

# KNIME TUTORIAL

---

Anna Monreale

KDD-Lab, University of Pisa

Email: [annam@di.unipi.it](mailto:annam@di.unipi.it)

# Outline

- Introduction on KNIME
- KNIME components
- Exercise: Data Understanding
- Exercise: Market Basket Analysis
- Exercise: Customer Segmentation
- Exercise: Churn Analysis

# What is KNIME?

- KNIME = Konstanz Information Miner
- Developed at University of Konstanz in Germany
- Desktop version available free of charge (Open Source)
- Modular platform for building and executing **workflows** using predefined components, called **nodes**
- Functionality available for tasks such as **standard data mining, data analysis** and **data manipulation**
- Extra features and functionalities available in KNIME by extensions
- Written in Java based on the Eclipse SDK platform

# KNIME resources

- Web pages containing documentation
  - [www.knime.org](http://www.knime.org) - tech.knime.org – tech.knime.org
  - installation-0
- Downloads
  - knime.org/download-desktop
- Community forum
  - tech.knime.org/forum
- Books and white papers
  - knime.org/node/33079

# Installation and updates

- Download and unzip KNIME
  - No further setup required
  - Additional nodes after first launch
- New software (nodes) from update sites
  - <http://tech.knime.org/update/community-contributions/release>
- Workflows and data are stored in a *workspace*



You are here: / [Home](#) / [Download KNIME Desktop & SDK](#)

## Forum & Documentation



## Download KNIME Desktop & SDK

Download the latest KNIME Desktop and KNIME SDK version 2.8.2 for Windows, Linux, and Mac OS X.

### KNIME Desktop

The KNIME Desktop version is intended for end users and provides everything needed to immediately begin using KNIME as well as extend KNIME with extension packages developed by others. The downloads also contain the [KNIME quickstart guide](#).

### Windows

Usually unzipping the archive somewhere on your hard drive is sufficient for the installation of KNIME. However, under Windows problems with the built-in unzip utility sometimes truncate file names. Therefore we offer self extracting archives:

- [KNIME for Windows 32bit \(self-extracting archive\)](#)
- [KNIME for Windows 64bit \(self-extracting archive\)](#)

If you are using a proper unzipper and want to use zip archives instead, you can find them [here](#).

### Linux

For Linux a 32 and 64bit build are available:

- [KNIME for Linux 32bit](#)
- [KNIME for Linux 64bit](#)

### Mac OS X

Since KNIME 2.3.0 we are proud to announce a fully supported KNIME build for Mac OS X. It requires a 64bit Intel-based architecture with Java 1.6:

# What can you do with KNIME?

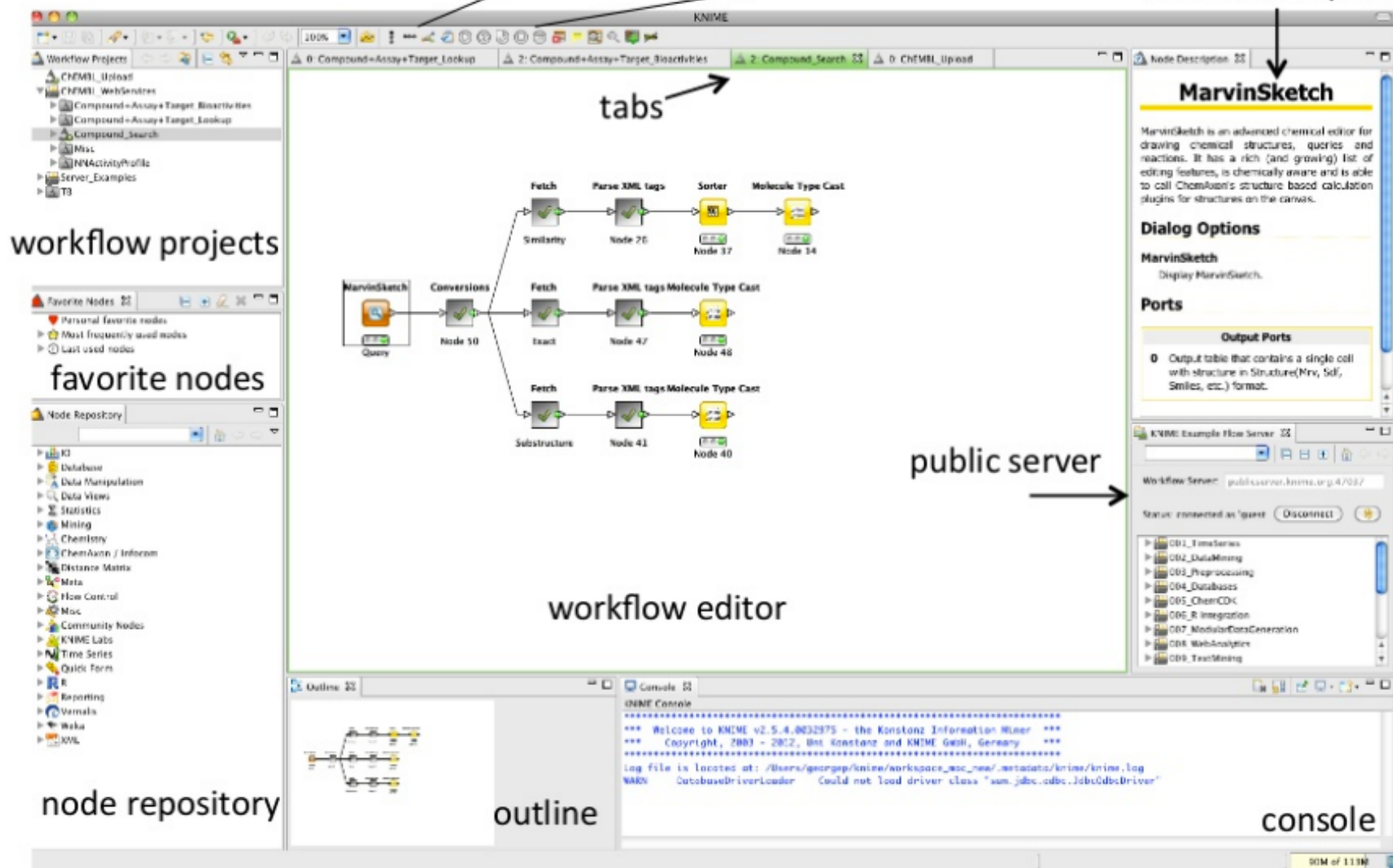
- **Data manipulation and analysis**
  - File & database I/O, filtering, grouping, joining, ....
- **Data mining / machine learning**
  - WEKA, R, Interactive plotting
- **Scripting Integration**
  - R, Perl, Python, Matlab ...
- **Much more**
  - Bioinformatics, text mining and network analysis

# KNIME Workbench

Auto-layout Execute Execute all nodes



Node description



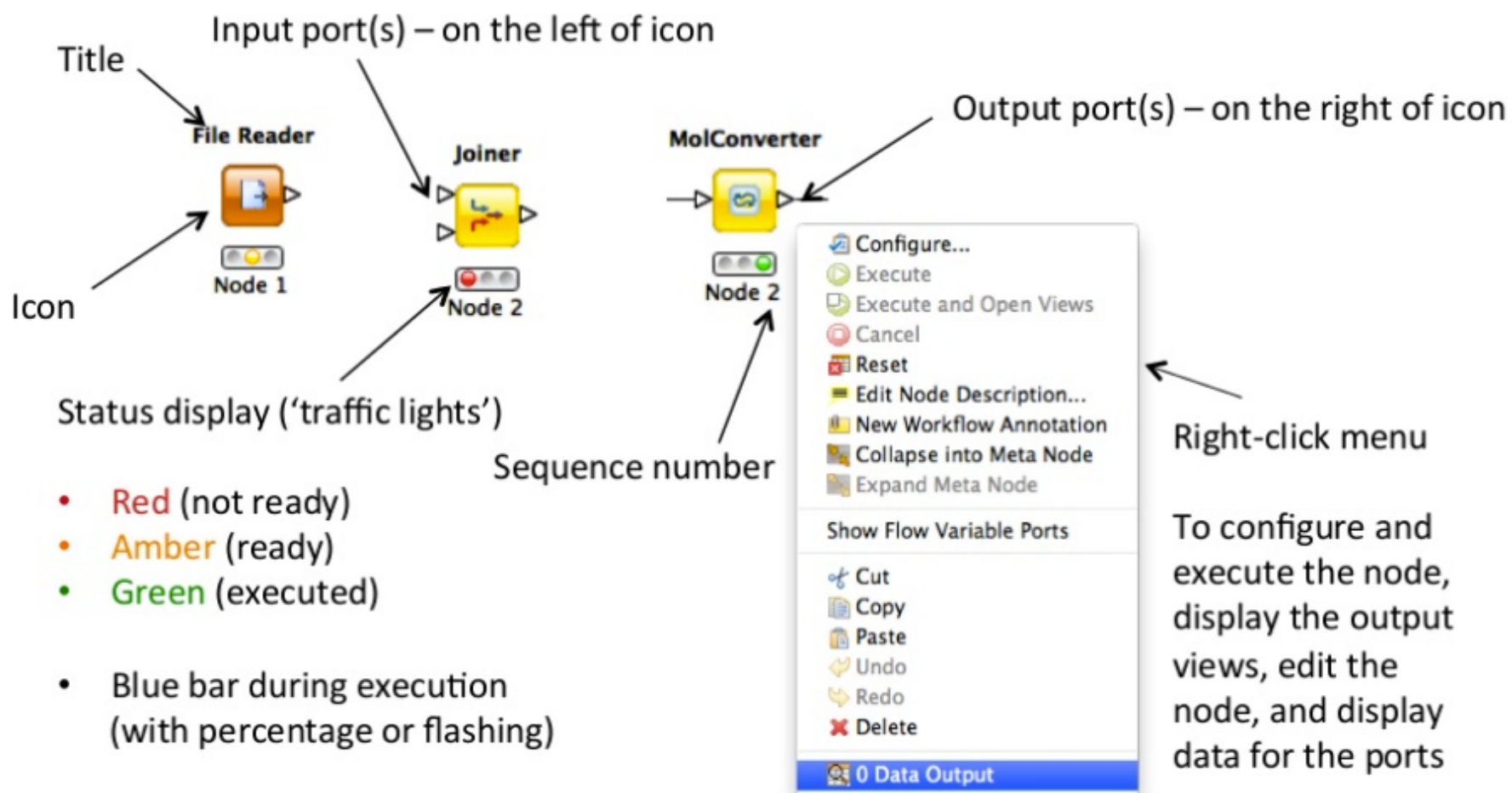
The screenshot displays the KNIME Workbench interface with several key components labeled:

- workflow projects:** Located in the top-left pane, showing a tree view of project folders like 'CHEMBL\_Upload' and 'Compound\_Search'.
- favorite nodes:** A pane below workflow projects showing 'Personal favorite nodes', 'Most frequently used nodes', and 'Last used nodes'.
- node repository:** A pane at the bottom left containing a categorized list of nodes such as 'Database', 'Data Manipulation', and 'Chemistry'.
- workflow editor:** The central workspace showing a workflow diagram with nodes like 'MarvinSketch', 'Conversions', 'Fetch', 'Parse XML tags', 'Sorter', and 'Molecule Type Cast'.
- public server:** A pane on the right side showing 'Workflow Server' information and a list of available workflows.
- node description:** A pane on the far right displaying the 'MarvinSketch' node's details, including its description and dialog options.
- outline:** A small thumbnail of the current workflow located at the bottom left of the main editor area.
- console:** A pane at the bottom right showing the 'KNIME Console' with system messages and error logs.



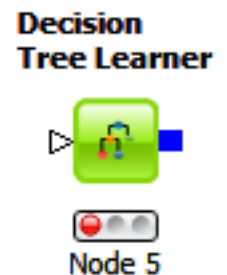
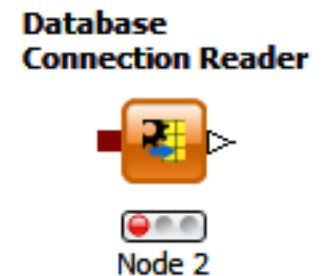
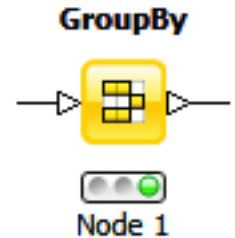
# KNIME nodes: Overview

Node = basic processing unit of KNIME workflow which performs a particular task



# Ports

- **Data Port:** a white triangle which transfers flat data tables from node to node
- **Database Port:** Nodes executing commands inside a database are recognized by their database ports (brown square)
- **PMML Ports:** Data Mining nodes learn a model which is passed to the referring predictor node via a blue squared PMML port



# Other Ports

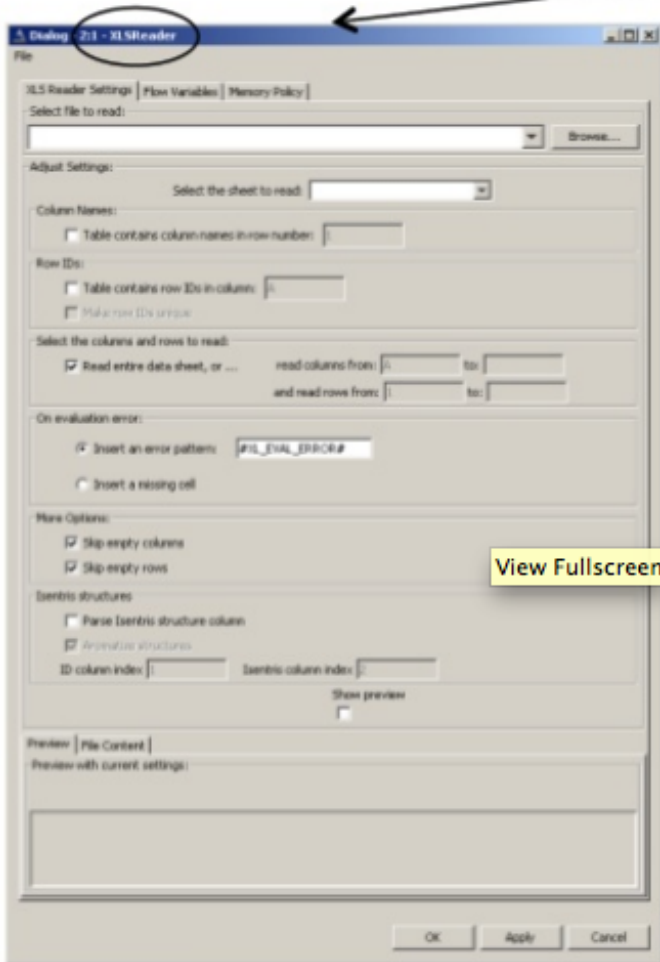
- Whenever a node provides data that does not fit a flat data table structure, **a general purpose port for structured data** is used (dark cyan square).
- All ports not listed above are known as **"unknown" types** (gray square).



# KNIME nodes: Dialogs

Double click to configure...

Configuration menus for selected nodes



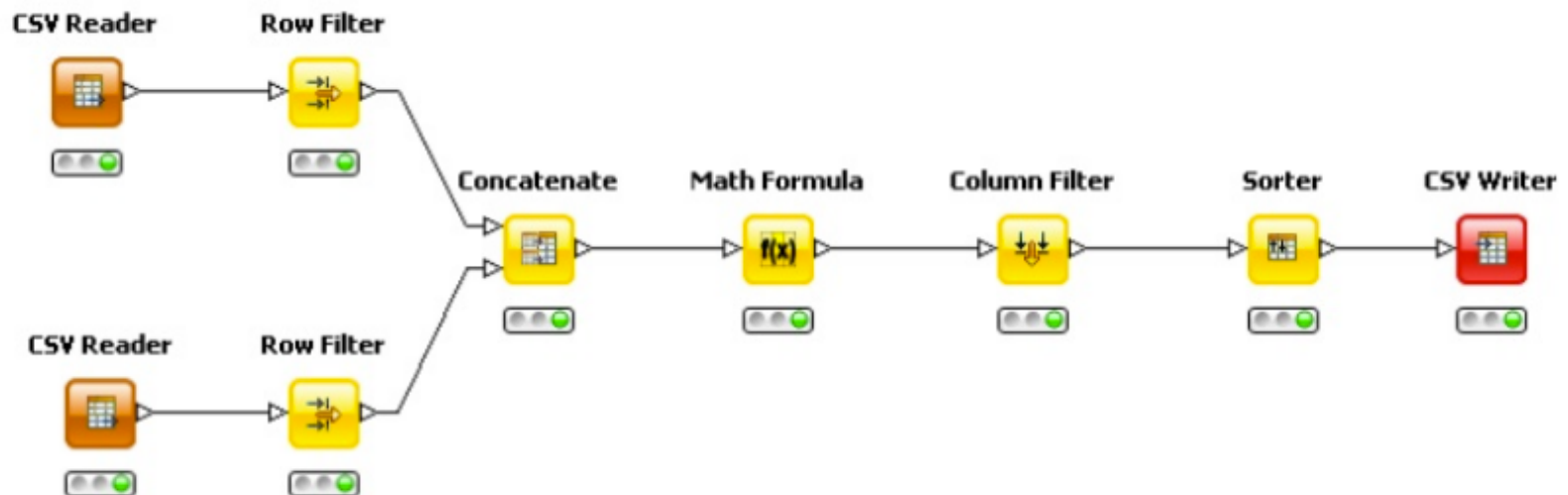
View Fullscreen

Explicit column type

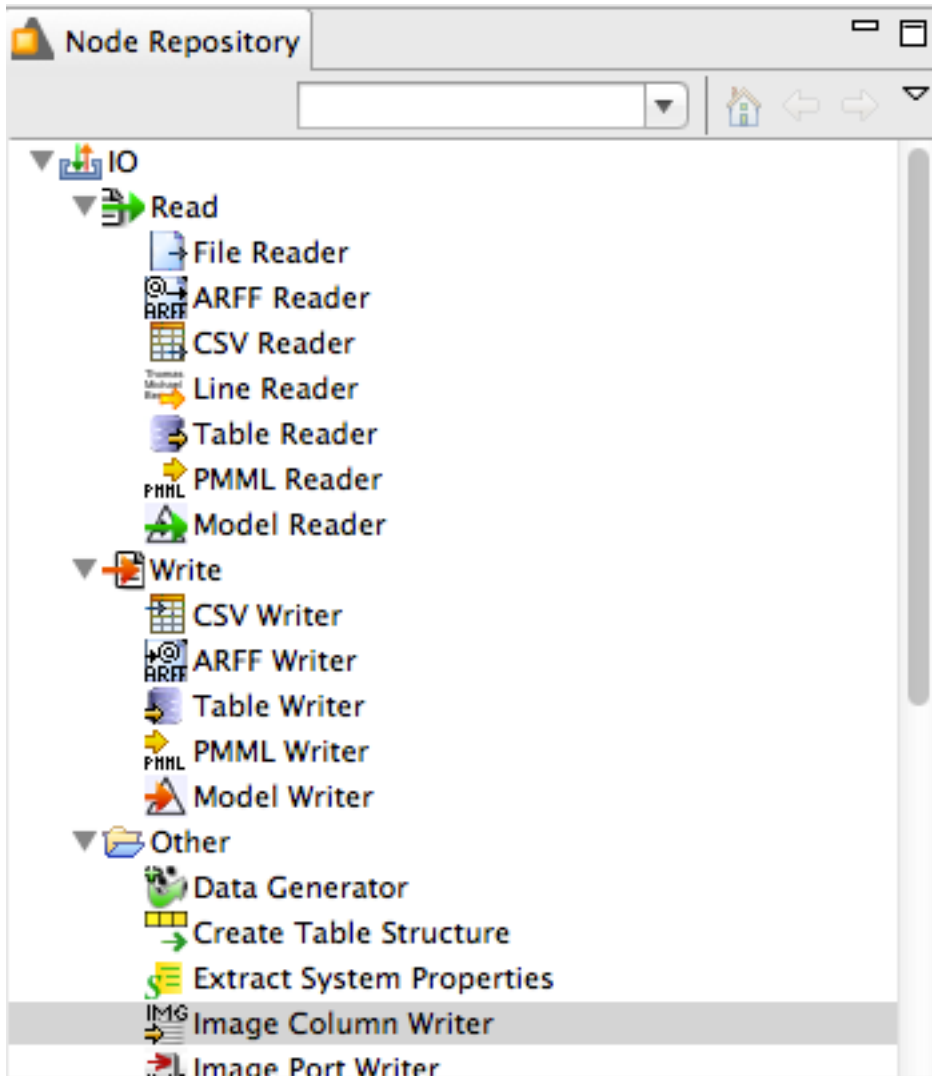


# An example of workflow

- Workflows can be imported and exported as .zip files
  - With or without the underlying data
  - File → Import KNIME workflow...
  - File → Export KNIME workflow...



# I/O Operations



**ARFF** (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

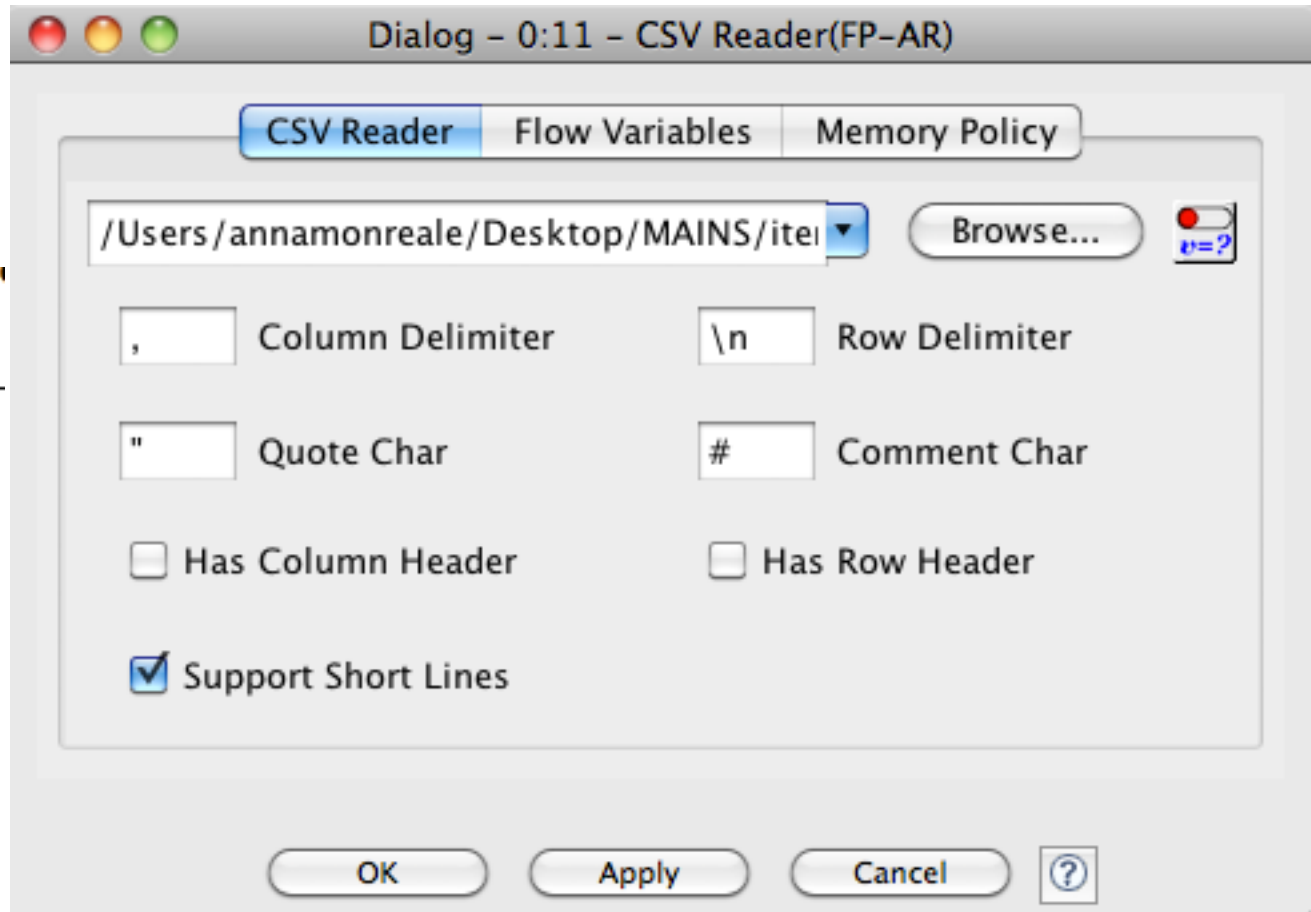
**CSV** (Comma-Separated Values) file stores tabular data (numbers and text) in plain-text form.

# CSV Reader

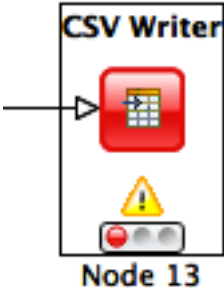
CSV Reader



FP-AR



# CSV Writer



Dialog - 0:13 - CSV Writer

Settings    Advanced    Quotes    Comment Header

Output file location:

Writer options:

- Write column header
- Don't write column headers if file exists
- Write row ID
- Compress output file (gzip)

If file exists...

Overwrite     Append     Abort

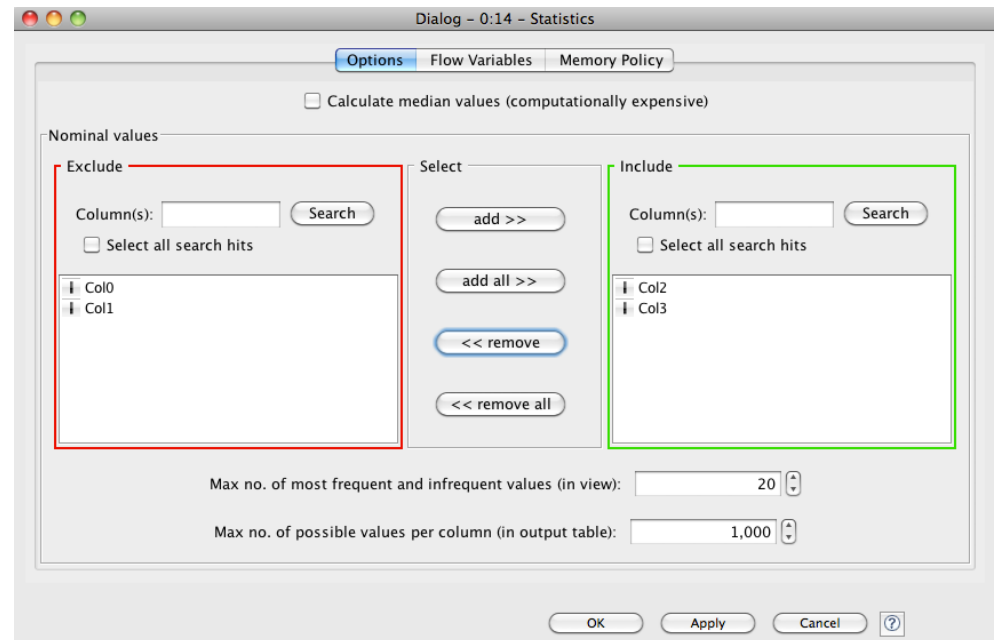
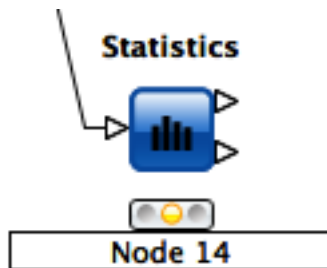


# Data Manipulation

- Three main sections
  - **Columns:** binning, replace, filters, normalizer, missing values, ...
  - **Rows:** filtering, sampling, partitioning, ...
  - **Matrix:** Transpose

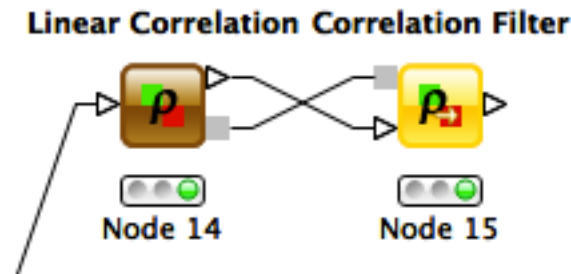
# Statistics node

- For all numeric columns computes statistics such as
- **minimum, maximum, mean, standard deviation, variance, median, overall sum, number of missing values and row counts**
- For all nominal values counts them together with their occurrences.



# Correlation Analysis

- **Linear Correlation node** computes for each pair of selected columns a correlation coefficient, i.e. a measure of the correlation of the two variables
  - Pearson Correlation Coefficient
- **Correlation Filtering node** uses the model as generated by a Correlation node to determine which columns are redundant (i.e. correlated) and filters them out.
  - **The output table will contain the reduced set of columns.**



# Data Views

- Box Plots
- Histograms, Pie Charts, Scatter plots, ...
- Scatter Matrix

# Mining Algorithms

- Clustering
  - Hierarchical
  - K-means
  - Fuzzy  $c$ -Means
- Decision Tree
- Item sets / Association Rules
  - Borgelt's Algorithms
- Weka

# EXERCISES

---

Anna Monreale

KDD-Lab, University of Pisa

Email: [annam@di.unipi.it](mailto:annam@di.unipi.it)

# MARKET BASKET ANALYSIS

---

Anna Monreale

KDD-Lab, University of Pisa

Email: [annam@di.unipi.it](mailto:annam@di.unipi.it)

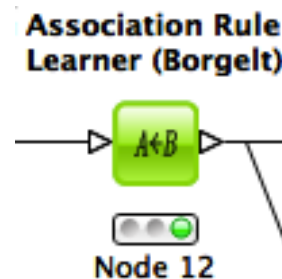
# Market Basket Analysis

- **Problem:** given a database of transactions of customers of a supermarket, find **the set of frequent items co-purchased** and analyze the **association rules** that is possible to derive from the frequent patterns.



# Frequent Patterns and AR in KNIME

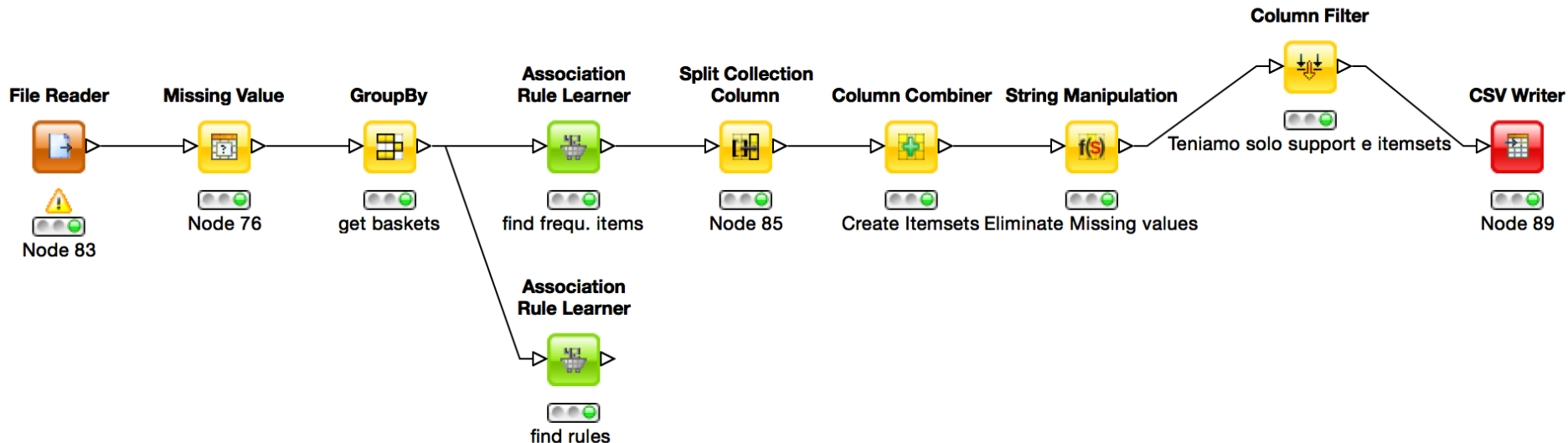
- Use the nodes implementing the Borgelt's Algorithms:



- It allows the extraction of both association rules and patterns
- AR Learner uses Apriori Algorithm

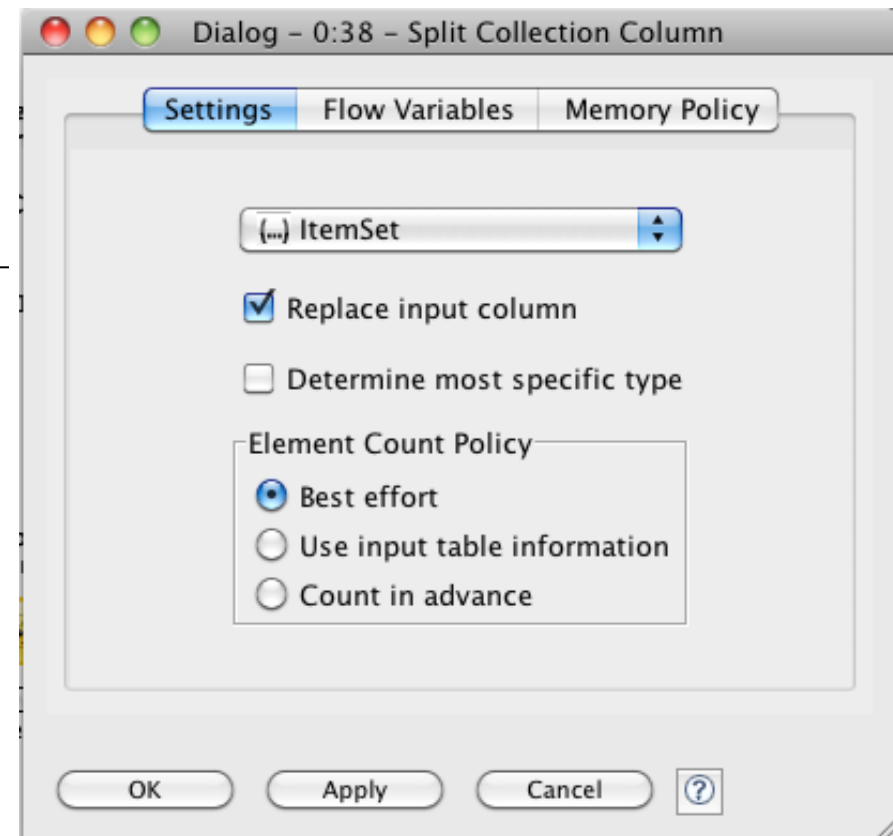
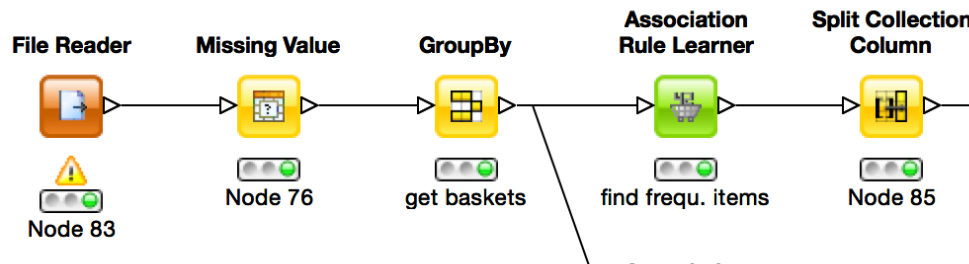
# Write the itemsets in a file

- Given the output of the Item set Finder node sometimes you cannot see all the components of the itemset
  - we need to transform it in a string and
  - then, we can also write the result in a file



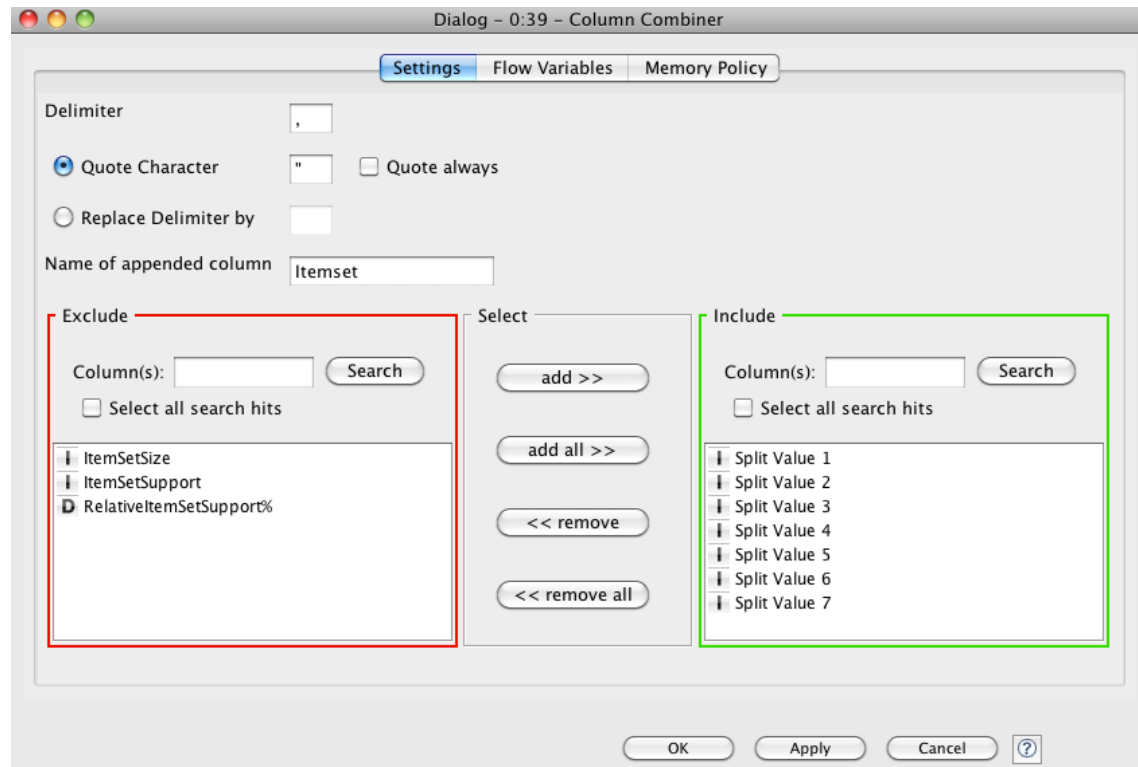
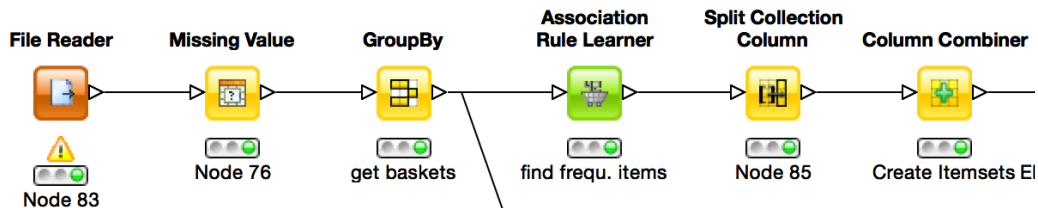
# Write the itemsets in a file

- First we need to split the collection



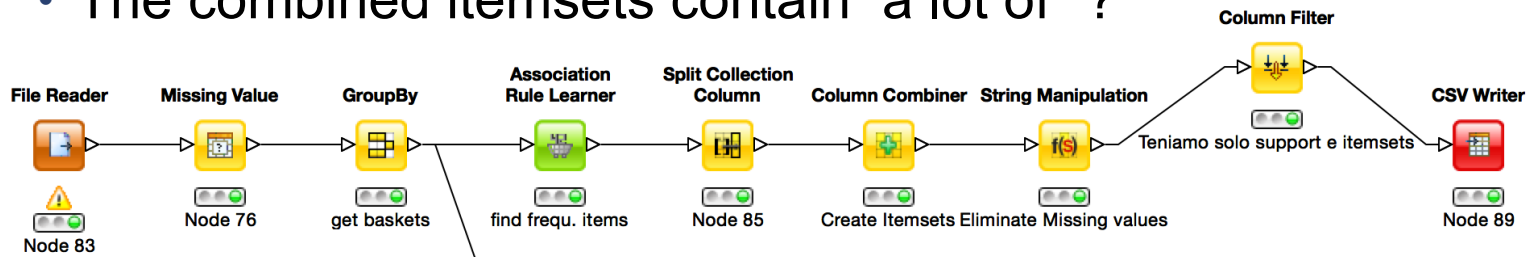
# Write the itemsets in a file

- Second we combine the columns that have to compose the itemset (string)

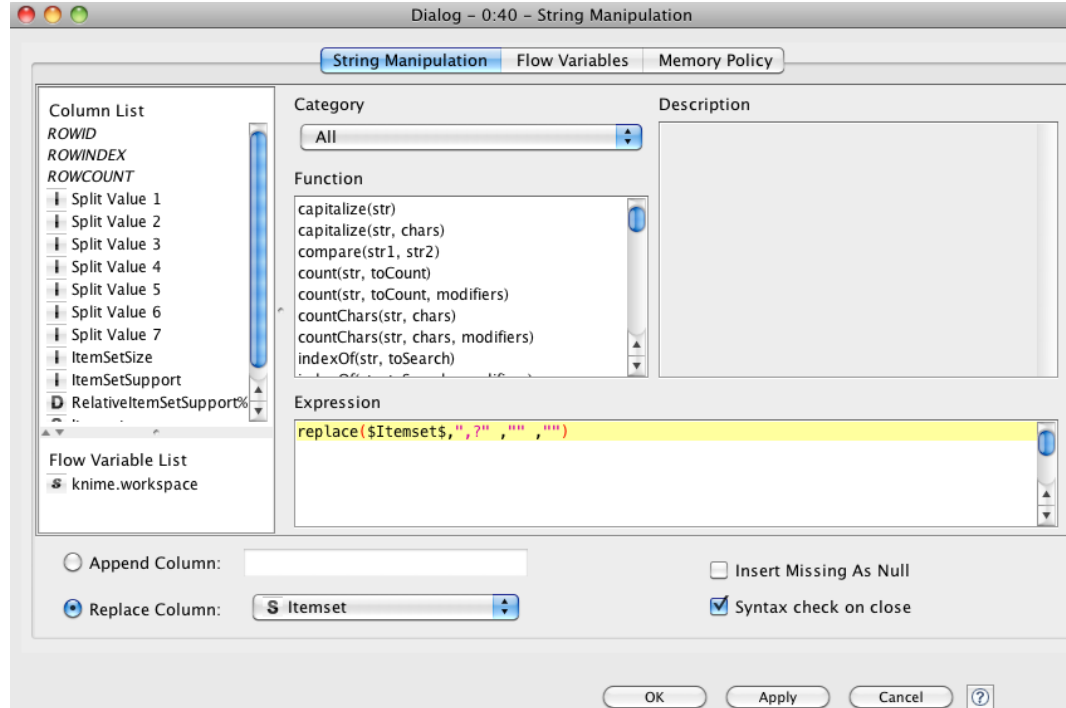


# Write the itemsets in a file

- The combiner does not eliminate the missing values “?”
- The combined itemsets contain a lot of “?”

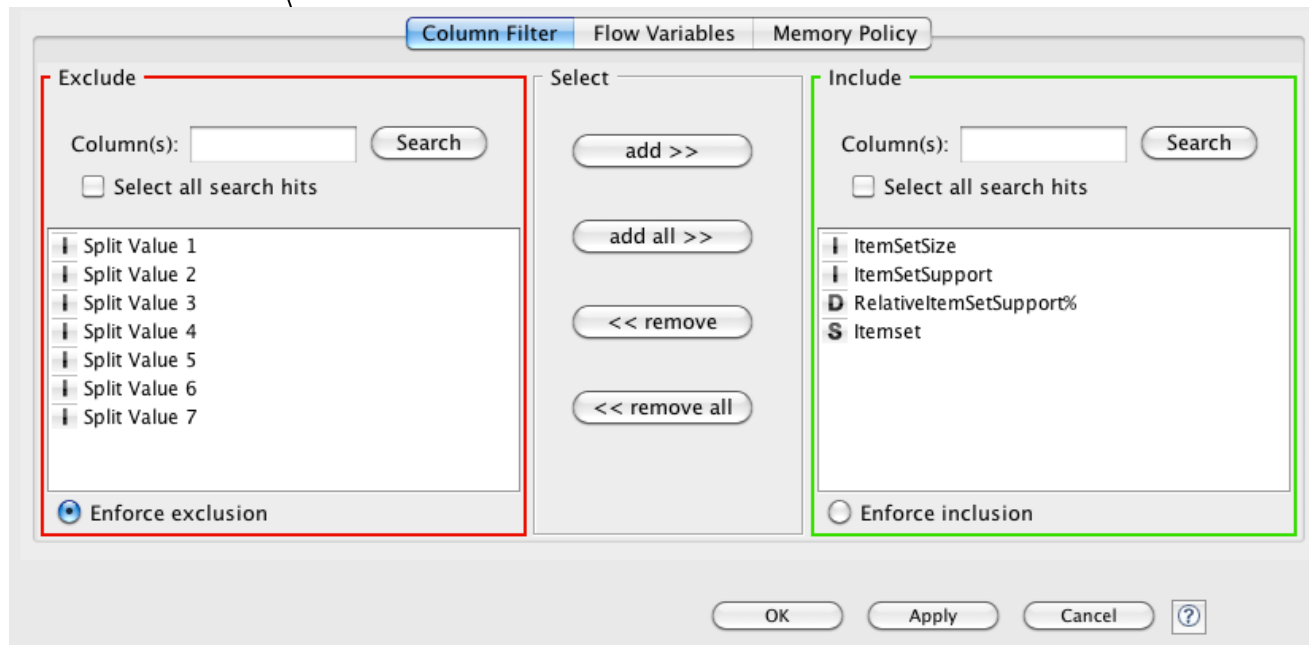
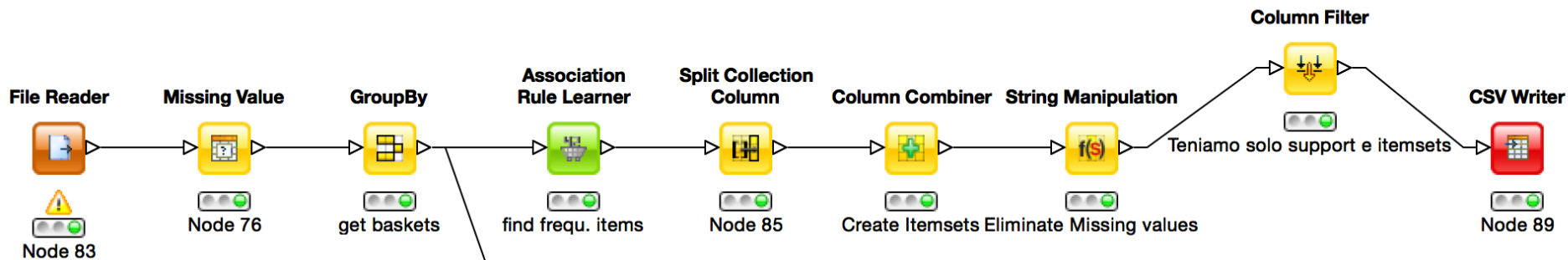


- We use the **replace** operation to eliminate them (String Manipulation)

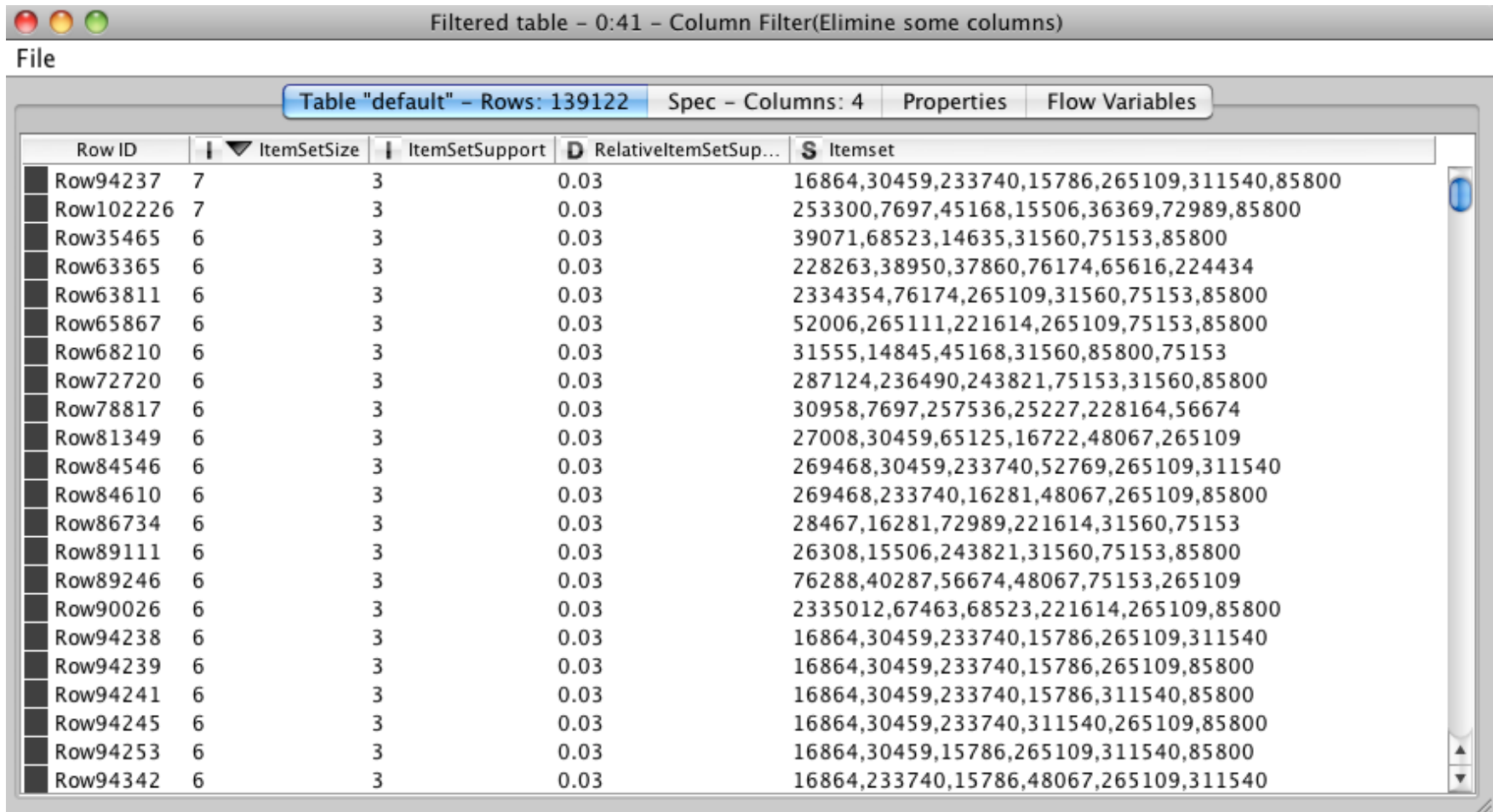


# Write the itemsets in a file

- Before writing in a file eliminate the split columns (Column filter)



# ..... The output table



Filtered table - 0:41 - Column Filter(Elimine some columns)

File

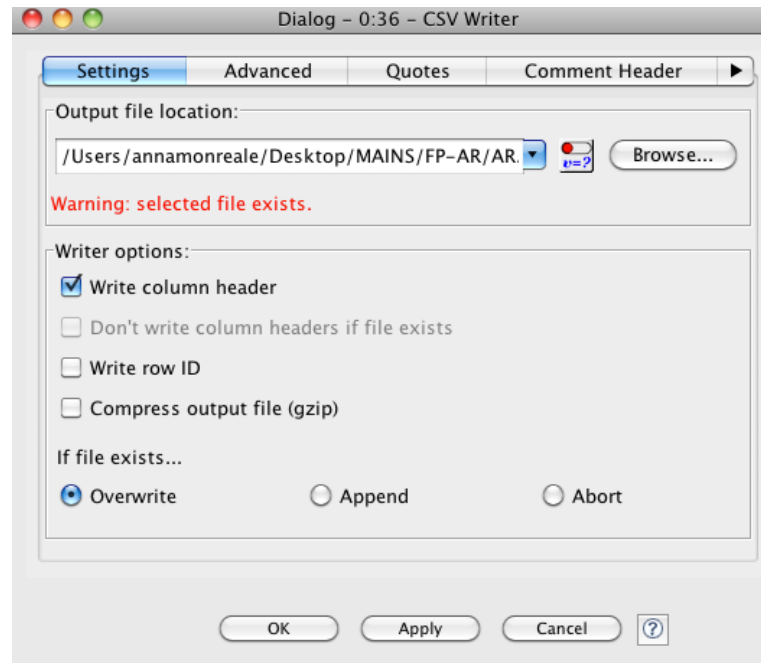
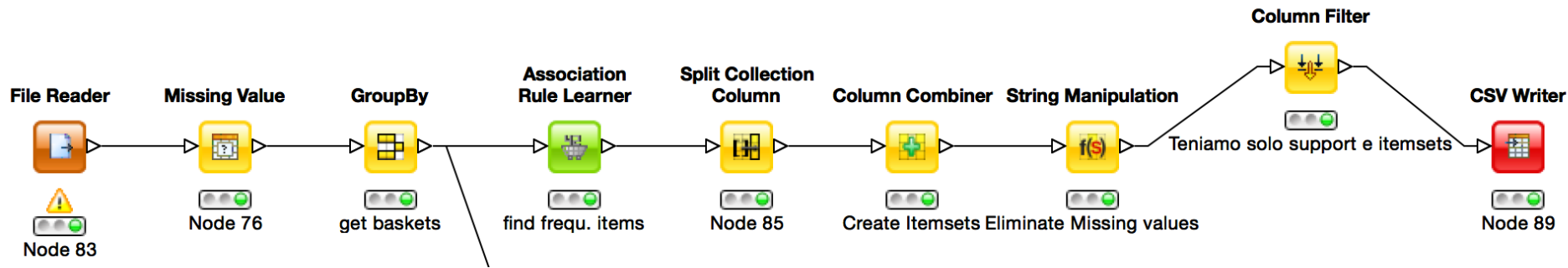
Table "default" - Rows: 139122 Spec - Columns: 4 Properties Flow Variables

Row ID	ItemSetSize	ItemSetSupport	RelativeItemSetSup...	Itemset
Row94237	7	3	0.03	16864,30459,233740,15786,265109,311540,85800
Row102226	7	3	0.03	253300,7697,45168,15506,36369,72989,85800
Row35465	6	3	0.03	39071,68523,14635,31560,75153,85800
Row63365	6	3	0.03	228263,38950,37860,76174,65616,224434
Row63811	6	3	0.03	2334354,76174,265109,31560,75153,85800
Row65867	6	3	0.03	52006,265111,221614,265109,75153,85800
Row68210	6	3	0.03	31555,14845,45168,31560,85800,75153
Row72720	6	3	0.03	287124,236490,243821,75153,31560,85800
Row78817	6	3	0.03	30958,7697,257536,25227,228164,56674
Row81349	6	3	0.03	27008,30459,65125,16722,48067,265109
Row84546	6	3	0.03	269468,30459,233740,52769,265109,311540
Row84610	6	3	0.03	269468,233740,16281,48067,265109,85800
Row86734	6	3	0.03	28467,16281,72989,221614,31560,75153
Row89111	6	3	0.03	26308,15506,243821,31560,75153,85800
Row89246	6	3	0.03	76288,40287,56674,48067,75153,265109
Row90026	6	3	0.03	2335012,67463,68523,221614,265109,85800
Row94238	6	3	0.03	16864,30459,233740,15786,265109,311540
Row94239	6	3	0.03	16864,30459,233740,15786,265109,85800
Row94241	6	3	0.03	16864,30459,233740,15786,311540,85800
Row94245	6	3	0.03	16864,30459,233740,311540,265109,85800
Row94253	6	3	0.03	16864,30459,15786,265109,311540,85800
Row94342	6	3	0.03	16864,233740,15786,48067,265109,311540

- **Now you can see all the items in a set!!!**

# Write the itemsets in a file

- Now we can complete the workflow with the **CSV Writer**





# CUSTOMER SEGMENTATION

---

Anna Monreale

KDD-Lab, University of Pisa

Email: [annam@di.unipi.it](mailto:annam@di.unipi.it)

# Customer Segmentation

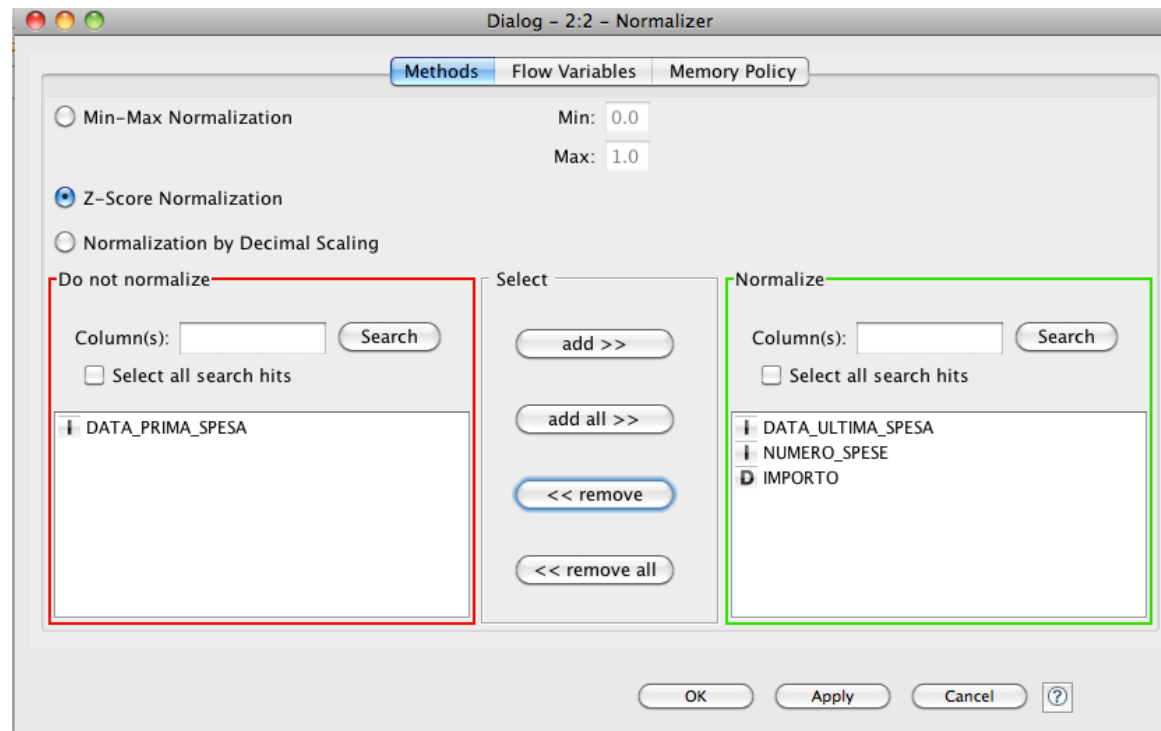
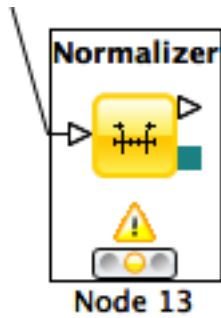
- **Problem:** given the dataset of RFM (Recency, Frequency and Monetary value) measurements of a set of customers of a supermarket, find a high-quality clustering using K-means and discuss the profile of each found cluster (in terms of the purchasing behavior of the customers of each cluster).
- Applying also the Hierarchical clustering and compare the results
- Provide a short document (max three pages in pdf, excluding figures/plots) which illustrates the input dataset, the adopted clustering methodology and the cluster interpretation.

# DATA

- **Dataset filename:** rfm\_data.csv.
- **Dataset legend:** for each customer, the dataset contains
  - *date\_first\_purchase*: integer that indicates the date of the first purchase of the customer
  - *date\_last\_purchase*: integer that indicates the date of the last purchase of the customer
  - *Number of purchases*: number of different purchases in terms of receipts
  - *Amount*: total money spent by the customer
- **Need to compute the columns for**
  - *Recency*: no. of days since last purchase
  - *Frequency*: no. of visits (shopping in the supermarket) in the observation period
  - *Monetary value*: total amount spent in purchases during the observation period.

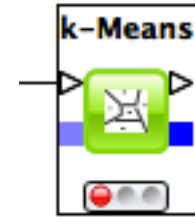
# Clustering in KNIME

- Data normalization
  - Min-max normalization
  - Z-score normalization
- Compare the clustering results before and after this operation and discuss the comparison

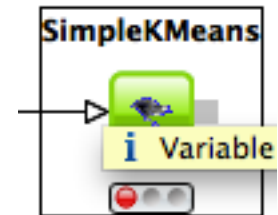


# K-Means

- Two options
  - K-means in Mining section of Knime
  - K-means in Weka section of Knime



Node 14

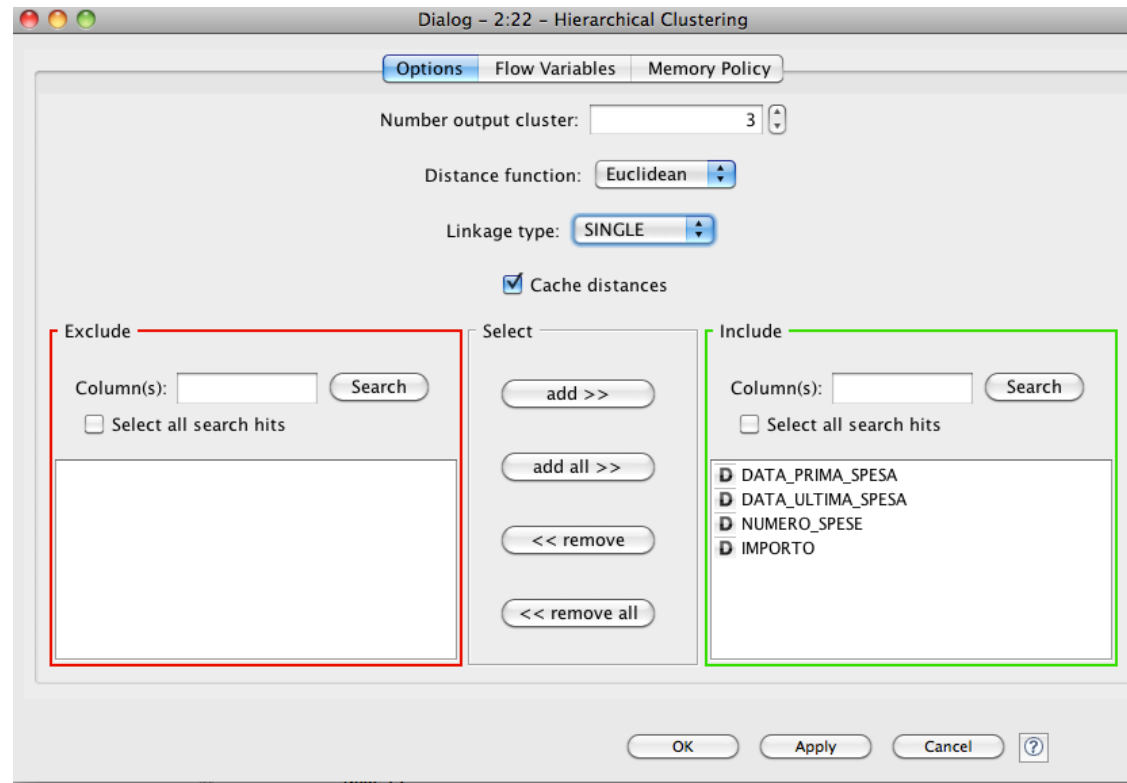


Node 15

- The second one allows the SSE computation useful for finding the best k value

# Hierarchical Clustering

- The node is in Mining section
- Allow to generate the dendrogram
- Various Distances



# CHURN ANALYSIS

---

Anna Monreale

KDD-Lab, University of Pisa

Email: [annam@di.unipi.it](mailto:annam@di.unipi.it)

# Churn Analysis

- **Problem:** Problem: given a dataset of measurements over a set of customers of an e-commerce site, find a high-quality classifier, using decision trees, which predicts whether each customer will place only one or more orders to the shop.
- Provide a short document (max three pages in pdf, excluding figures/plots) which illustrates the input dataset, the adopted clustering methodology and the cluster interpretation.

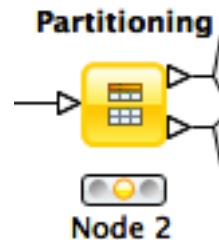


# Data

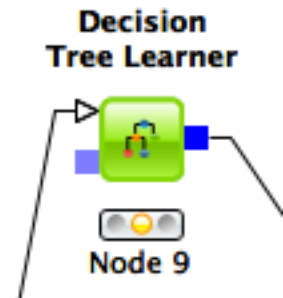
- Filename: *OneShotCustomersEX.csv*
  - Contains transactions from 15,000 online customers
- In the web page of the course you can download the attribute description
- The class of the data is **Customer Typology** that can be
  - **one shot** = only 1 purchase
  - **loyal** = more than one purchase

# Decision Trees in Knime

- For Classification by decision trees
  - Partitioning of the data in training and test set



- On the training set applying the learner



- On the test set applying the predictor

