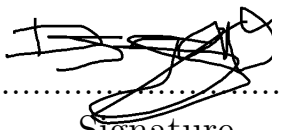
Dr. Janek Thomas
Lennart Schneider
Department of Statistics
LMU Munich

Term Project 2
Automated Machine Learning
Winter Term 2021/22
April 10, 2022

- Check whether the project sheet is complete. It should consist of 3 pages, including the cover sheet.
- Your solution has to be finished before April 19, 23:59 CET (Central European Time).
- Your solution has to be a single private github repository containing either R or Python code.
- Before the deadline, you have to add `ja-thomas` as a collaborator.
- We will fork the repository and use the last commit before the deadline for grading.
- There are no restrictions on the use of non-human assistance for this project. You may use both non-electronic (course material, books) and electronic assistance (searching online).
- This project has to be solved on your own. It is **NOT** allowed to use any kind of human assistance, either by asking others in person or online. It is not allowed to provide assistance to others. In particular, it is not allowed to give other people access to your repository. Noncompliance will result in an automatic failure of the project and a report to the examination board. You are responsible for your actions to reflect compliance with the above examination regulations.
- Upload a scan of this signed first page to your solution repository.

"By signing below, I hereby acknowledge that I have read and understood the instructions. I confirm that the present solution to this project is solely my own work and that if any code from books, papers, the web or other sources have been copied, all references, including those found in electronic media have been acknowledged and linked as a comment in the code."

Daniel	Saggau	12144037
Name	Surname	Matriculation number
WISO Statistics	MA	
Study program	MA/BA/DIPL	Signature

Question 1 .

First you need to prepare some setup to hand in your solution correctly. If you have trouble with the setup please contact us.

a) Create a github account at `https://github.com/` if you do not have one already.

b) Create a **private** repository named `LASTNAME_automl_ws21`, where `LASTNAME` should be your last name.

c) Add the github account `ja-thomas` as a collaborator to the repository[1].

d) Upload a scan (or picture) in sufficient quality of the signed first page of the exam to the repository.

Hint: If you have never worked with github, you might want to have a quick look at `https://docs.github.com/en/get-started/quickstart`.

Question 2 .

The goal is to implement a simple **multi-objective** AutoML tool yourself. You can either implement it in R or Python, other programming languages are not allowed. You don't need to write a complete R package or Python module, but the solution must

a) be executable from any machine (check for absolute paths).

b) have all dependencies specified in a `requirements.txt` file for Python or `DESCRIPTION` file for R.

c) have a clear entry-point, e.g. `Automl()` class or `automl()` function with meaningful arguments (think about what the inputs for an AutoML system should be).

d) be commented, especially arguments and return values of functions and classes need to be commented in an appropriate form.

e) have a consistent style following a style-guide.

Your AutoML tool needs to have the following features:

a) It should optimize 2 objectives simultaneously:
- The generalization performance measured by the misclassification rate.
- The relative number of features in the data used by the pipeline (lower is better!).

b) The search space should combine one feature filter technique with a gradient boosting algorithm. Think which hyperparameters are important to tune and choose meaningful bounds. **Hint:** Try to only use numeric or integer hyperparameters, as mixed-space multi-objective optimization is pretty difficult.

c) Either use multi-objective Bayesian optimization or a multi-objective evolutionary algorithm as an optimizer. **Hint:** Use an existing package and do not implement these from scratch.

[1] See `https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-user-account/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository` for help on how to to this.

d) Be able to provide a budget parameter either in runtime or iterations, i.e., the automl tool should terminate after a given runtime or after a given number of tuning iterations.

e) Be able to compute the current Pareto set and (dominated) hypervolume. **Hint:** You do not need to implement these yourself, the optimization framework you are using should provide functionality for this.

You can use machine learning packages such as `mlr3` or `scikit-learn` as well as generic optimization packages such `bbotk` or `optuna`, but no off the shelf AutoML tools.

Question 3 .

Evaluate your implementation on two datasets: `madeline` and `madelon`.

a) Download the data from `https://openml.org/search?type=data&status=active&id=1485` and `https://openml.org/search?type=data&status=active&id=41144`. You can either download them manually from the website or use a tool like `mlr3oml` or `OpenML-Python`.

b) Set up a script `experiment.R` or `experiment.py` that executes the full experiment described below.

c) Compare your AutoML implementation from the previous exercise to a

- majority vote baseline predictor and
- untuned random forest,

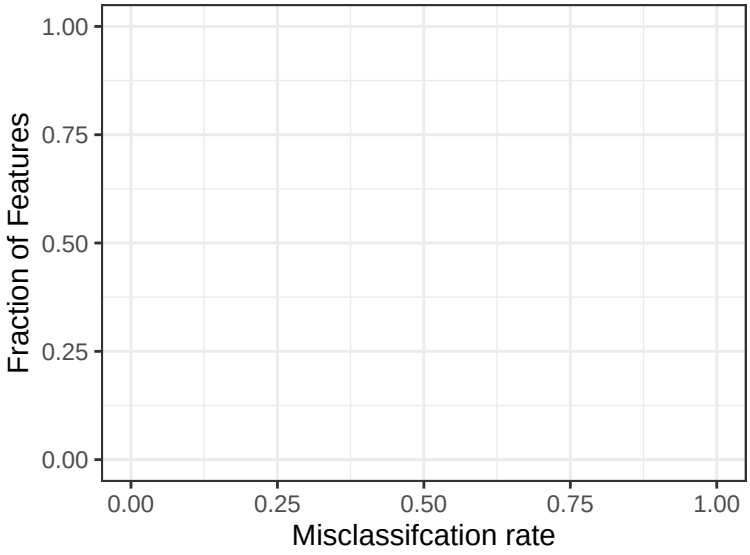
using a 10-fold stratified (w.r.t. target) cross-validation and misclassification rate as performance metric for both datasets. Since your AutoML tool generates a Pareto front of solutions, choose the configuration with the lowest misclassification from the Pareto front for comparison.
Ensure that

- your AutoML tool uses the correct performance metric for optimization and a reasonable compute budget.
- your AutoML tool, random forest and baseline predictor all use the same cross-validation folds.
- the experiment is seeded correctly and produces the exact same results when replicated.

d) What is the relative number of features of the best configuration with lowest misclassification error (averaged over the 10 folds)?

e) Produce a meaningful visualization to compare the performance of your tool, the random forest and baseline predictor.

f) For each of the 10 folds: Visualize the final Pareto front and all evaluated configurations in the objective space.


1.00
0.75
0.50
0.25
0.00
Fraction of Features
0.00
0.25
0.50
0.75
1.00
Misclassifcation rate