Introduction to Machine Learning | **Exercise sheet 8**
https://slds-lmu.github.io/i2ml/ | **CART**

Exercise 1: Splitting criteria

Given are the data set

x	1.0	2.0	7.0	10.0	20.0
y	1.0	1.0	0.5	10.0	11.0

and the same with log-transformed feature x:

$\log x$	0.0	0.7	1.9	2.3	3.0
y	1.0	1.0	0.5	10.0	11.0

a) Compute the first split point the CART algorithm would find for each data set (with pen and paper or in `R`, resp. `Python`).

b) State the optimal constant predictor for a node $\mathcal{N}$ when minimizing the empirical risk under $L2$ loss and explain why this is equivalent to minimizing "variance impurity".

Exercise 2: CART hyperparameters

In this exercise, we will have a look at two of the most important CART hyperparameters, i.e., design choices exogenous to training. Both `minsplit` and `maxdepth` influence the number of input space partitions the CART will perform.

a) How do you expect the number of splits to affect the model fit and generalization performance?

b) Using `mlr3`, fit a regression tree learner (`regr.rpart`) to the `bike_sharing` task for

- `maxdepth` $\in \{2, 4, 8\}$ with `minsplit` $= 2$
- `minsplit` $\in \{5, 1000, 10000\}$ with `maxdepth` $= 20$

What do you observe?

c) Which of the two options should we use to control the tree appearance?

Exercise 3: Impurity reduction [only for lecture group A]

We will now build some intuition for the Brier score / Gini impurity as a splitting criterion by showing that it is equal to the expected MCE of the resulting node.

The fractions of the classes $k = 1, \dots, g$ in node $\mathcal{N}$ of a decision tree are $\pi_1^{(\mathcal{N})}, \dots, \pi_g^{(\mathcal{N})}$, where

$$\pi_k^{(\mathcal{N})} = \frac{1}{|\mathcal{N}|} \sum_{(x^{(i)}, y^{(i)}) \in \mathcal{N}} [y^{(i)} = k].$$

For an expression that holds in expectation over arbitrary data, we need to introduce stochasticity. Assume we replace the (deterministic) classification rule in node $\mathcal{N}$

$$\hat{k} \mid \mathcal{N} = \arg\max_k \pi_k^{(\mathcal{N})}$$

by a randomizing rule

$$\hat{k} \sim \text{Cat}\left(\pi_1^{(\mathcal{N})}, \ldots, \pi_g^{(\mathcal{N})}\right),$$

in which we draw the classes from the categorical distribution of their estimated probabilities (i.e., class k is predicted with probability $\pi_k^{(\mathcal{N})}$).

a) Explain the difference between the deterministic and the randomized classification rule.

b) Using the randomized rule, compute the expected MCE in node $\mathcal{N}$ that contains n random training samples. What do you notice?