Understanding Decision Trees: How Impurity and Pruning Shape Model Performance (with the Palmer Penguins Dataset)

Decision Trees are appealing models because they closely resemble human decision-making: a sequence of questions leading to a final conclusion. In this tutorial, explore how Decision Trees behave on the Palmer Penguins dataset, focusing on three concepts that strongly influence their performance - impurity measures, model depth, and cost-complexity pruning. All results and figures come directly from the accompanying notebook.

The dataset contains bill length, bill depth, flipper length and body mass measurements for three penguin species. After removing rows with missing values, 342 complete samples remained, which were encoded and split into an 80/20 train–test split using stratification. Before fitting any model, examine the data visually.

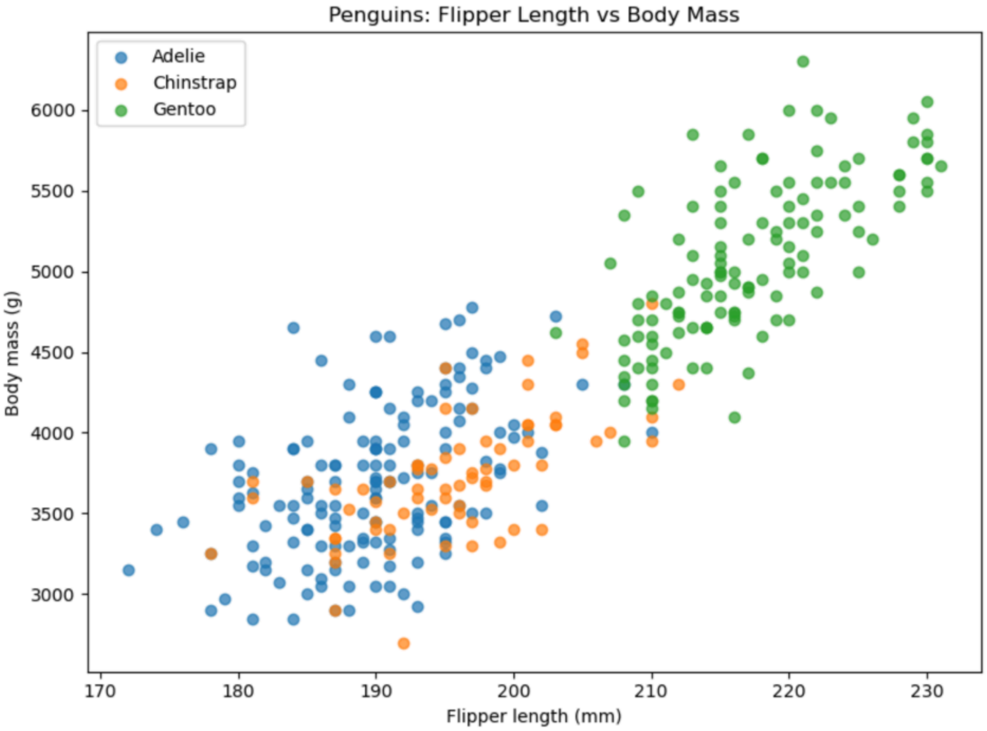


Figure 1: Flipper Length vs Body Mass

Scatter plot of flipper length against body mass, coloured by species is shown using Fig. 1. We can easily read the plot according to the instructions received. Gentoo penguins form a cluster having long flippers and greater mass. Furthermore, Adelie and Chinstrap penguins can be noticed at different places which overlap with each other to an extent only. This organic formation looks like a Decision Tree ought to create accurate boundaries using simple numbers.

Baseline Decision Tree using Gini Impurity

I began my training with a Decision Tree trained using Gini impurity. The parameters chose were without restrictions or pruning. Thus, the algorithm kept on splitting, till class purity could not be improved. The fully grown model attained perfect accuracy on the training and test sets – an incredible result (but not surprising on this data set which has such good natural separation between species). To get a better idea of how this model reached its decisions, visualised the whole structure of the tree which is seen in Figure 2.

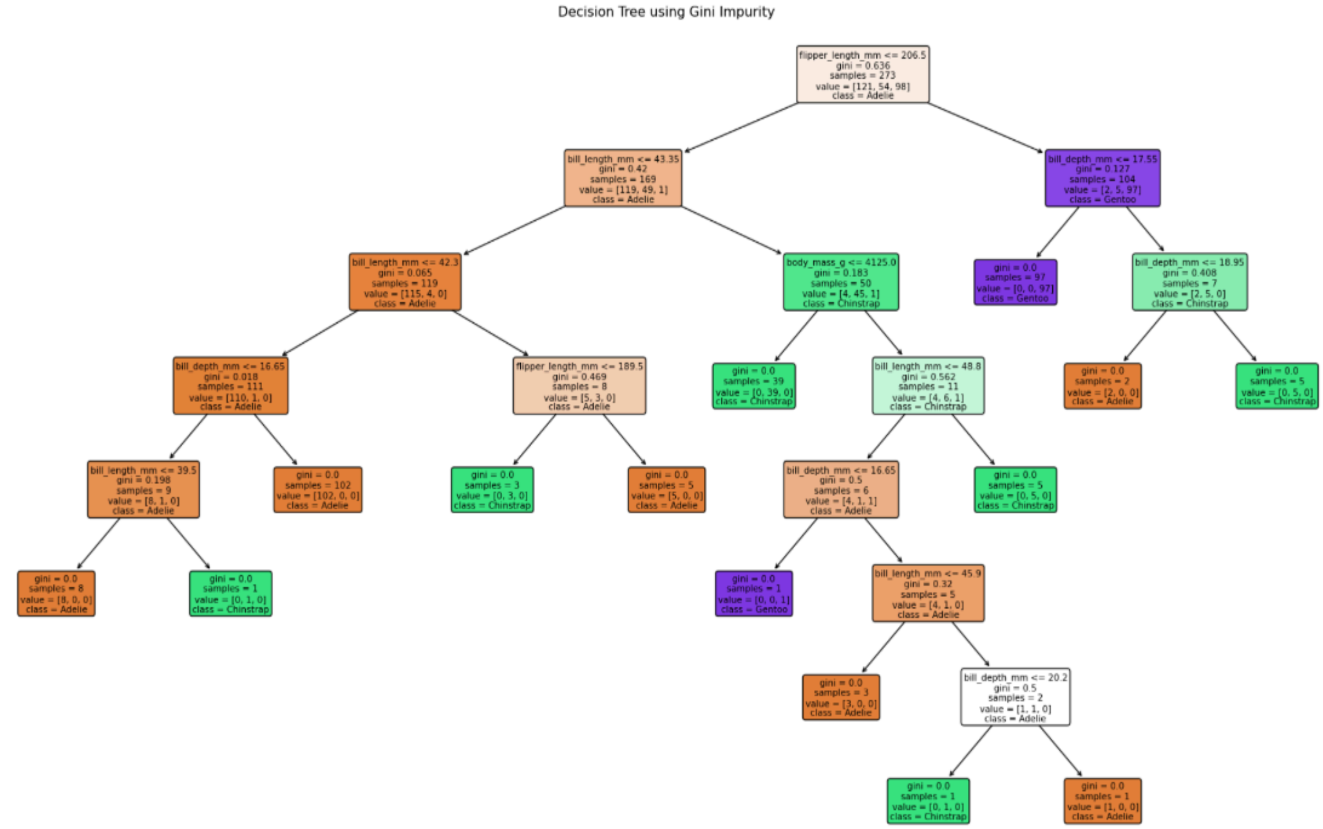


Figure 2 : Full Gini Tree

The first split on the tree is on flipper length, separating Gentoo penguins from the other two. According to the exploratory scatter plot, Gentoo penguins were grouped together in a region with significantly longer flippers. Because this feature provides an obvious clear boundary, it is classified by the algorithm to be the best split.

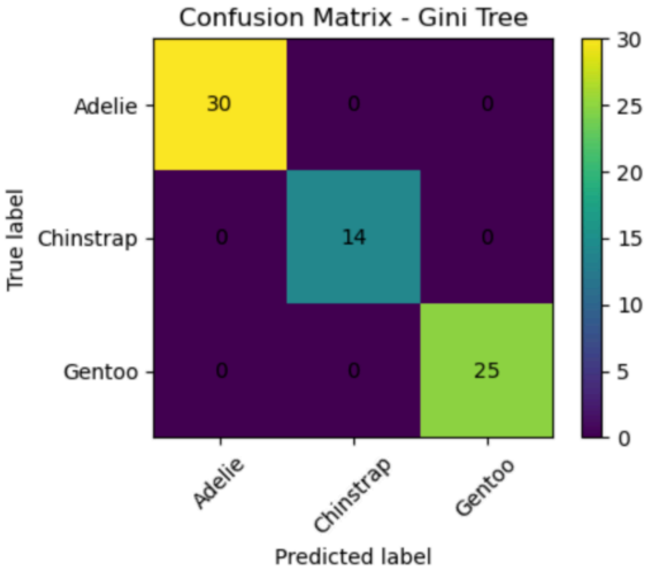
After separating from Gentoo, the next series of splits focuses on differentiating Adelie and Chinstrap penguins. Because these species are similar in size, combinations of bill depth and bill length refine the classification. For instance, chinstrap penguins have longer, narrower bills while adelie penguins have shorter, deeper ones. The tree uses these biological differences on a threshold basis to reduce impurity at each level.

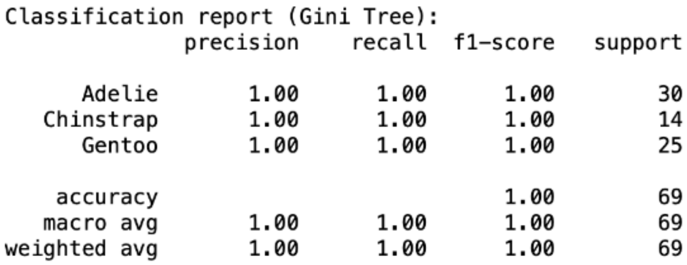
Lower down in the structure body mass is used in a few branches to make yet more precise separations. Though body mass is not the strongest variable in isolation, it is useful when the previous splits narrow the dataset down to smaller, homogeneous groups. The above shows a vital property of Decision Trees not so important features can still be important at deeper levels of the tree.

The tree has a depth of seven. The tree has fourteen leaves. Each leaf node is a pure species. Every path down the tree corresponds to a simple rule which humans can interpret easily.

- If a flipper is below a threshold, then check the depth of the bill.
- If the flipper length exceeds this limit, consider it a Gentoo.

The rules are simple to understand and reflect intuitive biological distinctions. For this reason, Decision Trees are highly valuable for interpretability. The model's 100% accuracy on the training set can be explained by the fact that every leaf is pure , and the high test accuracy suggests that these natural splits are likely to generalise to unseen penguins.





```
Classification report (Gini Tree):
              precision    recall  f1-score   support

      Adelie       1.00      1.00      1.00        30
   Chinstrap       1.00      1.00      1.00        14
      Gentoo       1.00      1.00      1.00        25

    accuracy                           1.00        69
   macro avg       1.00      1.00      1.00        69
weighted avg       1.00      1.00      1.00        69
```

Figure 3 : Confusion Matrix (Gini Tree)

To confirm the performance, generated a confusion matrix shown in Figure 3, which contains a perfect diagonal with no misclassifications. The classification report also shows precision, recall and F1-scores of 1.00 across all classes. This combination of perfect predictive performance and a readable tree structure forms a strong baseline for comparison.

Comparing Gini and Entropy Impurity

Impurity measures influence how Decision Trees choose splits. To explore this, trained a second model using entropy instead of Gini. Both models achieved perfect training accuracy, but the entropy tree achieved a slightly lower test accuracy of 98.55%. The entropy tree also grew differently: it had a depth of six and twelve leaves, making it slightly more compact.

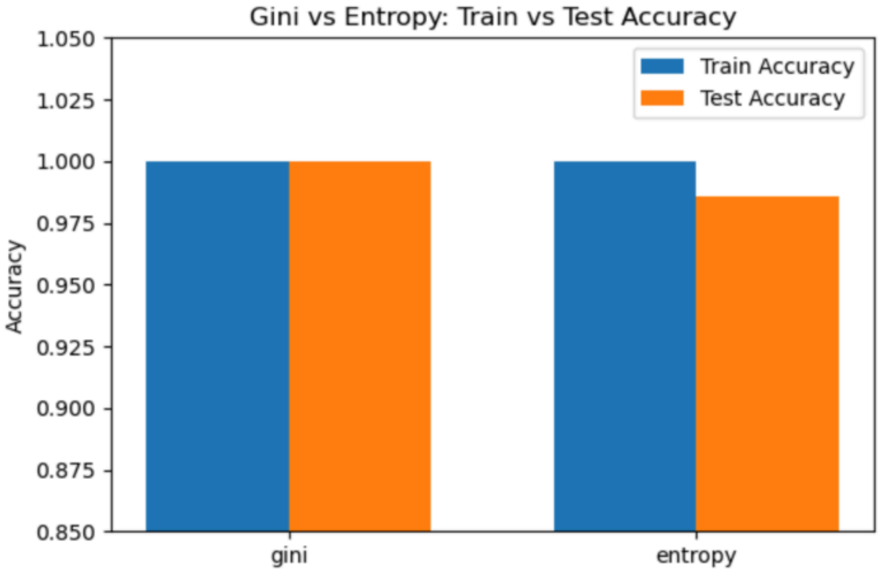


Figure 4 : Gini vs Entropy Accuracy Comparison

These differences are summarised in Figure 4, which compares the train and test accuracy of both impurity measures. Although the performance is similar, the structures differ, and those differences become clear when examining the entropy-based tree in Figure 5.

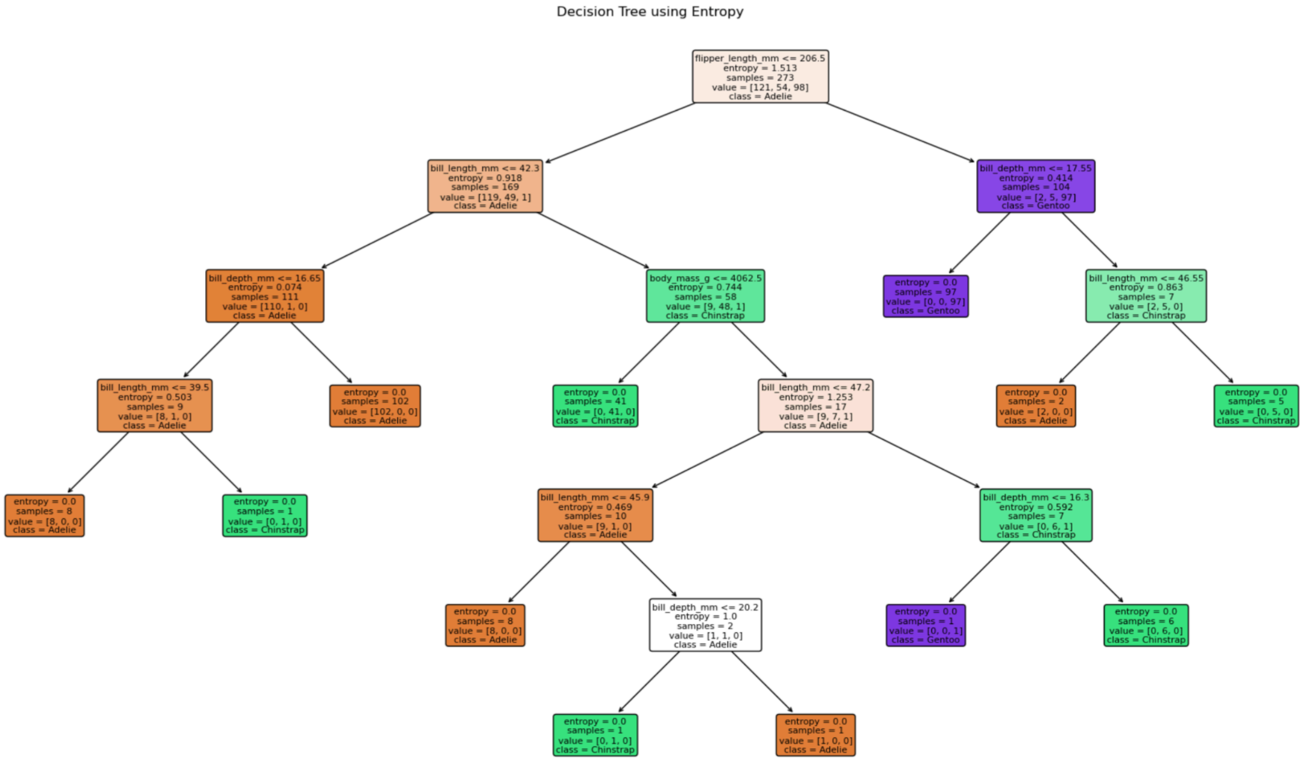


Figure 5 : Full Entropy Tree

The ordering of splits and the precise thresholds differ from the Gini tree, illustrating that impurity choice subtly affects the model's structure even when predictive accuracy remains high.

Depth and Overfitting

Decision Trees tend to overfit when allowed to grow too deep. To examine this, trained a sequence of trees with maximum depths ranging from 1 to 15.

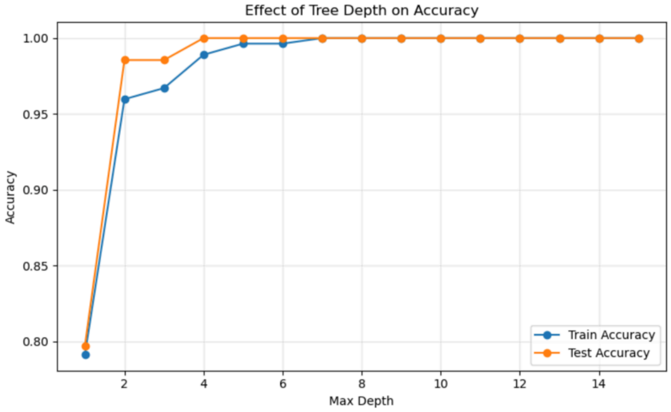


Figure 6 : Depth vs Accuracy Curve

Train and test accuracy with depth is reflected in figure 6. At shallow depths, both accuracies are low since the model is too simple to capture meaningful patterns. As we go deeper our accuracy improved and the test accuracy reached its maximum levels at around depth 4 – 6. The training accuracy on the other hand keeps improving as we go past 6 till it reaches 1.0. The test accuracy then saturates. The illustration shows that while deeper trees can memorize training data, this does not mean they generalize better.

This visualisation was particularly helpful in understanding overfitting. It provides a clear demonstration that more depth does not always equate to better performance, and that Decision Trees benefit from some form of complexity control.

Cost-Complexity Pruning

Cost-complexity pruning is a technique for removing branches of the decision tree that offer a low improvement in impurity. An example of a fully grown tree is the first tree constructed in this tutorial. Fully grown trees tend to be very deep and have many leaves. Trees can perfectly classify the training set, but they also consequently get noisy and fit in very specific aspects of the training set that do not generalise. We prune to achieve an acceptable trade-off between the complexity of the model and its predictive performance.

The parameter ccp_alpha allows Scikit-learn to prune by adding a penalty term to a tree's complexity. The computer values which would make a branch "worth cutting". A ccp_alpha of zero allows a deep and flexible tree and larger values encourage the model to collapse non-useful nodes.

Calculated the complete cost-complexity pruning path for the fully grown Gini tree, which gave a set of candidate ccp_alpha's. Trained a new pruned tree for each value in this sequence and calculated its accuracy on the training set and the test set.

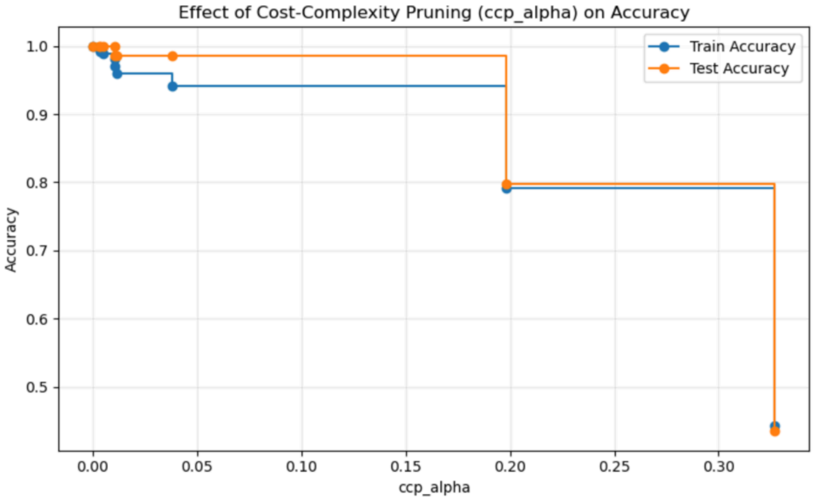


Figure 7 : Cost-Complexity Pruning Curve

The classifier accuracy is influenced by the pruning penalty, as depicted in Figure 7. When the ccp alpha is zero or quite small, the sized tree allows attaining perfect accuracy in the training dataset with high accuracy on the test dataset. When ccp_alpha slightly increases, small and trivial sub-branchings will be removed. Oftentimes, the pruning steps we take early on do not affect the accuracy of our previous test, as the branches deleted are more likely to show noise rather than reflection.

As ccp_alpha increases, pruning becomes bolder, more aggressively removing key segments of the tree. The model will become too simple that the training as well as test accuracy will fall down. This produces the downward section of the pruning curve.

The full tree had the highest accuracy at ccp_alpha = 0.0 (no pruned version was better). Since Palmer Penguins dataset is clean and low-noise, the model expects good performance. Pruning does not always enhance generalisation when there are strong class boundaries.

Though the tree's anatomy stays similar, the pruning curve remains highly useful. It illustrates how Decision Trees respond to penalization-based pruning and demonstrates the trade-off between underfitting and overfitting. Cost-complexity pruning helps produce smaller models that are more interpretable and generalise better.

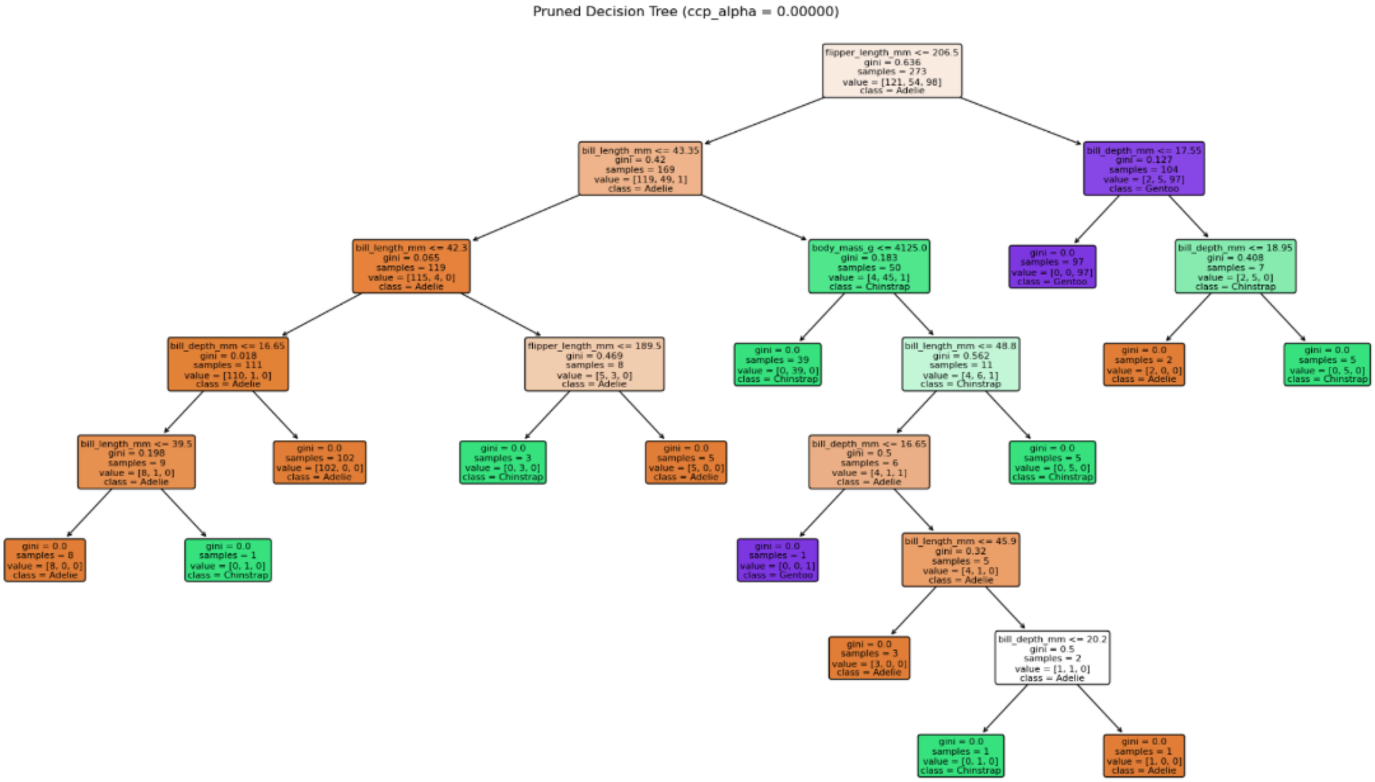


Figure 8 : Pruned Tree (Best ccp_alpha)

Since the best alpha value produced no pruning, the resulting tree is identical to the full Gini tree. Still visualised this model in Figure 8 to complete the conceptual explanation of pruning, even though the structure does not change in this dataset.

Feature Importance

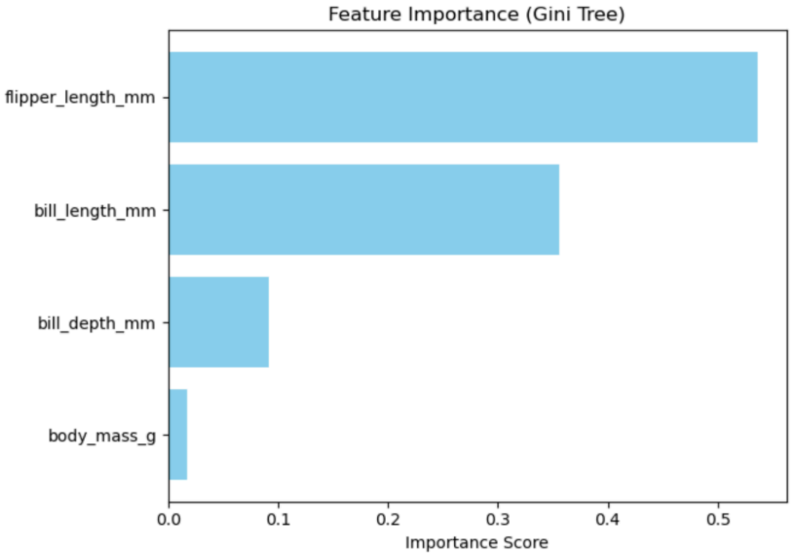


Figure 9 : Feature Importance Plot

Decision Trees offer built-in interpretability through feature importance scores.
Figure 9 shows the importance of each feature in the unpruned Gini tree. Flipper length is clearly the most influential feature, followed by bill length. Bill depth contributes moderately, while body mass has the smallest influence.

These results align with domain knowledge and with the earlier scatter plot. They show that the model is not only accurate but also biologically sensible. The ability to trace how features influence predictions is one of the strengths of Decision Trees, especially in contexts where interpretability matters.

Conclusion

In this tutorial, we looked at Decision Trees using the Palmer Penguins dataset and observed how impurity measures depth and pruning affect model behaviour. Gini and entropy give similar outcomes or results but give slightly different structures, as per the comparison. The depth experiment showed the bias–variance trade-off because deeper trees no longer improved test accuracy. Cost-complexity pruning showed that tree simplifications can affect performance. Though in this dataset the optimal pruning parameter does not reduce the model at all. By looking at feature importance, one can easily see why Decision Trees were able to perform so well.

To sum it up, the experiments give a good idea of how Decision Trees work and why they are used as base learners for ensemble learners like Random Forest and Gradient Boosting. Their transparency and flexibility and the strong performance of tree-based models on structured datasets make them powerful pedagogical devices and applied machine-learning models.

References

- allisonhorst.github.io. (n.d.). *palmerpenguins R data package*. [online] Available at: https://allisonhorst.github.io/palmerpenguins/.
- Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (2017). *Classification And Regression Trees*. [online] Routledge. doi:https://doi.org/10.1201/9781315139470.
- Google Books. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. [online] Available at: https://books.google.co.uk/books?hl=en&lr=&id=X5ySEAAAQBAJ&oi=fnd&pg=PT17&dq=G%C3%A9ron.
- scikit-learn (2019). *sklearn.tree.DecisionTreeClassifier — scikit-learn 0.22.1 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.
- scikit-learn (2025). *1.10. Decision Trees — scikit-learn 0.22 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/tree.html.

GitHub repository :

https://github.com/anittajoshy2001-dot/decision-tree-penguins-tutorial.git