**A red and grey logo

Description automatically generated**

**PROJECT TITLE:**

**COMPUTING THE ANALYSIS FOR CONCRETE COMPRESSIVE STRENGTH USING MACHINE LEARNING MODELS**

**Data Analysis in Construction Management (CNST 6308)**

**TERM: FALL – 2024**

**INSTRUCTOR:  
Dr. Lu Gao**

**GROUP MEMBER:  
NUPOOR V KUMBHAR (ID: 2351355)**

# **ABSTRACT**

Numerous research efforts have independently demonstrated that the development of concrete compressive strength is influenced by the composition of all the ingredients in the concrete and not only by the water-to-cement ratio[1]. Modeling the correct behavior of the concrete compressive strength is difficult as it involves many ingredients, making it a highly complex material[1]. Even though the basic formulation suggests that the water content and cement content can determine the paste consistency, which influences concrete's overall strength [2], this is not entirely true when the analysis considers the many different factors involved. Therefore, it is crucial to identify the key ingredients that influence the determination of concrete compressive strength, as it is a critical performance requirement for high-performance construction management and structural engineering [3]. The primary elements and factors influencing the compressive strength of concrete are identified in this project using a range of machine-learning methods. Numerous predictive models were developed to evaluate the behavior of each component and how it influences the concrete's compressive strength. High-performance concrete was investigated using an estimated dataset of 1,030 items. Nine input variables as cement, water, concrete age, blast furnace slag, fine aggregate, and superplasticizer were included in this dataset. The findings demonstrate that regression models such as Random Forest and Gradient Boosting produce more accurate predictions. A deep learning model was also developed using Keras with TensorFlow to complement this study[7].

Keywords: Concrete compressive strength, Machine learning models, Deep learning, Random forest, Gradient Boosting.

# **INTRODUCTION**

Concrete compressive strength plays a pivotal role in determining high-performance concrete, used in construction and structural engineering applications, which is the most modernized form of concrete featuring better durability and strength for performance in a variety of environmental conditions. Concrete compressive strength, which is dependent on many factors, such as the ingredient composition and curing process is a critical parameter concerning the quality and applicability assessment of HPC. Accurate modeling and prediction of concrete compressive strength are crucial for the optimization of material design, ensuring safety and cost efficiency in construction projects [1].

Traditional compressive strength predictive methods have often been empirically formulated or experimentally tested, usually in an extremely time and resource-intensive manner and often narrowly scoped. However, the recent stride in the use of technologies that utilize machine learning is comparatively plausible for the examination of the complex, nonlinear relations existing between many variables using techniques of predictive modeling. Data-driven approaches make for sensitivities of concrete compressive strength to ingredients and factors of processing [2].

In this project, a dataset consisting of 1,030 instances was sourced from the UCI Machine Learning Repository and used for modeling and prediction of the compressive strength of High-Performance Concrete. The dataset consists of nine major input variables: cement content, blast furnace slag, fly ash, water content, fine aggregate, superplasticizer, age of concrete, and so on[4]. The project study seeks to identify the most influential factors on compressive strength using regression machine learning models such as Random Forest, Gradient Boosting, and other regression techniques for accurate predictions. Further, a deep learning model will be developed using Keras with TensorFlow to improve predictability [3].

The results obtained illustrate that machine learning models improve not only the prediction accuracy but also provide valuable relationships between input variables and compressive strength. This work underlines the potential that machine learning techniques have for moving forward with the design and performance optimization of high-performance concrete.

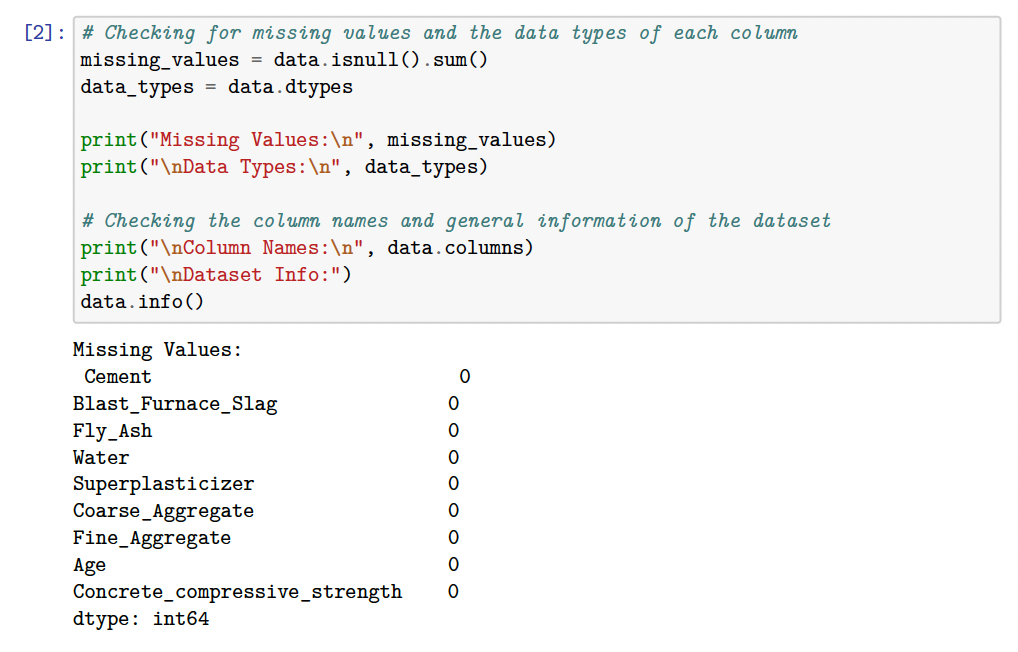
# **DATA OVERVIEW**

The dataset for performing the analysis on concrete compressive strength was obtained from the UC Irvine Machine Learning Repository, which is a collection of databases for the machine learning community to perform empirical analysis using machine learning algorithms and is very well known for providing real-time high-quality datasets for various research studies. This dataset focuses mainly on the information relevant to concrete which is the most important material in civil and construction engineering. The data set consists of 1030 instances with 8 features capturing all the important factors that derive concrete compressive strength. The 8 features comprised cement, blast furnace slag, Fly Ash, water, superplasticizer, coarse aggregate, fine aggregate, and Age of concrete. The variables varied in data type as some of them were continuous, and some Integers. The analysis of the dataset provided 8 quantitative input variables, and 1 quantitative output variable in the form of concrete compressive strength. The data set has no missing values which attests to accuracy, reliability, robustness, and dependability on the data to perform the predictive modeling analysis [4].

# **DATA PREPROCESSING**

In the data preprocessing steps, there were three major steps involved for preparing the data set for accurate predictive modeling, which are removal of missing values, normalization using min max scaler, and then the data split for training and testing. As a first step for data preprocessing, the file was imported into Python library and the sklearn library was utilized for performing all the required[3].

Initially computed to check if there are any missing values. The data set happened to be very reliable, and accurate, so there were no missing values were found.



**Figure 1: Missing Values Computation**

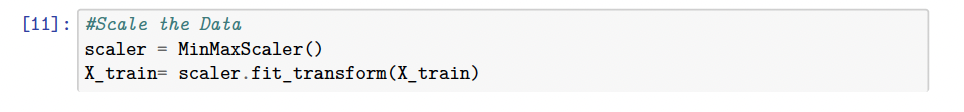
Then, the data split functioning was performed to split the data into training and testing data as it is an important aspect in machine learning modeling as we perform the predictive analysis on the training data set and evaluate the model testing on the test data set scenarios. This helps in understanding the model’s accuracy and generalizes how well the model behaves for the unseen data.

A screenshot of a computer program

Description automatically generated

**Figure 2: Splitting data into train and test data**

Lastly, the critical aspect of data preprocessing was the normalization of features to improve model performance and accuracy. The min-max scaler was used to perform the scaling on all the features. By doing so, all the features were transformed to a value in the range of [0,1].



**Figure 3: Normalization of data using min-max scaler**

# **EXPLORATORY DATA ANALYSIS (EDA)**

EDA  helps to understand the structure and quality of the data set, identifies the key variables, and establishes the relationships among features and concrete compressive strength. This helps in the detection of missing values, outliers, and multicollinearity issues, which will further guide the necessary preprocessing steps. EDA also informs feature engineering, helps validate assumptions for machine learning models, and optimizes the data for accurate predictions. With such a dataset, EDA gives insights into the conditions that control high-performance concrete compressive strength and offers model reliability and interpretability[5].

Pair plots were plotted to pictorially represent the relationship and interaction between features. These plots help in identifying the trend, outlier detection, and highlighting the correlation that would influence the target variable. In addition, pair plots provide insight into multivariate relationships; they give a comprehensive view of pairwise comparisons of features, which is essential for guiding feature selection and understanding data behavior in the pre-processing stage.

A screenshot of a graph

Description automatically generated

**Figure 4: Pair Plot between features**

Box plots were created for the analysis of this dataset because it gives a clear visualization of the distribution and spread of key variables, such as cement content and compressive strength. They help detect outliers that could adversely impact model accuracy and allow for easy comparison of variability across different features. Box plots also summarize such key statistical measures as median, quartiles, and ranges, thus providing compact representations regarding the central tendency and dispersion of the data. That is why they are so valuable in the identification of patterns and quality assurance of data at the preprocessing stage.

A graph showing a box plot

Description automatically generated

**Figure 5: Box plots for features**

A correlation heatmap was developed to emphasize relationships among features and the target variable of compressive strength to pinpoint the key influencers, such as cement, age of concrete, and water content. This finds multicollinearity among features for feature selection and reduction of dimensionality, simplifying model development by focusing resources on impactful features, reducing redundancies, and improving predictive accuracy and efficiency. The color-coded representations of the correlation between features facilitated the process of understanding the strong and weak relationships between features.

A chart of different colors

Description automatically generated with medium confidence

**Figure 5: Correlation Heat Map between features**

# **MODEL TRAINING AND EVALUATION**

In this phase of the project, several different models have been trained and evaluated on their performance against each other to predict the compressive strength of concrete. Models including Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine, K-Nearest Neighbors, and Artificial Neural Network each with various methods in predictive analytics had been used[3]. Therefore, the selected models provided a comprehensive comparison methodology in regression tasks. The metrics applied for model performance quantification included the lowest MAE, the lowest RMSE, and the highest R² Score to ensure that accuracy and reliability are thoroughly judged[6].

A graph of performance metrics

Description automatically generated

**Figure 6: Performance of Machine Learning models**

This bar chart represents the performance metrics of different models, namely Decision Tree, KNN, Random Forest, Gradient Boosting, SVR, and ANN, evaluated on MAE, RMSE, and R2 score. Inferences that could be derived from this graphical comparison are as follows:

MAE and RMSE: The lower the value, the better for both. Thus, Gradient Boosting and Random Forest have comparatively low MAE and RMSE values when compared to the rest. R2: The higher the score, the better the goodness-of-fit. Gradient Boosting and Random Forest probably have better performance in explaining the variation of the target variable.

From this comparison, the relative strengths and weaknesses of the different models emerge and can thus be chosen accordingly based on the accuracy and reliability required. Let me know if you'd like a detailed analysis of the chart.

Performance Summary of the model depicts that the Random Forest performs best overall with the lowest MAE of 3.71 and lowest RMSE of 5.44 as it denotes that it minimizes larger errors better than others, and the highest R2 of 0.89 explains 89% of the variance in data. Gradient Boosting provides results closer to Random Forest, with a little higher RMSE of 5.49 and a little lower R2 of 0.88. Applying more tuning methods might make this a good alternative model. As for KNN and SVR models, they performed terribly with both having the highest MAE and RMSE, R2 values are much lower, indicating they don't explain the variance in the data well. The Artificial Neural Network and Decision Tree models both perform average but do not yield as good of a result as Random Forest or Gradient Boosting.

# **FEATURE IMPORTANCE**

The purpose of identifying the importance of features is that, for improving model accuracy by having reduced noise, it is effective in avoiding overfitting. It helps increase model interpretability by including variable contributions, optimizes resource utilization by focusing on high-impact features, and helps drive data-driven decisions[5]. In this context, concrete compressive strength can be tuned to its key ingredients to achieve optimal performance by using feature importance. The analysis was conducted on the Random Forest Model, as it was identified as the best model approach.

A graph with blue squares

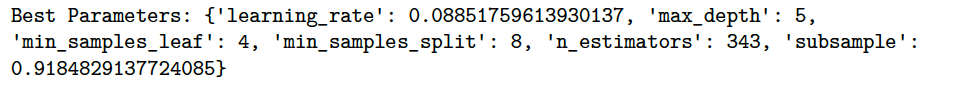
Description automatically generated

**Figure 7: Feature Importance.**

Key findings form the feature importance analysis was that the Age and Cement were the top ones identified as the most important features, with nearly equal importance(~0.35 each). These features have a strong influence on the target variable and drivemost of the model’s predictions.Water shows moderate importance (~0.15), indicatingit also plays a significant role but less than Age and Cement. Features like Fly\_Ash, Coarse\_Aggregate, and Fine\_Aggregate have minimal contribution. These features may have limited predictive power for this dataset.

# **HYPERPARAMETER TUNING**

Based on our observations, Gradient Boosting comes very close to Random Forest in terms of performance. To further explore its potential, conducting hyperparameter tuning on Gradient Boosting using RandomizedSearchCV.



**Figure 8: Best parameters for gradient boosting**

The key findings from the hyperparameter tuning were that the Gradient Boosting performance metrics increased which resulted in Gradient Boosting RMSE value of 4.44 and Gradient Boosting R² value of 0.92 compared to Random Forest RMSE value of 5.44 and Random Forest R² value of 0.89.

Gradient Boosting performs better overall with lower RMSE and higher R² value, and the hyper parameter tuning shows an improved performance but still slightly lags behind the random forest in terms of overall RMSE and R² values. Despite this, Gradient Boosting remains a strong alternative and performs competitively, especially in scenarios where fine-tuning and flexibility in learning rates are prioritized[7].

# **DEEP LEARNING MODEL IMPLEMENTATION**

Analysis was then performed to include a deep learning model implementation using keras via TensorFlow library to understand and observe the predictability nature of the data set[8]. Deep learning models tend to extract the relevant features form the raw data reducing the manual engineering of feature identification allowing in the discovery of nuanced patterns [9]. Considering there were some unidentified closer features which impacted the overall concrete compressive strength analysis, an additional model evaluation was considered to observe the effects when compared to the regression models [10].

A graph showing a graph of loss

Description automatically generated with medium confidence

**Figure 9: Model Loss during deep learning model training**

The key findings from deep learning model performance were that it yielded a Mean Absolute Error (MAE) value of 4.64*,* Root Mean Squared Error (RMSE) value of 6.03 and R² Score value of 0.86. The model did yield a competitive result when compared to Random Forest and gradient boosting, and could be considered an average performer. The conclusion can be driven that addition of more possible features with hyperparameter tuning higher datasets volume might yield good results on the deep learning models.

# **CONCLUSION**

A table with numbers and text

Description automatically generated

**Figure 10: Metrics Comparison for all models**

A graph of different colored bars

Description automatically generated with medium confidence

**Figure 11: Model Performance Comparison**

Of these, the Random Forest model had the best result in the prediction of concrete compressive strength and had the highest generalization by balancing bias and variance. It is robust, interpretable via feature importance analysis, and less prone to overfitting compared to single models like Decision Trees. Another strong competitor is Gradient Boosting; it may well be that further improvements could be achieved by tuning other hyperparameters, such as the learning rate or the number of estimators. Deep learning models are promising but require more significant datasets and tuning to show their real capabilities. On the other hand, SVR and KNN performed considerably worse and are not recommended for further comparisons without substantial tuning. Feature importance analysis explained that Age and Cement were the biggest contributing predictors, with close contributions of about 0.35, while that of Water was about 0.15, but rest of the features like Fly\_Ash, Coarse\_Aggregate, and Fine\_Aggregate had very negligible impacts. Again, this consolidates the choice of Random Forest as the best model for this dataset.

# **FUTURE SCOPE**

This project illustrates how ANN and ensemble methods could be fruitfully employed for accurate prediction of concrete compressive strength. This study can be further extended to enrich the dataset by considering a wide variety of concrete mixtures along with environmental considerations and industrial parameters such as curing time and temperature. The model’s generalization could be thereby enhanced. Besides, neural networks of more advanced architectures, such as convolutional or recurrent nets, may capture more involved relationships of the data. Second, it might be possible to enhance the performance of the predictive models by optimizing the ensemble methods through techniques such as stacking or hybrid methods. Finally, embedding these models in practical construction workflows-for example, automatic quality control and predictive maintenance systems significantly enhance the efficiency and sustainability of construction processes.

# **REFERENCES**

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

[2] Popovics, S. (1990). ANALYSIS OF THE CONCRETE STRENGTH VERSUS WATER CEMENT RATIO RELATIONSHIP*. Aci Materials Journal, 87, 517-529.*

[3] Al Yamani, W.H., Ghunimat, D.M., & Bisharah, M.M. (2023). Modeling and predicting the sensitivity of high-performance concrete compressive strength using machine learning methods.*Asian Journal of Civil Engineering, 24, 1943-1955.*

[4] Yeh, I. (1998). Concrete Compressive Strength [Dataset]. UCI Machine Learning Repository. *https://doi.org/10.24432/C5PK67.*

[5] Torre, A., García, F., Moromi, I., Espinoza, P., & Acuña, L. (2015). Prediction of compression strength of high performance concrete using artificial neural networks. *Journal of Physics: Conference Series, 582*.

[6] Yeh, I. (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement & Concrete Composites, 29*, 474-480.

[7] Hameed, M.M., & Alomar, M.K. (2019). Prediction of Compressive Strength of High-Performance Concrete: Hybrid Artificial Intelligence Technique. *International Conference on Applied Computing to Support Industry: Innovation and Technology*.

[8] Islam, N., Kashem, A., Das, P., Ali, M.N., & Paul, S. (2023). Prediction of high-performance concrete compressive strength using deep learning techniques. *Asian Journal of Civil Engineering*, 1-15.

[9] Zhang, P., Zhang, D., Zhao, X., Zhao, X., Iqbal, M., Tuerxunmaimaiti, Y., & Zhao, Q. (2024). Natural language processing‐based deep transfer learning model across diverse tabular datasets for bond strength prediction of composite bars in concrete. *Computer-Aided Civil and Infrastructure Engineering*.

[10] Xie, C., Wang, J., Wu, Z., Nie, S., Hu, Y., & Huang, S. (2023). Application of deep learning in civil engineering: boosting algorithms for predicting strength of concrete. *J. Intell. Fuzzy Syst., 45*, 9109-9122.