

EX : 7 Naïve bayes classification on a given data set



Preparing data for classification

We will use the same data set as in the previous example with weather features, **temperature** and **humidity**, and class **yes/no** for playing golf.

Data is stored in *arff* file format specific for WEKA software and looks like this:

```
@relation 'weather.symbolic-weka.filters.unsupervised.attribute.Remove-R1,4'
```

```
@attribute temperature {hot,mild,cool}
```

```
@attribute humidity {high,normal}
```

```
@attribute play {yes,no}
```

```
@data
```

```
hot,high,no
```

```
hot,high,no
```

```
hot,high,yes
```

```
mild,high,yes
```

```
cool,normal,yes
```

```
cool,normal,no
```

```
cool,normal,yes
```

```
mild,high,no
```

```
cool,normal,yes
```

```
mild,normal,yes
```

mild,normal,yes

mild,high,yes

hot,normal,yes

mild,high,no


Here we can see the attribute denominators: temperature, humidity, and play, followed by the data table. Using this data set, we will train the Naive Bayes model and then apply it to new data with temperature **cool** and humidity **high** to see to which class it will be assigned.

First of all, in WEKA explorer *Preprocess* tab, we need to open our ARFF data file:

The screenshot shows the Weka Explorer interface in the Preprocess tab. The 'Current relation' is 'weather...' with 14 instances and 3 attributes. The 'Selected attribute' is 'temperature', which is a nominal attribute with 3 distinct values and 0 missing values. The 'Attributes' list shows 'temperature', 'humidity', and 'play'. The 'Status' bar shows 'OK'. A bar chart at the bottom right visualizes the distribution of the 'temperature' attribute, with bars for 'hot' (count 4), 'mild' (count 6), and 'cool' (count 4). The bars are stacked with blue at the bottom and red on top.

No.	Label	Count	Weight
1	hot	4	4.0
2	mild	6	6.0
3	cool	4	4.0

Here we can see the basic statistics of attributes. If you click *the Edit* button, the new Viewer window with the data table will be loaded.

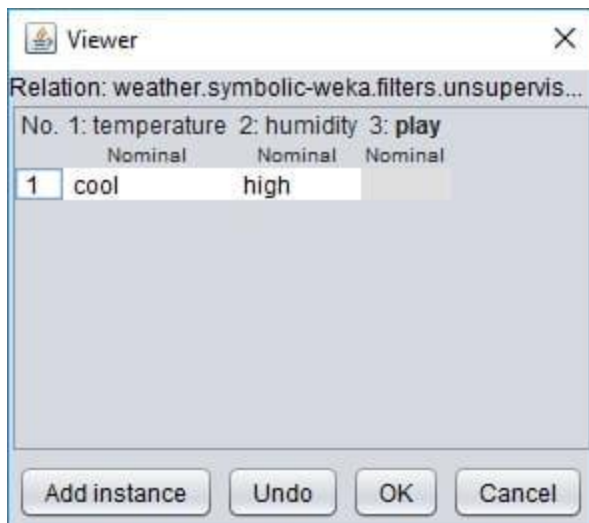


Relation: weather.symbolic-weka.filters.unsuper...

No.	1: temperature	2: humidity	3: play
	Nominal	Nominal	Nominal
1	hot	high	no
2	hot	high	no
3	hot	high	yes
4	mild	high	yes
5	cool	normal	yes
6	cool	normal	no
7	cool	normal	yes
8	mild	high	no
9	cool	normal	yes
10	mild	normal	yes
11	mild	normal	yes
12	mild	high	yes
13	hot	normal	yes
14	mild	high	no

Buttons: Add instance, Undo, OK

You can edit data as you like in the viewer, and then you can permanently save new data set with the Save button in explorer. We will do so when we create a test set with cool and high parameter values. For this, we delete all lines of data except the first one and edit values to look like this:



Relation: weather.symbolic-weka.filters.unsupervis...

No.	1: temperature	2: humidity	3: play
	Nominal	Nominal	Nominal
1	cool	high	

Buttons: Add instance, Undo, OK, Cancel

Select nothing on play attribute because we don't know it yet.

Click OK and then Save data as a separate file. The file should look like this:

```
@relation 'weather.symbolic-weka.filters.unsupervised.attribute.Remove-R1,4'
```

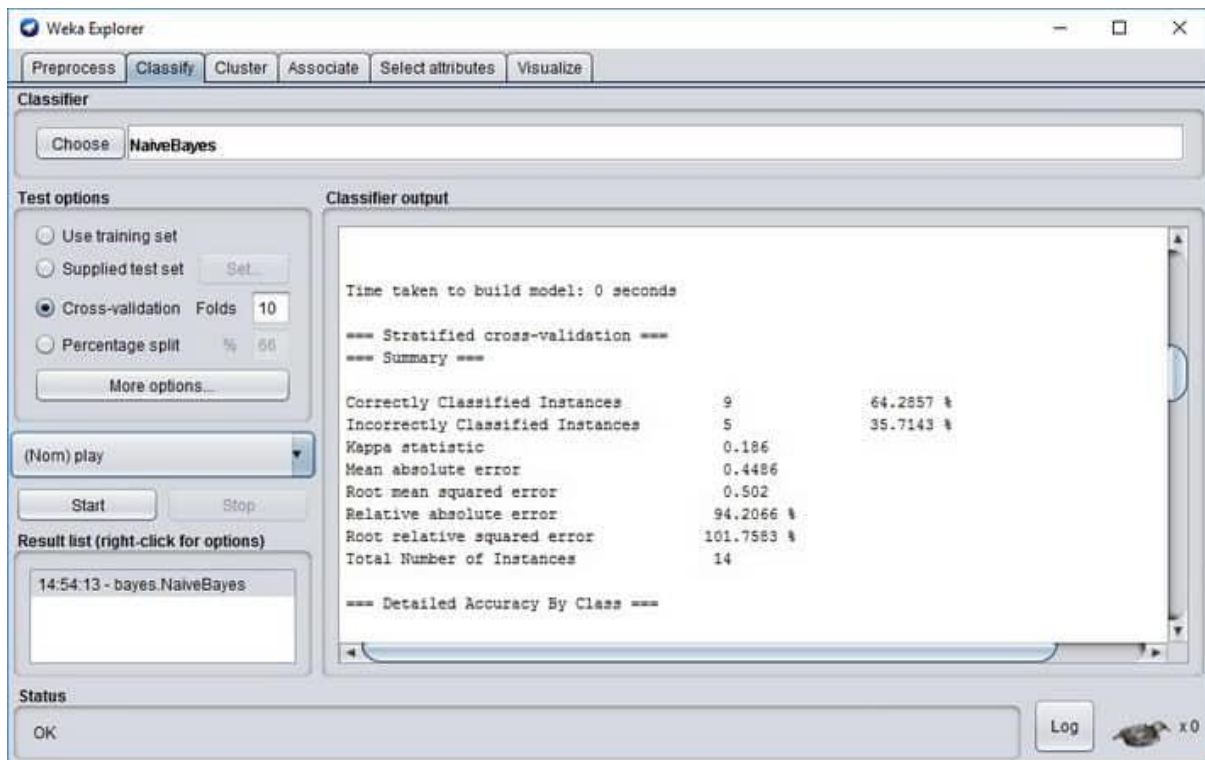
@attribute temperature {hot,mild,cool}
@attribute humidity {high,normal}
@attribute play {yes,no}
@data
cool,high,?

The question “?” mark is a standard way of representing the missing values in WEKA.

Building a Naive Bayes model

Now that we have data prepared, we can proceed with building the model. Load complete weather data set again in explorer and then go to *Classify* tab.

Here you need to press *the* Choose Classifier button, and from the tree menu, select NaiveBayes. Be sure that the Play attribute is selected as a class selector, and then press **the Start** button to build a model.



Model outputs some information on how accurate it classifies and other parameters.

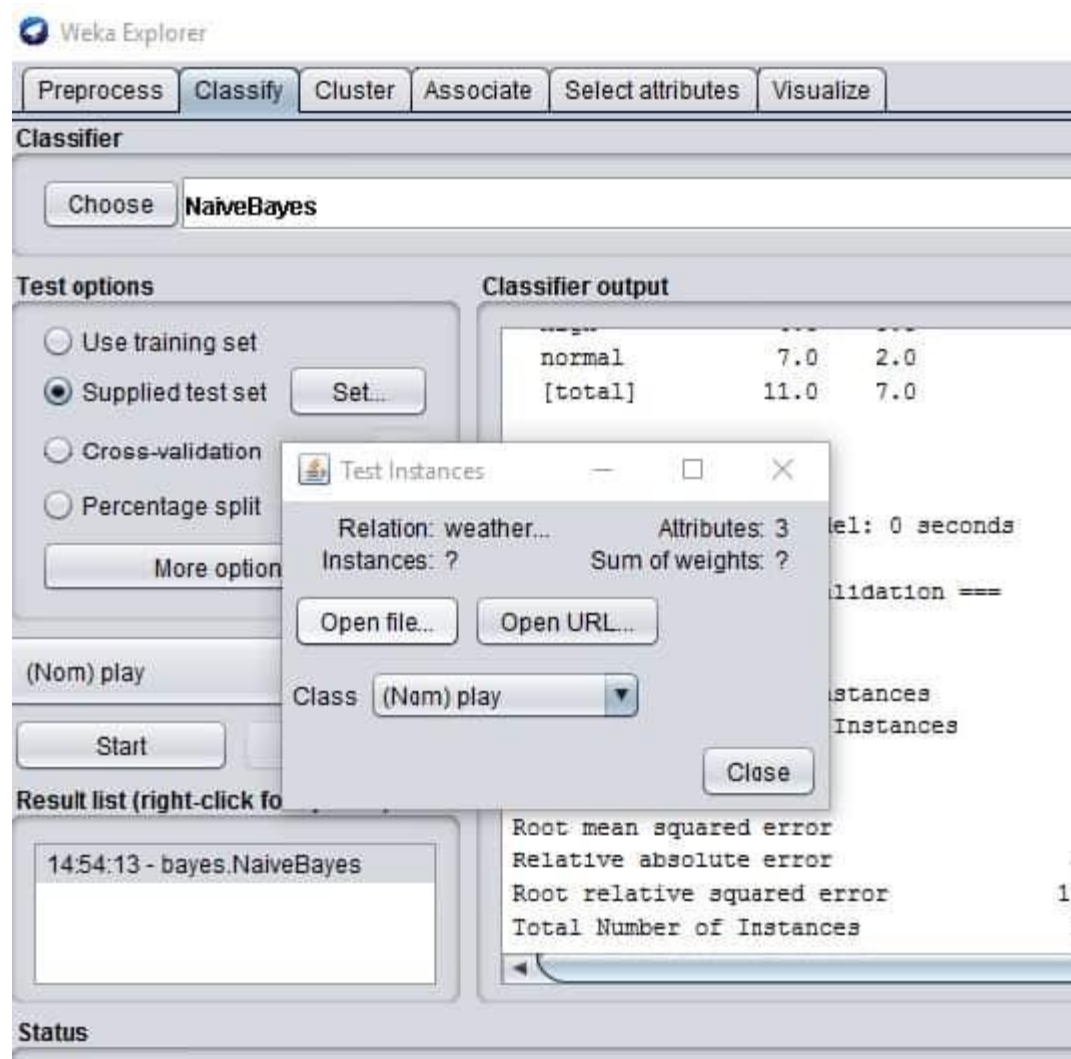
Correctly Classified Instances 9 64.2857 %

Incorrectly Classified Instances 5 35.7143 %

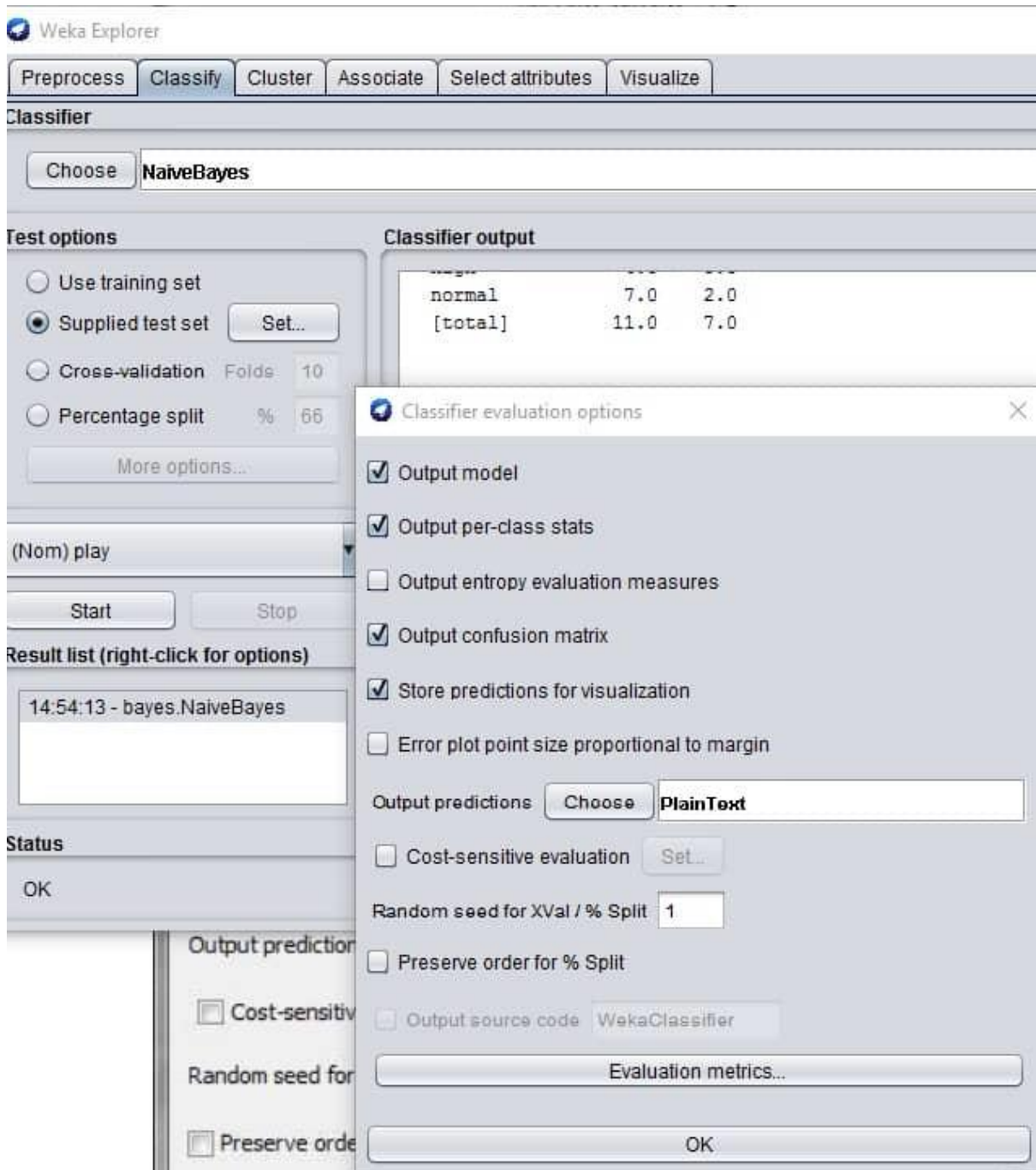
You can see that on a given data set, the classifier's accuracy is about 64%. So remember that you shouldn't always take the results for granted. To get better results, you might want to try different classifiers or preprocess data even further. We won't get into this right now. We need to demonstrate the usage of the model on new upcoming data.

Evaluating classifier with the test set

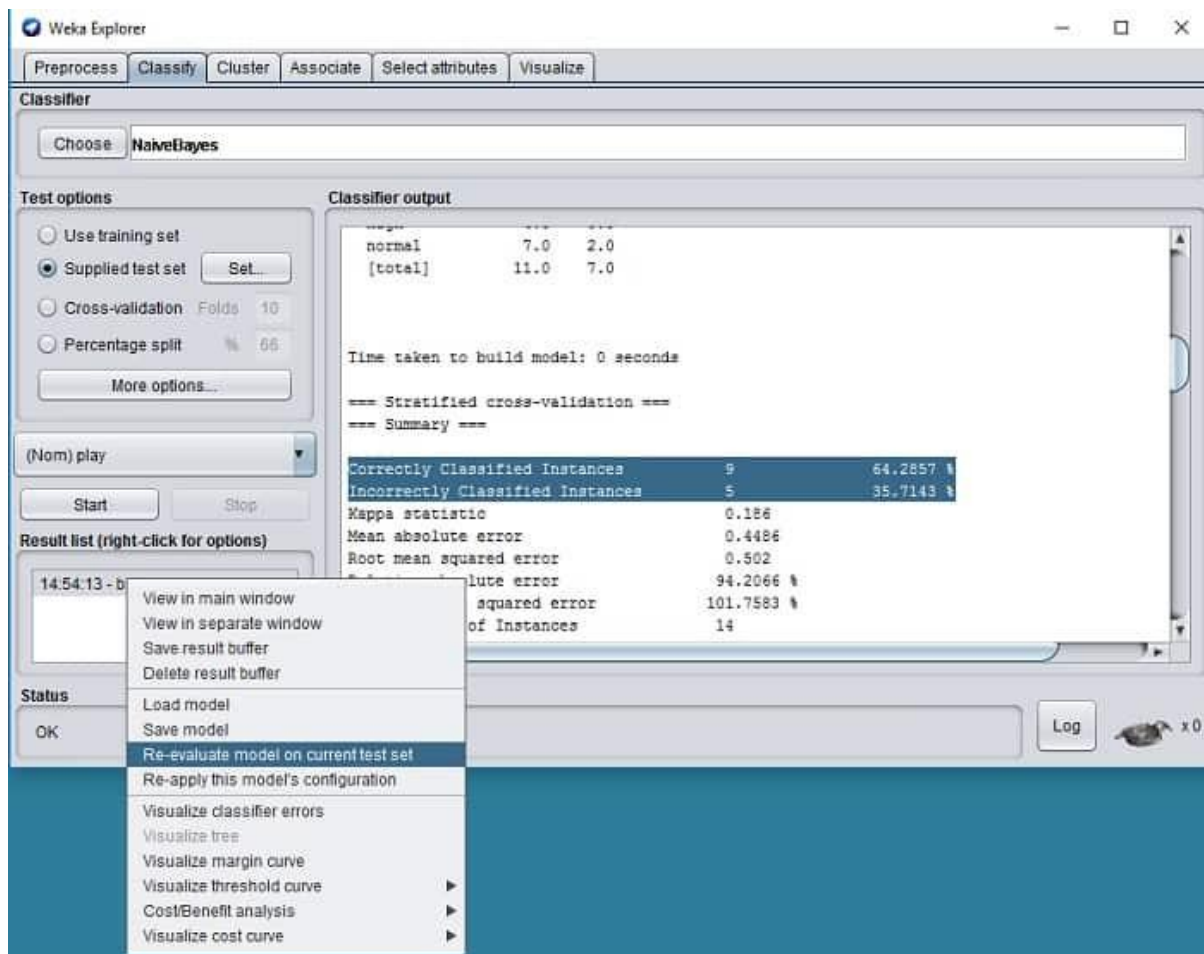
Now when we have a model, we need to load the test data we've created before. For this, select Supplied test set and click the button Set.



Click *More Options* wherein new window choose **PlainText** from *Output predictions* as follows:



Then click the left mouse button on a recently created model on the result list and select *Re-evaluate model on the current test set*.



And you should see the prediction for your given data cool and hot like this:

=== Predictions on user test set ===

inst#	actual	predicted	error	prediction
1	1:?	1:yes	0.531	

RESULT

It has been predicted as yes, with an error of 53.1%. In the previous analytical example, we've got a 50% error on prediction.