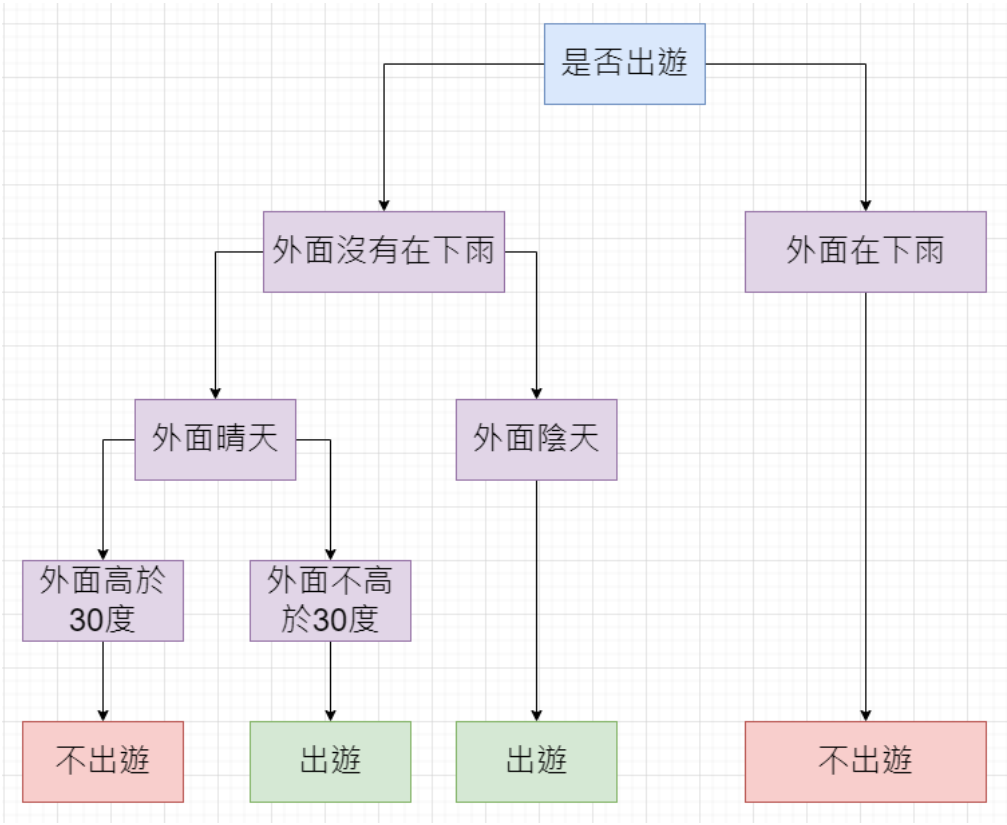
Decision Tree（決策樹）

> 主要參考資料：
>
> 決策樹 (Decision Tree) Dr. Tun-Wen Pai
>
> Business intelligence and data mining Anil K. Maheshwari, Ph.D.

決策樹

決策樹用來對於一個樹狀事件給出一條路線引導出一個決策，例如是否借款，或者更複雜的決策系統。

舉一個生活化的例子，若要決定是否要出遊，則我們可以畫出以下的樹狀事件。



可以發現從「是否出遊」到任意一種出遊決策都是唯一路徑，換言之，在這棵樹上共有 4 條路徑。

對於一個決策系統來說也可能會很複雜，例如判斷手寫數字時，龐大而精確的決策系統能夠有效的幫助我們判斷出手寫的數字。

使用決策樹的優點與缺點

> Reference website:

> 1. https://scikit-learn.org/stable/modules/tree.html
> 2. https://zh.wikipedia.org/wiki/%E5%86%B3%E7%AD%96%E6%A0%91%E5%AD%A6%E4%B9%A0

- 優點
  1. 樹可以被視覺化，方便理解
  2. 資料需要被整理，資料的空值不被接受
  3. 任何作出決策的操作都取決於樹的節點數量，且為對數時間複雜度
  4. 支援輸出一種以上的結果（multi-output）
  5. 使用白箱模型，因為決策樹可以簡單解釋決策的來源與邏輯
  6. 可用統計測試來驗證模型
  7. 對於噪聲有很強的控制性

- 缺點
  1. 可能會 overfitting
  2. 因為可能會 overfitting，所以不好實行外推
  3. 優化決策樹是 NP-Complete 問題，優化決策樹無法在多項式時間內解決
  4. 有些概念沒有辦法清楚解釋（例如 XOR）
  5. 資料集的平衡很重要，否則樹可能會被一些資料所支配，產生出帶有偏見的決策樹

建立決策樹的方式

我們以一個出遊的資料集為例，如下圖。

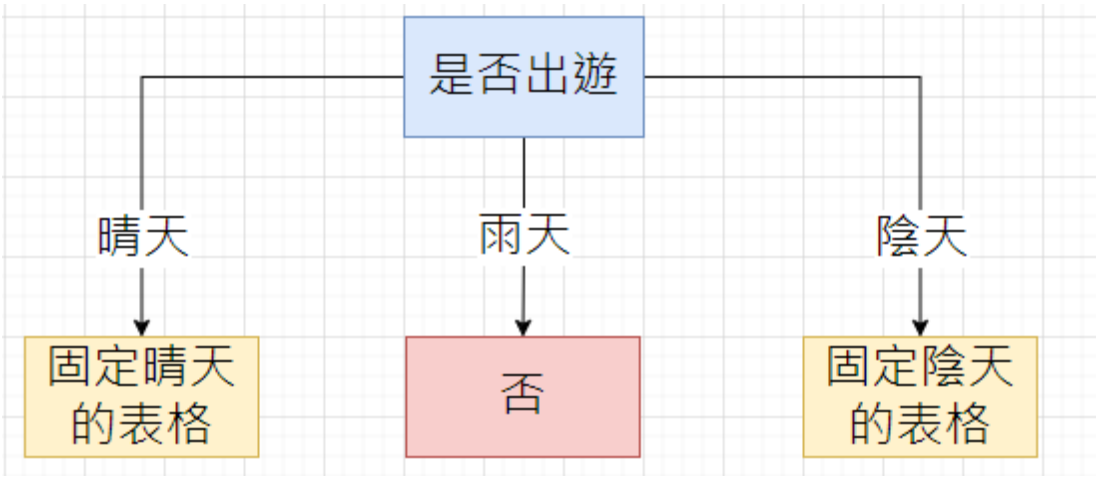
天氣	溫度	天	出遊與否
晴天	熱	工作日	否
晴天	熱	假日	否
晴天	冷	工作日	否
晴天	冷	假日	是
晴天	溫和	工作日	否
晴天	溫和	假日	是
陰天	熱	工作日	否
陰天	熱	假日	否
陰天	冷	工作日	否
陰天	冷	假日	是
陰天	溫和	工作日	否
陰天	溫和	假日	是
雨天	熱	工作日	否
雨天	熱	假日	否
雨天	冷	工作日	否
雨天	冷	假日	是
雨天	溫和	工作日	否
雨天	溫和	假日	是

決策樹是一種層級式的分支結構，我們可以分割表格，來計算錯誤合計。

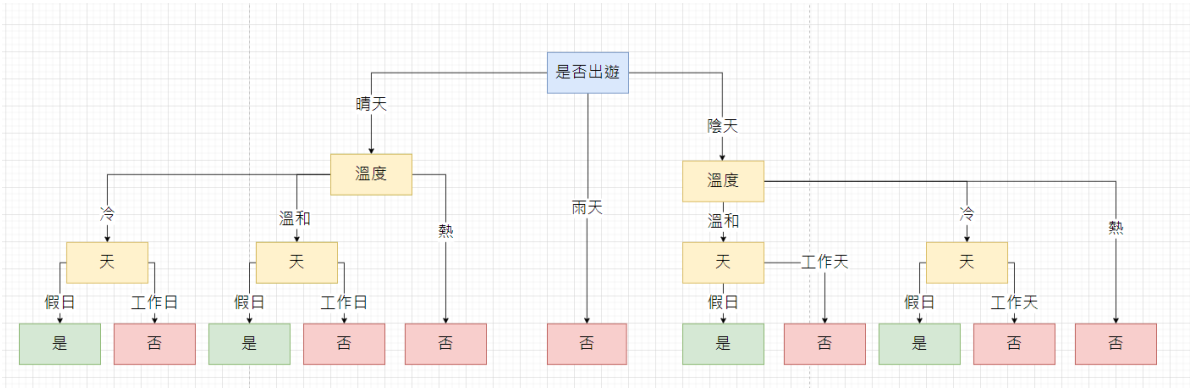
我們將現有資訊加上規則，可以得到以下的表格，可以看出錯誤統計均是 4/18，因此我們可以任意選擇一個來建樹。

屬性	規則	錯誤	錯誤總和
天氣	晴天->是	2/6	4/18
	陰天->是	2/6	
	雨天->否	0/6	
溫度	熱->否	0/6	4/18
	溫和->是	2/6	
	冷->是	2/6	
天	假日->是	4/9	4/18
	工作日->否	0/9	

例如我們選擇天氣，可以得到以下的事件樹狀圖。



按照同樣方式進行推廣，可以得到以下的樹狀結構。



這樣有點智障，因為晴天跟陰天應該可以合併，且冷跟溫和也該可以合併。

一個決策樹主要基於分支準則、停止條件與修剪而有所不同，也因此衍伸出了決策樹演算法。

決策樹的 Overfitting 問題

決策樹其中一個問題就是很容易 overfitting。

Overfitting 簡介

Overfitting 就是過度擬合，也就是對於一個訓練資料集，只對資料集的資料有作用，不利於推廣更多資料。

如果看不懂上面的文字，可以看下面找到的一張圖。

由於只對資料集的資料有作用，因此若在預判不是資料集的資料時，很容易出現預判失敗的問題。

迴避 Overfitting 的方法

迴避 Overfitting 可以使用剪枝來迴避，分成先剪枝（prepruning）與後剪枝（postpruning）。

1. 先剪枝：可以設定一個條件，使得後面的子樹停止構建，當前的節點變成葉節點。
2. 後剪枝：先建照一個完整的決策樹，再將這個樹進行修剪。

當然，資料集也很重要，所以如同 data mining 一樣，需要對資料集的品質有所把持。

決策樹演算法

主要分成：ID3、CART、CHAID，這份筆記主要會講解 ID3、CART 演算法。

ID3 演算法

Entropy

Entropy（資訊熵），一種量化資料同源的數值，介於 0 ~ 1 之間。

若一個資料是絕對同源，則 Entropy = 0，若一個資料絕對異源，則 Entropy = 1。

定義一個樣本的 Entropy 為：

$$Entropy(S) = -\sum_i P(x_i)\log_2(P(x_i)) - \sum_i Q(x_i)\log_2(Q(x_i))$$

其中 $\lim_{p\to 0} p\log p = 0$。

Information Gain

Information Gain（訊息增益）為資訊熵經過分割 Attribute 後所減少的數值，介於 0 ~ 1 之間。

當我們在分割樹的時候，我們會選擇 Infomation Gain 大的來當作我們的父節點或根節點，若 Entropy = 0 時則為子節點。

我們可以定義一個樣本在分割成一個 Attribute 之後的 Infomation Gain 為

$$IG(S,A) = Entropy(S) - \sum_{v \in D_A} \frac{|S_v|}{|S|} Entropy(S_v)$$

優缺點

- 優點：
  1. 好理解決策的邏輯，因為可以畫成一棵樹
  2. 建立迅速
  3. 建立較小的樹
  4. 只需要足夠數量的測試資料
  5. 只需要測試足夠多的屬性來讓所有資料都被分類
  6. 在分類測試資料時可以被修剪，利於減少測試數量
  7. 使用整個資料集，搜索空間完整
- 缺點
  1. 會因為測試資料過小導致 over-fitted 跟 over-classified，不利於推廣預測。
  2. 每次預測只測試一個 Attribute。
  3. 測試連續資料可能會導致大量運算，產生大量的樹。

範例

以這個資料為範例，利用 ID3 來建立決策樹。

outlook	temperature	humidity	wind	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cold	normal	FALSE	yes
rainy	cold	normal	TRUE	no
overcast	cold	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cold	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

首先先計算 $Entropy(Play)$，也就是

$$Entropy(Play) = -\frac{9}{14}\log_2\frac{9}{14} - (\frac{5}{14}\log_2\frac{5}{14}) \approx 0.94$$

接著可以考慮

1. 三種不同類型的 Outlook 所產生出來的 $Entropy(Play, Outlook)$

   $Entropy(Sunny) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) \approx 0.97$

   $Entropy(Overcast) = -\frac{4}{4}\log_2(\frac{4}{4}) - \frac{0}{4}\log_2(\frac{0}{4}) = 0$

   $Entropy(Rainy) = -\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5}) \approx 0.97$

   $Entropy(Play, Outlook) = \frac{5}{14}Entropy(Sunny) + \frac{4}{14}Entropy(Overcast) + \frac{5}{14}Entropy(Rainy) = 0.6936$

2. 三種不同類型的 Temperature 所產生出來的 $Entropy(Play, Temperature)$

   $Entropy(Hot) = -\frac{2}{4}\log_2(\frac{2}{4}) - \frac{2}{4}\log_2(\frac{2}{4}) = 1$

   $Entropy(Mild) = -\frac{2}{6}\log_2(\frac{2}{6}) - \frac{4}{6}\log_2(\frac{4}{6}) \approx 0.92$

   $Entropy(Cold) = -\frac{3}{4}\log_2(\frac{3}{4}) - \frac{1}{4}\log_2(\frac{1}{4}) \approx 0.81$

   $Entropy(Play, Temperature) \approx 0.911$

3. 兩種不同類型的 Humidity 所產生出來的 $Entropy(Play, Humidity)$

   $Entropy(High) = -\frac{3}{7}\log_2(\frac{3}{7}) - \frac{4}{7}\log_2(\frac{4}{7}) \approx 0.985$

   $Entropy(Normal) = -\frac{6}{7}\log_2(\frac{6}{7}) - \frac{6}{7}\log_2(\frac{6}{7}) \approx 0.592$

   $Entropy(Play, Humidity) \approx 0.7885$

4. 兩種不同類型的 Wind 所產生出來的 $Entropy(Play, Wind)$

   $Entropy(True) = -\frac{3}{6}\log_2(\frac{3}{6}) - \frac{3}{6}\log_2(\frac{3}{6}) = 1$

   $Entropy(False) = -\frac{6}{8}\log_2(\frac{6}{8}) - \frac{2}{8}\log_2(\frac{2}{8}) \approx 0.811$

   $Entropy(Play, Wind) \approx \frac{6}{14} \times 1 + \frac{8}{14} \times 0.811 \approx 0.892$

我們可以分別算出，分支這四種類型所產生的 $InformationGain$，得到

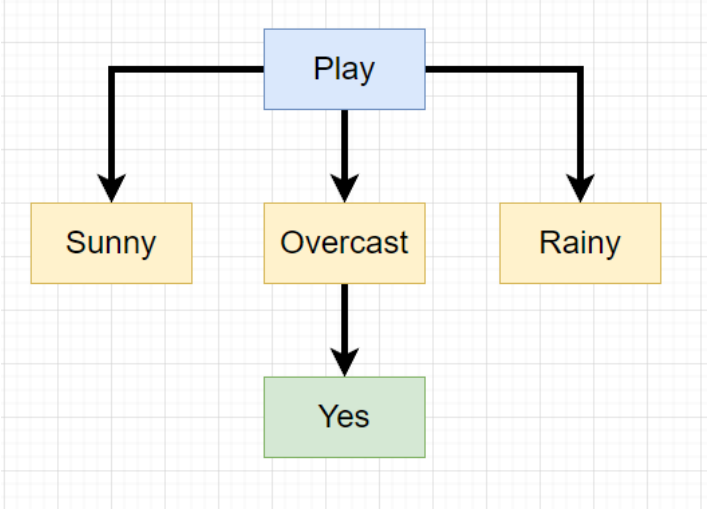
$IG(Outlook) = Entropy(Play) - Entropy(Play, Outlook) = 0.2464$

$IG(Temperature) = Entropy(Play) - Entropy(Play, Temperature) \approx 0.029$

$IG(Humidity) = Entropy(Play) - Entropy(Play, Humidity) \approx 0.1515$

$IG(Wind) = Entropy(Play) - Entropy(Play, Wind) \approx 0.048$

我們挑 IG 最高的當作分支條件，所以 $Outlook$ 先分支，如圖。



接著我們考慮 Sunny，得到

1. 三種不同類型的 temperature。

   $Entropy(Hot) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$。

   $Entropy(Mild) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

   $Entropy(Cold) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$

   可以得到 $Entropy(Sunny, Temperature) = \sum_i P(x_i)Entropy(x_i) = \frac{2}{5}$

2. 兩種不同類型的 humidity。

   $Entropy(High) = -\frac{0}{3}\log_2\frac{0}{3} - \frac{3}{3}\log_2\frac{3}{3} = 0$

   $Entropy(Normal) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$

   可以得到 $Entropy(Sunny, Humidity) = \sum_i P(x_i)Entropy(x_i) = 0$

3. 兩種不同類型的 wind。

   $Entropy(True) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

   $Entropy(False) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$

   可以得到 $Entropy(Sunny, Wind) = \sum_i P(x_i)Entropy(x_i) = 0.9508$

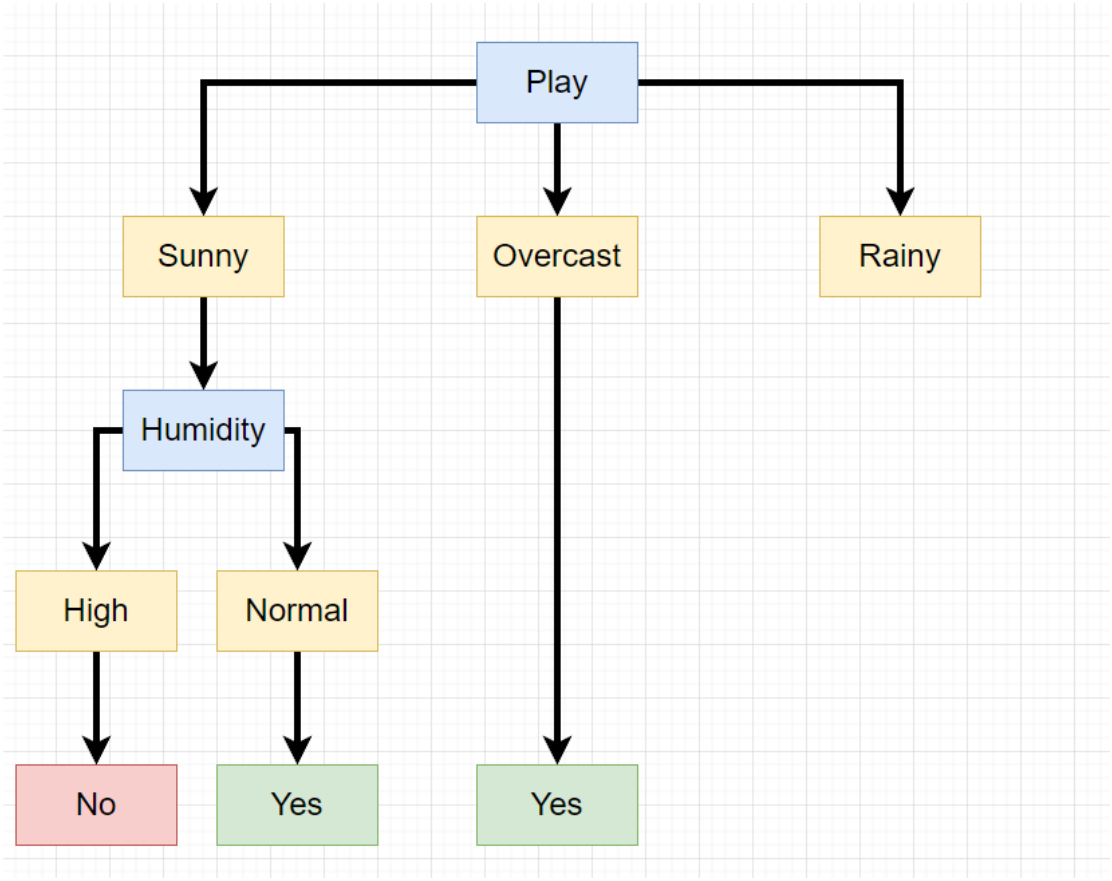
可以分別算出這三種分支的 Infomation Gain，也就是

$IG(Temperature) = Entropy(Sunny) - Entropy(Sunny, Temperature) = 0.57$

$IG(Humidity) = Entropy(Sunny) - Entropy(Sunny, Humidity) = 0.97$

$IG(Wind) = Entropy(Sunny) - Entropy(Sunny, Wind) = 0.0192$

選擇 Infomation Gain 最高的來分支，得到下圖。



接著我們考慮 Rainy，得到

1. 兩種不同類型的 temperature。

   $Entropy(Mild) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$

   $Entropy(Cold) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

   $Entropy(Rainy, Temperature) = \frac{3}{5}Entropy(Mild) + \frac{2}{5}Entropy(Cold) = 0.951$

2. 兩種不同類型的 humidity。

   $Entropy(High) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

   $Entropy(Normal) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$

   $Entropy(Rainy, Humidity) = \frac{2}{5}Entropy(High) + \frac{3}{5}Entropy(Normal) = 0.951$

3. 兩種不同類型的 wind。

   $Entropy(True) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$

   $Entropy(False) = -\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3} = 0$

   $Entropy(Rainy, Wind) = \frac{2}{5}Entropy(True) + \frac{3}{5}Entropy(False) = 0$

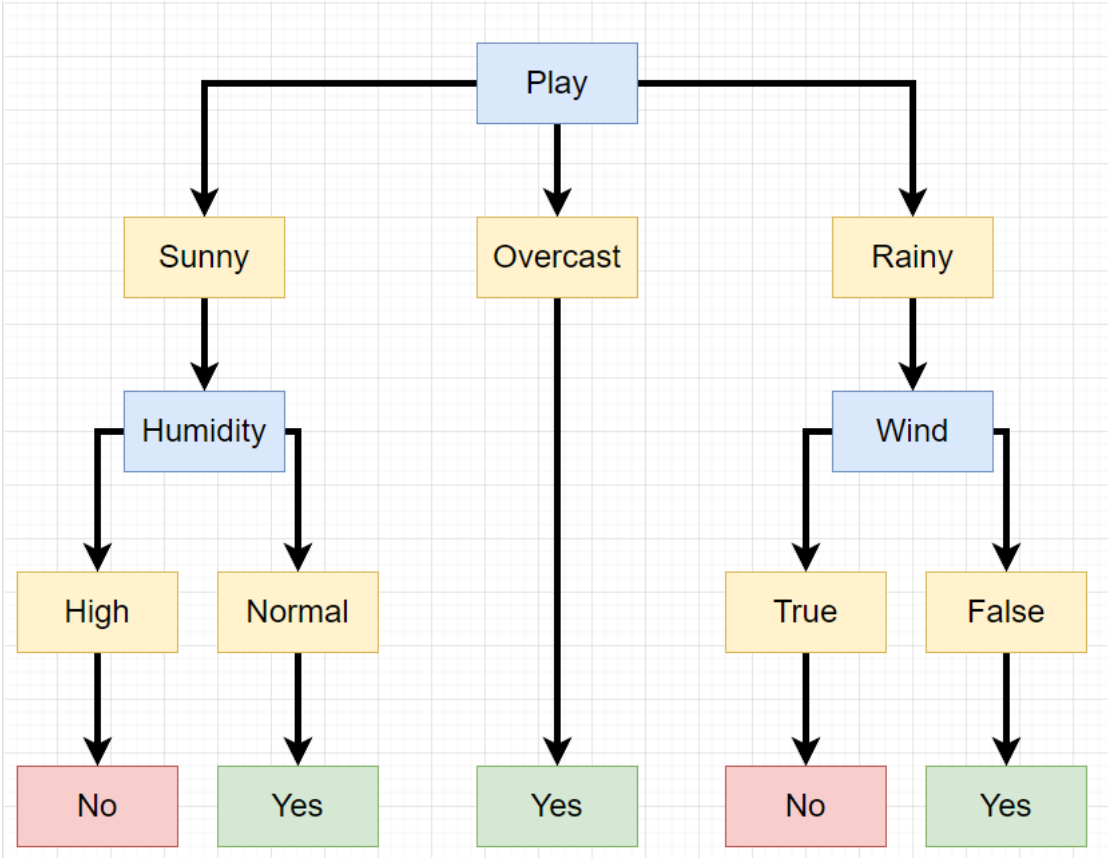
可以分別算出這三種分支的 Information Gain，如下：

$IG(Temperature) = Entropy(Rainy) - Entropy(Rainy, Temperature) = 0.97 - 0.951 = 0.019$

$IG(Humidity) = Entropy(Rainy) - Entropy(Rainy, Humidity) = 0.97 - 0.951 = 0.019$

$IG(Wind) = Entropy(Rainy), Entropy(Rainy, Wind) = 0.97 - 0 = 0.97$

選擇 Information Gain 大的當作分支，得到下圖。



此時決策樹已建立完成。

CART 演算法

> Reference:
>
> 1. https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0098450.s002&type=supplementary
> 2. https://www.youtube.com/watch?v=qrDzZMRm_Kw

演算法

一個基於吉尼不純度係數 (Gini Impurity) 作為分割標準的分類演算法，先前提到的 ID3 是基於 Information Gain。

演算法主要運行如下：

1. 尋找最佳特徵分割方式，例如有 K 種特徵，必有 K-1 種分割方式，對於每一種分割方式算出吉尼不純度係數。
2. 延續第一點，對於每一種特徵，算出每個特徵的吉尼不純度係數權重。
3. 選擇最低的吉尼不純度係數進行分割。
4. 重複1, 2, 3，直到達到結束條件。

Gini Impurity

對於一個 CART 演算法的分割標準，主要以吉尼不純度係數（Gini Impurity）來做參考標準，定義如下。

$$GI(A) = 1 - \sum_{k=1}^{m} p_k^2$$

其中 $0 \leq GI(A) \leq 1 - \frac{1}{m}$，當 $GI(A) = 0$ 時，所有資料都被歸類在同一類，$GI(A) = 1 - \frac{1}{m}$ 時所有類別均不分類。

範例

以這個資料為範例，利用 CART 來建立決策樹。

outlook	temperature	humidity	wind	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cold	normal	FALSE	yes
rainy	cold	normal	TRUE	no
overcast	cold	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cold	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

先考慮四種不同的分割方式

1. 考慮第一種 Outlook

   $GI(Outlook, Sunny) = 1 - (\frac{2}{5})^2 - (\frac{3}{5})^2 = 0.48$

   $GI(Outlook, Overcast) = 1 - (\frac{4}{4})^2 - (\frac{0}{4})^2 = 0$

   $GI(Outlook, Rainy) = 1 - (\frac{3}{5})^2 - (\frac{2}{5})^2 = 0.48$

   $GI(Outlook) = P(Sunny)GI(Sunny) + P(Overcast)GI(Overcast) + P(Rainy)GI(Rainy)$

   $= \frac{5}{14} \times 0.48 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.48 \approx 0.343$

2. 考慮第二種 Temperature

   $GI(Temperature, Hot) = 1 - (\frac{2}{4})^2 - (\frac{2}{4})^2 = 0.5$

   $GI(Temperature, Mild) = 1 - (\frac{4}{6})^2 - (\frac{2}{6})^2 \approx 0.44$

   $GI(Temperature, Cold) = 1 - (\frac{3}{4})^2 - (\frac{1}{4})^2 = 0.375$

   $GI(Temperature) = \frac{4}{14} \times 0.5 + \frac{6}{14} \times 0.44 + \frac{4}{14} \times 0.375 \approx 0.439$

3. 考慮第三種 Humidity

$$GI(Humidity, High) = 1 - (\frac{3}{7})^2 - (\frac{4}{7})^2 \approx 0.49$$

$$GI(Humidity, Medium) = 1 - (\frac{6}{7})^2 - (\frac{1}{7})^2 \approx 0.245$$

$$GI(Humidity) = \frac{7}{14} \times 0.49 + \frac{7}{14} \times 0.245 = 0.3675$$

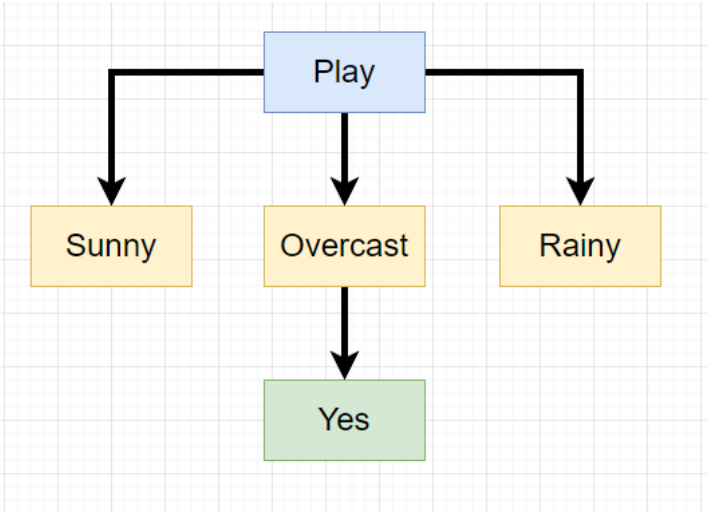
4. 考慮第四種 Wind

$$GI(Wind, True) = 1 - (\frac{3}{6})^2 - (\frac{3}{6})^2 = 0.5$$

$$GI(Wind, False) = 1 - (\frac{6}{8})^2 - (\frac{2}{8})^2 = 0.375$$

$$GI(Wind) = \frac{6}{14} \times 0.5 + \frac{8}{14} \times 0.375 \approx 0.429$$

選擇最低的 GI 來當作分割標準，故選擇 Outlook 來當作分割節點。



考慮固定 Sunny，計算剩餘的 Temperature, Humidity, Wind。

1. 考慮 Temperature

$$GI(Temperature, Hot) = 1 - (\frac{0}{2})^2 - (\frac{2}{2})^2 = 0$$

$$GI(Temperature, Mild) = 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$$

$$GI(Temperature, Cold) = 1 - (\frac{1}{1})^2 - (\frac{0}{1})^2 = 0$$

$$GI(Temperature) = \frac{2}{5} \times 0 + \frac{2}{5} \times 0.5 + \frac{1}{5} \times 0 = 0.2$$

2. 考慮 Humidity

$$GI(Humidity, High) = 1 - (\frac{0}{3})^2 - (\frac{3}{3})^2 = 0$$

$$GI(Humidity, Normal) = 1 - (\frac{2}{2})^2 - (\frac{0}{2})^2 = 0$$

$$GI(Humidity) = \frac{3}{5} \times 0 + \frac{2}{5} \times 0 = 0$$

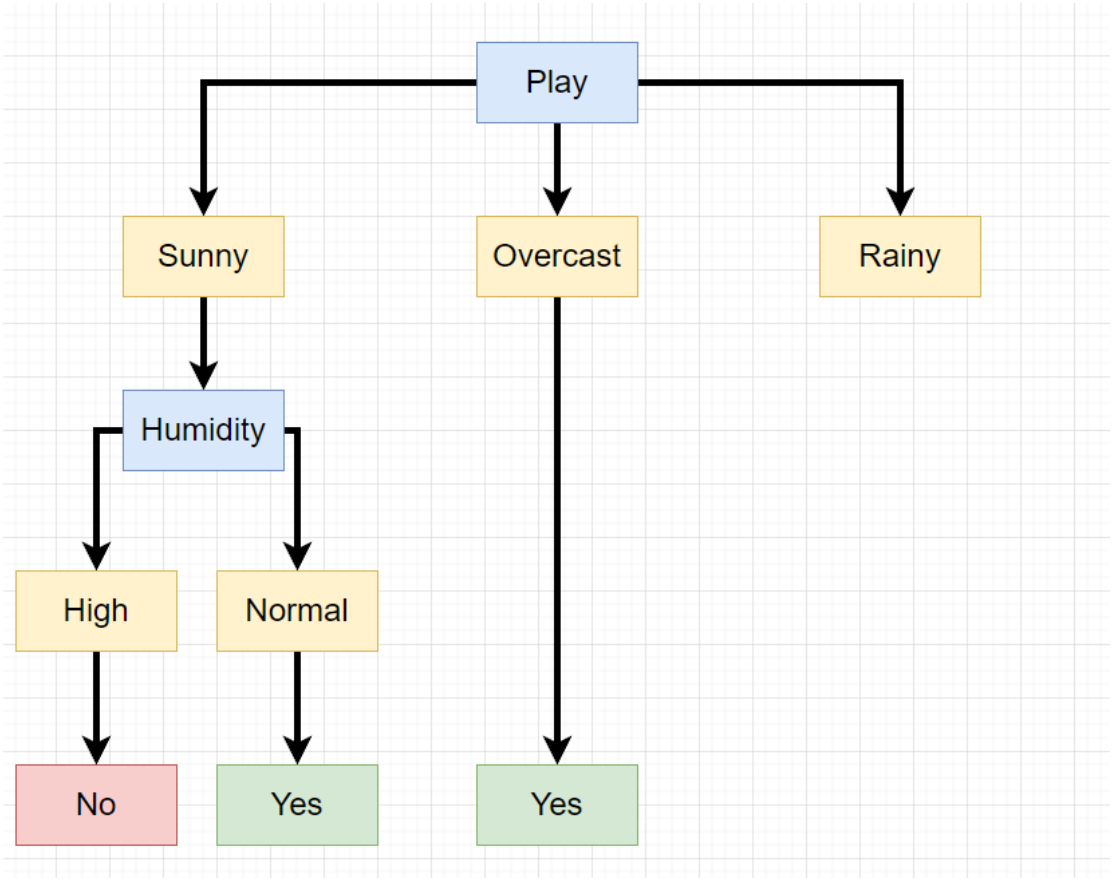
3. 考慮 Wind

$$GI(Wind, True) = 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$$

$$GI(Wind, False) = 1 - (\frac{1}{3})^2 - (\frac{2}{3})^2 \approx 0.444$$

$$GI(Wind) = \frac{2}{5} \times 0.5 + \frac{3}{5} \times 0.444 \approx 0.466$$

選擇最低的 GI 來當作分割標準，故選擇 Humidity 當作分割節點。



考慮固定 Rainy，計算剩餘的 Temperature, Humidity, Wind。

1. 考慮 Temperature

$$GI(Temperature, Mild) = 1 - (\frac{2}{3})^2 - (\frac{1}{3})^2 \approx 0.444$$

$$GI(Temperature, Cold) = 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$$

$$GI(Temperature) = \frac{3}{5} \times 0.444 + \frac{2}{5} \times 0.5 = 0.4664$$

2. 考慮 Humidity

$$GI(Humidity, High) = 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$$

$$GI(Humidity, Normal) = 1 - (\frac{2}{3})^2 - (\frac{1}{3})^2 \approx 0.444$$

$$GI(Humidity) = \frac{3}{5} \times 0.444 + \frac{2}{5} \times 0.5 = 0.4664$$

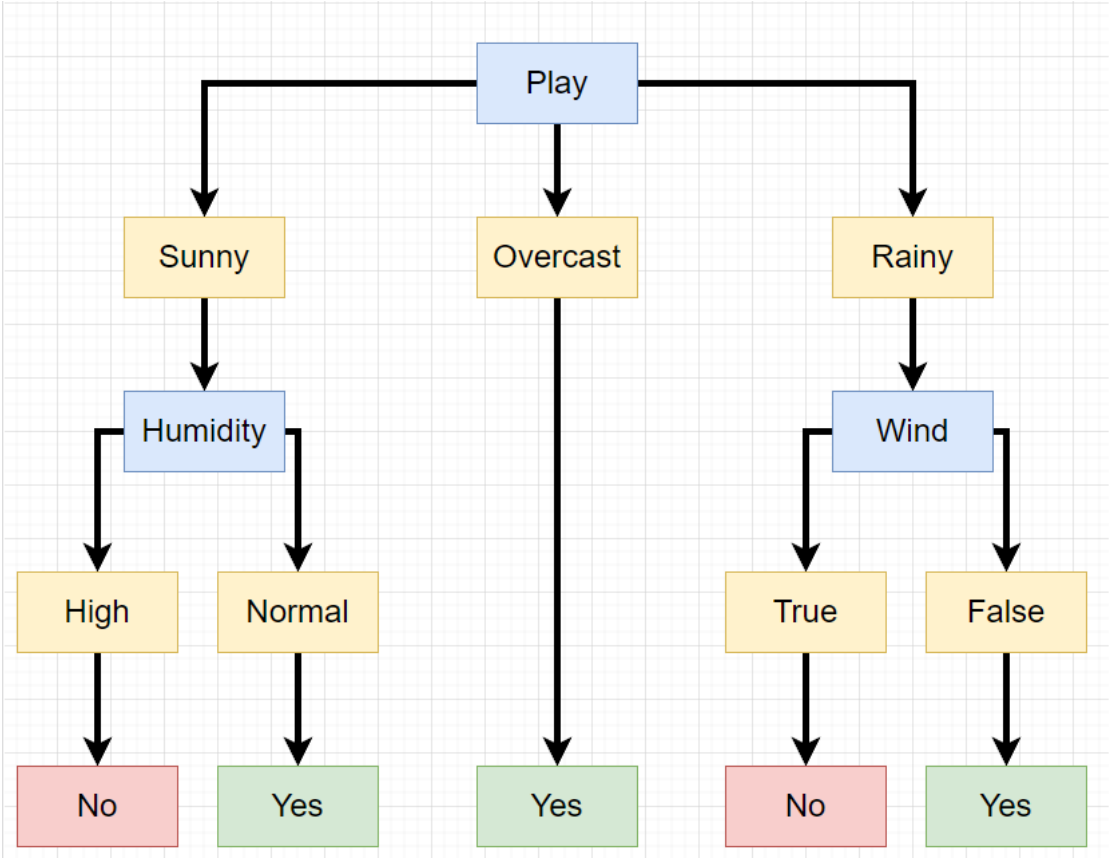
3. 考慮 Wind

$$GI(Wind, True) = 1 - (\frac{0}{2})^2 - (\frac{2}{2})^2 = 0$$

$$GI(Wind, False) = 1 - (\frac{3}{3})^2 - (\frac{0}{3})^2 = 0$$

$$GI(Wind) = \frac{2}{5} \times 0 + \frac{3}{5} \times 0 = 0$$

選擇最低的 GI 來當作分割標準，故選擇 Wind 當作分割節點。



此時分割已完成。

迴歸樹

迴歸樹是決策樹的一種種類，用來預判在某一種情況下時，能夠對應到的效果。

輸出值為連續變數，運作方式與分類樹較相同，與分類樹不同的地方是：

1. 對於一個二維的座標軸來說，迴歸樹通常是預判在這個二維座標軸特定條件所形成的一個長方形區塊內資料的平均值，
2. 判斷一個長方形區塊內資料的純度，使用了殘差平方和，定義為 $RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$。
3. 使用均方根誤差來測量效能。

Cost Complexity Pruning

> Reference: https://www.youtube.com/watch?v=D0efHEJsfHo

迴歸樹運作原理與分類樹較相同，但也同時會出現 over fitting 的問題。

為了避免 over fitting 的問題發生，需要對這棵樹進行適當的剪枝，以利於資料的預判。

對於剪枝的評斷，我們需要先窮舉每一種剪枝的方式，得到剪枝序列

根據每個序列每個區塊，做一次 RSS 的測量總和，再計算一種基於 RSS 出現的 Tree Score，定義為：

$$TreeScore = RSS + aT$$

其中 a 為一微調參數，且 T 為葉節點樹量。

找到這個參數 a 的方式是透過微調 a 來使當前的樹算出的 Tree Score 變小。

再只使用測試資料來做幾次交叉測試，找出能夠使 RSS 平均最小的子樹，即為最佳剪枝樹。

Random Forest

> Reference:
>
> 1. https://www.youtube.com/watch?v=J4Wdy0Wc_xQ
> 2. https://www.youtube.com/watch?v=sQ870aTKqiM

Random Forest 是一種包含多種決策樹的分類器，輸出的類別取決於所有決策樹的結果統計出的眾數。

大致上來說，Random Forest 使用以下的演算法：

1. 建立 bootstrapped table，隨機抽取一定數量的樣本來建立。
2. 建立決策樹的父節點，使用隨機不重複的變數來建立。
3. 重複第一步與第二步，利用這樣的方式建立多棵決策樹。

接下來即可將樣本丟入隨機森林，蒐集每一棵決策樹所產生出的結果，以及找出結果眾數。

Bagging

我們將樣本隨機抽取來建立 bootstrapped table，然後利用總計來得出結果，這樣的方式稱為 Bagging。

Out-of-bag dataset

由於我們前面說到，建立 bootstrapped table 的方式是使用隨機抽樣，那麼隨機抽樣這一步有可能會出現沒有抽到的資料。

我們將這些資料蒐集起來，稱為 Out-of-bag dataset。

處理這些 Out-of-bag dataset 的方式是，我們可以把這些資料集丟回去隨機森林，來確定所有的樹是否都回傳同一個結果。

就可以用這些結果來測量隨機森林的準確度，至於 Out-of-bag dataset 分類錯誤的部分即稱為 Out-of-bag error。

我們可以測量準確度，再來校正隨機抽取樣本的數量，來校正隨機森林。

Missing Data

對於 Random Forest 的缺值問題，主要分成兩種。

1. 知道結果、但不知道資料變數的某些值。
2. 不知道結果、但也不知道資料變數的某些值。

對於這兩種問題，可以分成兩種不同的方式來解決。

缺值問題 1

我們可以先根據先前的資料集進行投票，並且去尋找相似的先前資料來設定值，再逐步調整。

逐步調整的方式可以利用 updated proximity matrix 來調整，經過多次迭代之後找到一個再迭代後也變化不大的結果。

缺值問題 2

我們可以根據剩餘的資料來進行建立隨機森林。

建立模型後，先是使用第一個缺值問題的方式來設定變數值，以及窮舉所有可能出現的結果，產生出多個候選資料。

將這些候選資料丟入模型進行預判，來得到隨機森林對於哪一種結果有更高的投票數，即可選定投票結果。