# Foundations Of Data Science (CS F320) - Assignment 1

**Group Members:**

Abhipal Sharma - 2019A7PS0161H

Vansh Madan - 2018B4A70779H

Anish Sai Mitta - 2018A7PS1221H

*Under the guidance of :*

**Prof. NL Bhanumurthy**

# Table of Contents

# Introduction

In this assignment, the following needs to be performed:

A. Using Batch Gradient descent and Stochastic Gradient Descent, develop polynomial regression models of degrees 0,1,2,3,4,5,6,7,8 and 9 without using regularization and perform a comparative analysis among these models

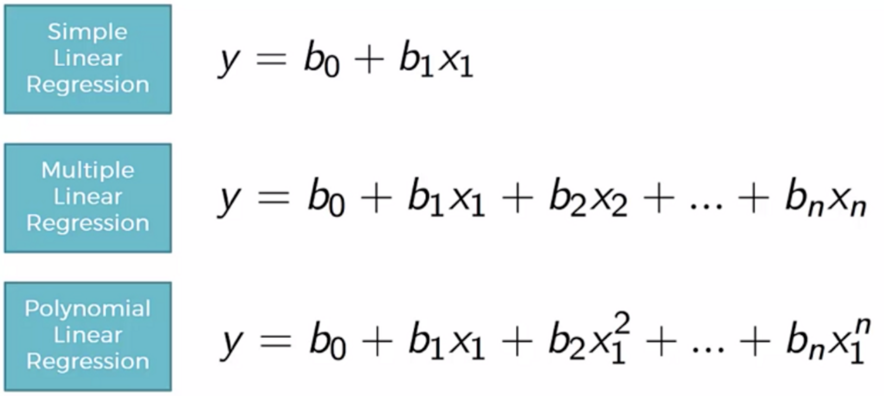
B. Using Batch Gradient descent and Stochastic Gradient Descent, develop polynomial regression models of degree 9 and implement a Ridge and Lasso regression regularization.

## Polynomial Regression :

______Polynomial Regression is a form of regression analysis in which the relationship between the independent variables and dependent variables are modeled in the nth degree polynomial.

Polynomial Regression models are usually fit with the method of least squares.The least square method minimizes the variance of the coefficients.

Polynomial Regression is a special case of Linear Regression where we fit the polynomial equation on the data with a curvilinear relationship between the dependent and independent variables.



$$y = b_0 + b_1 x_1$$

$$y = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \dots + b_n x_1^n$$

## Batch Gradient Descent :

Batch Gradient Descent involves calculations over the full training set at each step as a result of which it is very slow on very large training data. Thus, it becomes very computationally expensive to do Batch GD. However, this is great for convex or relatively smooth error manifolds. Also, Batch GD scales well with the number of features.

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$here \;\; J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\widehat{y^i} - y^i) X_j^i$$

## Stochastic Gradient Descent :

______SGD tries to solve the main problem in Batch Gradient descent which is the usage of whole training data to calculate gradients as each step. SGD is stochastic in nature i.e it picks up a “random” instance of training data at each step and then computes the gradient making it much faster as there is much fewer data to manipulate at a single time, unlike Batch GD.

$$for\ i\ in\ range\ (m):$$

$$\theta_j = \theta_j - \alpha(\widehat{y^i} - y^i)X_j^i$$

**Ridge Regression Regularization :**

______Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

$$\hat{\beta}^{ridge} = \underset{\beta\in\mathbb{R}}{argmin}\|y - XB\|_2^2 + \lambda\|B\|_2^2$$

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. Higher the values of alpha, bigger is the penalty and therefore the magnitude of coefficients is reduced.

It shrinks the parameters. Therefore, it is used to prevent multicollinearity

It reduces the model complexity by coefficient shrinkage

## Lasso Regression Regularization :

______Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models.

______

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

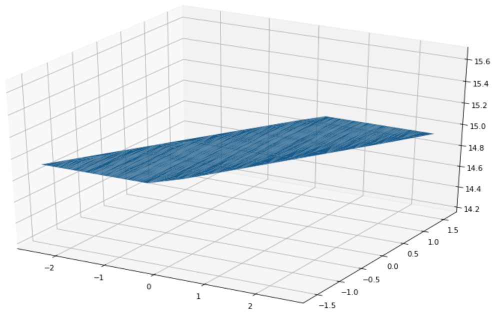
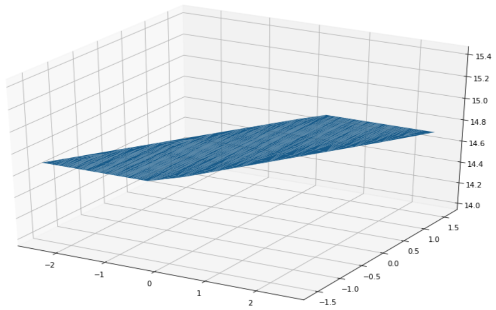
Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. Higher the values of alpha, bigger is the penalty and therefore the magnitude of coefficients is reduced.

It shrinks the parameters. Therefore, it is used to prevent multicollinearity

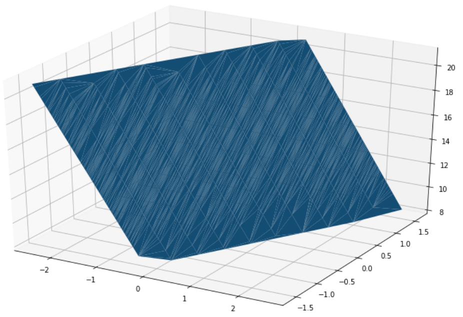
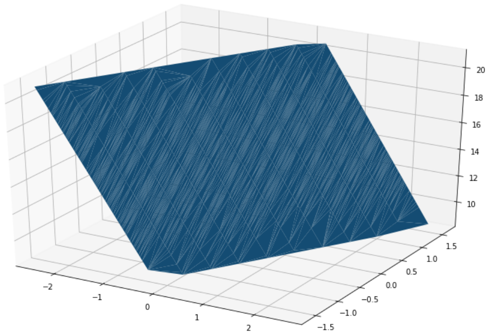
It reduces the model complexity by coefficient shrinkage

## Tabulated data of results for minimum testing error and training error for 0-9 degrees of BGD and SGD

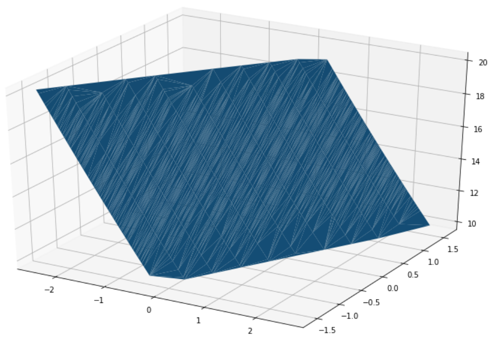
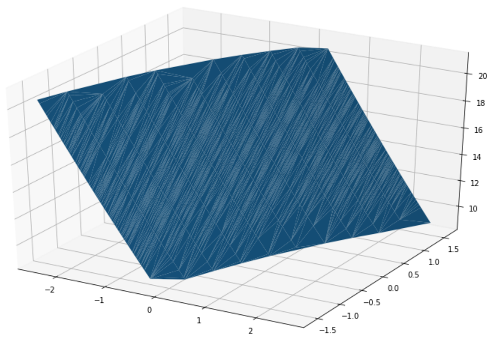
### Degree 0:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 5.768656509435732<br>Training Error : 2.568018485988157 | Minimum Testing Error : 5.792689536333653<br>Training Error : 2.597016472243353 |
|  |  |

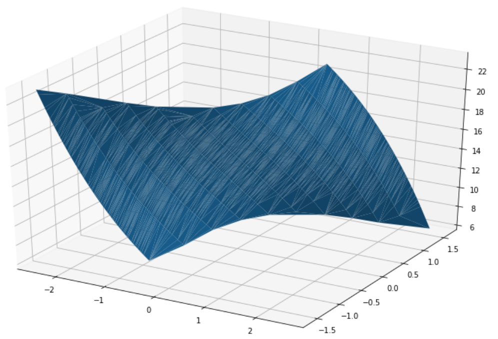
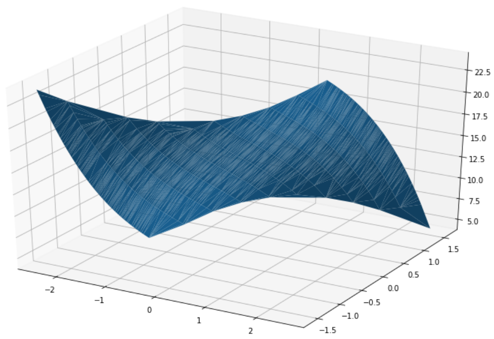
### Degree 1:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1765143319367577<br>Training Error : 2.461002459914413 | Minimum Testing Error : 1.1932346147567967<br>Training Error : 2.615411899335708 |
|  |  |

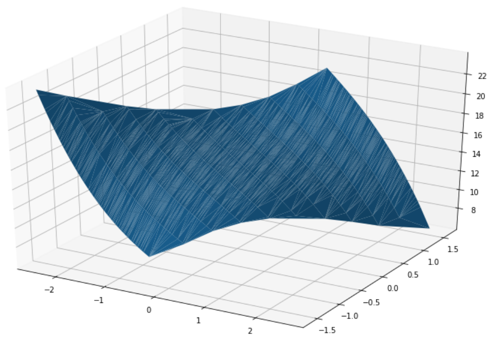
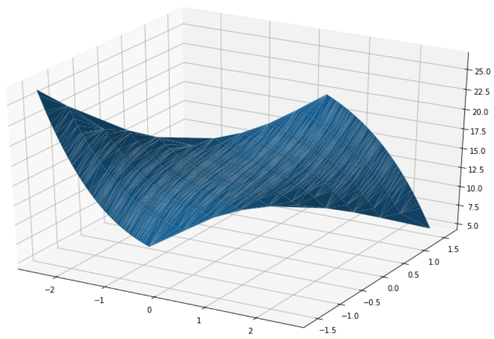
## Degree 2:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1657383976624354<br>Training Error : 2.449179105890716 | Minimum Testing Error : 1.2063391063927014<br>Training Error : 2.615411899335708 |
|  |  |

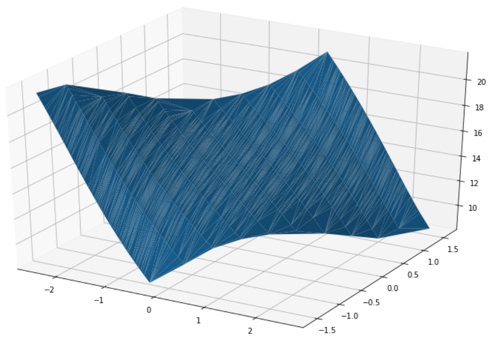
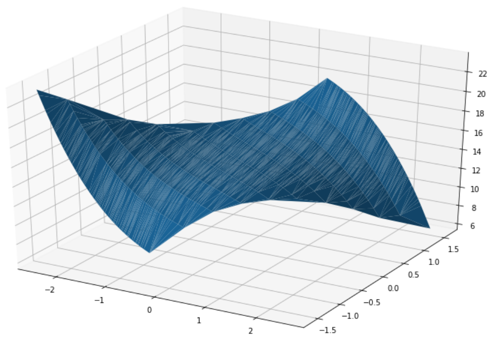
## Degree 3:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1290423054835583<br>Training Error : 2.3600090532134463 | Minimum Testing Error : 1.6018003824213949<br>Training Error : 3.500218217438337 |
|  |  |

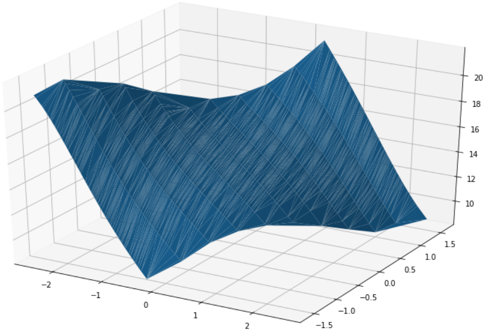
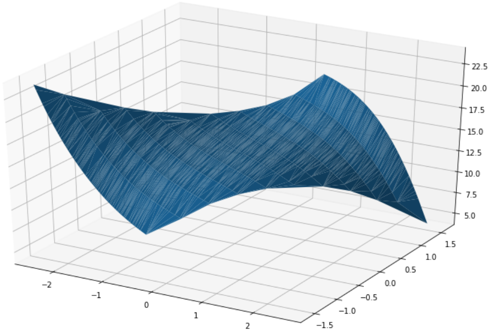
## Degree 4:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1265208243700002<br>Training Error : 2.3712700963894164 | Minimum Testing Error : 1.7031485615492195<br>Training Error : 3.927548920999474 |
|  |  |

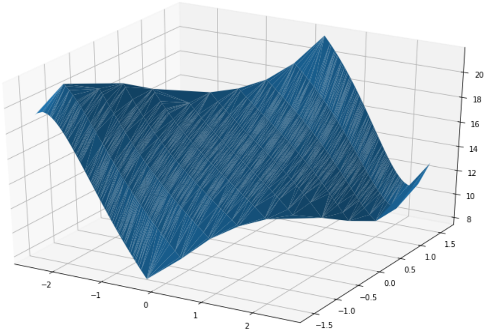
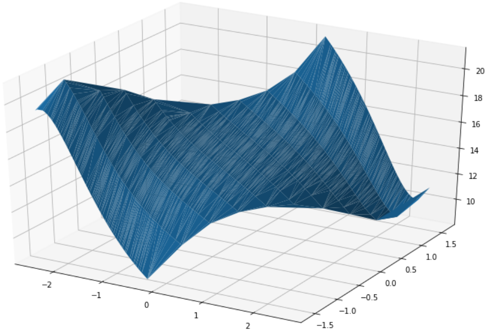
## Degree 5:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1127160388629713<br>Training Error : 3.1758707951494487 | Minimum Testing Error : 1.5022391710270029<br>Training Error : 3.1758707951494487 |
|  |  |

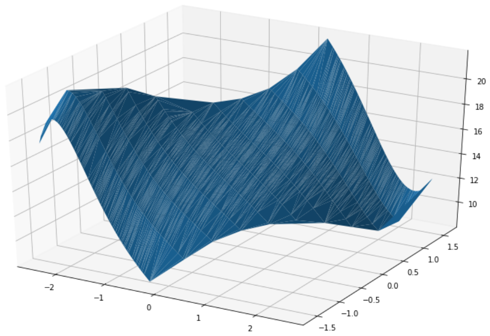
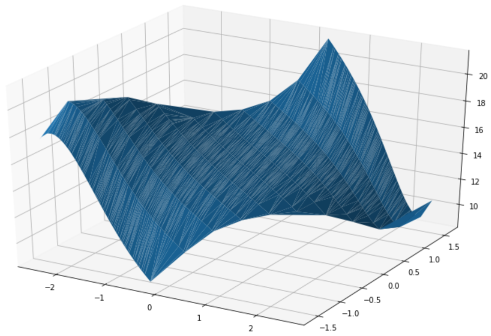
## Degree 6:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1119909798604926<br>Training Error : 3.292384073601101 | Minimum Testing Error : 1.6640276215113394<br>Training Error : 4.689811505739849 |
|  |  |

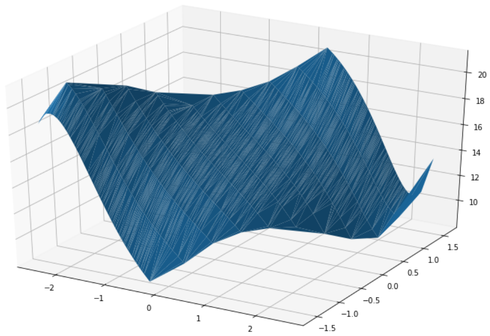
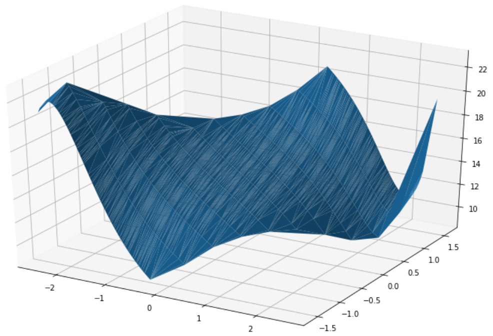
## Degree 7:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1135802536550163<br>Training Error : 3.2045839643638194 | Minimum Testing Error : 1.615631178573327<br>Training Error : 4.218013789278498 |
|  |  |

Degree 8:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1135280715665081<br>Training Error : 3.31164940045882 | Minimum Testing Error : 1.5453568143758412<br>Training Error : 4.142632030842901 |
|  |  |

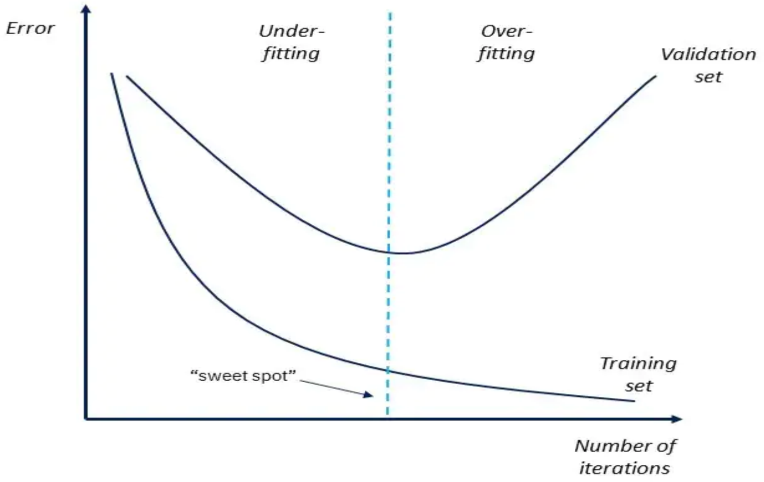
Degree 9:

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Minimum Testing Error : 1.1557970478661836<br>Training Error : 3.1920552176385804 | Minimum Testing Error : 1.6129583955455475<br>Training Error : 3.7073175055333643 |
|  |  |

## **Overfitting :**

______Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose. Generalization of a model to new data is ultimately what allows us to use machine learning algorithms every day to make predictions and classify data.

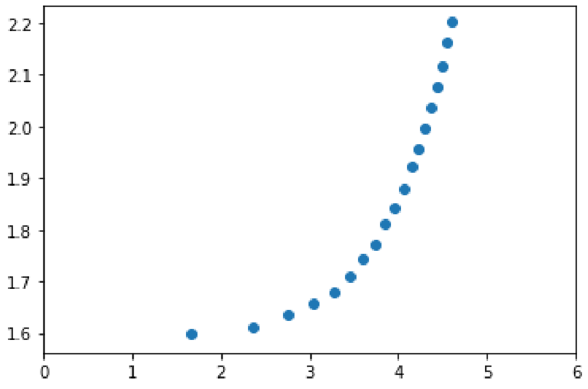
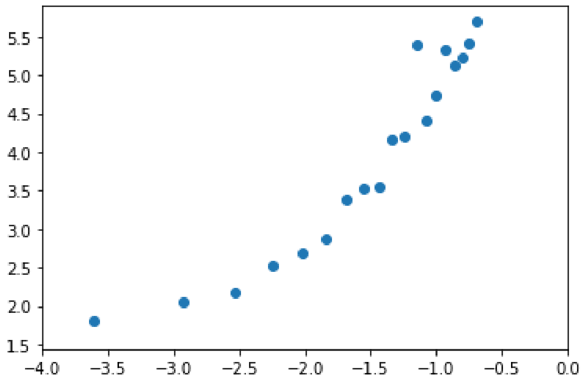
Low error rates and a high variance are good indicators of overfitting. In order to prevent this type of behavior, part of the training dataset is typically set aside as the “test set” to check for overfitting. If the training data has a low error rate and the test data has a high error rate, it signals overfitting.



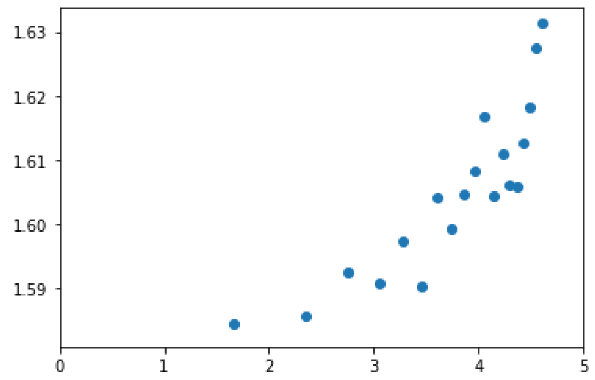
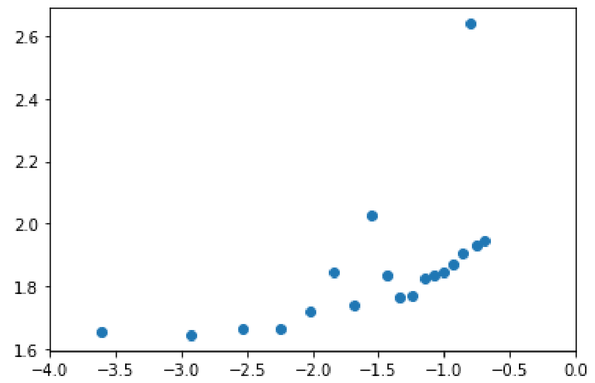
As can be seen from the tables with train and test errors constructed above, as we increase the order of the polynomial regression equation , the value of (testing error - testing error) decreases upto order 3 and then increases, indicating that in higher order polynomial regression models, overfitting is being encountered. In degree 0 however, the training and testing errors are both high as it doesn’t capture the data points well enough.

Hence, we shall use regularization with the 9th order polynomial regression equation to decrease the disparity between Testing and Training error.

## Tabulated data of results for Ridge Regression of BGD and SGD for the first 5 lambda values

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Lambda is 0.001<br>RMS error 1.5926829507936024<br>Training error is  1.2683194908743085<br>Test error is  1.3602677668165142<br>------------------------------------------------------------<br>Lambda is 5.264105263157894<br>RMS error 1.5996692247948054<br>Training error is  1.2794708143778069<br>Test error is  1.3936837100572672<br>------------------------------------------------------------<br>Lambda is 10.527210526315788<br>RMS error 1.6112710918767867<br>Training error is  1.2980972657589063<br>Test error is  1.4328364558229014<br>------------------------------------------------------------<br>Lambda is 15.790315789473683<br>RMS error 1.6371142713987474<br>Training error is  1.3400715688087257<br>Test error is  1.491514426559145<br>------------------------------------------------------------<br>Lambda is 21.053421052631577<br>RMS error 1.6582761770246057<br>Training error is  1.3749399396436708<br>Test error is  1.5336760505650413 | Lambda is 0.001<br>RMS error is 1.6431075257154217<br>Training error is  1.3499011705313277<br>Test error is  1.434708049337866<br>------------------------------------------------------------<br>Lambda is 0.027263157894736843<br>RMS error is 1.8120969670574116<br>Training error is  1.641847709009335<br>Test error is  1.9169716874484024<br>------------------------------------------------------------<br>Lambda is 0.053526315789473686<br>RMS error is 2.0550920000252995<br>Training error is  2.1117015642839925<br>Test error is  2.59358084127285<br>------------------------------------------------------------<br>Lambda is 0.07978947368421052<br>RMS error is 2.178876797067796<br>Training error is  2.3737520484002084<br>Test error is  2.8251298786415924<br>------------------------------------------------------------<br>Lambda is 0.10605263157894737<br>RMS error is 2.5223825389227166<br>Training error is  3.181206836331104<br>Test error is  3.7100686695064273 |
|  |  |

## Tabulated data of results for Lasso Regression of BGD and SGD for the first 5 lambda values

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|---|
| Lambda is 0.001<br>RMS error is 1.583257608015159<br>Training error is  1.2533523266689415<br>Test error is  1.343245498414478<br>------------------------------------------------------------------<br>Lambda is 5.264105263157894<br>RMS error is 1.5844359827816212<br>Training error is  1.2552186917665809<br>Test error is  1.3408988998217597<br>------------------------------------------------------------------<br>Lambda is 10.527210526315788<br>RMS error is 1.5856436994110505<br>Training error is  1.257132970740981<br>Test error is  1.3529449001138225<br>------------------------------------------------------------------<br>Lambda is 15.790315789473683<br>RMS error is 1.5925180863981785<br>Training error is  1.268056927752658<br>Test error is  1.362783705238573<br>------------------------------------------------------------------<br>Lambda is 21.053421052631577<br>RMS error is 1.5908119198474875<br>Training error is  1.2653412821644243<br>Test error is  1.3607778371224197 | Lambda is 0.001<br>RMS error is 1.8113389186991682<br>Training error is  1.640474339197136<br>Test error is  1.793347234212169<br>------------------------------------------------------------------<br>Lambda is 0.027263157894736843<br>RMS error is 2.286909971107861<br>Training error is  2.6149786079762785<br>Test error is  2.7889446273962224<br>------------------------------------------------------------------<br>Lambda is 0.053526315789473686<br>RMS error is 1.6888028680861846<br>Training error is  1.4260275636280615<br>Test error is  1.411331619872981<br>------------------------------------------------------------------<br>Lambda is 0.07978947368421052<br>RMS error is 1.6652570793101058<br>Training error is  1.386540570096212<br>Test error is  1.5117615622936775<br>------------------------------------------------------------------<br>Lambda is 0.10605263157894737<br>RMS error is 1.7039785207297462<br>Training error is  1.451771399554167<br>Test error is  1.629199744943512 |
|  |  |

## Comparison between best models obtained form part A and part B:

1. **Best model form part A:**
   - 3rd degree Regression model constructed using Batch Gradient descent
   - Training error : 1.129
   - Testing error : 2.360
2. **Best model form part B (Ridge regression) :**
   - 9th order polynomial regression model constructed using Batch Gradient descent
   - Lambda : 0.01
   - Training error : 1.268
   - Testing error : 1.360
3. **Best model form part B (Lasso regression) :**
   - 9th order polynomial regression model constructed using Stochastic Gradient descent
   - Lambda : 5.26
   - Training error : 1.255
   - Testing error : 1.340

Hence, the best model overall is the 9th order polynomial regression model constructed using Batch Gradient descent, with the application of Lasso regularization with a lambda value of 5.26.