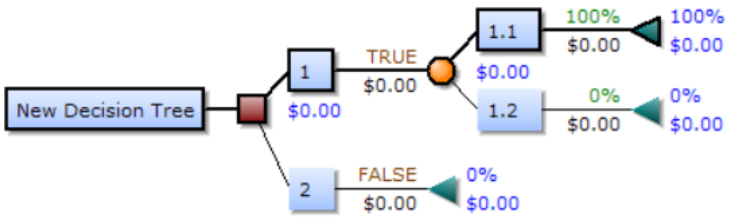
決策樹（Decision Tree）

一、 原理

決策樹是一個預測模型，它代表的是特徵（feature）與欲預測的對象（label）之間的一種映射關係。樹中每個節點表示某個特徵，而每個分叉路徑則代表具有某個特徵的屬性值，而每個葉節點則對應從根節點到該葉節點所經歷的路徑所表示的所有特徵的值。

一個決策樹包含三種類型的節點：

1. 根節點（用矩形框表示）：根節點包含所有的訓練數據。
2. 內部節點(用圓圈表示)：每一個內部節點依據所選擇的特徵去分割節點，直到達到停止條件。
3. 葉節點(用三角形表示)：對應從根節點到該葉節點所經歷的路徑所表示的所有特徵的值。

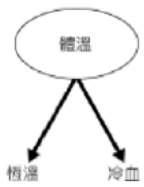


建立決策樹:

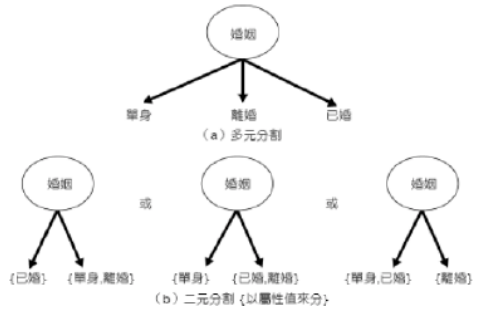
1. 找出最能將資料點均勻區分的問題作為樹的內部節點，並將節點分割以產生對應的分支。
2. 在每一個葉節點重複 Step 1，直到達到停止條件。

不同類型的節點分割:

1. 二元屬性:
   - 二元屬性的資料只有兩個不同的值。
   - 分割後會產生兩個不同方向的分支。

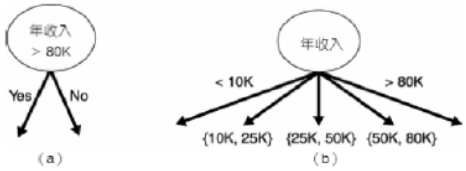


2. 名目屬性
   - 名目屬性的資料沒有前後次序關係。
   - 可分割出不同值域的分支。



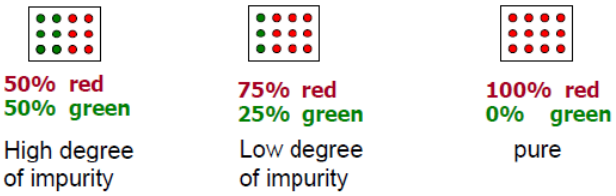
3. 順序屬性
   - 順序屬性的資料有前後次序關係。
   - 可產生出二元或是多元分割，順序屬性內的值可以被群組，但是在群組時比須沒有違反屬性值的順序。

4. 連續性屬性
   - 可採用離散化的方式，將資料分割成許多區間，但是仍需要保持資料的順序性。



如何決定分割結果何者較佳:

1. 分割結果中，若具有較高同質性（Homogeneous）類別的節點，則該分割結果愈佳。
2. 需檢驗每個內部節點的不純度（Node Impurity），不純度愈低愈好。



不純度（Node Impurity）檢驗的常見指標:

1. 獲利資訊（information gain，ID3）

- 熵: 資訊量的凌亂程度，當熵值愈大，代表資訊的凌亂程度愈高。

  如果資料集合 S具有 c 個不同的類別，那麼資料集合 S 的熵值計算方式為：

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

  其中p_i為類別 i 在資料集合 S 出現的機率

  舉例來說，給定一組丟銅板後之資料集合 S，該組資料的熵值計算公式:

  $$\text{Entropy(S)}=-p_+\log_2 p_+ - p_-\log_2 p_-$$

  若丟了 14 次銅板，出現了 9 個正面與 5 個反面，則熵為 0.94；若銅板丟出正面與反面的數量是一樣，則熵為 1(最凌亂)；若都丟出正面或反面，則熵為 0(最不凌亂)。

- 特徵 A 在資料集合 S 的資訊獲利 Gain(S,A)被定義為:

  $$Gain(S,A) = Entropy(S) - \sum_{j=1}^{v} \frac{|S_j|}{|S|} Entropy(S_j)$$

  - **假設屬性 A 中有 v 個不同值 $\{a_1, a_2, \ldots, a_v\}$，而資料集合 S 會因為這些不同值而產生(分割)出 v 個不同的資料子集合 $\{S_1, S_2, \ldots, S_v\}$**
  - $Entropy(S)$：**資料集合 S 整體的亂度**
  - $Entropy(S_j)$：**資料子集合 S_j 的亂度，其中 $j = 1, 2, \ldots, v$**
  - $\frac{|S_j|}{|S|}$：**第 j 個子集合之資料個數佔總資料集合的比率 (即：權重)**
  - $\sum_{j=1}^{v} \frac{|S_j|}{|S|} Entropy(S_j)$：**依據屬性A來判定資料集合S的亂度**
  - $Gain(S,A)$：**利用屬性A對資料集合S進行分割的獲利**
    - **Gain值愈大，表示屬性A內資料的凌亂程度愈小，用來分類資料會愈佳**
    - **Gain值愈小，表示屬性A內資料的凌亂程度愈大，用來分類資料會愈差**

2. 獲利比例（Gain ratio，C4.5):

- ID3 演算法所使用的資訊獲利會傾向選擇擁有許多不同數值的特徵，例如「產品編號」欄位中，每一個產品的產品編號都不同，若依產品編號進行分割，會產生許多分支，且每一個分支都是很單一的結果，其資訊獲利會最大，但這個特徵對於建立決策樹是沒有意義的。
- C4.5 演算法克服這個問題(正規化)。計算特徵 A 的獲利比率時，除了資獲利外，尚需計算該屬性的分割資訊值(Split Information):

  $$SplitInfo_A(S) = -\sum_{j=1}^{v} \frac{|S_j|}{|S|} \times \log_2 (\frac{|S_j|}{|S|})$$

- 獲利比率:

$$\mathbf{GainRatio(A) = Gain(S, A)/SplitInfo_A(S)}$$

擁有最大獲利比率的屬性被設為分割屬性

3. 吉尼係數 (Gini Index，CART):
   - 每一個節點都是採用二分法
   - Gini(S):

$$Gini(S) = 1 - \sum_{j=1}^{n} p_j^2$$

p_j為在S中的值組屬於類別j的機率

   - $Gini_A(S)$:

利用屬性A分割資料集合 S 為 S_1 與 S_2 (二元分割)。則根據此一分割要件的吉尼係數$Gini_A(S)$為

$$Gini_A(S) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2)$$

其中， S_1 與 S_2 是針對欄位 A 內的不同數值所構成的兩組資料子集合。

   - 不純度的降低值:

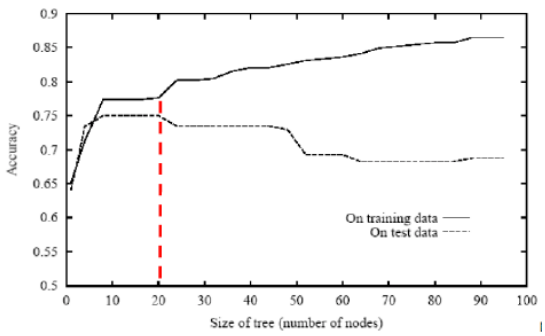
$$\Delta Gini(A) = Gini(S) - Gini_A(S)$$

   - 挑選擁有最大不純度的降低值，或吉尼係數$Gini_A(S)$最小的屬性作為分割屬性。

過度配適 (overfitting)

決策樹學習可能遭遇模型過度配適(overfitting) 的問題，過度配適是指模型對於資料的過度訓練，導致模型記住的不是資料的一般特性，反而是資料的局部特性。對預測樣本的分類將會變得很不精確。

過度適配訓練資料：



1. 存在雜訊或離異值
2. 訓練資料的數量太少，使得某一些屬性恰巧可以很好地分割目前的訓練範例，但卻與實際的狀況並無太多關係。
3. 樹的階層過多，分類的屬性過多，把資料分類的過細，即便可以把訓練資料精確分類，但對預測樣本的分類會變得很複雜且不精確。

修剪，減少樹狀結構

對離散屬性特徵改採用變異數的增減做判斷

二、 參考資料

1. 決策樹 (https://zh.wikipedia.org/wiki/%E5%86%B3%E7%AD%96%E6%A0%91)
2. Class Handout, Lee, Chia-Jung professor, MDM64001, School of Big Data Management, Soochow University