Classification

Exercise 5

Naïve Bayes Classification

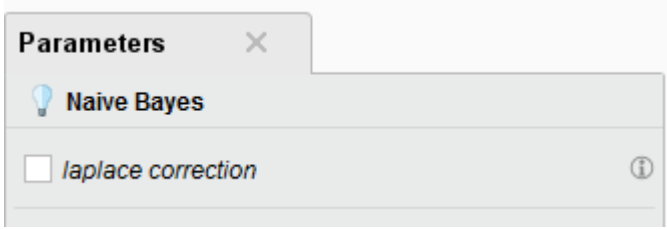
- If we know the prior probability and the likelihood
  - which we can estimate from the data
- Then we can calculate the posterior probability
  - Which we use for classification

$$P(C|A) = \frac{P(A|C)P(C)}{P(A)}$$

- Prior Probability
  - P(A), P(C) "35.7% chance of rain", "64.3% chance of play golf"
- Likelihood, given an observation
  - P(A|C) "33% chance of rain if go play golf"
- Posterior Probability
  - P(C|A) "66.7% chance of play golf of if no rain"

Operators: Naïve Bayes

- Input
  - Training data (Example Set)
- Output
  - Classification Model
  - Training data (Example Set)
- Parameters
  - Laplace Correction



- Distribution Table (in results) shows posterior probabilities

Attribute	Parameter	no	yes
Outlook	value=no rain	0.600	0.667
Outlook	value=rain	0.400	0.333
Outlook	value=unknown	0	0

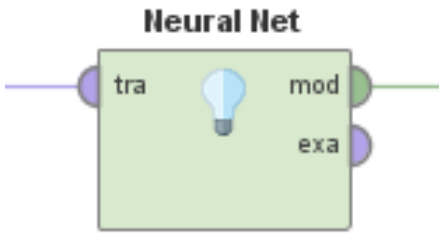
Naïve Bayes in Rapid Miner

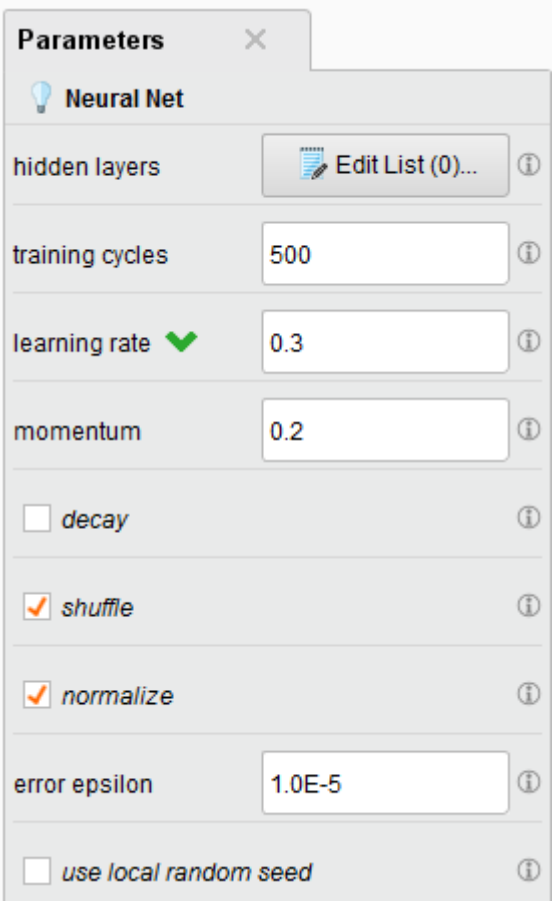
- Probabilities can be seen as confidences in the result

Ro...	Play	prediction(Play)	confidence(no)	confidence(yes)	Outlook
1	no	yes	0.333	0.667	no rain
2	no	yes	0.333	0.667	no rain
3	yes	yes	0.333	0.667	no rain
4	yes	yes	0.400	0.600	rain
5	yes	yes	0.400	0.600	rain
6	no	yes	0.400	0.600	rain
7	yes	yes	0.333	0.667	no rain
8	no	yes	0.333	0.667	no rain
9	yes	yes	0.333	0.667	no rain
10	yes	yes	0.400	0.600	rain
11	yes	yes	0.333	0.667	no rain
12	yes	yes	0.333	0.667	no rain
13	yes	yes	0.333	0.667	no rain
14	no	yes	0.400	0.600	rain

Operators: Neural Net

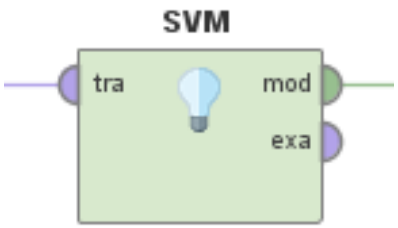
- Input Port
  - Training data (Example Set)
- Output Ports
  - Classification Model
  - Training data (Example Set)
- Parameters
  - Hidden layers (amount & sizes)
  - Training cycles
  - Learning rate
  - Momentum
  - …
- Requires numerical attributes

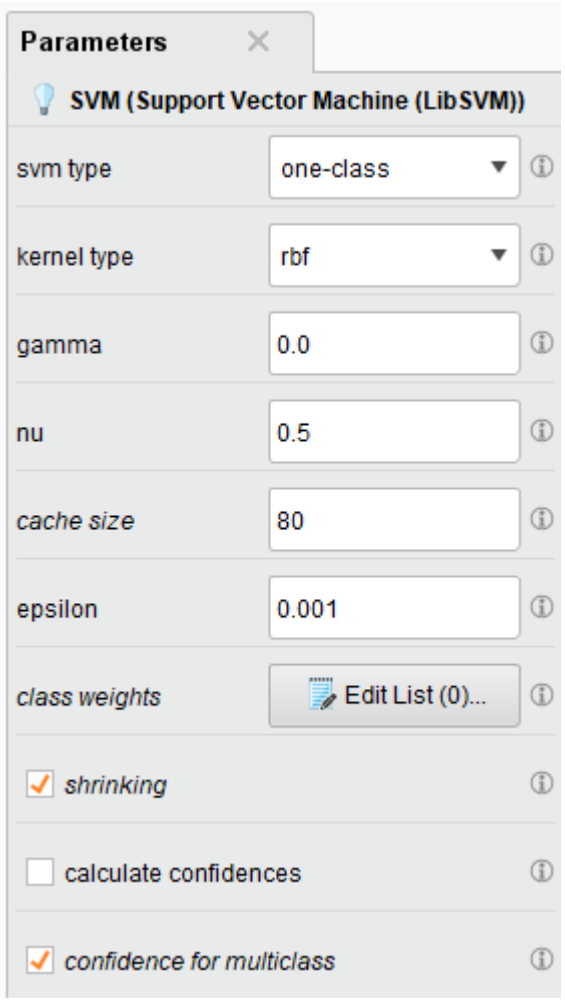




Operators: Support Vector Machine (LibSVM)

- Input Port
  - Training data (Example Set)
- Output Ports
  - Classification Model
  - Training data (Example Set)
- Parameters
  - SVM type
  - Kernel type
  - +more depending on SVM & Kernel type
- SVM Types
  - Last character indicates model type
  - …C – Classification …R - Regression
- Requires numerical attributes



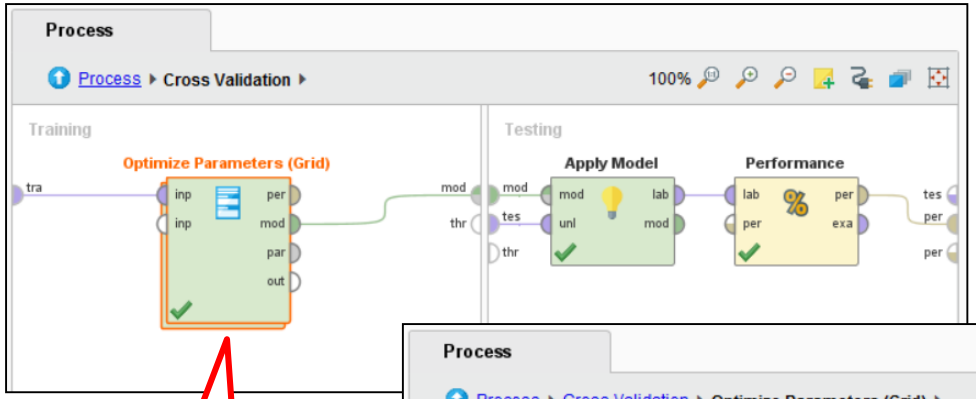


Parameter Tuning

- Most learning algorithms have parameters
  - Systematic testing of parameter values is called optimisation
- There are different strategies for optimisation
  - Here we use Grid Search
  - Given a set of parameter values or ranges, test all possible combinations
- Attention!
  - We learn the best parameter values from the data
  - So we must **evaluate on a different dataset!!!**
  - Again, overfitting can be a problem

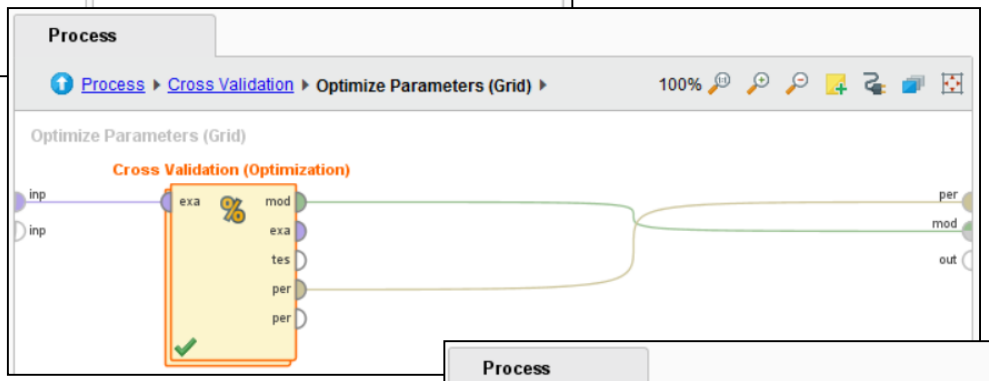
remember

Nested Cross-Validation for Parameter Optimization

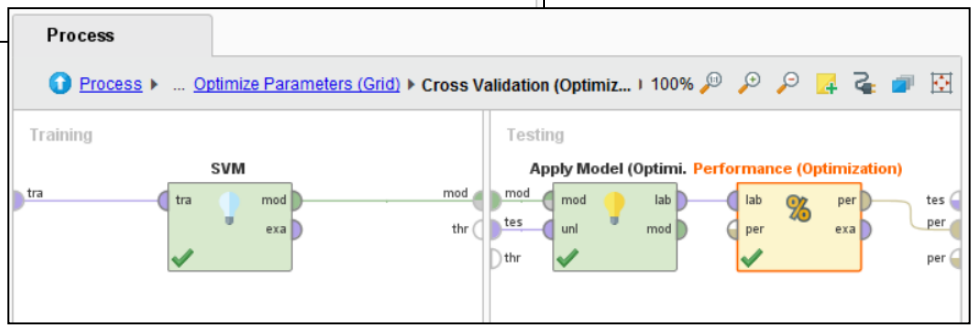


Outer Cross Validation

Optimize Parameters

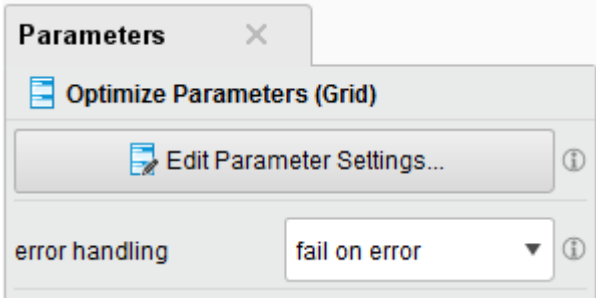


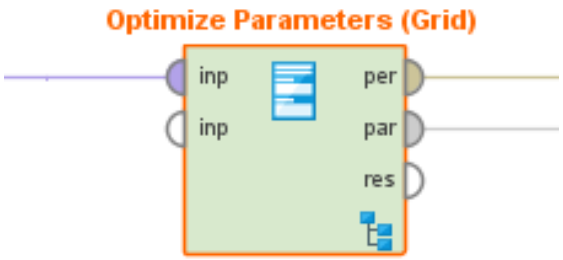
Inner Cross Validation

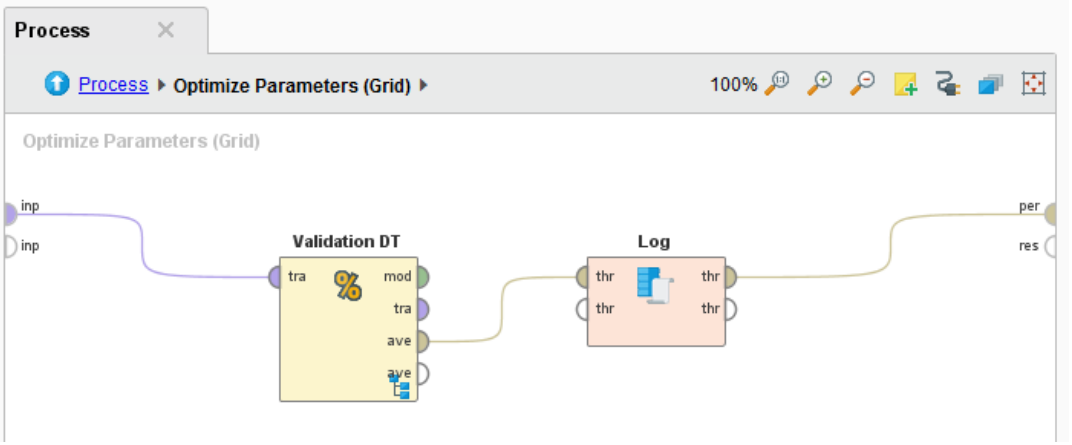


Operators: Optimize Parameters (Grid)

- Input Ports
  - Any input (whatever you need in the nested process)
- Output Ports
  - Performance Vector (for optimal parameters)
  - Parameter Set (optimal values)
  - Any additional results (from the nested process)
- Parameters
  - Parameter values to test

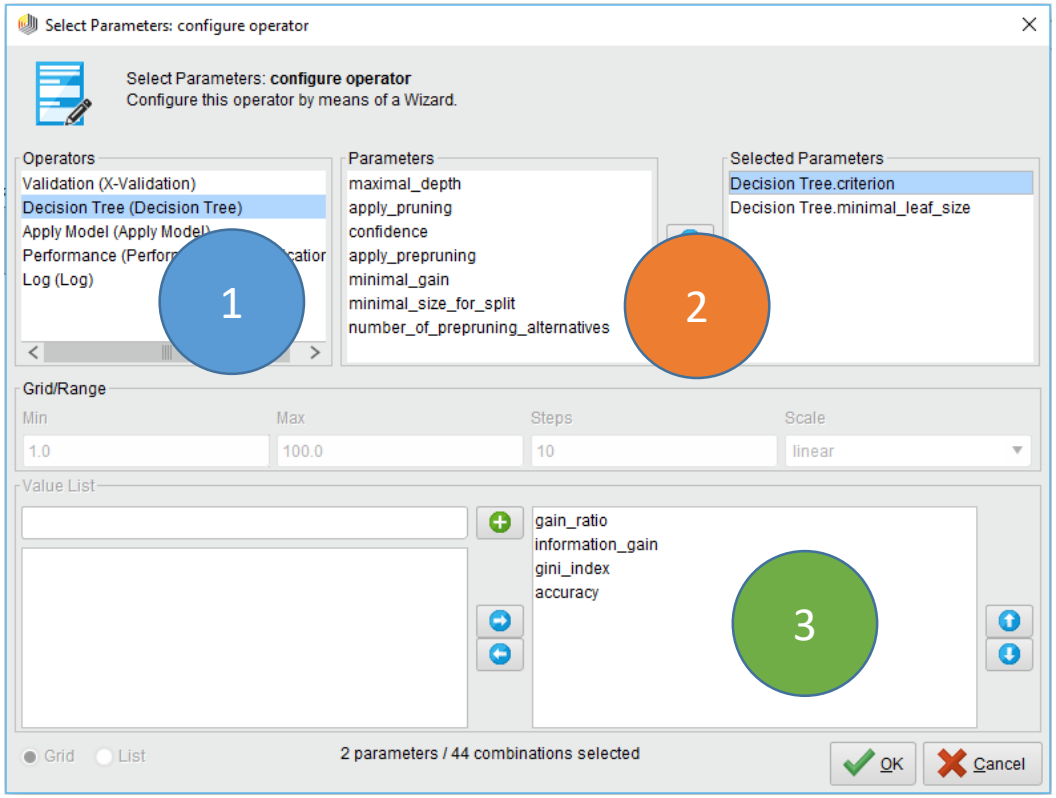






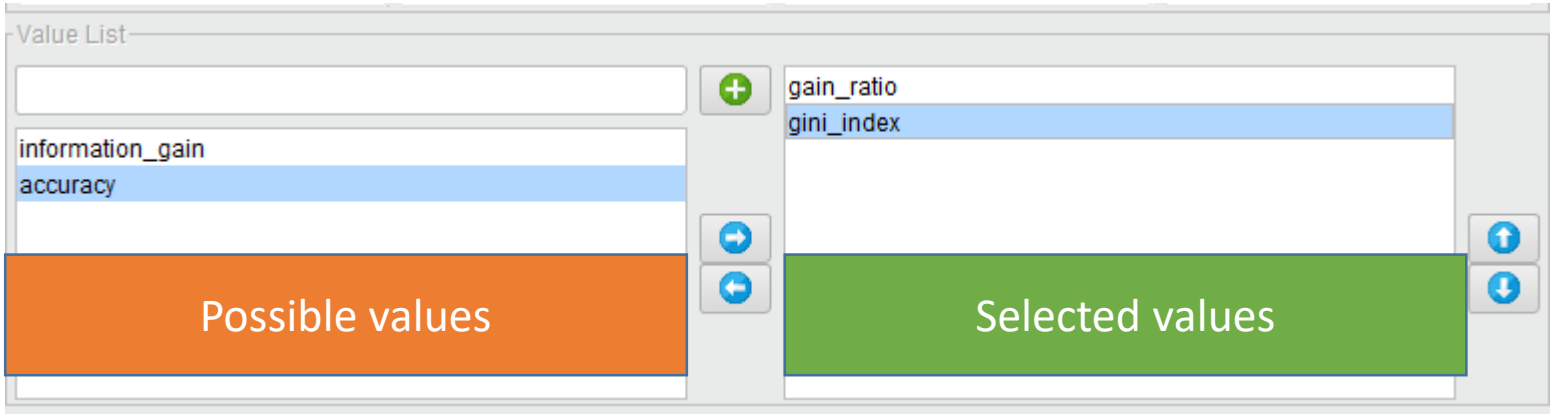
Operators: Optimize Parameters (Grid)

- Configuration steps:
  1. Select the operator you want to optimise
  2. Select the parameters of that operator
  3. Specify which values to test

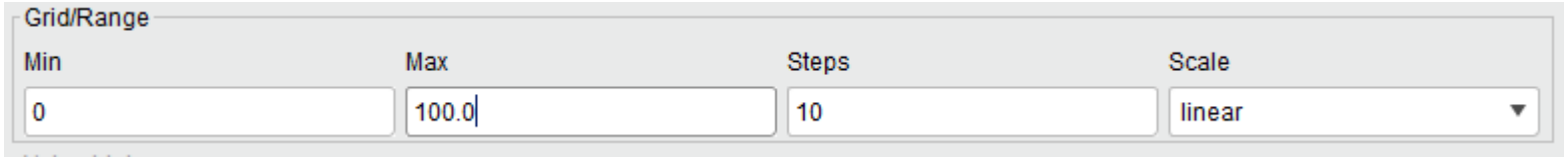


Operators: Optimize Parameters (Grid)

- Nominal parameter values
  - Select which values to use

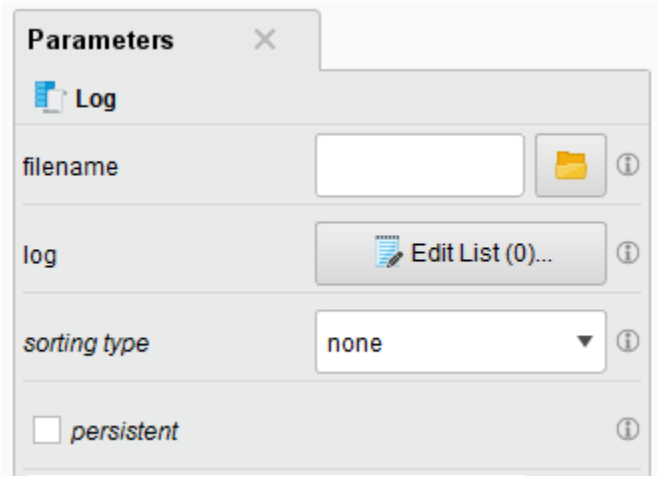


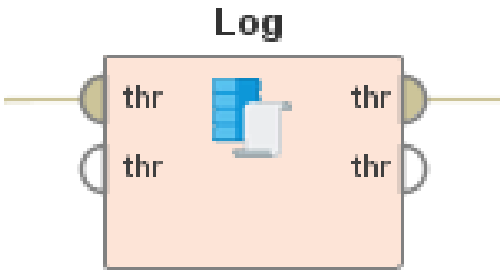
- Continuous parameter values
  - Specify values by steps
    - Linear: 0 / 10 / 20 / 30 / …
    - Quadratic: 0 / 1 / 4 / 9 / 16 / …
    - Logarithmic: 1 / 2 / 3 / 4 / 6 / 10 / 16 / …



Operators: Log

- Input Ports
  - Through
- Output Ports
  - Through (will simply forward what you connect to the input)
- Parameters
  - Filename
  - Log (what to log)

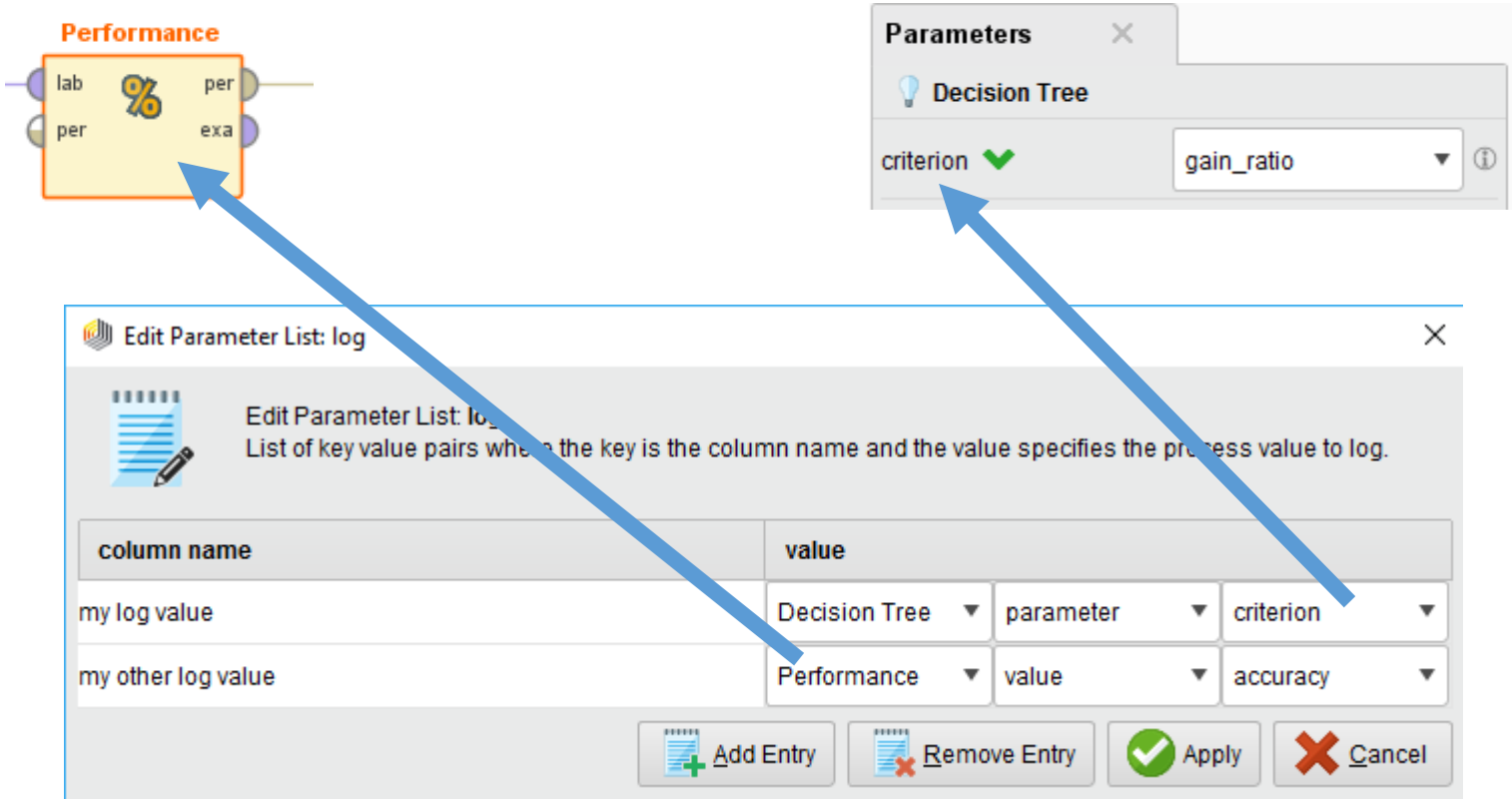




- Does not need any connections!
- But if connected, the order of execution is defined

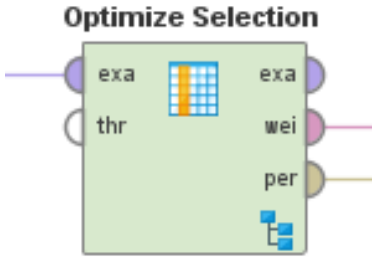
Operators: Log

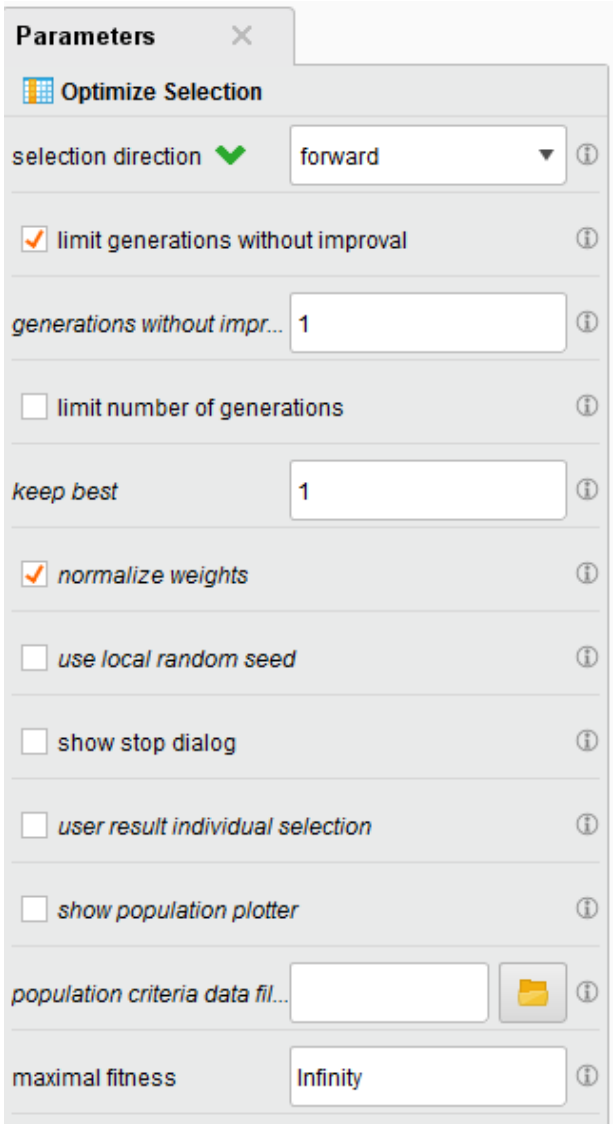
- You can log parameters and output values of operators
  - Parameters can be changed by the optimise operator
  - Values are the results of operators



Operators: Optimize Selection

- Input Ports
  - Training data (Example Set)
  - Any Input
- Output Ports
  - Training data (Example Set)
  - Attribute weights
  - Performance Vector
- Parameters
  - Direction (add or remove attributes during optimisation)
- Finds the optimal selection of attributes

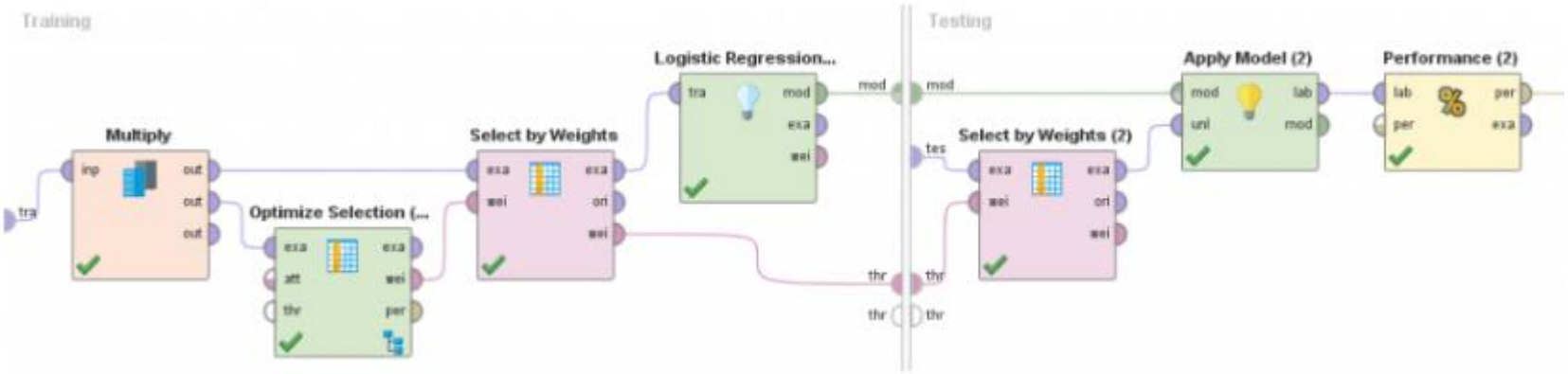




Optimize Selection with outer cross-validation

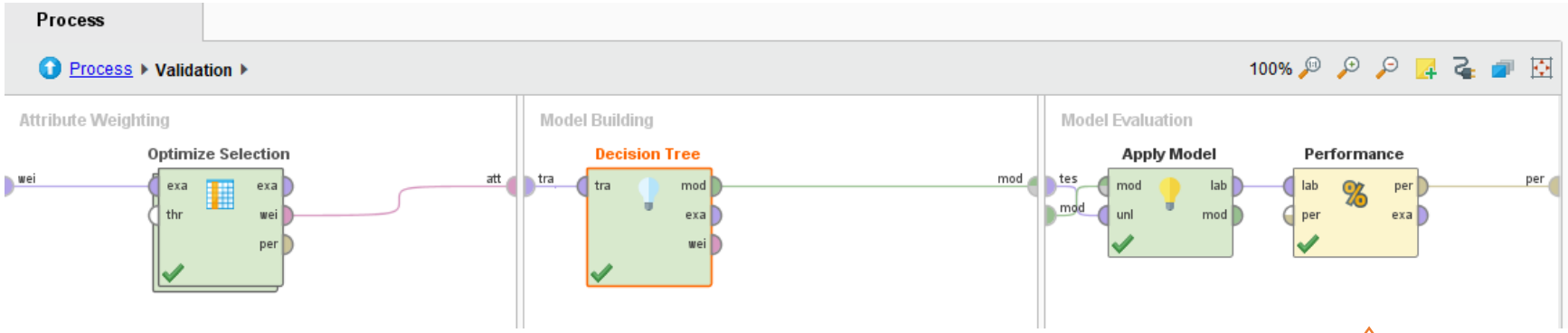
- Optimize Selection is a nested process
- In order to avoid „contaminating“ our results we would need to add an outer cross-validation (similarly to grid search)

Things become complicated...

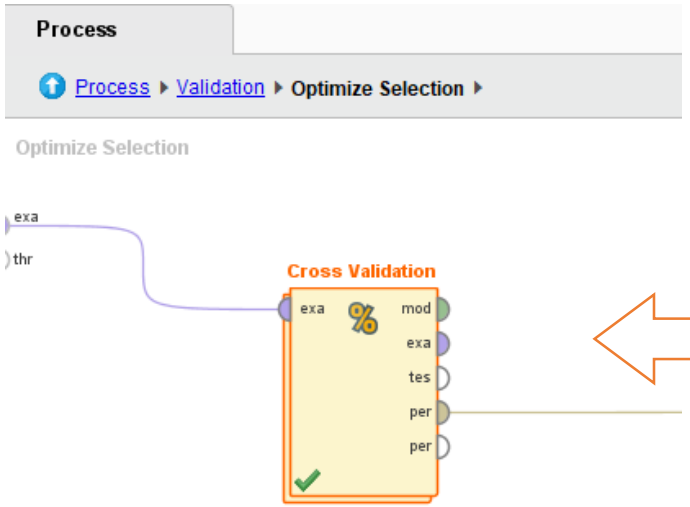


source: https://rapidminer.com/blog/learn-right-way-validate-models-part-4-accidental-contamination/ (accessed March 2020)

Optimize Selection with outer cross-validation



Use **Wrapper-X-Validation** which aligns the outer validation so that the test fold considers only the attributes selected by Optimize Selection operator.



Inner cross-validation for feature selection optimization

More Examples

- Rapidminer Tutorial on Accidental Contamination through Feature Selection and Parameter Optimization
  - https://rapidminer.com/blog/learn-right-way-validate-models-part-4-accidental-contamination/