Machine Learning Project

The goal of this machine learning project is to predict concrete compressive strength using various supervised learning methods, and to identify the model that provides the best predictive performance.

I) Data investigation

Dataset description

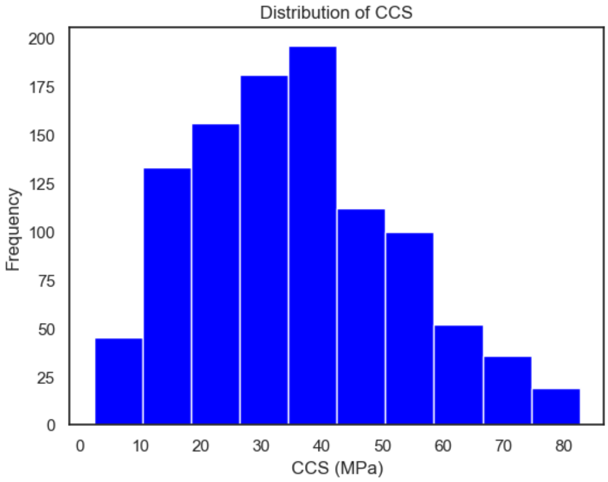
To predict concrete compressive strength, we use a dataset containing 1030 observations. The variables include the main ingredients of a concrete mixture as well as its compressive strength.

The target variable (Y) is the concrete compressive strength (MPa), while the other variables serve as features : Cement (kg/m^3), Blast Furnace Slag (kg/m^3), Fly Ash (kg/m^3), Water (kg/m^3), Superplasticizer (kg/m^3), Coarse Aggregate (kg/m^3), Fine Aggregate (kg/m^3), Age (days)

This dataset therefore provides the proportions of the different ingredients along with the compressive strength results for each mixture.

Target variable presentation

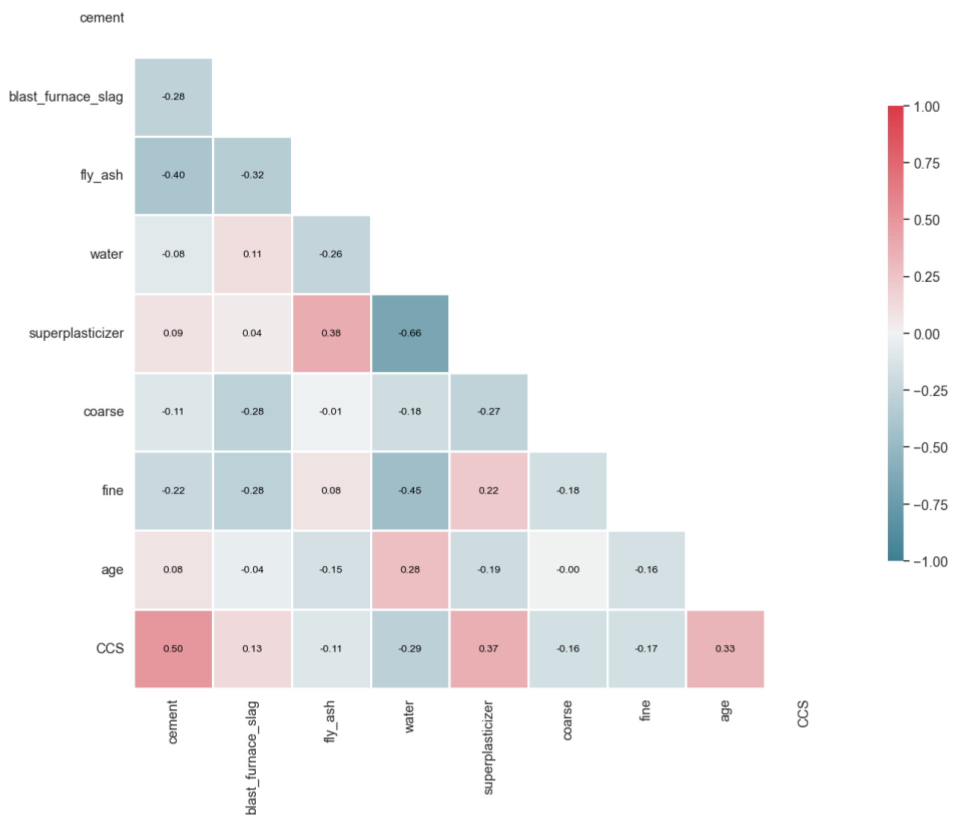
Figure 1 : Distribution of the target variable



The Concrete Compressive Strength (CCS) variable has a mean of 35.8 MPa, with a minimum of 2.3 MPa, indicating that some mixtures in our dataset result in very low strength, while the maximum reaches 82.6 MPa, corresponding to high-strength concrete. This wide range of values reflects high variability in the target variable, which is ideal for machine learning models, as they have more information to learn from and to understand how the features influence concrete strength. These observations suggest that the data will allow the creation of robust models.

Correlations

Figure 2 : Correlation Heatmap



The correlation heatmap allows us to observe the relationships between the different variables, particularly with the target variable (CCS) :

- Cement – CCS: 0.498 → higher cement content increases strength, confirming the central role of cement in concrete durability.

- Superplasticizer – CCS: 0.366 → adding superplasticizer improves strength by enhancing particle coating and reducing water content.
- Age – CCS: 0.329 → as the concrete ages, it becomes stronger.
- Water – CCS: -0.289 → higher water content decreases strength.

These observations indicate that these variables are interesting candidates for the models we will apply due to their significant correlation with CCS.

II) Data preparation for ML

Handling Missing Values and Outliers

The dataset has been sufficiently preprocessed, so no missing values are present. However, we identified outliers using the Z-score with a threshold of 3. A total of 49 outliers were detected, 33 of which concern the Age variable. This dispersion is explained by the range of sample ages, from 1 to 365 days.

In civil engineering, the characteristic strength of concrete is typically measured at 28 days. Higher values do not represent data entry errors but correspond to concrete tested at later ages, such as after one year. These are therefore extreme values rather than true outliers, and they can provide valuable information on long-term strength.

For the other variables, some extreme values reflect atypical mixtures compared to standard concrete, representing special formulations. These cases can help machine learning models learn to handle unusual scenarios.

Therefore, we decided to retain these extreme values due to their low number and the important information they provide for model training.

Train-Test split

To train and evaluate our models, we split the dataset into a training set (70 %) and a test set (30 %). The test set was then further divided into two equal parts: a validation set (15 % of the total data) and a final test set (15 %).

Therefore, our models learn from the training set, which contains 721 observations. Hyperparameters are tuned using the validation set (154 observations), where we make initial predictions and compare results to optimize the model parameters. Finally, once the model is selected and optimally configured, we evaluate it on the final test set (155 observations).

Table 1 : Data Split for Training, Validation, and Testing

	Training	**Validation**	**Test**
Percentage	70%	15%	15%
Number of observations	721	154	155

Scaling the data

Before training our models, we standardized the explanatory variables (features X). Standardization ensures that all variables contribute equally to the model. This step is particularly important in our case, as some variables exhibit highly heterogeneous distributions (e.g., Cement, Blast Furnace Slag, Fly Ash, Superplasticizer, or Age). After standardization, the values are centered around zero with a standard deviation of 1.

III) Application of Machine Learning Methods

In this section, we present the various machine learning methods applied. Based on the analysis of their performance, we will identify the best models, which we will then attempt to improve by tuning their hyperparameters in order to select the most effective model overall.

Models presentation

Table 2 : Models Applied in the Analysis

Models	**Linear regression**	**KNN**	**Decision Tree**	**Random Forest**	**Gradiant Boosting**	**XGBoost**
Types	Linear	Non-linear	Non-linear	Non-linear	Non-linear	Non-linear

These models were selected because they are suitable for predicting a continuous quantitative variable such as concrete compressive strength. They are capable of capturing simple linear relationships (linear regression), local patterns (KNN), and complex non-linear relationships (decision trees, Random Forest, Gradient Boosting, XGBoost).

Models performance

Initially, we compare the performance of the models using their default hyperparameters. Models are evaluated using R^2 to measure explained variance, RMSE to penalize large prediction errors, and MAE to assess the average magnitude of errors, providing a balanced evaluation of predictive accuracy.

Table 3 : Models performances

Model	**R2**	**RMSE**	**MAE**
Linear Regression	0.563423	10.625104	8.538887
K nearest neighbors	0.709515	8.666916	6.641972
Decision Tree	0.841770	6.396567	4.216588
Random Forest	0.913957	4.716948	3.637847
Gradient Boosting Tree	0.916052	4.659175	3.637605
XGB Regressor	0.915976	4.661273	3.092679

The table shows the performance of the models. Linear regression clearly underperforms compared to the more complex models. This is because, in the case of concrete compressive strength, the interactions between variables (the components) are complex and non-linear. Models capable of capturing these non-linear relationships, such as XGBoost, therefore achieve much better performance.

Among the tested models, three stand out with strong performance: Random Forest, Gradient Boosting, and XGBoost. These are the models we retain for the subsequent analysis.

Tuning the best models

After identifying the three best models with their default hyperparameters, we aim to improve their performance by tuning these parameters. Hyperparameter optimization allows the model to better adapt to the specifics of our dataset and to more precisely capture the complex relationships between the concrete components and its compressive strength.

To determine the optimal parameters, we used Grid Search CV from Scikit-learn. This method explores all possible combinations of hyperparameters, with each combination evaluated through cross-validation to select the one that minimizes prediction error.

Table 4 : Tested Hyperparameters and Optimal Values for Each Model

Model	Hyperparameter	Values Tested	Optimal Value
Random Forest	n_estimators	100, 200, 300	300
	max_depth	None, 5, 10, 20	20
	min_sample_split	2, 5, 10	2
	min_samples_leaf	1, 2, 4	1

Model	Hyperparameter	Values Tested	Optimal Value
XGBoost	n_estimators	100, 200, 300	300
	learning_rate	0.01, 0.05, 0.1	0.1
	max_depth	3, 5, 7	5
	subsample	0.8, 1.0	0.8
	colsample_bytree	0.8, 1.0	1.0

Model	Hyperparameter	Values Tested	Optimal Value
Gradient Boosting	n_estimators	100, 200, 300	300
	learning_rate	0.01, 0.05, 0.1	0.05
	max_depth	3, 5, 7	5
	subsample	0.8, 1.0	0.8

Performance after hyperparameter tuning

Table 5 : Models performance before and after tuning

Model	Before Tuning (R2 / RMSE / MAE)	After Tuning (R2 / RMSE / MAE)
Random Forest	0.913 / 4.717 / 3.638	0.911 / 4.786 / 3.653
Gradient Boosting	0.916 / 4.659 / 3.638	0.949 / 3.600 / 2.603
XGBoost	0.915 / 4.661 / 3.093	0.946 / 3.754 / 2.731

Hyperparameter tuning resulted in a modest yet meaningful performance improvement for two models, Gradient Boosting and XGBoost, while Random Forest experienced a negligible and non-significant decrease. Overall, Gradient Boosting emerged as the best-performing model, achieving an R^2 of 0.95, an RMSE of 3.600, and an MAE of 2.603, thereby slightly outperforming the other approaches.

Final model evaluation

As demonstrated earlier, Gradient Boosting stands out as the best-performing model, closely followed by XGBoost. We now proceed to evaluate it on the test set, which was not used during training. This step assesses its predictive ability on unseen data and helps determine its potential for deployment.

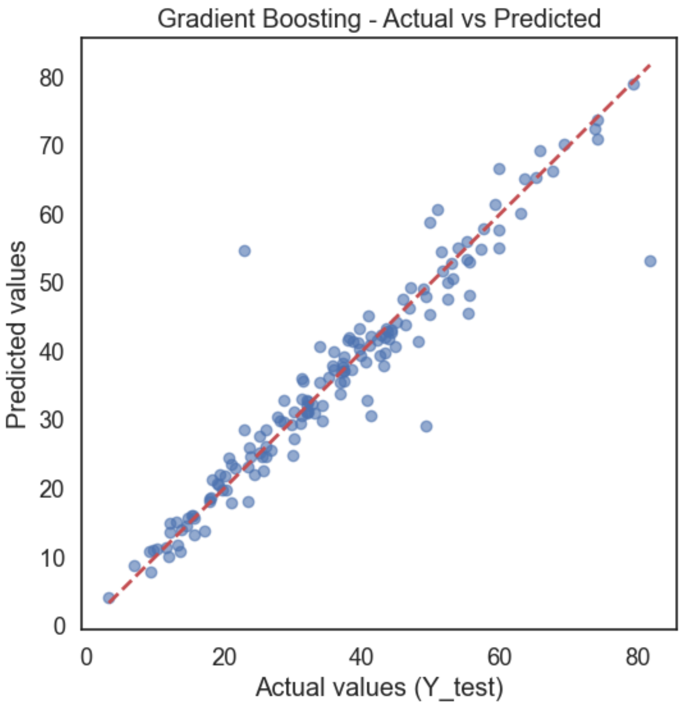
After selecting the optimal hyperparameters through cross-validation, we retrain the Gradient Boosting model on the full training set (train + validation) to leverage all available data. We then evaluate it on the test set to assess its generalization ability. Since XGBoost showed performances very close to Gradient Boosting, we also evaluate it on the test set.

Table 6 : Final Model Performance on Train/Validation vs Test Sets

Model	**Dataset**	**R2**	**RMSE**	**MAE**
Gradient Boosting	Train/Validation	0.949	3.600	2.603
	Test	0.914	4.814	2.65
XGBoost	Train / Validation	0.946	3.754	2.731
	Test	0.912	4.864	2.565

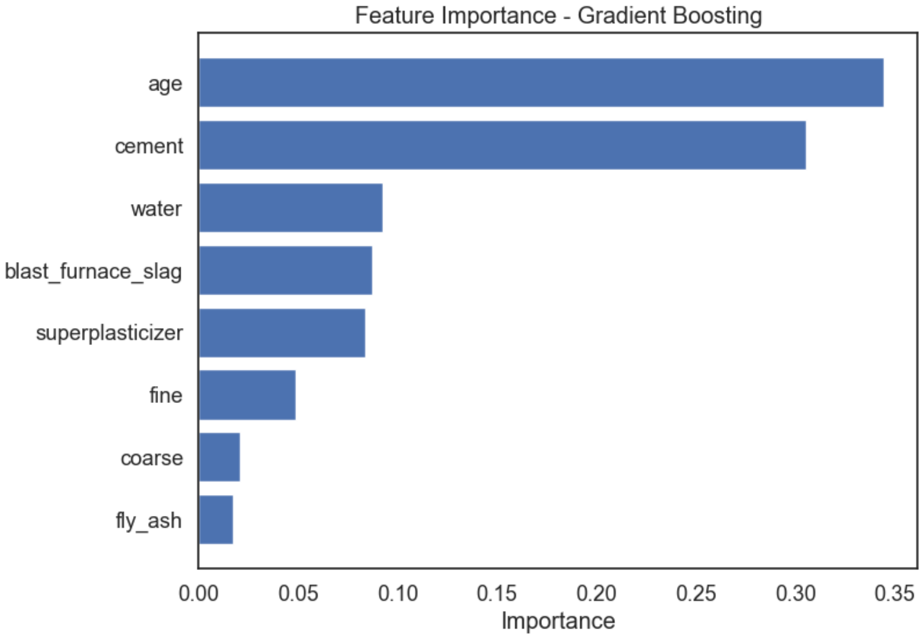
The test set results confirm the superiority of Gradient Boosting, although the difference with XGBoost remains marginal. The close performance between train/validation and test sets demonstrates good generalization ability for both models.

Figure 3 : Actual values vs Predicted values



The scatter plot shows that the predicted values from the Gradient Boosting model are closely aligned with the actual values, lying near the diagonal line representing perfect predictions. This visual confirmation supports the strong performance metrics obtained on the test set.

Figure 4 : Feature importance



Although the exact relationships between the variables and compressive strength cannot be directly interpreted in this non-linear model, feature importance analysis highlights which variables most influence the predictions. The most important variable is the concrete's Age, followed by Cement and Water

IV) Conclusion

In conclusion, this project allowed us to compare various machine learning methods for predicting concrete compressive strength. After evaluating initial performance and tuning hyperparameters, the Gradient Boosting model proved to be the most effective, demonstrating strong generalization on the test set. The results indicate that this model can be reliably used to predict concrete strength on new data, providing a robust tool for decision-making in civil engineering applications.