CS F320 - Foundations of Data Science

Assignment 1

BY

Name of the Student	ID Number
Rikil Gajarla	***2017A7PS0202H***
Kasuba Badri Vishal	***2017A7PS0270H***

Foundations of Data Science Report

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
I Semester, 2019-20

Abstract

This assignment implements Linear Regression using Gradient Descent, Stochastic Gradient Descent, Normal Equations and Regularization. The assignment involves predicting the altitude using longitude and latitude. The dataset used consists of 434874 data points. 80% of the data is used to train the model and the remaining 20% is used to test the developed model

Python is used to implement the regression model using numpy, pandas and matplotlib libraries

Part A - Gradient Descent Method

Gradient Descent is an iterative method used to minimize the loss/error on the training data to best fit the model. Main concept behind Gradient Descent involves in minimizing the loss function which is half of the sum of squares of errors.

Gradient Descent takes steps towards the nearest minima of the error function. In the case of linear regression, the error function is a convex figure and hence, we only have one global minima. Therefore, gradient descent is bound to converge to that point. To find a local minimum of a function using gradient descent, one takes steps proportional to the *negative* of the gradient of the function at the current point.

Results:

Loss function = $\frac{1}{2}\sum_{n=1}^{N}(y_n - f(x_n))^2$

Initial weights = (1, 1, 1)

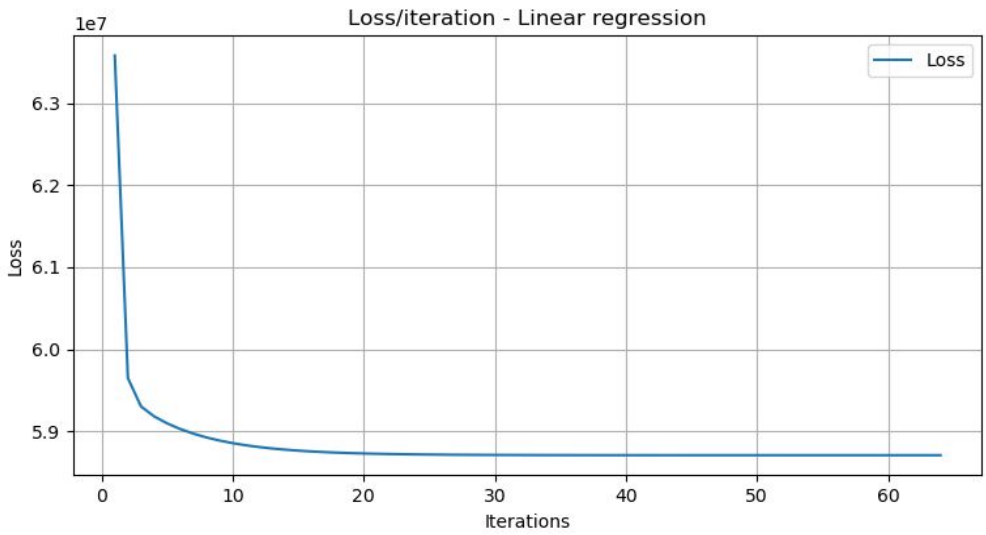
Learning rate = 3.5×10^{-6}

Stopping criteria = 1×10^{-3}

Results:

	MSE	**RMSE**	**R2**
Training error	337.5224	18.37178	0.02618
Testing error	337.2080	18.36322	0.02751

Time taken: 0.57758



Part B - Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is similar to normal gradient descent but only single data point is considered in each iteration for converging to minima. This method is faster on larger datasets as only one point is used instead of complete dataset in every iteration.

Parameters:

Loss function = $\frac{1}{2}\sum_{n=1}^{N}(y_n - f(x_n))^2$

Initial weights = (1, 1, 1)

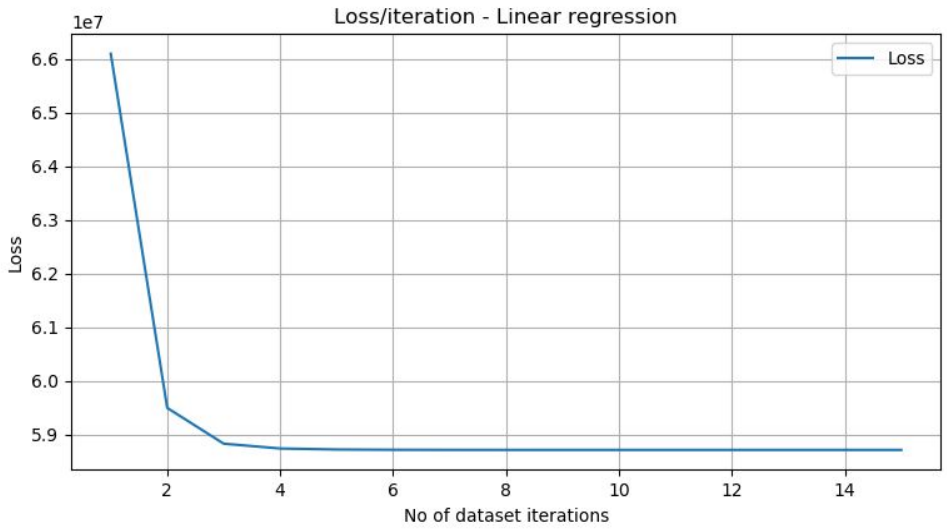
Learning rate = 3.5×10^{-6}

Stopping criteria = 1×10^{-3}

Results:

	MSE	**RMSE**	**R2**
Training error	337.52254	18.371786	0.02618
Testing error	337.20982	18.363273	0.02752

Time taken: 1.484



Part C - Regularization

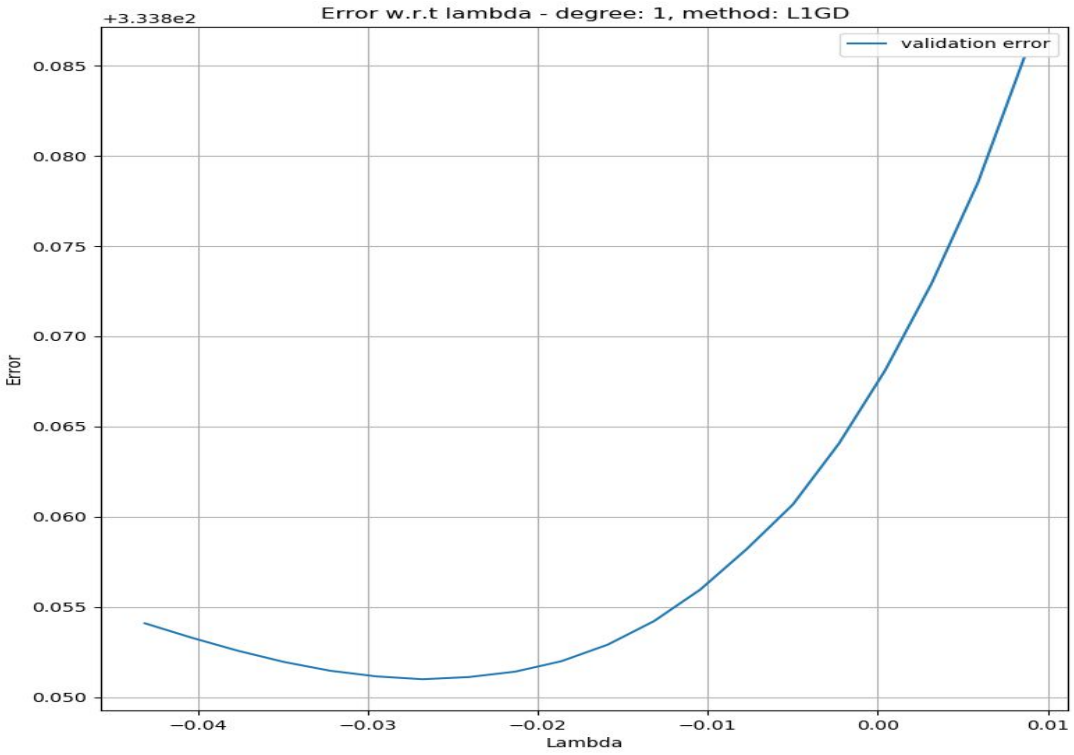
Regularization involves in constraining the weights using the following techniques. The training data is further split into validation dataset to validate the addition parameter λ (regularization parameter). This parameter then is set to the value which gives the lowest validation error

L1 Regularization

L1 Regularization, also known as least absolute shrinkage and selection operator (also Lasso) tries to reduce the overfitting of weights to the given data by putting a modulus constraint on the freedom of weights. The constraint is

$$\lambda \sum_{i=1}^{D} |W_i| = 1$$

The parameter λ is set by taking the value which gives the least validation error
The following plot give the variation of λ on validation set



Selected Value of λ is -0.034492

Results:

	MSE	**RMSE**	**R2**
Training error	337.547693	18.372295	0.026113402
Testing error	337.25927	18.364317	0.027366882

L2 Regularization

L2 Regularization, also known as Tikhonov regularization or ridge regularization tries to reduce the overfitting of weights to the given data by putting a square constraint on the freedom of weights. The constraint is

$$\lambda \sum_{i=1}^{D} W^2 = 1$$

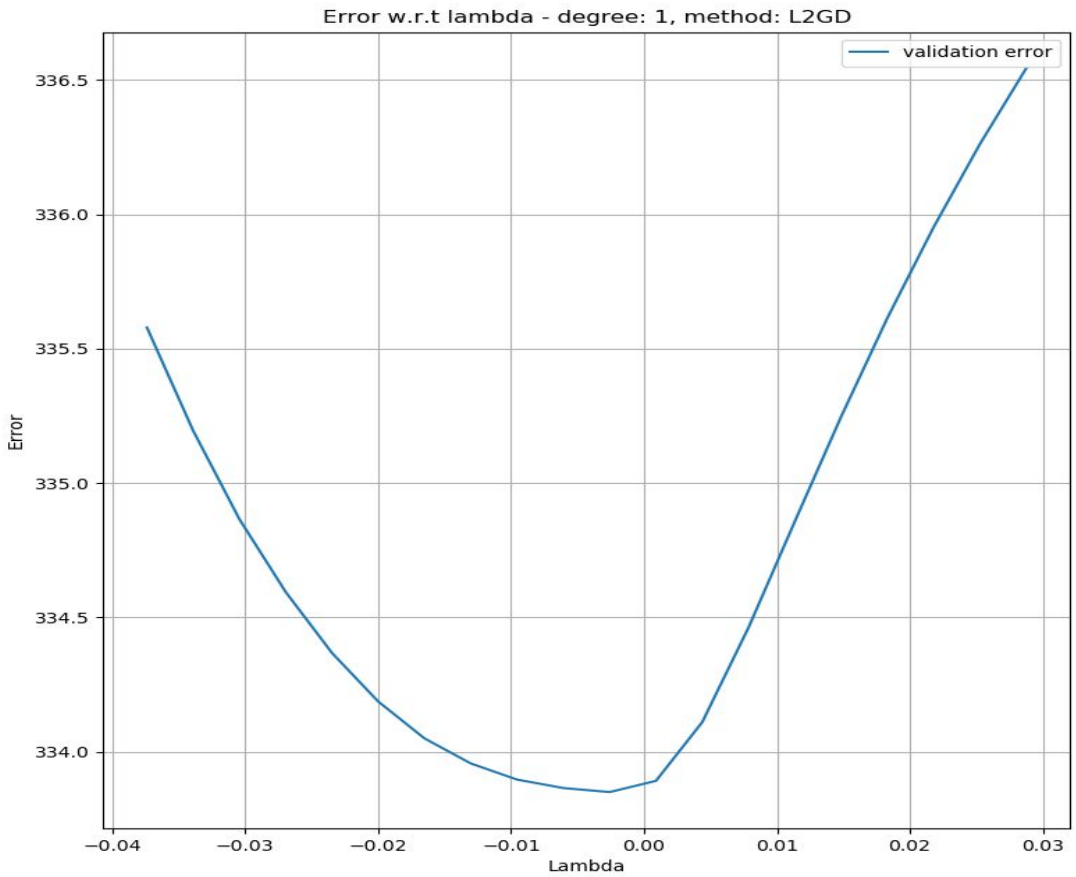
The parameter λ is set by taking the value which gives the least validation error
The learning rate in this case must be reduced to avoid overshooting.

Value of parameter λ is -0.04311596

Results:

	MSE	**RMSE**	**R2**
Training error	339.5736930	18.427525	0.0202680
Testing error	339.0556572	18.413464	0.0221862

The following plot gives the variation of λ on validation set:



Part D - Normal Equations Method

To minimize the error, when we equate the first derivative of the equation to 0, we get a set of linear equations. We know from calculus that the solution of the equations will give us the exact point of minima. Therefore when we solve the simultaneous equations, we get

$$W = (X^T X)^{-1} X^T Y$$

Where W is the set of weights, X is the training matrix and Y is the target matrix. On performing this operation, the results obtained are:

	MSE	**RMSE**	**R2**
Training error	337.52245	18.371784	0.02618
Testing error	337.20838	18.363234	0.02751

Time taken: 0.01724

Conclusion

We see that Normal equations method is the best of all as it gives the least error but, it becomes computationally difficult to get solution of equations of a larger dataset with many features.

Gradient descent, although better than Stochastic gradient descent, it takes time based on the size of the dataset. Stochastic gradient descent is a lot faster on very large datasets.

In this assignment, linear regression is not able to overfit the data as it can be seen through the coefficients of regularization of L1 and L2. they are almost close to 0. The approach of L1 and L2 regularization might be different, but in this case, both of them manage to get almost similar values.