Day 30 特徵工程

分類型特徵優化 - 葉編碼

知識地圖 分類型特徵優化 - 葉編碼

機器學習概論 Introduction of Machine Learning

監督式學習
Supervised Learning

前處理 Processing → 探索式數據分析 Exploratory Data Analysis → **特徵工程 Feature Engineering** → 模型選擇 Model selection → 參數調整 Fine-tuning → 集成 Ensemble

非監督式學習
Unsupervised Learning

- 分群 Clustering
- 降維 Dimension Reduction

特徵工程 Feature Engineering

概論

數值型特徵
類別型特徵
時間型特徵

→ 填補缺值 → 去離群值 → 去偏態 → 特徵縮放
→ 類別型特徵處理
→ 時間型特徵處理

→ 特徵組合 / 特徵篩選 → 特徵評估

本日知識點目標

- 多個分類預測結果，該如何合併成更準確的預測
- 葉編碼的目的是什麼？如何達成該項目的？
- 葉編碼編完後，通常該搭配什麼使用？

分類預測的集成 (1 / 3)

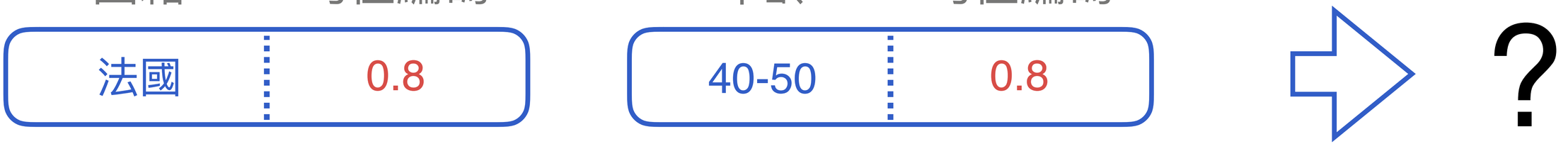
由於分類預測的集成，概念上與迴歸預測的集成有所不同，所以在最後做個補充

分類的預測結果，意義上是對機率的預估，而不同特徵表示不同的判斷條件

想一想：假如要估計鐵達尼號上的生存機率

已知來自法國的旅客生存機率是 0.8，且年齡 40 到 50 區間的生存機率也是 0.8

那麼同時符合兩種條件的旅客，生存機率應該是多少呢?



分類預測的集成 (2 / 3)

假如當作兩個預估模型，迴歸預測要集成兩種預測的做法有兩種：相加或平均

但是相加 0.8 + 0.8 = 1.6，機率會超過 1 ，不合理

平均 (0.8 + 0.8) / 2 = 0.8，法國機率已是 0.8，加上正向的事件居然還更低，也不合理

應該要比 0.8 更高，但又不能到 1

那麼，該如何集成才合理呢

分類預測的集成（3 / 3）

解法：邏輯斯迴歸(logistic regression)與其重組

我們可以將邏輯斯迴歸理解成 「線性迴歸 + Sigmoid 函數」

而 sigmoid 函數理解成 「**成功可能性**與**機率**的互換」

這裡的成功可能性正表示更可能，負表示較不可能

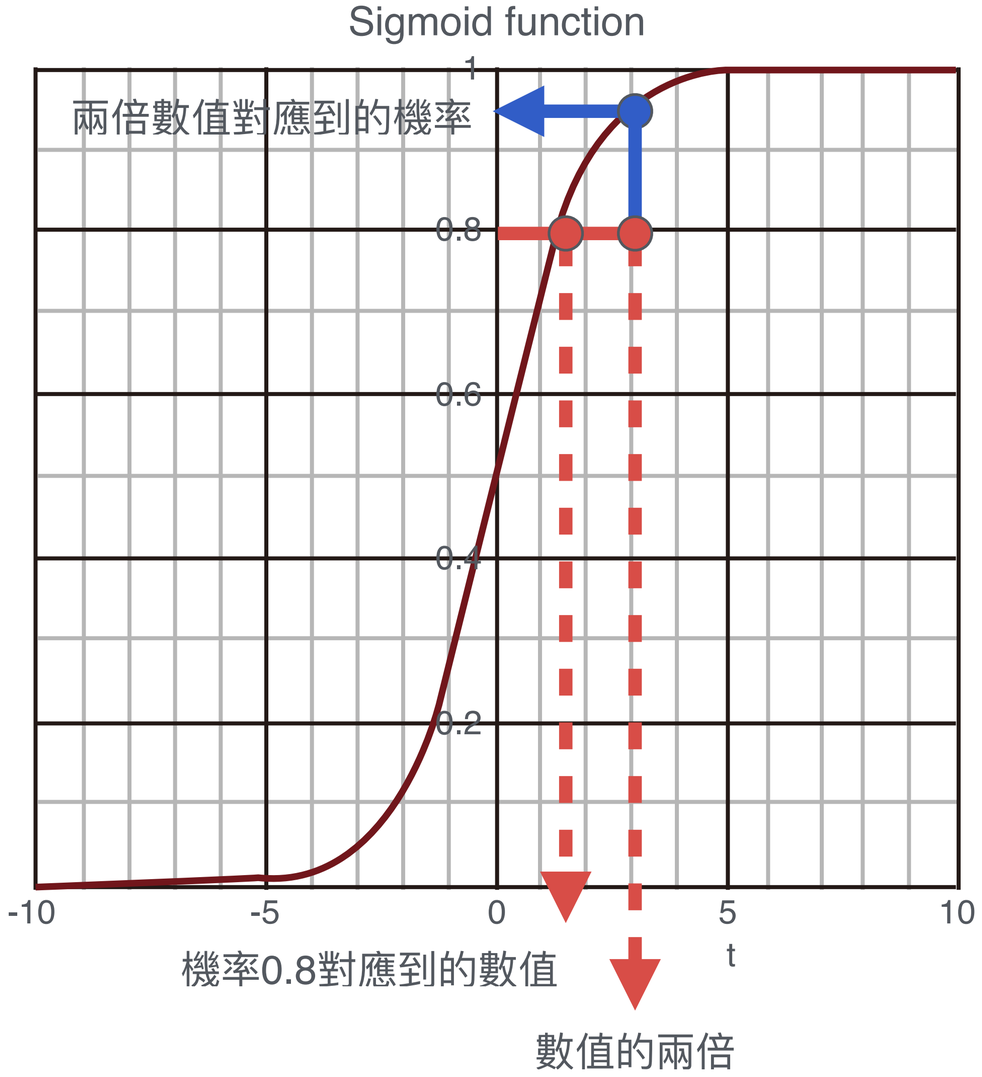
所以當我們使用 sigmoid 的反函數

就可以將機率重新轉為成功可能性

加完後再用 sigmoid 轉回機率

以此例而言，我們可以看到最後加成的結果是一個

介於 0.9 到 1 之間的機率



底圖圖片來源 ：wikipedia

葉編碼 (leaf encoding) 原理 (1 / 2)

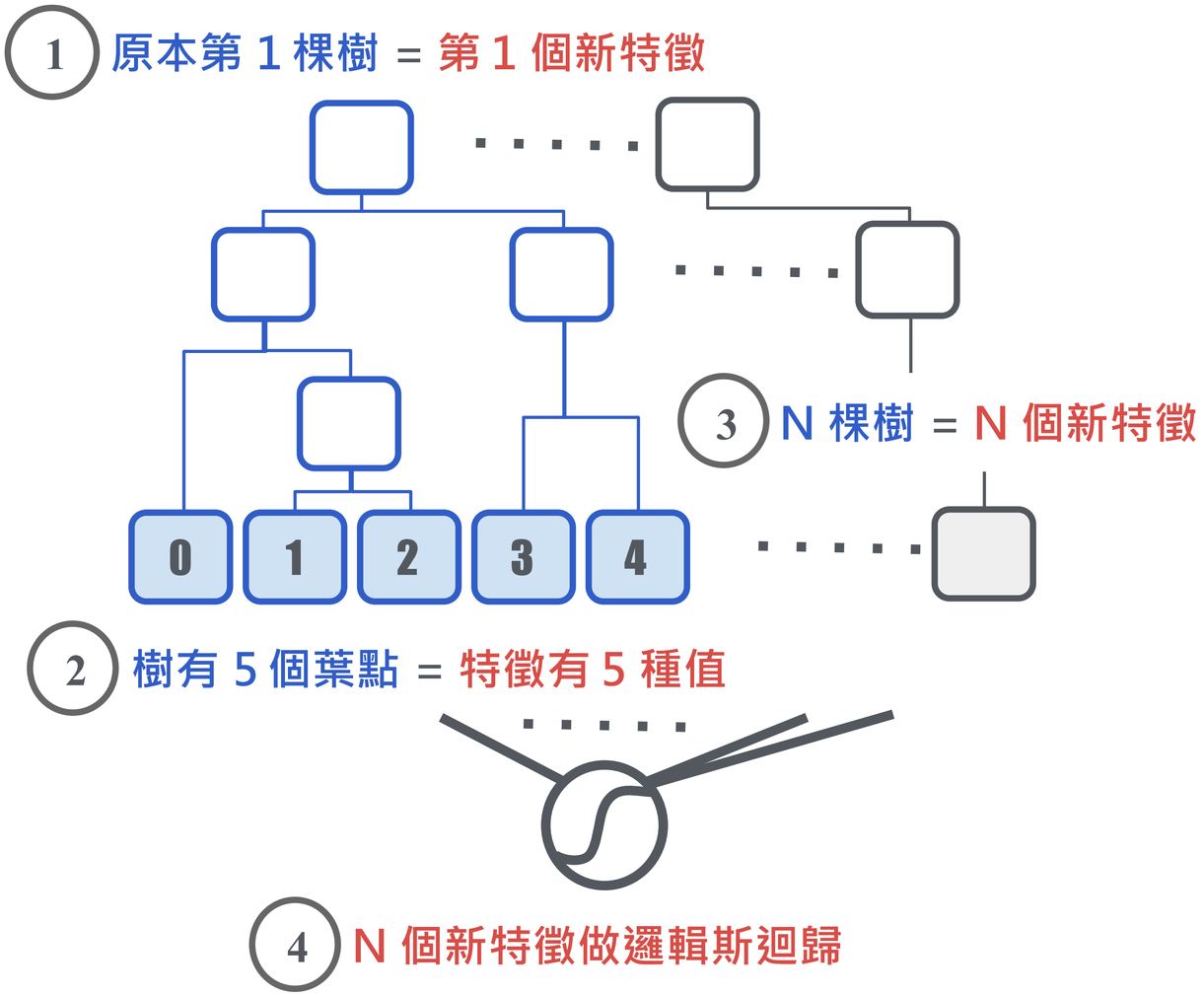
- **樹狀模型** 做出預測時，模型預測時就會將資料重新分成好幾個區塊，也就是決策樹的 **葉點** (決策樹最末端的點，詳見下頁) ，每個葉點的資料性質接近，可視為資料的一種分組方式
- 雖然不適合直接沿用樹狀模型機率，但分組方式有代表性，因此按照葉點將資料 離散化 ，比之前提過的**離散化**方式更精確，這樣的編碼我們就稱為 **葉編碼**
- 葉編碼的結果，是一組模型產生的**新特徵**，我們可以使用**邏輯斯回歸**，重新賦予機率 (如下葉圖)，也可以與其他算法結合 (例如：**分解機** Factorization Machine) 使資料獲得新生

葉編碼 (leaf encoding) 原理 (2 / 2)

- 葉編碼 (leaf encoding) 顧名思義，是採用**決策樹**的葉點作為編碼依據重新編碼

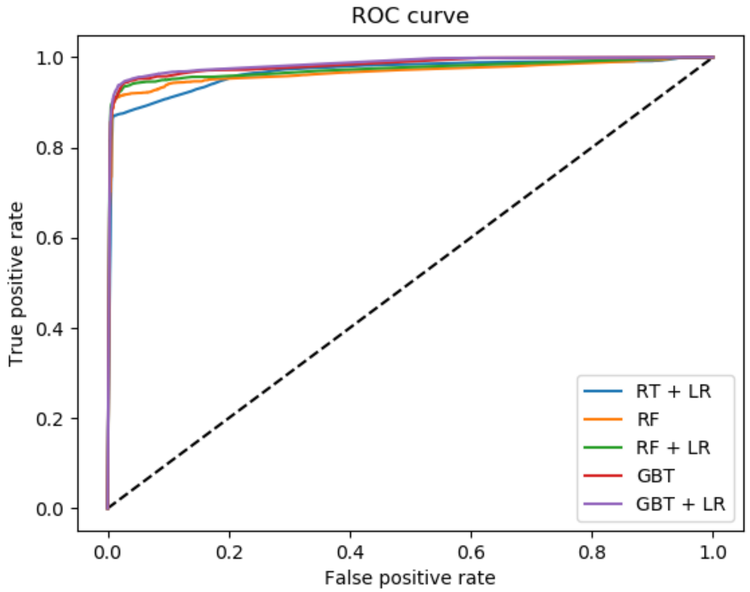
- **每棵樹**視為一個**新特徵**，每個新特徵均為**分類型特徵**，決策樹的葉點與該特徵標籤一一對應

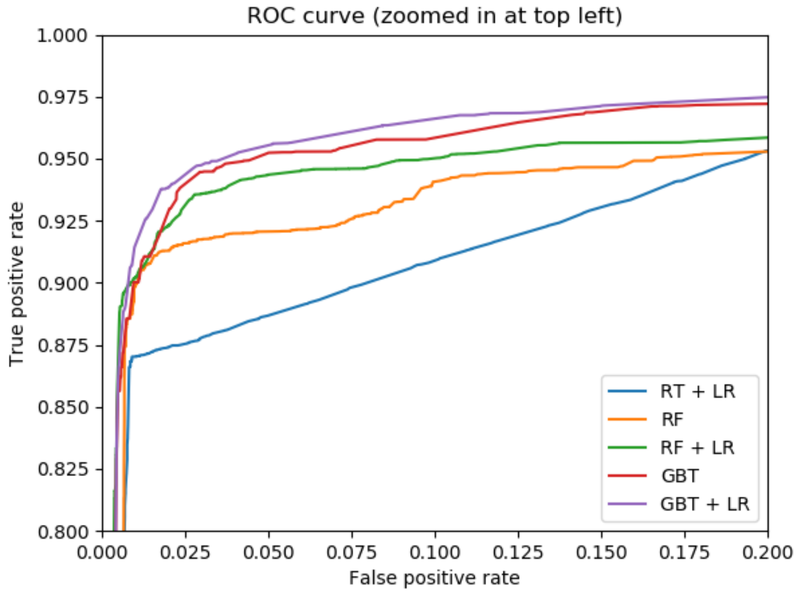
- 最後再以邏輯斯迴歸合併預測



葉編碼 (leaf encoding) + 邏輯斯迴歸

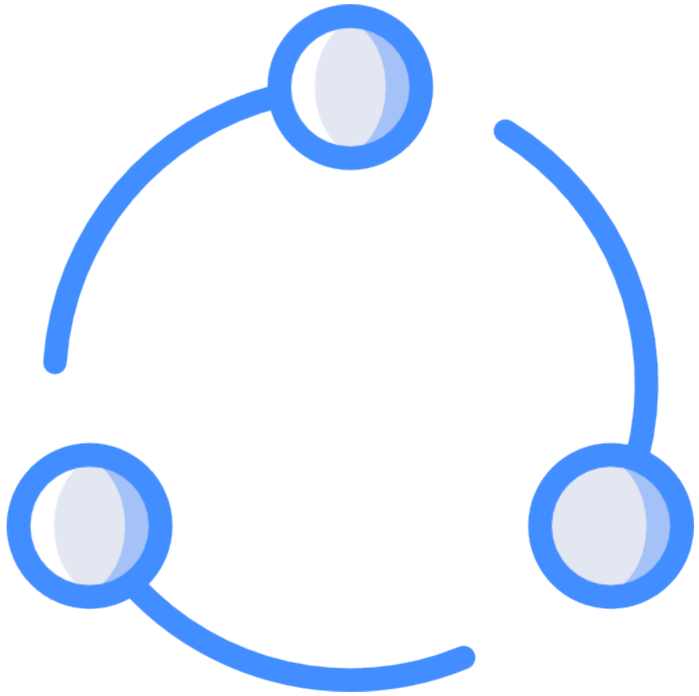
- 葉編碼需要先對樹狀模型擬合後才能生成，如果這步驟挑選了較佳的參數，後續處理效果也會較好，這點與特徵重要性類似
- 實際結果也證明，在分類預測中使用樹狀模型，再對這些擬合完的樹狀模型進行葉編碼+邏輯斯迴歸，通常會將預測效果再進一步提升





圖片來源：scikit-learn

重要知識點複習



- 多個分類預測結果，需要先將機率倒推回對應數值，相加後再由sigmoid 函數算回機率，類似**邏輯斯回歸**的算法
- 葉編碼的目的是**重新標記**資料，以擬合後的樹狀模型分歧條件，將資料**離散化**，這樣比人為寫作的判斷條件更精準，更符合資料的分布情形
- 葉編碼編完後，因為特徵數量較多，通常搭配**邏輯斯回歸**或者**分解機**做預測，其他模型較不適合

衍伸討論：有關樹狀模型與模型可解釋性

- 經由課程我們知道：樹狀模型有幾個重要的應用
  - **特徵重要性(feature importance)**：目前是特徵選擇的最主流作法
  - **葉編碼**：將特徵打散，完全依照樹狀模型的葉點重新編碼，再加上邏輯斯迴歸，可以再進一步提升分類預測能力
- 上述樹狀模型的獨特應用，都是基於人們對決策樹的理解與可解釋性(explainable)而有的設計
- 但目前深度學習的基礎：類神經網路，最缺乏的就是可以解釋性，若類神經網路能在可解釋性上更進一步，則可以想見也可以有更多的衍伸應用(例如：capsule 模型)

解題時間
It’s Your Turn

請跳出PDF至官網Sample Code＆作業
開始解題