Data Analysis and Model Classification
Guidesheet VI: Statistical significance and Final Classifier

Ruslan Aydarkhanov Bastien Orset Julien Rechenmann
Ricardo Chavarriaga José del R. Millán

In this (very short) guidesheet, you will have to evaluate the statistical significance of the overall test error performances. In a second part, based on the last guidesheets you will have to build your final classifier with the optimal hyperparameters that you found.

Statistical significance

Imagine that the mean test error across *outer* folds is 11%. This suggests that the accuracy is pretty good compared to the random level of 50% (2-class problem). Now imagine that the mean test error across *outer* folds is 46%. It is still lower than random level, but the difference might not be statistically significant.

Hands on

- What is the mean test error across *outer* folds? what is the standard deviation of the test error across *outer* folds?
- How do the test error values compare to the random level?
- Do a t-test of the hypothesis that the test error values across *outer* folds come from a distribution with mean 50% (`[h, p] = ttest(error, 0.5)`)
- Is your p-value significant?
- The function `ttest` assumes that data are normally distributed. Draw a histogram of the *outer* fold's test error values. Does it look normal?
- To test if your data are normally distributed, perform a Kolmogorov-Smirnov test (Matlab function: `h = kstest(x)`, where `x` is the standardized test error values (`=(error - mean(error))/std(error)`)
- If kstest rejects the null hypothesis, it means that the distribution is not normally distributed. In this case, you have to use a non-parametric statistical test, such as Wilcoxon signed rank test (Matlab function: `[p, h] = signrank(x, 0.5)`).
- How do `ttest` and `Wilcoxon signed rank` statistics compare?

Final Classifier

In the last guidesheets, you have been introduced to feature selection method *(feature forward selection, fisher score)* and how to optimize your number of features you need to train a classifier. We also introduced to you different type of classifiers *(diaglinear,diagquadratic,linear,quadratic)* as well as dimension reduction algorithm such as *Principal Component Analysis.* Now you have all the tools in hands to make a final "best" classifier (final model). How would you design it? Which hyperparameters would you use?

Include this last guidesheet in the report and submit it on Moodle before **November 19th, 11:59pm.**
Good luck for the mini-project!