監督式學習(Supervised Learning)

由外界提供大量資料，並提供資料對應結果，讓電腦知道資料的相關性，進而從其相關性來探討是否具有因果關係。換句話說，電腦會接收每一筆具有「標準答案」的資料，當有新資料要判斷時，就會參考先前資料產生的分布特性，來達生判別與預測。

「特徵值」是資料間的差異，每筆資料都具有各自的特徵，我們可以藉由特徵值的差異性與相似性將資料加以分類。

「分類(Classification)」是電腦藉由分析每一筆資料的特徵值，加以整理建構出一套區分物件的分類器(Classifier)，分類器能夠用來判斷新資料是屬於哪一個分類。

「最短距離分類器」是將要訓練的資料數值化後，讓每個資料皆能展現於n維座標中(n大於等於1)。每筆資料在各個類別中，將其座標取算數平均，求得類別中心點所在座標。若有一筆新座標加入進來，我們可以對新座標與類別中心座標作「歐幾里得距離」，求得與各個類別中心座標的距離。只要與類別中心座標距離越小，那就能將其歸類在此類別。

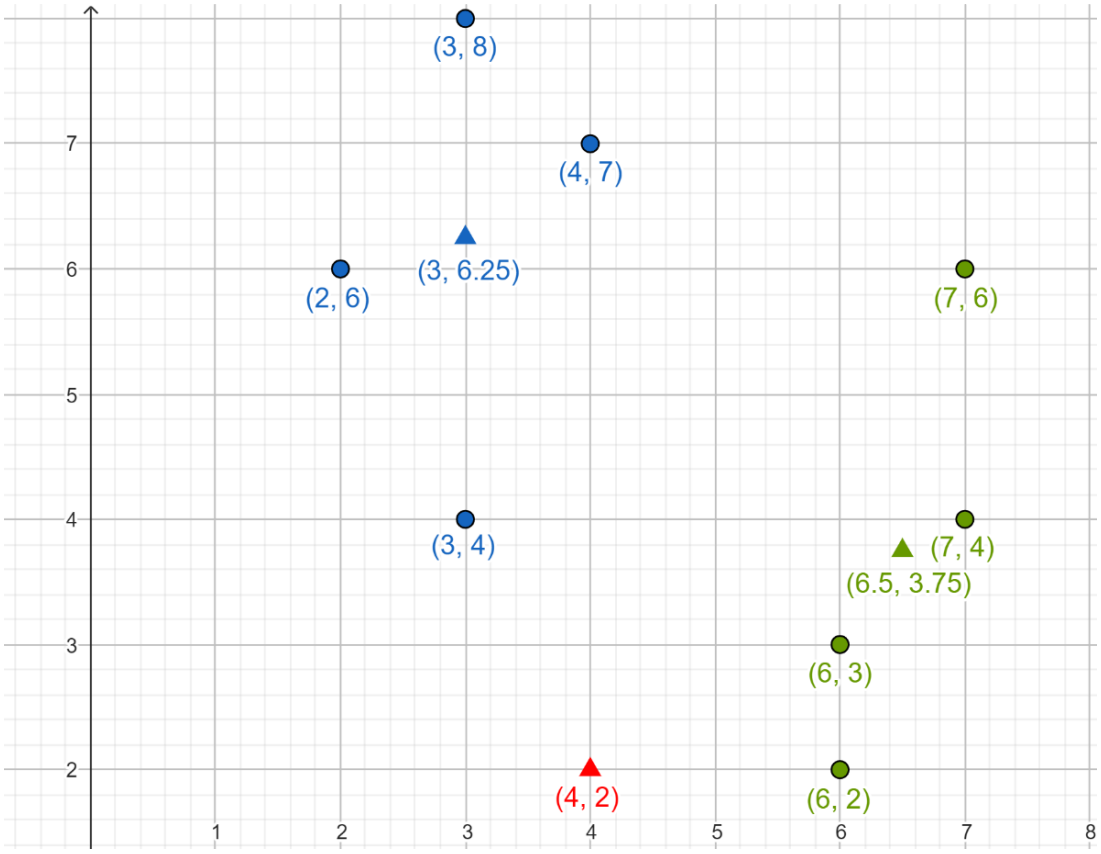
毆幾里得距離

$$d(x,y) = \sqrt{(x_1+y_1)^2+(x_2+y_2)^2 \ +.....+(x_n+y_n)^2}$$

已知有藍色和綠色兩種類別，其資料已數值化成二維座標。分別為:
藍:(*2,6*)、(*3,4*)、(*3,8*)、(*4,7*) 藍色的類別中心座標(*3,6.25*)
綠:(*6,2*)、(*6,3*)、(*7,4*)、(*7,6*) 綠色的類別中心座標(*6.5,3.75*)
如果加入一筆新資料(*4,2*)點，其類別判斷要先使用歐幾里得距離計算兩者距離。與藍色的類別中心座標距離為4.37，與綠色的類別中心座標距離為3.05。新資料座標點與綠色的類別中心座標較近，因此將新資料分類為綠色。

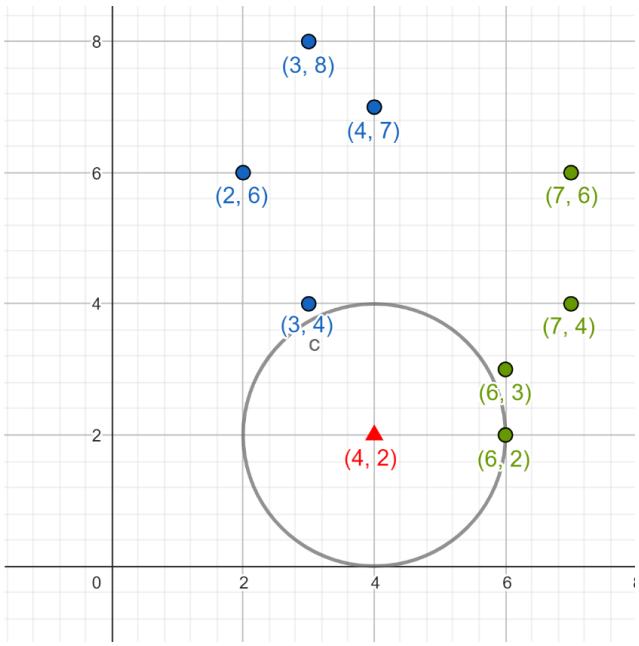
類別	X	Y	類別	X	Y
藍	2	6	綠	6	2
藍	3	4	綠	6	3
藍	3	8	綠	7	4
藍	4	7	綠	7	6
平均	3	6.25	平均	6.5	3.75

圖： 藍類別與綠類別二維座標數值



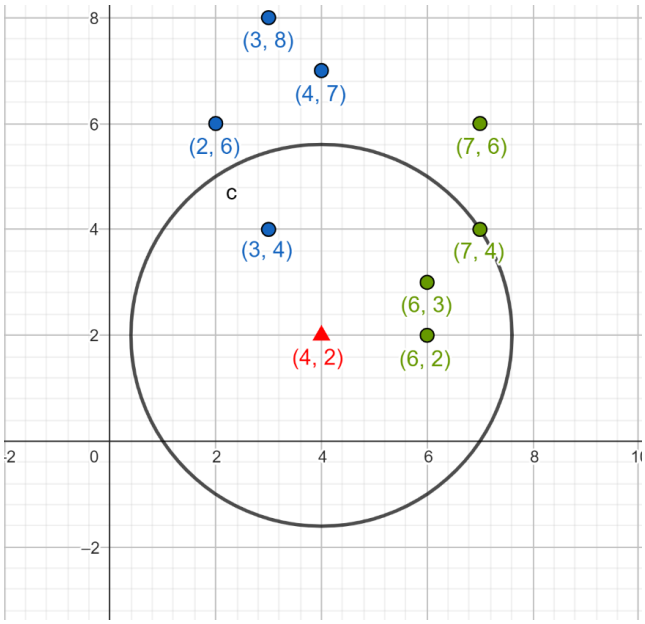
圖： 所有座標的二維座標分

「K-近鄰分群法(K-Nearest Neighbor, KNN)」會找出與該筆資料最近的k筆資料來進行分類，當k個資料中某個類別數量較多時，新資料就會被分到該類別。



圖：K值設定為1的座標圖

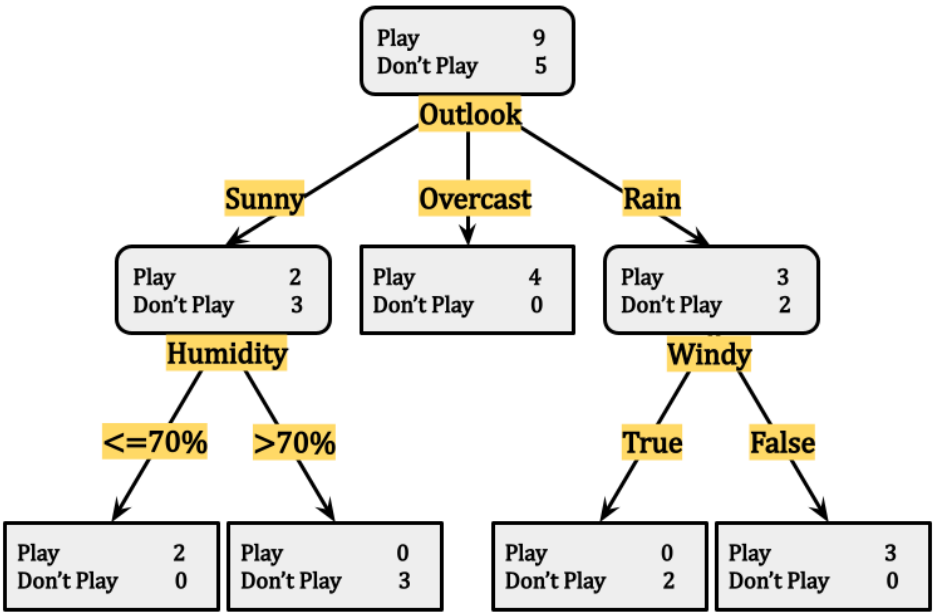
將K值設定為1，也就是要找到離新資料(4,2)點最近的一個座標。由上圖可知距離最近點為綠類別，因此新資料將會被分類到綠類別。



圖：K值設定為4的座標圖

將K值設定為4，也就是要找到離新資料(4,2)點最近的四個座標。由上圖可知距離最近點且數量最多為綠類別，因此新資料將會被分類到綠類別。

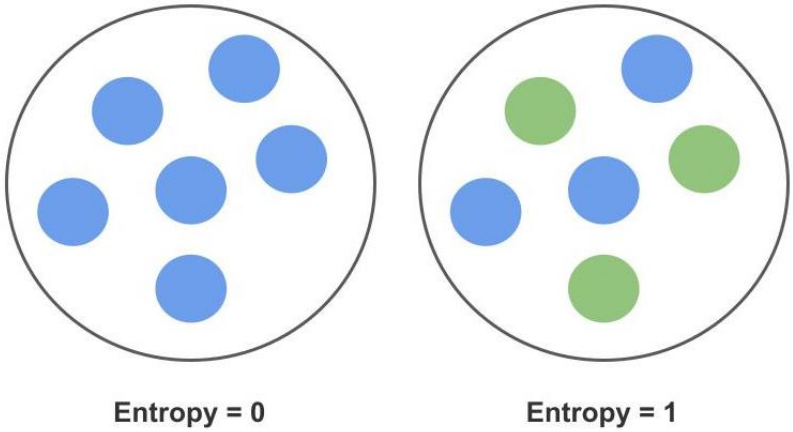
「決策樹(Decision tree)」是用來處理分類問題的樹狀結構，通常採用由上而下的方式，將整群資料從某個特徵開始展開成數個子群，直到所有子群資料都是同一個類別。



圖：決策樹概念圖

- 每個內部節點表示一個評估欄位
- 每個分枝代表一個可能的欄位輸出結果
- 每個樹葉節點代表不同分類的類別標記

為了知到哪一個特徵值可以建構最佳的決策樹，我們需要評斷特徵值的好壞，提供電腦選擇時的優先順序，以減少判斷條件的次數，而評斷特徵值的好壞是依據「熵(Entropy)」。熵可以呈現出資料的混亂程度，亂度越高代表資料越不一致。相反的，亂度越低代表資料越一致



圖：熵概念圖

熵(Entropy)

$$H(T) = -\sum_{i=1}^{n} P(t_i) log_2 P(t_i)$$

- $H(T)$：某項特徵值的熵值
  - 數值皆為 0 到 1 之間
  - 資料編碼大多數都是二進制，所以取對數時要以 2 為底數
- n：特徵值的總數
- p_i：某個類別在此特徵值出現的機率

上述公式只適用於單一特徵的亂度，假如要計算所有子集合資料亂度需要使用以下公式。

$$H(T|a) = \sum_{v \in vals(a)} \frac{|S_a(v)|}{|T|} \times H(S_a(v))$$

- $H(T|a)$：T 資料集經過 a 特徵分類之後所求的亂度
- $H(S_a(v))$：經由 a 特徵的 v 特徵值所求出的亂度
- $\frac{|S_a(v)|}{|T|}$：子資料集數占 T 資料集總數的比率(權重)

資訊獲利(Information Gain)

$$IG(T,a) = H(T) - H(T|a)$$

- $IG(T,a)$: 未經由 a 特徵分類的亂度減去資訊獲利
  - 原始亂度減去經由 a 特徵的 v 特徵值所求出的亂度，稱為「資訊獲利」
  - 值越大，資料越一致，比較優秀的分類特徵
  - 值越小，資料越不一致，比較差的分類特徵
- $H(T)$: 原始亂度
- $H(T|a)$: T 資料集經過 a 特徵分類之後所求的亂度

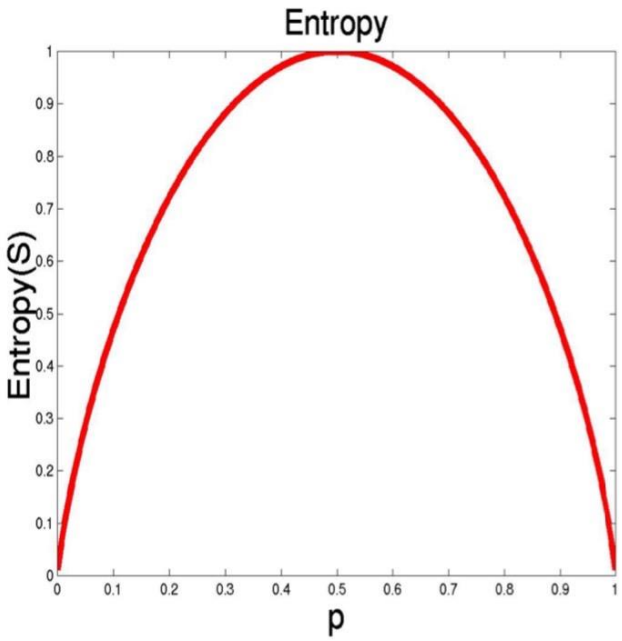


圖: 熵與出現機率分布圖

Play Bottle Cap Baseball

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	Don't Play
Sunny	80	90	True	Don't Play
Sunny	72	95	False	Don't Play
Sunny	69	70	False	Play
Sunny	71	70	True	Play
Overcast	83	78	False	Play
Overcast	64	65	True	Play
Overcast	72	90	True	Play
Overcast	81	75	False	Play
Rain	70	96	False	Play
Rain	68	80	False	Play
Rain	65	70	True	Don't Play
Rain	75	80	False	Play
Rain	71	80	True	Don't Play

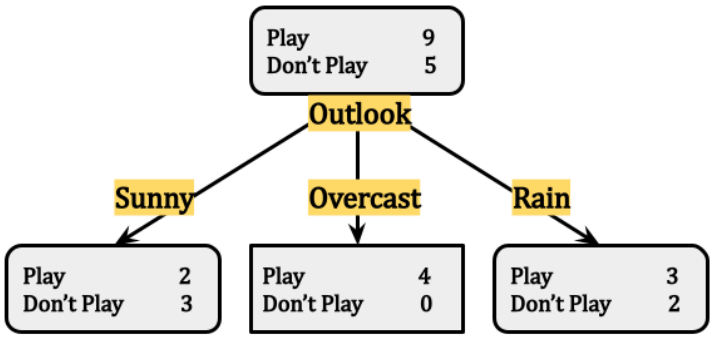
圖：建立決策樹範例資訊

原始亂度

$$H(T) = -\sum_{i=1}^{n} P(t_i) log_2 P(t_i)$$

- $H(T) = -\frac{9}{9+5} log \frac{9}{9+5} - \frac{5}{9+5} log \frac{5}{9+5} = 0.940$

第一分類依據(Outlook)



$$H(T) = -\sum_{i=1}^{n} P(t_i) log_2 P(t_i)$$

- $H(Sunny) = -\frac{2}{2+3} log \frac{2}{2+3} - \frac{3}{2+3} log \frac{3}{2+3} = 0.971$
- $H(Overcast) = -\frac{4}{4+0} log \frac{4}{4+0} - \frac{0}{4+0} log \frac{0}{4+0} = 0$
- $H(Rain) = -\frac{3}{3+2} log \frac{3}{3+2} - \frac{2}{3+2} log \frac{2}{3+2} = 0.971$

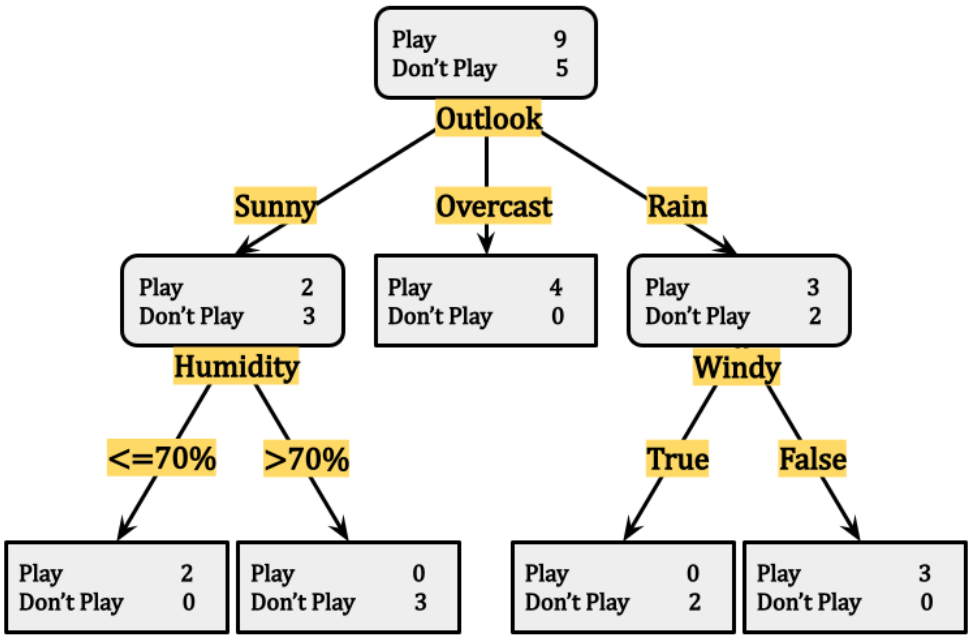
$$H(T|a) = \sum_{v \in vals(a)} \frac{|S_a(v)|}{|T|} \times H(S_a(v))$$

- $H(T|Outlook) = \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.694$

$$IG(T,a) = H(T) - H(T|a)$$

- $IG(T|Outlook) = 0.940 - 0.694 = 0.246$

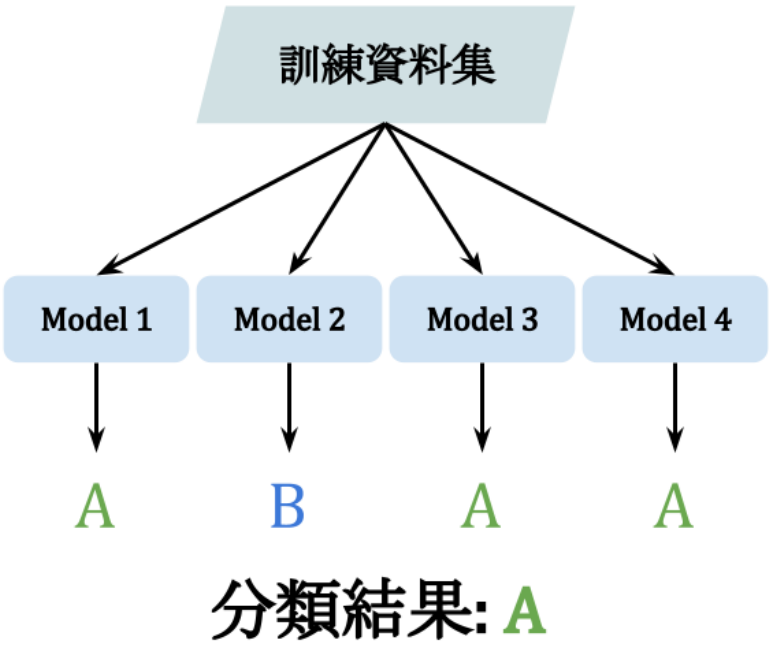
最後分類結果



根據上述計算模型，可以針對 Humidity 和 Windy 求得資訊獲利(省略計算過程)。我們能夠發現兩者的資訊獲利值會最大，代表兩個特徵會將資料分類成一致，最後求得到上圖中的最佳決策樹分類。

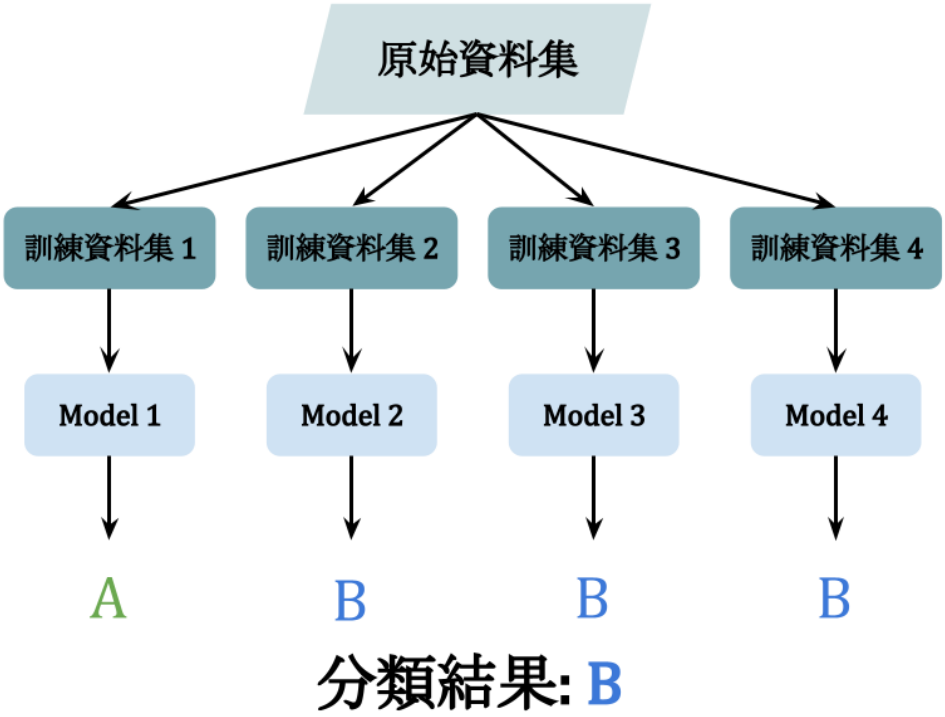
隨機森林(Random Forset)是非監督式學習的一種分類方式，是由多個分類器產生出多個分類結果，並整合所有結果求出「眾數」，此眾數就是分類的最後結果。「隨機」是由原始的資料中，隨機挑選某些資料作為訓練用資料集;「森林」是利用隨機挑選的訓練資料集所建構出的所有分類器總稱。

集成學習(Ensemble Learning)是透過多個獨立學習的分類器模型來解決單一預測問題。當輸入新資料集時，可以藉由多個分類器模型產生的預測結果，整合成單一結果，其結果優於僅靠單一分類器的訓練方式。然而挑選重複的資料集來訓練模型時，會導致分類結果趨於一致性，反而失去了集成學習的優點。因此要採取「重採樣估計」的方式，才能確保每個分類器模型之間具有差異性。



圖： 集成學習概念圖

自助採樣(Bootstrap Sampling)是一種隨機取樣的概念，由原始的資料集中，挑選固定個數的資料來作為訓練用資料集，但是在每次挑選完過後，必須將被挑選的資料集放回原始資料集中。自助採樣能夠讓部分資料被重複選取，而讓部分資料從未被選取。如此一來，就不會因為每個資料集無任何交集，導致最後分類結果差異性極大。



圖： 自助採樣概念圖