Programming Assignment 1

Name: Amit Sarker, ID: 500969

1 Introduction

The main goal of this assignment is to create a linear regression model that allows us to visualize the underlying link between variables within our data scope. The goal of uni-variate linear regression is to find a link between one independent (explanatory) variable and one dependent variable. Multi-variate linear regression is a technique for determining the degree to which more than one independent variables (explanatory) and dependent variables are linearly connected. To visualize the regression model, I have used the Concrete Compressive Strength Data Set from the UCI Machine Learning Repository.

1.1 Problem and Algorithm Description

In this assignment, I have implemented **Gradient Descend** algorithm for uni-variate and multi-variate linear regression. Linear Regression is a supervised learning algorithm which is both a statistical and a machine learning algorithm. It depicts the relationship between the dependent variable y and the independent variables x_i. The hypothetical function used for prediction is represented by $h(x)$.

$$h(x) = b + w * x \tag{1}$$

here, b is the bias, x represents the feature vector, and w represents the weight vector. Uni-variate linear regression refers to linear regression using only one variable. After initializing the weight vector, we can use the gradient descent learning to identify the weight vector that best fits the model. The cost function (or loss function) is used to evaluate regression model's performance or to quantify the difference between the expected and predicted values using our hypothetical function. We use Mean Squared Error as the cost function and it is represented by J.

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h(x^{(i)}))^2 \tag{2}$$

here, m is the total number of training examples in the dataset, $y^{(i)}$ represents the value of target variable for i^{th} training example. Our goal is to minimize the cost function J (or improve the performance of our regression model). To do so, we must determine the weights at which J is the smallest. Gradient Descent is one such technique that may be used to minimize any differentiable function. It's a first-order iterative optimization procedure that takes us to the function's minimum. For this, we will start with a randomly initialized weight vector w. We will keep changing w to reduce $J(w)$ until we hopefully end up at a minimum.

$$w_{new} = w_{old} - \alpha * \frac{\partial J(w)}{\partial w} \tag{3}$$

where alpha is the learning rate. **We select $\alpha = 10^{-7}$ that is determined by trial and error.** We will stop the update of gradient descend if the update of w_{old} and w_{new} is negligible or a maximum number of iterations are reached. For multi-variate linear regression, the approach is very similar as uni-variate regression. But If we have multiple independent variables, the formula for linear regression will look like:

$$h(x) = b + w_1x_1 + w_2x_2 + w_3x_3 + ... \tag{4}$$

The multi-variate Concrete Compressive Strength dataset contains 1030 observations and 9 features (8 quantitative input features, and 1 quantitative output feature). We use 900 observations as the training data and use the other 130 observations to test the quality of the regression model.

1.2 Pseudocode

Procedure 1: LinearRegression

```
Function LinearRegression(X, y, iterations, α, stop_condition):
    randomly initialize w from value between (0, 1)
    calculate J(w_old) using Equation 2
    while terminating condition is not met do
        update w using Equation 3
        calculate J(w_new) using Equation 2
        if J(w_new) > J(w_old) then
            α = α * 0.8
        else
            α = α * 1.15
    return w^T X
```

1.3 Data Normalization

I have used z-score normalization technique to normalize the data except the bias and output feature columns. Z-score normalization refers to the process of normalizing every value in a dataset such that the mean of all of the values is 0 and the standard deviation is 1. We use the following formula to perform a z-score normalization on every value in a dataset:

$$normalized\_value = \frac{X - \mu}{\sigma} \tag{5}$$

where, X is the original value in dataset, μ is the mean of data, and σ is the standard deviation of the data. Figure 1 shows the histograms for the original dataset and Figure 2 shows the histograms for the normalized dataset.

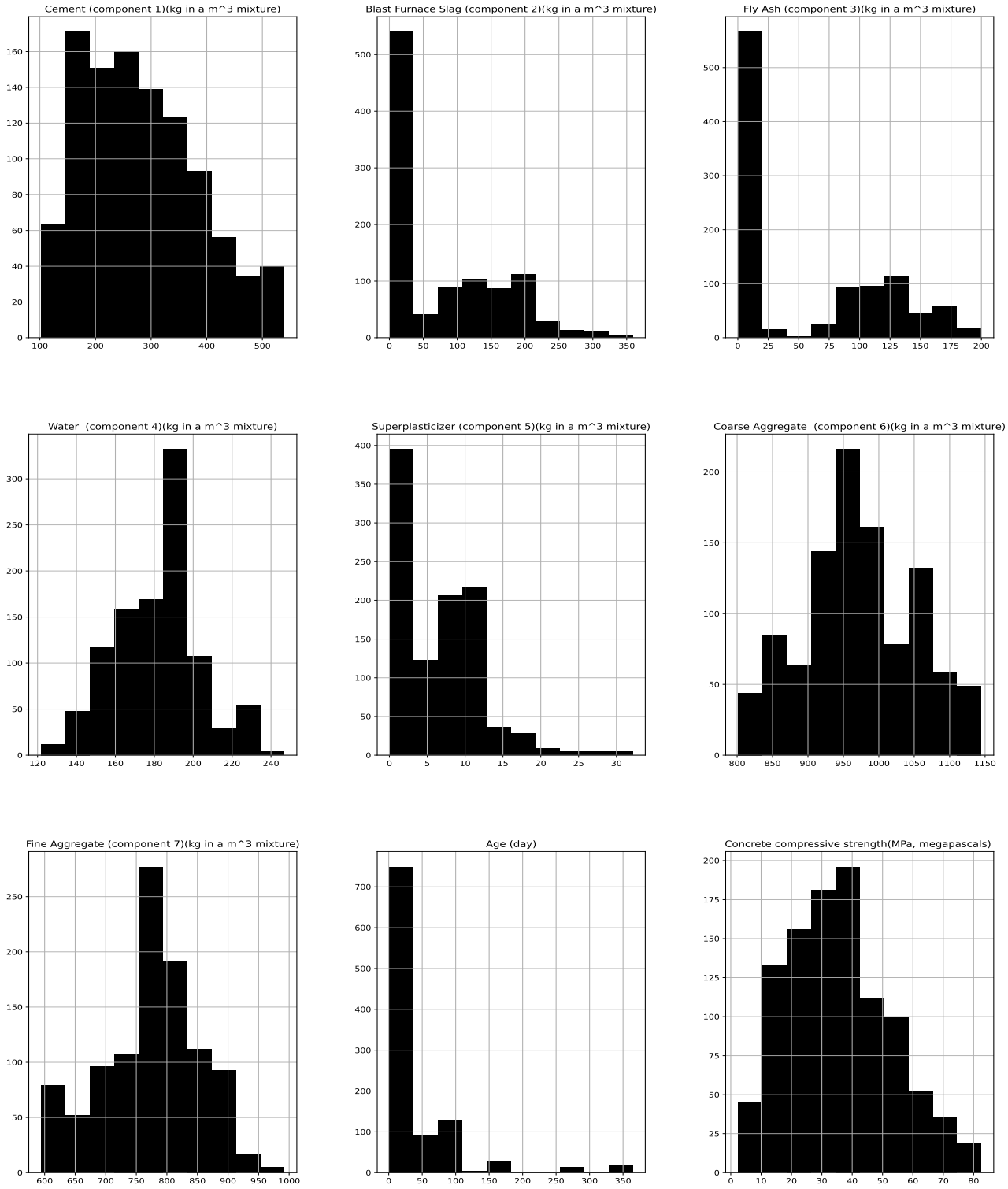


Figure 1: Histograms for the original dataset.

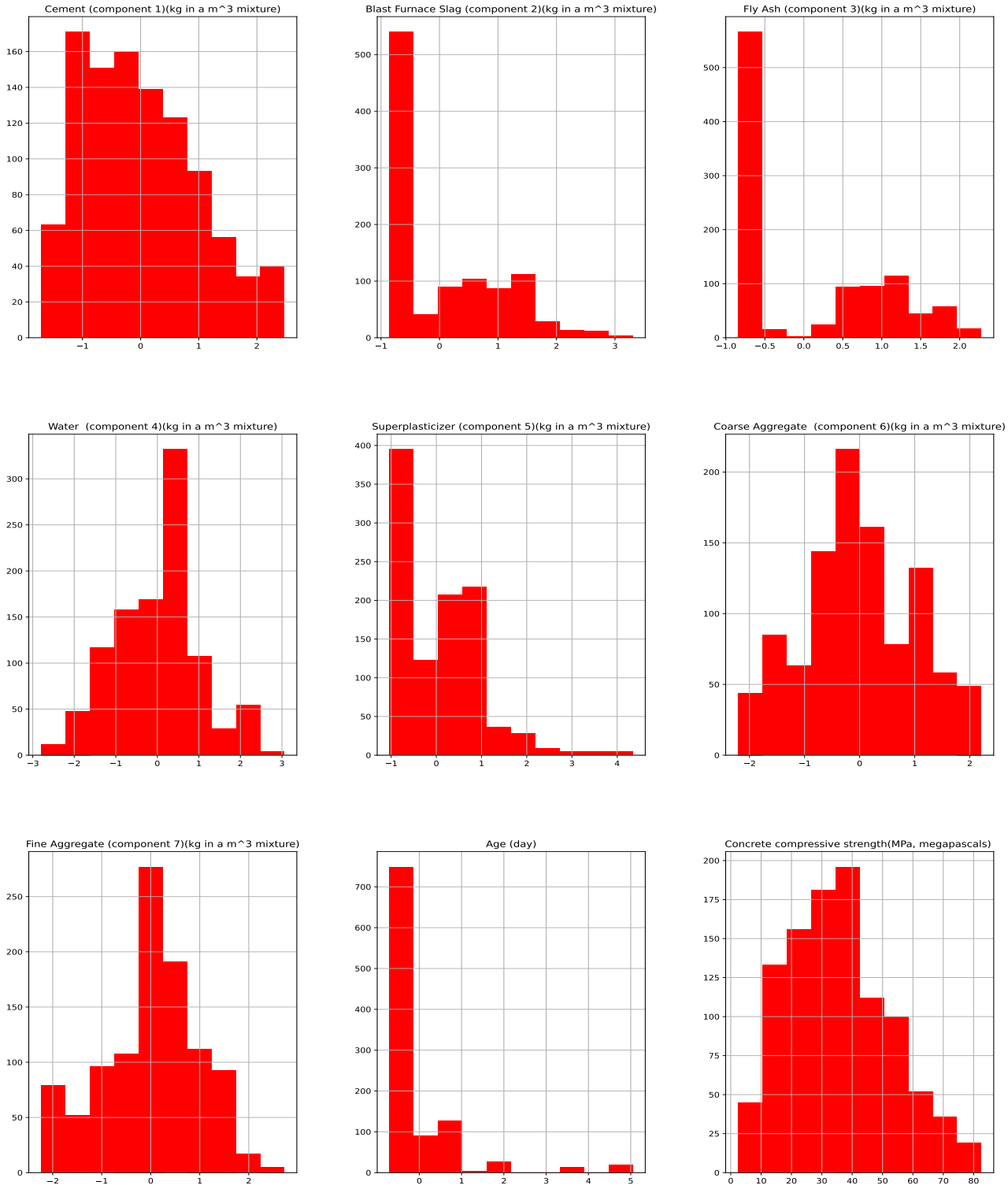


Figure 2: Histograms for the normalized dataset.

2 Results

2.1 Explained Variance

Explained variance is used to measure the discrepancy between a model and actual data. The formula is following:

$$explained\_variance = 1 - \frac{MSE}{variance(observed)} \tag{6}$$

Table 1: Explained Variance for uni-variate and multi-variate regression in original dataset.

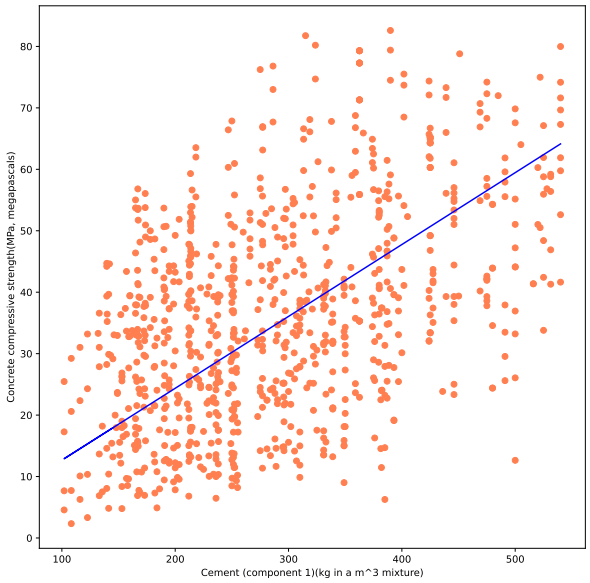
Input Features	Explained Variance (Training Data)	Explained Variance (Test Data)
Cement (component 1)	-2.087	-0.890
Blast Furnace Slag (component 2)	-3.738	-1.30
Fly Ash (component 3)	-2.865	-2.106
Water (component 4)	-3.417	-2.229
Superplasticizer (component 5)	-2.044	-3.102
Coarse Aggregate (component 6)	-3.014	-1.891
Fine Aggregate (component 7)	-3.147	-1.985
Age	-2.445	-1.523
Multi-variate Regression	-0.430	-0.0381

Table 2: Explained Variance for uni-variate and multi-variate regression in normalized dataset.

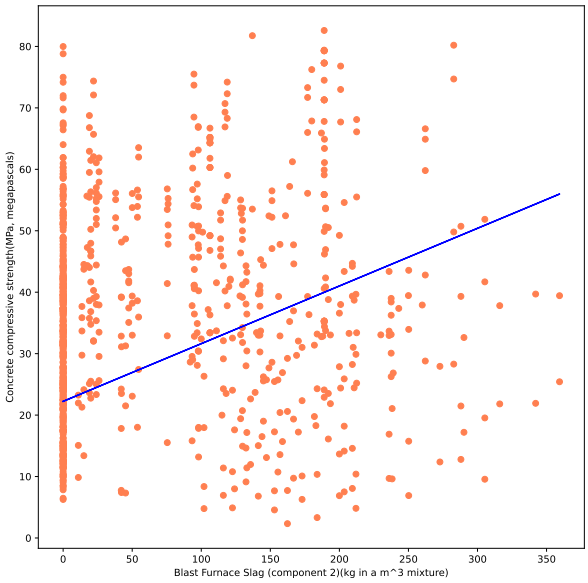
Input Features	Explained Variance (Training Data)	Explained Variance (Test Data)
Cement (component 1)	-1.844	-0.414
Blast Furnace Slag (component 2)	-2.622	-1.783
Fly Ash (component 3)	-2.678	-1.605
Water (component 4)	-2.365	-1.701
Superplasticizer (component 5)	-2.043	-3.101
Coarse Aggregate (component 6)	-2.544	-2.316
Fine Aggregate (component 7)	-2.566	-1.945
Age	-2.275	-1.630
Multi-variate Regression	-0.427	-0.038

Table 1 shows the explained variance for both training and test data for uni-variate and multi-variate regression in the original dataset. Table 2 shows the explained variance for both training and test data for uni-variate and multi-variate regression in the normalized dataset.

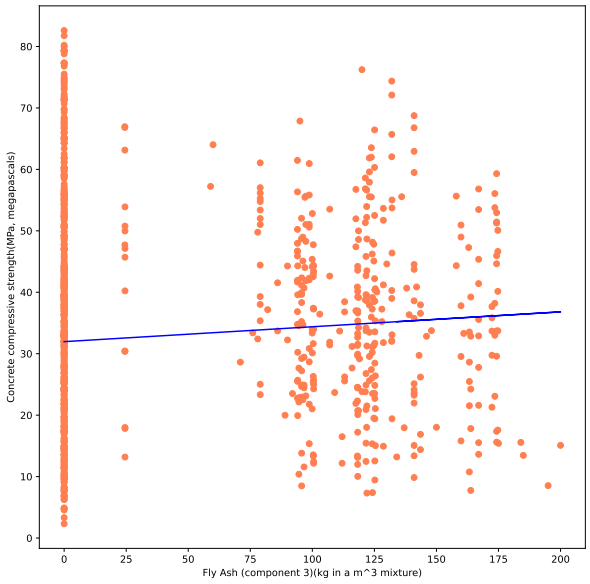
2.2 Scattrerplots for the original dataset



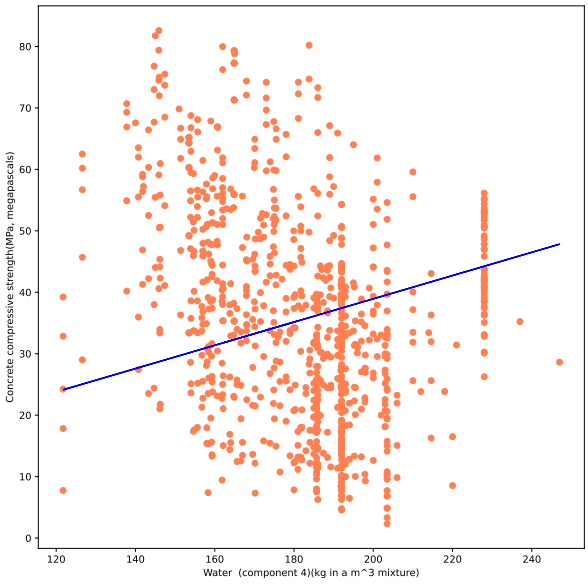
(a) Feature 1 – Cement (component 1)



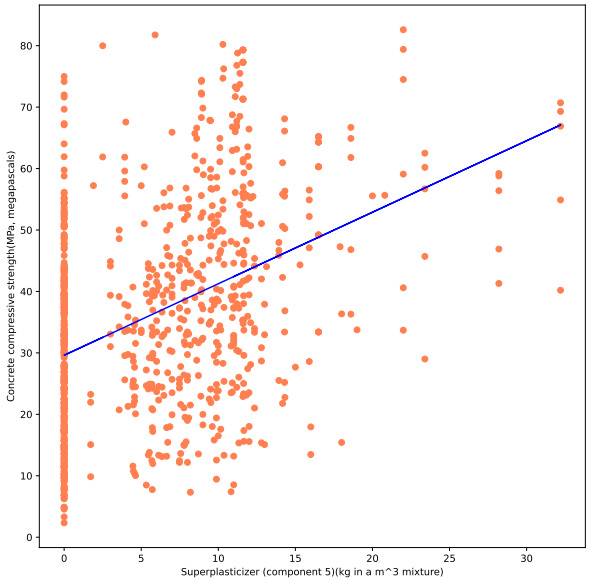
(b) Feature 2 – Blast Furnace Slag (component 2)



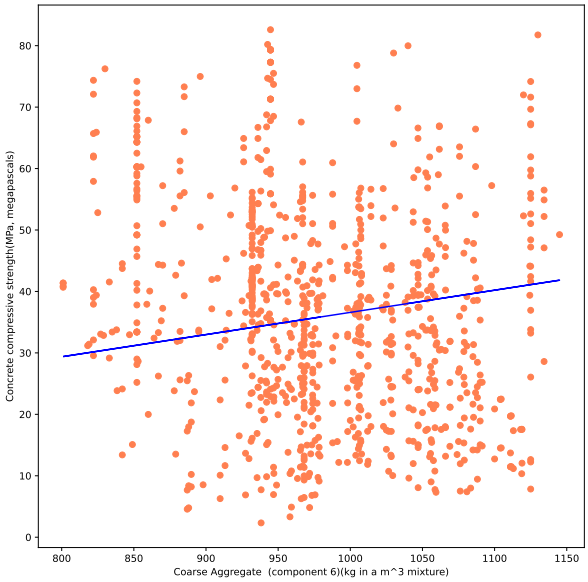
(a) Feature 3 – Fly Ash (component 3)



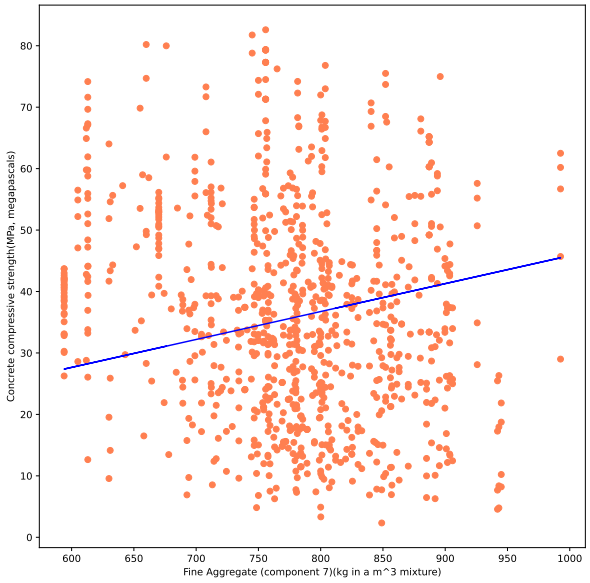
(b) Feature 4 – Water (component 4)



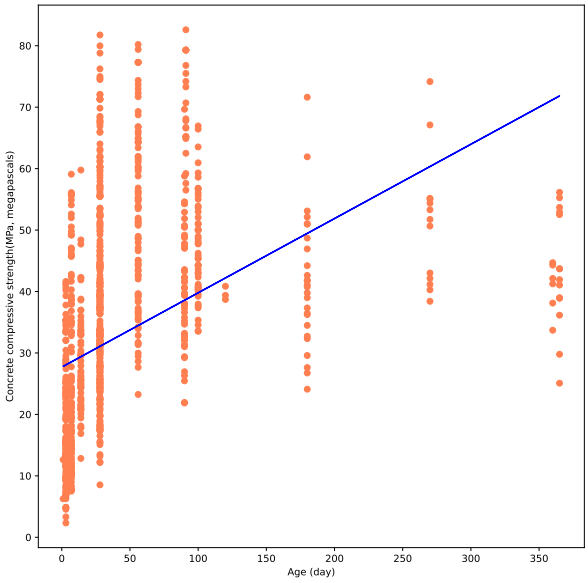
(a) Feature 5 – Superplasticizer (component 5)



(b) Feature 6 – Coarse Aggregate (component 6)

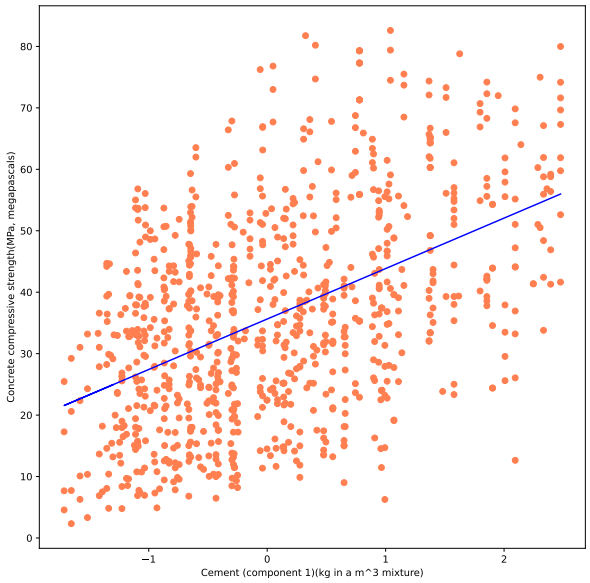


(a) Feature 7 – Fine Aggregate (component 7)

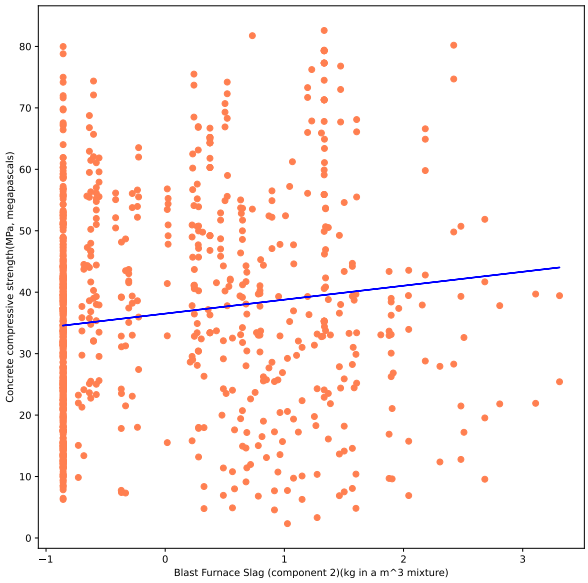


(b) Feature 8 – Age

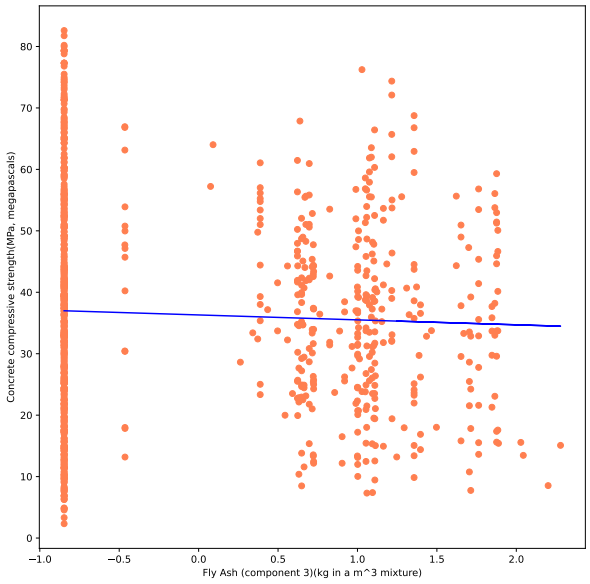
2.3 Scatterplots for the normalized dataset



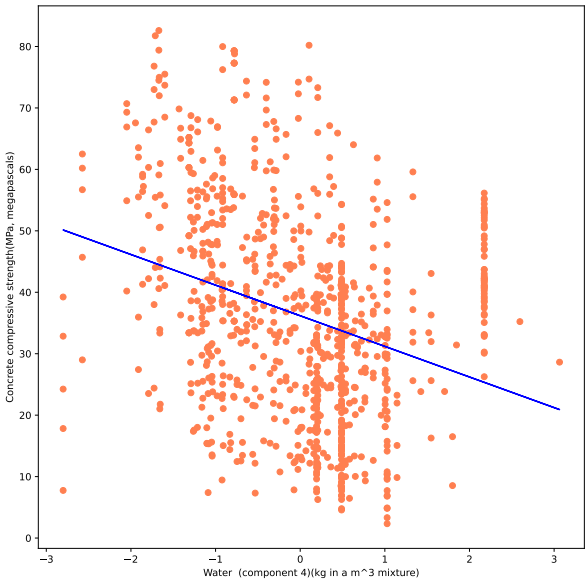
(a) Feature 1 – Cement (component 1)



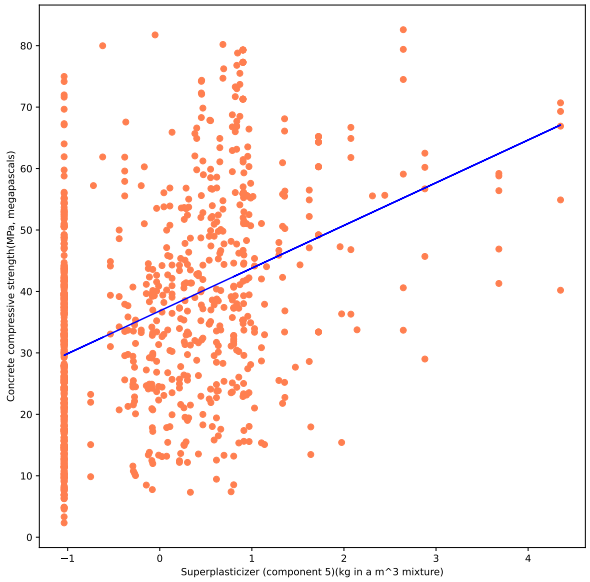
(b) Feature 2 – Blast Furnace Slag (component 2)



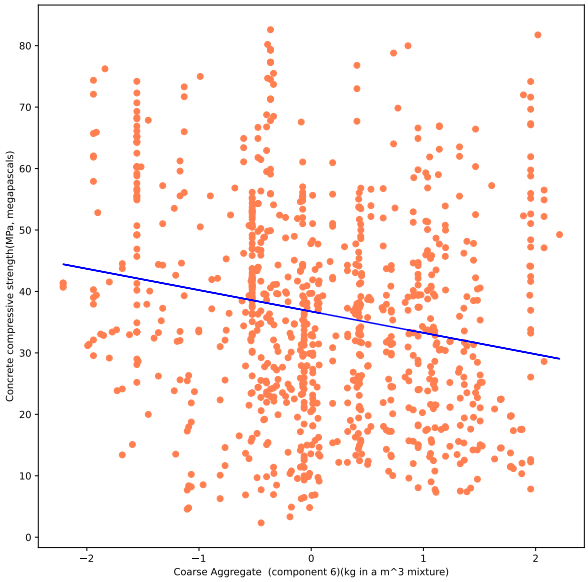
(a) Feature 3 – Fly Ash (component 3)



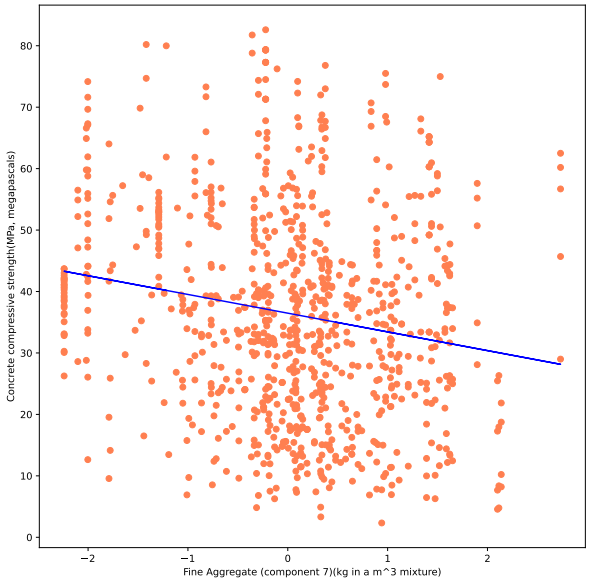
(b) Feature 4 – Water (component 4)



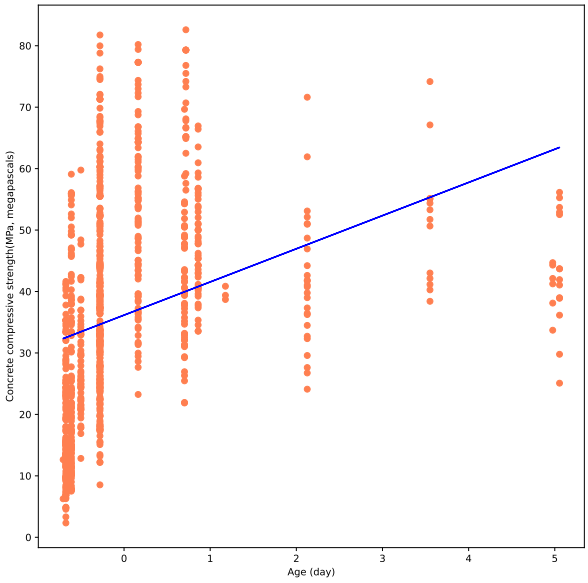
(a) Feature 5 – Superplasticizer (component 5)



(b) Feature 6 – Coarse Aggregate (component 6)



(a) Feature 7 – Fine Aggregate (component 7)



(b) Feature 8 – Age

3 Discussion

Table 3 shows the MSE for each of the features in uni-variate and multi-variate regression in the original dataset. We can see for uni-variate regression, training MSE is minimum for Feature 5 – Superplasticizer (component 5). Then Cement (component 1), Age, Fly Ash (component 3), and so on. But this is not the case for the testing data. Minimum MSE for test data is Feature 1 – Cement (component 1). On the other hand, the multi-variate model gives the least MSE, because it considers more variables. if we look the scatterplots, we will

get the same results. Feature 5, 1, and 8 have the highest positive correlations in the plots. Table 4 shows the MSE for each of the features in uni-variate and multi-variate regression in the normalized dataset. It's possible that the Feature – Cement, which has the strongest correlation with the output variable, Concrete Strength, is the explanation for Feature 1 having low MSE in both original and normalized datatsets.

Table 3: MSE for uni-variate and multi-variate regression in original dataset.

Input Features	MSE (Training Data)	MSE (Test Data)
Cement (component 1)	247.79	108.43
Blast Furnace Slag (component 2)	380.35	131.95
Fly Ash (component 3)	310.19	178.20
Water (component 4)	354.53	185.24
Superplasticizer (component 5)	244.30	235.33
Coarse Aggregate (component 6)	322.18	165.86
Fine Aggregate (component 7)	332.85	171.24
Age	276.53	144.80
Multi-variate Regression	114.77	59.55

Table 4: MSE for uni-variate and multi-variate regression in normalized dataset.

Input Features	MSE (Training Data)	MSE (Test Data)
Cement (component 1)	228.33	81.13
Blast Furnace Slag (component 2)	290.77	159.69
Fly Ash (component 3)	295.26	149.46
Water (component 4)	270.10	154.99
Superplasticizer (component 5)	244.30	235.33
Coarse Aggregate (component 6)	284.48	190.26
Fine Aggregate (component 7)	286.31	169.00
Age	262.91	150.92
Multi-variate Regression	114.77	59.55

Table 5 shows the coefficients of uni-variate and multi-variate regression for both the original and normalized datasets. The coefficient corresponding to each input variable may be used to check whether input features are responsible for a specific correct or incorrect prediction. For example, if a coefficient corresponds to an input variable that is close to zero, it means that variable has no influence on the prediction; similarly, if a coefficient corresponds to an input variable that is very high, it means that variable has a significant impact on the prediction or output variable.

Table 5: Coefficients for uni-variate and multi-variate regression in original and normalized datasets.

Input Features	Coefficient (uni-variate)		Coefficient (multi-variate)	
	Original Data	Normalized Data	Original Data	Normalized Data
Cement (component 1)	0.118	8.211	0.018	35.832
Blast Furnace Slag (component 2)	0.092	2.269	0.113	11.599
Fly Ash (component 3)	2.893	-0.802	0.097	8.182
Water (component 4)	0.186	-4.983	0.091	5.517
Superplasticizer (component 5)	1.163	6.952	-0.194	-3.751
Coarse Aggregate (component 6)	0.036	-3.476	0.186	1.862
Fine Aggregate (component 7)	0.045	-3.046	0.011	1.034
Age	0.114	5.399	0.011	0.838