CW1 Regression Challenge Report

K23154082

1 Introduction

This assignment is a supervised regression problem: given a tabular dataset, the goal is to use the remaining features to predict the target variable (`outcome`). The method for evaluating performance is to conduct out-of-sample R^2 evaluation using an independent test set from the same distribution. I constructed a unified preprocessing pipeline, used 5-fold cross-validation (CV) to compare the models, and then tuned the final model to achieve the best generalization effect.

2 Exploratory Data Analysis and Preprocessing

The training set contains 10,000 rows of data and 30 input features (excluding the target variable). These features consist of a mixture of numerical and categorical variables. I did not find any missing values (there are no columns with missing data; the maximum missing rate is approximately 0.00%), so no explicit filling processing is required.

For categorical variables, I used the "one-hot encoding" method. This expanded the feature space from 30 dimensions to 47 dimensions. Since the columns of one-hot encoding may differ in the training set and the test set, I aligned the test design matrix by re-indexing it with the same columns as the training set and filling the missing columns with zeros.

3 Model Selection

I compared multiple regression models using the same 5-fold cross-validation scheme (with random shuffling enabled and a fixed random seed), with the evaluation metric being R^2. Table 1 presents the average values and standard deviations for each fold. The performance of the linear model was inferior to that of the tree-based ensemble models, indicating the presence of non-linear effects and/or feature interactions. Gradient boosting achieved the strongest generalization ability, so I used it as the final model family.

4 Model Training and Hyperparameter Tuning

I chose `HistGradientBoostingRegressor` as the final predictor. Through the use of `RandomizedSearchCV` and combining 60 random configurations along with 5-fold cross-validation, the hyperparameters were optimized. The optimal configuration achieved an average R^2 value of **0.4748** in the cross-validation.

The tuned parameters were:

- `learning_rate = 0.01`, `max_iter = 2000`
- `max_depth = 2`, `max_leaf_nodes = 63`
- `min_samples_leaf = 120`, `max_bins = 128`
- `l2_regularization = 0.0`

This configuration (lower learning rate + many boosting iterations + shallow trees) can achieve a smoother fitting effect and reduce the risk of overfitting. The relatively larger minimum leaf sample size also regularizes the model by preventing the occurrence of extremely small leaf nodes, thereby improving the cross-validation performance on this dataset.

Table 1: 5-fold cross-validation performance (R^2).

Model	Mean R^2	Std.
Linear Regression	0.2711	0.0159
Ridge ($\alpha = 100$)	0.2855	0.0160
Random Forest	0.4417	0.0132
HistGradientBoosting (baseline params)	0.4632	0.0211
HistGradientBoosting (tuned)	0.4748	–

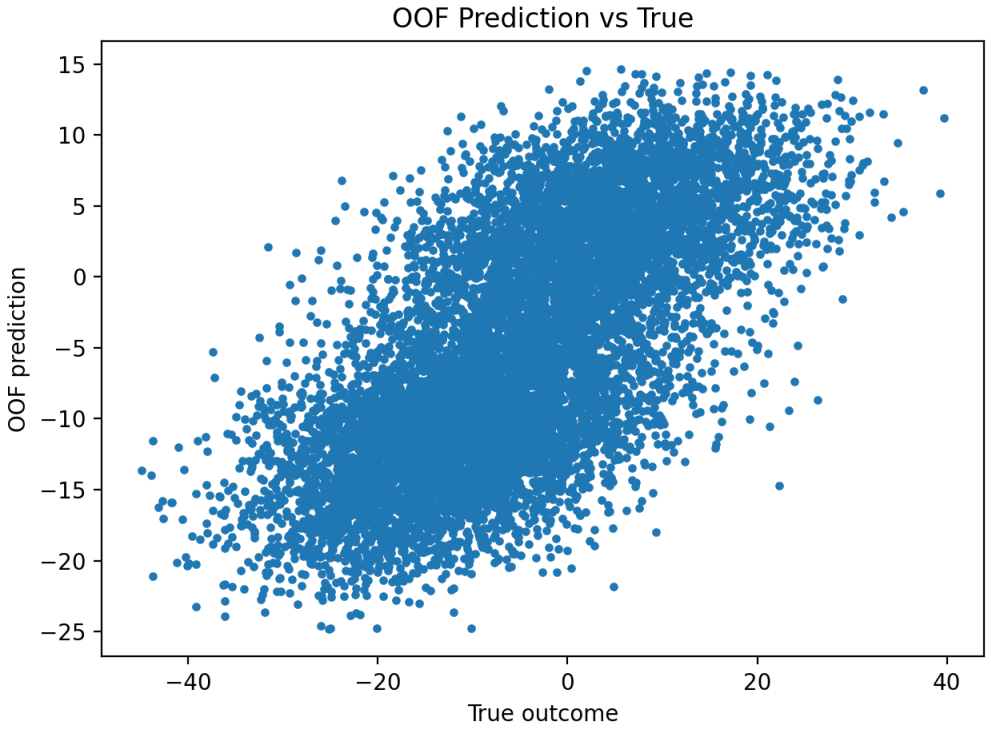

Figure 1: Out-of-fold predictions versus true outcomes for the tuned HistGradientBoostingRegressor.

5 Evaluation and Diagnostics

To test the generalization ability, I calculated the out-of-fold (OOF) predictions based on 5-fold cross-validation. Figure 1 shows a clear positive correlation between the OOF predictions and the true target values, indicating that the model can capture a large amount of information. The range of the predictions is narrower than that of the target values, suggesting a phenomenon of regression to the mean: extreme results are more difficult to predict and tend to be underestimated. The overall error is roughly concentrated around zero, with occasional large deviations, which is related to the presence of noise in the target values and the interaction that the available features fail to fully capture.

6 Code Supplement and Reproducibility

All experiments used a fixed random seed (`42`) and 5-fold cross-validation for consistent comparison. The repository includes scripts for model comparison, hyperparameter tuning, and generation of the final submission file named `CW1_submission_K23154082.csv`.

Code repository / appendix: `https://github.com/Irohas0416/MLF-CW1`.