

Module 8

8.1 How to use KNIME: workflows

By **María Martínez Rojas**

Associate Professor CD, University of Granada

By **José Manuel Soto Hidalgo**

Associate Professor, CEAR, University de Granada

1. INTRODUCTION

KNIME is free¹ multiplatform software² with a graphical interface that allows users to implement the complete data science cycle in a simple, visual, and intuitive environment through data flows defined with interconnected nodes and code. In this context, *KNIME* provides tools for:

- Data visualization
- Data pre-processing
- Model extraction by algorithms
- Model comparison
- Analysis of results

In addition, *KNIME* integrates other platforms such as: BIRT (for report creation), WEKA (for data mining), Python, and R (for statistical analysis and visualization), as well as other extensions for data reading, ETL (Extraction, Transformation, and Loading of data), report generation, and visualization.

1.1. KNIME INSTALLATION

KNIME can be downloaded and used free of charge through the following link: <https://www.knime.com/downloads>. Because it is multiplatform, you can install the appropriate version for your version of the Windows, Linux, or Mac platforms. A quick guide to the installation of the application, as well as the basic operation of each element in the system, is available at <https://www.knime.com/installation>.

¹ Free software refers to software that can be used at no economic cost.

² 'Multiplatform' means that it can be used on different operating systems such as Windows, Linux, or Mac.

1.2. THE KNIME ENVIRONMENT

Once *KNIME* is installed, when running it we must indicate a workspace, which consists of a folder in a user directory on your computer in which all the data streams created by the user will be stored. We recommend you use the workspace assigned by default. Once the workspace is assigned, the *KNIME* environment is launched; it is divided into different zones which are labeled as shown in figure 1.

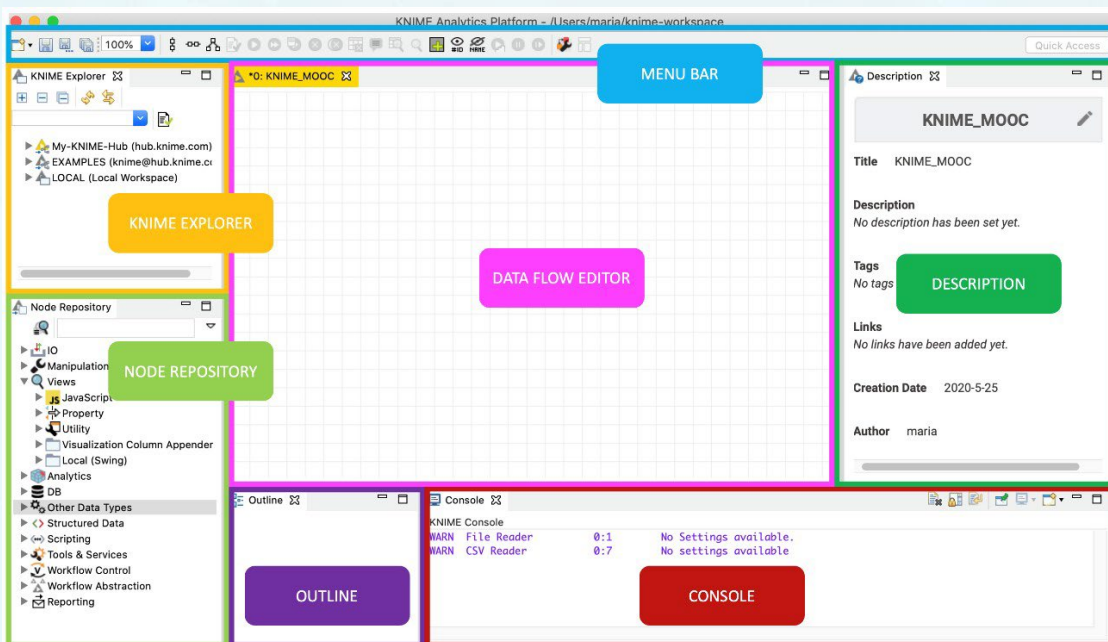


Figure 1. The home screen of the *KNIME* environment workspace.

1. Menu bar

The menu bar is located at the top. It consists of shortcuts to various options such as “save data flow”, “save as another workflow”, “align and configure nodes”, and “view results”, among others.

2. KNIME Explorer

The *KNIME Explorer* is located at the top left. This area includes the overview of the workflows as well as drop-down menus with various data flows available in the active *KNIME* workspaces (i.e., your local workspace and the *KNIME* servers). In the latter, you can download several specific data flows including “shopping cart analysis” with association rules or examples of node filtering.

3. Node repository

The node repository is located below the *KNIME Explorer* in the lower left corner. This section lists all the nodes available on the main *KNIME* platform as well as those of any installed extensions. The nodes are organized by category, but you can also use the search box at the top of the node repository to search for them. As we will see later, the use of *KNIME* is based on the design of data streams comprising interconnected nodes (icons).

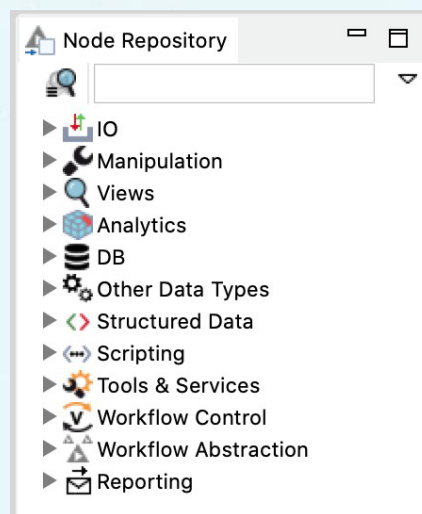


Figure 2. A node repository

The main categories of nodes are detailed below:

- Data input “[IO > Read]” and data output “[IO > Write]”.
- Preprocessing “[Data Manipulation]”, used to filter, discretize, normalize, and select variables, among other functions.
- Data visualization “[Views]” to display the results on screen, either textually or graphically.
- Data mining “[Mining]”, to build models such as association rules, clustering, classification, principal components analysis, etc.
- Other specific node types whose use is beyond the scope of this current course.

4. Outline

The description and overview of the currently active data stream can be seen in this section, located to the right of the node repository.

5. Console

This is located just to the right of the console outline and displays run messages that tell the user what happens when the workflow is executed.

6. Data flow editor (Workflow editor)

The area used to create and design the data flow is located in the central part of the screen.

7. Description

At the top right is the description of the selected nodes in both the node repository and the workflow nodes.

1.3. NODES AND DATA FLOWS

As previously mentioned, the use of *KNIME* is based on the design of data flows that represent the different stages of a knowledge extraction project. These data flows comprise nodes (icons) connected by a series of input and output connectors called ports that are interconnected to define the workflow. In this sense, the outputs of some nodes are used as inputs to other nodes where they transport data through the ports. Each node implements various procedures and processes, etc., and are basically data flow processing units.

1.3.1 Nodes

Nodes can perform all kinds of tasks, including reading and writing files, transforming data, generating models from datasets, and creating visualizations, etc. A node is visually represented in *KNIME* as a small icon comprising input and output ports, as well as the current node status. The input port is located to the left of the node, the output port to the right of the node, and the status at the bottom of the node; data flows through the ports.

1.3.2 Ports

Inputs are the data processed by the node and outputs are the resulting data sets. *KNIME* contemplates different types of input or output ports as data transmission mechanisms. These usually represent different types of data connections, especially data, databases, and models. The information is generally transferred in the form of Excel-like tables, with a header indicating the name of the variable and what type of data it is (e.g., text string, decimal, integer, etc.).

IMPORTANT! *Only ports of the same type can be connected, meaning we can connect two nodes through a data port but not through a data port and a model port.*

Figure 3 shows three nodes representing each of the types of ports mentioned above.

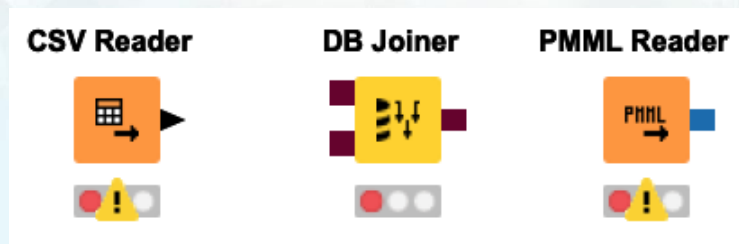


Figure 3. Three nodes with different port types.

As an example, the “CSV Reader” node has a data output port. This data type is tabulated, with each row being an instance of the problem and with each column being one of the variables. The output variable is usually placed at the end of the table and is represented by a black triangle. In contrast, the “DB Joiner” node has database input and output ports and is represented by a brown square.

The “PMML Reader” node has an output port represented by a blue square and provides the generated model. This port will be extensively used in this MOOC to connect “Learner” nodes with “Predictor” nodes in order to apply the taught model to another dataset. Figure 4 shows an example of a “Learner” node and how the taught model is transferred to another node through a model output port (blue square). It also shows how the “Learner” node is used as an input model to a “Predictor” node to perform predictions, specifically with a decision tree.

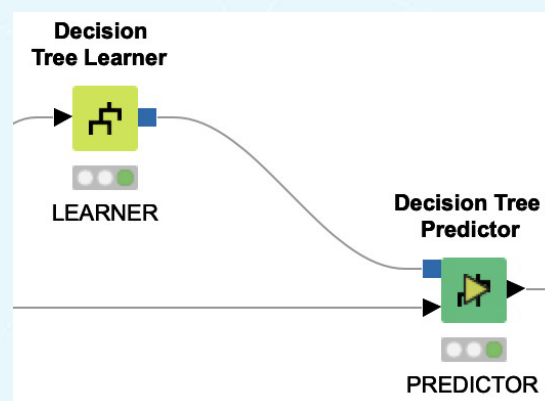


Figure 4. An example of model port.

1.3.3 The status of a node

The node status is represented by a traffic light with colors. Each color represents a status: red represents not configured, yellow represents configured but not executed, green represents executed, and a red circle with a cross represents an error in the node configuration. Figure 5 shows an example of a node, as well as the different statuses it represents.

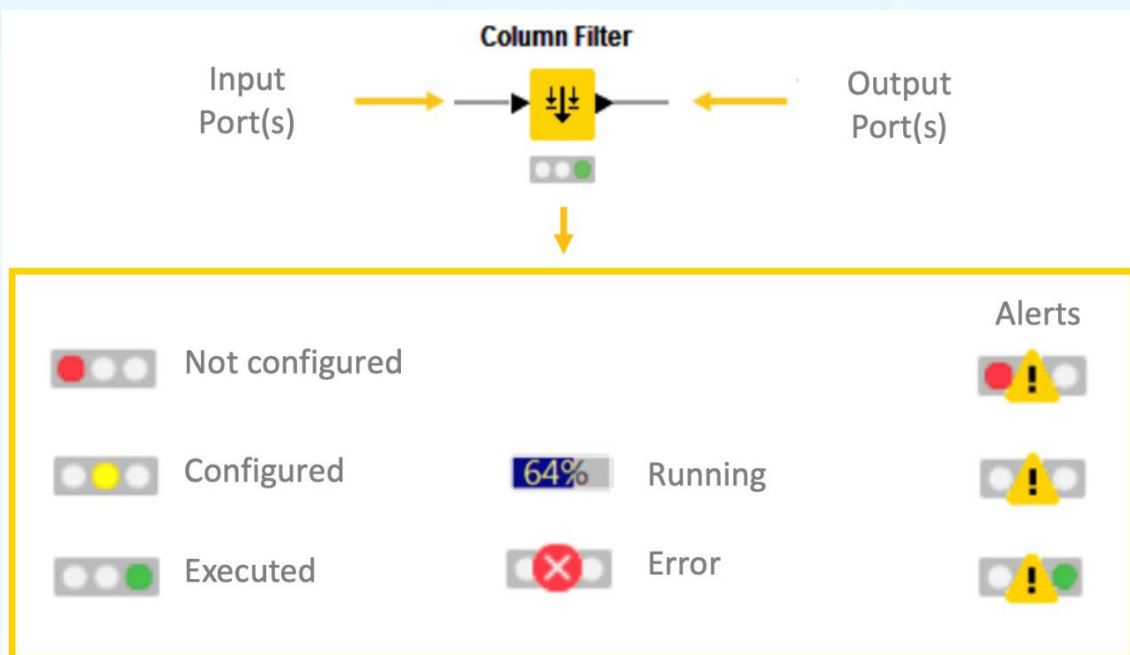


Figure 5. Examples of node statuses in KNIME.

- Red status: node not configured and not ready for execution; the node must be configured.
- Yellow status: node configured and ready for execution. If an exclamation triangle icon appears, it means that the node is ready for execution with the default parameters.
- Green status: the node is running and data is available on the node's output ports.
- Status red circle with a cross: the node cannot be executed because of a configuration error.

Alerts are displayed with a yellow triangle icon and an exclamation mark.

1.3.4 Metanodes

Metanodes are nodes that contain sub-flows of data, or in other words, although they can contain many nodes and even more metanodes inside them, they are seen as a single node in the main data flow. The use of metanodes is advanced and is beyond the scope of this course but we do present them here in this capsule as means to introduce them. Moreover, an example of a metanode for cross-validation is presented in Capsule 3 (*Supervised learning: regression and classification methods*).

To include a node in the data flow you can use the metanode wizard by selecting “Node/Add metanode” from the menu or by clicking on the button with the metanode icon in the toolbar (the workflow editor must be active). A predefined metanode can be created by selecting one of the templates and clicking “Finish”; it is then added to the data flow, as shown in figure 6.

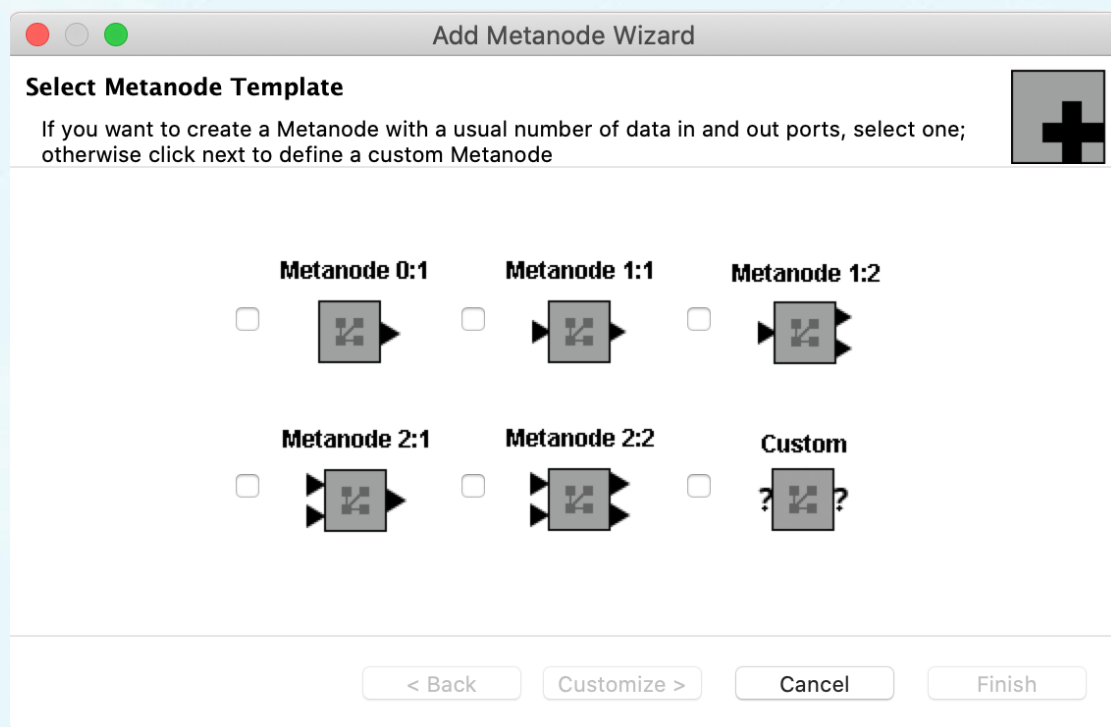


Figure 6. The menu used to create metanodes.

If you need to customize the metanode, for example, with a different number of input or output ports or if you want to use different types of ports, you can select one of the predefined metanodes as a template and then click on “Customize” to access the next page of the wizard. To open a metanode, you can double-click on it or choose “Open subdataflow editor” from its context menu. Once inside the metanode, you can include

nodes just as in the main data stream. Like nodes, metanodes have different states, as shown in figure 7:

Inactive/configured: if the metanode contains at least one node that is neither executed nor running.

- Running: if at least one node is running.
- Executed: if all the nodes included in the metanode are executed.

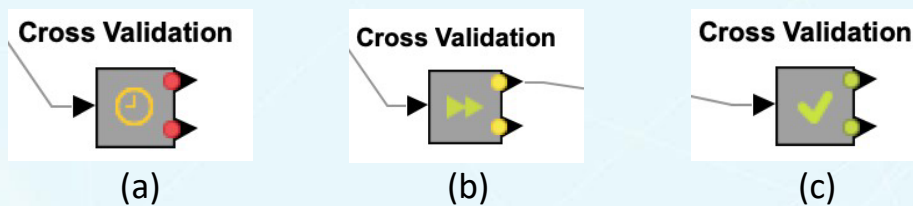


Figure 4. Possible statuses of the metanodes: (a) inactive, (b) running, and (c) correct.

As illustrated in figure 8, “Cross validation” is an example of a metanode. As shown, first the partitioning of the data set within this metanode is performed (via the “X-Partitioner” node), the classification algorithm is then executed (using the “Decision Tree Learner” and “Decision Tree Predictor” nodes), and finally the results of the cross validation are aggregated (by applying the “X-Aggregator” node).

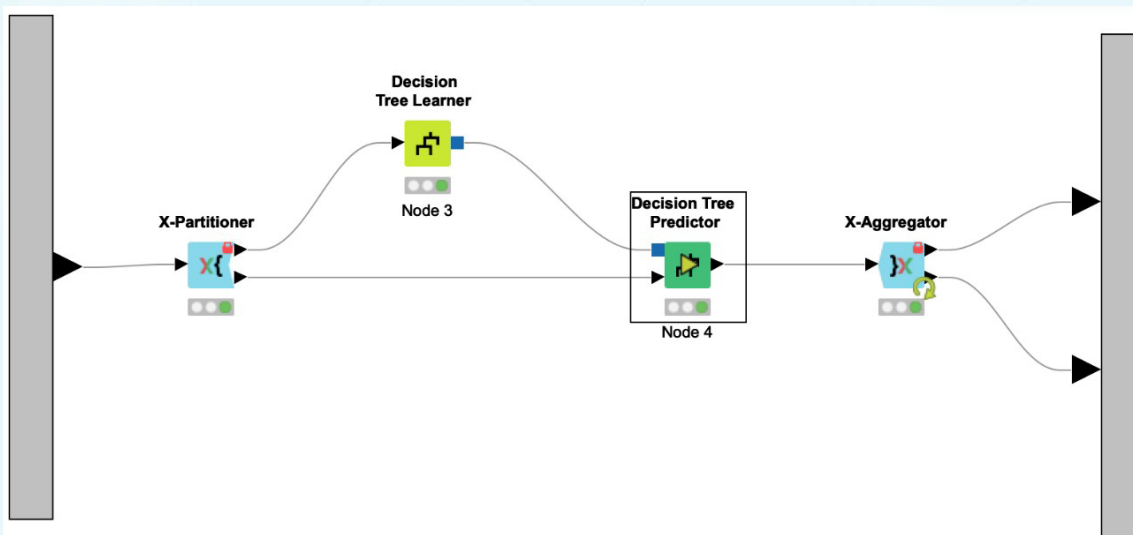


Figure 5. Cross validation of a metanode.

1.4 CREATING A DATA FLOW

A data stream comprises a collection of interconnected nodes. It generally represents part, or all, of a data science cycle. To create a data flow, the corresponding nodes are selected in the Node Repository and are dragged into the data flow editor area. As mentioned above, the outputs of some nodes are used as inputs of others and are connected by dragging the output port of one node to the input port of another one.

Here we will use the following data flow to illustrate how to create a data flow and how it displays the results:

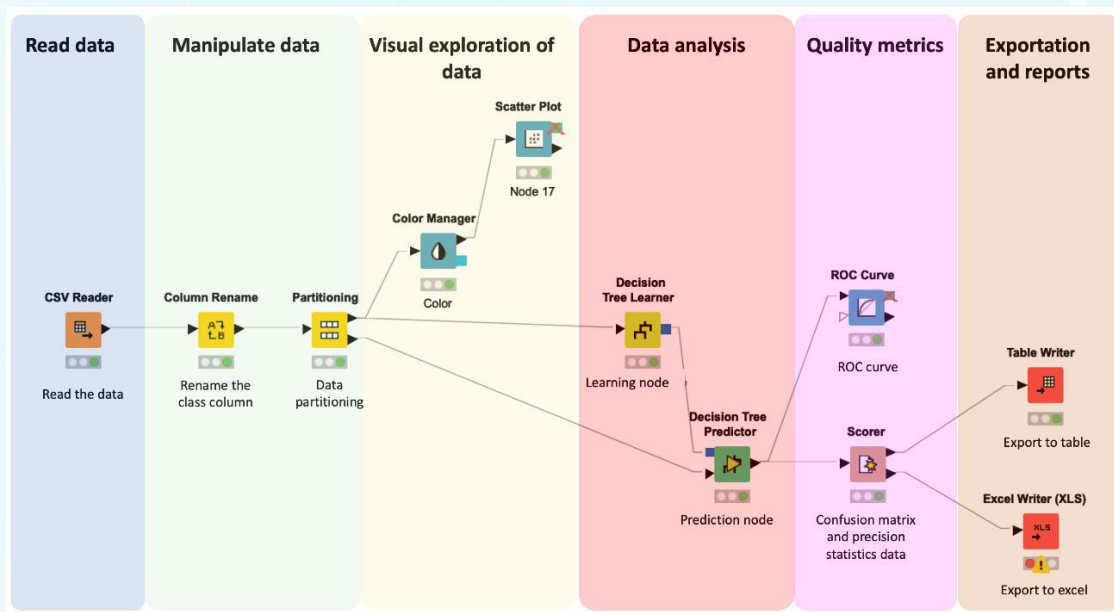


Figure 6 An example of a workflow.

The data flow in figure 9 shows the main stages of a knowledge extraction project. Starting with data reading (the initial blue area), moving to data manipulation (green area), visual exploration of the data (yellow area), the actual analysis to obtain a model (red area), an evaluation of the quality of the model obtained (violet area), and finally the report generation (beige area). The main aspects to consider when creating a data flow, such as setting up a node, running the node, and examining the node's output data are described below.

1.4.1 Configuring a node

To configure a node, simply select it in the workflow editor and press the 'F6' key or right click on your mouse to display the menu and select the "Configure" option (figure10).

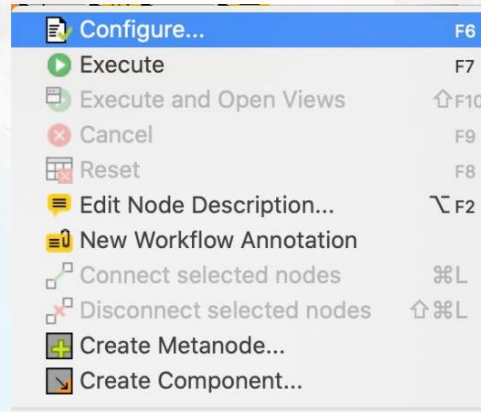


Figure 7. Menu used to configure a node.

1.4.2 Running a node

To execute a node, simply select it in the data flow editor and press the 'F7' key or right click on your mouse to display the menu and select the "Execute" option (figure 11). **IMPORTANT:** a node can only be executed if all its predecessor nodes in the flow have finished their execution. If we execute a node, all its non-executed predecessor nodes will automatically be executed (if they are configured and are ready to be executed).

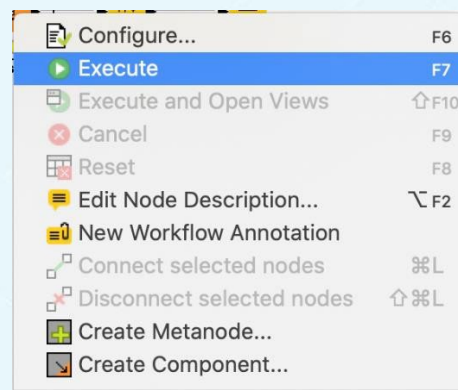


Figure 8: Node configuration menu.

1.4.3 Examine the output data of a node

If the node has been correctly executed (green traffic light), the data transported by the output ports can be displayed. To do this, click on the right mouse button and use the last part of the menu options to display the output port data. For example, for the "CSV Reader" node of the data flow in figure 12, which only has one output port, the data can be displayed in the form of a table by using the last menu option: "File Table" (figure 12a). While for the "Scorer" node, which has two output ports, the data corresponding

to the confusion matrix and statistics can be displayed through the “Confusion matrix” and “Accuracy statistics” options (figure 12b).

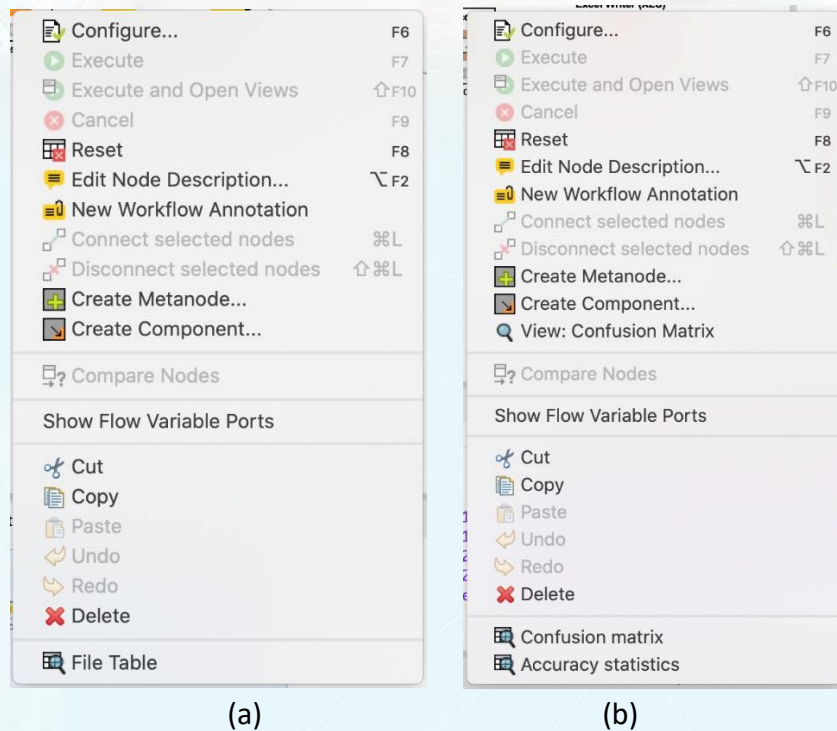


Figure 12: (a) Output of the CSV “Reader” node; (b) Output of the “Scorer” node.

REFERENCES

- Bakos, G. (2013). KNIME essentials. Packt Publishing Ltd.
- Blokdyk, G. (2019). KNIME a Complete Guide - 2019 Edition. Emereo Pty Limited, 2019.
- McCormick, K. (2019). Introduction to Machine Learning with KNIME. linkedin.com.
- Silipo, R. (2016). Introduction to Data Analytics with KNIME: A Data Science Approach to Analytics. O'Reilly.
- Silipo, R., & Mazanetz, M. P. (2012). The KNIME cookbook. KNIME Press, Zürich, Switzerland.
- Strickland, J. (2016). Data Analytics Using Open-Source Tools. Lulu. com.