mathematics

MDPI

Article

High-Performance Concrete Strength Regression Based on Machine Learning with Feature Contribution Visualization

Lei Zhen [1], Chang Qu [2], Man-Lai Tang [3] and Junping Yin [1,4,5,*]

[1] Academy for Advanced Interdisciplinary Studies, Northeast Normal University, Changchun 130024, China; zhenlei@nenu.edu.cn
[2] Department of Mathematics, Shantou University, Shantou 515063, China; quchang@stu.edu.cn
[3] Department of Physics, Astronomy and Mathematics, University of Hertfordshire, Hatfield AL10 9AB, UK; m.l.tang@herts.ac.uk
[4] Institute of Applied Physics and Computational Mathematics, Chinese Academy of Sciences, Haidian District, Beijing 100094, China
[5] Shanghai Zhangjiang Institute of Mathematics, Shanghai 201201, China
* Correspondence: yin_junping@iapcm.ac.cn

Abstract

Concrete compressive strength is a fundamental indicator of the mechanical properties of High-Performance Concrete (HPC) with multiple components. Traditionally, it is measured through laboratory tests, which are time-consuming and resource-intensive. Therefore, this study develops a machine learning-based regression framework to predict compressive strength, aiming to reduce experimental costs and resource usage. Under three different data preprocessing strategies—raw data, standard score, and Box–Cox transformation—a selected set of high-performance ensemble models demonstrates excellent predictive capacity, with both the coefficient of determination (R^2) and explained variance score (EVS) exceeding 90% across all datasets, indicating high accuracy in compressive strength prediction. In particular, stacking ensemble (R^2-0.920, EVS-0.920), XGBoost regression (R^2-0.920, EVS-0.920), and HistGradientBoosting regression (R^2-0.913, EVS-0.914) based on Box–Cox transformation data show strong generalization capability and stability. Additionally, tree-based and boosting methods demonstrate high effectiveness in capturing complex feature interactions. Furthermore, this study presents an analytical workflow that enhances feature interpretability through visualization techniques—including Partial Dependence Plots (PDP), Individual Conditional Expectation (ICE), and SHapley Additive exPlanations (SHAP). These methods clarify the contribution of each feature and quantify the direction and magnitude of its impact on predictions. Overall, this approach supports automated concrete quality control, optimized mixture proportioning, and more sustainable construction practices.

Keywords: High-Performance Concrete; concrete compressive strength; regression model; SHAP

MSC: 62J02; 68T30

Academic Editor: Adara M. Blaga

Received: 9 November 2025
Revised: 5 December 2025
Accepted: 10 December 2025
Published: 12 December 2025

Citation: Zhen, L.; Qu, C.; Tang, M.-L.; Yin, J. High-Performance Concrete Strength Regression Based on Machine Learning with Feature Contribution Visualization. *Mathematics* **2025**, *13*, 3965. https://doi.org/10.3390/math13243965

1. Introduction

High-Performance Concrete (HPC) is the backbone of modern infrastructure, providing essential strength, durability, workability, and exceptional volume stability. Its application is indispensable in supporting key national projects, including ultra-high-rise

buildings, long-span transportation systems, maritime and coastal engineering, and other specialized projects vital to economic development and public welfare. HPC serves not only as a foundation for technological innovation in contemporary engineering but also as a key driver in reducing project life-cycle costs and promoting low-carbon, sustainable development within the construction industry. In applications where concrete compressive strength plays a critical role, the long-term benefits of HPC—such as extended service life and reduced operating and maintenance costs—far outweigh its initial expense. For instance, in iconic structures such as the Burj Khalifa in Dubai, HPC of strength grades C80–C100 has been utilized to meet cubic compressive strength requirements, ensure core structural integrity, and enhance deformation resistance. Similarly, long-span bridges like the Hong Kong–Zhuhai–Macao Bridge have made extensive use of C60–C80 maritime-grade HPC. By incorporating 30% ground granulated blast-furnace slag (GGBS) and 15% silica fume, the chloride ion diffusion coefficient has been reduced to below 1.5×10^{-12} m^2/s, which is two orders of magnitude lower than that of conventional concrete. This improvement has ensured the bridge's long-term durability and corrosion resistance in harsh marine environments. However, the high cost and technical complexity of HPC become more apparent in ordinary structures with low load demands, short service cycles, or cost-sensitive requirements (such as temporary industrial facilities or low-rise residential buildings), where ordinary concrete remains a more economical option. Consequently, the scientific measurement and systematic evaluation of HPC's compressive strength have become crucial areas of focus in both engineering practice and academic research.

The proportion of each component must be precisely controlled during HPC preparation. Chemical admixtures such as superplasticizers and cementitious materials including fly ash and blast-furnace slag are essential for its formulation. Table 1 lists the dataset used in this study, comprising 1030 samples and 9 variables (available at: https://archive.ics.uci.edu/dataset/165, accessed on 7 August 2025). This dataset, a widely recognized benchmark, is frequently used in materials science, civil engineering, and regression modeling research [1,2]. Yeh [1] stated that High-Performance Concrete (HPC) is an extremely complex material. For example, concrete compressive strength is a highly nonlinear function of age; it depends on the water-to-cement ratio as well as on other constituent materials that make up the concrete. To further quantify the workability (primarily fluidity) of concrete, Yeh et al. [2] developed a workability model that examined the effects of all concrete components—excluding air content—and their relative proportions (e.g., the water-to-binder ratio) on concrete slump. Therefore, regression modeling of concrete compressive strength presents a significant scientific and computational challenge.

Prognostics and Health Management (PHM) is vital in transportation, economic and social production, materials science, and other important fields. Concrete compressive strength, which directly represents a material's capacity to withstand axial stresses without failure, is the primary metric used to evaluate the quality of concrete. In construction engineering, this property is essential for material optimization, service life prediction, and ensuring structural safety. The goal of this study is to develop a regression model for predicting concrete compressive strength and to investigate how various complex factors influence it.

Research on concrete strength prediction has shown increasingly diversified trends: predictive models have evolved from single architectures to hybrid/ensemble frameworks for more robust performance; focus has shifted beyond accuracy to model interpretability, quantifying each input feature's impact on outcomes; application scenarios have expanded from general-purpose to specialized/high-performance concrete. In the early stages, research on the regression modeling of High-Performance Concrete (HPC) compressive strength primarily focused on model accuracy. Studies have shown that Artificial Neural

Network (ANN)-based models outperform traditional regression analysis in predicting concrete compressive strength, while also simplifying numerical experimentation [1]. Moreover, advanced data mining techniques have emerged as powerful tools for addressing various civil engineering (CE) problems. A novel support vector regression (SVR) approach based on the smart artificial firefly colony algorithm has been proposed to significantly improve the prediction accuracy of concrete compressive strength [3]. Additionally, practical global solutions for regression can be obtained by formulating the problem as a mixed-integer nonconvex nonlinear optimization problem with convex penalty terms [4]. It is evident that machine learning (ML) techniques have been widely applied across numerous disciplines, including civil engineering, where they play an important role in analysis and predictive modeling [5]. To further enhance the precision of forecasting the compressive strength of HPC, researchers have conducted extensive analyses using machine learning algorithms and optimization-based approaches. For example, previous studies have explored the prediction of concrete compressive strength using a range of methods—such as support vector machines, neural networks, generalized linear models, and variable selection techniques—aiming to achieve high predictive accuracy and interpretability [6–13]. In pursuit of balancing model complexity and accuracy, researchers have also examined the convergence speed of optimization-based algorithms, including the Levenberg–Marquardt algorithm and Mamdani fuzzy inference systems [14–16]. Furthermore, this dataset has been utilized in several related studies [17–21], highlighting its importance as a benchmark for evaluating predictive models in civil engineering.

Table 1. Description of the attributes of HPC.

Name	Symbol for Plot	Data Type	Measurement	Description
x_1: Cement	cement	Continuous	kg/m^3	Amount of cement used in the concrete mixture
x_2: Blast Furnace Slag	blastFurnace	Integer	kg/m^3	Amount of blast furnace slag used as a partial replacement for cement
x_3: Fly Ash	flaAsh	Continuous	kg/m^3	Amount of fly ash used as a partial replacement for cement
x_4: Water	water	Continuous	kg/m^3	Amount of water used in the concrete mixture
x_5: Superplasticizer	superplasticizer	Continuous	kg/m^3	Amount of superplasticizer added to improve workability
x_6: Coarse Aggregate	courseAggregate	Continuous	kg/m^3	Amount of coarse aggregate in the mixture (gravel or crushed stone)
x_7: Fine Aggregate	fineAggregate	Continuous	kg/m^3	Amount of fine aggregate in the mixture (sand)
x_8: Age	age	Integer	Days (1∼365)	Curing age of the concrete (in days)
y : Concrete Compressive Strength	strength	Continuous	MPa	The strength of the concrete under compression

Currently, regression models based on machine learning have demonstrated strong predictive performance. Chou et al. [22] demonstrated that the proposed multiple additive regression tree (MART) algorithm outperforms in terms of prediction accuracy, training efficiency, and resistance to overfitting. Erdal et al. [23] proposed a hybrid framework coupling discrete wavelet transform (DWT) with artificial neural network (ANN) ensembles to enhance prediction performance. Their findings confirmed that DWT serves as an effective tool for improving the accuracy of ANN ensemble models. Feng et al. [24] utilized the adaptive boosting (AdaBoost) algorithm to construct a strong learner by integrating multiple weak learners, resulting in a significant enhancement of the overall prediction accuracy. Nguyen-Sy et al. [25] highlighted that the extreme gradient boosting (XGBoost, XGB) method exhibits greater robustness, faster training speed, and higher prediction accuracy compared to artificial neural network (ANN) and support vector machine (SVM) approaches reported in the literature. Zeng et al. [26] developed a deep learning-based

"factors-to-strength" approach that incorporates multiple explainable features to improve the interpretability and reliability of strength prediction models. The studies by [22–26] achieved accuracies of 0.91 (MART), 0.95 (DWT), 0.95 (AdaBoost), 0.93 (XGBoost), and 0.967 (deep learning-based "factors-to-strength" approach) in terms of the coefficient of determination (R^2) on the test sets. Similar studies have also been reported in the literature [27–31]. However, although these models achieved high accuracy through intricate hyperparameter tuning, they also have the drawback of being difficult to interpret. This raises several questions: Can additional techniques be incorporated to improve the interpretability and transparency of the results? Which machine learning methods perform best for regression modeling on this dataset, and how can we demonstrate that the model effectively evaluates the data? To address these issues, this paper focuses on the following aspects. First, we conduct an exploratory analysis of the dataset to understand the relationships and distributions among the features. Second, we compare the performance of several regression techniques under different data transformations using multiple evaluation metrics. Although some models include hyperparameters, they are implemented in their basic form without optimization methods such as cross-validation or grid search. Finally, we enhance model interpretability by applying explainable AI frameworks such as SHapley Additive exPlanations (SHAP), Individual Conditional Expectation (ICE), and Partial Dependence Plots (PDP).

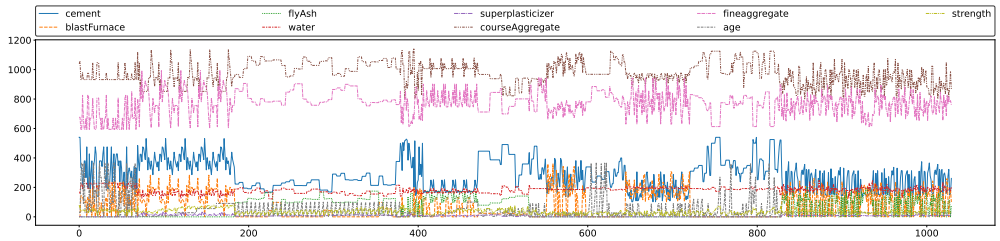
2. Materials and Methods

2.1. Database Description and Explanatory Data Analysis

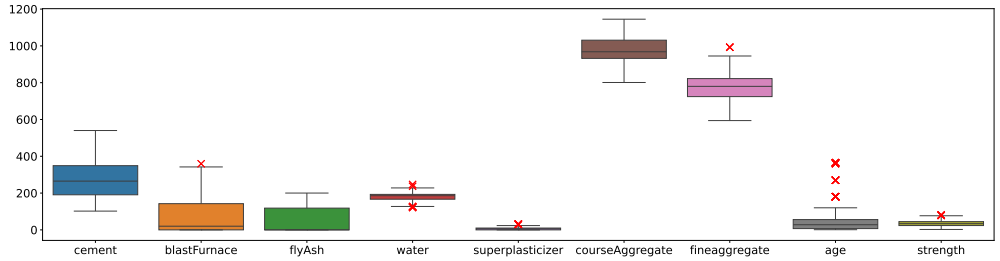
This section presents a visual analysis of the data to highlight feature distributions and interrelationships, which in turn guide modeling decisions. The HPC dataset exhibits high data integrity, containing only 25 duplicates out of 1030 samples (2.4%) and a complete absence of missing values. Here, unlike Yeh [1] which considered only 700 samples, we include all 1030 samples (i) to avoid overly aggressive preprocessing that may introduce selection bias, and (ii) to evaluate which models remain robust when trained on data of general (rather than highly curated) quality. However, all duplicate entries were removed prior to analysis.

A basic description of the numerical variation among the nine attributes is provided by the area plot and boxplot of concrete characteristics shown in Figure 1. The nine attributes exhibit clear patterns of variation, as shown in Figure 1a: horizontal comparisons reveal synchronized trends among several attributes, suggesting possible correlations, while vertical comparisons highlight notable differences in magnitude or behavior. Boxplots of the nine attributes, presented in Figure 1b, show significant differences in their value distributions. For example, Coarse Aggregate and Fine Aggregate display noticeably higher magnitudes than the other attributes. Additionally, with the exception of Cement, Fly Ash, and Coarse Aggregate, Figure 1b shows observable outliers in six attributes. Therefore, during the initial data-cleaning phase, operations such as outlier removal, categorical feature encoding, and other preprocessing steps were performed.

Figure 2 presents the correlation coefficients and paired distributions of the nine attributes. Figure 2a illustrates the pairwise correlations among the attributes, while Figure 2b categorizes Concrete Compressive Strength into three groups and displays scatter and density plots for each pair of attributes. As observed, Figure 2a indicates that most attributes exhibit weak or insignificant linear correlations, and Figure 2b similarly shows no visually apparent linear relationships among most attribute pairs.

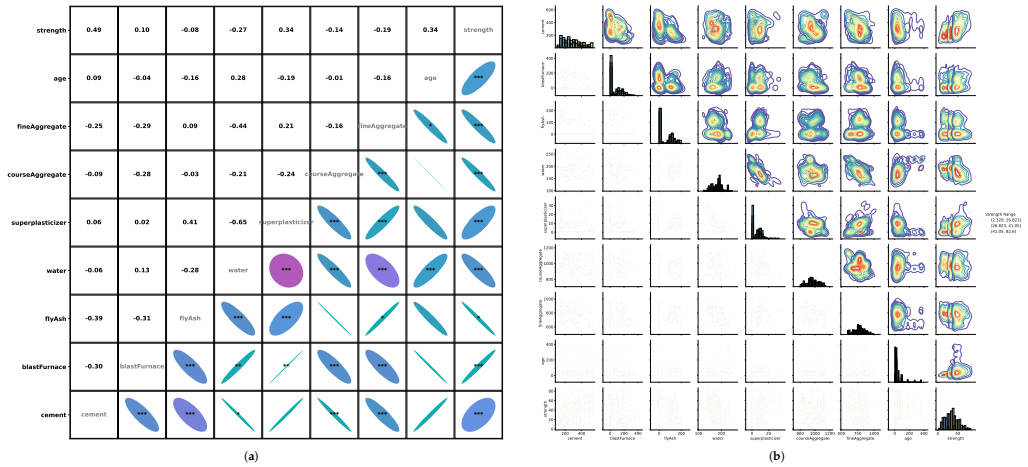


(**a**) Area plot



(**b**) Boxplots

Figure 1. Area plot and boxplots of 9 attributes.



(**a**) (**b**)

Figure 2. Correlation between 9 attributes. (**a**) Pearson correlation matrix with Spearman significance test in the bottom right; ellipses denote the strength and direction of correlations (The significance levels of Spearman are typically denoted using asterisks. A single asterisk (*) indicates a significance level of 0.1; two asterisks (**) denote a significance level of 0.05; three asterisks (***) represent a significance level of 0.01; and the absence of asterisks implies that the results are not statistically significant). (**b**) Pairwise relationships, the bar plots on the diagonal is the distribution characteristics of the attributes.

In particular, from the perspective of individual attributes, the following observations can be made:

- The main diagonal of Figure 2b shows considerable skewness for Blast Furnace Slag, Fly Ash, and Superplasticizer. Many sample values are zero or extremely low, indicating that these components are used less frequently in certain concrete mixtures.
- Water, Coarse Aggregate, and Fine Aggregate exhibit approximately normal distributions.
- The near-normal distribution of Concrete Compressive Strength suggests that it is suitable for various statistical analysis techniques that assume normality.

The following key findings are derived from the examination of relationships among the attributes:

- Cement content (0.49) and Water (-0.27) show linear correlations with Concrete Compressive Strength, while no comparably strong relationships are observed for Superplasticizer (0.34) or Age (0.34), as shown in the scatter plots in Figure 2b.
- The scatter plots of Concrete Compressive Strength in Figure 2b indicate that it results from a combination of multiple factors. For different strength levels, none of the eight features individually exhibit a clear dominant influence.
- Attributes such as Coarse Aggregate, Fine Aggregate, and Water do not display significant linear relationships with Concrete Compressive Strength, but they may interact in more complex, nonlinear ways.
- The attribute densities vary, with some regions showing higher concentrations, as revealed by the density maps in Figure 2b. This suggests that certain mixture components are more frequently used, which aligns with the attribute fluctuation patterns observed in Figure 1a.

To improve modeling performance, potential preprocessing steps could include feature transformations such as logarithmic, Box–Cox, or Yeo–Johnson transformations, which can convert skewed data distributions into approximately normal ones [32]. Most importantly, emphasis will be placed on employing advanced modeling techniques such as ensemble trees, gradient boosting, and neural networks, which are capable of capturing nonlinear interactions among features.

2.2. Machine Learning

Concrete compressive strength has traditionally been determined through laboratory testing, which is a time-consuming and resource-intensive process. With the advancement of machine learning [33–35], data-driven approaches for estimating concrete compressive strength from observed samples have become increasingly popular. By reducing the need for physical testing and resource consumption, such models enable automated quality control, mix optimization, and promote more sustainable construction practices.

Based on the data analysis presented above, this section employs a variety of regression methods from statistics, machine learning, and related fields to establish the following regression framework:

$$y = m(x) + \varepsilon, \tag{1}$$

where $x = (x_1, x_2, \ldots, x_8)^\top$ represents the covariates, y is the response variable, and ε denotes the model error. Here, the function $m(x)$ denotes the true but unknown regression function that maps the covariates x to the expected concrete compressive strength. In practice, $m(x)$ cannot be expressed in closed form due to the highly nonlinear, multi-factor nature of concrete behavior. Thus, $m(x)$ is introduced as a theoretical construct, and the model's task is to approximate it as closely as possible, eventually evaluated using metrics such as the coefficient of determination (R^2), explained variance score (EVS), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) in Section 2.2.2.

One of the objectives of this study is to conduct a comprehensive benchmark analysis using classical and widely adopted regression models. These were selected based on three criteria commonly emphasized in the literature: (i) coverage breadth across different methodological families, (ii) representativeness of mainstream algorithms used in concrete-strength prediction studies, and (iii) computational efficiency.

2.2.1. Overview of Regression Methods

To model Equation (1), most classical statistical regression approaches rely on the structure of parametric regression models, such as $f(x) = x^\top \beta$, where $\beta = (\beta_1, \beta_2, \ldots, \beta_8)^\top$

are the coefficients to be estimated. When using penalized least squares regression, the objective function to be minimized takes the form:

$$\mathcal{L}(\beta) = \underbrace{\sum_{i=1}^{n}(y_i - x_{i\cdot}^{\top}\beta)^2}_{\text{Sum of Squares of Residuals}} + \underbrace{\lambda \sum_{j=1}^{8} P(\beta_j)}_{\text{Penalty Function}}, \tag{2}$$

where $x_{i\cdot}$ represents the 8×1 covariate vector for the i-th sample, $P(\cdot)$ represents the penalty function, and λ is the penalty parameter [36,37]. Common penalized regression techniques include Ridge Regression [38], Lasso Regression [39], and Least Angle Regression [40].

To overcome the limitations of linear regression modeling, several nonparametric approaches can be applied, such as Gaussian Process Regression [41,42] and K-Nearest Neighbors (KNN) Regression [43,44].

To better capture nonlinear interactions among features, data-driven techniques such as tree-based models and boosting algorithms are explored. In boosting, the learning objective is defined as:

$$\mathcal{L} = \|y - F_M(x)\|^2, \text{ with } F_M(x) = \sum_{m=1}^{M} \nu \cdot h_m(x), \tag{3}$$

where $F_M(x)$ is the final strong learner corresponding to the model's prediction, $h_m(x)$ represents the m-th weak learner, and ν is the learning rate (also known as the shrinkage factor), typically a small positive value (e.g., 0.1). This parameter serves as a key regularization term that mitigates overfitting by controlling the contribution of each weak learner. Tree-based learning methods have been extensively studied in foundational works such as [45–50], while the development of boosting algorithms builds on key contributions including [51–56].

Deep learning is another well-known regression framework that can achieve exceptionally high predictive performance with appropriately designed architectures [57,58]. Building upon individual regression models, ensemble methods can further enhance predictive accuracy. A straightforward yet effective strategy is to take a weighted average of multiple model outputs as the final prediction. For instance, combining Gradient Boosted Trees (GBT) with clustering algorithms such as K-means or bisecting K-means can improve model robustness and generalization [59]. A more sophisticated approach is the stacking ensemble, which involves training a meta-model to optimally combine the predictions of several base learners. Two stacking ensemble frameworks are considered in this study:

- Stacking Ensemble-1, consisting of XGBoost, HistGradientBoosting, and LGBM Regressor;
- Stacking Ensemble-2, comprising AdaBoost, Decision Tree, Random Forest, Gradient Boosting, and Extra Trees Regressor.

In total, this study evaluates eighteen regression models, which include 16 individual regression models (e.g., XGBoost Regressor and HistGradientBoosting Regressor) and the aforementioned two stacking ensemble models (see Table 2 for a complete list). Together with the data transformation techniques discussed in subsequent sections, a comprehensive performance comparison is conducted using multiple evaluation metrics. Depending on the model, hyperparameters are either pre-set or predefined based on standard configurations. We recognize that performance may further improve with hyperparameter optimization, as reported in studies such as [22–26]. However, such improvements are typically modest (e.g., RMSE reductions of less than 5%). Our primary contribution lies in interpretable visualization rather than in proposing new machine learning techniques, we intentionally used standard configurations to ensure a fair and transparent baseline comparison.

Table 2. Performance evaluation of regression models on raw data.

	Model	R^2	EVS	RMSE	MAE	MAPE	Time (s)
1	Stacking Ensemble-1	0.920	0.921	4.781	3.204	0.111	9.163
2	XGBRegressor	0.919	0.919	4.820	3.167	0.109	0.549
3	Stacking Ensemble-2	0.915	0.916	4.940	3.431	0.112	4.414
4	HistGradientBoostingRegressor	0.913	0.914	4.995	3.379	0.119	0.578
5	LGBMRegressor	0.912	0.912	5.025	3.398	0.118	0.119
6	ExtraTreesRegressor	0.910	0.910	5.098	3.382	0.113	0.195
7	RandomForestRegressor	0.897	0.899	5.430	3.839	0.137	0.374
8	GradientBoostingRegressor	0.888	0.890	5.662	4.151	0.141	0.138
9	DecisionTreeRegressor	0.848	0.848	6.609	4.506	0.158	0.008
10	MLPRegressor	0.796	0.800	7.649	5.568	0.186	0.857
11	AdaBoostRegressor	0.782	0.787	7.917	6.593	0.287	0.180
12	KNN Regression	0.640	0.642	10.178	7.677	0.299	0.004
13	GaussianProcessRegressor	0.563	0.563	11.212	8.978	0.339	1.334
14	LassoLars Regression	0.562	0.563	11.221	8.976	0.339	0.005
15	Lasso Regression	0.562	0.563	11.221	8.976	0.339	0.003
16	Ridge Regression	0.561	0.562	11.234	8.984	0.339	0.024
17	Linear Regression	0.561	0.562	11.234	8.984	0.339	0.018
18	Neural Network Regression	−0.012	0.000	17.057	13.752	0.628	82.6253

2.2.2. Model Evaluation Methods and Indicators

Let $\hat{y}$ denote the predicted Concrete Compressive Strength from the model on the test set. The following metrics are used to evaluate model performance: the coefficient of determination (R^2), Explained Variance Score (EVS), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{4}$$

$$\mathrm{EVS} = 1 - \frac{\mathrm{Var}(y - \hat{y})}{\mathrm{Var}(y)}, \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{6}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{7}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{8}$$

where n is the number of test samples, and $\bar{y}$ is the mean of the observed (actual) values. The R^2 metric represents the proportion of variance in the dependent variable that is explained by the regression model—the closer its value is to 1, the better the model's goodness of fit. The EVS measures the fraction of variance captured by the model, where a value near 1 indicates superior performance. The RMSE quantifies the average magnitude of prediction errors; a smaller value indicates higher predictive accuracy. Similarly, MAE reflects the average absolute difference between the predicted and actual values, while MAPE expresses this difference as a percentage, offering an interpretable measure of relative error. Lower values of MAE and MAPE indicate greater prediction accuracy. Additionally, the computational efficiency of each model was assessed by recording its training time.

2.2.3. Data Preprocessing Workflow

A consistent data preprocessing pipeline was applied to ensure data quality, model compatibility, and reproducibility. First, duplicate samples were removed throughout the study. Both raw and standardized data were included in model training to evaluate the

impact of standardization on performance. Feature scaling was then performed using appropriate normalization techniques. For instance, the standard score [60] for a given feature x_i is calculated as

$$x_{i,new} = \frac{x_i - \bar{x}_i}{\sigma(x_i)}, i = 1, 2, \ldots, 8 \tag{9}$$

where $\bar{x}_i$ and $\sigma(x_i)$ denote the mean and standard deviation of the covariate x_i, respectively.

The Box–Cox transformation [61,62] was also employed to effectively reduce heteroscedasticity and inter-feature correlations. Its formulation is given by

$$x_{i,new}^{(\lambda)} = \begin{cases} \frac{x_i^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\ \ln x_i, & \lambda = 0 \end{cases} \tag{10}$$

where λ is a hyperparameter.

Figure 3 illustrates the histograms of the eight independent variables, along with their corresponding standardized and Box–Cox transformed versions. Although minor data irregularities were observed, no additional data cleaning was performed. Unlike Yeh [1], who used a refined subset of approximately 700 high-quality samples, the present study retains the full dataset to support a larger and more comprehensive analysis. Finally, a fixed random seed (42) was used to split the dataset into training and testing subsets with a 7:3 ratio for all regression models, i.e., 70% of the data is used for training and 30% for testing, ensuring consistency across experiments. In particular, for the deep learning framework, we constructed a three-layer backpropagation (BP) neural network with the following configuration: hidden layers with activation functions (ReLU, 128), (ReLU, 128), and (sigmoid, 128), respectively, and a total of 500 training epochs.

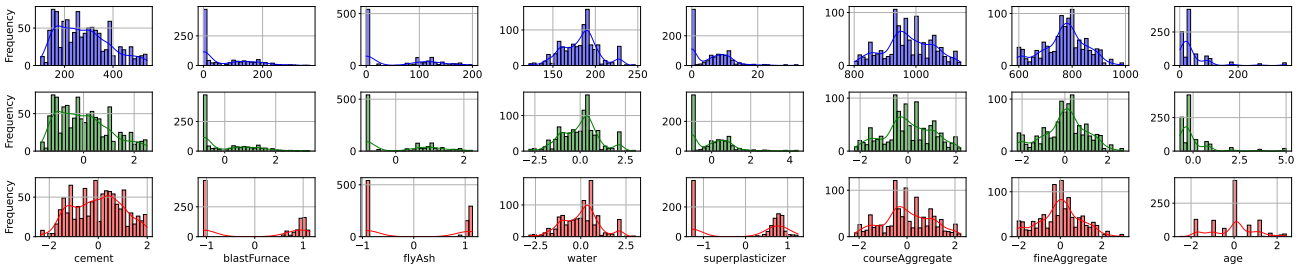


Figure 3. Histograms of the independent variables, their standard scores, and the Box–Cox-transformed versions.

3. Results

This section presents the implementation of regression models to predict Concrete Compressive Strength (CCS) using statistical regression techniques, machine learning algorithms, deep learning architectures, and model fusion approaches [63]. For the test set, model performance was evaluated using the metrics R^2, EVS, RMSE, MAE, and MAPE, as defined in Equations (4)–(8).

A variety of regression models were applied to the original dataset as well as two transformed versions. The detailed results on the test set—including performance metrics and training time—are reported in Tables 2–4. Overall, it was observed that most regression models maintained stable performance across different data transformations; however, their relative predictive rankings vary depending on the transformation applied. Key findings are summarized below:

Table 3. Performance evaluation of regression models on standard score's data.

	Model	R^2	EVS	RMSE	MAE	MAPE	Time (s)
1	Stacking Ensemble-1	0.919	0.919	4.829	3.235	0.111	6.019
2	XGBRegressor	0.919	0.919	4.820	3.167	0.109	0.225
3	HistGradientBoostingRegressor	0.913	0.913	5.002	3.384	0.119	0.437
4	Stacking Ensemble-2	0.912	0.913	5.016	3.418	0.111	3.441
5	LGBMRegressor	0.909	0.910	5.102	3.416	0.119	0.106
6	ExtraTreesRegressor	0.908	0.909	5.141	3.361	0.112	0.198
7	RandomForestRegressor	0.896	0.897	5.462	3.821	0.135	0.398
8	Neural Network Regression	0.892	0.893	5.580	3.494	0.115	79.302
9	GradientBoostingRegressor	0.888	0.890	5.662	4.153	0.141	0.125
10	DecisionTreeRegressor	0.857	0.858	6.409	4.322	0.151	0.009
11	AdaBoostRegressor	0.791	0.795	7.748	6.442	0.291	0.177
12	KNN Regression	0.647	0.649	10.070	7.830	0.297	0.004
13	Ridge Regression	0.561	0.562	11.227	8.984	0.339	0.001
14	Linear Regression	0.561	0.562	11.234	8.984	0.339	0.001
15	GaussianProcessRegressor	0.542	0.542	11.477	9.234	0.372	1.378
16	LassoLars Regression	0.516	0.517	11.793	9.523	0.398	0.003
17	Lasso Regression	0.516	0.517	11.793	9.522	0.398	0.002
18	MLPRegressor	0.419	0.425	12.926	10.288	0.393	0.946

Table 4. Performance evaluation of regression models on Box–Cox data.

	Model	R^2	EVS	RMSE	MAE	MAPE	Time (s)
1	Stacking Ensemble-1	0.920	0.920	4.785	3.214	0.112	5.961
2	XGBRegressor	0.919	0.919	4.820	3.167	0.109	0.227
3	HistGradientBoostingRegressor	0.913	0.914	4.996	3.381	0.119	0.434
4	LGBMRegressor	0.912	0.912	5.025	3.399	0.120	0.113
5	Stacking Ensemble-2	0.911	0.912	5.064	3.512	0.114	3.406
6	ExtraTreesRegressor	0.900	0.901	5.350	3.545	0.118	0.200
7	RandomForestRegressor	0.899	0.899	5.400	3.826	0.136	0.349
8	GradientBoostingRegressor	0.888	0.890	5.669	4.156	0.141	0.125
9	Neural Network Regression	0.884	0.885	5.782	3.359	0.107	78.223
10	DecisionTreeRegressor	0.852	0.853	6.512	4.449	0.154	0.008
11	KNN Regression	0.808	0.809	7.426	5.352	0.190	0.002
12	AdaBoostRegressor	0.790	0.793	7.760	6.429	0.286	0.179
13	Ridge Regression	0.779	0.782	7.972	6.147	0.217	0.001
14	Linear Regression	0.779	0.783	7.971	6.146	0.217	0.001
15	GaussianProcessRegressor	0.771	0.775	8.106	6.206	0.209	1.203
16	MLPRegressor	0.766	0.772	8.195	6.240	0.215	0.953
17	LassoLars Regression	0.746	0.749	8.542	6.474	0.215	0.003
18	Lasso Regression	0.746	0.749	8.542	6.474	0.215	0.002

Key observations are summarized as follows:

- Overall performance: Most statistical regression methods (e.g., linear regression and penalized regressions) performed relatively poorly in terms of predictive accuracy. Some machine learning models—such as MLP regression, KNN regression, and Gaussian Processes—also showed limited performance. In contrast, models such as XGBoost, HistGradientBoosting, and the Stacking Ensemble achieved outstanding predictive accuracy.
- Effectiveness of model ensembling: Based on R^2, EVS, and other metrics, the Stacking Ensemble-1 model consistently outperformed its individual base learners. This demonstrates the advantage of ensemble learning, which integrates multiple models to produce more robust and accurate predictions.
- Tree-based and boosting models: Models such as XGBoost, HistGradientBoosting, and LightGBM achieved R^2 and EVS values above 0.9, along with low RMSE, MAE, and MAPE. These results highlight their superior capability to capture complex nonlinear relationships within material composition data.

- Deep learning performance: When trained on raw data, deep learning models produced unstable or unreliable predictions. However, when applied to properly transformed data, their performance improved dramatically. This contrast underscores the importance of network architecture design, particularly in selecting suitable activation and loss functions. Additionally, training configurations—such as the number of epochs and iterations—significantly influence computational cost and model convergence.
- Training efficiency: Except for the stacked ensemble and deep learning models, most algorithms completed training in under one second.

Boosting-based models emerged as the most effective for this regression task, owing to their strong ability to model nonlinear dependencies, capture feature interactions, and accommodate diverse feature distributions without strict preprocessing requirements. Conversely, penalized regression techniques proved inadequate for this dataset. Among all models, XGBoost regression achieved the best individual performance. Therefore, we use the XGBoost regression model as a representative example to further analyze feature interactions under the Box–Cox transformation. We adopted a standard XGBoost model [53] with default parameters as configured in the XGBoost package (3.0.5): 100 estimators ($M = 100$), a gradient boosting tree (gbtree) as the weak learner, a maximum tree depth of 6, a learning rate of $\eta = 0.3$, and regularization parameters $\lambda = 1$ and $\gamma = 0$.

4. Discussion

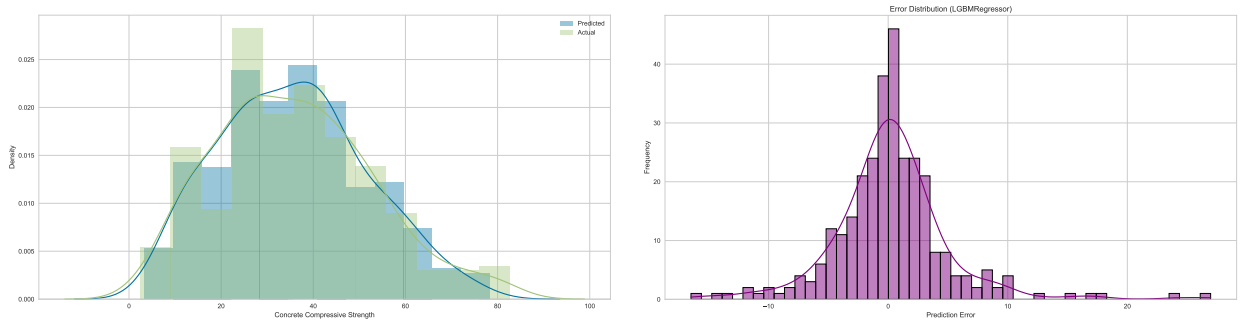
4.1. Predicting Performance

The model's generalization capability—that is, its predictive performance on unseen data—has consistently been the central focus and primary objective of our research and development efforts.

As shown in Figure 4, the XGBoost regression model demonstrates strong generalization performance on the test set. The predicted and actual values of Concrete Compressive Strength exhibit close alignment, while the residuals are symmetrically distributed around zero, indicating low systematic bias. Further improvement in prediction accuracy could be achieved through more refined hyperparameter optimization. For instance, as reported in Feng et al. [24], Nguyen-Sy et al. [25], R^2 values between 0.92 and 0.95 can be achieved under optimized configurations. Compared with existing models, the XGBoost model demonstrates competitive performance. These findings confirm that the XGBoost model provides a reliable and stable foundation for subsequent interpretability analysis and feature importance exploration.

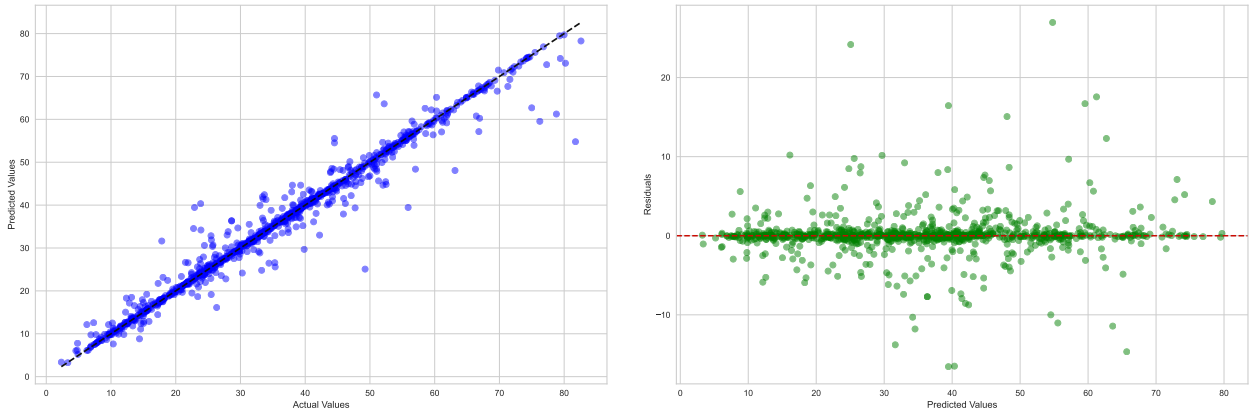
To further demonstrate the generalization of the developed model, we utilized the novel dataset proposed in [64], which includes 103 samples and is used for validation (the curing time is fixed to 28 days).

The XGBoost regression model trained above is employed to make predictions on 103 samples, and the results are presented in Figure 5. As illustrated, the predicted values exhibit a strong correspondence with actual values. The calculated quantitative performance metrics are as follows: $R^2 = 0.971$, EVS $= 0.971$, RMSE $= 2.788$, MAE $= 1.196$, and MAPE $= 4.1\%$. These results collectively demonstrate that the XGBoost regression model achieves high accuracy in predicting the compressive strength of concrete, even when applied to a completely independent dataset.



(**a**) (**b**)

Figure 4. The predictive performance of XGBoost regression model on the test set. (**a**) Density plot of of predicted values and actual values of Concrete Compressive Strength. (**b**) Density plot of residuals ($y - \hat{y}$).



(**a**) (**b**)

Figure 5. The predictive performance of XGBoost regression model on the on 103 samples. (**a**) Scatter plot of actual values vs. predicted values. (**b**) Scatter plot of residuals versus predicted values.

4.2. Feature Dependence

Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE) plots are essential tools for interpretable machine learning. They illustrate the relationships between input features and model predictions, revealing intricate patterns that summary statistics alone may overlook. While the PDP captures the average influence of a feature on model predictions, the ICE plot provides instance-level insights, showing how individual predictions respond to changes in that feature. Together, they offer a comprehensive understanding of both global trends and instance-specific variations, interactions, and anomalies.

Figure 6 shows the PDP and ICE curves. The thick line represents the PDP, while the thin lines correspond to individual ICE curves, each reflecting the behavior of a single observation. The PDP depicts the average effect of a feature on model predictions, whereas the ICE plot shows how the feature influences the prediction for each individual sample.

Based on the PDP results, the following observations can be made:

- The model's average prediction is primarily influenced by Age, Cement, and Water, while other features have a relatively minor effect.
- In line with the correlation coefficients, Age (0.34), Cement (0.49), Blast Furnace Slag (0.10), and Superplasticizer (0.34) exhibit positive associations with the output variable. This suggests that increasing these features tends to raise the predicted concrete strength. Conversely, Water (−0.27), Fine Aggregate (−0.19), and Coarse

Aggregate (−0.14) show negative associations, meaning that higher values of these features are linked to lower predicted strength. These findings are consistent with the correlations observed in Figure 2a.

- Fly Ash shows limited sensitivity in predicted values, supporting its weak correlation (−0.08) in Figure 2a, suggesting either minimal influence or interaction-driven nonlinear effects.

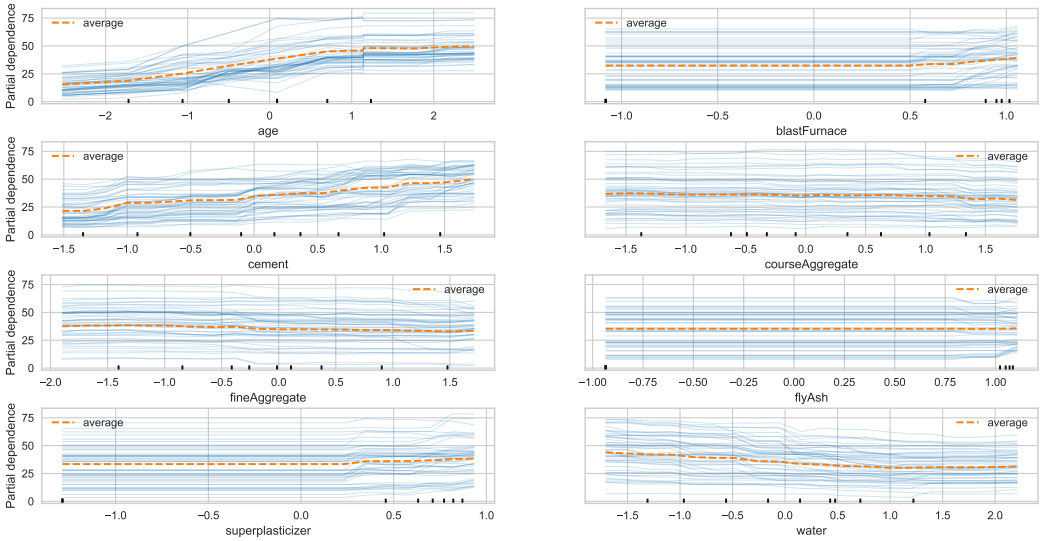


Figure 6. Partial Dependence Plot and Individual Conditional Expectation Plot.

From the perspective of the ICE curves:

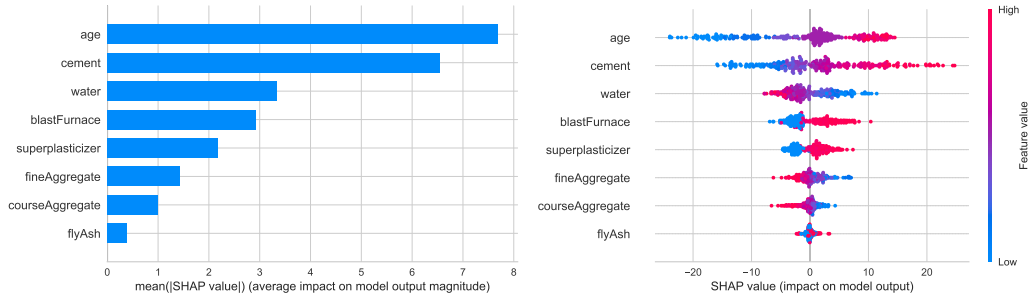
- The features Age, Cement, and Water have the strongest impact on individual predictions across test samples.
- Blast Furnace Slag, Coarse Aggregate, Fly Ash, and Superplasticizer have minimal effects at lower values but exert a greater influence at higher values, indicating potential non-linear or threshold effects. For instance, at low dosage levels, Blast Furnace Slag does not significantly alter the chemical environment of the mixture—its pozzolanic activity is overshadowed by the dominant cement, resulting in only a minimal contribution to the early-age mechanical properties of concrete. In contrast, at high dosage levels, Blast Furnace Slag markedly reduces the heat of hydration and refines the pore structure, thereby substantially improving the long-term compressive strength of the concrete.
- Fine Aggregate exhibits varying effects across its range: it has a stronger positive influence on the predicted value in the lower range than in the upper range.
- Many features—particularly Superplasticizer and Water—demonstrate strong interaction effects at higher values, producing complex, non-linear impacts on the predicted outcomes. These behaviors reflect underlying physico-chemical processes related to adsorption, solution chemistry, and hydration dynamics.

These PDP and ICE analyses provide critical insights into the nonlinear impacts of individual features and their mutual interaction effects on the model's predictive outcomes—both at the aggregate (average) level and the disaggregate (individual sample) level. Specifically, they empirically validate that nonlinear relationships and feature interactions are not merely marginal but exert a significant influence on the prediction of concrete compressive strength. To further quantify the contribution of each feature and improve the interpretability of the model, the next section employs SHapley Additive exPlanations (SHAP) to provide a more detailed, instance-level interpretation of feature importance.

4.3. Feature Importance

SHAP (Shapley Additive exPlanations) [65–67] is a game-theoretic approach for interpreting machine learning model predictions. Unlike model-specific or permutation-based feature importance methods [68], SHAP provides a theoretically grounded framework for fairly allocating feature contributions while ensuring local interpretability of each prediction. By extending classical Shapley values from cooperative game theory, SHAP offers an optimal and consistent way to quantify how much each feature contributes to a model's output relative to a baseline (the average model prediction). A positive SHAP value indicates that the feature increases the prediction relative to the base value, while a negative SHAP value indicates a decrease. The absolute SHAP value reflects the magnitude of a feature's influence. This method has been widely applied across disciplines—including model development, interpretability research, and risk analysis—because of its transparency and consistency [69–73].

As shown in Figure 7, the horizontal axis represents the SHAP values. Figure 7a shows the average absolute SHAP value of each feature, illustrating their overall importance in predicting concrete compressive strength. Age and Cement are the most influential variables, while Fly Ash has the least impact. Figure 7b provides a complementary view to Figure 6, visualizing both the magnitude and direction of each feature's effect on predictions. Each point represents a test sample: red indicates a feature value that increases the predicted strength, while blue indicates a feature value that decreases it. For example, higher values of Cement, Blast Furnace Slag, and Superplasticizer correspond to higher predicted compressive strength, whereas higher Fine or Coarse Aggregate values lower it. These results are consistent with the findings in Figure 6.



(**a**) (**b**)

Figure 7. Global importance of features on the test set. (**a**) Feature Importance ranking. (**b**) Visualization of features perturbation effects.

Figure 8 further illustrates the relationship between each feature and its SHAP value. In each subplot, the horizontal axis represents the primary feature, the left vertical axis corresponds to its SHAP value, and the color indicates the most interactive secondary feature. The following conclusions can be drawn:

- The scatter plots demonstrate not only how individual features contribute to model predictions across their value ranges but also hint at potential interaction patterns between key predictors. For instance, Cement and Age each exhibit an approximately linear positive relationship with SHAP values, indicating that higher values of these features typically enhance predicted compressive strength. Notably, the consistent directionality of their SHAP value trends suggests a synergistic interaction: the combined effect of increasing both Cement content and Age may amplify the positive impact on compressive strength beyond their individual contributions.
- The color distribution within each plot exhibits no distinct stratification, which underscores that the inter-feature interaction effects governing concrete compressive

strength are inherently complex and nonlinear, rather than mere simple additive adjustments. This lack of clear color separation reflects the non-separable nature of feature interactions: the combined influence of multiple predictors on the model's output cannot be decomposed into independent individual contributions, but instead arises from their mutual modulation in a nonlinear manner.

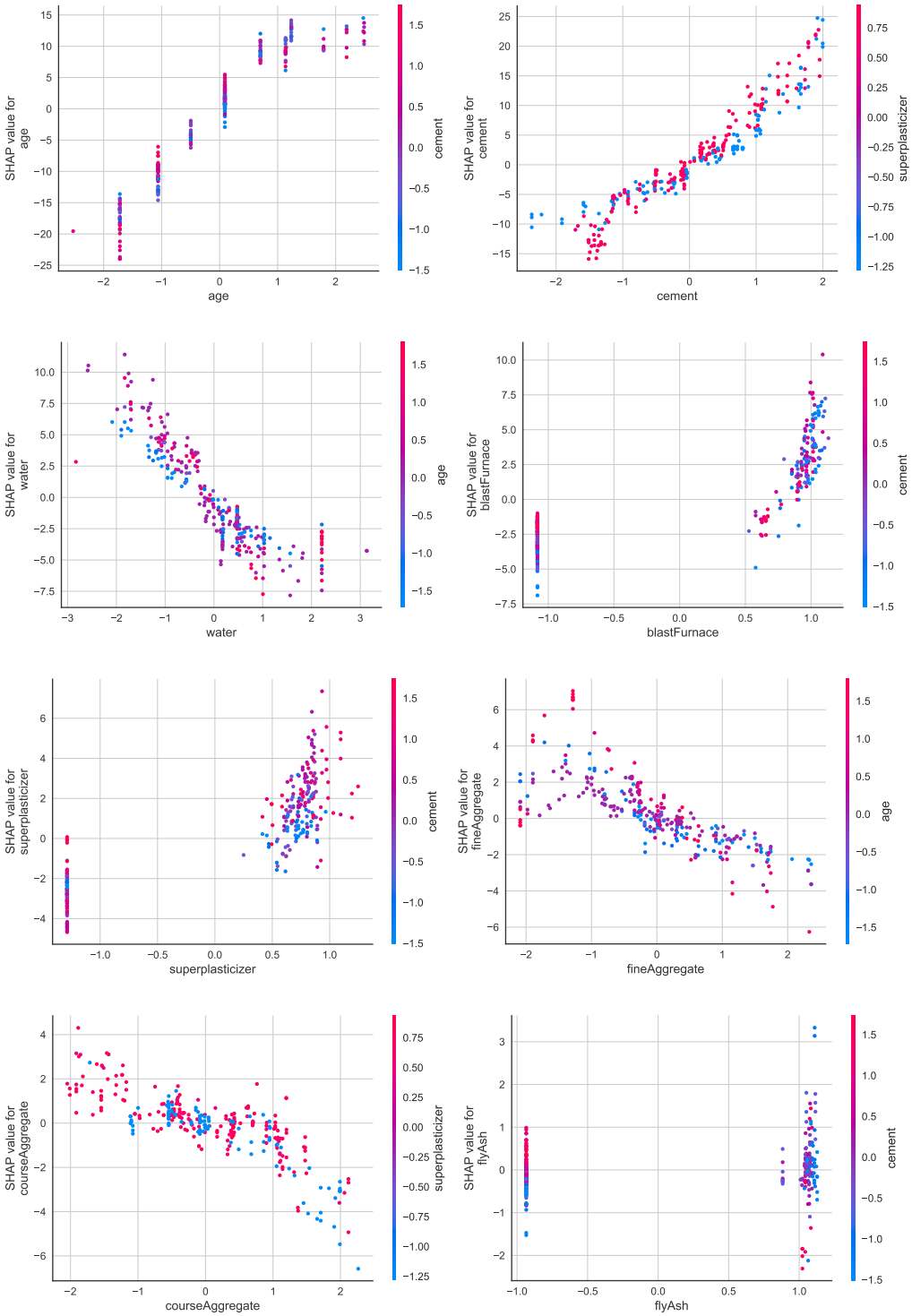


Figure 8. SHAP value scatter plot of features.

Overall, the SHAP analysis provides a transparent interpretation of the model's decision-making process. It confirms that Age and Cement are the dominant factors influencing Concrete Compressive Strength, while other features such as Blast Furnace Slag

and Superplasticizer contribute nonlinearly. These findings not only validate the model's predictive logic but also offer valuable insights for optimizing concrete mix design and improving material performance through interpretable machine learning.

Feature Contribution from the Sample Perspective

SHAP theory quantifies the contribution of each feature to a model's prediction by assigning a SHAP value to every sample. This value reflects both the magnitude and direction of the feature's influence on the prediction outcome.

SHAP force plots provide a compact and intuitive visualization of how individual features contribute to a specific prediction. In each force plot, a central dividing line represents the base value of the prediction, corresponding to the model's average output over the test set. Features are represented by colored bars:

- Red bars (positive SHAP values) act as pushing forces, driving the prediction above the base value.
- Blue bars (negative SHAP values) act as pulling forces, lowering the prediction below the base value.

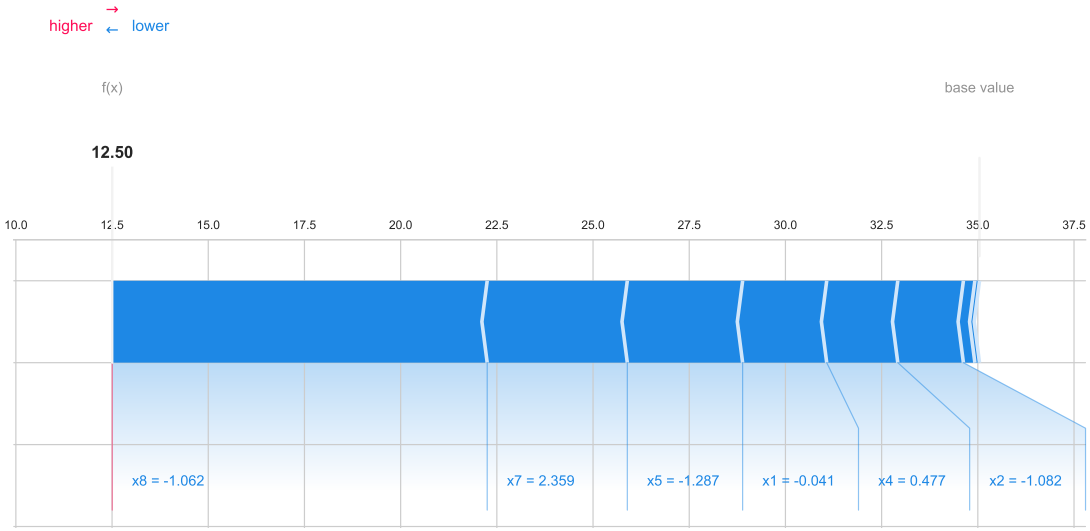
The width of each bar indicates the magnitude of the SHAP value. The final prediction, denoted as $f(x)$, equals the base value plus the sum of all SHAP values for that instance. Actual feature values are shown below each bar, illustrating each feature's quantitative contribution to the deviation from the base value.

To illustrate how SHAP values influence the final prediction, Figure 9 presents SHAP force plots for three instances from the test set at the 25th, 50th, and 75th percentiles. Table 5 lists the predicted, SHAP, and base values for these samples. The predicted value for each sample is obtained by summing its SHAP values with the base value. For the 25th percentile sample, all SHAP values are negative, meaning every feature contributes to decreasing the prediction relative to the base value. In contrast, for the 50th percentile sample, all features except x_5 reduce the base value, while for the 75th percentile sample, all features except x_2 and x_6 increase the prediction above it. Across all three examples, we examined how input feature values affect their corresponding SHAP values. These SHAP components quantitatively determine how the final predicted output deviates from the baseline. SHAP therefore provides a concise and interpretable description of the complex relationships between features. The force plots visually decompose each feature's contribution, clearly illustrating whether a feature pushes or pulls the model's prediction.

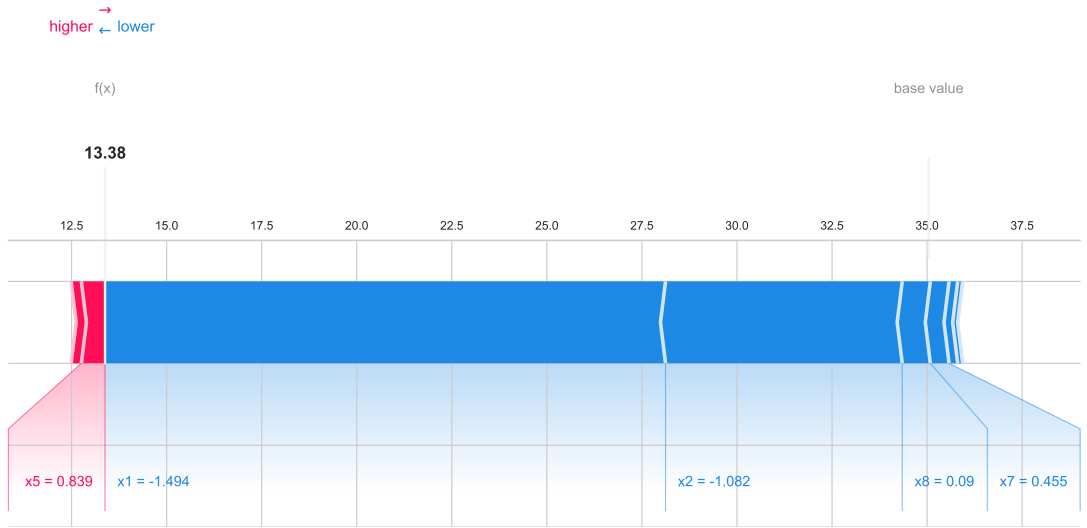
Table 5. The sample's prediction, SHAP value, and base value.

Sample	$f(x)$	SHAP of x_1	SHAP of x_2	SHAP of x_3	SHAP of x_4	SHAP of x_5	SHAP of x_6	SHAP of x_7	SHAP of x_8	Base Value
25%	12.50	−2.185	−1.705	−0.298	−1.847	−2.997	−0.120	−3.641	−9.745	35.043
50%	13.38	−14.741	−6.225	0.269	−0.227	0.622	−0.130	−0.492	−0.739	35.043
75%	66.58	20.464	−1.234	0.548	3.209	−0.086	−3.083	4.241	7.472	35.043

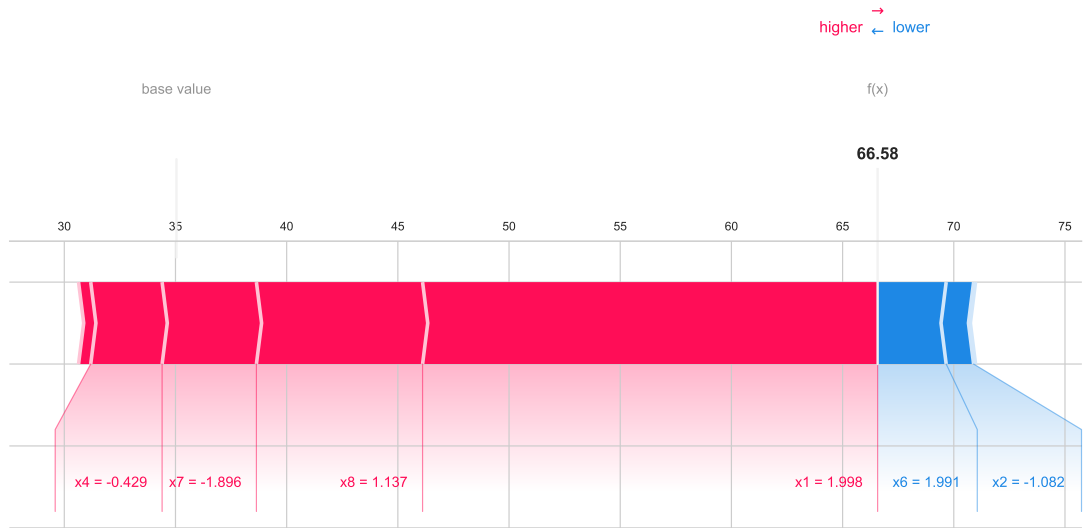
Such feature-level transparency is crucial for both model validation and practical engineering applications. By revealing how each input variable influences the predicted concrete compressive strength, SHAP analysis enhances the interpretability and reliability of machine learning models. This level of insight allows engineers to better understand material behavior, optimize mixture designs, and make data-driven decisions in high-performance concrete development.



(**a**) Sample 25%



(**b**) Sample 50%



(**c**) Sample 75%

Figure 9. Force plot of SHAP.

5. Conclusions

This study examines the influence of admixtures on the Concrete Compressive Strength of High-Performance Concrete (HPC). It compares statistical, machine learning, and other regression models by analyzing the relationships among key features and constructing predictive models. The purpose of those models is to establish a solid baseline across traditional and modern regression paradigms. To enhance understanding of feature contributions and model behavior, the study also provides visual interpretations of the regression models.

First, a multi-dimensional feature correlation analysis framework was established to systematically investigate the eight key factors affecting concrete strength using a combination of analytical techniques, including box plots, Pearson correlation coefficient matrices, and pairwise feature analysis. Box plots were used to visualize the distribution characteristics of each factor across different strength intervals, providing statistical insights into their behavior. The Pearson correlation coefficient matrix quantified the strength and direction of linear relationships between factors, revealing potential multi-factor synergistic effects. Pairwise feature analysis was performed to visualize inter-feature relationships across the dataset, offering a comprehensive understanding of the overall data structure and the interactions among variables. By integrating statistical visualization with quantitative analysis, this approach not only validates the influence patterns of key parameters established in classical concrete theory but also reveals nonlinear behavioral characteristics arising from the coupled effects of multiple factors.

Second, this study develops a machine learning-driven framework for predicting the Concrete Compressive Strength of HPC. The main findings are as follows:

(1) A systematic benchmark evaluation was performed across 18 machine learning algorithms. The results show that models such as Stacking Ensemble-1, XGBoost regression, and HistGradientBoosting regression consistently achieve optimal performance on both raw and transformed data, exhibiting strong generalization capability. Their predictive accuracy is substantially higher than that of traditional statistical methods such as linear regression.
(2) Most models run extremely efficiently. However, the performance of stacking ensemble models is limited by their base learners. While they achieve higher predictive accuracy, they are considerably more time-consuming.
(3) Tree-based and boosting methods demonstrate a superior ability to capture nonlinear relationships among features, leading to substantially improved predictive performance.
(4) Although neural network methods are computationally expensive, it should be noted that traditional machine learning algorithms face a bottleneck in further improving accuracy. In contrast, deep learning models have the potential to achieve superior precision with appropriate architecture design and hyperparameter tuning.

Finally, taking the XGBoost model as an example, the learning outcomes were visually interpreted:

(1) PDP and ICE plots illustrate the influence of individual features on prediction outcomes, revealing different dependency types such as strong marginal effects, interaction-driven patterns, and complex nonlinear relationships.
(2) Using SHAP values for feature importance ranking provides a prioritized list of influential features and analyzes variations in the marginal contributions of factors across different strength intervals. It also quantitatively assesses both the magnitude and direction of each feature's impact on model predictions.
(3) SHAP force plots visually represent the magnitude and sign of SHAP values, clearly demonstrating each feature's contribution to the model's prediction.

In conclusion, regression modeling for complex materials requires approaches capable of capturing detailed feature correlations while remaining interpretable, to support real engineering applications and promote data-driven decision-making. However, this study did not explore deep learning methodologies in depth. Moreover, we used only the original eight features for analysis to maintain methodological consistency and integrity. The feature analysis does not account for proportional factors, such as the Water/Binder ratio, Fly Ash/Binder ratio, and other derived parameters discussed in Yeh et al. [2] or specific characteristics listed in Table 1 by Bolbolvand et al. [74]. Future research should aim to enhance the model's ability to capture underlying physical relationships and incorporate mechanical property evaluations, including mixture proportion indicators such as the water/binder and ash/binder ratios, to achieve a more comprehensive understanding of the influencing factors.

Author Contributions: Conceptualization, M.-L.T. and J.Y.; Methodology, L.Z.; Software, C.Q.; Formal analysis, C.Q. and M.-L.T.; Investigation, L.Z., C.Q. and M.-L.T.; Resources, J.Y.; Data curation, C.Q.; Writing—original draft, L.Z.; Visualization, L.Z.; Supervision, J.Y.; Project administration, J.Y.; Funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Scientific research project of the Education Department of Jilin Province (JJKH20250299BS), the National Natural Science Foundation of China (Nos. 12292980 and 12292984), the National Key R & D Program of China (Nos. 2023YFA1009000, 2023YFA1009004, 2020YFA0712203 and 2020YFA0712201), Key Program of National Natural Science Foundation of China (NSFC12031016), Tianyuan Foundation of National Natural Science Foundation of China (NSFC12426529), Beijing Natural Science Foundation (BNSF-Z210003), the Department of Science, Technology and Information of the Ministry of Education (No. 8091B042240). Chang Qu's research was supported by STU Scientific Research Initiation Grant (No. NTF25027T). Man-Lai TANG's research was supported by Research Matching Grant (project: 700006 Applications of SAS Viya in Big Data Analytics) from the Research Grants Council of the Hong Kong Special Administration Region, and the Big Data Intelligence Centre in The Hang Seng University of Hong Kong.

Data Availability Statement: The data presented in this study are openly available in the UC Irvine Machine Learning Repository at https://www.openml.org/search?type=data&sort=runs&id=44959&status=active, accessed on 7 August 2025.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

1. Yeh, I.C. Modeling of strength of high-performance concrete using artificial neural networks. *Cem. Concr. Res.* **1998**, *28*, 1797–1808. [CrossRef]
2. Yeh, I. Modeling slump of concrete with fly ash and superplasticizer. *Comput. Concr.* **2008**, *5*, 559–572. [CrossRef]
3. Chou, J.S.; Pham, A.D. Smart artificial firefly colony algorithm-based support vector regression for enhanced forecasting in civil engineering. *Comput.-Aided Civ. Infrastruct. Eng.* **2015**, *30*, 715–732. [CrossRef]
4. Mistry, M.; Letsios, D.; Krennrich, G.; Lee, R.M.; Misener, R. Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. *INFORMS J. Comput.* **2021**, *33*, 1103–1119. [CrossRef]
5. Kamolov, S. Machine learning methods in analysis of concrete: A state-of-the-art review. *Ann. Math. Comput. Sci.* **2023**, *18*, 104–115. [CrossRef]
6. Cheung, R.C.; Aue, A.; Lee, T.C. Consistent estimation for partition-wise regression and classification models. *IEEE Trans. Signal Process.* **2017**, *65*, 3662–3674. [CrossRef]
7. Hamet, P.; Tremblay, J. Artificial intelligence in medicine. *Metabolism* **2017**, *69*, S36–S40. [CrossRef]
8. Hannah, L.A.; Blei, D.M.; Powell, W.B. Dirichlet process mixtures of generalized linear models. *J. Mach. Learn. Res.* **2011**, *12*, 1923–1953. [CrossRef]
9. Meinshausen, N. Forest garrote. *Electron. J. Stat.* **2009**, *3*, 1288–1304. [CrossRef]

10. Spjuth, O.; Brännström, R.C.; Carlsson, L.; Gauraha, N. Combining prediction intervals on multi-source non-disclosed regression datasets. In Proceedings of the Conformal and Probabilistic Prediction and Applications, Gold Sands, Bulgaria, 8–11 September 2019; pp. 53–65. [CrossRef]
11. Wang, K.; Zhang, J.; Chen, Y.; Zhong, P. Least absolute deviation support vector regression. *Math. Probl. Eng.* **2014**, *2014*, 169575. [CrossRef]
12. Yeh, I.C. Analysis of strength of concrete using design of experiments and neural networks. *J. Mater. Civ. Eng.* **2006**, *18*, 597–604. [CrossRef]
13. Žegklitz, J.; Pošík, P. Learning Linear Feature Space Transformations in Symbolic Regression. *arXiv* **2017**, arXiv:1704.05134. [CrossRef]
14. Antonelli, M.; Ducange, P.; Lazzerini, B.; Marcelloni, F. Learning concurrently partition granularities and rule bases of Mamdani fuzzy systems in a multi-objective evolutionary framework. *Int. J. Approx. Reason.* **2009**, *50*, 1066–1080. [CrossRef]
15. Lin, S.; Wang, Y.; Xu, L. Re-scale boosting for regression and classification. *arXiv* **2015**, arXiv:1505.01371. [CrossRef]
16. Transtrum, M.K.; Sethna, J.P. Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization. *arXiv* **2012**, arXiv:1201.5885. [CrossRef]
17. Chen, C.; Mac Parthaláin, N.; Li, Y.; Price, C.; Quek, C.; Shen, Q. Rough-fuzzy rule interpolation. *Inf. Sci.* **2016**, *351*, 1–17. [CrossRef]
18. Chou, J.S.; Pham, A.D. Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. *Constr. Build. Mater.* **2013**, *49*, 554–563. [CrossRef]
19. Filippone, M.; Engler, R. Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE). In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1015–1024.
20. Geifman, Y.; El-Yaniv, R. Selectivenet: A deep neural network with an integrated reject option. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 2151–2159.
21. Chou, J.S.; Tsai, C.F.; Pham, A.D.; Lu, Y.H. Machine learning in concrete strength simulations: Multi-nation data analytics. *Constr. Build. Mater.* **2014**, *73*, 771–780. [CrossRef]
22. Chou, J.S.; Chiu, C.K.; Farfoura, M.; Al-Taharwa, I. Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. *J. Comput. Civ. Eng.* **2011**, *25*, 242–253. [CrossRef]
23. Erdal, H.I.; Karakurt, O.; Namli, E. High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1246–1254. [CrossRef]
24. Feng, D.C.; Liu, Z.T.; Wang, X.D.; Chen, Y.; Chang, J.Q.; Wei, D.F.; Jiang, Z.M. Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Constr. Build. Mater.* **2020**, *230*, 117000. [CrossRef]
25. Nguyen-Sy, T.; Wakim, J.; To, Q.D.; Vu, M.N.; Nguyen, T.D.; Nguyen, T.T. Predicting the compressive strength of concrete from its compositions and age using the extreme gradient boosting method. *Constr. Build. Mater.* **2020**, *260*, 119757. [CrossRef]
26. Zeng, Z.; Zhu, Z.; Yao, W.; Wang, Z.; Wang, C.; Wei, Y.; Wei, Z.; Guan, X. Accurate prediction of concrete compressive strength based on explainable features using deep learning. *Constr. Build. Mater.* **2022**, *329*, 127082. [CrossRef]
27. Sadegh-Zadeh, S.A.; Dastmard, A.; Montazeri Kafshgarkolaei, L.; Movahedi, S.; Shiry Ghidary, S.; Najafi, A.; Saadat, M. Machine learning modelling for compressive strength prediction of superplasticizer-based concrete. *Infrastructures* **2023**, *8*, 21. [CrossRef]
28. Pourbaba, M.; Chakraborty, R.; Pourbaba, M.; Belarbi, A.; Yeon, J.H. A new insight into the design compressive strength of ultra-high performance concrete. *Buildings* **2023**, *13*, 2909. [CrossRef]
29. Thapa, J. Concrete compressive strength prediction by artificial neural network approach. *J. Eng. Issues Solut.* **2024**, *3*, 76–90. [CrossRef]
30. Palanisamy, A.K.; Jegatheeswaran, D.; Ananth, C.; Bhogayata, A.; Juneja, B.; Buribay, R.; Mezgebe, T.T. Prediction of early age compressive strength of concrete using machine learning. *Sci. Rep.* **2025**. [CrossRef]
31. Kumar, S.; Prakash, S.; Gaur, C.; Fida, H. Concrete Compressive Strength Prediction: A Data-Driven Approach. In Proceedings of the International Conference on Innovative Computing and Communication, Delhi, India, 14–15 February 2025; pp. 1–15. [CrossRef]
32. Lee, D.K. Data transformation: A focus on the interpretation. *Korean J. Anesthesiol.* **2020**, *73*, 503–508. [CrossRef] [PubMed]
33. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2022. Available online: https://oreilly.com/catalog/errata.csp?isbn=9781098125974 (accessed on 15 September 2025).
34. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
35. Zhou, Z.H. *Machine Learning*; Springer Nature: Basel, Switzerland, 2021. [CrossRef]
36. Rifkin, R.M.; Lippert, R.A. Notes on Regularized Least Squares. 2007. Available online: http://hdl.handle.net/1721.1/37318 (accessed on 15 September 2025).
37. Zou, H.; Hastie, T.; Tibshirani, R. On the degrees of freedom of the LASSO. *Ann. Stat.* **2007**, *35*, 1198–1232. [CrossRef]

38. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
39. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [CrossRef]
40. Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least angle regression. *Ann. Stat.* **2004**, *32*, 407–499. [CrossRef]
41. Williams, C.; Rasmussen, C. Gaussian processes for regression. *Adv. Neural Inf. Process. Syst.* **1995**, *8*. Available online: https://gaussianprocess.org/gpml (accessed on 15 September 2025).
42. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [CrossRef]
43. Kramer, O. K-nearest neighbors. In *Dimensionality Reduction with Unsupervised Nearest Neighbors*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 13–23. [CrossRef]
44. Song, Y.; Liang, J.; Lu, J.; Zhao, X. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing* **2017**, *251*, 26–34. [CrossRef]
45. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
46. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
47. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2002**, *2*, 18–22. Available online: http://CRAN.R-project.org/doc/Rnews/ (accessed on 15 September 2025).
48. Loh, W.Y. Classification and regression trees. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 14–23. [CrossRef]
49. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
50. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014. [CrossRef]
51. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
52. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, 1189–1232. [CrossRef]
53. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
54. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html (accessed on 15 September 2025).
55. Guryanov, A. Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees. In Proceedings of the International Conference on Analysis of Images, Social Networks and Texts, Kazan, Russia, 17–19 July 2019; pp. 39–50. [CrossRef]
56. Jiang, J.; Cui, B.; Zhang, C.; Fu, F. Dimboost: Boosting gradient boosting decision tree to higher dimensions. In Proceedings of the 2018 International Conference on Management of Data, Houston, TX, USA, 10–15 June 2018; pp. 1363–1376. [CrossRef]
57. Gurney, K. *An Introduction to Neural Networks*; CRC Press: Boca Raton, FL, USA, 2018. [CrossRef]
58. Murtagh, F. Multilayer perceptrons for classification and regression. *Neurocomputing* **1991**, *2*, 183–197. [CrossRef]
59. Rizkallah, L.W. Enhancing the performance of gradient boosting trees on regression problems. *J. Big Data* **2025**, *12*, 35. [CrossRef]
60. Fisher, R.A. Statistical methods for research workers. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: Berlin/Heidelberg, Germany, 1970; pp. 66–70. [CrossRef]
61. Box, G.E.; Cox, D.R. An analysis of transformations. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1964**, *26*, 211–243. [CrossRef]
62. Sakia, R.M. The Box-Cox transformation technique: A review. *J. R. Stat. Soc. Ser. D: Stat.* **1992**, *41*, 169–178. [CrossRef]
63. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. [CrossRef]
64. Yeh, I.C. Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cem. Concr. Compos.* **2007**, *29*, 474–480. [CrossRef]
65. Kuhn, H.W.; Tucker, A.W. *Contributions to the Theory of Games*; Number 28; Princeton University Press: Princeton, NJ, USA, 1953. [CrossRef]
66. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
67. Štrumbelj, E.; Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **2014**, *41*, 647–665. [CrossRef]
68. Krishnan, N.A.; Kodamana, H.; Bhattoo, R. Interpretable Machine Learning. In *Machine Learning for Materials Discovery: Numerical Recipes and Practical Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 159–171. [CrossRef]
69. Ashkari, A.; Rezig, E.K. CausalExplain: Causal Explanations of Black-box Models with Training Data Subsets. In Proceedings of the Companion of the 2025 International Conference on Management of Data, Berlin, Germany, 22–27 June 2025; pp. 19–22. [CrossRef]
70. Covert, I.; Lee, S.I. Improving kernelshap: Practical shapley value estimation via linear regression. *arXiv* **2020**, arXiv:2012.01536. [CrossRef]

71. Dejl, A.; Zhang, D.; Ayoobi, H.; Williams, M.; Toni, F. Hidden conflicts in neural networks and their implications for explainability. In Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency, Athens, Greece, 23–26 June 2025; pp. 1498–1542. [CrossRef]
72. Heuss, M.; de Rijke, M.; Anand, A. RankingSHAP-Faithful Listwise Feature Attribution Explanations for Ranking Models. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, Padua, Italy, 13–18 July 2025; pp. 381–391. [CrossRef]
73. Tjoa, E.; Guan, C. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4793–4813. [CrossRef]
74. Bolbolvand, M.; Tavakkoli, S.M.; Alaee, F.J. Prediction of compressive and flexural strengths of ultra-high-performance concrete (UHPC) using machine learning for various fiber types. *Constr. Build. Mater.* **2025**, *493*, 143135. [CrossRef]