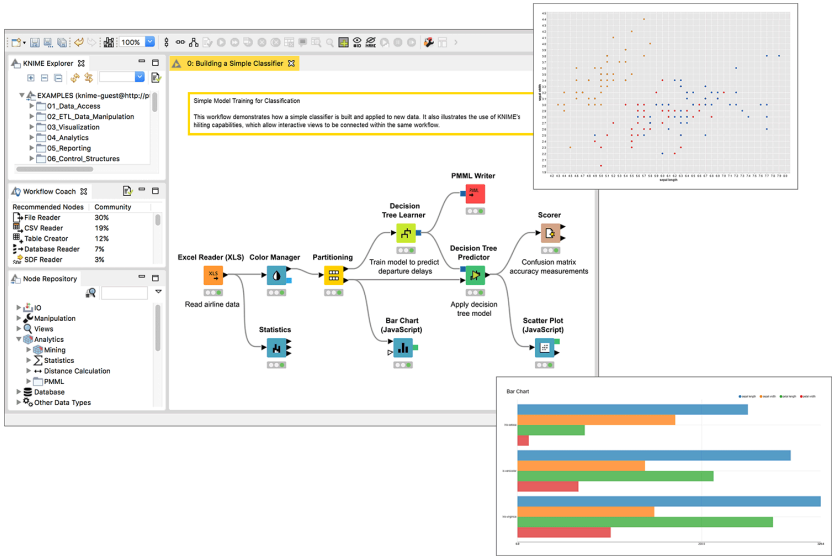
Overview
KNIME Analytics Platform

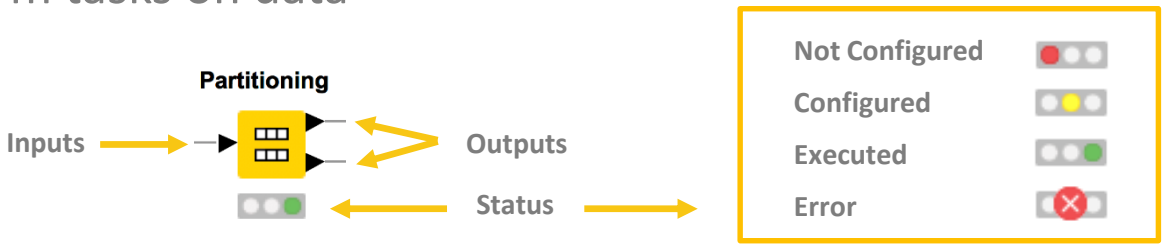
What is KNIME Analytics Platform?

- A tool for data analysis, manipulation, visualization, and reporting
- Based on the graphical programming paradigm
- Provides a diverse array of extensions:
  - Text Mining
  - Network Mining
  - Cheminformatics
  - Many integrations,
    such as Java, R, Python,
    Weka, Keras, Plotly, H2O, etc.

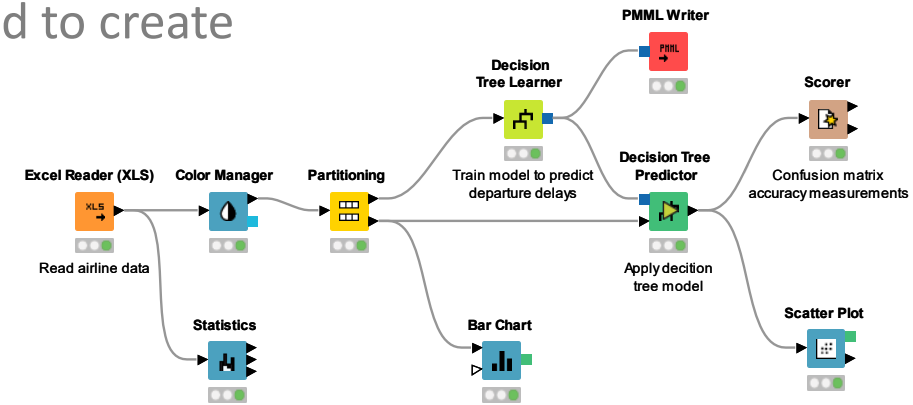


Visual KNIME Workflows

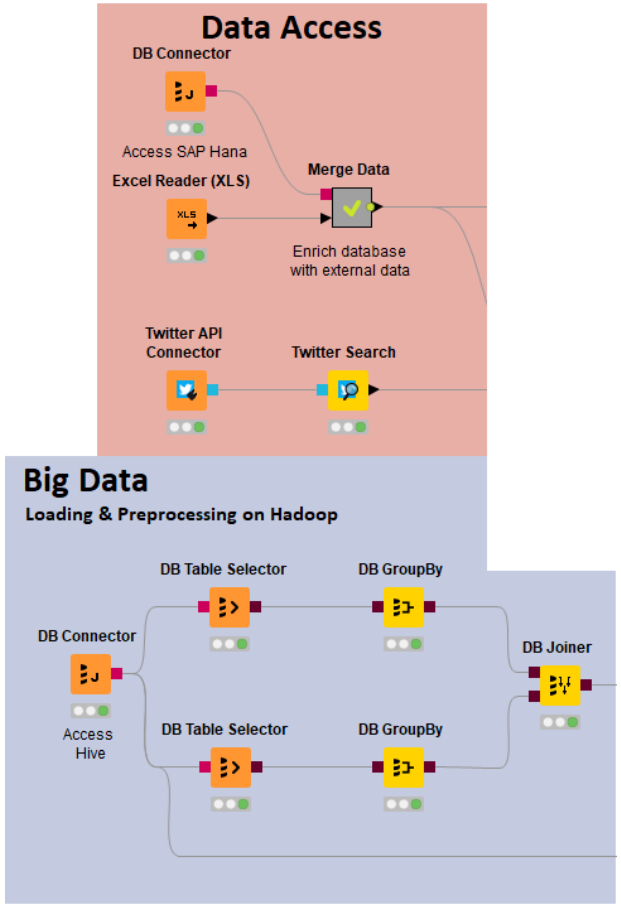
NODES perform tasks on data



Nodes are combined to create **WORKFLOWS**

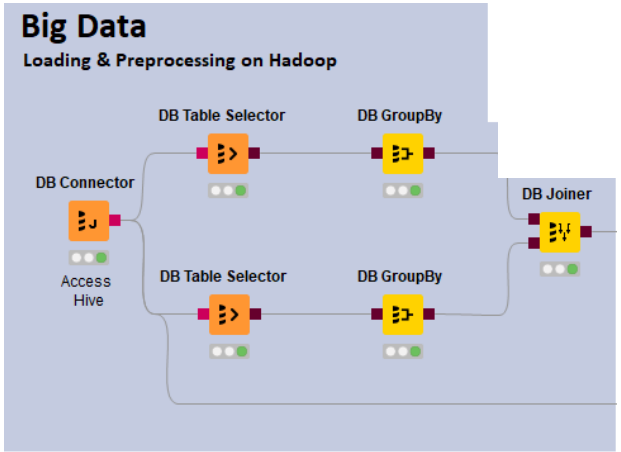


Data Access



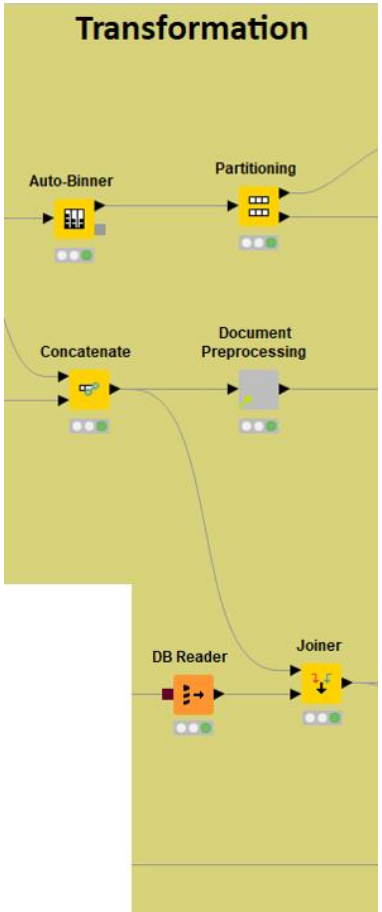
- Databases
  - MySQL, PostgreSQL
  - any JDBC (Oracle, DB2, MS SQL Server)
- Files
  - CSV, txt
  - Excel, Word, PDF
  - SAS, SPSS
  - XML
  - PMML
  - Images, texts, networks, chem
- Web, Cloud
  - REST, Web services
  - Twitter, Google

Big Data



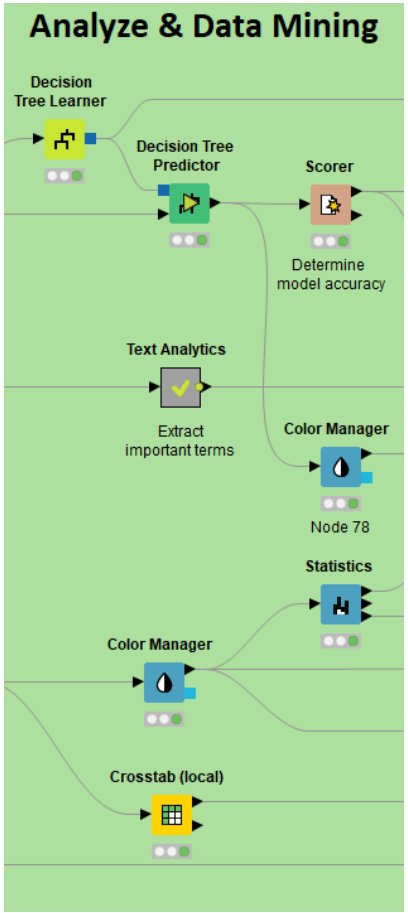
- Spark
- HDFS support
- Hive
- Impala
- In-database processing

Transformation



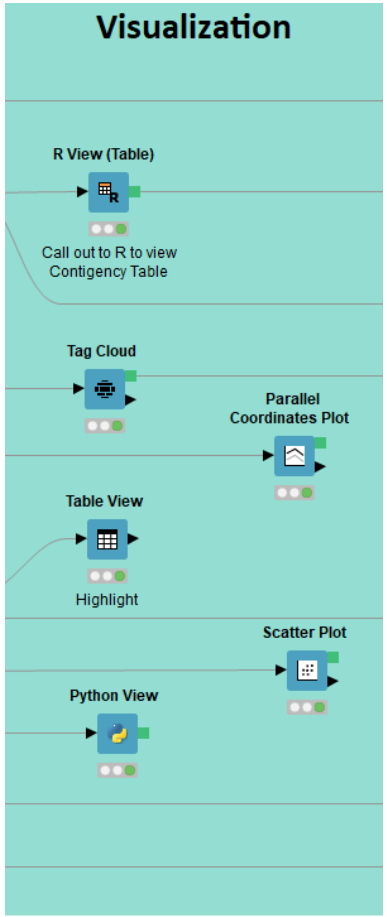
- Preprocessing
  - Row, column, matrix based
- Data blending
  - Join, concatenate, append
- Aggregation
  - Grouping, pivoting, binning
- Feature Creation and Selection

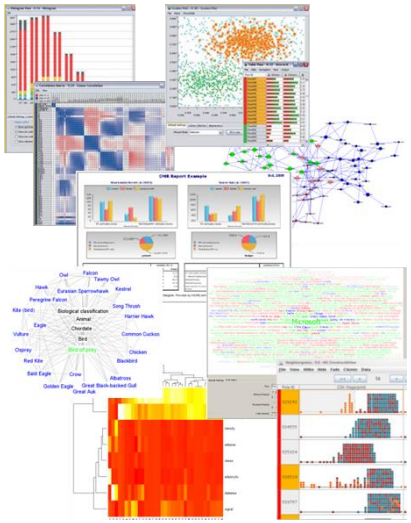
Analysis & Data Mining



- Regression
  - Linear, logistic
- Classification
  - Decision tree, ensembles, SVM, MLP, Naïve Bayes
- Clustering
  - k-means, DBSCAN, hierarchical
- Validation
  - Cross-validation, scoring, ROC
- Deep Learning
  - Keras, DL4J
- External
  - R, Python, Weka, H2O, Keras

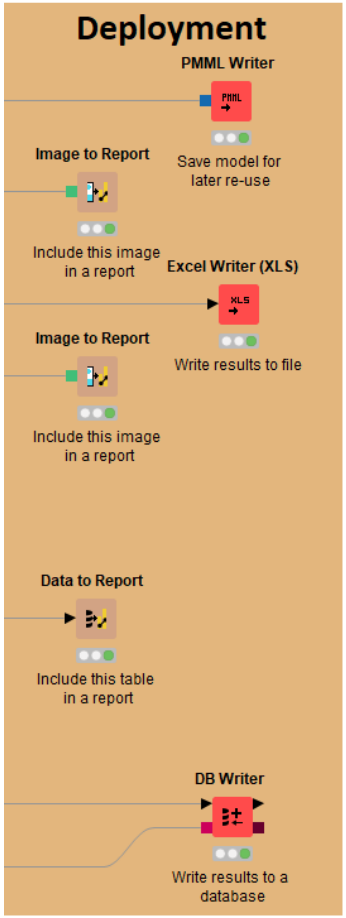
Visualization





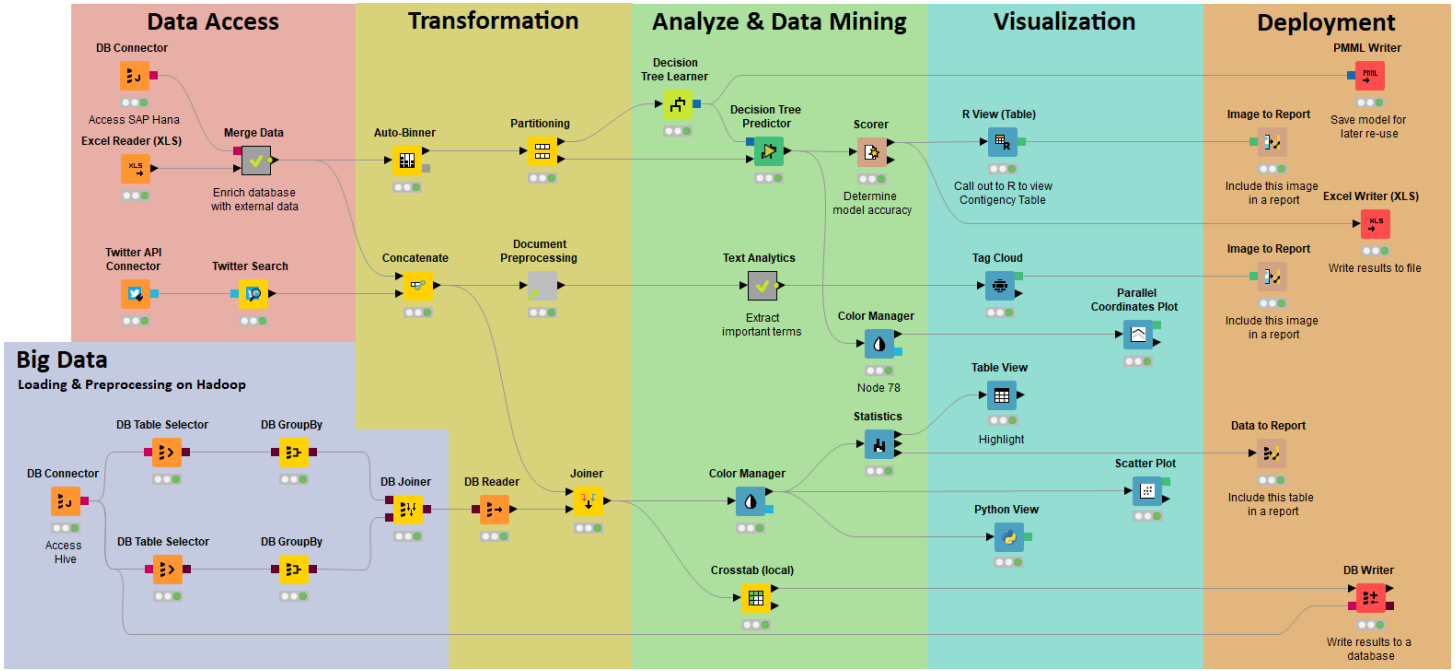
- Interactive Visualizations
- JavaScript-based nodes
  - Scatter Plot, Box Plot, Line Plot
  - Networks, ROC Curve, Decision Tree
  - Plotly Integration
  - Adding more with each release!
- Misc
  - Tag cloud, open street map, molecules
- Script-based visualizations
  - R, Python

Deployment



- Database
- Files
  - Excel, CSV, txt
  - XML
  - PMML
  - to: local, KNIME Server, SSH-, FTP-Server
- BIRT Reporting

Over 2000 Native and Embedded Nodes Included:



Data Access
MySQL, Oracle, ...
SAS, SPSS, ...
Excel, Flat, ...
Hive, Impala, ...
XML, JSON, PMML
Text, Doc, Image, ...
Web Crawlers
Industry Specific
Community / 3rd

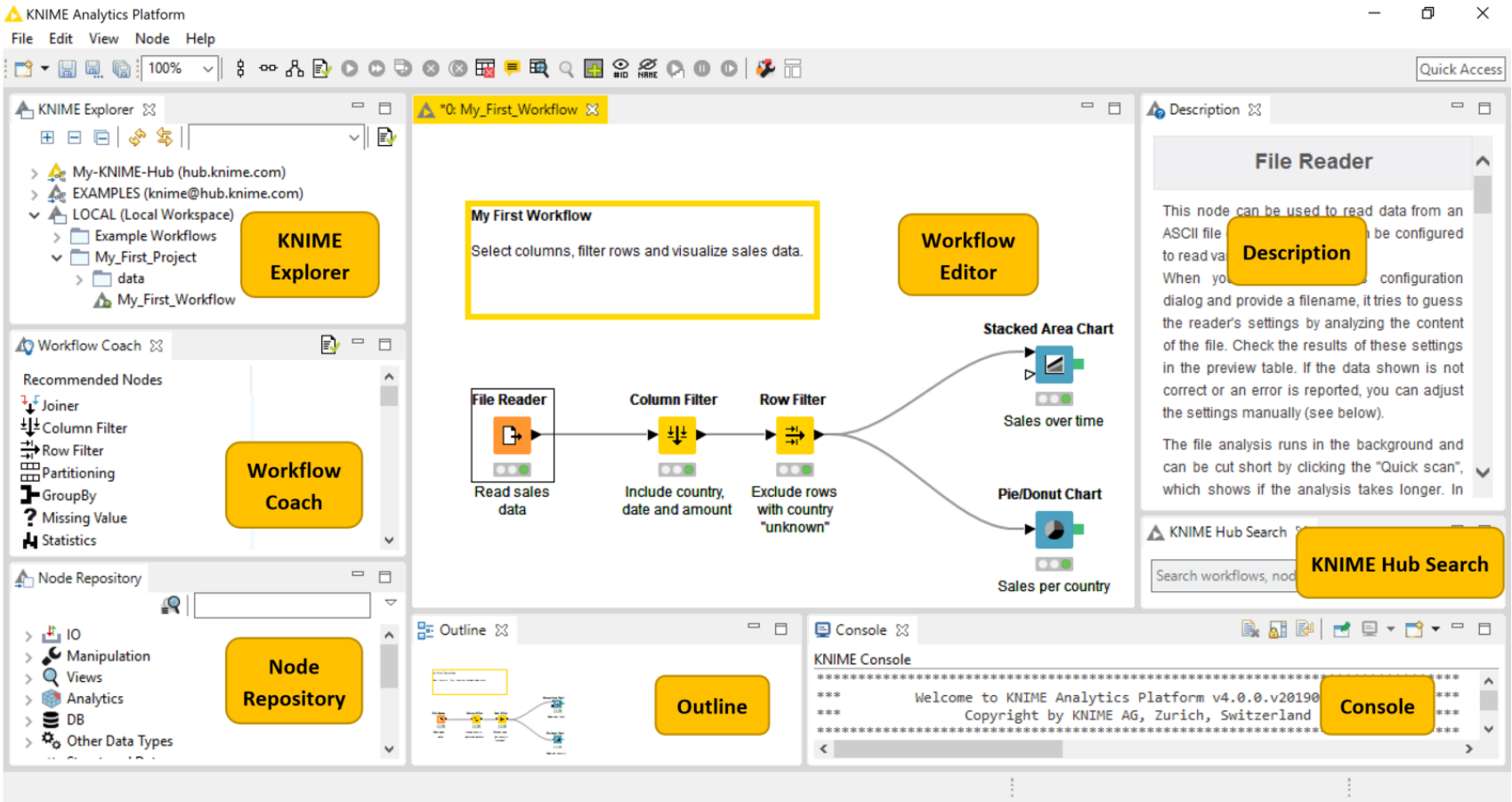
Transformation
Row
Column
Matrix
Text, Image
Time Series
Java
Python
Community / 3rd

Analysis & Mining
Statistics
Data Mining
Machine Learning
Web Analytics
Text Mining
Network Analysis
Social Media Analysis
R, Weka, Python
Community / 3rd

Visualization
R
JFreeChart
JavaScript
Plotly
Community / 3rd

Deployment
via BIRT
PMML
XML, JSON
Databases
Excel, Flat, etc.
Text, Doc, Image
Industry Specific
Community / 3rd

The KNIME Workbench

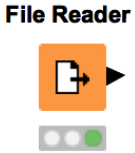


More on Nodes…

A node can have 3 states:

Not Configured:
The node is waiting for configuration or incoming data.

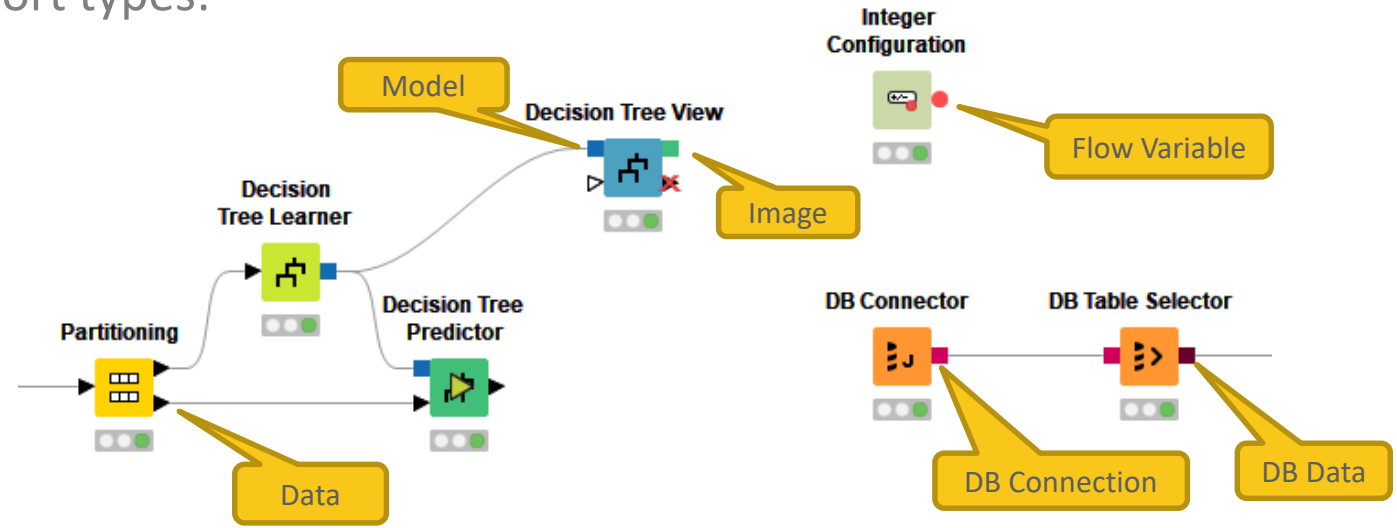
Configured:
The node has been configured correctly, and can be executed.



Executed:
The node has been successfully executed. Results may be viewed and used in downstream nodes.

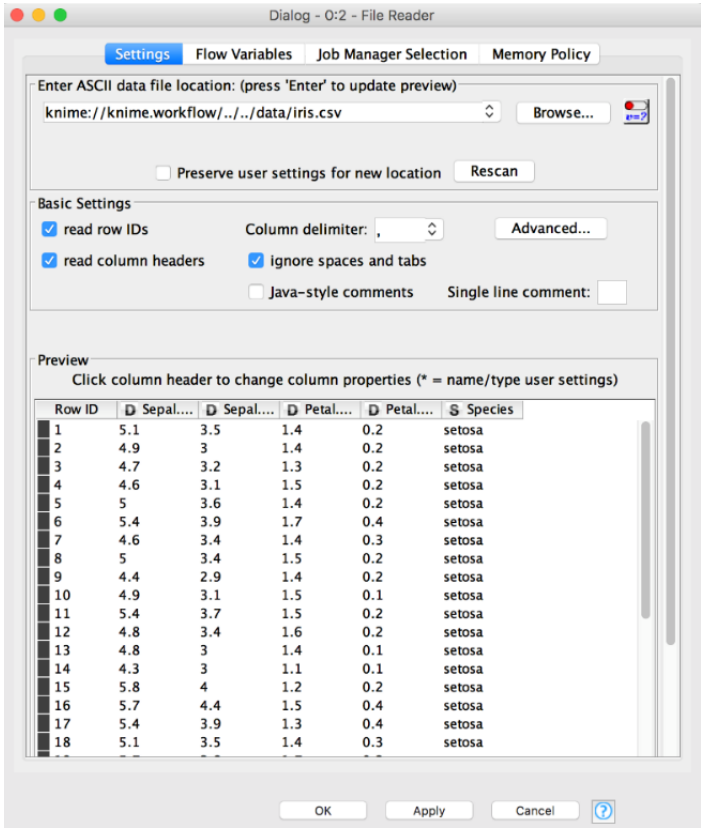
Inserting and Connecting Nodes

- Insert nodes into workspace by dragging them from Node Repository or by double-clicking in Node Repository
- Connect nodes by left-clicking output port of Node A and dragging the cursor to (matching) input port of Node B
- Common port types:



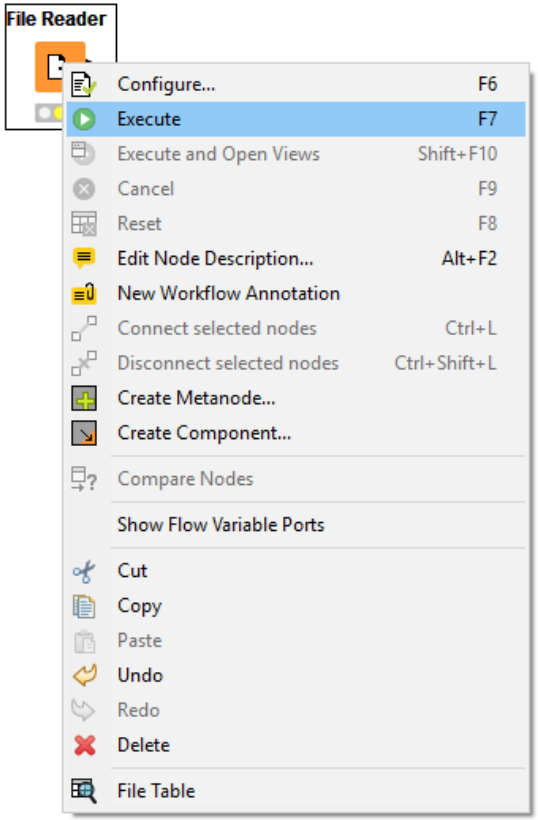
Node Configuration

- Most nodes require configuration
- To access a node configuration window:
  - Double-click the node
  - Right-click -> Configure



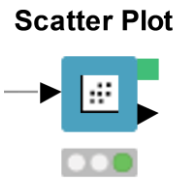
Node Execution

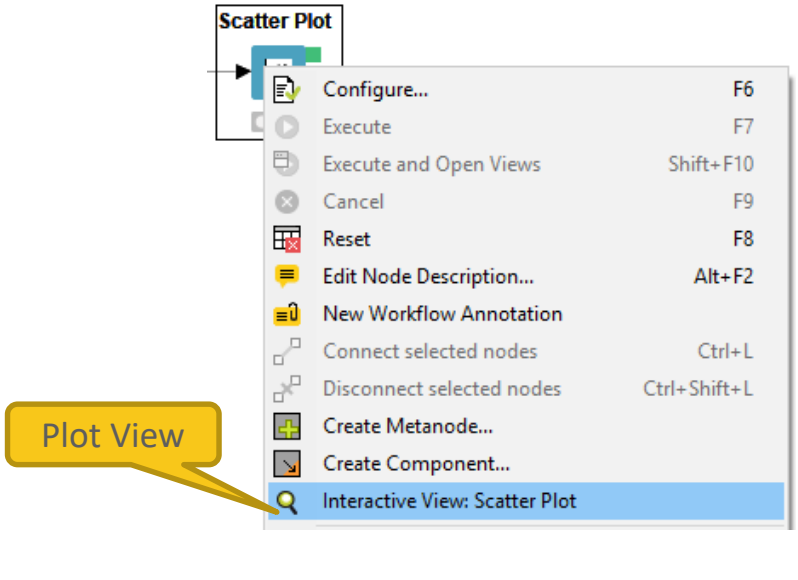
- Right-click node
- Select Execute in the context menu
- If execution is successful, status shows green light
- If execution encounters errors, status shows red light

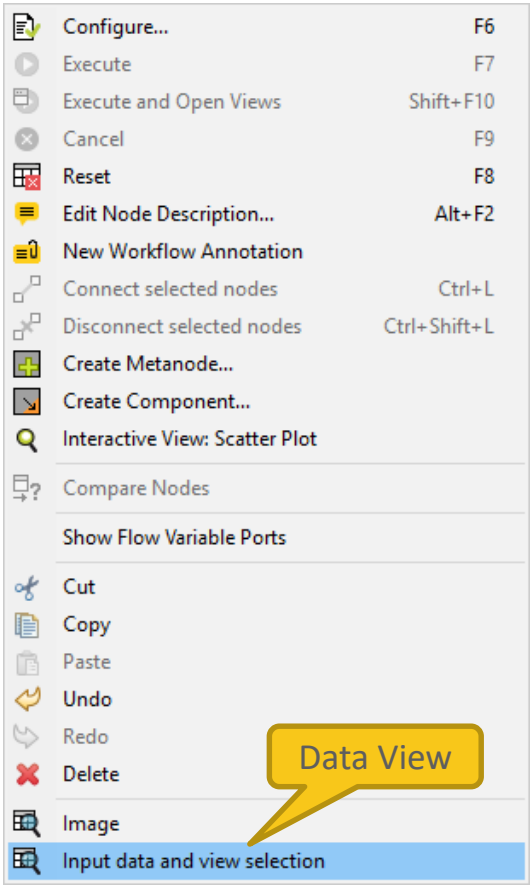


Node Views

- Right-click node
- Select Views in context menu
- Select output port to inspect execution results



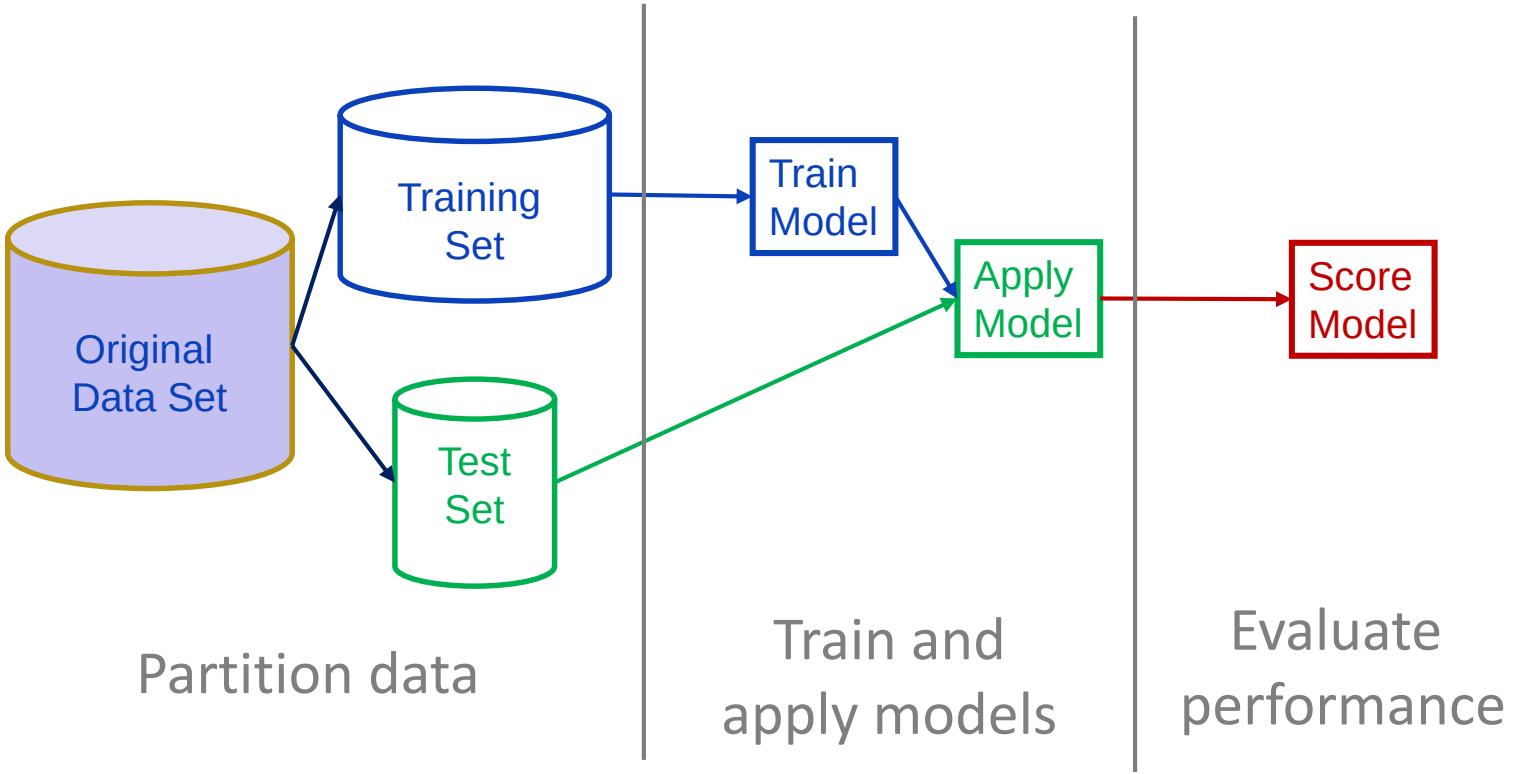




Data Mining

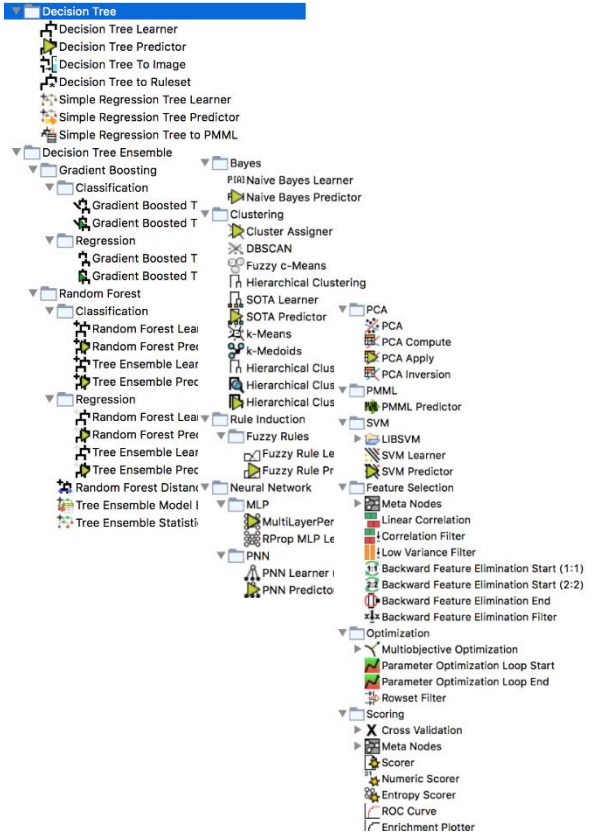
Partition, Learn, Predict, Score

Data Mining: Process Overview



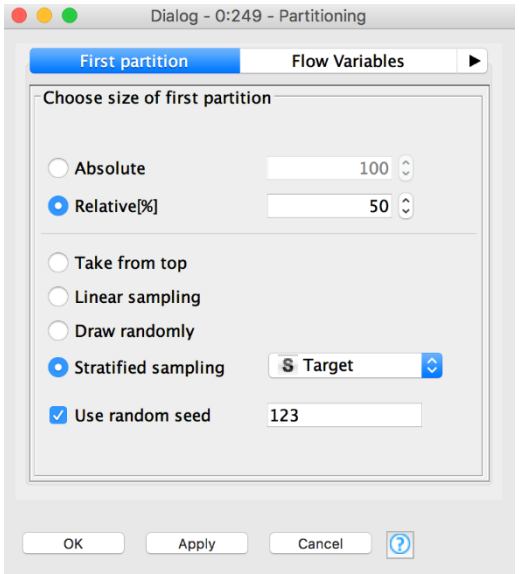
Data Mining in KNIME

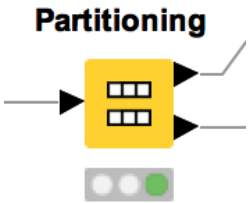
- KNIME has many modeling tools!
  - Decision tree, random forest, SVM, regression, neural networks, clustering, …
  - and integrations with other libraries: R, Python, H2O, WEKA, libSVM, etc.
- And many model evaluation nodes
  - ROC, standard, numeric and entropy scorers
  - Feature elimination
  - Cross validation

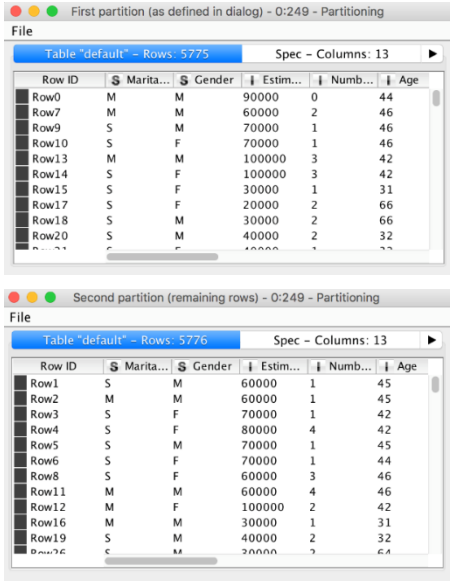


New Node: Partitioning

- Use to split data into training and evaluation sets
  - Partition by count (e.g. 10 rows) or fraction (e.g. 10%)
  - Sample by a variety of methods; random, linear, stratified

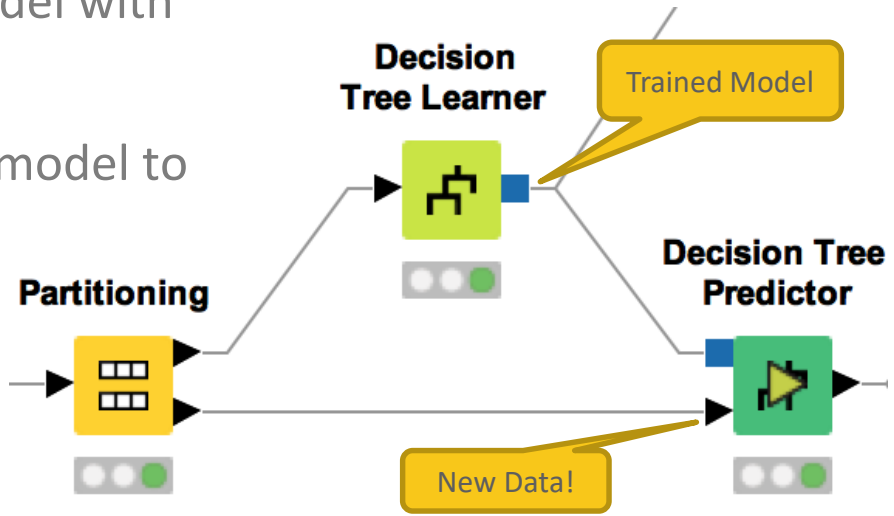






Learner-Predictor Motif

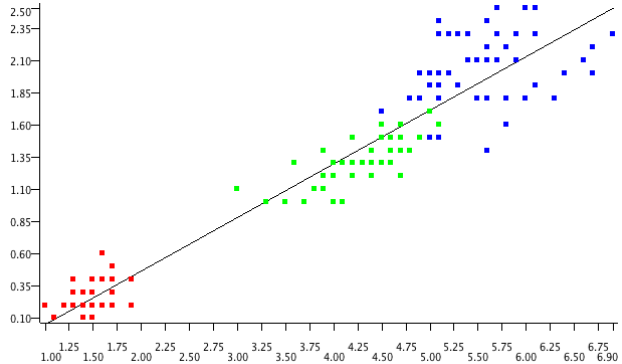
- Most data mining approaches in KNIME use a Learner-predictor motif.
- The Learner node trains the model with its input data.
- The Predictor node applies the model to a different subset of data.



Regression

Predict *numeric* outcomes on existing data (supervised)

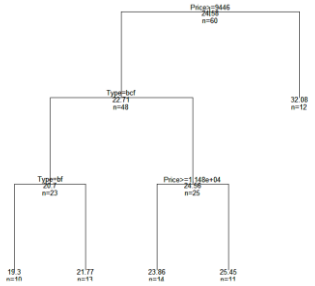
- Applications
  - Forecasting
  - Quantitative Analysis
- Methods
  - Linear
  - Polynomial
  - Regression Trees
  - Partial Least Squares



Statistics on Linear Regression

Variable	Coeff.	Std. Err.	t-value	P>|t|
Petal.Length	0.4158	0.0096	43.3872	0.0
Intercept	-0.3631	0.0398	-9.1312	4.44E-16

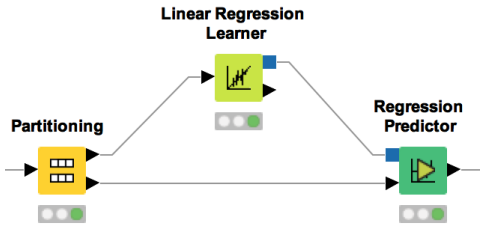
Multiple R-Squared: 0.9271
Adjusted R-Squared: 0.9266

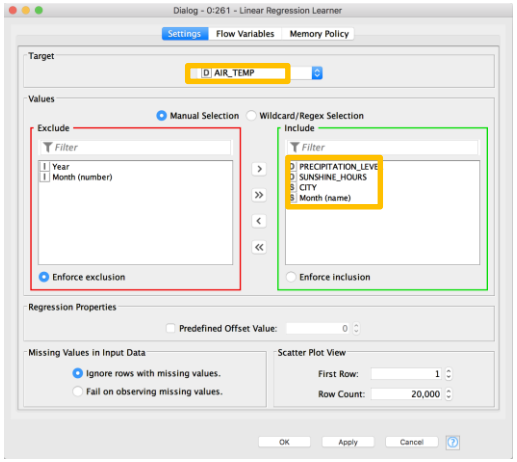


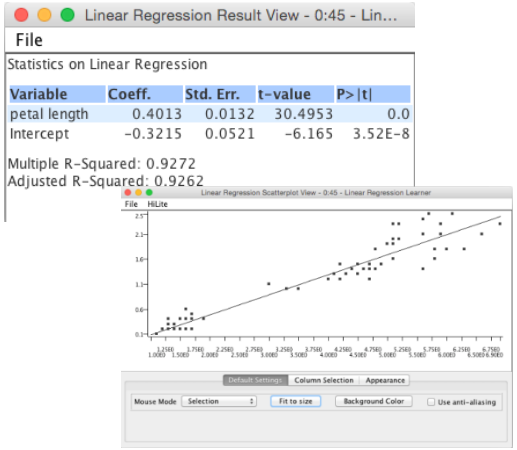
New Nodes: Linear Regression Learner & Regression Predictor

A linear model relating a dependent variable to 1 or more independent variables

- Model coefficients provided in 2nd output port
- Also available: Polynomial and Tree Ensemble Regression nodes



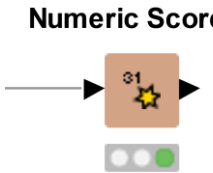


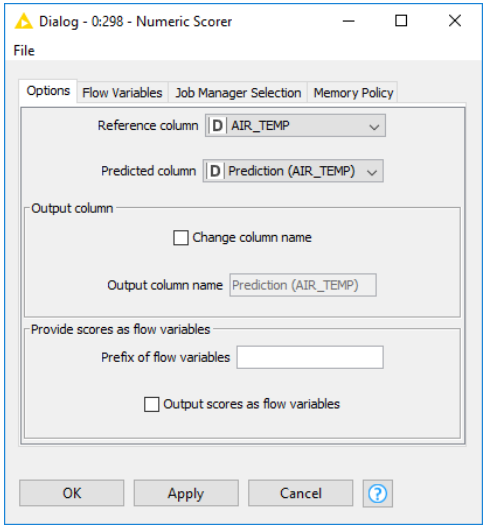


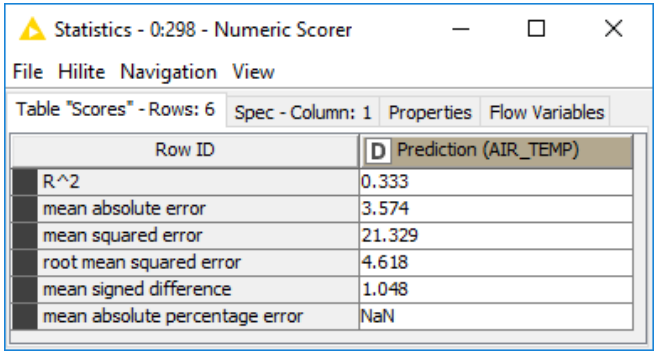
New Node: Numeric Scorer

Similar to scorer node, but for nodes with *numeric* predictions (e.g. linear/polynomial regression)

- Compare dependent variable values to predicted values to evaluate goodness of fit.
- Report R^2, RMSD, SEM etc.







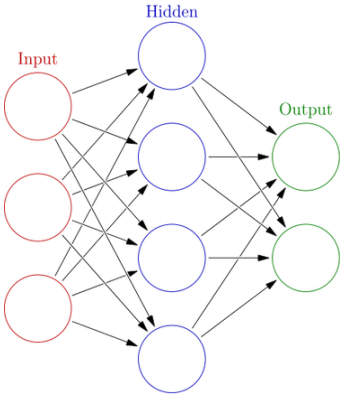
Row ID	Prediction (AIR_TEMP)
R^2	0.333
mean absolute error	3.574
mean squared error	21.329
root mean squared error	4.618
mean signed difference	1.048
mean absolute percentage error	NaN

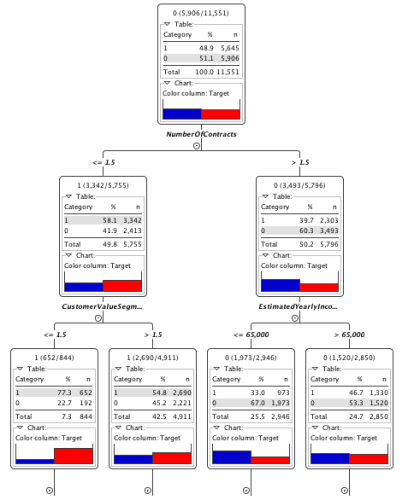
Classification

Predict *nominal* outcomes on existing data (supervised)

- Applications
  - Churn analysis (yes/no)
  - Chemical activity (active/inactive)
  - Spam detection (spam/not spam)
  - Optical character recognition (A-Z)

- Methods
  - Decision Trees
  - Neural Networks
  - Naïve Bayes
  - Logistic Regression





Class counts for Target

Class:	0	1
Count:	5906	5645

Total count: 11551

Threshold to used for zero probabilities: 0.0

Gaussian distribution for Age per class value

	0	1
Count:	5906	5645
Mean:	49.68557	46.82604
Std. Deviation:	12.27388	10.16363
Rate:	51%	49%

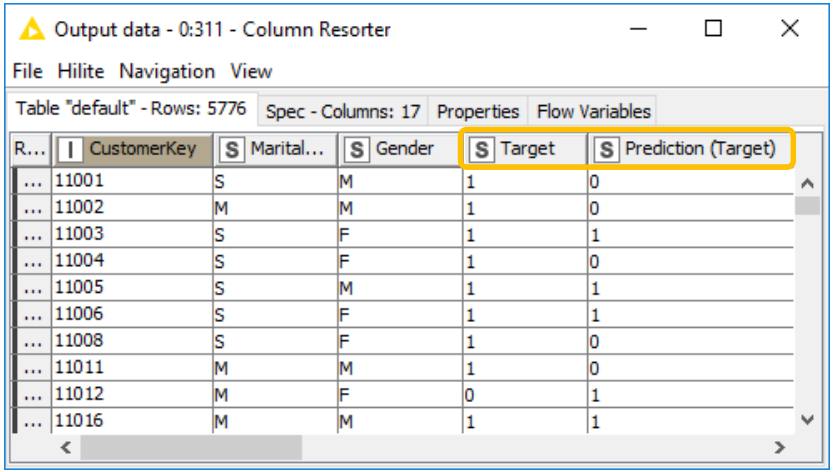
Gaussian distribution for Available401K per class value

	0	1
Count:	5906	5645
Mean:	0.68134	0.68485
Std. Deviation:	0.466	0.46462
Rate:	51%	49%

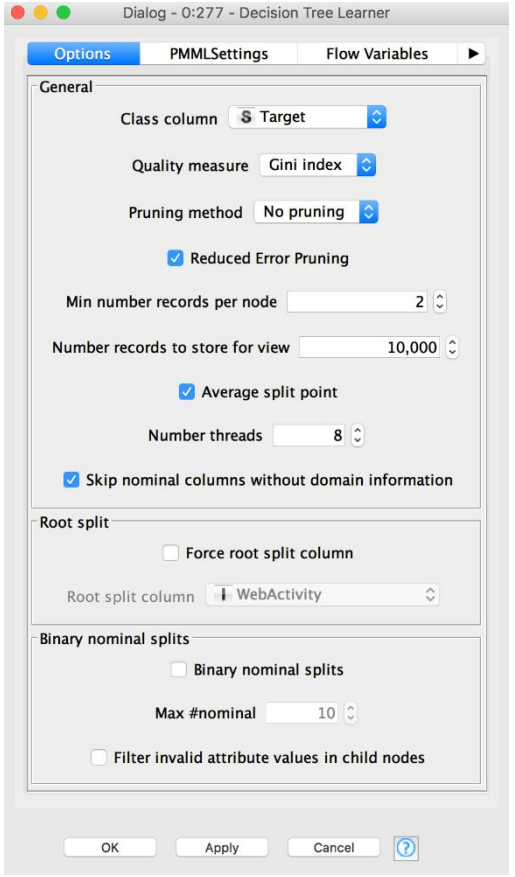
Target Column

- Target column contains values that are predicted by the classification model
- Binomial target values are often encoded to 1 and 0

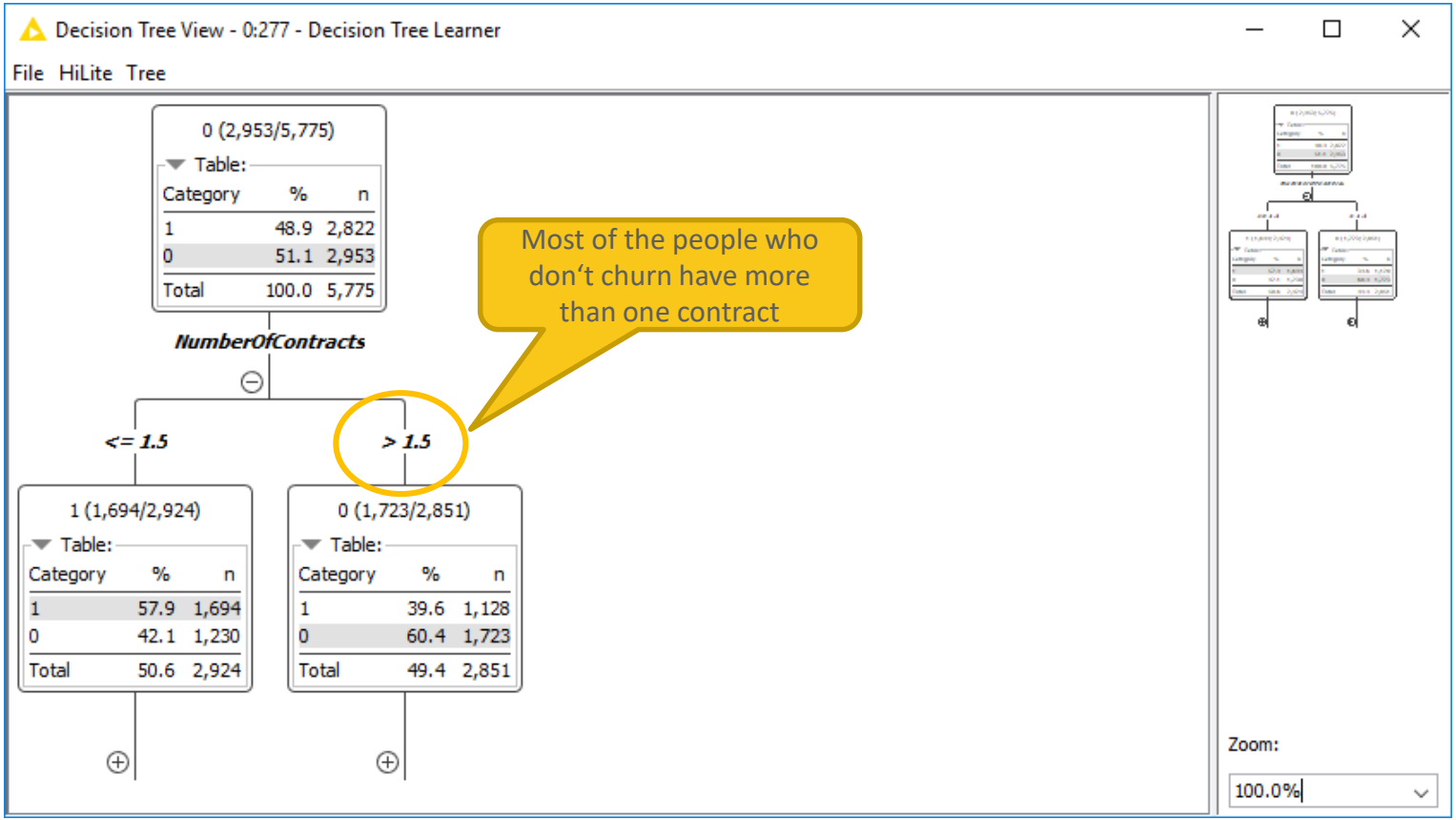
Application	Target Column	Target Values
Churn analysis	Churn	Yes/No or 1/0
Chemical activity	Active	Yes/No or 1/0
Spam Detection	Spam	Yes/No or 1/0
Optical Character Recognition	Character	A-Z



New Node: Decision Tree Learner

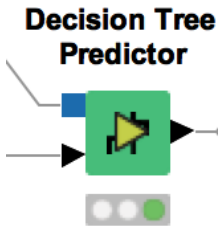


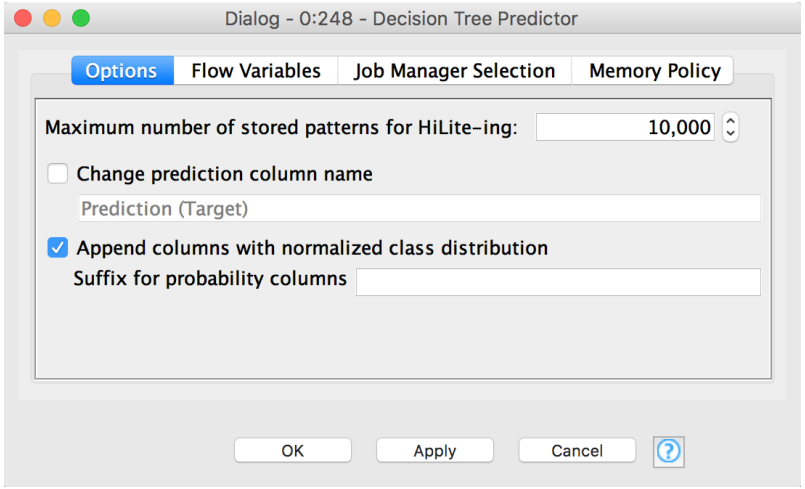
Decision Tree View



New Node: Decision Tree Predictor

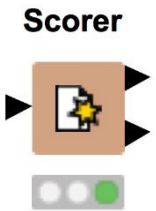
- Takes a decision tree model & applies it to new data
- Check the box to append class probabilities

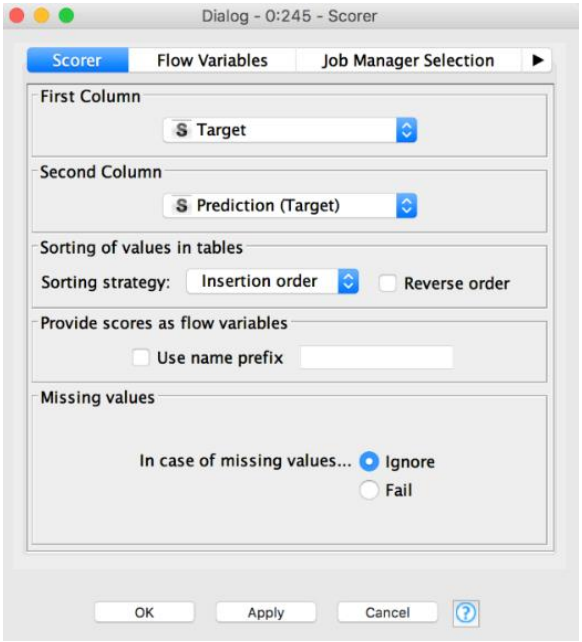




New Node: Scorer

Compare predicted results to known truth in order to evaluate model quality





New Node: Scorer

Confusion matrix shows the distribution of model errors

Confusion Matrix - 0:297 - Scorer

File Hilite

Target \ Prediction (Target)	1	0
1	2073	750
0	759	2193

Correct classified: 4,266 Wrong classified: 1,509

Accuracy: 73.87 % Error: 26.13 %

Cohen's kappa (κ) 0.477

An accuracy statistics table provides a detailed analysis of model quality

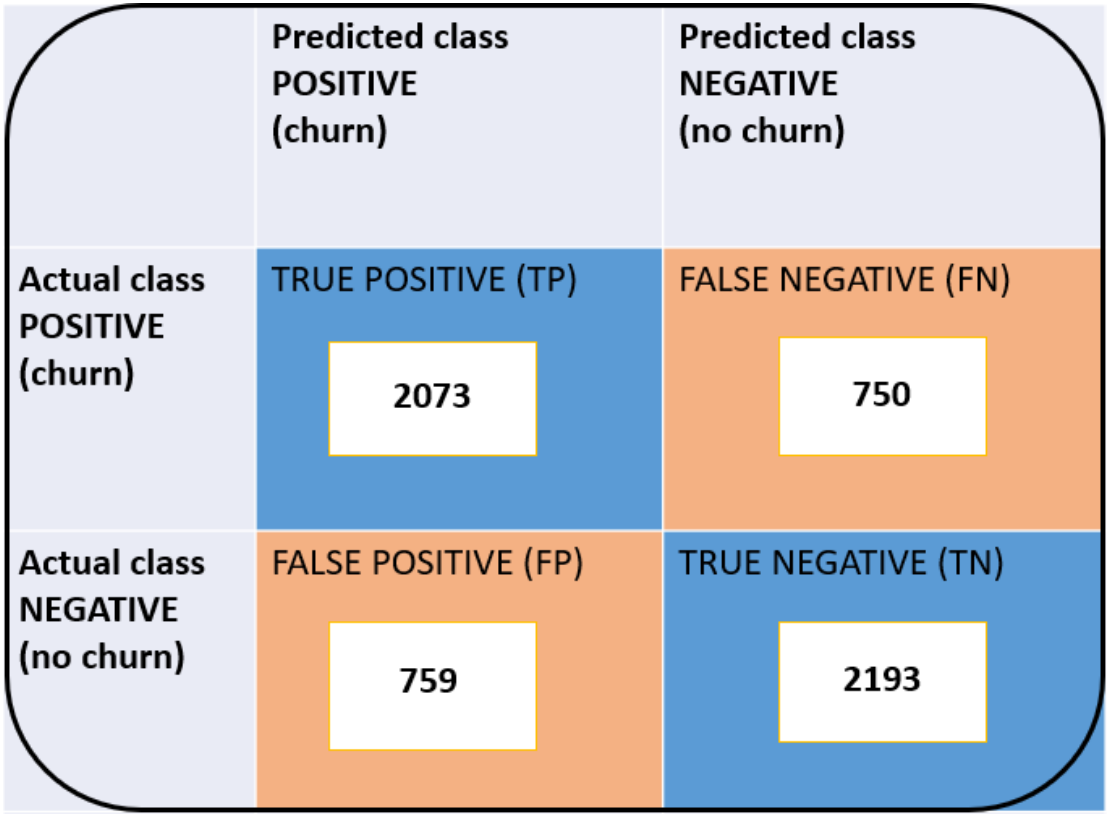
Accuracy statistics - 0:297 - Scorer

File Hilite Navigation View

Table "default" – Rows: 3 Spec – Columns: 11 Properties Flow Variables

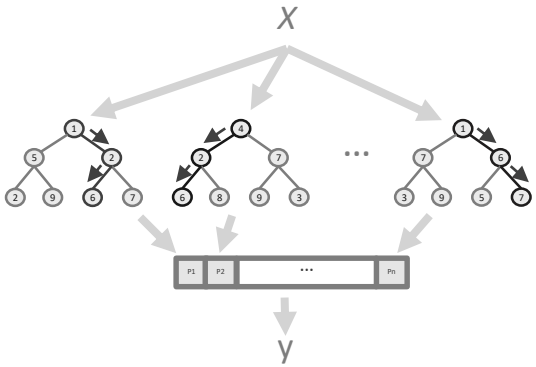
Row ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specifity	F-measure	Accuracy	Cohen's kappa
1	2073	759	2193	750	0.734	0.732	0.734	0.743	0.733	?	?
0	2193	750	2073	759	0.743	0.745	0.743	0.734	0.744	?	?
Overall	?	?	?	?	?	?	?	?	?	0.739	0.477

Confusion Matrix



KNIME's Tree Ensemble Models

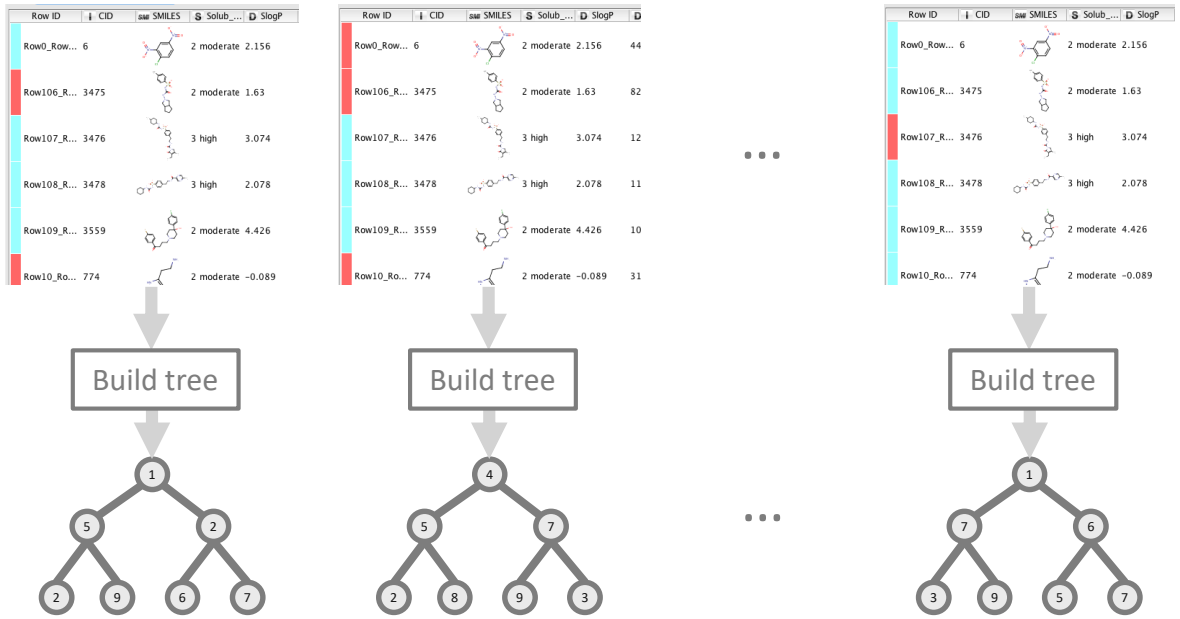
- The general idea is to take advantage of the "wisdom of the crowd"
- Ensemble models: Combining predictions from a large number of weak predictors, e.g. decision trees
- Leads to a more accurate and robust model
- This is called "bagging"



Typically: for classification the individual models vote and the majority wins; for regression, the individual predictions are averaged

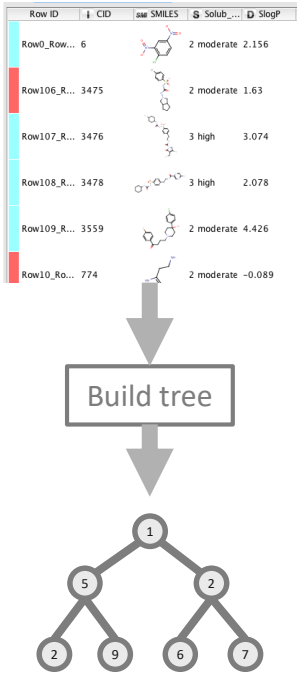
How Does Bagging Work?

- Pick a different random subset of the training data for each model in the ensemble (bag)



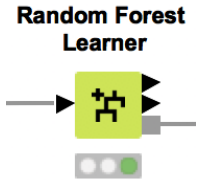
Random Forest

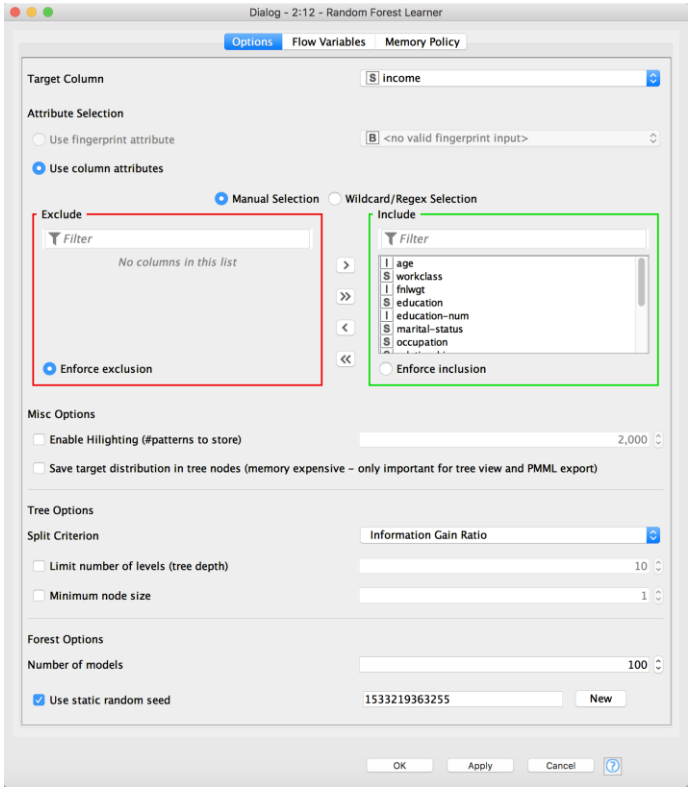
- Bag of decision trees, with an extra element of randomization when building the trees: **each node** in the decision tree only "sees" **a subset of the input columns**, typically $\sqrt{N}$
- Random forests tend to be very robust w.r.t. overfitting (though the individual trees are almost certainly overfit)
- Extra benefit: training tends to be much faster



New Nodes: Random Forest Learner

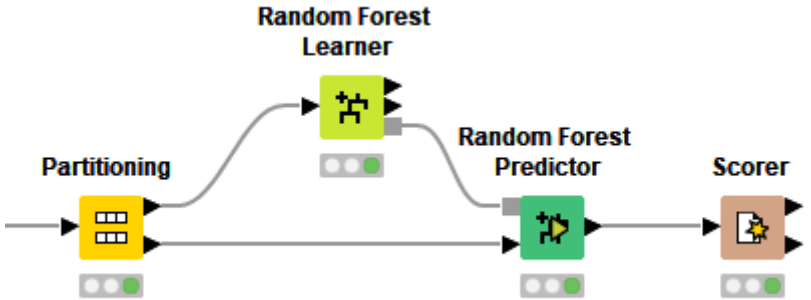
- The output model describes a random forest and is applied in the corresponding predictor node using a simple majority vote
- The statistics table on the attributes tells how often each attribute…
  - … is used in the first three splits
  - … was a possible candidate in the first three splits

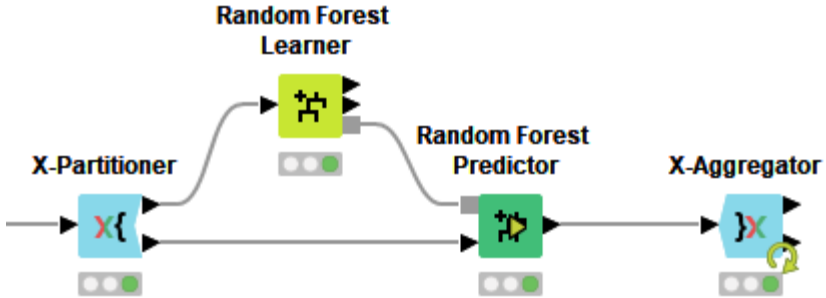




Cross Validation

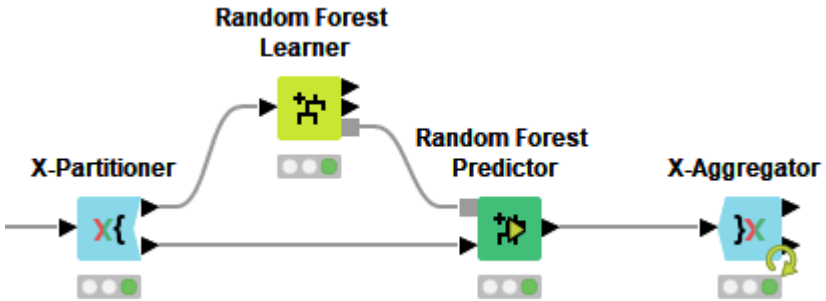
- Used to evaluate model stability
- Re-execute the modeling process many times using different data partitions
- Collect aggregated statistics on model accuracy

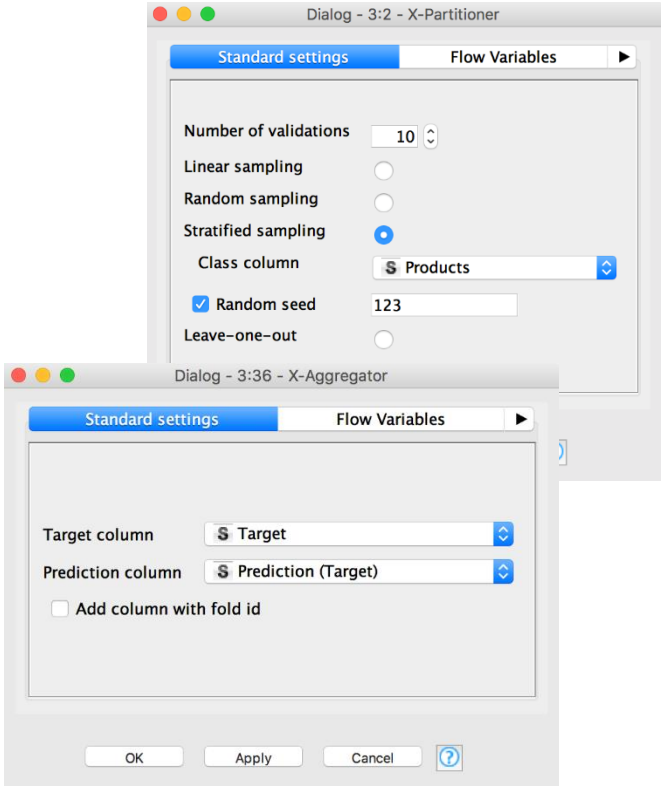




Example: Cross Validation

- X-Partitioner → X-Aggregator
- X-Partitioner replaces Partition
- X-Aggregator replaces Scorer
- Can be used with any learner/predictor





Thank You!

education@knime.com