AI 神經網絡深層學習與交易策略

應用實作班

Deep Learning on Trading Strategy Workshop

昀騰金融科技

技術長

董夢雲 博士

目　　錄

Part II 時間數列與機器學習測略介紹

七、時間數列預測基礎：穩態、可預測性與共整合

八、主成分分析與時間數列預測

九、ARIMA 與 GARCH 的合併預測

十、機器學習模型(一)：羅吉斯迴歸與鑑別分析

十一、機器學習模型(二)：支持向量機

十二、機器學習模型(三)：決策樹與隨機森林

十三、機器學習模型(四)：配對交易的選對

Part II 時間數列與機器學習測略介紹

主題十二、機器學習模型：

決策樹與隨機森林

一、決策樹

二、隨機森林

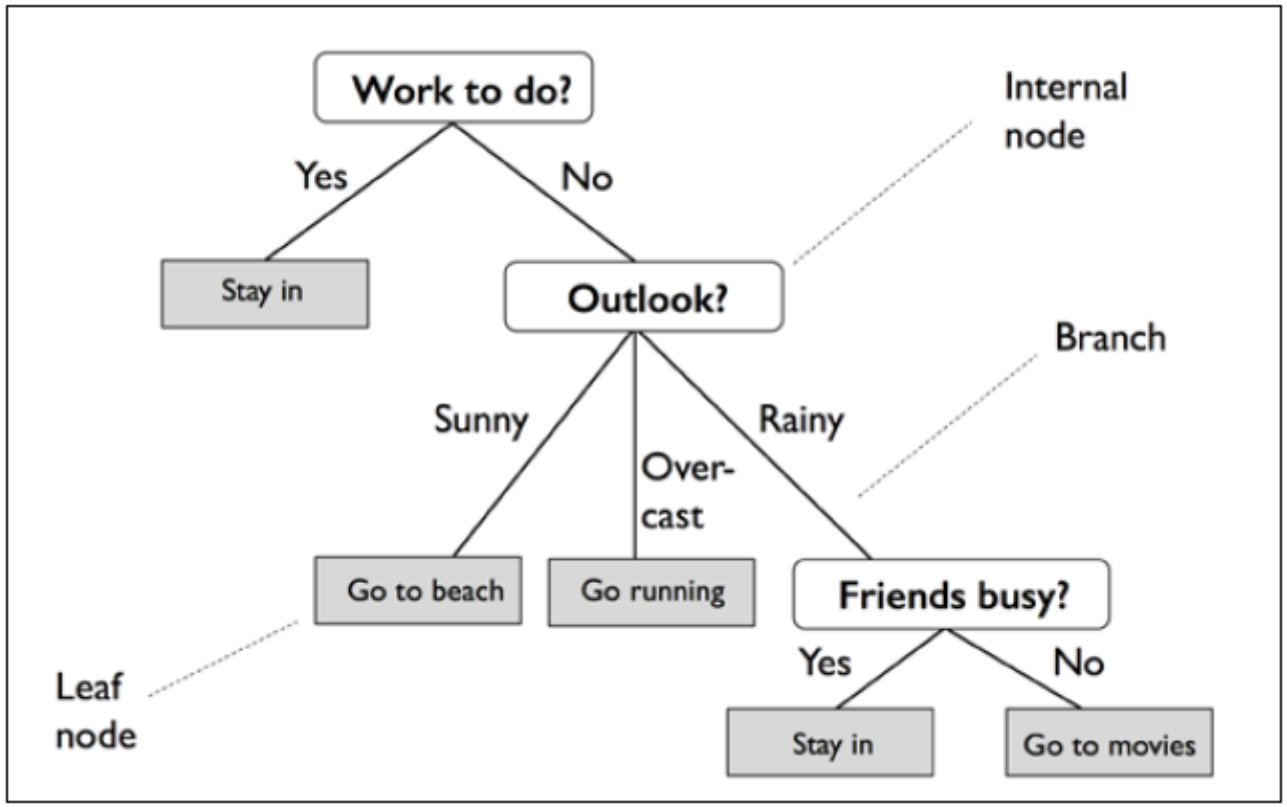
三、程式範例

◆ 決策樹分類器（Decision Tree Classifiers）與 Logistic 迴歸最大的不同點是，前者為多元分類（Multiclass classification）演算法。

- 隨機森林分類器屬於建構於決策樹之上的整體學習應用，每一個基本分類器都是一個決策樹。
- 經由詢問一系列問題，分割數據，並據此作出決策。

一、決策樹

- 決策樹簡單來說就是把資料集以一棵樹呈現。
  - 一顆決策樹包含節點、分支、葉子，並根據條件將子葉分支。
  - 下圖為某一天活動的決定邏輯。



- 根據訓練集中的特徵，前兩天的價格(P_{t-2}, P_{t-1})，決策樹會學出一系列的問題，並據此推論樣本的類別標籤。
  - 例如，前一天的報酬率是否大於零，$P_{t-2} - P_{t-1} > 0$。
  - P_t 是上漲或下跌。
- 操作原則是，依據特徵值將數據分割到不同邊，使我們得到較大的資訊增益(Information Gain, IG)。
  - 從根部開始，直到葉面節點。
  - 實務上，可能產生很深的決策樹，導致過度適合的現象。
    - ✓ 通常會限制樹的高度來預防此現象。

(一)最大資訊增益

◆ 目標函數：

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

- ➢ 父節點的不純度 減去 子節點(加權的)不純度的加總。
  - ✓ f：節點用來分割的特徵
  - ✓ Dp：父節點的數據集，Dj：第 j 個子節點的數據集
  - ✓ I：不存度(Impurity)的衡量
  - ✓ Np：父節點的樣本數，Nj：第 j 個子節點的樣本數
- ➢ 越大越好，子的不純度低，有效分類。

◆ 一般是以二元樹來實作

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

- ➢ 左右兩個節點。

甲、不純度三種衡量，

- 分類錯誤率，(I_E)

$$I_E(t) = 1 - \max\{p(i \mid t)\}$$

- Gini 不純度(I_G)

$$I_G(t) = \sum_{i=1}^{c} p(i \mid t)\big(1 - p(i \mid t)\big) = 1 - \sum_{i=1}^{c} p(i \mid t)^2$$

  - 全部集中在一類，分不出來，I_G=0。
  - c=2，完美分離，I_G=0.5，最大。

$$I_G(t) = \sum_{i=1}^{2} p(i \mid t)\big(1 - p(i \mid t)\big) = 1 - \sum_{i=1}^{2} 0.5^2 = 0.5$$

◆ Entropy(I_H)，熵，的定義：

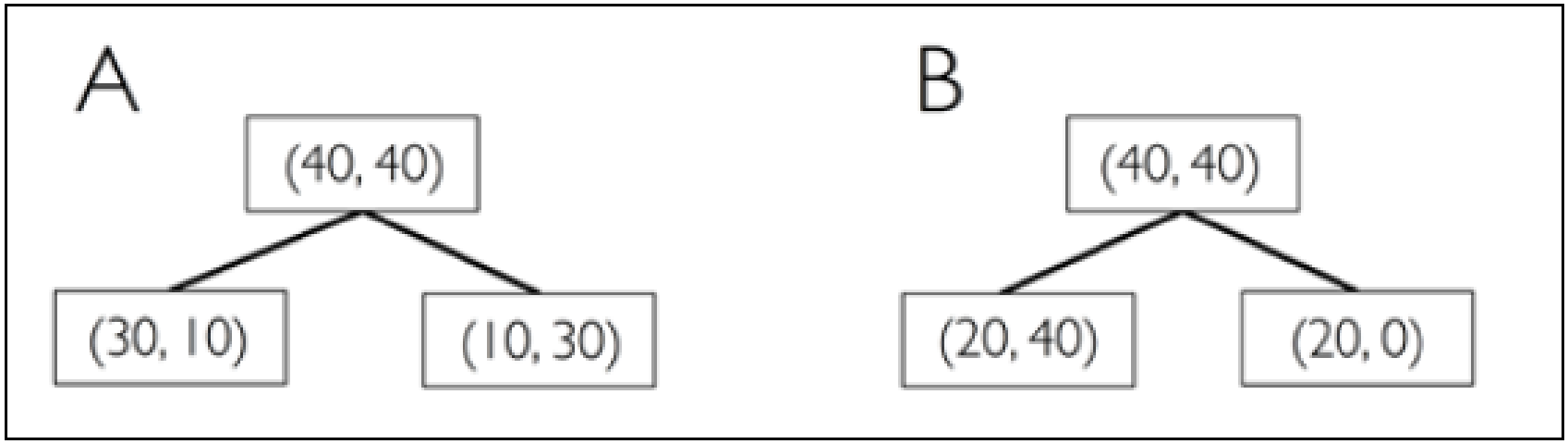
$$I_{\mathrm{H}}(t) = -\sum_{i=1}^{c} p(i \mid t) \log_2 p(i \mid t)$$

- p(i|t)：某特定 t 節點中，數於類別 c 的樣本比率。
- 如果都屬於同一類，p(i|t)=1，$\log_2 p(i|t)=0$。$I_H=0$，最小，分不出來。

(二)範例

◆ 分類方式：I_E 範例

- 初始有 40 個類別 1 的樣本，40 個類別 2 的樣本。分到兩個節點。
- 以 A 為例，父節點若全當類別 1，40 個錯誤。$I_E(D_p) = 1 - 40/80 = 0.5$。
- 以 A 左下子節點為例，若全當類別 2，30 個錯誤。$I_E(D_{left}) = 1 - 30/40 = 0.25$
- 以 A 右下子節點為例，若全當類別 1，30 個錯誤。$I_E(D_{right}) = 1 - 30/40 = 0.25$



◆ IE：A and B are same.

$$I_E\left(D_p\right)=1-0.5=0.5$$

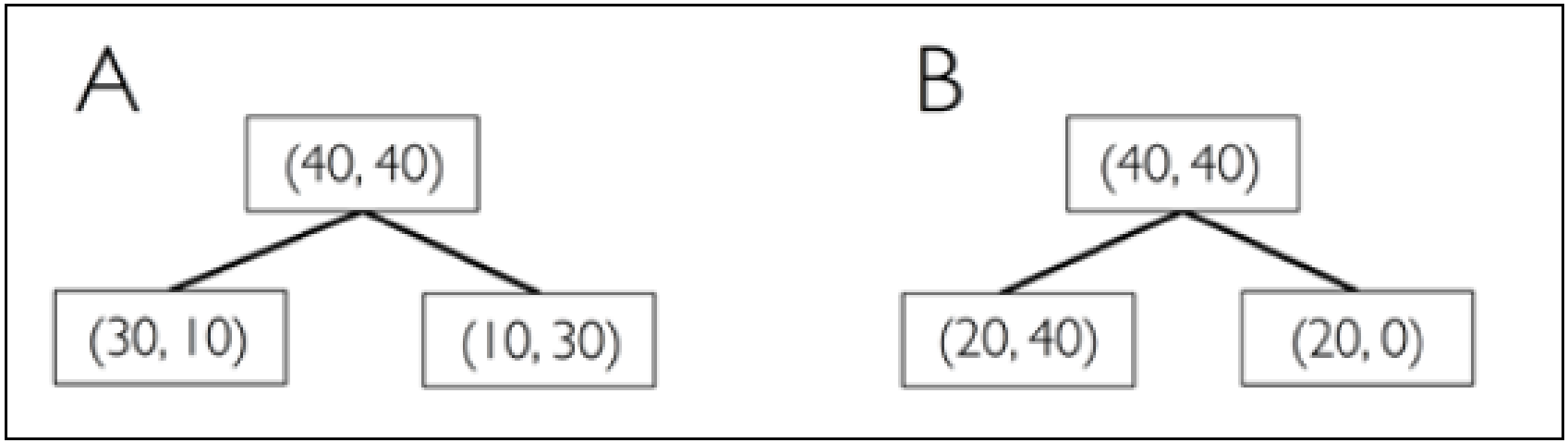
$$A: I_E\left(D_{left}\right)=1-\frac{3}{4}=0.25 \qquad B: I_E\left(D_{left}\right)=1-\frac{4}{6}=\frac{1}{3}$$

$$A: I_E\left(D_{right}\right)=1-\frac{3}{4}=0.25 \qquad B: I_E\left(D_{right}\right)=1-1=0$$

$$A: IG_E=0.5-\frac{4}{8}0.25-\frac{4}{8}0.25=0.25 \qquad B: IG_E=0.5-\frac{6}{8}\times\frac{1}{3}-0=0.25$$

◆ 分類方式：I_G 範例

- 初始有 40 個類別 1 的樣本，40 個類別 2 的樣本。分到兩個節點。
- 以 A 為例，父節點若全當類別 1。$I_G(D_p) = 1 - (40/80)^2 - (40/80)^2$
- 以 A 左下子節點為例。$I_G(D_{left}) = 1 - (30/40)^2 - (10/40)^2$
- 以 A 右下子節點為例。$I_G(D_{right}) = 1 - (10/40)^2 - (30/40)^2$



◆ IG：B is better.

$$I_G(D_p) = 1 - \left(0.5^2 + 0.5^2\right) = 0.5$$

$$A: I_G(D_{left}) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0.375$$

$$B: I_G(D_{left}) = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = \frac{4}{9} = 0.\overline{4}$$

$$A: I_G(D_{right}) = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0.375$$

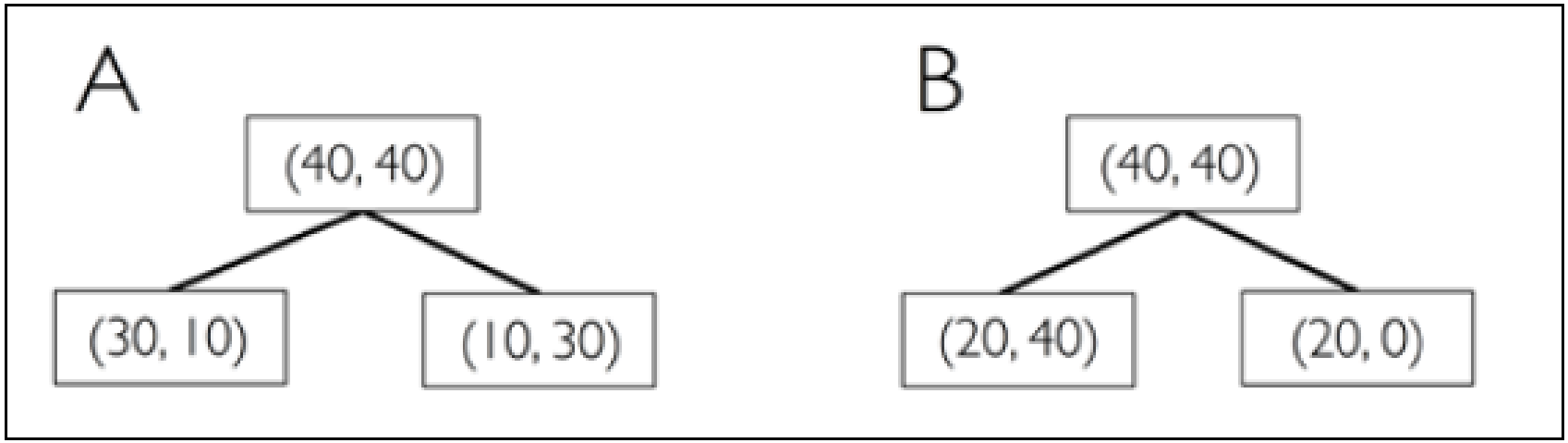
$$B: I_G(D_{right}) = 1 - \left(1^2 + 0^2\right) = 0$$

$$A: IG_G = 0.5 - \frac{4}{8}0.375 - \frac{4}{8}0.375 = 0.125$$

$$B: IG_G = 0.5 - \frac{6}{8}0.\overline{4} - 0 = 0.1\overline{6}$$

◆ 分類方式：I_H 範例

- 初始有 40 個類別 1 的樣本，40 個類別 2 的樣本。分到兩個節點。
- 以 A 為例。$I_H(D_p) = -[(40/80)*\log(40/80) + (40/80)*\log(40/80)]$
- 以 A 左下子節點為例。$I_H(D_{left}) = -[(30/40)*\log(30/40) + (10/40)*\log(10/40)]$
- 以 A 右下子節點為例。$I_H(D_{right}) = -[(10/40)*\log(10/40)] + (30/40)*\log(30/40)]$



◆ IH：B is better.

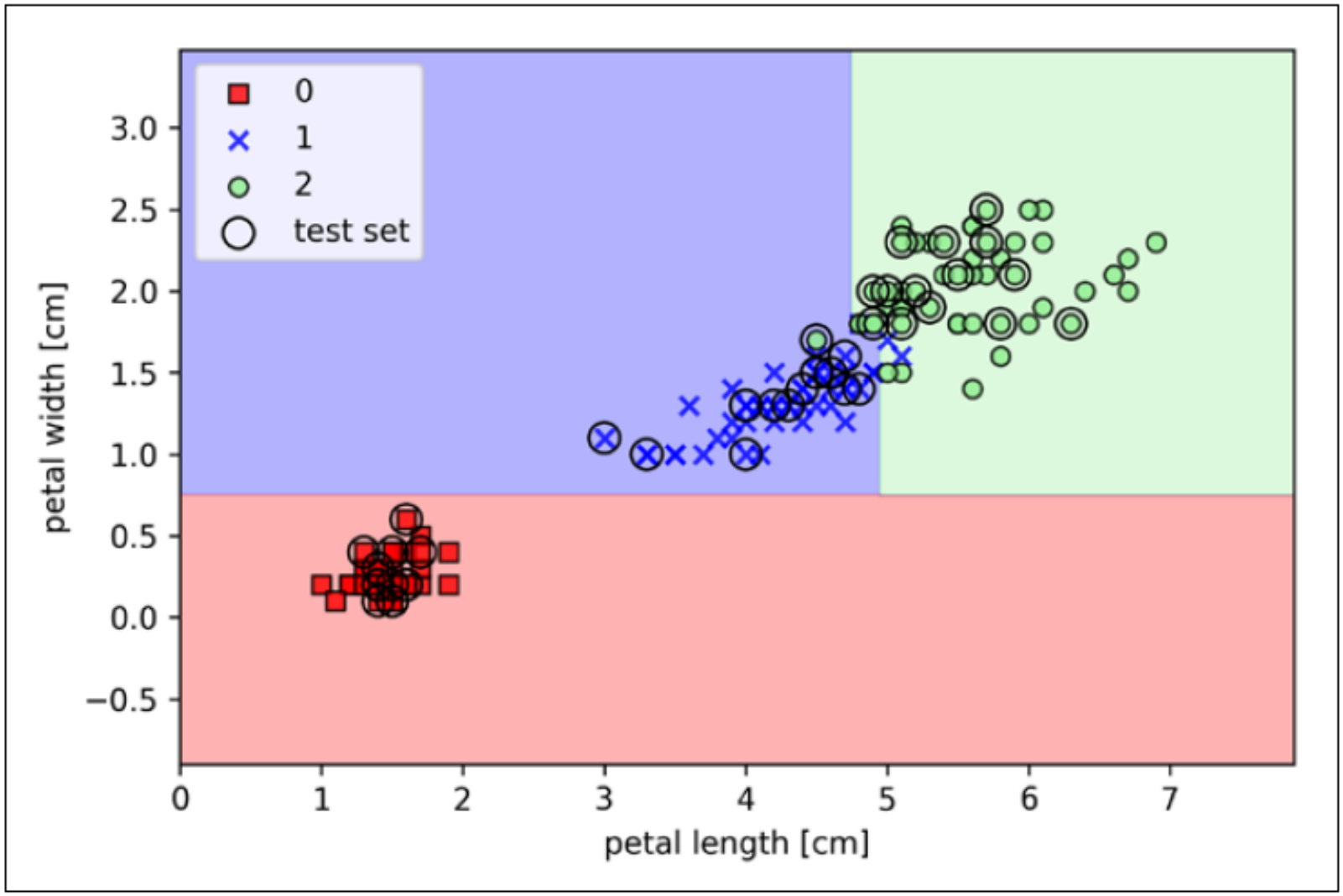
$$I_H(D_p) = -(0.5\log_2(0.5) + 0.5\log_2(0.5)) = 1$$

$$A: I_H(D_{left}) = -\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.81 \qquad B: I_H(D_{left}) = -\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) = 0.92$$

$$A: I_H(D_{right}) = -\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) = 0.81 \qquad B: I_H(D_{right}) = 0$$

$$A: IG_H = 1 - \frac{4}{8}0.81 - \frac{4}{8}0.81 = 0.19 \qquad B: IG_H = 1 - \frac{6}{8}0.92 - 0 = 0.31$$

◆ 兩個特徵，三個類別的分割。鳶尾花(Iris)資料集。



二、隨機森林

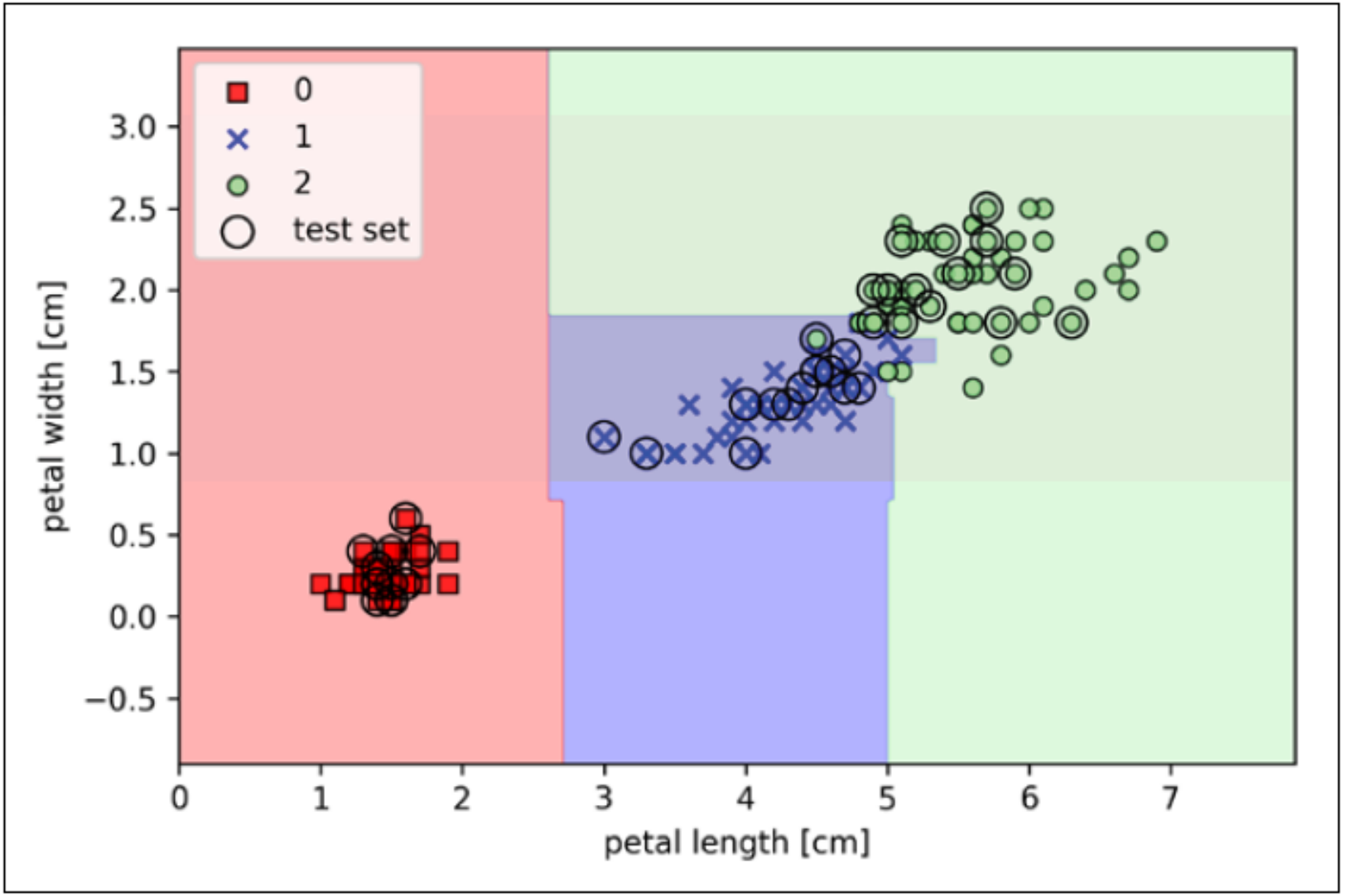
◆ 隨機森林(Random Forests)被視為是多個決策樹結合成的一個整體，分類效能高、可擴展性高、易於使用。

- ➢ 結合多個具有高變異的深度決策樹，將結果平均，來建構一個更堅固的模型。
- ➢ 該模型一般化誤差較低，也不會有過度適合的問題。
  - ✓ 決策樹經常會遇到擬合的問題，
  - ✓ 在隨機森林演算法中，因為 Forest 是由多個 Trees 所組成，所以對隨機森林反而希望以計算速度快為要點，不會追求單顆 Tree 適合的情形。

◆ 步驟：

- 一：定義大小為 n 的隨機「自助」(Bootstrap)樣本。
  - ✓ 從訓練集樣本中，採用放回式(取出會放回)隨機選擇 n 個樣本。
- 二：從自助樣本中導出決策樹。對每一節點：
  - ✓ (一)隨機選擇 d 個特徵(不放回式)
  - ✓ (二)使用特徵分割節點，依據目標函數(IG 最大化)找出最佳方式
- 三、重複 k 次步驟一、二
- 四、彙總所有決策樹的預測，以多數決的方式，來指定類別標簽。

◆ 兩個特徵，三個類別的分割。鳶尾花(Iris)資料集。



三、程式範例

```
# The test data is split into two parts: Before and after 1st Jan 2018.
start_test = '2018-01-01' # datetime.datetime(2018, 1, 1)

# Create training and test sets
X_train = X[X.index < start_test] # 503
y_train = y[y.index < start_test] # 503

#print(X_train)
#print(y_train)

X_test = X[X.index >= start_test] # 251
y_test = y[y.index >= start_test] # 251
#print(X_test)
#print(y_test)
```

```
# Create the (parametrised) models
print("Hit Rates/Confusion Matrices:\n")
models = [("RF", RandomForestClassifier(n_estimators=1000, criterion='gini',
                max_depth=None, min_samples_split=2,
                min_samples_leaf=1, max_features='auto',
                bootstrap=True, oob_score=False, n_jobs=1,
                random_state=None, verbose=0))]
# Iterate through the models
# Train each of the models on the training set
# model[1].fit(X_train, y_train)

for m in models:
   m[1].fit(X_train, y_train)
   pred = m[1].predict(X_test)
   print(m[0], m[1].score(X_test, y_test))
   print(confusion_matrix(pred, y_test))
```

```
#Create a single prediction
X_test.loc(1)
ind = pd.Index([datetime.datetime(2019, 1, 1)])
S1 = pd.Series(0.2, index=ind)
S2 = pd.Series(0.5, index=ind)
X1_test = pd.DataFrame({'Lag1':S1, 'Lag2':S2})
print(X1_test)
pred1 = m[1].predict(X1_test)
print(pred1)
```

```
Hit Rates/Confusion Matrices:
RF 0.4820717131474104
[[48 62]
 [68 73]]
            Lag1  Lag2
2019-01-01   0.2   0.5
[1.]
```