

Day 26

特徵工程

類別型特徵 - 其他進階處理



出題教練

陳明佑

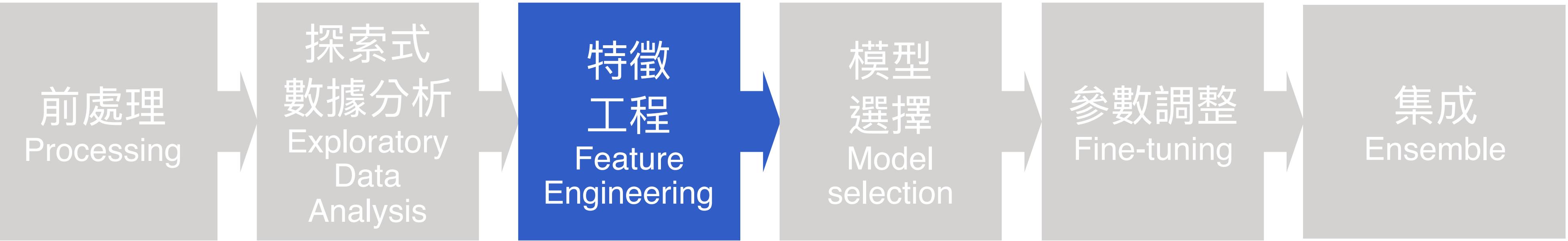


知識地圖 特徵工程 類別型特徵 - 其他進階處理

特徵工程

監督式學習

Supervised Learning



非監督式學習

Unsupervised Learning



特徵工程 Feature Engineering

數值型特徵

類別型特徵

時間型特徵

填補缺值

去離群值

類別型特徵處理

時間型特徵處理

概論

去偏態

特徵縮放

特徵組合

特徵篩選

特徵評估

本日知識點目標

- 什麼是計數編碼，在什麼條件下可以考慮使用
- 雜湊編碼在什麼情況下可以考慮使用

計數編碼 (Counting)

如果類別的目標均價與類別筆數呈正相關（或負相關），也可以將筆數本身當成特徵
例如：購物網站的消費金額預測

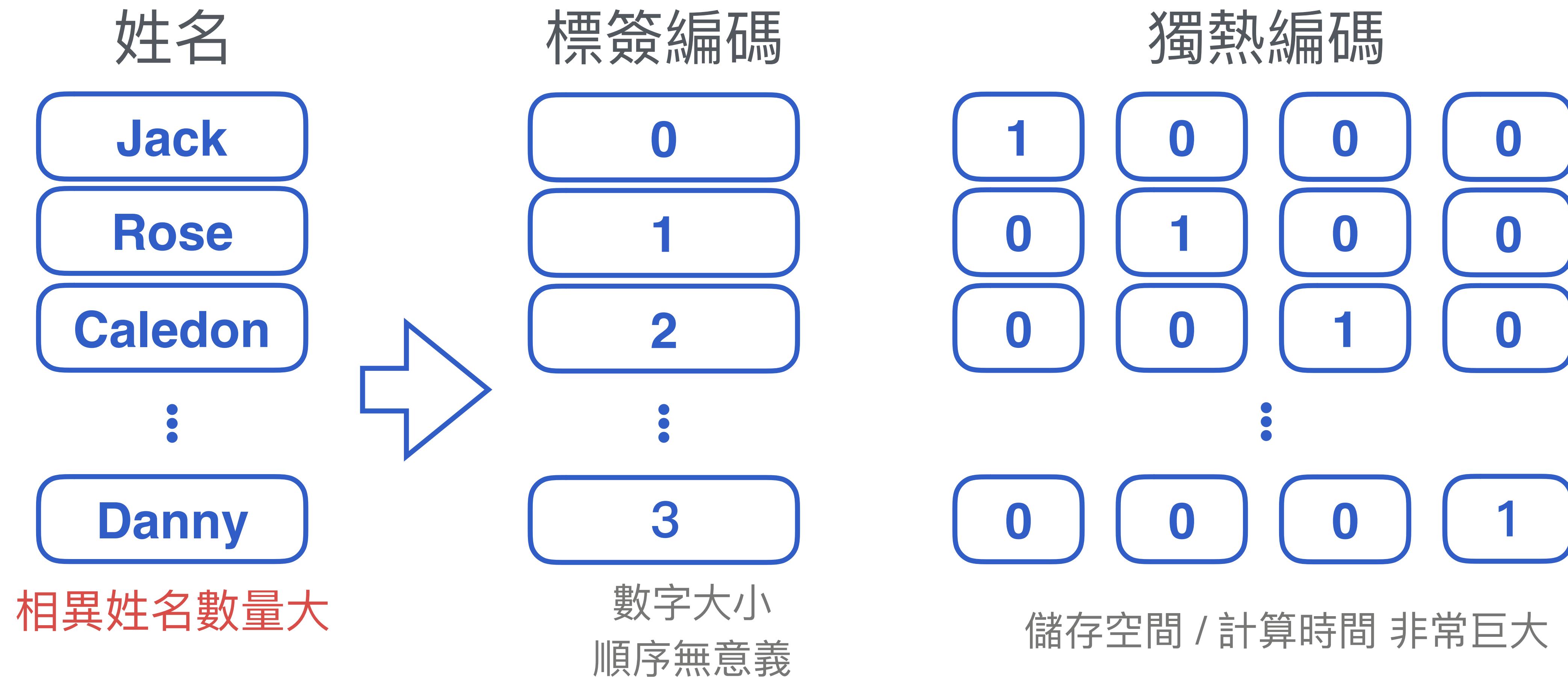


*自然語言處理時，字詞的計數編碼又稱詞頻，本身就是一個很重要的特徵

特徵雜湊 (Feature Hash) (1 / 2)

類別型特徵最麻煩的問題：相異類別的數量非常龐大，該如何編碼？

*舉例：鐵達尼生存預測的旅客姓名

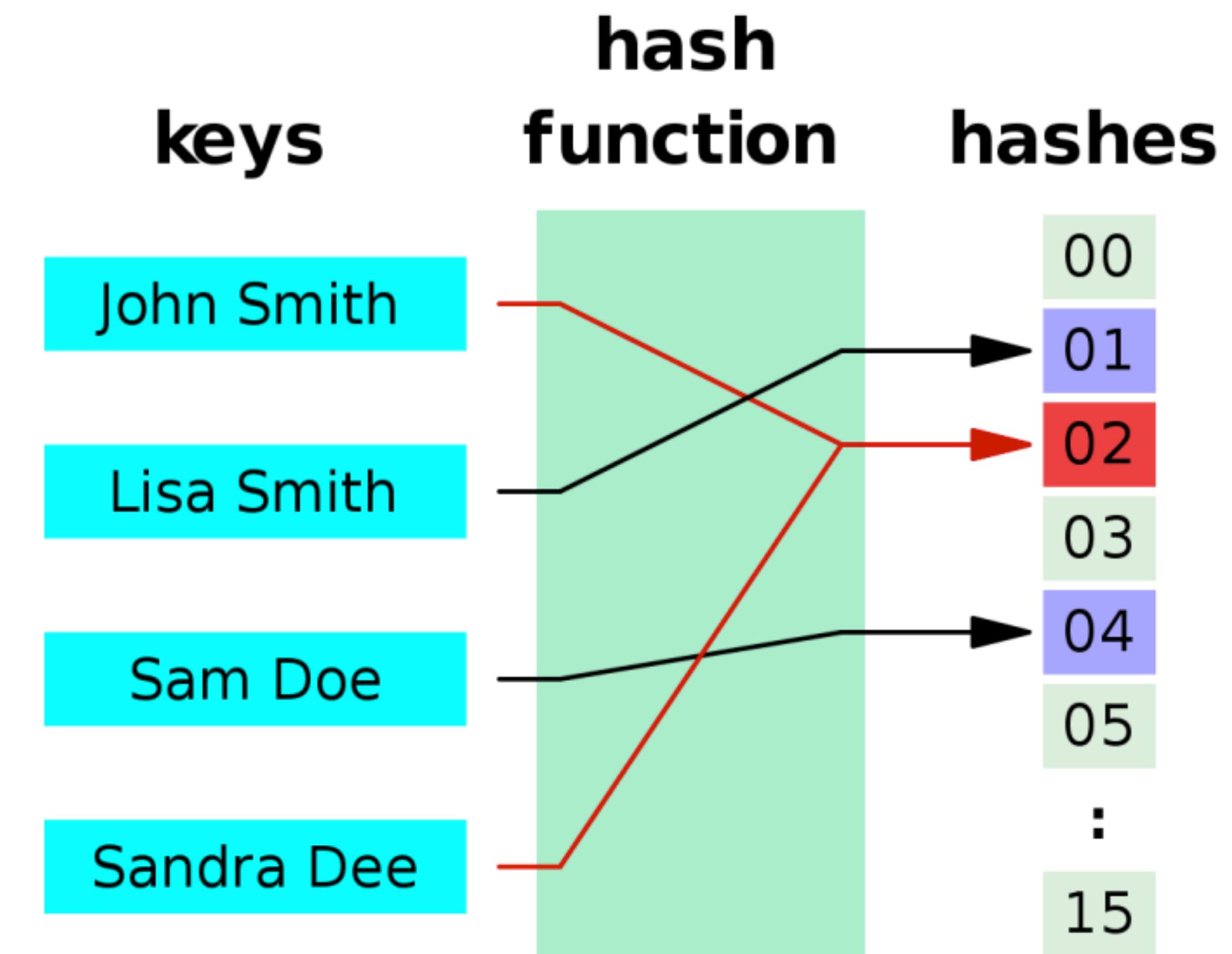


特徵雜湊 (Feature Hash) (2 / 2)

這個問題沒有很好的通用解法...只能採折衷方案或個別情況解決

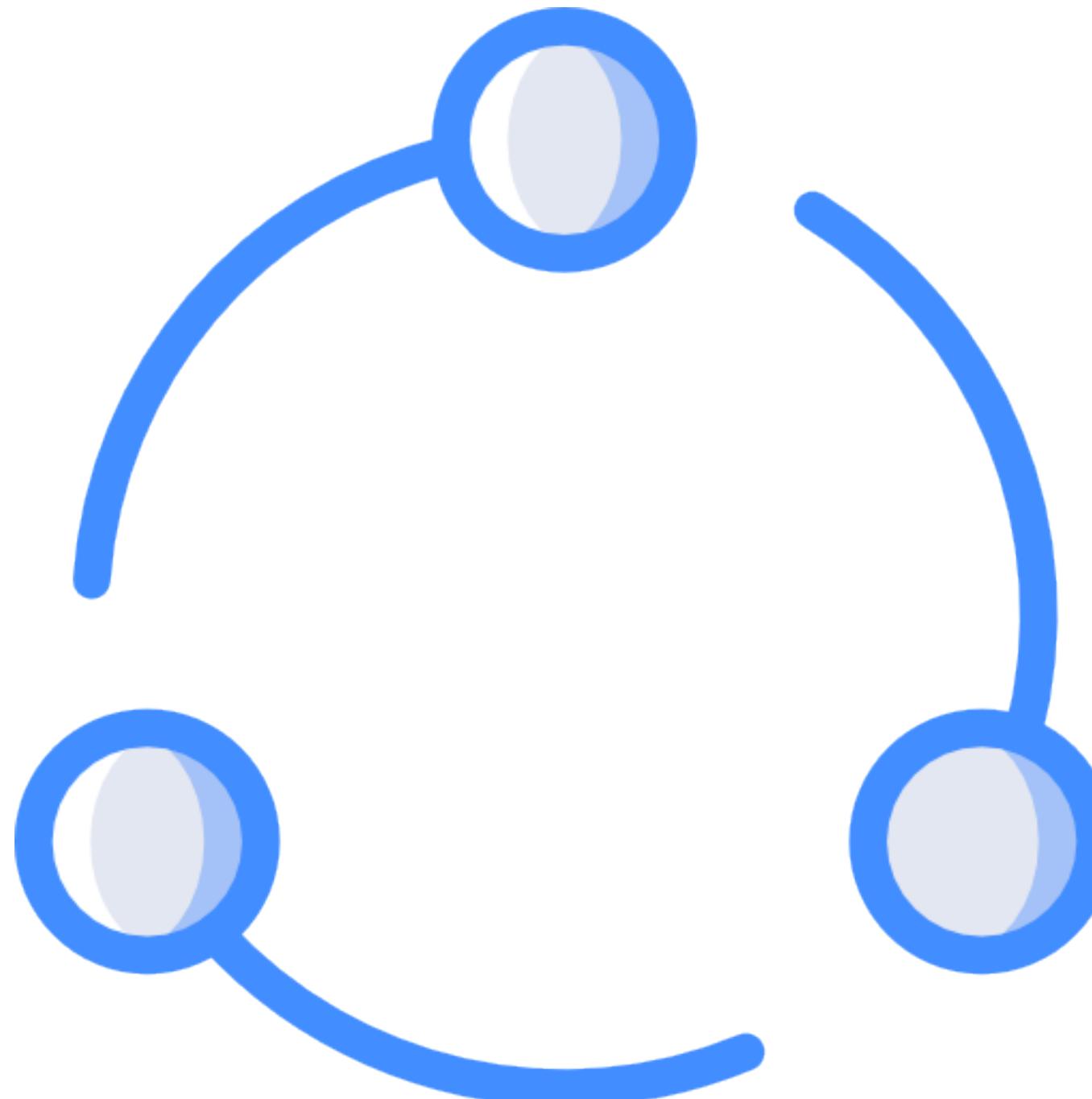
特徵雜湊

- 特徵雜湊是一種折衷方案
- 將類別由雜湊函數定應到一組數字
- 調整雜湊函數對應值的數量
- 在計算空間/時間與鑑別度間取折衷
- 也提高了訊息密度，減少無用的標籤

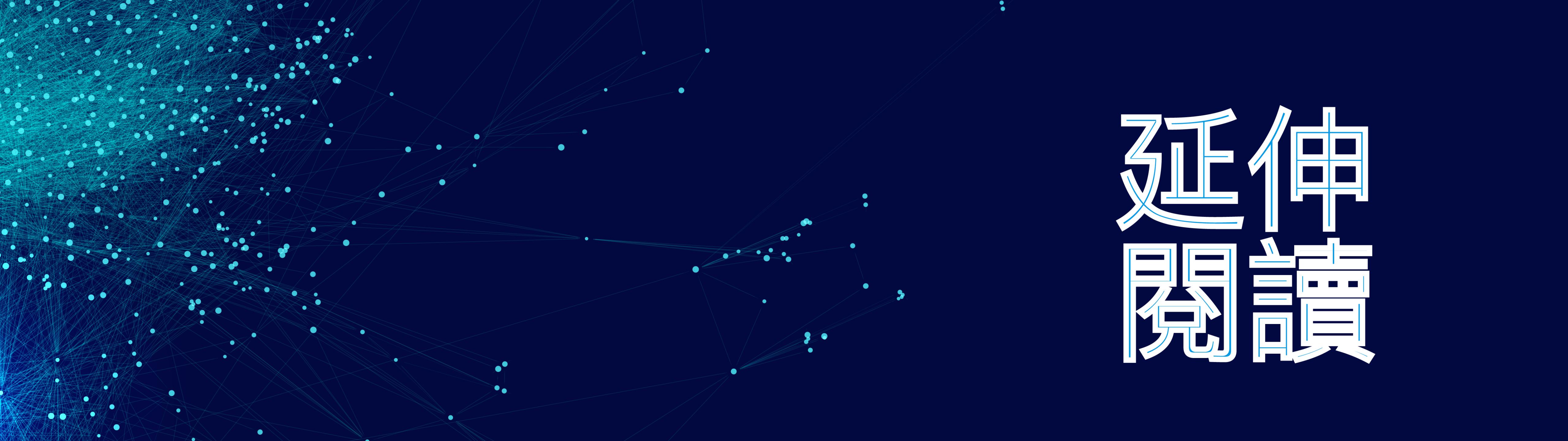


圖片來源：維基百科 https://en.wikipedia.org/wiki/Hash_function

重要知識點複習



- 計數編碼是計算類別在資料中的**出現次數**，當**目標平均值**與**類別筆數**呈正/負相關時，可以考慮使用
 - 當**相異類別數量**相當**大**時，其他編碼方式效果更差，可以考慮**雜湊編碼**以節省時間
- *註：雜湊編碼效果也不佳，這類問題更好的解法是**嵌入式編碼**(Embedding)，但是需要深度學習並有其前提，因此這裡暫時不排入課程



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，建議您解完每日題目後，若有
多餘時間，可再補充延伸閱讀文章內容。

推薦延伸閱讀

Feature hashing (特徵哈希)

CSDN 大師魯 網頁連結

- 由右圖可以理解：雜湊編碼是比標籤編碼(上表)更緊密的編碼方式(下表)，但要注意的是這樣的編碼：雖然在計算上比獨熱編碼省去很多時間，但是關鍵在雜湊後的特徵是否有意義。這邊有除了範例以外的細節講述，提供各位同學參考。

Term	Index
John	1
likes	2
to	3
watch	4
movies	5
Mary	6
too	7
also	8
football	9

$$\begin{pmatrix} \text{John} & \text{likes} & \text{to} & \text{watch} & \text{movies} & \text{Mary} & \text{too} & \text{also} & \text{football} \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

推薦延伸閱讀

基於sklearn的文本特徵抽取

簡書 網頁連結

- 這裡講到的是 count vectorizer 與 tfidf vectorizer，是自然語言處理 (NLP) 時用的基礎技術之一，其中 count vectorizer 就是一種計數編碼的變形。
- 雖然上述兩種編碼方式現階段暫時不用弄懂，但是我們可以藉此理解：計數編碼有其泛用性，甚至我們可以這樣理解 - 不需要局限於我們教會各位的編碼方式，只要在您的知識中有更適合的擷取特徵方式，並且能使用程式寫作出來的，建議不妨一試，就算不是泛用的編碼法，只要包含領域知識就可能有用。

特徵提取

```
from sklearn.feature_extraction.text import CountVectorizer  
from sklearn.feature_extraction.text import TfidfVectorizer
```

count vectorizer

```
c_vec = CountVectorizer()  
x_count_train = c_vec.fit_transform(x_train)  
x_count_test = c_vec.transform(x_test)
```



解題時間

It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

