Setup

Compared to the previous exercise sheet, you don't need to install any further packages. We will work with `matplotlib`, `pandas`, and `scikit-learn`.

Task: Classification and Evaluation

The aim of this exercise is to classify data with decision trees and to evaluate the results. To this end, we use the `iris` dataset from `sklearn.datasets` again. The type of flower, i.e., the species, is our target variable.

If you want, you can also try one or several of the other classification models and evaluation measures introduced in the lecture. `scikit-learn` contains implementations of many of them, and the workflow is similar to the one we apply here.

a) **[Training]** Create an object of class `DecisionTreeClassifier` from the package `sklearn.tree` and train it. For the beginning, you can leave the hyperparameters of the tree at their defaults and use the full dataset for training.
What are disadvantages of such an approach?

b) **[Inspection]** Create a visual representation of your decision tree with `plot_tree()` from `sklearn.tree`. Also, have a look at the properties you can retrieve from the tree object.
Does the structure (e.g., the splits) of the tree make sense, given the plots we created and the statistical tests we conducted for the first exercise sheet? Which features were the most important ones during training?

c) **[Evaluation]** Apply k-fold cross-validation to evaluate your model properly. You may use `StratifiedKFold` from `sklean.model_selection` or implement it manually. For each train-test split, fit a model, make predictions, and store train accuracy as well as test accuracy. You may compute accuracy with the method `score()` of the tree object, `accuracy_score()` from `sklearn.metrics`, or manually.
How do you interpret the results? Why did we stratify the split?

d) **[Baseline]** Determine the accuracy of simply predicting the most frequent class. You may use `DummyClassifier` from `sklearn.dummy` or compute it manually.
How does knowing this accuracy influence our evaluation of prediction models?

e) **[Hyperparameter Tuning]** Learn decision trees of different complexity by passing adequate hyperparameter values in the creation of `DecisionTreeClassifier` objects. You may, for example, vary the maximum depth of the tree.
How does varying this hyperparameter affect the resulting train and test accuracy? Which value would you recommend for the hyperparameter?