目次

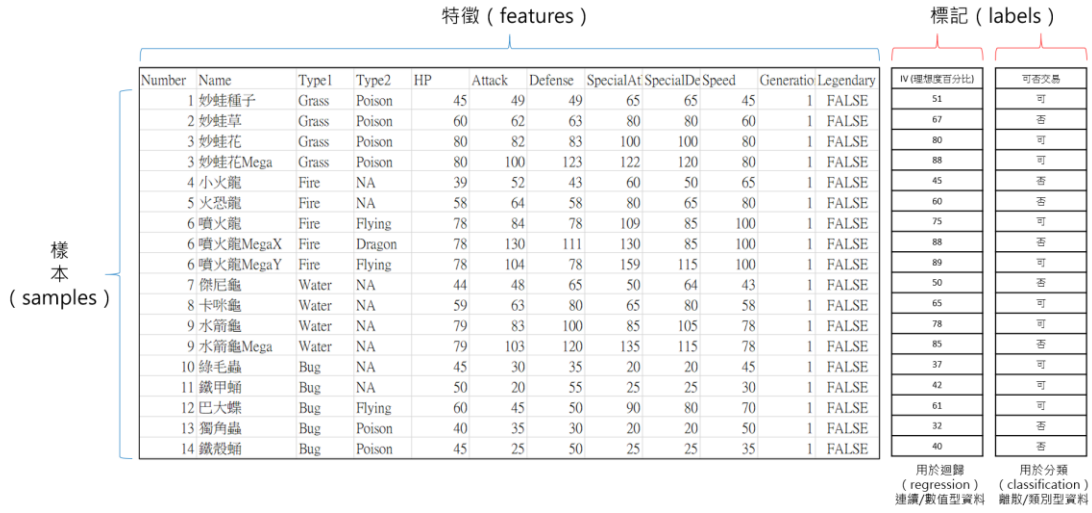
- GitHub 專案連結
  https://github.com/telunyang/python_machine_learning

Module 1. 機器學習的常見概念

能夠從現有的資料（data）當中，不斷地從中學習經驗（experience），找出共通的模式（pattern），以便引導未來的決策。在學習這門知識之前，強烈建議先行研讀統計學（含機率）、微積分以及線性代數，可以較容易地理解模型運作過程，以及它們背後的原理。

資料基本用語

資料（有時稱之為「數據」）的基本用語如下：



Number	Name	Type1	Type2	HP	Attack	Defense	SpecialAt	SpecialDe	Speed	Generatio	Legendary	IV (理想度百分比)	可否交易
1	妙蛙種子	Grass	Poison	45	49	49	65	65	45	1	FALSE	51	可
2	妙蛙草	Grass	Poison	60	62	63	80	80	60	1	FALSE	67	否
3	妙蛙花	Grass	Poison	80	82	83	100	100	80	1	FALSE	80	可
3	妙蛙花Mega	Grass	Poison	80	100	123	122	120	80	1	FALSE	88	可
4	小火龍	Fire	NA	39	52	43	60	50	65	1	FALSE	45	否
5	火恐龍	Fire	NA	58	64	58	80	65	80	1	FALSE	60	否
6	噴火龍	Fire	Flying	78	84	78	109	85	100	1	FALSE	75	可
6	噴火龍MegaX	Fire	Dragon	78	130	111	130	85	100	1	FALSE	88	否
6	噴火龍MegaY	Fire	Flying	78	104	78	159	115	100	1	FALSE	89	可
7	傑尼龜	Water	NA	44	48	65	50	64	43	1	FALSE	50	否
8	卡咪龜	Water	NA	59	63	80	65	80	58	1	FALSE	65	可
9	水箭龜	Water	NA	79	83	100	85	105	78	1	FALSE	78	可
9	水箭龜Mega	Water	NA	79	103	120	135	115	78	1	FALSE	85	否
10	綠毛蟲	Bug	NA	45	30	35	20	20	45	1	FALSE	37	可
11	鐵甲蛹	Bug	NA	50	20	55	25	25	30	1	FALSE	42	可
12	巴大蝶	Bug	Flying	60	45	50	90	80	70	1	FALSE	61	可
13	獨角蟲	Bug	Poison	40	35	30	20	20	50	1	FALSE	32	否
14	鐵殼蛹	Bug	Poison	45	25	50	25	25	35	1	FALSE	40	否

圖：資料的基本用語

註：樣本有時也會被稱為「觀察值」。

機器學習的種類

根據解決的任務類型不同，所採用的機器學習方法，可粗略地分為三種：

- 監督式學習（supervised learning）：
  透過已知的標記（label）來建立模型，可以對未見的樣本資訊，來預測出它可能的答案（預測未見樣本的標記）。比較常見的監督式學習任務是分類（classification）和迴歸（regression）。在分類任務中，程式必須學習從一個或多個特徵來預測一個或多個離散值，例如預測一支股票的價格會「上漲」或「下跌」，或是決定一篇新聞屬於「政治版」還是「娛樂版」等；在迴歸任務中，程式必須學習從一個或多個特徵來預測一個或多個連續型數值，例如預測一個新產品的銷售收入，或是根據一個職務的描述來預測其薪水。

- 非監督式學習（unsupervised learning）：
  在樣本沒有標記或不考慮標記的情況下，透過既有的特徵（feature）來探索資料的結構，進而發現有用的資訊。比較常見的非監督式學習任務，就是在資料集當中，發現互相關聯的樣本，基於一些相似性衡量標準，把樣本放在和其它群組（group）相比，更加類似的群組當中，這樣的過程稱之為分群（cluster），例如在市場中為某項產品發現客戶群體，透過了解特定客戶群體的共同屬性，銷售人員可以決定採用何種銷售策略，用於該客戶群體。
- 強化學習（reinforcement learning）：
  經由與環境互動產生的獎勵（reward）來改善模型的效能，驅使模型朝既定的目標前進。例如下象棋，每吃掉對方一個棋子作為正面獎勵，被吃掉為負面獎勵，根據棋盤狀態以及先前落子的順序，來決定下一步的程動行為（action），透過不斷地反饋與獎勵進行學習。

機器學習的流程

1. 收集資料：
   明確地定義問題，同時收集相關的資料，是機器學習的第一步。資料的完整性對模型的預測效能有著關鍵性的影響，除了資料量要足夠大以外，代表性也很重要。資料的收集夠完整，才能順利進行後續的分析，這部分往往都是最耗時間。
2. 資料前處理：
   真實世界的資料是非常雜亂的，經常會有缺漏、雜訊，或是資料不平衡的情況發生，於是需要進行探索性資料分析（exploratory data analysis，EDA），大致上瀏覽資料的樣式與分佈程度，依需求進行資料清理（data cleaning），將資料整理成特定的格式，也可以透過特徵工程（feature engineering）來抽取、轉換或產生重要的特徵，方便之後的建模工作。
3. 訓練模型與調校：
   在訓練模型之前，會將整個資料集分成三個部分，一個是訓練資料集（training dataset），一個是驗證資料集（validation dataset），一個是測試資料集（testing dataset）。一個完整的資料集，通常訓練資料集佔 50% ~ 75%，測試資料集佔 10% ~ 25%，剩餘的部分，就是驗證資料集。資料切分完成後，依照問題的類型，選擇一個或數個機器學習模型來擬合（fit）訓練資料，藉由擬合過程來學習模型的參數，以建立最終版本的模型。大部分機器學習的模型，都需要使用者預先提供的超參數（hyperparameters），這可以透過調校來取得，有效地發揮模型潛

力。

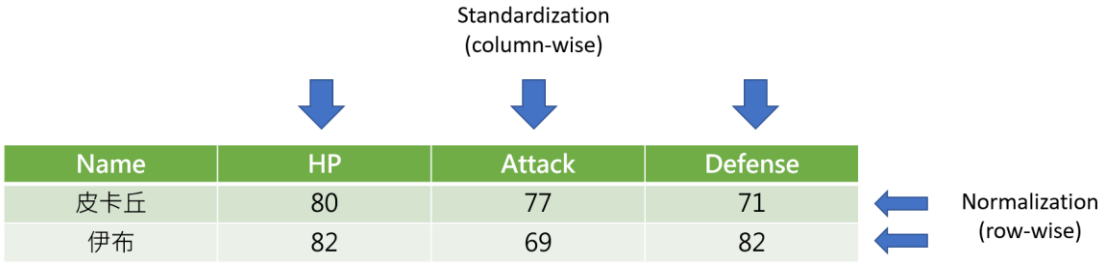
4. 測試模型：
   模型建立完成後，會使用先前切分出來的測試資料集，依照效能量測指標進行測試，來評估（evaluation）模型擬合結果的好壞，再從幾個模型當中挑選相對合適的模型。
5. 改善效能：
   不斷地重複觀察與調整 3、4 步驟的過程來進行改善；必要時，可以重新進行 1、2 步驟，來取得更多有利於資料規律的特徵。

垃圾進，垃圾出（Garbage In, Garbage Out）

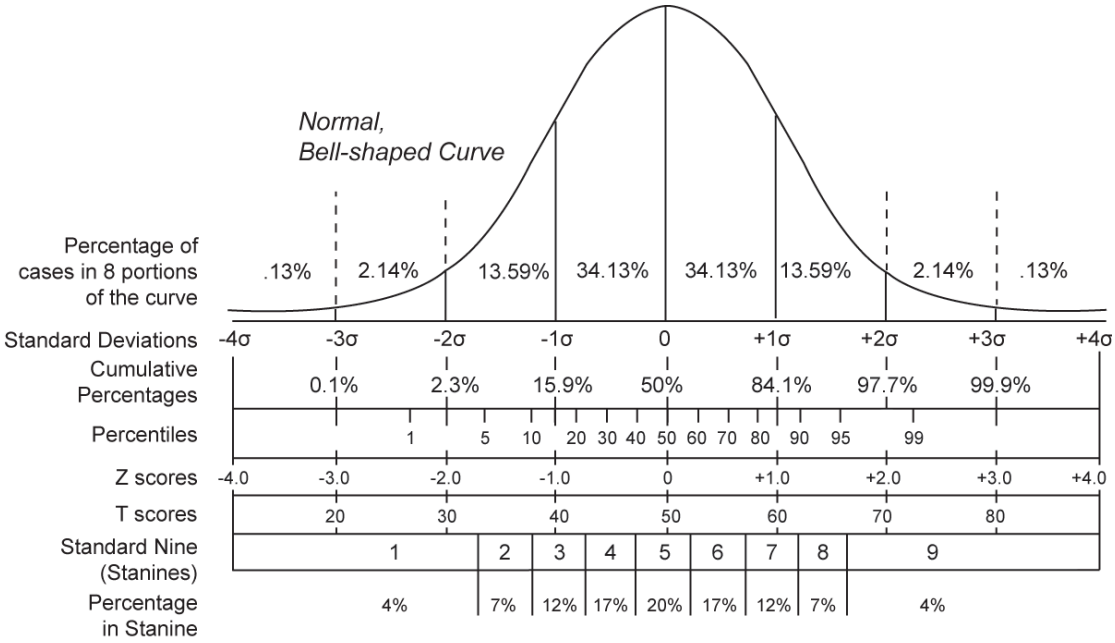
許多模型預測能力的提升，有賴於訓練資料數量的增加。然而，機器學習演算法（algorithm）遵循著格言「Garbage In, Garbage out」，假如一個學生透過閱讀一本錯誤百出、令人困惑的教材來準備考試，他的成績，並不會比閱讀教材篇幅短少，但內容質量較高的學生的成績來得好。在現實世界裡，如果在一個包括噪音（noise）、不相關或是帶有錯誤標記的資料集上進行訓練，其表現不會比一個在更能代表問題的小型資料集上所訓練的模型來得好。

Module 2. 資料前處理

數值型資料



圖：對欄位進行標準化，對樣本進行正規化



圖：取自 https://zh.wikipedia.org/zh-tw/標準分數

StandardScaler

一種常見的標準化方法，用來將資料進行縮放和中心化處理，使資料的平均值（mean）變為 0，標準差（standard deviation）變為 1，大部分的資料會集中在 [-1, 1] 的範圍內，通常大約 68% 的資料點會落在這個範圍。這種方法通常應用於需要使不同特徵保持一致資料範圍的情境，例如在機器學習模型中，確保不同特徵不因為範圍不同而對模型產生不均衡的影響。

平均值為 0、標準差為 1 的資料通常指的是「**標準化**」後的資料，也稱為「**Z-score 標準化**」或「**Z 標準化**」。這是一種常用的資料預處理方法，尤其在機器學習模型訓練中很常見。讓我們深入了解這兩個概念的具體意義。

「平均值為 0」的意思是，資料經過轉換後，其所有資料點的**平均值**被調整到 0。計算平均值的公式是：

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

經過標準化後，資料的整體中心被平移到 0，表示資料的中心點與原點對齊。換句話說，經過標準化後，大部分資料會集中在零附近，資料點大約一半位於零的左邊，另一半位於右邊。

「標準差為 1」的意思是，資料經過標準化後，其**離散程度**（分散程度）調整到 1。標準差衡量資料點相對於平均值的偏離程度，公式為：

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

當標準差為 1 時，資料的離散程度被縮放，使得資料點的變化範圍在 1 個標準差內。也就是說，絕大多數資料點都會落在 0 的附近，並在 -1 到 1 之間波動。

標準化資料的公式如下：

$$Z = \frac{X - \mu}{\sigma}$$

- Z 是標準化後的數值。
- X 是原始數值。
- μ 是資料的原始平均值。
- σ 是資料的原始標準差。

為什麼要標準化？

- **使模型對不同特徵同等重視**：
  如果資料中的不同特徵有著不同的範圍（例如收入用元表示，年齡用年表示），那麼在模型訓練過程中，某些數值較大的特徵可能會過分影響模型。標準化可以解決這個問題，讓所有特徵處於相同的範圍。
- **加速模型收斂**：

某些機器學習演算法（如梯度下降法）在資料標準化後，收斂速度會更快，從而提高訓練效率。

- **有利於距離度量**：
  在需要計算距離的模型（如 KNN 或 SVM）中，標準化確保距離計算不會被數值較大的特徵主導。

注意事項

- **異常值的影響**：
  StandardScaler 使用均值和標準差進行標準化，因此對於包含極端異常值的資料集，這些異常值可能會對均值和標準差產生顯著影響，從而影響資料的標準化效果。如果資料集中存在異常值，應該考慮使用對異常值更為穩健的標準化方法，如 RobustScaler。
- **資料分佈的假設**：
  StandardScaler 假設資料接近常態分佈。在資料分佈非常偏態（skewed）的情況下，標準化後的資料仍然可能存在一些問題，這時可能需要進行額外的資料處理。

StandardScaler 是將資料縮放為均值 0、標準差 1 的標準化方法，特別適合資料接近常態分佈的情況。它有助於機器學習模型的訓練過程，尤其是對於需要統一特徵範圍、提升收斂速度以及確保特徵影響均衡的模型。在資料集不包含異常值且資料接近常態分佈時，StandardScaler 是一個非常有效的標準化工具。

MinMaxScaler

一種常見的資料標準化方法，它將資料縮放到指定的區間（通常是 [0, 1] 之間）。MinMaxScaler 的核心概念是**線性縮放**，即將資料中的最小值轉換為 0，最大值轉換為 1，其餘資料根據其相對位置映射到這個範圍內。

MinMaxScaler 的標準化公式如下：

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- X 是原始資料值。
- X_{min} 是資料集中該特徵的最小值。
- X_{max} 是資料集中該特徵的最大值。

- X_{scaled} 是經過標準化後的資料值。

注意事項

- 異常值的影響：
  MinMaxScaler 對異常值較為敏感，因為它的縮放是基於最大值和最小值。如果資料中有極端值，這些異常值會對縮放範圍產生很大的影響。例如，如果資料中存在非常大的異常值，則其他正常資料可能會被壓縮得非常小。因此，如果資料集中存在異常值，應考慮先去除異常值或使用對異常值更為穩健的標準化方法，如 RobustScaler。
- 資料分佈的假設：
  MinMaxScaler 假設資料的分佈是均勻的。當資料的分佈非常不均勻時（例如，大部分資料集中在某個範圍，而少數資料遠離中心），可能會導致縮放後資料分佈過於壓縮。因此，對於這類資料分佈，應考慮其他標準化技術。

對於每一個特徵，MinMaxScaler 都會找到該特徵的最小值和最大值，然後將這些資料值縮放到指定的區間內。預設的區間是 [0, 1]，但也可以自定義範圍，比如 [-1, 1]。計算過程是將資料點相對於最小值和最大值的位置映射到 0 和 1 之間，從而保證資料不超出這個範圍。

RobustScaler

一種資料標準化的方法，它的目的是對資料進行縮放和中心化處理，但與傳統的標準化（如 StandardScaler）不同，RobustScaler 對異常值（outliers）的影響較小。這使得它特別適合處理具有極端數值的資料集。

核心思想是使用中位數（median）和四分位距（IQR，Interquartile Range）來進行縮放，這與 StandardScaler 使用的均值和標準差不同。

- 中位數：是資料集中間的數字，對極端資料點（異常值）不敏感。
- 四分位距（IQR）：是上四分位數（75% 分位數）與下四分位數（25% 分位數）之間的距離。它是資料分布範圍的一個穩健度量，同樣對異常值的影響較小。

RobustScaler 的標準化公式是：

$$X_{scaled} = \frac{X - Q2}{Q3 - Q1}$$

- X 是原始資料。
- Q_2 是中位數。
- Q_3 是第 75 個百分位數（上四分位數）。
- Q_1是第 25 個百分位數（下四分位數）。
- $Q_3 - Q_1$ 是四分位距。

為什麼使用 RobustScaler？

- 對異常值（outliers）的抗性：
  傳統的標準化方法，如 StandardScaler，是基於均值和標準差，這意味著資料集中存在的異常值會極大地影響均值和標準差，從而影響資料的標準化效果。而 RobustScaler 基於中位數和四分位距進行縮放，這兩個指標不受極端值的影響，因此能夠更加穩健地處理資料中的異常值。
- 適合具有異常值的資料集：
  如果資料集中存在極端值，使用 StandardScaler 會導致縮放後的資料不均衡，影響模型的表現。而 RobustScaler 可以有效減少這些極端值對資料標準化的影響，保證資料縮放後更加穩定。

RobustScaler 的應用場景

- 帶有極端資料點的資料集：
  當你的資料集包含一些異常值，而這些異常值可能會影響模型性能時，RobustScaler 是一個很好的選擇。
- 資料範圍變動較大：
  如果你的資料中存在大量分散且範圍寬廣的資料，RobustScaler 能夠在保持穩定性的同時，進行合理的縮放。

Normalizer

主要目的是將每個資料點（通常是向量）進行向量範數正規化，使得每個資料點的向量長度（即範數）為 1。與其他標準化方法（如 StandardScaler 或 MinMaxScaler）不同，Normalizer 是針對每一個資料點（樣本）進行處理，而不是針對特徵（行）。

Normalizer 的概念

Normalizer 通過將每個資料點縮放，使得該資料點的範數（norm）變為 1，這樣每個樣本都被縮放到單位範圍內。常用的範數有：

- L2 範數（歐幾里得範數）：也稱為平方和範數，通常用於標準的向量正規化，表示向量的平方和的平方根為 1。
- L1 範數（絕對值範數）：表示向量的元素絕對值之和為 1。
- Max 範數：即取向量中的最大絕對值進行縮放，使其最大值為 1。

對於一個資料點（向量）$X = [x_1, x_2, \ldots, x_n]$，L2 範數的正規化公式為：

$$X_{normalized} = \frac{X}{\|X\|_2} = \frac{X}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}}$$

這意味著，資料點中的每個元素都會除以該資料點的 L2 範數，使得結果向量的總長度為 1。

Normalizer 的工作原理

- 作用對象是資料點（樣本）：
  Normalizer 是針對每個資料點進行處理的，不像其他標準化方法那樣針對整個特徵列。每個資料點都被縮放，使其向量範數為 1。
- 使用範數進行縮放：
  Normalizer 根據指定的範數（L1、L2 或 Max）將每個樣本進行縮放，使得資料的整體方向保持不變，但範數為 1。
- 保持資料方向：
  Normalizer 只改變資料的範數，保持資料的方向。這意味著原資料中兩個資料點之間的方向關係不變，只是資料點的長度被縮放到單位範圍內。

什麼時候使用 Normalizer？

- 稀疏資料：
  Normalizer 常用於稀疏資料集，如文本資料或詞袋模型中的詞頻資料。當你有一組向量，並且更關心資料的方向或模式而不是其大小時，Normalizer 是理想選擇。
- 保持資料的幾何形狀：
  如果你的模型（例如基於距離的模型，如 KNN 或餘弦相似度）更關心向量之間的相對幾何形狀（方向）而不是數值的絕對大小，Normalizer 能幫助統一向量的大小。
- 特徵向量的處理：

在處理特徵向量時，尤其是高維度資料如圖像特徵、文本向量表示時，使用 Normalizer 來使每個資料點的向量保持一致的長度，可以讓模型更關注於特徵之間的相對關係，而不是數值本身的大小。

注意事項

- 範數的選擇：
  - 默認的範數是 L2，但在某些應用中，如當資料具有稀疏性時，L1 範數有時會更合適，因為它強調非零特徵的重要性。
  - Max 範數對於有極端值的資料可能更適合，因為它只考慮最大值來進行縮放。
- 與其他標準化技術的區別：
  - Normalizer 是針對每個樣本進行正規化，而其他標準化技術（如 StandardScaler 和 MinMaxScaler）則是針對每個特徵列進行處理。
  - Normalizer 保證了每個樣本向量的範數相同，但它不會改變資料點之間的相對大小（與標準化特徵的技術不同）。

Normalizer 是一種針對向量進行範數正規化的技術，主要用於保持資料點的方向一致，並使每個資料點的範數變為 1。它常用於需要保持資料相對幾何關係的場景，特別是當我們更關心資料點之間的方向或相似度時。

類別型資料

血型	獨熱編碼	類別 ID	二進位編碼
A	1 0 0 0	0	0 0
B	0 1 0 0	1	0 1
AB	0 0 1 0	2	1 0
O	0 0 0 1	3	1 1

圖：樣本資訊透過 1 和 0 組合而成的稀疏向量作為表示

名目特徵（Nominal Features）

名目特徵是沒有內在順序的類別型數據。例如，血型就是名目特徵，因為「A 型」、「B 型」、「AB 型」、「O 型」之間沒有大小、次序的關係。這些值只是表示不同的類別而已。

有序特徵（Ordinal Features）

有序特徵則是帶有內在順序的類別型數據。例如，如果我們在某個場景中有評分等級（如「差」、「中」、「好」），這些數值是有順序的。類別型資料圖片中的數據似乎不涉及有序特徵，因為血型沒有順序關係。

獨熱編碼（One-Hot Encoding）

獨熱編碼是一種常用來將類別型數據轉換為數值的方法。對於每一個類別，獨熱編碼會創建一個二進制向量，在對應類別的位置上設置為 1，其他位置設置為 0。例如，血型「A」的獨熱編碼是 1 0 0 0，而「B」的獨熱編碼是 0 1 0 0。

獨熱編碼的缺點：

- 高維度問題：
  如果類別數非常多，獨熱編碼會導致數據的維度急劇增加，引發「維度災難」（curse of dimensionality），這會帶來內存和計算資源的消耗問題。
- 稀疏矩陣：
  獨熱編碼產生的矩陣通常是稀疏的（大多數元素為 0），在機器學習模型中處理稀疏矩陣可能不夠高效。

二進制編碼（Binary Encoding）

二進制編碼將類別轉換成整數，然後將這些整數轉換為二進制表示。例如血型 A 對應的二進制是 00，血型 B 對應的是 01，血型 AB 是 10，血型 O 是 11。這種編碼方式可以有效地減少維度，因為我們用更少的位數表示類別（對比獨熱編碼的高維度）。

獨熱編碼和二進制編碼各有其適用場景：

- 獨熱編碼：
  適合於類別數目較少的數據，並且適合模型能處理高維度的情況。
- 二進制編碼：

適合於類別數量龐大的場景，因為它可以大大減少編碼維度。

選擇使用哪種編碼方式取決於數據的性質和模型的需求，獨熱編碼容易理解且適用範圍廣，但當類別數較多時，它的維度會變得過高。

LabelBinarizer

專門用來將類別型數據轉換成二進制表示。它常用於對類別型標籤進行處理，尤其是在多分類問題中。這個方法與獨熱編碼（One-Hot Encoding）類似，但也可以進行多重標籤（multi-label）的二進制化處理。

LabelBinarizer 的功能

- 將多類別標籤轉換為二進制數組。
- 在二分類情況下，則只返回一個二進制列。

主要特點：

- 二分類模式：
  如果只有兩個類別，LabelBinarizer 會將數據轉換為單列的二進制數組。例如，對於標籤 ['A', 'B']，它會輸出 [0, 1] 或 [1, 0]。
- 多類別模式：
  對於多個類別（如三個或更多），它會產生與獨熱編碼類似的結果，但不同於獨熱編碼，LabelBinarizer 更加靈活，特別是在處理多標籤問題時。
- 適用於多標籤數據：
  LabelBinarizer 也支持將多標籤數據轉換為二進制表示。這與多標籤分類有關（比如一個樣本可以屬於多個類別），在這種情況下，它可以將多標籤數據表示成一個二維二進制矩陣。

LabelBinarizer 的優點：

- 簡單易用：它能夠簡化標籤轉換過程，尤其在進行分類任務時。
- 靈活性高：能夠處理多標籤問題，並根據不同的需求調整輸出格式。

LabelBinarizer 是一個簡便的工具，可以將類別標籤轉換為二進制格式，尤其適合在需要對標籤進行二進制化的情況下使用，並且它可以處理多標籤和多分類的數據，讓數據更容易被模型所接受。

LabelEncoder

專門用來將類別型數據轉換成數字標籤，它將每個類別標籤映射到一個唯一的整數值。

LabelEncoder 的功能

- 將類別型標籤（如文字標籤）轉換成對應的數字編碼。
- 適合於那些具有名目特徵的類別型數據。

LabelEncoder 的主要用途是處理類別標籤，使得它們能夠被大多數機器學習模型直接接受，因為大部分模型只接受數值型數據，而不是文字標籤。

工作原理

例如，假設你有一組類別標籤 ['A', 'B', 'C']，LabelEncoder 會將這些類別標籤映射到對應的整數，如：

A -> 0
B -> 1
C -> 2

主要特點

- 單列數據：
  LabelEncoder 是針對單一特徵進行轉換的，並且它返回的是整數標籤（而不是二進制編碼）。
- 適合名目特徵：
  LabelEncoder 適用於那些沒有內在順序的類別（如顏色、血型、性別等），但是其編碼結果的數字並沒有順序關係。
- 反轉編碼：
  使用 LabelEncoder.inverse_transform() 方法，你可以將編碼過的數字重新轉換回原本的類別標籤。

遺漏值（又稱缺失值，missing value）

處理真實數據時常會遇到資料不完整或資料遺漏的情況，遺漏值除了容易造成分析結果的偏誤外，也難以直接交由機器學習模型來擬合。

SimpleImputer

用來處理數據中的缺失值。當數據集包含有遺漏值（如 NaN、空白或其他缺失標記）時，SimpleImputer 可以自動填補這些缺失值。它提供了多種填補策略，以確保數據完整性，從而能夠進行後續的分析或建模。常見的填補策略包括：

- 平均值填補（mean）：
  使用每列的平均值來填補缺失值。
- 中位數填補（median）：
  使用每列的中位數來填補缺失值，適合處理偏態較大的數據。
- 最常出現值填補（most_frequent）：
  用每列中最常見的值來填補缺失值。
- 常數填補（constant）：
  可以指定一個固定的值來填補缺失值。

KNNImputer

一個用於填補缺失值的類別，它基於 K 近鄰（K-Nearest Neighbors, KNN）演算法來估算和填補缺失的數據。與簡單的填補策略（如使用平均值或中位數）不同，KNNImputer 是通過考慮數據集中與缺失值最接近的觀測點來進行更智能的填補。具體來說，它會基於其他相似樣本的值來推測缺失值。

KNNImputer 的運作原理：

- 對於每一個缺失值，KNNImputer 會找到數據集中與該行最相似的 K 個樣本（即最接近的鄰居）。
- 它會計算這些鄰居的非缺失值，並使用這些值的平均值來填補缺失的數據點。

這種方法在某些情況下比簡單的填補方法更準確，因為它根據樣本的整體結構來推斷缺失值，而不僅僅依賴單個列的統計值，且數據之間具有相似性時表現較好。例如，對於地理數據或時間序列數據，KNNImputer 可能會比簡單的均值或中位數填補方法更有效。

切割資料集

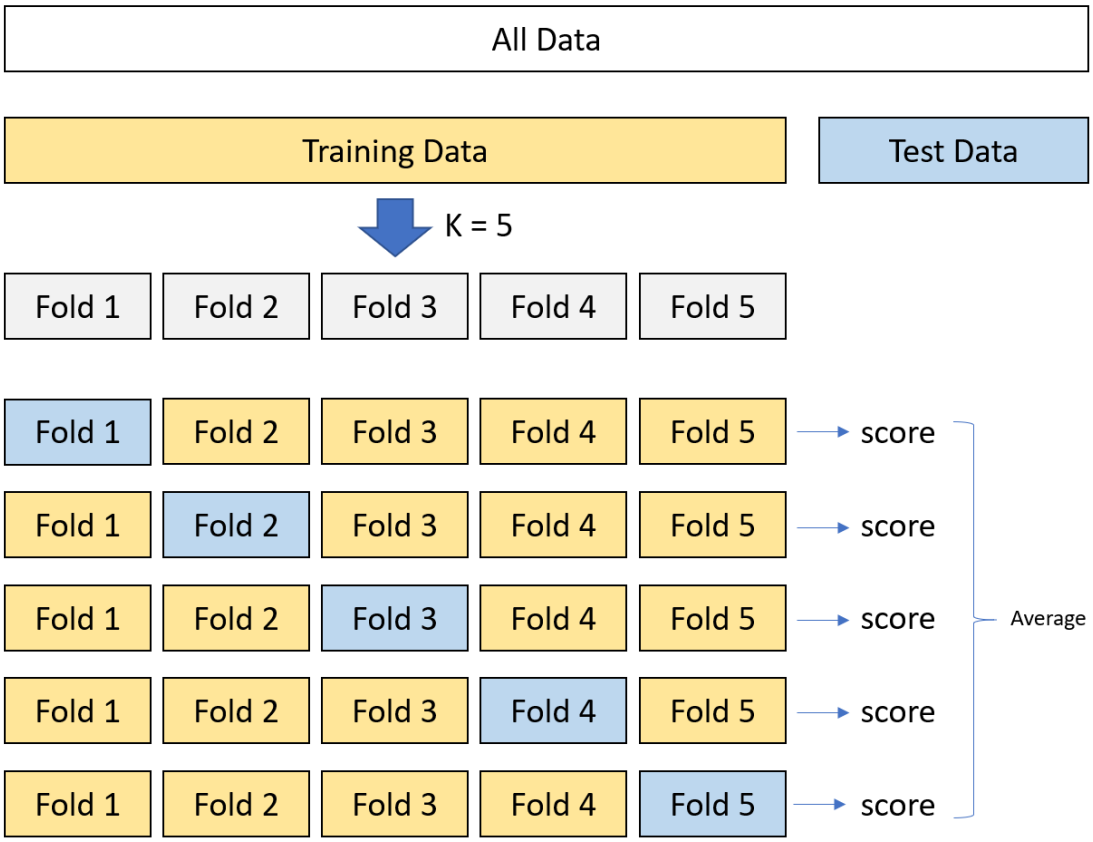
train_test_split

主要目的是將原始數據集劃分為兩部分，一部分用來訓練模型（訓練集），另一部分用來評估模型的表現（測試集）。這樣的分割可以幫助避免模型過度擬合，也能更準確地評估模型在未見數據上的泛化能力。

概念上，train_test_split 做了以下幾件事：

- 訓練集（Training Set）：
  用來訓練模型，讓模型學習數據中的模式或規律。這部分數據是模型在訓練過程中能夠「看到」的。
- 測試集（Test Set）：
  用來評估模型的準確性和泛化能力。測試集不會在模型訓練過程中使用，而是留到訓練完成後進行模型的最終評估。這能幫助檢查模型在未見數據上的表現。

K-fold Cross Validation



圖：交叉驗證，K = 5 的情形

是一種用來評估機器學習模型性能的技術，尤其適合於數據量有限的情況。它將數據集劃分為 K 個子集（folds），並進行多次訓練和測試。具體過程如下：

- 數據分割：
  將整個數據集隨機分為 K 個等份。
- 迭代訓練與測試：
  - 在第 i 次迭代中，選取第 i 個子集作為測試集，其餘的 K-1 個子集作為訓練集。
  - 訓練模型並在該測試集上評估性能（如準確率 Accuracy、F1-score 等）。
- 重複 K 次：
  每次都使用不同的子集作為測試集，並重複這個過程 K 次。
- 計算平均結果：
  最後，將 K 次測試的評估指標平均，作為模型的最終性能評估。

這樣的過程有助於模型的評估更加穩健，因為模型在不同的數據劃分下進行了多次測試，有效減少了因數據集劃分不同而造成的隨機誤差，同時也可以當作參數挑選器（從交叉驗證中找到合適的超參數）。

異常值

實務上，要判斷一個新的觀察值是否屬於已知樣本的分析，例如判斷信用卡交易是否異常，一般稱為異常偵測（anomaly detection），有時候也會異常值為雜訊（noise），機器學習對這一類的訓練樣本特別敏感，容易造成模型擬合後的效能低落。主要偵測方式分成兩種：

- 離群值（outlier）偵測：
  找出在訓練資料集當中，與其它樣本有相當大差異的觀察值，通常是非監督式（unsupervised）的作法。
- 新奇值（novelty）偵測：
  決定一筆新的觀察值是否為離群值，一般為半監督式（semi-supervised）作法（同時使有、無標籤的資料）。

偵測離群值

以下是一些常見的方法，包括 橢圓法（Elliptic Envelope）、四分位數區間（IQR method）、孤立森林（Isolation Forest） 和 局部離群因子（Local Outlier Factor）。這些方法各有特點和適用情境，以下為詳細的說明：

1. 橢圓法（Elliptic Envelope）

- 橢圓法基於 高斯分布 來偵測離群值。這個方法假設資料是多變量的常態分布，並嘗試擬合一個橢圓範圍來包住大部分的數據點。
- 使用時機：適合資料接近高斯分布的情況，例如金融、物理領域的資料，但不適合高度偏態或非高斯分布的資料。
- 優點：
  - 對高斯分布資料有較高的準確性。
  - 可以偵測出在多變量空間中明顯的離群值。
- 缺點：
  - 假設資料必須為高斯分布，對非高斯分布效果較差。
  - 在高維度空間的表現不佳，且需要大量資料來擬合高維橢圓。

2. 四分位數區間法（Interquartile Range, IQR method）

- IQR 方法使用數據的四分位數來定義離群值。通常定義小於 第一四分位數 - 1.5 倍 IQR 或大於 第三四分位數 + 1.5 倍 IQR 的數據點為離群值。
- 使用時機：適合簡單數據集且沒有多變量需求的情境，特別適合一維的資料分布。
- 優點：
  - 簡單易用且無需對數據分布做出假設。
  - 對於非高斯分布或長尾分布的數據效果良好。
- 缺點：
  - 不適合多維資料。
  - 對於擁有極端長尾的分布效果不佳，可能將正確的數據判為離群值。

3. 孤立森林（Isolation Forest）

- 孤立森林是一種基於隨機森林的無監督學習方法。它通過隨機選擇數據分割條件，將數據劃分為不同區域，並根據某些數據點容易被「隔離」的程度來判定其是否為離群值。越容易隔離的點（即只需少量切割次數），越可能是離群值。
- 使用時機：適用於大規模數據集，且數據分布不明的情境，如金融欺詐檢測或網絡入侵檢測。
- 優點：
  - 計算效率高，適合大型資料集。
  - 無需假設資料分布，且在高維度空間下也表現良好。
- 缺點：
  - 對於數據分布非常稀疏的數據效果有限。
  - 對於小型資料集可能不準確，因為隨機性可能導致不穩定性。

4. 局部離群因子（Local Outlier Factor, LOF）

- 局部離群因子是一種基於密度的方法，用來找出局部空間中的離群值。LOF 通過計算每個點的密度及其鄰近點的密度，來評估每個點是否與其鄰近點顯著不同。如果某點的密度遠低於鄰近點，則該點被視為離群值。
- 使用時機：適合需要發現群內異常或多群異常的情境，如空間異常偵測、基因數據分析等。
- 優點：
  - 能偵測出局部異常點，對於局部密度變化的數據效果好。
  - 不依賴整體數據分布，適合發現群體內的異常點。
- 缺點：
  - 計算密集，尤其是在高維度資料下。
  - 需要選定鄰居數參數，這對於不同數據集的效果影響很大，且需要適當調整。

總結

- Elliptic Envelope：
  適合高斯分布的多變量資料，但在高維度下效率低。
- IQR Method：
  適合一維數據，無需分布假設，但無法處理多變量數據。
- Isolation Forest：
  適合大規模、高維度且無分布假設的資料集，對小數據效果不佳。
- Local Outlier Factor：
  適合多變量且密度變化的資料，但在高維度下運算成本高，且需要選定合適的參數。

處理離群值

1. 直接刪除檢測到的離群值

- 此方法直接移除資料集中被識別為離群值的數據點。這種方法適合當資料集中離群值非常少，且這些離群值可能對模型性能造成負面影響的情況。
- 使用時機：
  - 離群值數量極少，刪除後對數據樣本總體結構影響較小。
  - 離群值顯著偏離正常數據範圍，且屬於顯然的異常點（例如儀器測量錯誤）。
  - 適用於不要求保留每筆數據樣本的場景，例如探索性資料分析或前期清理過程。
  - 可以使用 Isolation Forest 或 Local Outlier Factor 來檢測離群值，檢測後再刪除。

2. 新增一個特徵用以標示離群值

- 此方法在數據中增加一個新特徵，該特徵標示數據是否為離群值。例如，可使用二元變數標示：1 表示離群值，0 表示非離群值。這種方法保留了所有數據，同時向模型提供離群值的信息，有助於某些模型利用離群值進行更好的預測。
- 使用時機：
  - 離群值具有潛在信息，不應被刪除。例如，客戶信用風險中，高消費可能是潛在風險指標。
  - 模型可以有效利用該特徵，以增強預測能力。
  - 離群值標示能夠提升模型解釋性或提供額外的分析維度。
- 使用離群值檢測算法（如 Isolation Forest），並在檢測結果基礎上新增標示特徵。

3. 轉換特徵值來降低離群值影響

- 當離群值集中於某個特定的特徵上時，可以使用數據轉換技術降低其影響，例如對數變換、平方根變換或分位數截斷。這種方法能減少極端值對整體模型的影響，而不必刪除離群值。
- 使用時機：
  - 離群值主要存在於某一特徵上，且屬於極端偏態的情況。
  - 離群值可能包含有效信息，不希望簡單刪除或標示。
  - 常用於處理數據分佈不均勻或極端偏態數據，如收入、交易額等。
- 可使用 PowerTransformer 或 QuantileTransformer 來進行數據轉換，使數據分佈更趨於常態。

偵測新奇值

用來偵測資料中不符合訓練資料分佈的異常樣本的方法。主要有兩種方法：One-class SVM 和 局部離群因子 (Local Outlier Factor, LOF)。

1. One-Class SVM (單類別支持向量機)

- 針對僅有正常樣本的情況下進行訓練的無監督學習模型。該方法嘗試尋找正常樣本分佈的邊界，並將超出邊界的樣本視為異常值。One-Class SVM 使用支持向量機的原理，將大部分樣本隔離在邊界內，並用核函數（如高斯核）來構建非線性的邊界。當進行新奇值偵測時，新樣本若落在邊界外，就會被標記為異常值。
- 使用時機：
  - 新奇值偵測 (novelty detection)：One-Class SVM 適合用於新奇值偵測，尤其是當有標準的「正常」樣本，且我們希望偵測未出現在訓練集中的新奇樣本時。
  - 訓練樣本都是正常樣本：One-Class SVM 假設訓練資料是純正常樣本，這在某些情況下是理想的，比如製造業中的產品檢測，只有符合標準的產品才能進入生產線。
- 優點與局限：
  - 優點：能夠有效偵測新奇樣本，尤其在訓練資料單純為正常樣本時效果良好。
  - 局限：One-Class SVM 計算成本較高，尤其在高維資料集上，且對於大規模數據表現不佳。

2. 局部離群因子 (Local Outlier Factor, LOF)

- 基於局部密度的方法，用於判斷樣本是否偏離周圍的樣本分佈。該方法基於 k 近鄰演算法，計算每個樣本的「離群因子」，該因子表示該樣本的局

部密度與其鄰近點的密度差異。若一個樣本的密度顯著低於周圍鄰居的密度，則該樣本被標記為離群值。LOF 的評估方式是相對性的，即根據周圍的樣本來決定樣本的異常程度。

- 使用時機：
  - 離群值偵測 (outlier detection)：LOF 通常用於偵測資料集中異常樣本，尤其在異常值和正常值分佈同時存在的情況下效果較佳。
  - 無新奇值偵測功能：LOF 在 Scikit-learn 中不能用於新奇值偵測，因為 LOF 需要比較樣本之間的相對密度，這限制了其對新資料的適用性，LOF 更適合用於已知數據集中的異常偵測。
- 優點與局限：
  - 優點：在資料分佈不均勻或具有局部異常情況下表現出色，適合偵測離群值。
  - 局限：無法進行新奇值偵測，且計算成本隨著鄰居數增加而提高。

總結

- One-Class SVM：
  適用於新奇值偵測，特別是只有正常樣本的情況。適用於想要偵測「未見過」的新異常情況。
- LOF：
  適用於離群值偵測，特別適合有異常樣本存在於訓練集的情況，能有效發現訓練資料中的局部異常。

選取重要特徵

在機器學習中，「選取重要特徵」(Feature Selection) 和「特徵縮減」(Feature Reduction) 是非常重要的步驟。這些技術的主要目的是減少模型的複雜度、提升模型的效能，以及防止過擬合的發生。原因和方法如下：

1. 需要特徵選取與縮減的原因

- 減少模型複雜度：
  過多的特徵可能會增加模型的複雜度，導致計算成本過高，模型的訓練和預測速度變慢。
- 避免過擬合：
  當特徵數量遠超過訓練樣本數時，模型可能會過度擬合訓練數據，降低對新數據的泛化能力。
- 提升效能：
  移除冗餘或無關的特徵可以使模型專注於與目標變量最相關的特徵，提升預測精度。

- 可解釋性：

  去除無意義的特徵可以讓模型更簡單、更容易解釋，有助於了解哪些因素真正影響目標變量。

2. 特徵選取方法

- 依統計性質過濾：

  透過變異性或相關性來篩選特徵。變異性過濾通常是移除方差過低的特徵，因為它們對模型預測貢獻有限；相關性過濾則通過計算特徵與目標變量之間的相關係數，保留高度相關的特徵。

- 特徵相關性：

  皮爾森、斯皮爾曼和肯德爾相關係數，這三種相關係數常用來衡量兩個變量之間的線性或單調關係。

  - 皮爾森相關係數 (Pearson Correlation Coefficient)
    - 用途：用來衡量兩個連續型變數之間的線性相關性。
    - 原理：反映了兩個變量的線性相關程度，即變量之間的線性關係是否強烈。

$$\rho_{X,Y} = \frac{\operatorname{Cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum(X_i - \overline{X})^2}\sqrt{\sum(Y_i - \overline{Y})^2}}$$

  - 斯皮爾曼相關係數 (Spearman's Rank Correlation Coefficient)
    - 用途：適合評估兩個變數之間的單調關係，無論是否為線性。
    - 原理：透過比較數據排序來評估變量之間的關聯性，以變量的秩 (rank) 作為計算基礎，從而捕捉到變數之間的單調相關。

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

  - 肯德爾相關係數 (Kendall's Tau)
    - 用途：用於衡量兩個變量秩之間的一致性，適用於序列數據。
    - 原理：基於觀測對之間的一致性計算，肯德爾係數考慮了每一對數據之間是否一致，即若其中一個變數的值增加，另一個變數的值也相應增加或減少。

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \operatorname{sgn}(x_i - x_j)\operatorname{sgn}(y_i - y_j)$$

在特徵選取中，不同的特徵和目標項型態適合不同的相關性檢驗方法。以下介紹卡方檢驗、ANOVA 的 F-value 及 Mutual Information 的使用時

機。

1. 卡方檢驗 (Chi-Square Test)

使用時機：當特徵與目標項都是類別型資料時，適合使用卡方檢驗來衡量兩者的相關性。此檢驗的目的是評估觀察值與期望值之間的差異，判斷特徵和目標是否有顯著關聯。

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

解釋：卡方檢驗的結果值越大，代表特徵與目標變量之間的關聯越強。根據卡方分佈的臨界值可以判斷是否具有顯著性，通常搭配 p 值來確定顯著性。

2. ANOVA 的 F-value

使用時機：當數值型特徵與類別型目標項之間，需要衡量特徵對目標的影響程度時，適合使用單因子變異數分析（ANOVA）。ANOVA 的 F-value 用於判斷數值型特徵是否能有效區分不同的類別型目標。

$$F = \frac{\text{MS}_{\text{between}}}{\text{MS}_{\text{within}}}$$

解釋：F 值越大，代表分組間的差異相對於組內變異越大，表示該特徵對目標項有較大的區分力。通常會搭配 F 值的 p 值來判斷是否具有顯著性。

3. Mutual Information

使用時機：當特徵與目標項皆為數值型時，可以使用 Mutual Information (互信息) 來衡量兩者之間的依賴關係。Mutual Information 測量了變量之間的非線性關聯度，能捕捉線性和非線性關係。

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

解釋：Mutual Information 的值越大，代表兩者之間的依賴關係越強。當 $I(X;Y) = 0$時，代表特徵和目標變量完全獨立。Mutual Information 能夠衡量非線性關聯，是較為靈活的度量方式。

嘗試錯誤法挑選特徵

在特徵選擇過程中，使用遞迴特徵汰除 (Recursive Feature

Elimination, RFE) 和排列特徵重要性 (Permutation Feature Importance) 都是常用的「嘗試錯誤法」，它們能幫助識別對模型預測影響較大的特徵並減少不必要的特徵。以下是這兩種方法的概念與適合的使用情境：

1. 遞迴特徵汰除 (Recursive Feature Elimination, RFE)

- RFE 是一種遞迴的特徵選擇方法。它從一開始包含所有特徵的模型開始，訓練模型並根據特徵的重要性去除最不重要的特徵（例如權重較小的特徵），重複該過程，直到只剩下預定數量的特徵。RFE 通常和支持向量機 (SVM)、線性回歸等模型一起使用，以確保重要性評估的準確性。
- 使用時機：
  - 當資料量較小：
    RFE 適合於特徵數量適中或較少的數據集，因為該方法會重複訓練模型，對於較大數據集的運算資源需求較高。
  - 目標是選擇少數高相關特徵：
    如果我們希望篩選出對模型影響最大的特徵集，並保留少數幾個關鍵特徵以提升模型的可解釋性，RFE 是較好的選擇。
  - 與特定模型結合：
    RFE 會考慮與模型相關的特徵權重，因此適合需要與特定模型結合使用的特徵選擇場景。

2. 排列特徵重要性 (Permutation Feature Importance)

- 排列特徵重要性方法會隨機打亂單個特徵的數值，觀察模型準確性或損失的變化，來量化該特徵對模型的貢獻。打亂特徵後的模型表現若下降較多，則說明該特徵對模型預測的貢獻較大。此方法不依賴於特定的模型，因此適用於任何 scikit-learn 的模型。
- 使用時機：
  - 適用於大型數據集：
    此方法不需要重新訓練模型，因此在數據集較大或模型訓練時間較長的情況下，它是較為高效的選擇。
  - 適合任何模型：
    因為它不依賴於模型內部的特徵權重或結構，排列特徵重要性適合需要對多種模型進行特徵篩選的場景。
  - 需求模型的整體解釋性：
    如果我們想要了解每個特徵對於模型的整體影響，並非僅僅是為了提高模型的精度，排列特徵重要性能夠提供一個穩健的特徵貢獻評估。

使用建議

- 小型數據集且對運算資源不敏感：考慮使用 RFE。
- 大型數據集或複雜模型：選擇排列特徵重要性以節省時間和資源。

- 追求解釋性：排列特徵重要性能提供特徵對模型的全局性影響評估。

Module 3. 線性迴歸

在 Scikit-learn 中，線性迴歸（Linear Regression）有多種形式，包括簡單線性迴歸、多元線性迴歸、多項式迴歸等。除了基本的線性迴歸模型，還有正則化版本的線性迴歸模型，例如 Ridge 和 Lasso Regression，它們可以幫助控制模型的複雜度。以下是這些模型和它們的評估方式的詳細說明：

簡單線性迴歸（Simple Linear Regression）

定義：簡單線性迴歸是指單一自變量對應一個目標變量的回歸模型。

$$y = \beta_0 + \beta_1 x + \epsilon$$

- y 是目標變量（應變數）。
- x 是自變量（獨立變數）。
- β_0 是截距，表示當 $x = 0$ 時 y 的預測值。
- β_1 是回歸係數，表示 x 每變動一單位時 y的變動量。
- ϵ 是誤差項，用來表示實際觀測值與預測值之間的差異。

使用場景：適合用於描述兩個變量之間的線性關係。

多元線性迴歸（Multiple Linear Regression）

定義：多元線性迴歸擴展了簡單線性迴歸的概念，允許多個自變量對單一目標變量的影響。

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

- y 是目標變量（應變數）。
- $x_1, x_2, \ldots, x_p$ 是 p 個自變量（獨立變數）。
- β_0 是截距，表示當 $x_i = 0$ 時 y 的預測值。
- $\beta_1, \beta_2, \ldots, \beta_p$ 是回歸係數，表示自變量 x_i 每變動一單位時 y 的變動量（在其它變量固定不變的情況下）。
- ϵ 是誤差項，用來表示實際觀測值與預測值之間的差異。

使用場景：適合多個變量之間存在線性關係的場景，常用於描述和預測多個因素如何影響目標變量。

多項式迴歸（Polynomial Regression）

定義：多項式迴歸是一種將線性模型拓展成非線性模型的方式。它引入了自變量的多項式形式。

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \epsilon$$

- y 是目標變量（應變數）。
- $x, x^2, x^3, \ldots, x^n$ 是自變量的不同次方項。
- β_0 是截距。。
- $\beta_1, \beta_2, \ldots, \beta_p$ 是對應各個多項式項的迴歸係數，表示自變量不同次方項對目標變量的影響程度。。
- ϵ 是誤差項，用來表示實際觀測值與預測值之間的差異。

使用場景：適合自變量和目標變量之間存在非線性關係的情況。可用於曲線擬合，但需要注意避免過擬合問題。

正規化的線性迴歸

Ridge Regression

Ridge Regression 是線性迴歸的正則化版本，其目的是通過懲罰過大的係數來防止過擬合。Ridge 使用 L2 正則化，將懲罰項加入損失函數中。

$$\min_{\beta} \left(\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

在多重共線性（即自變量之間高度相關）的情況下，Ridge Regression 可以有效地限制模型的複雜度，從而提高預測穩定性。調整參數 λ 的大小來控制正則化強度，λ 越大，正則化強度越高。

Lasso Regression

定義：Lasso Regression 是另一種正則化迴歸方法，它使用 L1 正則化，將懲罰項加入損失函數中。這使得某些不重要的係數會縮小至 0，因此 Lasso 可以執行特徵選擇。

$$\min_{\beta}\left(\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2+\lambda\sum_{j=1}^{p}|\beta_j|\right)$$

適合用於需要同時控制模型複雜度和執行特徵選擇的情況，因為 Lasso 能夠自動將某些特徵的係數壓縮至零，從而在降低維度的同時提高模型解釋性。同樣是通過調整 λ 的大小來控制正則化強度，當 λ 增大時，更多特徵的係數會趨於零。

線性迴歸模型評估指標

在使用迴歸模型後，可以根據以下評估指標來衡量模型的表現：

均方誤差（Mean Squared Error, MSE）

計算預測值和實際值之間差異的平方平均值。

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

- n 是樣本數。
- y_i 是第 i 個樣本的實際值。
- $\hat{y}_i$ 是第 i 個樣本的預測值。

MSE 越小，模型的預測準確度越高。

均方根誤差（Root Mean Squared Error, RMSE）

RMSE 是 MSE 的平方根，具有與原始數據同樣的單位，便於解釋。

$$\text{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$$

- n 是樣本數。
- y_i 是第 i 個樣本的實際值。
- $\hat{y}_i$ 是第 i 個樣本的預測值。

平均絕對誤差（Mean Absolute Error, MAE）

計算預測值和實際值之間的絕對差異的平均值。

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- n 是樣本數。
- y_i 是第 i 個樣本的實際值。
- $\hat{y}_i$ 是第 i 個樣本的預測值。

MAE 的值越小表示預測結果越準確。

決定係數（R-squared, R^2）

衡量模型解釋變異的能力，範圍在 0 到 1 之間，越接近 1 表示模型解釋能力越強。

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

- y_i 是第 i 個樣本的實際值。
- $\hat{y}_i$ 是第 i 個樣本的預測值。
- $\bar{y}$ 是實際值的平均值。
- $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ 是模型的總誤差平方和（Residual Sum of Squares, RSS）。
- $\sum_{i=1}^{n}(y_i - \bar{y})^2$ 是實際值的總變異平方和（Total Sum of Squares, RSS）。

殘差分析

在機器學習和統計學中，殘差分析是一種評估模型擬合程度的工具。它可以幫助檢查回歸模型是否滿足基本假設，從而驗證模型的合理性和預測的可靠性。殘差分析通常涉及三個主要方面：常態性、獨立性 和 變異數同質性。

1. 常態性

常態性檢驗的目的是檢查殘差是否服從常態分佈。回歸模型的殘差通常假設應該是常態分佈的，這有助於確保模型的穩定性和預測精度。如果殘差偏離常態分佈，可能會影響模型的解釋力。

- Shapiro-Wilk Test (SW test)：
  SW 檢定是一種檢查樣本數據是否符合常態分佈的統計檢定。其原假設是「樣本數據來自常態分佈」，p 值越小越拒絕原假設，表示數據偏離常態。
- Kolmogorov-Smirnov Test (KS test)：
  KS 檢定也是用於檢驗數據是否符合特定分佈（例如常態分佈）的方法。KS 檢定通過比較樣本的累積分佈函數（CDF）與理論分佈的 CDF 之間的差異，來測量分佈的一致性。
- Omnibus 檢定：
  這是一種綜合性的檢定方法，用於同時檢查數據的偏度（Skewness）和峰度（Kurtosis），從而確定數據是否符合常態分佈。
- Jarque-Bera Test (JB test)：
  JB 檢定也使用偏度和峰度來測試常態性。JB 檢定的原假設為「數據來自常態分佈」。當偏度和峰度偏離常態分佈時，該檢定會拒絕原假設。

2. 獨立性

獨立性假設是指殘差之間應該是相互獨立的，特別是在時序數據中，殘差的自相關可能會導致模型不穩定。因此，獨立性檢驗可幫助確保模型的殘差不隨時間或其他變數而變化。

- Durbin-Watson Test：
  Durbin-Watson 檢定是一種專門用來檢查殘差自相關的檢定，特別是自迴歸模型。它的檢定統計值範圍為 0 到 4，接近 2 表示沒有自相關，接近 0 表示正自相關，接近 4 表示負自相關。當自相關存在時，回歸係數的顯著性可能會被過度估計。

3. 變異數同質性

變異數同質性（Homoscedasticity）假設是指殘差的變異數應該在所有解釋變數的範圍內保持不變。若變異數不等（即異質性），可能會導致回歸模型

中的參數估計失真，進而影響預測結果的可靠性。

- 局部加權散點平滑法 (Locally Weighted Scatterplot Smoothing, LOESS)：

  LOESS 是一種常用於檢查變異數同質性的工具。通過對不同區段的殘差進行加權平滑，LOESS 可以有效地繪製出殘差隨解釋變數的變化趨勢。如果平滑曲線呈現一定的模式，則可能存在異質性，這樣的模式會顯示出殘差與預測值之間的關係，而不是隨機分佈。

殘差分析提供了有效的工具來驗證回歸模型的基本假設。如果模型不符合常態性、獨立性或變異數同質性，則需要進行適當的變數變換、模型重建或引入更適合的模型來改善擬合效果。