# Task 5 : Apply Predictive Models on the processed data

**Introduction**

The purpose of this project is to perform regression analysis on a dataset containing properties of cement samples and their corresponding compression strengths. The dataset has 9 columns, all containing float values, which means we are dealing with continuous data. Our objective is to build regression models that can accurately predict the compression strength of cement samples based on their properties.

**Data Overview**

The dataset consists of 6139 samples, each with 9 columns. The first eight columns represent the input features, and the last column is the target variable "Compression Strength MPa." All features are of float type and represent quantities and components used in the cement formulation.

**Data Preprocessing**

We begin by importing the necessary libraries, including NumPy, Pandas, and various modules from scikit-learn for preprocessing and modeling. We also suppress any warnings to keep the output clean. Since it was asked by our mentor to add 10 extra rows in the data after preprocessing, we have removed the null 109 rows, and then added the ten extra rows by using data augmentation technique [refer folder Task3]. After in EDA [Task4] we have applied polynomial featuring on this data which left us with 6040 samples each with 53 columns.

**Data Splitting**

We then split the data into input variables (X) and the target variable (y). The input variables (X) consist of the first 9 columns, while the target variable (y) is the "Compression Strength MPa" column.

**Model Selection and Evaluation**

**Linear Regression**

To start the analysis, a Linear Regression model is applied. The model is trained on the training data and evaluated on the test data using R-squared.

* Test R-squared score: 0.280

To obtain a more robust estimate of model performance, cross-validation is performed. The average cross-validation R-squared score is found to be 0.251.

**Lasso Regression**

Due to the low R-squared score from Linear Regression, regularization techniques are applied to mitigate overfitting. Lasso Regression with an alpha of 0.3 is used, and the model is evaluated.

* Lasso R2 Score: 0.278

Cross-validation is conducted to validate the Lasso model, with an average R-squared score of 0.242.

**Ridge Regression**

Another regularization technique, Ridge Regression, is applied with an alpha of 0.4.

* Ridge R2 Score: 0.280

Cross-validation yields an average R-squared score of 0.242.

**Feature Scaling and SVR**

As the R-squared scores are still relatively low, we attempt to improve model performance by applying feature scaling using MinMaxScaler. We then train three SVR models with different kernels: linear, radial basis function (RBF), and polynomial.

Linear SVR

* MSE: 0.025
* R2 Score: 0.283

RBF SVR

* MSE: 0.023
* R2 Score: 0.336

Polynomial SVR

* MSE: 0.028
* R2 Score: 0.212

**Gradient Boosting Regression**

Finally, we apply the Gradient Boosting Regression model and perform hyperparameter tuning using GridSearchCV to find the best combination of hyperparameters.

Before Hyperparameter tuning

* MSE: 0.021
* R2 Score: 0.401

After Hyperparameter tuning

* MSE: 0.020
* R2 Score: 0.416

**Random Forest Regression**

As another ensemble model, we also apply the Random Forest Regression model and perform hyperparameter tuning using GridSearchCV to find the best combination of hyperparameters.

Before Hyperparameter tuning

* MSE: 0.021
* R2 Score: 0.397

After Hyperparameter tuning

* MSE: 0.020
* R2 Score: 0.427

**Conclusion**

After applying several regression models, including Linear Regression, Lasso Regression, Ridge Regression, Support Vector Regression (SVR) with different kernels, Gradient Boosting Regression, and Random Forest Regression, we observed varying levels of R-squared scores, indicating the accuracy of the models in predicting the compression strength of cement samples.

The Gradient Boosting Regression model achieved the highest R-squared score of approximately 0.40. By performing hyperparameter tuning on the Gradient Boosting and Random Forest models, we further improved the R-squared score for both models, achieving approximately 0.42 for the Random Forest model.