EAEE 4000 Machine Learning for Environmental Engineering and Sciences

Prediction of Concrete Compressive Strength with Machine Learning

Course Project

Submitted by:

Diandian Zhao (dz2442)

Faculty Instructor:

Prof. Pierre Gentine

December 24th, 2021

Columbia University in the City of New York

New York, New York

1. Introduction

Concrete is the second most used material worldwide, just after water. Each year, approximately 30 billion tonnes of concrete are being produced [1] and this number is expected to increase due to the ongoing development of infrastructure in the third world countries. Because of such enormous volume, the production of cement and concrete is estimated to account for ~5% to 8% of the global anthropogenic CO_2 emissions [2]. Therefore, how to develop low-carbon concrete has been the major task for the construction industry in order to curb the global warming.

Concrete is a composite material made of cement, water, sand and gravel. Depending on the construction methods and service environments, it is also very common to add chemical and mineral admixtures to adjust the properties of concrete to meet specific requirements. Among these ingredients, cement is the one associated with the highest CO_2 emissions and accounts for over 95% of the total emissions from concrete production. Portland cement, a calcium siliceous material, is manufactured through heating of limestone and clays at over 1600 °C. The CO_2 emissions from this process can be threefold: (i) the decomposition of limestone ($CaCO_3$) at high temperature generating CO_2 (~50%), (ii) the fossil fuels burned to heat the raw materials into cement clinkers (~35%), and (iii) indirect sources from the electricity used for grinding, packaging, and transport (~15%) [3]. Thus, the most effective and practical strategy to reduce CO_2 emissions associated with concrete is to reduce the cement content.

Industrial waste or byproducts, such as fly ash from coal firing power plant, blast furnace slag from the steel manufacturing industry, and silica fume from the silicon metal factories, have been successfully used as supplementary cementitious materials (SCMs) to partially replace cement in concrete. Through such substitutions, the CO_2 emissions of concrete can be reduced while the properties of concrete, such as compressive strength or durability, may also be benefited due to the synergy between cement and SCMs. However, the use of more ingredients in concrete makes the design of the concrete mixture even more complicated.

Mix design, referring to the proportioning of various components to optimize the performance of concrete (mainly compressive strength), is usually achieved via an empirical method. The relationships between different parameters with the resulting compressive strength have been experimentally measured. By using these pair relationships, different parameters can be

determined step by step to calculate the final mix design according to the American Concrete Institute (ACI) 211 Guide [4], as shown in Fig. 1.

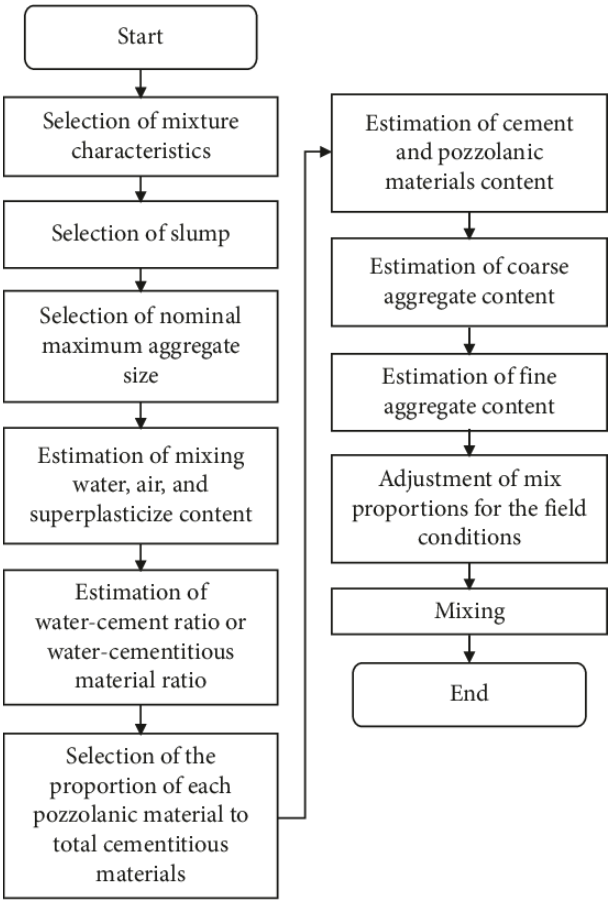

Fig. 1. Mix design approach based on American Concrete Institute (ACI) method.

The ACI mix design method can give more or less satisfying result, but it requires several trial-and-error experimental batches to verify the expected compressive strength and adjust the initial mix design accordingly. As such, this empirical approach cannot provide accurate outcomes and can be costly due to the labor-intensive trial-and-error process. The advent of machine learning can potentially provide an alternative solution to this problem. As up to date, the accurate mechanisms of how each ingredient governs the final compressive strength are not fully understood, the use of machine learning can be a good fit since it can circumvent such knowledge gap. The main goal of this project, therefore, is to explore the potential of using machine learning including neural network, decision tree, random forest and XGBoost to predict the compressive strength of concrete based on the initial mix design.

2. Data Collection and Exploratory Analysis

The data used in this research were collected from [5]. It consists of 1030 examples with 8 features (Cement, Slag, Fly ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, and Age) and 1 label (Compressive Strength).

For initial exploration, a heatmap and pair plots were constructed, as shown in Figs. 2 and 3, respectively. The heatmap represents the correlations between features quantitatively, while the pair plots give us a visual representation of the relationship between a pair of features. Based on the heat map, it can be seen that Cement (0.5), Superplasticizer (0.37), and Age (0.33) have positive correlations with Compressive Strength, whereas Water (-0.29) has a negative correlation with Compressive Strength. Other features do not seem to have a strong influence on Compressive Strength.

Superplasticizer (-0.66) and Fine Aggregate (-0.45) have negative correlations with Water, as superplasticizers are known to improve the fluidity of concrete and reduce the amount of water needed, and fine aggregates may reduce the water demand due to its spherical shape that can reduce friction.

Fly Ash, Slag, and Cement have negative correlations with each other. This is due to the fact that fly ash and slag are usually used to partially replace cement, and they are competitive with each other. All of them eventually act together as the binder in concrete and are mainly responsible for the development of the strength of concrete.

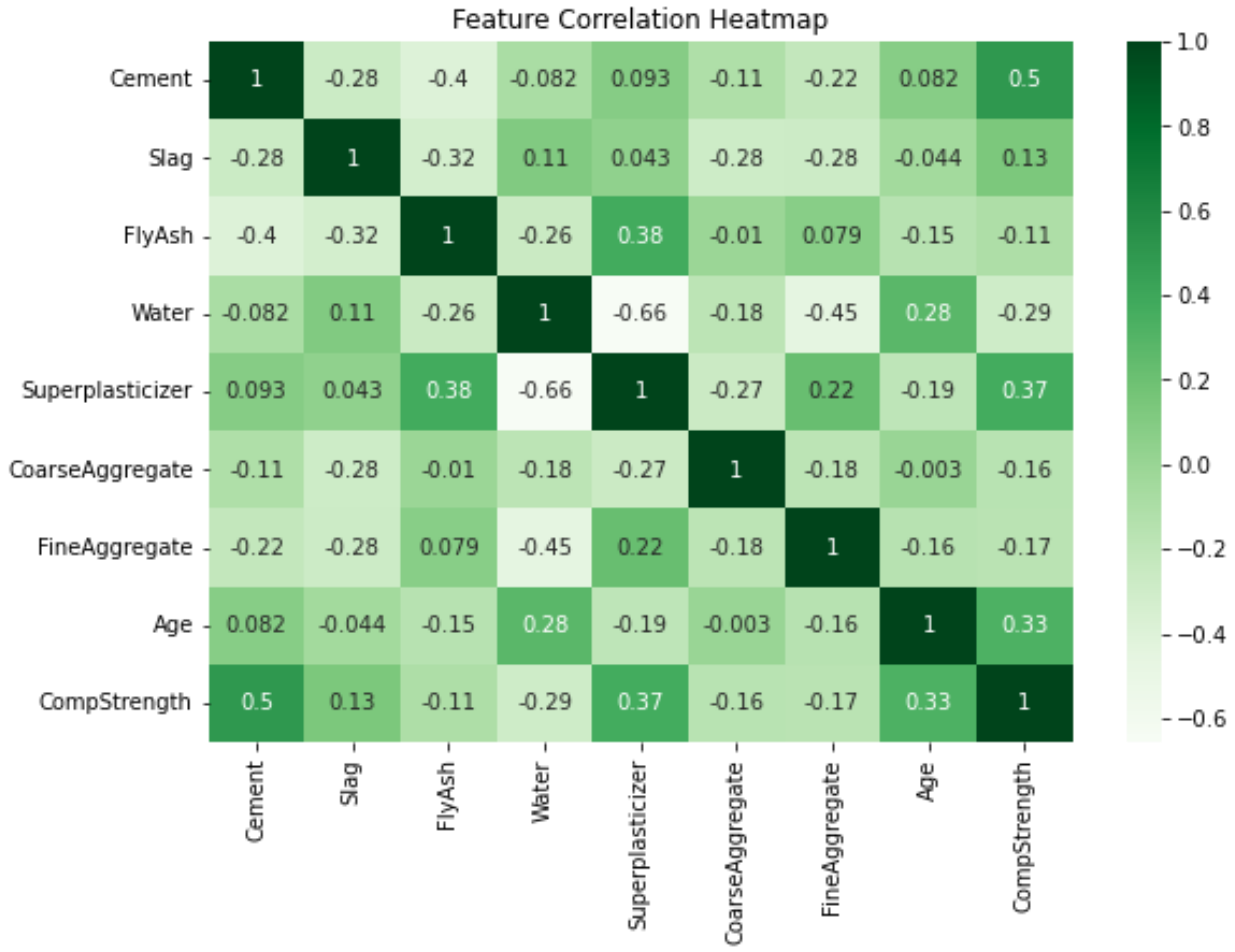

Fig. 2. Heatmap showing the correlations between different pairs of features.

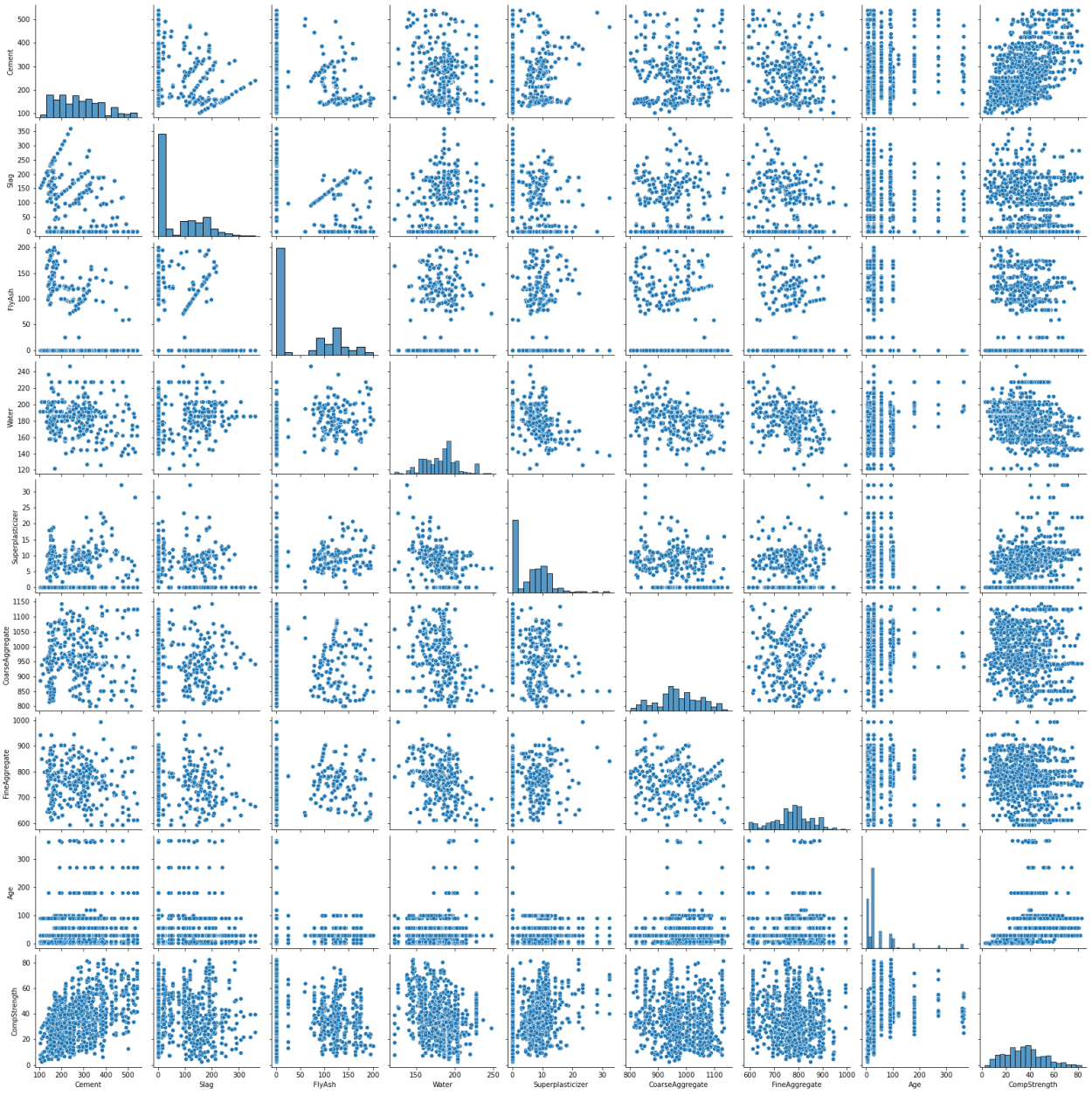

Fig. 3. Pair plots showing the visual representations of the relations between different pairs of features.

3. Data Pipeline

The datapoints were first separated into features and label. Before model building and training, the collected data were split into a training set and a test set. The test size was set as 0.2, i.e. 80% of the data will be used for training while 20% will be used for tests. The features then were normalized using StandardScaler to enable efficient training of machine learning models.

4. Linear Regression

The linear regression model was built using Keras. The model was configured with 8 input variables (features) and 1 output variable, i.e. Compressive Strength. Gradient descent optimizer with a learning rate of 0.01 to minimize the mean squared error. The model was trained using the entire training dataset as one batch size for 500 epochs. The loss was plotted to visualize the training process, as shown in Fig. 4. An early stop function with patience=20 was set to avoid overfitting. The actual epochs run before stop were 89 epochs.

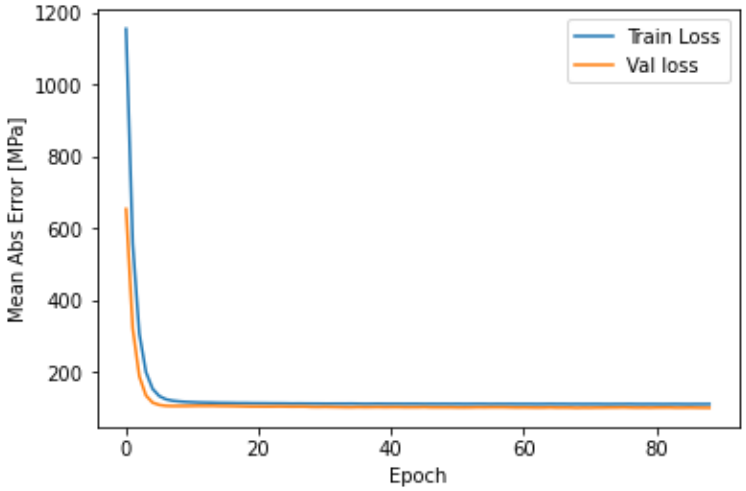

Fig. 4. The evaluation of loss upon the increasing epochs during linear regression model training with an early stop function to avoid overfitting.

After training, the linear regression model was used to predict the compressive strength in the test set. The results of predictions versus true values are shown in Fig. 5 and the numerical indicators are shown in Table 1. The linear regression model cannot provide satisfying predicted results, with

a R^2 value of only 0.56. Therefore, other machine learning models are tested in the following sections in order to improve the predicted results.

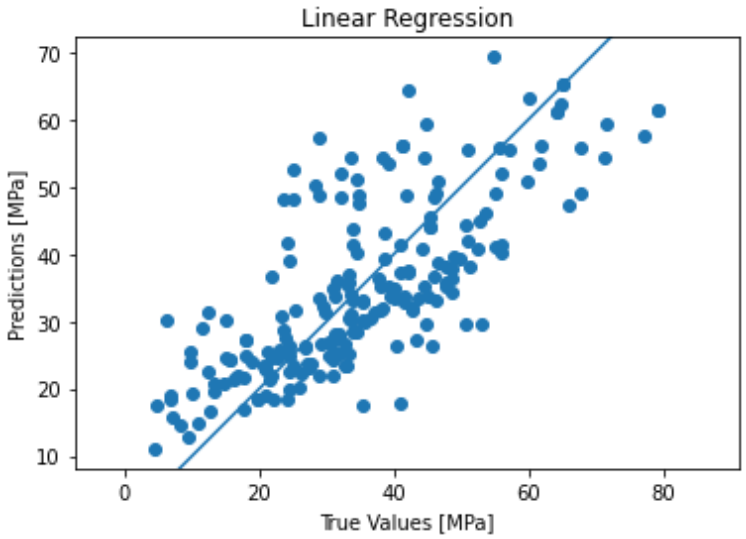

Fig.5. Scatter plot showing the predicted values versus the true values using linear regression model.

Table 1. Numerical metrics of the fitting results using linear regression model

Model	RMSE	MSE	MAE	R^2
Linear Regression	10.37	107.56	8.26	0.56

5. Neural Network

A neural network model was constructed using Keras. The model consists of 3 hidden layers with 128 neurons for each layer. Rectified Linear Unit (ReLU) was used as the activation function. The parameters of the model are summarized in Table 2.

Table 2. Summary of the neural network model

Layer (type)	Output Shape	Parameter Number
Hidden layer 1	(None, 128)	1152
Hidden layer 2	(None, 128)	16512

Hidden layer 3	(None, 128)	16512
Output layer	(None, 1)	129

Similar to the linear regression model, an early stop function was used to avoid overfitting. The patience number was set to be 20 and the maximum epoch was 500. The actual model was run for around 107 epochs before stop. The history of the loss during the training process is shown in Fig. 6.

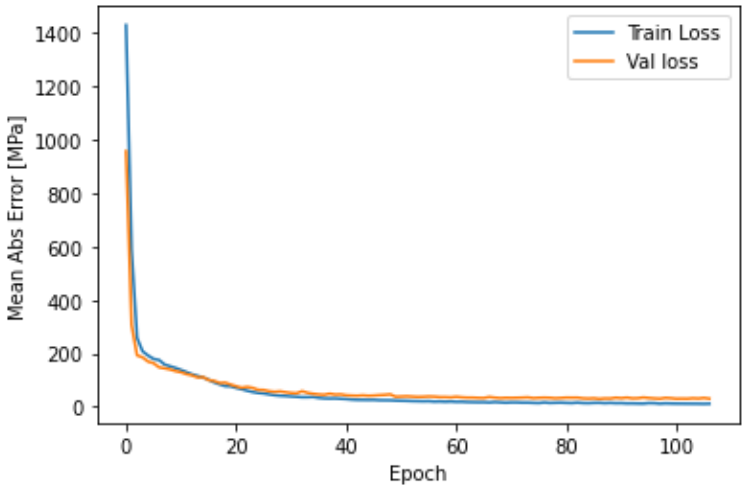

Fig.6. The evaluation of loss upon the increasing epochs during neural network model training with an early stop function to avoid overfitting.

Once the training process is completed, the neural network was used to predict the compressive strength using the test dataset. The comparison between predictions and true values is shown in Fig. 7, and the numerical metrics are shown in Table 3.

The predicted results are significantly improved compared to the linear regression model, and the R^2 value increases to 0.88. These results show the feasibility of using neural network to predict the compressive strength of concrete with the input mix design. Moreover, as the current input features only contain the quantities of each ingredient of concrete, there is still a lot of room for improvement in the future. For example, the compositions of ingredients such as cement and aggregates are highly variable, as concrete producers heavily rely on local materials to reduce the cost of transportation. If more experimental data can be collected across different locations in a

country or even worldwide, more features will be available to train the machine learning model, which in turn will become a more universal model that can be applied in different areas.

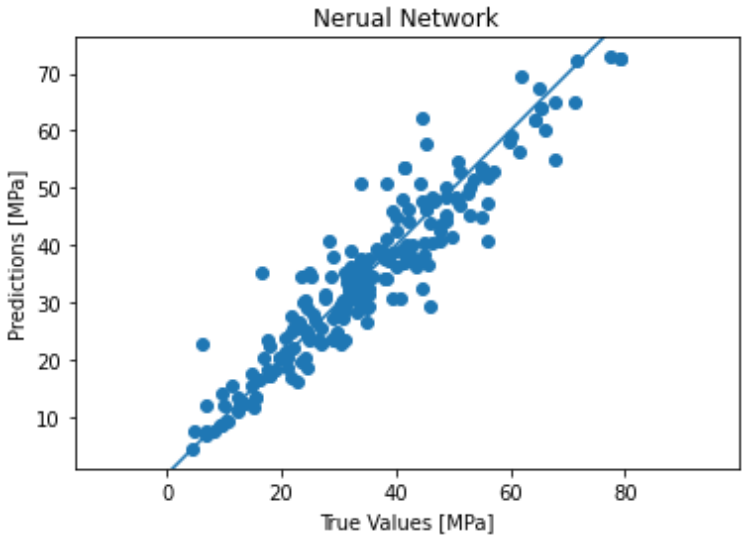

Fig.7. Scatter plot showing the predicted values versus the true values using neural network model.

Table 3. Numerical metrics of the fitting results using neural network model

Model	RMSE	MSE	MAE	R^2
Linear Regression	10.37	107.56	8.26	0.56
Neural Network	5.42	29.38	3.99	0.88

6. Tree-based models

In addition to the linear regression and non-linear neural network models, tree-based models are also very effective and can provide promising solutions to regression problems. Three tree-based models including decision tree, random forest and XGBoost are evaluated in this study and their performances are compared with the neural network model.

6.1. Decision Tree

Although decision tree algorithm is usually used for classification problems, it can also be very effective in solving regression problems. As the name implies, decision tree model is in the form of a tree structure with decision nodes to test the variables. The dataset will be broken down into

smaller subsets progressively, and these branches will incrementally develop, forming a complete tree after certain depth.

The decision tree regressor was used to train the model, and after training, the model was used to predict the compressive strength. The results of the predicted values versus true values are shown in Fig. 8, and the numerical metrics of this model are also compared with those of linear regression and neural network models. The decision tree did not yield results (R^2 = 0.78) as good as the neural network model (R^2 = 0.88), but it still greatly outperforms the linear regression model (R^2 = 0.56).

Fig. 8. Scatter plot showing the predicted values versus the true values using decision tree regressor.

Table 4. Numerical metrics of the fitting results using decision tree regressor model

Model	RMSE	MSE	MAE	R^2
Linear Regression	10.37	107.56	8.26	0.56
Neural Network	5.42	29.38	3.99	0.88
Decision Tree Regressor	7.32	53.57	4.50	0.78

6.2. Random Forest

The performance of the single decision tree model can be further improved by combing more trees into the model. Thus, a random forest regressor was used for data training. The maximum depth was set as 10. Larger numbers like 20 tend to overfit the model and reduce the overall model performance and therefore is not favorable. The results predicted by the random forest model versus the true values are plotted in Fig. 9 and numerical metrics are given in Table 5. It can be seen that the use of random forest with more trees significantly improves the prediction performance, with a R^2 score increasing from 0.78 to 0.89. The performance of this model is similar to that of neural network model (R^2 = 0.88).

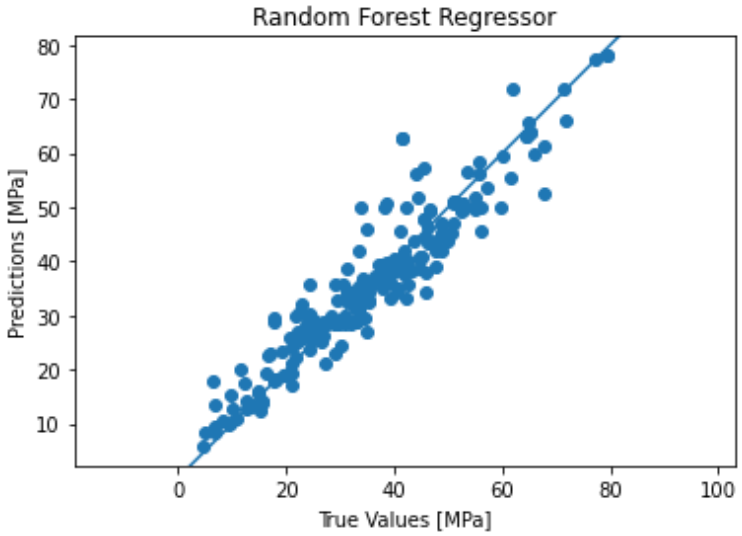

Fig. 9. Scatter plot showing the predicted values versus the true values using random forest regressor.

Table 5. Numerical metrics of the fitting results using random forest regressor model

Model	RMSE	MSE	MAE	R^2
Linear Regression	10.37	107.56	8.26	0.56
Neural Network	5.42	29.38	3.99	0.88
Decision Tree Regressor	7.32	53.57	4.50	0.78
Random Forest Regressor	5.26	27.68	3.75	0.89

6.3. XGBoost

Another algorithm that can be used to improve the decision tree model is XGBoost. This model optimizes the gradient through parallel processing and tree pruning. Similar to the random forest, to avoid overfitting, the maximum depth was also set as 10. The results of the compressive strength predicted by the XGBoost against the true values are shown in Fig. 10 and numerical indicators are presented in Table 6. The XGBoost model (R^2 = 0.92) generally outperforms the random forest model (R^2 = 0.89), and it is also the best model among all the models studied in this project.

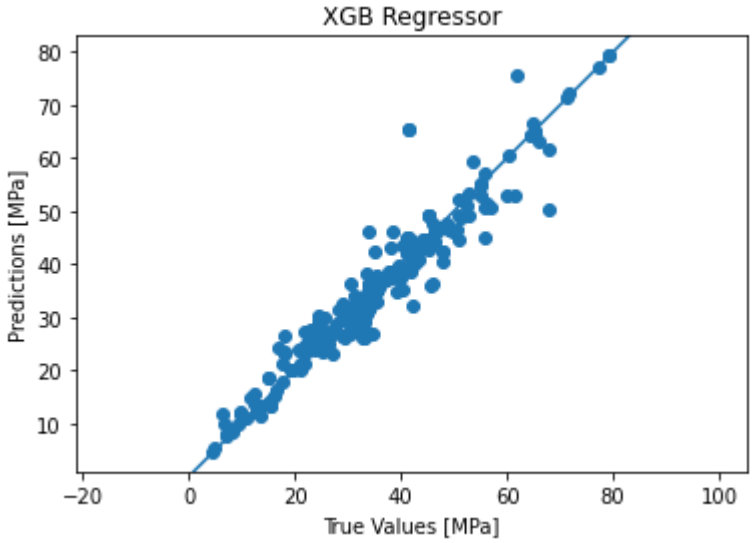

Fig. 10. Scatter plot showing the predicted values versus the true values using XGB regressor.

Table 6. Numerical metrics of the fitting results using XGB regressor model

Model	RMSE	MSE	MAE	R^2
Linear Regression	10.37	107.56	8.26	0.56
Neural Network	5.42	29.38	3.99	0.88
Decision Tree Regressor	7.32	53.57	4.50	0.78
Random Forest Regressor	5.26	27.68	3.75	0.89
XGB regressor	4.43	19.67	2.81	0.92

6.4. Feature Importance

To understand the training processes of the tree-based models, the feature importance can be plotted and compared, as shown in Fig. 11. Overall, the three most important features for the tree-based models are the same: Cement, Age, and Water. The major difference is that XGBoost model emphasize less on Cement but more on Age and Superplasticizer.

The feature importance derived from this machine learning project agree with the results obtained from experimental studies. Less water and more cement (i.e. a low water-to-cement ratio) are always favored in developing the compressive strength of concrete [6].

The underlying mechanism is that compressive strength development of concrete is highly dependent on the formation of material microstructure thorough cement hydration. Cement hydration refers to a series of chemical processes occurring between cement and water to form solid phases, which gradually fill the empty space to form a microstructure. The physical bonding of these solid phases gives the concrete strength to resist external pressure. This mechanism is illustrated in Fig. 12. A higher water-to-cement ratio will leave more empty space that is not filled by the solid hydrated phases, thus weakening the compressive strength of concrete.

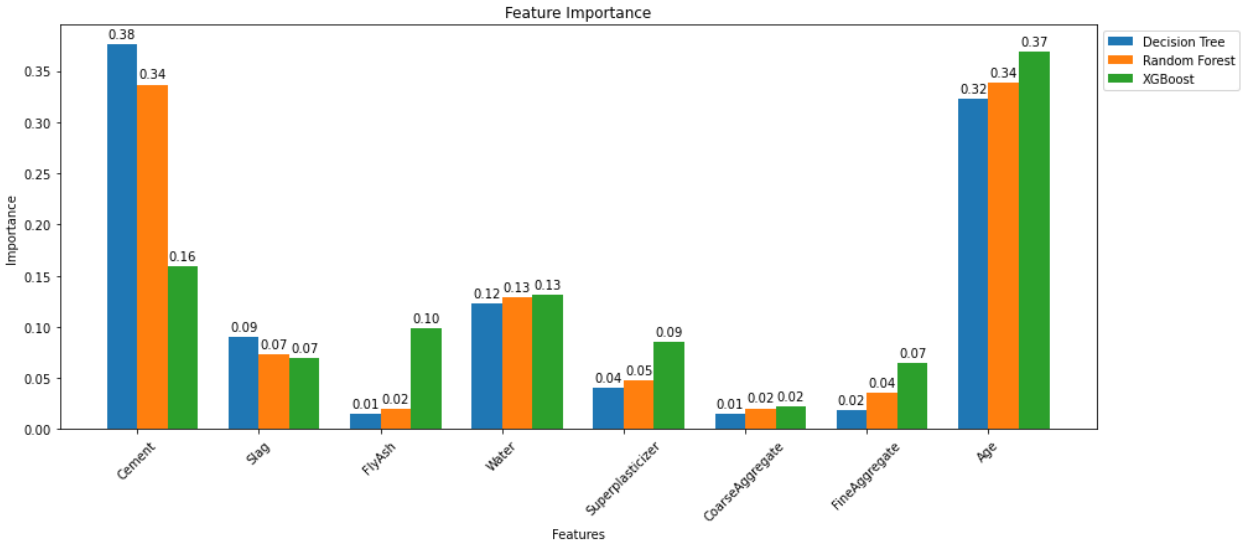

Fig. 11. Feature importance of decision tree, random forest, and XGBoost regression models.

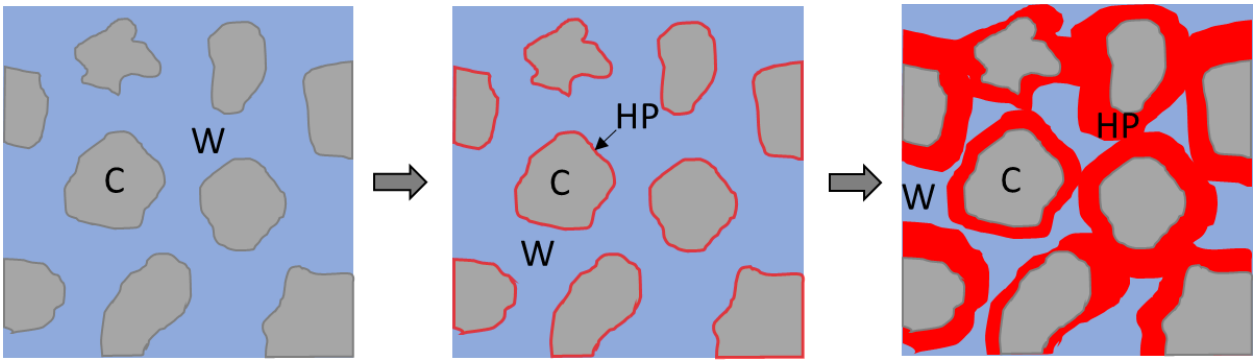

Fig. 12. Schematic representation of the cement hydration process and the development of microstructure. Water-to-cement ratio plays an important role as any water left unreacted will form pores that weaken the mechanical properties. C: cement; W: water; HP: hydration products.

7. Conclusions

This study explores the feasibility of using various machine learning to predict the compressive strength of concrete based on their initial mix designs. Through comparison of the numerical metrics, the performance of different models is as follows:

XGBoost > Radom Forest ≈ Neural Network > Decision Tree > Linear Regression

The highest R^2 score obtained by XGBoost is ~0.92, which demonstrates the feasibility of using such model to predict the compressive strength of concrete. The most important features obtained from the tree-based models are Cement, Water, and Age.

Future machine learning models can be further improved when more data on the composition of individual ingredients are available, since concrete is heavily relying on local materials, the composition of which varies across different locations.

Github: https://github.com/DiandZhao/Concreteproject

References

[1] P.J.M. Monteiro, S.A. Miller, A. Horvath, Towards sustainable concrete, Nat. Mater. 16 (2017) 698–699. doi:10.1038/nmat4930.

[2] K.L. Scrivener, Options for the future of cement, Indian Concr. J. 88 (2014) 11–21.

[3] E. Gartner, Industrially interesting approaches to "low-CO_2" cements, Cem. Concr. Res. 34 (2004) 1489–1498. doi:10.1016/j.cemconres.2004.01.021.

[4] S. Kosmatka, B. Kerkhoff, W. Panarese, Design and Control of Concrete Mixtures, 16th Editi, 2016.

[5] I.-C. Yeh, Concrete Compressive Strength Data Set.

https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength.

[6] S.B. Singh, P. Munjal, N. Thammishetti, Role of water/cement ratio on strength development of cement mortar, J. Build. Eng. 4 (2015) 94–100. doi:10.1016/j.jobe.2015.09.003.