CSE 514 Data Mining

Assignment1 --- Report

NAME: Yuxiao Wang   ID: 509794

Introduction (15 pts + 5 bonus points)

1. (4 pts) Description of the problem

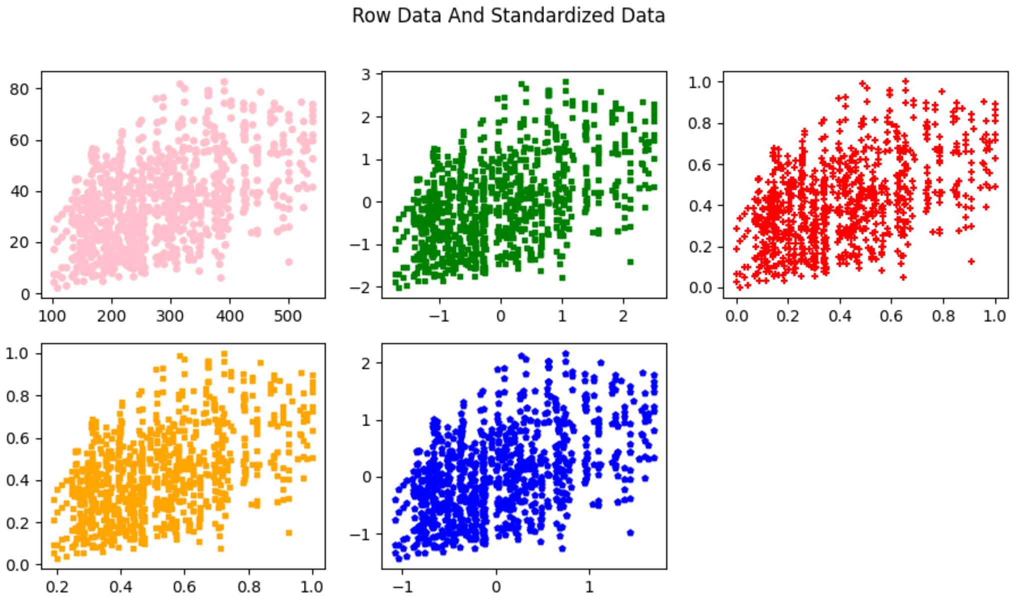
The dataset is the Concrete Compressive Strength dataset from UCI repository. There are 1030 instances in this dataset and there are 9 attributes include 8 quantitative input variables (Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate and Age) and 1 output variable (Concrete compressive strength).

The model could be used to test the compressive strength of a new type of concrete after learning some rules from this dataset, so it has a quite realistic meaning.

I am going to create three different regression methods which are uni-variate linear regression, multi-variate linear regression and ridge regression. At first of the process, I standardized the raw data. Meanwhile, I applied different loss functions, i.e., MSE loss and MAE loss, when training the data on those two data sets (original data and standardized data).

2. (3 pts) Description of how you normalized or standardized your data

I created a class called dataPreProcessing. I defined three built-in functions. Among them, in the function preprocessing, I used the methods in sklearn package and created four kinds of standardized data respectively by four methods: zscore, minmax, maxabs, and robust. And I selected the zscore standardized data as the processed data compared to the raw data.



The first sub-picture (the pink one) above is from the raw data. And the second sub-picture (the green one) is from the zscore standardized data. The third sub-picture (the red one) is from the minmax standardized data. The fourth sub-picture (the yellow one) is from the maxabs standardized data. And the last sub-picture (the blue one) is from the robust standardized data.

3. (5 pts) Details of your algorithm

I first set a small stop value for the regression process, when the new gradient value is smaller than the stop value, the process will stop. And I set the max iteration number as 100000, so the thread will stop automatically at the 100000th iteration even if the loss has not decreased to the stop value yet.

I applied the gradient descent method when training. For the learning rate value, I tried several values and find some small values make the training process better. Finally, I chose the value in range from 0.00001 to 0.0000001.

4. (3 pts) Pseudo-code of your algorithm

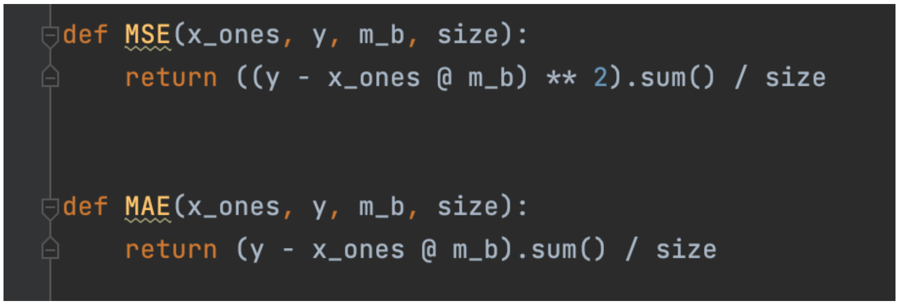
```
regression(x, y, learning_rate, stop_value, epochs):
        train_x = x
        train_y = y
```

```
inistalize parameters: m_b (m1, m2, …, mn, b)
initialize gradient value

while (gradient value > stop value) and (steps < epochs):
    calculate new gradient
    update m and b
    calculate mse loss
return m and b, loss
```

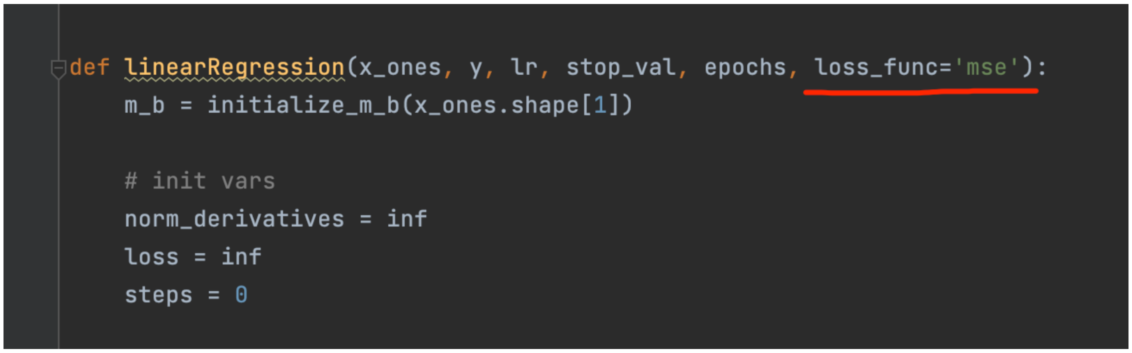
5. (+2 bonus pts) Description of how you implemented MAE

I define a function called MAE to calculate the mae loss value. By changing the ‘loss_func’ local parameter from ‘mse’ to ‘mae’, I can use mae loss as the gradient descent method loss.



```python
def MSE(x_ones, y, m_b, size):
    return ((y - x_ones @ m_b) ** 2).sum() / size


def MAE(x_ones, y, m_b, size):
    return (y - x_ones @ m_b).sum() / size
```



```python
def linearRegression(x_ones, y, lr, stop_val, epochs, loss_func='mse'):
    m_b = initialize_m_b(x_ones.shape[1])

    # init vars
    norm_derivatives = inf
    loss = inf
    steps = 0
```

6. (+3 bonus pts) Description of how you implemented Ridge Regression

I created a class called ridgeRegression and defined two built-in functions to do the regression and run the training process. I firstly set the lambda value as 0.2.

Results (52 pts + 8 bonus points)

1. (26 pts) Variance Explained (R-squared)

I calculated the r-squared values on the training data when using only one of the predictor variables (univariate regression) and when using all eight (multivariate regression). And there are a total of nine values from optimizing on the raw data, and nine values from optimizing on the pre-processed data.

R Squared Value On Training Data

Methods	Original Data	Standardized Data
ULR --- Feature 0	0.154148	0.224352
ULR --- Feature 1	-0.280885	0.017909
ULR --- Feature 2	-0.044890	0.001857
ULR --- Feature 3	-0.205294	0.086353
ULR --- Feature 4	0.176692	0.176692
ULR --- Feature 5	-0.090668	0.039555
ULR --- Feature 6	-0.126077	0.031302
ULR --- Feature 7	0.070721	0.112738
MLR	0.610086	0.612707

(10 pts) R Squared Value On Testing Data

Methods	Original Data	Standardized Data
ULR --- Feature 0	0.256139	0.441158
ULR --- Feature 1	0.071218	-0.108408
ULR --- Feature 2	-0.206036	-0.043400
ULR --- Feature 3	-0.293218	-0.077926
ULR --- Feature 4	-0.633216	-0.633214
ULR --- Feature 5	-0.155140	0.039555
ULR --- Feature 6	-0.191328	-0.172324
ULR --- Feature 7	-0.004373	-0.049501
MLR	0.588454	0.573326

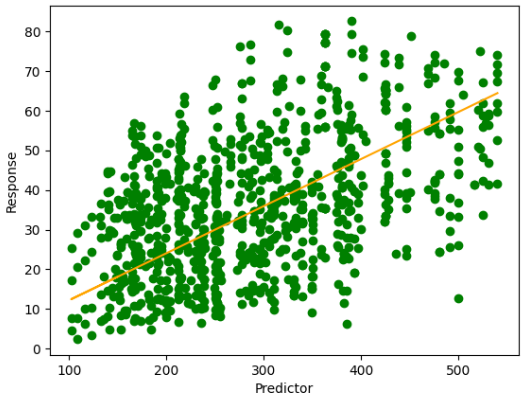
2. (16 pts) Plots

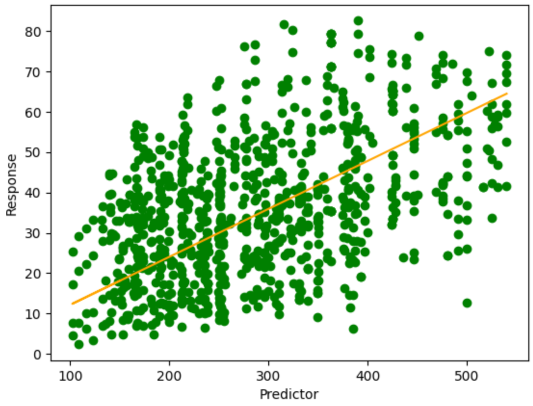
On The Raw Data

(+4 bonus points) (Left-MSE, Right-MAE)

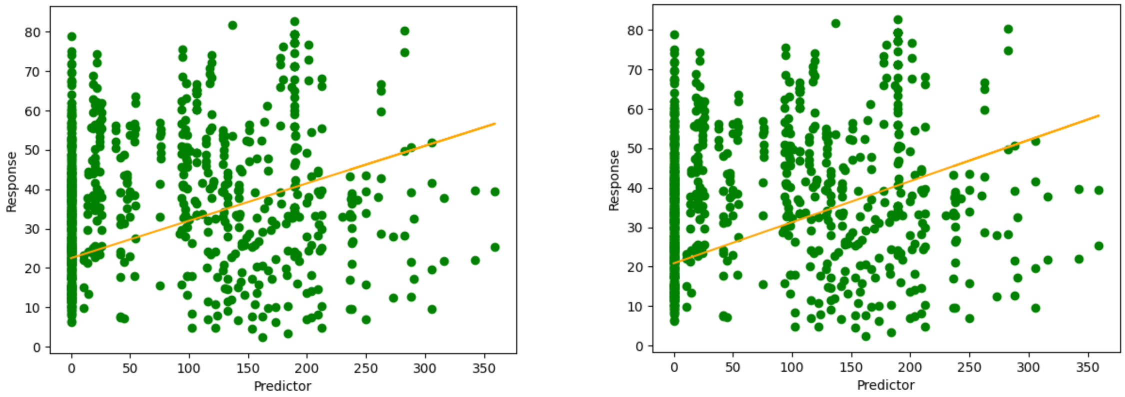
<u>Uni-variate Linear Regression</u>

Column 0: Cement

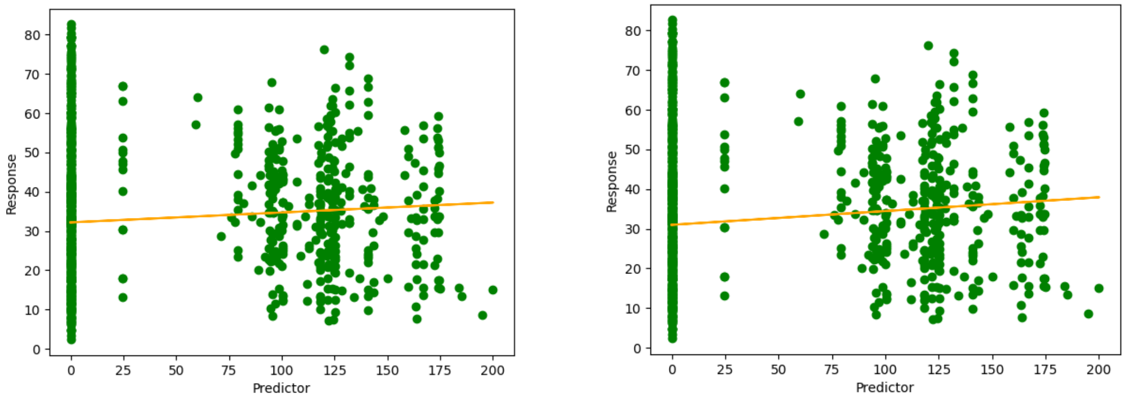




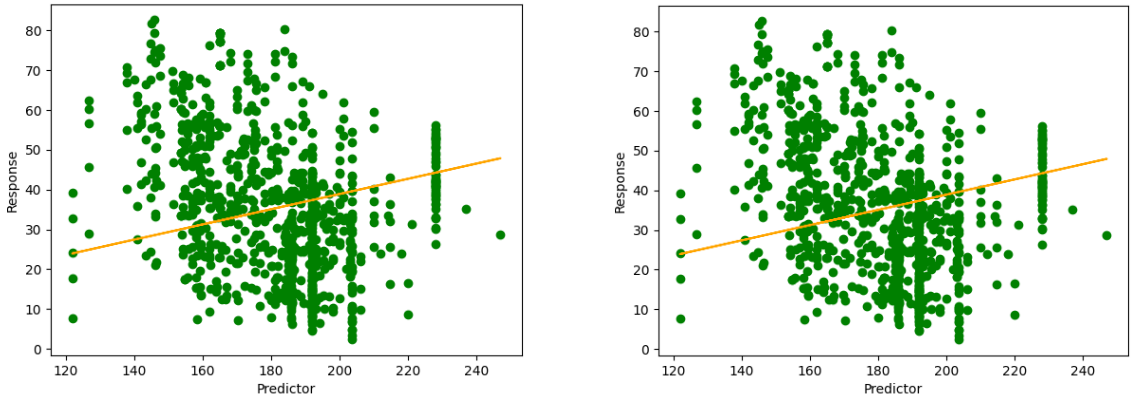
Column 1: Blast Furnace Slag



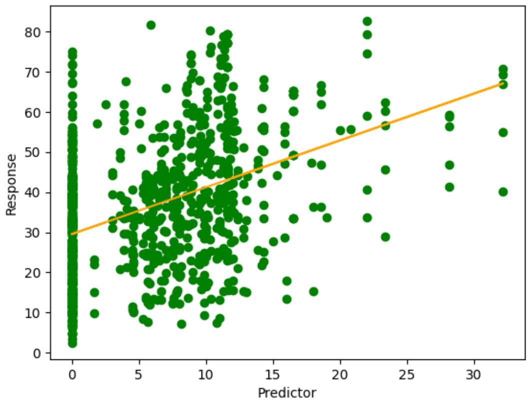
Column 2: Fly Ash

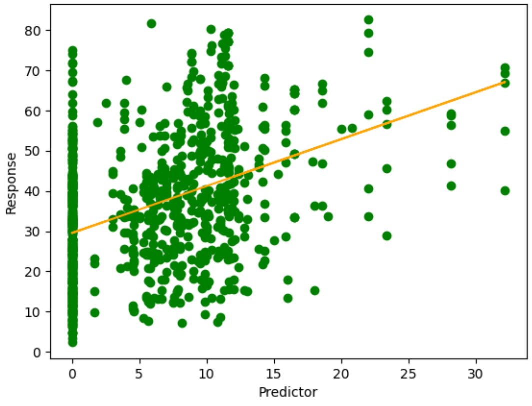


Column 3: Water

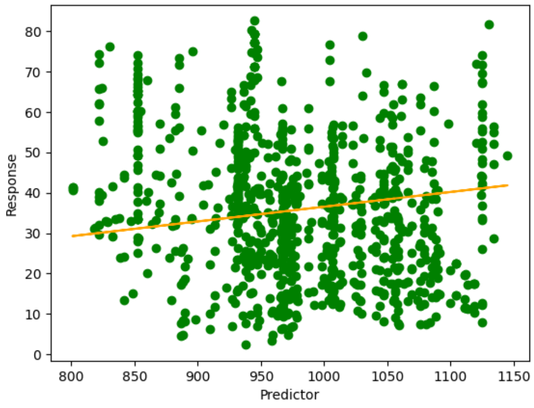


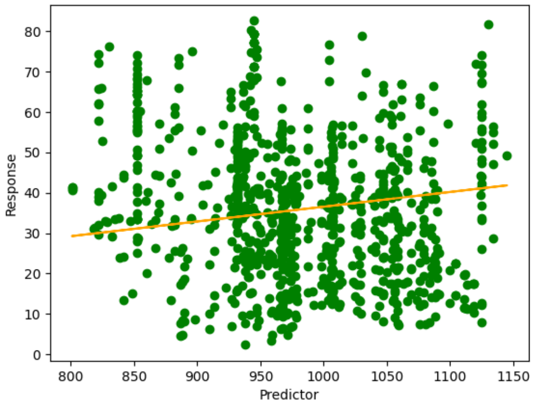
Column 4: Superplasticizer



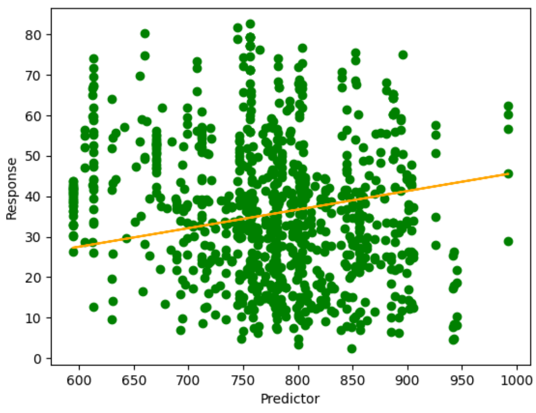


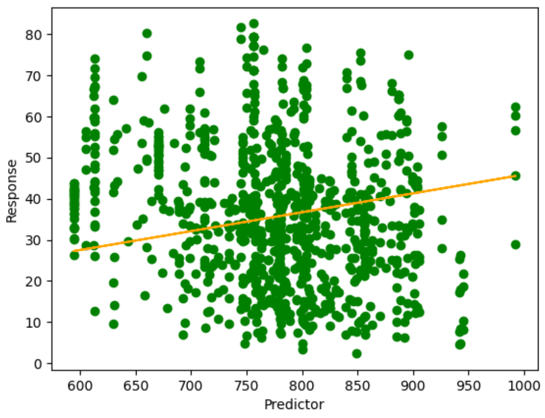
Column 5: Coarse Aggregate



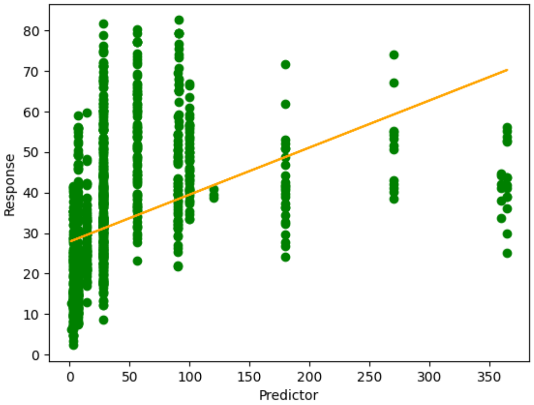


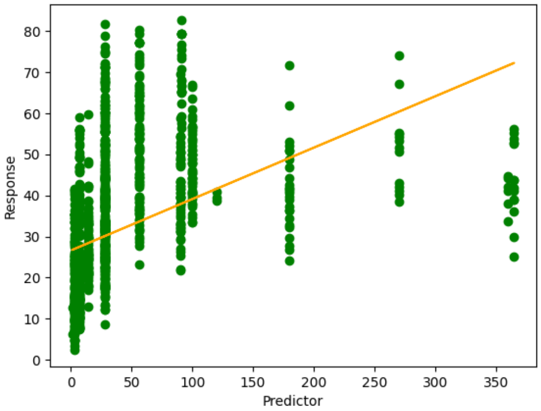
Column 6: Fine Aggregate



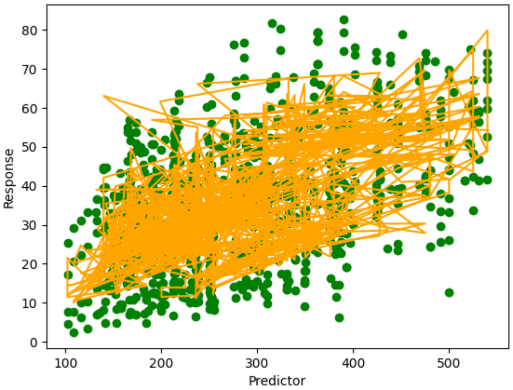


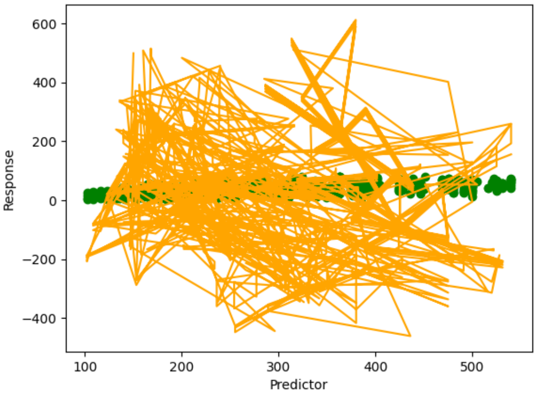
Column 7: Age



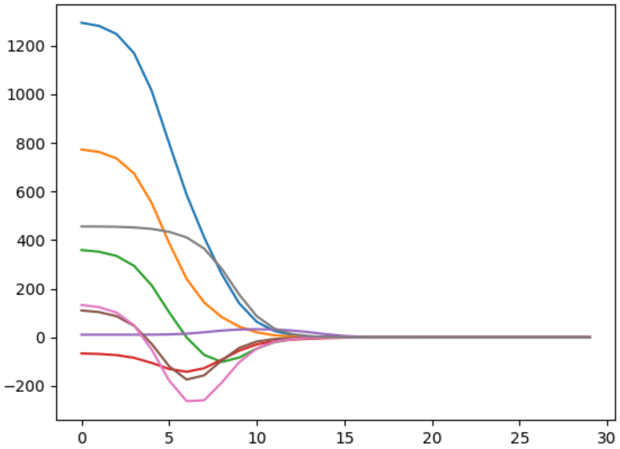


Multi-variate Linear Regression





(+4 bonus points) Ridge Regression

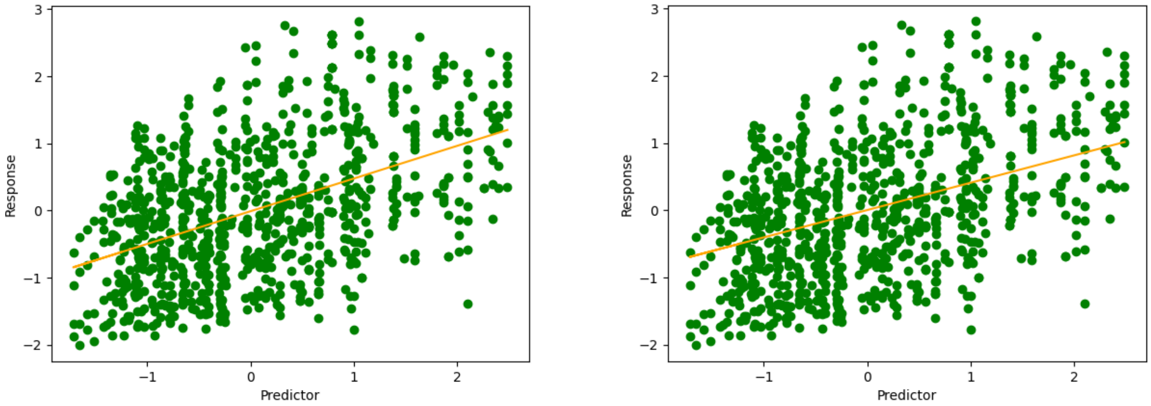


On the Standardized Data

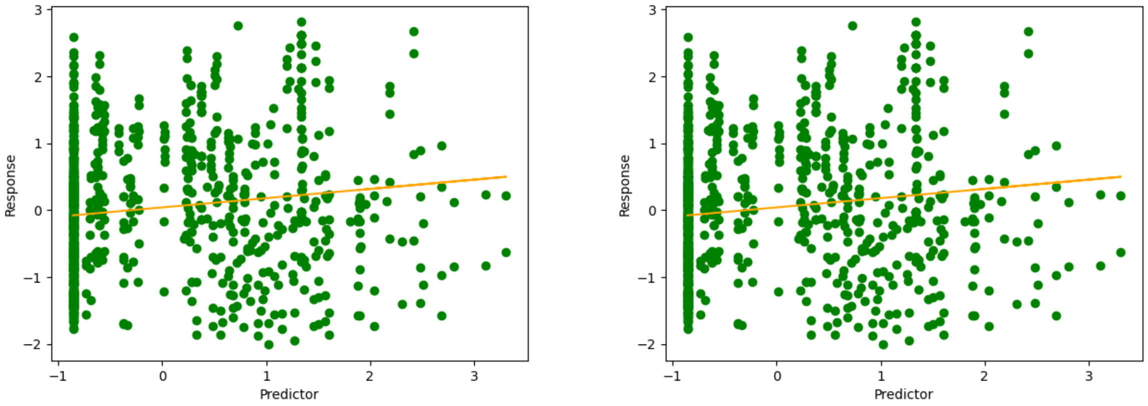
(Left-MSE, Right-MAE)

Uni-variate Linear Regression

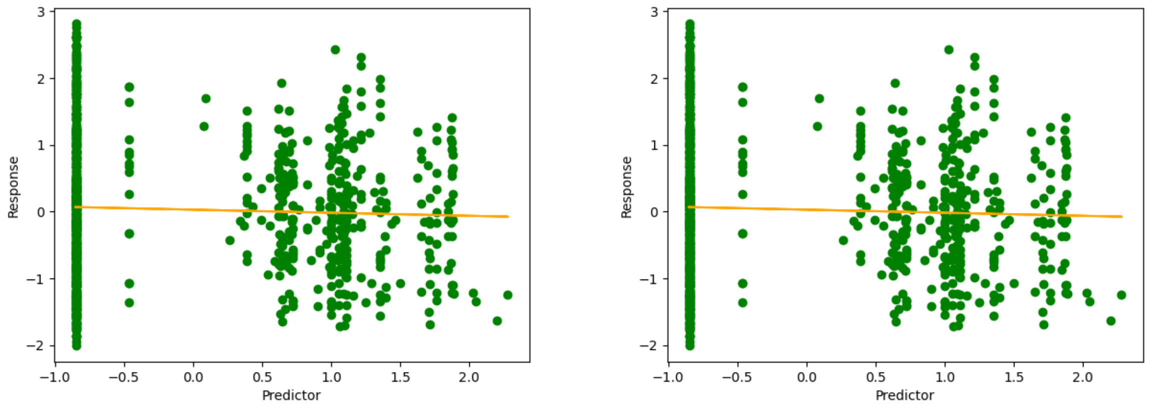
Column 0: Cement



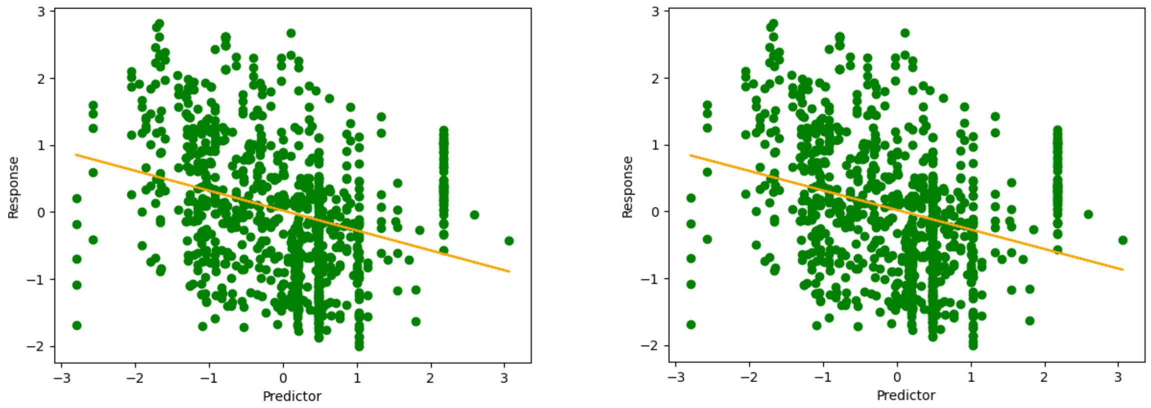
Column 1: Blast Furnace Slag



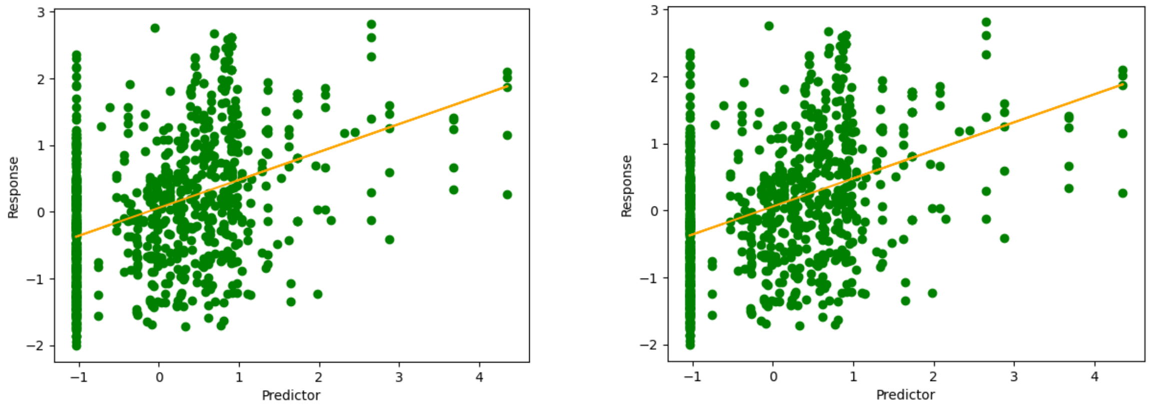
Column 2: Fly Ash



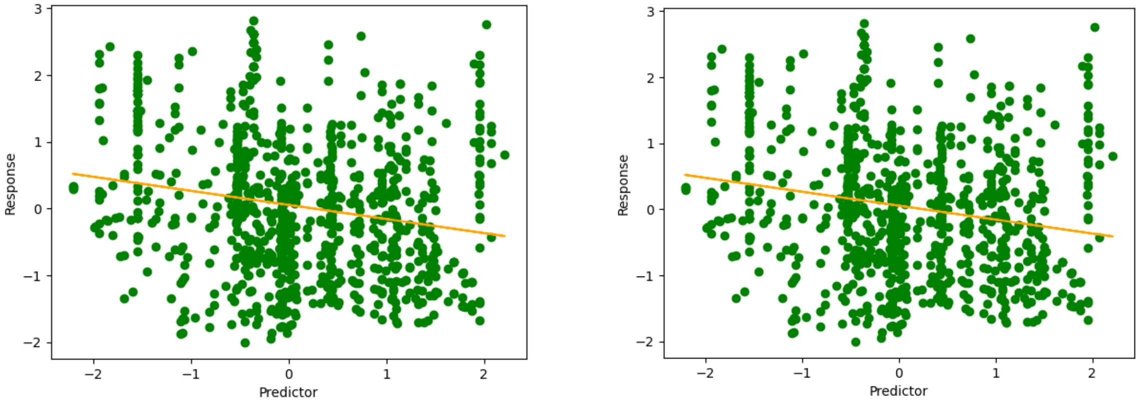
Column 3: Water



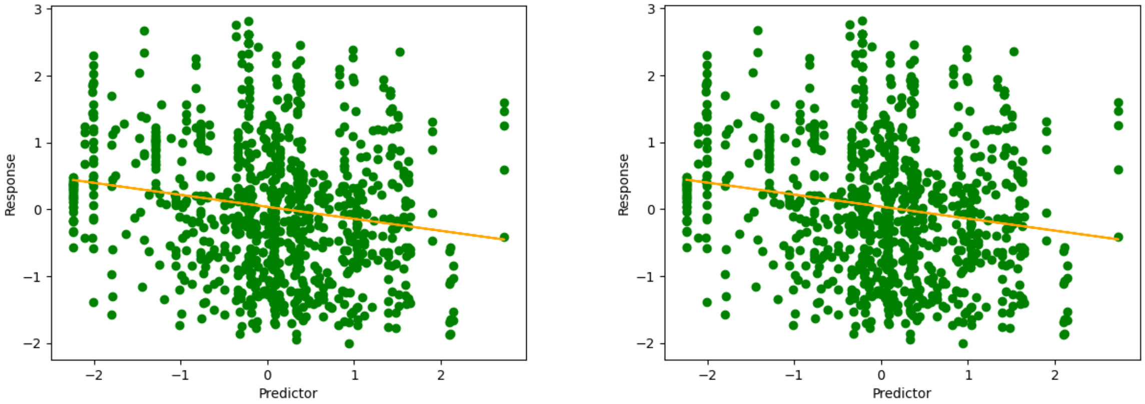
Column 4: Superplasticizer



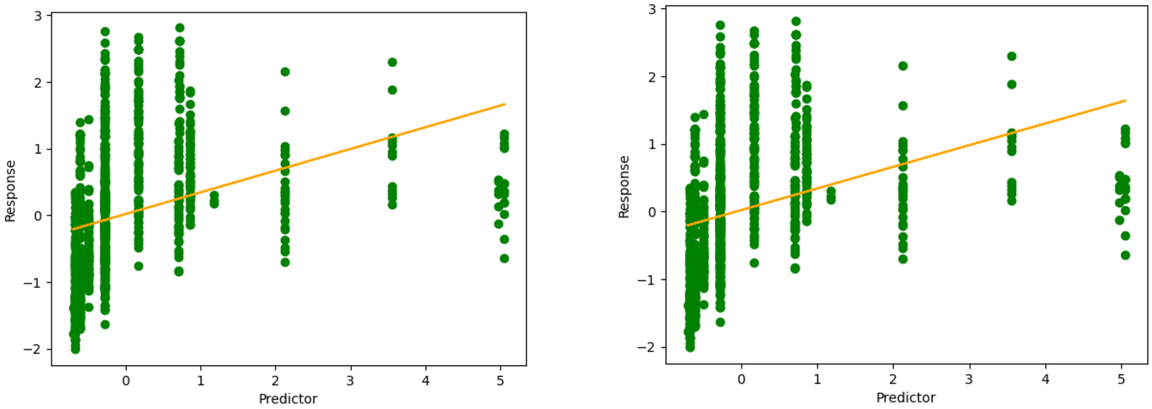
Column 5: Coarse Aggregate



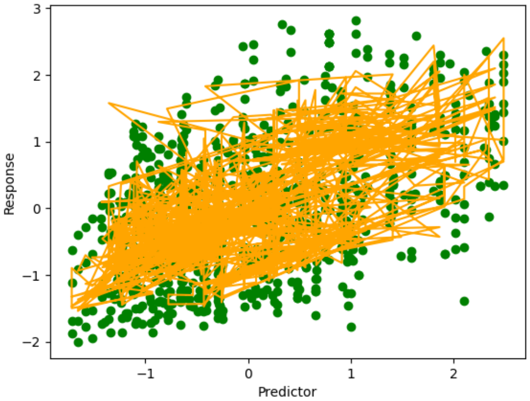
Column 6: Fine Aggregate

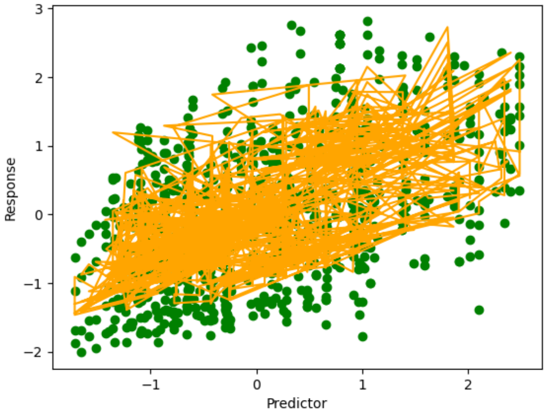


Column 7: Age

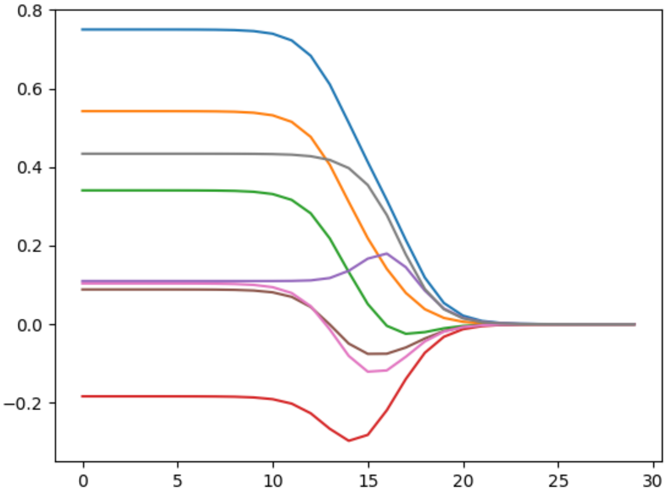


Multi-variate Linear Regression





Ridge Regression



Discussion (13 pts + 4 bonus points)

1. The uni-variate linear regression performed badly on the training data.
2. Although the one model predicted accurately on the training data, it might not predict accurately on the testing data.
3. If we want the concrete compressive strength be higher, we would like to put more of those features with higher and positive coefficients into the concrete.

4. We should avoid using too many featuresor combinations having significant negative coefficients.