SMART SLUMP

MCT449 SELECTED TOPICS IN INDUSTRIAL MECHATRONICS

PREDICTIVE MODELING FOR CONCRETE COMPRESSIVE STRENGTH IN IOT-ENABLED CONSTRUCTION

Smart Slump	
Name	ID
Omar Ahmed Elsayed Abdelrahman	19P8638
Nourhan Emad Sayed	19P8403

Under Supervision Of:

Dr. Wael Mohamed Farouk El-Sersy

ABSTRACT

The Internet of Things (IoT) has emerged as a transformative technology with significant applications in industrial and construction sectors. In industrial settings, IoT enables the creation of smart factories through the interconnection of machinery, sensors, and systems, facilitating real-time monitoring, predictive maintenance, and process optimization. This enhances production efficiency, reduces downtime, and improves overall operational performance.

In construction, IoT solutions revolutionize project management, safety, and resource utilization. IoT-enabled sensors embedded in construction equipment, materials, and infrastructure provide real-time data on progress, quality, and environmental conditions, enabling proactive decision-making and risk mitigation. This leads to improved project timelines, cost efficiencies, and enhanced worker safety.

Moreover, IoT technologies empower construction companies to implement smart building solutions, incorporating sensors for monitoring energy consumption, occupancy levels, and environmental quality. This facilitates sustainable building management, optimized resource usage, and improved occupant comfort and safety.

By leveraging IoT in industrial and construction applications, organizations can drive innovation, enhance productivity, and achieve greater operational excellence in an increasingly connected and data-driven environment.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

The construction industry is undergoing a profound transformation propelled by technological advancements, and the integration of Internet of Things (IoT) solutions is revolutionizing traditional practices across various facets of construction operations. Concrete batching plants, crucial hubs in the construction supply chain, are ripe for IoT implementation due to their central role in producing the primary building material—concrete. This introduction explores the burgeoning significance of IoT in concrete batching plants within the construction industry, highlighting its potential to enhance efficiency, quality control, and sustainability throughout the concrete production process.

Figure 1 shows a Concrete Batching Plant

Concrete, being the most widely used construction material globally, demands stringent quality control measures to ensure structural integrity and durability. Traditional concrete batching plants rely heavily on manual intervention and periodic testing to monitor and adjust material proportions, mixing processes, and environmental conditions, which often leads to inefficiencies, inconsistencies, and quality discrepancies. However, with IoT-enabled sensors and automation technologies, concrete batching plants can transition into smart, data-driven facilities capable of real-time monitoring, analysis, and optimization of critical parameters.

The utilization of IoT in concrete batching plants offers multifaceted benefits. Firstly, it enables precise monitoring of raw material quantities, moisture levels, and mixing parameters, ensuring adherence to desired concrete specifications and standards. Real-time data collection from sensors embedded within batching equipment, silos, and conveyors facilitates proactive decision-making, minimizing errors and optimizing resource utilization. Furthermore, IoT-driven automation streamlines production workflows, reducing manual interventions, labor costs, and operational risks associated with human error.

Moreover, IoT solutions empower concrete producers to implement predictive maintenance strategies, preemptively identifying equipment malfunctions, and optimizing maintenance schedules to prevent costly downtime and disruptions in production. By harnessing historical and real-time data insights, concrete batching plants can continuously improve their processes, driving efficiency gains, and enhancing overall productivity.

Beyond operational enhancements, IoT integration in concrete batching plants contributes to sustainability goals by enabling energy-efficient operations, waste reduction, and carbon footprint minimization. Real-time monitoring of energy consumption, emissions, and material usage enables plant operators to identify opportunities for optimization and implement eco-friendly practices, aligning with regulatory requirements and industry sustainability initiatives.

In summary, the adoption of IoT in concrete batching plants marks a significant paradigm shift in the construction industry, offering unprecedented levels of automation, efficiency, and quality control. As construction projects increasingly demand higher performance standards and sustainability targets, IoT-enabled concrete production facilities emerge as indispensable assets, driving innovation and competitiveness in the evolving landscape of modern construction practices.

2.0 ABOUT CONCRETE BATCHING PLANTS

Concrete batching plants play a crucial role in the construction industry by efficiently producing high-quality concrete for a wide range of applications. Let's delve deeper into the components and equipment of a typical concrete batching plant, as well as the flow of the batching process and the ingredients involved.

2.1 Components and Equipment of a Concrete Batching Plant:

- Aggregate Bins: These are large storage containers used to store various sizes of aggregates such as gravel, crushed stone, or sand.
- Conveyor Belts: Conveyor belts transport aggregates from the aggregate bins to the mixing unit.
- Weighing System: This system accurately measures the quantities of aggregates, sand, cement, and water used in the concrete mix.
- Mixing Unit: The heart of the batching plant, where the ingredients are combined and mixed thoroughly to produce the concrete. It typically consists of a mixer, which can be either a drum mixer or a twin-shaft mixer.
- Cement Silos: Storage containers for cement, which is a crucial ingredient in concrete production.
- Water Tanks: Tanks for storing water, which is added to the mix to achieve the desired consistency.
- Admixture Storage: Containers for storing additives or admixtures, which are used to modify the properties of the concrete, such as setting time or strength.

2.2 Flow of the Batching Process:

- Aggregates Weighing and Feeding: The process begins with aggregates being weighed individually and fed into the conveyor belts that transport them to the mixing unit.
- Cement and Additives Addition: Simultaneously, cement and any additives or admixtures are measured and added to the mixing unit.
- Mixing: In the mixing unit, the aggregates, cement, water, and additives are combined and mixed thoroughly to form the concrete mixture. The mixing process ensures uniform distribution of all ingredients and the desired consistency of the concrete.
- Quality Control: Throughout the batching process, various sensors and monitoring systems continuously monitor the quality and consistency of the concrete mixture, allowing for adjustments if necessary.
- Discharge: Once the mixing is complete, the ready-mix concrete is discharged from the mixing unit into trucks or concrete pumps for transportation to the construction site.

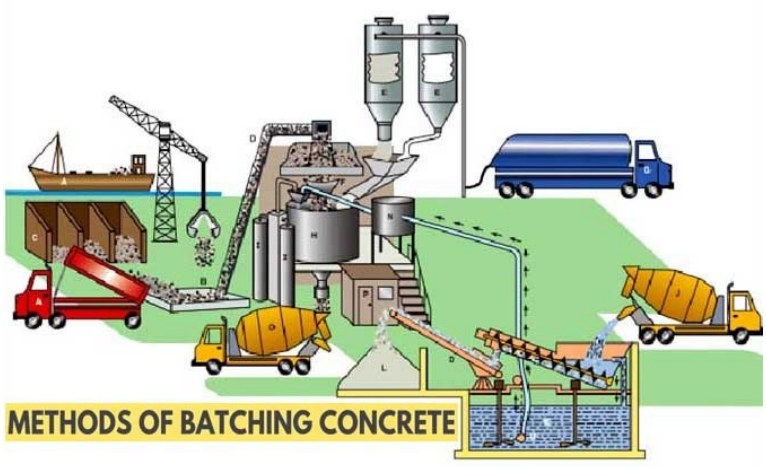

Figure 2 shows Batching Process Flow

2.3 Concrete Ingredients:

- Aggregates: These are the inert materials, including gravel, crushed stone, or sand, which provide bulk and stability to the concrete.
- Cement: The binding agent that holds the concrete together. Cement is typically a mixture of limestone, clay, and other minerals that are heated to form a powder.
- Water: Water is added to the mix to initiate the chemical reaction between cement and water, known as hydration, which causes the concrete to harden and set.

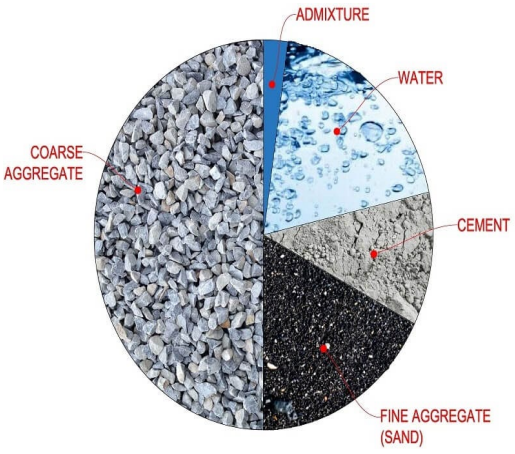

Figure 3 shows Concrete Mix Components

❖ Additives/Admixtures: These are optional ingredients added to the concrete mix to enhance certain properties such as workability, strength, durability, or setting time.

By efficiently combining these ingredients using state-of-the-art equipment and precise control systems, concrete batching plants ensure the consistent production of high-quality concrete for diverse construction applications, contributing to the success and durability of infrastructure projects worldwide.

2.4 Concrete Slump

Concrete slump is a measure of the consistency or workability of freshly mixed concrete. It refers to the vertical displacement or "slump" of the concrete when a slump cone is removed vertically from the center of the concrete mass.

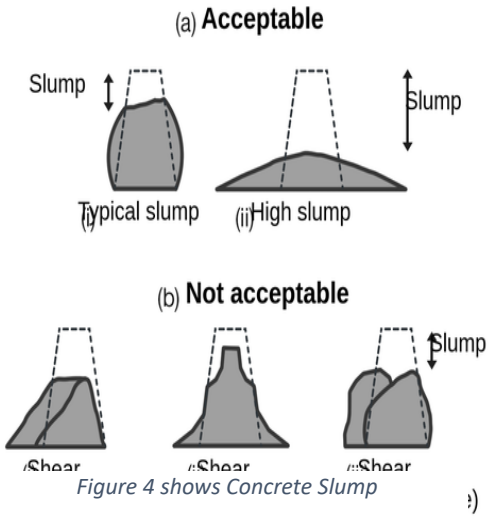

Figure 4 shows Concrete Slump

The slump test is a standard procedure used in the construction industry to assess the quality and workability of concrete batches. During the test, a slump cone is filled with freshly mixed concrete in layers and then lifted vertically upwards. After the cone is removed, the amount of settlement or "slump" of the concrete is measured in millimeters from the original height of the cone. This measurement indicates the degree of plasticity and flowability of the concrete mixture.

Concrete with a higher slump value is more fluid and easier to work with, making it suitable for applications such as pouring into molds or forming into shapes. Conversely, concrete with a lower slump value is stiffer and less flowable, which may be preferred for structural elements requiring greater stability and strength.

The desired slump value for a specific concrete mix depends on various factors such as the intended application, construction method, and environmental conditions. Achieving the correct slump is crucial to ensure that the concrete can be properly placed, compacted, and finished to meet project requirements and performance standards.

3.0 PROBLEM FORMULATION

3.1 Problem Statement

The primary challenge in the construction industry is to accurately predict the compressive strength of curing using IoT-enabled sensors and predictive modeling techniques. This entails developing algorithms and models that can effectively utilize data collected from IoT devices embedded within concrete mixers, transportation vehicles, and construction sites to forecast the compressive strength value at the end of the curing period. The prediction must be reliable and precise to guide construction professionals in making informed decisions regarding material handling, placement, and quality control measures.

3.2 Research Objectives

A. Develop IoT-enabled sensing solutions capable of continuously monitoring key parameters influencing concrete slump, such as temperature, humidity, aggregate properties, and mixing proportions.
B. Design and implement data collection and aggregation mechanisms to gather real-time sensor data from diverse sources across the construction workflow.
C. Investigate and apply machine learning and statistical modeling techniques to analyze historical and real-time data for identifying patterns and correlations between input factors and concrete compressive strength.
D. Build predictive models that can accurately forecast the compressive strength value after the curing period based on the initial mix design and environmental conditions.
E. Evaluate the performance and reliability of the predictive models through validation with independent datasets and field trials conducted in collaboration with construction industry partners.
F. Develop user-friendly interfaces and visualization tools to present the predicted slump values to stakeholders, facilitating decision-making and quality assurance processes on construction sites.

3.3 Significance of the Research

The successful development and implementation of predictive modeling for concrete compressive strength using IoT technologies have significant implications for the construction industry:

- ❖ Improved Quality Control: Enables proactive monitoring and management of concrete quality throughout the construction process, reducing the risk of defects and structural failures.
- ❖ Cost Savings: Minimizes material wastage and rework by optimizing concrete mix designs and placement strategies based on predicted compressive strength values.
- ❖ Time Efficiency: Streamlines construction timelines by providing early insights into concrete performance, allowing for timely adjustments and interventions.
- ❖ Sustainability: Promotes sustainable construction practices by optimizing resource utilization and reducing environmental impact associated with concrete production and construction activities.

4.0 ABOUT THE DATASET

The dataset's objective is to forecast the compressive strength of high performance concrete, as indicated by slump and slump flow measurements. Thus, our focal variables include:

Target Y:

✓ Concrete Compressive Strength [MPa]

Input X:

✓ Cement
✓ Fly ash
✓ Blast Furnace Slag
✓ Water
✓ Superplasticizer
✓ Fine aggregate
✓ Coarse aggregate
✓ Age

The dataset is available in the below link:

https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength

The below table shows the dataset variables table:

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
Cement	Feature	Continuous			kg/m^3	no
Blast Furnace Slag	Feature	Integer			kg/m^3	no
Fly Ash	Feature	Continuous			kg/m^3	no
Water	Feature	Continuous			kg/m^3	no
Superplasticizer	Feature	Continuous			kg/m^3	no
Coarse Aggregate	Feature	Continuous			kg/m^3	no
Fine Aggregate	Feature	Continuous			kg/m^3	no
Age	Feature	Integer			day	no
Concrete compressive strength	Target	Continuous			MPa	no

Figure5 shows Dataset Variables Table

4.1 Dataset Collection Methodology

Data collection involves creating a set of concrete samples under controlled conditions, varying factors like cement type, water-cement ratio, aggregate size, and curing duration. These samples undergo compressive strength tests using specialized equipment. The gathered data, including compressive strengths, are recorded for training machine learning models. It's crucial to adhere to standardized testing procedures and quality control measures to ensure consistency and reliability. This paper utilizes the Yeh30 standard dataset [5], which comprises 1030 concrete samples with eight features: cement, water, coarse aggregate, fine aggregate, superplasticizer, blast-furnace slag, and fly-ash. The samples were treated normally before data collection, and compressive strength was determined using conventional procedures with 150 mm-tall cylindrical specimens.

	Cement (m^3)	Slag (kg/m^3)	Flyash (kg/m^3)	Water (m^3)	sp (%)	Coarse (kg/m^3)	Fine (mm)	Age (days)	Concrete strength (psi)
0	540	0	0	162	2.5	1040	676	28	79.99
1	540	0	0	162	2.5	1055	676	28	61.89
2	332.5	142.5	0	228	0	932	594	270	40.27
3	332.5	142.5	0	228	0	932	594	365	41.05
4	198.6	132.4	0	192	0	978.4	825.5	360	44.3

Figure 6 shows Sample Records for Concrete Compressive Strength [6]

4.2 Dataset Exploration Methodology (Features Exploration)

Exploring concrete strength data entails analyzing and comprehending the gathered data to reveal patterns, correlations, and valuable insights that can aid in effectively training machine learning models. The exploration process commences with descriptive statistics, including mean, median, standard deviation, and quartiles, to grasp the data distribution and variability. Visualizations such as histograms, box plots, and scatter plots offer additional insights by demonstrating the relationships between features like cement, water, coarse aggregate, and fine aggregate, and pinpointing potential outliers or anomalies.

The below table summarizes the minimum and maximum values, mean, standard deviation (std), and quartile distribution of these attributes, serving as a condensed overview of the data exploration process for these features.

	Cement (m^3)	Slag (kg/m^3)	Flyash (kg/m^3)	Water (m^3)	sp (%)	Coarse (kg/m^3)	Fine (mm)	Age (days)	Concrete strength (psi)
Count	1030	1030	1030	1030	1030	1030	1030	1030	1030
Mean	281.167	73.89583	54.18835	181.56728	6.20466	972.91893	773.58049	45.66214	35.817961
Std	104.50	86.27934	63.997	21.354219	5.973841	77.753954	80.17598	63.16991	16.705742
Min	102	0	0	121.8	0	801	594	1	2.33
25 %	192.37	0	0	164.9	0	932	730.95	7	23.71
50 %	272.9	22	0	185	6.4	968	779.5	28	34.445
75 %	350	142.95	118.3	192	10.2	1029.4	824	56	46.135
Max	540	359.4	200.1	247	32.2	1145	992.6	365	82.6

Figure 7 shows Features Summarization [7]

4.3 Statistical Distribution of Relevant Features

The below figures exhibit histograms illustrating the statistical distribution of pertinent features. Each feature is represented on the x-axis, while the frequency of occurrences is indicated on the y-axis. This visualization facilitates a thorough assessment of these features.

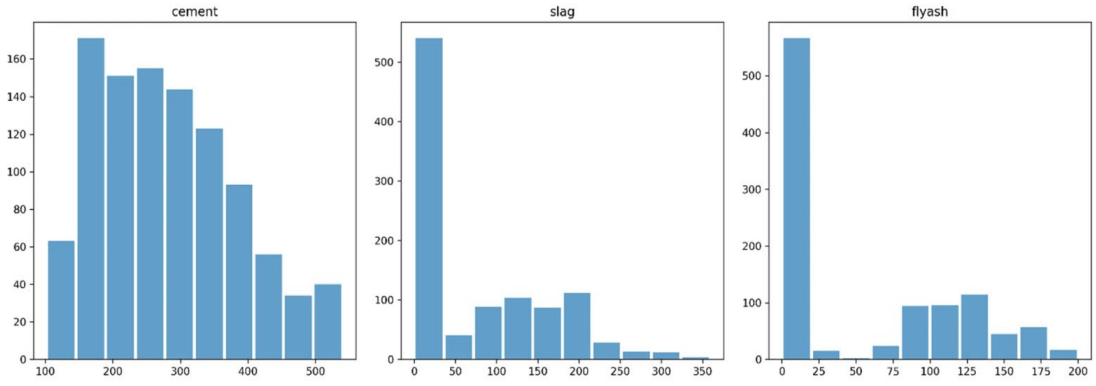

Figure 8 shows Cement, Slag, Flyash Statistical Distribution

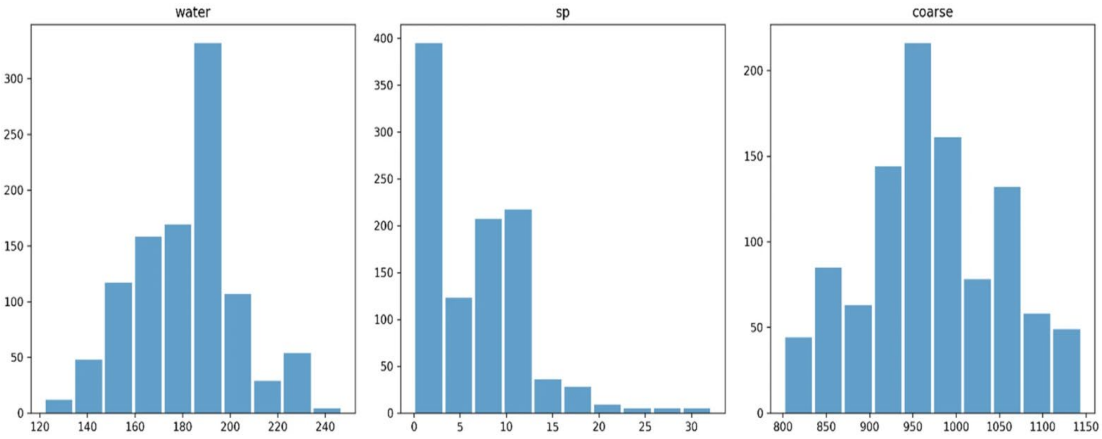

Figure 9 shows Water, sp, Coarse Statistical Distribution

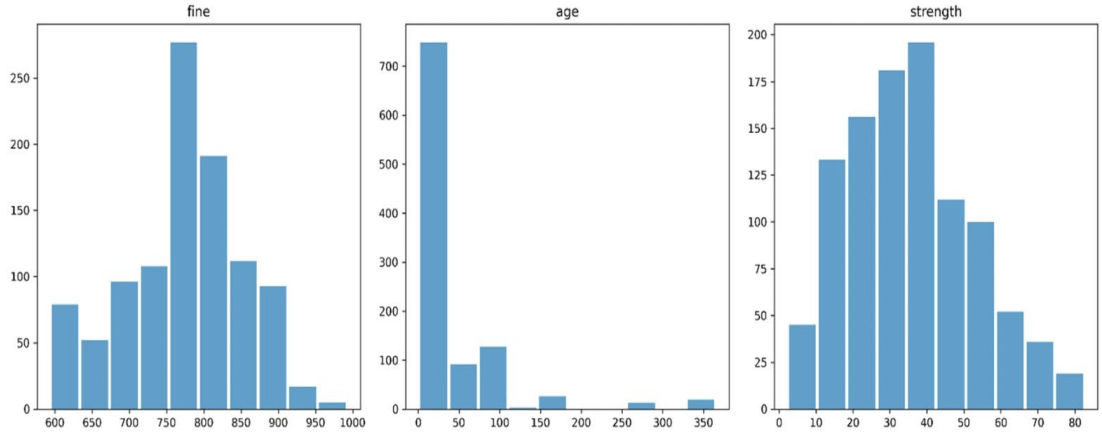

Figure 10 shows Fine, Age, Strength Statistical Distribution

Key observations include:

- Cement's distribution closely resembles a normal distribution.
- Blast-furnace slag ('slag') exhibits proper skewness and appears to follow a distribution with three peaks.
- Fly-ash ('flyash') is right-skewed and displays a bimodal distribution with two peaks.
- Water's distribution shows three peaks with a leftward tilt.
- Superplasticizer ('sp') demonstrates a distribution with two peaks and proper skewness.
- Coarse aggregate ('coarse') distribution is nearly normal and displays three peaks.
- Fine aggregate ('fine') distribution appears to be bimodal with two peaks, indicating a non-normal distribution.

The age feature showcases multiple peaks and a skewed distribution, which seems fitting for the dataset.

4.4 Correlation Between Features

Examining the correlation between features is crucial for comprehending the relationships between dependent features and the target strength factor, aiding in the identification of the optimal prediction model. In the below figure, a heatmap illustrates each variable's impact on all other variables. Notably, a strong correlation is evident between cement and strength, indicating that cement serves as a highly reliable predictor. Conversely, slag and fly-ash exhibit weak correlations with the target variable. Additionally, it's noteworthy to highlight the significant positive correlation between superplasticizer and fly-ash, contrasting with the relatively weaker correlation between superplasticizer and compressive strength. Remarkably, there exists a substantial negative correlation between water and superplasticizers, as well as between water and strength.

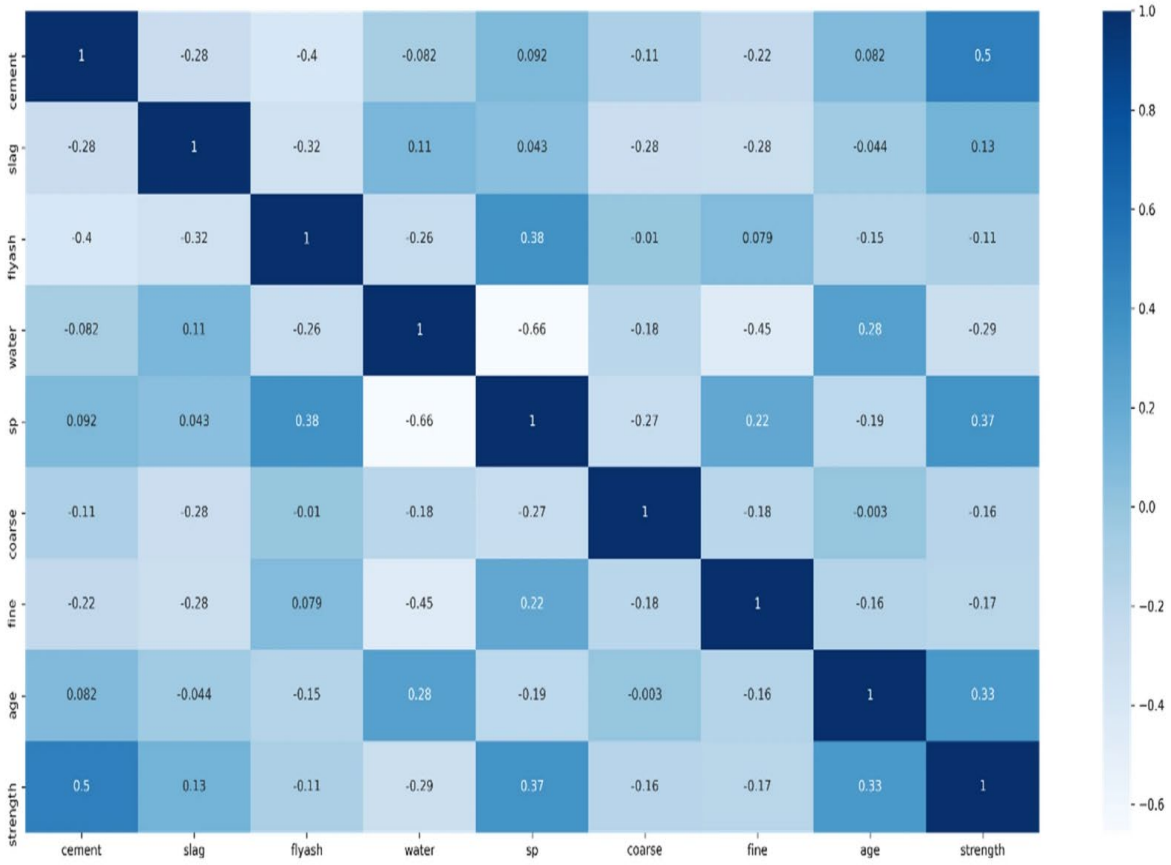

Figure 11 shows Features Correlation [8]

4.5 Features Correlation Pair Plot

The below figure, depicted as a Pair Plot, provides a visual representation of the correlation information among the features. This thorough analysis is critical because dimensions with strong correlations, approaching values near 1 or -1, are redundant, offering the model duplicated information. Consequently, we might opt to keep one dimension while discarding another. The decision regarding which dimension to retain and which to discard depends on domain expertise and an evaluation of which dimension is more susceptible to errors.

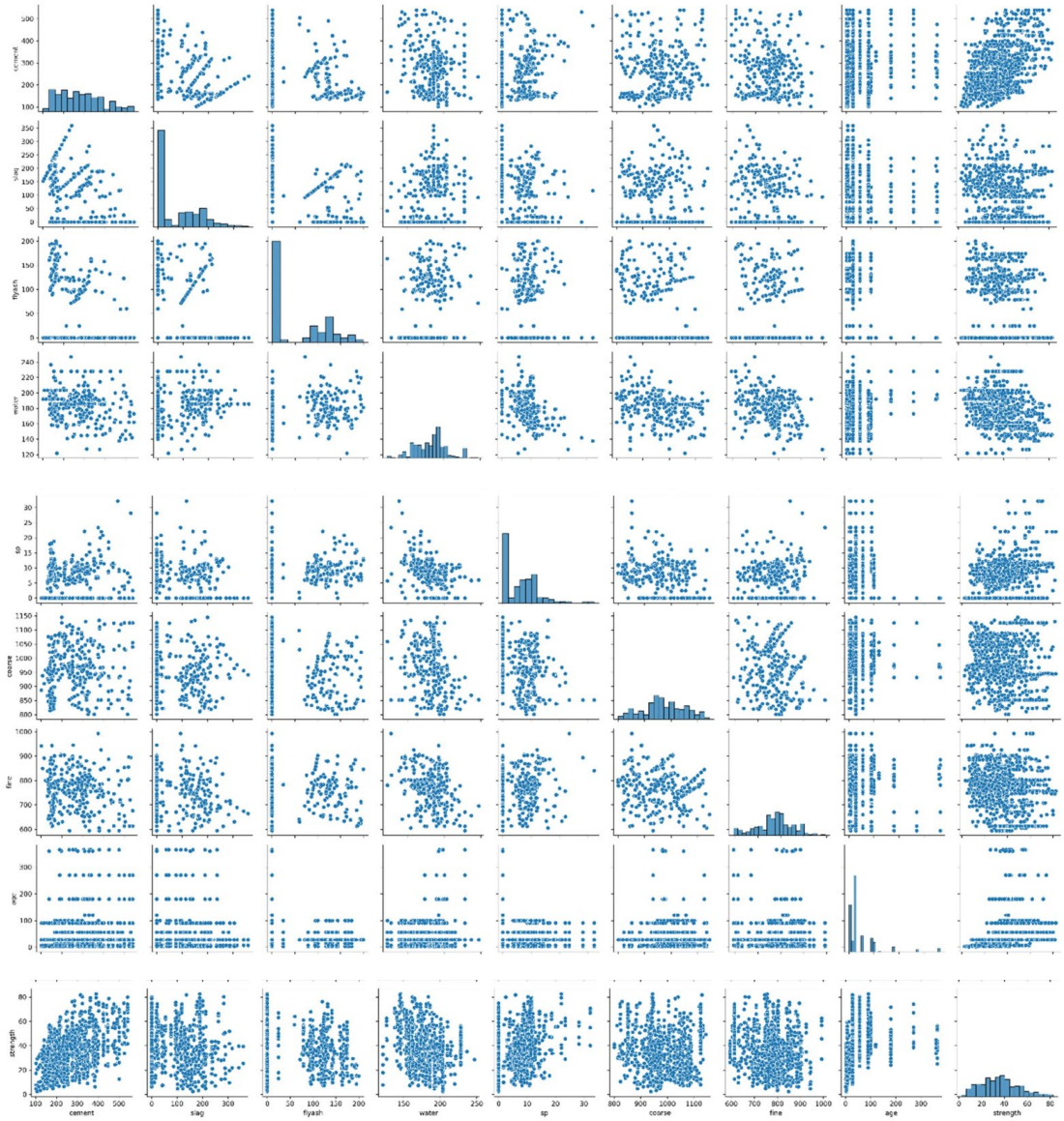

Figure 12 shows Features Correlation Pair Plot [9]

The pair plot analysis reveals the following:

- Cement shows no correlation with other features, including slag, fly-ash, water, superplasticizer, coarse aggregate, fine aggregate, and age.
- Slag also exhibits no correlation with fly-ash, water, superplasticizer, coarse aggregate, fine aggregate, or age.
- Fly-ash lacks significant correlation with water, superplasticizer, coarse aggregate, fine aggregate, or age, and shows minimal correlation with other independent attributes.
- Water has a negative linear association with superplasticizer and fine aggregate but doesn't exhibit meaningful correlation with other attributes. Notably, superplasticizers can reduce water content in concrete by up to 30% without affecting workability.
- Superplasticizer only shows a negative linear relationship with water and doesn't have a strong correlation with other variables.
- Coarse aggregate, like other attributes, doesn't display significant correlation with any other variables.
- Fine aggregate displays a linear inverse relationship with water and doesn't show meaningful correlation with other characteristics.

4.6 Scatter Plot Between Input Features and Target Predicted Variable

Figures 13 and 14 showcase scatter plots depicting the relationship between compressive strength, the target predicted variable, and input feature variables such as cement, water, age, and fly-ash. In Figure 14, a positive correlation is observed between cement content and compressive strength. As the cement content increases, so does the compressive strength of the concrete. Additionally, it's noted that as concrete ages, its strength increases, requiring more cement to achieve higher strength levels at a younger age. Conversely, older cement requires more water, so reducing water content in concrete enhances its strength. Figure 9's scatter plot reveals an inverse relationship between compressive strength and fly-ash content. The concentration of darker dots in the lower compressive strength values region underscores this relationship. Conversely, it's demonstrated that utilizing a superplasticizer enhances compressive strength, indicating a positive association between these two parameters.

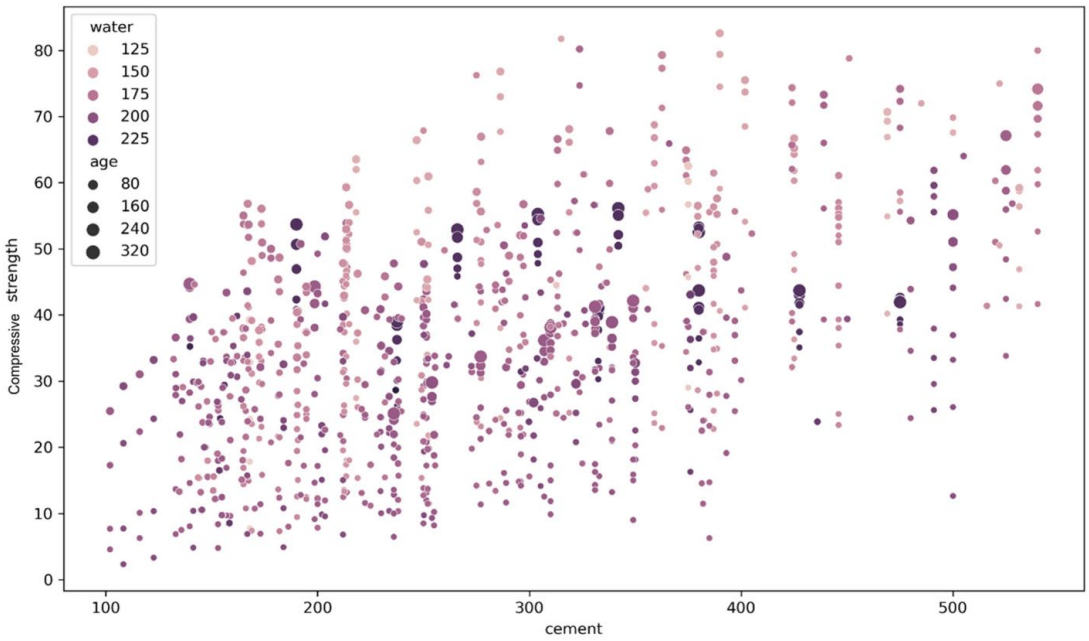

Figure 13 shows Scatter Plot of Cement vs Compressive Strength [10]

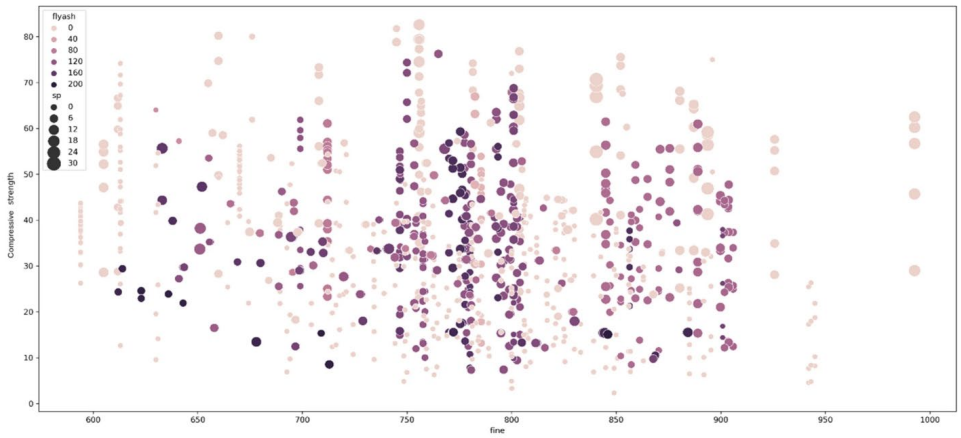

Figure 14 shows Scatter Plot of Fine Aggregates vs Compressive Strength [11]

5.0 RELATED RESEARCH METHODOLOGY

The adopted research that will be discussed thoroughly is the "Unboxing Machine Learning Models for Concrete Strength Prediction using XAI" paper by S.Elhishi and A. El-Ashry. The machine learning framework for predicting concrete strength consists of five key stages, as depicted in Figure 15. Initially, the data collection process involves preparing concrete samples under controlled conditions, varying factors like cement type, water-cement ratio, aggregate size, and curing duration—these factors serve as input features for the machine learning model. Following this, data exploration is conducted to analyze and comprehend the collected data, aiming to unveil patterns, relationships, and insights essential for training machine learning models effectively. Subsequently, the data preprocessing stage is undertaken to eliminate noise, address missing values, clean the data, and format it appropriately for training machine learning models. We then train eight machine learning models, encompassing statistical regression, ensemble learning, SVM, and ANN, to predict concrete strength, and evaluate the performance of these models accordingly.

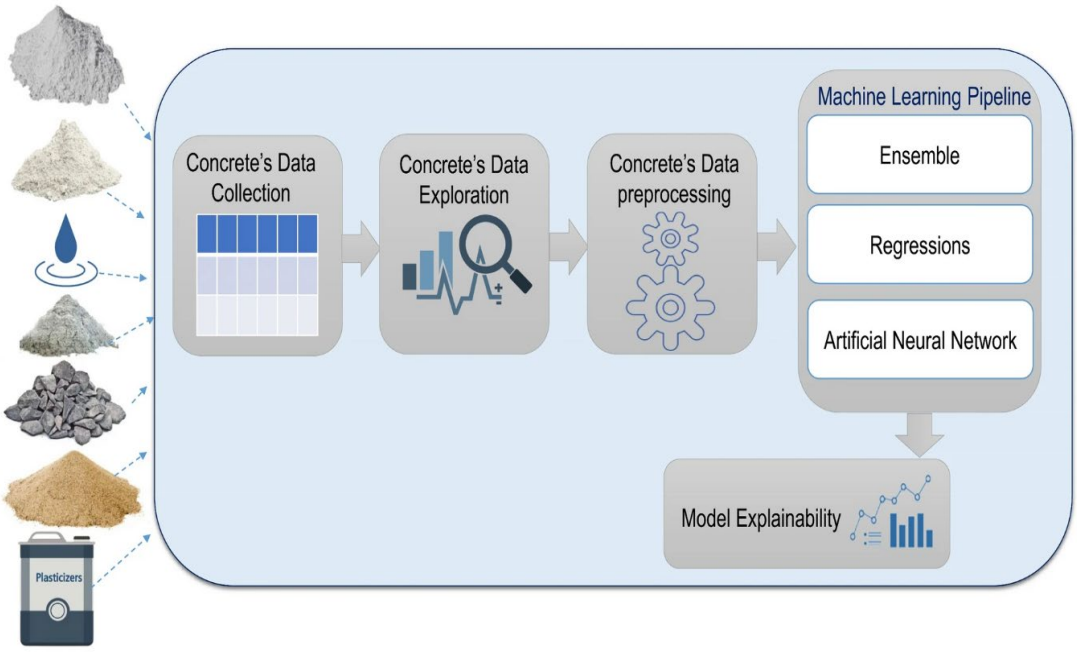

Figure 15 shows the Machine Learning Pipeline[12]

As outlined, previous studies on concrete strength prediction predominantly rely on three main methodologies: statistical regression techniques, ensemble learning based on Decision Trees, and the utilization of Artificial Neural Networks (ANNs). Our decision to incorporate representative machine learning models from these established approaches stems from our assessment of their effectiveness in achieving accurate and robust predictions. For example, we include Linear Regression as a fundamental model to act as a baseline for regression tasks, given its assumption of a linear relationship between input features and the target variable, which makes it a valuable initial exploration tool. LASSO Regression addresses potential multicollinearity issues within the dataset and enhances model interpretability by penalizing the absolute values of regression coefficients.

Similarly, Ridge Regression, another regularization method, mitigates overfitting and enhances model stability by introducing a penalty term to the loss function.

Decision Trees are renowned for their ability to capture complex non-linear relationships and interactions among features, rendering them suitable for datasets characterized by intricate decision boundaries. We also leverage Random Forests, an ensemble method, to capture complex patterns within the data, which is advantageous for concrete strength prediction. Additionally, we consider XGBoost, an algorithm based on gradient boosting, due to its robustness in handling missing data and outliers, attributes that can significantly enhance concrete strength prediction. SVM is introduced to explore its potential to provide a unique perspective on concrete strength prediction, given its effectiveness in managing high-dimensional data and intricate decision boundaries. Finally, ANNs, known for their ability to decipher intricate data patterns, are integrated into our analysis to investigate whether complex, non-linear models can outperform traditional regression models in predicting concrete strength. Table 2 provides a brief comparison of different machine learning approaches for concrete strength prediction. Notably, the XG Boost model yields the best-reported results among all eight evaluated models.

Approach	Algorithm	Task	Strengths	Weaknesses
Statistical	Linear regression	Fits a linear equation to the data	Simple Interpretable Works well with linear relationships between the features and the target variable	It may not capture complex patterns Sensitive to outliers Prone to overfitting
	LASSO regression	Fits a linear equation to the data with additive regularization (L1) parameter	Prevents overfitting Performs feature selection using the regularization (L1) parameter	Sensitive to data scaling Not ideal for highly correlated features
	Ridge regression	Fits a linear equation to the data with additive regularization (L2) parameter	Prevents overfitting Reduces multicollinearity in the model Better with high-dimensional datasets	Does not perform feature selection Sensitive to feature scaling
	SVM	Maximizes the margin between classes with multiple equations	Effective for high-dimensional data Good at handling imbalanced datasets	Sensitive to parameter settings May require feature scaling
Tree-based	Decision Trees	Divides the data into branches based on feature splits	Strong performance with non-linear relationships Easy to interpret	Prone to overfitting It may create deep trees with high variance
	Random forests	Ensemble version of decision trees	Reduces overfitting Provides feature importance scores	Computationally expensive Less interpretable than a single decision tree
	XGBoost	Uses gradient boosting techniques as a modified version of the Decision tree	High predictive accuracy supports L1 (LASSO) and L2 (Ridge) regularizations Computationally efficient	Sensitive to hyperparameter tuning Less interpretable
Artificial neural network	ANN	A multi-layer network of interconnected nodes (artificial neurons)	Captures complex, non-linear relationships, Scalability and flexibility with data of different sizes	Prone to overfitting Sensitive to feature scaling Sensitive to hyperparameter tuning

Figure 16 shows a Comparison Between Different ML Approaches for Concrete Strength Prediction

6.0 DATA PREPROCESSING

Data preprocessing is essential for accurate concrete strength prediction using machine learning techniques and involves multiple critical steps:

Data cleaning: This step addresses missing values, outliers, and inconsistencies in the dataset to ensure its quality and integrity.

- Missing values can be imputed using mean or median imputation techniques to maintain dataset completeness.
- Outliers, if detected, can be treated by either removing them or replacing them with more representative values to prevent skewing the model's understanding of typical concrete properties.

Feature scaling: Ensures that all features are on a similar scale, typically achieved through normalization or standardization.

- This step is necessary because concrete-related features often have different units and scales.
- Normalization or standardization ensures that all features contribute equally to the prediction, preventing any single feature from dominating the learning process.

Handling categorical variables: In concrete strength prediction, categorical variables like the type of cement used or curing method applied are common.

- Machine learning models require numerical input, so categorical variables are encoded into numerical values using techniques like one-hot encoding or label encoding.

These preprocessing steps ensure that the dataset is appropriately prepared, enhancing the effectiveness and performance of machine learning algorithms in predicting concrete strength.

7.0 PREDICTION MODEL EXPLAINED

Explainable Artificial Intelligence (XAI) techniques are increasingly valuable in concrete strength prediction, aiding engineers and researchers in understanding the crucial factors influencing concrete strength and guiding decisions in concrete mix design and construction practices. We utilized the SHAP (SHapley Additive exPlanations) method, a prominent XAI technique, to offer insights into the factors affecting concrete strength prediction. SHAP integrates game theory and machine learning, attributing the concrete strength prediction outcome to input features such as cement, water, coarse aggregate, fine aggregate, superplasticizer, blast-furnace slag, and fly-ash. By computing SHAP values, the importance of each feature in predicting concrete strength can be quantified. These values encapsulate the additive contribution of a feature across all possible feature subsets, considering their interactions and dependencies. SHAP values facilitate the identification of the relative influence of features on the model's output, providing a deeper understanding of the underlying mechanisms.

In the realm of AI explainability for concrete strength prediction, SHAP values pinpoint which features hold the greatest significance in determining the predicted strength.

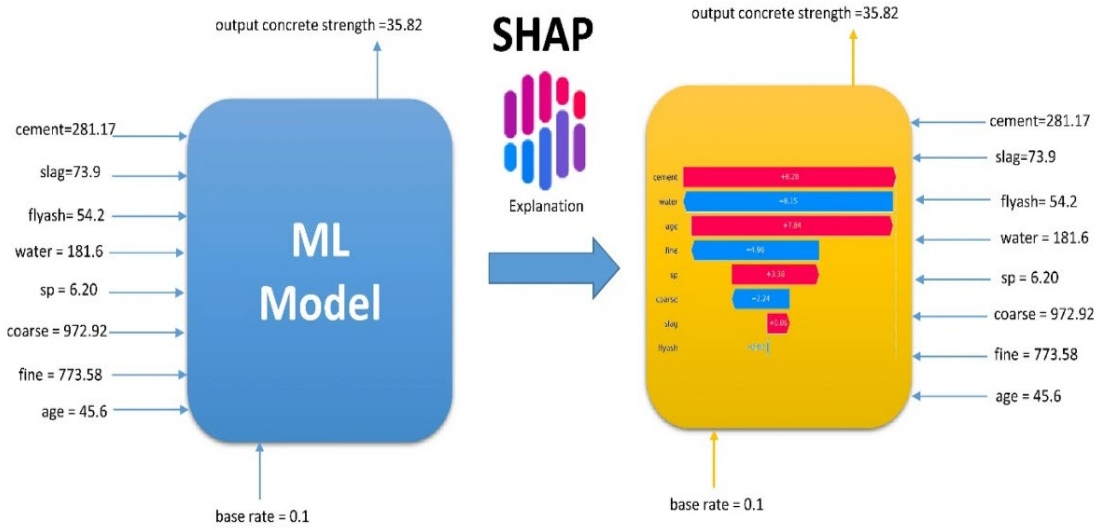

Figure 17 shows usage of SHAP for Concrete Strength Prediction

8.0 MACHINE LEARNING MODEL EVALUATION

In the study, eight machine learning models from three different approaches are assessed. These models encompass Linear Regression, Ridge Regression, LASSO Regression, Support Vector Machines (SVM), Random Forests, Decision Trees, XGBoost, and Artificial Neural Networks (ANN). The evaluation process involves using both training and testing datasets, applying various metrics including Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination, commonly known as R-squared (R2) score.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{N}\sum_{i=0}^{N}|(y_i - \hat{y}_i)|$$

MSE, RMSE, and MAE are metrics that depend on the actual data values and the predictions made by the machine learning models, while the R-squared score is based on data variance. Below are the statistical equations that describe these evaluation metrics:

$$R^2 = \frac{\sum_{i=0}^{N}\left(\hat{y}_i - \overline{Y}_i\right)^2}{\sum_{i=0}^{N}\left(y_i - \overline{Y}_i\right)^2}$$

Where y_i *is the actual value,* $\hat{y}_i$ *is the predicted value,*

and $\widehat{Y}_i$ *is the mean value of the actual values*

The below table presents the outcomes of various statistical tests conducted by the models on the dataset, according to the expected values. These results indicate the successful prediction of concrete compressive strength by all models, as evidenced by their statistical performance metrics. Notably, XGBoost achieved the highest R-squared value (R2 = 0.91), signifying its superior accuracy. Following closely, the Random Forests model achieved an R2 value of 0.89, while Decision Trees attained 0.82, and ANN reached 0.74. In contrast, the Linear Regression models demonstrated lower accuracy, with the basic model achieving an R2 value of 0.57, followed by LASSO Regression (R2 = 0.54), Ridge Regression (R2 = 0.57), and SVM (R2 = 0.66).

Machine learning approaches	Method	RMSE	MSE	MAE	R^2
Statistical	Linear regression	10.28	105.76	8.23	0.57
	LASSO regression	10.68	114.11	8.65	0.54
	Ridge regression	10.29	105.84	8.24	0.57
	SVM	9.13	83.39	7.44	0.66
Tree-based	Decision trees	6.65	44.24	4.47	0.82
	Random forests	5.21	27.17	3.53	0.89
	XGBoost	**4.37**	22.33	3.04	**0.91**
Artificial neural network	ANN	6.01	36.22	4.53	74.63

In terms of statistical error, the XGBoost model demonstrated the lowest Root Mean Square Error (RMSE) value at 4.37, indicating superior performance compared to the SVM and Regression models, which exhibited higher RMSE values around 9.13 and averaging around 10, respectively. Based on the accuracy criterion, it can be inferred that XGBoost currently stands out as the best-performing model. Figures 18 and 19 illustrate scatter plots showcasing the experimental (actual) and predicted compressive strengths of concrete, respectively.

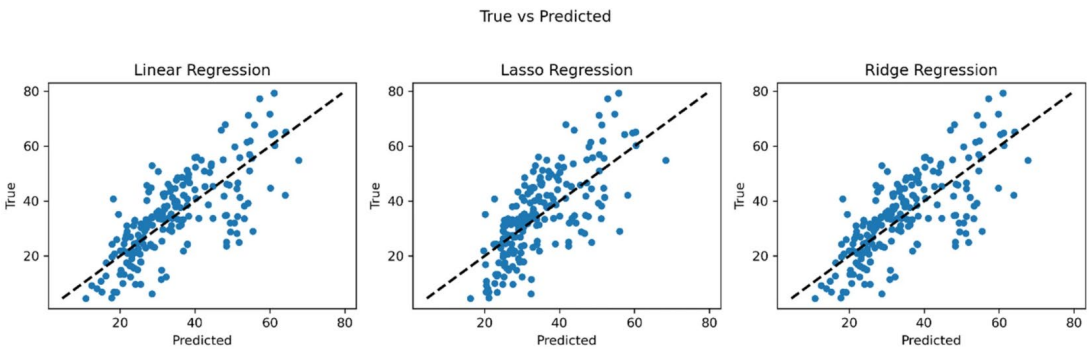

Figure 18 shows Linear, Ridge and LASSO regression models

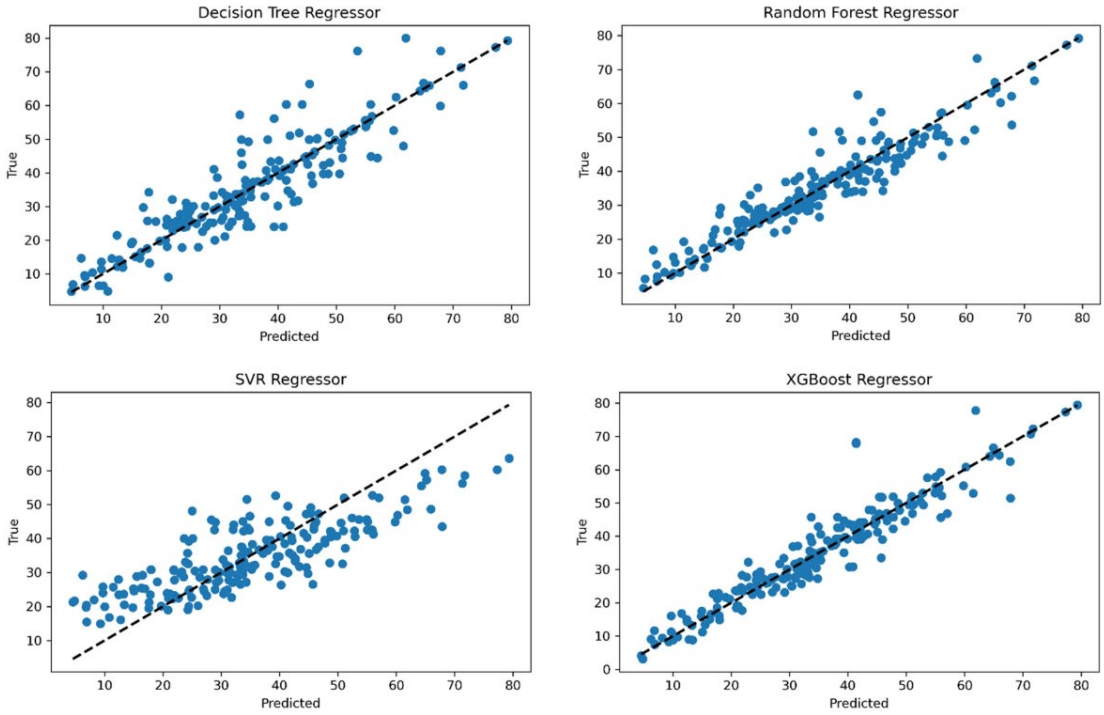

Figure 19 shows Decision trees, random forests, SVM, XGBoost models scatter plots

Figure 20 displays the R2 and RMSE values obtained from the seven models. Given the input feature variables of cement, water, coarse aggregate, fine aggregate, superplasticizer, blast-furnace slag, and fly-ash, this figure suggests that the XGBoost model may be effective and provide acceptable precision when utilized for calculating the compressive strength of concrete.

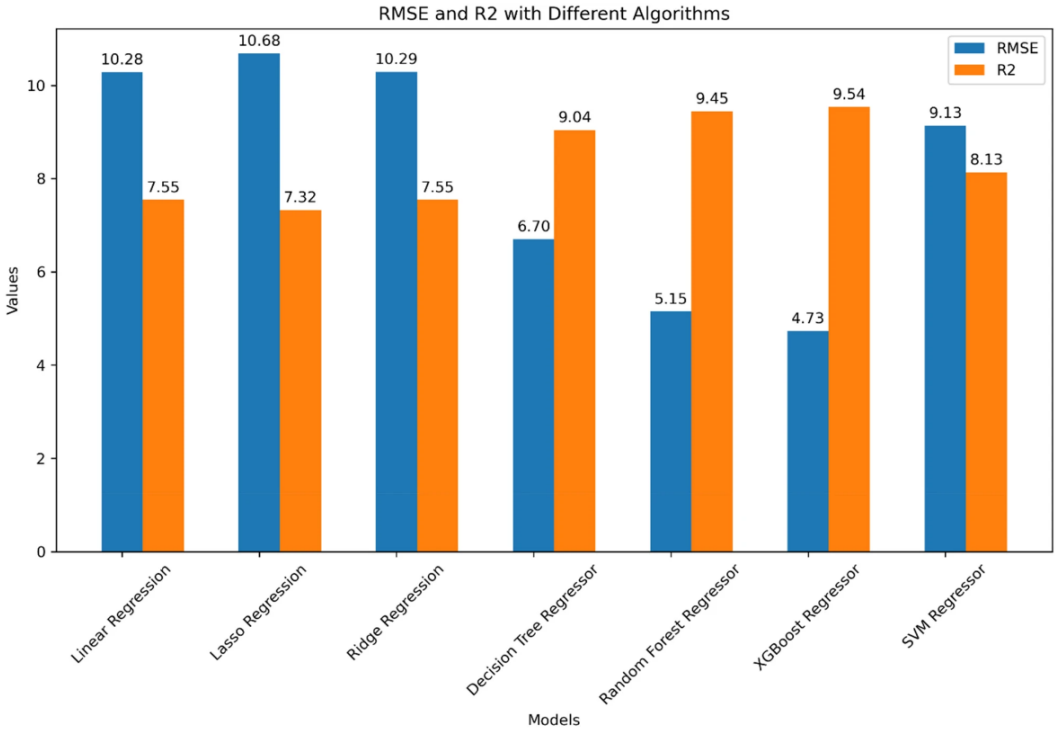

Figure 20 shows Concrete Strength Prediction Machine Learning Models Evaluation Metrics

9.0 RESULTS CONCLUSION

In summary (as depicted in Fig. 21), it is apparent that among all the tested classifiers, XGBoost outperforms others with an accuracy of 91%. Consequently, its results were selected to assess the impact of model features on the classifier output. Adhering to explainable AI methodologies, the SHAP method was utilized, offering a comprehensive analysis with supported visual explanations for the classification model. The analysis revealed that cement demonstrates the highest positive correlation with concrete strength, whereas water exhibits the most significant negative correlation with the compressive strength output. Notably, the most influential features in predicting concrete strength, ranked in descending order of importance, are cement, age, and water.

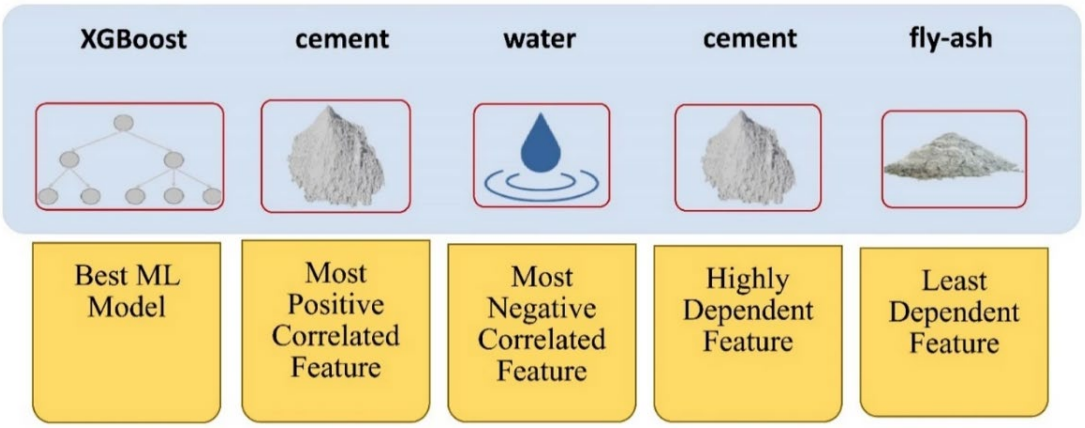

Figure 21 shows Summary of Concrete Prediction Results

10.0 RELATED WORK - RESULTS

Our work involves exploring various machine learning approaches to predict concrete strength. We focus on three main categories: statistical learning, tree-based learning, and artificial neural networks (ANNs).

Statistical learning methods, such as Linear Regression, Ridge Regression, LASSO Regression, and Support Vector Machines (SVM), aim to establish mathematical functions that accurately describe the relationship between input and output variables.

In tree-based learning, we utilize Decision Trees as fundamental building blocks to construct predictive models. These models represent input data as feature vectors containing predictor variables and a target variable. We explore Decision Trees, Random Forests, and XG Boost as examples of tree-based learning models.

Lastly, we delve into the ANN approach, inspired by the structure and function of biological neural networks. ANNs consist of interconnected neurons arranged in layers. During training, the network adjusts weights associated with connections to minimize prediction error. Once trained, ANNs can provide concrete strength predictions based on learned patterns and relationships in the training data.

Throughout our work, we implement these techniques to achieve comparable results to those outlined in existing research.

Our work is available through the following link:
https://colab.research.google.com/drive/10yNHpepxaJr2ieW_bwxL9r_njkLUcrNa?usp=sharing

10.1 Implemented Results Discussion

The project was conducted using Google Colab, a cloud-based platform that provides a convenient environment for running Python code, particularly for machine learning and data analysis tasks like the current task.

10.1.1 Sample records for Concrete strength prediction

Loading Dataset

```
[3]  # Load the dataset
     df = pd.read_excel('/content/Concrete_Data.xls')
```

Concrete Data Collection with Sample Records

```
[4]  # Display the first five rows of the dataset
     df.head()
```

	Cement (component 1)(kg in a m^3 mixture)	Blast Furnace Slag (component 2) (kg in a m^3 mixture)	Fly Ash (component 3)(kg in a m^3 mixture)	Water (component 4)(kg in a m^3 mixture)	Superplasticizer (component 5)(kg in a m^3 mixture)	Coarse Aggregate (component 6) (kg in a m^3 mixture)	Fine Aggregate (component 7) (kg in a m^3 mixture)	Age (day)	Concrete compressive strength(MPa, megapascals)
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.986111
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.887366
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.269535
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.052780
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.296075

Figure 22 Sample Records of the Concrete Dataset

10.1.2 Concrete features summarization

The below table provides detailed statistics for various features related to concrete strength prediction. The features include Cement (m3), Slag (kg/m3), Flyash (kg/m3), Water (m3), sp (%), Coarse (kg/m3), Fine (mm), Age (days), and Concrete strength (psi).

Count: Indicates the number of data points available for each feature, which is consistent at 1030.

Mean: Represents the average value of each feature across the dataset.

Std (Standard Deviation): Measures the dispersion or spread of data around the mean.

Min: Shows the minimum value observed for each feature.

25%, 50%, 75%: Correspond to the 25th, 50th (median), and 75th percentiles, respectively. These values divide the data into four quartiles and provide insights into the distribution of the data.

Max: Reflects the maximum value observed for each feature.

From the table, we can observe significant variations in the features, particularly in terms of their means, standard deviations, and ranges. For instance, Cement and Water exhibit relatively high mean values, indicating their significant presence in the concrete mixture. Slag and Flyash show notable standard deviations, suggesting variability in their usage across different concrete samples. Additionally, the quartile values provide insights into the spread of the data and the potential presence of outliers. Overall, this table serves as a valuable reference for understanding the distribution and characteristics of the dataset, which is crucial for subsequent analysis and model development in concrete strength prediction.

It is important to note that the concluded dataset features table is the exact similar of the one concluded in the research.

Concrete Data Exploration Process (Features Summarization)

```
# Calculate summary statistics
summary_stats = df.describe()

# Rename the index for better visualization
summary_stats.index = ['Count', 'Mean', 'Std', 'Min', '25%', '50%', '75%', 'Max']

# Display the summary statistics
summary_stats
```

	Cement (component 1)(kg in a m^3 mixture)	Blast Furnace Slag (component 2)(kg in a m^3 mixture)	Fly Ash (component 3)(kg in a m^3 mixture)	Water (component 4)(kg in a m^3 mixture)	Superplasticizer (component 5)(kg in a m^3 mixture)	Coarse Aggregate (component 6)(kg in a m^3 mixture)	Fine Aggregate (component 7)(kg in a m^3 mixture)	Age (day)	Concrete compressive strength(MPa, megapascals)
Count	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000
Mean	281.165631	73.895485	54.187136	181.566359	6.203112	972.918592	773.578883	45.662136	35.817836
Std	104.507142	86.279104	63.996469	21.355567	5.973492	77.753818	80.175427	63.169912	16.705679
Min	102.000000	0.000000	0.000000	121.750000	0.000000	801.000000	594.000000	1.000000	2.331808
25%	192.375000	0.000000	0.000000	164.900000	0.000000	932.000000	730.950000	7.000000	23.707115
50%	272.900000	22.000000	0.000000	185.000000	6.350000	968.000000	779.510000	28.000000	34.442774
75%	350.000000	142.950000	118.270000	192.000000	10.160000	1029.400000	824.000000	56.000000	46.136287
Max	540.000000	359.400000	200.100000	247.000000	32.200000	1145.000000	992.600000	365.000000	82.599225

Figure 23 Table showing Dataset Features Summarization

Statistical Distribution of Relevant Features

```
# Set up subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 15))

# Flatten axes for easier iteration
axes = axes.flatten()

# Plot histograms for each feature
for i, col in enumerate(df.columns):
    ax = axes[i]
    df[col].plot(kind='hist', ax=ax, bins=20, color='skyblue', edgecolor='black')
    ax.set_title(col)
    ax.grid(True)

# Adjust layout
plt.tight_layout()
plt.show()
```

Several noteworthy observations arise from the distribution patterns of the features which are similar to the ones concluded from the research:

- Cement showcases a distribution closely resembling a normal curve.
- Blast-furnace slag (abbreviated as 'slag') exhibits proper skewness and presents a distribution with three distinct peaks, reminiscent of Gaussian distributions.
- Fly ash (abbreviated as 'flyash') displays a right-skewed distribution with two prominent peaks, suggesting a bimodal distribution.
- Water's distribution shows three distinct peaks with a leftward inclination.
- Superplasticizer (abbreviated as 'sp') demonstrates a distribution with two peaks and appropriate skewness.
- Coarse aggregate (abbreviated as 'coarse') closely resembles a normal distribution with three noticeable peaks resembling Gaussians.
- Fine aggregate (abbreviated as 'fine') appears bimodal, with two distinct peaks indicating a non-normal distribution.
- The age feature exhibits multiple peaks and skewness, suggesting it may be suitable for the dataset.

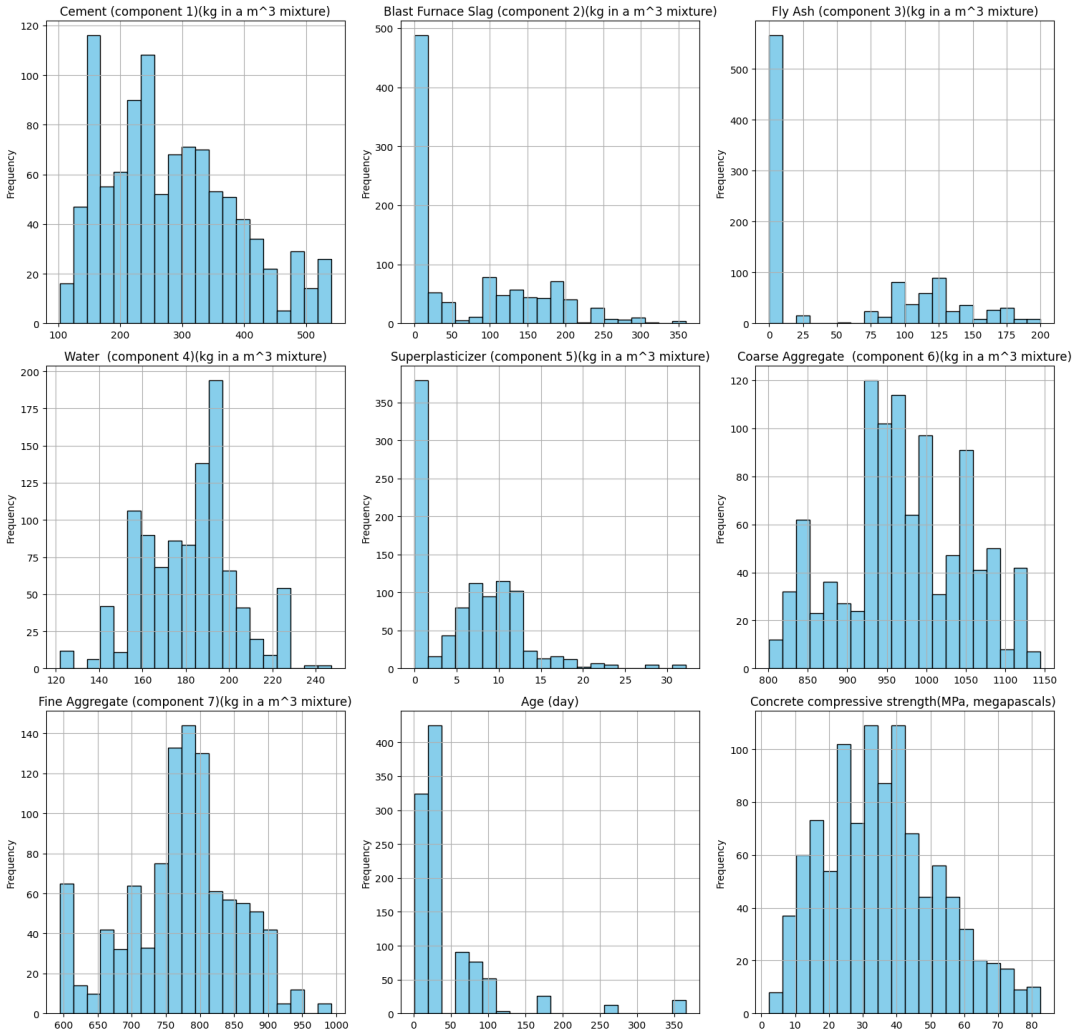

Figure 24 shows Statistical Distribution of Features

10.1.3 Concrete features correlation

Heat Map to Display Correlation Between Features

```
[7]  # Calculate the correlation matrix
     correlation_matrix = df.corr()

     # Create a heatmap
     plt.figure(figsize=(10, 8))
     sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
     plt.title('Correlation Heatmap')
     plt.show()
```

Significantly, a robust correlation is evident between cement and strength, underscoring cement's reliability as a predictor. Conversely, slag and fly ash exhibit weaker correlations with the target variable. Furthermore, noteworthy is the pronounced positive correlation between superplasticizer and fly ash, contrasting with the relatively weaker correlation between superplasticizer and compressive strength. Notably, there exists a considerable negative correlation between water and superplasticizers, as well as between water and strength.

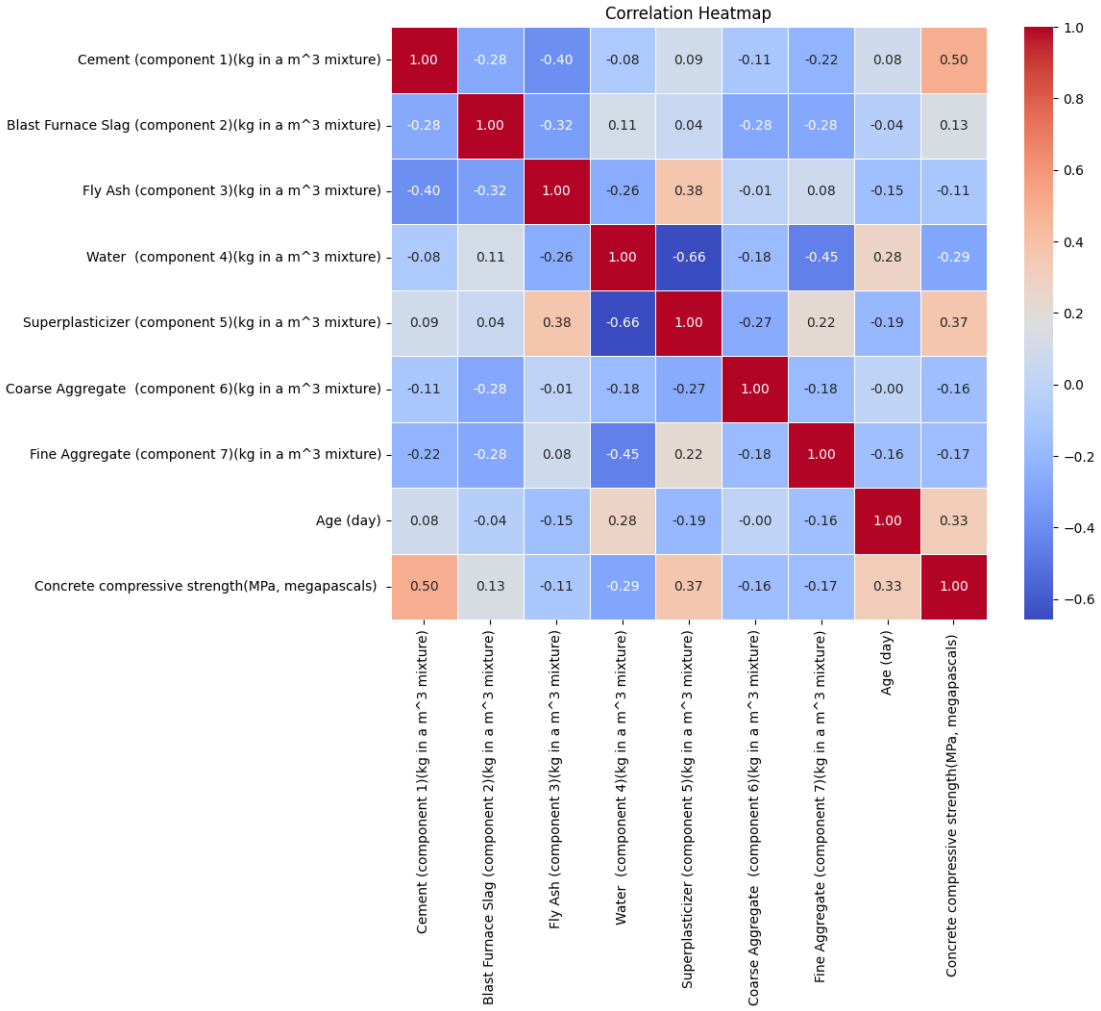

Figure 25 shows Features Heatmap

Pair Plot to Display Correlation Between Features

```
[8]  # Create a pair plot
     sns.pairplot(df)
     plt.show()
```

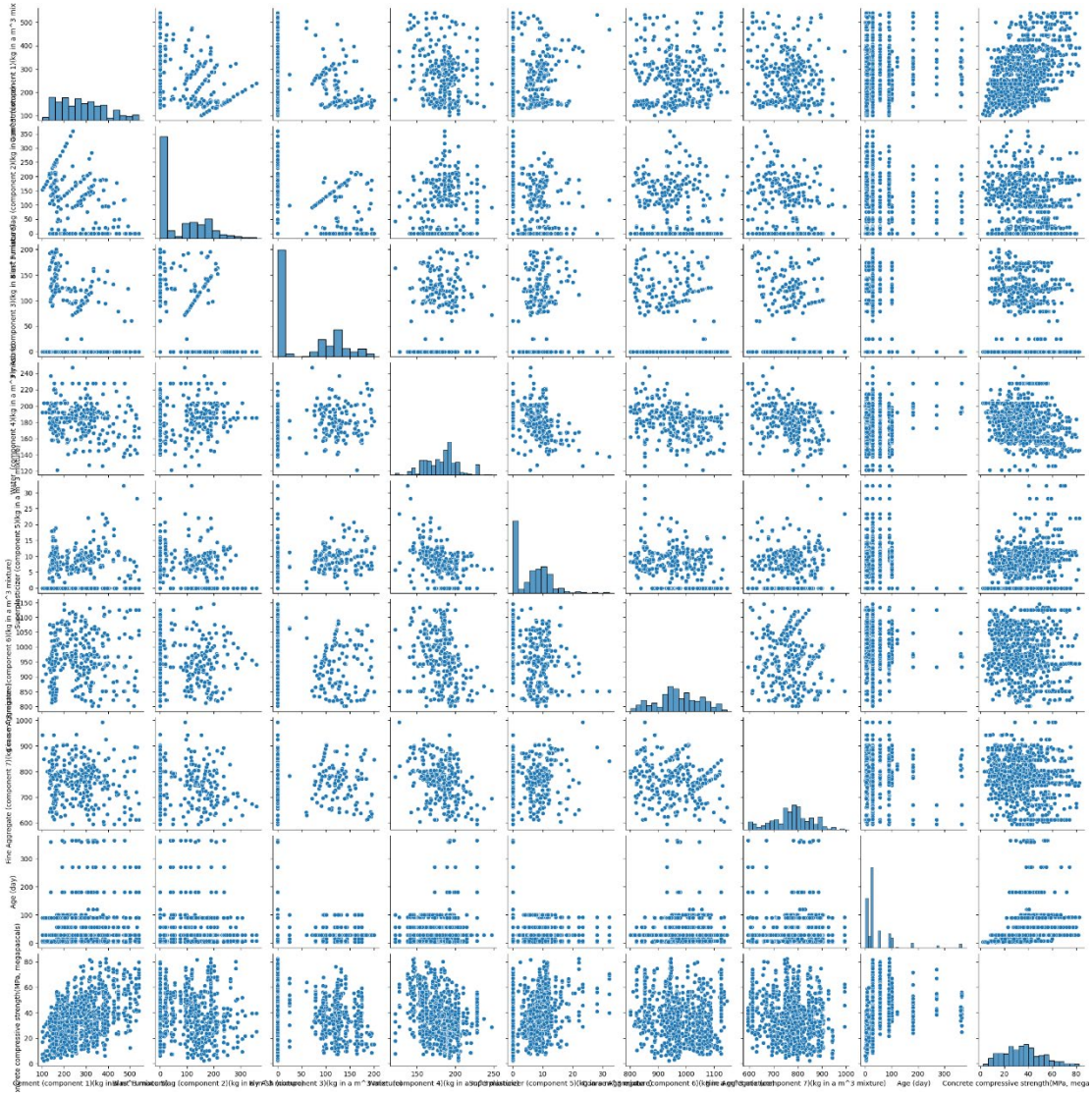

Figure 26 PairPlot Between Features

Figure 27 depicts a direct relationship between cement content and compressive strength, indicating that higher cement quantities lead to increased concrete strength. Furthermore, the study notes that as concrete ages, its strength tends to increase, requiring higher cement amounts to achieve greater strength at younger ages. Conversely, older cement necessitates more water, implying that reducing water content enhances concrete strength.

Scatter Plot for Visualizing the Cement and Compressive Strength Relationship

```
# Scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(df['Cement (component 1)(kg in a m^3 mixture)'], df['Concrete compressive strength(MPa, megapascals) '], alpha=0.5)
plt.title('Scatter Plot of Compressive Strength vs Cement')
plt.xlabel('Cement (kg/m^3)')
plt.ylabel('Compressive Strength (MPa)')
plt.grid(True)
plt.show()
```

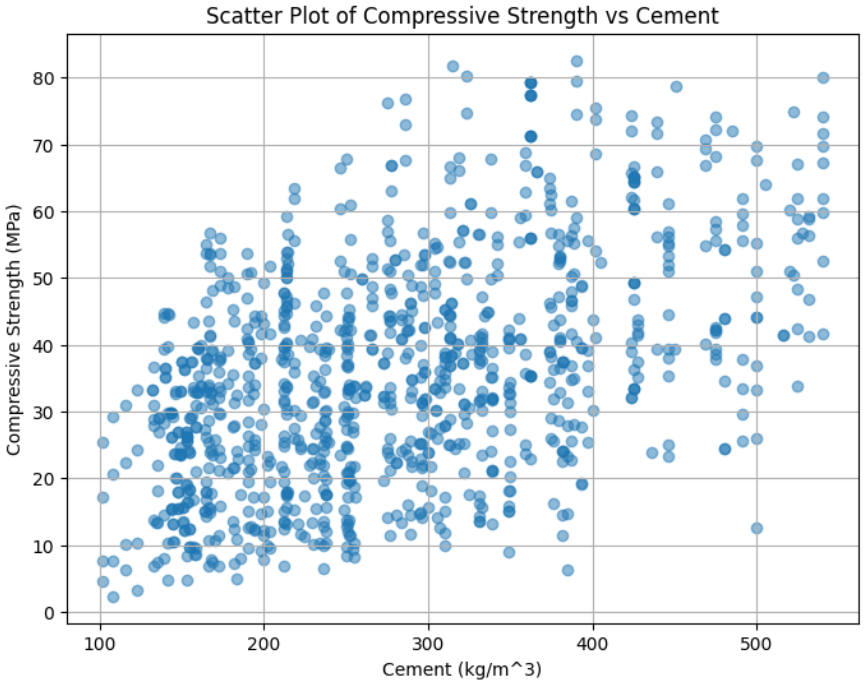

Figure 27 Cement vs Compressive Strength Scatter Plot

The scatter plot depicted in Figure 28 illustrates a negative correlation between compressive strength and fly-ash content. This relationship is evident from the concentration of darker dots in the area corresponding to lower compressive strength values.

Scatter Plot for Visualizing the Fine Aggregate and Compressive Strength Relationship

```
[10] # Scatter plot
     plt.figure(figsize=(8, 6))
     plt.scatter(df['Fine Aggregate (component 7)(kg in a m^3 mixture)'], df['Concrete compressive strength(MPa, megapascals) '], alpha=0.5)
     plt.title('Scatter Plot of Compressive Strength vs Fine Aggregate')
     plt.xlabel('Fine Aggregate (kg/m^3)')
     plt.ylabel('Compressive Strength (MPa)')
     plt.grid(True)
     plt.show()
```

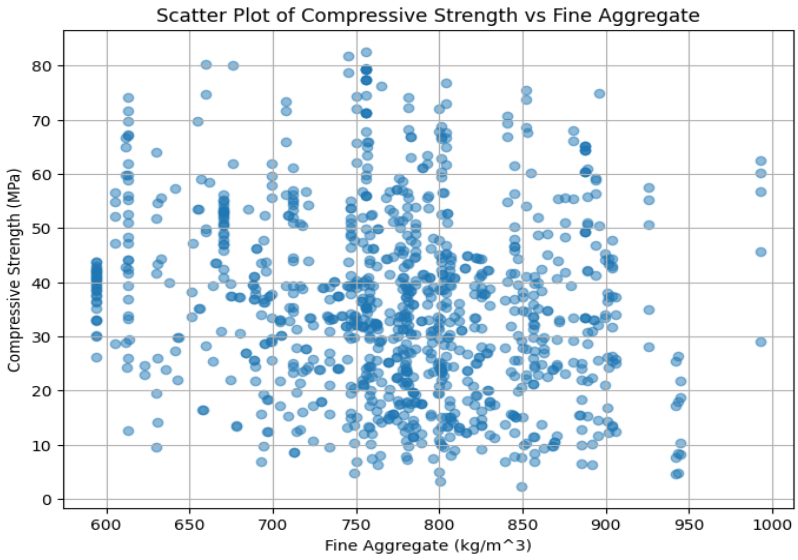

Figure 28 Fine Agg vs Compressive Strength Scatter Plot

10.1.4 Dataset splitting and Normalization

1. **Data Preparation:**

The dataset is split into two parts: features (X) and the target variable (y).

The features (X) contain all columns except the one representing the target variable (Concrete compressive strength (MPa, megapascals)).

The target variable (y) contains only the column representing the concrete compressive strength.

2. **Train-Test Split:**

The dataset is split into a training set (X_train, y_train) and a test set (X_test, y_test).

The split ratio is 80% for training and 20% for testing.

3. **Feature Scaling:**

The features are standardized using a StandardScaler.

Standardization involves transforming the features so that they have a mean of 0 and a standard deviation of 1.

The fit_transform method is used on the training set (X_train) to fit the scaler and transform the training features.

The transform method is then used on the test set (X_test) to transform the test features based on the scaling parameters learned from the training set.

Dataset Splitting

```
[11] # Split the dataset into features (X) and target variable (y)
     X = df.drop(columns=['Concrete compressive strength(MPa, megapascals) ']) # Features
     y = df['Concrete compressive strength(MPa, megapascals) '] # Target variable

     # Split the dataset into a training set and a test set (80% training, 20% test)
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Dataset Normalization using Standard Scaler (Zero Mean and STD = 1)

```
# Normalize the features using a standard scaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

10.1.5 Applying Linear Regression Model

Applying the Linear Regression Model

```
[13] # Initialize Linear Regression model
     lr_model = LinearRegression()

     # Fit the model on the training data
     lr_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred = lr_model.predict(X_train_scaled)
     y_test_pred = lr_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	9.796708
MSE	95.975484
MAE	7.745393
R-Squared	0.627542

10.1.6 Applying Lasso Regression Model

Applying the LASSO Regression Model

```
[15] # Initialize LASSO Regression model
     lasso_model = Lasso(alpha=1.2)  # You can adjust the alpha parameter as needed

     # Fit the model on the training data
     lasso_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred_lasso = lasso_model.predict(X_train_scaled)
     y_test_pred_lasso = lasso_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	10.682880
MSE	114.123929
MAE	8.785594
R-Squared	0.557112

10.1.7 Applying Ridge Regression Model

Applying the Ridge Regression Model

```
[17] # Initialize Ridge Regression model
     ridge_model = Ridge(alpha=170)  # You can adjust the alpha parameter as needed

     # Fit the model on the training data
     ridge_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred_ridge = ridge_model.predict(X_train_scaled)
     y_test_pred_ridge = ridge_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	10.291444
MSE	105.913819
MAE	8.440781
R-Squared	0.588973

10.1.8 Applying SVM Model

Given the Pearson correlation coefficients computed for the features in the dataset, if there are non-linear relationships or weak correlations present, opting for a non-linear kernel like RBF can help capture the underlying patterns effectively, hence justifying the choice of RBF kernel for the SVM regression model.

- **Cement:** There is a moderately positive correlation (0.497833) between cement content and concrete compressive strength. This indicates that as the amount of cement in the mixture increases, the compressive strength of the concrete tends to increase as well.
- **Blast Furnace Slag:** The correlation coefficient (0.134824) for blast furnace slag suggests a weak positive correlation with compressive strength. This implies that the presence of blast furnace slag in the mixture may have a slight positive effect on the concrete's compressive strength.
- **Fly Ash:** The correlation coefficient (-0.105753) for fly ash indicates a weak negative correlation with compressive strength. This suggests that higher levels of fly ash in the mixture may lead to a slight decrease in compressive strength.
- **Water:** There is a moderate negative correlation (-0.289613) between water content and compressive strength. This implies that higher water content in the mixture tends to result in lower compressive strength of the concrete.
- **Superplasticizer:** The correlation coefficient (0.366102) for superplasticizer indicates a moderate positive correlation with compressive strength. This suggests that the presence of superplasticizer in the mixture may contribute to an increase in compressive strength.
- **Coarse Aggregate and Fine Aggregate:** Both coarse aggregate and fine aggregate show weak negative correlations (-0.164928 and -0.167249, respectively) with compressive strength. This implies that higher levels of coarse and fine aggregates may lead to slight decreases in compressive strength.

- **Age:** There is a moderate positive correlation (0.328877) between the age of the concrete and its compressive strength. This suggests that as the concrete ages, its compressive strength tends to increase.

```
# Calculate Pearson correlation coefficient
correlation = df.corr()

# Extract correlation values with the target variable
correlation_with_target = correlation['Concrete compressive strength(MPa, megapascals) ']

# Display correlation values
print("Pearson correlation coefficient with the target variable:")
print(correlation_with_target)
```

```
Pearson correlation coefficient with the target variable:
Cement (component 1)(kg in a m^3 mixture)              0.497833
Blast Furnace Slag (component 2)(kg in a m^3 mixture)  0.134824
Fly Ash (component 3)(kg in a m^3 mixture)            -0.105753
Water  (component 4)(kg in a m^3 mixture)             -0.289613
Superplasticizer (component 5)(kg in a m^3 mixture)    0.366102
Coarse Aggregate  (component 6)(kg in a m^3 mixture)  -0.164928
Fine Aggregate (component 7)(kg in a m^3 mixture)     -0.167249
Age (day)                                              0.328877
Concrete compressive strength(MPa, megapascals)        1.000000
Name: Concrete compressive strength(MPa, megapascals) , dtype: float64
```

Applying the SVM Model

```
[20] # Initialize SVM Regression model
     svm_model = SVR(kernel='rbf')  # You can specify different kernels as needed

     # Fit the model on the training data
     svm_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred_svm = svm_model.predict(X_train_scaled)
     y_test_pred_svm = svm_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	9.432832
MSE	88.978327
MAE	7.515449
R-Squared	0.654696

10.1.8 Applying Decision Tree Model

Applying the Decision Tree Model

```
[22] # Initialize Decision Tree Regression model
     decision_tree_model = DecisionTreeRegressor()

     # Fit the model on the training data
     decision_tree_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred_dt = decision_tree_model.predict(X_train_scaled)
     y_test_pred_dt = decision_tree_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	6.808041
MSE	46.349417
MAE	4.366235
R-Squared	0.820129

10.1.9 Applying Random Forest Model

Applying the Random Forest Model

```
[24] # Initialize Random Forest Regression model
     random_forest_model = RandomForestRegressor()

     # Fit the model on the training data
     random_forest_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred_rf = random_forest_model.predict(X_train_scaled)
     y_test_pred_rf = random_forest_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	5.457730
MSE	29.786820
MAE	3.790815
R-Squared	0.884404

10.1.10 Applying XG Boost Model

Applying the XGBoost Model

```
[26] # Initialize XGBoost Regression model
     xgb_model = xgb.XGBRegressor()

     # Fit the model on the training data
     xgb_model.fit(X_train_scaled, y_train)

     # Make predictions on training and testing data
     y_train_pred_xgb = xgb_model.predict(X_train_scaled)
     y_test_pred_xgb = xgb_model.predict(X_test_scaled)
```

The concluded testing metrics were:

Metric	Value
RMSE	4.452198
MSE	19.822068
MAE	2.906828
R-Squared	0.923075

10.1.11 Applying ANN Model

Applying the ANN Model

```
[28] # Define the architecture of the neural network
     model = Sequential([
         Dense(128, activation='relu', input_shape=(X_train_scaled.shape[1],)),
         Dense(64, activation='relu'),
         Dense(1)  # Output layer with 1 neuron (for regression)
     ])

     # Compile the model
     model.compile(optimizer='adam', loss='mean_squared_error')

     # Fit the model on the training data
     history = model.fit(X_train_scaled, y_train, epochs=200 , batch_size=32, validation_split=0.2, verbose=0)

     # Make predictions on training and testing data
     y_train_pred_ann = model.predict(X_train_scaled).flatten()
     y_test_pred_ann = model.predict(X_test_scaled).flatten()

     26/26 [==============================] - 0s 2ms/step
     7/7 [==============================] - 0s 2ms/step
```

The concluded testing metrics were:

Metric	Value
RMSE	6.162814
MSE	37.980279
MAE	4.523561
R-Squared	0.852607

10.2 Implemented Results Comparison

As shown in Figure 29, XG Boost emerged as the top-performing model in terms of R-squared value, achieving an impressive score of 0.92, indicating its superior accuracy in predicting concrete strength. Following closely behind, Random Forests demonstrated strong performance with an R-squared value of 0.88. Decision Trees also exhibited good predictive capability, achieving an R-squared value of 0.82. The Artificial Neural Network (ANN) model performed reasonably well with an R-squared value of 0.85. In contrast, the traditional Linear Regression models showed lower accuracy, with the basic model achieving an R-squared value of 0.63. The LASSO Regression and Ridge Regression techniques also yielded similar R-squared values of 0.56 and 0.59, respectively. However, the Support Vector Machine (SVM) model outperformed the linear models, attaining an R-squared value of 0.65, indicating better predictive performance.

Model	RMSE	MSE	MAE	R-squared
Linear Regression	9.8	95.98	7.75	0.63
LASSO Regression	10.68	114.12	8.79	0.56
Ridge Regression	10.29	105.91	8.44	0.59
SVM	9.43	88.98	7.52	0.65
Decision Tree	6.81	46.35	4.37	0.82
Random Forest	5.46	29.79	3.79	0.88
XGBoost	4.45	19.82	2.91	0.92
ANN	6.16	37.98	4.52	0.85

Figure 29 shows Concluded Metrics of Different Models

The below table shows the mentioned metrics in the research:

Note that the R-Squared Error of ANN is unintentionally written incorrectly in the research (it is supposed to be 0.74 instead of 74.63)

Machine learning approaches	Method	RMSE	MSE	MAE	R^2
Statistical	Linear regression	10.28	105.76	8.23	0.57
	LASSO regression	10.68	114.11	8.65	0.54
	Ridge regression	10.29	105.84	8.24	0.57
	SVM	9.13	83.39	7.44	0.66
Tree-based	Decision trees	6.65	44.24	4.47	0.82
	Random forests	5.21	27.17	3.53	0.89
	XGBoost	**4.37**	22.33	3.04	**0.91**
Artificial neural network	ANN	6.01	36.22	4.53	74.63

Figure 30 shows Research Concluded Metrics

By comparing both metrics table of our work and the research results, all values are approximately the same except for the linear regression model metrics which varied from the expected research values. Small differences in the implementation of the linear regression model, such as the convergence criteria or optimization algorithm, can lead to variations in the model's output. In some cases, even with the same data and model, slight variations in the dataset's statistical properties or random noise can lead to differences in model performance metrics.

Also, we compared the RMSE and R-Squared values for our models and the research models and the following figures show our work and the research work metrics respectively after scaling the R-Squared values.

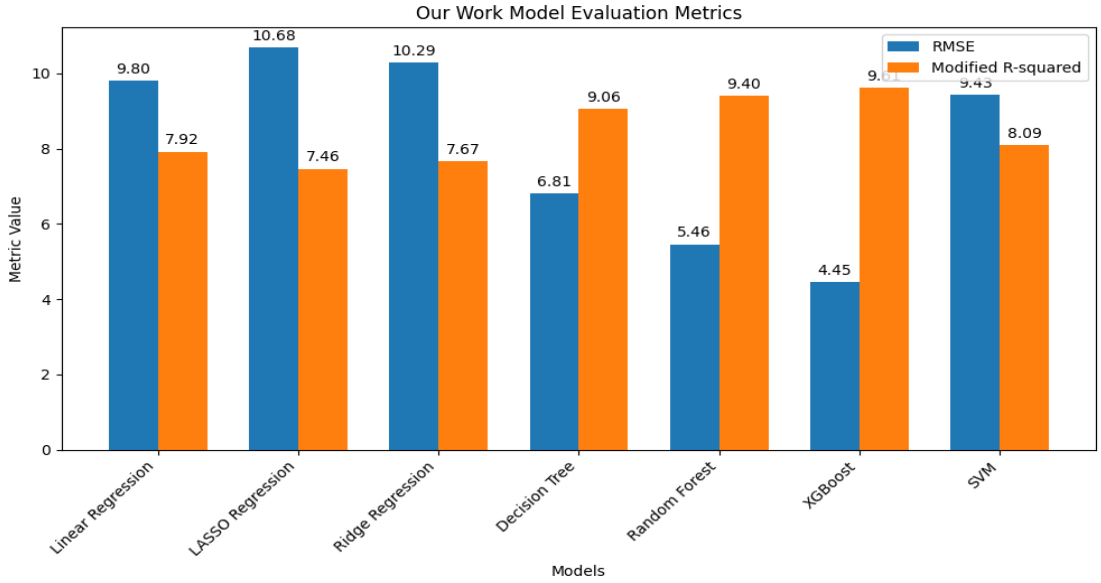

Figure 31 Our Work Metrics Comparison

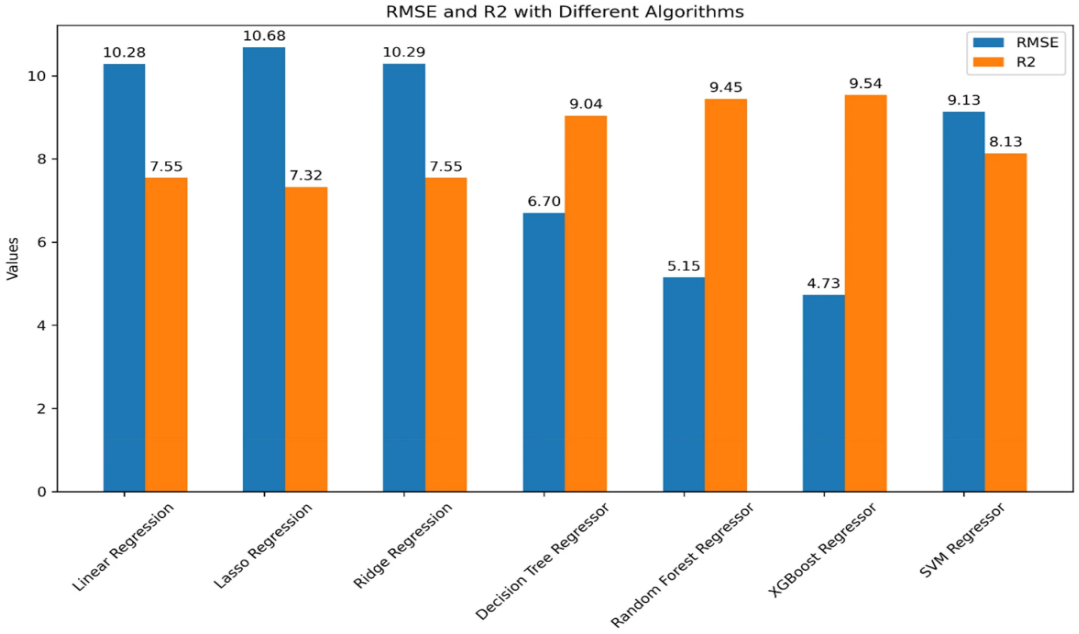

Figure 32 Research Metrics Comparison

The previous figures suggest that employing the XG Boost model may offer effective performance and reasonably accurate predictions when estimating the compressive strength of concrete.

Considering the criterion of accuracy, it can be inferred that XG Boost stands out as the top-performing model. Figure 33 illustrates scatter plots showing the experimental (actual) and predicted compressive strengths of concrete, respectively.

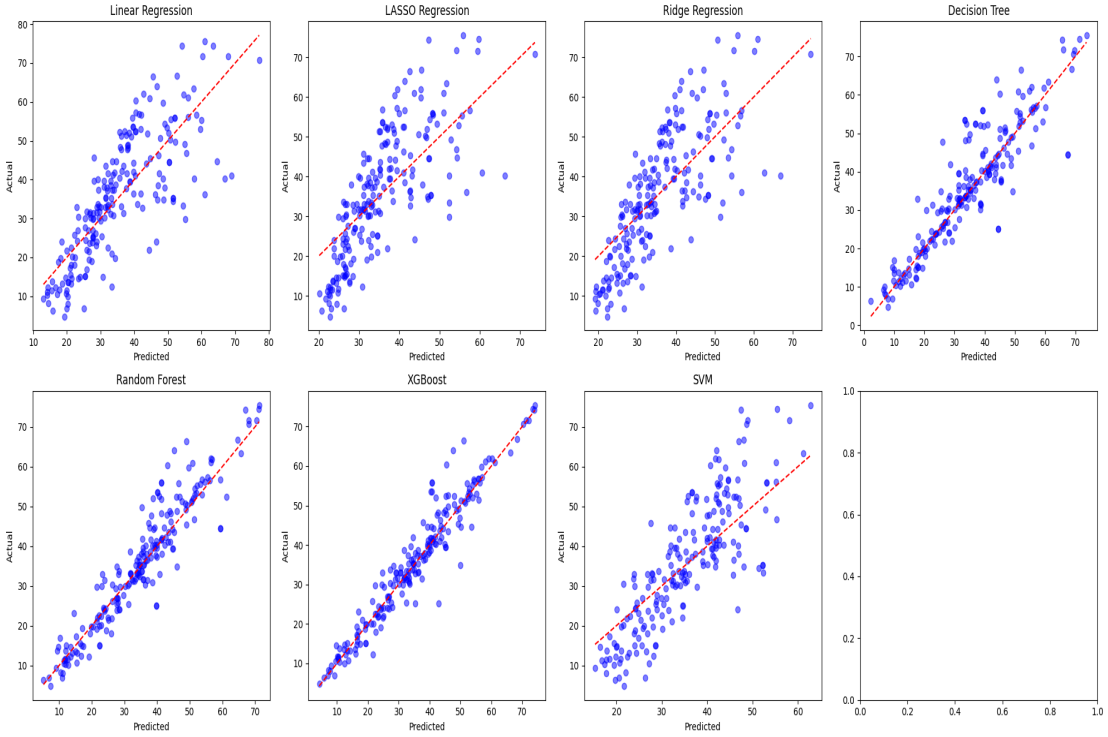

Figure 33 Scatter Plots between True and Predicted Values

11.0 REFERENCES

[1] "Concrete Batching Plants" https://www.mekaglobal.com/en/products/concrete-batching-plants

[2] "How Does the Concrete Batching Plant Work ?" https://controlmakers.ir/en/concerete-batching-plant/how-does-the-concrete-batching-plant-work/

[3] "Concrete Slump Test Regression", https://www.kaggle.com/code/emineildesegri/concrete-slump-test-regression-for-one-target

[4] "Revisiting Machine Learning Datasets – Concrete Slump Test," https://www.simonwenkel.com/2018/08/09/revisiting_ml_datasets_concrete_slump_test.html

[5] "Concrete Compressive Strength," https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength

[6]"Sample Records for Concrete Compressive Strength Prediction ," https://www.nature.com/articles/s41598-023-47169-7/tables/3

[7] "Concrete Data Exploration Process (Features Summarization)," https://www.nature.com/articles/s41598-023-47169-7/tables/4

[8] "Features Exploration," https://www.nature.com/articles/s41598-023-47169-7/figures/6

[9] "Features Correlation Pair Plot ," https://www.nature.com/articles/s41598-023-47169-7/figures/7

[10]"Scatter Plot for Visualizing Cement vs Compressive Strength," https://www.nature.com/articles/s41598-023-47169-7/figures/8

[11]"Scatter Plot for Visualizing Fine Aggregates vs Compressive Strength," https://www.nature.com/articles/s41598-023-47169-7/figures/9

[12] "Machine Learning Pipeline for Concrete Strength Prediction," https://www.nature.com/articles/s41598-023-47169-7/figures/1