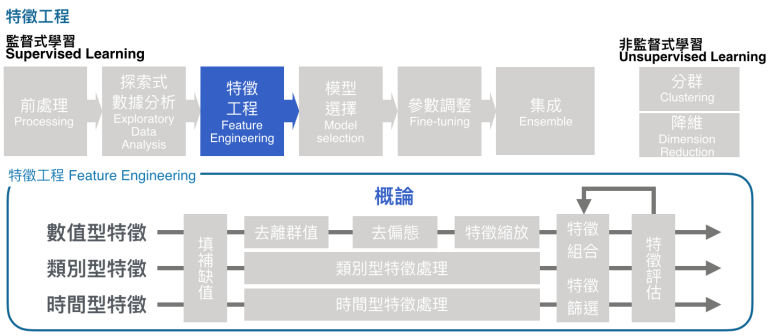
D32：分類型特徵優化 - 葉編碼

簡報閱讀 範例與作業 問題討論

分類型特徵優化 - 葉編碼

知識地圖



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

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?

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

?

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」

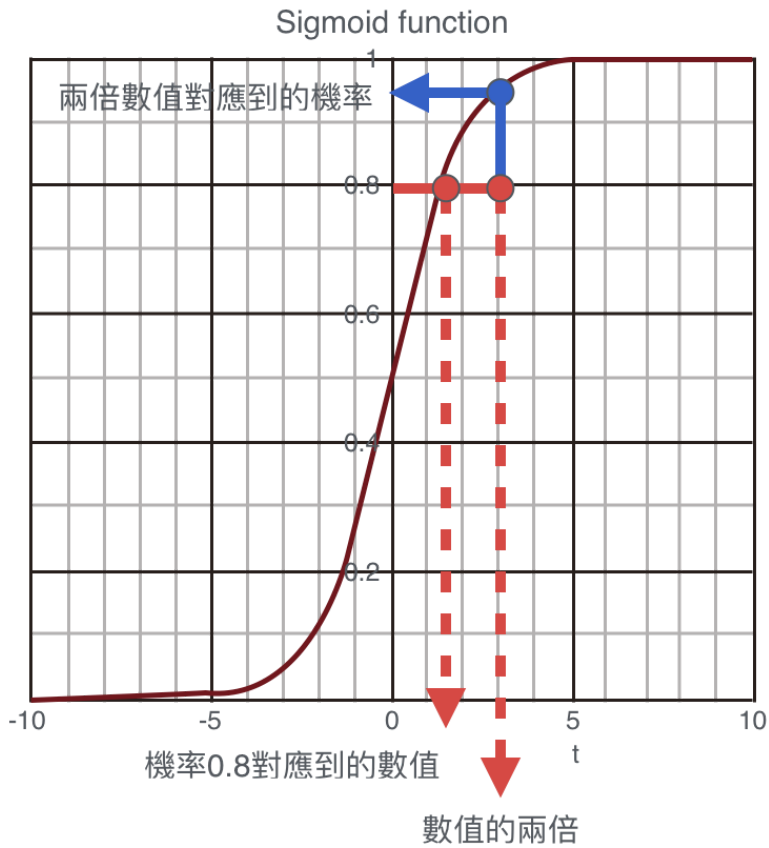
這裡的成功可能性正表示更可能，負表示較不可能

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

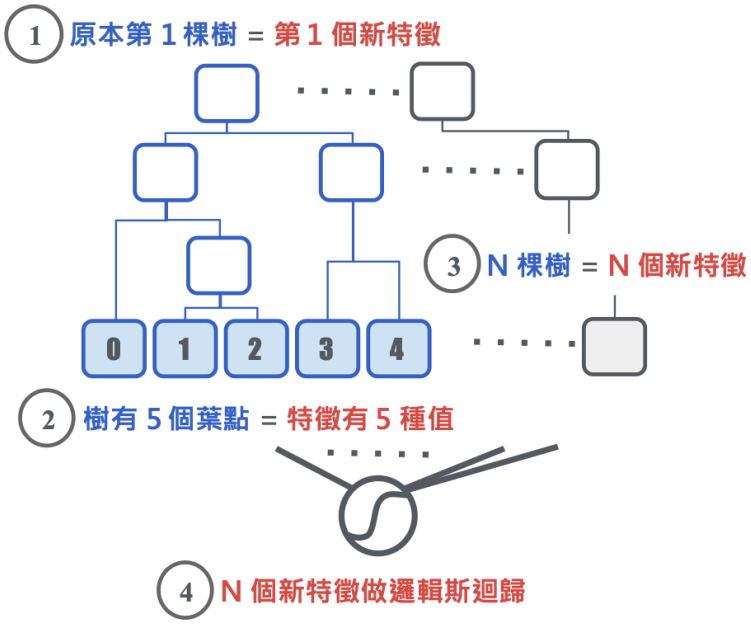
以此例而言，我們可以看到最後加成的結果是一個介於 0.9 到 1 之間的機率



葉編碼 (leaf encoding) 原理

(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

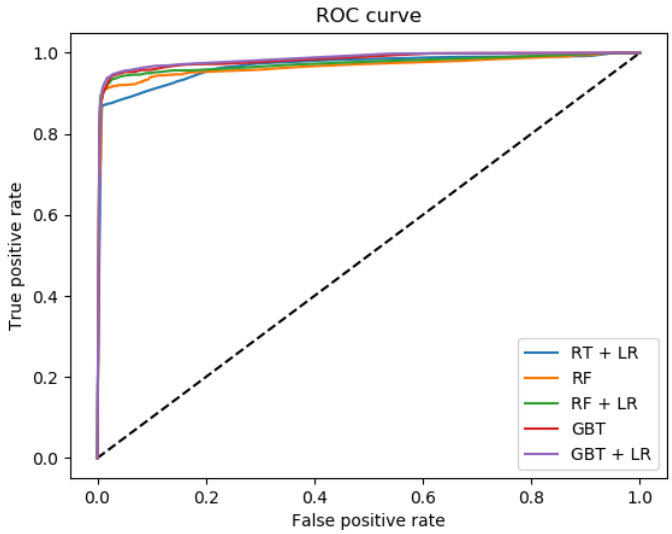
- 雖然不適合直接沿用樹狀模型機率，但分組方式有代表性，因此按照葉點將資料 離散化 ，比之前提過的**離散化**方式更精確，這樣的編碼我們就稱為 **葉編碼**
- 葉編碼的結果，是一組模型產生的**新特徵**，我們可以使用**邏輯斯回歸**，重新賦予機率 (如下頁圖)，也可以與其他算法結合 (例如：**分解機** Factorization Machine)使資料獲得新生
- 葉編碼 (leaf encoding) 顧名思義，是採用**決策樹**的葉點作為編碼依據重新編碼
- **每棵樹**視為一個**新特徵**，每個新特徵均為**分類型特徵**，決策樹的葉點與該特徵標籤一一對應
- 最後再以邏輯斯迴歸合併預測

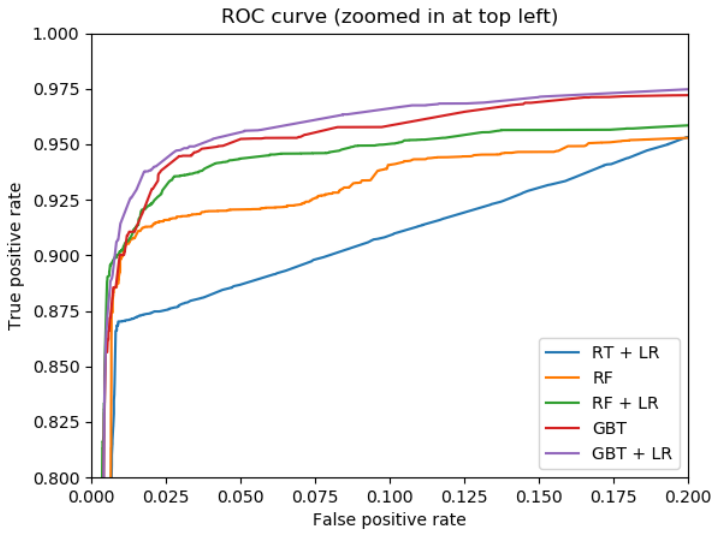


葉編碼 (leaf encoding) + 邏輯斯迴歸

- 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

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





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

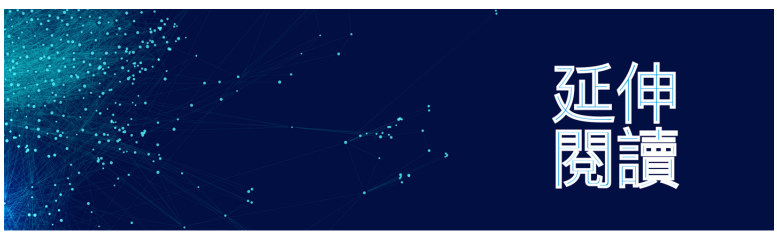
- 葉編碼的目的是**重新標記**資料，以擬合後的樹狀模型分歧條件，將資料**離散化**，這樣比人為

- 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)

延伸閱讀



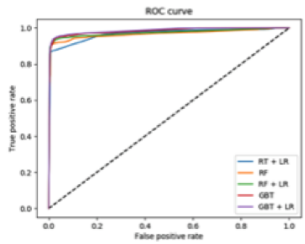
- 除了每日知識點的基礎之外，推薦的延伸閱讀能補足學員們對該知識點的了解程度

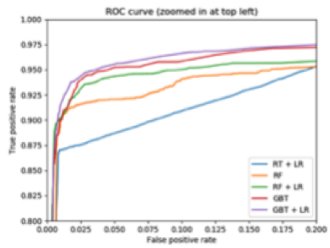
推薦延伸閱讀

Feature transformations with ensembles of trees
sklearn官網範例

> Feature transformations
> scikit-learn: machine learning in Python
> scikit-learn.org

- 這是講義內線面這張圖的出處， 裡面有完整的葉編碼程式，因為裡面的一些細節需要多做解釋，因此我們加以註解改寫後放作為今天的範例當中，同學若是有興趣也可以直接參考這篇原文，裡面有一些原文解說

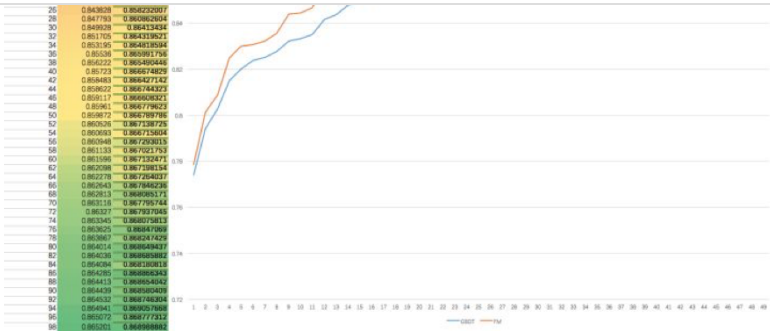




CTR 預估[十一]： Algorithm-GBDT Encoder
知乎 算法那些事兒

> CTR預估[十一]: Algorithm-
> GBDT編碼之後利用其稀疏特徵加入到傳統的LR/FM 框架中進行優化是
> zhuanlan.zhihu.com

- 這個網頁將葉編碼的應用，做了很詳盡的說明：包含使用的原因，包含的原理，以及葉編碼的戰史，如果很想弄清楚葉編碼，一定要看看這篇文章



三分鐘了解推薦系統中的分解機方法
（Factorization Machine, FM）
每日頭條

- 最後是有關分解機的解說，因為這部分稍微有點複雜，需要先了解矩陣分解 (Matrix Factorization) 與推薦系統，如果對FM沒有興趣，可以跳過此連結，但很好奇FM到底是什麼的同學，可以由此入門

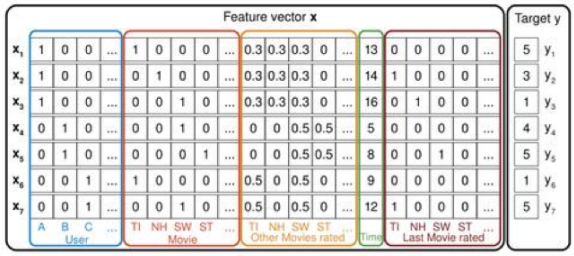


Fig. 1. Example (from Rendle [2010]) for representing a recommender problem with real valued feature vectors $\mathbf{x}$. Every row represents a feature vector $\mathbf{x}_i$ with its corresponding target y_i. For easier interpretation, the features are grouped into indicators for the active user (blue), active item (red), other movies rated by the same user (orange), the time in months (green), and the last movie rated (brown).

解題時間

Sample Code ＆作業
開始解題

下一步：閱讀範例與完成作業