巨匠線上真人

Python 資料科學應用開發

第十二堂：由 KNN 到 Logistic Regression

同學，歡迎你參加本課程

- 請關閉你的FB 、Line等溝通工具，以免影響你上課。
- 考量頻寬、雜音，請預設關閉攝影機、麥克風，若有需要再打開。
- 隨時準備好，老師會呼叫你的名字進行互動，鼓勵用麥克風提問。
- 如果有緊急事情，你必需離開線上教室，請用聊天室私訊給老師，以免老師癡癡呼喚你的名字。
- 軟體安裝請在上課前安裝完成，未完成的同學，請盡快進行安裝。

課程檔案下載


巨匠電腦線上真人
開課查詢
免費體驗專區
課程總覽
專業師資
學員專區
講師專區
最新消息
您好!
登出
點數卡產品兌換
APCS檢測專區
公告專區
我的課表
IT真人課程劃位
電腦分校課程劃位
外語真人課程劃
美語分校課程劃
取消劃位
課程檔案下載
上課權益查詢
教學平台測試
學習諮詢
常見問題
個資維護
忘記密碼
登出
課程檔案下載
程式語言好難學?
那是因為
你還沒學過Python!
線上老師LIVE直播教學．搶先看
巨匠電腦真人課程

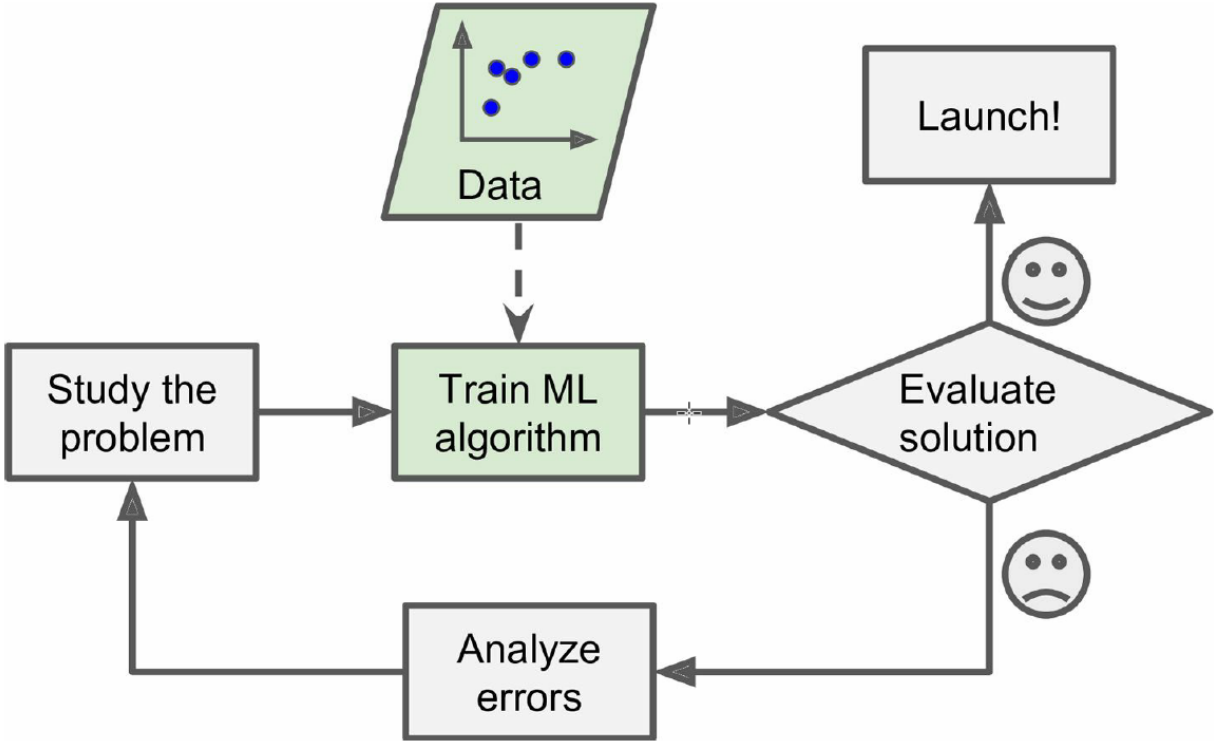
ZOOM 學員操作說明


www.pcschool.com.tw
您正在觀看蘇世宗的螢幕
查看選項
原始大小
請求遠端控制
共同註記
退出全螢幕
5 查看選項/共同註記/筆（連連看）
滑鼠
文字
筆
箭頭
擦除
格式
撤銷
重做
清除
與會者 (15)
查找參會者
張齡月 (我)
婷婷
解除靜音
舉手
2 共享螢幕（指導演練；點評作品）
老師須先停止共享螢幕
才能請學生共享螢幕
1 聊天
解除靜音
啟動視訊
邀請
與會者
共享螢幕
聊天
錄影
4 解除靜音
3 與會者/舉手

機器學習的訓練步驟

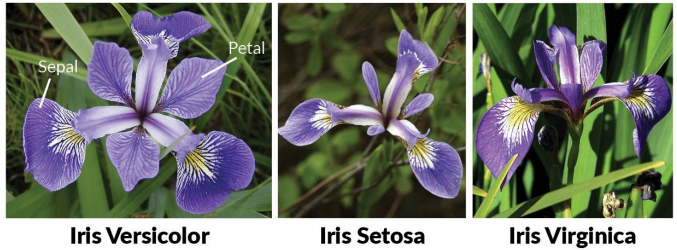
- 訓練步驟
  - 特徵選擇（Feature Selection）
  - 效能指標（performance metric）選擇
  - 分類器（classifier）與優化（optimization）演算法選擇
  - 效能評估（Evaluation）
  - 效能調校（Tuning）

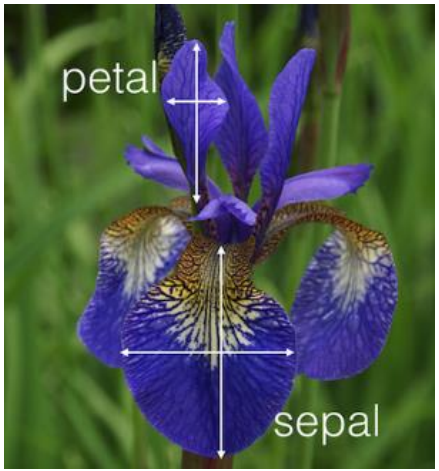
訓練流程


Data
Launch!
Study the problem
Train ML algorithm
Evaluate solution
Analyze errors

鳶尾花（Iris）資料集

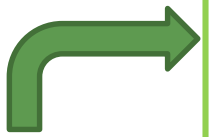
- 資料集：鳶尾花（Iris），https://archive.ics.uci.edu/ml/datasets/iris
  - 三個品種：Setosa、Versicolour、Virginica
  - 自變數
    - 花萼（sepal）長度
    - 花萼（sepal）寬度
    - 花瓣（petal）長度
    - 花瓣（petal）寬度
  - 150個樣本





特徵選擇（Feature Selection）

- 使用花瓣（petal）長度（height）與寬度（width）為特徵
  - from sklearn import datasets
  - import numpy as np
  - iris = datasets.load_iris()
  - print(iris.DESCR) # 資料集說明
  - X = iris.data[:, [2, 3]]
  - y = iris.target
- 標準化（Standardizing）
  - from sklearn.preprocessing import StandardScaler
  - sc = StandardScaler()
  - sc.fit(X_train)
  - X_train_std = sc.transform(X_train)
  - X_test_std = sc.transform(X_test)



Standardization:

$$z = \frac{x - \mu}{\sigma}$$

with mean:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

and standard deviation:

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

Min-Max scaling:

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

分類器（classifier）與優化（optimization）演算法選擇

- from sklearn.linear_model import Perceptron

選擇演算法

- ppn = Perceptron(n_iter=40, eta0=0.1, random_state=1)
- ppn.fit(X_train_std, y_train)

訓練

效能指標（performance metric）選擇

- 預測
  - y_pred = ppn.predict(X_test_std)
  - print('Misclassified samples: %d' % (y_test != y_pred).sum())
- 效能指標
  - from sklearn.metrics import accuracy_score
  - print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))

完整程式範例

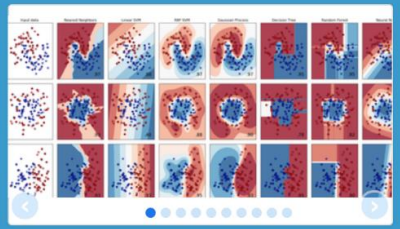
- 程式碼：ch03.ipynb

Scikit-learn 功能

scikit learn

Home Installation Documentation Examples

Google Custom Search Search



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

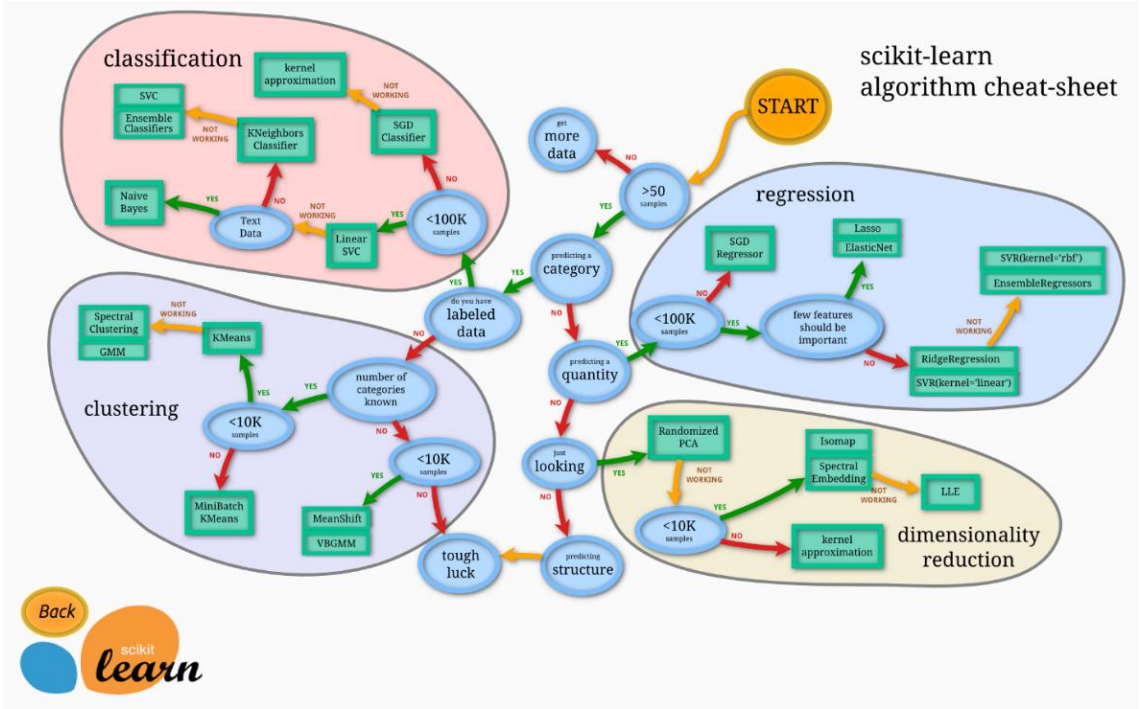
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Scikit-learn 功能

- 六大功能
  - 分類（Classification）
  - 迴歸（Regression）
  - 分群（Clustering）
  - 降維（Dimensionality reduction）
  - 模式選擇（Model selection）
  - 資料前處理（Preprocessing）

演算法選擇



http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

分類（Classification）

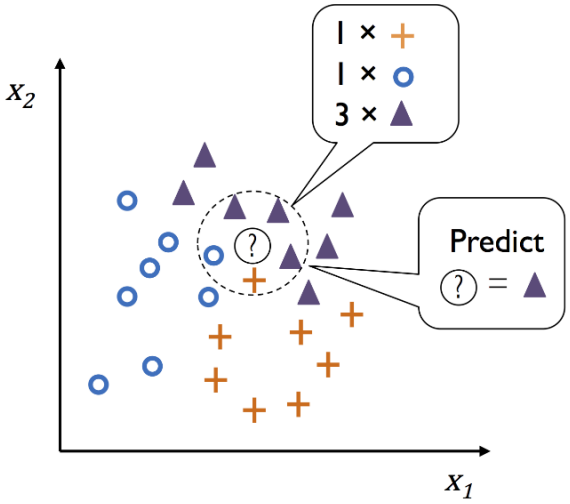
- 最近距離分群法（KNN）
- 羅吉斯迴歸（Logistic Regression）
- 支持向量機（Support Vector Machine）
- 決策樹（Decision Tree）
- 隨機森林（Random Forest）

分類（Classification）

- **最近距離分群法（KNN）**
- 羅吉斯迴歸（Logistic Regression）
- 支持向量機（Support Vector Machine）
- 決策樹（Decision Tree）
- 隨機森林（Random Forest）

K-nearest neighbors（KNN）

- 尋找距離預測值最近的 N 個樣本點
- 以多數決（Majority Voting）決定所屬分類。

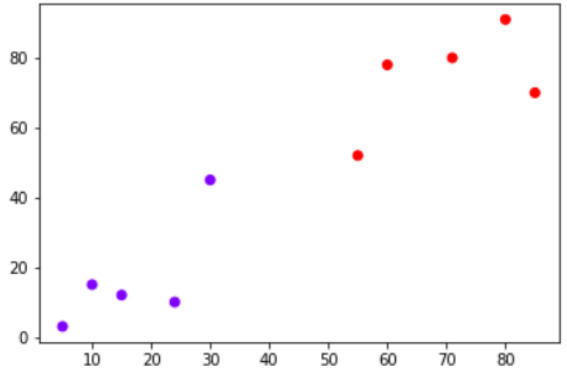


K-nearest neighbors（KNN）

- KNN 演算法是一種監督機器學習算法。
- KNN 演算法沒有專門的訓練階段。
  - 計算新數據點到所有其他訓練數據點的距離。
  - 距離可以是任何類型。
  - 它選擇 K 最近的數據點，其中 K 可以是任何整數。
  - 它將數據點分配給大多數 K 個數據點所屬的類型。

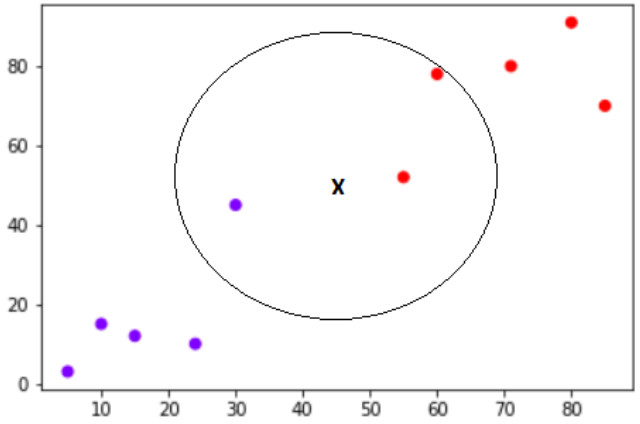
K-nearest neighbors（KNN）

- 假設您有一個包含兩種資料的資料集。
- 您的任務是將資料分為藍色或紅色。
- 假設資料點 X 的坐標值是 x = 45和 y = 50。
- 假設 K 的值為3 。
- KNN 演算法計算點X與所有點的距離。
- 然後它找到距離點X的距離最近的3個最近點。

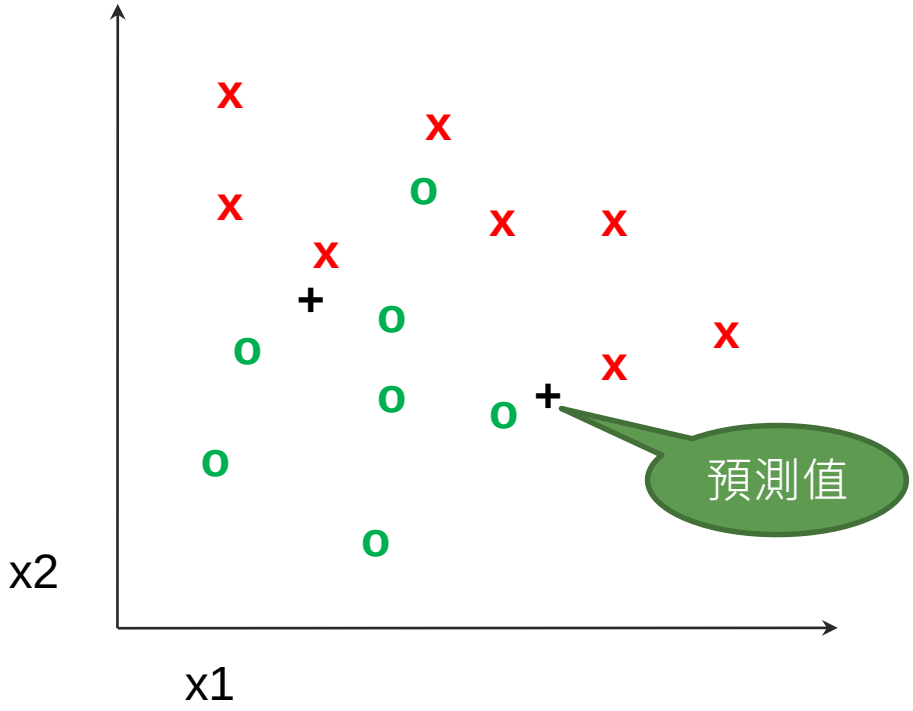


K-nearest neighbors（KNN）

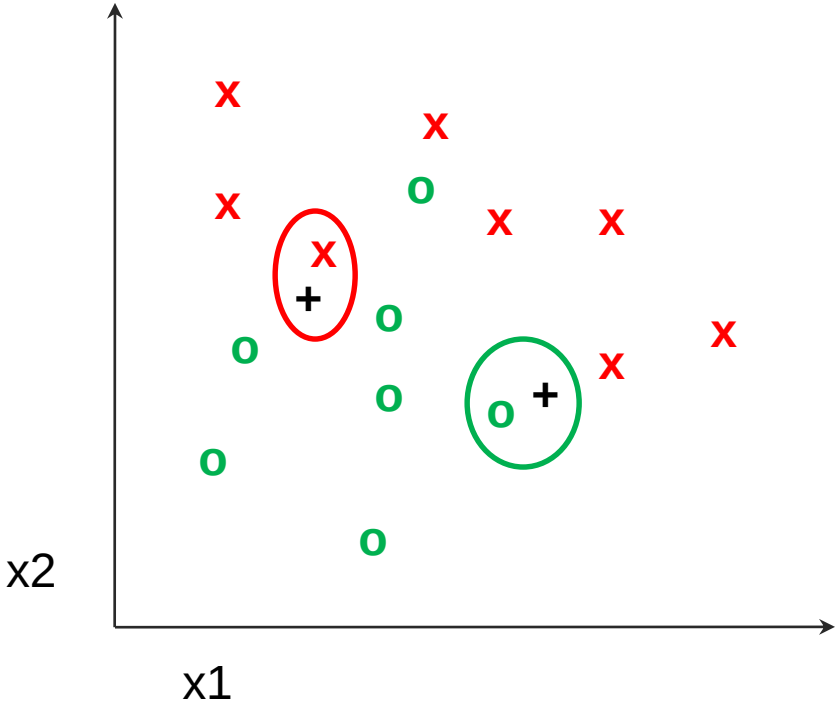
- KNN 算法的最後一步是將新點分配給三個最近點中的大多數所屬的類。
- 從圖中可以看到三個最近點
  - 兩個屬於紅色
  - 一個屬於藍色
  - 新數據點將被歸類為紅色



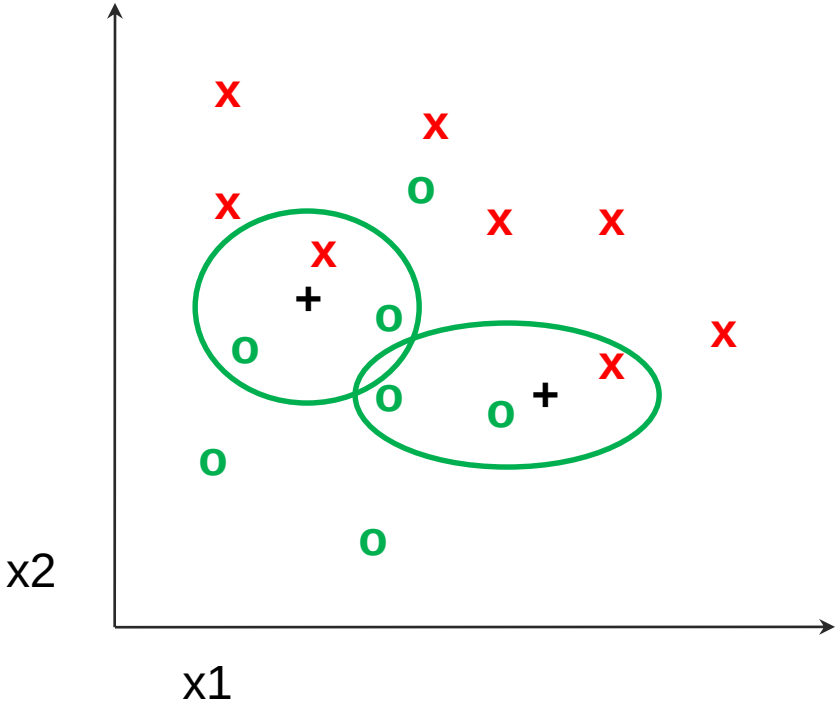
圖解



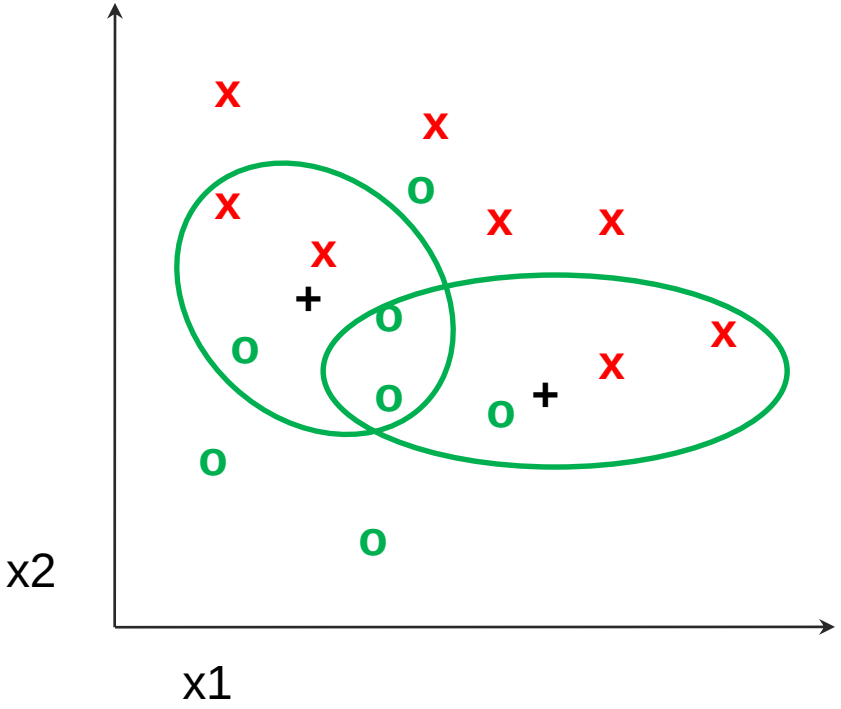
N = 1



N = 3



N = 5



實作

- 程式碼：KNN_test.ipynb
- 測試
  - 調整 N，觀察準確率變化

```
neighborpoint=knn.kneighbors(iris_x_test,5,False)
```

KNN 的優缺點

- KNN 的優點
  - 它非常容易實現。
  - 進行預測之前不需要訓練。這使得 KNN 演算法比需要訓練的其他演算法快得多。
  - 由於演算法在進行預測之前不需要訓練，因此可以快速地添加新資料。
  - 實現 KNN 只需要兩個參數，即 K 值和距離函數。
- KNN 的缺點
  - 不適用於高維度資料，因為具有大量維度，演算法難以計算每個維度的距離。
  - 對於大型資料集具有高預測成本。這是因為在大型資料集中計算新點與每個現有點之間的距離的成本變得更高。
  - 不適用於分類特徵，因為難以找到具有分類特徵的維度之間的距離。

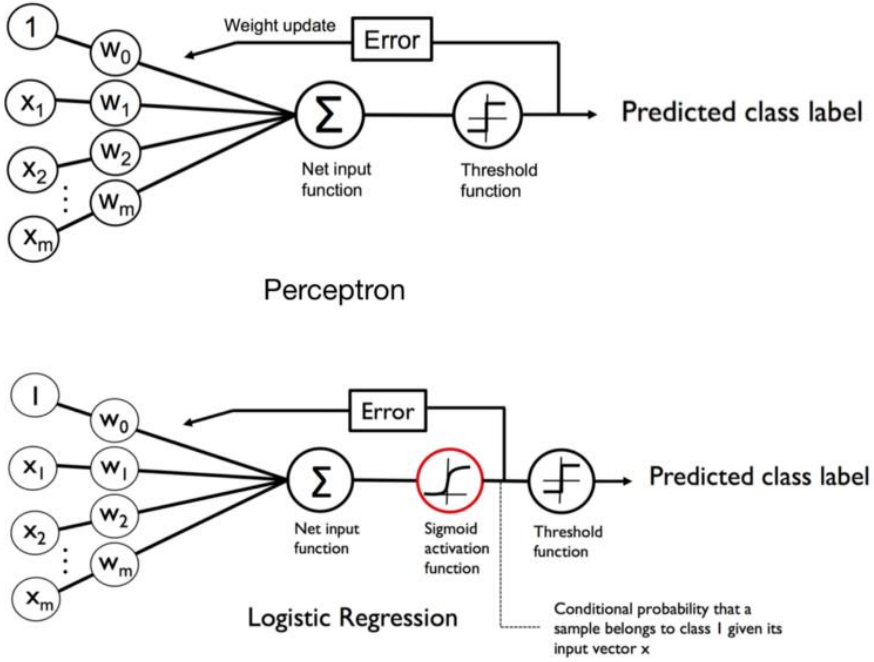
分類（Classification）

- 最近距離分群法（KNN）
- **羅吉斯迴歸（Logistic Regression）**
- 支持向量機（Support Vector Machine）
- 決策樹（Decision Tree）
- 隨機森林（Random Forest）

由 Perceptron 延伸

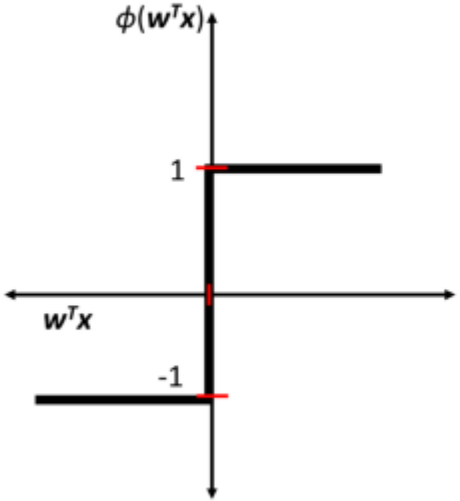
- Perceptron 感知器可以讓我們二元分類
  - 但我們只能知道預測結果是 A 還是 B。
  - 沒辦法知道是 A、是 B 的機率是多少。
- 機率判斷於生活中非常常見，比如說我們要根據今天的溫度、濕度、風向來預測明天的天氣，通常我們會需要知道明天是晴天的機率以及雨天的機率，來決定是否帶傘具出門。
- 使用 Logistic Regression 就可以幫我們達成這樣的目標
- 兩種分類方式差別
  - Perceptron 是根據 w0*x0+w1*x1+…+wn*xn >0 或 ≤0 來判斷成 A 或 B 類。
  - Logistic Regression 則是一個平滑的曲線，當 w0*x0+w1*x1+…+wn*xn 越大時，判斷成 A 類的機率越大，越小時判斷成 A 類的機率越小。

由 Perceptron 延伸

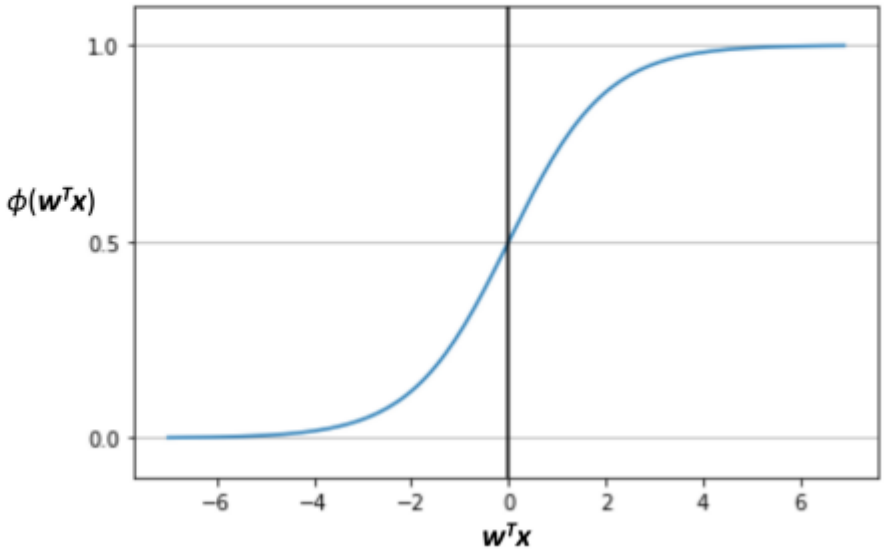


由 Perceptron 延伸

Perceptron



Logistic Regression



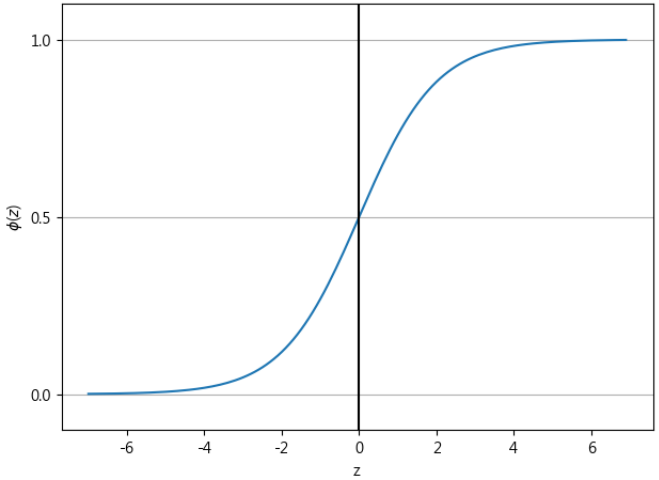
Solver 參數

- 若執行 model=LogisticRegression( ) 後，提示錯誤：
  - FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning
- Logistic的solver 必須加入以下幾種參數之一：
  - 'liblinear', 'newton-cg', 'lbfgs', 'sag', 'saga'
  - model = LogisticRegression(solver='liblinear')

實作

- 程式碼：
  - sigmoid.py
  - LogisticRegression_test.py
- 測試
  - 調整 X，觀察準確率變化

    X = iris.data[:, 1:]

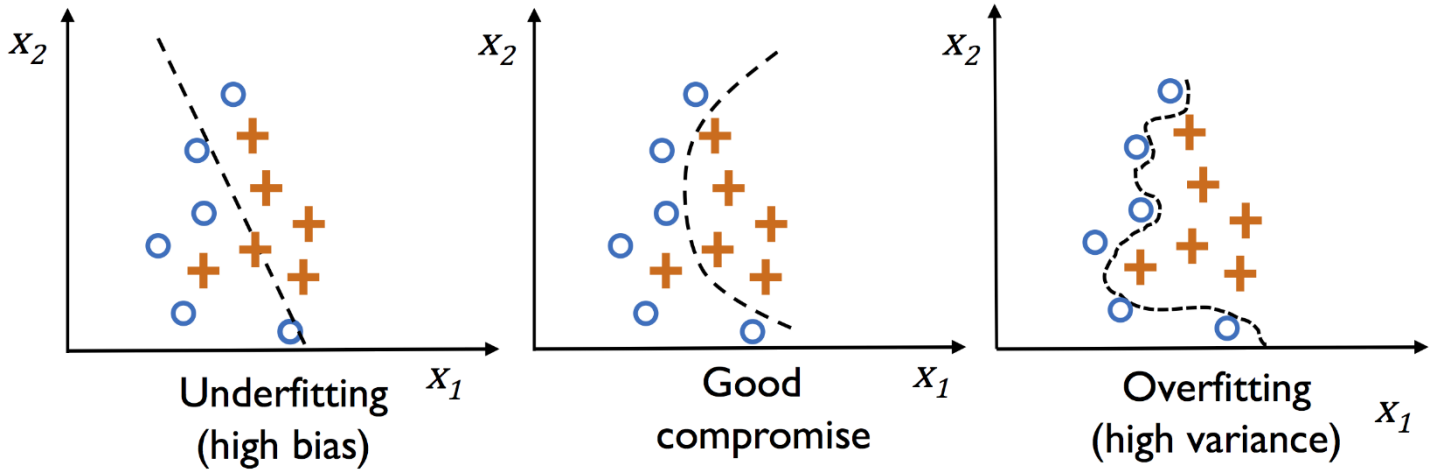


  - 調整 C（Inverse of regularization strength），觀察準確率變化

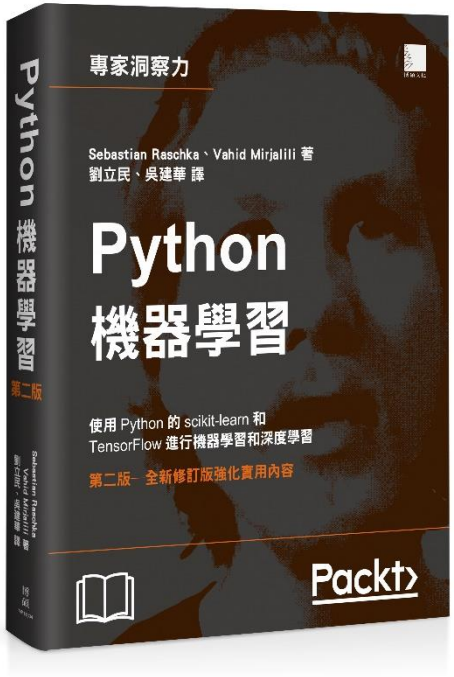
    C 需介於 (0,1)，越小表 regularization 越強

正規化（Regularization）

- 過度擬合（Overfitting）：模型在訓練時表現良好，但實際預測效果欠佳
- 正規化可避免過度擬合



參考用書



- 書名：Python機器學習（第二版）

  http://www.drmaster.com.tw/bookinfo.asp?BookID=MP11804
- 作者：Sebastian Raschka, Vahid Mirjalili ISBN
- 譯者：劉立民、吳建華
- 出版社：博碩

問卷

http://www.pcschoolonline.com.tw

www.pcschoolonline.com.tw