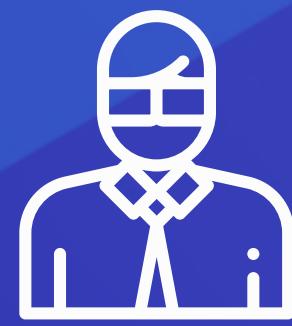




Day 27 特徵工程

時間型特徵



出題教練

陳明佑



知識地圖 特徵工程 時間型特徵

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

特徵縮放

特徵組合
特徵篩選

特徵評估

本日知識點目標

- 時間型特徵最常用的編碼方式
- 如何將時間最重要的特性「循環」改成特徵
- 時間常見的週期循環特徵有哪些，有什麼要注意的地方？

時間特徵分解 (1 / 2)

最常見的特殊欄位是時間欄位，想想看應該怎樣編碼？

時間戳記

2014-06-12 03:25:56

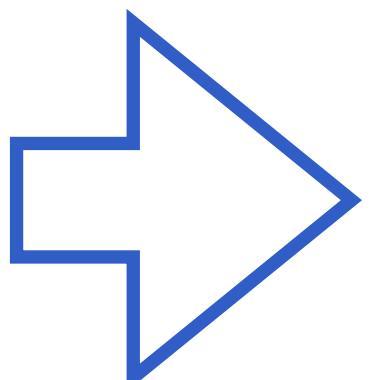
2011-07-16 01:19:59

2011-10-21 23:54:10

2015-02-03 10:42:03

2009-06-13 16:10:54

2010-08-09 14:46:03



?

時間特徵分解 (2 / 2)

最直覺的方式，就是依照原意義分欄處理，或加上第幾周或星期幾
但某些欄(例：分、秒)與目標值的關係很低，有沒有更有意義的特徵呢？

時間戳記	年	月	日	時	分	秒
2014-06-12 03:25:56	2014	6	12	3	25	56
2011-07-16 01:19:59	2011	7	16	1	19	59
2011-10-21 23:54:10	2011	10	21	23	54	10
2015-02-03 10:42:03	2015	2	3	10	42	3
2009-06-13 16:10:54	2009	6	13	16	10	54
2010-08-09 14:46:03	2010	8	9	14	46	3

週期循環特徵 (1 / 2)

時間也有週期的概念，可以用週期合成一些重要的特徵

聯想看看：有哪幾種時間週期，可串聯到一些可做特徵的性質？



- **年週期** 與春夏秋冬季節溫度相關
- **月週期** 與薪水、繳費相關
- **周週期** 與周休、消費習慣相關
- **日週期** 與生理時鐘相關

週期循環特徵 (2 / 2)

前述的週期所需數值都可由時間欄位組成，但還首尾相接

因此週期特徵還需以正弦函數(sin)或餘弦函數(cos)加以組合

例如：年週期 (正：冷 / 負：熱)

$$\cos((\text{月}/6 + \text{日}/180)\pi)$$

周週期 (正：精神飽滿 / 負：疲倦)

$$\sin((\text{星期幾}/3.5 + \text{小時}/84)\pi)$$

日週期 (正：精神飽滿 / 負：疲倦)

$$\sin((\text{小時}/12 + \text{分}/720 + \text{秒}/43200)\pi)$$

*註：此處小時是24小時制

時段特徵

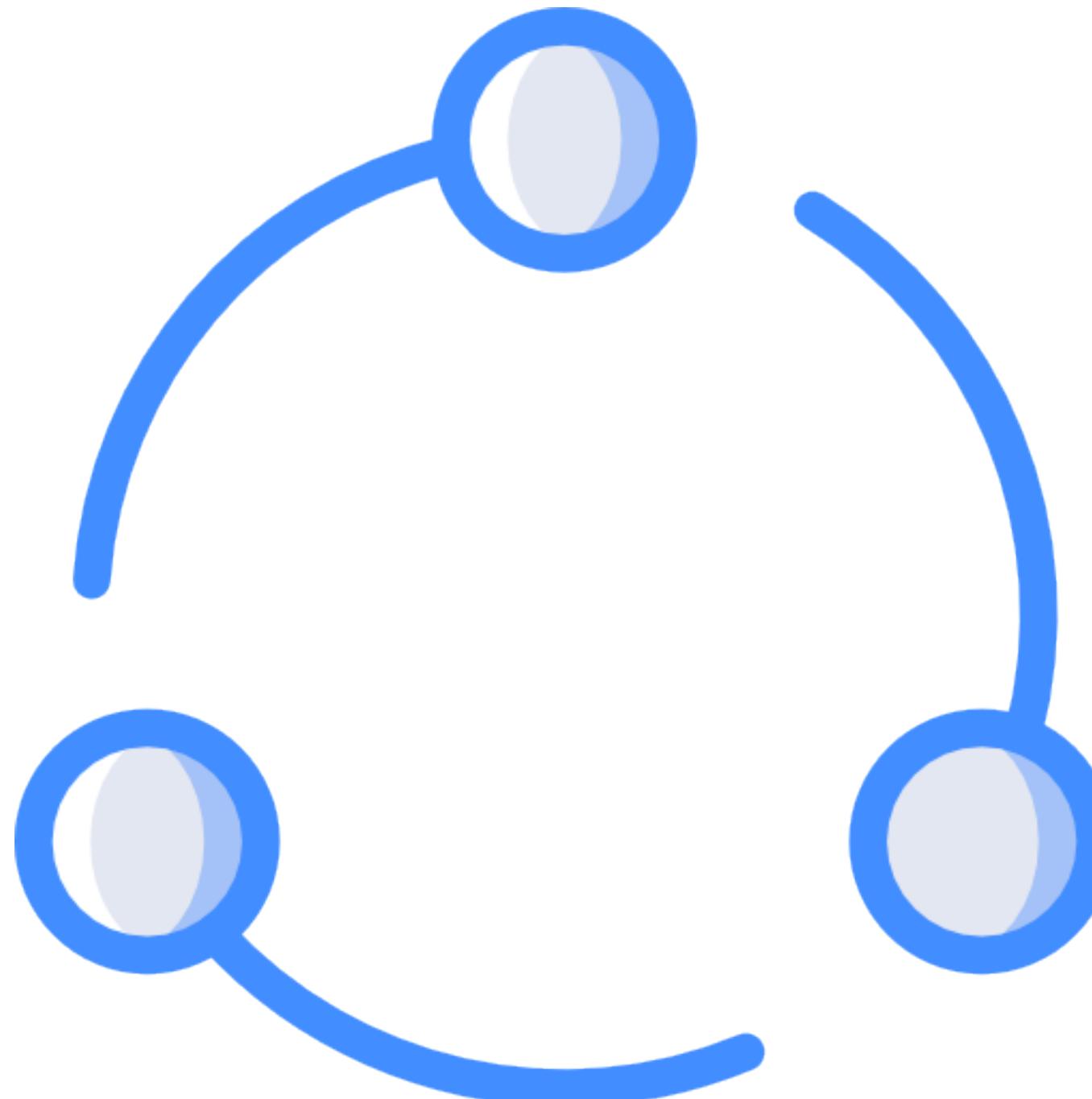
短暫時段內的事件計數，也可能影響事件發生的機率
如：網站銷售預測，點擊網站前 10分鐘 / 1小時 / 1天 的累計點擊量

以一筆 17:05 發生的網站瀏覽事件為例

同樣是1小時的統計，基礎分解會統計當日 17 時整個小時的點擊量
時段特徵則是會統計 16:05-17:04 的點擊量

兩者相比，後者較前者更為合理

重要知識點複習



- 時間型特徵最常用的是**特徵分解** - 拆解成年/月/日/時/分/秒的分類值
- 週期循環特徵**是將時間"循環"特性改成特徵方式, 設計關鍵在於**首尾相接**, 因此我們需要使用 \sin / \cos 等週期函數轉換
- 常見的週期循環特徵有 - 年週期(季節) / 周周期(例假日) / 日週期(日夜與生活作息), 要注意的是**最高與最低點的設置**



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-基礎-時間日期處理小結

wkiken [網頁連結](#)

- 這裡主要是摘錄一些 `datetime` 套件的時間欄位操作方式，使用這一個套件就能將各式時間格式轉成統一的時間物件，方便做轉換與特徵工程。

`datetime <=> string`

`datetime` -> `string`

```
>>> import datetime  
>>> datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")  
'2015-01-12 23:13:08'
```

`string` -> `datetime`

```
>>> import datetime  
>>> datetime.datetime.strptime("2014-12-31 18:20:10", "%Y-%m-%d %H:%M:%S")  
datetime.datetime(2014, 12, 31, 18, 20, 10)
```

推薦延伸閱讀

datetime – Basic date and time types

python官方文件 [網頁連結](#)

- 既然要使用 datetime 套件，詳細的內容當然要參考官方的文件囉。
- 除了各種細節的寫法，老師個人最常用到的還是 `.strptime`(將文字解析成時間物件) 以及 `.strftime` (轉出時間物件各欄位成文字)，以及右邊這張表(網頁最下方)，其中除了常見的年月日時分秒外，還有 `%w`(星期幾)，`%U`(第幾周) 這種方便的轉換函數呢!

Directive	Meaning	Example	Notes
<code>%a</code>	Weekday as locale's abbreviated name.	Sun, Mon, ..., Sat (en_US); So, Mo, ..., Sa (de_DE)	(1)
<code>%A</code>	Weekday as locale's full name.	Sunday, Monday, ..., Saturday (en_US); Sonntag, Montag, ..., Samstag (de_DE)	(1)
<code>%w</code>	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	0, 1, ..., 6	
<code>%d</code>	Day of the month as a zero-padded decimal number.	01, 02, ..., 31	
<code>%b</code>	Month as locale's abbreviated name.	Jan, Feb, ..., Dec (en_US); Jan, Feb, ..., Dez (de_DE)	(1)
<code>%B</code>	Month as locale's full name.	January, February, ..., December (en_US); Januar, Februar, ..., Dezember (de_DE)	(1)
<code>%m</code>	Month as a zero-padded decimal number.	01, 02, ..., 12	
<code>%y</code>	Year without century as a zero-padded decimal number.	00, 01, ..., 99	
<code>%Y</code>	Year with century as a decimal number.	0001, 0002, ..., 2013, 2014, ...	(2)



解題時間

It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

