**Assignment 2**

**Decision Tree**

**Overview:**

Our second assignment was to design and implement a decision tree to predict abalone age using the abalone data set from the UC Irvine machine learning repository. We were also give the option to use the existing Weka software to create a decision tree classifier, which is the route I chose.

**Preprocessing:**

The first step in using machine learning techniques with large amounts of data is preprocessing. Since part of the benefit of utilizing the Weka classifiers is that tree split points for continuous values are determined for you, the bulk of my data processing consisted of determining how to group the number of rings, which was the target prediction for age.

I used a visualization (Figure 1) of the distribution of the number of rings (y-axis) to determine age cut off points.



**Figure 1: Distribution of Rings**

Multiple ranges were tested with varying results. Initally, I experimented with having a more even distribution, categorizing the first 1000 samples and YOUNG, the middle 2000 samples as ADULT, and the remaining 1000 samples as OLD. This resulted in a mediocre prediction rate. The final distribution better mirrored actual I looked around for information on the actual life cycles of abalone; I found that abalone mature around 6 years, <http://www.vada.com.au/Anatomy.html>. Table 1 provides the best performing age cutoffs.

**Table 1: Age Cutoffs**

|  |  |  |
| --- | --- | --- |
| **YOUNG** | **ADULT** | **OLD** |
| 0-7 | 8-14 | 15+ |

The second step was to incorporate the .arff file format for Weka. A header was added naming each column as an attribute and specifying it as numeric or nominal (a list of values such as M, F or I). The start of the data, which had to be comma delimitated, was indicated.

**Training and Testing:**

For training a 10-fold cross-validation was used. Weka randomly reordered the dataset and then split it into 10 folds of equal size. “In each iteration, one fold is used for testing and the other n-1 folds are used for training the classifier. The test results are collected and averaged over all folds. This gives the cross-validation estimate of the accuracy.” (Weka Manual for Version 3-6-8, p 16)

For testing, a model is created during the testing phase. “Only the model on the training set is saved, not the multiple models generated via cross-validation.” (Weka Manual for Version 3-6-8, p 19) A testing data set, of 500 randomly chosen instances as per assignment directions, was fed into the model and the results displayed. I computed the accuracy, which was 82.8%

**Analysis and Visualization:**

As a lower bound, I ran the test data through the ZeroR classifier, which just assigns the most common class to every instance. For a decision tree classifier to prove worthwhile it would have to perform better than that. The results of the decision tree classifiers created by Weka are provided in Table 2.

**Table 2: Pruned vs Unpruned Performance**

|  |  |  |
| --- | --- | --- |
|  | **Accuracy %** | **Size of Tree** |
| **ZeroR** | 73.6716 |  |
| **Pruned** | 84.0833 | 8 |
| **Unpruned** | 83.82 | 163 |

The difference in accuracy performance for the pruned and unpruned trees was negligible. Of course, the difference in size of the trees was significant, which is more important for processing the test data. Table 2 shows the difference between the two. Table 3 shows the difference in parameters when running the two classifiers. I simply used the parameters set forth in the Data Mining for Abalone report, and was not able to improve on their performance using different parameters.

**Table 3: Pruned vs Unpruned Parameters**

|  |  |  |
| --- | --- | --- |
|  | **Confidence Factor** | **Minimum Number of Object** |
| **Pruned** | 3 | 30 |
| **Unpruned** | 0.25 | 2 |

Figure 2 displays a visualization of the pruned tree. “The numbers in (parentheses) at the end of each leaf tell us the number of examples in this leaf. If one or more leaves were not pure (= all of the same class), the number of misclassified examples would also be given, after a /slash/”



**Figure 2: Pruned Tree Visualization**

**Conclusion:**

Weka provides a powerful tool for data mining and machine learning by incorporating preprocessing and visualization techniques with powerful machine learning techniques. By utilizing the available java code, this tool can greatly streamline the machine learning process.

I was successfully able to analyze the abalone data and prepare it for the decision tree classifier. The parameters I fed into Weka produced a classifier that worked better than the lower bound algorithm and properly implementing the pruning parameters allowed for a tree that would be very practical when testing extremely large data sets.

**References:**

<http://maya.cs.depaul.edu/classes/ect584/weka/classify.html>

COMPUTER SCIENCE 4TF3 PROJECT Data Mining for Abalone

Weka Manual for Version 3-6-8