Decision Trees: Mastering the Bias-Variance Trade-Off with max_depth

Author: Gopal Reddy Marella
Date: 11 December 2025

Abstract

The current tutorial takes an actionable in-depth analysis of the most important hyperparameter of Decision Trees (DTs), max depth. We investigate the basic machine learning idea of the Bias-Variance Trade-Off through the visualization of the effect of depth on the decision boundary. This is aimed at providing practitioners with the knowledge required to regulate model complexity in order to achieve maximum generalization in practice.

GitHub Repository Link

The full source code, including the runnable Jupyter Notebook and all generated figures, is available here: https://github.com/gopalreddymarella402/decision-tree-tutorial

1. The Core Idea: What is a Decision Tree?

A Decision Tree (DT) is a non-parametric supervised learning algorithm that is applied for classification and regression. It works by dividing the data recursively on the basis of features to form it a flow chart-like data structure. The DT greatest strength (its ability to create complex, axis aligned decision boundaries) is actually its greatest weakness when its growth is left unchecked.

DTs, by default, will try to get 100% accuracy for the training data by splitting and splitting and splitting, until there are only a single class in all the leaf nodes (a "pure" node). This process, which is called fully growing the tree, always leads to overfitting on the noise.

2. Our Focus: Controlling Complexity with max_depth

The max_depth hyperparameter is an explicit limit on the number of levels that the tree is allowed to grow below the root node. It is the single most effective tool for overcoming the high variance of unconstrained Decision Trees.

- **Small max_depth (e.g., 2-4):** High Bias model. The model is too simple (underfitted) and doesn't capture the true underlying pattern and hence we get the accuracy on both training and testing data to be low.
- **Large/Unconstrained max_depth (e.g., None):** The model is a High Variance model. The model is too complicated (overfitted) binds the noise of the training data. Training accuracy is high (close to 100%), but test accuracy is low.

- **Optimal max_depth:** This sweet spot is where the model makes the least overall mistakes - in terms of a balancing act - between bias and variance, and is able to perform well not just in the training data, but in new, unseen data.

3. The Experiment: Visualizing the Trade-Off

We took the great synthetic make_moons example, which outlines a unique non-linear classification issue, for the purpose of showing the relationship of the different values of max_depth in terms of the decision boundary they generate. The accompanying notebook have three different models, which train a three spectrum of effectiveness.

3.1. Visualizing the Decision Boundaries

Scenario A: Underfitting (High Bias) - max_depth=2

Model trained with max_depth=2. The boundary is too Simple where two complex shapes are tried to be divided with minimum splits. Both training and testing performance is bad, which indicates that the model is fundamentally flawed for this problem.

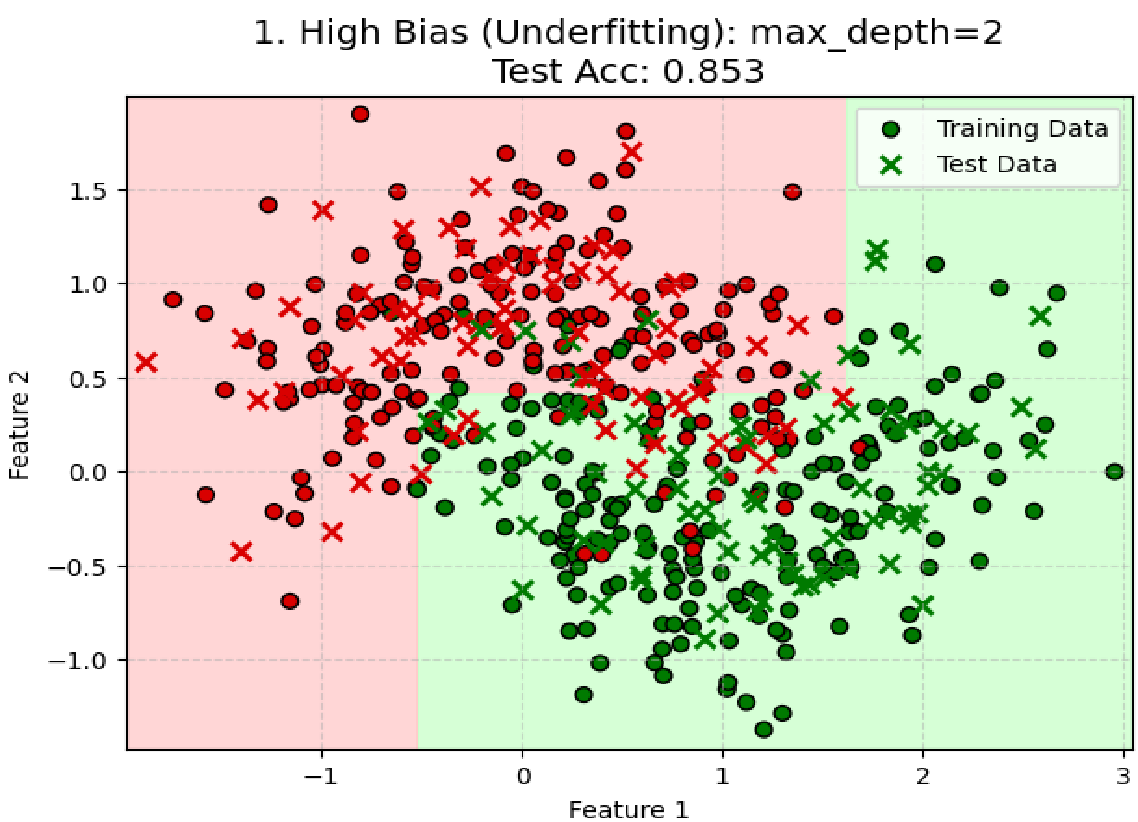
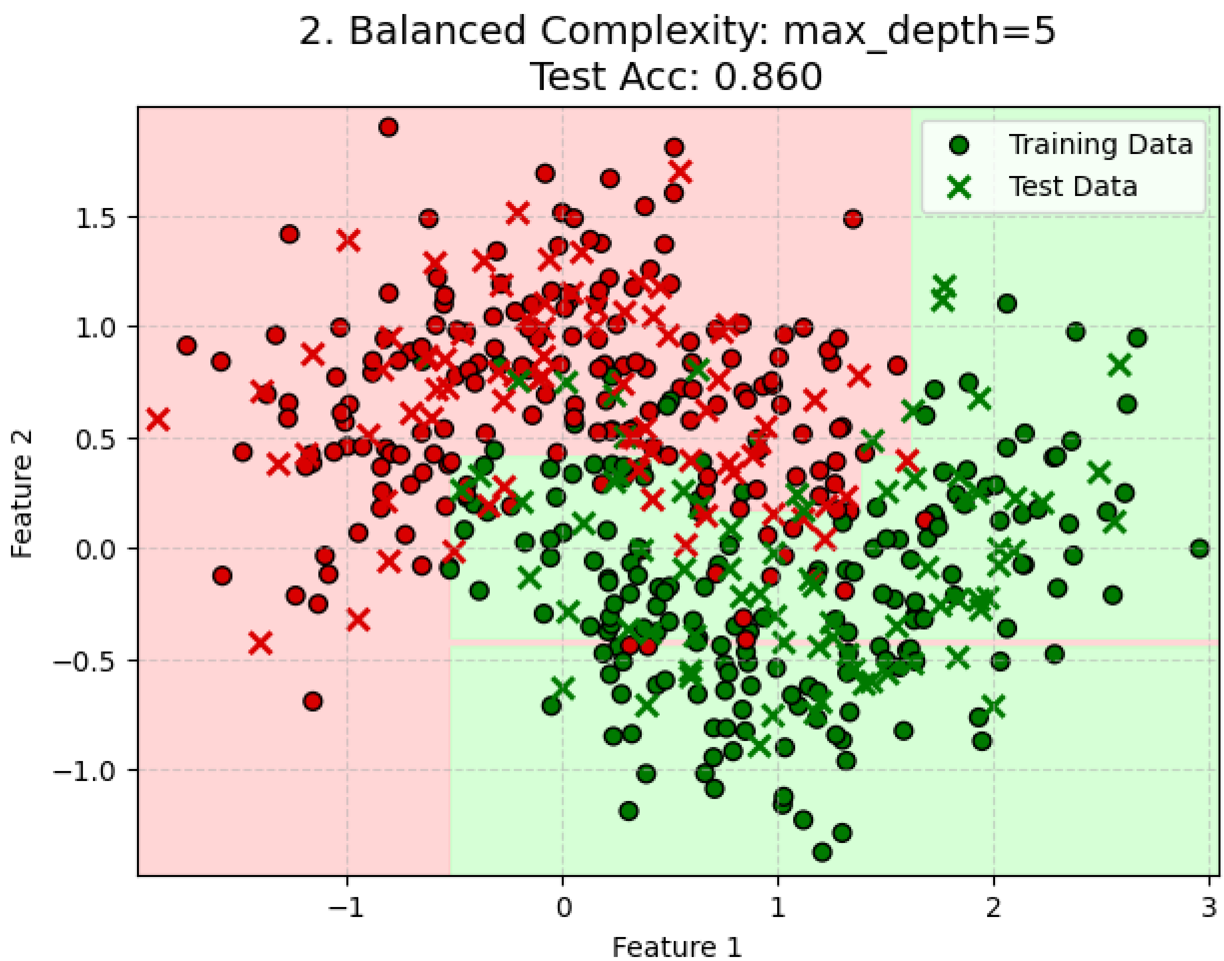


Figure 1: High Bias Decision Boundary. The model fails to capture the true non-linear structure of the data, resulting in poor performance.

- **Figure File:** figures/figure_1_high_bias.png

Scenario B: Balanced (Optimal) - max_depth=5

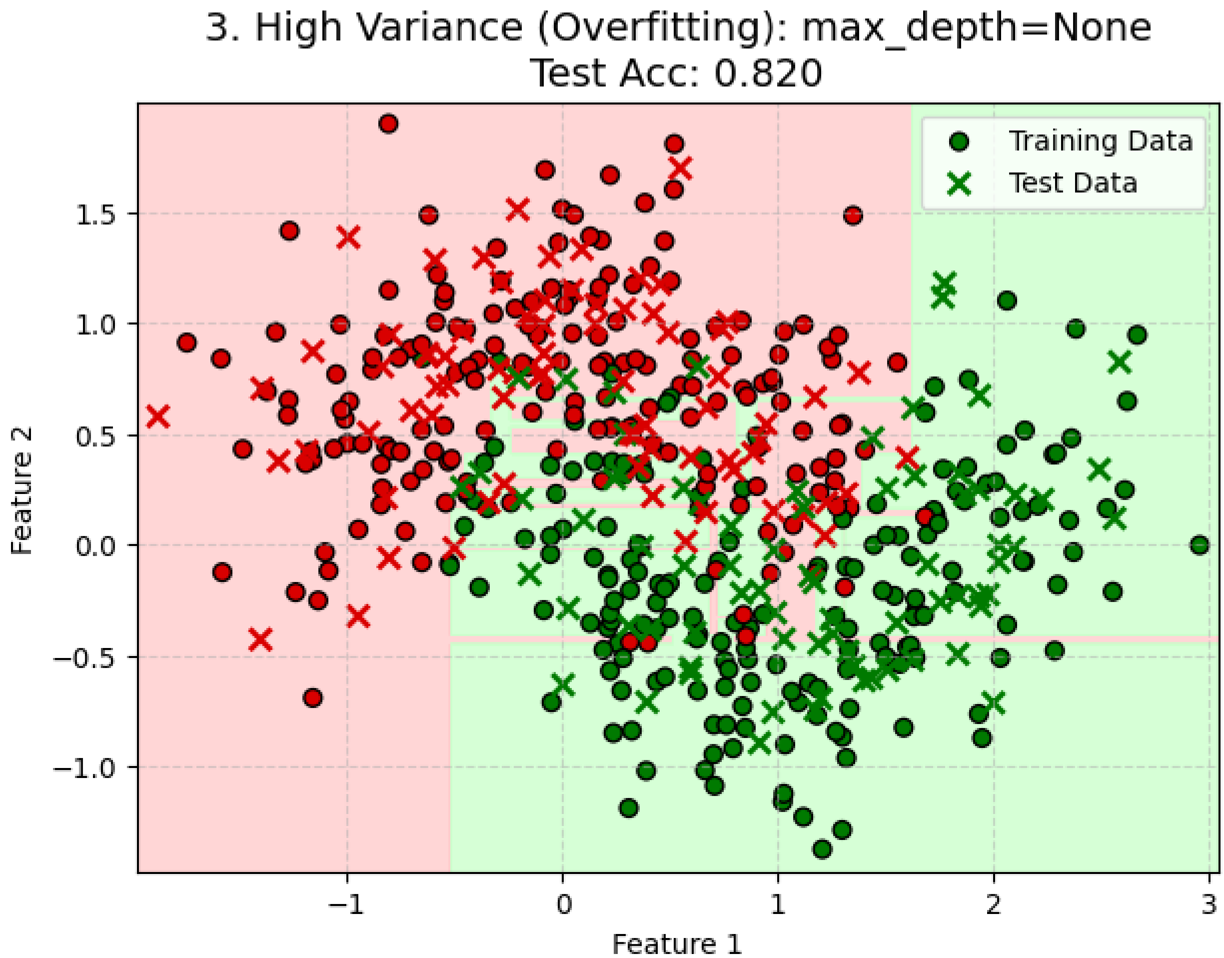
Model trained with max_depth=5. The boundary captures the "moon" shape perfectly without too complicated boundaries. The accuracy of testing is maximized here since the model has found that appropriate degree of complexity in order to generalize.



- **Figure 2:** Optimal Decision Boundary. The boundary is smooth, accurately capturing the data pattern and yielding the best generalization score.
- **Figure File:** figures/figure_2_optimal.png

Scenario C: Overfitting (High Variance) - max_depth=None

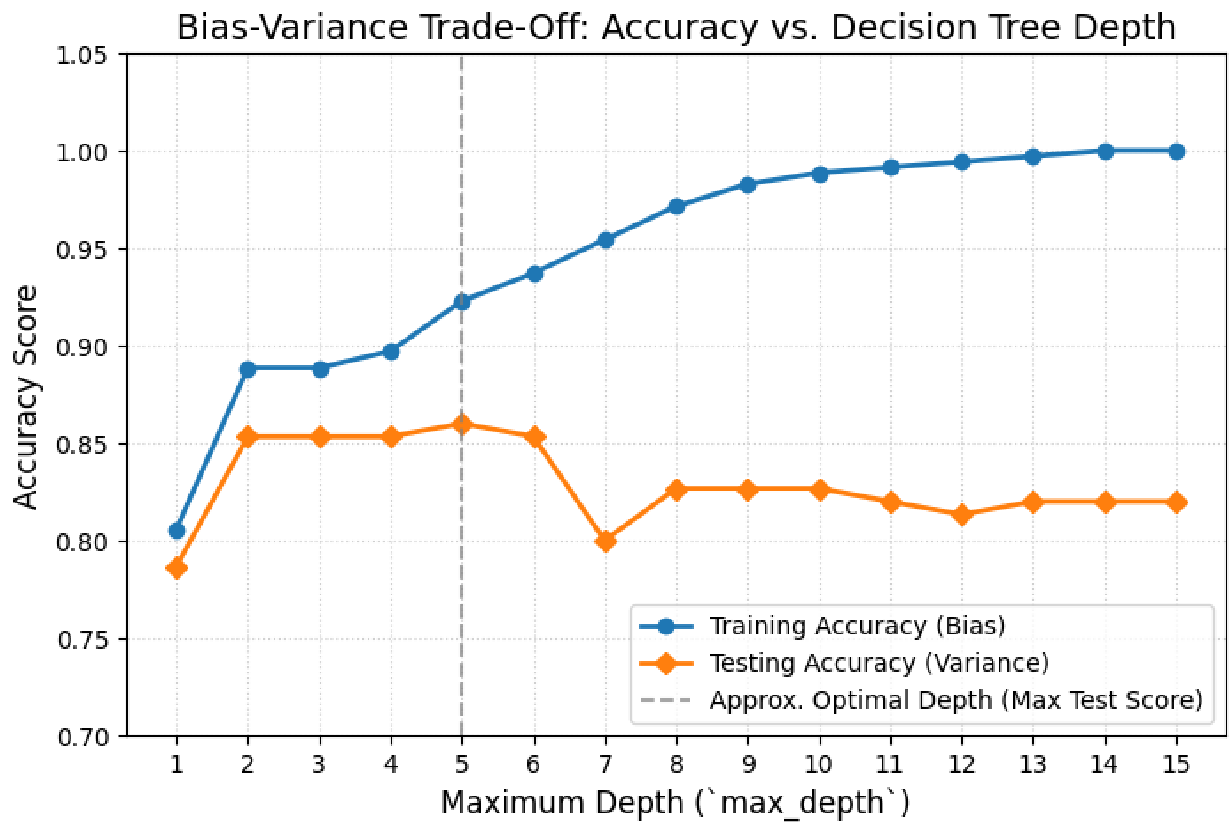
Model trained with max_depth=None (unconstrained). The border is left really ragged with small islands being built on and around noisy training points. The accuracy of the training is close to 1.0, but the accuracy of the testing is very low, as the model is just remembering noisy data, and not learning generalized patterns.



- Figure 3: High Variance Decision Boundary. The boundary aggressively fits noisy training points, leading to high training accuracy but poor test performance.
- **Figure File:** figures/figure_3_high_variance.png

3.2. The Trade-Off Curve

The most informative plot graph is the accuracy scores versus the complexity (depth). The most final evidence of the bias-variance trade-off is the visual divergence.



- Figure 4: The Bias-Variance Trade-Off Curve. The Training Accuracy continuously rises, while the Testing Accuracy peaks (around depth 5) and then falls as variance increases.
- **Figure File:** figures/figure_4_trade_off_curve.png

The accompanying notebook was subjected to a sweeping experiment between depths 1 and 15. The curve obtained shows that the accuracy of the testing (generalization power) is the highest, and then the model starts to overfit the training data.

4. Practical Implications and Ethical AI

The most important advice on the proper use of Decision Trees is that they should never be relying on default settings. Always cross-validate or have separate validation set and run through a set of max depths and pick out the one that performs best on the unknown data.

4.1. Ethical Considerations

Although it is typically hailed as a benefit of DTs, a deep overfitting tree loses the benefit of explainability. A model that has been overfit to memorize training noise is not reliable and they may give unfair or biased results when used on new data particularly when the training noise has an accidental correlation with the data being protected. It is not only a performance metric then, but also an essential measure towards performance to enhance **robustness and ethical reliability**.

5. Conclusion

The hyperparameter of the max depth is the complexity controller of the Decision Trees. Our manual control of it moves the model up and down the bias-variance curve, and this enables us to identify the balance in which the model is likely to generalize most effectively. This is the only parameter that should be mastered in order to take advantage of the power of the Decision Trees without falling prey to the fact that they are high-variance.

References and Resources

- Scikit-learn Documentation: Decision Tree Classifier. Source for implementation details and parameter definitions. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. Theoretical basis for bias-variance.
- Towards Data Science Article: Understanding the Bias-Variance Tradeoff in Machine Learning. Conceptual resource for intuition.