Stacking

Stack multiple models.

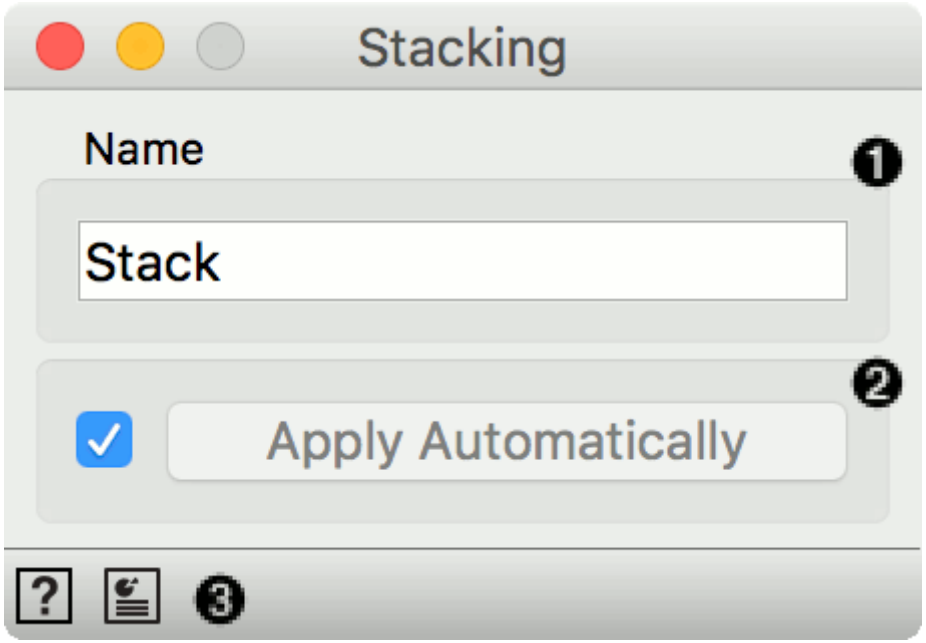
Inputs

- Data: input dataset
- Preprocessor: preprocessing method(s)
- Learners: learning algorithm
- Aggregate: model aggregation method

Outputs

- Learner: aggregated (stacked) learning algorithm
- Model: trained model

Stacking is an ensemble method that computes a meta model from several base models. The **Stacking** widget has the **Aggregate**input, which provides a method for aggregating the input models. If no aggregation input is given the default methods are used. Those are **Logistic Regression** for classification and **Ridge Regression** for regression problems.



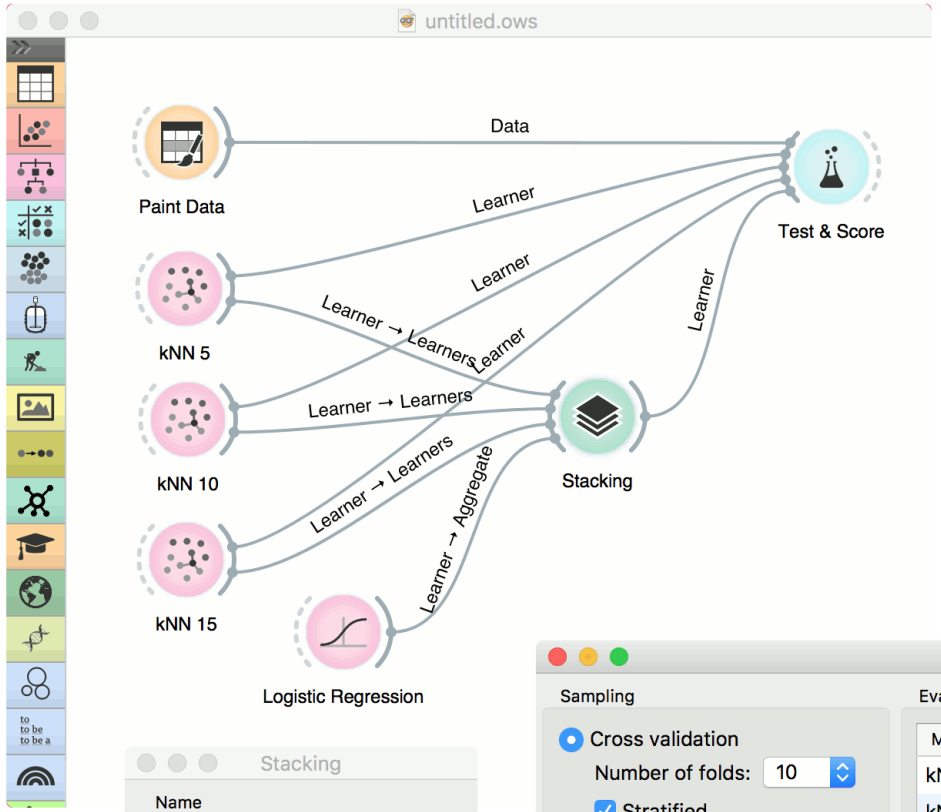
1. The meta learner can be given a name under which it will appear in other widgets. The default name is “Stack”.

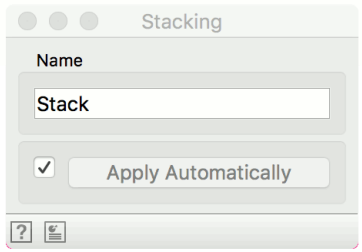
2. Click *Apply* to commit the aggregated model. That will put the new learner in the output and, if the training examples are given, construct a new model and output it as well. To communicate changes automatically tick *Apply Automatically*.
3. Access help and produce a report.

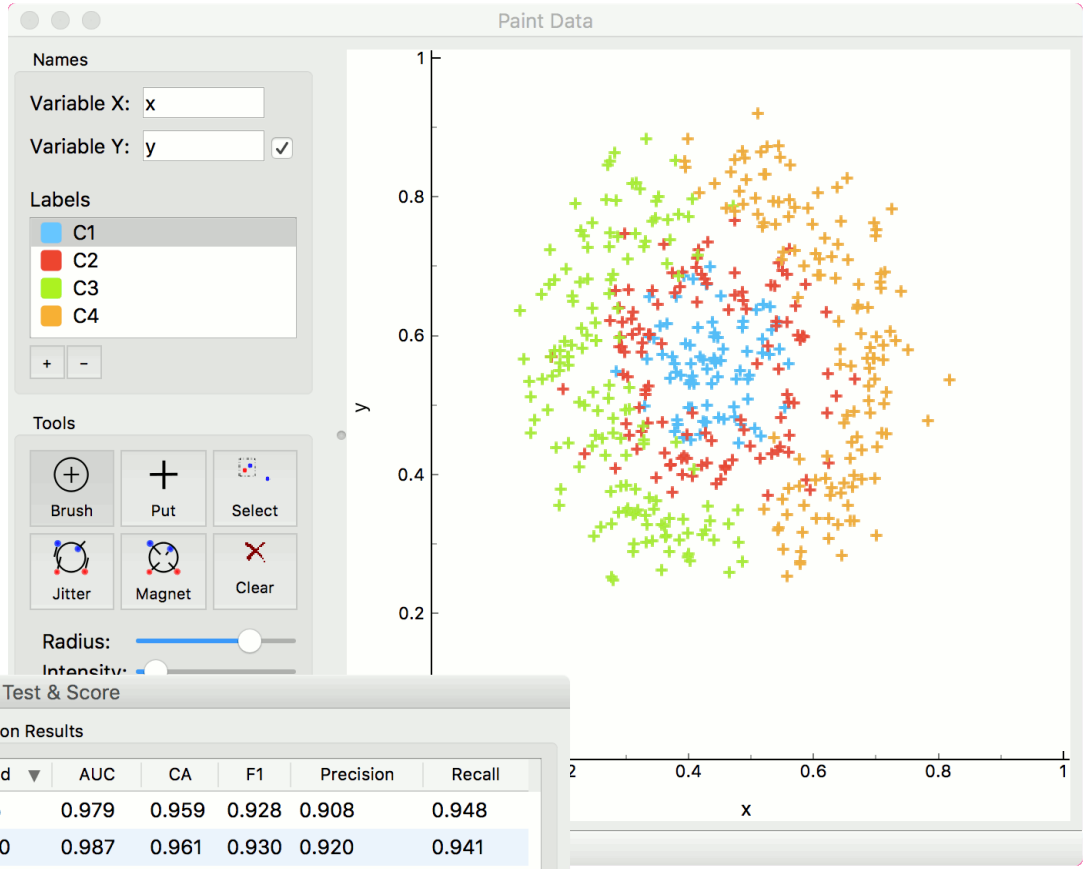
Example

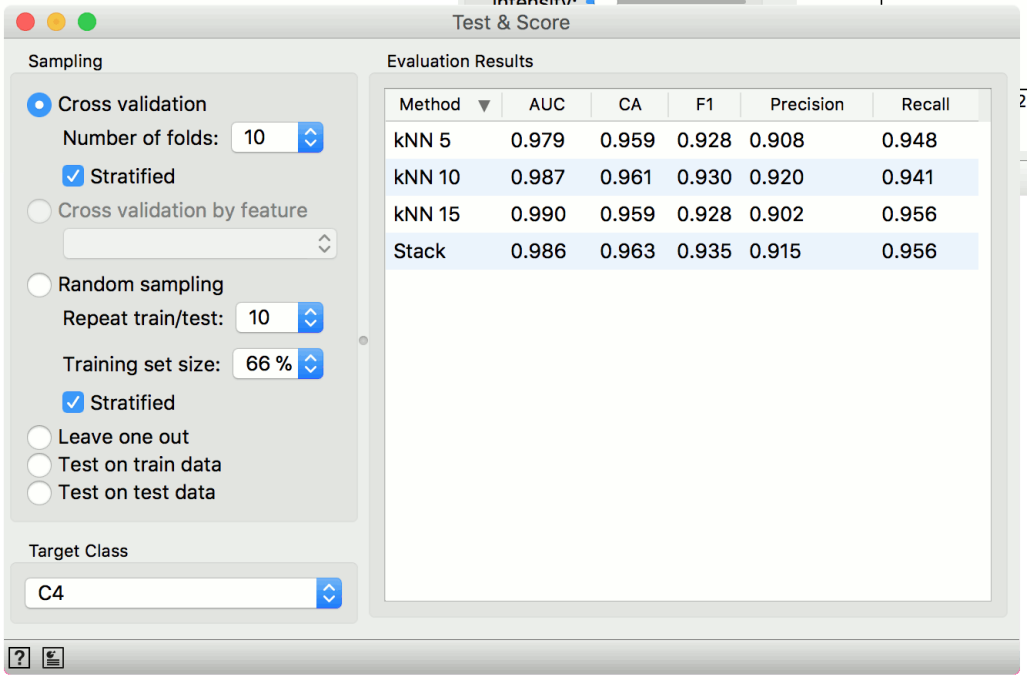
We will use Paint Data to demonstrate how the widget is used. We painted a complex dataset with 4 class labels and sent it to Test & Score. We also provided three kNN learners, each with a different parameters (number of neighbors is 5, 10 or 15). Evaluation results are good, but can we do better?

Let's use **Stacking**. **Stacking** requires several learners on the input and an aggregation method. In our case, this is Logistic Regression. A constructed meta learner is then sent to **Test & Score**. Results have improved, even if only marginally. **Stacking** normally works well on complex data sets.


untitled.ows
Paint Data
kNN 5
kNN 10
kNN 15
Logistic Regression
Stacking
Test & Score
Data
Learner
Learner → Learners
Learner → Aggregate


Stacking
Name
Stack
Apply Automatically


Paint Data
Names
Variable X: x
Variable Y: y
Labels
C1
C2
C3
C4
Tools
Brush
Put
Select
Jitter
Magnet
Clear
Radius:
1
0.8
0.6
0.4
0.2
y
0.4
0.6
0.8
1
x


Test & Score
Sampling
Cross validation
Number of folds: 10
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 66 %
Stratified
Leave one out
Test on train data
Test on test data
Target Class
C4
Evaluation Results
Method AUC CA F1 Precision Recall
kNN 5 0.979 0.959 0.928 0.908 0.948
kNN 10 0.987 0.961 0.930 0.920 0.941
kNN 15 0.990 0.959 0.928 0.902 0.956
Stack 0.986 0.963 0.935 0.915 0.956