Python
資料科學與
人工智慧
應用實務
Journey to
Data Scientists
with Python

Artificial
Intelligence

第16章
機器學習演算法
實作案例
– 分類與分群

16-1 決策樹

16-2 K鄰近演算法

16-3 K-means演算法

16-1 決策樹

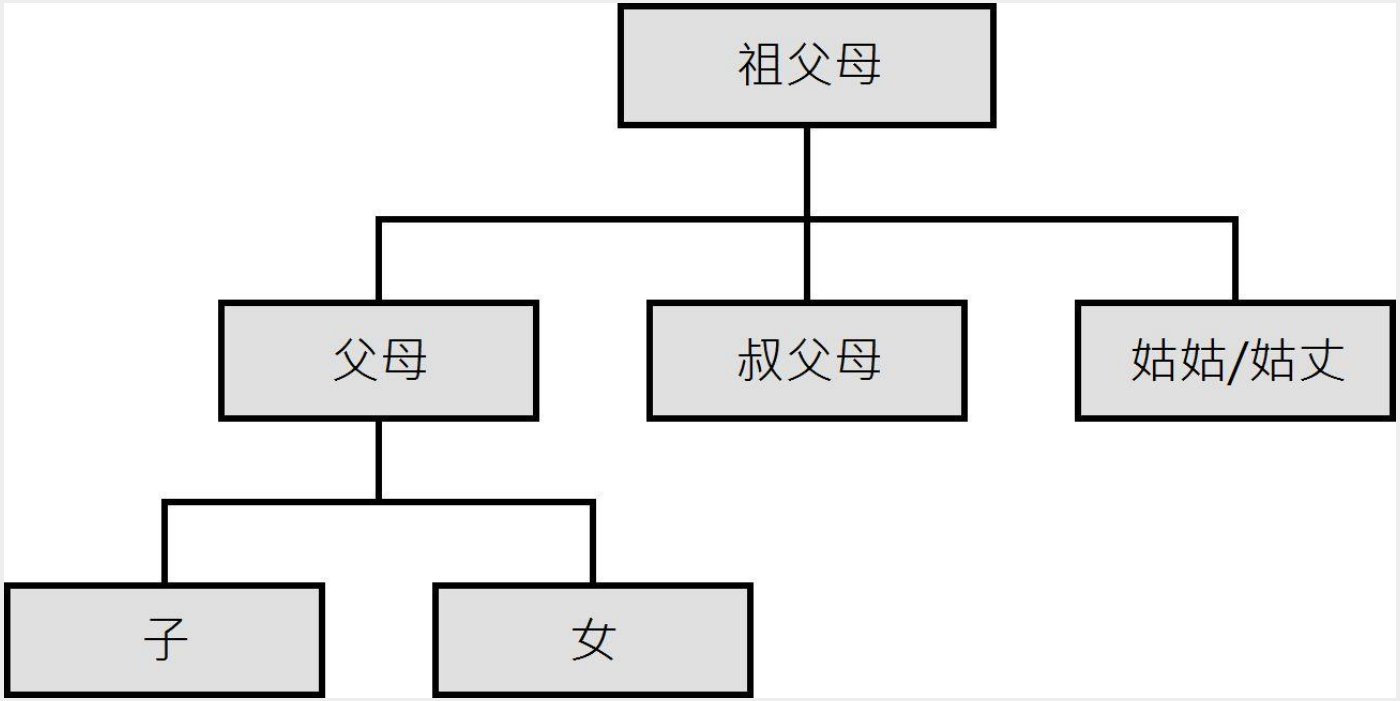
16-1-1 認識樹狀結構和決策樹

16-1-2 使用決策樹的鐵達尼號生存預測

16-1-3 使用決策樹分類鳶尾花

16-1-1 認識樹狀結構和決策樹 – 什麼是樹

- 「**樹**」（Trees）是一種模擬現實生活中樹幹和樹枝的資料結構，屬於階層架構的非線性資料結構，例如：家族族譜，如下圖所示：

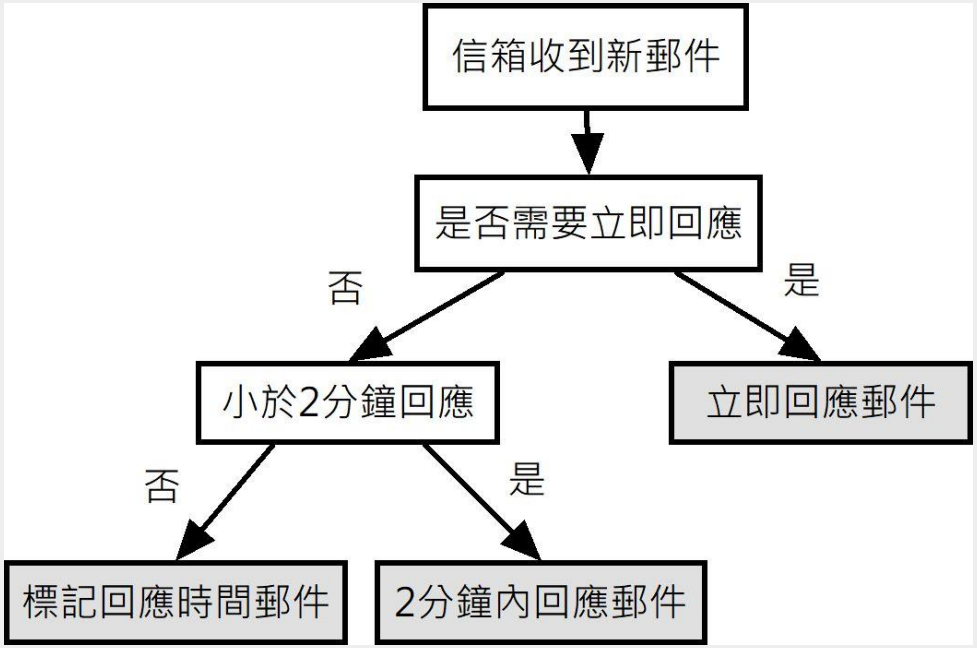


16-1-1 認識樹狀結構和決策樹 - 說明

- 「**決策樹**」（Decision Tree）是使用樹狀結構顯示所有可能結果和其機率，可以幫助我們進行所需的決策，換一個角度，也就是在分類我們觀察到的現象，所以，決策樹就是一種特殊類型的機率樹（Probability Tree）。
- 決策樹基本上是由一序列是與否的條件決策所組成，每一個分支（Branches）代表一個可能的決策、事件或反應，這是一個互斥選項，擁有不同的機率和分類來決定下一步，決策樹可以用來顯示如何和為什麼一個選擇可以導致下一步的選擇。

16-1-1 認識樹狀結構和決策樹 – 範例

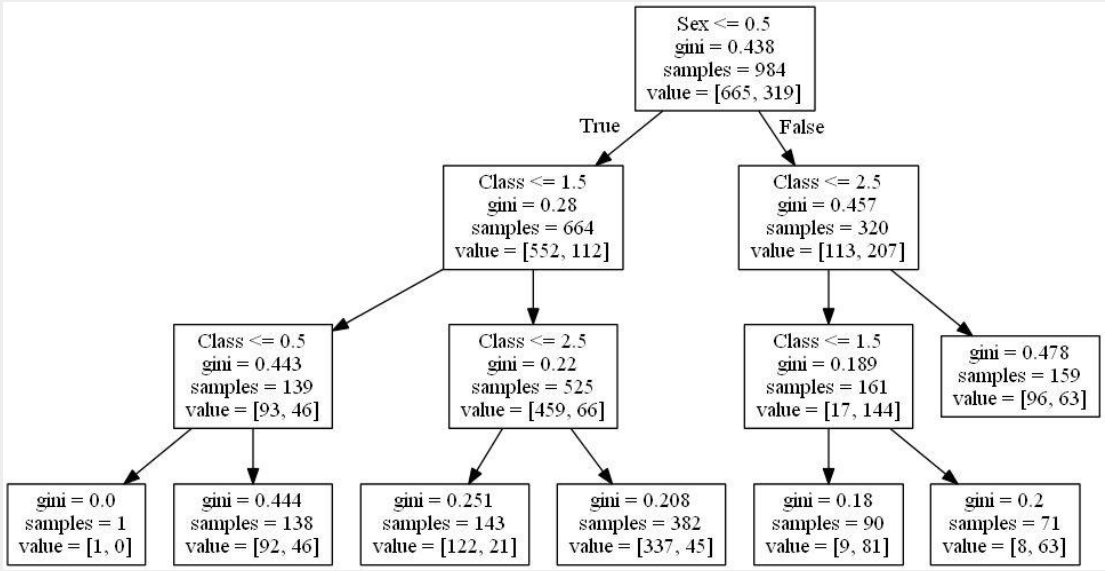
- 例如：電子郵件管理的決策樹，當信箱收到新郵件後，導致2個分支，我們需要決策是否需要立即回應郵件，如果是，就馬上回應郵件；如果不是，將導致另一個分支，是否在2分鐘內回應郵件，如果是，就在2分鐘內回應郵件；不是，就標記回應郵件的時間，如下圖所示：



16-1-2 使用決策樹的鐵達尼號生存預測

- 在第15-4-2節我們是使用Logistic迴歸進行鐵達尼號的生存預測，這一節我們準備改用Scikit-learn套件的決策樹分類器來重新處理鐵達尼號的生存預測（Python程式：Ch16_1_2.py），如下圖所示：

PClass	1		2		3	
SexCode	0	1	0	1	0	1
row_0						
0.100000	0	53	0	0	0	0
0.112676	0	0	0	36	0	0
0.603774	0	0	0	0	0	53
0.666667	41	0	0	0	0	0
0.853147	0	0	29	0	0	0
0.882199	0	0	0	0	117	0

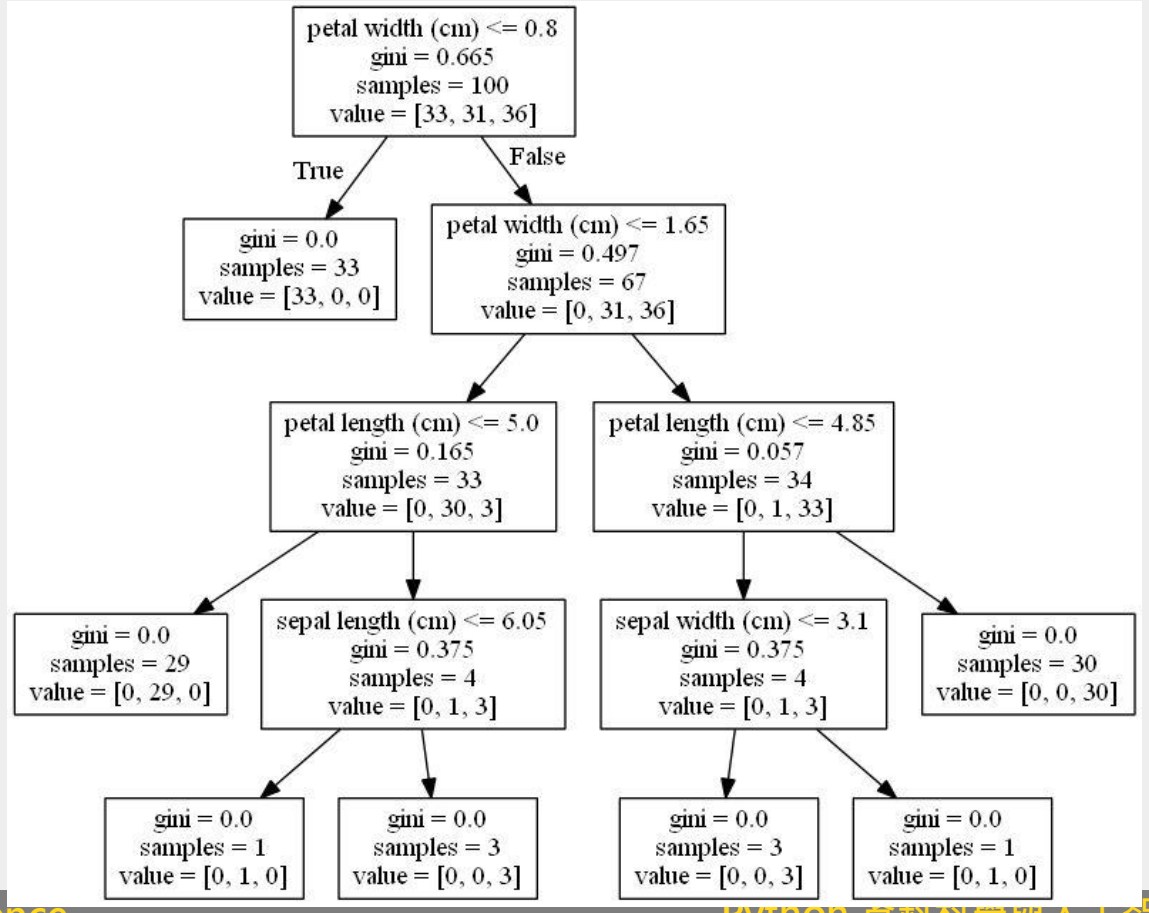


16-1-3 使用決策樹分類鳶尾花

- 在Scikit-learn套件內建的Iris資料集是鳶尾花的資料，可以讓我們訓練模型使用花瓣和花萼來分類鳶尾花。
  - 探索鳶尾花資料集
  - 建立決策樹模型分類鳶尾花

16-1-3 使用決策樹分類鳶尾花

- Python程式：Ch16_1_3b.py建立的tree2.dot檔是使用GraphViz繪出的決策樹圖形，如下圖所示：



16-2 K鄰近演算法

16-2-1 認識K鄰近演算法

16-2-2 使用K鄰近演算法分類鳶尾花

16-2-3 交叉驗證的K值最佳化

16-2-1 認識K鄰近演算法 – 說明

- 分類預測簡單的說，就是使用已知的分類資料建立預測模型來預測未知資料所屬的類別，除了使用第15-4節的Logistic迴歸，第16-1節的決策樹，另一個常見的分類演算法是**K鄰近演算法（KNN）**。
- K鄰近演算法（K Nearest Neighbor Algorithm，KNN）從英文原意即可知，K鄰近演算法是使用K個最接近目標資料的資料來預測目標資料所屬的類別。

16-2-1 認識K鄰近演算法 – 基本步驟(說明)

- 我們準備使用一個簡單實例透過計算的過程來說明K鄰近演算法。例如：某家面紙廠商使用問卷調查客戶對面紙的好惡，問卷共使用2個屬性(耐酸性, 強度)判斷面紙的好或壞，如下表所示：

編號	耐酸性	強度	分類
1	7	7	壞
2	7	4	壞
3	3	4	好
4	1	4	好

- 廠商在今年開發出面紙的新產品，其實驗室測試結果的耐酸性是3；強度是7，在K值3的情況下，請使用K鄰近演算法判斷新產品是好面紙，還是壞面紙。

16-2-1 認識K鄰近演算法 – 基本步驟(Step 1)

- Step 1：計算新產品與所有資料集的距離：我們需要計算新產品與所有資料集其他面紙產品的距離，其公式是各屬性與新產品屬性差的平方和，例如：編號1是(7, 7)，新產品是(3, 7)，各屬性差的平方和是：(7-3)2+(7-7)2 = 42 = 16，如下表所示：

編號	耐酸性	強度	分類	距離(3, 7)
1	7	7	壞	$(7-3)^2+(7-7)^2=16$
2	7	4	壞	$(7-3)^2+(4-7)^2=25$
3	3	4	好	$(3-3)^2+(4-7)^2=9$
4	1	4	好	$(1-3)^2+(4-7)^2=13$

16-2-1 認識K鄰近演算法 – 基本步驟(Step 2)

- Step 2：排序找出最近的K筆距離：在計算出距離後，因為K是3，我們可以找出距離最近3筆的編號是1、3和4，距離分別是16、9和13，距離25被排除，如下表所示：

編號	耐酸性	強度	分類	距離(3, 7)
1	7	7	壞	$(7-3)^2+(7-7)^2=16$
2	7	4	壞	$(7-3)^2+(4-7)^2=25$
3	3	4	好	$(3-3)^2+(4-7)^2=9$
4	1	4	好	$(1-3)^2+(4-7)^2=13$

16-2-1 認識K鄰近演算法 – 基本步驟(Step 3)

- Step 3：新產品分類是最近K筆距離的多數分類：現在，我們知道距離最近3筆編號是1、3和4，其分類分別是壞、好和好，2個好比1個壞，好比較多，所以新產品的分類是「好」，這就是K鄰近演算法。

16-2-1 認識K鄰近演算法 – 範例：分類面紙是好或壞

- 在了解K鄰近演算法的運算過程後，我們可以自行建立Python程式來實作K鄰近演算法，另一種方法是直接使用Scikit-learn套件的K鄰近分類器（Python程式：Ch16_2_1.py）。
- 程式碼建立KNeighborsClassifier物件，參數是K值，然後呼叫fit()函數訓練模型，在完成後，使用新產品資料進行預測分類，其執行結果的分類，如下所示：

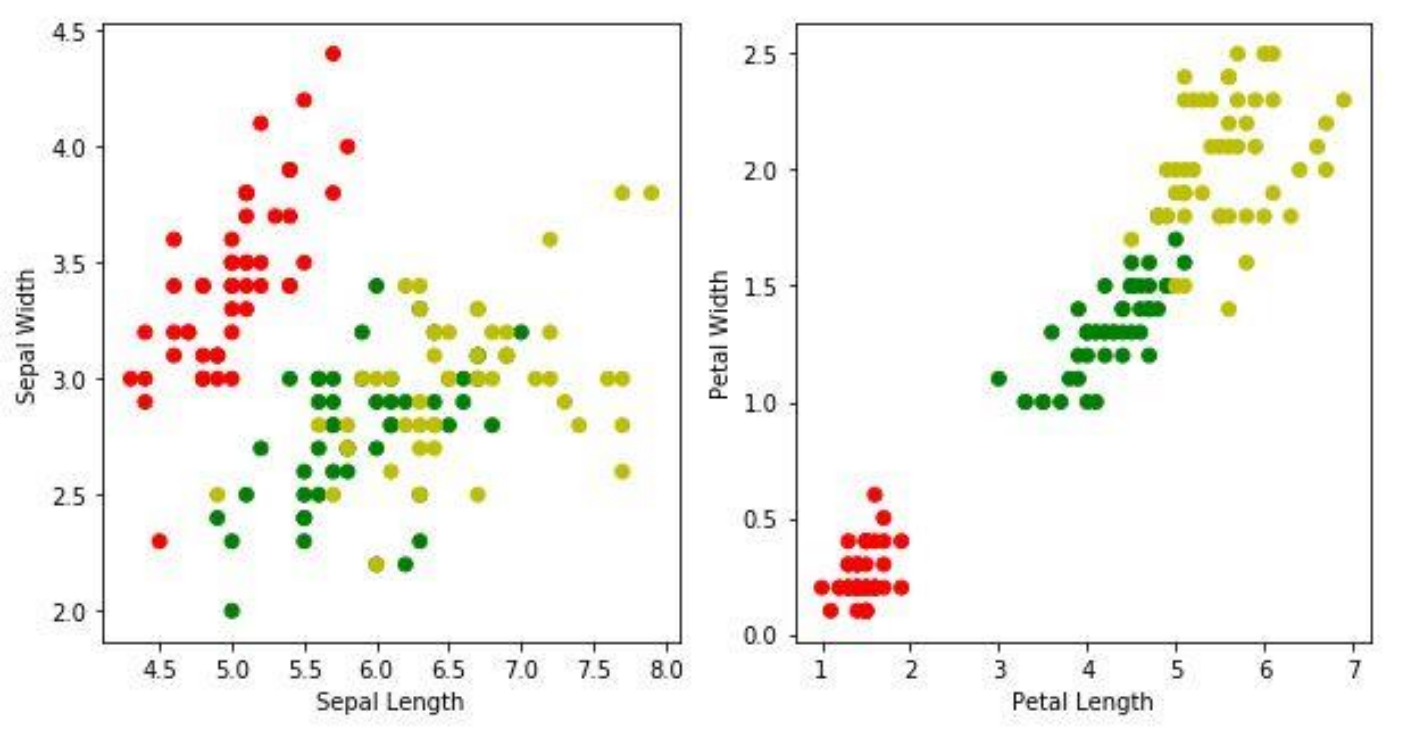
  [1]
- 上述預測結果1，就是「好」；值0是「壞」。

16-2-2 使用K鄰近演算法分類鳶尾花 – 說明

- 我們準備繼續第16-1-3節改用K鄰近演算法來分類鳶尾花，使用的是花瓣和花萼的尺寸，在實際分類之前，我們準備視覺化來探索鳶尾花資料集。
  - 使用散佈圖探索鳶尾花資料集
  - 建立K鄰近模型分類鳶尾花
  - 如何選擇K值

16-2-2 使用K鄰近演算法分類鳶尾花 – 使用散佈圖探索鳶尾花資料集

- 程式碼分別繪出花萼（Sepal）和花瓣（Petal）的長和寬為座標(x, y)的散佈圖，其執行結果如下圖所示：



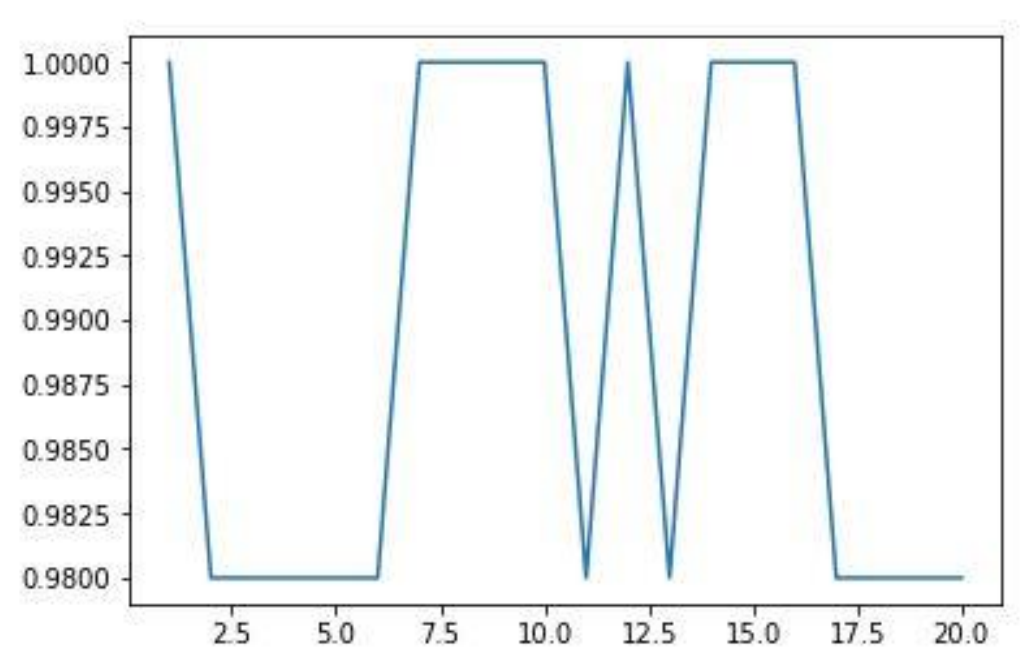
16-2-2 使用K鄰近演算法分類鳶尾花 – 建立K鄰近模型分類鳶尾花

- 現在，我們可以使用K鄰近演算法分類Scikit-learn內建的鳶尾花Iris資料集（Python程式：Ch16_2_2a.py）。
- 程式碼的第1列是測試資料集的預測分類，第2列是原始分類，其執行結果如下所示：

```
[0 1 1 0 2 1 2 0 0 2 1 0 2 1 1 0 1 1 0 0 1 1 1 0 2 1 0 0 1 2 1 2 1 2 2 0
 1 0 1 2 2 0 1 2 1 2 0 0 0 1]
[0 1 1 0 2 1 2 0 0 2 1 0 2 1 1 0 1 1 0 0 1 1 1 0 2 1 0 0 1 2 1 2 1 2 2 0
 1 0 1 2 2 0 2 2 1 2 0 0 0 1]
```

16-2-2 使用K鄰近演算法分類鳶尾花 – 如何選擇K值

- 因為K鄰近演算法的K值會影響分類的準確度，我們可以使用迴圈執行多次不同K值的分類來找出最佳的K值，一般來說，K值的上限是訓練資料集的20%（Python程式：Ch16_2_2b.py），如下圖所示：

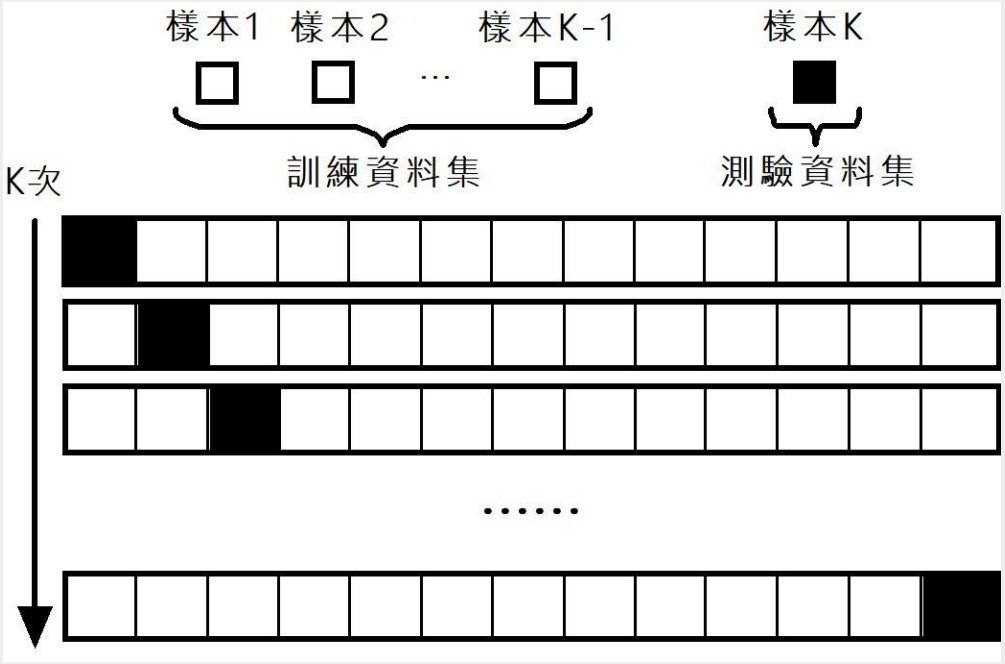


16-2-3 交叉驗證的K值最佳化 – 說明

- 在第15-3-3節是將資料集分割成訓練和測試資料集，使用訓練資料集訓練模型；測試資料集驗證模型，這種方式稱為「持久性驗證」（Holdout Validation）。問題是有些資料並沒有用來訓練，單純只用在驗證，也就是說，我們並沒有使用完整的資料集來進行模型的訓練。

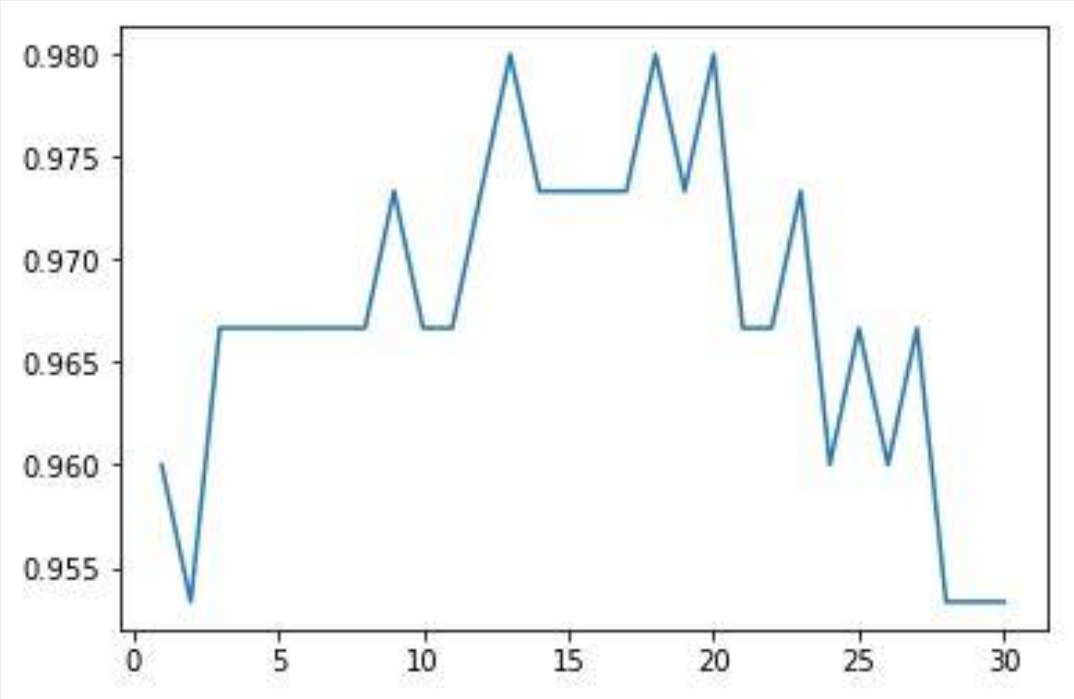
16-2-3 交叉驗證的K值最佳化 – K-fold交叉驗證（K-fold Cross Validation）

- 「交叉驗證」（Cross Validation）是在解決持久性驗證的問題，交叉驗證是將資料集分割成2或更多的分隔區（Partitions），並且將每一個分隔區都一一作為測試資料集，將其他分隔區作為訓練資料集，最常用的交叉驗證是K-fold交叉驗證，如下圖所示：



16-2-3 交叉驗證的K值最佳化 – 交叉驗證的K值最佳化

- 在了解K-fold交叉驗證後，我們可以使用K-fold交叉驗證的cross_val_score()函數來找出最佳K值（Python程式：Ch16_2_3.py），如下圖所示：



16-3 K-means演算法

16-3-1 認識K-means演算法

16-3-2 使用K-means演算法分群鳶尾花

16-3-1 認識K-means演算法 – 說明

- **分群和分類的差異**在於：分類是在已知資料集分類的情況下，替新東西進行分類，分群是在根本不知資料集分類的情況下，直接使用特徵來進行分類，K-means就是機器學習常用的一種分群演算法。
- **K-means分群**（K-means Clustering）也稱為K平均數分群，因為我們並不用知道資料集分類的情況下，即可進行分群，這是一種非監督式學習（Unsupervised Learning）。

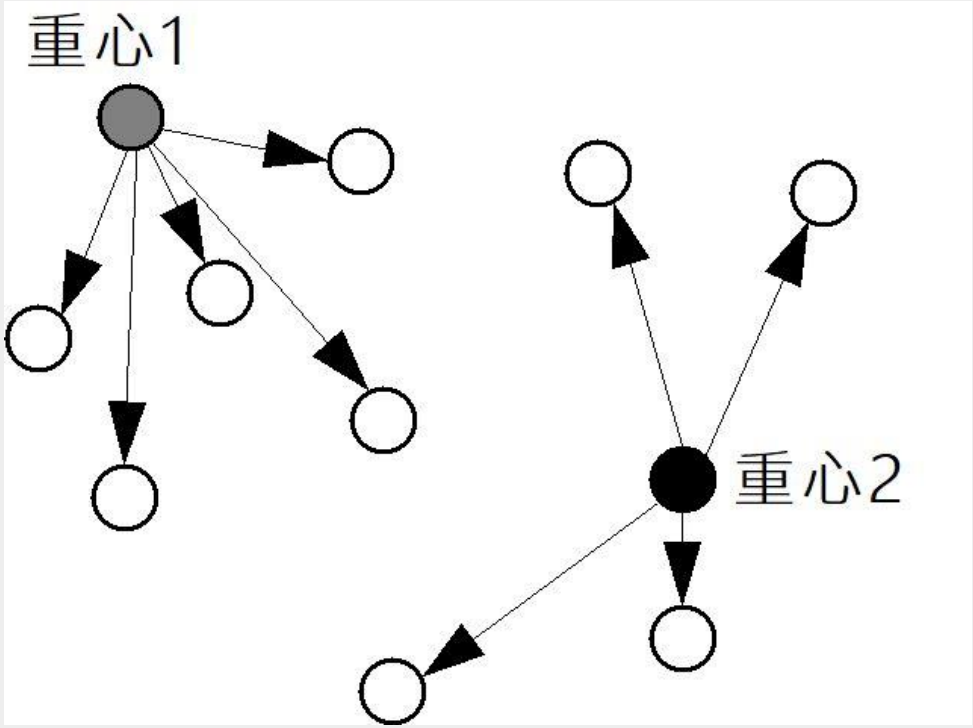
16-3-1 認識K-means演算法 – 基本步驟(Step 1)

- K-means分群的作法是先找出K個群組的重心（Centroid），資料集就以距離最近重心來分成群組後，重新計算群組的新重心後，再分群一次，重複操作來完成分群，其步驟如下所示：

- Step 1：依資料集數決定適當的K個重心，例如：2個。

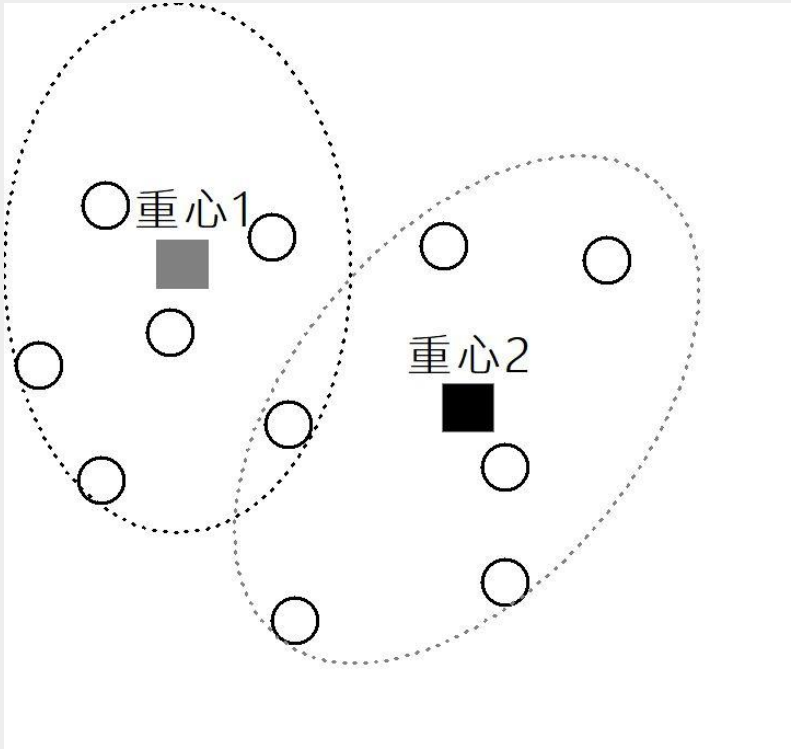
16-3-1 認識K-means演算法 – 基本步驟(Step 2)

- Step 2：計算資料集和重心的距離（公式和K鄰近演算法相同），然後以距離最近重心的資料來分成群組，如下圖所示：



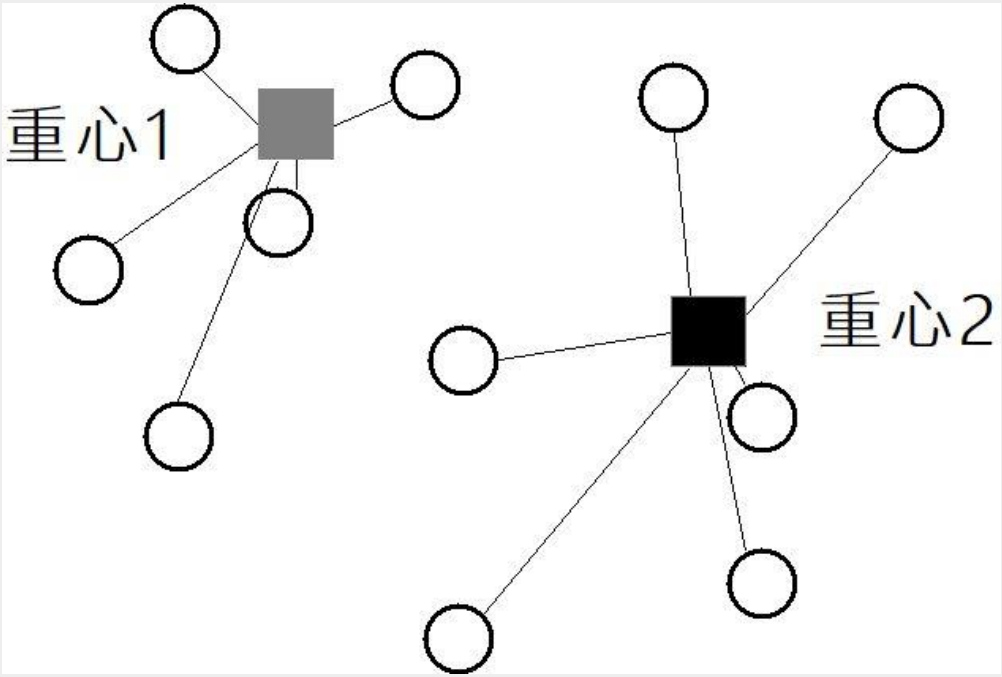
16-3-1 認識K-means演算法 – 基本步驟(Step 3)

- Step 3：重新計算群組資料集各特徵的算術平均數作為新的重心，如下圖所示：



16-3-1 認識K-means演算法 – 基本步驟(Step 4)

- Step 4：再次計算資料集和重心的距離，然後以距離最近重心來分成群組，如下圖所示：



16-3-1 認識K-means演算法 – 基本步驟(Step 5)

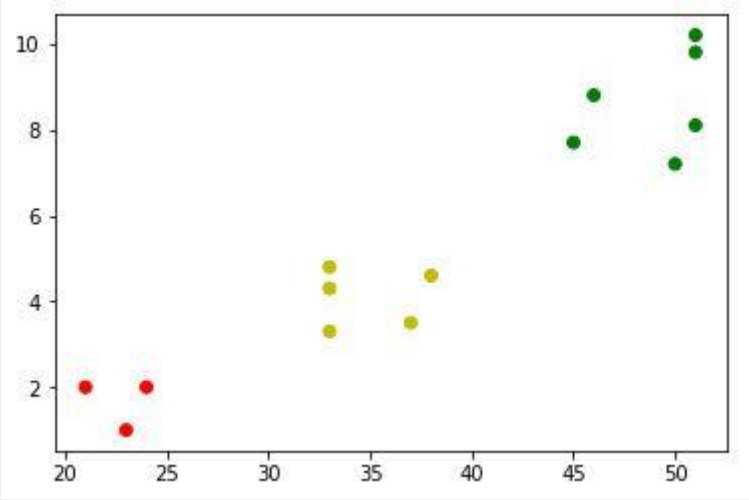
- Step 5：重複操作Step 3~4直到重心和群組不再改變為止。

16-3-1 認識K-means演算法 – 範例：依據動物的體重和身長來分群

- 在動物園收集到14隻動物的體重和身長資料，如下表所示：

身長	51	46	51	45	51	50	33	38	37	33	33	21	23	24
體重	10.2	8.8	8.1	7.7	9.8	7.2	4.8	4.6	3.5	3.3	4.3	2.0	1.0	2.0

- 在K值3的情況下，請使用K-means演算法替14隻動物進行分群，如下所示：



16-3-2 使用K-means演算法分群鳶尾花

- 在第16-2-2節是使用K鄰近演算法分類鳶尾花，和使用散佈圖來視覺化顯示鳶尾花資料集，這一節我們準備改用K-means演算法來分群鳶尾花，事實上，這也是在分類鳶尾花。
  - 建立K-means模型分群鳶尾花
  - 修正分群標籤錯誤重繪散佈圖
  - K-means模型的積效測量