

2.3 Decision Tree





Abstract

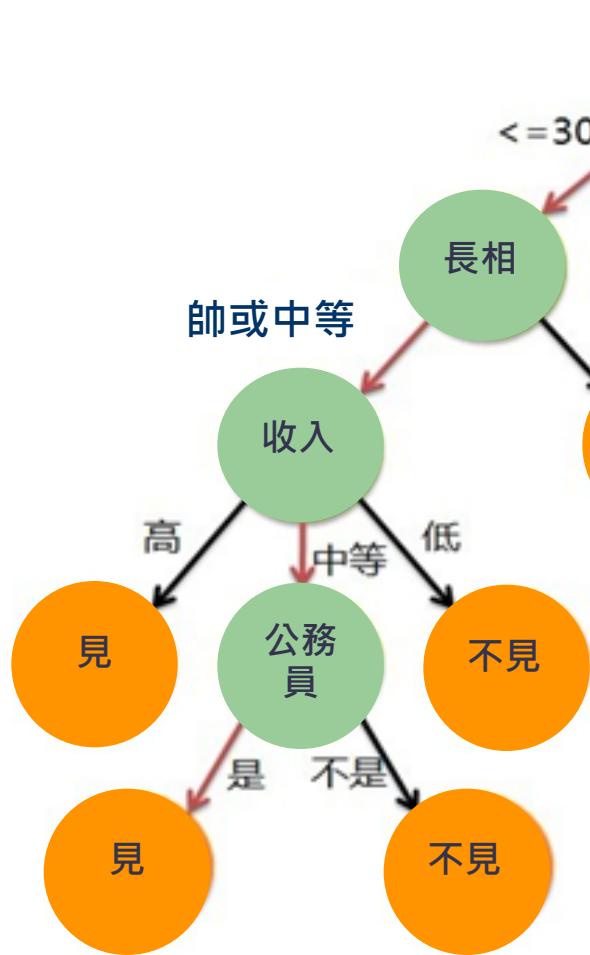
- 決策樹案例
- 模型架構
- 基尼不純度和熵
- 決策樹模型
- 決策樹優缺點

決策樹案例

Why Decision Tree

- Decision trees are powerful and popular tools for **classification** and **prediction**.
- Decision trees represent **rules**, which can be understood by humans and used in knowledge system.

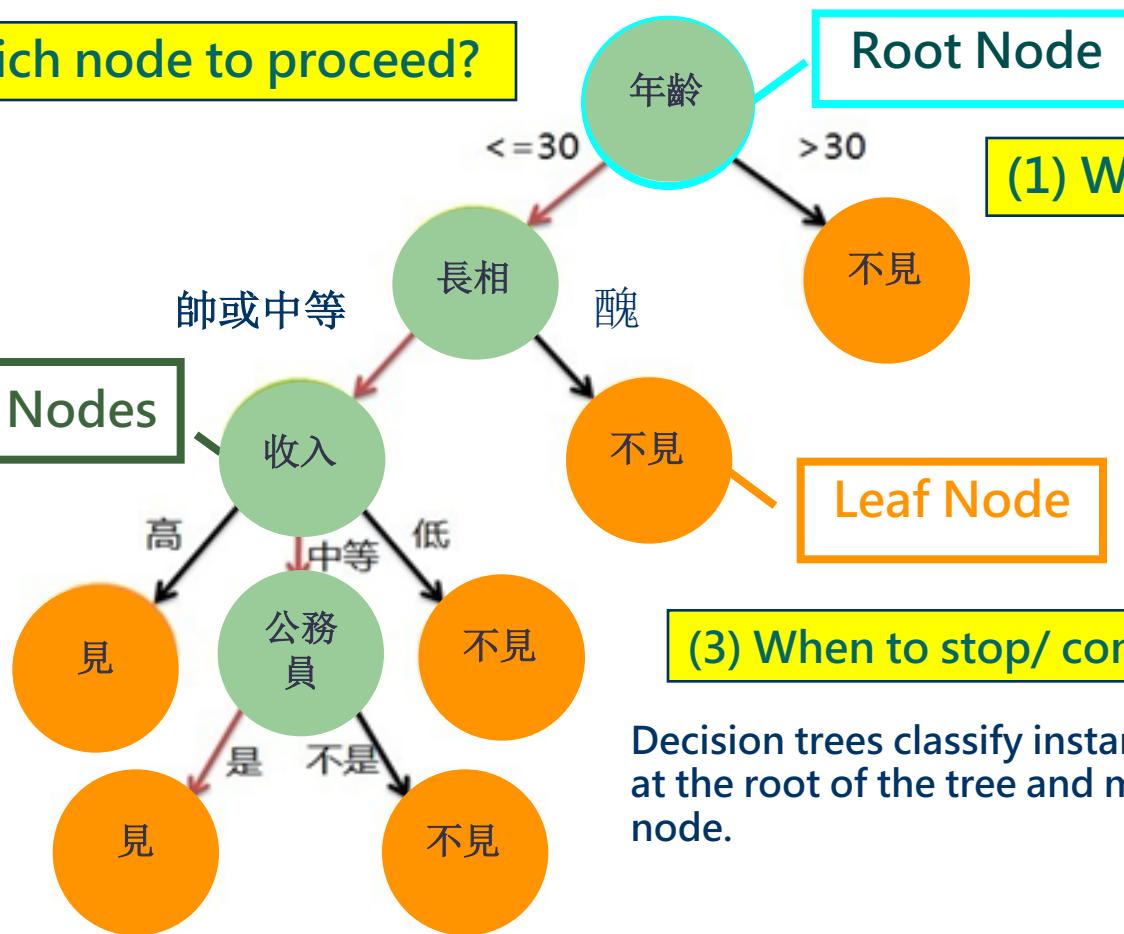
Example – 決策途徑



女兒：多大年紀了（年齡）
媽媽：26
女兒：長的帥不帥（長相）
媽媽：蠻帥的
女兒：收入高不高？（收入）
媽媽：不算很高，中等
女兒：是公務員嗎？（是否是公務員）
媽媽：是，在財政部上班呢。
女兒：那還OK，可以見一見。

Definition

(2) Which node to proceed?



Root Node

(1) Which to start?

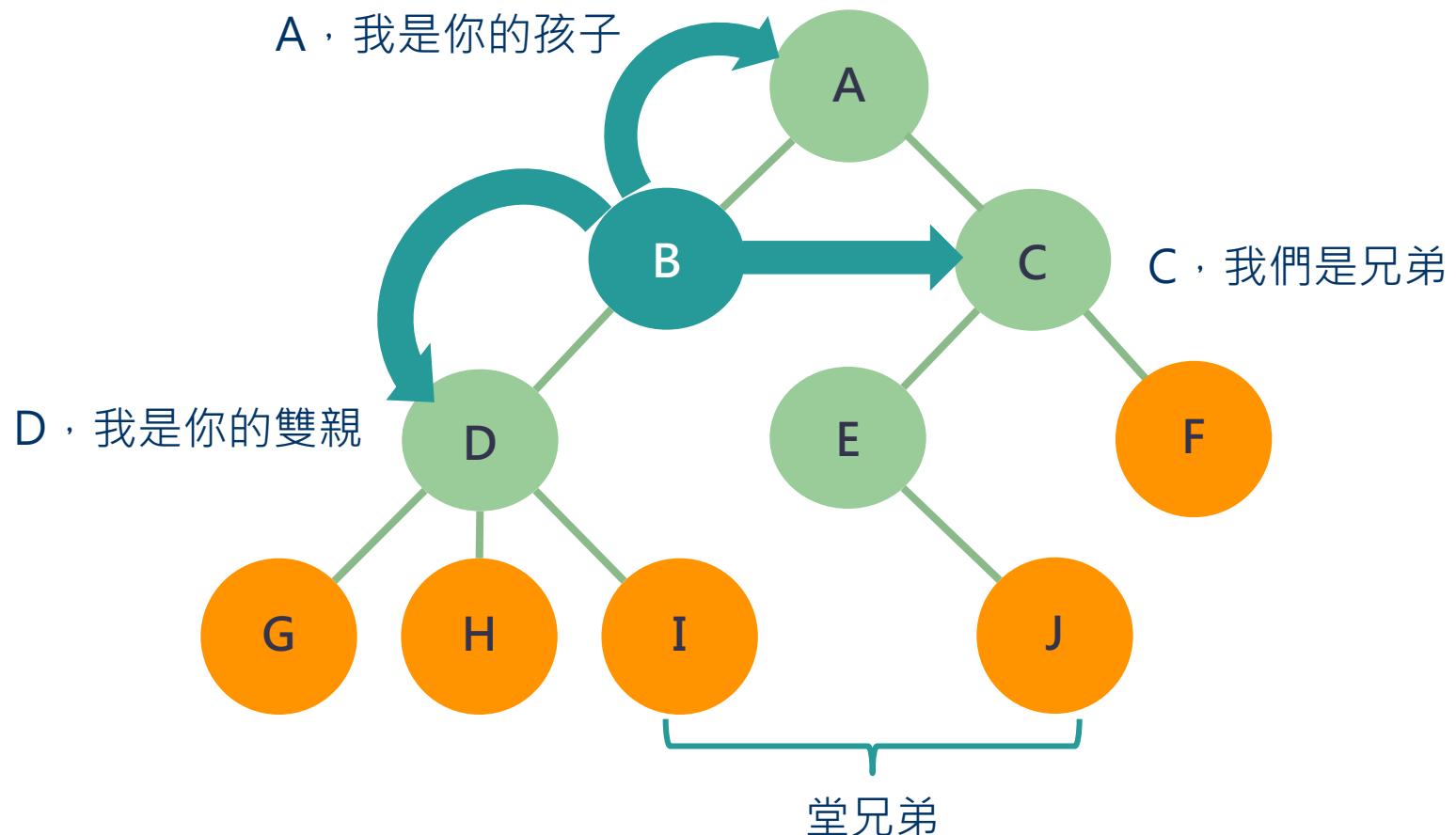
Internal Nodes

Leaf Node

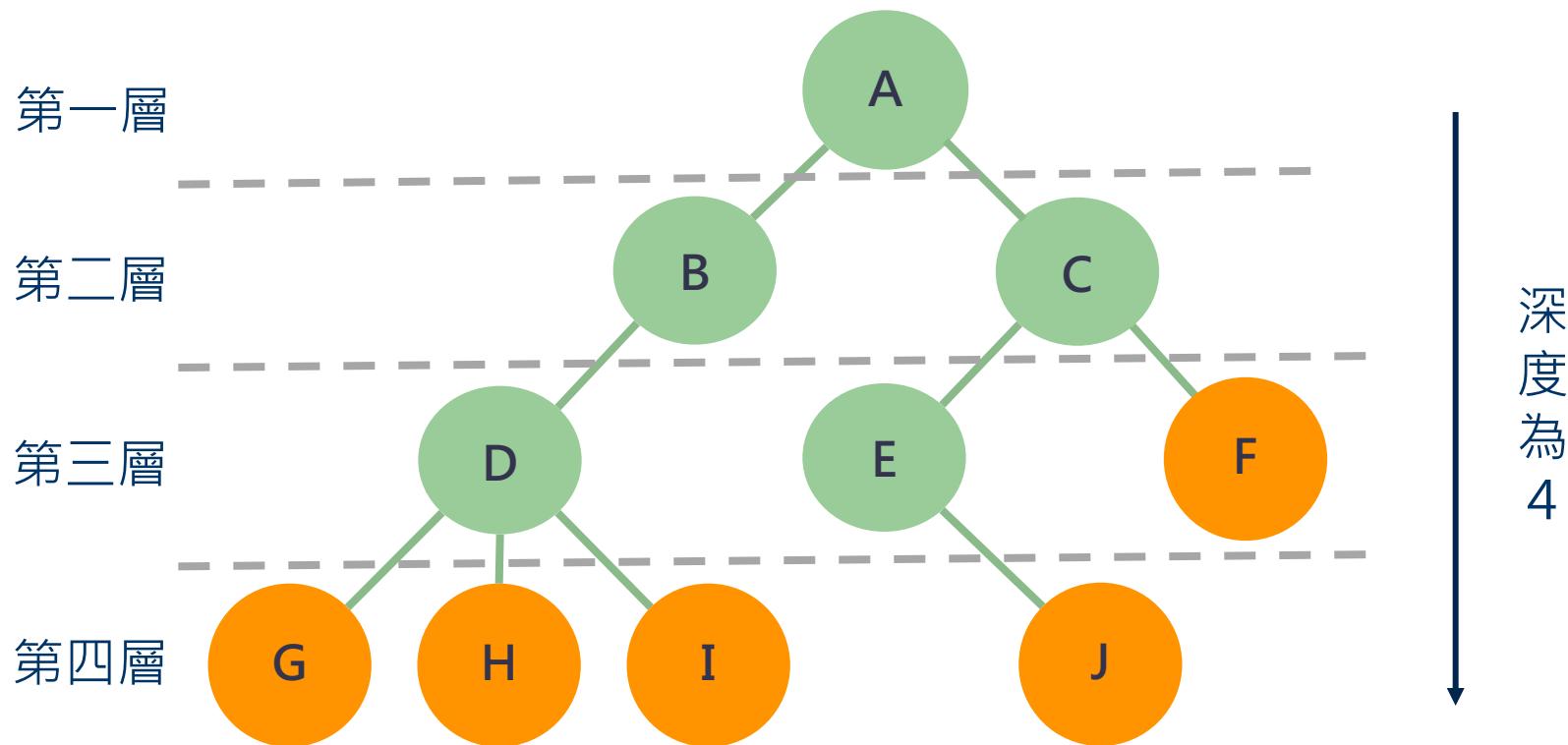
(3) When to stop/ come to conclusion?

Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

Definition



Definition

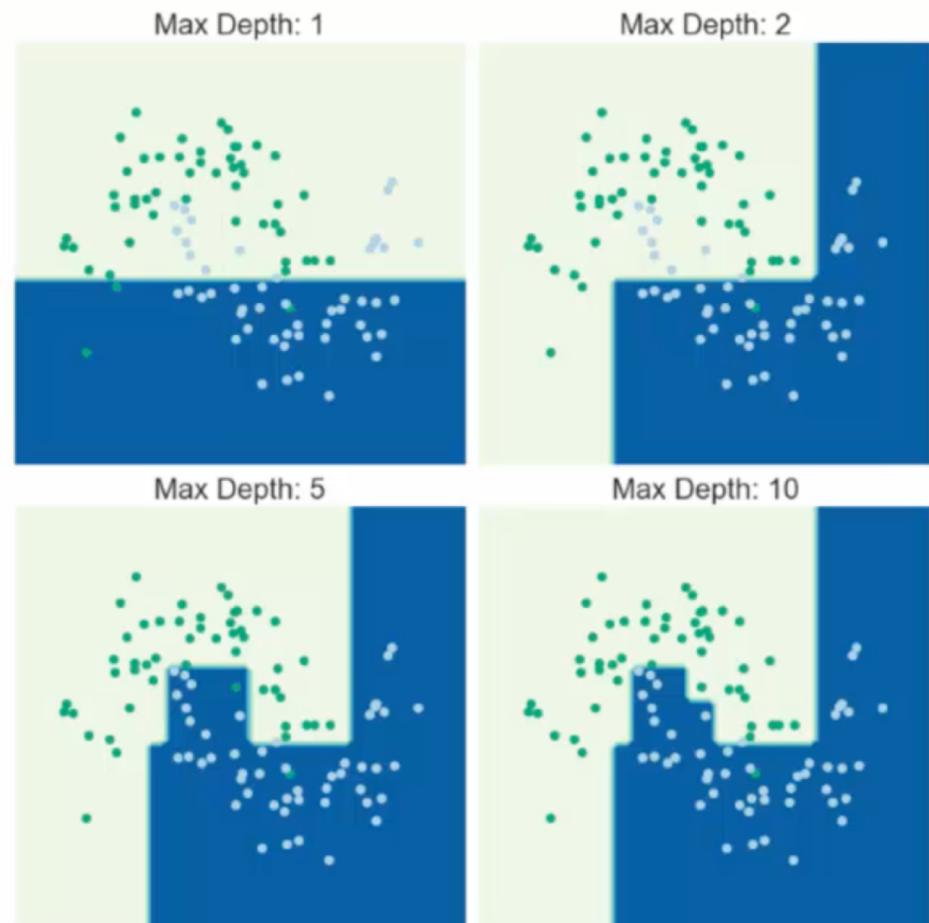


Definition

- 根節點 (Root Node)：無雙親，唯一。
- 葉節點 (Leaf Node)：無孩子，可以多個。
- 中間節點 (Internal Nodes)：一個雙親，多個孩子。

節點

- 增加節點相當於在數據中切一刀
- 節點越多越好嗎？



模型架構

越多資訊越能幫助判斷

	年齡	長相	收入	公務員	見面與否
候選人1	26	7	5萬	是	見
候選人2	35	9	10萬	不是	見
候選人3	22	6	3萬	不是	不見

對判斷目標
的幫助



判斷清楚
程度



清楚判斷和決策過程

清楚判斷

決策過程

目的

納入新變數，幫助
清楚判斷

不同因素對目標的
關係與貢獻度

方法

計算不確定性，衡
量是否清楚判斷

將降低不確定性的
過程視覺化

熵和基尼不純度

熵(Entropy)

衡量現有資訊複雜的程度

越低越好

$$H(X) = - \sum_i^C P(x_i) \log P(x_i)$$

C為類別總數

P 表示分類至該類別的機率

- 面對一個未知問題，資訊量越多，複雜度越高
- 揭露的資訊越多，會越有衡量把握

1. 選取一個或多個X，計算熵
2. 和選取前一次的結果做比較
3. 重複前兩步驟，直到熵不能下降為止

熵(Entropy)的大小代表



熵值高



熵值低

基尼不純度(Gini Impurity)

衡量納入新的X後能不能清楚判斷

$$\text{Gini}(K) = \sum_{i \in N} P_{i,K} (1 - P_{i,K}) = 1 - \sum_{i \in N} P_{i,K}^2$$

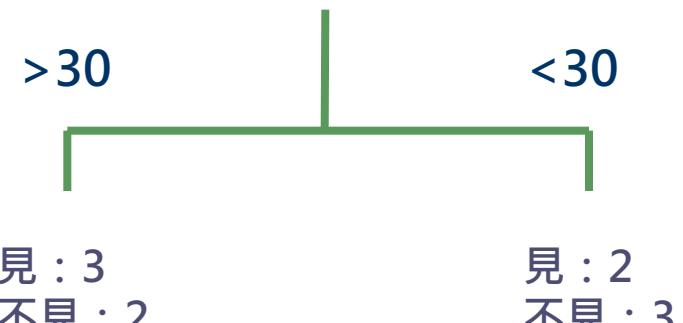
1. 選取一個或多個X，計算基尼不純度
2. 和選取前一次的結果做比較
3. 重複前兩步驟，直到基尼不純度不能下降為止

越低越好

衡量能不能清楚判斷

	年齡	見面與否
候選人1	26	見
候選人2	35	見
候選人3	22	不見
候選人4	37	見
候選人5	25	不見
候選人6	45	不見
候選人7	50	見
候選人8	22	見
候選人9	27	不見
候選人10	38	不見

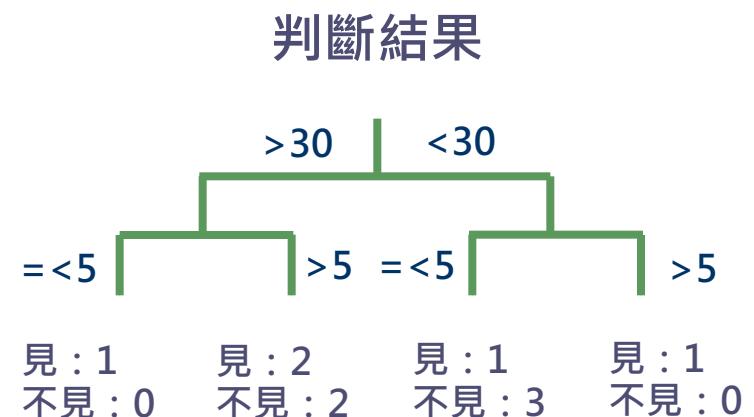
判斷結果



$$3/5 * 2/5 + 2/5 * 3/5 = 0.48$$

衡量加入新變數後能不能更清楚判斷

	年齡	長相	見面與否
候選人1	26	5	見
候選人2	35	9	見
候選人3	22	4	不見
候選人4	37	8	見
候選人5	25	2	不見
候選人6	45	7	不見
候選人7	50	3	見
候選人8	22	10	見
候選人9	27	4	不見
候選人10	38	9	不見



$$1/1*0/1 + 2/4*2/4 + 1/4*3/4 + 1/1*0/1 = 0.4375$$

熵 & 基尼不純度 - 以不同觀點衡量不確定性

熵

以資訊複雜度出發，探討新特徵能否降低整體複雜度，進而更容易判斷

基尼不純度

判斷預測結果是否有區分清楚，降低不純度，來衡量特徵對於判斷目標的幫助

強調每次結果的資訊複雜度

強調直觀理解判斷結果



信息增益(Information Gain)：
知道了某個條件後，原來事件不確定性降低的幅度

決策樹模型

3 種典型的決策樹算法

1. ID3 算法

- ID3 是最早提出的決策樹算法
- 利用信息增益來選擇特徵

2. C4.5 算法

- 是 ID3 的改進版
- 引入「信息增益比」指標作為特徵的選擇依據

3. CART (Classification and Regression Tree)

- 可用於分類，也可用於迴歸問題
- 使用基尼系數取代信息熵模型

3 種典型的決策樹算法

演算法	目的(Y)	分割規則	處理連續值(X)	剪枝	分類節點是否可循環使用	處理缺失值	樣本大小
ID3	分類	信息增益					小
C4.5	分類	信息增益比	✓	✓		✓	中
★ CART	分類 / 迴歸	基尼系數 / MSE	✓	✓	✓	✓	大

- CART，在每一個節點上都是採用二分法，一次只能夠有兩個子節點
- ID3、C4.5在每一個節點上可以產生不同數量的分枝

影響模型行為的判斷依據

在決策樹中，進行屬性劃分時，不同的判定准則會帶來不同的屬性選擇傾向

判斷準則	解釋	傾向
資訊增益 (Information Gain)	關注「資訊不確定性減少了多少」像是「問這個問題之後，情況有多清楚？」	會挑那種可以快速分出一大類別的變數 找區分力最強的問題
資訊增益比 (Gain Ratio)	在資訊增益的基礎上，再看看這個屬性是不是「太細了」就像問一個很小的問題雖然能區分，但其實沒什麼代表性	會避免選「選項太多的變數」，比較平衡 考慮區分力與穩定性的平衡
基尼指數 (Gini Index)	關注「這一群人的純度提高了多少」像是「這一組是不是清一色都是同一類的？」	偏好越快讓資料分類乾淨越好

決策樹三步驟



特徵選擇



決策樹生成



決策樹剪枝

剪枝



為什麼要剪枝？

避免決策樹**過擬合 (Overfitting)**

中的大多數將最終變得多餘，並且不會提高模型的準確性。

決策樹的**複雜性**定義為**樹中的分裂數**。

剪枝是一種去除這種分裂冗餘的技術，即修剪樹中不必要的分裂。剪枝將樹的一部分從嚴格決策邊界壓縮為更平滑和更通用的樹，從而有效地降低樹的複雜性

剪枝策略

- 預剪枝：邊建立決策樹邊進行剪枝操作(實用)
- 後剪枝：當建立完決策樹後，再進行剪枝

剪枝策略

特徵	預剪枝 (Pre-pruning)	後剪枝 (Post-pruning)
執行時機	建樹中	建樹後
是否先建完整樹	否	是
控制方式	設定參數限制	使用驗證集調整結構
優點	快速、簡潔	準確性高、泛化佳
缺點	可能欠擬合	需更多運算資源

實務上如何選擇

實務界主流用法是透過「預剪枝參數 + 交叉驗證」來控樹的深度與複雜度，快速達到泛化與效率的平衡。

原因	說明
效率優先	預剪枝在建樹過程中就停止無效分裂，節省計算時間與記憶體資源，對於大數據量或即時預測任務特別重要。
模型簡潔	通常業界偏好解釋性佳的模型，淺層決策樹配合預剪枝參數（如 <code>max_depth</code> ）就能達到不錯效果，不需要後剪枝才讓模型簡單。
自動化管線適配性高	在 AutoML、Auto-tuning 等場景中，預剪枝參數可輕鬆納入 GridSearch 調參流程，更符合自動化流程需求。
後剪枝成本高	後剪枝需要先建成完整樹，再使用交叉驗證等方式決定要修剪的節點，需要額外的驗證集與大量運算資源，不易大規模部署。

什麼時候用後剪枝比較有價值？

- 需要極高精度或細節保留的模型：如醫療、司法等情境，可先容忍建出複雜樹，再謹慎修剪以保留關鍵規則。
- 模型診斷分析（Model debugging）：後剪枝可視為一種「樹的簡化版比較」，有助於發現冗餘節點。

決策樹優缺點

決策樹的優點

- 易於理解和解釋，可以視覺化分析，容易提取出規則
- 可同時處理連續型和不連續數據(CART)
- 較適合處理有缺失屬性的樣本
- 測試數據時，運行速度較快
- 在相對短的時間內，能夠對大型數據做出可行，且效果良好的結果

決策樹的缺點

1. 易發生過擬合（隨機森林可以很大程度上減少過擬合）
2. 易忽略數據集中屬性的相互關聯：決策樹通常一次只看一個變數（屬性）來做決定，沒辦法理解「兩個變數加起來才有意義」的情況
3. 對不平衡資料很敏感（類別數量差太多會出問題）：當某一類的資料很多，模型會偏心那一類，選擇對它最有利的條件。

Thanks