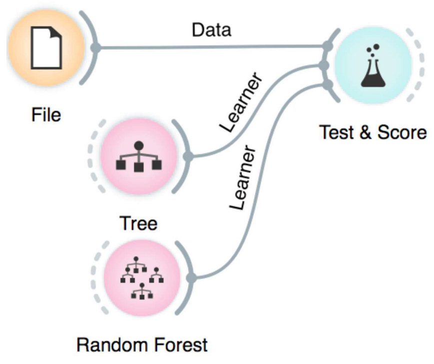
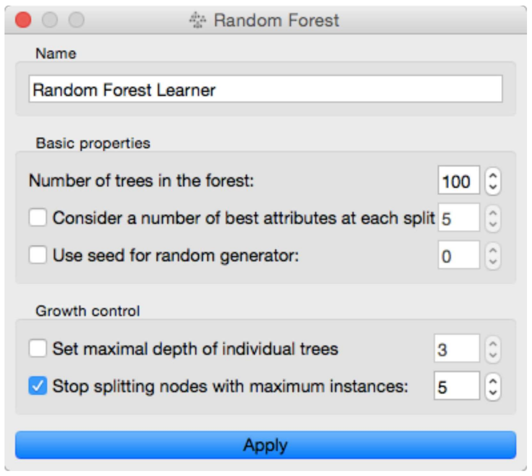
# Lesson 11: A Few More Classifiers

We have ended the previous lesson with cross-validation and classification trees. There are many other, much more accurate classifiers. A particularly interesting one is Random Forest, which averages across predictions of hundreds of classification trees. It uses two tricks to construct different classification trees. First, it infers each tree from a sample of the training data set (with replacement). Second, instead of choosing the most informative feature for each split, it randomly selects from a subset of most informative features. In this way, it randomizes the tree inference process. Think of each tree shedding light on the data from a different perspective. Just like in the wisdom of the crowd, an ensemble of trees (called a forest) usually performs better than a single tree.



Let us see if this is really so. We give two learners to the Test Learners widget and check if cross-validated classification accuracy is indeed higher for random forest. Choose different classification data sets for this comparison, starting with those we already know (hearth disease, iris, brown selected).



Name

Random Forest Learner

Basic properties

Number of trees in the forest: 100

☐ Consider a number of best attributes at each split 5

☐ Use seed for random generator: 0

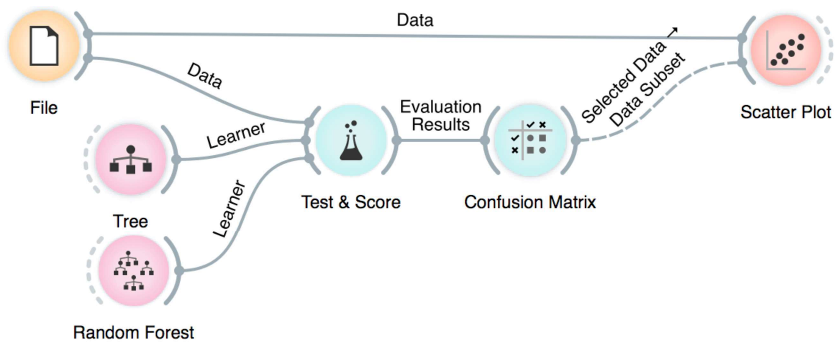
Growth control

☐ Set maximal depth of individual trees 3

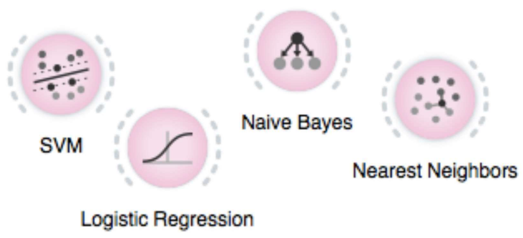
☑ Stop splitting nodes with maximum instances: 5

It may be interesting to compare where different classification methods make mistakes. We can use Confusion Matrix for this purpose, and then pass the signal from this widget to the Scatter Plot.

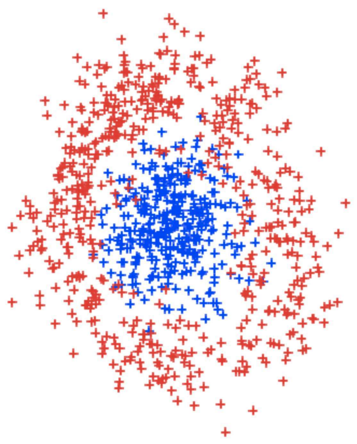
**What kind of object is sent from the Test & Score widget to the Confusion Matrix widget? So far, we have used widgets that send data, or even learners. But what could the Test & Score widget communicate to other widgets?**



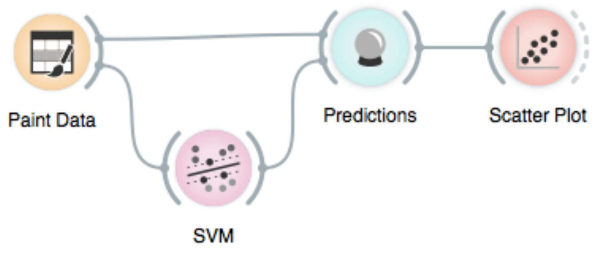
There are other classifiers we can try. We will briefly mention a few more, but instead of diving into what they do (we could spend a semester on this!), we'll pass on to other important topics in data mining. At this point, just add them to the workflow above and see how they perform.

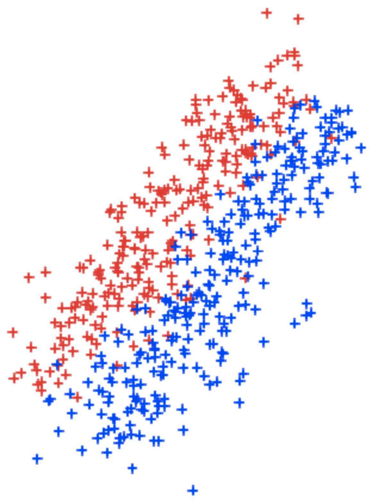


It would be nice if we could, at least on the intuitive level, understand the differences between all these methods and their variants (every method has some parameters). Remember, the classification tree finds hyperplanes orthogonal to the axis; those hyperplanes split the data space to regions with different class probabilities. The tree's decision boundaries are flat. Nearest neighbors classifies the data instance according to the few neighboring data instances in the training set. Decision boundaries with this approach could be very complex. Logistic regression infers just one hyperplane (decision boundary) in an arbitrary direction. This is similar to support vector machines with linear kernel, but then again, the kernels with SVM can be changed, resulting in more complex decision boundaries.

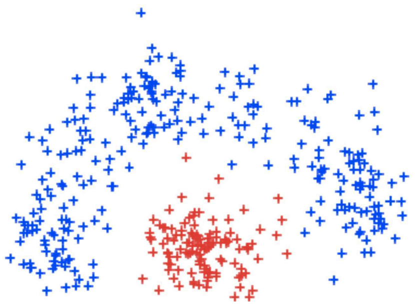


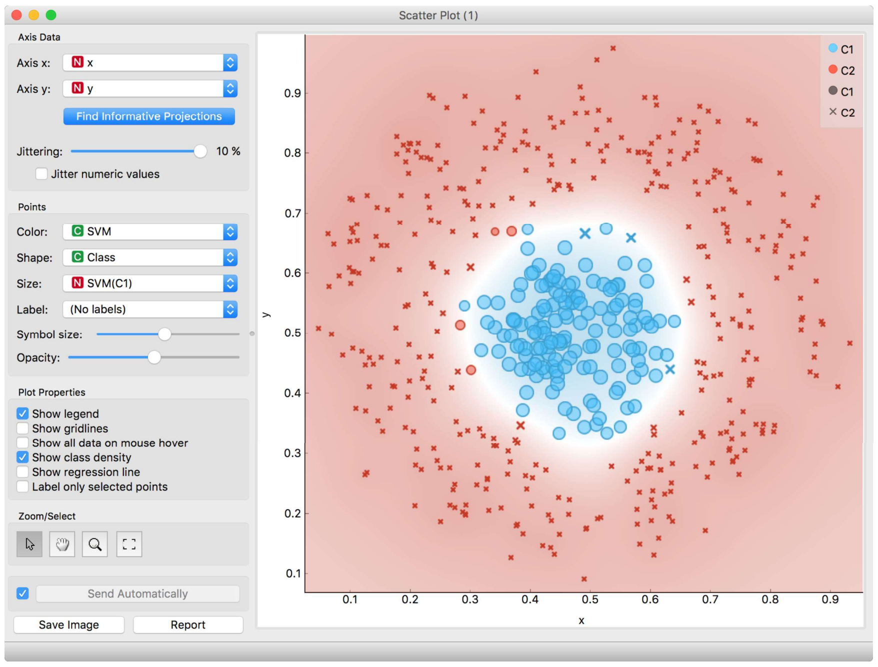
Ok, we have to admit: the above paragraph reads almost like gibberish. We would need a workflow where we could actually see the decision boundaries. And perhaps invent the data sets to test the classifiers. Best in 2D. Maybe, for a start, we could just paint the data. Time to stop writing this long passage of text, end the suspense, and construct a workflow that does this all.





Be creative when painting the data! Also, instead of SVM, use different classifiers. Also, try changing the parameters of the classifiers. Like, limit the depth of the decision tree to 2, or 3, 4. Or switch from SVM with linear kernel to the radial basis function. Appropriately set up the scatter plot to observe the changes.





Scatter Plot (1)

Axis Data

Axis x: x

Axis y: y

Find Informative Projections

Jittering: 10 %

☐ Jitter numeric values

Points

Color: SVM

Shape: Class

Size: SVM(C1)

Label: (No labels)

Symbol size:

Opacity:

Plot Properties

☑ Show legend

☐ Show gridlines

☐ Show all data on mouse hover

☑ Show class density

☐ Show regression line

☐ Label only selected points

Zoom/Select

☑ Send Automatically

Save Image | Report