HW 04: 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 The accepted. Please limit yourself exclusively to: Java, Python, Matlab, or R. 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 – in one ZIP file:

1. Your write-up as a PDF file,
2. Your well commented code,
3. Your results your classification results of the test file.

You are provided with a file of training data. This data has *only four* attributes to select from. The data is completely separable. No noise is added.

Using the training data, your goal is to write a program to create a decision tree. You must use binary splits that use the given attributes (the splits must be axially aligned). Each "if statement" can only test one attribute at a time. (No feature generation allowed.)

The decision tree training program:
Name this program HW_04_LastName_FirstName_Trainer…
You write a program that creates a decision tree, using the decision tree algorithms we discussed in class.

Use the Mixed-Entropy as a measure of purity, for consistency grading.

The output of your program goes into 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 four attributes.

So, don't feel like you need to write a generic decision tree training program.

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.)

(continued)

You want to submit second program: called **HW_04_LastName_FirstName_Classifier…**

The classifier program
HW_04_LastName_FirstName_Classifier…

This resulting decision tree classifier program looks somewhat like this.

```
….  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;

   # OR YOU COULD WRITE:
   if ( attribute-a >= threshold-a )
      class = [ 0 or 1], you choose;
   else
      if ( attribute-b >= threshold-b )
            class = [ 0 or 1], you choose;
      else
            class = [ 1 or 0], you choose;
      etc…

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, and so on.

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.

(Keep reading…)

1. **Files to Submit:**
   Turn in the following code items, and support files IN ONE ZIP FILE:
   a. **HW_04_LastName_FirstName_Trainer…**
   b. **HW_04_LastName_FirstName_Classifier…**
   c. **HW_04_LastName_FirstName_MyClassifications.txt** ✐ **YOUR RESULTS!!!**
   d. **HW_04_LastName_FirstName_Writeup.pdf** ✐ **PDF DOCUMENT or *.DOCX.**

2. **Rubrik:**
   a. Well-commented code that generates a resulting classifier is 5 pts.

3. **Write-UP:**
   a. Did your training program need to break any ties? If there were ties, how would your training program break them? (½) .**There were ties while calculating misclassification for B and D i.e. for attribute B and D, we have same misclassification rate for their respective thresholds. In such cases we can pick up any one of the two values.**

   b. What stopping criteria did you use? How did you make the training program terminate? Did you have to hard code it? (½) **Stopping condition for recursion is to stops when the data under consideration is less than 10% of what was given. We have to hardcode the condition but it is calculated using length of the file. So it can work for any other file as well (the other file can have any number of rows but structure has to be the same).**

   c. What structure did your final classifier have? What was the if-else tree you got?
   The grader might not want to look at any code. Please repeat the final result here. (2)

```
If (Attr2<7)
{
        If (Attr2<5)
        {
                If (Attr1<4)
                {
                        Class=0;
                }
                Else
                {
                        Class=1;
                }
        }
        Else
        {
                Class=1;
        }
}
Else
{
        Class=1;
}
```

d. Run the training data back through your classifier.
   What was the accuracy of your resulting classifier, on the training data? (½)

   Running the classifier on the training data, we get accuracy of approximately 46%.

e. Did your program actually create the classifier program, or did it just generate the attribute list and thresholds? (½)

   Yes the program does create a decision tree. This decision tree is sent to our classification file and the test data is run over that tree.

f. Classification accuracy (TP+TN) of test data was over 95%? (1)
   As we are doing a binary split , we are checking just the attributes with best info gain and assign for class value on the basis of only two attributes at a time. Hence the accuracy of test data is as low as approximately 46%.