1 Lab Qualification Task

To qualify for the lab course *Praktikum: Analyse großer Datenbestände*, we ask you to work on the Data Mining Cup task from 2019. It is a binary classification task with the goal to predict fraud at self-checkouts in retail. You can get the full task, including data and data description from `https://www.data-mining-cup.com/reviews/dmc-2019/`. Please only download the task and not the solution, as the class labels of the test data must not be used in your prediction pipeline.

1.1 Assessment

There are three criteria we asses your solution with:

1. Code readability/understandability.
2. Reproducibility of results and adhering to the prescribed submission format.
3. Prediction quality penalized by model complexity.

Hence, it is not just important to upload a reasonably good prediction, but also code of decent quality. In terms of the prediction quality, we take into account how complex your prediction pipeline is. Simpler methods are preferable – this is only a qualification task, after all. You should be able to solve the task with a standard notebook or desktop PC, not spending hours on extensive parameter tuning or training deep neural networks.

Note that the DMC organizers did not choose a standard evaluation measure like accuracy, but a custom score as defined in the task. We will use this measure as well. As the classes are very imbalanced, we have also uploaded a presentation with counter-measures against class imbalance, including corresponding R packages. However, it is not necessary to use this methods, or you might just pick a simple one like random undersampling.

1.2 Submission Format

You should hand in three files: a prediction file, a code file and the output of the code. All files have to be uploaded to ILIAS. **The deadline is the 29$^{\text{th}}$ of February 2020.**

The prediction file should have exactly the same format as defined by the DMC organizers in the task description. Name your submission file `FIRST_LASTNAME_prediction.csv` (e.g., `Klemens_Boehm_prediction.csv`).

You should hand in the code as a Rmarkdown[1] file named `FIRST_LASTNAME_code.Rmd` or as Jupyter notebook[2]. You might use R or Python. The Rmarkdown format allows to mix plain text (explaining your approach), code and (after executing/rendering it) results. Also hand in the generated/rendered HTML output file as `FIRST_LASTNAME_output.html`. The code should be well commented and simple to understand. Besides the pipeline used for your submitted prediction file, the notebook should also include code and descriptions of exploration, pre-processing/feature engineering, baseline predictions and (optional, not too many) other approaches you tried. However, make the separation of your final solution and alternatives clear. Also, remove any code which does not work. Your code should read in the DMC files `train.csv` and `test.csv` from a folder `data/` placed at the same location as the code. Furthermore, your code should create your prediction file in the folder `data/` when executed from start to end, without any need for manual intervention (like setting parameters, commenting in/out, non-linear execution order).

[1] `https://rmarkdown.rstudio.com/lesson-1.html`
[2] `https://jupyter.org/`