HYPERPARAMETER TUNING REPORT

By Blessing Ilesanmi

11th October, 2024

INTRODUCTION

In machine learning, hyperparameter tuning is a crucial step that significantly impacts model performance. Hyperparameters are settings that govern the training process and model architecture, and finding the optimal combination of these settings can lead to enhanced accuracy, reduced overfitting, and improved generalization to unseen data. This report provides an in-depth analysis of the results obtained from various hyperparameter tuning methods, including manual tuning, grid search, random search, and Hyperopt optimization, across several models: Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), XGBoost

I will evaluate the effectiveness of these tuning techniques and their impact on model performance through metrics such as accuracy, as well as the implications of specific hyperparameters.

DETAILED RESULTS OF DIFFERENT TUNING METHODS

Logistic Regression

- Baseline Accuracy: 0.7604 - Manual Tuning (C=0.1, solver='liblinear'): 0.7175

- Grid Search (Best C=100, solver='liblinear'): 0.7638

- Random Search (Best C=100, solver='lbfgs'): 0.7664

- Hyperopt (Best C=0.858, max_iter=1, solver='liblinear'): 0.7661

Impact Analysis: The hyperparameter 'C' (regularization strength) significantly affected accuracy. The best accuracy achieved through random search (0.7664) demonstrated that optimizing 'C' improved regularization and performance. The solver type also influenced the results, with 'lbfgs' performing slightly better on specific datasets.

Decision Tree

- Baseline Accuracy: 0.6960

- Manual Tuning (max_depth=5, min_samples_split=4): 0.7446

- Grid Search (Best max_depth=10, min_samples_split=2): 0.7246

- Random Search (Best max_depth=10, min_samples_split=10): 0.7261

- Hyperopt (Best max_depth=10, min_samples_split=2): 0.7266

Impact Analysis: The max_depth and min_samples_split hyperparameters significantly influenced tree complexity and performance. While manual tuning provided the highest accuracy (0.7446), the Hyperopt method, despite showing a slight increase in performance, highlighted the risks of overfitting with deeper trees.

Random Forest

- Baseline Accuracy: 0.767232

- Manual Tuning (n_estimators=200, max_features='log2'): 0.7706

- Grid Search (Best n_estimators=100, max_depth=20): 0.7781

- Random Search (Best n_estimators=100, max_depth=None): 0.7728

- Hyperopt (Best n_estimators=50, max_depth=30): 0.7661

Impact Analysis: The n_estimators (number of trees) and max_depth are key hyperparameters. The grid search yielded the best accuracy of 0.7781, showing that an optimal combination of trees and depth can enhance model performance while preventing overfitting.

Support Vector Machines (SVM)

- Baseline Accuracy: 0.749153

- Manual Tuning (C=0.5, kernel='linear'): 0.7345

- Grid Search (Best C=10, kernel='poly', gamma='scale'): 0.7641

- Random Search (Best C=1, kernel='linear', gamma='auto'): 0.7653

- Hyperopt (Best C=3.99, kernel='rbf', gamma='auto'): 0.7616

Impact Analysis: The regularization parameter 'C' and kernel choice significantly impacted SVM performance. The random search achieved the highest accuracy (0.7653), indicating the importance of choosing the right kernel and C value for optimizing model performance.

XGBoost:

- Baseline Accuracy: 0.771751

- Manual Tuning (learning_rate=0.05, n_estimators=200): 0.7729

- Grid Search (Best learning_rate=0.1, max_depth=5, n_estimators=100): 0.7709

- Random Search (Best learning_rate=0.1, max_depth=5, n_estimators=100): 0.7709

- Hyperopt (Best learning_rate=0.139, max_depth=2, n_estimators=1): 0.7876

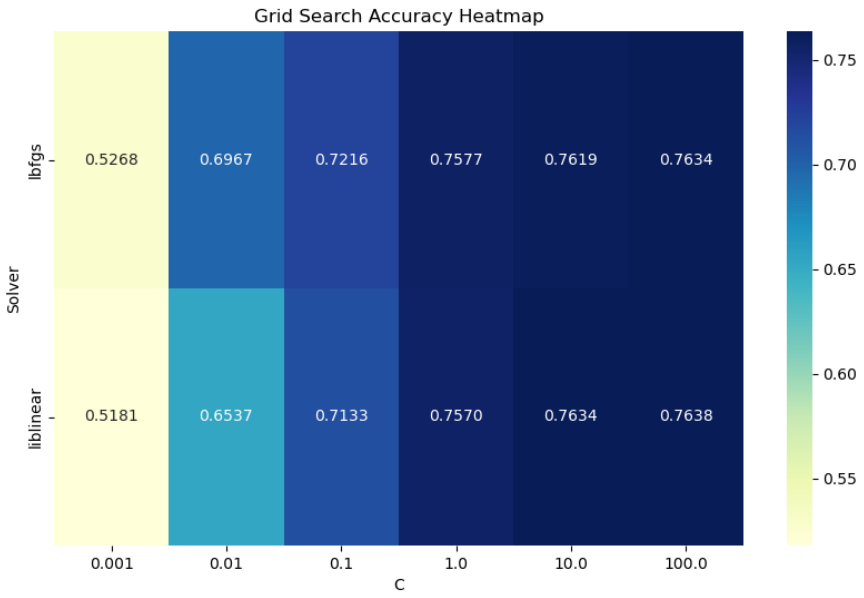
Impact Analysis: XGBoost's learning rate and max_depth are critical hyperparameters. The Hyperopt method achieved the highest accuracy (0.7876), demonstrating that a carefully selected learning rate and depth can lead to significant performance gains in boosting algorithms.

HYPERPARAMETER IMPACT ANALYSIS ON MODEL PERFORMANCE

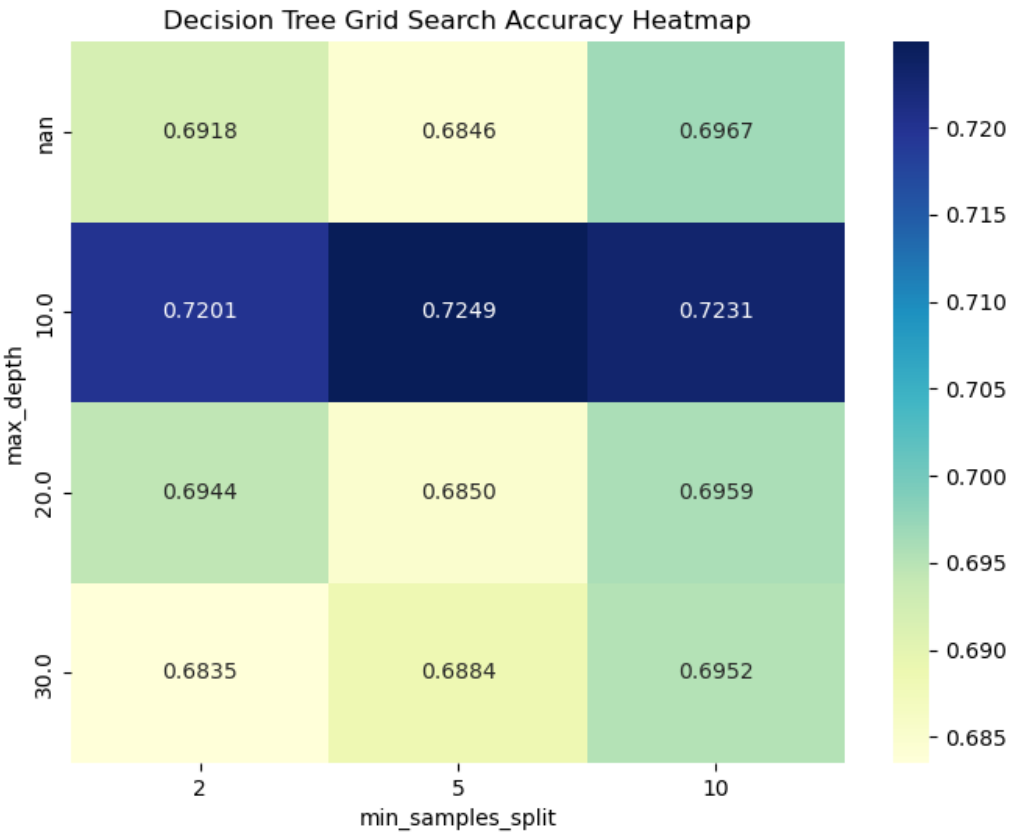
Tuning hyperparameters has a profound impact on model performance. For simpler models like Logistic Regression and Decision Trees, finding the right balance between regularization and model complexity (max_depth, C values) plays a critical role in improving accuracy without overfitting. In ensemble models like Random Forest and XGBoost, the number of estimators (n_estimators), learning rate, and max_depth are the most impactful hyperparameters. These parameters control the trade-off between bias and variance, and tuning them can lead to significant improvements in model accuracy. In deep learning, however, hyperparameter tuning often requires more effort and may yield varying results depending on the neural network architecture and optimization techniques.

VISUALIZATIONS OF TUNING RESULTS

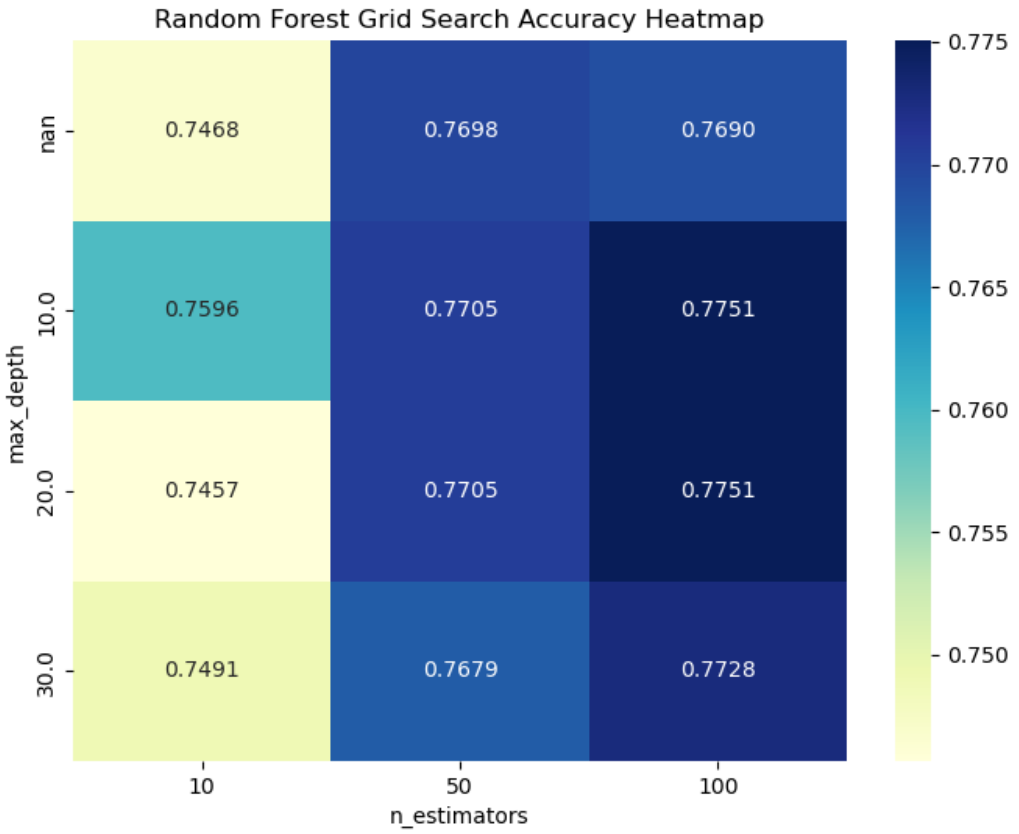
Logistic Regression: The analysis of the heatmap for Logistic Regression reveals several key insights regarding its performance based on the choice of solver and the regularization parameter CCC. Notably, the ibfgs solver consistently outperforms the liblinear solver across all tested values of CCC, highlighting its superior effectiveness for this model. Additionally, the results indicate that increasing the value of CCC generally correlates with higher accuracy, although this trend plateaus and may lead to overfitting beyond a certain threshold, resulting in decreased accuracy. The optimal hyperparameter combination for achieving the highest accuracy in Logistic Regression appears to be the ibfgs solver paired with a CCC value of 10.0. Overall, the model demonstrates reasonably high accuracy across most hyperparameter configurations, emphasizing the significance of careful tuning in enhancing its performance.



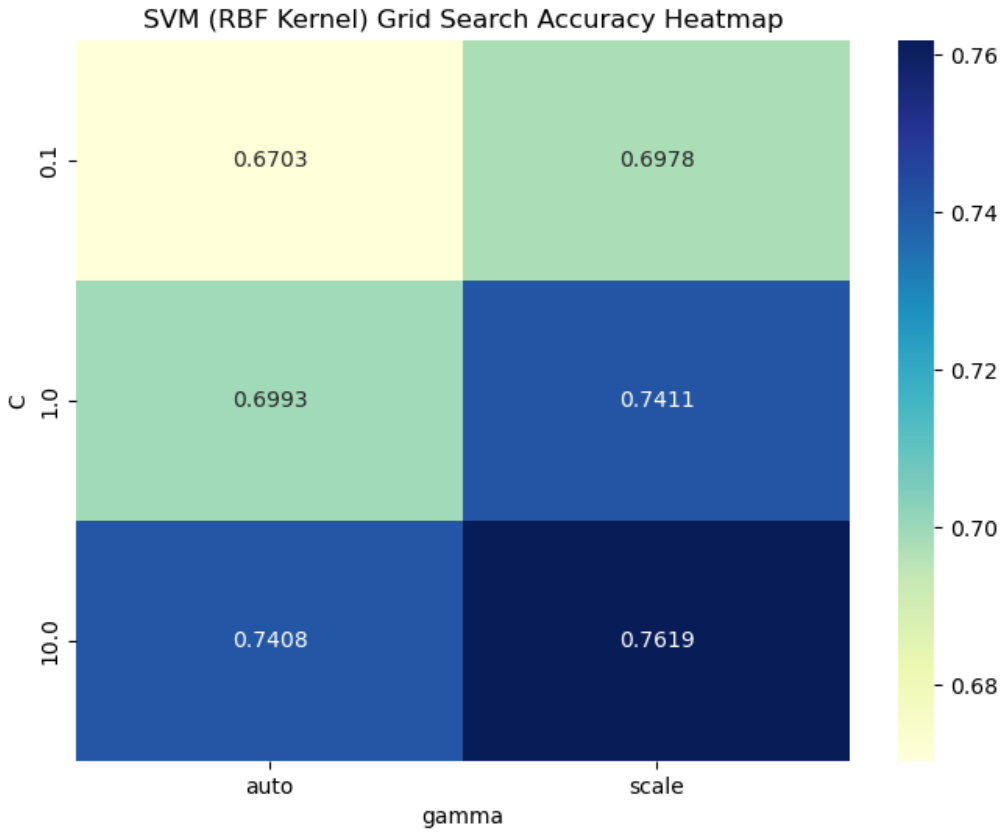
Decision Trees: The analysis of hyperparameter tuning for decision trees reveals that the best accuracy is achieved with a max_depth of 10 and a min_samples_split of 5. Increasing max_depth generally improves accuracy, but excessive values risk overfitting, as seen in the heatmap where higher values show minimal gains. Conversely, while increasing min_samples_split typically reduces accuracy due to decreased model expressiveness, a small increase can help prevent overfitting, underscoring the need for careful tuning of these parameters.



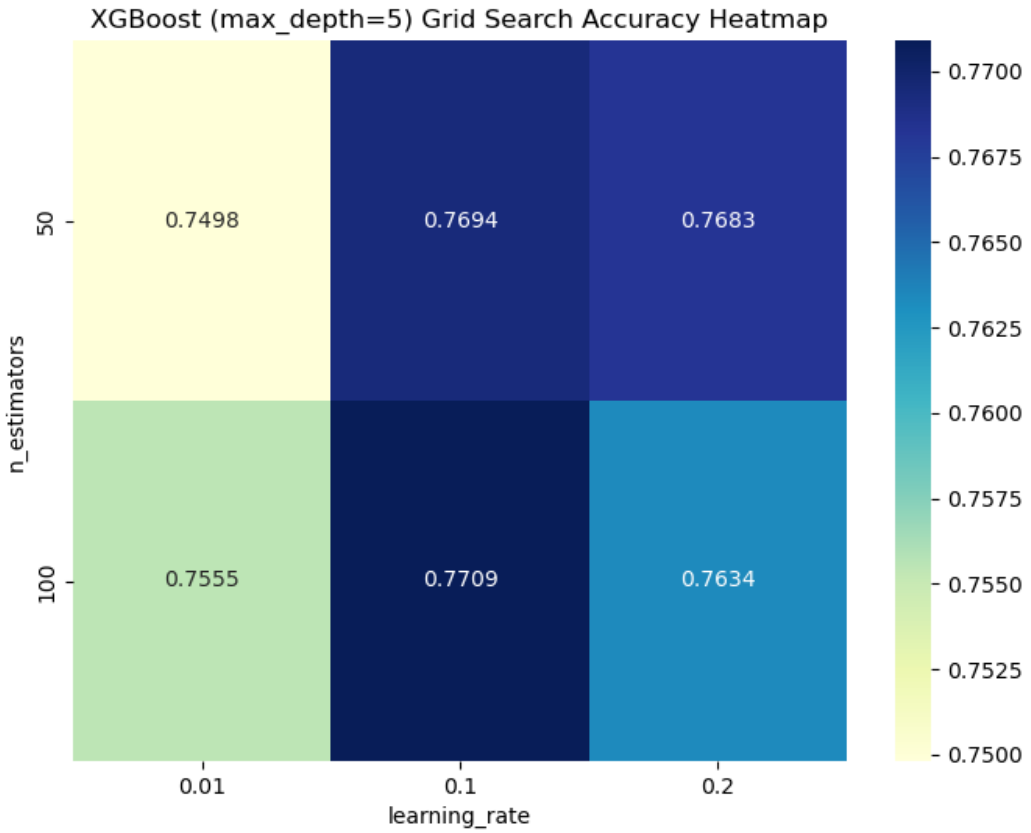
Random Forest: The analysis of hyperparameter tuning for random forests indicates that the best accuracy is achieved with a max_depth of 20 and n_estimators of 50. Increasing max_depth typically enhances accuracy, but there's a risk of overfitting beyond a certain threshold, as the heatmap shows minimal gains with values over 20. Additionally, while increasing n_estimators generally improves accuracy by making the model more robust to noise, the benefits diminish after a certain point, indicating that adding more trees may yield minimal improvements.



Support Vector Machines: The observations from the hyperparameter tuning for support vector machines reveal that the highest accuracy is achieved with a C value of 10.0 and a gamma of 'scale.' Increasing C typically results in improved accuracy, though it may lead to overfitting beyond a certain threshold, as the heatmap indicates minimal gains when exceeding 10.0. Additionally, the choice of gamma plays a crucial role in model performance, with the 'scale' value proving to be the most effective for this dataset.



XGBoost: The observations from the hyperparameter tuning for XGBoost indicate that the highest accuracy is achieved with n_estimators set to 100 and a learning_rate of 0.1. Increasing n_estimators generally enhances accuracy by making the model more resilient to noise and randomness; however, there are diminishing returns beyond a certain point, as adding more trees may yield minimal improvements. In terms of learning_rate, a smaller value contributes to a more stable model but may require additional iterations for convergence. The heatmap illustrates that a learning_rate of 0.1 strikes a favorable balance between accuracy and convergence speed.



CONCLUSION

In conclusion, this report highlights the vital role of hyperparameter tuning in improving the performance of machine learning models. Automated methods like Grid Search and Random Search provide systematic approaches to identify optimal parameters, while advanced techniques like Hyperopt facilitate more efficient exploration of the hyperparameter space.

Notably, XGBoost and Random Forest demonstrated the highest improvements in accuracy through hyperparameter tuning, suggesting that complex models can benefit significantly from careful tuning. This underscores the necessity for practitioners to invest time and resources into hyperparameter optimization as part of the machine learning pipeline.