

Artificial Intelligence 2 – SS 2020

Assignment 11: Decision Trees in Python

– Given July 26., Due July 12. –

Hint: Exercises need to be handed in via StudOn at 23:59 on the day they are due or earlier. Please use only the exercise group of your tutor to hand in your work.

If any concepts here seem unfamiliar to you or you have no idea how to proceed, consult the lecture materials, ask a fellow student, your tutor, or on the Forum.

If a problem asks for code, comment it or make it otherwise self-explanatory. You do not need to write a lot, but it should be enough to convince your tutor that you understand what the code does. We may deduct up to 30% for uncommented and unclear code, but would prefer not to.

Problems with no points (0pt) will not be graded, but might appear on the exam in a similar form. For these, we will provide a reference solution after the submission deadline. If you find the reference solution unclear, ask about it on the forum or in a tutorial.

Hint: Note that the submission date is in two weeks, on July 12th, and that we will post another (shorter) homework in a week (July 3rd), that will also be due on July 12th.

Problem 11.1 (Decision Tree Learning in Python)

Implement the *Decision Tree Learning* algorithm (DTL) in Python. Use information gain 100pt to choose the attribute for next node. You can find the necessary files at <https://kwarc.info/teaching/AI/resources/dtl.zip>.

Your task is to implement the `dtl` function in `dtl.py`. Use `test.py` for testing your implementation.

Important: We will test your code automatically. So please make sure that:

- The tests in `test.py` work on your code (without any modifications to `test.py`)
- You use a recent Python version (≥ 3.5)
- You don't use any libraries
- You only upload a single file `dtl.py` with your implementation of `dtl`

Otherwise you risk getting no points.

`dtl.py` contains an example query and expected answer based on homework problem 10.2. The same data set is used by `test.py`. For more interesting tests (with non-binary attributes), the restaurant-example from the slides is used. Since the restaurant example does not result in a unique decision tree, only the correctness and size of the resulting decision tree are displayed.

You can get partial points if your implementation returns a valid decision tree, even if you didn't use information gain.