School of Computing, Creative Technology and Engineering

Student ID	77356719
Student Name	ROSHNA DHAKAL
Module Name & CRN	Applied Machine Learning
Level	Level 5 Semester 2
Assessment Name & Part No.	Assessment 1 Part I
Project Title	Data Analysis of Concrete Strength
Data of Submission	31 March, 2024
Course	BSc. Computing
Academic Year	2024

Table Of Contents

Abstract

The report explores machine learning techniques for predicting concrete strength using data analysis, pre-processing, and model creation. It examines exploratory data analysis, scale, correlation analysis, and data preparation. The study aims to improve structural performance and optimize concrete mixtures.

Introduction

Traditional analytical techniques for prediction of concrete strength may not be sufficient due to complex concrete mixing interactions. Understanding factors influencing strength can optimize concrete mix designs, allocate resources effectively, and identify potential risks. Accurate predictions can enhance cost savings, structural efficiency, and project schedules. The main objective of this report is to advance efforts to optimize concrete mixes, ensure safety, and reliability in construction projects, emphasizing the importance of data-driven approaches in improving concrete strength forecasting models and decision-making processes.

Literature Review

Recent advancements in data analysis techniques have allowed scientists to predict concrete composition and strength, using machine learning and statistical methods (Kamath et al., 2022). The literature on concrete strength prediction includes a variety of methods, including regression models, neural networks, and ensemble methods (Song et al., 2021). (Obianyo et al., 2020) used a multivariate regression model, while (Nguyen-Sy et al., 2020) used machine learning techniques such as random forest and gradient boosting to predict strength. These studies emphasize the importance of feature selection, data processing, and model estimation in achieving reliable predictions. Regression analysis helped to predict concrete strength based on variables such as cement, blast furnace slag and others (Kioumarsi et al., 2023). Data preprocessing techniques such as imputation and outlier detection have improved model accuracy (Fan et al., 2021). Exploratory Data Analysis (EDA) is a crucial method used to analyze concrete data with visual exploration, providing valuable insights into the distribution of concrete strength and potential outliers (Al Yamani et

al., 2023). The application of machine learning algorithms, including regression and advanced statistical methods, has shown promise in optimizing concrete mixes and improving structural performance in construction projects (Gamil, 2023).

Exploratory Data Analysis

The main objective of EDA is to understand the data, understand the underlying structure, find patterns, and identify possible correlations between variables (Lipovetsky, 2022). To perform this analysis first we load the test and train data set into R Studio.

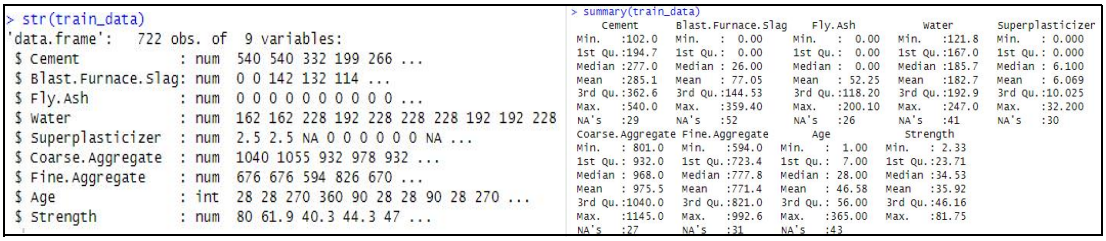
```
train_data <- read.csv("concrete_strength_train.csv")
test_data <- read.csv("concrete_strength_test.csv").
```

The description of variables used in concrete strength data set is given below:

1. Cement: binds materials together, measured in kg/m3.
2. Slag: a by-product of iron production, improves strength, measured in kg/m3.
3. Fly ash: a product of coal combustion, improves durability, measured in kg/m3.
4. Water: activates cement hydration, affects work ability, measured in kg/m3.
5. Super plasticizer: Improves work ability without sacrificing strength, measured in kg/m3.
6. Coarse Aggregate: Provides volume and stability, measured in kg/m3.
7. Fine Aggregate: Improves cohesion and surface treatment, measured in kg/m3.
8. Age: The development of strength is influenced by the amount of hardness in days.
9. Concrete Strength: Maximum bearing capacity, measured in MPa.

1.1 Train Dataset Overview

The concrete strength train data set consists of 722 samples and 9 columns representing various attributes related to concrete strength. It includes numerical variables such as Fly Ash, Coarse Aggregate, Blast Furnace Slag, Fine Aggregate, Superplasticizer, Age, Water, Cement, and the target variable Strength.
All variables appear to have numeric data types for their respective attributes.



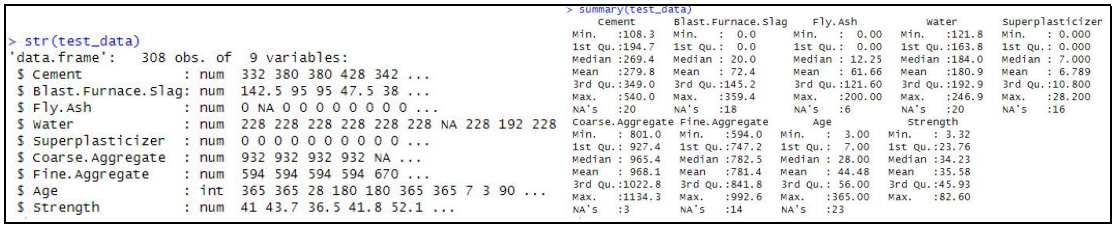
```
> str(train_data)
'data.frame':   722 obs. of  9 variables:
 $ Cement            : num  540 540 332 199 266 ...
 $ Blast.Furnace.Slag: num  0 0 142 132 114 ...
 $ Fly.Ash           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Water             : num  162 162 228 192 228 228 228 192 192 228
 $ Superplasticizer  : 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 ...
```

```
> summary(train_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.:194.7   1st Qu.:  0.00     1st Qu.:  0.00   1st Qu.:167.0   1st Qu.: 0.000
 Median :277.0   Median : 26.00     Median :  0.00   Median :185.7   Median : 6.100
 Mean   :285.1   Mean   : 77.05     Mean   : 52.25   Mean   :182.7   Mean   : 6.069
 3rd Qu.:362.6   3rd Qu.:144.53     3rd Qu.:118.20   3rd Qu.:192.9   3rd Qu.:10.025
 Max.   :540.0   Max.   :359.40     Max.   :200.10   Max.   :247.0   Max.   :32.200
 NA's   :29      NA's   :52         NA's   :26       NA's   :41      NA's   :30
 Coarse.Aggregate Fine.Aggregate       Age            Strength
 Min.   : 801.0   Min.   :594.0   Min.   :  1.00   Min.   : 2.33
 1st Qu.: 932.0   1st Qu.:723.4   1st Qu.:  7.00   1st Qu.:23.71
 Median : 968.0   Median :777.8   Median : 28.00   Median :34.53
 Mean   : 975.5   Mean   :771.4   Mean   : 46.58   Mean   :35.92
 3rd Qu.:1040.0   3rd Qu.:821.0   3rd Qu.: 56.00   3rd Qu.:46.16
 Max.   :1145.0   Max.   :992.6   Max.   :365.00   Max.   :81.75
 NA's   :27       NA's   :31      NA's   :43
```

Fig1: Structure & Summary of train_data

1.2 Test Dataset Overview

The test data set contains 308 samples with the same set of attributes as the train data set. Images provided below shows the data structure and summary statistics of test dataset.



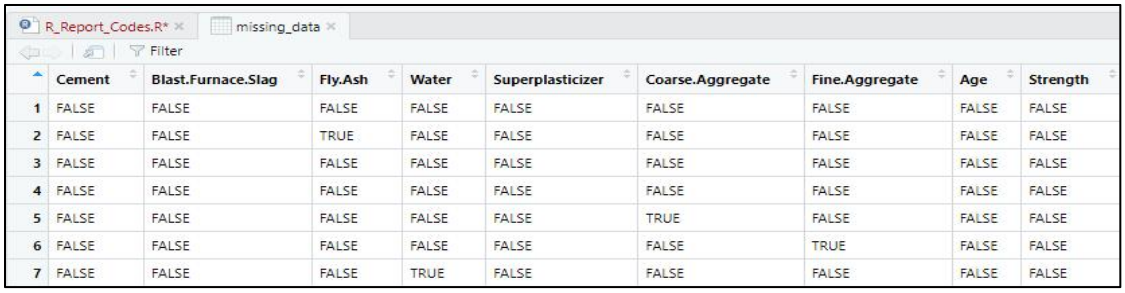
```
> str(test_data)
'data.frame':   308 obs. of  9 variables:
 $ Cement            : num  332 380 380 428 342 ...
 $ Blast.Furnace.Slag: num  142.5 95 95 47.5 38 ...
 $ Fly.Ash           : num  0 NA 0 0 0 0 0 0 0 0 ...
 $ Water             : num  228 228 228 228 228 228 NA 228 192 228
 $ Superplasticizer  : 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 ...
```

```
> summary(test_data)
     Cement      Blast.Furnace.Slag    Fly.Ash           Water       Superplasticizer
 Min.   :108.3   Min.   :  0.0      Min.   :  0.00   Min.   :121.8   Min.   : 0.000
 1st Qu.:194.7   1st Qu.:  0.0      1st Qu.:  0.00   1st Qu.:163.8   1st Qu.: 0.000
 Median :269.4   Median : 20.0      Median : 12.25   Median :184.0   Median : 7.000
 Mean   :279.8   Mean   : 72.4      Mean   : 61.66   Mean   :180.9   Mean   : 6.789
 3rd Qu.:349.0   3rd Qu.:145.2      3rd Qu.:121.60   3rd Qu.:192.9   3rd Qu.:10.800
 Max.   :540.0   Max.   :359.4      Max.   :200.00   Max.   :246.9   Max.   :28.200
 NA's   :20      NA's   :18         NA's   :6        NA's   :20      NA's   :16
 Coarse.Aggregate Fine.Aggregate       Age            Strength
 Min.   : 801.0   Min.   :594.0   Min.   :  3.00   Min.   : 3.32
 1st Qu.: 927.4   1st Qu.:747.2   1st Qu.:  7.00   1st Qu.:23.76
 Median : 965.4   Median :782.5   Median : 28.00   Median :34.23
 Mean   : 968.1   Mean   :781.4   Mean   : 44.48   Mean   :35.58
 3rd Qu.:1022.8   3rd Qu.:841.8   3rd Qu.: 56.00   3rd Qu.:45.93
 Max.   :1134.3   Max.   :992.6   Max.   :365.00   Max.   :82.60
 NA's   :3        NA's   :14      NA's   :23
```

Fig2: Structure & Summary of train_data

1.3 Exploring Missing Values:

Following the verification process, both data set have been identified with missing values (NA) which needs to be resolved to maintain data integrity.



R_Report_Codes.R* × | missing_data ×

Filter

	Cement	Blast.Furnace.Slag	Fly.Ash	Water	Superplasticizer	Coarse.Aggregate	Fine.Aggregate	Age	Strength
1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
5	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
7	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE

Fig3: Logical Table to Identify Missing Values(train)

In the figure above, the is.na(test_data) function creates a logical table i.e. missing_data and. Each member indicates whether the corresponding element is missing from the test_data or not. The function examines each member of the test_data returns TRUE if the element is missing and FALSE otherwise.

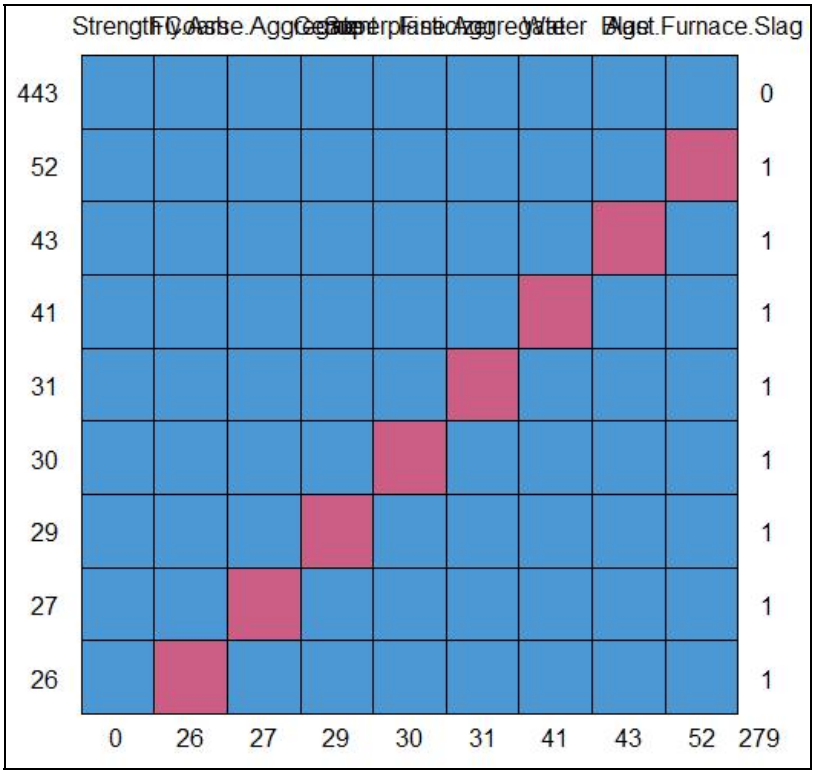


Fig4:md.pattern()(train_data)

```
> md.pattern(train_data)
    Strength Fly.Ash Coarse.Aggregate Cement Superplasticizer Fine.Aggregate Water Age Blast.Furnace.Slag
443        1      1                1      1                1              1     1   1                   1    0
52         1      1                1      1                1              1     1   1                   0    1
43         1      1                1      1                1              1     1   0                   1    1
41         1      1                1      1                1              1     0   1                   1    1
31         1      1                1      1                1              0     1   1                   1    1
30         1      1                1      1                0              1     1   1                   1    1
29         1      1                1      0                1              1     1   1                   1    1
27         1      1                0      1                1              1     1   1                   1    1
26         1      0                1      1                1              1     1   1                   1    1
           0     26               27     29               30             31    41  43                  52  279
>
```

Fig5: Missing Values Identified as 1 and 0 (train)

Further, The md.pattern() function in R, part of the mice package, is used to provide a summary of missing data patterns within a dataset.Each red box in fig 4 represents a

variable in the dataset that contains missing values identifies as md pattern. The height of the red box represents the number of missing values in that variable i.e 279. Similarly in figure 5, the numbers 1 and 0 represent the presence or absence of missing values for each variable in the data set respectively.

1.4. Descriptive Statistics

The main characteristics of a data set are described using descriptive statistics. They provide insight into the central tendency, distribution, and variability of the data.

```
> mean_strength
[1] 35.92154
> median_strength
[1] 34.535
> sd_strength
[1] 16.79693
> quantiles_strength
     0%     25%     50%     75%    100%
 2.3300 23.7100 34.5350 46.1575 81.7500
```

Fig 6: Descriptive statistics of strength variable

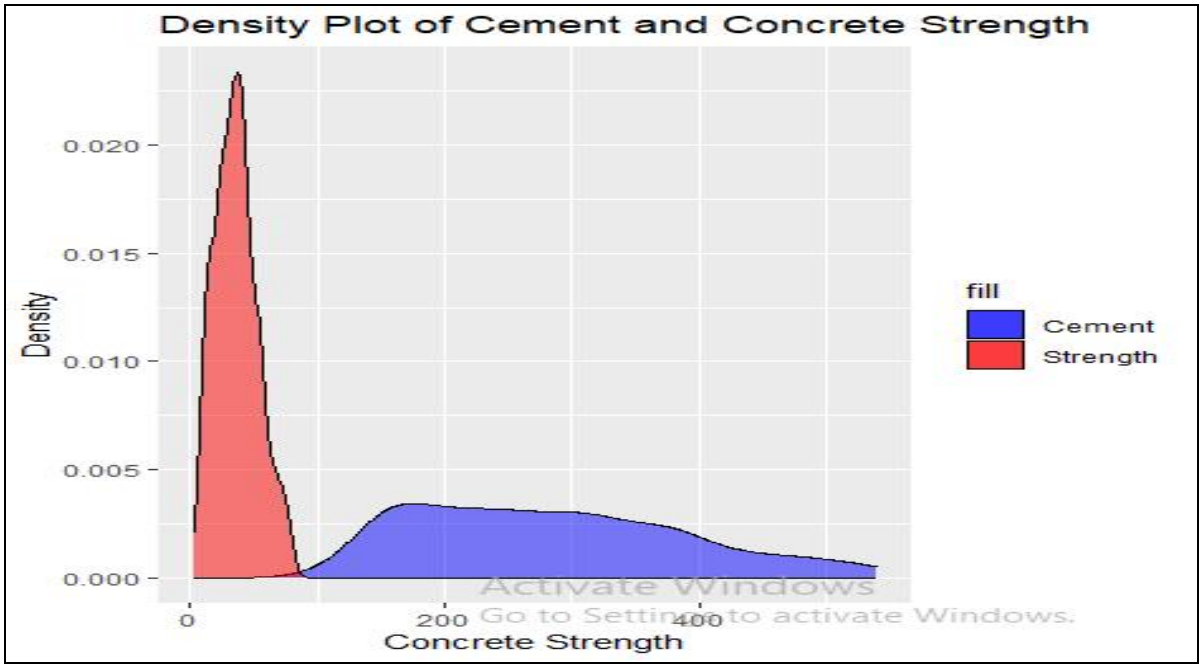


Fig7: Density plot of cement and concrete strength

The density plot in figure 7 displays the distributions of cement and concrete strength in a single graph, with the blue plot representing cement values and the red plot

representing concrete strength values. The x-axis shows the range of values, while the y-axis represents the density. Overlapping areas indicate areas where these distributions coincide, providing insights into their relationship and potential patterns.

1.4. Graphical Visualization

The below figure 8&9 illustrates a histogram and density plot of concrete strength. A line plot overlays the histogram and density plot, illustrating the relationship between concrete strength and cement content. A density plot provides insights into the continuous probability distribution of strength values.

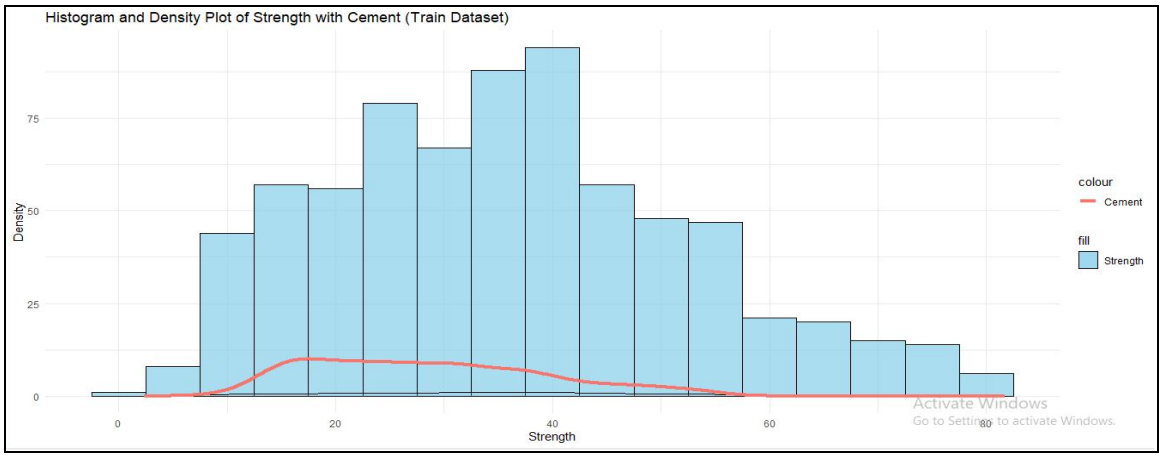


Fig8: Histogram and Density Plot of Strength with Cement (Train Data set)

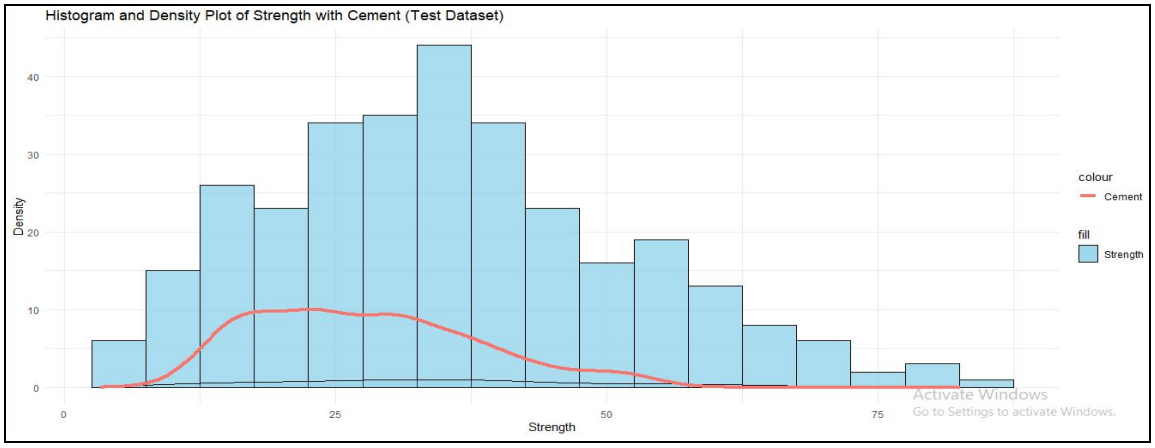


Fig9: Histogram and Density Plot of Strength with Cement (Test Data set)

Both test and train data set use a similar approach, using manual fill colors to distinguish between strength and cement representations. The strength is represented in blue, and cement is represented in red.

Data Preparation

2.1 Data Cleaning

Columns with missing values were handled to maintain data integrity. The process involved loading the datasets, checking for missing values, and implementing appropriate strategies to address them (Fan et al., 2021).

```
# Check for missing values
train_missing <- mean(colMeans(is.na(train_data)))
test_missing <- mean(colMeans(is.na(test_data)))
```

```
> train_missing
          Cement Blast.Furnace.Slag           Fly.Ash             Water   Superplasticizer
        4.016620           7.202216          3.601108          5.678670          4.155125
  Coarse.Aggregate     Fine.Aggregate               Age          Strength
        3.739612           4.293629          5.955679          0.000000
> test_missing
          Cement Blast.Furnace.Slag           Fly.Ash             Water   Superplasticizer
        6.493506           5.844156          1.948052          6.493506          5.194805
  Coarse.Aggregate     Fine.Aggregate               Age          Strength
        0.974026           4.545455          7.467532          0.000000
>
```

Fig10: Average of Missing Values in Each Column of Two Datasets

To handle missing values in the datasets, firstly column-wise missingness of data was examined. The average of missing values in each column was calculated for both test and train data individually using similar method to determine the extent of missing data. If the average of missing values in a column was greater than 1%, imputation using the mice () package was performed, otherwise, the missing values were removed.

```
> imputed_train_data <- mice(data =train_data, m = 1, method = "mean", maxit = 10)

 iter imp variable
  1   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  6   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  7   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  8   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  9   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  10   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
> imputed_test_data <- mice(data = test_data, m = 1, method = "mean", maxit = 10)

 iter imp variable
  1   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  6   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  7   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  8   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  9   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  10   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
> train_data_imputed <- complete(imputed_train_data)
> test_data_imputed <- complete(imputed_test_data)
```

Fig11: Imputed Train Data Using Mice Package

Data	
imputed_test_data	List of 22
imputed_train_data	List of 22
test_data	308 obs. of 9 variables
test_data_imputed	308 obs. of 9 variables
train_data	722 obs. of 9 variables
train_data_imputed	722 obs. of 9 variables

Fig12: Imputed Data Overview

The md.pattern() function was used to successfully impute missing data using mean imputation..md.pattern(train_data_imputed)

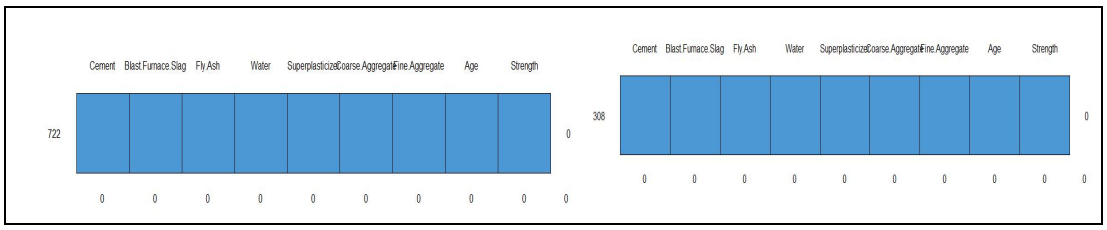


Fig13:md pattern of test and train data

This approach ensured that missing values were handled appropriately, preserving the integrity and completeness of data set for further analysis. A similar method was used to impute missing values for the training and test data sets.

2.2 Outliers

Data points that differ significantly from the rest of the data set are called outliers. Their values may be unusually high or low compared to most data points.Outliers in data set can vastly affect the performance of machine learning models, so it is important to handle them properly (Smiti, 2020).

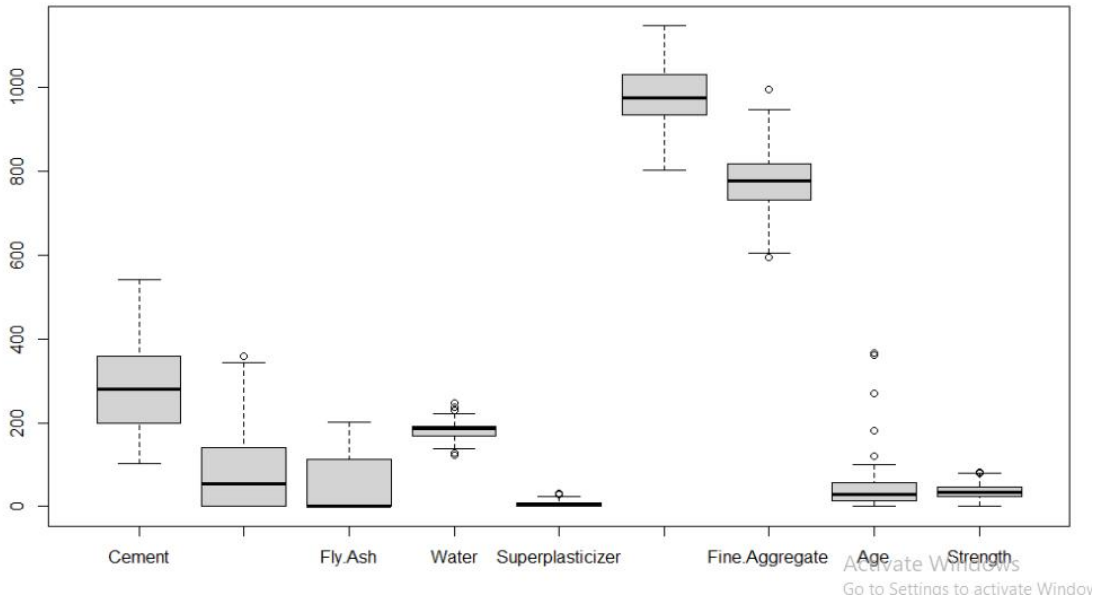


Fig14: Box plot with outliers for each column (train data set)

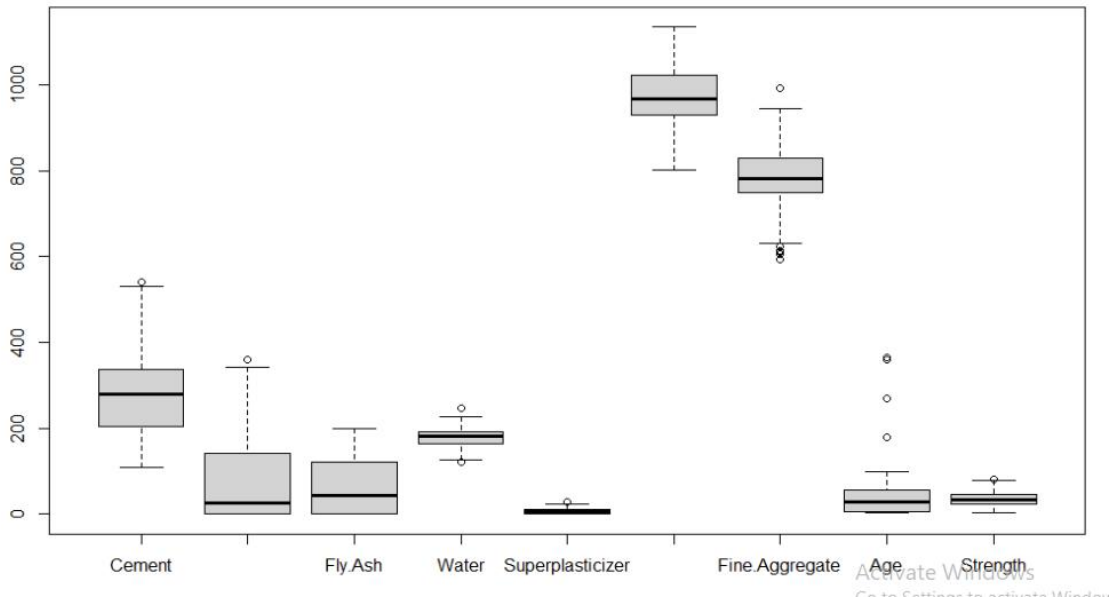


Fig15: Box plot with outliers for each column (test data set)

In fig 14 and 15, Box plots are generated for each column, with outliers represented by circles for both test and train data set. Outliers were observed for several variables and removed using the IQR method in both train and test data set.

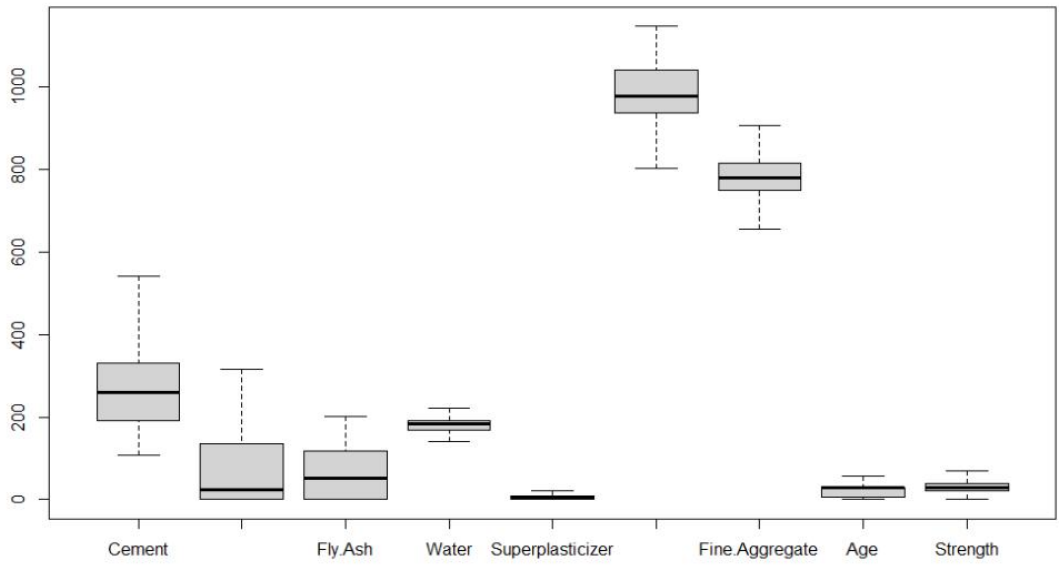


Fig16: Box plot after outlier removal for each column (train data)

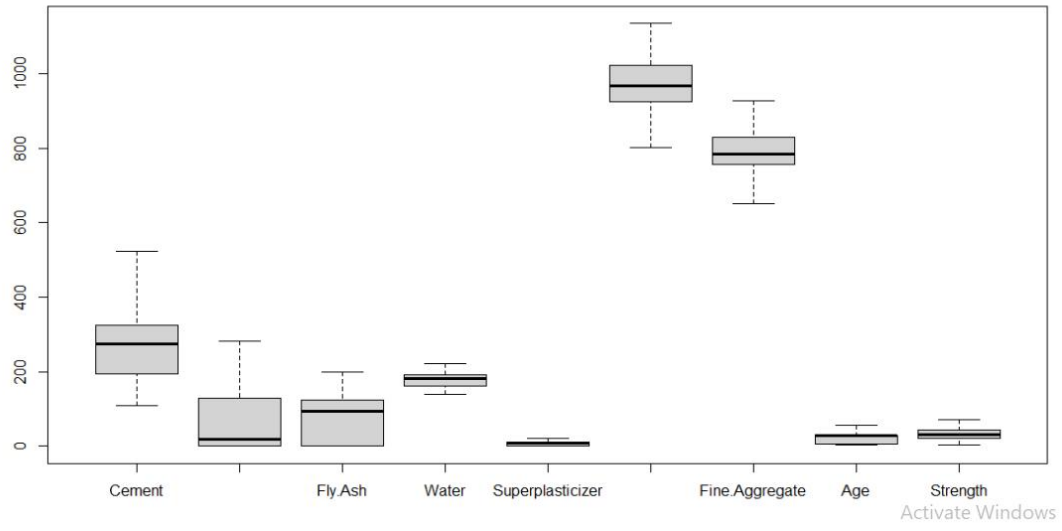


Fig17: Box plot after outlier removal for Each Column (test data set)

In figure 16 and 17 the outliers are removed hence there are no dots.

2.4. Scaling

During scaling, the mean of the variables is changed to 0 and the standard deviation to 1. With this approach, variables are standardized and placed on a common scale, which helps to make the model more stable and improve its performance through better predictions.

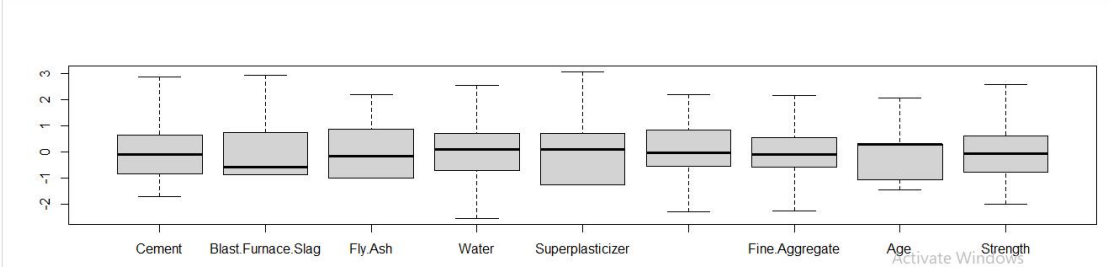


Fig18: Scaling Using scale () in Train Data set Without Outliers

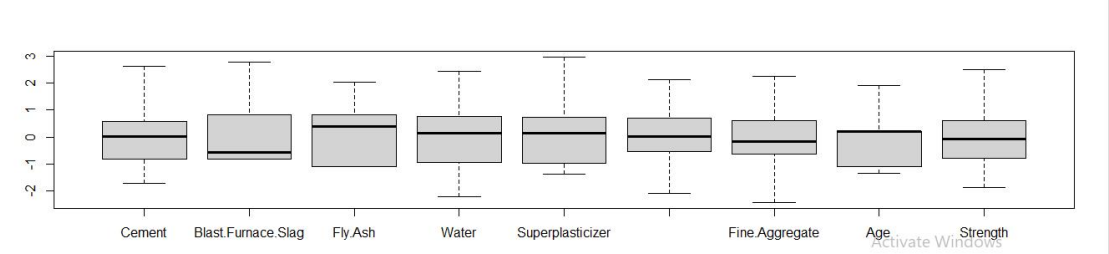


Fig19: Scaling using scale () in Test Data set Without Outliers

The above fig18 and 19s represents the box plots visualization of scaled train and test data set using scale () function in R which provides z-score normalization of data.

2.3 Correlation Analysis

The correlation matrix and visualization demonstrate the degree of association and direction of predictor variables. High correlations such as 1 or -1, signifies a strong relationship between variables. If the variables are highly correlated, correlation coefficient is > 0 (GfG, 2023).

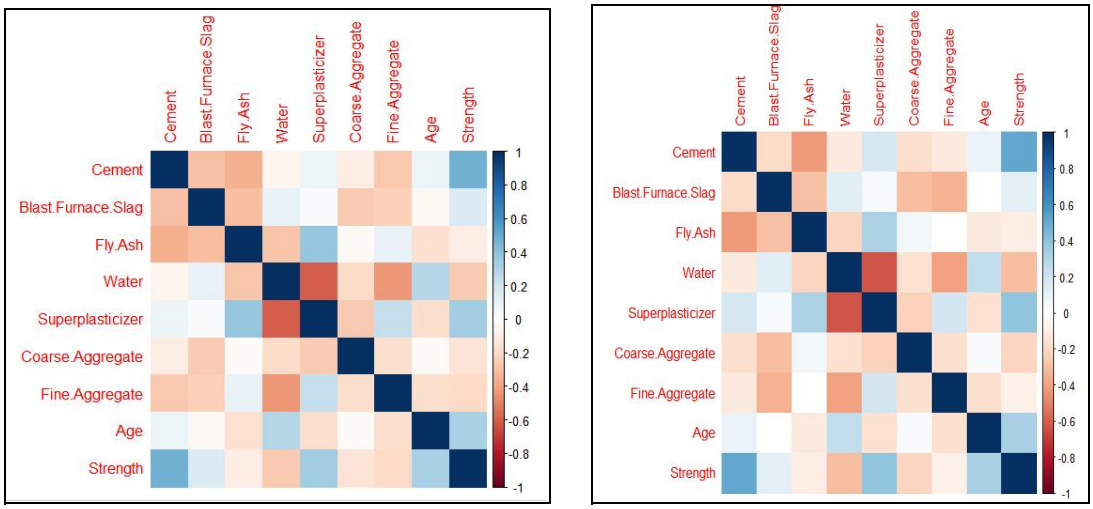


Fig20: Correlation Matrix of train(left) & test(right) dataset.

In the above Fig 20, Positive correlations are displayed in blue and negative correlations in shades of red. A stronger correlation is observed with darker colors, whereas a weaker correlation exits with lighter colors. The diagonal line of the correlation matrix shows each variable's correlation with itself which is always 1.

Since all the variables in concrete strength data set have equal significance in model prediction and further data analysis, removing the columns may hamper the data insights and integrity. Hence, it was decided to not remove the columns with higher correlation probability.

Bibliography

Al Yamani, W. H., Ghunimat, D. M. & Bisharah, M. M. (2023) Modeling and Predicting the Sensitivity of High-Performance Concrete Compressive Strength Using Machine Learning Methods. Asian Journal of Civil Engineering [Online], 24 (7) March, pp. 1943–1955. Available from: <http://dx.doi.org/10.1007/s42107-023-00614-4>.

Applied Sciences [Online], 12 (1) December, p. 361. Available from: <http://dx.doi.org/10.3390/app12010361>.

Fan, C., Chen, M., Wang, X., Wang, J. & Huang, B. (2021) A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. Frontiers in Energy Research [Online], 9 March. Available from: <http://dx.doi.org/10.3389/fenrg.2021.652801>.

Gamil, Y. (2023) Machine Learning in Concrete Technology: A Review of Current Researches, Trends, and Applications. Frontiers in Built Environment [Online], 9 February. Available from: <http://dx.doi.org/10.3389/fbuil.2023.1145591>.

GfG (2023) Correlation Matrix in R Programming [Online]. GeeksforGeeks. Available from: <https://www.geeksforgeeks.org/correlation-matrix-in-r-programming/>.

Kamath, M. V., Prashanth, S., Kumar, M. & Tantri, A. (2022) Machine-Learning-Algorithm to Predict the High-Performance Concrete Compressive Strength Using Multiple Data. Journal of Engineering, Design and Technology [Online], 22 (2) February, pp. 532–560. Available from: <http://dx.doi.org/10.1108/jedt-11-2021-0637>.

Kioumarsi, M., Dabiri, H., Kandiri, A. & Farhangi, V. (2023) Compressive Strength of Concrete Containing Furnace Blast Slag; Optimized Machine Learning-Based Models. Cleaner Engineering and Technology [Online], 13 April, p. 100604. Available from: <http://dx.doi.org/10.1016/j.clet.2023.100604>.

Lipovetsky, S. (2022) Explanatory Model Analysis: Explore, Explain and Examine Predictive Models,. Technometrics [Online], 64 (3) July, pp. 423–424. Available from: <http://dx.doi.org/10.1080/00401706.2022.2091871>.

Nguyen-Sy, T., Wakim, J., To, Q.-D., Vu, M.-N., Nguyen, T.-D. & Nguyen, T.-T. (2020) Predicting the Compressive Strength of Concrete from Its Compositions and Age Using the Extreme Gradient Boosting Method. Construction and Building Materials [Online], 260 November, p. 119757. Available from: <http://dx.doi.org/10.1016/j.conbuildmat.2020.119757>.

Obianyo, I. I., Anosike-Francis, E. N., Ihekweme, G. O., Geng, Y., Jin, R., Onwualu, A. P. & Soboyejo, A. B. O. (2020) Multivariate Regression Models for Predicting the Compressive Strength of Bone Ash Stabilized Lateritic Soil for Sustainable Building.

Construction and Building Materials [Online], 263 December, p. 120677. Available from: <http://dx.doi.org/10.1016/j.conbuildmat.2020.120677>.

Song, Y., Zhao, J., Ostrowski, K. A., Javed, M. F., Ahmad, A., Khan, M. I., Aslam, F. & Kinasz, R. (2021) Prediction of Compressive Strength of Fly-Ash-Based Concrete Using Ensemble and Non-Ensemble Supervised Machine-Learning Approaches.

Smiti, A. (2020) A Critical Overview of Outlier Detection Methods. Computer Science Review [Online], 38 November, p. 100306. Available from: <http://dx.doi.org/10.1016/j.cosrev.2020.100306>