CSE514 Data Mining
Programming Assignment 1

Xingjian Ding 502558

February 22, 2022

1 Introduction

1.1 description

The purpose of this project is to design several different models for regression of the data from https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength. The data has 900 different records each including 8 different components and one result. In this project, we are trying to create an algorithm and let the first 900 data train the algorithm until reaching our proposal. Then use the trained algorithm to test the remaining data.

1.2 details of algorithms

The three algorithms I implement are Uni-variate linear regression, Multi-variate linear regression, and Multi-variate polynomial regression. These three algorithms are similar. For example, in Uni-variate linear regression, the initial algorithm is y=mx+b(m=0, b=0), then use the training dataset to update m and b. Until the MSE loses value small enough or the number of iteration becomes too large, then the program stop.
When to stop: Until MSE loss value is small enough(< 0.00001 may update for different algorithms) or the number of iterations becomes too large(> 1000000may update for different algorithms).
Learning rate: The start learning rate I choose is 0.00001. In some cases (Component 6,7 in question A and question C), this learning rate is not small enough which update the learning rate smaller.

1.3 pseudo-code

Uni-variate linear regression:

```
define UnivariateLinearRegression()
    initial y = mx + b, MSE loss function
    while MSE loss function still change and iteration is not too large:
        use the training dataset to train algorithm
        update m, b, mse
    use trained m and b to test
    get the variance explained for the response variable which show the
       models'_performance.
```

Multi-variate linear regression:
$y = m_0x_0 + m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + m_5x_5 + m_6x_6 + m_7x_7 + b$

```
define multivariateLinearRegression()
initial a list which has 9 elements and MSE loss function
    while MSE loss function still changing and iteration is not too large
       :
        use the training dataset to train algorithm
        update 9 emelemt in the list
```

```
    use trained list to test
    get the variance explained for the response variable which show the
        models'␣performance.
```

Multi-variate polynomial regression:
The Multi-variate polynomial regression will include 45 elements which is 8! = 36 2 different variables from 8 variables, 8 variables and 1 constant variable. First modified data to fit 45 * 900, then training the polynomial regression as linear regression.

```
define multiVariatePolynomialRegression()
initial a list which has 45 elements and MSE loss function
    while MSE loss function still changing and iteration is not too large
        :
        use the training dataset to train algorithm
        update 45 emelemt in the list
    use trained list to test
    get the variance explained for the response variable which show the
        models'␣performance.
```

2 Result

2.1 Variance explained on the training dataset

Variance explained of your models on the training dataset when using only one of the predictor variables (uni-variate regression) and when using all eight (multi-variate regression):
A:
Cement (component 1)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: 0.2291154764014922

Blast Furnace Slag (component 2)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: 0.01830085677896154

Fly Ash (component 3)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: 0.0031440058701552864

Water (component 4)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: 0.08808359535695587

Superplasticizer (component 5)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: 0.174283029091848

Coarse Aggregate (component 6)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: -1476.3632500672502

Fine Aggregate (component 7)(kg in a m^3 mixture)
Variance explained of your models on the training dataset: -542.9161713409668

Age (day)
Variance explained of your models on the training dataset: 0.11236960854810452

B:
Variance explained of your models on the training dataset: 0.61037889178313

2.2 Variance explained on the testing data points

Variance explained of your models on the testing data points:
A:
Cement (component 1)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: 0.439820382280396

Blast Furnace Slag (component 2)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -0.10249585105497983

Fly Ash (component 3)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -0.032806706896149285

Water (component 4)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -0.06948449302971715

Superplasticizer (component 5)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -0.7030131810400233

Coarse Aggregate (component 6)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -2651.075820409897

Fine Aggregate (component 7)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -1065.0346032495297

Age (day)
Variance explained of your models on the testing data points: -0.04195544661292128

B:
Variance explained of your models on the testing data points: 0.5932375335429172

2.3 Plots

when leaning rate is 0.00001:

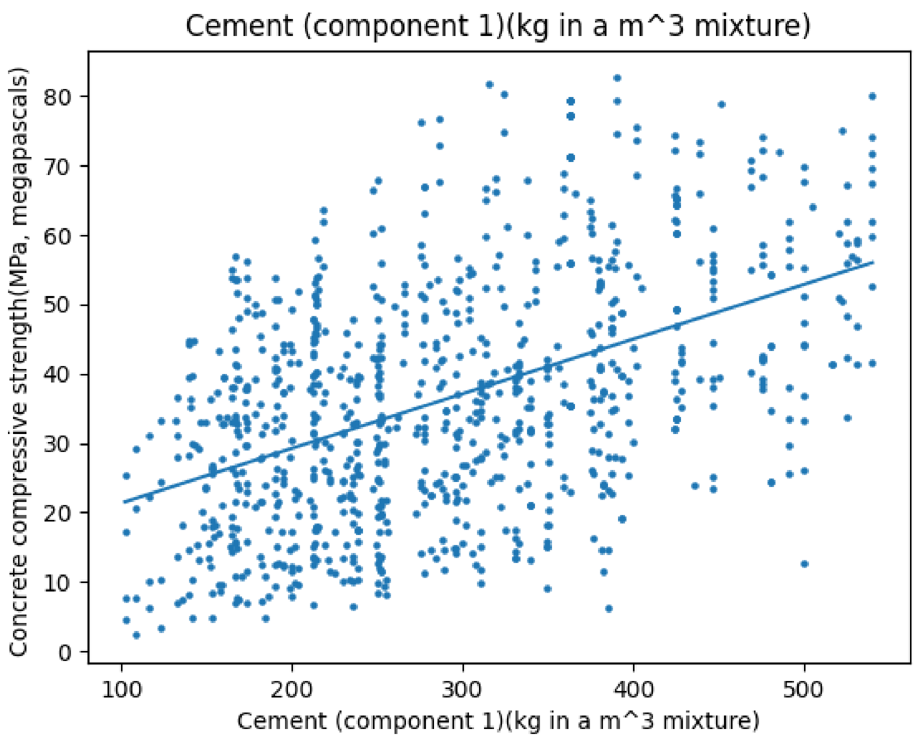

Figure 1: Cement (component 1)(kg in a m^3 mixture)

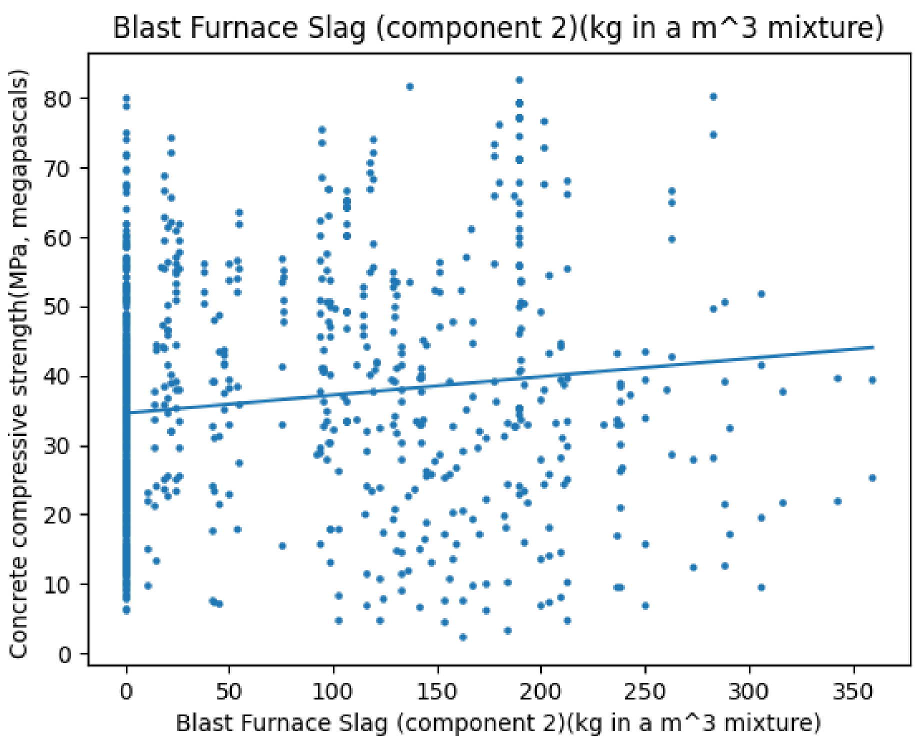

Figure 2: Blast Furnace Slag (component 2)(kg in a m^3 mixture)

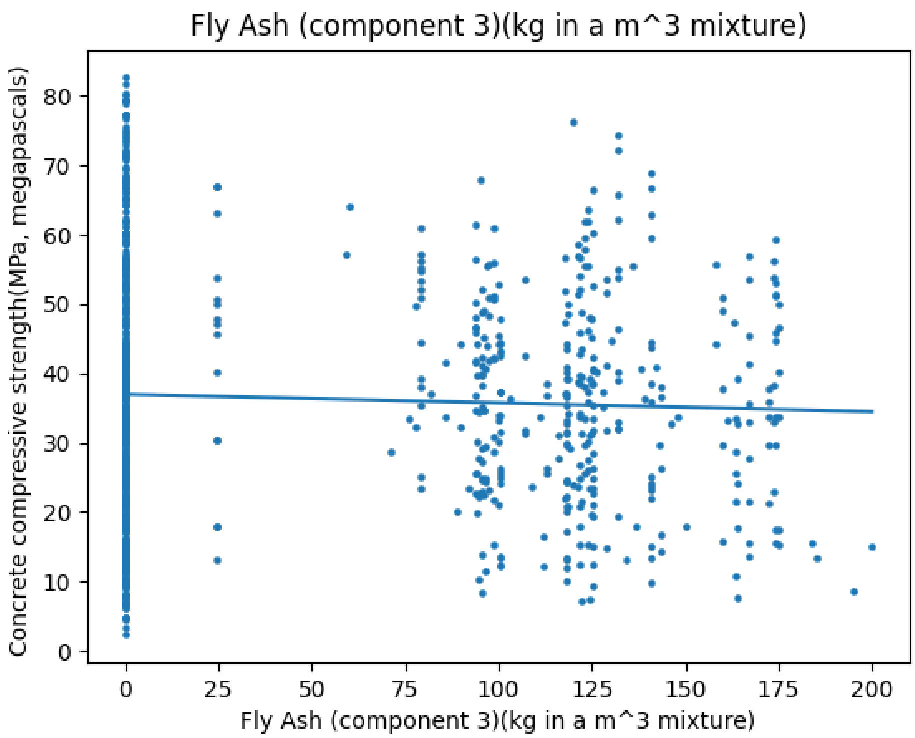

Figure 3: Fly Ash (component 3)(kg in a m^3 mixture)

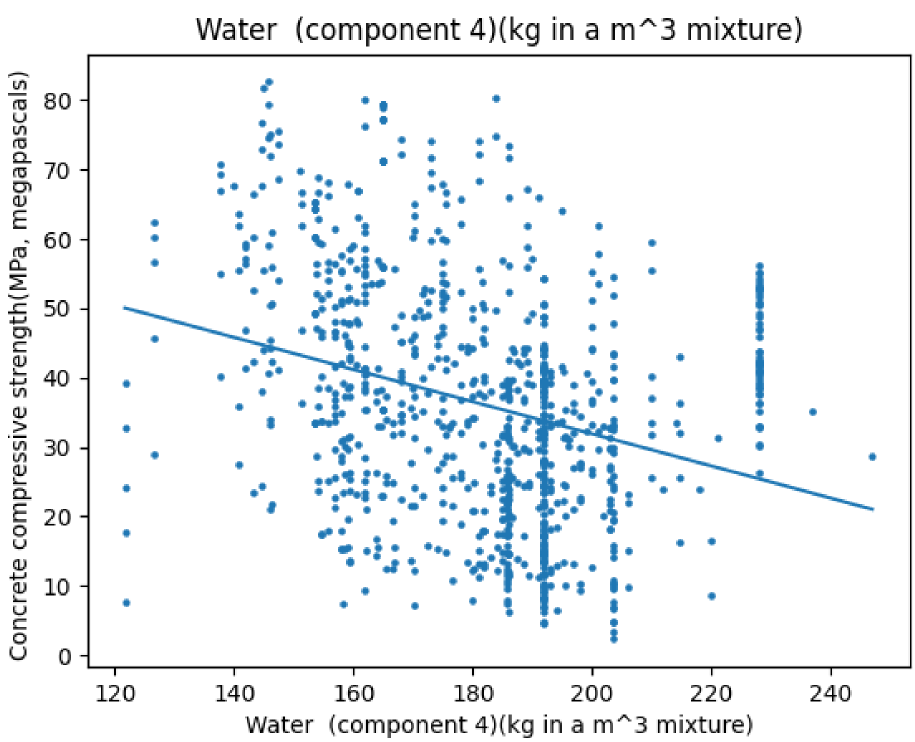

Figure 4: Water (component 4)(kg in a m^3 mixture)

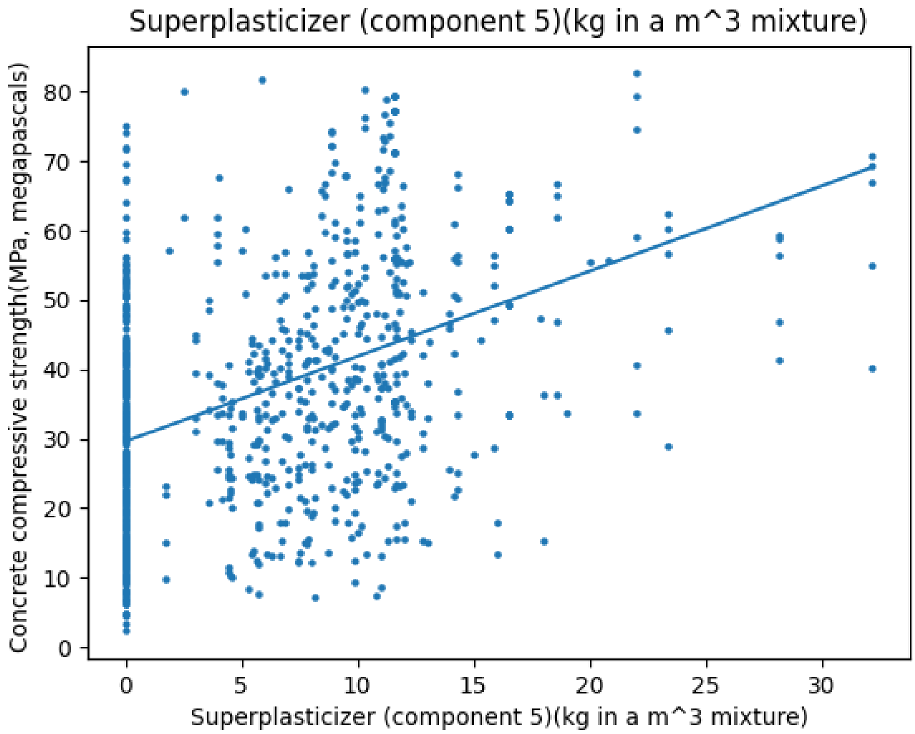

Figure 5: Superplasticizer (component 5)(kg in a m^3 mixture)

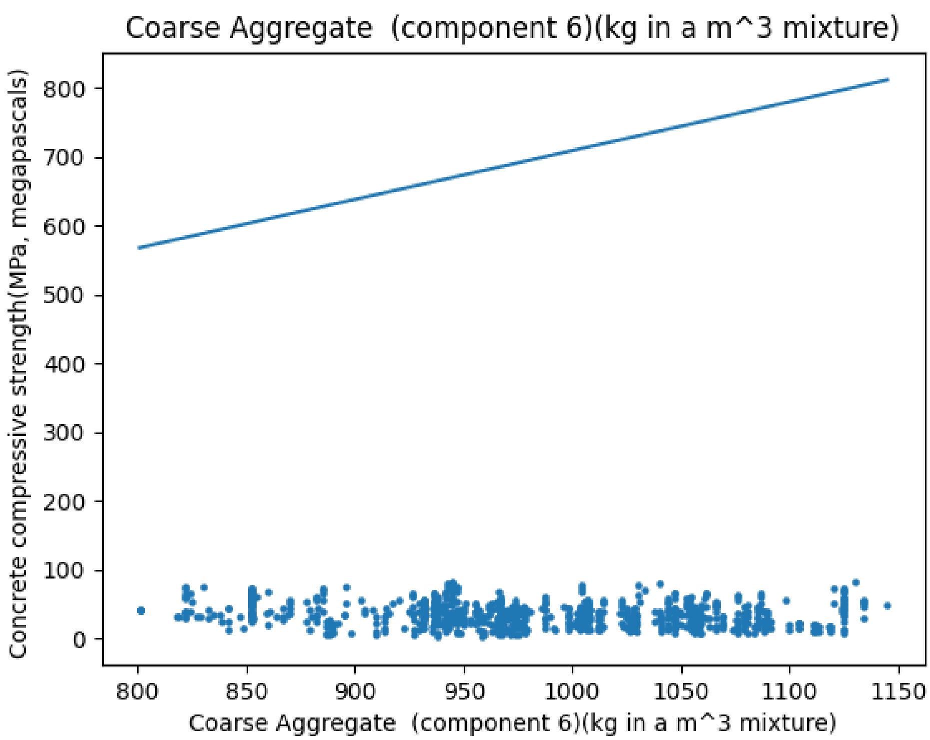

Figure 6: Coarse Aggregate (component 6)(kg in a m^3 mixture)

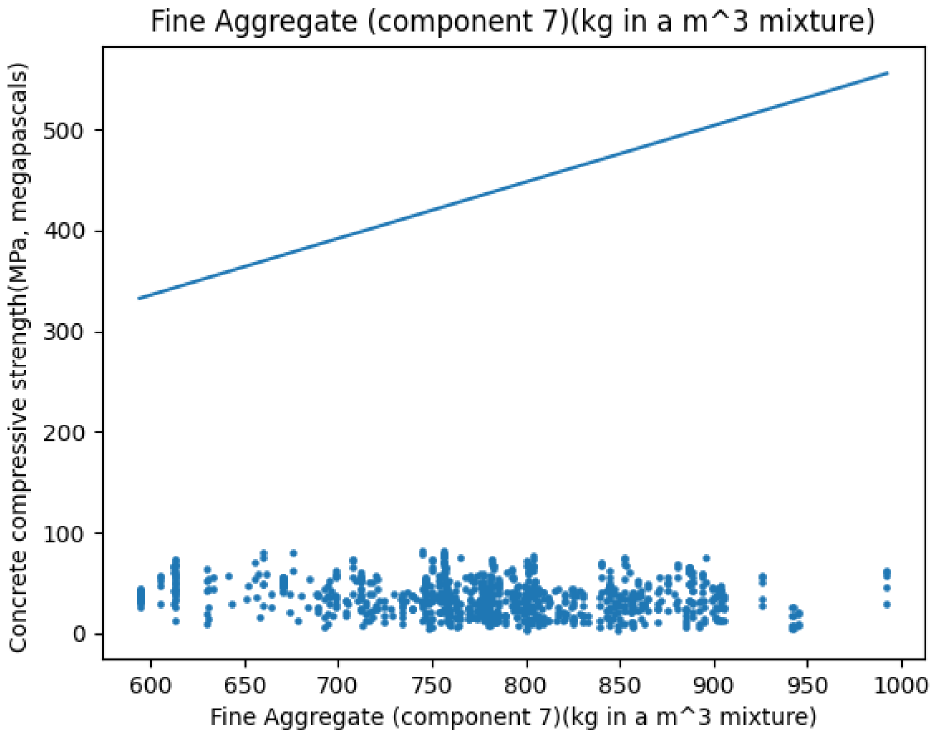

Figure 7: Fine Aggregate (component 7)(kg in a m^3 mixture)

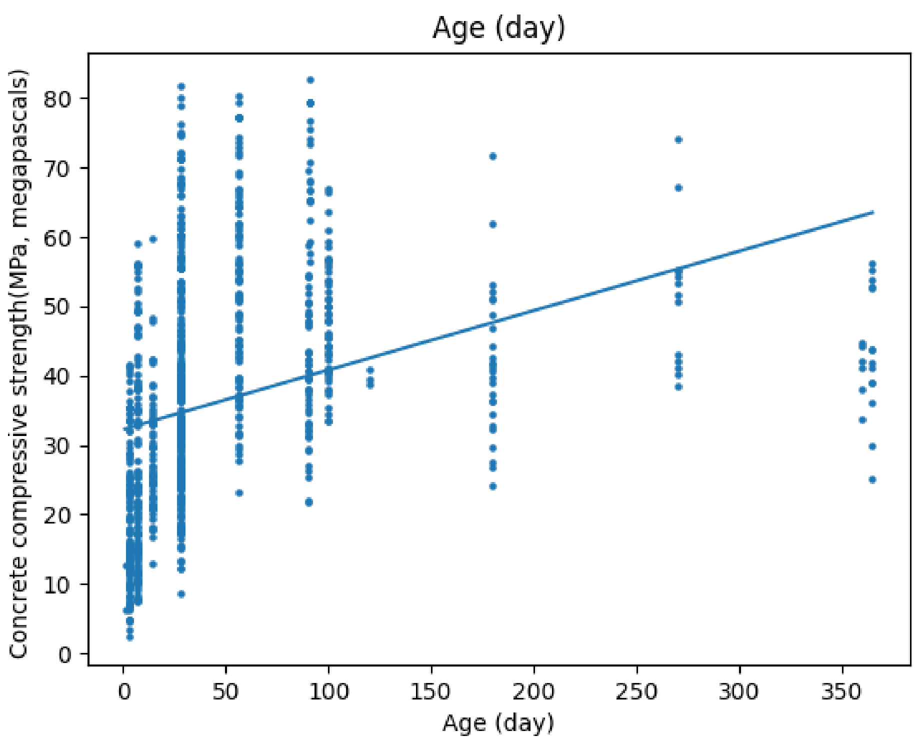

Figure 8: Age (day)

2.4 Optional extension 1 – Multi-variate polynomial regression

When alpha = 0.0000000000000000001:
Stop after 1000000 iteration
[6.77356880e-07 1.32763279e-07 7.31353435e-08 3.36055432e-07
1.45348714e-08 1.83831418e-06 1.44785054e-06 1.15862788e-07
7.76757147e-08 9.12719023e-09 8.06593244e-08 3.69206015e-09
4.28333547e-07 3.37345901e-07 2.62134294e-08 3.53318613e-08
4.77455625e-08 2.54214576e-09 2.77054153e-07 2.21779310e-07
1.26252209e-08 1.90280755e-07 7.02930320e-09 1.02420391e-06
8.03170588e-07 6.86002086e-08 5.79302145e-10 4.12937084e-08
3.46718501e-08 1.85948535e-09 5.65547255e-06 4.42652063e-06
3.55704887e-07 3.54715396e-06 2.72890572e-07 5.30240015e-08
1.90355205e-09 4.51914276e-10 2.81790321e-10 1.05532504e-09
4.34794499e-11 5.78444004e-09 4.56834735e-09 3.65076083e-10
0.00000000e+00]
Variance explained of your models on the testing data points: -3.0709957414552695
Variance explained of your models on the training dataset: -1.9775765867772015

3 Discussion

Csomponent 6 and 7 has bad result, change learning rate to 0.000000001 to recalculate. Get:
Coarse Aggregate (component 6)(kg in a m^3 mixture)
Stop after 3390 iteration
Variance explained of your models on the testing data points: -0.144582418508477742
Variance explained of your models on the training dataset: -0.0883013335118199

Fine Aggregate (component 7)(kg in a m^3 mixture)
Variance explained of your models on the testing data points: -0.18164807773729796
Variance explained of your models on the training dataset: -0.12574959255206042

Work! New plots for component 6 and 7:

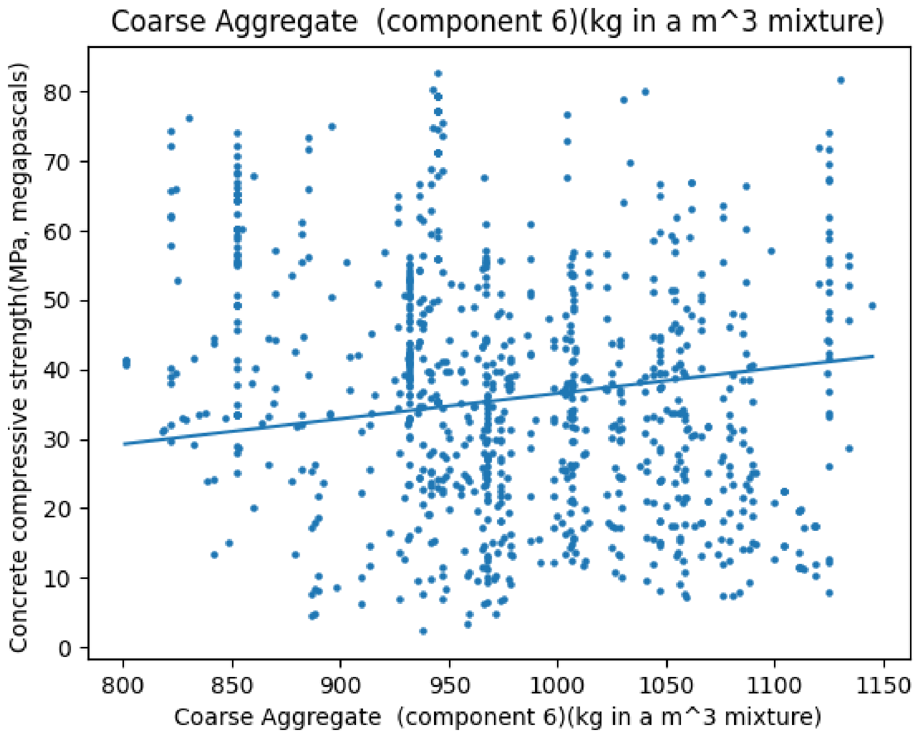

Figure 9: Coarse Aggregate (component 6)(kg in a m^3 mixture)

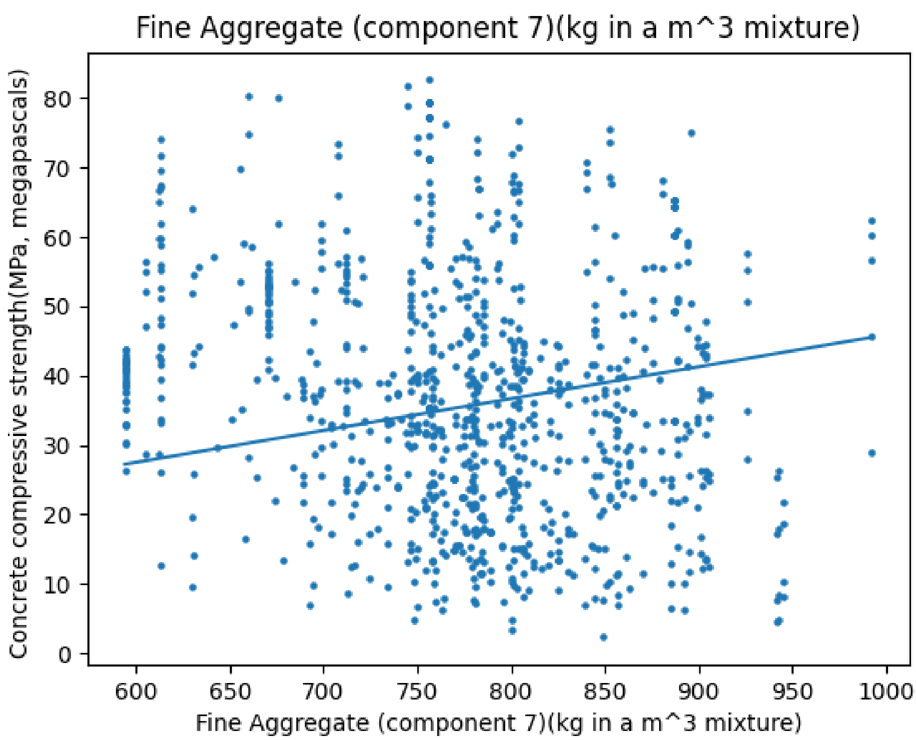

Figure 10: Fine Aggregate (component 7)(kg in a m^3 mixture)s

the variance explained for the response variable, which is:
$1 - \frac{MSE}{Variance(observed)}$

In my training models except model 6 and 7, all other model has good performance on the training data. However, on the testing data, only the first component in question A, question B has good performance. Other models need to improve by changing the learning rate.
For the univariate models, my learning rate is 0.00001 which is worked in the multi-variate model(s). However, in the multi-variate polynomial model(s), I first use 0.00001 as my training rate and failed. The MSE keeps increasing to infinite. Until I change the learning rate to 10^{-13}, I got MSE decreasing. After this assignment, I suggest starting with a large learning rate like 0.1 or 0.01 and calculating MSE. If MES increases to infinite, change the learning rate to a smaller one. Also, changing the learning rate can not get a good performance. Trying to change stopping conditions will help to improve