PYTHON機器學習入門

UNIT 5：CLASSFICATION

授課教師：江尚瑀

SUPERVISED LEARNING-
CLASSIFICATION

有許多分類演算法...

- **Logistic Regression**
- **KNN**
- **Support Vector Machine**
- **Decision Tree**
- **Random Forest**
- ...

LOGISTIC REGRESSION

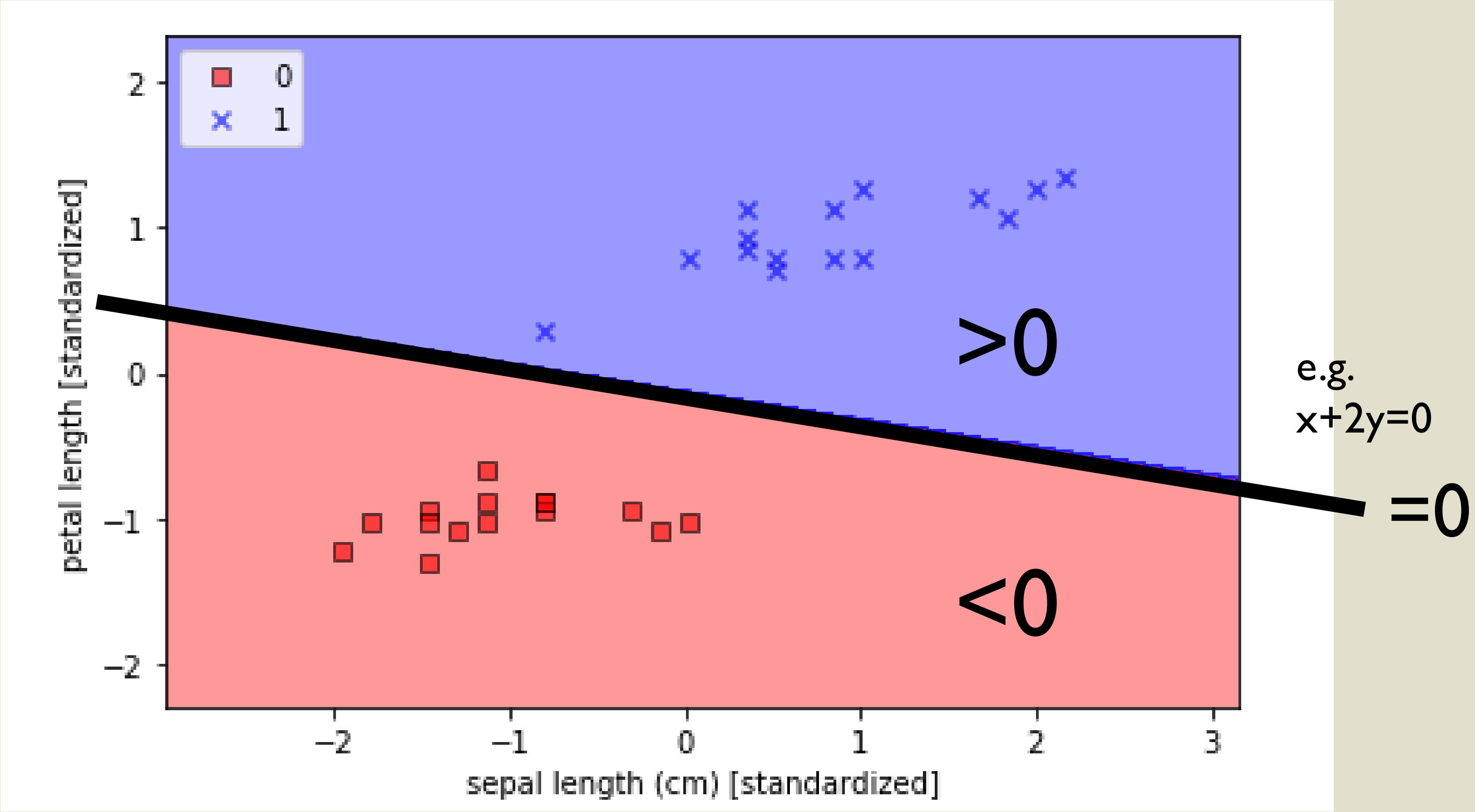
LOGISTIC REGRESSION

- Logistic regression model for binary classification
- *logistic regression*:
  when the prediction of the outcome is in a discrete form (0 or 1) or the output is in the form of yes or no
- Note: For data with more than 2 classes, softmax regression has to be used.

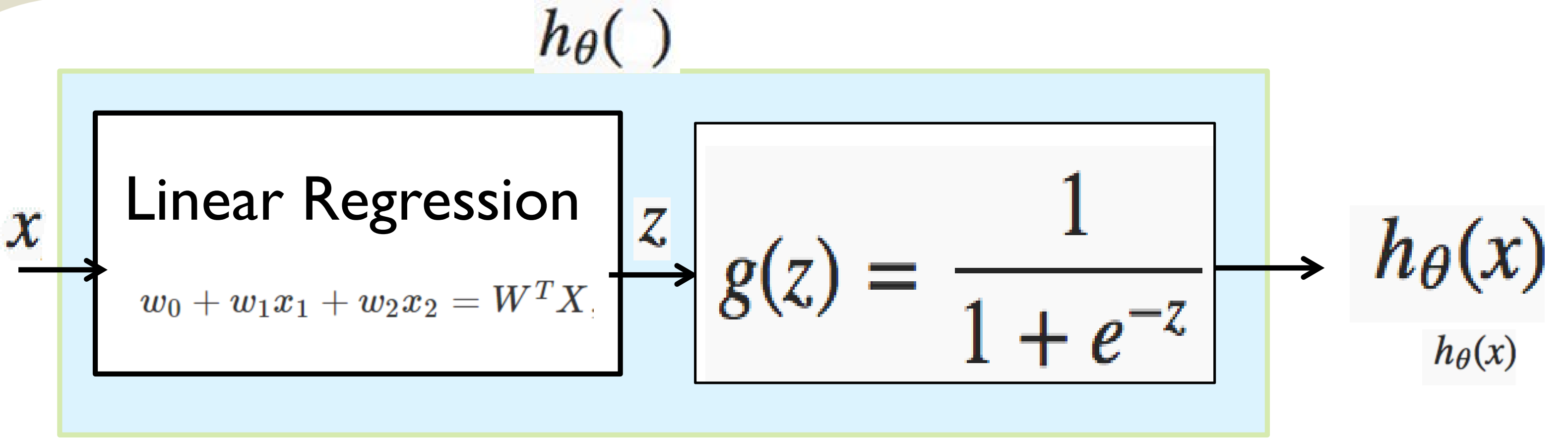
Logistic regression 作為二元分類並有對應的機率值
例如: 明天會下雨的機率是80%

LOGISTIC REGRESSION

- 基本概念即利用**Linear regression Line**將資料劃分成兩類



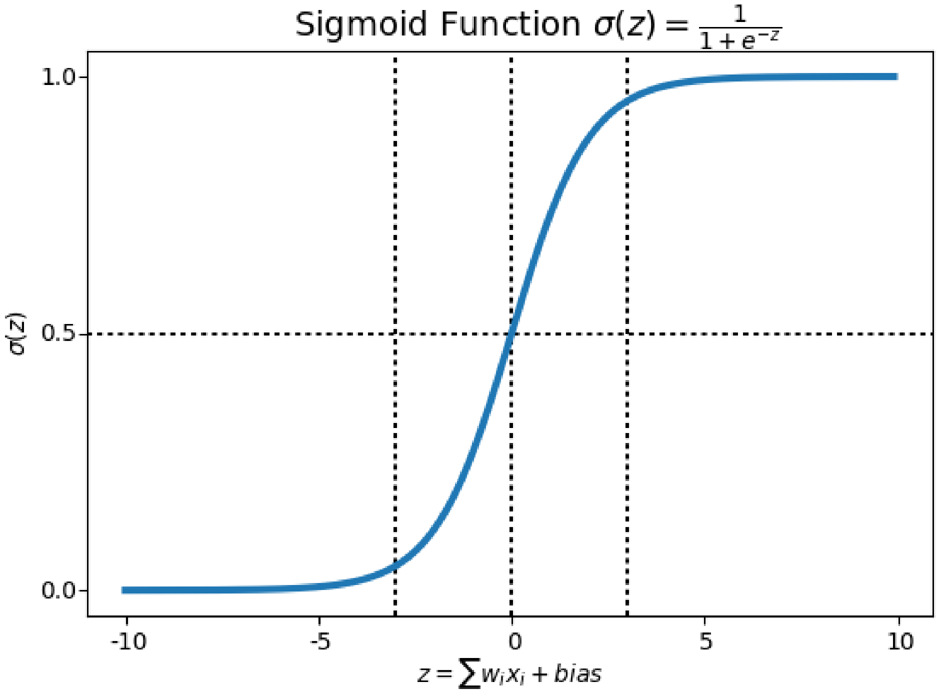
LOGISTIC REGRESSION



If > threshold
output 1;
else output 0

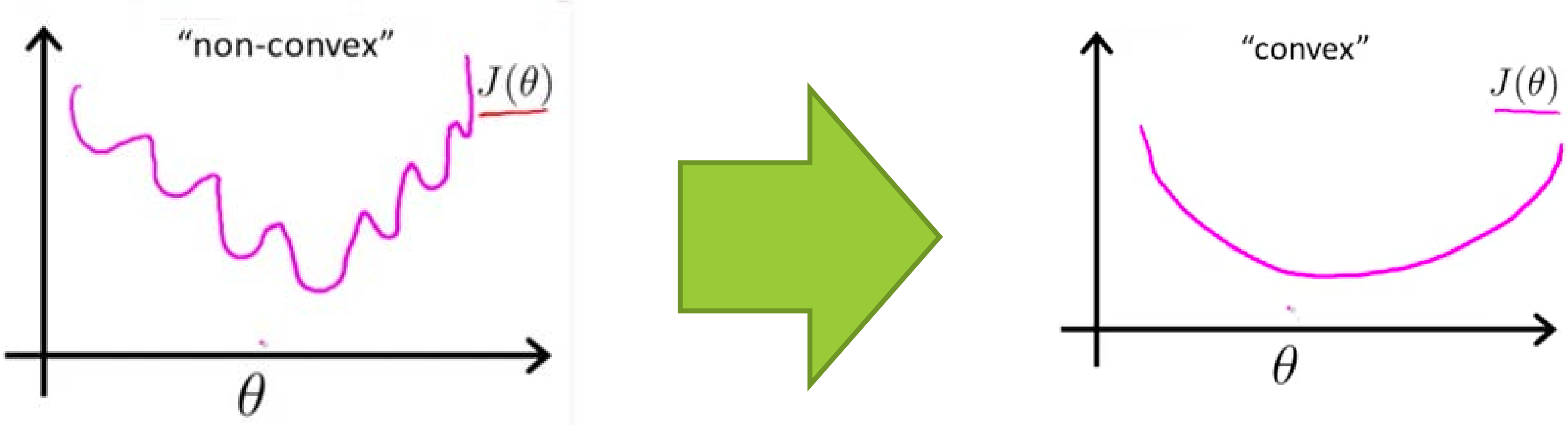
g(z) is a Logistic function
or "***Sigmoid function***"

$$0 \le sigmod(z) \le 1$$



COST FUNCTION

Linear regression: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$



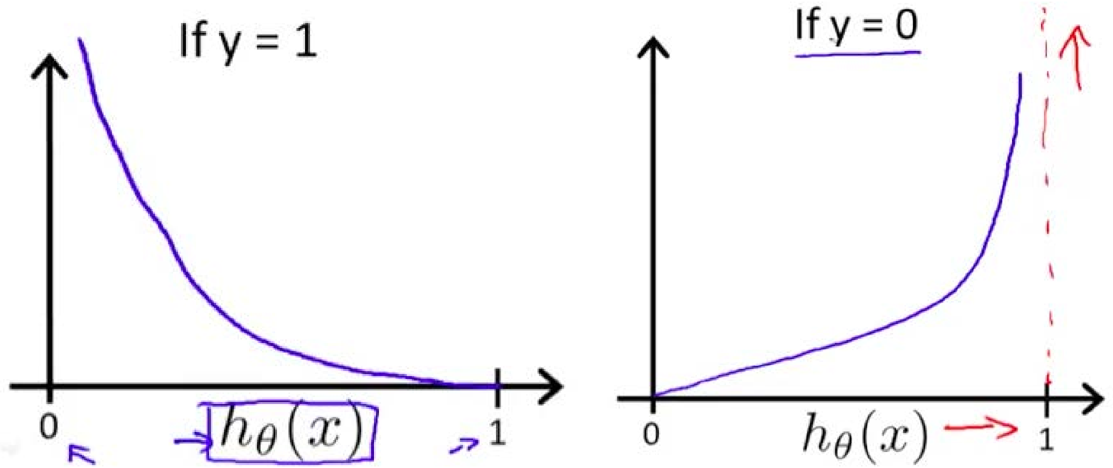
New Cost function for logistic Regression

$$L(h(x),y) = \begin{cases} -log(h(x)) & y=1 \\ -log(1-h(x)) & y=0 \end{cases} = L(h(x),y) = -ylog(h(x)) - (1-y)log(1-h(x))$$

猜愈準且機率高,則Loss應愈小; 反之. Loss應愈大

CONVEX FUNCTION

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



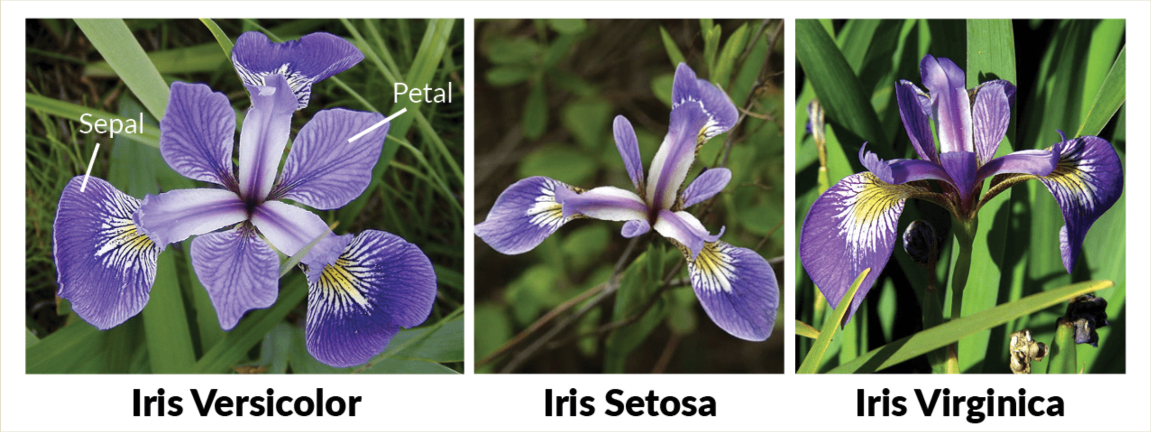
If $h_\theta(x) = y$, then $\mathrm{cost}(h_\theta(x), y) = 0$ (for $y = 0$ and $y = 1$).

If $y = 0$, then $\mathrm{cost}(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 1$.

If $y = 0$, then $\mathrm{cost}(h_\theta(x), y) \to 0$ as $h_\theta(x) \to 0$.

Regardless of whether $y = 0$ or $y = 1$, if $h_\theta(x) = 0.5$, then $\mathrm{cost}(h_\theta(x), y) > 0$.

EXAMPLE : LOGISTIC REGRESSION

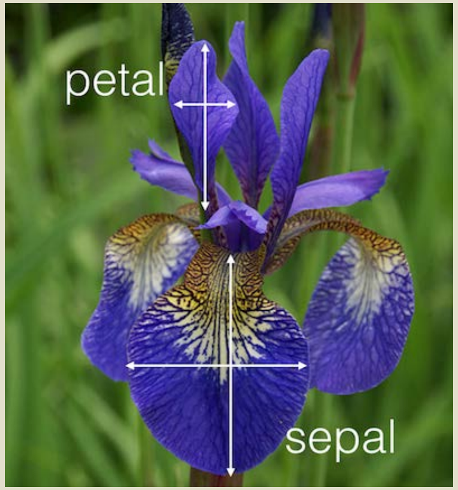


程式範例:
LogisticRegression.ipynb

特徵向量
(x_1, x_2, x_3, x_4)

特徵提取 → 分類器 Classifier →

Versicolor?
Sentosa?
Virginica ?

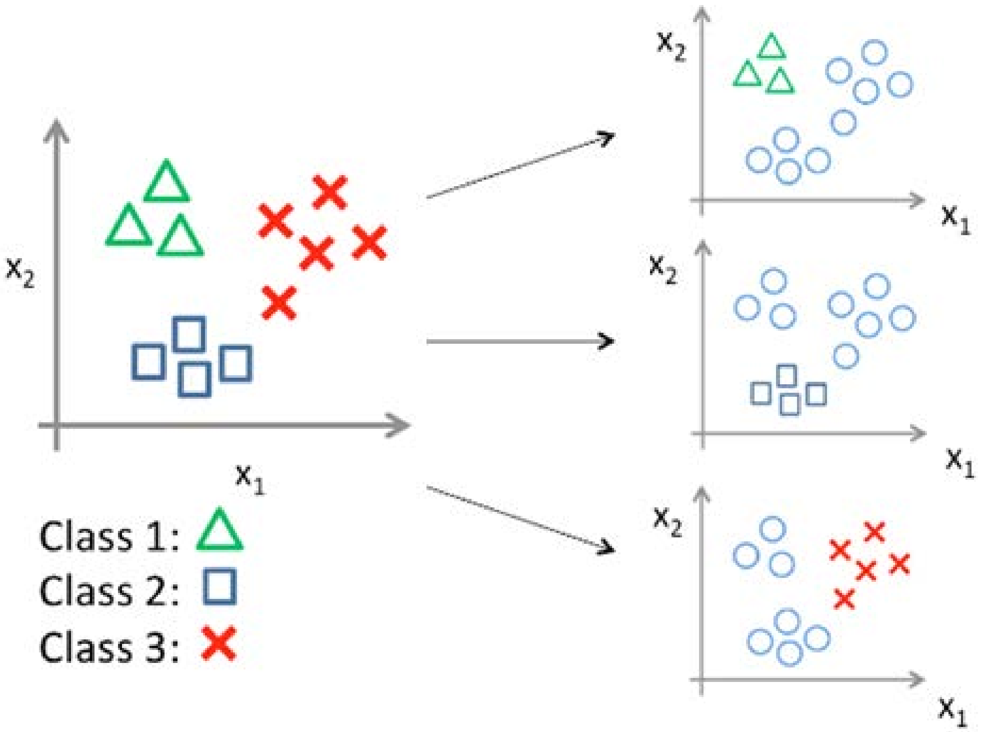


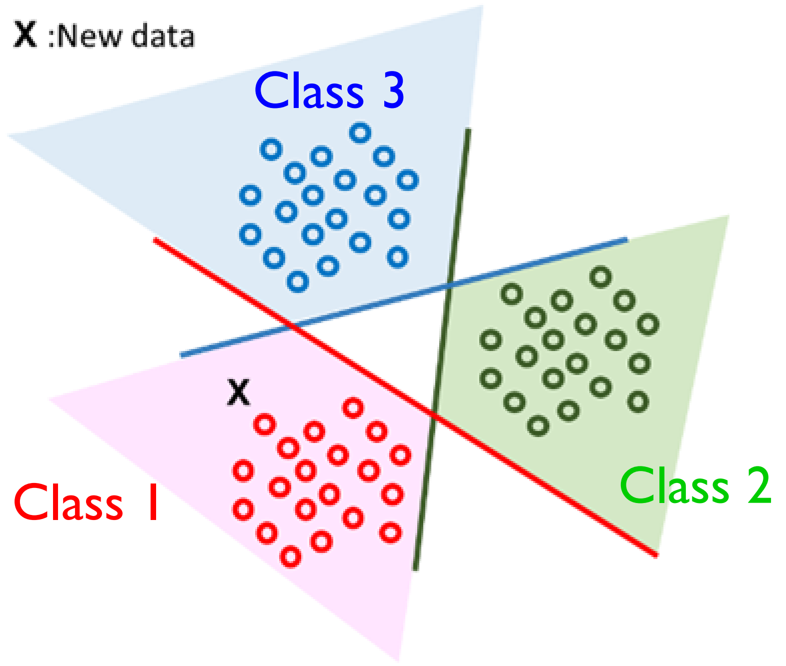
Load the data with
datasets.load_iris()
There are 150 records and 4 attributes each.
There are 3 different classes

MULTI-CLASS PROBLEM

- When the problem is multi-class problem, there are generally 2 algorithms.
- One versus rest:
  - The algorithm compares every class with all the remaining classes, building a model for every class. If you have **n** classes to guess, you have **n** models.
- One versus one:
  - The algorithm compares every class against every -individual remaining class, building a number of models equivalent to **n * (n-1) / 2**, where n is the number of classes.

ONE VERSUS ALL





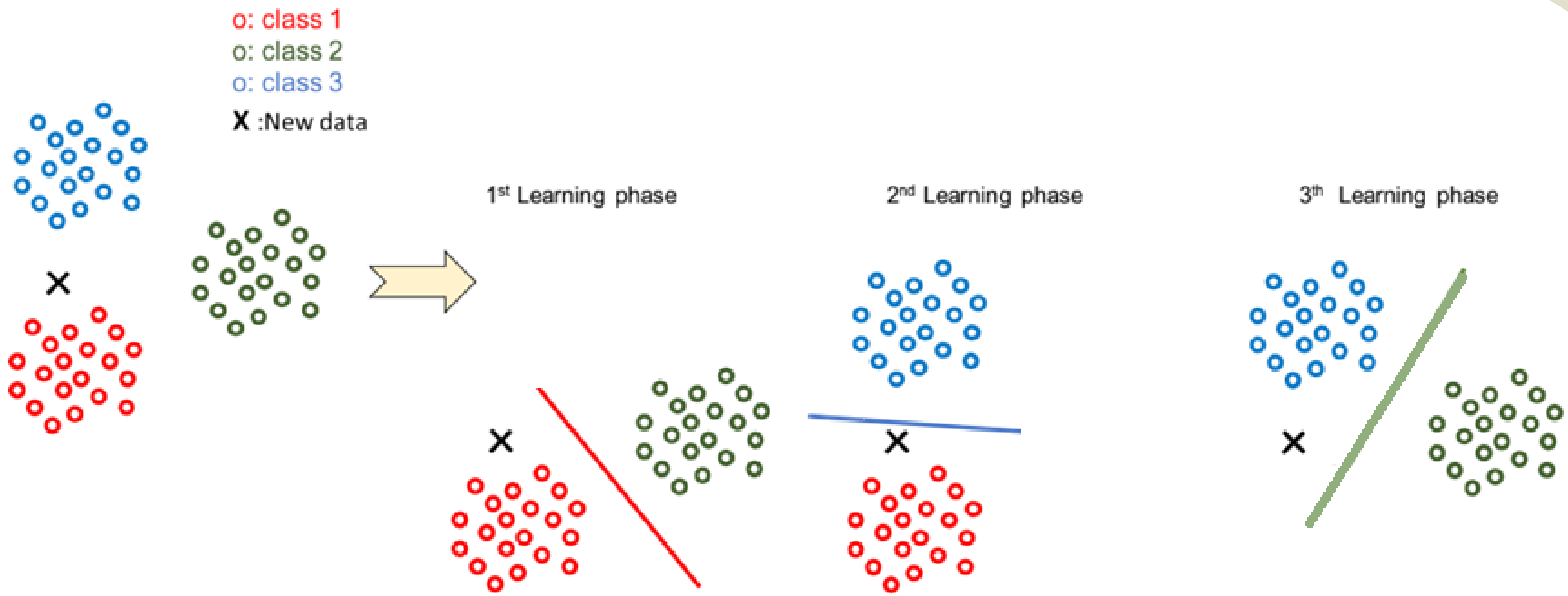
DV1(**X**)<0, 是 Class 1

DV2(**X**)<0, 不是 Class 2

DV3(**X**)>0, 不是 Class 3

This new data would be classified to class 1.

ONE VERSUS ONE



	Class 1	Class 2
Classify	1	0

	Class 1	Class 3
Classify	1	0

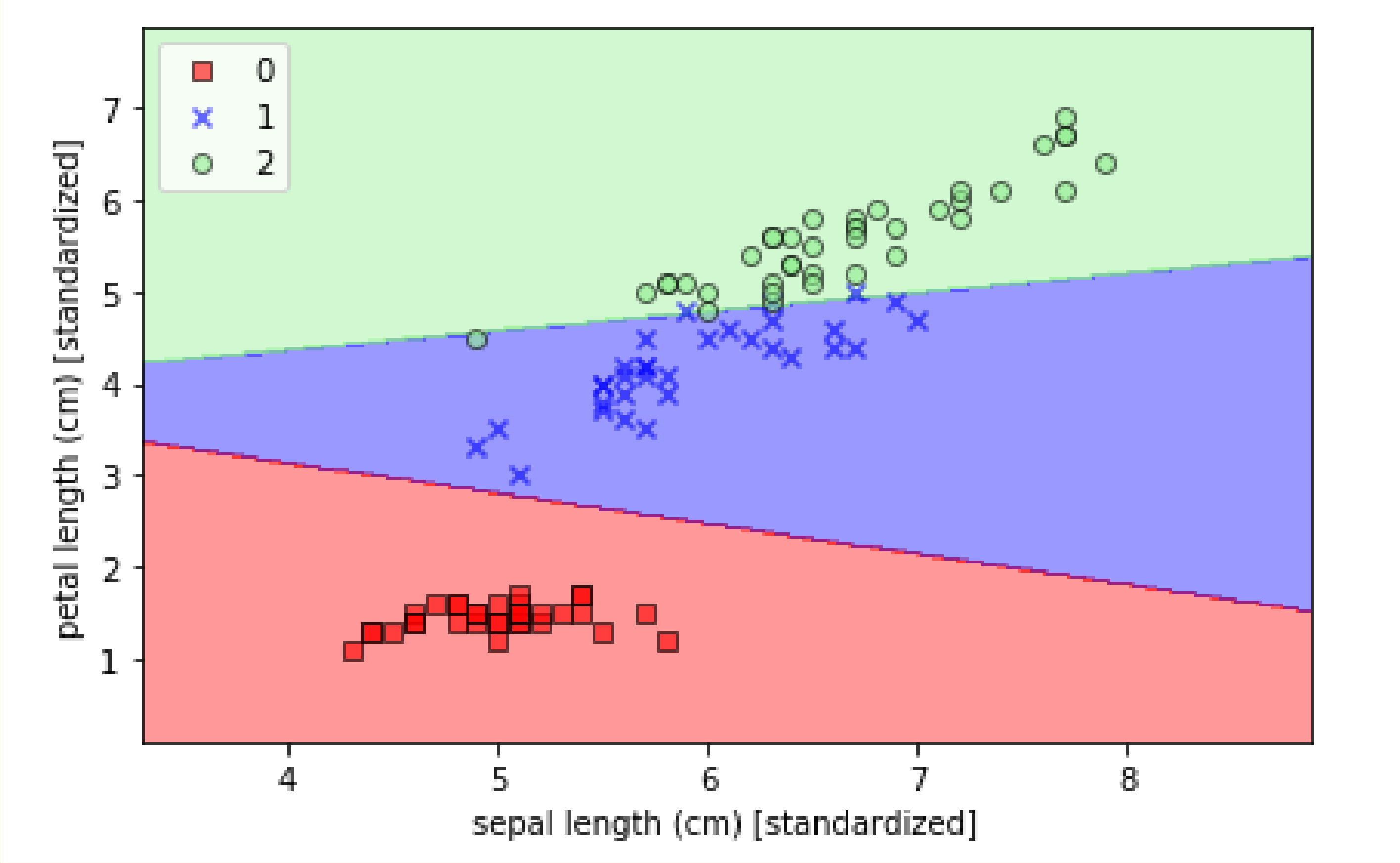
	Class 2	Class 3
Classify	0	1

Majority Vote

	Class 1	Class 2	Class 3
Count	2	0	1

EXAMPLE : MULTICLASS CLASSIFICATION

程式範例:Logistic_multiclass.ipynb



K NEAREST NEIGHBORHOOD

最近距離分類法(KNN)

- Eager Learning
  - Explicit description of target function on the whole training set
- Instance-based Learning
  - Learning=storing all training instances
  - Classification=assigning target function to a new instance
  - Referred to as “Lazy” learning

最近距離分類法(KNN)

- **KNN (K Nearest Neighbor)**是一個非常容易理解的分類演算法，在**2007**年時是**IEEE**統計排名前十名的知名資料採礦演算法之一。為一被廣泛使用、易於掌握且非常有效的演算法。

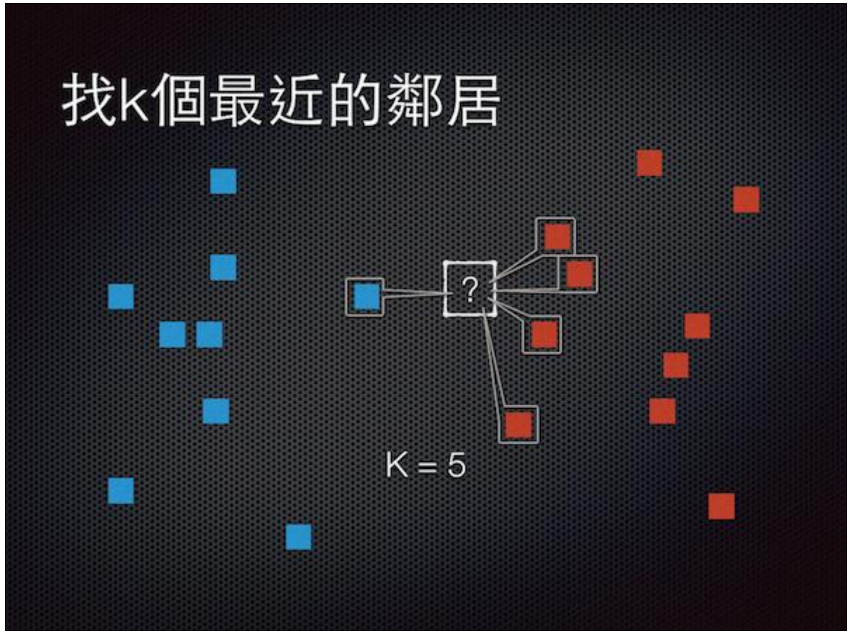
- 演算法思路

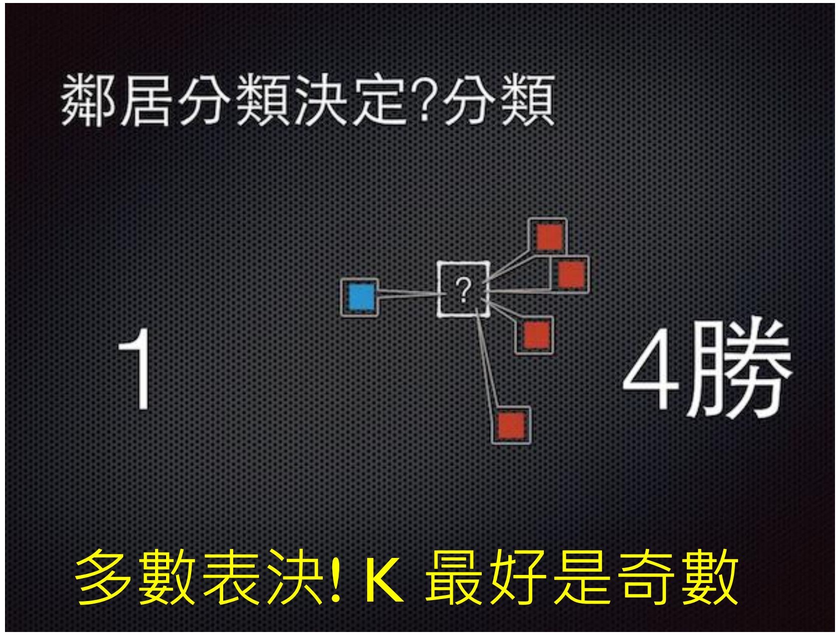
  NN分類演算法簡單來說就是要找和新數據最近的**K**個鄰居，這些鄰居是什麼分類，那麼新數據就是什麼樣的分類。

Source: D. Lowe

最近距離分類法(KNN)

- 一個新樣本是屬於藍色或紅色類別的那一個? 就從訓練集合找跟這新樣本距離最近的**K**個特徵樣本，看這些**K**個點是什麼顏色，來決定該新樣本點的最終類別。





如何決定特徵距離?

可以用座標距離、顏色相近程度、字詞重疊程度...等，這個會跟你的資料及分類問題有關。

$x = (x_1, x_2, x_3, \dots, x_n)$ and $y = (y_1, y_2, y_3, \dots, y_n)$

1.) Manhattan Distance: $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$

2.) Euclidean Distance: $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

3.) Cosine Distance: $\cos\theta = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$

KNN SUMMARY

- **kNN**分類演算法是一個基於實例**(instance-based)**的演算法，所以特徵樣本的好壞和樣本當中個分類的數量深深影響分類結果的準度。
- 如果特徵樣本當中的**A**分類數量遠大於**B**分類，那麼我們可以預期**K**個最近距離的鄰居也會有很高的機率是**A**分類，這樣就會分類失去準度。**(imbalanced data problem)**
- 整體來說，其優點在於簡單、不需要輸入資料的假設、對於異常值不敏感，而缺點在於計算量大、非常耗時，而且因為要載入所有特徵集合加入距離計算，所以記憶體空間用量也非常大。

Source: D. Lowe