决策树：理论与实践

2025 年 9 月 9 日

目录

1 引言

决策树（Decision Tree）通过递归划分特征空间，形成分段常数的预测模型。其优点是可解释性强、对数据预处理要求低，并能处理非线性边界。

2 原理与公式

以分类树为例，在每个节点选择能够最大化“纯度提升”的划分。设节点数据集为 $\mathcal{D}$，类别占比为 p_k。常见纯度指标有基尼与熵：

$$\mathrm{Gini}(\mathcal{D}) = 1 - \sum_k p_k^2, \tag{1}$$

$$\mathrm{Entropy}(\mathcal{D}) = -\sum_k p_k \log p_k. \tag{2}$$

若划分为左右子节点 L, R，则划分后的纯度为

$$I_{\text{split}} = \frac{|L|}{|\mathcal{D}|} I(L) + \frac{|R|}{|\mathcal{D}|} I(R), \tag{3}$$

最优划分使得 $\Delta I = I(\mathcal{D}) - I_{\text{split}}$ 最大。停止条件常包括：最大深度、叶子最小样本数、最小纯度提升等。

3 应用与技巧

- **优点**：可解释、能处理非线性、对特征尺度不敏感、可处理类别与数值特征（需编码）。
- **缺点**：容易过拟合、方差较大；可用集成方法缓解。
- **正则化**：调整 `max_depth`、`min_samples_leaf`，或使用复杂度剪枝。
- **基线对比**：与逻辑回归、SVM、随机森林等模型对比评估。

4 Python 实战

在本章节目录运行下述命令，图片将保存到本目录的 `figures/`：

Listing 1: 生成决策树配图

```
python gen_decision_tree_figures.py
```

5 结果

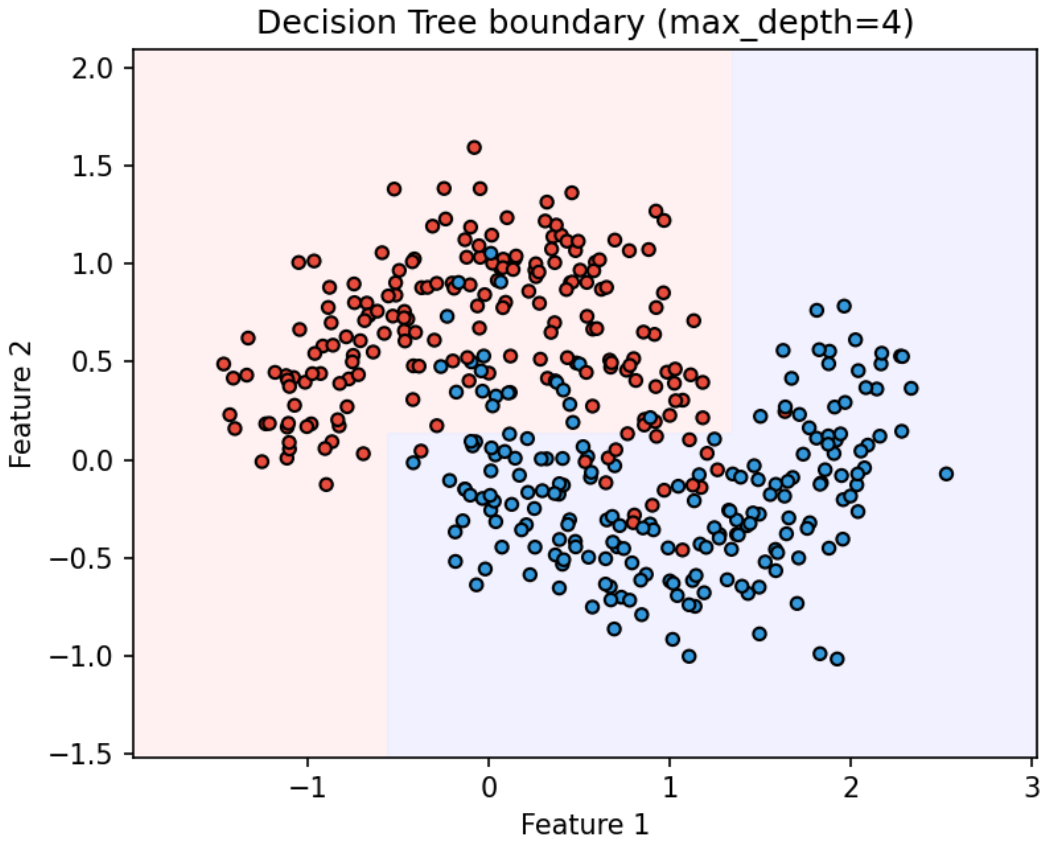

图 1: 决策树在两类数据上的决策边界。

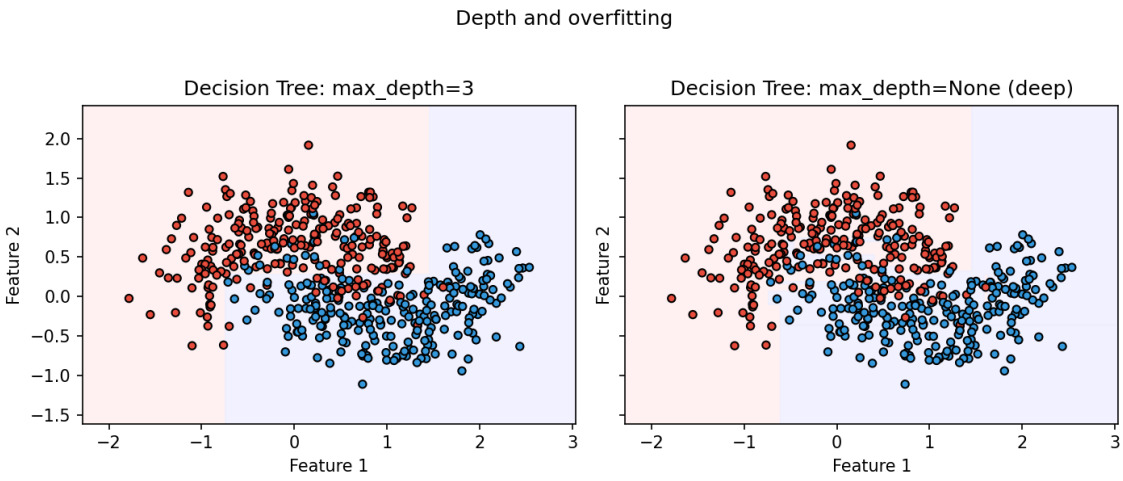

图 2: 树深度影响：浅层与深层（过拟合）对比。

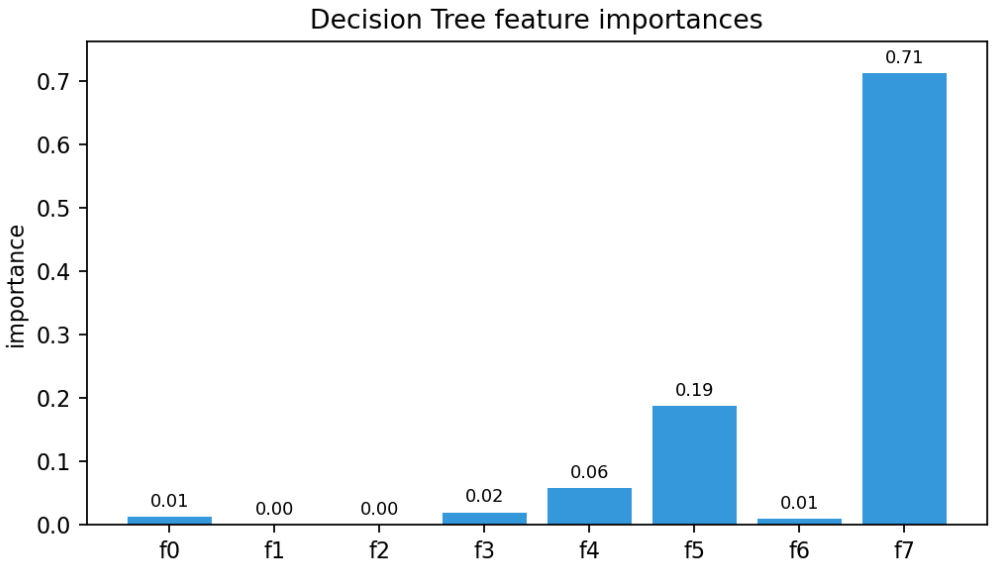

图 3: 决策树的特征重要性可视化。

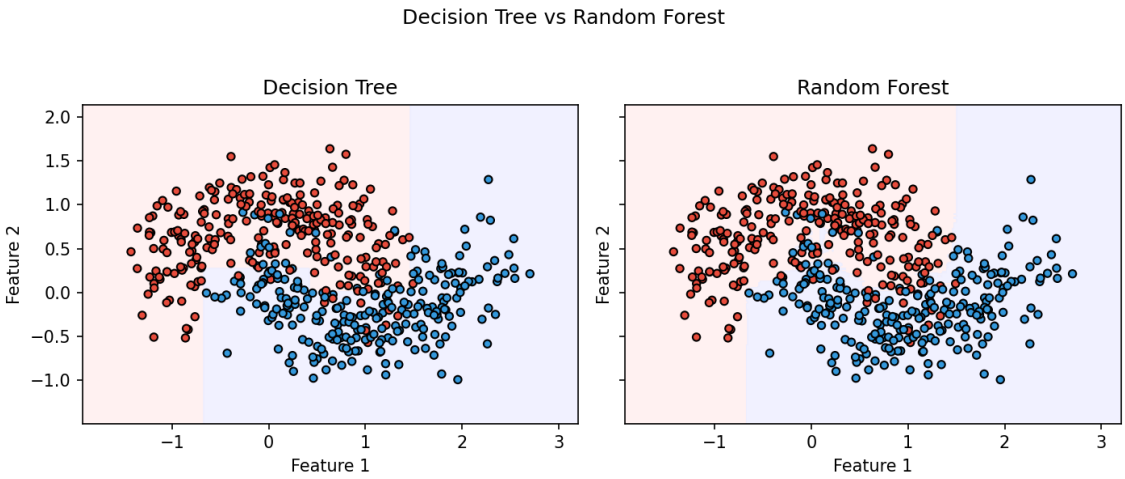

图 4: 单棵决策树与随机森林的决策边界对比。

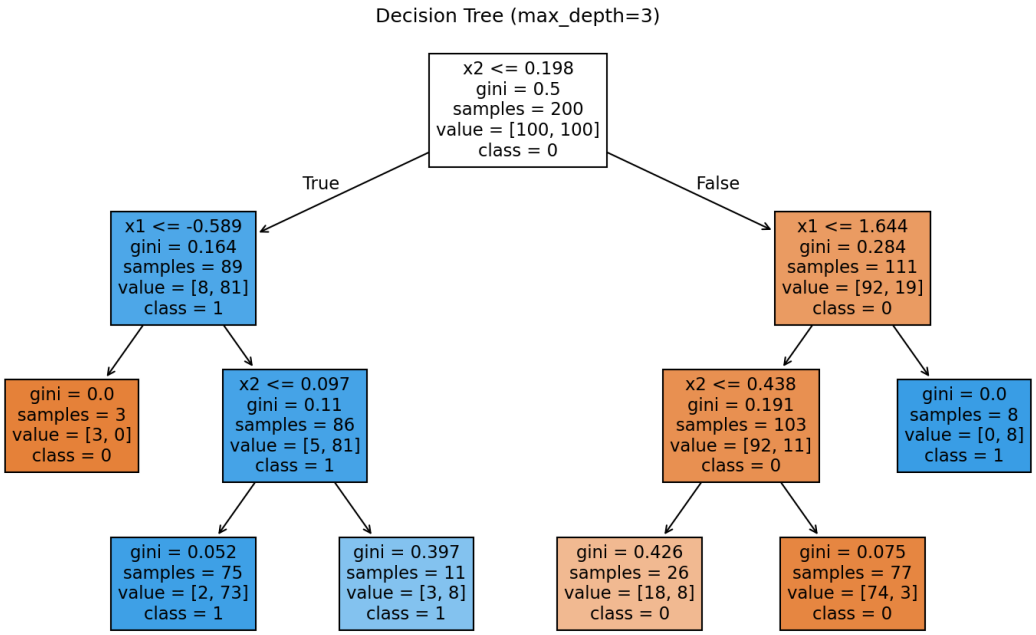

图 5: 树结构可视化（max_depth=3）。

6 总结

决策树作为可解释且灵活的基线模型，在适当的正则化或与集成方法（随机森林、梯度提升）结合时，能在多种任务上取得具有竞争力的表现。