

# Module IV.1

## Observation and Evaluation Techniques from Intelligent Resources: Introduction to Data Mining



Co-funded by  
the European Union



**Dr. Álvaro Arnaiz González**  
**Dr. José Francisco Díez Pastor**  
**Dr. Sandra Rodríguez Arribas**

“This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.”



## Observation and Evaluation Techniques from Intelligent Resources: Introduction to Data Mining

1. Data Mining
2. Types of Learning in Data Mining
3. Classification algorithms
4. Clustering algorithms
5. Regression algorithms
6. Knime
7. Additional Material: Using Knime

### 1. DATA MINING

**Data Mining** is the process of searching and analyzing large databases to find useful information that is useful for decision making.

There are numerous **DM** techniques that employ mathematical analysis to deduce the patterns and trends that exist in the data. Typically, these patterns cannot be detected by traditional data exploration because the relationships are too complex or because the volume of data to be analysed is too large.

Currently in the field of **Data Mining** it is continuously used for the analysis of large amounts of data in various fields of knowledge such as education, economics, business, the environment...



## 1.1 BASIC CONCEPTS IN DATA MINING

### Data set

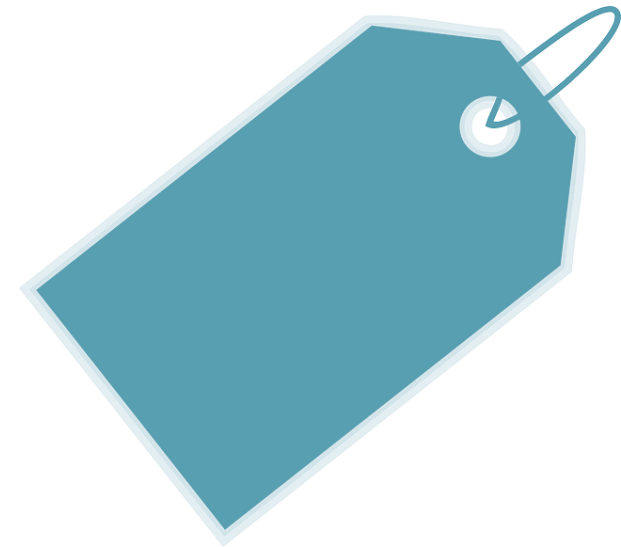
It is a large collection of data usually organized into rows and columns containing variables and attributes. Each of these values is known by the data name. The dataset can also consist of a collection of



## 1.1 BASIC CONCEPTS IN DATA MINING

### Classes or tags

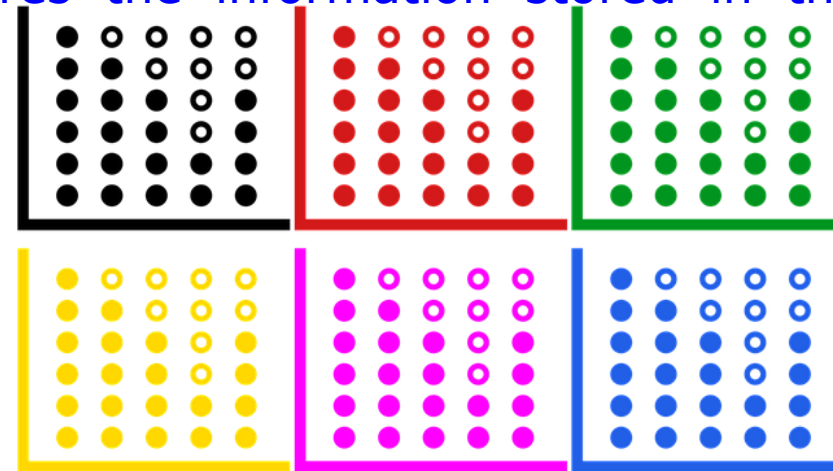
In the field of **Data Mining**, a class is the discrete attribute whose value you want to predict based on the values of other attributes. It is also known as a label.



## 1.1 BASIC CONCEPTS IN DATA MINING

### Instance

An instance is each of the data that is available for analysis. Each instance, in turn, is composed of features that describe it. For example, in a spreadsheet, the instances would be the rows and the features the information stored in the columns.



### 1.1 BASIC CONCEPTS IN DATA MINING

#### Algorithm

In computer science, an algorithm is a set of defined, ordered, and bounded instructions to solve a problem, perform a calculation or develop a task. In other words, it is a step-by-step procedure to get a result.



## 1.2 PROCESS OF APPLICATION OF DATA MINING TECHNIQUES

Problem definition

Data preparation and collection

Modeling and evaluation

Deployment



## 1.2 PROCESS OF APPLICATION OF DATA MINING TECHNIQUES

### Problem definition

- This is the first phase in which a specific problem is translated into a **data mining** problem in which the objectives of the analysis and research questions are raised

## 1.2 PROCESS OF APPLICATION OF DATA MINING TECHNIQUES

### Data preparation and collection

- It is the most extensive phase of the process since data quality is one of the most important challenges in **data mining**. Raw data must be identified, cleaned, and stored in a preset format

## 1.2 PROCESS OF APPLICATION OF DATA MINING TECHNIQUES

### Modeling and evaluation

- In this step, different data modeling techniques (algorithms) are selected and applied and then the optimal parameters and values of these techniques are established

## 1.2 PROCESS OF APPLICATION OF DATA MINING TECHNIQUES

### Deployment

- This is the last phase in which the results of **data mining** are organized and presented using graphs and reports

## 1.2 PROCESS OF APPLICATION OF DATA MINING TECHNIQUES

It is important to note that every **data mining** process is an iterative process, which means that the process does not stop when a particular solution is deployed. It may be just a new entry for another **data mining** process (Rodríguez-Arribas, 2021). That is, on many occasions the application of **DM** techniques requires several iterations and the use of different algorithms to be able to extract the final results of the research we are doing.



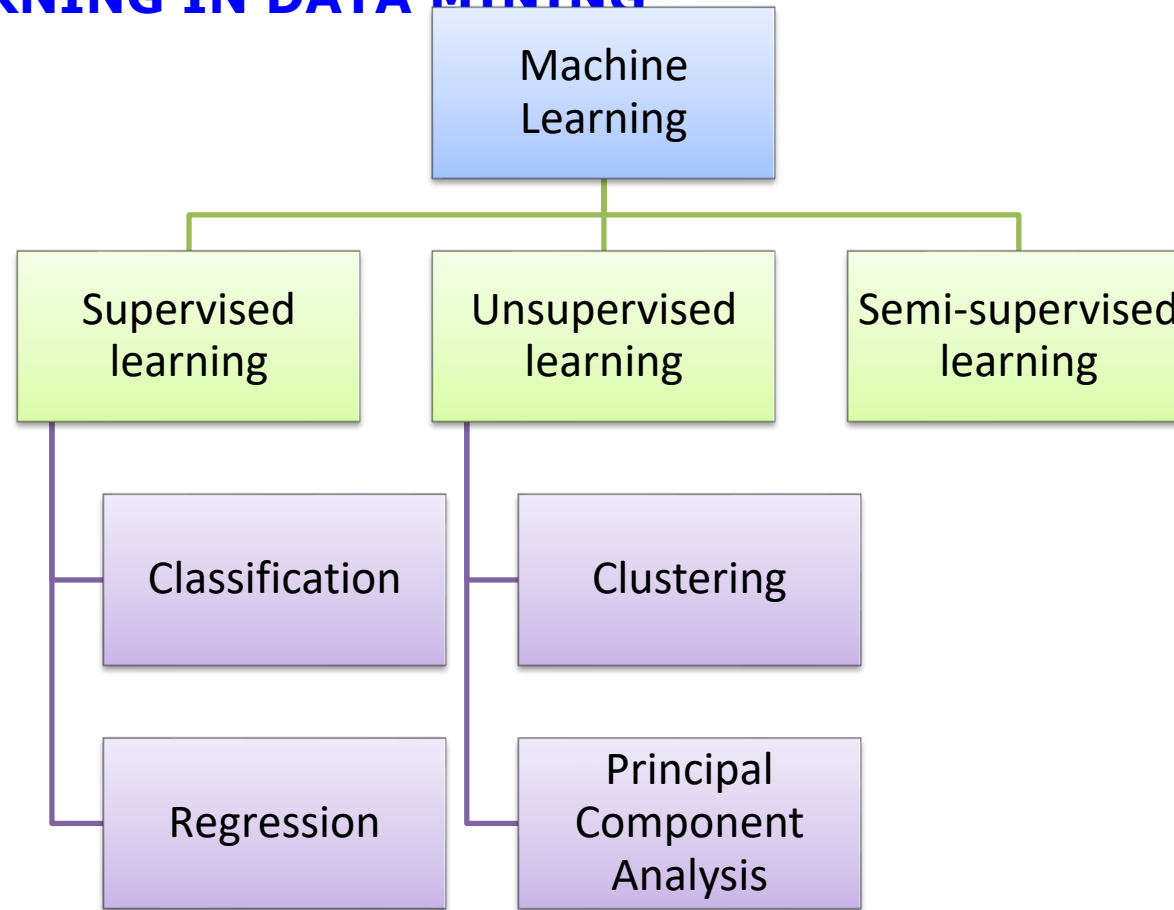
## 2. TYPES OF LEARNING IN DATA MINING

There are numerous classifications of the algorithms used in the world of **Data Mining**, but it is essential to understand that there are two basic approaches: supervised learning and unsupervised learning.

When we must decide which algorithm will be used to perform data analysis, it is important to take into account what type of learning is being used, that is, if we are talking about supervised or unsupervised learning (García, Luengo and Herrera, 2015). According to the type of learning used, different techniques and algorithms will be used as can be seen in the mind map.



## 2. TYPES OF LEARNING IN DATA MINING



## 2. 1 SUPERVISED LEARNING

The fundamental objective of supervised learning is the creation of a model that is able to predict values corresponding to input objects after having become familiar with a series of examples, training data.

This technique consists of **two fundamental steps**:

1. A training phase where a set of labeled data is used, which contain the input data and the desired results for that training data with an algorithm that allows to deduce a function from the data that we are providing to the algorithm

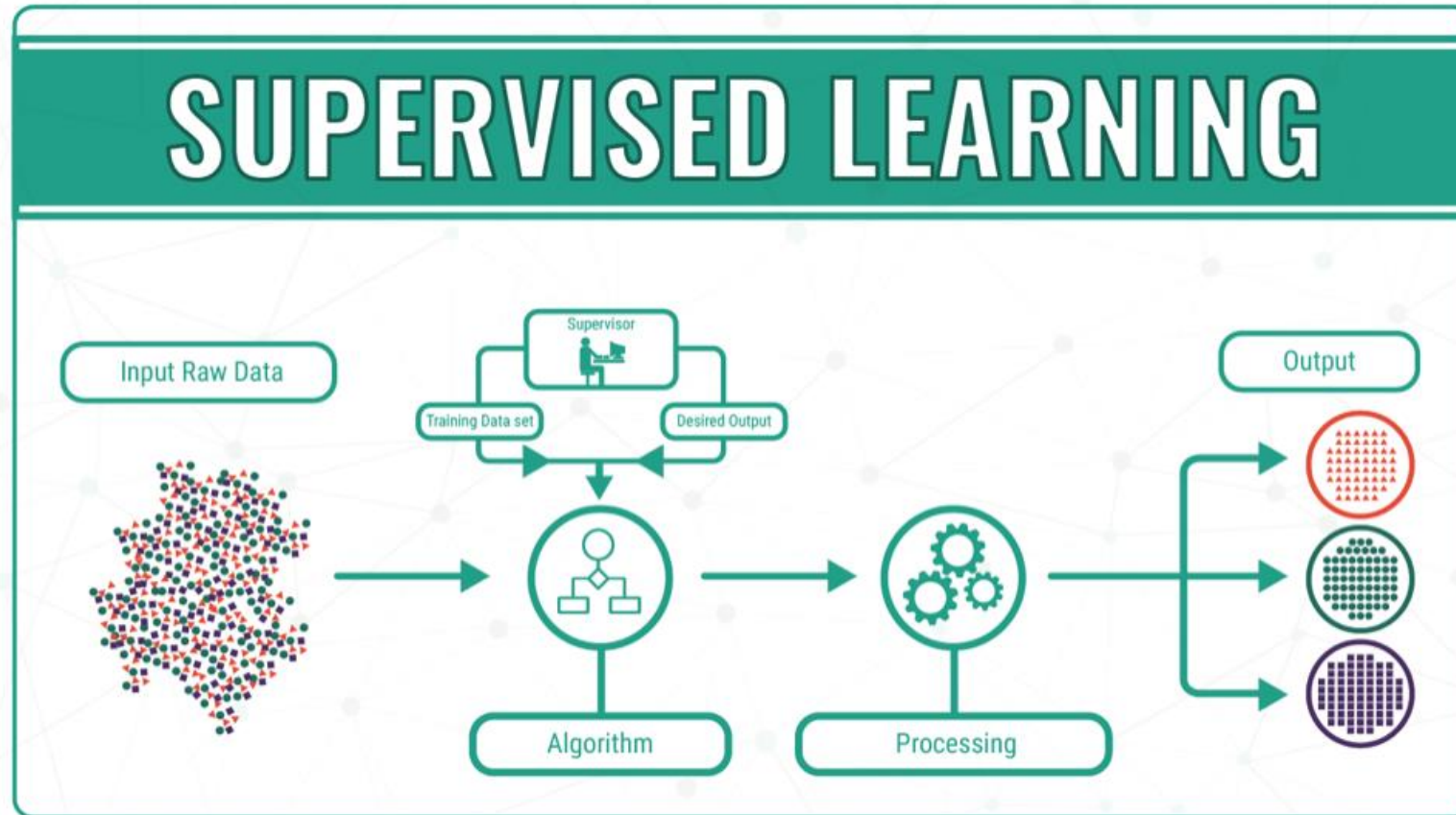
2. The test phase, where the function obtained in the previous step is used to

generate new predictions with new data sets.





## 2. 1 SUPERVISED LEARNING



### 2. 1 SUPERVISED LEARNING

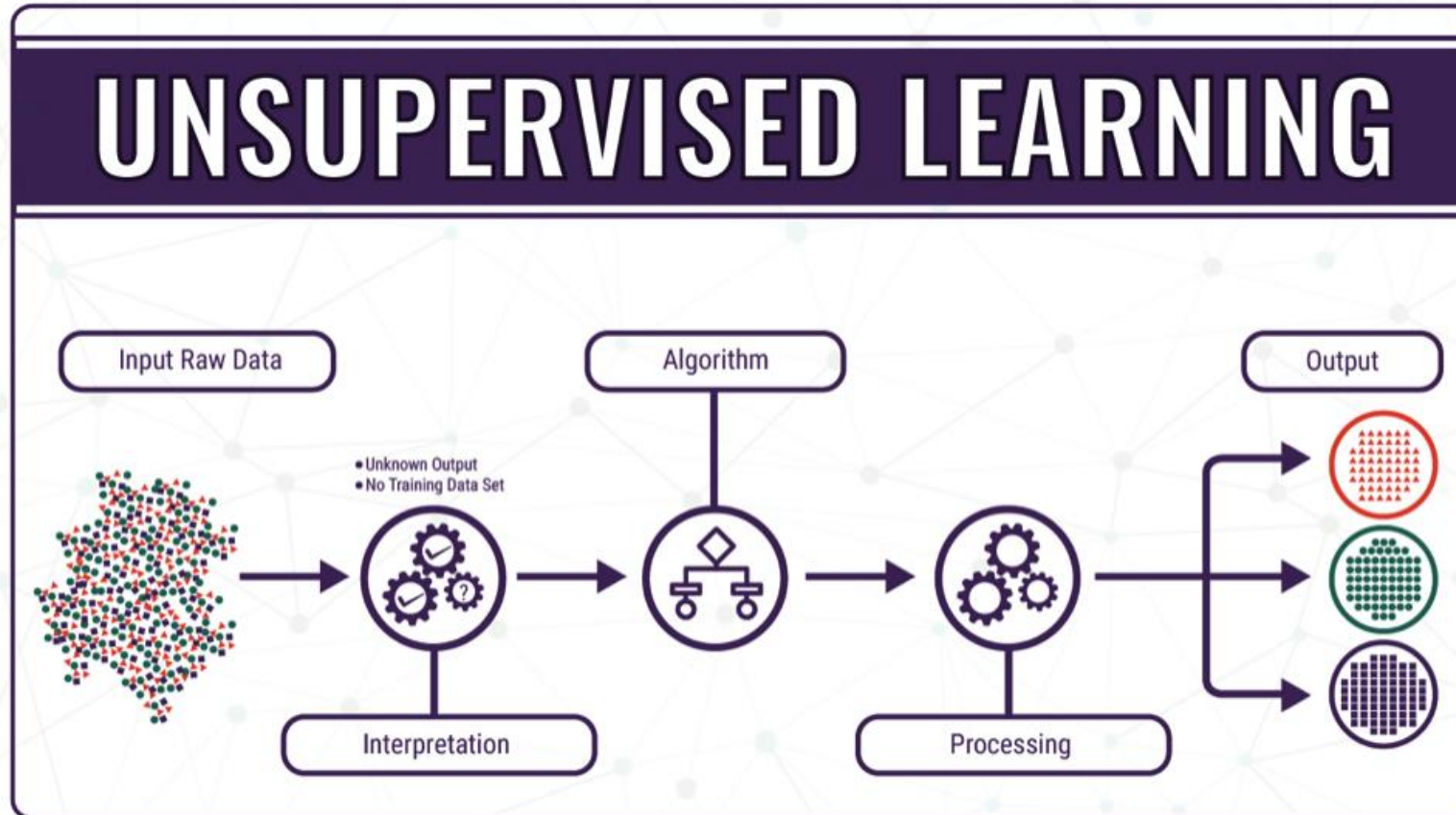
The process is known as supervised learning, since by knowing the responses of each example of the training set, it is possible to correct the function generated by the algorithm. The training of the algorithm is supervised by correcting its parameters, depending on the results obtained iteratively

### 2. 2 UNSUPERVISED LEARNING

This type of learning is the other basic approach to Machine Learning (ML). Unsupervised learning has unlabeled data that the algorithm must try to understand for itself.

The goal of this type of learning is to let the machine learn without help or directions from data scientists, that is, without supervision and without a training dataset. In addition, the machine itself will adjust the results and groupings when there are more suitable results, allowing the machine to understand the data and process it in the best way

## 2. 2 UNSUPERVISED LEARNING



## 2. 2 UNSUPERVISED LEARNING

Unsupervised learning is used to explore unknown and unlabeled data. It can reveal patterns that might have been overlooked or examine large data sets that would be too much for a single person to address.



### 2. 3 SEMI-SUPERVISED LEARNING

Numerous investigations are currently being conducted with semi-supervised learning methods. These Machine Learning techniques use both labeled and unlabeled training data: typically, a small amount of labeled data alongside a large amount of unlabeled data (Zhu and Goldberg, 2009). That is, they seek to improve the prediction models that are obtained by using exclusively labeled data by exploring the structural information contained in the unlabeled data.

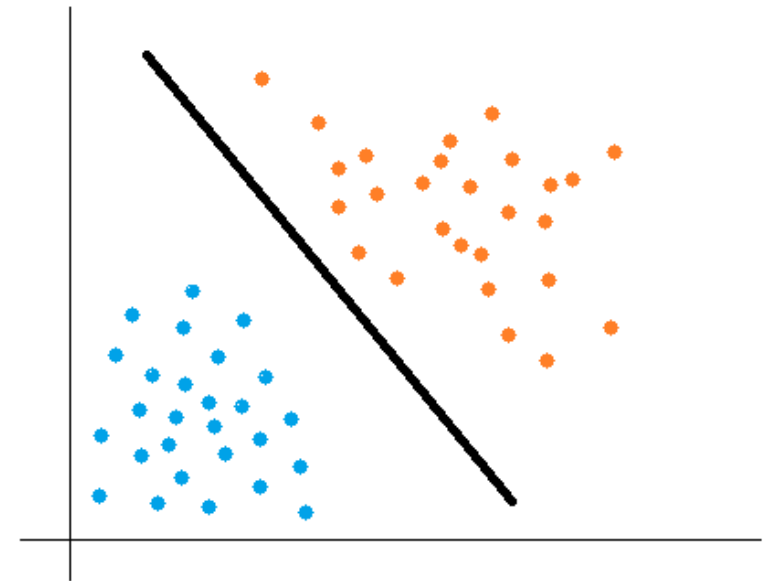
We can say semi-supervised learning tries to combine the two traditional approaches of data mining (supervised learning and unsupervised learning) to keep the best of each of them.



### 3. CLASSIFICATION ALGORITHMS

Classification algorithms are those we use when the expected result is a discrete label. That is, they are useful when the answer to the research question lies within a finite set of possible outcomes.

Classification is very similar to the learning process of people, since we possess the ability to classify food, books, animals, planets, that is, everything around us.



### 3. CLASSIFICATION ALGORITHMS

These algorithms generally work on the information delivered by a set of samples, patterns, examples, or training prototypes that are taken as representatives of the classes, and they retain a correct class label. This set of correctly labeled prototypes is called a training set, and it is the knowledge available for the classification of new samples. The objective of supervised classification is to determine, according to what is known, which class a new sample should concern, considering the information that can be extracted

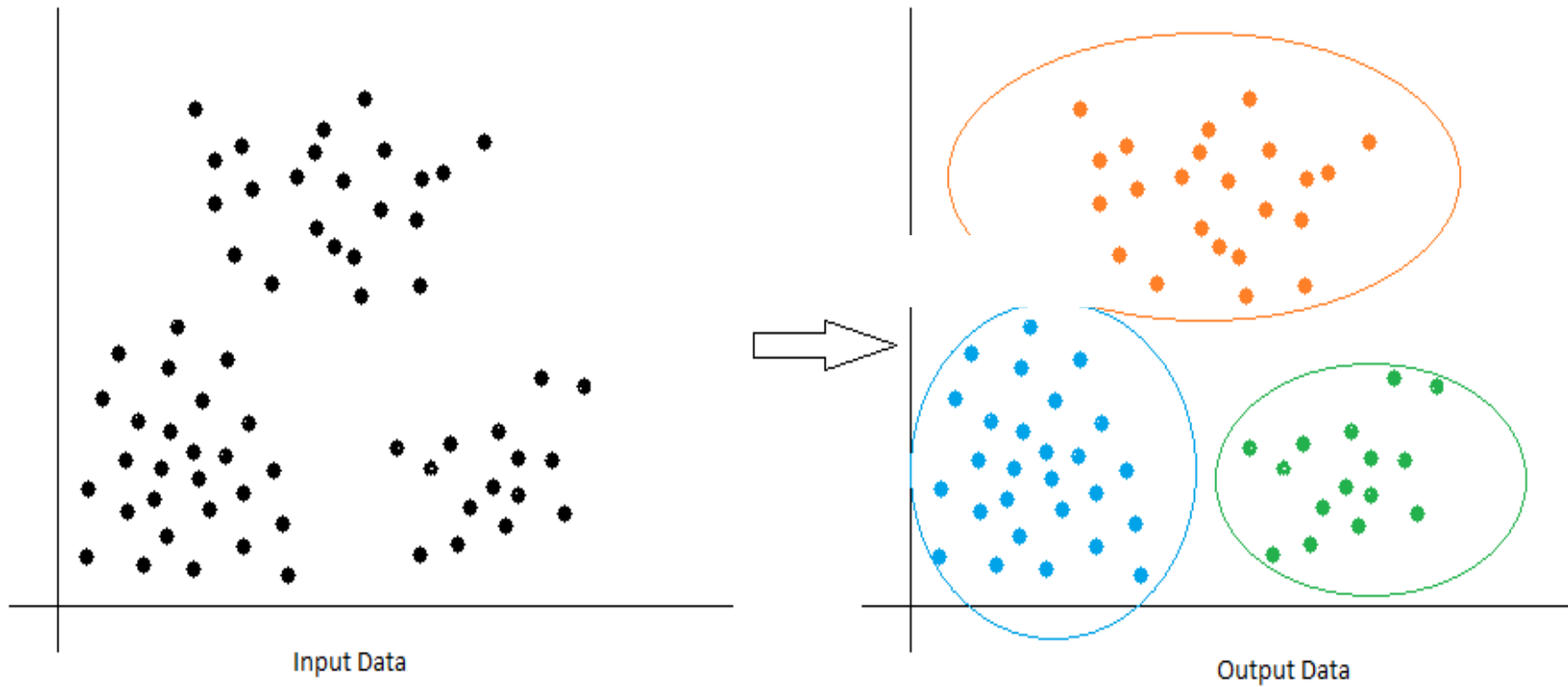


## 4. CLUSTERING ALGORITHMS

Clustering algorithms are responsible for grouping the objects in a dataset according to their similarities. In this way the objects that are within a cluster or group have more similarities between them than differences.

These algorithms work with unlabeled data, so it is the algorithm itself that analyzes the data to find the optimal number of groupings for the input data set since we do not have prior knowledge about the characteristics of the data and its classes.

## 4. CLUSTERING ALGORITHMS

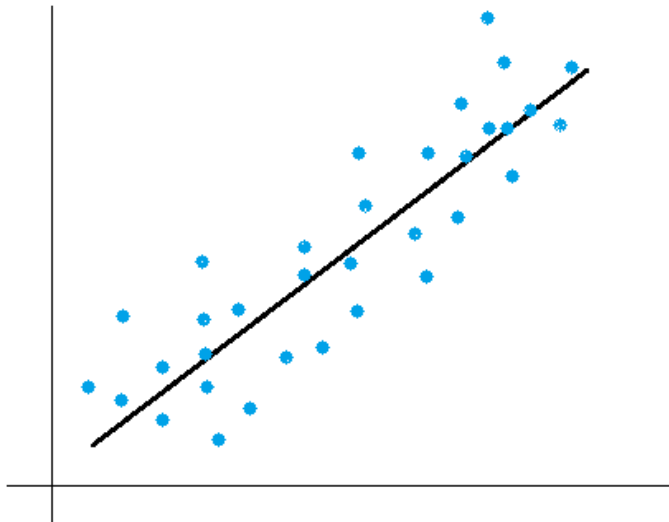


## 4. CLUSTERING ALGORITHMS

The groupings performed by the algorithms can be of two types:

- 1. Hard cluster:** each piece of data belongs exclusively to a group.
- 2. Soft (diffuse) cluster:** the data can belong to several groups in different degrees, that is, the same data can have a degree of belonging of 60% to group 1 and 40% in group 2.

## 5. REGRESSION ALGORITHMS



Regression algorithms are a subfield of supervised learning whose goal is to establish a method for the relationship between a certain number of characteristics and a continuous target variable.

These are algorithms that establish a line to provide the trend of a set of data, that is, the purpose of these algorithms is to relate a number of characteristics and a continuous objective variable.

## 5. REGRESSION ALGORITHMS

This technique is useful for predicting outcomes that are continuous values, that means that the answer to the research question is presented by a quantity that can be flexibly determined based on model inputs rather than being limited to a finite set of labels as in the case of classification

## 6. KNIME

### 6. 1 INTRODUCTION

KNIME is an open-source application that allows us to apply to our own datasets or to sample datasets:

1. Statistical methods
2. Data mining algorithms or Machine Learning.
3. Visualization techniques.

It is built on the Eclipse platform and is programmed in Java. Being an open-source software has many advantages, its code belongs to the community of users and developers, which guarantees that it will always be a free tool that can be downloaded and used free of charge under the terms of the GPLv3 licence. It also allows the incorporation of code developed in R or Python.



### 6. KNIME

It is a "Visual Programming" tool. Data analysis can be done intuitively by setting up the process simply by clicking the mouse. The "nodes" that we need are placed, without the need to know their name or how they are configured, since at all times we have help.

It is a tool designed to be simple to use. The most important concept in the use of the tool is that of workflow (in Spanish, workflow). A **workflow** is a sequence of steps configured by the user. Formally it is a set of nodes joined together with arrows that represent the flow of data from one node to another. A node encapsulates different jobs that can be done with the data, there are nodes for many tasks.



### 6. KNIME

There are nodes for:

- a. Load data from files or databases.
- b. Create, modify, or delete rows or columns from the dataset we are working with.
- c. Calculate statistics means, percentiles, correlations etc.
- d. Combine data from different data sources.
- e. Build and evaluate Machine Learning models such as: classification, regression, or clustering.
- f. Visualize the data using bar charts, pie charts, scatter charts, and also other more advanced chart types.
- g. Generation of reports.



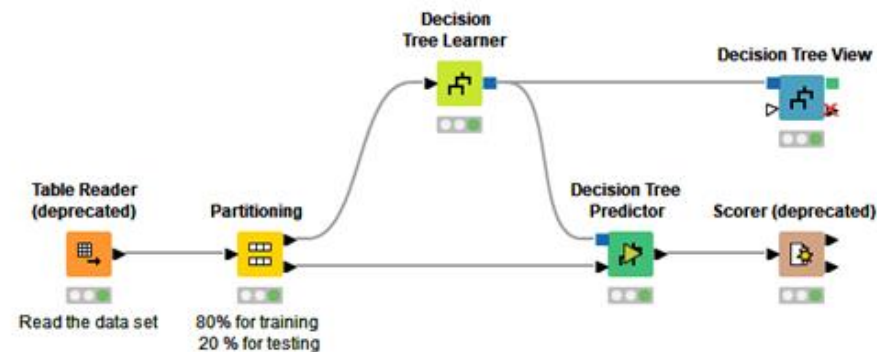


## 6. KNIME

A workflow might have a node to load a dataset from an Excel file, then a node to select attributes (columns) from that dataset, and then another node to display statistics for the selected attributes.

### Decision Tree: Training

This workflow is an example of how to build a basic prediction / classification model using a decision tree.  
Dataset describes wine chemical features. Output class is wine color: red/white



### Bibliography

- García, S., Luengo, J., y Herrera, F. (2015). Data Preprocessing in Data Mining / by Salvador García, Julián Luengo, Francisco Herrera. Springer
- Sáiz-Manzanares, M.C., Marticorena, R., y Arnaiz-Gonzalez, Á. (2022). Improvements for therapeutic intervention from the use of web applications and machine learning techniques in different affectations in children aged 0-6 years. *Int. J. Environ. Res. Public Health*, 19, 6558. <https://doi.org/10.3390/ijerph19116558>
- Sáiz-Manzanares, M.C., Marticorena, R., & Arnaiz, Á. (2020). Evaluation of Functional Abilities in 0–6-Year-Olds: An Analysis with the eEarlyCare Computer Application. (2020). *Int. J. Environ. Res. Public Health*, 17(9), 3315, 1-17 <https://doi.org/10.3390/ijerph17093315>
- Sáiz-Manzanares, M.C., Marticorena, R., Arnaiz-González, Á., Díez-Pastor, J.F., & Rodríguez-Arribas, S. (2019, March). Computer application for the registration and automation of the correction of a functional skills detection scale in Early Care. 13th International Technology, Education and Development Conference Proceedings of INTED2019 Conference 11th-13th (5322-5328). IATED: Valencia. doi: 10.21125/inted.2019.1320

### Images

Image 1 <https://pixabay.com/es/illustrations/grande-datos-teclado-computadora-895567/>

Image 2 <https://pixabay.com/es/vectors/etiqueta-equipaje-blanco-precio-309129/>

Image 3 <https://pixabay.com/es/illustrations/es-el-gr%c3%a1fico-tabla-varilla-5474235/>

Image 4 <https://pixabay.com/es/vectors/aprendizaje-autom%c3%a1tico-7271039/>

Image 5 <https://pixabay.com/es/photos/pir%c3%a1mide-gr%c3%a1fico-colores-infografia-2611048/>

Image 6 <https://chisoftware.medium.com/supervised-vs-unsupervised-machine-learning-7f26118d5ee6>

Image 7 <https://chisoftware.medium.com/supervised-vs-unsupervised-machine-learning-7f26118d5ee6>

Image 8 own elaboration

Image 9 own elaboration

Image 10 own elaboration

Image 11 own elaboration

# Module IV.1

## 7. Additional Material: Using KNIME



Co-funded by  
the European Union



## Additional Material: Using KNIME

1. KNIME INSTALLATION
2. KNIME WORKFLOWS.
  1. Nodes
  2. The workspace.
3. Generic example: Classifying flower species.
4. Example with data from intelligent therapeutic intervention (EarlyCare)

### 1. KNIME INSTALLATION

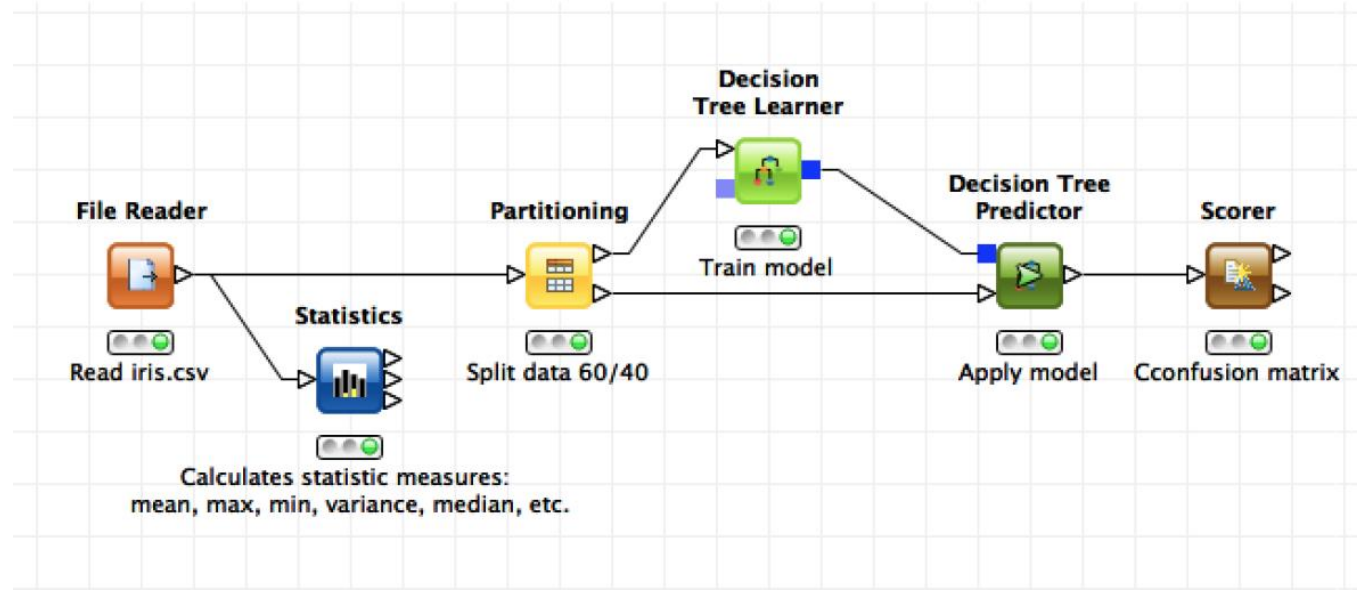
KNIME is a Java application, which means that you will need to have the Java virtual machine installed before you can install and run the program.

To install the software, we must go to <https://www.knime.com/downloads> , once there we will download "KNIME Analytics Platform" choosing the corresponding version for the personal computer we have: Mac, Windows 32 bits (old computers), Windows 64 (modern computers) or Linux.



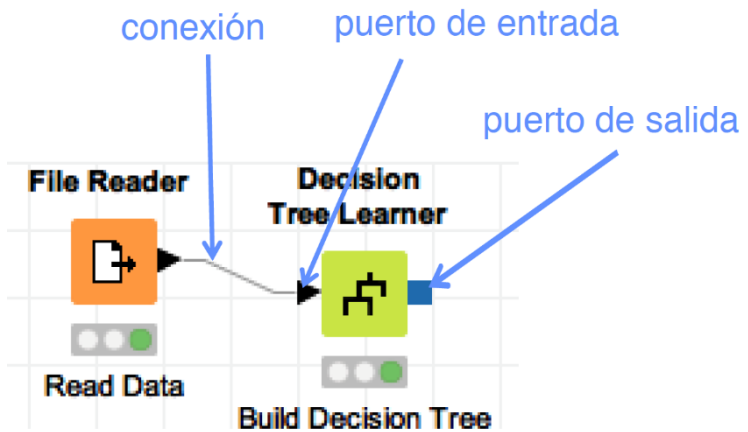
### 2. KNIME WORKFLOWS

They are a visual representation of the sequence of steps that take place in the data analysis process. They are composed of a series of linked nodes.



### 2.1 NODES

Nodes encapsulate the algorithms that implement the actions that can be performed on the data:



- Manipulation of rows, columns, etc.
- Creation of data mining models.
- Evaluation of models.
- Application of models on new data.
- ETL processes (Extract, Load, Transform).
- Creation of customised reports.



## Additional Material: Using KNIME

### 2.2 THE WORKSPACE

The screenshot displays the KNIME Analytics Platform workspace. The main area shows a workflow with three nodes: **Partitioning** (Random drawing 80% upper port 20% lower port), **Decision Tree Learner** (Train to predict class "income"), and **Decision Tree Predictor** (Apply decision tree model to test set). The **Decision Tree Learner** node is selected, and its description is shown in the right-hand pane. The description states: "This node induces a classification decision tree in main memory. The target attribute must be nominal. The other attributes used for decision making can be either nominal or numerical. Numeric splits are always binary (two outcomes), dividing the domain in two partitions at a given split point. Nominal splits can be either binary (two outcomes) or they can have as many outcomes as nominal values. In the case of a binary split the nominal values are divided into two subsets. The algorithm provides two quality measures for split calculation; the gini index and the gain ratio. Further, there exist a post pruning method to reduce the tree size and increase prediction accuracy. The pruning method is".

The **Recommended Nodes** pane on the left lists various nodes with their community usage percentages:

Node	Community
Decision Tree Predictor	85%
Decision Tree To Image	5%
Decision Tree to Ruleset	3%
PMML Writer	3%
Decision Tree View	1%
PMML To Cell	<1%
Boosting Learner Loop End	<1%
Model Writer	<1%
Model Loop End	<1%

The **Console** window at the bottom right shows the following output:

```
*** Welcome to KNIME Analytics Platform v4.0.1.v201908131317 ***
*** Copyright by KNIME AG, Zurich, Switzerland ***
Log file is located at: /Users/cgosorio/knime-workspace/.metadata/knime/kni
WARN Color Manager 0:2 Column "income" has no nominal values
```

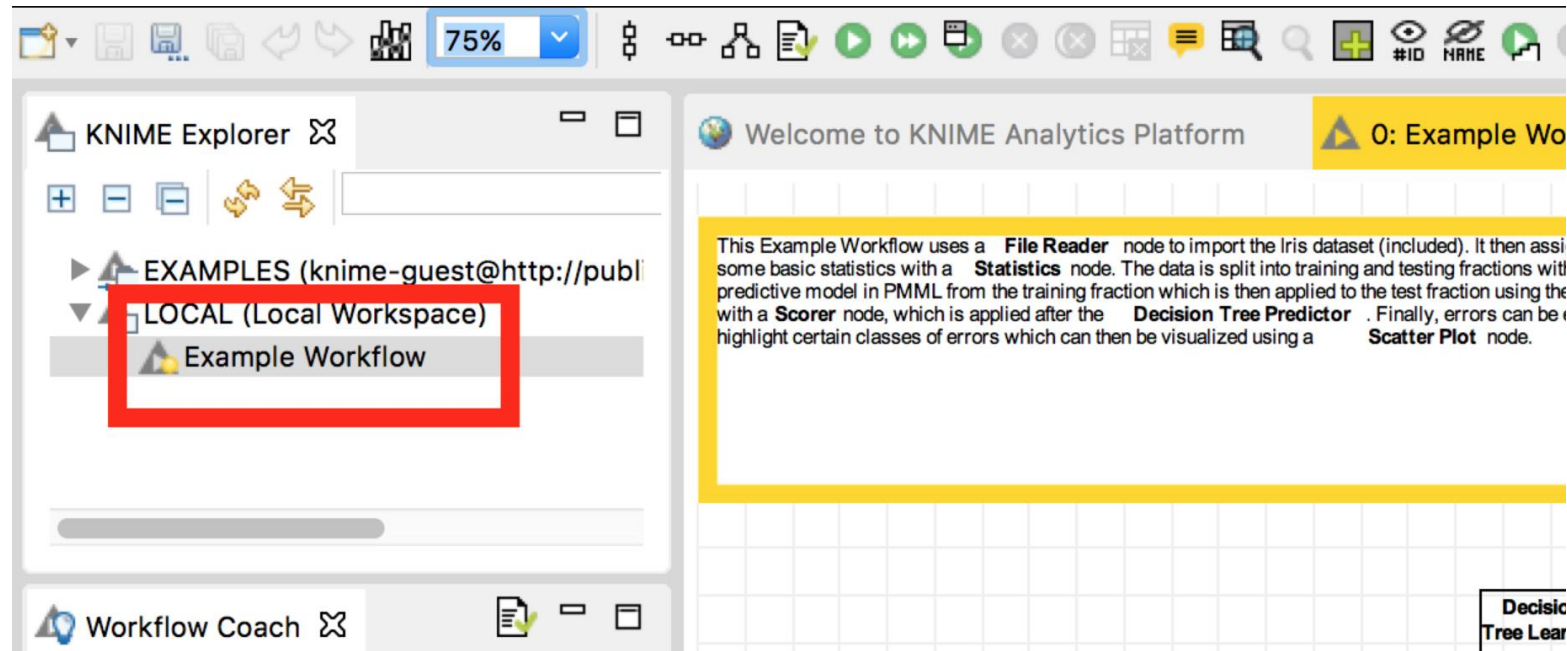


### 2.2 THE WORKSPACE

The workspace is the folder or directory of our computer where all the projects carried out with KNIME are stored. It will be necessary to choose a workspace before starting the program (you can also leave the folder that appears by default when installing).

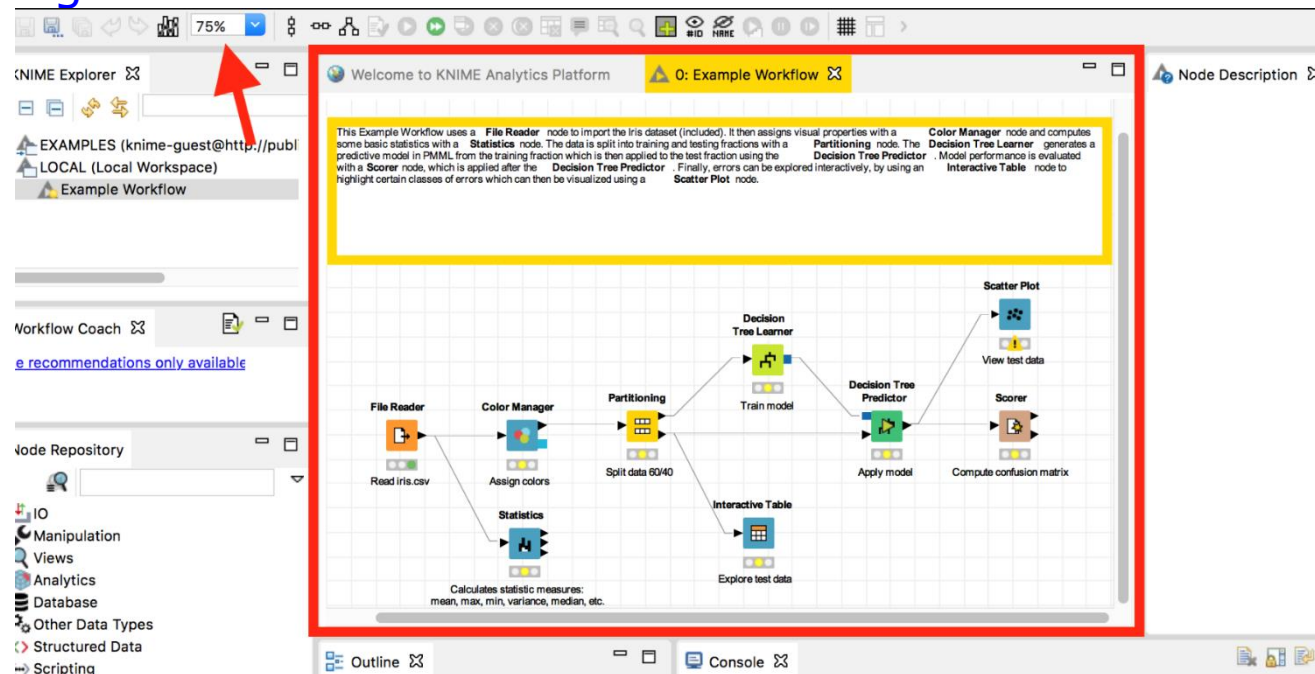
### 2.2 THE WORKSPACE: KNIME Explorer

This is the area where saved projects and workflows are managed. Where workflows are imported or exported.



### 2.2 THE WORKSPACE: Workflow editor.

This is the main working area, where nodes are dragged, connected and the workflow is configured.



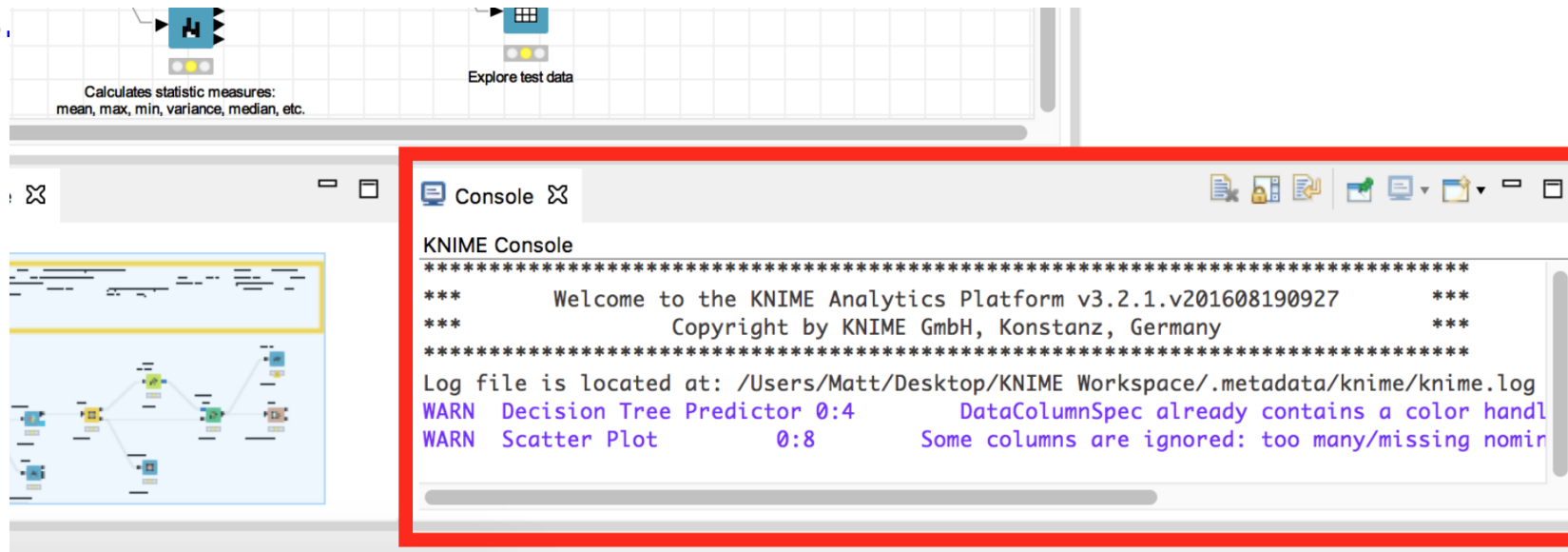
### 2.2 THE WORKSPACE: Outline.

It shows an overview of the workflow, making it easier to move from one part of the workflow to another when the workflow is very large.

The screenshot displays the KNIME workspace interface. On the left, a vertical toolbar lists various tool categories: IO, Manipulation, Views, Analytics, Database, Other Data Types, Structured Data, Scripting, Tool Integration, Community Nodes, KNIME Labs, Workflow Control, Social Media, Reporting, Chemistry, and ChemAxon / Infocom. The main workspace area shows a workflow with two nodes: 'Statistics' (labeled 'Calculates statistic measures: mean, max, min, variance, median, etc.') and 'Interactive Table' (labeled 'Explore test data'). A red rectangular box highlights the 'Outline' panel, which provides a hierarchical overview of the entire workflow. To the right of the Outline panel is the 'Console' panel, which displays the KNIME Console output, including a welcome message and warning messages: 'WARN Decision Tree Predictor 0:4' and 'WARN Scatter Plot 0:8'.

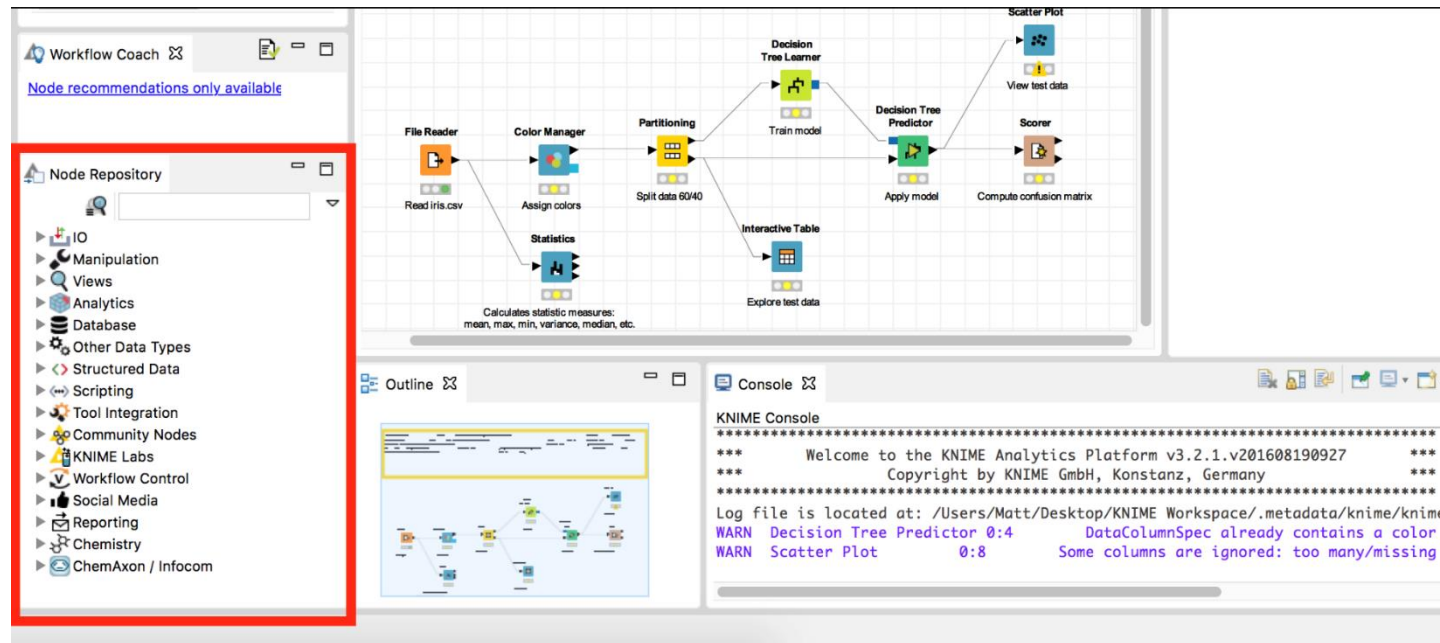
### 2.2 THE WORKSPACE: Console.

This is a text output field that displays warnings and errors that occur when executing the workflow. It also displays relevant information about the execution process.



### 2.2 THE WORKSPACE: Node repository.

This is the area where the nodes are organised by category. It is also possible to search for nodes by name. To use a node, simply select it and drag it into the editor.



## Additional Material: Using KNIME

### 2.2 THE WORKSPACE: Workflow coach.

If we have given permission for them to collect our usage data, this section makes suggestions as to which nodes we are most likely to need to use at any given time.

Recommended Nodes	Community
Decision Tree Predictor	85%
Decision Tree To Image	5%
Decision Tree to Ruleset	3%
PMML Writer	3%
Decision Tree View	1%
PMML To Cell	<1%
Boosting Learner Loop End	<1%
Model Writer	<1%
Model Loop End	<1%



## Additional Material: Using KNIME

### 2.2 THE WORKSPACE: Node description.

This is an information box that appears when a node is selected and displays information about the tasks the node performs and what its ports (inputs and outputs) are.

The screenshot displays the KNIME Analytics Platform interface. The main workspace shows a workflow with the following nodes: File Reader (Read Iris.csv), Column Filter (Postal ONLY), Color Manager (Assign colors), Statistics (Calculates statistic measures: mean, max, min, variance, median, etc.), Partitioning (Split data 60/40), Decision Tree Learner (Train model), Decision Tree Predictor (Apply model), Interactive Table (Explore test data), Scorer (Compute confusion matrix), and Scatter Plot (View test data). A yellow box highlights a text description of the workflow. A red box highlights the Node Description panel for the File Reader node, which contains the following text:

**File Reader**

**Click on the table header**

If the column header in the preview table is clicked, a new dialog opens where column properties can be set: name and type can be changed (and will be fixed then). A pattern can be entered that will cause a "missing cell" to be created when it's read for this column. Additionally, possible values of the column domain can be updated by selecting "Domain". And, you can choose to skip this column entirely, i.e. it will not be included in the output table then.

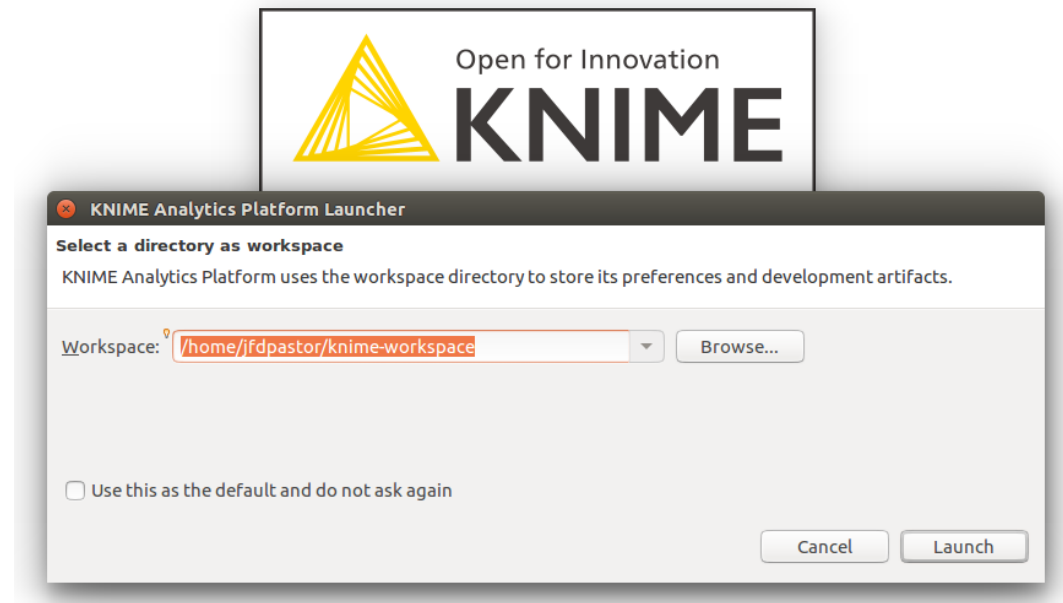
**Ports**

**Output Ports**

0 Datable just read from the file

### 3. Generic example: Classifying flower species.

Assuming that KNIME is already installed, go to the folder where it is located and run it by double **clicking** on its icon. When it opens, it will ask us for the "Workspace" folder. This is the folder where all our projects will be.



### 3. Generic example: Classifying flower species.

The screenshot displays the KNIME Analytics Platform interface. The main window shows a "Welcome back" message and a "Welcome to KNIME Analytics Platform" banner. The interface includes a "KNIME Explorer" sidebar on the left, a "Workflow Coach" section with a table of recommended nodes, a "Node Repository" sidebar, and a "Console" at the bottom right.

Recommended Nodes	Community
File Reader	24%
CSV Reader	18%
Excel Reader (XLS)	17%
Table Creator	12%
Database Reader (legacy)	7%
Table Reader	4%

```
KNIME Console
*****
*** Welcome to KNIME Analytics Platform v4.0.2.v201909300911 ***
*** Copyright by KNIME AG, Zurich, Switzerland ***
*****
Log file is located at: /home/jfdpastor/knime-workspace/.metadata/knime/knime.log
```

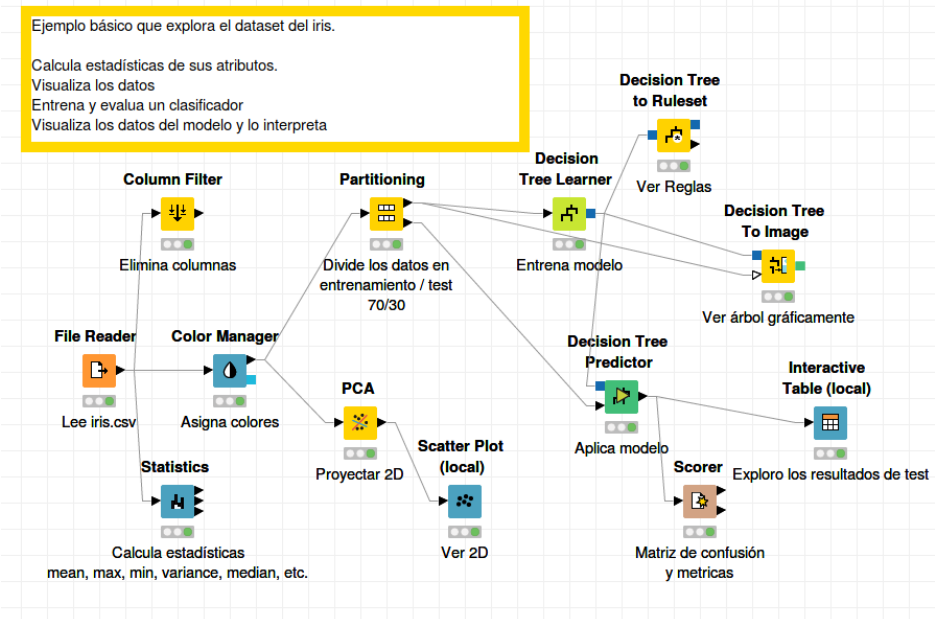
### 3. Generic example: Classifying flower species.

- You need to download the zip file called "Minería\_Ejemplo.zip".
  - Do NOT unzip it.
- In KNIME Explorer right click and then "Import KNIME workflow ...".
- Then the option "Select file" → "browse".
- Select it and click "Ok".

### 3. Generic example: Classifying flower species.

It is a basic workflow, which works with the iris dataset. Iris consists of 150 examples belonging to 3 different flower species. Each example has 4 attributes describing the flower: sepal length, petal length, sepal width, petal width. With this data set we will:

- Calculate statistics of its attributes.
- Visualise the data
- Train and evaluate a classifier.



### 3. Generic example: Classifying flower species.

In the Editor, we can see a series of interconnected nodes. On this editor, nodes are dragged, joined together, configured and executed to perform operations and analysis on the data.

It has navigation tools such as zoom in/out (make bigger or smaller) and allows to add comments.

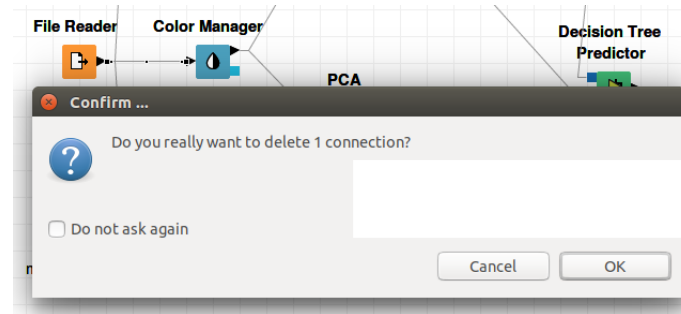
We can run each node or the whole workflow with buttons similar to "play".



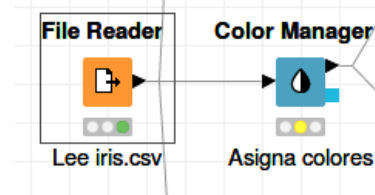
It is necessary to run the subsequent nodes each time a change is made to a node. That is to say, if we change any parameter of a node that is at the beginning of the workflow, we have to press the play button with two white arrows to re-execute all the subsequent nodes.

### 3. Generic example: Classifying flower species.

We are going to practice deleting and creating connections between nodes. For example, we can delete the connection between "File Reader" and "Color Manager".

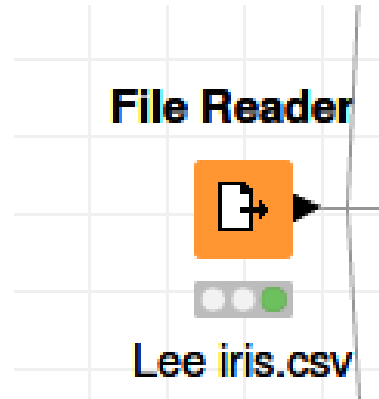


To recreate the connection: Select both, right-click, "connect selected nodes" or drag the mouse from the output port of "File Reader" to the input port of "Color Manager".



### 3. Generic example: Classifying flower species. Loading data

The "File Reader" node is the node used to load datasets (read the data from wherever it is stored). You can load data from a url (internet) or from your computer's hard disk.



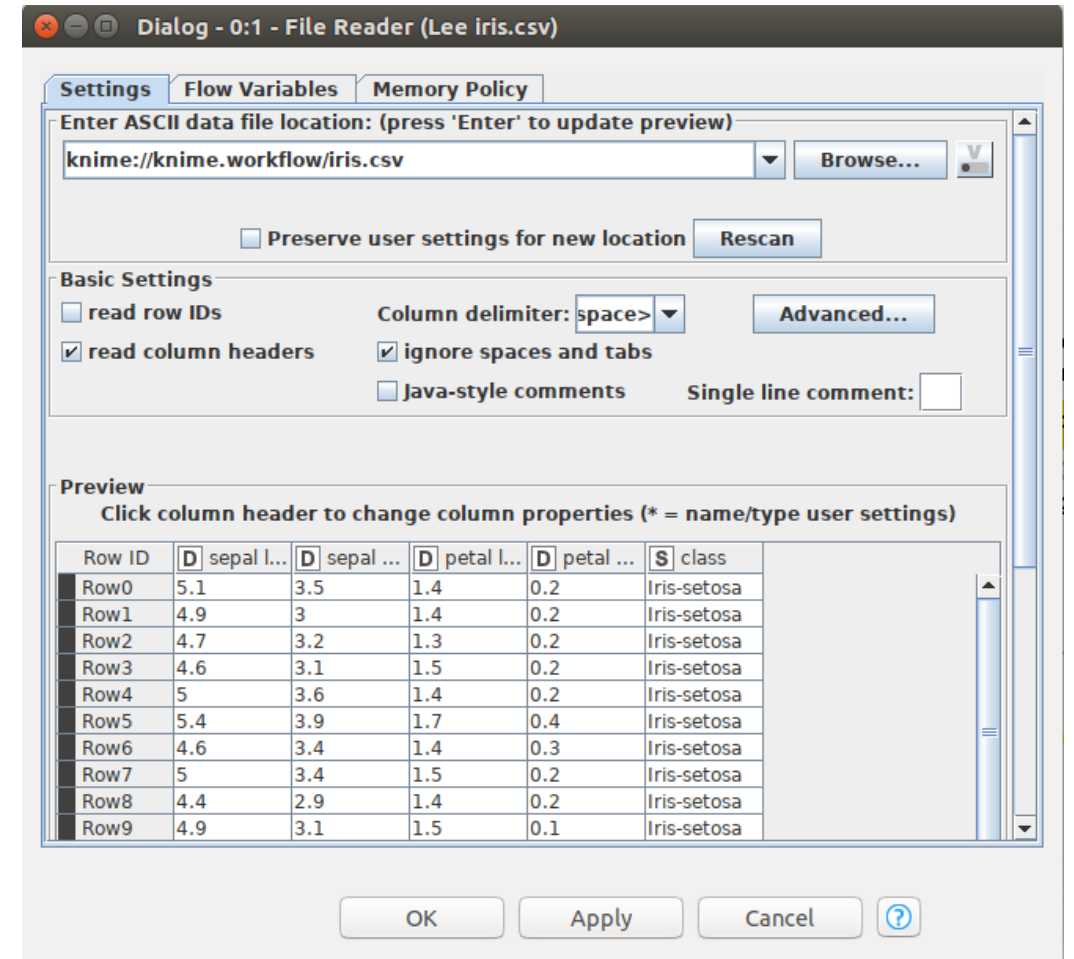


### 3. Generic example: Classifying flower species. Loading Data

We can configure the node by double clicking on it.

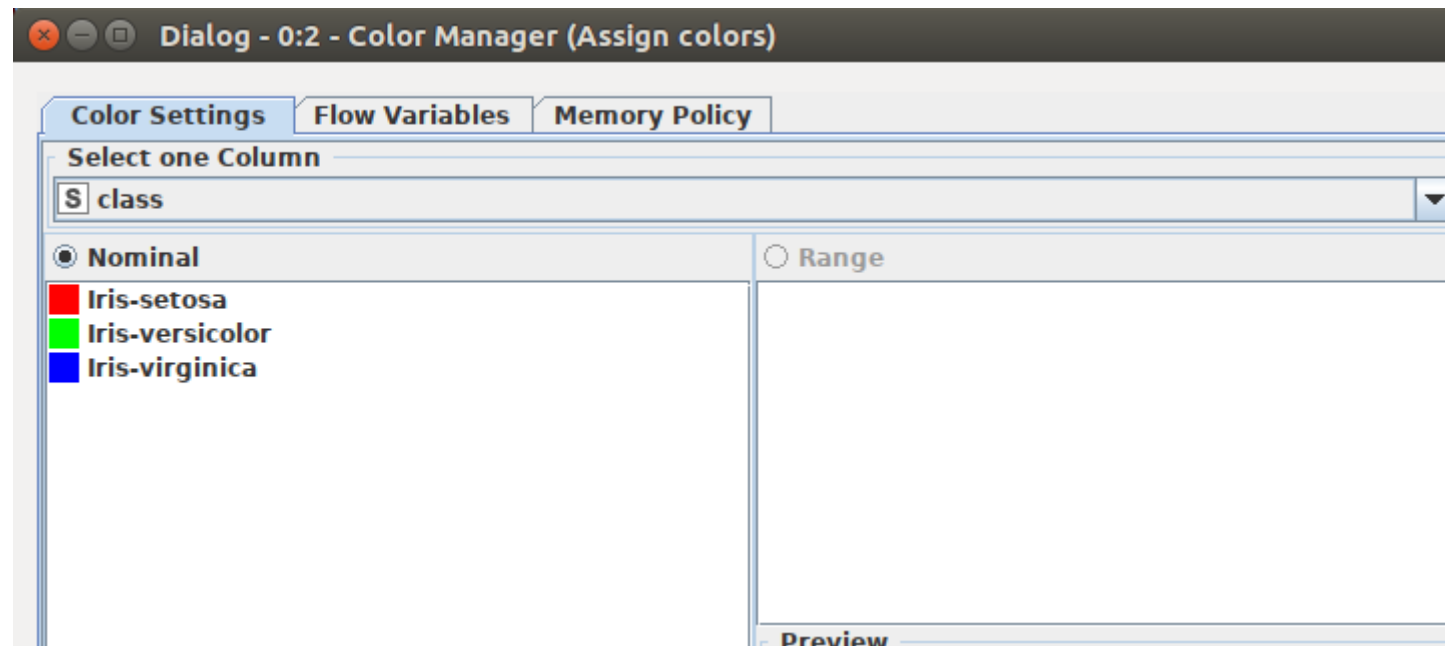
When configuring it, we can set the header or the delimiters of the file we want to load.

This is necessary, because sometimes we have data files separated by commas, sometimes by semicolons, sometimes by tabs, etc.



### 3. Generic example: Classifying flower species. Colouring data.

The "Colour Manager" node allows us to colour the dataset according to the values of one of its attributes.



### 3. Generic example: Classifying flower species. Colouring data.

The "Colour Manager" node allows us to colour the dataset according to the values of one of its attributes.

The result is a table in which each row is coloured according to the value of the attribute we have chosen.

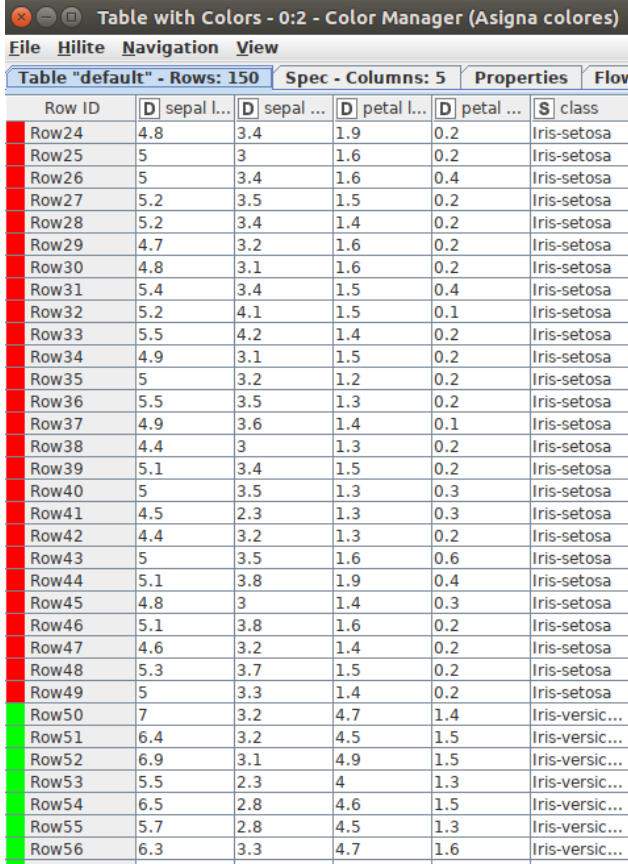


Table with Colors - 0:2 - Color Manager (Asigna colores)

File Hilite Navigation View

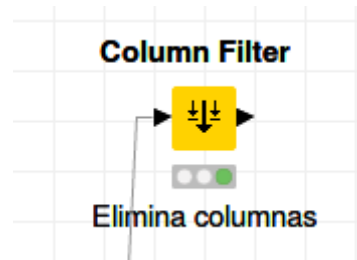
Table "default" - Rows: 150 Spec - Columns: 5 Properties Flow

Row ID	sepal l...	sepal ...	petal l...	petal ...	class
Row24	4.8	3.4	1.9	0.2	Iris-setosa
Row25	5	3	1.6	0.2	Iris-setosa
Row26	5	3.4	1.6	0.4	Iris-setosa
Row27	5.2	3.5	1.5	0.2	Iris-setosa
Row28	5.2	3.4	1.4	0.2	Iris-setosa
Row29	4.7	3.2	1.6	0.2	Iris-setosa
Row30	4.8	3.1	1.6	0.2	Iris-setosa
Row31	5.4	3.4	1.5	0.4	Iris-setosa
Row32	5.2	4.1	1.5	0.1	Iris-setosa
Row33	5.5	4.2	1.4	0.2	Iris-setosa
Row34	4.9	3.1	1.5	0.2	Iris-setosa
Row35	5	3.2	1.2	0.2	Iris-setosa
Row36	5.5	3.5	1.3	0.2	Iris-setosa
Row37	4.9	3.6	1.4	0.1	Iris-setosa
Row38	4.4	3	1.3	0.2	Iris-setosa
Row39	5.1	3.4	1.5	0.2	Iris-setosa
Row40	5	3.5	1.3	0.3	Iris-setosa
Row41	4.5	2.3	1.3	0.3	Iris-setosa
Row42	4.4	3.2	1.3	0.2	Iris-setosa
Row43	5	3.5	1.6	0.6	Iris-setosa
Row44	5.1	3.8	1.9	0.4	Iris-setosa
Row45	4.8	3	1.4	0.3	Iris-setosa
Row46	5.1	3.8	1.6	0.2	Iris-setosa
Row47	4.6	3.2	1.4	0.2	Iris-setosa
Row48	5.3	3.7	1.5	0.2	Iris-setosa
Row49	5	3.3	1.4	0.2	Iris-setosa
Row50	7	3.2	4.7	1.4	Iris-versic...
Row51	6.4	3.2	4.5	1.5	Iris-versic...
Row52	6.9	3.1	4.9	1.5	Iris-versic...
Row53	5.5	2.3	4	1.3	Iris-versic...
Row54	6.5	2.8	4.6	1.5	Iris-versic...
Row55	5.7	2.8	4.5	1.3	Iris-versic...
Row56	6.3	3.3	4.7	1.6	Iris-versic...
Row57	4.9	3.4	3.2	1	Iris-versic...

### 3. Generic example: Classifying flower species. Removing columns.

Column Filter: This is a node that allows you to choose which columns you want to exclude from the next steps of the analysis.

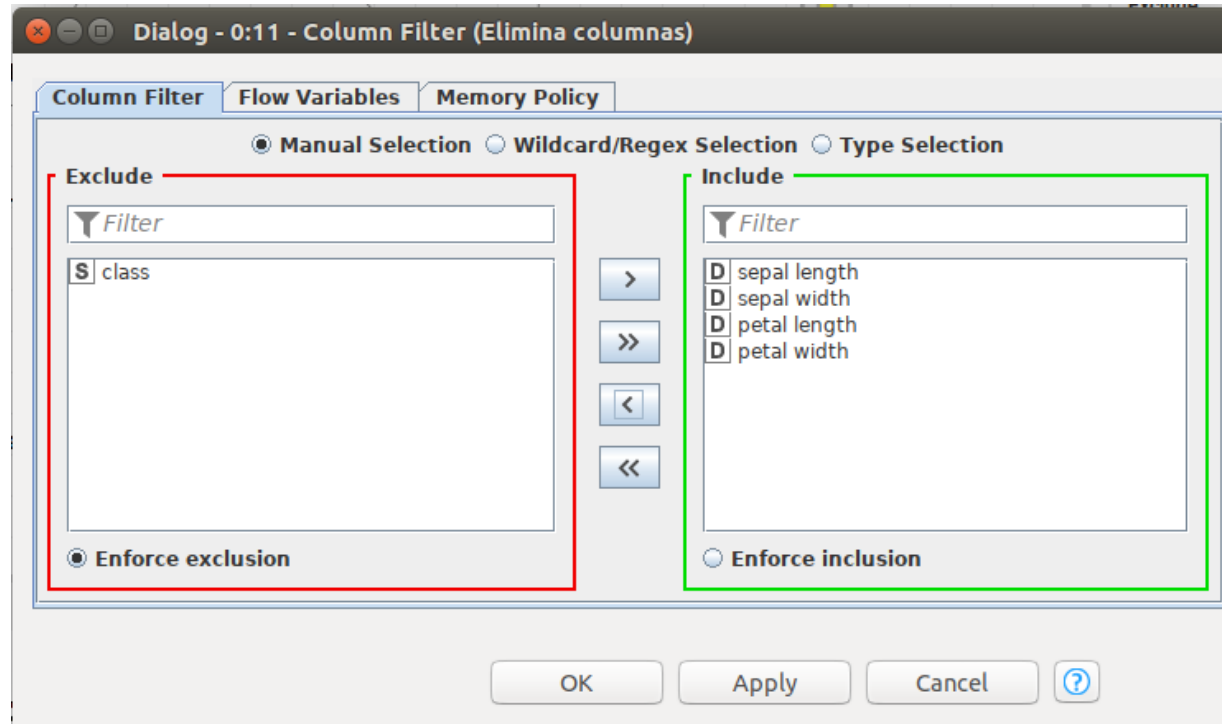
In some cases, it may be necessary to delete columns because they have unknown or erroneous values, in the example we are only going to delete to see what happens.



## Additional Material: Using KNIME

### 3. Generic example: Classifying flower species. Removing columns.

Double click on the node and choose the columns to be excluded from the following steps.

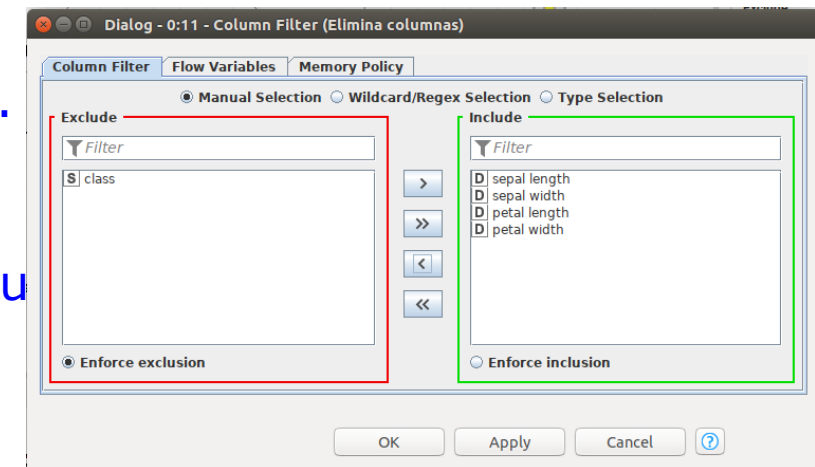


### 3. Generic example: Classifying flower species. Removing columns.

In the example, the column "Class", which contains the name of the species to which the flower described in each example belongs, will be deleted.

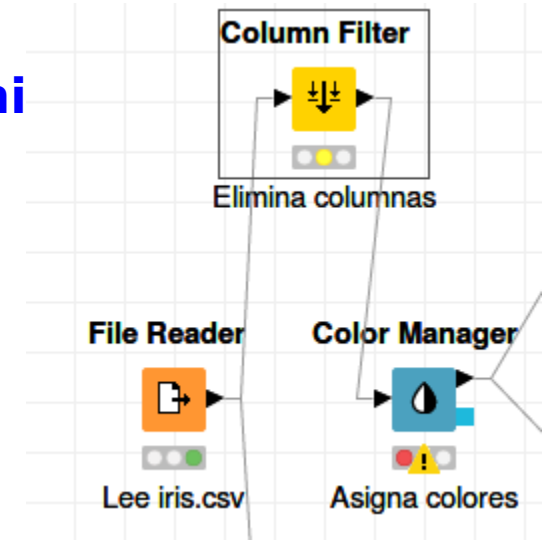
We are only going to do this to cause an error in the workflow.

Knowing how to identify the types of errors is fundamental to use KNIME



### 3. Generic example: Classifying flower species. Errors and warni

- Remove the connection between "File Reader" and "Color Manager".
- Configure "Column Filter" to remove the class.
- We connect "Column Filter" with "Color Manager".



Now we get an error in "Color Manager" because this node used the class to give colour to the examples.

**To continue, we re-establish the connection between "File Reader" and "Color Manager".**

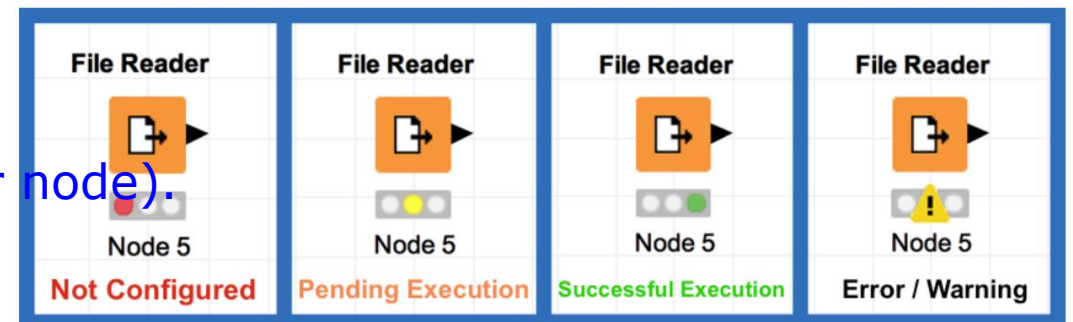
### 3. Generic example: Classifying flower species. Warnings and errors.

A node can be in 4 different states:

- Not configured. We must double click on it and choose some important parameter that the tool cannot choose for us.
- Pending. The execution button has yet to be pressed.
- Executed.
- Error/ Warning. Cannot be executed.

(as above, when deleting a column used by a later node).

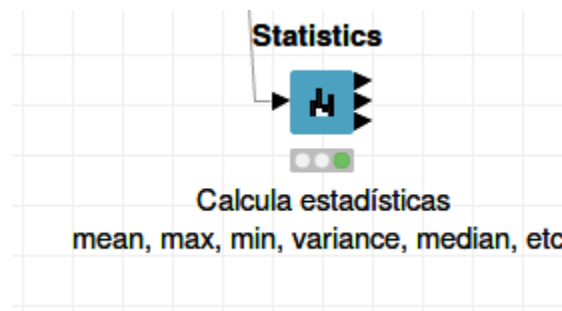
column used by a subsequent node)



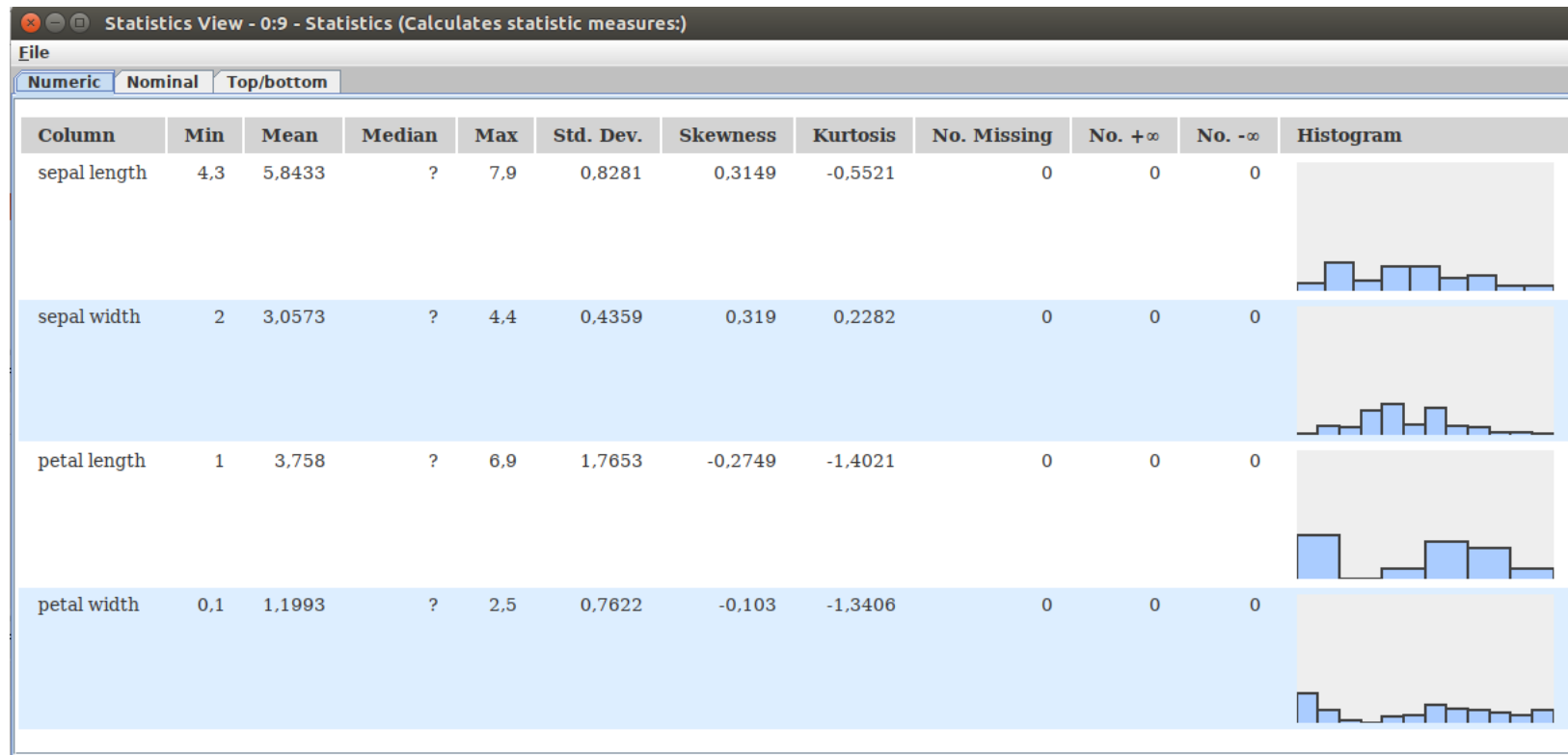


### 3. Generic example: Classifying flower species. Calculating statistics.

The "Statistics" node allows you to obtain statistics from a data table. By selecting the node and then clicking on "Statistics View" we can get a table with statistics for each of the attributes.

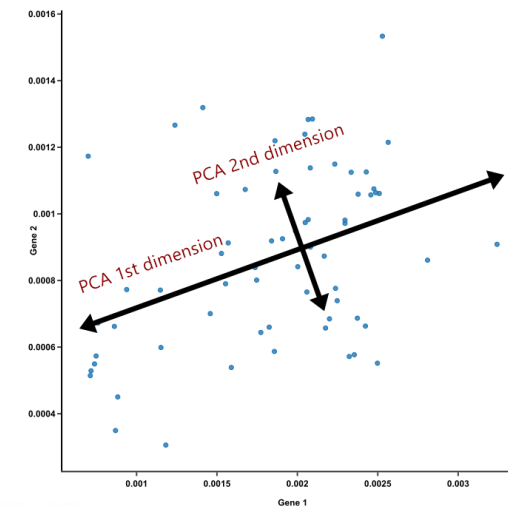
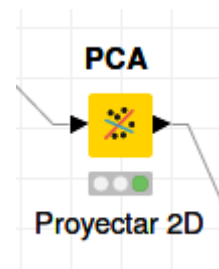


### 3. Generic example: Classifying flower species. Calculating statistics.



### 3. Generic example: Classifying flower species. Principal Component Analysis.

PCA (Principal Component Analysis) is a statistical technique used to describe a data set in terms of new, uncorrelated variables called "components". Components are ordered by the amount of original variance they describe, making the technique useful for reducing the dimensionality of a data set.

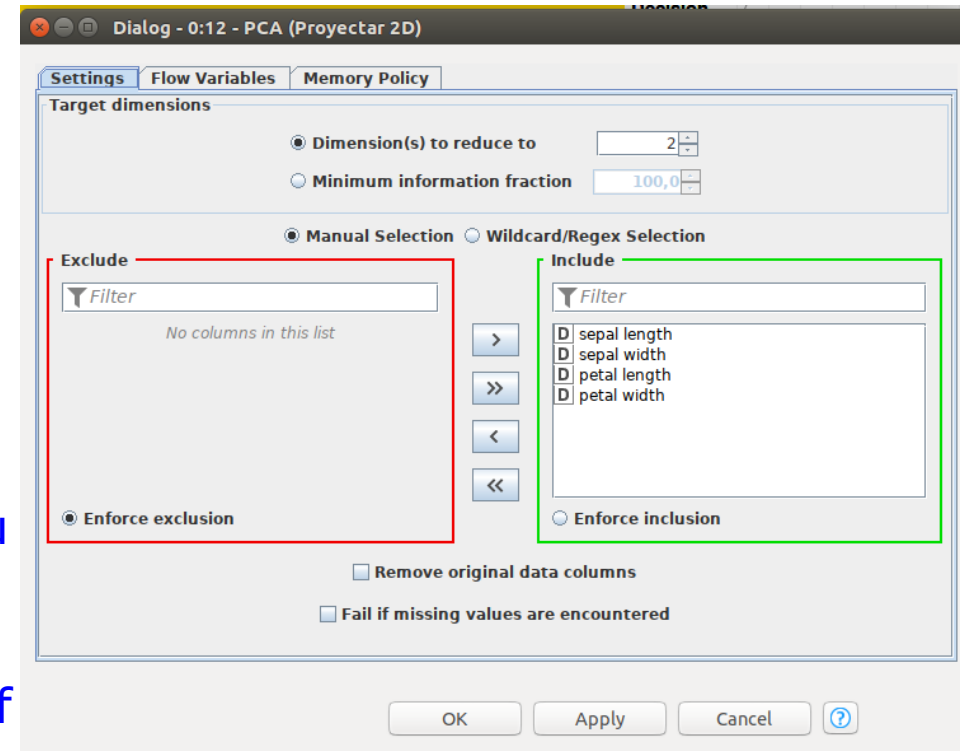


### 3. Generic example: Classifying flower species Principal Component Analysis

This technique helps us to visualise in 2D data sets that have more than 2 attributes, and so we can observe if there are outliers, overlapping between classes or if the boundary between classes is linear or non-linear.

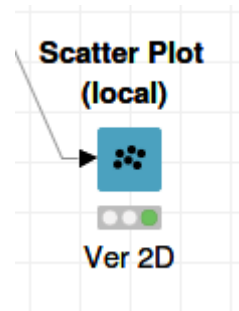
In the node you can configure how many components you want to calculate.

PCA creates new attributes, it does not visualise directly, if we want to visualise we have to connect a node to make graphs (we will see it below).



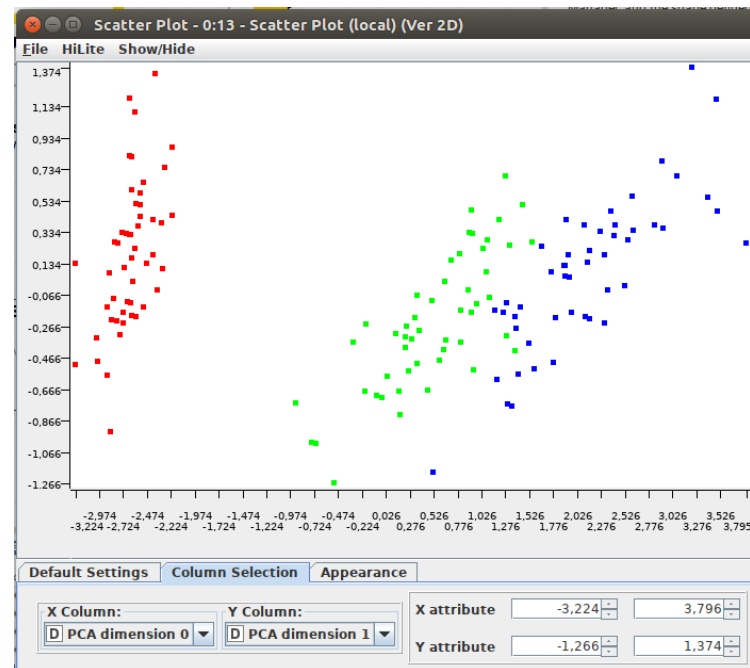
### 3. Generic example: Classifying flower species. Scatter plot.

The scatter plot allows us to visualise two attributes simultaneously. The examples will be represented as points in 2D space, at the coordinates defined by the value of its attributes.



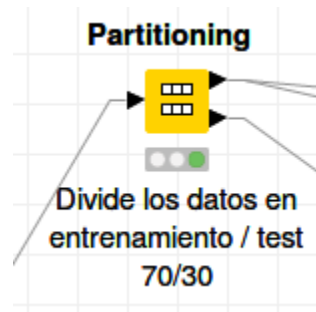
### 3. Generic example: Classifying flower species. Scatter plot.

In the example the first two principal components are being visualised, this technique allows to summarise and visualise (with two attributes) a data set with several columns.



### 3. Generic example: Classifying flower species. Data partitioning.

To properly evaluate a data mining algorithm, we must use a different set of data than the one used for training. We have often limited data, so we have to divide the data set into a training and a test part.

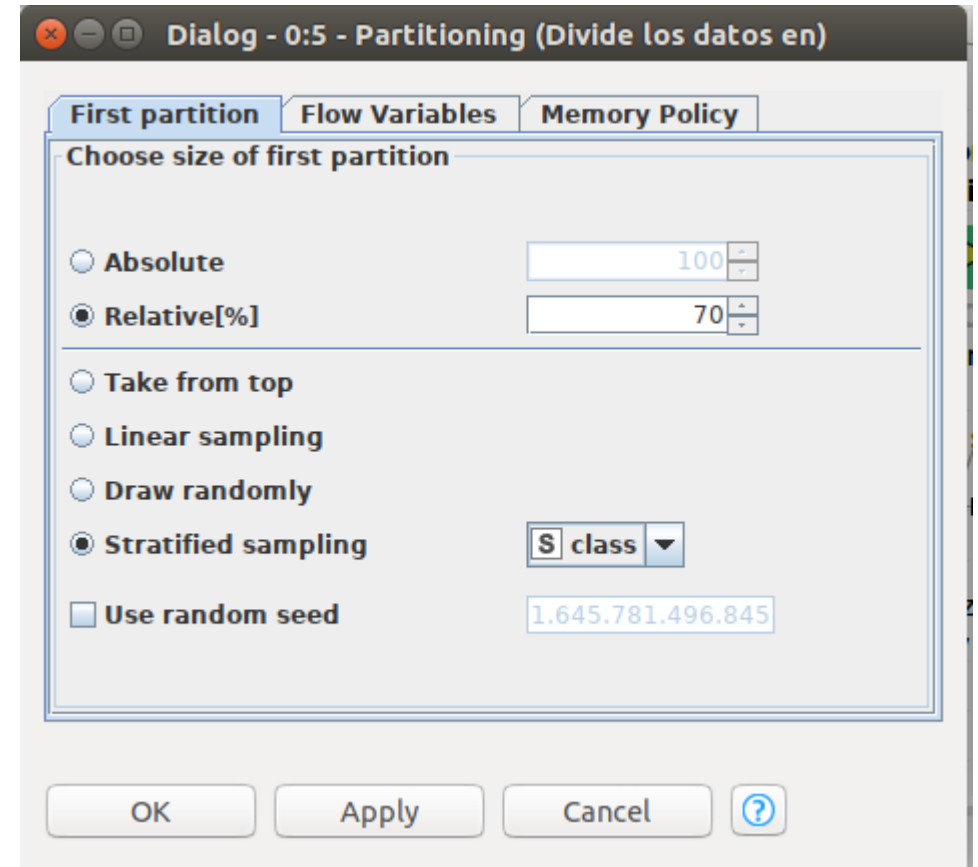


### 3. Generic example: Classifying flower species.

#### Data partitioning

In the node we can choose the % of instances we will use for training and the % we will use for testing.

There are options to make this partitioning completely random or "stratified" so that it maintains the proportion of classes in both training and testing.

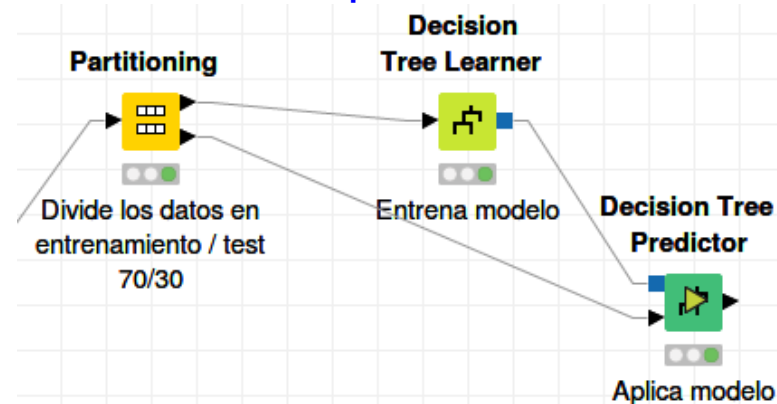




### 3. Generic example: Classifying flower species. Creation of data mining models.

In KNIME we have multiple learning algorithms. They are usually implemented by two nodes:

- "Learner" Builds the model (trains) from the data.
- "Predictor" Uses the trained model to predict labels from new, unlabelled data or to predict labels from test data and evaluate their performance.



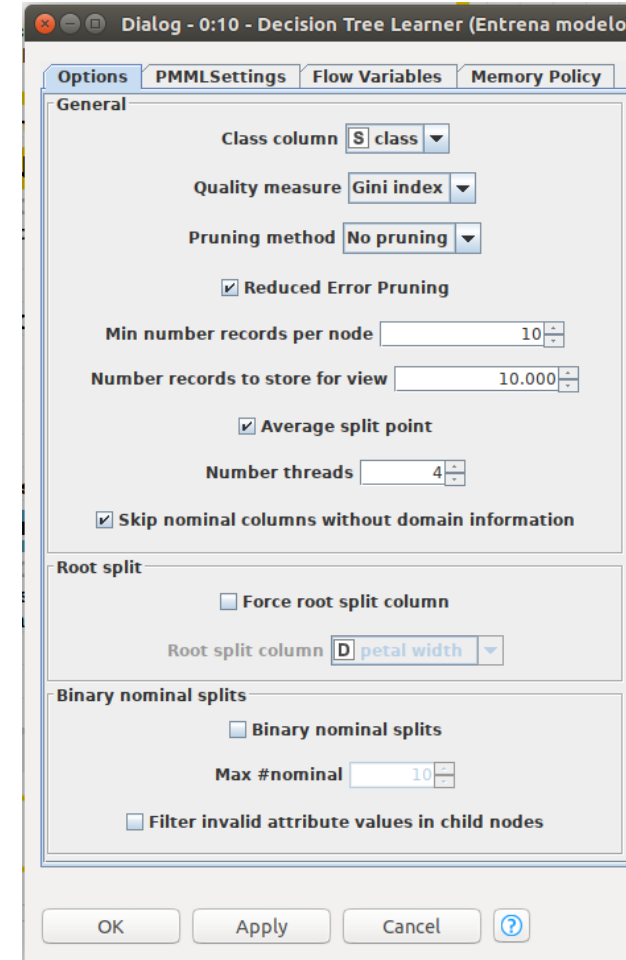
### 3. Generic example: Classifying flower species. Creation of data mining models.

In KNIME we have multiple learning algorithms. They are usually implemented by two nodes:

- "Learner" Builds the model (trains) from the data.
  - It results in a model that in some cases can be visualised.
- "Predictor" Uses the trained model to predict labels from the new data.
  - Results in a table, with a new column corresponding to the predictions.

### 3. Generic example: Classifying flower species. Creation of data mining models.

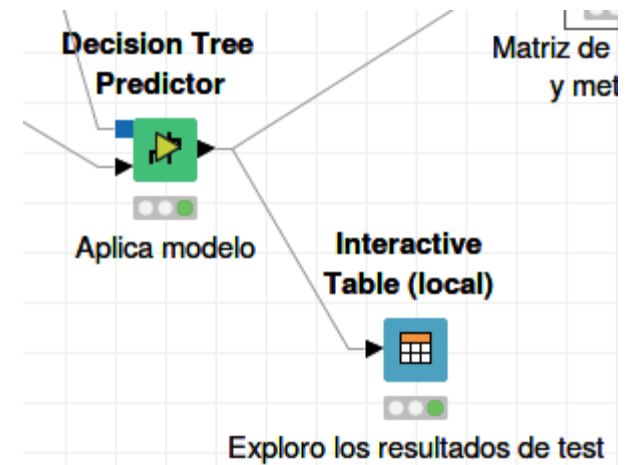
The example shows the interior of the KNIME node "Learner" of a classification tree. It allows you to configure, among other things, the quality measure of the attributes, whether it has pruning or not, etc.



### 3. Generic example: Classifying flower species. Display results.

We can use an interactive table type node to be able to display the values of the actual class and the predicted class, for all test examples.

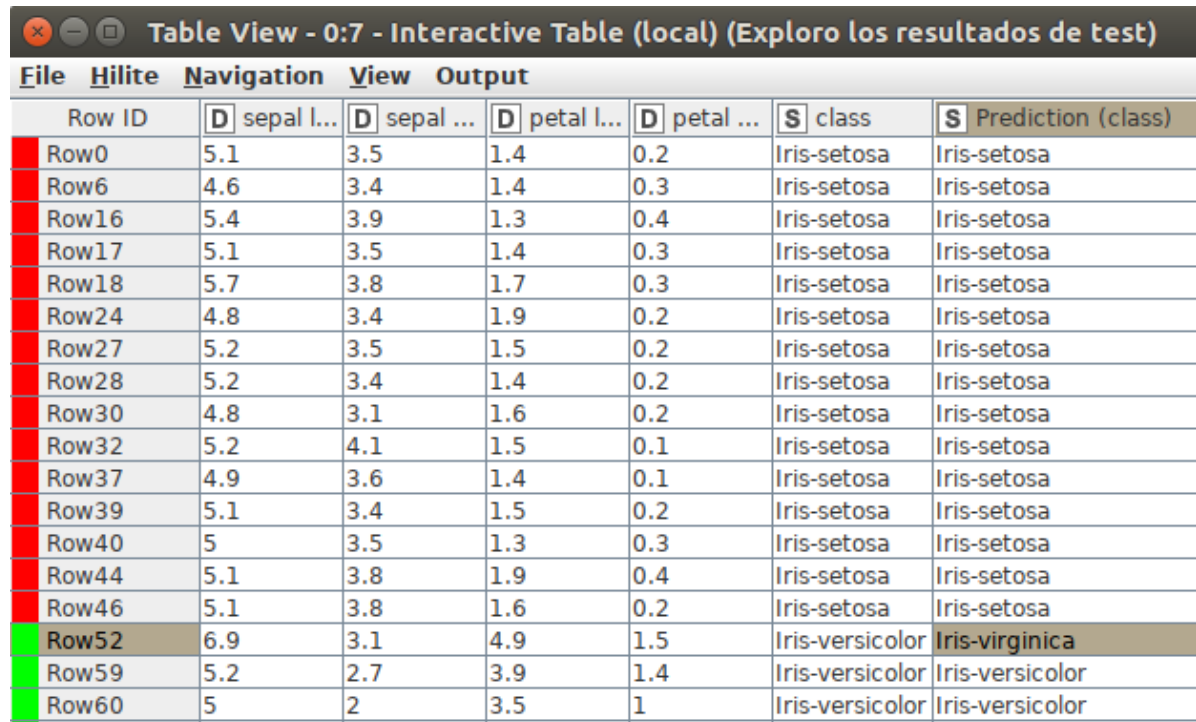
This way we can see the misclassified examples.



## Additional Material: Using KNIME

### 3. Generic example: Classifying flower species. Display results..

In this way we can see the misclassified examples.

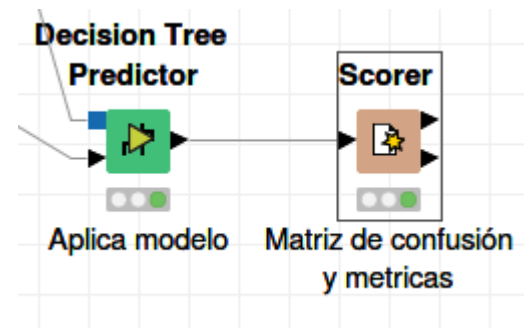


Row ID	sepal l...	sepal ...	petal l...	petal ...	class	Prediction (class)
Row0	5.1	3.5	1.4	0.2	Iris-setosa	Iris-setosa
Row6	4.6	3.4	1.4	0.3	Iris-setosa	Iris-setosa
Row16	5.4	3.9	1.3	0.4	Iris-setosa	Iris-setosa
Row17	5.1	3.5	1.4	0.3	Iris-setosa	Iris-setosa
Row18	5.7	3.8	1.7	0.3	Iris-setosa	Iris-setosa
Row24	4.8	3.4	1.9	0.2	Iris-setosa	Iris-setosa
Row27	5.2	3.5	1.5	0.2	Iris-setosa	Iris-setosa
Row28	5.2	3.4	1.4	0.2	Iris-setosa	Iris-setosa
Row30	4.8	3.1	1.6	0.2	Iris-setosa	Iris-setosa
Row32	5.2	4.1	1.5	0.1	Iris-setosa	Iris-setosa
Row37	4.9	3.6	1.4	0.1	Iris-setosa	Iris-setosa
Row39	5.1	3.4	1.5	0.2	Iris-setosa	Iris-setosa
Row40	5	3.5	1.3	0.3	Iris-setosa	Iris-setosa
Row44	5.1	3.8	1.9	0.4	Iris-setosa	Iris-setosa
Row46	5.1	3.8	1.6	0.2	Iris-setosa	Iris-setosa
Row52	6.9	3.1	4.9	1.5	Iris-versicolor	Iris-virginica
Row59	5.2	2.7	3.9	1.4	Iris-versicolor	Iris-versicolor
Row60	5	2	3.5	1	Iris-versicolor	Iris-versicolor

### 3. Generic example: Classifying flower species. Display results..

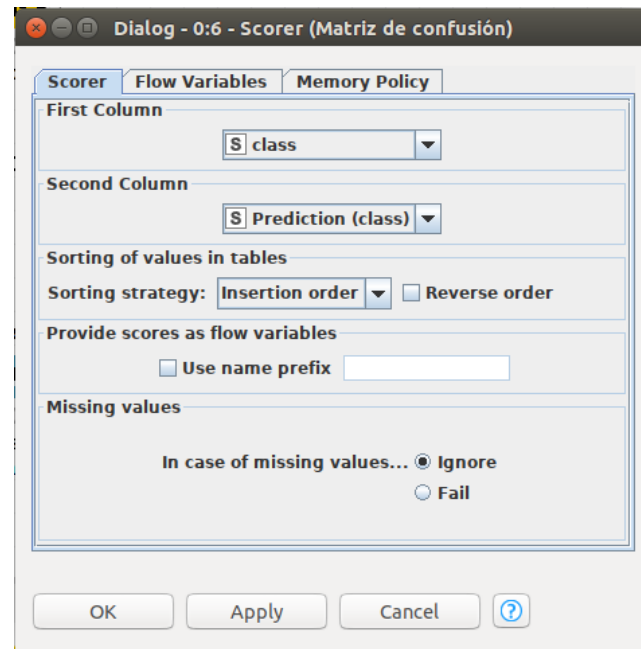
In addition to visualising the predictions, we can easily evaluate the quality of these predictions.

The "Scorer" node is used to evaluate the results. It can be used to obtain measures such as the hit rate and to obtain the confusion matrix.



### 3. Generic example: Classifying flower species. Display results..

To configure the "Scorer" node we must define which is the column representing the real class and which is the column for the class predicted by the model.



### 3. Generic example: Classifying flower species. Display results.

In the concrete example (iris data set, using a classification tree, 70% of the data to train and the remaining 30% to evaluate) it shows the overall hit rate of the model on the test data set (91%) and other metrics such as false positives or false negatives for each of the classes.

Row ID	TruePo...	FalseP...	TrueN...	FalseN...	Recall	Precision	Sensiti...	Specifity	F-mea...	Accur...	Cohen'...
Iris-setosa	15	0	30	0	1	1	1	1	1	?	?
Iris-versicolor	12	1	29	3	0.8	0.923	0.8	0.967	0.857	?	?
Iris-virginica	14	3	27	1	0.933	0.824	0.933	0.9	0.875	?	?
Overall	?	?	?	?	?	?	?	?	?	0.911	0.867



### 3. Generic example: Classifying flower species. Display results.

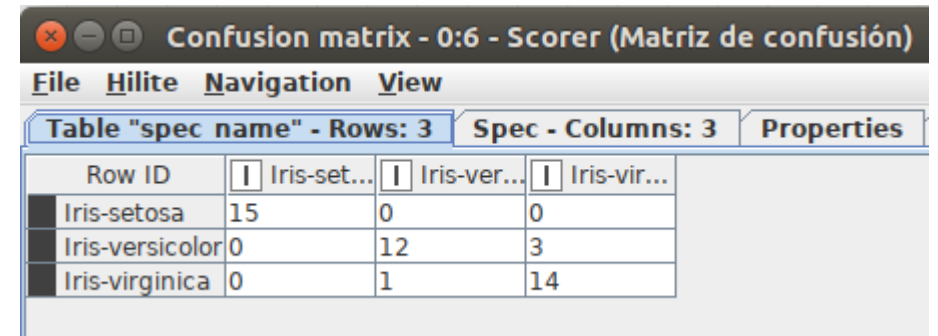
In the same node we can also obtain the confusion matrix, which relates the actual class to the predicted class, to observe the types of errors in the model.

In this table:

- Rows: These are the real classes.
- Columns: These are the predicted classes.

In the example:

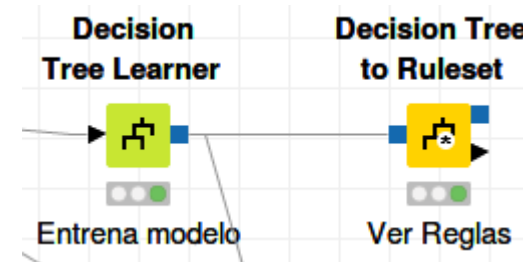
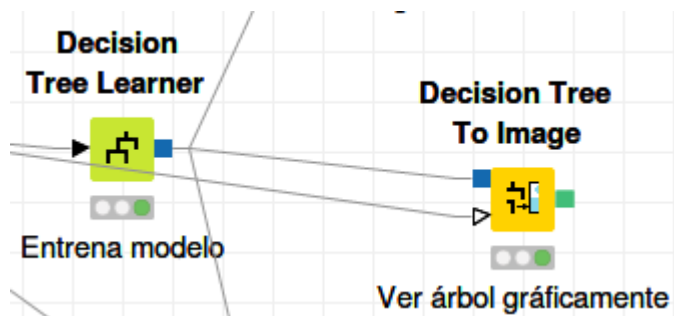
- 3 examples of iris-versicolour were misclassified as iris-virgin.
- 1 example of iris-virginia was wrongly classified as iris-versicolor.



Row ID	Iris-set...	Iris-ver...	Iris-vir...
Iris-setosa	15	0	0
Iris-versicolor	0	12	3
Iris-virginica	0	1	14

### 3. Generic example: Classifying flower species. Display the model.

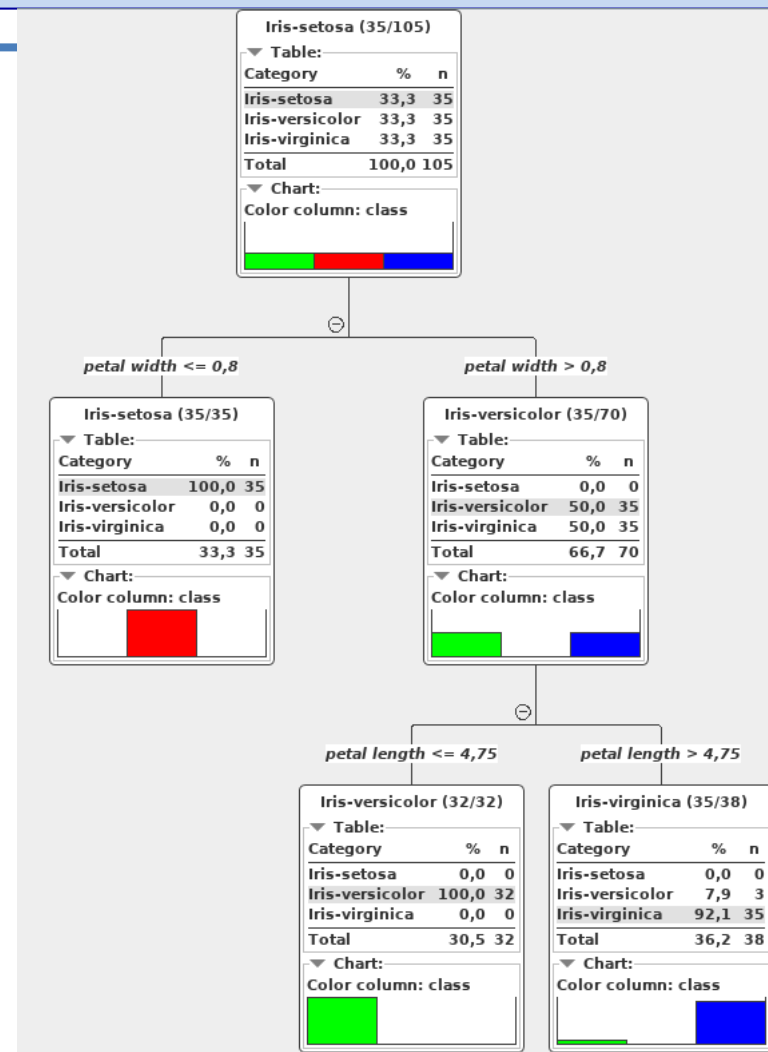
Some data mining models are interpretable, i.e. we can interpret how it arrives at a conclusion, how it classifies a certain example as one class rather than the opposite class. In conclusion, how it classifies a certain example as one class and not the opposite class. In the specific case of trees, we can see them in graphical form or in the form of rules, if they are too large to be interpreted correctly in graphical form.



### 3. Generic example: Classifying flower species. Display the model.

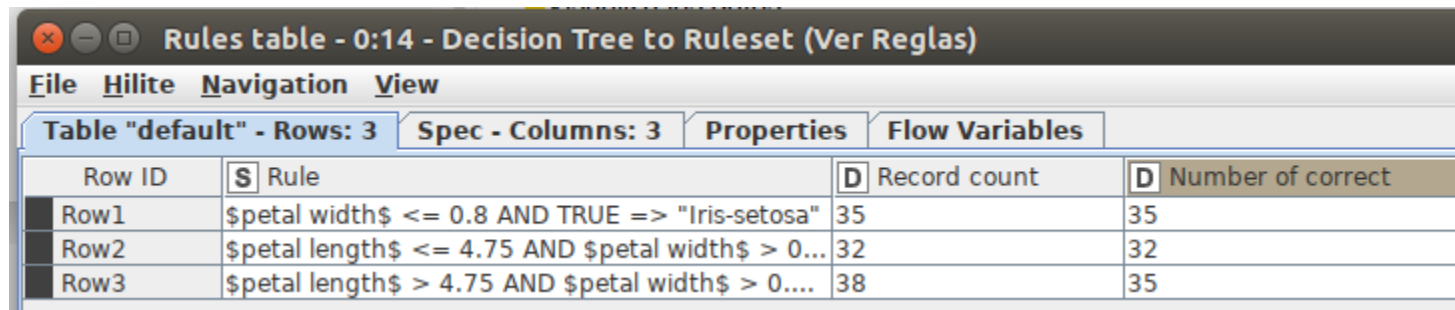
Visualising the classification tree.

- If the petal width is less than 0.8 we classify the example as setosa.
- If not
  - If the petal length is less than 4.75 we classify the example as versicolor.
  - If not, we classify the example as virgínica.



### 3. Generic example: Classifying flower species. Display the model.

A larger tree would be impractical to visualise graphically, so it can be translated into a set of rules which is a more compact representation.



Rules table - 0:14 - Decision Tree to Ruleset (Ver Reglas)

File Hilite Navigation View

Table "default" - Rows: 3 Spec - Columns: 3 Properties Flow Variables

Row ID	Rule	Record count	Number of correct
Row1	\$petal width\$ <= 0.8 AND TRUE => "Iris-setosa"	35	35
Row2	\$petal length\$ <= 4.75 AND \$petal width\$ > 0...	32	32
Row3	\$petal length\$ > 4.75 AND \$petal width\$ > 0...	38	35

## Additional Material: Using KNIME

### 4. Example with intelligent therapeutic intervention data (EarlyCare).

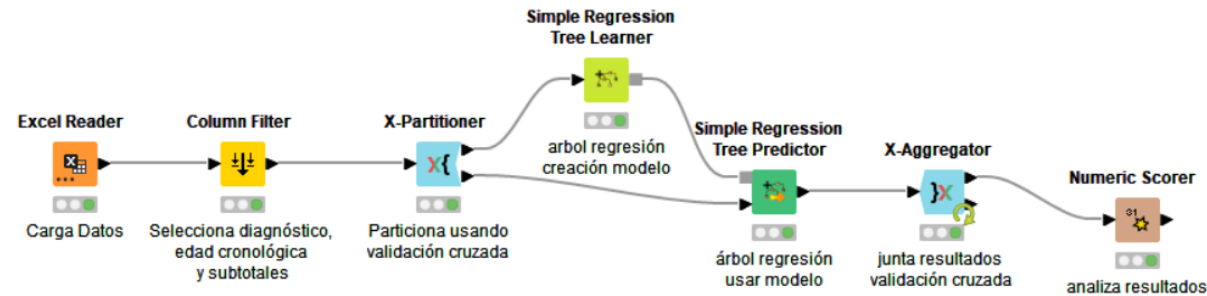
- You need to download the file called "eEarlyCare.knwf".
- In KNIME Explorer right click and then "Import KNIME workflow ...".
- Then the option "Select file" → "browse".
- Select it and click "Ok".

## Additional Material: Using KNIME

### 4. Example with intelligent therapeutic intervention data (EarlyCare).

It is a workflow using a dataset that uses the eEarlyCare scale items, chronological age and gender as independent variables and the main diagnosis as dependent variable.

Explore the workflow.



Ejemplo de regresión con los datos de eEarlyCare

## Web

KNIME → <https://www.knime.com/>



**THANK YOU VERY MUCH FOR  
YOUR ATTENTION!!!**



Co-funded by  
the European Union

