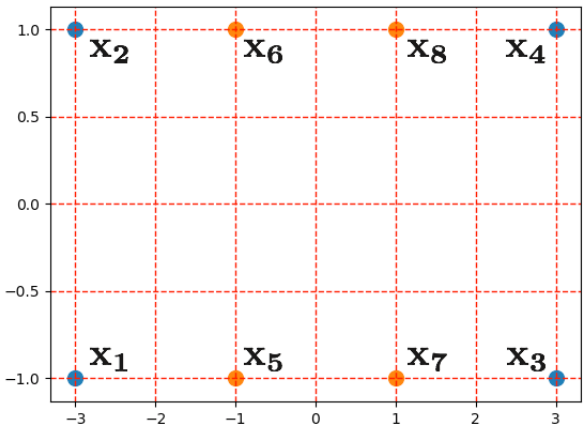
Consider the following 2-class binary problem. We will run Adaboost for two iterations.

Class 0 ($y = -1$): $\mathbf{x_1} = [-3, -1]$, $\mathbf{x_2} = [-3, 1]$, $\mathbf{x_3} = [3, -1]$, $\mathbf{x_4} = [3, 1]$
Class 1 ($y = 1$): $\mathbf{x_5} = [-1, -1]$, $\mathbf{x_6} = [-1, 1]$, $\mathbf{x_7} = [1, -1]$, $\mathbf{x_8} = [1, 1]$



(a) Assuming that the first classifier is $h_1(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} < -2 \\ +1 & \text{if } \mathbf{x} > -2 \end{cases}$, compute the classification error ϵ_1 and new weights $w_2(n)$ for every sample $n = 1, \ldots, 8$ that result from the first iteration ($t = 1$) of Adaboost.

(b) Based on the error that you computed in **(a)**, what would be a reasonable classification boundary $h_2(\mathbf{x})$ for the second iteration of Adaboost?

(c) Compute the classification error ϵ_2 and new weights $w_3(n)$ for every sample $n = 1, \ldots, 8$ that result from the second iteration ($t = 2$) of Adaboost.

(d) What would be the final decision rule based on the first two iterations of Adaboost and the computed variables h_1, h_2, β_1, and β_2? Would this result in all samples being correctly classified? What would you do if not?