Fine-Tuning Machine Learning Regressors for Optimal Hyper-parameter Selection

Mulubrhan Abebe Nerea
Msc Student in AI and Automation
Högskolan Väst/University west
Trollhättan, Sweden
mulubrhan-abebe.nerea@student.hv.se

***Abstract*—This report focuses on fine-tuning four regression models:Linear Regression, Decision Tree Regression, Random Forest Regression, and Support Vector Machine Regression (SVR) using California Housing dataset. The primary aim is to determine the most appropriate hyper-parameter combination for each model. We employed both Random Search and Grid Search methodologies along with 5-fold cross-validation to identify the best hyperparameters for each model. The Random Forest Regressor was the most effective model, achieving the lowest Root Mean Squared Error (RMSE) of 18,255.85 on the test set, illustrating its proficiency with handling complex data relationships via ensemble learning. The Decision Tree Regressor demonstrated notable enhancements with increased tree depth and more strict splitting criteria, resulting in an RMSE of 44,006.80. In contrast, Linear Regression demonstrated consistent performance, however its basic nature limited it, with an RMSE of 69,635.54. Concurrently, SVR encountered difficulties in encapsulating the dataset's intricacy, producing comparable RMSE values of 82,560.48 for Random Search and 82,560.50 for Grid Search. These findings highlight the essential importance of hyperparameter adjustment in improving model performance, especially for complex datasets.**

***Index Terms*—Fine-tune, Regressor, Grid, Random, RMSE**

code: https://github.com/Mulubrhan21/Fine-tune_Models

I. Introduction

Machine learning models often require fine-tuning to achieve optimal performance [1]. This report outlines the process of fine-tuning four regression algorithms using the California Housing dataset. The objective is to explore how Random Search and Grid Search can assist in identifying the best combination of hyperparameters for these models. By focusing on hyperparameter selection, this report aims to demonstrate how it impacts model performance and aids in selecting the most effective regressor for the dataset. Additionally, the fine-tuned models are evaluated on both the training and test datasets to assess their generalization performance, providing insights into how well each model predicts unseen data.

II. Fine-Tuning ML Models

Fine-tuning is a critical step in optimizing machine learning models. The process involves adjusting hyperparameters to improve the model's predictive performance. Hyperparameters, such as learning rate, number of trees in a Random Forest, or maximum depth of a decision tree, govern the training process. The goal is to identify optimal hyperparameter combinations that minimize error and enhance model accuracy [2] [3].

Two known methods for hyperparameter tuning are Grid Search and Random Search. Grid Search systematically evaluates a predefined subset of hyperparameter combinations by creating a grid of all possible settings and training the model on each combination. While this approach offers comprehensive coverage of the hyperparameter space, it can be computationally expensive, especially with numerous hyperparameters or lengthy training times. In contrast, Random Search randomly samples hyperparameter combinations from specified ranges, allowing for a more exploratory approach. This method is typically faster than Grid Search and can often yield better hyperparameter values due to its ability to explore more of the parameter space in less time. However, it may miss the best combination if insufficient samples are taken. The choice between these methods often depends on the size of the hyperparameter space and the available computational resources. After tuning, model performance is evaluated using techniques like cross-validation to ensure generalizability to unseen data [3].

III. Results and Analysis

In this section, we compare the performance and discuss the best hyperparameters discovered for each model during the fine-tuning process using both Grid Search and Random Search. Both training and test set results are evaluated to provide a comprehensive assessment of model generalization.

A. Linear Regression

Linear Regression does not have many tunable hyperparameters beyond regularization techniques (L1 or L2). In this case, L2 regularization was applied. The best training RMSE, found using both Random Search and Grid Search, was 68,827.71. For Ridge Regression, Random Search identified the optimal regularization parameter as $\alpha = 0.001$, while Grid Search found $\alpha = 0.01$ to be the best. Despite its stability across folds, Linear Regression was outperformed by more complex models, with its lack of significant hyperparameters to fine-tune limiting its ability to achieve better performance.

B. Decision Tree Regression

For the Decision Tree Regressor, the best training RMSE achieved through Random Search was 57,894.32,

using the hyperparameters: `min_samples_split = 10`, `min_samples_leaf = 10`, and `max_depth = 40`. Grid Search resulted in a slightly higher training RMSE of 60,967.10, with the best hyperparameters being: `max_depth = 10`, `min_samples_leaf = 2`, and `min_samples_split = 5`. Fine-tuning allowed the Decision Tree model to capture more complex patterns, especially with a deeper tree (`max_depth = 40`) in Random Search, which led to better generalization. However, limiting the depth in Grid Search hindered its performance.

C. Random Forest Regression

Random Forest Regression performed the best overall among all models. The best training RMSE achieved through Grid Search was 49,430.90, with the optimal hyperparameters being: `max_depth = None`, `max_features = 'sqrt'`, and `n_estimators = 200`. Random Search produced a similar training RMSE of 49,684.82, with the best hyperparameters: `n_estimators = 200`, `min_samples_split = 5`, `min_samples_leaf = 1`, `max_features = 'log2'`, and `max_depth = 30`. On the test set, Random Forest continued to excel, with Random Search yielding an RMSE of 24,563.57 and Grid Search achieving the lowest RMSE of 18,255.85. The ensemble learning technique of Random Forest, which combines multiple decision trees, enabled it to handle complex data interactions effectively, making it the best-performing model overall.

D. Support Vector Machine Regression (SVR)

SVR, using a linear kernel, was the weakest performer in this dataset. Both Random Search and Grid Search resulted in the same training RMSE of 81,705.94. The optimal hyperparameters identified by Random Search were `kernel = 'linear'`, `epsilon = 0.5`, and `C = 10`, while Grid Search found the best values to be `C = 10`, `epsilon = 0.2`, and `kernel = 'linear'`. Despite the fine-tuning efforts, SVR's linear kernel struggled to model the complexity of the dataset, suggesting that it may not be well-suited for this task.

Results Visualization

The efficiency of the models employing both Random Search and Grid Search methods is visually represented in Fig.1 for the training set results. The graphs indicate that Random Forest consistently achieved the lowest RMSE across both techniques, markedly beyond the other models. We evaluated the best model, Random Forest, on an unseen/test dataset, obtaining an RMSE of 24563.57 for Random Search and 18255.85 for Grid Search, respectively. Furthermore, we evaluate all models using the test dataset as shown in Fig. 2. Following Random Forest, Decision Tree indicated significant enhancement, especially with Random Search, while Linear Regression and SVR indicated more stable however higher RMSE values.

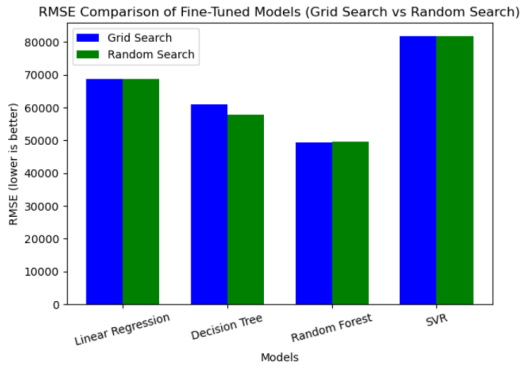

Fig. 1. Comparison of RMSE for fine-tuned models using Random Search and Grid Search methods.

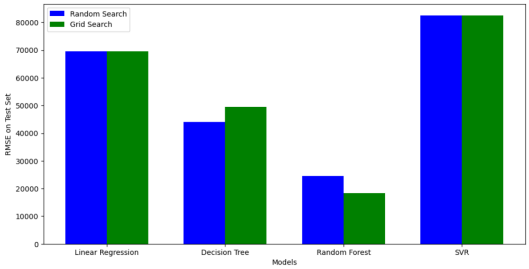

Fig. 2. Comparison of RMSE on Test Set(Random Search vs Grid Search).

IV. Conclusion

The fine-tuning process emphasized the critical role of hyperparameter selection in enhancing model performance. Among the four regressors tested, the Random Forest model, optimized using Grid Search, demonstrated the best performance, effectively managing complex data interactions. The Decision Tree model also showed notable improvements with deeper trees and stricter splitting criteria obtained via Random Search. In contrast, Linear Regression exhibited stable but relatively higher RMSE due to its simplicity and limited fine-tuning options. Finally, the Support Vector Machine with a linear kernel underperformed, as both fine-tuning methods yielded similar results, indicating that this model struggled to capture the dataset's complexity. Overall, this analysis highlights the value of Random Search and Grid Search for optimizing machine learning models and reinforces the importance of model selection based on dataset characteristics.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[2] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.

[3] S. Raschka, *Python Machine Learning: Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*. Birmingham, Mumbai: Packt Publishing, 2015.