Exercise for MOSES Course
iCog Labs
February 3, 2017: Version 2

Write a python program that uses MOSES to try to classify the three different types of Iris flowers with as much accuracy as possible: Iris-setosa, Iris-versicolor, and Iris-virginica.

Input: The following text file at http://wiki.opencog.org/wikihome/images/1/13/Iris.txt
Output: You need three different outputs for each flower set, written on three different text files both for the training phase and the test phase

**Date and time the evaluation finished
**The flower type for which experiment/evaluation is done
**Training time that moses took (only for the training stage)
**One sample combo program for the flower set
**Accuracy (i.e. how accurate MOSES predicted not only the 1's in the data set but also the 0's in the data set)
**Recall
**Precision
**True positives (recall) ratio (in the format number of true positives out of the total: e.g. 10/17 in the case of the test set), where 17 is the total number of ones for a certain flower type and 10 predictions by MOSES were true positives
**False positives ratio
**True negatives ratio
**False negatives ratio

The exact definitions of these ratios can be found on the MOSES online man page http://wiki.opencog.org/w/MOSES_man_page.

All the independent variables are continuous variables, so you need to expand each variable into bins (binarization procedure). Binning a continuous variable is done as follows. Let us say we want to bin the height of people. If we think that only three bins are needed, then we will have the following three categories,
A) height <=1.5m -- height_bin_1
B) 1.5m<height<=1.8m -- height_bin_2
C) height>1.8m -- height_bin3

So the height variable would be binarized into height_bin_1, height_bin_2, height_bin3. Let's say if we come across with a variable height=1.78, then it would be represented by [0,1,0]. That is bins 1 and 3 would have zero value while bin 2 would have a 1 value. It would be upto you to choose the number of bins and the border floating point number values you use to create the bins. The right number of bins as well as the border floating point values needs experimentation.

You can do experimentation to come up with the correct number of bins used, which increases the classification performance of your program.

Here is how to construct the training/test set. Take ⅓ of each flower set/type by using the randomly generated indexes that you find at the end of this exercise, merge them together and label them as test set and the remaining ⅔ as training set from each flower set. Each flower set has 50 elements, so ⅓ of 50 = 17. And ⅔ of 50 = 33. The test set would then consist of 17*3=51 elements in total, while the training set would have 33*3=99 elements in total.

Here is suggested workflow to do the given exercise.
1) Divide data into training and test sets.
2) Use moses to produce combo programs using the training set. Here you can fine-tune your model by further dividing the training set into *training* and *validation* sets. That is, divide the 99 elements into training and validation sets. Do your training on the training set and test it on the validation set. This means that, during the training stage, you are not allowed to test the predictive power of your model on the test set. Use the test set only once you get the best performance on the training/validation set. That is the model the does not overfit and give poor results on the test set.
3) Using the program eval-table in the moses package, use the resulting combo program(s) on the test set to evaluate the performance of your system using the test set.

Normally, you use one combo program to evaluate a row from a validation/test set. But to improve the performance of your system, you can make use of many combo programs, like 100 to evaluate a single row of the validation/test set; if 20 of the combos predict a zero while 80 of them predict a one, then, that particular row of the validation/test would have 1 for the output variable. This is called prediction by voting. Guys that implement this voting will be more favored in the grading. Note that during voting, the value that I used is just an example (the 20/80 value). It might need some experimentation to get the best value for the voting to work.

You can also experiment with various fitness functions and other moses options to get the best performance results. You can ask Mahlet for more on this or browse the online MOSES documentations.

Here are the indexes of the test sets for each flower type. Index numbering starts from zero and ends at 49 for each flower set.

Test set indexes for Iris-setosa: 20,34,30,28,32,26,0,5,4,15,24,45,19,33,47,46,13

Test set indexes for Iris-versicolor: 10,12,27,9,43,19,6,31,3,46,18,21,24,44,11,14,35

Test set indexes for Iris-virginica: 27,31,4,48,32,13,46,24,45,20,26,44,38,34,49,5,12