**[TI3150TU] Capstone Applied AI project – Course Design**

# Context

|  |  |  |  |
| --- | --- | --- | --- |
| Quarter 1 | | Quarter 2 | |
| Intro to Python | 5 EC | Intro to Machine Learning | 5 EC |
| Algorithms and Data Structures | 5 EC | Capstone Applied AI project | 10 EC |
| Intro to AI & Responsible AI | 5 EC |

Table 1: Minor programme Engineering with AI (bachelor)

Quarter 2 of the minor programme mainly revolves around the Capstone Applied AI project. However, the course Introduction to Machine Learning is a prerequisite, which will take most of the students’ attention during the first few weeks of Quarter 2. Intro to Machine Learning will completely wrap up with an exam at the beginning of week 5. Afterwards, students will work full-time (+/- 40 hours per week) on the Capstone project. This document briefly describes the organization of the Capstone project: the schedule, the responsibilities of the PhD and TA, the assessment, templates, an example project, etc. More background information on the minor programme can be found on [the AI minor website](https://www.tudelft.nl/ewi/studeren/minoren/engineering-with-ai).

# Summary

* (DAI) research groups write project proposals for the minor students of their major programme.
* Students bid on the projects from their major; we aim for student groups of size 5.
* Groups are supervised by a PhD of the corresponding research group and a TA from EEMCS. The PhD plays the role of the problem owner (client / stakeholder), providing domain knowledge and guidance, and the TA plays the role of the process owner, safeguarding the group process.
* PhDs and TAs both have (separate) weekly meetings of 1 hour with each student group.
* The goal is to align the research of the PhD student with the Capstone projects. Additional meetings may be necessary to introduce the necessary domain knowledge and tools during the first weeks.
* There is much freedom to formulate a project proposal for the PhD / research groups. Mandatory components are: (1) students use or analyse an AI technique (e.g., use algorithm on a dataset, theoretical analysis, build a model, user study), (2) build something that is reproducible (e.g., theorem, scripts, dataset, experiment, prototype, …), and (3) reflect on ethics and impact.
* Students negotiate the requirements and goals with the PhD and write a project plan before the project starts. This is to ensure a mutual understanding of the task at hand, and good match with the students’ level and interest. If it turns out not all requirements can be met, students inform the PhD as soon as possible during the project and negotiate a solution. Risk taking is allowed but if requirements are not met students should reflect on why this was the case and formulate lessons learned in the report (how to foresee this next time, where did risk management fail?).
* Assessment: final presentation and report. Students that fail will have to redo a new project next year. The obligatory midterm presentation in week 2.7 provides early feedback to the students by the PhD and TA. Here we use a signal: green/orange/red, red = fail if no change in behaviour.
* We provide a structure for the project: a schedule, deliverables and grading rubrics. Deliverables and documents can be adjusted to the needs of the PhD, we provide templates as guide.

# Table of contents

[Context 1](#_Toc72267012)

[Summary 1](#_Toc72267013)

[Table of contents 2](#_Toc72267014)

[Description of the course 3](#_Toc72267015)

[Schedule 4](#_Toc72267016)

[Responsibilities 4](#_Toc72267017)

[PhD student – responsibilities 4](#_Toc72267018)

[TA from EEMCS – responsibilities 5](#_Toc72267019)

[Workload TA and PhD 5](#_Toc72267020)

[Student – responsibilities 5](#_Toc72267021)

[Course organizer – responsibilities 5](#_Toc72267022)

[Version control 6](#_Toc72267023)

[Midterm and final presentation 6](#_Toc72267024)

[Learning objectives 6](#_Toc72267025)

[Assessment 7](#_Toc72267026)

[Midterm rubric 7](#_Toc72267027)

[Final rubric 7](#_Toc72267028)

[Overview of documents, basic templates, and deliverables 8](#_Toc72267029)

[Appendix A - Example proposal: Twitter sentiment prediction 9](#_Toc72267030)

[Appendix B - Project suggestions for project proposal 10](#_Toc72267031)

# Description of the course

The project enables students to use their skills of the minor, such as machine learning algorithms, algorithmic thinking, visualizations, etc. in an applied setting of their major. One large part of this project is, together with a mentoring PhD student, to translate a problem from the domain of their major into a form that can be approached by some AI technique(s). Here the students need to extract domain knowledge from the PhD and combine that with their AI knowledge obtained in the minor programme to come to a creative solution.

The students work as a team in collaboration with the PhD student and a TA provided by EEMCS. The TA from EEMCS is there to safeguard the process; he/she meets with the group to discuss progress, documentation, task division, communication, and to a lesser extent technical support. The PhD student provides domain knowledge and provide guidance during the project.

The students, in negotiation with the PhD student, write a project plan with requirements and goals for the project. The requirements should be measurable, and how the requirements are validated is also specified in the project plan. For example: “Improve the sentiment classifier” would be too vague and not measurable, but “Increase the accuracy of the classifier on the Kaggle Twitter dataset” would be measurable and specific.

Requirements are formulated using the Moscow framework (Must have, Should have, Could have, Won’t have). The plan includes a risk analysis to reflect on feasibility. During the project, the students are responsible to perform risk management: they should keep an eye on their progress and whether all requirements will be met. If it becomes clear a requirement may not be met by the students, the students will inform the PhD and TA as soon as possible, and the students will negotiate a solution (adjusted requirements). If essential requirements are not met, a reflection and formulation of lessons learned are expected in the final report to compensate.

Ideally, the project is aligned with the research of the supervising PhD student. However, one should be careful to match the level of difficulty of the project with the level of the students of the minor programme. To that end, the project proposal is checked by the course organizers, and the requirements are negotiated with the students to further adjust the goals to the group level. The PhD meets weekly with the group for an hour; but the PhD should keep in mind that at the beginning of the project more meetings may be necessary to get a group up to speed. If appropriate, the project could have a research question offered by the PhD for investigation (note that letting students formulate a question is out of the scope of the capstone project).

Possible ideas to align the project with the research of the PhD could be to reproduce a scientific work; to prove some theoretical or complexity result; to train, tune, and analyse machine learning models on data provided by the PhD; or to visualize, cluster or otherwise explore data of the PhD. Data should be provided timely by the PhD (in week 2.1) if required. A part of the project may also be devoted to collecting or processing datasets or developing a small piece of software. For more ideas, we have included an example project proposal and a list of suggestions at the end of this document.

Note that we have kept the project description and components broad on purpose: we offer the PhD and research labs much freedom in tuning the project to their needs and interests. As a minimum component, we expect that at least one AI technique is used or analysed by the students in a project, where ‘AI technique’ is often an algorithm that operates on data. Analysis can refer to theoretical or empirical analysis or a user study (the latter applies to explainable AI, for example). Developing new AI algorithms is out of the scope of the project but training or combining models should be feasible. A project cannot *only* involve collecting data or developing software, but this may be a component.

To offer structure to the project, we have developed a schedule, together with the deliverables, templates, deadlines and grading rubrics. There is a midterm presentation where the students get early feedback, from their peers within and outside their own group, and from the TA and PhD student. A final presentation wraps up the project in the last week together with a final report and the produced product (scripts, visualizations, a processed dataset, etc.). An important aspect is reproducibility: there should be sufficient documentation to get everything up and running for the PhD. Templates and components for the deliverables are suggestions and may be adapted by the PhD as seen fit.

# Schedule

|  |  |  |
| --- | --- | --- |
| Week | What | Project agenda |
| 1.8 | nothing scheduled | Students submit project preferences. |
| 2.1 | 1 lecture (2h), 1 project session (4h) | Groups assigned, meet PhD, make agreements, start project plan. |
| 2.2 | 1 lecture (2h), 1 project session (4h) | Meet PhD and discuss requirements (Moscow), discuss project plan. |
| 2.3 | 1 lecture (2h), 1 project session (4h) | Meet PhD (optional), finalize project plan, submit project plan to PhD. |
| 2.4 | nothing scheduled | PhD student provides feedback on plan with TA, approves if sufficient. |
| 2.5 | 3 project days (8h) | Adjust and resubmit plan for approval (if necessary). Draft design document. |
| 2.6 | 5 project days (8h) | Work on design, implementation. |
| 2.7 | 5 project days (8h) | Design, implementation, midterm presentation, peer feedback. |
|  | Christmas leave |  |
| 2.8 | 5 project days (8h) | Implementation, testing, evaluation, documentation. |
| 2.9 | 5 project days (8h) | Testing, evaluation, final report, presentations. |
| 2.10 | 2 project days (8h), presentations | Testing, evaluation, final report, presentations. |

**Lectures and learning activities:**

* Week 2.1: Kick-off lecture, overview of course
* Week 2.2: Collaboration in a team, requirements solicitation, Moscow, reproducibility
* Week 2.3: Ethical analysis in the context of the Capstone project

# Responsibilities

## PhD student – responsibilities

* Week 1.4: submit project proposal to course organizers.
* Week 1.6: receive feedback on project proposal, adjust if necessary.
* Week 1.7: submit projects to [project forum](https://projectforum.tudelft.nl/) for student bidding
* Week 2.1: first student meeting (introduce topic), deadline to provide dataset to students.
* Week 2.2: 2nd student meeting, requirements discussion and negotiation.
* Week 2.3: 3rd student meeting (optional), receive project plan from students.
* Week 2.4: provide feedback on project plan.
  + If insufficient: students need to revise and resubmit until approved.
* Weeks 2.5-2.10: weekly meeting (1h minimum) with students to provide guidance.
* Week 2.7: midterm presentation, provide feedback afterward using rubric / mail / meeting.
* Week 2.10: final presentation and receive final report, provide feedback on presentation.
* Week 2.11: provide feedback on final report, fill in rubric, send to course organizer.

The PhD plays the role of the problem owner. When students have been assigned to the corresponding project of the PhD, he meets with the students 2-3 times during weeks 1-3 before the project starts, to describe the requirements of the project and to provide the necessary domain and background knowledge. These meetings are used as input by the students to write the project plan containing the requirements. If the plan is deemed sufficient, the PhD approves the plan and the project can directly start in week 5. Otherwise, the project plan needs to be revised and resubmitted. During weeks 5–10 the PhD meets weekly with the group to provide further guidance. To provide adequate domain knowledge, the PhD may need to schedule additional meetings. The PhD attends the midterm and final presentation to offer (early) feedback to the student via the rubric per mail and / or verbally in the weekly meeting. The PhD also reads the final report and provides detailed feedback. The rubric and feedback are sent to the course coordinator who will decide the final grade for each student.

## TA from EEMCS – responsibilities

The TA meets weekly with the students. In this meeting the students present their progress. The TA’s responsibility is to safeguard and help the students with the process: planning, communication, risk management and general management of the project. To a lesser extent the TA will also help with more technical challenges such as programming. Each TA is assigned to 4-5 groups and each weekly meeting is scheduled for 60 minutes. Feedback and advise by the TA is delivered in the meeting verbally, but a copy of the feedback is sent per mail as well. The TA also fills out the rubrics for the assessment, but does not have to give detailed feedback on the final report.

## Workload TA and PhD

Estimate: 20-25 hours per group consisting of: 10-15 hours meetings, 2 hours presentations (midterm, final), 8 hours preparing the project template, project data, reading reports and verifying deliverables, providing feedback, filling out rubrics, communicating over email.

## Student – responsibilities

The students attend all weekly meetings, presentations, and other meetings with the TA and PhD student. The students in each group make agreements in the project plan, and all students are expected to stick to these agreements during the project. If conflicts arise, the students will notify the TA as soon as possible. The students prepare an agenda for the weekly meeting with the TA, to make the most of each meeting and to make the meetings run smoothly. Together with the TA the students reflect on the process and feasibility of the project, keeping an eye out if all ‘must haves’ are feasible. If it becomes clear that a ‘must have’ is infeasible for the final product, the students will notify the TA and PhD, and adjusted requirements are set up in negotiation with the PhD. In addition, if requirements are changed, students are asked to reflect on why this was the case and what lessons have been learned and how this could have been foreseen, which will go into the final report.

Overview of schedule:

* Week 1.8: submit project preferences.
* Week 2.3: project plan submission.
* Week 2.4-2.5: revise and resubmit project plan if insufficient.
* Week 2.7: midterm presentation and peer feedback.
* Week 2.10: final presentations, product and final report due.

## Course organizer – responsibilities

The course organizers will help to review the project proposals before they are submitted to the students for bidding. The course organizers will mainly focus on feasibility (is the project not asking too much from students in the time given?) and whether all mandatory components can be met. Further, the course organizers will meet regularly with all TAs to keep an eye out that all groups are on track and no problems occur. If necessary, the course organizers will contact student groups directly or join their meetings or meet with the PhD or TA to assist them. The course organizers in the end will determine the final grade of each student (based on input from TA & PhD). Furthermore, the course organizers are available during the course as contact for everyone involved.

## Version control

We must make a remark regarding version control. The minor program does not have any courses regarding software engineering, version control and the use of e.g., git. Given this, our plan is to not use git at all and even discourage its use, as it may lead to many problems with inexperienced users. To this end, we will also not offer any support by the TAs in the use of git. Students are expected to think about how to manage their files and versions using an approach of their own choosing, such as Surfdrive, Google Colab, or other platforms. The students maintain a design document which keeps tracks of all files and their versions. We will investigate whether Vocareum can be used for effective collaboration during the project.

# Midterm and final presentation

We will make a schedule with block structure of sessions consisting of 1 hour each. In each hour, 3 groups will present (10 min presentation, 10 min Q&A). The students, TA and PhD will stay the whole hour, so they can also ask questions and provide feedback to other student groups as well. Before the midterm presentation, we will ask each group to internally provide each other peer feedback. This peer feedback is discussed with the TA before the Christmas break.

The midterm should cover:

* Motivation and background
* Current product and progress
* Planning and risk assessment

The final presentation should cover:

* Motivation and background
* Final product description
* Justification of the design

# Learning objectives

LO1. Develop an AI solution for a problem in the field of your major taking ethical and societal considerations into account.

LO2. Analyse information from literature and problem owner.

LO3. Use the Moscow framework to describe, refine, adjust and prioritize requirements.

LO4. Justify design decisions with requirements, arguments, literature, knowledge of the problem owner or (experimental) results.

LO5. Manage progress and risk during a project.

LO6. Evaluate an AI solution against a list of requirements.

LO7. Collaborate within a team to develop an AI solution.

LO8. Present an AI solution.

# Assessment

The rubrics are largely based on the existing rubric of the CS BSc software project and still need to be worked out in more detail, but below should give an indication of how the grading will occur. Note that these are not checklists; instead, each of the items in the list will receive a partial score (insufficient, sufficient or excellent). Furthermore, the 4 components will receive a partial grade and are weighted together to determine the final group grade.

For the assessment, the PhD student and TA will fill in the rubric directly after the midterm and final presentation. The course organizers will use these as inputs together with the student deliverables and peer feedback to decide the final scores in the rubric and final grade for all students.

## Midterm rubric

Note: what constitutes the *product* for each group depends on the project proposal; it could refer to scripts, a small piece of software, visualizations, a collected and cleaned dataset, experiments, etc.

|  |  |
| --- | --- |
| **Product** | **Process** |
| * Problem analysis * Formulating requirements * Implementation quality * Use of AI techniques * Verification & validation * Quality of documentation * Reproducible | * Planning * Creativity * Group interaction and communication * Risk management / forecast * Proactive regarding risks and requirements |
|  | **Presentation** |
|  | * Motivation and background explained * Current product and progress * Planning and risk assessment |

For the midterm there is no grade, instead we use a traffic light signal. GREEN = good, ORANGE = can improve, RED = needs direct improvement and change of the process or students will likely receive a failing grade (e.g., not enough time spent, too ambitious, not focussing on the right requirements).

## Final rubric

|  |  |
| --- | --- |
| **Product (35%)** | **Process (25%)** |
| * Problem analysis * Formulating requirements * Implementation quality * Use of AI techniques * Verification & validation * Quality of documentation * Reproducible | * Planning * Creativity * Group interaction and communication * Risk management / forecast * Proactive regarding risks and requirements |
| **Report (25%)** | **Presentation (15%)** |
| * Justification of choices in product * Fair evaluation of own product, limitations * Estimation of (future) ethical, regulatory, societal and technological impact. * Reflection on the process, lessons learned | * Motivation and background explained * Final product described well * Justification of the design * Performance in Q&A |

Group grade, +1 point or -1 point per student for extraordinary / exceptionally poor performance.

The course organizers will use the PhD input most to determine the product grade, while the TA input will be most useful to determine the process grade. Note that must have requirements that are not met can be compensated with lessons learned.

# Overview of documents, basic templates, and deliverables

**Project proposal** (provided by PhD student in the [project forum](https://projectforum.tudelft.nl/) system)

* Background and project aim
* Moscow analysis (Must have, Should have, Could have, Won’t have) – preliminary
* Prerequisites (these could be related to courses of the major)
* Provided data (needs to be provided in week 2.1 at the latest)
* Recommended libraries, tools and code
* References

**Project plan**

Deadline in week 2.3, needs to be approved by PhD student before start of project

* Background and project aim
* Ethical considerations, including future societal, technological and regulatory impact
* Moscow requirements and risk analysis
* Validation approach (how to evaluate whether the final product has reached the goals?)
* Time planning, deadlines, responsibilities, how the process is organized
* Agreements regarding deadlines and communication
* Description of where the code, documentation, and other relevant files can be found, and how different versions are managed (to avoid overwriting files and loosing code).
* Libraries, tools and code that will be used

**Design document**

Ungraded, living document updated throughout the project by all group members

* Components of the project, where to find them, also including different versions
* Reasoning and justification for design choices
* For example: explain why these libraries, toolboxes, frameworks, algorithms, tools, approaches, metrics, visualizations, etc. were used / not used / are under consideration
* May also contain experimental results, comments from PhD, reference to requirements or project plan, to justify choices.

**Final report**

Length: 20 pages max (excluding appendices). In consultation with the PhD, form can be adjusted.

* Background and project aim
* Description of the end product, final components and justifications for the design
* Evaluation of whether the product meets the requirements (validation approach)
* Evaluation on which goals were reached and which weren’t and why, discuss lessons learned
* Reflection on the group process
* Ethical considerations, including future societal, technological and regulatory impact
* Description of where the code, documentation, and other relevant files can be found
* Libraries, tools and code that were used

**Final product**

* 2 ZIP files, 1 for source code, the other for generated / produced results, graphics, data, etc.
* Documentation in a readme provided in the ZIP file, covering the installation steps, and steps needed to reproduce the results, including settings, (links to) datasets, toolboxes, prerequisites
* Final presentation PowerPoint file / PDF

# Appendix A - Example proposal: Twitter sentiment prediction

**Background**

Every day, over 500 million Tweets are posted on the social media platform Twitter. These are short text messages of no more than 280 characters. Many use Twitter to share opinions and emotions.

**Project aim**

The objective is to create an application which estimates the mood or sentiment of a specified person or area based on Twitter data. The user provides a Twitter username or geographical area. The application gathers the most recent Tweets based on that input, and uses a machine learning model to estimate the general mood of those tweets and presents this to the user.

**Moscow analysis**

* **Must** take a Twitter handle or geographical area as input
* **Must** estimate the mood of Tweets (positive/negative/neutral) using a NLP model
* **Must** come up with a suitable evaluation metric for the task
* **Should** have multiple ML models trained on data
* **Should** perform a model comparison to find the best model for the task
* **Should** classify the mood on a scale
* **Should** have an interface (possibly command line interface)
* **Could** have a web interface
* **Could** interact with the Twitter API to collect tweets (time consuming!)
* **Could** use a dataset annotated by students themselves (time consuming!)
* **Could** use a state-of-the-art neural network fine-tuned on their own data
* **Won’t** interact with Twitter in any other way (send tweets, etc.)
* **Won’t** implement a machine learning algorithm from scratch

When submitting the proposal to the course organizers, please do include preliminary requirements (as above). The course organizers will have a look and try and assess whether the requirements are feasible for students of the minor.

Note: this analysis will not be included in the project proposal for the students, as the students need to figure out the requirements themselves by talking and negotiating with the PhD student. Note that the students can deviate from the requirement list from the PhD (it is a negotiation).

**Ethical aspects**

Please already have a thought about potential (future) ethical aspects of the proposed project. Students will need to reflect on this multiple times during the project. This information will be removed from the project proposal so students can analyse this themselves.

One issue regarding the Twitter sentiment classification example is that this application cannot be released in the EU. There are two problems: (1) according to the EU AI act users need to be informed when their mood is analysed through automated means, and (2) that Twitter users their data is used for a different purpose than originally intended. As such, this project would require HREC approval. In case it is in doubt whether such approval is necessary, please inform us as soon as possible as it may take some time to get such approval (+/- 2 months).

**Prerequisites**

None

**Provided data**

Data is collected using the Twitter API or an existing sentiment twitter dataset is used ([example](https://www.kaggle.com/kazanova/sentiment140))

**Recommended libraries, tools and code**

NLTK, Huggingface Transformers, Flask

**References**

Alshammari, Norah Fahad, and Amal Abdullah AlMansour. "State-of-the-art review on Twitter Sentiment Analysis." 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS). IEEE, 2019.

# Appendix B - Project suggestions for project proposal

Below are some suggestions in no particular order:

* Exploratory data analysis, developing visualizations, performing statistical tests
* Building a regression or classification model, finding the most important features
* Use a pre-trained model to make predictions, analyse its performance
* Find the right evaluation metric, determine meaningful baseline performance
* Explain model decisions, analyse failure cases
* Investigate the fairness / bias of a model
* Perform a user study to investigate how explainable / understandable a model is or an explanation technique
* Explaining decisions of a black box model
* Train models, perform model selection, hyper-parameter tuning
* Clustering, dimensionality reduction, development of visualization
* Reproduce an experiment of a published paper (code should be available, implementation from scratch is probably too demanding)
* Theoretical analysis of an algorithm, for example in terms of runtime, memory, etc.
* Analyse an AI technique empirically by applying it or retraining it on new datasets
* Combine multiple machine learning models into a larger system

Out of scope are:

* Coming up with a research question
* A literature study
* A project that only involves data collection or building software

These activities can be part of the project but cannot play the central role:

* Data scraped / collected / cleaned / pre-processed / pipeline built
* Build annotation tool, annotate dataset, curate a dataset, crowdsource a dataset
* Develop interface or GUI to explore, visualize a dataset, train a model, cluster data

Note that these activities could also be more time consuming than expected. Especially for data collection it is useful to have a plan B regarding a dataset – e.g., if the data collection turns out to be too difficult or time consuming, that there is an alternative dataset available.