Deadline: Th. Dec. 20, 10:00 am Drop your printed or legible handwritten submissions into the boxes at Samelsonplatz, or upload them as `.pdf` or `.ipynb` files onto the LearnWeb.

Exercise 1 (Decision Trees - 10 points)**.**

1. (5) In the lecture you have seen the following image of a partition

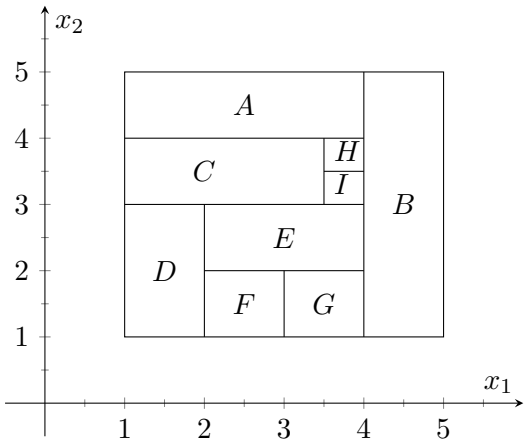


Figure 1: Partition

   Construct a decision tree which models this partition.

2. (5) Given the training data

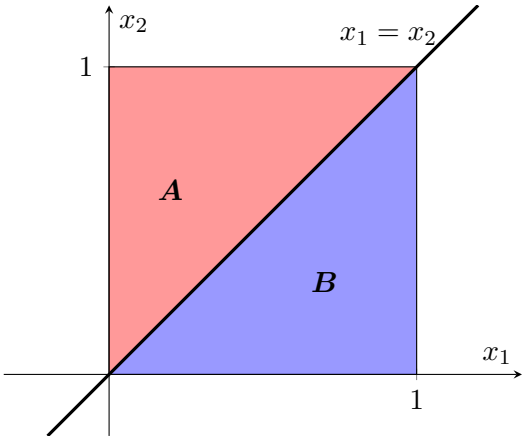
Weather	**Temp.**	**Humidity**	**Wind**	**PlayTennis**
sunny	hot	high	weak	no
sunny	hot	high	strong	no
sunny	mild	normal	strong	yes
sunny	mild	high	weak	no
sunny	cool	normal	weak	yes
cloudy	hot	high	weak	yes
cloudy	hot	normal	weak	yes
cloudy	mild	high	strong	yes
cloudy	cool	normal	strong	yes
rainy	mild	high	weak	no
rainy	mild	normal	weak	yes
rainy	mild	high	strong	no
rainy	cool	normal	strong	no

   Create a binary decision tree of depth 1 which predicts "PlayTennis" by using the greedy search algorithm with the information gain selection criterion!

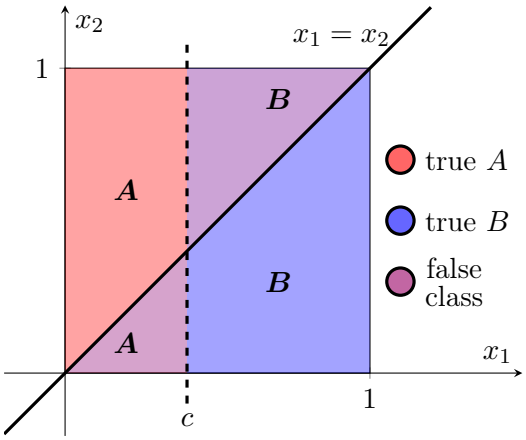
Exercise 2 (Decision tree learning - 10+2 points)**.** In the lecture, it was discussed that certain multivariate splits[1] like $x_1 \leq x_2$ cannot be represented by a conjugation of univariate splits.[2] Assume we have data, uniformly distributed in $[0,1] \times [0,1]$, such that (x_1, x_2) belongs to class A if and only if $x_1 \leq x_2$ and else to class B. That is, the ground truth is:

$$y(x) = \begin{cases} A: & x_1 \leq x_2 \\ B: & \text{else} \end{cases} \tag{1}$$

In this exercise we want to see what happens if one applies the greedy decision tree learning algorithm to this problem.



(a) Ground truth



(b) Splitting at $x_1 \leq c$

1. (2) For which $c \in (0,1)$ does the univariate, binary decision rule

$$\hat{y}(x) = \begin{cases} A: & x_1 \leq c \\ B: & \text{else} \end{cases} \tag{2}$$

   give the best classification rate?[3]

2. (6) Use the greedy learning algorithm to compute the decision tree up to depth 3. Draw the decision tree and a picture of the learned partition.

3. (2) How does the decision tree try to learn the decision boundary $x_1 = x_2$? How would one need to change the algorithm, such that the error rate goes to zero as the tree gets deeper?

4. (2*) Provide a sharp upper bound for the maximum possible classification rate.

[1] Split depending on multiple variables.
[2] Splits depending on only a single variable.
[3] Here, the classification rate is simply the size of the area which gets correctly classified.

Note: In part (2), the greedy search should be used to find the best univariate split among

$$\{x_1 \leq c,\ x_2 \leq c,\ x_1 \geq c,\ x_2 \geq c\} \tag{3}$$

for each node (with classification rate as the selection criterion). If for no $c \in (0, 1)$ any of these rules yield any improvement, the node becomes a leaf.