資料前處理

製作人：黃宥輔

什麼是資料前處理？

- 將原始資料整理成機器學習模型適合的格式
- 改善模型執行效能
- 改善模型效能

為什麼要做資料前處理？

- 大多時候，在資料科學的研究中，50%到80%的時間都是在對資料進行整理
- 資料決定了模型的品質上限，好的資料才有可能產出好的模型，故需要避免GIGO(garbage in garbage out)的窘境

為什麼要做資料前處理？

情況	問題	解決
非結構化的資料	無法放入機器學習模型	將資料結構化
資料為文字	電腦直接無法辨識文字	將文字進行統計、字詞分析處理，轉換成
資料有空值、雜質	可能使模型無法運行 或是邏輯錯誤	將資料進行填空、轉換
特徵過多	模型訓練緩慢，可能會干擾模型	進行特徵工程，篩選特徵

資料有無結構

種類	說明	範例
結構化資料	• 有組織的資料 • 可以分為觀察值(observations)和特徵(characteristics) • 通常可以用表格形式表示，列(row)為觀察值，欄(column)為特徵	• 證交所公開資料
非結構化資料	• 沒有組織、未經處理的資料 • 通常不能以表格顯示	• 圖片或影音資料 • 一疊報紙或雜誌

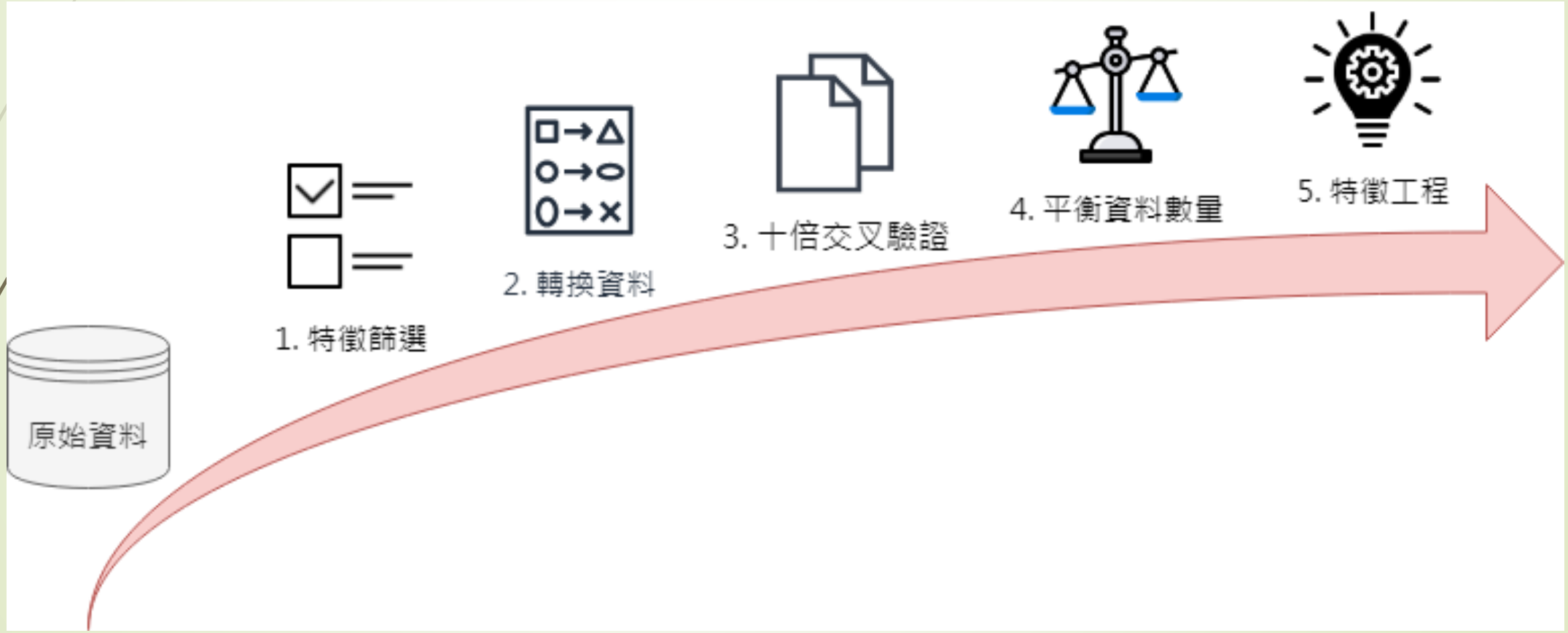
特徵與標籤

- 在監督式機器學習中，需要有特徵與標籤才能進行訓練與驗證

項目	別稱	說明
特徵(Feature)	X	敘述一個實體或結果的屬性
標籤(Label)	Y、反應變數	一個實體或結果 例如：有無感染、是哪種花朵

資料前處理的流程

- 若把建置機器學習模型比喻成做菜，
  模型預測的成果為菜餚，
  前處理即是備料



何謂特徵(Feature)？

- 特徵為對於機器學習過程有意義的屬性，即對於一個實體或結果的描寫
- 如四肢行走、叫聲為汪汪聲，我們可以根據這些特徵來臆測這個實體是一條狗

使用套件簡介

套件	功能簡介
numpy	Python的資料處理套件，可以輕鬆建構向量、矩陣等資料型態
pandas	基於numpy開發，資料整理、Python中的Excel，可以與其他套件結合
scikit-learn	簡稱sklearn，機器學習套件，也提供前處理的函式與模型檢定
keras	神經網路套件，可以自由建構各種神經網路模型

0. 了解資料集背景

資料集背景

- 資料來源(可靠性、完整性)
  - ✓不可靠的資料 – 一錯百錯
  - ✓不完整的資料 – 盲人摸象
- 資料領域知識(Domain Knowledge)
  - ✓發揮資料長處
  - ✓釐清因果關係
- 決定研究議題
  - ✓決定解釋資料的角度
  - ✓決定結論的方向

鯊魚冰淇淋

- 研究結果顯示，全國冰淇淋的銷量越好，會導致全國越多人被鯊魚攻擊

鯊魚冰淇淋 – 原因探討

- 若只考慮這兩個因素，推斷出以下可能關聯
  1. 吃冰淇淋後下水游泳身上有特別的味道，讓鯊魚更容易攻擊？
  2. 容易被鯊魚攻擊的人喜歡吃冰淇淋？
  3. 吃某些口味或廠牌冰淇淋特別容易被鯊魚攻擊？

鯊魚冰淇淋 – 加入其他因素

- 此研究為夏天進行的，高溫炎熱
- 加入以上因素，推斷出以下結論

  天氣炎熱，冰淇淋銷量稱加，去海邊游泳的人也增加，故此冰淇淋銷量與鯊魚攻擊案件數因果關係薄弱

鯊魚冰淇淋 –研究議題

- 研究目標改為研究某海灘的資料，當地政府為了振興觀光，與高檔冰淇淋廠商合作，推出冰淇淋1折優惠
- 根據因素，推斷出以下結論

  政策奏效，吸引大批人潮前來觀光，去海邊游泳的人也增加，遊客更可能被鯊魚攻擊，故冰淇淋銷量與鯊魚攻擊案件數有一定程度的因果關係

1. 特徵篩選

1. 特徵篩選

- 去除與研究議題不相關的特徵
  如：顧客編號
- 去除缺值過多的資料
  如：去除缺值超過 50% 以上的特徵
- 去除其他與研究主題不相干的特徵
  如：預測個人信用違約時，排除「剩餘未償還金額」，因為此為事後資料，不是研究主題，也會干擾研究

皮馬印地安人糖尿病預測資料

懷孕	葡萄糖	血壓	皮膚厚度	胰島素	BMI	糖尿病譜系功能	年齡	是否有糖尿病
6	148	72	35		33.6	0.627	50	是
1	85	66	29		26.6	0.351	31	否
8	183	64			23.3	0.672	32	是
1	89	66	23	94	28.1	0.167	21	否
	137	40	35	168	43.1	2.288	33	是
5	116	74			25.6	0.201	30	否
3	78	50	32	88	31	0.248	26	是
10	115				35.3	0.134	29	否
2	197	70	45	543	30.5	0.158	53	是
8	125	96				0.232	54	是
4	110	92			37.6	0.191	30	否
10	168	74			38	0.537	34	是

皮馬印地安人糖尿病預測資料

屬性	判斷
懷孕	與標籤相關，予以保留
葡萄糖	
血壓	
皮膚厚度	
胰島素	
BMI	
糖尿病譜系功能	
年齡	
是否有糖尿病	作為標籤

皮馬印地安人糖尿病預測資料

行號	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	df = pd.read_excel("iris.xlsx")
4	print(df)
5	df.info( )

皮馬印地安人糖尿病預測資料讀取

	懷孕	葡萄糖	血壓	皮膚厚度	胰島素	BMI	糖尿病譜系功能	年齡	是否有糖尿病
0	6.0	148.0	72.0	35.0	NaN	33.6	0.627	50	是
1	1.0	85.0	66.0	29.0	NaN	26.6	0.351	31	否
2	8.0	183.0	64.0	NaN	NaN	23.3	0.672	32	是
3	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21	否
4	NaN	137.0	40.0	35.0	168.0	43.1	2.288	33	是
..	...	...	...	...	...	...	...	..	...
763	10.0	101.0	76.0	48.0	180.0	32.9	0.171	63	否
764	2.0	122.0	70.0	27.0	NaN	36.8	0.340	27	否
765	5.0	121.0	72.0	23.0	112.0	26.2	0.245	30	否
766	1.0	126.0	60.0	NaN	NaN	30.1	0.349	47	是
767	1.0	93.0	70.0	31.0	NaN	30.4	0.315	23	否

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   懷孕        657 non-null    float64
 1   葡萄糖       763 non-null    float64
 2   血壓        733 non-null    float64
 3   皮膚厚度      541 non-null    float64
 4   胰島素       394 non-null    float64
 5   BMI       757 non-null    float64
 6   糖尿病譜系功能  768 non-null    float64
 7   年齡        768 non-null    int64
 8   是否有糖尿病    768 non-null    object
dtypes: float64(7), int64(1), object(1)
memory usage: 54.1+ KB
```

讀取資料練習

鳶尾花(iris)資料

花萼長度(cm)	花萼寬度(cm)	花瓣長度(cm)	花瓣寬度(cm)	屬種
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa

鳶尾花(iris)屬性

屬性	判斷
花萼長度(cm)	與標籤相關，予以保留
花萼寬度(cm)	
花瓣長度(cm)	
花瓣寬度(cm)	
屬種	作為標籤

讀取鳶尾花(iris)資料

行號	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	df = pd.read_csv("iris.csv")
4	print(df)
5	df.info( )

鳶尾花(iris)讀取資料

	花萼長度(cm)	花萼寬度(cm)	花瓣長度(cm)	花瓣寬度(cm)	屬種
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

[150 rows x 5 columns]

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   花萼長度(cm)  150 non-null    float64
 1   花萼寬度(cm)  150 non-null    float64
 2   花瓣長度(cm)  150 non-null    float64
 3   花瓣寬度(cm)  150 non-null    float64
 4   屬種         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

2. 轉換資料

轉換資料

- 將各種資料型別轉換成比率尺度

種類	處理範例	處理順序
名目尺度 (Norminal)	轉換為虛擬變數(dummy variable)	2
順序尺度 (Ordinal)	將資料進行編號， 如：將5個職等依據職位高低，編碼為1到5的形式	1
區間尺度 (Interval)	找一個共同的規則，將資料進行差值計算 如：計算股票的持有天數，以平倉日期減去建倉日期 計算年齡，以某一個日期與生日的差異天數	
比率尺度 (Ratio)	不須特別轉換類別，但需要填空、去雜質	3

轉換名目尺度資料

- 將每一種資料都給予一個特徵欄位
- 並以1代表該特徵項，其餘項目以0表示

水果	轉換 →	Apple	Banana	grape
Apple		1	0	0
Banana		0	1	0
Banana		0	1	0
grape		0	0	1

轉換名目尺度資料

- 通常會刪除其中一個特徵，減少特徵數量，因為N-1個特徵即可表達所有特徵項目

水果	轉換 →	Apple	Banana
Apple		1	0
Banana		0	1
Banana		0	1
grape		0	0

轉換順序尺度資料

- 根據職位的高低順序，將職位進行編號

職位	順序編號
董事長	1
總經理	2
經理	3
副理	4
職員	5

職位		職位
董事長		1
總經理		2
經理		3
副理		4
職員		5
經理	→	3
副理		4
職員		5
副理		4
職員		5
職員		5
職員		5

資料轉換

轉換方法	用途
pd.get_dummies(　資料, 　prefix =欄位名稱, 　drop_first = False)	將一個欄位轉換為虛擬變數 prefix 為轉換後前面加上的名稱 drop_first 為刪除轉換後的第一個欄位
mapper = { "A" : 0, "B" : 1 } df[欄位名稱].replace(mapper)	將特定的數值根據mapper進行轉換

資料轉換

行號	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	df = pd.read_csv("pima.csv")
4	print(df["是否有糖尿病"])
5	trans_data = pd.get_dummies(df["是否有糖尿病"], prefix = "是否有糖尿病")
6	print(trans_data)
7	trans_data = pd.get_dummies(df["是否有糖尿病"], prefix = "是否有糖尿病", drop_first = True)
8	print(trans_data)
9	
10	mapper = { '是' : 0, '否' : 1 }
11	print(df["是否有糖尿病"].replace(mapper))

資料轉換

原始資料

```
0      是
1      否
2      是
3      否
4      是
      ..
763    否
764    否
765    否
766    是
767    否
Name: 是否有糖尿病, Length: 768, dtype: object
```

轉換虛擬變數

```
     是否有糖尿病_否  是否有糖尿病_是
0              0            1
1              1            0
2              0            1
3              1            0
4              0            1
..           ...          ...
763            1            0
764            1            0
765            1            0
766            0            1
767            1            0

[768 rows x 2 columns]
```

轉換虛擬變數
刪除其中一欄

```
     是否有糖尿病_是
0              1
1              0
2              1
3              0
4              1
..           ...
763            0
764            0
765            0
766            1
767            0

[768 rows x 1 columns]
```

根據規則取代變數

```
0      0
1      1
2      0
3      1
4      0
      ..
763    1
764    1
765    1
766    0
767    1
Name: 是否有糖尿病, Length: 768, dtype: int64
```

資料轉換

行號	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	df = pd.read_csv("iris.csv")
4	print(df["屬種"])
5	trans_data = pd.get_dummies(df["屬種"], prefix = "屬種")
6	print(trans_data)
7	trans_data = pd.get_dummies(df["屬種"], prefix = "屬種", drop_first = True)
8	print(trans_data)
9	
10	mapper = {'setosa' : 0, 'versicolor' : 1, 'virginica' : 2}
11	print(df["屬種"].replace(mapper))

資料轉換

原始資料

```
0          setosa
1          setosa
2          setosa
3          setosa
4          setosa
          ...
145     virginica
146     virginica
147     virginica
148     virginica
149     virginica
Name: 屬種, Length: 150, dtype: object
```

轉換虛擬變數

	屬種_setosa	屬種_versicolor	屬種_virginica
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
..	...	...	...
145	0	0	1
146	0	0	1
147	0	0	1
148	0	0	1
149	0	0	1

轉換虛擬變數
刪除其中一欄

	屬種_versicolor	屬種_virginica
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
..	...	...
145	0	1
146	0	1
147	0	1
148	0	1
149	0	1

[150 rows x 2 columns]

根據規則取代變數

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: 屬種, Length: 150, dtype: int64
```

轉換區間尺度資料

生日
1996/12/13
1960/5/30
2001/1/8
2005/3/4
1970/8/25

→

年齡(天)
8,925
22,271
7,438
5,922
18,532

當天日期
2021/5/21

溫度	溫標
50	攝氏
185	華氏
80	攝氏
104	華氏
30	攝氏

→

攝氏溫度
50
85
80
40
30

處理比率尺度資料

- 此處為處理原則，但都可以依據情況做個別調整
- 處理完成後所有的特徵皆為比率尺度且無空值

問題	說明	處理方法
有雜質	比率尺度之中有不合理的數值 如：1. 用文字標示空值、缺值 2. 滿分100分的考試出現500分的分數	刪除文字或不合理之數值，形成空值
有空值	特徵並非全部都有數值	填入零、平均值
特徵之間範圍不一	每個特徵之間值的範圍不一 導致機器學習模型可能會有偏頗	進行標準化處理(下一節處理)

去雜質、填空值

轉換方法	用途
pd.to_numeric(資料, errors = "coerce")	將資料轉換為數值資料格式，即整數或小數 errors參數可以選擇要如何處理無法轉換的值 raise為發生錯誤，停止程式，預設為此 ignore為忽略錯誤 coerce為置換為空值
df.fillna(0) df.fillna(df.mean( ))	將空值填入自訂數值或是平均值 可以作用於整個 df 或 個別欄位

去雜質、填空值

行號	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	df = pd.read_csv("pima.csv“)
4	print(df)
5	df.info( )
6	df = df.fillna(df.mean( ))
7	print(df)
8	df.info( )

填空前結果

填空前資料

	懷孕	葡萄糖	血壓	皮膚厚度	胰島素	BMI	糖尿病譜系功能	年齡	是否有糖尿病
0	6.0	148.0	72.0	35.0	NaN	33.6	0.627	50	是
1	1.0	85.0	66.0	29.0	NaN	26.6	0.351	31	否
2	8.0	183.0	64.0	NaN	NaN	23.3	0.672	32	是
3	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21	否
4	NaN	137.0	40.0	35.0	168.0	43.1	2.288	33	是
..	...	...	...	...	...	...	...	..	...
763	10.0	101.0	76.0	48.0	180.0	32.9	0.171	63	否
764	2.0	122.0	70.0	27.0	NaN	36.8	0.340	27	否
765	5.0	121.0	72.0	23.0	112.0	26.2	0.245	30	否
766	1.0	126.0	60.0	NaN	NaN	30.1	0.349	47	是
767	1.0	93.0	70.0	31.0	NaN	30.4	0.315	23	否

[768 rows x 9 columns]

轉換前資訊

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   懷孕        657 non-null    float64
 1   葡萄糖      763 non-null    float64
 2   血壓        733 non-null    float64
 3   皮膚厚度    541 non-null    float64
 4   胰島素      394 non-null    float64
 5   BMI         757 non-null    float64
 6   糖尿病譜系功能  768 non-null    float64
 7   年齡        768 non-null    int64
 8   是否有糖尿病  768 non-null    object
dtypes: float64(7), int64(1), object(1)
memory usage: 54.1+ KB
```

填空後結果

填空後資料

	懷孕	葡萄糖	血壓	皮膚厚度	胰島素	BMI	糖尿病譜系功能	年齡	是否有糖尿病
0	6.000000	148.0	72.0	35.00000	155.548223	33.6	0.627	50	是
1	1.000000	85.0	66.0	29.00000	155.548223	26.6	0.351	31	否
2	8.000000	183.0	64.0	29.15342	155.548223	23.3	0.672	32	是
3	1.000000	89.0	66.0	23.00000	94.000000	28.1	0.167	21	否
4	4.494673	137.0	40.0	35.00000	168.000000	43.1	2.288	33	是
..	...	...	...	...	...	...	...	..	...
763	10.000000	101.0	76.0	48.00000	180.000000	32.9	0.171	63	否
764	2.000000	122.0	70.0	27.00000	155.548223	36.8	0.340	27	否
765	5.000000	121.0	72.0	23.00000	112.000000	26.2	0.245	30	否
766	1.000000	126.0	60.0	29.15342	155.548223	30.1	0.349	47	是
767	1.000000	93.0	70.0	31.00000	155.548223	30.4	0.315	23	否

[768 rows x 9 columns]

轉換後資訊

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   懷孕        768 non-null    float64
 1   葡萄糖      768 non-null    float64
 2   血壓        768 non-null    float64
 3   皮膚厚度    768 non-null    float64
 4   胰島素      768 non-null    float64
 5   BMI         768 non-null    float64
 6   糖尿病譜系功能  768 non-null    float64
 7   年齡        768 non-null    int64
 8   是否有糖尿病  768 non-null    object
dtypes: float64(7), int64(1), object(1)
memory usage: 54.1+ KB
```

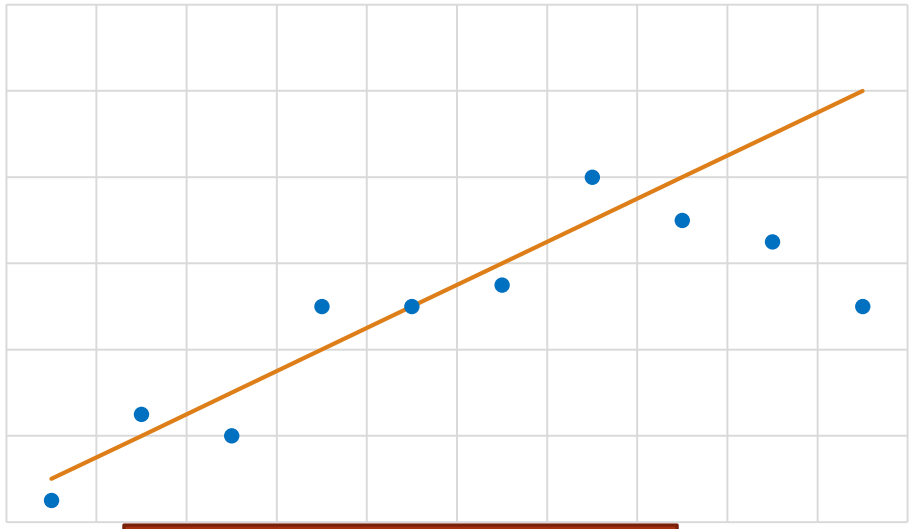
3. N倍交叉驗證

資料集的切割

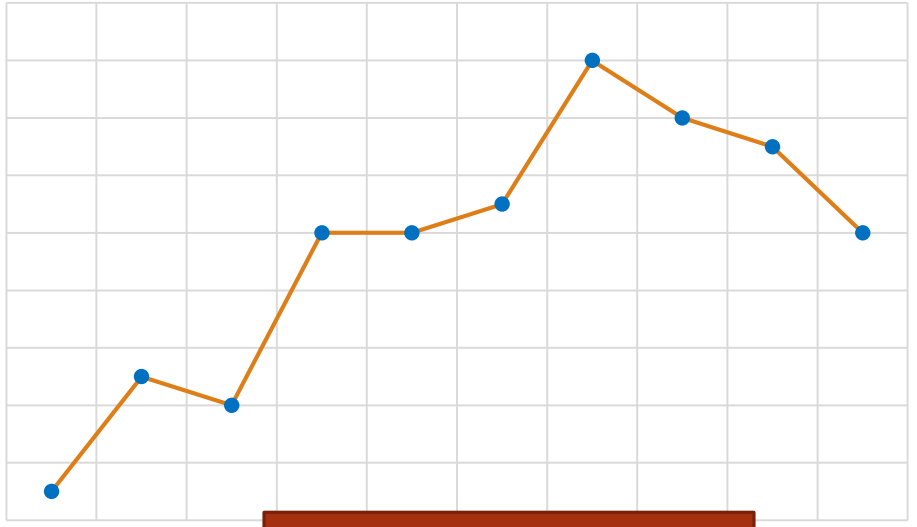
- 進行實驗的時候，會將資料分成以下三部分，防止過度擬合(Overfitting)
- 比率可以視情況調整

資料集	說明	比率
訓練集(Training)	讓機器學習模型學習的資料集	90%
驗證集(Validation)	模型在學習的過程中，驗證學習成果的資料集，並協助改善模型	
測試集(Testing)	模型學習完成後，驗證模型的效力的資料集	10%

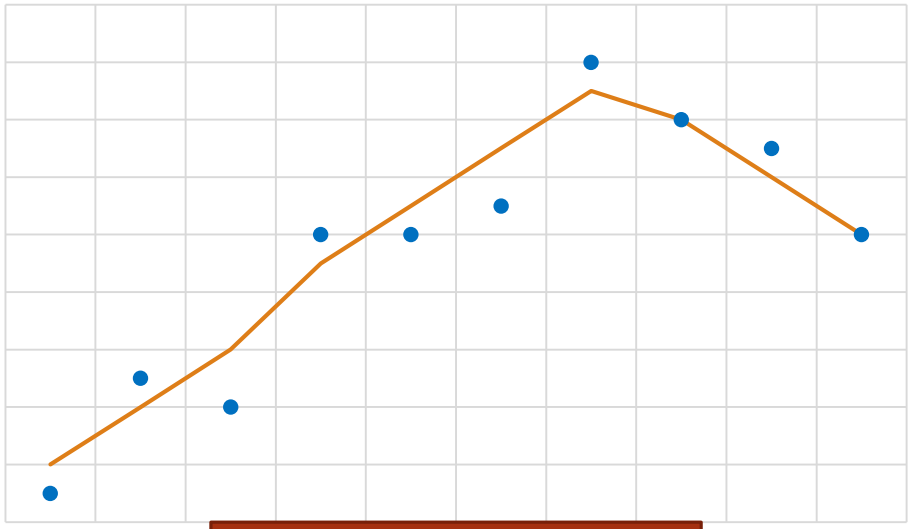
模型的擬合(Fitting)



欠擬合(Underfitting)



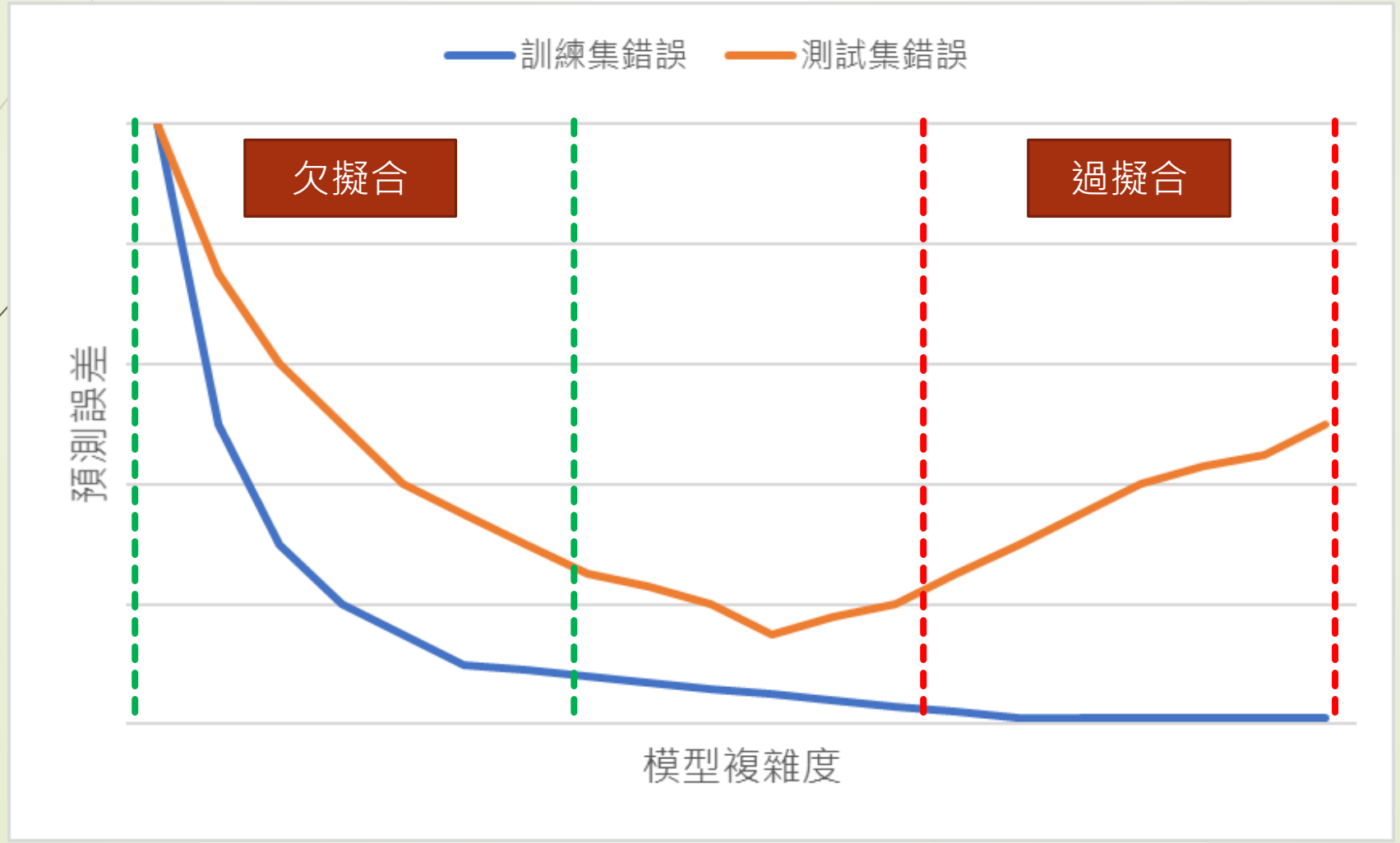
過擬合(Overfitting)



適當

模型的擬合(Fitting)

- 過擬合(Overfitting)或欠擬合(Underfitting)都會造成模型穩定性不足，遇到新的資料常常會判斷錯誤



交叉驗證(k-fold cross-validation)

- 何謂交叉驗證？
- 將資料集切割成N份，每一份分別與其他資料組成N個資料集，進行模型訓練，增強信效度
- 常用於資料間關聯性低的資料集，10次為最常用的次數

交叉驗證

- 以5倍交叉分析為例，將資料切割成5個驗證集
- 每一部分與其他未被選擇的部分形成一個子資料集，分別放入機器學習模型中訓練

資料集編號	1	2	3	4	5
	驗證集	訓練集	訓練集	訓練集	訓練集
	訓練集	驗證集	訓練集	訓練集	訓練集
	訓練集	訓練集	驗證集	訓練集	訓練集
	訓練集	訓練集	訓練集	驗證集	訓練集
	訓練集	訓練集	訓練集	訓練集	驗證集

交叉驗證

資料集	測試集準確率
1	90%
2	92%
3	86%
4	87%
5	91%
平均	89.2%

- 以準確率(accuracy)為指標，此實驗的準確率為89.2%

交叉驗證後，對每個資料及進行標準化處理

- 每個訓練集與測試集的特徵範圍不一，某些模型可能會因此偏重某個特徵
- 例如：某資料集的體溫範圍在35.5到38.5之間，體重在40到110之間，範圍不一，模型可能會偏重其中一方

標準化處理(standardized) – 標準化值(z-score)

- 將特徵的平均值轉換為0，標準差轉換為1，公式如下：

$$z = \frac{x - \mu}{\sigma}$$

- X 表示個別數值
- μ 表示平均數(Mean)
- σ 表示標準差(Standard)

標準化處理(standardized)
– 最大最小縮放(min-max normalization)

- $X_i = \frac{X_i - X_{min}}{X_{max} - X_{min}}$

- X_i代表該變數之特定值
- X_{min}代表該變數之最小值
- X_{max}代表該變數之最大值
- 將變數映射到(0,1)之範圍，消除變數之間範圍不一致的問題，以便演算法提取特徵，可以提升模型的精準度。

4.平衡資料數量

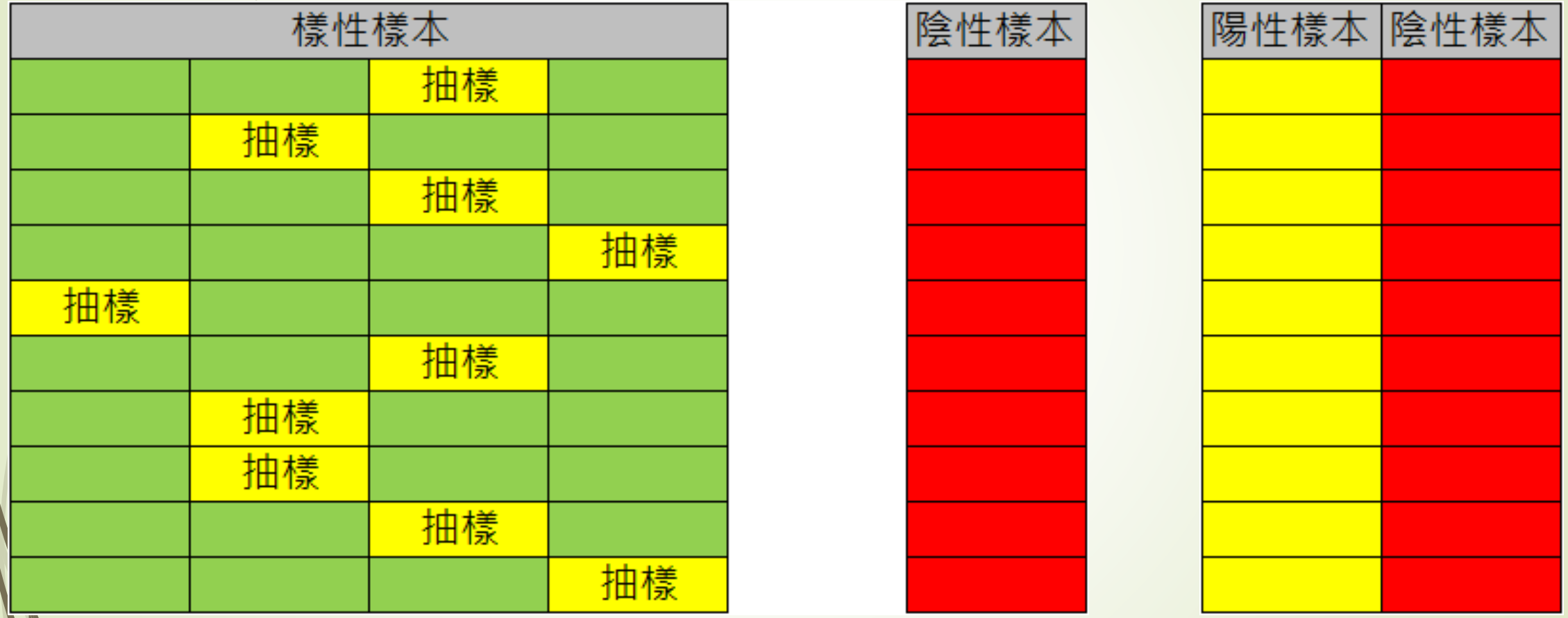
不平衡的資料

- 以二分類來說，有時資料集的標籤比例不是每次都是50：50，有時會到80：20的差異，甚至更高，產生不平衡資料 (imbalanced data，又稱資料傾斜)的問題
- 在此情形下，預測的結果大多會比較偏向於較多數的一方，故視情況要做資料平衡

資料平衡

- 欠取樣(Undersampling)
  以二分類而言，將較多一方的資料隨機取樣，使資料比率平衡
- 過取樣(Oversampling)
  以二分類而言，將較少一方的資料依據透過演算法，產生新的資料樣本，使資料比例平衡

欠取様(Undersampling)



5. 特徵工程

何謂特徵工程？

- 將資料轉換成能夠更好地表示「潛在問題」的特徵，進而提高機器學習的效能
- 挑選有利的特徵，壓縮以減少特徵

特徵工程的類型

- 統計學方法(選擇特徵)：
  假設檢定(Hypothesis test)
- 機器學習方法(轉換特徵)：
  線性判別分析(Linear Discriminant Analysis, LDA)

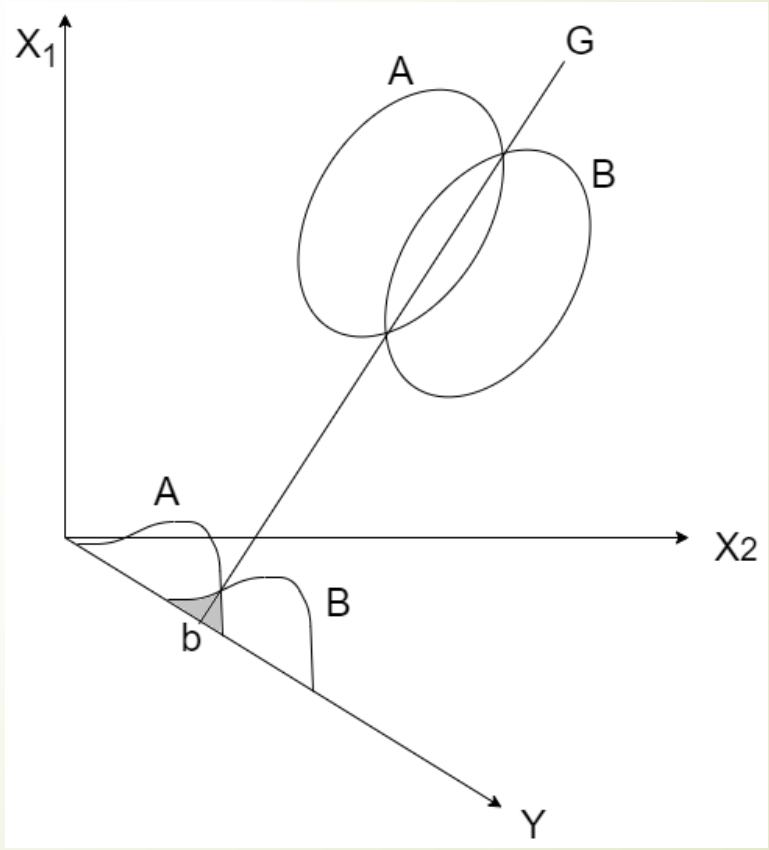
假設檢定(Hypothesis test)

- 虛無假設(H_0)：特徵與標籤(Y)沒有關係
- 對立假設(H_1) ：特徵與標籤(Y)有關係

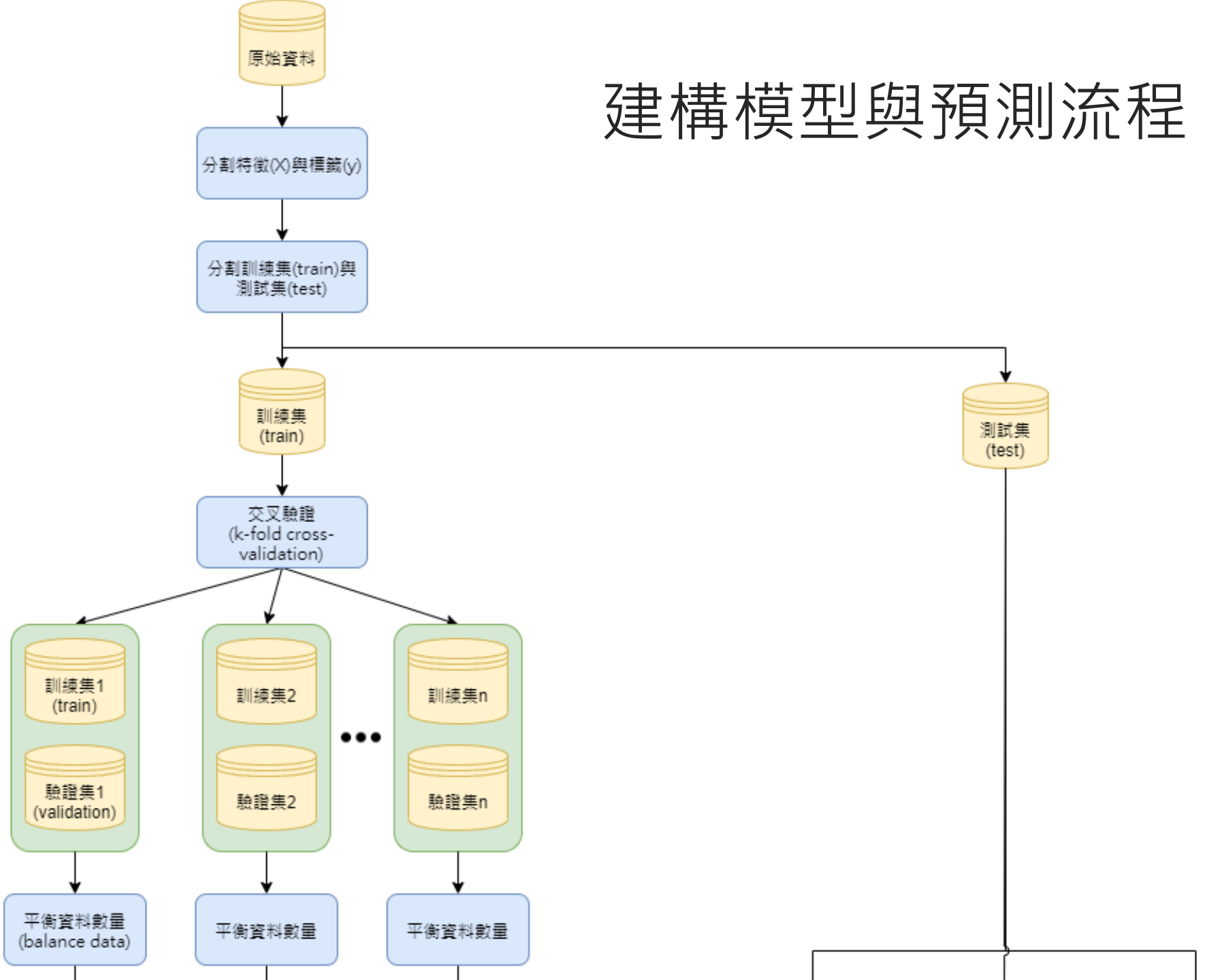
- p值是介於0到1的數字，表示在給定資料後，偶然出現H_0的機率，即p值越小，拒絕H_0的機會就越高，代表該特徵與標籤的關係越強烈，應該被保留

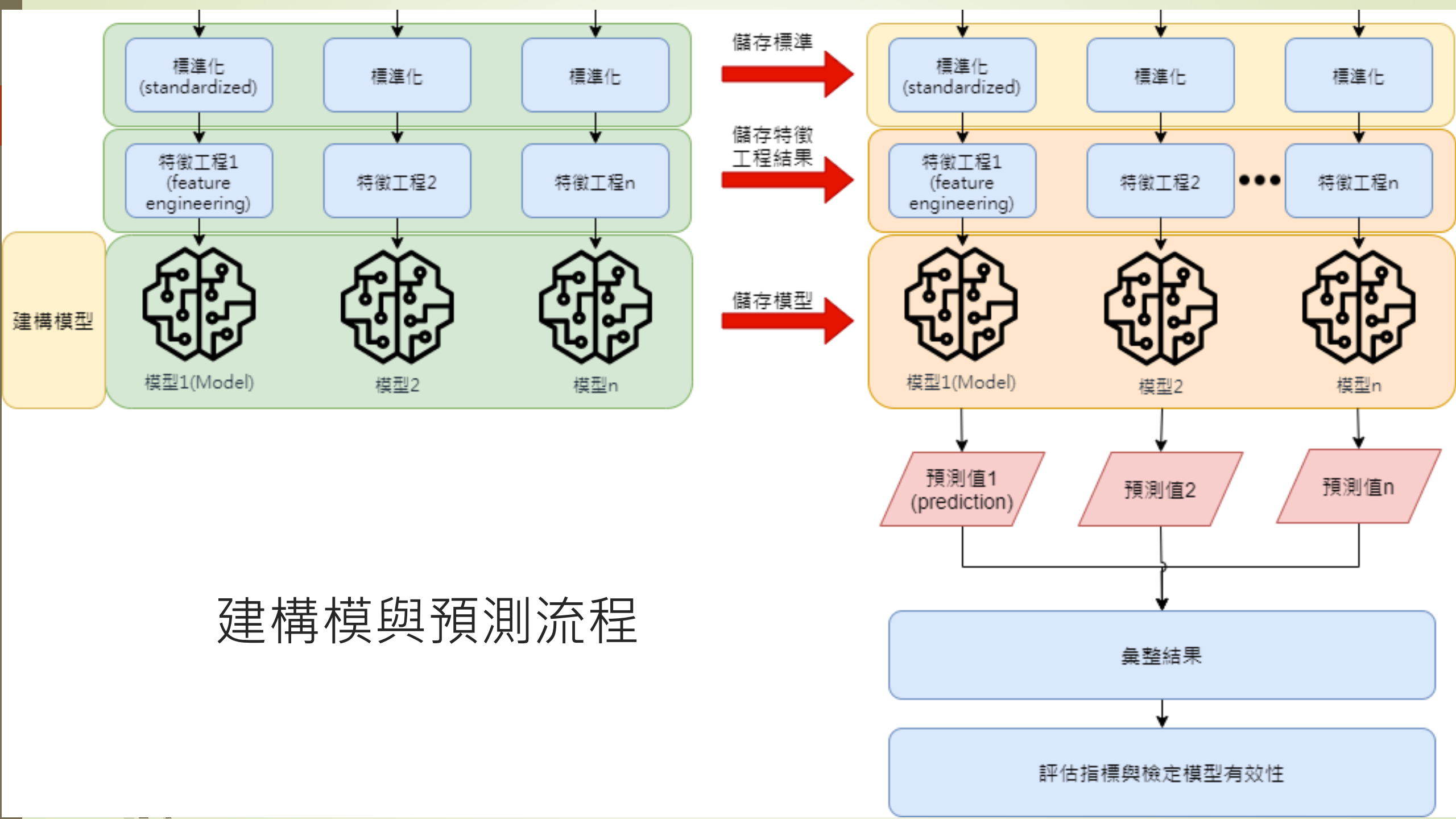
線性判別分析(Linear Discriminant Analysis, LDA)

- 為一個將資料降維的方法，設有A、B兩資料群體，將兩個群體所有資料點投影到一直線Y，該直線是與兩資料群體相交產生的直線G垂直



建構模型與預測流程


原始資料
分割特徵(X)與標籤(y)
分割訓練集(train)與測試集(test)
訓練集(train)
測試集(test)
交叉驗證(k-fold cross-validation)
訓練集1(train)
驗證集1(validation)
訓練集2
驗證集2
訓練集n
驗證集n
平衡資料數量(balance data)
平衡資料數量
平衡資料數量


標準化
(standardized)
標準化
標準化
特徵工程1
(feature
engineering)
特徵工程2
特徵工程n
建構模型
模型1(Model)
模型2
模型n
儲存標準
儲存特徵
工程結果
儲存模型
標準化
(standardized)
標準化
標準化
特徵工程1
(feature
engineering)
特徵工程2
特徵工程n
模型1(Model)
模型2
模型n
預測值1
(prediction)
預測值2
預測值n
彙整結果
評估指標與檢定模型有效性

建構模與預測流程

謝謝聆聽