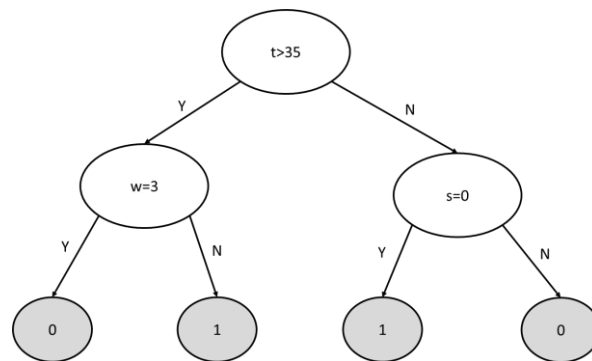


In this assignment, you will learn how to implement decision trees in C and compare their performances.

A decision tree is a simple model for classification and regression in machine learning. It gives the steps of a set of comparisons to come up with a decision on a given input. For example, if you want to build a system to automatically turn on the heater in a room based on several inputs, you can use the following decision tree.



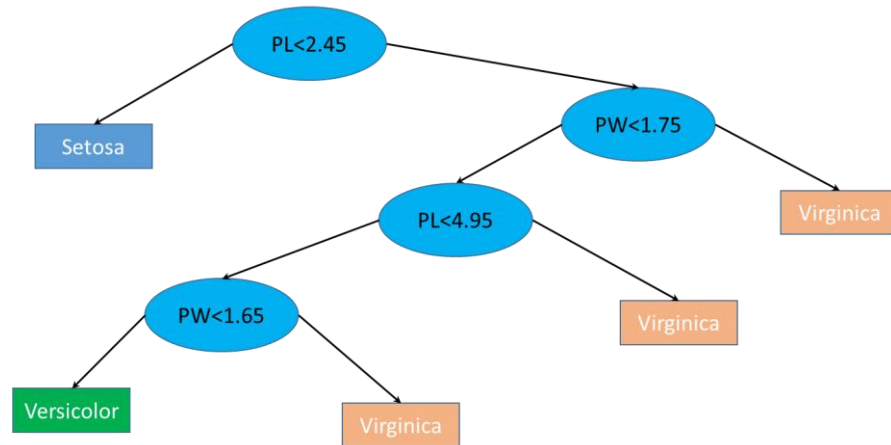
Given *temperature(t)*, *pressure(p)*, *humidity(h)*, *sunny_or_not(s)* and *day_of_the_week(w)* as your input, this decision tree answers whether or not to turn on the AC (0=turn off, 1=turn on).

Complete the code in the given project with the following tasks:

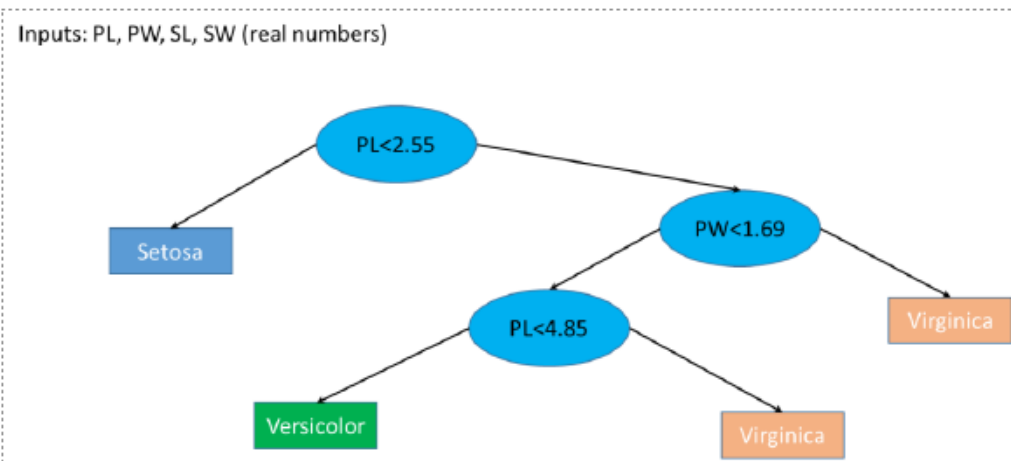
1. **main.c:** In the main function you will be solving three problems using two decision trees for each. Your program will ask which problem to solve (1, 2 or 3). Based on the answer, user's inputs will be read (as many as required with the right types). The input will be processed by the two alternative decision trees (you will implement these in functions as described below). The results will be compared and the final result will be printed. In the case of classifier (output is a number from a set of possibilities), if both answers are the same, that result will be given as the answer to the problem. If the results differ, both decisions will be reported. Similarly, for regression (the output is an ordinal number), if the two results are similar (within a threshold, defined as the constant `CLOSE_ENOUGH` in `util.h`), the average will be printed otherwise both results will be output.
2. **util.c (and util.h):** There are six functions in this file. Each pair of functions are solutions to the same problem. First four decision trees are given in the figures below. The last two are supposed to be designed by you in the following manner:
 - a. The decision trees will solve a classification problem.
 - b. It should have at least 10 decision nodes.
 - c. The input should be 5 dimensional with two real numbers and 3 categorical (one binary, the others with more than 5 possible values).
 - d. Each of these inputs should at least once used in the decision nodes.
 - e. Provide two significantly different such trees and implement them in `dt3a` and `dt3b`.

The first decision tree for P1:

Inputs: PL, PW, SL, SW (real numbers)

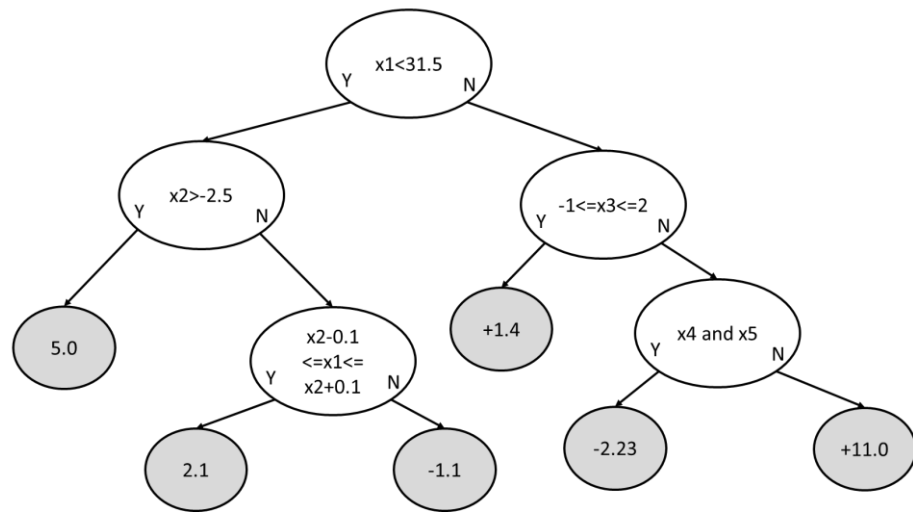


The second decision tree for P1:



The first decision trees for P2:

Inputs: x_1, x_2, x_3 (real numbers) and x_4, x_5 (binary values)



The second decision trees for P2:

Inputs: x_1, x_2, x_3 (real numbers) and x_4, x_5 (binary values)

