Machine Learning Operations (MLOps): Machine Learning from R&D to Production

Final Project (Deadline: 7. 1.22)

In groups of three (same as the mid-term exercise groups), please **submit the following:**

1. **The fixed mid-term exercise (with the comments addressed)**
2. **Baseline report (https://github.com/Azure/Azure-TDSP-ProjectTemplate/blob/master/Docs/Model/Baseline/Baseline%20Models.md) for one of the baseline models.**
3. **Model report (https://github.com/Azure/Azure-TDSP-ProjectTemplate/blob/master/Docs/Model/FinalReport.md) including the target metric comparison to the relevant baseline model**
4. **Final Report (https://github.com/Azure/Azure-TDSP-ProjectTemplate/blob/master/Docs/Project/Exit%20Report.md) , including any diffs from the original (mid-term exercise) design and their rationale**
5. **Presentation (content: target metric, architecture, theory behind the step, comparison of both datasets with the baseline models, live/recorder run of automatic step, suggested future steps for further improvement based on the current results). Length: 25 minutes + 15 minutes of Q&A. The deadline for the presentation is different the abovementioned and appears in the groups list document on the course's website.**
6. **The code of the project, including a requirements.txt file, so dependencies can be installed with a single pip install command + instructions file of how to run (one pip command to install, one python command to run)**

The maximal accumulated length of the baseline report, model report and final report should be 10 pages (combined), font size 12 and default margins. The file format must be PDF (to avoid compatibility issues). The fixed mid-term exercise maximal length should be 6 pages, and all the modifications to the original exercise should be highlighted. Please send it to my email by the abovementioned deadline.

The project definition is:
"You are given the bank marketing dataset (which can be downloaded from: https://archive.ics.uci.edu/ml/datasets/bank+marketing) and the German credit risk dataset (which can be downloaded from: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)). You should use the following simple XGBoost models on each as your baseline: https://www.kaggle.com/kevalm/xgboost-implementation-on-bank-marketing-dataset (Version 3) and https://www.kaggle.com/hendraherviawan/predicting-german-credit-default (Version 4; The baseline model for this dataset is 'XGBoost 1b: Unbalance Dataset (ROC_AUC:0.79)', which appears in: https://www.kaggle.com/hendraherviawan/predicting-german-credit-default?scriptVersionId=5889660&cellId=26), respectively.
The goal of the project is to design and implement a basic ML pipeline for each model (you can, but don't have to, use Scikit-Learn pipelines, as shown in: https://nabeelvalley.netlify.app/docs/data-science-with-python/xgboost-and-pipelines/) that

would contain an **automatic step** that would improve the overall performance of the model **in any method that does NOT involving changing the model's type or hyperparameters (although retraining on different data IS allowed) and is NOT dataset-specific** (although it might support only datasets with tabular data only, that is, you don't have to support, e.g., textual or image data).
Example: **"We will add a feature selection phase that will only use the top 90% features with the highest Shapley values, in an attempt to improve the AuROC metric of the baseline model."**
The improvement is to be measured by the metrics you should define and specify in the Metric section of the charter document (e.g., accuracy, precision, recall, a specific fairness metric, etc.).
Like in a real industry ML project, in order to implement the automatic step, you are welcome to use any free or open-source tool we saw during class, you are familiar with, or you find on the internet. Proprietary or private tools are not allowed. A list of suggested tools is specified below, but any other tool can be used instead."

The final project will be graded by these criteria:

1. A working (as in: not crashing and producing an output in less than one hour) code. The code should be installed via a single pip command and run via a single python command.
2. Automatic (as opposed to a manual/human-in-the-loop) improvement step.
3. You will successfully explain your implementation and understand the theory behind it. The theory should be specified in the Learnings part of the final project document.
4. The step is generic and isn't dataset specific. In order to demonstrate that, you should use the exact same automatic step in the pipelines of both the bank marketing dataset and of the German credit risk dataset.
5. The relative performance improvement size (from the baseline model to the model including the automatic step), measured by the metric you determined and specified in the character document.
6. Novelty: Students presenting (and submitting) their projects first would get bonus points.
7. Non-trivial an interesting (theory-wise) implementation.

Examples of possible tools/methods to integrate/implement

1. https://towardsdatascience.com/how-to-find-weaknesses-in-your-machine-learning-models-ae8bd18880a3 - A BIG bonus will be given to the student group who would implement and use this method.
2. https://aif360.mybluemix.net/
3. https://fairlearn.org/
4. https://github.com/ydataai/ydata-synthetic
5. https://ai.googleblog.com/2021/10/baselines-for-uncertainty-and.html - A bonus will be given to the student group who would use this method.
6. https://blogs.oracle.com/ai-and-datascience/post/macest-release - A bonus will be given to the student group who would use this method.

7. https://research.ibm.com/blog/uncertainty-quantification-360 - A bonus will be given to the student group who would use this library.
8. https://towardsdatascience.com/how-can-i-measure-data-quality-9d31acfeb969
9. https://concept-drift.fastforwardlabs.com/ - A bonus will be given to the student group who would implement and use this method.
10. https://medium.com/a3data/data-quality-with-hermione-46233529517b
11. https://towardsdatascience.com/great-expectations-always-know-what-to-expect-from-your-data-51214866c24
12. https://adversarial-robustness-toolbox.org/
13. https://www.tensorflow.org/tfx/guide/tfdv
14. https://www.tensorflow.org/tfx/model_analysis/get_started
15. https://feature-engine.readthedocs.io/en/1.1.x/
16. https://syntheticdata.community/