CSE 514A Programming Assignment

Kaitlin Day
ID #473393

I. Introduction

A. *Problem Description*

The data set provides characteristics of concrete used in civil engineering applications and a score for comprehensive concrete strength based on these characteristics. Understanding the impact and influence of individual, groups of, or all these characteristics on the concrete strength score can provide useful insights for optimizing concrete quality and type for future construction projects. Most buildings and large-scale infrastructure require concrete foundations to ensure structural integrity. Better awareness of how various characteristics contribute to concrete strength allows project managers to minimize costs and materials required for future projects and encourages continued innovation in building construction.

B. *Pre-Processing*

1) Check for Nulls

The data set is first preprocessed by checking for null values and none are found.

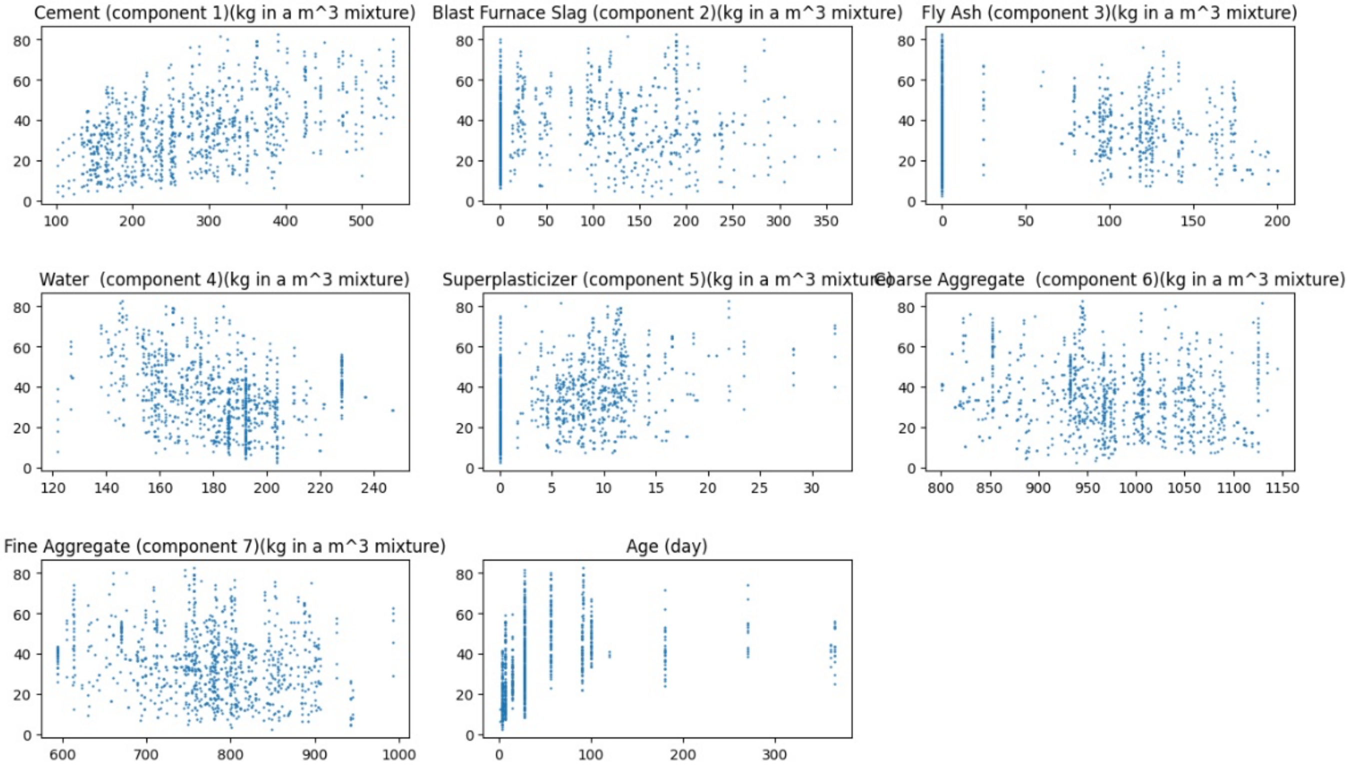
2) Standardize (Raw Data -> New Data)

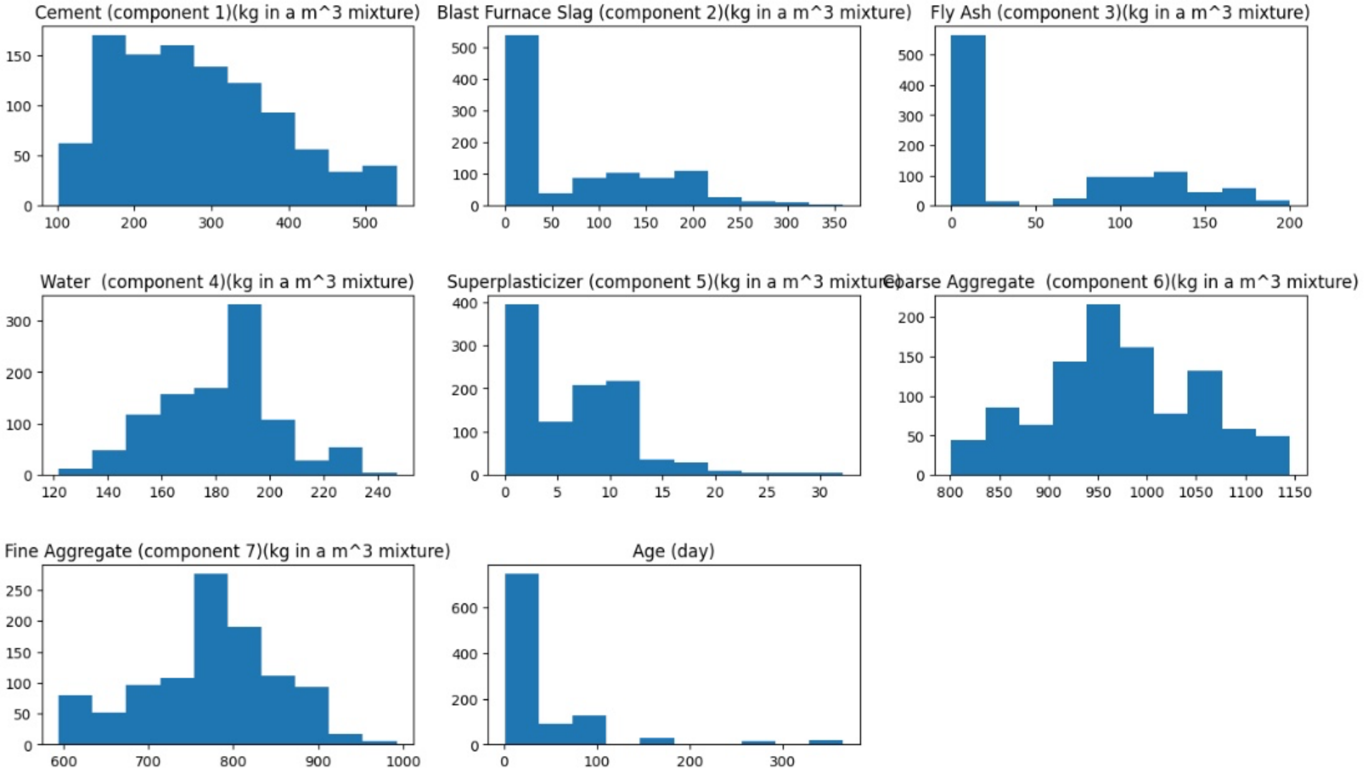
The data has been standardized by subtracting the average of a column from each value in that column and dividing the value by the column's standard deviation. This is a common form of data pre-processing and creates a dataset with common values across all features because everything is centered with an average of 0. The standardized values are labeled as "x_new" and "y_new" in the code. Throughout the code, the data before normalizing is called referred to as "raw" and the normalized data is referred to as "new".

3) Splitting Training & Testing

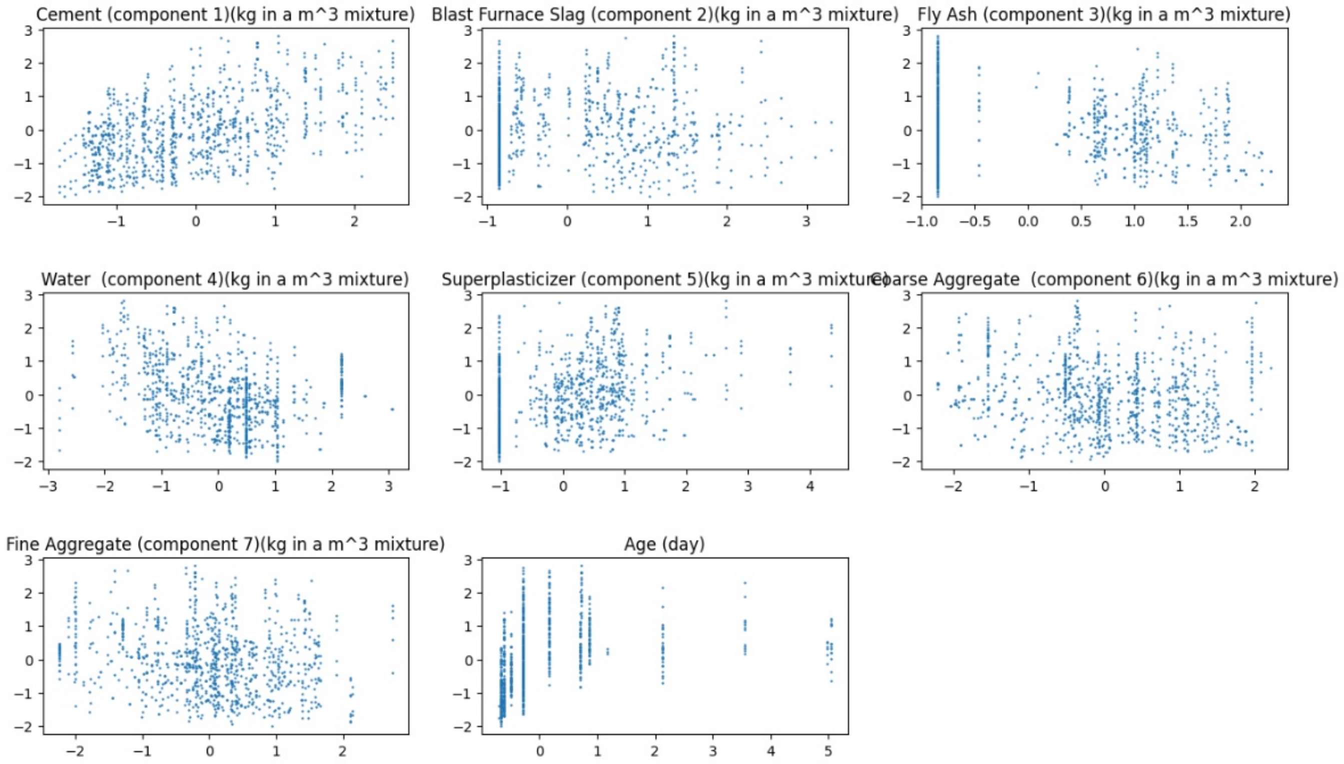
Next, the raw and new data are split using the built in sklearn function "train_test_split" with a test size of 0.1255 which creates 900 training data points and the remaining 130 data points are testing data points.

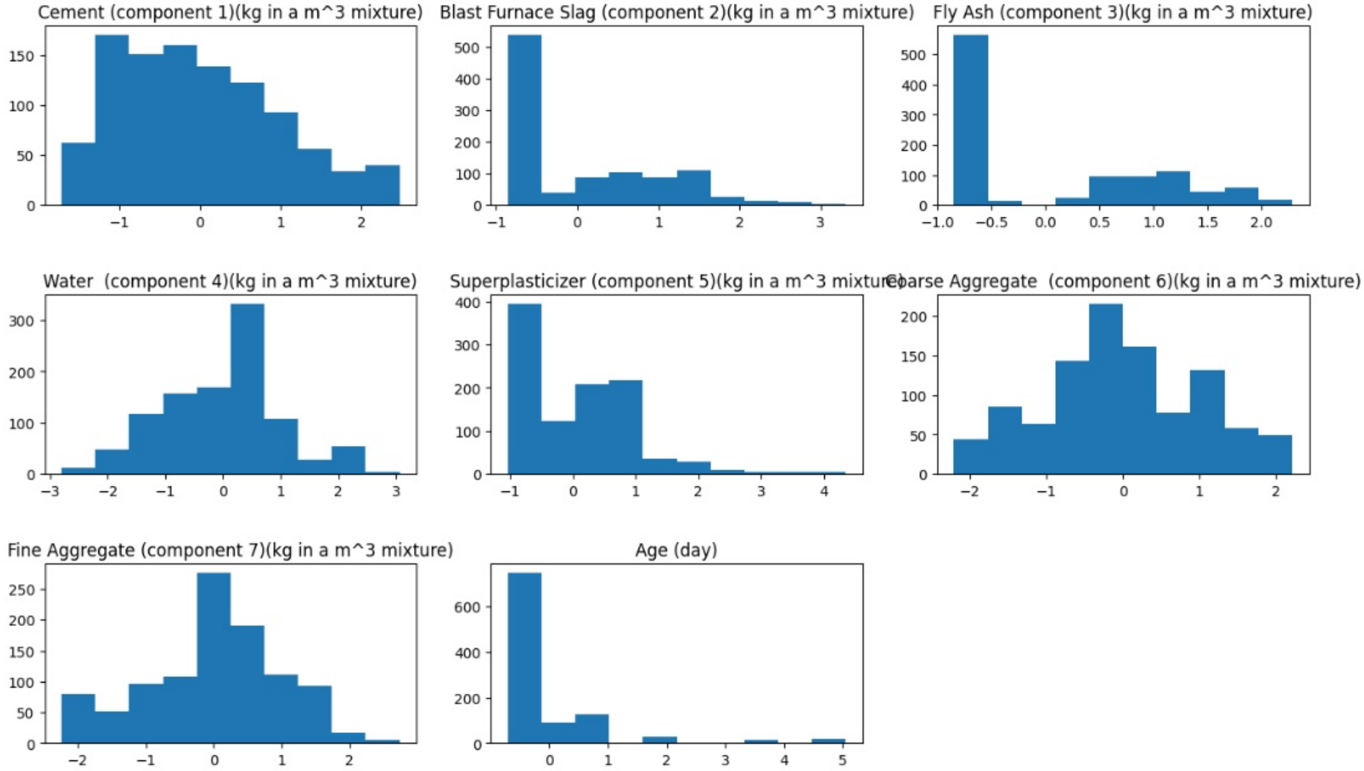
Features vs. label before pre-processing:





Features vs. label after pre-processing:





C. *Algorithm Details*

1) Stopping Criterion

The stopping criterion chosen for each of the models is when the iteration count exceeds 2000. Several trials using values of the gradient of the loss function as the stopping criterion were experimented with, but each case resulted in worse results than an interation count of 2000. The iteration count of 2000 was chosen because it balances both model accuracy and complexity and ensures positive R^2 values for the vast majority of the models.

2) Parameter Update Procedure

The parameter procedure is performed using standard gradient descent, not stochastic gradient descent. This is largely done because stochastic gradient descent would drastically increase the volume of computations required to update m and b to values that would result in an acceptable MSE and variance explained. Therefore, m and b are updated for each data point until the stopping criterion is met and the final values are used as the optimal m and b of the model.

3) Learning Rate

The learning rate for each model was selected using the best_alphaUni and best_alphaMulti (and their variations for MAE & Ridge) methods which take in the x and y vectors used to train the model. A vector of 5 alpha values evenly spaced between two values is initialized inside each method. For each α value, a univariate or multivariate model is created and the MSE (initialized at 1000000 so any MSE is better than the intial) is compared to the current best MSE of other models with different α values. The α value of the model with the lowest MSE is returned as the best_alpha for the given x and y vectors. For univariate models, a separate α value is determined for each feature.

D. *Algorithm Pseudo-Code*

1) Find best alpha values for given model

```
Check range of alpha values for each feature in univariate & entire model in multivariate
    Compare MSE values of models with different alpha values & chose model with lowest MSE
```

2) Find m, b & MSE for models w/ optimized alpha values

```
Gradient Descent algorithm:
Input: x, y (labels), feature number (only for univariate), alpha (learning rate), lambda (Ridge Regression)
Initialize parameters (m = 0, b = mean(y)) & other variables (sum_m, sum_b, dLdm, dLdb, iteration count)
Perform gradient descent while gradient > 10e-6 or iteration count < 2000
    For each data point in x, find the predicted value using the current m & b: $y^* = m * x_i + b$
    Add $\frac{\partial L}{\partial m} = \frac{1}{n}\sum_{i=1}^{n} -2 * x_i * (y - (m * x_i + b))$ to a sum
    Add $\frac{\partial L}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} -2 * (y - (m * x_i + b))$ to a sum
Find final m & b gradient for all data points by dividing both sums by n
```

Update m: $m_{new} = m_{old} - \alpha * \frac{\partial L}{\partial m}$

Update b: $b_{new} = b_{old} - \alpha * \frac{\partial L}{\partial b}$

Set $m_{new} = m$ and $b_{new} = b$

Calculate Variance, MSE, MAE (if applicable) & R^2

Return final m, b, MSE, MAE (if applicable) and R^2

3) Use final m & b values to evaluate model performanc on testing data

Initialize MSE, Variance, and R^2 lists

Find the Variance, MSE, MAE (if applicable), and R^2 values for each model

- Variance is evaluated using the built in numpy function np.var( )
- MSE is evaluated using the appropriate MSE function defined earlier in the code where

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - (m * x + b)^2$$

- R^2 or variance explained is evaluated using the previous two values in the equation below

$$R^2 = 1 - \frac{MSE}{Variance(observed)}$$

E. MAE Implementation

The implementation of MAE is very similar to the algorithm for MSE with MAE as the loss function

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(y - (m * x + b)^2$$

This changes the gradient functions for the parameter weights m and b to

$$\frac{\partial L}{\partial m} = \frac{1}{n}\sum_{i=1}^{n} -2 * x_i * |\, y - (m * x_i + b)|$$

$$\frac{\partial L}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} -2 * |y - (m * x_i + b)|$$

Additionally, the “best” α values used for each model are different. These values are found by comparing models with different α values like the MSE α selection method but uses MAE as the comparison measure rather than MSE. Therefore, the best α value for the model for each feature is the value that results in the lowest MAE out of the range of values specified in the method (chosen on trial and error).

F. *Ridge Regression Implementation*

The implementation of Ridge Regression is very similar to the algorithm for MSE with an L_2 norm term added to the sum with a hyperparameter λ multiplied by the weight(s) of the features. As a result, the gradient functions for the parameter weights m and b to.

$$\frac{\partial L}{\partial m} = \frac{1}{n}\sum_{i=1}^{n} -2 * x_i * (y - (m * x_i + b)) + \lambda$$

$$\frac{\partial L}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} -2 * (y - (m * x_i + b)) + \lambda$$

The optimal values for α are computed using the same method as the MSE models because the effect of λ on the step size is assumed to be negligent. However, the λ hyperparamter is tuned using a similar method to α hyperparameter tuning for MSE and MAE. In the method, the λ used in the model with the lowest MSE is choses as the optimal hyperparameter for the feature or features in the model.

II. Results

A. *MSE*

Range of alpha values:

- Univariate Data: [1e-6, 1e-5]
- Multivariate Data: [1e-7, 1e-6]

1) *Raw Data*

	Best alpha	MSE	Variance observed	Train R^2 (Variance Explained)	Test R^2 (Variance Explained)
Feature 1	3.25e-6	209.8614	300.8179	0.2283	0.3024
Feature 2	3.25e-6	297.0304	300.8179	0.0119	0.0126
Feature 3	3.25e-6	290.0713	300.8179	0.0061	0.0357
Feature 4	1e-5	286.5146	300.8179	0.0652	0.0475
Feature 5	5.5e-6	273.1951	300.8179	0.1206	0.0918
Feature 6	1e-6	301.6408	300.8179	0.0013	-0.0027
Feature 7	1e-6	302.5149	300.8179	0.0024	-0.0056
Feature 8	3.25e-6	279.2512	300.8179	0.0847	0.0717
Multivariate	3.25e-7	176.5178	300.8179	0.4026	0.4132

2) New Data

	Best alpha	MSE	Variance observed	Train R^2 (Variance Explained)	Test R^2 (Variance Explained)
Feature 1	7.75e-6	0.7534	1.0779	0.2385	0.3002
Feature 2	7.75e-6	1.0518	1.0779	0.0167	0.0242
Feature 3	7.75e-6	1.0481	1.0779	0.0079	0.0276
Feature 4	7.75e-6	1.0104	1.0779	0.0866	0.0626
Feature 5	7.75e-6	0.9451	1.0779	0.1350	0.1232
Feature 6	7.75e-6	1.0281	1.0779	0.0235	0.0462
Feature 7	7.75e-6	1.0651	1.0779	0.0299	0.0119
Feature 8	7.75e-6	0.9479	1.0779	0.1045	0.1206
Multivariate	3.25e-7	0.4215	1.0779	0.5447	0.6089

B. *MAE*

- Univariate Data: [1e-9, 1e-7]
- Multivariate Data: use value from Multivariate w/ MSE as loss function

1) New Data

	Best alpha	MAE	MSE	Variance observed	Train R^2 (Variance Explained)	Test R^2 (Variance Explained)
Feature 1	7.525e-7	0.6961	0.7635	1.0779	0.2318	0.2917
Feature 2	7.525e-7	0.8101	1.0718	1.0779	0.0100	0.0056
Feature 3	5.05e-7	0.8086	1.0611	1.0779	0.0052	0.0156
Feature 4	7.525e-7	0.7872	1.0257	1.0779	0.07661	0.0484
Feature 5	1e-6	0.7602	0.9775	1.0779	0.1210	0.0931
Feature 6	1e-6	0.7839	1.0519	1.0779	0.0123	0.0241
Feature 7	5.05e-7	0.8078	1.0743	1.0779	0.0273	0.0034
Feature 8	5.05e-7	0.7527	0.9722	1.0779	0.1017	0.0981
Multivariate	7.75e-7	0.5275	0.4415	1.0779	0.5346	0.5904

C. *Ridge Regression*

Lambda Optimization

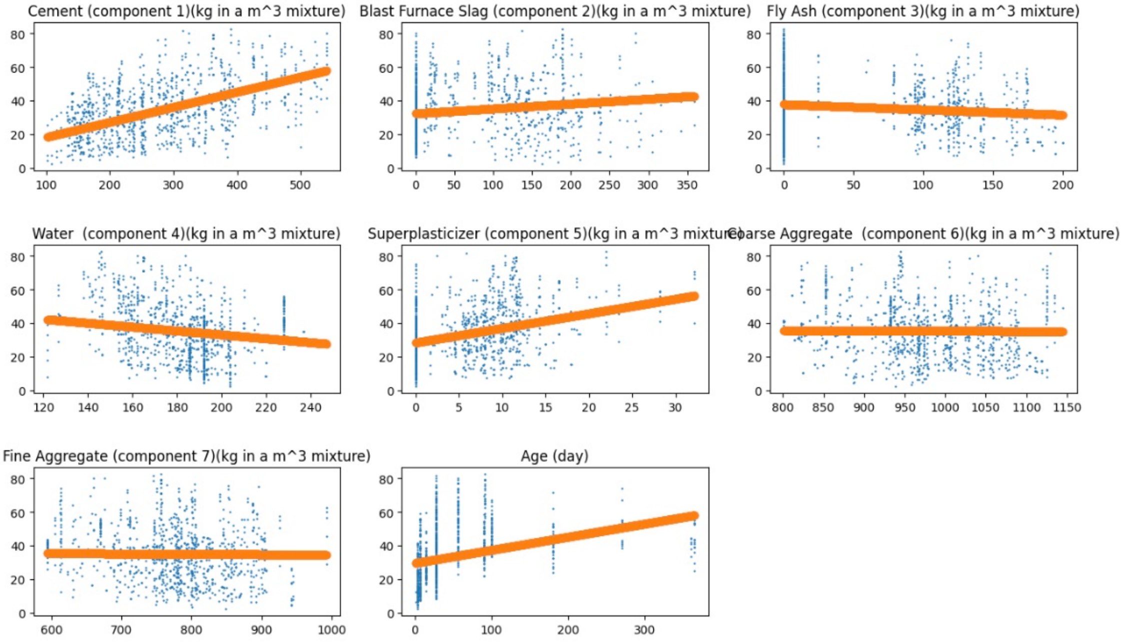
- Univariate Data: [7.75e-9, 7.75e-7]
- Multivariate Data: use value from Multivariate w/ MSE as loss function

1) *New Data*

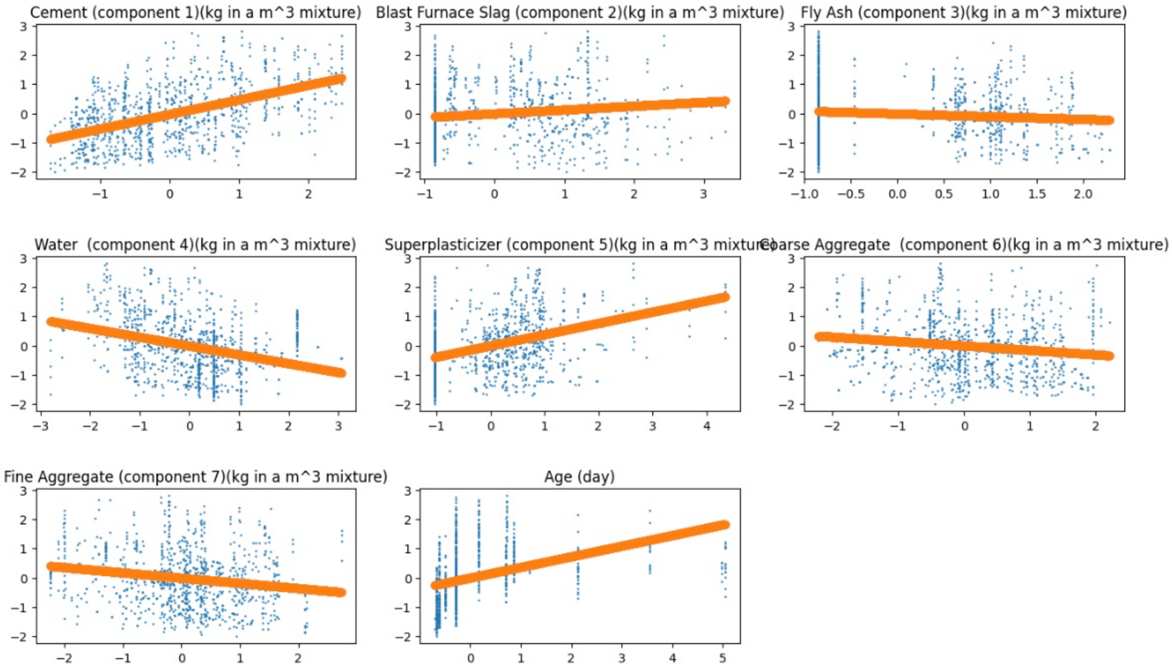
	Best lambda	Best alpha	MSE	Variance observed	Train R^2 (Variance Explained)	Test R^2 (Variance Explained)
Feature 1	7.75e-8	7.525e-7	1.0596	1.0779	0.0963	0.2318
Feature 2	7.75e-8	7.525e-7	1.0797	1.0779	0.0068	0.0077
Feature 3	7.75e-8	5.05e-7	1.0799	1.0779	0.0064	0.0282
Feature 4	7.75e-8	7.525e-7	1.0754	1.0779	0.0347	-0.0051
Feature 5	7.75e-8	1e-6	1.0713	1.0779	0.0119	-0.0052
Feature 6	7.75e-8	1e-6	1.0784	1.0779	0.0024	0.0261
Feature 7	7.75e-8	5.05e-7	1.0797	1.0779	0.0244	-0.0065
Feature 8	7.75e-8	5.05e-7	1.0779	1.0779	0.0848	0.1067
Multivariate	3.25e-10	3.25e-7	0.4215	1.0779	0.5447	0.6089

D. *Plots of Data w/ Model*

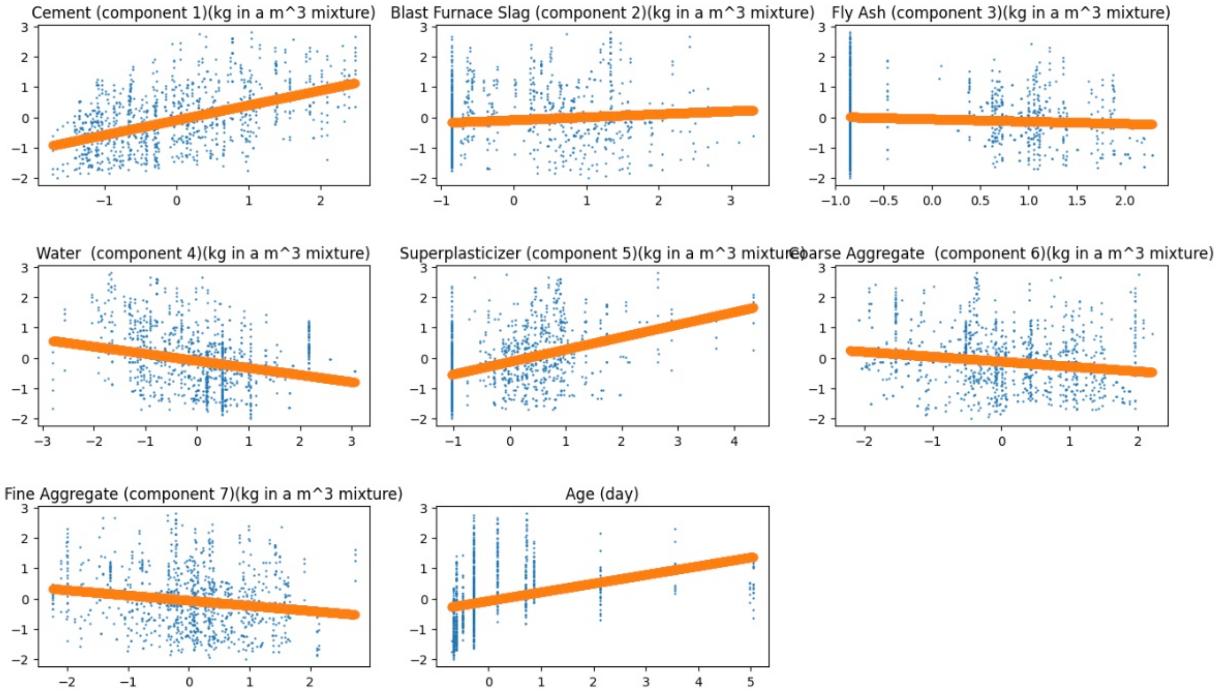
1) *MSE on Raw Data*



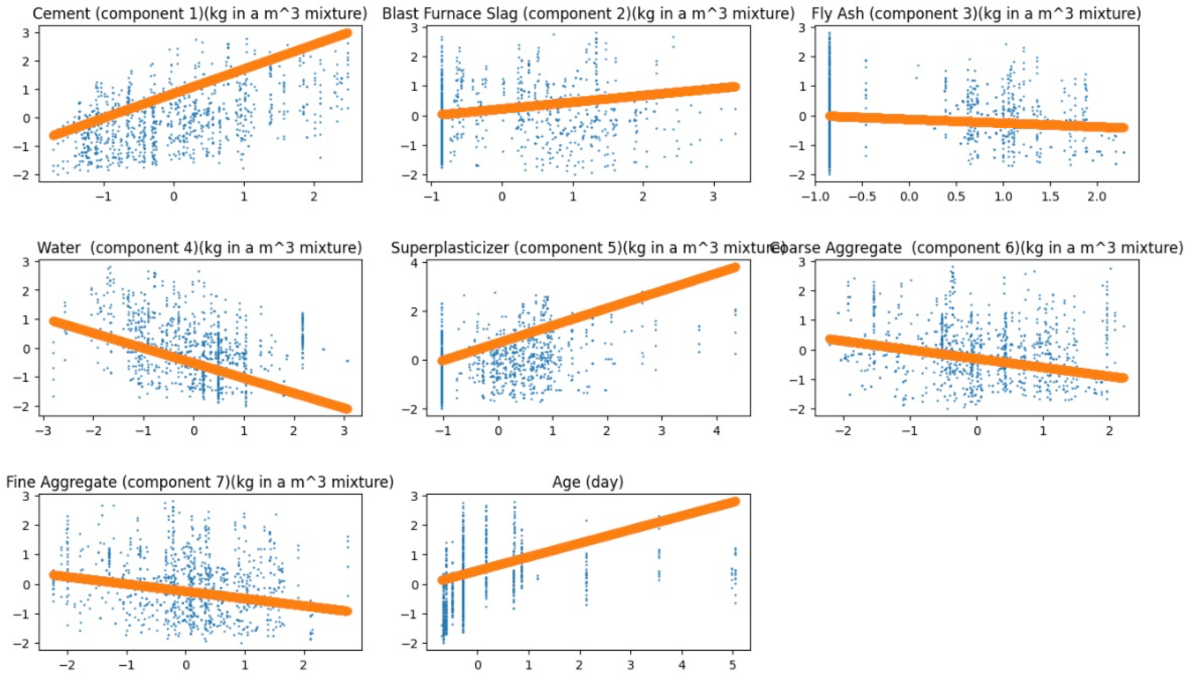
2) *MSE on New Data*



3) *MAE on New Data*



4) *Ridge Regression on New Data*



III. Discussion

A. *Model Comparision*

1) *Variance Explained (R^2 values)*

When looking at both the raw and new data, the best models were the multivariate models that contain all 8 features, rather than the univariate models that only contain one feature. This is expected because the more features that make up a model, the higher the likelihood that the model complexity will match the complexity and trends in the training and testing data it predicts.

Among the univariate models created on raw data, the models with the lowest MSE are built using features 1, 5, and 8 (Cement, Superplasticizer, and Age) with MSE values of 209.8614, 273.1951, and 279.2512 respectively. These models also had the highest variance explained (R^2) values of 0.3024, 0.0918, and 0.0717 respectively, which is expected because variance explained depends on MSE and variance observed, and variance observed is constant across all models. Similarly, for the univariate models on new data, the best features were 1, 5, and 8 with MSE values of 0.7534, 0.9451, and 0.9479 and R^2 values of 0.3002, 0.1232, and 0.1206 respectively. The multivariate models using MSE as the loss function have a R^2 of 0.4132 for raw data and 0.6089 for new data. Based on the R^2 values for raw and new data, the new data models were better predictors of cement concrete strength, so pre-processing the data strengthened our models and reduced error.

Comparing the univariate models on new data using MAE as the loss function reveals that the features with the best models are 1, 5, and 8 (Cement, Superplasticizer, and Age). The MAE values of for each model are 0.6961, 0.7602, and 0.7527 and they have R^2 values of 0.2917, 0.0931, and 0.0981 respectively. These models are the sane top performing models as the algorithm using MSE as the loss function, indicating that these features are likely the strongest predictors of concrete strength, especially Cement which has the lowest error and highest R^2 consistently. Predictably, the multivariate model performed much better than the univariate models with an MSE of 0.4415 and R^2 of 0.5904.

The Ridge Regression top performing univariate models are built using features 1 and 8 with MSE values of 1.0596 and 1.0779 and R^2 values of 0.2318 and 0.1067 respectively. Interestingly, feature 5's model does not perform well in comparison to other features as it did with MSE and MAE. This may indicate that feature 5 may not have as significant of a contribution to Concrete Strength as previously believed based on the MSE and MAE models. Despite this variation from previous results, the multivariate regression model still performs noticeably better than the univariate models with a R^2 of 0.6089.

2) *Training vs. Testing Models*

For most of the univariate models and all the multivariate models, the model performed better on the testing data than on the training data. This is an indication of good model performance because the models are not overfit to the training data but have enough information from the training data to accurately predict unseen data. This conclusion is demonstrated by the increased R^2 values in the testing data models compared to the training data models.

3) *Hyperparamter Tuning*

When tuning the hyperparameter α to find an optimal learning rate for updating the weight parameters and intercept, trial and error was implemented to find the best range of values to select α from. For univariate MSE, the α value is selected from 5 evenly spaced values in the range [1e-6, 1e-5]. For multivariate MSE, the alpha value was lowered because there are more features in the model, so a smaller step size allows to for fine tuning of all features without any stepping too far. The multivariate MSE α is selected from 5 evenly spaced values in the range [1e-7, 1e-6].

In univariate MAE, the α value is selected from 5 evenly spaced values in the range [1e-8, 1e-6] and the multivariate MAE value is selected from 5 evenly spaced values in the range [1e-7, 1e-6].

In univariate Ridge Regression, the α values used are the same α values from the MSE univariate model because like multivariate MAE, there was not a large difference in the computed optimal α values so using a previously established variable is more efficient. The optimal λ value is selected from 5 evenly spaced values in the range $[\frac{\alpha}{100}, \alpha]$ corresponding to the feature. The result is the same λ= 7.75e-8 for each univariate model. This is clearly the optimal value because λ is not on the boundary of the range. In the multivariate model, many values of λ were tested using the α from the multivariate model.

B. *Conclusions*

The factors that best predict concrete comprehensive strength are Cement (feature 1), Superplasticizer (feature 5), and Age (feature 8). The most influential factor is Cement because it consistently had the highest R^2 value among all univariate models, followed by Age, and finally by Superplasticizer. Therefore, when mixing concrete to optimize comprehensive strength, it is most important to consider Cement where higher values result in higher strength (positive parameter m value). Next, the Age of the mixture should be considered where the younger the mixture, the stronger the Concrete (negative parameter m value). Then, the Superplasticizer amount should be optimized where a higher concentration in the Concrete results in stronger

Concrete (positive parameter m value). The remaining characteristics cannot be ignored completely because statistical significance tests have not been performed to determine whether they can be dropped from the multivariate models and not affect the accuracy. Therefore, the remaining 5 features should be considered with equal weight when predicting Concrete Comprehensive Strength.

IV. INSTRUCRTIONS

1) Open the Python Jupyter notebook (.ipynb) in your desired interface tool (Jupyter Notebook, Visual Studio Code, Linux, etc.)

2) Select “Cell” -> “Run All” from the Menu Bar

** *It will take a few minutes to run all the way through but results will be printed after each cell as the program runs!*