Machine Learning

Practice 4: Decision trees

Academic year 2025-2026

Nantes University

Loading data and preliminary statistical analysis

Movies

This time, we are dealing with a dataset related to movies. You probably already downloaded the data `movies.csv` from Madoc, and loaded it. Before diving into learning models, be sure to proceed to a first statistical (and understanding) analysis. First, we will not take into the analysis the title of the movie. This is NLP and out of the scope for the moment. The other attributes are self explanatory, except wr, which stands for Weighted Rating. This value has been proposed by IMDB and is defined by

$$wr = \left(\frac{v}{v+m}R\right) + \left(\frac{m}{v+m}C\right)$$

where

- v is the number of votes for the movie
- m is the minimum votes required to be listed in the chart
- R is the average rating of the movie
- C is the mean vote across the whole report

Data handling

- Get a first glimpse at the data by using **`describe, info, head`**
- Get a second one by using visualization tools **`pairplot, catplot`** on numerical and categorical variables, respectively.
- Finally, compute, observe and conclude about the dependency of the variables using correlation values.

Learning model

Now it's time to learn a model. We want to predict the popularity of movie. Attribute 'Popularity' is numeric.

- We could still make this into a classification task, how ?
- We decide to go for regression instead. Does this make sense ?
- In the lectures we were told about linear regression. Would this work here? Why not?

Once we are convinced with this we decide to go for regression trees.

Learning model - 2

As usual, first split your data into two (X, Y) sets for proper evaluation of the model.

- Learn the model using default parameters and a depth of 3. What are the important features ? Is it related to the initial study of relation between features ?
- Change the depth of the tree, and check the importances.
- What is the metric used to obtain the best split ? Try others and comment.
- Using `plot_tree`, visually observe the learned tree.

(*Training and ... testing*)^K

A drawback of training/testing split is that we do not use all the data at hand to learn the model. A more interesting, rather costly, technique is to perform several splits of the data, and to learn a model for each split. This is called cross-validation (CV). Note that simple train/test splitting we already know is a specific case of CV called holdout.

Perform this cross-validation on your model, using scikit library. Change the value of K. Draw some conclusions regarding: numbers of learned models, computational complexity, robustness of evaluation.

Going further

That was only the beginning ! One decision tree is surely interesting, but may lack of generalization capabilities. In such cases, people often learn not only one tree but several, thus transforming the model into a ... forest. These kind of models are called Random Forests[1], and give very good performances, even compared to deep models.

- Explore `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html` and try to improve the results obtained with one single tree.
- How can we interpret the model now ?

[1] L. Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.