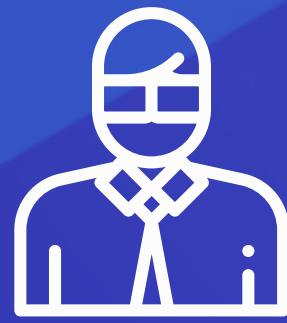




Day 7

機器學習前處理

特徵類型



陳明佑

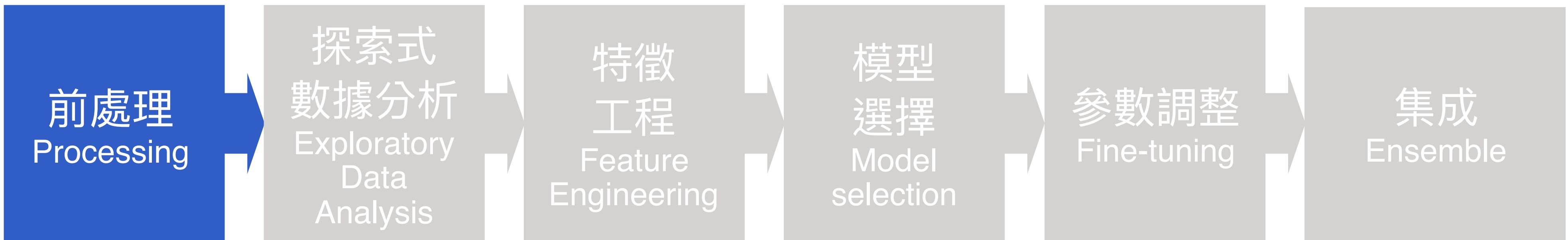
出題教練



知識地圖 機器學習前處理 特徵類型

機器學習前處理

監督式學習 Supervised Learning



非監督式學習 Unsupervised Learning



前處理 Processing



本日知識點目標

- 思考資料中常見的有哪些特徵類？
- 上面這些類型：要轉換成對目標的猜測，有哪些要特別注意的地方？

常見特徵類型：數值型特徵 / 類別型特徵

常見特徵有兩大類 (舉Day017的範例)

數值型

坪數

20坪

每坪50點

類別型

行政區

新北市

台北市+500點
其餘+0點

類別型

性別

男

男-2點
女+1點

數值型

年齡

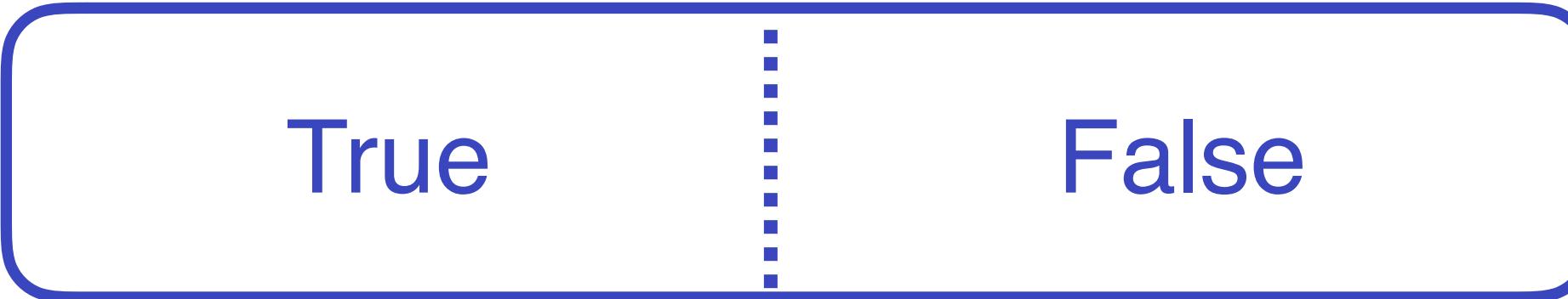
50

60以上0點
30~59 +1點
29以下+2點

- 數值型特徵：有不同轉換方式，函數 / 條件式都可以
- 類別型特徵：通常一種類別對應一種分數

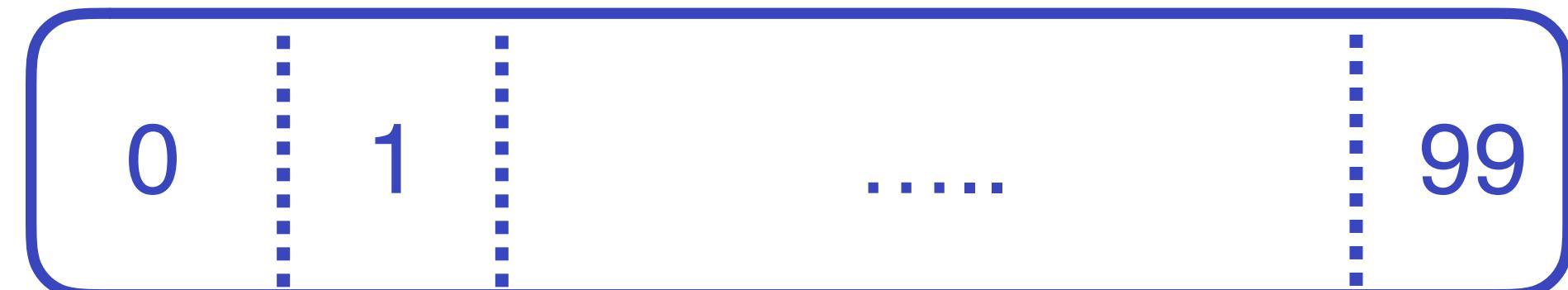
其他特徵類型 (1 / 2)

二元特徵



- 只有 True / False 兩種數值的特徵
- 可以當作類別型，也可當作數值型特徵 (True:1 / False: 0)

排序型特徵



- 例如名次 / 百分等級，有大小關係，但並非連續數字
- 通常當作數值型特徵處理，因為當作類別型會失去排序資訊

其他特徵類型 (2 / 2)

時間型特徵

2018/12/15 09:00:00

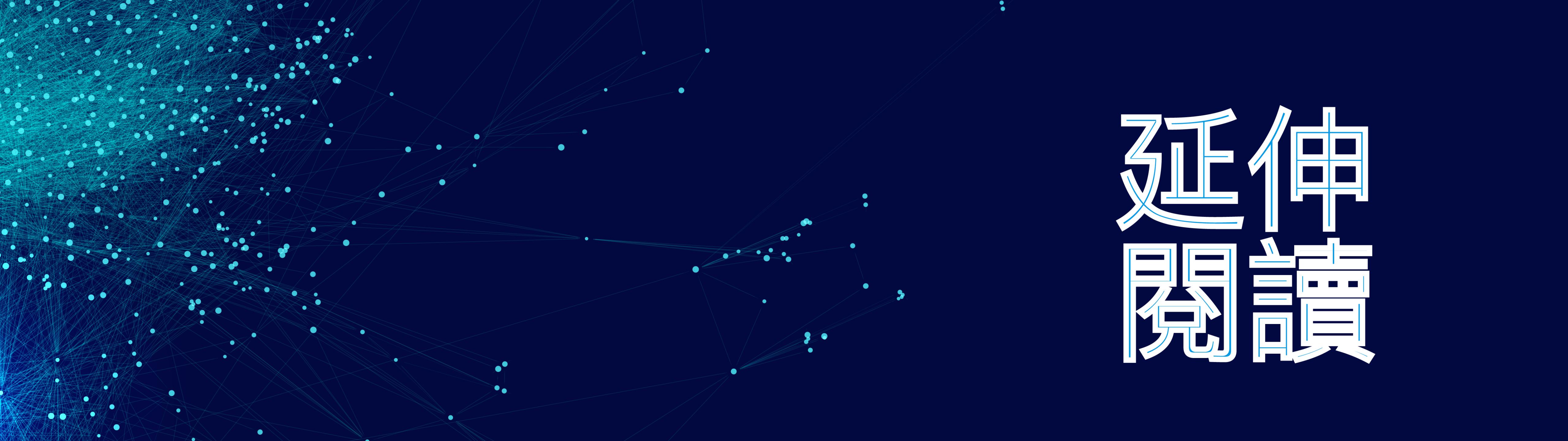
- 雖然時間型特徵可當作數值型特徵或類別型特徵，但都不適合
 - 取總秒數雖可變為數值，但會失去週期性 (ex 月 / 星期)
 - 使用本身可以當作類別，但會失去排序資訊，類別數量也過大
- 因此時間型特徵我們會個別於之後的課程中講解

補充說明

因為程式講解需要，會以 `cross_val_score` 顯示改善效果：分數越高表示效果越好，但不會在現階段講解這部分原理，有興趣提前了解的同學請研讀延伸閱讀內容：[k-fold cross validation](#)

重要知識點複習

- 資料中最常見的特徵類型是**數值型特徵**與**類別型特徵**，雖然還有二元特徵、排序型特徵、時間型特徵等多種特徵類型，但仍以前兩者為主
- **數值型特徵**：最容易轉成特徵，但需要注意很多**細節**
- **類別型特徵**：通常一種類別對應一種分數，問題在**如何對應**
- **時間型特徵**：特殊之處在於有**週期性**
- 上述三種特徵，會在之後的課程講述對應的特徵工程



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

Python Tutorial 第二堂 - 數值與字串型態

網頁連結

焦點主要在於：搭配一些 python 語法，向同學更清楚的展示型態的差異，如果同學看完今天的課程仍不是很了解，建議可以搭配此內容理解，當然如果覺得已經能自己分辨，就可以跳過這份參考資料了。

```
~$ python3.5
Python 3.5.0+ (default, Oct 11 2015, 09:05:38)
[GCC 5.2.1 20151010] on linux
Type "help", "copyright", "credits" or "license" for more information.

>>> type(1)          # 1 是什麼型態?
<class 'int'>
>>> type(1111111111111111111111111111111111111111111111111111111111111111) # 很長的整數也是 int
<class 'int'>
>>> type(3.14)      # 浮點數是 float 型態
<class 'float'>
>>> type(True)     # 布林值是 bool 型態
<class 'bool'>
>>> type(3 + 4j)   # 支援複數的 complex 型態
<class 'complex'>
>>> 2 ** 100       # 2 的 100 次方
1267650600228229401496703205376
>>>
```

推薦延伸閱讀

Python3.7.2 : Built-in Types Python 官方說明

網頁連結

這份官方網頁內容資料則是從語言的角度，列舉這些型態，以及對應的操作，因為是官方網頁，會隨著版本更新而略有修改，建議同學有需要的時候(例如版本變更)，再來查詢即可，不用細讀。

Operation	Result	Notes	Full documentation
<code>x + y</code>	sum of <code>x</code> and <code>y</code>		
<code>x - y</code>	difference of <code>x</code> and <code>y</code>		
<code>x * y</code>	product of <code>x</code> and <code>y</code>		
<code>x / y</code>	quotient of <code>x</code> and <code>y</code>		
<code>x // y</code>	floored quotient of <code>x</code> and <code>y</code>	(1)	
<code>x % y</code>	remainder of <code>x / y</code>	(2)	
<code>-x</code>	<code>x</code> negated		
<code>+x</code>	<code>x</code> unchanged		
<code>abs(x)</code>	absolute value or magnitude of <code>x</code>		abs()
<code>int(x)</code>	<code>x</code> converted to integer	(3)(6)	int()
<code>float(x)</code>	<code>x</code> converted to floating point	(4)(6)	float()
<code>complex(re, im)</code>	a complex number with real part <code>re</code> , imaginary part <code>im</code> . <code>im</code> defaults to zero.	(6)	complex()
<code>c.conjugate()</code>	conjugate of the complex number <code>c</code>		
<code>divmod(x, y)</code>	the pair <code>(x // y, x % y)</code>	(2)	divmod()
<code>pow(x, y)</code>	<code>x</code> to the power <code>y</code>	(5)	pow()
<code>x ** y</code>	<code>x</code> to the power <code>y</code>	(5)	



解題時間

It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

