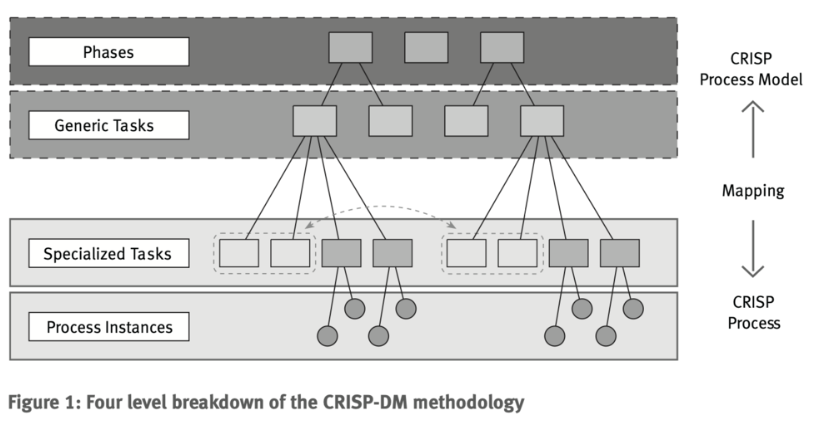
# CRISP-DM

Cross-industry standard process for data mining，如果直接翻譯的話，是為資料探勘跨領域作業流程。

是一套針對管理資料科學專案所設計的專案管理工具，用以推動與解決資料科學與商業上的問題。

**CRISP-DM將問題分成四個抽象的層次**

* Phases：表示不同的資料探勘情境
* Generic Tasks：表示在個別的情境底下，所有的資料分析解決方案
* Specialized Tasks：資料的處理方法，例如清理空值等
* Process Instances：記錄所有採取的行動、決策、結果



**Business Understanding**

這是整個專案最初的階段，目的在於分析商業目標與需求，並**將商業問題轉化為資料分析的問題**，擬定專案計畫。

在這份專案計畫當中，會包含以下的元素：

* Business background 商業背景：基本的商業概覽、要解決什麼樣的商業問題、專案的涵蓋範圍、需要使用什麼技術來解決
* Business objectives and success criteria：商業的目標與評估目標成功的方式
* Inventory of resource 專案所需資源：包括人、資料來源、分析設備、其他會被專案使用到的資源
* Requirements, assumption, constrains 專案需求、假設、限制：在專案上的需求、商業情境的假設、專案的限制
* Risk and contingencies 風險與替代方案：可能會在專案中發生的問題、對其專案的影響、有什麼行動可以去避免
* Terminology 專業術語：專案中可能混淆的專業術語
* Costs and benefits 成本與效益：專案的成本、預期效益
* Data mining goals and success criteria 資料分析的目標與評估基準：哪些資料分析的目標可以幫助商業上的成功，並列出資料分析可以達成的事情
* Project plan 專案規劃：包括專案進行的方式、時程、資源需求、跟專案有關的事情、可能的話也列出跟模型相關的作業
* Initial assessment of tools and techniques 專案會使用的技術與工具：包括分析工具、分析技術、與兩者對應到需要解決的需求

**Data Understanding**

這個階段會需要**產生許多不同的報告，來描述資料探勘的結果**，這些報告會包含：

* Initial data collection report 資料如何取得的報告：描述資料源與取得資料的方式
* Data description report 資料描述的報告：針對每份資料本身的細節進行描述，包含資料表、資料庫...等
* Data exploration report 資料探勘的報告：描述資料分析的方法、資料的模式、資料探勘過程中的發現...等
* Data quality report 資料品質報告：針對資料的品質、完整性、精準度等資訊進行描述

**Data Preparation**

到了第三階段，在資料的準備上，需要去**進行資料的前處理**，並在此階段撰寫**資料集處理前後的模樣，與交付處理後的資料**，這部分的細節會包含：

* 資料前處理的方法
* 資料處理完的結果與細節
* 在資料前處理過程中發現的新資訊
* 總結對資料的描述

**Modeling**

在資料處理完畢之後，就可以針對**模型進行開發與訓練**。

在這個階段，會需要產出的資訊如下：

* Modeling assumption 模型訓練的假設：針對模型訓練的需求，所做出的假設訊息
* Test design 設計測試方式：模型如何拆分訓練資料與測試資料、模型如何測試、讓該領域的專業人員測試模型的方式、測試計畫
* Model description 模型描述：描述要交付的模型與模型中的細節，包含開發過程中發現的資訊、模型訓練的參數、該模型與資料分析目標之間的關係、總結模型訓練與開發
* Model assessment 模型結果：描述模型測試結果、準確度、專業人員所提供的建議、哪些技巧可以提升模型的準確度

**Evaluation**

這個階段的目標，在於**評估資料分析的目標是否有影響到商業上的目標**。

此階段應該要包含到的資訊有：

* 檢視商業目標與對應的評估方式：包含每個資料分析目標對應到的商業目標、描述達成商業目標的可行性與使用資料分析技術的合適性
* 回顧整體專案：描述專案是否有達成商業目標、與後續針對其他資料分析專案的建議
* 哪些重要的因子會影響到專案的成功

**Deployment**

這部分應該算是專案的最後一個階段了，針對前面的開發內容與發現的資訊，進行最後的總結。

到最後一個階段，專案會產生的產出如下：

* Deployment plan：資料分析專案的部署計畫
* Monitoring and maintenance plan：針對部署出去的服務，如何做後續的監管與維運
* Final report：總結整個專案，包含商業分析、資料分析結果、部署與維運管理計畫、成本效益分析、總結這次整體的專案、未來的行動...等

# Decision tree

**使用不同的方式來評估分枝的好壞(亂度)**

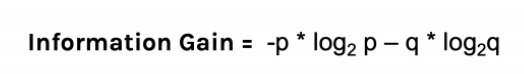
常見的資訊量評估方法有兩種：

* Information gain (資訊獲利)

主要是計算熵(Entropy)，因此經由決策樹分割後的**資訊量要越小越好**

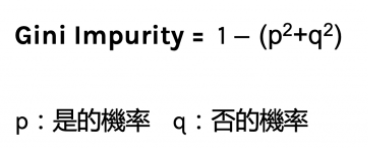
當所有的資料都被分類一致的時候 Entropy 即為 0，當資料各有一半不同時 Entropy 即為 1。

p 代表是的機率、q 為代表否的機率



* Gain ratio (吉尼獲利)

Gini 不純度是另一種亂度的衡量方式，它的**數字越大代表序列中的資料越混亂，0表示被完美的分類**。公式如下所示**，其中 p 代表是的機率、q 為代表否的機率**。當所有的資料都被分類一致的時候混亂程度即為 0，當資料各有一半不同時混亂程度即為 0.5



**決策數優缺點:**

* 優點:

1. 簡單且高度可解釋性
2. 時間複雜度低
3. 每個決策階段都相當清楚
4. 幾乎沒有要調整的超參數

* 缺點:

1. 模型容易過度擬和(overfitting)
2. 當標籤類別種類多時，樹會很複雜

**產生過度擬合數據問題的原因有哪些**

* 原因1：樣本問題

（1）樣本的noice數據干擾過大，大到模型過分記住了噪音特徵，反而忽略了真實的輸入輸出間的關係

（2）樣本抽取錯誤，包括（但不限於）樣本數量太少，抽樣方法錯誤，抽樣時沒有足夠正確考慮業務場景或業務特點，等等導致抽出的樣本數據不能有效足夠代表業務邏輯或業務場景；

（3）建模時使用了樣本中太多無關的輸入變量。

* 原因2：構建決策樹的方法問題

在決策樹模型搭建中，我們使用的算法對於決策樹的生長沒有合理的限制和修剪的話，決策樹的自由生長有可能每片葉子裏只包含單純的事件數據或非事件數據，可以想象，這種決策樹當然可以完美匹配（擬合）訓練數據，但是一旦應用到新的業務真實數據時，效果是一塌糊塗。

**如何解決過度擬合數據問題的發生**

合理、有效地抽樣，用相對能夠反映業務邏輯的訓練集去產生決策樹

針對原因2的解決方法（主要）：

剪枝：提前停止樹的增長或者對已經生成的樹按照一定的規則進行後剪枝。

(1)先剪枝（prepruning）：通過提前停止樹的構建而對樹“剪枝”，一旦停止，節點就成爲樹葉。該樹葉可以持有子集元組中最頻繁的類；

※限制決策樹的高度和葉子結點處樣本的數目

1.**定義一個高度**，當決策樹達到該高度時就可以停止決策樹的生長，這是一種最爲簡單的方法；

2.達到某個結點的實例具有相同的特徵向量，即使這些實例不屬於同一類，也可以停止決策樹的生長。這種方法對於處理數據中的數據衝突問題非常有效

3.定義一個閾值，當達到某個結點的實例個數小於該閾值時就可以停止決策樹的生長

(2)後剪枝（postpruning）：它首先構造完整的決策樹，允許樹過度擬合訓練資料，然後對那些confidence不夠的節點子樹用葉子節點來代替。

# CBR(Case-Based Reasoning)

**CBR system**：利用案例資料庫中的案例去對一個新問題的解答做出推論，相較於一般機器學習和資訊檢索的系統，它會在修正推論的過程中更新資料庫裡案例的資料

儲存代表性描述（元組或案例）—— 而不是歐幾里德空間中的點

**CBR cycle 分成相似的四個步驟：**

1) Retrieve：**根據索引找出與問題相似的案件。**

這是通過將當前問題的屬性和特徵與存儲的案例進行比較來完成的。

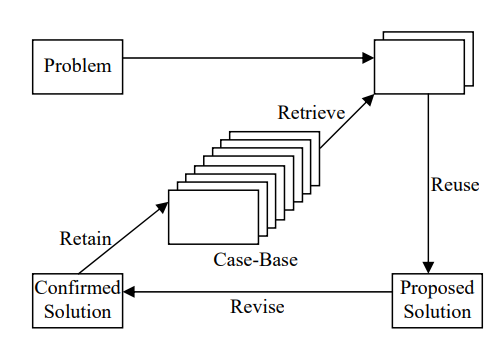
2) Reuse：**以案例解決問題，根據兩者的不同之處從案例的舊答案中推論出適合問題的新答案**。

一旦相似案例被檢索出來，系統會根據這些案例中的解決方案或行動來解決當前的問題。這個步驟涉及到將以前的解決方案適應於當前問題，並進行必要的調整。

3) Revise：**對新答案進行測試，驗證是否真能解決問題。**

在重用步驟後，系統會對解決方案進行評估，並進行必要的修訂。這可能涉及到調整解決方案的某些方面以適應當前問題的特定要求。

4) Retain：**在解決了當前問題之後，系統將該問題與相應的解決方案儲存在資料庫中**，以備將來使用。系統的資料庫會不斷增長，可以處理更多的問題。



**CBR的優點**

CBR的優點之一是它可以利用以往的經驗來解決新問題，強調從經驗中學習。它適用於那些難以用傳統基於規則或基於模型的方法進行建模和求解的問題，也允許系統利用過去的解決方案並使其適應新情況來處理複雜和動態的問題CBR在許多領域都有應用，包括醫療診斷、故障排除、智能推薦系統等。

**方法**

* 由豐富的代表描述來表示實例（例如，函數圖）
* 搜索相似案例，可以合併多個檢索到的案例
* 案例檢索、基於知識的推理和問題解決之間的緊密耦合

**挑戰**

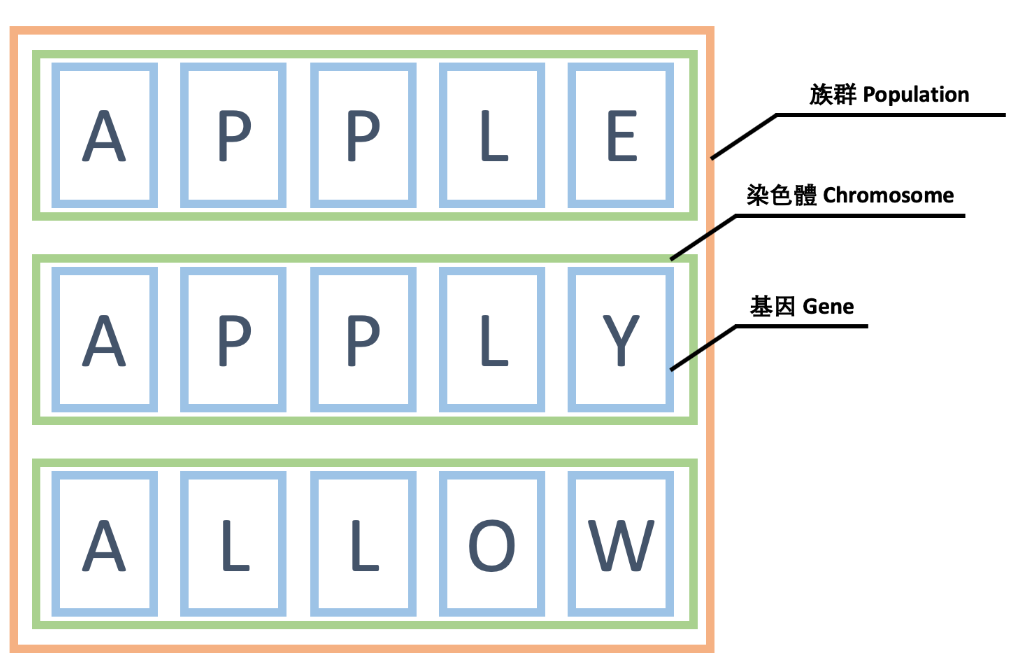
* 找到一個好的相似性度量
* 基於句法相似性度量的索引，以及失敗時、回溯和適應其他情況

# GA(Genetic Algorithmns)

**甚麼是基因演算法**

「基因演算法」(GA, Genetic Algorithm)，又叫作「遺傳演算法」，為所有演化式演算法(EA, Evolutionary Algorithms)著名的理論，也是應用最為廣泛的演化式演算法。基因演算法是一種最佳化空間搜尋法，主要用在解決一個最佳化問題的演算法，相當適合用來解決最佳化的問題，目前GA已被廣泛應用在求解最佳化的問題、資料搜尋、人工智慧及機器學習等領域上。

**基因演算法組成的元素**



* 最基本的元素 － 基因 (Gene)

在GA的過程中，要變化每一個個體中的基因片段，基因就是最基本的編碼單位

有點像是「APPLE」這個個體是由「A」、 「P」、 「P」、 「L」、 「E」這五個基因所組成的感覺，而可以根據要解決的類型來決定基因的型態，不同的基因型態就可以表示成不同的基因演算法

* 基因的集合 － 染色體 (Chromosome)

基因的集合，以上面的例子來說「APPLE」這個染色體是由「A」、 「P」、 「P」、 「L」、 「E」這五個基因所組成

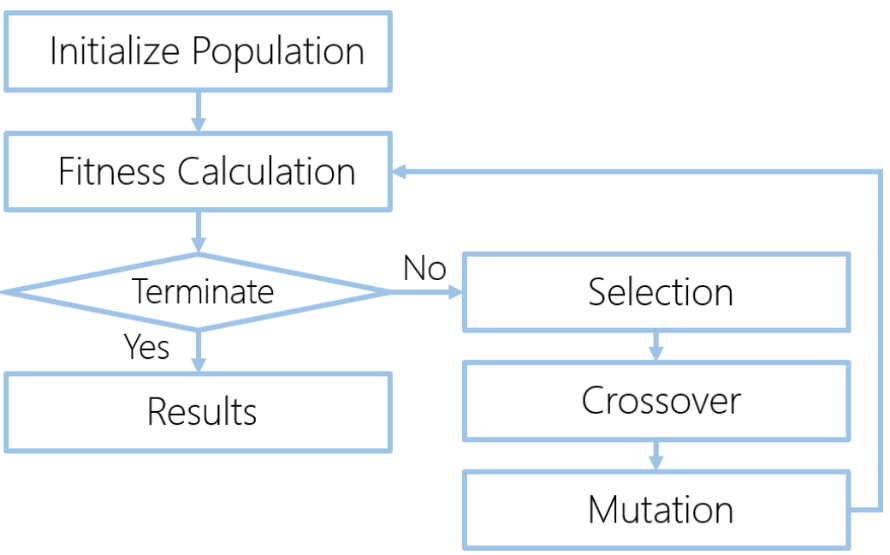
* 染色的集合 － 族群(Population)

族群就是前面染色體的集合

當每一次的世代交替中，全部的染色體就是候選人，而這些候選解救是所謂的族群啦，用上面的例子繼續延續，「APPLE」 、「GRAPE」 、「GUAVA」 、「LEMON」這些染色體合起來就是所謂的族群

**基因演算法的流程**

基因演算法的整體架構和運作的流程可以產生高度的解答搜尋能力，也具備較高的問題獨立性。在基因演算法的執行過程中以達到適應函數標準的適應值(Fitness Value)為判斷標準，而適應值的好壞同時決定了該個體是否保留或是淘汰，進而確保下一代的個體優於上一代。



* 初始化群體 ( Initialize Population )

一開始需要確定群體的大小( Population Size )。染色體有多少

例如我要透過50個染色體運算，所以我該次的群體大小就是50。

* 適應函數 ( Fitness Function)

適應函數用來計算染色體好壞的標準，用來篩選出適應程度較佳的個體(染色體)。

適應值( Fitness Value )**就是用來顯示其適應程度的數值表現，而不同的適應函數在不同的應用情境下將對應到不同的計算方式**。

設定適應值，**當該組染色體的適應值達到目標適應值時，該組染色體即為欲尋找的解**。

* 選擇 ( Selection )

選擇的精神為【留下好的基因序列】，使得好的基因流傳下去。

**根據不同的染色體所對應到的適應值 ( Fitness Value )，當適應值越高，該群體則有越高的機率被選取，以確保好的基因序列將被保留。**

白話一點說，就想像是我們要做基因改造，會把比較好的基因給留下來，不好的基因給去除這樣的概念。

例如:基因改造的人，把可能得到愛滋病的基因拿掉，就有點這樣的概念，把有疑慮不好的基因去除，保留好的。

* 交配( Crossover )

如同名稱所述，**將兩兩基因進行交配進而產生出新的染色體。而兩兩染色體是否交配也會透過交配機率進行控制。**

交配機率:發生交配的機率，並不是每次都會發生，而是達到交配機率才會進行交配

交配的方式也有很多種，單點、多點等等的方式

* 突變 ( Mutation )

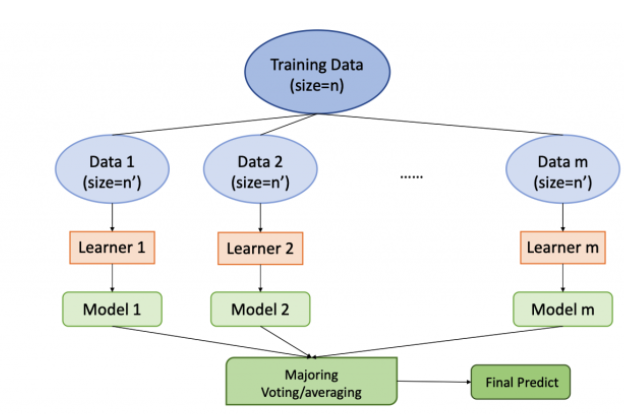
突變會在選中的chromosome中對某些基因進行變化，以二進位來說0->1，1 -> 0。

突變也是會根據突變機率(突變率會設定在0.005~0.01之間)來改變，**當該染色體的突變率低於所設定的突變率時則會產生突變**，有了Crossover之後，一定有人會有疑問為什麼還要突變，這是因為避免最後的結果陷入區域最佳解的狀況。

# Ensemble Learning

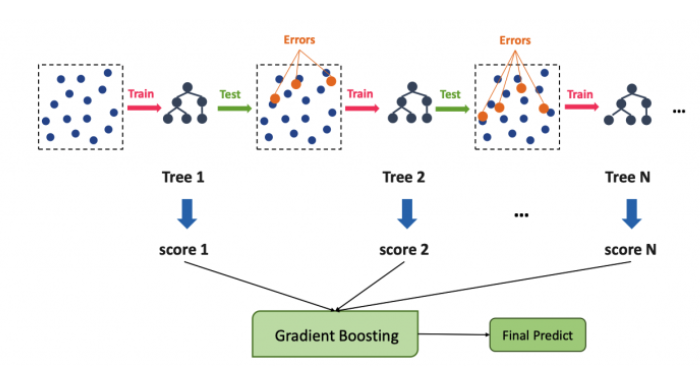
**Bagging:**

Bagging 指的是我們把訓練資料重新採樣產生不同組的訓練資料，如下圖為整體學習 Bagging 之架構。根據不同組的訓練資料即使我們用同一種演算法我們也會得到不一樣的模型，他的樹是各自獨立因此可以平行化處理。代表的方法是隨機森林，隨機森林除了 Bagging 之外，還有另一個隨機的因素是每一棵樹都只能看到一部分的特徵，這些特徵是由隨機決定的。



**Boosting:**

Boosting 則會根據每一筆訓練資料的難或簡單給予不同的權重。如下圖所示，首先我們會訓練一個 base learner 然後根據 base learner 預測的結果對或錯來分辨該筆資料是一個簡單還是困難的資料。對於難的資料我們加強他的權重再訓練一個新的分類器或回歸器。我們目標是希望訓練後，新的模型在這些難的資料能夠表現得更好。我們不斷重複這些步驟，不斷地加入新的 base learner，且新的 base learner 把過去表現不好的地方改善，這就是 Boosting 精神。因此 Boosting 的每一棵樹是互相有關聯性的做完第一棵樹可能進行下一棵樹的生成。代表的方法有 AdaBoost、Gradient Boosting，兩種都是產生非常多棵的樹，但是每一棵都是很簡單的決策樹。Boosting 目標是希望新的樹可以針對舊的樹預測不太好的部分做一些補強。所以最終我們要把這麼多簡單的樹合再一起才能當最後的預測。



**閾值移動(threshold-moving)**指根據實際情況調整劃分類別的閾值，常被用於解決類別不平衡問題。對於分類問題，可以不直接預測實例的類標號，而是通過預測概率值，後指定一個閾值來將實例化分為正類和負類 ，通常指定閾值為0.5，但可根據實際情況對閾值進行移動，進而增加某一類的權重 ，以達到解決類不平衡的目的。

# SOM

**理論:**

Kohonen 的自組織地圖以無監督模式學習

Kohonen 的算法形成”特徵圖”(feature map)，其中建構了神經元的鄰域

這些鄰域的組織方式使得拓撲上接近的神經元對模型相似的input敏感

自組織圖或自組織特徵圖，有時可用於開發對data的早期洞察

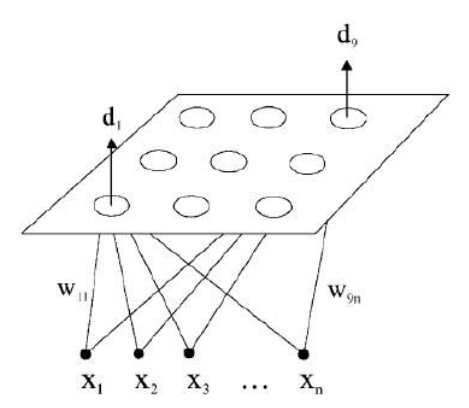
神經元格（“節點”）接受並responds一組輸入信號

比較反應； 從晶格中選出的”winner”神經元

選定的神經元與”鄰域”神經元一起激活

自適應過程(Adaptive process)改變權重以更接近輸入

SOM有一個很重要的優點為，將N維(N-dimension)的資料映射(mapping)到2維(2-dimension)的空間上，並且維持資料中的拓撲(topology)特性，將資料映射到2 維空間時則可以使用視覺化(visualization)的方式呈現，以方便後續的觀察及分析。



**舉例:**

假設我們想要了解全世界各地所有不同品種的羊之間的關係為何, 若總共有5個品種(a.亞洲, b.非洲, c.歐洲, d.美洲及e.大洋洲), 每一個品種都有5個屬性(5維). 對每一個品種收集100 隻羊的資料, 因此資料總數為500 筆. 接下來我們利用SOM 將5維的資料映射到2維的空間上, 以視覺化的方式來呈現這5個品種的羊之間的拓樸關係.

**優點:**

* SOM 是一種將高維數據投影到二維地圖上的算法。
* 投影保留了數據的拓撲結構，因此相似的數據項將被映射到地圖上附近的位置。

SOM在模式識別、語音分析、工業和醫療診斷、數據挖掘等方面仍有許多實際應用。

**缺點:**

* 需要大量高質量的代表性訓練數據
* 沒有普遍接受的 SOM”品質”衡量標準

例如：平均量化誤差（數據分類的好壞）