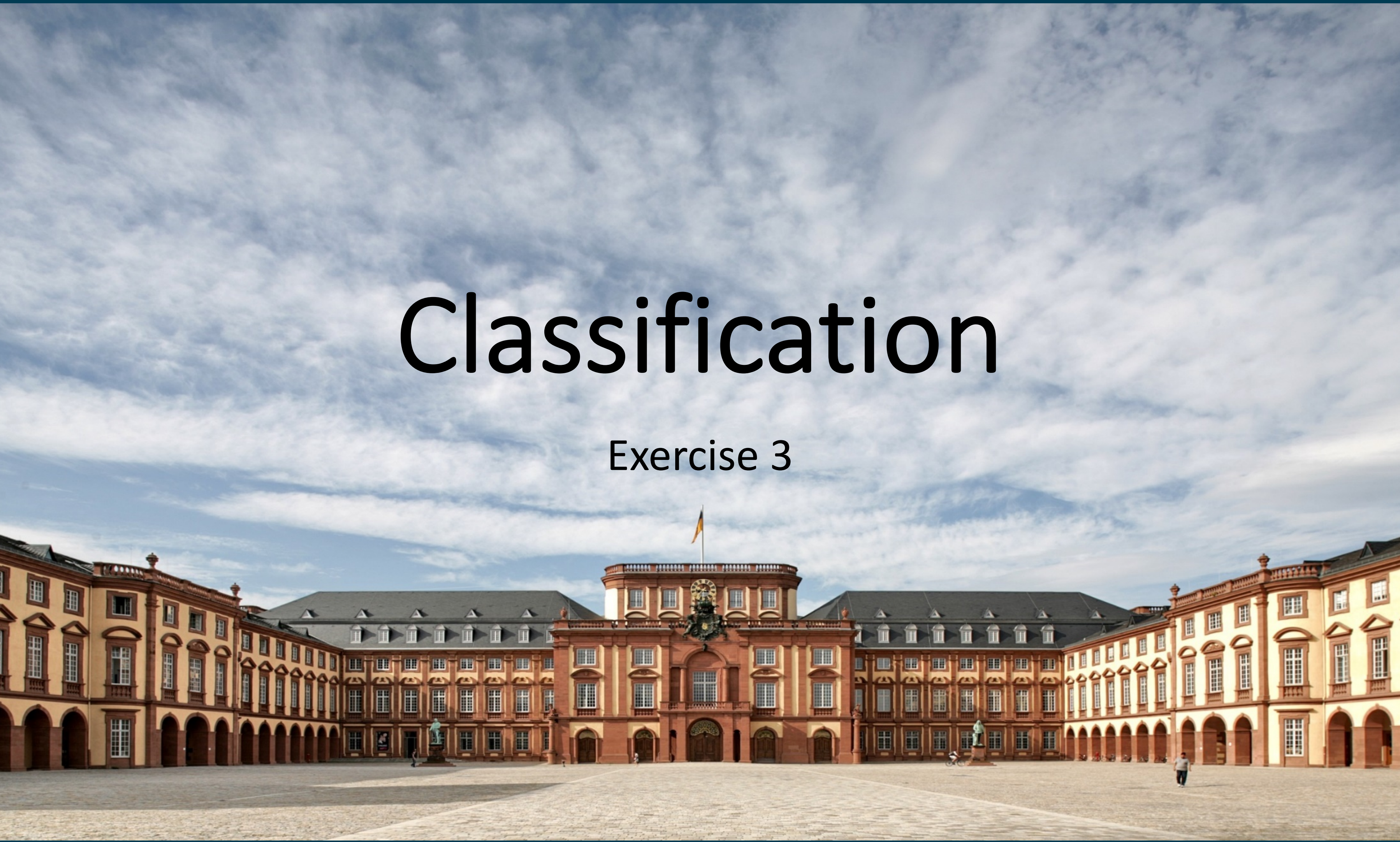
Classification

Exercise 3



Classification

"tree"

"tree"

"tree"

"not a tree"

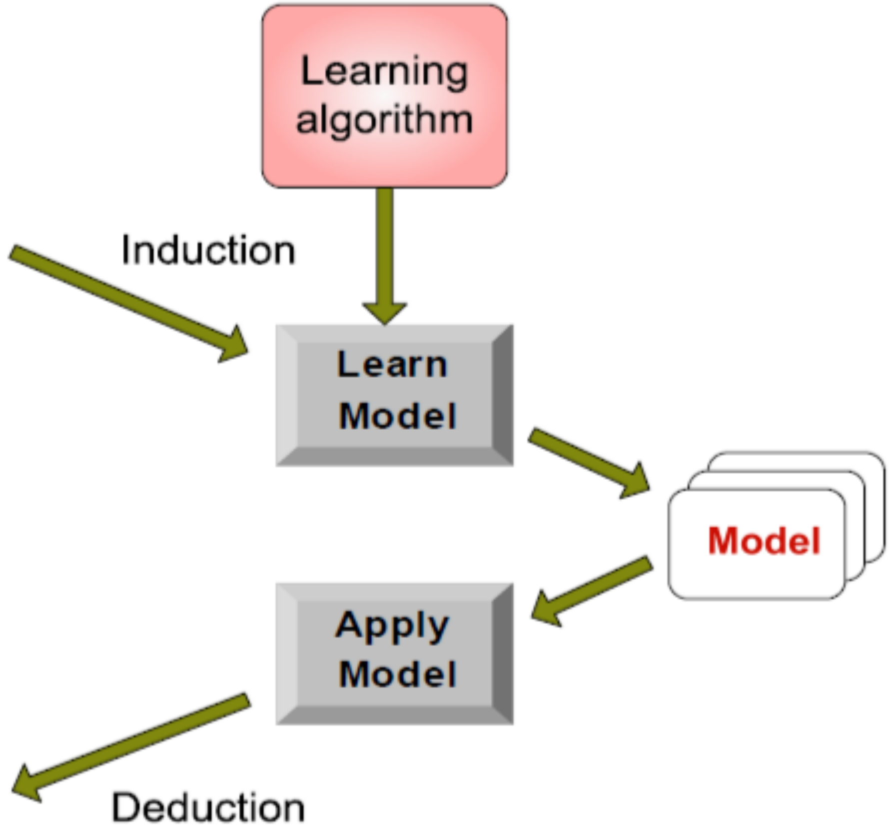
"not a tree"

"not a tree"

The Classification Workflow

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

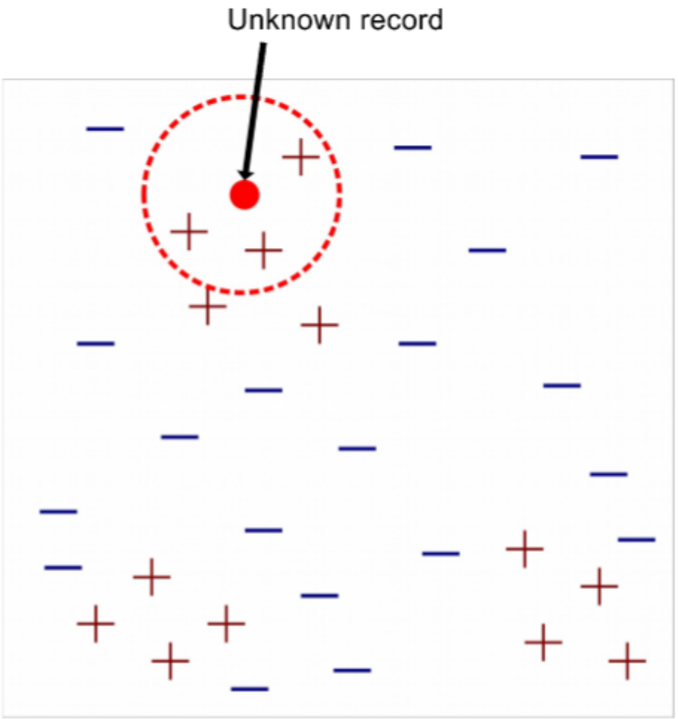
Unseen Records

K-Nearest-Neighbour

- Calculate the distance to all other points
- Choose the nearest K neighbours
- Let them vote for a class

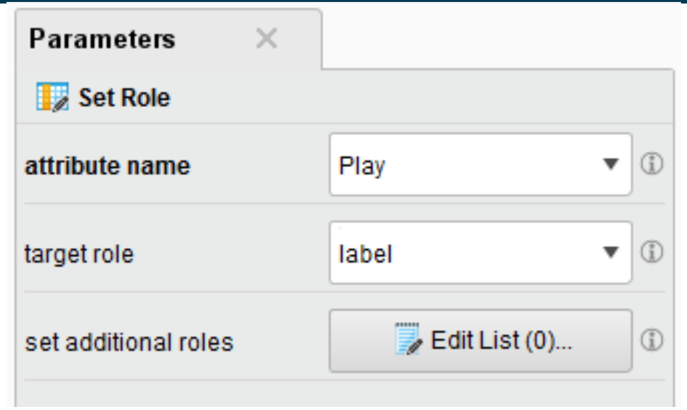
- Requires
  - All known records
  - Distance metric

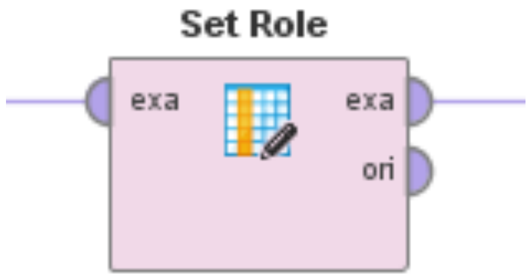
- Often very accurate
- But also slow



Operators: Set Role

- Input Port
  - Example Set
- Output Ports
  - Changed Example Set
  - Original Example Set
- Parameters
  - Attribute Name
  - Target Role

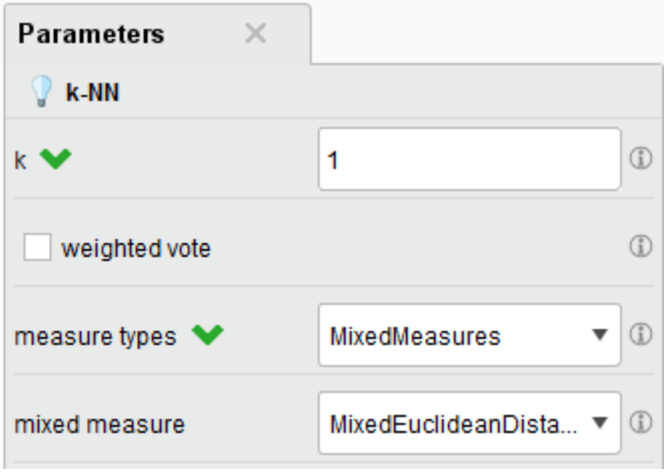


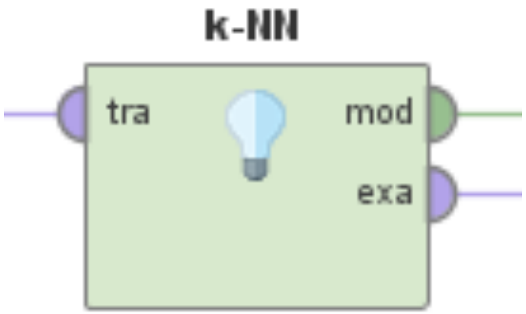


- Classification Operators need an attribute of type ‘label’

Operators: K-NN

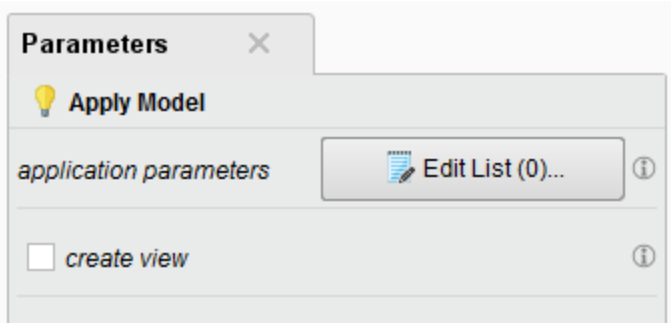
- Input Port:
  - Training data (Example Set)
- Output Ports
  - Classification Model
  - Training data (Example Set)
- Parameters
  - K
  - Weighted Vote
  - Similarity Measure

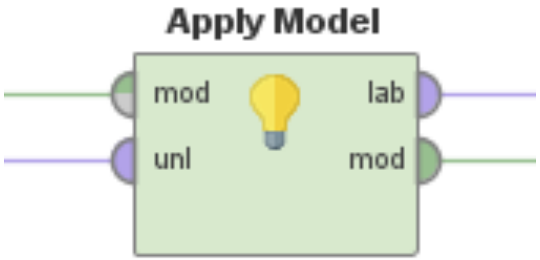




Operators: Apply Model

- Input Ports
  - Model
  - Unlabelled data (Example Set)
- Output Ports
  - Labelled data (Example Set)
  - Model

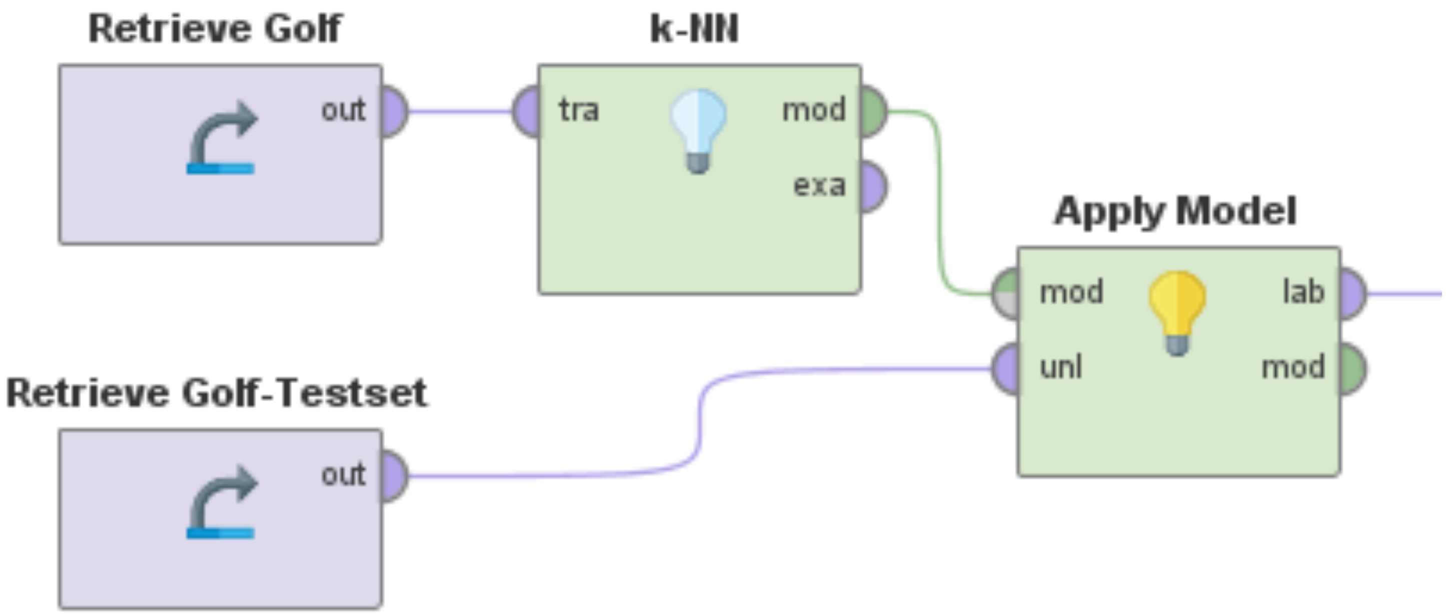




- Classification Operators do not apply the model they learn!
- You have to apply the model to ***a different*** example set

Process: Classification

- Learn the model from the training data
- Apply the model to the testing data
- Check the results



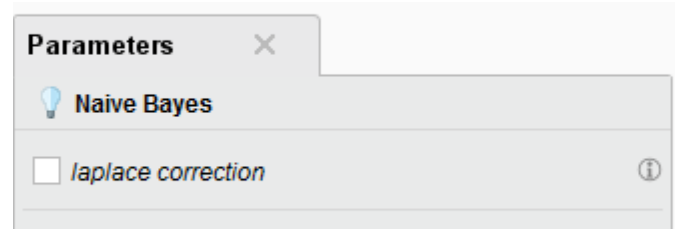
Naïve Bayes Classification

- 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

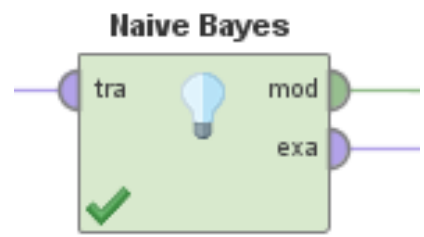
$$P(C|A)=\frac{P(A|C)P(C)}{P(A)}$$

- 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 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

Evaluation

- How do we know the model actually works?
  - By counting the number of errors
  - On a ***different*** dataset

Important!!!

- What's the purpose of a model?
  - To apply it to new data where we don't know the label

- What happened if we used the same dataset?
  - How many errors for a K-NN classifier with K=1?
  - How good would that model be on a different dataset?

Evaluation: Confusion Matrix

- For every class in our dataset, the classifier can produce one of four possible results:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	True Positive (TP)	False Negative (FN)
	Class=No	False Positive (FP)	True Negative (TN)

Evaluation Measures: Accuracy

- A single measure that tells you the overall accuracy of the result

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- "Number of correctly classified examples divided by the total number of examples."

- Problem: Unbalanced data
  - If 99% belong to class "yes"
  - And classifier always says "yes" – 99% Accuracy

Evaluation Measures: Precision and Recall

- Measure two aspects of the result for every class
- Precision: How many of the examples that were labelled "yes" are really "yes"?
  - "the number of correctly labelled examples divided by the number of all examples that were labelled with this class"
- Recall: How many of the examples that are really "yes" were labelled "yes'?
  - "the number of correctly labelled examples divided by the number of all examples that actually belong to this class"

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Evaluation Measures: Precision and Recall

- An example:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

ID	Prediction	Actual Class
1	Yes	Yes
2	Yes	No
3	No	No
4	Yes	No

- For class "yes"
  - 1 true positive (ID 1)
  - 2 false positives (ID 2 & 4)
  - 1 true negative (ID 3)

$$Precision_{yes} = \frac{1}{1+2} = \frac{1}{3}$$

$$Recall_{yes} = \frac{1}{1+0} = \frac{1}{1}$$

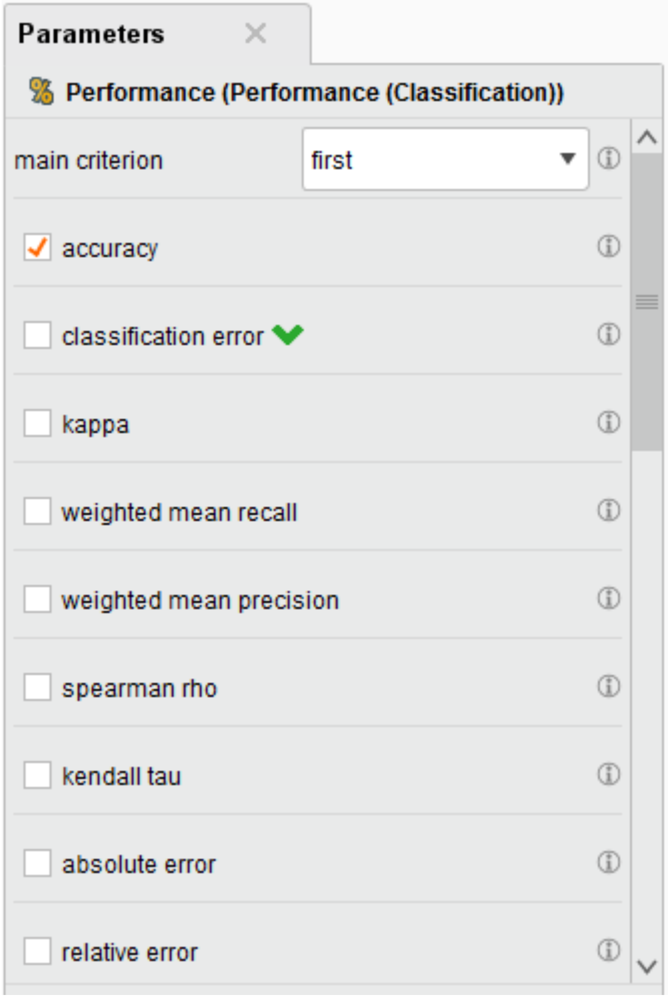
- For class "no"
  - 1 true negative (ID 1)
  - 2 false negatives (ID 2 & 4)
  - 1 true positive (ID 3)

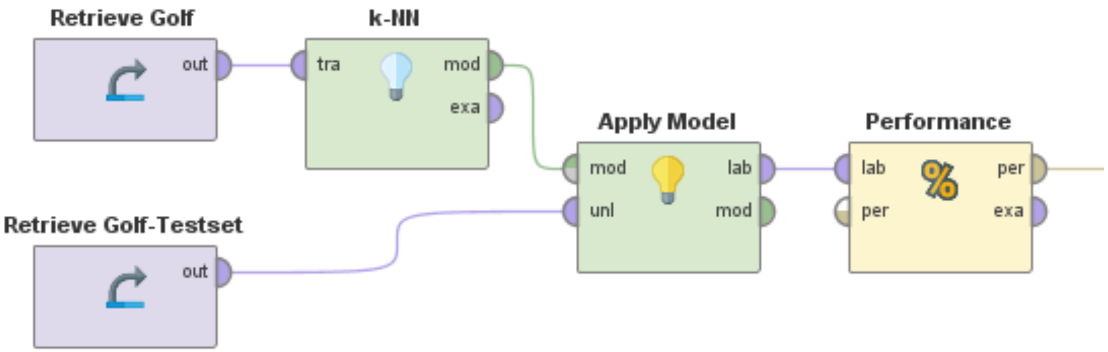
$$Precision_{no} = \frac{1}{1+0} = \frac{1}{1}$$

$$Recall_{no} = \frac{1}{1+2} = \frac{1}{3}$$

Operators: Performance (Classification)

- Input
  - Labelled Example Set
- Output
  - Performance
- Parameters
  - Performance Measures





Split-Validation / Cross-Validation

- What can you do if you only have one dataset?
  - Use one part of the data for training
  - Use *the other part* of the data for testing
- What if by accident all the easy examples are in the training set?
  - Then your model will not perform that good
  - Better to repeat the learning on different splits of the data
- X-Validation (Cross-Validation)
  - Split the dataset into X parts
  - Select one part for testing, use the rest for training
  - Repeat this until every part was used for training once

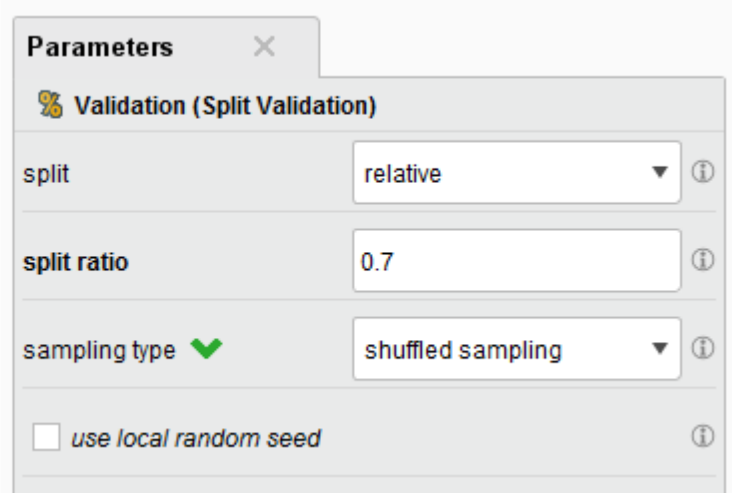
Just a reminder …

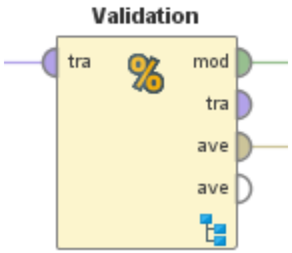
- If you use the same data for training and evaluation…

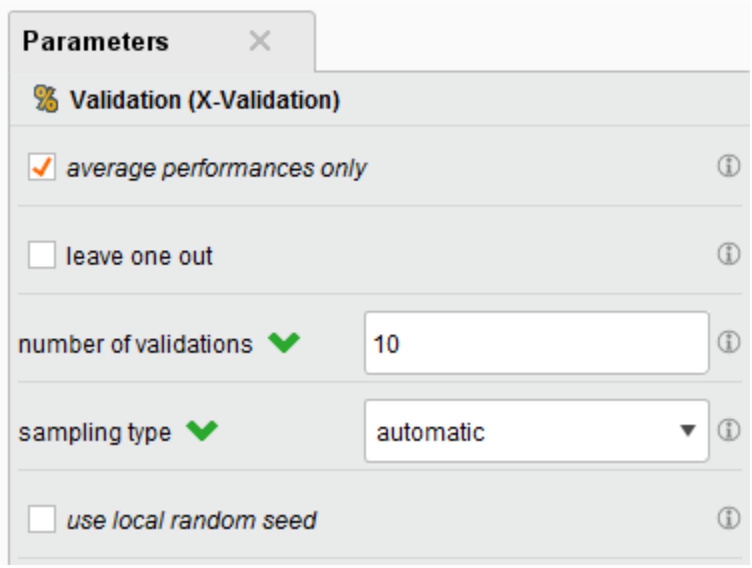
- … there will be no Christmas!

Operators: Split Validation / X-Validation

- Input Port
  - Training data (Example Set)
- Output Ports
  - Classification Model
  - Training data (Example Set)
  - Averageable 1 … n
- Parameters
  - Split type
  - Split ratio
  - Sampling type

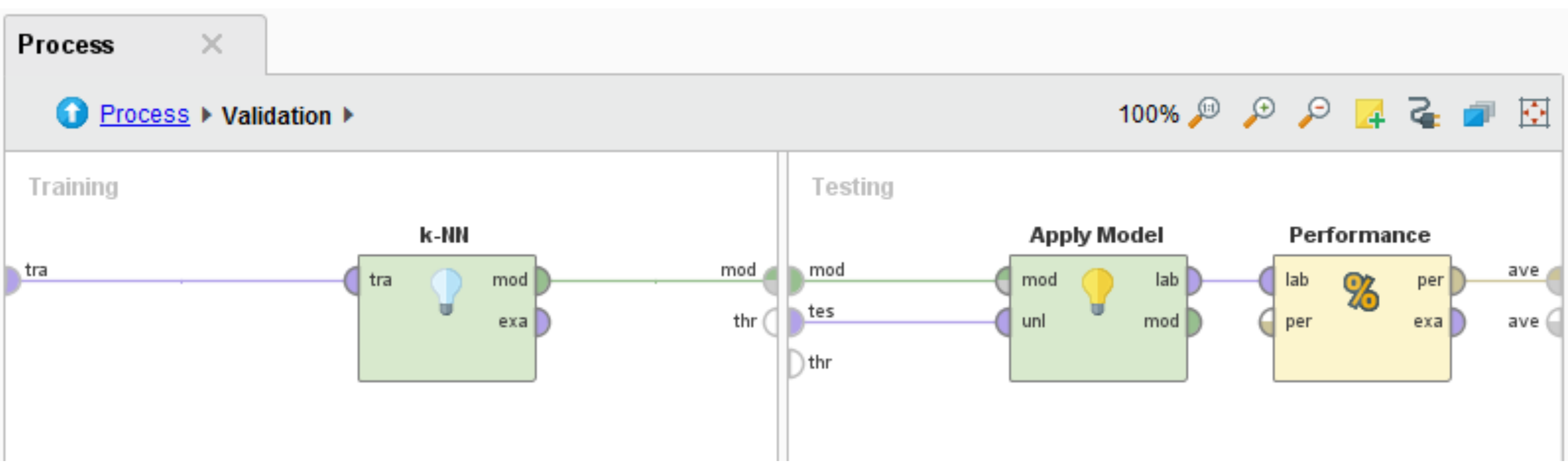




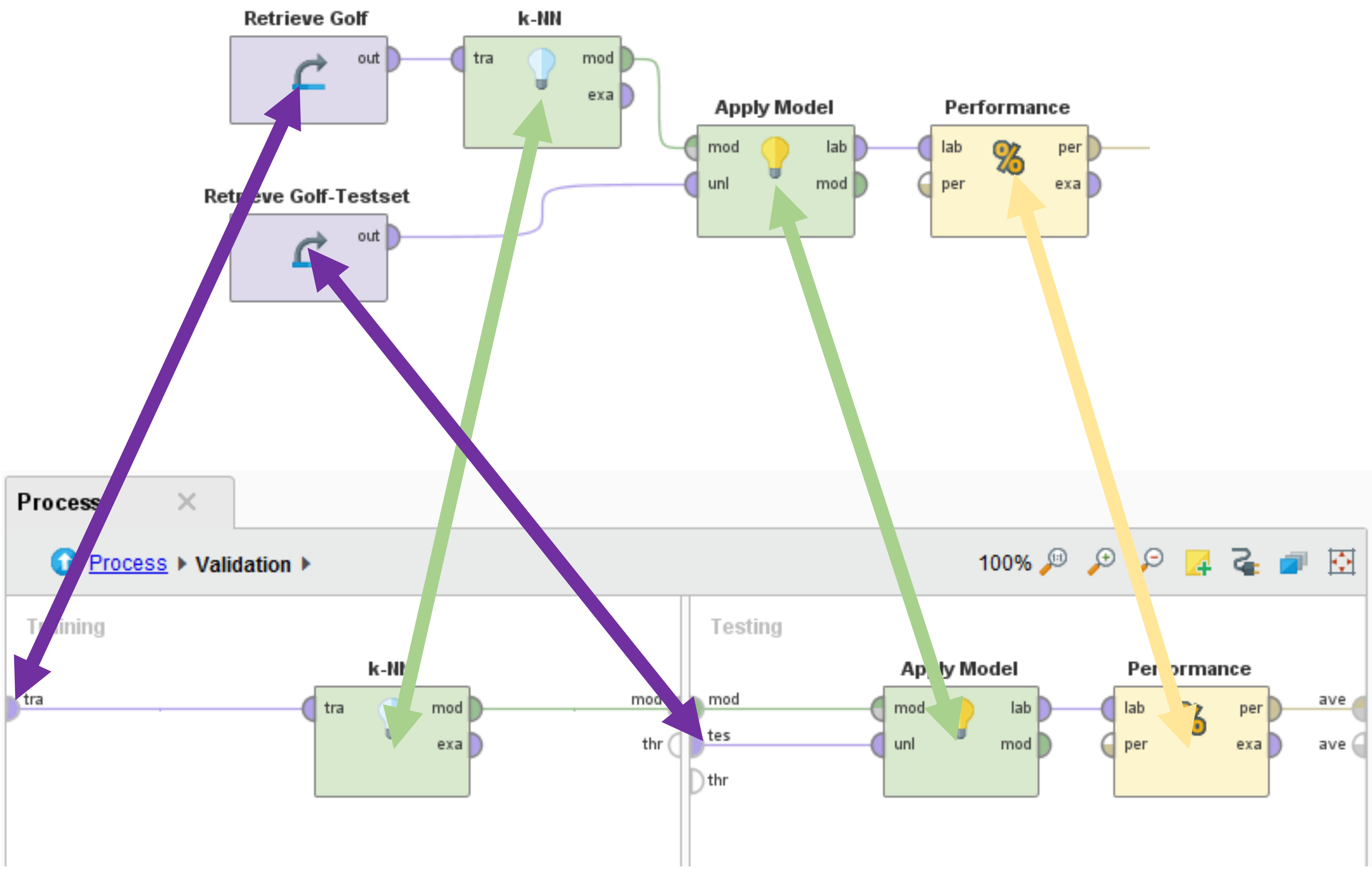


Nested Processes in Rapid Miner

- Operators can have "inner" processes that define their behaviour
- Split/X-Validation Operators have a "Training" and a "Testing" phase
  - Training: This is where you learn your model
  - Testing: This is where you evaluate



From two datasets to Split-Validation



The Mannheim RapidMiner Toolbox

- A Rapid Miner Extension with many great operators
- Developed by researchers from the Data and Web Science Group
- Contains the nearest centroid classifier