



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Class 4 – Lesson 1

使用「包裝器」方法選擇屬性

Attribute selection using the “wrapper” method

Ian H. Witten

Department of Computer Science University of
Waikato
New Zealand

Lesson 4.1: 使用「包裝器」方法選擇屬性

Class 1 探索Weka的介面；處理大數據

Class 2 離散以及文本分類

Class 3 分類規則、關聯規則、聚類

Class 4 選擇屬性以及計算成本

Class 5 神經網路，學習曲線和表現優化

Lesson 4.1 「包裝器」屬性選擇法

Lesson 4.2 The Attribute Selected Classifier

Lesson 4.3 Scheme-independent selection

Lesson 4.4 Attribute selection using ranking

Lesson 4.5 Counting the cost

Lesson 4.6 Cost-sensitive classification



Lesson 4.1: 使用「包裝器」方法選擇屬性

更少的屬性有更好的分類結果

❖ Data Mining with Weka, Lesson 1.5

- 開啟 *glass.arff*; 執行 J48 (*trees>J48*): 使用交叉驗證法，準確率67%
- 除了RI和Mg，移除其他屬性: 得到69%準確率
- 除了RI、Na、Mg、Ca和Ba，移除其他屬性: 得到74%準確率

❖ “Select attributes”面板幫我們避開了費力的實驗過程

- 開啟 *glass.arff*; 屬性評估器 *WrapperSubsetEval* ; 選擇 J48，10層交叉驗證，閾值 = -1
- 搜尋方法: *BestFirst*; 選擇 *Backward*
- 取的相同屬性子集: RI, Na, Mg, Ca, Ba: “merit” 0.74

❖ 多少的實驗?

- 設定 *searchTermination* = 1
- 共評估了36個子集
完整集合(1個); 移除一種屬性(9個); 再移除一種(8); 再移除一種(7); 再移除一種(6); 最後再移除一種(5)以檢查再刪除的屬性已經不會產生改進; $1+9+8+7+6+5 = 36$

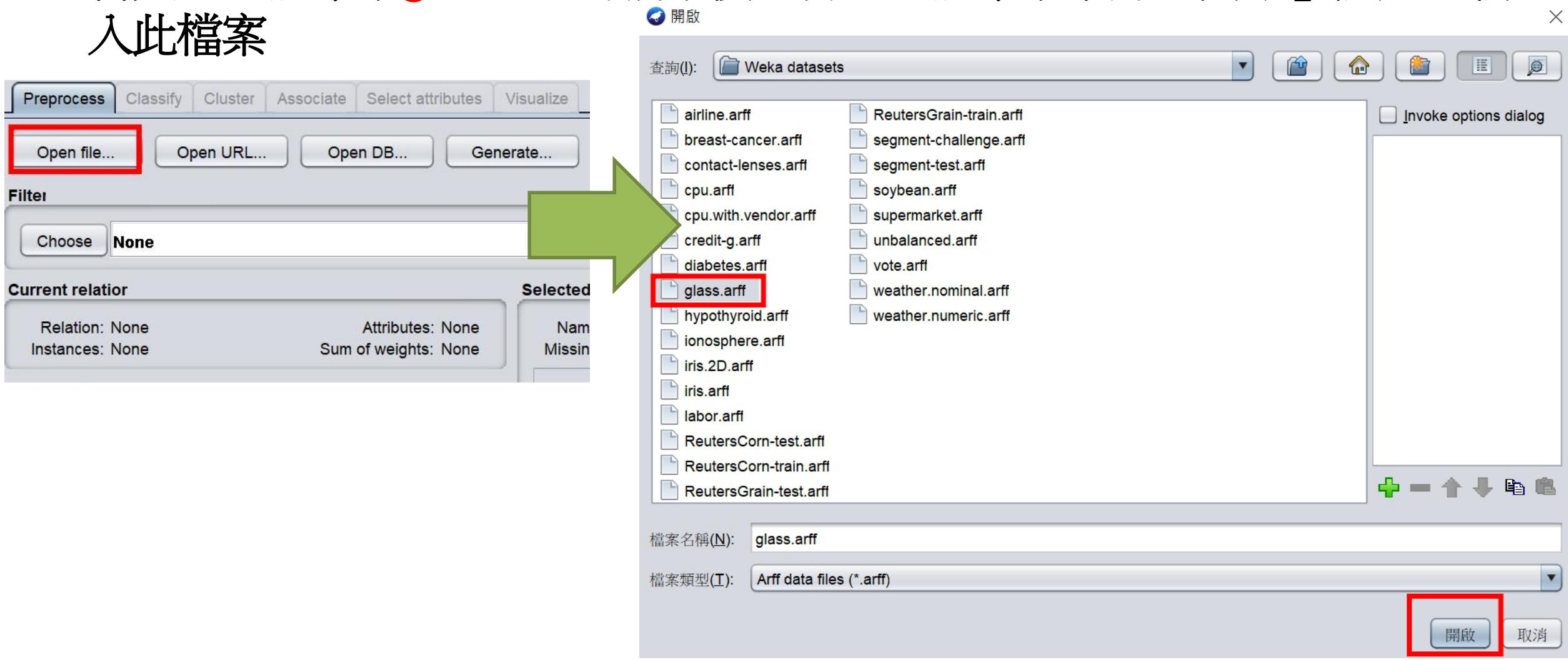
Lesson 4.1: 使用「包裝器」方法選擇屬性

1. 開啟Weka的Explorer



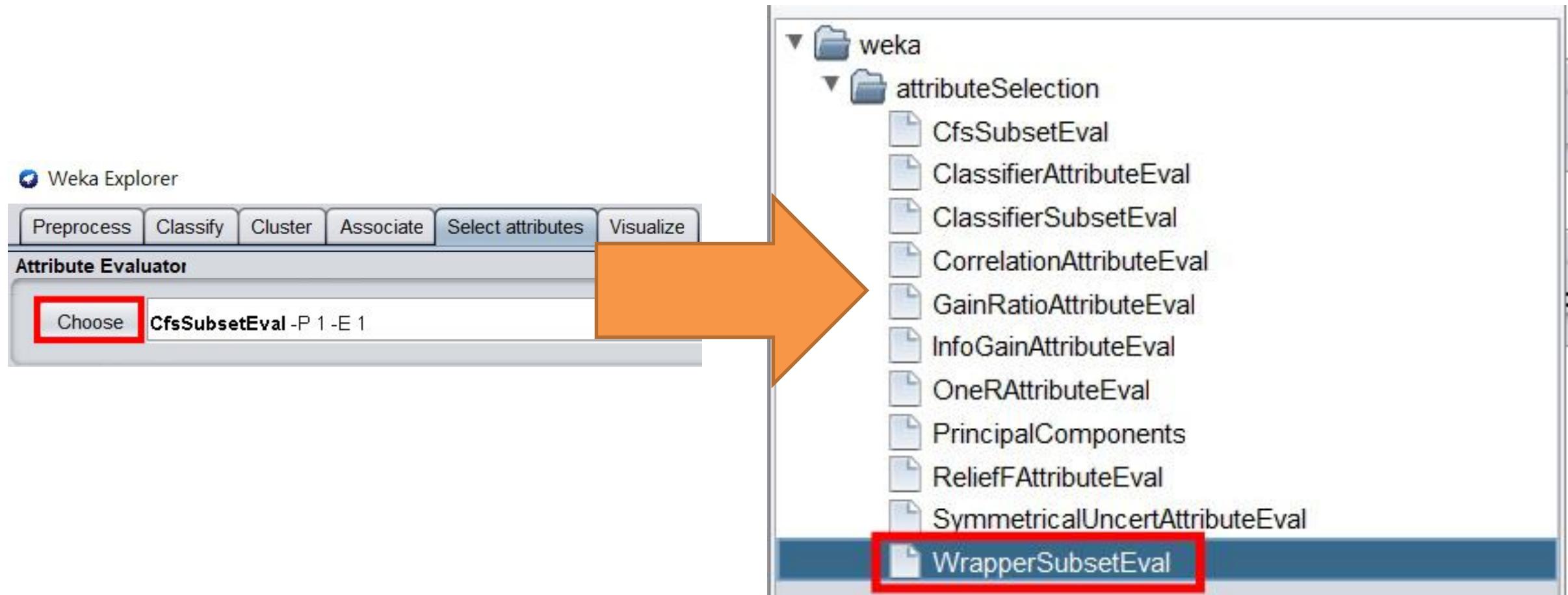
Lesson 4.1: 使用「包裝器」方法選擇屬性

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊glass.arff的檔案後，再以左鍵單擊下方「開啟」按鈕以載入此檔案



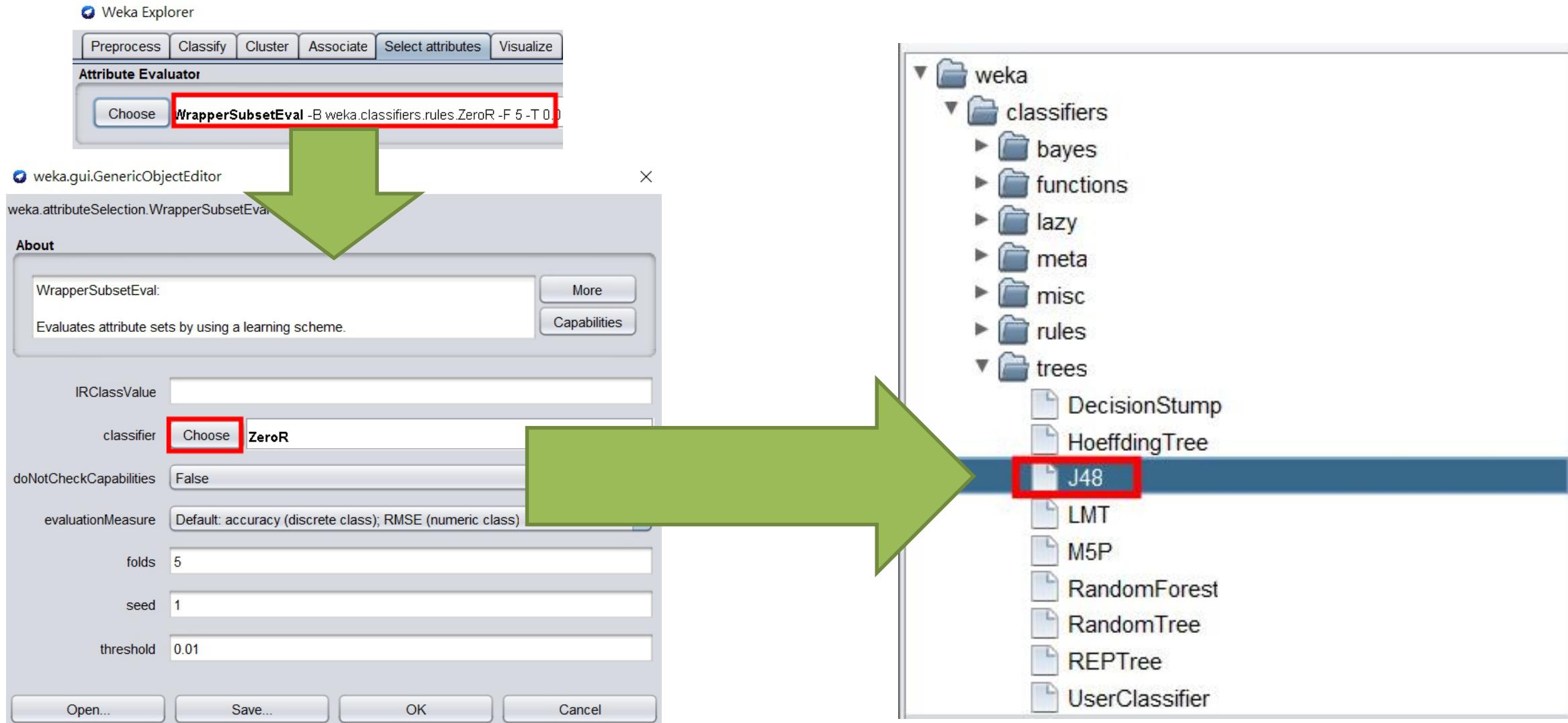
Lesson 4.1: 使用「包裝器」方法選擇屬性

3. 切換到Select attributes面板，以滑鼠左鍵點擊Choose按鈕，在彈出的選單中選擇attributeSelection資料夾下的WrapperSubsetEval屬性選擇器



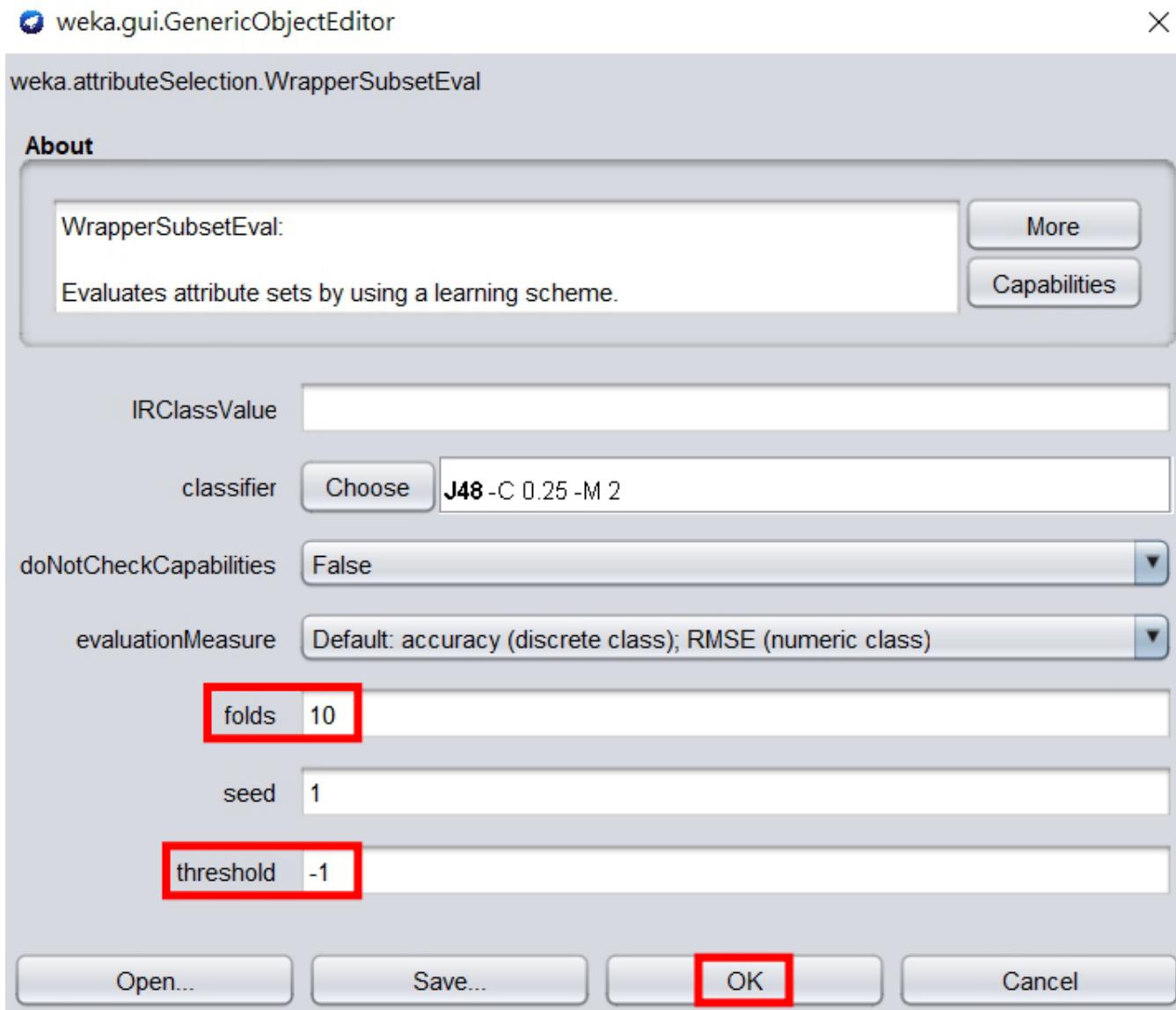
Lesson 4.1: 使用「包裝器」方法選擇屬性

4. 左鍵單擊屬性選擇器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊Choose按鈕，再以左鍵單擊J48分類器。



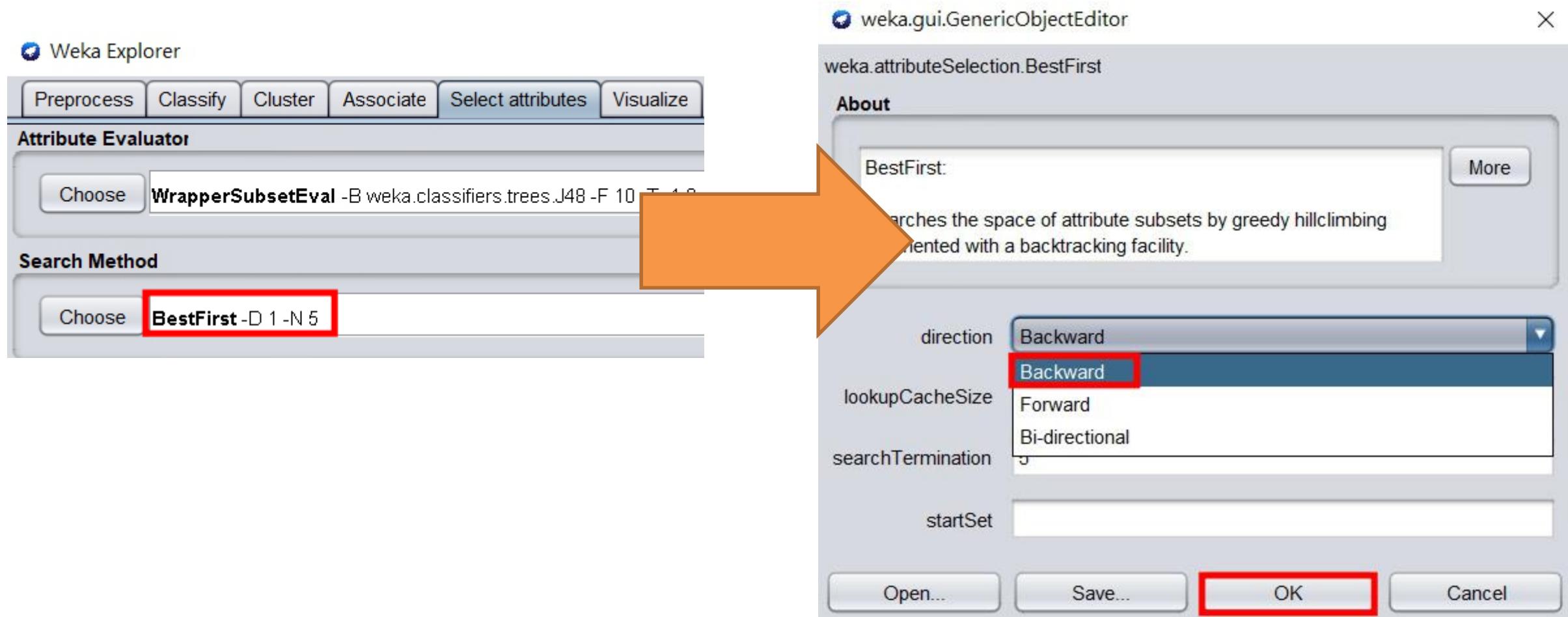
Lesson 4.1: 使用「包裝器」方法選擇屬性

5. 將參數folds改為10，參數threshold改為-1，接著左鍵單擊下方OK按鈕。



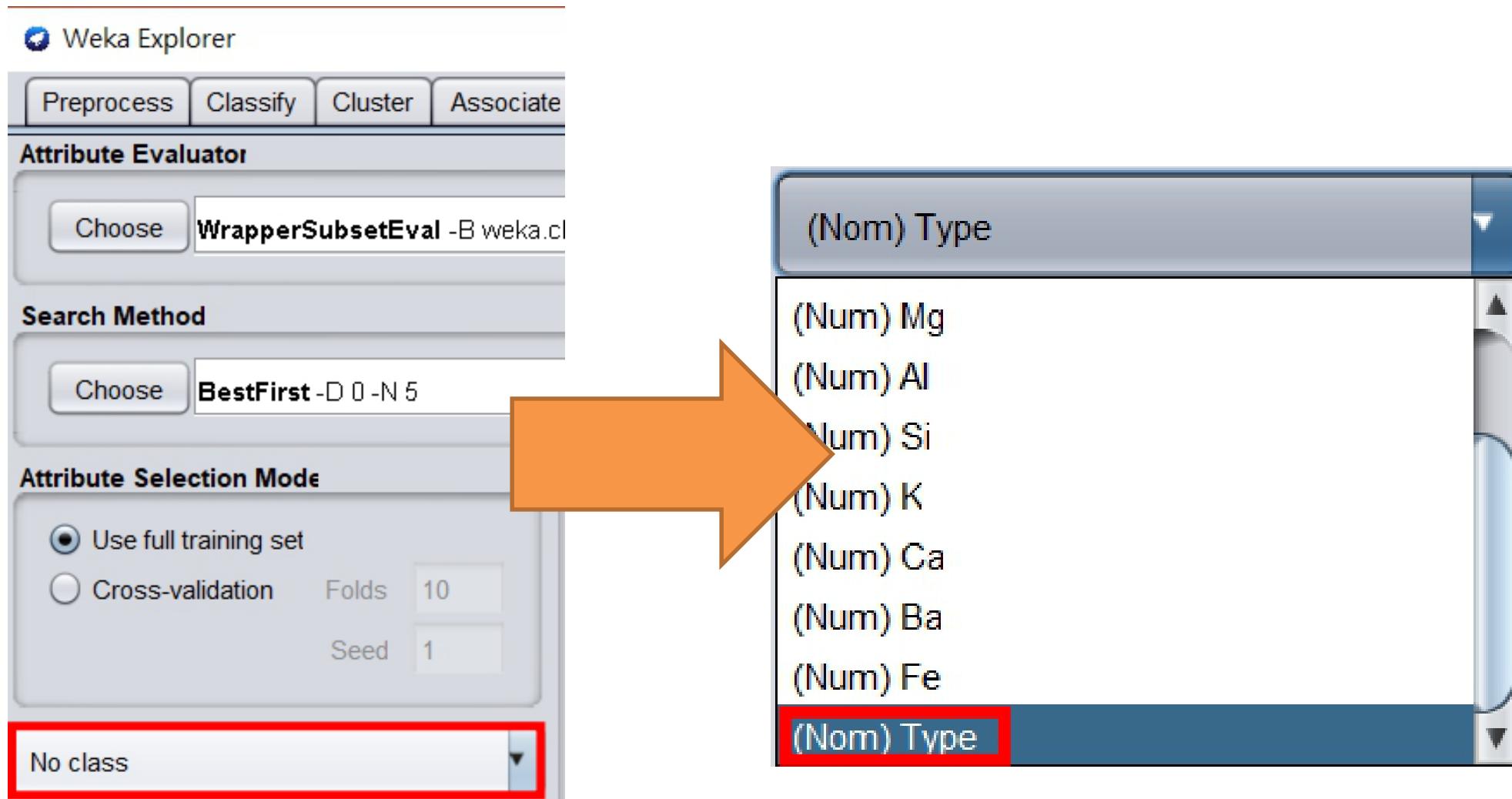
Lesson 4.1: 使用「包裝器」方法選擇屬性

6. 左鍵單擊Search Method中的搜尋方式名稱(左圖紅框處)，開啟配置視窗(右圖)。在配置視窗中左鍵單擊參數direction的下拉式選單，並以左鍵單擊Backward，然後左鍵單擊下方OK按鈕。



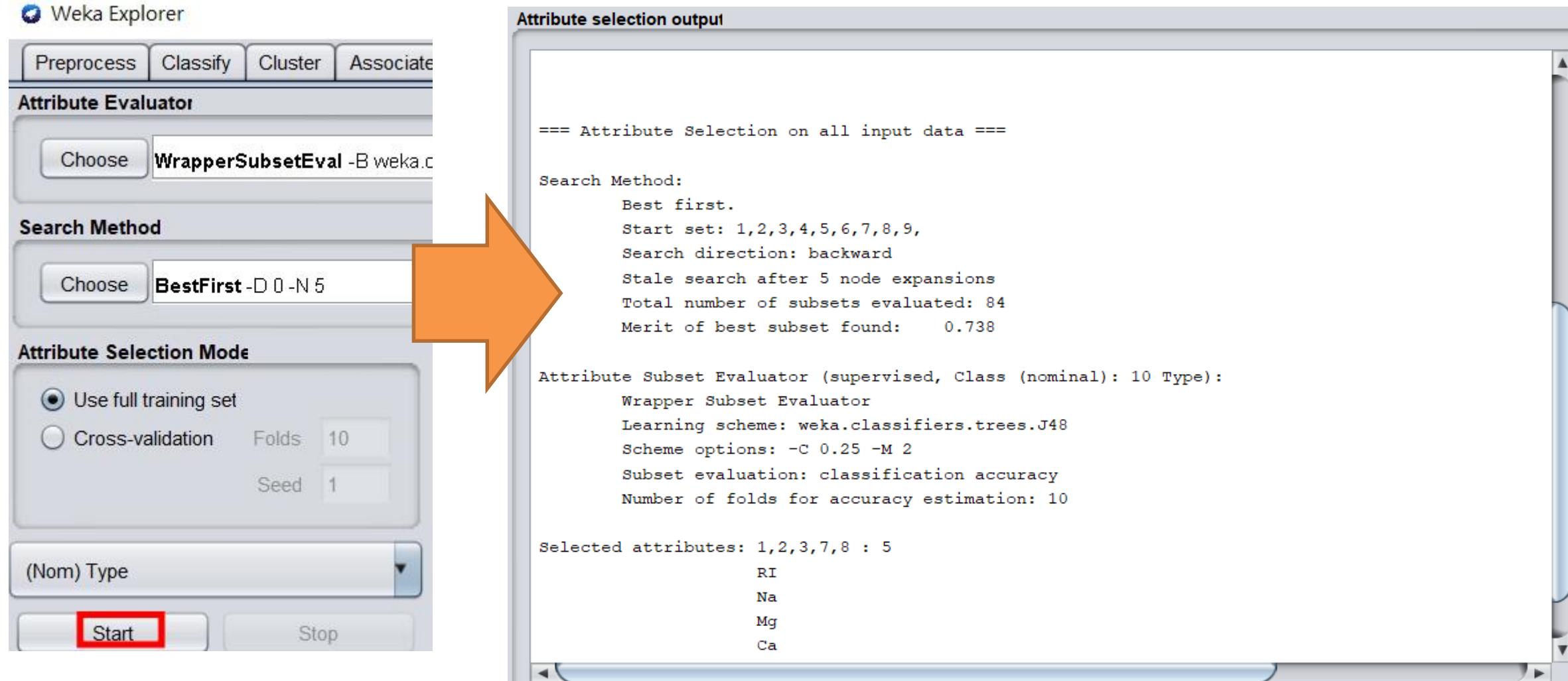
Lesson 4.1: 使用「包裝器」方法選擇屬性

7.回到Select attributes面板，左鍵單擊分類屬性的下拉式選單，並以左鍵單擊(Nom)Type選項。



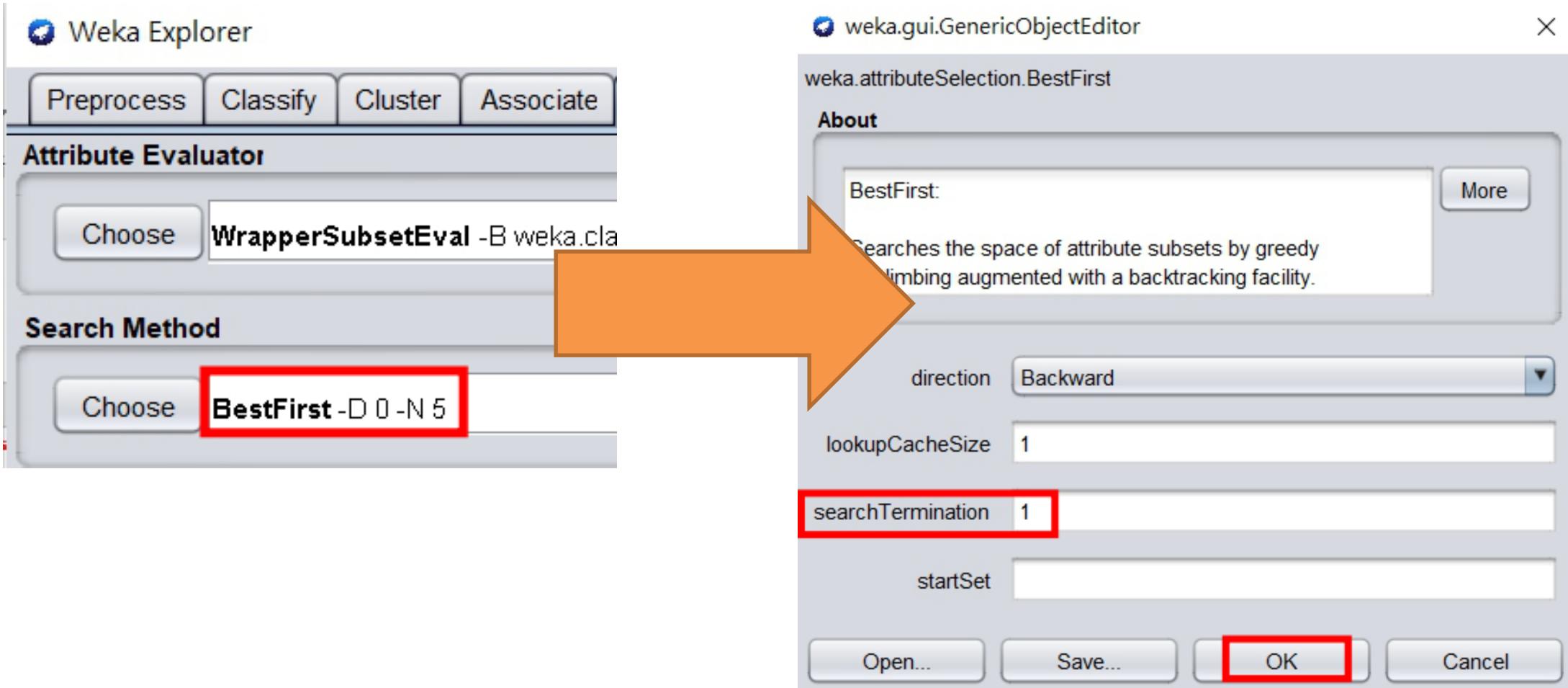
Lesson 4.1: 使用「包裝器」方法選擇屬性

8. 左鍵單擊Start按鈕，執行結果如右圖。



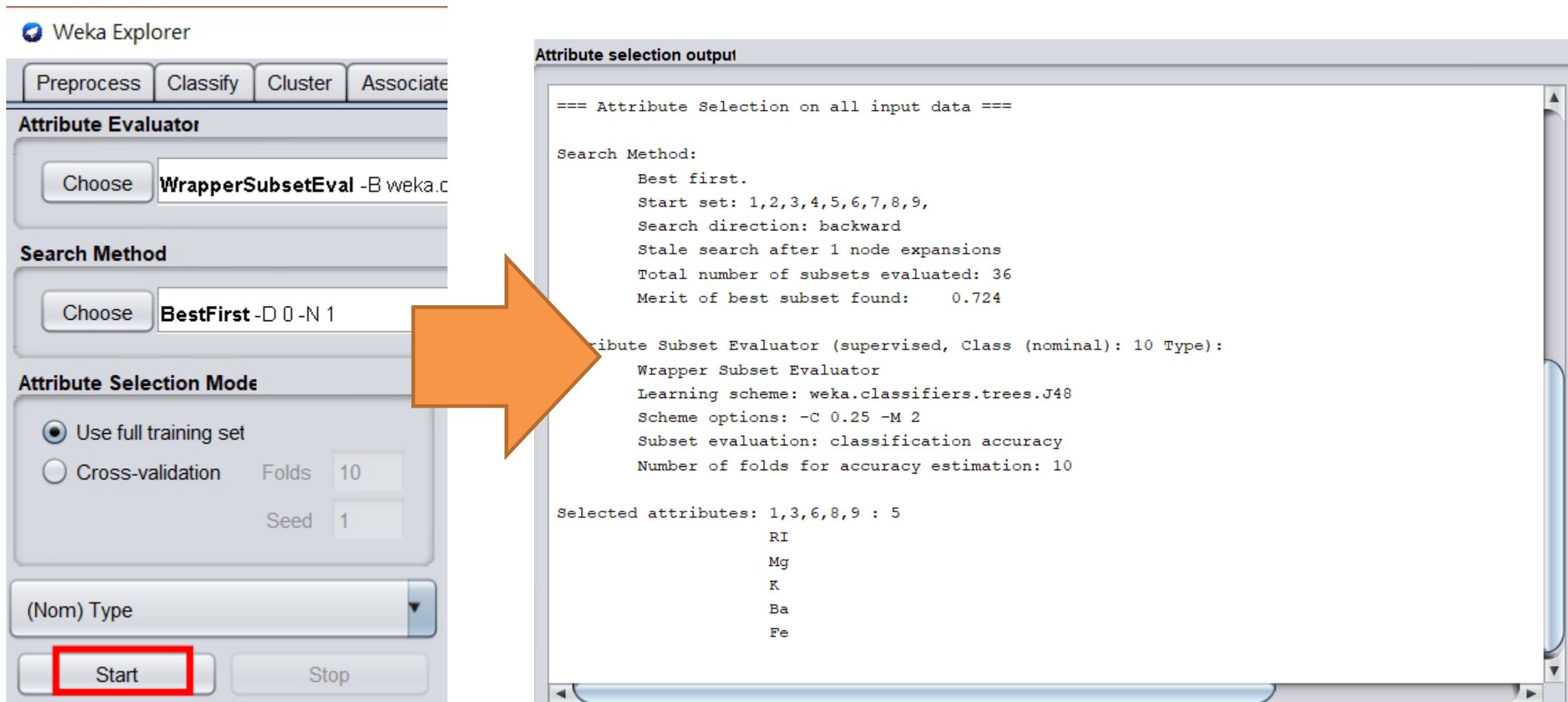
Lesson 4.1: 使用「包裝器」方法選擇屬性

9. 左鍵單擊Search Method中的搜尋方式名稱(左圖紅框處)，開啟配置視窗(右圖)。在配置視窗中將參數searchTermination改為1，然後左鍵單擊下方OK按鈕。



Lesson 4.1: 使用「包裝器」方法選擇屬性

10. 左鍵單擊Start按鈕，執行結果如右圖。

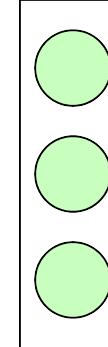
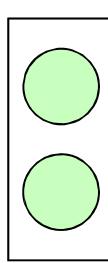
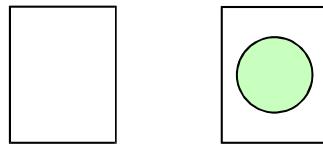


Lesson 4.1: 使用「包裝器」方法選擇屬性

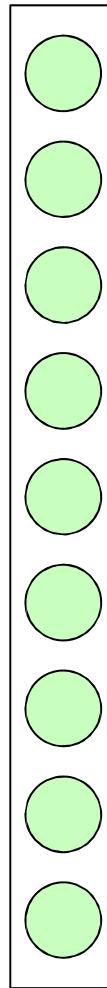
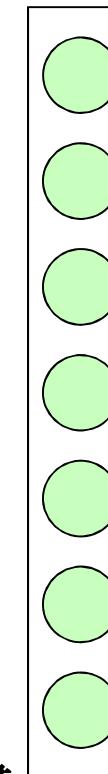
搜尋

- ❖ 窮舉搜尋: $2^9 = 512$ 子集
- ❖ 前向搜尋, 後向搜尋
 - + 何時停止? (參數`searchTermination`(搜尋終止))

0 屬性
(ZeroR)



...

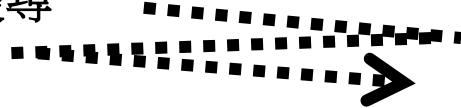


所有的9種屬性

前向搜尋 →

雙向搜尋

← 後向搜尋



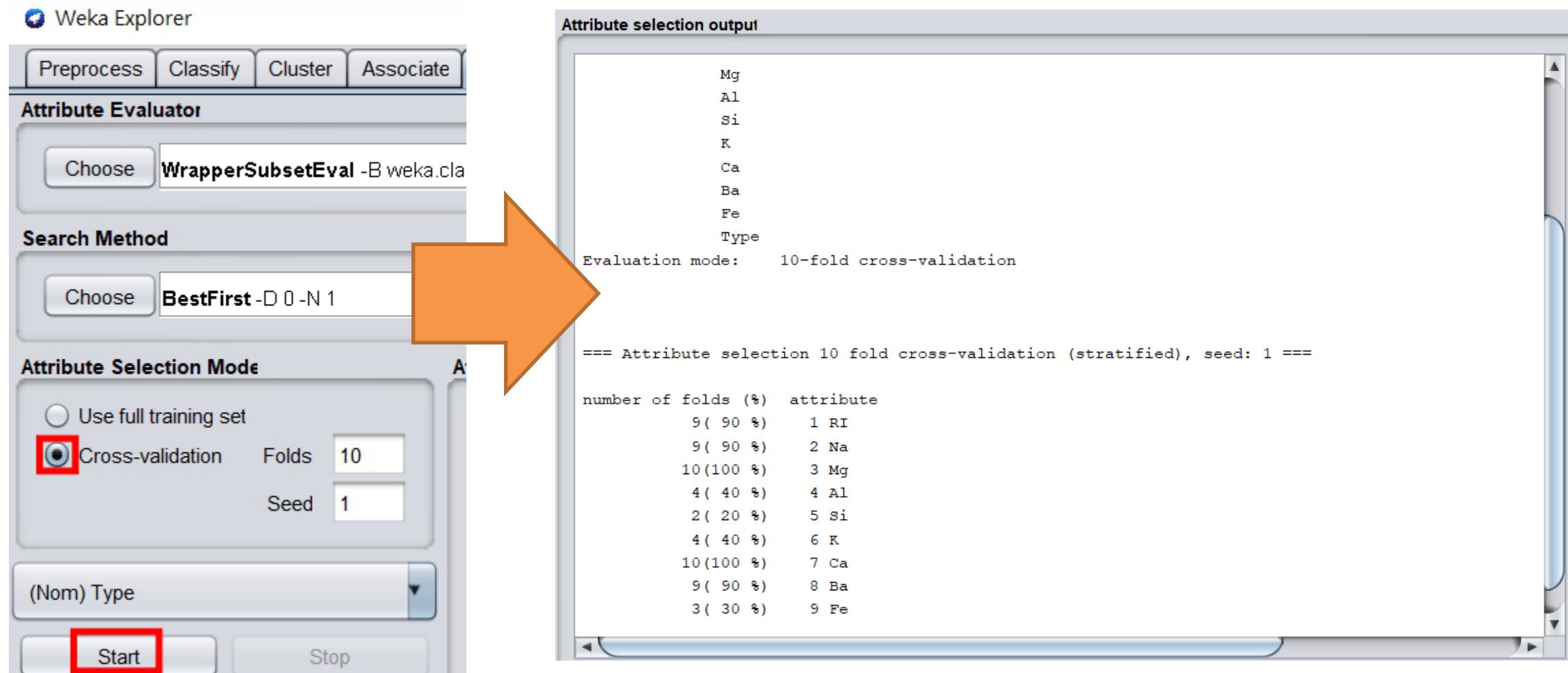
Lesson 4.1: 使用「包裝器」方法選擇屬性

試試不同的搜尋法(*WrapperSubsetEval* 層數(folds) = 10, 闕值(threshold) = -1)

- ❖ 後向 (*searchTermination* = 1): *RI, Mg, K, Ba, Fe* (0.72)
 - *searchTermination* = 5 or more: *RI, Na, Mg, Ca, Ba* (0.74)
- ❖ 前向: *RI, Al, Ca* (0.70)
 - *searchTermination* = 2 or more: *RI, Na, Mg, Al, K, Ca* (0.72)
- ❖ 雙向: *RI, Al, Ca* (0.70)
 - *searchTermination* = 2 or more: *RI, Na, Mg, Al* (0.74)
- ❖ 注意: 局部(local) vs 全體最佳值
 - *searchTermination* > 1可以在搜索範圍內另闢捷徑找到更好的局部最佳值
- ❖ **Al是最合適的屬性(OneR可以向你證明)**
 - 所以前向搜尋的結果包含Al
- ❖ 真詭的是，Al也是最合適放棄的屬性
 - 所以後向搜尋的結果不包含Al

Lesson 4.1: 使用「包裝器」方法選擇屬性

10. 設置交叉驗證看看：左鍵單擊Cross-validation前方圓圈，然後左鍵單擊Start按鈕，執行結果如右圖。Weka做的是10次獨立的屬性評估，並且告訴我們在最終的屬性集中，屬性RI出現的次數(9)。



Lesson 4.1: 使用「包裝器」方法選擇屬性

交叉驗證

後向搜尋 (*searchTermination*=5)

number of folds (%)	attribute
10 (100 %)	1 RI
8 (80 %)	2 Na
10 (100 %)	3 Mg
3 (30 %)	4 Al
2 (20 %)	5 Si
2 (20 %)	6 K
7 (70 %)	7 Ca
10 (100 %)	8 Ba
4 (40 %)	9 Fe

在最終的子集中
屬性RI出現在多
少折？

肯定選擇 RI, Mg, Ba; 可能選擇Na, Ca; 可能部會選擇Al, Si, K, Fe

但如果我們操作前向搜尋，我們一定會選擇AL!

Lesson 4.1: 使用「包裝器」方法選擇屬性

底層細節

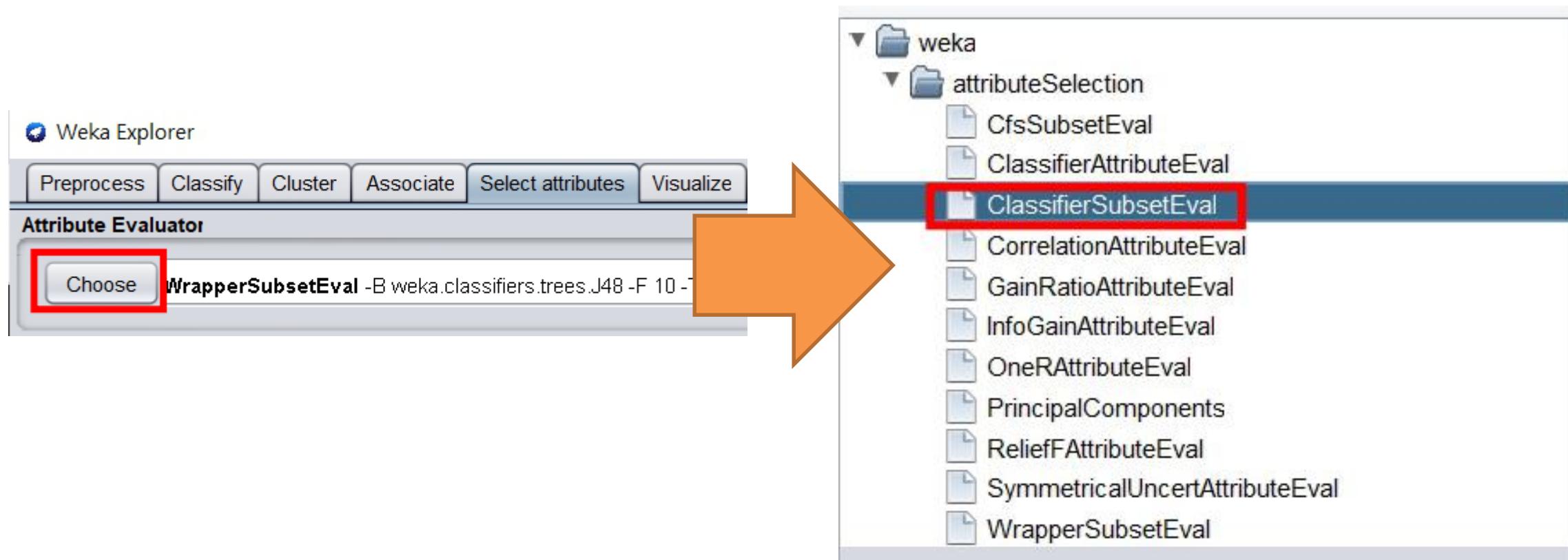
(一般而言, Weka方法遵循研究文獻中的描述)

- ❖ *WrapperSubsetEval* 屬性評估器
 - 預設: 5層交叉驗證
 - 運行至少2-5次交叉驗證然後取平均準確率
 - 當標準差小於Weka使用者設定的臨界值時停止(預設: 平均值的1%)
 - 設置負數臨界值，每次只進行一次交叉驗證
- ❖ *BestFirst* 搜尋法
 - *searchTermination* 預設為5
- ❖ 選擇*ClassifierSubsetEval* 來使用包裝器方法(wrapper method), 但是使用單獨的測試集，而不是交叉驗證

Lesson 4.1: 使用「包裝器」方法選擇屬性

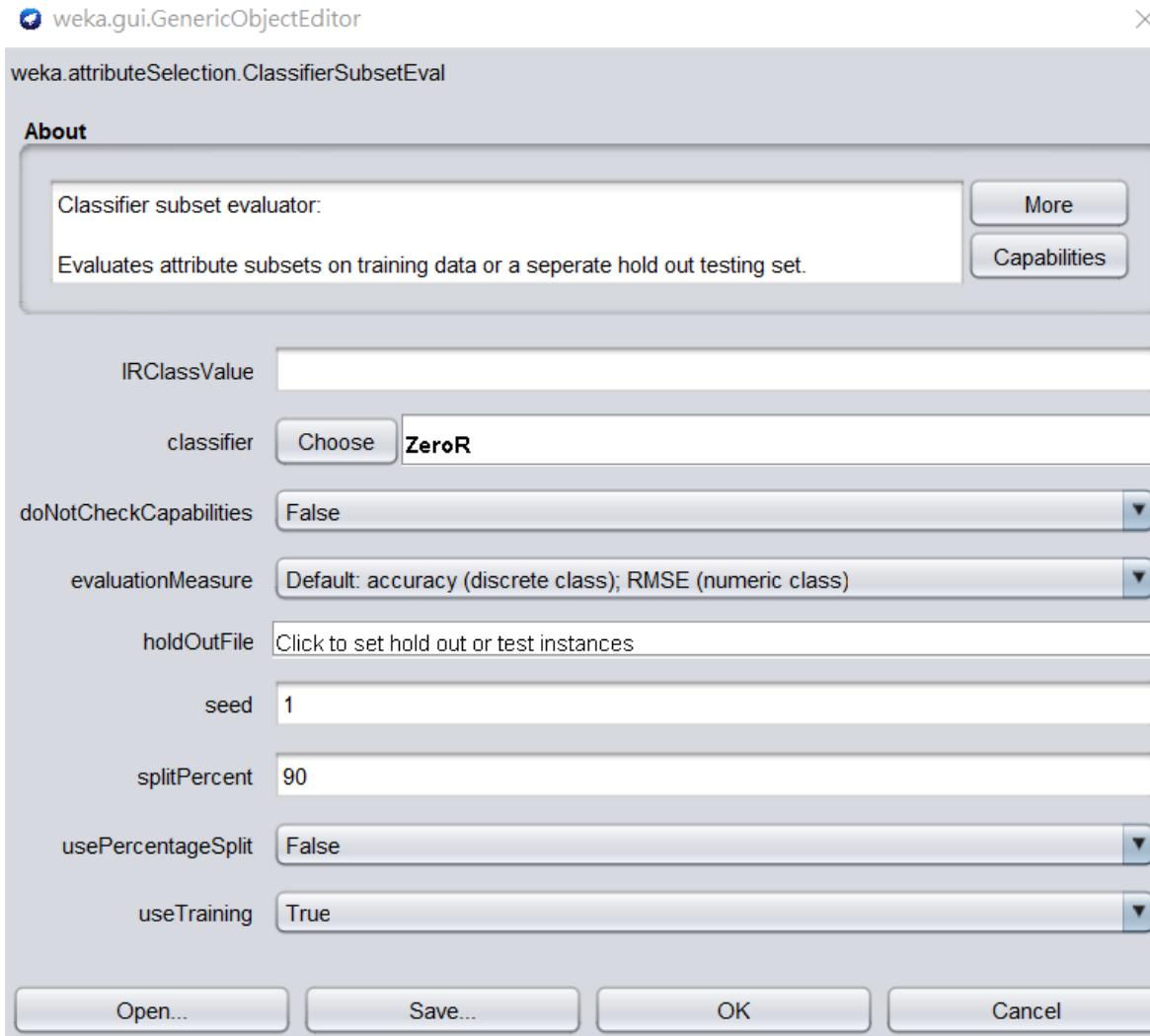
回到Weka，這有另一個屬性評估器，叫做**ClassifierSubsetEvaluator**。

1.回到Select attributes面板，以滑鼠左鍵點擊Choose按鈕，在彈出的選單中選擇attributeSelection資料夾下的**ClassifierSubsetEval**屬性選擇器。



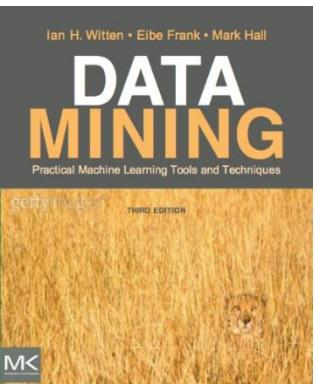
Lesson 4.1: 使用「包裝器」方法選擇屬性

2. ClassifierSubsetEval可以指定一個分類器和一個備用評估文檔（HoldOutFile），我們用HoldOutFile來依次評估子集。



Lesson 4.1: 使用「包裝器」方法選擇屬性

- ❖ 使用分類器找到好的屬性集(方案獨立(scheme-dependent))
 - 我們使用J48
- ❖ 使用交叉驗證循環包裝分類器
- ❖ 包含了屬性評估器(Attribute Evaluator)和搜尋方法
- ❖ 搜尋可以開始於任何子集，前向、後向、或雙向
 - 包裝法的計算量很大;對於 m 個屬性,需要評估 m^2 的平方個子集
 - 也可以使用窮盡方法(exhaustive search method)評估 2^m 個子集
- ❖ 貪婪搜尋總能在搜尋的區域找到局部最佳值
 - 你可以增加搜尋終止參數再多次嘗試



課程文本

- ❖ Section 7.1 *Attribute selection*



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Class 4 – Lesson 2

屬性選擇分類器

The Attribute Selected Classifier

Ian H. Witten

Department of Computer Science University of Waikato
New Zealand

Lesson 4.2: 屬性選擇分類器

Class 1 探索Weka的介面；處理大數據

Class 2 離散以及文本分類

Class 3 分類規則、關聯規則、聚類

Class 4 選擇屬性以及計算成本

Class 5 神經網路，學習曲線和表現優化

Lesson 4.1 「包裝器」屬性選擇法

Lesson 4.2 屬性選擇分類器

Lesson 4.3 Scheme-independent selection

Lesson 4.4 Attribute selection using ranking

Lesson 4.5 Counting the cost

Lesson 4.6 Cost-sensitive classification

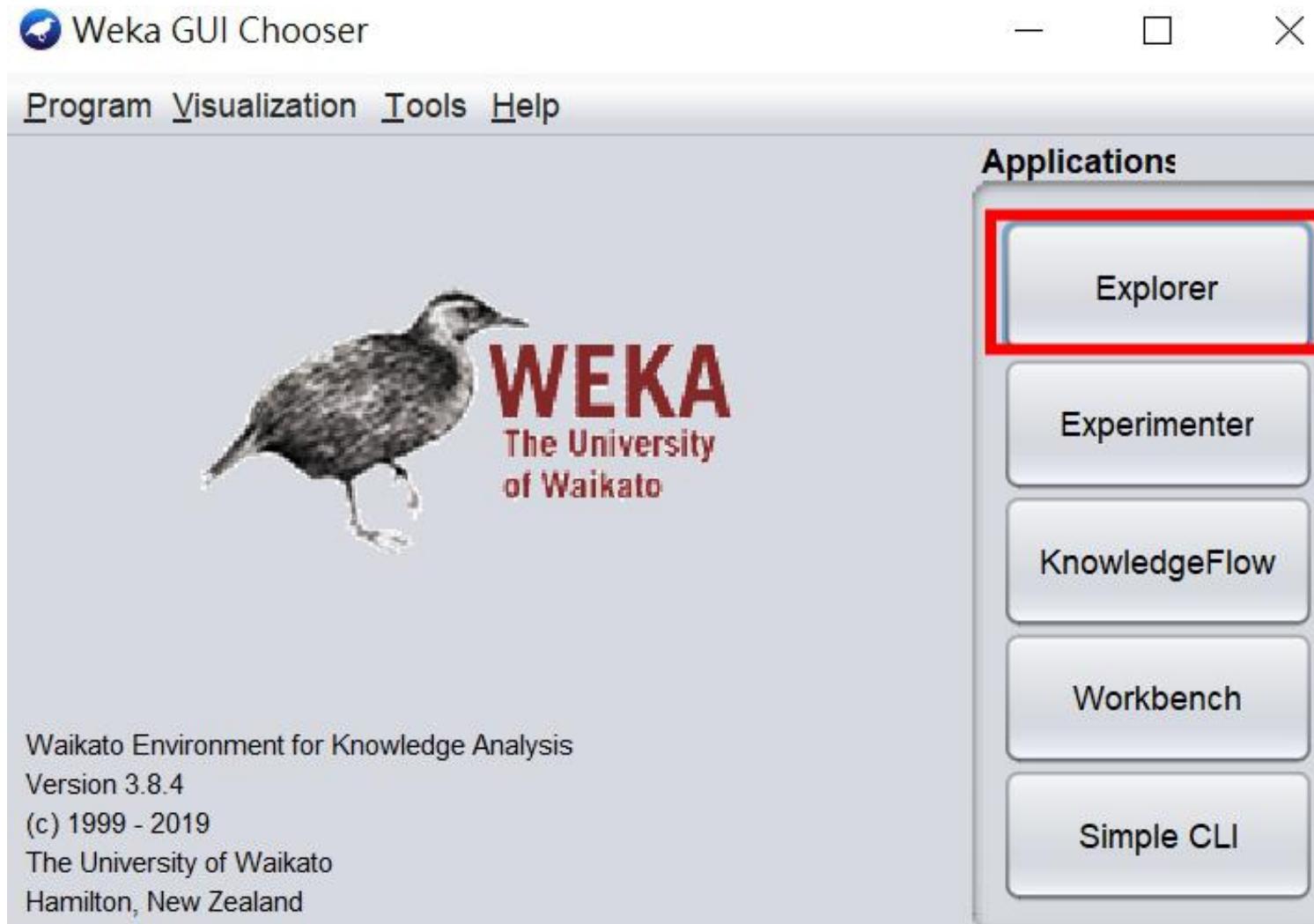


Lesson 4.2: 屬性選擇分類器

- ❖ 選擇屬性並且應用分類器到結果上
 - *glass.arff* 使用預設參數 J48 67%
 - 使用J48的包裝器方法{RI, Mg, Al, K, Ba} IBk 71%
 - 使用Bk{RI, Mg, Al, K, Ca, Ba} 78%
 - ❖ 這是作弊嗎? – yes!
 - ❖ **AttributeSelectedClassifier** (在meta內)
 - 僅根據訓練數據選擇屬性
 - ... 然後對分類器進行訓練並對測試數據進行評估
 - 就像用於監督離散化的*FilteredClassifier* (Lesson 2.2)
 - 使用AttributeSelectedClassifier 來包裝 J48
 - 使用AttributeSelectedClassifier 來包裝 IBk
- 72% 74%
69% 71%
- (略驚訝的結果)

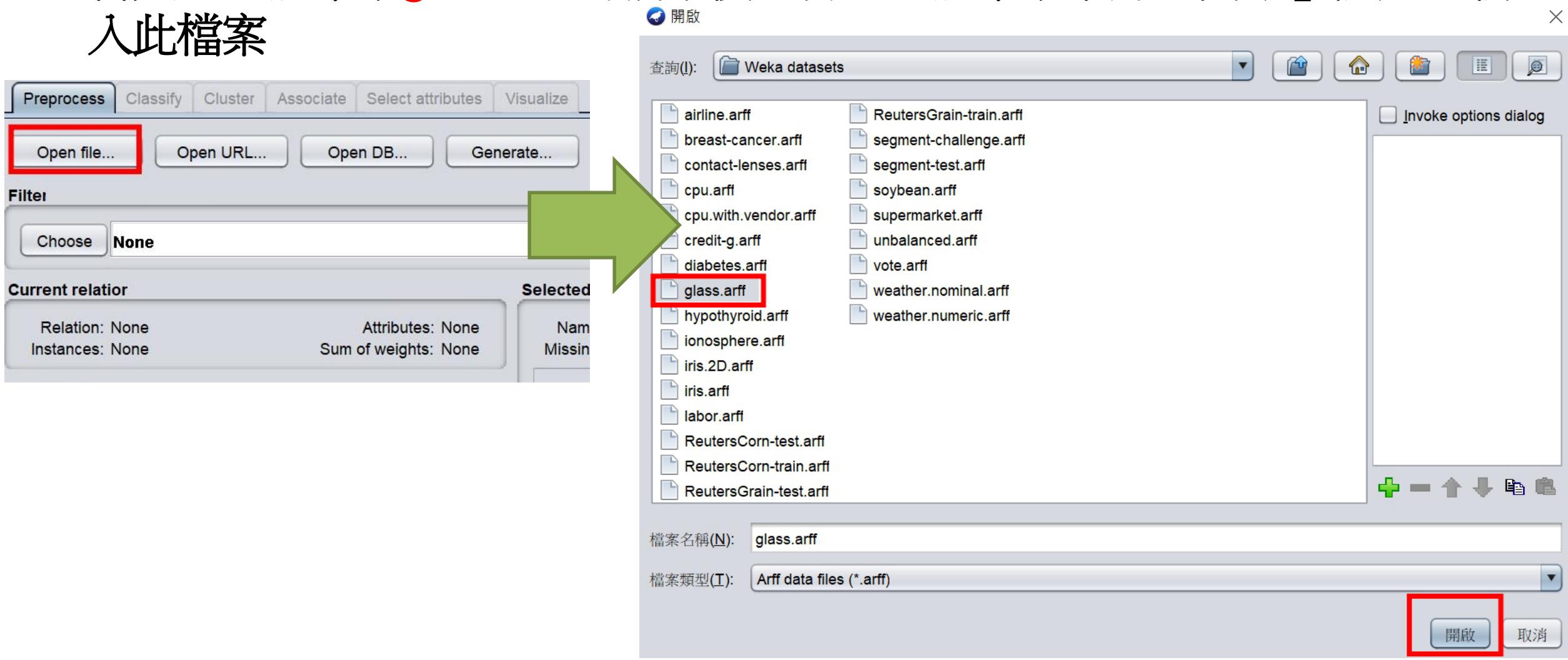
Lesson 4.2: 屬性選擇分類器

1. 開啟Weka的Explorer



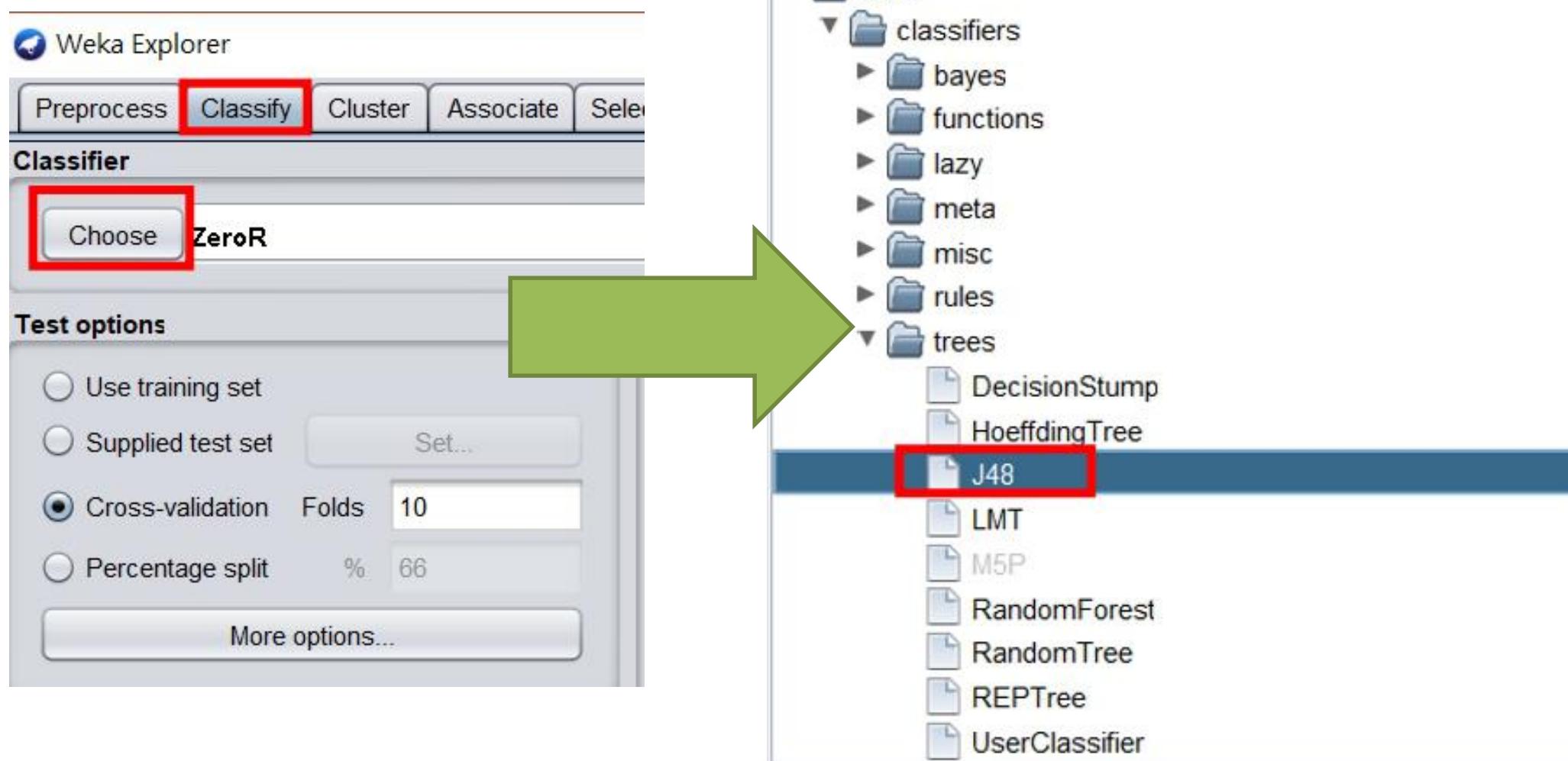
Lesson 4.2: 屬性選擇分類器

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊glass.arff的檔案後，再以左鍵單擊下方「開啟」按鈕以載入此檔案



Lesson 4.2: 屬性選擇分類器

3. 切換到Classify界面點選Choose鈕，在出現的選單中左鍵單擊trees資料夾下的J48



Lesson 4.2: 屬性選擇分類器

4. 左鍵單擊Start按鈕，執行結果如右圖，得到66.8224%準確率。

The screenshot shows the Weka interface for a classification task. On the left, the 'Test options' panel is visible, featuring a 'Choose' button set to 'J48 -C 0.25 -M 2'. Below it, the 'Cross-validation' option is selected with 'Folds' set to 10. A large orange arrow points from the 'Start' button in the 'Test options' panel to the 'Classifier output' window on the right.

Classifier output

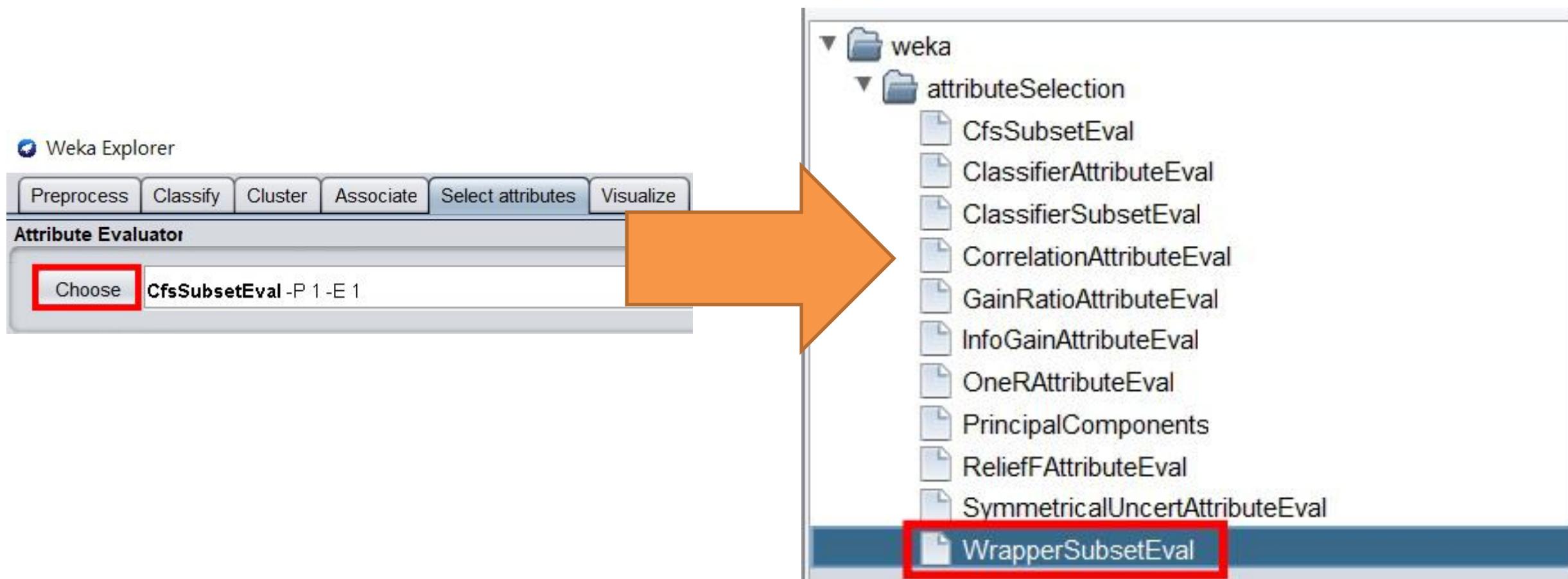
==== Summary ====
Correctly Classified Instances 143 66.8224 %
Incorrectly Classified Instances 71 33.1776 %
Kappa statistic 0.55
Mean absolute error 0.1026
Root mean squared error 0.2897
Relative absolute error 48.4507 %
Root relative squared error 89.2727 %
Total Number of Instances 214

==== Detailed Accuracy By Class ====
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Cl
0.714 0.174 0.667 0.714 0.690 0.532 0.806 0.667 bu
0.618 0.181 0.653 0.618 0.635 0.443 0.768 0.606 bu
0.353 0.046 0.400 0.353 0.375 0.325 0.766 0.251 ve
? 0.000 ? ? ? ? ? ? ve
0.769 0.010 0.833 0.769 0.800 0.788 0.872 0.575 co
0.778 0.029 0.538 0.778 0.636 0.629 0.930 0.527 ta
0.793 0.022 0.852 0.793 0.821 0.795 0.869 0.738 he
Weighted Avg. 0.668 0.130 0.670 0.668 0.668 0.539 0.807 0.611

==== Confusion Matrix ====
a b c d e f g <-- classified as
50 15 3 0 0 1 1 | a = build wind float
16 47 6 0 2 3 2 | b = build wind non-float
5 5 6 0 0 1 0 | c = vehic wind float

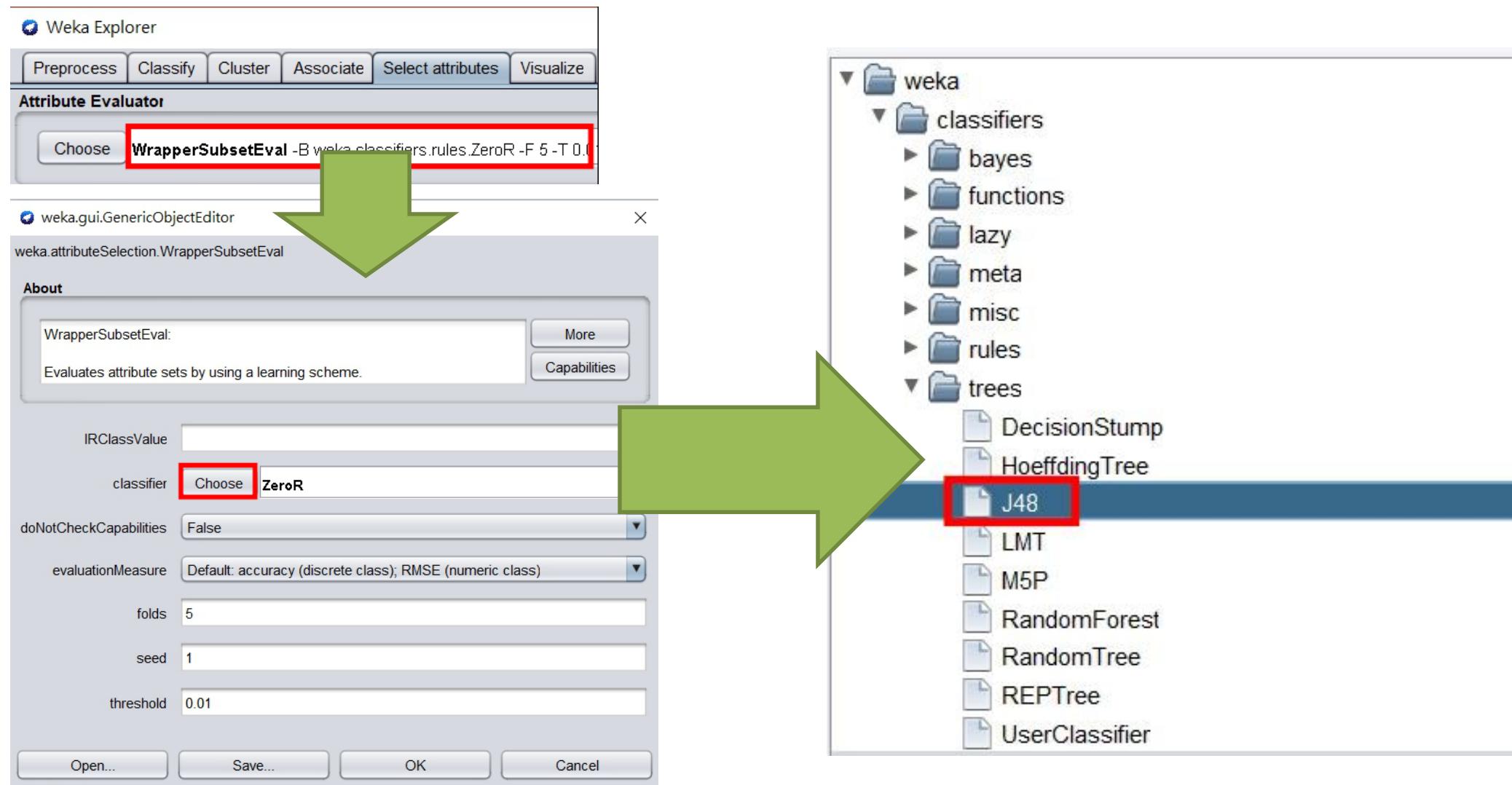
Lesson 4.2: 屬性選擇分類器

5. 切換到Select attributes面板，以滑鼠左鍵點擊Choose，在彈出的選單中選擇attributeSelection資料夾下的WrapperSubsetEval屬性選擇器



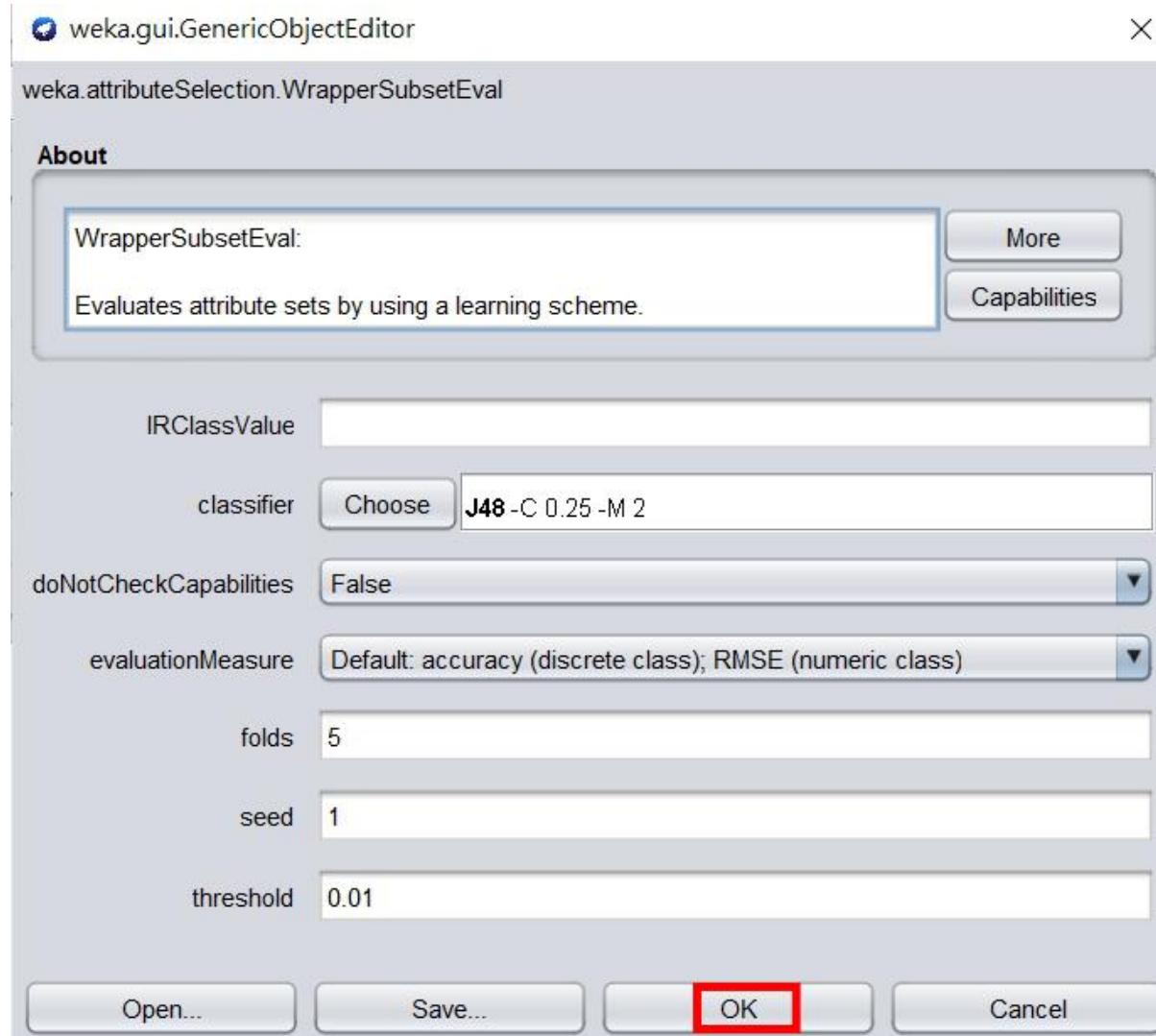
Lesson 4.2: 屬性選擇分類器

6. 左鍵單擊屬性選擇器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊Choose按鈕，再以左鍵單擊J48分類器。



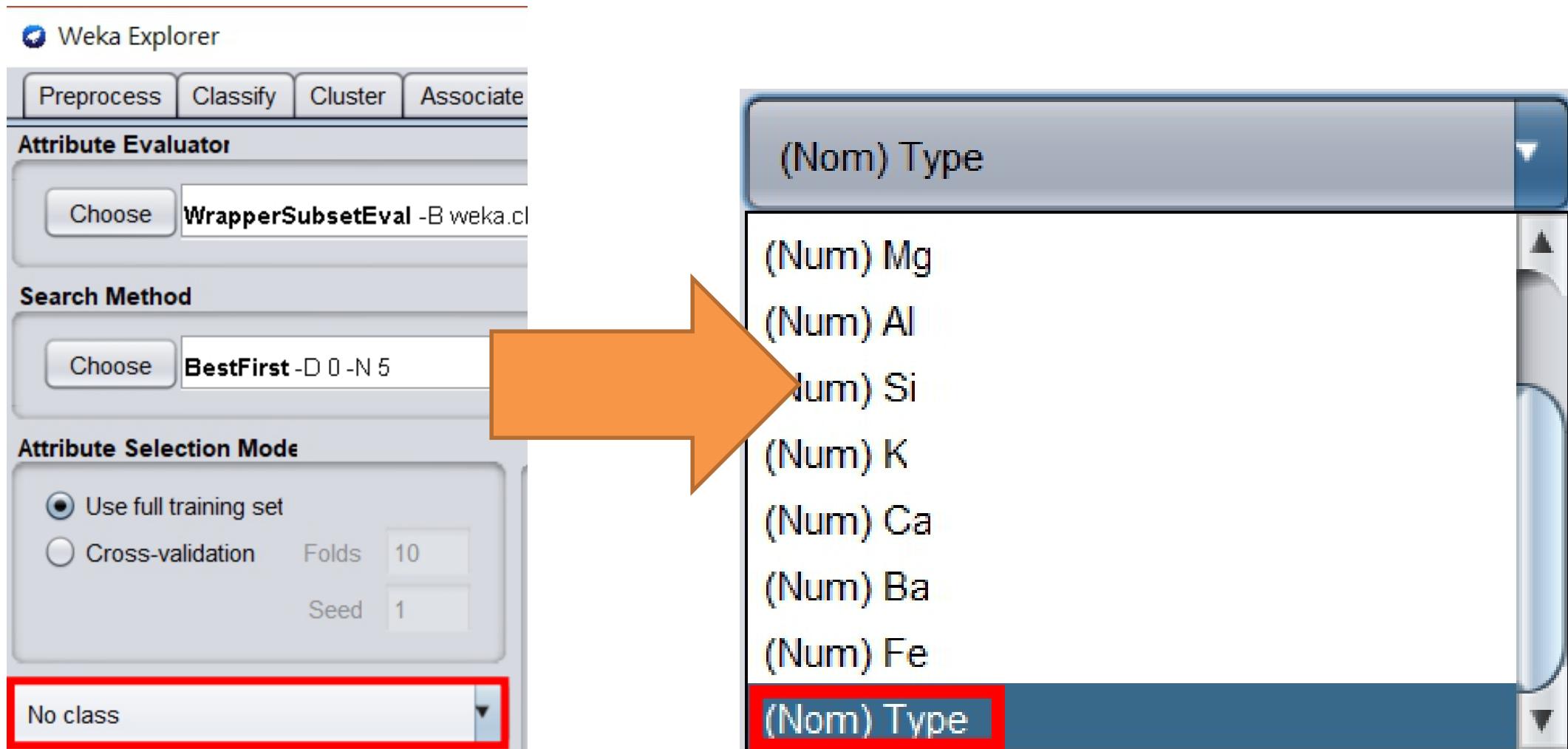
Lesson 4.2: 屬性選擇分類器

7. 左鍵單擊OK按鈕。



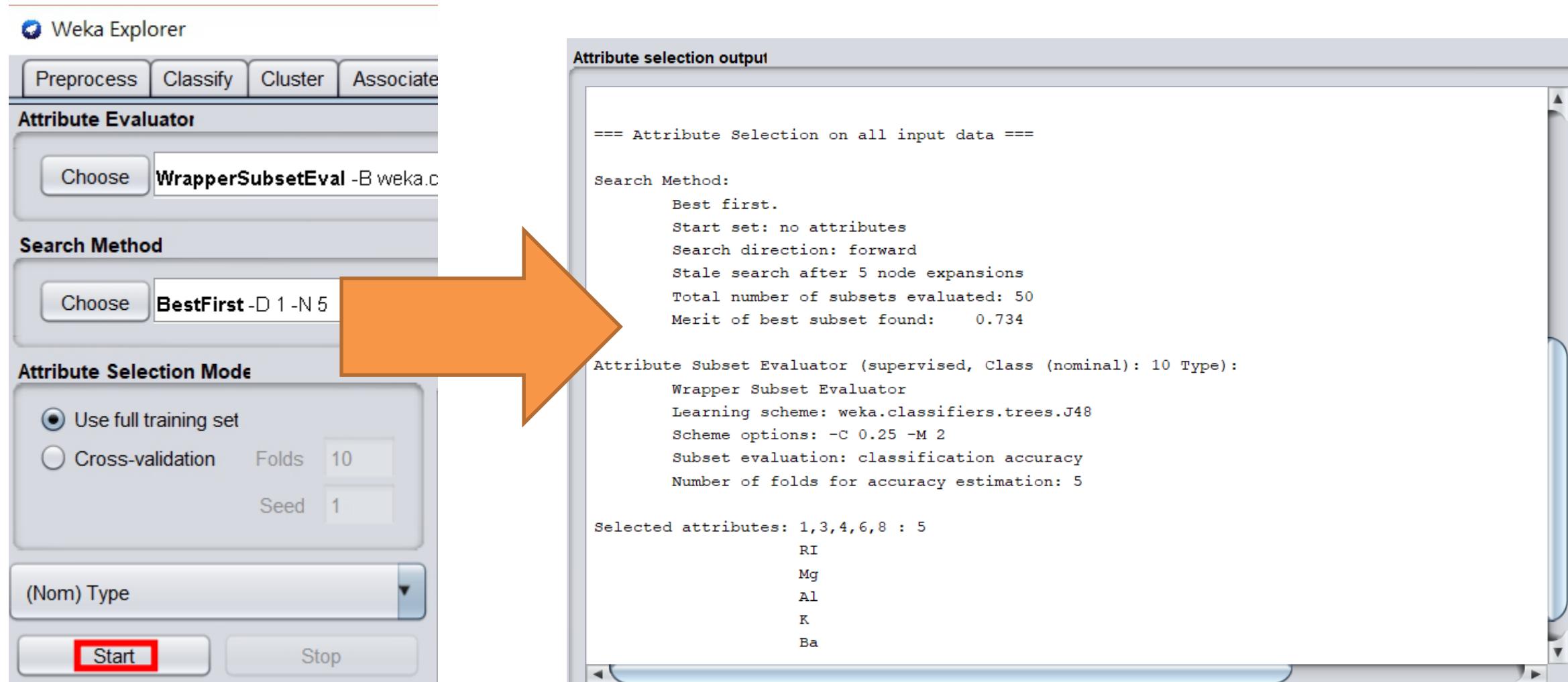
Lesson 4.2: 屬性選擇分類器

8. 回到Select attributes面板，左鍵單擊分類屬性的下拉式選單，並以左鍵單擊(Nom)Type選項。



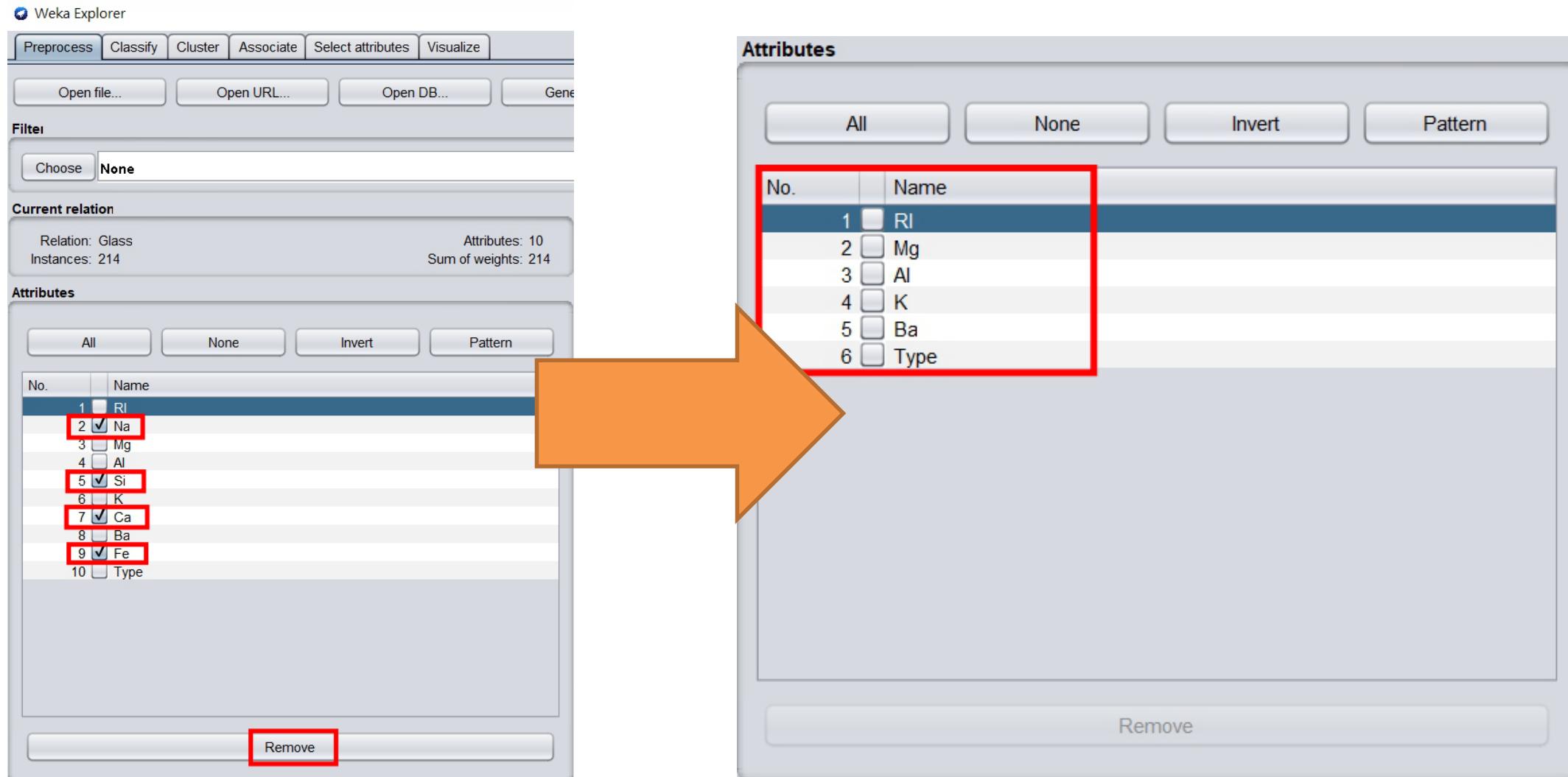
Lesson 4.2: 屬性選擇分類器

9. 左鍵單擊Start按鈕，執行結果如右圖。



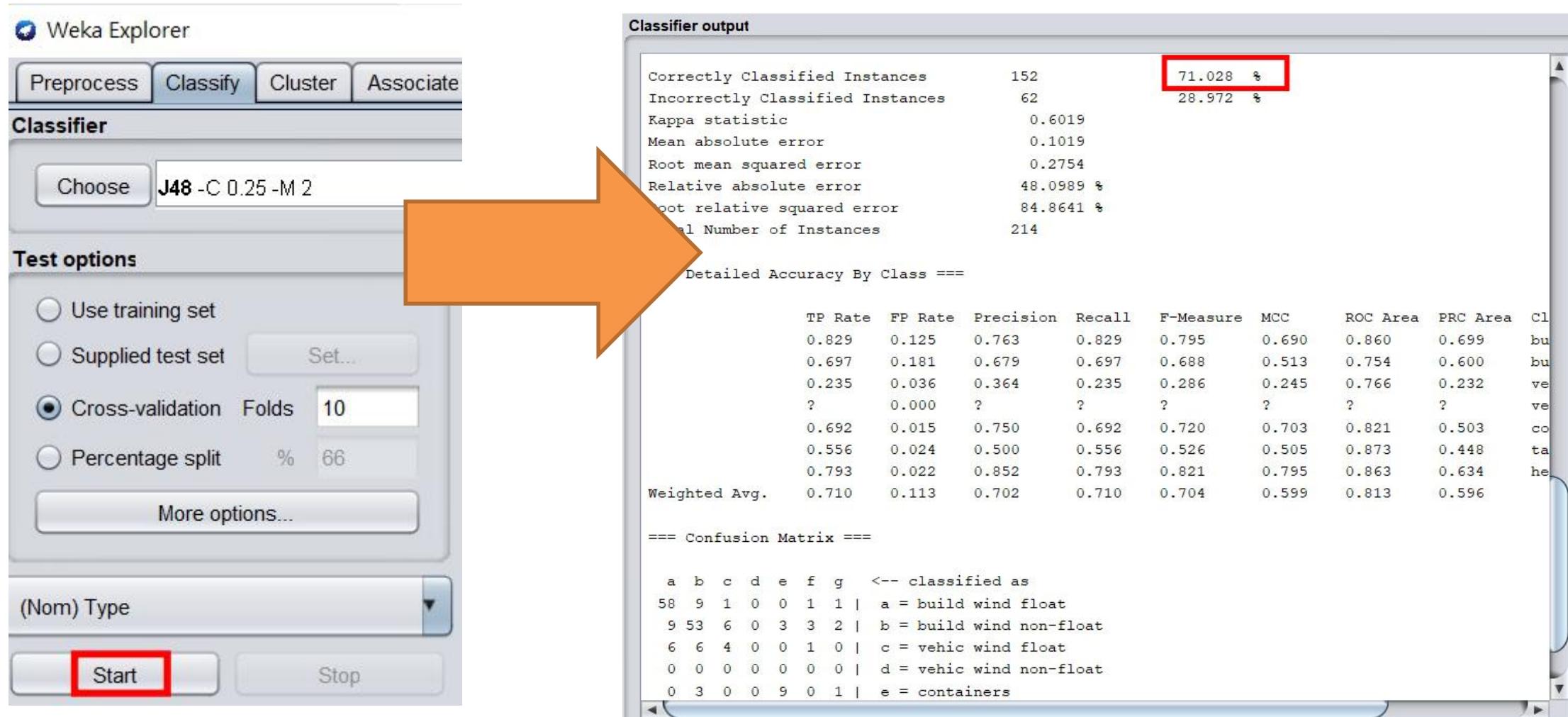
Lesson 4.2: 屬性選擇分類器

10. 切換到Preprocess面板，左鍵單擊Na、Si、Ca、Fe前方勾選方塊，接著按下下方Remove按鈕，最後剩下的屬性應如右圖所示。



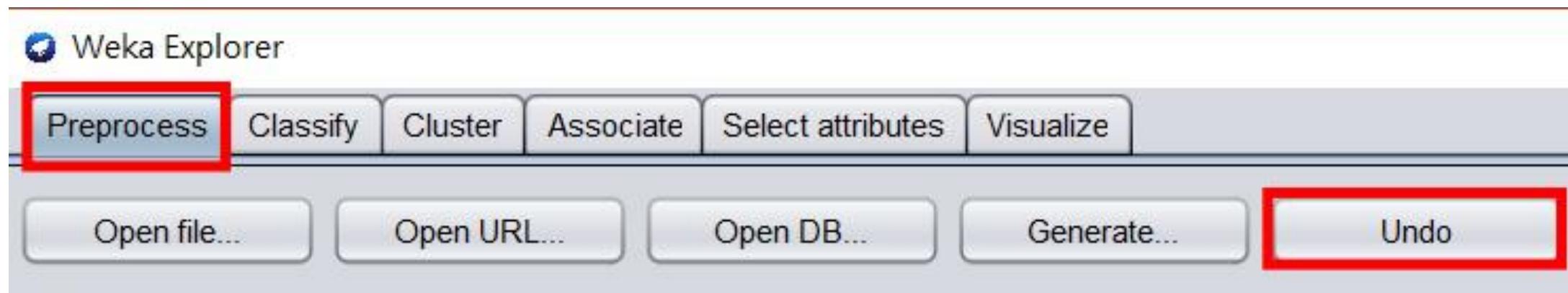
Lesson 4.2: 屬性選擇分類器

11.切換到Classify面板，左鍵單擊Start按鈕，執行結果如右圖，得到71.028%準確率。



Lesson 4.2: 屬性選擇分類器

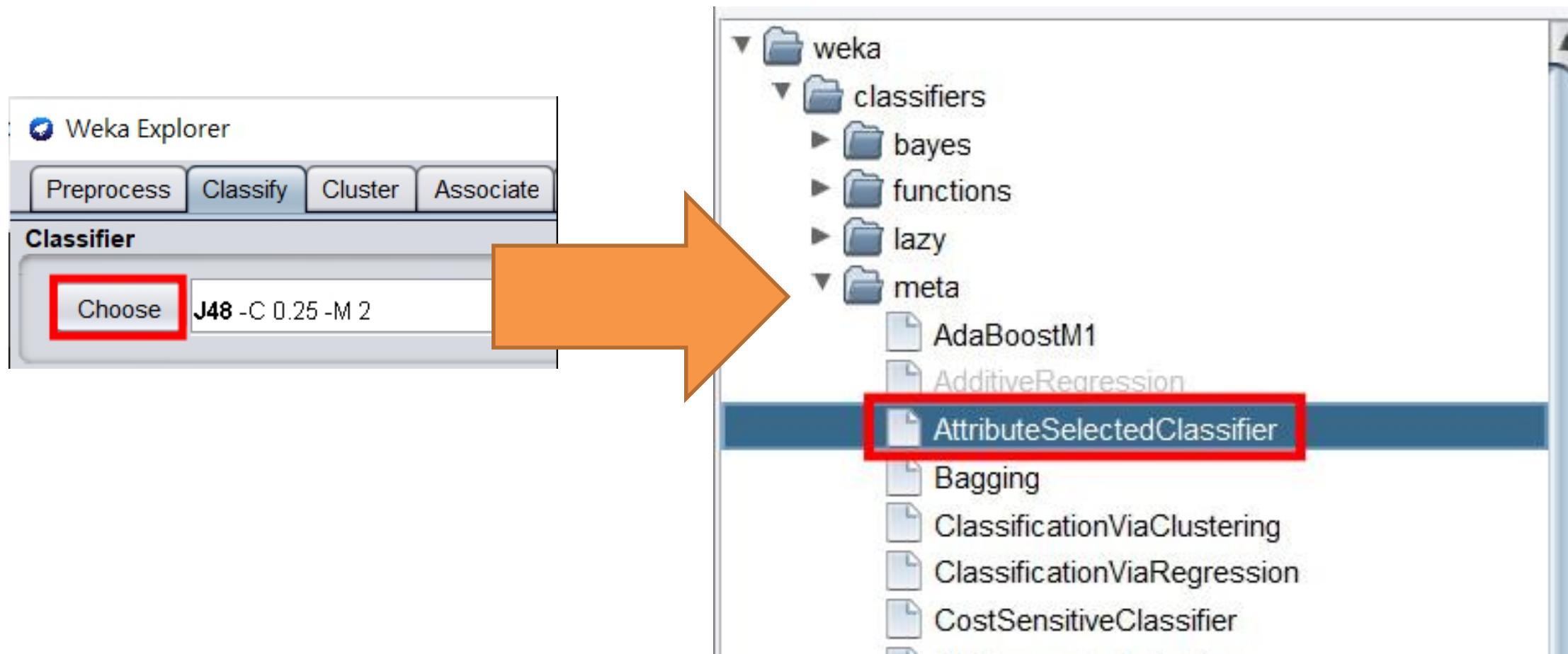
12.切換到Preprocess面板，左鍵單擊Undo按鈕還原剛才刪除的屬性。



Lesson 4.2: 屬性選擇分類器

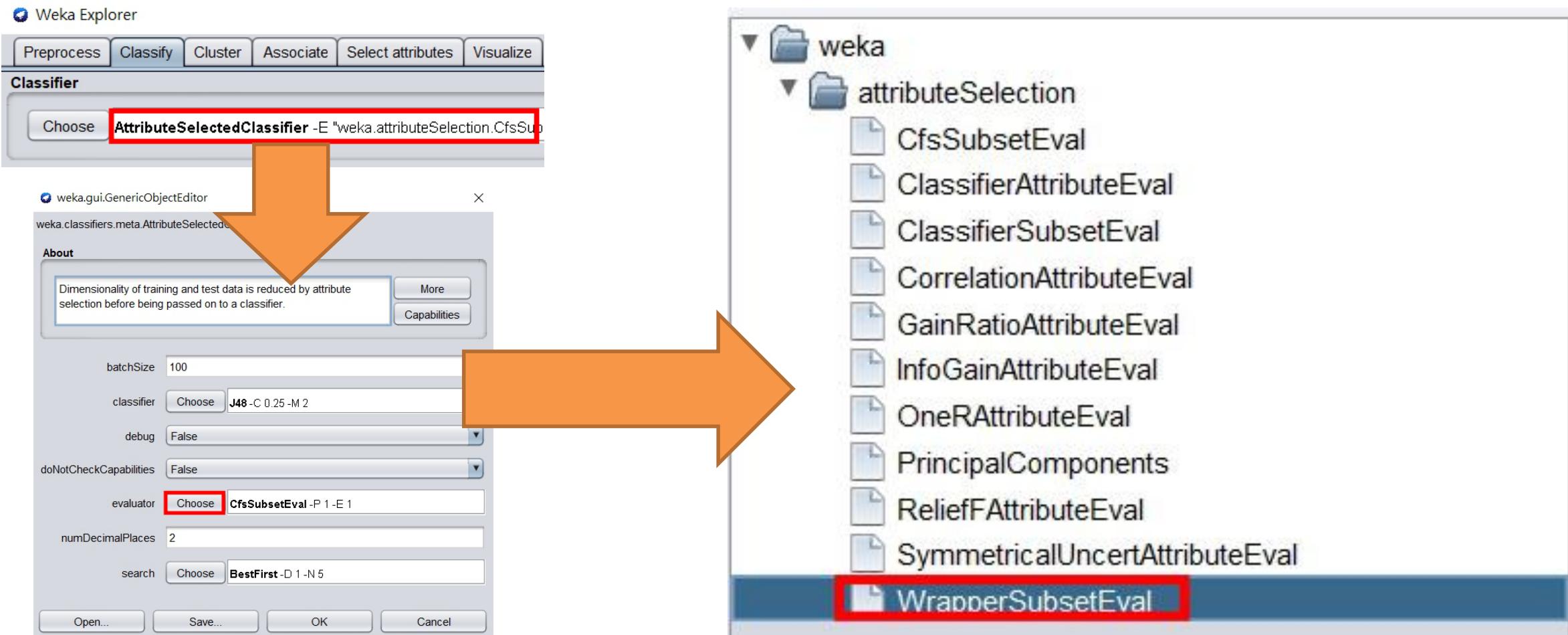
接著使用AttributeSelectedClassifier包裝J48。

1.切換到Classify面板，左鍵單擊Choose按鈕，在出現的選單中以左鍵單擊AttributeSelectedClassifier分類器。



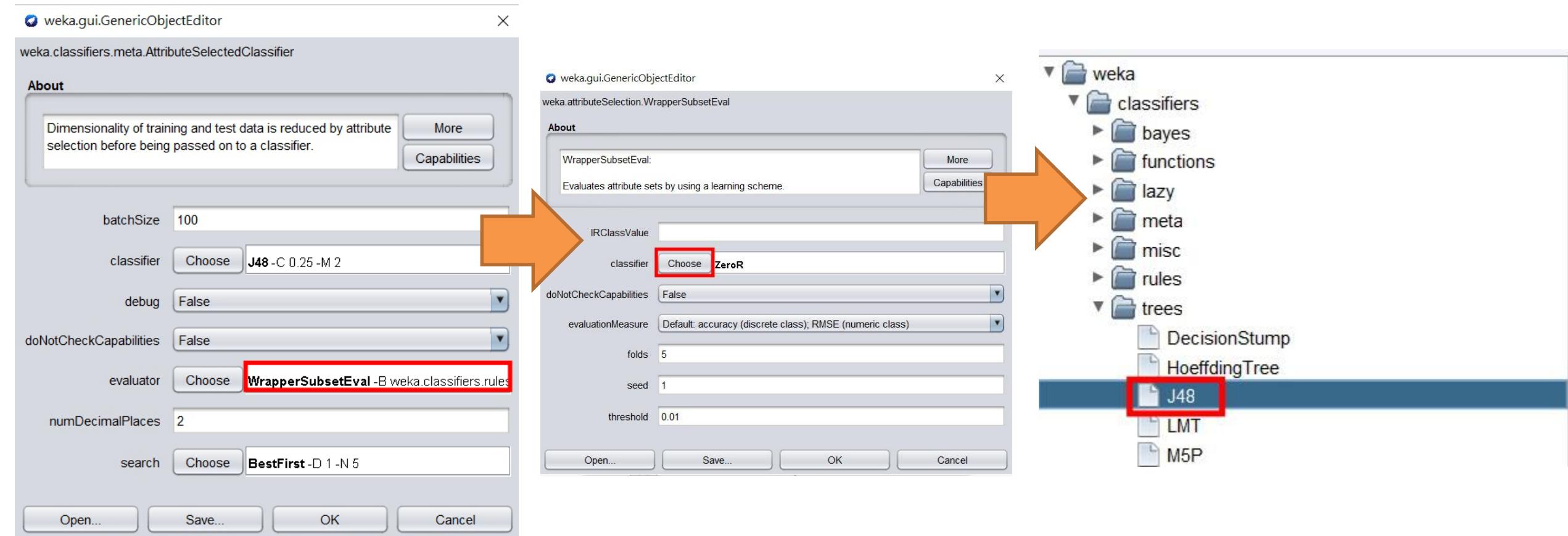
Lesson 4.2: 屬性選擇分類器

2. 左鍵單擊分類器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊Choose按鈕，並在出現的選單中左鍵單擊WrapperSubsetEval。



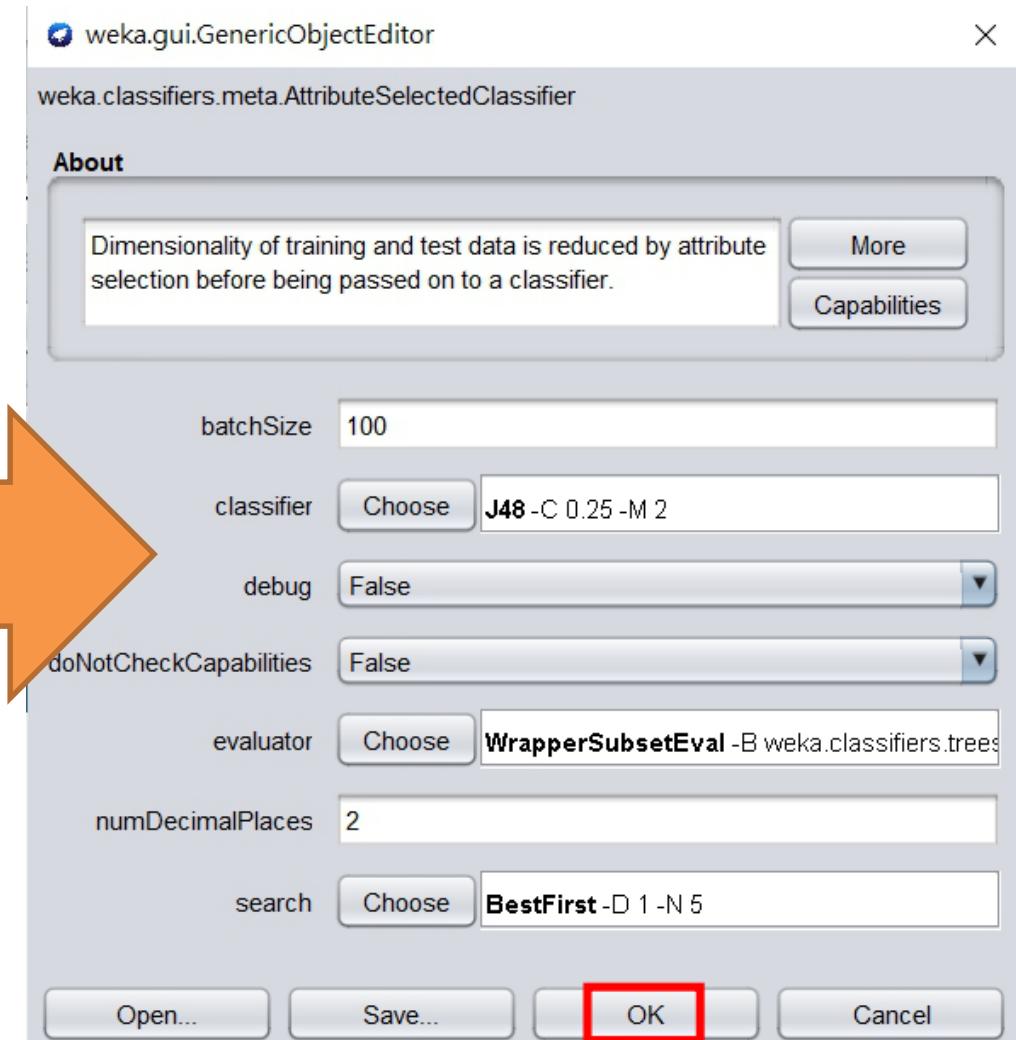
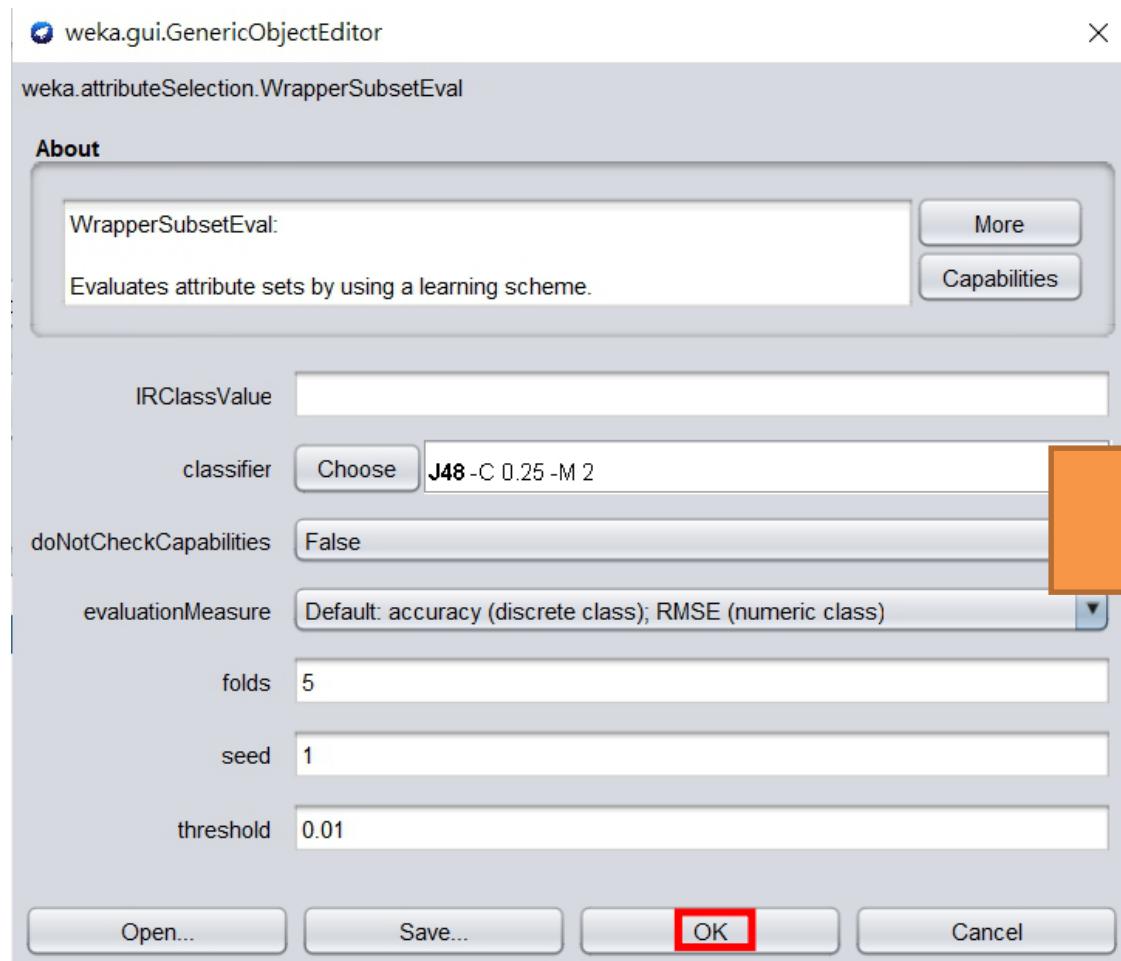
Lesson 4.2: 屬性選擇分類器

3. 左鍵單擊屬性選擇器名稱(最左圖紅框處)，開啟配置視窗(中間圖)。在配置視窗中左鍵單擊Choose按鈕，並在出現的選單中以左鍵單擊J48分類器。



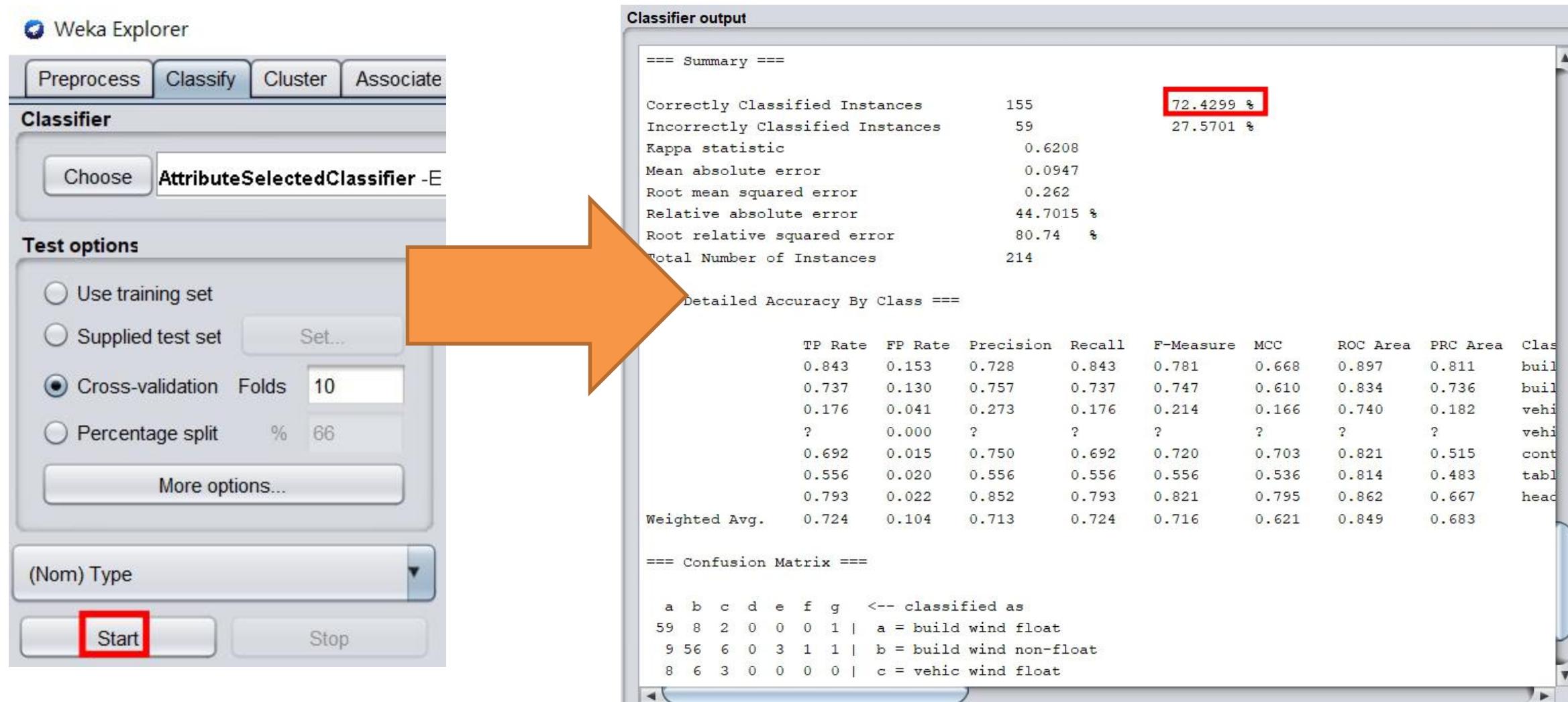
Lesson 4.2: 屬性選擇分類器

4. 左鍵單擊下方OK按鈕回到分類器配置視窗，再以左鍵單擊下方OK按鈕回到Classify面板。



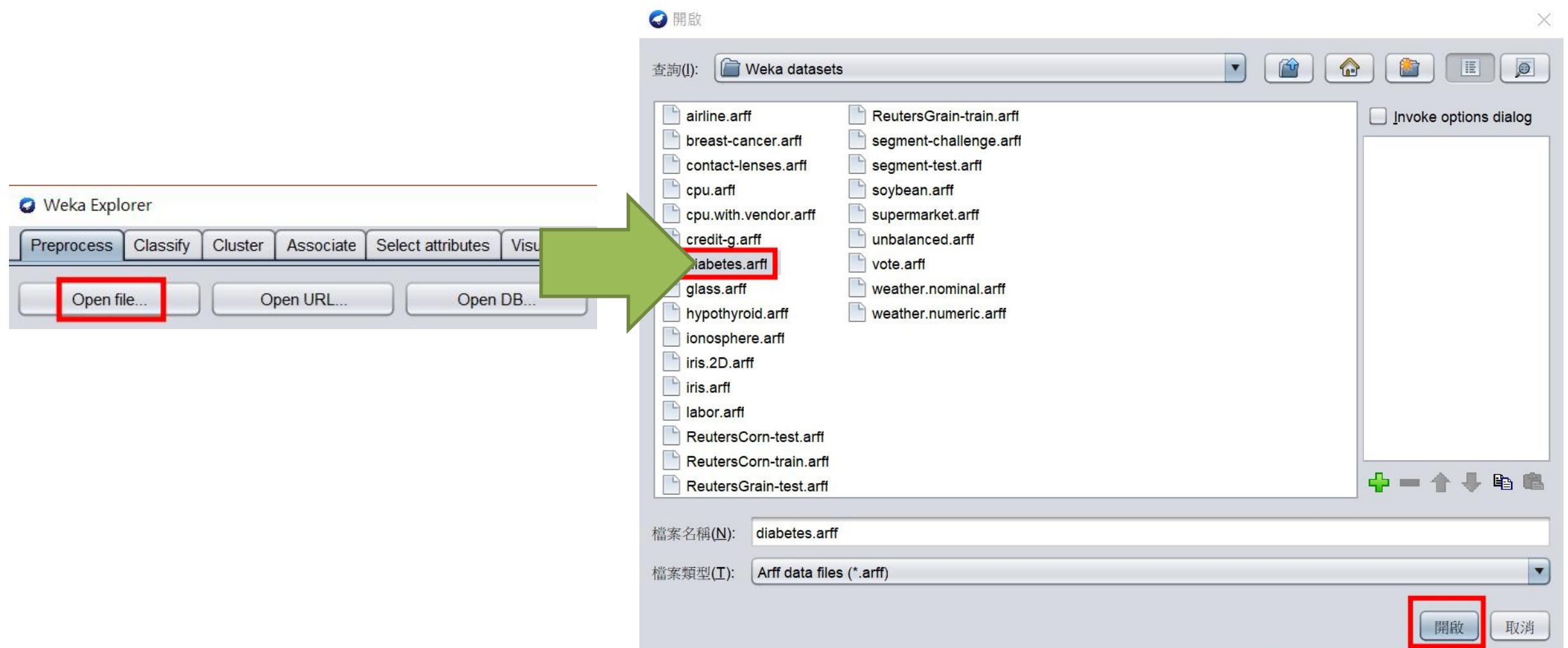
Lesson 4.2: 屬性選擇分類器

5. 左鍵單擊Start按鈕，執行結果如右圖，得到72.4299%準確率。



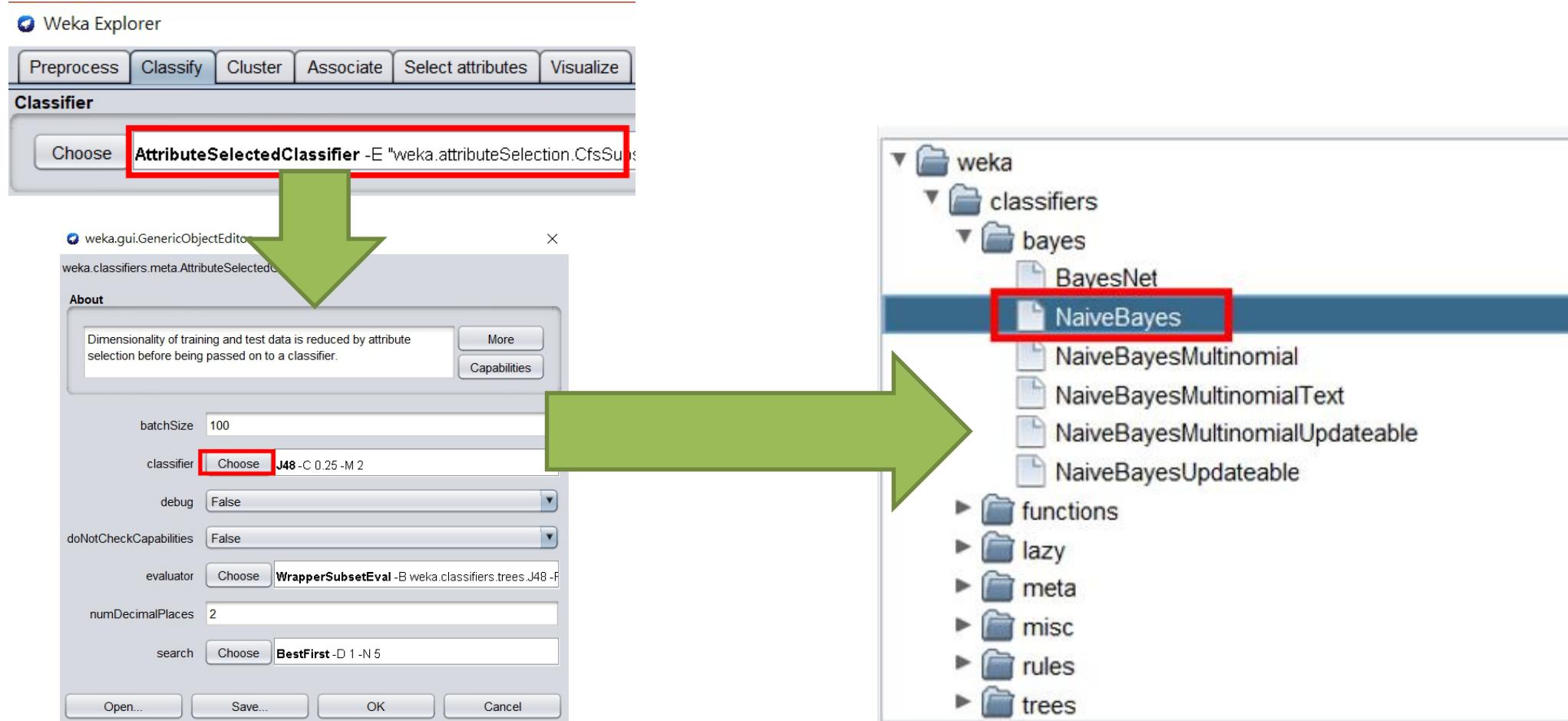
Lesson 4.2: 屬性選擇分類器

1. 切換到Preprocess面板，左鍵單擊Open file按鈕，在彈出的視窗以左鍵單擊diabetes.arff檔案，並按下下方開啟按鈕載入此檔案。



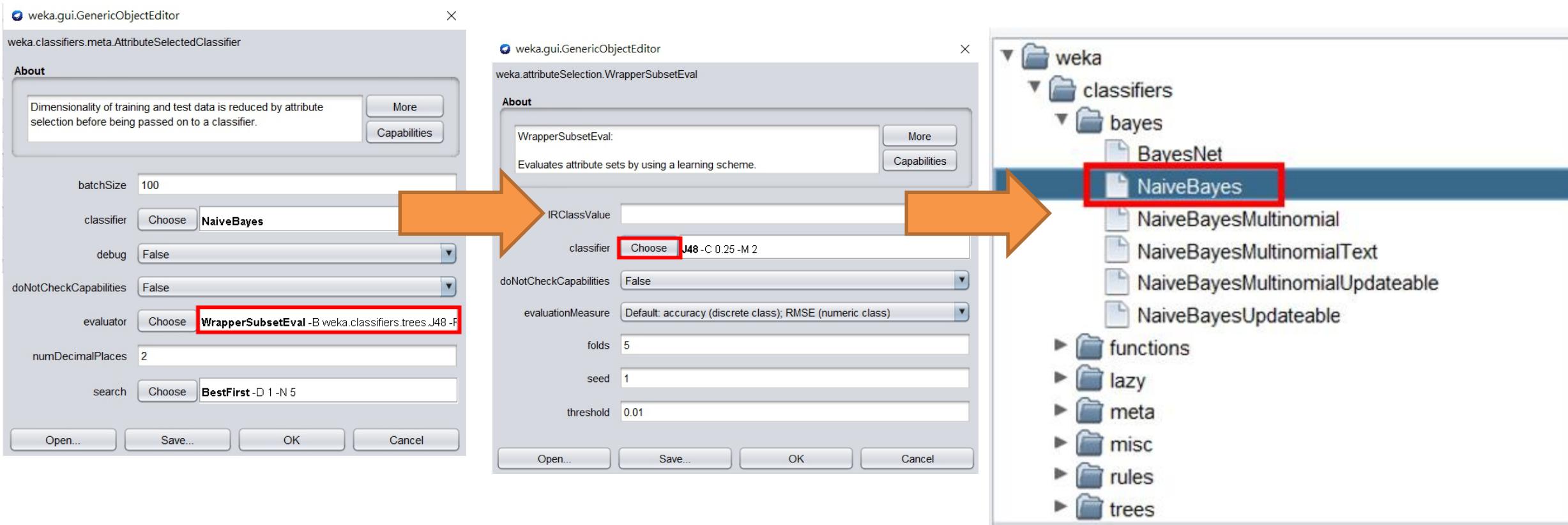
Lesson 4.2: 屬性選擇分類器

2. 切換到Classify面板，左鍵單擊分類器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊Choose按鈕，並在彈出的選單左鍵單擊NaiveBayes分類器。



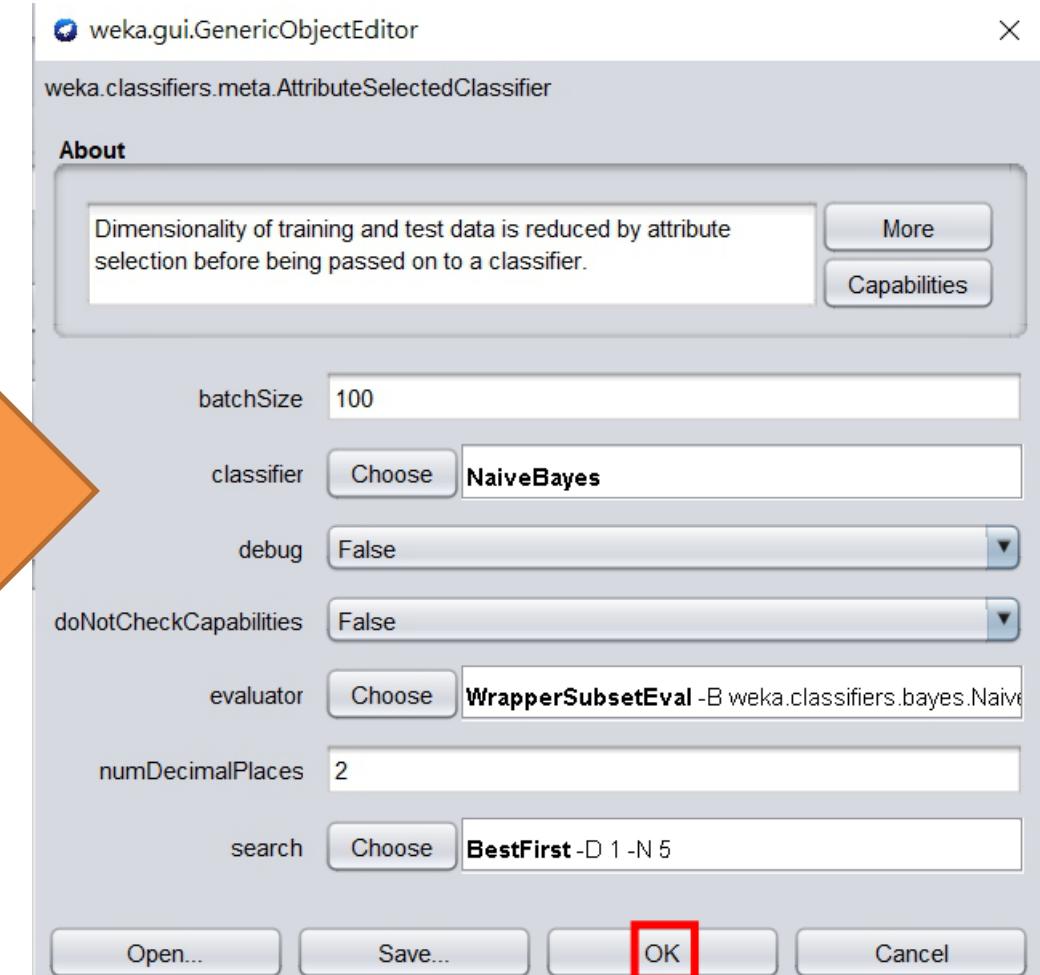
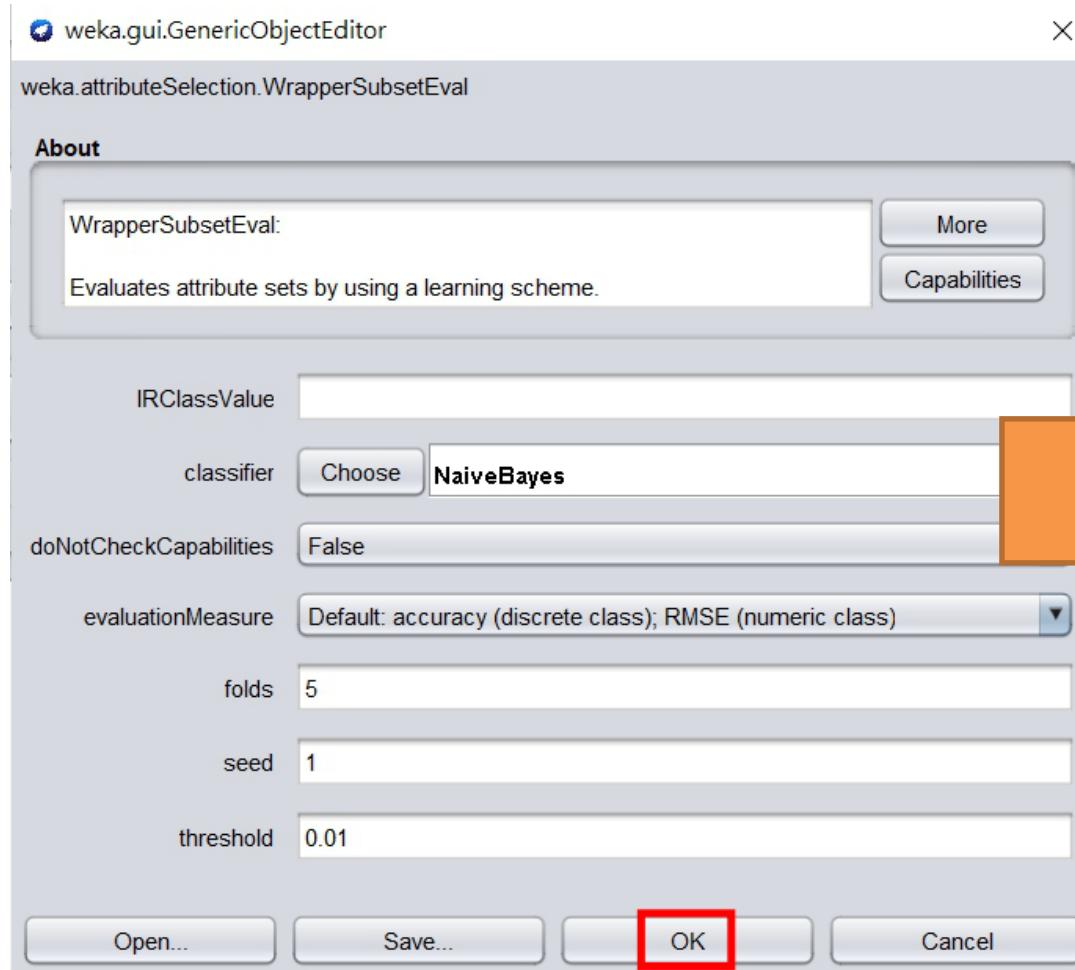
Lesson 4.2: 屬性選擇分類器

3. 左鍵單擊屬性選擇器名稱(最左圖紅框處)，開啟配置視窗(中間圖)。在配置視窗中左鍵單擊Choose按鈕，並在彈出的選單中左鍵單擊NaiveBayes分類器。



Lesson 4.2: 屬性選擇分類器

4. 左鍵單擊下方OK按鈕回到分類器配置視窗，再以左鍵單擊下方OK按鈕回到Classify面板。



Lesson 4.2: 屬性選擇分類器

5. 左鍵單擊Start按鈕，執行結果如右圖，得到75.651%準確率。

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'AttributeSelectedClassifier -E "weka' is chosen. The 'Test options' section has 'Cross-validation' selected with 'Folds' set to 10. A large orange arrow points from the 'Start' button in the bottom left of the explorer window to the 'Classifier output' window on the right.

Classifier output

```
Time taken to build model: 2.81 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      581
Incorrectly Classified Instances   187
Kappa statistic                   0.4373
Mean absolute error               0.3053
Root mean squared error           0.4056
Relative absolute error           67.1652 %
Root relative squared error      85.0872 %
Total Number of Instances         768

==== Detailed Accuracy By Class ====

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area
          0.868    0.451    0.782     0.868    0.823    0.443   0.817    0.881
          0.549    0.132    0.690     0.549    0.611    0.443   0.817    0.693
Weighted Avg.    0.757    0.340    0.750     0.757    0.749    0.443   0.817    0.816

==== Confusion Matrix ====

  a   b  <-- classified as
434  66 |  a = tested_negative
121  147 |  b = tested_positive
```

Lesson 4.2: 屬性選擇分類器

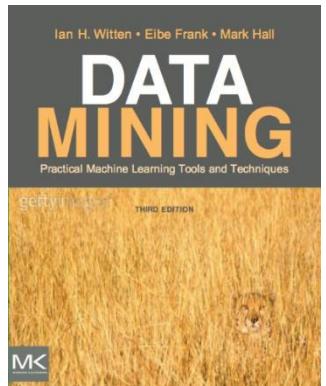
- ❖ 檢驗**AttributeSelectedClassifier**移除多餘屬性的效能 **NaiveBayes**
 - *diabetes.arff* 76.3%
 - *AttributeSelectedClassifier, NaiveBayes, WrapperSubsetEval, NaiveBayes* 75.7%
- ❖ 添加屬性的副本
 - 複製第一個屬性(*preg*); *NaiveBayes* 75.7%
 - 使用**AttributeSelectedClassifier**/同上操作 75.7%
 - 再增加9份*preg*的副本; *NaiveBayes* 68.9%
 - 使用**AttributeSelectedClassifier**/同上操作 75.7%
 - 再增加1份副本 ; *NaiveBayes* 更糟
 - 使用**AttributeSelectedClassifier**/同上操作 75.7%
- ❖ 屬性選擇能很好地去除多餘的屬性

Lesson 4.2: 屬性選擇分類器

- ❖ AttributeSelectedClassifier基於訓練數集選擇屬性
 - 即使當交叉驗證只是用來評估
 - 這是正確的作法!
 - 這裡我們使用J48
- ❖ (可能)最好在包裝中使用同樣的分類器
 - 如: 包裝J48來為J48選擇屬性
- ❖ Explorer中的一次性實驗可能不可靠

課程文本

- ❖ Section 7.1 *Attribute selection*





THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Class 4 – Lesson 3

方案獨立的屬性選擇

Scheme-independent attribute selection

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

Lesson 4.3: 方案獨立的屬性選擇

Class 1 探索Weka的介面；處理大數據

Lesson 4.1 「包裝器」屬性選擇法

Class 2 離散以及文本分類

Lesson 4.2 屬性選擇分類器

Class 3 分類規則、關聯規則、聚類

Lesson 4.3 方案獨立選擇法

Class 4 選擇屬性以及計算成本

Lesson 4.4 Attribute selection using ranking

Class 5 神經網路，學習曲線和表現優化

Lesson 4.5 Counting the cost

Lesson 4.6 Cost-sensitive classification

Lesson 4.3: 方案獨立的屬性選擇

包裝方法是直接、簡單-但非常慢

❖ 替換方案:

1. 使用單一屬性評估器，逐一評估屬性並排序(*Lesson 4.4*)

- 在此基礎上做屬性選擇

2. 將屬性子集評估與搜索方法相結合

- 可以刪除多餘屬性和不相關屬性

❖ 我們已經學習了不同的搜索方法(*Lesson 4.1*)

- 貪婪前向，後向，雙向搜尋

❖ 屬性子集評估器

- 是依賴某一特定方案的評估屬性子集的方法

- 其他子集評估器是方案獨立的(*scheme-independent*)

Lesson 4.3: 方案獨立的屬性選擇

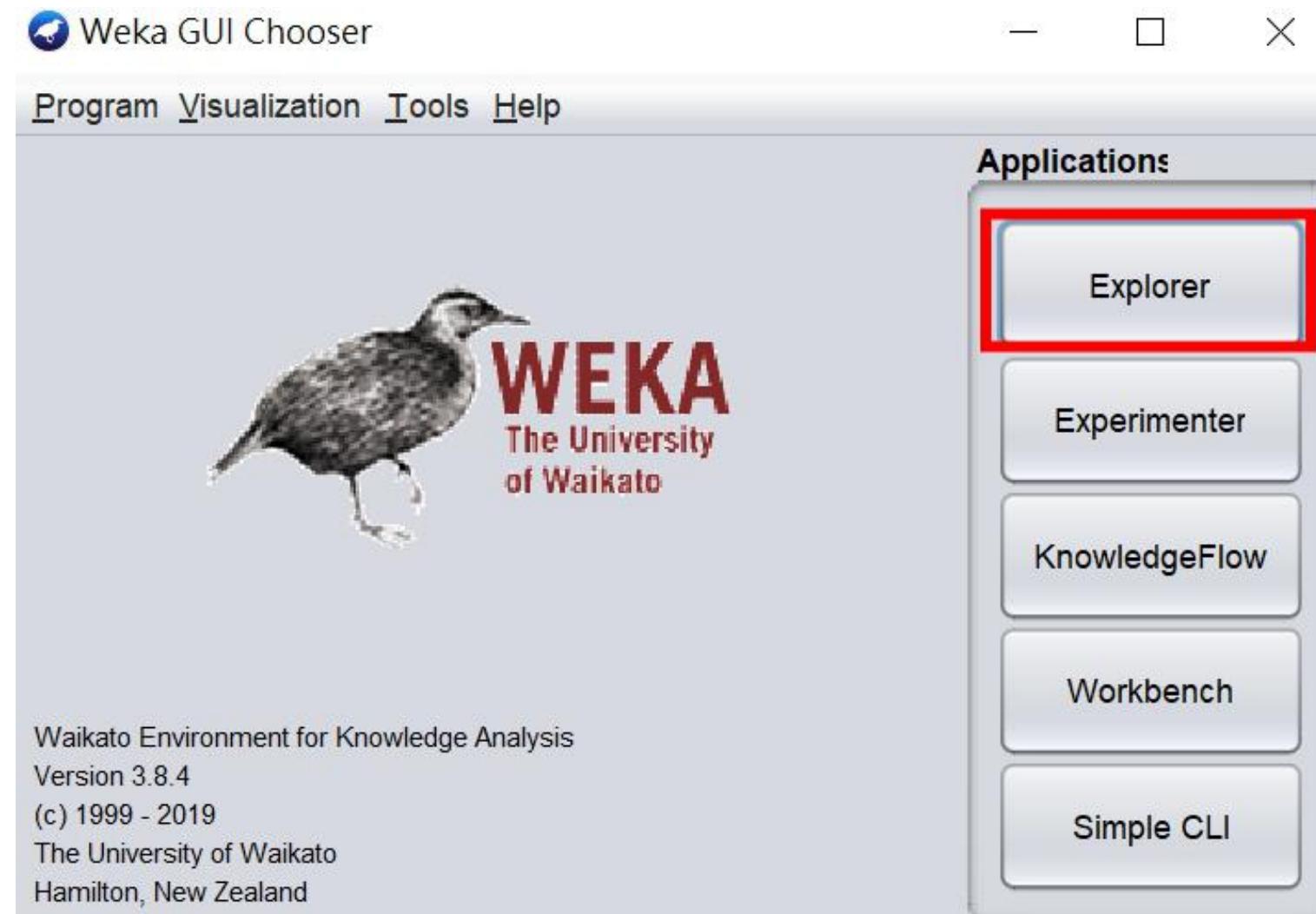
CfsSubsetEval:一種方案獨立的屬性子集評估器

- ❖ 當有一種屬性被判斷是好的，是因為它：
 - 與類屬性高度相關
 - 與其他屬性彼此之間沒有很強的相關性
- ❖ 屬性子集的良度 =
$$\frac{\sum_{\text{all attributes } x} C(x, \text{class})}{f \sum_{\text{all attributes } x} \sum_{\text{all attributes } y} C(x, y)}$$
- ❖ C 度量兩個屬性之間的相關性
- ❖ 使用了一種基於熵的度量，稱為「對稱不確定性」

Lesson 4.3: 方案獨立的屬性選擇

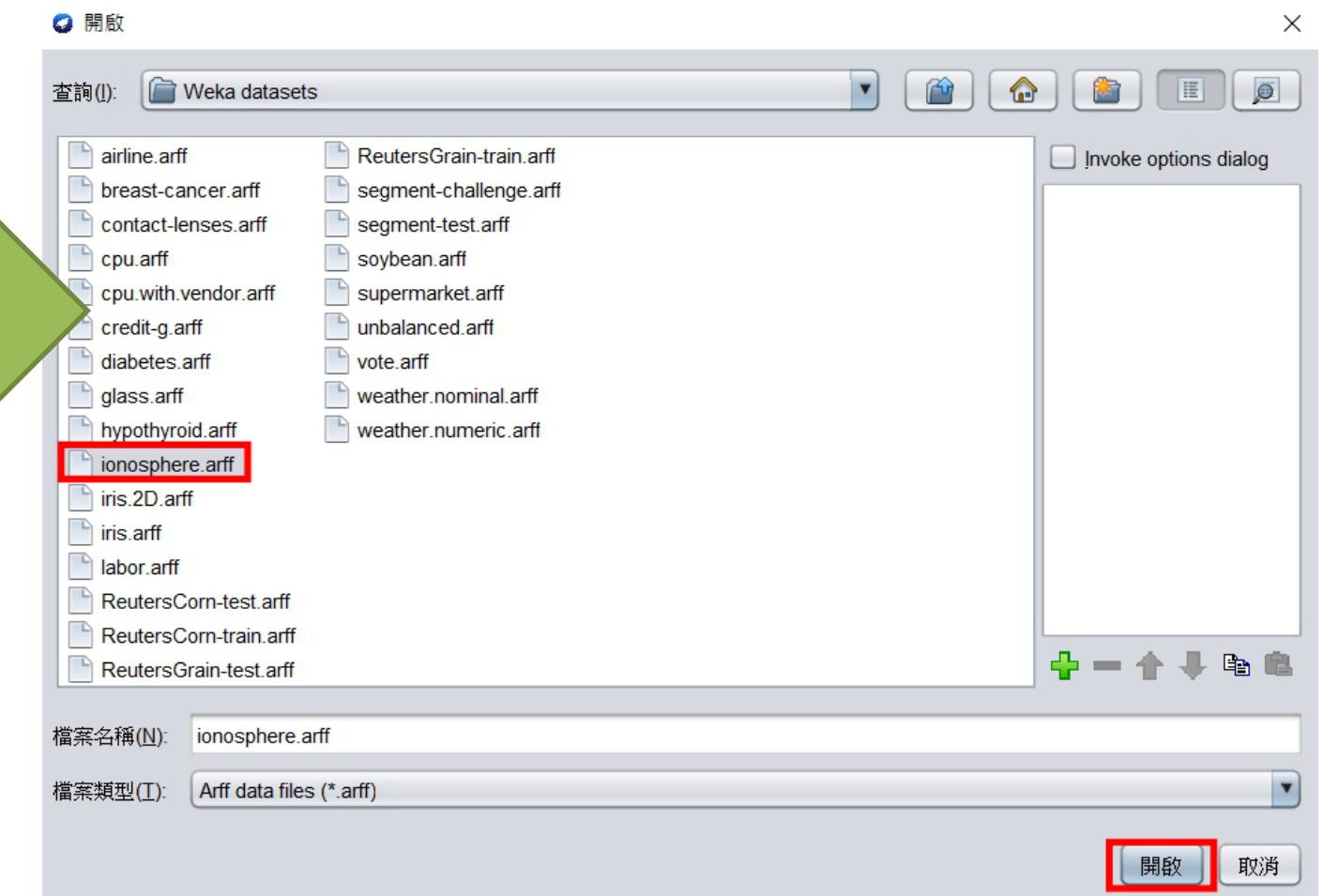
我們試著使用ionosphere數據來比較CfsSubsetEval和包裝選擇法。
首先使用Naive Bayes。

1. 開啟Weka的Explorer。



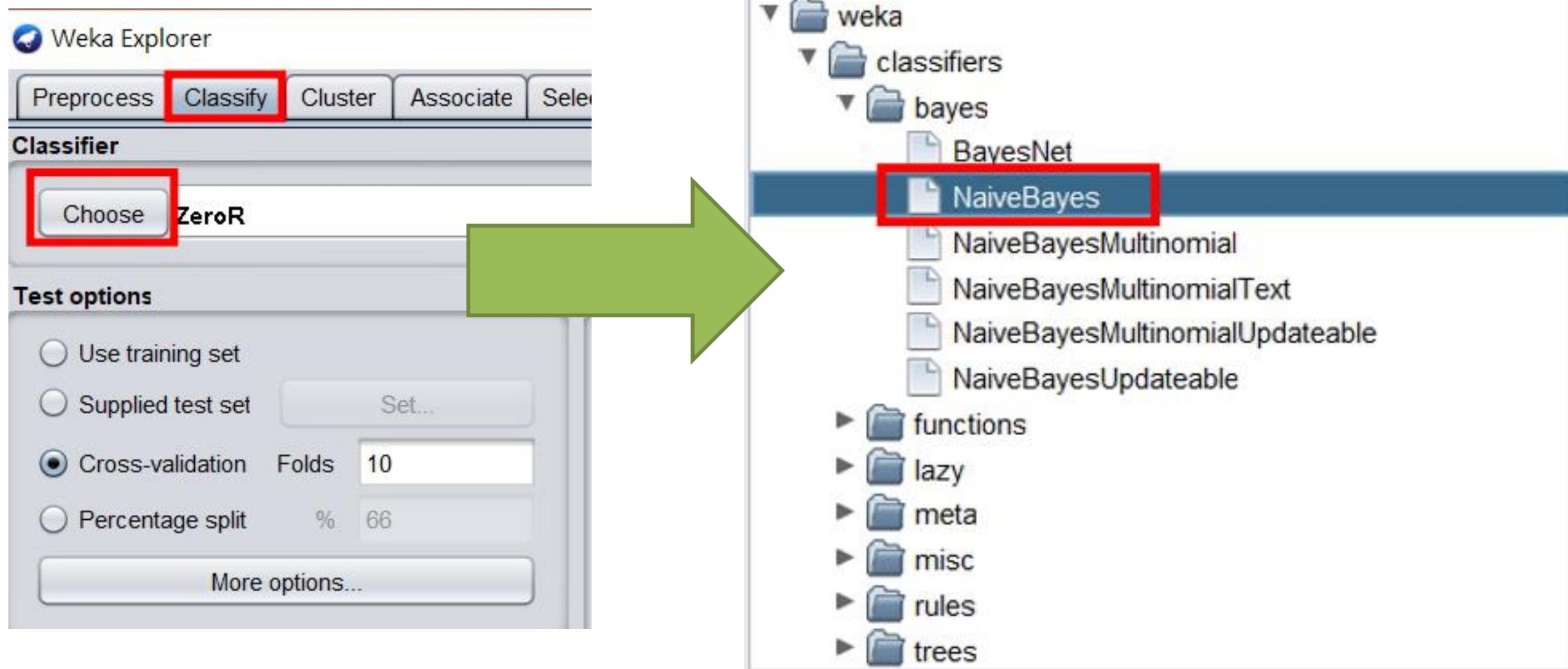
Lesson 4.3: 方案獨立的屬性選擇

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊**ionosphere.arff**的檔案後，再以左鍵單擊下方「開啟」按鈕以載入此檔案



Lesson 4.3: 方案獨立的屬性選擇

3. 切換到Classify面板點選Choose鈕，在出現的選單中左鍵單擊bayes資料夾下的NaiveBayes



Lesson 4.3: 方案獨立的屬性選擇

4. 左鍵單擊Start按鈕，執行結果如右圖，得到82.6211%準確率。

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'NaiveBayes' is chosen. Under 'Test options', 'Cross-validation' is selected with 'Folds' set to 10. A large orange arrow points from the 'Start' button in the bottom left of the main window towards the 'Classifier output' window on the right.

Classifier output

```
Time taken to build model: 0.08 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances           290          82.6211 %
Incorrectly Classified Instances        61           17.3789 %
Kappa statistic                         0.6394
Mean absolute error                     0.1736
Root mean squared error                 0.3935
Relative absolute error                  37.7001 %
Root relative squared error            82.0203 %
Total Number of Instances                351

==== Detailed Accuracy By Class ====

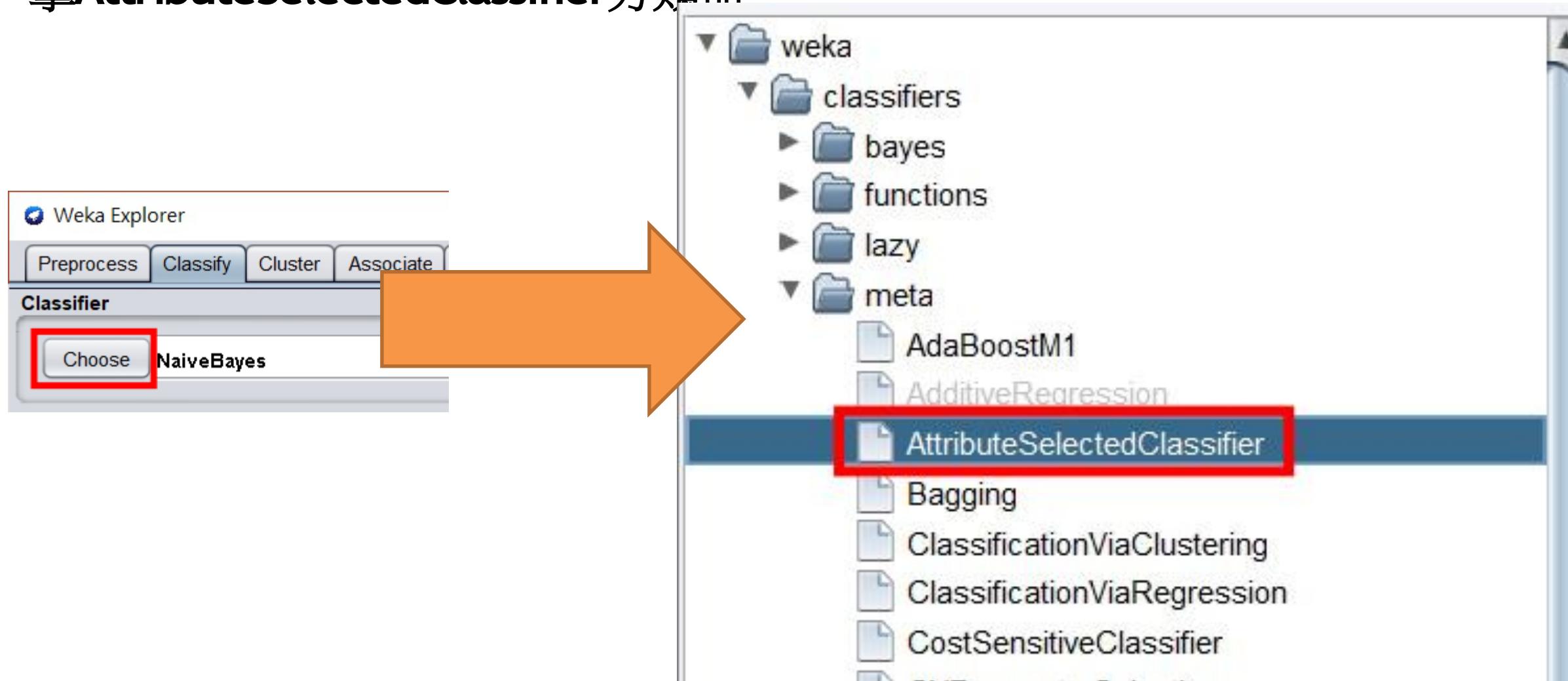
           TP Rate   FP Rate   Precision   Recall   F-Measure   MCC   ROC Area   PRC Area   Cl
           0.865     0.196     0.712      0.865     0.781      0.648    0.935     0.917     b
           0.804     0.135     0.914      0.804     0.856      0.648    0.935     0.958     g
Weighted Avg.       0.826     0.157     0.842      0.826     0.829      0.648    0.935     0.943

==== Confusion Matrix ====

    a    b    <-- classified as
109  17 |  a = b
 44 181 |  b = g
```

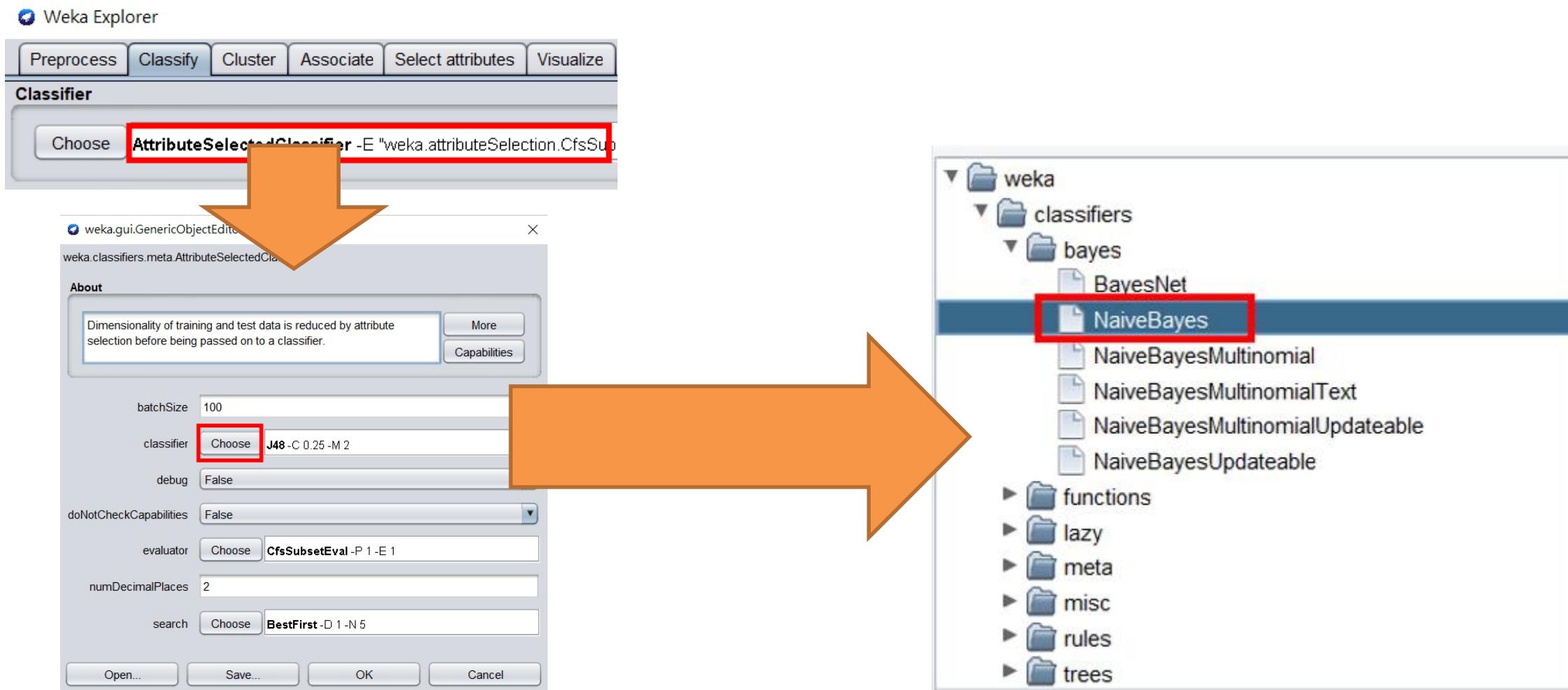
Lesson 4.3: 方案獨立的屬性選擇

5.回到Classify面板，左鍵單擊Choose按鈕，在出現的選單中以左鍵單擊AttributeSelectedClassifier分類器。



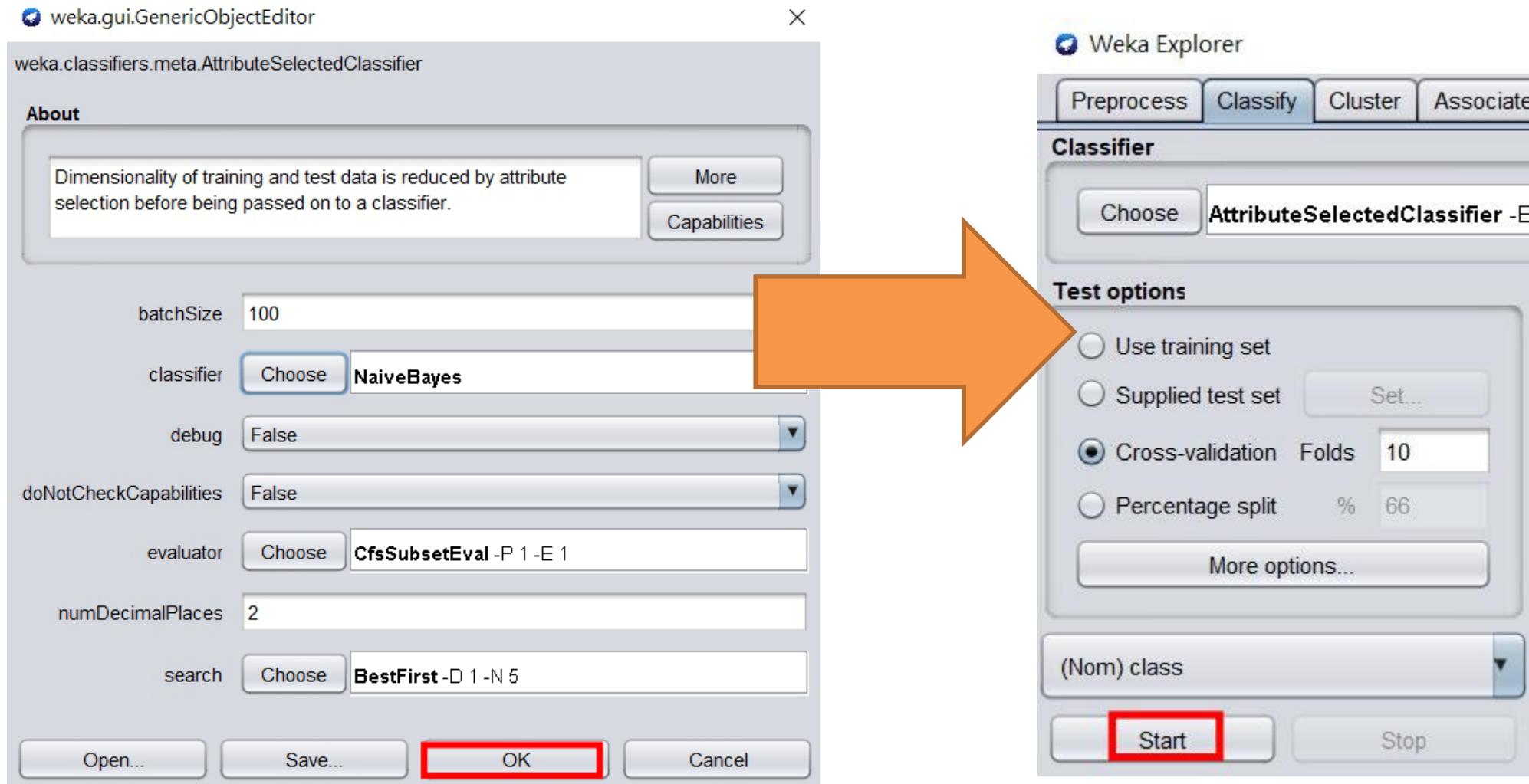
Lesson 4.3: 方案獨立的屬性選擇

6. 左鍵單擊分類器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊參數classifier的Choose按鈕，並在出現的選單中左鍵單擊NaiveBayes分類器。



Lesson 4.3: 方案獨立的屬性選擇

7. 左鍵單擊下方OK按鈕回到Classify面板，以左鍵單擊Start按鈕。



Lesson 4.3: 方案獨立的屬性選擇

執行結果，得到88.604%準確率。

Classifier output

Correctly Classified Instances	311	88.604 %
Incorrectly Classified Instances	40	11.396 %
Kappa statistic	0.7558	
Mean absolute error	0.1173	
Root mean squared error	0.3097	
Relative absolute error	25.4643 %	
Root relative squared error	64.5594 %	
Total Number of Instances	351	

==== Detailed Accuracy By Class ====

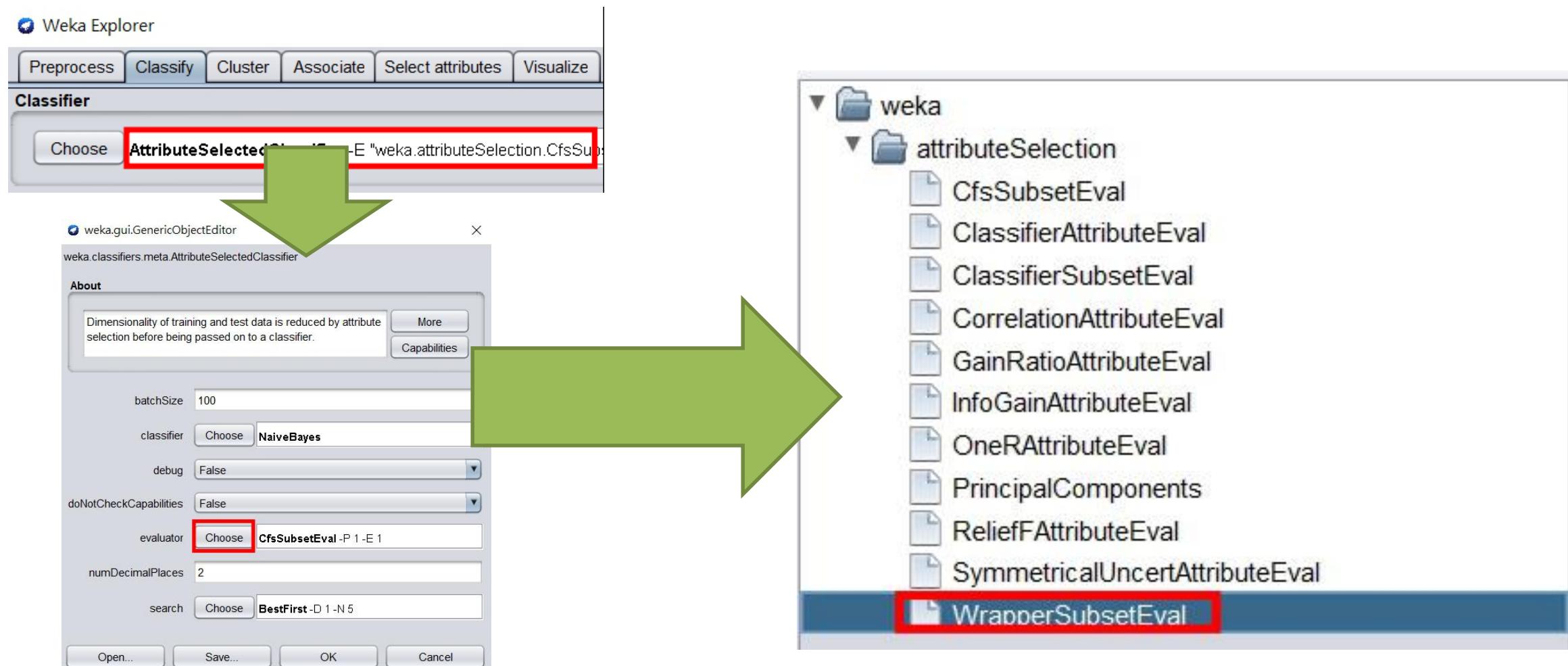
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Ar
	0.873	0.107	0.821	0.873	0.846	0.757	0.936
	0.893	0.127	0.926	0.893	0.910	0.757	0.936
Weighted Avg.	0.886	0.120	0.888	0.886	0.887	0.757	0.936

==== Confusion Matrix ====

a	b	<-- classified as
110	16	a = b
24	201	b = g

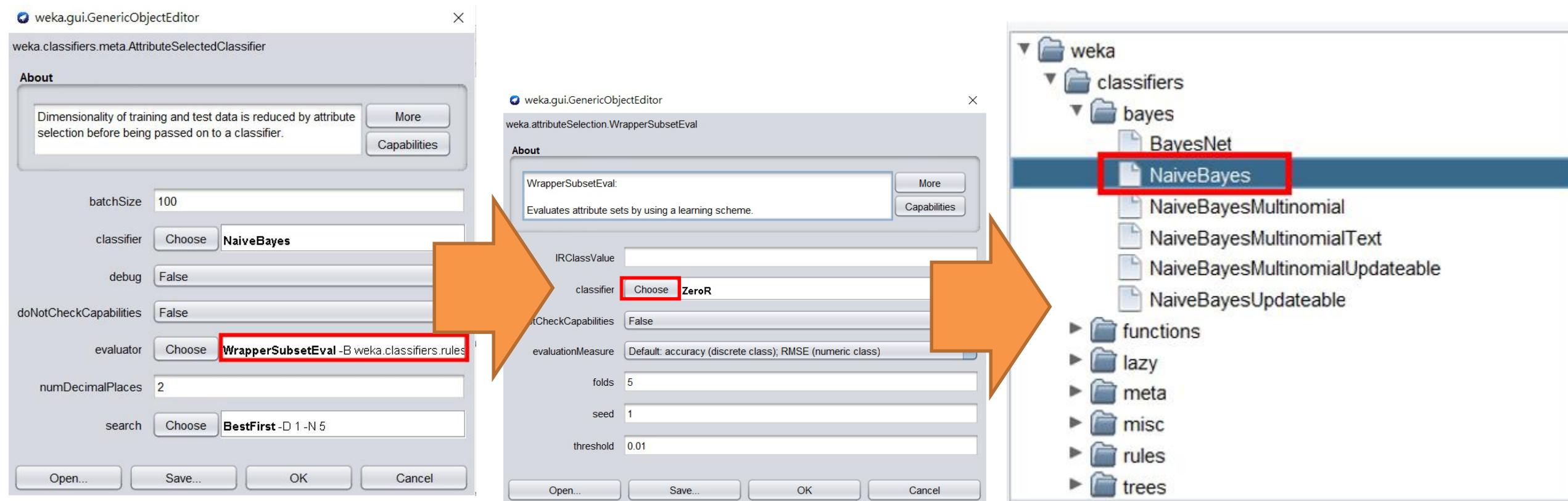
Lesson 4.3: 方案獨立的屬性選擇

8. 左鍵單擊分類器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊參數evaluator的Choose按鈕，並在出現的選單中左鍵單擊WrapperSubsetEval屬性選擇分類器。



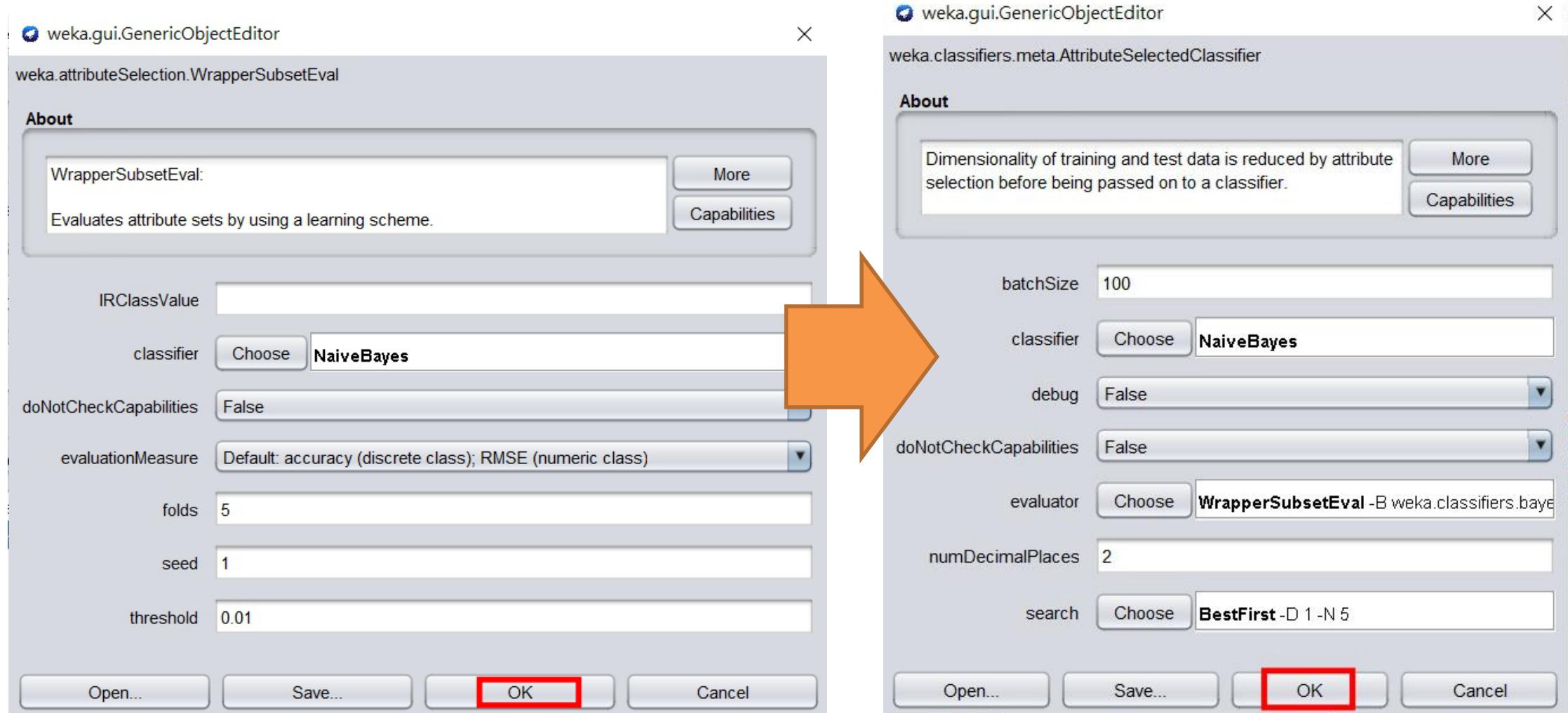
Lesson 4.3: 方案獨立的屬性選擇

9. 左鍵單擊屬性選擇器名稱(左圖紅框處)，開啟配置視窗(中間圖)。在配置視窗中左鍵單擊Choose按鈕，再以左鍵單擊NaiveBayes分類器。



Lesson 4.3: 方案獨立的屬性選擇

10. 左鍵單擊下方OK按鈕回到分類器配置視窗，再以左鍵單擊下方OK按鈕回到Classify面板。



Lesson 4.3: 方案獨立的屬性選擇

11. 左鍵單擊Start按鈕，執行結果如右圖，得到90.8832%準確率。

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'AttributeSelectedClassifier' is chosen. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. A large orange arrow points from the 'Start' button in the bottom left to the 'Classifier output' window on the right. The 'Classifier output' window displays classification statistics and a confusion matrix.

Classifier output

		90.8832 %
Correctly Classified Instances	319	90.8832 %
Incorrectly Classified Instances	32	9.1168 %
Kappa statistic	0.8012	
Mean absolute error	0.1132	
Root mean squared error	0.2768	
Relative absolute error	24.5918 %	
Root relative squared error	57.7041 %	
Total Number of Instances	351	

= Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Ar
0.865	0.067	0.879	0.865	0.872	0.801	0.945	
0.933	0.135	0.925	0.933	0.929	0.801	0.945	
Weighted Avg.	0.909	0.110	0.909	0.909	0.909	0.801	0.945

== Confusion Matrix ==

		a b <-- classified as
109 17		a = b
15 210		b = g

Lesson 4.3: 方案獨立的屬性選擇

使用 **ionosphere.arff** 比較 **CfsSubsetEval** 和包裝選擇法(**Wrapper selection**)

	<i>NaiveBayes</i>	<i>IBk</i>	<i>J48</i>
❖ 沒有屬性選擇	83%	86%	91%
❖ 有屬性選擇(使用 AttributeSelectedClassifier)			
– CfsSubsetEval (非常快)	89%	89%	92%
– 包裝選擇法(Wrapper selection) (非常慢) (包裝器中使用相應的分類器，例如 <i>IBk</i> 的包裝器使用 <i>IBk</i>)	91%	89%	90%
❖ 結論: CfsSubsetEval 的表現雖然跟 Wrapper 差不多，但速度更快			

Lesson 4.3: 方案獨立的屬性選擇

Weka中的屬性子集評估器

方案相依(Scheme-dependent)

- ❖ WrapperSubsetEval (使用內部交叉驗證)
- ❖ ClassifierSubsetEval (分離held-out測試集)

方案獨立(Scheme-independent)

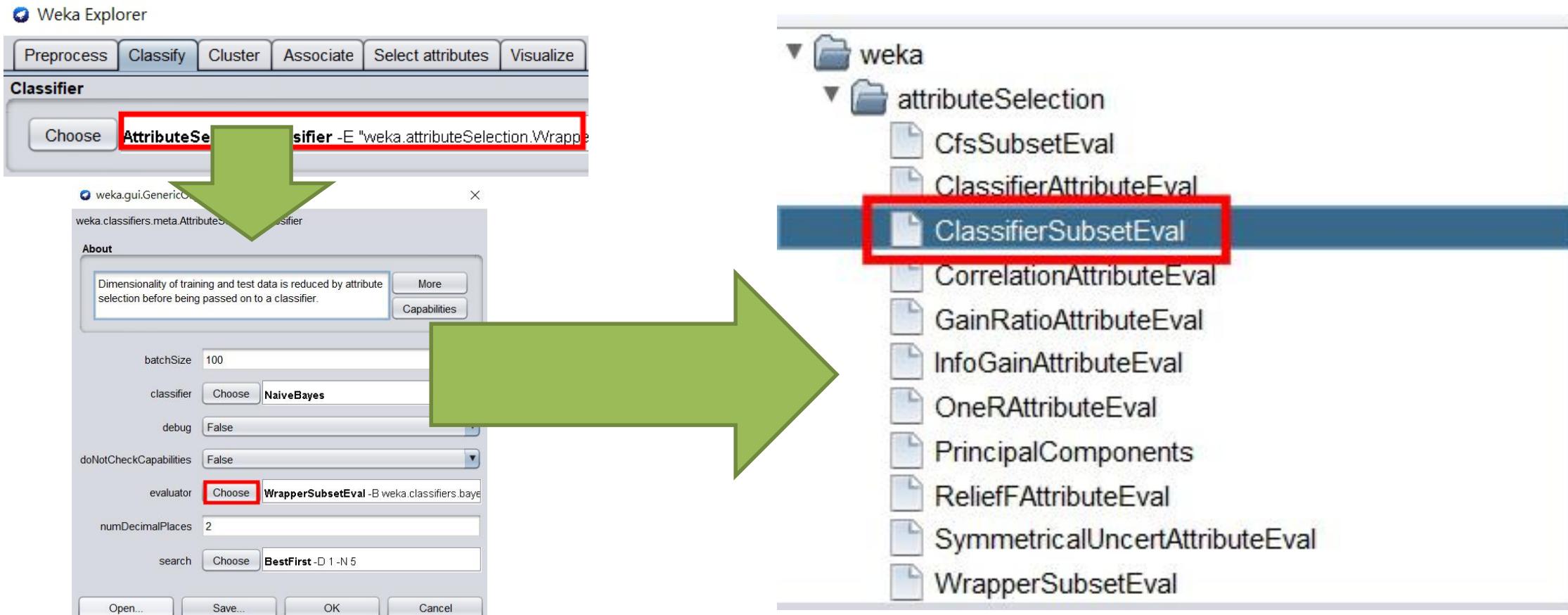
- ❖ CfsSubsetEval
 - 考慮每個屬性的預測值以及相互冗餘的程度
- ❖ ConsistencySubsetEval
 - 以訓練數據中類值相對於屬性的一致性來評估
 - 尋找一致性不低於全集的最小屬性集

(還有一種是元評估器，它包含了其他操作)

Lesson 4.3: 方案獨立的屬性選擇

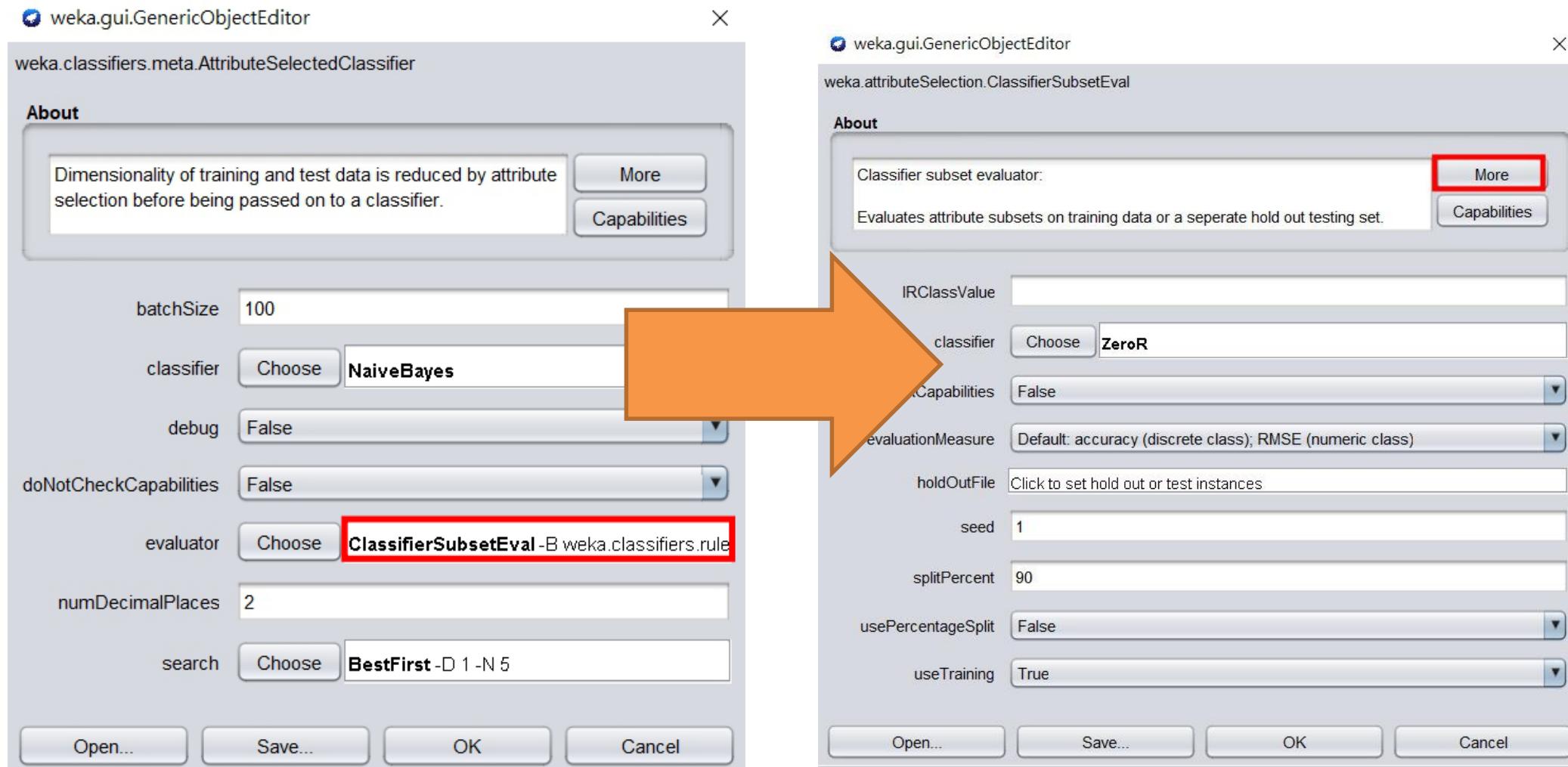
這是我們剛剛提到的**ConsistencySubsetEval**，

1. 左鍵單擊分類器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊參數**evaluator**的**Choose**按鈕，並在出現的選單中左鍵單擊**ClassifierSubsetEval**屬性選擇器。



Lesson 4.3: 方案獨立的屬性選擇

2. 左鍵單擊屬性選擇器名稱(左圖紅框處)，開啟配置視窗(右圖)。在配置視窗中左鍵單擊More按鈕查看屬性選擇器相關資訊。



Lesson 4.3: 方案獨立的屬性選擇

它以類值的一致性來評估屬性子集，若要學習這種方法，你可以閱讀它的相關文章。

Information X

NAME
weka.attributeSelection.ClassifierSubsetEval

SYNOPSIS
Classifier subset evaluator:

Evaluates attribute subsets on training data or a separate hold out testing set. Uses a classifier to estimate the 'merit' of a set of attributes.

OPTIONS

seed -- The random seed to use for randomizing the training data prior to performing a percentage split

useTraining -- Use training data instead of hold out/test instances.

classifier -- Classifier to use for estimating the accuracy of subsets

IRClassValue -- The class label, or 1-based index of the class label, to use when evaluating subsets with an IR metric (such as f-measure or AUC). Leaving this unset will result in the class frequency weighted average of the metric being used.

doNotCheckCapabilities -- If set, evaluator capabilities are not checked before evaluator is built (Use with caution to reduce runtime).

evaluationMeasure -- The measure used to evaluate the performance of attribute combinations.

holdOutFile -- File containing hold out/test instances.

splitPercent -- The percentage split to use

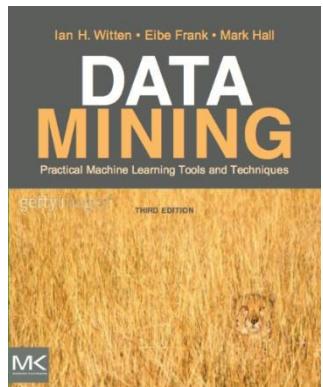
usePercentageSplit -- Evaluate using a percentage split on the training data

Lesson 4.3: 方案獨立的屬性選擇

- ❖ 屬性子集選擇包含
 - 一個子集評估方法
 - 一個搜尋方法
- ❖ 一些方法是方案相依的(scheme-dependent)
 - 如: 包裝法; 但非常慢
- ❖ ... 其他方法是方案獨立的(scheme-independent)
 - 如: CfsSubsetEval; 比較快速
- ❖ 甚至更快 ... 當使用單一屬性評估器並進行分等的時候 (下一堂課會提到)

課程文本

- ❖ Section 7.1 *Attribute selection*





THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Class 4 – Lesson 4

使用分等的快速屬性選擇

Fast attribute selection using ranking

Ian H. Witten

Department of Computer Science University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 4.4: 使用分等的快速屬性選擇

Class 1 探索Weka的介面；處理大數據

Lesson 4.1 「包裝器」屬性選擇法

Class 2 離散以及文本分類

Lesson 4.2 屬性選擇分類器

Class 3 分類規則、關聯規則、聚類

Lesson 4.3 方案獨立選擇法

Class 4 選擇屬性以及計算成本

Lesson 4.4 使用分等進行屬性選擇

Class 5 神經網路，學習曲線和表現優化

Lesson 4.5 Counting the cost

Lesson 4.6 Cost-sensitive classification

Lesson 4.4: 使用分等的快速屬性選擇

- ❖ 屬性子集選擇包含:
 - 子集評估方法
 - 搜尋方法
- ❖ 搜尋的速度很慢!

- ❖ 替代方案: 使用分等的單一屬性評估器
 - 可以消除無關的屬性... 但不是多餘的屬性
- ❖ 選擇單屬性評估器時，選擇「ranking」搜索方法

Lesson 4.4: 使用分等的快速屬性選擇

我們之前已經看到過屬性評估的幾個指標

- ❖ OneR 使用單一屬性分類器的準確率 OneRAttributeEval
- ❖ C4.5 (i.e. J48) 使用信息增益 InfoGainAttributeEval
- ... 事實上, 它使用的是增加比率 GainRatioAttributeEval
- ❖ CfsSubsetEval 使用「對稱的不確定性」 SymmetricalUncertAttributeEval

ranker 搜尋方法根據屬性的評估對屬性進行排序

- ❖ 參數
 - 要保留的屬性數量(預設: 保留所有)
 - 或丟棄其評估值低於閾值的屬性(預設值 $-\infty$)
 - 可以指定一組要忽略的屬性

Lesson 4.4: 使用分等的快速屬性選擇

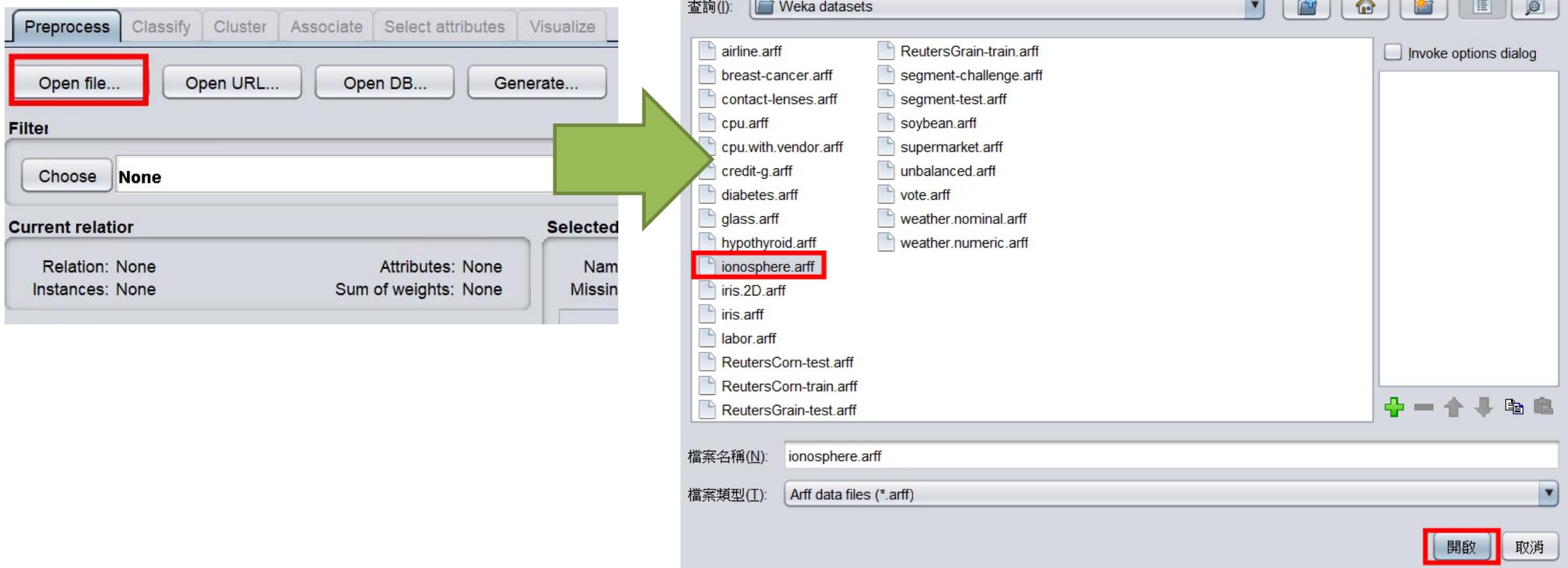
我們用 ionosphere 數據比較 GainRatioAttributeEval 和其他方法。

1. 開啟 Weka 的 Explorer



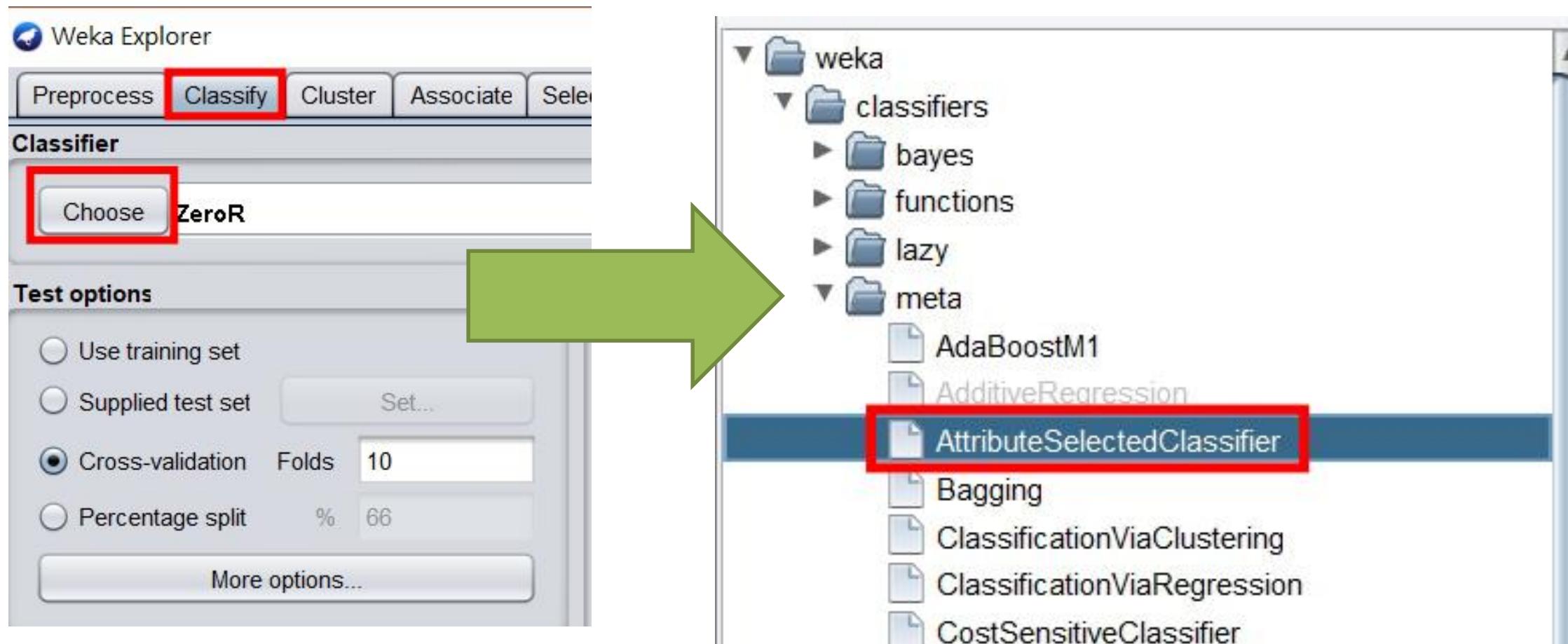
Lesson 4.4: 使用分等的快速屬性選擇

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊**ionosphere.arff**的檔案後，再以左鍵單擊下方「開啟」按鈕以載入此檔案



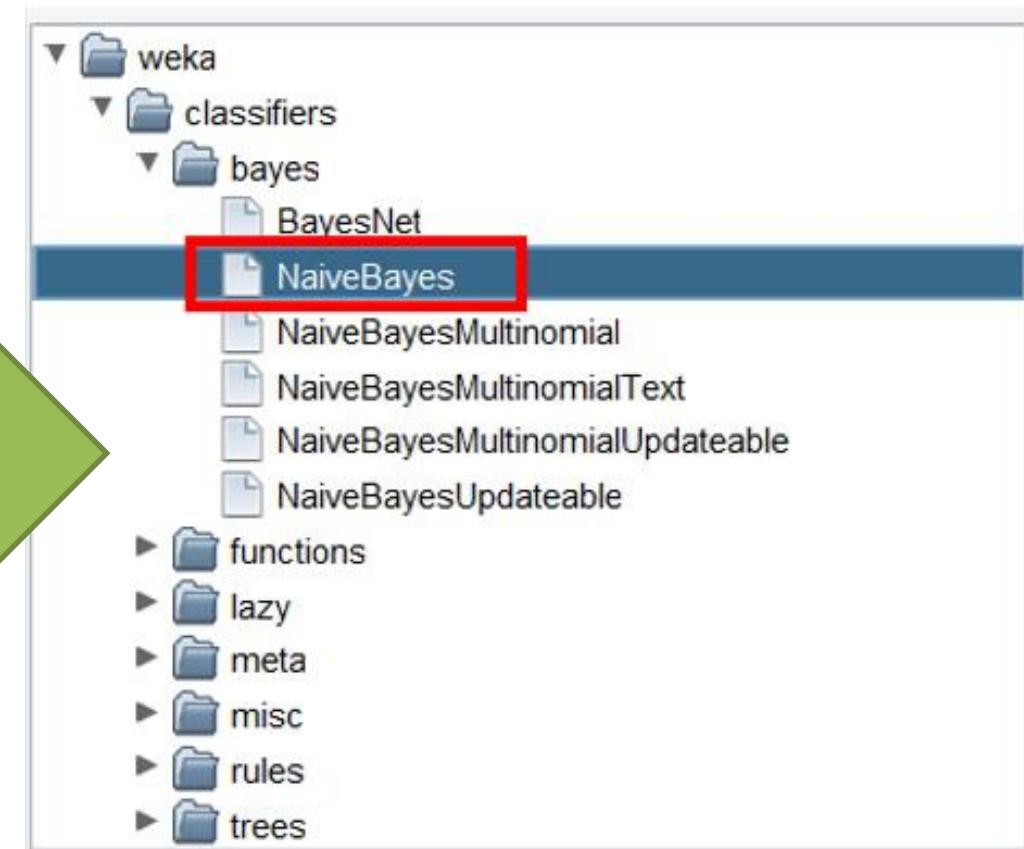
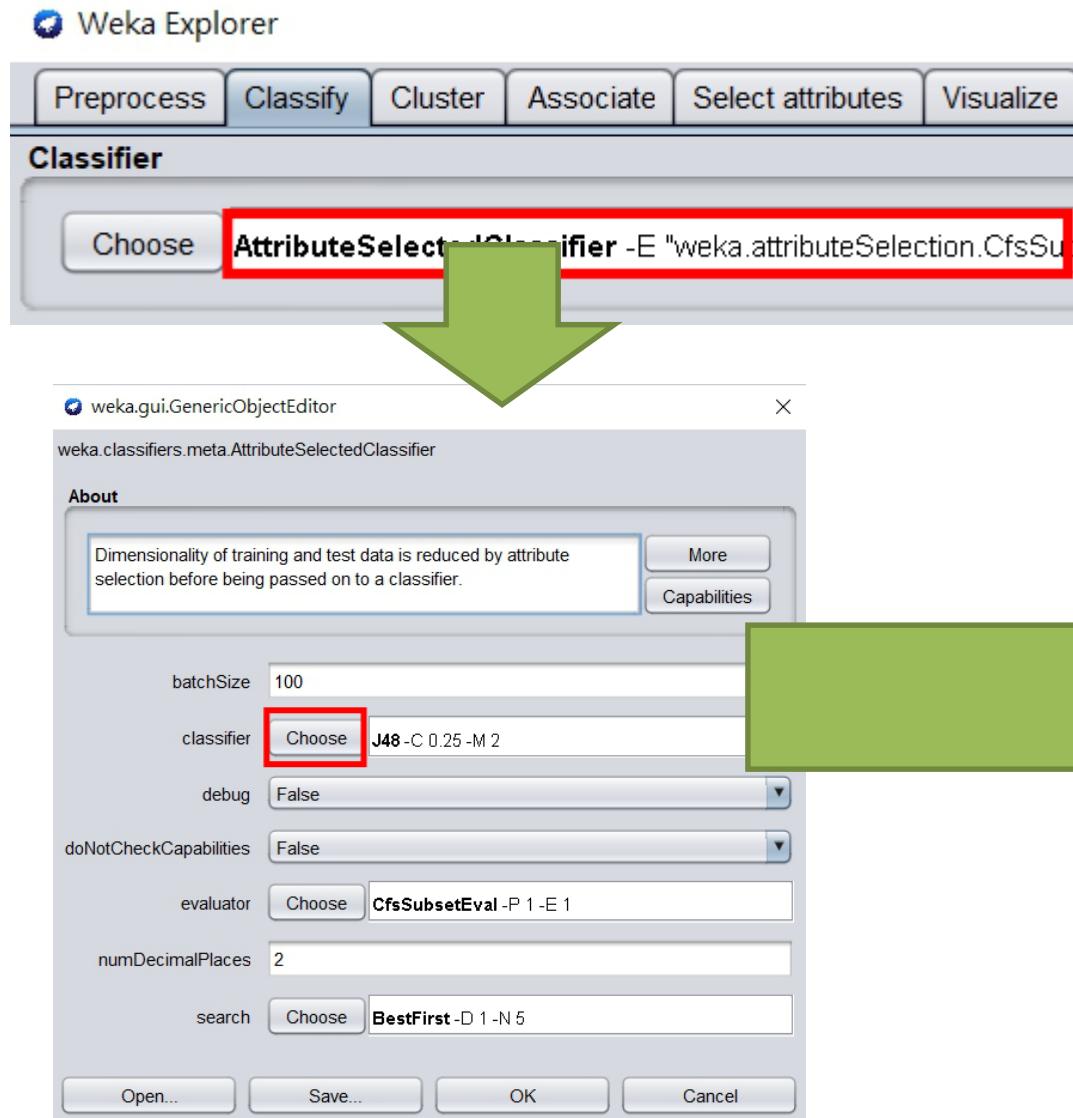
Lesson 4.4: 使用分等的快速屬性選擇

3. 切換到Classify界面點選Choose鈕，在出現的選單中左鍵單擊meta資料夾下的AttributeSelectedClassifier



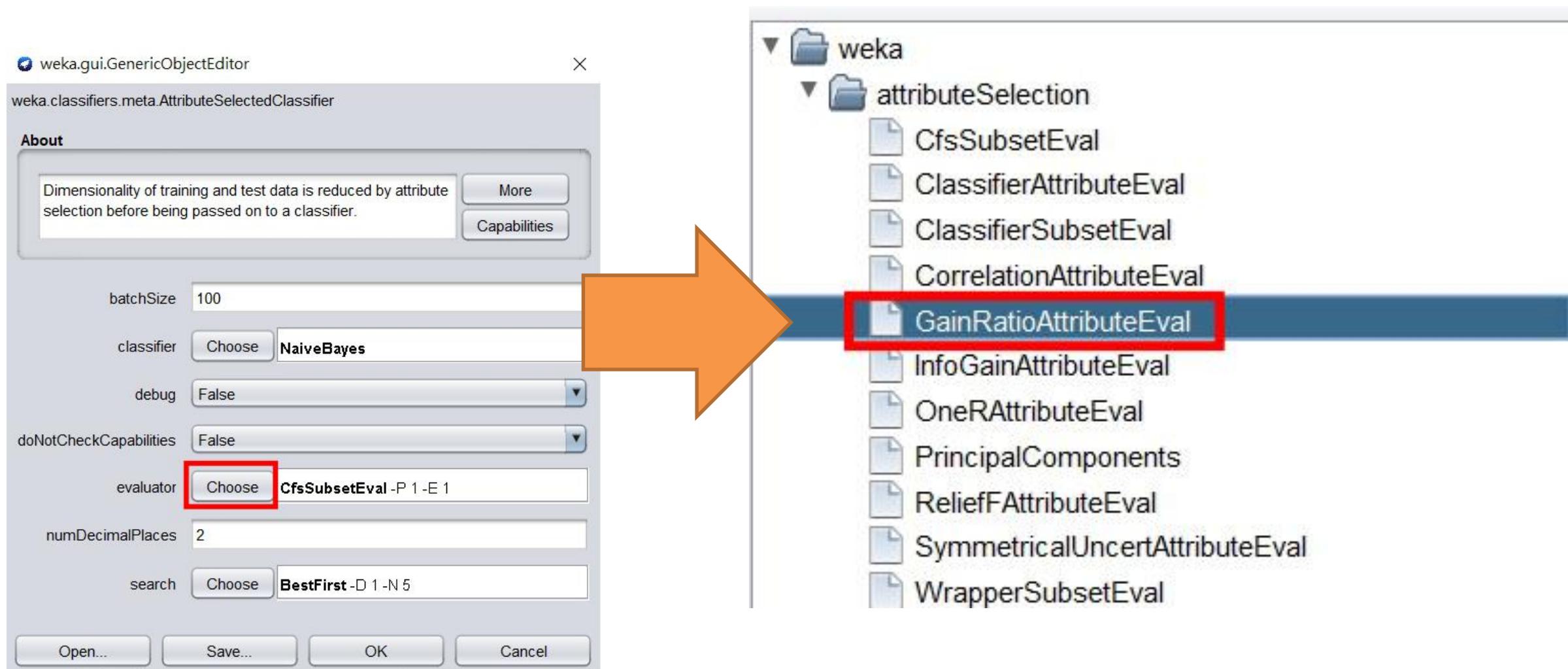
Lesson 4.4: 使用分等的快速屬性選擇

4. 左鍵單擊分類器名稱(左上圖紅框處)，開啟配置視窗(左下圖)。在配置視窗中左鍵單擊參數classifier的Choose按鈕，並在出現的選單中左鍵單擊NaiveBayes分類器。



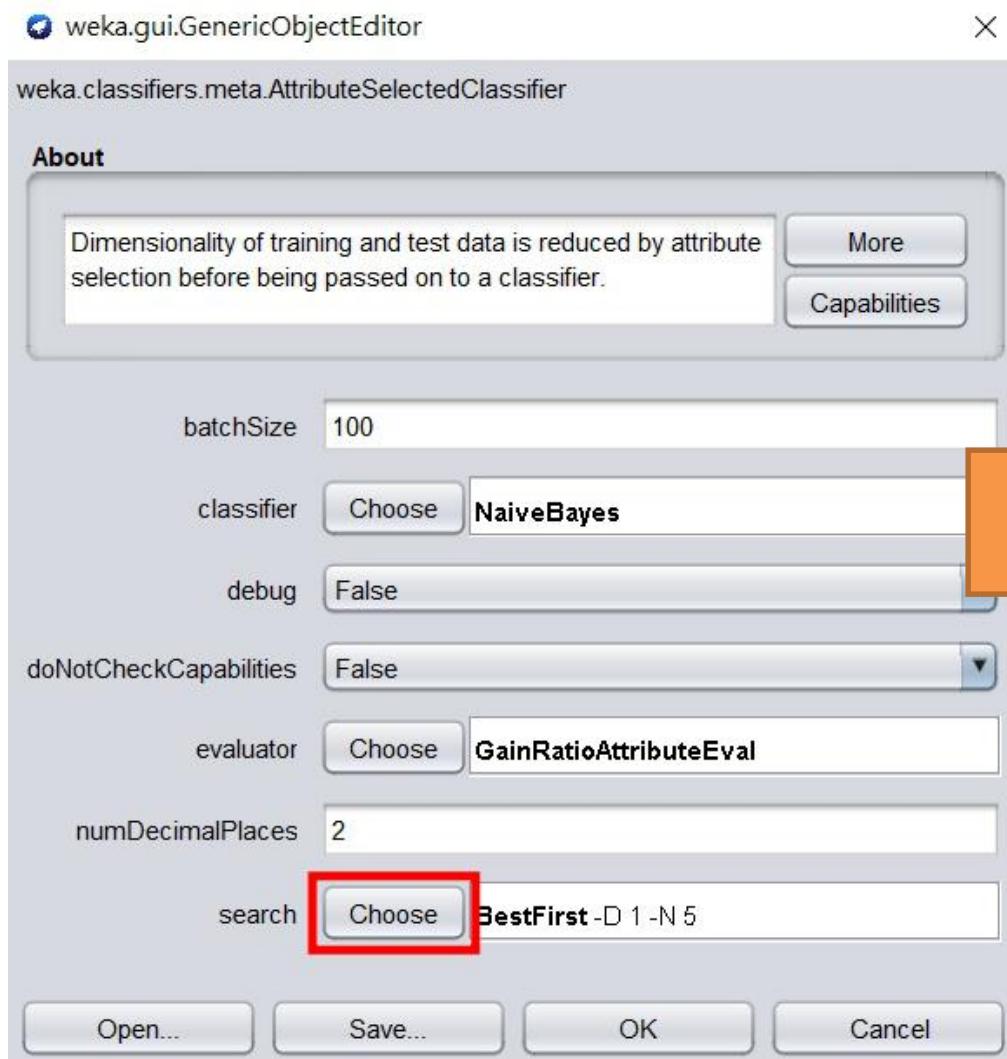
Lesson 4.4: 使用分等的快速屬性選擇

5. 在配置視窗中左鍵單擊參數evaluator的Choose按鈕，並在出現的選單中左鍵單擊GainRatioAttributeEval屬性選擇器。



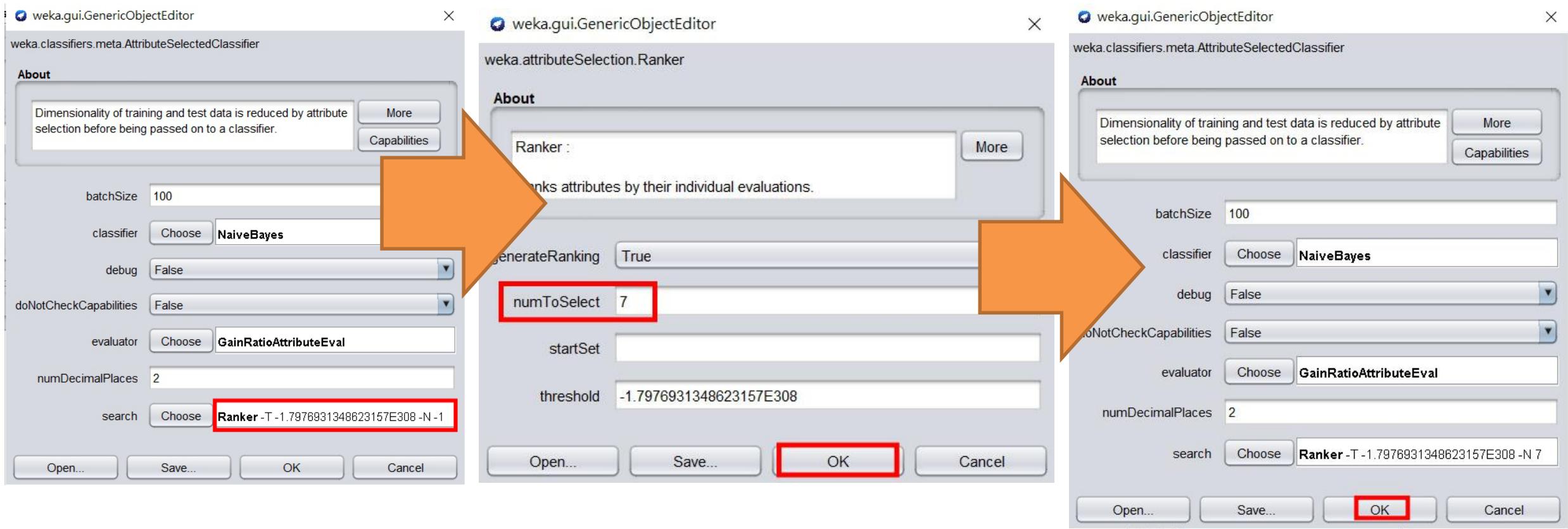
Lesson 4.4: 使用分等的快速屬性選擇

6. 在配置視窗中左鍵單擊參數search的Choose按鈕，並在出現的選單中左鍵單擊Ranker搜尋法。



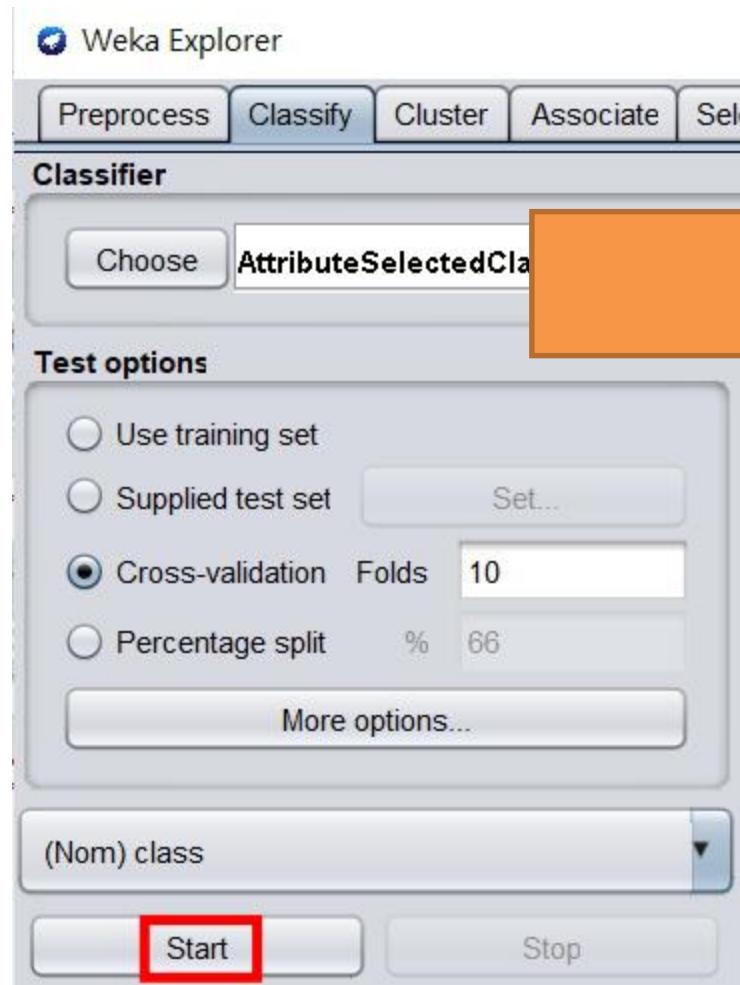
Lesson 4.4: 使用分等的快速屬性選擇

7. 左鍵單擊搜尋法名稱(最左圖紅框處)，開啟配置視窗(中間圖)。在配置視窗中將參數numToSelect改為7，再以左鍵單擊OK按鈕回到分類器配置視窗，然後以左鍵單擊OK按鈕。



Lesson 4.4: 使用分等的快速屬性選擇

8. 左鍵單擊Start按鈕，執行結果如右圖，得到89.7436%準確率。



		Classifier output						
Correctly Classified Instances		315		89.7436 %				
Incorrectly Classified Instances		36		10.2564 %				
Kappa statistic		0.774						
Mean absolute error		0.1282						
Root mean squared error		0.3063						
Relative absolute error		27.8471 %						
Root relative squared error		63.8534 %						
Total Number of Instances		351						
==== Detailed Accuracy By Class ====								
		TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Ar.
		0.825	0.062	0.881	0.825	0.852	0.775	0.930
		0.938	0.175	0.906	0.938	0.921	0.775	0.930
	Weighted Avg.	0.897	0.134	0.897	0.897	0.897	0.775	0.930
==== Confusion Matrix ====								
		a	b	<- classified as				
104	22		a = b					
14	211		b = g					

Lesson 4.3: 方案獨立的屬性選擇

使用 **ionosphere.arff** 比較 **CfsSubsetEval** 和包裝選擇法(**Wrapper selection**)

	<i>NaiveBayes</i>	<i>IBk</i>	<i>J48</i>
❖ 沒有屬性選擇	83%	86%	91%
❖ 有屬性選擇(使用 AttributeSelectedClassifier)			
– CfsSubsetEval (非常快)	89%	89%	92%
– 包裝選擇法(Wrapper selection) (非常慢) (包裝器中使用相應的分類器，例如 <i>IBk</i> 的包裝器使用 <i>IBk</i>)	91%	89%	90%
❖ 結論: CfsSubsetEval 的表現雖然跟 Wrapper 差不多，但速度更快			

Lesson 4.3: 方案獨立的屬性選擇

使用`ionosphere.arff`讓`GainRatioAttributeEval`跟其他方法比較

(灰色部分這部分是我們上一節課得到的結果，沒有屬性選擇。)

	<i>NaiveBayes</i>	<i>IBk</i>	<i>J48</i>
❖ 沒有屬性選擇	83%	86%	91%
❖ 有屬性選擇(使用 <code>AttributeSelectedClassifier</code>)			
- <code>CfsSubsetEval</code> (非常快)	89%	89%	92%
- 包裝選擇法(Wrapper selection) (非常慢)	91%	89%	90%
- <code>GainRatioAttributeEval</code> 保留 7 種屬性	90%	86%	91%
❖ 非常快...			
但是性能對保留的屬性數量很敏感			

Lesson 4.4: 使用分等的快速屬性選擇

Weka中的屬性評估器

- ❖ OneRAttributeEval
- ❖ InfoGainAttributeEval
- ❖ GainRatioAttributeEval
- ❖ SymmetricalUncertaintyAttributeEval

另外還有

- ❖ ChiSquaredAttributeEval - 計算類中各屬性的 χ^2 統計量
- ❖ SVMAttributeEval - 使用支持向量機(SVM)確定屬性的值
- ❖ ReliefFAttributeEval - 基於實例的屬性求值程序
- ❖ PrincipalComponents - 主成分變換，選擇最大的特徵向量
- ❖ LatentSemanticAnalysis - 執行潛在語義分析和轉換

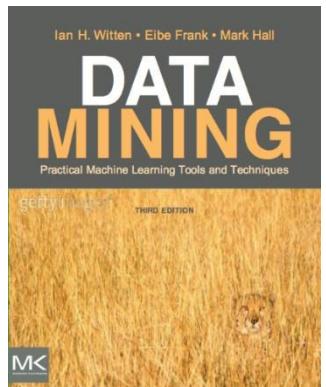
(還有一種是元評估器，它包含了其他操作)

Lesson 4.4: 使用分等的快速屬性選擇

- ❖ 屬性子集評估
 - 涉及搜索必然是緩慢的
- ❖ 單一屬性評估器
 - 包含分等，這會快很多
 - 指定一個合適的截止是很困難的 (需要做實驗)
 - 不能應對冗餘屬性
(如:你有一個屬性的副本，它們將不斷被選入，因為單一屬性評估器會還是對每一屬性單獨進行評估)
- ❖ 許多單一屬性評估器是基於我們已經學習過的機器學習方法。

課程文本

- ❖ Section 7.1 *Attribute selection*





THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Class 4 – Lesson 5

計算成本

Counting the cost

Ian H. Witten

Department of Computer Science University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 4.5: 計算成本

Class 1 探索Weka的介面；處理大數據

Class 2 離散以及文本分類

Class 3 分類規則、關聯規則、聚類

Class 4 選擇屬性以及計算成本

Class 5 神經網路，學習曲線和表現優化

Lesson 4.1 「包裝器」屬性選擇法

Lesson 4.2 屬性選擇分類器

Lesson 4.3 方案獨立選擇法

Lesson 4.4 使用分等進行屬性選擇

Lesson 4.5 計算成本

Lesson 4.6 Cost-sensitive classification



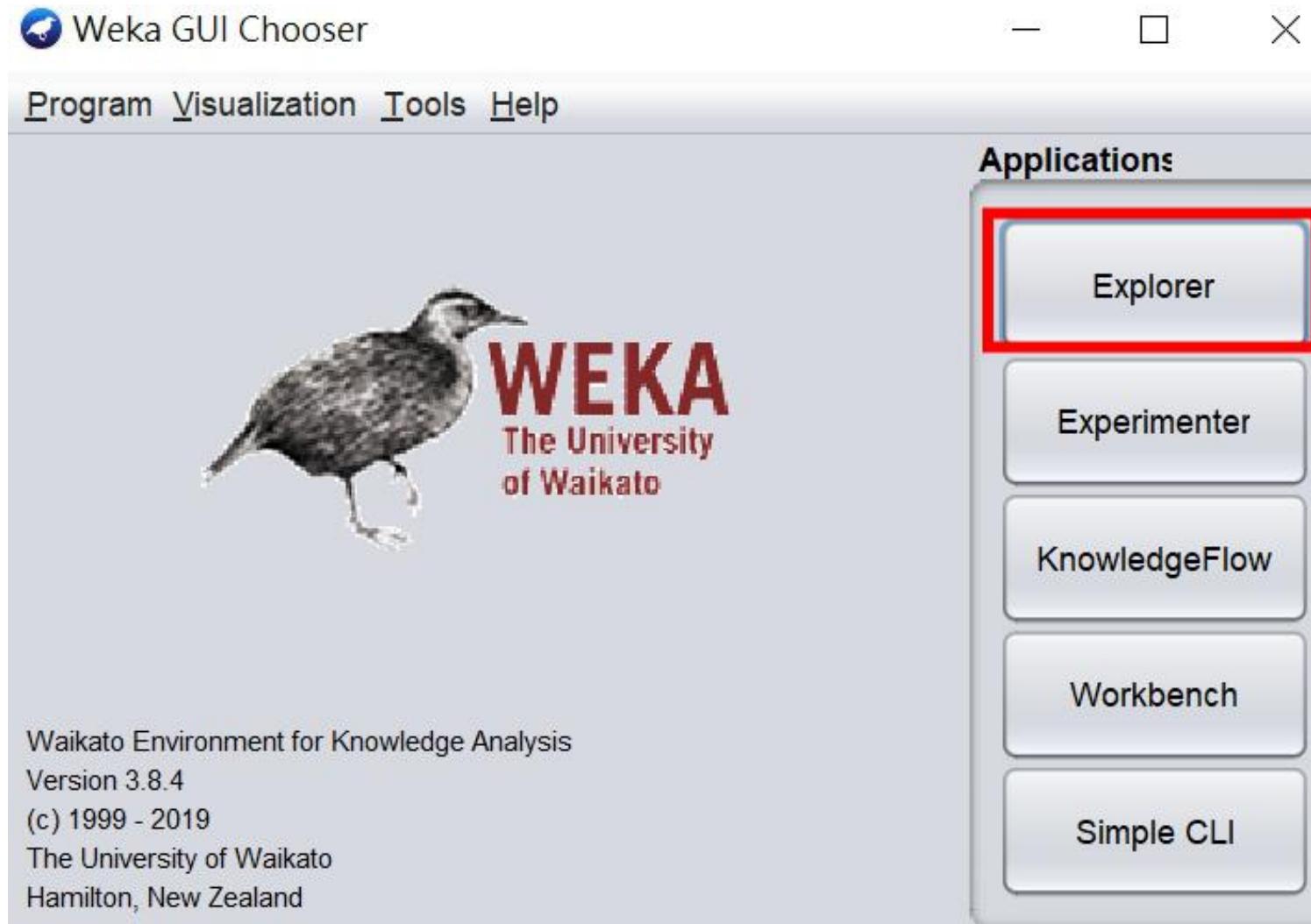
Lesson 4.5: 計算成本

什麼是成功？

- ❖ 評價資料探勘方法，至今我們使用過分類正確率
(基於測試數據、預留(*holdout*)或交叉驗證(*cross-validation*))
- ❖ 不同種類的誤差的成本代價是不同的
- ❖ 最小化整體誤差可能是不合適的
藉由第二課的ROC曲線，反映了不同誤差的成本權衡
- ❖ 信用評級數據集 **credit-g.arff**
將「壞」客戶歸類到「好」客戶的代價高於將「好」
客戶歸類到「壞」客戶。
- ❖ 經濟的模型：假設成本比是5比1

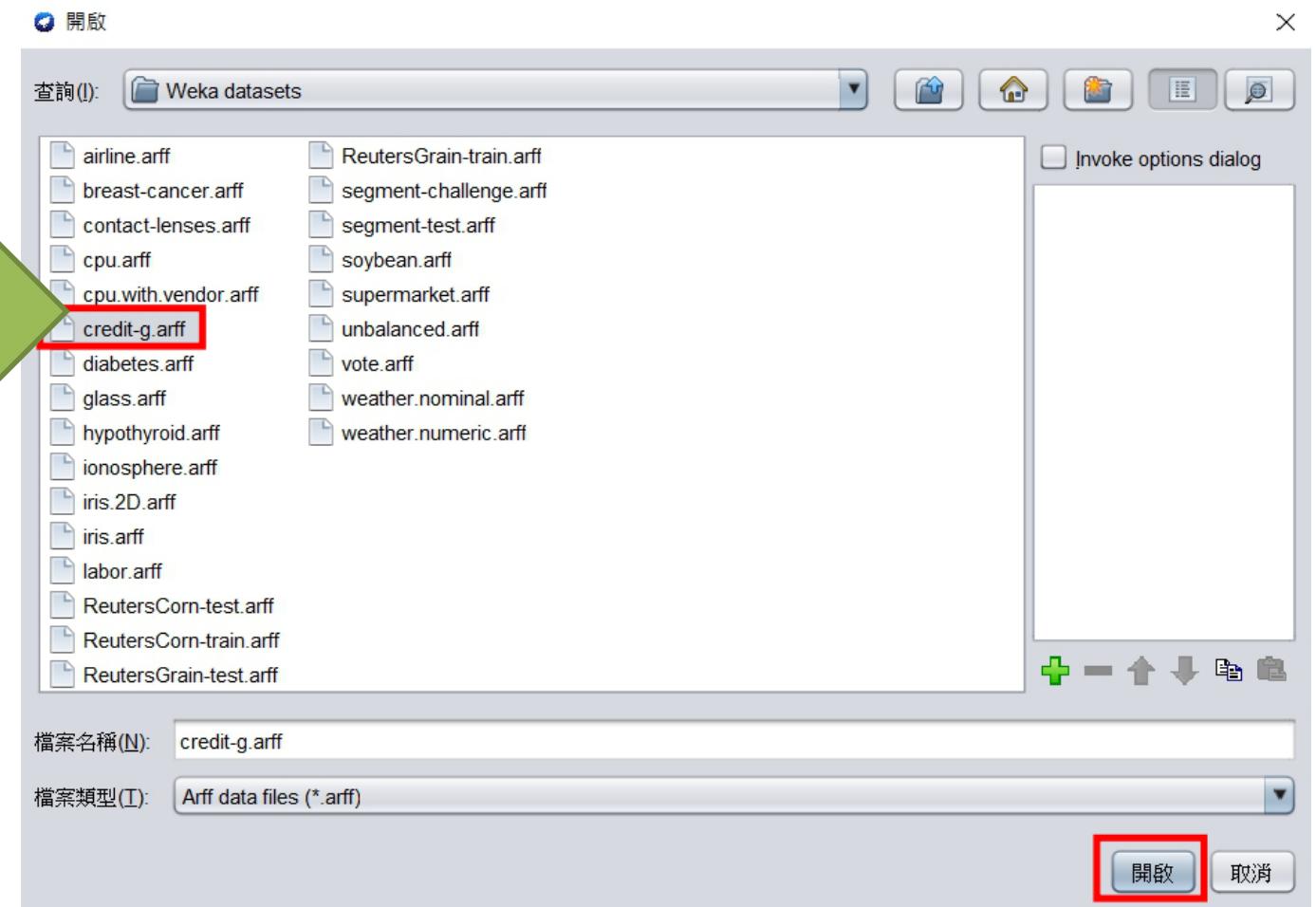
Lesson 4.5: 計算成本

1. 開啟Weka的Explorer。



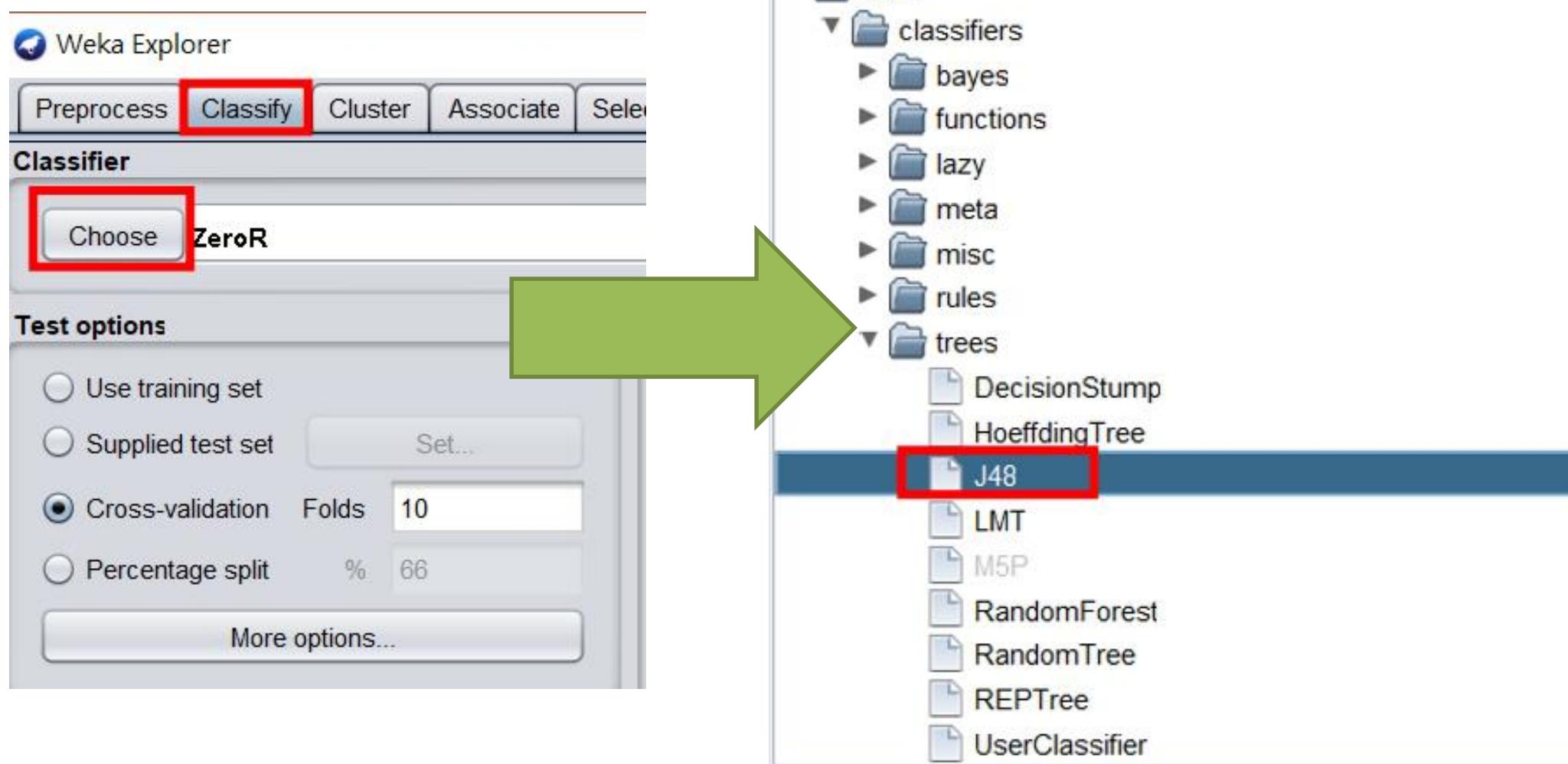
Lesson 4.5: 計算成本

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊**credit-g.arff**的檔案後，再以左鍵單擊下方「開啟」按鈕以載入此檔案



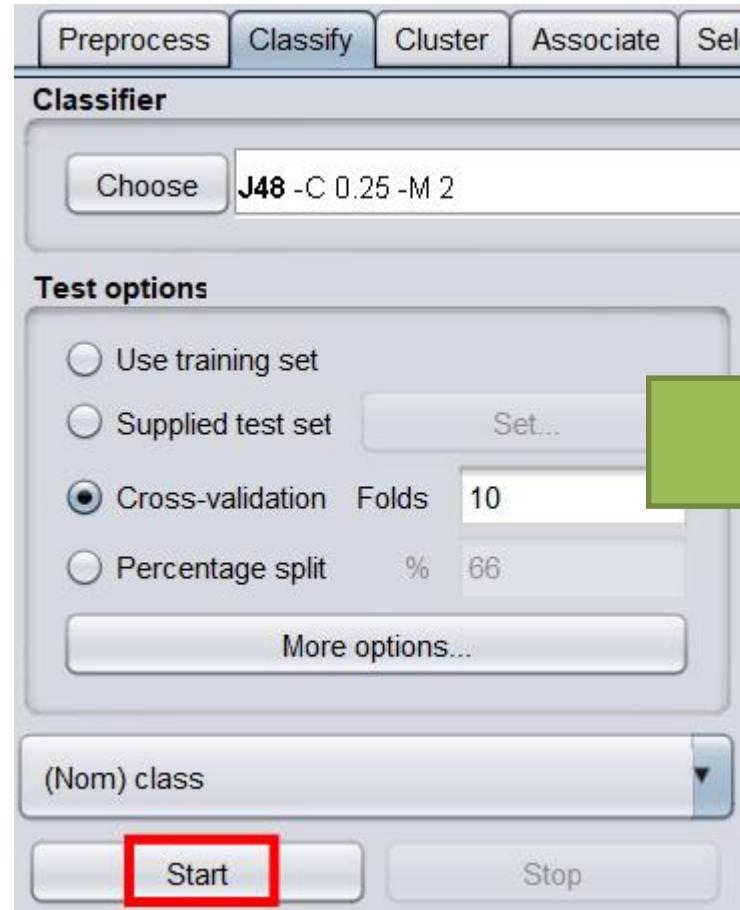
Lesson 4.5: 計算成本

3. 切換到Classify介面點選Choose鈕，在出現的選單中左鍵單擊trees資料夾下的J48



Lesson 4.5: 計算成本

4. 左鍵單擊Start按鈕，執行結果如右圖，得到70.5%準確率。



Classifier output

```
Time taken to build model: 0.52 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances          705      70.5 %
Incorrectly Classified Instances       295      29.5 %
Kappa statistic                         0.2467
Mean absolute error                     0.3467
Root mean squared error                 0.4796
Relative absolute error                  82.5233 %
Root relative squared error            104.6565 %
Total Number of Instances               1000

==== Detailed Accuracy By Class ====

           TP Rate   FP Rate   Precision   Recall   F-Measure   MCC   ROC Area   PRC Area
           0.840     0.610     0.763     0.840     0.799     0.251     0.639     0.746
           0.390     0.160     0.511     0.390     0.442     0.251     0.639     0.449
Weighted Avg.    0.705     0.475     0.687     0.705     0.692     0.251     0.639     0.657

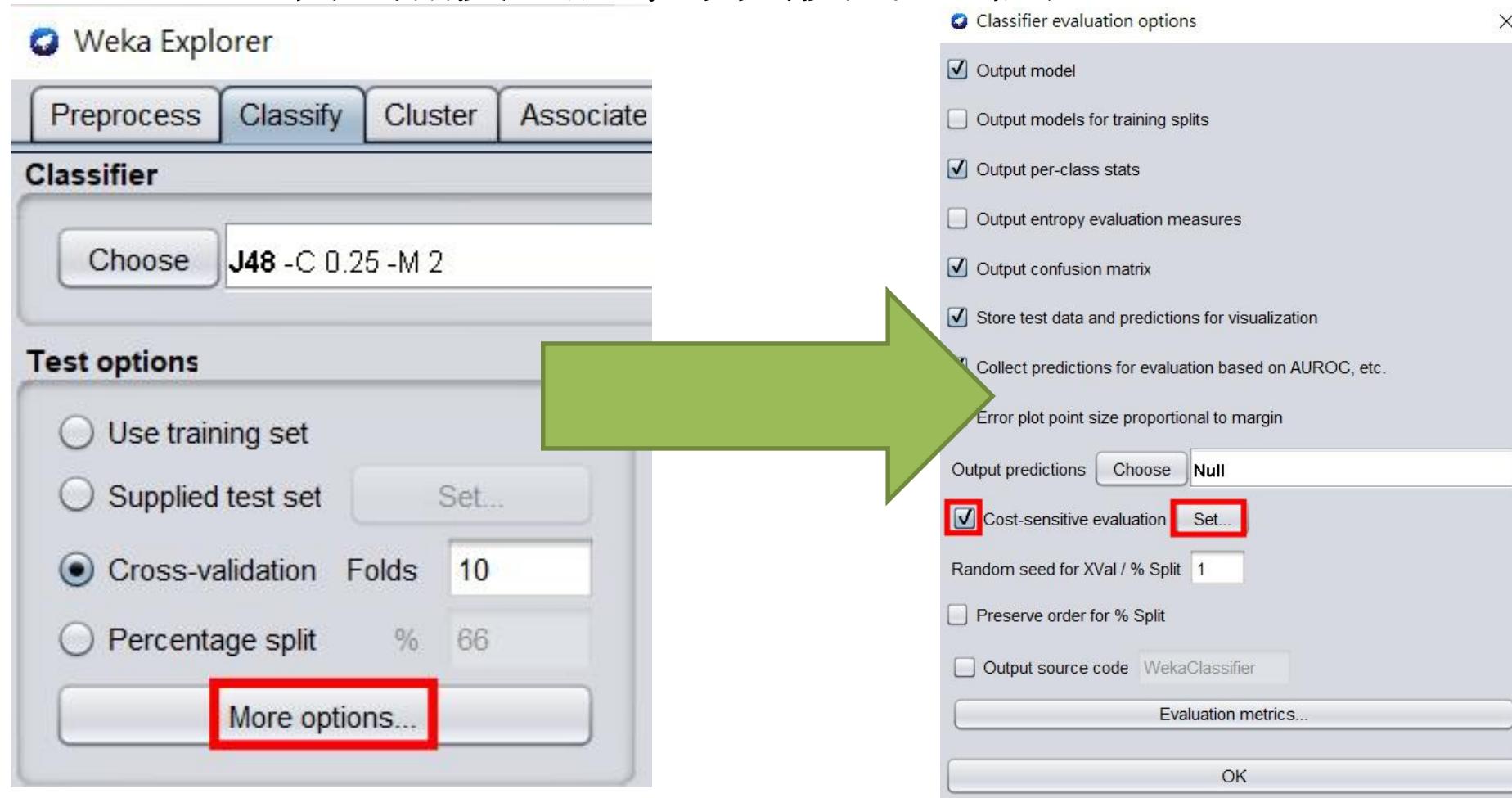
==== Confusion Matrix ====

  a   b   <-- classified as
588 112 |   a = good
183 117 |   b = bad
```

Lesson 4.5: 計算成本

接著，我們試著做成本敏感評估。

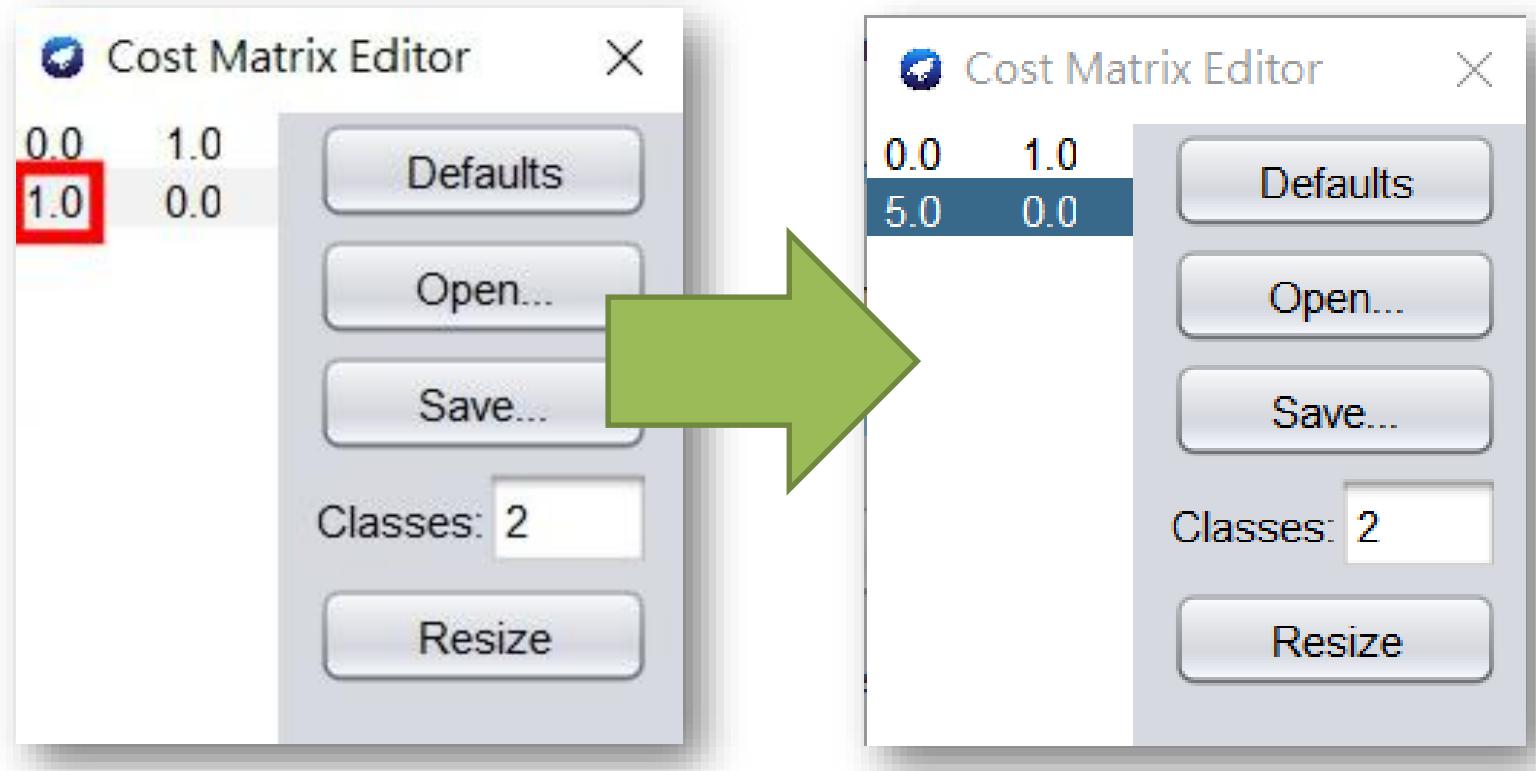
1. 左鍵單擊More options按鈕，並在彈出的視窗勾選Cost-sensitive evaluation選項。然後左鍵單擊其後的Set按鈕。



Lesson 4.5: 計算成本

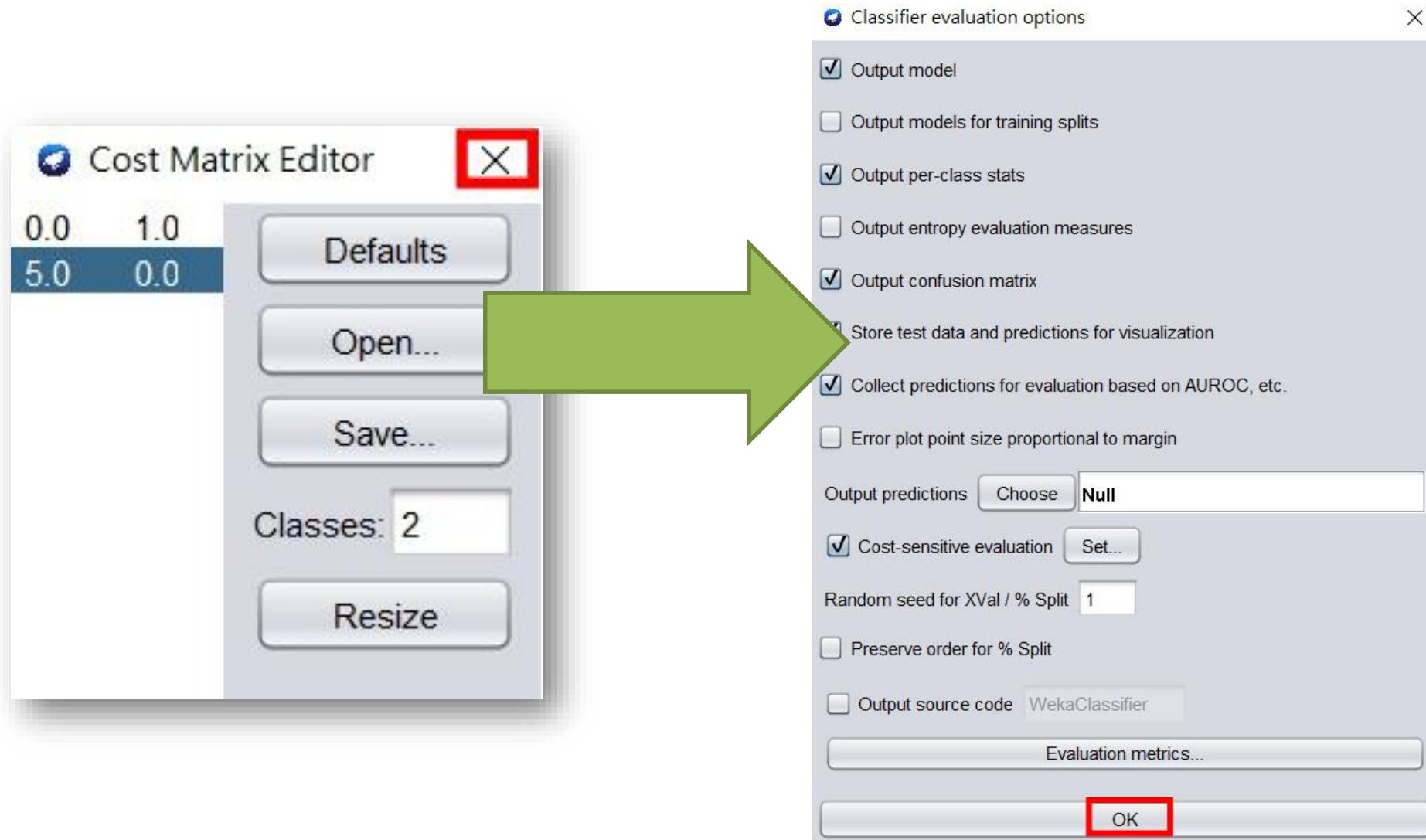
我們需要在彈出的視窗設置一個成本矩陣：一個2乘2的矩陣。並將誤差成本設定成5。

2. 左鍵雙擊左圖紅框處，輸入5後按下視窗內空白處或enter鍵確定此設置。



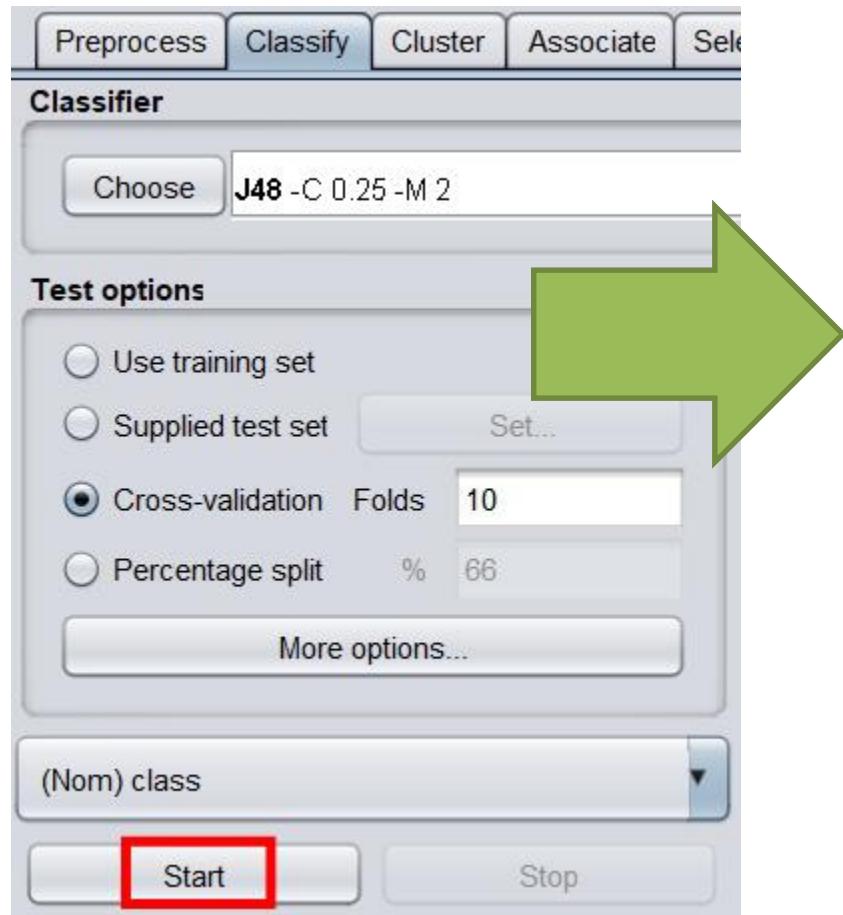
Lesson 4.5: 計算成本

3. 左鍵單擊視窗右上方關閉按鈕(左圖紅框處)，並在More option的視窗中左鍵單擊OK按鈕。



Lesson 4.5: 計算成本

4. 回到Classify面板，左鍵單擊Start按鈕，執行結果如右圖，得到一樣的混淆矩陣，但是多了「總成本」1027和「平均成本」1.027。

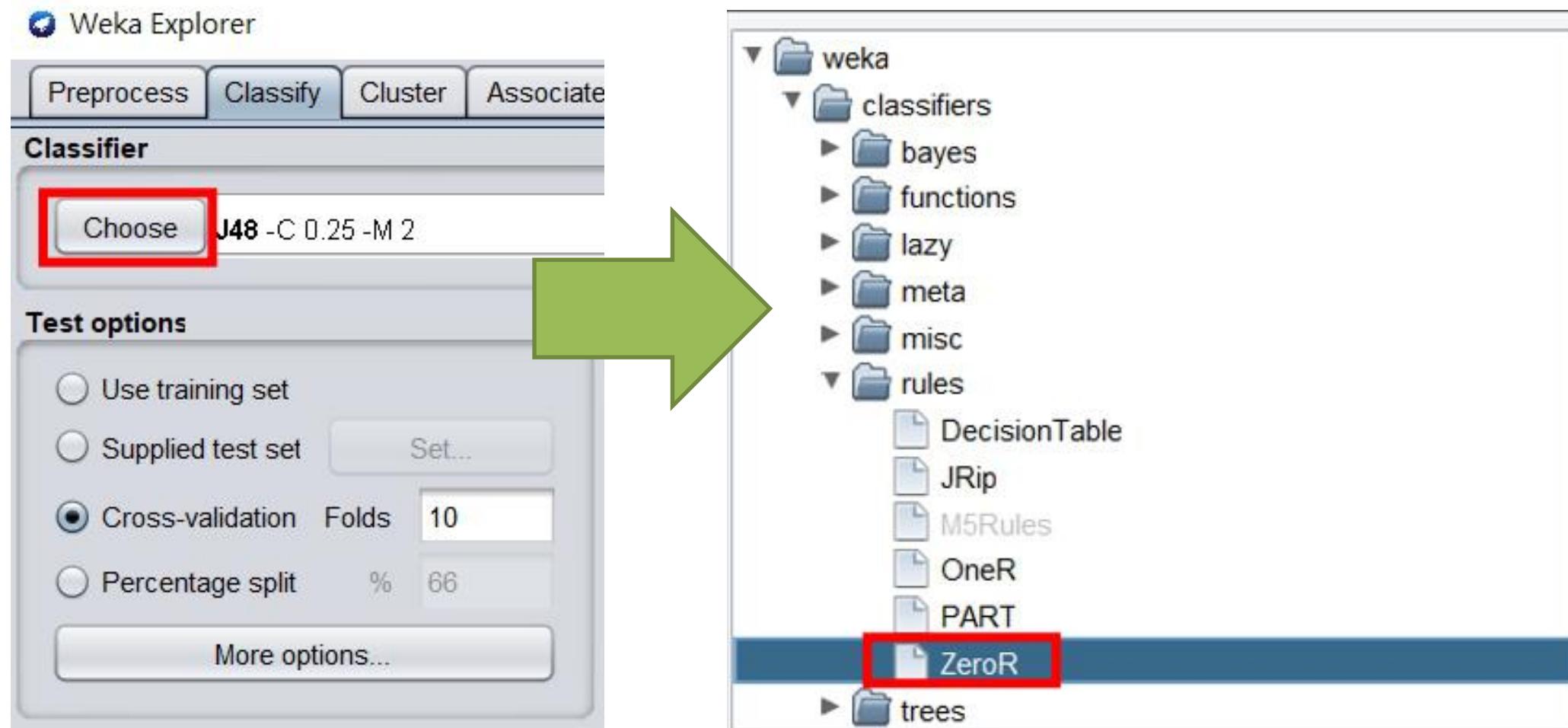


Classifier output								
==== Stratified cross-validation ====								
==== Summary ====								
Correctly Classified Instances	705	70.5	%					
Incorrectly Classified Instances	295	29.5	%					
Kappa statistic	0.2467							
Total Cost	1027							
Average Cost	1.027							
Mean absolute error	0.3467							
Root mean squared error	0.4796							
Relative absolute error	82.5233 %							
Root relative squared error	104.6565 %							
Total Number of Instances	1000							
==== Detailed Accuracy By Class ====								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Weighted Avg.	0.840	0.610	0.763	0.840	0.799	0.251	0.639	0.746
	0.390	0.160	0.511	0.390	0.442	0.251	0.639	0.449
	0.705	0.475	0.687	0.705	0.692	0.251	0.639	0.657
==== Confusion Matrix ====								
a	b	<-- classified as						
588	112		a = good					
183	117		b = bad					

Lesson 4.5: 計算成本

如果我們想要基線成本，則使用ZeroR。

1. 在Classify面板左鍵單擊Choose按鈕，並在出現的選單中左鍵單擊ZeroR分類器。



Lesson 4.5: 計算成本

2. 左鍵單擊Start按鈕，執行結果如右圖，得到另一個混淆矩陣且總成本為1500。

The image shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'ZeroR' is chosen. Under 'Test options', 'Cross-validation' is selected with 10 folds. A green arrow points from the 'Start' button in the bottom left of the Weka Explorer window to the 'Classifier output' window on the right.

Classifier output

```
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances      700           70    %
Incorrectly Classified Instances   300           30    %
Kappa statistic                   0
Total Cost                        1500
Average Cost                      1.5
Absolute error                    0.4202
Mean squared error                0.4583
Relative absolute error           100    %
Root relative squared error      100    %
Total Number of Instances         1000

==== Detailed Accuracy By Class ====

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area
          1.000    1.000    0.700     1.000   0.824     ?     0.500    0.700
          0.000    0.000    ?          0.000   ?        ?     0.500    0.300
Weighted Avg.                     0.700    0.700    ?          0.700   ?        ?     0.500    0.580

==== Confusion Matrix ====

  a   b   <- classified as
700  0 |   a = good
300  0 |   b = bad
```

Lesson 4.5: 計算成本

Weka: 成本敏感評估

- ❖ 信用資料集credit-g.arff
- ❖ J48 (70%)

a	b	<- - classified as
588	112	a = good
183	117	b = bad

成本: 295個分類不正確的實例數量
(183+112)

- ❖ Classify面板中的“More options”: Cost-sensitive evaluation

成本矩陣:

$$\begin{matrix} 0 & 1 \\ 5 & 0 \end{matrix}$$

成本: $183 \times 5 + 112 \times 1$
 $= 1027$ (1.027/instance)

- ❖ 基線 (ZeroR)

a	b	<- - classified as
700	0	a = good
300	0	b = bad

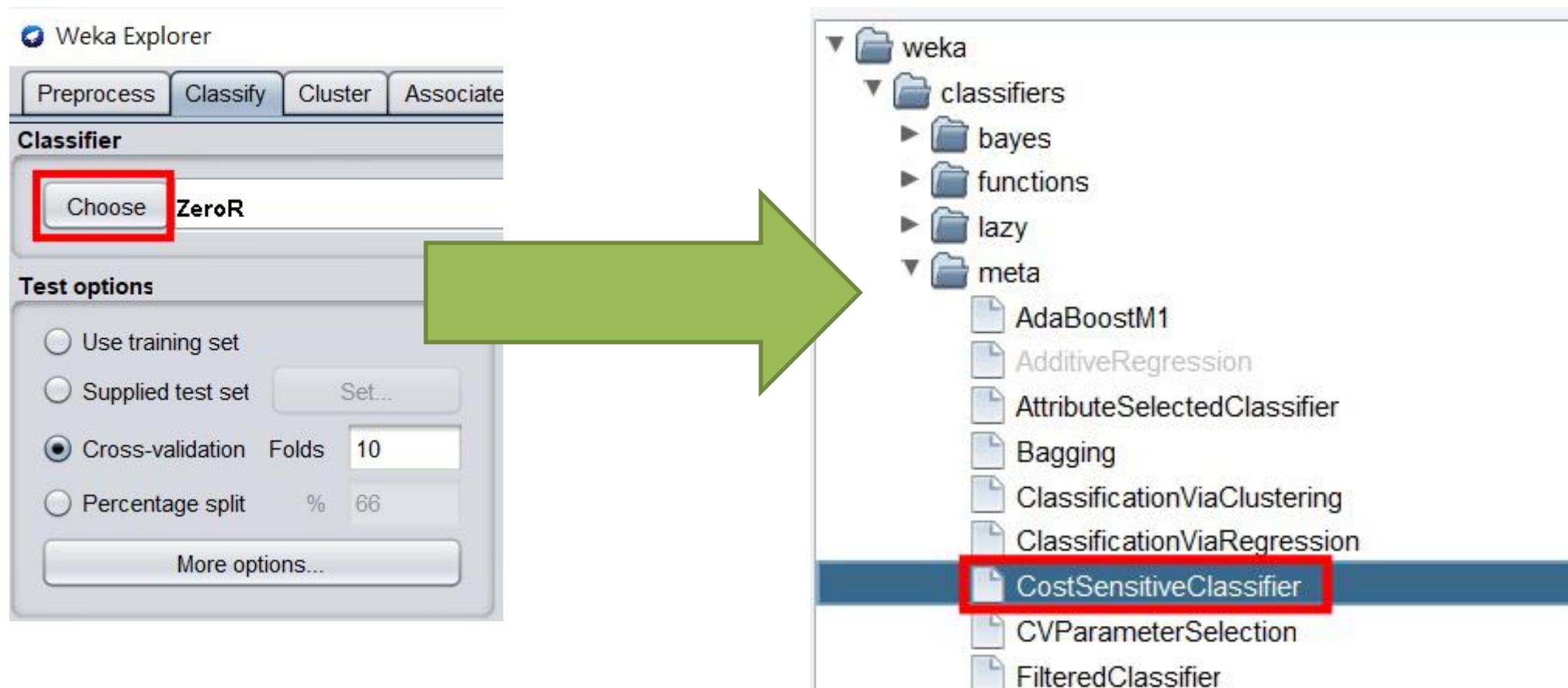
成本: $300 \times 5 = 1500$

- ❖ 如果我把所有客戶都歸為「壞」，總成本將是700

Lesson 4.5: 計算成本

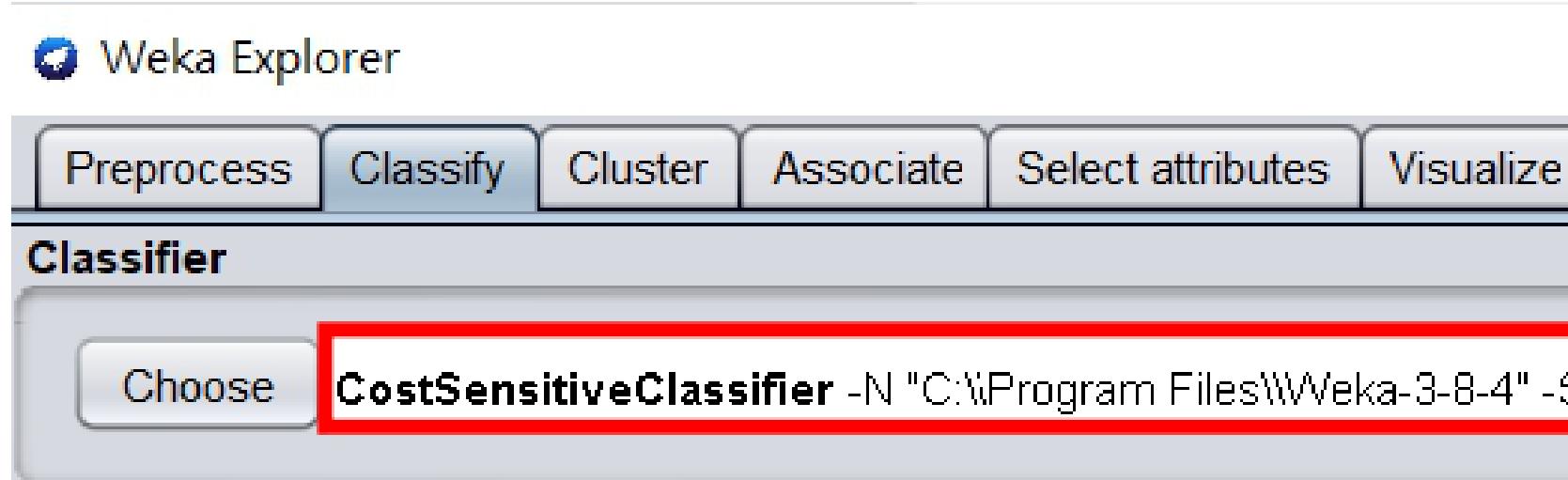
接著我們試著使用CostSensitiveClassifier分類器。

1.回到Classify界面點選Choose鈕，在出現的選單中左鍵單擊meta資料夾下的CostSensitiveClassifier。



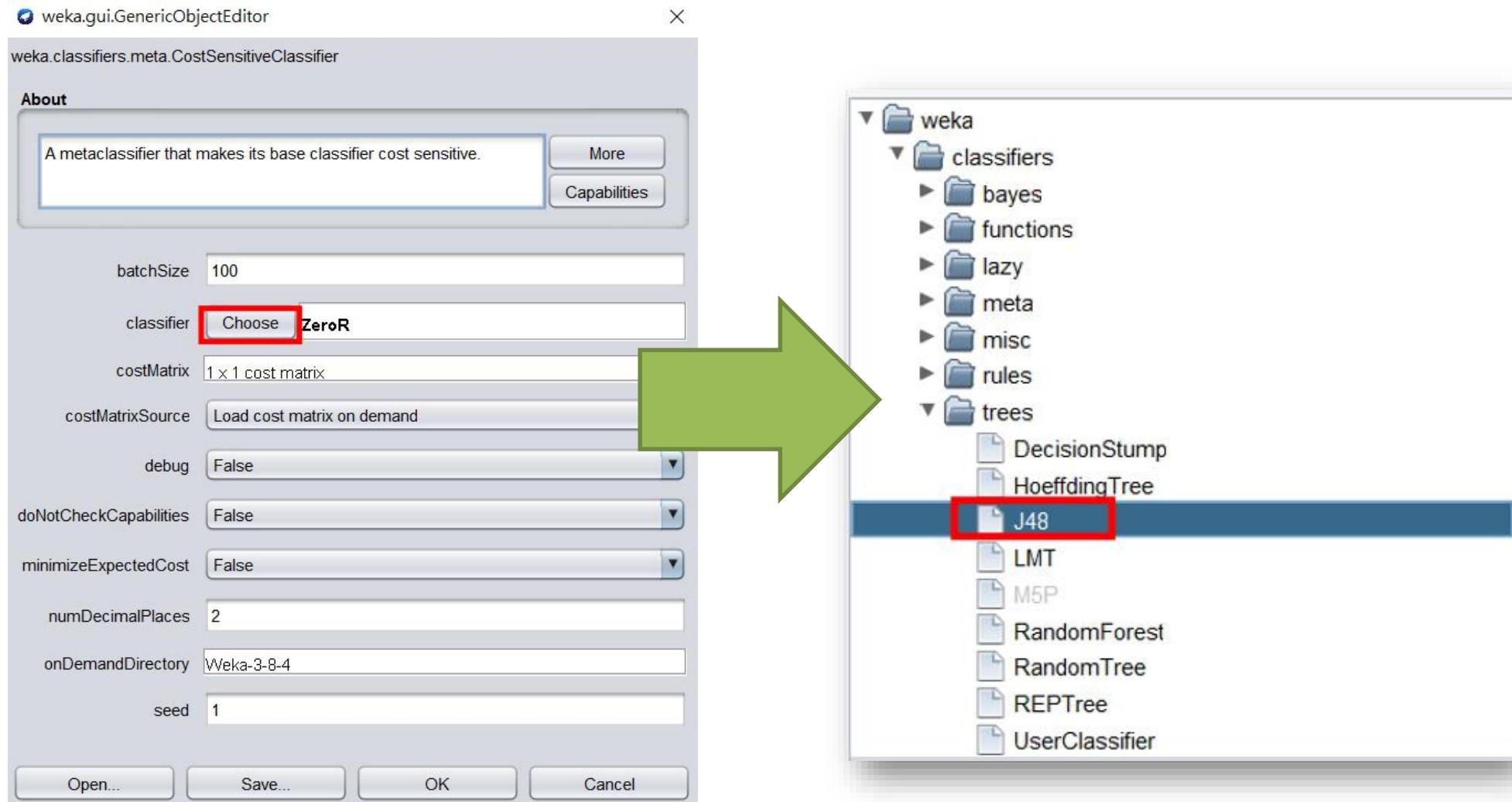
Lesson 4.5: 計算成本

2. 左鍵單擊分類器名稱(左圖紅框處)，開啟配置視窗。



Lesson 4.5: 計算成本

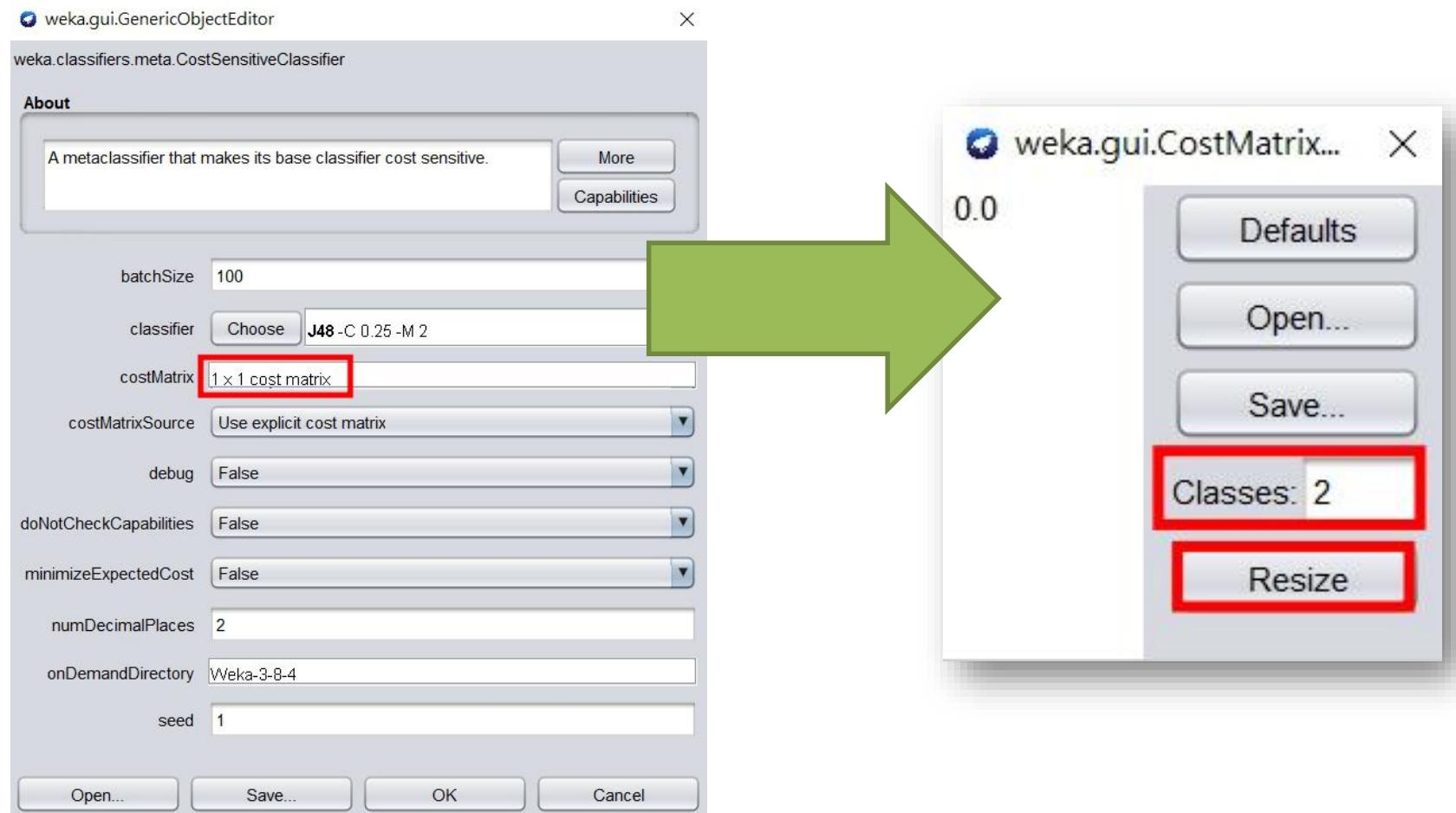
3. 在配置視窗中左鍵單擊Choose按鈕(左圖紅框處)，並在彈出的選單中以左鍵單擊J48分類器。



Lesson 4.5: 計算成本

我們需要設定我們的成本矩陣: 一個2乘2的矩陣。

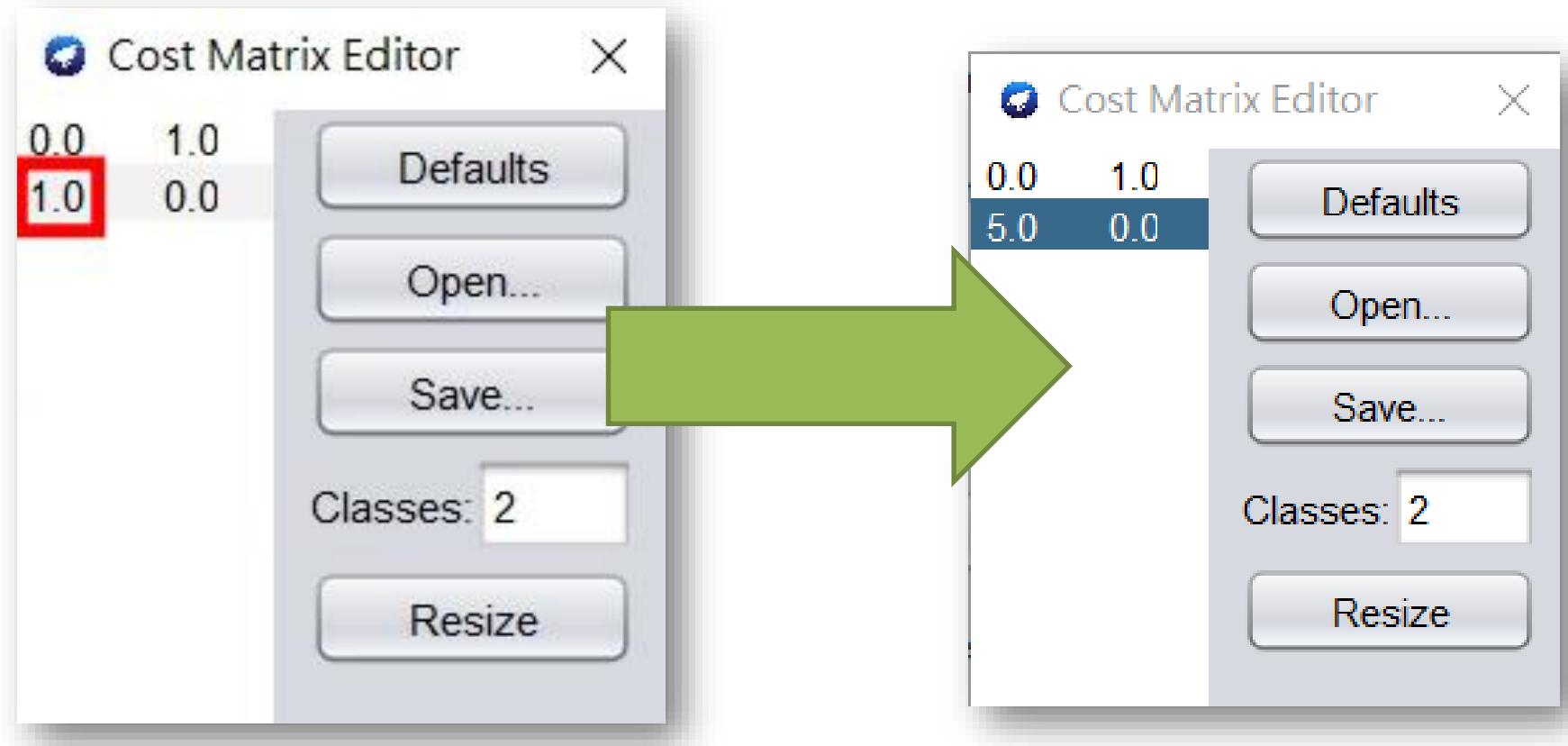
4. 左鍵單擊costMatrix的參數(左圖紅框處)開啟右圖視窗，並在Classes後方的輸入框中輸入2，接著左鍵單擊Resize按鈕。



Lesson 4.5: 計算成本

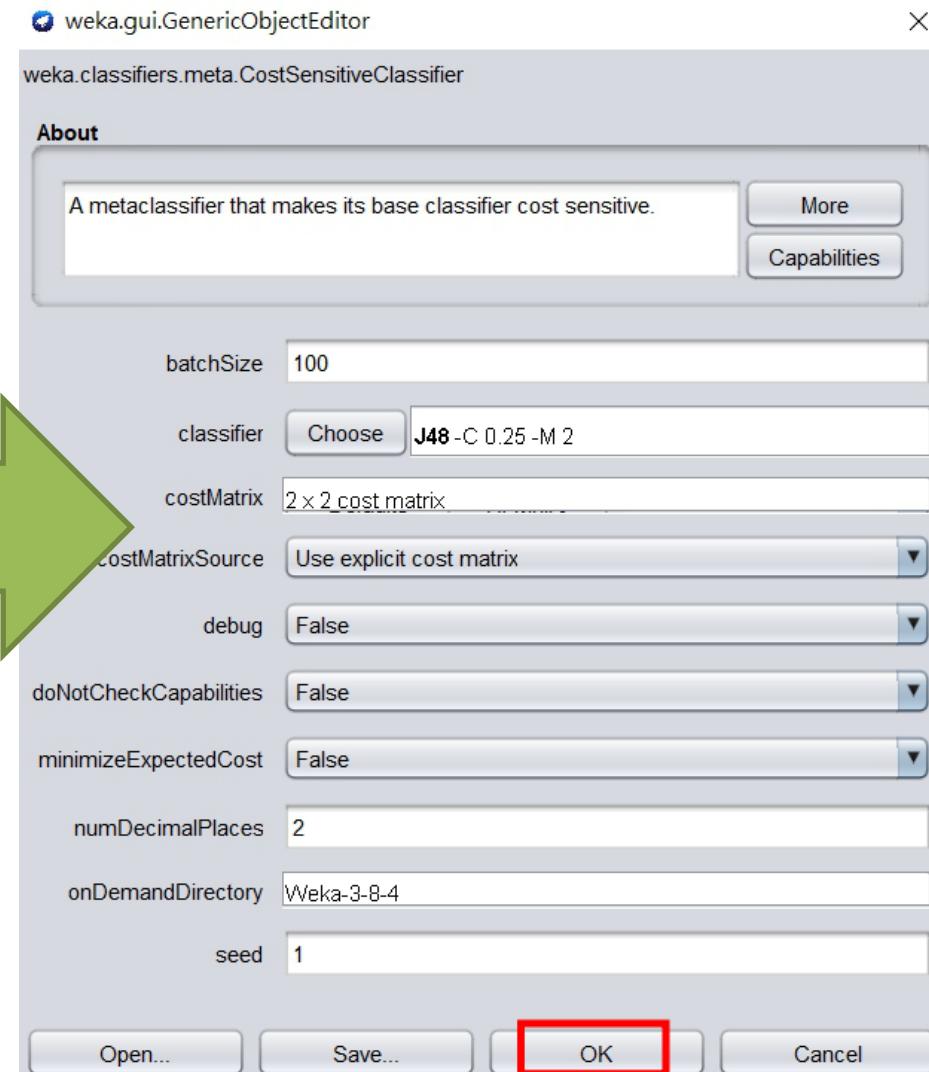
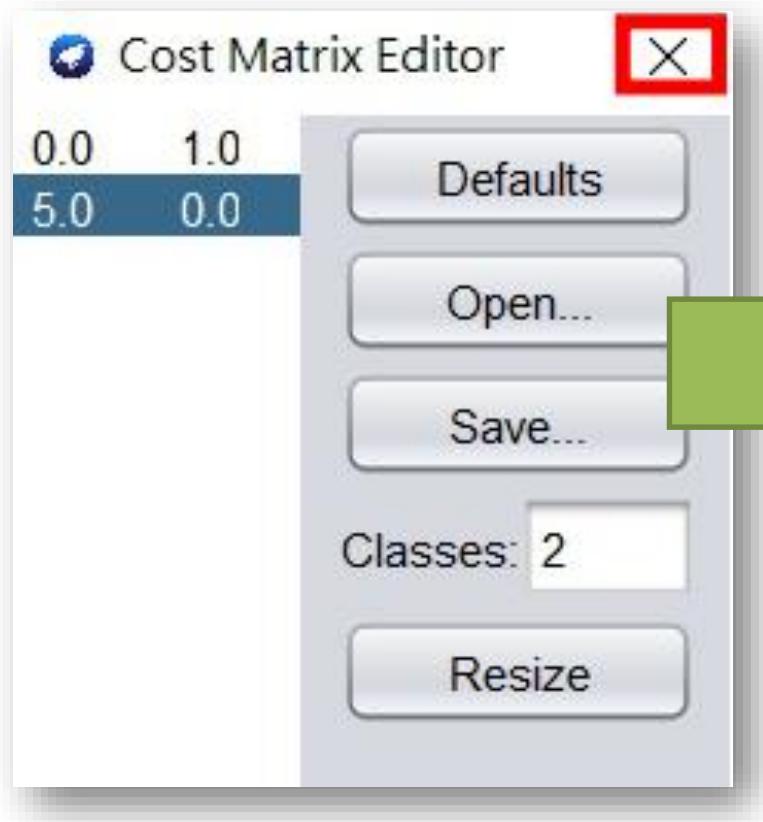
接著，將誤差成本設定成5。

5. 左鍵雙擊左圖紅框處，輸入5後按下視窗內空白處或enter鍵確定此設置。



Lesson 4.5: 計算成本

6. 左鍵單擊視窗右上方關閉按鈕(左圖紅框處)，並在分類器配置視窗中左鍵單擊OK按鈕。



Lesson 4.5: 計算成本

7.回到Classify面板後，左鍵單擊Start按鈕，執行結果如右圖：僅僅得到60.6%的正確率，但是成本只有658。另外還得到一個不同的混淆矩陣。

The screenshot shows the Weka Classifier interface. On the left, under 'Test options', 'Cross-validation' is selected with 'Folds' set to 10. A large green arrow points from the classifier settings to the 'Classifier output' window on the right. The 'Classifier output' window displays the following information:

Classifier output

```
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances      606      60.6 %
Incorrectly Classified Instances   394      39.4 %
Kappa statistic                   0.2492
Total Cost                        658
Average Cost                      0.658
Mean absolute error                0.397
Root mean squared error           0.5455
Relative absolute error            94.4783 %
Root relative squared error       119.0321 %
Total Number of Instances         1000

==== Detailed Accuracy By Class ====

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area
          0.531    0.220    0.849     0.531    0.654     0.288   0.655    0.800
          0.780    0.469    0.416     0.780    0.543     0.288   0.655    0.384
Weighted Avg.        0.606    0.295    0.719     0.606    0.621     0.288   0.655    0.675

==== Confusion Matrix ====

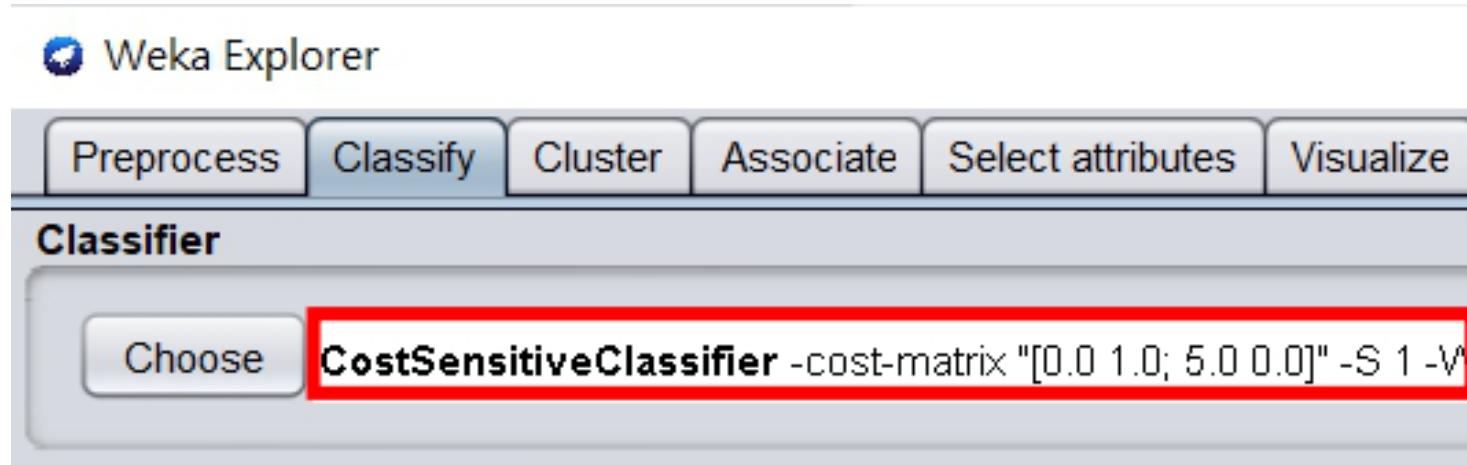
  a   b  <- classified as
372 328 |  a = good
 66 234 |  b = bad
```

The 'Total Cost' value of 658 and the 'Confusion Matrix' section are highlighted with red boxes.

Lesson 4.5: 計算成本

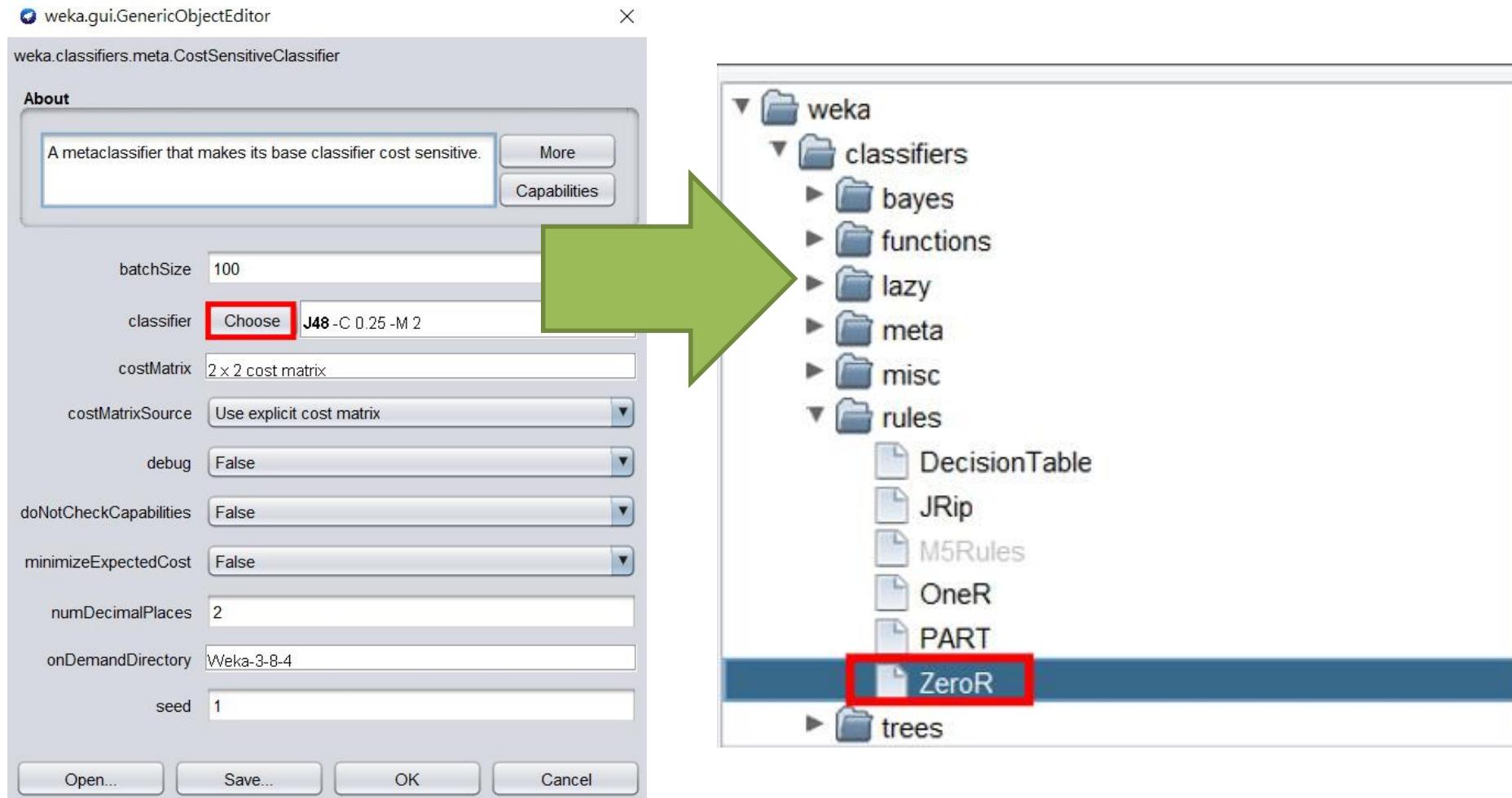
接著，我們看ZeroR在CostSensitiveClassifier裡的表現。

1. 左鍵單擊分類器名稱(左圖紅框處)，開啟配置視窗。



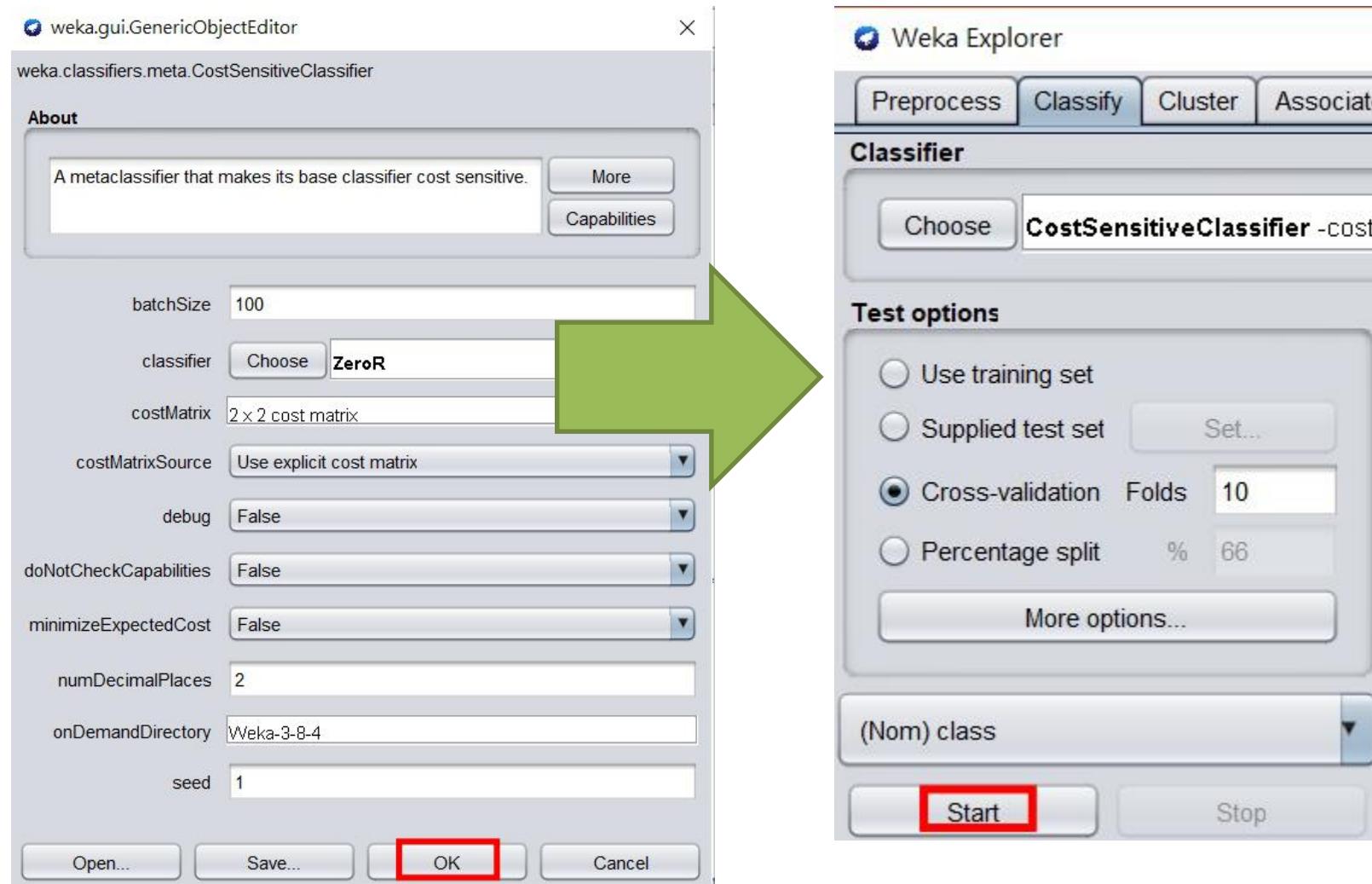
Lesson 4.5: 計算成本

2. 在配置視窗中左鍵單擊Choose按鈕(左圖紅框處)，並在彈出的選單中以左鍵單擊ZeroR分類器。



Lesson 4.5: 計算成本

3. 在分類器配置視窗中左鍵單擊OK按鈕回到Classify面板，左鍵單擊Start按鈕。



Lesson 4.5: 計算成本

執行結果如下圖：與上次直接執行ZeroR時將所有實例都歸為「好」不同，所有的客戶都變成了「壞」客戶。總成本只要700。

```
Classifier output

==== Stratified cross-validation ====
==== Summary ====

    Correctly Classified Instances      300
    Incorrectly Classified Instances   700
    Kappa statistic                   0
    Total Cost                       700
    Average Cost                     0.7
    Mean absolute error              0.5726
    Root mean squared error          0.5962
    Relative absolute error          136.2677 %
    Root relative squared error     130.1057 %
    Total Number of Instances        1000

==== Detailed Accuracy By Class ====

    TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
    0.000    0.000    ?         0.000    ?         ?       0.500    0.700
    1.000    1.000    0.300    1.000    0.462    ?
    Weighted Avg. 0.300    0.300    ?         0.300    ?         ?       0.500    0.580

==== Confusion Matrix ====

    a   b   <-- classified as
    0 700 |   a = good
    0 300 |   b = bad
```

Lesson 4.5: 計算成本

Weka: 成本敏感(cost-sensitive)分類器

- ❖ 此分類器在學習的時候應該知道成本!
- ❖ meta > CostSensitiveClassifier
- ❖ 選擇 J48
- ❖ 定義成本矩陣:

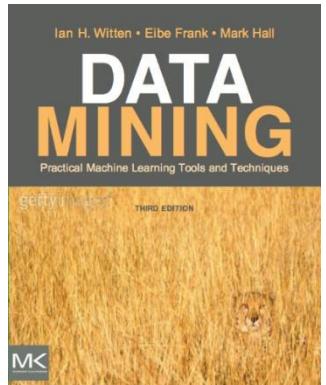
0	1
5	0
- ❖ 更差的分類誤差 (61% vs. 70%)
- ❖ 更低的平均成本(0.66 vs. 1.027)
- ❖ 在混淆矩陣中誤差的影響
- ❖ ZeroR: 平均成本0.7

		舊的		新的	
a	b	a = good	b = bad	a	b
588	112		a = good	372	328
183	117		b = bad	66	234

Lesson 4.5: 計算成本

- ❖ 分類正確率是最好的衡量標準嗎？顯然不是。
- ❖ 經濟的模型: 誤差成本
 - 在不同的誤差成本之間權衡 – ROC曲線可以幫助你
- ❖ Cost-sensitive 評估
- ❖ Cost-sensitive 分類
- ❖ **meta > CostSensitiveClassifier**
 - 使得分類器變成成本敏感分類器

- ❖ Section 5.7 *Counting the cost*





THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Class 4 – Lesson 6

成本敏感分類 vs. 成本敏感學習

Cost-sensitive classification vs. cost-sensitive learning

Ian H. Witten

Department of Computer Science University of
Waikato
New Zealand

Lesson 4.6: 成本敏感分類vs.成本敏感學習

Class 1 探索Weka的介面；處理大數據

Class 2 離散以及文本分類

Class 3 分類規則、關聯規則、聚類

Class 4 選擇屬性以及計算成本

Class 5 神經網路，學習曲線和表現優化

Lesson 4.1 「包裝器」屬性選擇法

Lesson 4.2 屬性選擇分類器

Lesson 4.3 方案獨立選擇法

Lesson 4.4 使用分等進行屬性選擇

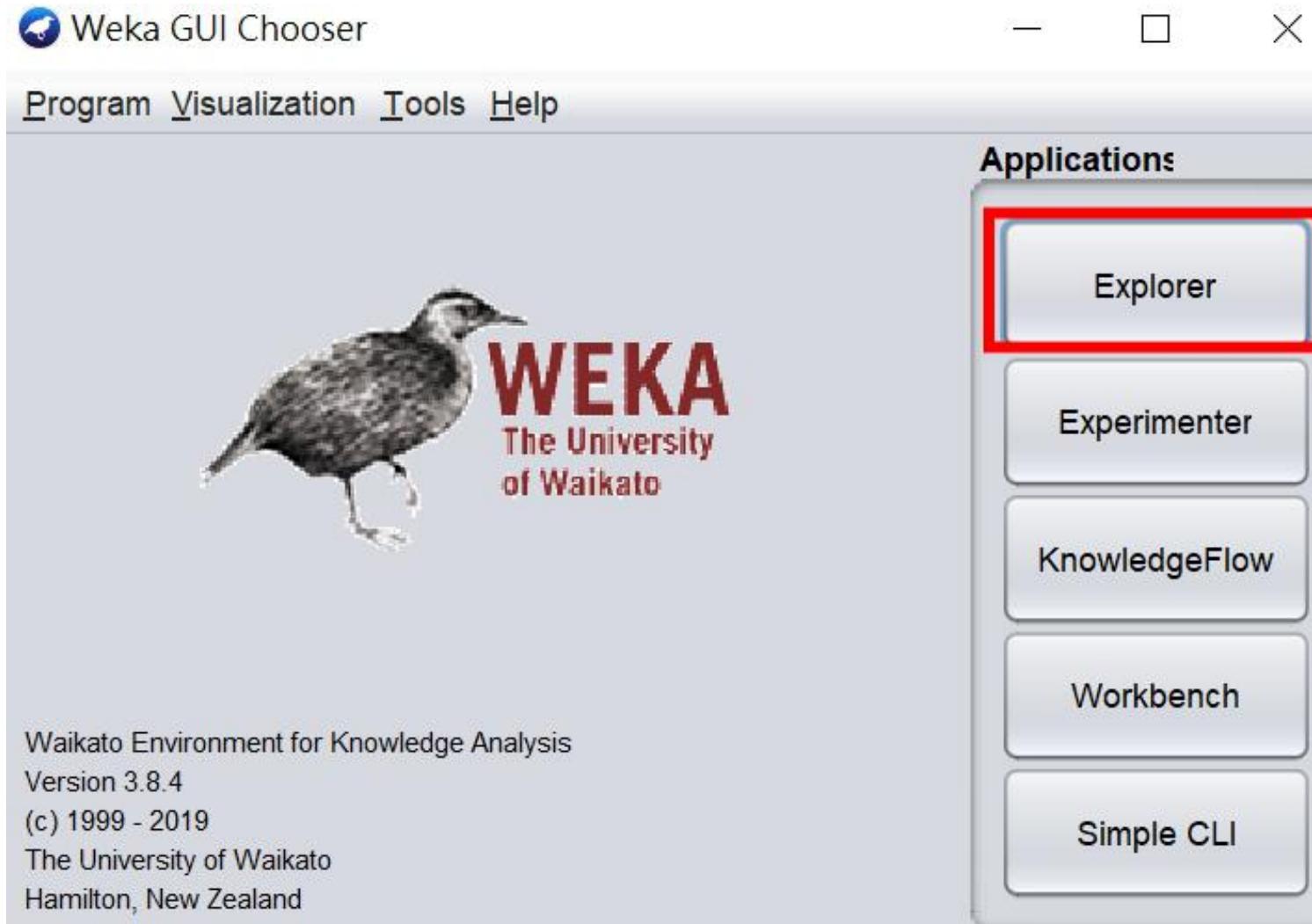
Lesson 4.5 計算成本

Lesson 4.6 成本敏感分類



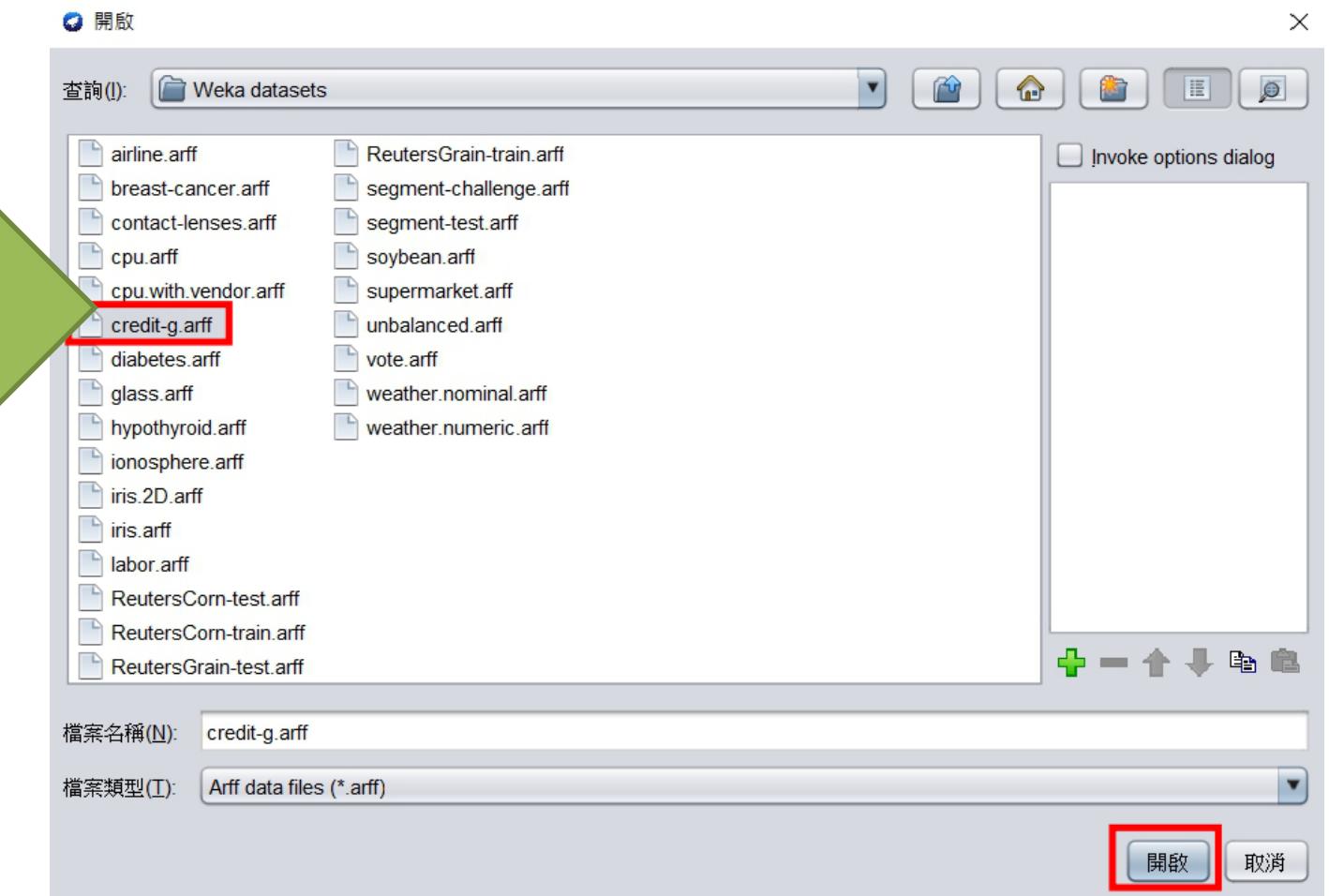
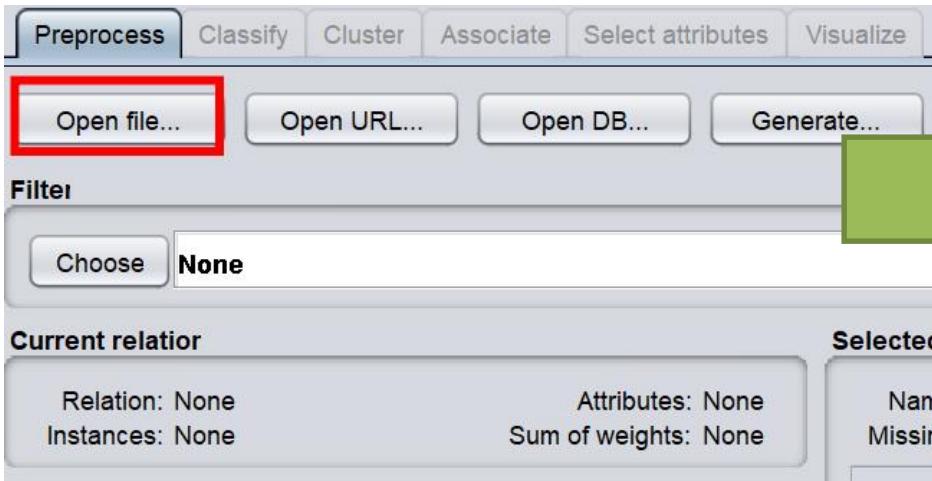
Lesson 4.6: 成本敏感分類vs.成本敏感學習

1. 開啟Weka的Explorer。



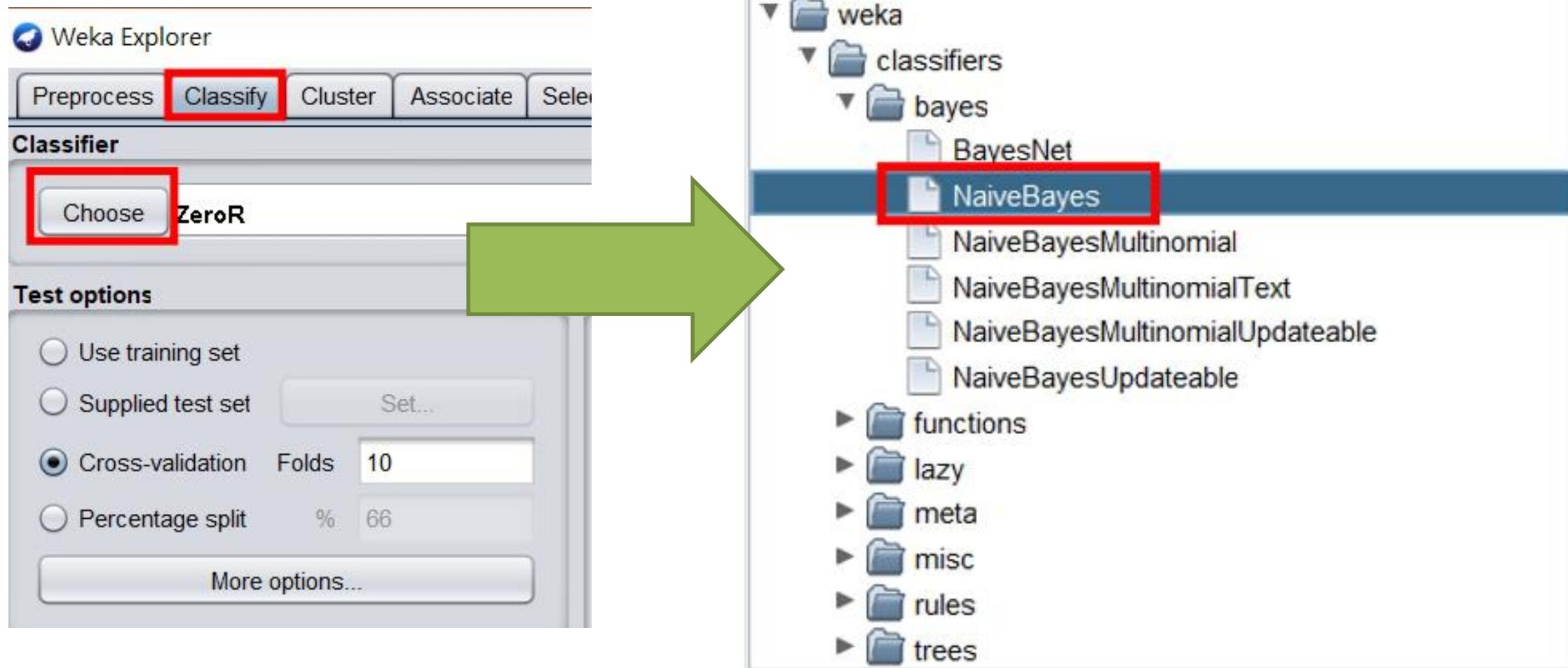
Lesson 4.6: 成本敏感分類vs.成本敏感學習

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊**credit-g.arff**的檔案後，再以左鍵單擊下方「開啟」按鈕以載入此檔案



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

3. 切換到Classify面板點選Choose鈕，在出現的選單中左鍵單擊bayes資料夾下的NaiveBayes分類器



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

4. 左鍵單擊Start按鈕，執行結果如右圖，得到75.4%分類準確率。

The image shows the Weka Explorer interface with the Classifier tab selected. Under 'Classifier', 'NaiveBayes' is chosen. In the 'Test options' section, 'Cross-validation' is selected with 'Folds' set to 10. A large green arrow points from the 'Start' button in the bottom left of the Weka Explorer to the 'Classifier output' window on the right. The 'Classifier output' window displays the following text:

```
Time taken to build model: 0.06 seconds

==== Stratified cross-validation ====
==== Summary ===

Correctly Classified Instances           754
Incorrectly Classified Instances        246
Kappa statistic                         0.3813
    an absolute error                  0.2936
    mean squared error                 0.4201
    true absolute error                69.8801 %
    relative squared error             91.6718 %
Number of Instances                     1000

==== Detailed Accuracy By Class ====

          TP Rate   FP Rate   Precision   Recall   F-Measure   MCC   ROC Area   PRC Area
          0.864     0.503     0.800      0.864     0.831     0.385     0.787     0.891
          0.497     0.136     0.611      0.497     0.548     0.385     0.787     0.577
Weighted Avg.       0.754     0.393     0.743      0.754     0.746     0.385     0.787     0.797

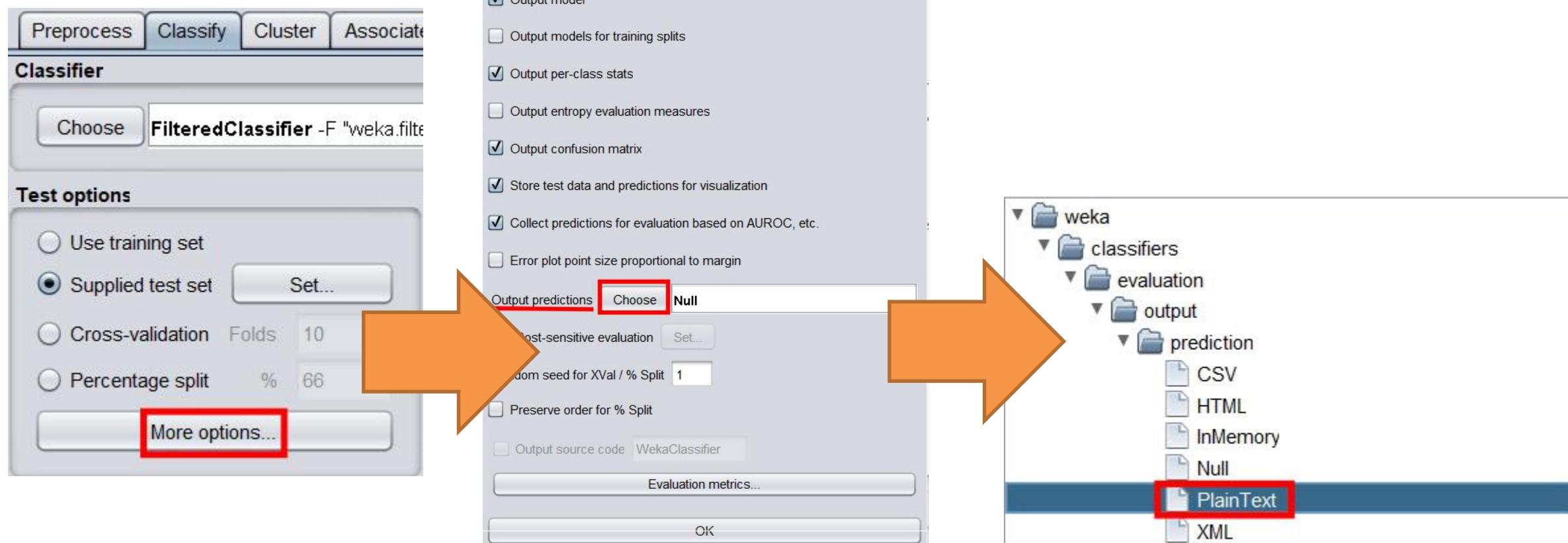
==== Confusion Matrix ===

    a     b     <-- classified as
605   95 |   a = good
151  149 |   b = bad
```

Lesson 4.6: 成本敏感分類vs. 成本敏感學習

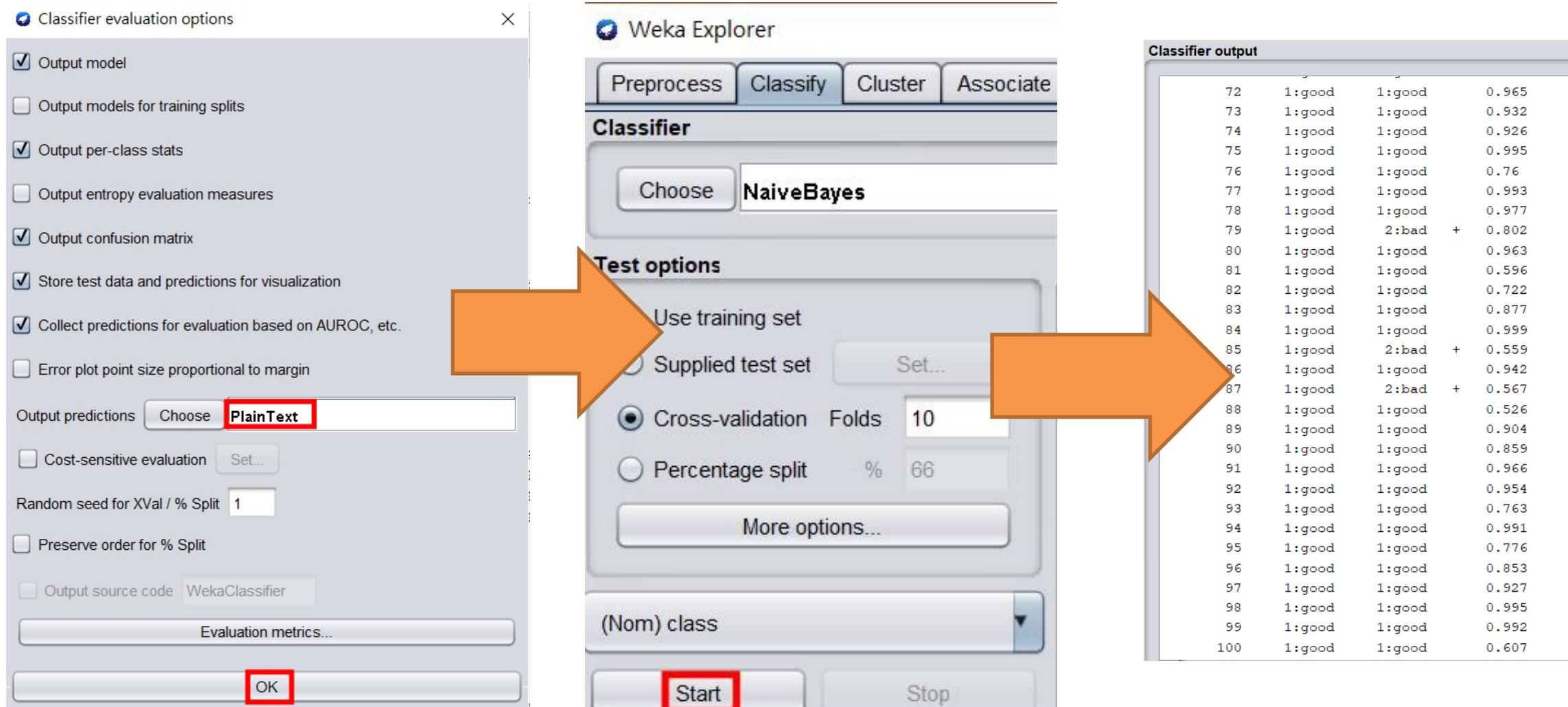
我們試著輸出預測結果。

5. 左鍵單擊Classify面板中Test option區域內的More options...按鈕，再以左鍵單擊Output predictions右方的Choose按鈕，於出現的選單中左鍵單擊PlainText。



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

6.確定選擇好PlainText後，左鍵單擊OK按鈕回到Classify面板，左鍵單擊Start按鈕。



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

使得分類器對成本敏感之方法1: 成本敏感分類

通過重新計算概率閾值(Threshold)
來調整分類器的輸出

- ❖ 信用資料集credit-g.arff
- ❖ NaiveBayes, Output predictions

```
a    b    <- classified as  
605  95 |  a = good  
151 149 |  b = bad
```

- ❖ 閾值: 0.5
 - 預測756個實例為`good`，其中有151個分類錯誤
 - 244個實例為`bad`，其中有95個分類錯誤

	actual	predicted	p _{good}
0	good	good	0.999
50	good	good	0.991
100	good	good	0.983
150	good	good	0.975
200	good	good	0.965
250	bad	good	0.951
300	bad	good	0.934
350	good	good	0.917
400	good	good	0.896
450	good	good	0.873
500	good	good	0.836
550	good	good	0.776
600	bad	good	0.715
650	good	good	0.663
700	good	good	0.587
750	bad	good	0.508
800	good	bad	0.416
850	bad	bad	0.297
900	good	bad	0.184
950	bad	bad	0.04

Lesson 4.6: 成本敏感分類vs. 成本敏感學習

重新計算概率閾值 - Naive Bayes

- ❖ 成本矩陣

	a	b	
0	1	a = good	
5	0	b = bad	

- ❖ 閾值 = $5/6 = 0.833$

	a	b	<- classified as
448	252	a = good	
53	247	b = bad	

總成本517 (vs. 850)

- ❖ 一般的成本矩陣: $\begin{matrix} 0 & \lambda \\ \mu & 0 \end{matrix}$

- ❖ 為了最小化把實例歸為 **good** 所帶來的成本:

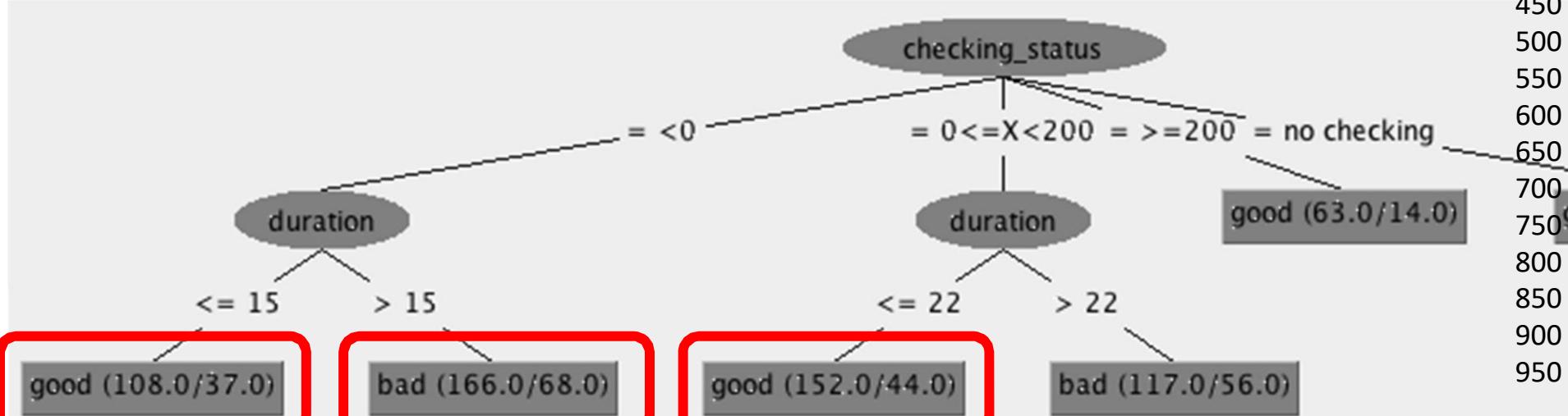
$$p_{\text{good}} > \frac{\mu}{\lambda + \mu}$$

actual	predicted	p_{good}
0	good	0.999
50	good	0.991
100	good	0.983
150	good	0.975
200	good	0.965
250	bad	0.951
300	bad	0.934
350	good	0.917
400	good	0.896
450	good	0.873
500	good	0.836
550	good	0.776
600	bad	0.715
650	good	0.663
700	good	0.587
750	bad	0.508
800	good	0.416
850	bad	0.297
900	good	0.184
950	bad	0.04

Lesson 4.6: 成本敏感分類vs. 成本敏感學習

對於那些不產生概率的方法應該怎麼做呢？

- ❖ 它們幾乎都會產生概率
- ❖ 使用 `minNumObj = 100` 的 J48 (為了取得最小樹)
- ❖ 從此樹看來,
 $1 - 37/108 = 0.657, 68/166=0.410, 1 - 44/152 = 0.711, \text{etc}$
- ❖ 其他方法(如rules)都是相似的

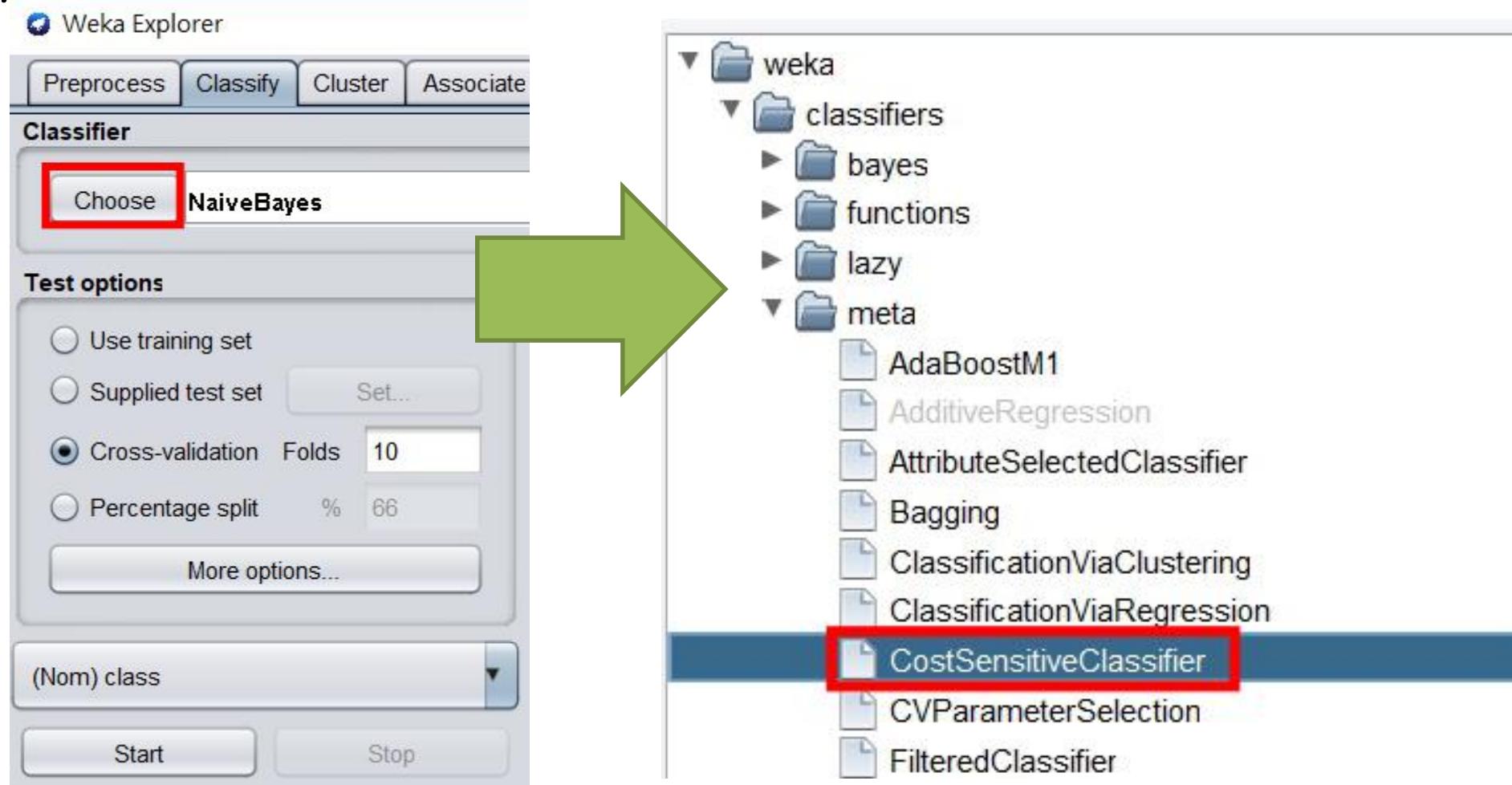


actual	predicted	p _{good}
0	good	0.883
50	good	0.883
100	good	0.883
150	good	0.883
200	good	0.883
250	good	0.883
300	good	0.883
350	good	0.883
400	good	0.778
450	bad	0.778
500	bad	0.711
550	good	0.711
600	good	0.711
650	good	0.657
700	bad	0.657
750	good	0.479
800	good	0.479
850	bad	0.410
900	good	0.410
950	bad	0.410

Lesson 4.6: 成本敏感分類vs.成本敏感學習

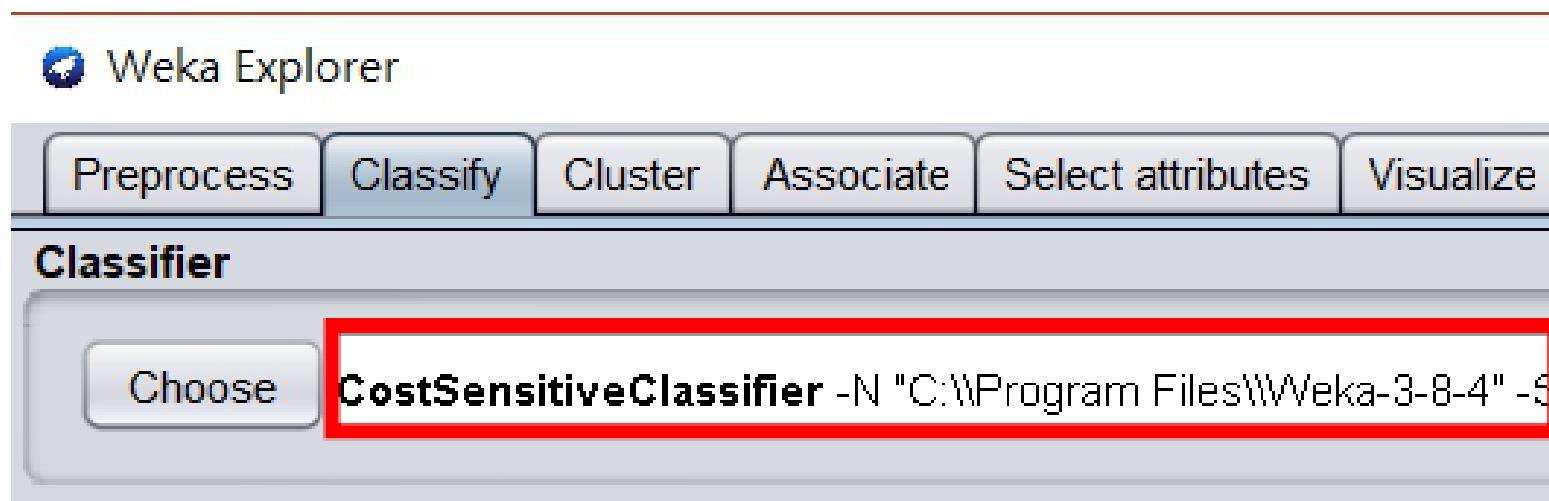
我們試著用CostSensitiveClassifier，並設定為最小化成本期望。

1.回到Classify界面點選Choose鈕，在出現的選單中左鍵單擊meta資料夾下的CostSensitiveClassifier。



Lesson 4.6: 成本敏感分類vs.成本敏感學習

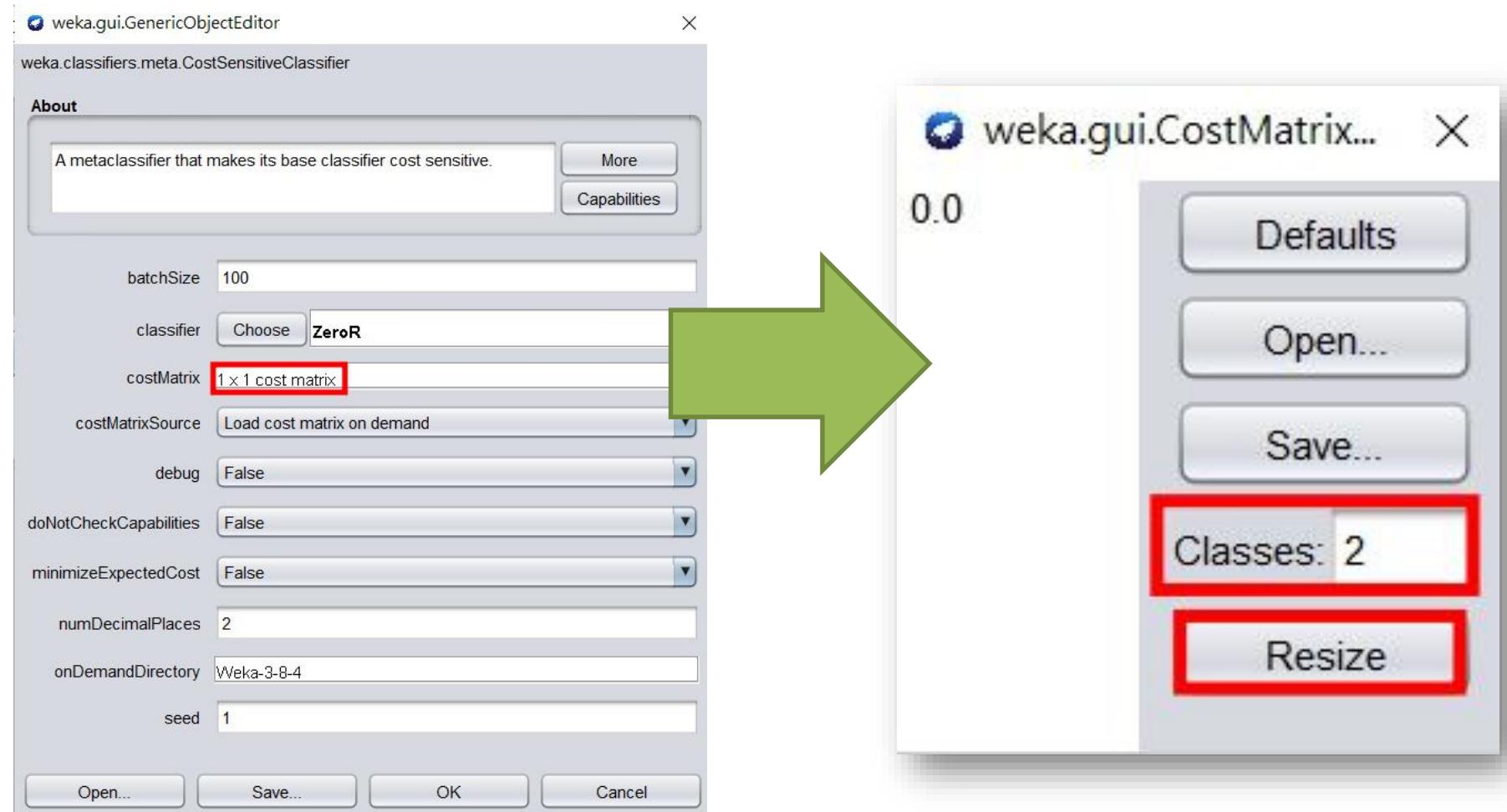
2. 左鍵單擊分類器名稱(左圖紅框處)，開啟配置視窗。



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

我們需要設定我們的成本矩陣：一個2乘2的矩陣。

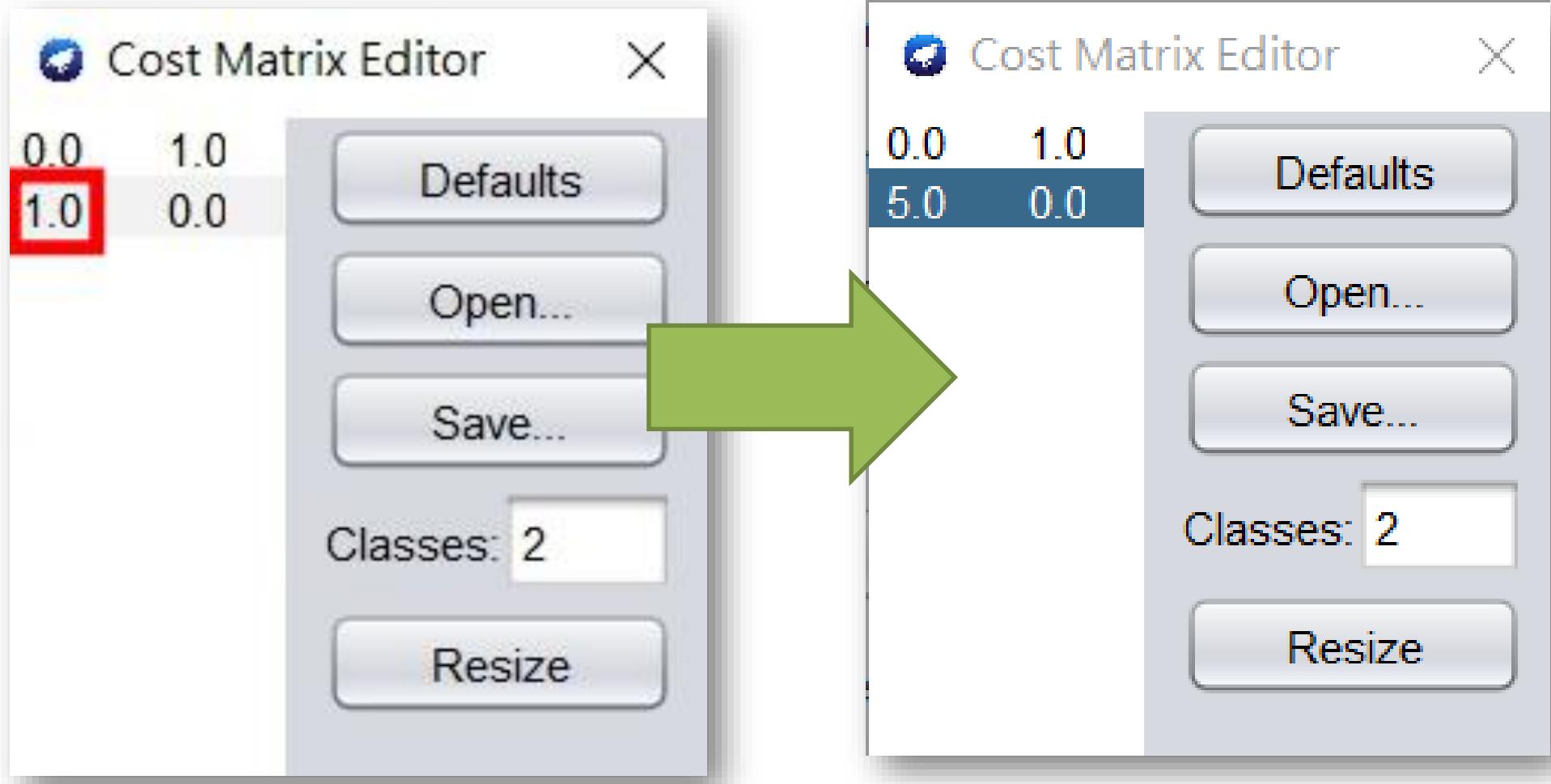
3. 左鍵單擊costMatrix的參數(左圖紅框處)開啟右圖視窗，並在Classes後方的輸入框中輸入2，接著左鍵單擊Resize按鈕。



Lesson 4.6: 成本敏感分類vs.成本敏感學習

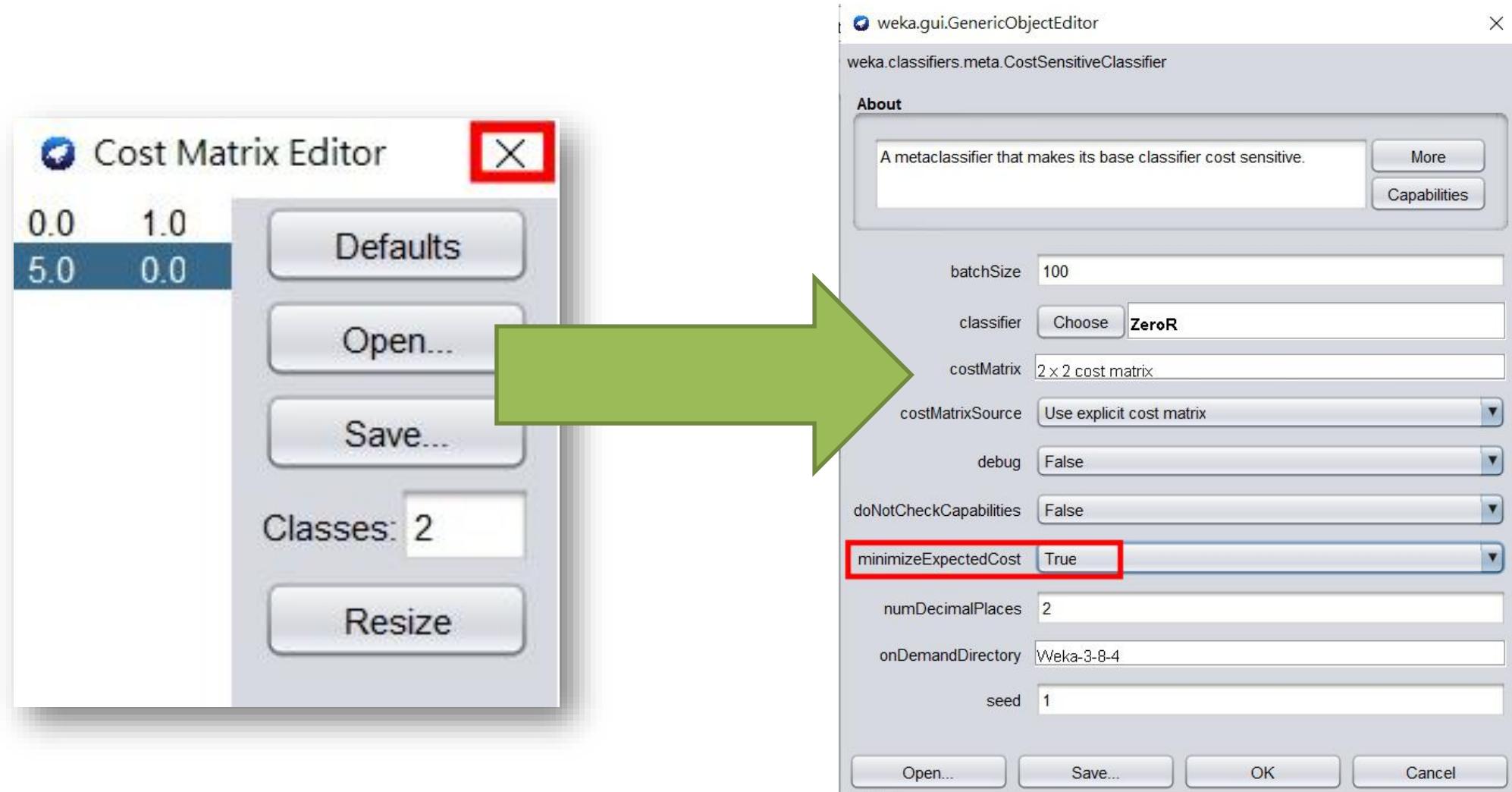
接著，將誤差成本設定成5。

4. 左鍵雙擊左圖紅框處，輸入5後按下視窗內空白處或enter鍵確定此設置。



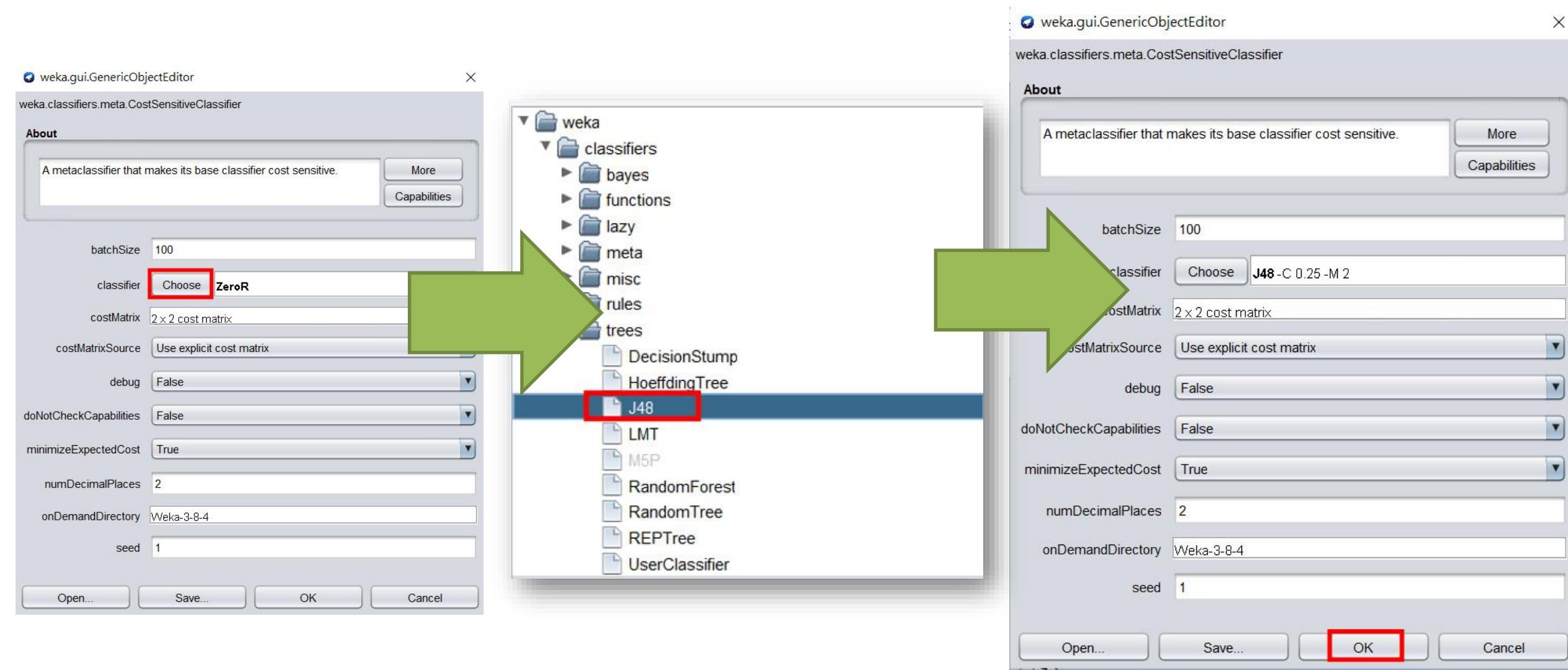
Lesson 4.6: 成本敏感分類vs.成本敏感學習

5. 左鍵單擊視窗右上方關閉按鈕(左圖紅框處)，並在分類器配置視窗中將參數minimizeExpectedCost變更為True。



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

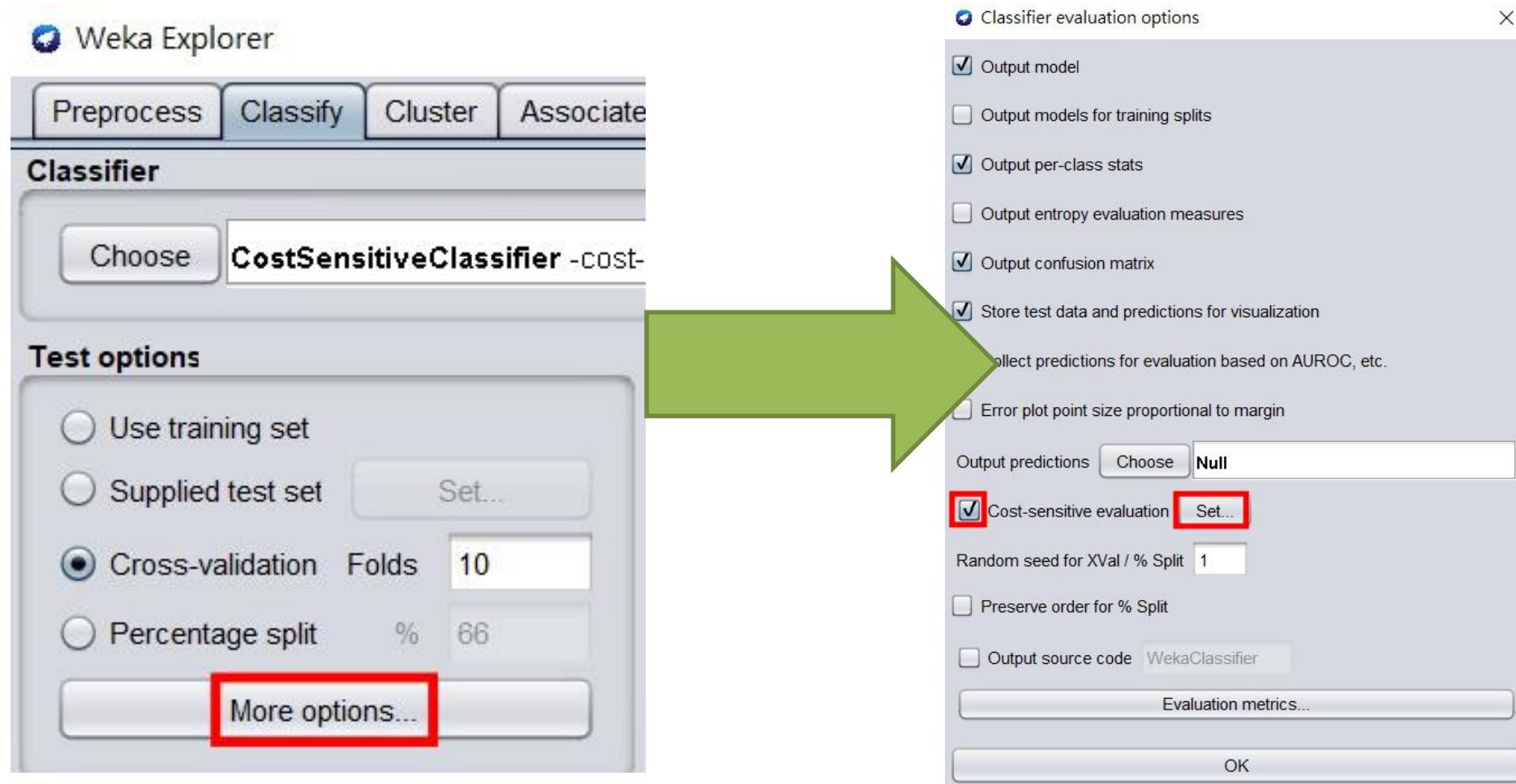
6. 在配置視窗中左鍵單擊Choose按鈕(左圖紅框處)，並在彈出的選單中以左鍵單擊J48分類器。接著左鍵單擊OK按鈕。



Lesson 4.6: 成本敏感分類vs.成本敏感學習

接著設定成本敏感評估。

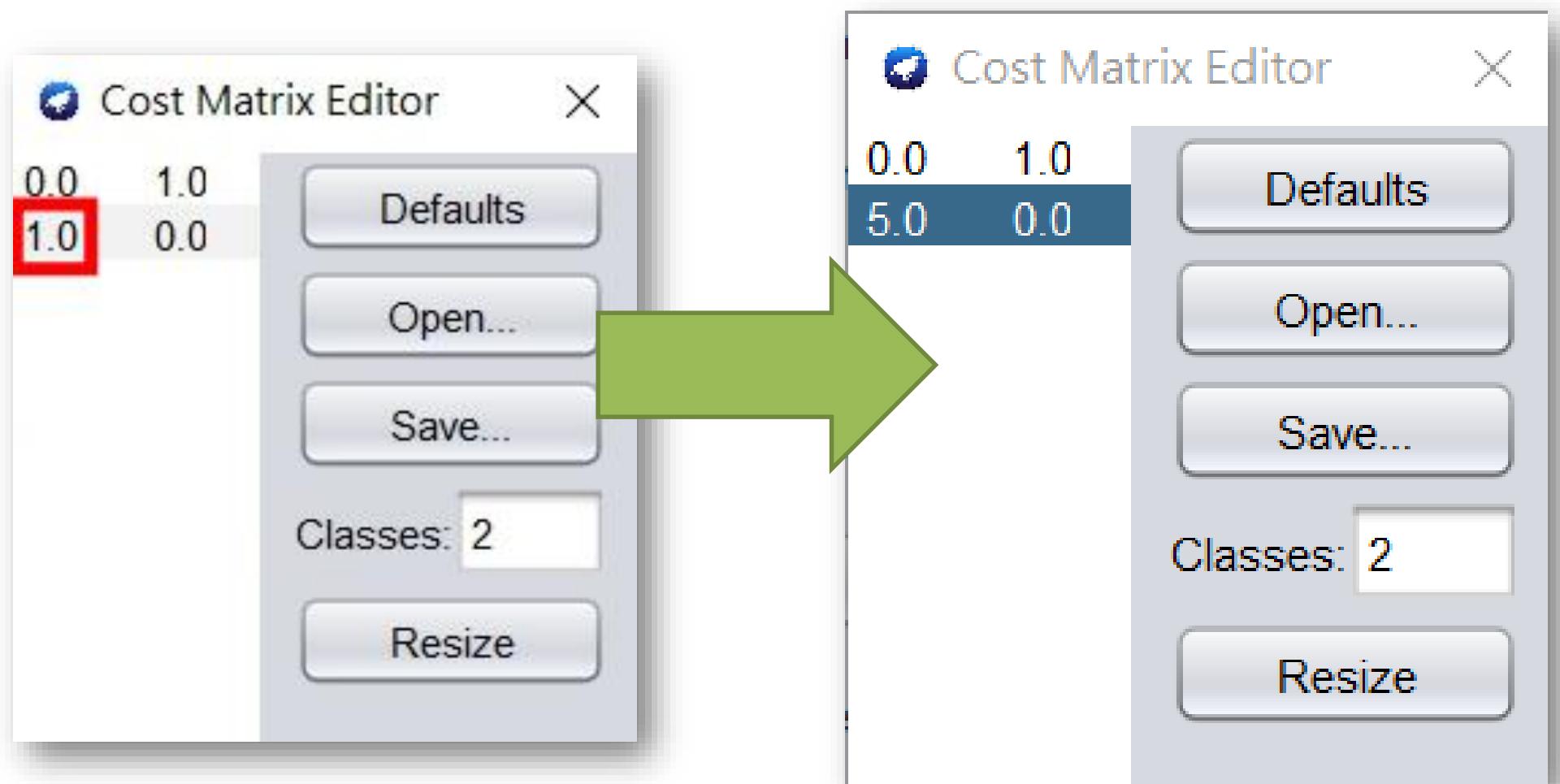
7. 左鍵單擊More options按鈕，並在彈出的視窗勾選Cost-sensitive evaluation選項。然後左鍵單擊其後的Set按鈕。



Lesson 4.6: 成本敏感分類vs.成本敏感學習

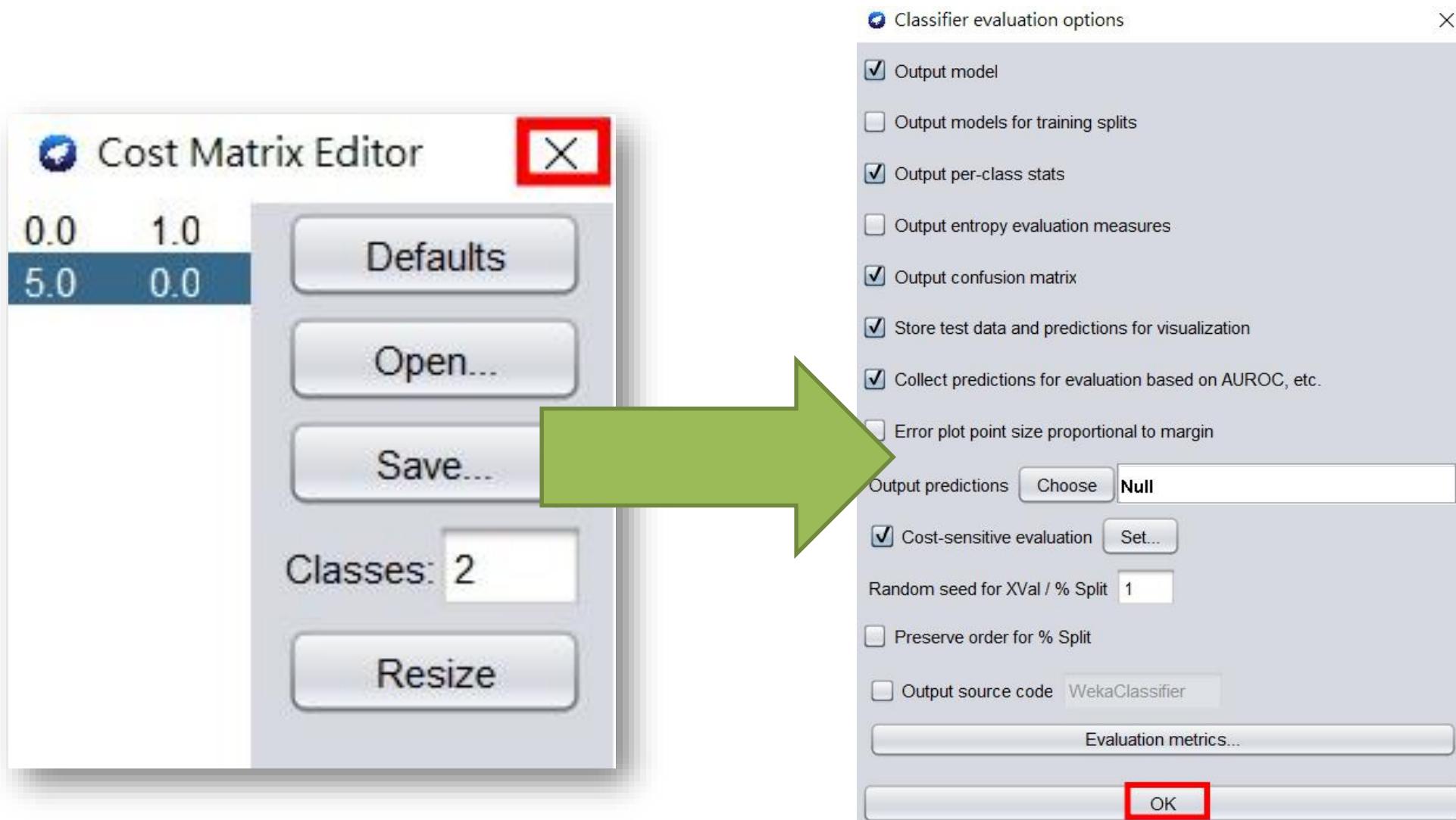
將誤差成本設定成5。

8. 左鍵雙擊左圖紅框處，輸入5後按下視窗內空白處或enter鍵確定此設置。



Lesson 4.6: 成本敏感分類vs.成本敏感學習

9. 左鍵單擊視窗右上方關閉按鈕(左圖紅框處)，並在More option的視窗中左鍵單擊OK按鈕。



Lesson 4.6: 成本敏感分類vs.成本敏感學習

10.回到Classify面板後，左鍵單擊Start按鈕，執行結果如右圖：得到總成本為770。

Weka Explorer

Preprocess Classify Cluster Associate

Classifier

Choose CostSensitiveClassifier -COST-r

Test options

Use training set

Supplied test set Set...

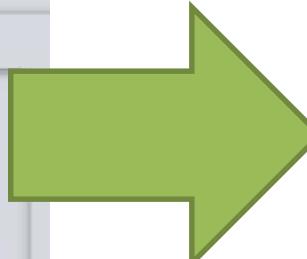
Cross-validation Folds 10

Percentage split % 66

More options...

(Nom) class

Start Stop



Classifier output

```
==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      650           65   %
Incorrectly Classified Instances   350           35   %
Kappa statistic                   0.2647
Total Cost                        770
Average Cost                      0.77
Mean absolute error                0.35
Root mean squared error            0.5916
Relative absolute error             83.2982 %
Root relative squared error       129.0994 %
Total Number of Instances         1000

==== Detailed Accuracy By Class ====

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
          0.650    0.350    0.813     0.650    0.722     0.277   0.650    0.773    good
          0.650    0.350    0.443     0.650    0.527     0.277   0.650    0.393    bad
Weighted Avg.    0.650    0.350    0.702     0.650    0.664     0.277   0.650    0.659

==== Confusion Matrix ====

      a   b  <- classified as
455 245 |  a = good
105 195 |  b = bad
```

Lesson 4.6: 成本敏感分類vs. 成本敏感學習

參數 `minimizeExpectedCost = true` 的
`CostSensitiveClassifier`

- ❖ 信用資料集 `credit-g.arff`; J48

- ❖ 成本矩陣:
a b
0 1 | a = good
5 0 | b = bad

a	b	<-- classified as
588	112	a = good
183	117	b = bad

成本: 1027

- ❖ `meta > CostSensitiveClassifier; minimizeExpectedCost = true`; 設定 `cost matrix`

- ❖ 選擇 `J48`

a	b	<-- classified as
455	245	a = good
105	195	b = bad

成本: 770

- ❖ 使用裝袋法(bagging) (Data Mining with Weka, Lesson 4.6)

... J48 產生一組有限的概率

- ❖ 裝袋(bagged) J48

a	b	<-- classified as
367	333	a = good
54	246	b = bad

成本: 603

Lesson 4.6: 成本敏感分類vs.成本敏感學習

方法 2:成本敏感學習(Cost-sensitive learning)

- ❖ 成本敏感分調節一個分類器的輸出
- ❖ 成本敏感學習學習一個不同的分類器r
- ❖ 創建具有複製的一些實例的新資料集
- ❖ 為了模擬成本矩陣
 - a b
 - 0 1 | a = good
 - 5 0 | b = bad
- ❖ 為每個bad實例添加4個副本

資料集credit-g 有700 個good 和300 個bad的實例(1000)

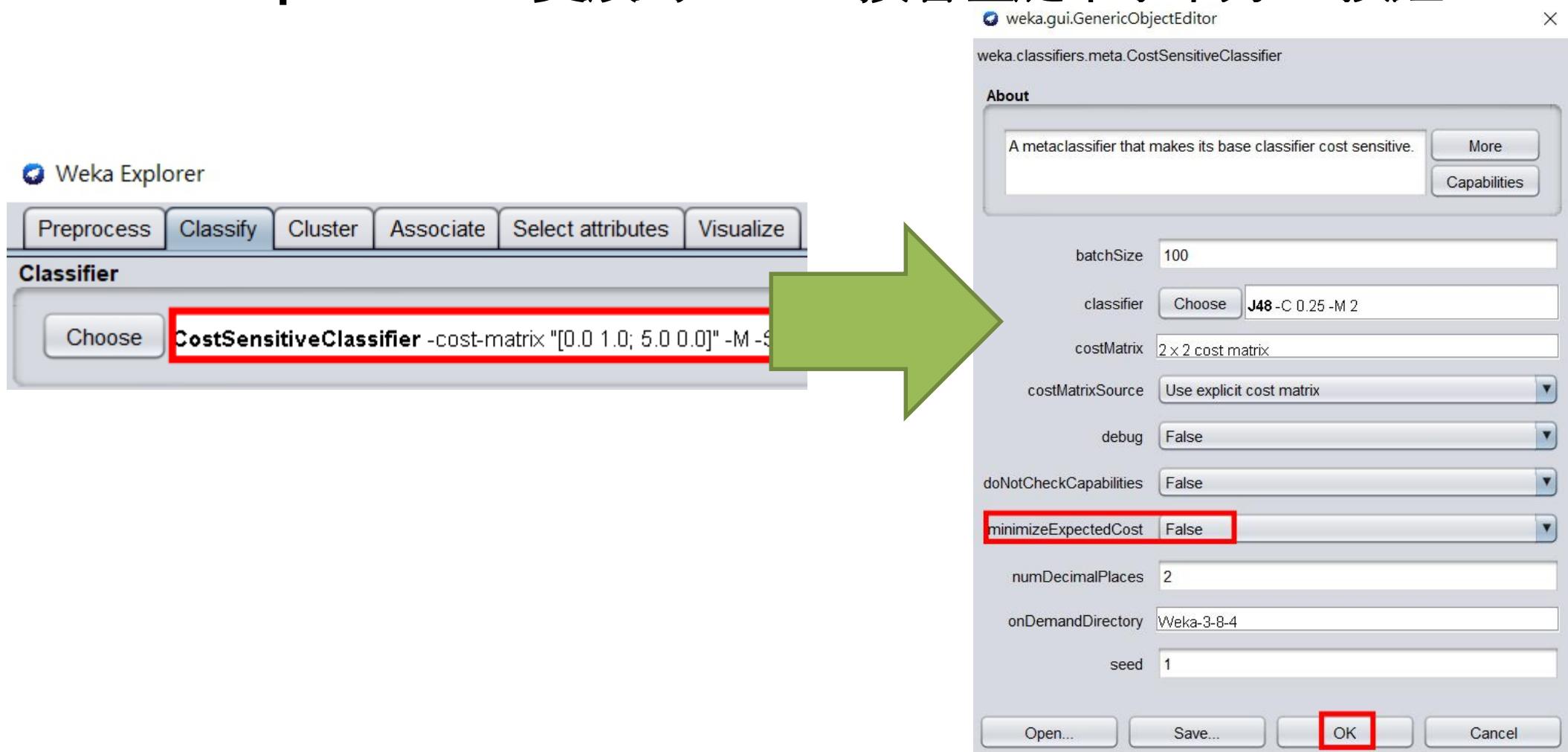
→ 新的資料集有700 個good和1500個bad的實例(2200)

... 然後重新學習!

Lesson 4.6: 成本敏感分類vs.成本敏感學習

我們試著minimizeExpectedCost設定為False。

1.左鍵單擊分類器名稱(左圖紅框處)，開啟配置視窗(右圖)。並將參數minimizeExpectedCost更改為False，接著左鍵單擊下方OK按鈕。



Lesson 4.6: 成本敏感分類vs. 成本敏感學習

2. 回到Classify面板後，左鍵單擊Start按鈕，執行結果如右圖：得到總成本為658。

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'CostSensitiveClassifier -cost-r' is chosen. The 'Test options' section shows 'Cross-validation' selected with 10 folds. A green arrow points from the 'Start' button in the bottom-left to the 'Classifier output' window. The output window displays stratified cross-validation results, including a summary table and detailed accuracy by class, ending with a confusion matrix.

Classifier output

```
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances      606          60.6 %
Incorrectly Classified Instances   394          39.4 %
Kappa statistic                   0.2492
Total Cost                        658
Average Cost                      0.658
Mean absolute error                0.397
Mean squared error                 0.5455
Root mean absolute error          94.4783 %
Root relative squared error       119.0321 %
Total Number of Instances         1000

==== Detailed Accuracy By Class ====

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
good	0.531	0.220	0.849	0.531	0.654	0.288	0.655	0.800	good
bad	0.780	0.469	0.416	0.780	0.543	0.288	0.655	0.384	bad
Weighted Avg.	0.606	0.295	0.719	0.606	0.621	0.288	0.655	0.675	

```
==== Confusion Matrix ====

```

	a	b	-- classified as
a	372	328	a = good
b	66	234	b = bad

Lesson 4.6: 成本敏感分類vs. 成本敏感學習

Weka中的成本敏感學習：

參數**minimizeExpectedCost = false** (預設)的**CostSensitiveClassifier**

- ❖ 信用資料集, 成本矩陣和之前的使用J48的credit-g.arff一樣
- ❖ meta > CostSensitiveClassifier; minimizeExpectedCost = false
- ❖ NaïveBayes

a	b	<- classified as
445	255	a = good
55	245	b = bad

成本 530

- ❖ J48

a	b	<- classified as
372	328	a = good
66	234	b = bad

成本 658

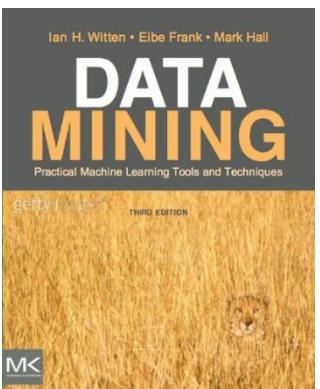
- ❖ bagged J48

a	b	<- classified as
404	296	a = good
57	243	b = bad

成本 581

Lesson 4.6: 成本敏感分類vs.成本敏感學習

- ❖ 成本敏感分類:調節一個分類器的輸出
- ❖ 成本敏感學習: 學習一個新的分類器
 - 通過適當地複製實例(效率低下!)
 - 或者在內部調整原始實例的權重
- ❖ **meta > CostSensitiveClassifier**
 - 應用在成本敏感分類和成本敏感學習
- ❖ 成本矩陣可以自動存儲和載入
 - 如:german-credit.cost
- ❖ *Section 5.7 Counting the cost*





THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

More Data Mining with Weka

Department of Computer Science
University of Waikato
New Zealand



Creative Commons Attribution 3.0 Unported License



creativecommons.org/licenses/by/3.0/

weka.waikato.ac.nz