**Decision Tree:**

For this project, decision tree builds classification model in the form of a tree structure. The final result is a tree with decision nodes and leaf nodes. The advantage of using decision tree is to produce a tree that people can understand. The aim is to get the smallest tree and top down tree induction methods use some kind of heuristic and the most popular heuristic to produce pure nodes is an information theory based heuristic. The idea of information theory is to use entropy to measure information in bits. The formula is:

**entropy(P1,P2,…,Pn) = -P1logP1-P2logP2…-PnlogPn**

**Information gain (amount of information gained by knowing the value of the attribute) = (Entropy of distribution before the split) – (entropy of distribution after the split)**

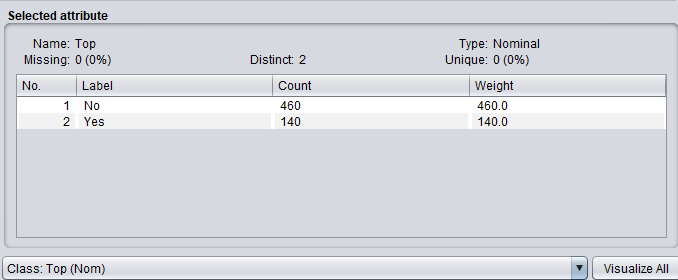
**Classifier:**

In this project, we use J48 tree classifier. In the WEKA data mining tool, J48 is an open source Java implementation of the C4.5 algorithm. The basic steps in the algorithm are: [1]

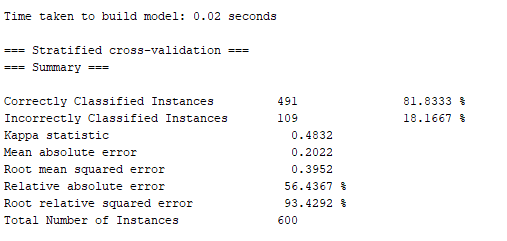
1. In case the instances belong to the same class the tree represents a leaf so the leaf is returned by labeling with the same class.
2. The potential information is calculated for every attribute, given by a test on the attribute. Then the gain in information is calculated that would result from a test on the attribute.
3. Then the best attribute is found on the basis of the present selection criterion and that attribute selected for branching.

**Implementation and analysis:**

1. First combine the top and non-top songs data into one csv file. Add one more column named “Top” to show whether the song is a top song. We have 460 non-top songs and 140 top songs.

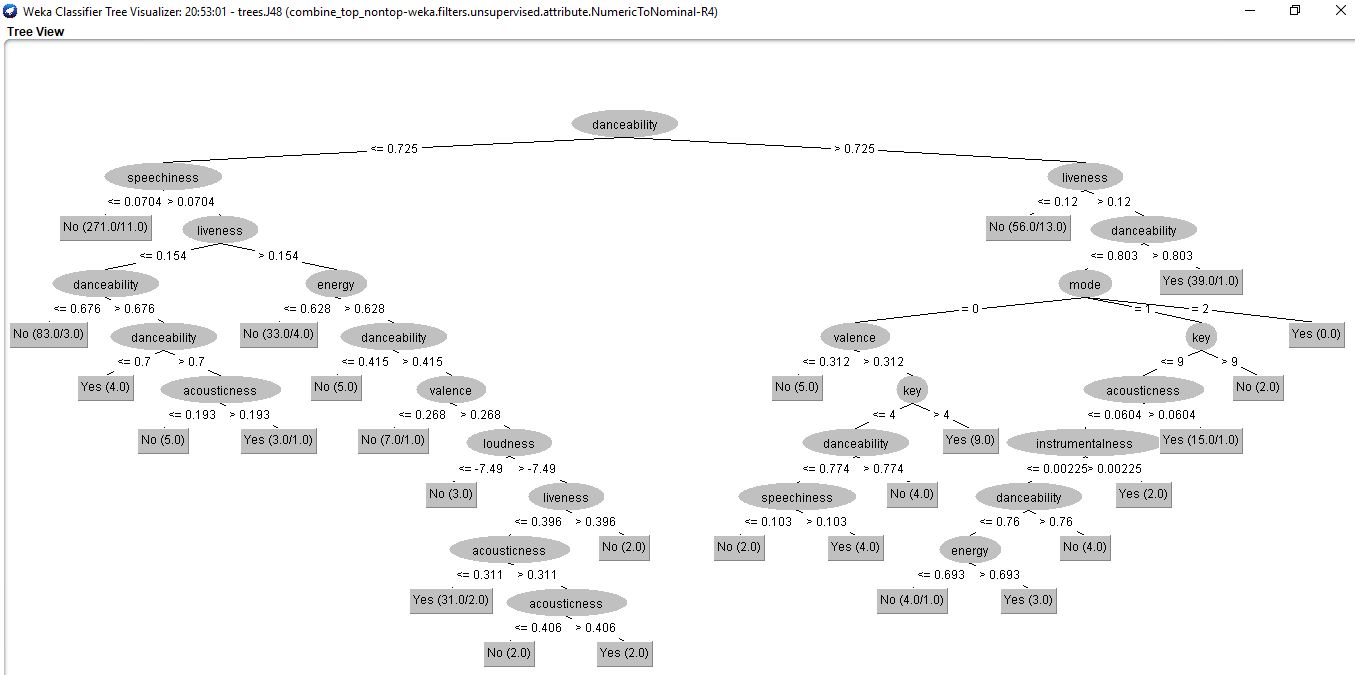


1. Under the classifier, choose J48, use the data to train and select 10 folds cross-validation to test. Use the default pruning and other default settings, here is the result we can get:



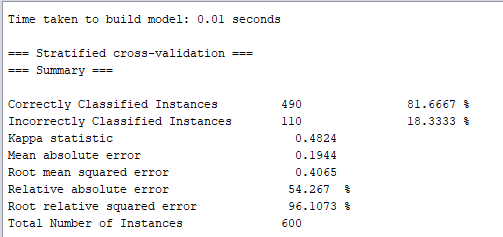
We get an 81.8% accuracy in 0.02 seconds. Compare to the simplest method “always choose the most” which provides 460/600=76.7% accuracy, J48 has great improvement.

In addition, we can easily visualize how the decision tree looks like as shown below:

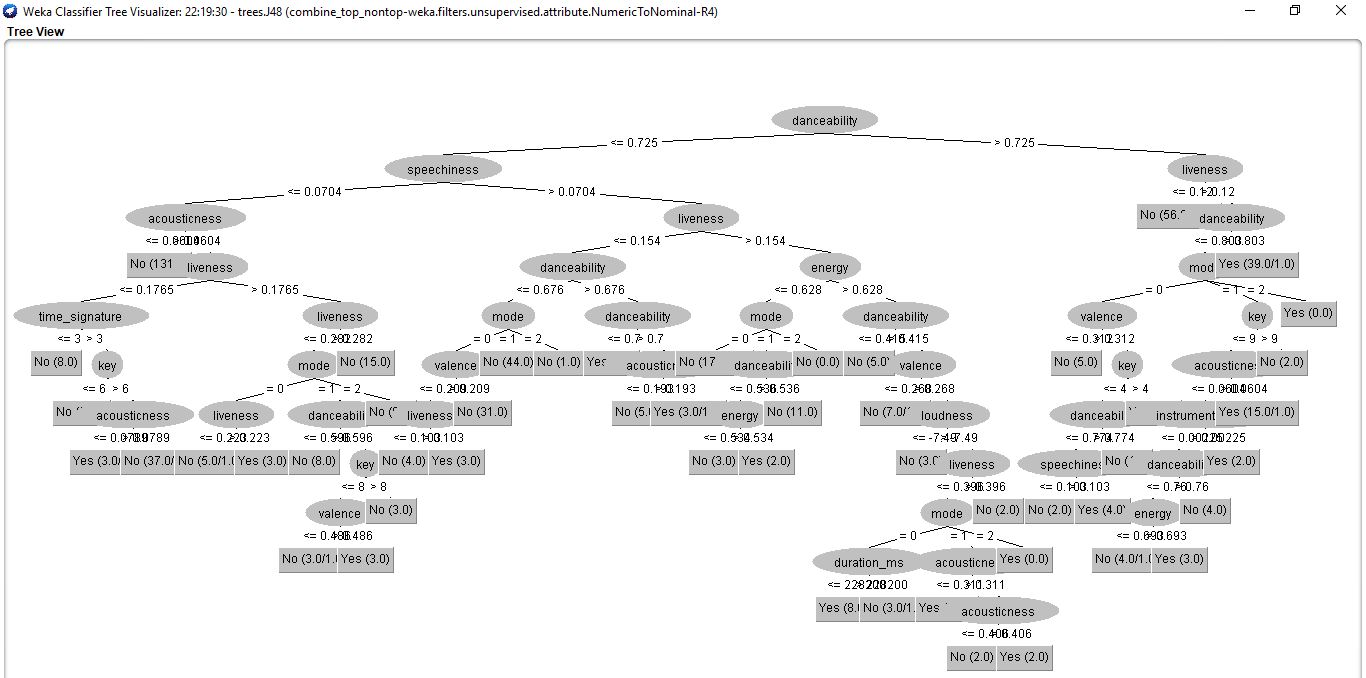


If there are two numbers in one bracket, for example (271.0/11.0), the former number 271.0 means the correctly predicted number of instances, the latter number 11.0 means the wrongly predicted number of instances. The tree has 27 leaves and the number of the tree is 52.

1. If we don’t use prune, here is the result:



We can get an 81.7% accuracy which does not have much difference compare to pruned model. However, the tree is much more complicated as shown below:



The tree has 50 leaves and the size of the tree is 94.

**Conclusion:**

**Confusion Matrix and Mesurement:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predict Class | | |
| Actual Class |  | Class = No | Class = Yes |
| Class = No | 409 | 51 |
| Class = Yes | 58 | 82 |
| Accuracy | 81.83% | | |
| Precision | 0.876 (Class = No); 0.617 (Class = Yes) | | |
| Recall | 0.889 (Class = No); 0.586 (Class = Yes) | | |
| Cost | 409\*0+51\*50+58\*50+82\*0= 5450 | | |

Pruning allows us to avoid over-fitting. In this case, simplifying a decision tree gives us a slightly better result. Overall the J48 classifier gives an 81.8% accuracy.