Data Mining I: Basic Methods and Techniques
Final Laboratory Assignment:

1. (10pt)Choose the area of your preference and create a dataset. For example: actresses/actors, food, movies, sports, music bands, or anything you would like. Create a data file in .arff format (please attach dataset with your submission) containing at least 50 instances, each described by at least 4 attributes, the last attribute containing your preference (or class attribute), e.g.

```
@relation food
@attribute calories numeric
@attribute taste {sweet, sour, bitter, salty}
@attribute course {appetizer, main, dessert, drink}
@attribute vegetarian {yes, no}
@attribute like_it {yes, no}
@data
100, sweet, dessert, yes, yes%icecream
80, bitter, drink, yes, yes%beer
```

In your own words please describe the dataset. Use data mining to explore and create models to explain the dataset.

The data set that was chosen consist of 340 samples of 40 different plant species. The data set has 16 attributes which describe shape and texture of the plant's leaves. The nominal class are the scientific names of the plant species and was added to the original data set for this exercise. There is also a unique number associated with each leaf in the data set. Plants can have complex or simple leaves, this data set only consist of plants with simple leaves. The original data set was prepared by Silva and et al in 2014. [1] The original data came in a CSV format which Weka could not read so the data set we are not using an exact copy their of the data set for this analysis.

The shape features are Eccentricity, Aspect Ratio, Elongation, Solidity Stochastic Convexity, Isoperimetric Factor, Maximal Indentation Depth, and Lobedness.

The texture features are Average Intensity, Average Contrast, Smoothness, Third moment, Uniformity and Entropy.

An explanation and mathematical definition of the features and photographs of the leaves can be found at the below link:
https://www.researchgate.net/publication/289982979_CLASSIFICATION_OF_LEAF_TYPE_USING_ARTIFICIAL_NEURAL_NETWORKS

Data Repository:

UCI Leaf Data Set:https://archive.ics.uci.edu/ml/datasets/leaf

[1] Evaluation of Features for Leaf Discrimination", Pedro F. B. Silva, Andre R.S. Marcal, Rubim M. Almeida da Silva (2013), Springer Lecture Notes in Computer Science, Vol. 7950, 197-204

Create and compare at least 3 algorithms on your data set (ex. decision trees, a classification or an association rule learner, naive Bayes, etc.) For each algorithm evaluate the model and discuss your findings. What was the performance, is the model relevant, which algorithm can explain your personal liking the best, and observe the generated rules and if they tell you anything interesting? If the model is not good, discuss why and some techniques on how you might improve.

We used the the Weka Experimenter to compare OneR to J48 and the Naive Bayes models. Naive Bayes performed the best based on the following evaluation methods:

```
Tester:     weka.experiment.PairedCorrectedTTester
Analyzing:  Kappa_statistic
Dataset                 (1) rules.OneR '- | (2) trees.J48  (3) bayes.Naive
------------------------------------------------------------------------------
leaves-weka.filters.unsup(100)   0.01(0.02) |   0.61(0.09) v   0.73(0.07) v
------------------------------------------------------------------------------
                              (v/ /*) |       (1/0/0)       (1/0/0)

Analyzing:  Percent_correct
Dataset                 (1) rules.OneR '- | (2) trees.J48 ' (3) bayes.Naive
------------------------------------------------------------------------------
leaves-weka.filters.unsup(100)   6.12(1.00) |   62.12(8.56) v   73.53(6.53) v
------------------------------------------------------------------------------
                              (v/ /*) |       (1/0/0)       (1/0/0)

Analyzing:  Percent_incorrect
Dataset                 (1) rules.OneR '-B | (2) trees.J48 ' (3) bayes.Naive
------------------------------------------------------------------------------
leaves-weka.filters.unsup(100)   93.88(1.00) |   37.88(8.56) *   26.47(6.53) *


Analyzing:  Mean_absolute_error
Dataset                 (1) rules.OneR '- | (2) trees.J48  (3) bayes.Naiv
------------------------------------------------------------------------------
```

leaves-weka.filters.unsup(100) 0.06(0.00) | 0.03(0.01) * 0.02(0.00) *

(v/ /*) | (0/0/1) (0/0/1)

Analyzing: Relative_absolute_error
Dataset (1) rules.OneR '-B | (2) trees.J48 ' (3) bayes.Naive
---
leaves-weka.filters.unsup(100) 97.15(1.03) | 41.29(8.35) * 27.69(6.43) *

(v/ /*) | (0/1/0) (0/1/0) (0/1/0)

The Naive Bayes model improved when removing some attributes. Perhaps if we removed the attributes that are not truly independent we would see better results. It is unlikely all the leaf data attributes are normally distributed which could impact the Naive Bayes model negatively. More data would most likely improve the accuracy of the model. Naive Bayes is attractive because it is simple it provides probabilistic information around the attributes in the data sets

The J48 Decision Tree model was chosen for the visualization of the tree and the purity ranking of the attributes. The root node was " Isoperimetric Factor" so we would not consider eliminate this attribute when testing other models against this data set. I am not sure how to improve the accuracy of the J48 model.

We choose the OneR model as baseline model and we expected better results but performed poorly on every measurement, with a typical rule looking like this:

Aspect Ratio:
- < 1.1122 -> Acer palmatum
- < 1.1902 -> Tilia tomentosa
- < 1.24635 -> Erodium sp
- < 1.29285 -> Betula pubescens
- < 1.4025 -> Corylus avellana
- < 1.5302 -> Hydrangea sp.
- < 1.64005 -> Arisarum vulgare
- < 1.8159 -> Ilex aquifolium
- < 1.953 -> Buxus sempervirens
- < 2.203 -> Acca sellowiana
- < 2.4575 -> Castanea sativa
- < 2.67535 -> Magnolia grandiflora
- < 3.58185 -> Magnolia soulangeana
- < 8.0665 -> Nerium oleander
- < 9.8502 -> Podocarpus sp.
- < 10.8245 -> Pseudosasa japonica
- >= 10.8245 -> Podocarpus sp.

(115/340 instances correct)

We also experimented with the Expectation Maximization model to investigate how the clusters looked we fond the accuracy was not too high, please see a typical result below:

Scheme: weka.clusterers.EM -I 100 -N 15 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100
Relation: leaves-weka.filters.unsupervised.attribute.Remove-R15
Instances: 340
Attributes: 15
Eccentricity
Aspect Ratio
Elongation
Solidity
Stochastic Convexity
Isoperimetric Factor
Maximal Indentation Depth
Lobedness
Average Intensity
Average Contrast
Smoothness
Third moment
Uniformity
Entropy
Ignored:
Name

Test mode: Classes to clusters evaluation on training data

Clustered Instances

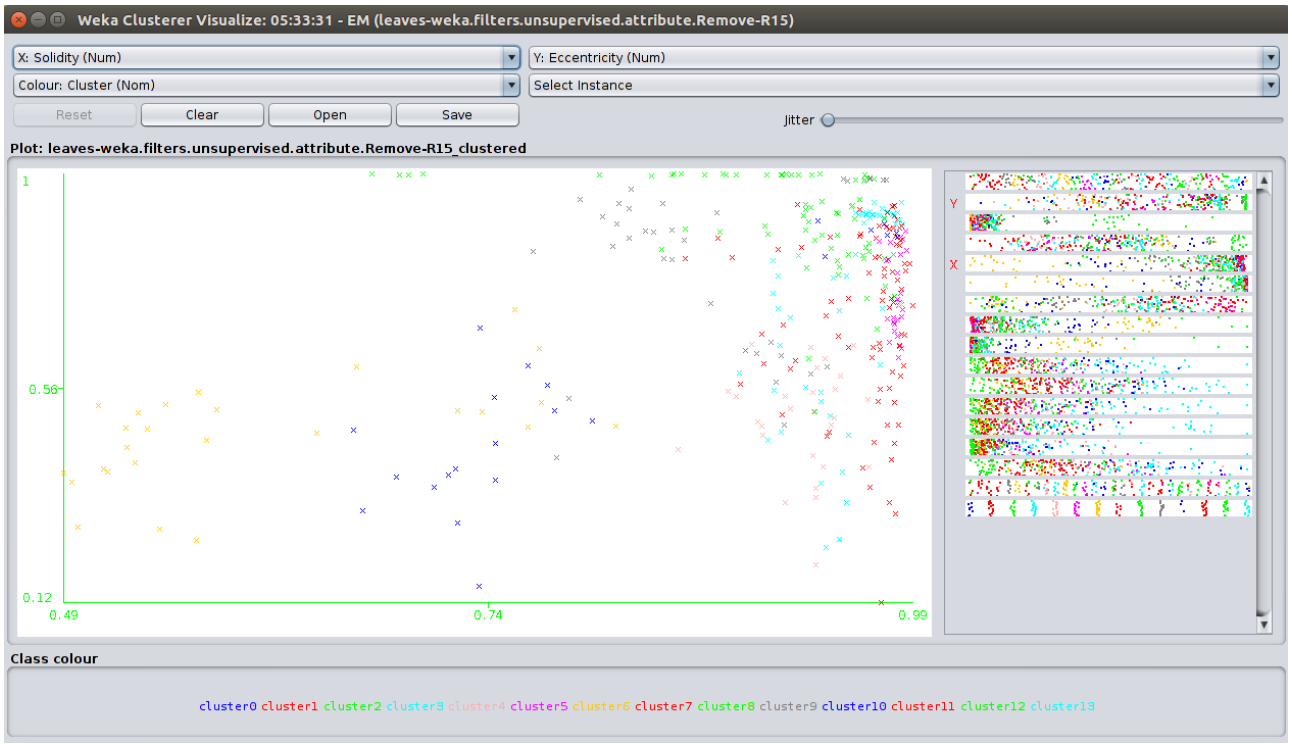
0	25 (7%)
1	18 (5%)
2	28 (8%)
3	18 (5%)
4	30 (9%)
5	38 (11%)
6	17 (5%)
7	12 (4%)
8	27 (8%)
9	26 (8%)
10	17 (5%)
11	37 (11%)
12	20 (6%)
13	27 (8%)

Log likelihood: 32.59309

Cluster 0 <-- Populus nigra
Cluster 1 <-- Quercus robur
Cluster 2 <-- Celtis sp.
Cluster 3 <-- Geranium sp.
Cluster 4 <-- Acer palmatum
Cluster 5 <-- Corylus avellana
Cluster 6 <-- Tilia tomentosa
Cluster 7 <-- Erodium sp
Cluster 8 <-- Pseudosasa japonica
Cluster 9 <-- Castanea sativa
Cluster 10 <-- Buxus sempervirens
Cluster 11 <-- Bougainvillea sp.
Cluster 12 <-- Betula pubescens

Cluster 13 <-- Salix atrocineraIncorrectly clustered instances : 201.0 59.1176 %

We we were interested in how a Neural Network would perform against this data having read in the literature that 15 neuron, 1 input layer, a hidden layer with 23 neurons, and one output layer using the Levenber-Marquart method and obtained a 92% successful classification rate.



We experimented with the "weka.classifiers.functions.MultilayerPerceptron"

Weka's Multi Layer Perception was able correctly classify 80% of our instances:

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 23
Relation: leaves-weka.filters.unsupervised.attribute.Remove-R15
Instances: 340
Attributes: 15
Test mode: 10-fold cross-validation

Correctly Classified Instances	272	80	%
Incorrectly Classified Instances	68	20	%

2. Use the following learning schemes to compare the training set vs. 10-fold stratified cross-validation scores of the labor data in labor_neg_nominal.arff:
   - k-nearest neighbors (IBk) with decision trees (j48.J48)
   - k-nearest neighbors (IBk) with decision trees j48.J48 with option -M 3, so that each leaf has at least 3 instances.

   A) **What does the training set evaluation score tell you?**

The performance test for lazy.IBk and J48 shows that J48 model is significantly worst than the lazy.IBk method with 95% confidence interval for M set to 2

Analysing: Percent_correct
Dataset (1) lazy.IBk '-K 1 | (2) trees.J48 '
---
leaves-weka.filters.unsup(100) 72.97(7.57) | 62.12(8.56) *

J48 with M set 3 performance is not as good when M is 2. The TP rate was also worst for th J48 tree but this test indicate the difference was not significant. Weighted average true positive rate metric also indicate the J48 tee model was significantly worse the lazy.Ibk model. The different evaluation criteria taken as whole tells us that lazy.IBk does better on this data set than J48 model.

B) **What does the cross-validation score evaluate?**

When we examine the data via the Weka Experiment Environment produces, we can see that 10 experiments were run for ever model and a variety evaluation criteria Weka were calculated and averaged over the data set. Weka then gives us final evaluation for our comparisons against each model (or for how one model) did against the data set. The data is partitioned into n equally sized folds and the folds are used for validation against the remaining n – 1 folds training data and this procedure was performed n times.

C) **What did you learn from the models about the data?**

We came to understand from the J48 tree structure the root node was “pension” the next pure node was “wage increase first year”. And that we may need to be careful when thinking about what the “unknown” edges are really telling us.

D) **Which one of these models would you say is the best? Why?**

The J48 trees model has the advantage of being able to visualize the data tree and it is nice to know the purity ranking of the attributes so the J48 model good but on this data set the experimental data shows lazy.IBk is the best.

3. Use the following learning schemes to analyze the Titanic data (in titanic.arff).

C4.5	- weka.classifiers.j48.J48
Association rules	-weka.associations.apriori
Decision List	- weka. Classifiers.PART

A) **What is the most important descriptor (attribute) in titanic.arff, and how can you tell?**

What is the most important descriptor (attribute) in titanic.arff, and how can you tell?

Sex is the most important attribute. The PART model generated 7 rules and sex is found in six of the rules:

PART decision list
---
sex = male AND class = 2nd AND age = adult: no (168.0/14.0)
sex = male AND class = crew: no (862.0/192.0)
sex = male AND class = 3rd: no (510.0/88.0)
sex = female AND class = 3rd: no (196.0/90.0)
sex = female: yes (274.0/20.0)
age = adult: no (175.0/57.0)
: yes (16.0)

The J48 model's tree generated a tree where the root node was the attribute indicating the information gain from this attribute was the greatest and this attribute was sex:

J48 unpruned tree
---
sex = male
| class = 1st
| | age = adult: no (175.0/57.0)
| | age = child: yes (5.0)
| class = 2nd
| | age = adult: no (168.0/14.0)
| | age = child: yes (11.0)
| class = 3rd: no (510.0/88.0)
| class = crew: no (862.0/192.0)
sex = female
| class = 1st: yes (145.0/4.0)
| class = 2nd: yes (106.0/13.0)
| class = 3rd: no (196.0/90.0)
| class = crew: yes (23.0/3.0)
Number of Leaves : 10
Size of the tree : 15

The 7 out 10 rules for the "weka.associations.Apriori" model contained sex so we feel this also shows it importance.

Best rules found:

1. class=crew 885 ==> age=adult 885 conf:(1)
2. class=crew sex=male 862 ==> age=adult 862 conf:(1)
3. sex=male survived=no 1364 ==> age=adult 1329 conf:(0.97)
4. class=crew 885 ==> sex=male 862 conf:(0.97)
5. class=crew age=adult 885 ==> sex=male 862 conf:(0.97)
6. class=crew 885 ==> age=adult sex=male 862 conf:(0.97)
7. survived=no 1490 ==> age=adult 1438 conf:(0.97)
8. sex=male 1731 ==> age=adult 1667 conf:(0.96)
9. age=adult survived=no 1438 ==> sex=male 1329 conf:(0.92)
10. survived=no 1490 ==> sex=male 1364 conf:(0.92)

B) **How well were these methods able to learn the patterns in the dataset? Quantify your answer?**

I am not sure if the "weka.associations.Apriori" model was
appropriate for this model. The rules don't refer to sex = female until we generated 20 rules. Perhaps I am not sure I understand this patterns in this model my bias is leaning toward sex = female to be very crucial in weather a person survives the disaster. Rule 10 implies if you did not survive you were most likely male, which I would agree with.

The J48 model incorrectly classified 461 instances for a correct classified rate of 79% which is not outstanding but pretty good for
the amount of data the model had to work with. The of diagonal entries gives you a numerical an visual indication that this model
did fair job predicting who would survive. This is also supported by the host of evaluation methods calculated by the Weka platform:

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1740	79.055 %
Incorrectly Classified Instances	461	20.945 %
Kappa statistic	0.4334	
Mean absolute error	0.31	
Root mean squared error	0.3945	
Relative absolute error	70.8637 %	
Root relative squared error	84.3537 %	
Total Number of Instances	2201	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.38	0.013	0.931	0.38	0.539	0.75	yes
0.987	0.62	0.769	0.987	0.864	0.75	no

Weighted Avg. 0.791 0.424 0.821 0.791 0.759 0.75

=== Confusion Matrix ===

```
   a    b   <-- classified as
 270  441 |    a = yes
  20 1470 |    b = no
```

The PART model had similar mediocre performance as the J48 model.
Correctly classifying 1740 instances out of 2201 instances for "True Positive" percentage of 79.055 %

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1740	79.055 %
Incorrectly Classified Instances	461	20.945 %
Kappa statistic	0.4334	
Mean absolute error	0.3106	
Root mean squared error	0.3947	
Relative absolute error	70.9957 %	
Root relative squared error	84.3999 %	
Total Number of Instances	2201	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.38	0.013	0.931	0.38	0.539	0.749	yes
	0.987	0.62	0.769	0.987	0.864	0.749	no
Weighted Avg.	0.791	0.424	0.821	0.791	0.759	0.749	

=== Confusion Matrix ===

```
   a    b   <-- classified as
 270  441 |    a = yes
  20 1470 |    b = no
```

C) **Compare the training set and 10-fold cross-validations scores of the methods.**

The training set did not significantly do worst or improve over the 10
10-fold cross-validations score.

D) **Would you trust these models? Did they really learn what was important to survive the Titanic disaster?**

If one had to be a passenger on the Titanic it would most advantageous to be a young female in first class and suboptimal to be an older male in the 3rd class if one wanted to survive this disaster.

We may not be able to trust the model to exactly say who would survive but the model captures the overall trend of who tended to survive and who did not, I trust both models to help understand who in general would tend to survive this disaster.

E) **Which one would you trust more, even if just very slightly? Why?**

I did not trust the Associations Apriori model as much as the other two models.
I trust the PART as it uses J48 to build partial trees as part of its rule generating algorithm. In this case, the models(PART and J48) are statistically the same against this data set:

```
 Tester:     weka.experiment.PairedCorrectedTTester
Analyzing:  Percent_correct
Datasets:   1
Result sets: 2
Confidence: 0.05 (two tailed)

Dataset                  (1) rules.PA | (2) trees
----------------------------------------------------
relation                (100)   79.00 |   78.55
----------------------------------------------------
                          (v/ /*) |   (0/1/0)
Key:
(1) rules.PART '-M 2
(2) trees.J48 '-C 0.25 -M 2
```

4. Choose one of the following three files: soybean.arff, autoprice.arff, hungarian, zoo.arff or zoo2_x.arff and use any two sachems of your choice to build and compare the models.

Evaluate and discuss the models. What was learned by the models (be specific to the dataset)? Which one of the models would you keep? Why?

We will be working with the "Zoo" data for the following experiments. We examined the data to see there was any missing data in the columns and if there was data with very large values or numerics data with larger variation. The data was mostly populated with "true" and "false" entries and there was no missing data. Sometimes we run the Naive Bayes model to see the how the individual column's data are distributed but we did not do this in this case because we mostly have nominal data with "true" and "false" entries. We used the Weka Experimenter to compare the data from zoo.arff and zoo2_x.arff against the J48 Tree and Rule PART. We also to a quick peek at Support Vector Machine and the Multilevel Perceptron. Our goal is to determine which would model was sadistically significantly better or worst to Naive Bayes.

The lazy.IBk model beat out the Naive Bayes, J48, and PART models but not significantly.
The Naive Bayes model did slightly better on zoo2_x.arff model which has does not have the extra attribute "animal".

As expected the Multilevel Perceptron with 5 hidden layers
performed the best but not significantly better than the other models with a significance of 0.05.

We will use the lazy.IBk and J48 Tree models in the Weka Explorer. The J48 model will allow us to visualize the tree and examine the purity rankings to find the "best" attributes.

The Weka Experiment summary is below:

```
Tester:     weka.experiment.PairedCorrectedTTester
Analyzing:  Percent_correct
Datasets:   2
Resultsets: 6
Confidence: 0.05 (two tailed)


Dataset                 (1) bayes.NaiveBay | (2) lazy.IBk '- (3) trees.J48 ' (4) rules.PART  (5) functions.M (6) functions.S
-------------------------------------------------------------------------------------------------------------------------------
zoo                           (100)  93.98(7.14) |  96.15(5.58)    92.61(7.33)    93.41(7.28)    96.23(5.44)    96.24(5.04)
zoo_filtered-weka.filters(100)  94.37(6.79) |  96.15(5.58)    92.61(7.33)    93.41(7.28)    95.75(5.32)    96.24(5.04)
-------------------------------------------------------------------------------------------------------------------------------
Key:
(1) bayes.NaiveBayes
(2) lazy.IBk
(3) trees.J48
(4) rules.PART
(5) functions.MultilayerPerceptron
(6) functions.SMO
```

Our experiment with lazy.IBk correctly classified 97 instances and failed on 4. With K = 3:

```
weka.classifiers.lazy.IBk
K-nearest neighbors classifier. Can select the appropriate value of K based on cross-validation.

=== Run information ===
```

```
Scheme:weka.classifiers.lazy.IBk -K 3 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A
\"weka.core.EuclideanDistance -R first-last\""
Relation:     zoo_filtered-weka.filters.unsupervised.instance.NonSparseToSparse
Instances:    101
Test mode:10-fold cross-validation
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 3 inverse-distance-weighted nearest neighbour(s) for classification
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          97               96.0396 %
Incorrectly Classified Instances         4                3.9604 %
Kappa statistic                          0.9477
Mean absolute error                      0.0125
Root mean squared error                  0.0953
Relative absolute error                  5.7069 %
Root relative squared error             28.8766 %
Total Number of Instances              101

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area

Weighted Avg.    0.96       0.005       0.962       0.96       0.96             0.999

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = 1
  0 20  0  0  0  0  0 |  b = 2
  0  1  3  1  0  0  0 |  c = 3
  0  0  0 13  0  0  0 |  d = 4
  0  0  1  0  3  0  0 |  e = 5
  0  0  0  0  0  8  0 |  f = 6
  0  0  1  0  0  0  9 |  g = 7
```

Our experiment with the J48 Tree model correctly classified 93 instances and incorrectly classified 8 instances. We also learned the "feather" and "milk" were are two "purest" attributes:

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     zoo_filtered-weka.filters.unsupervised.instance.NonSparseToSparse
Instances:    101
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

feathers = false
|   milk = false
|   |   backbone = false
|   |   |   airborne = false
|   |   |   |   predator = false
|   |   |   |   |   legs = 0: 7 (2.0)
|   |   |   |   |   legs = 1: 6 (0.0)
|   |   |   |   |   legs = 2: 6 (0.0)
|   |   |   |   |   legs = 3: 6 (0.0)
|   |   |   |   |   legs = 4: 6 (0.0)
|   |   |   |   |   legs = 5: 6 (0.0)
|   |   |   |   |   legs = 6: 6 (2.0)
```

```
|  |  |  |  |  legs = 7: 6 (0.0)
|  |  |  |  |  legs = 8: 6 (0.0)
|  |  |  |  |  legs = 9: 6 (0.0)
|  |  |  |  predator = true: 7 (8.0)
|  |  |  airborne = true: 6 (6.0)
|  |  backbone = true
|  |  |  fins = false
|  |  |  |  tail = false: 5 (3.0)
|  |  |  |  tail = true: 3 (6.0/1.0)
|  |  |  fins = true: 4 (13.0)
|  milk = true: 1 (41.0)
feathers = true: 2 (20.0)

Number of Leaves  :     17
Size of the tree :     25

=== Stratified cross-validation ===
=== Summary ===


Correctly Classified Instances          93               92.0792 %
Incorrectly Classified Instances         8                7.9208 %
Kappa statistic                          0.8955
Mean absolute error                      0.0225
Root mean squared error                  0.1375
Relative absolute error                 10.2478 %
Root relative squared error             41.6673 %
Total Number of Instances              101

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area

Weighted Avg.    0.921     0.008      0.922      0.921      0.92        0.976

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = 1
  0 20  0  0  0  0  0 |  b = 2
  0  0  3  1  0  1  0 |  c = 3
  0  0  0 13  0  0  0 |  d = 4
  0  0  1  0  3  0  0 |  e = 5
  0  0  0  0  0  5  3 |  f = 6
  0  0  0  0  0  2  8 |  g = 7
```

The lazy.IBk is the best model because it correctly classified the highest percentage of the instance but we appreciate the insight we gain from seeing the tree constructed by the J48 model.