The final project can be worked on in **groups of up to 3 students**. The purpose of this project is for you to you get hands-on experience on most topics of the course and to show that you can present and explain the results of your work. To get access to the template please use the following github link:
`https://classroom.github.com/a/HATfd6yP`.
You will have to write a paper of **max. 4 pages** with an optional appendix of **1 page per student in the team**. In the repository you can find a template for the paper. In the appendix the specific contributions of every student need to be specified. The paper needs to be submitted before August 31st 2022 to Openreview:
`https://openreview.net/group?id=automl.cc/AutoML/2022/Student/Conference`.
After submission every student will be assigned a paper to write a review before September 09th 2022 . Guidelines on how to write a review can be found in the repository. On September 15th & 16th 2022 we will organize a virtual conference on `https://app.gather.town/`. For the conference your group should create a poster and will have to present their results. The presentation should be 5 minutes and is followed by a 15 minute QA. After the conference you have until September 23rd 2022 to upload an improved and final version of your paper. Your grade is determined by the paper, poster, review, presentation and quality of code.

Automated Machine Learning for Multi-Label classification

Multi-label classification is type of supervised learning task. For each observation we have m binary labels $y_1^{(i)}, \ldots, y_m^{(i)}$ and a feature vector $x^{(i)}$. A brief introduction to multi-label classification can be found here:
`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.9401`.
Your task is to build an Automl tool for multi-label classification outperforming a simple random forest baseline on a subset of a standard multi-label classification benchmark:

For simplicity we defined a subset that is already uploaded to OpenML and can easily be downloaded:

Name	data_id
enron	40590
genbase	40591
image	40592
langLog	40593
reuters	40594
scene	40595
slashdot	40596
yeast	40597
birds	40588
emotions	40589

You are allowed a **maximum runtime of 1 hour** per dataset. In the end, you should convince us that you outperformed a random forest baseline by your approach. To this end, you could consider the following:

- Classifiers that inherently support multi-label classification.
- Extend other classifiers to `MultiOutputClassifier` and `ClassifierChain`.
- Data preprocessing and feature engineering.
- Ensembling and stacking.
- Efficient optimization, either by BO or EAs.

- Multi-fidelity optimization based on increasing data subsamples.

You are not allowing to (re-)use existing AutoML frameworks with multi-label capabilities. To evaluate your approach please choose the way you evaluate well; you could consider the following:

- Measure and compare against the performance of the random forest;
- Use a reasonable number of folds for crossvalidation, both internally for the AutoML system and externally for evaluation;
- Plot the performance of your AutoML approach over time;

You are allowed to use all scripts and tools you already know from the exercises; however, you are not limited to them. Overall, you should respect the following constraints:

- **Metric:**
  - The final performance has to be measured in terms of F_1-score.
- **Experimental Constraints:**
  - Your code for making design decisions should run no longer than 1 hour per data set (without additional validation) on a single machine.
  - You can use any kind of hardware that is available to you. For example, you could also consider using Google Colab (which repeatedly offers a VM with a GPU for at most 12h for free) or Amazon SageMaker (which offers quite some resources for free if you are a first-time customer). *Don't forget to state in your paper what kind of hardware you used!*

General constraints for code submissions Please adhere to these rules to make our and your life easier! We will deduct points if your solution does not fulfill the following:

- We will use exclusively Python 3.7+.
- We expect Python scripts that conduct the experiments and creates results and visualizations.
- Add comments and docstrings, so we can understand your solution.
- (If applicable) The `README` describes how to install requirements or provides addition information.
- (If applicable) Add required additional packages to `requirements.txt`. Explain in your `README` what this package does, why you use that package and provide a link to it's documentation or GitHub page.
- (If applicable) All prepared unittests have to pass.
- (If applicable) You can (and sometimes have to) reuse code from previous exercises.

Grading:

- Paper: (at most 75):
  - Convincing motivation of the main idea in the introduction: 10 points
  - Sound and complete explanation of the approach: 20 points
  - Solution idea in general: 10 points for approaches from the lecture, further 10 points for ideas beyond the lecture
  - Thorough, insightful, and reproducible experiments: 20 points
  - Language quality (typos, gramma): 5 points
- Code (at most 25):
  - Well documented: 5 points
  - DocString: 5 points
  - Code quality: 5 points
  - Requirements: 5 points
  - Reproducibility[1]: 5 points
- Review (at most 25)
  - Reasonable summary of the paper at hand: 5 points
  - List of strong and weak points of the paper at hand: 10 points
  - Constructive feedback: 10 points
- Poster Presentation: (at most 60)
  - Well structured poster: 10 points
  - Clear message and insights: 15 points
  - Good illustrations, figures and plots: 15 points
  - Comprehensive short pitch (5min): 20 points

[1] `https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf`