1 Confusion matrix revisited

A three-class classifier is evaluated on a test set of 900 samples which contains all three classes in equal proportions.

- Classes 2 and 3 are always classified correctly
- Class is 1 is confused with class 2 in 50% of the cases, and with class 3 in 20% of the cases.

Write down the confusion matrix!

2 Binary Classification

A classifier which always chooses the positive class is evaluated on a test set in which the proportion of positive to negative samples is 3:1. Compute:
a) fpr, fnr
a) tpr, tnr
c) the classification error

3 Binary classification

A binary classifier which randomly selects its output is evaluated on a test set in which the proportion of positive and negative samples is 1:4. Compute:
a) fpr, fnr
b) tpr, tnr
c) the classification accuracy

4 Binary classification

A binary classifier with a precision of 0.75 and a recall of 0.5 is evaluated on a balanced dataset. Compute:
a) tpr, fnr
b) fpr
b) tnr

5 Binary classification

A binary classifier with a sensitivity of 0.8 and a specificity of 0.5 is evaluated on a balanced dataset. Compute:
a) tpr, tnr

b) fpr, fnr
c) the classification accuracy and error
d) precision using Bayes' theorem!

6 Implementation

Implement the two classifiers from ex. 2 and 3 in Python/numpy on $N = 1000$ dummy data samples. Change the positive rate between 1% to 50% and observe the change classification rates! Hint: create a function *fake_data(pr)* for generating fake data samples X of length 1 and corresponding one-hot-coded target values T, with a positive rate parameter *pr*. Then, create two classifier functions *class2(X)* and *class3(X)* that take a data matrix X and return one-hot-coded classification results. Lastly, create a function *ce(X,T)* that measures the classification error.