國立雲林科技大學資訊管理系

資料探勘

Department of Information Management
National Yunlin University of Science & Technology
Data Mining

資料探勘 專案作業一

M11423033 周子齊

M11423036 李映澍

M11423053 童棋逸

指導教授：許中川 博士

Advisor: Chung-Chian Hsu, Ph.D.

中華民國 114 年 10 月

October 2025

摘要

本研究旨在以 UCI Machine Learning Repository 中之 Adult Dataset 為基礎，探討不同決策樹演算法 (ID3、C4.5、C5.0、CART) 於分類問題中的績效差異。研究動機源於決策樹模型兼具高可解釋性，而不同演算法在屬性分裂準則及剪枝策略上的差異，可能導致模型準確率與穩定性之變化。研究方法以 Python 為實作環境，依序進行資料前處理、模型建構、訓練、測試與績效評估，其中 ID3、C4.5、C5.0 採手動建構，CART 則使用 scikit-learn 函式庫，並以準確率 (Accuracy)、精確率 (Precision)、召回率 (Recall) 及 F1-score 為主要評估指標。實驗結果顯示，在演算法比較中，C5.0 模型在分類準確率 (86.40%) 與 F1-score (68.13%) 上表現最佳；CART 雖有最高精確率 (80.95%)，但召回率 (51.92%) 最低；ID3 整體表現最為遜色。在 C5.0 參數調校實驗中，進一步證實「參數優化模型」(Optimized Tree) 透過合理的剪枝並結合 Boosting 技術，能在防止過度擬合（相較於 100%訓練準確率的 Lax Tree）的同時，達到 86.40% 的最佳測試準確率。綜合而言，本研究結論驗證了 C5.0 演算法的穩健性與優越性，並凸顯了參數調校對於模型泛化能力的重要性。

關鍵字： 資料探勘、決策樹、分類、ID3、C4.5、C5.0、CART

1. 緒論

1.1 動機

隨著資料量持續增長，如何有效從龐大資料中萃取有意義的知識，已成為資料科學與人工智慧領域的重要課題。決策樹因具備邏輯清晰、結果可解釋且易於視覺化之特性，廣泛應用於分類與預測任務之中。本研究選擇使用 UCI Machine Learning Repository 所提供之 Adult Dataset 作為實驗資料集，其來源為美國人口普查資料，內容涵蓋個人年齡、教育程度、職業類別、工作時數與收入等多項社會經濟屬性。該資料集之主要任務為預測個人年收入是否超過 50,000 美元，具備明確的二元分類目標與實務應用價值。

Adult Dataset 為機器學習領域中廣為採用的基準資料集之一，具有良好的代表性與可重複性。其同時包含連續變數與類別變數，且部分屬性存在缺失值，能有效檢驗各決策樹演算法在異質資料處理與缺失值補償上的能力。資料集規模適中(共 14 個特徵、約 48,842 筆樣本)，在運算可行性與分析深度之間取得良好平衡。由於該資料集已被廣泛應用於分類模型研究，可作為比較不同演算法績效的共同基準，有助於提升研究結果的可比性與信度。

綜上所述，Adult Dataset 不僅具備理論與實務研究的雙重價值，亦能反映真實社會經濟資料的特性，因此被選為本研究探討不同決策樹演算法 (ID3、C4.5、C5.0、CART) 分類效能差異之理想資料來源，期能深入分析改良型決策樹在實務應用中的表現與優勢。

1.2 目的

本研究旨在比較四種決策樹演算法 (ID3、C4.5、C5.0、CART) 於分類問題中的效能表現，探討不同演算法在分裂準則、剪枝機制與資料處理能力上的差異如何影響模型準確率與泛化能力。具體研究目的如下：

- 建立四種決策樹分類模型，分析其對 Adult Dataset 的分類預測表現。
- 比較各模型於不同資料前處理條件下之分類績效。
- 評估不同演算法於資料屬性 (連續值與類別值) 混合資料集中的適應性。
- 綜合討論各演算法之優缺點，作為未來選擇決策樹模型之依據。

2. 方法

2.1 研究架構

本研究為達成緒論所述之研究目的 ，即比較 ID3、C4.5、C5.0 與 CART 四種決策樹演算法之分類績效，因此建立一套系統性的研究框架。此框架依循資料探勘的標準流程，加入研究之目的，從資料獲取與前處理、模型建構、比較四種演算法績效、比較特定演算法不同參數績效、評估模型績效與比較不同參數績效等六個主要階段。整體研究流程如圖 1 所示。

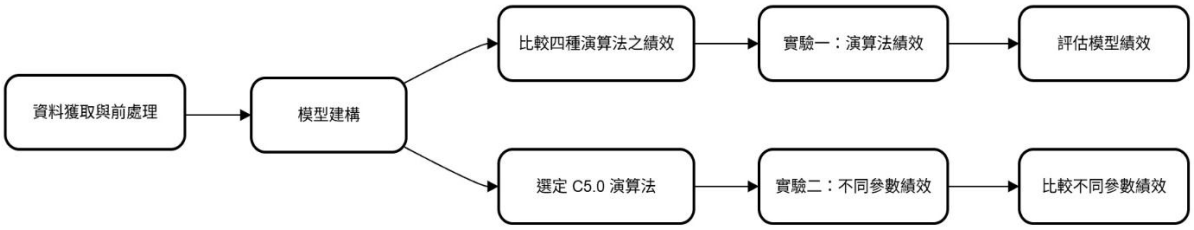

圖 1. 研究流程圖

本研究之詳細執行步驟說明如下：

1. **資料獲取**：本研究採用之 Adult Dataset，係取自 UCI 機器學習儲存庫 (UCI Machine Learning Repository)。該資料集因其屬性多樣性及明確的分類目標，廣泛適用於評估分類模型之效能。
2. **資料前處理**：此階段為模型建構的基礎。由於原始資料包含缺失值、重複值、多樣的類別型與連續型變數，故需進行系統性的資料清理與轉換，以符合後續不同演算法的輸入要求。此外，官方資料集已預先劃分為訓練集(約 66.7%)與測試集(約 33.3%)，其比例接近 2：1。 本研究將直接採用此劃分進行後續的模型訓練與測試。
3. **模型建構與訓練**：分別運用 ID3、C4.5、C5.0 與 CART 四種決策樹演算法，在相同的訓練集上建構分類模型。
4. **模型評估與比較**：為客觀評量各模型的泛化能力，本階段使用獨立的測試集對訓練完成的四個模型進行績效驗證。評估指標採用混淆矩陣 (Confusion Matrix) 衍生之準確率 (Accuracy)、精確率(Precision)、召回率 (Recall) 與 F1-score ，藉此進行全面性的績效比較。
5. **實驗結果與結論**：本階段將彙整所有實驗數據，透過圖表進行視覺化呈現與分

析，深入比較四種演算法在 Adult 資料集上的表現差異，最終提出本研究的結論與未來展望 。

2.2 決策樹演算法理論

決策樹是一種廣泛應用於分類與迴歸的監督式學習模型。其核心思想是透過一系列的決策規則，將資料集遞迴地劃分為更小、更同質的子集。一個決策樹模型由節點 (Nodes) 和有向邊 (Edges) 組成，其中包含根節點 (Root Node)、內部節點 (Internal Nodes) 和葉節點 (Leaf Nodes)。

本研究旨在比較四種經典的決策樹演算法：ID3、C4.5、CART 與 C5.0 。這四種演算法在屬性劃分準則、剪枝策略及對不同資料類型的處理能力上各有區別。

2.2.1 ID3(Iterative Dichotomiser 3)

ID3 演算法由 Ross Quinlan [2] 於 1986 年提出 ，是早期決策樹的代表。ID3 的核心是採用資訊增益 (Information Gain, IG) 作為屬性選擇的準則 。

- Entropy：用以衡量一個資料集 S 的不純度 (Impurity)。Entropy 越高，表示資料集的不確定性越高。其定義如下：

$$Entropy(S) = \sum_{i=1}^{c} -p_i \, log_2(p_i)$$

- Information Gain： IG(S, A) 衡量的是「在使用屬性 A 對資料集 S 進行劃分後，所帶來的 Entropy 降低量」。ID3 會選擇具有最大資訊增益的屬性作為劃分節點。ID3 演算法缺乏剪枝機制 ，容易產生過度擬合 (Overfitting) ；其偏好選擇具有較多值的屬性；且無法直接處理連續型屬性。其中 Values(A)是屬性 A 的所有可能值，S_v 是 S 中屬性 A 值為 v 的子集。

$$IG(S\,,A) = Entorpy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E\,ntorpy(S_v)$$

2.2.2 C4.5

為了解決 ID3 的缺點，Quinlan [3] 後續提出了 C4.5 演算法 。C4.5 採用增益率 (Gain Ratio) 作為劃分準則 ，以校正資訊增益偏好多值屬性的問題。

- Gain Ratio：Gain Ratio 在 Information Gain 的基礎上，引入了一個懲罰項，稱為

分裂資訊 (SplitInfo)，用以衡量屬性 A 本身的 Entropy。C4.5 選擇具有最高 Gain Ratio 的屬性進行劃分。C4.5 能夠處理連續型屬性(透過二元切分找到最佳閾值)與缺失值。此外，它引入了最小錯誤剪枝 (Minimum Error Pruning) 的後剪枝策略，以提升模型的泛化能力。

$$GR(S,A) = \frac{IG(S,A)}{SplitInfo(S,A)}$$

$$SplitInfo(S,A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} log_2(\frac{|S_v|}{|S|})$$

2.2.3 CART (Classification and Regression Trees)

CART 演算法由 Breiman [1] 等人於 1984 年提出，其特點是無論屬性類型，皆建構二元樹 (Binary Tree)。CART 在分類任務中採用 Gini 不純度 (Gini Index) 作為劃分準則。

- Gini 不純度 (Gini Index)：Gini Index 衡量從資料集 S 中隨機選取兩個樣本，其類別標記不一致的機率。Gini Index 越小，表示資料集的純度越高。CART 會選擇使 Gini Index 下降最多的屬性及其切分點。其定義如下：

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

- 剪枝策略： CART 採用成本複雜度剪枝 (Cost-Complexity Pruning, CCP)， CCP 透過一個複雜度參數 α 來平衡模型的擬合程度與樹的複雜度。

2.2.4 C5.0

C5.0 是 Quinlan [4] 在 C4.5 基礎上發展的商業版本，在效能和記憶體使用上進行了優化。C5.0 的核心改進之一是引入了 Boosting (提升法) ，使其不僅是一個單一的決策樹，更可視為一個決策樹的集成模型。此外，它採用了更為複雜的剪枝與規則簡化策略 。

2.3 實作環境與工具

為確保研究的可重複性，本節將詳細說明實驗所使用的軟體環境及程式庫。

2.3.1 開發環境

本研究所有實驗均在 Anaconda 虛擬環境中進行 ，採用 Python 3.11 版本 。程式碼的撰寫、執行與調適，均使用 Jupyter Notebook 互動式介面 。

2.3.2 程式庫

- Pandas：用於讀取、清理 Adult 資料集，並進行資料框架(DataFrame)的操作。
- Numpy：提供高效能的陣列運算，支援資料前處理中的數值計算。
- CART：採用業界主流的 scikit-learn (sklearn) 函式庫，使用其 DecisionTreeClassifier 模組。
- Scikit-learn：除了 CART 模型的實作外，本研究亦使用其 metrics 模組來計算模型的績效指標。
- Graphvia：用於繪製實驗二的視覺化決策樹。

2.4 績效評估與指標

為客觀且全面地評估四種決策樹模型在 Adult 資料集上的分類效能，本研究採用混淆矩陣 (Confusion Matrix) 及其衍生指標。

2.4.1 混淆矩陣

淆矩陣是用於視覺化分類模型準確性的矩陣。針對本研究的二元分類問題(預測收入是否 >50K)，矩陣定義如下：

- True Positive, TP：實際 >50K，模型預測 >50K。
- True Negative, TN：實際 <=50K，模型預測 <=50K。
- False Positive, FP：實際 <=50K，模型預測 >50K (Type I Error)。
- False Negative, FN：實際 >50K，模型預測 <=50K (Type II Error)。

2.4.2 評估指標

基於混淆矩陣，本研究選用以下四個關鍵指標：

1. 準確率 (Accuracy)：衡量模型「整體預測正確」的比例。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. 精確率 (Precision)：衡量在所有被模型預測為「> 50K」的樣本中，有多少比例是「真正 >50K」的。

$$Precision = \frac{TP}{TP + FP}$$

3. 召回率 (Recall / Sensitivity)：衡量在所有實際為「>50K」的樣本中，有多少比例被模型「成功找出」。

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score：F1-Score 是 Precision 和 Recall 的調和平均數 (Harmonic Mean)，可作為一個綜合性指標，尤其適用於類別不平衡的資料集。

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3. 實驗

3.1 資料集

本研究使用之資料集為 Adult Dataset，來自 UCI Machine Learning Repository 。該資料集最初由美國人口普查局 (U.S. Census Bureau) 於 1994 年之調查資料中擷取，為機器學習與資料探勘領域中常用的資料集之一，主要任務為根據個人基本屬性與就業特徵，預測其年收入是否高於 50,000 美元。

原始筆數 (Total Instances) 總共 48,842 筆記錄(包含訓練集和測試集)欄位數 (Number of Attributes) 共有 15 個欄位(包含 14 個特徵欄位與 1 個目標欄位)。

目標變數 (target variable) 為 income，其為二元分類，預測年收入是否大於 $50K 或小於等於 $50K。

3.2 前置處理

由於 Adult Dataset 含有遺漏值及多類別屬性，需先進行資料清理與轉換。針對不同演算法，本研究採用以下前置處理策略：

- 缺失值處理：將 "?" 視為缺失資料，刪除該筆紀錄或以眾數補值。
- 類別型資料編碼：ID3、C4.5、C5.0 採用 Label Encoding 轉換為整數代碼。
- CART 使用 One-Hot Encoding 處理多類別屬性。
- 連續變數離散化：針對 ID3 與 C4.5 不支援連續屬性之情況，使用等寬分箱 (Equal-width Binning) 將數值屬性離散化。
- 資料標準化：對連續屬性進行 Min-Max Normalization，以確保特徵值範圍一致，使所有特徵映射至 [0,1] 範圍。

3.3 實驗設計

本研究的實驗設計分為兩部分。第一部分旨在全面比較四種決策樹演算法的基礎效能；第二部分則深入探討特定演算法在不同參數設定下的績效變化，以分析決策樹深度、節點分裂、葉節點數量等參數對模型的影響。

3.3.1 實驗一：四種決策樹演算法績效比較

本實驗旨在系統性地比較 ID3、C4.5、C5.0 及 CART 在 Adult 資料集上的分類效能。為深入理解各演算法的核心機制，本研究在實作上採取了兩種策略：

1. 理論為本的手動建構：針對 ID3、C4.5 與 C5.0 三種演算法，本研究依據其發表的原始論文與核心理論，使用 Python 語言從零開始手動建構演算法邏輯。

    - ID3: 依據 Quinlan [2] 的定義，以資訊增益 (Information Gain) 為核心分裂準則進行實作。
    - C4.5: 依據 Quinlan [3] 的後續研究，以增益率 (Gain Ratio) 取代資訊增益，並內建處理連續值與缺失值的機制。
    - C5.0: 實作版本整合了 Boosting 框架，以多棵樹的投票結果來提升模型的準確性與穩定性。

2. 標準函式庫的基準：針對 CART 演算法，本研究採用 scikit-learn 函式庫中之 DecisionTreeClassifier 類別進行實作，該函式庫以 Gini 不純度 (Gini Impurity) 為分裂準則，其穩定性與效能已通過廣泛驗證，可作為其他三種手動建構演算法的可靠效能基準線。

3.3.2 實驗二：C5.0 演算法參數調校與模型比較

在實驗一比較四種演算法後，本研究選定 C5.0 演算法進行深入的參數調校分析。C5.0 演算法允許透過多項參數來控制決策樹的生長，以平衡模型的準確性與複雜度，從而避免過度擬合 (Overfitting) 或擬合不足(Underfitting)。

本研究旨在探討不同參數配置對 C5.0 模型效能的影響。設計三種具代表性的參數情境，並比較其在 Adult 資料集上的分類預測正確率：

1. Simple Tree：此設定旨在建立一個結構簡單、高可解釋性的基礎模型。嚴格限制樹的最大深度 (max_depth = 3)，並大幅提高節點分割 (min_samples_split = 2000) 與葉節點所需的最小樣本數 (min_samples_leaf = 1000)。此舉能有效防止模型學習到資料中的噪訊，但可能導致擬合不足。
2. Lax Tree：此設定代表一種極端的參數配置，旨在觀察模型在幾乎不受限制下的學習能力與過度擬合的程度。不限制最大深度 (max_depth = None)，並將節點分割 (min_samples_split = 2)與葉節點的樣本數降至最低 (min_samples_leaf = 1)。預期此模型將在訓練資料上達到極高的準確率，但泛化能力(測試資料準確率)可能不佳。

3. Optimized Tree：此設定旨在透過合理的參數調校，在模型的複雜度與準確性之間尋求最佳平衡。設置中等的最大深度 (max_depth = 15)，並配合 C5.0 特有的 Boosting 功能(boosting = 20)，同時設定合理的葉節點 (min_samples_split = 300) 與分割樣本數(min_samples_leaf = 50)。此組參數期望能在測試資料上達到最佳的泛化能力與預測準確率。

本研究將分別使用這三種參數訓練模型，並比較其在測試集上的準確率，以及其最終的樹狀結構(樹深與節點數)。

3.3.3 演算法剪枝設定

表 1 模型剪枝策略表

模型	分裂準則	剪枝策略	實作來源
ID3	Information Gain	無剪枝	自行實作
C4.5	Gain Ratio	Minimum Error Pruning	自行實作
C5.0	Boosted Gain Ratio	Adaptive Pruning & Rule-based Simplification	自行實作
CART	Gini Index	Cost-Complexity Pruning	scikit-learn

3.4 實驗結果

3.4.1 演算法績效比較

四種演算法在相同前處理條件下進行訓練與測試，結果如下：

表 2 模型績效分析比較表

模型	Accuracy	Precision	Recall	F1-score
ID3	83.03%	67.24%	54.97%	60.49%
C4.5	85.22%	72.66%	60.06%	65.77%
C5.0	86.40%	76.31%	61.54%	68.13%
CART	85.75%	80.95%	51.92%	63.27%

根據表 2 的比較數據，C5.0 演算法在整體準確率 (Accuracy) 上表現最佳，達到 86.40% 。其次依序為 CART (85.75%) 和 C4.5 (85.22%)，而 ID3 演算法的準確率最低，為 83.03% 。

在精確率 (Precision) 指標上，CART 模型的表現最為突出，達到 80.95% ，顯著高於其他三種模型，這意味著在所有被 CART 預測為「收入 >50K」的樣本中，其正確的比例最高。

在召回率 (Recall) 方面，即模型成功找出所有實際為「>50K」樣本的能力，C5.0 表現最好，為 61.54%，C4.5 (60.06%) 緊隨其後 。CART 的召回率則在四者中最低，僅 51.92% 。

F1-Score 作為精確率與召回率的調和平均數，C5.0 獲得了 68.13% 的最高分，顯示其在兩項指標間達成了最佳的綜合平衡。

綜合而言，C5.0 演算法在準確率、召回率、F1-Score 三項關鍵指標上均排名第一，顯示其綜合分類效能最為穩健且均衡。CART 模型呈現出高精確率與低召回率的顯著特徵，表明其預測策略較為保守，傾向於在有高度把握時才將樣本預測為正類，但也因此遺漏了較多實際為正類的樣本。相較之下，ID3 演算法在所有四項績效指標中均表現最差 。

3.4.2 C5.0 參數調校分析

根據 3.3.2 節實驗二的設計，本研究使用 C5.0 演算法搭配三種不同參數組合進行訓練與測試，其訓練資料與測試資料的準確率結果彙整如表 3 所示。

表 3 C5.0 三種參數績效表

	Simple Tree	Lax Tree	Optimized Tree
訓練資料準確率	81.53%	100.00%	86.74%
測試資料準確率	81.25%	82.03%	86.40%

- Simple Tree

此模型由於受到嚴格的參數限制，結構非常簡單。其訓練準確率 (81.53%) 和測試準確率 (81.25%) 非常接近，顯示模型具有穩定的泛化能力，幾乎沒有發生過度擬合。然而，其整體準確率在三者中最低，表明模型可能存在擬合不足 (Underfitting) 的情況，未能充分學習資料中的複雜模式。

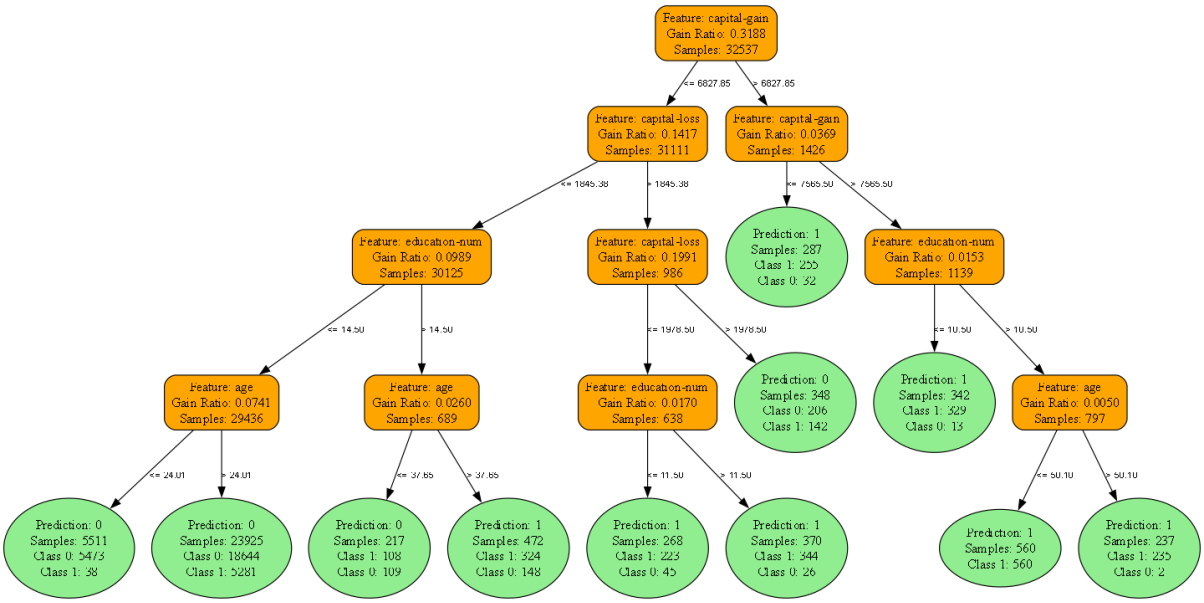

圖 2 Simple Tree

- Lax Tree

此模型在完全不受限制下生長，於訓練資料上達到了 100.00% 的完美準確率，顯示其幾乎記住了所有訓練樣本的特徵。然而，其在測試資料上的準確率僅為 82.03%，遠低於訓練準確率。訓練與測試準確率之間有近 18% 的差距，明確證實了此模型已發生嚴重的過度擬合 (Overfitting)，導致其泛化能力不佳。

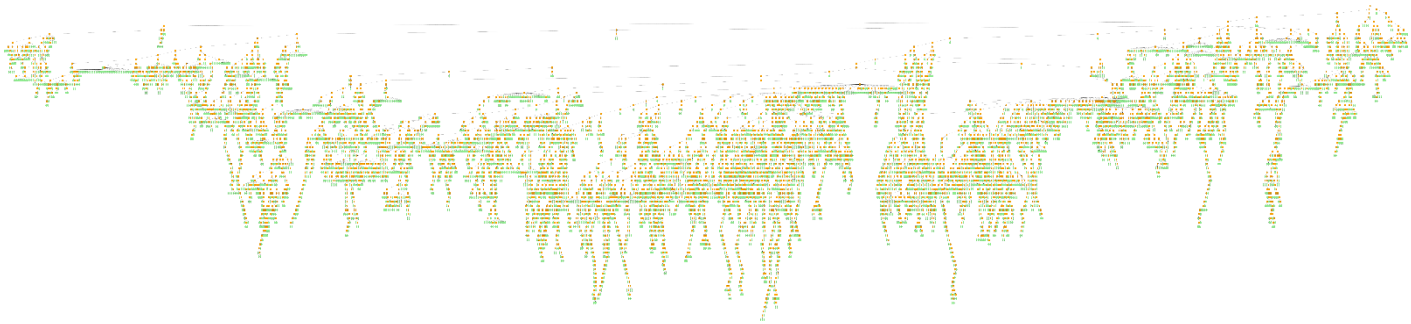

圖 3 Lax Tree

- Optimized Tree

此模型透過合理的參數限制(max_depth=15)並啟用了 20 次 Boosting，在訓練資料 (86.74%) 與測試資料 (86.40%) 上均達到了三組實驗中的最高準確率。更重要的是，兩者的準確率僅差距 0.34%，顯示此模型在保持高準確性的同時，也具備了優秀的泛化能力，成功地在擬合不足與過度擬合之間找到了最佳平衡點。

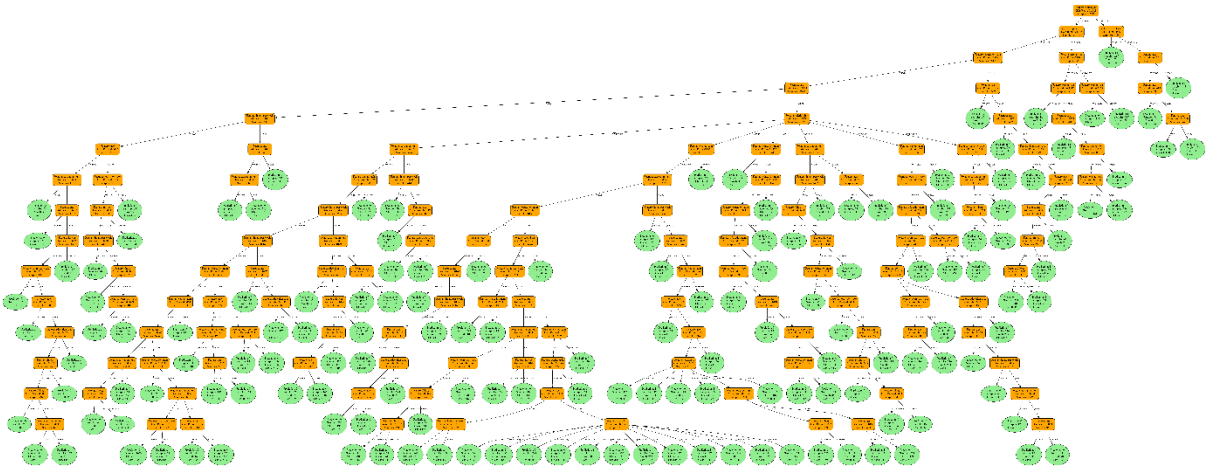

圖 4 Optimized Tree

4. 結論

本研究旨在以 Adult 資料集為基礎，系統性地比較 ID3、C4.5、CART 與 C5.0 四種經典決策樹演算法的分類效能，並深入探討 C5.0 演算法的參數調校對模型泛化能力的影響。本團隊依據原始論文理論，手動建構了 ID3、C4.5、C5.0 演算法，並以 scikit-learn 函式庫中的 CART 作為效能基準。在第一階段的基礎模型比較中，實驗結果表明 C5.0 在測試集上展現了最佳的綜合效能（測試準確率 86.40%），其表現優於 C4.5（85.22%）與 CART（85.75%），而 ID3 演算法（83.03%）則因其在處理連續值與屬性選擇上的理論限制，效能明顯落後，此發現與各演算法的理論預期相符。

接著，在第二階段的 C5.0 參數調校實驗中，本研究進一步量化了模型複雜度對效能的關鍵影響。結果顯示，「高度剪枝模型」因限制過嚴而導致擬合不足，其測試準確率僅為 81.25%；與此相對，「完全生長模型」雖在訓練集達到 100.00% 準確率，卻因模型過於複雜而導致嚴重的過度擬合，其測試準確率驟降至 82.03%。最終，「參數優化模型」透過設定合理的最大深度並結合 20 次的 Boosting 技術，在準確性與泛化能力間達成了最佳平衡，獲得了本次研究的最高準確率，其訓練集與測試集準確率分別為 86.74% 與 86.40%，有力地證明了合理的剪枝與 Boosting 技術是提升 C5.0 效能的有效途徑。

參考文獻

[1] BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., AND STONE, C. J. 1984. *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software, Belmont, CA, USA.

[2] QUINLAN, J. R. 1986. Induction of decision trees. *Machine Learning* 1, 1 (Mar. 1986), 81–106. DOI:https://doi.org/10.1007/BF00116251

[3] QUINLAN, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[4] QUINLAN, J. R. *C5.0: An Informal Tutorial*. RuleQuest Research. (Retrieved October 23, 2025). https://www.rulequest.com/see5-unix.html

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. https://archive.ics.uci.edu/ml

Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Sciences. Retrieved from https://archive.ics.uci.edu/dataset/2/adult

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.

Scikit-learn Developers. (2025, October 17). Cost Complexity Pruning Example. https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html

Scikit-learn Developers. (n.d.). Decision Trees. scikit-learn documentation. Retrieved from https://scikit-learn.org/stable/modules/tree.html

D. Pettersson and O. Nordander (svaante). (n.d.). decision-tree-id3: ID3 Decision Tree Classifier. GitHub repository. Retrieved from https://github.com/svaante/decision-tree-id3

Breiman,L.,Friedman,J.H.,Olshen,R.A.,&Stone,C.J.(1984).Classification and Regression Trees. Wadsworth International Group.

Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning,1(1), 81–106.

Quinlan, J. R. (1996). Improved Use of Continuous Attributes in C4.5.

Journal of Artificial Intelligence Research, 4, 77–90.

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and

Techniques (3rd ed.). Morgan Kaufmann.

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python.

Journal of Machine Learning Research, 12, 2825–2830.