ARGUSEYES: Screening Native Machine Learning Pipelines

Stefan Grafberger, Shubha Guha, Olivier Sprangers, Sebastian Schelter
AIRLab, University of Amsterdam
[s.grafberger,s.guha,o.sprangers,s.schelter]@uva.nl

Software systems that learn from data are being deployed in increasing numbers in real-world application scenarios. It is a difficult and tedious task to ensure at development time that the end-to-end ML pipelines for such applications adhere to sound experimentation practices, such as the strict isolation of train and test data. Furthermore, there is a dire need to enforce legal and ethical compliance in automated decision-making with ML. For example, in order to determine whether a model works equally well for different groups. For enforcing privacy rights (such as the 'right to be forgotten' [1]), we must identify which models actually consumed the user's data for model training, in order to retrain them without this data. Moreover, model predictions can be corrupted due to undetected data distribution shift, e.g., when the train/test data was incorrectly sampled [2] or changed over time (covariate shift) or when the distribution of the target label changed (label shift) [3]. Data scientists also require support for uncovering *erroneous data*, e.g., to identify samples that are not helpful for the classifier and potentially dirty or mislabeled [4] or to identify subsets of data for which a model does not work well.

Towards automated low-effort screening of ML pipelines. Most of the listed issues are typically addressed manually in an ad-hoc way and require a lot of expertise and extra code. In many cases, there is no system support for detecting particular issues, and typically, data has to be integrated first, as common libraries assume the input to be in a single table. Furthermore, specialised solutions are often incompatible with popular libraries in the ecosystem. This situation is in stark contrast to the software engineering world, with established best practices and infrastructure for testing and continuous integration.

Provenance is all you need. We find that we can automate the detection of many common correctness issues in ML pipelines with access to (i) the materialised artifacts of a pipeline (its input relations, and its outputs, e.g., the feature matrices, labels, and predictions of a classifier) as well as (ii) their why-provenance [5] (e.g., the information which input records were used to compute a particular output). This allows us to design lightweight screening techniques with low invasiveness for natively written ML pipelines, which combine code from different libraries from the rapidly evolving data science ecosystem.

Pipeline screening with ARGUSEYES. Based on these insights, we present our ARGUSEYES prototype, which operates on a natively written ML pipeline in Python, extracts intermediate results and provenance (in the form of provenance polynomials [6]) with MLINSPECT [7], and infers the semantics of their artifacts based on predefined "templates" (e.g., for a classification task). Our prototype enables the automatic detection of common issues w.r.t. best practices in ML, and the computation of metadata such as group fairness metrics, record usage by the model, or data valuation with Shapley values. Our prototype handles classification pipelines natively written in pandas/sklearn and keras, stores their artifacts and run data via mlflow [8], and can be easily hooked into continuous integration workflows.

Current State & Future Work. An abstract about this work has been accepted at CIDR 2022. We provide a prototypical implementation of our proposed approach at https://github.com/schelterlabs/arguseyes. In the future, we plan to add support for additional pipeline types (e.g., clustering, recommendation, learning-to-rank), and more detection techniques for correctness violations, especially over multiple pipeline executions. A current limitation of our approach is that we rely on the pipeline being written based on many declarative constructs from pandas and scikit-learn, which might often not be the case for data science code. We intend to increase the robustness of MLINSPECT and ARGUSEYES against such scripts.

This work was supported by Ahold Delhaize. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their employers.

REFERENCES

[1] S. Schelter, ""amnesia"-a selection of machine learning models that can forget user data very fast," *CIDR*, 2020.
[2] P. W. Koh, S. Sagawa, S. M. Xie *et al.*, "Wilds: A benchmark of in-the-wild distribution shifts," in *ICML*, 2021.
[3] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *NeurIPS*, 2019.
[4] R. Jia, D. Dao, B. Wang *et al.*, "Efficient task-specific data valuation for nearest neighbor algorithms," *PVLDB*, vol. 12, no. 11, 2019.
[5] M. Herschel, R. Diestelkämper, and H. B. Lahmar, "A survey on provenance: What for? what form? what from?" *The VLDB Journal*, 2017.
[6] T. J. Green, G. Karvounarakis, and V. Tannen, "Provenance semirings," *PODS*, 2007.
[7] S. Grafberger, J. Stoyanovich, and S. Schelter, "Lightweight inspection of data preprocessing in native machine learning pipelines." *CIDR*, 2021.
[8] M. Zaharia, A. Chen, A. Davidson *et al.*, "Accelerating the machine learning lifecycle with mlflow." *IEEE Data Engineering Bulletin*, 2018.