# Machine Learning on Compressive Strength of Concrete

Matthew Walden Lua[1], Neil Matthew Lua[1], Kurt Bradley Tanting[1] and Richard Alvin Zapanta[1]

[1] De La Salle University, Manila, Philippines

**Abstract.** Concrete is an integral part of our environment today as it is widely used in buildings and structures, therefore studying its compressive strength is important as it can lower the casualty during disaster. Concrete compressive strength is a measurement to determine if the component mixture meet the requirement of the specification. The goal of the model is to predict the concrete compressive strength of cement given a set of values for the components of the cement mixture. The program used the dataset donated by Professor I-Cheng Yeh, with 1030 instances and 9 attributes, which composed of cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age and the compressive strength. The program performs ridge regression, a variation of linear regression with regularization applied in it. The data of the system was split into 80% train set and 20% test set, however, there was an extra step of cross validation and hyperparameter tuning to avoid overshooting the optimal spot when generating the regression graph and to minimal the bias and variance. The chosen model's performance on the test set was below than what was expected. However, performing k-fold cross validation gave a better representation of the actual performances of the model with an $r^2$ cross validated score of 0.621966 with the chosen $\lambda$ of 1402. The model's MAE and RMSE are 8.10 and 10.29 respectively. Thus, it may pose concerns to the model usability to predict the compressive strength of the concrete.

**Keywords:** Machine Learning, Ridge Regression, K-Fold Cross Validation, Compressive Strength

## 1. Introduction

Concrete is an integral part of our environment today, as it is a staple material in most if not all modern buildings and structures. Because of this, studying its strength and durability is very important, as this can spell the difference between a sturdy building and a man-made disaster. Strength tends to be easier to test than other properties as well [1]. As a result, this is the most common performance measure when building structures [2]. The specified strength, fc', is the benchmark used for testing concrete mixtures to determine whether they are fit for use in building. Multiple tests are performed on the same sample of concrete. The average of all tests must be at least fc', and no strength test should result in values more than 3.45MPa (megapascals) less than fc', or 10% of the fc' when the specified strength is at least 35MPa [3]. Concrete is generally a mixture of cement, water, and aggregate. Aggregate is inert, and comprises 60 - 80% of the volume and 70 - 85% of the weight of concrete [4]. These, among other components, are used to measure the strength of concrete. As mentioned previously, studying strength of concrete is very important, as it is integral to all modern structures. This can prevent severe damage when natural disasters such as earthquakes occur. The goal of this study is to determine the concrete compressive strength of cement given a set of values for the different components of its cement mixture. Over a sample of 1000 data sets was used, and was trained using multivariate linear regression, a machine learning method involving the prediction of the dependent variable's value with multiple features and their corresponding weights.

A model which could accurately predict the concrete compressive strength of a mixture given a combination of different inputs could help in determining the right amount of each component to be added to a cement mixture to maximize its compressive strength. This could help in reducing costs for the creation of cement mixtures by simply inputting test values instead of creating the mixture itself then testing for its concrete compressive strength.

## 2. Concrete Compressive Strength Dataset

The dataset used for this Machine Learning project is the Concrete Compressive Strength Dataset. It was retrieved from the Center for Machine Learning and Intelligent Systems of the University of California, Irvine. This dataset was donated by Professor I-Cheng Yeh from the Department of Information Management, Chung-Hua University of Hsin Chu, Taiwan on August 3, 2007 and was used in his study, "Modeling of strength of high performance concrete using artificial neural networks" [5]. The following Table 1 gives a summary statistic of the dataset.

**Table 1.** Summary Statistic of the Dataset.

| Features | Statistics |
|---|---|
| Number of instances (observations) | 1030 |
| Number of attributes | 9 |
| Attribute breakdown | 8 quantitative input variables, and 1 quantitative output variables |
| Missing attribute | None |

The 8 quantitative input variables are the different variables that affect the compressive strength of the concrete. These variables are cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, and age. While the 1 quantitative output variable is the compressive strength of the concrete from the different components of the input variables. More detailed information on the attributes can be found in Table 2.

**Table 2.** Information about the Attributes.

| Name | Data Type | Measurement | Description |
|---|---|---|---|
| Cement (component 1) | Quantitative | kg in a m^3 mixture | Input Variable |
| Blast Furnace Slag (component 2) | Quantitative | kg in a m^3 mixture | Input Variable |
| Fly Ash (component 3) | Quantitative | kg in a m^3 mixture | Input Variable |
| Water (component 4) | Quantitative | kg in a m^3 mixture | Input Variable |
| Superplasticizer (component 5) | Quantitative | kg in a m^3 mixture | Input Variable |
| Coarse Aggregate (component 6) | Quantitative | kg in a m^3 mixture | Input Variable |
| Fine Aggregate (component 7) | Quantitative | kg in a m^3 mixture | Input Variable |
| Age | Quantitative | Day (1-365) | Input Variable |
| Concrete Compressive Strength | Quantitative | MPa | Output Variable |

The compressive strength of concrete is the strength of hardened concrete measured by the compression test. It is also a measurement of the concrete's ability to resist loads that tend to compress it. It is measured by crushing cylindrical concrete specimens in the compression testing machine. The results are primarily used to determine if the concrete mixture meets the requirements of the specified strength, fc', in the job specification [3].

There are several components that affect the compressive strength of the concrete: 1) The water (component 4)/cement (component 1) ratio is the most important factor for gaining the strength of the concrete. The lower ratio water/cement ratio leads to higher strength of concrete, while the higher the water/cement ratio leads to segregation and voids in concrete. Thus, there is an inversely proportional relationship of water/cement ratio to the compressive strength of concrete [6]; 2) Blast-furnace slag (component 2) is a nonmetallic coproduct produced in the process of production of iron. A study made by Norrarat, Tangchirapat, Songpiriyakij, and Jaturapitakkul confirms that blast furnace slag contributes to the compressive strength at the later ages (28 days and beyond) [7]; 3) Fly Ash (component 3) is a product of coal-burning plants. It can replace 15%-30% of the cement in the mix. Cement and fly ash together in the same mix make up the total cementitious material [8]; 4) Superplasticizer (component 5) is a type of water reducer. The main difference between the superplasticizer and water reducers is that superplasticizer will significantly reduce the water required for concrete mixing. A study made by Alsadey found out that the use of superplasticizer as a chemical admixture on concrete has slightly increased in compressive strength than normal concrete [9]; 5) The ratio of coarse aggregate (component 6) and fine aggregate (component 7) also affects the compressive strength of the concrete, because if the proportion of the fine aggregate is increased in relation to the coarse aggregate, there will be an increase in the aggregate surface area, thus, the water demand will also increase [10]. If the water demand increases, the water/cement ratio will also be increased, thus the compressive strength of the concrete will be decreased; and 6) Lastly, the age of the concrete plays an important role in the compressive strength of the concrete. The degree of hydration is synonymous with the age of concrete provided the concrete has not been allowed to dry out or the temperature is too low. The hydration process is the chemical reaction between water and cement. It produces a gel, that plays a significant role in the bonding of the particles of the concrete ingredients. Therefore, the strength of the concrete increases as its age increased [6].

Given the variable name, variable type, measurement unit and a brief description. The concrete compressive strength is a regression problem and the order of this listing corresponds to the order of numerals along the rows of the dataset. The analysis of the dataset made by the Python can be seen in Table 3.

**Table 3.** Analysis on Dataset.

| | Cement (component 1)(kg in m^3 mixture) | Blast Furnace Slag (component 2)(kg in a m^3 mixture) | Fly Ash (component 3)(kg in a m^3 mixture) | Water (component 4)(kg in m^3 mixture) | Superplasticizer (component 5)(kg in a m^3 mixture) | Coarse Aggregate (component 6)(kg in a m^3 mixture) | Fine Aggregate (component 7)(kg in a m^3 mixture) | Age (day) | Concrete compressive strength (MPA, megapascals) |
|---|---|---|---|---|---|---|---|---|---|
| count | 1030.000000 | 1030.000000 | 1030.000000 | 1030.000000 | 1030.000000 | 1030.000000 | 1030.000000 | 1030.000000 | 1030.000000 |
| mean | 281.165631 | 73.895485 | 54.187136 | 181.566359 | 6.203112 | 972.918592 | 773.578883 | 45.662136 | 35.817836 |
| std | 104.507142 | 86.279104 | 63.996469 | 21.355567 | 5.973492 | 77.753818 | 80.175427 | 63.169912 | 16.705679 |
| min | 102.000000 | 0.000000 | 0.000000 | 121.750000 | 0.000000 | 801.000000 | 594.000000 | 1.000000 | 2.331808 |
| 25% | 192.375000 | 0.000000 | 0.000000 | 164.900000 | 0.000000 | 932.000000 | 730.950000 | 7.000000 | 23.707115 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 50% | 272.900000 | 22.000000 | 0.000000 | 185.000000 | 6.350000 | 968.000000 | 779.510000 | 28.000000 | 34.442774 |
| 75% | 350.000000 | 142.950000 | 118.270000 | 192.000000 | 10.160000 | 1029.400000 | 824.000000 | 56.000000 | 46.136286 |
| max | 540.000000 | 359.400000 | 200.100000 | 247.000000 | 32.200000 | 1145.000000 | 992.600000 | 365.000000 | 82.599225 |

## 3. Methodology

The researchers have chosen to apply a variation of linear regression to the dataset— specifically the ridge regression. Linear regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output) [11]. What sets ridge regression apart from the vanilla linear regression is that ridge regression has a kind of regularization applied to it. The regularization will penalize the sum of squared value of the weights (which is the only term in linear regression) so that the model would not overfit whenever there are multiple features involved already [12]. The loss function of ridge regression is as stated below:

$$L = \Sigma (\widehat{Y}_i - Y_i)^2 + \lambda \Sigma \beta^2 \qquad (1)$$

In equation 1, the loss function L is calculated by getting the sum of the difference of distances between the predicted and the actual truth, squared, and the $\lambda$ multiplied by the slope $\beta$ for each feature, squared. The $\lambda$ in this equation is used to "penalize" the loss function for the high values of coefficients $\beta$. By doing so, instances of multicollinearity can be addressed so that it will not throw off inaccurate estimates and inflate the standard errors that will lead to giving false and nonsignificant values [13].

The process of applying ridge regression onto the dataset chosen follows certain steps as most machine learning models would normally do. The process is summarized and compressed into a pipeline that was made using RapidMiner. The following figures below are the parts of the pipeline showing the process of deriving the optimized predictions.

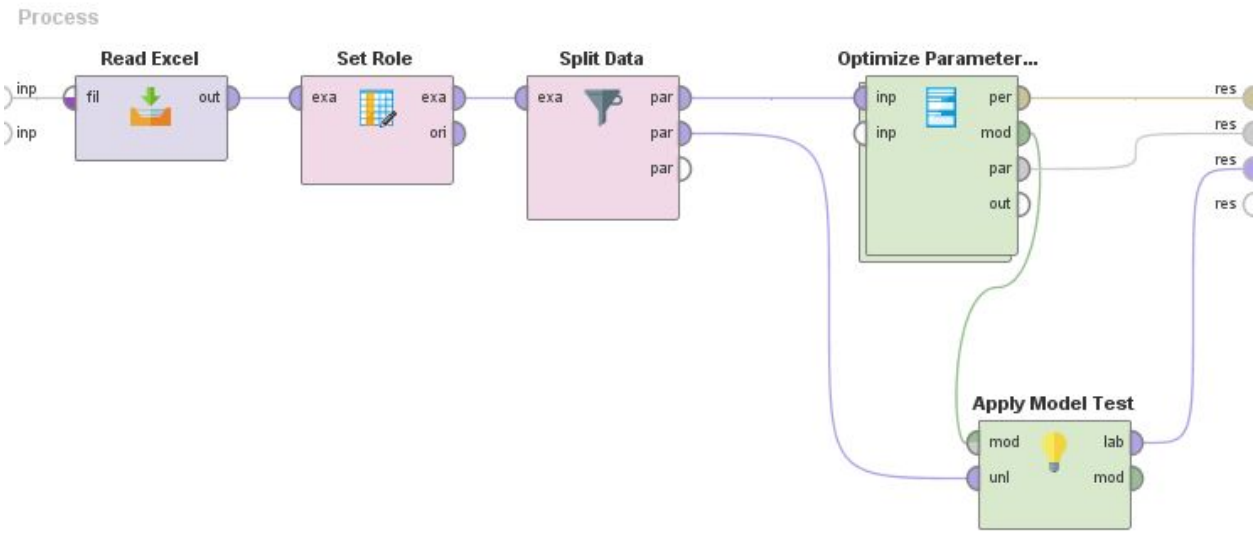


**Fig. 1.** Overview of the pipeline model of the ridge regression process.

First of all, the excel file containing the dataset was read by the machine and the details were tabulated into the system. This was done first so that the system has a copy of the dataset that is ready for processing. Now that the system is able to manipulate the dataset, the next process that the researchers did is setting the roles of the features. In this part, the researchers identify which feature would be the dependent one, and which would be the independent ones. The goal of the system here is predicting the value of comprehensive strength of the concrete, hence that is the dependent variable. One thing to note here is that all of the features in the dataset chosen have their relevance in determining the compressive strength of the concrete, thus, no feature was filtered out. The data is now to be split into a train set and a test set. The purpose of this is so that the model would still have that uncertainty when testing on the test set since it is data that the model has not yet seen before in the train set. The researchers have decided to allot 80% of the data for the model to train with, leaving 20% for the model to test with. In addition to that, that 80% alloted will also be used for hyperparameter tuning that is to be discussed later. The next process in a vanilla machine learning system would be going straight to train and test, but to make things more comprehensive, the researchers have applied cross validation as well as hyperparameter tuning to the dataset with roles assigned. This process is explained more in-depth in the figure below.

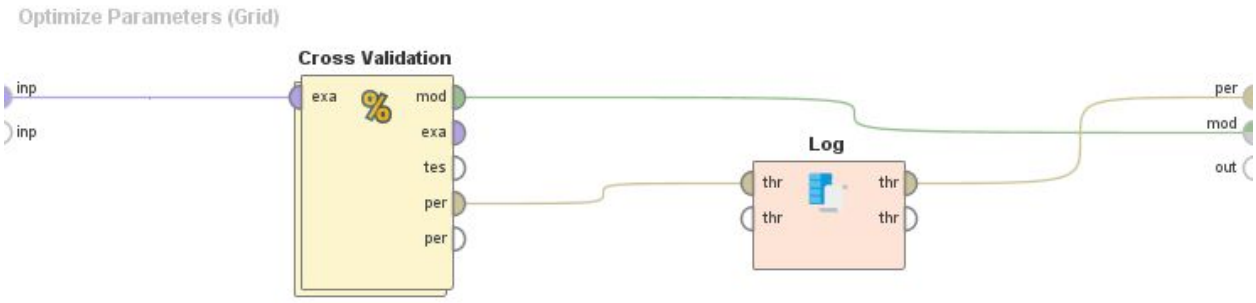


**Fig. 2.** Inside the optimizing parameter (grid) module.

Hyperparameter tuning was done because in order for the system to predict the outcomes as accurate as possible, the researchers must carefully tweak the bias λ so that it will not overshoot the optimal spot when generating the regression graph. By doing so, the relatively irrelevant features' impact will be minimized when it comes to cross validation [14]. With that in mind, the next process is choosing the best hyperparameter tuning method for this scenario.

The hyperparameter tuning method chosen was the grid search. This is because grid search handles datasets with small sample sizes well due to its exhaustive search nature. This makes sure that all possible combinations will be tested and the best λ will be determined by the search [15]. In addition to that, the researchers have decided to use $r^2$ as the scoring basis. This is because $r^2$ is a statistical measure to knowing how well the data is fitted in the regression line. Having a higher $r^2$ means that the variation is lower and the data is closer to the regression line, which also means that the fitted values are closer to the observed actual values [16]. Looking inside the optimizing parameter grid module, there are two modules nested within namely: cross validation and log. In grid search, cross validation is always used to score the trained model predictions. Then, scores are compared with one another. On the other hand, the log module simply takes logs of the cross validation's performances and produces the summarized results for output. The cross validation will be explained in detail after the figure below.

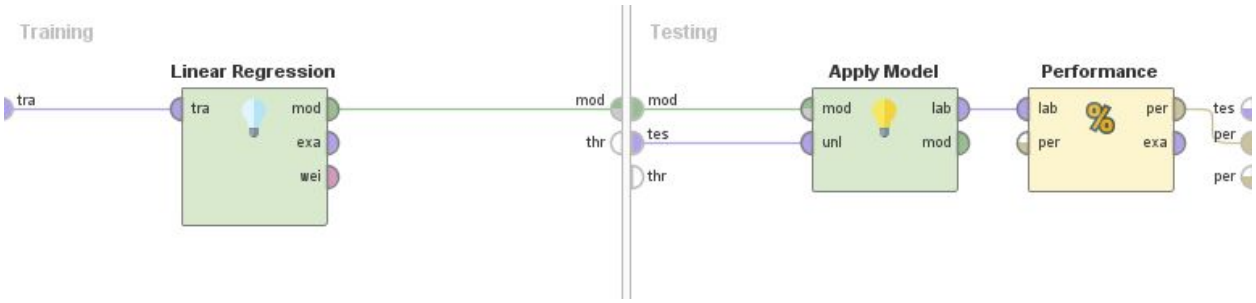


**Fig. 3.** Inside the cross validation module.

The goal of the cross validation is to obtain the average performance of the model and to make sure that the model has minimal bias and variance [17]. The cross validation method used is the popular k-fold cross validation, having k = 10. This means that the data is split into 10 folds, and 1 out of 10 of those folds is set as the test set while the rest are the train sets. 10 folds is the common amount of folds done because it's a value that has been proven to generally estimate well with minimal bias and just the right amount of variance [17][18]. In cross validation, the training set is accepted by the ridge linear regression module and the model produced by it will be accepted by the apply model

module which is the testing part of cross validation. After producing the labels, it will pass it over to the performance module that will calculate for the root mean squared error as well as the $r^2$ correlation which is to be passed on for output.

## 4. Results and Analysis

The primary measure used for hyperparameter tuning was the $r^2$ scoring. Hyperparameter tuning was done on different values for $\lambda$ which ranged from 0 up to 1500. Initially, testing was done on smaller values with ranges from 0 to 100. However, there was a trend which saw the $r^2$ score increase to a certain number. Thus, the researchers procedurally incremented the possible values for $\lambda$ until the value began going down. The graph showing the relationship of the chosen $\lambda$ and the $r^2$ values can be seen below:

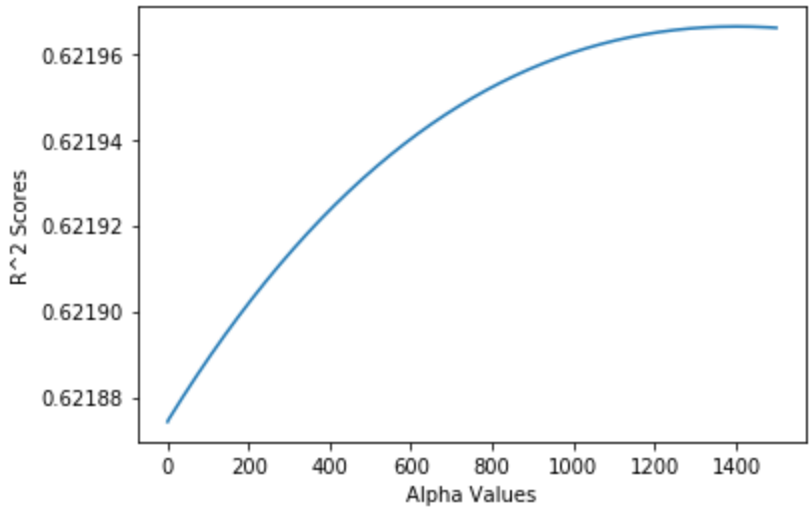


**Fig. 4.** Relationship between $\lambda$ and $r^2$ scores.

The value of $r^2$ peaks at $\lambda$ with the value of 1402. Afterwards, the values of $r^2$ begin to decrease. Thus, the chosen value for $\lambda$ was 1402 whose $r^2$ value was 0.621966. The value of $\lambda$ penalizes increases the size of the loss function which means insignificant features are pushed closer towards 0 and this could be the reason as to why the $r^2$ is increasing as the value for $\lambda$ is increasing. However, once it reaches a certain point, the coefficients might become penalized too much that the model begins to become more inaccurate in predicting the values since the coefficients may be becoming too small. The value of r2 with the chosen $\lambda$ isn't actually that higher than the value for other $\lambda$ with it only being around 0.008 higher than value if the $\lambda$ was at zero. The values for the final coefficients can be seen below:

**Table 4.** Final Coefficient Values for the Features.

| Attribute (Input Variables) | Coefficient |
|---|---|
| Cement (component 1)(kg in a m^3 mixture) | 0.124373 |
| Blast Furnace Slag (component 2)(kg in a m^3 mixture) | 0.107789 |
| Fly Ash (component 3)(kg in a m^3 mixture) | 0.092347 |
| Water (component 4)(kg in a m^3 mixture) | -0.145738 |

| | |
|---|---|
| Superplasticizer (component 5)(kg in a m^3 mixture) | 0.318862 |
| Coarse Aggregate (component 6)(kg in a m^3 mixture) | 0.015910 |
| Fine Aggregate (component 7)(kg in a m^3 mixture) | 0.019428 |
| Age (day) | 0.120048 |

After training the model with the chosen hyperparameter, the model was used to predict the values of Y (which in this case is the concrete compressive strength). These values were then plotted against the actual values of Y given the feature data from the test portion from the train test split performed. The actual values which was plotted along the X-axis and the predicted values for compressive strength was plotted along the Y-axis. The graph of the two values display some form of linear relationship between the two. However, as the values begin increasing, the values also begin being more dispersed. This could mean that the model is more accurate in predicting the values of smaller values of compressive strength. The graph of which can be seen below:

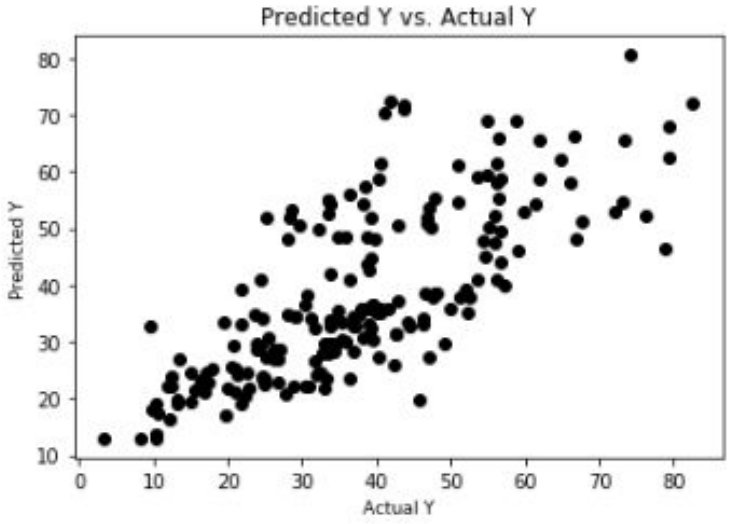


**Fig. 5.** Relationship between values of predicted Y and actual Y

The metrics then used on the final model include $r^2$ scoring, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). These are the common metrics used in analyzing regression models. The $r^2$ score helps in determining how well the model fits the observations, while the different errors give us an idea of how "off" the model was at predicting the values for the concrete compressive strength with the actual values of the output variable.

**Table 5.** Performance Metrics of Chosen Model on Test Data and Through K-Fold Cross Validation.

| Performance Metric | Test Data | Cross-Validated Scores |
|---|---|---|
| $r^2$ score | 0.5225109155787108 | 0.6219662523945074 |
| Mean Absolute Error | 8.816928643918235 | 8.095735274045733 |
| Mean Squared Error | 124.4796243599265 | 105.87993929069258 |

| | | |
|---|---|---|
| Root Mean Squared Error | 11.157043710585995 | 10.289797825549956 |

The performance of the model on the testing data is actually worse than what was predicted based on the tests done in hyperparameter tuning. However, through cross-validation using k-fold cross validation where k was set to 10, the performance seen is now closer to what was estimated during hyperparameter tuning.

Interpreting the cross-validated scores, the MAE and RMSE show generally how far off the model was at predicting the values for compression strength with the actual values [19]. The RMSE is more sensitive to outliers which makes sense as to why it has a higher value than the MAE however the difference between the RMSE and MAE is not that large. However, the values of 8.10 and 10.29 are large when thinking about the mean value of the output variable being at 35.81. In comparison, the average error would be around +/- 22% of the actual value.

Prospective users of the model may use it to get a very rough estimate of what the compressive strength of their concrete would be when supplied with the given features. However, users may may be discouraged to use the model since the error range between the predicted value and the actual value is quite large.

## 5. Conclusions and Recommendations

In summary, a ridge regression model was created with 80% of the data used as the training set and 20% used as the testing set. Hyperparameter tuning was then performed on the training portion of the set. The chosen value for $\lambda$ was 1402 which gave the best $r^2$ cross-validated score of 0.621966. The chosen model's performance on the test set was below than what was expected. However, performing k-fold cross validation gives a better representation of the actual performance of the model with an $r^2$ score of 0.621966. However even with hyperparameter tuning, the model's MAE and RMSE which are 8.10 and 10.29 respectively may pose a concern to the model's usability in predicting the compressive strength of cement. Therefore, the researchers recommend either increasing the number of data inputs to the training model in the hopes that it will help in reducing the margin of errors in the predictions. However, if even this does not improve the performance then the researchers would recommend implementing a different model in the case that the features and the output variable exhibit nonlinear relationships or if the features exhibit multicollinearity with one another. The research could also be expanded to take into account other possible components to add into the cement mixture, making it more versatile in modelling different cement mixture combinations.

## 6. Contributions

Richard is the one who researched for the dataset that was used in the study. After finalizing the domain and the dataset, Neil, Matthew and Kurt are the members who coded the Machine Learning program using Anaconda. All of the members contributed in the paper as Matthew, Kurt and Richard are the one who wrote the documentation; while Neil is the one who made the illustrations. Summary of the contributions of each member can be found at Table 6 of Appendix A.

## 7. References


1. Nemati, K. (n.d.). Strength of Concrete. Retrieved from http://courses.washington.edu/cm425/strength.pdf.
2. Testing the Compressive Strength of Concrete -- What, why, & how? (n.d.). Retrieved from https://www.nevadareadymix.com/concrete-tips/testing-the-compressive-strength-of-concrete/.
3. Jamal, H. (2017, January 29). Compressive Strength of Concrete after 7 and 28 days. Retrieved from https://www.aboutcivil.org/compressive-strength-of-concrete.html.
4. The Effect of Aggregate Properties on Concrete. (n.d.). Retrieved from https://www.engr.psu.edu/ce/courses/ce584/concrete/library/materials/aggregate/aggregatesmain.htm.

5. 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).
6. Patel, H. (n.d.). 11 Factors that can Affect the Strength of Concrete. Retrieved from https://gharpedia.com/blog/factors-that-affect-strength-of-concrete/.
7. Norrarat, P., Tangchirapat, W., Songpiriyakij, S., & Jaturapitakkul, C. (2019). Evaluation of Strengths from Cement Hydration and Slag Reaction of Mortars Containing High Volume of Ground River Sand and GGBF Slag. *Advances in Civil Engineering*, *2019*, 1–12. doi: 10.1155/2019/4892015
8. What is concrete - Concrete Defined. (2019, November 25). Retrieved from https://www.concretenetwork.com/concrete.html.
9. Alsadey, S. (2015). Effect of Superplasticizer on Fresh and Hardened Properties of Concrete. *Journal of Agricultural Science and Engineering*, *1*(2), 70–74. Retrieved from https://pdfs.semanticscholar.org/ce67/7dfb9cf993bd69132b896d11d3b3fce7d49a.pdf
10. Mishra, G. (2017, November 29). Factors Affecting Strength of Concrete. Retrieved from https://theconstructor.org/concrete/factors-affecting-strength-of-concrete/6220/.
11. Gandhi, R. (2018, May 27). Introduction to Machine Learning Algorithms: Linear Regression. Retrieved from https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a.
12. Kim, K. (2019, January 3). Ridge Regression for Better Usage. Retrieved from https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db.
13. Hintze, J. (2007). Ridge Regression. In *User's Guide III: Regression and Curve Fitting*(pp. 335–1-335–21). Kaysville, Utah: NCSS.
14. Chakon, O. (2017, August 3). Practical machine learning: Ridge regression vs. Lasso. Retrieved from https://codingstartups.com/practical-machine-learning-ridge-regression-vs-lasso/.
15. Worcester, P. (2019, June 6). A Comparison of Grid Search and Randomized Search Using Scikit Learn. Retrieved from https://blog.usejournal.com/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85.
16. Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit? (2013, May 30). Retrieved from https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit.
17. Brownlee, J. (2019, August 8). A Gentle Introduction to k-fold Cross-Validation. Retrieved from https://machinelearningmastery.com/k-fold-cross-validation/.
18. Gupta, P. (2017, June 6). Cross-Validation in Machine Learning. Retrieved from https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f.
19. Guanga, A. (2019, January 2). Understand Regression Performance Metrics. Retrieved from https://becominghuman.ai/understand-regression-performance-metrics-bdb0e7fcc1b3.

## 8. Appendix A. Contributions of Members

**Table 6.** Contribution of Members.

| Member | Contribution |
|---|---|
| Lua, Matthew Walden | Code, Documentation |
| Lua, Neil Matthew | Code, Illustrations |
| Tanting, Kurt Bradley | Code, Documentation |
| Zapanta, Richard Alvin | Documentation, Research |