Multivariate Statistical Methods for Big Data Analysis and Process Improvement  
Concrete Compressive Strength Prediction and Analysis

Harsh Shah, B.Sc.

Dr. Brandon Corbett, Ph.D.

Dr. Leila Hoseinzadeh, Ph.D.

McMaster University

Author Note

This paper will explain in detail the techniques of building and explaining various prediction models for this dataset taken from the UCI repository.

Abstract

A machine learning model predicting the concrete compressive strength is built using various algorithms. A lot of methods produced wrong results due to overfitting, miscalculation, and also as a result of the incomprehensibility of python code. During the process, the data was explained and explored thoroughly to ensure that the resulting output is proved. All the data explored have images corresponding to curious insights that the author thought was positively or negatively affecting the output. The model was cross validated, and the results explained to describe how each input can affect the prediction model.

Keywords: Prediction, Machine Learning, R2, Concrete strength, Regression.

Multivariate Statistical Methods for Big Data Analysis and Process Improvement  
Concrete Compressive Strength Prediction and Analysis

1. **Introduction:**

This paper will introduce various methods of building a machine learning model to try and predict what the compressive strength will be (MPa). Mixing of concrete is an arduous task because there are various purposes depending on the task at hand, where different strengths are needed. This paper explores how to use different concrete mixtures with varying strengths depending on its use – industrial, commercial, construction, etc. These purposes make it one of the most widely used man-made materials on earth, and therefore it is essential that an accurate prediction model be built, where an input of different mixtures can be computed, and an accurate strength is outputted.

There can be various constraints that can skew the results, this paper does not explore those, but rather it will focus on dissecting the machine learning algorithms and data analysis methods, through exploration, implementation, and experimentation of different permutations and combinations on what inputs most affect the outcome of *Strength*.

1. **Overview**

The 7 main components the paper will be dealing with are inputs, of which only Age is in a unit of the no. of days. The remaining 6 inputs were measured in this means that how much of each of the 6 inputs are added in proportion to a volumetric unit of measurement of the mix. This information is vital because each mixture is unique, and each measurement will skew the strength positively or negatively. Some might not have any effect; and if that is the case, I will deduce why an effect occurs the way it does. Any anomalies will be investigated and with the data as evidence, I will deduce whether it is an anomaly or not. In a practical world, this would be essential in large batch process manufacturing for targeted strength concrete mixtures. My model will be able to monitor and control the entire mixing process reducing the amount of waste, improve reliability, and more importantly predict when and if something goes wrong.

1. **Data Exploration and Analysis**

In this section, I will introduce various ways I explored the data [1]. This includes some snippets of code where I will feed the dataset into my python IDE to identify and summarize some of my findings. I will also explain each of the 7 inputs. It is also noted and proven in *Figure 1* that we can see there are **no** missing values in the 1030 instances of the dataset.

* 1. **Cement**

This is in powder form. As a key ingredient this will be the first thing added to the mixture.

* 1. **Blast Furnace Slag (BF Slag in the data)**

Like the name suggests, the production of slag when coke, iron ore, and limestone are melted together in a blast furnace [2]. It is smelted in the chamber and will later be cooled in various ways. In this dataset it is not mentioned specifically which cooling process was used.

* 1. **Fly Ash**

Fly Ash is a powder like substance that is a byproduct of burning coal. This is used in the mixture to increase the strength of concrete and make it more resistant in colder climates. Although this might seem true, according to *Figure 2* it is noticeable that Fly Ash, when used actually reduces the overall strength of concrete. While this is not true for all cases, it is a noticeable insight that at 0 FLY ASH the clustering of the attribute of Strength is higher.

* 1. **Water**

Water is always present in any mixture of concrete. However, the amount of water used can skew the results a lot. In Addition, water is a scarce resource and while the goal of this project is to only predict the strength of the concrete mix, there is a lot of research being done to try to reduce the amount of water required. Fly Ash and superplasticizer can also be introduced to reduce the amount of water needed. *Figure 3* can be referenced to gain a better understanding of this process.

* 1. **Superplasticizer**

This element of the mixture is added to reduce the amount of water needed for a strong, reliable compressive strength. *Figure 3* is a particularly interesting example to investigate this. I have plotted a dynamically weighted scatter plot of Superplasticizer vs Fly Ash, and it is very evident that just having this constituent in the mixture reduces water intake significantly without hurting the targeted strength of the compressive strength. *Figure 4* will further emphasize that even the strongest of mixtures can have as little water as approximately 150 .

**3.6 Coarse Aggregate**

Generally, a mixture of gravel and stones greater than 0.19” [3]

* 1. **Fine Aggregate**

A mixture of gravel and stone smaller than 0.19” [3]

**3.7 Age**

This is defined as the number of days the mixture is allowed to mature. As seen in the article [4] this is a highly debated issue worldwide and for certain strengths, a mixture is believed to have to be aged for an expected amount of time, yet it varies by country.

1. **Regression Analysis – Techniques, Algorithms, Key Insights, and Discussion of Results**

My data analysis consisted of more than one algorithm. I will go step by step, introducing different algorithms, comparing them to show why my predictor worked in some cases and failed in others. I am confident of the results because I introduced a cross validation technique (K-Fold Cross validation) towards the end of my analysis; ensuring that at some point, every single one of my data points was used in a testing and/or training set.

* 1. **Principal Component Analysis (PCA)**

A principal Component Analysis identifies what is the best summary of my dataset, all 7 components, with the fewest number of summary variables, in my case 4. Although a good prediction model does not entirely depend on a R2 score (the amount of variance explained by the model), it is a great indicator of how much of the data is represented by the Principal Component (PC). PCA’s are highly useful for reducing the size of the dataset by projecting all the data onto a plane/hyperplane. This can be done many times and planes are added in perpendicular until a satisfactory amount of variance is explained.

The data is only projected after all units have been removed. This is the part that will deal with any underlying bias and identify patterns in the data sometimes. The units are removed by a method of Scaling and Centering the data. This process is shown in Eq. 1 and Eq. 2.

… Eq. 1

… Eq. 2

Once this process is done, a PC is projected into a dimension. If the amount of variance explained is not sufficient, A plane that is perpendicular to the first PC is added. This process is done 4 times, but it was only able to explain approximately 74% of the data. This does not suggest that the prediction model will be bad, but it means that this does not look like the best approach for the dataset. The bar graph showing the PCA implementation is shown in *figure 6*. The Square Predicted Error (SPE) was also unusually high, and the Q2 (The amount of variance explained by the prediction of the model) was poor. This is not the right approach for the dataset.

* 1. **OLS (Ordinary Least Squares)**

OLS is a type of linear regression method for estimating unknown parameters in a linear regression model. [5] The coefficients are found by minimizing the error of prediction. This implementation was surprisingly good for my dataset. The inflated R2 value in *figure 5* does raise some concerns. Here I will explain what the OLS does to the model and explain why the model is overfit even though the R2 and adjusted R2 values are really high. It is perhaps easy to assume that there is nothing wrong with this prediction, but the inability of a PCA’s components to correctly justify the goodness of the data is a bit of a concern, so I needed to probe the model, test its abilities, and obtain further information from other types of prediction model to even consider this a good result.

The equation used by a model is:

… Eq. 3

Where the = the slope of the line, = data point, = the intercept, and = the error

The OLS also does not deal with the collinearity of the models. In this case, there is low co-linearity in the models, refer to *figure 8*. Collinearity deals with the statistical significance of the data. In a given dataset, the goal of a data analyst is always to first and foremost remove any bias, noise, and reduce the collinearity so a good regression prediction model can be made.

I will now prove overfitting of the OLS model. This is done using two further equations. The first equation will calculate the R2 and the second an additional predicted R2.

…Eq 4

Eq. 4 is the normal R2 value. This will indicate how well the data is represented by the linear regression, however, a fundamental mistake is to focus on attaining a high value, because it can stop being effective.

A picture containing object, clock, watch

Description automatically generated… Eq 5

This equation is highly relevant. This is the adjusted value. This equation has the ability to classify a good and bad model. It is of utmost importance to find the difference between the two for the concrete compressive strength dataset.

Overfitting is basically adding complexity to the model when it’s not supported by the data. For overfitting to be true, this value should be very close to 0%, and according to *figure 5* not only is the ‘goodness’ of my model corrupted, but in fact the adjusted value also predicts no change. This will eliminate any doubt of my claim that the model is overfit, and a prediction model needs to be built using a different approach.

* 1. **Linear Regression**

A linear regression is a very basic method of analyzing the data from a line drawn from the intercept. If there is anything that has been proved so far, it is the fact that the regression analysis is definitely non-linear for this dataset. To test this out, I ran a linear regression, and was not surprised to get moderate results. *Figure 9* reproduces the results that I obtained in python. As expected, it performed poorly. *Figure 12* is evidence that a linear model is not good enough. There is a 20% discrepancy between the original value and the predicted value. This means that the error does not lie within the threshold (+5% or -5%) , and hence can be assumed to be a bad prediction model.

* 1. **Partial Least Squares (PLS)**

A PLS is better than a PCA because, much like a PCA, it sequentially extracts each component. The objective of a PLS is to try its best to explain the X-Space, the Y-Space, and maximize the relationship between the two. An advantage of PLS is that no other algorithm offers a dimensionality reduction, by linearly extracting the most useful features without losing a lot of significant information. The emphasis of a PLS is in the prediction model itself and not the underlying data insights. [6] I used a PLS because it has wide applications in process control and very good for outlier detection. A PLS is a quick, efficient, and computationally advantageous method to predict and monitor the strength of concrete. In my attempt to apply a PLS regression, to ensure fair results and an equal chance of giving me good results, I first attempted (and failed) at identifying any outliers before I ran the algorithm.

While the outlier selection process did not work, I simply cross validated my results using a Training and Testing split dataset. In the most ideal case, there would be an entire dataset that would be dedicated to training, and one for testing. However, for the purpose of this project and due to a time (and coding comprehension) constraint, the dataset was split 80:20, with 80% of the data for training and 20% of data for testing respectively. This granted me a fair trial run.

I will now discuss the manner in which the PLS Regression was attempted. First the Training model was fit. Then the number of components decided. The next step in the process was to use a 10-fold cross validation to build and test the prediction model. The R2 score was calculated for the training model and the testing model, and to make sure those values could be justified, the mean squared error was also computed for both my models. The results, however, were disappointing. The testing model did almost 9% better than my training model. While this cannot be investigated further, I decided that while the MSE reduced, as expected for a higher R2 score, this model was still no good. Even after varying the number of cross validations – 2, 5, 7, and 10 again, as well as the training testing split – 50:50, 60:40, 70:30, the results still seemed to vary, but never really got any better and in some cases got worse!

* 1. **Decision Tree Regression**

A decision tree regression has many advantages and disadvantages, in my dataset, there is no real continuity in my data. For every mixture the strength changes, and my goal was to focus on how to build an accurate prediction model of identifying the strength of concrete. So, it made a lot of sense when I chose to try and approach the problem from a Decision Tree regression standpoint. First and foremost, a big advantage of the decision tree is that it is non-linear. The algorithm is completely unique in each implementation but is notoriously prone to being overfit. The overfitting occurs, very much like in a PCA, but instead of PC’s the tree just does not know when to stop splitting, one of its key deciding input factors. Very simply put, the DTR will create a lot of ‘what if’ statements, and then do the regression of the model to determine the best prediction model. That is because the results are pretty straightforward. If a % of regression is greater than a type of cement it is stronger. I built a function that does this regression in a very clear and concise way! The results are shown in *figure 11* and also the additional document where the evidence of the foundation of the algorithm is found. It summarizes, step by step, what each of the algorithms do, and how to do it.

* 1. KNN (K-Nearest Neighbors)

KNN is a very adjustable algorithm. It has multiple uses, but the advantage of it is that it can be used for both classification and regression. It uses feature similarity to group together similar strengths of concrete, this is relevant because there are many similar strengths of concrete at different compositions. This means that my KNN algorithm was able to categorize the different strengths and group them together and then apply a prediction and place the new data into the graph and identify strengths. While this was a good idea to regress the strength of concrete, this was not a very effective method. After a k-fold cross validation of 10, only 67% of the data was predicted correctly. This still means that the data was incorrectly predicted for almost 30% of data.

5.0 Summary

In a nutshell, the various methods used, all provide various insights of the data. Some models were good at prediction (DTR), some were good for describing the variables individually (PCA and PLS), some were amazing for detecting outliers (Linear Regression), and eventually, a successful prediction was identified by the Decision Tree algorithm. The data was split using testing and training data in modifications, they were also cross validated at the end, more specifically in *Figure 12*. While the dataset was not the best in terms of building a model, it introduced a lot of new ways to identify how each component will vary the output, even if it is ever so slightly. The greatest insights I learned from the dataset is that the target strength of cement is controlled using various mixtures, some of the cement standards require a specific number of days of aging, and that the predictive strength of concrete compressive strength can definitely be calculated, but a lot of care needs to be given in handling the inputs due to the sheer number of permutations of different mixtures of concrete.

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. Blast Furnace Slag. (n.d.). Retrieved from http://www.nationalslag.org/blast-furnace-slag.
3. (n.d.). Retrieved from https://www.cement.org/cement-concrete-applications/concrete-materials/aggregates.
4. elaty, M. abd allah A. (2013, October 12). Compressive strength prediction of Portland cement concrete with age using a new model. Retrieved December 12, 2019, from https://www.sciencedirect.com/science/article/pii/S1687404813000783#bi0005.
5. Hutcheson, G. D. (1999). Ordinary least-squares regression. In Hutcheson, G. D. *The multivariate social scientist* (pp. 56-113). : SAGE Publications, Ltd. doi: 10.4135/9780857028075
6. Tobias, R. (n.d.). *An Introduction to Partial Least Squares Regression*. Retrieved from https://stats.idre.ucla.edu/wp-content/uploads/2016/02/pls.pdf