Decision Trees: Theory and Implementation

Machine Learning Foundations

November 24, 2025

Abstract

This document presents a comprehensive analysis of decision tree algorithms for classification and regression. We explore information gain, Gini impurity, and variance reduction as splitting criteria, implement tree construction from scratch, examine pruning techniques, and analyze feature importance measures.

1 Introduction

Decision trees partition the feature space recursively using axis-aligned splits. For classification, a node's impurity can be measured using:

Entropy:

$$H(S) = -\sum_{c=1}^{C} p_c \log_2 p_c \tag{1}$$

Gini Impurity:

$$G(S) = 1 - \sum_{c=1}^{C} p_c^2 \tag{2}$$

where p_c is the proportion of class c in set S.

Information Gain:

$$IG(S, A) = H(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} H(S_v) \tag{3}$$

2 Computational Environment

3 Data Generation

Dataset characteristics: $n =$ **??** samples, $p =$ **??** features, $C =$ **??** classes.

4 Impurity Measures

5 Decision Tree Implementation

6 Decision Boundary Visualization

Training accuracy: **??**, Test accuracy: **??**.

7 Pruning Analysis

Optimal depth: **??** with test accuracy **??**.

8 Feature Importance

Top 3 features: **??** (??), **??** (??), **??** (??).

9 Gini vs Entropy Comparison

10 Results Summary

Table 1: Decision Tree Performance Summary

Metric	Value
Dataset Size	**??**
Number of Features	**??**
Number of Classes	**??**
Optimal Tree Depth	**??**
Best Test Accuracy (Gini)	**??**
Best Test Accuracy (Entropy)	**??**

11 Conclusion

This analysis demonstrated:

- Decision tree construction using Gini impurity and entropy
- The bias-variance tradeoff controlled by tree depth
- Feature importance computation via information gain
- Hyperparameter tuning (max depth, min samples split)
- Visualization of decision boundaries in 2D

The optimal tree depth of **??** balances model complexity with generalization, achieving **??** test accuracy.