

Naive Bayes

A fast and simple probabilistic classifier based on Bayes' theorem with the assumption of feature independence.

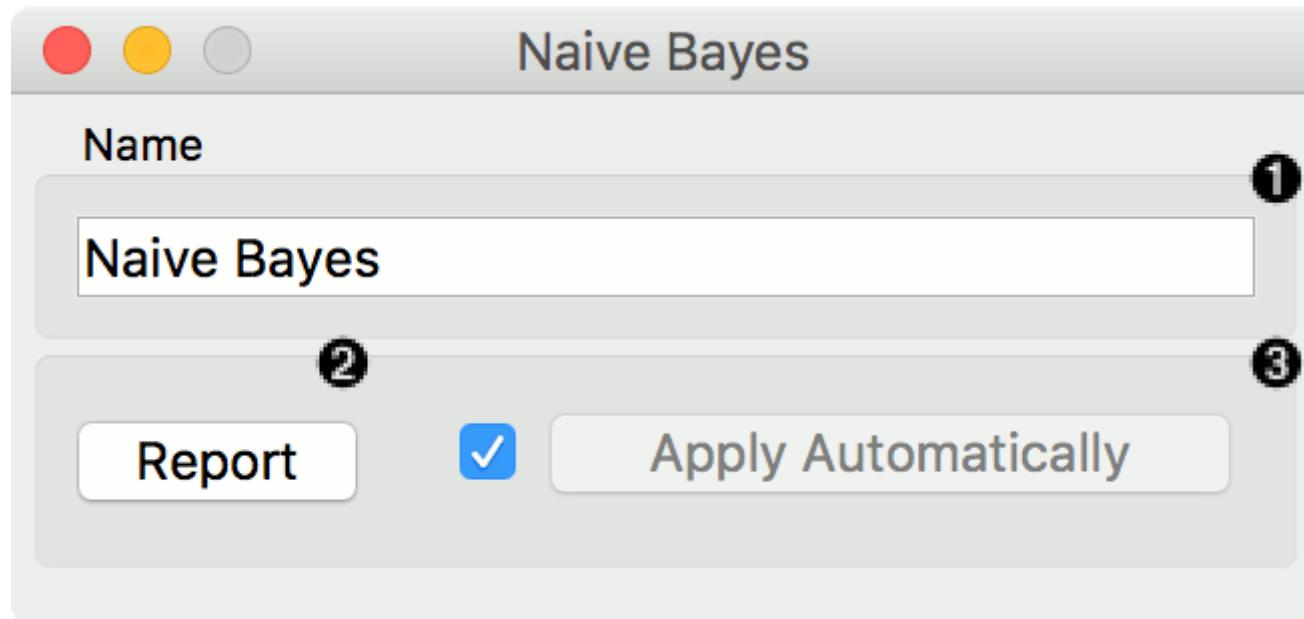
Inputs

- Data: input dataset
- Preprocessor: preprocessing method(s)

Outputs

- Learner: naive bayes learning algorithm
- Model: trained model

Naive Bayes learns a **Naive Bayesian** model from the data. It only works for classification tasks.



This widget has two options: the name under which it will appear in other widgets and producing a report. The default name is *Naive Bayes*. When you change it, you need to press *Apply*.

Examples

Here, we present two uses of this widget. First, we compare the results of the **Naive Bayes** with another model, the **Random Forest**. We connect *iris* data from **File** to **Test & Score**. We also connect **Naive Bayes** and **Random Forest** to **Test & Score** and observe their prediction scores.

The screenshot displays the Orange Data Mining software interface. On the left, a vertical toolbar contains various widget icons. The main workspace shows a workflow: a 'File' widget (orange circle with a document icon) is connected to a 'Test & Score' widget (light blue circle with a flask icon). Two model widgets, 'Naive Bayes' and 'Random Forest' (both pink circles with tree icons), are also connected to the 'Test & Score' widget. Two windows are open on the right. The 'Naive Bayes' window has a 'Name' field containing 'Naive Bayes', a 'Report' button, a checked checkbox, and an 'Apply Automatically' button. The 'Test & Score' window shows 'Sampling' options with 'Cross validation' selected (10 folds, stratified) and 'Evaluation Results' for Naive Bayes and Random Forest.

Naive Bayes

Name: Naive Bayes

Report ☒ Apply Automatically

Test & Score

Sampling

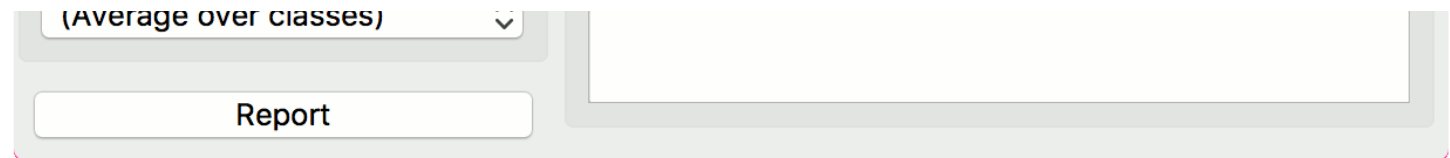
- ☒ Cross validation
 - Number of folds: 10
 - ☒ Stratified
- ☐ Random sampling
 - Repeat train/test: 10
 - Training set size: 66 %
 - ☒ Stratified
- ☐ Leave one out
- ☐ Test on train data
- ☐ Test on test data

Target Class

(Average over classes)

Evaluation Results

Method	AUC	CA	F1	Precision
Naive Bayes	0.983	0.900	0.900	0.900
Random Forest	0.997	0.960	0.960	0.960



The second schema shows the quality of predictions made with **Naive Bayes**. We feed the **Test & Score** widget a Naive Bayes learner and then send the data to the **Confusion Matrix**. We also connect **Scatter Plot** with **File**. Then we select the misclassified instances in the **Confusion Matrix** and show feed them to **Scatter Plot**. The bold dots in the scatterplot are the misclassified instances from **Naive Bayes**.

