

Randomize

Shuffles classes, attributes and/or metas of an input dataset.

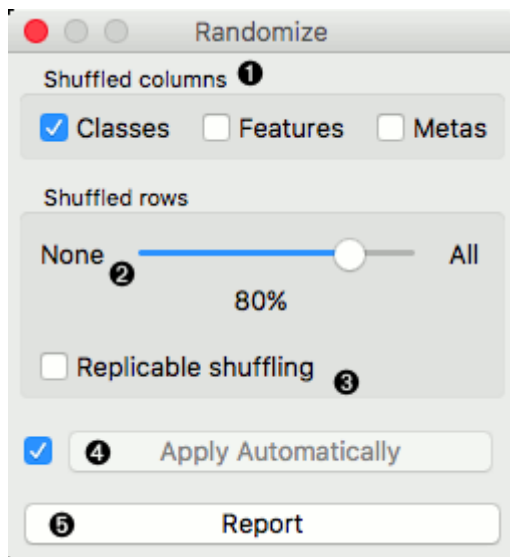
Inputs

- Data: input dataset

Outputs

- Data: randomized dataset

The **Randomize** widget receives a dataset in the input and outputs the same dataset in which the classes, attributes or/and metas are shuffled.



1. Select group of columns of the dataset you want to shuffle.
2. Select proportion of the dataset you want to shuffle.
3. Produce replicable output.
4. If *Apply automatically* is ticked, changes are committed automatically. Otherwise, you have to press *Apply* after each change.
5. Produce a report.

Example

The **Randomize** widget is usually placed right after (e.g. **File** widget). The basic usage is shown in the following workflow, where values of class variable of Iris dataset are randomly shuffled.

The screenshot displays the Orange Data Mining software interface. A workflow is visible in the main canvas, consisting of a 'File' widget connected to a 'Randomize' widget, which is then connected to a 'Data Table' widget. The 'Randomize' widget's settings are shown in a pop-up window, and the 'Data Table' widget's data is displayed in a table.

Randomize Widget Settings:

- Shuffled columns:** ☒ Classes, ☐ Features, ☐ Metas
- Shuffled rows:** None All (100%)
- ☒ Replicable shuffling
- ☒ Apply Automatically
- Report**

Data Table Widget Settings:

- Info:** 150 instances (no missing values), 4 features (no missing values), Discrete class with 3 values (no missing values), No meta attributes
- Variables:** ☒ Show variable labels (if present), ☐ Visualize continuous values, ☒ Color by instance classes
- Selection:** ☒ Select full rows
- Buttons:** Restore Original Order, Report, ☒ Send Automatically

Data Table Data:

	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.100	3.500	1.400	0.200
2	Iris-setosa	4.900	3.000	1.400	0.200
3	Iris-setosa	4.700	3.200	1.300	0.200
4	Iris-setosa	4.600	3.100	1.500	0.200
5	Iris-setosa	5.000	3.600	1.400	0.200
6	Iris-setosa	5.400	3.900	1.700	0.400
7	Iris-setosa	4.600	3.400	1.400	0.300
8	Iris-setosa	5.000	3.400	1.500	0.200
9	Iris-setosa	4.400	2.900	1.400	0.200
10	Iris-setosa	4.900	3.100	1.500	0.100
11	Iris-setosa	5.400	3.700	1.500	0.200
12	Iris-setosa	4.800	3.400	1.600	0.200
13	Iris-setosa	4.800	3.000	1.400	0.100
14	Iris-setosa	4.300	3.000	1.100	0.100
15	Iris-setosa	5.800	4.000	1.200	0.200
16	Iris-setosa	5.700	4.400	1.500	0.400
17	Iris-setosa	5.400	3.900	1.300	0.400
18	Iris-setosa	5.100	3.500	1.400	0.300
19	Iris-setosa	5.700	3.800	1.700	0.300

Data Table (1) Widget Settings:

- Info:** 150 instances (no missing values), 4 features (no missing values), Discrete class with 3 values (no missing values), No meta attributes
- Variables:** ☒ Show variable labels (if present), ☐ Visualize continuous values, ☒ Color by instance classes
- Selection:** ☒ Select full rows
- Buttons:** Restore Original Order, Report, ☒ Send Automatically

Data Table (1) Data:

	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.100	3.500	1.400	0.200
2	Iris-versicolor	4.900	3.000	1.400	0.200
3	Iris-versicolor	4.700	3.200	1.300	0.200
4	Iris-setosa	4.600	3.100	1.500	0.200
5	Iris-virginica	5.000	3.600	1.400	0.200
6	Iris-versicolor	5.400	3.900	1.700	0.400
7	Iris-virginica	4.600	3.400	1.400	0.300
8	Iris-setosa	5.000	3.400	1.500	0.200
9	Iris-setosa	4.400	2.900	1.400	0.200
10	Iris-virginica	4.900	3.100	1.500	0.100
11	Iris-versicolor	5.400	3.700	1.500	0.200
12	Iris-setosa	4.800	3.400	1.600	0.200
13	Iris-virginica	4.800	3.000	1.400	0.100
14	Iris-versicolor	4.300	3.000	1.100	0.100
15	Iris-versicolor	5.800	4.000	1.200	0.200
16	Iris-setosa	5.700	4.400	1.500	0.400
17	Iris-versicolor	5.400	3.900	1.300	0.400
18	Iris-versicolor	5.100	3.500	1.400	0.300
19	Iris-setosa	5.700	3.800	1.700	0.300

In the next example we show how shuffling class values influences model performance on the same dataset as above.

The diagram illustrates a workflow in Orange Data Mining. It starts with a 'File' widget connected to a 'Data' widget. The 'Data' widget is connected to a 'Randomize' widget. The 'Randomize' widget is connected to another 'Data' widget. This second 'Data' widget is connected to a 'Learner' widget (Logistic Regression) and a 'Test & Score' widget. The 'Learner' widget is also connected to the 'Test & Score' widget. The 'Test & Score' widget is connected to a 'Test & Score (Randomize)' widget. The 'Test & Score' widget displays evaluation results for Logistic Regression, and the 'Test & Score (Randomize)' widget displays evaluation results for Logistic Regression after shuffling.

Test & Score

Sampling

- ☒ Cross validation
- Number of folds: 10
- ☒ Stratified
- ☐ Random sampling
- Repeat train/test: 10
- Training set size: 66 %
- ☒ Stratified
- ☐ Leave one out
- ☐ Test on train data
- ☐ Test on test data

Target Class

(Average over classes)

Report

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
Logistic Regression	0.990	0.960	0.960	0.962	0.960

Randomize

Shuffled columns

- ☒ Classes
- ☐ Features
- ☐ Metas

Shuffled rows

None All

100%

☒ Replicable shuffling

☒ Apply Automatically

Report

Test & Score (Randomize)

Sampling

- ☒ Cross validation
- Number of folds: 10
- ☒ Stratified
- ☐ Random sampling
- Repeat train/test: 10
- Training set size: 66 %
- ☒ Stratified
- ☐ Leave one out
- ☐ Test on train data
- ☐ Test on test data

Target Class

(Average over classes)

Report

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
Logistic Regression	0.518	0.347	0.335	0.332	0.347

