Introduction

Problem Formulation

The data we are using is a list of features that likely affect the compressive strength of concrete in MPa. There are 1030 different data points and 8 different features provided, including the concentration of cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate in kg/m^3 as well as the age of the cement in days.

Questions

I have two separate questions I want to answer based on the univariate and multivariate cases. For the...

1. Univariate case: Which single feature is the most predictive of compressive strength?
2. Multivariate case: how well do the 8 different features of concrete predict its compressive strength?

Application-wise, understanding the former question might help a construction company to determine which feature is most important to creating strong concrete. Understanding the latter question will help that same company predict the strength of their concrete samples even without measuring strength directly.

Algorithm Details

Objective function: Mean squared error

Update step: gradient descent (non-stochastic)

Stopping criterion:

- If the change in the objective function is smaller than a defined threshold
- Else, when a defined maximum number of iterations has been reached

Learning rate (alpha):

- Univariate models: 0.0001, 0.00008, or 0.000001
- Multivariate model: 0.004
- Univariate models (normalized): 0.003
- Multivariate model (normalized): 0.01

I chose these learning rates experimentally. This would prevent my learning rates from either being too small and converging too slowly or too large and performing badly. For example, in the univariate models without normalization, I tried alpha values within the range of 1e-8 to 0.1 and noticed the best MSE at 0.000001 for most models. Because a few features had weights that grew to very large numbers, I reran specific features with higher learning rates and adjusted according to 0.0001 or 0.00008 depending on the speed of learning for that feature.

Pseudo-Code

```
function univariateLinearModel(x, y, alpha, maxIters, threshold):
    mse = infinity; m = 0; b = 1
   mse_star = mse; m_star = m; b_star = b
   iterations = 0

   WHILE iterations < maxIters and absolute change in mse > threshold:
       Calculate the mse using vectorized operations to square the error and divide by the
       number of data points.
       Update m using the alpha and partial derivative of m
       Update b using the alpha and partial derivative of b

       IF mse < mse_star:
           m_star = m; b_star = b

       increment iterations

    RETURN mse_star, m_star, b_star

function trainLinearRegression(X, y, alpha, maxIters, threshold): // multivariate model
     mse = infinity; m = array of zeros (length of the number of features); b = 1
    mse_star = mse; m_star = m; b_star = b
    iterations = 0


    // gradient descent step
    WHILE iterations < maxIters and absolute change in mse > threshold:
        sse = 0
        b_update = 0
        m_update = vector with size as the number of features

        // update step
        FOR each data point:
            Add to the sse using (y − (mx + b))^2, which is the actual - predicted. Divide by n to
            get the mse
            Add to the m_update value using the partial derivative of m. Divide the total by n
            Add to the b_update value using the partial derivative of b. Divide the total by n
```

// update step
m = m + m_update
b = b + b_update

IF the mse is smaller than the best mse (mse_star):
m_star = m
b_star = b

RETURN mse_star, m_star, b_star

function calculateVarianceExplained(X, y, m, b, univariate):
IF X has one feature:
Calculate the squared error for each data point, sum the vector and divide by n to get the mse

ELSE:
sse = 0

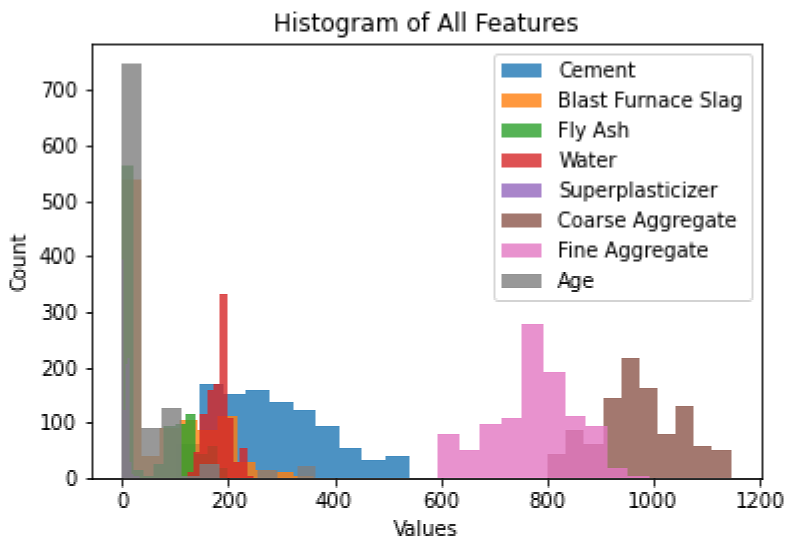
FOR each data point:
add to the sse using $(y - (mx + b))^2$*, which is the actual - predicted. Divide by n to get the mse*

Calculate the variance explained using $1 - \frac{mse}{variance\ observed}$*, where variance observed is the variance of y*

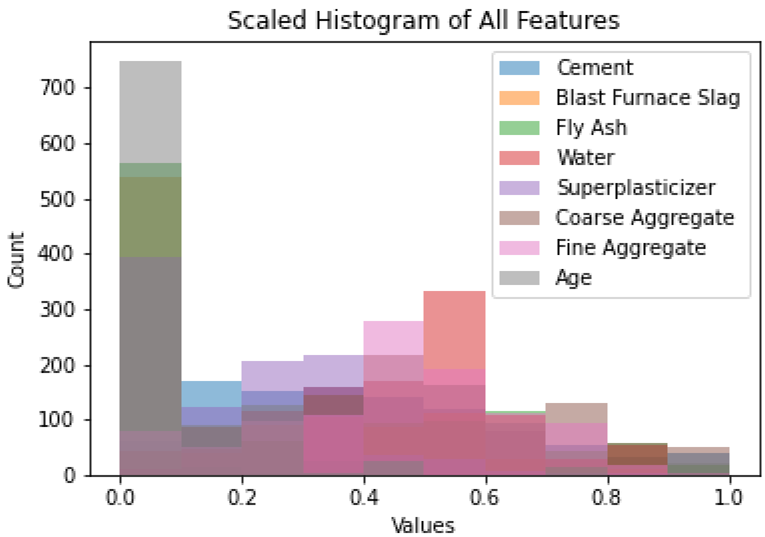
RETURN the variance explained

Data Normalization

I chose a min-max scaler to normalize my data. Below are the histograms for all features which helped me make this decision:



The two main things I noticed are the different scales of measurement and the non-normal distributions. For one, knowing that most of the features are in the same units of kg/m^3, seeing the large spread of data in this plot is already good motivation to normalize the data. That way, certain features will not drastically influence the gradient. Secondly, although coarse aggregate and fine aggregate look normally distributed, the right-skew of distributions like age and fly ash shows that not all features collected are normally distributed. As a result, I decided against standardizing my data, as standardization seems to work best when all features are normally distributed. (I end up revisiting this topic in my discussion) This led me to min-max scaling, which I did using the help of sci-kit learn. Min-max scaling is applied to each datapoint where the new value $x' = \frac{x - x_{max}}{x_{max} - x_{min}}$. Given this equation, the values should conform to a range of [0,1] but the distributions should stay the same. This is explained by the histograms shown below:



All of the features now seem to be within the same scale! Though difficult to tell on this graph, only the values—not the shapes of the distributions—have changed because this is a scaling method. As a summary, I used the histograms to determine that there was a large spread of data and that some features were non-normal, so I applied min-max scaling over standardization.

Results

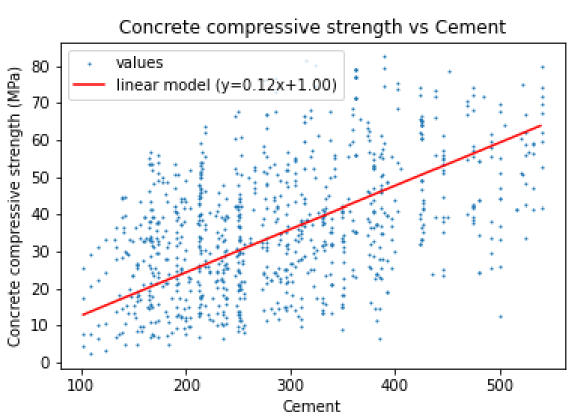
Variance Explained

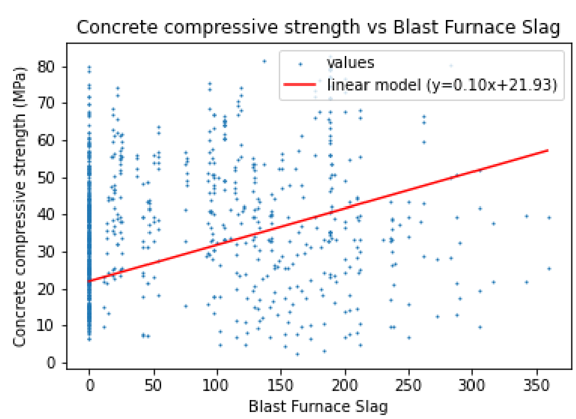
Below is the variance explained values for the 8 univariate models (each feature) and the 1 multivariate model (all features together) separated by training and test set.

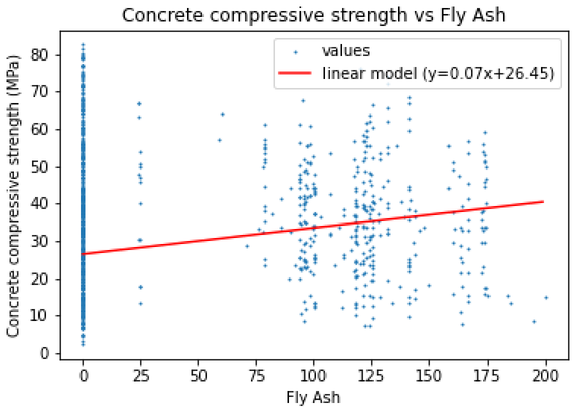
Models (features used)	MSE (training set)	MSE (test set)	Variance Explained (training set)	Variance Explained (test set)
Cement	246.380	106.859	0.167	0.256
Blast Furnace Slag	388.300	134.838	-0.312	0.062
Fly Ash	363.442	223.499	-0.228	-0.555
Water	355.971	185.030	-0.203	-0.287
Superplasticizer	296.065	234.400	-0.001	-0.631
Coarse Aggregate	321.414	165.766	-0.086	-0.153
Fine Aggregate	332.017	170.908	-0.122	-0.189
Age	308.916	159.847	-0.044	-0.112
All Eight Features	119.809	63.152	0.595	0.561

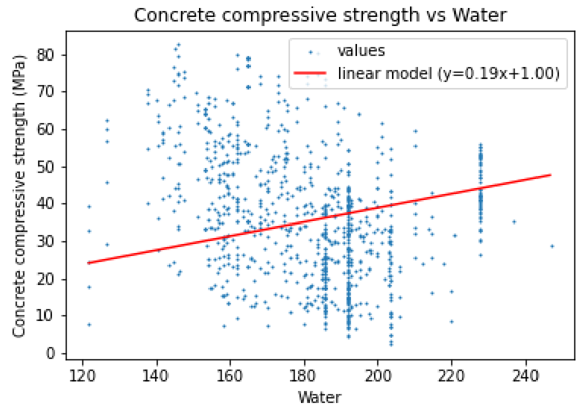
Univariate Training Set Plots

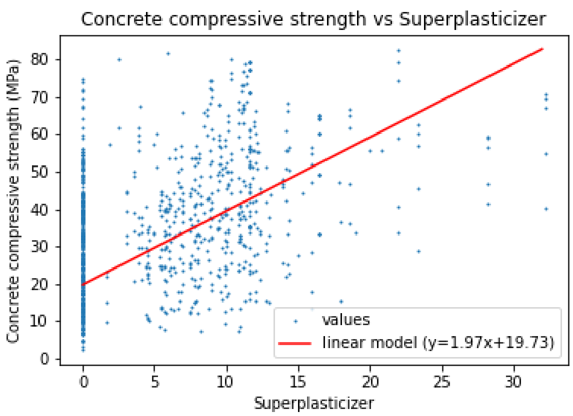
Below are the target vs feature plots for all eight features individually which includes the datapoints in blue and its corresponding model fit in red.

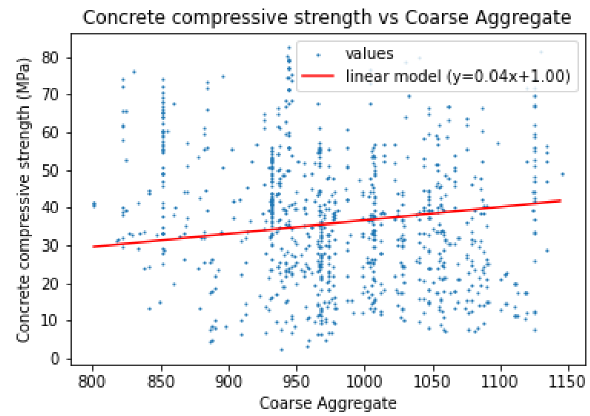


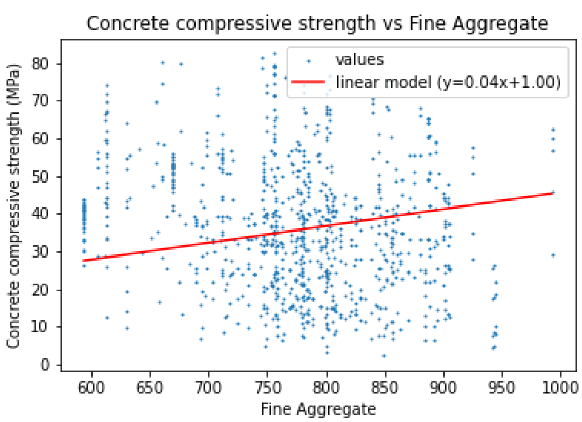


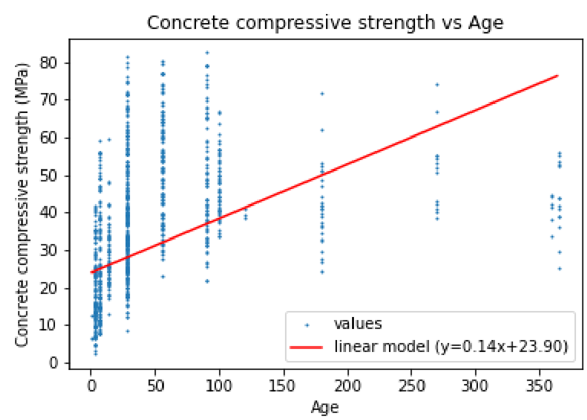

Concrete compressive strength vs Fly Ash
values
linear model (y=0.07x+26.45)
Concrete compressive strength (MPa)
Fly Ash


Concrete compressive strength vs Water
values
linear model (y=0.19x+1.00)
Concrete compressive strength (MPa)
Water


Concrete compressive strength vs Superplasticizer
values
linear model (y=1.97x+19.73)
Concrete compressive strength (MPa)
Superplasticizer


Concrete compressive strength vs Coarse Aggregate
values
linear model (y=0.04x+1.00)
Concrete compressive strength (MPa)
Coarse Aggregate


Concrete compressive strength vs Fine Aggregate
values
linear model (y=0.04x+1.00)
Concrete compressive strength (MPa)
Fine Aggregate


Concrete compressive strength vs Age
values
linear model (y=0.14x+23.90)
Concrete compressive strength (MPa)
Age

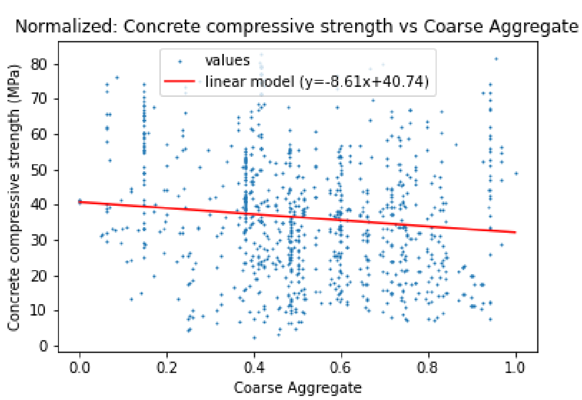
Normalized Variance Explained

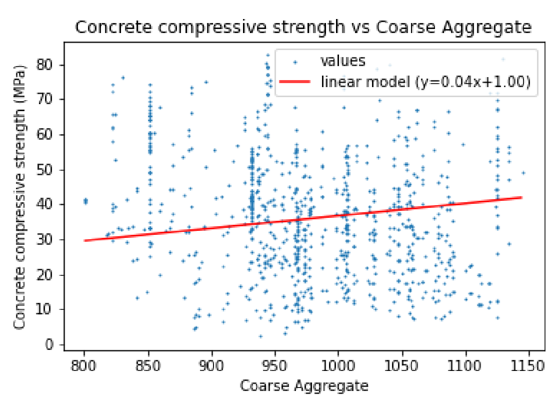
To compare results between the normalized and not normalized datasets, I recalculated the variance explained for the scaled datasets.

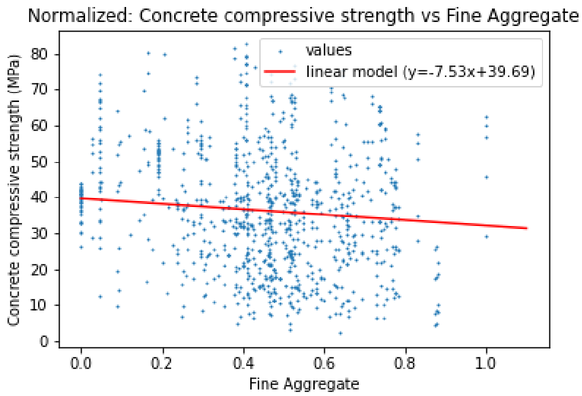
Models (features used)	MSE (training set)	MSE (test set)	Variance Explained (training set)	Variance Explained (test set)
Cement	230.075	84.686	0.222	0.411
Blast Furnace Slag	291.066	157.051	0.016	-0.093
Fly Ash	296.190	166.051	-0.001	-0.155
Water	273.491	152.837	0.076	-0.063
Superplasticizer	246.646	216.159	0.166	-0.504
Coarse Aggregate	286.707	172.296	0.031	-0.199
Fine Aggregate	288.744	163.051	0.024	-0.135
Age	265.398	153.942	0.103	-0.071
All Eight Features	117.800	72.947	0.602	0.492

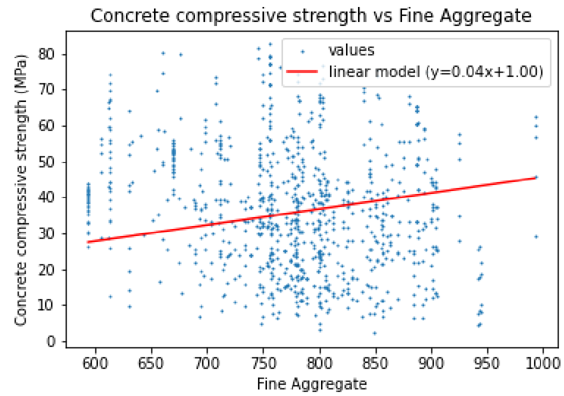
Normalized Univariate Training Set Plots

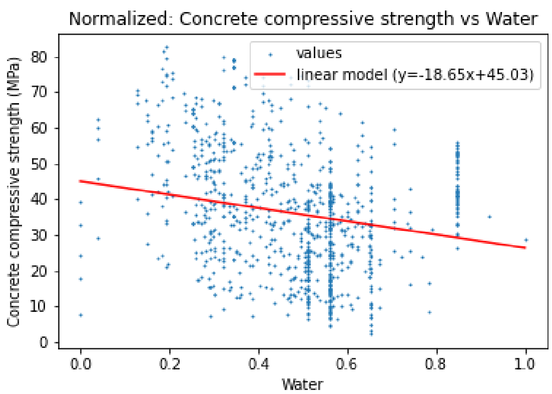
I also did similar scatter plots to describe the normalized plots. After scaling, some of the slope coefficients for the training plots flipped from positive to negative, including coarse aggregate, fine aggregate, and water. The normalized plots (left) along with their original plots (right) are described below:

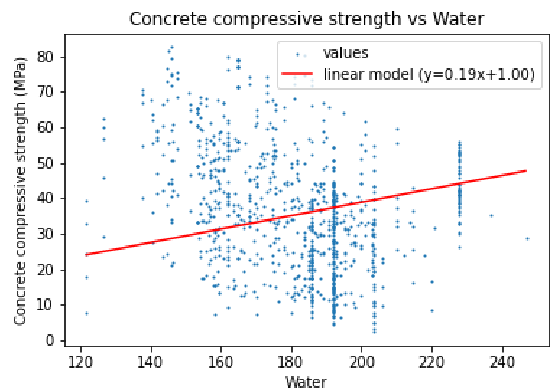




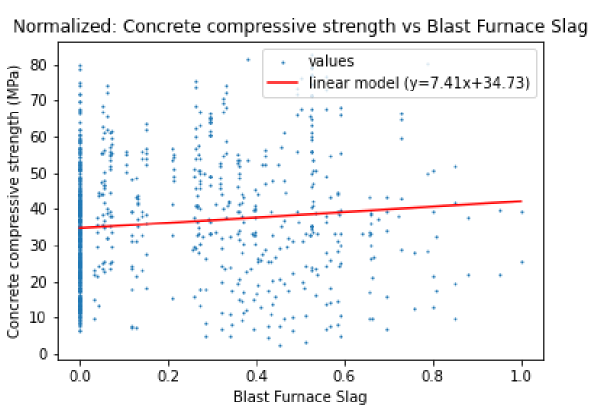


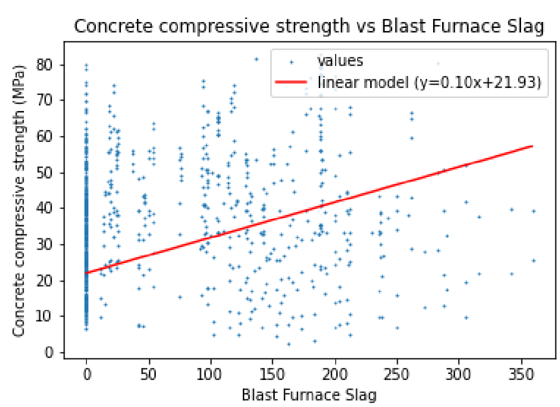


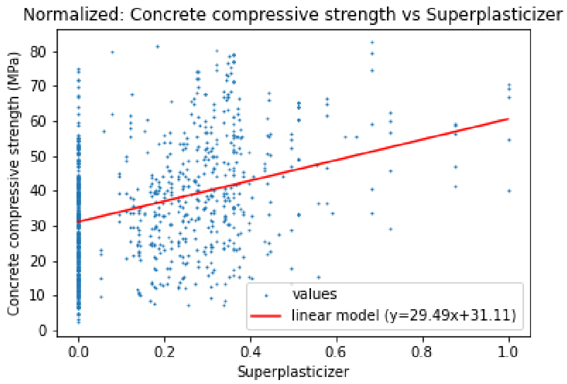


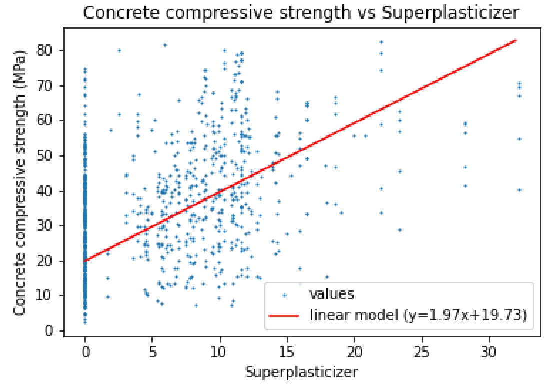


It is important to note that the slopes can be qualitatively compared, but not numerical compared as the features are now on different scales (see the x-axis values), while the target values have stayed the same. Another thing we noticed was that the slopes seem to be less drastic for features like Blast Furnace Slag and Superplasticizer.






Normalized: Concrete compressive strength vs Superplasticizer
Concrete compressive strength (MPa)
Superplasticizer
values
linear model (y=29.49x+31.11)


Concrete compressive strength vs Superplasticizer
Concrete compressive strength (MPa)
Superplasticizer
values
linear model (y=1.97x+19.73)

Discussion

Model Comparisons

Given that there is some generalization error that should occur as a result of testing on data points outside of the training dataset, I would expect that the explained variance for the testing set should be lower than the training dataset. After calculating these variances for the 9 models, 6 of them (all but Cement, Blast Furnace Slag, and the multivariate models) had a lower variance explained, which aligned with my initial hypothesis. However, there were 3 models that had a higher variance explained. The increased explained variation might be indicative of 'getting lucky' on the testing dataset due to a small test data set (130 samples), which is about 13% of the total data (130/1030). Notice that the MSEs had the opposite behavior, as all models decreased in MSE from the training to the test dataset.

The only two models that seemed to perform well on the training dataset, i.e. having a positive explained variance, were the cement and all-feature model. These two models also performed relatively well on the testing, explaining around the same variance as the training set or more. However, one surprising result was that the Blast Furnace Slag feature went from having a negative explained variance value to a slightly positive value. This may be because the samples in the test set were not as clustered around 0 as it was in the training set (see the univariate plot above). As for the models that already performed badly, they mostly tended to perform worse by explaining even less variance than before.

Comparison of Coefficients

Below are the features and their associated coefficients of the individual univariate models compared to the single multivariate model:

```
Features and their coefficients...
Feature         univariate_m    multivariate_m  univariate_b
Cement          0.117           0.115           1.000
Blast           0.098           0.095           21.934
Fly As          0.070           0.103           26.454
Water           0.189           -0.128          1.001
Superp          1.969           0.035           19.732
Coarse          0.036           -0.000          1.000
Fine A          0.045           0.011           1.000
Age             0.144           0.105           23.904
```

Based on these values, it seems like most of the m coefficients were pretty well predicted with most off around 0.04 at maximum. The coefficients that were not well predicted at all were water and superplasticizer. Looking at the training data plots helps clarify why this is the case, as these three features had lots of data points that were collected at similar x-values. These similar x-values can be seen by the more ‘dense’ vertical lines formed by the blue data points. That being said, it seems surprising that blast furnace slag and fly ash did not have as large of a change in coefficient, as they both also had many points near zero (left-skew).

Predictive Factors for Concrete Compressive Strength

First looking at the explained variance, it seems like focusing on the concentration of cement is really important in making concrete, as it was able to capture around a quarter (~25%) of the observed variance in the test dataset and nearly half (0.256/0.561) of the variance compared to the multivariate model with all features. More specifically, because of its positive coefficient value, I would expect that a larger concentration of cement in concrete is correlated to its compressive strength.

Furthermore, although they do not seem as useful on their own, the other seven features seem to help contribute to the multivariate model. Notably, the age of the concrete seems positively correlated with compressive strength (0.105 in multivariate), though it does seem to have more of a decrease in the larger values which is not captured by the non-curved linear fit. Also, the water feature seems negatively correlated (-0.128 in multivariate). For coefficients that are relatively small or zero, one conclusion would be that the concentrations of such materials are not as useful in predicting compressive strength. All in all, to build the hardest possible concrete, I would recommend focusing on increasing the cement concentration as well as giving it more time and less water.

Comparisons to Normalized Data

Since the weights are on a different scale, I will mostly refer to the differences in MSE and explained variance. Below is the combined table comparing the normalized data to the original data in terms of MSE:

Models (features used)	**MSE (training set)**	**MSE scaled (training set)**	**MSE (test set)**	**MSE scaled (test set)**
Cement	246.380	230.075	106.859	84.686
Blast Furnace Slag	388.300	291.066	134.838	157.051
Fly Ash	363.442	296.190	223.499	166.051
Water	355.971	273.491	185.030	152.837
Superplasticizer	296.065	246.646	234.400	216.159
Coarse Aggregate	321.414	286.707	165.766	172.296
Fine Aggregate	332.017	288.744	170.908	163.051
Age	308.916	265.398	159.847	153.942
All Eight Features	119.809	117.800	63.152	72.947

The normalization of data helped during learning, but seemed to fall short during the testing phase. For one, because all values fell within the same interval of [0,1], normalization allowed me to use the same learning rate for all my univariate models. This made the learning rate larger than the previous learning rates I used for the original data (0.01 vs 0.004 for the multivariate model). In terms of MSEs, the normalized data seemed promising, having lower training and test MSEs for most of the features. However, in order to consider the full picture, the explained variance must also be compared:

Models (features used)	**Variance Explained (training set)**	**Scaled Var Explained (training set)**	**Variance Explained (test set)**	**Scaled Var Explained (test set)**
Cement	0.167	0.222	0.256	0.411
Blast Furnace Slag	-0.312	0.016	0.062	-0.093
Fly Ash	-0.228	-0.001	-0.555	-0.155
Water	-0.203	0.076	-0.287	-0.063
Superplasticizer	-0.001	0.166	-0.631	-0.504
Coarse Aggregate	-0.086	0.031	-0.153	-0.199

Fine Aggregate	-0.122	0.024	-0.189	-0.135
Age	-0.044	0.103	-0.112	-0.071
All Eight Features	0.595	0.602	0.561	0.492

Looking at the train-test difference for the normalized data, the explained variance seems to decrease for all models except for cement. This might point to overfitting in the normalized dataset, which is surprising considering that the only difference between the two datasets is scaling. This might be because the number of iterations caused the data to approach even closer to the minimum of the training data and hence causing overfitting when testing the model.

Seeing the mixed improvements of the data leads me to wonder if I should have standardized my data instead of normalizing it. Knowing that there were quite a few features that looked normally distributed, it might have been useful to standardize so that we could see improvements in the model. All in all, I expected there to be more gains in standardizing the model, but the explained variance, in particular, shows that this was not necessarily the case.

In summary, as a response to my two focus questions:

1. Cement concentration seemed to be the greatest indicator of compressive strength of the two, as it explained about a quarter of the variance in the dataset.
2. Using all of the features in the dataset allows us to predict over half of the variance, but it seems like there are other features outside of the dataset or more complex models that need to be considered to better predict concrete compressive strength.