1. Install PyMC (see https://www.pymc.io/projects/docs/en/latest/installation.html)
    - Install Anaconda (if possible a fresh install of the latest vesion)
    - `conda create -c conda-forge -n pymc_env "pymc>=5"`
    - `conda activate pymc_env`

2. Run the example presented in the first lecture

3. The next two exercises are based on the following information from L. Devroye, L. Gÿorfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, 1996.

Let (X, *Y)* be a pair of random variables taking their respective values from R^d and {0, 1}. The random pair (X, *Y)* may be described by the pair (μ, η), where μ is the probability measure for *X* and η is the regression of *Y* on *X*:

$$\eta(x) = \mathbf{P}\{Y = 1 | X = x\} = \mathbf{E}\{Y | X = x\}.$$

Thus, $\eta(x)$ is the conditional probability that *Y* is 1 given *X* = *x*.

Any function $g : \mathcal{R}^d \to \{0, 1\}$ defines a *classifier* or a *decision function*. The error probability of g is $L(g) = \mathbf{P}\{g(X) \neq Y\}$. Of particular interest is the Bayes decision function

$$g^*(x) = \begin{cases} 1 & \text{if } \eta(x) > 1/2 \\ 0 & \text{otherwise.} \end{cases}$$

This decision function minimizes the error probability.

Theorem 2.1. *For any decision function* $g : \mathcal{R}^d \to \{0, 1\}$,

$$\mathbf{P}\{g^*(X) \neq Y\} \leq \mathbf{P}\{g(X) \neq Y\},$$

that is, g^* *is the optimal decision.*

2.2 A Simple Example

Let us consider the prediction of a student's performance in a course (pass/fail) when given a number of important factors. First, let $Y = 1$ denote a pass and let $Y = 0$ stand for failure. The sole observation X is the number of hours of study per week. This, in itself, is not a foolproof predictor of a student's performance, because for that we would need more information about the student's quickness of mind, health, and social habits. The regression function $\eta(x) = \mathbf{P}\{Y = 1|X = x\}$ is probably monotonically increasing in x. If it were known to be $\eta(x) = x/(c + x)$, $c > 0$, say, our problem would be solved because the Bayes decision is

$$g^*(x) = \begin{cases} 1 & \text{if } \eta(x) > 1/2 \text{ (i.e., } x > c) \\ 0 & \text{otherwise.} \end{cases}$$

The corresponding Bayes error is

$$L^* = L(g^*) = \mathbf{E}\{\min(\eta(X), 1 - \eta(X))\} = \mathbf{E}\left\{\frac{\min(c, X)}{c + X}\right\}.$$

While we could deduce the Bayes decision from η alone, the same cannot be said for the Bayes error L^*—it requires knowledge of the distribution of X. If $X = c$ with probability one (as in an army school, where all students are forced to study c hours per week), then $L^* = 1/2$. If we have a population that is nicely spread out, say, X is uniform on $[0, 4c]$, then the situation improves:

$$L^* = \frac{1}{4c}\int_0^{4c} \frac{\min(c, x)}{c + x} dx = \frac{1}{4} \log \frac{5e}{4} \approx 0.305785.$$

a) Write the PyMC code that computes the value of L^* (given the above formula) when $X = c$ with probability one and when X is a random variable uniform on [0, 4c]. How the results compare with the theoretical ones presented above?
b) Write the PyMC code that computes the value of L^* based only on the definition of $g^*(x)$ (given above) and the definition of L^* ($L^* = L(g^*) = P(g^*(x) \neq Y)$). How the results compare with the theoretical ones and the results obtained at a)?