

D11 pandas 類別資料與缺失值處理

[簡報閱讀](#)[範例與作業](#)[問題討論](#)[學習心得\(完成\)](#)

重要知識點



- 認識缺值處理方法與應用函式

認識類別資料

變數的特徵屬於非數值型態。需利用一組的標記、類別、性質或名稱以區別每個基本單位的特徵、屬性。無法以數值表示的統計資料，如航班編號、性別、學歷、旅遊同伴、頭髮顏色、宗教等。類別資料中可以分為兩類順序性與一般性兩種。

- 順序性：類別之間存在順序性，例如：衣服尺寸[XL,L,M]、長度[短,中,長]
- 一般性：類別之間沒有順序關係，例如：顏色[黃,綠,藍]、性別[男,女]

大部分的模型都是基於數學運算，字串無法套入數學模型進行運算，在此先對其進行 encoding 編碼(將類別資料轉成數字)才能進一步對其做分析。

- 對於順序性的類別資料，需要有順序性的 encoding 方法，可以使用 sklearn 中的 `LabelEncoder()`。
- 對於一般性的類別資料，則不需要有順序的編碼，可以使用 pandas 中的 `get_dummies()`

認識類別資料：順序性 `LabelEncoder()`



類別之間會有順序關係 $0 < 1 < 2 < \dots$ ，排序依照 python 內建順序，可以藉由 `ord()` 查看內建順序。

```
[1] import pandas as pd
```

```
[2] df = pd.DataFrame([[ 'green', 'M', 'male', 'short'],
                      [ 'red', 'L', 'female', 'normal'],
                      [ 'blue', 'XL', 'male', 'long']])
df.columns = ['color', 'size', 'sex', 'length']
df
```

	color	size	sex	length
0	green	M	male	short
1	red	L	female	normal
2	blue	XL	male	long

```
[3] from sklearn.preprocessing import LabelEncoder
df['size_label'] = LabelEncoder().fit_transform(df['size'].values)
df
```

	color	size	sex	length	size_label
0	green	M	male	short	1
1	red	L	female	normal	0
2	blue	XL	male	long	2

```
[3] from sklearn.preprocessing import LabelEncoder
df['size_label'] = LabelEncoder().fit_transform(df['size'].values)
df
```

	color	size	sex	length	size_label
0	green	M	male	short	1
1	red	L	female	normal	0
2	blue	XL	male	long	2

認識類別資料：一般性
`get_dummies()`



資料表的欄位代表所屬的類別，如下欄位 color 中有 green、red、blue 將他們一一編入欄位中 color_blue、color_green、color_red 這個一對一的關係通常稱為 One-hot Encoding(一位有效編碼)是沒有順序性的編碼。

```
[7] pf = pd.get_dummies(df[['color']])
df = pd.concat([df, pf], axis=1)
df
```

	color	size	sex	length	size_label	color_blue	color_green	color_red
0	green	M	male	short	1	0	1	0
1	red	L	female	normal	0	0	0	1
2	blue	XL	male	long	2	1	0	0

認識缺值處理方法與應用函式

資料缺失時常發生在問卷資料上，填寫人時常會漏寫或不願意填寫，導致資料上有缺失值，只要缺失值將會填上 nan 代替缺失值，大部分的模型不能處理缺失值的問題，一般來說會將有缺失值的資料整筆直接刪除，但是這樣會損失其它欄位的資料，所以如果缺失情況不嚴重，傾向於將缺失值補上數值，以下最常見兩種補值方式。

1. 定值補值：將缺失值都補上一個定值
2. 前(後)補值：補前(後)一列的值

認識缺值處理方法與應用函式：定值

函式 fillna() 可以將所有缺失值填補上固定的數值

temp_data

	date	current_temp
0	2020-11-01	24.8
1	2020-11-02	24.8
2	2020-11-03	NaN
3	2020-11-04	25.0

```
[21] #以0填補
temp_data.fillna(0)
```

	date	current_temp
0	2020-11-01	24.8
1	2020-11-02	24.8
2	2020-11-03	0.0
3	2020-11-04	25.0

也可以補上平均值、中位數、...等的數值

```
[20] #以該欄位所有資料的算術平均數做填補
temp_data.fillna(temp_data.current_temp.mean())
```

	date	current_temp
0	2020-11-01	24.800000
1	2020-11-02	24.800000
2	2020-11-03	24.866667
3	2020-11-04	25.000000

```
[24] #以該欄位所有資料的中位數做填補
temp_data.fillna(temp_data.current_temp.median())
```

	date	current_temp
0	2020-11-01	24.8
1	2020-11-02	24.8
2	2020-11-03	24.8
3	2020-11-04	25.0



前(後)補值最常使用在金融上，有時候因為颱風天導致沒有開盤，這時沒開盤那天的數值空了通常都會補前一天的價錢。

函式一樣使用 `fillna()`，我們只需要進一步運用參數 `method='ffill'` 即可填補前一列數值，`method='bfill'` 填補後一列數值。

```
[26] temp_data.fillna(method='ffill')
```

	date	current_temp
0	2020-11-01	24.8
1	2020-11-02	24.8
2	2020-11-03	24.8
3	2020-11-04	25.0

```
[27] temp_data.fillna(method='bfill')
```

	date	current_temp
0	2020-11-01	24.8
1	2020-11-02	24.8
2	2020-11-03	25.0
3	2020-11-04	25.0

知識點回顧

a. 順序性 LabelEncoder()

b. 一般性 get_dummies()

- 缺值處理方法共有三種

a. 定值補值

b. 前(後)補值

參考資料

使用 get_dummies 進行 one-hot 編碼

網站：itread01.com

離散特徵的編碼分為兩種情況：

1、離散特徵的取值之間沒有大小的意義，比如color：[red,blue],那麼就使用one-hot編碼

2、離散特徵的取值有大小的意義，比如size:[X,XL,XXL],那麼就使用數值的對映{X:1,XL:2,XXL:3}

使用pandas可以很方便的對離散型特徵進行one-hot編碼

```
1 import pandas as pd
2 df = pd.DataFrame([
3     ['green', 'M', 10.1, 'class1'],
4     ['red', 'L', 13.5, 'class2'],
5     ['blue', 'XL', 15.3, 'class1']])
6
7 df.columns = ['color', 'size', 'prize', 'class label']
8
9 size_mapping = {
10     'XL': 3,
11     'L': 2,
12     'M': 1}
13 df['size'] = df['size'].map(size_mapping)
14
15 class_mapping = {label:idx for idx,label in enumerate(set(df['class label']
16 df['class label'] = df['class label'].map(class_mapping)
```

說明：對於有大小意義的離散特徵，直接使用對映就可以了，{'XL':3,'L':2,'M':1}

	color	size	prize	class label
0	green	1	10.1	0
1	red	2	13.5	1
2	blue	3	15.3	0

1 | Using the get_dummies will create a new column for every unique string in a

Label encoding

網站：[初學Python手記 # 3-資料前處理 \(標籤編碼，一種熱編碼\)](#)

重要知識點

認識類別資料

認識類別資料：順序性
LabelEncoder()



Apple	1	95	→	1	0	0	95
Chicken	2	231		0	1	0	231
Broccoli	3	50		0	0	1	50

...

認識類別資料：一般性
get_dummies()



1.標籤編碼

```
import numpy as np
import pandas pd
country = ['Taiwan', 'Australia', 'Ireland', 'Australia', 'Ireland', 'Taiwan']
age = [25, 30, 45, 35, 22, 36]
薪金= [20000, 32000, 59000, 60000, 43000, 52000]
dic = {'Country': country, 'Age': age, 'Salary': salary}
data = pd.DataFrame(dic)
data
```

認識缺值處理方法與應用
函式



認識缺值處理方法與應用
函式：定值補值



	Country	Age	Salary
0	Taiwan	25	20000
1	Australia	30	32000
2	Ireland	45	59000
3	Australia	35	60000
4	Ireland	22	43000
5	Taiwan	36	52000

認識缺值處理方法與應用
函式：前(後)補值



[下一步：閱讀範例與完成作業](#)

