D24：類別型特徵 - 其他進階處理

PDF 下載 全螢幕

Sample Code & 作業內容

本範例中，將數值型特徵做類別型編碼

作業1：參考範例Day_024_CountEncoder_and_FeatureHash.ipynb，將鐵達尼的艙位代碼('Cabin')欄位使用特徵雜湊 / 標籤編碼 / 目標均值編碼三種轉換後，與其他數值型欄位一起預估生存機率。

作業2：承上題，三者比較效果何者最好?

作業請提交Day_024_HW.ipynb

檢視範例

參考資料

Feature hashing (特徵哈希)

CSDN 大師會 **網頁連結**
由下圖可以理解：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

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

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)
```

提交作業

請將你的作業上傳至 Github，並貼上該網網址，完成作業提交

https://github.com/ 確定提交

如何提交

到 Cupoy 問答社區提問，讓教練群回答你的疑難雜症

向專家提問

如何提問