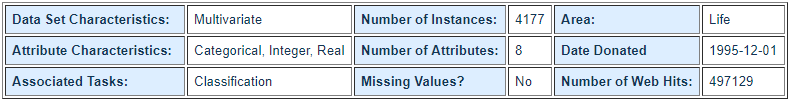
1. Describe your process and methods. What parameters (testing/training size, classification target, subtree raising etc.) did you try using with the decision tree/ what, if anything, did you do to preprocess the dataset? Why did you select those parameters/preprocessing?

**Answer:**

**Proses**

1. Download abalone dataset dari <https://archive.ics.uci.edu/>, file akan berupa .data. Pada data ini terdapat 8 atribut, 3 class, dan 4177 data. Class terdiri dari {M, F, I}, atribut terdiri dari length, diameter, height, whole weight, shucked weight, viscera weight, shell weight, rings.
2. Convert file dataset .data menjadi .arff dengan cara buka dengan notepad, tambahkan:

@relation abalone

@attribute whole\_weight real

@attribute shell\_weight real

@attribute sucked\_weight real

@attribute viscera\_weight real

@attribute rings real

@attribute height real

@attribute diameter real

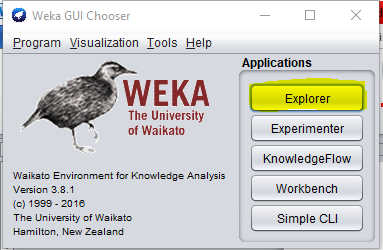
@attribute length real

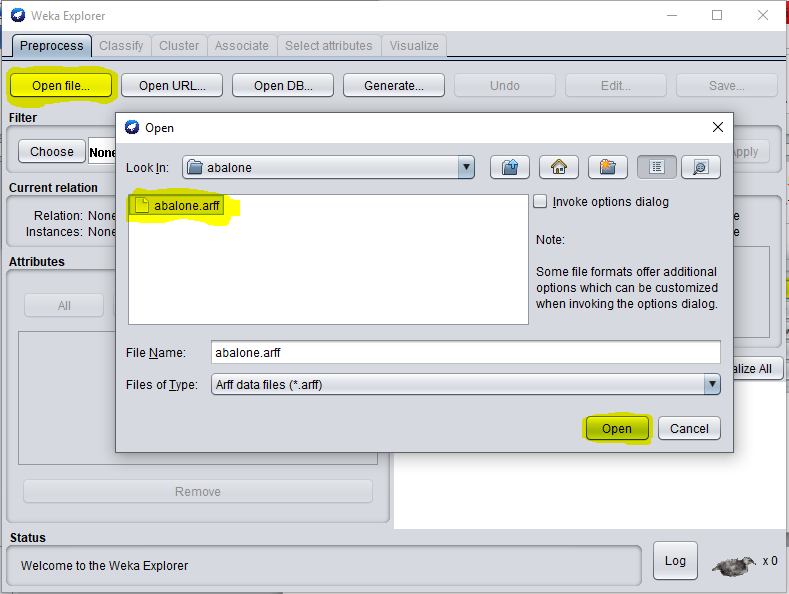
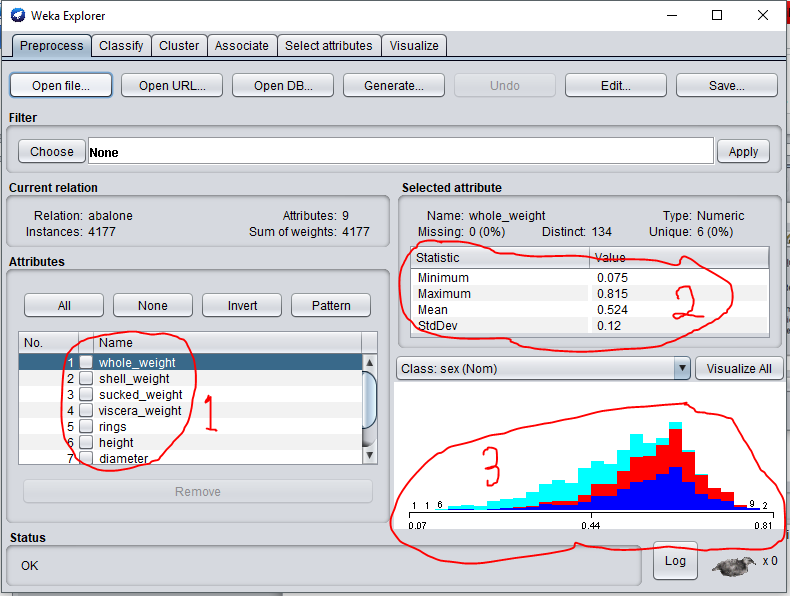
@attribute sex {M, F, I}

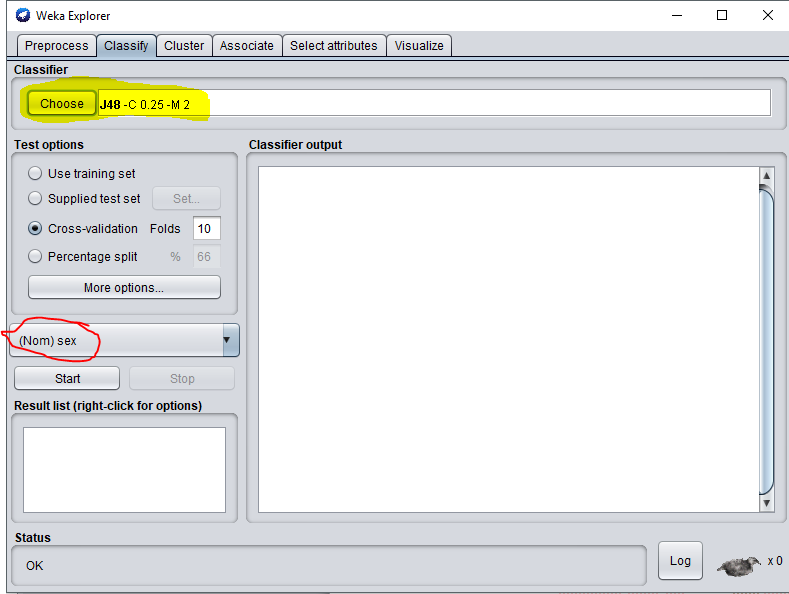
@DATA

Sebagai definisi nama setiap atributnya, urut berdasarkan kolom. Setelah itu simpan file dalam bentuk .arff. Disini sex merupakan class nya

1. Buka dataset (.arff) pada aplikasi Weka untuk diproses.
2. Pilih pada tab classify lalu pilih metode J48, pilih use training set, pilih sex sebagai class nya lalu start

 pilih explorer untuk mengolah dataset

pilih open file lalu pilih file dataset yang ingin diproses lalu open.

Setelah dibuka maka tampilannya akan seperti diatas, 1. Menunjukkan nama kolom untuk data, 2. Keterangan tiap atribut seperti nilai minimum maximum dll, 3. Meunjukkan grafik dari data pada suatu atribut.

Pilih metode yang akan digunakan dengan choose J48, pada lingkarang merah diatas untuk memilih kolom classnya. Pada test options ada beberapa pilihan yaitu

1. Use training set : semua data akan digunakan menjadi training set, setelah decision treenya ketemu maka seluruh data training yang tadi akan digunakan untuk menjadi data testing
2. Supplied test set : menggunakan file lain untuk menjadi testing set
3. Cross-validation : membagi dataset menjadi k bagian dimana nanti akan ada k kali proses dan setiap subset akan menjadi training set sebanyak k-1 kali dan menjadi testing set 1 kali. Error yang dihasilkan akan di rata-rata
4. Percentage split : mambagi data menjadi 2 bagian berdasarkan split yang diinginkan dan salah satu akan digunakan sebagai training set dan lainnya menjadi testing set

Kalau sudah memilih test options maka klik start untuk memulai, test options adalah preprocessing yang dilakukan untuk menentukan training set dan testing set.

**Metode**

1. Metode yang digunakan adalah dengan decision tree C4.5/J48 (implementasi C4.5 di Weka). Metode ini dapat mengklasifikasikan data dengan metode pohon keputusan yang memiliki kelebihan dapat mengolah data numerik (kontinyu) dan diskret, dapat menangani nilai atribut yang hilang dan menghasilkan aturan-aturan yang mudah diinterpretasikan

**Parameter/Preprocessing**

Disini saya **menggunakan cross-validation** untuk membagi data training dan data testingnya. Cross-validation sendiri aka nada pilihan berapa fold yang akan digunakan, semisal 10. Mekanismenya yaitu data akan dibagi menjadi k buah subset dimana k adalah nilai dari fold. Selanjutnya untuk tiap subset akan dijadikan data testing dari hasil klasifikasi yang dihasilkan dari k-1 subset lainnya. Jadi aka nada 10 kali testing dimana setiap data subset akan menjadi data testing satu kali dan menjadi data training sebanyak k-1 kali. Kemudian error dari k tes tersebut akan dihitung rata-ratanya.

Kenapa saya menggunakan cross-validation ini karena dengan setiap subset pernah menjadi data testing maka akurasi yang dihasilkan akan lebih menyeluruh karena mencakup semua data meskipun akan lebih kecil akurasinya dibandingkan dengan use training set atau percentage split.

Disini saya menggunakan k=10 karena sudah defaultnya.

1. What were your result? (overall accuracy, accuracy of each leaf/branch, false positives, false negatives, etc. on the training and testing sets). Show what decision trees you found.

**Answer:**

1. Menggunakan cross-validation

Number of Leaves : 360

Size of the tree : 719

Time taken to build model: 0.1 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 2206 52.813 %

Incorrectly Classified Instances 1971 47.187 %

Kappa statistic 0.2886

Mean absolute error 0.3405

Root mean squared error 0.4806

Relative absolute error 76.7932 %

Root relative squared error 102.0728 %

Total Number of Instances 4177

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0,506 0,336 0,465 0,506 0,485 0,168 0,602 0,437 M

0,376 0,223 0,434 0,376 0,403 0,159 0,626 0,396 F

0,702 0,156 0,681 0,702 0,691 0,542 0,797 0,606 I

Weighted Avg. 0,528 0,243 0,525 0,528 0,525 0,285 0,672 0,479

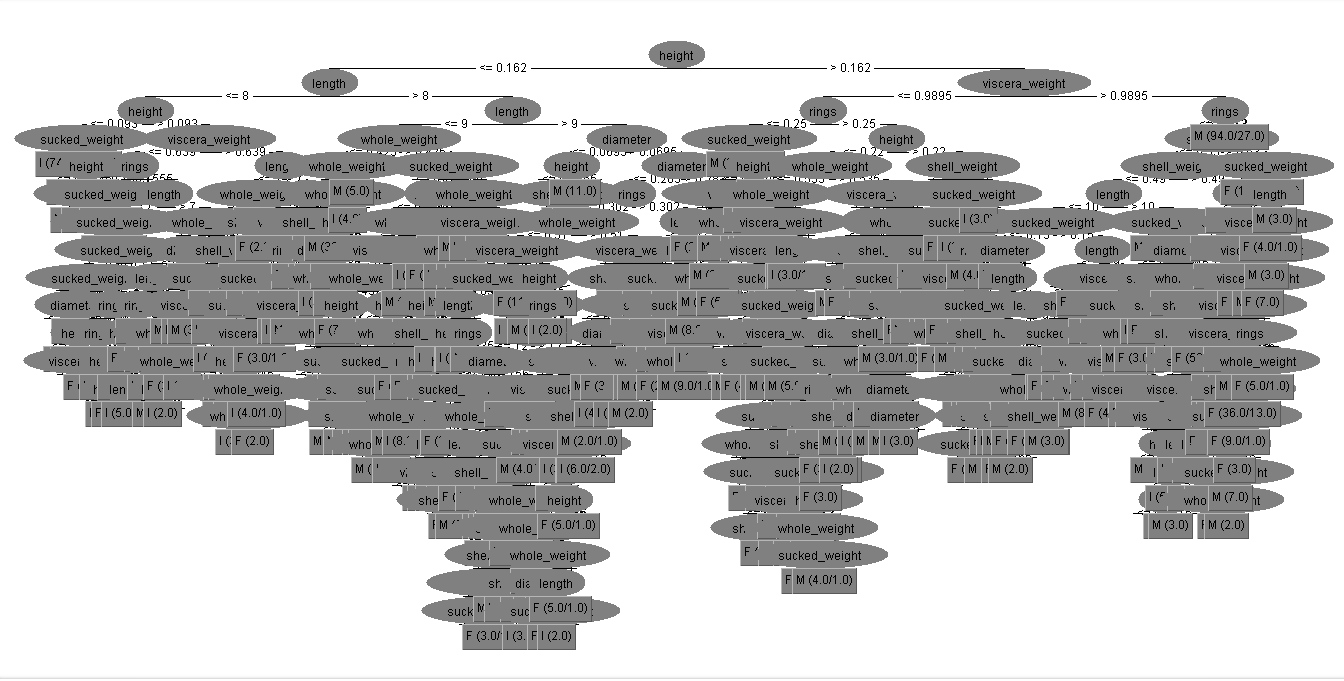
=== Confusion Matrix ===

a b c <-- classified as

773 489 266 | a = M

641 491 175 | b = F

248 152 942 | c = I

1. Menggunakan use training set

Number of Leaves : 360

Size of the tree : 719

Time taken to build model: 0.33 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.06 seconds

=== Summary ===

Correctly Classified Instances 3131 74.9581 %

Incorrectly Classified Instances 1046 25.0419 %

Kappa statistic 0.6228

Mean absolute error 0.2232

Root mean squared error 0.3341

Relative absolute error 50.3422 %

Root relative squared error 70.9523 %

Total Number of Instances 4177

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0,726 0,190 0,689 0,726 0,707 0,531 0,868 0,795 M

0,653 0,124 0,706 0,653 0,679 0,541 0,884 0,772 F

0,870 0,066 0,861 0,870 0,865 0,801 0,959 0,890 I

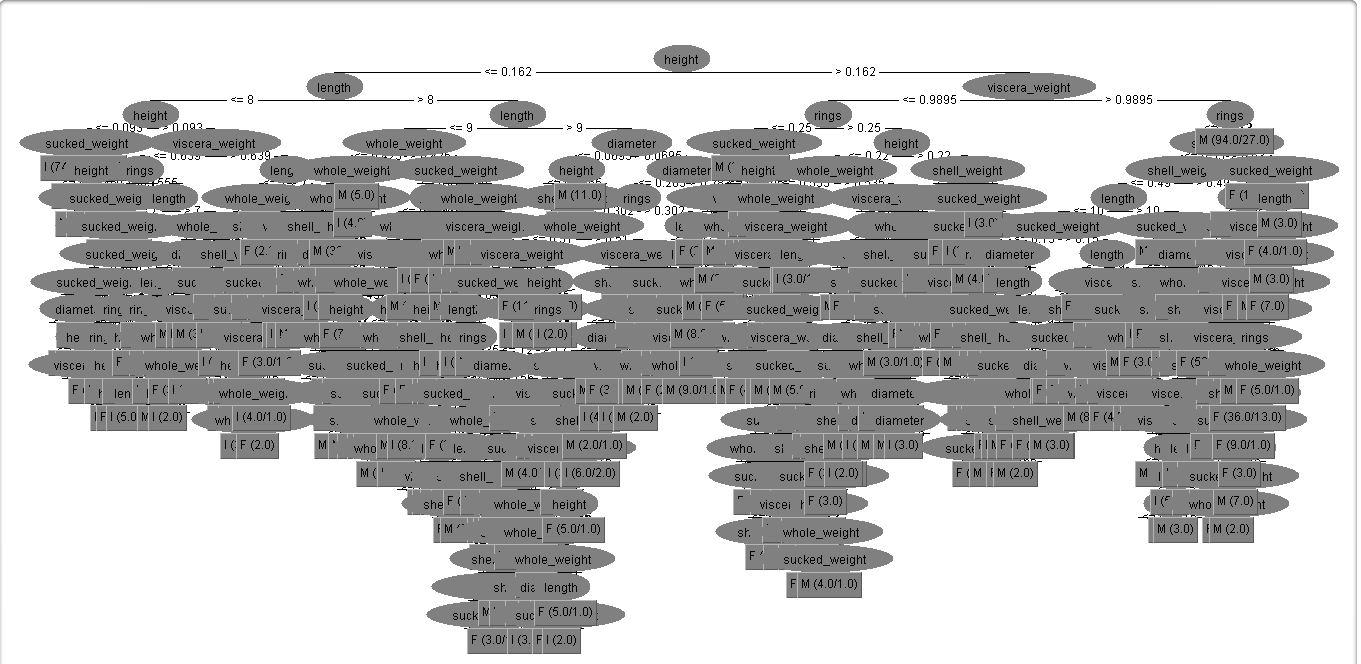
Weighted Avg. 0,750 0,129 0,749 0,750 0,749 0,621 0,902 0,818

=== Confusion Matrix ===

a b c <-- classified as

1110 295 123 | a = M

388 854 65 | b = F

 114 61 1167 | c = I

1. Menggunakan percentage split

Number of Leaves : 360

Size of the tree : 719

Time taken to build model: 0.14 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances 749 52.7465 %

Incorrectly Classified Instances 671 47.2535 %

Kappa statistic 0.2869

Mean absolute error 0.3462

Root mean squared error 0.4694

Relative absolute error 78.073 %

Root relative squared error 99.6823 %

Total Number of Instances 1420

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0,524 0,360 0,456 0,524 0,488 0,161 0,595 0,433 M

0,305 0,203 0,413 0,305 0,351 0,112 0,634 0,402 F

0,757 0,155 0,692 0,757 0,723 0,588 0,825 0,673 I

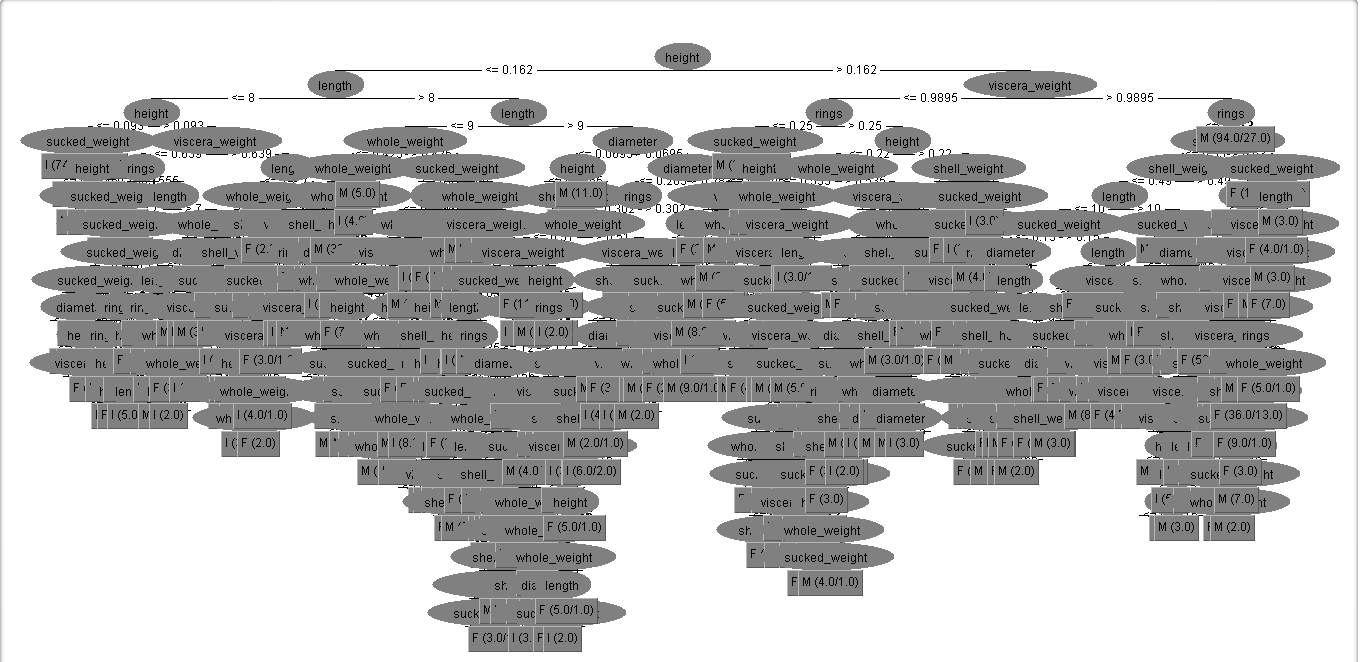
Weighted Avg. 0,527 0,245 0,517 0,527 0,518 0,280 0,680 0,499

=== Confusion Matrix ===

a b c <-- classified as

272 151 96 | a = M

260 138 55 | b = F

 64 45 339 | c = I

1. What do the result tell us? Why are the results (in)accurate? What class were some samples misclassified as, and why? Why did changing parameter(s) improve/degrade accuracy?

**Answer:**

* Hasilnya masih ada yang tidak akurat karena tree yang terbuat tidak semuanya sesuai dengan training set, sehingga akurasi yang dihasilkan akan tidak 100% akurat.
* a b c <-- classified as

773 489 266 | a = M

641 491 175 | b = F

248 152 942 | c = I

Hal ini terjadi karena tree yang terbentuk juga tidak terlalu sesuai dengan data trainingnya, dan data testing sendiri berbeda dari data trainingnya

* Mengganti parameter bisa jadi memperbaiki akurasi yang dihasilkan sehingga tree yang terbentuk belum tentu menghasilkan akurasi ang terbaik. Dan dengan mengganti parameter bisa juga membuat kedalaman tree yang terbentuk lebih sedikit.