Assignment 6

TDT4171 — Artificial Intelligence Methods

March 2024

Information

- **Delivery deadline: March 3, 2024 by 23:59**. No late delivery will be graded! Deadline extensions will only be considered for extraordinary situations such as family or health-related circumstances. These circumstances must be documented, e.g., with a doctor's note ("legeerklæring"). Having a lot of work in other classes is not a legitimate excuse for late delivery.
- Cribbing("koking") from other students is not accepted, and if detected, will lead to immediate failure of the course. The consequence will apply to both the source and the one cribbing.
- Students can **not** work in groups. Each student can only submit a solution individually.
- Required reading for this assignment: Chapter 19. Learning from Examples (the parts in the curriculum found on Blackboard "Sources and syllabus" → "Preliminary syllabus") of Artificial Intelligence: A Modern Approach, Global Edition, 4th edition, Russell & Norvig.
- For help and questions related to the assignment, **ask the student assistants during the guidance hours**. The timetable for guidance hours can be found under "Course work" on Blackboard. For other inquires, an email can be sent to tdt4171@idi.ntnu.no.
- Deliver your solution on Blackboard. Please upload your assignment as one PDF report and one source file containing the code (i.e., one .py file) as shown in Figure 1.

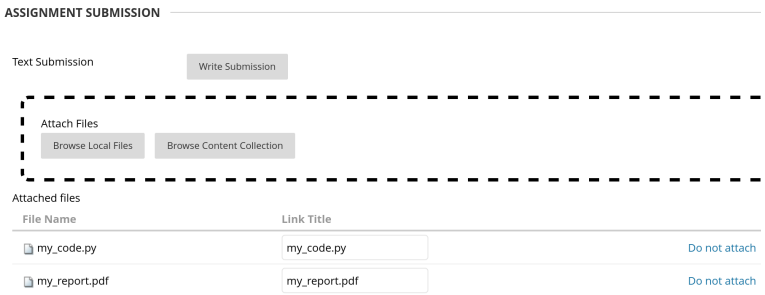

Figure 1: Delivery Example

Assignment Information

The main goal of this assignment is to implement the decision tree learning algorithm, pseudocode shown in Figure 19.5 on Page 678, and the IMPORTANCE function used by this algorithm.

We recommend using Python as the programming language for this assignment. A Python file (assignment_6.py) containing helper functions is available at Blackboard. You may use this file as a starting point for your implementation. It is specifically the *importance* and *learn_decision_tree* functions as detailed in this file that should be implemented for the assignment.

You may also choose to implement the algorithm with necessary functions from scratch in Python or in another programming language. If you choose to use another programming language, we cannot guarantee that we can help you with implementation if needed.

For the programming part of the assignment, the code must be runnable without any modifications after delivery. In addition, the code must be human-readable and contain explaining comments where appropriate. Commenting is especially important if the code does not work.

It is not allowed to use packages such as Scikit-learn to implement the decision tree learning algorithm. Furthermore, copying an implementation of the decision tree learning algorithm from the internet is not allowed.

Experiment data

Two data files (train.csv and test.csv) can be found on Blackboard along with this assignment text. The two data files have the same format: Each line describes an example, the first seven numbers are the attributes, the last number is the class of that example. All attributes, as well as the class, take values 1 or 2, and you can take advantage of this to simplify your code if you want.

Example: The first line in the training data is: 1,1,2,2,1,1,1,1. This means that for this example, we have

- Attribute 0 = 1,
- Attribute 1 = 1,
- Attribute 2 = 2,
- Attribute 3 = 2,
- Attribute 4 = 1,
- Attribute 5 = 1,
- Attribute 6 = 1,
- ... and that the example comes from class 1.

The test data has the same format. However, the class label for the test data is not to be used by the decision tree during learning, only to quantify the learning algorithm's accuracy afterward.

Exercise 1

Two different versions of IMPORTANCE function used by the pseudocode in Figure 19.5 on Page 678 should be implemented.

1. Allocate a random number as importance to each attribute.
2. Allocate a number using the expected information gain as importance to each attribute. For more information about this, see Section 19.3.3 Choosing attribute tests on Page 679.

To compare the two versions of IMPORTANCE, you should examine each of them by doing the following steps:

1. Implement the decision tree algorithm and learn a decision tree from the data in train.csv, using both versions of IMPORTANCE.
2. Classify all examples in the test set given in the data file test.csv. Calculate the accuracy of the learner by comparing them to the correct classification of the examples in the test set.

The PDF report should include:

- Using the accuracies of the two models, write a short discussion comparing the performace of the two models. Make sure to run both versions several times to observe overall performance. Which IMPORTANCE function is better, and why?

If you do not succeed in implementing a working decision tree algorithm, write a few sentences in the PDF report about what you would expect of the performace of the random version compared to the information-gain version, and make sure to deliver your code even if it does not work.