Decision Trees lab report

Moisés Montaño Copca

A01271656

- **Explain the advantages and disadvantages of writing a program on your own vs using a pre-created suite such as WEKA.**

When writing a program the most notorious advantage is that you get a deeper knowledge of the algorithm, so whenever you need to make adjustments, based on the problem or data you have, it is a lot easier than using a tool with the algorithms already defined. However, when using a pre-created suite, it's possible that the already implemented algorithm is much more 'complete' in the sense that it may have been tested against very special test cases, at least to ensure it won't crash and it may also be more efficient than a program of my own, considering it probably was implemented by experts in both programming and the algorithms.

- **Explain what criteria you followed to choose the datasets for your tree and the WEKA tests.**

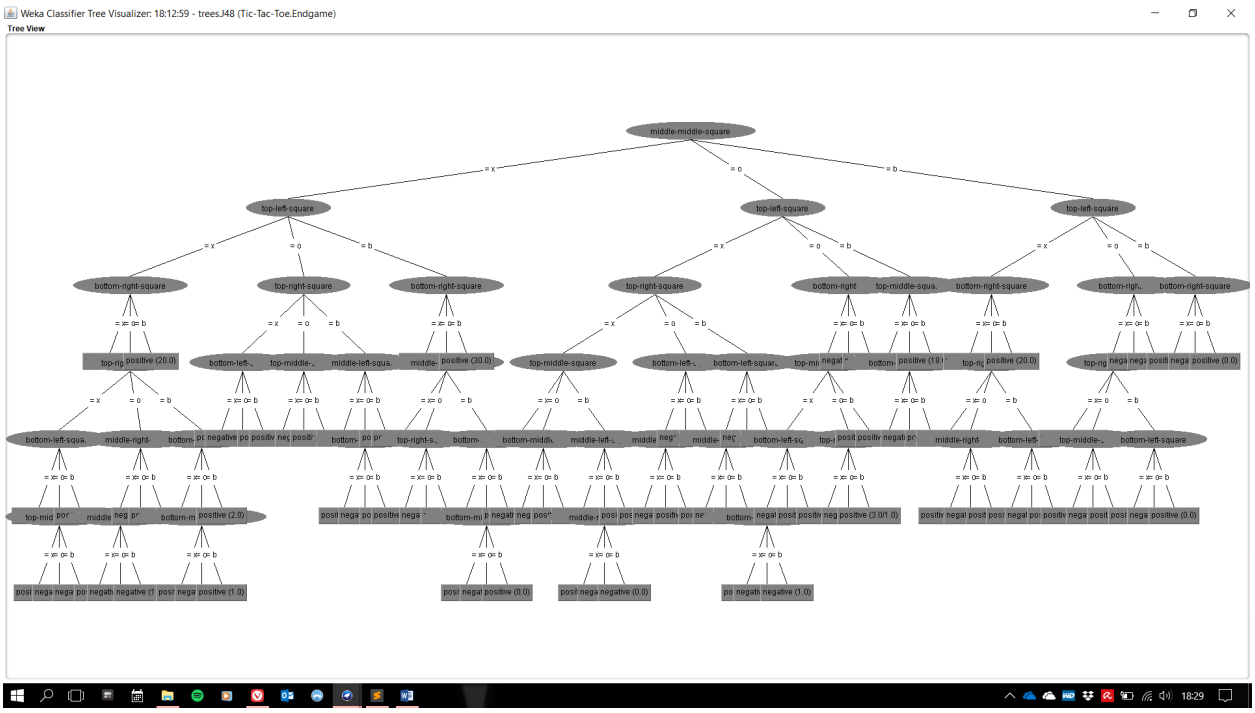
First I looked for information that had a similar format to the one I implemented in the lab (arff) to ensure that my code would work correctly with it.

Then I looked for a lot of data, I didn't focus on the attributes having many different values (actually all of them share the same possible values, except the answer) because the testcases in alphagrader already involve this.

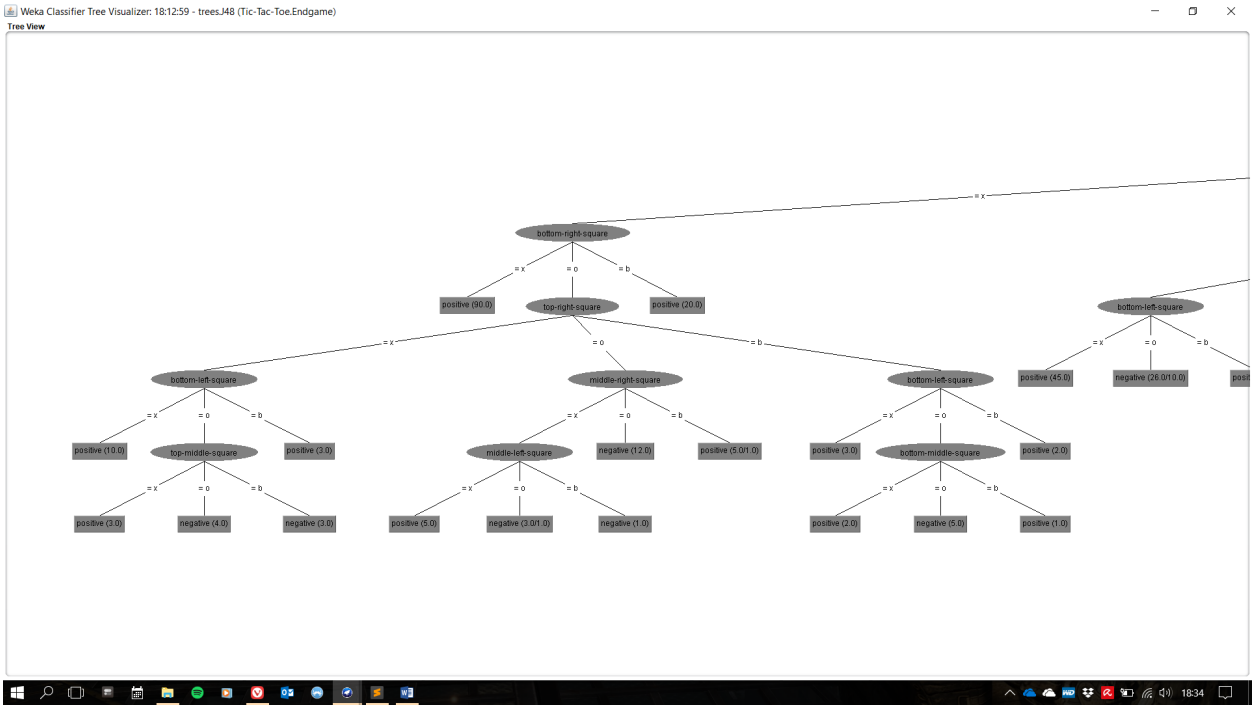
Finally, I fixed some details of the input format and I tested it first in WEKA to see how was the final tree generated. I noticed there were some answers that were pruned, so I adapted my code for it and then tested that everything was working well in my code (i.e. no infinite loops or strange behavior).

- **Include the graphics of the trees or part of the trees you generated in WEKA and your own program. Are they different, and if so, why?**

This is the complete version:



Going to the left, when middle-middle-square =x and top-left-square =x (I noticed there are three out of the six deepest nodes in the tree, which means the longest paths to determine the endgame):



For my program, the complete output is on the ‘out’ file of the repository, however the part where middle-middle-square =x and top-left-square =x is as follows:

```
middle-middle-square: x
  top-left-square: x
    bottom-right-square: x
      ANSWER: positive
    bottom-right-square: o
      top-right-square: x
        bottom-left-square: x
          ANSWER: positive
        bottom-left-square: o
          top-middle-square: x
            ANSWER: positive
          top-middle-square: o
            ANSWER: negative
          top-middle-square: b
            ANSWER: negative
        bottom-left-square: b
          ANSWER: positive
      top-right-square: o
        middle-right-square: x
          middle-left-square: x
            ANSWER: positive
          middle-left-square: o
            bottom-left-square: x
              ANSWER: negative
            bottom-left-square: o
              ANSWER: positive
            bottom-left-square: b
```

```
        ANSWER: ?
    middle-left-square: b
      ANSWER: negative
  middle-right-square: o
    ANSWER: negative
  middle-right-square: b
    bottom-left-square: x
      ANSWER: positive
    bottom-left-square: o
      middle-left-square: x
        ANSWER: negative
      middle-left-square: o
        ANSWER: ?
      middle-left-square: b
        ANSWER: positive
    bottom-left-square: b
      ANSWER: positive
top-right-square: b
  bottom-left-square: x
    ANSWER: positive
  bottom-left-square: o
    bottom-middle-square: x
      ANSWER: positive
    bottom-middle-square: o
      ANSWER: negative
    bottom-middle-square: b
      ANSWER: positive
```

```
        bottom-left-square: b

          ANSWER: positive

    bottom-right-square: b

      ANSWER: positive
```

The outputs are mostly the same, except when there is pruning by the J48 algorithm in WEKA, in my program it continues to split and when a subset is empty it prints ‘ANSWER: ?’. But the structure and splitting is the same for both outputs.

- **Based in what you have learned so far where would you use decision trees?**

When talking about machine learning, I would use decision trees on some classification problems because the output is categorical and I could then *predict* an output given certain features. It would also be very useful for strategical decisions (even related to business or other fields), for example, given the decision tree of the Tic-Tac-Toe I could follow the paths where I have more winning results.