Table of contents

INTRODUCTION

KEEL is a software tool developed to build and use different Data Mining models. We would like to remark that this is the first software tool of this type containing a free code Java library of Evolutionary Learning Algorithms. The main features of KEEL are:

- It contains pre-processing algorithms: transformation, discretization, instance selections and feature selections.
- It also contains a Knowledge Extraction Algorithms Library, supervised and unsupervised, remarking the incorporation of multiple evolutionary learning algorithms.
- It has a statistical analysis library to analyze algorithms.
- It contains an user-friendly interface, oriented to the analysis of algorithms.
- KEEL's environment can connect to Internet to download new data files for using them in future analysis.

We can distinguish three parts in the graphic environment:

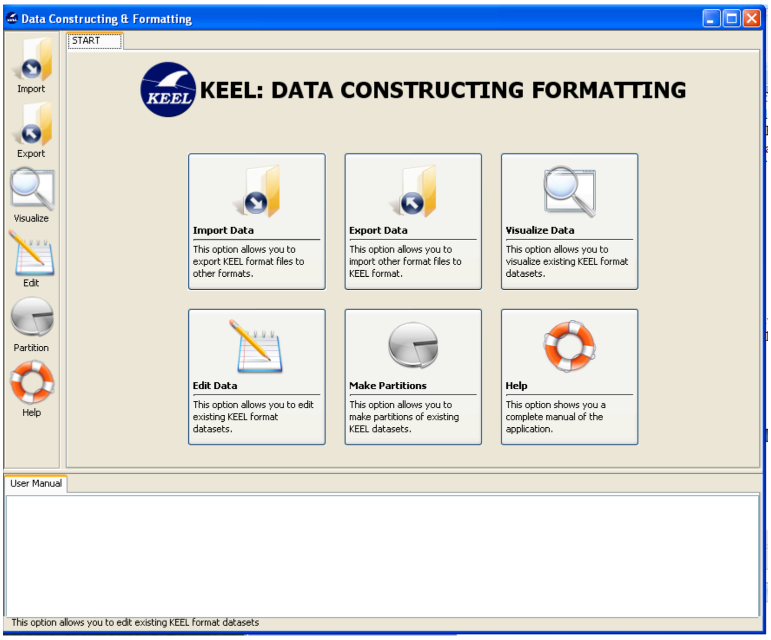
- The Preparation of the Data Bases part allows users to create different partitions of his own data bases or the data bases available in the KEEL web. Also, it is possible to edit, apply transformations, generate DataSets in the correct format from C4.5 files or view detailed plots about a concrete DataSet.
- The Design of Experiments part has the objective of designing the desired experiments using a graphical interface. After the experiment is designed, the interface generates a .ZIP file containing a directory structure with all the necessary files needed to run those experiments in the local computer.

  The interface also allows the user to add its own algorithms to the experimentation being designed. The only requirement is to accept the input file format of KEEL. Even, it is not needed to use the Java language for the own algorithms of the user. This provides a very flexible way for the user to compare its own methods with the ones in KEEL.
- The Generation of Evolutionary Algorithms with the JCLEC library allows the user to create his own evolutionary algorithms using a graphical interface. In this version of KEEL, this part is NOT implemented.

DATA PREPARATION

The next tasks are possible to carry out in this section:

- **Visualize Data:** This option allows you to visualize existing KEEL format datasets.
- **Import Data:** This option allows you to export KEEL format files to other formats.
- **Export Data:** This option allows you to import other format files to KEEL format.
- **Make Partitions:** This option allows you to make partitions of existing KEEL datasets.
- **Edit Data:** This option allows you to edit existing KEEL format datasets.


Data Constructing & Formatting
START
Import
Export
Visualize
Edit
Partition
Help
KEEL
KEEL: DATA CONSTRUCTING FORMATTING
User Manual

Import Data
This option allows you to export KEEL format files to other formats.

Export Data
This option allows you to import other format files to KEEL format.

Visualize Data
This option allows you to visualize existing KEEL format datasets.

Edit Data
This option allows you to edit existing KEEL format datasets.

Make Partitions
This option allows you to make partitions of existing KEEL datasets.

Help
This option shows you a complete manual of the application.

This option allows you to edit existing KEEL format datasets

Visualize Data

Visualize option allows you to view detailed information about an existing KEEL format data set. There are different options to show the information, you can see the content of data set, specific information about the attributes and to compare by means of charts two attributes.

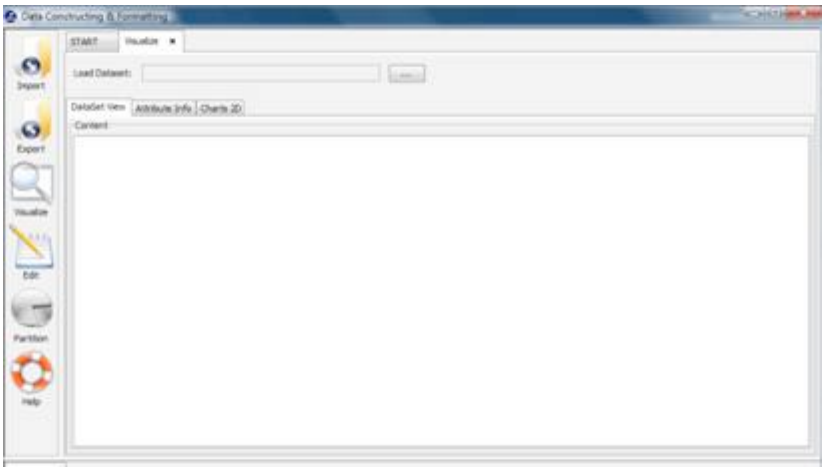
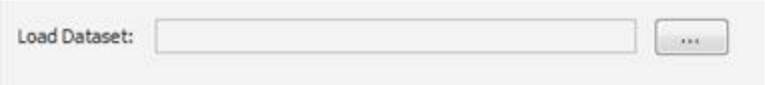


Figure 1. Process of visualizing data.

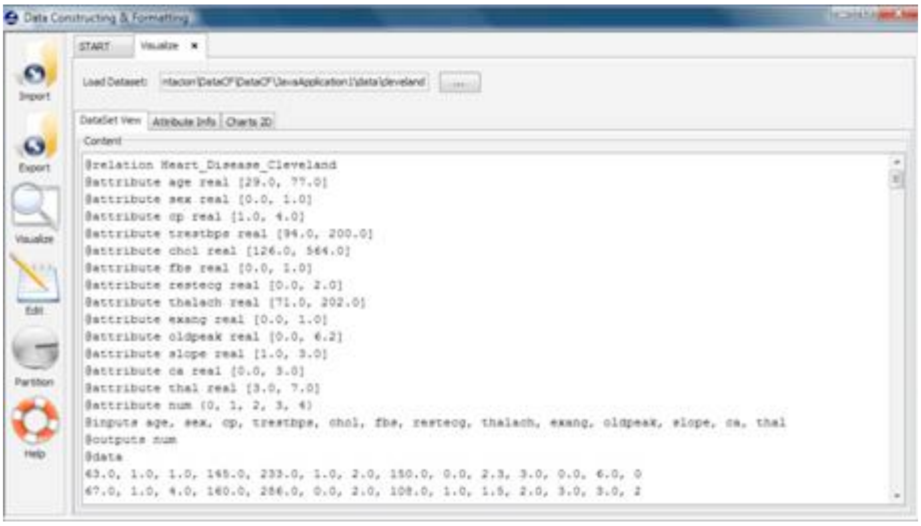
Figure 1 shows the main window of this option. First of all, you must select the path of source data set (in KEEL format) that you want to visualize (see Figure 2). When the file is loaded, different information about the data set is shown according to the option selected.



Load Dataset:

Figure 2. Load Dataset to view its content.

View Dataset. If you select to visualize this information, you can visualize the content of data set selected. The information cannot be modified; you only can visualize it (see Figure 3).

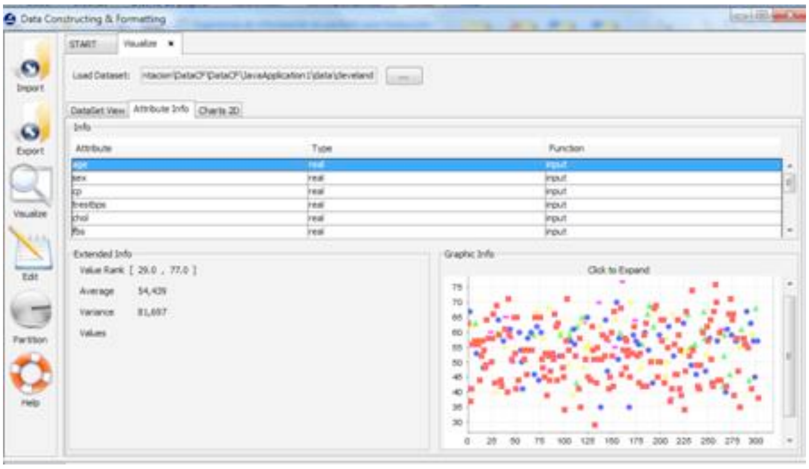


Load Dataset: ntacion\DataCF\DataCF\JavaApplication1\data\cleveland

```
@relation Heart_Disease_Cleveland
@attribute age real [29.0, 77.0]
@attribute sex real [0.0, 1.0]
@attribute cp real [1.0, 4.0]
@attribute trestbps real [94.0, 200.0]
@attribute chol real [126.0, 564.0]
@attribute fbs real [0.0, 1.0]
@attribute restecg real [0.0, 2.0]
@attribute thalach real [71.0, 202.0]
@attribute exang real [0.0, 1.0]
@attribute oldpeak real [0.0, 6.2]
@attribute slope real [1.0, 3.0]
@attribute ca real [0.0, 3.0]
@attribute thal real [3.0, 7.0]
@attribute num {0, 1, 2, 3, 4}
@inputs age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal
@outputs num
@data
63.0, 1.0, 1.0, 145.0, 233.0, 1.0, 2.0, 150.0, 0.0, 2.3, 3.0, 0.0, 6.0, 0
67.0, 1.0, 4.0, 160.0, 286.0, 0.0, 2.0, 108.0, 1.0, 1.5, 2.0, 3.0, 3.0, 2
```

Figure 3. Visualize the content of data set.

Attribute Info. In this option, you can obtain detailed information about the attributes defined in the data set. The information showed at the top of the windows is: attribute's type (Integer, Real or Nominal) and if the attribute is input or output. More information appears at the bottom left side of the window. The information showed depends on the attribute type, in the case of integer or real attribute the values of rank, average and variance are shown. In the case of Nominal attribute, you only view its possible values. Finally, on the bottom right side of the window, a chart with the distribution of the attribute's values is shown too. Figure 4 show the information of an attribute real and Figure 5 shows the information of a nominal attribute.



Load Dataset: ntacion\DataCF\DataCF\JavaApplication1\data\cleveland

Attribute	Type	Function
age	real	input
sex	real	input
cp	real	input
trestbps	real	input
chol	real	input
fbs	real	input

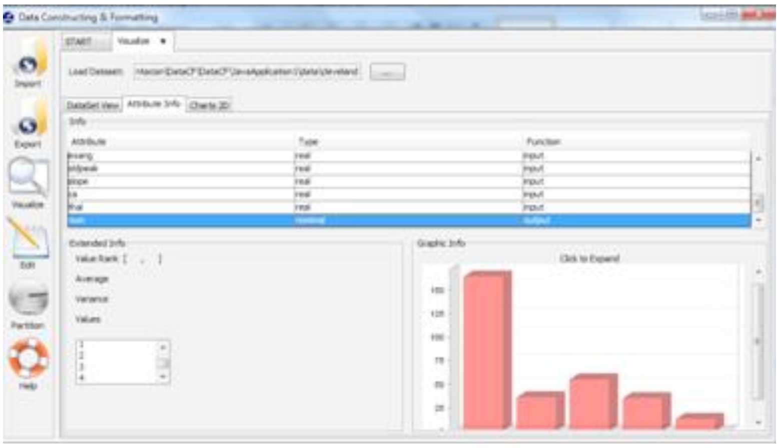
Value Rank [29.0 , 77.0]

Average 54,439

Variance 81,697

Values

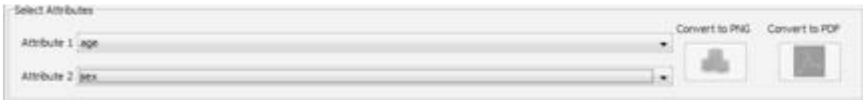
Figure 4. General and graphic information about real attribute.



Attribute	Type	Function
exang	real	input
oldpeak	real	input
slope	real	input
ca	real	input
thal	real	input
num	nominal	output

Figure 5. General and graphic information about nominal attribute.

Charts 2D. In this option, you can compare different attributes. First, you have to select the two attributes that you want to compare, for each attribute you have available a list with all attribute of data set (see Figure 6) .



Attribute 1: age

Attribute 2: sex

Figure 6. Select attributes to compare.

Once the attributes are selected, you have to click the "View chart" button and a graphic is shown. If you need to include the generated chart in other document, you can use the buttons: "Convert to PNG": this option saves the graph as a PNG image and "Conver to PDF": this option saves the chart as a PDF document.

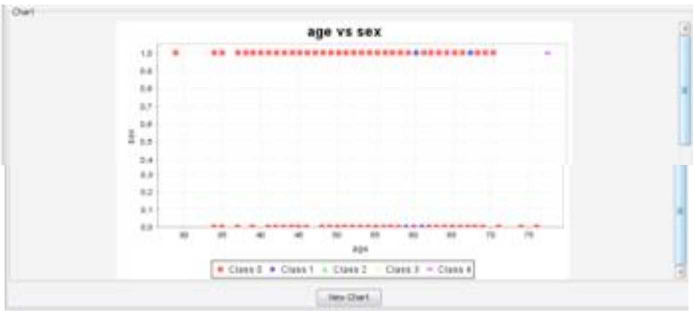


Figure 7. Chart to compare the two attribute selected

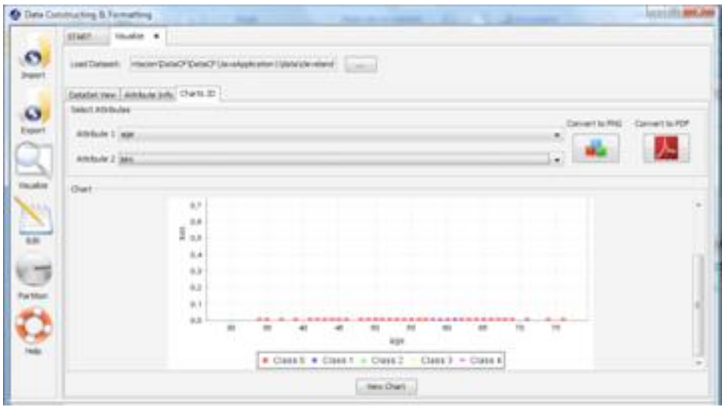
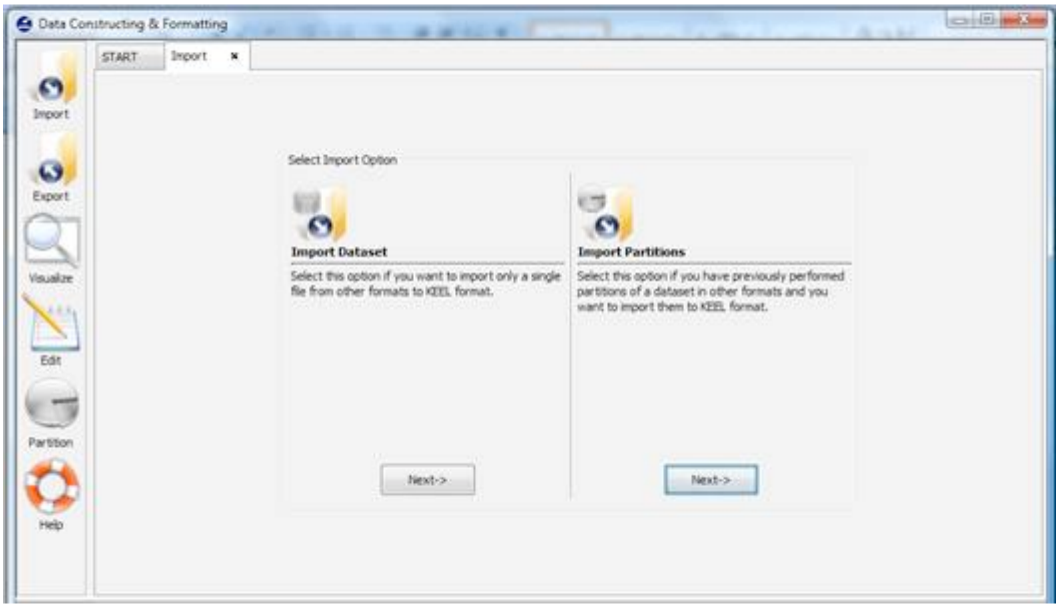


Figure 8. Visualize Charts 2D

Data import

Import option allows you to transform your files in different formats (txt, excel, xml, etc.) to KEEL format. Notice that if you want to use your own Data Sets, the design of the experiments will only use Data Sets according to KEEL format, therefore a previous step of import will be required.



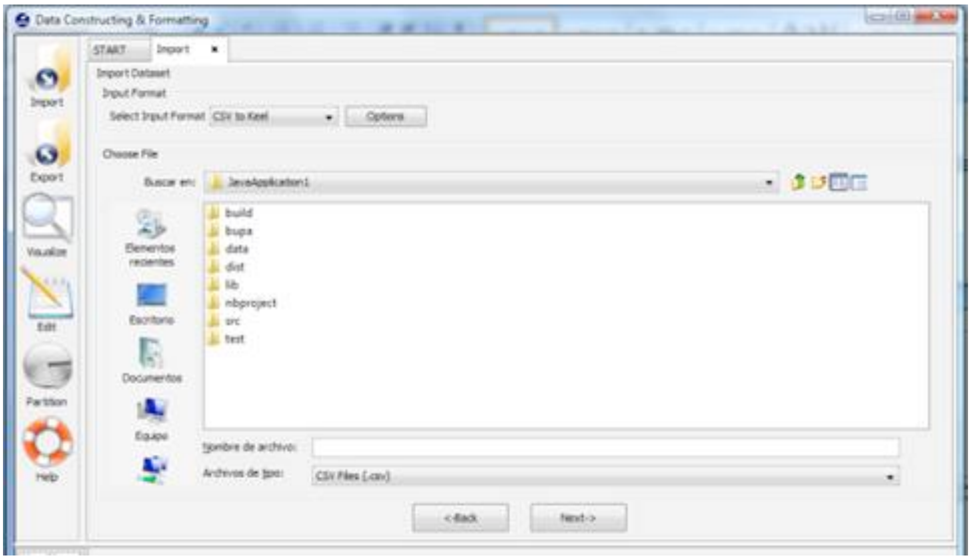
Import Dataset: Select this option if you want to import only a single file from other formats to KEEL format.

Import Partitions: Select this option if you have previously performed partitions of a dataset in other formats and you want to import them to KEEL format.

Figure 1. Two possibilities to carry out the import process.

Figure 1 shows the two possible options to import data sets. One option consists of importing one data set, the other option consists of importing a set of partitions which you have available in other formats different to KEEL format. In continuation, we show the process of both options.

1. Import Dataset. Select this option if you want to import only a single file from other formats to KEEL format. Figure 2 shows the window to this option.



Select Input Format: CSV to Keel

Buscar en: JavaApplication1

Nombre de archivo:

Archivos de tipo: CSV Files (.csv)

Figure 2. Import Dataset option.

To import a data set, it is necessary the next parts:

Step1. **Select Input Format**. First of all, you must select the source file format of the dataset. The format admitted are CVS, TXT, PRN, C4.5, Excel, Dif, PropertyList and Weka. The different options are shown in Figure 3.

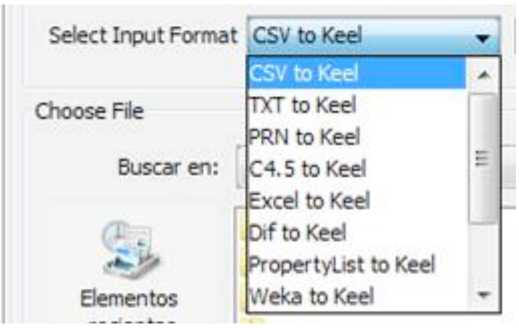


Figure 3. Format admitted to convert to KEEL format.

The "Options" button allows you to configure if it is necessary a certain separator and null value used in the source file.

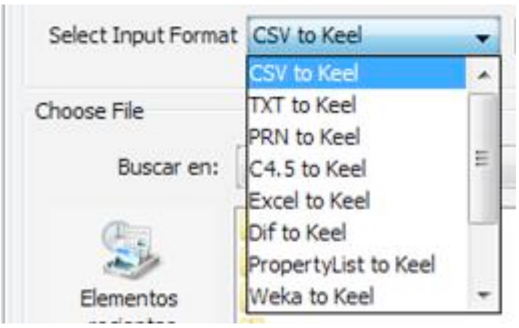
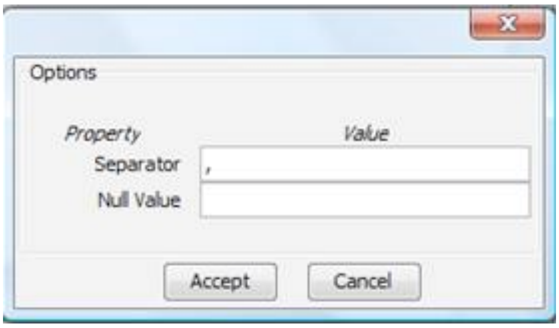


Figure 4. Specify certain options of format of source file.

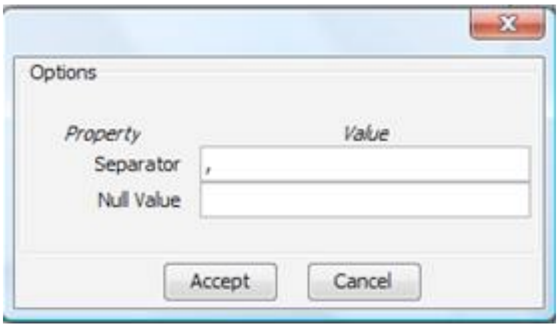
Step2. Select the source file. After specifying the file format used in source file, the path of this file must be specified (a browser commonly known from many other GUI programms is used to define this path).



Property	Value
Separator	,
Null Value	

Figure 5. Specify the source file path**.**

Step 3. Save the files. Once the type of conversion and the source file have been configured, you must click *Next* button and then, the original and the imported file are shown (see Figure 6).



Separator ,

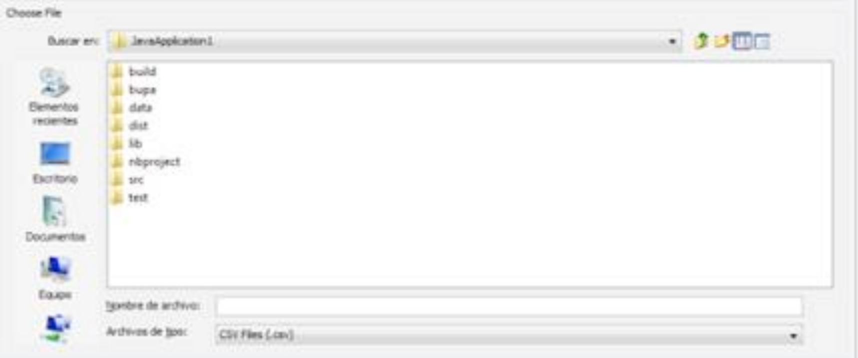
Null Value

Figure 6. The original and imported file.

If we agree with the conversion done, there are two options to save the imported file:

1. Select the *Import to the experiments section*: if you mark this option and click the *Save* button, the data set converted will be included as option in the KEEL experiments. This data set will be available to execute with the methods of KEEL.
2. Not select the *Import to the experiments section*: if you do not select this option, when you click the Save button, you have to select the destination directory for the transformed data set.

2. Import Partitions. Select this option if you have previously performed partitions of a dataset in other formats and you want to import them to KEEL format. This option allows to select a set of training and test files separately. Figure 7 shows the window with respect to this option.



Buscar en: JavaApplication1

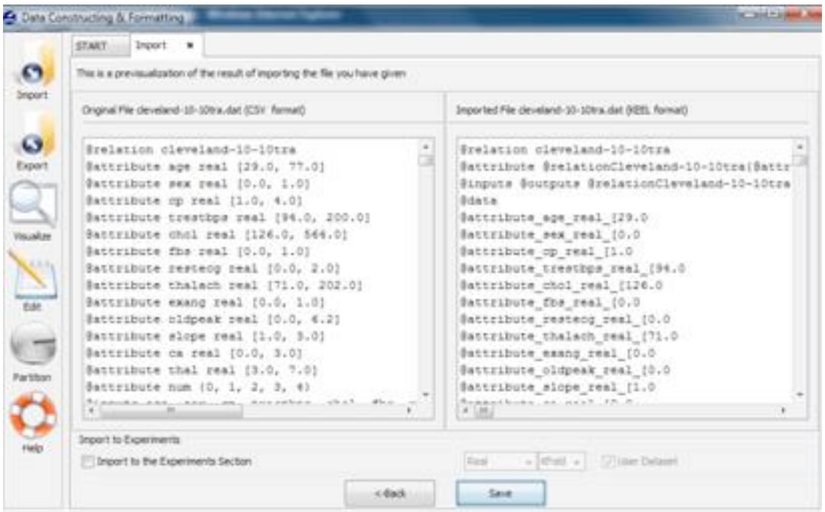
Nombre de archivo:

Archivos de tipo: CSV Files (.csv)

Figure 7. Import Partitions option.

To import partitions, it is necessary the next parts:

Step1. Select Input Format. First of all, you must select the source file format of the dataset. The formats admitted are CVS, TXT, PRN, C4.5, Excel, Dif, PropertyList and Weka. The different options are shown in Figure 8.



This is a previsualization of the result of importing the file you have given

Original File cleveland-10-10tra.dat (CSV format)

```
@relation cleveland-10-10tra
@attribute age real [29.0, 77.0]
@attribute sex real [0.0, 1.0]
@attribute cp real [1.0, 4.0]
@attribute trestbps real [94.0, 200.0]
@attribute chol real [126.0, 564.0]
@attribute fbs real [0.0, 1.0]
@attribute restecg real [0.0, 2.0]
@attribute thalach real [71.0, 202.0]
@attribute exang real [0.0, 1.0]
@attribute oldpeak real [0.0, 6.2]
@attribute slope real [1.0, 3.0]
@attribute ca real [0.0, 3.0]
@attribute thal real [3.0, 7.0]
@attribute num {0, 1, 2, 3, 4}
```

Imported File cleveland-10-10tra.dat (KEEL format)

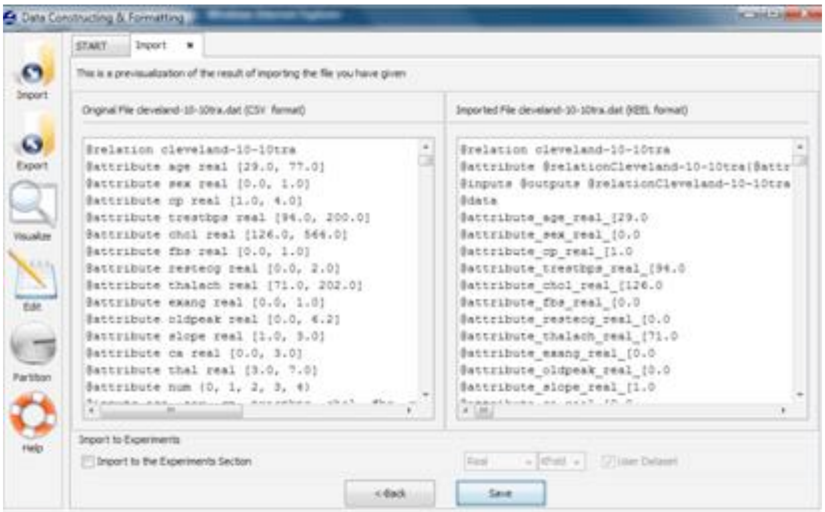
```
@relation cleveland-10-10tra
@attribute @relationCleveland-10-10tra{@attr
@inputs @outputs @relationCleveland-10-10tra
@data
@attribute_age_real_[29.0
@attribute_sex_real_[0.0
@attribute_cp_real_[1.0
@attribute_trestbps_real_[94.0
@attribute_chol_real_[126.0
@attribute_fbs_real_[0.0
@attribute_restecg_real_[0.0
@attribute_thalach_real_[71.0
@attribute_exang_real_[0.0
@attribute_oldpeak_real_[0.0
@attribute_slope_real_[1.0
```

Import to Experiments

□ Import to the Experiments Section

Figure 8. Format admitted to convert to KEEL format.

The "Options" button allows you to configure if it is necessary a certain separator and null value used in the source file.



This is a previsualization of the result of importing the file you have given

Original File cleveland-10-10tra.dat (CSV format)

```
@relation cleveland-10-10tra
@attribute age real [29.0, 77.0]
@attribute sex real [0.0, 1.0]
@attribute cp real [1.0, 4.0]
@attribute trestbps real [94.0, 200.0]
@attribute chol real [126.0, 564.0]
@attribute fbs real [0.0, 1.0]
@attribute restecg real [0.0, 2.0]
@attribute thalach real [71.0, 202.0]
@attribute exang real [0.0, 1.0]
@attribute oldpeak real [0.0, 6.2]
@attribute slope real [1.0, 3.0]
@attribute ca real [0.0, 3.0]
@attribute thal real [3.0, 7.0]
@attribute num {0, 1, 2, 3, 4}
```

Imported File cleveland-10-10tra.dat (KEEL format)

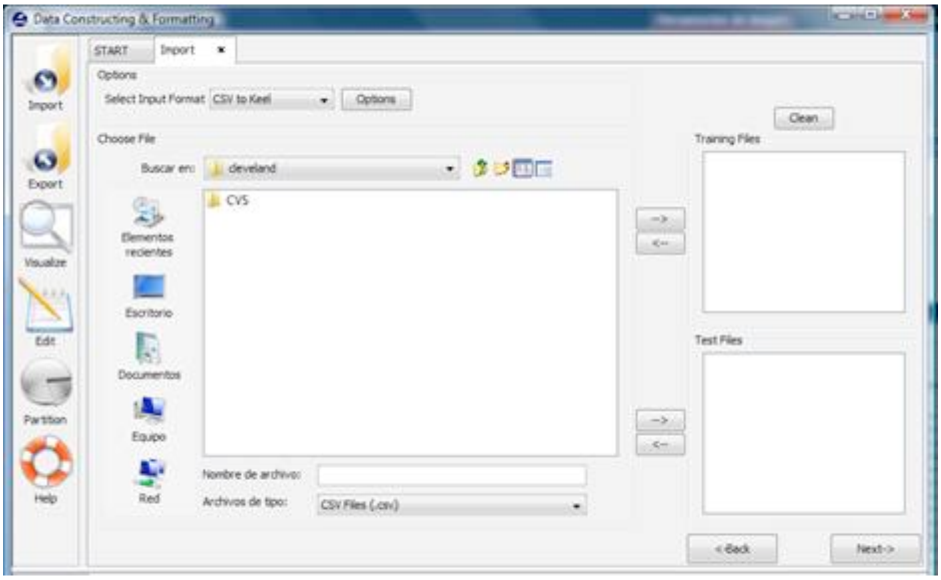
```
@relation cleveland-10-10tra
@attribute @relationCleveland-10-10tra{@attr
@inputs @outputs @relationCleveland-10-10tra
@data
@attribute_age_real_[29.0
@attribute_sex_real_[0.0
@attribute_cp_real_[1.0
@attribute_trestbps_real_[94.0
@attribute_chol_real_[126.0
@attribute_fbs_real_[0.0
@attribute_restecg_real_[0.0
@attribute_thalach_real_[71.0
@attribute_exang_real_[0.0
@attribute_oldpeak_real_[0.0
@attribute_slope_real_[1.0
```

Import to Experiments

□ Import to the Experiments Section

Figure 9. Specify certain options of format of source file.

Step2. Select the source file. After specifying the file format used in source file, the path of this file must be specified. You have to use the arrows to include the files in training or test properly.



Select Input Format: CSV to Keel
Options

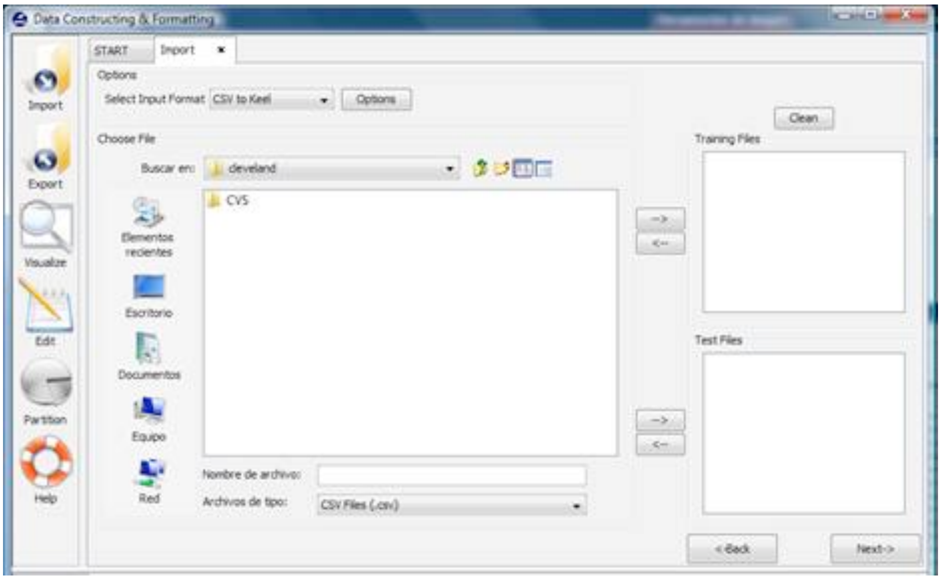
Buscar en: develand

Nombre de archivo:

Archivos de tipo: CSV Files (.csv)

Figure 10. Specify the source file paths.

Step 3. Save the files. Once type of conversion and source file have been configured, you must click *Next* button and the original and the imported file are shown (see Figure 11).



Select Input Format: CSV to Keel
Options

Buscar en: develand

Nombre de archivo:

Archivos de tipo: CSV Files (.csv)

Figure 11. The original and imported file.

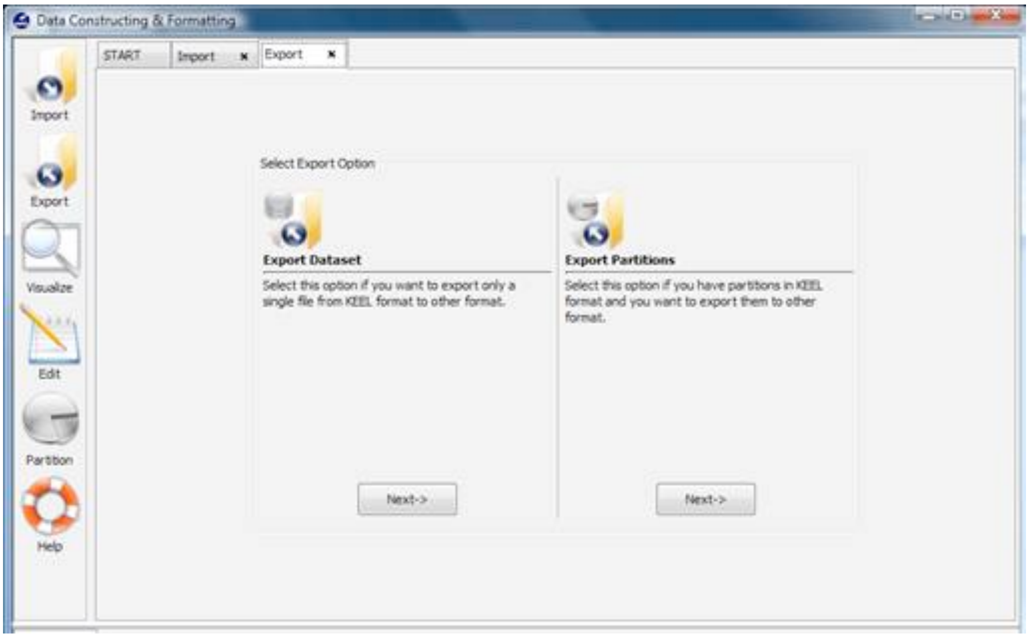
If we agree with the conversion done, there are two options to save the imported file:

1. Select the *Import to the experiments section*: if you mark this option, two new options are available. With this option you configure if the data set is a real or laboratory data set and the partitions that you are used. Two partitions are applicable: k fold or 5x2 cross validation. Then, when you select the save button, the data set that you are converted will be included as option in the KEEL experiments.

2. Not select the *Import to the experiments section*: if you do not select Import to the experiments section, when you click the save button, you have to select the destination directory for the transformed data sets.

Data export

Data export allows you to transform the datasets in KEEL format to the desired format (txt, excel, xml, html table, etc.).



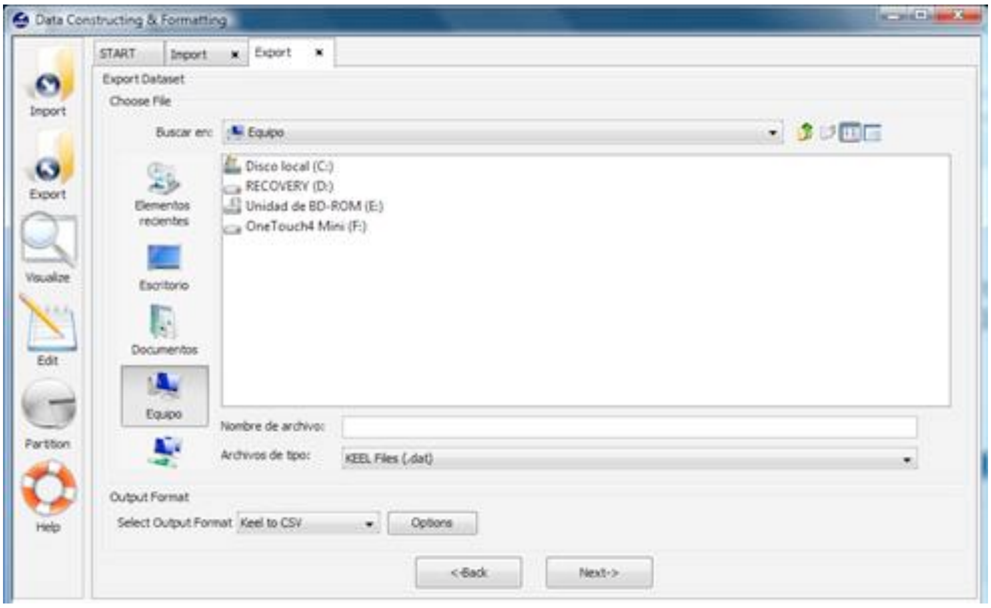
Export Dataset: Select this option if you want to export only a single file from KEEL format to other format.

Export Partitions: Select this option if you have partitions in KEEL format and you want to export them to other format.

Figure 1. Two possibilities to carry out the export process.

Figure 1 shows the two possible options to export data sets. One option consists of exporting one data set, the other option consists of exporting a set of partitions which you have available in other formats different to KEEL format. In continuation, we show the process of two options.

1. **Export Dataset.** Select this option if you want to export only a single file from KEEL format to other format (see Figure 2).



Buscar en: Equipo

Disco local (C:)
RECOVERY (D:)
Unidad de BD-ROM (E:)
OneTouch4 Mini (F:)

Nombre de archivo:

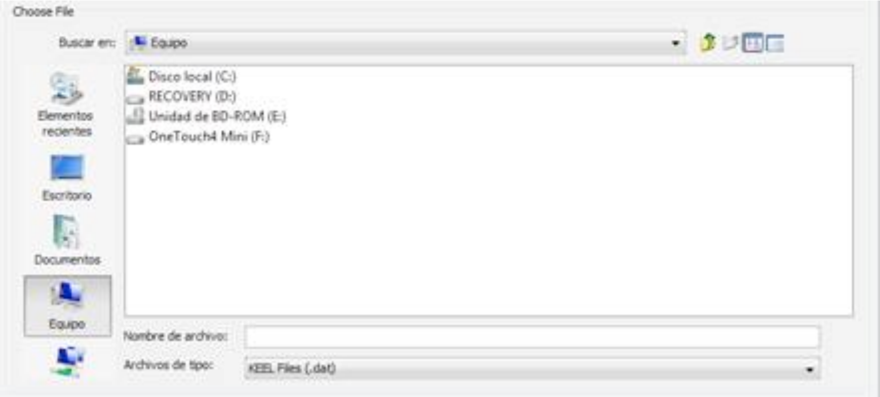
Archivos de tipo: KEEL Files (.dat)

Select Output Format Keel to CSV

Figure 2. Export Dataset option.

This option consists of the next parts:

Step1. Select the source file. First of all, the path of source file must be specified (a browser commonly known from many other GUI programms is used to define this path).



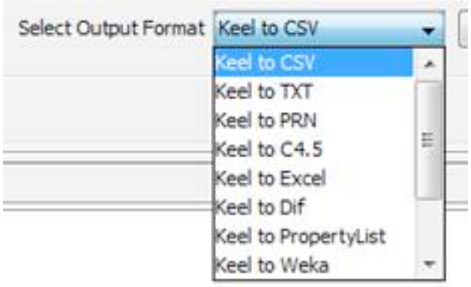
Choose File

Buscar en: Equipo

Archivos de tipo: KEEL Files (.dat)

Figure 3. Specify the source file path.

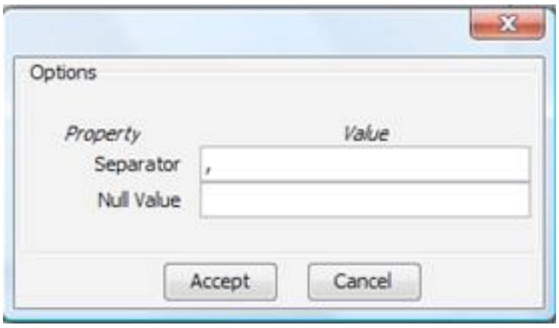
Step2. Select Input Format. After choosing the file, you must select the format of destination file. The formats admitted are CVS, TXT, PRN, C4.5, Excel, Dif, PropertyList and Weka. The different options are shown in Figure 4.



Select Output Format: Keel to CSV

Figure 4. Format admitted to convert from KEEL format.

The "Options" button allows you to configure if it is necessary a certain separator and null value used in the source file.

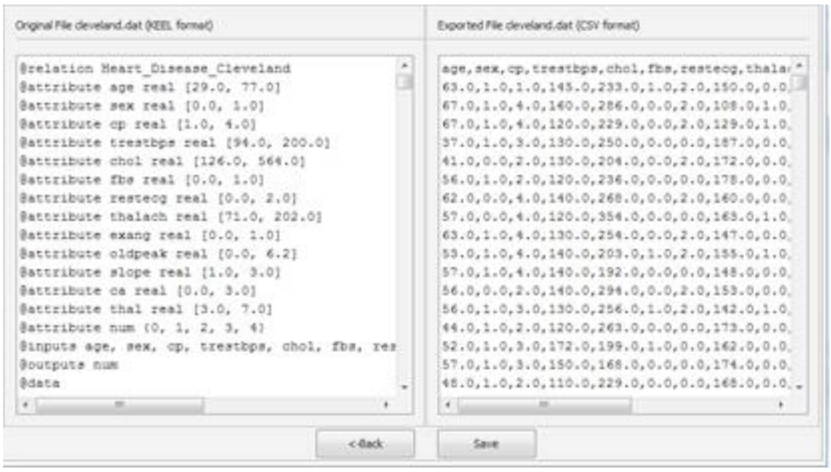


Options

Property	Value
Separator	,
Null Value	

Figure 5. Specify certain options of format of destination file.

Step 3. Save the files. Once the type of conversion and path of file have been configured, you must click *Next* button and then, the original and the exported file are shown (see Figure 6).



Original File cleveland.dat (KEEL format)

```
@relation Heart_Disease_Cleveland
@attribute age real [29.0, 77.0]
@attribute sex real [0.0, 1.0]
@attribute cp real [1.0, 4.0]
@attribute trestbps real [94.0, 200.0]
@attribute chol real [126.0, 564.0]
@attribute fbs real [0.0, 1.0]
@attribute restecg real [0.0, 2.0]
@attribute thalach real [71.0, 202.0]
@attribute exang real [0.0, 1.0]
@attribute oldpeak real [0.0, 6.2]
@attribute slope real [1.0, 3.0]
@attribute ca real [0.0, 3.0]
@attribute thal real [3.0, 7.0]
@attribute num {0, 1, 2, 3, 4}
@inputs age, sex, cp, trestbps, chol, fbs, res
@outputs num
@data
```

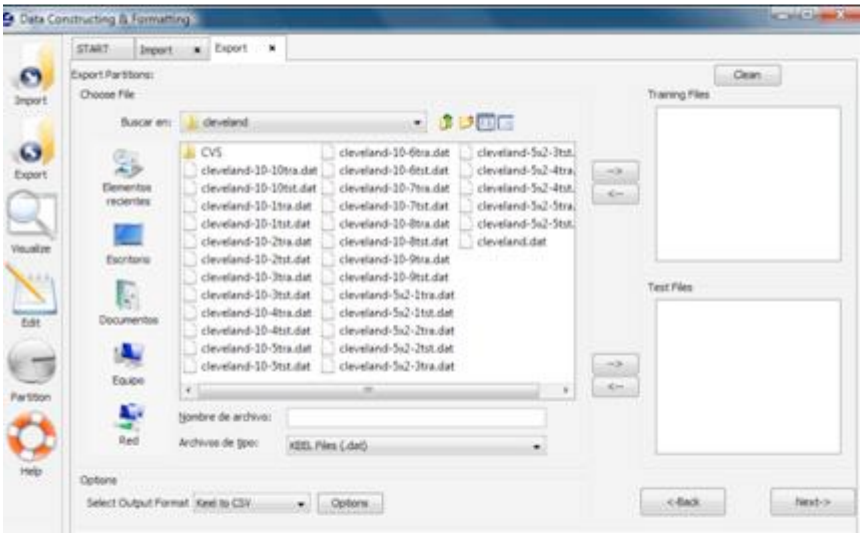
Exported File cleveland.dat (CSV format)

```
age,sex,cp,trestbps,chol,fbs,restecg,thala
63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0
67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0
67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0
37.0,1.0,3.0,130.0,250.0,0.0,0.0,187.0,0.0
41.0,0.0,2.0,130.0,204.0,0.0,2.0,172.0,0.0
56.0,1.0,2.0,120.0,236.0,0.0,0.0,178.0,0.0
62.0,0.0,4.0,140.0,268.0,0.0,2.0,160.0,0.0
57.0,0.0,4.0,120.0,354.0,0.0,0.0,163.0,1.0
63.0,1.0,4.0,130.0,254.0,0.0,2.0,147.0,0.0
53.0,1.0,4.0,140.0,203.0,1.0,2.0,155.0,1.0
57.0,1.0,4.0,140.0,192.0,0.0,0.0,148.0,0.0
56.0,0.0,2.0,140.0,294.0,0.0,2.0,153.0,0.0
56.0,1.0,3.0,130.0,256.0,1.0,2.0,142.0,1.0
44.0,1.0,2.0,120.0,263.0,0.0,0.0,173.0,0.0
52.0,1.0,3.0,172.0,199.0,1.0,0.0,162.0,0.0
57.0,1.0,3.0,150.0,168.0,0.0,0.0,174.0,0.0
48.0,1.0,2.0,110.0,229.0,0.0,0.0,168.0,0.0
```

Figure 6. Previsualization of original and exported file.

If we agree with the conversion done, you click the Save button and you can select the destination directory for the transformed dataset.

2. Export Partitions. Select this option if you have previously performed partitions in KEEL format and you want to export them to other format. This option allows selecting a set of training and test files separately. Figure 7 shows the window with respect to this option.



Export Partitions:

Choose File

Buscar en: cleveland

```
CVS
cleveland-10-10tra.dat
cleveland-10-10tst.dat
cleveland-10-1tra.dat
cleveland-10-1tst.dat
cleveland-10-2tra.dat
cleveland-10-2tst.dat
cleveland-10-3tra.dat
cleveland-10-3tst.dat
cleveland-10-4tra.dat
cleveland-10-4tst.dat
cleveland-10-5tra.dat
cleveland-10-5tst.dat
cleveland-10-6tra.dat
cleveland-10-6tst.dat
cleveland-10-7tra.dat
cleveland-10-7tst.dat
cleveland-10-8tra.dat
cleveland-10-8tst.dat
cleveland-10-9tra.dat
cleveland-10-9tst.dat
cleveland-5x2-1tra.dat
cleveland-5x2-1tst.dat
cleveland-5x2-2tra.dat
cleveland-5x2-2tst.dat
cleveland-5x2-3tra.dat
cleveland-5x2-3tst.
cleveland-5x2-4tra.
cleveland-5x2-4tst.
cleveland-5x2-5tra.
cleveland-5x2-5tst.
cleveland.dat
```

Nombre de archivo:

Archivos de tipo: KEEL Files (.dat)

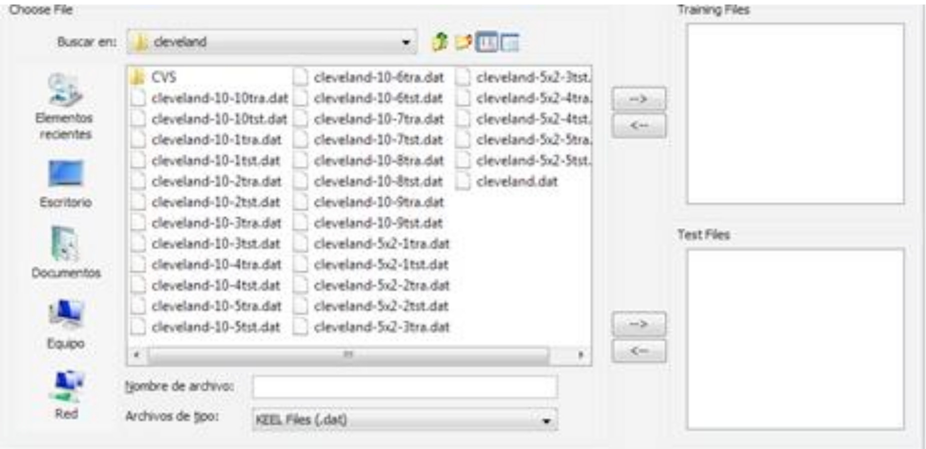
Options

Select Output Format: Keel to CSV

Figure 7. Export Partitions option.

This option consists of the next parts:

Step1. Select the source files. First of all, the path of source file must be specified . You have to use the arrows to include the files in training or test properly.

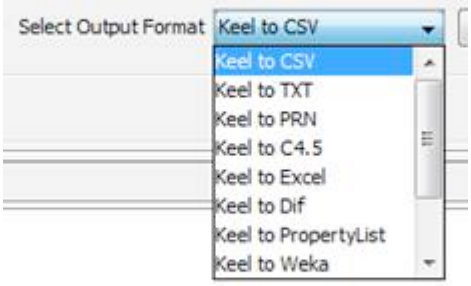


Buscar en: cleveland

Archivos de tipo: KEEL Files (.dat)

Figure 8. Specify the source file paths.

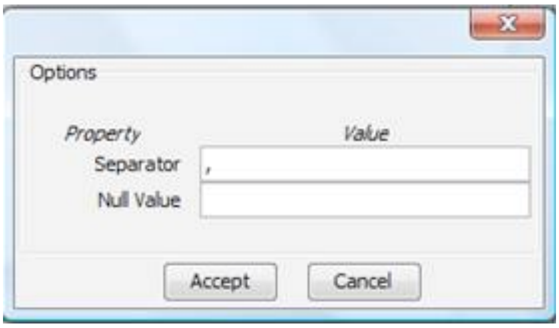
Step2. Select Input Format. After choosing the file, you must select the type of conversion. The formats admitted are CVS, TXT, PRN, C4.5, Excel, Dif, PropertyList and Weka. The different options are shown in Figure 9.



Select Output Format: Keel to CSV

Figure 9. Format admitted to convert to KEEL format.

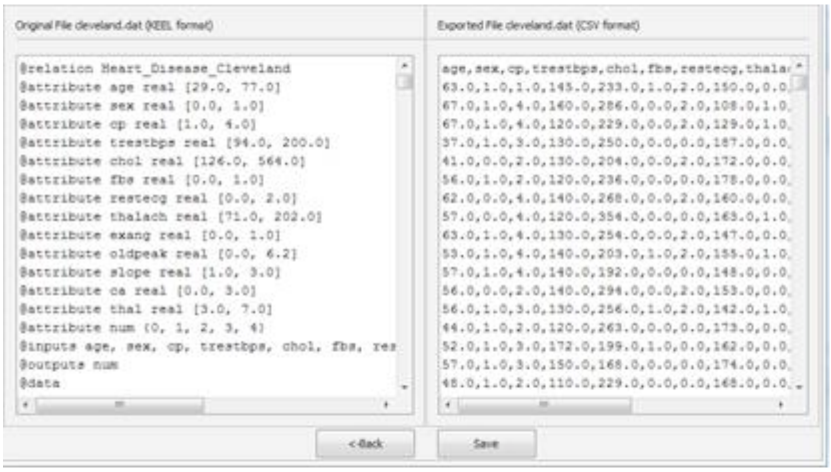
The "Options" button allows you to configure if it is necessary a certain separator and null value used in the source file.



Property	Value
Separator	,
Null Value	

Figure 10. Specify certain options of format of destination file.

Step 3. Save the files. Once the type of conversion and path of file have been configured, you must click *Next* button and the original and the exported file are shown (see Figure 11).


Original File cleveland.dat (KEEL format)
Exported File cleveland.dat (CSV format)
<-Back
Save

```
@relation Heart_Disease_Cleveland
@attribute age real [29.0, 77.0]
@attribute sex real [0.0, 1.0]
@attribute cp real [1.0, 4.0]
@attribute trestbps real [94.0, 200.0]
@attribute chol real [126.0, 564.0]
@attribute fbs real [0.0, 1.0]
@attribute restecg real [0.0, 2.0]
@attribute thalach real [71.0, 202.0]
@attribute exang real [0.0, 1.0]
@attribute oldpeak real [0.0, 6.2]
@attribute slope real [1.0, 3.0]
@attribute ca real [0.0, 3.0]
@attribute thal real [3.0, 7.0]
@attribute num {0, 1, 2, 3, 4}
@inputs age, sex, cp, trestbps, chol, fbs, res
@outputs num
@data
```

```
age,sex,cp,trestbps,chol,fbs,restecg,thala
63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0
67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0
67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0
37.0,1.0,3.0,130.0,250.0,0.0,0.0,187.0,0.0
41.0,0.0,2.0,130.0,204.0,0.0,2.0,172.0,0.0
56.0,1.0,2.0,120.0,236.0,0.0,0.0,178.0,0.0
62.0,0.0,4.0,140.0,268.0,0.0,2.0,160.0,0.0
57.0,0.0,4.0,120.0,354.0,0.0,0.0,163.0,1.0
63.0,1.0,4.0,130.0,254.0,0.0,2.0,147.0,0.0
53.0,1.0,4.0,140.0,203.0,1.0,2.0,155.0,1.0
57.0,1.0,4.0,140.0,192.0,0.0,0.0,148.0,0.0
56.0,0.0,2.0,140.0,294.0,0.0,2.0,153.0,0.0
56.0,1.0,3.0,130.0,256.0,1.0,2.0,142.0,1.0
44.0,1.0,2.0,120.0,263.0,0.0,0.0,173.0,0.0
52.0,1.0,3.0,172.0,199.0,1.0,0.0,162.0,0.0
57.0,1.0,3.0,150.0,168.0,0.0,0.0,174.0,0.0
48.0,1.0,2.0,110.0,229.0,0.0,0.0,168.0,0.0
```

Figure 11. Previsualization of original and exported file.

If we agree with the conversion done, you click the Save button and you can select the destination directory for the transformed dataset.

File Formats

There are different formats of the data with which you can work on KEEL tool. Following, we will show the different available formats.

CVS DATA FILE FORMAT

The CSV file (comma-separated-values). CSV is one implementation of a delimited text file, which uses a comma to separate values. The CSV file format is very simple and supported by almost all spreadsheets and database management systems

The characteristics of these files are the following:

- The first record in a CSV file may be a header record containing name of the columns.
- Each record in a file can have less fields that the number of header columns. In this case, empty values are considered missing values.
- Each row must have the same number of fields separated by commas.
- Two adjacent commas or comma at the beginning or end of the line (space-characters) indicate null values.
- The separation of the whole and fractional part in the actual numbers is done through a point instead of a comma.
- The separation symbol for decimals numbers is a point instead of a comma.
- Leading and trailing space-characters adjacent to comma field separators are ignored.
- Each record is one line terminated by a newline character or a carriage return.
- The blank lines will be ignored.
- Fields that contain double quote characters must be surrounded by double-quotes, and the embedded double-quotes must each be represented by a pair of consecutive double quotes.
- Fields with leading or trailing spaces or commas must be delimited with double-quote characters.
- The delimiter of values can be other character different to comma. Many implementations of CSV allow an alternate separator to be used, such as tab character and the resulting format is TSV (Tab Separated Values).
- The last record in a file can be finished or not with the character end of line.
- These files are stored, by default, with the extension. "Csv".

The CSV (Comma-Separated Values) data files must have the following format:

attribute$_1$*, attribute*$_2$*, ..., attribute*$_N$
value$_{11}$*, value*$_{12}$*, ..., value*$_{1N}$
...
value$_{M1}$*, value*$_{M2}$*, ..., value*$_{MN}$

One example of valid CSV file is:

```
FirstName, LastName, Company, EmailAddress
Johnathan,Doe,"ABC Company","johndoe@abccompany.com"
Harrie,Wong,"Company Inc.","hwong@myprovider.com"
Mary,"Jo Smith","Any Corp.","mjsmith@myprovider.com"
```

In this example we can see the use of certain rules explained before, such as null value expressed in two consecutive commas, the use of the decimal point as a separator for real numbers and the use of double quotes to use the value of the comma simple as part of the data and not as a separator.

Another example of valid CSV file is:

```
OBS,CAREXPEND,DISPOSINC,DOLLARVALUE,WAGES
"1960:1",14.2,362,,270.7
"1960:2",14.1,365.9,,273.4
"1960:3",14.6,367.6,,273.9
"1960:4",13.2,369.2,,273.3
"1961:1",10.8,72.9,,273.7
"1961:2",11.7,378.4,,277.6
"1961:3",12.2,385.1,,282.2
"1961:4",13.7,393.2,,288.4
```

TXT and TVS DATA FILE FORMAT

The TXT (Text Separated by Tabs) or TSV (Tab Separated Values), is a simple text data that allows tabular data to be exchanged between applications with a different internal format. Values separated by tabs have been officially registered as a MIME type (Multipurpose Internet Mail Extensions) under the name text/tab-separated-values.

The characteristics of these files are the following:

- A file in TXT format consists of lines. Each line contains fields separated from one another by the tab character (horizontal tab, HT, code control 9 in ASCII).

- Fields can be any string of characters, excluding tabs. However, tabs usually don't appear in data items that you wish to tabulate, so this is seldom a restriction. There are various other formats which are very similar to TSV but use a different separator, such as Comma Separated Values (CSV) which uses the comma as separator. Commas, spaces, and other characters often used as separators in such formats appear rather often in data to be tabulated, at least in header fields.

- Each line must contain the same number of fields.

- The first line contains the name of the fields or attributes, i.e. the column headers.

- An empty value is displayed as an empty field between tabs.

- Such files can be read and edited by any text editors.

- Although TSV is a text format, this type of format is *not* expected appearing with a nice tabular format when it is printed with an editor or left on the screen.

- The extension for this type of file is. "Txt" 'or ". Tsv.

The TXT (Text Separated by Tabulators) or TSV (Tab/Text Separated Values) data files must have the following format:

attribute$_1$*<TAB> attribute<TAB>...<TAB>attribute*$_N$
value$_{11}$*<TAB> value*$_{12}$*<TAB> ... <TAB> value*$_{1N}$
...
value$_{M1}$*<TAB> value*$_{M2}$*<TAB> ... <TAB> value*$_{MN}$

One example of valid TXT or TSV file is the following:

FirstName <TAB> LastName <TAB> Company <TAB> EmailAddress
Johnathan <TAB> Doe <TAB> ABC Company <TAB> johndoe@abccompany.com
Harrie <TAB>Wong <TAB> Company <TAB> Inc. hwong@myprovider.com
Mary <TAB> Jo Smith <TAB> Any <TAB> Corp <TAB> mjsmith@myprovider.com"

PRN DATA FILE FORMAT

This format has the same features and restrictions that the CSV format, the difference is the separator between fields in PRN format are spaces. However, the spaces in PRN format have a different role than in CVS files.

The characteristics of these files are the following:

- The first record in a PRN file may be a header record containing name of the columns.
- Each record in a file with headers in columns can have less fields than the number of headers. In this case, empty values are considered missing values.
- Each row must have the same number of fields separated by spaces.
- Several spaces together will be treated as a single space.
- The spaces at the beginning or end of the line indicated null values.
- The separation symbol for decimals numbers is a point instead of a comma.
- Each record is one line terminated by a newline character or a carriage return.
- The blank lines will be ignored.
- The fields can contain double quote, carriage return (or any other character).
- Fields that contain space character as value must be surrounded by double-quotes.
- A record with a single field without any value must have the requirements of type text to prevent that it is not ignored.
- The last record in a file can be finished or not with the end of line symbol.
- These files are stored by default, with the extension ".prn".

The PRN files have the data separated by blank spaces. So, these data files must have the following format:

$attribute_1\ attribute_2\ \ldots\ attribute_N$
$value_{11}\ value_{12}\ \ldots\ value_{1N}$
...
$value_{M1}\ value_{M2}\ \ldots\ value_{MN}$

One example of a valid PRN file is the following:

```
OBS DELL GE YAHOO
1 26.99 48.5 22.92
2 26 49.93 20.83
3 26.24 49.96 20.13
4 25.76 49.48 19.98
5 26.73 49.43 19.74
6 24.93 49.83 18.86
7 25.84 49.01 18.23
8 25.91 49.73 17.79
9 24.6 50.15 17.1
```

DIF DATA FILE FORMAT

DIF (Data Interchange Format) is a text file that is used to import/export between different spreadsheet programs such as Excel, StarCalc, dBase, and so on.

This type of format is stored with the extension ". dif"

The characteristics of these files are the following:

- The format consists of a header followed by a data block. The header starts with a file with ASCII text format.

```
TABLE
0,1
”string”

VECTORS
0,columns
””
TUPLES
0,rows
””
DATA
0,0
””
```

  - **string** is any string, it is often the filename or another information.
  - **columns** is the number of columns of a excel spreadsheet by means of name.
  - **rows** indicates the number of rows of a excel spreadsheet by means of name.

- The header ends with the following:

```
DATA
0,0
””
```

  This header is followed by the cells and records of the spreadsheet with the information.

- The structure of the data record has the following format:

```
data-type, data
”string”
```

where **data-type** admits various types: SPECIAL, NUMERIC, and STRING, represented by -1, 0 and 1 respectively.

- ***SPECIAL type***

```
-1,0
BOT
…
.1,0
EOD
```

where BOT and EOD are strings without quotation marks. BOT represents the start of the table and EOD the end of data section.

- **NUMERIC type**

```
0,data
Value-indicator
```

where value-indicator indicates the data type stored in data:

- TRUE:1.
- FALSE: 0.
- V: any numerical value.
- NA: missing value.
- ERROR: 0.

- **STRING type**

```
1,0
“string”
```

where string is any text characters.

One example of a valid DIF file is the following:

Month	Week	Vehicle	Quantity
January	1	Auto	105.000
January	1	Lorry	1.050
January	1	Bus	1.575
January	1	Lorry	2.100
January	1	Motorbike	583

The internal format of DIF file generated is the following:

```
TABLE
0,1
”EXCEL”
VECTORS
0,6
””
TUPLES
0,4
””
DATA
0,0
””
-1,0
BOT
1,0
“Month”
1,0
“Wek”
1,0
“Vehicle”
1,0
“Cantity”
-1,0
BOT
1,0
“January”
0,1
V
1,0
“Car”
0,105.000
V
-1,0
BOT
1,0
“January”
0,1
V
1,0
“Lorry”
0,1.050
V
-1,0
BOT
1,0
“January”
0,1
“Bus”
0,1.575
V
-1,0
BOT
1,0
“January”
0,1
“Lorry”
0,2.100
V
-1,0
BOT
1,0
“January”
0,1
V
1,0
“Motorbike”
0,583
V
-1,0
EOD
```

C4.5 DATA FILE FORMAT

Files are encoded according to C4.5 format. This format consists of two files, one of them it is a name file with extension ".names", the other one it is a data file with extension ".data".

The characteristics of name files are the following:

- The .names file contains a series of entries that describe the classes, attributes and values of the dataset. Each record is terminated with a point, but the point can be omitted if it would have been the last character on a line). Each name consists of a string of characters without commas, quotes or colon (unless escaped by a vertical bar, |).
- A name can contain a point, but this point must be followed by a white space.
- Embedded white spaces is permitted but multiple white spaces are replaced by a single space.

- The first record in the file lists the names of the classes, separated by commas and terminated by a point. Each successive line then defines an attribute, in the order in which they will appear in the .data files, with the following format:

  ```
  <attribute-name: attribute-type>.
  ```

  The attribute-name is an identifier followed by a colon. The attribute type which must be one of:

  - ***continuous***: if the attribute has a continuous values.
  - ***discrete <n>:*** the word 'discrete' followed by an integer which indicates how many values the attribute can take.
  - ***ignore:*** indicates that this attribute should be ignored.

- A | (vertical bar) means that the remainder of the line should be considered as a comment.

- These files are stored, by default, with the extension. "Names".

The format of the '.name' file is the following:

class-1, class-2, ..., class-N.
characteristic-1: domain.
characteristic-2: domain.
...
characteristic-M: domain.

The characteristics of data files are the following:

- The file contains one line by object. Each line contains values of the attributes sorted according to .names file, followed by the class of object, with all entries separated by commas.

- The format is same than CVS file (comma separated values), explained in CVS Data File Format.

- A missing values are indicated by '?'.

- These files are stored, by default, with the extension. "Data".

The format of the '.data' file is the following:

$value_{11}$, $value_{12}$, ..., $value_{1N}$
$value_{21}$, $value_{22}$, ..., $value_{2N}$
...
$value_{M1}$, $value_{M2}$, ..., $value_{MN}$

An example of a C4.5 data file is the following

- *Content of the '.name' file:*

```
| Firstly the name of classes
good, bad.
|Then the attributes
dur: continuous.
wage1: continuous.
wage2: continuous.
wage3: continuous.
cola: tc, none, tcf.
hours: continuous.
pension: empl contr, ret allw, none.
stby_pay: continuous.
shift_diff: continuous.
educ_allw: yes, no.
holidays: continuous.
vacation: average, generous, below average.
lngtrm_disabil: yes, no.
dntl_ins: half, none, full.
bereavement: yes, no.
empl_hplan: half, full, none.
```

- *Content of the '.data' file:*

```
2,5.0,4.0,?,none,37,?,?,5,no,11,below average,yes,full,yes,full,good
3,2.0,2.5,?,?,35,none,?,?,?,10,average,?,?,yes,full,bad
3,4.5,4.5,5.0,none,40,?,?,?,no,11,average,?,half,?,?,good
3,3.0,2.0,2.5,tc,40,none,?,5,no,10,below average,yes,half,yes,full,bad
```

EXCEL DATA FILE FORMAT

Microsoft Excel is a spreadsheet program written and distributed by Microsoft. It is currently the most widely used spreadsheet for operative systems Microsoft Windows and Apple Macintosh. It is integrated as part of Microsoft Office.

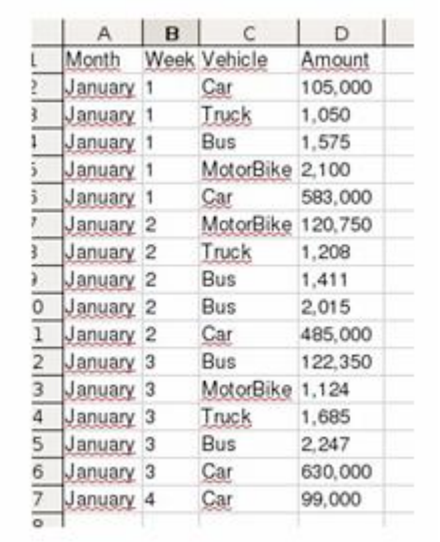
A spreadsheet is a program that allows you to manipulate numerical and alphanumeric data. Spreadsheets are arranged in rows and columns. The intersection of a row/column is called cell

Each cell can contain data or a formula that can refer to the contents of other cells. A spreadsheet contains 256 columns, which are labelled with letters (from A to IV) and the rows with numbers (from 1 to 65.536), making a total of 16.777.216 cells by spreadsheet.

Because of the versatility of modern spreadsheets, they are used to sometimes to make smaller databases, reports, and other uses.

Microsoft Excel format has extension ".xls".

One example of a valid EXCEL file is:



A	B	C	D
Month	Week	Vehicle	Amount
January	1	Car	105,000
January	1	Truck	1,050
January	1	Bus	1,575
January	1	MotorBike	2,100
January	1	Car	583,000
January	2	MotorBike	120,750
January	2	Truck	1,208
January	2	Bus	1,411
January	2	Bus	2,015
January	2	Car	485,000
January	3	Bus	122,350
January	3	MotorBike	1,124
January	3	Truck	1,685
January	3	Bus	2,247
January	3	Car	630,000
January	4	Car	99,000

WEKA DATA FILE FORMAT

The weak data files are in the following format:

- **Headline**. The relation name is defined as the first line in the ARFF file. The format is:

  @ relation <name-of-relation>

  where <relation-name> is a string. The string must be quoted if the name includes spaces.

- **Declaration of attributes**. Attribute declarations take the form of an orderd sequence of *@attribute* statements. Each attribute in the data set has its own *@attribute* statement which uniquely defines the name of that attribute and it's data type. The order the attributes are declared indicates the column position in the data section of the file. For example, if an attribute is the third one declared then Weka expects that all that attributes values will be found in the third comma delimited column. The format for the *@attribute* statement is:

  @ attribute <attribute-name> <datatype>

  <attribute-name>: must start with an alphabetic character. If spaces are to be included in the name then the entire name must be quoted.

  <datatype>: can be any of the four types currently (version 3.2.1) supported by Weka:

1) **NUMERIC or REAL.** Numeric attribute can be real numbers.

2) **INTEGER.** Integer attribute can be integer numbers.

3) **DATE**. Date attribute is an optional string specifying how date values should be parsed and printed. The default format string accepts the ISO-8601 combined date and time format: "yyyy-MM-dd'T'HH:mm:ss".

4) **STRING.** String attributes allow us to create attributes containing arbitrary textual values.

5) **ENUMERATE.** Enumerate attribute consists of a set of possible values separated by commas (Characters or strings), which can take the attribute. For example, if we have an attribute that indicates the time podr'ıa Express:
   @ attribute time {sunny, rainy, cloudy}

- **Section data**. The data section of the file contains the data declaration line and the actual instance lines. The **@data** declaration is a single line denoting the start of the data segment in the file. The format is:

  @ data
  X11, x12, ... , X1N
  X21, x22, ... , X2N

  Each instance is represented on a single line, with carriage returns denoting the end of the instance.

  Attribute values for each instance are delimited by commas. They must appear in the order that they were declared in the header section (i.e. the data corresponding to the nth @attribute declaration is always the nth field of the attribute).

  Missing values are represented by a single question mark, as in:

  ```
  @data
  4.4,?,1.5,?,Iris-setosa
  ```

Some of the specifications of this format are:

- The name of the relationship and the attributes are string type. This string type is same than string type used on Java.
- If any name contains spaces it is necessary to include double quote.
- If you need to indicate a missing values, you have to use symbol '?'.
- The separation symbol for decimals numbers is a point instead of a comma.
- The separation symbol for data in section @ data is comma.
- A % symbol means that the remainder of the line should be considered as a comment.
- These files are stores, by default, with the extension ".arff’.

The WEKA data files must have the following format:

@relation <relation-name>
@attribute <attribute-name-1> <datatype>
...
@attribute <attribute-name-N> <datatype>
@data
$value_{11}$,$value_{12}$,$value_{1N}$
...
$value_{M1}$,$value_{M2}$,$value_{MN}$

One example of a valid WEKA file is:

```
% Comment

@relation weather
@attribute outlook sunny, overcast, rainy
@attribute temperature real
@attribute humidity real
@attribute windy TRUE, FALSE
@attribute play yes, no
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
```

XML DATA FILE FORMAT

XML (Extensible Markup Language) is a set of rules to define semantic labels that organize a document in different parts. XML is a meta-language that defines the syntax to define other structured label languages.

We will explain the XML format to be followed to convert data file correctly:

- The first line must follow the next structure:

  <? Xml version = "1.0" encoding = "UTF-8" standalone = "yes">

  You can have several attributes, some mandatory and others are not:

- *version*: indicates XML version used in the document. This field is compulsory.
- *encoding*: indicates the way that has been encoded document. The default option is UTF-8, but could be others, as UTF-16, US-ASCII, ISO-8859-1, etc. This field is not obligatory unless.
- *standalone:* specifies whether further documents, such as a DTD, are required to process the document. The default value is "no"..

- XML documents must follow a hierarchical structure by means of labels. XML elements can contain other elements. Elements may also have **attributes**, these are always expressed as name-value pairs in the element's open tag.

- A well-formed document must conform to the following rules:

  - Element names are case sensitive, that is, the following is a well-formed matching pair: <step>…<step>, whereas this is not <step>…</step>.

  - Non-empty elements are delimited by both a start-tag and an end-tag.

  - Attribute values must always be quoted, using single or double quotes, and each attribute name should appear only once in any element

  - All spaces and carriage returns are taken into account in the elements.

  - The element names must not begin with the letters “xml”.

  - The element names should not use character ":".

  - Although it is permissible to use the characters "." And "-" in element names, it is not recommended because the application processing XML file may interpret these signs as operators. Therefore these characters will be replaced in our tool by the character "_”.

  - It should not be used characters "\" in the names of elements.

  - The names may contain any alphanumeric character, but they can not start with a numerical or punctuation character.

- Special characters can be represented either using entity references, or by means of numeric character references. An example of a numeric character reference is "&#x20AC;", which refers to the Euro symbol by means of its Unicode codepoint in hexadecimal.

  An entity reference is a placeholder that represents that entity. It consists of the entity's name preceded by an ampersand ("&") and followed by a semicolon (";"). XML has five predeclared entities:

  - & (ampersand) → &amp;
  - < (less than) → &lt;
  - > (greater than) → &gt;
  - ' (apostrophe) → &apos;
  - " (quotation mark) → &quot;

- **Comments** can be placed anywhere in the tree, including in the text if the content of the element is text. XML comments start with **<!-** and end with **->**.

```
<!- This is a comment. ->
```

- XML requires that elements be properly nested, that is, elements may never overlap. For example, the code below is not well-formed XML, because the <em> and <strong> elements overlap:

```
<!-- WRONG! NOT WELL-FORMED XML! -->
<p>Normal <em>emphasized <strong>strong emphasized</em> strong</strong></p>
```

- All XML documents must contain a single tag pair to define the root element. All other elements must be nested within the root element. All elements can have sub (children) elements. Sub elements must be in pairs and correctly nested within their parent element.

- The label <root> indicates the start point of the data. This label can have any name. If all the children of <root> do not have the same name on the label <row>, the user must enter the name of this tag, otherwise it is assumed that all children have the same value.

- Each label <row> is parent of as labels as attributes exist. The name on the label of each of these children will be the attribute name, and the value of the label is the data value of the attribute.

- There are as labels <row> as rows of data.

One XML format valid to Keel is the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
        <row1>
                <attribute-name-1> attribute-value-11 </attribute-name-1>
                <attribute-name-2> attribute-value-12 </attribute-name-2>
                <attribute-name-N> attribute-value-1N </attribute-name-N>
        </row1>
        ...
        <rowM>
                <attribute-name-1> attribute-value-M1 </attribute-name-1>
                <attribute-name-2> attribute-value-M2 </attribute-name-2>
                <attribute-name-N> attribute-value-MN </attribute-name-N>
        </rowM>
</root>
```

Another XML format valid to Keel is the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
        <row1>
                <field name="attribute-name-1">attribute-value-11 </field>
                <field name="attribute-name-2">attribute-value-12 </field>
                <field name="attribute-name-N">attribute-value-1N </field>
        </row1>
        ...
        <rowM>
                <field name="attribute-name-1">attribute-value-M1 </field>
                <field name="attribute-name-2">attribute-value-M2 </field>
                <field name="attribute-name-N">attribute-value-MN </field>
        </rowM>

</root>
```

One example of a valid XML file is the following:
In this example there are:

- 9 attributes: id, course, name, summary, numbering, disableprintg, customtitles, timecreated and timemodified.

- 2 instances with these 9 attributes.

- The main label is <root>

- The label <customer> contains each instance. In xml data file export to our tool, the name of this label will be the same than the name of data relation stores in keel format.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
        <clientes>
                <id>5</id>
                <course>66</course>
                <name>My book</name>
                <summary>Book summary</summary>
                <numbering>2</numbering>
                <disableprinting>0</disableprinting>
                <customtitles>1</customtitles>
                <timecreated>1114095924</timecreated>
                <timemodified>1114097355</timemodified>
        </clientes>
        <clientes>
                <id>6</id>
                <course>207</course>
                <name>My book</name>
                <summary>A test summary</summary>
                <numbering>1</numbering>
                <disableprinting>0</disableprinting>
                <customtitles>0</customtitles>
                <timecreated>1114095966</timecreated>
                <timemodified>1114095966</timemodified>
        </clientes>
</root>
```

The following example has another xml structure, but the same data than the previous example. You can see that there are 9 attributes and 2 instances of this.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <row>
        <field name="id">5</field>
        <field name="course">66</field>
        <field name="name">My book</field>
        <field name="summary">Book summary</field>
        <field name="numbering">2</field>
        <field name="disableprinting">0</field>
        <field name="customtitles">1</field>
        <field name="timecreated">1114095924</field>
        <field name="timemodified">1114097355</field>
    </row>
    <row>
        <field name="id">6</field>
        <field name="course">207</field>
        <field name="name">My book</field>
        <field name="summary">A test summary</field>
        <field name="numbering">1</field>
        <field name="disableprinting">0</field>
        <field name="customtitles">0</field>
        <field name="timecreated">1114095966</field>
        <field name="timemodified">1114095966</field>
    </row>
</root>
```

HTML DATA FILE FORMAT

HTML, an extension of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document (denoting certain text as headings, paragraphs, lists, and so on) and to supplement that text with *interactive forms*, embedded *images*, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets.

HTML is an application of SGML according to the international standard ISO 8879. XHTML is a reformulation of HTML 4 as an XML application 1.0, and allows compatibility with user agents already admitted HTML 4 following a set of rules.

The basic HTML tags are:

- **<HTML>:** is the label that defines the beginning of the document.
- **<HEAD>:** defines the header of the document, this header normally Contains information about the page such as the TITLE, META tags for proper Search Engine indexing, STYLE tags, which determine the page layout, and **JavaScript coding** for special effects. Within the header <HEAD> we find:

    - <TITLE>: defines the title of the page. This will be visible in the title bar of the viewers' browser.
    - <LINK>: defines some advanced features, for example style sheets used for the design of the page.
- **<BODY>:** contains the main content or body of the paper, this is where you will begin writing your document and placing your **HTML codes. It defines common** properties to the entire page, such as background color and margins. Within the body can <BODY> you can use a great variety labels. The label which we use on our tool is
    - **<TABLE>:** This label defines the beginning of a table (the <TR> represents rows and <TD> represents cells).

The format explained above corresponds to an HTML page is :

```
<HTML>
        <HEAD>
        ...
        </HEAD>
        <BODY>
        ...
                <TABLE>
                ...
                </TABLE>
        ....
        </BODY>
</HTML>
```

Tag <TABLE>

The HTML table model allows authors to arrange data -- text, preformatted text, images, links, forms, form fields, other tables, etc. -- into rows and columns of cells.

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

Different Tags which will define the structure of the table for obtaining a valid data file are:

- **TR:** The label <TR> will allow us to insert rows in the table.
- **TH:** The label <TH> will allow us to define the table head table.
- **TD:** The label <TD> will allow us to insert cells in each row. We can insert any element: pictures, lists, formatted text and even other tables.

The HTML format valid to Keel is the following:

```
<table>
        <tr>
                <th>Header 1</th>
                <th>Header 2</th>
                <th>Header 3</th>
        </tr>
        <tr>
                <td>Value 1</td>
                <td>Value 2</td>
                <td>Value 3</td>
        </tr>
        <tr>
                <td>Value 4</td>
                <td>Value 5</td>
                <td>Value 6</td>
        </tr>
</table>
```

One example of a valid HTML file is the following:

```
<html>
        <head>
                <h1 align="center">VEHICLES</h1>
        </head>
        <body>
                <table border="1" cellspacing="1" cellpadding="0">
                        <tr align="center">
                                <td>Month</td>
                                <td>Week</td>
                                <td>Vehicle</td>
                                <td>Amount</td>
                        </tr>
                        <tr>
                                <td>January</td>
                                <td>1</td>
                                <td>Car</td>
                                <td>105.0</td>
                        </tr>
                        <tr>
                                <td>January</td>
                                <td>1</td>
                                <td>Truck</td>
                                <td>1.05</td>
                        </tr>
                        <tr>
                                <td>January</td>
                                <td>1</td>
                                <td>MotorBike</td>
```

```
                                        <td>1.575</td>
                                </tr>
                                <tr>
                                        <td>January</td>
                                        <td>1</td>
                                        <td>Car</td>
                                        <td>2.1</td>
                                </tr>
                        </table>
                </body>
        </html>
```

Data Partition

Data partition allows you to make partitions of any existing KEEL format data set. Figure 1 shows the main window of this option.

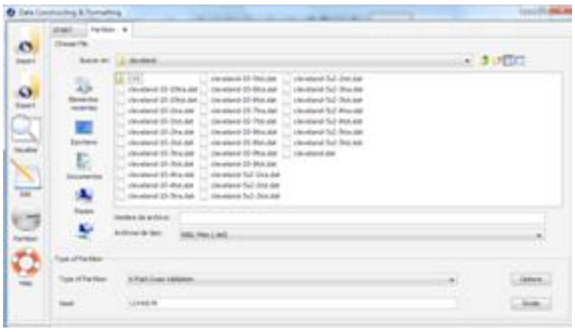


Figure 1. Process of Making Partitions.

The partition process consists of the next parts:

Step 1. Choose File. First of all, you must choose the path of complete data set (in KEEL format) that you want to make partitions of it (see Figure 2).

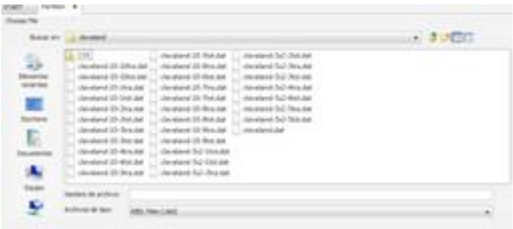


Figure 2. Choose data set to make partitions.

Step 2. Type of partition. Once the file is selected, you have to choose the type of partition. The different types considered are:

1. *K-fold cross validation:* this partition allows you to configure the number of fold to the partitions (if you want to configure the different options, you have to click on "Options" button).

2. *5x2 cross validation:* this partition does not allow you to configure options.

3. *Hold-Out:* this partition allows you to configure the number of partitions and the percentages of training and test sets (if you want to configure the different options, you have to click on "Options" button).

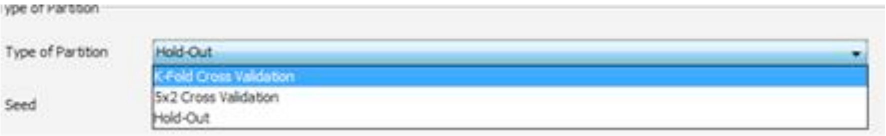
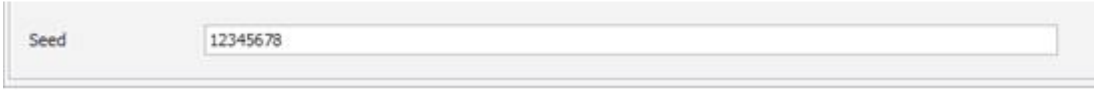


Figure 3. Type of Partition.

Step 3. Seed. To make the partitions of the data set, it is necessary to specify the random generator see to perform the process of division.



Seed 12345678

Figure 4. Select Seed.

Step 4: Make the partitions. Once that the previous steps have been configured, the partitions will be carried out when you click on “Divide” button. The files generated during the partition process are stored in the directory specified in Step 1 and the name of these files is given automatically and depends on the type of the partition selected.

Edit Data

Edit data allows you to edit any existing KEEL format data sets in order to add new attributes, to delete others, to correct some errors, etc.

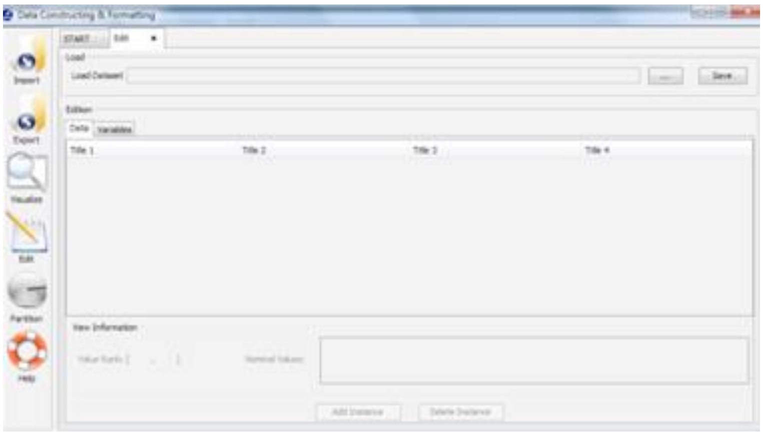


Figure 1. Process of Editing Data.

Figure 1 shows the main window of this option. First of all, you must select the path of source data set (in KEEL format) that you want to edit (see Figure 2).

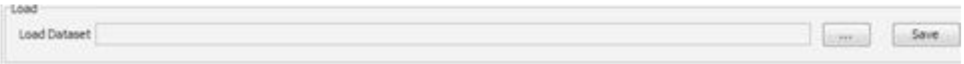
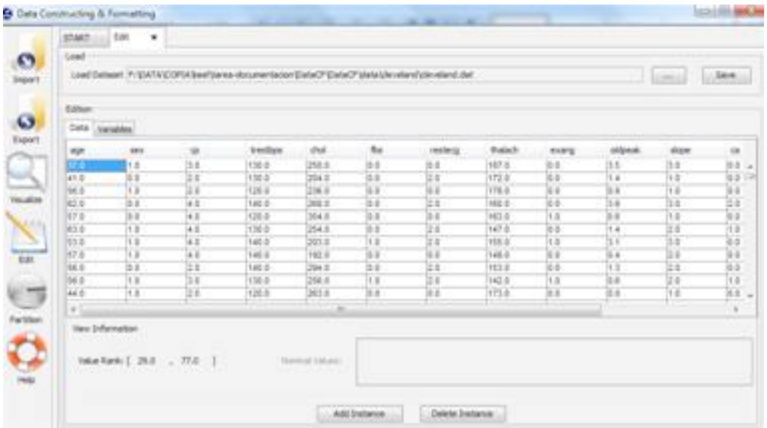


Figure 2. Load Dataset to edit its content.

Once the file is loaded, its content appears on the table. The modifications could be carried out both in instances and in variables. In following, the two options are shown.

1. Data Edition. In this option new instances could be added or existing instances could be deleted or modified (see Figure 3).



Load Dataset: F:\DATA\COPIA\keel\tarea-documentacion\DataCF\DataCF\data\cleveland\cleveland.dat

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0
41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0
56.0	1.0	2.0	120.0	236.0	0.0	0.0	178.0	0.0	0.8	1.0	0.0
62.0	0.0	4.0	140.0	268.0	0.0	2.0	160.0	0.0	3.6	3.0	2.0
57.0	0.0	4.0	120.0	354.0	0.0	0.0	163.0	1.0	0.6	1.0	0.0
63.0	1.0	4.0	130.0	254.0	0.0	2.0	147.0	0.0	1.4	2.0	1.0
53.0	1.0	4.0	140.0	203.0	1.0	2.0	155.0	1.0	3.1	3.0	0.0
57.0	1.0	4.0	140.0	192.0	0.0	0.0	148.0	0.0	0.4	2.0	0.0
56.0	0.0	2.0	140.0	294.0	0.0	2.0	153.0	0.0	1.3	2.0	0.0
56.0	1.0	3.0	130.0	256.0	1.0	2.0	142.0	1.0	0.6	2.0	1.0
44.0	1.0	2.0	120.0	263.0	0.0	0.0	173.0	0.0	0.0	1.0	0.0

Value Rank: [29.0 , 77.0]

Figure 3. Data Edition.

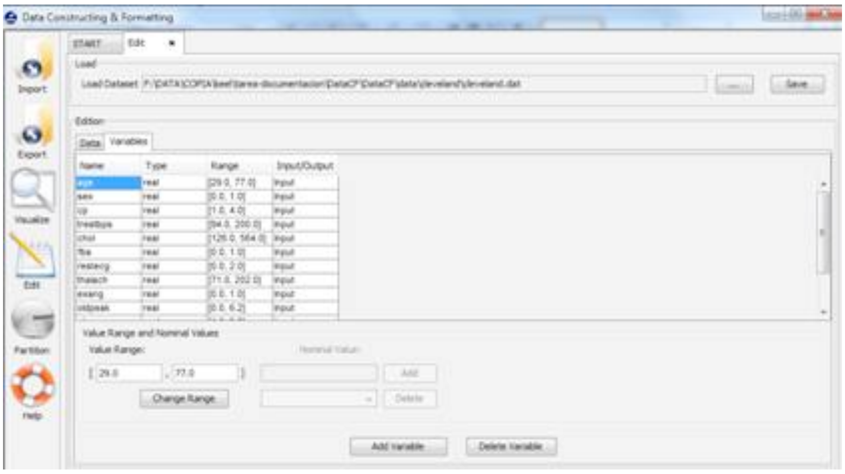
Using the table and the buttons situated over it you can perform the next operations:

- ***Modify the cell content***: if you want to modify the cell content you have to click on the cell that you want to change. Then, you only have to write the new value, and if it is valid, the cell content is modified. Otherwise an error dialog is shown.

- ***Delete an instance***: if you want to delete an instance, you have to click on any cell which corresponds with the instance (row) that you want to delete. Then, a "Delete Instace" button is activated. When you click on it, the full instance will be deleted.

- ***Add a new instance:*** if you want to add a new instance, you have to click the "Add Instance" button. Then, a new instance is added, if a cell was selected, the new instance is added at top of the instance where you had the cell selected. If no cell was selected, a new instance is added at the bottom of the table. The new instance has empty values and the different variables can be filled out clicking on the different cells.

2. Variable Edition. In this option different modifications on the variables can be carried out (see Figure 4).



Load Dataset: F:\DATA\COPIA\keel\tarea-documentacion\DataCF\DataCF\data\cleveland\cleveland.dat

Name	Type	Range	Input/Output
age	real	[29.0, 77.0]	input
sex	real	[0.0, 1.0]	input
cp	real	[1.0, 4.0]	input
trestbps	real	[94.0, 200.0]	input
chol	real	[126.0, 564.0]	input
fbs	real	[0.0, 1.0]	input
restecg	real	[0.0, 2.0]	input
thalach	real	[71.0, 202.0]	input
exang	real	[0.0, 1.0]	input
oldpeak	real	[0.0, 6.2]	input

Value Range: [29.0 , 77.0]

Figure 4. Variable Edition.

Using the table and the buttons situated over it you can perform the next operations:

- ***Add new variables***: if you want to add new variables, the “Add Variable” button has to be clicked on. Then, a new row is added to the bottom of the table and the different features can be filled out.

- ***Add new variables***: if you want to delete a variable, first a variable has to be selected and then the “Delete Variable” button has to be clicked.

- ***Change the rank values***: if you want to change the rank values, a real or integer variable has to be selected. If the variable selected is "Integer" or "Real", you can modify the rank values and then click on “Change Range” button to change the information.

- ***Change nominal values***: if you want to change the nominal values, a nominal variable has to be selected. If the variable selected is nominal, you can add or remove the values allowed for that variable using the list and “Delete” and “Add” button.

- ***Change attribute type***: if you want to change the variable type, you have to click on the cell that you want to change of “Type” column. Then, you can modify the variable type by means of the list specifying any other type.

- ***Change attribute 'function'***: if you want to change the attribute function, you have to click on the cell that you want to change of “Function” column. Then, you can modify the variable 'function' by means of the list specifying if it is "input" or "output".

When you have made all the changes, you can save them to a file pressing the "Save" button.

EXPERIMENT DESIGN

The Experiments Design part has the objective of designing the desired experiments using a graphical interface. Doubtless, this is the more innovative tool integrated in this program. The objective is to use available datasets and algorithms to generate a directory structure with all the necessary files needed to run the designed experiments in the local computer selected by the user. Now, you can forget scripts and others parameter files that made arduous the design of an experiment, and begin to use the new windows based interface.

With this program, you only need to select the input data (data sets), the algorithms you want to use and to make the opportune connections between them. Also it is possible to concatenate methods, insert statistical tests, etc...

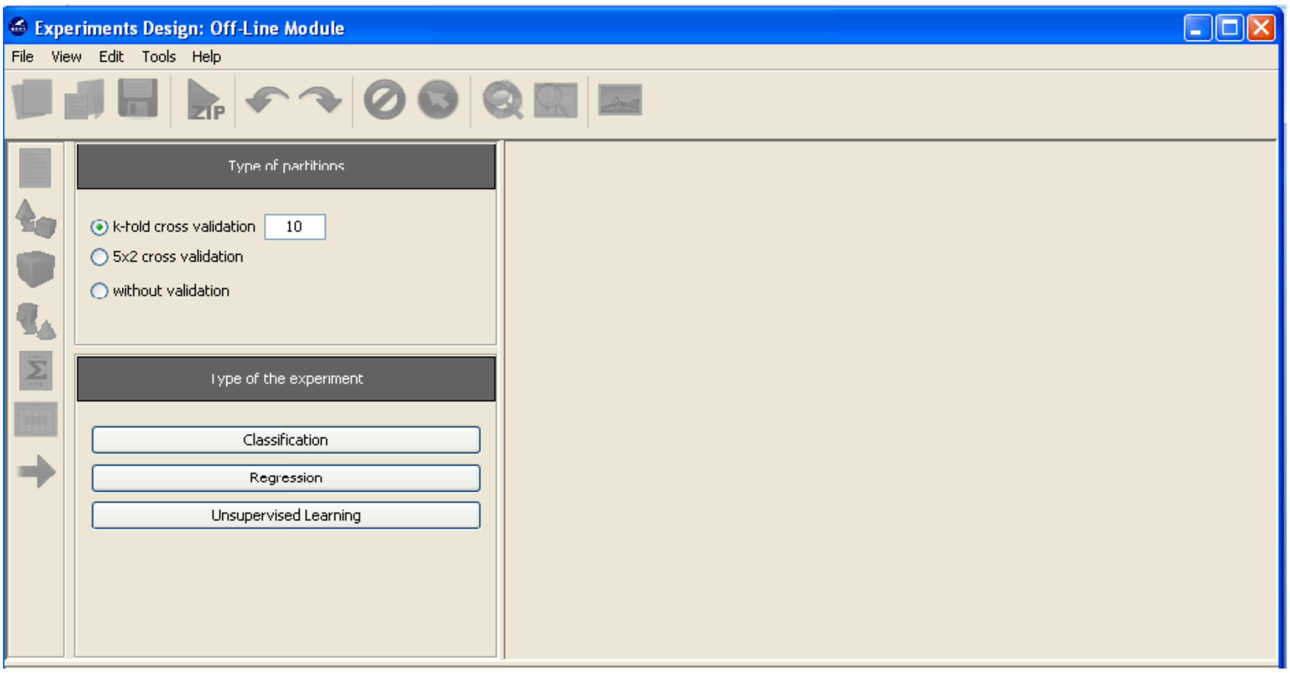
The task which is more simplified is probably the configuration of the parameters; everything can be done from a simple dialog without requirement of external configuration files.

This part of KEEL has two main objectives: on the one hand, you can use the software as a test and evaluation tool during the development of an algorithm. On the other hand, it is also a good option in order to compare new developments with standard algorithms already implemented and available in KEEL 1.0.

The interface allows the user to add new algorithms to the experiment being designed. The only requirement is to accept the input and output file format of KEEL (refer to them in the KEEL Reference Manual). This provides a very flexible way for the user to compare new methods with the ones in KEEL 1.0.

Configuration of experiments

When the *Experiments* option is selected, the main window of the Experiments module will appear:



Type of partitions

● k-fold cross validation 10

○ 5x2 cross validation

○ without validation

Type of the experiment

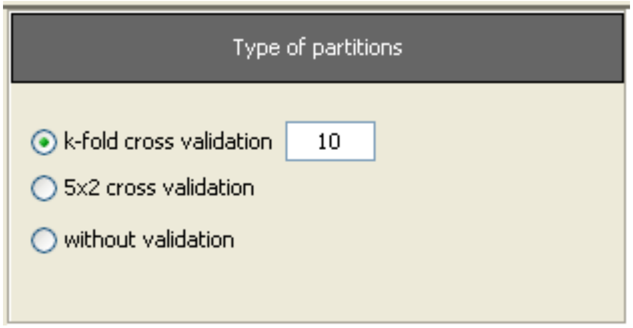
Classification

Regression

Unsupervised Learning

First, it is necessary to select the type of experiment and the type of partitions to employ; the options selected will determine the kind of methods and data sets that will be available to design the experiment.

The types of partitions available are the following:



Type of partitions

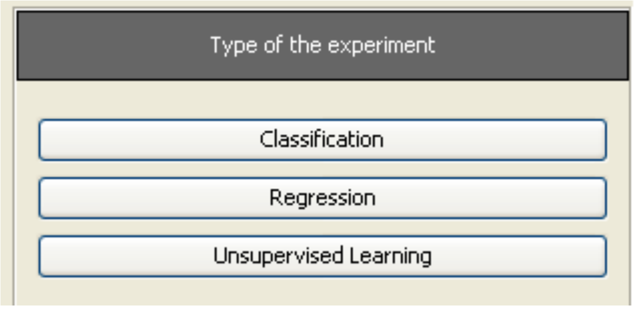
● k-fold cross validation 10

○ 5x2 cross validation

○ without validation

- k-fold cross validation (the value of k must be specified)
- 5x2 cross validation
- without validation

Currently, the KEEL Experiments module offers the following types of experiments:

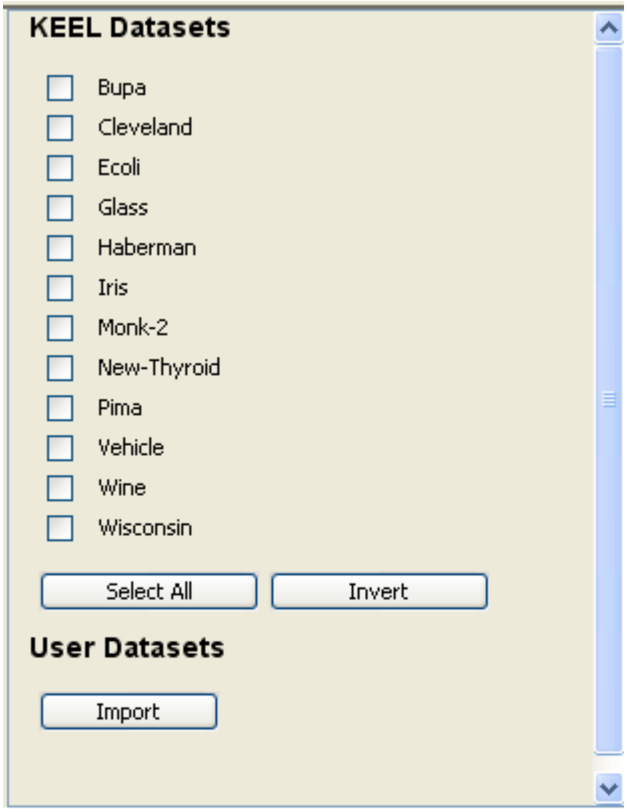


- Classification
- Regression
- Unsupervised Learning

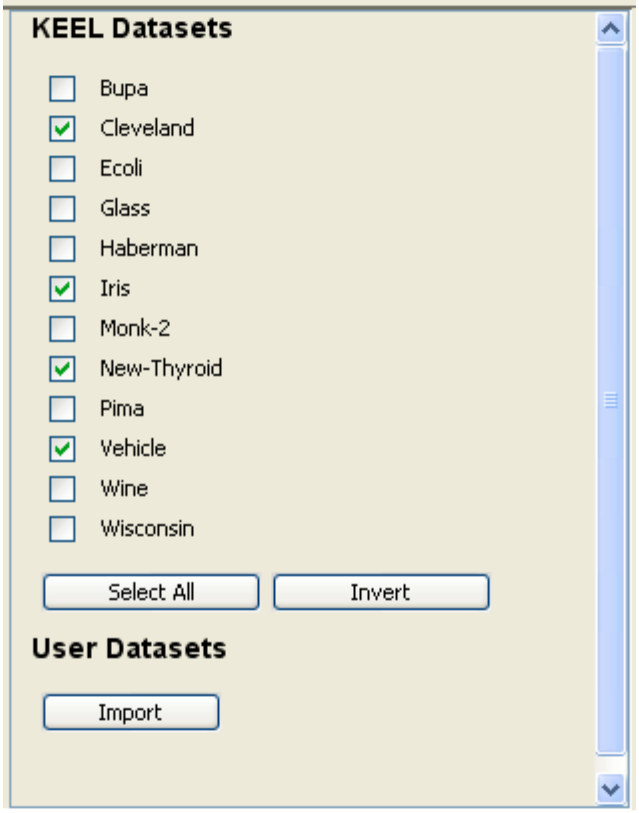
When the type of experiment has been selected, the data sets selection panel will be shown, allowing continuing the experiment design.

Selection of data sets

The data sets selection panel shows the available datasets for the current experiment. Its contents will depend of the type of experiment already selected:



The next step is to choose the wished data sets from the panel. The buttons *Select All* and *Invert* allows making the selection easily:



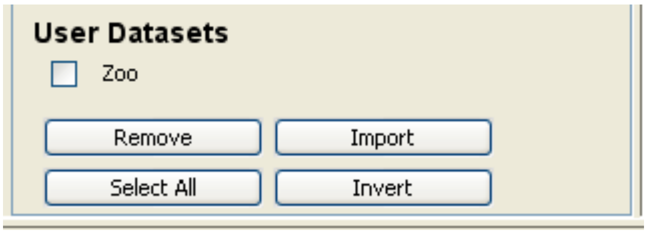
KEEL Datasets

- □ Bupa
- ☑ Cleveland
- □ Ecoli
- □ Glass
- □ Haberman
- ☑ Iris
- □ Monk-2
- ☑ New-Thyroid
- □ Pima
- ☑ Vehicle
- □ Wine
- □ Wisconsin

User Datasets

The Import Button allows importing an existing data set into the KEEL environment, ready to be selected for the current experiment. By clicking on it, the main window of the Data Import Tool will be shown. The process to import a new data set can is described in the Data Management module section of the manual.

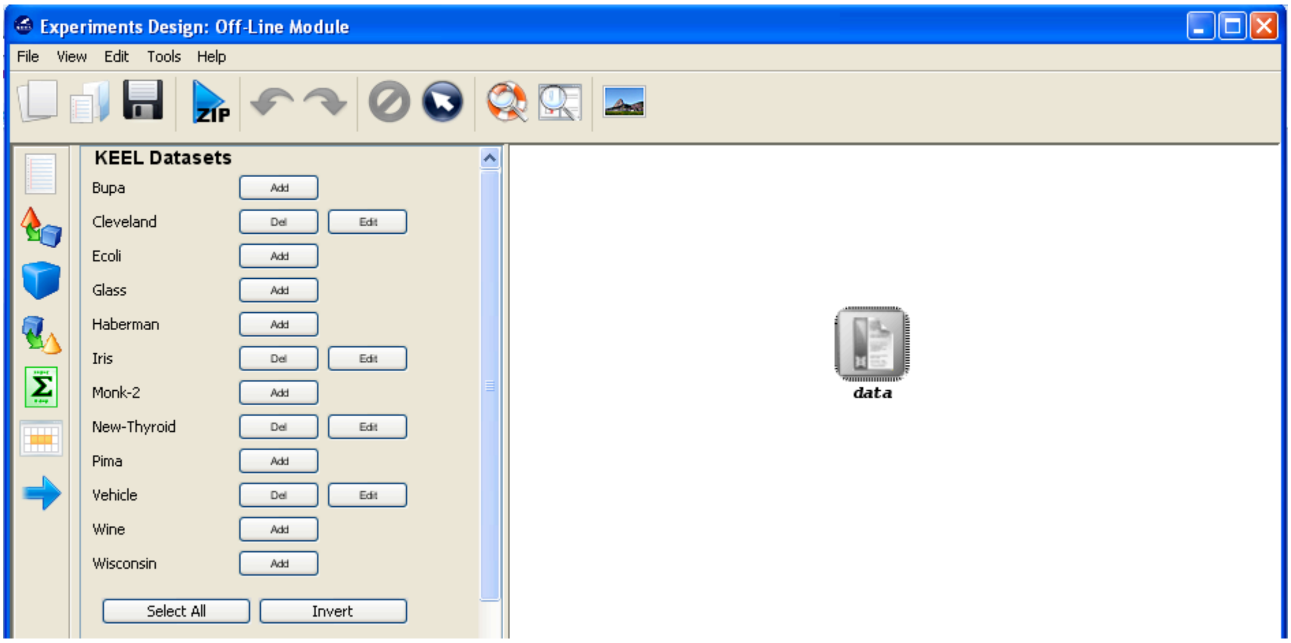
If a new data set is added, new buttons will appear allowing the user to *Invert* the current selection of User data sets, or to *Select All* of them. Furthermore, it is possible to add even more data sets (with the *Import* button), or to *Remove* the data sets selected.



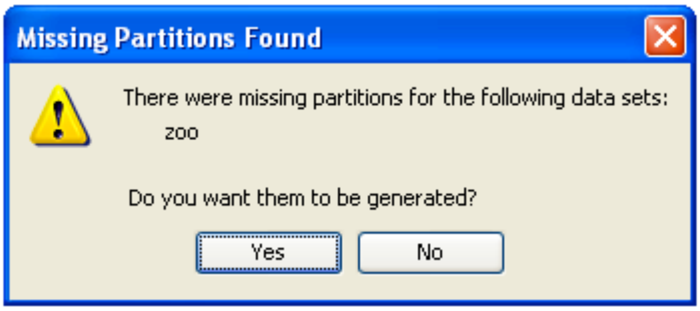
User Datasets

- □ Zoo

When all the necessary data set are selected, the experiment design process can continue. To do so, the user must click on the white graph panel to set the data sets node of the experiment.



At this point, the KEEL Experiments module will check if all the necessary partitions of the current selected data sets are present. If some missing partitions are found (e.g. if the user selected a k value different from the sets available in the standard distribution), the tool will prompt the following message:



Missing Partitions Found

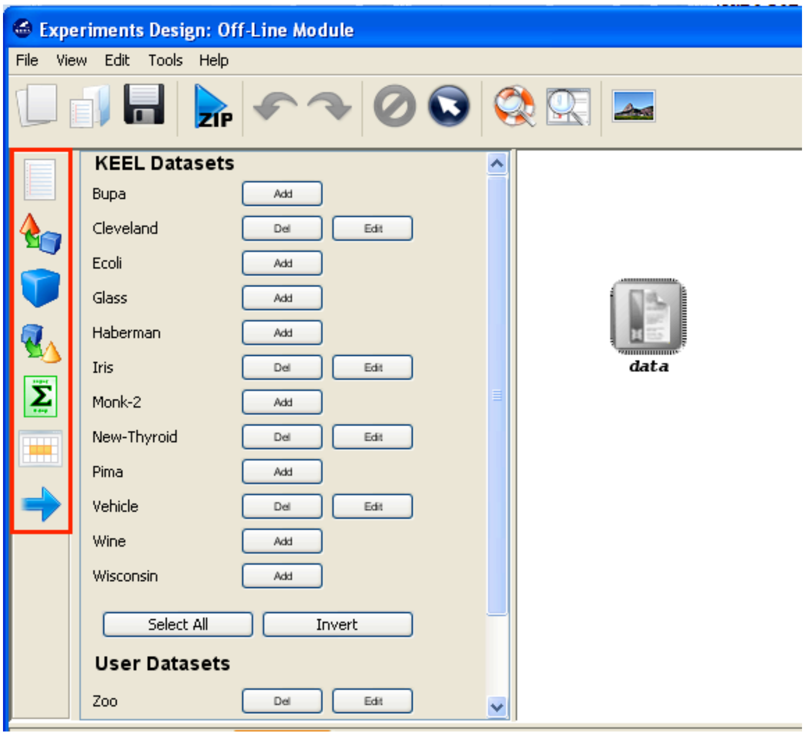
There were missing partitions for the following data sets:
zoo

Do you want them to be generated?

Clicking on yes will result on the generation of the missing partitions inside the KEEL environment. If the user selects to *No* generate the partitions, this warning will be shown again before the generation of the experiment graph.

Experiment Graph

The experiment graph shows the components of the current experiment and describes the relationships between them. The user can add new components by using the left menu:



This menu has the following categories available:

Data sets: Modify the data sets of the experiments.

Preprocessing methods: Preprocess over the initial data sets.

Standard methods: Data mining methods.

Postprocessing methods: Post-process over the results of standard methods.

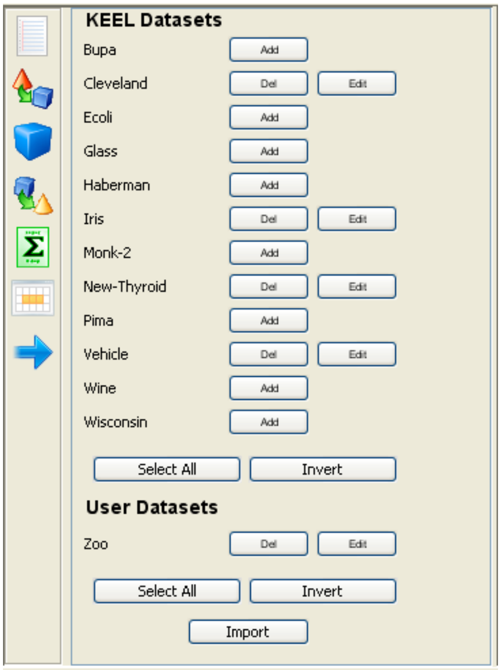
Statistical tests: Statistical procedures to contrast the results achieved in the experiment.

Visualization modules: Show the results of the experiments in an upgraded way.

Connections: Links between the components of the experiment.

Data sets

This module lets the user edit the current data sets selected for the experiment.



KEEL Datasets

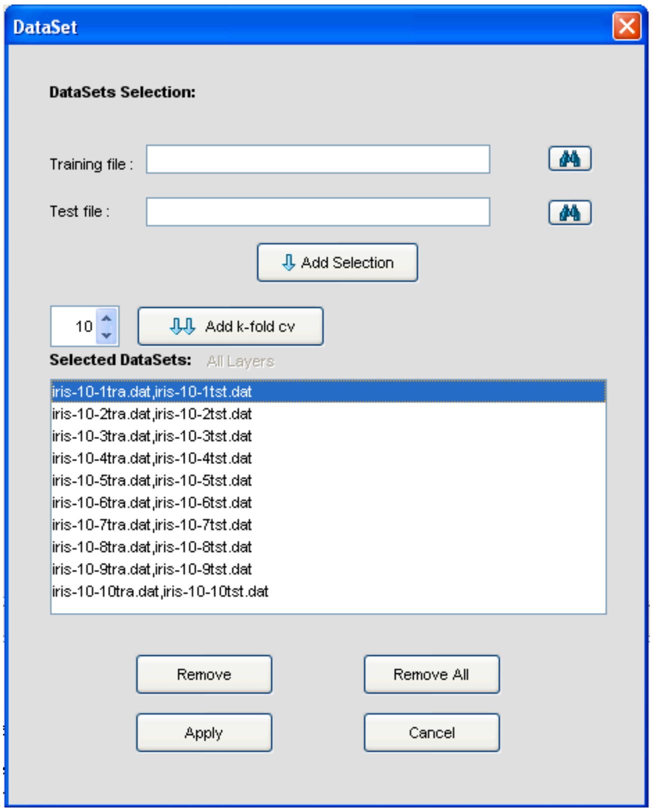
- Bupa
- Cleveland
- Ecoli
- Glass
- Haberman
- Iris
- Monk-2
- New-Thyroid
- Pima
- Vehicle
- Wine
- Wisconsin

User Datasets

- Zoo

As in the *Select Datasets* panel, the user can still *Add* and *Delete* data sets to the experiment (from those already registered in the KEEL environment). Also, it is still possible to import new data sets.

Furthermore, the button *Edit* allows the user to indicate which partitions (training and test) desires to use. This way, it is possible to temporally alter the files which will be included in the experiment.



DataSet

DataSets Selection:

Training file :

Test file :

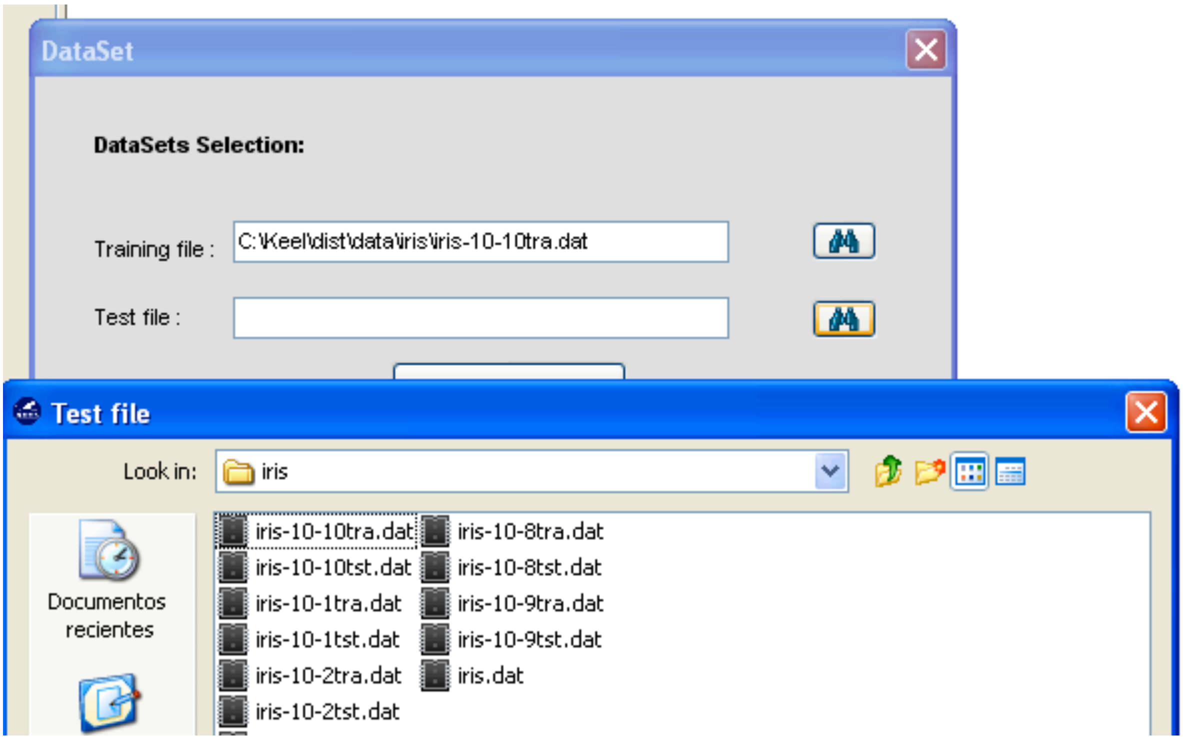
10

Selected DataSets: All Layers

```
iris-10-1tra.dat,iris-10-1tst.dat
iris-10-2tra.dat,iris-10-2tst.dat
iris-10-3tra.dat,iris-10-3tst.dat
iris-10-4tra.dat,iris-10-4tst.dat
iris-10-5tra.dat,iris-10-5tst.dat
iris-10-6tra.dat,iris-10-6tst.dat
iris-10-7tra.dat,iris-10-7tst.dat
iris-10-8tra.dat,iris-10-8tst.dat
iris-10-9tra.dat,iris-10-9tst.dat
iris-10-10tra.dat,iris-10-10tst.dat
```

This dialog shows the initial files of the data set. From it, is possible to *Remove* a pair of training a pair of training/test files, to *Remove All* files.

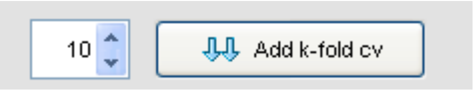
Also, the dialog allows to *Add* a new pairs of training and test files. To do so, they must be selected by using the search buttons:



Training file : C:\Keel\dist\data\iris\iris-10-10tra.dat

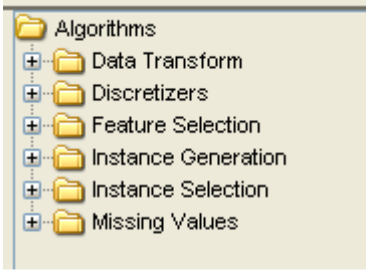
Test file :

Finally, it is also possible to add a complete set of k-fold cross validation files by selecting the adequate number of folds and pressing the button *Add k-fold cv*.



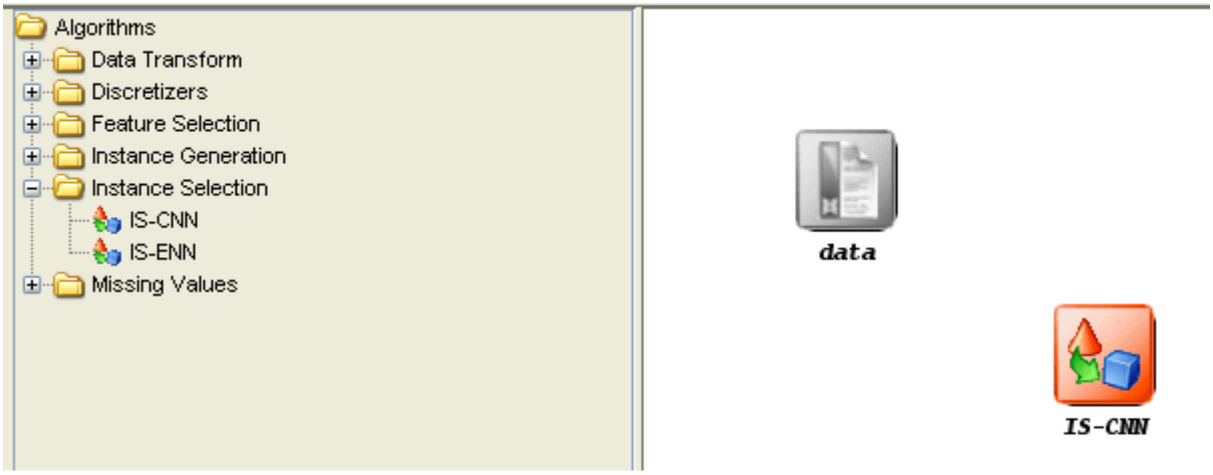
Preprocessing methods

This category includes several preprocessing methods



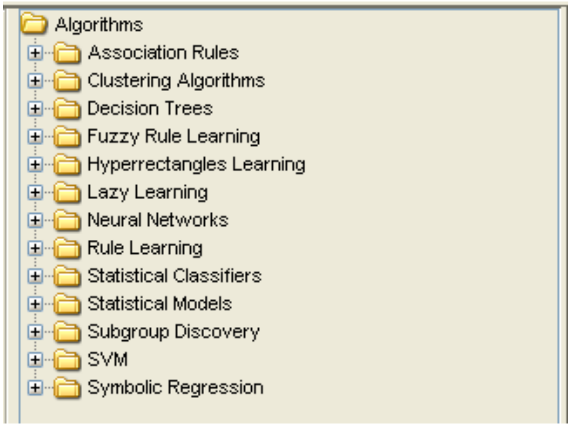
- **Data Transform:** Methods for transforming the format of data: Nominal to binary, decimal scaling….
- **Discretizers:** Method to convert real or numeric data into nominal data.
- **Feature Selection:** Methods to select features of the data.
- **Instance Generation:** Methods to generate new instances from the original instances of the data set.
- **Instance Selection:** Methods to select instances of the data.
- **Missing Values:** Methods to asses data containing missing values.

To add any preprocessing method to the current experiment, it is only needed to select it and click in the graph of the experiment:



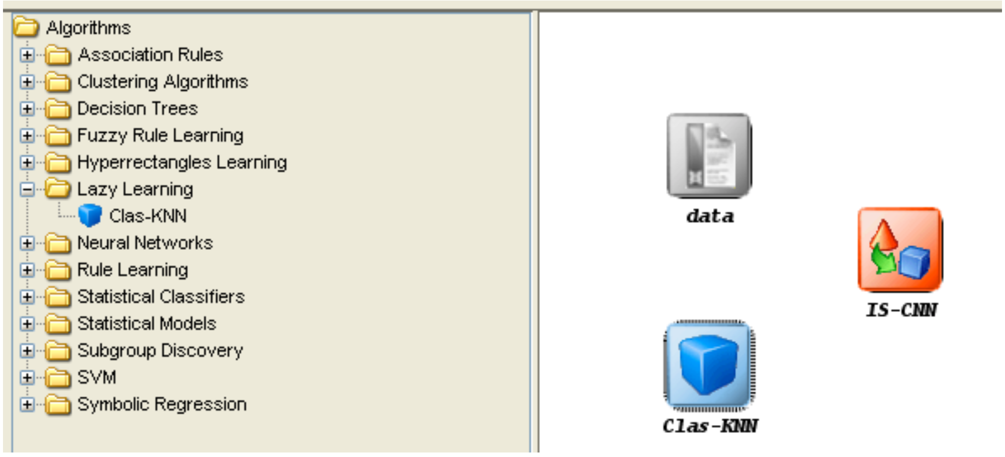
Standard methods

This category includes the data mining methods included in the KEEL tool:



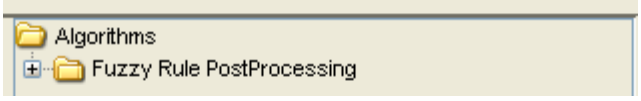
- **Association Rules**: Methods for extracting association rules from data.
- **Clustering Algorithms**: Clustering methods
- **Decision Trees**: Methods for building decision trees.
- **Fuzzy Rule Learning**: Methods for performing fuzzy rule-based learning.
- **Hyperrectangles Learning**: Methods using hyperrectangles to extract knowledge from data.
- **Lazy Learning**: Learning methods which do not build a model in its training phase.
- **Neural networks**: Artificial neural networks.
- **Rule Learning**: Methods for performing rule-based learning.
- **Statistical Classifiers**: Classifiers based on statistical models.
- **Statistical Models**: Construction of statistical models based on data.
- **Subgroup Discovery**: Methods for subgroup discovery.
- **SVM**: Support vector machines.
- **Symbolic regression**: Methods for performing symbolic regression procedures.

To add any method to the current experiment, it is only needed to select it and click in the graph of the experiment:



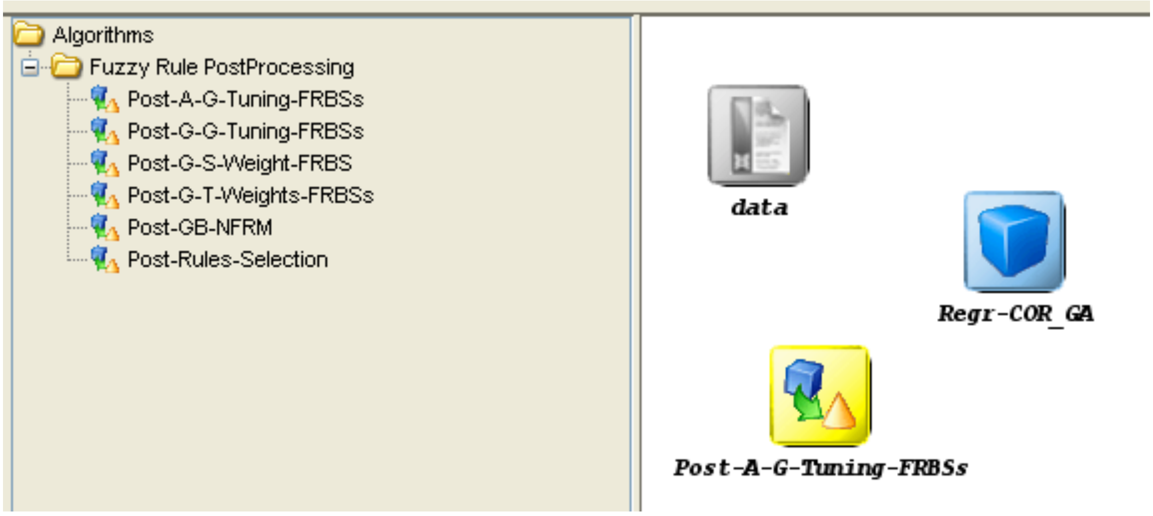
Postprocessing methods

This category includes the postprocessing methods included in the KEEL tool:



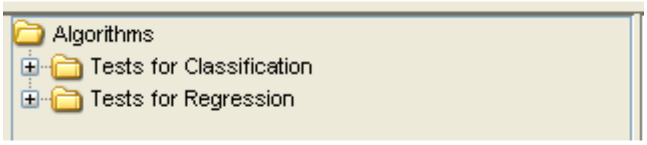
- **Fuzzy Rule PostProcessing**: Methods for performing a postprocess phase over the results of a Fuzzy Rule extraction method

To add any postprocessing method to the current experiment, it is only needed to select it and click in the graph of the experiment:



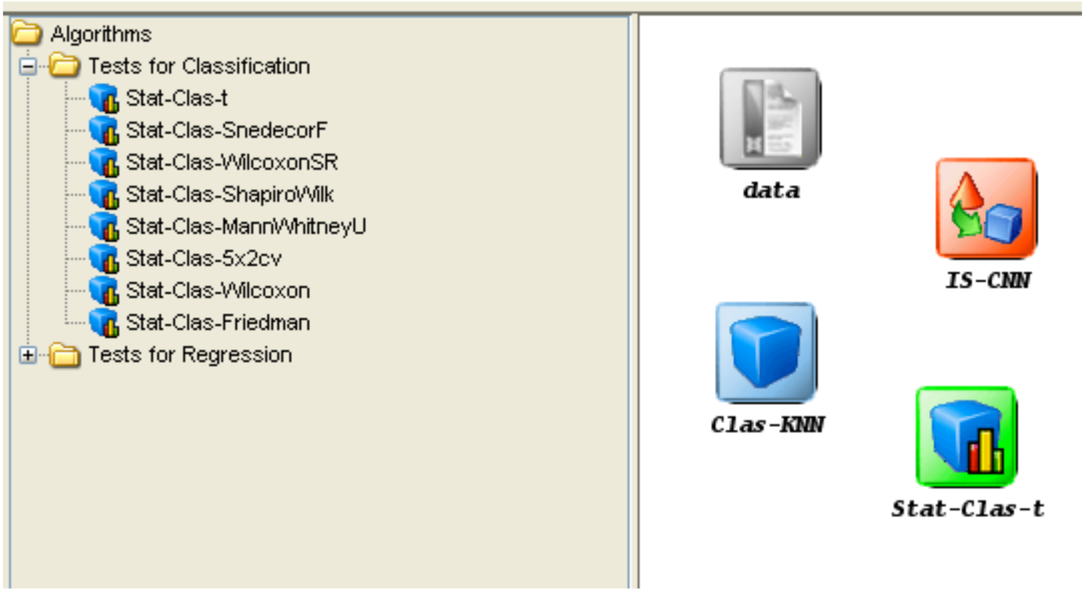
Statistical tests

This category includes several statistical modules available to contrast experiments performed with the KEEL software tool:



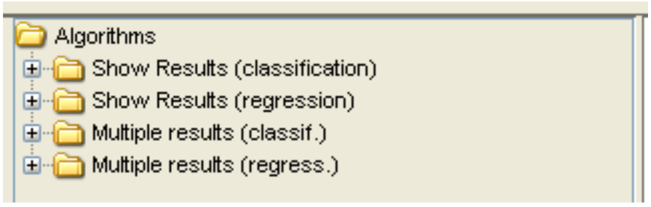
- **Tests for Classification**: Statistical procedures for contrasting the results of classification experiments.
- **Tests for Regression**: Statistical procedures for contrasting the results of regression experiments.

To add any statistical procedure to the current experiment, it is only needed to select it and click in the graph of the experiment:



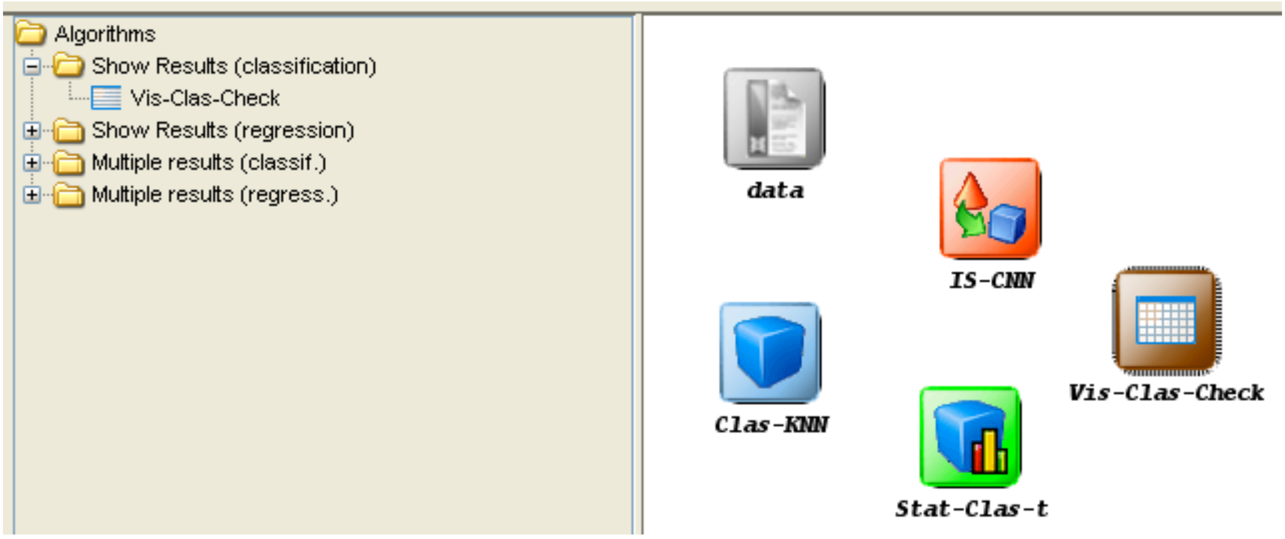
Visualization modules

This category includes several visualization modules developed to analyze and summarize the results achieved in the experiments:


Algorithms
Show Results (classification)
Show Results (regression)
Multiple results (classif.)
Multiple results (regress.)

- **Show Results (classification)**: Modules for summarizing results achieved in classification problems.
- **Show Results (regression)**: Modules for summarizing results achieved in regression problems.
- **Multiple Results (classification)**: Modules for analyzing results achieved in classification problems with multiple algorithms.
- **Multiple Results (regression)**: Modules for analyzing results achieved in regression problems with multiple algorithms.

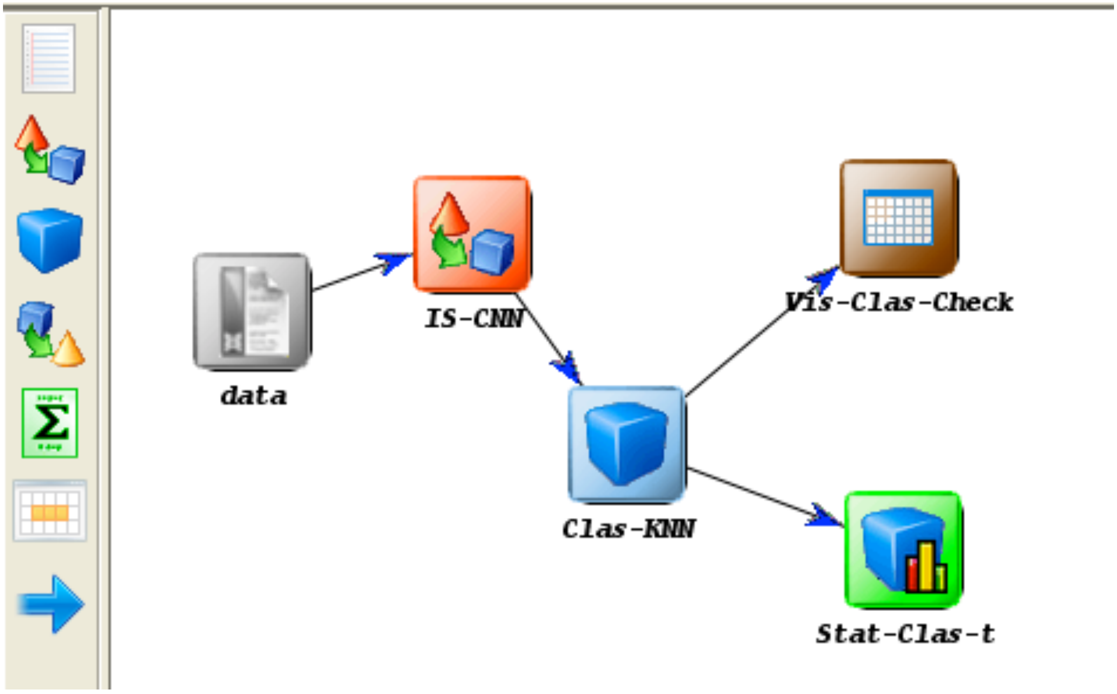
To add any visualization module to the current experiment, it is only needed to select it and click in the graph of the experiment:


Algorithms
Show Results (classification)
Vis-Clas-Check
Show Results (regression)
Multiple results (classif.)
Multiple results (regress.)
data
IS-CNN
Vis-Clas-Check
Clas-KNN
Stat-Clas-t

Connections

The connections allow finishing the designing of the experiment, by connecting the included modules with flows which represent the data flow in the experiment. They can be used both as inputs or outputs of the modules.

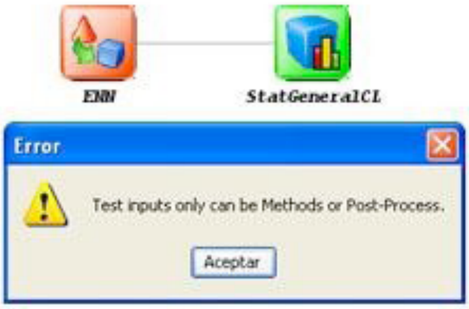
1. **Insert connection:** to make a connection, select the button from the left tool bar. Then, click on the source node and finally click on the target node.



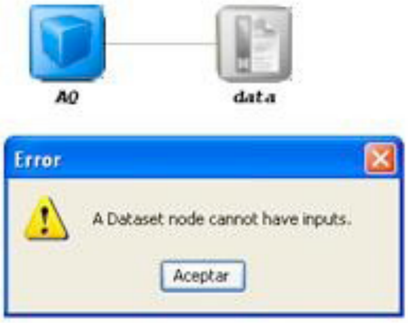
2. **Restrictions:** there are some restrictions that must be considered when making connections between the different elements:

   - A dataset cannot have inputs.
   - The pre-processing algorithms can only receive inputs from a data set or another pre-process method.
   - Knowledge extraction methods can receive a flow from a data set, from a pre-processing algorithm or from a previous method.
   - The test and visualization modules must receive input data from a method or from a post-processing algorithm.
   - Test and visualization modules cannot have outputs.
   - The graph cannot have any cycle.

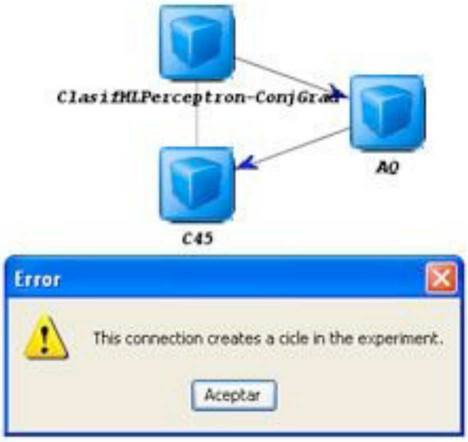
All these restrictions are verified in execution time when a connection is been created. If one of these connections is not allowed, the application will show an error message. In next figures some examples of incorrect graphs are shown:



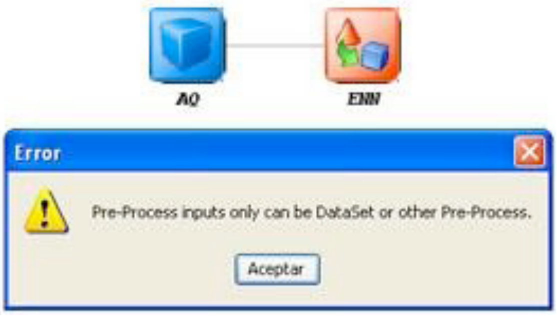
Test inputs only can be Methods or Post-Process.



A Dataset node cannot have inputs.



This connection creates a cicle in the experiment.

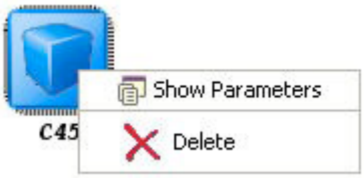


Pre-Process inputs only can be DataSet or other Pre-Process.

Graph Management

The graph allows performing the following operations over its elements:

1. **Context menu:** it is possible to access to the context menu by clicking with the right button of the mouse on a certain node in the draw area. This menu depends on the selected object and allows us to remove objects, to configure algorithms parameters, etc...



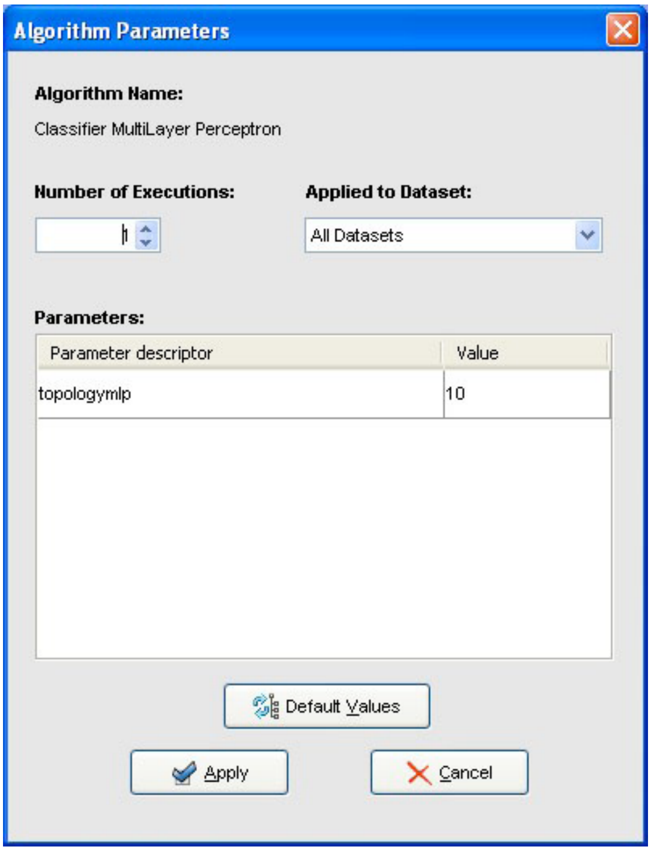
2. **Objects selection:** in order to select a single element, just click with the left button of the mouse over it. But it is possible to select several elements, clicking in an empty zone of the draw area and dragging the mouse until covering all the objects wished.

3. **Move objects:** It is possible to move one or several elements selected with the aid of the left button of the mouse, dragging them to the desired position. Another way is to use the keyboard cursors.

4. **Remove objects:** To remove a module, select it and press *Supr* key. It is also possible to remove it by the context menu, of from the tool bar.

Algorithm parameters configuration

Once a module has been inserted in the graph, it is possible to configure the value of its parameters. To do so, the user have to double click on the algorithm symbol and a dialog will be shown; also, this dialog can be shown through the emergent menu that will appear when right button of the mouse is pressed (option Show Parameters).



Algorithm Parameters

Algorithm Name:

Classifier MultiLayer Perceptron

Number of Executions: 1

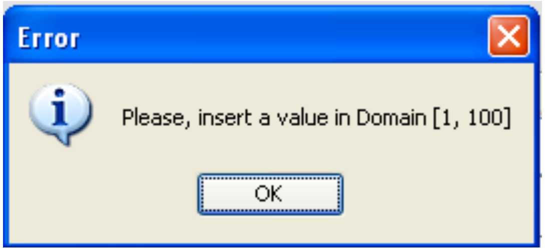
Applied to Dataset: All Datasets

Parameters:

Parameter descriptor	Value
topologymlp	10

At the top of this dialog it is possible to set the number of times that the algorithm will be executed (only available for random methods). Each execution will be made using a seed generated from the initial seed. The second list allows specifying in which data sets the parameters will be changed.

In the table located in the center of the window, all the algorithm parameters are established to its initial values. These values can be modified, as far as the new values will be appropriate for the concrete method; otherwise, an error message will appear:



Error

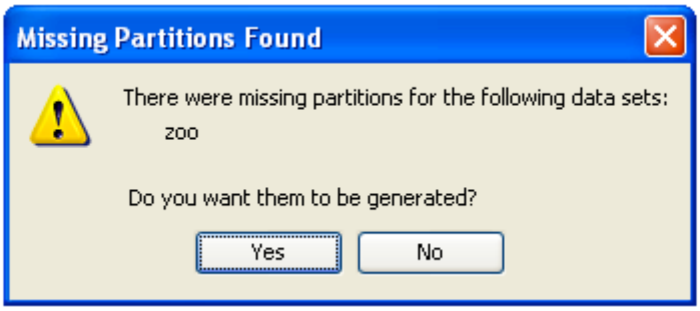
Please, insert a value in Domain [1, 100]

Finally, the *Default Values* button allows returning all parameter to its default values.

Generation of Experiments

Once a experiment has been designed, the user can generate it through the option Run Experiment of the 'Tools' menu. Furthermore, it is possible to use the tools bar button.

At this point, the software tool will perform several test about the completeness of the experiment. Firstly, if it detects that there are missing partitions for some of the data sets employed, the following dialog will be shown, allowing to regenerate them:



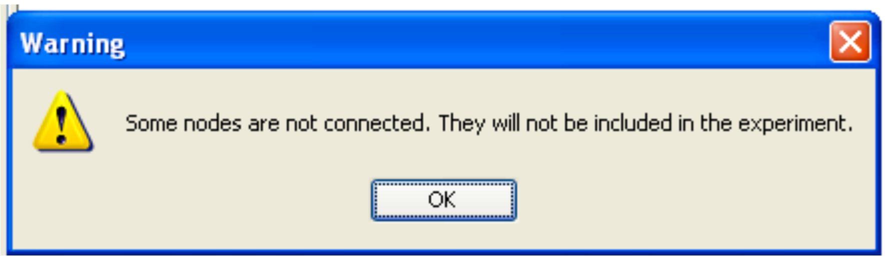
Missing Partitions Found

There were missing partitions for the following data sets:
zoo

Do you want them to be generated?

This is the last opportunity to generate them. Else, the experiment will be generated incorrectly.

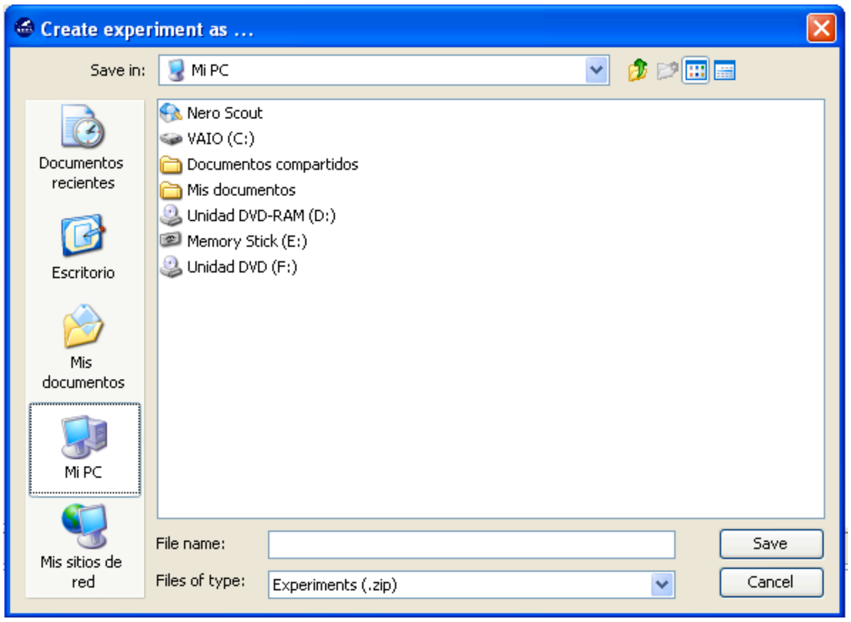
Secondly, if some of the elements of the graph are not connected by flows, the following warning will be prompt, and the isolated nodes will be discarded.



Warning

Some nodes are not connected. They will not be included in the experiment.

If everything is correct, the user will have to select a path for the experiment’s zip file:



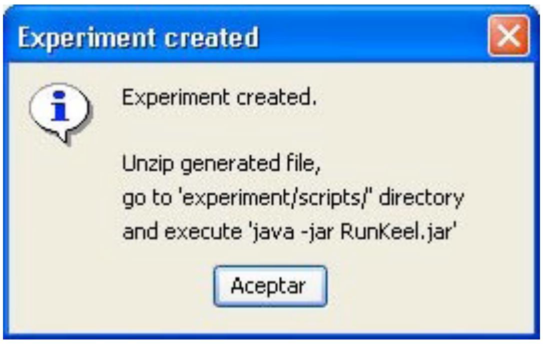
Create experiment as ...

Save in: Mi PC

File name:

Files of type: Experiments (.zip)

The generation process generates a ZIP file containing all the elements needed to run the experiment. If the experiment generation is completed successfully, the following message will be shown.



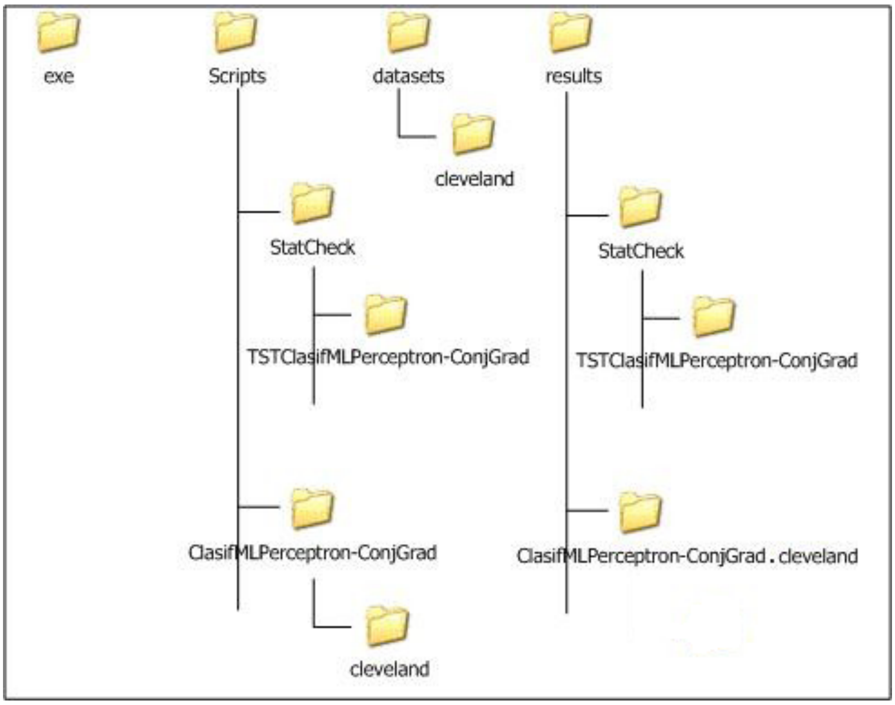
Experiment created

Experiment created.

Unzip generated file,
go to 'experiment/scripts/' directory
and execute 'java -jar RunKeel.jar'

The experiment must be run using the *RunKeel* jar file located at "experiment/scripts"
In the following picture, we can see an example of the structure of directories that is created. We see that four directories are created:

- ***exe*** directory**:** it contains all the jar files of the algorithms inserted in the experiment.
- ***scripts*** directory**:** it contains the configuration files sorted by algorithm. Also, it contains the *RunKeel.jar* file which is used in order to run the experiment.
- ***datasets*** directory: it contains the used in the experiment. A directory for each dataset is created.
- ***results*** directory: it contains the output files generated by each algorithm.

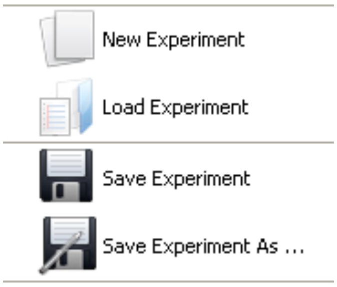


exe
Scripts
datasets
results
cleveland
StatCheck
StatCheck
TSTClasifMLPerceptron-ConjGrad
TSTClasifMLPerceptron-ConjGrad
ClasifMLPerceptron-ConjGrad
ClasifMLPerceptron-ConjGrad.cleveland
cleveland

Menu bar

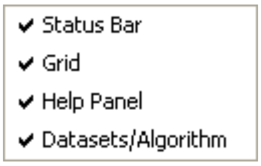
Each item of the menu bar contains different submenus. These are the different options available:

1. *File Menu*



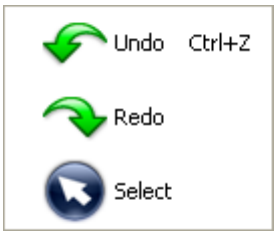
- **New Experiment:** creates a new experiment.
- **Load Experiment:** opens an existing experiment. In the open dialog box, select a filename, and click Open. Experiments files usually are saved in XML format.
- **Save Experiment:** saves the current experiment to a XML file. If it is the first time that the experiment is saved, you will be asked about destination path.
- **Save Experiment As:** saves current experiment to a XML file. You will be asked about destination path.
- **Exit:** closes the experiment design tool. If the experiment has not been saved yet, you can do it at this moment.

2. *View Menu*



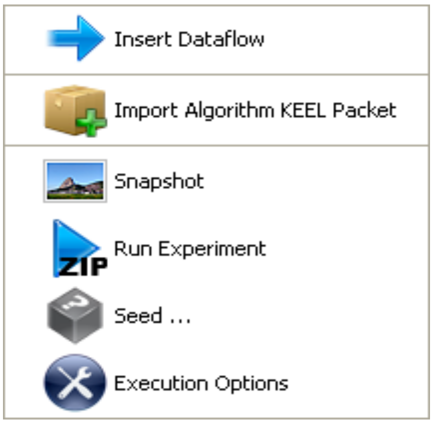
- **Status Bar:** shows/hides the status bar (at the bottom of the windows). Initially, it is active.
- **Grid:** shows/hides the alignment grid. It helps the user to make easy the alignment of the elements inserted in the draw area. Initially, it is inactive.
- **Help Panel:** shows/hides the help panel. Initially, it is active.
- **DataSets/Algorithm:** shows/hides the panel containing the datasets/algorithms. Initially, it is active.

3. Edit Menu



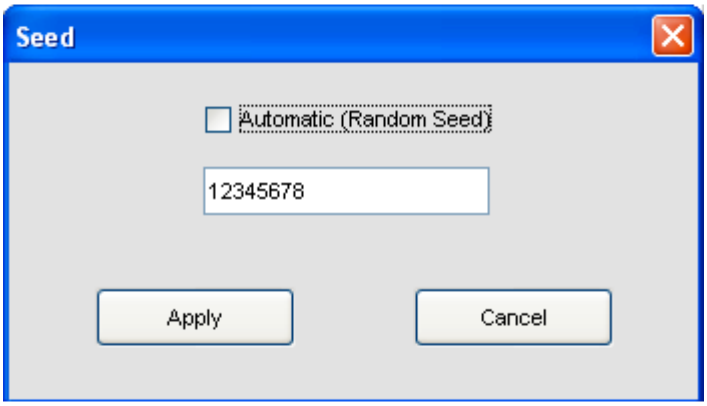
- **Undo:** with this option the user can undo some actions.
- **Redo:** with this option user can redo some undone actions.
- **Select:** allows users to select one or more elements in the draw area.

4. Tools Menu



- **Insert Dataflow:** allows connecting algorithm outputs (or dataset) to the inputs of another algorithm, creating a data flow that will be run later. This option is also available from the left bar via the button ➡
- **Import algorithm KEEL packet:** in order to insert a user's method, select this option and explore the path to choose the method.
- **Snapshot:** it is possible to save the experiment design into an image format file (JPEG). This way allows you to insert it in any document, article, etc...
- **Run Experiment:** when experiment is fully designed, use this option to create a ZIP file containing a directory structure with all the files needed to run the designed experiment in the local computer selected by the user.
- **Seed:** sets up the value of the seed used by the random number generator. If there is any algorithm (inserted in the experiment) that needs to generate random numbers, it will take in a seed created from the initial seed value. This seed can be established

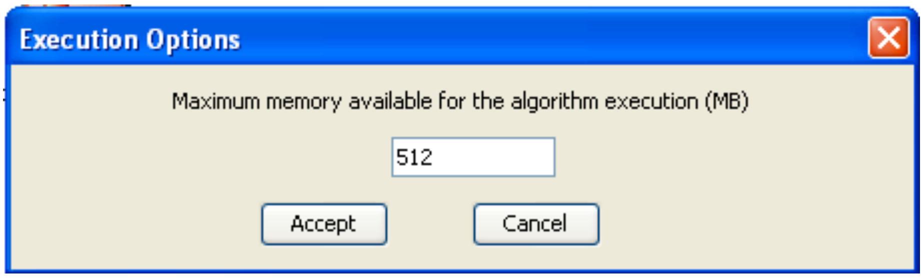
automatically or you can insert a value manually. The following picture shows the dialog prompted by this option:



□ Automatic (Random Seed)

12345678

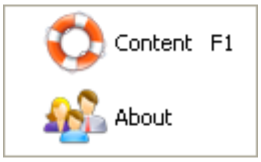
- **Execution Options:** allows selecting some performance options to apply to the experiment. In this version, the option defined is the following:
  - *Java Heap Size*: Indicate the number of MB that will be allocated in each execution of the algorithm. Default value is 512MB. Please do not set a higher value than your actual amount of RAM. The *minimum* accepted value has been set to 32MB.



Maximum memory available for the algorithm execution (MB)

512

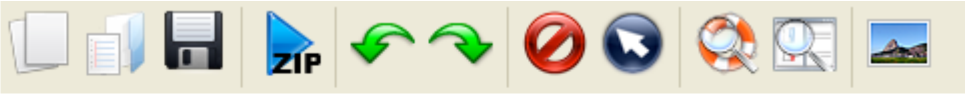
5. *Help Menu*



- **Content:** show a help dialog that contains information about how to use this program.
- **About ...:** shows a dialog with basic information about the program as name, authors, version, etc...

Tool bar

To help the user finding the most relevant operations, the KEEL Experiments software tool provides a tool bar with shortcuts to them.



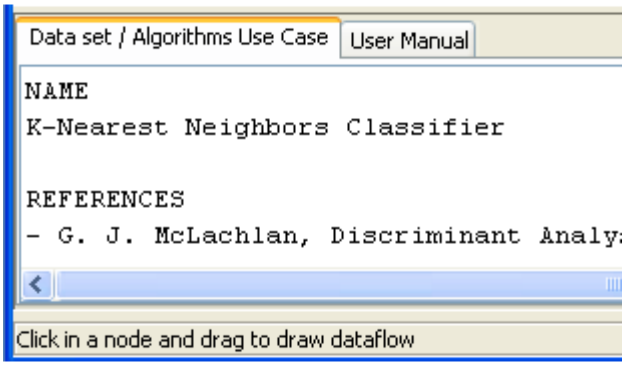
Most of them also appear in the Menu bar (thus, refer to the Menu bar section to get additional information about them). The only option that does not appears already in the Menu bar is:

Delete: This option allows the user to delete the selected module.

Status bar

The status bar is a simple way to provide the user useful information during the generation of experiments.

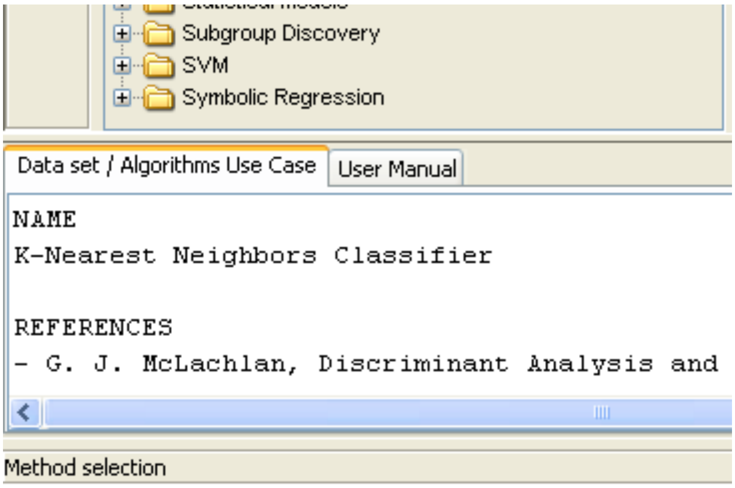
It is located at the bottom of the window. Here it will appear information about the action being carried out, helping the user to understand the meaning of each command or button. Several examples are shown below:



NAME
K-Nearest Neighbors Classifier

REFERENCES
- G. J. McLachlan, Discriminant Analy

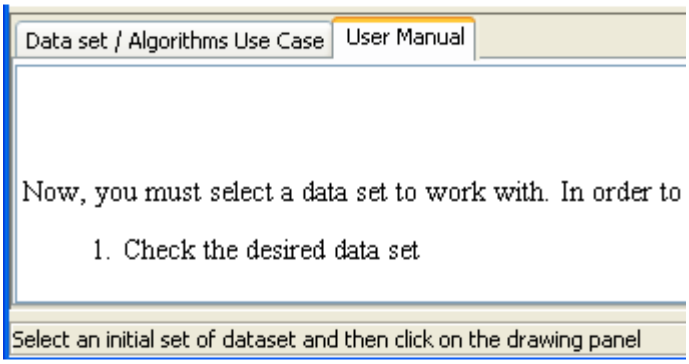
Click in a node and drag to draw dataflow



NAME
K-Nearest Neighbors Classifier

REFERENCES
- G. J. McLachlan, Discriminant Analysis and

Method selection



Now, you must select a data set to work with. In order to

1. Check the desired data set

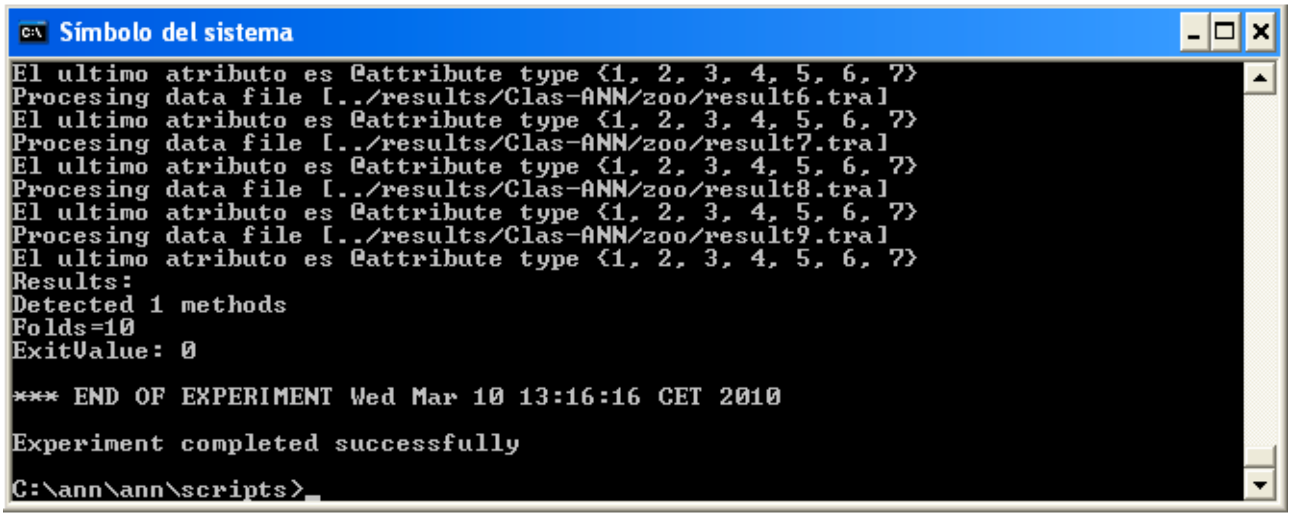
Select an initial set of dataset and then click on the drawing panel

RUN KEEL

Launching RunKeel

When the user designs an experiment, a .ZIP file will be obtained containing all the files needed to run the experiment in any computer that has a Java Virtual Machine installed. It is necessary to extract the content of the compressed file and then you will obtain a directory called *"experiment"* (as you had called your experiment). Inside it, there is a new directory called "scripts" in which you can launch the RunKEEL application by typing **java -jar RunKeel.jar** in a console. Then, the experiment begins to run.

When it finishes, the user will obtain at the command prompt the message **"EXPERIMENT COMPLETED SUCCESSFULLY"**



```
El ultimo atributo es @attribute type {1, 2, 3, 4, 5, 6, 7}
Procesing data file [../results/Clas-ANN/zoo/result6.tra]
El ultimo atributo es @attribute type {1, 2, 3, 4, 5, 6, 7}
Procesing data file [../results/Clas-ANN/zoo/result7.tra]
El ultimo atributo es @attribute type {1, 2, 3, 4, 5, 6, 7}
Procesing data file [../results/Clas-ANN/zoo/result8.tra]
El ultimo atributo es @attribute type {1, 2, 3, 4, 5, 6, 7}
Procesing data file [../results/Clas-ANN/zoo/result9.tra]
El ultimo atributo es @attribute type {1, 2, 3, 4, 5, 6, 7}
Results:
Detected 1 methods
Folds=10
ExitValue: 0

*** END OF EXPERIMENT Wed Mar 10 13:16:16 CET 2010

Experiment completed successfully

C:\ann\ann\scripts>_
```

View results

Once the run of experiment has finished, the result files are can be found at the **results** directory.

Depending of the type of methods used in the experiment, the following directories and files will be available:

- **Methods**: For each combination of a method and a data set, there will be a directory, named <methodName>**.**<datasetName>.

  Inside, it is possible to find the output files of the method (generally, a training and a test output file for each partition, plus every additional output file defined by the method). For further reference, see the KEEL Reference Manual .

- **Tests:** For each test module employed, a new directory named with the name of the test will be available. This directory will contain the output files obtained as a result of the application of the test method.

On the other hand, note that the new data sets obtained as the result of the execution of a preprocessing method will be placed in the **datasets** directory of the experiment, to allow a further employment of them with linked methods in the same experiment.

EDUCATIONAL

Introduction

KEEL is a software tool developed to build and use different Data Mining models. We would like to remark that this is the first software tool of this type containing a free code Java library of Evolutionary Learning Algorithms. The main features of KEEL are:

- It contains pre-processing algorithms: transformation, discretization, instance selections and feature selections.
- It also contains a Knowledge Extraction Algorithms Library, supervised and unsupervised, remarking the incorporation of multiple evolutionary learning algorithms.
- It has a statistical analysis library to analyze algorithms.
- It contains an user-friendly interface, oriented to the analysis of algorithms.
- KEEL's environment can connect to Internet to download new data files for using them in future analysis.

We can distinguish three parts in the graphic environment:

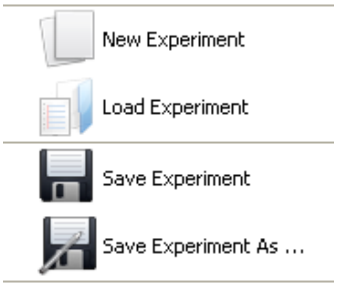
- The Preparation of the Data Bases part allow users to create different partitions of his own data bases or the data bases available in the KEEL web. Also, it is possible to edit, apply transformations, generate DataSets in the correct format from C4.5 files or view datailed plots about a concrete DataSet.
- The Design of Experiments part has the objective of designing the desired experiments using a graphical interface. After the experiment is designed, the interface generates a .ZIP file containing a directory structure with all the necessary files needed to run those experiments in the local computer

  The interface also allows the user to add his own algorithms to the experimentation being designed. The only requirement is to accept the input file format of KEEL. Even, it is not needed to use the Java language for the own algorithms of the user. This provides a very flexible way for the user to compare his own methods with the ones in KEEL.
- The Generation of Evolutionary Algorithms with the JCLEC library allows the user to create his own evolutionary algorithms using a graphical interface. In this version of KEEL, this part is NOT implemented.

Menu Bar

Each item of the menu bar contains different submenus. These are the different options available:

1. *File Menu*



2. **New Experiment:** creates a new experiment.

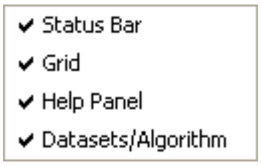
3. **Load Experiment:** opens an existing experiment. In the open dialog box, select a filename, and click Open. Experiments files usually are saved in XML format.

4. **Save Experiment:** saves the current experiment to a XML file. If it is the first time that the experiment is saved, you will be asked about destination path.

5. **Save Experiment As:** saves current experiment to a XML file. You will be asked about destination path.

6. **Exit:** closes the experiment design tool. If the experiment has not been saved yet, you can do it at this moment.

3. View Menu



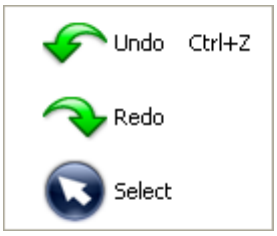
- **Status Bar:** shows/hides the status bar (at the bottom of the windows). Initially, it is active.

- **Grid:** shows/hides the alignment grid. It helps the user to make easy the alignment of the elements inserted in the draw area. Initially, it is inactive.

- **Help Panel:** shows/hides the help panel. Initially, it is active.

- **DataSets/Algorithm:** shows/hides the panel containing the datasets/algorithms. Initially, it is active.

4. Edit Menu

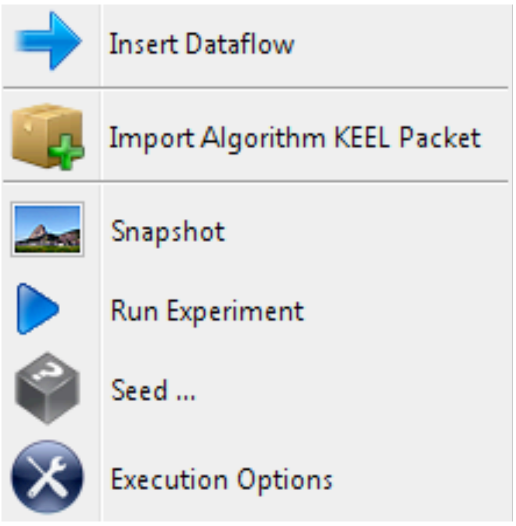


- **Undo:** with this option the user can undo some actions.

- **Redo:** with this option user can redo some undone actions.

- **Select:** allows users to select one or more elements in the draw area.

5. Tools Menu



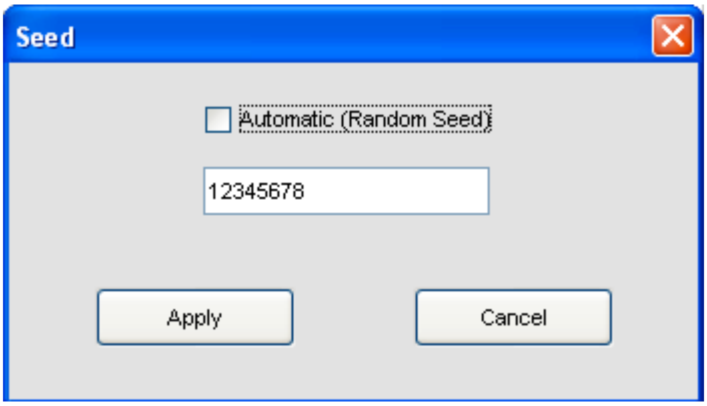
- **Insert Dataflow:** allows connecting algorithm outputs (or dataset) to the inputs of another algorithm, creating a data flow that will be run later. This option is also available from the left bar via the button ➡

- **Import algorithm KEEL packet:** in order to insert a user's method, select this option and explore the path to choose the method.

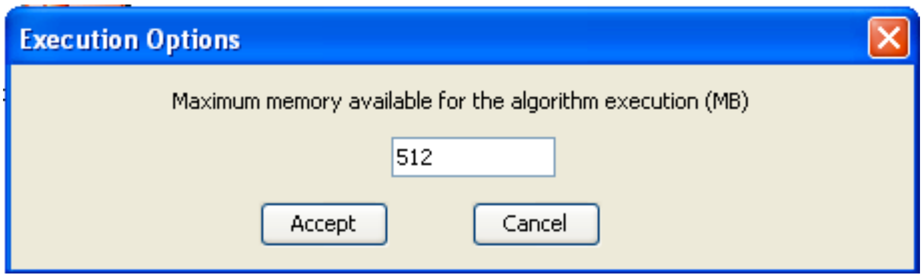
- **Snapshot:** it is possible to save the experiment design into an image format file (JPEG). This way allows you to insert it in any document, article, etc...

- **Run Experiment:** when experiment is fully designed, use this option to run the designed experiment and view the results.

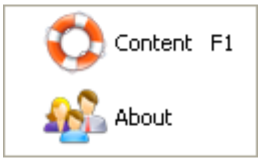
- **Seed:** sets up the value of the seed used by the random number generator. If there is any algorithm (inserted in the experiment) that needs to generate random numbers, it will take in a seed created from the initial seed value. This seed can be established automatically or you can insert a value manually. The following picture shows the dialog prompted by this option:



- **Execution Options:** allows selecting some performance options to apply to the experiment. In this version, the option defined is the following:
  - *Java Heap Size*: Indicate the number of MB that will be allocated in each execution of the algorithm. Default value is 512MB. Please do not set a higher value than your actual amount of RAM. The *minimum* accepted value has been set to 32MB.



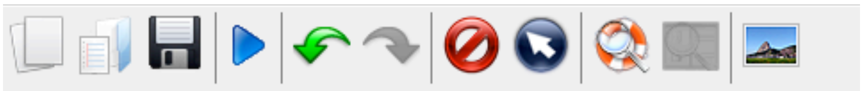
6. *Help Menu*



- **Content:** show a help dialog that contains information about how to use this program.
- **About ...:** shows a dialog with basic information about the program as name, authors, version, etc...

Tool bar

To help the user finding the most relevant operations, the KEEL Experiments software tool provides a tool bar with shortcuts to them.



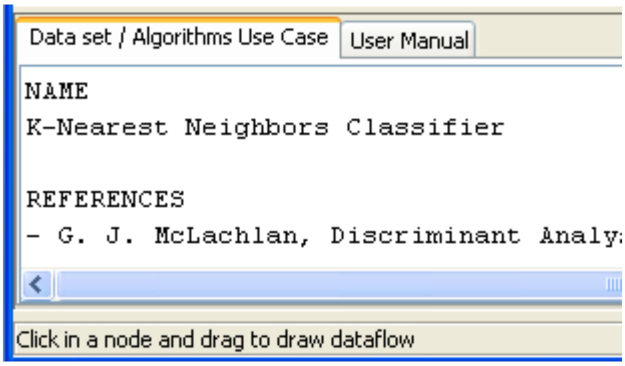
Most of them also appear in the Menu bar (thus, refer to the Menu bar section to get additional information about them). The only option that does not appears already in the Menu bar is:

🚫 **Delete**: This option allows the user to delete the selected module.

Status bar

The status bar is a simple way to provide the user useful information during the generation of experiments.

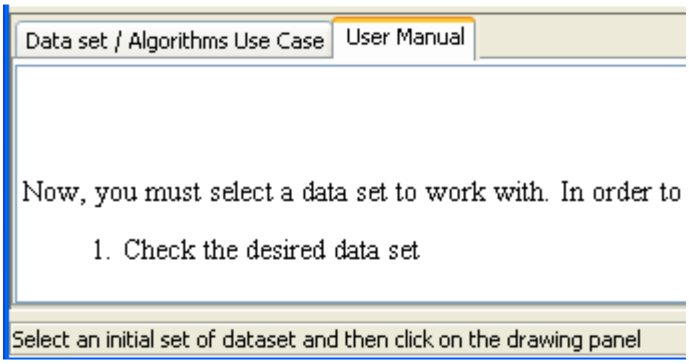
It is located at the bottom of the window. Here it will appear information about the action being carried out, helping the user to understand the meaning of each command or button. Several examples are shown below:



```
NAME
K-Nearest Neighbors Classifier

REFERENCES
- G. J. McLachlan, Discriminant Analy
```

Click in a node and drag to draw dataflow



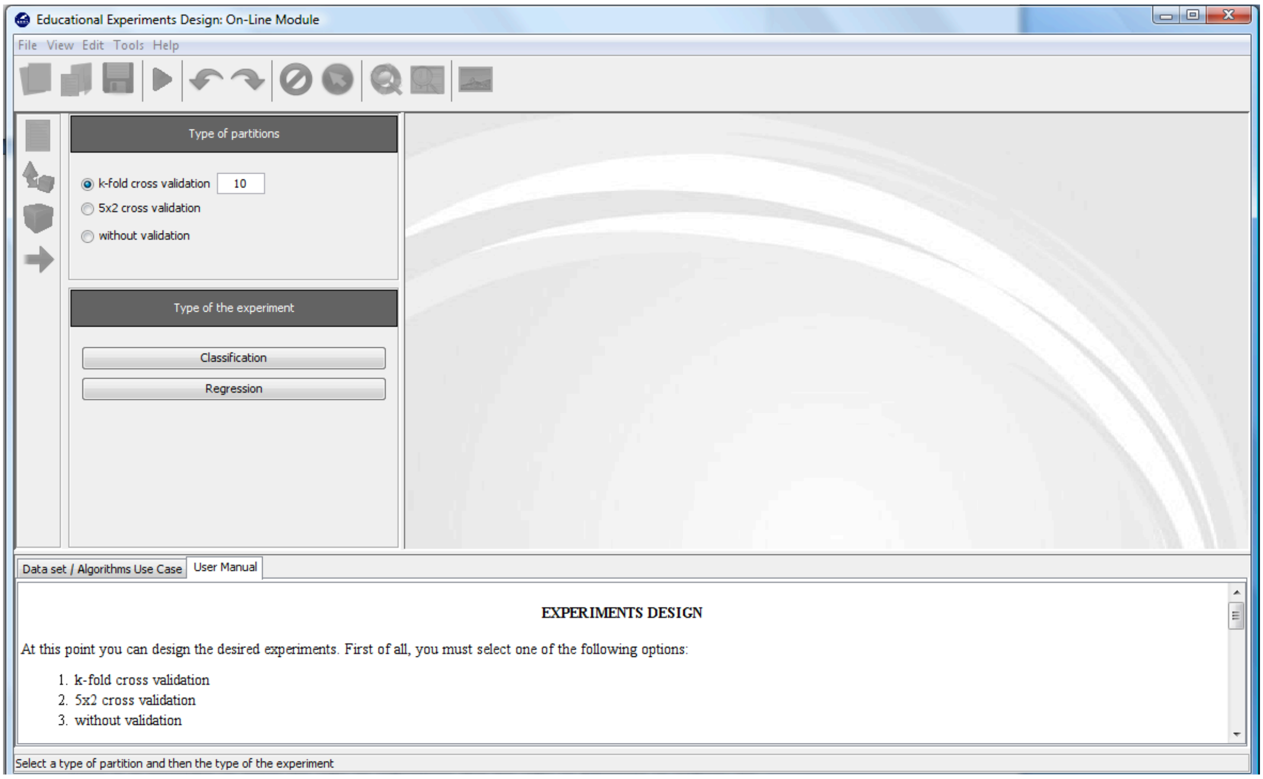
Now, you must select a data set to work with. In order to

1. Check the desired data set

Select an initial set of dataset and then click on the drawing panel

Configuration of experiments

When the *Educational* option is selected, the main window of the Experiments module will appear:


Educational Experiments Design: On-Line Module
File View Edit Tools Help
Type of partitions
Type of the experiment
Classification
Regression
Data set / Algorithms Use Case
User Manual

● k-fold cross validation 10
○ 5x2 cross validation
○ without validation

EXPERIMENTS DESIGN

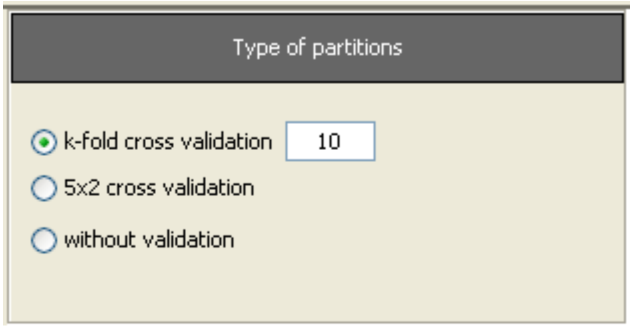
At this point you can design the desired experiments. First of all, you must select one of the following options:

1. k-fold cross validation
2. 5x2 cross validation
3. without validation

Select a type of partition and then the type of the experiment

First, it is necessary to select the type of experiment and the type of partitions to employ; the options selected will determine the kind of methods and data sets that will be available to design the experiment.

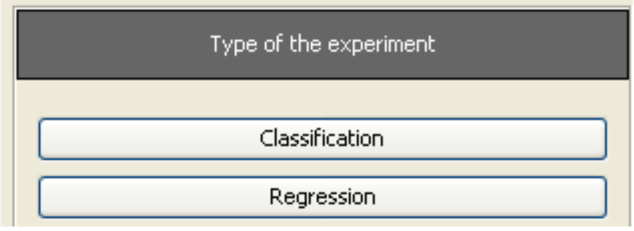
The types of partitions available are the following:



● k-fold cross validation 10
○ 5x2 cross validation
○ without validation

- k-fold cross validation (the value of k must be specified)
- 5x2 cross validation
- without validation

Currently, the KEEL Experiments module offers the following types of experiments:

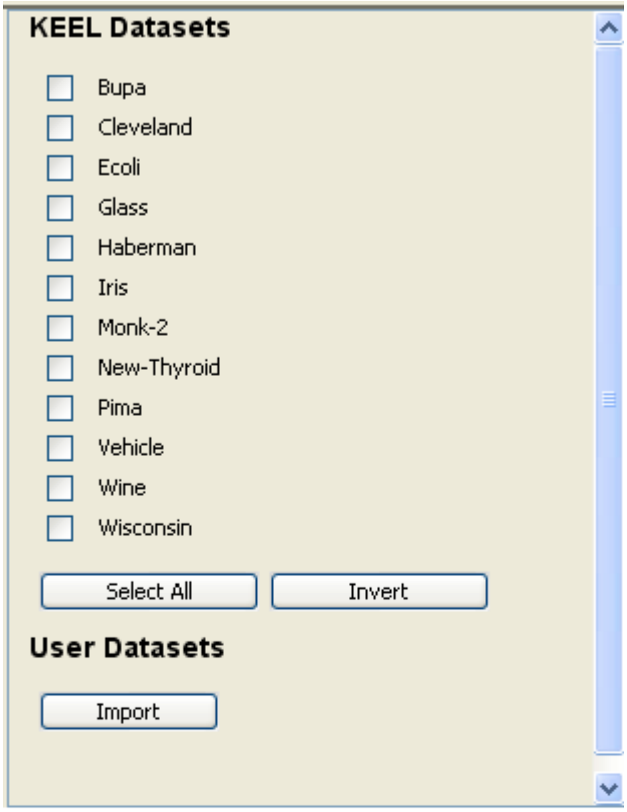


- Classification
- Regression

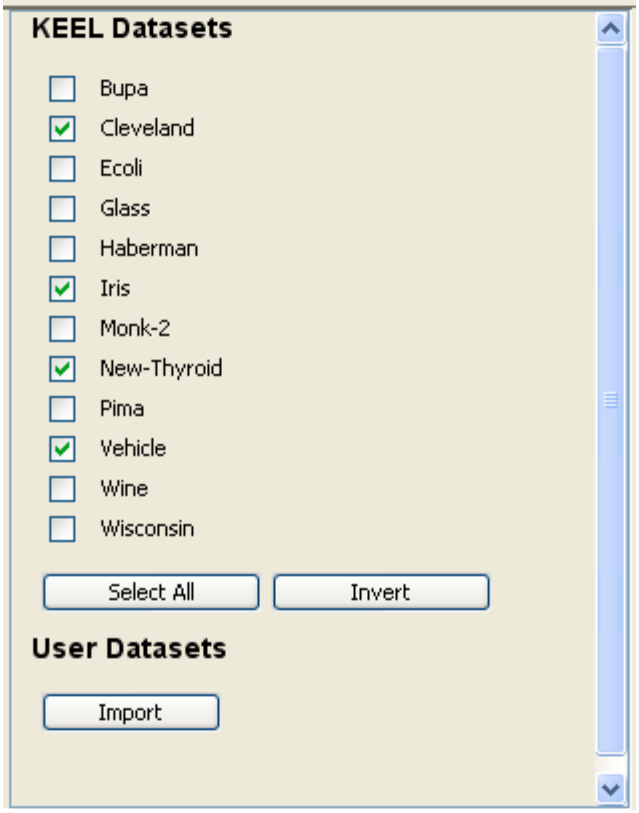
When the type of experiment has been selected, the data sets selection panel will be shown, allowing continuing the experiment design.

Selection of data sets

The data sets selection panel shows the available datasets for the current experiment. Its contents will depend of the type of experiment already selected:



The next step is to choose the wished data sets from the panel. The buttons *Select All* and *Invert* allows making the selection easily:



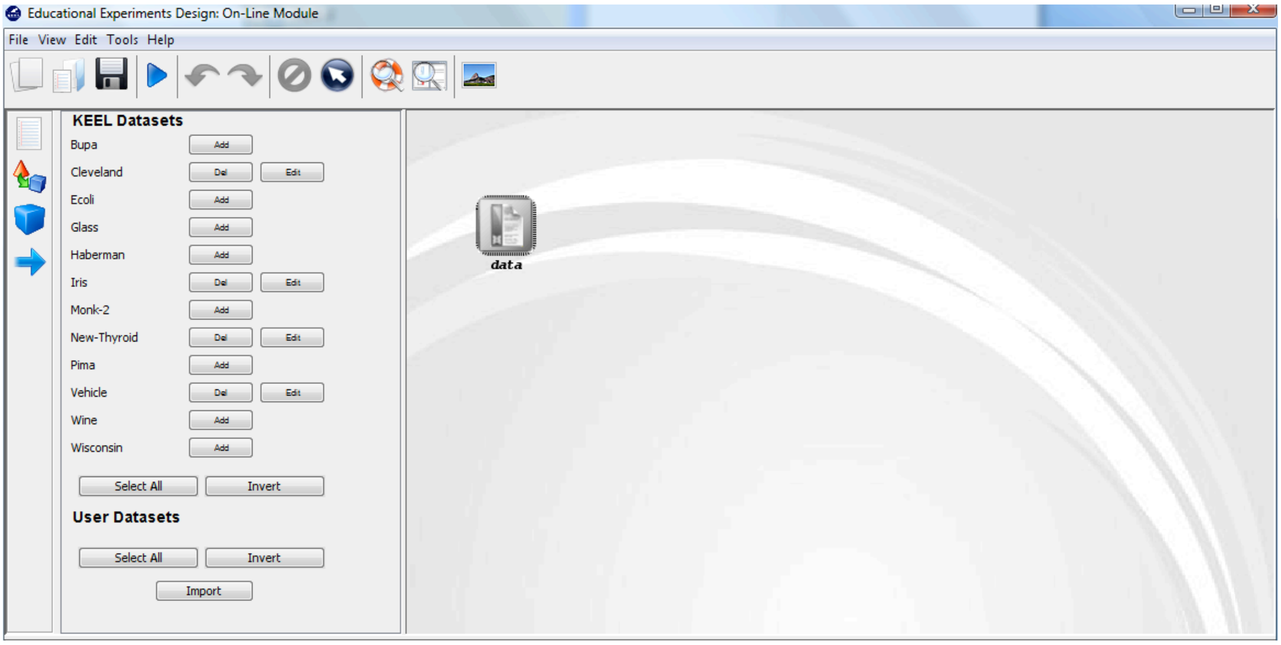
KEEL Datasets

- □ Bupa
- ☑ Cleveland
- □ Ecoli
- □ Glass
- □ Haberman
- ☑ Iris
- □ Monk-2
- ☑ New-Thyroid
- □ Pima
- ☑ Vehicle
- □ Wine
- □ Wisconsin

User Datasets

The Import Button allows importing an existing data set into the KEEL environment, ready to be selected for the current experiment. By clicking on it, the main window of the Data Import Tool will be shown. The process to import a new data set can is described in the Data Management module section of the manual.

When all the necessary data set are selected, the experiment design process can continue. To do so, the user must click on the white graph panel to set the data sets node of the experiment.



Educational Experiments Design: On-Line Module

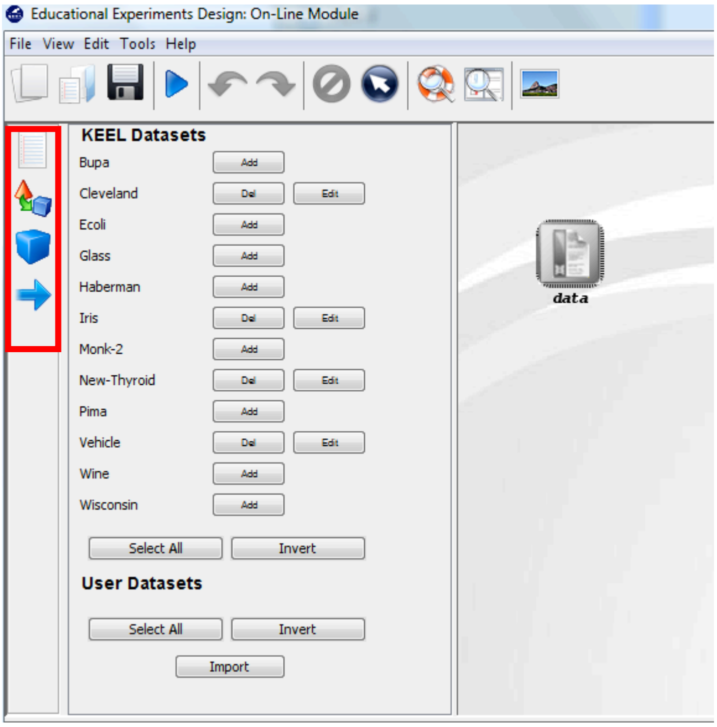
KEEL Datasets

Bupa	Add	
Cleveland	Del	Edit
Ecoli	Add	
Glass	Add	
Haberman	Add	
Iris	Del	Edit
Monk-2	Add	
New-Thyroid	Del	Edit
Pima	Add	
Vehicle	Del	Edit
Wine	Add	
Wisconsin	Add	

User Datasets

Experiment Graph

The experiment graph shows the components of the current experiment and describes the relationships between them. The user can add new components by using the left menu:


Educational Experiments Design: On-Line Module
File View Edit Tools Help
data

KEEL Datasets

Bupa	Add	
Cleveland	Del	Edit
Ecoli	Add	
Glass	Add	
Haberman	Add	
Iris	Del	Edit
Monk-2	Add	
New-Thyroid	Del	Edit
Pima	Add	
Vehicle	Del	Edit
Wine	Add	
Wisconsin	Add	

Select All | Invert

User Datasets

Select All | Invert

Import

This menu has the following categories available:

Data sets: Modify the data sets of the experiments.

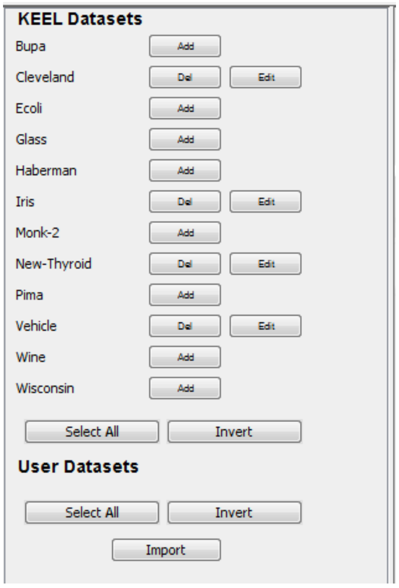
Preprocessing methods: Preprocess over the initial data sets.

Standard methods: Data mining methods.

Connections: Links between the components of the experiment.

Data sets

This module lets the user edit the current data sets selected for the experiment.

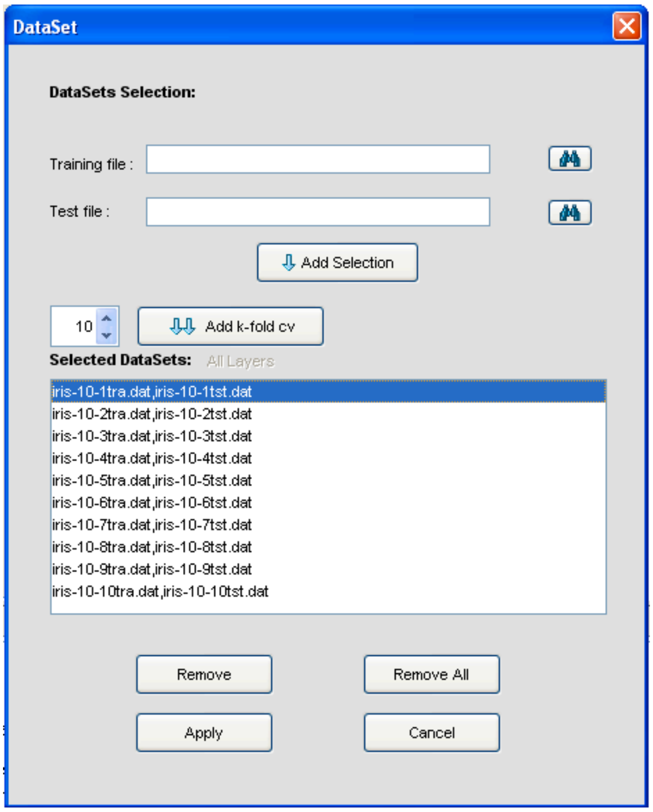


KEEL Datasets

Bupa	Add	
Cleveland	Del	Edit
Ecoli	Add	
Glass	Add	
Haberman	Add	
Iris	Del	Edit
Monk-2	Add	
New-Thyroid	Del	Edit
Pima	Add	
Vehicle	Del	Edit
Wine	Add	
Wisconsin	Add	

As in the *Select Datasets* panel, the user can still *Add* and *Delete* data sets to the experiment (from those already registered in the KEEL environment). Also, it is still possible to import new data sets.

Furthermore, the button *Edit* allows the user to indicate which partitions (training and test) desires to use. This way, it is possible to temporally alter the files which will be included in the experiment.

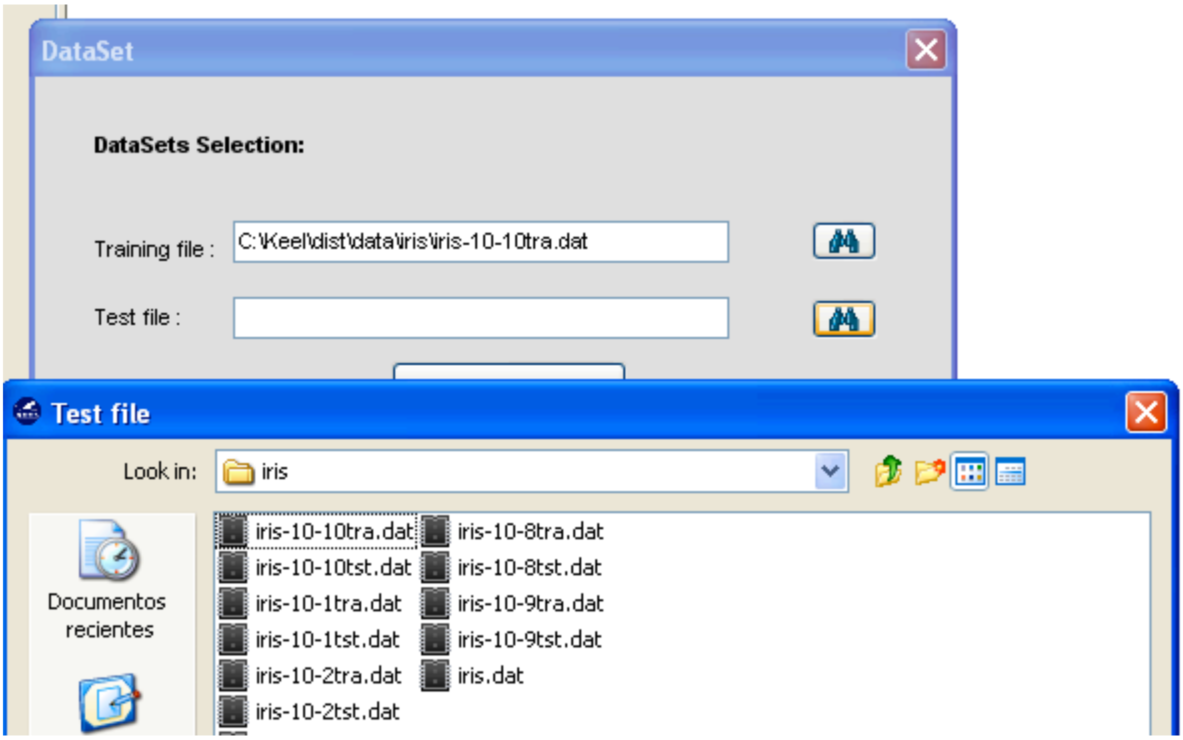


10

iris-10-1tra.dat,iris-10-1tst.dat
iris-10-2tra.dat,iris-10-2tst.dat
iris-10-3tra.dat,iris-10-3tst.dat
iris-10-4tra.dat,iris-10-4tst.dat
iris-10-5tra.dat,iris-10-5tst.dat
iris-10-6tra.dat,iris-10-6tst.dat
iris-10-7tra.dat,iris-10-7tst.dat
iris-10-8tra.dat,iris-10-8tst.dat
iris-10-9tra.dat,iris-10-9tst.dat
iris-10-10tra.dat,iris-10-10tst.dat

This dialog shows the initial files of the data set. From it, is possible to *Remove* a pair of training a pair of training/test files, to *Remove All* files.

Also, the dialog allows to *Add* a new pairs of training and test files. To do so, they must be selected by using the search buttons: 🔍



DataSets Selection:

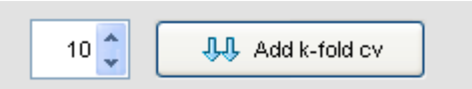
Training file : C:\Keel\dist\data\iris\iris-10-10tra.dat

Test file :

Look in: iris

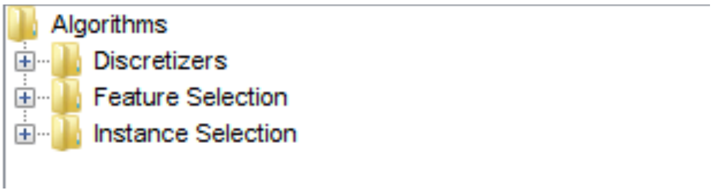
iris-10-10tra.dat
iris-10-10tst.dat
iris-10-1tra.dat
iris-10-1tst.dat
iris-10-2tra.dat
iris-10-2tst.dat
iris-10-8tra.dat
iris-10-8tst.dat
iris-10-9tra.dat
iris-10-9tst.dat
iris.dat

Finally, it is also possible to add a complete set of k-fold cross validation files by selecting the adequate number of folds and pressing the button *Add k-fold cv*.



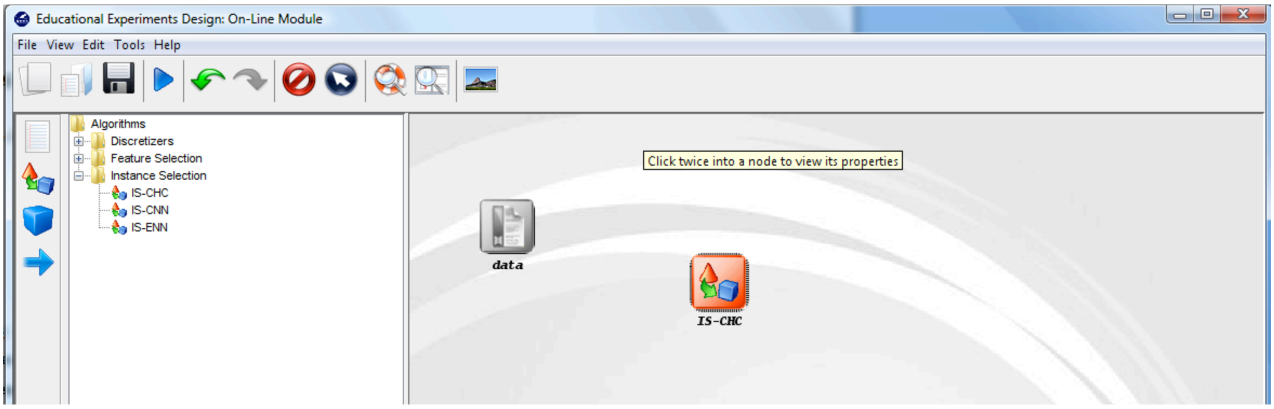
Preprocessing methods

This category includes several preprocessing methods


Algorithms
Discretizers
Feature Selection
Instance Selection

- **Discretizers:** Method to convert real or numeric data into nominal data.
- **Feature Selection:** Methods to select features of the data.
- **Instance Generation:** Methods to generate new instances from the original instances of the data set.

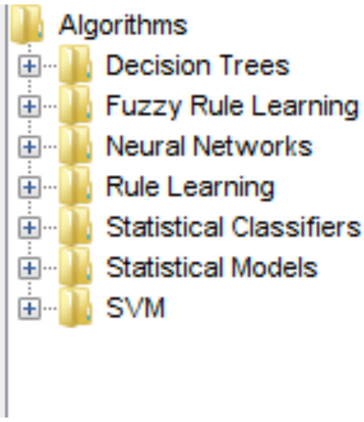
To add any preprocessing method to the current experiment, it is only needed to select it and click in the graph of the experiment:



Click twice into a node to view its properties

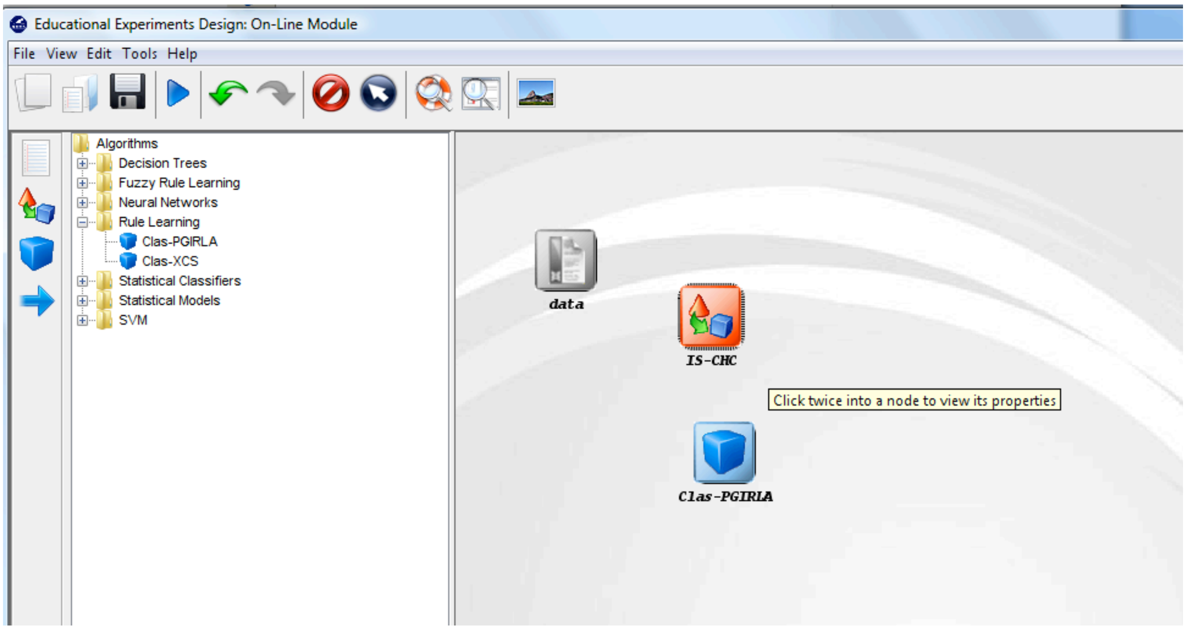
Standard methods

This category includes the data mining methods included in the KEEL tool:



- **Decision Trees**: Methods for building decision trees.
- **Fuzzy Rule Learning**: Methods for performing fuzzy rule-based learning.
- **Neural networks**: Artificial neural networks.
- **Rule Learning**: Methods for performing rule-based learning.
- **Statistical Classifiers**: Classifiers based on statistical models.
- **Statistical Models**: Construction of statistical models based on data.
- **SVM**: Support vector machines.

To add any method to the current experiment, it is only needed to select it and click in the graph of the experiment:

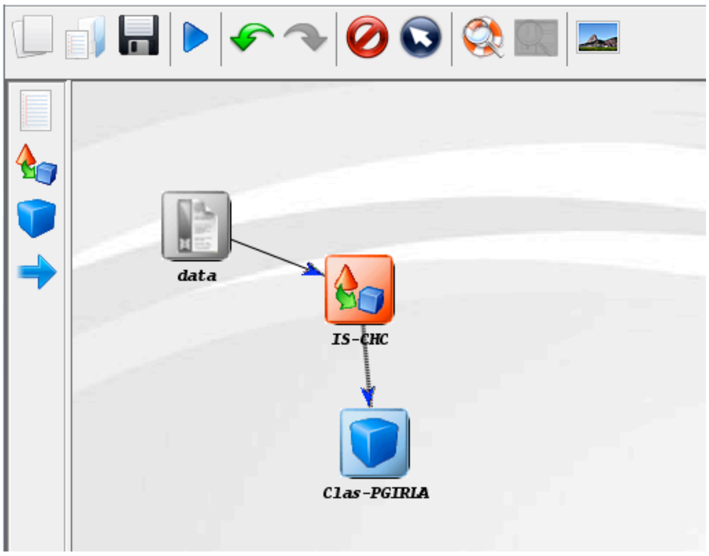


Click twice into a node to view its properties

Connections ➡

The connections allow finishing the designing of the experiment, by connecting the included modules with flows which represent the data flow in the experiment. They can be used both as inputs or outputs of the modules.

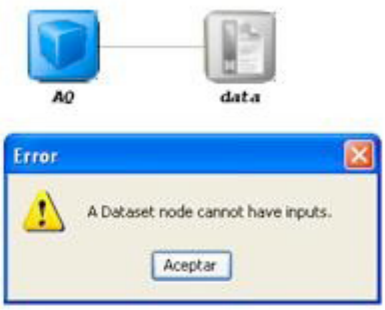
1. **Insert connection:** to make a connection, select the ➡ button from the left tool bar. Then, click on the source node and finally click on the target node.Ç



2. **Restrictions:** there are some restrictions that must be considered when making connections between the different elements:

    1. A dataset cannot have inputs.
    2. The pre-processing algorithms can only receive inputs from a data set or another pre-process method.
    3. Knowledge extraction methods can receive a flow from a data set, from a pre-processing algorithm or from a previous method.
    4. The graph cannot have any cycle.

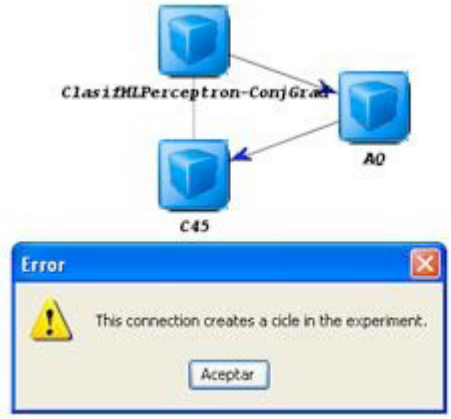
All these restrictions are verified in execution time when a connection is been created. If one of these connections is not permitted, the application will show an error message. In next figures you can see some examples of incorrect graphs.



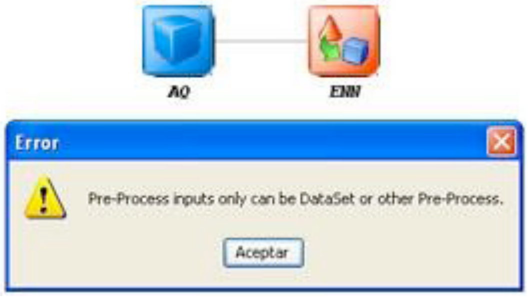
Error

A Dataset node cannot have inputs.

Aceptar



This connection creates a cicle in the experiment.

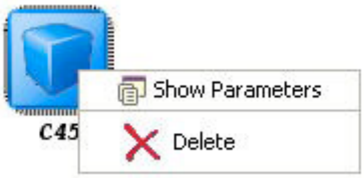


Pre-Process inputs only can be DataSet or other Pre-Process.

Graph Management

The graph allows performing the following operations over its elements:

1. Context menu: it is possible to access to the context menu by clicking with the right button of the mouse on a certain node in the draw area. This menu depends on the selected object and allows us to remove objects, to configure algorithms parameters, etc...



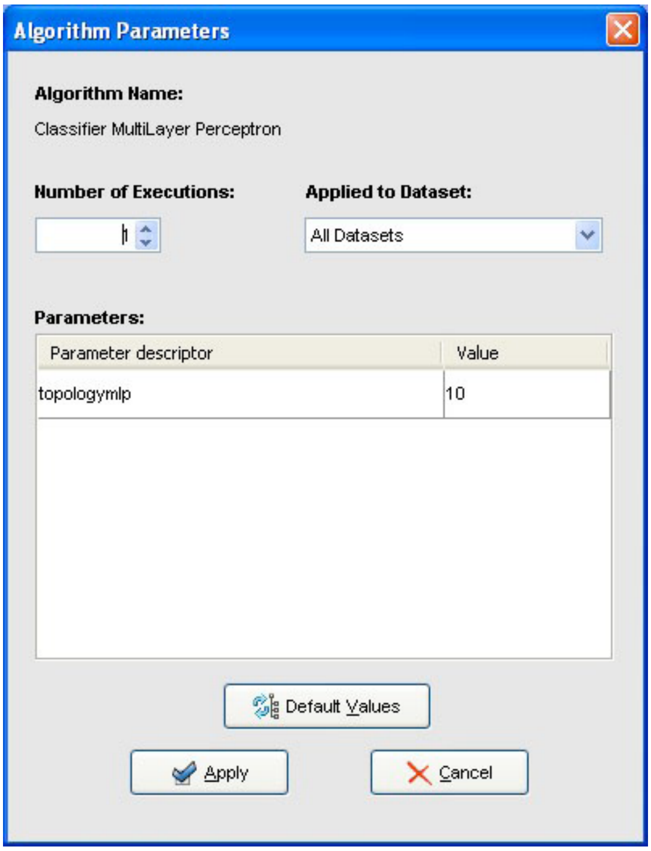
2. Objects selection: in order to select a single element, just click with the left button of the mouse over it. But it is possible to select several elements, clicking in an empty zone of the draw area and dragging the mouse until covering all the objects wished.

3. Move objects: It is possible to move one or several elements selected with the aid of the left button of the mouse, dragging them to the desired position. Another way is to use the keyboard cursors.

4. Remove objects: To remove a module, select it and press *Supr* key. It is also possible to remove it by the context menu, of from the tool bar.

Algorithm parameters configuration

Once a module has been inserted in the graph, it is possible to configure the value of its parameters. To do so, the user have to double click on the algorithm symbol and a dialog will be shown; also, this dialog can be shown through the emergent menu that will appear when right button of the mouse is pressed (option Show Parameters).



Algorithm Name:

Classifier MultiLayer Perceptron

Number of Executions: 1

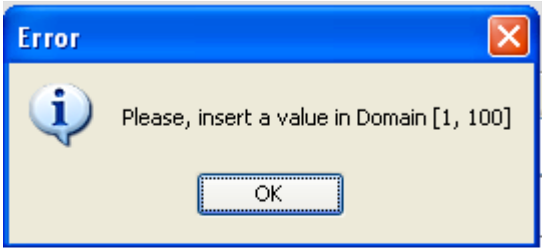
Applied to Dataset: All Datasets

Parameters:

Parameter descriptor	Value
topologymlp	10

At the top of this dialog it is possible to set the number of times that the algorithm will be executed (only available for random methods). Each execution will be made using a seed generated from the initial seed. The second list allows specifying in which data sets the parameters will be changed.

In the table located in the center of the window, all the algorithm parameters are established to its initial values. These values can be modified, as far as the new values will be appropriate for the concrete method; otherwise, an error message will appear:



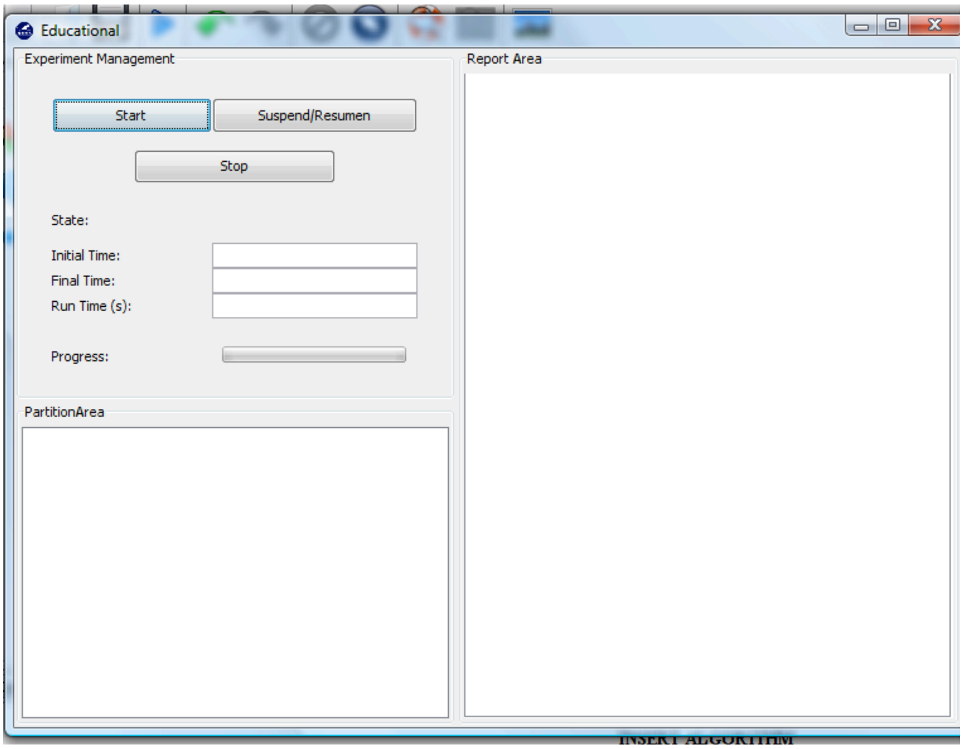
Please, insert a value in Domain [1, 100]

Finally, the *Default Values* button allows returning all parameter to its default values.

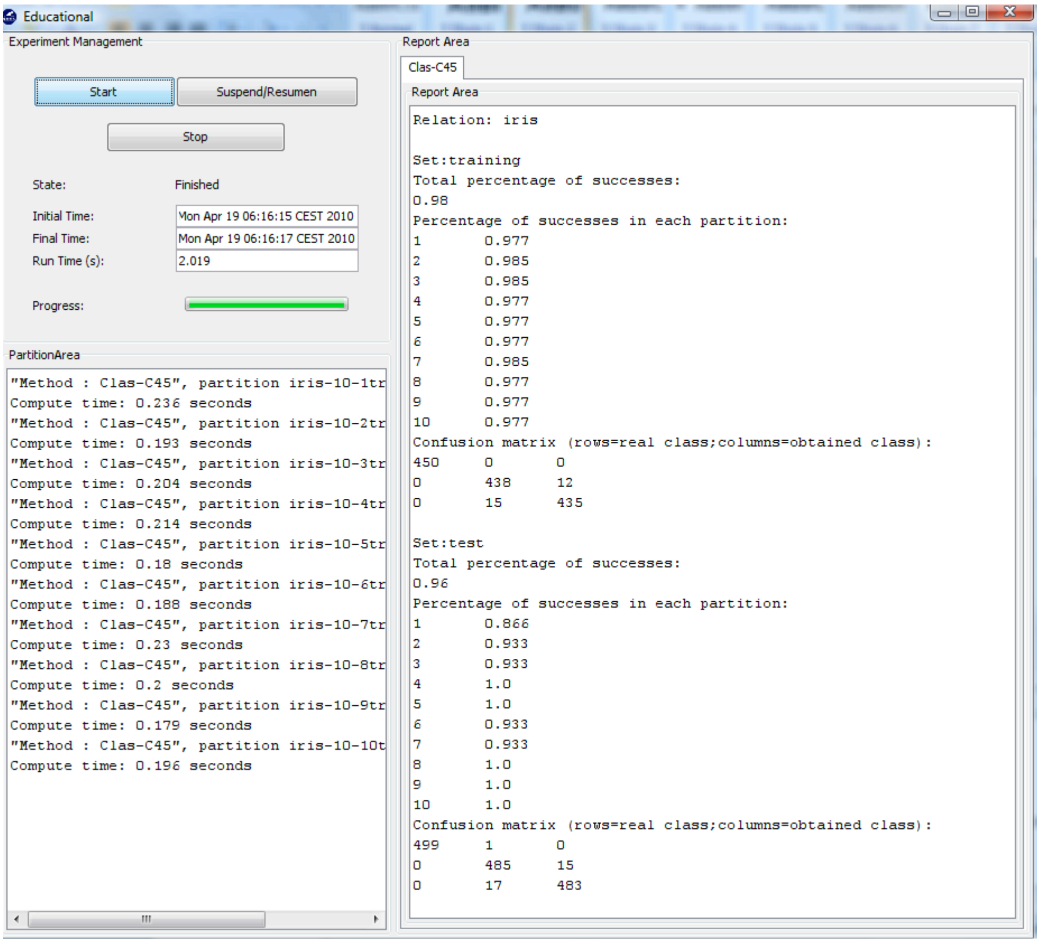
Run experiment

Once an experiment has been designed, the user can generate it through the option Run Experiment of the 'Tools' menu. Furthermore, it is possible to use the tools bar button. ▶

At this point, new window allows us to run the experiment.



If start button is pressed the experiment will be run and the results will be shown in the "report area" section. In this section, the information reported is the total percentage of successes considering all partitions, the percentage of successes in each partition and the confusion matrix. This information is indicated both for training and test data set. Moreover, in the section of "partition area", the computation time for each partition is specified. An example the execution is shown in the following figure.



State: Finished

Initial Time: Mon Apr 19 06:16:15 CEST 2010

Final Time: Mon Apr 19 06:16:17 CEST 2010

Run Time (s): 2.019

```
"Method : Clas-C45", partition iris-10-1tr
Compute time: 0.236 seconds
"Method : Clas-C45", partition iris-10-2tr
Compute time: 0.193 seconds
"Method : Clas-C45", partition iris-10-3tr
Compute time: 0.204 seconds
"Method : Clas-C45", partition iris-10-4tr
Compute time: 0.214 seconds
"Method : Clas-C45", partition iris-10-5tr
Compute time: 0.18 seconds
"Method : Clas-C45", partition iris-10-6tr
Compute time: 0.188 seconds
"Method : Clas-C45", partition iris-10-7tr
Compute time: 0.23 seconds
"Method : Clas-C45", partition iris-10-8tr
Compute time: 0.2 seconds
"Method : Clas-C45", partition iris-10-9tr
Compute time: 0.179 seconds
"Method : Clas-C45", partition iris-10-10t
Compute time: 0.196 seconds
```

```
Relation: iris

Set:training
Total percentage of successes:
0.98
Percentage of successes in each partition:
1       0.977
2       0.985
3       0.985
4       0.977
5       0.977
6       0.977
7       0.985
8       0.977
9       0.977
10      0.977
Confusion matrix (rows=real class;columns=obtained class):
450     0       0
0       438     12
0       15      435

Set:test
Total percentage of successes:
0.96
Percentage of successes in each partition:
1       0.866
2       0.933
3       0.933
4       1.0
5       1.0
6       0.933
7       0.933
8       1.0
9       1.0
10      1.0
Confusion matrix (rows=real class;columns=obtained class):
499     1       0
0       485     15
0       17      483
```

With the suspend/resume button, the experiment can be suspended and resumed in any moment. Finally, the stop button allows us to stop the experiment and see the results.