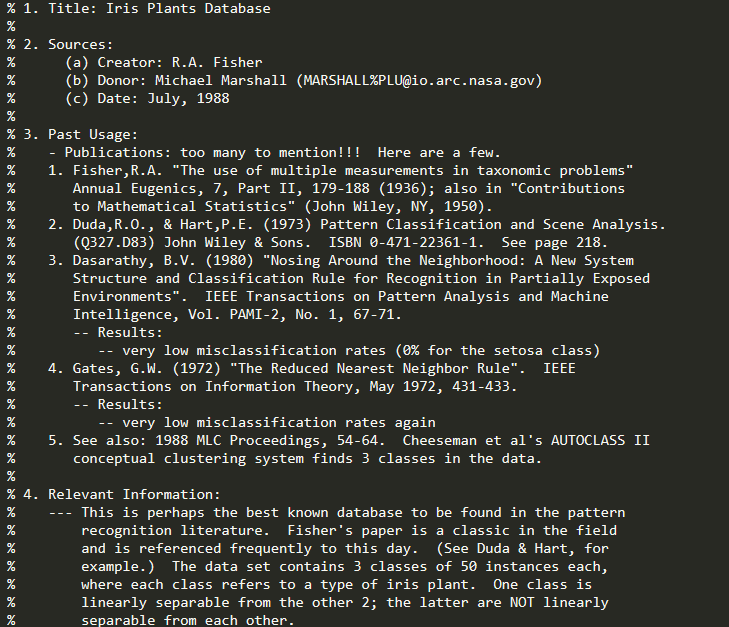
Lab 6 – Classification Dimitrije Prosevski

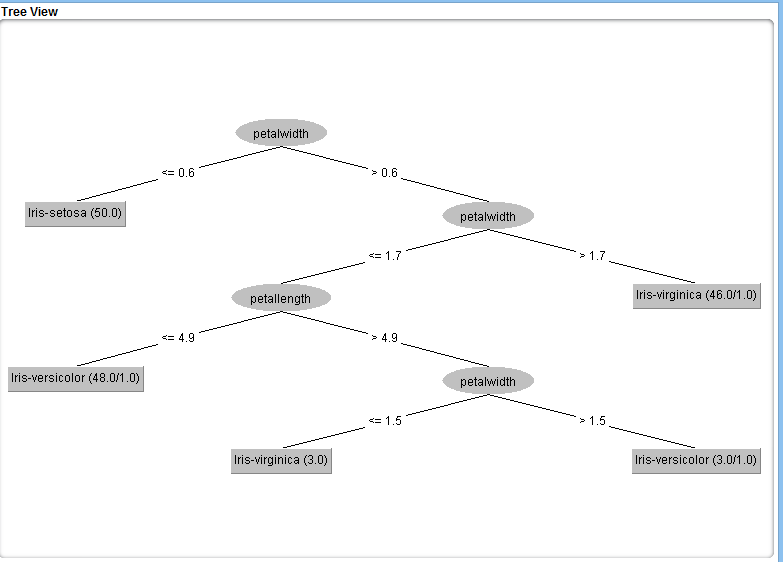
1. Data is about iris data set, one of the most famous datasets for pattern recognition between three type of flowers.

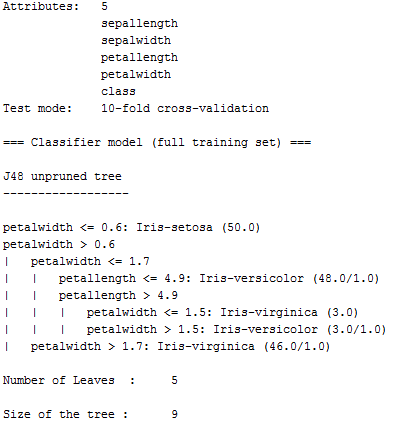


Cross validation = 10

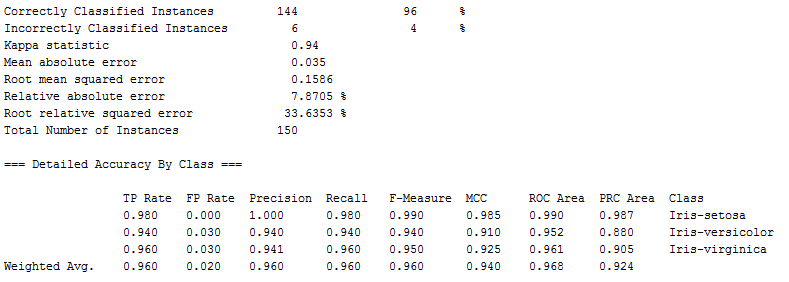
unpruned = true

minNumObj -> minimum leaf size = 2, by default

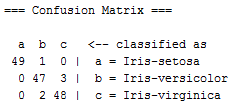




1. There are 6 incorrect instances, which is 4% of the data.

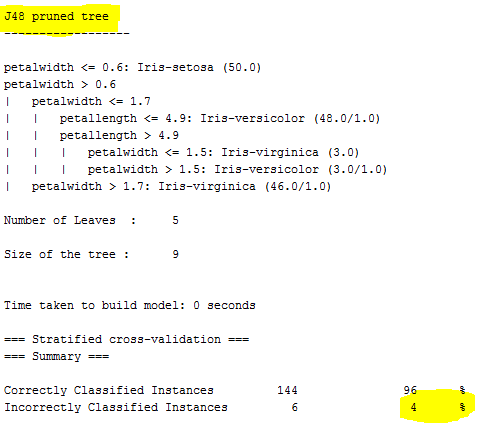


1. Confusion matrix shows the confusion of the classifier when it tries to make predictions. It presents the difference of predicted and actual data. If accuracy was to be 100% the class distribution should be 33.3% for each of the class. Since my example has 6 instances wrong, we can see from the confusion matrix where were those mistakes made and how many instances of the class got classified as the other class.

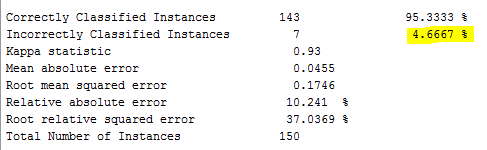


Ideally, a(1,1), b(2,2), c(3,3) should be 50, and other slots of the matrix should be 0.

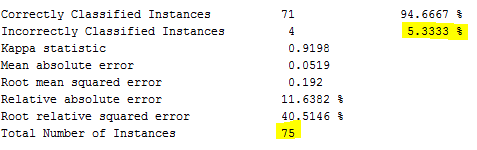
1. On the first run, pruned tree and the statistics of the data were the same as unpruned version.



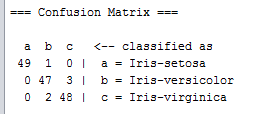
On the second run, I noticed that tree’s error changes for the first time when the minimum instances of the leaf are set to 5. Error percent jumped from 4% to 4.67%. The slope is positive, the larger number of instances we choose for the leaf, the larger the error percentage will be. When min instances = 5:



On the third run, I ran 50% percentage split (instead of 150 instances total, it looked only at 75 of them) and the error rate increased from 4% to 5.33%.



1. I would probably choose the pruned tree just because it had same percent of accuracy as unpruned. Plus, pruning is efficient because it adds an extra step to look at what nodes to remove without changing the performance by much. Confusion matrix stayed the same.



1. I created a new model and predicted the class using “Re-evaluate model on current test set” to get the prediction for 10 arbitrary values and the results were:

