線性回歸 Linear Regression

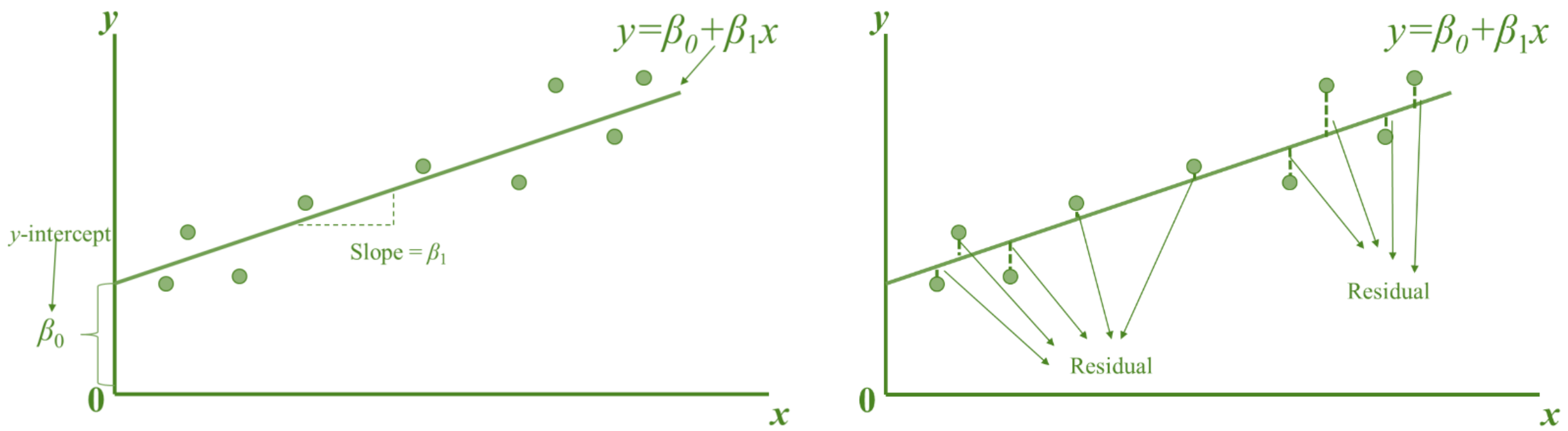
國立東華大學電機工程學系 楊哲旻

Outline

1. 參數與超參數
2. 損失函數
3. 梯度下降法
4. 學習速率
5. 批量梯度下降法
6. 隨機梯度下降法
7. 小批量梯度下降法
8. 線性回歸實作

機器學習 – 線性回歸 01. 參數與超參數

線性回歸為回歸模型，是統計上在找多個自變數和依變數之間的關係建出來的模型。訓練過程是從訓練集中以梯度下降法來確定權重與偏差



- 參數(Parameter)：模型從數據中可以自動學習出的變量。例如，權重(Weights)，偏差(Bias)，即變數
- 超參數(Hyper-parameter)：確定模型的一些數值，此數值不同則模型預測能力也會不同的。超參數的數值可根據經驗確定的變量，或其他搜索演算法來決定

損失函數 (Loss Function)

損失函數又稱為**目標函數**與**成本函數**，是用來估量模型的預測值$f(x)$與真實值的不一致程度。

機器學習的目的是找出一個預測函數，並透過設計一個目標函數能透過學習降低誤差，降低誤差是指預測值越接近真實值。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y^{(i)} - y^{(i)})^2$$

$Y^{(i)}$表示實際值， $y^{(i)}$表示預測值

梯度下降法(Gradient descent)

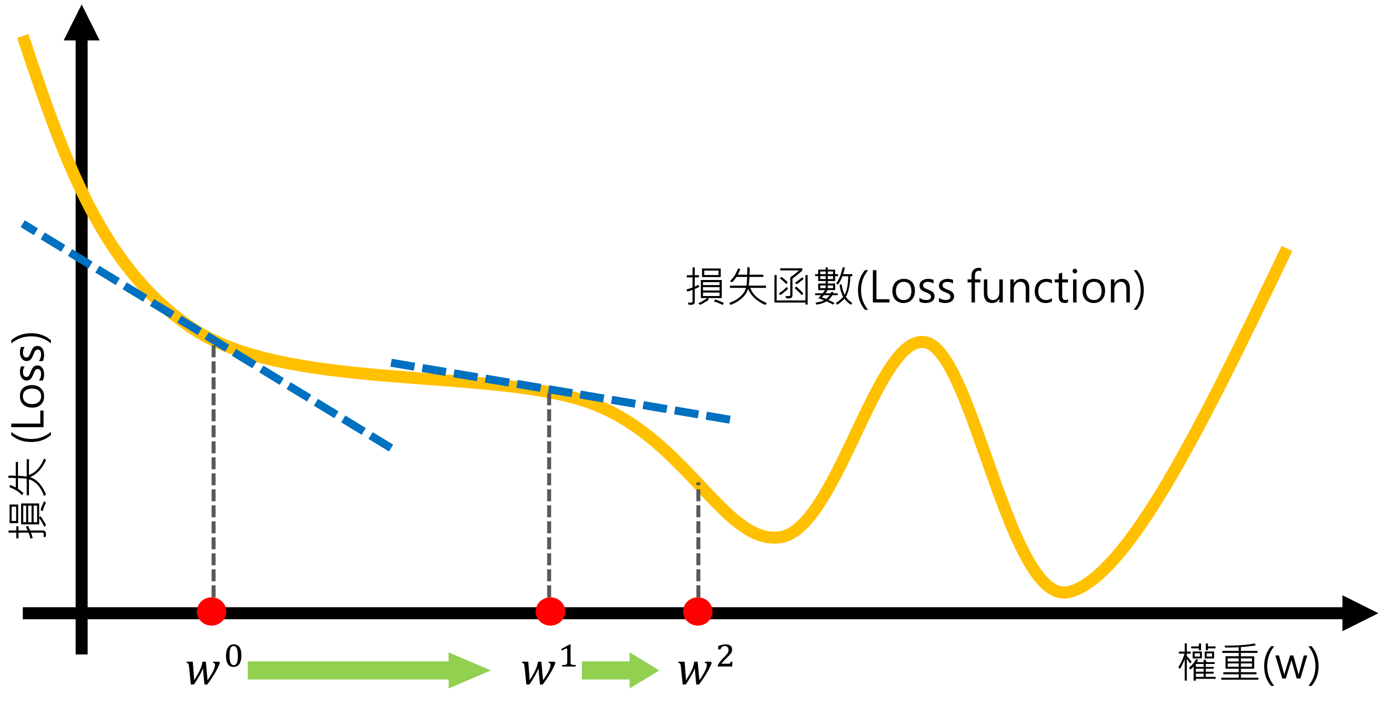
1.隨機選擇一個初始值w^0

2.計算$\left.\frac{dL}{dw}\right|_{w=w^0}$　$w^0 - \eta \left.\frac{dL}{dw}\right|_{w=w^0} \rightarrow w^1$

3.計算$\left.\frac{dL}{dw}\right|_{w=w^1}$　$w^1 - \eta \left.\frac{dL}{dw}\right|_{w=w^1} \rightarrow w^2$

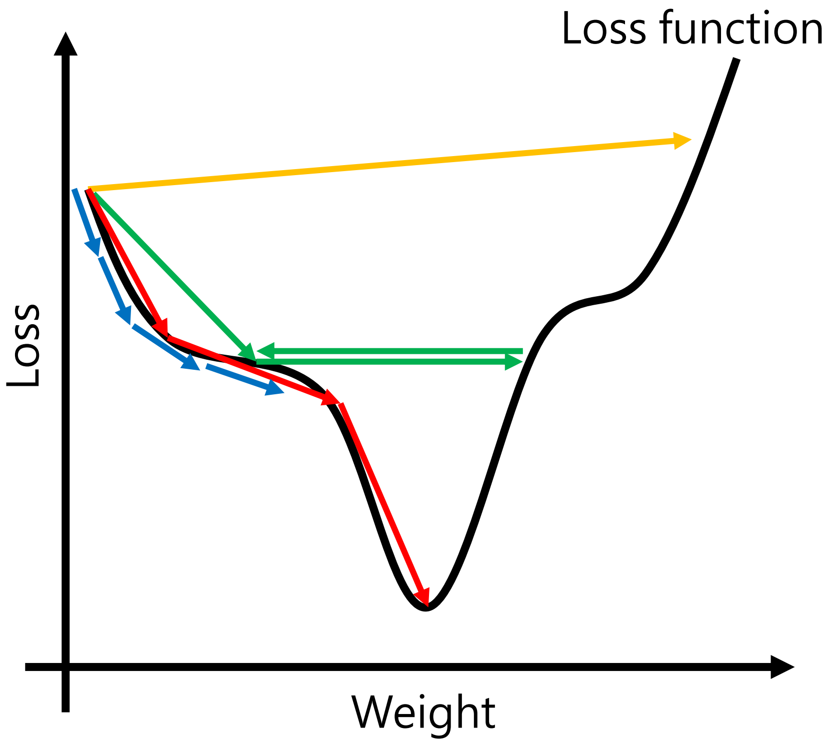
其中η為學習速率(Learning Rate)

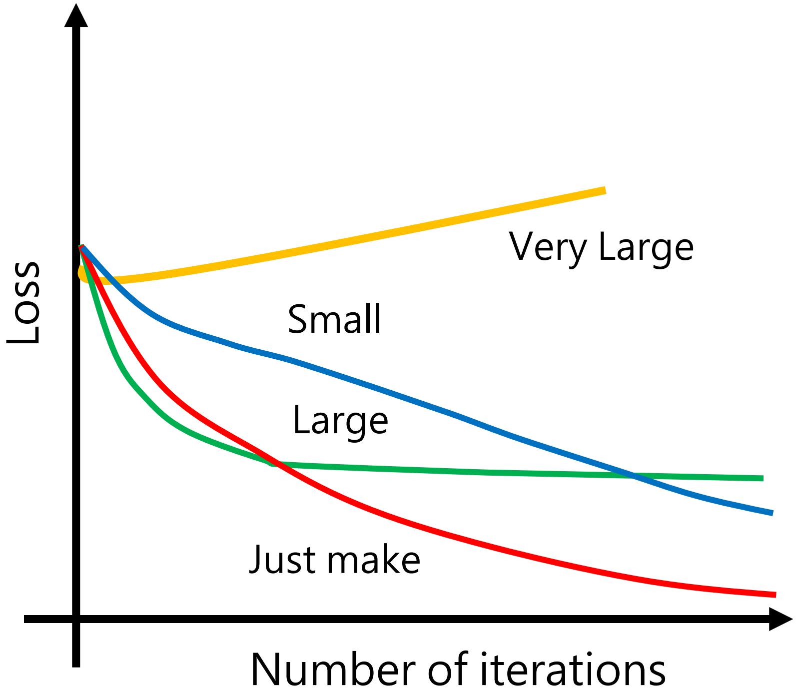
是一個超參數



學習速率(Learning Rate)

$$w^{i} = w^{i-1} - \eta \frac{dL}{dw}\bigg|_{w = w^{i-1}} \qquad \Delta w_k = -\eta \frac{\partial L}{\partial w_k}$$



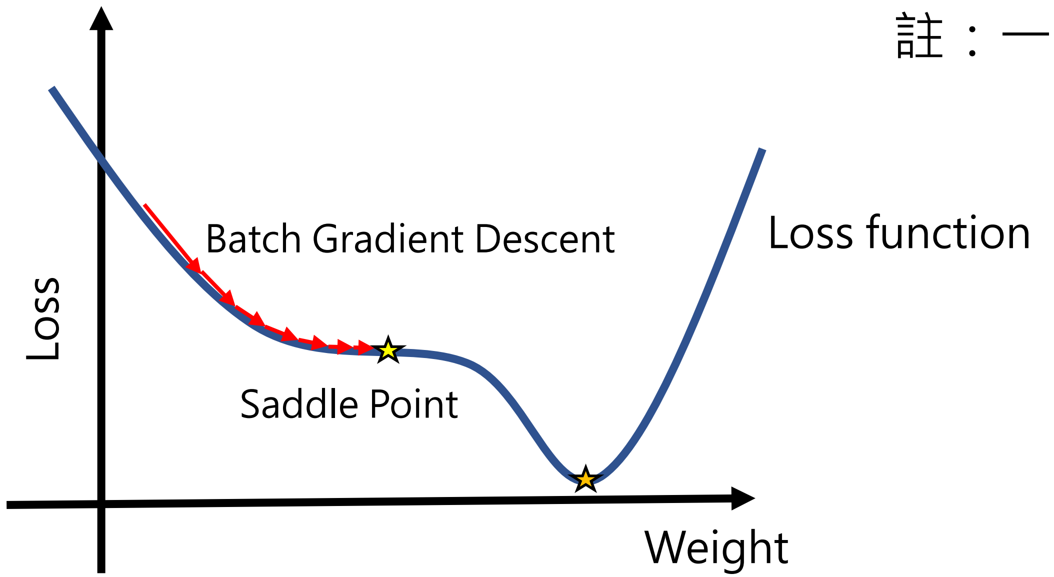


批量梯度下降法 (Batch Gradient Descent)

運用**所有資料(訓練集)**來計算誤差曲面，即算出損失函數當下的斜率，並更新權重

- 缺點：誤差曲面(損失函數)出現「鞍點」(Saddle Point)，導致梯度下降法卡住

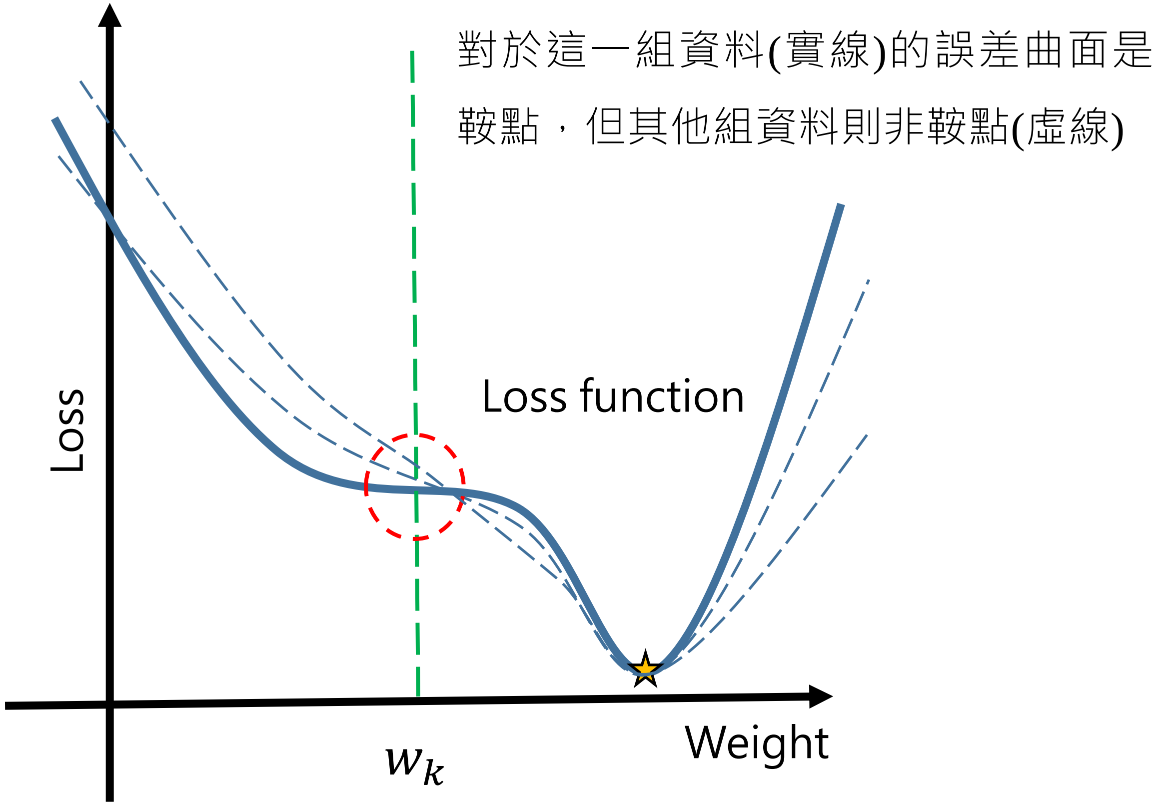
註：一個不是局部極值點的駐點稱為**鞍點**



隨機梯度下降法(Stochastic Gradient Descent, SGD)

每次只**隨機取樣一組資料(訓練集內的一個樣本)**來計算誤差曲面，即算出損失函數當下的斜率來更新權重

- 優點：可以避免限於鞍點
- 缺點：每次只看一組資料，若當下的誤差曲面不是很恰當的，權重更新可能有倒回去的情況發生，會導致梯度下降過程耗費更時間



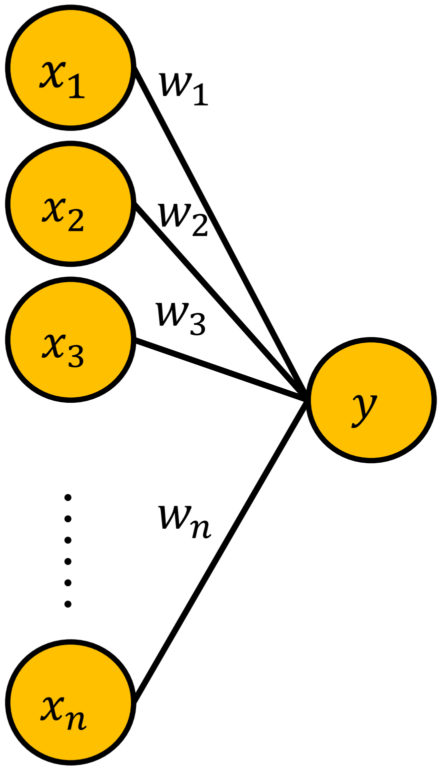
小批量梯度下降法(Mini-Batch Gradient Descent)

每次迭代都會從所有資料中取出**一小部分**(非單筆)**資料**來推算誤差曲面，小批量資料取樣的多寡，它又是一個超參數(Hyper-parameters)。

線性回歸的超參數

- 學習速率 (Learning Rate)
- 批量 (Batch)
- 迭代次數 (Number of iterations)

機器學習 – 線性回歸



- 線性回歸模型：

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

$$Loss\ function = \frac{1}{2}\sum_{i=1}^{n}(Y^{(i)} - y^{(i)})^2$$

其中$Y^{(i)}$是實際值，$y^{(i)}$是第i 個樣本的輸出，n為全部樣本數。

- 權重更新：$\Delta w_k = -\eta \frac{\partial L}{\partial w_k} = -\eta \frac{\partial}{\partial w_k}[\frac{1}{2}\sum_{i=1}^{n}(Y^{(i)} - y^{(i)})^2]$

$$= (-\eta) \times \frac{1}{2} \times 2 \times \sum_{i=1}^{n}[\left(Y^{(i)} - y^{(i)}\right)\left(\frac{-\partial y^{(i)}}{\partial w_k}\right)]$$

$$= \sum_{i=1}^{n}[\eta\left(Y^{(i)} - y^{(i)}\right)\left(\frac{\partial y^{(i)}}{\partial w_k}\right)]$$

$$= \sum_{i=1}^{n}[\eta {x_k}^{(i)}\left(Y^{(i)} - y^{(i)}\right)]$$

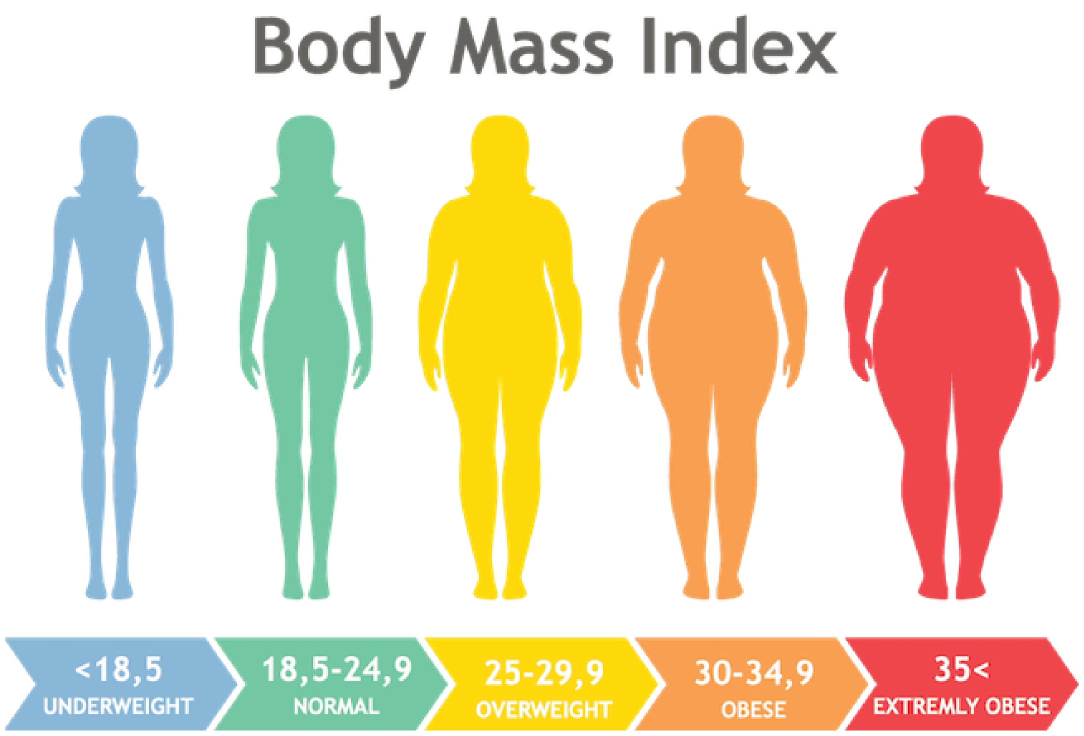
線性回歸 – 實作

線性回歸 – 實作 A Collection of Patients BMI Data

Kaggle Dataset

https://www.kaggle.com/freego1/bmi-data

- 此數據集具有25,000個患者，為csv副檔名的檔案，**輸入特徵**為性別、年齡(年)、身高(英吋)與體重(磅)，**輸出特徵**為BMI(公斤/公尺平方)
- 其中缺失值數量：身高為19，體重為16，BMI缺失值為50

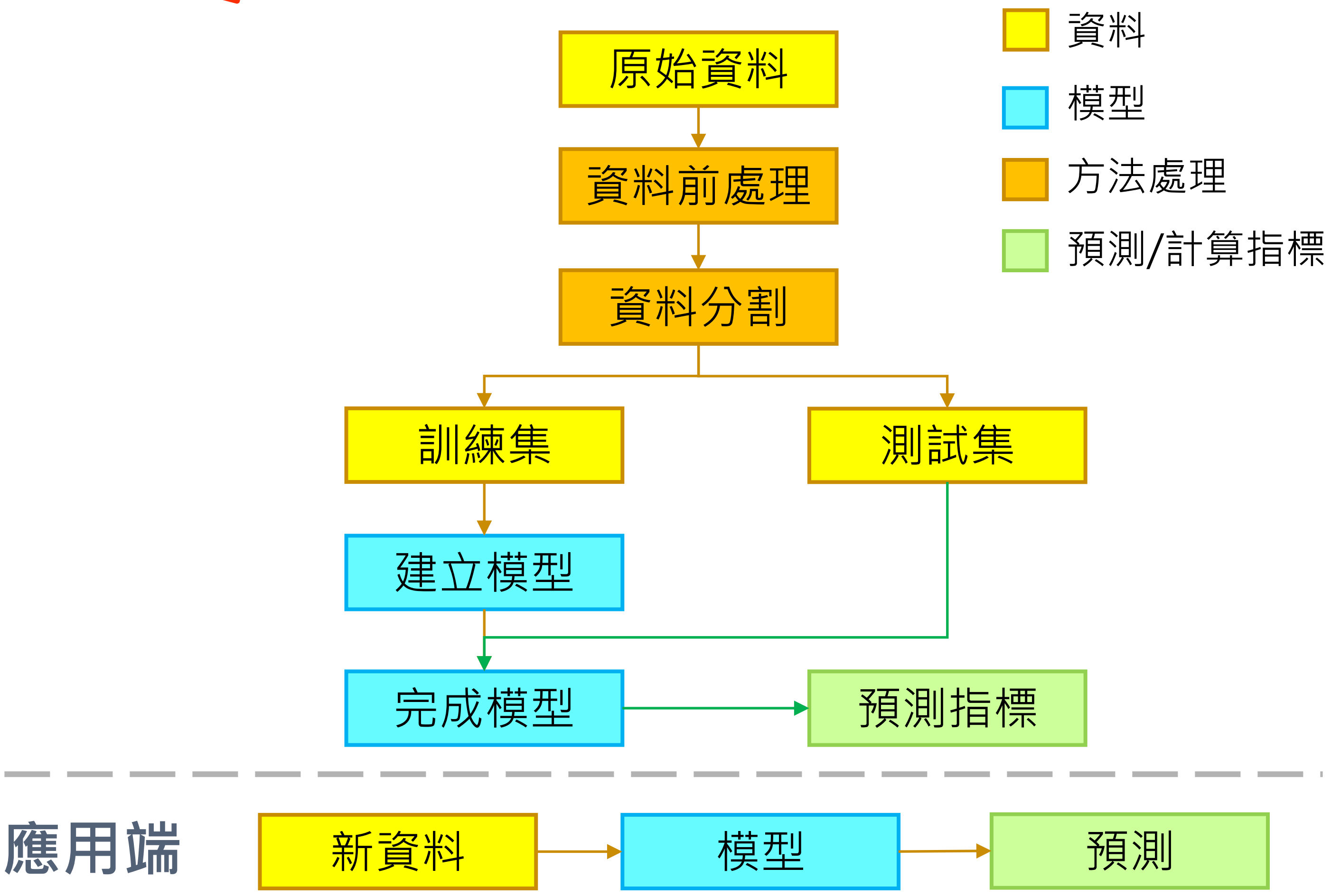


線性回歸 – 實作　A Collection of Patients BMI Data

問題

1. 缺失值如何處理？
    - A. 補零
    - B. 刪除缺值那筆資料
    - C. 補平均值、中位數與眾數
2. 非數值式的特徵如何處理？連續值/離散值？
3. 每個特徵數值範圍不同會影響模型？

流程圖



線性回歸 – 實作 A Collection of Patients BMI Data

作業1 ★★☆☆☆

1. 設計訓練二種模型，分別為**缺值補零(四個輸入)**、**刪除那筆資料(兩個輸入：體重與身高)**。其中資料集請分割為訓練集80%，測試集20%，random_state=0，且需要作<u>正規化</u>處理。並將繪出散佈圖、模型權重重要性圖、計算MSE與決定係數指標，並儲存模型為.pkl檔案。
2. 設計GUI介面(可以用PyQt5或Tkinter)，可以輸入身高、體重，點擊確定後將輸入值帶入模型(刪除那筆資料，兩個輸入：體重與身高)，並跳出視窗顯示<u>模型預測結果</u>與<u>實際BMI數值</u>。

作業2 ★★★☆☆

1. 在Kaggle找一回歸任務的資料集，透過線性回歸進行預測，並設計GUI介面。