Programming Assignment 1: Decision Trees

Part 1: Implementation [7 points]

Your job in this exercise is to predict whether you will have a good night-out in Jerusalem for the coming New Year's Eve. Assume that you have kept a record of your previous night-outs with the following attributes.

- How densely the place is usually **occupied** {High, Moderate, Low}
- How the **prices** are {Expensive, Normal, Cheap}
- Volume of the **music** {Loud, Quiet}
- The **location** {Talpiot, City-Center, Mahane-Yehuda, Ein-Karem, German-Colony}
- Whether you are a frequent customer (**VIP**) {Yes, No}
- Whether this place has your **favorite beer** {Yes, No}
- Whether you **enjoyed** {Yes, No}

We have provided a data file (dt-data.txt) that contains the relevant records.

(a) Write a program to construct decision trees based on the idea of splitting by Information Gain.
(b) Run your program on the data file.
(c) Make a prediction for (occupied = Moderate; price = Cheap; music = Loud; location = City-Center; VIP = No; favorite beer = No).

You can write your program in any programming language. However, you will have to implement the decision tree algorithm yourself instead of using library functions. In your code, in case there is a tie between attributes, please use the order that the attributes are listed as the priority. For example, to break tie between **prices** and **location**, since **prices** are listed closer to the front of the list, use **prices** instead of **location**. Provide a description of the data structures you use, any code-level optimizations you perform, any challenges you face, and of course, the requested prediction.

It is recommended that your code should print the decision tree that it produces. If you don't use this recommended format, you need to describe the format in your report. The recommended format looks like the follows

attribute A on the 1^{st} level
  first value of attribute A: 1^{st} attribute B_1 on the 2^{nd} level
    first value of attribute B_1: 2^{st} attribute B_1 on the 3^{nd} level
    …
      $n^{th\ value}$ of attribute X: yes
  second value of attribute A: 2^{nd} attribute B_2 on the 2^{nd} level
    first value of attribute B_2: 2^{st} attribute B_2 on the 3^{nd} level
    …

The leaves are marked as "yes" or "no". For example, if we use **VIP** as the root, **prices** and **music** as the attributes on the second level corresponding to VIP={Yes, No} respectively, then it should look like the following:

VIP

```
Yes: prices
  Expensive: location
  ...
No: music
  Loud: favorite beer
    ...
  Quiet: prices
    ...
```

Part 2: Software Familiarization [2 points]

Do your own research and find out about a library function that offers a good implementation of the decision tree algorithm. Learn how to use it. Compare it against your implementation and suggest some ideas for how you can improve your code. Describe all these in your report.

Part 3: Applications [1 point]

Do your own research and describe some interesting applications of decision trees.

Submission Guidelines

In your report, please include the names of both group members and mention their individual contributions. The report should be in PDF format. Your submission should include the code as well as the report and is due before **02/07**, **11:59pm** in an archive in zip, tar.gz or tar.xz format. Your source code should have a comment line that contains both group members. Only one submission is required for each group by one of the group members. Please submit your homework on **BlackBoard** (do NOT email the homework to the instructor or the TA).