Mini-project instructions

Summary

In MLME, part of the examination is based on an *individual* project, which focuses on the development/deployment of a ML-based media technology system. The system is selected by the student and approved by the course coordinators. The project requires to design, implement, document and validate the system with suitable data set. The assessment is performed by the documented project files (e.g., a Jupyter Notebook). An initial version of the mini-project must be demonstrated within the allocated workshop slots (I or II, or on demand) latest by <2023-11-30 Thu>, before the DE hand-in by December 8th.

**** Submission deadline is 08 December 2023 before 12:00 (noon)****

Preparation

Before starting to work on the project, it is absolutely necessary that you review the Lectures 2-7. Selecting a project (or thinking about possible project ideas) without being familiar with ML is s significant waste of time.

After completing the review, you should get familiar with the course learning objectives, mini-project deliverable and the assessment criteria. You must have clear understanding of:

- what you are supposed to do
- how, what and when to deliver
- how you are going to be evaluated

Once you are good with the above, you should start thinking about what is going to be the ML application that you will be working on for your mini project. The ML canvas template from Made With ML by Goku Mohandas (License: CC BY-SA 4.0) should help you elaborate your ideas.

If you wish to **receive feedback** on your mini project idea, hand in your filled ML canvas during the Workshop II (the earlier the better as you may receive more rounds of feedback). You will receive constructive feedback at early stage. Even if you do not wish to receive feedback, your project must be **approved** by the course coordinator.

Project Ideas

If you are are struggling to find a suitable *technical* idea, please check the resources and ideas at the EAAI Mentored Undergraduate Research Challenge:

- 2024: AI for Accessibility in Communication
- 2023: Human-Aware AI in Sound and Music
- 2022: AI-Assisted Game Design
- 2021: Gin Rummy
- 2019: Birds of a Feather

You can also check Kaggle for student challenges and competitons. Moreover, you can explore the vast collection of scikit-learn examples, which is the main classical machine learning package in Python. Most scikit-learn examples are *not* related to media technology, but you can consider to repurpose and expand these in the medialogy domain.

Recommendations

Do not spend too much time thinking about what you wan to work on. Time is scarce and precious. The novelty or the value of your application is not part of any assessment criteria, as well as the absolute performances. You only need to ensure that the application is aligned with the broad field of media technology. You can select an application that is on the technical side, on the creative side, or anywhere in the between. Take this as a chance to work on something you are interested in, or to do a preliminary exploration of a topic you want to further develop (for example, in your master thesis). The rationale of your design, the extent to which it has been implemented, evidences of critical thinking with respect to application of existing ML techniques are at the core of your assessment.

Explore the unknown at your own risk. This introductory course focuses on a variety of common ML techniques, and you will be introduced in implementing these using on scikit-learn. Primarily we will be looking at techniques for classification, regression, clustering and dimensionality reduction. For your project you can use other packages or other techniques not covered in the course, but be aware that you may not receive support or advice on these. If you are new to ML, it is recommended to stay within the boundary of the course material, and if you wish to further explore should stick to techniques available through scikit-learn. Check out the related documentation to see the extensive collection of ML techniques available in scikit-learn.

Due to the short duration of the course it is not recommended to work on a

system requiring deep-learning techniques and framweorks (unless you have a prior experience with it and you are aware of the implications related to the computational complexity). Deep-learning is generally needed for generative ML media applications (e.g., generate media either as a symbolic representation or raw audio).

Stay away from real-time. Do not aim for any real-time functionality of your system (e.g. real-time interactivity, real-time media, video, or audio generation) even if you work on a ML application that is meant to be used in real-time settings. In particular, the development / deployment environments you will be using for your mini-project (e.g, Jupyter notebooks) are not real-time friendly, and the implementation of a real-time system provides additional challenges which are not relevant for this course.

Instead, work on an offline implementation of the system where the data is always read and written (in case you have output data as well) to files. That will still give you full marks. Using input/output data from/to files is a standard practice as this allows you to modify the system and evaluate/compare the output behavior (improved or not) in a consistent and reliable way. If the data feeding your ML system comes from sensors you are sampling in real-time (whether is a microphone or an accelerometer), you should to store these into a files and build a consistent and reliable database (clean, align and equalize the data into the files, avoid outliers and noise).

Reuse, Understand, Explain and Credit. You are not requested to write a large program implementing a ML-based system from scratch. You can use as a starting point what is provided to you in the course, what you find in the documentation, tutorial and examples of the recommended tools and beyond. However, it is necessary that you understand and explain the code as required in the assessment criteria, as well as give credit to the original authors.

Use an online repository that provides you version control. You are recommended to Github (or similar) to backup frequently the software you develop and your data set. This prevents you from loosing your work and allows you to rollback to earlier version in case something went wrong in your development. Moreover, keep a backup copy of the submitted material. If your assessment goes astray, whether your fault or ours, you will be required to reproduce it.

Plan and manage your time wisely. Training your system may take time. This depends on any factors such as the machine you are using, the features you compute, the ML algorithm you select and the size of your data set. You may use this time to work on your report. Moreover, when you implement a system (whether is hardware or software), you never get it working at the first attempt due to bugs. Allocate time for testing and debugging of your system. To minimize the chance of introducing bugs difficult to identify, you should develop your system incrementally, which means execute and briefly verify that everything works every time you add a few lines of code. This may seem time consuming

works every-time you add a few lines of code. This may seem time consuming,

but it will definitely take you less time than writing the large blocks of code, and then trying to understand where the bugs are (yes there may be more than one, and fixing may take days).

It's all about data. Creating or finding a data set to train and test your system can be time consuming. If you create a dataset expect to spend time in finding, organizing, and checking all examples in your dataset. Instead if you start from an existing database, expect to spend time in reading the documentation for understanding what is in the database, how it is organized, and how you can interface or use it with your ML-based system. Whatever choice you make, there is no quick and easy way out. It is very important that the data is consistent, clean, and properly organized. A well designed system will produce poor performances is the dataset is messy. Remember that ML learns from data, so ensure there's something that can be derived from your data! If you find hard to find or collect the data you wish to use, consider to emulate it, e.g. rather than collect some data from a performer playing an instrument, you can make reasonable assumptions and generate the data numerically. **Be sure to include a dataset card in your documentation.**

Write for your audience. Assume that the reader of your report is an expert with a good background in media technology and ML. You can use a technical language without the need of explaining or simplifying all concepts. Use a simple and concise language, go straight to the point, and focus on facts. Do not write a story on how you get there (e.g. everything that you tried but did not work), focus on your final design, rationale, outcomes and reflections. Your mastery in the selected techniques should clearly show through your report.

Handing in

You will hand in a report pdf and code as an appendix (archieved as zip or similar) to the digital exam, by **08 December 2023 before 12:00**

Your report should be an extension to the questions you answered in preperation. It can be exported from a Jupyter Notebook or written from scratch using markdown, LaTeX, Word, or similar. There is no specific template, any structure that supports scientific dissemination with a title, sections, figures and their captions, etc., could be used. There is no page limit, but the text should be no longer than 1000 words (about 2 pages of written text, including references). As the word count is limited, make a good use of illustrations, tables, diagrams and equations to communicate facts and figures.

The code should include everything to reproduce your results. Checkpoints or pre-trained models are great help for reproduction without re-training, and are encouraged to include, especially if you use deep learning. If you have sourced your data, a pointer is enough. **Do not submit a large dataset!**

your data, a pointer is enough. **Do not submit a large dataset.**

Last modified: Monday, 6 November 2023, 1:08 PM

Jump to...

Contact ITS Support Tel: +45 9940 2020 support@its.aau.dk

AAU It services - Aalborg University
www.its.aau.dk

Contact site support