Hyperparameter Optimization

Narges Helmisiasifariman

University of Wyoming

Introduction

This paper chooses a set of machine learning algorithms and their corresponding hyperparameters to improve the accuracy of various ML models. For this purpose, SVM, gradient boosting, decision tree, and random forest are selected. For each model, a set of hyperparameters is set to improve the model performance. For example, results display that hyperparameters like estimators, the depth of the tree, learning rate, the samples split, etc., enhance the predictive accuracy significantly. This work uses the randomized search technique to tune the hyperparameters of various ML models.

Dataset description

In this study, a dataset "winequality-red" contains twelve attributes of selected wine such as acidity measures, alcohol, sulfur dioxide content, quality measures, etc. (see Table 1). The dataset includes 1599 entries, and each column has no missing data. While most columns store decimal numbers (float 64), quality columns have integer numbers (int64). Figure 1 describes red wine attributes, stating that most quality is rated between 5 and 6. This study sets the wine quality column as the target variable, and other red wine attributes are considered feature variables.

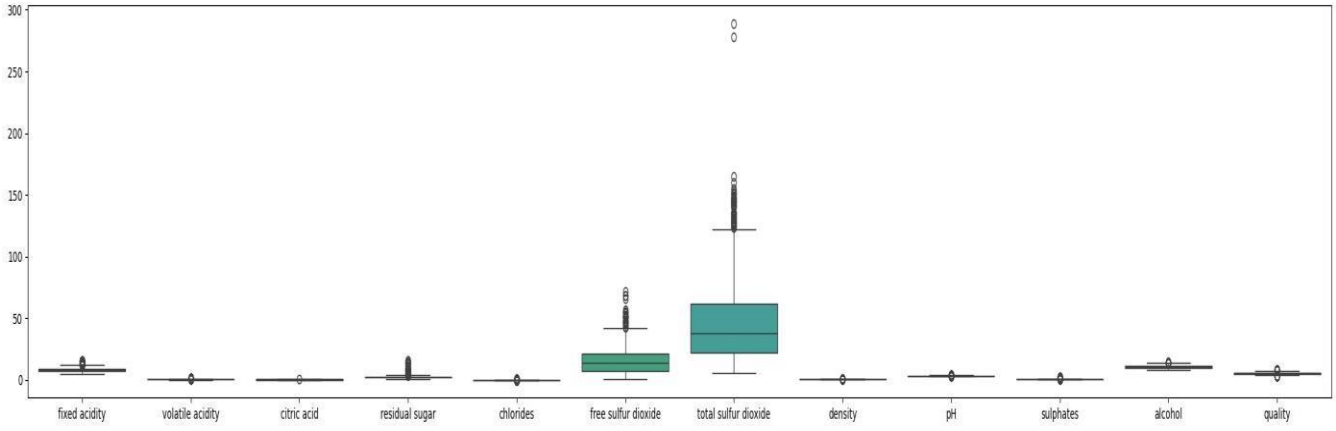

Figure 1 Attributes of selected wine and its rating quality

Table 1 Dataset description

Column	Attribute	Datatype
0	Fixed acidity	float64
1	Volatile acidity	float64
2	Citric acid	float64
3	Residual sugar	float64
4	Chlorides	float64
5	Free sulfur dioxide	float64
6	Total sulfur dioxide	float64
7	Density	float64
8	PH	float64
9	Sulphates	float64
10	Alcohol	float64
11	Quality	int64

Experimental setup

This section is crucial as it provides a detailed explanation of programming languages and libraries, data processing approaches, and machine learning algorithm selection, all of which are fundamental to data analysis and machine learning.

Programming languages and libraries

This work uses Python as a programming language for data analysis and machine learning. Also, the following libraries are used [1]:

- Pandas: Python data manipulation and analysis.
- NumPy supplies a library of high-level numerical computing that operates on arrays and matrices.
- Matplotlib and Seaborn: used for data visualization to analyze and achieve results
- Scikit-learn: a free and open-source machine-learning library for the Python programming language.

Data Processing Approach

The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation more suitable for downstream estimators. In general, many learning algorithms benefit from standardizing the data set.

1- **Standardization**

Standardization of datasets is a common requirement for many machine learning algorithms implemented in scikit-learn that remove mean and scale based on variance.

2- **Splitting data**

The data was split into training (80%) and testing (20%) data.

ML algorithm and hyperparameters

With a focus on the dataset, a select number of ML algorithms and their associated hyperparameters are being evaluated. The goal is to identify the most suitable machine learning algorithm and hyperparameter settings for this specific dataset.

To aim, I apply several machine learning models, including random forest, gradient boosting, decision tree, and SVM. The performance of each model was evaluated through a randomized search technique, which searches the best hyperparameters by considering accuracy. In this work, random forest hyperparameters include the number of estimators that describe how many trees are included in the forest. This work is defined as a range from 100 to 1000. The max depth parameter is defined as a range from 3 to 100 that helps prevent overfitting. Min sample split is a range from 2 to 100, representing the minimum number of samples required to split in an internal node. Min sample leaf parameter describes the minimum number of samples- for example, a range from 1 to 4 defined to be considered at a leaf node. In the gradient boosting model, the number of estimators specifies the number of trees or stages that are considered to be from 100 to 300. A Learning rate parameter, which is from 0.01 to 0.9, limits and contributes each tree to the overall model. A max depth parameter, considered from 3 to 10, defines the maximum depth of each tree to prevent complexity.

Three hyperparameters, max depth, min sample split, and min sample leaf, are defined for a decision tree model. A max depth parameter describes the tree's maximum depth, which helps prevent complexity. This work is considered to be from 3 to 20 in this report. A range of 3 to 20 as a minimum sample of split helps us to provide a minimum number of splits in the internal node. A range from 3 to 10 is defined to optimize the tree structure as a sample leaf. Indeed, this parameter displays the minimum samples necessary for smoothing the mode.

In an SVM model, parameter C represents a regularization parameter, and gamma is considered. C controls to get low training error as well as low testing error. Also, this parameter helps to minimize the complexity of the model. Gamma is a parameter that defines the impacts of each training set reach. The high value displays that the impact is close, while the low value means far. For this work, gamma values are sampled from a log-uniform distribution from at least 0.001 to at most 1.

RandomizedSearchCV

A randomized search is implemented in this study to tune the hyperparameters of various ML models. RandomizedSearchCV is a technique in the scikit-learn library that implements a "fit" and a "score" method [1]. Also, the parameters of the estimator used to apply these methods are optimized by cross-validated search over parameter settings. A fixed number of parameter settings are defined as follows. Param distribution defines a dictionary with parameter names as keys and distributions or lists of parameters to try. In this study, each ML model runs the corresponding hyperparameters. The number of iterations represents the number of parameter settings sampled, which is set to various values for each ML model. Also, scoring is defined as accuracy for model evaluation. In this work, n_jobs controls the number of parallel jobs that are used to run the computation. For this purpose, -1 means using all processors. A cv parameter that displays a cross-validation generator determines the number of folds. Also, the random state helps to control the randomness of the hyperparameter sampling. In addition, the error score parameter controls the errors during the model running, and error score =' raise' could stop the program and display the error.

Results

Figure 2 displays the accuracy of a few ML algorithms before and after using hyperparameter optimization. Also, Table 2 demonstrates the role of hyperparameter optimization in improving predictive accuracy for SVM, gradient Boosting, random forest, and decision tree models. Based on these results, SVM and random forest models have the highest accuracy after using hyperparameter optimization.

Table 2 ML models and associated improved accuracy through hyperparameter optimization

Model	Accuracy before Hyperparameter	Accuracy after hyperparameter	Improved (%)
SVM	0.6	0.65	8.3%
Gradient Boosting	0.64	0.67	4.6%
Random Forest	0.65	0.69	6.15%
Decision Tree	0.56	0.59	5.35%

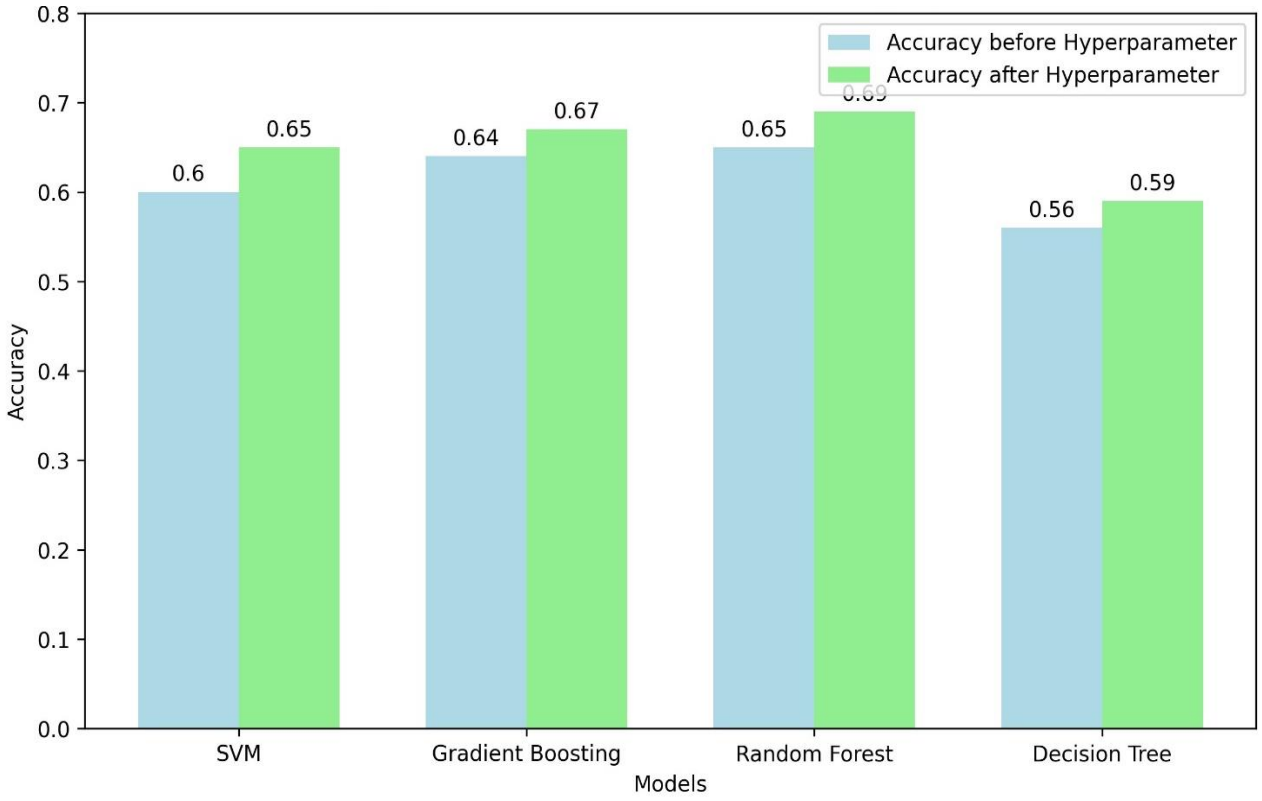

Figure 2 Accuracy for four ML models before and after hyperparameter optimization

For SVM, the highest accuracy (0.65) was achieved in the iteration of 17, C= 1.34, and gamma was approximately 0.7. Among four ML models, the random forest model has the highest accuracy before performing hyperparameter and then after using parameters like max depth of 24, min samples split of 3, min samples leaf of 1, and estimators of 443 at the iteration of 5, achieve an accuracy about 0.69. The gradient boosting model uses estimators of 120, a learning rate of 0.201105, and a max depth of 6 to get an improved accuracy of about %4.6 (see Table 2). The last ML model used is the decision tree, which performs three hyperparameters, which are max depth of 4, min samples split of 12, min samples leaf of 9, and iteration of 99, achieving a good and reasonable accuracy of about 0.59 (which is improved by about 5.35% compared to initial accuracy).

Reference

1. https://scikit-learn.org.