B.Sc. in Computing
Data Mining

Lab sheet #3 = classification with Decision Trees

rapid-i.com

G. Gray

Overview

Objective:

- Using a Decision tree to classify a dataset

- Agenda:
  - Creating test and training datasets
  - Creating a decision tree model
  - Changing parameters

Exercises

There are a number of exercises through out the slides, and repeated again on the last slide.

At the end of the lab, you must complete an MCQ test based on your answers to these exercises.

Modeling the Iris dataset

1. Start a new process calling it lab3-Iris-Holdout
2. Go to rapidminers repository, called samples. Navigate to Data and then Drag the Iris dataset onto the process window.
3. Connect out to res, and run the process to view the dataset.
4. The next step is to split the dataset into a training dataset and a test dataset. Rapidminer has a number of ways to divide the dataset into a training dataset and a test dataset located under: Evaluation / Validation:
   - **Spit validation**: is the holdout method
   - **X-Validation**: is cross validation
   - **Bootstrap validation** bootstraps the dataset first, and then does cross-validation

Generating test and training data

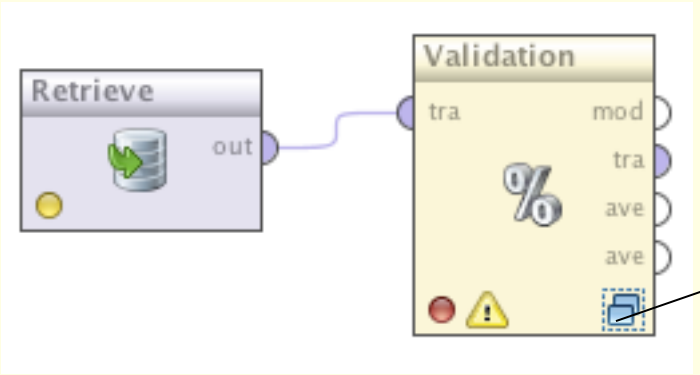
Just generating a decision tree on its own does not give information on how accurately the tree is likely to preform against a test dataset or live data.

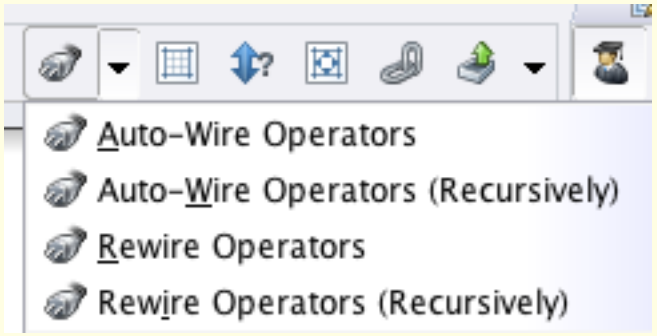
To do this, we will need to divide the dataset into two parts, a test dataset and a training dataset.

Rapidminer has a number of ways to divide the dataset into a training dataset and a test dataset located under: Validation / Evaluation. For now, we will use Split Validation.

Generating test and training data using the holdout method

1. Add a split validation process to the window.
   - Note: a quick way to find this is to start typing split validation into the filter text box on the left hand side operator window.
   - Also, if autowire is enabled, Rapidminer will connect operators automatically.
2. You should now have the following:

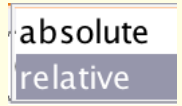




Indicates this operator can have operators nested within it. Double click to drill down

Parameters for split validation

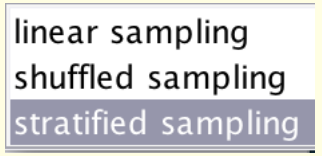
Click on the Split Validation operators to view its three parameters. Hover the mouse over the name of the parameter for an explanation:



Split: If set to absolute, the second parameter will be the number of rows to include in the training dataset; if set to relative, , the second parameter will be the percentage of rows to include in the training dataset. Leave it as relative.

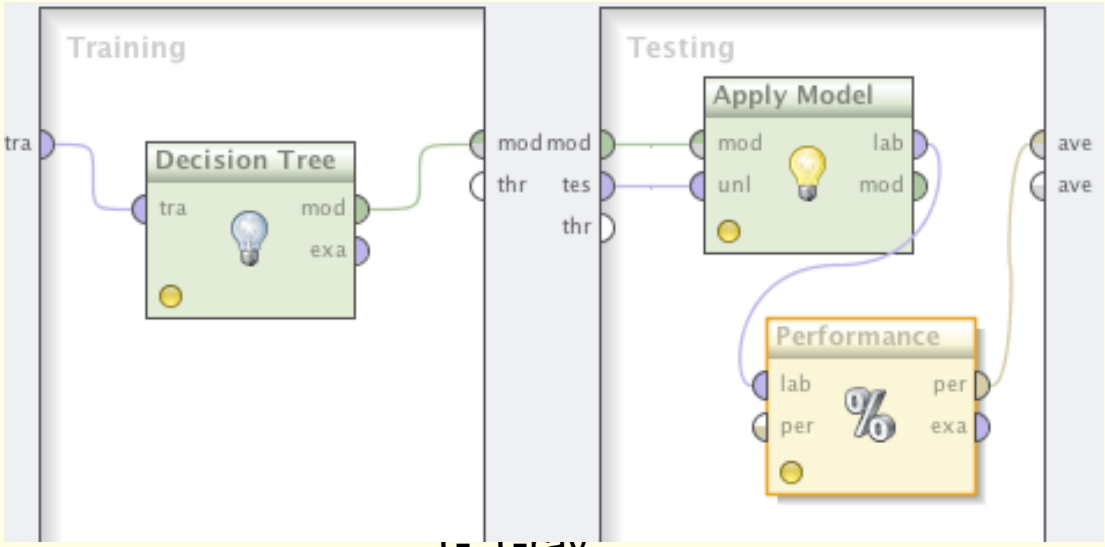
Split ratio: What percentage of rows to include in the training dataset. The default, 0.7, means 70%.

Sampling type: **linear** sample will pick the first 70% of the rows as the training dataset; **shuffled** will randomly select 70% of rows for the training dataset; **stratified** will select 70% of rows from each class, i.e. 70% of the *iris setosa* rows, 70% of the *iris virginica* row and 70% of the *iris veriscolor* rows. **Stratified** is the best option.



Generating test and training data

5. Double click on Validation to drill down to operators nested within it.
   - This presents two sub-process windows. One is for operators running on the TRAINING dataset (training the model). The other is for operators running on the TEST dataset (apply the model to a test dataset and evaluate the model).
6. Add a decision tree to the Training side: in the search box under '**Operators**', start typing Decision Tree until you see the operator you want.
7. Add an Apply Model operator, and then a Performance (classification) operator to the Test side. This should give you the following:



Generating test and training data

Click on the blue 'up arrow' to return to the parent process: 

We want to output three things from this process:

- The dataset itself (tra)
- The decision tree (mod)
- The accuracy of the decision tree when tested on the test dataset (ave)

Each are outputs from Validation. Connect each to a result port on the process window and run the process.

Exercise

Exercise 1:

How accurate is the decision tree classifier on the iris dataset when using the holdout method?

What is the recall on Iris Virginica?

What is the precision on Iris Veriscolor?

Which class is the classifier best at predicting?

Which class is the classifier worst at predicting?

Exercise

Experiment with other split ratios to assess their impact on overall accuracy:

Exercise 2:

If the split ratio is set to 50% does overall accuracy go up or down?
Why do you think this is the case?

If the split ratio is set to 90% does overall accuracy go up or down?
Why do you think this is the case?

Building blocks

In Rapidminer, groups of operators that are frequently used together can be grouped together into a bulding block, and saved for use again in another process.

Rapidminer comes with a few building blocks already defined.

Start a new process called lab3-Iris-XValidation

Retrieve the Iris dataset

Select edit / new building block

Select the 2nd building block – nominal X-Validation, and click OK to return to your process.

You should now have an X-Validation operator (cross validation), which has a decision tree as an inner operator on the training side, and an apply model and performance operator on the test side.

Building blocks

Move the validation operator to the right of the Retrieve dataset and connect them. Connect up the three outputs.

X-validation has a number of parameters:

Average Performances only: just calculate an average for overall accuracy.) It can calculate other metrics as well). Leave at default value.

Leave one out: The number of partitions equals the number of rows in the dataset, and each time the test dataset is just one row.

Number of validations: how many partitions to create. Leave this at the default of 10.

Sampling type: Explained under Holdout method (slide 7)

Exercise

Exercise 3:

What is the average accuracy of a decision tree classifier on the iris dataset when using cross validation?

What is the standard deviation?

What is the recall on Iris Virginica?

What is the precision on Iris Veriscolor?

The titanic dataset

Download the Titantic dataset from Moodle and import it into your Rapidminer repository.
The class label is survived.
Start a new process, call it lab3-Titanic
Retrieve the titanic dataset, and connect it to the output port.

Run the projcess and do and Exploratory Data Analysis on the dataset:

Exercise 4:
Are the missing values?
Are there outliers?
Looking at a scatter matrix, do you think a model of this dataset will be accurate?

The titanic dataset

Add a X-Validation block to the process, connect the ports, and run the process again again.

Exercise 5:
How accurate is the decision tree model?

Which class is predicted more accurately?

Recall on the class 'Yes' is 53.51%. What does that mean?

Take a look at the decision tree. How many attributes is it using?