Report: Model Comparison for Predicting Concrete Strength

Problem Statement:

The task is to predict the **compressive strength of concrete** based on various features in the dataset. Concrete strength is a continuous variable, and the goal is to apply regression techniques to accurately forecast its value for unseen data. We will compare two regression models, **Gradient Boosting Regressor (GBR)** and **Random Forest Regressor (RFR)**, using **Root Mean Squared Error (RMSE)** and **R-squared (R^2)** as evaluation metrics to determine which model is more effective in predicting the compressive strength of concrete.

Dataset Overview:

The dataset `maajdl/yeh-concret-data` includes the following features:

- **Cement**: Amount of cement in the mix.
- **Blast Furnace Slag**: Amount of blast furnace slag.
- **Fly Ash**: Amount of fly ash.
- **Water**: Amount of water.
- **Superplasticizer**: Amount of superplasticizer.
- **Coarse Aggregate**: Coarse aggregate in the mix.
- **Fine Aggregate**: Fine aggregate in the mix.
- **Age**: Age of the concrete (in days).

The target variable is the **compressive strength** of the concrete, and the objective is to predict this value based on the above features.

Model Comparison:

To assess the models' effectiveness, we used two popular regression algorithms:

1. **Gradient Boosting Regressor (GBR)**: GBR is a powerful ensemble method that combines the predictions of multiple weak models (typically decision trees). We tuned the hyperparameters of GBR using **Bayesian Optimization** via **Optuna**, which allowed us to find the optimal set of hyperparameters such as:
   - `n_estimators` (number of trees)
   - `learning_rate` (step size for updates)
   - `max_depth` (maximum depth of trees)
   - Other parameters like `subsample`, `min_samples_split`, and `min_samples_leaf`.
   - Also tuned the model by implementing a feature selection process with threshold value 0.05.
2. **Random Forest Regressor (RFR)**: Random Forest is another ensemble learning method, which builds multiple decision trees, each using a random subset of the data. In this case, we used the default hyperparameters of the **Random Forest Regressor** to provide a baseline for comparison.

Key Insights:

1. **Gradient Boosting Regressor (Tuned)**:
   - After applying **Bayesian Optimization** with **Optuna**, the **Gradient Boosting Regressor** significantly outperformed the **Random Forest Regressor** in terms of both **normalized RMSE** and **R-squared**.
   - The **normalized RMSE** for the tuned Gradient Boosting Regressor was **0.5335**, indicating better accuracy in predicting compressive strength.
   - The **R^2** value for the tuned Gradient Boosting Regressor was **0.9288**, meaning the model explained **92.88%** of the variance in concrete compressive strength.
   - The hyperparameter tuning process allowed the model to better capture the complex relationships in the data.
2. **Random Forest Regressor (Default)**:
   - The **Random Forest Regressor**, without hyperparameter tuning, achieved a **normalized RMSE of 0.6806**.
   - The **R^2** for the Random Forest Regressor was **0.8841**, indicating that the model explained **89.42%** of the variance in the target variable.
   - Although it performed well, Random Forest was not as optimized as the Gradient Boosting model, resulting in a slightly lower R^2 and RMSE value.

Comparison and Conclusion:

Tuned Gradient Boosting Regressor outperformed **Random Forest Regressor** in both **normalized RMSE** and **R^2**, indicating better predictive power and model fit. While **Random Forest** performed well out-of-the-box, it would benefit from hyperparameter optimization to further improve its performance.

Written by Daniel.