

D24：類別型特徵 - 基礎處理



簡報閱讀



範例與作業



問題討論

類別型特徵 - 基礎處理 >

知識地圖 >

本日知識點目標 >

類別型特徵的處理 >

基礎編碼 1：標籤編碼 (Label Encoding) >

基礎編碼 2：獨熱編碼 (One Hot Encoding) >

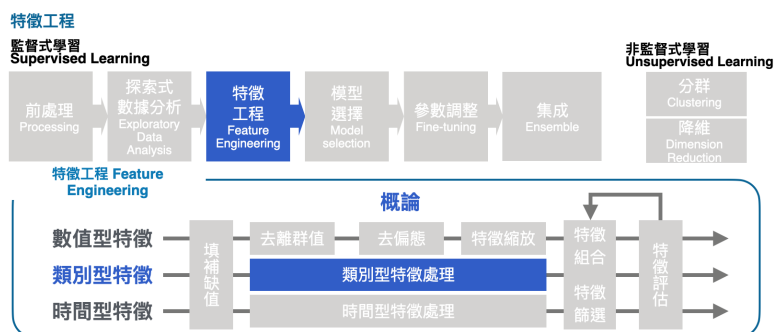
標籤編碼 / 獨熱編碼的比較 >

重要知識點複習 >

類別型特徵 - 基礎處理



知識地圖



本日知識點目標

今日知識點目標

- 類別型特徵有哪兩種基礎編碼方式?
- 兩種基礎編碼方式中，哪一種比較常用? 為什麼?
- 在什麼情況下，比較適合獨熱編碼?

類別型特徵的處理

前面提過：特徵工程是事實到對應分數的轉換

請先回憶一下，已學過哪些類別型特徵的轉換方式，您是否可以想到其他的轉換方法?



基礎編碼 1：標籤編碼 (Label Encoding)

- 類似於流水號，依序將新出現的類別依序編上新代碼，已出現的類別編上已使用的代碼
- 確實能轉成分數，但缺點是分數的**大小順序只與轉換對照有關，與標籤的意義無直接相關**。



基礎編碼 2：獨熱編碼 (One Hot Encoding)

- 為了改良數字大小沒有意義的問題，將不同的類別分別獨立為一欄
- 缺點是需要較大的記憶空間與計算時間，且類別數量越多時越嚴重



標籤編碼 / 獨熱編碼的比較

	儲存空間/計算時間	適用學習模型
標籤編碼 Label Encoding	小	非深度學習模型
獨熱編碼 One Hot Encoding	較大	深度學習模型

綜合建議

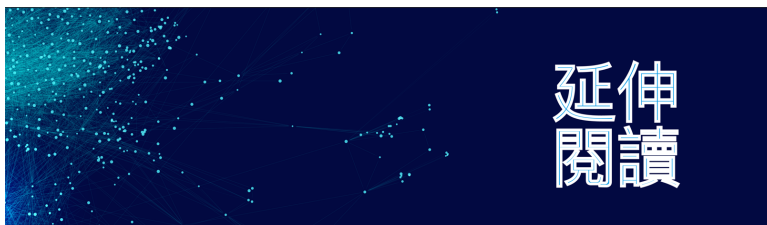
- 非深度學習時，類別型特徵建議預設採標籤編碼；深度學習時，預設採獨熱編碼

檻就能分隔關鍵類別；但深度學習主要依賴倒傳遞，標籤編碼會不易收斂

重要知識點複習

- 類別型特徵有**標籤編碼** (Label Encoding) 與**獨熱編碼** (One Hot Encoding) 兩種基礎編碼方式
- 兩種編碼中標籤編碼比較常用
- 當**特徵重要性高**，且**可能值較少**時，才應該考慮獨熱編碼

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

數據預處理：獨熱編碼 (One-Hot Encoding) 和 LabelEncoder標籤編碼

一、問題由來 在很多機器學習任務中，特徵並不總是連續值，而有可能

www.twblogs.net

其實One Hot Encoding與Label Encoder是類別型資料最常見的編碼方式，因此實現的程式碼也頗為常用，其中 One Hot Encoding 常見的兩種做法：

pandas.get_dummies 與 sklearn 的 OneHotEncoder 在這網頁中都有清楚的展示，本課程今日範例中會用到前者，在之後的葉編碼中則會用到後者，所以同學不妨先了解一下寫法。

```
In [10]: s1 = ['a', 'b', np.nan]
pd.get_dummies(s1, dummy_na=True)
```

```
Out[10]:
```

	a	b	nan
0	1	0	0
1	0	1	0
2	0	0	1

```
In [12]: df2 = pd.DataFrame({'A': ['a', 'b', 'a'], 'B': ['b', 'a', 'c'], 'C': [1, 2, 3]})
pd.get_dummies(df2, prefix=['col1', 'col2'])
```

```
Out[12]:
```

	C	col1_a	col1_b	col2_a	col2_b	col2_c
0	1	1	0	0	1	0
1	2	0	1	1	0	0
2	3	1	0	0	0	1

解題時間



Sample Code & 作業
開始解題



