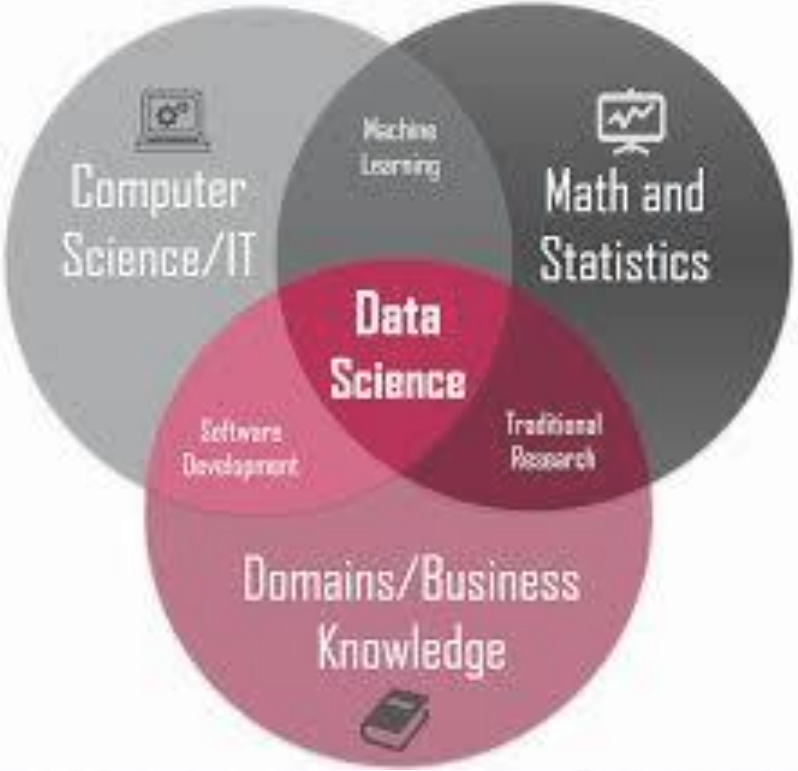
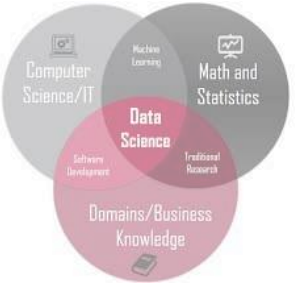
AI & DS



U06-資料預先處理2
Scikit-Learn

2023.10_V1.2

單元大綱



※使用Scikit-Learn 套件

- Scikit-Learn主要功能介紹
- 數值標準化
- 非數值資料準換

※特徵選擇

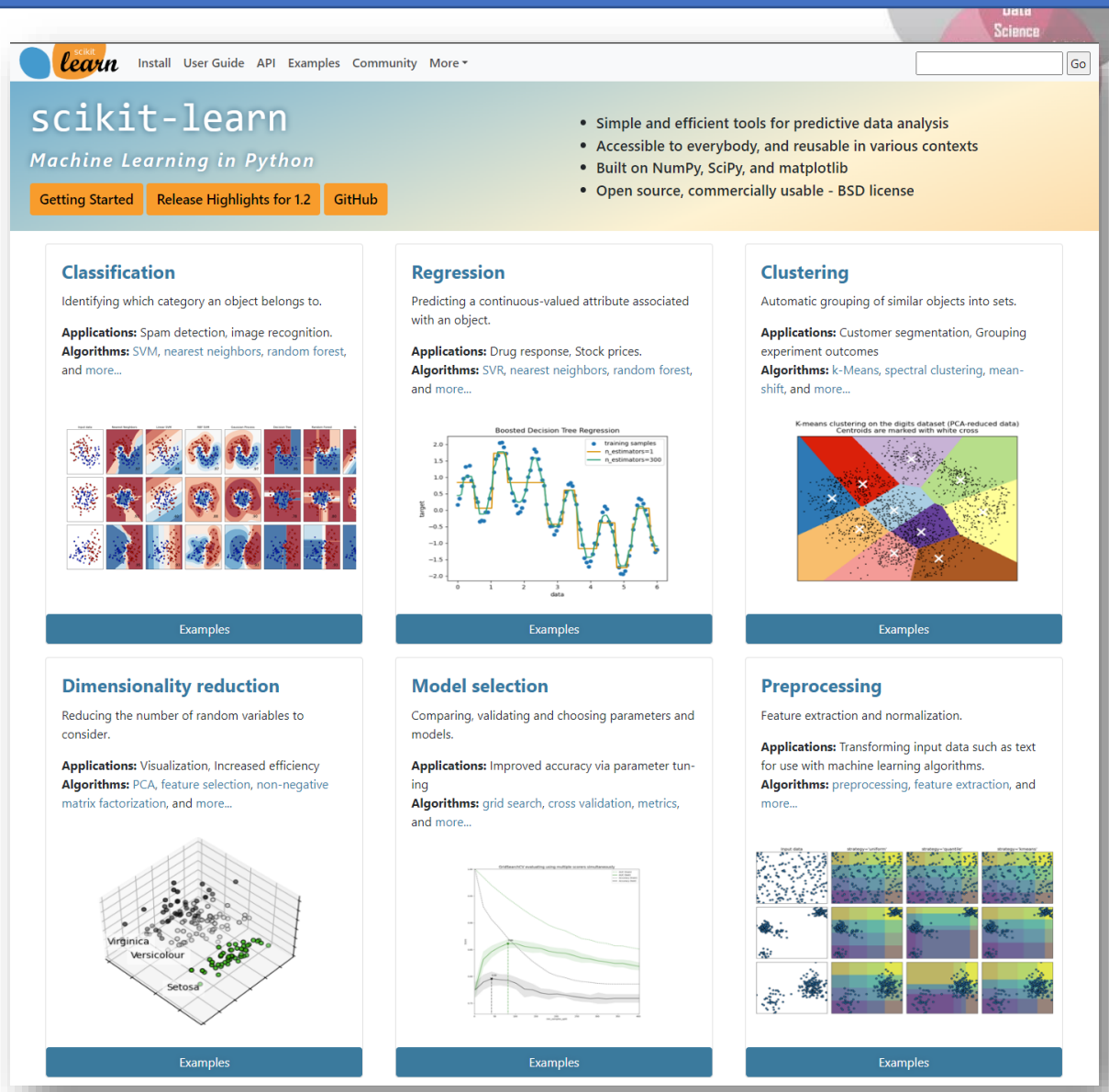
- 使用Pandas進行特徵選擇
- 使用Scikit-Learn進行特徵選擇

Part 1

關於Scikit-Learn

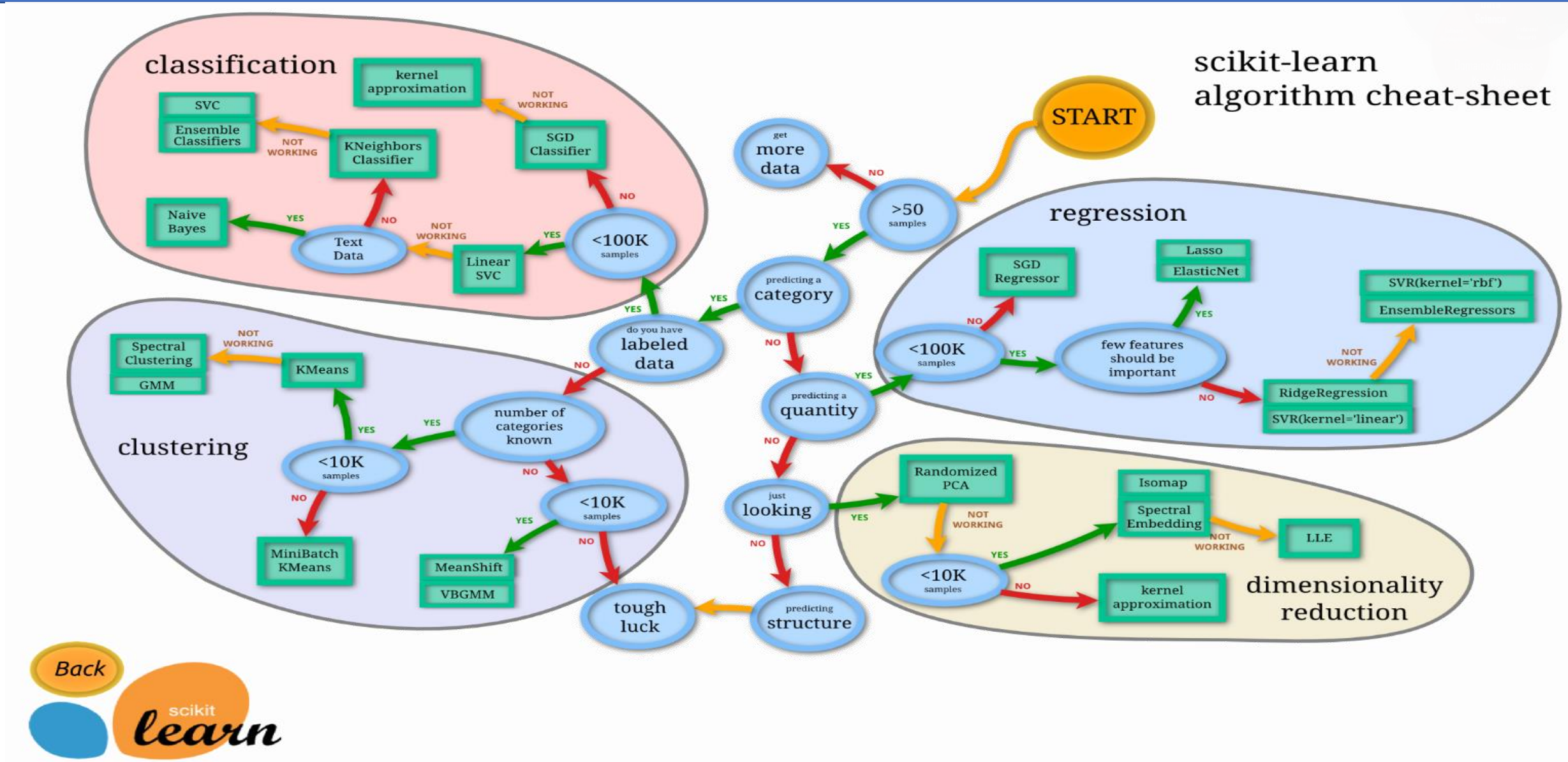
Scikit-Learn機器學習的開發利器

- Scikit-learn 計劃開始於scikits.learn，它是David Cournapeau的Google編程之夏計劃。它的名字來源自成為「SciKit」（SciPy工具箱）的想法，即一個獨立開發和發行的第三方SciPy擴展。
- 在2010年，來自法國羅康庫爾的法國國家信息與自動化研究所的Fabian Pedregosa、Gael Varoquaux、Alexandre Gramfort和Vincent Michel，領導了這個項目並在2010年2月1日進行了首次公開發行。
- Scikit-learn主要用Python書寫的，並廣泛使用NumPy進行高性能線性代數和數組運算。此外，一些核心算法用Cython書寫來以提高性能。
- Scikit-learn 是基於 NumPy、SciPy 和 matplotlib 所構建。
- 官網: https://scikit-learn.org/stable/index.html



[Ref] 維基百科: https://zh.wikipedia.org/zh-tw/Scikit-learn

Scikit-Learn 演算法地圖



Back

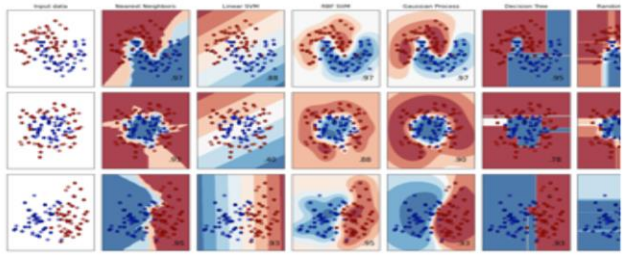
[Ref]: https://scikit-learn.org/stable/index.html

Scikit-Learn 主要功能

分類

識別對象屬於哪個類別。

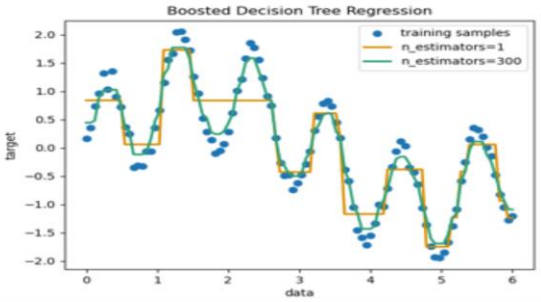
應用： 垃圾郵件檢測、圖像識別。
算法： 支持向量機、 最近鄰、 隨機森林等...



回歸

預測與對象關聯的連續值屬性。

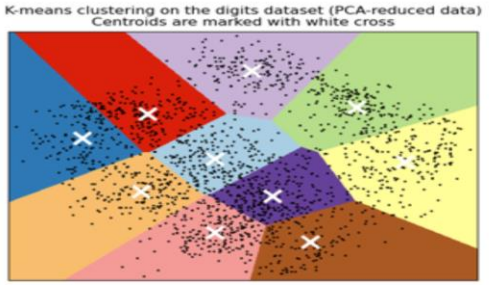
應用： 藥物反應、股票價格。
算法： SVR、 最近鄰、 隨機森林等...



聚類

將相似的對象自動分組到集合中。

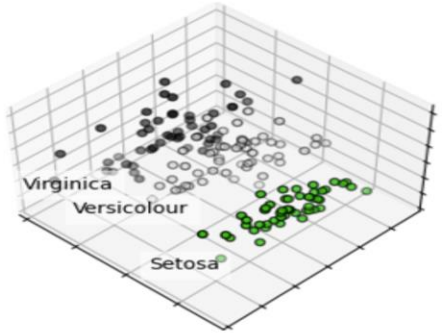
應用： 客戶細分、分組實驗結果
算法： k-Means、 譜聚類、 均值偏移等...



降維

減少要考慮的隨機變量的數量。

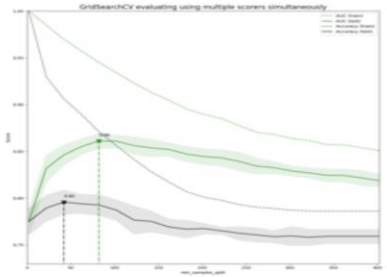
應用： 可視化、提高效率
算法： PCA、 特徵選擇、 非負矩陣分解等...



選型

比較、驗證和選擇參數和模型。

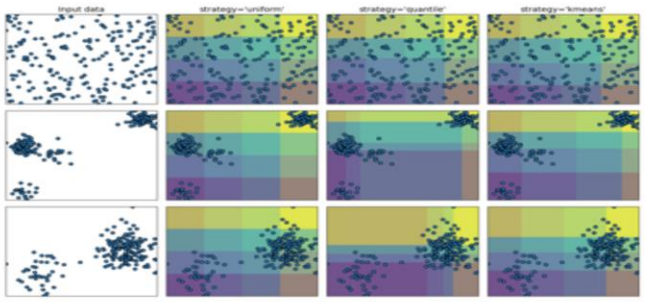
應用： 通過參數調整
算法提高準確性： 網格搜索、 交叉驗證、 指標等...



預處理

特徵提取和歸一化。

應用： 轉換輸入數據，例如用於機器學習算法的文本。
算法： 預處理、 特徵提取等...



Part 2 資料正規化與標準化

再複習一下機器學習的流程

在資料前置處理步驟中，需要完成:

- 資料清理（包含清理缺失值、異常值、離群值等）
- 資料正規化
- 資料標準化

- [Ref] https://ithelp.ithome.com.tw/articles/10216967?sc=rss.iron

為何需要做正規化(Normalization)與標準化(Standardization)

- 資料中不同變數往往有不同的單位、不同的數值範圍，例如：加速度計所記錄的加速度值單位為m/s^2，若資料內有光照計的紀錄資料，則單位為0~100不等。若將原始資料直接放入模型中，可能會影響分析的結果。
- 在需要計算變數之間的距離(distance)之模型中就會產生影響，如：KNN。距離的數值就會和變數的尺度(scale)非常相關，scale越大對於distance的計算就會有越大的影響。
- 但在tree-based的模型中則不會受到影響，如：決策樹(decision tree)、隨機森林(random forest)等。
- 在進行模型預測時，往往會使用多個不同的模型來進行比較，因此使用校正後的資料放入模型中是常見的作法，以避免出現不同模型放入不同資料的問題。
- 為了消除不同單位可能會帶來的影響，因此需要將數據進行**正規化(Normalization)**與標準化(Standardization)處理，使得不同變項之間具有可比性。
- 若要進行模型訓練，則需要分別對訓練資料集以及測試資料集個別進行正規化和標準化
- **正規化**為將原始資料的數據**按比例縮放至[0,1]的區間**中，且**不改變原本的分佈情形**。
- **標準化**則會使資料的**平均值為0**，**標準差為1**，經過標準化之後，資料會較**符合常態分佈**，並可以減小**離群值**對於模型的影響。

[Ref] https://ithelp.ithome.com.tw/articles/10293893

正規化(Normalization)

- 在數據的世界，資料會以多種不同的方式呈現。其中屬於「名目尺度」的資料，常以數字或文字符號，進行替換。例1：性別，F:女性，M:男性。例2:UCI Dataset的Mushroom資料集中，在紀錄氣味的欄位(odor)，出現了a, l, c, y, f, m, n, p, s，分別紀錄不同類型的氣味。
- 在機器的世界中，無法了解到這些英文字母(類別的資料)對應之間的關係，只能將這些資料轉換為數值，讓機器識別。

Out[1]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	populatio
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	
5	e	x	y	y	t	a	f	c	b	n	...	s	w	w	p	w	o	p	k	
6	e	b	s	w	t	a	f	c	b	g	...	s	w	w	p	w	o	p	k	
7	e	b	y	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	
8	p	x	y	w	t	p	f	c	n	p	...	s	w	w	p	w	o	p	k	

原始資料

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	1	5	2	4	1	6	1	0	1	4	...	2	7	7	0	2	1	4	2	3	5
1	0	5	2	9	1	0	1	0	0	4	...	2	7	7	0	2	1	4	3	2	1
2	0	0	2	8	1	3	1	0	0	5	...	2	7	7	0	2	1	4	3	2	3
3	1	5	3	8	1	6	1	0	1	5	...	2	7	7	0	2	1	4	2	3	5
4	0	5	2	3	0	5	1	1	0	4	...	2	7	7	0	2	1	0	3	0	1
5	0	5	3	9	1	0	1	0	0	5	...	2	7	7	0	2	1	4	2	2	1
6	0	0	2	8	1	0	1	0	0	2	...	2	7	7	0	2	1	4	2	2	3
7	0	0	3	8	1	3	1	0	0	5	...	2	7	7	0	2	1	4	3	3	3
8	1	5	3	8	1	6	1	0	1	7	...	2	7	7	0	2	1	4	2	4	1

Label Encoding

[Ref] https://ithelp.ithome.com.tw/articles/10216967?sc=rss.iron

標準化(Standardization)

- 在數據資料中，是用不同資料欄位與資料值所組成，可能分佈狀況可能都不盡相同，因此，就必須將特徵資料按比例縮放，讓資料落在某一特定的區間。
- [例]如右表，學生A的數學成績比學生B高1分，學生B的英文成績比學生A的高1分，整體來說A的成績還是B的成績比較好？

學生	數學成績	英文成績
A	90	80
B	91	79
成績分佈	0 ~ 100	60~80

評比項目\公司	A	B
X	90	80
Y	70	80
Z	60	90

資料發散

評比項目\公司	A	B	N化後A	N化後B
X	90	80	0.41	0.32
Y	70	80	0.32	0.32
Z	60	90	0.27	0.36
SUM	220	250	1.00	1.00

資料收斂到[0,1]

[Ref] https://ithelp.ithome.com.tw/articles/10216967?sc=rss.iron

Part 3

數值資料標準化

Z-score

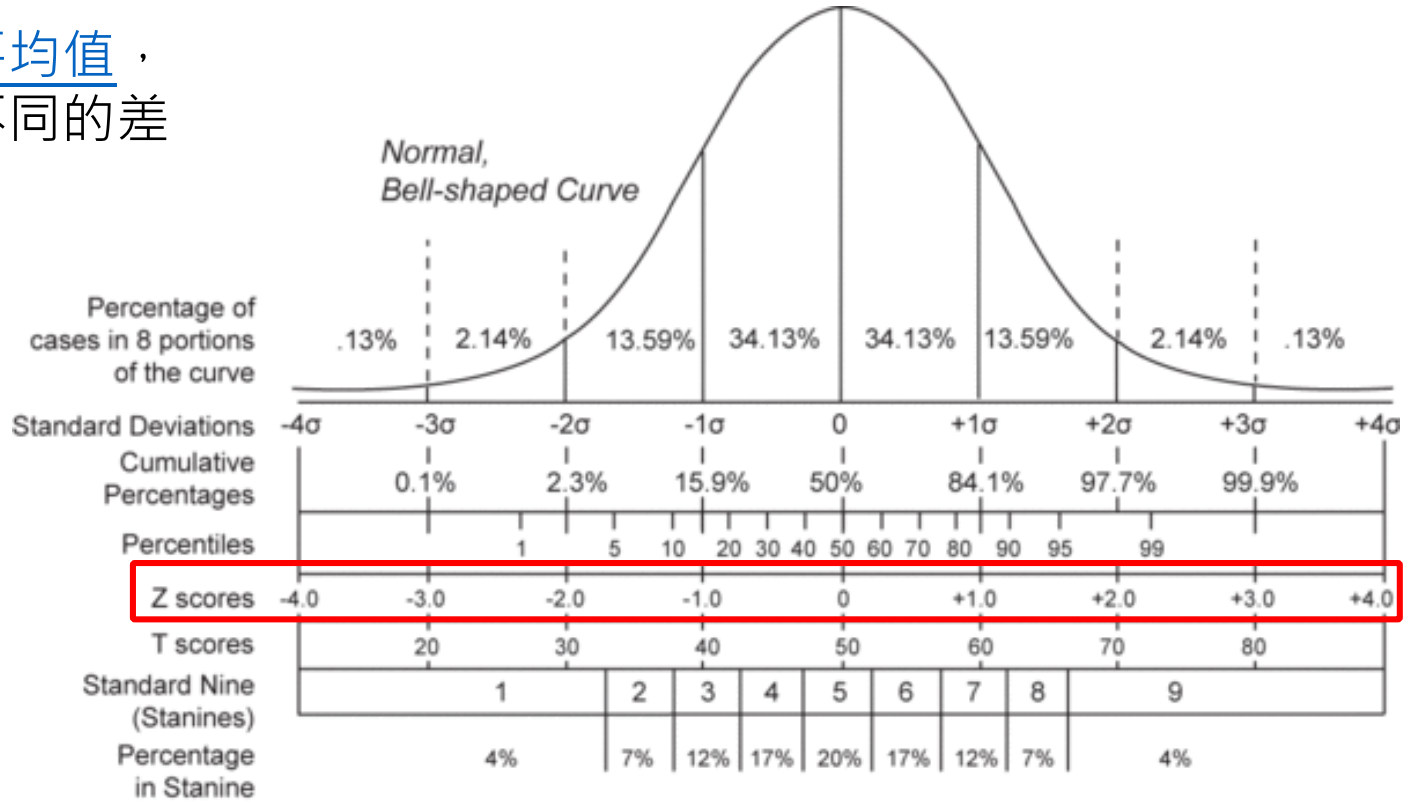
- **標準分數**（Standard Score，又稱**z-score**，中文稱為**Z-分數**或**標准化值**）在統計學中是一種無因次值，就是一種純數字標記。
- 藉由從單一（原始）分數中減去母體的平均值，再依照母體（母集合）的標準差分割成不同的差距，

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

其中 $\sigma \neq 0$。

其中

- x 是需要被標準化的原始分數
- μ 是母體的平均值
- σ 是母體的標準差



- Z值的量代表著原始分數和母體平均值之間的距離，是以標準差為單位計算。在原始分數低於平均值時Z則為負數，反之則為正數。

[Ref] https://pse.is/4uegvy

Scikit-Learn 的Z分數標準模組


```
! pip install sklearn
```

※ Colab預設已經安裝

※ 載入Scikit-Learn的Z分數標準化模組

```
from sklearn.preprocessing import StandardScaler
```

※ 建立StandardScaler物件

```
Z分數物件 = StandardScaler()
```

※ 利用StandardScaler物件的fit_transform()方法轉換

```
轉換物件 = Z分數物件. fit_transform(數值資料)
```

最大最小值標準化: MinMaxScaler

- 對原始數據進行線性變換，無論原始數據是正值還是負值，結果都會落到 [0,1] 區間。並且正負指標均可轉化為正向指標，作用方向一致。

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Scikit-Learn的最大最小值標準化模組

```
from sklearn.preprocessing import MinMaxScaler
```

```
std= MinMaxScaler()
```

最大絕對值標準化: MaxAbsScaler

- MaxAbs方法跟Max-Min用法類似，也是將數據落入一定區間，但該方法的數據區間為[-1,1]。MaxAbs也具有不破壞原有數據分佈結構的特點，因此也可以用於稀疏數據。

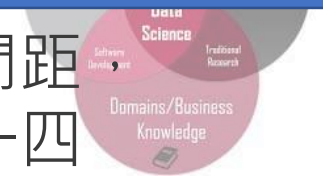
$$x' = \frac{x}{x_{max}}$$

Scikit-Learn的最大絕對值標準化模組

```
from sklearn.preprocessing import MaxAbsScaler
```

```
std= MaxAbsScaler()
```

穩健縮放標準化: RobustScaler



- Robust Scaling 是將變數值減掉中位數(median)再除以變數的四分位數間距，四分位數間距是第三四分位數(the 3rd quartile or 75th quantile)減第一四分位數(the 1st quartile or 25th quantile)。這個過程基本上和Min-Max Scaling相似，但它提供了一個較好的範圍給高度偏態分布的資料。縮放後，每個變數間的變異數、最大值、最小值不一樣，也可能改變原始資料分布的型態，不容易受到異常值影響。

$$\bar{x} = \frac{X - median(X)}{\text{75th quantile(x)} - \text{25th quantile(x)}}$$

Scikit-Learn的最大絕對值標準化模組

```
from sklearn.preprocessing import RobustScaler
```

```
std= RobustScaler()
```

[Ref] https://ithelp.ithome.com.tw/m/articles/10238774

數值資料標準化方法比較

比較項目 \ 方法	Z-score	MinMaxScaler	MaxAbsScaler	RobustScaler
優點	受異常值影響小	原理簡單,運算快	原理簡單,運算最快	受異常值影響最小
缺點	公式難以理解,運算時 耗費電腦資源	容易受異常值影響	容易受絕對值很大數值的影響	原理不容易被理解
適用範圍	常態分配	沒有異常的資料	沒有異常且需要保留正負值	包含較多的異常值

Part 4 非數值資料轉換

對應字典法

- 要轉換的特徵值不多,可用此方法直接將文字替換為數值。
- 對應字典法是將要替換的文字與對應的數值建立成字典, 再以 DataFrame 的: df.replace()與df.map()方法,進行替換。

資料取代: df.replace()

- 先以 df.unique()方法查出指定欄位的所有特徵值。例如:
  `df['婚姻'].unique()` ➔ `array(['單身', '已婚', '離婚', '未知'], dtype=object)`
- 再建立特徵值與對應數值的字典

字典變數 ={ '特徵值1' :數值1, '特徵值2' :數值2, ... }

- 再以 df.replace() 轉換為數值資料

DataFrame欄位.replace(字典變數, inplace = True)

inplace = True :表示索取代後的結果將直接更新原始的DataFrame中的資料

- 若將數值資料還原為對應的文字,只需要將字典改為數值對應文字型態即可還原:

字典變數 ={數值1 : '特徵值1' ,數值2: '特徵值2' , ... }

資料對應: df.map()

DataFrame欄位=DataFrame欄位.map(字典變數)

標籤編碼法(1/2)

- 若轉換的特徵值數量較多,建立字典會耗費時間。
- 標籤編碼法會自動偵測所有特徵值,將N個特徵值以0到N-1數值取代。

Scikit-Learn的 LabelEncoder模組

- 載入模組:

```
from sklearn.preprocessing import LabelEncoder
```

- 建立LabelEncoder物件:

```
標籤物件 = LabelEncoder()
```

- 以標籤物件的 fit_transform() 進行轉換:

```
DataFrame欄位 = 標籤物件. fit_transform(DataFrame欄位)
```

查詢轉換後數值代表特徵值

- 標籤編碼法會將特徵值轉換成數值, 可使用LabelEncoder()物件的 classes_查詢特徵值:

```
標籤.classes_
```

標籤編碼法(2/2)

將轉換後數值還原為特徵值

- LabelEncoder()物件的 inverse_transform()方法,可將數值還原成文字特徵值:

```
DataFrame欄位 = 標籤物件. inverse_transform(DataFrame欄位)
```

批次轉換所有非數值特徵

- 利用迴圈法可一次對所有非數值特徵進行數值轉換:

```
欄位變數=DataFrame變數._select_dtypes(exclude=[np.number]).column
```

One-Hot編碼(1/2)

- **One-Hot 編碼**在數位電路中被用來表示一種特殊的位元組合,在一個位元組裡,僅能容許單一位元是1,其他位元都必須是0。

訂購
一次
偶爾
從未
頻繁

轉換為數值資料

訂購
0
1
2
3

One-Hot 編碼

訂購
[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]

- 機器學習中的類別資料多採用 One-Hot編碼,最大優點是可消除數值大小的關係,讓每個元素都處於相同的地位。由於機器學習是基於數學的運算,有時會導致結果產生較大的誤差。透過本編碼方法後,每個元素值皆為1,即能得到正確的運算結果。
- 首先載入模組:

```
from sklearn.preprocessing import OneHotEncoder
```

- 建立OneHotEncoder物件:

```
Onehot物件 = OneHotEncoder(sparse = 布林值)
```

sparse: 若為True (預設)表示建立sparse格式,若為False,表示建立串列格式。

sparse格式雖節省記憶體,但可讀性較差,不易理解。

One-Hot編碼(2/2)

- 以 fit_transform()方法進行One-Hot轉換:

```
陣列變數=onehot物件. fit_transform(DataFrame 欄位)
```

- 如果在機器學習中使用One-Hot編碼,此串列資料傳給機器學習演算法即可。
- 若要在Pandas中觀察One-Hot編碼,則需使用 OneHotEncoder物件的 get_feature_names_out() 方法轉為DataFrame格式,語法如下:

```
DataFrame 變數 = pd.DataFrame(陣列變數,
                              column=onehot. get_feature_names_out(特徵名) )
```

Part 4

使用Pndas進行特徵選擇

認識特徵選擇

- **何謂特徵選擇**

在機器學習和統計學中，特徵選擇(feature selection)也被稱為**變量選擇**、**屬性選擇** 或**變量子集選擇** 。是指：為了構建模型而選擇相關特徵（即**屬性**、**指標**）子集的過程。使用特徵選擇技術有三個原因：

1.降低資料量
2.簡化模型，使之更易於被理解
3.縮短訓練時間
4.改善通用性、降低過擬合。

- **如何選擇重要的特徵**

常使用「相關係數」進行兩組資料之間的的相關性。相關係數很常用在機器學習或是統計分析上使用，主要衡量兩變數間「線性」關聯性的高低程度。常見的關係數通常有:

- 用於連續數據的分析➔「皮爾森相關係數(Pearson’s correlation coefficient)」
- 順序尺度的分析➔「斯皮爾曼等級相關係數(Spearman’s rank correlation coefficient)」

[Ref] https://pse.is/4v9rrm

- **Pandas提供特徵選擇的方法**

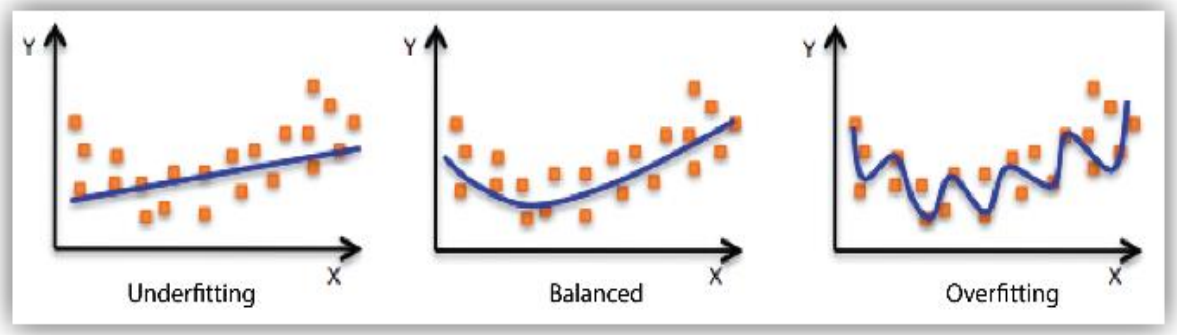
1.皮爾森 (Pearson)
2.肯德爾(Kendall)
3.斯皮爾曼 (Spearman)

以上三種方法都適用於迴歸分析(即目標值Y為數值的分析,例如:房價或股價預設)

- **[補充]低度擬合與過度擬合**

1.低度擬合: 當模型對訓練資料的執行效能不佳，這是因為模型無法擷取輸入範例 (通常稱為 X) 和目標值 (通常稱為 Y) 之間的關係。

2.過度擬合:當模型對訓練資料有很好的執行效果，但是對於評估資料無法執行得很好。



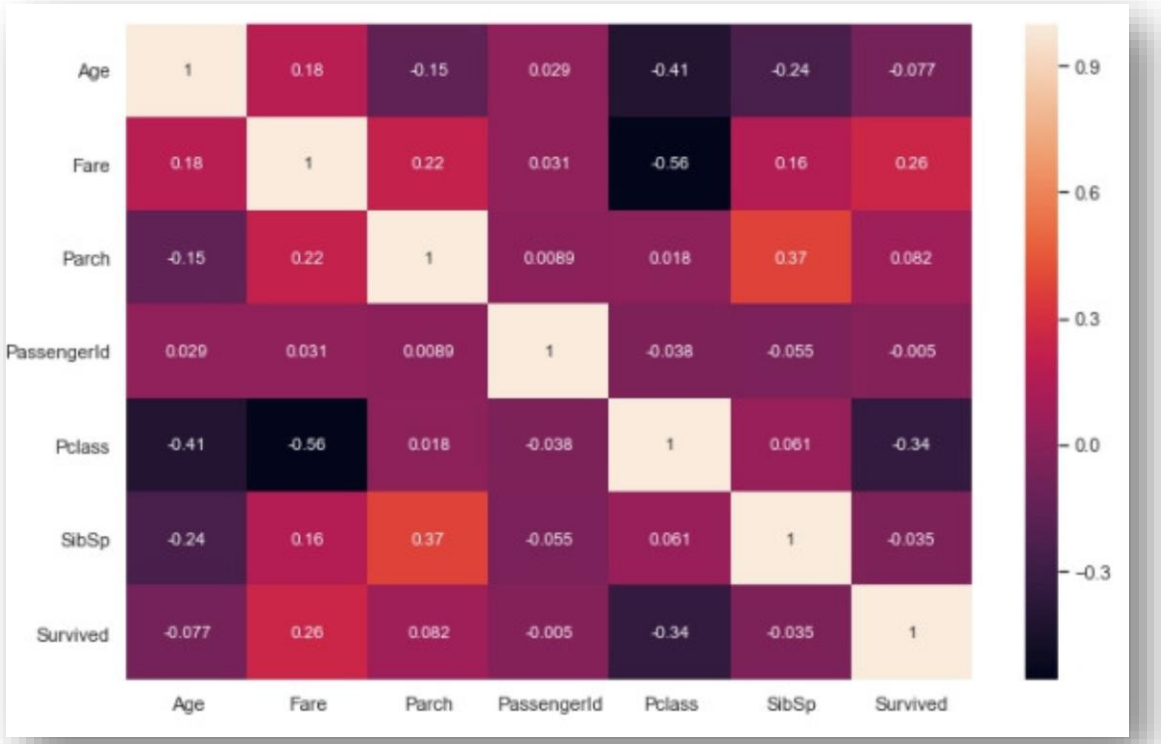
[Ref] https://pse.is/4v7rl9

使用 Pandas 計算相關係數

- Pandas 計算相關係數的語法:

相關變數 = DataFrame 變數.corr(method= ‘計算方式’ , min_periods=數值)

- **method**: 設定計算相關係數的方法:
  1.pearson: (預設)皮爾森相關係數
  2.kendall: 肯德爾相關係數
  3.spearman:斯皮爾曼相關係數
- **min_periods**: 設定最小資料數量, 預設值=1。



使用 Pearson相關係數

- **意義**: 兩個特徵的共變異數的標準差乘積之關係
- **值域**: -1與1之間。1: 完全正相關, 0: 無相關, -1:完全負相關
- **適用**: 線性分布且常態分佈的特徵值
- **語法**: **相關變數 = DataFrame 變數.corr(method= ‘pearson’ , min_periods=數值)**
  - 傳回值是所有特徵相關係數形成的矩陣列表
  - 左上到右下對角線是各特徵與自己的相關係數=1。
  - 列表中最後一欄是**Y標記值**:**房價**特徵對其他特徵的相關係數。例如:

    1.房價對房間數比為: 0.696515(高度正相關)

    2.房價對低收入比為:-0.735179(高度負相關)
  - 選取**相關係數絕對值最大者**,列為主要特徵。若選取過多特徵,會太耗費時間,並導致錯誤結果。

[補充] 閾值、門檻值。英語中的同義詞是threshold，不完全等價於臨界值（critical value），常設為:0.4。

	犯罪率	豪宅比	公設比	臨公園	NO濃度	房間數	屋齡	賣場距離	捷運距離	繳稅率	師生比	低收入比	房價
犯罪率	1.000000	-0.198184	0.403882	-0.056159	0.418423	-0.219135	0.350374	-0.378109	0.624691	0.581744	0.286694	0.453140	-0.386681
豪宅比	-0.198184	1.000000	-0.530497	-0.041534	-0.512648	0.315653	-0.567482	0.659829	-0.309246	-0.314601	-0.383729	-0.409970	0.363700
公設比	0.403882	-0.530497	1.000000	0.062161	0.760634	-0.391208	0.640189	-0.706669	0.592272	0.719742	0.373900	0.598912	-0.478614
臨公園	-0.056159	-0.041534	0.062161	1.000000	0.091162	0.094316	0.084713	-0.098918	-0.006554	-0.035305	-0.124759	-0.055581	0.179651
NO濃度	0.418423	-0.512648	0.760634	0.091162	1.000000	-0.302315	0.729869	-0.768413	0.607865	0.665860	0.176829	0.586526	-0.422592
房間數	-0.219135	0.315653	-0.391208	0.094316	-0.302315	1.000000	-0.232807	0.206958	-0.212325	-0.293659	-0.354519	-0.616665	0.696515
屋齡	0.350374	-0.567482	0.640189	0.084713	0.729869	-0.232807	1.000000	-0.748993	0.454357	0.505248	0.247797	0.596512	-0.364754
賣場距離	-0.378109	0.659829	-0.706669	-0.098918	-0.768413	0.206958	-0.748993	1.000000	-0.492819	-0.535611	-0.223204	-0.493881	0.247956
捷運距離	0.624691	-0.309246	0.592272	-0.006554	0.607865	-0.212325	0.454357	-0.492819	1.000000	0.909937	0.461487	0.484412	-0.377440
繳稅率	0.581744	-0.314601	0.719742	-0.035305	0.665860	-0.293659	0.505248	-0.535611	0.909937	1.000000	0.457685	0.540277	-0.464384
師生比	0.286694	-0.383729	0.373900	-0.124759	0.176829	-0.354519	0.247797	-0.223204	0.461487	0.457685	1.000000	0.365302	-0.504142
低收入比	0.453140	-0.409970	0.598912	-0.055581	0.586526	-0.616665	0.596512	-0.493881	0.484412	0.540277	0.365302	1.000000	-0.735179
房價	-0.386681	0.363700	-0.478614	0.179651	-0.422592	0.696515	-0.364754	0.247956	-0.377440	-0.464384	-0.504142	-0.735179	1.000000

使用 Kendall相關係數

- **意義**: 按照特定特徵排序,其他特徵常為亂序。計算同序對和異序對的差異,以求係數。
- **值域**: -1與1之間。1: 完全正相關, 0: 無相關, -1:完全負相關
- **適用**: 非線性分布且資料數量較小的特徵值
- **語法**: **相關變數 = DataFrame 變數.corr(method= 'kendall' , min_periods=數值)**

  - 傳回值是所有特徵相關係數形成的矩陣列表
  - 左上到右下對角線是各特徵與自己的相關係數=1。
  - 列表中最後一欄是**Y標記值**:**房價**特徵對其他特徵的相關係數。例如:

    1.房價對房間數比為: 0.484126(高度正相關)

    2.房價對低收入比為:-0.66242(高度負相關)

	犯罪率	豪宅比	公設比	臨公園	NO濃度	房間數	屋齡	賣場距離	捷運距離	繳稅率	師生比	低收入比	房價
犯罪率	1.000000	-0.460379	0.520270	0.034488	0.603402	-0.215448	0.500570	-0.539368	0.563798	0.547322	0.312456	0.455192	-0.404431
豪宅比	-0.460379	1.000000	-0.531968	-0.038050	-0.506646	0.278587	-0.426987	0.474682	-0.233688	-0.291470	-0.356645	-0.383358	0.339110
公設比	0.520270	-0.531968	1.000000	0.075058	0.610078	-0.290649	0.486181	-0.563801	0.354393	0.484720	0.330475	0.462395	-0.415849
臨公園	0.034488	-0.038050	0.075058	1.000000	0.055708	0.050547	0.053710	-0.065191	0.023653	-0.037600	-0.117967	-0.042643	0.117118
NO濃度	0.603402	-0.506646	0.610078	0.055708	1.000000	-0.216459	0.589950	-0.683932	0.436235	0.454030	0.273242	0.449213	-0.393477
房間數	-0.215448	0.278587	-0.290649	0.050547	-0.216459	1.000000	-0.183353	0.180823	-0.081648	-0.192688	-0.220673	-0.469532	0.484126
屋齡	0.500570	-0.426987	0.486181	0.053710	0.589950	-0.183353	1.000000	-0.610570	0.310727	0.362669	0.245115	0.481129	-0.384108
賣場距離	-0.539368	0.474682	-0.563801	-0.065191	-0.683932	0.180823	-0.610570	1.000000	-0.362540	-0.384072	-0.218994	-0.407415	0.311898
捷運距離	0.563798	-0.233688	0.354393	0.023653	0.436235	-0.081648	0.310727	-0.362540	1.000000	0.558429	0.252236	0.290080	-0.248842
繳稅率	0.547322	-0.291470	0.484720	-0.037600	0.454030	-0.192688	0.362669	-0.384072	0.558429	1.000000	0.287771	0.384324	-0.413675
師生比	0.312456	-0.356645	0.330475	-0.117967	0.273242	-0.220673	0.245115	-0.218994	0.252236	0.287771	1.000000	0.324036	-0.394713
低收入比	0.455192	-0.383358	0.462395	-0.042643	0.449213	-0.469532	0.481129	-0.407415	0.290080	0.384324	0.324036	1.000000	-0.666242
房價	-0.404431	0.339110	-0.415849	0.117118	-0.393477	0.484126	-0.384108	0.311898	-0.248842	-0.413675	-0.394713	-0.666242	1.000000

使用 Spearman相關係數

- **意義**: 將兩個特徵值分別依大小排序後成對等級,再以各對等級差來求取係數。Spearman相關係數的計算速度比Kendall相關係數為快。
- **值域**: -1與1之間。1: 完全正相關, 0: 無相關, -1:完全負相關
- **適用**: 非線性分布且具有異常值的特徵值
- **語法**: **相關變數 = DataFrame 變數.corr(method= 'spearman', min_periods=數值)**

- 傳回值是所有特徵相關係數形成的矩陣列表
- 左上到右下對角線是各特徵與自己的相關係數=1。
- 列表中最後一欄是**Y標記值**:**房價**特徵對其他特徵的相關係數。例如:

1.房價對房間數比為: 0.635398(高度正相關)

2.房價對低收入比為:-0.850714(高度負相關)

	犯罪率	豪宅比	公設比	臨公園	NO濃度	房間數	屋齡	賣場距離	捷運距離	繳稅率	師生比	低收入比	房價
犯罪率	1.000000	-0.569671	0.734762	0.042197	0.821355	-0.314092	0.706687	-0.744287	0.727641	0.731273	0.463736	0.634460	-0.558502
豪宅比	-0.569671	1.000000	-0.638285	-0.040454	-0.629175	0.361196	-0.541037	0.610052	-0.277894	-0.372251	-0.442181	-0.485576	0.436183
公設比	0.734762	-0.638285	1.000000	0.088866	0.788662	-0.413749	0.676212	-0.755432	0.456539	0.665352	0.427196	0.634162	-0.574273
臨公園	0.042197	-0.040454	0.088866	1.000000	0.067601	0.061829	0.065462	-0.079724	0.026749	-0.044428	-0.138738	-0.052163	0.142943
NO濃度	0.821355	-0.629175	0.788662	0.067601	1.000000	-0.310809	0.794984	-0.880266	0.587780	0.650474	0.384365	0.632562	-0.559078
房間數	-0.314092	0.361196	-0.413749	0.061829	-0.310809	1.000000	-0.272016	0.264631	-0.114412	-0.274610	-0.309278	-0.642547	0.635398
屋齡	0.706687	-0.541037	0.676212	0.065462	0.794984	-0.272016	1.000000	-0.802176	0.423539	0.528556	0.347743	0.652083	-0.542918
賣場距離	-0.744287	0.610052	-0.755432	-0.079724	-0.880266	0.264631	-0.802176	1.000000	-0.496508	-0.576185	-0.316907	-0.561163	0.443469
捷運距離	0.727641	-0.277894	0.456539	0.026749	0.587780	-0.114412	0.423539	-0.496508	1.000000	0.705047	0.318965	0.396541	-0.347042
繳稅率	0.731273	-0.372251	0.665352	-0.044428	0.650474	-0.274610	0.528556	-0.576185	0.705047	1.000000	0.453059	0.533801	-0.560836
師生比	0.463736	-0.442181	0.427196	-0.138738	0.384365	-0.309278	0.347743	-0.316907	0.318965	0.453059	1.000000	0.459183	-0.550923
低收入比	0.634460	-0.485576	0.634162	-0.052163	0.632562	-0.642547	0.652083	-0.561163	0.396541	0.533801	0.459183	1.000000	-0.850714
房價	-0.558502	0.436183	-0.574273	0.142943	-0.559078	0.635398	-0.542918	0.443469	-0.347042	-0.560836	-0.550923	-0.850714	1.000000

Part 5

使用Scikit-Learn進行特徵選擇

認識特徵選擇(1/2)

- 特徵工程可以分為兩大部分:
  1.根據現有的資料特徵進行篩選，選出較有影響力的特徵進行訓練
  2.根據現有的資料特徵，去衍生出資料集中沒有的特徵來讓模型學習

一、**特徵選取(Feature Selection)**:

用特徵選取的目的與時機有以下幾點:

1. 用少量的變數/特徵來保有原有的重要資訊
2. 當變數/特徵個數(# of p)遠大於樣本數(# of n)
3. Avoiding Curse of Dimensionality (避免維度災難)

[Ref] https://pse.is/4u8cyh

認識特徵選擇(2/2)

二、過濾法(Filter)：

過濾法是列入一些篩選特徵的標準，檢測與目標變數相關的特徵，挑選出具變化性以及中高度相關的特徵，方法包含：

1.移除低變異數的特徵:

- **常數特徵(Constant Feature)**：一個特徵下的值完全一樣，沒有變化
- **半常數特徵(Quasi-Constant Feature)**：特徵裡大部分都是同一個數值
- **重複特徵(Duplicated Feature)**：資料集有兩個以上完全一樣的特徵

2.單變量特徵選取:

- **SelectKBest**：選取 K 個最好的特徵，k 為參數，代表選擇的特徵數
- **f_regression**：用於連續型目標變數

三、包裝法(Wrapper)：

是一種特徵選擇和演算法訓練同時進行的方法，根據某一種評量標準，每次選擇某些特徵或排除某些特徵，常用的方法為遞歸特徵消除(RFE)。

- **RFE**: 是根據問題為離散或連續，利用機器學習的模型進行挑選，為一貪婪優化演算法,目的在找尋最佳的特徵子集。
- **Stepwise Selection**:開始有全部k個變數，從k個變數中選取對 y (label)變異最沒顯著影響的變數刪除，直到剩餘的變數對解釋y (label)剩餘變異皆有顯著影響才停止。

[Ref] https://pse.is/4vkear

[案例說明] https://pse.is/4wn6ud

利用卡方檢定進行特徵值選取

- **關於卡方檢定(Chi-square Test)**

一、使用狀況：

- 卡方獨立性檢定適用於分析兩組類別變數的關聯性。
- 同一樣本中，兩個變項的關聯性檢定，也就是探討兩個類別變項(例如：性別和結婚狀態)之間，是否為相互獨立，或者是有相依的關係存在，若是達到顯著，則需進一步查看兩個變項的關連性強度。

二、前提假設：

- 所有的變項為類別變項(categorical variable)
- 樣本須為獨立變項(Independent variable)→第一組的樣本不影響第二組的樣本；第二組樣本也不影響第一組。
- 每一檢定細格(cell)內的數據應該設為頻率或計數數目，而不是百分比或是經過轉換之數據。
- 至少有80%以上的細格，其樣本數大於5，亦即樣本數目至少要為細格數目的五倍。

三、計算方式:

- 以觀察值及期望值差異來取得卡方值,進而判斷兩特徵的相關程度。
- 通常卡方值愈大,兩特徵的相關程度愈高。

$$\chi^2 = \sum \frac{(觀察值_i - 期望值_i)^2}{期望值_i}$$

[Ref] https://pse.is/4vkear

[案例說明] https://pse.is/4wn6ud

使用 Scikit-Learn進行特徵選擇

載入Scikit-Learn 的卡方驗證模組

- 先載入模組

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

取出判斷特徵與目標特徵

1.**建立判斷特徵:**

取出資料集前9項欄位為判斷特徵。

DataFrame變數.iloc[列範圍, 行範圍]

	團塊厚度	細胞大小均勻性	細胞形狀均勻性	邊緣粘附	上皮細胞大小	裸核	淡染色質	正常核仁	有絲分裂
0	1	1	1	1	1	1	1	1	1
1	5	1	2	1	2	1	3	1	1
2	1	1	1	3	2	3	1	1	1
3	9	1	2	6	4	10	7	7	2
4	3	1	1	3	2	1	2	1	1
...	...	...	...	...	...	...	...	...	...

使用 Scikit-Learn進行特徵選擇

2.**建立目標特徵:**

取出資料集最後1項欄位為目標特徵, 也就是「種類」的資料是良性或惡性。

```
0    1
1    1
2    1
3    2
4    1
```

進行卡方驗證

- 首先建立 SelectBest物件

卡方物件 = SelectKBest(chi2, k=數值)

k:設定篩選後的特徵數量,預設值=10

- 再利用卡方變數的fit_transform方法進行特徵篩選

卡方陣列變數 = 卡方物件. fit_transform(判斷特徵, 目標特徵)

```
array([[1, 1, 1, 1, 1],
       [1, 2, 1, 1, 1],
       [1, 1, 3, 3, 1],
       ...,
       [1, 1, 1, 1, 1],
       [8, 7, 2, 8, 8],
       [1, 1, 1, 1, 1]])
```

使用 Scikit-Learn進行特徵選擇

取得特徵驗證值

- 以卡方變數的 scores_ 屬性取得所有特徵的卡方驗證值

```
卡方變數 = 卡方變數.scores_
```

```
array([ 613.78738369, 1349.48325388, 1253.69419874,  962.80470656,
        487.94465088, 1616.63540027,  666.18098173, 1113.86253889,
        229.21688093])
```

傳回值是特徵卡方驗證值依序所組成的陣列:第一個特徵值就是「團塊厚度」,餘則依此類推

取得相關特徵名稱

- 卡方驗證值愈大,則相關性愈高;在本範例中取除前五項特徵(n=5)
- 可使用numpy的argsort()函式,可將數值由小到大排序
- 再使用flipud()函式將陣列反轉成由大到小排序
- 取出原始陣列的索引值組成陣列

```
排序變數 = np.argsort(卡方值變數)
排序變數 = np.flipup(排序變數)
```

```
array([5, 1, 2, 7, 3, 6, 0, 4, 8])
```

使用 Scikit-Learn進行特徵選擇

- 使用迴圈由索引值取得特徵名稱並顯示:

```
選擇的特徵:
裸核: 1616.6354002736975
細胞大小均勻性: 1349.4832538808205
細胞形狀均勻性: 1253.6941987391392
正常核仁: 1113.8625388898593
邊緣粘附: 962.8047065573521
```