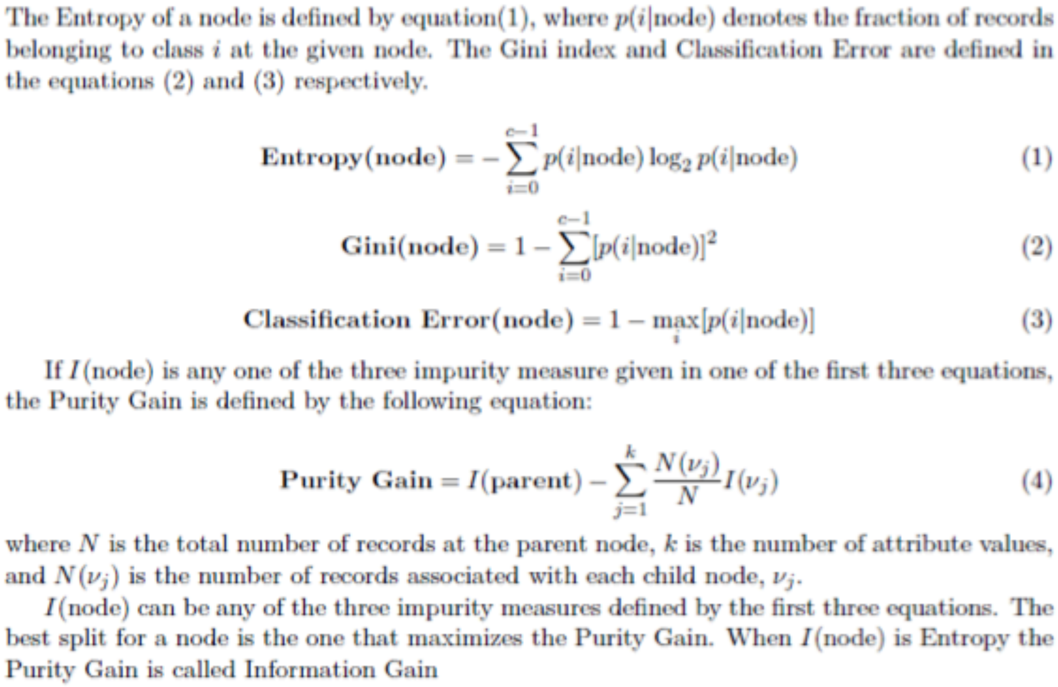
Homework on Trees
Homework 2 Patricia Hoffman, PhD.

Participants: Daniel Charlu, Gustavo Rearte, Saurabh Madaan, Sabesan Saidapet Pachai, Huong Hoang, Jinhua Li, Steve Banville



The Entropy of a node is defined by equation(1), where $p(i|\text{node})$ denotes the fraction of records belonging to class i at the given node. The Gini index and Classification Error are defined in the equations (2) and (3) respectively.

$$\textbf{Entropy(node)} = -\sum_{i=0}^{c-1} p(i|\text{node}) \log_2 p(i|\text{node}) \quad (1)$$

$$\textbf{Gini(node)} = 1 - \sum_{i=0}^{c-1} [p(i|\text{node})]^2 \quad (2)$$

$$\textbf{Classification Error(node)} = 1 - \max_i [p(i|\text{node})] \quad (3)$$

If $I(\text{node})$ is any one of the three impurity measure given in one of the first three equations, the Purity Gain is defined by the following equation:

$$\textbf{Purity Gain} = I(\textbf{parent}) - \sum_{j=1}^{k} \frac{N(\nu_j)}{N} I(\nu_j) \quad (4)$$

where N is the total number of records at the parent node, k is the number of attribute values, and $N(\nu_j)$ is the number of records associated with each child node, ν_j.

$I(\text{node})$ can be any of the three impurity measures defined by the first three equations. The best split for a node is the one that maximizes the Purity Gain. When $I(\text{node})$ is Entropy the Purity Gain is called Information Gain

1) This question uses the following ages for a set of trees: 19,23,30,30,45,25,24,20. Store them in R using the syntax ages<-c(19,23,30,30,45,25,24,20).

a) Compute the standard deviation in R using the sd() function. Also compute the mean and median.

```
> sd(ages)
[1] 8.315218
> mean(ages)
[1] 27
> median(ages)
[1] 24.5
```

b) Compute the same value in R without the sd function.

Using the variance function:

```
> sqrt( var(ages) )
 [1] 8.315218
```

Using the formula for sample standard deviation:

```
> sqrt( sum( (ages-mean(ages))^2 ) / (length(ages)-1))
[1] 8.315218
```

c) Using R, how does the standard deviation from part a) change if you add 10 to all the values?

```
> sd(ages+10)
[1] 8.315218
```

It stays the same.

d) Using R, how does the standard deviation in part a) change if you multiply all the values by 100?

```
> sd(ages * 100)
[1] 831.5218
```

It increases.

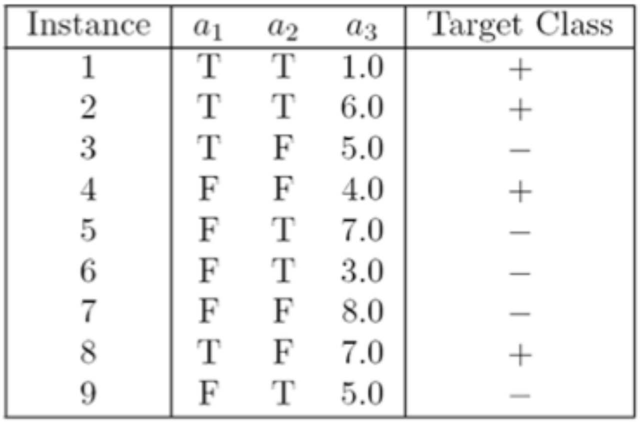
e) Next add another tree of age 70 to the sample. Compute the mean and median with this tree added to the sample. How have the mean and median changed?

```
> newTree <- 70
> mean(   c(ages, newTree) )
[1] 31.77778
> median( c(ages, newTree) )
[1] 25
```

The mean increased by about 4.8 and the median increased by 0.5. The mean is more sensitive to outliers.

2) For binary classification, consider the training examples in the table below (shown in Table 4.8 from the book on page 198). For this problem ignore columns a_1 and a_2. For column a_3, which is a continuous attribute, compute the Information Gain (using entropy as the purity measure) for every possible split. Where would be the

best place to split the attribute a_3? Next compute the classification error for every possible split of that same variable. Using classification error as the impurity measure, where would be the best place to split the attribute a_3? Is there a difference in where to make the split? (You can code this in r or you can do this by hand.)



Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

```
> binClass
  a3 TargetClass
1  1           +
6  3           -
4  4           +
3  5           -
9  5           -
2  6           +
5  7           -
8  7           +
7  8           -
> pg_resdf[ order(pg_resdf$PurityGain), ]
  SplitVal  PurityGain
1        1 0.000000000
3        4 0.002565287
5        6 0.007214618
6        7 0.018310782
4        5 0.072780226
7        8 0.102187171
2        3 0.142690279
> ce_resdf[ order(ce_resdf$CError), ]
  SplitVal    CError
2        3 0.3333333
4        5 0.3333333
1        1 0.4444444
```

```
5       6 0.4444444
6       7 0.4444444
7       8 0.4444444
3       4 0.4444444
```

The highest purity gain is 0.14 for a split of “3” (higher PG is better)
The lowest Classification Error (CE) is 0.33 for a split of “3” or “5” (lower CE is better)

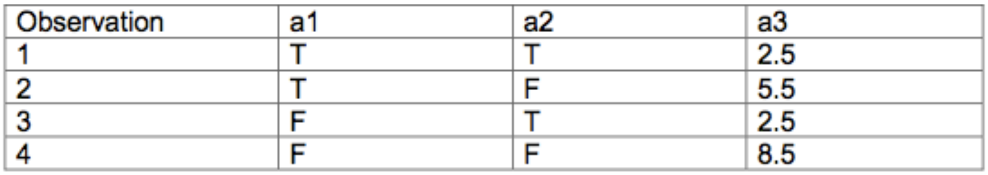
These two methods roughly agree.

3) The following tree was created using rpart for the table given in this homework problem number one.

```
node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 9 4 0 (0.5555556 0.4444444)
 2) a1< 0.5 5 1 0 (0.8000000 0.2000000)          -
   4) a2>=0.5 3 0 0 (1.0000000 0.0000000) *      -
   5) a2< 0.5 2 1 0 (0.5000000 0.5000000)        -
    10) a3>=6 1 0 0 (1.0000000 0.0000000) *      -
    11) a3< 6 1 0 1 (0.0000000 1.0000000) *      +
 3) a1>=0.5 4 1 1 (0.2500000 0.7500000)          +
   6) a2< 0.5 2 1 0 (0.5000000 0.5000000)        -
    12) a3< 6 1 0 0 (1.0000000 0.0000000) *      -
    13) a3>=6 1 0 1 (0.0000000 1.0000000) *      +
   7) a2>=0.5 2 0 1 (0.0000000 1.0000000) *      +
```

Use this tree to predict the class labels (either a + or -) for the following test observations:

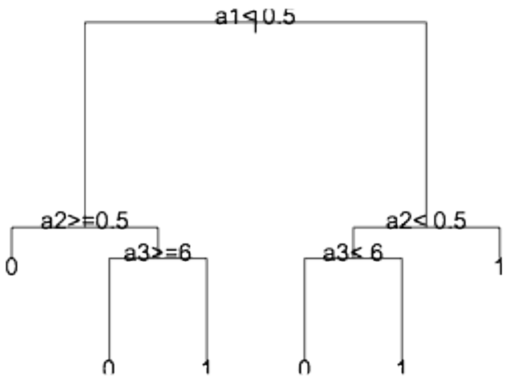


Observation	a1	a2	a3
1	T	T	2.5
2	T	F	5.5
3	F	T	2.5
4	F	F	8.5

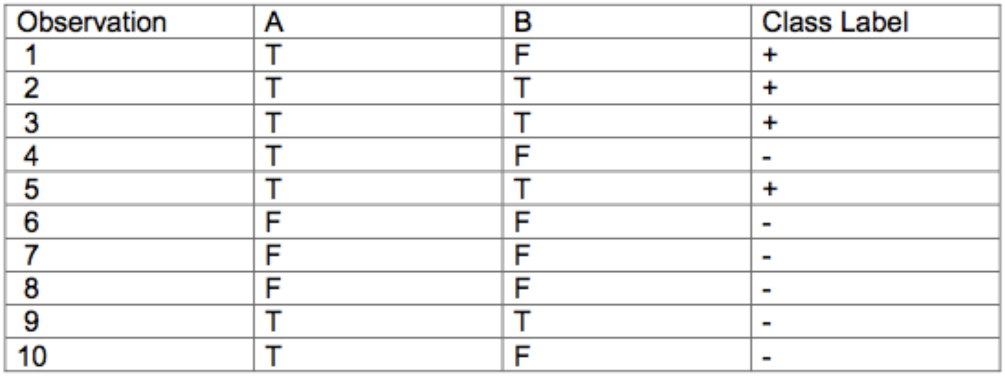
NOTE: I annotated the “rpart” output above, indicating which are “+” (1) and which are “-“ (0). Below are the answers for each of the observations:

```
1: +  (node path: 3, 7)
2: -  (node path: 3, 6, 12)
3: -  (node path: 2, 4)
4: -  (node path: 2, 5, 10)
Below is the tree built to show the rpart output
```

graphically. In the tree below “+” is reprepresented by “1” and “-“ by “0”



4) Consider the table given in the text on page 200 in the book exercise number five (copied below). It is a binary class problem. Would it be possible to create a model which would correctly classify this training data? If it is possible create a tree which gives the correct answer (either + or -) for each training observation. Otherwise, give the reason that it is not possible to do so.



Observation	A	B	Class Label
1	T	F	+
2	T	T	+
3	T	T	+
4	T	F	-
5	T	T	+
6	F	F	-
7	F	F	-
8	F	F	-
9	T	T	-
10	T	F	-

No, it is not possible to create a training model to correctly predict this training data. The possibilities would be to train on A, B, or A & B. However, in all 3 cases, rows 1 and 4 have the same input but different class labels so they are contradictory. For example:

Obs 1: A = T, B = F, Class = +
Obs 4: A = T, B = F, Class = -

No model can predict which one it should be.

In some situations, this may not be obvious by inspection, in which case one could use “rpart” to build trees to various depths with various parent nodes to show this can’t be done. Here is the result of running R code to see this:

```
> result
  tree_depth error_w_A error_w_B error_w_AB
1          1       0.3       0.2        0.2
2          2       0.3       0.2        0.2
3          3       0.3       0.2        0.2
4          4       0.3       0.2        0.2
5          5       0.3       0.2        0.2
```

The above output shows the error results by looking at 3 different models, “y ~ A”, “y ~ B”, and “y ~ A + B”. It can be seen that the error is non-zero for all models at depths 1 to 5.

5) The UC Irvine web site has many interesting data sets. Sonar data is described at the web site: http://archive.ics.uci.edu/ml/machine-learning-databases/undocumented/connectionist-bench/sonar/sonar.names The sonar data set can be found at http://archive.ics.uci.edu/ml/machine-learning-databases/undocumented/connectionist-bench/sonar/ Divide the sonar data set into a training set (sonar_train.csv) and a test set (sonar_test.csv). Use R to compute the classification error on the test set when training on the training set for a tree of depth 5 using all the default values except control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5). Remember that the 61st column is the response and the other 60 columns are the predictors. Documentation for the rpart package can be found at http://cran.r-project.org/web/packages/rpart/rpart.pdf

Classification Error on training set:

```
> errVal_train
[1] 0.009615385
```

Classification Error on test set:

```
> errVal_test
[1] 0.3461538
```

The classification error on the training set (half of the data randomly picked), was 0.1% for the training data but 35% on the test data. This would suggest overfitting. Using cross validation would likely help reduce this error.

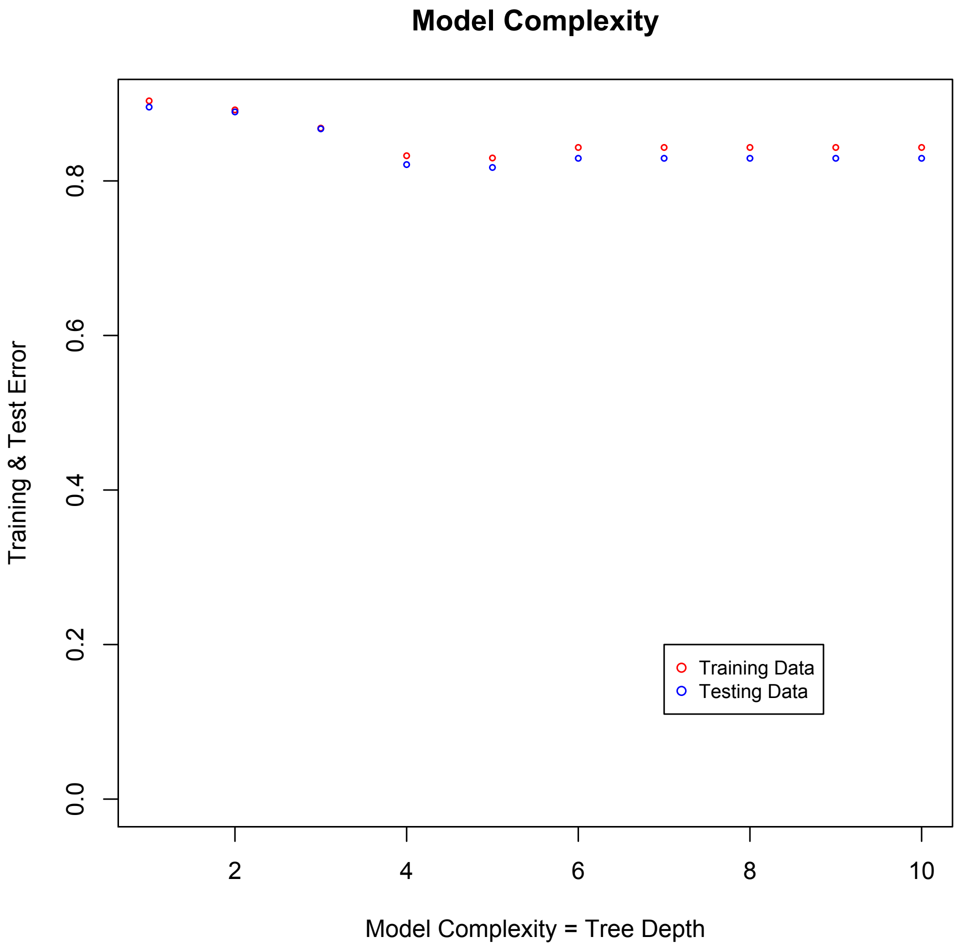
6) Check out the web page which describes a wine quality data set:

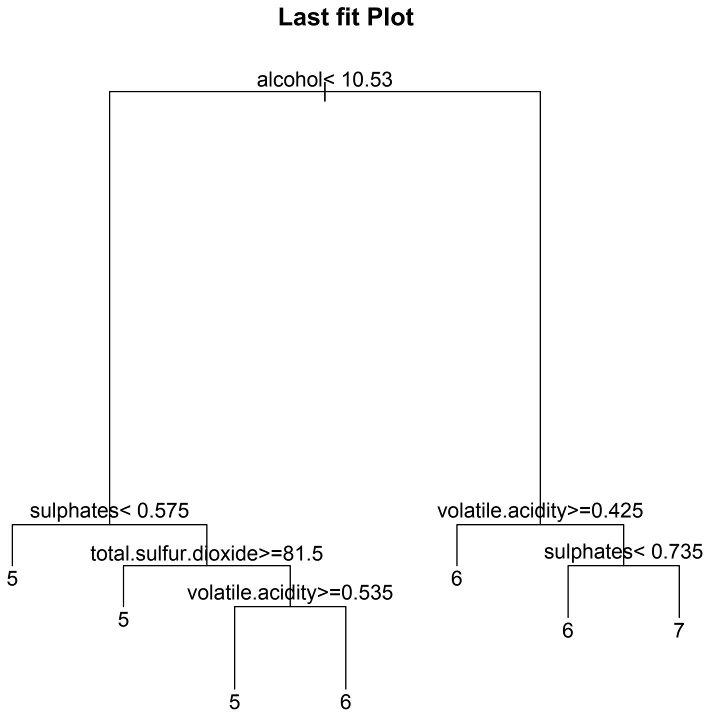
http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality.names

Use the Red Wine data set: winequality-red.csv This data set contains 1599 observations of 11 attributes. The median score of the wine tasters is given in the last column. Note also that the delimiter used in this file is a semi colon and not a comma. Use rpart on this data to create trees for a range of different tree depths. Use cross validation to generate training error and test error. Plot these errors as a function of tree depth. Which tree depth results in the best Test Error? What is that Test Error? Hint: look at the cross validation example given in the lecture.

A minimum error was seen at a tree depth of 5. The Test Error was 0.82.

Below is a “Tree Depth” vs. “Training & Test Error” plot. This shows the Error for both the Training and Testing data. The plot after that shows the last fit that was created just to show an example of a tree built in this process.


Model Complexity
Training & Test Error
0.0
0.2
0.4
0.6
0.8
2
4
6
8
10
Model Complexity = Tree Depth
Training Data
Testing Data



Code is in ".R" file.