Team projects & Software projects in KMD Lab for SoSe 2022: "Treatment recommendation engine"

OVERVIEW

Goal of the teamprojects in KMD Lab is to enable master students in solving complex real tasks in teamwork. In doing so, they use and occasionally extend methods they already learned in their studies thus far. They are exposed to the data science challenges of business understanding, of data preparation and of communicating the results to an application expert.

Goal of the software projects in KMD Lab is to enable bachelor students in solving complex real tasks in teamwork. The design is the same as for a teamproject, but the tasks are simpler. It is also possible that bachelor students of high semesters team-up with master students in a mixed project.

The term 'student project' (StudP9 is jointly used hereafter.

Structure of a teamproject & of a software project: A StudP has an objective and must deliver predefined outputs within the timeline of the semester. It is run by a team of typically **three members**. Larger teams are possible, if there are enough tasks.

Prerequisites: Each StudP has different prerequisites. Common to all are

- programming skills
- familiarity with data mining / machine learning software suites, eg scikitlearn
- familiarity with data processing utilities
- familiarity with data mining / machine learning
- basic skills on project management, as disseminated eg in the compulsory part of FIN bachelor degrees or through practical experience

Some StudPs require familiarity with learning on timestamped data.

Modalities of enrolling and completing a StudP:

1. Students apply for a StudP as a team with nominated team leader. The application refers to a specific content and demonstrates (through CV, certificates, transcript of records etc) that the team members, together, possess the skills needed to accomplish the StudP. The application also contains a very rough sketch of the tasks planned by the team for the StudP.
2. The team receives a detailed description and tasks of the StudP through the supervisor. This involves one-two meetings.
3. The team makes a workplan and timeplan, including milestones, and presents them as a StudP "solution proposal".
4. The proposal is evaluated by the StudP supervisor(s). If approved (eventually with modifications), the team registers for the StudP and starts work.
5. The team members meet regularly and report to the supervisor. Reporting and Q&A are agreed individually.
6. The StudP is finalized by the submission of the StudP "Deliverable", which is graded.

Deliverable: It is of the type "Hausarbeit" (aka: Term paper, Homework paper) and consists of

- Description of the StudP goals, overview of the approach, original workplan and timeplan, modifications to the work/timeplan (if any) and justification thereof
- Discussion of the scientific literature used in the StudP solution
- Evaluation criteria towards the StudP goals
- Detailed description of the approach
- Detailed description of the evaluation procedure and its results
- Conclusion about the StudP success in achieving its goals
- List of the tasks performed by each team member

StudP evaluation: Grading of the StudP as a whole is based on following criteria:

- Technical quality of the approach (30%)
- Coverage of the state of the art during the design and the evaluation of the approach (20%)
- Quality and reliability of the evaluation, including statistical testing (30%)
- Quality of the documentation of the solution (10%)
- Quality of the presentation of the results (10%)

subject to the following weights:

- Adherence to the timeline as a weight in (0,1]; key values are "poor adherence"=0.2, "as expected"=1
- Writing quality as a weight in (0,1.2]; key values are "medium quality"=0.5, "as expected": 1, "excellent": 1.2
- Difficulty of the StudP and self-standing work in addressing it, as a weight in {0.5, 1, 1.5} where "normal project"=1, "very difficult project"=1.5 and "project simplified to prevent total failure of the team"=0.5

Each team member gets a separate grade according to their contribution to the above. However, the best teamprojects are those where the team members worked so tightly together that they all receive the same grade.

CONTENT

Area of all topics: *Treatment recommendation engine*

Dataset: Sensitive data! Their use requires signing a Non Disclosure Agreement.
The team acquires:

- the schema, i.e. the list of features F
- the set of treatments
- the target variable in two forms
  - treatment response predictor TRP in (0,100] – smaller numbers are better
  - treatment response TR_class in Y/N
- access to one prediction model per treatment (it is either a classifier or a regressor)

StudP1 – frontend of a treatment recommendation engine serving a treatment response variable TR_class

I. It reads one patient record x at a time

II. It allows the user to specify the ID of the TR_class variable from a dropdown list of variables

III. It issues a query on x to each prediction model and receives for each treatmentID

a) the expected value of TR_class for x

b) the confidence of the prediction model

IV. It lists the treatments for which the TR_class is Y, ordered by confidence.

It supports following functionalities:

V. For each feature f in F, it computes the contribution of f in the treatement prediction for x

VI. It presents the features, ordered by contribution to x

StudP2: as StudP1, but using TRP instead of TR_class in functionalities II, III.a and IV – sorting on response value (ascending) and confidence (descending)

StudP3: as StudP1, but in step I it reads a patient record x that contains only values for a subset F' of the features from F. Then:

- In step V: For each feature f in F-F', it computes the contribution of f in the treatment prediction in general.
- In step VI: It lists the missing features, ordered by contribution

StudP4: Active feature selector for step V of StudP3.