探索性資料分析(EDA)_從資料中生成特徵

簡報閱讀 範例與作業 問題討論 學習心得(完成)

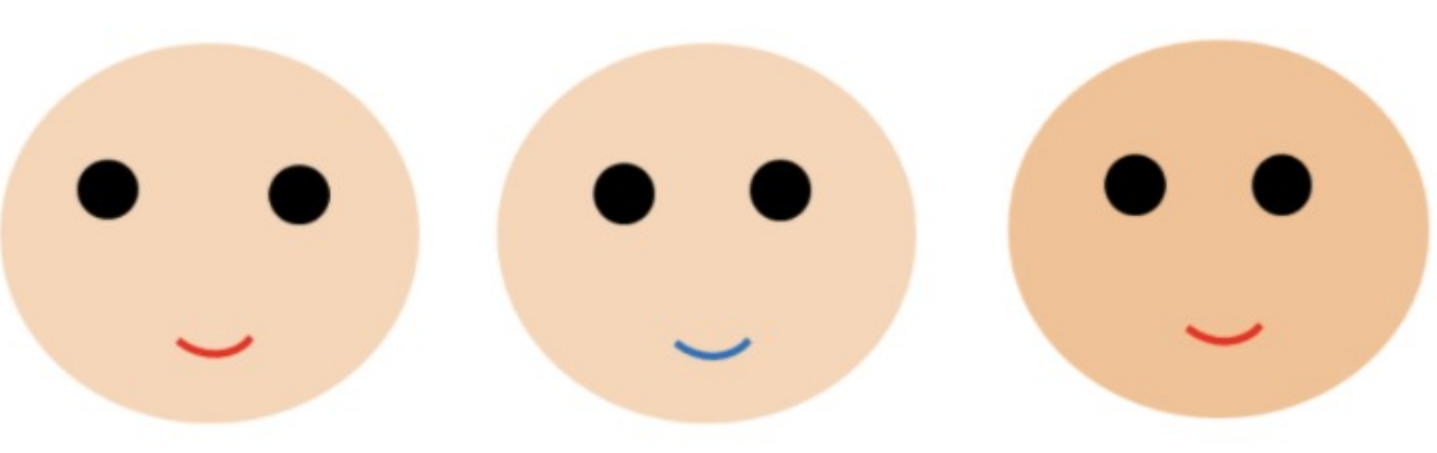
- 重要知識點 >
- 觀察 >
- 什麼是特徵？ >
- 什麼是特徵工程？ >
- 特徵工程可以分坐兩大類 >

重要知識點

- 掌握特徵的定義
- 掌握不同情境下，產生特徵方法
- 運用 python 產生特徵

觀察

我們來做一個小測驗，下面有一張圖，每一張圖的人臉，都有三種原始資料，眼睛型態，嘴巴顏色和膚色，你能在5秒內，找出某些資料能分辨出下列三個人？請回答你使用那些方法來判斷？



你答對了嗎？答案是**嘴巴顏色和膚色**，不知道你是怎麼找出**嘴巴顏色和膚色**？
從眼睛型態，嘴巴顏色和膚色中，選取出嘴吧顏色與膚色來分辨這三個人，就是一種特徵工程。

什麼是特徵？

然而透過上面小小的例子，你有發現特徵有什麼特性？

- 在原始資料集中**有變化性**，才能稱為特徵
- 透過這些特徵能把目標做清楚的分類與預測，才能稱為好的特徵。

什麼是特徵工程？

特徵工程是基於原始資料中創造出特徵，藉此改善模型性能的過程。

特徵工程可以分坐兩大類

- 特徵工程可以分成兩大類，包含衍生和添加。
- 衍生：以現有收集的資料為主，透過探索性分析，了解資料與目標之間的關係後，產生出特徵。
- 添加：現有收集資料以外的資訊，外部數據很多情況下沒有被重充分利用，實際上它們可以為模型的性能帶來最重大的突破。
- 今天的課程中，以衍生為主，說明如何透過 python 語法產生出衍生的特徵。

Step1：找出變化性的特徵

眼睛在三個人的臉上都是一樣的，這就是代表沒有變化性，所以我們無法用眼睛這個資料來判斷這三個人。

找出變化性的特徵，我們分兩個步驟來看：
- 對資料而言，怎麼找出具有變異性的資料？
- 對目標變數而言，怎麼找出與目標變數具有相關性的資料？

對資料而言，怎麼找出具有變異性的資料？

- 怎麼找出具有變異性的資料，大致上可以根據連續型資料與離散型資料來看。
- 運用**四分位數**、**全距**、**百分位數**、**標準差**、**變異數**，分析連續型資料的變異性。
- 運用**類別數量統計**，分析離散型資料資料的變異性。

PS：大家可以回顧，敘述統計的單元內容。

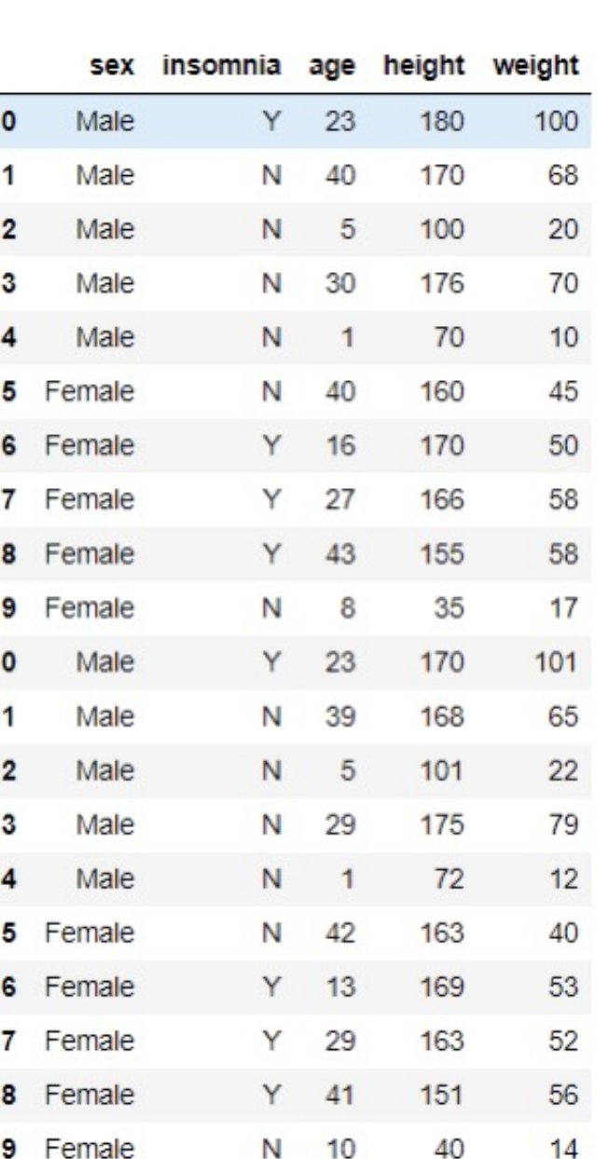
怎麼找出與目標變數具有相關性的資料？

運用昨天的課程內容，透過挖掘資料和目標資料是否具有相關性

- 如是高度相關，這個資料可以單獨為特徵。
- 如果中低相關，這個資料可能需要做轉換，才有可能變成特徵。

課程案例

20 筆資料，收集到 5 種資料，包含
- sex：性別
- insomnia：失眠
- age：年齡
- height：身高
- weight：體重



	sex	insomnia	age	height	weight
0	Male	Y	23	180	100
1	Male	N	40	170	68
2	Male	N	5	100	20
3	Male	N	30	176	70
4	Male	N	1	70	10
5	Female	N	40	160	45
6	Female	Y	16	170	50
7	Female	Y	27	166	58
8	Female	Y	43	155	58
9	Female	N	8	35	17
10	Male	Y	23	170	101
11	Male	N	39	168	65
12	Male	N	5	101	22
13	Male	N	29	175	79
14	Male	N	1	72	12
15	Female	N	42	163	40
16	Female	Y	13	169	53
17	Female	Y	29	163	52
18	Female	Y	41	151	56
19	Female	N	10	40	14

此資料集中，目標資料為失眠這一個欄位，我們想建立一個失眠的預測模型。

中低相關的資料，我們怎麼更進一步萃取出可用特徵？

我們可以透過衍生資料的方法，把原始資料做一些轉換，萃取出和目標變數相關的特徵，大致上可以分成以下幾種類型， 稱作 ICR。

步驟：
- 指示器變量(Indicator)
- 資料組合(Combination)
- 資料重新定義(Reshape)

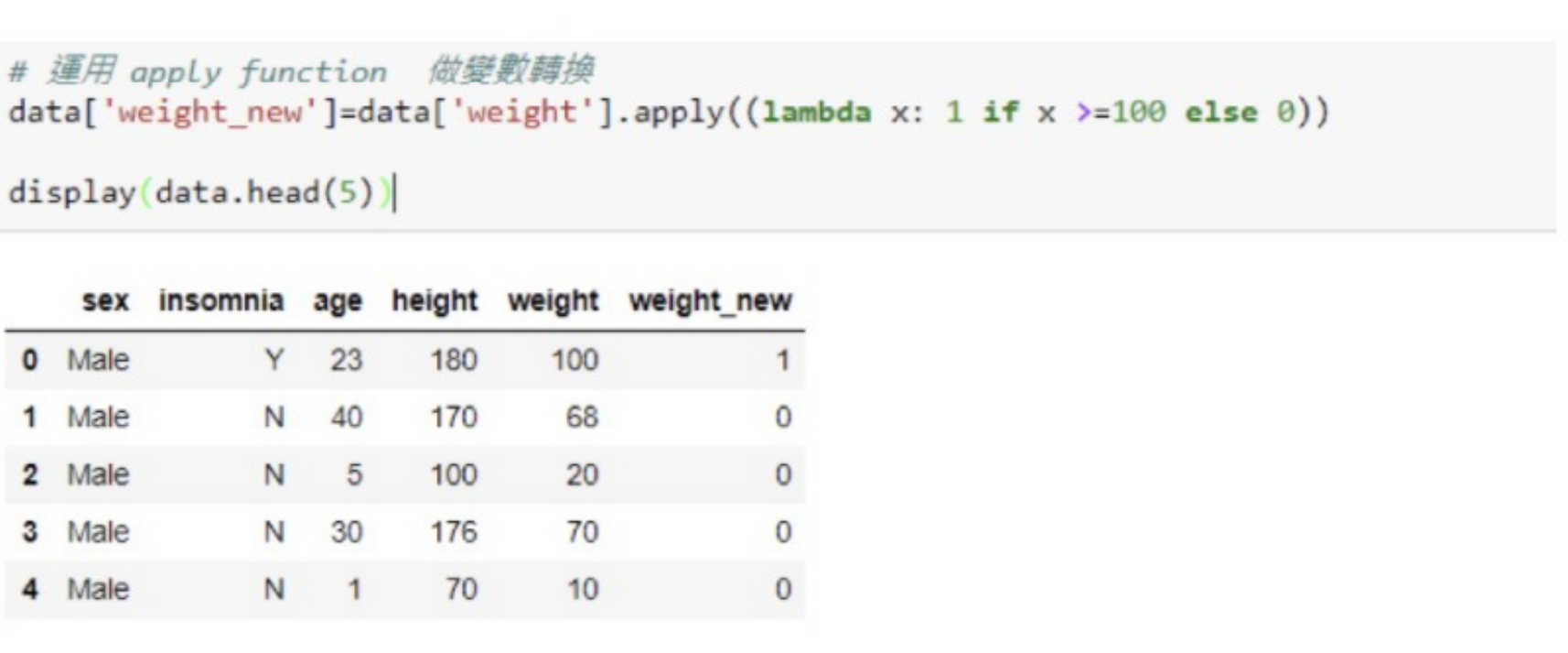
指示器變量(Indicator)

假設透過文獻分析發現，體重和失眠有高度相關性，當體重超過 100 公斤時，則得到失眠的機會會大於體重小於100公斤，則我們必須要產生一個新的資料。

$$weight_{new} = \begin{cases} 1, weight \geq 100 \\ 0, weight < 100 \end{cases}$$

Python - 指示器變量(Indicator)

透過 apply function 做指示器變量轉換



```
# 運用 apply function   做變數轉換
data['weight_new']=data['weight'].apply((lambda x: 1 if x >=100 else 0))

display(data.head(5))
```

	sex	insomnia	age	height	weight	weight_new
0	Male	Y	23	180	100	1
1	Male	N	40	170	68	0
2	Male	N	5	100	20	0
3	Male	N	30	176	70	0
4	Male	N	1	70	10	0

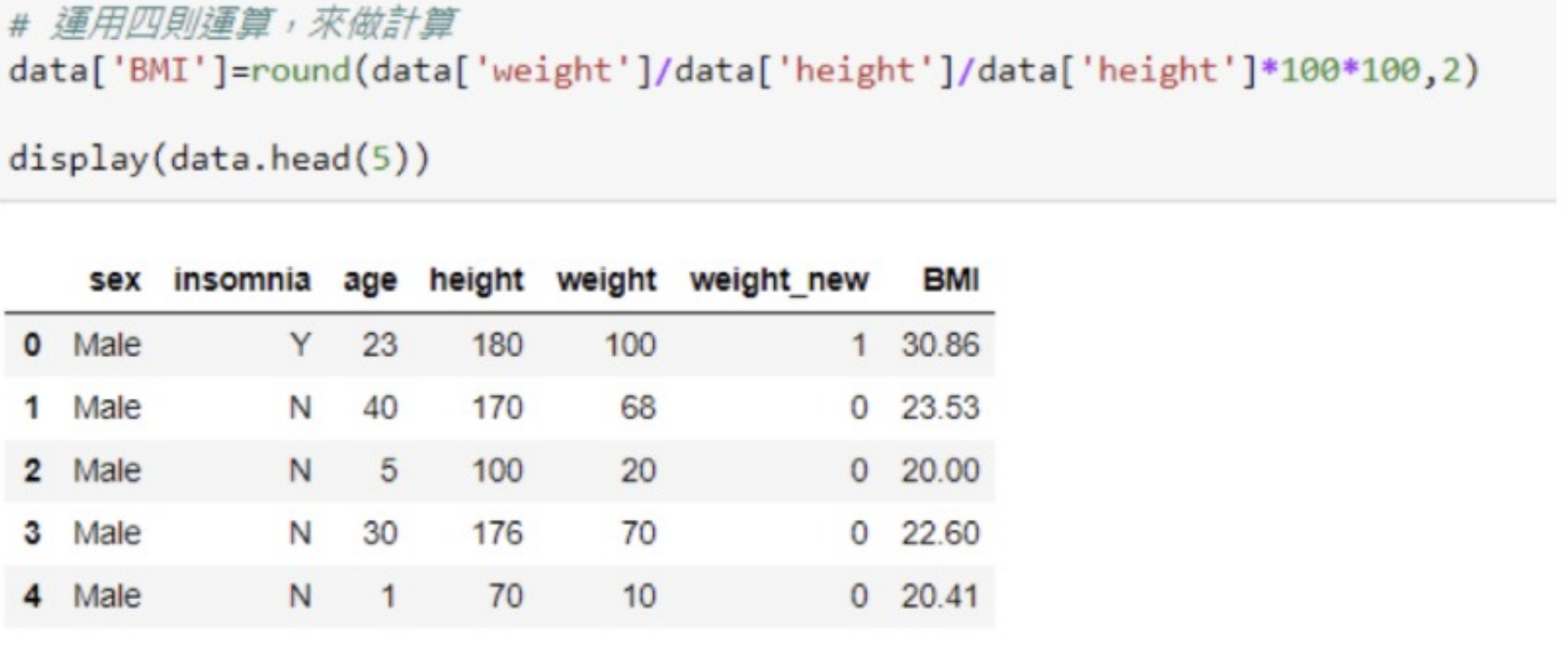
資料組合(Combination)

假設透過文獻分析發現，失眠和體重和身高所組成的BMI指數相關時，則我們可以根據資料中的體重和身高，做重新的組合與四則運算，產生出 BMI 的資料，在透過 BMI 這個新資料預測失眠，而 BMI 就是預測失眠的特徵之一。

$$BMI = \frac{體重(公斤)}{身高(公尺)^2}$$

Python - 資料組合(Combination)

資料組合，透過資料欄位間的四則運算產生出。



```
# 運用四則運算，來做計算
data['BMI']=round(data['weight']/data['height']/data['height']*100*100,2)

display(data.head(5))
```

	sex	insomnia	age	height	weight	weight_new	BMI
0	Male	Y	23	180	100	1	30.86
1	Male	N	40	170	68	0	23.53
2	Male	N	5	100	20	0	20.00
3	Male	N	30	176	70	0	22.60
4	Male	N	1	70	10	0	20.41

資料重新定義(Reshape)

資料收集時間長度調整：
- 預測地下水水位，時雨量比 10 分鐘及時雨量還好，沒有時間遞延問題，透過調整增強數據所能表達的信息。

數值到分類的映射：
- 可以將年齡，對應成兒童、青少年與成年的資料。

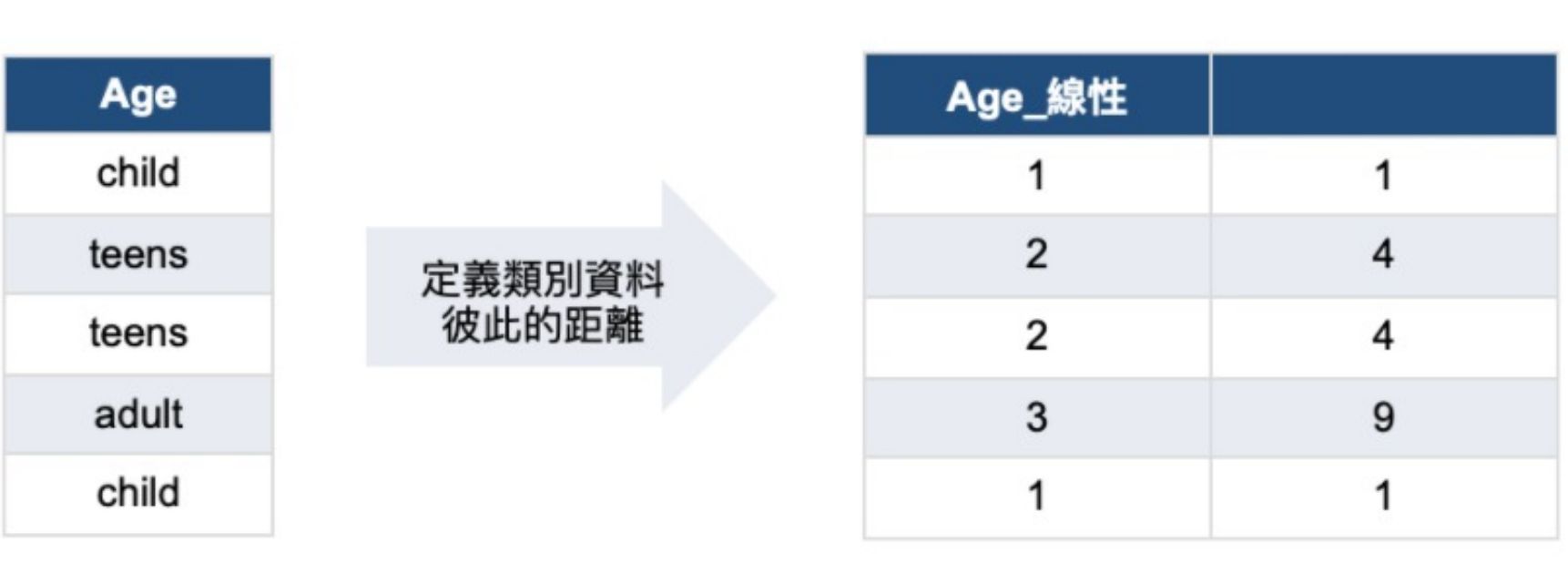
合併稀疏分類：
- 發現年齡中，某一個年齡層人數偏少，可以做合併的動作。

表達類別型資料的距離：
- 定義類別資料距離：比如年齡資料，兒童、青少年與成年可轉換為1，2，3。

創造虛擬資料：
- 這取決與你選擇的機器學習算法，如果是以距離來量測資料的遠近，則需將類別特徵轉換到虛擬變量中去，稱作 one-hot encoding。

重新定義類別資料距離

- 假設我們的年紀，從連續變成離散，分成兒童、青少年、成年人，三種類型。
- 假設這三個群組對於失眠的貢獻不同，年紀越大失眠狀態越嚴重，我們可以轉譯成 1，2，3 也可以轉譯成 1，4，9。



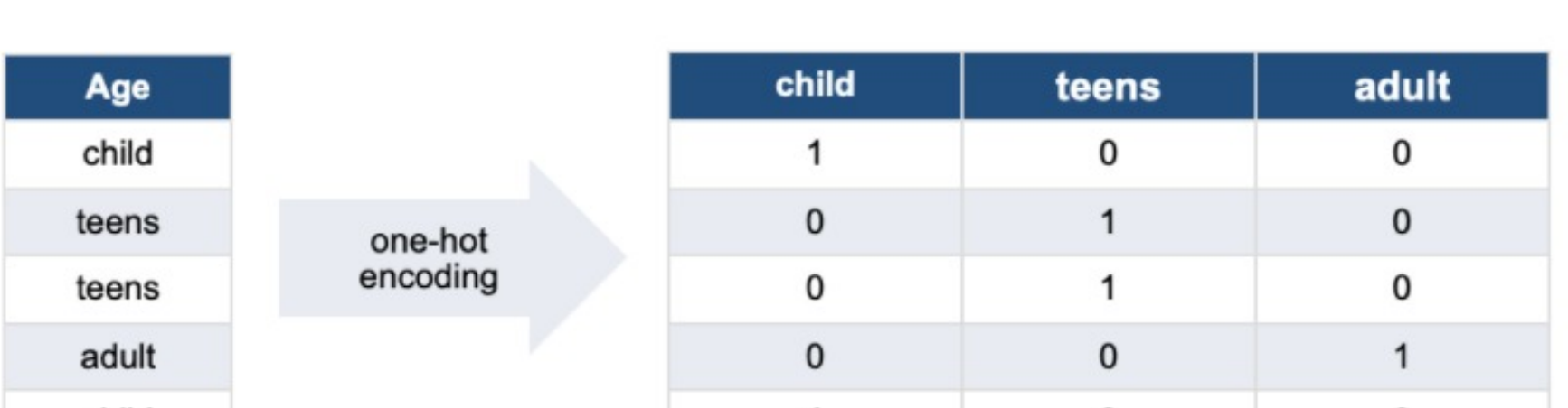
Age
child
teens
teens
adult
child

定義類別資料
彼此的距離

Age_線性	
1	1
2	4
2	4
3	9
1	1

創造虛擬變量 - One hot encoding

- 假設我們的年紀，從連續變成離散，分成兒童、青少年、成年人，三種類型。
- 經過 one hot encoding 會產生三個變數。



Age
child
teens
teens
adult
child

one-hot
encoding

child	teens	adult
1	0	0
0	1	0
0	1	0
0	0	1
1	0	0

Python - 數值到分類的映射

```
## 數值到分類的映射
# 運用 apply function   做變數轉換
def age_map(x):
    if(x<=12):
        return('child')
    else:
        if(x<=18):
            return('teens')
        else:
            return('adult')

data['age_category']=data['age'].apply(age_map)

display(data)
```

	sex	insomnia	age	height	weight	weight_new	BMI	age_category
0	Male	Y	23	180	100	1	30.86	adult
1	Male	N	40	170	68	0	23.53	adult
2	Male	N	5	100	20	0	20.00	child
3	Male	N	30	176	70	0	22.60	adult
4	Male	N	1	70	10	0	20.41	child
5	Female	N	40	160	45	0	17.58	adult

定義一個轉換函數，運用 apply 函數，將數值資料轉換成類別型資料。

Python - 合併稀疏分類

- 透過交叉列連表，統計各類型的資料筆數，發現有小於5的資料點，建議合併。
- 透過定義新的轉換函數，進行調整。

```
## 合併稀疏分類
contTable = pd.crosstab(data['age_category'], data['insomnia'])
contTable
# 有兩個零的存在，太過稀疏，有時候會將 age_category child  和 teens
```

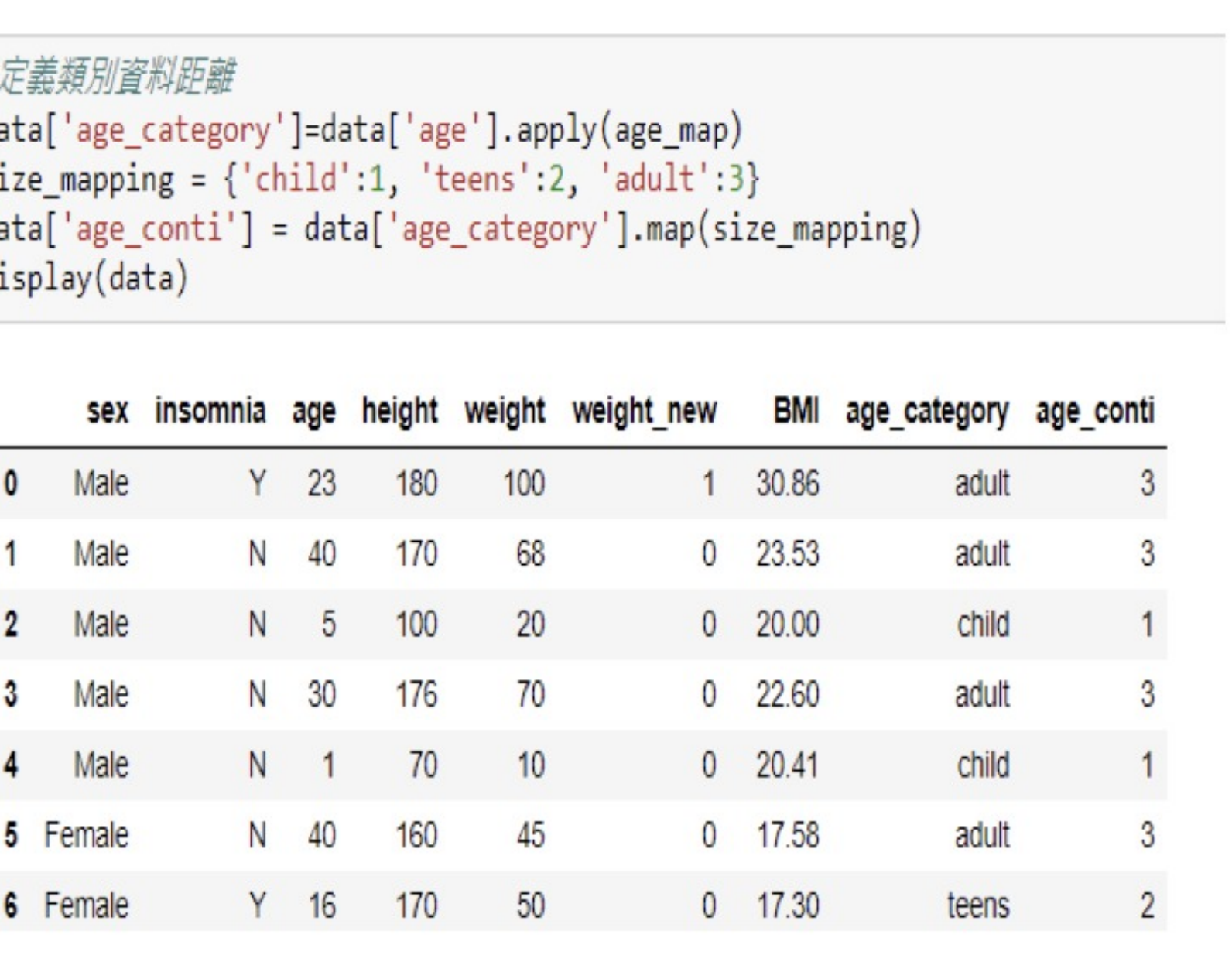
insomnia	N	Y
age_category		
adult	6	6
child	6	0
teens	0	2

```
##現在假設要合併的話可以怎麼做?
#產生一個新的 mapping function  然後做調整
def age_map_2(x):
    if(x<=18):
        return('child_teens')
    else:
        return('adult')

data['age_category']=data['age'].apply(age_map_2)

display(data)
```

	sex	insomnia	age	height	weight	weight_new	BMI	age_category
0	Male	Y	23	180	100	1	30.86	adult
1	Male	N	40	170	68	0	23.53	adult
2	Male	N	5	100	20	0	20.00	child_teens
3	Male	N	30	176	70	0	22.60	adult

Python - 定義類別資料距離

運用map 函數，把離散型資料應對成連續型資料。

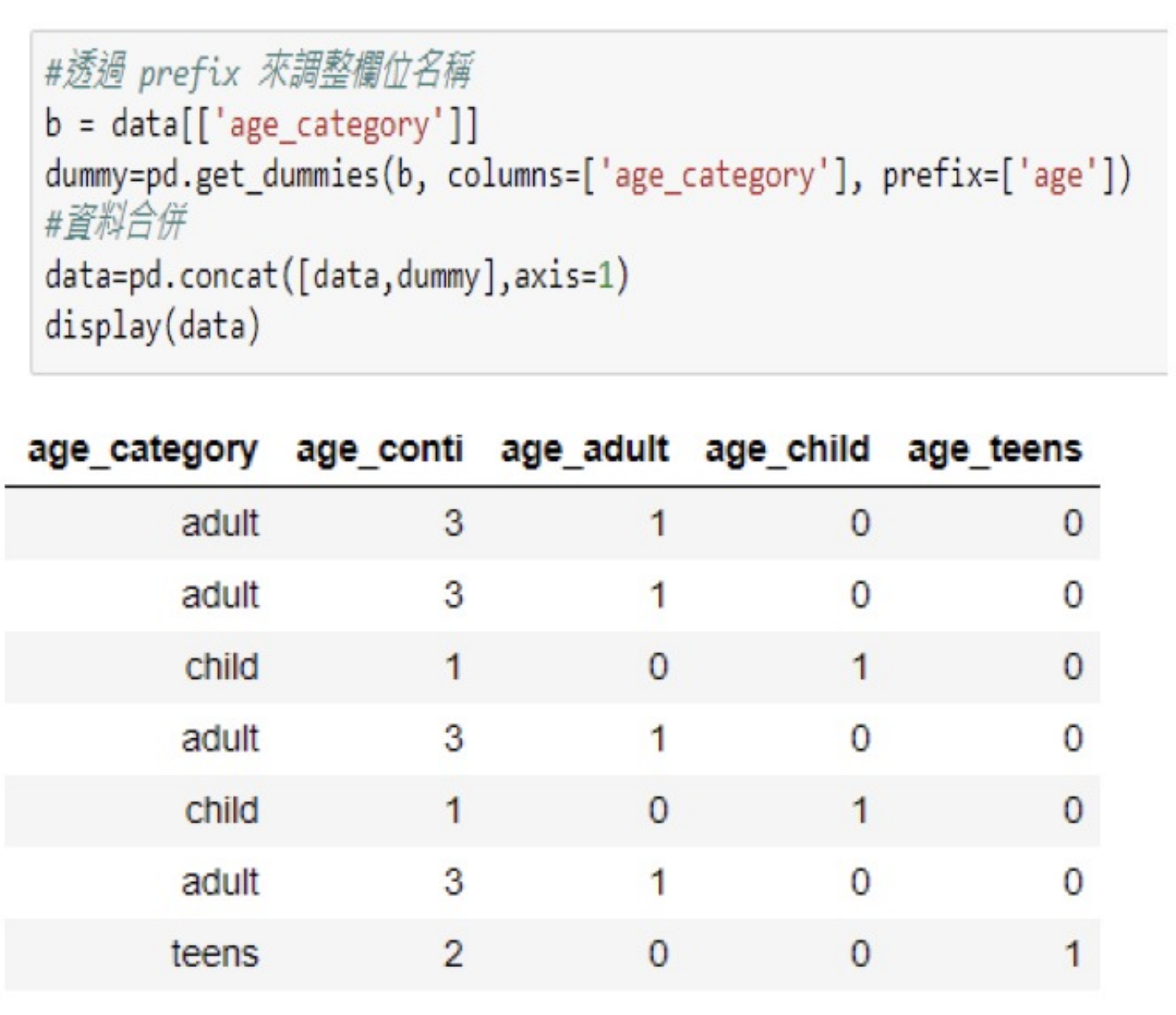


```
#定義類別資料距離
data['age_category']=data['age'].apply(age_map)
size_mapping = {'child':1, 'teens':2, 'adult':3}
data['age_conti'] = data['age_category'].map(size_mapping)
display(data)
```

	sex	insomnia	age	height	weight	weight_new	BMI	age_category	age_conti
0	Male	Y	23	180	100	1	30.86	adult	3
1	Male	N	40	170	68	0	23.53	adult	3
2	Male	N	5	100	20	0	20.00	child	1
3	Male	N	30	176	70	0	22.60	adult	3
4	Male	N	1	70	10	0	20.41	child	1
5	Female	N	40	160	45	0	17.58	adult	3
6	Female	Y	16	170	50	0	17.30	teens	2

Python - 創建虛擬資料

1. 取出要創建虛擬資料的欄位
2. 透過 pd.get_dummies，進行轉換。
3. 在將資料集合合併。



```
#透過 prefix 來調整欄位名稱
b = data[['age_category']]
dummy=pd.get_dummies(b, columns=['age_category'], prefix=['age'])
#資料合併
data=pd.concat([data,dummy],axis=1)
display(data)
```

age_category	age_conti	age_adult	age_child	age_teens
adult	3	1	0	0
adult	3	1	0	0
child	1	0	1	0
adult	3	1	0	0
child	1	0	1	0
adult	3	1	0	0
teens	2	0	0	1

產生出特徵後，然後？

記得做完特徵轉換後，都要去檢驗心產生出來的特徵和目標特徵，是否具有相關性。

方法	Pearson皮爾森	Cramer's V克雷莫	
變數特性	兩個成對連續變數	兩個成對離散變數	成對的一個連續 一個離散型變數

知識點回顧

今天的課程中，特徵由原始資料而來，其具有以下特性
- 有變化性
- 有辨識能力與預測能力

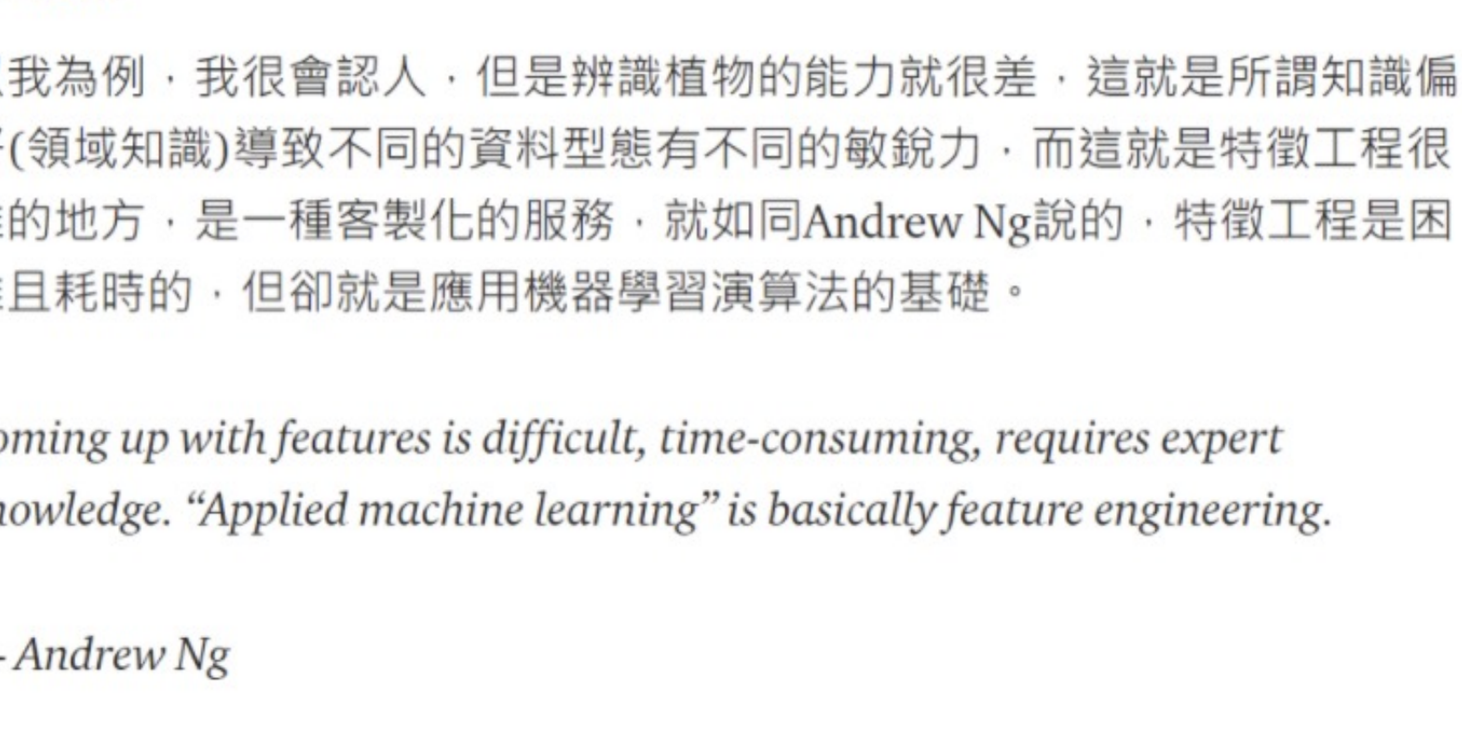
特徵工程可以分成兩大類，包含衍生和添加，衍生包含指示器變量(Indicator)、資料組合(Combination) 與資料重新定義(Reshape)。

回到今天的程式範例

看投影片的結果怎麼一步一步實作出來

延伸閱讀

拿到資料就可以套用各式各樣的演算法模型？
網站：reurl



> 以我為例，我很會認人，但是辨識植物的能力就很差，這就是所謂知識偏好(領域知識)導致不同的資料型態有不同的敏銳力，而這就是特徵工程很難的地方，是一種客製化的服務，就如同Andrew Ng說的，特徵工程是困難且耗時的，但卻就是應用機器學習演算法的基礎。
>
> *Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.*
>
> *— Andrew Ng*

下一步：閱讀範例與完成作業