CSCI924/CSCI424 Reasoning and Learning   Autumn Session 2019

Assignment 1 (updated): (5% of the total assessment in this subject)

Submission: Submit before 5:30pm 27 October via the submission link for assignment 1 on the subjects' Moodle site.

One ZIP file needs to be submitted. The ZIP file is to contain the files question1.lisp and question2.lisp

This assignment is individual work. **Late submission and plagiarism will result in a penalty** (refer to subject outline). Non-compliance to guidelines (i.e. submission of files or file types other than those requested) will be rewarded with zero marks.

Question 1 (1.5 marks)

Create a one single file named 'question1.lisp' which defines the following functions in the Lisp programming language. Make sure that your code runs correctly under clisp. The tree parts attract equal marks.

1. Write a function (NumberOfPrimes <list>) which counts the number of primes in a (possibly nested) list.
   Example: (NumberOfPrimes '(((1)(2))(5)(3)((8)3)) returns 4.
2. Define a function (even <list>) which returns the subset (a list) of even numbers contained in a given numeric, possibly nested list. Example: (even '(1 2 (3 4) -4)) returns (2 4 -4). Note that the result must maintain the order of the even numbers in the original list, and the result must be a flat list.
3. Define a function (SumIfNot <list1> <list2>) that returns the sum of all elements in list2 that do not appear in list1. Both lists may be nested lists.
   Example: (SumIfNot '(1 8 (2)) '(1 (3 (8)) 8 9)) returns 12.

   Note: Your code will be tested on examples that may differ from the ones shown here. Hence, write your code such that it would work correctly for any list of arguments.

Question 2 (3.5 marks)

Implement the ID3 algorithm (see Decision Tree in Lecture notes) in lisp. Your code is to read from a local data file called "data.txt", then creates a nested list that corresponds to the Decision Tree for the data in data.txt.
Assume that the data file contains N entries and M columns. The columns are separated by a single space, and the rows are terminated by a single newline '\n' (unix text format). The last column shall contain the target value or class label. The first row contains the name of the attributes. You can assume that all attribute values and the targets are symbolic (ie. Not numeric). A sample data file is provided with this assignment.
Your code produces a list as output which corresponds to the DT generated. The list must follow the following syntax:

LIST = (Attribute Value LIST) | CLASS,

Where CLASS is a terminal (the leaf node) containing one of the permissible class labels (as defined in the data file).
Thus, the correct output for the sample file provided would be as follows:

```
((Dividend AboveAverage (Turnover High (NO))(Turnover Medium (Reissued Yes
(YES))(Reissued No (NO)))(Turnover Low (YES)))(Dividend Average (YES))(Dividend
BelowAverage (Reissued No (YES))(Reissued Yes (NO))))
```

Note that your code will be tested on data files that differ in content from the provided sample. Thus, you need to ensure that your code works correctly for any data file.

Use comment lines in your code to document key points of your code.

Your code should have a comment header which contains your full name and student number.

Marking guidelines:

- The correctness of your code will be assessed. In other words, the ability of your code to produce the correct output will be assessed. Hence, if you implement incomplete code that does not produce a correct output to any given input then you may be awarded zero marks.
- There are solutions in Lisp to implementing some of the functions in this assignment (i.e. the Decision Tree algorithm) on the Internet**. You are required to develop and write all of the code for this assignment by yourself. Plagiarizing code or parts thereof WILL attract zero marks!**