Problem 3.1 (performance metrics) **(23 P.)**

Download "output1.txt" from Stud.IP. The file contains the true labels of 100 instances that were classified and the corresponding classifier outputs (scores).

a) Assume we have a threshold of 0.5, i.e., all instances with a score equal to or above 0.5 will be assigned to the positive class. Calculate (no implementation needed!) the performance metrics: accuracy, specificity, F-measure, normalized mutual information and Matthew's correlation coefficient. Is accuracy a good performance measure in this case? Justify your answer. (11 P.)

b) Implement a program that reads the output of a ranking/scoring classifier and plots a ROC graph. Extend your program that it also calculates AUC. Provide an electronic copy of your implementation (use the skeleton provided at Stud.IP *evaluation.py*) and test the program on the above classifier output. (10 P.)

c) Which threshold should be used? Interpret the ROC graph. (2 P.)

Problem 3.2 (precision-recall curves vs ROC) **(7 P.)**

Precision-recall graphs are commonly used in information retrieval for evaluating classification.

a) Plot (manually or with a program) the data of output2.txt (downloaded from Stud.IP) in a graph where precision is on the y-axis and recall on the x-axis. (2 P.)

b) Describe interesting points and regions in the precision-recall graph. (3 P.)

c) Now use dataset A (output1.txt) to plot a precision-recall graph and dataset B (output2.txt) to plot a ROC curve. Compare the ROC graph from problem 3.1 with this one and then the two precision-recall graphs. What is the reason for your observations? (2 P.)

Problem 3.3 (cross-validation) **(20 P.)**

Cross-validation is the most often used method to handle limited data.

a) Implement a function (use skeleton in *evaluation.py*) that generates train/test data pairs according to the cross-validation method. Integrate stratification as well as randomization and repetition in your function. (11 P.)

b) Test your implementation on the **IRIS** data set using the *black_box_classifier* with following parameters:

i) 10-fold cross-validation
ii) 10-fold cross-validation with randomization, 10 repetitions
iii) 10-fold cross-validation with randomization, 10 repetitions, stratification

Calculate and report the accuracy in each case (i) – iii)). Explain which parametrization you would recommend for evaluation. (7 P.)

c) Describe one example where it is *not* appropriate to use randomization/stratification in the cross-validation process. (2 P.)

Problem 3.4 (BONUS) **(5 EP.)**

Imagine a data set with 1000 randomly generated features and binary class labels. A poor student uses a feature selector on that dataset which retained 10 features. He then splits the whole data set into training and test set and evaluates the built classifier on the test set. 70% accuracy. He wonders about that result.

a) Why did the student get such a high accuracy? What was wrong with his procedure?

b) What would be a correct and confident way to build a classifier and determine a performance measure?

On the hand-in date, **23.11.2016**, you must hand-in the following: [1]

a) a text file stating how much time you (all together) used to complete this exercise sheet
b) your solutions / answers / code

for problem 3.1 and 3.2 and 3.3 and 3.4.

[1] upload via StudIP (if there are problems with the upload contact me **beforehand**: krell@uni-bremen.de)