第四類 AI 機器學習原理

一、 機器學習概論

機器學習是人工智慧的一個分支。人工智慧的研究歷史有著一條從以「推理」為重點，到以「知識」為重點，再到以「學習」為重點的自然、清晰的脈絡。顯然，機器學習是實現人工智慧的一個途徑，即以機器學習為手段解決人工智慧中的問題。機器學習在近 30 多年已發展為一門多領域交叉學科，涉及概率論、統計學、逼近論、凸分析、計算複雜性理論等多門學科。機器學習理論主要是設計和分析一些讓電腦可以自動「學習」的演算法。機器學習演算法是一類從資料中自動分析獲得規律，並利用規律對未知資料進行預測的演算法。因為學習演算法中涉及了大量的統計學理論，機器學習與推斷統計學聯絡尤為密切，也被稱為統計學習理論。演算法設計方面，機器學習理論關注可以實現的，行之有效的學習演算法。很多推論問題屬於無程式可循難度，所以部分的機器學習研究是開發容易處理的近似演算法。

機器學習已廣泛應用於資料探勘、電腦視覺、自然語言處理、生物特徵辨識、搜尋引擎、醫學診斷、檢測信用卡欺詐、證券市場分析、DNA 序列測序、語音和手寫辨識、戰略遊戲和機器人等領域。

1.機器學習有下面幾種定義：

- 機器學習是一門人工智慧的科學，該領域的主要研究物件是人工智慧，特別是如何在經驗學習中改善具體演算法的效能。
- 機器學習是對能通過經驗自動改進的電腦演算法的研究。
- 機器學習是用資料或以往的經驗，以此最佳化電腦程式的效能標準。
- 一種經常參照的卡內基梅隆大學 Tom Mitchell（1998）定義是：

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

一個計算機程序能夠從經驗 E 中學習（學習任務是 T，學習的表現用 P 衡量），如果這個程序在任務 T 與表現衡量 P 下，可以通過經驗 E 得到改進。

用上面的國際象棋舉個例子：

- 經驗 e：就是程序是上萬次的自我練習然後的出來的經驗。
- 任務 t：就是下棋。
- 學習的表現 p：則是它與一些新的對手比賽時，贏得比賽的概率。

- E = the experience of playing many games of checkers
- T = the task of playing checkers.
- P = the probability that the program will win the next game.

[Quiz 01]

假設您的電子郵件程序監視您將哪些電子郵件標記為垃圾郵件或不將其標記為垃圾郵件，並以此為基礎來學習如何更好地過濾垃圾郵件。此設置中的任務 T 是什麼？

A.將電子郵件分類為垃圾郵件或非垃圾郵件。

B.看著您將電子郵件標記為垃圾郵件或非垃圾郵件。

C.正確分類為垃圾郵件/非垃圾郵件的電子郵件數量（或分數）。

D.以上都不是機器學習算法。

nswer: A

分析：將垃圾郵件和非垃圾郵件分類就是所要選的任務 T

觀察你的郵件是不是垃圾郵件這個就屬於經驗 E

成功識別出是垃圾郵件的數量這個就是它的性能度量值 P

2. 監督學習（Supervised learning）和無監督學習（Unsupervised learning）：

目前主要兩種學習算法類型為：監督學習（Supervised learning）和無監督學習（Unsupervised learning）。此外，諸如強化學習（Reinforcement learning）和推薦系統（Recommender systems）等名詞。

(1)監督學習（Supervised learning）：

假如你想預測一下現在的房價，這是一個數據集(Data sets)，橫軸是房子的大小，縱軸是房價。例如你有一間 100 平方公尺的房子你想知道能賣多少錢，那麼學習算法怎麼幫助你呢？它會根據數據集擬合出一個函數，讓函數盡可能匹配到所有的數據。當你輸入你房子的大小，它就會返還給你一個目前市場上比較合理的價格。這是一個監督學習的例子。

監督學習，就是給出一個算法，通過數據集可以得到正確的答案。例如上面的房價，給定房價的數據集，對於里面的每個房子大小數據，算法都知道其對應的正確房價。

這也是一種迴歸（Regression）問題，意指要預測一個連續值的輸出。我們可以把看作是一個標量值，一個連續值的數，而回歸意味著要預測這類連續值屬性的種類。

[案例]

一個胸部腫瘤的數據集，橫軸表示腫瘤的大小，縱軸表示腫瘤是否為良性的。假如非常不幸，我們的胸部長了腫瘤，對應的機器學習算法就是，根據你的這個尺寸，估算出一個概率，即腫瘤為良性腫瘤的概率或者惡性腫瘤的概率。

這也是一個分類（Classification）問題。分類就是要預測一個離散值的輸出，就是 0 or 1，也就是良性/惡性。當然在分類問題當中，有時候會超過兩個值的輸出。

[Quiz 02]

您正在經營一家公司，並且您想開發學習算法來解決兩個問題。

- 問題 1：您有大量相同的物品庫存，您想預測接下來三個月將出售多少這些物品。
- 問題 2：您希望軟件檢查單個客戶帳戶，並為每個帳戶確定是否已被黑客入侵/破壞。您應該將它們視為分類問題還是回歸問題？

A.將兩者都視為分類問題。

B.將問題 1 視為分類問題，將問題 2 視為回歸問題。

C.將問題 1 作為回歸問題，將問題 2 作為分類問題。

D.將兩者都視為回歸問題。

Answer: C

根據上面的定義：

Problem 1 你有大量相同項目的庫存，你想預測在接下來的 3 個月裡有多少產品會銷售出去，回歸問題。

Problem 2 你希望軟件檢查個人客戶帳戶，並為每個帳戶決定是否已被入侵/損害，分類問題。

[Quiz 03]

一天中的降雨量通常以毫米（mm）或英寸為單位。假設您使用學習算法來預測明天會有多少降雨。

您會將其視為分類問題還是回歸問題？

A.回歸

B.分類

Answer:A

分析：預測明天的降水量，回歸問題。

[Quiz 04]

假設您正在進行天氣預報，並且您想預測明天下午 5 點是否會下雨。您想為此使用學習算法，您會將其視為分類問題還是回歸問題？

A.回歸

B.分類

Answer:B

分析：預測明天下午是否下雨，分類問題。

[Quiz 05]

假設您正在進行天氣預報，並使用學習算法預測明天的溫度（以攝氏度/華氏度為單位），您會將其視為分類問題還是回歸問題？

A.回歸

B.分類

Answer:A

分析：預測明天的天氣溫度，回歸問題。

[Quiz 06]

假設您正在進行股票市場預測，通常每天交易（即購買/出售）數千萬股 Microsoft 股票。您想預測明天將要交易的微軟股票的數量。您將其視為分類還是回歸問題？

A.分類

B.回歸

Answer:B

分析：預測明天 Microsoft shares 的人數，回歸問題。

[Quiz 07]

假設您正在進行股票市場預測，並且您想預測明天某隻股票的價格（以美元為單位）。您想為此使用學習算法，您會將其視為分類問題還是回歸問題？

A.分類

B.回歸

Answer:B

分析：預測明天股票的價格，回歸問題。

[結論]

- 在有監督的學習中，我們得到一個數據集，並且已經知道我們正確的輸出應該是什麼樣子，認為輸入和輸出之間有一個關係。有監督的學習問題分為“迴歸”和“分類”問題。
- 在回歸問題中，我們試圖在連續輸出中預測結果，這意味著試圖將輸入變量映射到一些連續函數。
- 在分類問題中，我們試圖用離散輸出來預測結果。換句話說，我們正試圖將輸入變量映射成離散的類別。

(2) 無監督學習（Unsupervised Learning）：

在上面的監督學習中，在其數據集裡面，每一個樣本都被標註為正樣本或者負樣本。對於每一個樣本，我們也清楚知道什麼才是正確的答案。

在無監督學習中，我們沒有屬性或者標籤這個概念了。也就是所有的數據都是一樣的，沒有什麼區別。所以在無監督學習中，我們只有一個數據集，沒人告訴我們應該怎麼做，我們也不知道每個數據點是什麼意思。它只告訴你，這裡是一個數據集，你能在其中找到某種結構或者規律嗎？

基於給出的數據集，無監督學習算法可以給出不同的聚類(又稱「集群」, cluster)，這就是所謂的聚類算法。

[案例 1] news.google.com

谷歌新聞每天在幹嗎呢，它就是每天去收集成千上萬的網絡新聞，然後將他們分組，組成一個個的新聞專題。當我們點進去一個新聞專題的時候，就會有全球各地的媒體對於該新聞的報導。

[案例 2]

給定一組不同的個體，對於每一個不同的個體，檢測它們是否擁有某個特定的基因，圖中綠色、紅色、黑色等就是展示了這些不同的個體是否擁有一個特定的基因的不同程度。運行一個聚類算法，把不同的個體歸入不同的類，這就是無監督學習了。

因為我們沒有提前告訴我們的算法，這種基因類型是具體屬於哪一類的人，我們只是告訴算法，這裡有一堆數據，我也不知道這些數據是什麼東東，但是你要幫我自動找到這些數據中的類型哦。

[Quiz 08]

在以下示例中，您將使用無監督學習算法解決哪些問題？(複選題)

A.給出標為垃圾郵件/非垃圾郵件的電子郵件，了解垃圾郵件過濾器。

B.給出在網絡上找到的一組新聞文章，將它們分組為關於相同故事的文章集。

C.提供客戶數據數據庫，自動發現市場細分並將客戶分組到不同的市場細分中。

D.給出一個被診斷為患有或未患有糖尿病的患者的數據集，學習將新患者分類為患有或不患有糖尿病。

Answer: B, C

A.監督學習，因為你已經給出了什麼是垃圾郵件什麼不是垃圾郵件的判斷標準。

D.監督學習裡面的分類。

[結論]

- 無監督學習使我們能夠解決那些不怎麼知道結果的問題。雖然我們並不知道變量的影響，但是我們可以從數據中提取結構。可以根據數據中變量之間的關係對數據進行聚類，從而得出這種結構。
- 無監督學習沒有基於預測結果的反饋。

(➔ https://zhuanlan.zhihu.com/p/32300281)

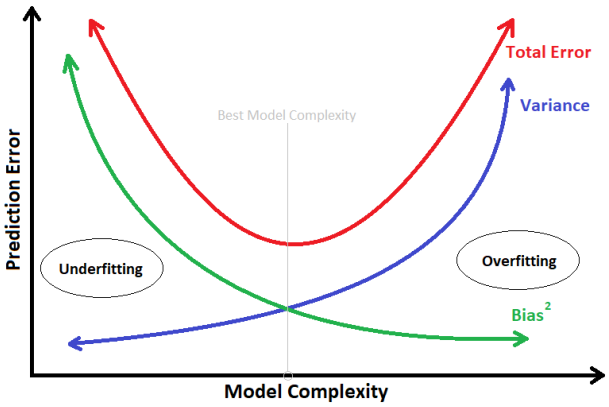
(3) 其他機器學習的分類

除了我們前面講過的大分類「Supervised」(監督式)與「Unsupervised」(非監督式)之外，也有以下幾種比較特別的類型：

- 半監督學習 (Semi-supervised learning)：半監督是學習方法，指的是拿到的訓練資料中有些有標籤，有些沒有。
- 增強學習 (reinforcement learning)：會根據機器正確或錯誤進行調整，像是答對了就給獎勵，失敗了就處罰的概念，逐步調整權重。

[考古題] 4-13, 4-19, 4-27

二、偏差與變異(方差)之權衡 Bias-Variance Tradeoff

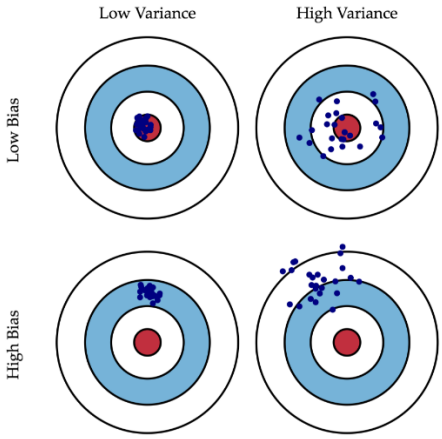


(1) 什麼是偏差(Bias)? 什麼是變異(Variance)?

其實可以套到生活中 "準" 跟 "確" 這兩個概念，如果用高中軍訓課打靶的經驗來說，那就是：

- 如果說你打靶打得很精 "準"，意味你子彈射中的地方離靶心很近，即 Low Bias(低偏差)。
- 如果說你打靶打得很精 "確"，意味你在發射數槍之後這幾槍彼此之間在靶上的距離很近，即 Low Variance(低變異)。

接著下面用一張圖來說明:



在機器學習領域我們都希望能夠把 Model 訓練到非常的 "準確" ，即你的模型是可以用來描述數據背後的真實規律、真實意義，以便後續我們用這個模型來執行一些描述性的任務或預測性的任務。一般在實作上會碰到的誤差有：

- 隨機誤差
- 偏差
- 方差

其中隨機誤差的部分源於數據本身，基本上沒有辦法消除。而偏差與方差的部分又跟 Overfitting & Underfitting 的問題緊緊綁在一起。

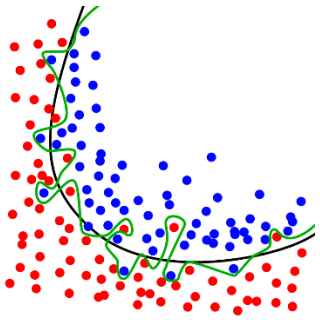
(2) Overfitting & Underfitting

- Overfitting(擬合過度)：找出來的模型受到訓練資料的影響太大，使得對預測的效果不佳。

在統計學中，過適（overfitting，或稱擬合過度）是指過於緊密或精確地匹配特定資料集，以致於無法良好地拟合其他資料或預測未來的觀察結果的現象。

- Underfitting(擬合不足)：模型對於資料的描述能力太差，無法正確解釋資料。

乏適（underfitting，或稱：擬合不足）；它是指相較於資料而言，模型參數過少或者模型結構過於簡單，以至於無法捕捉到資料中的規律的現象。發生乏適時，模型的偏差大方差小。



上圖綠線代表過適模型，黑線代表正規化模型。雖然綠線完美的符合訓練資料，但調適得太過緊密或精確；並且與黑線相比，在新的測試資料上會有更高的錯誤率。
(參考網址➔ https://zh.wikipedia.org/wiki/%E9%81%8E%E9%81%A9)

(3) Bias-Variance Tradeoff (偏差和變異之權衡)
顧名思義，就是透過權衡 Bias Error 跟 Variance Error 來使得總誤差(Total Error) 達到最小。
[考古題] 4-01, 4-02, 4-03, 4-06

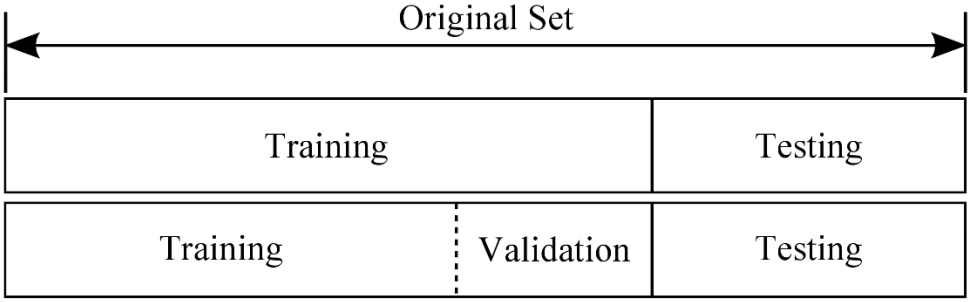
三、訓練集、驗證集、測試集

將 training set 分為 A 和 B，A 用來訓練，B 用來測試，機器學習也做了類似的劃分。
機器學習將樣本分為三種：訓練集、驗證集、測試集。

- 訓練集：即 training set，主要用來訓練得到我們的 hypothesis function，即確定普通參數
- 驗證集：即 validation set，主要用來確定超參數（大家可以暫時不用管什麼是超參數），做模型選擇；通過訓練集，我們可能得到了多個 hypothesis function，驗證集就是來確定最佳 hypothesis function
- 測試集：即 testing set，在最佳模型確定後，可以通過測試集進行模型預測並評估模型的性能

需要注意的是：當模型沒有需要人為設定的超參數時，可以不需要驗證集（validation set）。

樣本到底該怎麼做劃分？訓練集、驗證集、測試集占比重該是多少？比重不確定，可以根據自己的需求更改，一般為 8:1:1。其實有很多的劃分方
(原文網址：https://kknews.cc/news/xjmnbz8.html)
[考古題] 4-05



Ripley, B.D（1996）在他的經典專著 Pattern Recognition and Neural Networks 中給出了這三個詞的

定義:

- Training set: A set of examples used for learning, which is to fit the parameters [i.e., weights] of the classifier.
- Validation set: A set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.
- Test set: A set of examples used only to assess the performance [generalization] of a fully specified classifier.
- 訓練集

作用：估計模型

學習樣本資料集，通過匹配一些引數來建立一個分類器。建立一種分類的方式，主要是用來訓練模型的。

- 驗證集

作用：確定網路結構或者控制模型複雜程度的引數

對學習出來的模型，調整分類器的引數，如在神經網路中選擇隱藏單元數。驗證集還用來確定網路結構或者控制模型複雜程度的引數。

- 測試集

作用：檢驗最終選擇最優的模型的效能如何

主要是測試訓練好的模型的分辨能力（識別率等）

- 一個典型的劃分是訓練集佔總樣本的 50%，而其它各佔 25%，三部分都是從樣本中隨機抽取。(➔ https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/404219/)

四、**機器學習相關演算法**：

1.構造間隔理論分布：聚類分析和圖形識別

- 人工神經網路
- 決策樹
- 感知器
- 支援向量機
- 整合學習 AdaBoost
- 降維與度量學習
- 聚類
- 貝葉斯分類器

2. 構造條件機率(概率)：**迴歸分析**和**統計分類**

- 高斯過程回歸
- 線性判別分析
- 最近鄰居法
- 徑向基函式核

3. 通過再生模型構造概率密度函式：

- 最大期望演算法

- 機率圖模型：包括貝葉斯網和 Markov 隨機場
- Generative Topographic Mapping

4. 近似推斷技術：
- 馬爾可夫鏈
- 蒙特卡羅方法
- 變分法

5. 最佳化：大多數以上方法，直接或者間接使用最佳化演算法。
(參考網址: https://zh.wikipedia.org/wiki/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0)
[考古題] 4-09, 4-11, 4-12, 4-14, 4-16, 4-17,4-18,4-20

五、機器學習\統計方法: 模型評估-驗證指標(validation index)

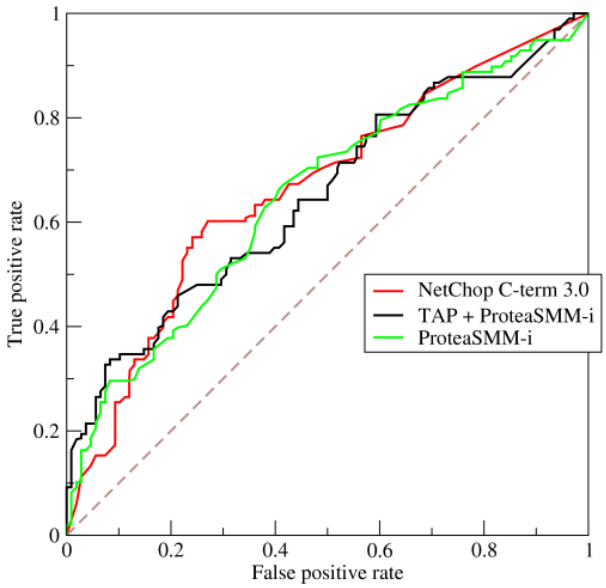
誤差是用於衡量機器學習模型的準確率，可以透過特徵工程或著是調參數來提升模型的準確率，直到你認為訓練的模型最低可以接受多少%的誤差為止。而在機器學習當中評估誤差的指標除了偏差 Bias 以外還有以下這幾種：

(1) 分類指標:
- 二元相關：二元混淆矩陣相對應驗證指標（參考(3)）
- ROC 曲線

  ROC 曲線也是用來評估二元分類問題的一個方法，可知閾值(threshold)在往左往右變化的時候代表的是，FP 和 FN 是會變化的，這時候就會有 trade off 的問題。原因是這兩個指標跟 FP 和 FN 也是直接相關，所以 FP 和 FN 變化基本上靈敏性和特異性也是跟著變，且 FP 和 FN 說的是個數，靈敏性和特異性才是百分比的指標。

  ROC 曲線怎麼來的?閾值(threshold)變化可以得到靈敏性和特異性指標，將所有可能的閾值(threshold)都去設定，然後可以跑出很多組靈敏性和特異性，一個靈敏性會對上一個特異性，因此把所有可能的連起來得到的就是 ROC 曲線了。

  ROC 曲線解讀方式，會以對角線為基準，若是算出來的 ROC 曲線等於對角線的話，代表你的模型完全沒有鑑別性(簡單說你的模型就沒啥屁用，回去重新 training)，若 ROC 往左上角移動，代表模型對疾病的敏感度越高(偽陽性率(1-Specificity)越低)，代表模型的鑑別力越好。



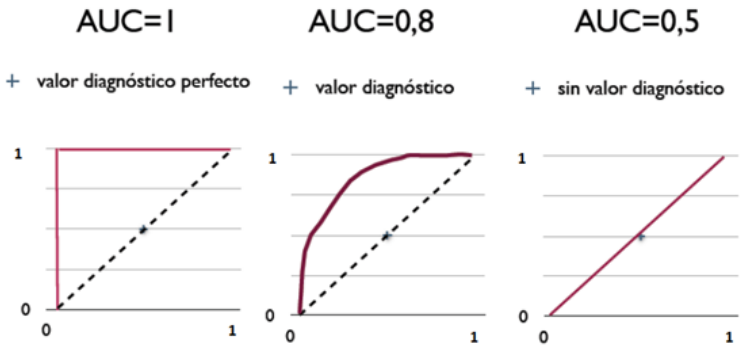
- AUC(Area Under Curv, 曲線下的面積)：

  AUC 指標是分類問題常用的指標，通常分類問題都需要定一個閾值(threshold)來決定分類的類別 (通常為機率 > 0.5 判定為 1, 機率 < 0.5 判定為 0)。AUC 是衡量曲線下的面積，因此可考量所有閾值下的準確性。

  常利用 (Area Under Curve; AUC)來判別 ROC 曲線的鑑別力，AUC 數值的範圍從 0 到 1，數值愈大愈好。以下為 AUC 數值一般的判別規則：

  - AUC=0.5 (no discrimination 無鑑別力)，ROC 剛好是對角線。
  - 0.7≦AUC≦0.8 (acceptable discrimination 可接受的鑑別力)
  - 0.8≦AUC≦0.9 (excellent discrimination 優良的鑑別力)
  - 0.9≦AUC≦1.0 (outstanding discrimination 極佳的鑑別力)

  AUC 也可以當作比較不同模型之間比較的指標，AUC 越大代表那個模型越好。



- 多元相關(多元混淆矩陣和相對應驗證指標)
- F1-Score

  分類問題中，我們有時會對某一類別的準確率特別有興趣。例如瑕疵/正常樣本分類，我們希望任何瑕疵樣本都不能被漏掉。Precision，Recall 則是針對某類別進行評估

- Precision(精準率): 模型判定瑕疵，樣本確實為瑕疵的比例
- Recall(召回率): 模型判定的瑕疵，佔樣本所有瑕疵的比例

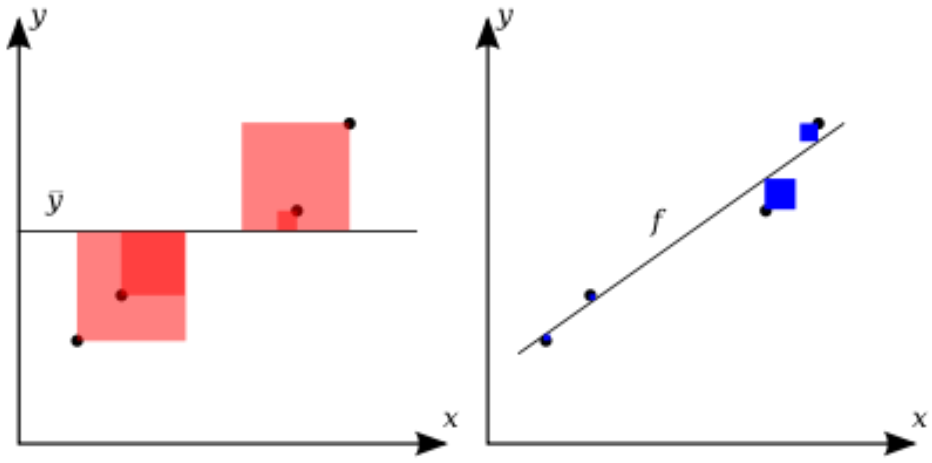
F1-Score 則是 Precision, Recall 的調和平均數

[考古題] 4-07

(2) 迴歸指標：(Ref: https://blog.csdn.net/reallocing1/article/details/56292877)

- **平均均方誤差(Mean Squared Error, MSE)**
  - 均方誤差是指參數估計值與參數真值之差平方的期望值;
  - MSE 可以評價數據的變化程度，MSE 的值越小，說明預測模型描述實驗數據具有更好的精確度。
- **均方根誤差(RMSE, Root Mean Square Error)**
  - 均方誤差:均方根誤差是均方誤差的算術平方根
  - 是觀測值與真值偏差的平方和與觀測次數 m 比值的平方根。
  - 是用來衡量觀測值同真值之間的偏差
- **平均絕對誤差(MAE, Mean Absolute Error)**
  - 平均絕對誤差是絕對誤差的平均值
  - 平均絕對誤差能更好地反映預測值誤差的實際情況.
- **平均均方對數誤差(Mean Squared Logarithmic Error, MSLE)**
- **R-square**(決定係數，英語：coefficient of determination，記為 R^2 或 r2）

  (Ref. https://zh.wikipedia.org/wiki/%E5%86%B3%E5%AE%9A%E7%B3%BB%E6%95%B0)
  在統計學中用於度量因變量的變異中可由自變量解釋部分所占的比例，以此來判斷統計模型的解釋力。



決定係數$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$ 示意圖

線性回歸（右側）的效果比起平均值（左側）越好，決定係數的值就越接近於 1。 藍色正方形表示線性回歸的殘差的平方， 紅色正方形數據表示對於平均值的殘差的平方。

[考古題] 4-08, 4-21, 4-24

(3) 二元混淆矩陣(Confusion matrix)介紹

(Ref. https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%B5%B1%E8%A8%88%E6%96%B9%E6%B3%95-%E6%A8%A1%E5%9E%8B%E8%A9%95%E4%BC%B0-%E9%A9%97%E8%AD%89%E6%8C%87%E6%A8%99-b03825ff0814)

在二元分類基本上就是分「有」和「沒有」、「真」跟「假」、「正」和「負」。
下表是二元分類的混淆矩：

		True Condition	
	Total Population (T)	**Positive**	**Negative**
Predicted outcome	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

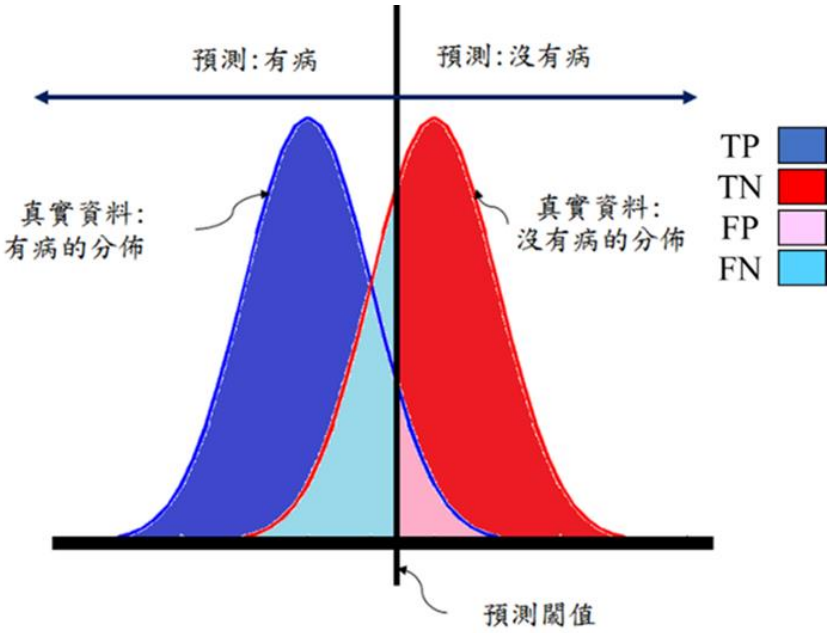
- True condition 就是你資料的答案
- Predicted outcome 就是模型預測出來的結果
- Positive 就是「有」、「真」或是「正」，在醫學上通常用「有發病」
- Negative 就是「沒有」、「假」或是「負」，在醫學上通常用「沒有發病」
- True Positive (TP)「真陽性」:真實情況是「有」，模型說「有」的個數
- True Negative(TN)「真陰性」:真實情況是「沒有」，模型說「沒有」的個數
- False Positive (FP)「偽陽性」:真實情況是「沒有」，模型說「有」的個數
- False Negative(FN)「偽陰性」:真實情況是「有」，模型說「沒有」的個數

- 常用指標

- **Sensitivity 靈敏性**: 也稱為 True Positive Rate (**TPR**), Recall，「有病的偵測率」，所以是越高越好。
- **Specificity 特異性**: 也稱為 True negative rate (**TNR**)，「沒病的偵測率」，也是越高越好。
- **Accuracy 正確率**: 基本上就是模型的整體判斷的正確率，所以有時候也稱為 overall accuracy，越高越好。
- **False Negative Rate 偽陰性率**: 預測模型判成沒病，但實際上有病的比率，越小越好。
- **False Positive Rate (FPR) 偽陽性率**: 預測模型判成有病，但實際上沒有病的比率，越小越好。
- **Positive predictive value (PPV) 陽性預測值**: 也稱為 Precision，在臨床上也是很常用的指標，模型診斷結果呈現有病且確實有病者的比率，越高越好。
- **Negative predictive value (NPV) 陰性預測值**: 模型診斷結果呈沒病且實際上也沒有病的比率，越高越好。

[考古題] 4-04

FP 和 FN 常容易混淆，下圖是用來解釋在混淆矩陣:



在統計學上會將

- FP 稱為：「型一錯誤 (Type I error)，上圖淺藍色那塊」
- FN 稱為：「型二錯誤 (Type II error)，上圖粉紅色那塊」

在統計學的假設檢定：做的事情就是利用統計方法**推測虛無假設(H0)是否成立**，也就是拒絕或是不拒絕虛無假設。

- 若是虛無假設事實上成立，但檢定結果拒絕虛無假設時，這個錯誤就稱做為**型一錯誤**
- 若是虛無假設事實上不成立，但檢定結果不拒絕虛無假設時，這個錯誤就稱做為**型二錯誤**。

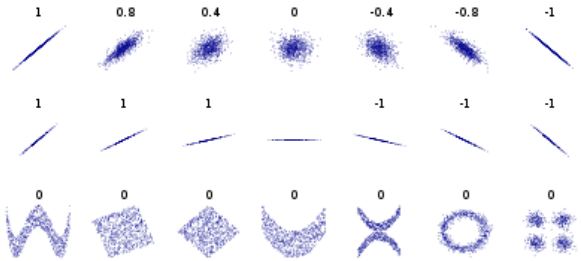
[考古題] 4-32

(4) 相關係數(Ref.

https://zh.wikipedia.org/wiki/%E7%9A%AE%E5%B0%94%E9%80%8A%E7%A7%AF%E7%9F%A9%E7%9B%B8%E5%85%B3%E7%B3%BB%E6%95%B0)

在統計學中，皮爾遜積矩相關係數（英語：Pearson product-moment correlation coefficient，又稱作 PPMCC 或 PCCs, 文章中常用 r 或 Pearson's r 表示）用於度量兩個變數 X 和 Y 之間的相關程度（線性相依），其值介於-1 與 1 之間。

[考古題] 4-23



(5) 對數損失函數(Logarithmic Loss Function)

(Ref. https://www.itread01.com/content/1529761131.html)

對數損失，即對數似然損失(Log-likelihood Loss)，也稱羅吉斯回歸損失(Logistic Loss)或交叉熵損失(cross-entropy Loss)，是在概率估計上定義的.它常用於(multi-nominal，多項)邏輯斯諦回歸和神經網絡,以及一些期望極大算法的變體．可用於評估分類器的概率輸出.

對數損失通過懲罰錯誤的分類,實現對分類器的準確度(Accuracy)的量化．最小化對數損失基本等價於最大化分類器的準確度.為了計算對數損失，分類器必須提供對輸入的所屬的每個類別的概率值，不只是最可能的類別.

[考古題] 4-22

(6) 支援向量機（SVM, Support Vector Machine）

在機器學習中，支援向量網路是在分類與迴歸分析中分析資料的監督式學習模型與相關的學習演算法。

(Ref. https://zh.wikipedia.org/wiki/%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA)

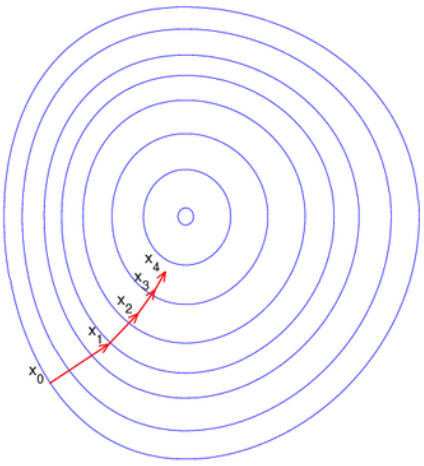
[考古題] 4-18, 4-27

(7) 梯度下降法（Gradient descent）

(Ref. https://zh.wikipedia.org/wiki/%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D%E6%B3%95)

是一個一階最佳化算法，通常也稱為最陡下降法，但是不是近似積分的最陡下降法（英語：Method of steepest descent）。

要使用梯度下降法找到一個函數的局部極小值，必須向函數上當前點對應梯度（或者是近似梯度）的反方向的規定步長距離點進行疊代搜索。如果相反地向梯度正方向疊代進行搜索，則會接近函數的局部極大值點；這個過程則被稱為梯度上升法。

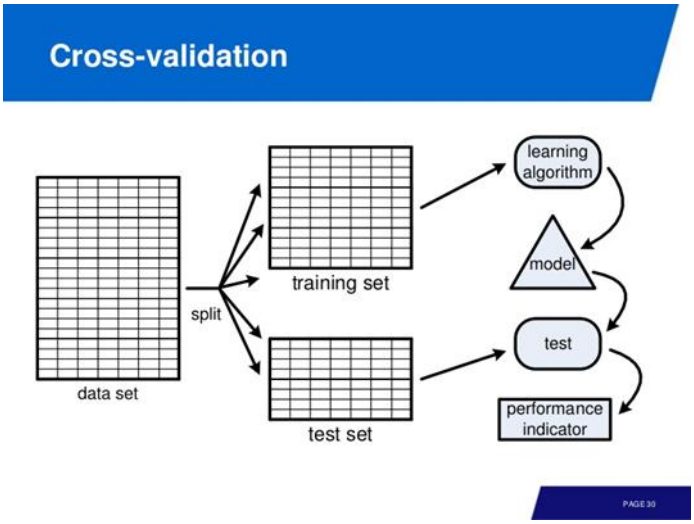


[考古題] 4-34, 4-40

(8) K 折交叉驗證(k-folder cross-vailation)

(Ref. https://ithelp.ithome.com.tw/articles/10197461)

一般來說我們會將數據分為兩個部分，一部分用來訓練，一部分用來測試，交叉驗證是一種統計學上將樣本切割成多個小子集的做測試與訓練。



A.交叉驗證主要分為以下幾類：

- **k-folder cross-vailation**
- kk folder cross-vaildation
- least-one-out cross-validation
- **10-fold corss validation**

b.為什麼需要交叉驗證

為了避免依賴某一特定的訓練和測試資料產生偏差。

which is intended to avoid the possible bias introduced by relying on any one particular division into test and train components, is to partition the original set in several different ways and to compute an average score over the different partitions.

一個更好的方式是把原始資料按不同的方法分，計算不同部分的平均得分。

誇張點來說，我們都只有用一部分特定的測試資料去測試我們訓練的結果，假設剛好那一部分測試資料剛好百分之百一樣，而其他部分剛好都不準確，我們就以為這個訓練結果是百分之百的。

c.交叉驗證怎麼做

K-Fold Cross Validation is used to validate your model through generating different combinations of the data you already have. For example, if you have 100 samples, you can train your model on the first 90, and test on the last 10. Then you could train on samples 1-80 & 90-100, and test on samples 80-90. Then repeat. This way, you get different combinations of train/test data, essentially giving you 'more' data for validation from your original data.

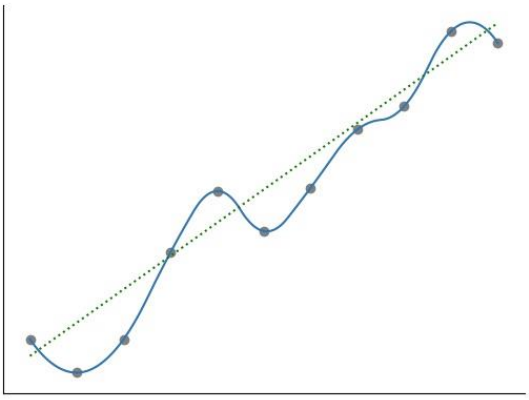
k 折交叉驗證（英語：k-fold cross-validation），將訓練集分割成 k 個子樣本，一個單獨的子樣本被保留作為驗證模型的數據，其他 k－1 個樣本用來訓練。交叉驗證重複 k 次，每個子樣本驗證一次，平均 k 次的結果或者使用其它結合方式，最終得到一個單一估測。這個方法的優

勢在於，同時重複運用隨機產生的子樣本進行訓練和驗證，每次的結果驗證一次，**10 次交叉驗證**是最常用的。
(Ref. https://zh.wikipedia.org/wiki/%E4%BA%A4%E5%8F%89%E9%A9%97%E8%AD%89)

在 k 交叉驗證中，是使用不同的資料組合來驗證你訓練的模型，舉例來說，假設你有 100 個樣本，你可以第一次先使用前 90 個做訓練，另外 10 個做測試，然後再用第 80 到 90 個，不斷重複這個動作，這樣你可以得到不同的訓練/測試資料組合，提供更多數據去驗證。
(Ref. https://randomforests.wordpress.com/2014/02/02/basics-of-k-fold-cross-validation-and-gridsearchcv-in-scikit-learn/)
[補充] https://jason-chen-1992.weebly.com/home/-cross-validation
[考古題] 4-09

(9) 正規化參數(Regularization,又稱:L1 / L2 正規化)

在訓練模型時，常常會遇到訓練資料(training set)的資料正確性高，但套用至測試資料(testing set)時表現卻不如預期。以下圖為例，為了讓訓練資料的錯誤率(error rate)降至最低，因此訓練出的模型(藍線)貼近資料卻極其複雜，但其實所需的模型僅為一條簡單的線性迴歸(綠線)，而這種情況便是模型過適(model overfitting)。



a.如何處理 Overfitting

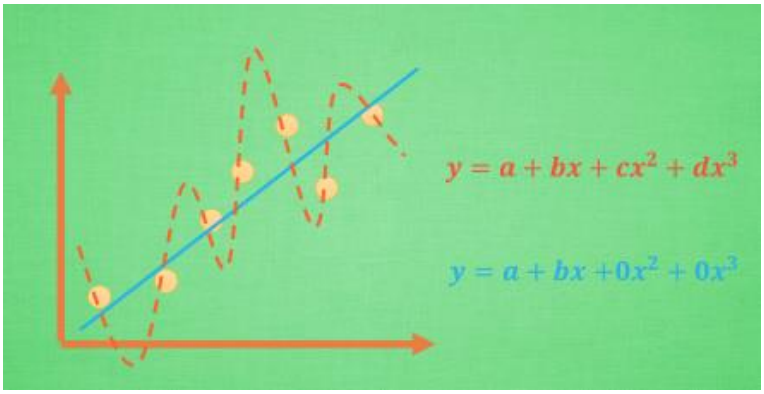
Overfitting 的原因多為採用過多參數，這有點像違反 Occam’ s razor 原則，因此模型面的調整方式有下列幾種:

- 特徵降維(feature extraction)
- 選擇其他適合的模型
- 針對模型作正規化(normalization)

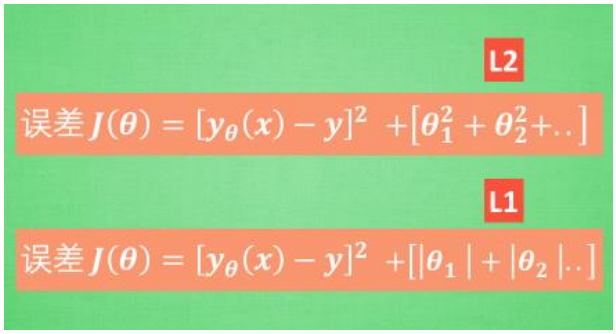
模型正規化主要針對**迴歸模型(regression model)**的損失函數(loss function)作調整，避免模型受到權重係數較高的參數影響，進而導致 overfitting。

(Ref.
https://medium.com/%E7%94%A8%E8%B3%87%E6%96%99%E8%AA%AA%E6%95%85%E4%BA%8B/%E8%BF%B4%E6%AD%B8%E6%A8%A1%E5%9E%8B%E6%AD%A3%E8%A6%8F%E5%8C%96-normalization-9141ed8b9edf)

b. L1, L2 Regularization



過擬合就是所謂的模型對可見的數據過度自信，非常完美的擬合上了這些數據，如果具備過擬合的能力，那麼這個方程就可能是一個比較複雜的非線性方程，正是因為這裡的 x^3 和 x^2 使得這條虛線能夠被彎來彎去，所以整個模型就會特別努力地去學習作用在 x^3 和 x^2 上的 cd 參數. 但是我們期望模型要學到的卻是這條藍色的曲線. 因為它能更有效地概括數據.而且只需要一個 y=a+bx 就能表達出數據的規律. 或者是說，藍色的線最開始時，和紅色線同樣也有 cd 兩個參數，可是最終學出來時, c 和 d 都學成了 0，雖然藍色方程的誤差要比紅色大，但是概括起數據來還是藍色好. 那我們如何保證能學出來這樣的參數呢? 這就是 L1 L2 正規化出現的原因.



對於剛剛的線條，使用一般用這個方程來求得模型 y(x) 和真實數據 y 的誤差，而 L1 L2 就只是在這個誤差公式後面多加了一個東西，讓誤差不僅僅取決於擬合數據擬合的好壞，而且取決於像剛剛 cd 那些參數的值的大小。

- L1 正規化- 每個參數的絕對值
- L2 正規化-每個參數的平方

(Ref. https://morvanzhou.github.io/tutorials/machine-learning/ML-intro/3-09-l1l2regularization/)

[考古題] 4-10

六、機器學習模型選擇

1.**隨機決策森林**（random decision forests）

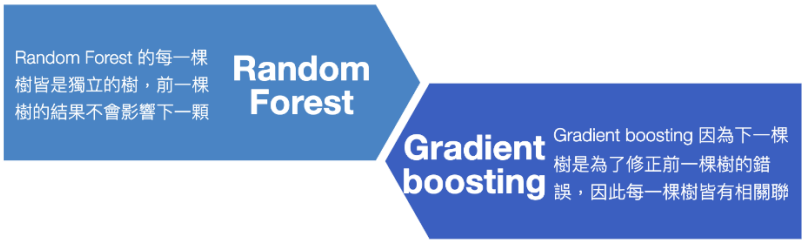
(Ref: https://zh.wikipedia.org/wiki/%E9%9A%8F%E6%9C%BA%E6%A3%AE%E6%9E%97)

在機器學習中，隨機森林是一個包含多個決策樹的分類器，並且其輸出的類別是由個別樹

輸出的類別的眾數而定。 Leo Breiman 和 Adele Cutler 發展出推論出隨機森林的演算法。而 "Random Forests"是他們的商標。這個術語是 1995 年由貝爾實驗室的 Tin Kam Ho 所提出的隨機決策森林(random decision forests)而來的。這個方法則是結合 Breimans 的"Bootstrap aggregating" 想法和 Ho 的"random subspace method" 以建造決策樹的集合。

*** Bagging 和 Boosting 的差別**

- Bagging 是透過抽樣 (Sampling) 的方式來生成每一棵樹，樹與樹之間是獨立生成的。
- Boosting 是透過序列 (Additive) 的方式來生成每一顆樹，每棵樹都會與前面的樹關聯，因為後面的樹要能夠修正。



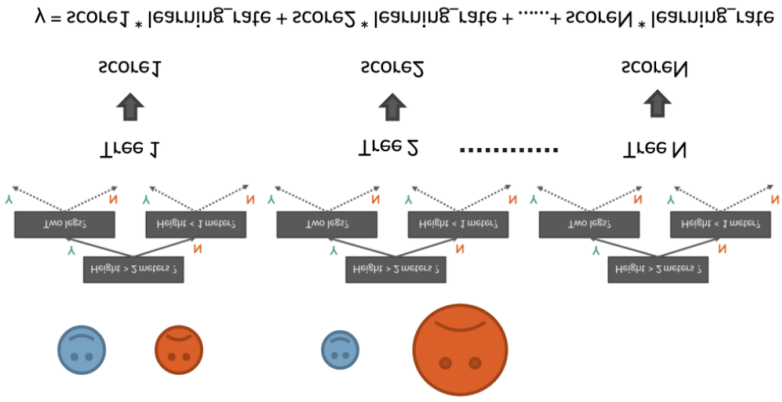
[考古題] 4-14, 4-15, 4-31

2.梯度提升機 (Gradient Boosting Machine)

(Ref. https://medium.com/uxai/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E9%A6%AC%E6%8B%89%E6%9D%BE-045-tree-based-model-%E6%A2%AF%E5%BA%A6%E6%8F%90%E5%8D%87%E6%A9%9F%E4%BB%8B%E7%B4%B9-a84d3059e13)

隨機森林使用的集成方法稱為 Bagging (Bootstrap aggregating)，用抽樣的資料與 features 生成每一顆樹，最橫在取平均。

Boosting 則是另一種集成方法，希望能夠由後面生成的樹，來修正前面樹學不好的地方，如何修正前面學錯的地方呢？透過計算 Gradient。每次生成樹都是要修正前面樹預測的錯誤，並乘上 Learning Rate，讓後面的樹有更多學習空間。



[考古題] 4-25

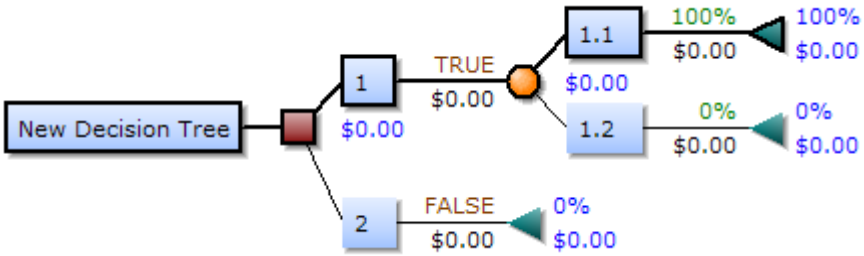
3.決策樹(Decision tree)

(Ref. https://zh.wikipedia.org/wiki/%E5%86%B3%E7%AD%96%E6%A0%91)

機器學習中，決策樹是一個預測模型；代表的是對象屬性與對象值之間的一種映射關係。樹中每個節點表示某個對象，而每個分叉路徑則代表某個可能的屬性值，而每個葉節點則對應從根節點到該葉節點所經歷的路徑所表示的對象的值。決策樹僅有單一輸出，若欲有複數輸出，可以建立獨立的決策樹以處理不同輸出。 數據挖掘中決策樹是一種經常要用到的技術，可以用於分析數據，同樣也可以用來作預測。從數據產生決策樹的機器學習技術叫做決策樹學習,通俗說就是決策樹。

一個決策樹包含三種類型的節點：

- 決策節點：通常用矩形框來表示
- 機會節點：通常用圓圈來表示
- 終結點：通常用三角形來表示



決策樹學習也是數據挖掘中一個普通的方法。在這裡，每個決策樹都表述了一種樹型結構，它由它的分支來對該類型的對象依靠屬性進行分類。每個決策樹可以依靠對源資料庫的分割進行數據測試。這個過程可以遞歸式的對樹進行修剪。當不能再進行分割或一個單獨的類可以被應用於某一分支時，遞歸過程就完成。另外，隨機森林分類器將許多決策樹結合起來以提升分類的正確率。

決策樹同時也可以依靠計算條件概率來構造。

決策樹如果依靠數學的計算方法可以取得更加理想的效果。 資料庫已如下所示：

(x, y) = (x1, x2, x3..., xk, y)

相關的變量 Y 表示我們嘗試去理解，分類或者更一般化的結果。 其他的變量 x1, x2, x3 等則是幫助我們達到目的的變量。

常見的決策樹類型:

- **分類樹**分析是當預計結果可能為離散類型（例如三個種類的花，輸贏等）使用的概念。
- **回歸樹**分析是當局域結果可能為實數（例如房價，患者住院時間等）使用的概念。
- **CART** 分析是結合了上述二者的一個概念。CART 是 Classification And Regression Trees 的縮寫.
- **CHAID**（Chi-Square Automatic Interaction Detector）

[考古題] 4-11, 4-12

4. **K-means (集群分析,又稱 c-means Clustering，中文: k-平均演算法)**

(Ref. https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E9%9B%86%E7%BE%A4%E5%88%86%E6%9E%90-k-means-clustering-e608a7fe1b43)

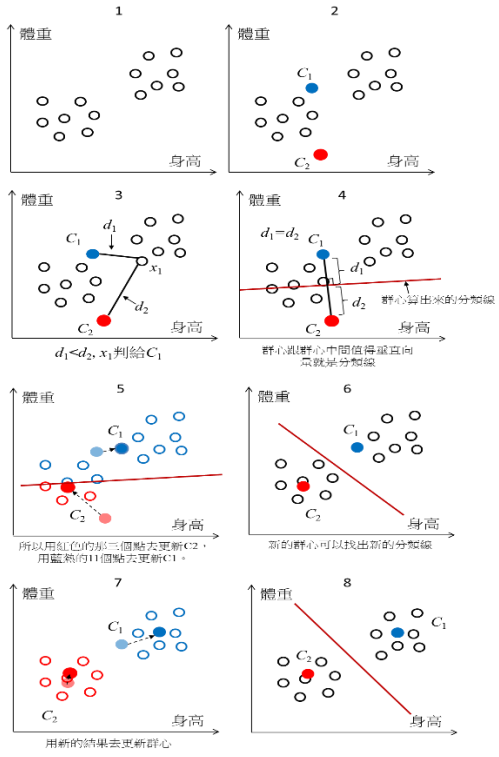
基本上 Clustering 的方法大都是非監督式學習(Unsupervised learning)，K-means 也是非監督式學習。

K-means Clustering 這個方法概念很簡單，一個概念「物以類聚」。男生就是男生，女生就是女生，男生會自己聚成一群，女生也會自己聚成一群。但在這群男生自己不會動成一群，女生也不會動成一群，在機器學習內，我們有的就是一組不會動的身高和體重的資料。那是什麼會動，讓男生女生可以區隔開的是什麼? 回頭看看演算法的名字，k-means，這邊的 k 是你想分成幾群，means 就是每一群群心，所以會動的東西就是群心。

K-means 運作概念步驟:

1. 我們先設定好要分成多少(k)群。
2. 然後在 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。
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，這邊距離當然也可以換成別種距離公式，但基本上都還是以歐式距離為主)。
4. 將每筆資料分類判給距離最近的那個群心。
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。
6. 一直重複 3 – 5，直到所有群心不在有太大的變動(收斂)，結束。

K-means 運作概念圖解:



[考古題] 4-17

5.主成分分析(Principal Component Analysis, PCA)

(Ref. https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8-%E7%B5%B1%E8%A8%88%E5%AD%B8%E7%BF%92-%E4%B8%BB%E6%88%90%E5%88%86%E5%88%86%E6%9E%90-principle-component-analysis-pca-58229cd26e71)

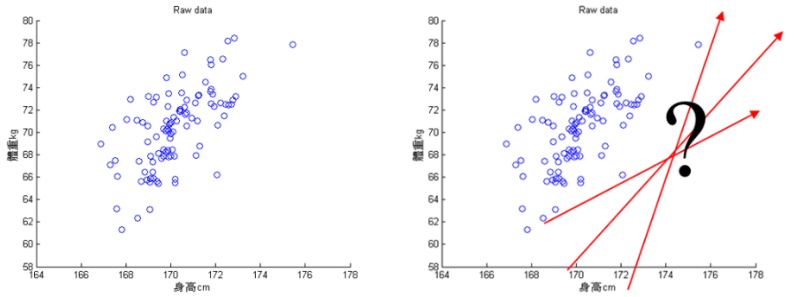
「主成分分析在機器學習內被歸類成為降維(Dimension reduction)內特徵擷取(Feature extraction)的一種方法，降維就是希望資料的維度數減少，但整體的效能不會差異太多甚至會更好」簡單說法是，降維(Dimension reduction)是當資料維度數(變數)很多的時候，有沒有辦法讓維度數(變數)少一點，但資料特性不會差太多。

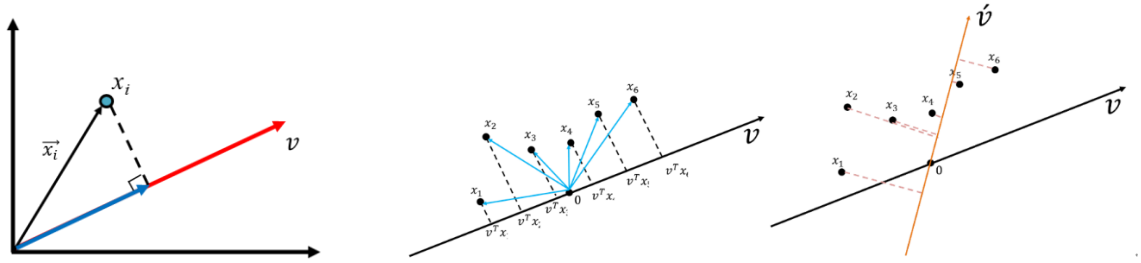
機器學習主要是希望用 PCA 達到 dimension reduction 的目的，主要是為了避免 Hughes 現象(Hughes Phenomenon)/ 維度詛咒(curse of dimensionality)。

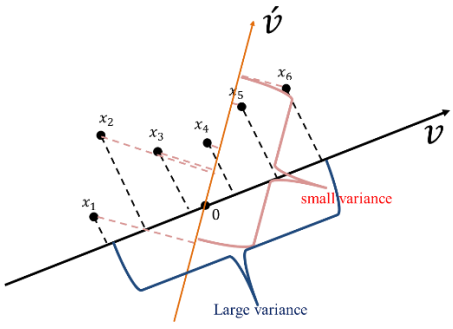
這邊先簡單說維度詛咒，預測/分類能力通常是隨著維度數(變數)增加而上生，但當模型樣本數沒有繼續增加的情況下，預測/分類能力增加到一定程度之後，預測/分類能力會隨著維度的繼續增加而減小。

主成份分析的基本假設是希望資料可以在特徵空間找到一個投影軸(向量)投影後可以得到這組資料的最大變異量，好文謬謬這是什麼意思哩?

我們來看一個圖(假設我們有一組資料變數是身高和體重)，我們希望找到一個向量可以投影(紅色的線)







[考古題] 4-16

七、類神經網路

(Ref. https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C)

對人類中樞神經系統的觀察啟發了人工神經網路這個概念。在人工神經網路中，簡單的人工節點，稱作神經元（neurons），連接在一起形成一個類似生物神經網路的網狀結構。人工神經網路目前沒有一個統一的正式定義。不過，具有下列特點的統計模型可以被稱作是「神經化」的：

- 具有一組可以被調節的權重（被學習演算法調節的數值參數）
- 可以估計輸入資料的非線性函式關係

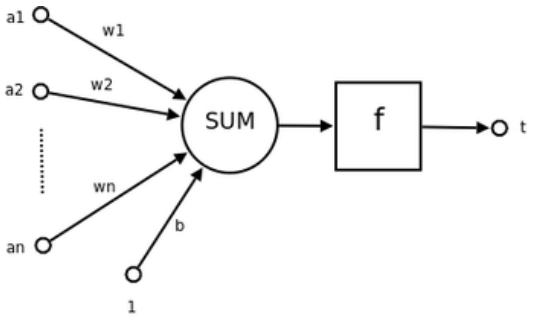
這些可調節的權重可以被看做神經元之間的連接強度。

人工神經網路與生物神經網路的相似之處在於，它可以集體地、並列地計算函式的各個部分，而不需要描述每一個單元的特定任務。神經網路這個詞一般指統計學、認知心理學和人工智慧領域使用的模型，而控制中央神經系統的神經網路屬於理論神經科學和計算神經科學。

典型的人工神經網路具有以下三個部分：

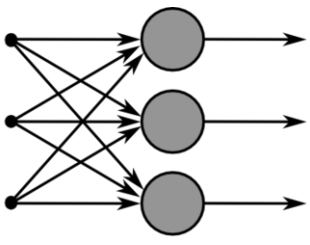
- **結構**（**Architecture**）結構指定了網路中的變數和它們的拓撲關係。例如，神經網路中的變數可以是神經元連接的權重（weights）和神經元的激勵值（activities of the neurons）。
- **激勵函式**（**Activation Rule**）大部分神經網路模型具有一個短時間尺度的動力學規則，來定義神經元如何根據其他神經元的活動來改變自己的激勵值。一般激勵函式依賴於網路中的權重（即該網路的參數）。
- **學習規則**（**Learning Rule**）學習規則指定了網路中的權重如何隨著時間推進而調整。這一般被看做是一種長時間尺度的動力學規則。一般情況下，學習規則依賴於神經元的激勵值。它也可能依賴於監督者提供的目標值和目前權重的值。

例如，用於手寫辨識的一個神經網路，有一組輸入神經元。輸入神經元會被輸入圖像的資料所激發。在激勵值被加權並通過一個函式（由網路的設計者確定）後，這些神經元的激勵值被傳遞到其他神經元。這個過程不斷重複，直到輸出神經元被激發。最後，輸出神經元的激勵值決定了辨識出來的是哪個字母。



- a1~an 為輸入向量的各個分量
- w1~wn 為神經元各個突觸的權值
- b 為偏置
- f 為傳遞函式，通常為非線性函式。一般有 traingd(),tansig(),hardlim()。以下預設為 hardlim()
- t 為神經元輸出

單層神經元網路是最基本的神經元網路形式，由有限個神經元構成，所有神經元的輸入向量都是同一個向量。由於每一個神經元都會產生一個純量結果，所以單層神經元的輸出是一個向量，向量的維數等於神經元的數目。



output layer

類神經網路基本結構

一種常見的多層結構的前饋網路（Multilayer Feedforward Network）由三部分組成:

- 輸入層（Input layer），眾多神經元（Neuron）接受大量非線形輸入訊息。輸入的訊息稱為輸入向量。
- 輸出層（Output layer），訊息在神經元鏈接中傳輸、分析、權衡，形成輸出結果。輸出的訊息稱為輸出向量。
- 隱藏層（Hidden layer），簡稱「隱層」，是輸入層和輸出層之間眾多神經元和鏈接組成的各個層面。隱層可以有一層或多層。隱層的節點（神經元）數目不定，但數目越多神經網路的非線性越顯著，從而神經網路的強健性（robustness）（控制系統在一定結構、大小等的參數攝動下，維持某些效能的特性）更顯著。習慣上會選輸入節點 1.2 至 1.5 倍的節點。

這種網路一般稱為感知器（對單隱藏層）或多層感知器（對多隱藏層），神經網路的類型已經演變出很多種，這種分層的結構也並不是對所有的神經網路都適用。

[補充] 關於啟動函數(又稱激活函數/激勵函數, Activation Function)

(Ref. http://cvfiasd.pixnet.net/blog/post/275774124-%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92%E6%BF%80%E5%8B%B5%E5%87%BD%E6%95%B8%E4%BB%8B%E7%B4%B9)

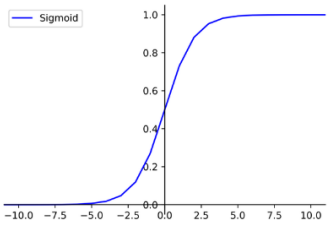
激活函數負責爲神經網絡引入非線性特徵。它把值壓縮到一個更小範圍，即一個 Sigmoid 激活函數的值區間爲 [0,1]。深度學習中有很多激活函數，ReLU、SeLU 、TanH 較 Sigmoid 更爲常用。更多激活函數，請參見《一文概覽深度學習中的激活函數》。

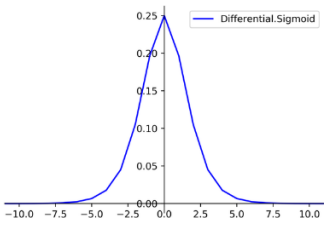
在類神經網路中使用激勵函數，主要是利用非線性方程式，解決非線性問題，若不使用激勵函數，類神經網路即是以線性的方式組合運算，因為隱藏層以及輸出層皆是將上層之結果輸入，並以線性組合計算，作為這一層的輸出，使得輸出與輸入只存在著線性關係，而現實中，所有問題皆屬於非線性問題，因此，若無使用非線性之激勵函數，則類神經網路訓練出之模型便失去意義。

而某些激勵函數，如 Sigmoid 函數在深度網絡的權重更新中，常會因為層數過多，導致導數逐漸變為 0，使得前幾層之權重參數無法順利更新，造成神經網絡無法被優化，而無法找出最佳結果，原因如以下兩點:

1.在下圖中容易看出，當中 Sigmoid 導數較大或較小時(區間[-5,+5]之外)，導數接近 0，而後向傳遞時需要將當層導數與之前各層導數的數值進行乘積，幾個趨近於 0 的小數相乘，結果很接近 0。

2.如圖中可見 Sigmoid 導數的最大值為 0.25，因此在每一層會因為導數被壓縮為原來的，意味著導數在每一層至少會被壓縮為原來的 1/4，通過兩層後被變為 1/16，通過 n 層後為 1/4 的 n 次方 ，這邊還是以最大導數為例，因此過深的網路架構也會使導數相乘逐漸為 0。

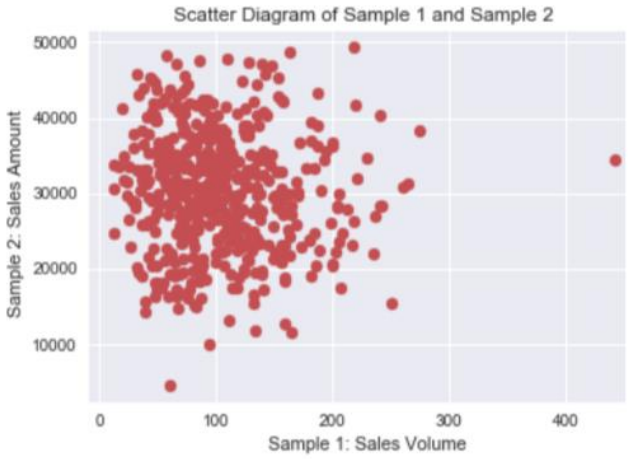


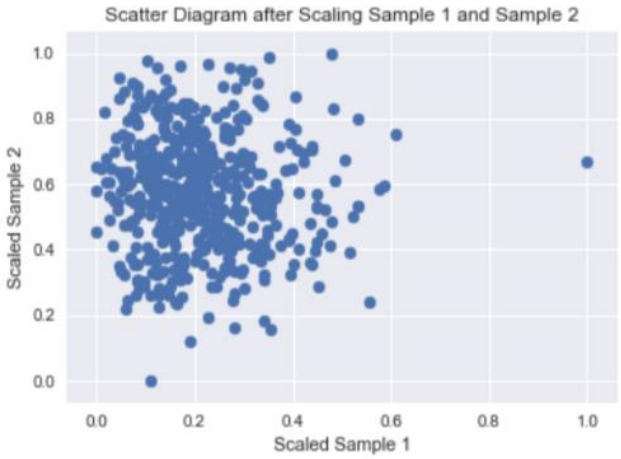


2.資料的正規化(Normalization)及標準化(Standardization)

在比較分析兩組數據資料時，可能會遭遇因單位的不同(例如：身高與體重)，或數字大小的代表性不同(例如：粉專 1 萬人與滿足感 0.8)，造成各自變化的程度不一，進而影響統計分析的結果；為解決此類的問題，可利用資料的正規化(Normalization)與標準化(Standardization)，藉由將原始資料轉換成無量綱(Dimensionless)的純量後，來進行數據的比較及分析。

資料的正規化(Normalization)是將原始資料的數據按比例縮放於 [0, 1] 區間中，且不改變其原本分佈。舉例來說，若我們現有兩組數據資料，分別表示 500 項商品的銷售量 Sample 1 及銷售額 Sample 2，如下圖所示，很明顯地，此兩組資料的單位不同，且數字上有著懸殊的差異，分別透過資料正規化後，兩組資料將同時轉換成純量縮放於 [0,1] 區間中，如下右圖所示；這樣的資料轉換，能排除資料單位的限制，提供我們一個相同的基準來進行後續比較分析。





資料的標準化(Standardization)可運用在機器學習演算法中，它能帶給模型下面兩個好處：

1. 提升模型的收斂速度

在建構機器學習模型時，我們會利用梯度下降法(Gradient Descent)來計算成本函數(Cost Function)的最佳解；假設我們現有兩個特徵值 x1 in [0,1] 與 x2 in [0,10000]，則在 x1-x2 平面上成本函數的等高線會呈窄長型，導致需較多的迭代步驟，另外也可能導致無法收斂的情況發生。因此，若將資料標準化，則能減少梯度下降法的收斂時間。

2. 提高模型的精準度

將特徵值 x1 及 x2 餵入一些需計算樣本彼此的距離(例如:歐氏距離)分類器演算法中，則 x2 的影響很可能將遠大於 x1，若實際上 x1 的指標意義及重要性高於 x2，這將導致分析的結

果失真。因此，資料的標準化是有必要的，可讓每個特徵值對結果做出相近程度的貢獻。

如何進行標準化及正規化？

1. Z 分數標準化(Z-Score Standardization)

假設資料的平均數與標準差分別為 μ 及 σ ，Z 分數標準化可利用下列公式進行：

$$Z = \frac{X - \mu}{\sigma} \sim N(0, 1)$$

經 Z 分數標準化後，資料將符合標準常態分佈(Standard Normal Distribution)，轉換後的平均值=0、標準差=1，且用標準分數或稱 Z 分數(Z-Score)來作為單位。Z 分數標準化適用於分佈大致對稱的資料，因為在非常不對稱的分佈中，標準差的意義並不明確，此時若標準化資料，可能會對結果做出錯誤的解讀，另外，當未知資料的最大值與最小值，或存在超出觀察範圍的離群值時，可透過 Z 分數標準化來降低離群值對整個模型的影響。

2.最小值最大值正規化(Min-Max Normalization)

最小值最大值正規化的用意，是將資料等比例縮放到 [0, 1] 區間中，可利用下列公式進行轉換：

$$X_{nom} = \frac{X - X_{min}}{X_{max} - X_{min}} \in [0, 1]$$

其中 X_{max} 與 X_{min} 分別為資料中的最小值與最大值。此種方法有一點需我們特別注意，即若原始資料有新的數據加入，有可能導致最小值 X_{min} 及最大值 X_{max} 的改變，則這時候我們需再重新定義公式中的 X_{min} 及最大值 X_{max}。另外，若將轉換公式修改成下列：

$$X_{nom} = \frac{X - \mu}{X_{max} - X_{min}} \in [-1, 1]$$

其中 μ 為資料的平均值，則資料將縮放到 [-1, 1] 區間中且平均值 = 0，我們稱這為平均值正規化(Mean Normalization)。

(Ref. https://aifreeblog.herokuapp.com/posts/54/data_science_203/)

3.遞迴內神經網路(RNN)

4.CNN

Ref. https://blogs.nvidia.com.tw/2018/09/whats-the-difference-between-a-cnn-and-an-rnn/

5.殘差網路

6. Q- Learning