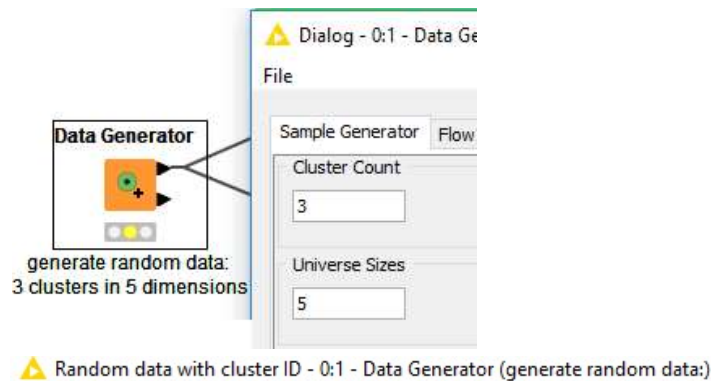


MSc Data Science and Advanced Computing CSMDM21 - Data Analytics and Mining

Practical 5: Advanced KNIME

1. Simple use of flow var with row filter

- Generate some (e.g., 300) data points randomly with the node *Data Generator*: 3 clusters in 5 dimensions. The output data table will also contain the attribute "ClusterMembership" in the last column.



File Edit Hilite Navigation View

Table "default" - Rows: 300 Spec - Columns: 6 Properties Flow Variables

Row ID	D Univers...	D Univers...	D Univers...	D Univers...	D Univers...	S Cluster ...
Row0	0.69	0.748	0.679	0.865	0.596	Cluster_0
Row1	0.807	0.637	0.57	0.921	0.361	Cluster_0
Row2	0.894	0.643	0.605	0.861	0.599	Cluster_0
Row3	0.864	0.422	0.504	0.761	0.467	Cluster_0
Row4	0.82	0.528	0.642	0.913	0.551	Cluster_0
Row5	0.774	0.608	0.677	0.93	0.417	Cluster_0

- Use the node *Table Creator* to input some value, e.g. "Cluster_1". You need to convert the value in the table into a flow variable by using the node *Table Column to Variable*.
- Use the node *Row Filter* to select only rows associated with a particular cluster, e.g. "Cluster_1". The flow variable can be used to control the parameter of the matching criteria in the configuration of the node *Row Filter*.
- Perform the same filtering task as above using the node *String Input* instead of the *Table Creator*. The node *String Input* provides an interactive view in which the user can enter the value (e.g., "Cluster_2").

2. Metanode var-k-Means with loop and flow variable

In this exercise we build a workflow to execute the k-Means algorithm many times, each run with a different value of k (e.g., from values in {2,3,4,5,10,15,20,50,150}).

- Load some data: you can try this exercise on the Iris dataset and on some artificially generated dataset with 5 or more clusters.

- Execute k-means in a loop with different values of k. The values for k could be provided by using the nodes *Table Creator* and *Table Row to Variable Loop Start*.
- Generate an output table reporting iteration number, the value k, the number of data points, some external indices of cluster validity (e.g., the entropy), like in the figure below.

Row ID	I Iteration	I numClusters	I Size	D Entropy	D Normalized Entropy	D Quality
Overall#0	0	3	500	0.8	0.345	0.655
Overall#1	1	5	500	0	0	1
Overall#2	2	7	500	0.4	0.172	0.828
Overall#3	3	10	500	0.4	0.172	0.828
Overall#4	4	20	500	0.028	0.012	0.988
Overall#5	5	50	500	0	0	1
Overall#6	6	150	500	0	0	1


- Collapse the workflow into a metanode (name it as "var-k-Means"). The metanode should take two input tables: the input data to be clustered and the list of values for the number of clusters (k).

3. Decision Tree accuracy estimation with the hold-out method

- Load the wine dataset and add column headers:
<http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>
- Convert the class attribute from numeric to String. This is necessary because the node *Decision Tree Learner* we are going to use requires string labels as class values.
- Use the node *Partitioning* to split (e.g. 90%-10%) the rows into training and test sets.
- Use the node *Decision Tree Learner* to generate a predictive model from the training set.
- Use the node *Decision Tree Predictor* to apply the predictive model to the test set.
- Use the node *Scorer* to compute the confusion matrix, the accuracy and the error rate.



4. Parameter optimisation for Decision Tree using the hold-out method

- Install the KNIME Optimization extension:

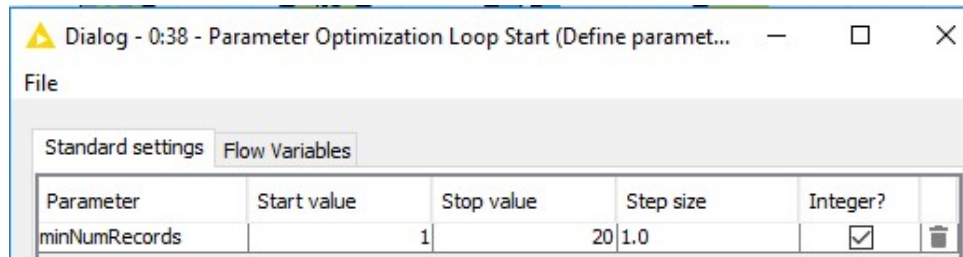
 Install

Available Software
 Check the items that you wish to install.

Work with:

Name	Version
<input checked="" type="checkbox"/>  KNIME & Extensions	
<input checked="" type="checkbox"/>  KNIME Optimization extension	4.4.0.v202106101426

- Duplicate the workflow of the previous exercise.
- Use the node *Parameter Optimization Loop Start* to define and inject a flow variable into the node *Partitioning*. Name the variable “minNumRecords”. Select a “Brute Force” strategy with the following value range.



- In the configuration dialog of the node *Decision Tree Learner*, go to the tab “Flow Variables”: for the parameter “minNumRecordsPerNode” select the flow variable “minNumRecords”.
- Use the node *Parameter Optimization Loop End* after the node *Scorer* to collect the accuracy values of each iteration: connect the node *Scorer* and the node *Parameter Optimization Loop End* with a variable (red) link. In the configuration dialog of the node *Parameter Optimization Loop End* select to maximise the accuracy.

The loop will iterate for all the desired values of the parameter and the node *Parameter Optimization Loop End* will return the best possible parameter to maximise the accuracy. Final challenge (no solution provided): can you use this optimal parameter value to train a final decision tree model? Do not set the parameter value manually: of course, you should use a flow variable to set the parameter in the *Decision Tree Learner* node to have a completely automatic workflow with parameter optimisation that produces a final model.

Solutions:

Sample solutions to these exercises are available on Blackboard (Bb) in two forms: images of the workflows and the actual KNIME workflows (as a single zip archive). You should first try to build your own workflows for each exercise. During the practical session, you may use the images to see the proposed solutions and to reconstruct them. During or at the end of the practical session, the archives with the actual KNIME workflows will be made available for you to import and test them. These are KNIME archives (file extension ". knwf ") that can be imported in KNIME. Note that after importing the KNIME archives, you may need to change the file locations of the source data file and the output file destination folder.