機器學習題目：

統計學

定性數據(Qualiative Data)：對事物性質描述數據。ex 股票所屬行業數據即為定性數據。ex 國巨屬於半導體業。
定量數據(Quantitative Data)：事物數量的數據，由數字所組成。ex 每個股票的開盤價。

透過計算得到單變量的集中趨勢：平均值 Mean、中位數 Median、眾數 Mode。
或者計算資料分散程度：最小值 Min、最大值 Max、範圍 Range、四分位差、變異數 Variance、標準差 Standard deviation。

數據的離散度
反應數據分佈的特徵也稱為數據的變異性，常用的離散度指標有

1. **全距(Range)** = max - min
2. **平均絕對偏差(Mean Absolute Deviation, MAD)：**數據點與其平均值之間差異程度。MAD通常用於衡量數據集的離散程度，它不受極端值的影響，因為它是所有數據點與平均值的絕對偏差的平均值。與標準差相比，MAD更加穩健，對於具有極端值或者異常值的數據集更加適用。

$$MAD = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

3. **變異數(Variance)：**變異數也是用來描述數據的離散程度

樣本變異數

變異數公式：

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1} = \frac{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}{n-1}$$

4. **標準差(Standard Deviation)：**標準差就是變異數的平方根。標準差是衡量數據分散程度的一個指標，它表示數據相對於均值的平均偏差。計算標準差的方法是先計算每個數據點與均值的差值的平方，然後將這些平方差值相加，再取平均值，最後取平方根。標準差越大，數據的變異性越高。

$$s = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n-1}}$$

5. 四分位數（Quartile）是指將數據集分成四等份的統計量，通常用來描述數據的分佈情況和數值的位置。在統計學中，常用的四分位數有三個，分別是第一個四分位數（Q1）、第二個四分位數（Q2，即中位數）和第三個四分位數（Q3）。

   在計算四分位差時，需要先將數據集按大小順序排列，然後將數據集分成四個部分，每個部分包含大約 25% 的數據點。這四個部分分別是：

1. 第一個四分位數（Q1）：將數據集的25%的數據點放在左側，表示25%的數據點小於或等於第一個四分位數。
2. 第二個四分位數（Q2）：即為中位數，將數據集的50%的數據點放在左側，表示50%的數據點小於或等於中位數。
3. 第三個四分位數（Q3）：將數據集的75%的數據點放在左側，表示75%的數據點小於或等於第三個四分位數。

四分位差（Interquartile Range，簡稱IQR），也稱為內距或四分間距。是第三個四分位數和第一個四分位數之間的範圍(上四分位數與下四分位數之差)，即：

IQR=Q3−Q1

IQR可以衡量數據集中間50%的資料離散程度，它不受數據集中的極端值影響，因此通常用作衡量數據的離散程度的穩健指標。四分位差數值越小，表示中間的資料越集中；數值越大，表示中間的資料越分散。

均值（Mean）：

均值是一組數據的平均值，計算方法是將所有數據值相加，然後除以數據的總數。均值可以表示數據的集中趨勢，是數據的一個重要摘要統計量。

樣本（Sample）：

在統計學中，樣本是從整體母體中抽取的一部分數據。通常，我們無法從整個母體中獲得所有數據，因此我們將從中隨機選取一些數據構成樣本，然後根據樣本的統計量來估計母體的特徵。

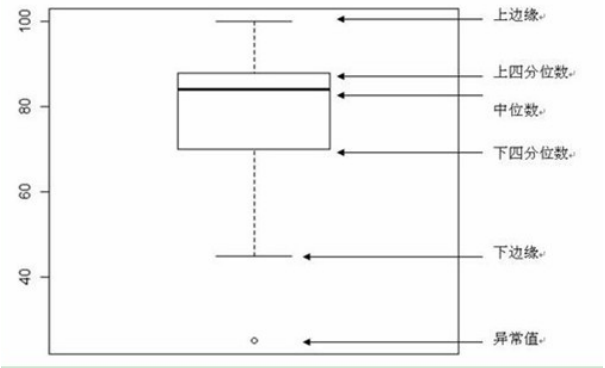
母體（Population）：

母體是指我們想要研究的全部數據集合，它包含了我們感興趣的所有數據。母體可以是有限的，也可以是無限的。在統計學中，我們通常關心的是對母體進行推斷，但由於無法直接觀察到母體的所有數據，因此我們通常根據樣本來進行推斷。

資料前處理、資料清洗。

機器學習中的資料預處理（Data Preprocessing）是指在應用機器學習演算法之前對原始資料進行處理，提升資料的品質，提升模型的性能或訓練過程。 資料預處理通常包括資料清洗、特徵選擇、特徵變換、特徵縮放、資料集劃分等步驟。

檢查離群值、異常值：透過資料視覺化(如直方圖、盒鬚圖（Box Plot，也稱為盒狀圖、箱型圖或箱形圖）、分布圖等)，或透過統計方法(平均數及標準差等) 來找出離群值。



1. **資料清洗（Data Cleaning）**：資料清洗是指識別和處理資料中的不完整、不一致、重複、錯誤或異常值等問題，以確保資料品質和準確性。
   常見的資料清洗步驟包括：
   1. **缺失值處理**：辨識並處理資料中的缺失值，常用的方法包括刪除包含缺失值的樣本、填入缺失值（如平均值、中位數或眾數）或使用K最近鄰法、內插法進行填入。
   2. **異常值處理**：辨識並處理資料中的異常值，常用的方法包括刪除異常值、將異常值替換為特定的值（如平均值或中位數）、進行平滑處理等；分箱處理：將數據分成不同的區間，將異常值劃分到特定的區間中。
   3. **重複值處理**：常用的方法包括刪除重複樣本、保留唯一的數據。合並重複值：將數據中的重複值合並為單個值，可以根據特定的規則進行合並，如取平均值、合計值等。
   4. **錯誤值處理**：辨識並處理資料中的錯誤值，例如資料型別錯誤、範圍錯誤等，常用的方法包括手動修正錯誤、使用規則進行自動校正等。
   5. **利用機器學習模型來預測缺失值或者填充異常值。**
   6. **使用多個方法組合：可以使用多種不同的方法來處理缺失值、重複值和異常值。**

2. 資料轉換：將資料標準化、歸一化等處理，使得資料符合演算法的輸入要求。

資料預處理的目的是提高機器學習模型的效能和泛化能力，確保模型建立在高品質的資料基礎上，從而更好地應對實際問題。 因此，資料清洗是資料預處理的重要步驟之一，對於資料品質和準確性有重要影響。

探索式資料分析EDA (Exploratory Data Analysis)

探索式資料分析是運用視覺化或基本統計等工具，來對資料有個初步的認識。
幫助：
1. 獲取資料的資訊、結構和特點。
2. 檢查有無離群值或異常值，看資料是否有誤。
3. 分析各變數間的關聯性，找出重要的變數。

進行EDA能檢查資料是否符合分析前的假設、在模型建立前先發現潛在的錯誤，並進一步調整分析方向。
四圖表
1. 趨勢圖
2. 散佈圖 - 資料符合隨機分布時，如圖資料無結構性且隨機散落。
3. 直方圖 - 資料常態分布時，如圖呈鐘形分布。
4. 正態概率圖 - 資料常態分佈時，如圖呈線性。

一般而言，EDA能提供我們一些概括性的解答像：
1. 資料中標準值為何？
2. 資料的百分位數？
3. 哪些屬於重要資訊？
4. 多變量中有無結構？

5. 資料中有沒有離群值？

好的EDA能幫助我們對資料認識，並有助於之後去除離群值、革除雜訊等資料前處理。Pandas中已經有許多寫好的函數，能用以觀察資料，下篇文章會做初步介紹。

處理類別型數據（Categorical Data）和數值型數據（Numerical Data）

處理類別型數據（Categorical Data）和數值型數據（Numerical Data）需要使用不同的方法，這些方法將數據轉換為可用於建模和分析的格式。

類別型數據處理：

編碼：將類別型數據轉換為數字表示，常見的編碼方法包括：

1. 獨熱編碼（One-Hot Encoding）：將每個類別轉換為一個二進制特徵，對於每個類別，只有一個特徵的值為1，其餘特徵的值為0。
2. 標籤編碼（Label Encoding）：將每個類別映射為一個整數，通常用於有序類別型數據。

數值型數據處理：

1. 標準化（Normalization）：將數值型數據縮放到一個特定的範圍，通常是[0, 1]或者[-1, 1]，常見的標準化方法包括：
2. 最小-最大標準化（Min-Max Scaling）：將數據縮放到指定的最小值和最大值之間。
3. Z-score標準化：將數據按照均值和標準差進行標準化，使其均值為0，標準差為1。
4. 離散化（Discretization）：將連續型數據劃分為不同的區間或者分箱，將其轉換為類別型數據。

根據具體的數據集和分析需求，可以選擇不同的處理方法來處理類別型數據和數值型數據。在應用這些方法時，需要注意保持數據的信息完整性和可解釋性，避免數據轉換過程中造成信息損失。此外，建議在使用任何數據處理方法之前，仔細檢查數據集的特性和結構，並根據實際情況進行調整和優化。

請描述不同類型的機器學習。

監督式學習（Supervised Learning）：

- 其訓練數據包含輸入和對應的標籤。模型通過這些標記數據進行訓練，以預測新的輸入數據。常見的監督式學習算法包括線性回歸、支持向量機（SVM）、決策樹、隨機森林、神經網絡等。

非監督式學習（Unsupervised Learning）：

- 非監督式學習的訓練數據沒有對應的標籤或對應的輸出，模型自行發現數據中的結構和模式。常見的非監督式學習算法包括聚類（如K均值聚類、層次聚類）、降維（如主成分分析（PCA）和t-SNE）、關聯規則學習等。

半監督式學習（Semi-Supervised Learning）：

- 半監督式學習是監督式學習和非監督式學習的結合，使用少量標記數據和大量未標記數據來進行訓練。這種方法可以在標記數據有限的情況下擴展模型的性能。

強化學習（Reinforcement Learning）：

- 強化學習是一種透過與環境互動，獲得試錯懲罰和成功獎勵，來學習最優行動策略的機器學習方法。常見的強化學習算法包括Q學習、深度Q網絡（DQN）、策略梯度等。

自監督式學習（Self-Supervised Learning）：

- 自監督式學習是一種從未標記的數據中自動生成標籤來進行學習的方法。模型通常通過預測數據中的某些部分來生成標籤，然後使用這些標籤來進行監督式學習。自監督式學習在圖像、語音和文本處理等領域中得到了廣泛應用。

偏差（bias）和方差（variance）

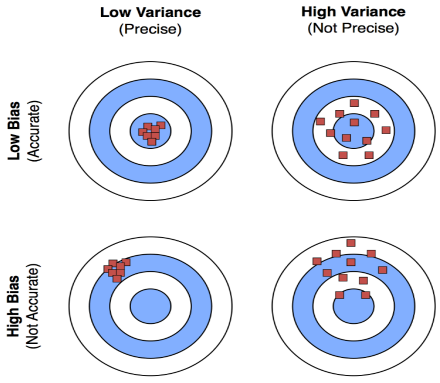
在機器學習中，偏差和方差通常是相互競爭的，稱為“偏差-方差折衷”(Bias-Variance Tradeoff)。一個理想的機器學習模型需要在偏差和方差之間取得平衡，以實現良好的泛化性能。

1. 偏差（Bias）：
   - 偏差描述了模型的預測與真實值之間的平均差異。高偏差的模型通常意味著對於訓練數據的拟合能力不足(欠擬合的問題)，導致在訓練集和測試集上都表現不佳，模型沒有足夠的靈活性來捕捉數據中的複雜模式。
2. 方差（Variance）：
   - 方差描述了模型對於不同訓練數據的敏感性。高方差的模型在訓練集上可能表現很好，但在測試集上表現可能較差。這是因為高方差的模型對於訓練數據中的噪聲和隨機性過度敏感，導致過度擬合訓練數據，無法泛化到新的數據上。

以下是幾個解決偏差和方差問題的方法：

- 增加模型複雜度：降低偏差，但可能增加方差。
- 減少模型複雜度：降低方差，但可能增加偏差。
- 正則化：通過限制模型參數的大小來降低方差，同時對模型的複雜度進行控制。
- 交叉驗證：通過交叉驗證技術來選擇最佳的模型，以平衡偏差和方差。

總的來說，理解和管理偏差和方差之間的平衡是訓練有效機器學習模型的關鍵。



成本函數（Cost Function）與損失函數（Loss Function）差別

在機器學習中，成本函數（Cost Function）和損失函數（Loss Function）是兩個關鍵概念，它們都用於評估模型的性能。

1. 成本函數（Cost Function）：
    - 成本函數是用來衡量模型對整個訓練集的性能。
    - 成本函數的目標是最小化訓練集上的平均損失，以此來使模型在整個數據集上的性能最佳化。通常，成本函數是由損失函數計算得到的，並且可能包含額外的正則化項。
2. 損失函數（Loss Function）：
    - 損失函數是用來評估模型對於單個訓練樣本的預測值與實際值之間的差異。
    - 損失函數的目標是尋找最佳的模型參數，以使每個單獨的數據樣本的預測盡可能地接近其真實值。在監督式學習中，損失函數通常用於計算梯度下降算法中的梯度，以便更新模型參數。

總的來說，成本函數和損失函數的目標都是評估模型的性能，並且它們都是為了最小化預測值與實際值之間的差異。但是，成本函數通常用於整個訓練集上的模型性能評估，而損失函數則用於單個數據樣本的性能評估，並且通常用於梯度下降等優化算法中。

根據數據類型決定使用的機器學習技術

為了開發最佳擬合方法，我們必須首先使用探索性數據分析（EDA）檢查數據並理解利用數據集的目標。

- 當數據是線性時，使用線性回歸。
- 如果數據表明非線性，bagging 方法會表現得更好。
- 如果必須出於商業目的評估或解釋數據，我們可以使用決策樹或 SVM。
- 如果數據集包括照片、視頻和音頻，神經網絡可能有助於獲得準確的答案。

如何選擇分類器

選擇分類器時，通常需要考慮以下幾個因素：

1. 訓練集大小：
    - 小型訓練集：當訓練集較小時，傾向於使用複雜度較低、參數較少的模型，以防止過度擬合（Overfitting）。此外，具有較小的複雜度的模型通常訓練速度較快，更適合用於小型數據集。
    - 大型訓練集：當訓練集較大時，可以考慮使用更複雜、更強大的模型，因為這些模型有能力從更大的數據集中學習到更多的模式和特徵。

2. 數據類型：
    - 線性分類器：適用於線性可分的問題，如線性回歸、邏輯回歸等。
    - 非線性分類器：適用於非線性問題，如支持向量機（SVM）、決策樹、隨機森林、深度神經網絡等。
3. 特徵數量：
    - 少量特徵：對於少量特徵的問題，可以使用較簡單的模型，如邏輯回歸、決策樹等。
    - 大量特徵：對於大量特徵的問題，可以考慮使用支持向量機（SVM）、隨機森林、深度神經網絡等較複雜的模型，這些模型可以更好地處理高維數據。
4. 計算效率：
    - 輕量級模型：對於計算資源有限的場景，如移動設備或嵌入式系統，需要考慮模型的計算效率，選擇較輕量的模型，如邏輯回歸、輕量級神經網絡等。
    - 高性能模型：對於計算資源充足的場景，如大型伺服器或雲端計算平台，可以選擇更複雜、更強大的模型，如深度神經網絡、集成學習等。

總的來說，選擇分類器時需要綜合考慮訓練集大小、數據類型、特徵數量、計算效率等因素，並根據具體的問題情況和應用需求來進行選擇。在實際應用中，可以通過交叉驗證等技術來評估不同模型的性能，並選擇最適合的模型。

協方差（Covariance）與相關性（Correlation）：

協方差和相關性常用於特徵工程和數據分析中，都是理解不同變量之間的關係的統計量，或者評估特徵與目標變量之間的關聯性。這些統計量可以幫助我們篩選特徵、理解模型的複雜性，或者檢測數據中的多重共線性。

協方差（Covariance）：協方差是衡量兩個變量之間的變動趨勢的統計量，它描述了變量之間的共變異性。具體來說，如果兩個變量的變化趨勢是相似的，即當其中一個變量增加時，另一個變量也增加，或者當其中一個變量減少時，另一個變量也減少，那麼它們之間的協方差將是正的。如果它們的變化趨勢相反，則協方差將是負的。如果協方差為零，則兩個變量沒有線性相關性。協方差的範圍是負無限到正無限。協方差的絕對值表示兩個變量變化的趨勢程度，但無法標準化，因此在比較不同數據集之間的關係時，可能不太方便。

公式：$\operatorname{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{n-1}$

相關性（Correlation）：相關性是協方差的標準化版本(將協方差除以每個變量的標準差)，衡量兩個變量之間的線性關係強度和方向。從而使其值落在 -1 到 1 的範圍內。相關性為 1 意味著兩個變量之間存在完全正向的線性關係，-1 表示完全負向的線性關係，而 0 則表示沒有線性關係。相關性的絕對值越接近 1，表示兩個變量之間的關係越強。

$\operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$

協方差（Covariance）與相關性（Correlation）的差別：

定義：

- 協方差是衡量兩個隨機變量之間的變動趨勢，它描述了變量之間的共變異性。
- 相關性是協方差的標準化版本(將協方差除以每個變量的標準差)，衡量兩個變量之間的線性關係強度和方向，

範圍：

- 協方差的範圍是負無限到正無限。
- 相關性的範圍是 -1 到 1 之間，其中 -1 表示完全負相關，0 表示無相關，1 表示完全正相關。

解釋性：

- 相關性的解釋性更強，因為它是標準化的且在固定範圍內。例如，相關性為 0.8 表示兩個變量之間有較強的正線性關係，而相關性為 -0.5 表示兩個變量之間有較強的負線性關係。
- 協方差的解釋性較差，因為它的取值受到變量單位的影響，且範圍較廣。

您首選的機器學習算法是什麼？

以下是一些需要考慮的典型機器學習算法：

- 線性回歸
- 邏輯回歸
- 樸素貝葉斯
- 決策樹
- K 表示
- 隨機森林算法
- K-最近鄰（KNN）

相關係數 (Correlation Coefficient)

一般說的相關係數通常是指「皮爾森相關係數(Pearson's correlation coefficient)」。相關係數是用來衡量兩個變量之間相關性的統計量，它描述了兩個變量之間的線性相關程度。如果兩個變量的相關係數接近1，則表示它們之間存在著強烈的正線性相關性；如果相關係數接近-1，則表示它們之間存在著強烈的負線性相關性；如果相關係數接近0，則表示它們之間幾乎沒有線性相關性。

相關係數通常用符號「r」表示，計算公式如下：

其中，相關係數通常用符號「r」。n 表示樣本數，x 和 y 分別表示兩個變量的值。

$$\rho = \frac{x和y的共變異數}{x的標準差 \times y的標準差}$$

共變異數(covariance): $\text{cov}(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$

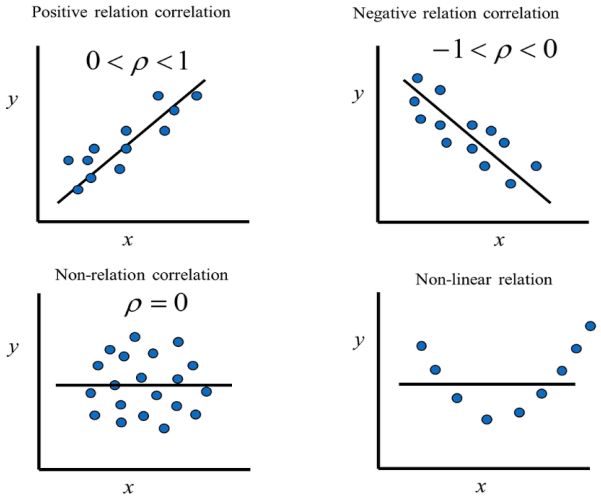
變異數(variance): $\text{var}(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)^2$

標準差(standard deviation): $std(x) = \sqrt{\text{var}(x)}$

$$r = \frac{S_{xy}}{S_x S_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

相關係數的取值範圍在-1到1之間。如果相關係數為正，則表示兩個變量呈現正相關，即當一個變量增加時，另一個變量也增加；如果相關係數為負，則表示兩個變量呈現負相關，即當一個變量增加時，另一個變量減少；如果相關係數為0，則表示兩個變量之間沒有線性相關性。

這邊共變異數、變異數都是除上(n-1)而不是n的原因是，只用少部分樣本在推論母體時因為偏量(bias)的關係，在推論時樣本推估會少一個自由度



共變異數(covariance)

我們從公式看，我們將x和y減去各自的平均數後相乘最後算總和，這邊如果我們假設變數x等於變數y時，那這個共變異數不就等於變異數，這時候ρ的值不就等於1，所以x和y就完全相關(x=y)。

所以共變異數其實就等於在算x和y的相關程度，但是兩變數的單位還存在。必須除於標準差以消除單位，成為相關係數(-1~1)，沒有任何單位。

最小平方法（Least Squares Method）是一種常用的數學和統計方法，用於估計一組數據點所構成的模型的參數。它通常應用於回歸分析中，用於找到一個最符合數據點的模型，使得這個模型的預測值與實際觀測值的殘差平方和最小。

最小平方法的目標是最小化殘差平方和的總和，即最小化以下損失函數：

$$L(\theta) = \sum_{i=1}^{n}(y_i - f(x_i;\theta))^2$$

R平方(R squared)
又稱為判定係數(coefficient of determination)，是一種衡量回歸模型表現的指標，代表從獨立變數X可以解釋依變數Y變異的比例。
殘差(residual)：每個樣本點的真實值y_i-預測值f_i

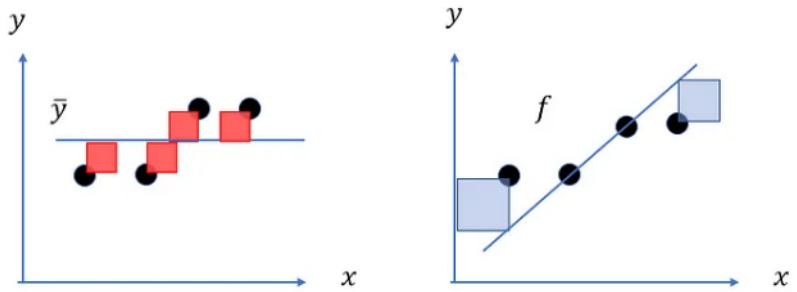
總平方和（Total Sum of Squares，TSS）評估一組數據的變異性(數據的離散程度)，也叫總變異量，表示整個數據集中所有數值與平均值之間的變異性。

$$TSS = \sum_{i=1}^{n}(x_i - \bar{x})^2$$

殘差平方和（Sum of Squares Residual，SSR）就是『不能解釋的部分』，表示模型未能解釋的變異性，等於模型的預測誤。

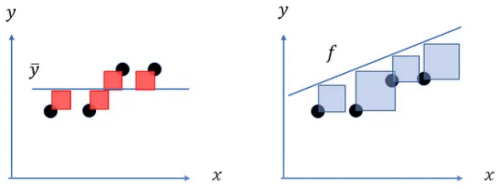
$$SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$R^2 = 1 - \frac{SS_{\text{res}}}{TSS}$$



殘差平方和越小，表示不能解釋的部分越小，則R平方自然越高，如果找到一條完美回歸線穿越所有資料點，使殘差變成0，R平方就會變成1，表示模型解釋所有變異量。因此，正常情況下R平方的值會落在[0,1]。

R平方有沒有可能是負的？
有，當模型的殘差平方合大於總變異量時，R平方就是負的，如下圖：



參數模型和非參數模型

參數模型和非參數模型是兩種不同的模型類型，它們在建模和訓練方式上有著不同的特點。

參數模型（Parametric Models）：

1. 參數模型假設數據的分佈屬於某個已知的參數化分佈，並且試圖通過對這些參數進行估計來擬合模型。
2. 這些模型的主要特點是模型的結構是固定的，而且通常有一個固定數量的參數。一旦模型結構和參數確定，訓練過程就是通過最大化似然函數或最小化損失函數來求解這些參數的過程。
3. 常見的參數模型包括線性回歸、邏輯回歸、高斯混合模型等。

非參數模型（Non-parametric Models）：

1. 非參數模型不對數據的分佈做出明確的假設，非參數模型的模型結構是彈性的，根據數據的複雜性和分佈來自適應調整模型的複雜度(模型的複雜度不固定)。
2. 常見的非參數模型包括決策樹、k最近鄰算法（k-Nearest Neighbors，KNN）、核密度估計等。

主要差異：

1. 模型結構：
    - 參數模型具有固定的模型結構，並且模型的複雜度通常由參數的數量決定。
    - 非參數模型的模型結構是彈性的，可以根據數據的複雜性和分佈自動調整。
2. 估計方法：
    - 參數模型的參數通常通過最大化似然函數或最小化損失函數等優化目標來進行估計。
    - 非參數模型的參數個數不固定，通常通過機器學習算法自適應地調整。
3. 應用場景：
    - 參數模型通常適用於數據集相對簡單，並且符合某種已知分佈的情況。
    - 非參數模型通常適用於數據集複雜，並且難以通過參數化分佈進行建模的情況。

超參數（Hyperparameters）與模型參數（Model Parameters）

超參數（Hyperparameters）是指在模型訓練之前需要設定的參數，不是透過訓練資料學習得到的，而是由人工設定的。超參數影響模型的性能與學習過程，例如控制模型的複雜度、學習速率、正則化強度等。 常見的超參數包括學習率、正規化參數、樹的深度、神經網路的隱藏層節點數等。選擇合適的超參數組合對於模型的性能和泛化能力至關重要。

模型參數（Model Parameters），模型參數是在訓練過程中透過最佳化演算法自動學習得到的。

超參數之所以與模型參數不同，主要體現在以下幾個方面：

1. **設定方式不同：**模型參數是模型透過訓練資料學習得到的，而超參數是在訓練模型之前手動設定的，通常需要經過調優或搜尋來找到最優的超參數組合。
2. **控制模型學習過程：**超參數直接影響了模型的學習過程，例如學習率的大小會影響模型在每一步更新時的步長，正規化參數會影響模型對複雜度的懲罰程度等。
3. **不同的超參數組合會導致不同的模型性能：**對於同一個模型，不同的超參數組合會導致不同的模型性能，因此選擇合適的超參數組合對於模型的性能至關重要。
4. **超參數通常需要調優**：由於超參數需要手動設置，並且對模型性能有重要影響，因此通常需要進行超參數調優來找到最優的超參數組合，以達到模型性能的最佳水平。

總之，超參數在機器學習中起著重要的作用，透過控制模型的學習過程和性能，影響模型的表現和泛化能力。 選擇合適的超參數組合對於模型的性能和泛化能力至關重要。

函數逼近（Function Approximation）

函數逼近（Function Approximation）是指利用給定的數據集來逼近或近似出一個未知函數的過程。這個未知函數可能是某種規律、趨勢或者複雜的映射關係，通常用數學函數表示。

函數逼近最常見的應用之一是在監督式學習中。在監督式學習中，我們有一組帶有標籤的數據集（輸入和對應的輸出），我們的目標是找到一個函數，使其能夠將給定的輸入映射到對應的輸出。

常見的函數逼近方法包括：

- 線性回歸（Linear Regression）：通過擬合一條或多條直線/平面/超平面來逼近數據中的關係。線性回歸假設輸入和輸出之間存在線性關係。
- 多項式回歸（Polynomial Regression）：通過擬合一個多項式函數來逼近數據中的關係。多項式回歸可以捕捉到非線性的數據關係。
- 支持向量機（Support Vector Machines，SVM）：通過在高維空間中找到一個最大邊界超平面來逼近數據中的分類或回歸關係。SVM 可以處理線性和非線性的函數逼近問題。
- 決策樹（Decision Trees）：通過樹結構來將輸入空間劃分為多個區域，每個區域對應一個預測值。決策樹可以處理分類和回歸問題。
- 神經網絡（Neural Networks）：通過多層神經元組成的網絡來逼近數據中的複雜映射關係。深度學習中的神經網絡可以處理非常複雜的函數逼近問題。

惰性學習器（Lazy Learner）與 主動學習器（Eager Learner）

惰性學習器（Lazy Learner）：惰性學習器在訓練階段不會構建模型構建或估計參數，而是將訓練數據集存儲在內存中(會記住訓練數據集)，並在預測時在訓練數據中檢索相關訊息和計算相似性。

優點：

1. 訓練速度快：因為惰性學習器不需要在訓練階段進行計算，所以訓練速度通常很快。
2. 對新數據適應性好：由於模型直接使用訓練數據進行預測，因此對新數據的適應性較好。

缺點：

1. 預測速度較慢：在預測時，惰性學習器需要在訓練數據中檢索訊息和計算相似性，因此預測速度可能較慢。
2. 對噪音數據敏感：惰性學習器直接使用訓練數據進行預測，對噪音數據較為敏感。

以下是幾種常見的惰性學習器：

1. 最近鄰居算法（k-Nearest Neighbors，KNN）：KNN是一種基於實例的惰性學習算法。在預測時，KNN從訓練數據中找到與測試樣本最接近的k個鄰居，然後根據這些鄰居的標籤進行預測。
2. 核密度估計（Kernel Density Estimation，KDE）：KDE是一種通過估計樣本點周圍的密度分佈來進行概率密度估計的方法。在預測時，KDE通過計算新樣本點周圍的密度分佈來進行預測。
3. 案例檢索（Case-Based Reasoning，CBR）：CBR是一種通過在訓練數據集中查找與新問題相似的先前案例來進行預測的方法。當找到相似的案例後，CBR將該案例的結果應用於新問題。

主動學習器（Eager Learner）：主動學習器在訓練階段會生成一個特定的模型，並進行參數估計。當需要預測時，直接使用已經生成的模型進行預測。

優點：

1. 預測速度快：主動學習器在預測時不需要額外計算，因此預測速度通常較快。
2. 對噪音數據魯棒性較好：主動學習器通常會對訓練數據進行適當的處理和模型擬合，因此對噪音數據較為魯棒。

缺點：

1. 訓練速度較慢：主動學習器需要在訓練階段進行計算和模型構建，因此訓練速度可能較慢。

2. 對新數據適應性差：主動學習器在訓練階段生成的模型可能過於特定，對新數據的適應性較差。

差異比較：

1. 行為方式：惰性學習器在預測時才進行計算，而主動學習器在訓練階段進行計算和模型構建。
2. 速度：惰性學習器的訓練速度快，但預測速度較慢，而主動學習器的預測速度快，但訓練速度較慢。
3. 對數據的適應性：惰性學習器對新數據的適應性較好，而主動學習器的適應性取決於訓練階段生成的模型。

正則化（Regularization）

正則化是一種用於防止機器學習模型過度擬合（Overfitting）的技術。正則化通過向模型的損失函數添加一個正則項（Regularization Term），來限制模型的複雜度，防止模型對訓練數據過度擬合。

常見的正則化方法包括 L1 正則化（L1 Regularization）和 L2 正則化（L2 Regularization）。L1 正則化將模型參數的絕對值加到損失函數中，L2 正則化將模型參數的平方和加到損失函數中。

正則化的目標是優化模型的泛化性能，並在訓練和測試數據之間取得更好的平衡。

數據規範化=歸一化（Normalization）、標準化（Standardization）

在統計學上沒有Standardization，只有Normalization。因為有些模型對數據範圍非常的敏感，例如KNN。歸一化是一種數據預處理技術，用於調整數據的範圍和分佈，使得所有特徵都具有相似的尺度。這有助於模型訓練過程中更快地收斂，並且可以提高模型的性能和穩定性。

常見的歸一化方法包括將數據縮放到一個特定的範圍（例如 [0, 1] 或 [-1, 1]）、標準化（將數據縮放成均值為0、方差為1）、最大最小歸一化等。

1. **最小-最大歸一化（Min-Max Normalization）**：最小-最大歸一化將每個特徵的原始值減去最小值，然後除以最大值減去最小值的範圍來實現。將數據縮放到一個固定的範圍內（[0, 1] 或 [-1, 1]），消除數值範圍大小不同的影響，保留原始資料的分佈形態和相對大小關係。更快地收斂，提高模型的性能和穩定性。具體公式如下：

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$ x 是原始特徵值，min 是特徵的最小值， max 是特徵的最大值。

缺點：

1. 對異常值敏感：最小-最大歸一化是基於最大值和最小值進行縮放的，如果數據中存在極端異常值，這些異常值會導致所有其他值都接近於 0 或 1，使得數據的差異性降低。
2. 不保證標準化：最小-最大歸一化將數據縮放到一個固定範圍內，但數據不一定為正態分佈，也無法保證標準化。

2. **標準化（Standardization）**：在統計學上沒有Standardization，只有Normalization。標準化是將數據縮放成均值為 0、標準差為 1 的標準常態分佈，數據分佈已改變，可減小離群值對於模型的影響。這種歸一化方法通過將每個特徵的原始值減去特徵的平均值，然後除以特徵的標準差來實現。有助於模型訓練過程中更快地收斂，提高模型的性能和穩定性。具體公式如下：

$$x_{\text{std}} = \frac{x-\mu}{\sigma}$$ x 是原始特徵值，μ 是特徵的平均值，σ 是特徵的標準差。

缺點：

1. 不適用於所有分佈：標準化假設數據服從正態分佈，如果數據不服從正態分佈，標準化可能會產生不理想的結果。
2. 難以解釋：標準化後的數據往往難以直觀地解釋，因為特徵值的尺度已經被轉換，且可能與原始數據不同。
3. 需要計算平均值和標準差：標準化需要提前計算每個特徵的平均值和標準差，這增加了計算成本，特別是對於大型數據集而言。

3. **歸一化到單位長度（Unit Length Normalization）**：歸一化到單位長度是將數據樣本的每個特徵向量縮放成歐氏長度為 1。這種歸一化方法通過將每個樣本的特徵向量除以其歐氏長度來實現。具體公式如下：

$$x_{\text{unit}} = \frac{x}{\|x\|}$$ x 是原始樣本，‖x‖ 表示樣本 x 的歐氏長度。

4. **魯棒縮放（Robust Scaler）**：Robust Scaler 是一種魯棒的歸一化方法，對於有較多異常值的數據集效果較好。它將數據縮放到一個指定的範圍，通常是 [-1, 1] 或 [0, 1]。該方法使用中位數和四分位範圍（interquartile range，IQR）而不是最小值和最大值來計算縮放因子。如果數據中含有異常值在縮放中會捨去。

歸一化（Normalization）、標準化（Standardization）主要區別：

- 資料規範化將資料縮放到一個特定的範圍內，常見的是[0, 1] 或[-1, 1]；而標準化將資料轉換為平均值為0、標準差為1 的標準常態分佈。
- 資料規範化保留了原始資料的分佈形態和相對大小關係；而資料標準化改變資料分佈，變成標準常態分佈，使得資料更易於比較和解釋，適用於一些要求資料平均值和變異數相近的演算法（如PCA）。
- 資料規範化可能對異常值敏感，因為它受到最大值和最小值的影響，異常值會導致所有其他值都接近於 0 或 1，使得數據的差異性降低。；而資料標準化在一定程度上對異常值不那麼敏感，因為它是基於平均值和標準差。

聚類（Clustering）

是一種無監督學習的方法(不需要事先標記的訓練數據)，通過計算樣本之間的相似度或距離來將樣本劃分為幾個互相區分的群組。同一群組的數據的特徵和質量是相似的，而屬於不同群組的數據點的特徵和質量是不同的。

聚類在許多領域都有廣泛的應用，包括市場分析、社交網絡分析、醫學影像處理、文本分類、圖像分割等。常見的聚類算法包括K均值聚類、層次聚類、DBSCAN、高斯混合模型等。

聚類的主要步驟包括：

- 選擇合適的特徵：確定要在聚類過程中使用的特徵。
- 選擇聚類算法：選擇適合應用場景的聚類算法。
- 初始化簇中心：對於一些聚類算法，需要初始化簇中心。
- 計算相似度或距離：計算樣本之間的相似度或距離。
- 分配樣本到簇：根據相似度或距離將樣本分配到最接近的簇中。
- 更新簇中心：對於一些聚類算法，需要根據分配的樣本更新簇中心。
- 重複迭代：重複執行分配樣本到簇和更新簇中心的步驟，直到收斂條件滿足。

線性回歸

有監督的機器學習算法 線性回歸：它用於預測分析以確定因變量和自變量之間的線性關係。

線性回歸的方程如下：

- Y = A + BX

- 輸入或自變量稱為 X。
- 因變量或輸出變量是 Y。
- X的係數為b，截距為a。

Ridge Regression和 Lasso Regression的區別

Ridge 迴歸（Ridge Regression）和 Lasso 迴歸（Lasso Regression）是搭配正規化技術的線性迴歸，它們在處理多重共線性和特徵選擇方面有不同的特性。 它們的差異主要體現在正則化項的不同和對係數的影響：

正規化項的不同：

- Ridge 迴歸使用 L2 範數作為正規化項，即將係數向量的 L2 範數（係數的平方和）加入損失函數中，公式為：

$$\text{Ridge Loss} = \text{MSE} + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Lasso 迴歸使用 L1 範數作為正規化項，即將係數向量的 L1 範數（係數的絕對值總和）加入損失函數中，公式為：

$$\text{Lasso Loss} = \text{MSE} + \lambda \sum_{j=1}^{p} |\beta_j|$$

對係數的影響：

- Ridge 迴歸的 L2 正規化會使得係數的值被約束在一個較小的範圍內(+平方(數值大))，但不會將係數壓縮到零，因此 Ridge 迴歸傾向於保留所有特徵。
- Lasso 回歸的 L1 正則化具有稀疏性，它傾向於將不重要的特徵的係數壓縮到零，因此 Lasso 回歸可以用於特徵選擇，即自動選擇具有較大影響力的特徵。

解的穩定性：

- 由於 L1 範數具有稀疏性，Lasso 迴歸傾向於產生稀疏解，有助於模型的解釋性和特徵選擇。
- Ridge 迴歸傾向於產生稠密解，因此對於高維度資料中的多重共線性問題更為穩定，但可能難以進行特徵選擇。

綜上所述，Ridge 迴歸和 Lasso 迴歸在正規化項的選擇和對係數的影響上有所不同，因此在實際應用中需要根據問題的特徵和需求選擇合適的方法。 如果需要稀疏性和特徵選擇，則可以考慮使用 Lasso 回歸；如果需要處理多重共線性並保留所有特徵，則可以考慮使用 Ridge 回歸。

貝葉斯定理 (Bayes' Theorem)==貝氏定理(Bayes' theorem):

貝葉斯定理（Bayes' Theorem）= 貝氏定理：是一種機率論定理，描述在已知一些先驗條件的情況下，計算某一個事件發生的機率，透過條件機率公式達到條件機率的互換。

貝葉斯定理是機率論中的一個基本定理，用於計算在給定一些先驗條件的情況下，某一事件的條件概率。假設 A 和 B 是兩個事件，且 P(B)>0，則貝葉斯定理表示為：

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

其中，P(A|B)是在 B 發生的條件下 A 發生的概率；P(B|A) 是在 A 發生的條件下 B 發生的概率，P(A) 和 P(B) 分別是事件 A 和事件 B 的先驗機率。

條件機率：的公式是P(A|B)=P(A∩B)/P(B)，意味著在事件B發生的條件下，事件A發生的機率，而我們將分子P(A∩B)透過P(B∩A)條件機率公式替換成P(B|A)P(A)，來達到條件機率的互換，如下。

$$P(A|B) = \frac{P(\mathrm{A} \cap \mathrm{B})}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

經過上述的轉換不難發現貝氏定理其實就是從『給定A事件已發生的條件，B事件發生的條件機率』轉變成『給定B事件已發生的條件下，A事件發生的條件機率』的互換過程而已。公式用常理推斷符合生活上的經驗，比如在已知一個人抽菸，計算他得肺癌的機率，假設為P(肺癌|抽菸)，如果得肺癌的人越多人有抽菸行為，也就是P(抽菸|肺癌)的值越大，則P(肺癌|抽菸)的值也會比較大。

貝氏定理可以計算出為廣告訊息的機率，進而大幅提高過濾垃圾訊息的精準度，因此時常作為攔截垃圾信件的用途。

貝氏分類器：

就是透過貝氏定理計算樣本在不同類別條件下的條件機率，並取得條件機率最大者作為預測類別。

計算上的小技巧，眼尖的讀者會發現，代入不同的類別，影響的只有分子，分母都是固定為P(x)，因此倘若只要比較各類別條件機率的大小，分母可以不用計算，單純比較分子就好！比如只有a、b兩個類別計算各類別的條件機率時，分母都固定為p(x)，因此貝氏分類器就單純比較

P(x|a)P(a)和P(x|b)P(b)兩者之間大小(各類別條件機率的大小)，如果前者值比較大就判給類別a作為預測結果，反之亦然。

當我們知道其他概率時，我們可以使用貝葉斯定理來確定概率。 換句話說，它基於先驗信息提供了發生的後驗概率。

樸素貝葉斯 (簡單貝葉斯)

https://kknews.cc/zh-tw/code/ax2xkox.html
樸素貝葉斯分類器是一種基於貝葉斯定理的弱分類器，所有樸素貝葉斯分類器都假設每個屬性都同樣重要和獨立(每個特徵與其他特徵都不相關)。舉個例子，如果一種水果其具有紅，圓，直徑大概3英寸等特徵，該水果可以被判定為是蘋果。儘管這些特徵相互依賴或者有些特徵由其他特徵決定，然而樸素貝葉斯分類器認為這些屬性在判定該水果是否為蘋果的概率分布上獨立的。

容易建立，特別適合用於大型數據集，是一種勝過許多複雜算法的高效分類方法。在文本分析、垃圾郵件過濾和推薦系統中，它們被使用。

貝葉斯最優分類器（Bayes Optimal Classifier）：

貝葉斯最優分類器是在給定先驗概率和損失函數的情況下，最小化期望損失的分類器。換句話說，它是根據貝葉斯定理以及給定的損失函數來做出最佳的分類決策。貝葉斯最優分類器能夠在理論上實現最佳的分類性能，但通常情況下，由於先驗概率和損失函數都是未知的，因此很難實現。樸素貝葉斯算法可以視為是貝葉斯最優分類器的一種近似方法，在實際應用中獲得了廣泛的應用和成功。

詞袋模型（Bag-of-Words model）

詞袋模型（Bag-of-Words model）是自然語言處理（NLP）中一種常用的文本表示方法，用於將文本轉換為數字向量。該模型假設文本中的每個單詞都是獨立存在的，忽略單詞之間的語序和上下文關係，僅關注單詞的出現頻率(就像拿袋子裝下每個語詞，不考慮順序及文法)。

生成模型（Generative Models）

生成模型（Generative Models）是一類機器學習模型，其目標是對給定數據集的概率分布進行建模，從而能夠生成與原始數據類似的新數據樣本。與之對應的是判別模型（Discriminative Models），判別模型關注的是對給定輸入進行分類或者回歸等任務。

生成模型的核心思想是通過學習數據的概率分布來生成新的數據樣本，這樣可以獲得更多關於數據的結構信息，並且能夠生成具有一定意義的新數據。生成模型在許多應用領域都有重要的應用，包括自然語言處理、圖像生成、音頻生成、樂譜生成等。

常見的生成模型包括：

1. 生成對抗網絡（Generative Adversarial Networks，GANs）：
    - GANs 是一種深度學習模型，由一個生成器和一個鑑別器組成，它們通過對抗訓練來提高生成器的性能。生成器的目標是生成與真實數據相似的假數據，而鑑別器的目標是區分真實數據和假數據。
2. 變分自編碼器（Variational Autoencoders，VAEs）：
    - VAEs 是一種深度學習模型，結合了自編碼器和變分推理的思想，通過學習潛在變量的概率分布來生成新的數據。VAEs 可以學習數據的連續潛在表示，並且可以通過從潛在空間中取樣來生成新的數據樣本。
3. 生成式對抗網絡（Generative Adversarial Nets，GAN）：
    - GAN 是一種神經網絡結構，由生成器和鑑別器組成。它們通過對抗訓練來學習數據的分布。生成器試圖生成與真實數據相似的假數據，而鑑別器試圖區分真實數據和假數據。
4. 自編碼器（Autoencoders）：
    - 自編碼器是一種無監督學習模型，它通過學習將輸入數據編碼成一個低維度的表示，然後再解碼回原始輸入數據。通過這種方式，自編碼器可以生成與原始數據相似的新數據。

生成模型的目標是學習數據的概率分布，並且能夠從中生成新的數據樣本，這對於很多需要生成新數據的應用非常有價值。

集成學習（Ensemble Learning）

集成學習（Ensemble Learning）是一種機器學習技術，旨在通過將多個弱學習器組合成一個強大的學習器，結合不同模型的預測(平均或投票)，補充彼此的優勢、抵消彼此的弱點，從而提高了模型的準確性和泛化能力。這種技術利用了“眾人的智慧”。

步驟

1. 抽取樣本、選擇變數
2. 從訓練資料集中訓練預測/分類器子集們(member classifiers)
3. 組合這些子集預測/分類器們的預測結果(ensemble members)，形成集成學習。

整合結果的方式

1. 類別型輸出結果(Class Labels Outputs)
2. 連續型輸出結果(Continuous Outputs)
    a. 平均 Mean Rule
    b. 加權平均 Weighted Average
    c. 截尾平均 Trimmed mean (去除極端值後計算的算術平均值)
    d. Generalized Mean
    e. Minimum, Maximum, Median, Product Rule...

常見的集成學習技術包括：

1. 投票法（Voting）：
    - 投票法是一種簡單的集成學習技術，通過統計多個基本學習器的預測結果，選擇最終的預測結果。常見的投票方式包括硬投票（根據多數決原則選擇）和軟投票（根據概率加權選擇Weighted majority voting）、波達計數法（Borda Count）。
2. Bagging（Bootstrap Aggregating, 引導聚集演算法，又稱裝袋演算法）：
    - Bagging : 在訓練集中進行隨機的取後放回抽樣(Bootstrap抽樣)，抽樣多次生成多個子集，然後在每個子集上訓練一個基本學習器，最後平均或投票這些學習器的預測結果，來得到最終的預測。Bagging 模型中。每個子模型的 Bias 都很低，但是個別的variance 較高容易過度擬合，因此把多個模型平均起來降低variance。
      常見的 Bagging 應用包含 Random Forest 隨機森林(容易過度擬合)。

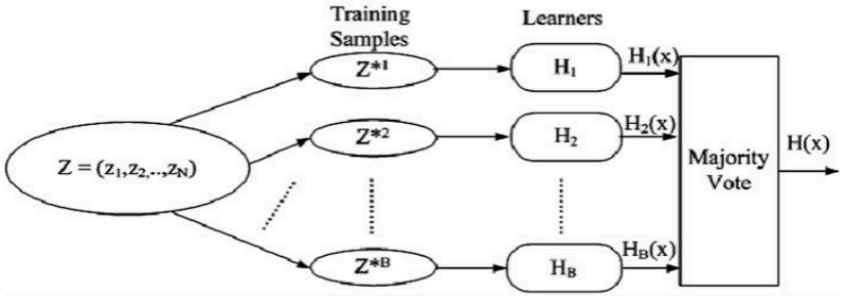


Fig. 2.3 The bagging approach to classification. Using bootstrap, we produce several training samples; each of these samples is fed into a weak learner. The final classification decision is produced by a majority vote on the weak learners output

3. Boosting：
    - Boosting 被稱為提升法，是一種迭代的集成學習方法。原則上是先去建立一個較弱的模型，將前一輪分類錯誤的資料進行加權，再用梯度下降法優化模型，重複迭帶下，變成強學習器，進而降低模型的 Bias。

    - 常見的 Boosting 算法包括 AdaBoost、Gradient Boosting Machine（GBM）、XGBoost、LogitBoost、 Gradient Boosted Decision Trees (GBDT) 和 LightGBM。

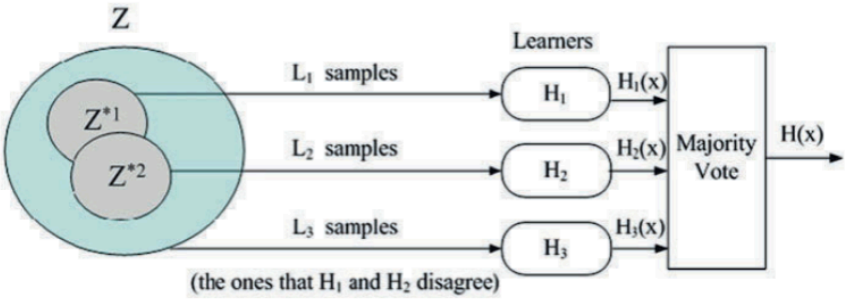


Fig. 2.4 A graphical idea of the first boosting approach proposed in [95]. Notice that each learner can be itself learned by the boosting algorithm in a recursive fashion

4. Stacking (Stacked Generalization)：

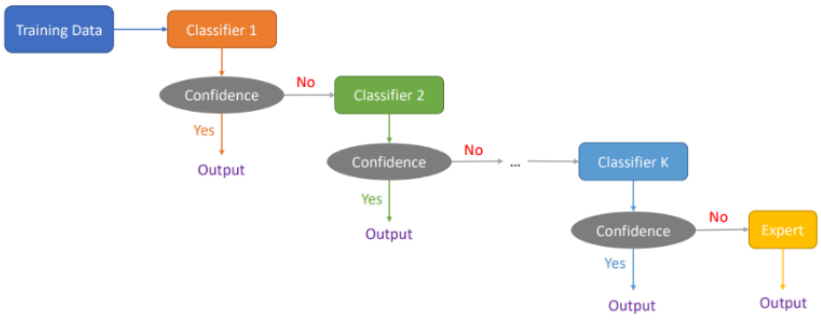
- Stacking 被稱為堆疊法，是一種層次化的集成學習方法。通過將多個獨立學習器的預測結果作為新的特徵，再訓練一個元學習器（或稱為元分類器 Meta-Classifier）來進行最終的預測。主要被用來提高模型的預測/分類準確率。Stacking 通常需要一個留出的驗證集來訓練元學習器。

5. Cascading：
    - Cascading 又稱為級聯法，它通常應用於分類問題。多個分類器按照一定的順序進行組合，將整個分類問題分解為多個子問題，然後各分類器依次解決每個子問題，最終組合所有子問題的解決方案得到最終的預測結果。

    - 每個分類器的輸出都作為下一個分類器的輸入。每個分類器都被設計為解決整個問題的一部分，並且能夠補充前一個分類器的弱點，從而提高整體的分類性能。這種方法有助於簡化整個分類問題的複雜度，並且提高了模型的泛化能力。

    - Cascading 方法的一個常見的應用是在人臉識別中。例如，一個 Cascading 分類器可以首先檢測圖像中的臉部區域，然後將這些區域傳遞給下一個分類器進行性別識別，最後將這些信息組合起來進行最終的識別。

    - Cascading 的優點包括：
        - 分解問題：將大型的分類問題分解為多個子問題，使得每個子問題更容易解決。
        - 組合優勢：每個子分類器都專注於解決特定的問題，能夠利用不同的特徵和模型來提高整體的預測性能。
        - 抗干擾能力：由於每個子分類器只處理整個問題的一部分，因此整個系統對於噪聲和干擾具有一定的魯棒性。



Adaboost (Adaptive Boosting 自適應增強)

Adaboost（自適應增強）是一種集成學習算法，用於改進弱學習器的性能，以提高整體預測的準確性。Adaboost是一種迭代算法，每一輪迭代訓練一個弱分類器（例如決策樹），連續訓練一系列的弱分類器，並根據之前的分類結果增加錯誤分類的樣本權重，調整弱分類器的權重，最終組合所有弱分類器以生成一個強分類器。

以下是Adaboost算法的基本步驟：

1. 初始化權重：將每個樣本的權重初始化為相等值。
2. 迭代訓練弱分類器：
    a. 對於每一輪迭代：
    b. 使用當前樣本權重訓練一個弱分類器。
    c. 通過使用這個弱分類器對訓練數據進行預測，計算出錯誤率（被錯誤分類的樣本的權重總和）。
    d. 計算這個弱分類器的權重，權重高的弱分類器在最終組合模型中將有更大的影響。
    e. 通過更新樣本權重，將錯誤分類的樣本權重增加，正確分類的樣本權重減少。
3. 組合弱分類器：將所有弱分類器根據其權重組合成一個強分類器。
4. 生成最終預測：將新數據輸入到強分類器中，並生成最終的預測結果。

Adaboost的優點包括對多種類型的數據和分類器的通用性，以及對噪聲數據的魯棒性。它還可以自適應地調整弱分類器的權重，從而提高整體的分類性能。

然而，Adaboost也有一些缺點，例如對噪聲和異常值敏感，並且當弱分類器過於複雜時容易過擬合。為了解決這些問題，通常需要對弱分類器進行適當的調參，以及對訓練數據進行預處理和特徵工程。

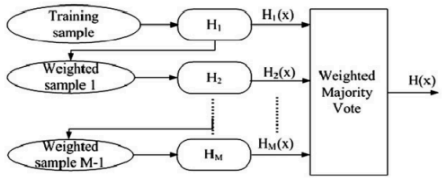


Fig. 2.6 Graphical idea of the adaptive boosting algorithm (adapted from [58]). Each weak learner is trained on a different weighted version of the training data sample. There is no sampling of the training data and the weight of each instance for the following round depends on the performance of the previous learner

Algorithm 4 (Discrete) AdaBoost algorithm for binary classification

Input: Dataset $Z = \{z_1, z_2, \ldots, z_N\}$, with $z_i = (\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathscr{X}$ and $y_i \in \{-1, +1\}$. M, the maximum number of classifiers.
Output: A classifier $H : \mathscr{X} \to \{-1, +1\}$.

1: Initialize the weights $w_i^{(1)} = 1/N$, $i \in \{1, \ldots, N\}$, and set $m = 1$.
2: **while** $m \leq M$ **do**
3: Run weak learner on Z, using weights $w_i^{(m)}$, yielding classifier $H_m : \mathscr{X} \to \{-1, +1\}$.
4: Compute $\text{err}_m = \sum_{i=1}^{N} w_i^{(m)} \, h\left(-y_i \, H_m(\mathbf{x}_i)\right)$, the weighted error of H_m.
5: Compute $\alpha_m = \frac{1}{2} \log\left(\frac{1-\text{err}_m}{\text{err}_m}\right)$. {/* Weight of weak classifier. */}
6: For each sample $i = 1, \ldots, N$, update the weight $v_i^{(m)} = w_i^{(m)} \exp(-\alpha_m \, y_i \, H_m(\mathbf{x}_i))$.
7: Renormalize the weights: compute $S_m = \sum_{j=1}^{N} v_j$ and, for $i = 1, \ldots, N$, $w_i^{(m+1)} = v_i^{(m)} \Big/ S_m$.
8: Increment the iteration counter: $m \leftarrow m + 1$
9: **end while**
10: Final classifier: $H(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^{M} \alpha_j \, H_j(\mathbf{x})\right)$.

梯度提升和隨機森林（Random Forest）

梯度提升（Gradient Boosting）和隨機森林（Random Forest）都是常用的集成學習算法，用於解決監督學習問題，例如分類和回歸。雖然它們都是基於決策樹的算法，但在一些方面存在明顯的差異。

1. 構建方式：

- 隨機森林是通過構建多棵獨立的決策樹來進行集成，每棵樹是在隨機抽樣的訓練集上進行構建的。
- 梯度提升是通過逐步構建一系列的決策樹來進行集成，每棵樹都是為了最小化前一棵樹的預測殘差而構建的。

2. 樹的關聯性：
    - 隨機森林中的每棵樹是獨立構建的，並且彼此之間沒有關聯性，因此可以同時進行訓練。
    - 梯度提升中的每棵樹是串行構建的，每棵樹都是在前一棵樹的殘差基礎上構建的，因此需要依次進行訓練。
3. 擬合方式：
    - 隨機森林通常對噪聲具有較好的魯棒性，因為它通過多個樹的投票來進行預測，可以減少過擬合的風險。多類目標檢測 適用於隨機森林。
    - 梯度提升通常對噪聲敏感，因為它是逐步擬合每棵樹，容易將異常值與噪聲也納入到模型中，導致模型變得過擬合。在不平衡的數據時，就像在實時風險評估中一樣，梯度提升表現良好。
4. 調參方式：
    - 隨機森林的調參相對較簡單，主要包括樹的數量、每棵樹的深度、特徵抽樣比例等。
    - 梯度提升的調參相對較複雜，需要調整學習率、樹的數量、每棵樹的深度、子樣本的比例等。梯度提升則在一些競賽和應用中獲得了更好的預測性能，但需要較多的調參和訓練時間

支持向量機（Support Vector Machine，SVM）

支持向量機（Support Vector Machine，SVM）是一種用於分類和回歸分析的監督式學習算法。SVM的目標是找到一個最佳的超平面，以將不同類別的數據點有效地分離開來。SVM的主要思想是最小化損失和最大化邊界Margin (超平面和支持向量之間的距離)「間隔帶」的間距為ϵ，對所有落入到間隔帶內的樣本不計算損失。這樣做可以使分類器的泛化能力更強，

以下是SVM的一些主要特點和概念：

1. **分隔超平面（Separating Hyperplane）：**在二維空間中，一個超平面就是一條直線，它將兩個類別的數據點分開。在更高維度空間中，超平面是一個超曲面。SVM 的目標是找到一個最佳的分隔超平面，使得兩個類別之間的邊界最大化。

2. **支持向量（Support Vectors）**：支持向量是距離分隔超平面最近的那些訓練數據點。這些數據點對於定義分隔超平面至關重要，因為它們決定了超平面的位置和方向。
3. **最大化邊界（Maximizing Margin）**：利用支援向量來算出Margin，SVM的主要思想是最大化邊界Margin(Margin就是超平面和最近的支持向量之間的距離)。Margin越大，代表分類器的泛化能力更強，因為它更不容易受到新數據的影響。「間隔帶」的間距為ϵ，對所有落入到間隔帶內的樣本不計算損失，並最小化損失和最大化"間隔帶"的寬度來找到最佳的模型，
4. **核技巧（Kernel Trick）**：當數據點無法通過一個線性超平面完美地分離時，可透過核函數將數據映射到更高維度的空間中，使其在該空間中變得可分離。這種映射函數通常被稱為核函數，SVM通過核技巧來計算在這個高維度空間中的內積，而不需要顯式計算出映射的結果。
5. **軟邊界（Soft Margin）**：在實際應用中，數據通常是線性不可分的，或者可能存在噪音。為了處理這些情況，SVM引入了軟邊界概念，允許一些數據點不完全符合分類要求，這些數據點的損失被添加到模型的目標函數中，同時通過一個參數來平衡邊界的硬度和分類的準確性。

SVM被廣泛應用於許多領域，包括文本分類、圖像分類、生物信息學等。它的優點包括對於高維度數據的適應能力、對於較小的訓練集也能夠很好地工作、以及對於非線性問題的處理能力。然而，對於大型數據集，SVM的訓練時間可能會比較長，且對於參數的選擇也較為敏感，時間複雜度為$O(n^2)$，需要仔細調整以獲得最佳性能。

時間序列與時間序列分析

時間序列：

- 是按照時間順序排列的一系列數據點，通常是按照固定的時間間隔收集的觀測值或測量值。時間序列可以是一維或多維的，而時間通常是指時間戳，如日期、時間、時間戳或時間序列的編號。時間序列可用於描述和分析各種現象的變化，如股票價格、氣象變化、經濟指標、交通流量、感測器數據等。

時間序列分析：

- 是對時間序列數據進行統計分析、建模和預測的過程。時間序列分析的目標發現數據中的模式、趨勢和周期性，以及對未來數據的預測。

該分析通常包括以下幾個主要方面：

1. 數據探索和可視化：對時間序列數據進行可視化和摘要統計，包括繪製時間序列圖、直方圖、自相關圖等，以了解數據的基本特徵。
2. 模型擬合：建立數學模型來描述時間序列中的趨勢、季節性、周期性和隨機波動等特徵。常見的模型包括移動平均模型（MA）、自回歸模型（AR）、自回歸移動平均模型（ARMA）、自回歸整合移動平均模型（ARIMA）等。
3. 模型評估：通過檢查模型的殘差、自相關函數、預測準確度等指標來評估模型的適合性和預測能力。

4. 預測：基於已有的時間序列數據，使用擬合的模型來預測未來的數據點。預測方法可以包括簡單的滑動平均、指數平滑、ARIMA模型等。
5. 模型優化：根據模型評估的結果，進行模型參數的調整和優化，以提高模型的準確性和魯棒性。

混淆矩陣（Confusion Matrix）

混淆矩陣（Confusion Matrix）是用於評估分類模型性能的一種矩陣表示方法，它將模型的預測結果與真實標籤進行比較，以顯示模型在不同類別上的預測情況。

混淆矩陣通常是一個 N×N 的矩陣，其中 N 表示類別的數量。在二元分類問題中，混淆矩陣是一個 2×2 的矩陣，包含了以下四個項目：

真陽性（True Positive，TP）：模型正確地將正類樣本預測為正類。
假陰性（False Negative，FN）：模型將正類樣本錯誤地預測為負類。
假陽性（False Positive，FP）：模型將負類樣本錯誤地預測為正類。
真陰性（True Negative，TN）：模型正確地將負類樣本預測為負類。

混淆矩陣能夠提供豐富的信息，幫助我們評估分類模型的性能。基於混淆矩陣，可以計算出多個性能指標，例如準確度（Accuracy）、精度（Precision）、召回率（Recall）、F1分數等，進而評估模型的表現。

為什麼需要混淆矩陣？因為單一的分類準確率不能提供模型的完整性能評估。混淆矩陣能夠提供更全面的信息，例如模型在不同類別上的表現，以及對於不同類別的錯誤分類情況。透過混淆矩陣，我們可以更好地理解模型的優點和缺點，進而進行更精確的模型改進和優化。

主成分分析（Principal Component Analysis，PCA）

主成分分析（Principal Component Analysis，PCA）是一種常見的降維技術，用於將高維數據轉換為低維表示，同時保留數據集中的大部分變異性。

以下是主成分分析的基本過程：

1. **標準化數據：**對原始數據進行標準化處理，使每個特徵具有相同的尺度。(這是因為PCA是基於數據的協方差矩陣進行計算的，需要確保每個特徵的尺度一致。)
2. **計算協方差矩陣：**計算標準化後數據的協方差矩陣。協方差矩陣反映了數據特徵之間的相關性和變異性。

計算標準化後的數據 $\bar{X}$ 的協方差矩陣 Σ。假設 $\bar{X}$ 的維度為 $n \times m$，則協方差矩陣的公式如下：

$$\Sigma = \frac{1}{m}\bar{X}\bar{X}^T$$

3. **計算特徵值和特徵向量**：通過對協方差矩陣進行特徵值分解或奇異值分解，得到其特徵值和對應的特徵向量。特徵值表示相應特徵向量方向上的變異程度。
   通過對協方差矩陣 Σ 進行特徵值分解（或奇異值分解），得到特徵值 λ_i 和對應的特徵向量 v_i。特徵值和特徵向量滿足以下公式：

$$\Sigma v_i = \lambda_i v_i$$

4. **選取主成分**：按照特徵值的大小，選取前 k 個大特徵值對應的特徵向量作為主成分，這些主成分涵蓋了數據中大部分的變異性。
5. **數據轉換**：將數據與所選主成分進行矩陣相乘來將原始數據投影到所選的主成分上，得到新的低維表示。
   將原始數據集 X 投影到所選的主成分上，得到新的低維表示 Y。假設所選的前 k 個特徵向量組成的矩陣為 V_k，則數據轉換的公式如下：

$$Y = V_k^T X$$

通常，PCA的應用有以下幾個方面：

- **數據壓縮**：將高維數據轉換為低維表示，減少存儲空間和計算成本。
- **特徵提取**：識別出數據中最重要的特徵，並提取重要的特徵們進行後續的分析。
- **可視化**：將高維數據可視化到二維或三維空間，以便更直觀地理解數據的分布和結構。

值得注意的是，PCA的一個重要限制是它是一種線性降維技術，可能無法有效處理非線性結構的數據。在這種情況下，可以考慮使用核PCA等非線性降維方法。PCA通常用於數據壓縮、可視化、特徵提取等應用中，並且在許多領域，如機器學習、模式識別、數據分析等中都有廣泛的應用。

成分輪換

成分輪換在主成分分析（PCA）中是非常重要的，因為它可以改變主成分的解釋性和可解釋性，進而影響到PCA降維後的特徵表示效果。總的來說，成分輪換對PCA非常重要。

以下是成分輪換對PCA的重要性：

- 解釋性：成分輪換可以改變主成分的順序和解釋方向，使得主成分的解釋性更好。通常，第一個主成分（第一軸）解釋最大的變異性，第二個主成分（第二軸）解釋次大的變異性，依此類推。通過成分輪換，我們可以確保主成分的順序和解釋方向符合我們的期望，使得降維後的特徵表示更容易理解和解釋。
- 可解釋性：成分輪換可以使得主成分的特徵向量更容易解釋。在原始特徵空間中，主成分的特徵向量是原始特徵的線性組合，因此往往不容易解釋。通過成分輪換，我們可以將主成分轉換到一個新的特徵空間中，使得主成分的特徵向量更加直觀和易於理解，進而提高模型的可解釋性。
- 視覺化：成分輪換可以使得主成分的視覺化更容易。將主成分轉換到新的特徵空間後，我們可以更容易地將其可視化到二維或三維空間中，以便進行視覺化分析和探索。
- 分類和回歸：成分輪換可以改善PCA特徵表示的性能，進而提高分類和回歸等機器學習任務的效果。通過成分輪換，我們可以找到最優的主成分表示，使得數據的分佈和分類邊界更加清晰和可區分。

交叉驗證（Cross-Validation）

交叉驗證（Cross-Validation）是一種用於評估模型性能和選擇超參數的常用技術。 它透過將資料集劃分為多個子集，然後進行多次訓練和驗證，以評估模型的泛化能力。特別是在資料集較小或模型複雜度較高時更為重要

常見的交叉驗證方法包括 K 折交叉驗證（K-Fold Cross-Validation）、留一交叉驗證（Leave-One-Out Cross-Validation，LOOCV）和隨機分割交叉驗證（Random Split Cross-Validation）等。

以下是 K 折交叉驗證的基本步驟：

1. 將資料集隨機分成 K 個大小相似的子集，其中一個子集作為驗證集，其餘 K-1 個子集作為訓練集，得到一個模型效能的評估指標。。
2. 重複上述步驟 K 次，每次選擇一個不同的驗證集，得到 K 個模型效能評估指標。
3. 對 K 個評估指標取平均值作為模型最終的效能評估。

交叉驗證的優點包括：

1. 充分利用數據：交叉驗證能夠充分利用數據，每個樣本都被用於訓練和驗證，提高了數據的使用率。

2. 模型泛化能力的評估更準確：透過多次訓練和驗證，降低樣本劃分不合理導致的誤差。可以更準確地評估模型的泛化能力。
3. 選擇超參數更合理：透過交叉驗證，可以在一定程度上避免模型對特定資料集的過度擬合，從而更合理地選擇超參數。

ROC curve 與 AUC（Area Under the ROC Curve）

1. ROC 曲線（Receiver Operating Characteristic Curve）是一種用於評估分類模型效能的圖形工具。它顯示了二元分類模型在不同閾值下的真陽性率（True Positive Rate，又稱為召回率）與假陽性率（False Positive Rate）之間的關係。 ROC 曲線的橫軸是假陽性率（FPR），縱軸是真陽性率（TPR）。
2. AUC（Area Under the ROC Curve）是ROC 曲線下的面積，它是一個單一的度量來評估二元分類模型的效能。 AUC 的取值範圍通常在0 到1 之間，AUC 值越接近1，表示模型的表現越好，能更好地將正例和負例區分開。如果AUC 等於0.5，表示模型表現與隨機猜測相當；如果AUC 小於0.5，表示模型表現不佳，甚至比隨機猜測還差。

簡而言之，ROC 曲線和AUC 是評估二元分類模型效能的重要工具，ROC 曲線透過繪製真陽性率和假陽性率之間的關係，直觀地展示了模型在不同閾值下的表現；而AUC 提供了一個單一的指標來量化模型的效能，方便比較不同模型的優劣。

F1 分數、召回率和準確率

F1 分數、召回率和準確率是評估分類模型表現的常用指標

1. 準確率（Accuracy）：準確率是指分類模型在所有樣本中正確分類的比例，即正確預測的樣本數除以總樣本數。

   $$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$$

   公式：其中，TP 表示真例（True Positive），TN 表示真負例（True Negative），FP 表示假正例（False Positive），FN 表示假負例（False Negative）。

2. 召回率（Recall，也稱為真正例率）： 召回率是指在所有真實陽性例子中，模型成功預測為陽性的比例，即真正例的數量除以真實正例的總數量。召回率越高，越不會遺漏陽姓，但是可能誤報。公式：

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

3. 精度（Precision）：預測為陽性的樣本中，真正陽性的比例，精確度越高，模型的預測結果越可信。

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$$

4. F1 分數：F1 分數是綜合考慮了準確率和召回率，是準確率和召回率的調和平均數。公式：

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$$

在這些指標中，準確率衡量了模型整體的正確分類程度，召回率衡量了模型對正例的識別能力，F1 分數綜合考慮了模型的精確率和召回率，是一個綜合評估指標。 在不同的場景中，根據需求可以選擇合適的評估指標來評估模型的效能。

召回率為主的任務：

在涉及重要資訊遺漏（False Negative）對任務結果影響較大的情況下，通常會更關注召回率，不希望遺漏、漏報。

1. 醫學診斷：如果某個疾病很嚴重，漏診（即將患者診斷為健康，但其實患有該疾病）會導致嚴重後果，此時需要確保盡可能多地診斷出真正患病的病人，即需要較高的召回率。
2. 垃圾郵件偵測：將正常郵件誤識別為垃圾郵件（漏報）可能會導致使用者錯失重要訊息，因此需要確保盡可能地將垃圾郵件偵測出來。

精度為主的任務：

在涉及誤報（False Positive）對任務結果影響較大的情況下，通常更關注精確度。希望模型的預測結果越可信。

1. 金融詐欺偵測：誤將正常交易識別為詐欺交易可能導致客戶信任度下降，因此需要確保盡可能減少誤報，即需要較高的精確度。
2. 網路安全：將正常用戶誤識別為惡意攻擊者可能導致正常用戶被誤攔截或拒絕服務，因此需要確保盡可能減少誤報。

假設檢驗（Hypothesis Testing）

假設檢驗（Hypothesis Testing）是統計學中的一種方法，用於檢驗關於一個或多個母體參數的假設是否成立。在假設檢驗中，我們提出一個關於母體參數的假設（稱為原假設），然後利用樣本數據來判斷這個假設是否在統計上是合理的。

假設檢驗的一般步驟包括：

1. 建立假設：提出原假設（null hypothesis，簡稱H0）和對立假設（alternative hypothesis，簡稱H1），通常原假設是我們要進行檢驗的假設，而對立假設是相對的假設，它描述了我們要觀察的效果。
2. 選擇檢驗統計量：選擇一個合適的統計量（如平均值、比例、變異數等），用來檢驗假設。
3. 設定顯著性水準：設定顯著性水準（significance level，通常用α表示），這是一個閾值，用來評估結果是否顯著。常見的顯著性水準包括0.05、0.01等。
4. 計算統計量的值：根據樣本數據計算出所選統計量的值。
5. 假設檢驗：利用統計理論計算原假設成立下，得到該統計量或更極端情況的機會（p值）。若p值小於設定的顯著性水準，我們將拒絕原假設，否則我們接受原假設。
6. 作出結論：根據檢驗結果，對原假設進行判斷，進行結論。

舉個例子，假設一個藥物公司開發了一種新藥，聲稱這種藥物的平均治癒時間短於傳統治療方法的平均治癒時間。我們可以提出兩個假設：

1. 原假設（H0）：新藥的平均治癒時間與傳統治療方法的平均治癒時間相同。
2. 對立假設（H1）：新藥的平均治癒時間短於傳統治療方法的平均治癒時間。

然後，我們可以收集一組使用新藥治療的患者的數據，計算其平均治癒時間。接下來，我們可以使用假設檢驗來判斷這組數據是否支持或反駁原假設。如果假設檢驗的結果顯示，p值小於預先設定的顯著性水準，則我們可以拒絕原假設，接受對立假設，即新藥的效果顯著優於傳統治療方法。

方差膨脹因子（Variance Inflation Factor，VIF）

方差膨脹因子（Variance Inflation Factor，VIF）是一種統計量，用於評估多重共線性（Multicollinearity），通常用於多變量回歸分析中。

多重共線性：指的是在回歸模型中，自變量之間存在高度相關性的情況，這會導致模型參數估計的不穩定性，使得模型的解釋性下降。VIF的作用就是檢測這種自變量之間的高度相關性。

VIF的計算方式是通過將每個自變量作為因變量，其他自變量作為自變量，來擬合回歸模型，然後計算每個自變量的方差膨脹因子。方差膨脹因子計算每個自變量的方差因子，即該自變量的方差與其他自變量相比膨脹了多少倍。如果方差膨脹因子大於某個閾值（通常是10），則認為存在多重共線性。

VIF 的公式如下所示：

$$\text{VIF}_j = \frac{1}{1-R_j^2}$$

VIF_j 是自變量 j 的方差膨脹因子，R_j^2 是將自變量 j 作為因變量，其他自變量作為自變量擬合的回歸模型的決定係數。

VIF 越大，表示自變量之間的相關性越高，存在的多重共線性問題也就越嚴重。因此，通過檢測 VIF，可以評估模型中自變量之間的相關性，並進行必要的修正，以提高模型的穩定性和解釋性。

協同過濾（Collaborative Filtering）

協同過濾（Collaborative Filtering）和基於內容的過濾是推薦系統中常見的兩種方法，用於根據用戶的行為歷史或物品的屬性來預測用戶對物品的喜好。

協同過濾（Collaborative Filtering）：

- 協同過濾是一種根據用戶與物品之間的相似性來進行推薦的方法。它不依賴於物品的內容特徵，而是通過分析用戶對物品的行為（如評分、點擊、購買等）來計算用戶和物品之間的相似性。

協同過濾分為兩種主要類型：

- 基於用戶的協同過濾：該方法通過比較用戶之間的相似性來進行推薦。例如，如果兩個用戶在過去喜歡相似物品，那麼他們可能對其他相似物品也有相似的喜好。
- 基於物品的協同過濾：該方法通過比較物品之間的相似性來進行推薦。例如，如果一個用戶喜歡某個物品，那麼他可能也會喜歡與該物品相似的其他物品。

基於內容的過濾（Content-based Filtering）：

- 基於內容的過濾是一種根據物品的屬性特徵來進行推薦的方法。它利用物品的描述信息或屬性，如關鍵詞、類別、特徵向量等，來衡量物品之間的相似性，然後根據用戶已喜歡的物品的特徵向量來推薦相似的物品。
- 例如，如果一個用戶喜歡某個電影，那麼基於內容的過濾可以通過比較該電影的類型、演員、導演等屬性來推薦其他類似類型或相同演員、導演的電影。

主要區別：

協同過濾主要關注用戶與物品之間的相似性，並且不需要事先對物品進行特徵提取或分類。
基於內容的過濾則主要基於物品的屬性特徵進行推薦，需要事先對物品進行特徵提取和分類。
兩種方法各有優缺點，協同過濾可以捕捉到用戶間的隱式相似性，但對於冷啟動問題（新物品或新用戶）較為敏感；而基於內容的過濾則能夠提供更具解釋性的推薦，但可能會忽略用戶的隱式偏好。通常，推薦系統會結合這兩種方法來獲得更準確和全面的推薦結果。

統計學中，自助法（Bootstrap Method，Bootstrapping，或自助抽樣法、拔靴法）是一種從給定訓練集中有放回的均勻抽樣

統計上的顯著性（Statistical Significance）

統計上的顯著性（Statistical Significance）是指一個統計結果在統計學上是否具有實際意義或者是否不僅僅是由於隨機變動而導致的。在假設檢驗中，顯著性水準（Significance Level）通常用來衡量統計結果的顯著性。

具體來說，如果結果在顯著性水準（通常用α表示）下是顯著的，則我們可以拒絕原假設，接受對立假設；反之，如果結果不顯著，則我們無法拒絕原假設。通常，顯著性水準被設定為0.05或0.01，這意味著我們對犯錯的風險分別為5%或1%。

在假設檢驗中，顯著性水準是用來評估觀察到的統計量或效應是否可能是由於隨機變動而產生的。如果p值（p-value）小於顯著性水準，則我們認為結果是顯著的，我們有足夠的證據拒絕原假設；反之，如果p值大於顯著性水準，則我們無法拒絕原假設，結果不被認為是顯著的。

簡而言之，統計上的顯著性是指我們是否有足夠的證據來支持我們的研究假設或者結論，以區分結果是真實存在的還是僅僅是由於隨機變動引起的。

歸納學習與演繹學習

歸納學習（Inductive Learning）和演繹學習（Deductive Learning），兩種不同的學習方法，它們在推理方式和邏輯思考上有所不同：。

歸納學習：歸納學習是從特定的例子或資料推斷出一般性規律或模式的過程。 它透過觀察現實世界中的資料來推理出普遍性規則、模式或模型，具有一定的概括性和泛化能力。(具體到抽象的推理過程)

演繹學習：演繹學習是利用現有的知識或規則，利用邏輯推理的方法來推導出新的結論或解決具體的問題。涉及對知識和規則的推理。

總的來說，歸納學習強調從具體到抽象的推理過程，透過觀察和總結現實世界中的資料來學習；而演繹學習強調從一般到特殊的推理過程，透過已有的知識或規則來推導出新的結論或解決問題。

情境題：

假設模型存在顯著差異。 您認為哪種算法最適合處理這種情況。

當發現機器學習模型存在顯著差異時，可以嘗試使用集成學習演算法來處理這種情況。 整合學習透過結合多個基本模型的預測結果，以提高模型的性能和穩定性，從而減少模型的差異性。 以下是一些常用的集成學習演算法：

1. **隨機森林（Random Forest）**：隨機森林是一種基於決策樹的整合學習演算法，它透過隨機選擇特徵和樣本來建立多棵決策樹，並透過投票或平均的方式來整合這些決策樹的預測結果。
2. **梯度提升（Gradient Boosting）**：梯度提升是一種迭代的整合學習演算法，它透過迭代訓練多個弱學習器（通常是決策樹），每一輪都試圖修正前一輪模型的預測誤差，最終將所有弱學習器的預測結果 進行加權求和。
3. **AdaBoost（Adaptive Boosting）自適應增強**：AdaBoost 是一種自適應增強演算法，它透過逐步訓練一系列弱學習器（例如決策樹），並根據前一輪學習器的分類結果來調整每個弱分類器的權重，從而使得後一輪學習器更關注前一輪中被錯誤分類的樣本，最終組合這些弱分類器以生成一個強分類器。
4. **Bagging**：Bagging 是一種平行的整合學習方法，它透過隨機抽樣訓練集來建立多個基本模型，然後透過投票或平均的方式來整合這些模型的預測結果。
5. 以上演算法都能夠有效地降低模型的方差，提高模型的泛化能力，從而減少模型之間的顯著差異。 選擇合適的集成學習演算法取決於具體的問題和數據特點，可以透過交叉驗證等方法來評估不同演算法的性能並選擇最合適的演算法。

哪個更重要：模型性能還是模型精度？

模型效能和模型精確度在不同的情況下可能具有不同的重要性。下面我將討論它們的差異和應用場景：

模型性能：

- 模型表現通常指的是模型在解決特定任務或問題上的整體表現，它可能涉及多個指標，如準確率、召回率、F1 分數、AUC 等。
- 模型效能更適用於全面地評估模型的效果，考慮了模型的泛化能力、穩定性和適應性等因素(多個任務的情況下)。

模型精度：

- 模型精度通常指的是模型在預測或分類任務中的準確性。
- 模型精確度更關注模型的預測準確率，即模型能夠正確預測的樣本類別。
- 模型精確度更適用於簡單的二分類或迴歸問題，或對準確度要求較高的場景。

如果任務要求綜合考慮多個指標或對模型的泛化能力和適應性有較高要求，那麼模型表現可能更重要。
如果任務的關鍵是確保模型的預測準確性，例如在醫療診斷或金融詐欺偵測中，那麼模型精確度可能更為重要。

不平衡資料集

處理不平衡資料集是機器學習中常見的問題，因為在實際應用中，許多情況下正例和負例的數量差異很大。 以下是處理不平衡資料集的一些常見方法：

1. 重新採樣（Resampling）：
   - 過採樣（Oversampling）：增加少數類樣本的數量，使得正例和負例數量接近平衡。 常用的過採樣方法包括隨機複製樣本、SMOTE（Synthetic Minority Over-sampling Technique）:Synthetic Minority Over-sampling Technique - 使用來自少數類的數據切片作為示例，然後生成與其相當的額外人工實例並將其添加到原始數據集中。 此方法適用於數字數據點。。
   - 欠採樣（Undersampling）：減少多數類樣本的數量，縮小多數類的大小以匹配少數類，使得正例和負例數量接近平衡。這有助於提高存儲和運行時執行的速度，但也可能導致有價值數據的丟失。 常用的欠採樣方法包括隨機刪除樣本、ClusterCentroids、NearMiss等。

2. 基於演算法的方法：
   - 使用能夠處理不平衡資料的演算法，例如整合學習中的平衡隨機森林（Balanced Random Forest）和平衡AdaBoost 等，或支援類別權重調整的演算法，如邏輯迴歸中的class_weight 參數、決策樹中的class_weight 參數 等。

4. 使用適合不平衡資料集的損失函數：
   - 使用適合不平衡資料集的損失函數，如 Focal Loss、Weighted Cross-Entropy Loss 等，透過調整損失函數的權重來平衡正負樣本之間的影響。
     模型傾向於偏向於常見類別，而對於罕見類別的預測效果不佳。Focal Loss通過調整損失的權重來解決這個問題。Focal Loss的計算公式如下：

     $$\mathrm{FL}(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

     其中，pt是模型預測的概率，γ 是一個調整參數，通常取值在0到5之間。
     Focal Loss的特點在於它引入了一個平衡因子(1−pt)^γ，這個因子會對預測概率pt 進行調整，當模型對某個樣本的預測概率pt 越小（即模型對該樣本的預測不確定性越大）時，平衡因子的值就會越大，對錯誤的預測給予更高的損失，使得模型更加關注難以分類的樣

本。換句話說，這個平衡因子會使得模型更加關注難以分類的樣本。通過調整 γ 的值，可以控制這個平衡因子的影響程度，從而調整模型對正負類別之間的平衡。

總的來說，Focal Loss通過引入一個平衡因子，有效地解決了類別不平衡問題，使得模型能夠更好地處理罕見類別，提高了模型的性能。

5. 生成合成樣本：
   - 使用生成對抗網路（GANs）或其他生成模型生成合成樣本，以增加少數類樣本的多樣性，使得正例和負例數量接近平衡，從而提高模型的泛化能力。

5. 考慮後處理技術：
   - 在模型預測後，透過後處理技術對預測結果進行調整，例如閾值移動（Threshold Moving）、閾值調整（Threshold Adjustment）等，以達到更好的平衡效果。

在處理不平衡資料集時，通常需要進行實驗比較不同方法的效果，並根據評估指標（如準確率、召回率、F1 分數等）來評估模型的效能。

異常值與雜訊過多，如何解決?

處理機器學習中存在大量異常值和雜訊的資料是一項挑戰性的任務，因為異常值和雜訊可能會影響模型的效能和泛化能力。

異常值檢測：
識別並處理異常值和雜訊。可以使用統計方法（如Z-score（Z值）、箱線圖、標準差等）、視覺化方法（如散點圖、直方圖等）、基於距離的方法（如K 近鄰）、基於密度的方法（如局部離群因子LOF）。

1. Z-score（Z值）方法：
   Z-score是一種統計方法，通過計算數據點與平均值的偏差除以標準差，來評估數據點是否為異常值。一般來說，如果數據點的Z-score超出一定的閾值，則被視為異常值。
2. 箱形圖（Boxplot）：
   箱形圖是一種可視化方法，通過顯示數據的四分位範圍來檢測異常值。通常情況下，超出上下四分位數一定範圍的數據點被認為是異常值。
3. K最近鄰（K-nearest neighbors，KNN）方法：
   KNN方法通過計算每個數據點與其最近的K個鄰居之間的距離來檢測異常值。如果數據點的平均鄰居距離遠於期望值，則被視為異常值。
4. 孤立森林（Isolation Forest）方法：

    a. 孤立森林是一種基於樹模型的方法，通過隨機劃分數據空間來找到異常值。由於異常值通常需要更少的劃分次數才能被隔離，因此孤立森林可以有效地檢測異常值。
5. 局部異常因子（Local Outlier Factor，LOF）方法：
    a. LOF方法是一種基於密度的方法，通過比較每個數據點周圍鄰域的密度來檢測異常值。密度較低的數據點被認為是異常值。
6. One-Class SVM方法：
    a. One-Class SVM是一種支持向量機方法，用於學習正常數據的分佈，從而識別異常值。它將正常數據視為一個類別，並且在超平面之外的數據被視為異常值。

異常值修復：

1. 資料清洗:
   對於雜訊數據，可以考慮刪除、平滑或插值法處理來減少其影響。
2. 特徵選擇和特徵變換：
    - 透過特徵選擇方法來篩選對目標變數影響較大且被異常值和雜訊影響較小的特徵，從而減少異常值和雜訊的影響。
    - 對資料進行特徵變換，如對數轉換、歸一化、標準化等，使資料更符合模型的假設並減少異常值和雜訊的影響。
3. 整合學習：
   使用整合學習方法，如隨機森林、梯度提升樹等，透過多個弱學習器的組合來降低異常值和噪音的影響，對於異常值和噪音的穩健性較強。
4. 模型選擇：
   選擇對異常值和雜訊較為穩健的模型，如支援向量機、決策樹等，這些模型對於異常值和雜訊的影響較小。
   在模型訓練過程中，可以使用特定的損失函數或正規化項來降低異常值和雜訊的影響，例如使用Huber Loss 或L1/L2 正規化。
5. 數據增強：
   對數據進行增強，透過產生合成數據來減少異常值和雜訊的影響。可以使用生成對抗網路（GAN）或基於插值方法來產生合成資料。

數據可視化的重要性

數據可視化可直觀地理解數據的趨勢、模式、關聯性。以下是數據可視化的一些重要性：

1. **幫助直觀理解：**數據可視化可以將抽象的數據轉換為易於理解的圖表或圖像，幫助取有價值的洞察和知識。
2. **發現數據的模式和趨勢：**通過視覺化數據，人們可以發現數據中的模式、趨勢和異常值，從而更好地理解數據的變化和動態，並及時做出相應的決策。

3. **發現關聯性和相互作用：**數據可視化可以幫助人們發現數據中不同變量之間的關聯性和相互作用，從而揭示潛在的因果關係。
4. **幫助決策：**數據可視化提供了直觀和靈活的工具，可以幫助決策者快速理解數據，做出基於數據的決策。通過視覺化，決策者可以更全面地考慮不同的選擇和後果，並更好地解釋和辯護他們的決策。

不同類型的圖表

因為不同類型的數據需要不同的視覺表示方式才能有效地傳達信息。以下是一些常見的數據類型及其對應的適當圖表類型：

1. 單變量數據（Univariate Data）：
    - 直方圖（Histogram）：用於展示單變量數據的分布情況，特別是數值型數據的分布情況。
    - 餅圖（Pie Chart）：用於展示類別數據的相對比例，但需要注意餅圖不適合展示過多類別的數據。
2. 雙變量數據（Bivariate Data）：
    - 散點圖（Scatter Plot）：用於展示兩個變量之間的關係，可以用顏色或形狀區分不同類別的數據點。
    - 熱圖（Heatmap）：用於展示兩個變量之間的相關性，通常用顏色來表示相關性的程度。
3. 時間序列數據（Time Series Data）：
    - 折線圖（Line Chart）：用於展示時間序列數據的趨勢和變化，特別適用於長期趨勢的展示。
    - 面積圖（Area Chart）：與折線圖類似，用於展示時間序列數據的趨勢，但更加強調區域之間的差異。
4. 多變量數據（Multivariate Data）：
    - 散點矩陣圖（Scatterplot Matrix）：用於展示多個變量之間的關係，每個格子內是兩個變量的散點圖。
    - 雷達圖（Radar Chart）：用於展示多個變量之間的相對關係，特別適用於比較多個類別的數據。
5. 地理數據（Geospatial Data）：
    - 地圖（Map）：用於展示地理數據的分布情況，可以使用不同顏色或大小來表示不同的數值或類別。