Velmi stručný úvod do použití systému WEKA pro Data Mining

(Jan Žižka, ÚI PEF)

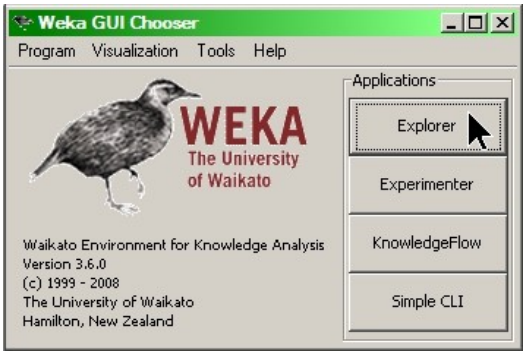
Systém WEKA, implementovaný v jazyce Java, lze získat nejlépe z následující URL:

<**http://www.cs.waikato.ac.nz/ml/weka/**>.

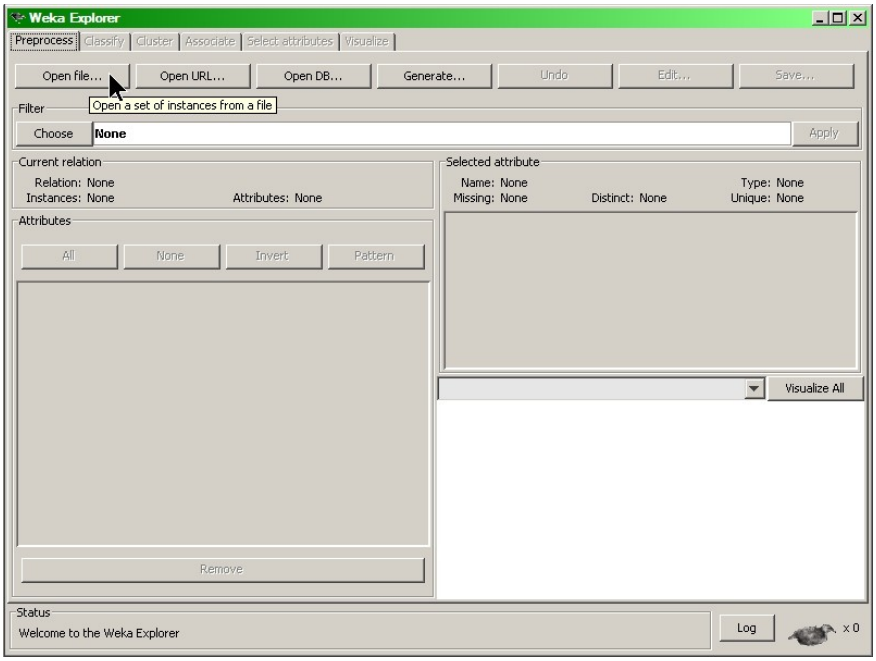
Dále je nutno vybrat **Download/Stable GUI version** dle operačního systému (MS Windows, Linux, Mac OS X). Doporučuji poslední verzi 3.6.0 (ke dni 10.3.2009).

Možnosti: **weka-3-6-0jre** instaluje i Java VM (Java Virtual Machine), **weka-3-6-0** je bez instalace Java VM (pokud už je Java VM nainstalováno z dřívějška).

Po instalaci lze systém WEKA spustit prostřednictvím GUI (existuje i možnost spuštění z příkazového řádku). Objeví se úvodní okno:

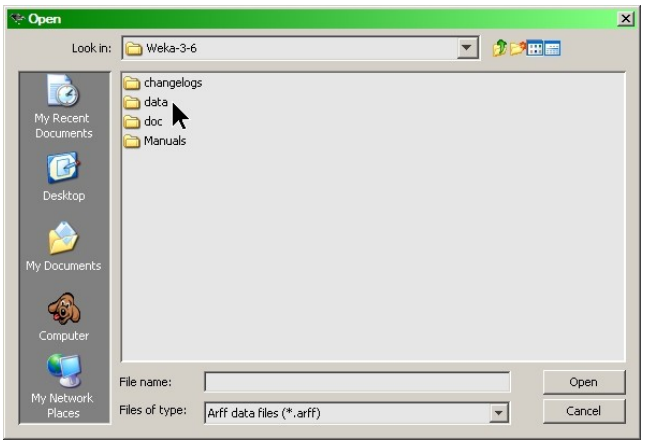


Dobré je začít s variantou **Explorer**:

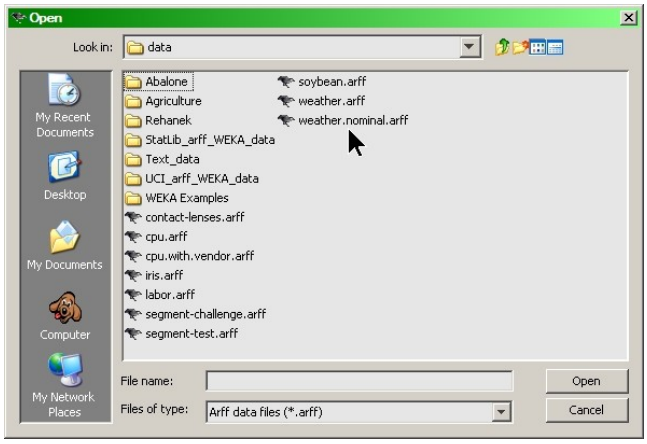


Menu **Preprocess** umožňuje otevřít nějaká existující data přes **Open file**. Lze si vybrat příklad, který je součástí instalace systému WEKA – například klasická ukázka klasifikace pro předpověď **hrát** či **nehrát** tenis podle typu počasí: **weather-nominal.arff**.

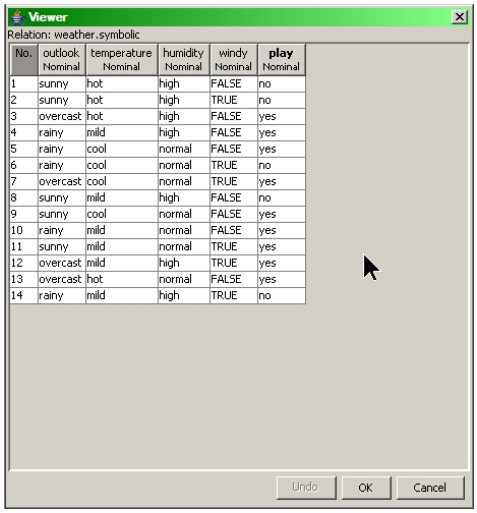
Formát **arff** je vlastní formát systému WEKA. WEKA dokáže přečíst i některé další formáty, např. **csv**, **c4.5**, aj.



Standardní instalace vytvoří adresář **data**, kde jsou instalovány některé klasické jednoduché příklady, na nichž si lze vyzkoušet funkčnost možností WEKA.

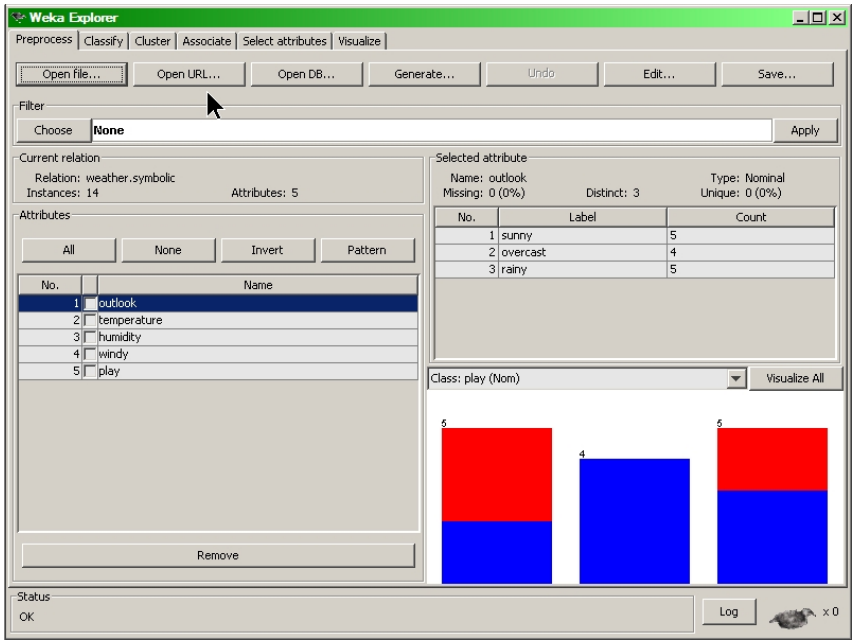


Data **weather.nominal.arff** obsahují 14 příkladů (pozorování situace během dvou týdnů) se známou klasifikací, kdy pro určitou kombinaci *nominálních* (nečíselných) hodnot atributů **předpověď počasí** (**slunečno**, **zataženo**, **deštivo**), **teplota** (**horko**, **mírně**, **chladno**), **vlhkost** (**vysoká, normální**) a **větrno** (**ano**, **ne**) někdo šel (**ano**) nebo nešel (**ne**) hrát tenis (klasifikační třída je konvenčně uváděna v posledním sloupci tabulky). Tabulku typu *spreadsheet* lze otevřít v editoru dat systému WEKA (menu **Edit**):

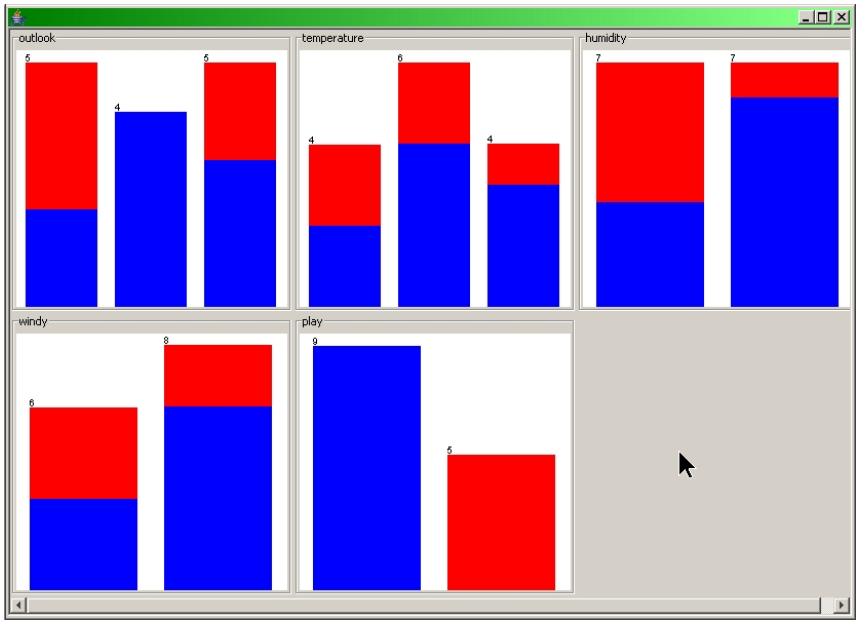


Celkem tedy existuje 36 možných kombinací vstupních hodnot atributů (3 x 3 x 2 x 2 = 36), což popisuje všechny možné situace. Známo jich je ale pouze 14. Klasifikaci pro zbývajících neznámých 16 kombinací vstupních atributů lze založit na znalosti známých příkladů (to jsou ty v tabulce) a tak odhadnout, zda v situaci doposud nepozorované půjde sledovaná osoba hrát nebo ne.

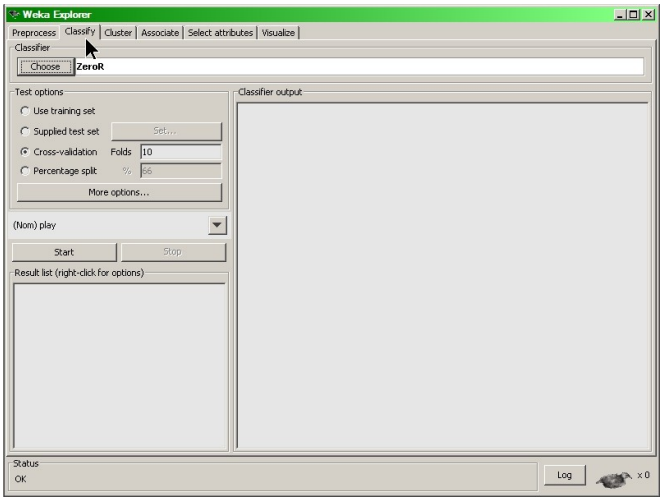
Tabulka představuje *trénovací příklady*. Zvolený klasifikační algoritmus si na základě jejich zpracování stanoví konkrétní hodnoty svých parametrů vzhledem k dané aplikaci (jít/nejít hrát tenis?). WEKA obsahuje mnoho klasifikačních algoritmů (obsahuje i jiné algoritmy, např. shlukovací, atd.). Otázka je, jaký algoritmus vybrat? Na to není obecně jednoznačná odpověď, často se musí hledat experimentováním a pak vybrat podle nejlepších výsledků. Po stanovení dat WEKA zobrazí základní statistické údaje a umožní případnou editaci:

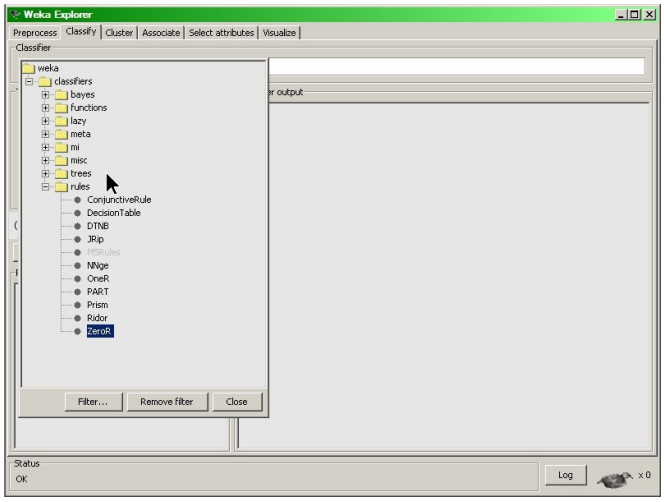


Případně po volbě **Visualize All** se objeví další okno:



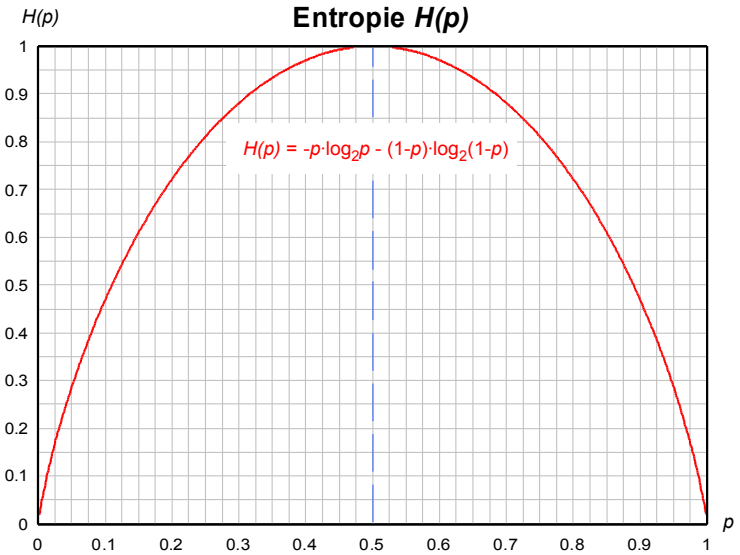
Menu **Classify** poskytuje možnost výběru z několika skupin různých klasifikátorů:



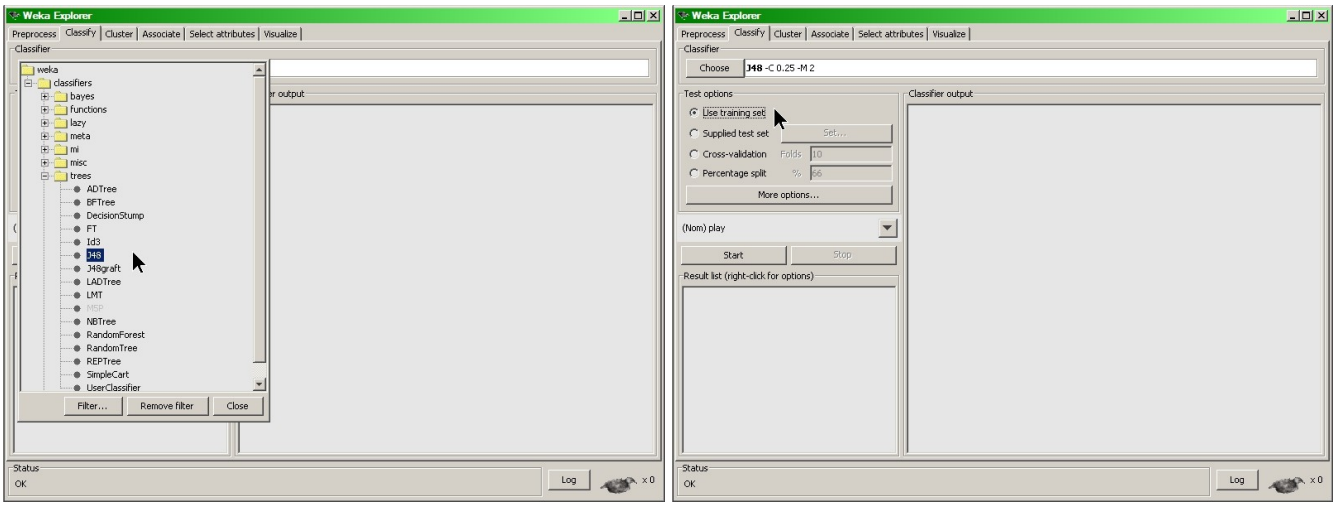


Pro demonstraci dobře poslouží rozhodovací strom **J48**, založený na minimalizaci entropie. Algoritmus **J48** je do jazyka Java převedený algoritmus **c4.5** ve verzi **c4.8**. Autorem (nekomerční verze) c4.5/c4.8 je Ross Quinlan z University of Sydney, Austrálie (Quinlan, J. Ross: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993). Více lze najít na jeho website <**http://www.rulequest.com/Personal/**> a na website jeho komerční verze **C5** (Unix) resp. **See5** (Windows) <**http://www.rulequest.com/**>, případně na detailnější stránce <**http://www.rulequest.com/see5-info.html**>.

Stručný princip c4.5/J48: Algoritmus generování rozhodovacího stromu z poskytnutých příkladů je založen na myšlence, že původní heterogenní množina příkladů se dá postupně rozdělit na homogennější podmnožiny. Homogennější množina má nižší entropii (chaos) než heterogennější – vychází se zde z informatického chápání pojmu *entropie*. Základem je stanovení pravděpodobnosti výběru prvku z množiny tak, aby patřil do určité třídy. Množina obsahující prvky výhradně jedné třídy je dokonale homogenní, tj. má nulovou entropii: není nutno provádět pokusy, protože pravděpodobnost = 1.0 (resp. 0.0 vzhledem k výběru prvku z jiné třídy). Naopak heterogenní množina obsahující směs prvků z více tříd má pravděpodobnost výběru < 1.0. Např. pro dvě třídy a stejný počet prvků v každé je pravděpodobnost p náhodného výběru pro určitou třídu = 0.5 (max. entropie = 1). Algoritmus hledá takový atribut, který rozdělí heterogenní množinu na homogennější podmnožiny s nejnižší možnou entropií ≥ 0.0. Rekursivně se postup opakuje tak dlouho, dokud lze entropii $H(p)$ zmenšovat. V nejhorším případě vzniknou jako listy stromu podmnožiny obsahující pouze jeden prvek (to je případ tzv. *přeučení se*).



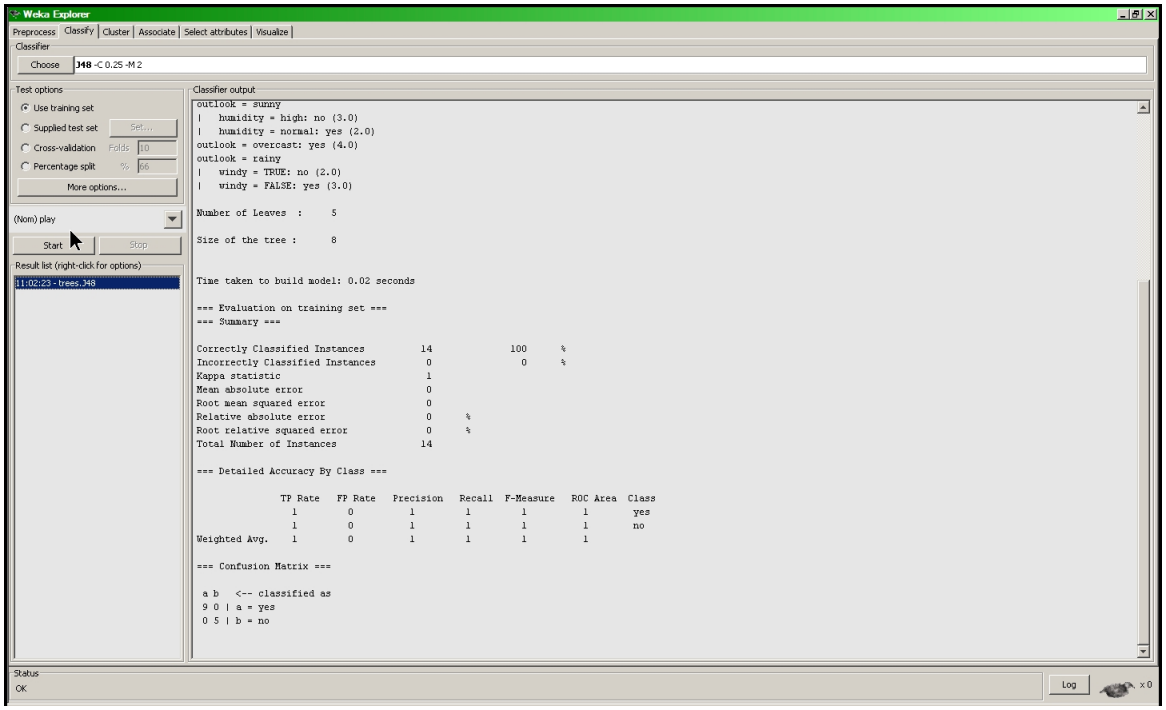
Volba algoritmu ve WEKA je jednoduchá. Menu **Choose**, pak se vybere skupina **trees** a v ní **J48**:



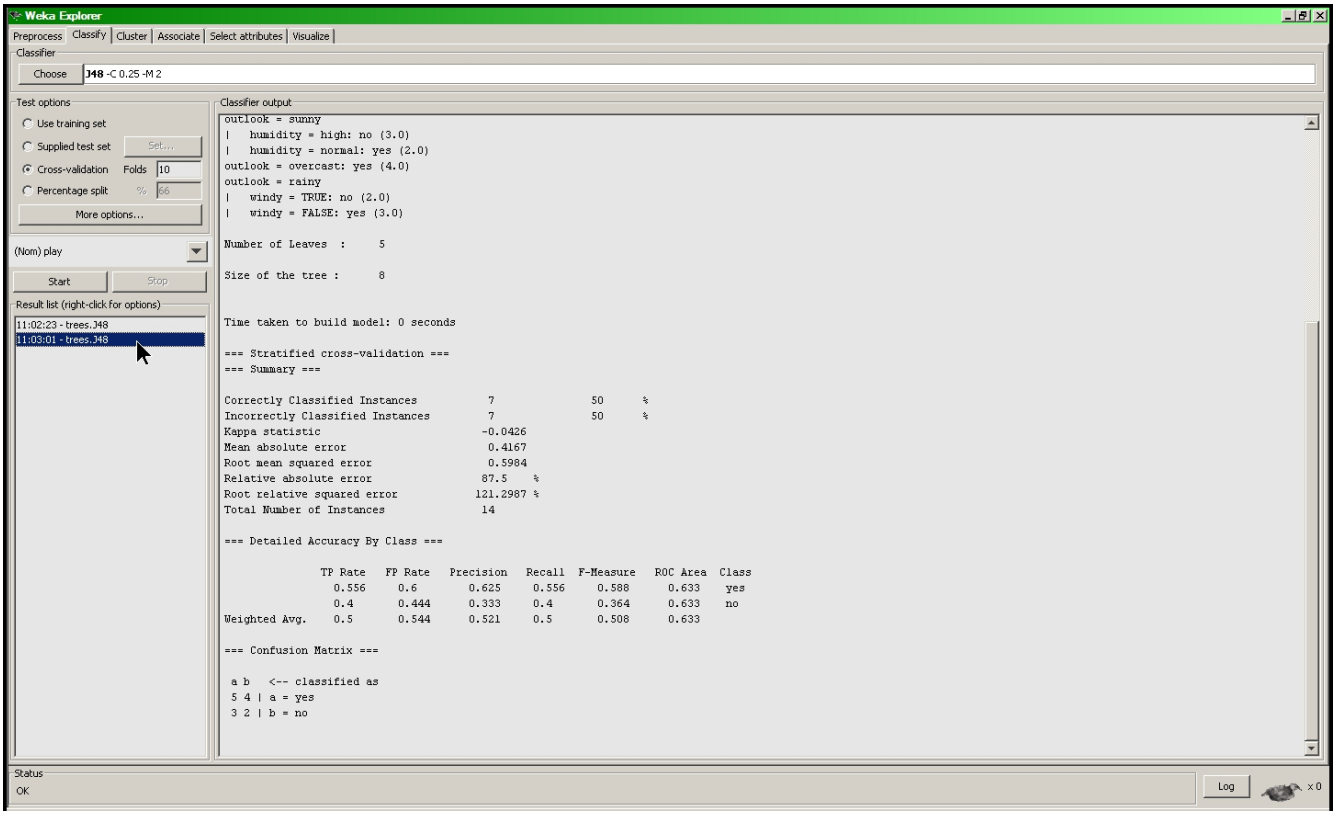
Za názvem algoritmu jsou jeho implicitní (default) parametry. Pro **J48** to je **-C 0.25** (tzv. *confidence factor*; rozhoduje o prořezání stromu, *pruning*, což je důležité vzhledem k získání co nejvyšší klasifikační obecnosti, aby se strom nenatrénoval výhradně na poskytnutné příklady a aby budoucí data klasifikoval co nejlépe) a **-M 2** (rozhoduje o *minimálním* počtu prvků v listu, aby se bylo možno vyhnout např. přeučení – zde dva). Pro případ s hraním tenisu lze parametry ponechat. Pozn.: kliknutím levým tlačítkem do řádku se jménem algoritmu lze vyvolat okno s detaily o algoritmu, kde lze nastavovat další parametry.

Nyní jsou známa trénovací data a algoritmus. Je dále zapotřebí stanovit způsob testování (správnost predikce pro případy, které nebyly použity pro trénování, např. jak správně bude strom klasifikovat budoucí data). Jednou z možností je použití tzv. *cross-validation* (nebo použití trénovacího souboru, *Use training set*; nebo vlastní množiny testovacích dat, *Supplied test set*).

Použití trénovacích dat: slouží pouze pro počáteční orientaci například toho, zda zvolený algoritmus je vůbec schopen daná data přijatelně zpracovat. Tlačítko Start spustí trénování. Výsledek je k dispozici v hlavním okně:

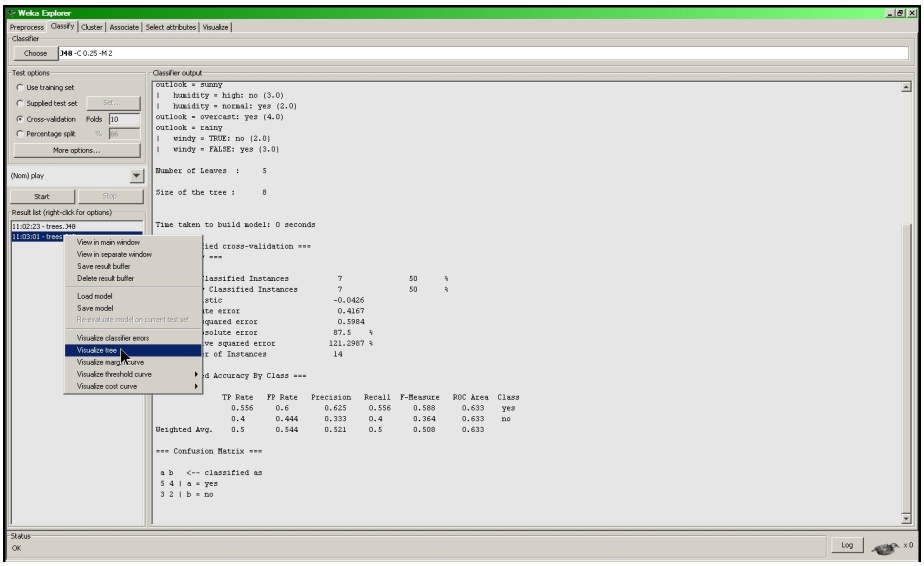


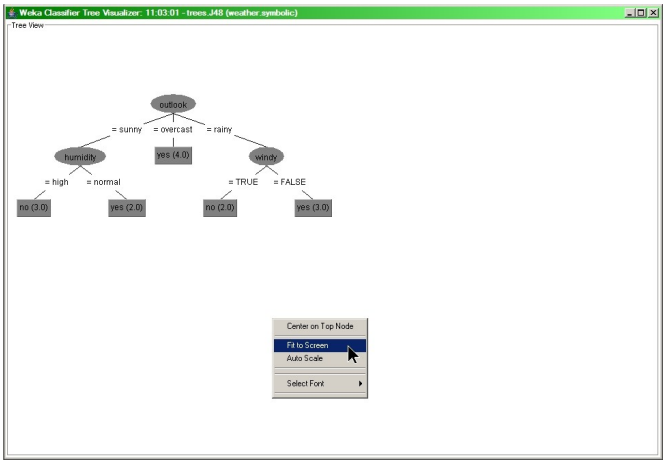
Trénovací data poskytla při testování nulovou chybu (100 % přesnost). Při volbě *cross-validation* lze určit, jaké procento dat se použije na trénování a jaké na testování – podle toho proběhne postupně příslušný počet trénování a testování. Předpokládaný výsledek je pak průměrem – ale vlastní algoritmus je pro použití vždy natrénován pomocí všech příkladů! Testování slouží jen pro odhad budoucí chyby:

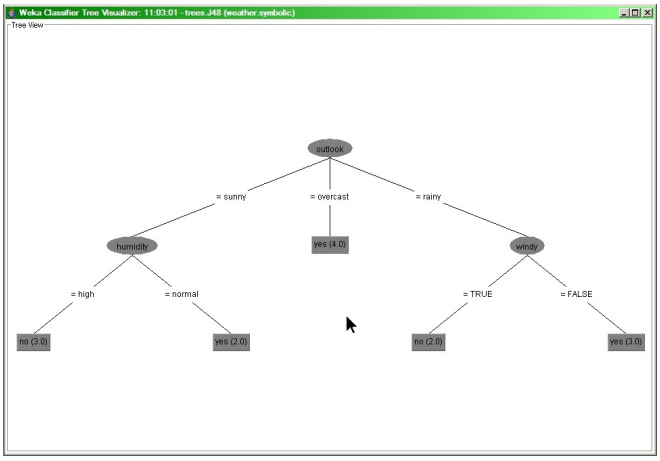


Zde tzv. *10-fold cross-validation* (náhodné rozdělení trénovacích dat na 10 částí, z nichž postupně vždy 9 se použije pro trénování a 1 pro testování) dala chybu 50 %. Je vidět velký rozdíl oproti testování pomocí pouze trénovacích dat. Příčinou zde je malý počet příkladů (jen 14), kdy odejmutí 1 nebo 2 příkladů zhorší výrazně podmínky učení. Lze vyzkoušet i jiné hodnoty než 10, např. 14 (tzv. *leave-one-out* metoda), nebo 3, apod.

J48 umožňuje zobrazit výsledný strom graficky (kliknutím pravým tlačítkem myši na řádek v **Result list**):







V elipsách jsou uzly rozhodující o směru postupu ve větvi, v obdélnících dole jsou listy. Číslo v listu, např. 3.0 znamená, kolik prvků je v této podmnožině obsaženo (ať už správně nebo nesprávně).

Většina algoritmů umí klasifikovat jen do nominálních tříd. Některé umějí i do numerických tříd (regrese), například M5P je rozhodovací regresní strom.

Na tatáž data lze takto vyzkoušet za stejných podmínek trénování a testování různé algoritmy a hledat optimální hodnoty parametrů – může to být zdlouhavý proces, který však nakonec najde nějaký výsledek. Po počátečním orientačním vyzkoušení algoritmů lze některé vybrat a pomocí skriptu (např. *batch* pro MS DOS) hledat nejlepší řešení spouštěním v cyklech s postupnou změnou hodnot parametrů, atd. Parametry jednotlivých algoritmů lze zjistit tak, že se spustí příslušný algoritmus bez parametrů, např.:

java weka.classifiers.trees.J48

(je nutné zajistit správné nastavení cest/*path*, nebo pracovat v příkazovém okně spuštěném v adresáři s WEKA; záleží také na tom, zda jsou použita MS Windows, nebo Linux, nebo MacIntosh). Příklad je uveden na konci.

Některé doplňující poznámky:

- Některé algoritmy pracují jen s nominálními atributy (např. ID3), jiné i s numerickými (J48 je rozšířený ID3).
- WEKA má v sobě zabudovaný systém náhrady chybějících dat (ne pro všechny algoritmy). Např. v editoru WEKA je chybějící hodnota zobrazena jako prázdné políčko v tabulce (pro příslušný *atribut* jako sloupec a *příklad* jako řádek).
- Pro statistické prokázání předpokládané chybovosti klasifikátoru je vhodné opakovat vícenásobně *cross-validation*, např. *10-times 10-fold cross-validation* tak, že náhodné rozdělení na 10 částí (*ten-folds*) se zopakuje desetkrát, ale pokaždé je nutno zadat jinou násadu generátoru pseudonáhodných čísel (tzv. *seed*, což se nastavuje v okénku parametrů algoritmu, které se v GUI vyvolá kliknutím levým tlačítkem na řádek s algoritmem).
- Výsledek trénování a testování z GUI okna lze zkopírovat přes *clipboard* tak, že se klikne do výsledkového okna a pak se provede *Ctrl-C*.
- WEKA poskytuje objevenou znalost ve formě, která odpovídá použité metodě a algoritmu. Mohou to být rozhodovací stromy, pravidla, natrénované algoritmy (třeba umělá neuronová síť), apod. Výsledek trénování lze uložit (nebo vyvolat) jako tzv. *model* (v menu, které se objeví po kliknutí pravým tlačítkem na řádek v okně **Result list**.
- Součástí instalace je i určitá dokumentace. Další informace lze hledat na website WEKA: <**http://www.cs.waikato.ac.nz/ml/weka/**>.

Příklad spuštění J48 v příkazovém okně MS Windows (a kopie výpisu běhu programu):

```
java weka.classifiers.trees.J48 -t "c:\Program Files\Weka-3-6\data\weather.nominal.arff"

J48 pruned tree
------------------

outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5

Size of the tree :      8


Time taken to build model: 0.02 seconds
Time taken to test model on training data: 0 seconds

=== Error on training data ===

Correctly Classified Instances          14               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               14


=== Confusion Matrix ===

 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no


=== Stratified cross-validation ===

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14


=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no


C:\Program Files\Weka-3-6\data>
```

(WEKA byl spuštěn z adresáře, kde byla data.)

■