University of Minho
School of Engineering

Machine Learning and Decision-Making

ADI @ LEI/3º, MiEI/4º - 2º Semestre

Filipe Gonçalves, Inês Alves, Cesar Analide

Part VI – March 2022

Contents

- The Learner-Predictor concept and Tree-based models
- (Loops)
- (Model Tuning)
- Hands On

A ML Model Goes to a Job Interview

Interviewer: What’s your biggest strength?
ML Candidate: I learn very well!

A ML Model Goes to a Job Interview

Interviewer: What’s your biggest strength?
ML Candidate: I learn very well!

Interviewer: Ok! So, what’s 20+15?
ML Candidate : It’s 5.

A ML Model Goes to a Job Interview

Interviewer: What’s your biggest strength?
ML Candidate: I learn very well!

Interviewer: Ok! So, what’s 20+15?
ML Candidate : It’s 5.

Interviewer: Not even close. It’s 35.
ML Candidate : It’s 20.

A ML Model Goes to a Job Interview

Interviewer: What’s your biggest strength?
ML Candidate: I learn very well!

Interviewer: Ok! So, what’s 20+15?
ML Candidate : It’s 5.

Interviewer: Not even close. It’s 35.
ML Candidate : It’s 20.

Interviewer: I said 35.
ML Candidate : 33.

A ML Model Goes to a Job Interview

Interviewer: What's your biggest strength?
ML Candidate: I learn very well!

Interviewer: Ok! So, what's 20+15?
ML Candidate : It's 5.

Interviewer: Not even close. It's 35.
ML Candidate : It's 20.

Interviewer: I said 35.
ML Candidate : 33.

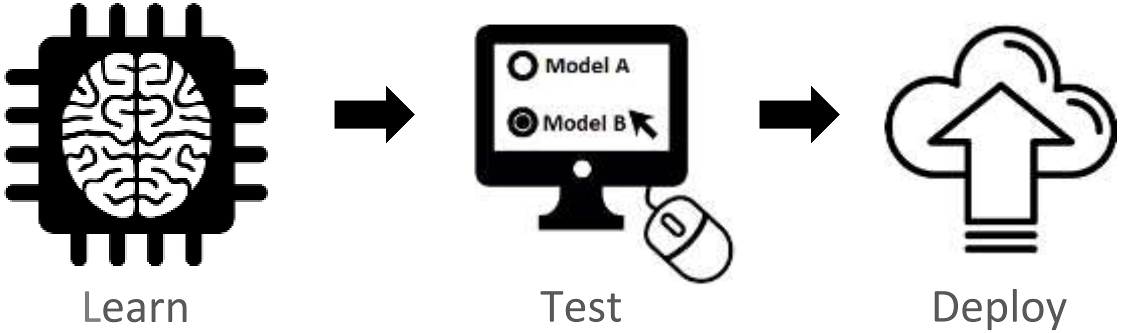
Interviewer: It's 35.
ML Candidate : It's 35.

Interviewer: Hired!

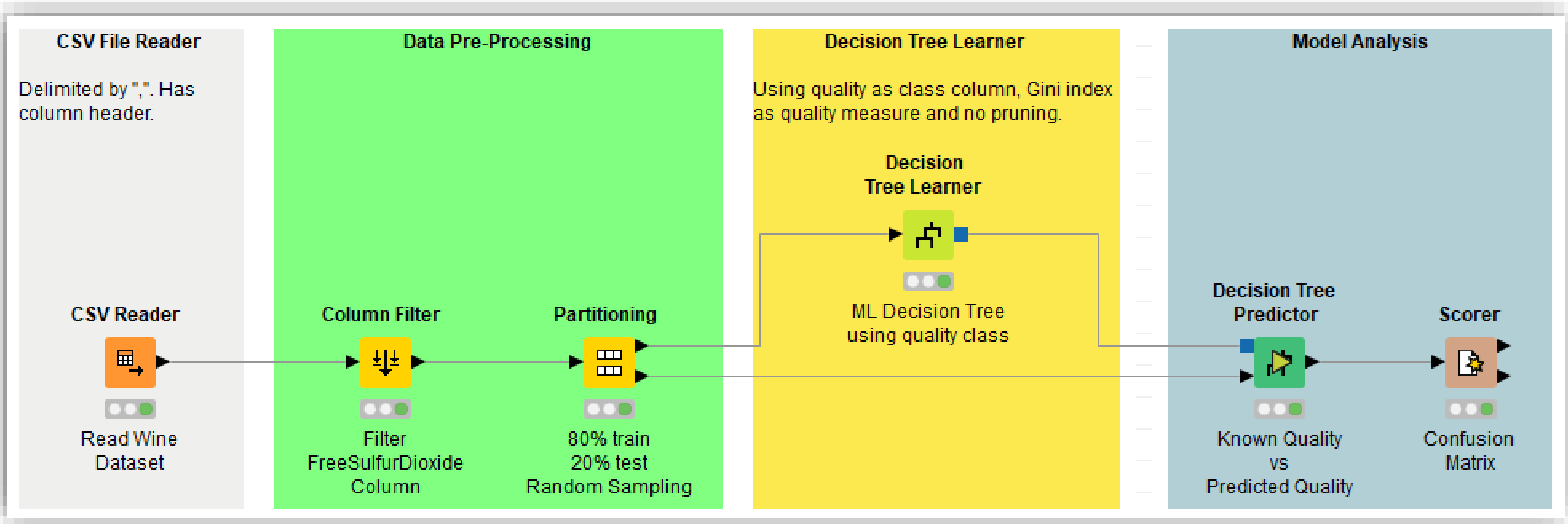
The Learner-Predictor Concept

Supervised algorithms imply a learning phase before applying the model to new data. But they also require a testing phase and a tuning phase!

In KNIME we implement supervised algorithms with Learner and Predictor nodes



Learner-Predictor



CSV File Reader

Delimited by ",". Has column header.

Data Pre-Processing

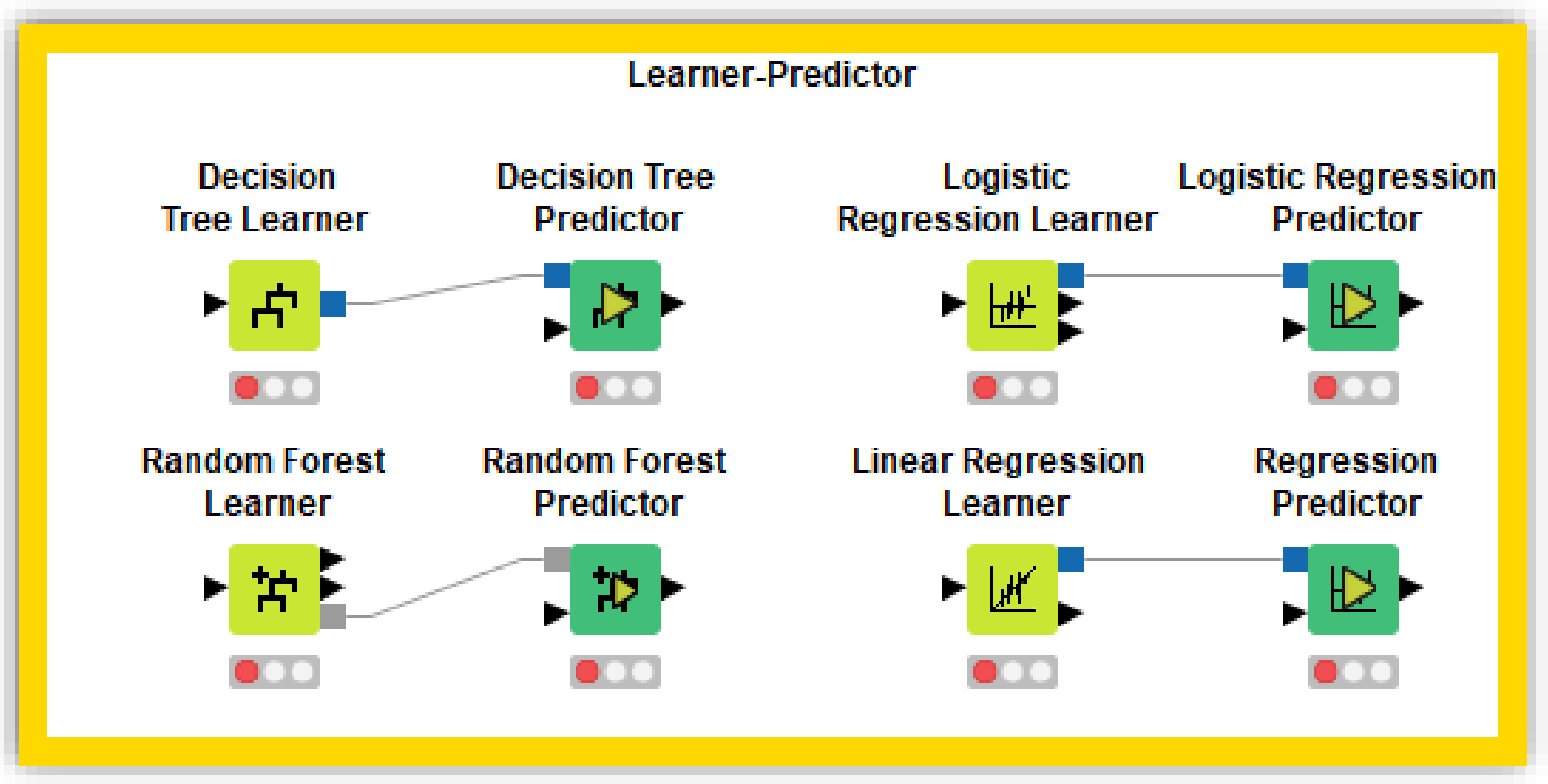
Decision Tree Learner

Using quality as class column, Gini index as quality measure and no pruning.

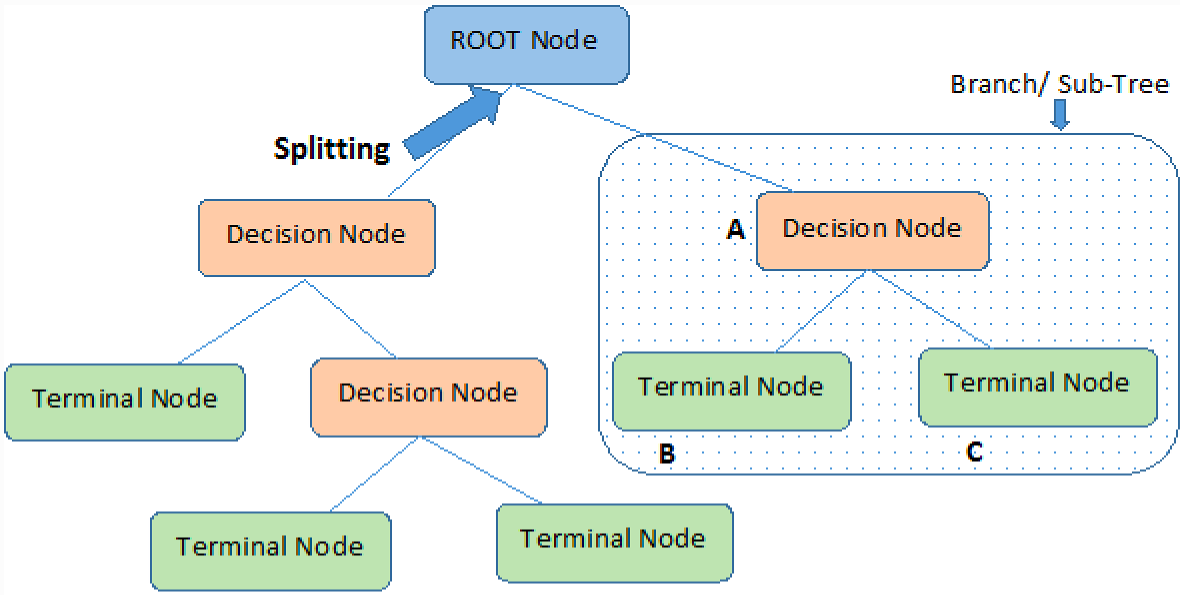
Model Analysis

CSV Reader
Read Wine Dataset
Column Filter
Filter FreeSulfurDioxide Column
Partitioning
80% train 20% test Random Sampling
Decision Tree Learner
ML Decision Tree using quality class
Decision Tree Predictor
Known Quality vs Predicted Quality
Scorer
Confusion Matrix

Some Learner-Predictor Nodes

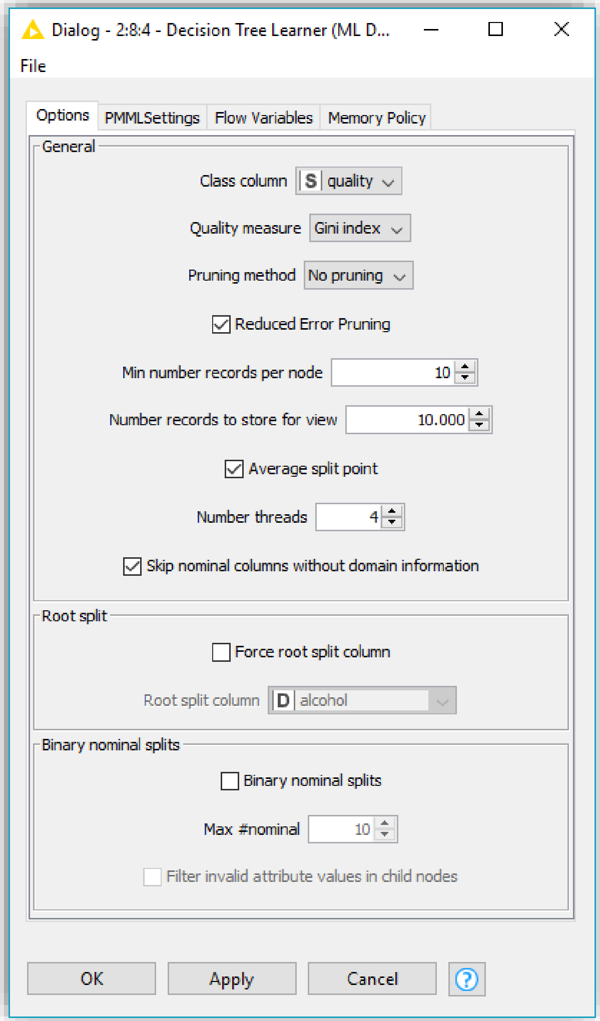


Decision Trees



Note:- A is parent node of B and C.

Decision Trees



Dialog - 2:8:4 - Decision Tree Learner (ML D...

General

Class column: S quality

Quality measure: Gini index

Pruning method: No pruning

☑ Reduced Error Pruning

Min number records per node: 10

Number records to store for view: 10.000

☑ Average split point

Number threads: 4

☑ Skip nominal columns without domain information

Root split

□ Force root split column

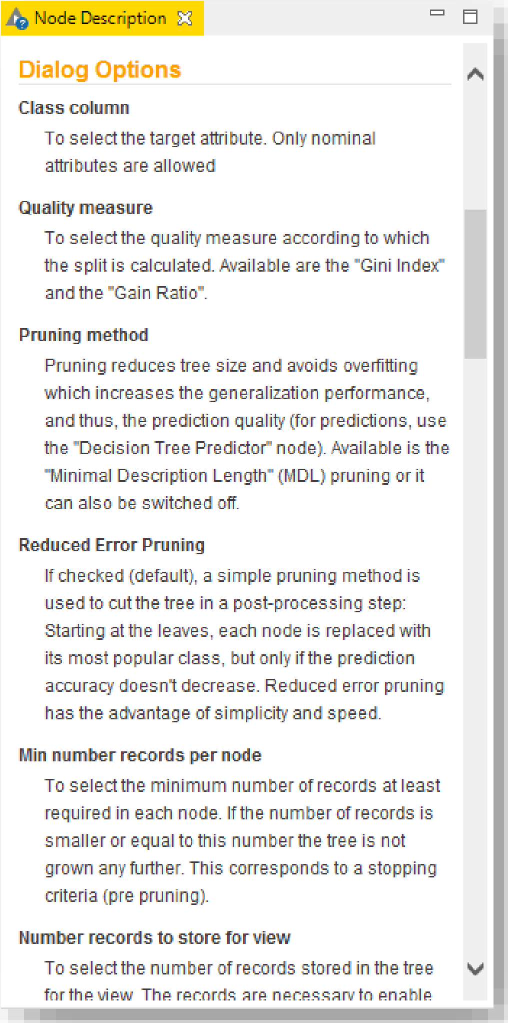
Root split column: D alcohol

Binary nominal splits

□ Binary nominal splits

Max #nominal: 10

□ Filter invalid attribute values in child nodes



Node Description

Dialog Options

Class column

To select the target attribute. Only nominal attributes are allowed

Quality measure

To select the quality measure according to which the split is calculated. Available are the "Gini Index" and the "Gain Ratio".

Pruning method

Pruning reduces tree size and avoids overfitting which increases the generalization performance, and thus, the prediction quality (for predictions, use the "Decision Tree Predictor" node). Available is the "Minimal Description Length" (MDL) pruning or it can also be switched off.

Reduced Error Pruning

If checked (default), a simple pruning method is used to cut the tree in a post-processing step: Starting at the leaves, each node is replaced with its most popular class, but only if the prediction accuracy doesn't decrease. Reduced error pruning has the advantage of simplicity and speed.

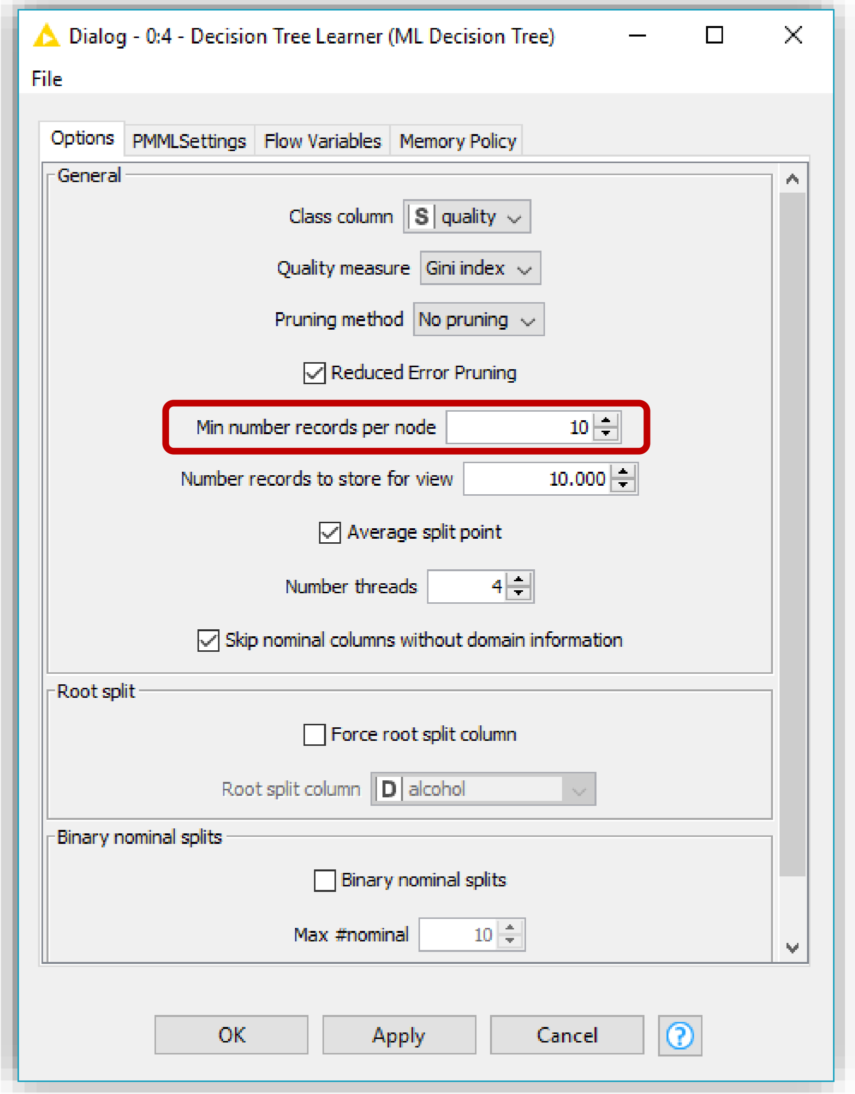
Min number records per node

To select the minimum number of records at least required in each node. If the number of records is smaller or equal to this number the tree is not grown any further. This corresponds to a stopping criteria (pre pruning).

Number records to store for view

To select the number of records stored in the tree for the view. The records are necessary to enable

Tuning Numeric Parameters



Dialog - 0:4 - Decision Tree Learner (ML Decision Tree)

General

Class column: S quality

Quality measure: Gini index

Pruning method: No pruning

☑ Reduced Error Pruning

Min number records per node: 10

Number records to store for view: 10.000

☑ Average split point

Number threads: 4

☑ Skip nominal columns without domain information

Root split

□ Force root split column

Root split column: D alcohol

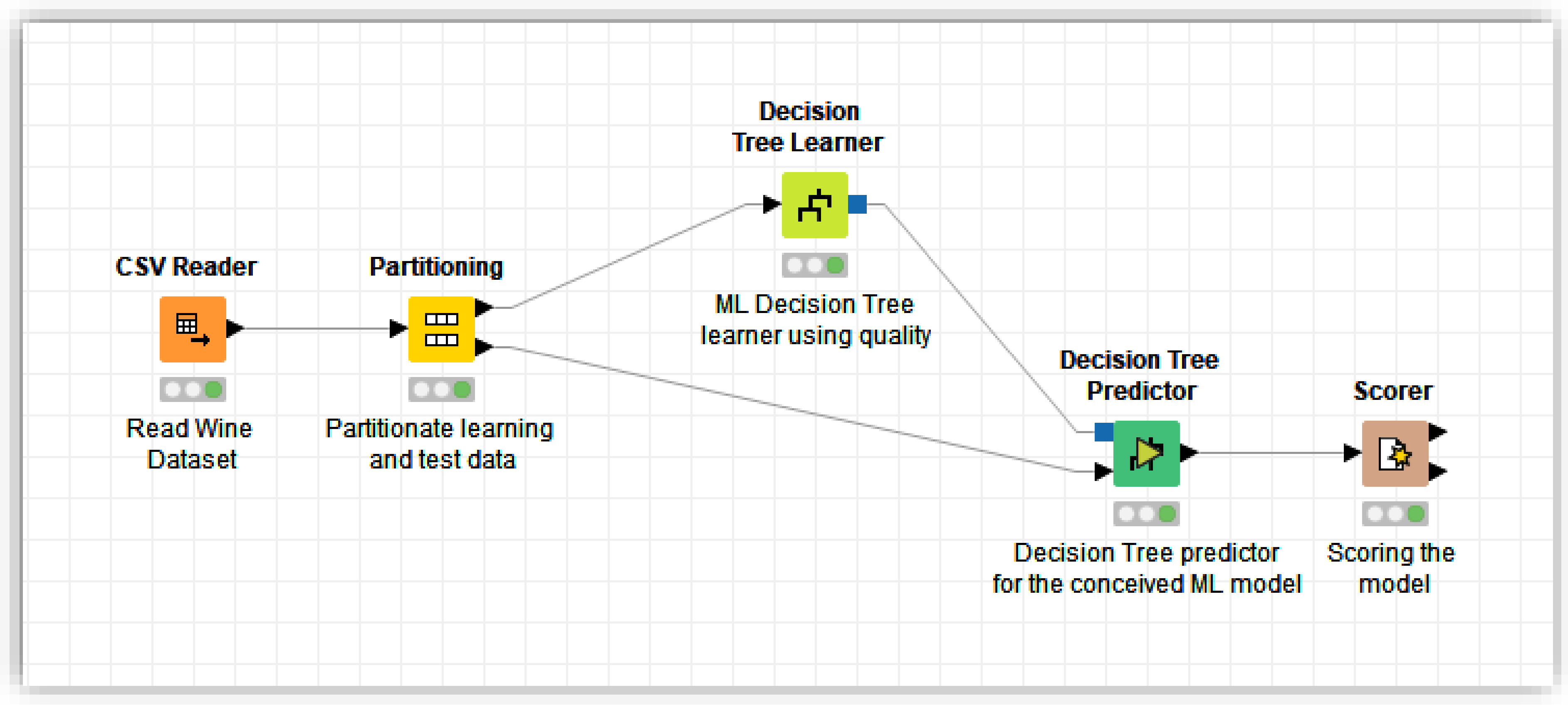
Binary nominal splits

□ Binary nominal splits

Max #nominal: 10

Tuning Numeric Parameters
Parameter Optimization Loop Nodes

Optimize the value of some parameters with respect to a cost function


CSV Reader
Read Wine Dataset
Partitioning
Partitionate learning and test data
Decision Tree Learner
ML Decision Tree learner using quality
Decision Tree Predictor
Decision Tree predictor for the conceived ML model
Scorer
Scoring the model

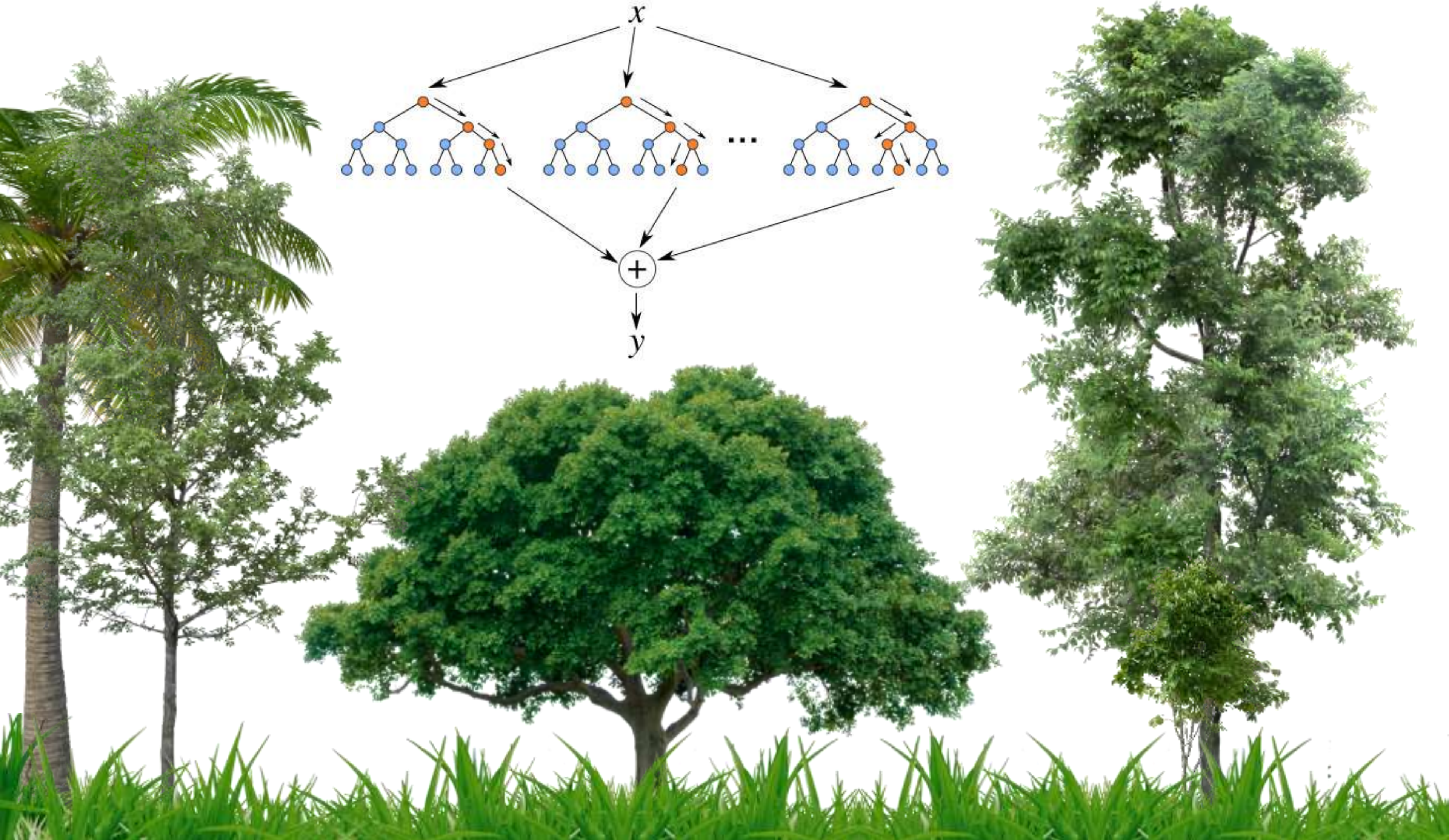
Model Hyper-parameters

Revelant concepts…

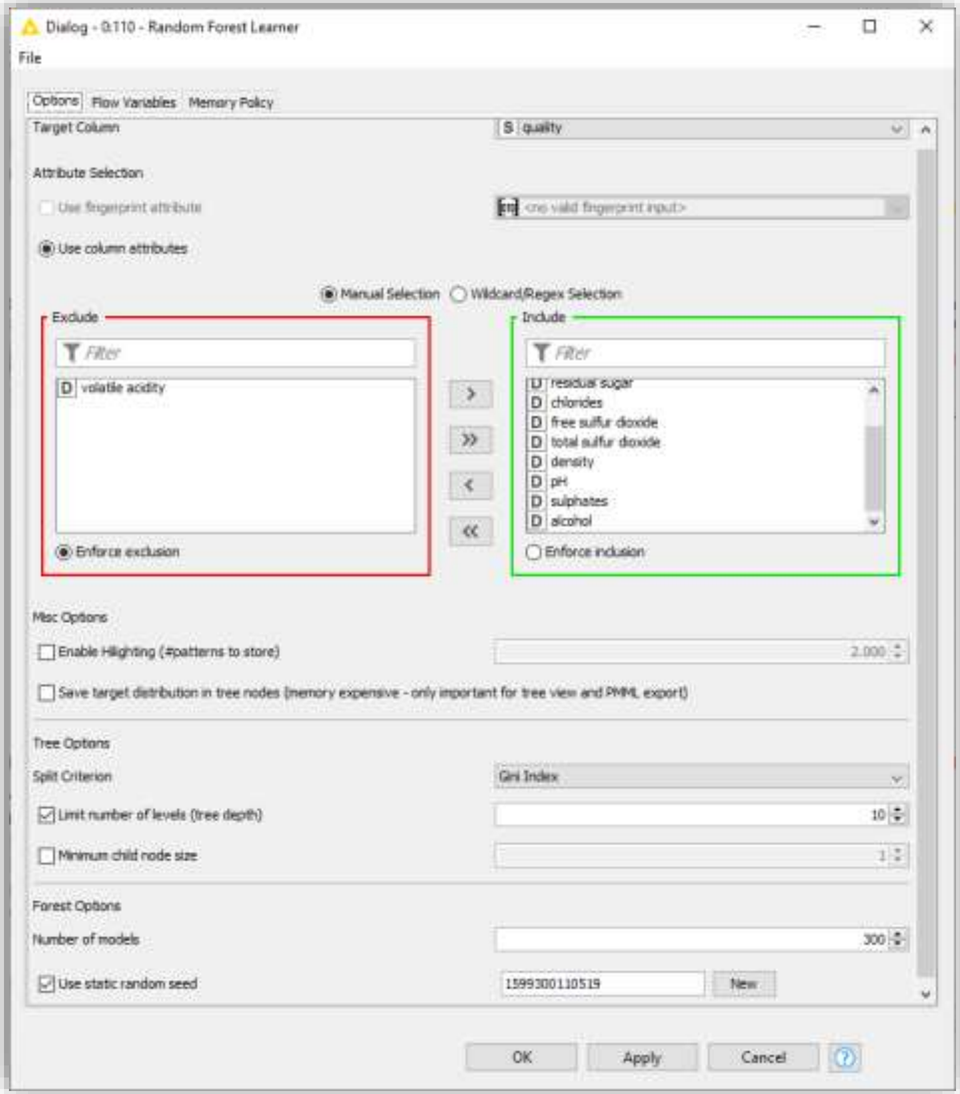
- Model Parameters: a model's (internal) configuration variable whose value is estimated from training data, i.e., the value is not set manually. Some examples include:
  - Weights in Artificial Neural Networks
  - Support vectors in Support Vector Machines

- Model Hyperparameters: a model's (external) configuration variable whose value can be set manually. It is difficult to know, beforehand, the best value of each hyperparameter. Tuning a model consists in finding the best (or, at least, a good) configuration of hyperparameters. Examples include:
  - Optimizer and learning rate in Artificial Neural Networks
  - C and sigma in Support Vector Machines
  - Quality measure and pruning method in Decision Trees

Random Forests



Random Forests



Target Column: S quality

Attribute Selection

□ Use fingerprint attribute <no valid fingerprint input>

● Use column attributes

● Manual Selection ○ Wildcard/Regex Selection

Exclude: D volatile acidity

● Enforce exclusion

Include: D residual sugar, D chlorides, D free sulfur dioxide, D total sulfur dioxide, D density, D pH, D sulphates, D alcohol

○ Enforce inclusion

Misc Options

□ Enable Hilighting (#patterns to store) 2.000

□ Save target distribution in tree nodes (memory expensive - only important for tree view and PMML export)

Tree Options

Split Criterion: Gini Index

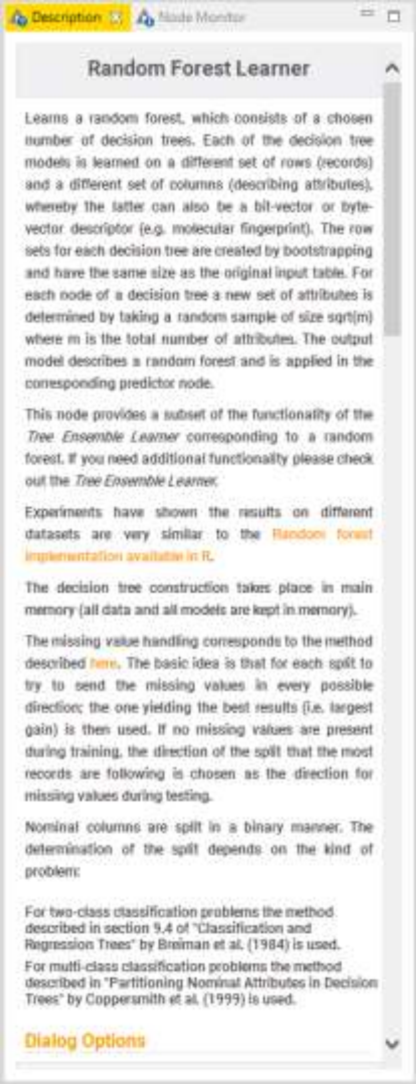
☑ Limit number of levels (tree depth) 10

□ Minimum child node size 1

Forest Options

Number of models: 300

☑ Use static random seed 1599300110519



Random Forest Learner

Learns a random forest, which consists of a chosen number of decision trees. Each of the decision tree models is learned on a different set of rows (records) and a different set of columns (describing attributes), whereby the latter can also be a bit-vector or byte-vector descriptor (e.g. molecular fingerprint). The row sets for each decision tree are created by bootstrapping and have the same size as the original input table. For each node of a decision tree a new set of attributes is determined by taking a random sample of size sqrt(m) where m is the total number of attributes. The output model describes a random forest and is applied in the corresponding predictor node.

This node provides a subset of the functionality of the *Tree Ensemble Learner* corresponding to a random forest. If you need additional functionality please check out the *Tree Ensemble Learner*.

Experiments have shown the results on different datasets are very similar to the Random forest implementation available in R.

The decision tree construction takes place in main memory (all data and all models are kept in memory).

The missing value handling corresponds to the method described here. The basic idea is that for each split to try to send the missing values in every possible direction; the one yielding the best results (i.e. largest gain) is then used. If no missing values are present during training, the direction of the split that the most records are following is chosen as the direction for missing values during testing.

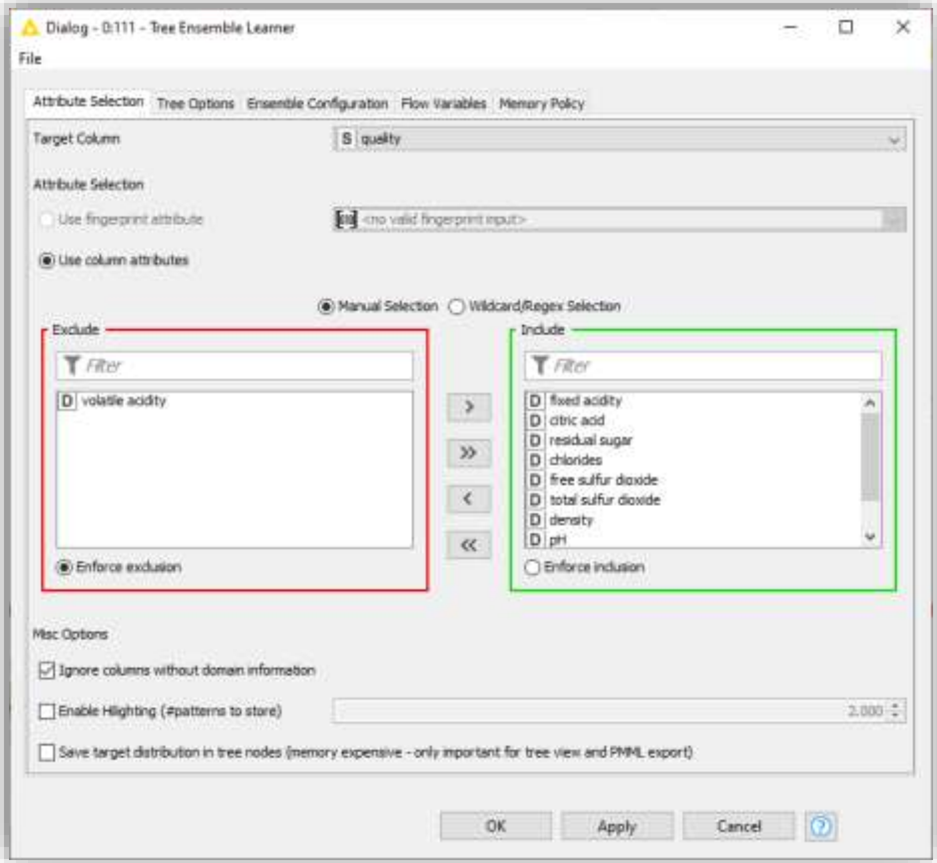
Nominal columns are split in a binary manner. The determination of the split depends on the kind of problem:

For two-class classification problems the method described in section 9.4 of "Classification and Regression Trees" by Breiman et al. (1984) is used.

For multi-class classification problems the method described in "Partitioning Nominal Attributes in Decision Trees" by Coppersmith et al. (1999) is used.

Dialog Options

Tree Ensembles



Target Column: S quality

Attribute Selection

○ Use fingerprint attribute <no valid fingerprint input>

● Use column attributes

● Manual Selection ○ Wildcard/Regex Selection

Exclude: D volatile acidity

● Enforce exclusion

Include: D fixed acidity, D citric acid, D residual sugar, D chlorides, D free sulfur dioxide, D total sulfur dioxide, D density, D pH

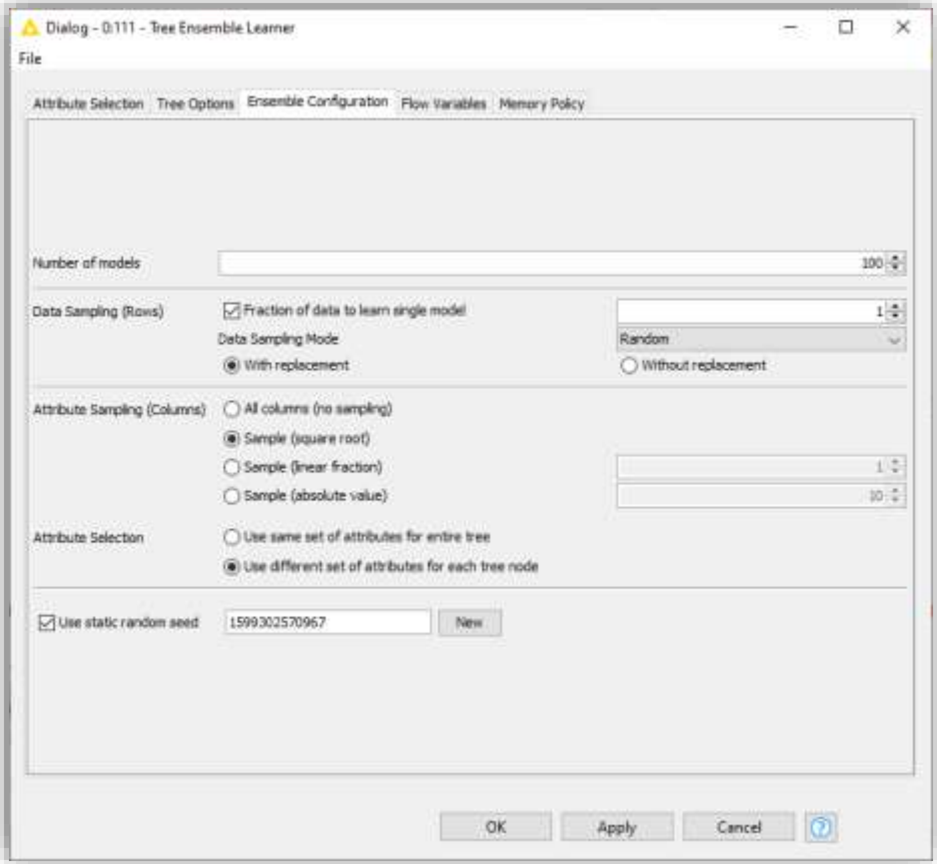
○ Enforce inclusion

Misc Options

☑ Ignore columns without domain information

□ Enable Hilighting (#patterns to store) 2,000

□ Save target distribution in tree nodes (memory expensive - only important for tree view and PMML export)



Number of models: 100

Data Sampling (Rows): ☑ Fraction of data to learn single model 1

Data Sampling Mode: Random

● With replacement ○ Without replacement

Attribute Sampling (Columns):

○ All columns (no sampling)

● Sample (square root)

○ Sample (linear fraction) 1

○ Sample (absolute value) 10

Attribute Selection:

○ Use same set of attributes for entire tree

● Use different set of attributes for each tree node

☑ Use static random seed 1599302570967

Hands On

KNIME Analytics Platform
File Edit View Node Help
100%
Quick Access
KNIME Explorer
My-KNIME-Hub (hub.knime.com)
EXAMPLES (knime@hub.knime.com)
LOCAL (Local Workspace)
Aulas Exemplos
Aulas Exercicios
Example Workflows
ML Projects
KNIME_Aula_Exercicio
0: KNIME_Aula_Exercicio
Description
KNIME Hub Search
Search workflows, nodes, and more...
Workflow Coach
Recommended Nodes
File Reader
CSV Reader
Excel Reader (XLS)
Table Creator
Database Reader (legacy)
Table Reader
List Files
Database Connection T...
Node Repository
IO
Manipulation
Views
Analytics
DB
Other Data Types
Structured Data
Scripting
Tools & Services
Workflow Control
Workflow Abstraction
Reporting
Console
Outline
HANDS ON