Concrete Compressive Strength

David Flores Diaz

ITESM Campus Querétaro

1. ABSTRACT

The use of machine learning, in this case a linear regression algorithm to model the behavior of the concrete compressive strength, trained with 1030 samples.

2. INTRODUCTION

First let's talk about the value in having a model that can predict the compressive strength of concrete. Concrete is used in every building, every road, and knowing how much stress it can sustain, can help engineers develop more accurate models and sketches of how a building can behave after its built, but also it can help see if the concrete passes construction standards.

The normal way to proceed to get the compressive strength of concrete is through experimentation, using a machine that applies strength to a sample until the sample cracks or fails.

A linear regression algorithm will provide a model that is based in linear behavior, how each variable is correlated to the output in a linear influence. Using this type of model, we could predict how the concrete will behave before testing, and also trying other variants of concrete or mixes.

3. DATASET

The dataset is composed by 1030 samples each with its own value for the following variables

- Amount of Cement $\left[\frac{kg}{m^3}\right]$
- Amount of Blast Furnace Slag $\left[\frac{kg}{m^3}\right]$
- Amount of Fly Ash $\left[\frac{kg}{m^3}\right]$
- Amount of Water $\left[\frac{kg}{m^3}\right]$
- Amount of Superplasticizer $\left[\frac{kg}{m^3}\right]$
- Amount of Coarse Aggregate $\left[\frac{kg}{m^3}\right]$
- Amount of Fine Aggregate $\left[\frac{kg}{m^3}\right]$
- Age of drying (Days)

And output

- Compressive Strength $[MPa]$

All this information is contained on a CSV file which we proceeded to open in Excel so we could handle it easier.

4. NORMALIZATION

Since the values of each variable vary in magnitude, we need to scale them, so the algorithm handles them better and it has a much faster convergence. We apply the NORMALIZATION function that Excel has integrated using the average of the samples and the standard deviation, the values obtained are in terms of Z values.

These are the values used for this normalization process and the ones needed to be used with new samples

	Average	Standard De
cement	281.167864	104.506364
slag	73.8958252	86.2793417
Fly ash	54.1883495	63.9970042
water	181.567282	21.3542186
superplasticizer	6.20466019	5.97384139
Coarse aggregate	972.918932	77.753954
Fine aggregate	773.580485	80.1759801
age	45.6621359	63.1699116
csMPa	35.8179612	16.705742

5. CROSS VALIDATION

When training this model we use a method called cross validation which consists in separating the samples in n blocks and then using all of them but 2 to train the model, then one of those 2 blocks that were left used to validate that the model has been learning and then the las block to test it with never seen samples, the blocks of validation and training change with each iteration of training so that we have n-2 models and we can see if there is some sort of convergence in the results and that the diminishing of the error it's not just over-fitting or under-fitting.

6. ALGORITHM

The linear regression algorithm provides values with the function, or hypothesis function or evaluation function.

$$y = \sum_{i=0}^{n} m_n * x_n$$

It's the sum of the multiplication between each variable and a "parameter" that it's a constant that will be changing to fit the model to the actual behavior.

The linear regression algorithm has an error function Mean Squared Error.

$$MSE = \frac{1}{n}\sum (y - \overline{y})^2$$

The sum of the square of the difference between the hypothesized result and the actual result divided by the number of samples, this will tell us the error of the model.

The way the algorithm learns is through the gradient descent method and combined with the MSE to provide how much does a parameter need to change.

$$m_{a\,new} = m_{a\,old} - \frac{\alpha}{n} * \sum_{b=0}^{n} (y_b - \overline{y}_b) * x_{ab}$$

7. TRAINING

We performed a linear correlation analysis to check the strength of correlation between the output and the input variables.

water	-0.28963338
fine aggregate	-0.16724125
coarse aggregate	-0.16493461
fly ash	-0.10575492
slag	0.13482926
age	0.328873
super plasticizer	0.36607883
cement	0.49783192

The ones colored green were the ones that we were considering training the model and after training we removed one at a time to see the effect in the error, removing the water variable improved the accuracy so we discarded the use of it.

8. RESULTS

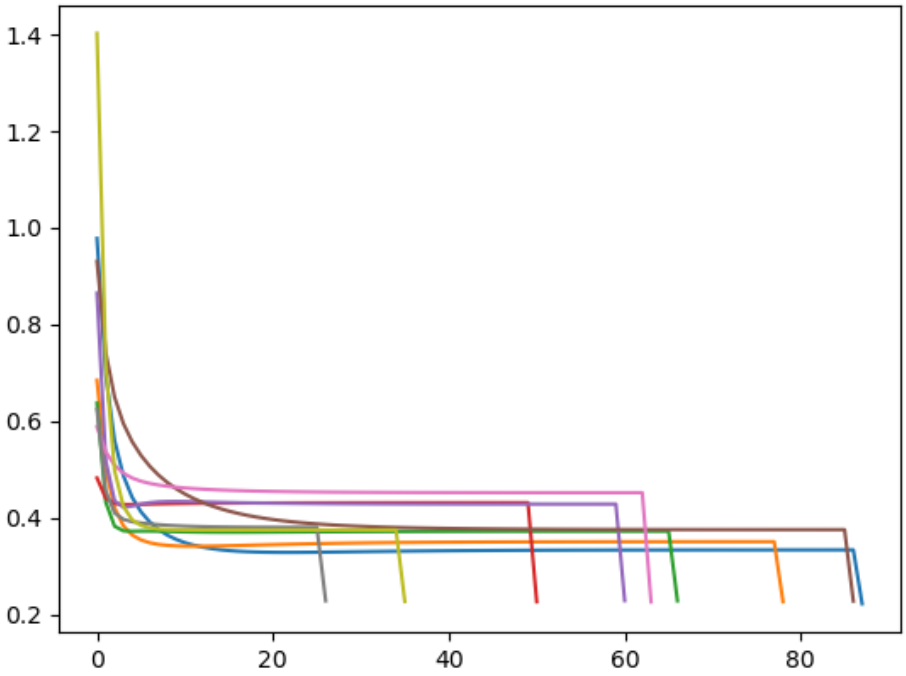

Figure 1 Error graph, each line is a run of cross-validation and the final spot is the error against the test sample

We can see the performance of the several training session were each time the validation set was changing so we have 9 different training results to see if the performance is consistent, each attempt finishes at a different epoch because it reached a convergence status differently, and all of them presented a consistent value around 25 % of error meaning that the result would be off by +- 4 MPa. The model could be further improved with more complex models, perhaps a self-evolving neural network, that uses variation of the best parameters from the previous generation, or a model that involved some more complex relations like exponential, quadratic or logarithmic relations rather than a linear relation between inputs and output.

9. REFERENCES

1. I-Cheng Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, No. 12, pp. 1797-1808 (1998).

2. Prof. I-Cheng Yeh. 2007Concrete Compressive Strength Data Set Available at: https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength