

Data Preprocessing Applying Filters Example

Applying Filters Example

Some of the machine learning techniques such as association rule mining requires categorical data.

To illustrate the use of filters, we will use weather-numeric.arff database that contains two numeric attributes - temperature and humidity.

We will convert these to nominal by applying a filter on our raw data.

Click on the Choose button in the Filter subwindow and select the following filter: weka->filters->supervised->attribute->Discretize

Applying Filters Example

The screenshot shows the Weka Explorer application window. The 'Filter' tab is active, and the 'Discretize' filter is selected from the 'attribute' category. The 'Selected attribute' section shows 'Name: outlook', 'Missing: 0 (0%)', 'Distinct: 3', and 'Type: Nominal'. Below this, a table displays the distribution of the 'outlook' attribute:

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

The 'Class: play (Nom)' dropdown is set to 'play (Nom)', and the 'Visualize All' button is visible. The visualization area shows three stacked bar charts for the 'outlook' attribute. The first bar (sunny) has a red top half and a blue bottom half. The second bar (overcast) is entirely blue. The third bar (rainy) has a red top half and a blue bottom half. The status bar at the bottom shows 'OK' and a 'Log' button.

Applying Filters Example

Click on the Apply button and examine the temperature and/or humidity attribute.

Name: temperature		Type: Nominal	
Missing: 0 (0%)		Unique: 0 (0%)	
Distinct: 1			
No.	Label	Count	Weight
1	'All'	14	14.0

Select the temperature or humidity attribute You will notice that these have changed from numeric to nominal types.

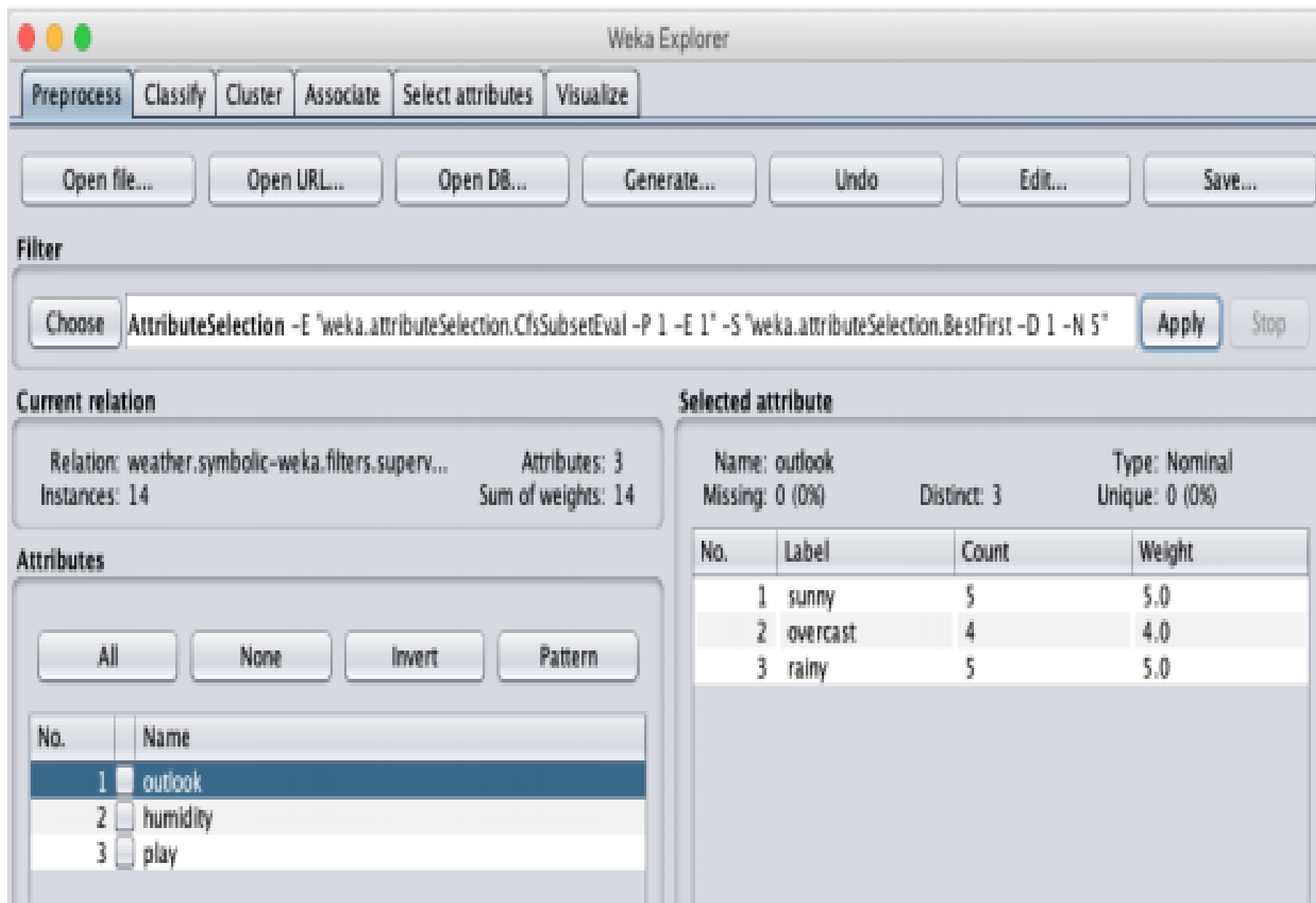
Applying Filters Example

Let us look into another filter now. Suppose you want to select the best attributes for deciding the play.

Select and apply the following filter:

```
weka->filters->supervised->attribute-  
>AttributeSelection
```

You will notice that it removes the temperature and humidity attributes from the database.



WEKA — Classifiers

Many machine learning applications are classification related.

For example, you may like to classify a tumor as malignant or benign.

You may like to decide whether to play an outside game depending on the weather conditions.

Generally, this decision is dependent on several features/conditions of the weather.

So you may prefer to use a tree classifier to make your decision of whether to play or not.

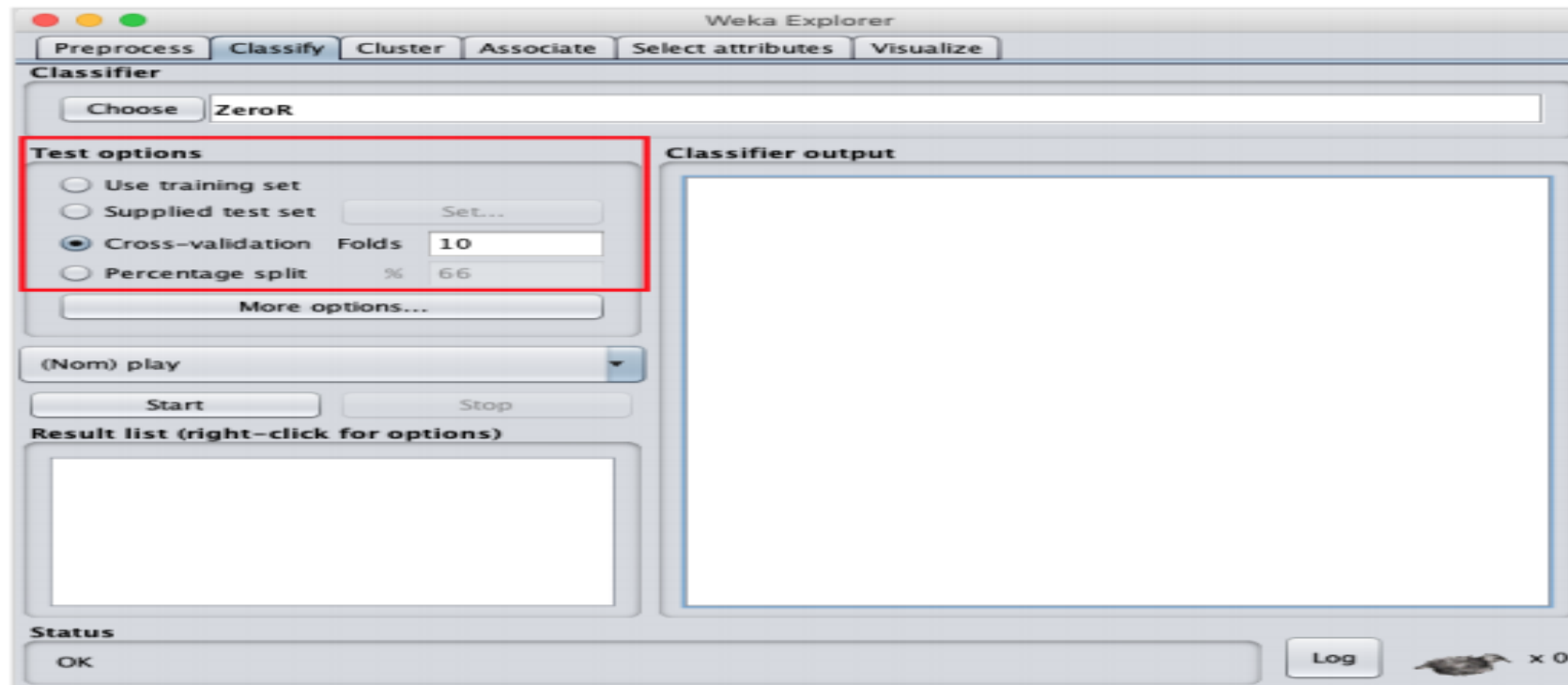
WEKA – Classifiers

we will learn how to build such a tree classifier on weather data to decide on the playing conditions.

Setting Test Data

We will use the preprocessed weather data file from the previous slides above.

Open the saved file by using the Open file ... option under the Preprocess tab, click on the Classify tab, and you would see the following screen:



WEKA — Classifiers

Before you learn about the available classifiers, let us examine the Test options.

You will notice four testing options as listed below:

- ▣ Training set
- ▣ Supplied test set
- ▣ Cross-validation
- ▣ Percentage split

Unless you have your own training set or a client supplied test set, you would use crossvalidation or percentage split options.

WEKA — Classifiers

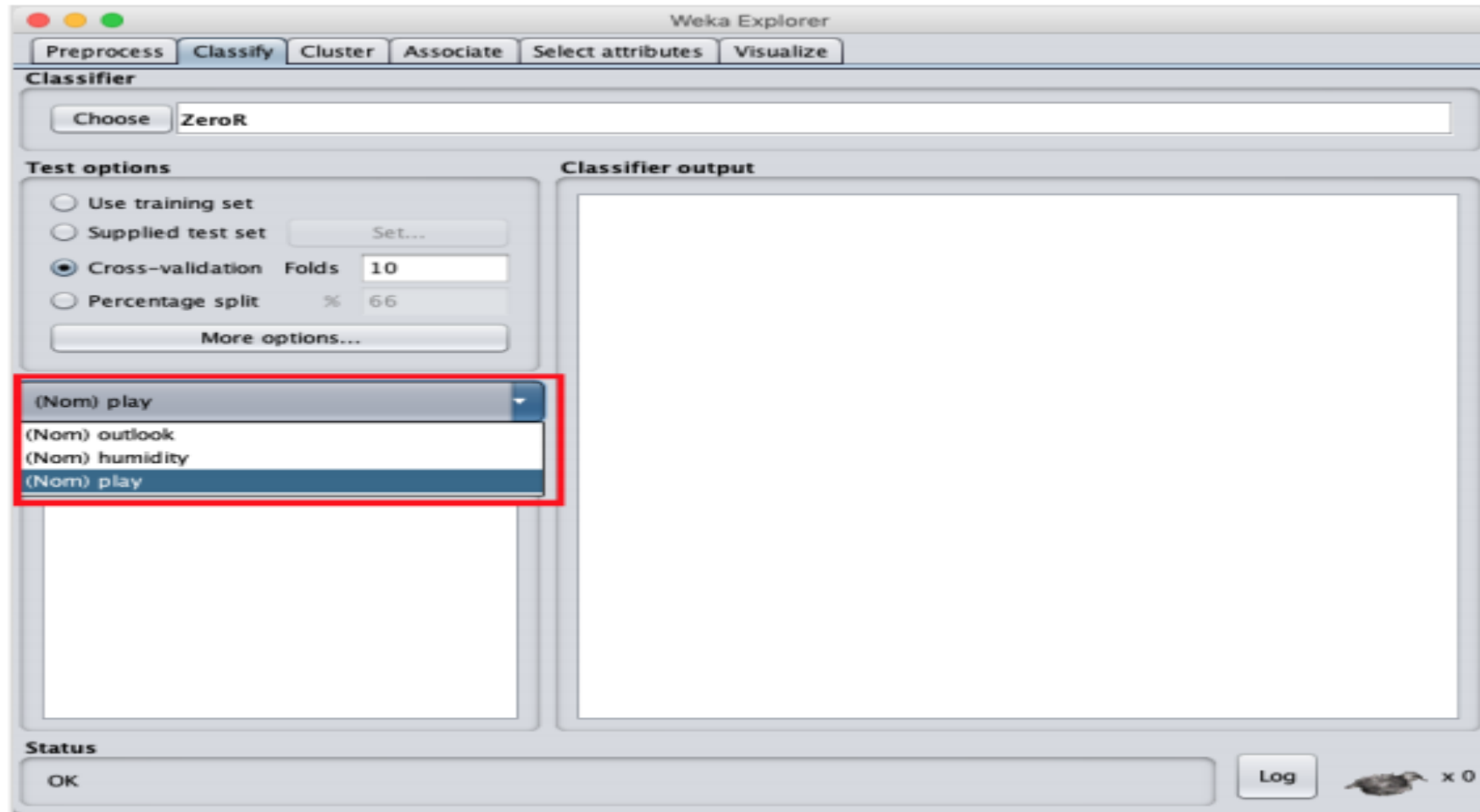
Under cross-validation, you can set the number of folds in which entire data would be split and used during each iteration of training.

In the percentage split, you will split the data between training and testing using the set split percentage.

Now, keep the default play option for the output class:

WEKA — Classifiers

Now, keep the default play option for the output class:

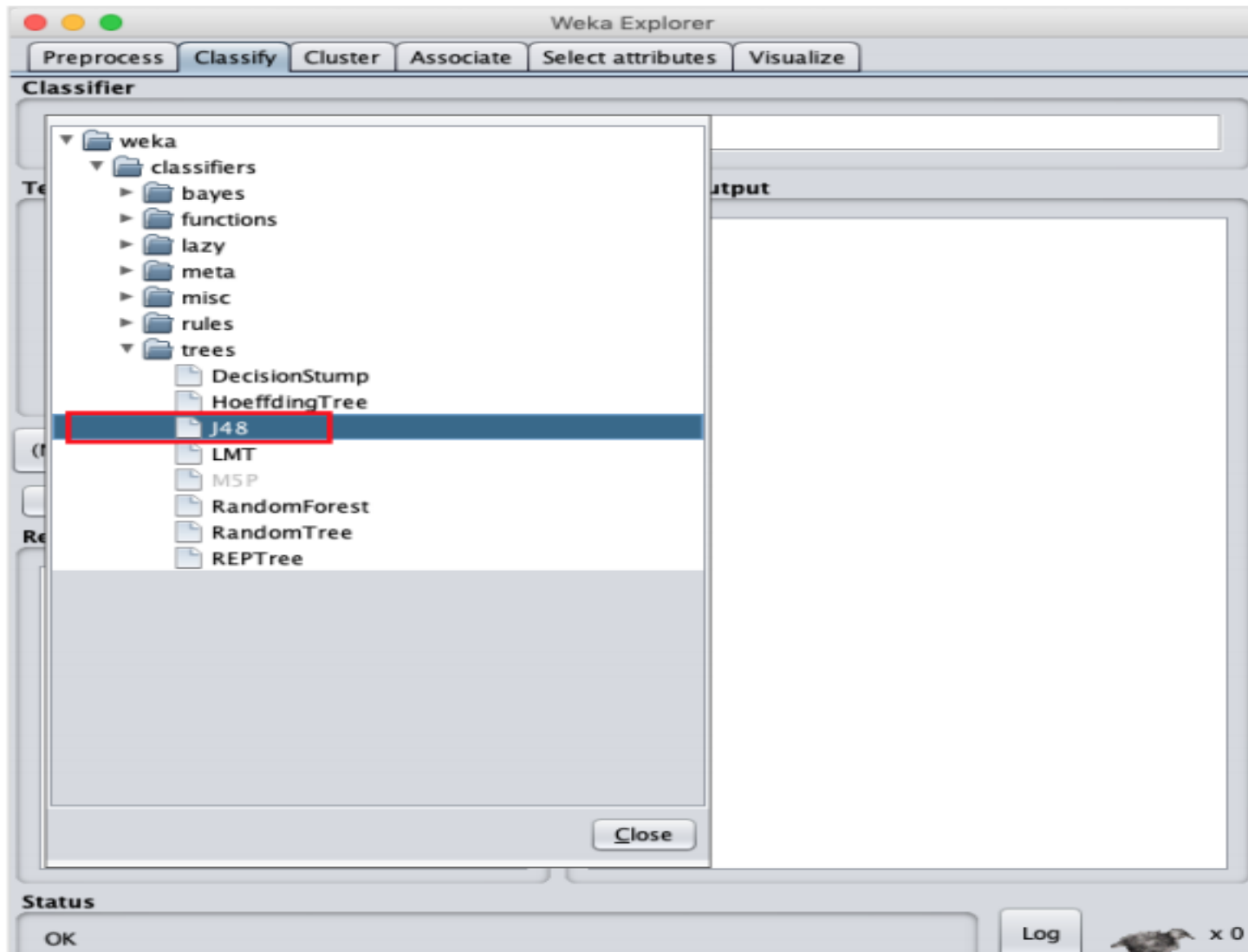


WEKA — Classifiers

Next, you will select the classifier.

Click on the Choose button and select the following classifier:

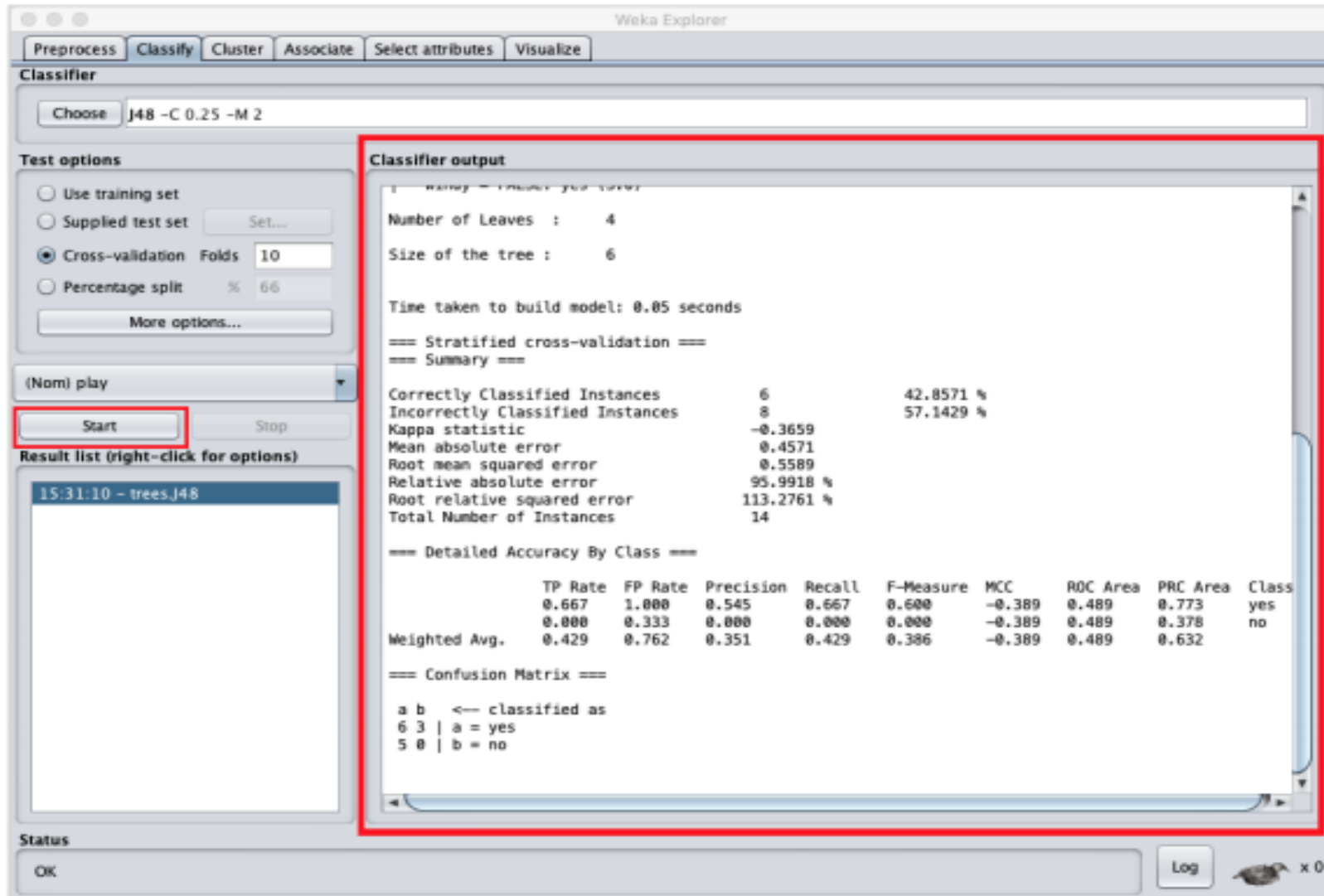
weka->classifiers>trees>J48 This is shown in the screenshot below:



WEKA — Classifiers

Click on the Start button to start the classification process.

After a while, the classification results would be presented on your screen as shown here:



The screenshot shows the Weka Explorer window with the 'Classify' tab selected. The classifier chosen is 'J48 -C 0.25 -M 2'. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Start' button is highlighted with a red box. The 'Classifier output' window is also highlighted with a red box and contains the following information:

Classifier output

Number of Leaves : 4
Size of the tree : 6
Time taken to build model: 0.05 seconds

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

Correctly Classified Instances	6	42.8571 %
Incorrectly Classified Instances	8	57.1429 %
Kappa statistic	-0.3659	
Mean absolute error	0.4571	
Root mean squared error	0.5589	
Relative absolute error	95.9918 %	
Root relative squared error	113.2761 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.667	1.000	0.545	0.667	0.600	-0.389	0.489	0.773	yes
	0.000	0.333	0.000	0.000	0.000	-0.389	0.489	0.378	no
Weighted Avg.	0.429	0.762	0.351	0.429	0.386	-0.389	0.489	0.632	

=== Confusion Matrix ===

a b	<-- classified as	
6 3	a = yes	
5 0	b = no	

The 'Result list' at the bottom left shows a single entry: '15:31:10 - trees.J48'.

WEKA — Classifiers

Let us examine the output shown on the right hand side of the screen.

It says the size of the tree is 6.

it says that the correctly classified instances as 2 and the incorrectly classified instances as 3,
It also says that the Relative absolute error is 110%.

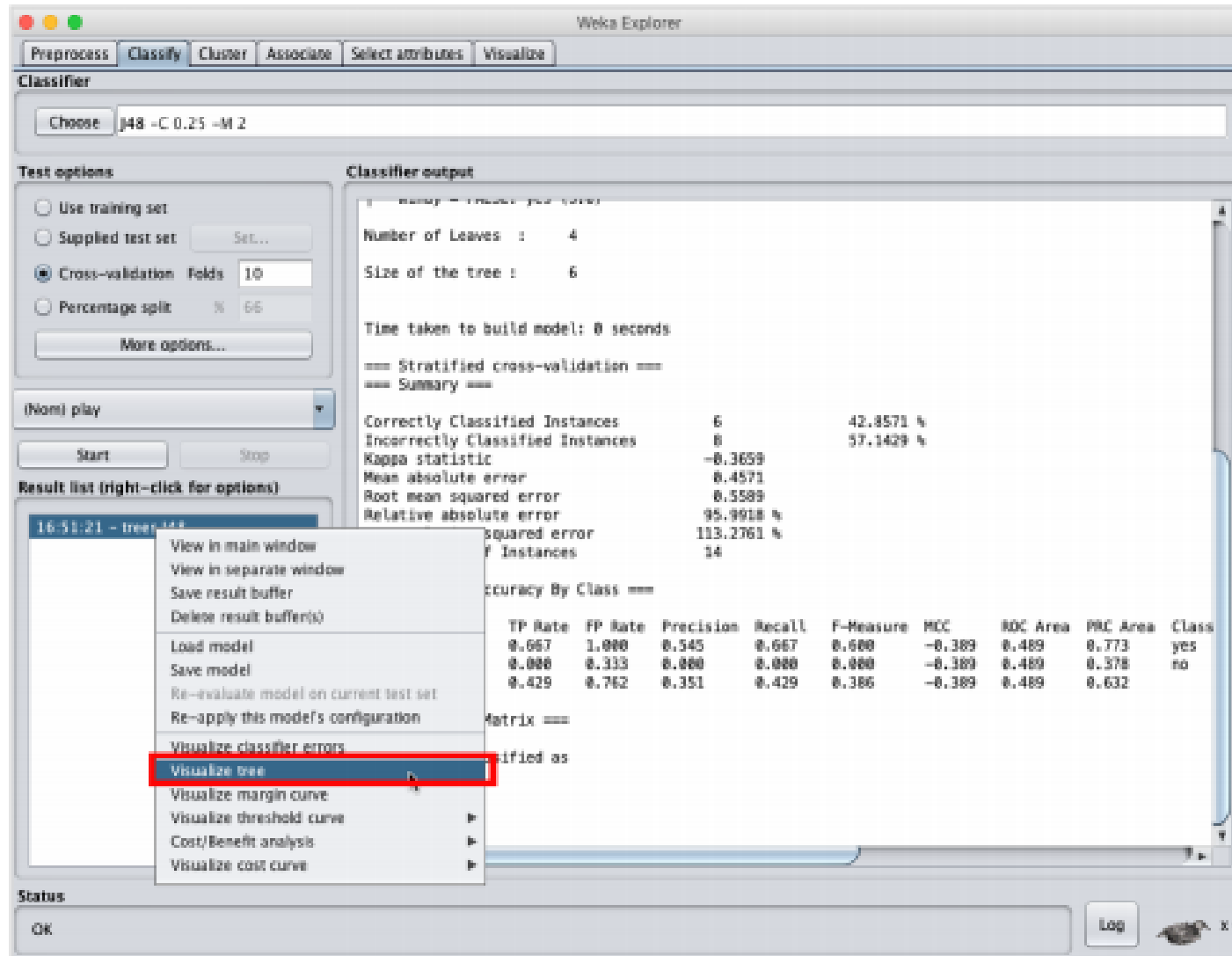
It also shows the Confusion Matrix.

rebuild the model and so on until you are satisfied with the model's accuracy.

Anyway, that's what WEKA is all about. It allows you to test your ideas quickly.

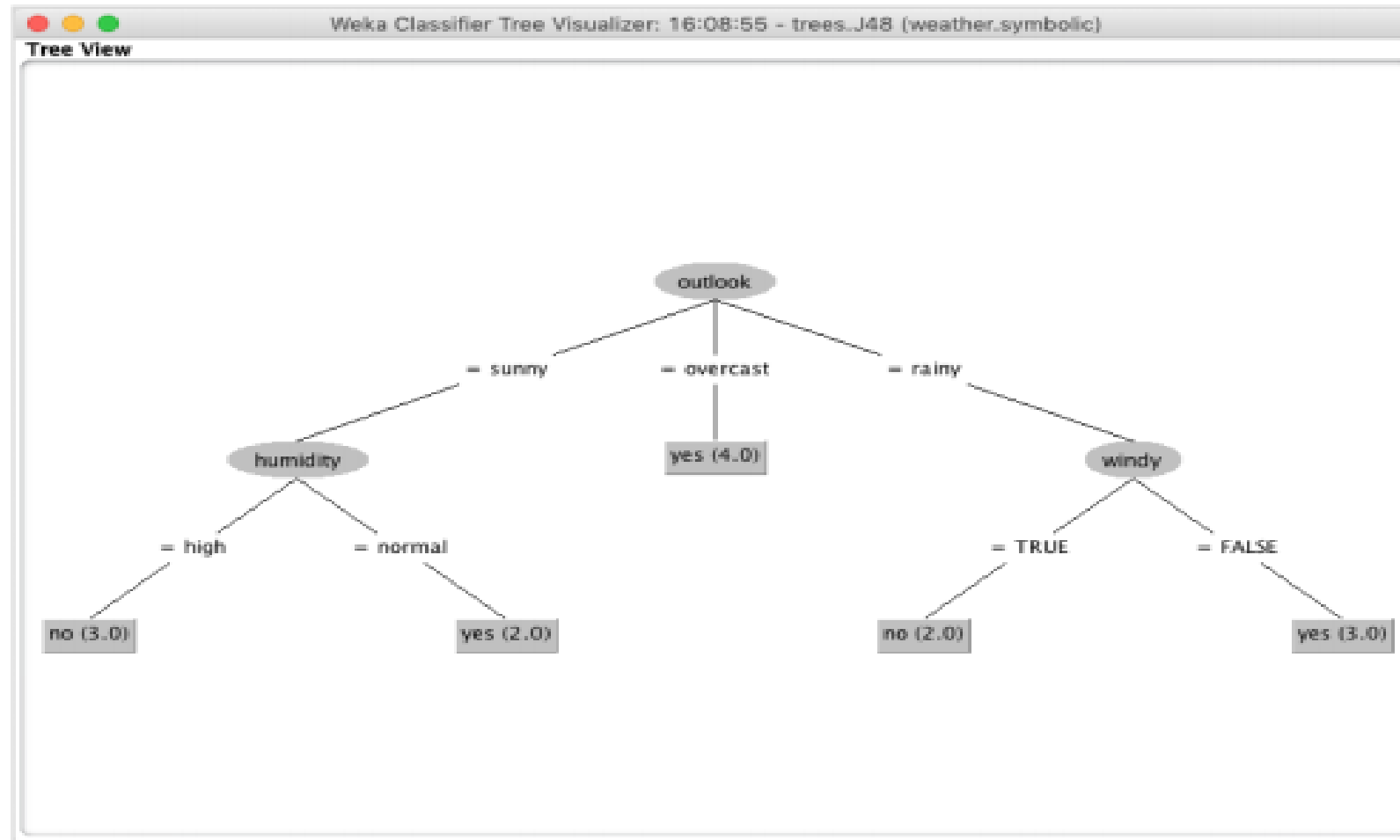
Visualize Results

To see the visual representation of the results, right click on the result in the Result list box. Several options would pop up on the screen as shown here:



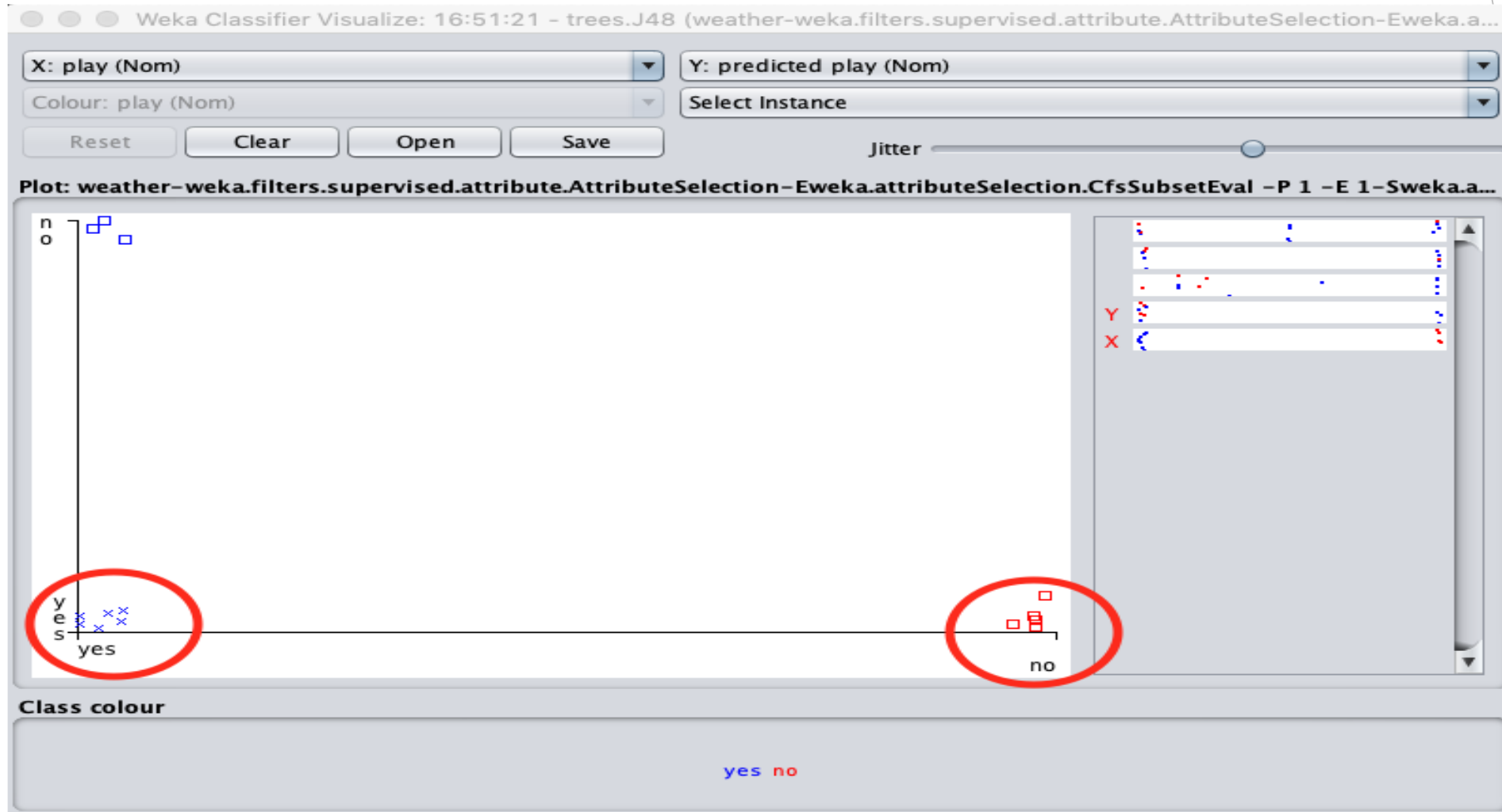
Visualize Results

Select Visualize tree to get a visual representation of the traversal tree as seen in the screenshot below:



Visualize Results

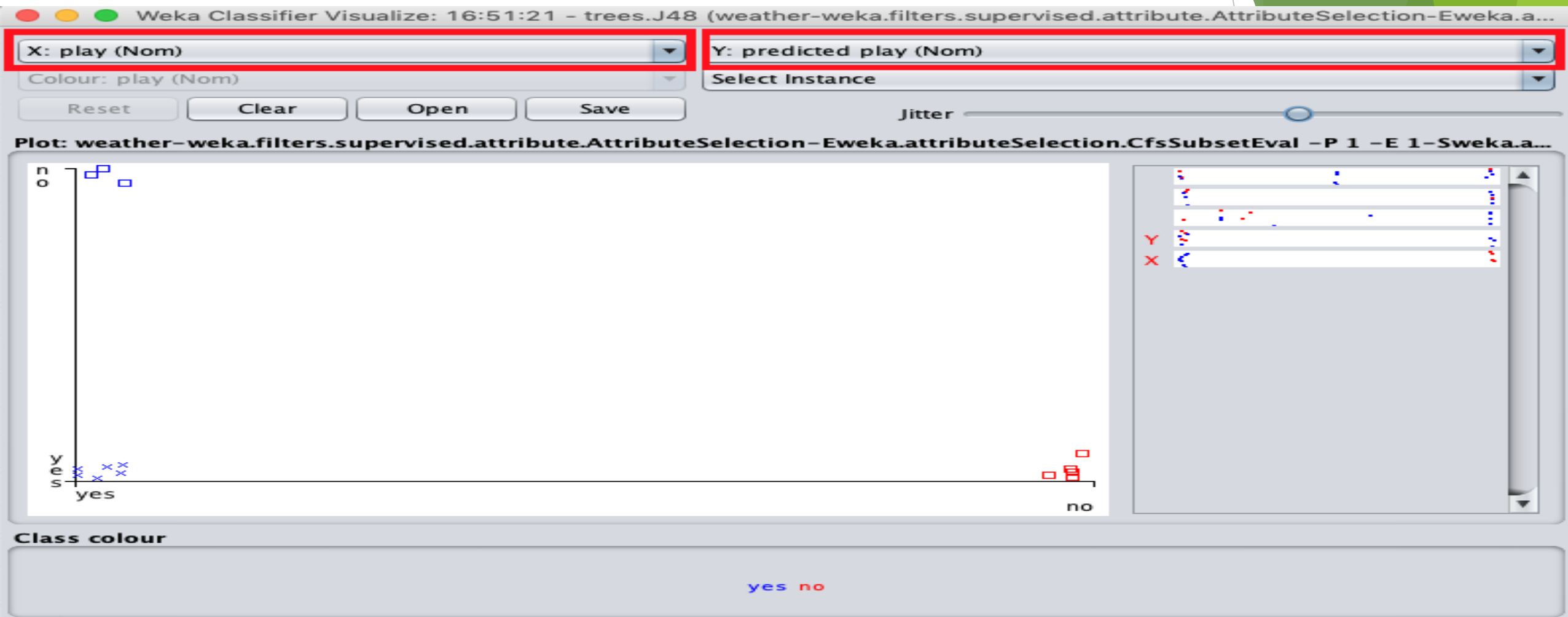
Selecting Visualize classifier errors would plot the results of classification as shown here:



A **cross** represents a correctly classified instance while **squares** represents incorrectly classified instances. At the lower left corner of the plot you see a **cross** that indicates if **outlook** is sunny then **play** the game. So this is a correctly classified instance. To locate instances, you can introduce some jitter in it by sliding the **jitter** slide bar.

Visualize Results

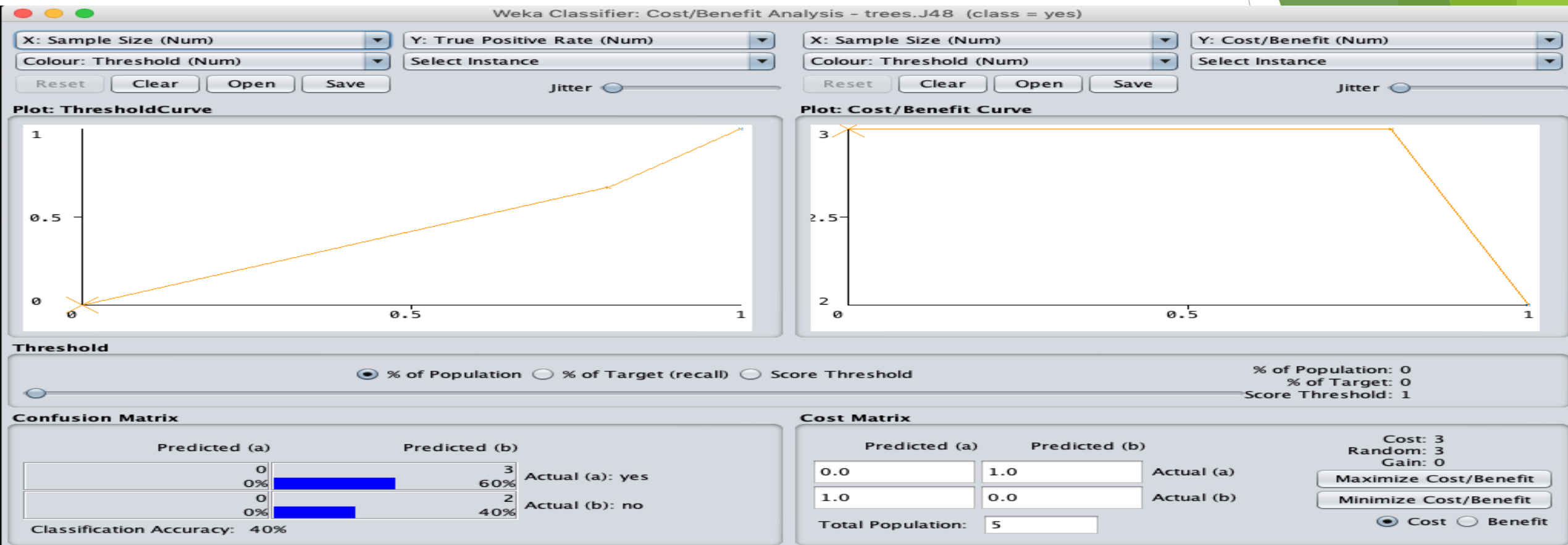
The current plot is **outlook** versus **play**. These are indicated by the two drop down list boxes at the top of the screen.



Now, try a different selection in each of these boxes and notice how the X & Y axes change. The same can be achieved by using the horizontal strips on the right hand side of the plot. Each strip represents an attribute. Left click on the strip sets the selected attribute on the X-axis while a right click would set it on the Y-axis.

Visualize Results

There are several other plots provided for your deeper analysis. Use them judiciously to fine tune your model. One such plot of Cost/Benefit analysis is shown below for your quick reference.



Now, try a different selection in each of these boxes and notice how the X & Y axes change. The same can be achieved by using the horizontal strips on the right hand side of the plot. Each strip represents an attribute. Left click on the strip sets the selected attribute on the X-axis while a right click would set it on the Y-axis.