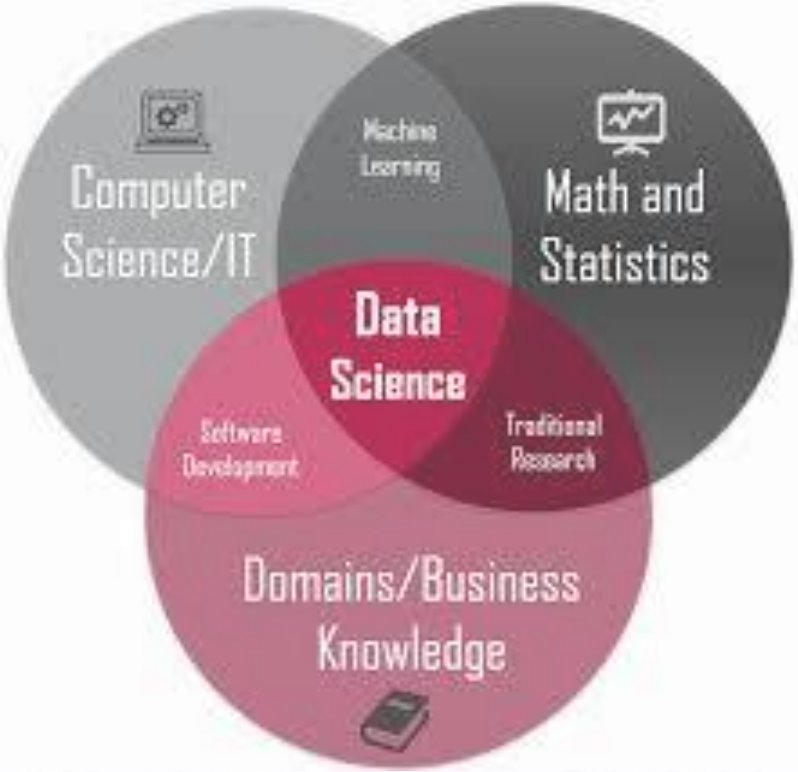
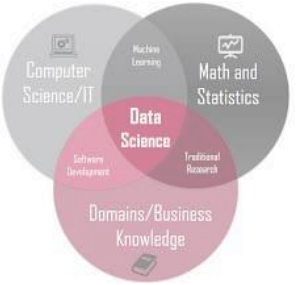
AI & DS



U07-機器學習:非監督式學習

2023.04_V1.0

單元大綱



※機器學習的三大類型與相關演算法

※非監督式學習介紹

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

Part 1 機器學習的三大類型與相關演算法

機器學習的三大類型

機器學習可分為:

1. **監督式學習（Supervised Learning）**

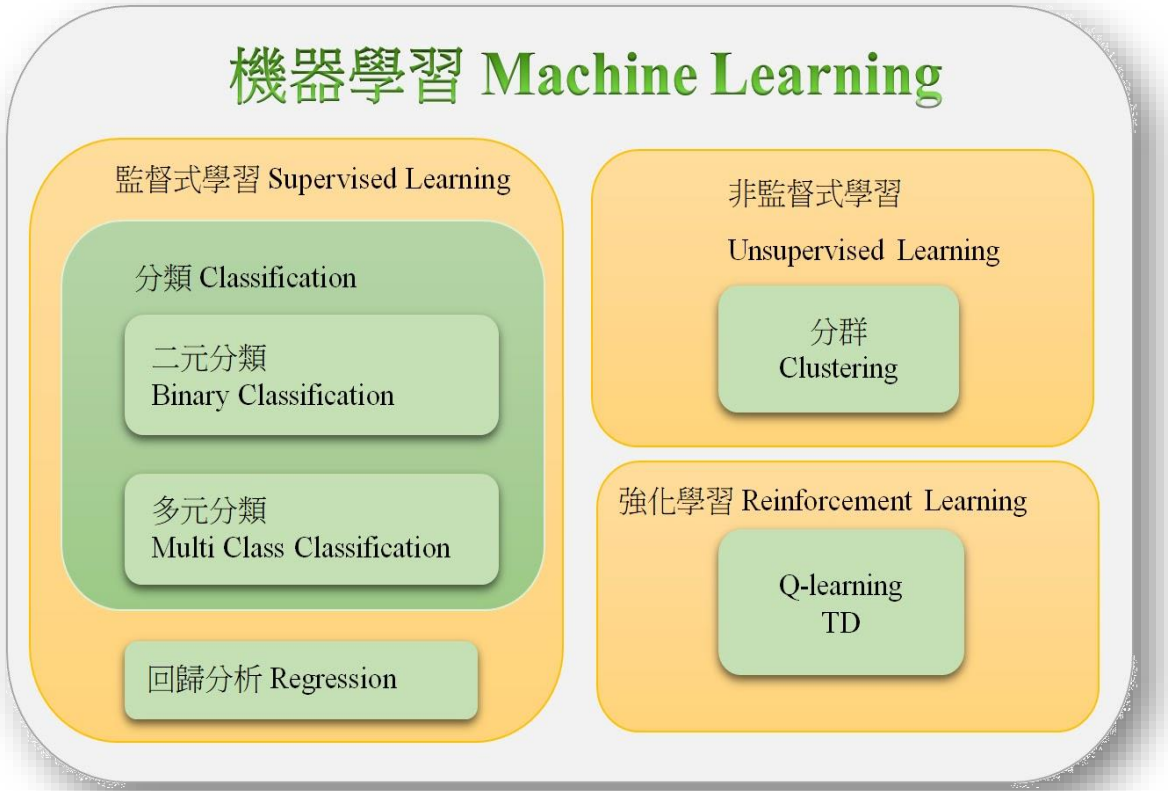
在訓練的過程中告訴機器答案、也就是「有標籤」的資料。

2. **非監督式學習（Unsupervised Learning）**

- 資料沒標籤、讓機器自行摸索出資料規律。
- 訓練資料沒有標準答案、不需要事先以人力輸入標籤，故機器在學習時並不知道其分類結果是否正確。
- 訓練時僅須對機器提供輸入範例，它會自動從這些範例中找出潛在的規則。

3. **增強式學習（Unsupervised Learning）**

- 透過觀察環境而行動，並會隨時根據新進來的資料逐步修正、以獲得最大利益。
- 若環境的變化是離目標更接近、我們就會給予一個正向反饋 (Positive Reward)
- 若離目標更遠、則給予負向反饋 (Negative Reward)



[Ref]: https://ithelp.ithome.com.tw/m/articles/10217849

機器學習的三大類型

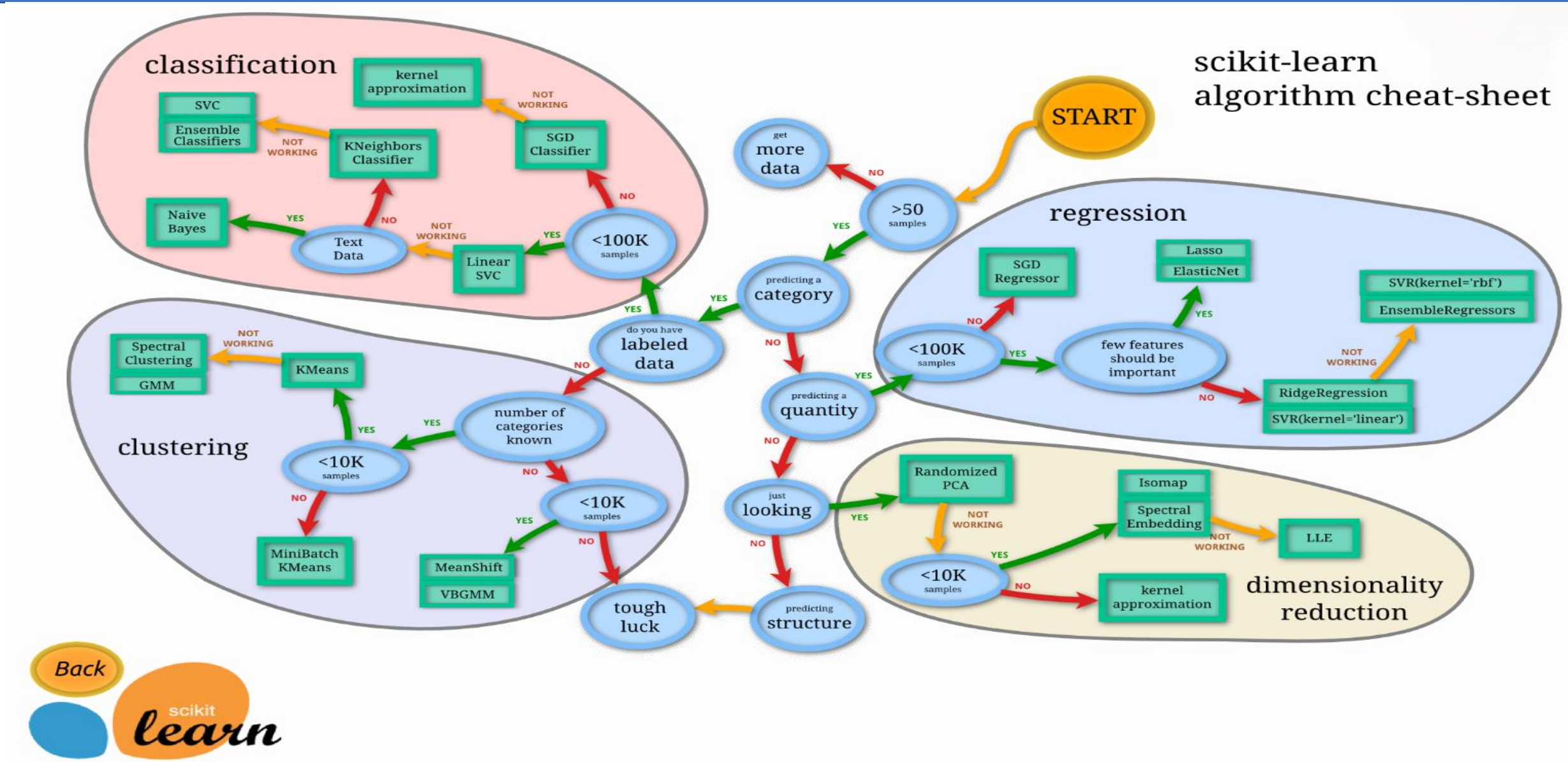
分類	細分類	Features（特徵）	Label（預測目標）
監督式學習	Binary Classification 二元分類	濕度、風向、風速、季節、氣壓…	只有0與1選項（是非題）0：不會下雨、1：會下雨
監督式學習	Multi-Class Classification 多元分類	濕度、風向、風速、季節、氣壓…	有多個選項（選擇題）1：晴天、2：雨天、3：陰天、4：下雪
監督式學習	Regression 回歸分析	濕度、風向、風速、季節、氣壓…	值是數值（計算題）溫度可能是-50～50度的範圍
非監督式學習	Clustering 群集	濕度、風向、風速、季節、氣壓…	無label　Cluster集群分析；目的是將資料依照特徵，分成幾個相異性最大的群組，而群組內的相似程度最高
強化學習	Q-learning、TD（Temporal Difference）		強化學習的原理，藉由定義：動作（Actions）、狀態（States）、獎勵（Rewards）的方式，不斷訓練機器循序漸進，學會執行某項任務的演算法，常用於動態系統及機器人控制等。

[Ref]: https://ithelp.ithome.com.tw/m/articles/10217849

類別	功能	演算法
監督式學習 Supervised	預測 Predicting	Linear Regression Decision Tree Random Forest Neural Network Gradient Booting Tree
	分類 Classification	Decision Tree Naive Bayes Logistic Regression Random Forest SVM Neural Network Gradient Booting Tree
非監督式學習 Unsupervised	分群 Clustering	K-means
	關聯 Association	Apriori
	降維 Dimension Reduction	PCA

[Ref]: https://pse.is/4x9d2t

Scikit-Learn 演算法地圖



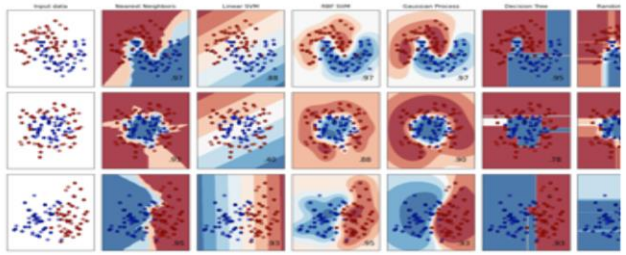
[Ref]: https://scikit-learn.org/stable/index.html

Scikit-Learn 主要功能

分類

識別對象屬於哪個類別。

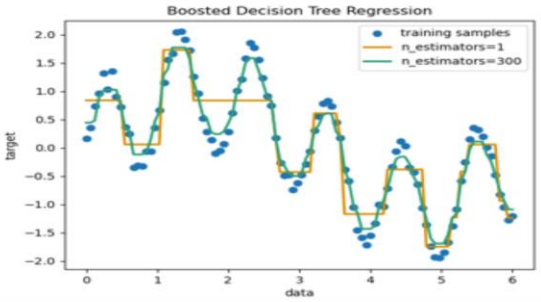
應用： 垃圾郵件檢測、圖像識別。
算法： 支持向量機、 最近鄰、 隨機森林等...



回歸

預測與對象關聯的連續值屬性。

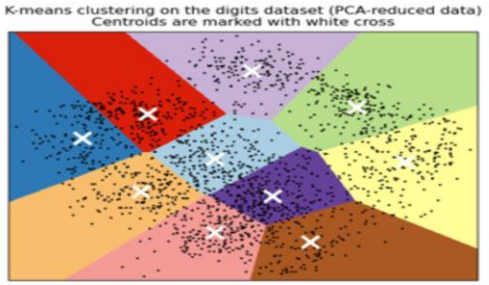
應用： 藥物反應、股票價格。
算法： SVR、 最近鄰、 隨機森林等...



聚類

將相似的對象自動分組到集合中。

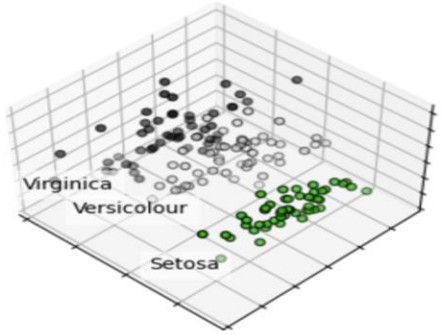
應用： 客戶細分、分組實驗結果
算法： k-Means、 譜聚類、 均值偏移等...



降維

減少要考慮的隨機變量的數量。

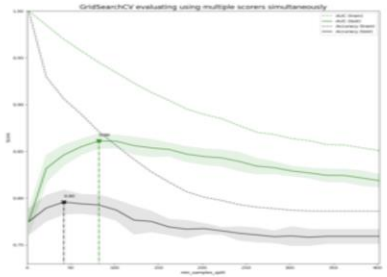
應用： 可視化、提高效率
算法： PCA、 特徵選擇、 非負矩陣分解等...



選型

比較、驗證和選擇參數和模型。

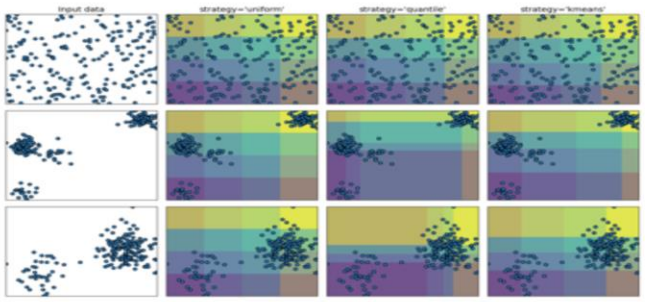
應用： 通過參數調整
算法提高準確性： 網格搜索、 交叉驗證、 指標等...



預處理

特徵提取和歸一化。

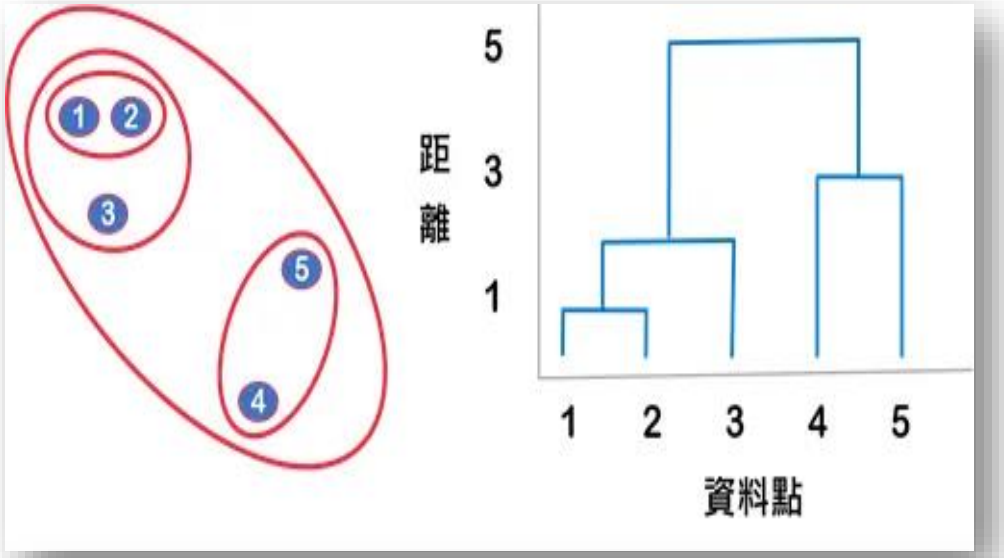
應用： 轉換輸入數據，例如用於機器學習算法的文本。
算法： 預處理、 特徵提取等...



Part 2 群聚分析(Clustering)與K-Means 演算法

Clustering- 群聚(集群)分析介紹(1/2)

- **集群分析**是一種精簡資料的方法，依據樣本之間的共同屬性，將比較相似的樣本聚集在一起，形成集群(cluster)。通常以距離作為分類的依據，相對距離愈近，相似程度愈高，分群之後可以使得群內差異小、群間差異大。



[比較]區別分析(右下圖) v.s 集群分析(右上圖)

- 區別分析→將**事先已分類好的觀察值**，選取有分類效果的樣本，求出其判別函數，再將觀察值進行適當分類。
- 集群分析→**不需事先將觀察值分類**，直接以觀察值的屬性進行分析。

[Ref] https://pse.is/4wft9e

Clustering- 群聚(集群)分析介紹(2/2)

1.**階層式集群分析法(Hierarchical method)**:

(1) **凝聚分層法(Agglomerative)**：開始時，每一個體為一群，將距離最近的兩個個體合成一群，一步步地使群組越變越少，最後所有的個體結合成一群。

(2) **分離分層法(Divisive)**：先將所有個體視為同一個群體，再將相異性較大的個體一步步分成兩群、三群，直到每個體為一群。(※此法較不常用)

2.**非階層式集群分析法(Non-hierarchical method)**:

將原有的集群打散，並重新形成新的集群。Ex: K平均數集群分析法 (K-Means) (非監督式)

3.**兩階段法(最常使用)**:

(1) 第一階段以階層式集群分析法分群，決定集群個數。

(2) 第二階段再以K平均數集群分析法移動各群集內的個體，保持全部集群為k群為止。

[Ref] https://www.yongxi-stat.com/cluster-analysis/

K平均數集群分析法 (K-Means)的演算步驟

S-1.先設定好要分成多少(k)群。選定K個初始集群的中心，其中 K 是欲分群的數目

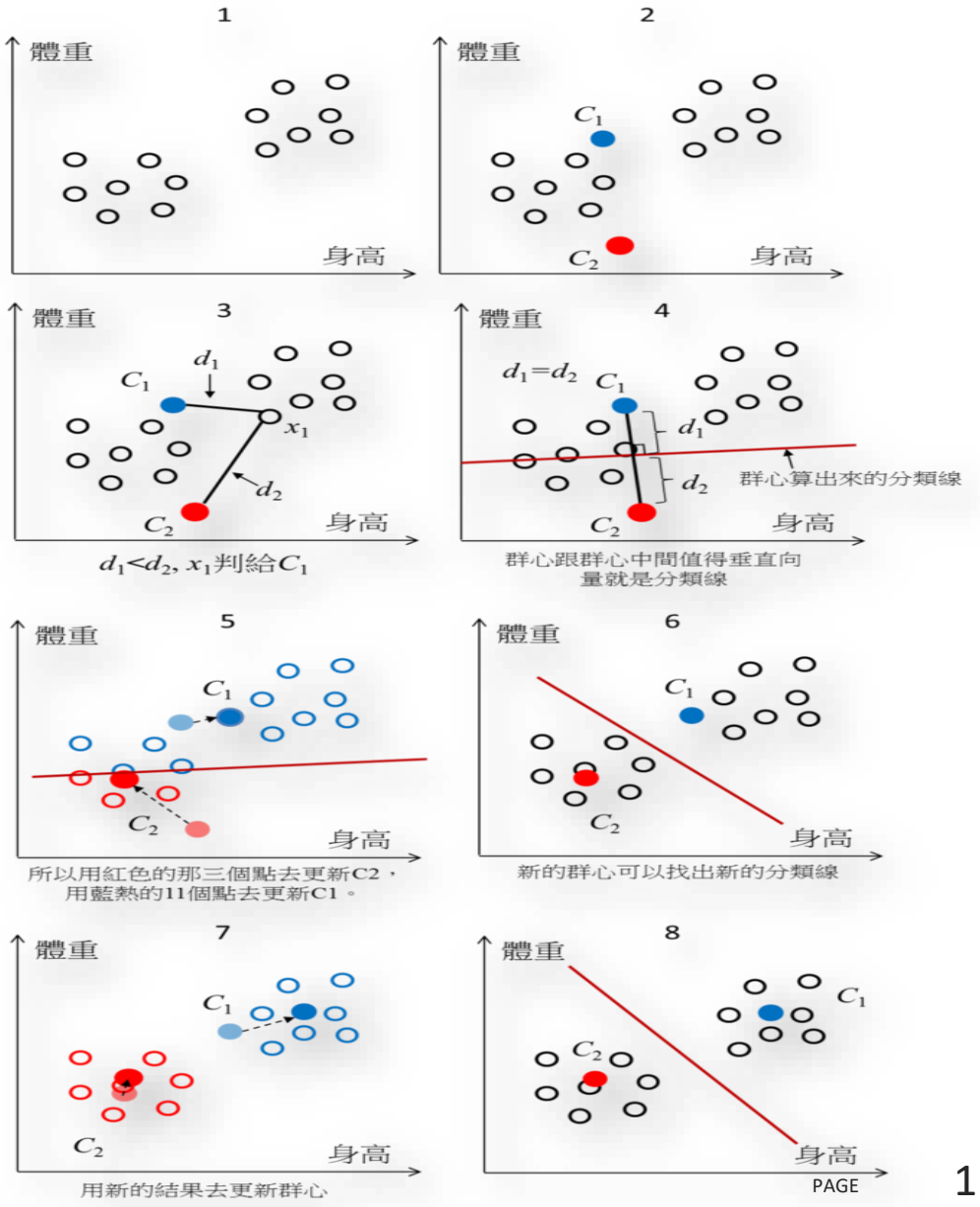
S-2.在feature space(x軸身高和y軸體重組出來的2維空間，假設資料是d維，則會組出d維空間)隨機給k個群心。

S-3.每個資料都會所有k個群心算歐式距離(歐基李德距離Euclidean distance，其實就是直線距離公式，從小學到大的那個距離公式，也可以換成別種距離公式，基本上都還是以歐式距離為主)。

S-4.將每筆資料分類判給距離最近的那個群心。

S-5.每個群心內都會有被分類過來的資料，用這些資料更新一次新的群心。

一直重複S-3~S-5，直到所有群心不在有太大的變動(完成收斂) 。



[Ref] https://pse.is/4wk57g

K-Means 演算方式

S-1.衡量尺度(Scalling):

1.**標準化(Standardization)**:

適用於近似常態分配的定量變數，計算:

x_i 離 $\bar{x}$ 有多少標準差

其中x_i :原始值, $\bar{x}$:平均數, s:標準差

臨界值 $z = \frac{x_i - \bar{x}}{s}$, $s \neq 0$

2.**正規化(Normalization)**:

定量變數若未近似常態分配也可適用，此方法會將數據收斂到0到1之間。

$$\frac{x_i - \text{最小值}}{\text{最大值} - \text{最小值}} \in [0,1]$$

3.**平均值正規化**:

此方法會將數據收斂到-1到1之間, 且平均值為0。

$$\frac{x_i - \text{平均值}}{\text{最大值} - \text{最小值}} \in [-1,1]$$

S-2.決定權重:

目的是越接近的樣本影響力愈大。

S-3.計算距離:

最常使用的是**歐基里德距離(Euclidean Distance)**:

兩點間最短的距離，也就是斜邊距離。

AB距離 $= \sqrt{(X_B - XA)^2 + (Y_B - YA)^2}$

Scikit-Learn 的K-Means模組

※ 載入Scikit-Learn的K-Means模組

```
from sklearn.cluster import KMeans
```

※ 建立KMeans物件

```
Kmeans 變數= KMeans(n_clusters=數值)
```

n_clusters: 要建立的群組數量, 即k值。Ex: km = KMeans(n_clusters=5)

※ 利用fit ()方法進行訓練

```
Kmeans 變數. fit (訓練資料)
```

訓練完成後的分群結果儲存在傳回值的 label_ 屬性中, 是個0到k-1數值組成的串列。

calinski_harabasz_score模組

K-Means演算法分群效果的評分原理:

- 群內資料的距離, 越小越好。不同群的資料,距離愈大愈好
- calinski_harabasz_score模組可對K-Means演算法分群結果評估。
- 此模組計算:不同群資料平均距離與群內資料平均距離的比值,分數越大,分群效果佳。

※ 載入calinski_harabasz_score模組

```
from sklearn.metrics import calinski_harabasz_score
```

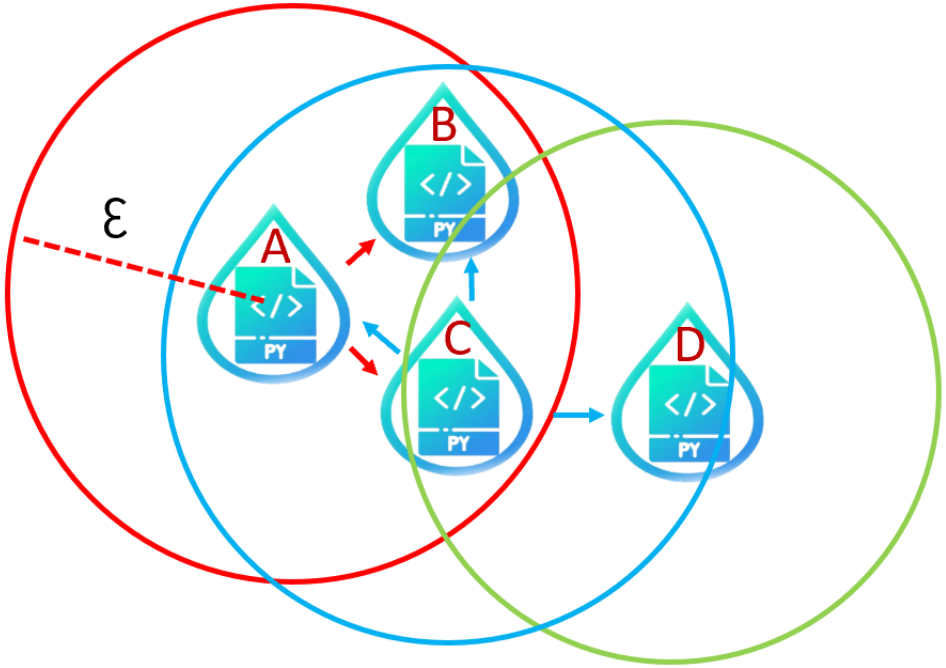
※ 使用calinski_harabasz_score 語法

```
評估變數= calinski_harabasz_score(原始資料, KMeans變數.labels_)
```

Part 3 DBSCAN演算法

DBSCAN演算法介紹(1/2)

- DBSCAN的全名是Density-based spatial clustering of applications with noise。基於密度的分群方式。
- 簡單來說，就是會將特徵相近且密度高的樣本劃分為一群，並標示出特徵較遠密度較稀疏的局外點，另做分群。
- 在DBSCAN中，有兩個主要的參數:
- **半徑: ε (eps)**，由這個參數值為半徑劃出的圓型區域稱為 ε-鄰域。(eps變大, 分群數量與異常值就減少!反之,則增加!)
- **距離: minPts**，構成高密度區域需要最少有幾個點。
- 將樣本當圓心，以距離為半徑畫圓，觀察圓內會包含多少樣本點。最少點數為設定的門檻值，也就是說這個圓內需要至少包含的樣本數目。
- 假設設定最少點數為3，若圓內**樣本數少於最少點數**，則稱此中心點的樣本為**非核心點**(如右圖D)，不論圓內有多少樣本點都不可達。
- 若**樣本數大於等於最少點數**，則此中心點的樣本為**核心點**(如上圖A,C)，可到達圓內任何一點，稱為**可達性**(單向箭頭)。若樣本與樣本之間為雙向可達，稱為**連結性**(雙向箭頭)。
[Ref] https://pse.is/4x6smx

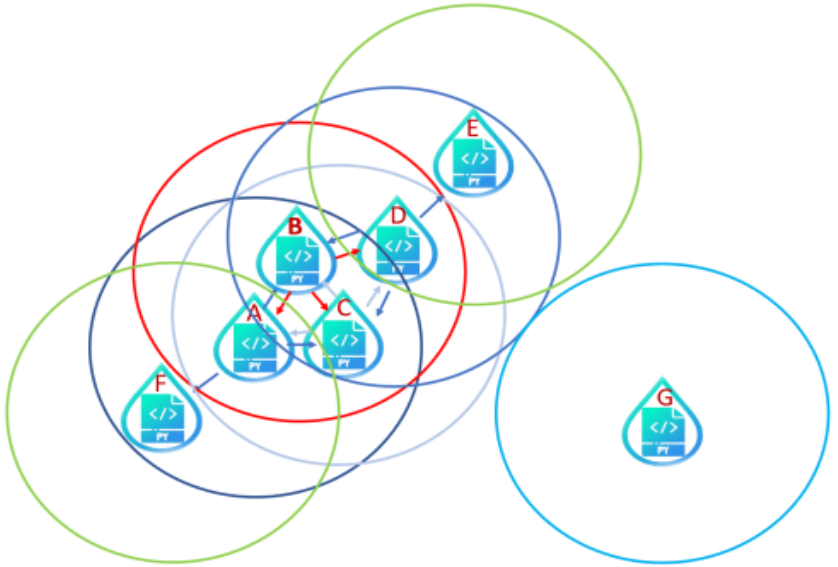


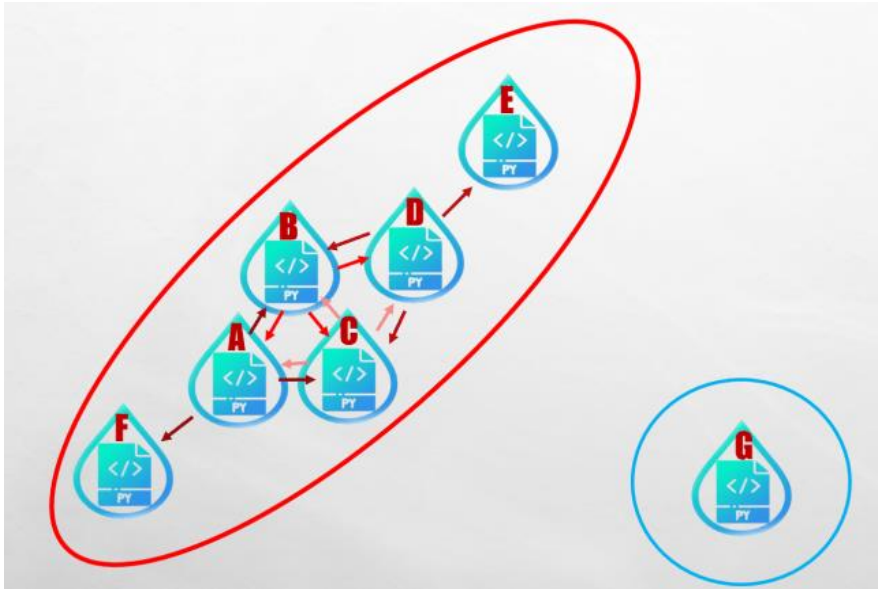
DBSCAN演算法介紹(2/2)

- **DBSCAN流程:**

1. 參數設定: 決定距離(半徑) ε 與 最少點數(門檻值)
2. 任意選取一個樣本當作中心點，以步驟1設定好的半徑畫圓。
   (1)若**圓內樣本數大於等於門檻值**，則此一樣本為**核心點**，標記可達到圓內任一點。
   (2)若**圓內樣本數小於門檻值**，雖本身不可達，但被核心點可達，則稱之為**邊界點**。
   (3)若**圓內樣本數小於門檻值**，且不被核心點可達，則稱之為**局外點**(或**雜訊**)。
3. 對每一個樣本重複步驟2的動作，直至所有樣本都當過中心點為止。
4. 分群: 將有連結性(雙向可達)的樣本點劃分為一群，並納入單向可達的邊界點。其他局外點(雜訊)則劃分為另一群。

- 在圖中ABCD之間互有雙向箭頭，也就是所謂的連結性。在A與D之間雖然沒有直接的連結性，可透過BC兩點連結。因此，ABCD屬於同一群。而E與F雖然與中間ABCD樣本沒有連結，但同被這個群體可達，因此也屬於ABCD這群。
  [Ref] https://pse.is/4x6smx





Scikit-Learn 的DBSCAN模組

※ 載入Scikit-Learn的DBSCAN模組

```
from sklearn.cluster import DBSCAN
```

※ 建立DBSCAN物件

```
DBSCAN 變數= DBSCAN(eps=數值, min_samples=數值)
```

eps: 密度半徑, 預設值=0.5。
min_samples: 以密度半徑畫的圓內包含的最小資料數量,預設值=5。

※ 利用fit ()方法進行訓練

```
DBSCAN 變數. fit (訓練資料集)
```

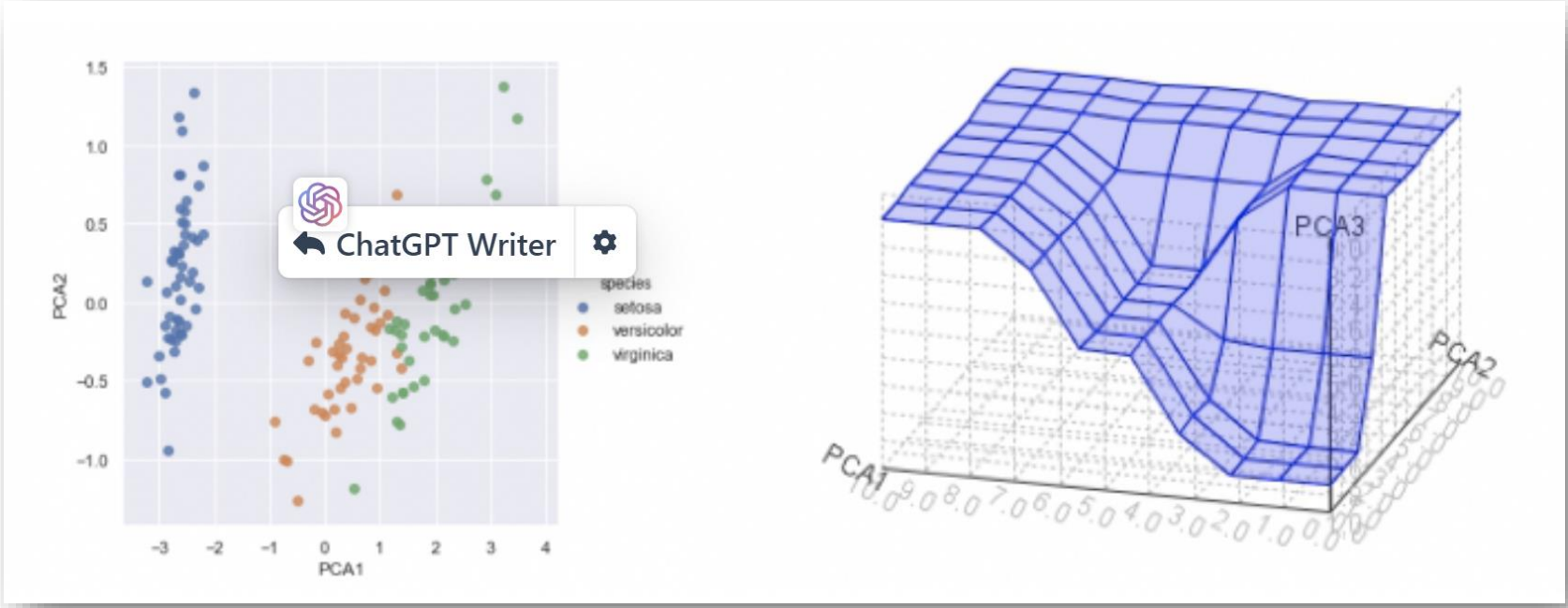
訓練完成後的分群結果儲存在傳回值的 label_ 屬性中, 表示每一個資料分配的群組。

Part 4

降維演算法

降維 (Dimension Reduction)演算法介紹

- 一般資料常見的表示方法有一維(數線)、二維(XY平面)和三維(XYZ立體)。當大於三維的資料就難以視覺化呈現，該如何表示高維度的資料同時又不能壓縮原本資料間彼此的關連性呢？
- 降維顧名思義，就是原本的資料處於在一個比較高的維度作標上，我們希望找到一個低維度的作標來描述它，但又不能失去資料本身的特質。
- 為何降維?可以用比較少的空間，或是計算時用比較少的資源就可以得到跟沒有做資料壓縮之前得到相似的結果。
- 資料降維可以進行資料視覺化，二維可以用平面圖表示、三維可以用立體圖作表示，而大於三維的空間難以視覺化做呈現。

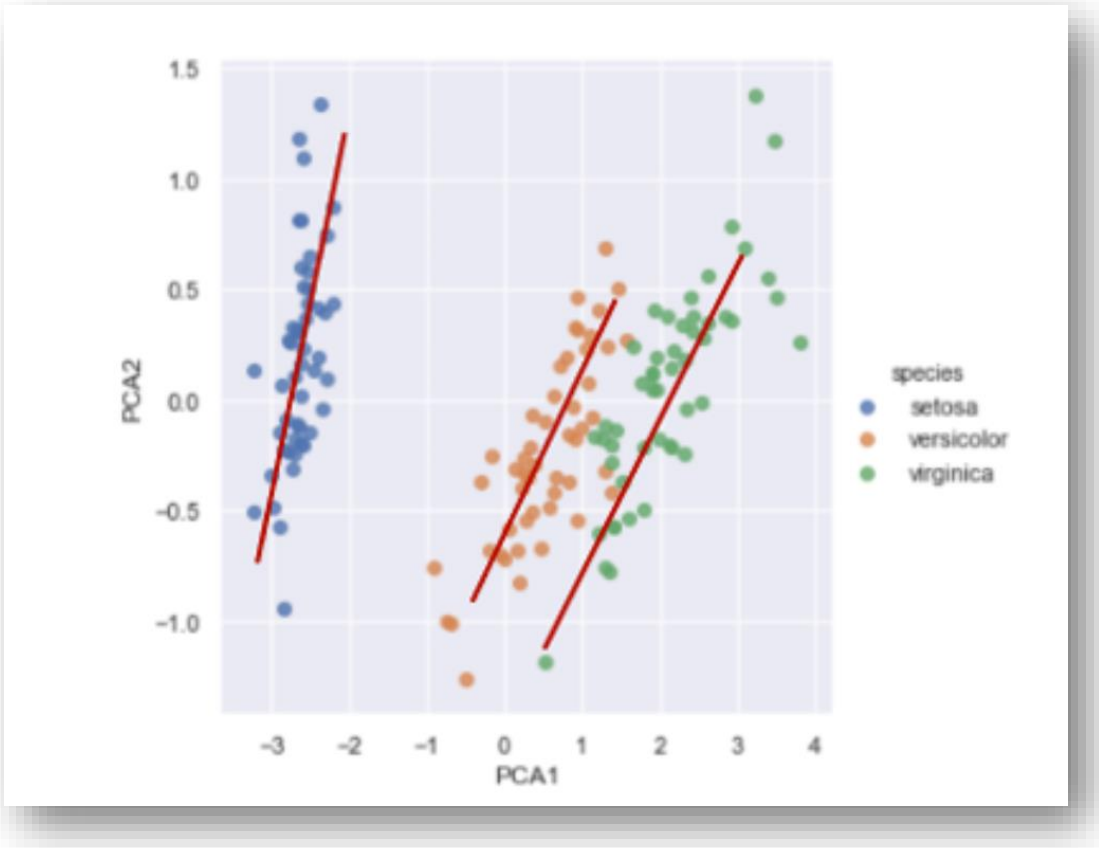


- [Ref] https://pse.is/4w6hcg

主成份分析(Principal component analysis, PCA)介紹

- PCA主要目的是把高維的點頭影到低維的空間上，並且低維度的空間保有高維空間中大部分的性質。
- 透過將一個具有 n 個特徵空間的樣本，轉換為具有 k 個特徵空間的樣本，其中 k 必定要小於 n。此外 PCA 只允許線性的轉換。
- 如右圖所示，將鳶尾花朵資料集進行 PCA 降維。將原有四個特徵分別有花瓣與花萼的長與寬，透過線性轉換成兩維並投射在平面上。可以發現三種花的類別在平面上各自都有線性的趨勢，也就是下圖中紅色的線條。
- PCA的主要步驟:

  1.先求出所有資料點中心 μ
  2.將每一個資料點減去 μ
  3.計算特徵的協方差矩陣
  4.對矩陣進行特徵值分解
  5.取出最大的 k 個特徵值對應的特徵向量
  6.將資料點投影到選取的特徵向量上
- [Ref] https://pse.is/4w6hcg



Scikit-Learn 的PCA模組

※ 載入Scikit-Learn的PCA模組

```
from sklearn.decompostiton import PCA
```

※ 建立PCA物件

```
主成份變數= PCA(n_components=數值)
```

n_components: 此參數有兩種設定方式:

小數:表示要保留原始資料的比例(值一般都在0.9 - 0.95之間)。
整數:表示要保留原始資料的特徵數量,尾數通常是小數。

※ 利用fit_transform ()方法進行訓練

```
轉換變數 = 主成份變數. fit (數值資料串列)
```