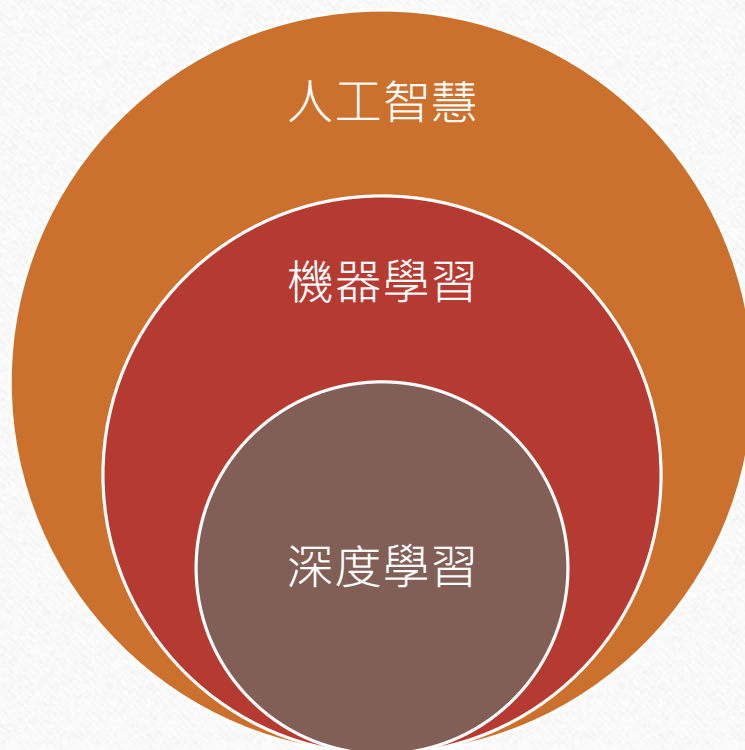


建構預測模型

名詞關係



模型分類

- 統計模型 – 統計模型為利用數學與統計學為基礎，對資料進行分類、預測等
- AI模型 – AI模型為透過提取、解析資料特徵，通過訓練資料來建立模型，進行預測、分類

統計模型

羅吉斯迴歸

- 羅吉斯迴歸模型或稱logit模型，此模型的基本形式與線性回歸(linear regression)很相似，但不須服從相依變數需為連續性、預測變數需為常態分配的假設。常用以描述多個獨立變數(independent Variables) x_i 與一個相依變數(dependent Variable) y_i 之間的關係

羅吉斯迴歸

- $y_i = \begin{cases} 1, & \text{if } y_i^* > 0 \\ 0, & \text{otherwise} \end{cases}$
- $y_i^* = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} +$
- y_i^* 表示第*i*個樣本客戶之違約機率， $i = 1, 2, \dots, k$ ；
- α 表示羅吉斯迴歸的常數項；
- x_{ij} 表示第*i*個樣本客戶之第*j*個解釋變數， $j = 1, 2, \dots, k$ ；
- β_j 表示第*j*個迴歸係數

貝氏分類器

- 此模型為運用貝氏定理為基礎的簡單機率分類器，其為根據條件機率建構的分類模型，單純貝氏自1950年代已廣泛研究，並在1960年代進入到文字分類界中，目前仍然是一種熱門的方法，如判別郵件是否為垃圾郵件、是否合法等二分類問題

貝氏分類器

- $P(C | F_1, F_2, \dots, F_n) = \frac{P(C)P(F_1, F_2, \dots, F_n | C)}{P(F_1, F_2, \dots, F_n)}$
- C ：表示類別變數
- F_i ：為特徵變數

貝氏分類器

性別	年齡	職業	月收入	會不會違約
男	21~30	學生	低	不會
男	21~30	上班族	低	不會
男	41~50	上班族	高	不會
男	31~40	待業	低	會
男	31~40	管理階層	高	不會
女	21~30	學生	低	會
女	21~30	學生	低	不會
女	21~30	待業	低	不會
女	31~40	上班族	高	會
女	41~50	管理階層	高	不會

貝氏分類器

性別	年齡	職業	月收入	預測結果
男	31~40	學生	低	會

會違約的可能： $P(\text{會}) P(\text{男}|\text{會}) P(31\sim40|\text{會}) P(\text{低}|\text{會}) = \frac{3}{10} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} = 0.0148$

不會違約的可能： $P(\text{不會}) P(\text{男}|\text{不會}) P(31\sim40|\text{不會}) P(\text{低}|\text{不會}) = \frac{7}{10} \times \frac{4}{7} \times \frac{1}{7} \times \frac{2}{7} \times \frac{4}{7} = 0.0093$

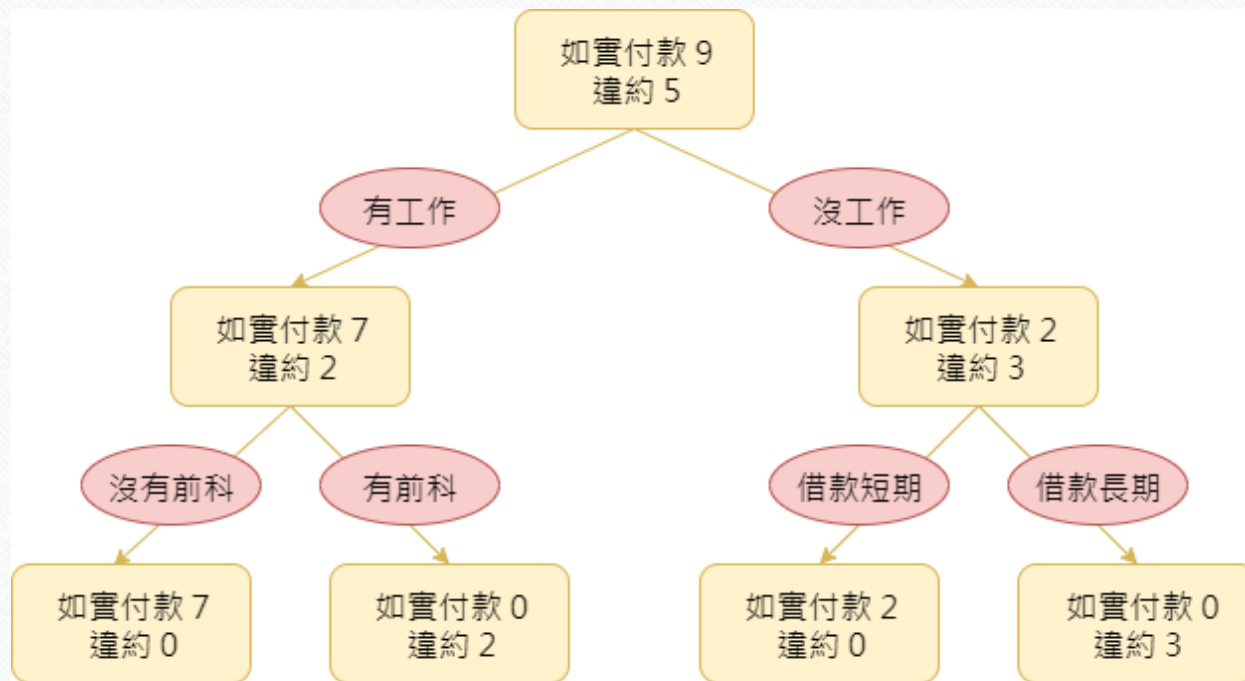
其中會違約的可能性(0.0148)大於不會違約的可能性(0.0093)，故將樣

本X預測為會違約樣本。

AI模型

決策樹

- 決策樹為一個預測模型，他代表獨立變數 x_i 與相依變數 y_i 之間的映射關係，樹中的每個節點都表示某個對象，每個分岔路徑都代表某個可能的值



決策樹

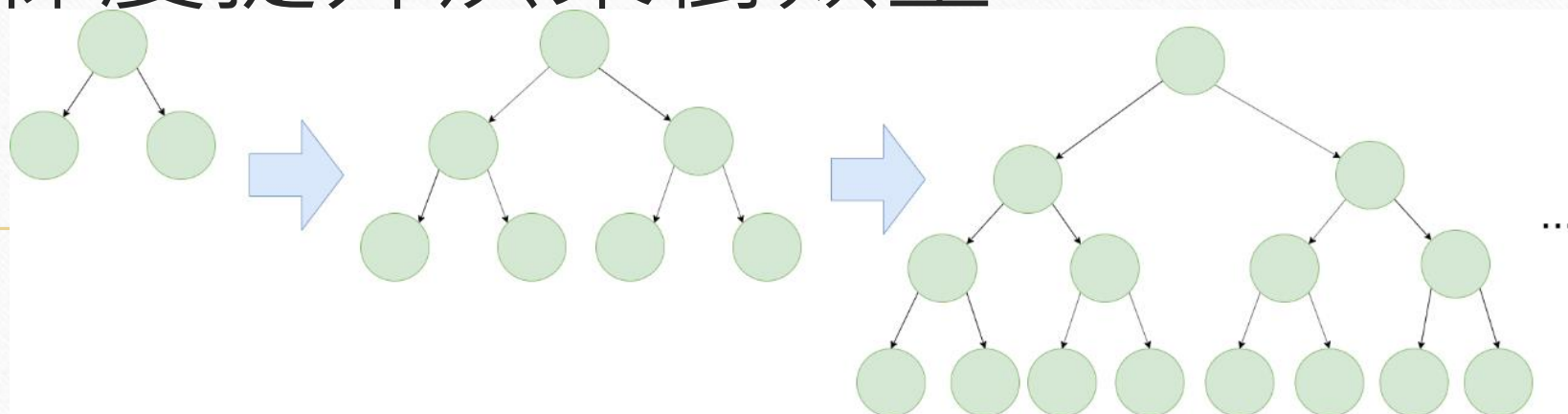
- $(X, y) = (X_1, X_2, X_3, \dots, X_k, y)$
- 算出的Entropy (熵) 越低，則該變數優先度越大，會被先用以分類資料
- $\text{Entropy} = -p * \log_2 p - q * \log_2 q$
- p ：成功的機率 (或true的機率) q ：失敗的機率 (或false的機率)
- 熵值會介於0到1之間，0最為穩定，1則最為混亂需要被分類

梯度提升決策樹

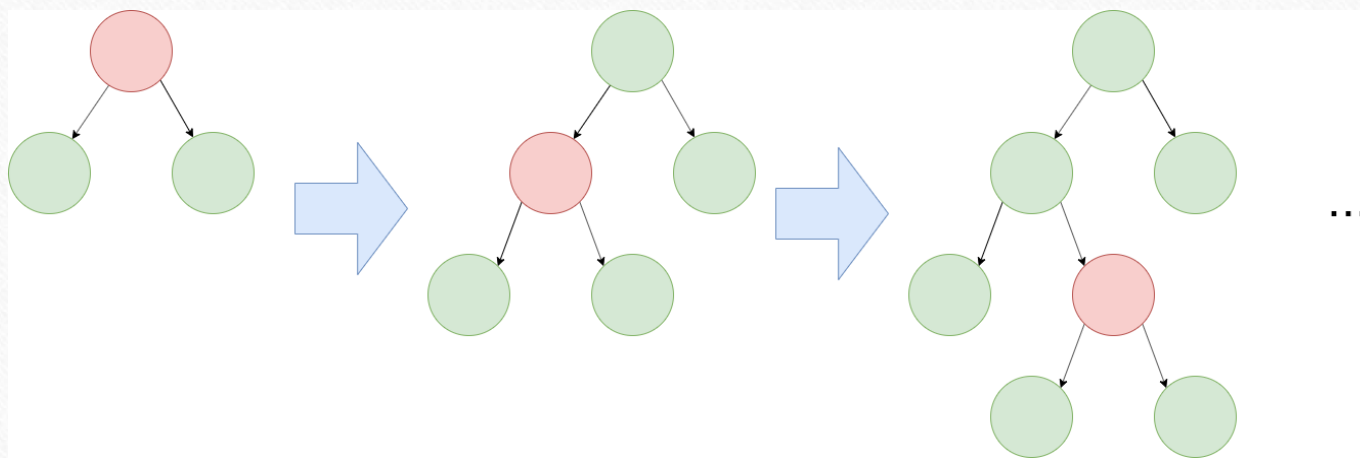
- 梯度提升決策樹 (Gradient Boosting Decision Tree , GBDT) , 為透過建立一組弱決策樹模型 , 將多個決策樹之結果累積起來 , 又稱多元加性迴歸樹(multiple additive regression tree , MART) , 將決策樹集成的概念有效的整合 , 自此演算法誕生時 , 就常與SVM一起被認為是泛化度(generalization)的演算法。

梯度提升決策樹類型

Level-wise learning



Leaf-wise learning



LightGBM

- 基於直方圖(Histogram)演算法的GBDT，即將特徵轉換為直方圖在後再尋找最優分割點
- 此演算法為實現GBDT的一個輕量框架，與傳統GBDT不同，不但會限制決策樹的生長深度，此模型會進行預剪枝，即根據節點的資料量決定是否刪除節點，防止過度擬合，為Leaf-wise learning之方法

神經網路

- $a^L = f^L(w^L p + b^L)$

- $X = [X_1, X_2 \dots X_N]^T$

- $O = [(0,1), (0,1)]^T$

- $W_L = \begin{bmatrix} W_{1,1} & \dots & W_{L-1,1} \\ \vdots & \ddots & \vdots \\ W_{1,S} & \dots & W_{L-1,S} \end{bmatrix}$

- $b_L = [b_1, b_2, \dots, b_S]^T$

- $a_L = [a_1, a_2, \dots, a_S]^T$

輸入層

隱藏層1

隱藏層2

隱藏層L

輸出層

