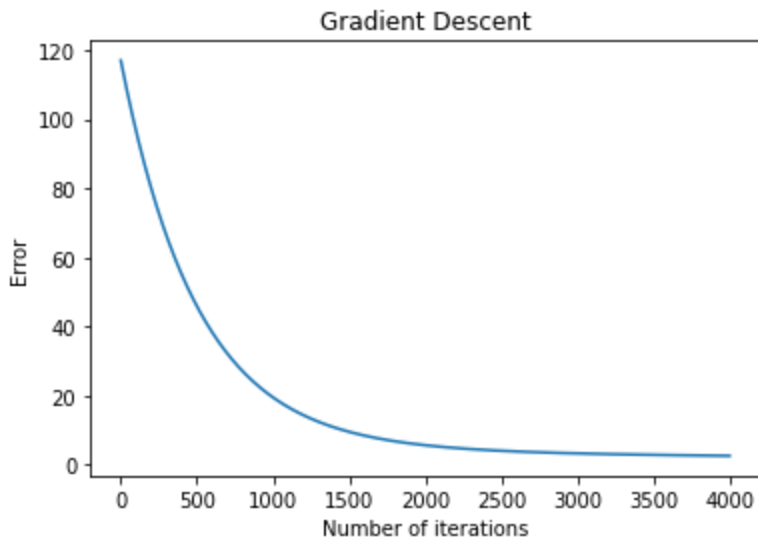
Degree	Evaluation Metrics on the Training set			Evaluation Metrics on Testing set		
	MSE	MAE	correlation_ coefficient	MSE	MAE	correlation_ coefficient
0	11.6118	2.9764	0	12.1830	3.0549	0
1	5.1138	1.9260	0.8621	5.3841	1.9835	0.8883
2	8.0408	2.3174	0.6210	8.5122	2.4121	0.6409
3	7.4366	2.1754	0.6750	8.0173	2.2714	0.6829
4	8.1597	2.2950	0.6285	8.8857	2.3928	0.5984
5	147.5548	11.1101	0.1223	150.1271	11.2573	0.1010
6	145.7597	10.9582	0.1110	148.5640	11.0304	0.1247
7	145.0896	10.8939	0.1288	147.8378	10.9807	0.1421
8	138.30653	10.36130	0.123634	141.91144	10.458250	0.08045
9	137.73673	10.326138	0.159511	140.33923	10.3652	0.12440

Plots

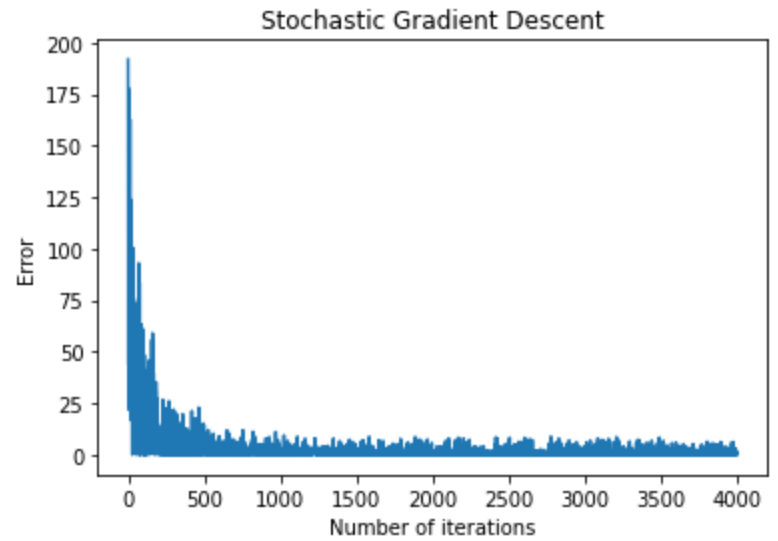
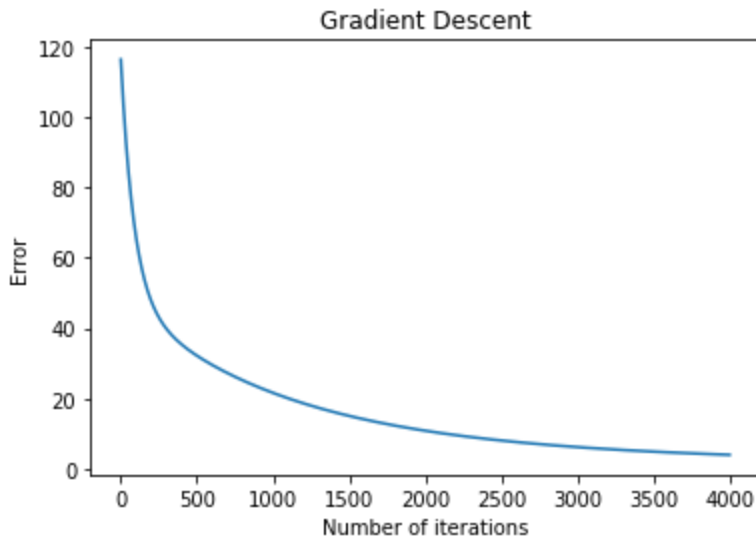
FOR 0 DEGREE POLYNOMIAL



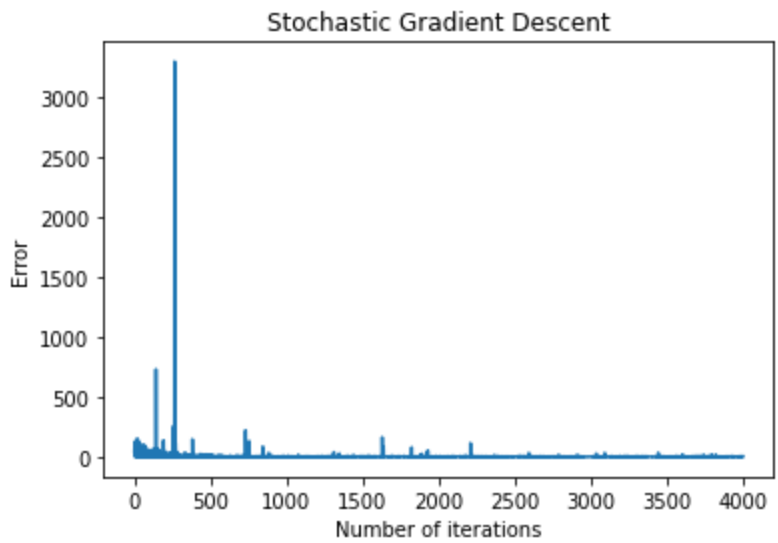
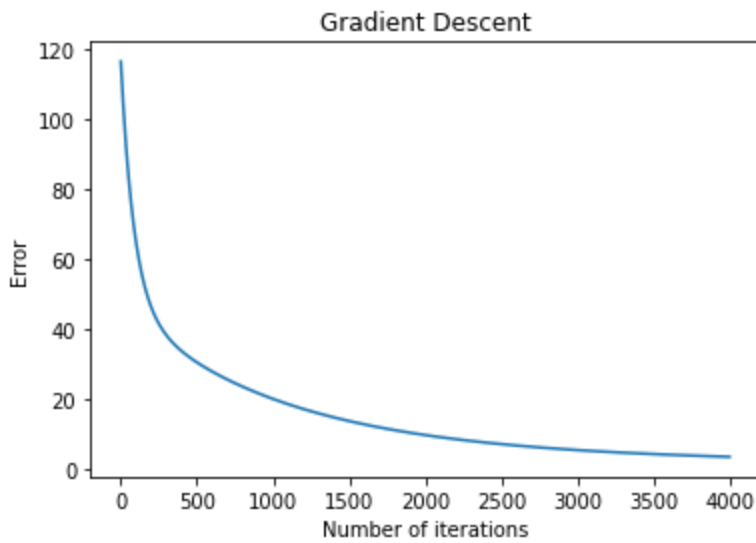
FOR 1 DEGREE POLYNOMIAL



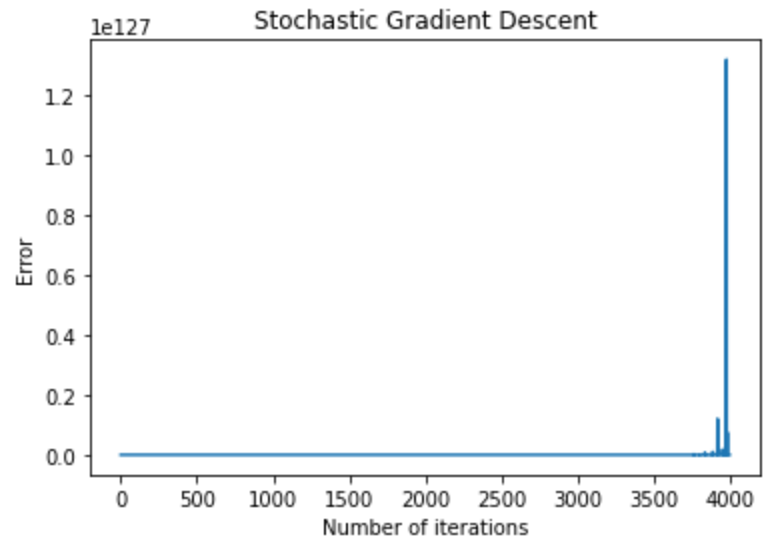
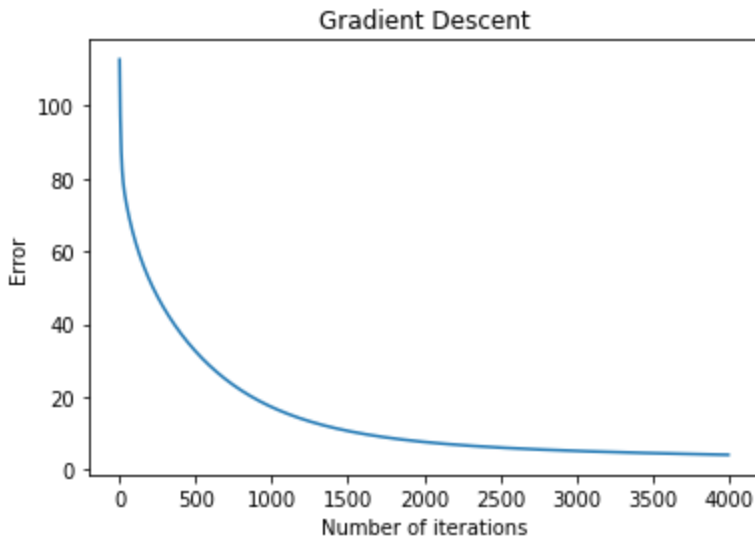
FOR 2 DEGREE POLYNOMIAL



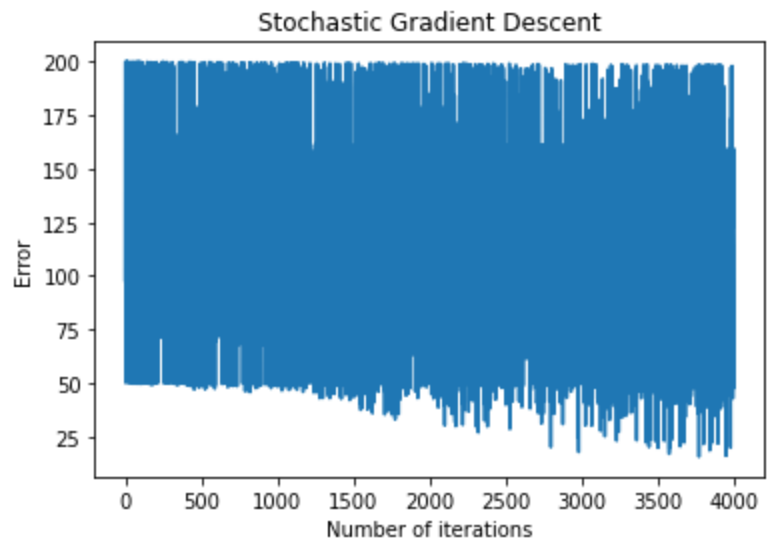
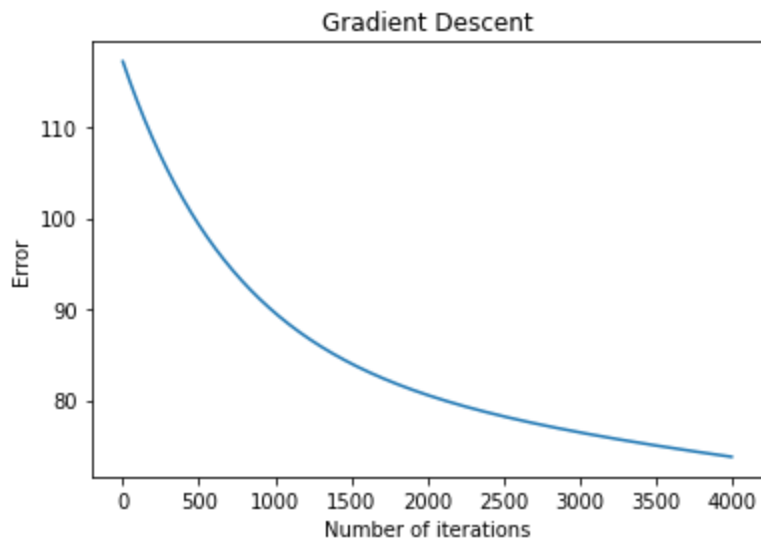
FOR 3 DEGREE POLYNOMIAL



FOR 4 DEGREE POLYNOMIAL

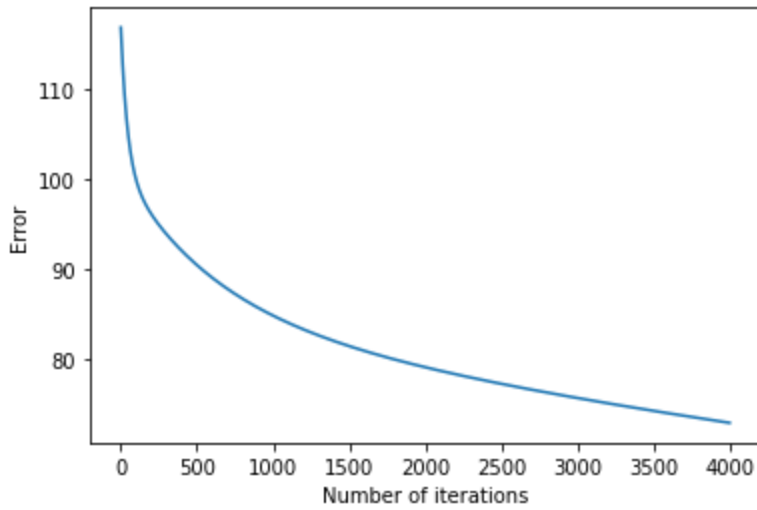


FOR 5 DEGREE POLYNOMIAL

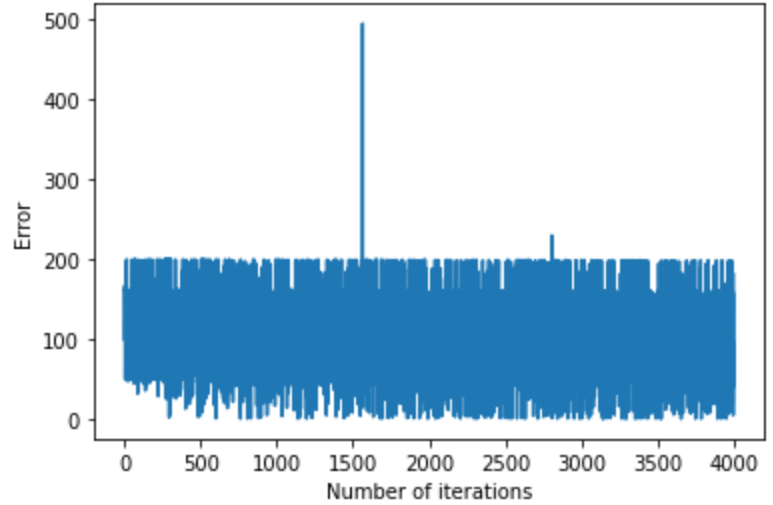


FOR 6 DEGREE POLYNOMIAL

Gradient Descent

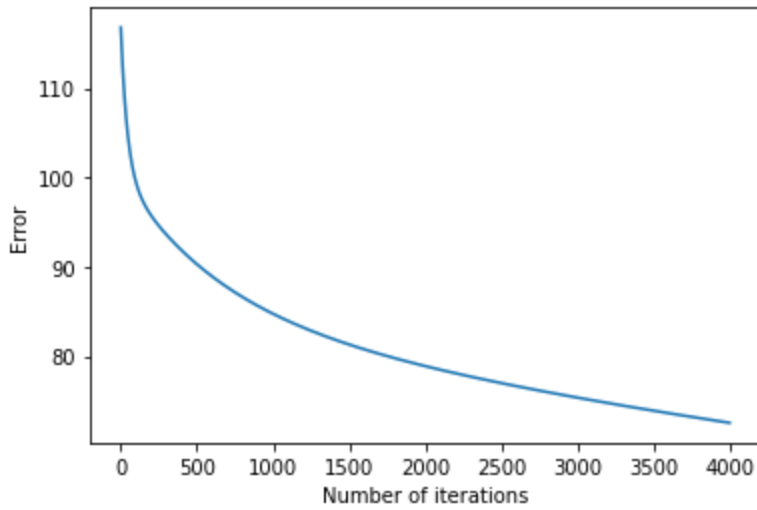


Stochastic Gradient Descent

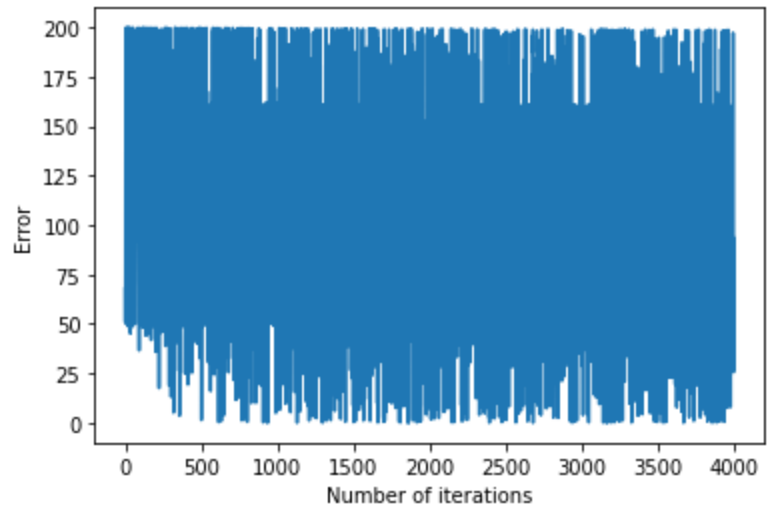


FOR 7 DEGREE POLYNOMIAL

Gradient Descent

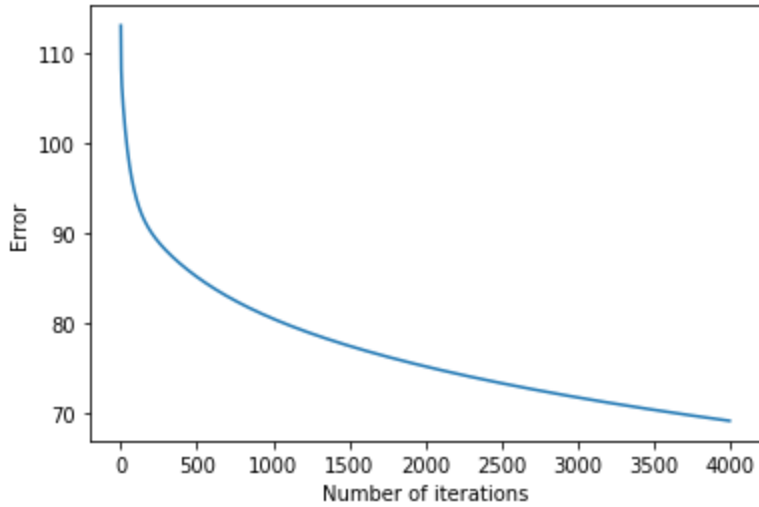


Stochastic Gradient Descent

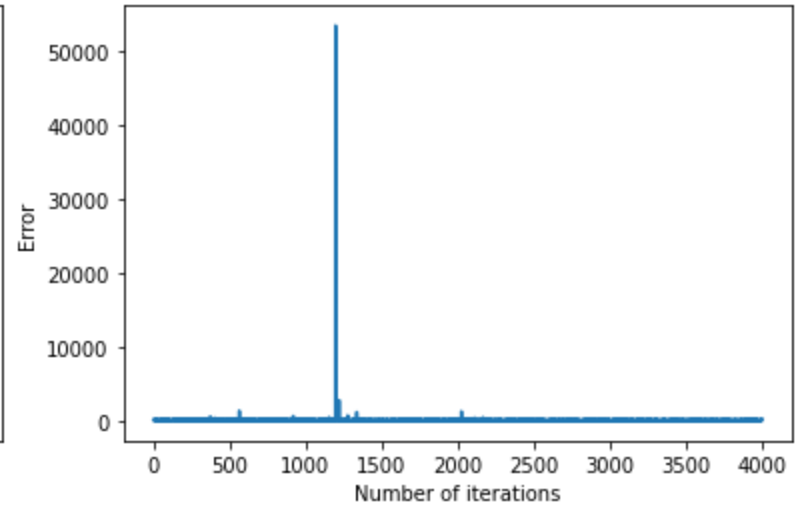


FOR 8 DEGREE POLYNOMIAL

Gradient Descent

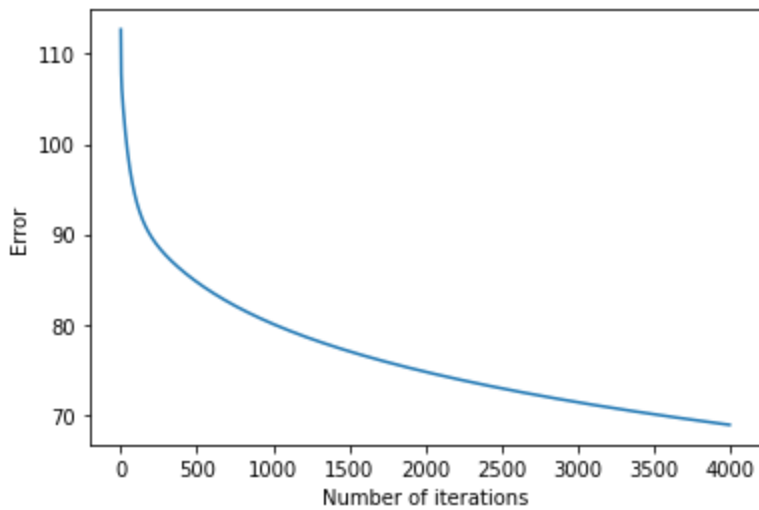


Stochastic Gradient Descent

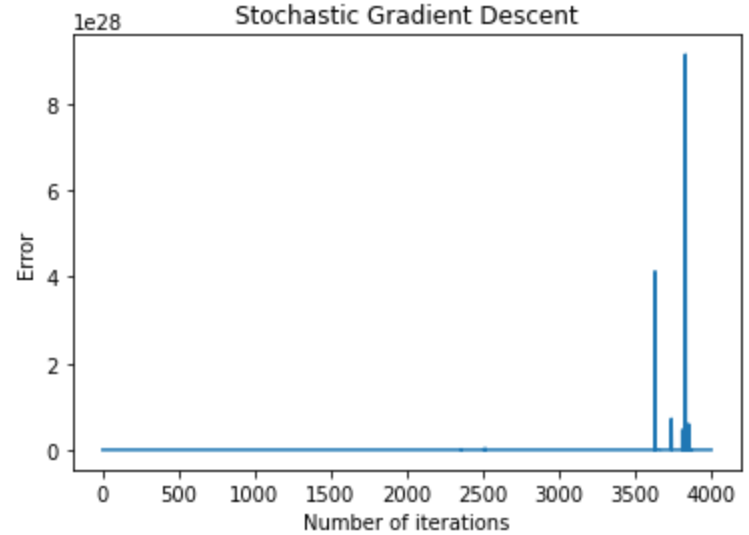


FOR 9 DEGREE POLYNOMIAL

Gradient Descent



Stochastic Gradient Descent



Comments on Overfitting

Overfitting arises when we get good accuracy on training data and comparatively poor on testing data. To overcome overfitting, we restrict weights with the help of norms. We used Ridge Regression (L2 Norm) and Lasso Regression (L1 Norm).

From the performance we observed, there is no significant difference between training and testing error, and hence no regularization is required. The Log(Lambda) vs RMS error plot also justifies the same as it gives a minimum error when lambda equals zero and increases as lambda increases.

Part B

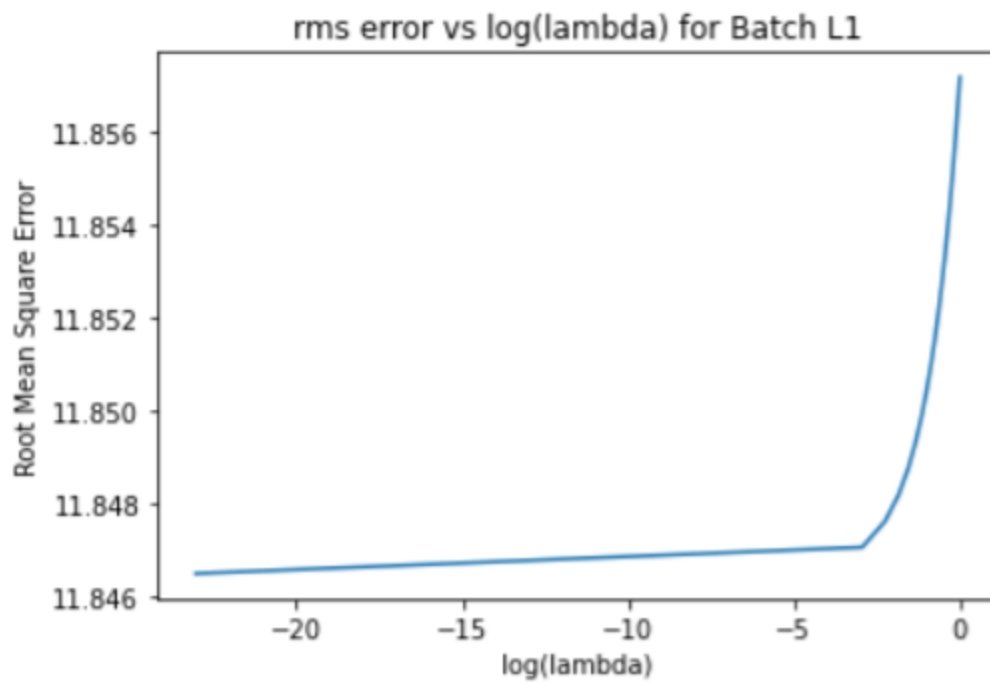
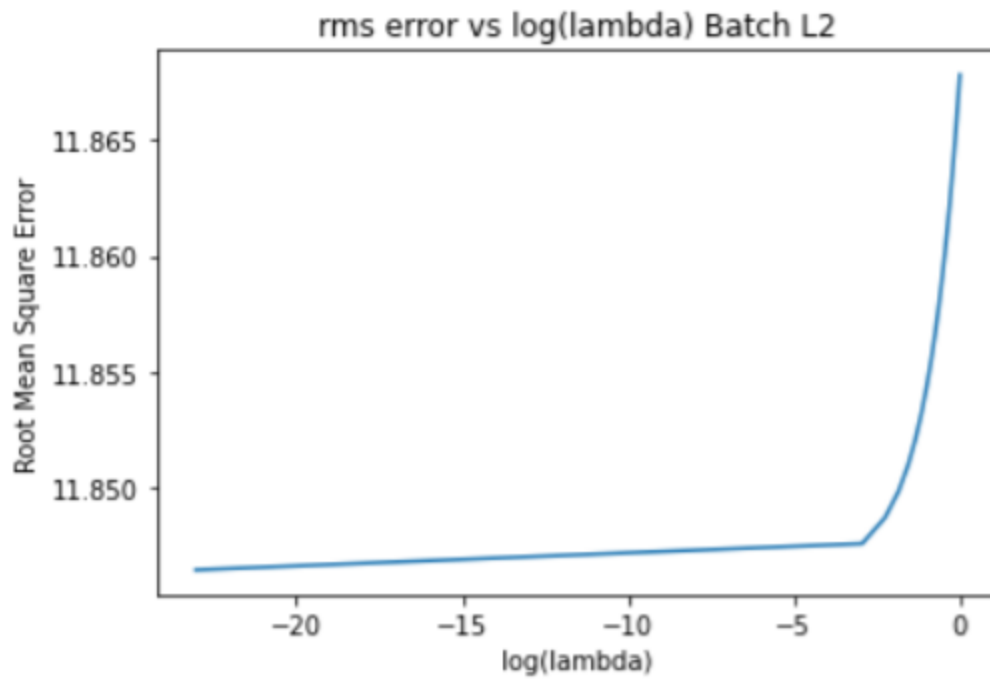
In this Part, we have tabulated the minimum training and testing error achieved by our model for 20 different values of λ . We have then drawn the plot of the root-mean-square error vs the logarithm of λ to figure out the optimal model.

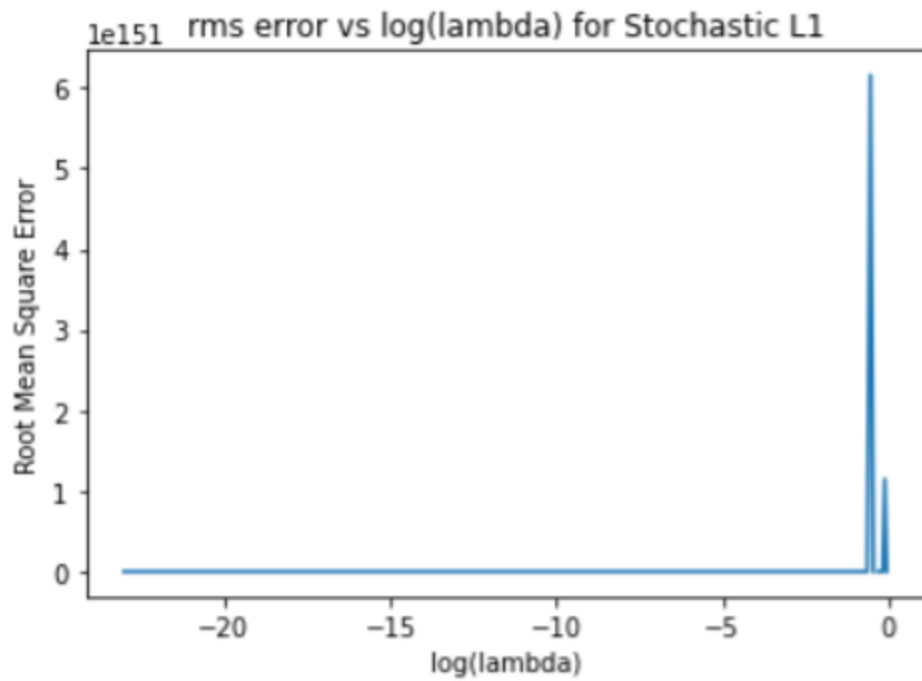
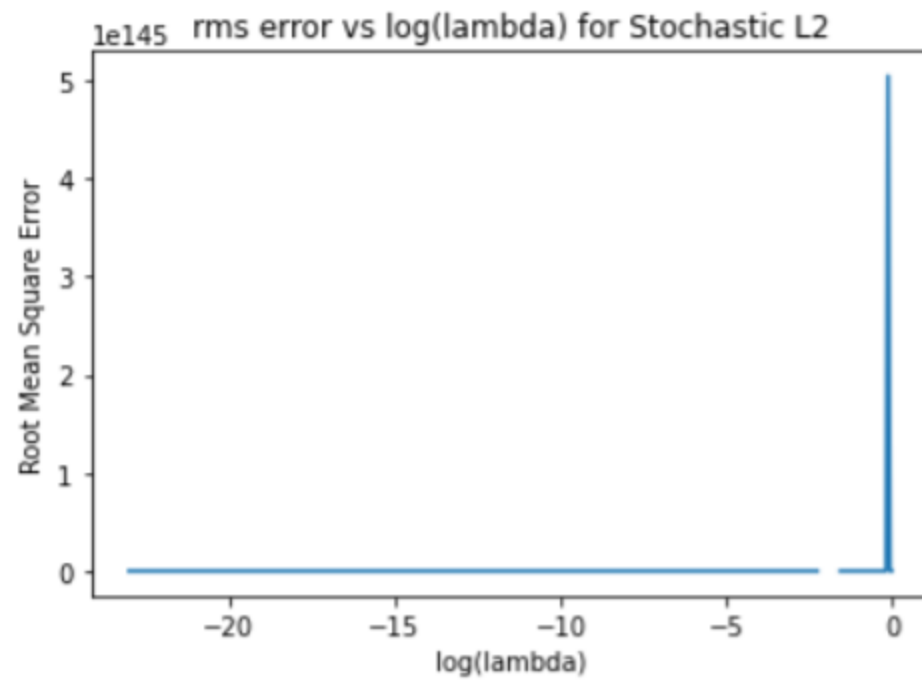
Tabular Information**1. LASSO REGRESSION**

lambda	Gradient Descent			Stochastic Gradient Descent		
	MSE	MAE	correlation_ coefficient	MSE	MAE	correlation_ coefficient
0.0	140.3392	10.3652	0.1244	241.7454	15.1595	0.2982
0.0526315	140.3526	10.3659	0.1244	241.7454	15.1595	0.1756
0.1052631	140.3659	10.3665	0.1244	241.7454	15.1595	-0.1667
0.1578947	140.3793	10.3672	0.1244	241.7454	15.1595	0.3054
0.2105263	140.3926	10.3678	0.1244	241.74545	15.15959	0.0509
0.2631578	140.4060	10.3685	0.1244	241.7454	15.1595	0.2437
0.3157894	140.4193	10.3692	0.1244	241.7454	15.1595	0.2528
0.368421	140.4327	10.3698	0.1244	241.7454	15.1595	0.2925
0.421052	140.4460	10.3705	0.1244	241.7454	15.1595	0.24969
0.4736842	140.4594	10.3711	0.1244	241.7454	15.1595	0.28874
0.5263157	140.4727	10.3718	0.1244	241.7454	15.1595	-0.16232
0.5789473	140.4860	10.3724	0.1244	241.7454	15.1595	0.05613
0.6315789	140.4994	10.3731	0.1244	241.7454	15.1595	0.30251
0.6842105	140.5127	10.3737	0.1244	241.7454	15.1595	0.24452
0.7368421	140.5260	10.3744	0.1245	241.7454	15.1595	-0.22285
0.7894736	140.5393	10.3750	0.1245	241.7454	15.1595	-0.1180
0.8421052	140.5526	10.3757	0.1245	241.7454	15.1595	-0.16629
0.8947368	140.5659	10.3763	0.1245	241.7454	15.1595	-0.11416
0.9473684	140.5792	10.3770	0.1245	241.7454	15.1595	0.30369
1.0	140.5925	10.3776	0.1245	241.7454	15.1595	-0.04717

2. RIDGE REGRESSION

lambda	Evaluation Metrics on Training set			Evaluation Metrics on Testing set		
	MSE	MAE	correlation_ coefficient	MSE	MAE	correlation_ coefficient
0.0	140.33923	10.3652	0.12440	241.74545	15.1595	0.29667
0.0526315	140.36596	10.36659	0.1244	241.74545	15.1595	0.266288
0.1052631	140.39268	10.36789	0.1244	241.74545	15.1595	-0.08597
0.1578947	140.41938	10.36920	0.1244	241.74545	15.1595	0.07778
0.2105263	140.44607	10.3705	0.1244	241.74545	15.1595	0.29503
0.2631578	140.47275	10.37182	0.1244	241.74545	15.1595	-0.24501
0.3157894	140.49940	10.3731	0.1244	241.74545	15.1595	-0.2568
0.368421	140.52605	10.37444	0.1245	241.74545	15.1595	0.27712
0.421052	140.5526	10.37574	0.1245	241.74545	15.1595	0.276734
0.4736842	140.57928	10.37704	0.1245	241.74545	15.1595	-0.108439
0.5263157	140.60588	10.37834	0.1245	241.74545	15.1595	-0.18284
0.5789473	140.6324	10.3796	0.1245	241.74545	15.1595	0.24768
0.6315789	140.6590	10.380946	0.1245	241.74545	15.1595	0.304500
0.6842105	140.6855	10.38224	0.1245	241.74545	15.1595	0.25033
0.7368421	140.71211	10.38353	0.1245	241.74545	15.1595	-0.16181
0.7894736	140.73863	10.38482	0.1246	241.74545	15.1595	-0.10469
0.8421052	140.76514	10.3861	0.1246	241.74545	15.1595	0.25549
0.8947368	140.79163	10.3874	0.1246	241.74545	15.1595	0.24608
0.9473684	140.81810	10.38869	0.1246	241.74545	15.1595	0.12218
1.0	140.8445	10.3899	0.1246	241.74545	15.1595	-0.00037

Root-mean-square error vs the logarithm of lambda plot



Comparison

We observe that as lambda increases, our error increases. Minimum error is achieved when Lambda is 0 or when no regularization is used.

When we used regularization in part b, optimal lambda came out to be 0, which signifies no regularization is needed.

We evaluated our model on three different evaluation metrics, namely, Mean Squared Error, Mean Absolute Error, and Correlation Coefficient, and found a Model with 1-degree polynomial regression using stochastic gradient descent to perform best with a correlation coefficient of 0.89 and Mean Squared Error of 2.3 followed by 2-degree polynomial regression using stochastic gradient descent both without regularization.