HW04: Decision Trees
See the associated Dropbox for due date.
Thomas Kinsman

Homework is to be programmed only in one of the following languages. No other languages will be accepted. Please limit yourself exclusively to: Java, Python, Matlab, or R. The last three had good native graphics and plotting support.

Assume that the grader has no knowledge of the language or API calls, but can read comments. Use prolific block comments before each section of code, or complicated function call to explain what the code does, and why you are using it. Put your name and date in the comments at the heading of the program.

Hand in:

1. Your write-up,
2. Your well commented code,
3. Your results your classification results of the test_data.csv file.

You are provided with a file of training data. This data has only three attributes to select from.

Using the training data, your goal is to write one decision tree that uses only three if statements to classify the results. Each if statement can only test one attribute at a time. (No feature generation allowed.)

The decision tree training program:
Name this program HW04_LastName_FirstName_Trainer…
You write a program that creates a decision tree, using the decision tree algorithms we discussed in class. Use any measure of purity you want. The output of your program is another program.

This program must be able to read in the *.csv file that is used to train the decision tree. So, the input to the decision tree training program is a flat file.

If you alter the training file, by deleting the header line, then turn it back into the dropbox for training.

For simplicity's sake, you can write this program to make hard-coded assumptions. It might assume that the there are only numeric attributes (which is true) and that there are only three attributes (which is also true). So, don't feel like you need to write a generic decision tree training program.

Your decision tree program can use any measure of purity you would like to use: Missclassification Error, Gini Index, Entropy, Information Gain, maximum purity of a node, or any of the others listed in the notes.

The goal is to get the best accuracy on your *testing* data as possible. But, of course, you don't know the class of the testing data. (That's the point of classification.)

Again the output of this program is the core of another program, to be called
HW04_LastName_FirstName_Classifier…

You program should actually generate the other program, it is not hard to do so. However, if you get stuck, your program might just print out the resulting attribute and threshold choices. (Keep reading.)

The classifier program
HW04_LastName_FirstName_Classifier…

This resulting decision tree classifier program looks somewhat like this. You can use at most three if statements, and they must have the following structure.

```
…. header and prolog …

Given one parameter --
the string containing the filename of the training file.

read in the training file
for each line in the test_data:
   if ( attribute-a >= threshold-a )
      if ( attribute-b >= threshold-b )
            class = [ 0 or 1], you choose;
      else
            class = [ 1 or 0], you choose;
   else
      if ( attribute-c >= threshold-c )
            class = [ 0 or 1], you choose;
      else
            class = [ 1 or 0], you choose;
print out the class value for each line of the test data file.
```

The goal of your decision tree training program is to select attribute-a and threshold-a, then attribute-b and threshold-b, and then attribute-c and threshold-c.

You will use the data in the *training_data* file to write this program.
Then you will run this program on the *test_data* file, and guess the classification of each entry in the *test_data*.

Then you will run the classifier, have it print out one classification per line, so that the grader can quickly go down the list and compare to his classifications.

The output should simply be a file called **HW04_LastName_FirstName__MyClassifications.txt.**

(Keep reading…)

1. (½ pts) Read through the entire homework, and estimate how long it will take to do this homework before you start the homework. Again, this is for your education. Don't cheat. Write it down before you start coding.

2. **Files:**
   Turn in the following code items, and support files:
   a. **HW04_LastName_FirstName_Trainer…**
   b. **HW04_LastName_FirstName_Classifier…**
   c. **HW04_LastName_FirstName__MyClassifications.txt**

3. **Write-UP:**
   a. Which measure for node purity did you try?
      Misclassification Error, Gini, or what?
   b. Which measure did you finally use?
   c. How did you come to select this measure?
      Did you try several, or just the first thing that seemed to work?
   d. How did you break any ties?
   e. Did your program need to break any ties?
   f. What stopping criteria did you use?
   g. Did you do any noise-cleaning on the Training Data?
      (It does have noise in it.)
   h. What were your final values for each of the attributes and thresholds?

	Attribute Number Used	Threshold Used
a		
b		
c		

   i. What was the accuracy of your resulting classifier, on the training data?
      (This is what you are trying to maximize.)
   j. Did your program actually create the classifier program, or did it just generate the attribute list and thresholds?
   k. What was the hardest part of getting all this working?
   l. Did anything go wrong?

4. (½) How many hours did this actually take?

5. (2 pts) BONUS:
   Write a function called **HW04_LastName_FirstName_GenerateScatterPlots…**
   Plot a scatter plot of each attributes versus all other attributes. This is attribute 1 vs 2, 2 vs 3, and 1 vs 3.
   Do this using a program API. Do not use Excel.

   You might want to generate this first, to gain some insights into the data.