



Article

Advanced Machine Learning Techniques for Predicting Concrete Compressive Strength

Mohammad Saleh Nikoopayan Tak^{1,2}, Yanxiao Feng^{2,*} and Mohamed Mahgoub²

¹ School of Architecture, New Jersey Institute of Technology, Newark, NJ 07102, USA; mn552@njit.edu

² School of Applied Engineering and Technology, New Jersey Institute of Technology, Newark, NJ 07102, USA; mohamed.mahgoub@njit.edu

* Correspondence: yanxiao.feng@njit.edu

Abstract: Accurate estimation of concrete compressive strength is very important for the improvement of mix design, quality assurance, and compliance with engineering specifications. Most empirical traditional models have failed to capture the complex relationships inherent within varied constituents of concrete mixes. This paper develops a machine learning model for compressive strength prediction using mix design variables and curing age from a “Concrete Compressive Strength Dataset” obtained from the UCI Machine Learning Repository. After comprehensive data preprocessing and feature engineering, various regression and classification models were trained and evaluated, including gradient boosting, random forest, AdaBoost, k-nearest neighbors, linear regression, and neural networks. The gradient boosting regressor (GBR) achieved the highest predictive accuracy with an R^2 value of 0.94. Feature importance analysis showed that the water–cement ratio and age are the most crucial factors affecting compressive strength. Advanced methods such as SHapley Additive exPlanations (SHAP) values and partial dependence plots were used to attain deep insights about feature interaction with a view to enhancing interpretability and fostering trust in models. Results highlight the potential of machine learning models to improve concrete mix design with the aim of sustainable construction through the optimization of material usage and waste reduction. It is recommended that future research be undertaken with expanding datasets, more features, and richer feature engineering to enhance predictive power.



Academic Editors: Darius Bačinskas and Chris Goodier

Received: 23 November 2024

Revised: 16 January 2025

Accepted: 17 January 2025

Published: 21 January 2025

Citation: Nikoopayan Tak, M.S.; Feng, Y.; Mahgoub, M. Advanced Machine Learning Techniques for Predicting Concrete Compressive Strength. *Infrastructures* **2025**, *10*, 26. <https://doi.org/10.3390/infrastructures10020026>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: machine learning models; compressive strength prediction; feature importance analysis; SHAP values; mix design optimization; sustainable construction

1. Introduction

Concrete is the most frequently used construction material globally due to its versatility, durability, and cost-effectiveness [1]. Its mechanical properties, particularly compressive strength, are critical for ensuring the safety and longevity of structures. Accurate prediction of concrete’s compressive strength is essential for mix design optimization, quality control, and compliance with engineering standards [2]. Traditional empirical methods for estimating compressive strength often involve extensive laboratory testing and simplistic models that may not capture the complex interactions among the multitude of variables in concrete mixtures. This complexity has led researchers to explore advanced computational techniques, particularly machine learning (ML), to model and predict concrete behavior more accurately [3].

In recent years, ML algorithms have gained prominence in civil engineering applications due to their ability to model nonlinear relationships and handle large datasets. These algorithms learn patterns from historical data and can make accurate predictions based on

input features, which makes them suitable for predicting the properties of various concrete types, including those modified with Supplementary Materials such as fly ash, nano-silica, recycled aggregates, and other industrial by-products.

Several studies have applied ML models to predict concrete compressive strength with notable success. Alghrairi et al. [4] developed nine ML models to estimate the compressive strength of lightweight concrete modified with nanomaterials. Among these, the gradient-boosted trees (GBT) model outperformed others by achieving a coefficient of determination (R^2) of 0.90 and a root mean square error (RMSE) of 5.286 MPa. The study highlighted that water content was the most influential factor affecting compressive strength predictions and emphasized the critical role of the water-to-cement ratio in concrete mix design. Similarly, Ding et al. [5] investigated ML models to predict the compressive strength of alkali-activated cementitious materials using solid waste components. They employed six ML algorithms, including support vector machine (SVM), random forest (RF), radial basis function neural network (RBF), and long short-term memory network (LSTM). The SVM model achieved the highest performance with an R^2 of 0.9054 and a normalized root mean square error of 0.0997.

In addition to the evaluation of prediction accuracy, feature importance analysis using SHapley Additive exPlanations (SHAP) revealed key influencing factors such as calcium oxide content, water-to-binder ratio, silicon dioxide content, modulus of water glass, and aluminum trioxide content. Ekanayake et al. [6] addressed the “black-box” nature of ML models by employing SHAP to interpret predictions of concrete compressive strength. Utilizing tree-based algorithms including XGBoost and light gradient boosting machine (LGBM), they achieved high accuracy with an R-value of 0.98. The SHAP analysis provided insights into feature importance and confirmed that age and cement content were the most influential features. This approach demonstrated that ML models could capture complex relationships among variables and lead to enhanced trust among domain experts.

Despite these advancements, a persistent limitation in the existing literature is the inadequate exploration of feature interactions and their cumulative impact on model predictions. Most studies emphasize achieving high predictive accuracy without thoroughly investigating how input variables interact within the models. For instance, Paudel et al. [7] compared the performance of non-ensemble and ensemble ML models in predicting the compressive strength of concrete containing fly ash. The study identified age, cement content, and water content as the most influential features but lacked a comprehensive analysis of feature interactions. Similarly, Song et al. [8] employed ML algorithms, including gene expression programming (GEP), artificial neural network (ANN), decision tree (DT), and bagging regressor, to predict the compressive strength of concrete with fly ash admixture. While the study confirmed that the selection of input parameters and regressors significantly affects the accuracy of predicted outcomes, it did not extensively explore feature interactions. Tran et al. [9] evaluated the compressive strength of concrete made with recycled concrete aggregates using six ML models. The GB_PSO model achieved the highest prediction accuracy with an R^2 of 0.9356. Feature importance analysis revealed that cement content and water content were the most important factors affecting compressive strength. However, the study primarily focused on individual feature importance rather than the interactions between variables. Ahmad et al. [10] compared supervised ML algorithms, including ANN, AdaBoost, and boosting, to predict the compressive strength of geopolymers containing high-calcium fly ash. This study demonstrated the potential of ensemble methods in capturing complex patterns in data, which can lead to more accurate predictions. Nevertheless, it did not explore the interactions among input features. Anjum et al. [11] applied ensemble ML methods, including gradient boosting, RF, bagging regressor, and AdaBoost regressor, to estimate the compressive strength of fiber-reinforced nano-silica modified concrete. SHAP analysis revealed that the coarse

aggregate to fine aggregate ratio had a stronger negative correlation with compressive strength, while specimen age positively affected it. The study highlighted the importance of considering the interaction and effects of input parameters but did not provide a detailed feature interaction analysis. Ullah et al. [12] predicted the compressive strength of sustainable foam concrete using individual and ensemble ML approaches, including SVM, RF, bagging, boosting, and a modified ensemble learner. The study suggested that ensemble learners significantly enhance the performance and robustness of ML models but did not explore feature interactions in depth. Moreover, Kumar and Pratap [13] investigated the use of ML models to predict the compressive strength of high-strength concrete and focused on the influence of superplasticizer, sand, and water content. The study acknowledged the significant influence of superplasticizer on compressive strength but lacked a comprehensive analysis of feature interactions. Nguyen et al. [14] proposed a machine learning approach using multivariate polynomial regression and automated feature engineering to predict the compressive strength of ultra-high-performance concrete (UHPC). While this study provided insights into feature interactions, it was specific to UHPC and did not address broader concrete types.

These studies collectively demonstrate that while ML models can achieve high accuracy in predicting concrete compressive strength, they often lack interpretability due to insufficient analysis of feature interactions. Most focus on individual feature importance without exploring how variables interact within the model to influence predictions. This limitation hinders the practical application of ML models in concrete mix design optimization, as understanding the synergistic effects among key variables is crucial. To address this gap, there is a pressing need for research that not only leverages advanced ML models for predicting concrete properties but also provides a thorough analysis of feature interactions and their collective impact on model predictions. Such an approach would enhance the interpretability of the models, allow for more informed decision-making in mix design optimization, and promote the development of high-performance, durable, and sustainable concrete materials.

Recent research has also begun integrating advanced predictive modeling with sustainability considerations. For example, Ref. [15] developed an ANN-based approach for recycled aggregate concrete, offering high-accuracy compressive strength predictions and practical closed-form solutions. In a related study, Ref. [16] examined ultra-high-performance lightweight concrete incorporating rice husk ash, applying life cycle assessment (LCA) to evaluate the environmental performance alongside compressive strength. Similarly, Ref. [17] employed multiple AI and optimization techniques to investigate interactions between fly ash content, mechanical properties, and environmental impact, thereby informing multi-objective optimization of sustainable concrete mixes. These contributions underscore a growing emphasis on not only predicting performance but also considering environmental implications. Nevertheless, even with these advancements, a persistent gap remains in the literature: the need for a more thorough exploration of feature interactions and their collective influence on model predictions. Addressing this gap is crucial for both interpretability and practical utility in concrete mix design.

Unlike prior work that predominantly focuses on predictive accuracy, our approach not only aims to achieve high accuracy but also provides in-depth interpretability by examining feature interactions using SHAP and partial dependence plots. This dual focus on accuracy and interpretability represents a key advancement over current methodologies to enable more informed decision-making in concrete mix design. This study aims to fulfill this need by developing machine learning models capable of predicting the compressive strength of various concrete types, including diverse input variables related to mix composition. By employing advanced feature importance analysis methods such as SHAP and interaction effects such as partial dependence plots, we investigate the interactions among these input

variables and their collective impact on compressive strength predictions. Additionally, we classify concrete samples into predefined strength categories closely aligned with industry standards and thresholds defined by the American Concrete Institute (ACI) [18] to make our models more applicable for industry uses that may require knowledge of the concrete class rather than the exact strength value.

This study is guided by several key research questions. First, it aims to explore how effectively machine learning models can predict concrete compressive strength using mix design parameters and curing age, while also examining how input variables interact within these models to influence the predictions. Additionally, the study investigates whether advanced feature importance analysis techniques, such as SHAP values, can enhance the interpretability of machine learning models in concrete strength prediction by revealing feature interactions and their impact on model outputs. Finally, the research seeks to determine how accurately machine learning models can classify concrete samples into predefined strength categories.

To answer these questions, the research follows a multi-step process that includes comprehensive data preprocessing to address missing values, outliers, and inconsistencies, followed by exploratory data analysis (EDA) to uncover patterns and relationships within the data. Feature selection techniques are employed to identify the most relevant variables affecting concrete strength to enhance model performance and interpretability. A range of machine learning algorithms, including regression models and classification models for strength categorization, are trained and evaluated using performance metrics such as accuracy and mean squared error. By integrating advanced feature interaction analysis into ML models for concrete strength prediction, this study contributes to the advancement of data-driven approaches in concrete technology. The findings are expected to provide valuable insights for optimizing mix designs and ensuring quality control in the construction industry.

2. Materials and Methods

This study employed a comprehensive methodology to analyze and predict the compressive strength of concrete using various machine learning models. The research process, as illustrated in Figure 1, involved data collection, data preprocessing, exploratory data analysis, feature engineering, and the development and evaluation of multiple regression and classification models. The aim was to identify the most effective predictive models and understand the underlying factors influencing concrete strength through the application of machine learning techniques and feature interaction analysis.

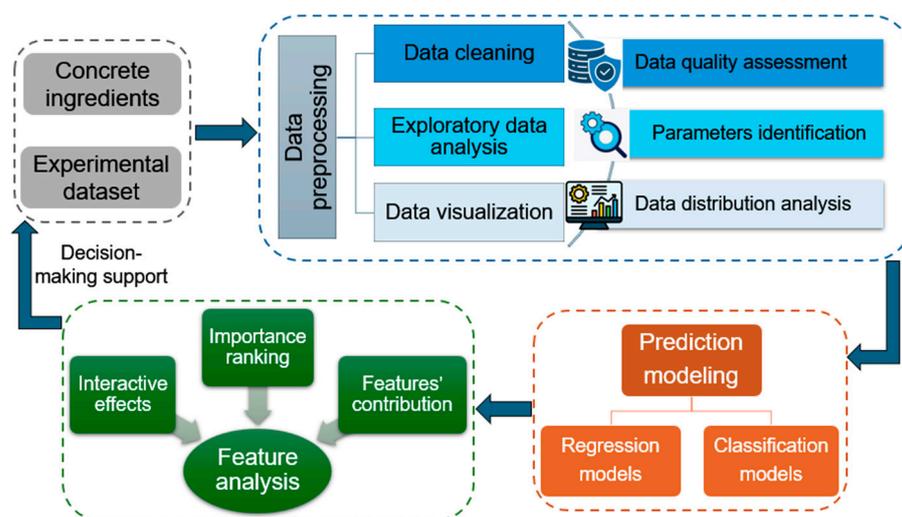


Figure 1. Framework for modeling analysis of concrete compressive strength.

2.1. Data Collection and Description

The study utilized the “Concrete Compressive Strength Dataset” from the UCI Machine Learning Repository, generously provided by Prof. I-Cheng Yeh [19]. The dataset comprises 1030 observations with nine variables, each representing a unique concrete mix design. The features include the quantities of different concrete components measured in kilograms per cubic meter (kg/m^3) and the age of the concrete in days. The target variable is the concrete compressive strength measured in megapascals (MPa). The dataset contains 1030 instances (rows) and 9 attributes (columns), where each row represents a concrete sample, and the columns correspond to the features described in Table 1.

Table 1. Variable description in the dataset.

Variable	Type	Unit	Description
cement	quantitative	kg/m^3	Input
blast furnace slag	quantitative	kg/m^3	Input
fly ash	quantitative	kg/m^3	Input
water	quantitative	kg/m^3	Input
superplasticizer	quantitative	kg/m^3	Input
coarse aggregate	quantitative	kg/m^3	Input
fine aggregate	quantitative	kg/m^3	Input
age	quantitative	Days	Input
compressive strength	quantitative	MPa	Output

2.2. Data Preprocessing

The data preprocessing phase was critical to ensure data quality and prepare the dataset for modeling. It involved data cleaning, exploratory data analysis, handling of outliers, feature engineering, and data scaling.

2.2.1. Data Cleaning

The dataset was initially inspected for missing values, duplicates, and inconsistencies. Using Python 3.10.13 with the pandas library [20], it was confirmed that there were no missing values in any of the variables. Duplicate entries were identified using the duplicated() function, which revealed 25 duplicate rows. These duplicates were removed to ensure data quality, reducing the dataset to 1005 unique observations. Additionally, a preliminary analysis, as shown in Figure 2, indicated the existence of outliers in the dataset. To mitigate the potential impact of these outliers on model performance, they were identified and removed using the interquartile range (IQR) method [21]. The IQR was calculated as the difference between the 75th (Q3) and 25th (Q1) percentiles, and any data points lying below $Q1 - 1.5 \text{ IQR}$ or above $Q3 + 1.5 \text{ IQR}$ were considered outliers. Significant outliers were found in variables such as age, and these outliers were removed from the dataset to improve model accuracy and generalizability. After outlier removal, the final dataset consisted of 911 observations.

2.2.2. Exploratory Data Analysis

Exploratory data analysis was performed to assess the distribution and characteristics of the variables related to concrete strength. We followed the EDA practices outlined by [22], which emphasize visualizing data distributions using histograms and frequency plots to uncover potential skewness or anomalies. Histograms and frequency plots were generated using Matplotlib [23] to visualize these distributions distinctly (see Figure 3). The analysis revealed a wide range of cement content with a peak of around $160 \text{ kg}/\text{m}^3$, which suggests variability in mix designs used across different concrete samples. The distribution of water content was mostly centralized around $190 \text{ kg}/\text{m}^3$, which indicates a common

standard in water usage for these concrete mixtures. Most samples contained low amounts of blast furnace slag, with a significant peak at 0 kg/m³, which highlights its optional use in the mixtures. The majority of the data points were clustered at low superplasticizer content, with a significant number of observations showing zero usage, emphasizing its selective application depending on specific mix requirements. There was a significant spike in age at 28 days, which is commonly recognized as a standard curing time for testing concrete strength [24], although other ages were also represented to a lesser extent. The strength of concrete showed a normal distribution with a mean of around 35 MPa and illustrates the common range of strength encountered in typical concrete applications. This exploratory analysis provided a foundation for understanding the key characteristics of the dataset, which inform the subsequent predictive modeling efforts.

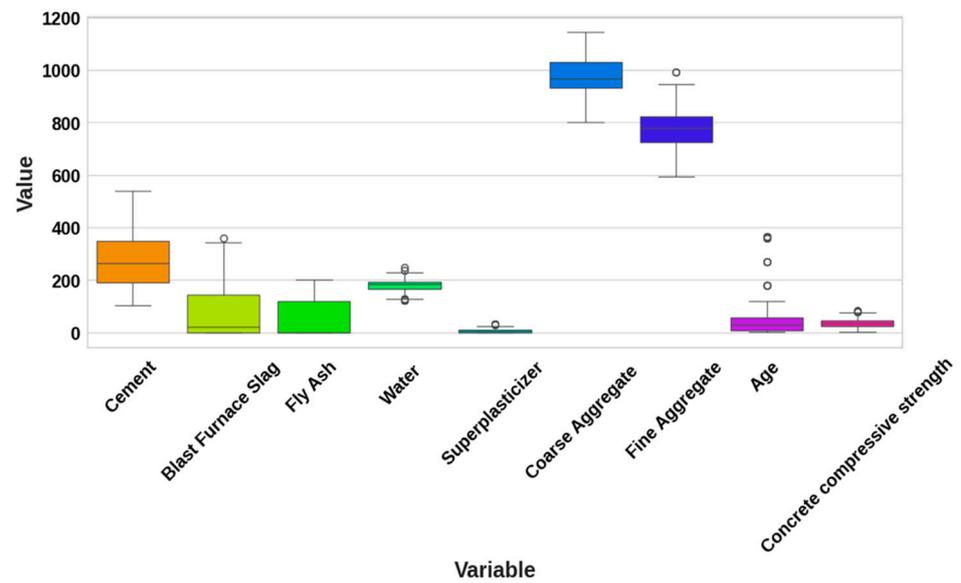


Figure 2. Distribution of concrete mix components and compressive strength.

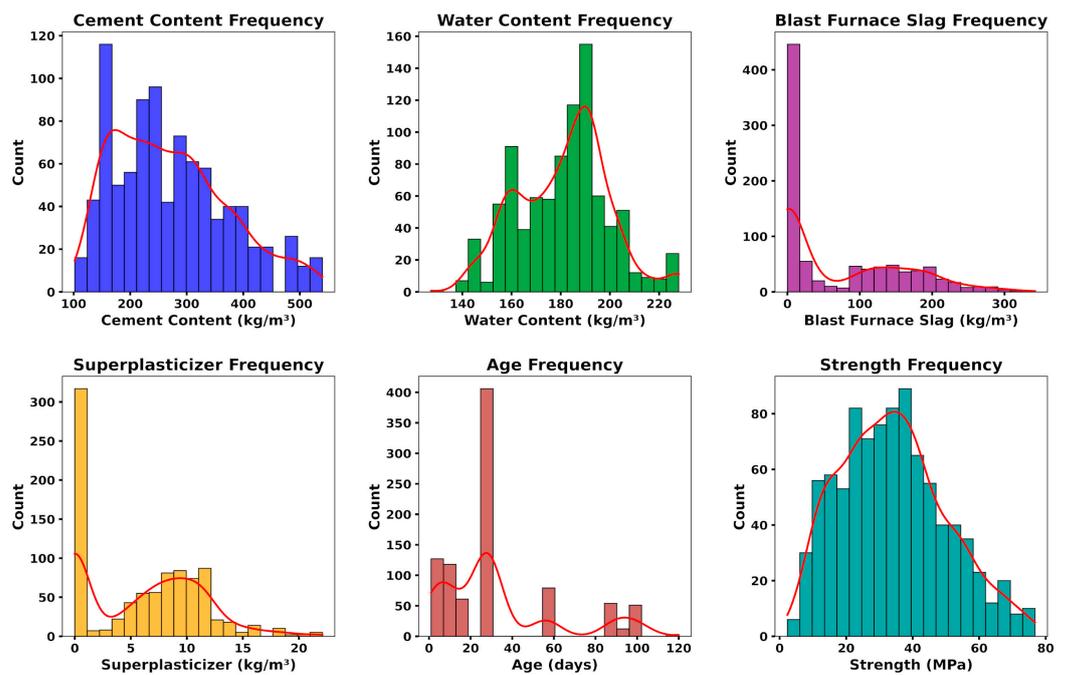


Figure 3. Concrete mix design attributes and their relationship with compressive strength. Red lines represent smoothed density curves for each histogram.

2.2.3. Correlation Analysis and Preparation of Predictor Variables

The Pearson correlation coefficient was calculated using Pandas [25] to identify the relationships between the input features and the target variable, compressive strength. A correlation matrix was visualized using the heatmap function from the Seaborn library [26] to illustrate these relationships (see Figure 4). The correlation matrix reveals a moderate positive correlation between cement content and compressive strength. This correlation indicates that increases in cement content are associated with increases in compressive strength, although the relationship is not exceptionally strong. Blast furnace slag and fly ash show moderate negative correlations with cement content. These findings suggest their use as partial cement replacements and imply that mixes with higher quantities of blast furnace slag and fly ash tend to have lower cement content. The data also reveal a strong negative correlation between water content and superplasticizer usage. This correlation emphasizes the role of superplasticizers in reducing water demand to maintain workability, thereby enhancing the concrete’s performance and durability. Moreover, a moderate positive correlation exists between superplasticizer usage and compressive strength. Interestingly, both coarse and fine aggregates display weak negative correlations with compressive strength, with R-values of -0.15 and -0.18 , respectively. Finally, concrete age shows a moderate positive correlation with compressive strength, indicated by an R-value of 0.52 . This relationship highlights the importance of the curing process, as the ongoing chemical reactions during this time enhance the concrete’s structural integrity and compressive capabilities.

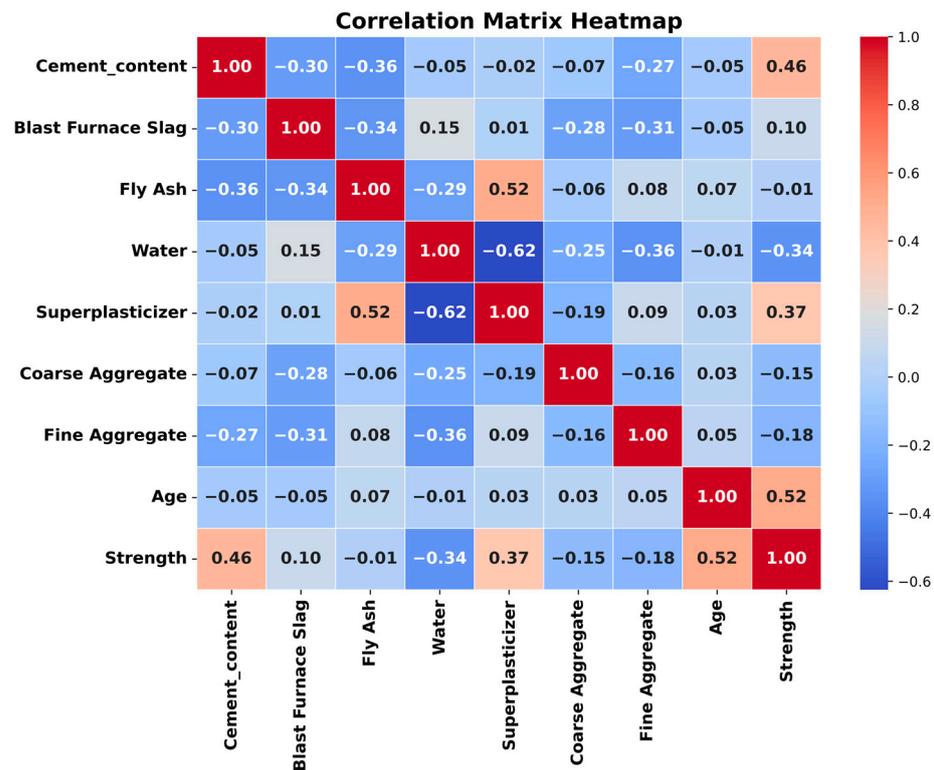


Figure 4. Correlation matrix between the input features and the target variable.

2.2.4. Feature Engineering and Multicollinearity Analysis

Multicollinearity among predictor variables can negatively impact the stability and interpretability of regression models by inflating the variance of coefficient estimates [27]. To quantify the degree of multicollinearity among the predictor variables, the variance inflation factor (VIF) was calculated using the variance_inflation_factor() function from statsmodels.stats.outliers_influence in Python. The VIF for each feature is computed as

VIF = $1/(1 - R^2)$, where R^2 is obtained by regressing that feature against all other features. The initial VIF analysis, presented in Figure 5a, revealed significant multicollinearity issues. Notably, the VIF values for water, coarse aggregate, fine aggregate, and cement were exceptionally high, with water exhibiting a VIF of 95.27, coarse aggregate at 84.71, fine aggregate at 76.82, and cement at 14.15. Such high VIF scores indicate that these variables are highly correlated with other predictors, which can destabilize regression models and obscure the true relationships between variables and the target outcome.

To mitigate multicollinearity and enhance the predictive power of the models, feature engineering was employed based on domain knowledge in concrete technology [28,29]. Two new features were created: the water–cement ratio (W/C ratio) and the coarse aggregate–fine aggregate ratio (C/F ratio). The W/C ratio was calculated by dividing the water content by the cement content. This ratio is a critical factor influencing concrete strength, as it affects the hydration process and the microstructure of the hardened concrete. A lower W/C ratio generally leads to higher strength and durability. The C/F ratio was determined by dividing the coarse aggregate content by the fine aggregate content. This ratio impacts the workability, compaction, and overall strength of concrete by influencing the particle packing and void content within the mix [30].

$$W/C \text{ Ratio} = \frac{\text{Water}}{\text{Cement}} \tag{1}$$

$$C/F \text{ Ratio} = \frac{\text{Coarse Aggregate}}{\text{Fine Aggregate}} \tag{2}$$

By transforming the original highly correlated variables into ratios, the absolute quantities, previously exhibiting high multicollinearity, were converted into relative measures that capture the essential proportional relationships in the concrete mix. This approach reduced redundancy among predictors while retaining the critical information necessary for accurate strength prediction. After feature engineering, the VIF was recalculated for the updated set of features. The results, shown in Figure 5b, indicated a substantial reduction in multicollinearity across the dataset. The VIF values for the newly engineered features were significantly lower, with the water–cement ratio at 10.24 and the coarse aggregate–fine aggregate ratio at 7.98. While these values are still above the commonly accepted threshold of 5, they represent a marked improvement from the initial VIF scores. These features were retained due to their significant practical importance and contribution to the predictive capability of the models. Other features also exhibited acceptable VIF values, all below the threshold of 5.

VIF Results for Input Feature Selection

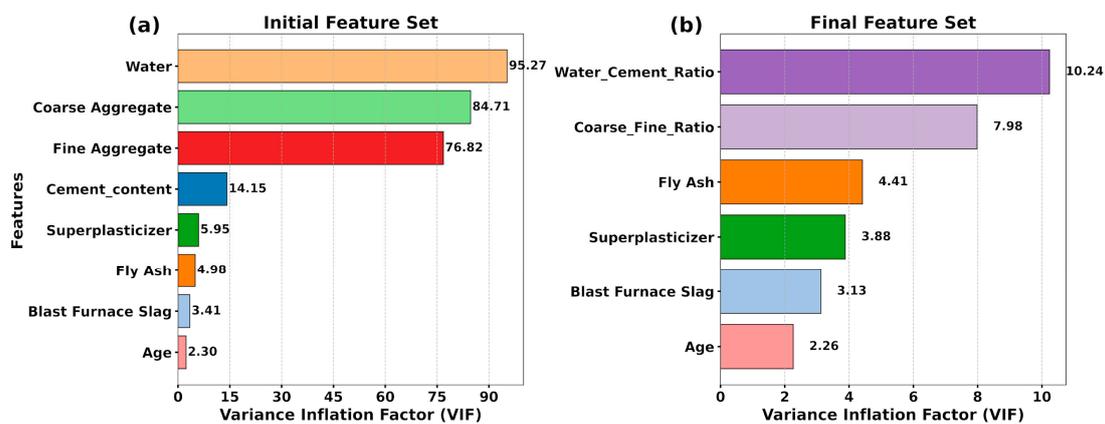


Figure 5. VIF results for input feature selection: (a) all initial features, (b) revised feature set.

2.2.5. Data Scaling

Machine learning algorithms, especially those involving gradient descent optimization, can be sensitive to the scale of the input features. To ensure all features contribute equally to the model training and to improve convergence, data scaling was performed using min–max normalization [31]. The MinMaxScaler from scikit-learn’s preprocessing module was applied to rescale all features to a range between 0 and 1.

2.2.6. Discretization of the Target Variable for Classification

Before applying classification techniques, the continuous target variable (compressive strength) was converted into categorical classes based on scales aligning closely with common practices in the construction industry [15]. The categories and their corresponding count are shown in Table 2. By assigning each concrete sample to one of these categories, the continuous numeric target values were transformed into discrete labels suitable for classification algorithms. This approach ensured that classifiers could effectively distinguish among these defined strength categories rather than attempting to predict a continuous value.

Table 2. Concrete compressive strength categories.

Strength Classification	Threshold (MPa)	Count
very high strength	≥ 60	62
high strength	[41, 59.99]	215
normal strength	[30, 40.99]	250
weak	[20, 29.99]	190
very weak	<20	194

2.3. Model Development and Evaluation

The core of the methodology involved developing and evaluating various machine learning models for both regression and classification tasks. The objective was to predict the compressive strength of concrete accurately and to classify concrete samples into predefined strength categories. Multiple machine learning models were developed and evaluated for these regression and classification tasks. The models considered are shown in Table 3. The dataset was split into training and testing sets using an 80–20 split with the train_test_split function from the scikit-learn library [32].

Table 3. Overview of machine learning models evaluated.

Regression Models	Classification Models
linear regression	RF classifier
k-nearest neighbors (KNN) regression	logistic regression
decision tree regression	SVM
RF regression	k-nearest neighbors (KNN) classifier
gradient boosting regression	bagging classifier
AdaBoost regression	
neural network	

An 80–20% training–testing split was selected to align with common machine learning practices for robust evaluation [33]. To ensure that the training and testing subsets share similar statistical characteristics, we first divided the target variable in the dataset into ten quantile-based bins (num_bins = 10) and then performed a stratified split. After this procedure, we computed descriptive statistics—record count, minimum, maximum, range,

mean, variance, and standard deviation—for each numeric feature. As presented in Table 4, the training and testing sets exhibited very similar statistics. Additionally, Kolmogorov–Smirnov tests [34] for each feature yielded high *p*-values (all > 0.05), which indicated no statistically significant differences between the distributions of the two subsets. These results confirmed that the testing set is representative of the training set and ensured that the performance metrics derived from the test set are both reliable and unbiased. The models were then trained on the training set and evaluated on the testing set.

Table 4. Descriptive statistics of features for training and testing sets.

	Feature	Number	Min	Max	Range	Mean	Variance	Std Dev
Training Set	Blast Furnace Slag	728	0	342.1	342.1	71.75	7453.08	86.33
	Fly Ash	728	0	200.1	200.1	59.92	4102.09	64.05
	Superplasticizer	728	0	22	22	6.06	27.27	5.22
	Age	728	1	120	119	31.86	792.56	28.15
	Water_Cement_Ratio	728	0.3	1.88	1.58	0.77	0.1	0.31
	Coarse_Fine_Ratio	728	0.92	1.87	0.95	1.28	0.03	0.18
Testing Set	Blast Furnace Slag	183	0	305.3	305.3	70.21	7331.27	85.62
	Fly Ash	183	0	195	195	59.97	4437.47	66.61
	Superplasticizer	183	0	22.1	22.1	5.88	28.08	5.3
	Age	183	3	120	117	33.15	871.41	29.52
	Water_Cement_Ratio	183	0.28	1.66	1.38	0.76	0.09	0.31
	Coarse_Fine_Ratio	183	0.94	1.84	0.89	1.26	0.03	0.17

2.3.1. Regression and Classification Models

Multiple regression models were developed to predict concrete compressive strength, using a range of techniques to capture both linear and non-linear relationships within the data. These models are shown below.

- Linear regression: This serves as a baseline model to establish a benchmark and assess the extent of linear relationships between the features and the target variable (compressive strength).
- Decision tree regression: This model is employed to capture non-linear relationships by partitioning the data based on feature thresholds, effectively creating a tree-like structure of decisions to arrive at a prediction.
- RF regression: This ensemble method combines multiple decision trees to improve predictive accuracy and mitigate overfitting, by leveraging the wisdom of the crowd for a more robust prediction.
- Gradient boosting regression: This technique builds models sequentially, with each subsequent model correcting errors made by previous ones. This iterative approach enhances performance, particularly on complex datasets with intricate patterns.
- AdaBoost regression: Similar to gradient boosting, AdaBoost focuses on instances where prior models struggled and adjusts weights accordingly to improve prediction accuracy on challenging data points.
- KNN regression: This model predicts target values based on the average of the nearest neighbors in the feature space and leverages the similarity between data points for prediction.
- Neural network model: A neural network model was implemented using TensorFlow [35] and Keras [36] to capture complex, non-linear relationships within the data. The architecture comprises an input layer, hidden layers, and output layers.

For the classification task, five classification models were developed and evaluated, including RF classifier, logistic regression, SVM, KNN classifier, and bagging classifier with

decision trees. Each classifier underwent hyperparameter tuning to optimize performance. Bayesian optimization (BayesSearchCV from skopt) was employed for all models, which iteratively refines the hyperparameter search space based on model parameters [37,38].

The hyperparameters considered for the regression and classification models are detailed in Table 5. Additional details on the default values, tuned values, and optimization processes employed are provided in Section 3.

Table 5. Hyperparameters considered for regression and classification models in this study.

Regression		Classification	
Model	Hyperparameters Considered	Model	Hyperparameters Considered
Linear Regression	None (used ordinary least squares)	Logistic Regression	penalty (l1, l2), C (regularization strength), solver (saga)
K-Nearest Neighbors	n_neighbors, metric, weights	Support Vector Machine	C (regularization), gamma (kernel coefficient), kernel (linear, rbf, poly, sigmoid), degree (if kernel = poly)
Decision Tree Regressor	max_depth, min_samples_split, min_samples_leaf	k-Nearest Neighbors	n_neighbors, weights (uniform, distance), p (distance metric: 1 = Manhattan, 2 = Euclidean)
Random Forest Regressor	n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features	Random Forest Classifier	n_estimators, max_depth, min_samples_split, max_features
Gradient Boosting Regressor	n_estimators, learning_rate, max_depth, subsample, min_samples_split	Bagging Classifier (with DT)	n_estimators, max_samples, max_features, bootstrap, bootstrap_features, estimator__max_depth, estimator__criterion (for DecisionTreeClassifier)
AdaBoost Regressor	n_estimators, learning_rate, base_estimator (DT max_depth)		
Neural Network (MLP)	Number of layers, units per layer, activation, dropout rate, batch size, epochs, optimizer, learning_rate, L2 regularization		

2.3.2. Model Evaluation Metrics

The performance of the developed models was assessed using appropriate evaluation metrics for both regression and classification tasks. For regression models, the mean squared error (MSE) and the coefficient of determination, known as R^2 , were employed to quantify the accuracy of the predictions. The MSE measures the average squared difference between the predicted values (\hat{y}_i) and the actual observed values (y_i). It is defined by Equation (3). The R^2 metric represents the proportion of variance in the dependent variable that is predictable from the independent variables. It is calculated using Equation (4).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{3}$$

where n is the number of observations. A lower MSE indicates that the model’s predictions are closer to the actual values, which signifies better predictive accuracy.

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

where \bar{y} is the mean of the observed data. An R^2 value closer to 1 indicates that a higher proportion of variance is explained by the model, and it reflects a better fit.

Additionally, to provide a more comprehensive and intuitive visual comparison of the regression models' performance, a Taylor diagram was employed. The Taylor diagram plots correlation (with the observed values), the ratio of the standard deviation of the model predictions to that of the observations, and the centered RMS error, all on a single polar coordinate plot [39]. This approach allows simultaneous evaluation of how well each model's variability and pattern of predictions match the observed data.

For classification models, accuracy was calculated to determine the overall effectiveness of the model in correctly predicting the class labels. It is given by Equation (5). However, in datasets with class imbalances, accuracy can be misleading because it may be biased towards the majority class. To address this, balanced accuracy was used, which adjusts for imbalanced classes by averaging the recall (sensitivity) obtained for each class. It is defined by Equation (6).

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{5}$$

$$\text{Balanced Accuracy} = \frac{1}{K} \sum_{k=1}^K \frac{TP_k}{TP_k + FN_k} \tag{6}$$

where K is the number of classes, TP_k is the number of true positives for class k , and FN_k is the number of false negatives for class k .

To gain deeper insights into the model's performance on individual classes, precision, recall, and F1-score [40] were calculated for each class. Precision measures the proportion of correct positive predictions among all positive predictions, defined in Equation (7). Recall, also known as sensitivity, assesses the model's ability to correctly identify all positive instances (see Equation (8)). The F1-score, as defined in Equation (9), is the harmonic mean of precision and recall, which provides a single metric that balances both concerns.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

where TP is the number of true positives, and FP is the number of false positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

In multiclass classification settings with imbalanced classes, evaluating overall model performance requires aggregating these per-class metrics. To account for the varying number of instances in each class, weighted average precision, weighted average recall, and weighted average F1-score were calculated. These metrics are computed by weighting the per-class metrics by the number of true instances in each class to ensure classes with more samples have a proportionally greater impact on the overall score.

The weighted average precision is calculated as follows:

$$\text{Weighted Precision} = \frac{\sum_{k=1}^K n_k \times \text{Precision}_k}{\sum_{k=1}^K n_k} \tag{10}$$

where n_k is the number of true instances in class k . Similarly, the weighted accuracy, average recall, and F1-score were calculated.

The use of balanced accuracy and weighted metrics is particularly important in the presence of class imbalance, which was evident in our dataset (see Table 3). Certain strength categories had significantly more samples than others, which could bias the model’s performance towards those classes. The confusion matrix was also utilized to visualize the performance of the classification models by displaying the counts of true positive, true negative, false positive, and false negative predictions for each class. This matrix allowed for a detailed error analysis by highlighting specific areas where the model was misclassifying observations.

To optimize model performance and ensure robust hyperparameter selection, bayesian optimization was conducted using 5-fold cross-validation. This involved partitioning the training dataset into five equal subsets, training the model on four subsets, and evaluating its performance on the remaining subset. By averaging the performance across folds, this approach provides a more reliable estimate of the model’s generalization ability and helps mitigate the risk of overfitting during hyperparameter tuning.

2.3.3. Minimum Dataset Size Analysis

To address the question of the optimal dataset size for stable and reliable predictions, a subsampling analysis was conducted. The size of the training subset was incrementally increased from 30 samples to 900 samples (in increments of 5), and the performance of the best-performing regression model was evaluated on each subset. For each subset size, we performed multiple runs with 30 different random seeds to obtain the mean and confidence intervals for both R^2 and MSE. This approach allowed us to identify the point at which further increases in dataset size yield diminishing returns in terms of predictive accuracy.

2.4. Feature Importance Analysis

Understanding the contribution of each feature to the predictions of the best-performing regression model was essential for interpreting the model and gaining insights into the factors influencing concrete compressive strength. Therefore, the best-performing regression model was analyzed using two methods: mean decrease in impurity (MDI) [41] and SHAP (SHapley Additive exPlanations) values [42]. These methods provided both global and local interpretability of the model and helped to identify the most influential features.

2.4.1. Mean Decrease in Impurity

The mean decrease in impurity is a feature importance metric intrinsic to tree-based models like the GBR. It quantifies the importance of a feature by measuring how much each feature reduces the impurity in a tree, averaged over all trees in the ensemble. For regression trees, impurity was measured using variance. The impurity $I(m)$ at node m is defined as follows:

$$I(m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - \bar{y}_{N_m})^2 \tag{11}$$

where N_m is the number of samples at node m ; y_i is the target value of sample i ; and \bar{y}_{N_m} is the mean target value at node m . When a node m is split on feature j , the decrease in impurity $\Delta I(j, m)$ due to that feature is calculated as follows:

$$\Delta I(j, m) = I(m) - \left(\frac{N_{left}}{N_m} I(left) + \frac{N_{right}}{N_m} I(right) \right) \tag{12}$$

where N_{left} and N_{right} are the numbers of samples in the left and right child nodes, and $I(left)$ and $I(right)$ are the impurities of the left and right child nodes. The mean decrease in impurity for feature j across all trees T in the ensemble is then as follows:

$$MDI_j = \frac{1}{|T|} \sum_{t \in T} \sum_{m \in M_t} \Delta I_t(j, m) \tag{13}$$

where M_t is the set of all nodes where feature j is used to split in tree t , and $\Delta I_t(j, m)$ is the decrease in impurity for feature j at node m in tree t . A higher MDI value indicates greater importance of the feature in reducing the overall impurity of the model.

2.4.2. SHAP Values

SHAP values provide a unified approach to interpreting model predictions by assigning each feature an importance value for a particular prediction [42]. Based on cooperative game theory, SHAP values consider all possible combinations of features to ensure a fair allocation of the contribution of each feature. The SHAP value ϕ_j for feature j is calculated as follows:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{j\}}(x_{S \cup \{j\}}) - f_S(x_S)] \tag{14}$$

where F is the set of all features, $\{j\}$ denotes the set containing only feature j , S is a subset of features not containing feature j , $|S|$ is the number of features in subset S , $f_S(x_S)$ is the model trained with features in subset S evaluated at x_S , and $f_{S \cup \{j\}}(x_{S \cup \{j\}})$ is the model trained with features in subset $S \cup \{j\}$ evaluated at $x_{S \cup \{j\}}$.

2.4.3. Ablation Study

An ablation study was also conducted to assess the impact of progressively removing features on the model’s performance. Starting with all features, features were removed one at a time in order of increasing importance based on the MDI ranking. After each removal, the GBR was retrained, and its performance was evaluated using the R^2 metric. The R^2 values were then plotted against the number of features retained.

2.4.4. Partial Dependence Plot

To further interpret the influence of key features on the predicted concrete compressive strength, partial dependence plots (PDPs) [43] were employed. This method provides insights into the relationship between the target variable and the features and helps to understand whether the relationship is linear, monotonic, or more complex. The partial dependence function for a feature x_s is defined by Equation (15). For a pair of features x_{s1} and x_{s2} , the two-way partial dependence function is shown in Equation (16).

$$\hat{f}_{PD}(x_s) = \frac{1}{n} \sum_{i=1}^n \hat{f}(x_s, x_C^{(i)}) \tag{15}$$

where \hat{f} is the trained predictive function (the best-performing regressor model), x_s is the feature (or set of features) for which the partial dependence is computed, $x_C^{(i)}$ represents the values of all other features C (the complement of s) for instance i in the dataset, and n is the number of instances in the dataset.

$$\hat{f}_{PD}(x_{s1}, x_{s2}) = \frac{1}{n} \sum_{i=1}^n \hat{f}(x_{s1}, x_{s2}, x_C^{(i)}) \tag{16}$$

In this study, PDPs were generated for the top two most influential features identified in the feature importance analysis. Additionally, a two-way PDP was created to examine the interaction effect between these two features on the predicted compressive strength. The partial dependence functions $\hat{f}_{PD}(x_s)$ and $\hat{f}_{PD}(x_{s1}, x_{s2})$ were calculated using the `PartialDependenceDisplay.from_estimator` method from the scikit-learn library.

The method systematically varies the feature(s) of interest while averaging out the effects of all other features.

2.5. Model Implementation and Validation

The final models were implemented using optimized hyperparameters, and their validation involved evaluating performance metrics on the test set to assess how well they generalized to new, unseen data. For the regression tasks, actual versus predicted values were visualized using scatter plots to qualitatively assess predictive accuracy, while residual analysis was conducted to identify potential patterns that might reveal model bias or heteroscedasticity.

3. Results

3.1. Regression Analysis

The regression models were evaluated based on their MSE and R^2 values, as summarized in Table 6. This table provides a clear comparison of their effectiveness in predicting concrete compressive strength. The GBR emerged as the top performer with an MSE of 15.79 and an R^2 value of 0.94, which indicates its ability to explain 94% of the variance in compressive strength. Following closely, the RF regressor captured a significant portion of the target variable’s variance with an R^2 value of 0.91 and an MSE of 21.61. Both the neural network model and AdaBoost also showed strong results, each with R^2 values of 0.90. The KNN model demonstrated a moderate fit with an R^2 of 0.84 and an MSE of 39.88, while the decision tree regressor posted an MSE of 42.67 and an R^2 of 0.83. The linear regression model, simpler and less robust, managed an R^2 of 0.69 and an MSE of 71.25, which highlights its limited capacity to capture complex patterns in the data.

Table 6. Performance of regression models.

Model	MSE	R^2
gradient boosting regressor	15.79	0.94
RF regressor	21.61	0.91
neural network model	24.20	0.90
AdaBoost	24.27	0.90
k-nearest neighbors	39.88	0.84
decision tree regressor	42.67	0.83
linear regression	71.25	0.69

Our R^2 of 0.94 closely matches Alghairi et al. [4]’s R^2 of 0.90 using a gradient-boosted trees model for nanomaterial lightweight concrete. This improvement is possibly due to our ratio-based features (W/C and C/F) and enhanced hyperparameter tuning. Similarly, Ding et al. [5] found that ensemble methods like RF and SVM outperformed single models in predicting the compressive strength of alkali-activated materials.

To complement the statistical summary in Table 6, Figure 6 presents a Taylor diagram that visually compares the predictions of each model to the observed compressive strengths. In this diagram, the distance from the origin corresponds to the models’ standard deviations, and their angular position represents the correlation with the observed data. Additionally, the annotations near each model’s marker show the centered RMS (CRMS) error, which provides a measure of how closely the model predictions match the observed values after removing any bias. From Figure 6, we see that the GBR and RF models not only rank highly in terms of MSE and R^2 but also cluster closer to the observed standard deviation reference point, exhibit higher correlations, and have lower CRMS errors. These visual insights confirm and reinforce the numerical findings presented in Table 6. Meanwhile,

the neural network and AdaBoost models maintain strong correlations and relatively low CRMS errors, which align well with their high R^2 values. In contrast, the KNN and decision tree models, while moderately correlated, display larger CRMS errors, consistent with their higher MSE values. The linear regression model stands out as having the weakest correlation and the highest CRMS error, mirroring its poor performance in terms of MSE and R^2 .

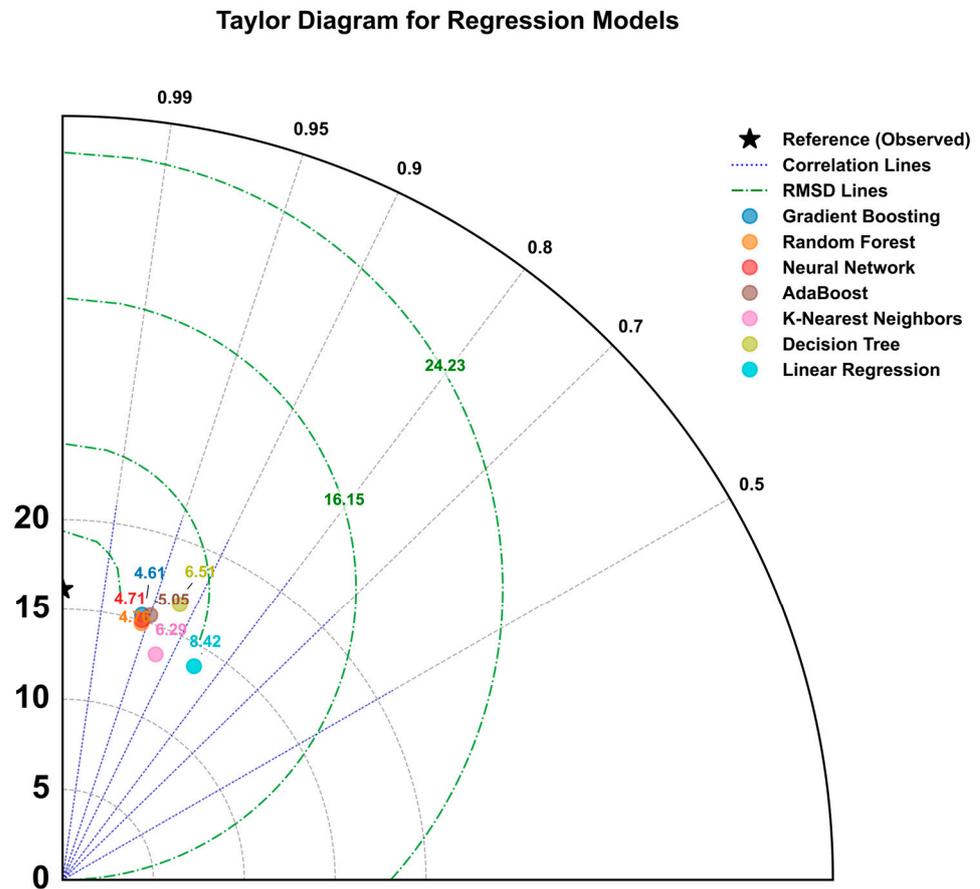


Figure 6. Taylor diagram for regression models.

The robustness of the GBR is further supported by Figure 7a,b. In Figure 7a, the residual plot demonstrates that the residuals are randomly scattered around zero, which indicates the absence of systematic patterns or biases. The residual variance is consistent across the predicted values and suggests that the model performs reliably across the range of compressive strengths. This uniformity reinforces the model’s superior fit. In Figure 7b, the “actual vs. predicted values” plot shows points closely aligned with the ideal red dashed line, which highlights the model’s accuracy in predicting the actual values. The tight clustering around this line supports the model’s ability to make precise predictions.

In addition to evaluating model performance on the full dataset, we investigated how model accuracy changes with different training set sizes. Figure 8 illustrates the relationship between subset size and gradient-boosting regressor performance. Initially, as the subset size increases from 30 samples upward, the R^2 score improves dramatically, while the MSE decreases significantly. Beyond approximately 400 samples, the improvement in R^2 and reduction in MSE become marginal, suggesting that the model has captured the underlying data patterns sufficiently well. Hence, while larger datasets can still provide benefits, a dataset size of around 400 observations appears to be a practical lower bound for achieving near-optimal performance in this particular problem. This analysis suggests that the current cleaned dataset size of 911 observations is more than sufficient for stable

and high-quality predictions, and smaller datasets (on the order of a few hundred samples) could still achieve near-optimal results, given a similar data distribution and complexity.

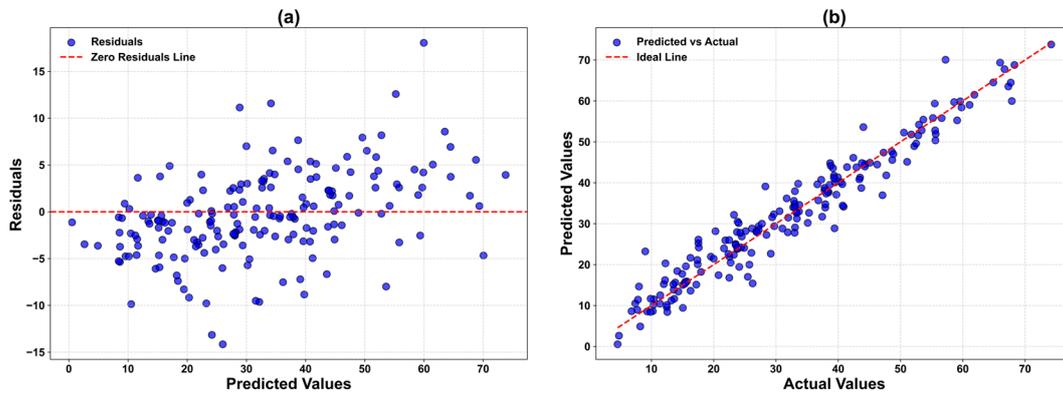


Figure 7. Residual analysis and prediction accuracy of the GBR. (a): residual plot; (b): actual vs. predicted values.

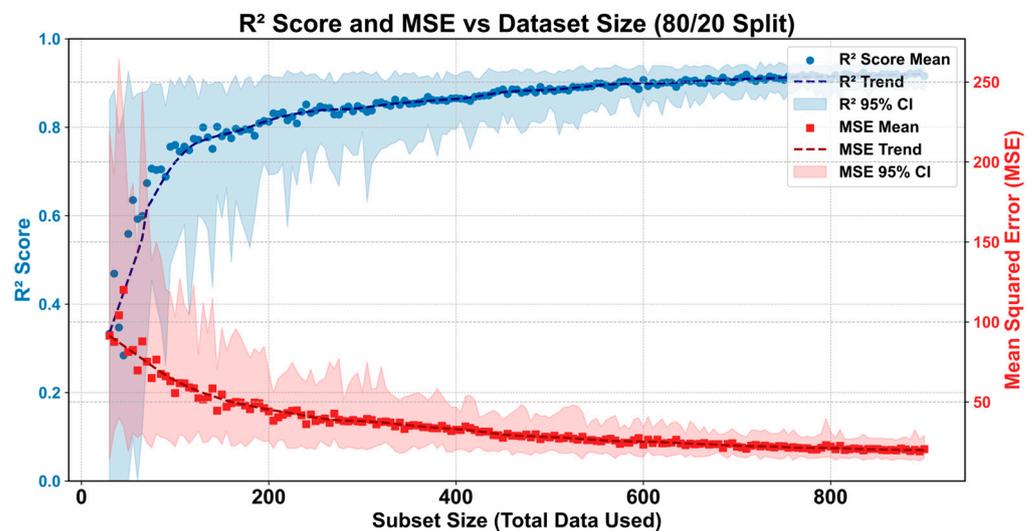


Figure 8. R² score and MSE vs. dataset size (80/20 split) for the GBR model.

3.2. Classification Analysis

For the classification analysis, the models were evaluated using metrics such as accuracy, precision, recall, and F1-score. The classification models demonstrated varied performance, as summarized below and detailed in Table 7. The SVM classifier achieved the highest overall accuracy among the tested models at 0.80. The SVM classifier proved to be the best-performing model, and its confusion matrix is shown in Figure 9. It balanced precision and recall effectively across all classes and showed particular strength in correctly classifying the “very weak” category. It also handled the nuances between “high strength”, “normal strength”, and “weak” categories better than other models, which indicates its ability to capture more complex patterns in the data. The bagging classifier, with consistent scores of 0.76 and above across all metrics, also showed strong and balanced performance. The RF model demonstrated acceptable precision at 0.76 but had slightly lower accuracy and recall scores compared to SVM and bagging, which suggests effectiveness in correctly identifying certain classes but with some limitations in achieving consistent accuracy across all predictions. The logistic regression model and KNN model displayed lower performance metrics, with balanced accuracies of 0.63 and 0.62, respectively.

Table 7. Performance of classification models.

Model	Balanced Accuracy	Weighted Accuracy	Weighted Avg Precision	Weighted Avg Recall	Weighted Avg F1-Score
RF classifier	0.74	0.73	0.76	0.75	0.75
logistic regression	0.63	0.62	0.63	0.64	0.63
SVM classifier	0.76	0.78	0.80	0.80	0.80
KNN	0.62	0.53	0.69	0.69	0.68
bagging with decision trees	0.77	0.78	0.77	0.76	0.76

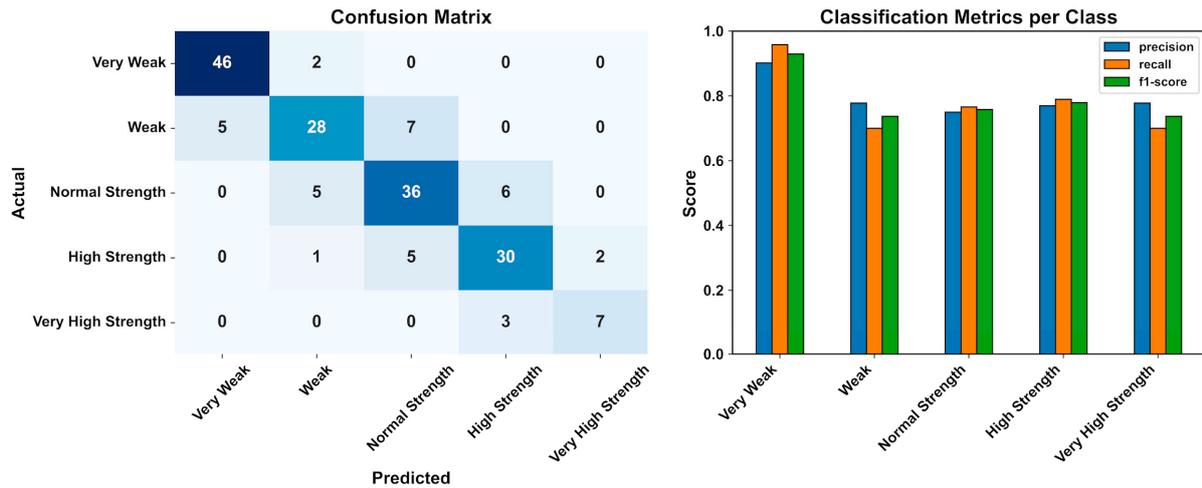


Figure 9. Confusion matrix for SVM (heatmap colors darken as count increase) and classification matrix by class.

To provide further clarity on model reproducibility, Table 8 presents the final hyperparameter configurations obtained through Bayesian optimization for the top-performing regression model (GRB) and the top-performing classification models (SVM). Detailed hyperparameters and tuning procedures for all other models are available in the Supplementary Materials.

Table 8. Best hyperparameters for selected top-performing models.

Model	Hyperparameters Considered	Initial/Default Values	Hyperparameter Tuning Method	Best/Tuned Values
GRB	n_estimators, learning_rate, "max_depth, subsample, min_samples_split	n_estimators = 100, learning_rate = 0.1, max_depth = 3, subsample = 1.0, min_samples_split = 2	Bayesian Optimization	n_estimators = 500, learning_rate = 0.2057, max_depth = 10, subsample = 0.5, min_samples_split = 0.242
SVM	C (regularization), gamma (kernel coefficient), kernel (linear, rbf, poly, sigmoid), degree (if kernel = poly)	C = 1.0, kernel = 'rbf', gamma = 'scale', degree = 3	Bayesian Optimization (BayesSearchCV)	C ≈ 5.68 × 10 ⁵ , gamma ≈ 0.1434, kernel = 'rbf', degree = 5

3.3. Feature Importance Ranking and Feature Ablation

Understanding feature contributions is crucial for interpreting concrete strength prediction models and identifying influential factors. Therefore, the feature importance values were extracted from the GBR to provide a measure of each feature’s influence on the predictive model. Figure 10a illustrates the feature importance ranking. The importance scores are normalized to sum up to 1 to allow for direct comparison among features. Analysis shows water–cement ratio (0.425) and age (0.301) are the most significant predictors of concrete compressive strength. Blast furnace slag (0.106) and superplasticizer (0.080) show moderate influence, while coarse aggregate–fine aggregate ratio (0.059) and fly ash (0.029) have lesser impacts.

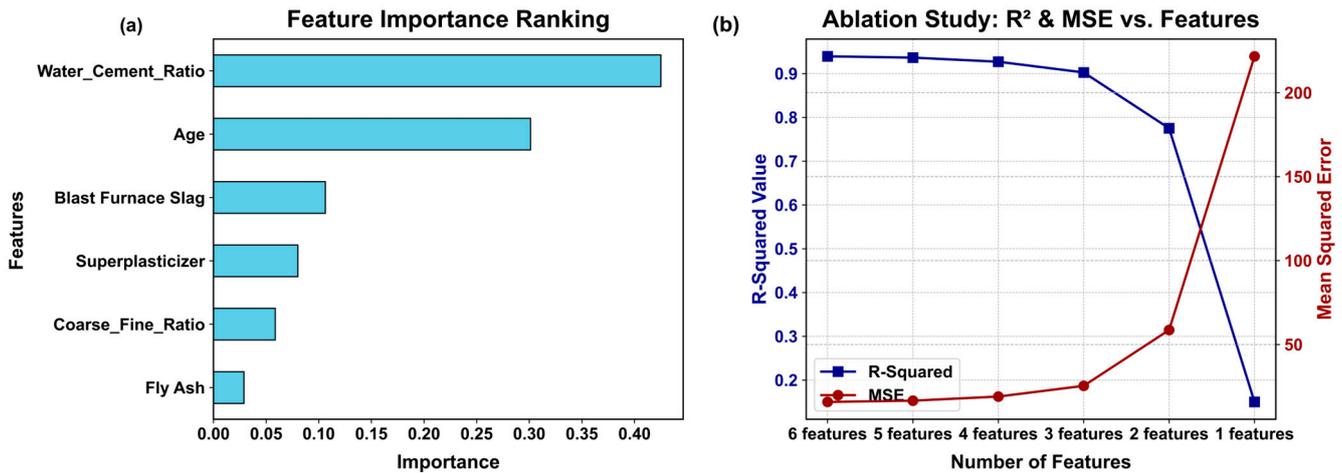


Figure 10. (a) Feature importance ranking; (b) contribution of each feature to model performance.

To further assess the impact of each feature on the model’s performance, an ablation study was conducted. In this study, features were progressively removed from the model in order of increasing importance (starting with the least important feature), and the model was retrained each time. The MSE and R² values were recorded at each step to evaluate how the removal of features affected the model’s predictive accuracy.

The ablation study (Figure 10b) shows the impact of incrementally removing features on both R² and MSE and clarifies each feature’s individual contribution to the model’s performance. Starting with all six variables (Water_Cement_Ratio, Age, Blast Furnace Slag, Superplasticizer, Coarse_Fine_Ratio, and Fly Ash), we obtained an R² of 0.9394 and an MSE of 15.7961. Removing Fly Ash had a minimal effect on accuracy (R² = 0.9366, MSE = 16.5504), which indicates that although it adds some predictive value, its contribution is relatively modest compared to the top-ranked features. Further reducing the feature set led to more substantial declines: while retaining only the top three predictors—Water_Cement_Ratio, Age, and Blast Furnace Slag—still achieved a commendable R² of 0.9027, the MSE increased to 25.3888. Narrowing down to just two features (Water_Cement_Ratio and Age) caused R² to drop to 0.7752 and MSE to rise to 58.6519, and relying solely on Water_Cement_Ratio produced a drastic decline (R² = 0.1501, MSE = 221.6960). These results emphasize the importance of multiple synergistic features in achieving both high R² and low MSE, with Water_Cement_Ratio, Age, and Blast Furnace Slag being particularly influential. Conversely, features like Fly Ash and Coarse_Fine_Ratio demonstrate lower predictive accuracy due to weaker direct correlations with compressive strength or their effects being overshadowed by more dominant parameters. Fly Ash, for instance, may improve strength and durability under certain conditions but exerts a more subtle or context-dependent influence on early-age compressive strength, which makes its overall contribution less

pronounced in a broad dataset. Similarly, the Coarse_Fine_Ratio’s influence is secondary to that of Water_Cement_Ratio and Age, which directly shape hydration kinetics and microstructural development. Thus, while these lower-ranked features are not without value, their marginal improvements are minimal relative to the top three predictors. Taken together, these findings suggest that a simplified model using only a few key variables can still achieve near-optimal accuracy, providing practical guidance for future model development and feature selection.

3.4. Understanding Feature Contributions with SHAP Analysis

To gain a deeper understanding of the GBR’s predictive behavior and interpret its predictions, we employed SHAP analysis. This method allows for both global and local interpretability and reveals the contribution of each feature to the model’s output across the entire dataset and for individual predictions. Figure 11a presents the SHAP summary plot, which displays the global feature importance. Each point on the plot represents a SHAP value for a feature and an instance. The features are ordered by their overall importance, with the most important feature at the top. The color of the points indicates the feature value, with red representing high values and blue representing low values. The SHAP summary plot confirms the findings from the MDI analysis and highlights the water–cement ratio and age as the most influential features. Higher values of water–cement ratio generally contribute negatively to the predicted strength, while higher values of age have a positive impact. Blast furnace slag and superplasticizer also show moderate influence, with higher values of blast furnace slag typically decreasing the predicted strength and higher values of superplasticizer increasing it. Fly ash and coarse aggregate–fine aggregate ratio have relatively smaller impacts on the predictions.

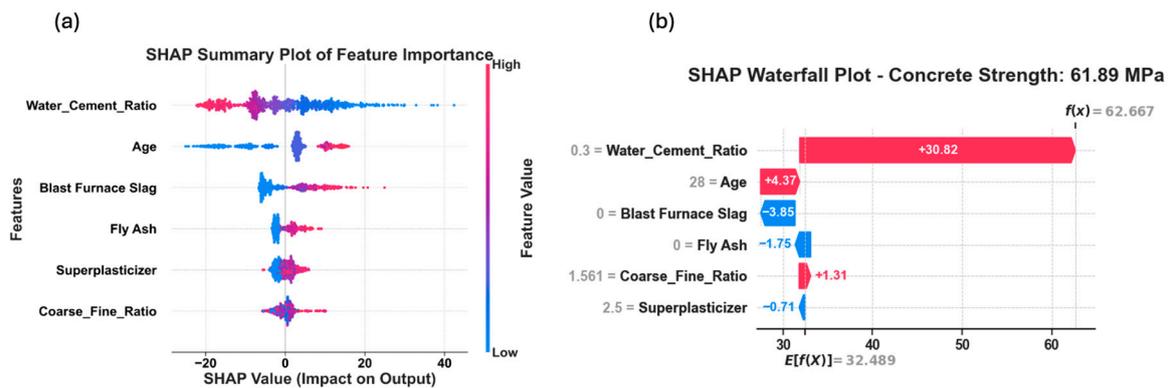


Figure 11. (a) Feature importance analysis: SHAP summary plot; (b) contribution analysis: SHAP waterfall plot showing feature contributions for an actual concrete strength of 61.89 MPa.

Figure 11b shows a SHAP waterfall plot for a specific instance with an actual concrete strength of 61.89 MPa. This plot provides a local explanation to illustrate how each feature contributes to the model’s prediction for this particular instance. The base value, represented by $E[f(X)]$, is the average prediction of the model across the entire dataset (32.489 MPa). Each bar in the plot represents a feature, and its length corresponds to the SHAP value to indicate the magnitude and direction of the feature’s contribution to the final prediction. For this instance, the water–cement ratio of 0.3 has the largest positive contribution (+30.82), significantly increasing the prediction from the base value. The age of 28 days also contributes positively (+4.37), further increasing the predicted strength. Conversely, the absence of blast furnace slag (−3.85), fly ash (−1.75), and a moderate amount of superplasticizer (−0.711) contribute negatively, slightly lowering the prediction. The coarse

aggregate–fine aggregate ratio has a small positive impact (+1.31). The final prediction ($f(x)$) of 62.667 MPa is the sum of the base value and all the individual feature contributions.

The partial dependence plots presented in Figure 12 highlight the influence of water–cement ratio and age on the compressive strength of concrete, as predicted by the gradient boosting model. Figure 12a displays a marked decrease in concrete compressive strength as the water–cement ratio increases from around 0.3 to 1.25. Initially, the decline is substantial, particularly between ratios of 0.3 to 0.75 which indicates that lower ratios significantly enhance the concrete’s strength. Beyond a ratio of 0.75, the negative impact on strength continues but becomes less pronounced, eventually less significant after a ratio of 1.0. This suggests that maintaining a water–cement ratio below 0.75 is critical for optimal concrete strength. The age of concrete (Figure 12b) shows a robust positive correlation with its compressive strength. From day 0 to approximately 50 days, there is a sharp increase in strength, which reflects the critical curing phase, in which concrete gains most of its compressive strength. Beyond 50 days, the rate of increase in strength diminishes, becoming more gradual up to 100 days. The step increase in strength at around 100 days might indicate specific curing or environmental conditions affecting the concrete’s long-term strength characteristics. The interaction plot (Figure 12c) elucidates how combinations of water–cement ratio and age impact concrete strength. At early ages (0–20 days) and lower water–cement ratios (0.3–0.5), the concrete strength is highest, which emphasizes the importance of both proper mixture ratios and sufficient curing time. As the age increases, even higher water–cement ratios (up to 1.5) show a less detrimental effect on the strength, particularly in concrete aged over 60 days. This interaction suggests a diminishing influence of the water–cement ratio on strength as the concrete matures.

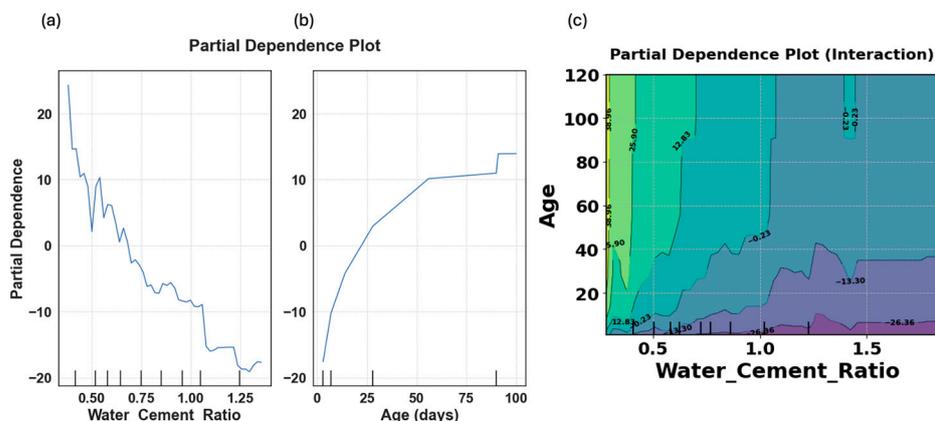


Figure 12. (a) Partial dependence plot for water–cement ratio; (b) partial dependence plot for age; (c) partial dependence plot showing the combined influence of water–cement ratio and age (cooler colors (purple zones) indicate lower partial dependence values, warmer colors (greenish zones) indicate higher values).

4. Discussion

The findings of this study highlight the significant potential of machine learning models in accurately predicting and classifying the compressive strength of concrete based on its mix design parameters and curing age. The superior performance of the GBR underscores the effectiveness of ensemble methods in capturing the complex, non-linear relationships inherent in concrete materials. This discussion elaborates on the implications of these results, the insights gained from feature importance analyses, the challenges encountered, and the broader impact on the field of concrete technology.

4.1. Model Performance Insights

Gradient boosting regression emerged as the most effective model for predicting concrete compressive strength, by achieving an R^2 of 0.94 and an MSE of 15.79. This model's superior performance is attributed to its ability to capture complex non-linear relationships between predictors and the target variable, which are inherent in concrete behavior due to intricate chemical and physical interactions. While RF also demonstrated strong performance with an R^2 value of 0.91, neural networks did not surpass gradient boosting, likely due to the dataset size limitations. The comparatively lower performance of linear regression underscores the inadequacy of linear models for capturing the non-linear dynamics of concrete properties.

The classification task revealed that the SVM classifier achieved the highest accuracy, correctly classifying compressive strength categories with a balanced accuracy of 0.76 and a weighted F1-score of 0.80. The SVM's ability to handle high-dimensional spaces and its effectiveness with non-linear kernels likely contributed to its superior performance. However, the challenge of classifying intermediate strength classes was evident across all models. Misclassifications often occurred between "high strength" and "normal strength" categories, possibly due to overlapping feature distributions and the inherent variability in concrete mixes. The use of balanced accuracy and weighted metrics was crucial in this context, as it accounted for class imbalances within the dataset. Some strength categories, such as "very high strength", had significantly fewer samples, which could bias the models toward the majority classes. By employing these metrics, the evaluation provided a more accurate reflection of the models' capabilities across all categories.

Our results underscore the effectiveness of ensemble methods for compressive strength prediction and align with prior studies that similarly reported boosted trees or hybrid approaches outperforming conventional regressors [4,5,8,9]. For instance, Song et al. [8] and Paudel et al. [7] each found that bagging- or boosting-based models attained R^2 values exceeding 0.90, whereas simpler models such as linear or decision tree regressors lagged. Notably, Tran et al. [9] and Ahmad et al. [10] showed that hybrid or ensemble algorithms could achieve R^2 values above 0.93 for recycled and geopolymer concretes, respectively, further evidencing that these advanced architectures generalize effectively across various binder systems. Our GBR's $R^2 = 0.94$ and the SVM classification accuracy of 0.80 for strength categories thus corroborate the conclusion that robust ensemble approaches can accommodate the heterogeneous nature of concrete composites and yield superior predictive accuracy.

4.2. Feature Importance and Practical Implications

Analysis using MDI and SHAP values revealed that water–cement ratio and age are the most critical factors influencing compressive strength. A lower water–cement ratio reduces porosity, enhancing strength, while increased age allows continued hydration and microstructure densification, with strength gains leveling off after about 50 days. Blast furnace slag and superplasticizer also contribute moderately. Blast furnace slag improves long-term strength through latent hydraulic reactions, and superplasticizers enhance workability, enabling lower water content without sacrificing performance. These insights aid mix design optimization by highlighting key components. Focusing on optimizing water–cement ratio and curing time can significantly boost compressive strength efficiently and offers economic and environmental benefits by potentially reducing cement usage.

The use of SHAP values provided a nuanced understanding of how individual features influenced model predictions at both global and local levels. The waterfall plot for a specific instance illustrated how feature values contribute to a single prediction and enhanced the model interpretability. This level of interpretability is crucial for gaining trust in machine

learning models within the construction industry, where decisions have significant safety and financial implications. By demonstrating that the model's behavior aligns with domain knowledge, stakeholders are more likely to adopt these data-driven approaches. The ability to predict strength without extensive laboratory testing accelerates the design process and enhances project efficiency.

It is also important to note that the strong influence of the water–cement ratio and curing age in our analysis concurs with numerous prior investigations. For instance, Ding et al. [5] and Ekanayake et al. [6] both identified age (or curing duration) as a dominant factor in concrete strength evolution, while Alghrairi et al. [4] and Anjum et al. [11] emphasized the significant role of water content. Our SHAP-based interpretability analysis (Section 3.4) parallels these findings and demonstrates that small changes in W/C ratio lead to sizable shifts in predicted strength. Moreover, partial dependence plots revealed synergy between W/C ratio and curing time, aligning with earlier studies that used SHAP or feature-importance techniques for clarity [6,11]. As a result, our results substantiate that data-driven ranking of variables (e.g., W/C ratio, age) resonates strongly with well-established concrete fundamentals.

4.3. Challenges, Limitations, and Future Research

Despite the promising results, several challenges were encountered during the study. One notable challenge was multicollinearity among the original features, which was addressed through feature engineering by creating ratios, such as the water–cement ratio and the coarse aggregate–fine aggregate ratio. While this approach reduced multicollinearity and improved model performance, the engineered features still exhibited higher-than-desirable VIF values. This suggests that further refinement or alternative methods, such as regularization techniques, may be necessary to fully mitigate multicollinearity.

Another limitation pertains to the dataset used in this study. The subsampling analysis revealed that a more moderate sample size of around 400 instances would be sufficient to achieve stable predictive performance. This is particularly valuable for future studies that may face data availability constraints, suggesting that similar models can be developed with smaller datasets without compromising accuracy. However, the dataset's comprehensiveness presents challenges. While it covers a wide range of mix designs and curing ages, it may not fully capture variations in raw materials, environmental conditions, and construction practices across different regions. This could affect the model's generalizability to other contexts. Therefore, it is suggested that future research incorporate larger and more diverse datasets, including various cement types, aggregate sources, and environmental conditions, to enhance the model's robustness and applicability in varied settings. Additionally, the removal of outliers, while improving model performance, presents challenges. Excluding outliers may omit valid but extreme cases, potentially limiting the model's ability to predict accurately in scenarios. Future studies should explore methods that balance outlier removal with the retention of essential data points to maintain comprehensive predictive capabilities.

To advance these findings, future research is suggested to focus on expanding datasets by incorporating diverse sources and geographical locations. Advanced feature engineering techniques, such as polynomial features, interaction terms, and real-time monitoring data such as temperature and humidity during curing, can capture more nuanced data patterns and improve predictive accuracy. Exploring deep learning approaches may reveal complex relationships not identified by traditional machine learning models, particularly when combined with non-traditional data sources such as imaging or sensor data. Also, enhancing model interpretability through methods such as layer-wise relevance propaga-

tion or integrated gradients is crucial for industry adoption, ensuring that complex models remain transparent and trustworthy.

Furthermore, integrating predictive models into user-friendly decision support systems, such as software tools or mobile applications, and incorporating optimization algorithms can facilitate practical use by practitioners, enabling automated mix design suggestions tailored to specific project requirements. The adoption of machine learning models in concrete technology also raises ethical and environmental considerations. Optimizing mix designs for strength and cost must be balanced with sustainability goals, such as reducing carbon emissions associated with cement production. Future models could incorporate environmental impact metrics to support eco-friendly decision-making.

5. Conclusions

The present study demonstrated the effectiveness of machine learning algorithms, especially ensemble techniques such as gradient boosting, in making precise predictions and classifying the compressive strength of concrete based on mix design parameters and curing duration. Using advanced feature importance analysis techniques, including SHAP values and partial dependence plots, allowed us to delve into the details of how the input variables interact with each other in these models to affect the predictions. These results have shown the potential of machine learning models to enhance mix design optimization, quality assurance, and fulfillment of engineering standards. SHAP analysis allowed a better insight into feature contributions on both a global and local level, thus possibly increasing model interpretability. The ability to predict strength without extensive laboratory testing accelerates the design process, reduces costs, and promotes more efficient project timelines.

However, the study acknowledges certain limitations. While the dataset used is comprehensive, it may not capture all possible variations in raw materials, environmental conditions, and construction practices across different regions. Exploring deep learning approaches and integrating real-time monitoring data could uncover more complex relationships and enhance the model's robustness. Additionally, improving model interpretability is essential for ensuring widespread adoption in the industry.

Supplementary Materials: The following supporting information, including the code used for data preprocessing, feature engineering, model development, and analysis in this study, can be downloaded at: <https://github.com/mnikoopayan/Concrete-Compressive-Strength> (accessed on 12 July 2024).

Author Contributions: Conceptualization, M.S.N.T. and Y.F.; methodology, M.S.N.T. and Y.F.; software, M.S.N.T.; validation, M.S.N.T., Y.F. and M.M.; formal analysis, M.S.N.T. and Y.F.; investigation, M.S.N.T.; resources, M.S.N.T. and M.M.; data curation, M.S.N.T. and M.M.; writing—original draft preparation, M.S.N.T. and Y.F.; writing—review and editing, Y.F. and M.M.; visualization, M.S.N.T. and Y.F.; supervision, Y.F.; project administration, Y.F.; funding acquisition, Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in the UC Irvine Machine Learning Repository at 10.24432/C5PK67.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Griffiths, S.; Sovacool, B.K.; Furszyfer Del Rio, D.D.; Foley, A.M.; Bazilian, M.D.; Kim, J.; Uratani, J.M. Decarbonizing the Cement and Concrete Industry: A Systematic Review of Socio-Technical Systems, Technological Innovations, and Policy Options. *Renew. Sustain. Energy Rev.* **2023**, *180*, 113291. [[CrossRef](#)]

2. Young, B.A.; Hall, A.; Pilon, L.; Gupta, P.; Sant, G. Can the Compressive Strength of Concrete Be Estimated from Knowledge of the Mixture Proportions?: New Insights from Statistical Analysis and Machine Learning Methods. *Cem. Concr. Res.* **2019**, *115*, 379–388. [[CrossRef](#)]
3. Li, Z.; Yoon, J.; Zhang, R.; Rajabipour, F.; Srubar, W.V., III; Dabo, I.; Radlińska, A. Machine Learning in Concrete Science: Applications, Challenges, and Best Practices. *npj Comput. Mater.* **2022**, *8*, 127. [[CrossRef](#)]
4. Alghrairi, N.S.; Aziz, F.N.; Rashid, S.A.; Mohamed, M.Z.; Ibrahim, A.M. Machine Learning-Based Compressive Strength Estimation in Nanomaterial-Modified Lightweight Concrete. *Open Eng.* **2024**, *14*, 20220604. [[CrossRef](#)]
5. Ding, Y.; Wei, W.; Wang, J.; Wang, Y.; Shi, Y.; Mei, Z. Prediction of Compressive Strength and Feature Importance Analysis of Solid Waste Alkali-Activated Cementitious Materials Based on Machine Learning. *Constr. Build. Mater.* **2023**, *407*, 133545. [[CrossRef](#)]
6. Ekanayake, I.U.; Meddage, D.P.P.; Rathnayake, U. A Novel Approach to Explain the Black-Box Nature of Machine Learning in Compressive Strength Predictions of Concrete Using Shapley Additive Explanations (SHAP). *Case Stud. Constr. Mater.* **2022**, *16*, e01059. [[CrossRef](#)]
7. Paudel, S.; Pudasaini, A.; Shrestha, R.K.; Kharel, E. Compressive Strength of Concrete Material Using Machine Learning Techniques. *Clean. Eng. Technol.* **2023**, *15*, 100661. [[CrossRef](#)]
8. Song, H.; Ahmad, A.; Farooq, F.; Ostrowski, K.A.; Maślak, M.; Czarnecki, S.; Aslam, F. Predicting the Compressive Strength of Concrete with Fly Ash Admixture Using Machine Learning Algorithms. *Constr. Build. Mater.* **2021**, *308*, 125021. [[CrossRef](#)]
9. Quan Tran, V.; Quoc Dang, V.; Si Ho, L. Evaluating Compressive Strength of Concrete Made with Recycled Concrete Aggregates Using Machine Learning Approach. *Constr. Build. Mater.* **2022**, *323*, 126578. [[CrossRef](#)]
10. Ahmad, A.; Ahmad, W.; Chaiyasarn, K.; Ostrowski, K.A.; Aslam, F.; Zajdel, P.; Joyklad, P. Prediction of Geopolymer Concrete Compressive Strength Using Novel Machine Learning Algorithms. *Polymers* **2021**, *13*, 3389. [[CrossRef](#)] [[PubMed](#)]
11. Anjum, M.; Khan, K.; Ahmad, W.; Ahmad, A.; Amin, M.N.; Nafees, A. Application of Ensemble Machine Learning Methods to Estimate the Compressive Strength of Fiber-Reinforced Nano-Silica Modified Concrete. *Polymers* **2022**, *14*, 3906. [[CrossRef](#)]
12. Ullah, H.S.; Khushnood, R.A.; Farooq, F.; Ahmad, J.; Vatin, N.I.; Ewais, D.Y.Z. Prediction of Compressive Strength of Sustainable Foam Concrete Using Individual and Ensemble Machine Learning Approaches. *Materials* **2022**, *15*, 3166. [[CrossRef](#)]
13. Kumar, P.; Pratap, B. Feature Engineering for Predicting Compressive Strength of High-Strength Concrete with Machine Learning Models. *Asian J. Civ. Eng.* **2024**, *25*, 723–736. [[CrossRef](#)]
14. Nguyen, N.-H.; Abellán-García, J.; Lee, S.; Vo, T.P. From Machine Learning to Semi-Empirical Formulas for Estimating Compressive Strength of Ultra-High Performance Concrete. *Expert Syst. Appl.* **2024**, *237*, 121456. [[CrossRef](#)]
15. Onyelowe, K.C.; Gnananandarao, T.; Ebid, A.M.; Mahdi, H.A.; Ghadikolaee, M.R.; Al-Ajamee, M. Evaluating the Compressive Strength of Recycled Aggregate Concrete Using Novel Artificial Neural Network. *Civ. Eng. J.* **2022**, *8*, 1679–1693. [[CrossRef](#)]
16. Onyelowe, K.C.; Ebid, A.M.; Mahdi, H.A.; Riofrio, A.; Eidgahee, D.R.; Baykara, H.; Soleymani, A.; Kontoni, D.-P.N.; Shakeri, J.; Jahangir, H. Optimal Compressive Strength of RHA Ultra-High-Performance Lightweight Concrete (UHPLC) and Its Environmental Performance Using Life Cycle Assessment. *Civ. Eng. J.* **2022**, *8*, 2391–2410. [[CrossRef](#)]
17. Onyelowe, K.C.; Kontoni, D.-P.N.; Ebid, A.M.; Dabbaghi, F.; Soleymani, A.; Jahangir, H.; Nehdi, M.L. Multi-Objective Optimization of Sustainable Concrete Containing Fly Ash Based on Environmental and Mechanical Considerations. *Buildings* **2022**, *12*, 948. [[CrossRef](#)]
18. ACI Committee 318; American Concrete Institute. *Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary*; American Concrete Institute: Farmington Hills, MI, USA, 2008; ISBN 978-0-87031-264-9.
19. Yeh, I.-C. Concrete Compressive Strength. *UCI Mach. Learn. Repos.* **2007**, *10*, C5PK67.
20. McKinney, W. Pandas: A Foundational Python Library for Data Analysis and Statistics. *Python High Perform. Sci. Comput.* **2011**, *14*, 1–9.
21. Vinutha, H.P.; Poornima, B.; Sagar, B.M. Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In *Information and Decision Sciences, Proceedings of the 6th International Conference on FICTA, Bhubaneswar, India, 14–16 October 2017*; Satapathy, S.C., Tavares, J.M.R.S., Bhateja, V., Mohanty, J.R., Eds.; Springer: Singapore, 2018; pp. 511–518.
22. Tukey, J.W. *Exploratory Data Analysis*; Addison-Wesley Pub. Co.: Reading, MA, USA, 1977; ISBN 978-0-201-07616-5.
23. Hunter, J.D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [[CrossRef](#)]
24. American Concrete Institute. *Building Code Requirements for Structural Concrete (ACI 318-19) and Commentary*; American Concrete Institute: Farmington Hills, MI, USA, 2019.
25. McKinney, W. Data Structures for Statistical Computing in Python. *Proc. Python Sci. Conf.* **2010**, *445*, 56–61.
26. Waskom, M.L. Seaborn: Statistical Data Visualization. *J. Open Source Softw.* **2021**, *6*, 3021. [[CrossRef](#)]
27. O'Brien, R.M. A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Qual. Quant.* **2007**, *41*, 673–690. [[CrossRef](#)]
28. Hover, K.C. The Influence of Water on the Performance of Concrete. *Constr. Build. Mater.* **2011**, *25*, 3003–3013. [[CrossRef](#)]
29. Hashemi, M.; Shafiq, P.; Karim, M.R.B.; Atis, C.D. The Effect of Coarse to Fine Aggregate Ratio on the Fresh and Hardened Properties of Roller-Compacted Concrete Pavement. *Constr. Build. Mater.* **2018**, *169*, 553–566. [[CrossRef](#)]

30. Iqbal Khan, M.; Abbass, W.; Alrubaidi, M.; Alqahtani, F.K. Optimization of the Fine to Coarse Aggregate Ratio for the Workability and Mechanical Properties of High Strength Steel Fiber Reinforced Concretes. *Materials* **2020**, *13*, 5202. [[CrossRef](#)] [[PubMed](#)]
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
32. 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.
33. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: New York, NY, USA, 2009; ISBN 978-0-387-84857-0.
34. Massey, F.J., Jr. The Kolmogorov-Smirnov Test for Goodness of Fit. *J. Am. Stat. Assoc.* **1951**, *46*, 68–78. [[CrossRef](#)]
35. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. {TensorFlow}: A System for {Large-Scale} Machine Learning. In Proceedings of the 12th USENIX symposium on operating systems design and implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
36. Chollet, F. Fchollet/Keras-Resources. Available online: <https://github.com/fchollet/keras-resources> (accessed on 20 November 2024).
37. Bayesian Optimization in Action. Available online: <https://www.manning.com/books/bayesian-optimization-in-action> (accessed on 15 January 2025).
38. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. Available online: <https://arxiv.org/abs/1206.2944> (accessed on 15 January 2025).
39. Taylor, K.E. Summarizing Multiple Aspects of Model Performance in a Single Diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [[CrossRef](#)]
40. Powers, D.M.W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *arXiv* **2020**, arXiv:2010.16061.
41. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
42. Lundberg, S.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *arXiv* **2017**, arXiv:1705.07874.
43. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.