EPiC Series in Built Environment

Volume 7, 2026, Pages 1112–1121

Proceedings of Associated Schools of Construction 62nd Annual International Conference

Using AI with Machine Learning Models to Predict Concrete Compressive Strength

Tamer Breakah[1], Mohsen Goodarzi[1], Mohamed Abdel Raouf[1], and Tarek Mahfouz[1]
[1]Ball State University

This paper evaluates the performance of AI generated machine learning models. Three commonly used AI platforms were selected for this study; ChatGPT, Gemini, and Perplexity. The dataset used was uploaded to the AI platforms and the choice of models was left open. The number of models investigated by Perplexity was the highest. Another iteration of the modelling was done by asking the AI platforms to run the same analysis using nine different models and the models were specified in this case. The results showed different prediction accuracy among the models, while the prediction accuracy was close among the platforms, in most cases. The drop in prediction accuracy from the training set to the test set was also checked for all the models and platform to study the overfitting tendency. The results showed that Perplexity had the highest overfitting tendency, while ChatGPT had the lowest. The results show that the use of AI worked well with the different machine learning techniques. On the other hand, the regression models did not work well and require human intervention to make them work better.

Keywords: Artificial Intelligence, Machine Learning, Concrete compressive strength Prediction

Introduction

Concrete is a material that is used extensively across the world. Concrete compressive strength can only be reliably evaluated after placement and testing, a process that requires a significant time delay. Predictive equations serve as a valuable tool, offering users advanced confidence in the concrete's expected performance. These models can also be instrumental in making early-stage design adjustments. While such equations will never substitute for hands-on laboratory and field validation, they do provide supplemental assurance for the anticipated results. In fact, various researchers have already utilized statistical regression models and neural networks for the purpose of predicting concrete compressive strength. Given the growing implementation of artificial intelligence (AI) in modern applications, it has emerged as a viable tool for the prediction of concrete compressive strength. Although AI is featured in many things that we use in our daily lives, generative AI is the branch of AI that became accessible to everyone and is causing a paradigm shift in people’s reliance on AI. In this research, the use of different generative AI platforms will be investigated for potential use in the prediction of concrete compressive strength, using different machine learning techniques, while evaluating the accuracy and reliability of the results.

W. Collins, A. Perrenoud and J. Posillico (eds.), ASC 2026 (EPiC Series in Built Environment, vol. 7), pp. 1112–1121

Predictive Capabilities of Machine Learning

Machine learning (ML) approaches demonstrate superior predictive capabilities, often surpassing conventional statistical models. For example, Gandomi et al. (2014) demonstrated that genetic programming could forecast concrete compressive strength more accurately than standard regression analysis. Similarly, Zhang et al. (2019) applied artificial neural networks (ANN) to predict geopolymer compressive strength, achieving a high degree of precision with a mean absolute error below 5%.

Feature Importance and Optimization

Feature importance methods, such as SHAP values (Lundberg & Lee, 2017), assist in identifying the most crucial variables influencing material strength. This information provides valuable insights for mix design optimization. Furthermore, algorithms such as random forest (Das et al., 2020) are particularly effective in modeling complex, non-linear interactions between these factors.

Model Interpretability and Validation

Contemporary machine learning workflows prioritize model interpretability and rigorous validation to ensure reliability. Techniques such as cross-validation, Bayesian optimization, and uncertainty quantification are commonly employed. These methods serve to enhance the robustness of model forecasts, particularly when addressing the constraints of limited datasets.

Introduction to Different Machine Learning Techniques

In this study, nine different machine learning and statistical approaches were utilized to predict concrete compressive strength. A comparative summary of the techniques employed, highlighting their specific advantages and disadvantages in the context of material strength prediction, is presented (see Table 1).

Table 1. Comparison of machine learning techniques

Technique	***Advantages***	***Disadvantages***
Linear Regression	Interpretable coefficients; simple to implement; low variance on small datasets	Assumes linear relationships; performs poorly with complex, non-linear material interactions
Ridge Regression (Hoerl & Kennard, 1970)	Reduces overfitting compared to standard regression; handles correlated predictors well	Cannot model non-linear interactions without feature engineering
Lasso Regression (Tibshirani, 1996)	Produces sparse models; useful for feature selection	Unstable when predictors are highly correlated; cannot capture non-linearities
Decision Tree (Breiman et al., 1984)	Naturally handles non-linearities; easy to interpret visual rules	Prone to overfitting; high variance
Random Forest (Breiman, 2001)	Robust performance; reduces overfitting relative to single trees	Less interpretable than single trees; computationally heavier
Gradient Boosting (Friedman, 2001; Ke et al., 2017)	Often achieves high accuracy; flexible loss functions	Requires careful hyperparameter tuning; complex to interpret

Table 1. Comparison of machine learning techniques (cont'd)

Technique	***Advantages***	***Disadvantages***
Support Vector Regression (SVR) (Drucker et al., 1997; Smola & Schölkopf, 2004	Effective in high-dimensional spaces; captures complex relationships via kernels	Computationally expensive for large datasets; sensitive to kernel selection
k-Nearest Neighbors (KNN) (Cover & Hart, 1967; Altman, 1992)	Simple and intuitive; no explicit model training required	Sensitive to feature scaling; performance degrades with large datasets
Artificial Neural Network (ANN) (Bishop, 1995; Khashman & Akpinar, 2017)	Extremely flexible; models complex, high-order interactions	Requires large datasets; "black box" nature makes interpretation difficult

Scope of Work

The primary objective of this research is to evaluate the capability of Large Language Model (LLM) based Generative AI platforms to function as accessible data analysis tools for the construction industry. While traditional ML modeling requires significant coding expertise, Generative AI offers a natural language interface that may democratize these predictive capabilities. This study investigates whether commercially available platforms (ChatGPT, Gemini, and Perplexity) can accurately predict concrete compressive strength using historical data without requiring the user to write complex code. The study aims to determine: (1) the comparative accuracy of these platforms against established statistical benchmarks, (2) the reliability of the code and models generated by these AI agents, and (3) the need for human intervention for some of the statistical models.

Methodology

This study utilizes a retrospective analysis of a well-established dataset to benchmark the performance of Generative AI models.

Dataset Description

The study utilized the 1,030-point dataset published by Yeh (1998), accessible via the UC Irvine Machine Learning Repository. This dataset was selected due to its prevalence in literature, allowing for reliable baseline comparisons. The independent variables (predictors) included cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, and age. The dependent variable was concrete compressive strength. The details of the mixture design proportions are provided (see Table 2).

Table 2. Mixture design proportions

Constituent	**Range (kg/m^3)**	**Average Value (kg/m^3)**
Cement	71-600	232.2
Fly ash	0-175	46.4
Blast furnace slag	0-359	79.2
Water	120-238	186.4
Superplasticizer	0-20.8	3.5
Coarse aggregate	730-1322	943.5
Fine aggregate	486-968	819.9

Experimental Procedure

The analysis was conducted using three AI platforms: ChatGPT-4o, Google Gemini Pro, and Perplexity Pro. The process followed these steps:

1. Data Upload: The dataset was uploaded directly to the analysis interface of each platform.
2. Prompt Engineering: To simulate a typical user scenario, the prompt was kept concise and non-technical. The specific prompt used was:
   > *"Perform a machine learning analysis on the attached concrete dataset to predict compressive strength. Use multiple regression and machine learning techniques to find the best fit. Split the data into training and testing sets to validate results."*
3. Iterative Analysis: Initially, the platforms were allowed to select their own models. In a second iteration, the platforms were prompted to run specific models (e.g., forcing SVR or Random Forest) to ensure a direct comparison across all nine techniques listed in Table 1.
4. Validation: The code generated by the AI platforms (Python/Pandas/Scikit-Learn) was reviewed to verify the train/test split ratios (which varied between 70/30 and 80/20 depending on the platform's default) and to ensure no data leakage occurred.

Data Analysis of AI Models

The performance of the models was evaluated using three key metrics:

- R-Squared (R^2): Represents the proportion of variance in the compressive strength that is predictable from the mixture ingredients.
- Root Mean Square Error (RMSE): Measures the average magnitude of the errors.
- Mean Absolute Error (MAE): Represents the average absolute difference between predicted and actual values.

Results

ChatGPT Results

The evaluation was carried out using a train-test split (80/20) to ensure independent testing on unseen data. The first iteration of ChatGPT modeled the data using Random Forest, Decision Tree, Lasso Regression, Ridge Regression and Linear Regression. The other models presented were added to the analysis to include the similar models in the different platforms. The results of the different models are presented in order of accuracy of prediction of the test set (see Table 3). It can be seen in the table that the Random Forest Model yielded the best results, followed by the Gradient Boosting Model. The worst prediction results were achieved by the regression models.

Table 3. ChatGPT AI tool results

Rank	Model	R^2	RMSE (MPa)	MAE (MPa)
1	Random Forest	0.910	4.76	3.54
2	Gradient Boosting	0.900	5.19	3.89
3	Artificial Neural Network	0.890	5.78	4.22
4	k-Nearest Neighbors	0.830	6.80	5.20
5	Decision Tree	0.820	7.02	5.10
6	Support Vector Regression	0.760	8.50	6.40
7	Ridge Regression	0.611	11.05	8.92

Table 3. ChatGPT AI tool results (cont'd)

Rank	Model	R^2	RMSE (MPa)	MAE (MPa)
8	Linear Regression	0.600	11.09	8.98
9	Lasso Regression	0.569	11.57	9.28

To study whether the model was overfitted or not, the results of R^2 for the training set were compared to that of the testing set. The results show that in all the models, the reduction in R^2 was insignificant except for the decision tree, which has the biggest reduction in R^2 (see Figure 1).

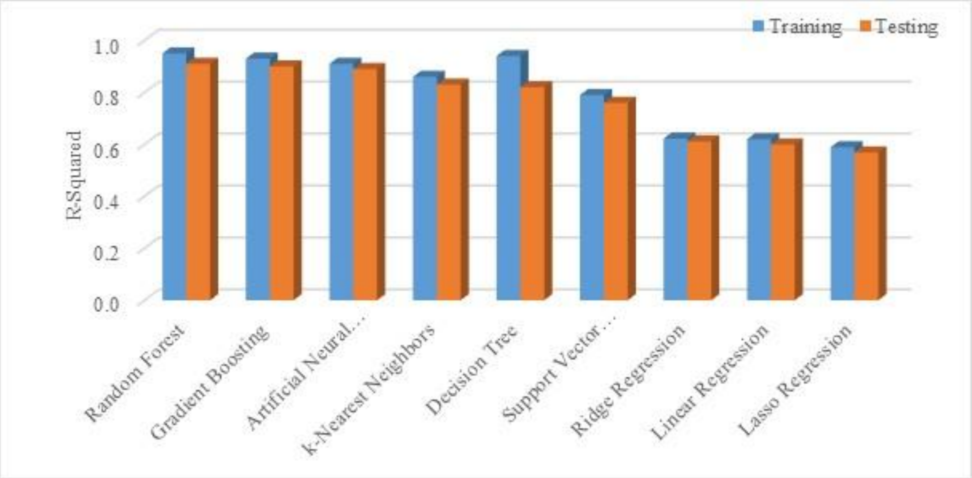

Figure 1. R^2 for the training and testing sets using ChatGPT

Gemini Results

The evaluation was carried out using a train-test split (80/20). The first iteration of Gemini modeled the data using Random Forest, Artificial Neural Network, Support Vector Regression, and Linear Regression. The results of the different models are presented in order of accuracy of prediction of the test set (see Table 4). It can be seen in the table that the Gradient Boosting Model yielded the best results, followed by the Random Forest Model. The worst prediction results were achieved by the regression models.

Table 4. Gemini AI tool results

Rank	Model	R^2	RMSE (MPa)	MAE (MPa)
1	Gradient Boosting	0.950	3.74	2.73
2	Random Forest	0.984	2.14	1.35
3	Artificial Neural Network	0.894	5.43	4.05
4	Support Vector Regression	0.932	4.37	2.64
5	Decision Tree	0.977	2.51	1.35
6	K-Nearest Neighbors	0.815	7.19	5.37
7	Lasso Regression	0.625	10.23	8.07
8	Ridge Regression	0.625	10.23	8.05
9	Linear Regression	0.625	10.23	8.05

The comparison of the results of R^2 for the training set to that of the testing set showed that in the models with higher R^2 showed more reduction in R^2 (see Figure 2). This indicates that these models were overfitted.

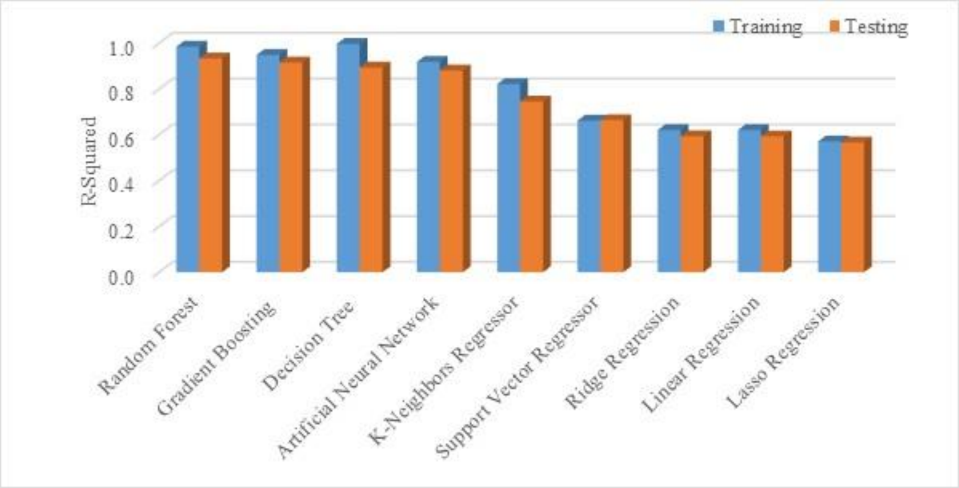

Figure 2. R^2 for the training and testing sets using Gemini

Perplexity Results

The evaluation was carried out using a train-test split (70/30). The first iteration of Perplexity modeled the data using Random Forest, Gradient Boost, Artificial Neural Network, Support Vector Regression, Decision Tree, K-Nearest Neighbor, and Linear Regression. It is worth noting that out of the three platforms in the study, Perplexity was the one that tested the highest number of models in the first iteration. The results of the different models are presented in order of accuracy of prediction of the test set (see Table 5). It can be seen in the table that the Random Forest Model yielded the best results, followed by the Gradient Boosting Model. The worst prediction results were achieved by the regression models.

Table 5. Perplexity AI tool results

Rank	Model	R^2	RMSE (MPa)	MAE (MPa)
1	Random Forest	0.933	4.48	3.30
2	Gradient Boosting	0.916	5.03	3.68
3	Decision Tree	0.893	5.66	3.74
4	Artificial Neural Network	0.880	5.99	4.60
5	K-Neighbors Regressor	0.745	8.74	6.93
6	Support Vector Regressor	0.664	10.05	7.92
7	Ridge Regression	0.593	11.05	8.97
8	Linear Regression	0.593	11.05	8.97
9	Lasso Regression	0.566	11.41	9.35

The comparison of the results of R^2 for the training set to that of the testing set showed that there is a reduction in R^2 for all the models, which indicates that Perplexity tends to overfit the models (see Figure 3).

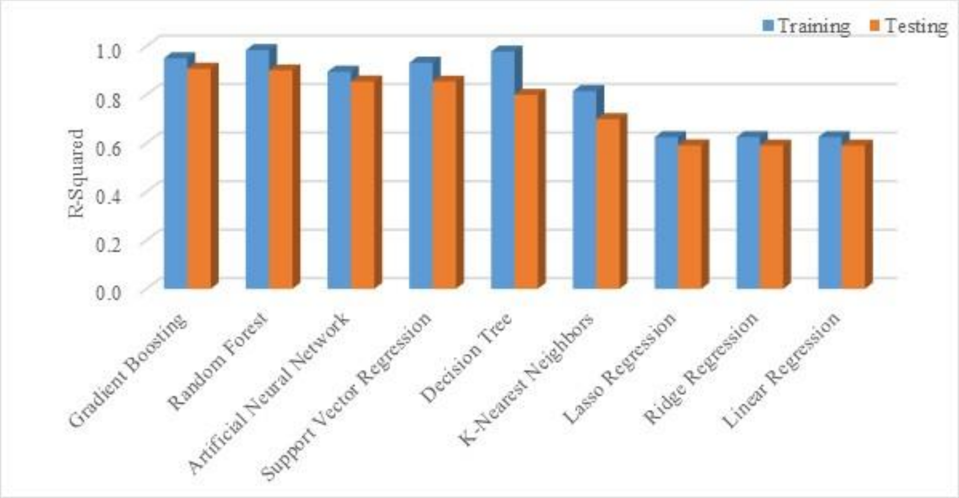

Figure 3. R^2 for the training and testing sets using Perplexity

Platform Comparison

In order to compare the results of the different platforms, the values of R^2 for each model from the different platforms were compared (see Figure 4). The results show that the three platforms had comparable results. Perplexity had lower results in Artificial Neural Networks. Support Vector Regression, K-Nearest Neighbor, and Decision Tree had the highest variability in the R^2 between the results of the different platforms.

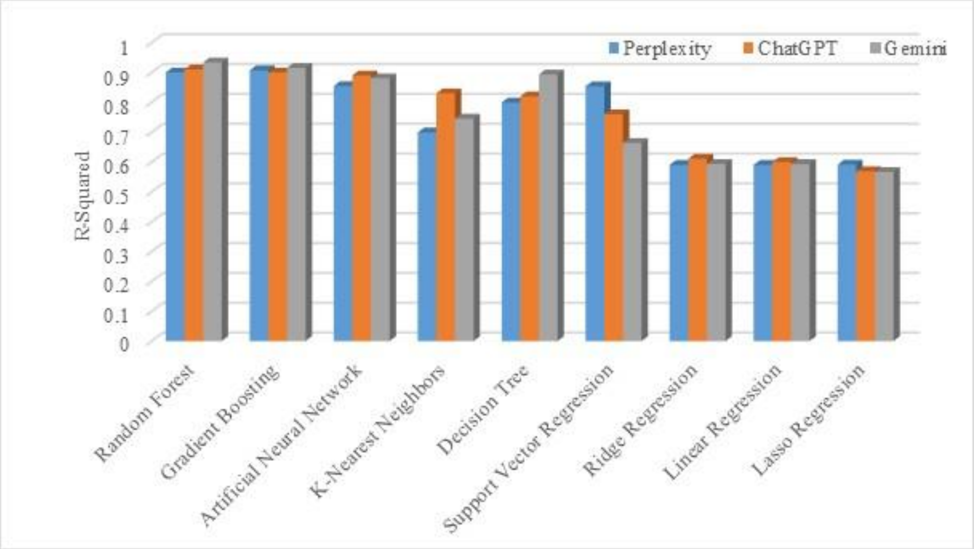

Figure 4. R^2 for the different models using the different platforms

To evaluate whether the models have been overfit or not, the percentage decrease in R^2 for the different models was calculated for each model for the results achieved from the different platforms. A higher reduction in R^2 indicates a model overfit and that the model cannot be used on different datasets. The results show that Perplexity had the highest reduction in R^2 among the three platforms. This result is consistent for all the models. The best results were achieved by ChatGPT for most of the models. For the models, Decision Tree and KNN had the worst results.

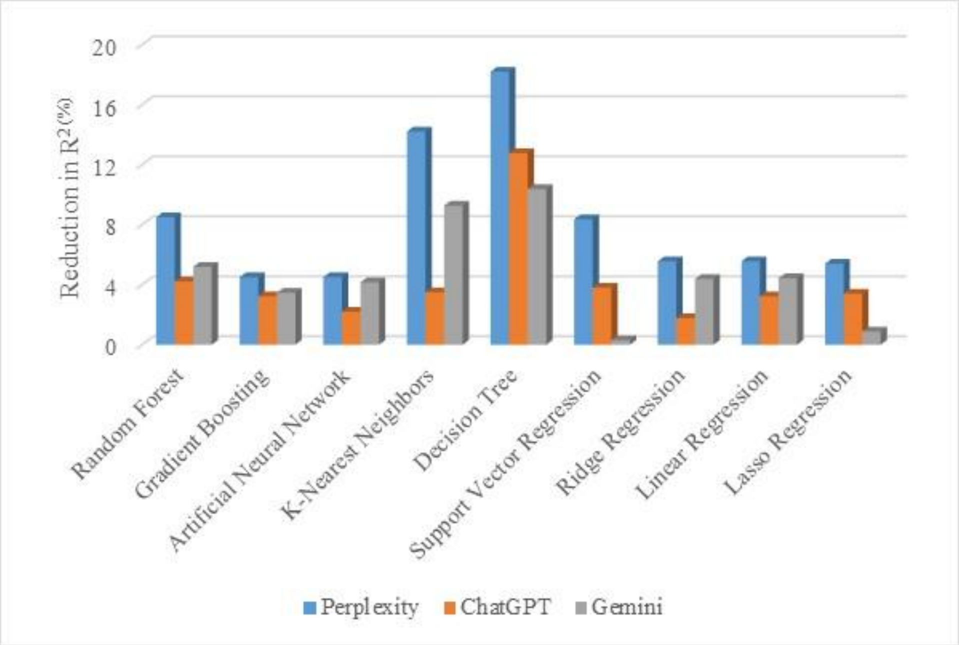

Figure 5. Percentage reduction in R^2 from training set to testing set

Discussion

In addition to the results presented earlier for each model. An additional comparison was done to evaluate the performance of the models as compared to traditional models. The same dataset was analyzed earlier using linear regression analysis (Breakah, Osei Safo-Kantanka, & Attallah, 2021) and had an R^2 value of 0.809, which is significantly higher than the results achieved in this study. This shows that regression modeling can be enhanced by human intervention and that the basic generative AI analysis does not overcome the weakness of the regression method. It can also be seen from the results that the decision tree model also had the highest overfitting, which is also a known drawback of this method. The same problem is also seen in the KNN models. It can also be seen that Perplexity models, in general, had the highest tendency to overfit the models. The following observations were observed by the authors during the study and the different steps of the analysis.

- Perplexity provided the best graphics that summarize the results and these were provided automatically without any intervention from the authors.
- Perplexity and ChatGPT automatically give the user the Python code of the analysis, which can be used later to replicate the analysis.
- All the analysis was done twice and the results achieved were very close. This means that the analysis is repeatable using the generative AI platforms.

It is important to note that the dataset used (Yeh, 1998) reflects concrete materials available in the late 1990s. Significant changes in material technology have occurred since then, such as the adoption of Type 1L (Portland-Limestone) cement and advanced polycarboxylate ether (PCE) superplasticizers.

While the AI models demonstrated high predictive capability on the historical data, the specific equations derived here should be recalibrated with modern datasets before being applied to current construction projects. This highlights the need for continuous data collection in the construction industry to keep predictive models relevant.

Practical Application in Construction Management

To illustrate the application of these tools, consider a scenario where a construction manager needs to validate a proposed mix design that slightly deviates from standard ratios. Instead of waiting days for laboratory validation batches, the manager can input the specific mix proportions into a generative AI platform (e.g., ChatGPT) using the prompt structure defined in this study. The model can provide an immediate compressive strength prediction, allowing the manager to make a risk-informed decision on whether to proceed with the pour or request a mix adjustment. This "screening" capability can significantly reduce project delays and material waste.

Conclusions and Recommendations

This study explored the potential of Generative AI to serve as an accessible analytical tool for predicting concrete compressive strength. Based on the comparative analysis of ChatGPT, Gemini, and Perplexity, the following conclusions are drawn:

1. Accessibility for Industry: Generative AI platforms successfully automated the complex workflow of machine learning—including data cleaning, splitting, model training, and validation—making high-level statistical analysis accessible to construction professionals who may lack programming skills.
2. Model Performance: Ensemble methods, specifically Random Forest and Gradient Boosting, consistently outperformed traditional linear regression and neural networks across all platforms. This suggests these models should be the default choice for concrete mix prediction.
3. Reliability and Overfitting: While results were generally accurate, overfitting was observed with Perplexity, which showed significant performance drops between training and testing data.
4. Implementation Strategy: The authors recommend a "Human-in-the-Loop" approach. Construction managers can use these AI tools to generate initial mix verifications and identify non-linear trends that simple spreadsheets miss. However, final mix designs must still undergo physical laboratory testing. AI should be viewed as a screening tool that optimizes the number of trial batches required, rather than a replacement for physical validation.

Future Research

Future work should expand this evaluation to include Large Language Models (LLMs) specifically fine-tuned on engineering datasets, rather than general-purpose models. Additionally, research should investigate the environmental impact of these tools, exploring how AI-optimized mix designs can contribute to reducing the carbon footprint of concrete production by minimizing cement usage while maintaining required strength.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175–185.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Breakah, T. M., Osei Safo-Kantanka, J., & Attallah, S. O. (2021). Development of a regression equation to predict concrete strength. *International Journal of Engineering Research & Innovation, 13*(1), Spring/Summer.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Das, S. K., Samui, P., & Khan, S. Z. (2020). Machine learning techniques for the prediction of compressive strength of concrete. *Informatics*, *7*(3), 28. https://doi.org/10.3390/informatics7030028

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 9, 155–161.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Gandomi, A. H., Babanajad, S. K., Alavi, A. H., & Mirzahosseini, M. R. (2014). Novel approach to strength modeling of concrete using genetic programming. *Journal of Materials in Civil Engineering*, *26*(1), 60–67. https://doi.org/10.1061/(ASCE)MT.1943-5533.0000780

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*, 3146–3154.

Khashman, A., & Akpinar, P. (2017). Non-destructive prediction of concrete compressive strength using neural networks. *Procedia Computer Science*, *108*, 2358-2362.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4765–4774). Curran Associates, Inc.

Yeh, I.-C. (1998). *Modeling of strength of high-performance concrete using artificial neural networks*. Cement and Concrete Research, 28(12), 1797–1808

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Zhang, J., Liu, Y., Wang, Y., & Li, Q. (2019). Application of ANN to predict geopolymer compressive strength. *Materials*, *12*(17), 2735. https://doi.org/10.3390/ma12172735