Be Fair! Can Machine Learning Engineering Agents Adhere to Fairness Constraints?

Anna Richter
BIFOLD & TU Berlin
a.richter@tu-berlin.de

Julia Stoyanovich
New York University
stoyanovich@nyu.edu

Sebastian Schelter
BIFOLD & TU Berlin
schelter@tu-berlin.de

ABSTRACT

Machine learning engineering (MLE) agents promise to automate end-to-end ML pipeline development from raw data and natural language instructions, potentially making ML accessible to non-technical domain experts. However, in sensitive and regulated domains, this abstraction creates a responsibility gap: end-users may lack visibility into design choices that affect correctness, robustness, fairness, and regulatory compliance. We argue that existing benchmarks are insufficient to assess whether MLE agents can be safely applied in such settings. We propose desiderata for a responsibility-centered evaluation framework and conduct an exploratory study on melanoma classification, focusing on fairness across skin tones as a responsibility constraint. When evaluating two recent MLE agents, we find that agent-generated pipelines show high variance and consistently underperform manually designed baselines in both predictive quality and fairness, despite fairness-oriented prompts. These preliminary results suggest that further research is needed towards redesigning MLE agents to allow humans to guide the search process and reliably assess the compliance and quality of the generated ML pipelines.

VLDB Workshop Reference Format:
Anna Richter, Julia Stoyanovich, and Sebastian Schelter. Be Fair! Can Machine Learning Engineering Agents Adhere to Fairness Constraints?. VLDB 2026 Workshop: DASHSys: Systems for Data-centric Agents with Human-in-the-loop.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/anna-richter/be-fair.

1 INTRODUCTION

Machine learning (ML) is increasingly used to automate impactful decisions, and the risks arising from this widespread use are garnering attention from policy makers, scientists, and the media [24]. The resulting applications are often brittle with respect to their input data, which leads to concerns about their correctness, reliability, and fairness [11, 13, 15, 22].

The rise and promises of machine learning engineering agents. Designing, implementing and evaluating end-to-end ML pipelines for real-world decision making systems is tedious and requires a high level of technical expertise [24]. The recent progress in the code generation capabilities of large language models steered by agentic harnesses lead to a new class of agentic systems referred to as machine learning engineering (MLE) agents [7, 8, 14, 17]. These agents autonomously generate and optimize ML pipelines for a prediction task via agentic pipeline search [19]. They take raw data and a task description in natural language as input, and mimic the iterative process of a data scientist implementing, evaluating and improving their ML pipeline, with the search guided by the candidate pipeline's predictive performance on held-out data. The output of the MLE agent usually consists of an executable Python script, representing the highest performing pipeline found, together with this pipeline's predictions.

The responsibility gap. Marketing slogans for MLE agents promise full automation, claiming to transform "raw data into ready-to-use models and prediction outputs with minimal human intervention" [7] and to "enable individuals with limited domain expertise to address complex ML challenges effectively" [7]. Making ML engineering accessible to non-technical domain experts is a worthy goal: it could democratize data work, broaden participation in the field, and allow professionals such as doctors to build ML pipelines informed by their domain knowledge [2, 20]. Yet decisions made during data preparation, feature engineering, model selection, and pipeline development profoundly affect not only predictive performance, but also the correctness, robustness, fairness, and interpretability of automated decision systems [5, 10, 13, 15, 21, 24]. If MLE agents abstract away the development process, where and how can end-users exercise their duty of oversight over the resulting pipeline? This responsibility gap is especially concerning because people tend to overtrust machine-generated outputs [23]. In sensitive and regulated domains such as medicine, insufficient oversight is particularly problematic: ML pipelines must satisfy not only performance goals, but also standards of safety, accountability, and fairness [6, 9].

The need to evaluate the responsibility properties of MLE agents. The strong marketing claims for MLE agents are accompanied by impressive results on MLE-bench [3, 25], a popular OpenAI benchmark derived from Kaggle competitions. MLE-bench evaluates agents in the controlled setting of predictive modeling competitions, yet it neither tests whether generated pipelines satisfy regulatory compliance requirements nor measures what level of technical expertise is required to generate such pipelines successfully. This leaves open whether MLE agents perform reliably in real-world settings, and what level of expertise domain experts need to apply them safely. This limitation is especially important in sensitive and regulated domains such as healthcare, which account for a substantial share of MLE-bench: 14 of its 75 Kaggle competitions are medical tasks. We therefore ask: *Can existing MLE agents adhere to responsibility constraints when used by non-technical domain experts in sensitive application areas?*

Proceedings of the VLDB Endowment. ISSN 2150-8097.

Overview and contributions. In this exploratory work, we take a first step toward addressing this research question.

Contribution 1: Desiderata for a new evaluation framework. We outline the desiderata for a new evaluation framework for the responsibility properties of MLE agents in sensitive domains. Then we assess the widely-used MLE-bench benchmark against them (Section 2).

Contribution 2: Exploratory experiment. Based on the desiderata, we evaluate two recent MLE agents on a melanoma classification task under the responsibility constraint of classifier fairness across skin tones (Section 3). Our preliminary findings are concerning: despite explicit prompts to produce fair outputs, agent-generated pipelines exhibit high variance and consistently underperform manually designed baselines in both predictive quality and fairness (Section 4).

Contribution 3: Open artifacts. We release our code, detailed experimental logs and agentic trajectories under an open license at https://github.com/anna-richter/be-fair.

2 RESPONSIBILITY-CENTERED EVALUATION OF MLE AGENTS

We propose desiderata for the responsibility-centered evaluation of MLE agents and assess MLE-bench against them.

Domain-centric evaluation design. Evaluation tasks should preserve the real-world complexity of the application area. Rather than optimizing for a single metric, agents should be evaluated on their ability to account for multiple domain-specific objectives and trade-offs. This mirrors broader critiques of fairML benchmarking, which argue that intrinsic, single-metric evaluation strips away the normative context in which fairness harms materialize [18]. Moreover, performance should be measured on an independent test set unseen by the agent, and compared against peer-reviewed expert pipelines and human expert decisions.

Adherence to responsibility constraints. The evaluation framework should quantify the extent to which agent-generated pipelines satisfy relevant responsibility constraints, such as fairness requirements. This requires defining task-specific criteria and conducting a data-centric analysis of the generated pipelines, considering data access patterns, the variability of outputs, and the existence of fairness interventions.

Impact of technical expertise level. Finally, the evaluation framework should assess whether MLE agents are usable by domain experts with limited technical expertise, such as medical doctors or human resources professionals. In particular, it should measure the quality agents achieve out of the box, without technically refined prompting or debugging, and determine what level of refinement is needed to obtain high-quality results. This requires varying the specificity and technical detail of the task instructions given to the agent.

MLE-bench and its limitations. Most recently proposed MLE agents, including [7, 14, 17, 25], are evaluated on OpenAI's MLE-bench [3], a benchmark derived from 75 Kaggle competitions. Agents receive raw data and a natural language task description; success is measured by Kaggle's "medal winning rate," i.e., how often an agent places in the top decile of competition submissions. MLE-bench spans a wide range of domains, including 14 tasks from the highly regulated medical domain [6, 9]. Measured against the desiderata above, however, it has several conceptual limitations.

Limited relevance of the benchmark metric. It is unclear whether placing in the top decile of Kaggle submissions translates to sufficient quality in sensitive real-world settings.

Lack of multi-objective optimization. Kaggle solutions are typically ranked by a single competition-specific performance score. This setup does not reveal whether agents can reliably optimize pipelines for multiple objectives, such as accuracy and fairness.

Lack of metadata for responsibility assessment. Although MLE-bench includes 14 medical use cases, it often lacks the metadata needed to assess responsibility properties. For example, the SIIM-ISIC Melanoma Classification task [26] does not provide skin-tone annotations, making it impossible to evaluate the fairness of an agent-generated pipeline with respect to skin tone.

3 EXPLORATORY EXPERIMENT

Based on the outlined desiderata, we design an example evaluation task in the highly regulated medical domain [6, 9], focusing on melanoma classification for skin cancer detection. As a responsibility constraint, we require pipelines to provide comparable detection performance for individuals with different skin tones. Figure 1 provides an overview of the experimental design.

Curated datasets with rich metadata. We use "Fitzpatrick17k" [12] as the training dataset ❶, which contains over 17,000 images with skin-tone annotations based on the Fitzpatrick scale (a categorization of skin tones by their response to ultraviolet light). To reduce the risk that agents rely on memorized solutions, we obfuscate the dataset by renaming and shuffling columns and removing external URLs. As an independent test set, we use the "Diverse Dermatology Images (DDI)" [4] dataset, which is designed to evaluate skin-lesion classifiers for fairness across skin tones. DDI contains 656 images balanced across light and dark skin tones, with matched patient characteristics and diagnoses. It also includes baseline decisions from a panel of dermatologists who manually classified the images.

Task instructions of varying expertise. We provide each MLE agent with one of four task instructions ❷, designed to reflect increasing levels of technical expertise and specification detail.

- `base` provides a colloquial task description: *"I'm a dermatologist. My colleagues and I have curated a skin lesion dataset over the past several years. I need a model trained on this data to classify lesions as malignant or benign."*
- `fairness-hint` adds the instruction *"Be fair towards skin tone."*
- `fairness-metric` additionally specifies the fairness objective: *"The AUROC gap between light and dark skin should be minimal, without compromising overall model performance."*
- `fairness-methods` additionally suggests possible fairness interventions: *"To improve fairness across skin tones, you may apply techniques such as filtering, upsampling, group reweighting in the loss function, or any other appropriate methods."*

Agentic pipeline generation. We provide the training data and one selected task instruction to an MLE agent ❸, which then executes its search and optimization procedure to generate a classification pipeline as a Python script, potentially accompanied by a report.

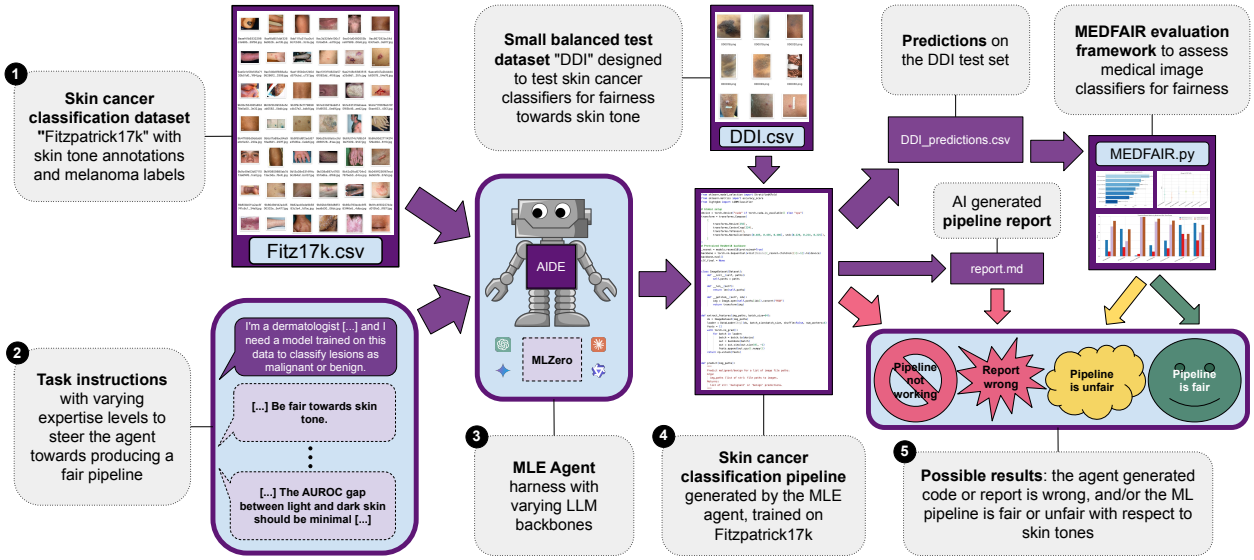

Figure 1: Overview of our exploratory experiment. ❶ A dataset about melanoma classification for skin cancer detection (with skin tone annotations), combined with ❷ natural language task instructions of varying technical expertise levels is given to an ❸ MLE agent. The agent generates an ML pipeline ❹ and an accompanying report for the task, which are subsequently evaluated ❺ for correctness, predictive performance and fairness, and compared to manually designed expert baselines.

Evaluation metrics and expert comparisons. We evaluate the agent-generated pipelines ❹ on DDI, following the peer-reviewed MEDFAIR study [27]. We reuse its code and metrics ❺: classification performance is measured by AUC, and fairness by the AUC gap, defined as the difference in AUC between the best- and worst-performing skin-tone groups. We compare the agent-generated pipelines against decisions from the DDI human expert panel and three human-written expert pipelines.

The expert pipelines apply "minimax pareto selection" [16] (a Pareto-efficient choice that minimizes the maximum downside across the overall AUC and the AUC gap across skin-tone groups) to decide on early stopping during training and the selection of the model checkpoint to return. They differ in the data cleaning strategies they employ: the first pipeline `MEDFAIR` follows [27] and removes images with missing Fitzpatrick labels, binarizes labels into malignant versus all other classes, resizes images during preprocessing, and fine-tunes a ResNet-18 model for 30 epochs. The second variant `MEDFAIR-dedup` trains the same pipeline on a cleaned version of Fitzpatrick17k [1] with duplicate and noisy images removed, while the third variant `MEDFAIR-filtered` further filters out images showing non-neoplastic conditions, which are not relevant to the binary classification task.

4 PRELIMINARY RESULTS

We present preliminary results for the exploratory experiment outlined in the previous section. We evaluate the MLE agent AIDE [14] (the original winner in MLE-bench) and the more recent AutoGluon-based agent MLZero [7] from Amazon. We give both a search budget of 20 steps, access to an A100 GPU, and a maximum execution time of one hour per step. We use the default configuration for AIDE which applies a combination of OpenAI's `gpt-4.1`, `gpt-4.1-mini`, and `o4-mini` models with a temperature of 0.5, and use OpenAI's `gpt-4.1` as LLM backbone for MLZero with the default temperature of 0.1. We run both agents seven times for each of the four task instructions (28 runs per agent in total). We report the mean and standard deviation of the prediction quality in terms of AUC and the fairness in terms of the AUC gap.

Baselines. As discussed in Section 3, we compare the agent-generated pipelines against several expert baselines. Most importantly, we include the expert decisions from [4] (referred to as `Dermatologists`), where a panel of three dermatologists manually classified the data. We also include the expert-written pipelines `MEDFAIR`, `MEDFAIR-dedup`, and `MEDFAIR-filtered`.

Results and discussion. We plot the results in Figure 2. AIDE achieves higher AUC scores and a lower fairness gap than MLZero, despite using substantially fewer tokens: approximately 6M versus 86M. However, we find that the agent-generated pipelines from AIDE and MLZero are strictly dominated by expert solutions in terms of both quality and fairness, and at the same time exhibit a much higher variance in their results. Furthermore, agents were unable to turn the skin-tone fairness instructions (`fairness-metrics` and `fairness-methods`) into actual fairness gains. All AIDE and MLZero pipelines score at least 10 points lower in AUC than expert decisions, and all agent-generated pipelines are drastically outperformed in terms of fairness by the expert baseline `MEDFAIR-dedup`, which applies the outlined minimax pareto selection as fairness intervention [16].

Detailed analysis of the generated pipelines. We analyze the code generated by AIDE and MLZero to determine whether the pipelines monitor or improve fairness with respect to skin tone, as requested in the task instructions. For AIDE, pipelines generated from the

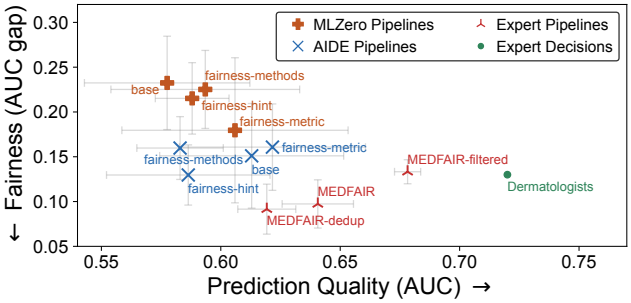

Figure 2: Prediction quality (AUC, higher is better) and fairness (AUC gap, lower is better) on the skin cancer detection task, comparing agent-generated pipelines, manually written expert pipelines, and expert decisions from dermatologists. Expert solutions strictly dominate the agent-generated pipelines on both quality and fairness, while the agents' results also show much higher variance.

base instruction, which does not mention fairness, and the `fairness-hint` instruction, which only asks the agent to be fair toward skin tone, neither compute fairness metrics nor apply fairness interventions. They also do not access the `skin_tone` column. Under the `fairnessmetric` instruction, 3 out of 7 pipelines correctly monitor and attempt to improve fairness with respect to skin tone. Under the highly specific `fairness-methods` instruction, all pipelines include fairness monitoring and interventions, often by reweighting samples according to skin-tone frequency. However, these interventions do not translate into improved fairness, as illustrated in Figure 2. Manual inspection of the pipelines' intermediate results showed that the darkest group's AUC was often close to random, suggesting that sample reweighting cannot remedy the group's weak discriminatory performance. For MLZero, about half of the pipelines generated from the base instruction pass `skin_tone` to AutoGluon as a tabular feature, but none apply explicit fairness interventions. Pipelines from the `fairness-hint` instruction also use `skin_tone` as a tabular feature and add class reweighting (independent of skin tone) as a fairness measure. The `fairness-metric` pipelines are similar, although two additionally report per-group AUC. Under the highly specific `fairness-methods` instruction, 2 of 7 pipelines implement explicit skin-tone-group reweighting, which again is accompanied by a close to random model performance on the the darkest group's samples and thus does not result in fairness gains.

Hallucinated reports and execution bugs. When experimenting with AIDE we encountered severe bugs and had to manually patch the framework to ensure that generated pipelines were actually executed. In particular, many generated scripts failed because they relied on an `if __name__ == __main__:` block but were launched in subprocesses without the correct "main" context. As a consequence, some runs completed without executing any pipeline successfully, yet the agent still produced final reports containing hallucinated summaries and fabricated performance metrics.

5 CONCLUSION & NEXT STEPS

Our exploratory study suggests that agent-generated pipelines underperform expert-designed baselines in both predictive quality and fairness, while also exhibiting substantial variance and implementation flaws. In our preliminary setting, prompting agents to produce fairer pipelines was not sufficient to reliably improve fairness outcomes. These findings point to a mismatch between the automation promises of current MLE agents and their readiness for high-stakes settings, motivating a comprehensive evaluation framework based on the desiderata from Section 2. We plan to cover a broader range of high-stakes scenarios, use frontier LLM backbones and complement benchmark scores with extrinsic, context-sensitive assessment [18]. Ablations will further help isolating the effects of MLE agent design, LLM backbone capability, and task complexity on responsibility constraint adherence and expertise-level requirements. In the long term, such a framework will guide research toward MLE agents that let humans meaningfully steer the search process and assess the compliance and quality of generated pipelines.

ACKNOWLEDGMENTS

The authors acknowledge the Scientific Computing of the IT Division at the Charité - Universitätsmedizin Berlin for providing computational resources that have contributed to the research results reported in this paper. This research was supported in part by NSF Awards No. 2312930 and 2326193.

REFERENCES

[1] Abhishek et al. Investigating the Quality of DermaMNIST and Fitzpatrick17k Dermatological Image Datasets. *Scientific Data* 12, 1, 2025.
[2] Boulamwini et al. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *FAccT*'18.
[3] Chan et al. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *ICLR*'25.
[4] Daneshjou et al. Disparities in dermatology AI performance on a diverse, curated clinical image set. *Science Advances* 8, 32, 2022.
[5] Erfanian et al. Chameleon: Foundation Models for Fairness-aware Multi-modal Data Augmentation to Enhance Coverage of Minorities. *VLDB*'24.
[6] EU AI Act, Regulation 2024/1689, https://eur-lex.europa.eu/eli/reg/2024/1689/oj.
[7] Fang et al. Mlzero: A multi-agent system for end-to-end machine learning automation. *NeurIPS*'25.
[8] Fathollahzadeh et al. CatDB: Data-Catalog-Guided, LLM-Based Generation of Data-Centric ML Pipelines. *VLDB*'25.
[9] FDA. Artificial Intelligence-Enabled Device Software Functions: Lifecycle Management and Marketing Submission Recommendations.
[10] Galhotra et al. Dataprism: Disconnect between data and systems. *SIGMOD*'22.
[11] Grafberger et al. Data distribution debugging in ML pipelines. *VLDBJ*'21.
[12] Groh et al. Evaluating deep neural networks trained on clinical images in dermatology with the fitzpatrick 17k dataset. *CVPR*'21.
[13] Guha et al. Automated data cleaning can hurt fairness in machine learning-based decision making. *TKDE*'23.
[14] Jiang et al. AIDE: AI-Driven Exploration in the Space of Code. *arXiv:2502.13138.*
[15] Karlaš et al. Navigating data errors in ML pipelines. *SIGMOD*'25.
[16] Martinez et al. Minimax pareto fairness: A multi objective perspective *ICML*'20.
[17] Nam et al. Mle-star: Machine learning engineering agent via search and targeted refinement. *NeurIPS*'25.
[18] Pechenizkiy et al. From Benchmarking to Understanding FairML. *ECAI*'25.
[19] Phani et al. stratum: A System Infrastructure for Massive Agent-Centric ML Workloads. *arXiv:2603.03589.*
[20] Sambasivan et al. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. *CHI*'25.
[21] Shahbazi et al. Through the fairness lens: Experimental analysis and evaluation of entity matching. *VLDB*'23.
[22] Schelter et al. Taming Technical Bias in ML Pipelines *IEEE DEBull*'20.
[23] Skitka et al. Does automation bias decision-making? *International Journal of Human-Computer Studies* 51, 5, 1999.
[24] Stoyanovich et al. Responsible data management. *Comm. ACM* 65, 6.
[25] Toledo et al. AI Research Agents for Machine Learning: Search, Exploration, and Generalization in MLE-bench. *NeurIPS*'25.
[26] Zawacki et al. SIIM-ISIC Melanoma Classification 2020, Kaggle. *https://kaggle.com/competitions/siim-isic-melanoma-classification.*
[27] Zong et al. MEDFAIR: benchmarking fairness for medical imaging. *ICLR*'22.