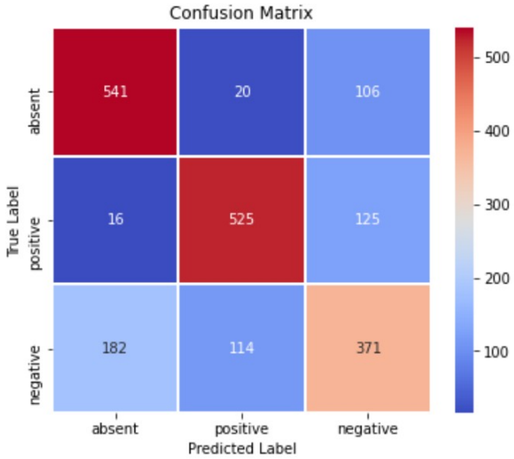
LCPB 20-21 exercise 3 (Convolutional Neural Network, CNN)

```
import seaborn
from sklearn import metrics
```



1. Visualize the "confusion matrix" (look for similar examples in the material available from Mehta's review or by googling)
2. By reducing the signal-to-noise ratio, namely the amplitude of the external signal in the data (A) with respect to the step typical size DX of the jump process, check where the network (defined in the class) starts to fail discriminating the categories.

Then switch to the data in google drive:
x_ts_comp_N10000.csv, y_ts_comp_N10000.csv

3. Try another version where only one convolutional layer is introduced instead of two, and where any number of dense layers may be used, with the global <u>constraint</u> of using a network with <u>at most 600 trainable parameters.</u>

   Is the number of parameters scaling more quickly by adding Dense layers or Conv1D layers?

4. Check if regularization may improve the performances by varying the parameter lambda of the L1 (LASSO) or of the L2 (Ridge) regularization; see lambda in eqs.(43) and (52) in the review. There is also a mixed version (l1_l2) that can be tried.
   1. Is there any intermediate value of lambda where the performances of the network are better?
   2. Is there any improvement in the visualization and understanding of the weights in the filters?

   Note that the regularization we introduced acts on the w's, not on the biases. One can also try the equivalent procedure for biases or for the output of the relu units (see Keras doc.), if there is any reason for suspecting that it may help. In our case, the logic was to let the weights of the filters go to zero if not needed, hence that kind of regularization was selected.

If there is time, draw the confusion matrix for a CNN model trained with the sequence data of the exercise 02.

Competition 20-21

(deadline 21/3)

This is optional, it is for fun and for the glory and it does not replace the exercise

Each group can submit two files in the “Competition 20-21” section in moodle:

1. One file named
   GROUPNAME_parameters.h5
   with one set of parameters obtained by training a CNN with data from files
   x_ts_comp_N10000.csv, y_ts_comp_N10000.csv

2. One jupyter notebook named
   GROUPNAME_competition.ipynb
   reading the trained parameters from the first file and applying the fit of the network to predict the labels for a file named
   x_test_N10000.csv
   and write the predictions y=0,1,2 (one per row) in a final file named
   GROUPNAME_yhat.h5

The constraint remains that of building a model with at most 600 trainable parameters.

The jupyter notebook should be as brief as possible, it should print a model summary to prove that the 600 parameters threshold has not been overtaken, and it may highlight the features that the group introduced for improving its performance (data manipulation, architecture, minimizer, etc.).

To be confident that the jupyter works on prof’s laptop, do not use special packages and test it on different architectures.