

Lab 2: Creating models with Orange

In this second lab, we will dive into creating and evaluating machine learning models.

**Exercise I: Building models and making predictions**

**See the complete exercise at** [**https://youtu.be/D6zd7m2aYqU**](https://youtu.be/D6zd7m2aYqU)

This exercise will show you how to make a model and then use it to make predictions.

Open a new document, then create a new file widget and paste the following link in the URL inside your widget:

A screenshot of a computer

Description automatically generated

You should be able to see the table loaded in the file widget as follows:

A screenshot of a computer

Description automatically generated

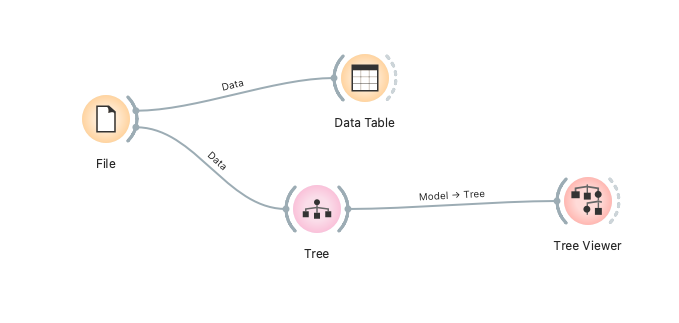
You can add a “Data Table” widget to explore the different values of the dataset. You can see that a class called “Classification” represents the model's outputs. We are going to create our first model and then visualize it.

In the data table widget, you will see blue and red lines. Why do you think these lines are in different colours?

Once you load the model using the file, you will use a “Tree” model, a.k.a Decision tree to generate an ML model. If you want to get more information on what a tree is and how it is created you could explore the following widget and also do your research:

<https://orangedatamining.com/widget-catalog/model/tree/>

Now, create the following structure. Once you do that, click on “Tree Viewer,” and you will see how the ML tree model has created a tree, which will help you understand the dataset in better detail.



Now click on the “Tree Viewer” widget. This will display the Decision Tree generated by the model, which will look as follows:

A diagram of a diagram

Description automatically generated

There are a few questions that are relevant to this decision tree:

* What do the colours “Blue” and “Red” represent?
* What does the intensity of the colour represent?
* Where are the features in this tree?
* What do the end nodes represent?
* What do the fractions represent? Is this fraction related to the small circle?
* What made the model select the calories as the main feature?
* What are the two most important things when selecting a feature when building a tree?

**Exercise II: Evaluating models**

**See the complete exercise at** [**https://youtu.be/pYXOF0jziGM**](https://youtu.be/pYXOF0jziGM)

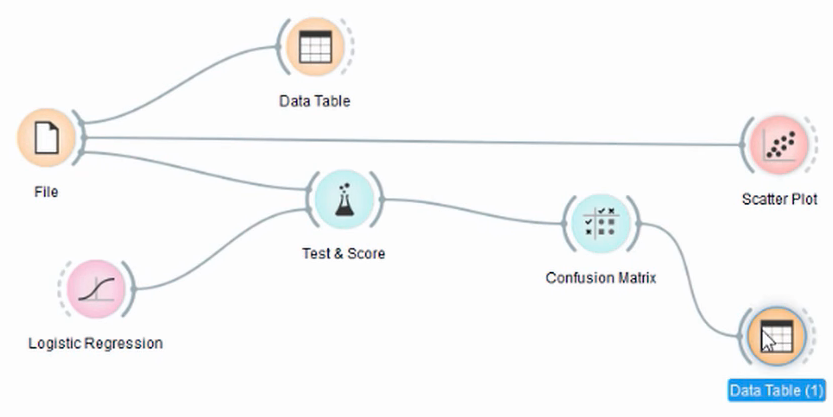
In this part, we will introduce some of the essential and day-to-day concepts in ML. We are going to see the following concepts:

* **Confusion Matrix:** It is used to see the correct and wrong predictions for the model across multiple classes. False positives and false Negatives are essential concepts related to the confusion matrix.

<https://www.digitalocean.com/community/tutorials/confusion-matrix-in-r>

* **Cross-Validation:** It is a widespread technique for calculating the performance of the models in a more rigorous statistical way. We get a dataset and divide it into ten equal parts. Then, we use nine parts for training and one for testing. We do this ten times and calculate the final accuracy as the average.

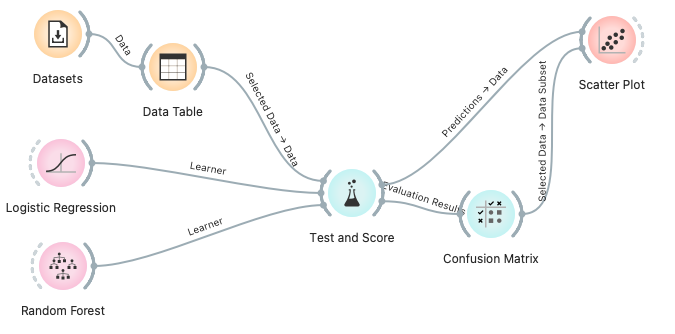
Follow the video's instructions to calculate the performance of the famous dataset Iris.

****

What should make us more concerned about this exercise is why the model misclassified some instances. Is this a problem with the mode? Is this because the dataset has some wrong information? What explanation can we give for these misclassifications?

To learn more, we will create a “Data Table” widget and connect it to the confusion matrix (CM). We can select the misclassified instances in the CM by clicking “Select Missclassified.” Then, connect the “Confusion Matrix” widget to the scatter plot.

In the “Scatter plot” widget, select the following attributes. Can you explain why the model failed to predict those values from the picture you see? Can we compare a simple “Logistic Regression” model with a better “Random Forest” model for a more complex dataset? Let's try a more challenging dataset.



Now, we will install the widget called “Datasets.” You can download multiple datasets with more instances and classes in that widget. Let’s try, for example, the dataset called “Car evaluation,” which has 1,728 cases and six variables. Can you see any difference in the performance of the models?

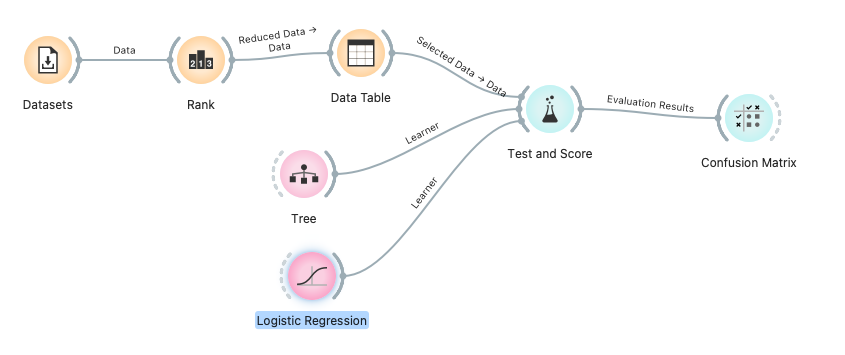
To see a real difference, let’s use a dataset with many instances, such as the “Adult” dataset, which predicts a person's annual income based on different attributes. If you apply CV, can you see the difference in performance between Logistic regression and Random Forest? RF is almost 89%, and LG is around 58%. This is a huge difference.

**Exercise III: Feature selection**

Feature selection is essential for steps along the pipeline of creating models. Feature selection allows one to select the most essential features of a dataset and remove the features that add little or no information. This could be because features are correlated to other features or because they are not related to the input.

To do this exercise, select the widget Datasets, then select the dataset called “Wines”, and use the Rank widget to detect the best features automatically. The Rank widget implements the best feature selection filter methods, such as “Information Gain” or “Gini Decrease”, to rank the best variables.

Try comparing the performance of the models using the best features, the best two features, and the best three features. See how the number of features can be drastically diminished to create simpler models without undermining performance. Here, you have a trade-off between the number of features and the model's performance. What do you think? Is it worthwhile to use all the variables to increase the model's performance? Can some models be improved by reducing the number of variables?



On top of ranking the features, you can make combinations of them using Principal Component Analysis (PCA). PCA is a prevalent method for reducing the number of variables. If you, for example, have 100 variables, PCA will combine them to generate a new set of more useful features for the model. Each new feature is called an eigenvalue.

PCA can be used to explore and create new models. Try it on various datasets to see how it can reduce the number of features.