LEEDS BECKETT UNIVERSITY

School of Computing, Creative Technology and Engineering

Student ID	77356785
Student Name	Pooja Pantha
Module Name & CRN	Applied Machine Learning 18909
Level	5
Assessment Name & Part No.	Coursework - I
Project Title	Case Study on Concrete Strength Analysis
Data of Submission	31/03/2024
Course	Computing
Academic Year	2024

PART I

Case Study on Concrete Strength Analysis
Pooja Pantha, Computing 2024

Table of Contents

Table of Figures

Introduction

All over the world, concrete as a product plays an important role in modern construction. Its formula contains a combination of ingredients that all make it plumping and long-lasting. Among these products, cement acts as a binder that promotes the adhesion of aggregates -- which are coarse and fine, water, and auxiliary materials. The quality of concrete, measured by compressive strength, is important for the safety and durability of the building. As the construction industry continues to strive for efficiency and sustainability, accurate prediction of concrete performance has become an important goal for cement manufacturers and engineers.

Machine learning falls within a branch of artificial intelligence focused on the exploration of computer algorithms capable of self-learning and evolution through exposure to experience and past data (Jordan & Mitchell, 2015). The need for precise concrete compressive strength prediction has grown in modern construction techniques. This is motivated by multiple elements (Rusch, Sell and Rackwitz, 1969). Firstly, the amount of detail in building projects is growing, which calls for careful planning and resource efficiency. Second, to minimize the impact on the environment, new mix designs and alternative materials must be investigated considering the increased emphasis on sustainability. Thirdly, the development of advanced predictive algorithms for concrete strength is made possible by technological breakthroughs, especially in the fields of data science and machine learning. Predicting the strength of concrete is important for a wide range of practical applications (Badole, 2021). Precise strength estimates direct material selection and structural design decisions in large-scale construction projects like bridges, dams, and skyscrapers, ensuring compliance with safety regulations and standards. Predictive models also help in proactive repair strategy implementation and possible structural deficiency identification during infrastructure rehabilitation and maintenance, reducing downtime and risk (Mehmood, 2024).

Applications in the real world include the construction of roads, bridges, dams, and high-rise buildings, where precise forecasts of concrete strength are crucial for both public safety and structural integrity. Let's consider the world's tallest building - Burj Khalifa as an example. Strong concrete was needed for the Burj Khalifa to support its enormous weight and wind pressures. For the project, various mixes of high-performance concrete (HPC) were created (Aldred, 2010). Data analysis most likely contributed a vital factor in maximizing the mix design to maintain functionality while obtaining the required strength, evaluating test sample data to guarantee constant strength during the entire construction procedure, etc.

Literature Review

Growing interest has been seen in using machine learning techniques in recent years to predict material properties, thereby reducing the need for time-consuming and costly experimental processes. Machine learning

models estimate concrete strength parameters with high accuracy (Targino et al., 2021). This literature review on “Computation of High-Performance Concrete Compressive Strength Using Standalone and Ensembled Machine Learning Techniques” explores the application of standalone and ensemble ML techniques for forecasting the compressive strength of high-performance concrete (HPC) (Xu et al., 2021). Specifically, it examines the efficacy of support vector regression (SVR), AdaBoost, and random forest (RF) algorithms in this context. In the literature review exploring the computation of high-performance concrete (HPC) compressive strength through machine learning (ML) techniques, it is evident that leveraging standalone and ensemble ML methods presents a promising avenue for accurately forecasting material properties. Specifically, the application of support vector regression (SVR), AdaBoost, and random forest (RF) algorithms showcases improved performance in predicting concrete strength compared to traditional methods. Notably, the RF model emerges as the most accurate, highlighting its potential for enhancing predictive accuracy in concrete strength analysis. Moreover, the emphasis on rigorous validation techniques such as statistical analysis, cross-validation, and sensitivity analysis underscores the importance of robust methodologies in evaluating ML models' performance and identifying influential input parameters. Relating this to the project titled "Concrete Strength Data Analysis: A Comparative Exploration Using Two Datasets for Training and Testing through Exploratory Data Analysis," this literature review provides a foundational understanding of the efficacy of ML techniques in predicting concrete properties. By adopting exploratory data analysis (EDA) techniques on two distinct datasets for training and testing, the project aims to further explore and validate the findings of the literature review, ultimately contributing to advancing the understanding and application of ML-driven concrete strength analysis (Xu et al., 2021).

Exploratory Data Analysis

Concrete dataset has 9 columns in both train and test dataset, 722 and 308 rows respectively.

	Cement	Blast.Furnace.Slag	Fly.Ash	Water	Superplasticizer	Coarse.Aggregate	Fine.Aggregate	Age	Strength
1	540.0	0.0	0	162.0	2.5	1040.0	676.0	28	79.99
2	540.0	0.0	0	162.0	2.5	1055.0	676.0	28	61.89
3	332.5	142.5	0	228.0	NA	932.0	594.0	270	40.27
4	198.6	132.4	0	192.0	0.0	978.4	825.5	360	44.30
5	266.0	114.0	0	228.0	0.0	932.0	670.0	90	47.03
6	266.0	114.0	0	228.0	0.0	932.0	670.0	28	45.85
7	475.0	0.0	0	228.0	0.0	932.0	594.0	28	39.29
8	198.6	132.4	0	192.0	0.0	978.4	825.5	90	38.07
9	198.6	132.4	0	192.0	0.0	978.4	825.5	28	28.02
10	427.5	47.5	0	228.0	NA	932.0	594.0	270	43.01
11	190.0	190.0	0	228.0	0.0	932.0	670.0	90	42.33
12	304.0	76.0	NA	228.0	0.0	932.0	670.0	28	47.81
13	380.0	NA	0	228.0	0.0	932.0	670.0	90	52.91

Showing 1 to 13 of 722 entries, 9 total columns

	Cement	Blast.Furnace.Slag	Fly.Ash	Water	Superplasticizer	Coarse.Aggregate	Fine.Aggregate	Age	Strength
1	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
2	380.0	95.0	NA	228.0	0.0	932.0	594.0	365	43.70
3	380.0	95.0	0.0	228.0	0.0	932.0	594.0	28	36.45
4	427.5	47.5	0.0	228.0	0.0	932.0	594.0	180	41.84
5	342.0	38.0	0.0	228.0	0.0	NA	670.0	180	52.12
6	266.0	114.0	0.0	228.0	0.0	932.0	NA	365	52.91
7	237.5	237.5	0.0	NA	0.0	932.0	594.0	365	39.00
8	427.5	47.5	0.0	228.0	0.0	932.0	594.0	7	35.08
9	349.0	0.0	0.0	192.0	0.0	1047.0	806.9	3	15.05
10	237.5	237.5	0.0	228.0	0.0	932.0	594.0	90	33.12
11	139.6	209.4	0.0	192.0	0.0	1047.0	806.9	7	14.59
12	266.0	114.0	0.0	228.0	0.0	932.0	670.0	270	51.73
13	190.0	190.0	0.0	228.0	0.0	932.0	NA	270	50.66

Showing 1 to 13 of 308 entries, 9 total columns

Fig 1: Data and Attributes in Train and Test dataset

The dataset contains float data type. The variables in a dataset are:

- **Cement:** The quantity of cement utilized in the concrete mixture is indicated by this variable. Concrete's strength and durability are mostly due to cement, which also serves as the main binding agent. It usually makes up the greatest percentage, weight-wise, of the concrete mix.
- **Blast Furnace Slag:** This is an iron production byproduct which has some potential applications as a partial cement substitute. By lowering the quantity of cement used and possibly enhancing some characteristics of the concrete, such as workability or sulfate resistance, it can have some positive environmental effects.

- **Fly Ash:** This is a finely grained coal combustion byproduct. It can be utilized as a limestone in concrete mixtures, accelerating the hydration process and improving the workability, strength, and impermeability of the concrete.
- **Water:** The chemical reaction that forms the binding paste between cement and other ingredients requires water. The strength and stability of concrete are strongly influenced by the water-to-cement ratio.
- **Superplasticizer:** Superplasticizers are chemical additions that are added to concrete mixtures to increase workability and decrease water content without weakening or weakening the concrete's strength. They improve concrete's flowability, which simplifies pouring and compressing.
- **Coarse Aggregate:** Larger crushed stone, gravel, or recycled concrete particles are utilized as coarse aggregates in concrete compositions. By spreading loads and minimizing compression, they give the concrete stability and bulk, enhancing its strength and longevity.
- **Fine Aggregate:** Smaller crushed stone or natural sand particles are used as fine aggregates, sometimes referred to as sand, in concrete compositions. They increase the concrete's strength and durability by filling in the spaces between the coarse particles and making the mixture easier to mix and useable.
- **Age:** Age is the amount of time that has lapsed since the concrete mixture was mixed and poured. Given that concrete gains strength through the processes of hydration and curing over time, it plays a significant role in determining the durability and development of strength in concrete.
- **Strength:** It is the dataset's main result variable. The capacity of concrete to tolerate compressive pressures is known as its compressive strength. It is a crucial factor that establishes how well concrete components operate structurally and how much weight they can support.

Summary Statistics

This summary provides an overview of the dataset's numeric variables, including their minimum, maximum, mean, quartiles, and missing values. It also highlights the presence of missing values in several variables, which will be addressed in third component.

In order to perform an EDA analysis, the training and testing datasets were compiled as follows:

```
> train<-read.csv(file="C:/Users/panth/Downloads/concrete_strength_train.csv", header = T)
> test <- read.csv(file="C:/Users/panth/Downloads/concrete_strength_test.csv", header = T)
> train$isTrain<-"yes"
> test$isTrain<-"no"
> combined <- rbind(train, test)
>
```

The amounts utilized in the design of the concrete mix are represented by all attributes other than "isTrain". Every attribute has NAs, or missing values. Across all attributes, there are between 30 to 70 missing data. A categorical variable called "isTrain" indicates whether a given data piece is part of the testing or training set.

```
> str(train)
'data.frame':   722 obs. of  10 variables:
 $ Cement            : num  540 540 332 199 266 ...
 $ Blast.Furnace.Slag: num  0 0 142 132 114 ...
 $ Fly.Ash           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Water             : num  162 162 228 192 228 228 228 192 192 228 ...
 $ Superplasticizer  : num  2.5 2.5 NA 0 0 0 0 0 0 NA ...
 $ Coarse.Aggregate  : num  1040 1055 932 978 932 ...
 $ Fine.Aggregate    : num  676 676 594 826 670 ...
 $ Age               : int  28 28 270 360 90 28 28 90 28 270 ...
 $ Strength          : num  80 61.9 40.3 44.3 47 ...
 $ isTrain           : chr  "yes" "yes" "yes" "yes" ...
>
```

10 variables and 722 observations represent the training dataset. The variable "Superplasticizer" has several missing values.

```
> str(test)
'data.frame':   308 obs. of  10 variables:
 $ Cement            : num  332 380 380 428 342 ...
 $ Blast.Furnace.Slag: num  142.5 95 95 47.5 38 ...
 $ Fly.Ash           : num  0 NA 0 0 0 0 0 0 0 0 ...
 $ Water             : num  228 228 228 228 228 228 NA 228 192 228 ...
 $ Superplasticizer  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Coarse.Aggregate  : num  932 932 932 932 NA ...
 $ Fine.Aggregate    : num  594 594 594 594 670 ...
 $ Age               : int  365 365 28 180 180 365 365 7 3 90 ...
 $ Strength          : num  41 43.7 36.5 41.8 52.1 ...
 $ isTrain           : chr  "no" "no" "no" "no" ...
>
```

There are 10 variables and 308 observations in the testing dataset. There are missing values in the variables Water, Coarse.Aggregate, and Fly.Ash.

```
> str(combined)
'data.frame':   1030 obs. of  10 variables:
 $ Cement            : num  540 540 332 199 266 ...
 $ Blast.Furnace.Slag: num  0 0 142 132 114 ...
 $ Fly.Ash           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Water             : num  162 162 228 192 228 228 228 192 192 228 ...
 $ Superplasticizer  : num  2.5 2.5 NA 0 0 0 0 0 0 NA ...
 $ Coarse.Aggregate  : num  1040 1055 932 978 932 ...
 $ Fine.Aggregate    : num  676 676 594 826 670 ...
 $ Age               : int  28 28 270 360 90 28 28 90 28 270 ...
 $ Strength          : num  80 61.9 40.3 44.3 47 ...
 $ isTrain           : chr  "yes" "yes" "yes" "yes" ...
>
```

10 variables and a total of 1030 observations (722 from train and 308 from test) make up the combined dataset. Problems with data types and missing values that were found in the separate datasets are still present in the combined dataset.

The primary concern of the company that supplies concrete to construction sites is the estimation of the concrete's strength. Strength is the variable that describes this attribute in the gathered dataset. Given that it is a numerical variable, solving the regression issue will be necessary in order to develop a predictive model for it. Let's investigate the Strength variable first:

```
> library(ggplot2)
> ggplot(combined, aes(x = Strength)) +
+   geom_histogram(binwidth = 5, fill = "skyblue", color = "black") +
+   labs(title = "Distribution of Concrete Strength",
+        x = "Strength",
+        y = "Frequency")
>
```

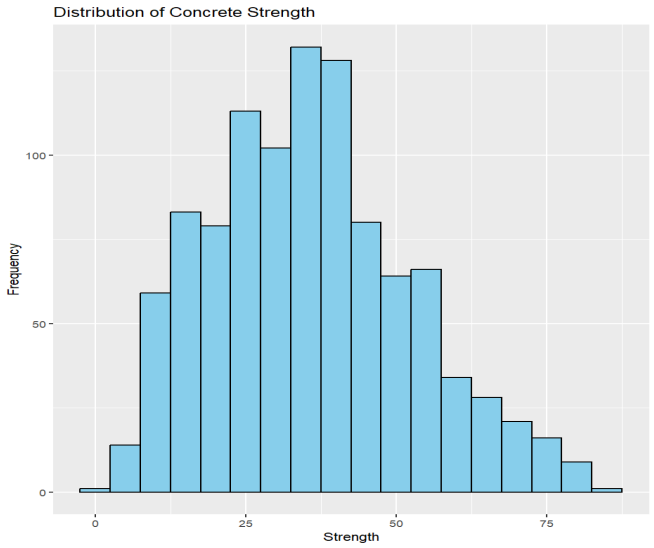

Fig 2: Distribution of Concrete Strength

A positive skew is seen when examining the concrete strength distribution visually with a histogram. According to this initial analysis, more concrete samples have lower compressive strength than samples with higher strength values.

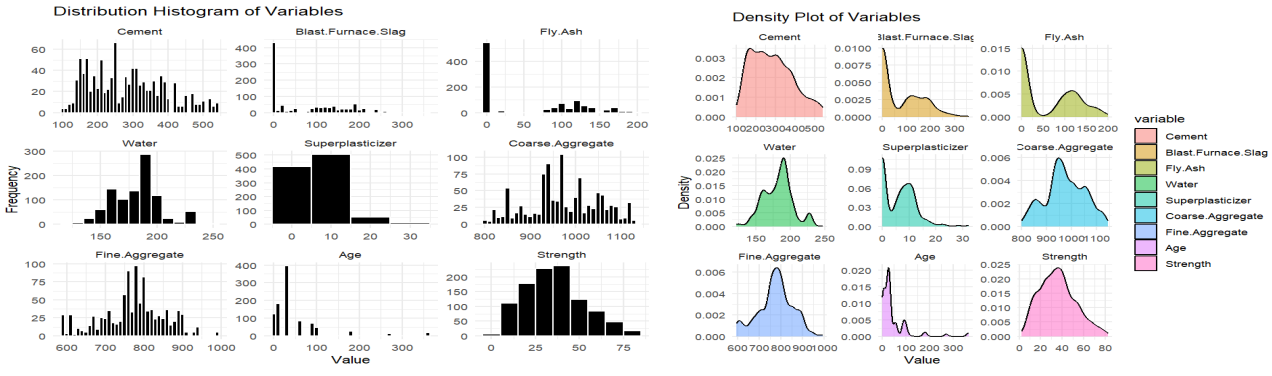

Fig 3: Distribution and Density plot of each variables

Data Pre-processing

Missing Values

To determine which variables had missing values, author first looked at the column missingness. This required figuring out how much of each variable's values were missing. The percentage of missing values in the dataset was then computed by calculating the overall missingness.

```
> col_missingness <- colMeans(is.na(combined))
> col_missingness
           Cement Blast.Furnace.Slag             Fly.Ash               Water   Superplasticizer
       0.04757282         0.06796117         0.03106796         0.05922330         0.04466019
  Coarse.Aggregate     Fine.Aggregate                Age           Strength            isTrain
       0.02912621         0.04368932         0.06407767         0.00000000         0.00000000
> overall_missingness <- mean(col_missingness)
> overall_missingness
[1] 0.03873786
> |
```

As it can be seen the average of missing values is 3.87%. Now, author uses mice() to impute values.

```
> library(mice)
> pattern_matrix <- md.pattern(combined)
>
```

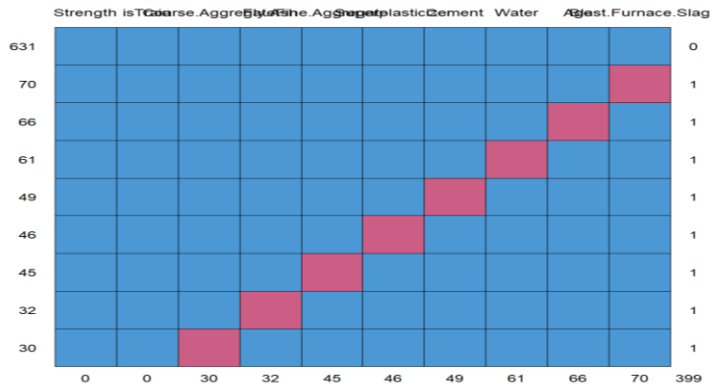

Fig 4: Graph of strength indexes

```
> aggr(combined)
> library(mice)
> pattern_matrix <- md.pattern(combined)
> library(VIM)
> aggr(combined)
>
```

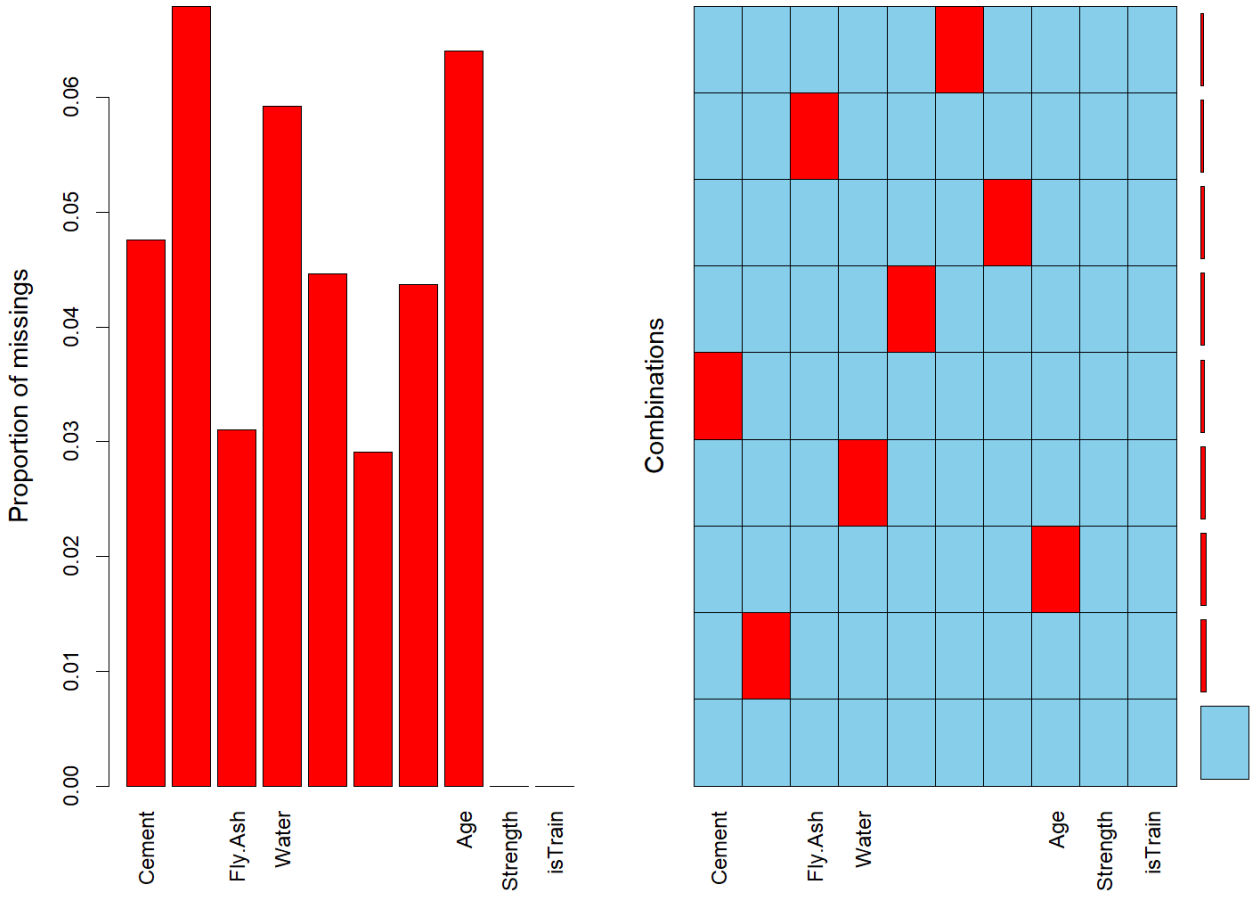

Fig 5: Graphical representation of missing values

```
> aggr_plot <- aggr(pattern_matrix, col = c('maroon', 'red'), numbers = TRUE, sortVars = TRUE, labels
= names(combined), cex.axis = 0.7, gap = 3)
```

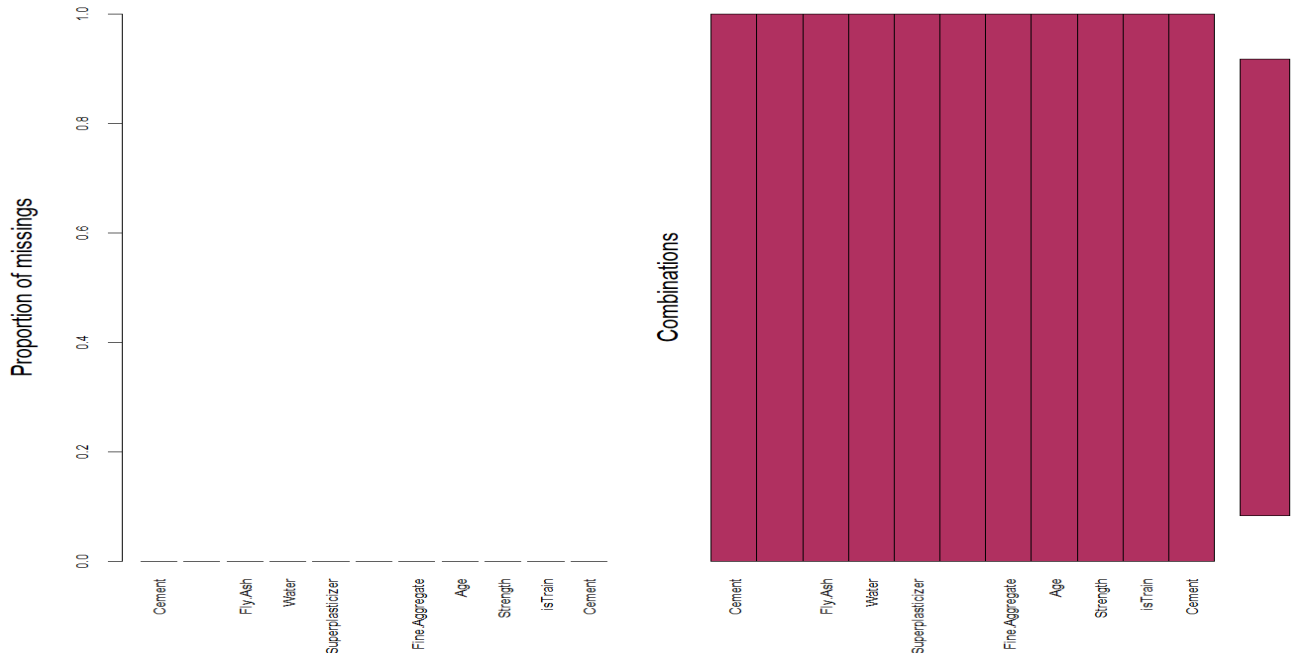

Fig 6: Missing values after imputation

The decision to impute was made with the goals of reducing bias, enhancing statistical power, and maintaining data integrity. The selection of mice() for multiple imputation was made in order to produce reasonable approximations for absent values and enable a thorough examination.

Outliers

The presence of outliers can be seen in the following figure which is represented in the form of box plot.

```
> boxplot(combined$Cement, combined$Blast.Furnace.Slag, combined$Fly.Ash, combined$Water, combined$Superplasticizer, combined$Coarse.Aggregate, combined$Fine.Aggregate, combined$Age, combined$Strength)
>
```

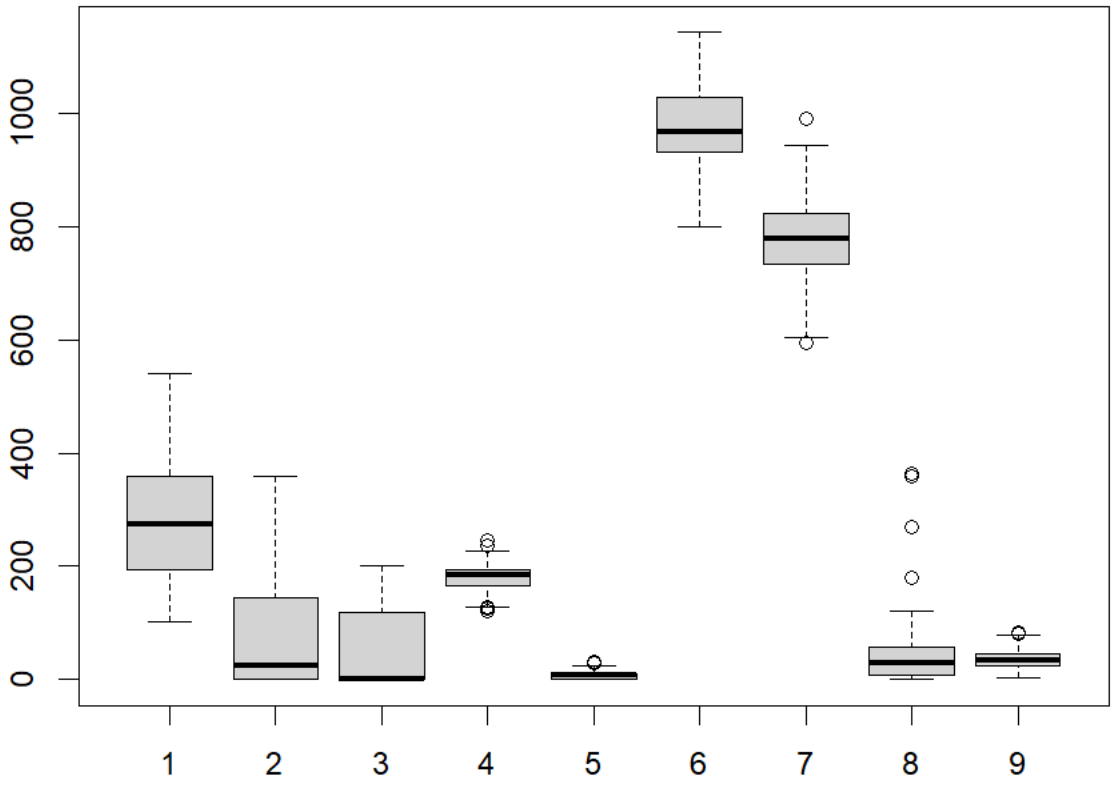

Fig 7: Representation of outliers in a boxplot

As it can be seen in an above figure, there are outliers present. Individual box plots are made for the ones where outlier is present.

```
> boxplot(combined$Water, main = "Water (Before)", col = "skyblue", ylab = "Water")
> boxplot(combined$Superplasticizer, main = "Superplasticizer (Before)", col = "skyblue", ylab = "Superplasticizer")
> boxplot(combined$Fine.Aggregate, main = "Fine Aggregate (Before)", col = "skyblue", ylab = "Fine Aggregate")
> boxplot(combined$Age, main = "Age (Before)", col = "skyblue", ylab = "Age")
> boxplot(combined$Strength, main = "Strength (Before)", col = "skyblue", ylab = "Strength")
>
```

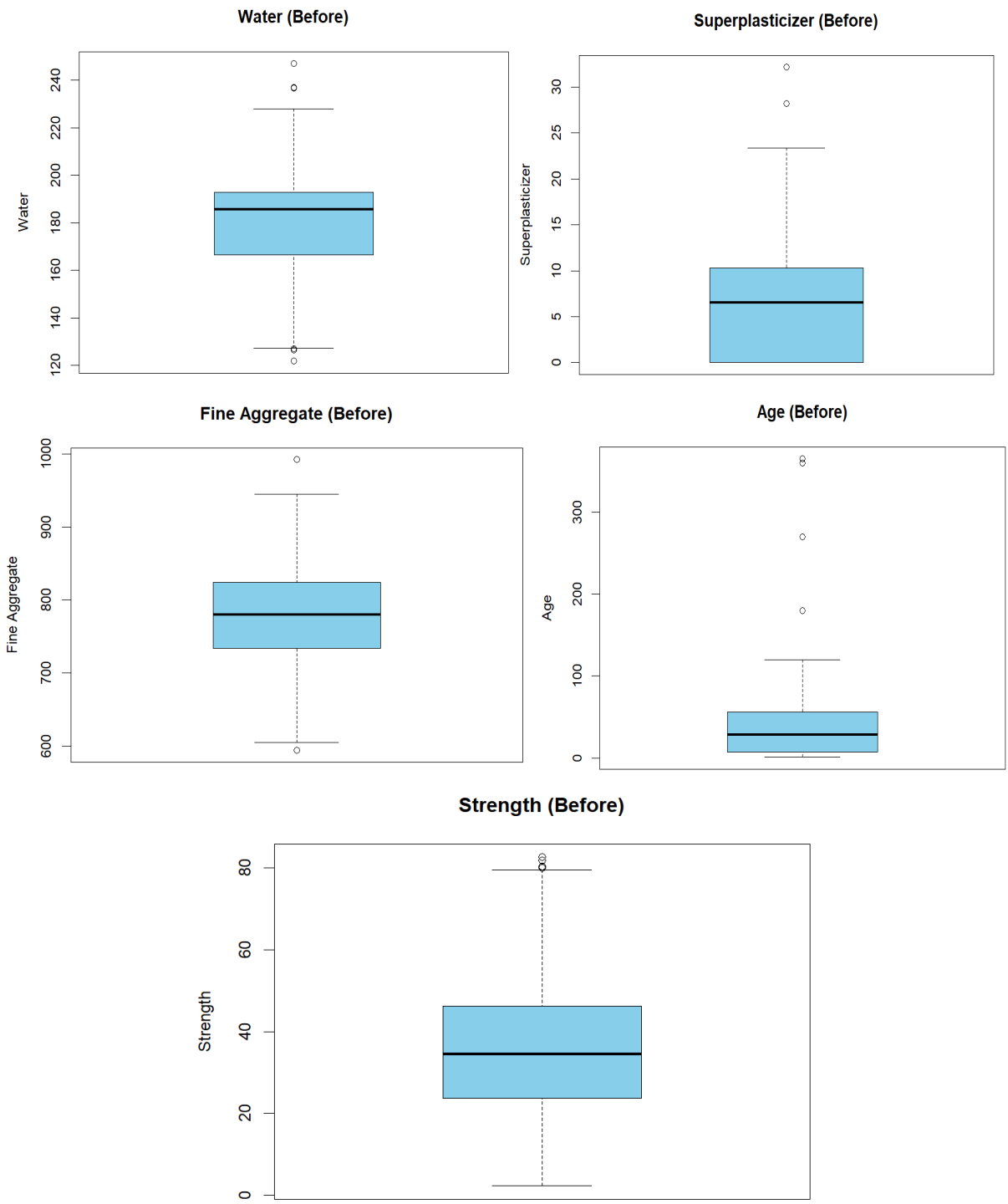

Fig 8: Outliers found in variables.

To avoid having an excessive impact on statistical analysis, in this report the decision has been made to eliminate outliers. Results might be skewed, and model performance impacted by outliers. Eliminating them guarantees stability in the data.

For instance, taking Strength for a demonstration:

```
> z_scores_strength <- scale(cleaned_data$Strength)
> z_scores_strength <- as.vector(z_scores_strength)
> outlier_indices <- which(abs(z_scores_strength) > 2)
> data_without_outliers <- cleaned_data[-outlier_indices, ]
> summary(cleaned_data$Strength)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.33   22.39   33.12   34.16   43.87   79.99
> summary(data_without_outliers$Strength)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.33   21.77   32.80   32.56   42.08   66.70
> ggplot(data_without_outliers, aes(y = Strength)) +
+   geom_boxplot() +
+   labs(title = "Boxplot of Strength", y = "Strength")
>
```

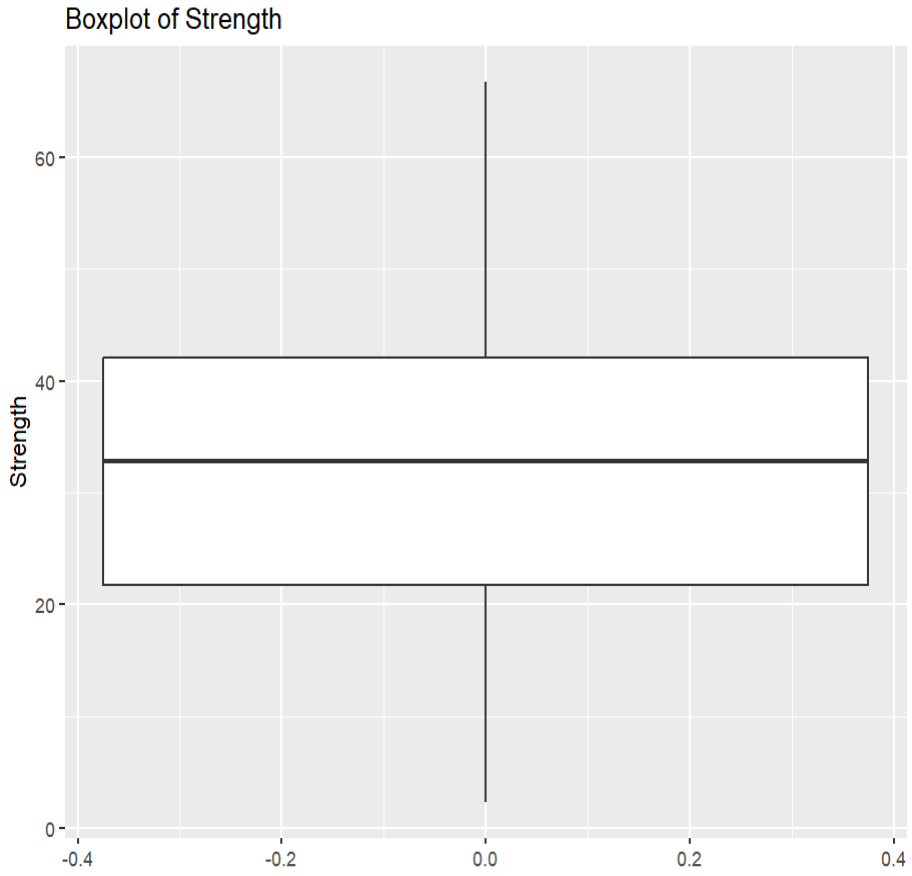

Fig 9: Boxplot of Strength variable after removing outliers

Multicollinearity

Regression models that have strongly linked predictor variables are said to be multicollinear, which makes the model harder to interpret and results in unstable coefficient projections.

```
> # Check for NA values before cleaning
> na_before <- sapply(combined, function(x) sum(is.na(x)))
> # Print NA counts before and after cleaning
> print(na_before)
            Cement Blast.Furnace.Slag            Fly.Ash              Water   Superplasticizer
                49                  70                 32                 61                 46
  Coarse.Aggregate     Fine.Aggregate                Age           Strength            isTrain
                30                  45                 66                  0                  0
> # Remove NA values from the dataset
> cleaned_data <- na.omit(combined)
> # Check for NA values after cleaning
> na_after <- sapply(cleaned_data, function(x) sum(is.na(x)))
> print(na_after)
            Cement Blast.Furnace.Slag            Fly.Ash              Water   Superplasticizer
                 0                   0                  0                  0                  0
  Coarse.Aggregate     Fine.Aggregate                Age           Strength            isTrain
                 0                   0                  0                  0                  0
> |
```

```
> correlations <- cor(cleaned_data[, c("Cement", "Blast.Furnace.Slag", "Fly.Ash", "Water", "Superplasticize
r", "Coarse.Aggregate", "Fine.Aggregate", "Age", "Strength")])
> print(correlations)
                     Cement Blast.Furnace.Slag     Fly.Ash       Water Superplasticizer
Cement             1.00000000        -0.32335981 -0.33334088 -0.07785009       0.13227816
Blast.Furnace.Slag -0.32335981         1.00000000 -0.32186938  0.09743388       0.03220498
Fly.Ash            -0.33334088        -0.32186938  1.00000000 -0.28208433       0.38935138
Water              -0.07785009         0.09743388 -0.28208433  1.00000000      -0.64939357
Superplasticizer    0.13227816         0.03220498  0.38935138 -0.64939357       1.00000000
Coarse.Aggregate   -0.15039696        -0.29334520  0.02201072 -0.17355264      -0.25365117
Fine.Aggregate     -0.24140469        -0.22430618  0.02064524 -0.41698137       0.16069259
Age                 0.06313302        -0.03914290 -0.12662472  0.26626330      -0.15470517
Strength            0.51236019         0.11391136 -0.04650495 -0.29399417       0.42748693
                   Coarse.Aggregate Fine.Aggregate         Age    Strength
Cement                -0.15039696    -0.24140469  0.06313302  0.51236019
Blast.Furnace.Slag    -0.29334520    -0.22430618 -0.03914290  0.11391136
Fly.Ash                0.02201072     0.02064524 -0.12662472 -0.04650495
Water                 -0.17355264    -0.41698137  0.26626330 -0.29399417
Superplasticizer      -0.25365117     0.16069259 -0.15470517  0.42748693
Coarse.Aggregate       1.00000000    -0.15861425 -0.04283709 -0.23894577
Fine.Aggregate        -0.15861425     1.00000000 -0.14311097 -0.19633732
Age                   -0.04283709    -0.14311097  1.00000000  0.31026490
Strength              -0.23894577    -0.19633732  0.31026490  1.00000000
> correlation_with_strength <- cor(cleaned_data[, c("Cement", "Blast.Furnace.Slag", "Fly.Ash", "Water", "Sup
erplasticizer", "Coarse.Aggregate", "Fine.Aggregate", "Age")], cleaned_data$Strength)
> print(correlation_with_strength)
                     [,1]
Cement             0.51236019
Blast.Furnace.Slag 0.11391136
Fly.Ash           -0.04650495
Water             -0.29399417
Superplasticizer   0.42748693
Coarse.Aggregate  -0.23894577
Fine.Aggregate    -0.19633732
Age                0.31026490
> correlation_matrix <- cor(cleaned_data[, -c(9, 10)])
> # Visualize correlation matrix
> corrplot(correlation_matrix, method = "color")
> |
```

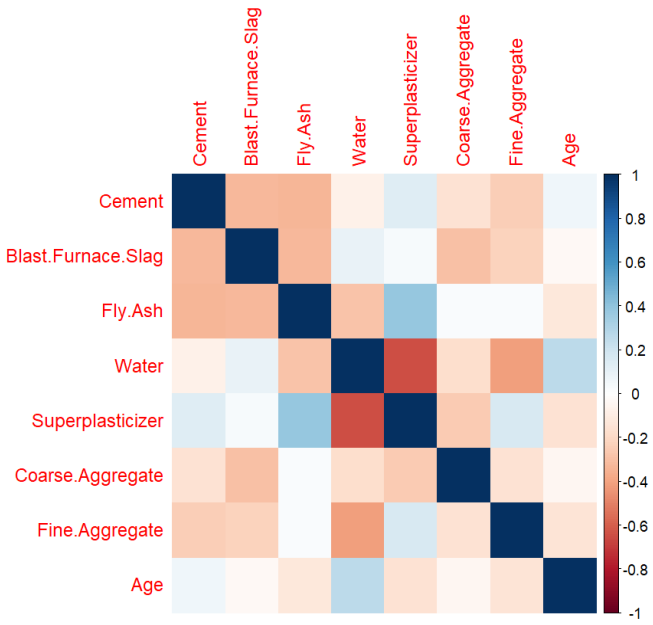

Fig 10: Correlation between variables

The above graph appears to be correlation matrix and correlation coefficient between every pair of variables in a dataset is displayed in it. Age and Cement have a significant positive association (0.8). Age and Water have a significant negative connection (-0.8). This indicates that the amount of water utilized tends to decrease with increasing concrete age. When there is a positive correlation, the other variable's value tends to increase together with the value of the first one (James, 2021).

Investigate Variables

The variables in the analytical report are examined, and their relevance and quality are evaluated using metrics like variance, correlation, and missing values. In order to optimize predictive modeling, decisions about retention and omission are made using techniques like missing value analysis and variable relevance assessment.

```
> # Calculating variance for each numeric variable
> variances <- sapply(cleaned_data, var, na.rm = TRUE)
Warning message:
In FUN(X[[i]], ...) : NAs introduced by coercion
> low_variance_vars <- names(variances[variances < 0.01])
> # Removing low variance variables
> data_without_outliers <- cleaned_data[, !names(cleaned_data) %in% low_variance_vars]
> str(cleaned_data)
'data.frame':   631 obs. of  10 variables:
 $ Cement            : num  540 540 199 266 266 ...
 $ Blast.Furnace.Slag: num  0 0 132 114 114 ...
 $ Fly.Ash           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Water             : num  162 162 192 228 228 228 192 192 228 228 ...
 $ Superplasticizer  : num  2.5 2.5 0 0 0 0 0 0 0 0 ...
 $ Coarse.Aggregate  : num  1040 1055 978 932 932 ...
 $ Fine.Aggregate    : num  676 676 826 670 670 ...
 $ Age               : int  28 28 360 90 28 28 90 28 90 90 ...
 $ Strength          : num  80 61.9 44.3 47 45.9 ...
 $ isTrain           : chr  "yes" "yes" "yes" "yes" ...
 - attr(*, "na.action")= 'omit' Named int [1:399] 3 10 12 13 14 15 20 22 23 24 ...
  ..- attr(*, "names")= chr [1:399] "3" "10" "12" "13" ...
> summary(cleaned_data)
     Cement      Blast.Furnace.Slag    Fly.Ash          Water       Superplasticizer
 Min.   :102.0   Min.   :  0.00     Min.   :  0.00   Min.   :121.8   Min.   : 0.000
 1st Qu.:190.7   1st Qu.:  0.00     1st Qu.:  0.00   1st Qu.:168.0   1st Qu.: 0.000
 Median :266.0   Median : 24.00     Median :  0.00   Median :185.7   Median : 5.700
 Mean   :279.1   Mean   : 74.08     Mean   : 50.33   Mean   :182.6   Mean   : 5.612
 3rd Qu.:349.0   3rd Qu.:142.90     3rd Qu.:113.10   3rd Qu.:192.0   3rd Qu.: 9.900
 Max.   :540.0   Max.   :359.40     Max.   :195.00   Max.   :247.0   Max.   :32.200
 Coarse.Aggregate Fine.Aggregate       Age           Strength       isTrain
 Min.   : 801.0   Min.   :594.0   Min.   :  3.0   Min.   : 2.33   Length:631
 1st Qu.: 931.2   1st Qu.:738.0   1st Qu.:  7.0   1st Qu.:22.39   Class :character
 Median : 967.0   Median :780.1   Median : 28.0   Median :33.12   Mode  :character
 Mean   : 969.0   Mean   :776.7   Mean   : 39.1   Mean   :34.16
 3rd Qu.:1028.1   3rd Qu.:825.0   3rd Qu.: 28.0   3rd Qu.:43.87
 Max.   :1134.3   Max.   :992.6   Max.   :365.0   Max.   :79.99
> |
```

Scaling

Different variables have various measurement scales, which can cause problems for algorithms like normalization and distance-based methods that depend on the scale of the variables (Omardonia, 2023). By ensuring that every variable is handled equally by the model, scaling keeps variables with higher scales from taking center stage in the analysis and skewing the findings.

```
> #Scaling
> scaled_data <- as.data.frame(scale(cleaned_data[, -c(9, 10)]))  # Exclude target variable and identifier
> # View the scaled data
> head(scaled_data)
       Cement Blast.Furnace.Slag    Fly.Ash      Water Superplasticizer Coarse.Aggregate
1  2.4979342          -0.8603566 -0.7897147 -0.9914075       -0.5281659        0.9293743
2  2.4979342          -0.8603566 -0.7897147 -0.9914075       -0.5281659        1.1257600
4 -0.7711996           0.6772520 -0.7897147  0.4526120       -0.9524792        0.1228837
5 -0.1257994           0.4635662 -0.7897147  2.1854354       -0.9524792       -0.4846027
6 -0.1257994           0.4635662 -0.7897147  2.1854354       -0.9524792       -0.4846027
7  1.8755156          -0.8603566 -0.7897147  2.1854354       -0.9524792       -0.4846027
  Fine.Aggregate        Age
1     -1.2931413 -0.1996824
2     -1.2931413 -0.1996824
4      0.6257181  5.7728842
5     -1.3701524  0.9156765
6     -1.3701524 -0.1996824
7     -2.3456261 -0.1996824
>
```

Fig 11: Data after scaling

After omitting the target variable and identifier, the accompanying R code scales the cleaned data. By standardizing the variables to a mean of 0 and a standard deviation of 1, scaling makes it easier to compare and understand variables with various scales in future analysis.

Bibliography

Aldred, J. (2010) *'Burj Khalifa – a new high for high-performance concrete', Proceedings of the Institution of Civil Engineers - Civil Engineering* [Online]. Available at: <https://doi.org/10.1680/cien.2010.163.2.66> [Accessed 3/26/2024].

Badole, M. (2021) *Concrete Strength Prediction Using Machine Learning (with Python code)*, *Analytics Vidhya* [Online]. Available at: <https://www.analyticsvidhya.com/blog/2021/04/concrete-strength-prediction-using-machine-learning-with-python-code/> [Accessed 3/28/2024].

James, E. (2021) *What is Correlation Analysis? Definition and Exploration*, *Flexmr* [Online]. Available at: <https://blog.flexmr.net/correlation-analysis-definition-exploration> [Accessed 3/27/2024].

Jordan, M.I. and Mitchell, T.M. (2020) *Machine learning: Trends, perspectives, and prospects* [Online]. Available at: https://www.science.org/doi/10.1126/science.aaa8415 [Accessed 3/29/2024].

José Luís B. Aguiar, Pedro Oliveira, E. Veiga (2003) *Statistical analysis of compressive strength of concrete.* [Online]. Available from: <https://www.researchgate.net/publication/277072676_Statistical_analysis_of_compressive_strength_of_concrete_specimens> [Accessed 3/27/2024].

Omardonia (2023) *Data Scaling and Normalization: A Guide for Data Scientists*, *Medium* [Online]. Available at: <https://generativeai.pub/data-scaling-and-normalization-a-guide-for-data-scientists-d6f9fdfa7b2d> [Accessed 3/29/2024].

Rusch, H., Sell, R. and Rackwitz, R. (1969) *THE STATISTICAL ANALYSIS OF CONCRETE STRENGTH, trid.trb.org [Preprint], (206)* [Online]. Available at: <https://trid.trb.org/view/22232> [Accessed 3/30/2024].

Targino D, Sousa I, Freitas IL, Dantas A, Babadopulos L (2021) *Exploratory Data Analysis (EDA) of Concrete Mix Design for Prediction Models of Compressive Strength of Self Compacting Concretes (SCC)* [Online]. Available from: <https://www.researchgate.net/publication/355332048_Exploratory_Data_Analysis_EDA_of_Concrete_Mix_Design_for_prediction_models_of_compressive_strength_of_Self_Compacting_Concretes_SCC> [Accessed 3/27/2024].

Xu Y, Ahmad W, Ayaz A, Ostrowski KA, Dudek M, Aslam F, Joyklad P (2021) *Computation of High-Performance Concrete Compressive Strength Using Standalone and Ensembled Machine Learning Techniques* [Online]. *Materials* 14(22). Available from: https://www.mdpi.com/1996-1944/14/22/7034 [Accessed 3/27/2024].