## I. Introduction:

1. The problem:

- The data used here is the Concrete Compressive Strength dataset in the UCI Machine Learning Repository. There are 8 features total, including: Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate (kg in a m^3 mixture) and Age (day). The target (output) is Concrete comprehensive strength (MPa, megapascals).
- First, I perform univariate linear regression for each feature of the 8 features mentioned above. Then, I perform multivariate linear regression using all 8 features at the same time to compare which model performs better using training and testing loss. The loss function for both algorithm is Mean Squared Error.

2. Algorithm:

a. Uni-variate regression:

- For a univariate linear model: $f(x) = mx + b$.
- The MSE loss function: $L(m,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$
- Since we need to consider the impact each parameter m and b has on the final prediction, I use the gradient which is the partial derivative of the loss function with respect to each parameter (for this assignment, I choose stepsize $\alpha = 0.000000001$:

$$\frac{\partial L}{\partial m} = \frac{1}{n}\sum 2(y_i - (mx_i + b))(-x_i) = \frac{-2}{n}\sum x_i(y_i - (mx_i+b))$$

$$\hookrightarrow m = m - \alpha\frac{\partial L}{\partial m} = m - \alpha\left(\frac{-2}{n}\sum x_i(y_i - (mx_i + b))\right)$$

$$\frac{\partial L}{\partial b} = \frac{-2}{n}\sum(y_i - (mx_i + b))$$

$$\hookrightarrow b = b - \alpha\frac{\partial L}{\partial b} = b - \alpha\left(\frac{-2}{n}\sum(y_i - (mx_i+b))\right)$$

- I stop updating the objective function when the previous and current MSE differs by less than 0.0001.

b. Multi-variate regression:

- For a multivariate linear model: $f(x) = (\vec{a} \cdot \vec{x})$
- The MSE loss function: $L(m,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}|\vec{a} \cdot \vec{x} - y_i|^2$ , where $\vec{a} = (a_0, a_1, \ldots, a_n)^T$ $and$ $\vec{x} = (1, x_0, x_1, \ldots, a_n)^T$, with $a_0$ being the intercept *b* in model.
- Same as univariate linear regression, I use partial derivative with respect to *a* to update the parameter, same as above, stepsize $\alpha = 0.000000001$ and stop updating when the loss function differs by less than 0.0001:

$$\frac{\partial L}{\partial a} = \frac{1}{n}2((\vec{a} \cdot \vec{x}) - y_i)(\vec{x}) = \frac{2}{n}(\vec{x})(\vec{a} \cdot \vec{x} - \vec{y})$$

$$a = a - \alpha\left(\frac{2}{n}(\vec{x})(\vec{a} \cdot \vec{x} - \vec{y})\right)$$

3. Pseudocode:

a. Uni-variate regression:

**Input:** training and testing dataset including input variables $x$ and target $y$.

**Output:** Training and testing loss, number of iterations, final parameter $m$ $and$ $b$.

```
$mse = +\infty$
$m = b = 0$
iteration = 0
$while$ $\Delta mse > 0.0001$ $do$:
    $m = m - \alpha\left(-\frac{2}{n}\sum x_i(y_i - (mx_i + b))\right)$
    $b = b - \alpha\left(-\frac{2}{n}\sum (y_i - (mx_i + b))\right)$
    $update$ $mse$
    $iteration$ += 1
end
return loss(x_train, y_train, m, b, n_train), loss(x_test, y_test, m, b, n_test), mse, iteration, m, b
```

b. Multi-variate regression:

**Input:** training and testing dataset including input variables $x$ and target $y$.

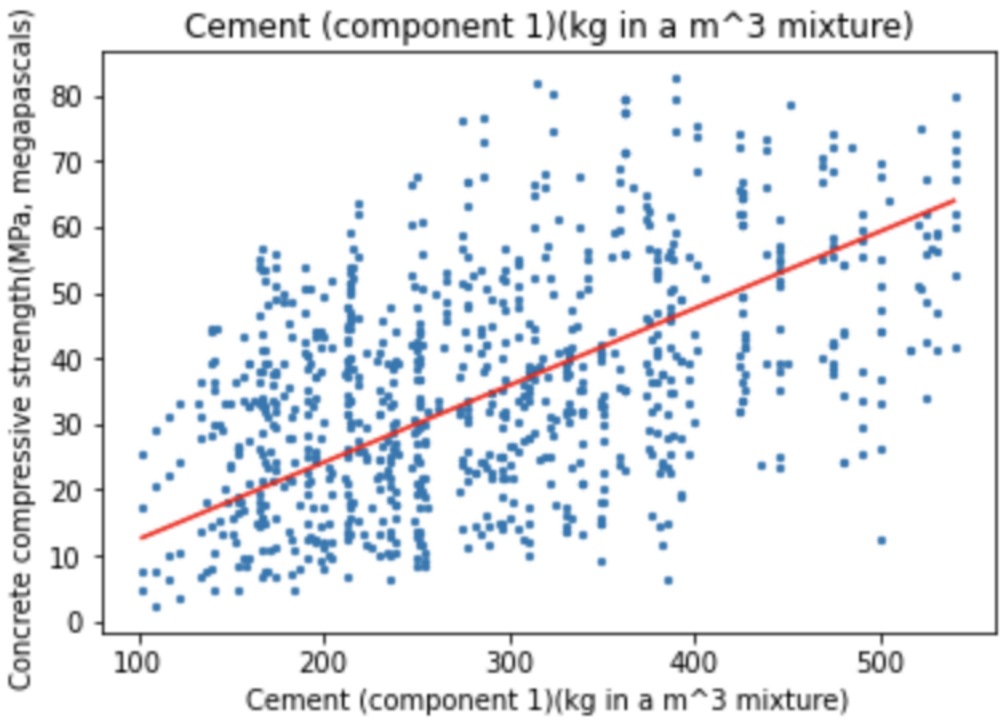
**Output:** Training and testing loss, number of iterations, final parameter $a$.

```
# add a column including ones to the training and testing set
x_train = [1, x_train]
x_test = [1, x_test]
Initialize a = [0, 0, …,0]
$mse = +\infty$
while $\Delta mse > 0.0001$ do:
    $a = a - \alpha\left(\frac{2}{n}(\vec{x})(\vec{a}\cdot\vec{x} - \vec{y})\right)$
    update mse
    $iteration \mathrel{+}= 1$
End
return loss(x_train, y_train, a, n_train), loss(x_test, y_test, a, n_test), mse, iteration, a
```

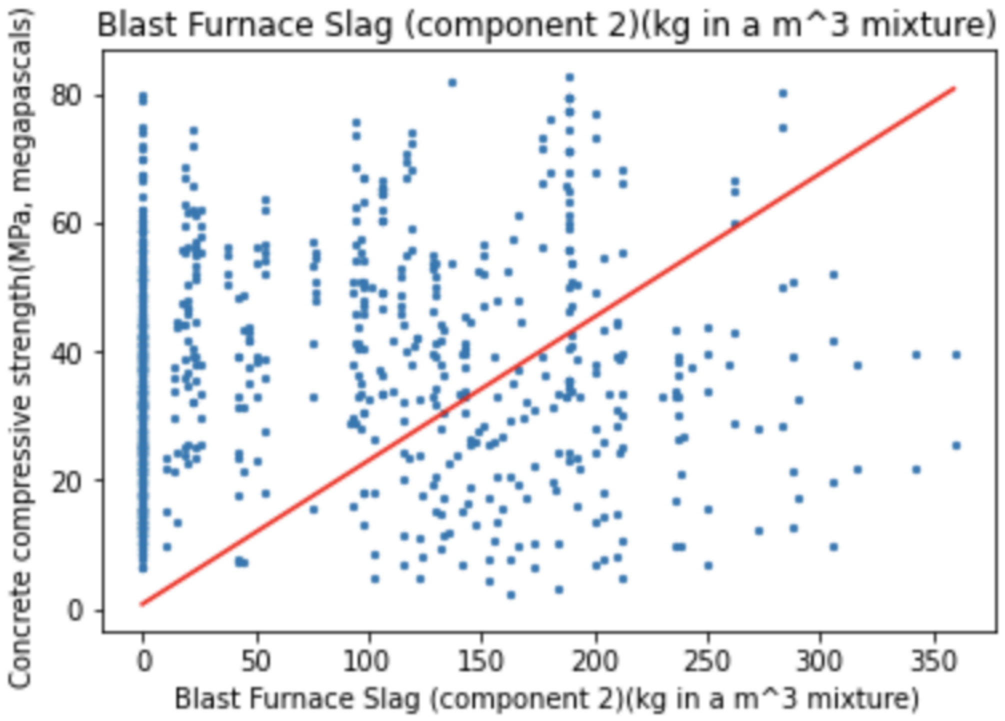
## II. Results:

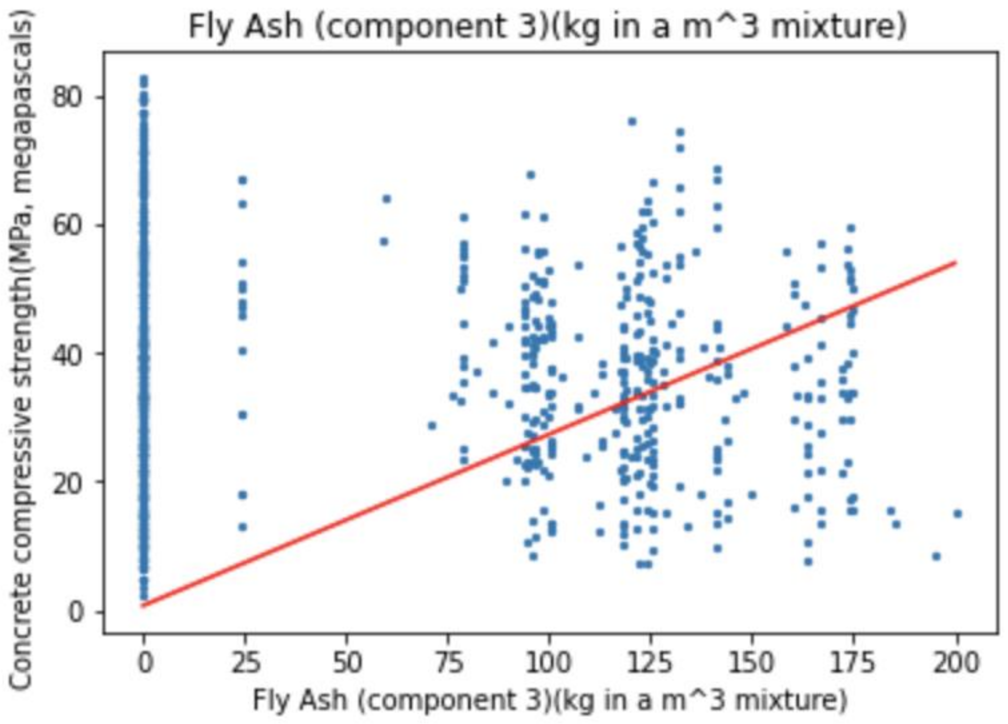
| Features | MSE on training dataset | MSE on testing dataset |
|---|---|---|
| Cement | 249.3974 | 112.271 |
| Blast Furnace Slag | 1020.3884 | 348.9608 |
| Fly Ash | 1135.236 | 749.2194 |
| Water | 358.2695 | 184.1205 |
| Superplasticizer | 720.3892 | 293.9542 |
| Coarse Aggregate | 322.3485 | 165.7727 |
| Fine Aggregate | 333.4409 | 171.1391 |
| Age | 949.1309 | 690.002 |
| All features | 119.8119 | 63.0345 |

**Table 1**: MSE on training/testing dataset using different features

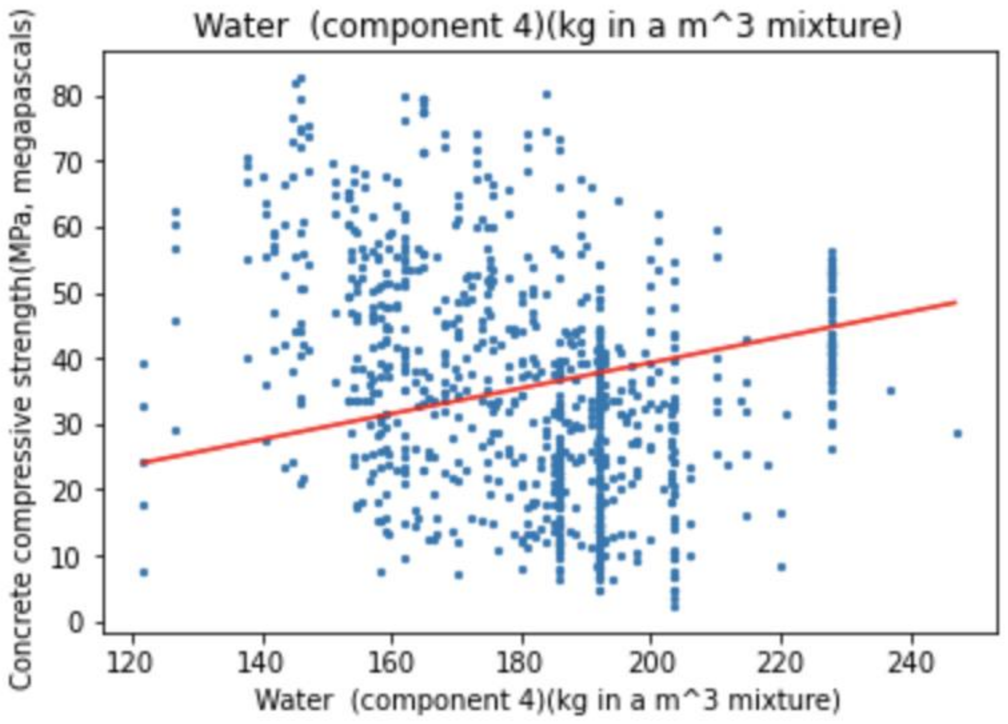

Cement (component 1)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
Cement (component 1)(kg in a m^3 mixture)
0
10
20
30
40
50
60
70
80
100
200
300
400
500

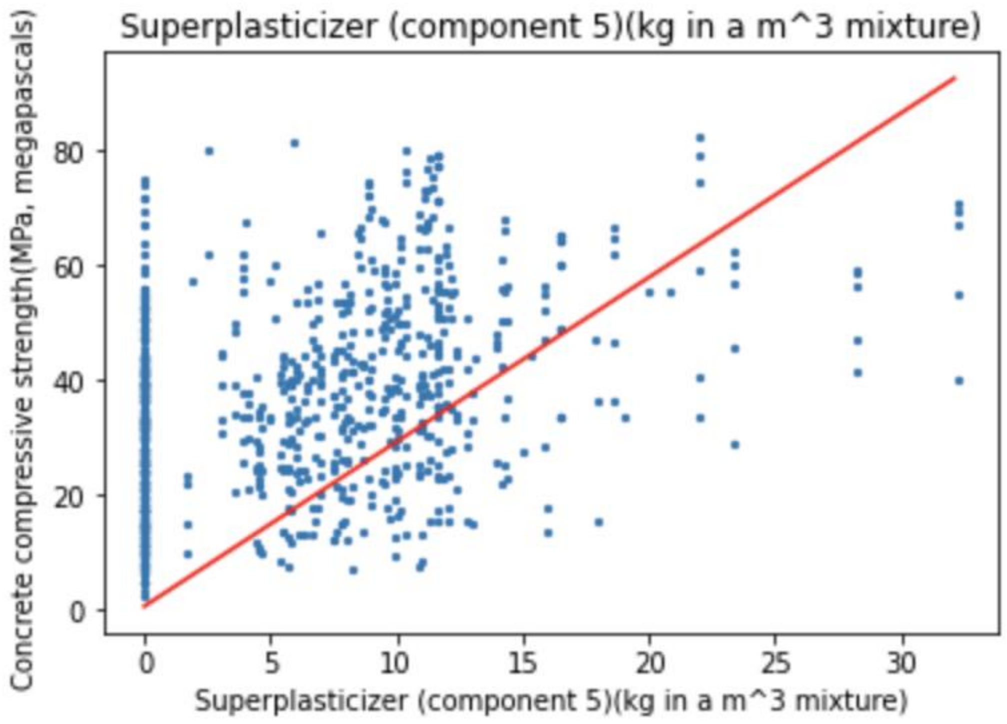
---

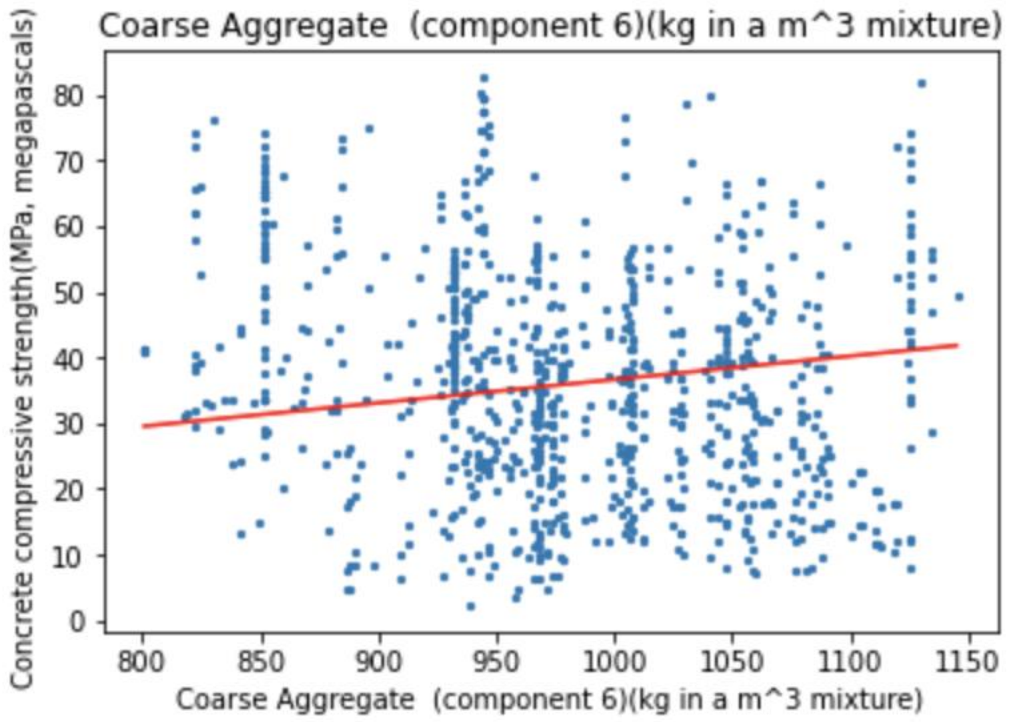

Blast Furnace Slag (component 2)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
Blast Furnace Slag (component 2)(kg in a m^3 mixture)
0
20
40
60
80
0
50
100
150
200
250
300
350

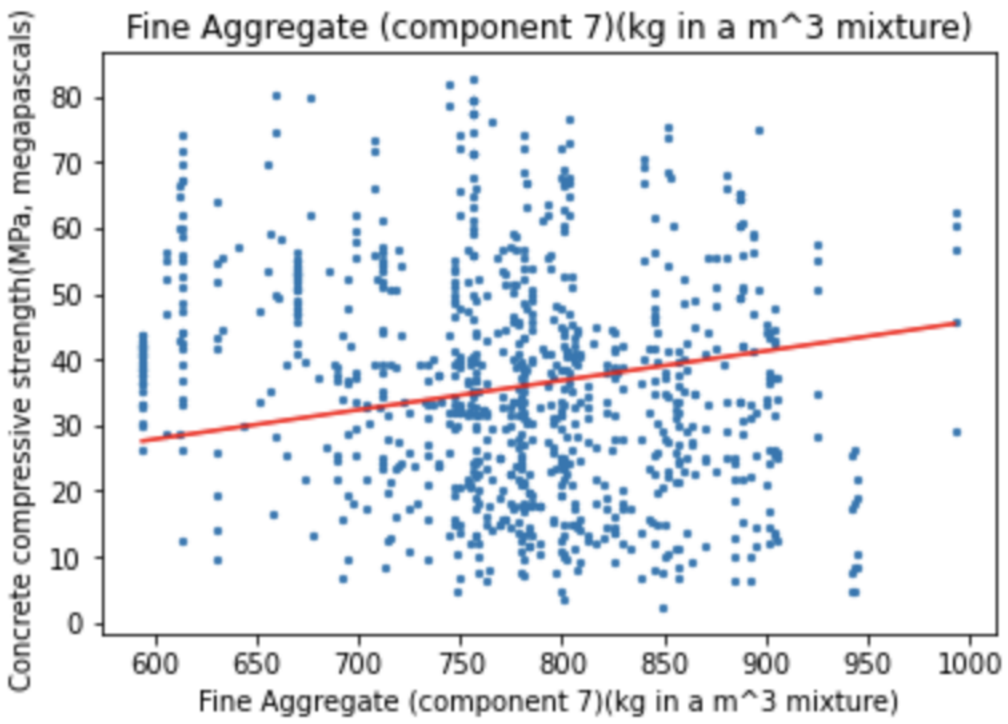

Fly Ash (component 3)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
0
20
40
60
80
0
25
50
75
100
125
150
175
200
Fly Ash (component 3)(kg in a m^3 mixture)

---

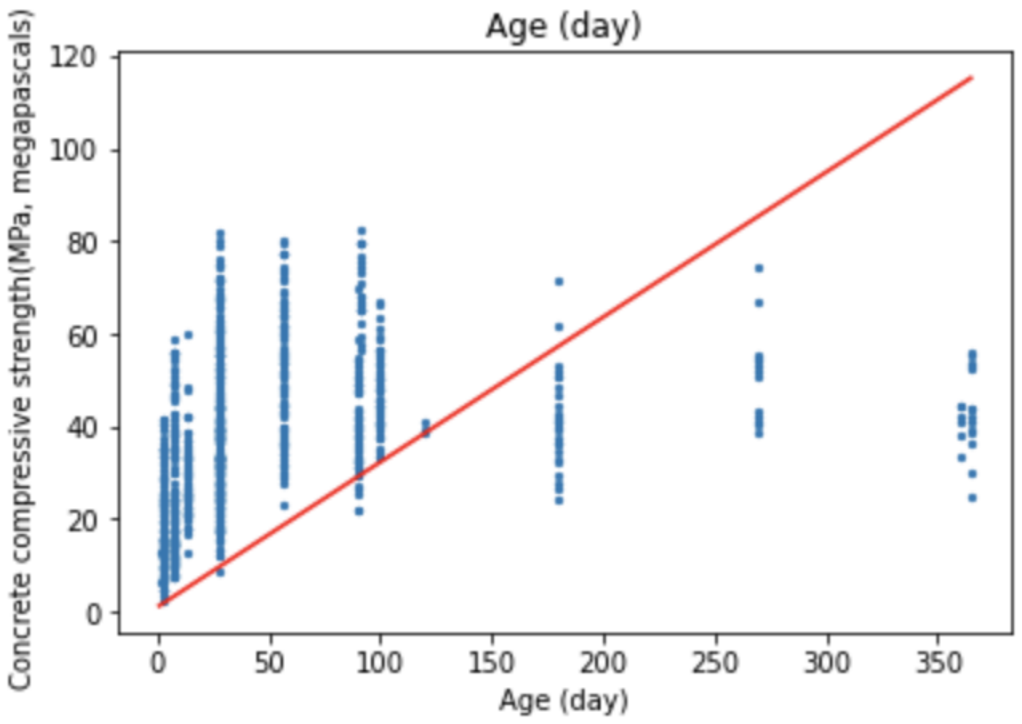

Water (component 4)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
0
10
20
30
40
50
60
70
80
120
140
160
180
200
220
240
Water (component 4)(kg in a m^3 mixture)


Superplasticizer (component 5)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
Superplasticizer (component 5)(kg in a m^3 mixture)
0
20
40
60
80
0
5
10
15
20
25
30


Coarse Aggregate (component 6)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
Coarse Aggregate (component 6)(kg in a m^3 mixture)
0
10
20
30
40
50
60
70
80
800
850
900
950
1000
1050
1100
1150


Fine Aggregate (component 7)(kg in a m^3 mixture)
Concrete compressive strength(MPa, megapascals)
0
10
20
30
40
50
60
70
80
600
650
700
750
800
850
900
950
1000
Fine Aggregate (component 7)(kg in a m^3 mixture)

------------------------------------------------


Age (day)
Concrete compressive strength(MPa, megapascals)
0
20
40
60
80
100
120
0
50
100
150
200
250
300
350
Age (day)

## III. Discussion:

- From table 1, we can conclude that multivariate linear regression model performs best compared to univariate linear regression. This makes sense because instead of one single feature, I used a vector including multiple features to train the model, which leads to lower errors on training and testing data.
- Among the 8 individual features, the order of MSE on **training** dataset from smallest to largest: is Cement (249.3974), Coarse Aggregate (322.3485), Fine Aggregate (333.4409), Water (358.269), Superplasticizer (720.3892), Age (949.1309), Blast Furnace Slag (1020.3884), Fly Ash (1135.236)
- Among the 8 individual features, the order of MSE on testing dataset from smallest to largest: is Cement (112.271), Coarse Aggregate (165.7727), Fine Aggregate (171.1391), Water (184.1205), Superplasticizer (293.9542), Blast Furnace Slag (348.9608), Age (690.002), Fly Ash (749.2194).
- Generally, if the same model that performs well on the training data will do well on the testing data. We can see that the overall order of MSE is maintained except for *Blast Furnace Slag* and *Age* where *Age* performs better on training data, while *Blast Furnace Slag* performs much better on testing data than training data. Other than that, all features provide the same ordering.

| Features | m | Multivariate regression model |
|---|---|---|
| Cement | 0.119 | 0.151 |
| Blast Furnace Slag | 0.219 | 0.0955 |
| Fly Ash | 0.271 | 0.1033 |
| Water | 0.194 | -0.1269 |
| Superplasticizer | 2.868 | 0.0346 |
| Coarse Aggregate | 0.037 | 0.00002 |
| Fine Aggregate | 0.046 | 0.0116 |
| Age | 0.306 | 0.1054 |

**Table 2**: Model Coefficients

- I believe the performance of the univariate models failed to predict which features were more important in the multivariate model because the order from smallest to largest of the features' coefficients do not match with the order from smallest to largest of m in univariate linear regression. If the performance of the univariate model were to succeed, the magnitude of m and coefficients should have similar values. Here, for example, *Superplasticizer* (2.868 – 1st) should be a good factor predicting concrete strength alone, but when used in the multivariate model, its coefficient is relatively small (0.0346 – 5th) Moreover, I noticed that I should use different step size and different threshold for different feature like what we saw in class but here, I kept stepsize and threshold constant so that could affect the accuracy of the model.
- Looking at the plots and MSE on training and testing dataset, *Cement* must be the most important factor in predicting concrete comprehensive strength. Other than that, I would choose water and the aggregates as well based on the MSEs (*Coarse Aggregate* and *Fine Aggregate*) and relatively high coefficient (*water* more specifically).

**IV. Extra Credit:**

**1.** Multivariate quadratic regression model:

- For a multivariate quadratic model: $f(x) = x^2 + bx + c$
- Create additional columns for $x^2$ in the training and testing set.
- Since squaring the input would make the numbers much bigger, I changed the threshold to 0.000001 and the stepsize for this model is 0.00000000000001 to try to minimize the objective function.
- MSE on training dataset: 185.737
- MSE on testing dataset: 112.807