Studying Algorithmic Fairness in Moodle Learning Analytics Using Code Analysis

Hassan Tagharobi, Katharina Simbeck
University of Applied Sciences HTW Berlin, Berlin, Germany
Hassan.Tagharobi@HTW-Berlin.de
Katharina.Simbeck@HTW-Berlin.de

Abstract: Online learning platforms gained popularity in recent years. These platforms often provide learning analytics, which offer educators prediction on students' progress. For this purpose, machine learning algorithms are employed. As machine learning reproduces bias from the training data, such systems can potentially deliver unfair or discriminatory results. If the system outcomes are used to provide assistance or guidance to learners or if they influence educators' grading decisions, biased predictions affect students notably. It is therefore necessary to assess the fairness of learning analytics systems, especially when they use machine learning.
There are different ways to assess fairness of systems. In this paper, we discuss fairness assessment using code analysis. While it has been shown multiple times using real world training data that machine learning systems are biased and potentially unfair the objective of code analysis without the use of training data is to show the fairness of a system and its limitations. Code analysis thus complements the existing data driven fairness metrics. We propose a code analysis procedure to study the algorithmic fairness that consists of acquiring code and documentation, identifying and description of relevant system components, and fairness risk assessment. We conclude that the use of code analysis for the purpose of fairness auditing requires specialized knowledge about the application domain, programming, and machine learning. It is very time consuming and dependent on quality of code documentation.
We apply this approach on Moodle, a widely used open-source platform, and propose a code analysis procedure to study the algorithmic fairness of Moodle learning analytics. We identify relevant components (learning analytics is only a tiny fraction of the Moodle system), exam their role, and discuss the effect of data and user on fairness. Our analysis shows that Moodle learning analytics does not use protected attributes such as age, gender, or ethnicity. However, users must be aware of other potential fairness aspects. Furthermore, user's knowledge about machine learning, and evaluation metrics effects the fairness.

Keywords: Moodle, learning analytics, algorithmic fairness, code analysis, machine learning

1. Introduction

Learning analytics assists educators to evaluate and support students. To protect students from discriminatory, unfair treatment, it is crucial that these systems act fairly. Assessing data driven systems for fairness has become a widely discussed challenge. The European Union has recently proposed a framework for the regulation of artificial intelligence (AI) (European commission, 2021), stressing the importance of fairness especially for high-impact sector applications. In public discussions on the fairness of AI systems a call for code audits is often heard, reflecting the wide-spread opinion, that if only adequately trained experts had access to the source code, system fairness could be evaluated, even certified (Datenethikkommission der Bundesregierung, 2019; Stenkamp and Skierka, 2019). The outcome of such code audit is expected to be a recommendation on whether the system is fair, or under which circumstances it is not. In this paper we aim to discuss how code analysis can contribute to assessing algorithmic fairness. We will explore the process, benefits, and limitations of code analysis by applying it to the learning analytics functionality of Moodle.

2. Theoretical background

2.1 Learning analytics and algorithmic fairness

Siemens and Long (Siemens and Long, 2011) define learning analytics (LA) as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs". LA-applications and their importance have risen quickly in

recent years (Dawson et al., 2019), many studies aim to predict student success (Cambruzzi, Rigo and Barbosa, 2015; Lu et al., 2018; Xing et al., 2015).
Fairness of AI systems can be measured on an individual, a group, or a subgroup level (Mehrabi et al., 2019). Several toolkits have been proposed to measure fairness, e.g., (Bellamy et al., 2018; Madnani et al., 2017; Saleiro et al., 2018). Fairness in learning analytics has been studied for graduation prediction (Anderson, Boodhwani and Baker, 2019), or with regards to predictive quality between learner sub-groups (Riazy and Simbeck, 2019).

2.2 Code analysis and algorithmic fairness

Code analysis is one of multiple possibilities to assess the fairness of systems. Sandvig et al. propose four more possible audit designs: noninvasive user audit (studying user interactions with a platform), scraping audit (using repeated queries to a platform), sock puppet audit (researcher as users), and collaborative or crowdsourced audit (Sandvig et al., 2014). The latter four designs analyze system behavior using in and output, whereas code audit relies on source code understanding.
Binkley defines code analysis as "the process of extracting information about a program from its source code" (Binkley, 2007). Source code analysis can focus solely on technical issues, for example with regards to the use of variables and pointers, in those cases it can be supported using toolkits (Novak and Krajnc, 2010). With regards to AI accountability Raji et. al. introduce a framework for an internal fairness auditing process (Raji et al., 2020). This framework includes scoping, mapping, artifact collection, testing and reflection stages. Wilson et al. use code, documentation and representative datasets to audit an HR-candidate-search platform for fairness (Wilson et al., 2021).

2.3 Learning Analytics in Moodle

Moodle is an open-source learning management system that provides learning environments for educators. Moodle includes a learning analytics functionality to analyze learning process and progress of students. Organizations can use Moodle without it.
The Moodle learning analytics functionality has been the subject of research, e.g., (Fenu, Marras and Meles, 2017; Verdu et al., 2021; Zhang, Ghandour and Shestak, 2020) but not with regards to fairness. Bognar and Fauszt have analyzed the impact of dataset size on predictors and time splitting method in Moodle (Bognar and Fauszt, 2020), their focus was mainly on performance and accuracy, not fairness. Thus, we believe, a comprehensive analysis of the Moodle learning analytics system, its abilities limitations, and fairness constitutes a research gap. This is becoming more essential, since educators tend to rely on outcomes of LA systems in their perception and evaluation of students (Mai, Köchling and Wehner, 2021).

3. Fairness of Moodle Learning Analytics

In this paper, we are aiming to identify potential fairness risks arising from using Moodle learning analytics. Therefore, we will identify relevant components of Moodle software for learning analytics and examine them. Machine learning (ML) has been associated with fairness issues in learning analytics and other applications. Thus, we will investigate the employment of machine learning in Moodle LA and its impact. It is crucial to ensure, student success does not depend on protected attributes, e.g., gender, ethnicity, or age. Consequently, we will check whether such attributes are implicitly or explicitly used in Moodle LA.
We are proposing a structured approach to assess fairness of learning analytic systems using code analysis on the example of Moodle. For this purpose, we will first confine relevant resources (source code and documentation). We will then conduct a descriptive analysis of code and documentation to enhance understanding of the dimensions of the project and accessible information. Next, we identify important classes and study their functions and relationships. We look at major components individually and analyzed their influence on fairness of the system. Finally, we examine their combined implication and the dependencies on data, user, and proper employment.

3.1 Confining the LA system

In scope for this analysis is only the learning analytics functionality of Moodle. The analysis has been conducted on Moodle 3.10.4 as of May 2021. Both the source code and the documentation are used for analysis. As a first step, the code of the learning analytics functionality needs to be identified in the source code. We excluded 37% of code files that only implements web functionality and design of Moodle, such as HTML, CSS and JavaScript. In Moodle there are over 2.6 Mio. PHP lines of code (LOC), of which 1.5 Mio. are non-comment lines of code

(NCLOC) in 11,175 files. With over 16 thousand NCLOC learning analytics represents only about 1% of Moodle (Table 1). Similar ratios also apply across classes and packages. The Python repository contains additional 1,140 LOC. For calculating properties of code, we reimplemented and compared three approaches: LOC and NCLOC (AlDaniel, 2021), classes (Bergmann, 2021) and packages using our own-developed method.

Table 1: Number of files, packages, classes, and lines of code in Moodle and Moodle learning analytics (only PHP)

	Files	LOC	NCLOC	Packages	classes
Moodle	11,175	2,642,663	1,539,664	565	12,201
Moodle LA	254	34,389	16,766	4	214
LA share in Moodle (%)	2.27	1.30	1.08	0.71	1.75

Moodle provides a multilingual documentation for users in ten different languages, and an English documentation for developers. To estimate the dimension of the documentation, we collected all corresponding pages and counted words. The length of the documentation varies between different languages. For example, English documentation has about 5% more words than German documentation. The user documentation offers information and instructions for optimal usage of the platform, whereas the developer documentation dives deeper in the structure of the project and functionality of modules. The developer documentation is much shorter than the user documentation and does not contain a comprehensive module-structure, class, or code explanation. Table 2 shows an overview of the number of words in the different language documentations in English and German documentations.

Table 2: English and German Moodle documentation in numbers

	User doc	Dev. doc	LA user doc	LA dev doc
English	414,993	198,098	14,337	7,605
German	392,194	n/a	12,574	n/a

3.2 Descriptive analysis

Moodle learning analytics offers descriptive analysis of student learning, and predictions about success of students (Moodle, 2021e). There are two kinds of models in Moodle LA: static and machine learning based. Static models use simple rules to detect specific situations whereas machine learning models employ machine learning algorithms and previous data to predict a situation, e.g., "Students at risk of dropping out". The ML-based LA provide two machine learning backends (PHP and Python). Users (administrators, educators) can choose a backend. Although various classification and regression algorithms are implemented, the PHP backend (default) currently only uses logistic regression. The Python backend applies a single hidden layer neural network using TensorFlow (Moodle, 2021d). PHP code is part of Moodle main project whereas Python is available via an extra repository.

To apply ML-based learning analytics and predict an event, the user must first create or choose a model. There are some predefined models already included in Moodle. For example, “Students at risk of dropping out”. Users can also create their own models, which contains multiple features and predict specific events. Figure 1 shows the major components of ML-based LA in Moodle.

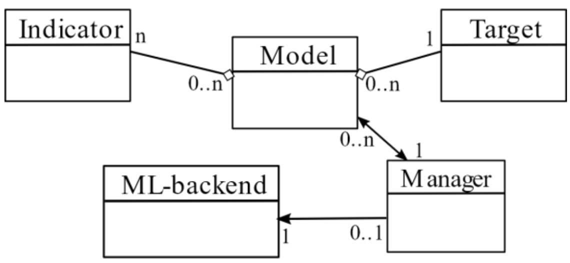

Figure 1: Simplified class diagram of ML-based LA in Moodle, based on (Moodle, 2021a)

3.3 Major components and their implication for fairness

In Moodle, indicators are features that measure a specific property or activity. There are some general features, e.g., “any write action”, “read actions amount”, “User is tracking forums”, as well as a set of indicators based on the “Community of Inquiry model of student engagement”. These exist in two types: cognitive depth and social breadth (Moodle, 2021b). Users can also import and use further indicators. To train and use ML-based LA, users need to provide training data. This fact has two-sided implications. On the one hand, it assures the used data is known by the users. On the other hand, it is more probable that the ML-Method is trained with a smaller, less diverse dataset, which could lead to unfavorable results. To ensure fair LA, indicators must be defined carefully. It is important that features do not draw a conclusion from activity, nor rely on actions that are not decisive for the purpose. Furthermore, relying heavily on on-site activities neglect offline work of students. In Moodle, protected attributes like gender, ethnicity etc. cannot be used for indicators. Yet, it must be examined whether a feature could leak protected information.

Targets are events, that LA tries to predict. Moodle provides a growing list of targets, e.g., “Students at risk of dropping out”, “Students at risk of not achieving the minimum grade to pass the course”, “Students at risk of not meeting the course completion conditions” (Moodle, 2021c). Meanwhile, the value of target in previous data represent the label for ML-algorithm. Regarding fairness, it is important to ensure whether the target is predictable given the available indicators. For example, is the available data conclusive enough to surely infer, if a student will pass or fail a class. Additionally, since the outcome of previous courses act as labels for machine learning algorithm, it is vital that the outcome was a result of the available interaction in Moodle. For example, in case there were on-site and offline activities, the resulting grade or outcome (pass or fail) might not be correlating solely to data and create inaccurate implications.

A model is a set of features (indicators) that should predict an outcome (target). Moodle provides some predefined models with predefined features and targets (Moodle, 2021a). For example, “Students at risk of dropping out” uses 49 indicators to predict students at risk. Users can also create their own models with an individual set of indicators. Regarding fairness the composition of the model is crucial. Choosing a proper set of indicators for the predicted event can improve fairness of the LA-system. It is also vital to examine, if the event is predictable using available features, as well as prediction fairness regarding offline activities.

The ML-backend implements the functions as the engine of ML-based LA. It uses previous data and associated labels to train and later predict the selected event. The two different ML-backends in Moodle use logistic regression (PHP backend) and neural networks (Python backend) (Moodle, 2021d). Different stages of the ML-backend play a role in ensuring fairness. Firstly, choosing the proper algorithm could improve prediction accuracy, and thus fairness. This decision is best made regarding data, outcome, and resources. It is debatable if a single pre-selected algorithm will suit different models. Secondly, training is the most important part of ML-prediction. To reach suitable results, the dataset must satisfy basic criteria, e.g., size of the dataset, balance among different labels, proper randomization of data splitting for training and testing (for example through multiple execution and comparison). Thirdly, the evaluation of results is a vital part of ensuring fairness. ML-algorithms are not perfect. This should be clear for all stakeholders, especially users. Each algorithm or application will have their strength and weaknesses. The evaluation can be represented with different metrics. For proper and fair application of ML-processes it is important that users have clear understanding of these metrics and their implications.

Table 3: Major components of ML-based LA and their relevance for fairness

Component	Relevant for fairness in implementation	Relevant factors for fairness in application
Indicator	properly measurable indicators explicit, unambiguous features	indicator selection
Target	predictable target using available data	measurable target
Model	model definition model composition	model composition consideration offline-activity
ML-Backend	suitable algorithm proper configuration rigorous evaluation	understanding of ML-process familiarity with evaluation metrics

Fourthly, in all stages of the ML-process, choosing the proper parameters leads to significant changes in results. Thus, it is important to continuously examine their effect on the result and its fairness. Moodle LA enables users to quickly employ a model and create predictions regardless of their ML-knowledge. This can lead to severe fairness problems since these users will not be able to assess the validity of the predictions, and in worst case trust unreliable results. The fairness risks of the Moodle LA components are summarized in table 3.

4. Discussion

It is often thought that access to source code of an intelligent system can significantly contribute to assessing fairness. Although, this can improve possibilities for the assessment, especially regarding possible problems in design or implementation, it has its own difficulties.
Firstly, source code review is a very time and resource consuming process. On developing projects like Moodle, this might need to be a recurring process. A comprehensive documentation is also greatly beneficial. In case of Moodle, documentation is not sufficient for this purpose. Furthermore, reviewers need to have a solid understanding of the project and related areas, e.g., programming, machine learning. Consequently, this process cannot be administered at end-user level, like schools or universities.
Secondly, data influences the outcome of ML-based systems notably. In an untrained delivered system like Moodle, the same model could result in quite different predictions for different users. Therefore, assessing fairness of a system cannot be achieved independently of the data.
Thirdly, proper usage of a predictive system determines the fairness of the outcome. The user should be able to assess the results and their validity. This is especially crucial for systems, like Moodle, where users must configure, train, and apply the model. In order to ensure fairness in a LA-system, users must be properly informed about these challenges and be provided with necessary information about abilities, and limitations of the system, as well as data science know-how for interpretation of results.
Despite these factors, code analysis offers valuable insights about abilities and limitations of learning analytics systems like Moodle LA. It can expose vulnerabilities and potential fairness risks caused by assumptions and parametrization in source code. Our approach attempts to address this research gap, as majority of other studies focus on performance optimization (Bognar and Fauszt, 2020) or primarily use data-based analysis to assess algorithmic fairness (Anderson, Boodhwani and Baker, 2019).
We believe, code analysis and data-based analysis can complement each other and thus, ensure more comprehensive fairness examination. We aim to conduct such extensive study in future works.

5. Conclusion

In this paper we make two major contributions. Firstly, we discuss an approach to assessing AI fairness using code analysis. Secondly, we apply this approach to the Moodle learning analytics and thus give insights about its inherent fairness. In order to assess AI systems for fairness, the following steps are necessary:

- Acquire system code and documentation,
- Identify relevant system components for fairness analysis,
- Descriptive analysis of relevant parts of codes (size, class diagram),
- Description of relevant classes as well as input and output data,
- Fairness risk assessment considering application, dataset, and user interaction.

Even though it is widely discussed that AI systems' code should be analyzed for fairness, code analysis can only provide limited insights into the fairness of an AI system. Our analysis shows that Moodle LA doesn't use protected attributes such as age, gender or ethnicity. Nevertheless, users must be aware of other potential individual fairness perspectives. Moodle offers two types of LA, one of them uses machine learning algorithm. Different components carry fairness risk potentials. A cautious use of ML-based LA, especially regarding to data, model and evaluation is crucial for a fair LA-usage. User's knowledge about machine learning, evaluation metrics etc. can have big impact on fair, proper application.
In future works, we will continue fairness-assessment of Moodle by a more detailed modular examination of LA-components and their dependencies to data and users. We will use synaptic data and a study group to assess measures for ensuring fairness in learning analytics predictive systems.

Acknowledgement
This research has been funded by the Federal Ministry of Education and Research of Germany in the project "Fair Enough?" (Project ID: 16DHB4002).

6. Reference list

AlDaniel (2021) AlDanial/cloc, 11 June 2021. Available at: https://github.com/AlDanial/cloc (Accessed: 11 June 2021).
Anderson, H., Boodhwani, A. and Baker, R.S. (2019) 'Assessing the Fairness of Graduation Predictions', EDM.
Bellamy, R.K.E. et al. (2018) 'AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias', arXiv preprint arXiv:1810.01943.
Bergmann, S. (2021) sebastianbergmann/phploc, 11 June 2021. Available at: https://github.com/sebastianbergmann/phploc (Accessed: 11 June 2021.132Z).
Binkley, D. (2007) 'Source Code Analysis: A Road Map', Future of Software Engineering (FOSE'07), pp. 104–119.
Bognar, L. and Fauszt, T. (2020) 'Different learning predictors and their effects for Moodle Machine Learning models', 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), pp. 405–410.
Cambruzzi, W.L., Rigo, S.L. and Barbosa, J.L. (2015) 'Dropout prediction and reduction in distance education courses with the learning analytics multitrail approach', J. UCS, 21(1), pp. 23–47.
Datenethikkommission der Bundesregierung (2019) 'Gutachten der Datenethikkommission der Bundesregierung'.
Dawson, S. et al. (2019) 'Increasing the Impact of Learning Analytics', Proceedings of the 9th International Conference on Learning Analytics & Knowledge, pp. 446–455.
European commission (2021) 'Regulation of the european parliament and of the council, Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, COM(2021) 206 final'.
Fenu, G., Marras, M. and Meles, M. (2017) 'A Learning Analytics Tool for Usability Assessment in Moodle Environments'. Journal of e-Learning and Knowledge Society, 13(3), pp. 331–342.
Lu, O.H.T. et al. (2018) 'Applying learning analytics for the early prediction of Students' academic performance in blended learning', Journal of Educational Technology & Society, 21(2), pp. 220–232.
Madnani, N. et al. (2017) 'Building Better Open-Source Tools to Support Fairness in Automated Scoring', Proceedings of the First ACL Workshop on Ethics in Natural Language Processing, pp. 41–52.
Mai, L., Köchling, A. and Wehner, M. (2021) ''This Student Needs to Stay Back': To What Degree Would Instructors Rely on the Recommendation of Learning Analytics?' CSEDU (1), pp. 189–197.
Mehrabi, N. et al. (2019) 'A Survey on Bias and Fairness in Machine Learning', arXiv preprint arXiv:1908.09635.
Moodle (2021a) Analytics API - MoodleDocs, 18 May.2021. Available at: https://docs.moodle.org/dev/Analytics_API (Accessed: 4 June 2021).
Moodle (2021b) Learning analytics indicators - MoodleDocs, 18 May.2021. Available at: https://docs.moodle.org/310/en/Learning_analytics_indicators (Accessed: 4 June 2021).
Moodle (2021c) Learning analytics targets - MoodleDocs, 18 May.2021. Available at: https://docs.moodle.org/310/en/Learning_analytics_targets (Accessed: 4 June 2021).
Moodle (2021d) Machine learning backends - MoodleDocs, 18 May.2021. Available at: https://docs.moodle.org/dev/Machine_learning_backends (Accessed: 5 June 2021).
Moodle (2021e) Analytics - MoodleDocs, 7 June.2021. Available at: https://docs.moodle.org/311/en/Analytics (Accessed: 16 June 2021).
Novak, J. and Krajnc, A. (2010) 'Taxonomy of static code analysis tools', The 33rd International Convention MIPRO, The 33rd International Convention MIPRO, pp. 418–422.
Raji, I.D. et al. (2020) 'Closing the AI accountability gap', Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 33–44.
Riazy, S. and Simbeck, K. (2019) 'Predictive Algorithms in Learning Analytics and their Fairness', DELFI 2019, pp. 223–228.
Saleiro, P. et al. (2018) 'Aequitas: A Bias and Fairness Audit Toolkit', arXiv preprint arXiv:1811.05577.
Sandvig, C. et al. (2014) 'Auditing algorithms: Research methods for detecting discrimination on internet platforms', Data and discrimination: converting critical concerns into productive inquiry (22), pp. 4349–4357.
Siemens, George and Long, Phil (2011) 'Penetrating the Fog: Analytics in Learning and Education', EDUCAUSE Review, 46(5), p. 30.
Stenkamp, D. and Skierka, S. (2019) 'Ein TÜV für Algorithmen', Handelsblatt, 27 November, p. 48.
Verdu, M.J. et al. (2021) 'MSocial: Practical Integration of Social Learning Analytics Into Moodle', IEEE Access, 9, pp. 23705–23716.

Wilson, C. et al. (2021) 'Building and Auditing Fair Algorithms: A Case Study in Candidate Screening', Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event Canada, pp. 666–677.
Xing, W. et al. (2015) 'Participation-based student final performance prediction model through interpretable Genetic Programming: Integrating learning analytics, educational data mining and theory', Computers in Human Behavior, 47, pp. 168–181.
Zhang, Y., Ghandour, A. and Shestak, V. (2020) 'Using Learning Analytics to Predict Students Performance in Moodle LMS', International Journal of Emerging Technologies in Learning (iJET), 15(20), p. 102.