

Day 38

機器學習

# Regression 模型-程式碼



# Scikit-learn 中的 linear regression

---

- 使用 Sklearn 套件中的線性回歸非常簡單
- 第一行引入所需的套件
- 第二行建立線性回歸模型，並將  $X, y$  資料送進模型中訓練
- Reg 就是訓練好的模型，可以用 `.predict` 來預測新資料

```
from sklearn.linear_model import LinearRegression  
reg = LinearRegression().fit(X, y)
```

# Scikit-learn 中的 Logistic Regression

---

- 使用 Sklearn 套件中的 Logistic 回歸也非常簡單
- 第一行引入所需的套件
- 第二行建立 Logistic 回歸模型，並將 X, y 資料送進模型中訓練
- Reg 就是訓練好的模型，可以用 .predict 來預測新資料

```
from sklearn.linear_model import LogisticRegression  
reg = LogisticRegression().fit(X, y)
```



# Scikit-learn 中的 Logistic Regression 參數

---

- Penalty: “L1”, “L2”。使用 L1 或 L2 的正則化參數，後續有更詳細介紹
- C: 正則化的強度，數字越小，模型越簡單
- Solver: 對損失函數不同的優化方法。更詳細的內容請參考[連結](#)
- Multi-class: 選擇 one-vs-rest 或 multi-nominal 分類方式，當目標是 multi-class 時要特別注意，若有 10 個 class，ovr 是訓練 10 個二分類模型，第一個模型負責分類 (class1, non-class1)；第二個負責 (class2, non-class2)，以此類推。multi-nominal 是直接訓練多分類模型。詳細參考[連結](#)

# 常見問題

---

Q: Logit regression 跟 Logistic regression 有什麼差別？

A: Logit function 跟 Logistic function 互為反函數，但這兩個 regression 其實是指同樣的模型，更詳細可參考[連結](#)

# 解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業  
開始解題

