Hyperparameter Optimization and Interpretation in Machine Learning

Farid Soroush, PhD
soroushfarid@gmail.com

2023

Abstract

Machine learning has undergone tremendous advancements, paving the way for a myriad of applications across industries. In the midst of this progress, the significance of hyperparameter tuning and model evaluation can't be understated, as they play a critical role in achieving optimal model performance. This project delves into the realm of ML model optimization and evaluation, harnessing Bayesian Optimization, SHAP (SHapley Additive exPlanations), and traditional evaluation matrices. By focusing on a decision tree classifier, the study investigates the efficiency of various hyperparameter tuning methods, the interpretability of model decisions, and the robustness of performance metrics. Preliminary results suggest that Bayesian Optimization may offer advantages in efficiency over traditional tuning methods. Furthermore, SHAP values provide deeper insights into model decision-making, fostering better transparency and trust in ML applications.

1 Introduction

Machine learning's potential is undeniably vast, spanning from healthcare diagnostics to financial forecasting [1]. As the applicability of ML expands, the necessity to refine, understand, and trust these models becomes paramount [2]. Central to achieving these goals are hyperparameter tuning and model evaluation [3].

Hyperparameters, distinct from model parameters, are set before the learning process begins and significantly influence model performance [4]. Conventional methods like Grid Search and Random Search offer a systematic or random exploration of the search space, respectively [3]. However, they can be computationally intensive or miss optimal configurations [5]. Enter Bayesian Optimization: a probabilistic model-based optimization technique that has recently gained traction for its efficiency in hyperparameter tuning [6].

While achieving optimal performance is crucial, understanding why a model makes a particular decision is equally important. In many real-world scenarios, a black-box model, regardless of its accuracy, isn't sufficient [7]. Decision-makers require transparency to trust the model's predictions, especially in sensitive areas like healthcare or finance [8]. SHAP values, derived from game theory, have emerged as a potent tool to decipher the contributions of each feature towards a prediction [9].

Moreover, model evaluation extends beyond accuracy. In imbalanced datasets or multi-class scenarios, matrices like the confusion matrix, ROC curves, and others, offer more nuanced insights into model performance [10].

This project aims to bridge these pivotal areas, offering insights into:

1. The comparative efficiencies of hyperparameter tuning methods, focusing on Bayesian Optimization.

2. The interpretability of DecisionTreeClassifier decisions using SHAP values.

3. A comprehensive evaluation of model performance using traditional matrices.

The subsequent sections delve deep into methodologies, results, and implications of the findings.

2 Methods

2.1 Dataset

The Iris dataset, often referred to as Fisher's Iris dataset, is a classic dataset introduced by the British biologist and statistician Ronald A. Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems* as an example of discriminant analysis [17]. It has since become one of the most well-known and frequently used datasets for testing and illustrating various data analysis techniques, especially in the domain of classification.

The dataset comprises 150 observations from three species of Iris flowers: Iris setosa, Iris versicolor, and Iris virginica. Each species contributes 50 observations, making the dataset well-balanced. For each observation, four features were measured from the flowers: sepal length, sepal width, petal length, and petal width, all in centimeters. These measurements were used by Fisher to develop a linear discriminant model to distinguish the species from each other.

The simplicity, along with its clear separability between classes, makes the Iris dataset an excellent choice for introducing classification techniques. Over the years, it has been used in various fields of study, from machine learning to statistics, and remains a staple for demonstrating the basics of classification algorithms [18].

Given its historical significance and frequent appearance in literature, the Iris dataset serves as a benchmark in the evaluation of many machine learning algorithms. Its characteristics provide a controlled environment to compare different methods and techniques, making it a valuable asset for researchers and practitioners alike [19].

While the dataset is simple, it allows for the exploration of various aspects of data analysis, such as data visualization, preprocessing, feature extraction, and model evaluation. Furthermore, because of its ubiquity, results obtained on the Iris dataset can be easily compared across different studies and methodologies [20].

2.2 Decision Tree Classifier and Hyperparameter Tuning

Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

To optimize the performance of a Decision Tree, hyperparameter tuning is essential. Various techniques are employed, such as:

- **Grid Search:** An exhaustive search over a specified parameter grid. Grid Search builds a model on each parameter combination possible and evaluates each model.
- **Random Search:** Sets up a grid of hyperparameter values and selects random combinations to train the model and score.
- **Bayesian Optimization:** Uses probability to find the minimum of a function. This method, based on the Bayes theorem, aims to reduce the number of iterations needed to find optimal hyperparameters compared to grid or random search methods. The objective function for Bayesian Optimization focuses on maximizing the accuracy of a DecisionTreeClassifier, validated using 3-fold cross-validation on the training data. The parameters optimized include:
  - **criterion:** Determines the function to measure the quality of a split, either "gini" for Gini impurity or "entropy" for information gain. The search space ranges between [0, 1], where values below 0.5 map to "gini" and above map to "entropy".
  - **max_depth:** Represents the maximum depth of the tree, with [0, 50] as its range and 0 implying no restrictions.
  - **min_samples_split:** Minimum number of samples required to split an internal node, with a search space of [2, 10].
  - **min_samples_leaf:** Minimum samples required at a leaf node, ranging between [1, 4].

  Using the defined objective function, the Bayesian Optimization library iteratively searches for optimal hyperparameter values over 25 iterations, with the initial 5 being random explorations. Post optimization, the best parameters are employed to train a DecisionTreeClassifier, resulting in the 'model_bayesian'.

3 Results and Evaluation

The hyperparameter tuning process employed three optimization techniques: Random Search, Grid Search, and Bayesian Optimization. For each of these methods, the best hyperparameters and their corresponding cross-validation scores were identified. The details of these hyperparameters and the cross-validation scores are summarized in Table 2. Additionally, the optimal parameters obtained from the Bayesian Optimization method are presented in Table 4.

Performance evaluation of the models on the test set utilized several metrics and tools, which include the confusion matrix, classification reports, ROC curves, and SHAP values.

3.1 Confusion Matrix

Visualized as a heatmap, the confusion matrices in Figure 1 offer insights into the classifiers' accuracy for each optimization method. These matrices contrast Actual vs. Predicted values, which are crucial for gauging the effectiveness of the models.

3.2 Classification Report

The classification reports, shown in the table after Figure 4, provide a comprehensive summary of performance metrics. They encapsulate precision, recall, F1-score, and support for each class in the dataset. The overall accuracy, macro average, and weighted average are also included in these reports.

3.3 ROC Curve

Figure 2 presents the Receiver Operating Characteristic (ROC) curves. These curves graphically represent the classifiers' performance across varied classification thresholds, offering insights into the balance between the true positive rate and the false positive rate.

3.4 Feature Importance

Figure 3 illustrates the importance of features in the model, as determined by different search methods. The two sub-figures provide a comparative analysis of feature importance when using Random Search and Grid Search.

3.5 SHAP Value

SHAP (SHapley Additive exPlanations) values, as depicted in Figure 4, offer a comprehensive measure of feature importance across machine learning models. These values are essential for understanding individual feature contributions to model predictions. Three sub-figures present SHAP values corresponding to Random Search, Grid Search, and Bayesian Optimization methods.

3.6 Test Scores

The models' performances, when trained with the best hyperparameters obtained from both Random and Grid Search, are displayed in the table following Table 4. These scores give a direct understanding of how well the models generalize on unseen data.

For more detailed performance metrics, including specific figures, confusion matrices, classification reports, and ROC curve representations for each model, readers can refer to the subsequent sections.

4 Conclusion

Throughout this project, we delved deep into the nuances of hyperparameter optimization techniques for machine learning models, comparing traditional methods with advanced Bayesian optimization. The results obtained provided significant insights into the benefits and challenges associated with each method. Notably, while grid search and random search offered more straightforward implementation and broad hyperparameter exploration, Bayesian optimization showcased its power by efficiently narrowing down the search space and often reaching better results with fewer evaluations.

The integration of SHAP values further enriched our analysis by offering interpretable insights into the model's predictions. This not only demonstrated the importance of model interpretability in practical scenarios but also how SHAP can complement any model optimization strategy. The visualization tools, like confusion matrices and ROC curves, allowed for an easy comparison of model performance across different optimization techniques.

In conclusion, the balance between model accuracy, interpretability, and computational efficiency remains a pivotal concern in machine learning. While Bayesian optimization emerges as a potent tool for hyperparameter tuning, understanding the model's inner workings through tools like SHAP ensures that our models are not just statistically sound but also transparent and trustworthy.

References

[1] LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. *Nature*, 521(7553), 436-444, 2015.

[2] Doshi-Velez, F., & Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[3] Bergstra, J., & Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281-305, 2012.

[4] Goodfellow, I., Bengio, Y., & Courville, A. *Deep learning*. MIT press, 2016.

[5] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *arXiv preprint arXiv:1603.06560*, 2016.

[6] Snoek, J., Larochelle, H., & Adams, R. P. Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2951-2959, 2012.

[7] Ribeiro, M. T., Singh, S., & Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135-1144, 2016.

[8] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1721-1730, 2015.

[9] Lundberg, S. M., & Lee, S. I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765-4774, 2017.

[10] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874, 2006.

[11] Kearns, M., & Vazirani, U. V. *An introduction to computational learning theory.* MIT press, 1997.

[12] Chollet, F. *Deep Learning with Python.* Manning Publications Co., 2018.

[13] Domingos, P. A few useful things to know about machine learning. *Communications of the ACM*, 55(10), 78-87, 2012.

[14] Hutter, F., Kotthoff, L., & Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges.* Springer, 2019.

[15] Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232, 2001.

[16] Murphy, K. P. *Machine Learning: A Probabilistic Perspective.* MIT press, 2012.

[17] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, **7**(2), 179-188.

[18] Rao, C. R. (2009). Linear statistical inference and its applications. John Wiley & Sons.

[19] Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification. John Wiley & Sons.

[20] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

5 Appendix

5.1 Code

The Python code used in this project is available on GitHub (in Jupyter Notebook format) at the following link:

`https://github.com/FaridSoroush/HyperP-Opt-Interpretation-ML`

5.2 Figures

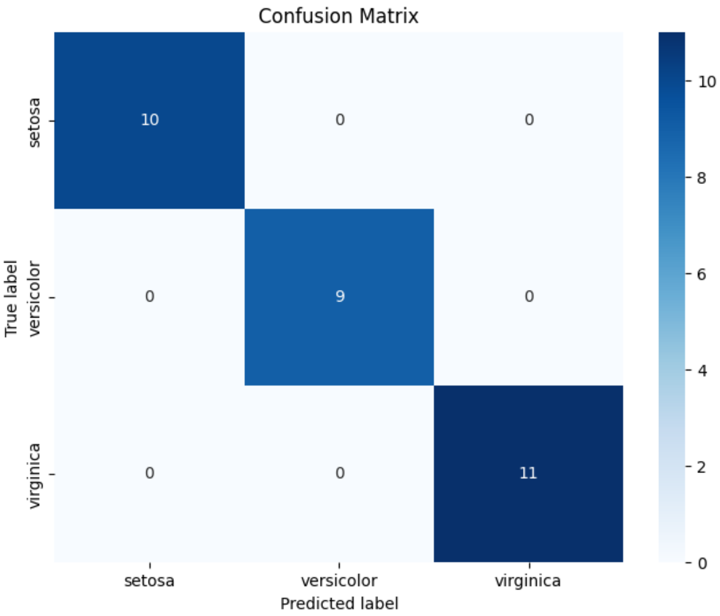


Figure 1: Confusion Matrix

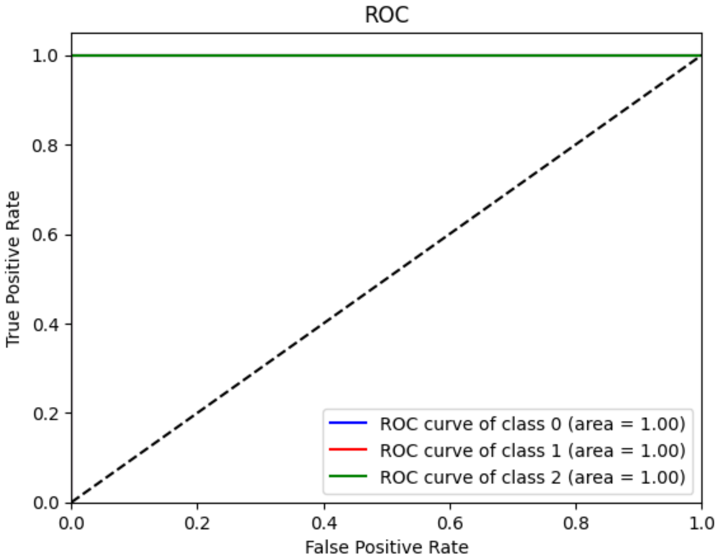
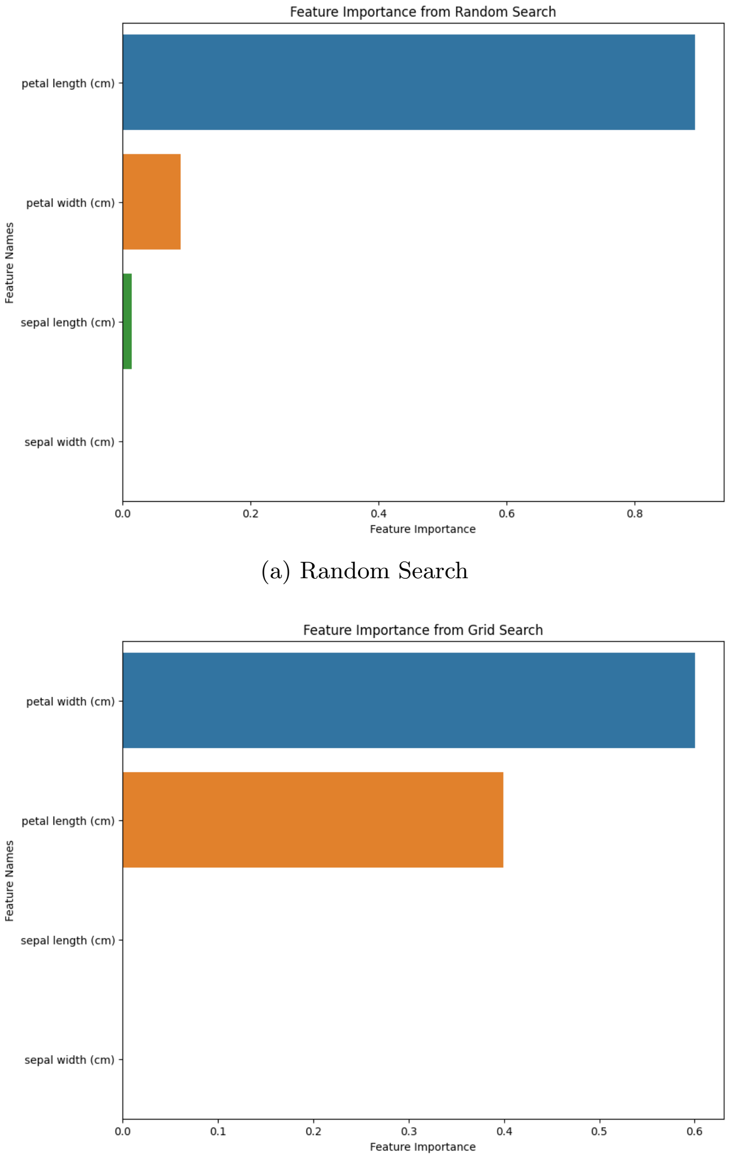

ROC
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve of class 0 (area = 1.00)
ROC curve of class 1 (area = 1.00)
ROC curve of class 2 (area = 1.00)

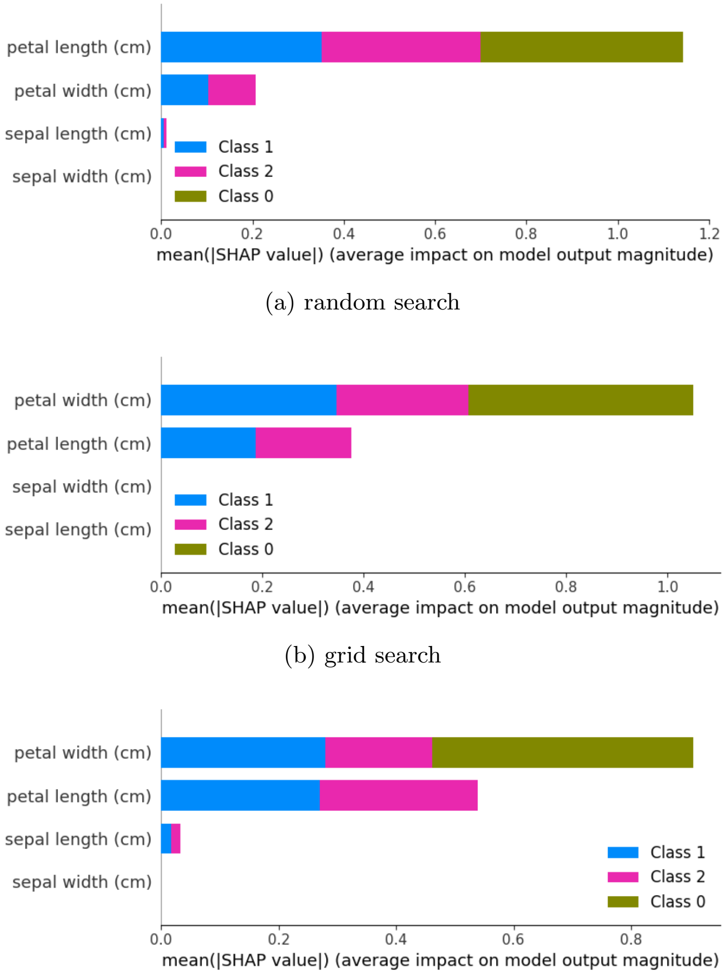
Figure 2: ROC



(a) Random Search

(b) Grid Search

Figure 3: Feature Importance in Different Search Methods



(a) random search

(b) grid search

(c) ayesian optimization

Figure 4: SHAP values for the first instance of different optimization models

5.3 Tables

Table 1: Classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
setosa	1.00	1.00	1.00	10
versicolor	1.00	1.00	1.00	9
virginica	1.00	1.00	1.00	11
Accuracy			1.00	30
Macro avg		1.00	1.00	30
Weighted avg		1.00	1.00	30

Search Method	Best Parameters	Best Cross-validation Score
Random Search	’max_depth’: 50, ’min_samples_split’: 10, ’min_samples_leaf’: 4, ’criterion’: ’gini’	0.950
Random Search	’max_depth’: None, ’min_samples_split’: 2, ’min_samples_leaf’: 1, ’criterion’: ’entropy’	0.950
Grid Search	’criterion’: ’gini’, ’max_depth’: None, ’min_samples_leaf’: 4, ’min_samples_split’: 2	0.950
Grid Search	’criterion’: ’gini’, ’max_depth’: None, ’min_samples_leaf’: 1, ’min_samples_split’: 10	0.950

Table 2: Random and Grid Search Best Parameters and Scores

Search Method	Test Score with Best Parameters (Random)	Test Score with Best Parameters (Grid)
Both Methods	1.0	1.0

Table 3: Test Scores for Best Parameters

Bayesian Parameter	Best Value
Criterion	0.737
Max Depth	7.792
Min Samples Leaf	2.000
Min Samples Split	9.024

Table 4: Bayesian Optimization Best Parameters