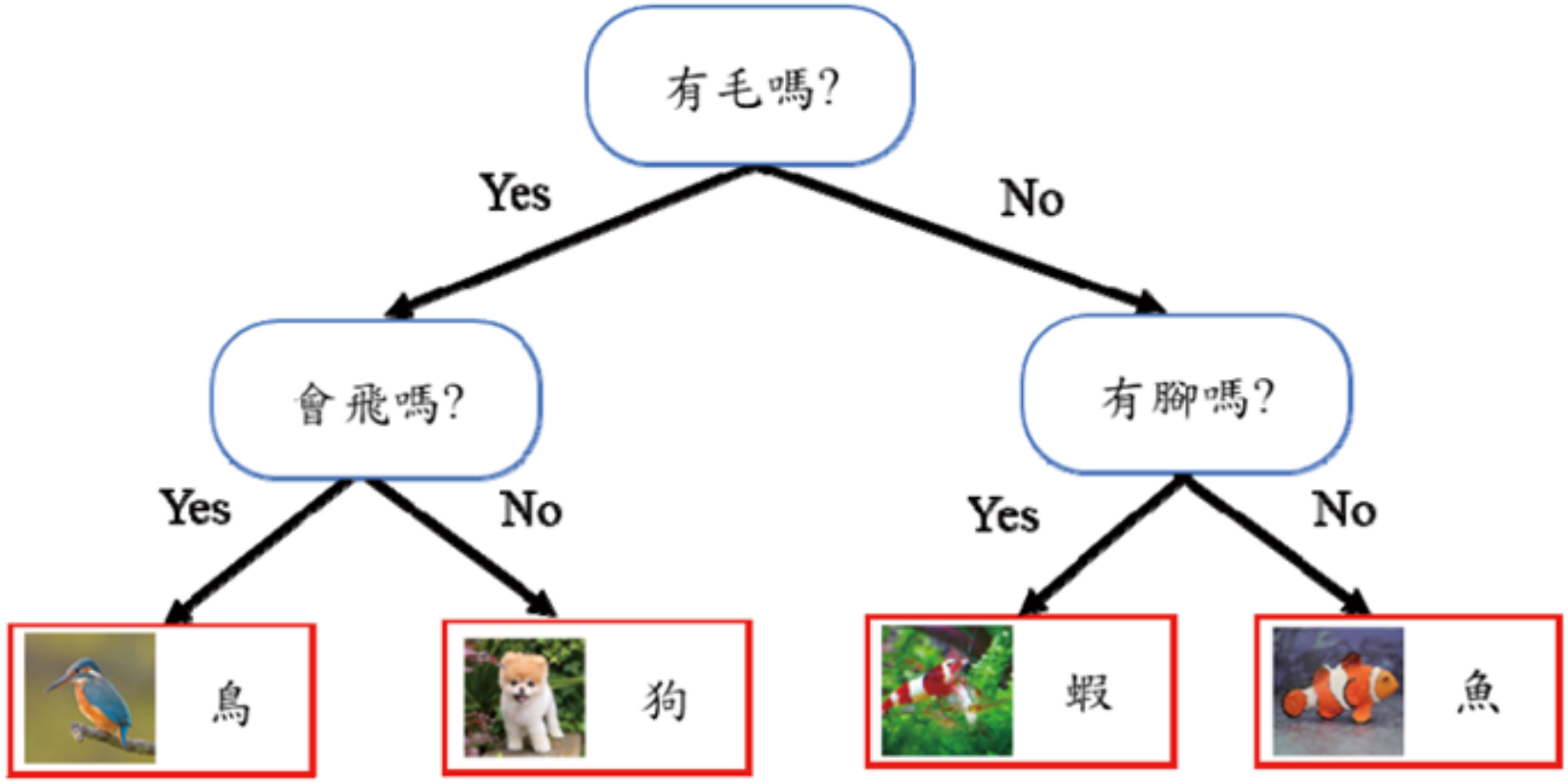
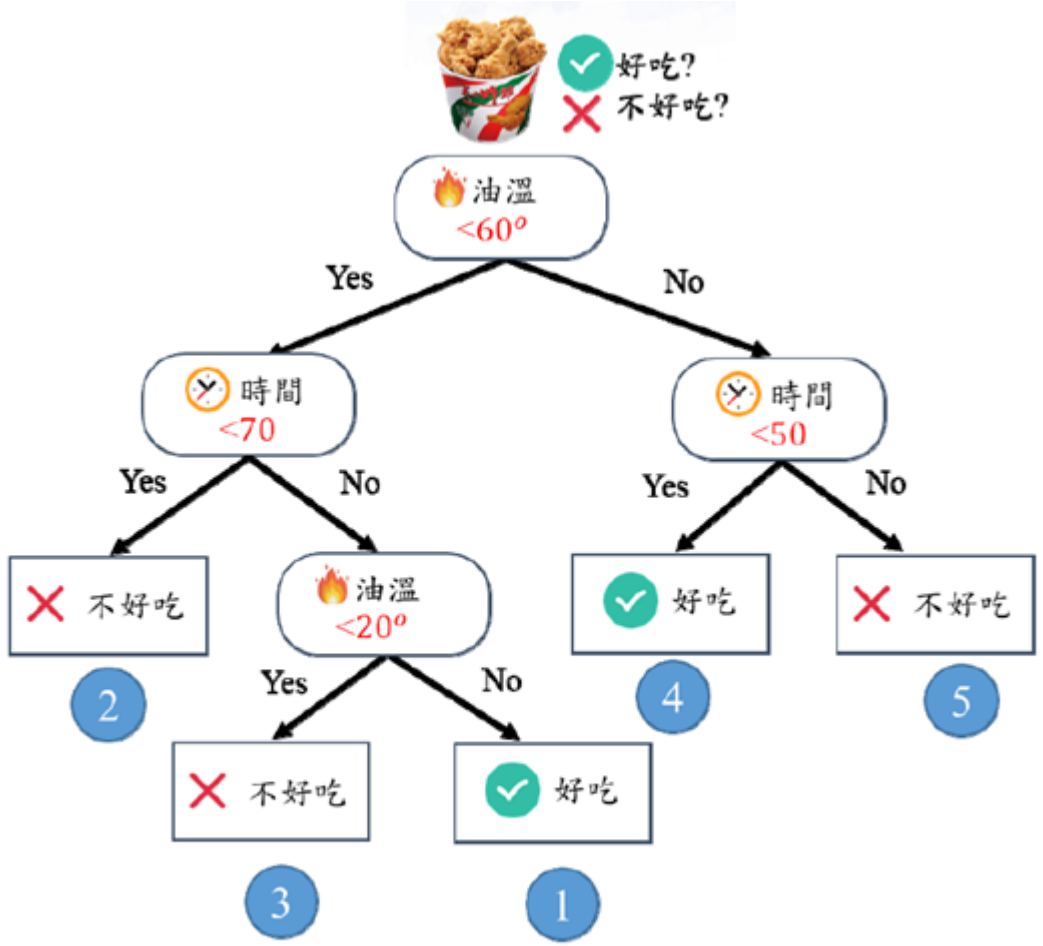
決策樹（Decision Tree）



分辨何種動物的決策樹

決策樹（Decision Tree）



判斷炸雞是否好吃的決策樹

炸雞的數據

編號	油溫	油炸時間	好不好吃
1	50	80	好
2	45	60	不好
3	19	100	不好
4	100	30	好
5	100	70	不好
6	70	30	?

決策樹（Decision Tree）

- 分割原則-資訊增益(Information gain, 簡稱IG)
  - 熵(Entropy)
  - Gini不純度(Gini Impurity)

資訊增益

獲得的資訊量　原本的資訊量　經由分割後的資訊量

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

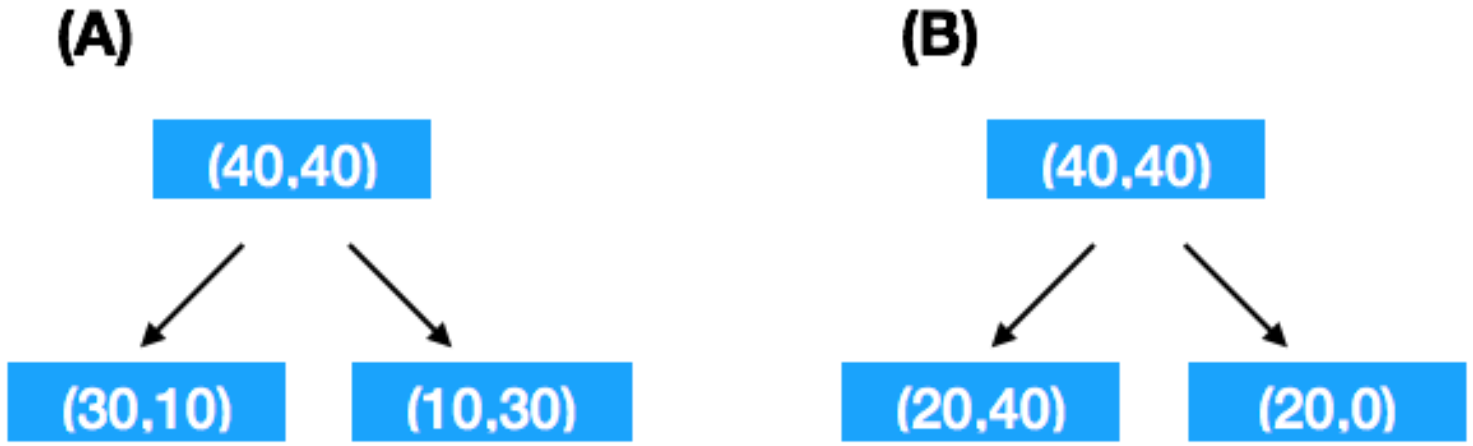
二元分類資訊增益

獲得的資訊量　原本的資訊量　分割後左邊資訊量　分割後右邊資訊量

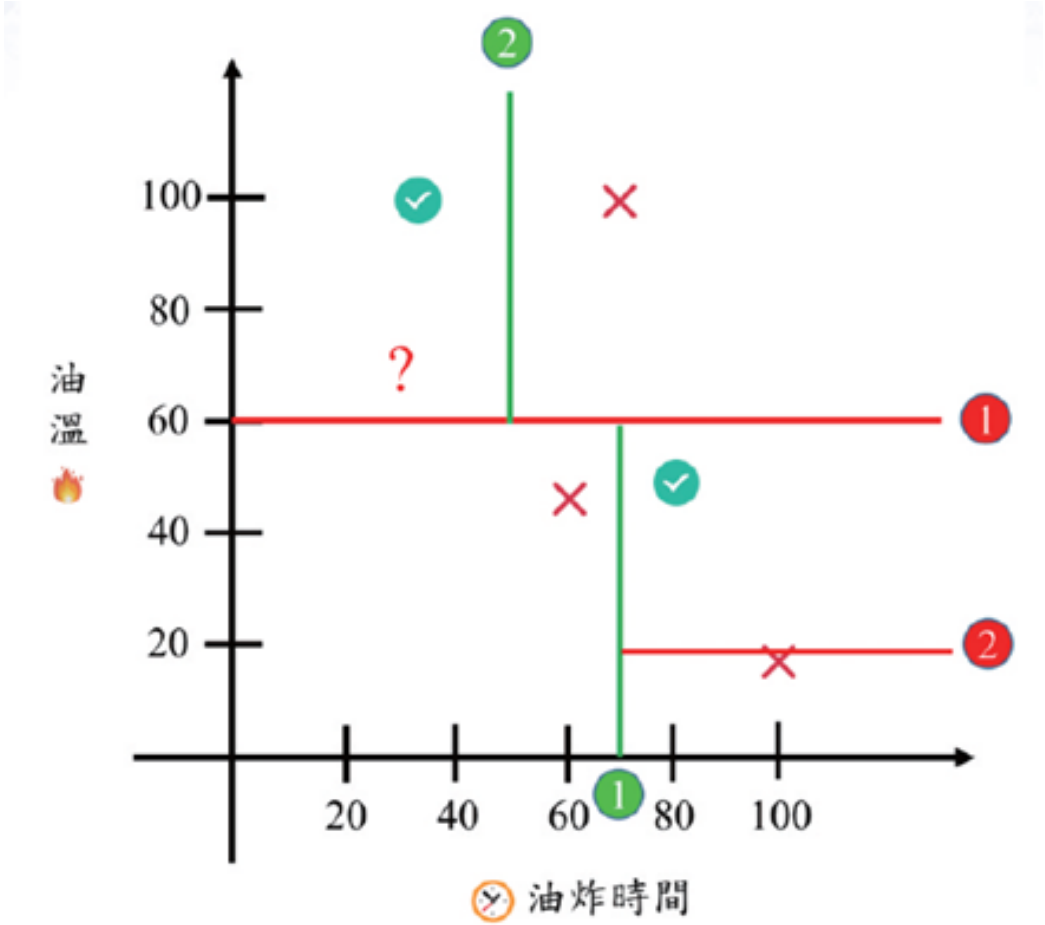
$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

決策樹（Decision Tree）

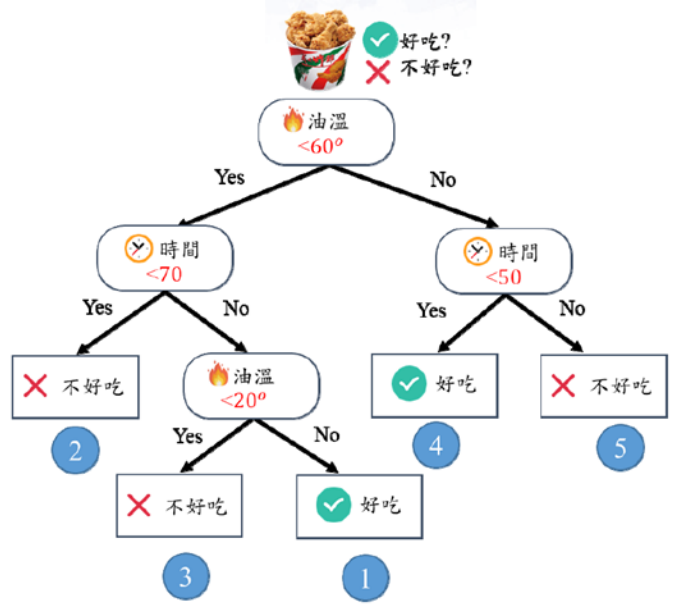
- 分割原則-資訊增益(Information gain, 簡稱IG)
- 有80筆資料，有40是1類別、40筆是2類別。使用兩種不同的切割方法A與B



決策樹（Decision Tree）



以油溫和油炸時間來分類



判斷炸雞是否好吃的決策樹

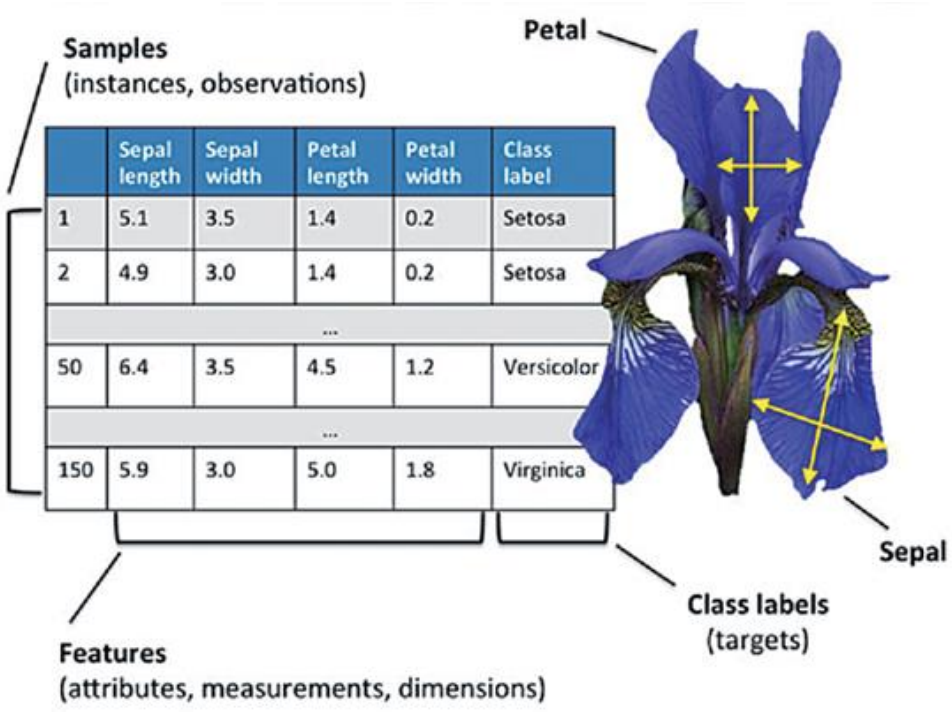
決策樹（Decision Tree）

範例練習

- 以下的範例是鳶(ㄩㄢ)尾花的分類資料集。
- 共有4個欄位，150筆資料。
  - sepal length (cm)：花萼的長度。
  - sepal width (cm)：花萼的寬度。
  - petal length (cm)：花瓣的長度。
  - petal width (cm)：花瓣的寬度。

決策樹（Decision Tree）

- Setosa
- Versicolor
- Virginica

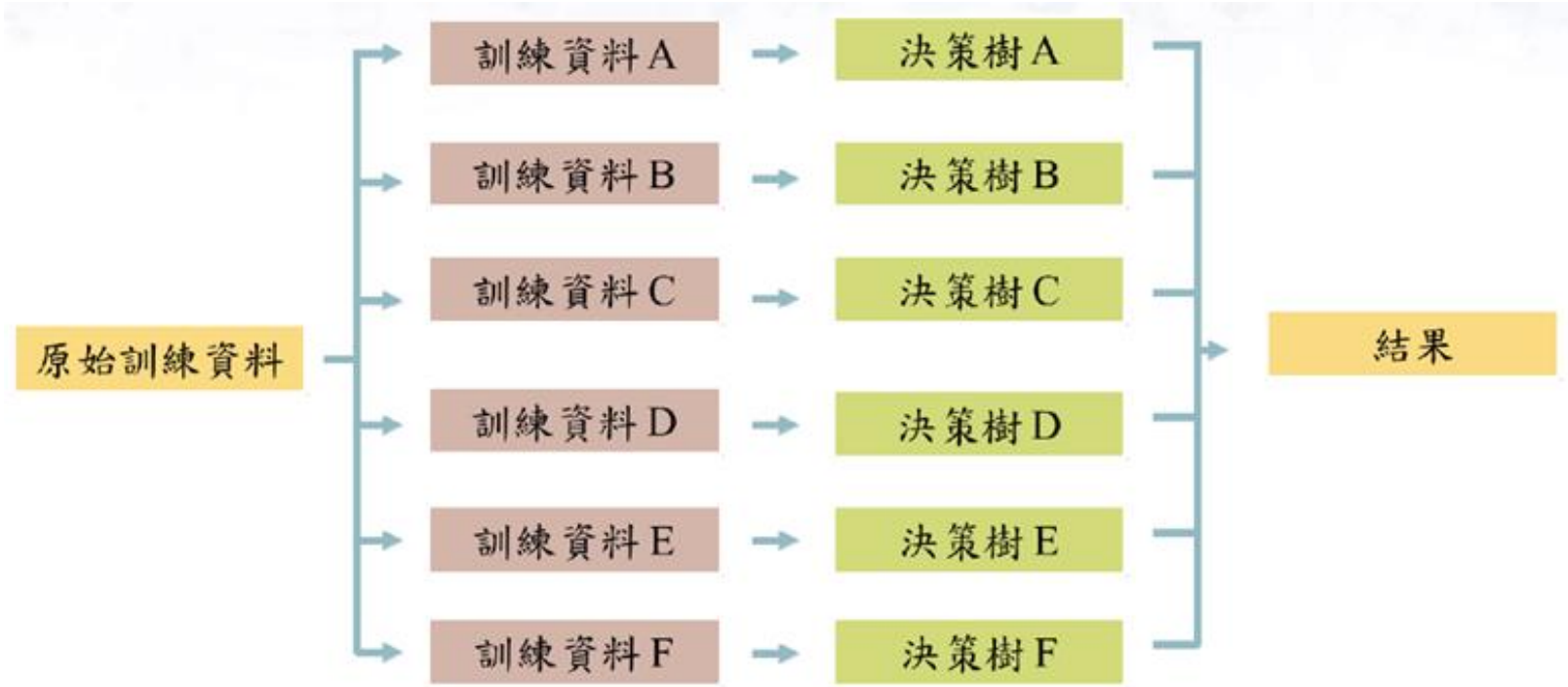


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

鳶尾花的花瓣長寬和花萼長寬

- Entropy:使用每種特徵分類後的資訊量，資訊量越多就越優先作為決策條件
- Gini Impurity:指的是分類器的分錯機率，越小代表錯誤機率越小故選為決策的條件。

隨機森林樹（Random Forest）



隨機森林的概念

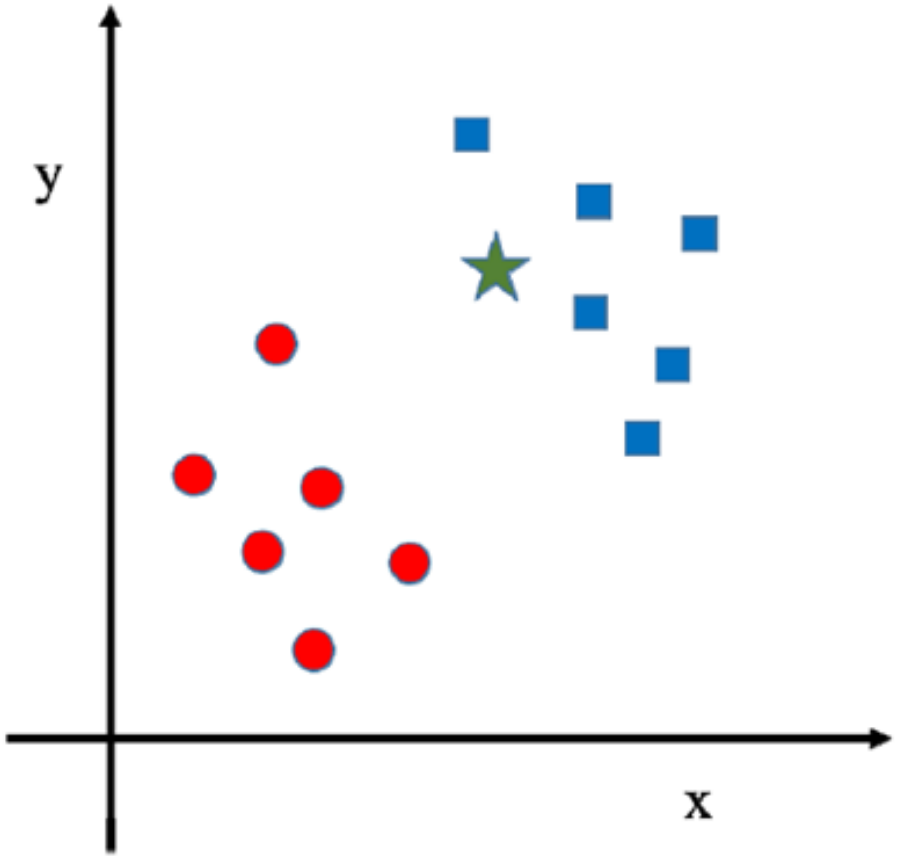
隨機森林樹（Random Forest）

範例練習

- 接續上一節決策樹的範例，一樣使用鳶尾花的資料集。
- 此處的隨機森林使用RandomForestClassifier()建構隨機森林訓練模型，n_estimators = 10代表使用10個決策樹形成隨機森林。

K-最近鄰居法（K-NN）

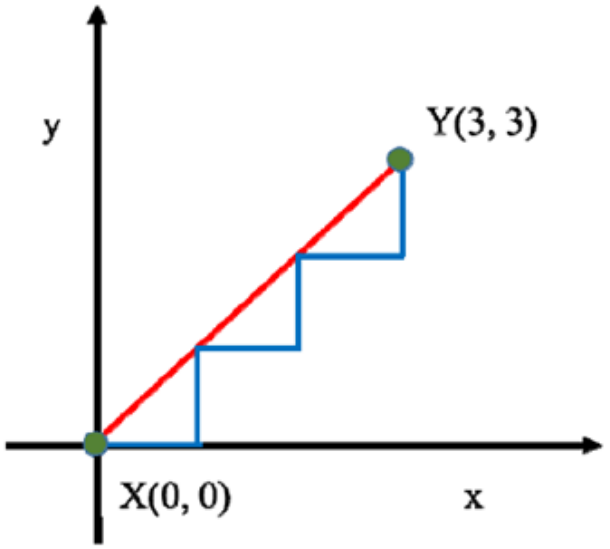
- k-Nearest Neighbor
- 時間複雜度為$O(n^2)$



最近鄰居法的概念

K-最近鄰居法（K-NN）

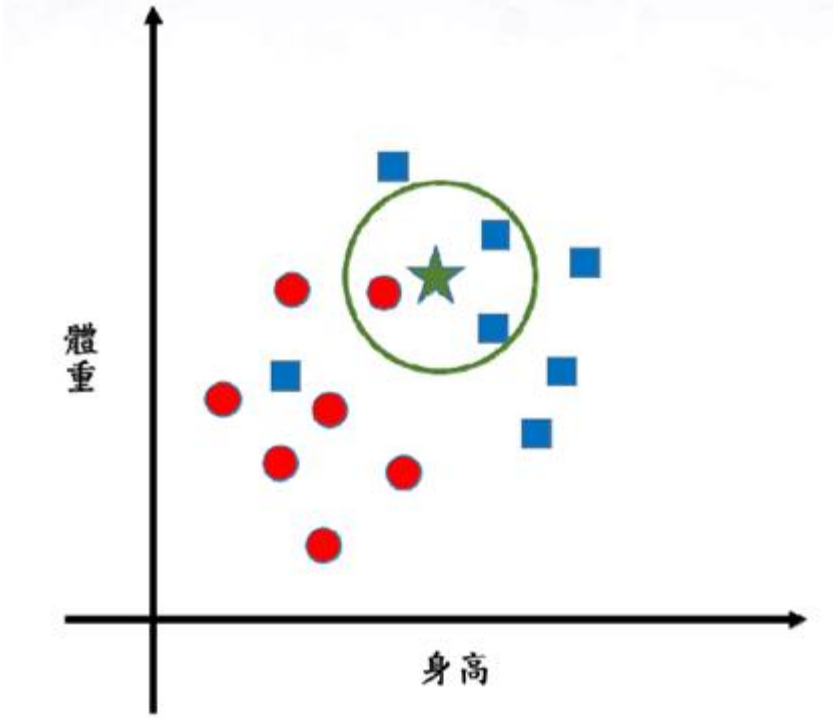
- 距離計算方法
  - 歐幾里得距離： $X(x_1, y_1)$和 $Y(x_2, y_2)$，$D(X,Y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
  - 曼哈頓距離： $X(x_1, y_1)$和 $Y(x_2, y_2)$，$D(X,Y) = |x_1 - x_2| + |y_1 - y_2|$
  - 餘弦距離： $X(x_1, y_1)$和 $Y(x_2, y_2)$，$1 - \cos\theta = 1 - \frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2+y_1^2}\sqrt{x_2^2+y_2^2}}$
  - 傑卡德距離： $D(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$



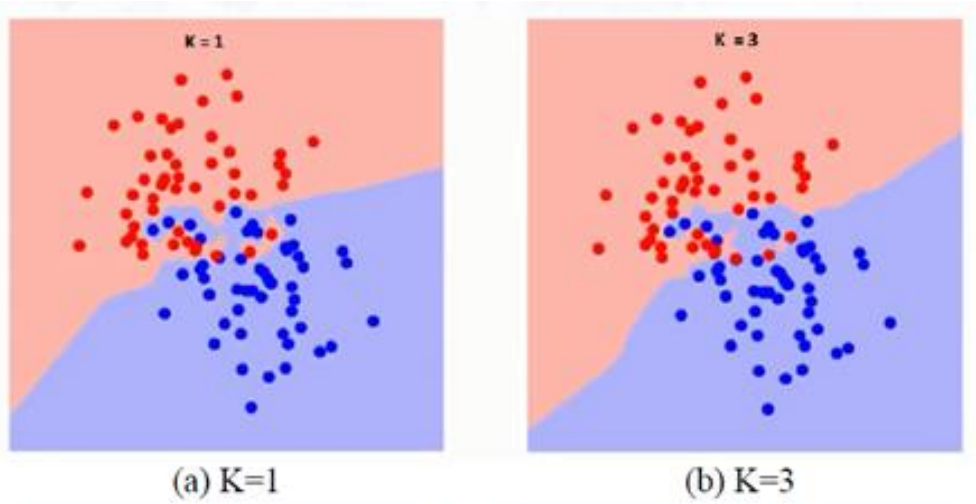
歐幾里得距離和曼哈頓距離的比較

K-最近鄰居法（K-NN）

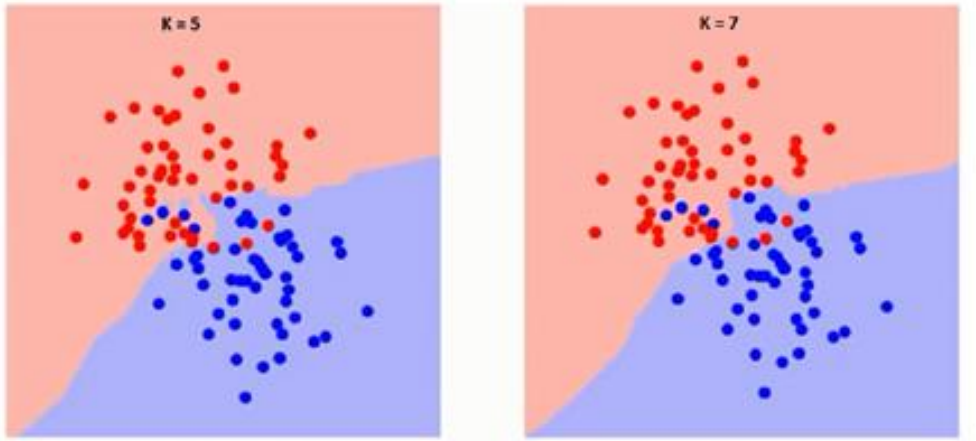
- K值的重要性



分辨男性或女性的例子



(a) K=1　(b) K=3



(c) K=5　(d) K=7

K 值對於邊界的影響

K-平均分群（K-means）

- K-means 分群 (K-means Clustering)，其實就有點像是以前學數學時，找重心的概念。

1. 先決定要分k組，並隨機選k個點做群集中心。
2. 將每一個點分類到離自己最近的群集中心(可用直線距離)。
3. 重新計算各組的群集中心(常用平均值)直到不再變化分群結束。
4. 繼續執行2、3步驟

- 時間複雜度O(NKT) ， N 是數據數量， K 是群集數量， T 是重複次數

K-平均分群（K-means）

1. 先決定要分k組，並隨機選k個點做群集中心。

(a)資料集各點座標

點	座標
A	(0, 0)
B	(1, 1)
C	(2, 1)
D	(1, 3)
E	(2, 4)
F	(3, 3)

(b)各點到群中心 B 和 C 的距離

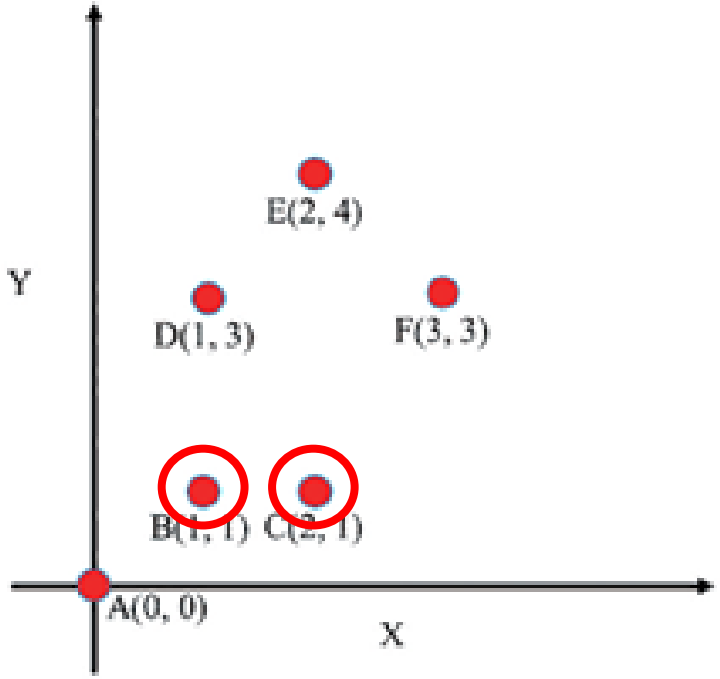
	B	C
A	1.4	2.2
D	2	2.2
E	3.2	3
F	2.8	2.2

(c)各點到群中心(0.7, 1.3)和(2.3, 2.7)的距離

	(0.7, 1.3)	(2.3, 2.7)
A	1.5	3.5
B	0.4	2.1
C	1.3	1.7
D	1.7	1.3
E	3	1.3
F	2.9	0.8

(d)各點到群中心(1, 0.7)和(2, 3.3)的距離

	(1, 0.7)	(2, 3.3)
A	1.2	3.9
B	0.3	0.5
C	1	2.3
D	2.3	1
E	3.6	0.7
F	3	1



(a) K-平均分群法的範例

K-平均分群（K-means）

2. 將每一個點分類到離自己最近的群集中心(可用直線距離)。
3. 重新計算各組的群集中心(常用平均值)直到不再變化分群結束。

第一群((0+1+1)/3,(0+1+3)/3)=(0.7 ， 1.3)

第二群((2+2+3)/3,(1+4+3)/3)=(2.3 ， 2.7)

(a)資料集各點座標

點	座標
A	(0, 0)
B	(1, 1)
C	(2, 1)
D	(1, 3)
E	(2, 4)
F	(3, 3)

(b)各點到群中心 B 和 C 的距離

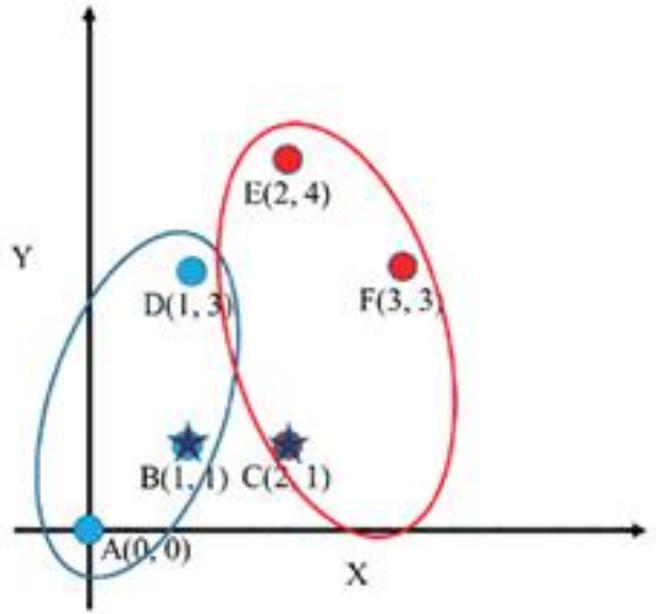
	B	C
A	1.4	2.2
D	2	2.2
E	3.2	3
F	2.8	2.2

(c)各點到群中心(0.7, 1.3)和(2.3, 2.7)的距離

	(0.7, 1.3)	(2.3, 2.7)
A	1.5	3.5
B	0.4	2.1
C	1.3	1.7
D	1.7	1.3
E	3	1.3
F	2.9	0.8

(d)各點到群中心(1, 0.7)和(2, 3.3)的距離

	(1, 0.7)	(2, 3.3)
A	1.2	3.9
B	0.3	0.5
C	1	2.3
D	2.3	1
E	3.6	0.7
F	3	1



(b)第一回合分群結果

K-平均分群（K-means）

3. 重新計算各組的群集中心(常用平均值)直到不再變化分群結束。

(a)資料集各點座標

點	座標
A	(0, 0)
B	(1, 1)
C	(2, 1)
D	(1, 3)
E	(2, 4)
F	(3, 3)

(b)各點到群中心 B 和 C 的距離

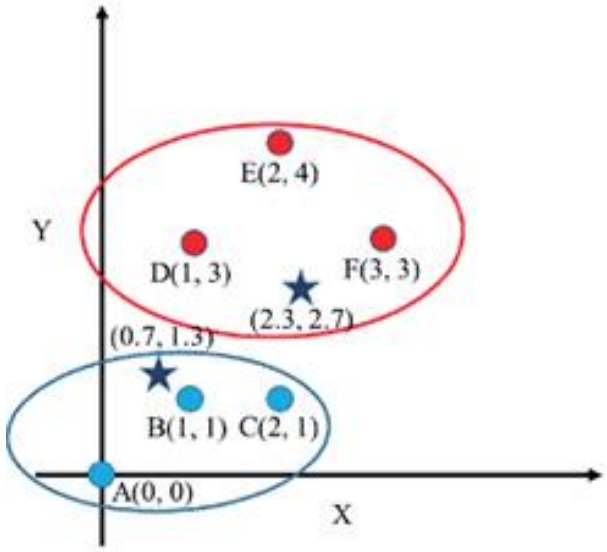
	B	C
A	1.4	2.2
D	2	2.2
E	3.2	3
F	2.8	2.2

(c)各點到群中心(0.7, 1.3)和(2.3, 2.7)的距離

	(0.7, 1.3)	(2.3, 2.7)
A	1.5	3.5
B	0.4	2.1
C	1.3	1.7
D	1.7	1.3
E	3	1.3
F	2.9	0.8

(d)各點到群中心(1, 0.7)和(2, 3.3)的距離

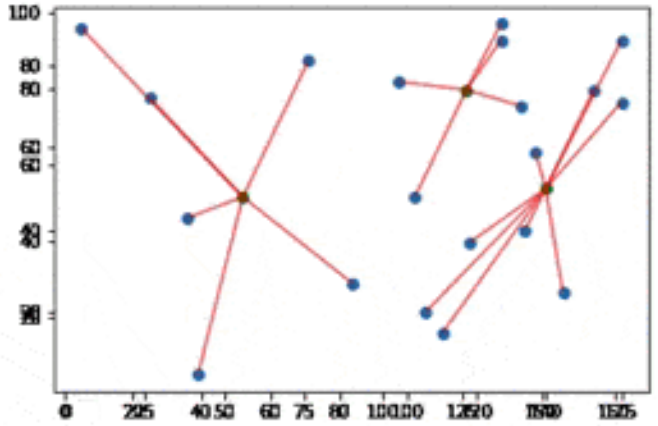
	(1, 0.7)	(2, 3.3)
A	1.2	3.9
B	0.3	0.5
C	1	2.3
D	2.3	1
E	3.6	0.7
F	3	1



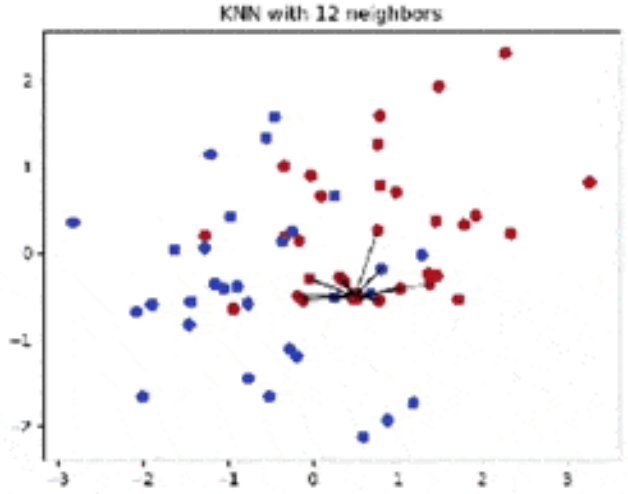
(c)第二回合分群結果

K-means VS K-NN

K-means:　K代表設定集群的類別中心點數量
K-NN:　　K是設定鄰居的數量採多數決作為輸出的依據



K-means



KNN

K-means VS K-NN

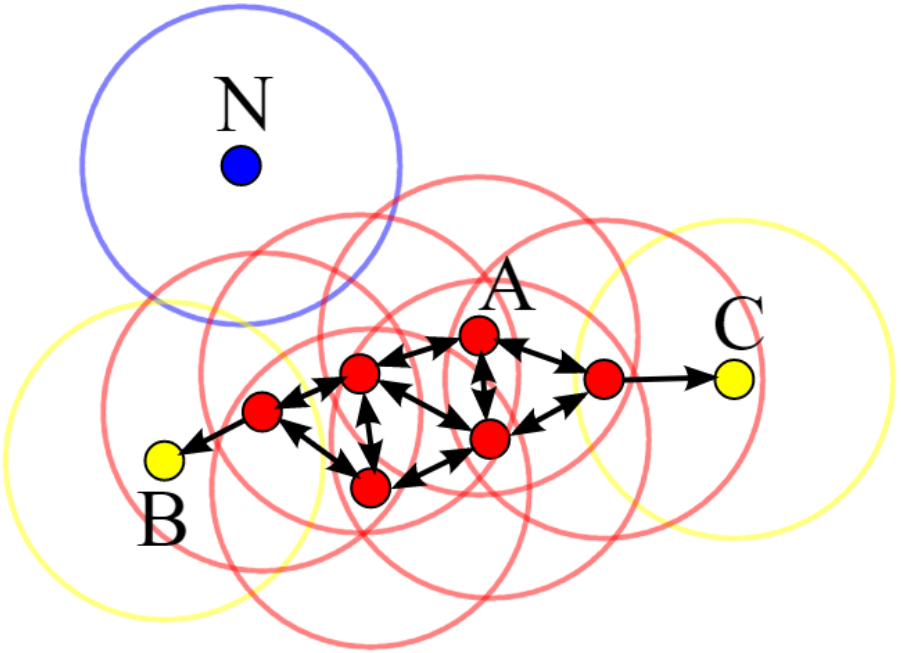
範例練習

- 以下的範例是使用make_blobs()產生300個亂數點的資料集，分成三群，每群的標準差為1。

DBSCAN

- Density-based spatial clustering of applications with noise

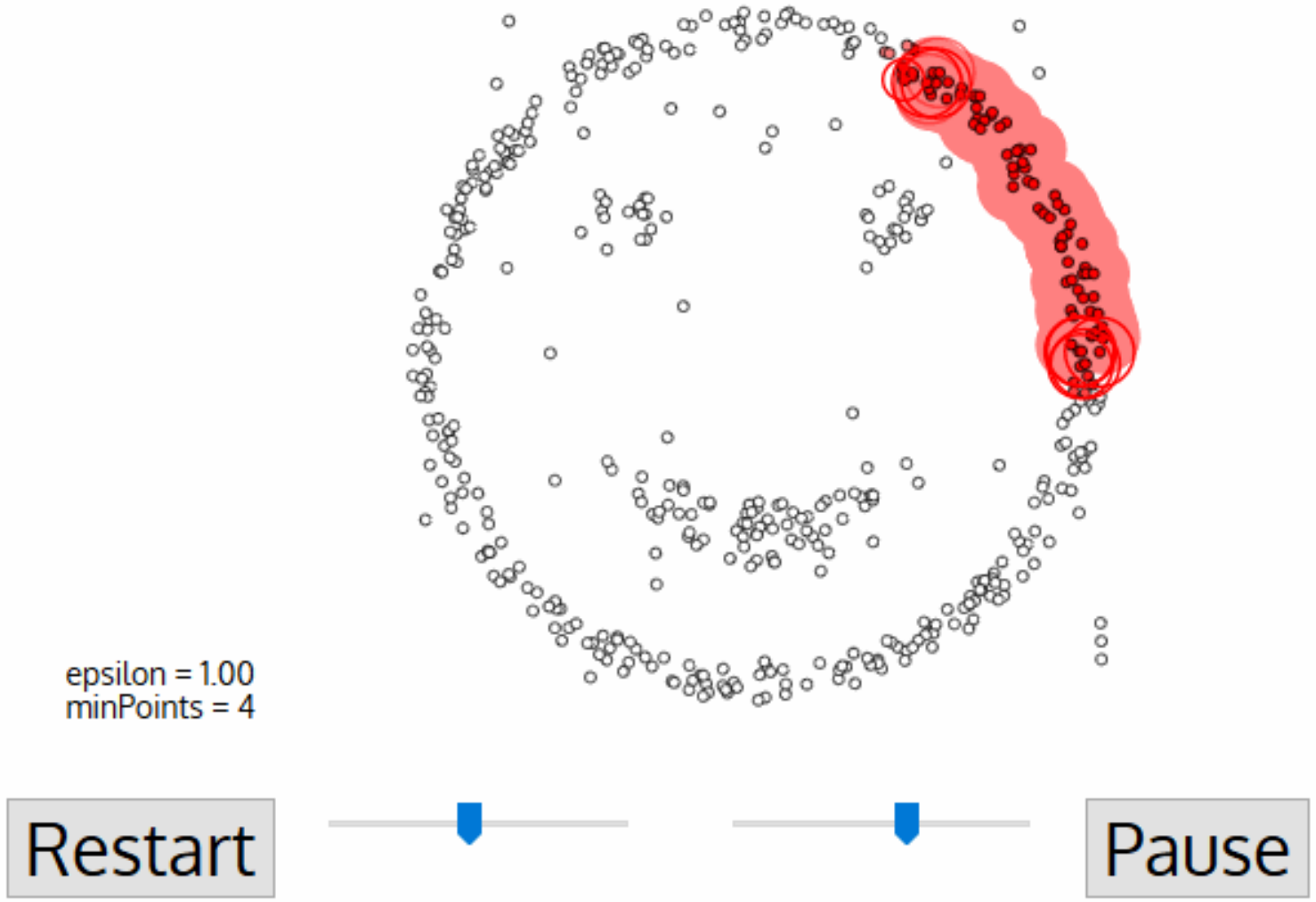
1. DBSCAN是所謂Density-Based的方法，也就是他最重視的是data的密度
2. DBSCAN能夠自動處理noise
- DBSCAN會依據data性質自行決定最終Cluster的數量



- eps: neighborhood radius
- min_samples: 4

- A: Core
- B, C: not core
- N: noise

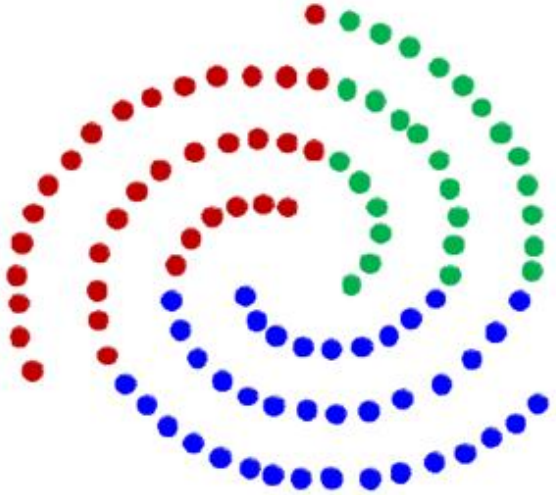
DBSCAN



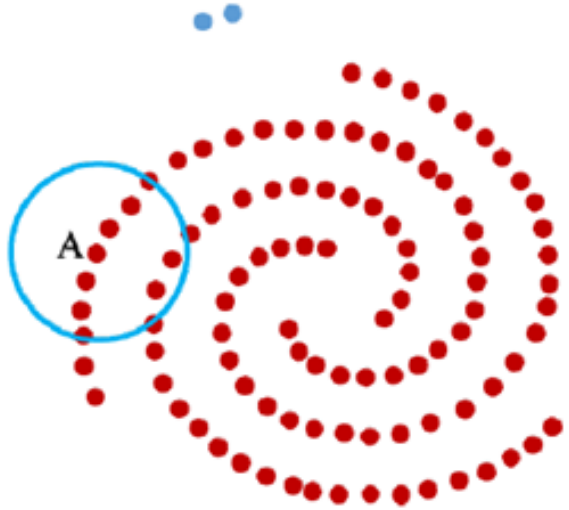
https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

DBSCAN

(a)待分群的資料分布



(b) K-平均分群的分群結果



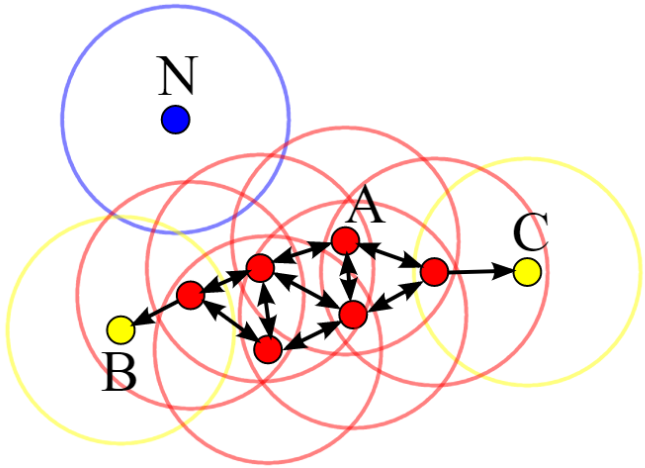
(g)半徑距離 ε 的大小影響分群的結果

使用 K-平均分群法

DBSCAN-參數設置

- eps: neighborhood radius
- Min_sample

1. 依照noise的數量調整
2. 依據Domain Knowledge
3. 依據k_distance
4. 持續做Data Analysis



- eps: neighborhood radius
- min_samples: 4

- A: Core
- B, C: not core
- N: noise

DBSCAN

範例練習

- 此處的資料集產生是以原點為同心圓，在圓周上使用自訂的函數CreatePointsInCircle(r, n=100)產生資料集，其中r代表同心圓的半徑，n代表產生資料集的數量。
- 首先算出在半徑r的圓周上，平均分成n個點，第i點的座標為(cos(2π/n*i)*r, sin(2π/n*i)*r)，為了產生亂數的效果，所以每個點x和y座標分別加上(-30, 30)間的亂數