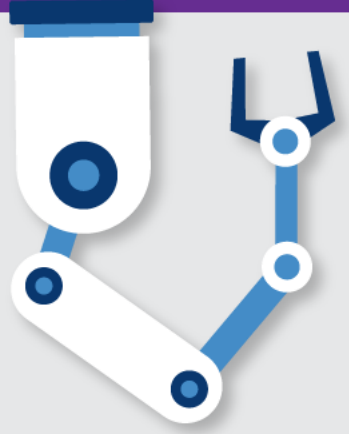
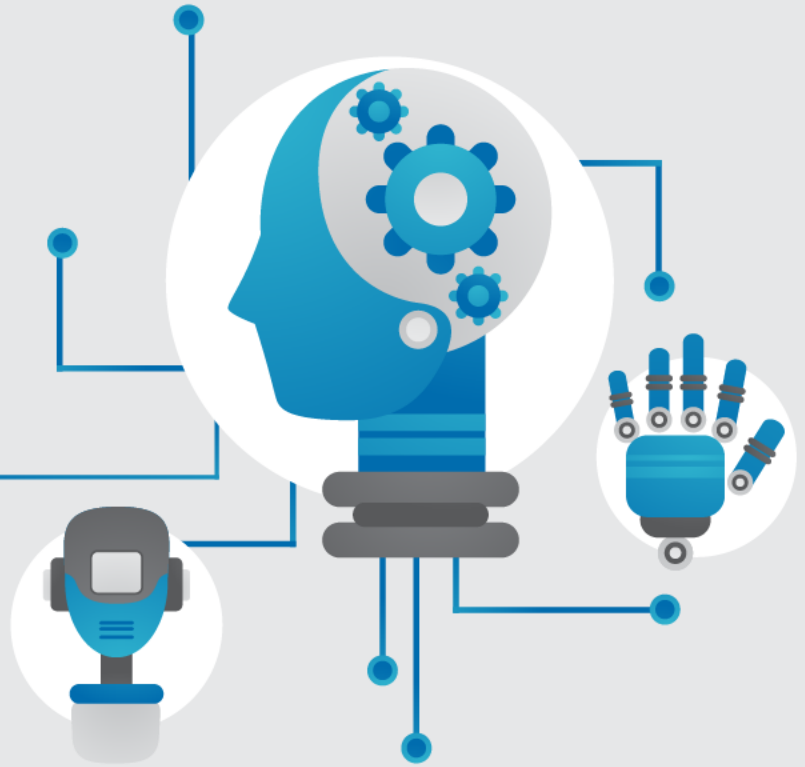


決策樹與隨機森林



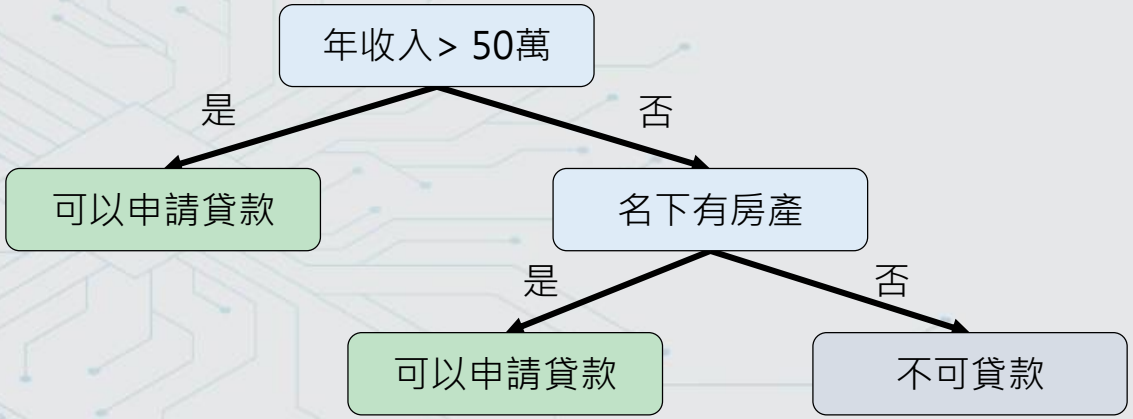
決策樹

- **決策樹（Decision Tree）**

  是用於分類（classification）和迴歸（regression）的監督式學習方法。

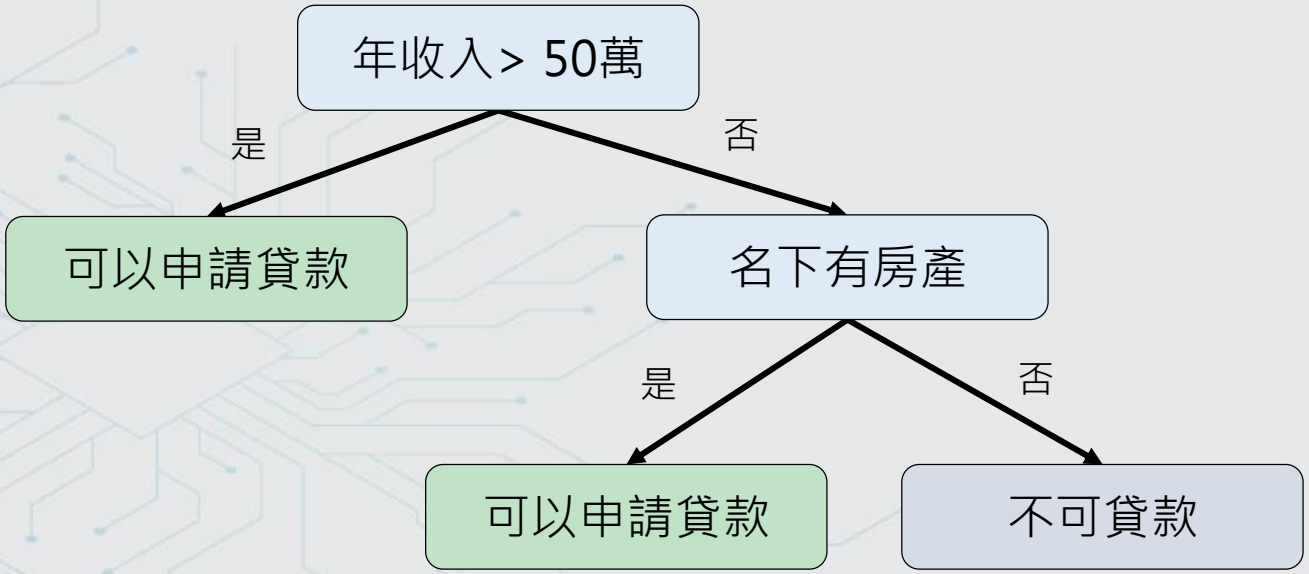
  目標是創建一個模型，通過學習從數據特徵推斷出的簡單決策規則來預測目標變量的值。

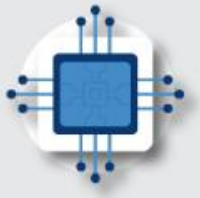
- **下面是一個是否有貸款資格的決策樹**



決策樹

- 樹的中間節點（non-leaf nodes）代表測試的條件
- 樹的分支（branches）代表條件測試的結果
- 樹的葉節點（leaf nodes）代表分類後所得到的分類標記，也就是表示分類的結果





決策樹的產生程序與用途

決策樹的產生程序

步驟1：建立樹狀結構

✓首先，所有的訓練樣本都在根節點

✓依據選取的屬性，重複地將樣本分隔開來

步驟2：修剪樹狀結構

✓辨識並且移除導致雜訊或特例的分支

決策樹的用途：分類未知的樣本

- 依據決策樹測試樣本的屬性值

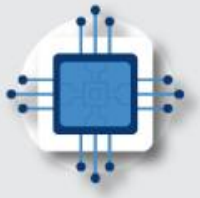
決策樹推論演算法

基本演算法

- 樹結構是以**由上而下**，**遞迴**方式建立
- 無法處理連續性的數值，數值屬性必須先轉換

運作方式

- 首先，所有的訓練樣本都在根節點
- 屬性都是類別型態（若是連續型數值，事先做離散化）
- 依據選取的屬性，反複地將樣本分隔開來
- 測試各屬性是否是以統計性測量（例如資訊獲利 information gain）為基礎而挑選出來的



決策樹推論演算法

停止分支的條件

- 當某分支子集合內的所有樣本都屬於同一個類別
- 所有的屬性都用完了，用多數投票法以樣本數較多的類別來代表此葉節點。
- 選取某屬性之後，產生的某分支完全沒有測試樣本

建構決策樹

- 若節點順序改變，同樣的資料構建出的決策樹也會不同。
- 決策樹決定節點特徵順序的 3 個常用演算法

ID3 演算法：使用資訊獲利（Information Gain）最大的特徵優先。

C4.5 演算法：使用資訊獲利比最大的特徵優先

CART 演算法：使用基尼指數（Gini）最小的特徵優先

資訊獲利、資訊獲利比和基尼指數

- **資訊獲利（Information Gain）=**資訊熵 - 條件熵
  是指在一個條件下**資訊不確定性減少**的程度。

  - 資訊熵：樣本的類別的複雜度

    $$H(X) = -\sum_{i=1}^{n} p(x_i)\, log\, p(x_i)$$

  - 條件熵：在某一特徵下，樣本的類別的複雜度

    $$H(Y|X) = \sum_{i \in X} p(x_i) H(Y|X=x)$$

資訊獲利、資訊獲利比和基尼指數

- **資訊獲利比 = 資訊獲利 / 資訊熵**

- **基尼指數（Gini）**：用來表示資料的不純度，基尼指數越大，表示資料越不平均。當資料中樣本的類別都一樣時，基尼指數為**0**。
  - 公式：**Gini（D）= 1 - $\sum_{i=1}^{n} p(x_i)^2$**，其中 D為資料集全體樣本，$p(x_i)$表示每個類別出現的機率。

範例說明

- 假設有一表示買水果標準的資料集如下

index	色澤？	重量？	價錢？	買與否？
1	漂亮	重	不貴	買
2	不漂亮	輕	貴	不買
3	漂亮	中	貴	不買
4	漂亮	輕	不貴	買
5	不漂亮	重	不貴	買

範例說明

- 資訊熵：$H(X) = -\sum_{i=1}^{n} p(x_i)\, log\, p(x_i)$
- 資料集顯示：買的有3個，p(買) = 3/5, p(不買) = 2/5

H(X) = - 3/5 log (3/5) - 2/5 log (2/5)

= 0.133 + 0.159 = 0.292

範例說明

- 條件熵： $H(Y|X) = \sum_{i \in X} p(x_i) H(Y|X=x)$
- 選擇是否漂亮作為特徵，則 {index = 1,3,5} 為漂亮，其中2個買，1個不買

$H(Y|X = \text{漂亮}) = -\frac{2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right)$

$= 0.117 + 0.159 = 0.276$

$H(Y|X = \text{不漂亮}) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)$

$= 0.151 + 0.151 = 0.302$

$P(X = \text{漂亮}) = 3/5 = 0.6$, $P(X = \text{不漂亮}) = 2/5 = 0.4$

$H(Y|X) = 0.276 \times 0.6 + 0.302 \times 0.4 = 0.2864$

範例說明

- **資訊獲利= 資訊熵 – 條件熵** = 0.292 – 0.2864 = 0.0056
- 即知道這水果是否漂亮，對於選擇買與不買的不確定度減少了0.0056
- 資訊獲利比 = 資訊獲利/ 資訊熵

  = 0.0056 / 0.292 = 0.0191

範例說明

- 基尼指數越大，表示資料越不平均

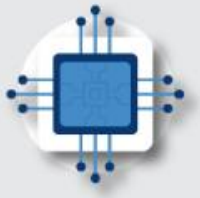
Gini（D）= 1 - $\sum_{i=1}^{n} p(x_i)^2$

p(買) = 3/5, p(不買) = 2/5

Gini（D）= 1 -（ (3/5)^2 + (2/5)^2 ）

= 1 -（0.36 + 0.08）

= 0.56



決策樹的優缺點

決策樹的優點

- 模擬人的直觀決策規則，訓練速度快

決策樹的缺點

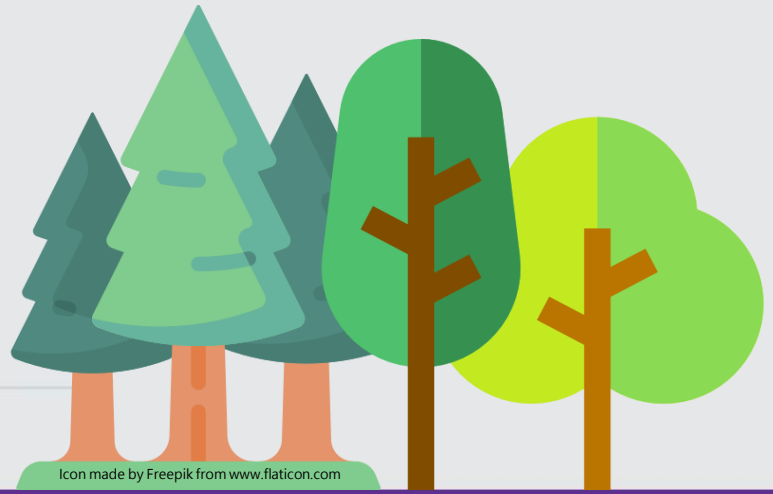
- 容易忽略資料集裡面屬性的相互關聯
- 若不進行一定的限制，如剪枝等，
  會導致模型很符合訓練資料卻不適用於新的資料，
  造成過度學習（overfitting）

隨機森林

- 隨機森林是一個包含多個決策樹的分類器，其輸出的類別是依據個別決策樹輸出的類別的眾數而定。

- 隨機森林的引入最初是由華裔美國人何天琴於1995年先提出**隨機決策森林**（random decision forests）。然後由Leo Breiman於2001年在一篇論文中提出**隨機森林**。

- 隨機森林的本質屬於機器學習的**集成學習**（Ensemble Learning）方法。



隨機森林演算流程

從資料集中(有放回的)隨機採樣，
選出n個樣本函示庫

從所有特徵中隨機選擇k個特徵，對選出的樣本，
利用這些特徵建立決策樹

重複以上步驟m次，即生成m棵決策樹，
形成隨機森林

經過每棵樹決策，最後投票決定輸出的類別

隨機森林的優缺點

優

特徵很多　　因為特徵子集是隨機選擇

- 能處理很高維度的資料，且不用做特徵選擇
- 可判斷出不同特徵之間的相互影響
- 可判斷特徵的重要程度
- 訓練速度快，容易做成並行化方法

訓練時樹與樹之間是相互獨立的

- 有採樣過程的隨機性，不容易過度學習

隨機森林的優缺點

缺

- 已被證明在某些噪音較大的分類或回歸問題上會過度學習。
- 對於有不同取值屬性的數據，取值劃分較多的屬性會對隨機森林產生較大的影響。