Data Mining
(Mining Knowledge from Data)

K-Nearest Neighbors

Magda Friedjungová, Marcel Jiřina

Models

- Deductive or <u>inductive</u>
- Methods/tasks of learning inductive models:
  - clasification
  - regression
  - prediction
  - clustering

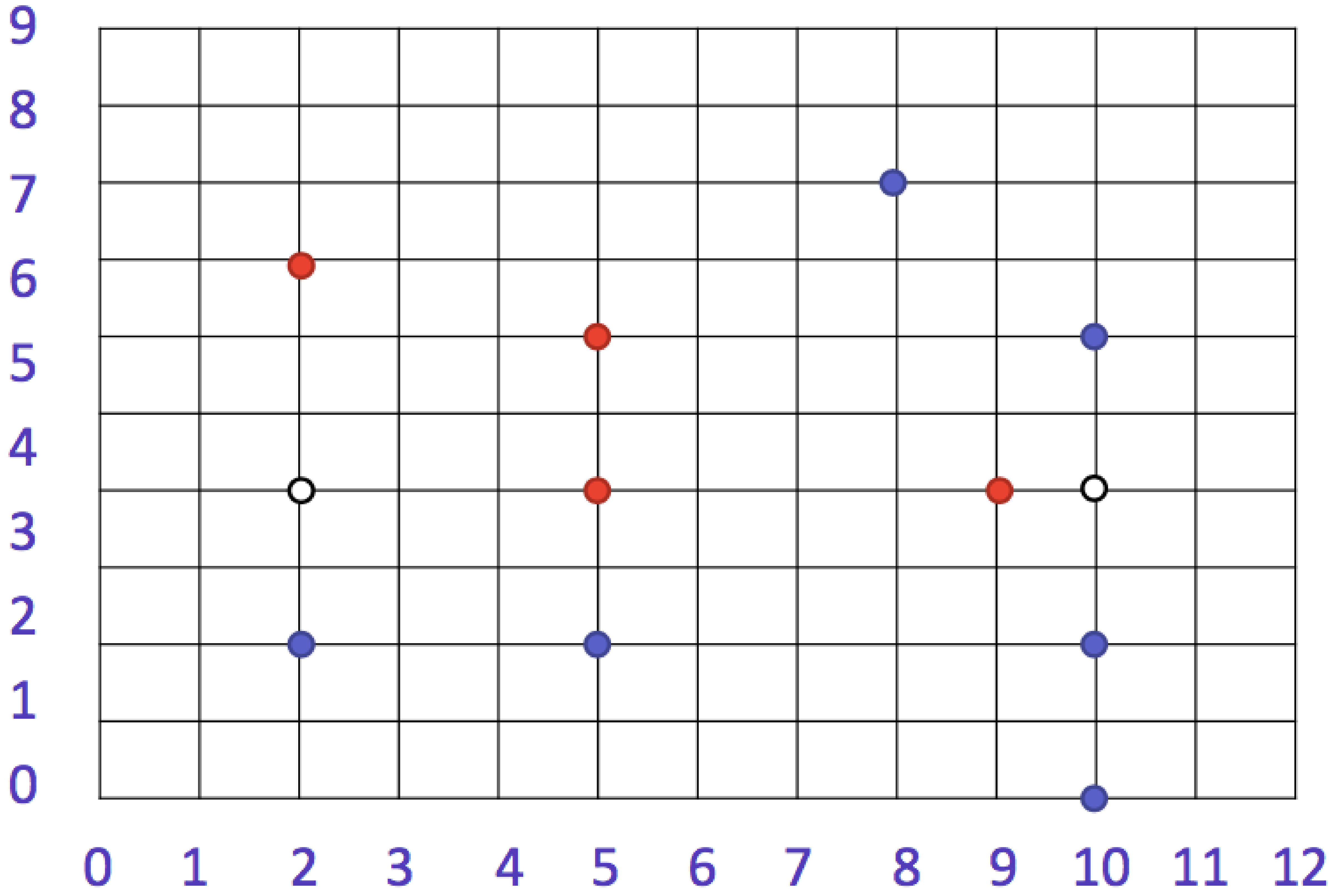
Overview of methods

Task	Algorithms
Classification	K-Nearest Neighbors, Linear Separation, Decision Trees, Bayes Classifier, Neural Networks
Regression, forecasting	K-Nearest Neighbors, Linear Regression, Regression Trees, Neural Networks
Clustering	K-means, Hierarchical Clustering
Rules, associations etc.	Association Rules, K-means

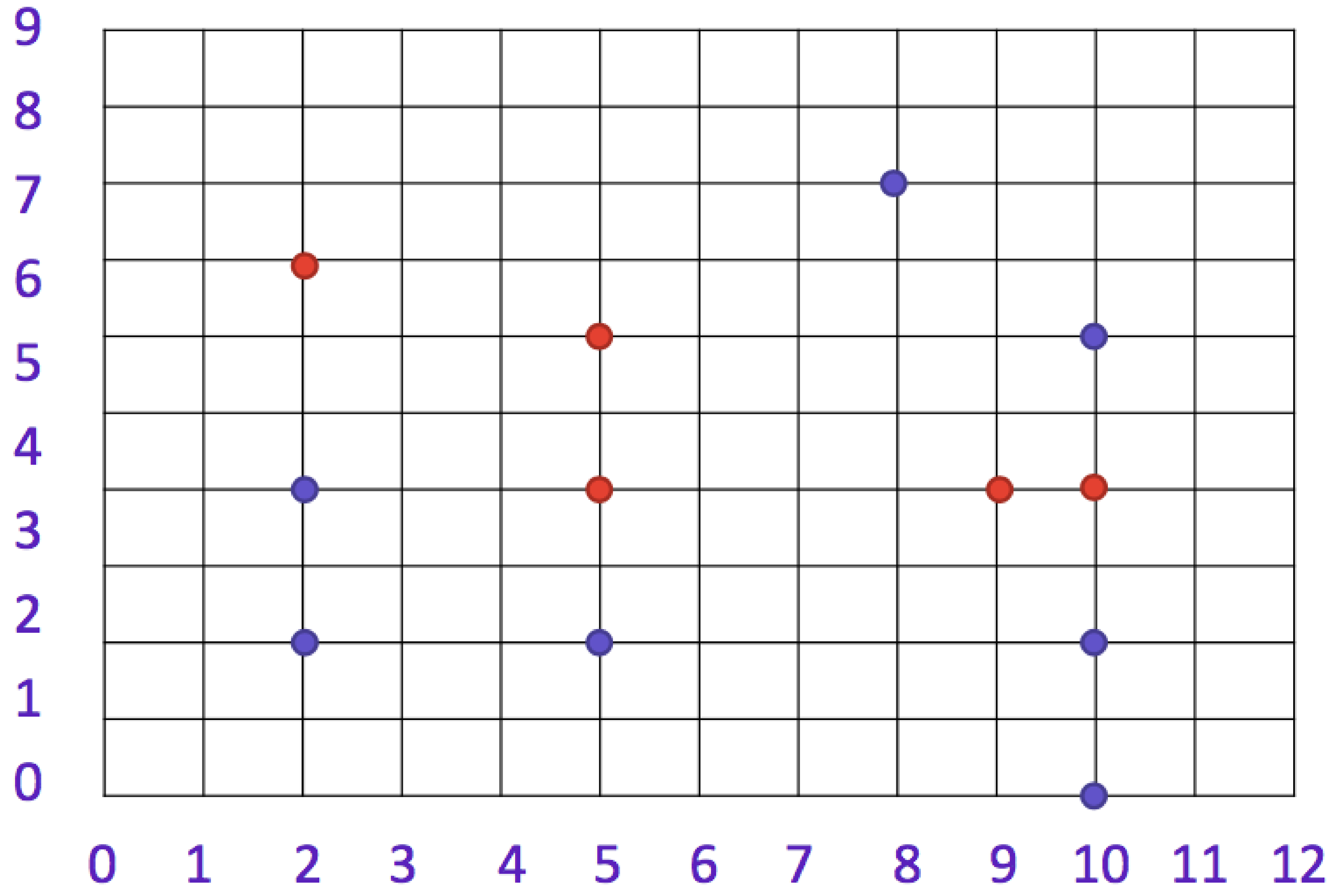
Creating and using model

- Two stages:
  - Stage of learning, training
  - Stage of use, equipping

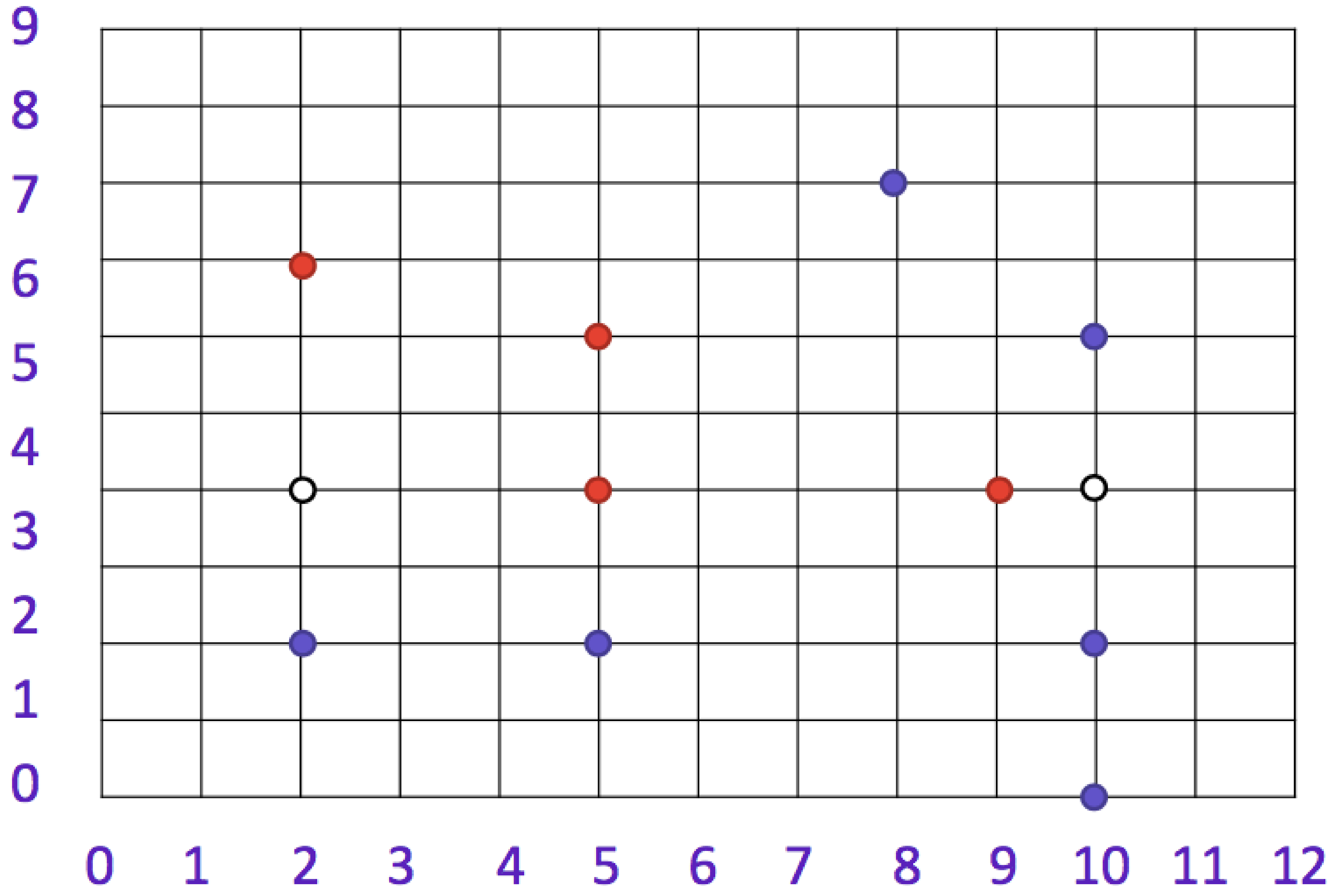
Determine the class by the 1-NN method



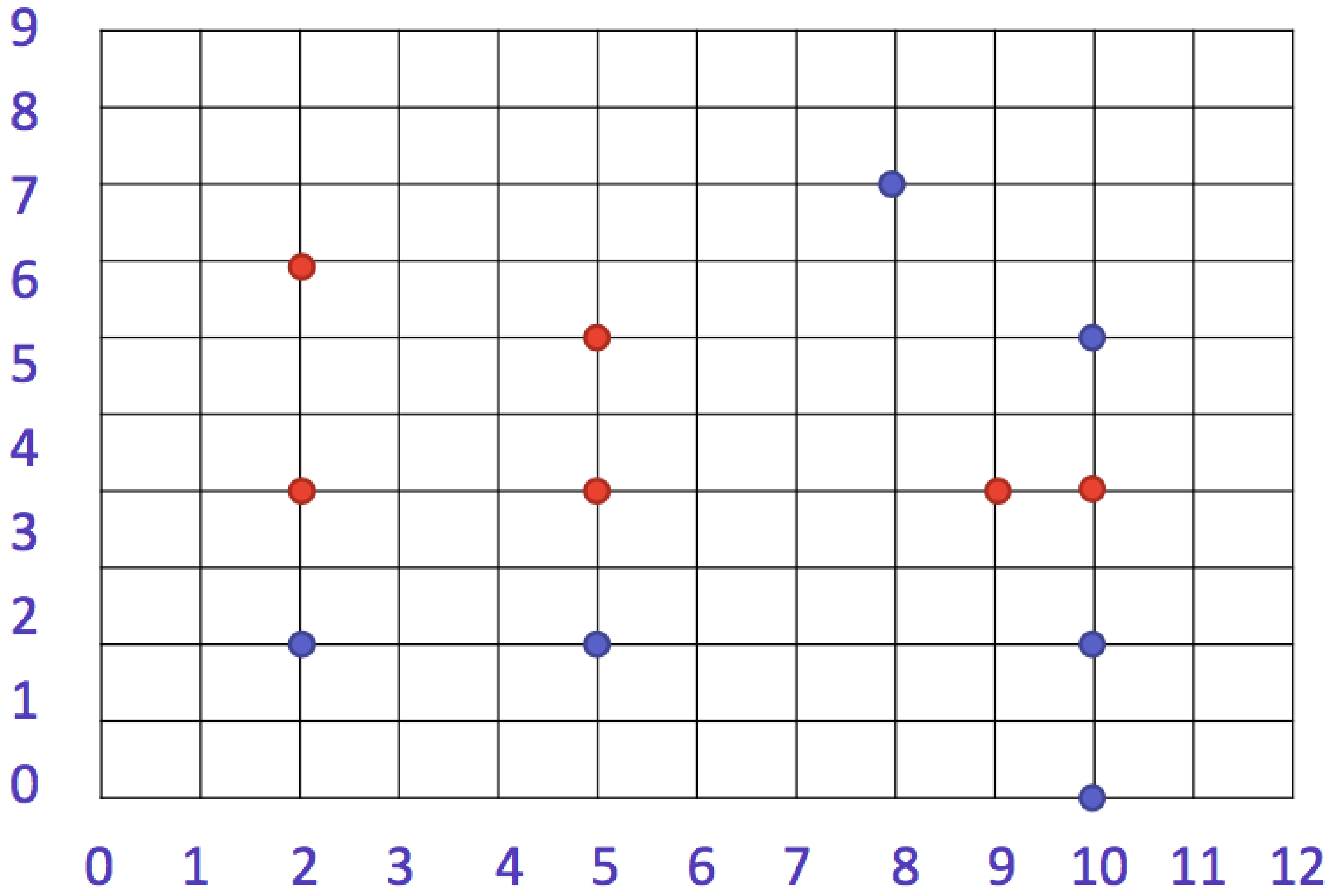
Result



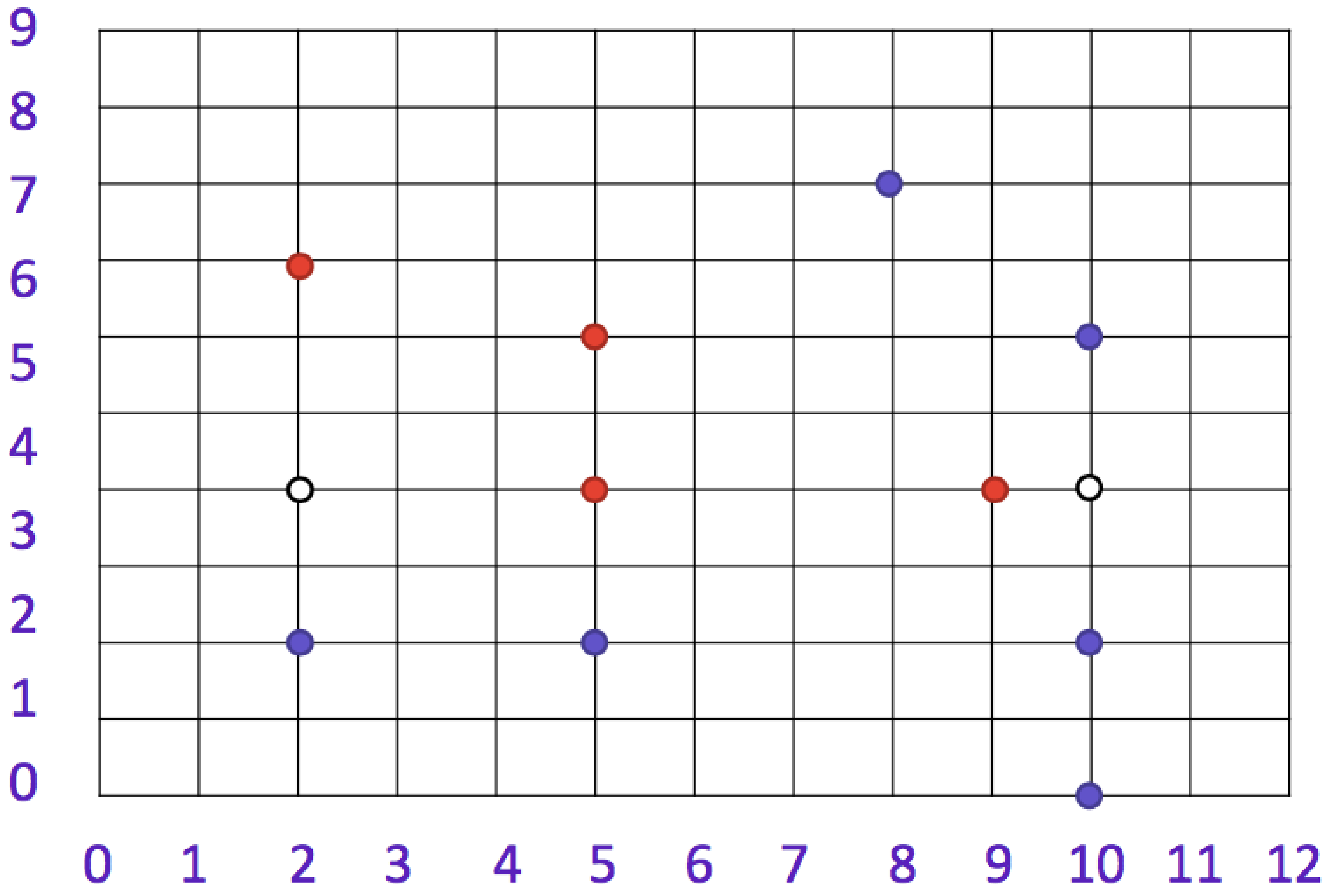
Determine the class by the 2-NN method



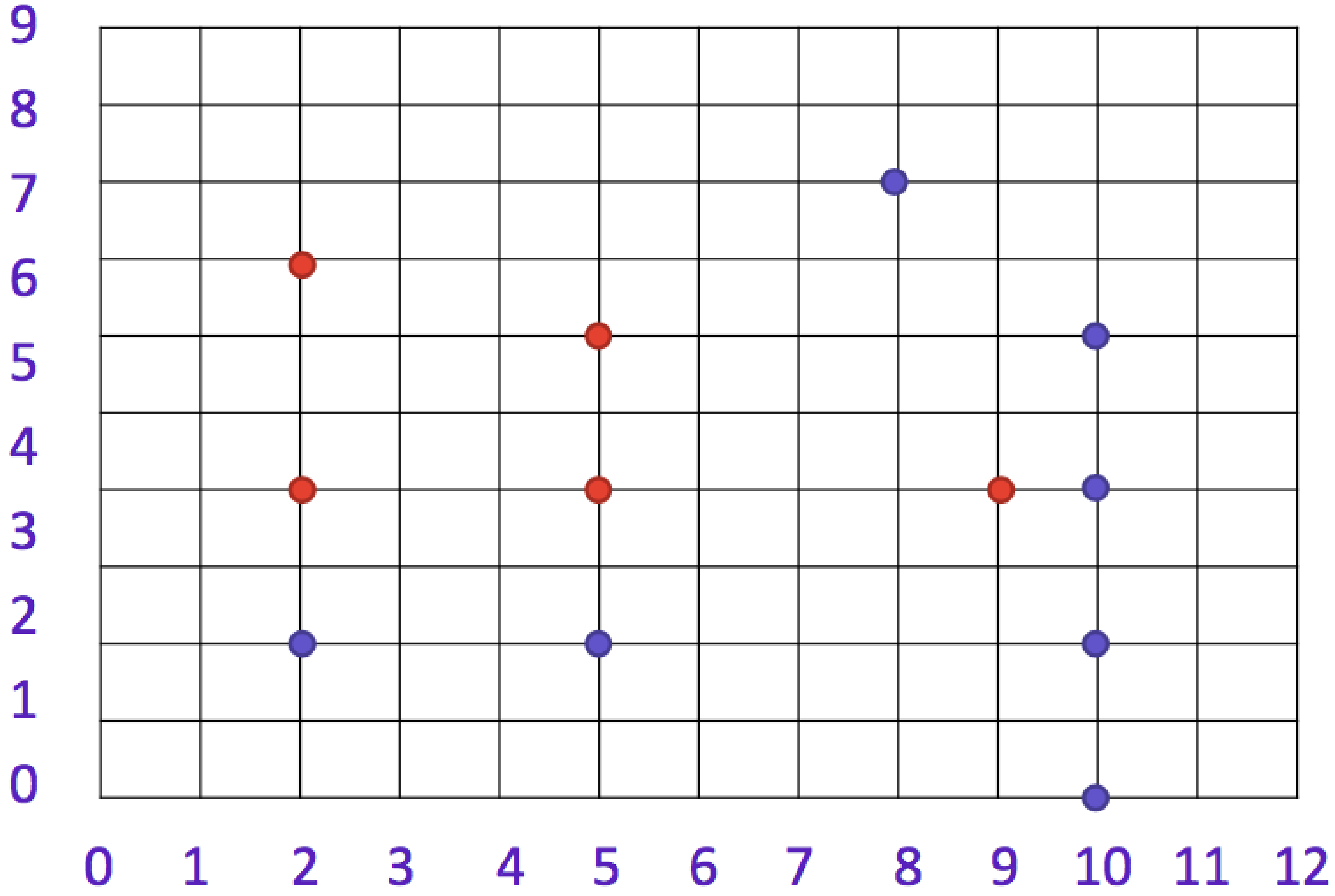
Result



Determine the class by the 3-NN method



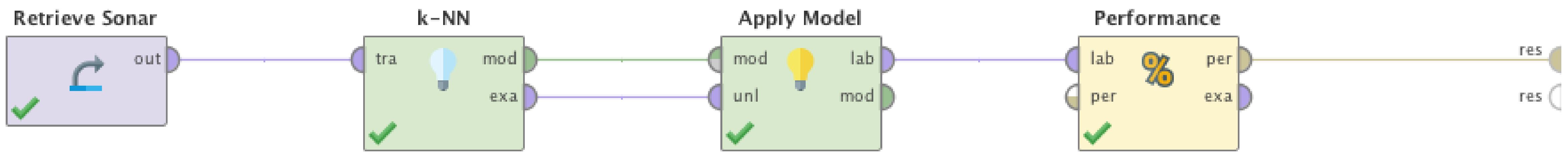
Result



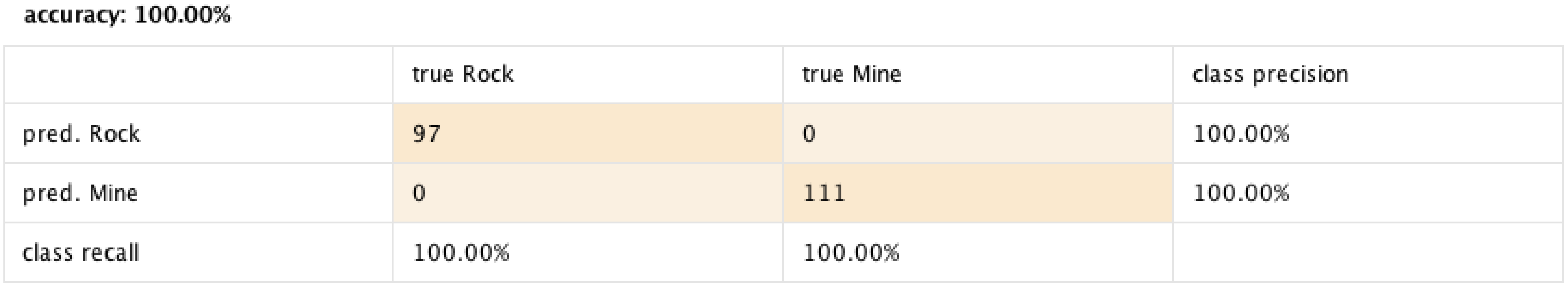
K-NN on Sonar Data 1/3

- Load the Sonar dataset.
- Learn the k-NN classifier.
- Apply the model on the training data.
- Visualize the output and confidence of the model.
- Calculate the performance on the training data.

Sonar 1/3



- The accuracy of classification: 100%

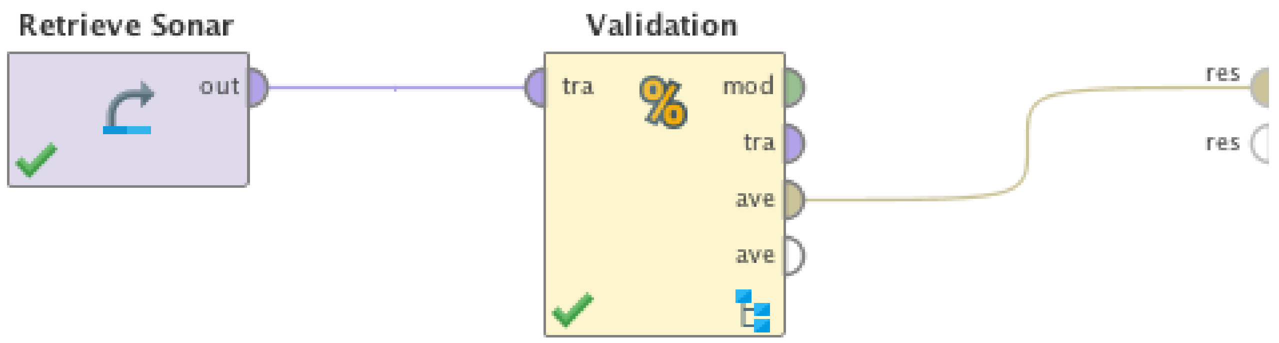


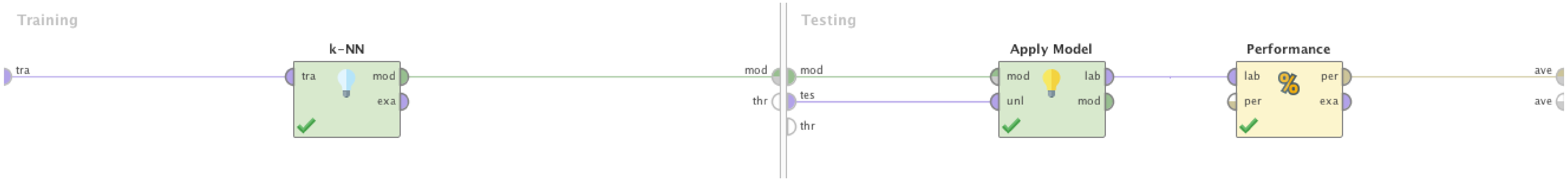
accuracy: 100.00%

	true Rock	true Mine	class precision
pred. Rock	97	0	100.00%
pred. Mine	0	111	100.00%
class recall	100.00%	100.00%	

Sonar 2/3

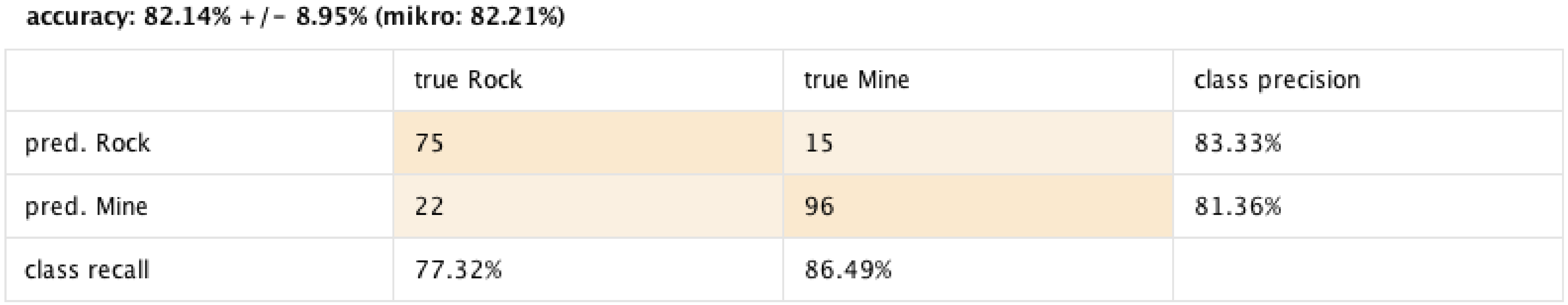
- Calculate generalized performance (X-validation).





Sonar 2/3

- The accuracy of x-validation: 82%



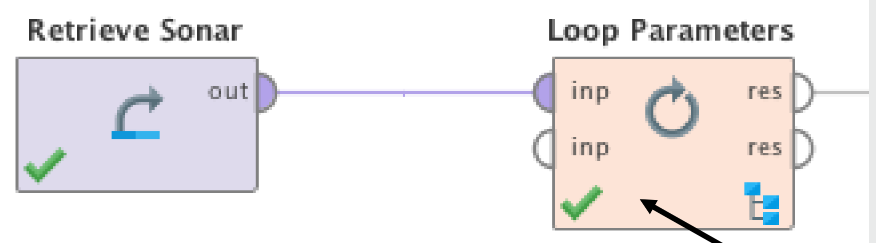
accuracy: 82.14% +/− 8.95% (mikro: 82.21%)

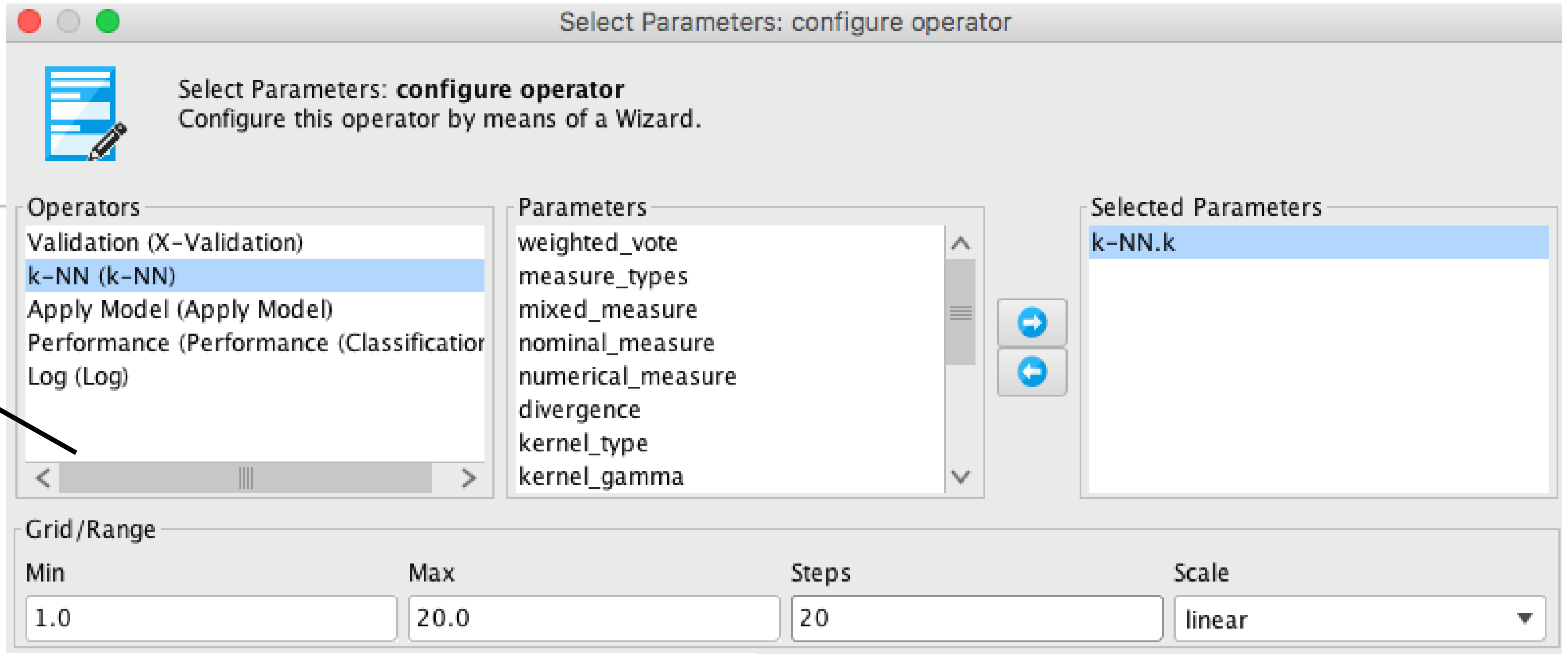
	true Rock	true Mine	class precision
pred. Rock	75	15	83.33%
pred. Mine	22	96	81.36%
class recall	77.32%	86.49%	

Sonar 3/3

- Measure the classification accuracy for k={1, 2, 3, …, 20} and draw the plot.
- Explain the observation.

Sonar 3/3



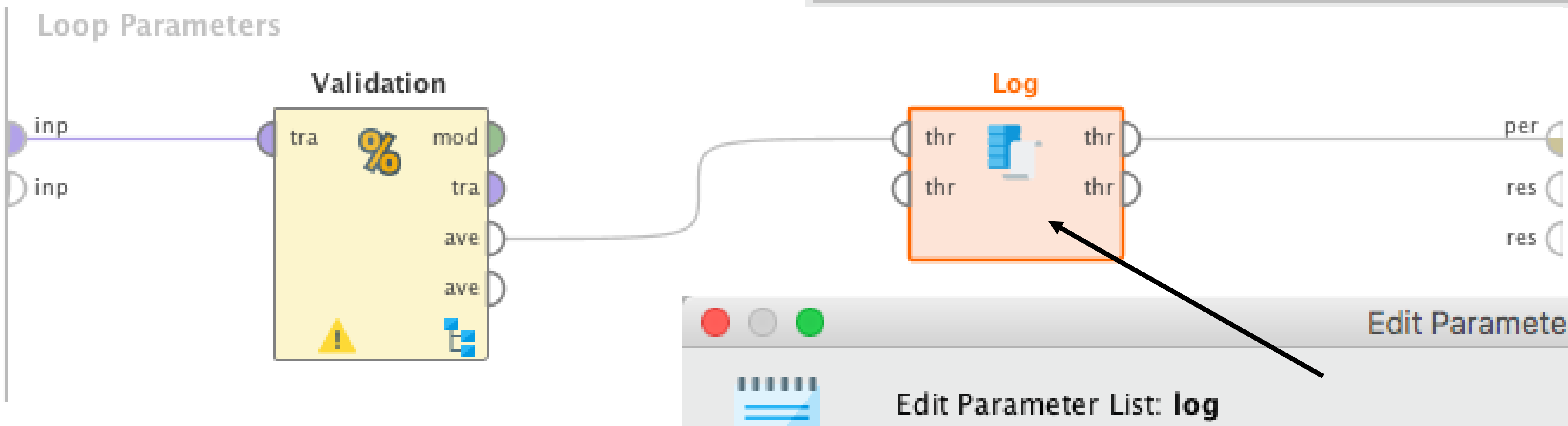

Select Parameters: configure operator

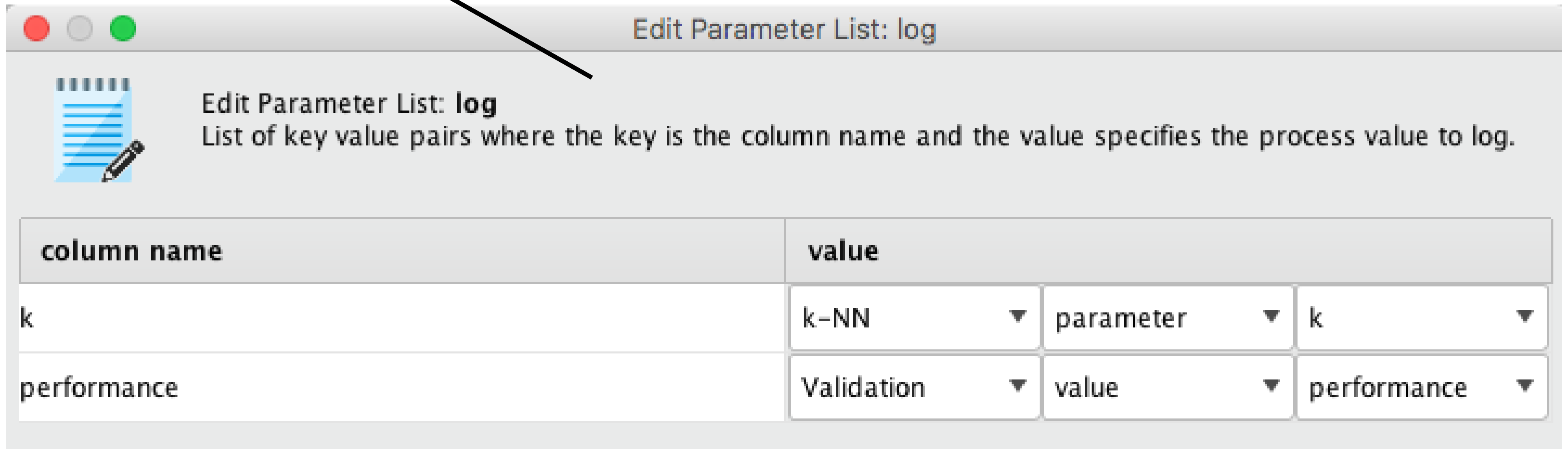
Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators	Parameters	Selected Parameters
Validation (X-Validation)	weighted_vote	k-NN.k
k-NN (k-NN)	measure_types	
Apply Model (Apply Model)	mixed_measure	
Performance (Performance (Classificatior	nominal_measure	
Log (Log)	numerical_measure	
	divergence	
	kernel_type	
	kernel_gamma	

Grid/Range

Min	Max	Steps	Scale
1.0	20.0	20	linear




Edit Parameter List: log

Edit Parameter List: **log**
List of key value pairs where the key is the column name and the value specifies the process value to log.

column name	value		
k	k-NN	parameter	k
performance	Validation	value	performance

Sonar 3/3

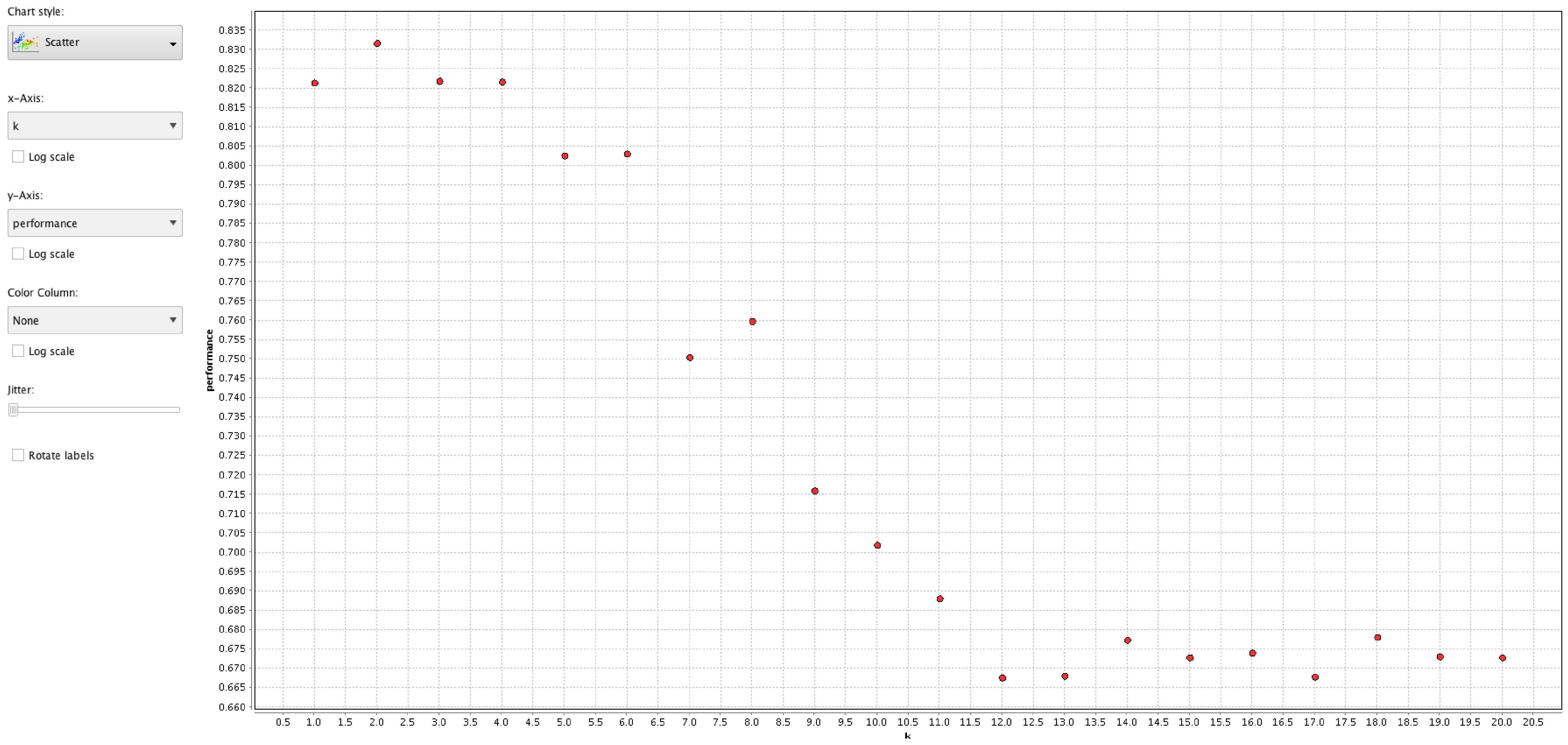


Chart style: Scatter

x-Axis: k

Log scale

y-Axis: performance

Log scale

Color Column: None

Log scale

Jitter:

Rotate labels

Lazy Modeling

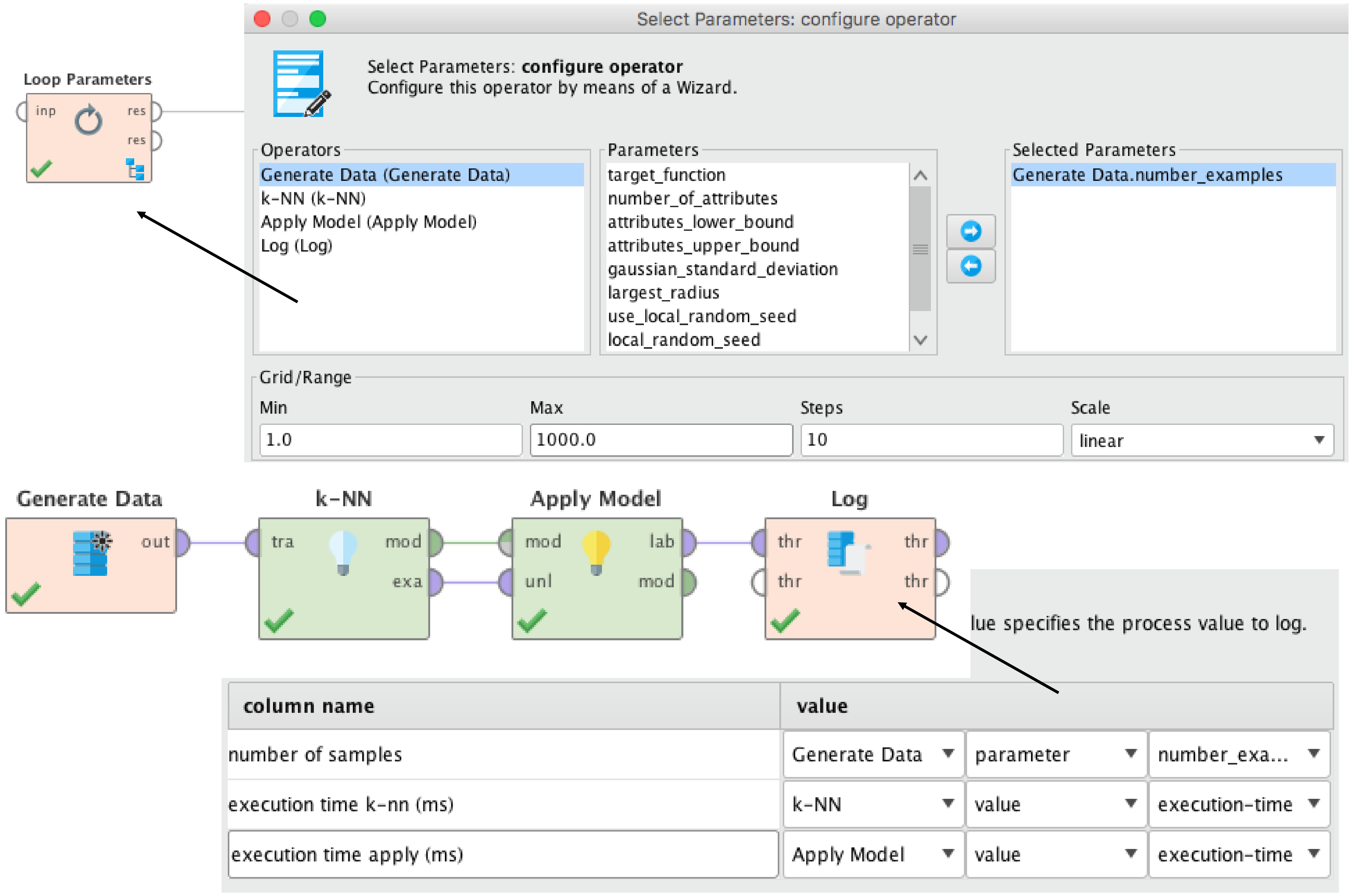
- What is lazy modeling?
  - Processing of training data take place just after receiving tha sample to be classified.
  - Response to the request by a combination of queries on training data.
  - Discard the calculated response and all intermediate results.

    What does this mean in practice?

Measure “execution time” of “k-NN” and “apply”

- Start from the previous schema:
  - Remove the x-validation.
  - Instead of the Sonar data use the Generate data. Target function set to “checkerboard classification”, number of attributes to 2.
  - In the Loop set “Generate Data.number_examples” on 1…1000 and 10 steps.

Measure “execution time” of “k-NN” and “apply”



Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators	Parameters	Selected Parameters
Generate Data (Generate Data)	target_function	Generate Data.number_examples
k-NN (k-NN)	number_of_attributes	
Apply Model (Apply Model)	attributes_lower_bound	
Log (Log)	attributes_upper_bound	
	gaussian_standard_deviation	
	largest_radius	
	use_local_random_seed	
	local_random_seed	

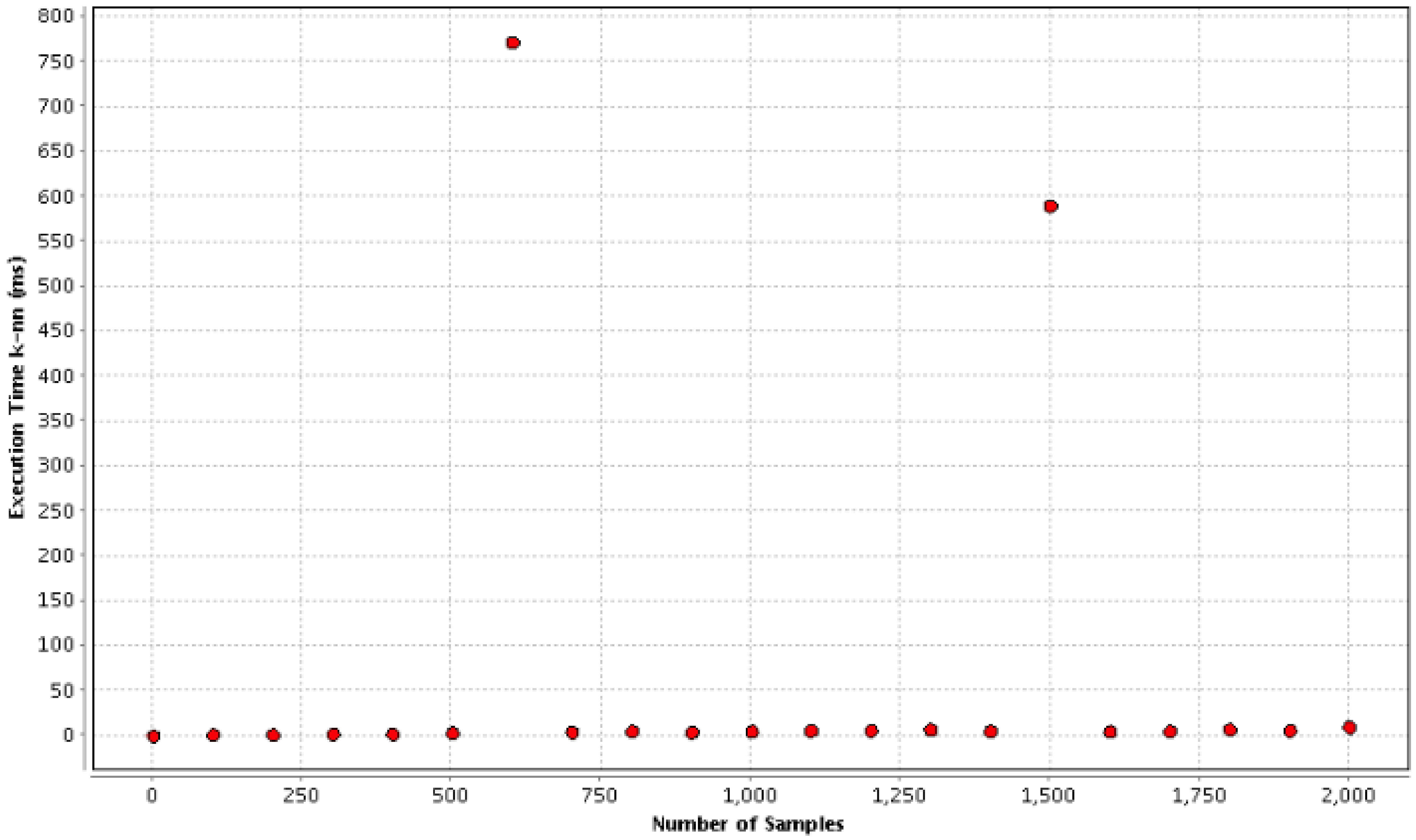
Grid/Range

Min	Max	Steps	Scale
1.0	1000.0	10	linear

lue specifies the process value to log.

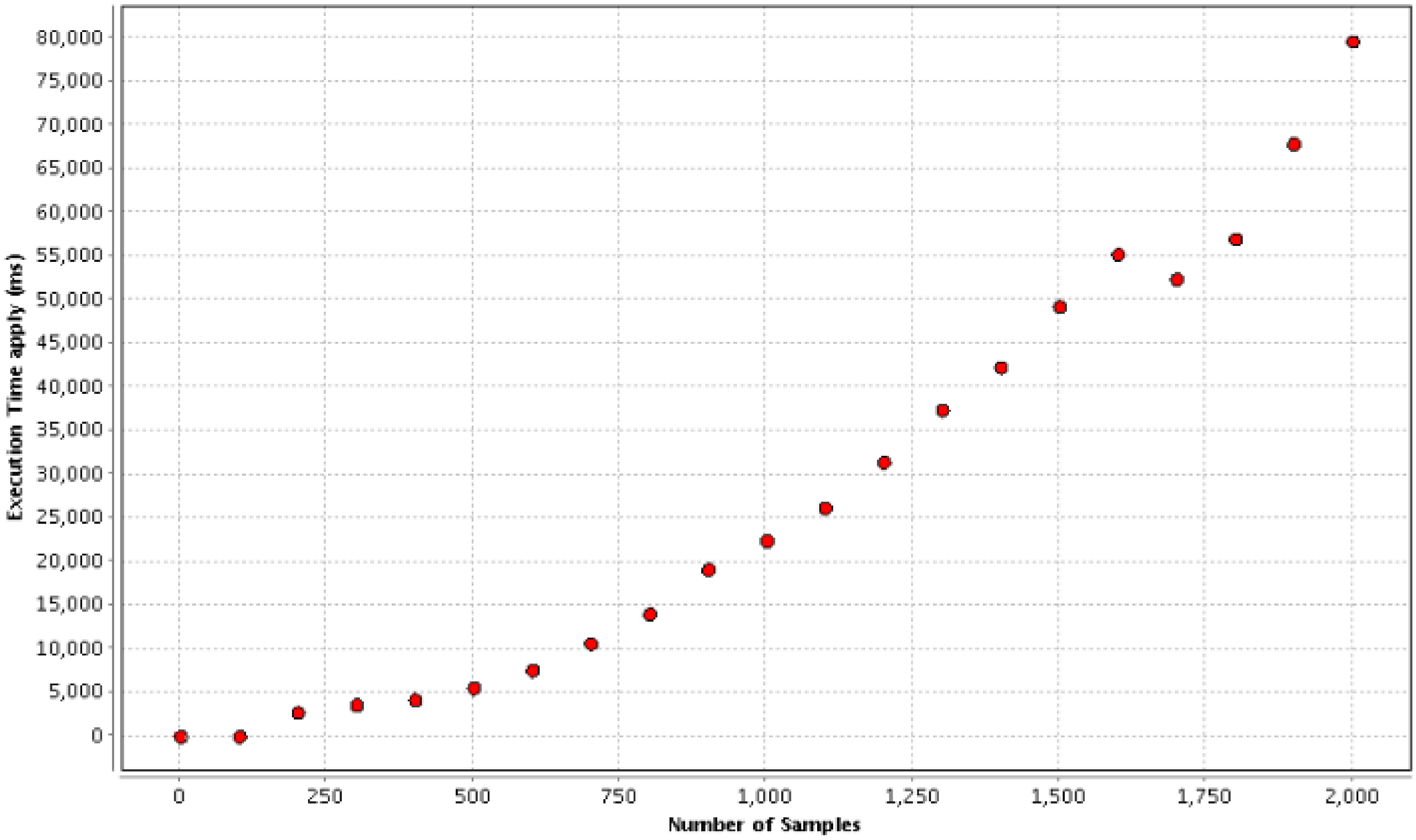
column name	value		
number of samples	Generate Data	parameter	number_exa...
execution time k-nn (ms)	k-NN	value	execution-time
execution time apply (ms)	Apply Model	value	execution-time

Measure “execution time” of “k-NN” and “apply”



The execution time is for k-NN approximately constant.

Measure “execution time” of “k-NN” and “apply”



The execution time for Apply is approximately quadratic.

Normalization

- k-NN method uses the Euclidean metric:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

- What if all dimensions ranges of values are 0 and 1, but only in one dimension the range is between 0 and 1000?

- How k-NN will behave?

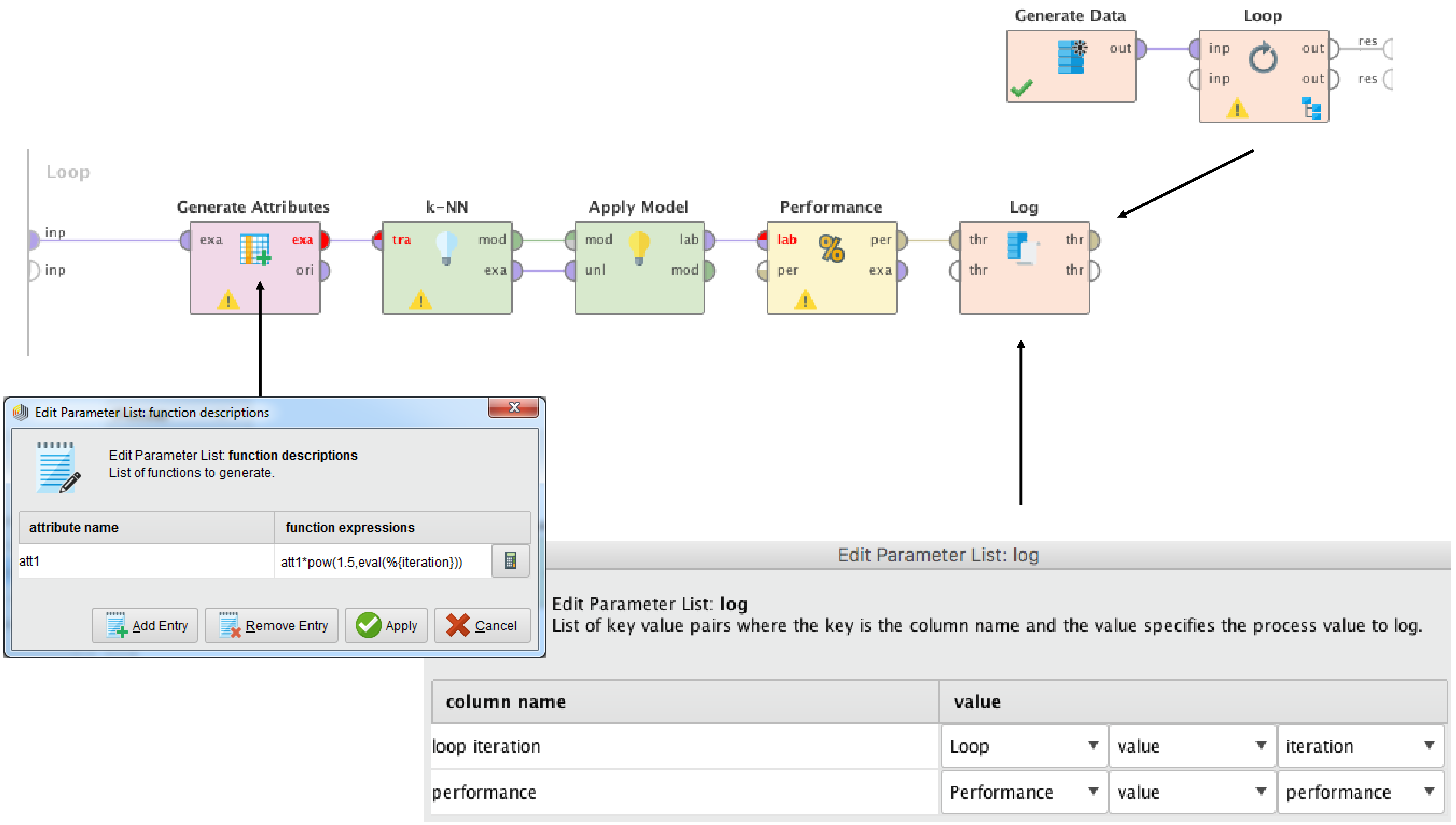
Normalization

- Use the schema from the previous task. As not all parameters can be changed in the “Loop Parameter” use “Loop” instead.
  - The size of a dimension can be changed by the block “Generate Attributes”. Set it to:

    “att1*pow(1.5,eval(%{iteration}))”

  - Set the logging properly.
  - Place the data generator outside the Loop (we want to compare the same data, but modified other ways).

Normalization



Edit Parameter List: function descriptions

Edit Parameter List: **function descriptions**
List of functions to generate.

attribute name	function expressions
att1	att1*pow(1.5,eval(%{iteration}))

Edit Parameter List: log

Edit Parameter List: **log**
List of key value pairs where the key is the column name and the value specifies the process value to log.

column name	value		
loop iteration	Loop	value	iteration
performance	Performance	value	performance

Normalization

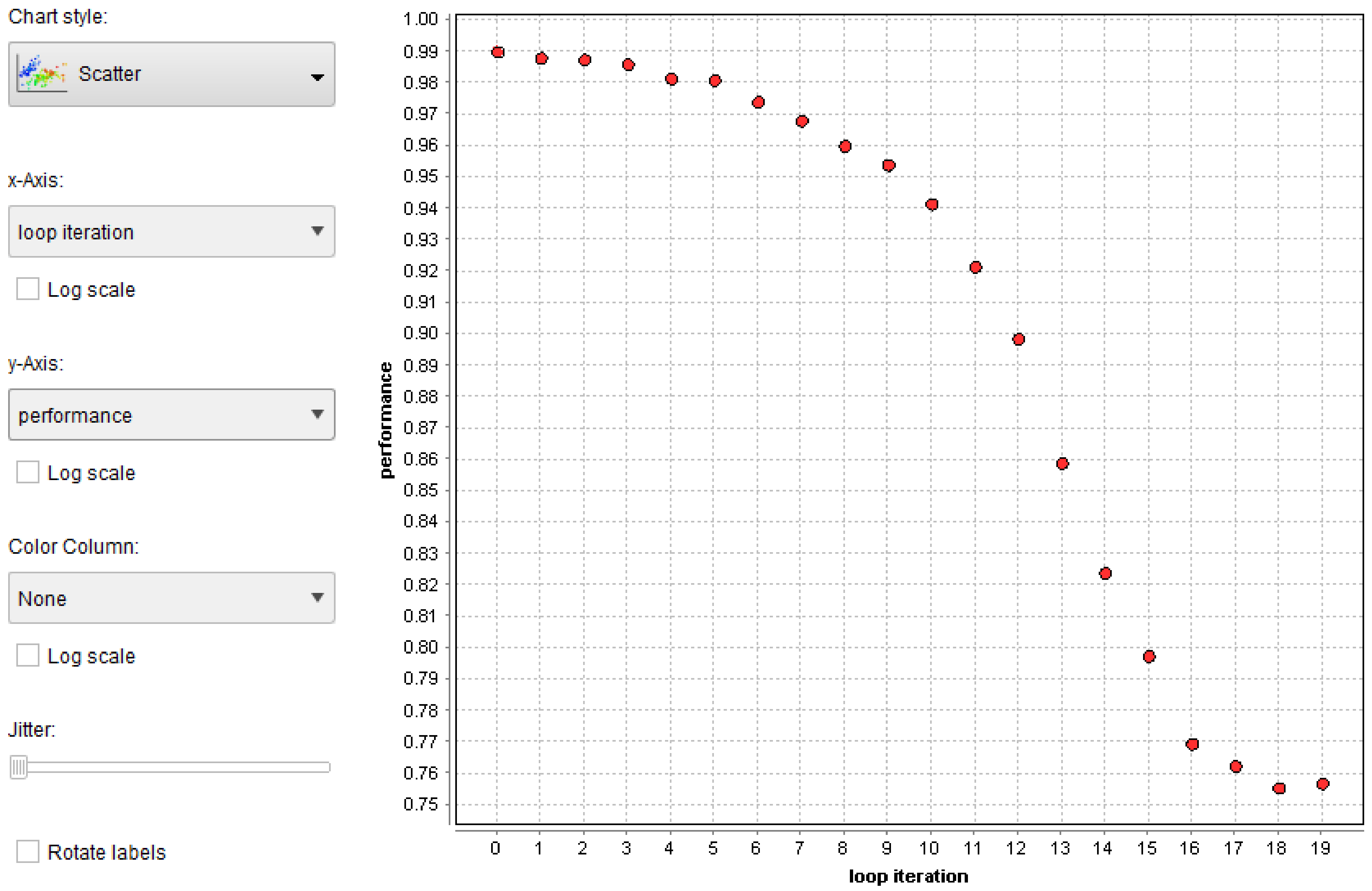


Chart style: Scatter

x-Axis: loop iteration

Log scale [ ]

y-Axis: performance

Log scale [ ]

Color Column: None

Log scale [ ]

Jitter:

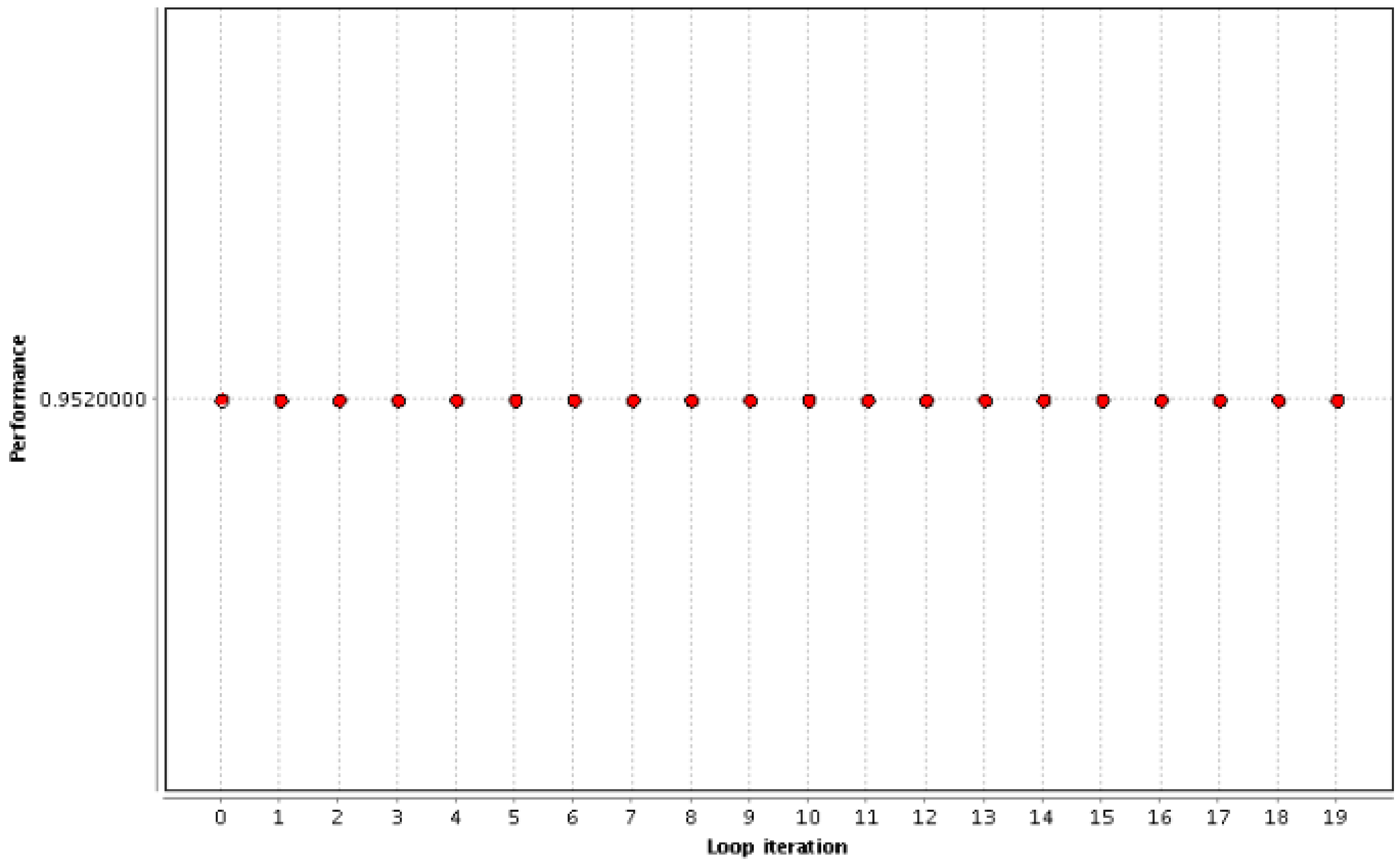
Rotate labels [ ]

- k-NN favors parameters of a large scale.

Normalization

- Repeat the experiment, but insert the “Normalize” before the “k-NN”.
- What happens?

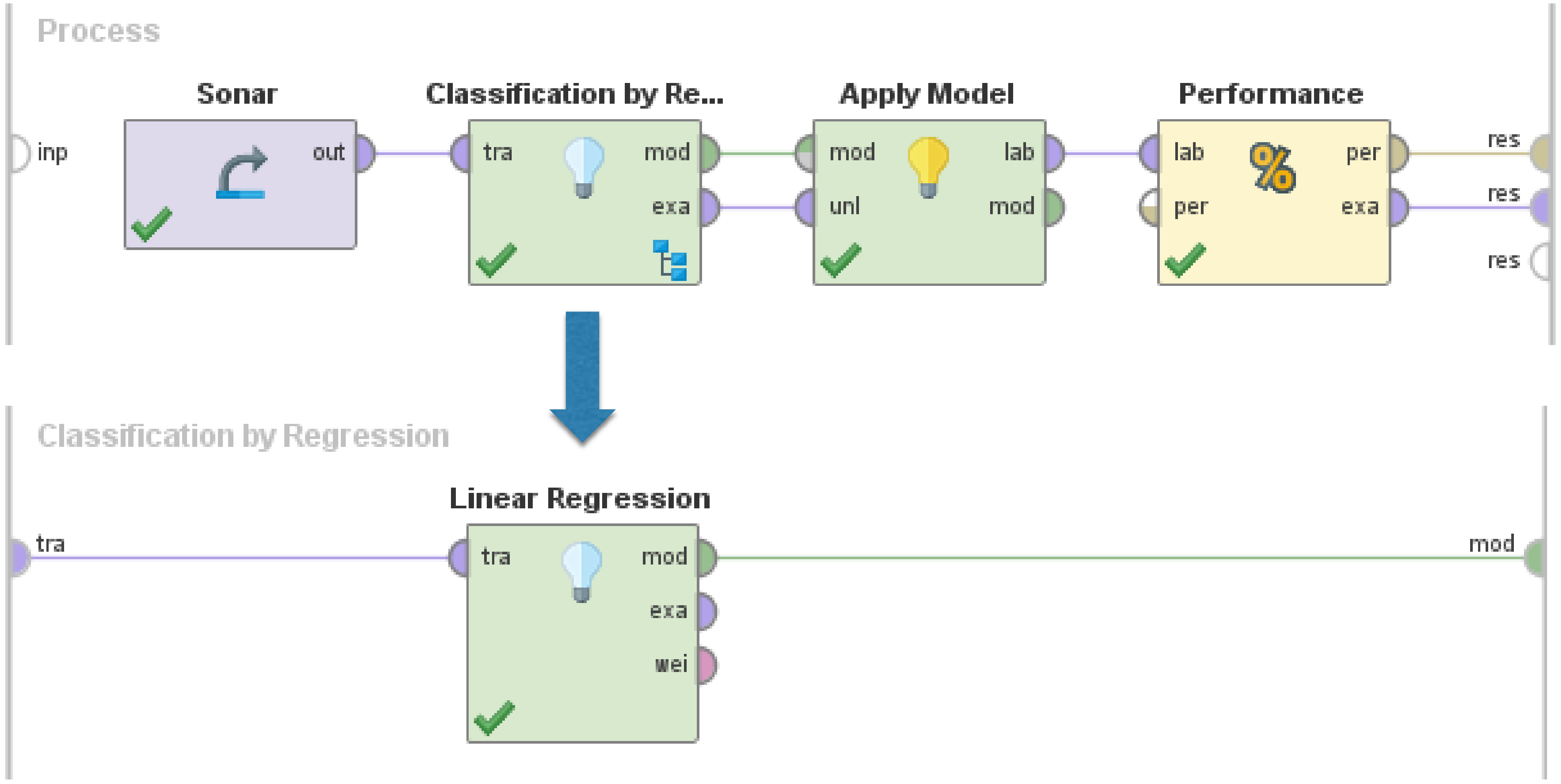
Normalization



- The accuracy is now 92,5% all the time.

Linear Separation

- Operator “Classification by Regression” -> Insert „Linear Regression“



Linear Separation

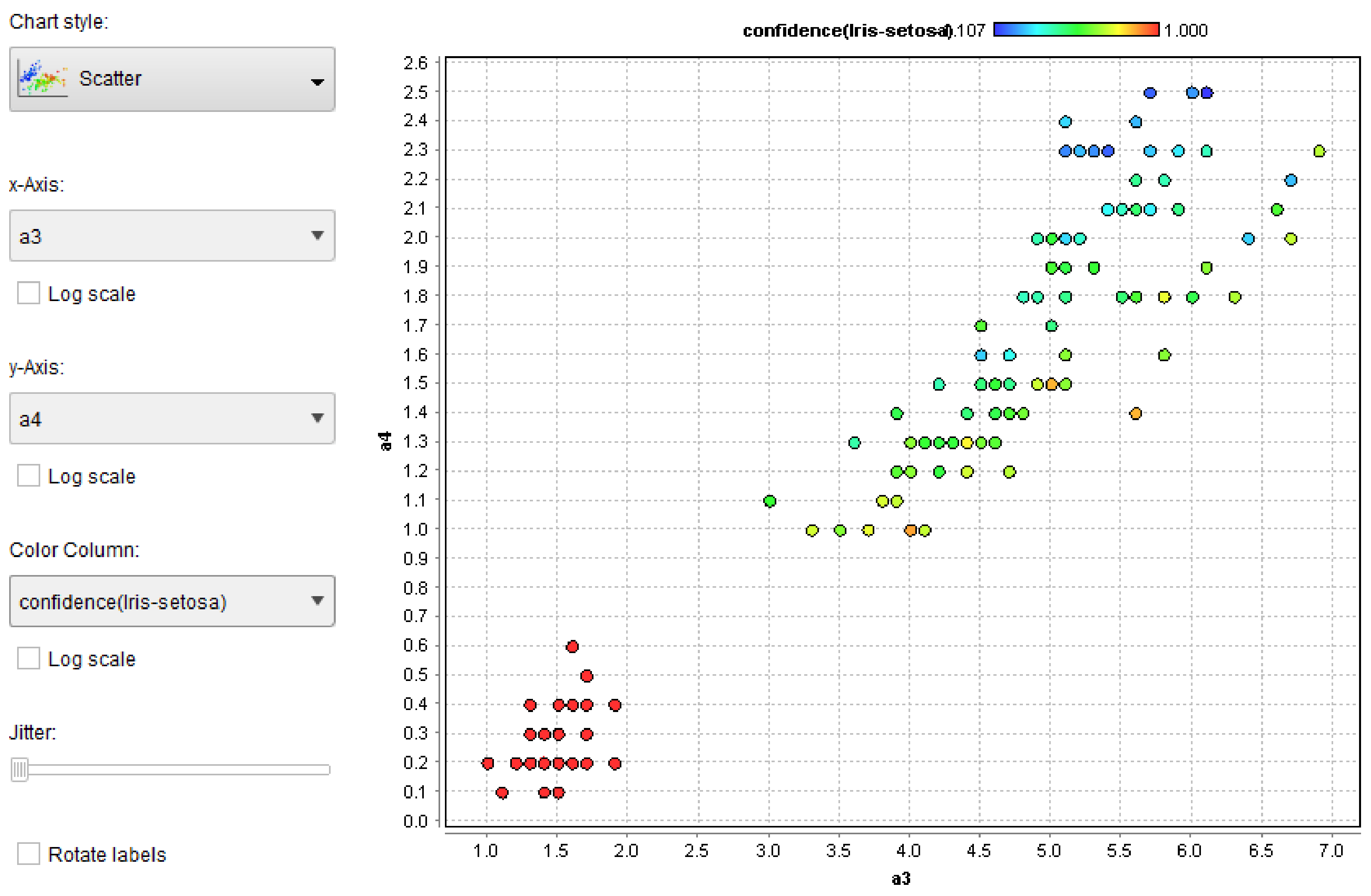


Chart style: Scatter

x-Axis: a3

Log scale

y-Axis: a4

Log scale

Color Column: confidence(Iris-setosa)

Log scale

Jitter:

Rotate labels