**Concrete Compressive Strength Dataset (Dataset3)**

I uploaded the dataset **changed** it to a **DataFrame** and did the following:

1. **Changed** the **names** of the column **headings** respectively to:

Cement

Slag

Ash

Water

Superplasticizer

Coarse

Fine

Age

Strength

1. df.isna().sum() to check if there are any missing data(0missing)

Cement 0

Slag 0

Ash 0

Water 0

Superplasticizer 0

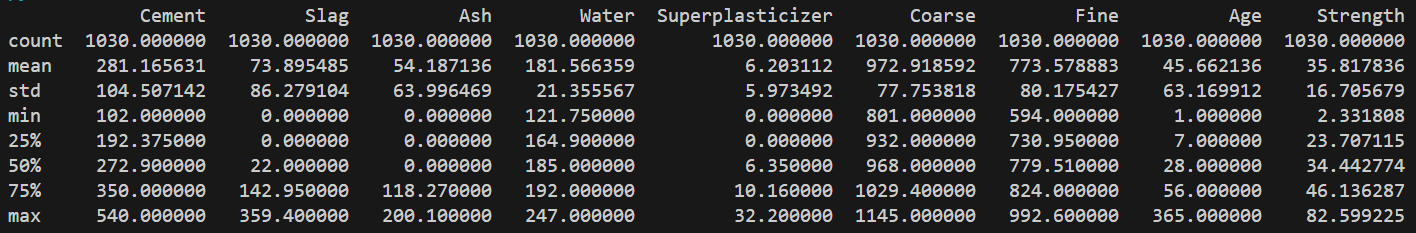
Coarse 0

Fine 0

Age 0

Strength 0

1. df.describe() to get an overview over the quartiles, std, mean, etc.



1. Named a **variable X** for the features (8 columns without “Strength”)

And **variable Y** to represent the output (Strength)

1. Plotted the Strength (output) against frequency on a **histogram**.

A blue graph with white text

Description automatically generated

1. **Split** the data into **training and testing** using:

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.25, random\_state=42)

1. A graph with blue bars

   Description automatically generatedUsed **RandomForest** to find Feature and permutation importance and found that:

Age 0.327427

Cement 0.325368

Water 0.121575

Slag 0.073777

Superplasticizer 0.065047

Fine 0.038756

Coarse 0.027756

Ash 0.020294

Fine, Coarse and Ash don’t contribute much (they don’t affect) to the output so…

1. Calculated the values of MeanAbsoluteError(MAE), MeanSquaredError(MSE), RootMeanSquaredError(RMSE), R-Squared using **SVR**
2. before scaling:

MAE: 11.72049

MSE: 209.90670

RMSE: 14.48815

R-squared: 0.22507

1. After applying **StandardScaler**:

MAE: 7.86030

MSE: 98.04289

RMSE: 9.90166

R-squared: 0.63804

1. After applying **MinMaxScaler**, we find that StandardScaler gave better results:

MAE: 8.46325

MSE: 112.12074

RMSE: 10.58870

R-squared: 0.58607

1. After **dropping columns** (Coarse and Ash) we get better values than the best run(ii.).

MAE: 7.34956

MSE: 84.72013

RMSE: 9.20435

R-squared: 0.68723

From this we conclude that Coarse and Ash do not contribute to the output as much as of the other features, so their columns can be dropped (feature reduction)

For better results:

1. MAE, MSE, RMSE should have small values (the smaller the better)
2. **A higher R-squared value shows a better-fitting model (1 is a perfect fit, 0.7-0.9 strong fit, etc.) Keep in mind that a very high R-squared value might indicate overfitting.**