ACRES 3 User Guide

Konstantinos Kovas
Department of Computer Engineering and Informatics
University of Patras
{kobas@ceid.upatras.gr}

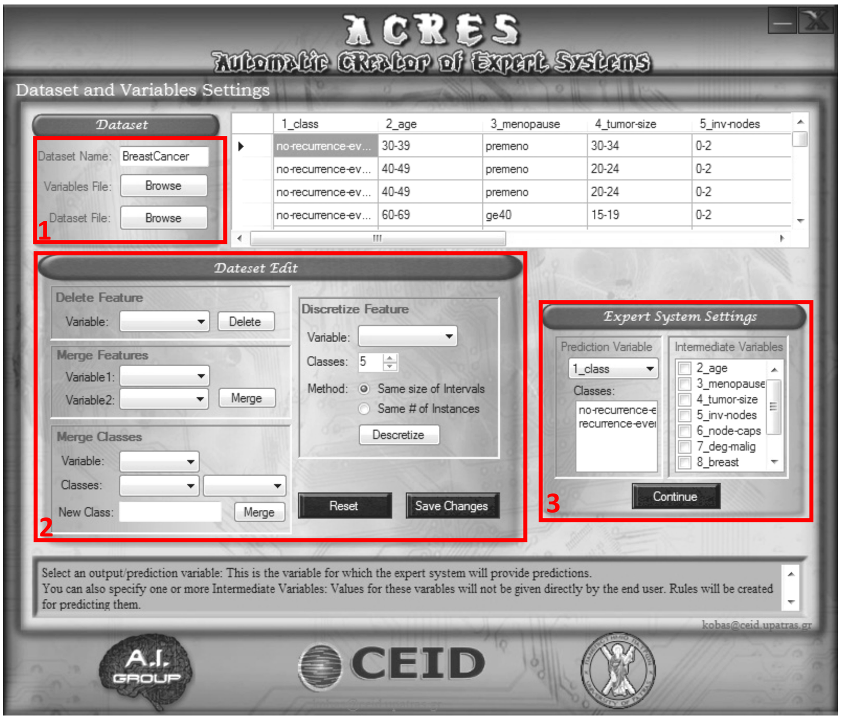
Version 3.0.3
12/06/2014

ACRES (Automatic Creator of Expert Systems) is a tool initially developed as way to test and compare different methods of combining Certainty Factors in expert systems. In its second version, we extended the architecture to apply for the problem of multiclass classification, but the overall architecture remained simple, focusing on the goal of comparing certainty factor combination methods.

The third version is our attempt towards a more generalized tool for generating expert systems. More specifically an extension of the system made it possible to generate classification rules for additional variables (apart from the output variable), for which the final user of the expert system cannot provide values. This gives the ability to design more complex rule hierarchies, which are represented in an easy-to-interpret tree structure. Feature ranking and subset selection techniques help achieve the generation task in a more automatic and efficient way. Other enhancements include the ability to produce expert systems that dynamically update the certainty factors in their rules, the generation of rules and functions for interaction with the end-user and a graphical interface for the produced expert system.

1. CREATE EXPERT SYSTEM

a. Dataset & Variables Settings



- Dataset Import (1)
- Dataset Edit (2)
- Variables (3)

Dataset Import

Dataset Name: Specify a name for the expert system that will be created.

Variables File: A file containing a name for each variable in the dataset.

```
1_class
2_age
3_menopause
4_tumor-size
5_inv-nodes
6_node-caps
7_deg-malig
8_breast
9_breast-quad
10_irradiat
```

example variables file

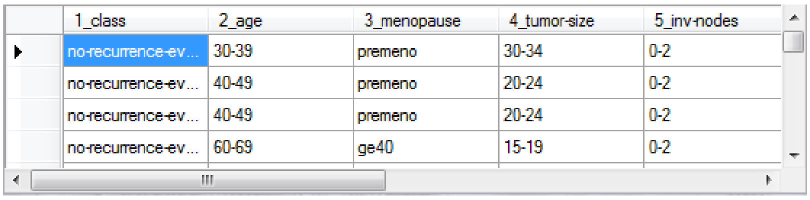
Dataset File: The dataset file containing known instances about a problem [comma delimited format].

```
no-recurrence-events,50-59,ge40,15-19,0-2,yes,2,left,central,yes
no-recurrence-events,50-59,premeno,25-29,0-2,no,1,left,left_low,no
no-recurrence-events,60-69,ge40,25-29,0-2,no,3,right,left_low,no
recurrence-events,50-59,premeno,15-19,0-2,no,2,left,left_low,no
recurrence-events,40-49,premeno,40-44,0-2,no,1,left,left_low,no
recurrence-events,50-59,ge40,35-39,0-2,no,2,left,left_low,no
recurrence-events,50-59,premeno,25-29,0-2,no,2,left,right_up,no
recurrence-events,30-39,premeno,0-4,0-2,no,2,right,central,no
recurrence-events,50-59,premeno,25-29,0-2,no,2,left,right_up,no
…
```

example dataset file

Dataset Edit (Optional)

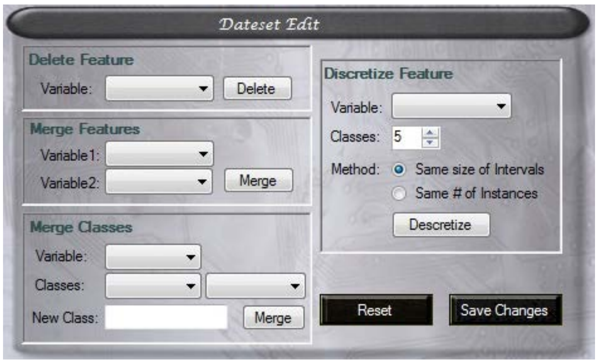
After importing the variables and dataset files the dataset is imported as a grid. The user can manually edit the values in the grid.



Εικόνα 1: Dataset as a grid

Additionally the user can perform the following operations:

- **Delete Variable:** Specify a variable and the corresponding column will be deleted.
- **Merge Variables:** Specify two variables. The corresponding cells will be merged. The values will be separated with '_'
- **Merge Classes:** Specify a variable and two of its classes. Then press 'Merge' to merge these classes as a new one with the name specified.
- **Descretize Variable:** Choose a variable with real values. Specify the number of classes and a discretization method.



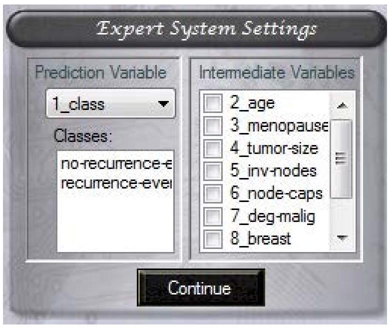
The **Reset button** will undo all changes made and reload the dataset you initially imported.

The **Save Changes** button will save all modifications as a new dataset file. You must manually edit the variables file if necessary. Then import both files again to continue with the expert system creation.

Variables

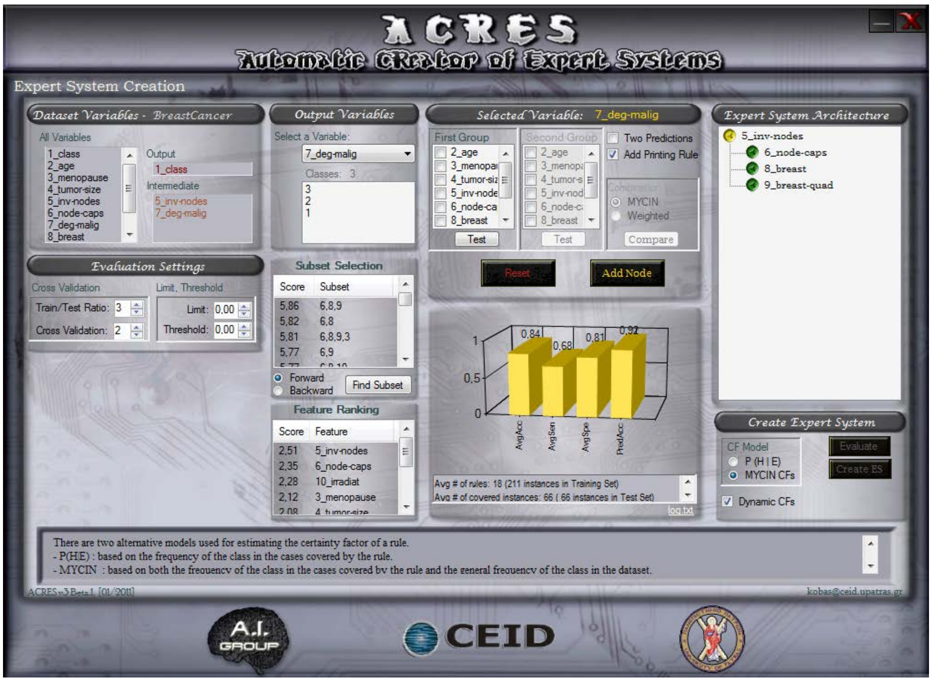
The user must specify an output/prediction variable. This is the variable for which the expert system will provide predictions.

Optional: You can also specify one or more Intermediate Variables: Values for these variables will not be given directly by the end user. Rules will be created for predicting them.



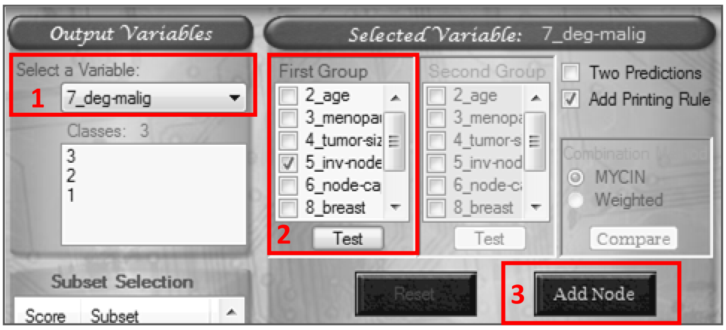
The Continue button loads the Expert System Creation frame.

b. Expert System Creation



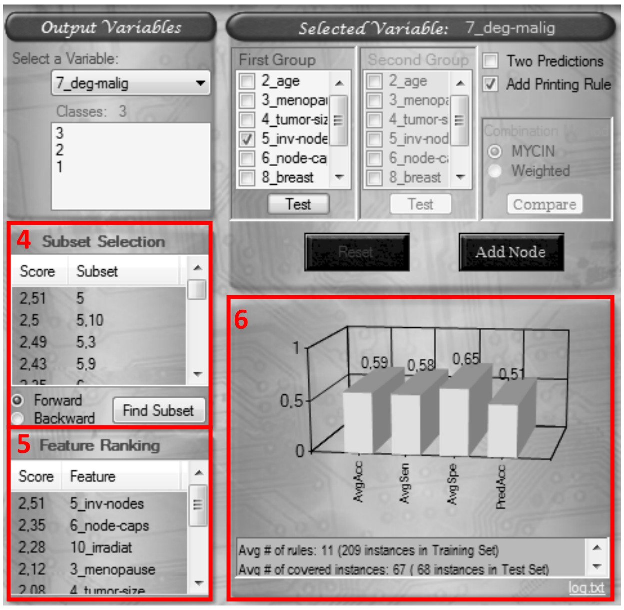
For each intermediate variable specified and then for the output variable:

- Select a variable (1).
- Specify a subset of variables for creating prediction rules (2)
- Add the variable as a node to the architecture Tree (3)



Εικόνα 2: Main Loop

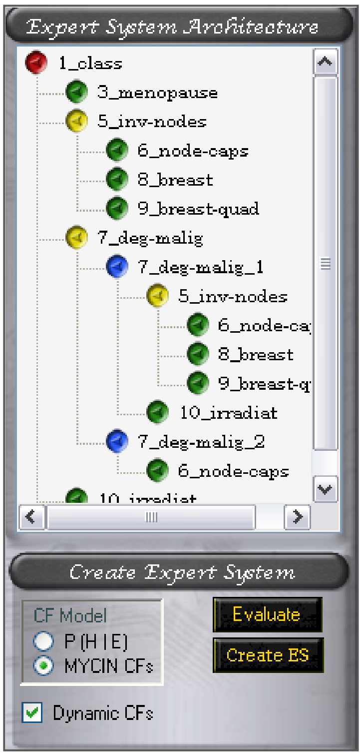
- <u>Optional:</u> A second subset can be specified by checking “Two Predictions”
    - o There are two alternative methods for combining these two predictions about the same conclusion: The method used in MYCIN (**MYCIN**) and a generalized version using weights (**WEIGHTED**).
- To help the user in choosing a subset, three facilities are offered:
    - o Feature Ranking -**5** (Automatically produced when selecting a variable)
    - o Subset Selection -**4** (by clicking “Find Subset”)
    - o Selected Subset Evaluation -**6** (by clicking “Test”)



Εικόνα 3: Facilities

When nodes for all intermediate and the output variables have been added, the expert system can be created.

- **“Create ES”** will create the expert system as a clips file
- **“Evaluate”** will create an expert system using a training set and evaluate it with a testing set.



Rule Generation and CF estimation

Given a variable for which we want predictions made and a subset of variables to be used for the prediction, we can generate a set of rules from a training set with the following steps:

1. Cluster instances in groups, so that each group contains instances that have identical values in the variables of the subset
2. From each such group produce one rule that has as conditions the common attribute-value pairs of the instances and as conclusion the possible classes of the output variable
3. Associate each possible class *i* with a certainty factor using the formula:

$$CF_i = n_i / N \tag{1}$$

Where n_i is the number of instances of class i in the group and N the number of all instances in it. That is, a CF for a class is defined as the frequency of the class in the group. It is obvious that the certainty factor would be a value between 0 and 1. We can easily convert this value in the interval [-1,1] with the formula:

$$CF = 2*CF - 1 \tag{2}$$

We give a simple example of a rule created with this method:

```
(defrule group_1_class_16
 (declare (salience 70))
 (data
  (3_menopause ge40)
  (5_inv-nodes 3-5)
  (7_deg-malig 3)
  )
=>
 (assert (1_class (no-recurrence -0.64)    (recurrence 0.64)))
)
```

A simple example of a generated rule

Certainty Factor Combination

If we repeat the above procedure more than one time, for different set of variables, we can create a rule set that given a new instance can provide more than one conclusions about the output variable. According to the model of certainty factors used in MYCIN, two certainty factors about the same fact can be combined using suitable formulas depending on the signs of the certainty factors combined.

For example, if we have two rules with the same conclusion and CF1, CF2 respectively the certainty factors associated with them, and they are both positive numbers, the combined certainty factor CF for conclusion, according to MYCIN theory, is given by the formula:

$$CF = CF1 + CF2\ (1\text{-}CF1) = CF1 + CF2 - CF1 * CF2 \tag{3}$$

In the expert system PASS [4], the remark was made that in formula 3 both certainty factors contribute equally to the final result. In practice, rules are often not equally reliable since their certainty factors are either bound to an expert's judgment or based on data containing noise, so they proposed a generalized version of the formula (1):

$$CF = w1 * CF1 + w2 * CF2 + w * CF1\ CF2 \tag{4}$$

where w1, w2 and w are numeric weights that should satisfy the following equation:

$$w1 + w2 + w = 1 \tag{5}$$

to assure that $0 \leq CF \leq 1$.

To use formula (4) however, the weights w_1, w_2, w should be first determined. In PASS, statistical data about the problem were used, as a training data set to determine the weights by hand.

In ACRES we offer both combination methods when multiple rule sets are specified for the output variable. The system produces the necessary weights for the generalized formula automatically, utilizing a genetic algorithm to search the space of possible weight combinations for an optimum one.

CF Models

The system offers two alternative methods for estimating Certainty Factors. Consider an output variable *C* associated with *n* possible classes $C_{i..n}$ and a dataset *N* containing |*N*| instances. Evidence *E* is a certain pattern of values for a set of variables of the dataset and *D* is the set of instances in the dataset that this pattern occurs. We represent the absolute frequency of class C_i in *D* as $f(C_i,D)$ and the absolute frequency of class C_i in *N* as $f(C_i,N)$.

P(H|E):

Our initial approach used in previous versions relied solely on the probability found from the frequency of a class in *D*. For a class C_i the certainty factor is estimated using the conditional probability that an instance is classified in class C_i , given that evidence *E* is true.

$$P(Ci \mid E) = \frac{f(Ci, D)}{|D|}$$

Obviously the above value would be between 0 and 1, so we use the following formula to produce a value in the interval [-1 , 1].

$$CF(Ci, E) = 2 \times P(Ci \mid E) - 1$$

MYCIN CFs:

An alternative method added in the new version combines the above probability with the a priori probability found from the general frequency of class C_i in the entire dataset. This probability can be easily computed following the formula:

$$P(Ci) = \frac{f(Ci, N)}{|N|}$$

Using the definition of certainty factors in the expert system MYCIN we can combine these two probabilities to produce the measures of Belief $MB(C_i, E)$ and Disbelief $MD(C_i, E)$.

$$MB(Ci, E) = 1 \qquad if\ P(Ci) = 1$$

$$MB(Ci, E) = \max\{0, \frac{P(Ci|E) - P(Ci)}{1 - P(Ci)}\} \qquad \text{otherwise}$$

$$MD(Ci, E) = 1 \qquad if\ P(Ci) = 0$$

$$MD(Ci, E) = \max\{0, \frac{P(Ci) - P(Ci|E)}{P(Ci)}\} \qquad \text{otherwise}$$

Finally we can estimate the Certainty Factor using these measures of Belief and Disbelief:

$$CF(Ci,E) = \frac{MB(Ci,E) - MD\ (Ci,E)}{1 - \min\{MB(Ci,E), MD(Ci,E)}$$

It is important to point out the underlying characteristic of this method which is that the certainty factor produced is not a measure of our confidence in C_i, but rather a measure of the change of our confidence in C_i, given the evidence E. This means that a positive value represents an increase of our confidence, whereas a negative value represents a decrease of our confidence.

Dynamic CFs

Another new feature is the ability to generate expert systems that can update the Certainty Facts of their rules when new instances of the problem become available. To accomplish this, the certainty factors are not hard coded inside the generated rules, but are instead dynamically computed at run time. The required frequencies for computing the certainty factor are saved in a separate file in the form of CLIPS facts. Thus, for each prediction rule of the expert system, there is a corresponding rule that updates the corresponding frequencies. The expert system consists of two files. The main expert system that remains constant contains all the rules, functions and templates. The secondary file contains facts that store the frequencies required in the rules for computing the certainty factors which can change during runtime if the end user provides new instances.

Evaluation

Evaluation Metrics for Classification Problems

Evaluation of a classification model is usually based on the following metrics: accuracy, precision, sensitivity and specificity, which for two classes (positive and negative) are defined as follows:

$$acc = \frac{TP + TN}{TP + FP + FN + TN}, \qquad prec = \frac{TP}{TP + FP}, \qquad sen = \frac{TP}{TP + FN},$$

$$spec = \frac{TN}{TN + FP}$$

where, TP is the number of cases classified correctly as positive, FP is the number of cases that were incorrectly classified as positive, TN is the number of cases correctly classified as not positive and FN is the number of cases that are incorrectly classified as not positive.

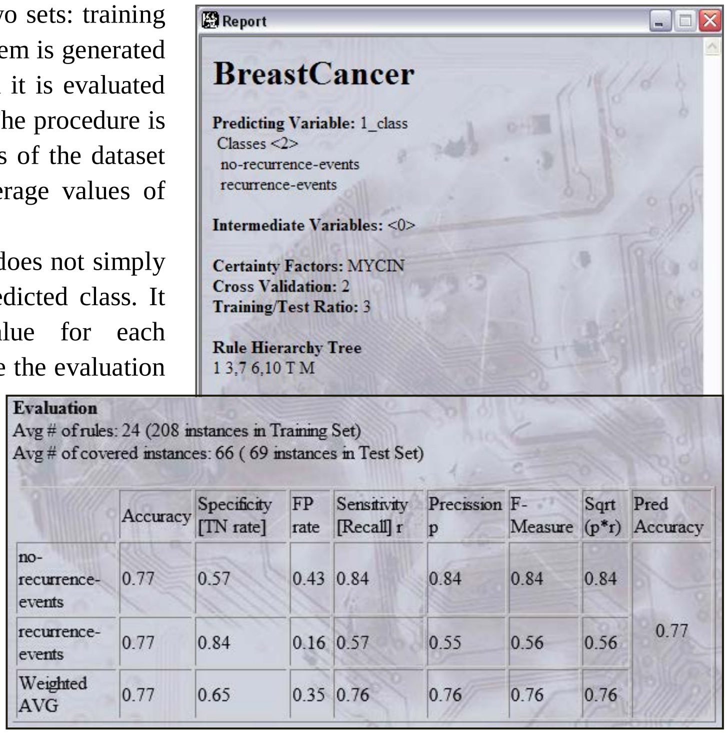
In case of more than two classes, one can view each class as a separate binary classification problem where positive are the cases of that class, whereas negative are the cases of all other classes. This way one can produce a confusion matrix for each class. Unlike the binary classification problem, with this approach a correctly classified case as negative does not necessary mean that the case was classified to the correct class. For this reason the value of TN is not credible, and therefore cannot be used for estimating evaluation metrics. The metrics used are **Precision** (as defined above) and **Recall** (corresponds to Sensitivity). For a possible class A, Precision is the fraction of instances that were classified to class A, that actually belong to that class, while Recall is the fraction of instances that belong to class A that were correctly classified to that class. Since TP+FN is the sum of all cases that truly belong to the positive class, the Recall metric is also referred to as TP rate. Another useful metric is the F-measure combining the recall and precision values.

$$recall = \frac{TP}{TP + FN}, \quad precision = \frac{TP}{TP + FP}, \quad F\_measure = \frac{2 \times precision \times recall}{precision + recall}$$

Finally the weighted average of these metrics for all classes can be calculated, taking into account the number of occurrences of each class in the dataset.

These metrics are widely used in classification performance evaluations and corresponding tools like the data mining tool Weka so using them allows the direct comparison with various classification models.

Evaluation Report in ACRES


Report

BreastCancer

Predicting Variable: 1_class
Classes <2>
no-recurrence-events
recurrence-events

Intermediate Variables: <0>

Certainty Factors: MYCIN
Cross Validation: 2
Training/Test Ratio: 3

Rule Hierarchy Tree
1 3,7 6,10 T M

Evaluation
Avg # of rules: 24 (208 instances in Training Set)
Avg # of covered instances: 66 (69 instances in Test Set)

	Accuracy	Specificity [TN rate]	FP rate	Sensitivity [Recall] r	Precission p	F-Measure	Sqrt (p*r)	Pred Accuracy
no-recurrence-events	0.77	0.57	0.43	0.84	0.84	0.84	0.84	0.77
recurrence-events	0.77	0.84	0.16	0.57	0.55	0.56	0.56	
Weighted AVG	0.77	0.65	0.35	0.76	0.76	0.76	0.76	

The dataset is partitioned in two sets: training and testing set. The expert system is generated using the training set and then it is evaluated on the data of the testing set. The procedure is repeated for different partitions of the dataset (cross validation) and the average values of the metrics are presented.

The produced expert system does not simply classify an instance to the predicted class. It provides an uncertainty value for each possible class. In order to make the evaluation of a produced expert system easier, we consider that the system classifies an instance to the class for which the uncertainty factor is the highest.

As described above, to evaluate an expert system for more than two classes, we evaluate the performance for each class separately. For each class *i*, we treat the problem as binary, with the first class being *i* and the second class being a class consisting of all other classes. We then form a confusion matrix and compute the metrics for each class *i*.

We are mostly interested in the *Sensitivity* and *Precision* metrics**.** We also combine these two metrics producing their mean proportional, SQRT(p*r) as a more general metric and the *F-Measure* metric that we defined previously.

For a measure of the general classifying performance of the expert system we use the *Predictive Accuracy* metric which shows the percentage of instances in the testing set that were correctly classified to the class they belong.

It is common when the rules produced are very specific (having many conditions) that some (or even all) instances of a class in the testing set cannot be classified (are not covered by the rules). This can result in zero value for TP, FN and FP which results in zero value in the denominator of some of the above metrics. In these cases we cannot calculate the metrics for that class and the system informs the user accordingly.

Confusion Matrix Results [1]
c1 TP:9 FP:2 FN:2 TN:6
c2 TP:6 FP:2 FN:2 TN:9
c3 TP:0 FP:0 FN:0 TN:19

Confusion Matrix Results [2]
c1 TP:9 FP:2 FN:0 TN:8
c2 TP:8 FP:0 FN:2 TN:9
c3 TP:0 FP:0 FN:0 TN:19

Evaluation
Avg # of rules: 57 (101 instances in Training Set)
Avg # of covered instances: 19 (31 instances in Test Set)

	Accuracy	Specificity [TN rate]	FP rate	Sensitivity [Recall] r	Precission p	F-Measure	Sqrt (p*r)	Pred Accuracy
c1	0.84	0.78	0.23	0.91	0.82	0.86	0.86	
c2	0.84	0.91	0.09	0.78	0.88	0.83	0.82	
c3	1	1	0	*	**	***	***	0.84
Weighted AVG	0.88	0.88	0.12	*	**	***	***	

* For class c3, Sensitivity denominator TP+FN was always 0 so it could not be calculated.
** For class c3, Precission denominator TP+FP was always 0 so it could not be calculated.
*** Both Precission and Sensitivity are necessary to calculate these values.
-- Most common problem is that the produced rules are very specific and cannot cover any case of one of the classes (check the confusion matrix above).
-- **Try selecting fewer variables in the group(s), to produce more general rules.**

2. EXPERT SYSTEM INTERFACE

- Browse for an expert system file previously created by ACRES
- Give a value for each input variable and assert the fact to get a prediction.