Assignment 2

Sam Delaney CS 773

For this assignment we were to develop a decision tree classifier that would learn the Abalone data set provided by UC Irvine, and then run that classifier on a random set of 500 abalone to determine their age (aka number of rings). To do this I choose to use WEKA in order to build a decision tree classifier for me, and to provide awesome pictures along the way. The first step in this process was to decide how best to discretize my data, as the J48 algorithm which is the C4.5 algorithm for WEKA, requires nominal attributes to create a tree. Conveniently the attribute concerning the sex of the abalone was already done providing a starting point for the rest of the values to be sorted by. WEKA provides excellent tools in the creation of bins in which to sort data. Using unsupervised discretization I was able to arbitrarily sort my data into 4 bins each, which while not useful for predicting age, would come in handy for allowing me to pick my training and test sets. Using these bins I then trained my tree to see which of the folds would produce the highest success rate using a simple 4 bin classifier. Below was the best result I got of 76.5%

Correctly Classified Instances 3199 76.5861 %

Incorrectly Classified Instances 978 23.4139 %

Kappa statistic 0.5078

Mean absolute error 0.1693

Root mean squared error 0.2917

Relative absolute error 64.7806 %

Root relative squared error 80.7264 %

Total Number of Instances 4177



Using this, I then created my files, my training file abaoloneTrain.arff, and my test file abaloneTest.arff. This is where things started to get a little more elaborate. With my training set in hand I began the tedious process of tweaking my bins for each category to maximize classification accuracy. WEKA luckily has a built in filter to take care of this for us. Using the discretize filter we can then set the number of bins for each attribute. We can also assign weights to each bin. Using the unsupervised discretization I was then able to tune each category to increase my accuracy. This proved to be more difficult as some of the attributes had plenty of outliners, or heavily biased values. This required plenty of tweaking to get the best classifier possible. While WEKA itself can setup bins I had issues with getting them to improve accuracy as opposed to losing accuracy.



Finally once I was happy with the discretization of all the attributes it was time to save out the tree for use with the test set. The best use of WEKA is here were it will automatically prune the tree for us. The rules and table below was my resulting tree and rule set and its accuracy run on a random sample of 500 abalone.

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 2804 67.1295 %

Incorrectly Classified Instances 1373 32.8705 %

Kappa statistic 0.3382

Mean absolute error 0.1454

Root mean squared error 0.2716

Total Number of Instances 4177

 My success rate was only 67% but by increasing the number of bins for my rings I was able to get a close range for age, as opposed to getting a broad age range for a final age prediction. With a larger tree it might have been possible to get a better accuracy but, from my discretion values I kept loosing accuracy as my rules became too specific.

J48 pruned tree

------------------

Shellweight = '(-inf-0.16875]'

| Length = '(-inf-0.198333]': '(-inf-5.666667]' (43.0/3.0)

| Length = '(0.198333-0.321667]'

| | Diameter = '(-inf-0.154167]': '(-inf-5.666667]' (17.0/1.0)

| | Diameter = '(0.154167-0.253333]'

| | | Sex = M: '(5.666667-10.333333]' (35.0/12.0)

| | | Sex = F: '(5.666667-10.333333]' (5.0/2.0)

| | | Sex = I: '(-inf-5.666667]' (181.0/89.0)

| | Diameter = '(0.253333-0.3525]': '(5.666667-10.333333]' (2.0)

| | Diameter = '(0.3525-0.451667]': '(-inf-5.666667]' (0.0)

| | Diameter = '(0.451667-0.550833]': '(-inf-5.666667]' (0.0)

| | Diameter = '(0.550833-inf)': '(-inf-5.666667]' (0.0)

| Length = '(0.321667-0.445]': '(5.666667-10.333333]' (712.0/92.0)

| Length = '(0.445-0.568333]': '(5.666667-10.333333]' (437.0/53.0)

| Length = '(0.568333-0.691667]': '(5.666667-10.333333]' (3.0/1.0)

| Length = '(0.691667-inf)': '(5.666667-10.333333]' (0.0)

Shellweight = '(0.16875-0.336]'

| Shuckedweight = '(-inf-0.248833]'

| | Sex = M

| | | Wholeweight = '(-inf-0.472583]': '(5.666667-10.333333]' (3.0/1.0)

| | | Wholeweight = '(0.472583-0.943167]': '(10.333333-15]' (66.0/25.0)

| | | Wholeweight = '(0.943167-1.41375]': '(10.333333-15]' (0.0)

| | | Wholeweight = '(1.41375-1.884333]': '(10.333333-15]' (0.0)

| | | Wholeweight = '(1.884333-2.354917]': '(10.333333-15]' (0.0)

| | | Wholeweight = '(2.354917-inf)': '(10.333333-15]' (0.0)

| | Sex = F: '(10.333333-15]' (75.0/33.0)

| | Sex = I: '(5.666667-10.333333]' (65.0/20.0)

| Shuckedweight = '(0.248833-0.496667]': '(5.666667-10.333333]' (1226.0/450.0)

| Shuckedweight = '(0.496667-0.7445]'

| | Length = '(-inf-0.198333]': '(5.666667-10.333333]' (0.0)

| | Length = '(0.198333-0.321667]': '(5.666667-10.333333]' (0.0)

| | Length = '(0.321667-0.445]': '(5.666667-10.333333]' (0.0)

| | Length = '(0.445-0.568333]': '(5.666667-10.333333]' (19.0)

| | Length = '(0.568333-0.691667]'

| | | Sex = M: '(5.666667-10.333333]' (171.0/39.0)

| | | Sex = F: '(5.666667-10.333333]' (130.0/41.0)

| | | Sex = I: '(10.333333-15]' (12.0/3.0)

| | Length = '(0.691667-inf)'

| | | Wholeweight = '(-inf-0.472583]': '(5.666667-10.333333]' (0.0)

| | | Wholeweight = '(0.472583-0.943167]': '(5.666667-10.333333]' (0.0)

| | | Wholeweight = '(0.943167-1.41375]': '(5.666667-10.333333]' (3.0)

| | | Wholeweight = '(1.41375-1.884333]': '(10.333333-15]' (3.0)

| | | Wholeweight = '(1.884333-2.354917]': '(5.666667-10.333333]' (0.0)

| | | Wholeweight = '(2.354917-inf)': '(5.666667-10.333333]' (0.0)

| Shuckedweight = '(0.7445-0.992333]'

| | Sex = M: '(5.666667-10.333333]' (2.0)

| | Sex = F: '(10.333333-15]' (3.0/1.0)

| | Sex = I: '(5.666667-10.333333]' (0.0)

| Shuckedweight = '(0.992333-1.240167]': '(5.666667-10.333333]' (0.0)

| Shuckedweight = '(1.240167-inf)': '(5.666667-10.333333]' (0.0)

Shellweight = '(0.336-0.50325]': '(10.333333-15]' (812.0/387.0)

Shellweight = '(0.50325-0.6705]': '(10.333333-15]' (130.0/59.0)

Shellweight = '(0.6705-0.83775]'

| Visceraweight = '(-inf-0.127083]': '(10.333333-15]' (0.0)

| Visceraweight = '(0.127083-0.253667]': '(10.333333-15]' (1.0)

| Visceraweight = '(0.253667-0.38025]'

| | Sex = M: '(15-19.666667]' (4.0/1.0)

| | Sex = F: '(19.666667-24.333333]' (3.0/1.0)

| | Sex = I: '(15-19.666667]' (0.0)

| Visceraweight = '(0.38025-0.506833]': '(10.333333-15]' (7.0/2.0)

| Visceraweight = '(0.506833-0.633417]': '(10.333333-15]' (2.0)

| Visceraweight = '(0.633417-inf)': '(10.333333-15]' (0.0)

Shellweight = '(0.83775-inf)': '(15-19.666667]' (5.0/2.0)

Number of Leaves : 51

Size of the tree : 64

