CSE514 – Fall 2022 Programming Assignment 1

Introduction

In this assignment, I used UCI Machine Learning Repository: Concrete Compressive Strength Data Set as the original dataset, then I implemented univariate and multivariate linear regression on it. This dataset has eight input features (Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, Age) and one output value (Concrete compressive strength). The goal of regression is to find a well fit model that can predict the output value based on the input features. In detail, I need to try to find a model with small Mean Square Error (MSE) and Mean Absolute Error (MAE), and its variance explained (VE) need to be as close to 0 as possible.

Details of Algorithm

Before I implement the regressions, I used the sk-learn package's train_test_split to randomly split the dataset into the test and train datasets (In the code file, I used the random seed 13). For the D part of the two files, I used the sk-learn package's preprocessing to define a normalize function to normalize the original dataset and then split the new dataset to implement regressions. (The difference between normalized data and original data will talk in the Result section)

Univariate Linear Regression Formula:

$$Y = mX + b$$

Multivariate Linear Regression Formula:

$$\vec{Y} = f(\vec{m} \cdot \vec{X})$$

Normalization Formula:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

For both univariate and multivariate linear regression, they are all the stochastic gradient descent algorithm. For univariate regression (y = mx + b), first I choose the random numbers between 0

and 1 as the initial value of parameters m and b. For multivariate linear regression, I add a bias parameter matrix consists of numbers between 0 and 1 to satisfy the calculation need. Then I updated the parameters based on their derivates and learning rate to find the fittest ones of the model. I set the max iterations as the stop criterion. During this process, I used MSE and MAE to evaluate the difference between the predicted y values and observed y values. Further, I also calculated the MSE, MAE, and VE of the train and test dataset to see if the final model I got well fits the original dataset or maybe overfitting or underfitting.

In the beginning, I choose 1000000 as the maximum number of iterations and 0.000000001 as the learning rate, but I find the model got from this value cannot fit the data well, so I add a while loop to dynamically adjust the learning rate dynamically. After that, I constantly adjusted various parameters (learning rate, the maximum number of iterations, learning rate dynamic update rate), and finally got the algorithm to searching the well fit models of dataset.

Evaluation formulas

Mean squared error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\, Y_i - \hat{Y}_i \,\right|$$

Variance Explained (VE):

$$\text{Variance Explained} = 1 - \frac{\text{MSE}}{\text{Variance (Observed)}}$$

Pseudo-code of algorithms

Uni-variate linear regression （MSE）:

1. Set learning rate, Max Iteration.

2. Set a random number between 0 and 1 to m, b.

3. Calculate MSE and set Iteration as 0.

4. *Repeat*
5. $previous\ MSE \leftarrow MSE$
6. $m_{new} = m - learningrate * \left(-\frac{2}{n}\sum_{i=1}^{n} x_i\left(Y_i - \hat{Y}_i\right)\right)$
7. $b_{new} = b - learning\ rate * \left(-\frac{2}{n}\sum_{i=1}^{n} Y_i - \hat{Y}_i\right)$
8. *Calculate new MSE based on* m_{new} *and* b_{new}
9. *Compare the new MSE and previous MSE, if the new MSE is bigger decreases the learning rate, and else increases the learning rate.*
10. *Iteration adds 1.*
11. $Until$ *stopping criterion (Iteration > Max Iteration) is satisfied.*

Uni-variate linear regression （MAE）:

1. *Set learning rate, Max Iteration.*
2. *Set a random number between 0 and 1 to m, b.*
3. *Calculate MAE and set Iteration as 0.*
4. *Repeat*
5. $previous\ MAE \leftarrow MAE$
6. $m_{new} = m - learningrate * \left(-\frac{2}{n}\sum_{i=1}^{n} x_i\left(Y_i - \hat{Y}_i\right)\right)$
7. $b_{new} = b - learning\ rate * \left(-\frac{2}{n}\sum_{i=1}^{n} Y_i - \hat{Y}_i\right)$
8. *Calculate new MAE based on* m_{new} *and* b_{new}
9. *Compare the new MAE and previous MAE, if the new MAE is bigger decreases the learning rate, and else increases the learning rate.*
10. *Iteration adds 1.*
11. $Until$ *stopping criterion (Iteration > Max Iteration) is satisfied.*

Multi-variate linear regression （MSE）:

1. *Set learning rate, Max Iteration.*
2. *Set a random matrix* $\vec{m}$ *consist of numbers between 0 and 1.*
3. *Calculate MSE and set Iteration as 0.*
4. *Repeat*
5. $previous\ MSE \leftarrow MSE$
6. $\vec{m}_{new} = \vec{m} - learningrate * \left(-\frac{2}{n}\sum_{i=1}^{n} \vec{x_i}\left(\vec{Y_i} - \vec{\hat{Y}_i}\right)\right)$
7. *Calculate new MSE based on* m_{new} *and* b_{new}
8. *Compare the new MSE and previous MSE, if the new MAE is bigger decreases the learning rate, and else increases the learning rate.*
9. *Iteration adds 1.*
10. *Until stopping criterion (Iteration > Max Iteration) is satisfied.*

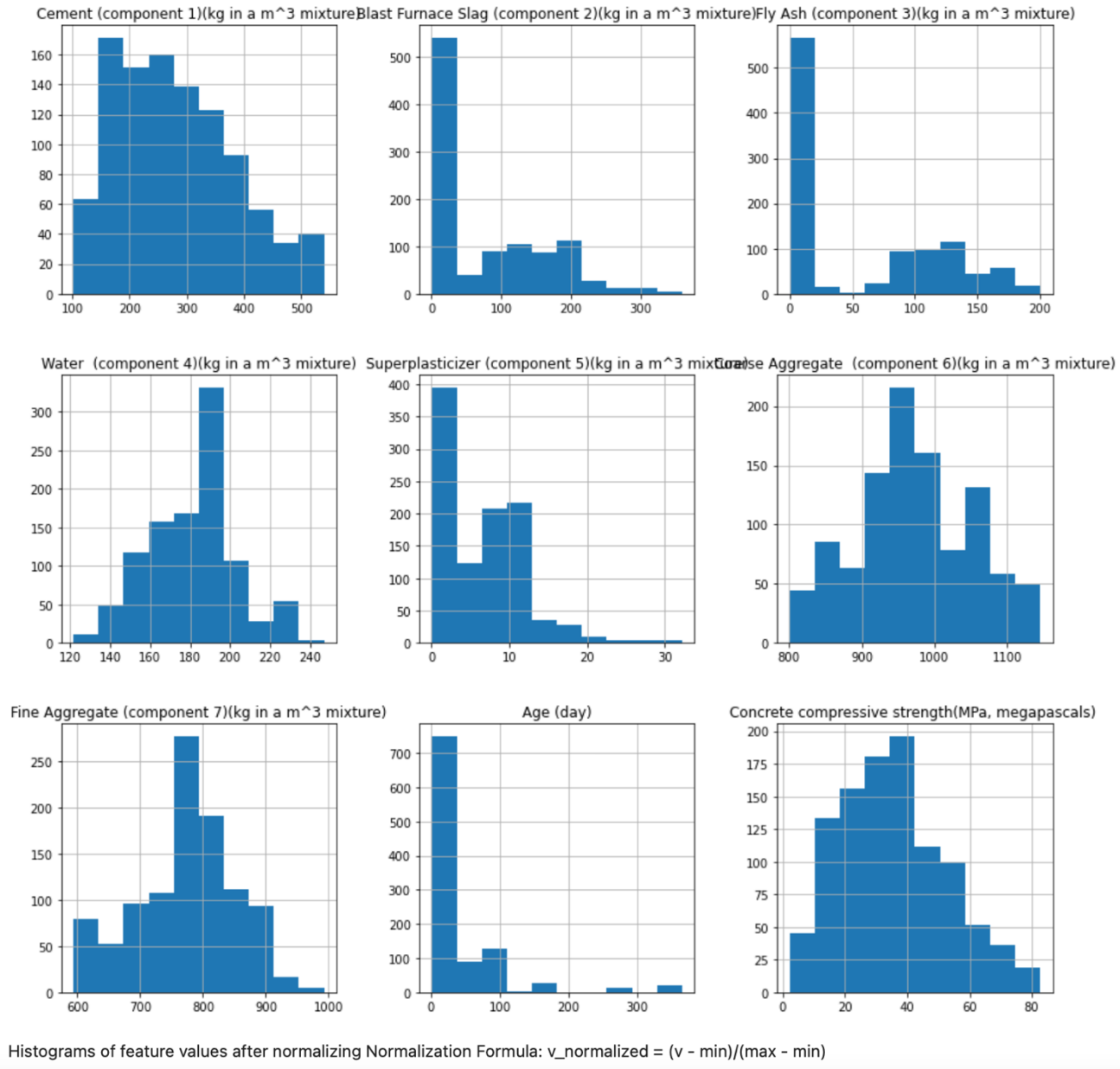
Multi-variate linear regression （MAE）:

1. *Set learning rate, Max Iteration.*
2. *Set a random matrix* $\vec{m}$ *consist of numbers between 0 and 1.*
3. *Calculate MAE and set Iteration as 0.*
4. *Repeat*
5. $previous\ MAE \leftarrow MAE$
6. $\vec{m}_{new} = \vec{m} - learningrate * \left(-\frac{2}{n}\sum_{i=1}^{n} \vec{x_i}\left(\vec{Y_i} - \vec{\hat{Y}_i}\right)\right)$
7. *Calculate new MAE based on* m_{new} *and* b_{new}
8. *Compare the new MAE and previous MAE, if the new MAE is bigger decreases the learning rate, and else increases the learning rate.*
9. *Iteration adds 1.*
10. *Until stopping criterion (Iteration > Max Iteration) is satisfied.*

Result

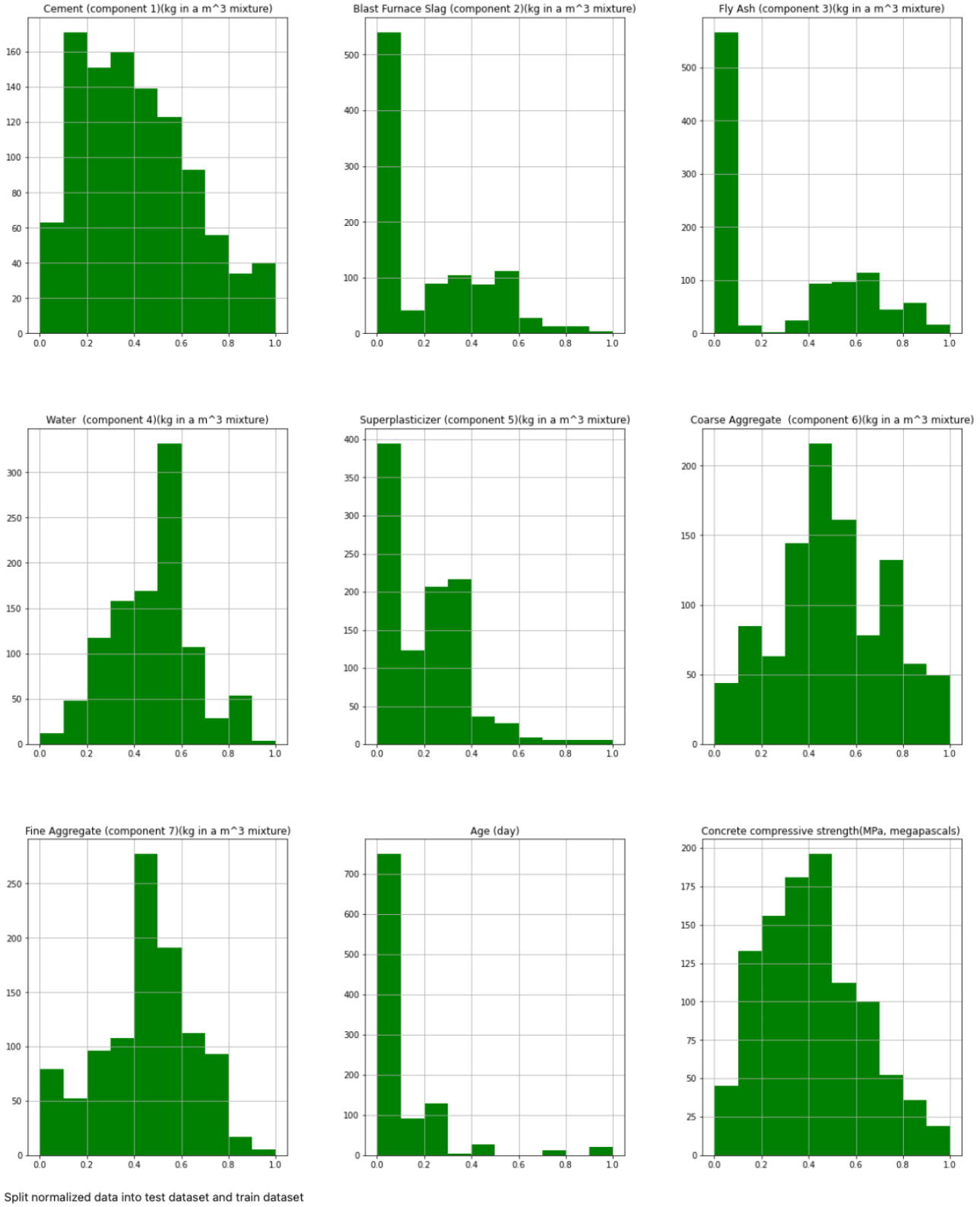
Normalized Data

Following are the histograms of original data:



Histograms of feature values after normalizing Normalization Formula: v_normalized = (v - min)/(max - min)

Following are the histograms of normalized data:



Split normalized data into test dataset and train dataset

From above histograms, the original and normalized data have similar distribution. The difference between is just the size of data. Moreover, due to the small size of normalized data, training normalized data spend less time than training original data.

Univariate linear regression (Original Data, MSE)

	m	b	Train MSE	Test MSE	Train Variance Explained	Test Variance Explained
Cement (component 1)	0.083748046	12.29102358	207.4302387	226.9945818	-1.584239615	-1.940871978
Blast Furnace Slag (component 2)	0.027356868	33.86109375	275.9804652	258.360084	-2.438262692	-2.347233776
Fly Ash (component 3)	-0.034344314	37.75959462	276.6388656	270.6495509	-2.446465278	-2.506452328
Water (component 4)	-0.048260923	44.19989486	272.5969132	252.429863	-2.396109199	-2.270403656
Superplasticizer (component 5)	1.027164771	29.43478009	243.2363211	229.0063021	-2.030324511	-1.966935207
Coarse Aggregate (component 6)	0.035291835	1.157060366	304.2757231	282.626278	-2.790775069	-2.66161912
Fine Aggregate (component 7)	0.04223844	2.601380572	310.8560769	299.8320602	-2.872755455	-2.884531942
Age (day)	0.086694603	31.88103544	250.2947486	237.2916391	-2.118260909	-2.074277484

Univariate linear regression (Original Data, MAE)

	m	b	Train MAE	Test MAE	Train Variance Explained	Test Variance Explained
Cement (component 1)	0.090562635	10.00027888	11.88701713	11.83962321	0.851907317	0.846609484
Blast Furnace Slag (component 2)	0.15908499	10.00404929	20.300987	20.86307294	0.747083091	0.729704445
Fly Ash (component 3)	-0.031295893	37.36407091	13.44154643	12.94593524	0.832540439	0.832276445
Water (component 4)	0.135932343	10.00079167	14.30713192	13.39267706	0.821756667	0.826488596
Superplasticizer (component 5)	1.138087081	28.10640107	12.73844311	11.5917877	0.84129995	0.849820364
Coarse Aggregate (component 6)	0.026250328	10.00002707	13.86389539	12.95369639	0.827278665	0.832175894
Fine Aggregate (component 7)	0.032550011	10.0000417	14.09557129	13.37312858	0.824392365	0.82674186
Age (day)	0.098514231	30.27767043	12.69096692	11.52441825	0.841891425	0.850693182

Univariate linear regression (Normalized Data, MSE)

	m	b	Train MSE	Test MSE	Train Variance Explained	Test Variance Explained
Cement (component 1)	0.443979635	0.237015323	0.032184051	0.035135971	0.967815949	0.963461402
Blast Furnace Slag (component 2)	0.122491285	0.392803043	0.042835098	0.040100228	0.957164902	0.958298972
Fly Ash (component 3)	-0.085617533	0.441371956	0.042937289	0.042007684	0.957062711	0.956315371
Water (component 4)	-0.352439013	0.586254853	0.039992808	0.03723736	0.960007192	0.961276126
Superplasticizer (component 5)	0.412056434	0.33765846	0.037752859	0.035544209	0.962247141	0.963036867
Coarse Aggregate (component 6)	-0.157482877	0.496871034	0.042450078	0.039672881	0.957549922	0.958743379
Fine Aggregate (component 7)	-0.156509787	0.488979746	0.042679602	0.037899839	0.957320398	0.960587201
Age (day)	0.393146252	0.369214844	0.038848402	0.036830182	0.961151598	0.961699559

Multivariate linear regression

$\overrightarrow{M}$:

Original Data, MSE:

$$\begin{pmatrix} 0.11053072 \\ 0.11258366 \\ 0.09666151 \\ 0.07548495 \\ -0.18196253 \\ 0.27267005 \\ 0.01117704 \\ 0.01052633 \\ 0.11085249 \end{pmatrix}$$

Original Data, MAE:

$$\begin{pmatrix} 0.52754337 \\ 0.11246958 \\ 0.09652626 \\ 0.07533291 \\ -0.18254003 \\ 0.27215925 \\ 0.01103613 \\ 0.01036944 \\ 0.11084857 \end{pmatrix}$$

Normalized Data, MSE:

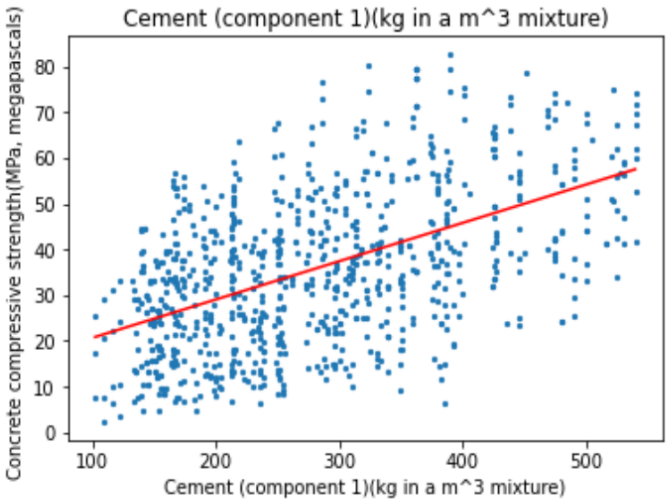
$$\begin{pmatrix} -0.22531384 \\ 0.71329528 \\ 0.52907059 \\ 0.24842999 \\ -0.14069416 \\ 0.14195362 \\ 0.14388145 \\ 0.1761126 \\ 0.5055289 \end{pmatrix}$$

	Train MSE	Test MSE	Train Variance Explained	Test Variance Explained
Original Data	105.2245845	122.3809484	-0.310925261	-0.585529922
Normalized Data	0.016228432	0.01995909	0.983771568	0.999741416
	Train MAE	Test MAE	Train Variance Explained	Test Variance Explained
Original Data	8.199316147	8.649318488	0.897850006	0.88794209

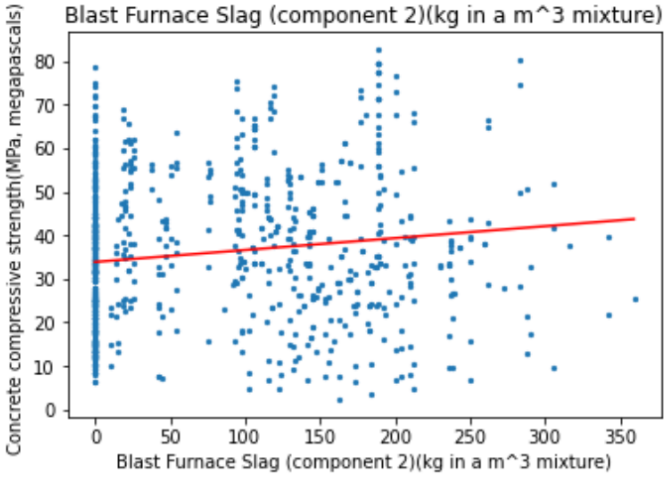
Plots of trained univariate linear regression models

Univariate linear regression with original data and MSE:

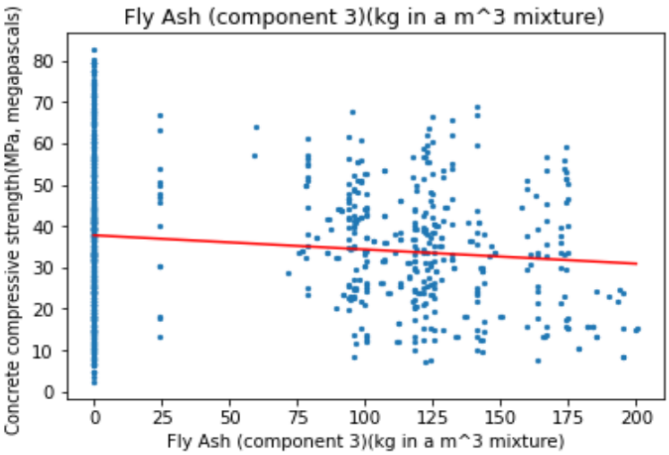
```
Cement (component 1)(kg in a m^3 mixture)
y = 0.08374804559619671x + 12.291023579453011
Train MSE 207.43023871634782
Explained varianvce for train dataset:  -1.584239614924241
Test MSE 226.99458183781022
Explained varianvce for test dataset:  -1.9408719775550773
```



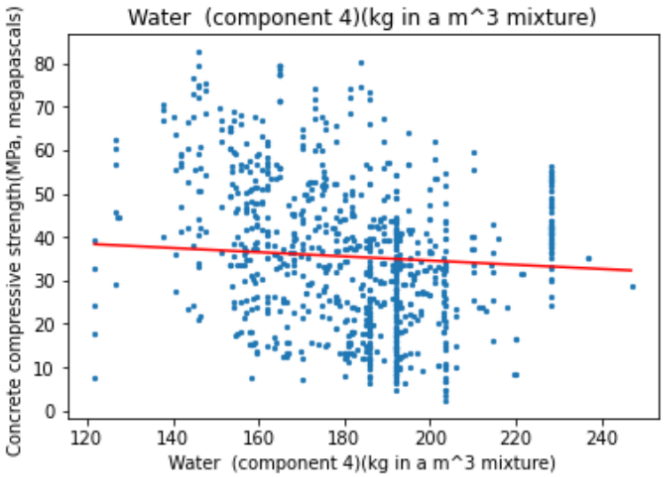

```
Blast Furnace Slag (component 2)(kg in a m^3 mixture)
y = 0.027356867666548355x + 33.861093753662686
Train MSE 275.9804651592843
Explained varianvce for train dataset:  -2.4382626921869015
Test MSE 258.36008403450944
Explained varianvce for test dataset:  -2.3472337758209174
```



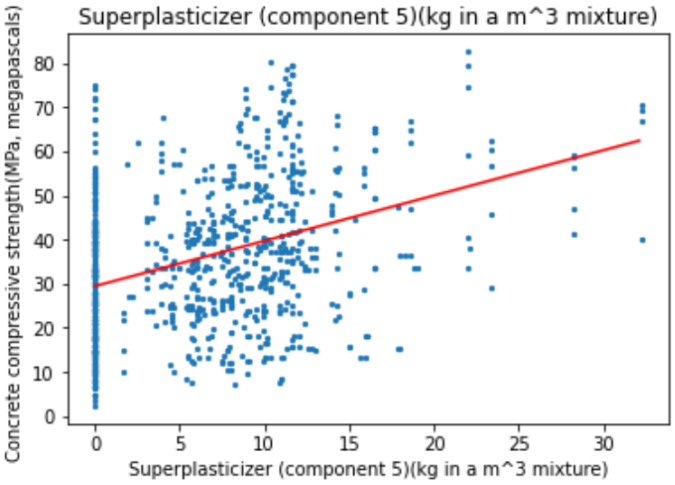
```
Fly Ash (component 3)(kg in a m^3 mixture)
y = -0.03434431355339346x + 37.75959461933475
Train MSE 276.638865555915
Explained varianvce for train dataset:  -2.4464652782610643
Test MSE 270.6495508757912
Explained varianvce for test dataset:  -2.5064523279114757
```



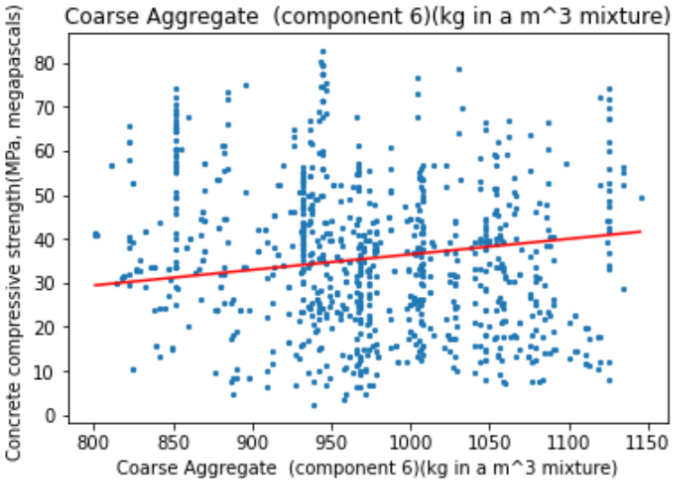

```
Water  (component 4)(kg in a m^3 mixture)
y = -0.048260923152680046x + 44.19989486391541
Train MSE 272.5969131672566
Explained varianvce for train dataset:  -2.3961091992773618
Test MSE 252.42986296912306
Explained varianvce for test dataset:  -2.2704036558651794
```



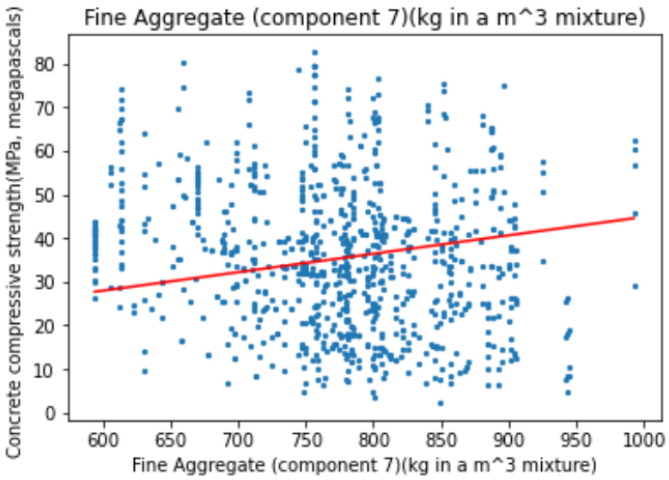
```
Superplasticizer (component 5)(kg in a m^3 mixture)
y = 1.0271647707981104x + 29.434780085335955
Train MSE 243.23632108857566
Explained varianvce for train dataset:  -2.030324511196692
Test MSE 229.0063021130302
Explained varianvce for test dataset:  -1.966935206625016
```



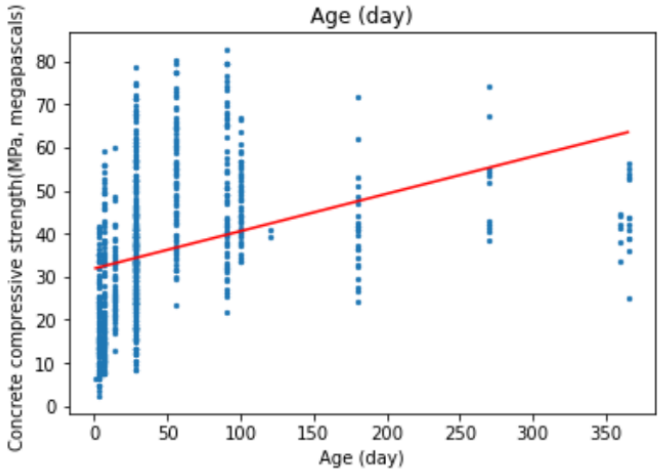

```
Coarse Aggregate  (component 6)(kg in a m^3 mixture)
y = 0.0352918346774269x + 1.1570603655137859
Train MSE 304.27572309815207
Explained varianvce for train dataset:  -2.790775069035258
Test MSE 282.6262779818655
Explained varianvce for test dataset:  -2.6616191201930848
```



```
Fine Aggregate (component 7)(kg in a m^3 mixture)
y = 0.04223844012619321x + 2.601380572020406
Train MSE 310.8560769178466
Explained varianvce for train dataset:  -2.8727554549534684
Test MSE 299.83206022227847
Explained varianvce for test dataset:  -2.884531942310132
```

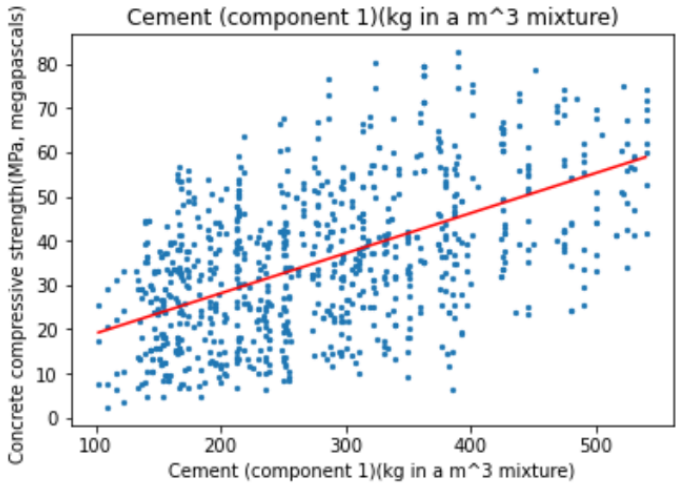



```
Age (day)
y = 0.08669460261125252x + 31.88103544125717
Train MSE 250.29474860872563
Explained varianvce for train dataset:  -2.1182609091371405
Test MSE 237.29163907630306
Explained varianvce for test dataset:  -2.0742774837076485
```

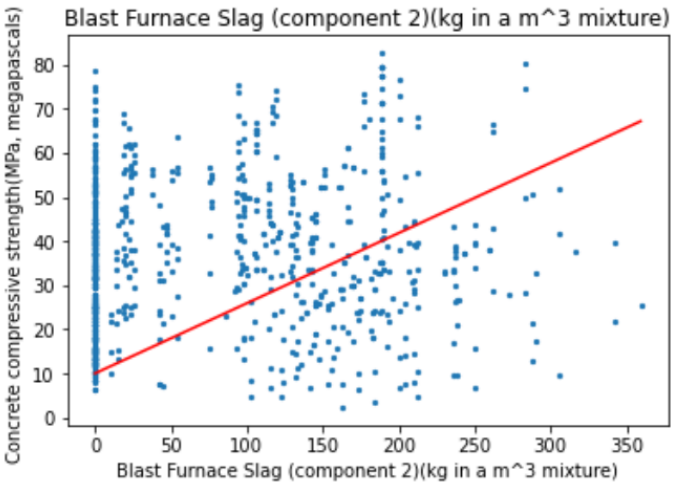


Univariate linear regression with original data and MAE:

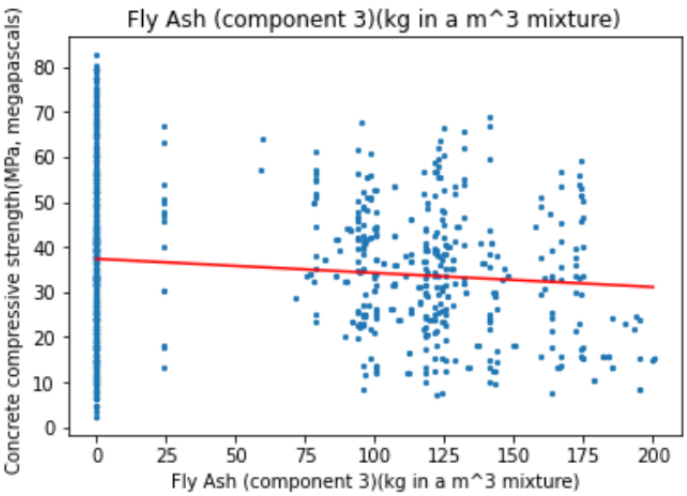
```
Cement (component 1)(kg in a m^3 mixture)
y = 0.0905626345492932x + 10.000278879229878
Train MAE 11.8870171326149
Explained varianvce for train dataset:  0.8519073170455471
Test MAE 11.839623212459472
Explained varianvce for test dataset:  0.8466094835901807
```



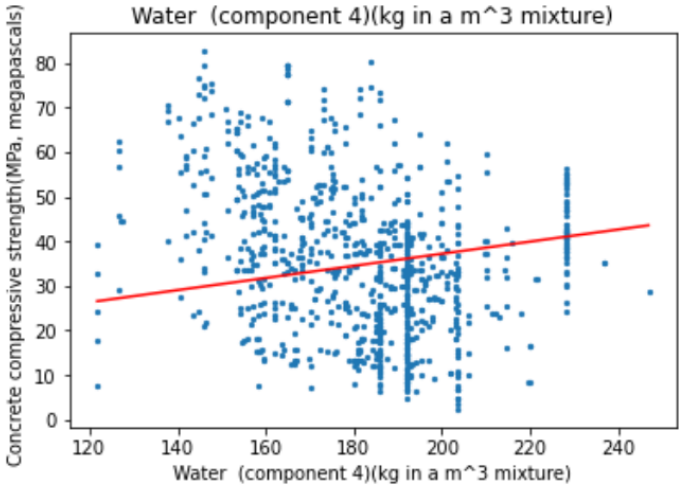

```
Blast Furnace Slag (component 2)(kg in a m^3 mixture)
y = 0.15908499001187737x + 10.004049290168371
Train MAE 20.30098700214903
Explained varianvce for train dataset:  0.7470830908855287
Test MAE 20.86307294275797
Explained varianvce for test dataset:  0.7297044445453609
```



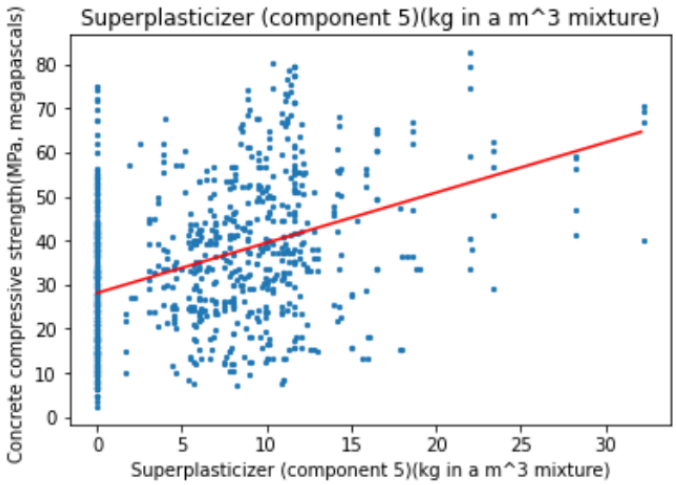
```
Fly Ash (component 3)(kg in a m^3 mixture)
y = -0.03129589324618782x + 37.36407090540438
Train MAE 13.441546428105843
Explained varianvce for train dataset:  0.832540438750327
Test MAE 12.94593524005124
Explained varianvce for test dataset:  0.8322764452681408
```



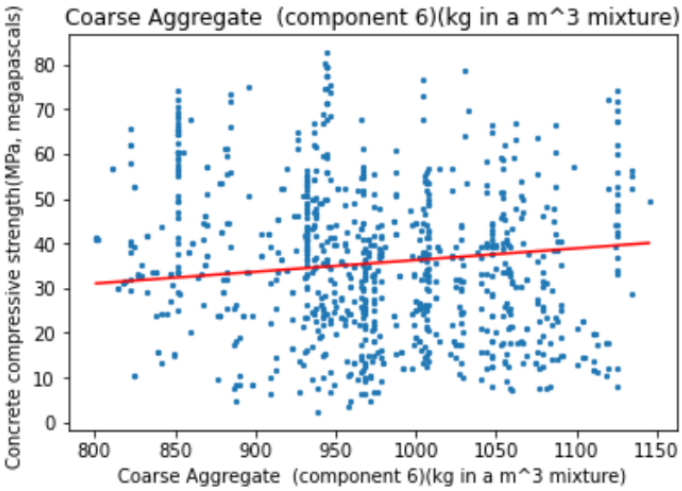

```
Water  (component 4)(kg in a m^3 mixture)
y = 0.13593234324163211x + 10.000791669123952
Train MAE 14.307131922768754
Explained varianvce for train dataset:  0.821756667110983
Test MAE 13.39267706001814
Explained varianvce for test dataset:  0.8264885956687995
```



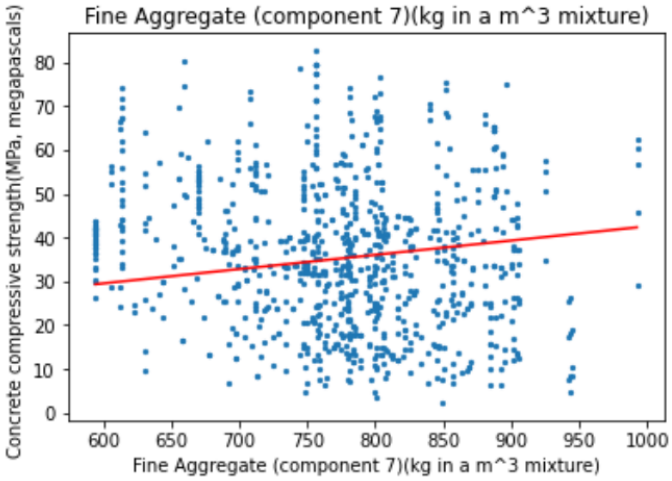
```
Superplasticizer (component 5)(kg in a m^3 mixture)
y = 1.1380870805735346x + 28.106401074588046
Train MAE 12.73844310975634
Explained varianvce for train dataset:  0.8412999497064326
Test MAE 11.591787700602076
Explained varianvce for test dataset:  0.8498203642462892
```



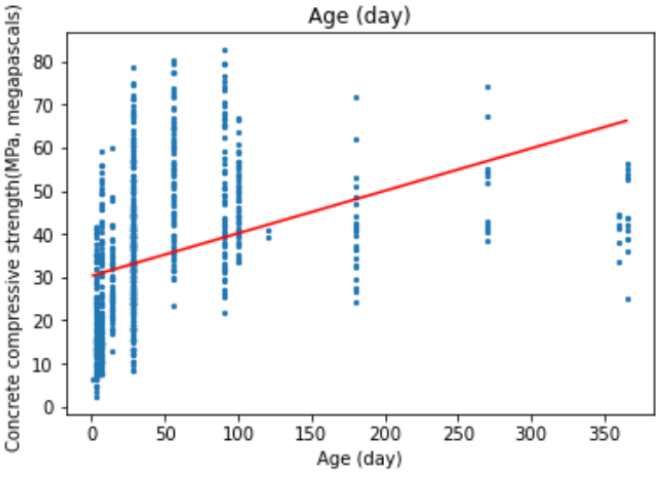

```
Coarse Aggregate  (component 6)(kg in a m^3 mixture)
y = 0.026250327980512314x + 10.000027074463134
Train MAE 13.86389538695772
Explained varianvce for train dataset:  0.8272786653582636
Test MAE 12.953696392936548
Explained varianvce for test dataset:  0.8321758941587305
```



```
Fine Aggregate (component 7)(kg in a m^3 mixture)
y = 0.03255001108246929x + 10.000041697168696
Train MAE 14.09557129102991
Explained varianvce for train dataset:  0.8243923646297308
Test MAE 13.373128583984046
Explained varianvce for test dataset:  0.8267418597110825
```

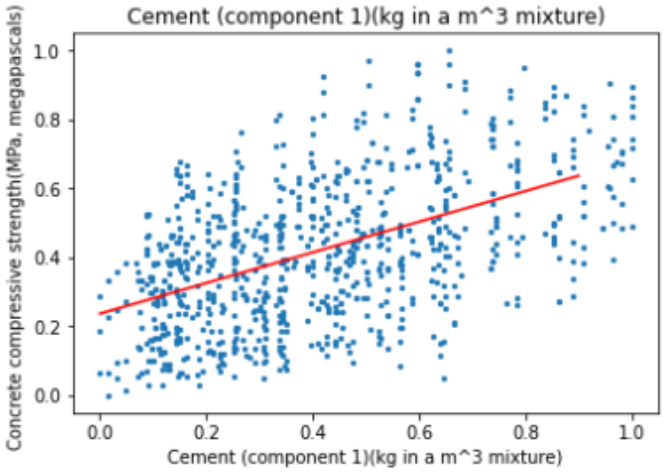



```
Age (day)
y = 0.09851423115985664x + 30.277670431089728
Train MAE 12.690966920322593
Explained varianvce for train dataset:  0.8418914249429247
Test MAE 11.524418248886747
Explained varianvce for test dataset:  0.8506931821395127
```

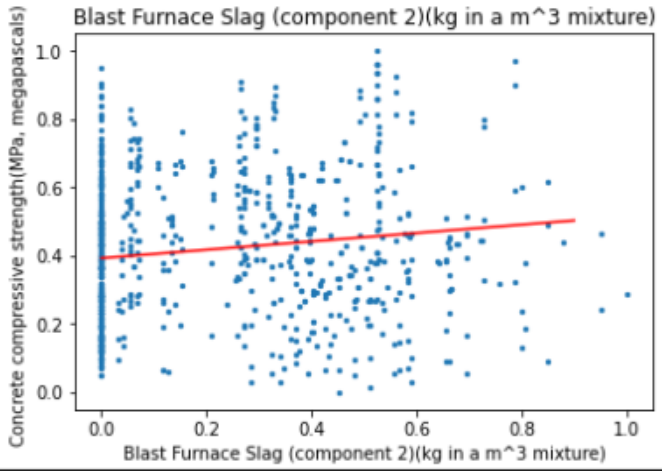


Univariate linear regression with normalized data and MSE:

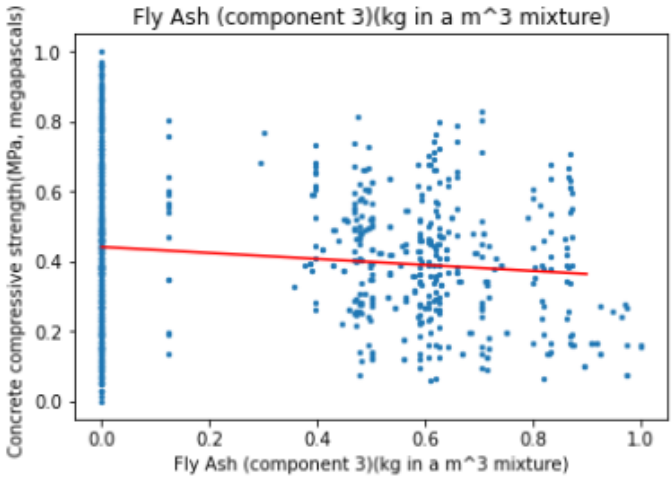
```
Cement (component 1)(kg in a m^3 mixture)
y = 0.443979635442528x + 0.23701532334804312
Train MSE 0.032184050727085765
Explained varianvce for train dataset:  0.9678159492729143
Test MSE 0.03513597098497504
Explained varianvce for test dataset:  0.963461402333841
```



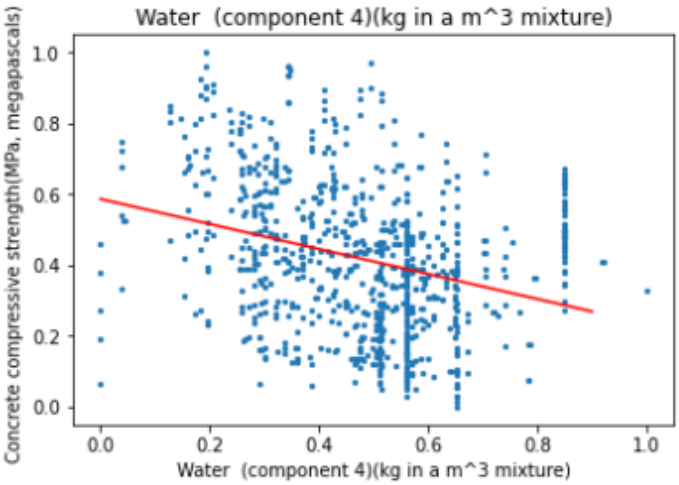

```
Blast Furnace Slag (component 2)(kg in a m^3 mixture)
y = 0.12249128474396392x + 0.3928030432123886
Train MSE 0.0428350982511076
Explained varianvce for train dataset:  0.9571649017488923
Test MSE 0.04010022806029113
Explained varianvce for test dataset:  0.9582989722970016
```



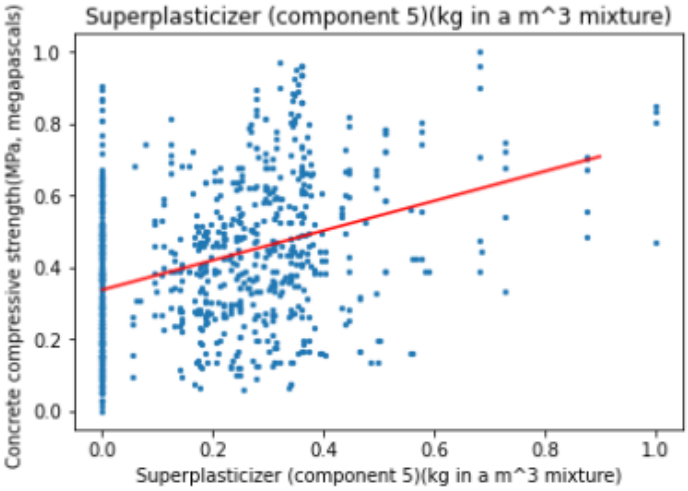
```
Fly Ash (component 3)(kg in a m^3 mixture)
y = -0.08561753286297082x + 0.4413719559558537
Train MSE 0.04293728898283913
Explained varianvce for train dataset:  0.9570627110171609
Test MSE 0.04200768406882004
Explained varianvce for test dataset:  0.9563153707141298
```



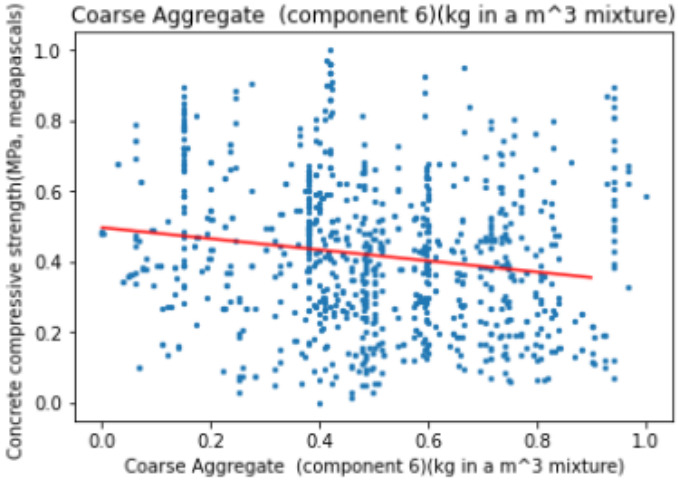

```
Water  (component 4)(kg in a m^3 mixture)
y = -0.3524390133947921x + 0.5862548527586093
Train MSE 0.03999280809343465
Explained varianvce for train dataset:  0.9600071919065654
Test MSE 0.03723735989856429
Explained varianvce for test dataset:  0.9612761260514068
```



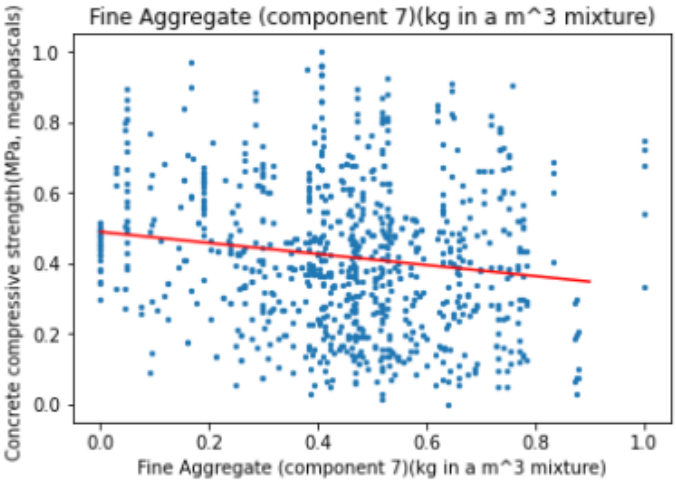
```
Superplasticizer (component 5)(kg in a m^3 mixture)
y = 0.4120564343748915x + 0.33765846040834263
Train MSE 0.03775285944986598
Explained varianvce for train dataset:  0.962247140550134
Test MSE 0.03554420942105398
Explained varianvce for test dataset:  0.9630368670342722
```



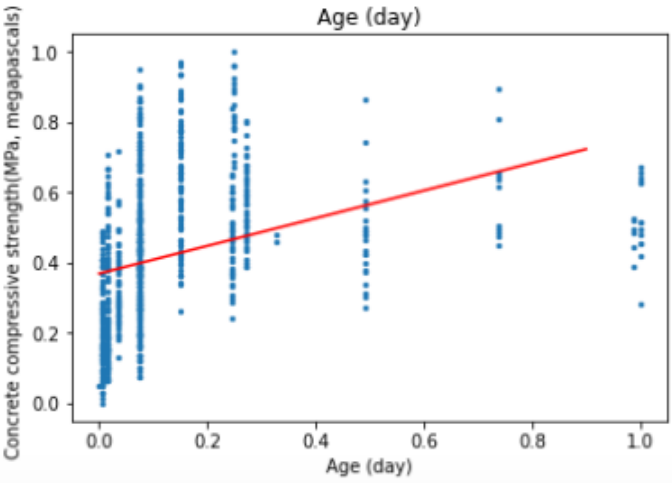

```
Coarse Aggregate  (component 6)(kg in a m^3 mixture)
y = -0.15748287726735644x + 0.49687103362516893
Train MSE 0.04245007769954551
Explained varianvce for train dataset:  0.9575499223004544
Test MSE 0.039672880762266084
Explained varianvce for test dataset:  0.958743379283589
```



```
Fine Aggregate (component 7)(kg in a m^3 mixture)
y = -0.1565097866773764x + 0.488979745935101
Train MSE 0.04267960177827226
Explained varianvce for train dataset:  0.9573203982217278
Test MSE 0.03789983923809205
Explained varianvce for test dataset:  0.9605872005607893
```




```
Age (day)
y = 0.3931462524270607x + 0.3692148441748599
Train MSE 0.03884840233964791
Explained varianvce for train dataset:  0.9611515976603521
Test MSE 0.03683018166896248
Explained varianvce for test dataset:  0.9616995588210947
```



Discussion

Describe how the different models compared in accuracy on the training data. Did the same models that accurately predicted the training data also accurately predict the testing data?

For the univariate linear regression model with original data when using MSE as loss function, the MSE for the train data is on 207~311, for the test data is on 226~300. MSE of test data has smaller range mean it is more accurate. And the MSE for train data and test data have a difference which means the model does not accurately predict the train data as predict the test data. In addition, according to VE, this model fit worst for the dataset compared with other univariate linear regression models.

For the univariate linear regression model with original data when using MAE as loss function, the MAE for the train data is on 11~21, for the test data is on 11~21. Therefore, according to the MAE, this model predicted the train data accurately as predict the test data. Moreover, according to VE, it has the closet VE to 0, which means this model fit best for the dataset compared with other univariate linear regression models.

For the univariate linear regression model with normalized data when using MSE as loss function, the MSE for the train data is on 0.032~0.043, for the test data is on 0.035~0.042. Although there is a little difference between MSE value between MSE value range of train and test data, it still can say for the normalized data the model performs well for both train and test data. In addition, the range of MSE is small, therefore this model also has the better accuracy than other models. Further, according to VE, the absolute VE value of it is bigger than original data with MAE model but less than original data with MSE model which means it performs better than first model but worse than the second model.

For the multivariate linear regression models, using MSE as loss function, the model with original has bigger difference on MSE range between test data and train data than the model with normalized data. Thus, the model with normalized data predicts more accurate. But, according to the VE, the former fit data better. For the model using MAE as loss function, based on VE, it fit data better than the model with original data using MSE as loss function E but worse than the model with normalized data using MSE as loss function. Its MAE for train and test data is very close, which shows it predicted the train data accurately as predict the test data.

Describe how the different models compared to train/test. Did different models take longer to train? Did you have to use different hyperparameter values for different models?

According to the answer to above question, univariate and multivariate linear regression model using MAE as loss function have least difference between train and test data. For using MSE as loss function models, the models with normalized data have less difference due to the small size of data.

Multivariate linear regression spent more time to train than univariate linear regression models. Using normalized data will decrease the time of training data. There is not obviously difference on spent time for using MSE or MAE.

According to my work, I do not use different hyperparameter values for different models, I tend to set a random initial value and then to get the hyperparameter values which fit data well during the training. And the result I got shows the different features models has different hyperparameter values, but different model types with same feature do not have very different hyperparameter values.

- Above answers have already include the comparisons from using MAE and normalized data.

Draw some conclusions about what factors predict concrete compressive strength. What would you recommend for making the hardest possible concrete?

According to the result (Plots of trained univariate linear regression models), increasing the cement, superplasticizer and days can improve the concrete compressive strength.