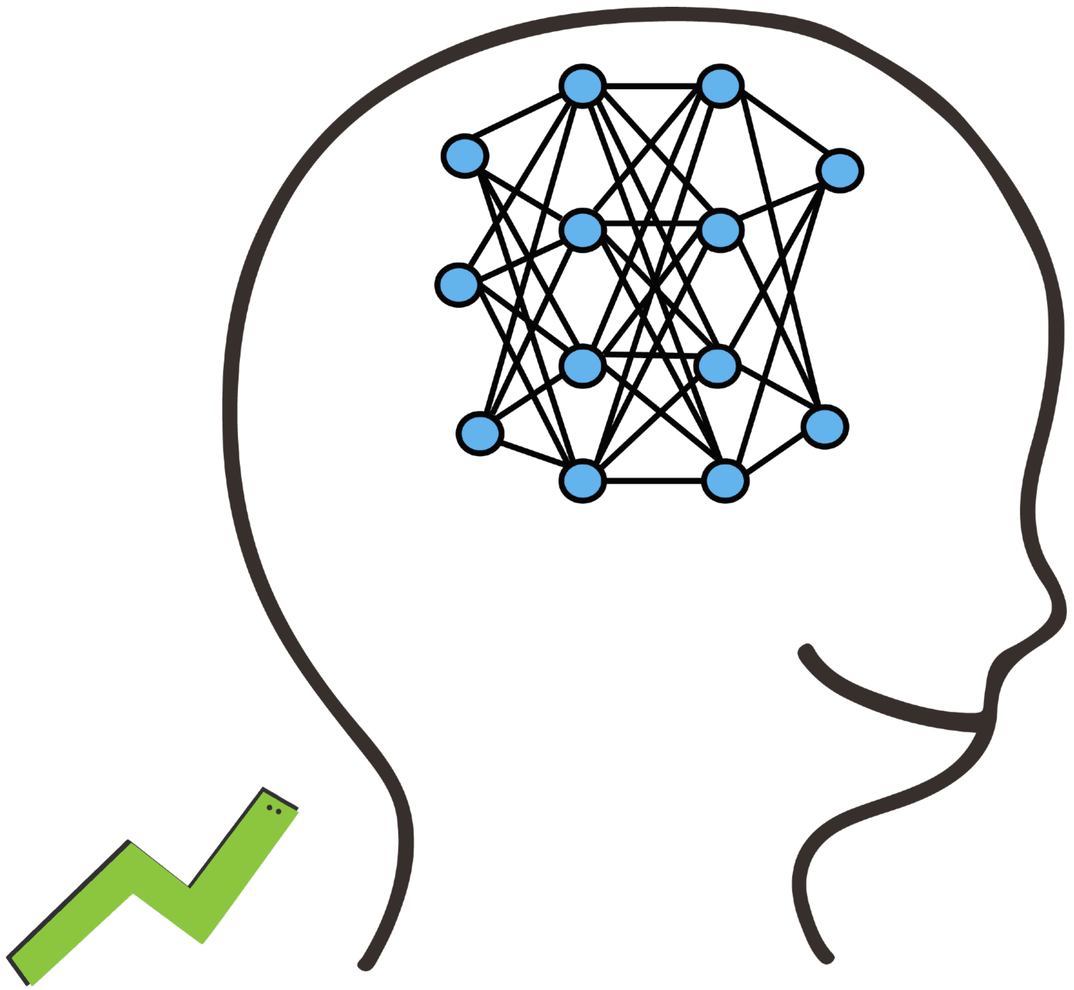
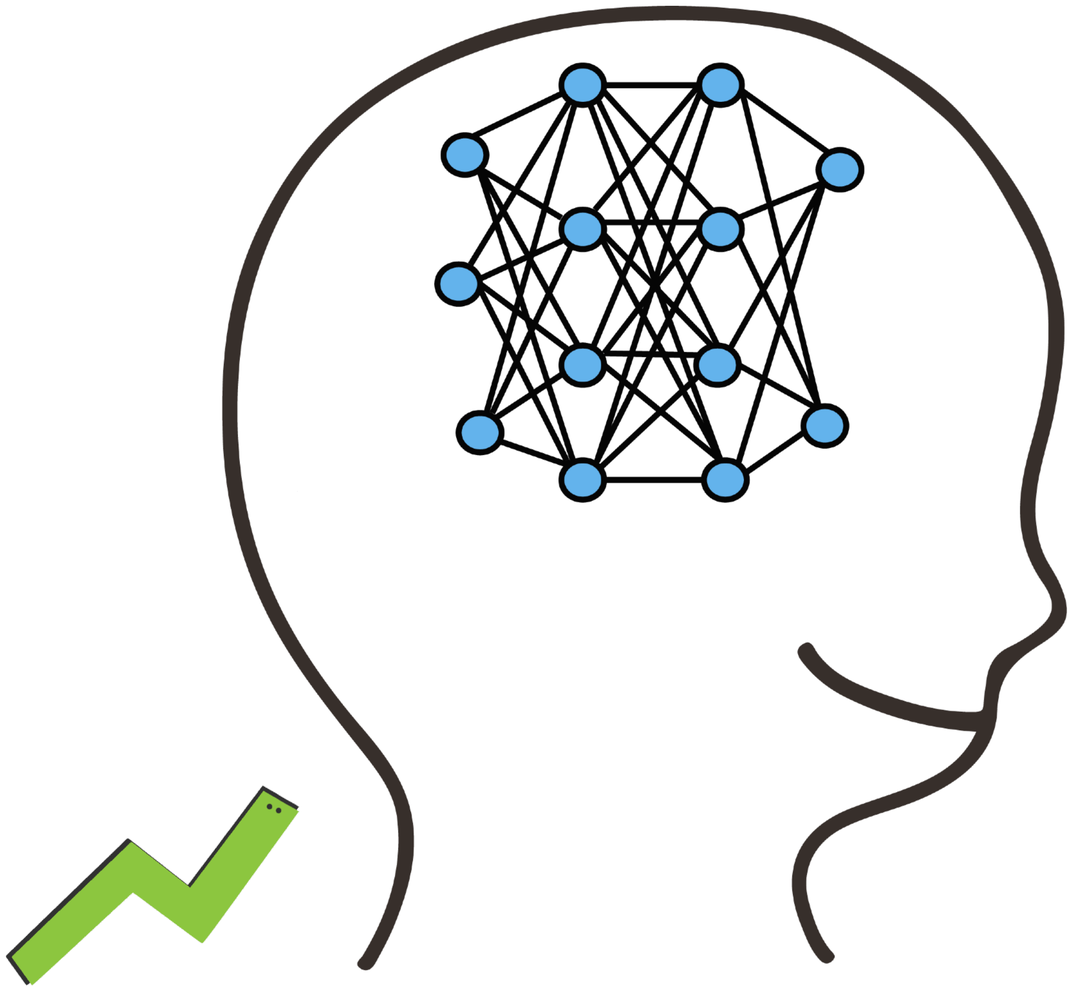


資料預處理

Data Preprocessing



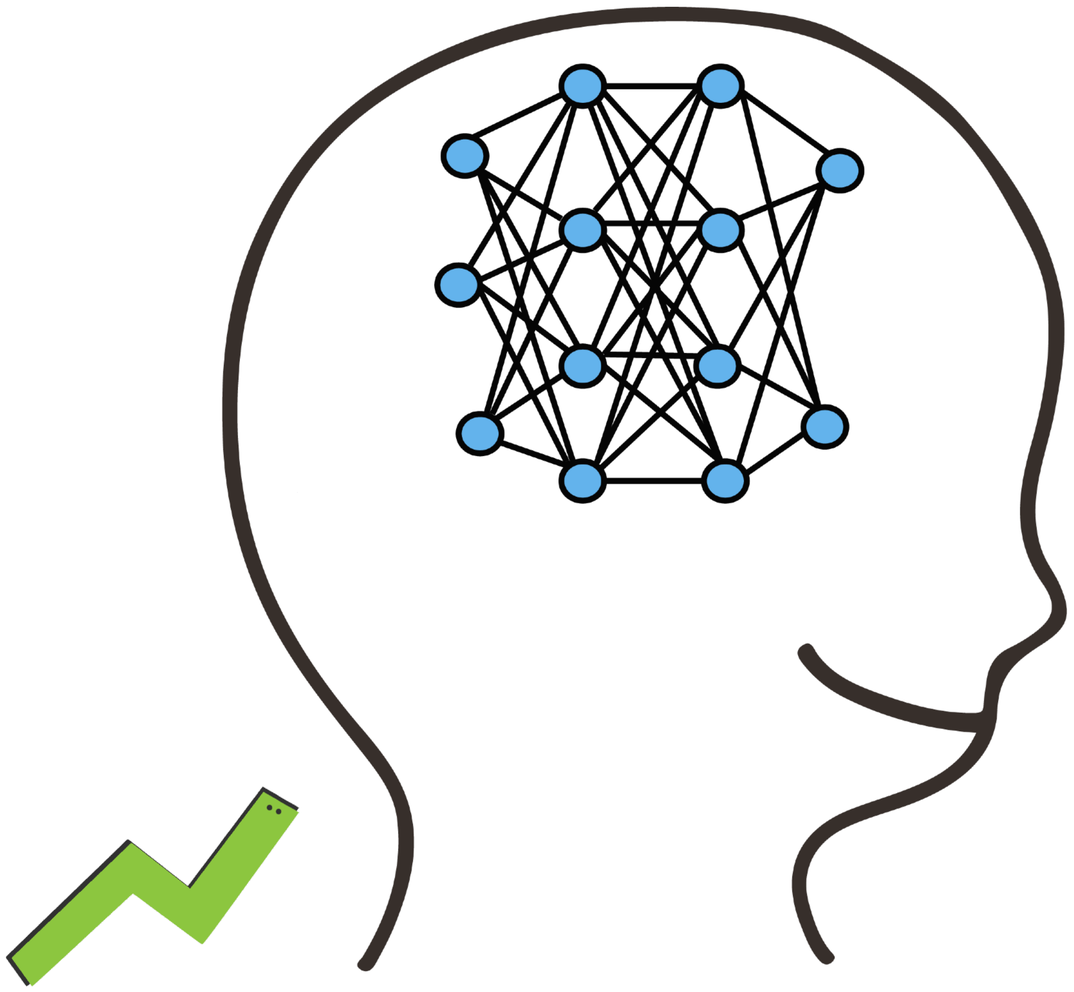
資料種類與測量量尺

Data Types & Scales of Measurement

測量量尺

- Scales of measurement（Stevens, 1946）

	次序	加減運算	乘除運算	自然零點	適用運算	Example
名義量尺 (Nominal Scale)					眾數、百分比	女:0, 男:1
順序量尺 (Ordinal Scale)	v				中位數、百分比	滿意度（1~5）
等距量尺 (Interval Scale)	v	v			可加減、不可乘除、平均、標準差	攝氏10、20度 (20度不是10度的兩倍)
比率量尺 (Ratio Scale)	v	v	v	v	加減乘除、平均、標準差	10kg、20kg (20kg是10kg的兩倍)



資料清理與轉換

Data Cleaning & Transformation

重複值（Duplicated Data）

- 搜尋重複值（列）：DataFrame.duplicated(*[columns]*)
  - 回傳每一列的True/False（第一次出現為False，第二次之後出現就是True）
- 移除重複值（列）：DataFrame.drop_duplicates(*[columns]*)
  - 預設只留下第一個（keep='last'可改成留下最後一個）

遺失值（Missing Data）

- 判斷是否為遺失值：Series.isnull()
  - 可判斷None、NaN（not a number）
  - e.g. df[df[‘item’].isnull()]
- 移除遺失值（留下非遺失值）：Series.notnull()
  - e.g. df = df[df[‘item’].notnull()]

刪除遺失值

- 也可以使用DataFrame.dropna()
- how : {‘any’, ‘all’}
    1. any(預設)：只要該列有NaN就移除該列
    2. all: 整列都是NaN才移除
    - e.g. df.dropna(how=‘all’)

轉換（Tranformation）

- 取代：DataFrame.replace() / Series.replace()
  - e.g. df.replace(float('NaN'), 0) # df裡的NaN取代為0
  - e.g. df.replace([float('NaN'), -1], 0) # df裡的NaN和-1都取代為0
  - e.g. df.replace({'女':0,'男':1}) # df裡的'女'取代為0、'男'取代為1
  - e.g. df['col2'].replace(float('NaN'), 0) # col2欄位裡的NaN取代為0

Notes

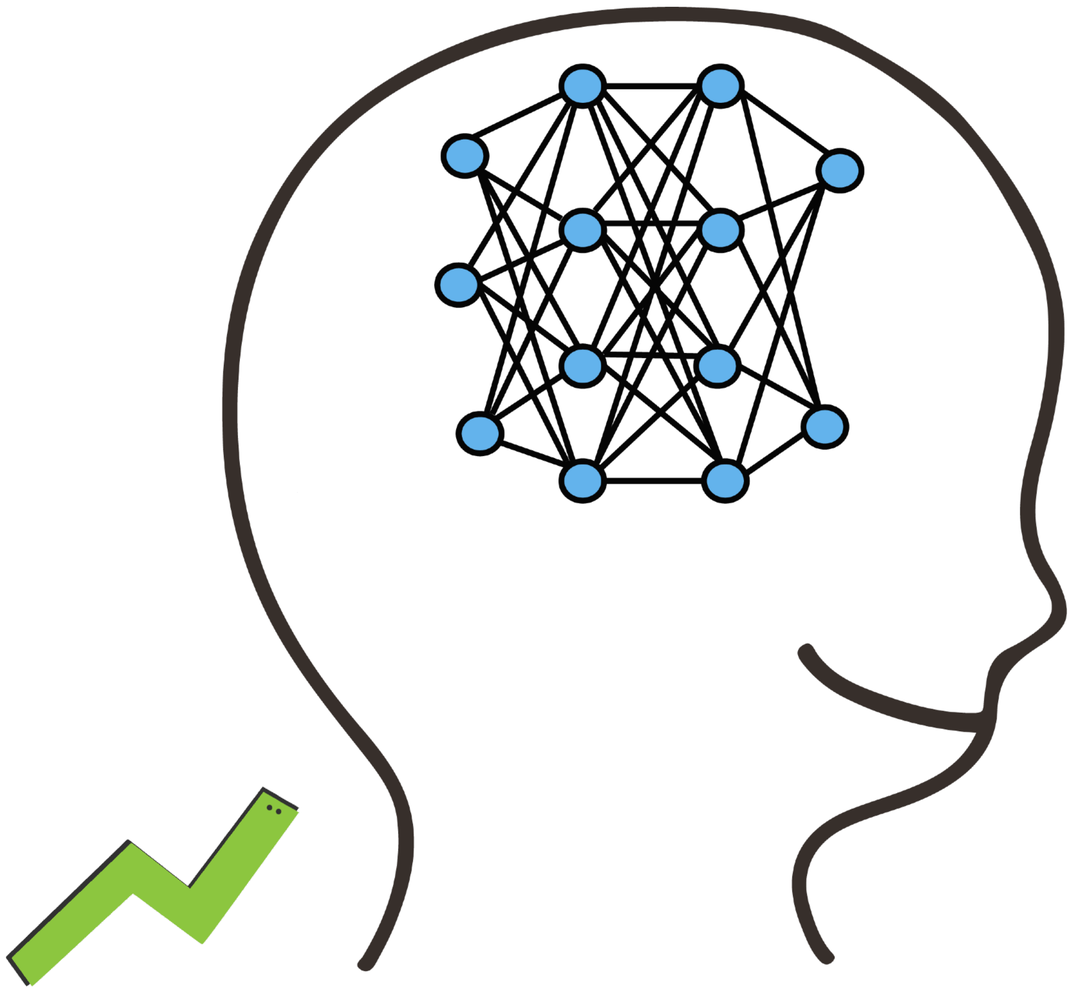
- **產生 NaN (Not a number)**
  - **float('NaN')**
  - **np.nan (先 import numpy as np)**

分箱後轉換

- 分箱（bins）：pd.cut(*Series, bins*)
  - e.g. bins = [0, 60, 70, 80, 90, 100]
  - labels = ['F','D','C','B','A']
  - df[‘label'] = pd.cut(df['score'],bins, right=False, labels=labels)
  - Categories (5, object): [[0, 60) < [60, 70) < [70, 80) < [80, 90) < [90, 100)]

	id	score
0	s01	74
1	s02	59
2	s03	98
3	s04	84
4	s05	60

	id	score	label
0	s01	74	C
1	s02	59	F
2	s03	98	A
3	s04	84	B
4	s05	60	D



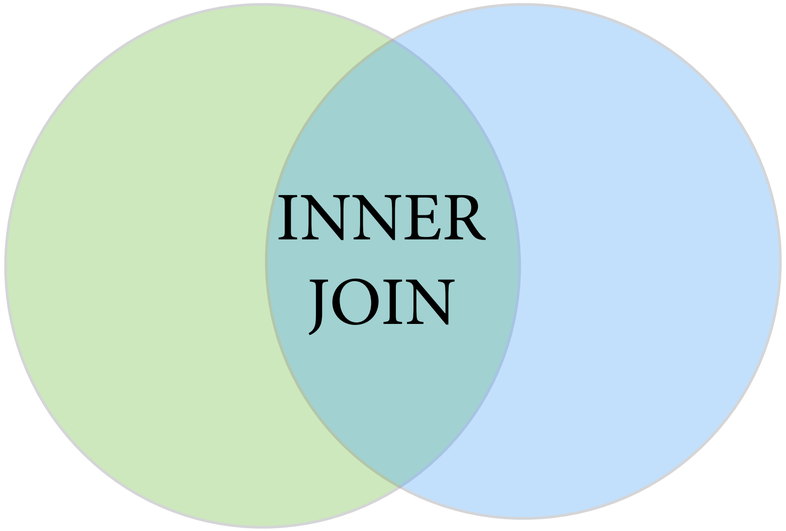
DataFrame的四種JOIN

4 types of JOIN with DataFrame

INNER JOIN

- 取交集：pd.merge(*df1*,*df2*, *left_on='cid'*, *right_on='cid'*, *how='inner'*)

	cid	item
0	c05	i01
1	c12	i02
2	c02	i03
3	c02	i04
4	c03	i05



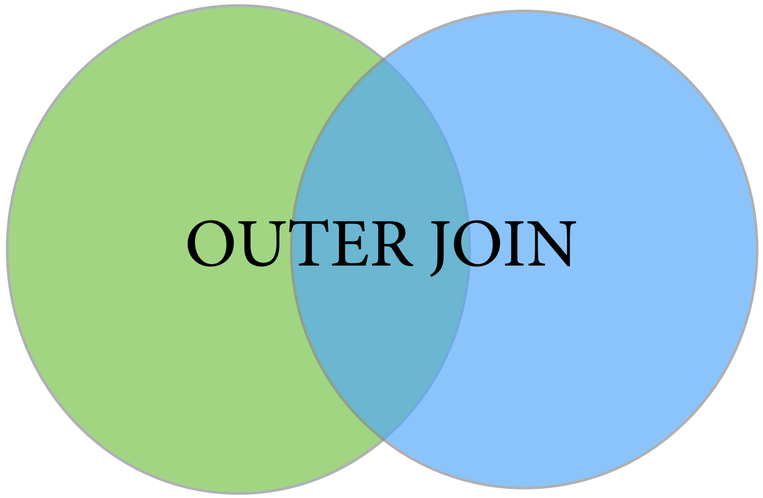
	cid	num_products
0	c01	2
1	c02	3
2	c03	12
3	c04	5
4	c05	8

	cid	item	num_products
0	c05	i01	8
1	c02	i03	3
2	c02	i04	3
3	c03	i05	12

OUTER JOIN

- 取聯集：pd.merge(*df1*,*df2*, *left_on='cid'*, *right_on='cid'*, *how='outer'*)

	cid	item
0	c05	i01
1	c12	i02
2	c02	i03
3	c02	i04
4	c03	i05



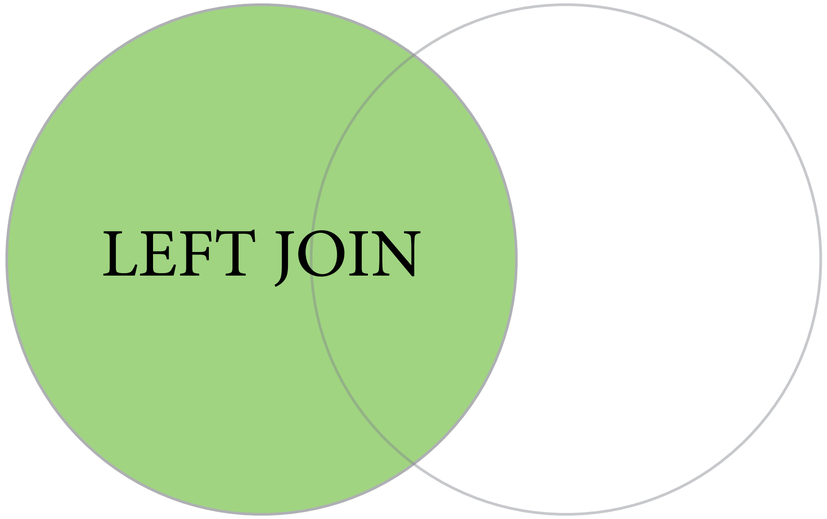
	cid	num_products
0	c01	2
1	c02	3
2	c03	12
3	c04	5
4	c05	8

	cid	item	num_products
0	c05	i01	8.0
1	c12	i02	NaN
2	c02	i03	3.0
3	c02	i04	3.0
4	c03	i05	12.0
5	c01	NaN	2.0
6	c04	NaN	5.0

LEFT JOIN

- 以左DataFrame欄位為主體：pd.merge(*df3*,*df4*, *left_on='cid'*, *right_on='cid'*, *how='left'*)

	cid	item
0	c05	i01
1	c12	i02
2	c02	i03
3	c02	i04
4	c03	i05



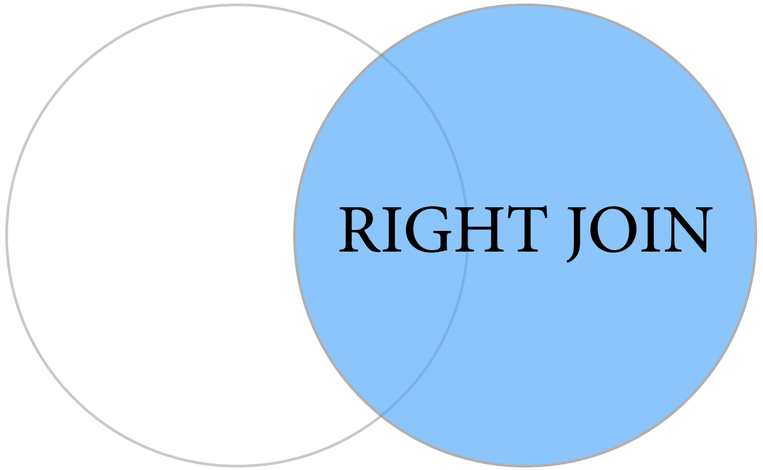
	cid	num_products
0	c01	2
1	c02	3
2	c03	12
3	c04	5
4	c05	8

	cid	item	num_products
0	c05	i01	8.0
1	c12	i02	NaN
2	c02	i03	3.0
3	c02	i04	3.0
4	c03	i05	12.0

RIGHT JOIN

- 以右DataFrame欄位為主體：pd.merge(*df3,df4, left_on='cid', right_on='cid', how='right'*)

	cid	item
0	c05	i01
1	c12	i02
2	c02	i03
3	c02	i04
4	c03	i05



	cid	num_products
0	c01	2
1	c02	3
2	c03	12
3	c04	5
4	c05	8

	cid	item	num_products
0	c05	i01	8
1	c02	i03	3
2	c02	i04	3
3	c03	i05	12
4	c01	NaN	2
5	c04	NaN	5