General hints for code submissions To make it easier for us and others to understand your solutions please follow these guidelines:

- If available use the template file to create your solution.
- Please add comments so we can understand your solution.
- Please make sure to load all required packages at the beginning of your code.
- Only use relative file paths for `source()`, `load()`, etc.
- Each exercise directory contains a `skeleton` folder where preliminary `R` files are located.
- Use these `R` files as a basis for creating your solution which should be contained in a main `R` file named as the content of the exercise, e.g., `evaluation.R`.
- You can (and sometimes have to) reuse code from previous exercises.
- The points indicate the difficulty of the task.
- If not stated otherwise, we will use exclusively R 4.0 or greater.

Your next task is to get familiar with algorithm data and how to model an algorithms behavior on toy data as well as scenarios from the Algorithm Selection library (ASlib, `www.aslib.net`).

For this exercise assignment, we provide two (simplified) scenarios from ASlib: `SAT11-INDU` and `SAT11-RAND`. Each scenario consists of three files:

- `algorithm_runs.arff`: performance of each algorithm on each instance
- `feature_values.arff`: instance features of each instance
- `cv.arff`: cross-validation splits

We recommend to use the R package `farff` to read these files.
Both scenarios are runtime scenarios with a runtime cutoff (κ) of 5 000 seconds. To simplify your tasks, we will ignore costs induced by using instance features. Please note that some instance features can be missing. We provided you with code that replaces missing values with the mean feature value.

1. **Single Best and Virtual Best Performance** [4 points]
   Given an ASlib scenario, your task is to read the `algorithm_runs` files and to compute the PAR10[1] performance of the Single Best algorithm (SB) and the Virtual Best (VB) performance.
2. **Models for Individual Algorithms** [5 points]
   The lecture taught you about how individual regression models can be used for algorithm selection. Fit a regression model per algorithm to the data. Use these models to predict the expected performances and select which algorithm should solve an instance, given its features. As regression model you can use anything available in `mlr3learners` (You are not required to adjust the hyperparameters of the chosen regressor).
3. **Hybrid Models** [5 points]
   Further you have learned about hybrid models. Implement pairwise classification models to vote on which algorithm should be chosen to solve an instance. Similar to the prior exercise you can use `mlr3` to choose any classifier you want to use.

[1] PAR10 is the penalized average runtime where a timeout is counted as $10 \cdot \kappa$.