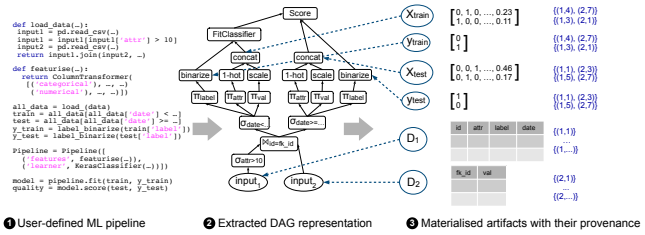
Screening Native ML Pipelines with "ArgusEyes"

Sebastian Schelter* Stefan Grafberger* Shubha Guha*
Olivier Sprangers* Bojan Karlaš† Ce Zhang†
*AIRLab, University of Amsterdam †ETH Zürich
[s.schelter,s.grafberger,s.guha,o.sprangers]@uva.nl [karlasb,ce.zhang]@inf.ethz.ch

Software systems that learn from data are being deployed in increasing numbers in industrial and institutional scenarios. Developing these machine learning (ML) applications imposes additional challenges beyond those of traditional software systems. The behavior of such applications very much depends on their input data, and they are based on systems and libraries from a relatively young data science ecosystem, which is rapidly evolving all the time. Experience shows that it is difficult to ensure that such ML applications are implemented correctly [Polyzotis et al. 2018, Stoyanovich et al. 2020], and as a consequence, data scientists building these applications require fundamental system support.

Correctness challenges in ML pipelines. Data scientists and ML engineers often unintentionally violate *sound experimentation practices for ML*, such as the strict isolation of train and test data, which can lead to non-reliable experimentation outcomes. Furthermore, there is a dire need to enforce *legal and ethical compliance* in decision making with ML, which introduces several technical challenges. For example, in order to determine whether a model works equally well for different groups [Stoyanovich et al. 2020], one needs to compute group fairness metrics for different subsets of the data, which is difficult, as sensitive attributes which identify groups may not directly be used by the model. For enforcing privacy rights (such as the 'right to be forgotten' [Schelter 2020]), we must identify which models actually consumed the user's data for model training, in order to retrain them without this data. Data scientists also require support for uncovering *erroneous data*, e.g., to identify samples which are dirty or mislabeled. Most of the listed issues are typically addressed manually in an ad-hoc way, due to a lack of system support for detecting particular issues. Furthermore, specialised solutions assume single-table input data and are often incompatible with the rest of the data science ecosystem.

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 (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) with mlinspect [Grafberger et al. 2021], 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 the 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, and can be easily hooked into continuous integration workflows. ArgusEyes is available at https://github.com/schelterlabs/arguseyes.

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

1. REFERENCES

[Grafberger et al. 2021] Stefan Grafberger et al. Lightweight Inspection of Data Preprocessing in Native Machine Learning Pipelines. *CIDR* (2021).

[Polyzotis et al. 2018] Neoklis Polyzotis et al. Data lifecycle challenges in production machine learning: a survey. *SIGMOD Record* 47, 2 (2018).

[Schelter 2020] Sebastian Schelter. "Amnesia"-A Selection of Machine Learning Models That Can Forget User Data Very Fast. *CIDR* (2020).

[Stoyanovich et al. 2020] Julia Stoyanovich et al. Responsible data management. *PVLDB* (2020).

12th Annual Conference on Innovative Data Systems Research (CIDR '22) January 10-13, 2022, Chaminade, USA.