Algorithmic Bias in Multi-Class Classification: Fairness Metrics and Methods

Giordano d'Aloisio, Giovanni Stilo

Over the past decade, the study of bias and fairness in machine learning has gained significant importance. Numerous definitions and metrics have been introduced to address various types of bias and fairness [23]. In this document, we first outline the definitions of bias and fairness that underpin the basis analysis we conducted in the academic domain. Subsequently, for this purpose, we review related research on bias mitigation within the context of binary and multi-class classification tasks. Finally, we will provide details on the implementation needed by the low-code platform we used to conduct the fairness assessment in academia.

1 Bias and Fairness definitions

Bias and unfairness can originate from various sources and be conceptualized in multiple ways. In [23], the authors identified several potential sources of bias:

- the **data** used to train the ML algorithms (e.g., *Measurement bias* [27], *Omitted Variable bias* [9, 6], or *Representation bias* [27]);
- the **algorithm** which may introduce bias in the users' behavior (e.g., *Algorithmic bias* [4]);
- the **population**, which generates the data used to train the models (e.g., *Historical bias* [27], *Population bias* [24], or *Social bias* [4]).

The former definitions of bias, with the only exception of *Algorithmic bias*, which is strongly related to the ML algorithm, can be grouped into two macro-categories of bias:

- **Unbalanced Groups bias**: in which the bias is generated by an unequal distribution of instances in the population (e.g., *Representation bias*, *Historical bias*, *Social bias*, *Population bias*)

- **Confounding Variables bias**: in which the bias is generated by a wrong interpretation or representation of instances in the population (e.g., *Measurement bias*, *Omitted Variable bias*)

Most of the methods available in the literature address the first category of bias [14, 25, 7, 19], while the second category is more common in neural networks [30].

Concerning fairness definitions, *Demographic (Statistical) Parity (DP)* [22, 13] is one of the most used definitions of *group fairness* [23], which assumes the independence among the predicted positive label y_p and the sensitive variables $S_1, \ldots, S_n$. It is defined formally as follows:

Definition 1 (Demographic Parity) *Let $\hat{Y}$ be the predicted value, y_p the positive label, and S a generic binary sensitive variable where $S = 1$ and $S = 0$ identify, respectively, the privileged and unprivileged groups. A predictor is fair under Demographic Parity if:*

$$P(\hat{Y} = y_p | S = 1) = P(\hat{Y} = y_p | S = 0) \tag{1}$$

A different formulation for the DP is the *Disparate Impact (DI)* [15], which considers the ratio among the two probabilities. In this case, following the *80% rule* [15], the value must be between 0.8 and 1.2 to have *fairness*. DI is defined formally as follows:

Definition 2 (Disparate Impact) *Let $\hat{Y}$ be the predicted value, y_p the positive label, and S a generic binary sensitive variable where $S = 1$ and $S = 0$ identify the privileged and unprivileged groups, respectively. A predictor is fair under Disparate Impact if:*

$$0.8 \leq \frac{P(\hat{Y} = y_p | S = 1)}{P(\hat{Y} = y_p | S = 0)} \leq 1.2 \tag{2}$$

Equalised Odds (EO) [18] is the third definition of fairness we consider which overcomes the limitation of DP by not removing the correlation between the true and predicted outcomes [28, 18]. In fact, a classifier is considered fair under EO if the probability of an item being positively classified is the same concerning the sensitive variable and the ground truth. EO is formally defined as follows:

Definition 3 (Equalized Odds) *Let $\hat{Y}$ be the predicted value, Y the true value, y_p the positive label, and S a generic binary sensitive variable where $S = 1$ and $S = 0$ identify the privileged and unprivileged groups, respectively. A predictor is fair under Equalized Odds if:*

$$P(\hat{Y} = y_p | Y = y, S = 1) = P(\hat{Y} = y_p | Y = y, S = 0) \quad y \in \{y_1, \ldots, y_n\} \tag{3}$$

Both DP and DI fall into the *We Are Equal* metrics family, which holds that all groups have similar abilities concerning the task (i.e., have the same probability of being classified in a certain way). On the contrary, EO resides in the *What You See Is What You Get* family, which states that the observations reflect the ability with respect to the task (i.e., an item should be classified in a certain way only if the other attributes imply it) [16].

All these definitions were initially proposed for binary classification problems ($y_p = 1$). Still, they can be extended to the multi-class classification domain by identifying one positive label value among the possible ones ($y_p \in \{y_1, \ldots, y_n\}$).

2 Multi-Class Fairness Methods

Over the years, many methods have been proposed to mitigate bias at different levels of data processing[23, 8]. In particular, we distinguish among [10]:

- **Pre-processing** methods, which modify the data to remove the underlying bias, such as, [19, 15];
- **In-processing** methods, which change the learning algorithm to remove discrimination during the model training process, such as [12, 3];
- **Post-processing** methods, which re-calibrate an already trained model using a holdout set not used during the training phase, such as [18, 26].

The sooner a technique can be applied, the better because it can be chained with other bias mitigation methods in the later processing phases [29, 2].

Among the different machine learning problems (i.e. classification, regression, clustering, etc.), the classification task has been the most addressed in bias mitigation [23, 8]. In the following, we will focus on stable methods[1] to improve fairness in the classification task.

Most of the methods available in the literature focus only on binary classification with one sensitive variable [23]. Among them, one widely adopted *pre-processing* method is the *Sampling* algorithm proposed by [19]. This method balances privileged and unprivileged users in the case of binary classification with a single sensitive variable. Formally, let be S the sensitive variable with $\{w, b\} \in S$ representing the privileged and unprivileged groups, respectively, and let be Y the target label with $\{+, -\} \in Y$ defining the positive and negative outcomes. The *Sampling* algorithm first splits the original dataset into four groups:

- Deprived group with Positive label (DP): all instances with $S = b \wedge Y = +$;

[1]Stable methods are the ones having an available and stable implementation.

- Deprived group with Negative label (DN): all instances with $S = b \wedge Y = -$;
- Favored group with Positive label (FP): all instances with $S = w \wedge Y = +$;
- Favored group with Negative label (FN): all instances with $S = w \wedge Y = -$.

Then, for each group, the algorithm computes its *observed* and *expected* sizes. Finally, it balances the groups iteratively by randomly adding and removing instances until the *observed* sizes of the groups are equal to their *expected* ones.

Very few methods are able to mitigate the bias in the multi-class classification problems [26, 3, 14]. Among those, there is the *Blackbox post-processing* method proposed by [26]. The authors extend the method proposed by [18] to the multi-class setting. Their approach involves the construction of a linear program over the conditional probabilities of the adjusted predictor $P(Y^{adj} = y^{adj} | \hat{Y} = \hat{y}, A = a)$ such that the desired fairness criterion is satisfied by those probabilities. In order to build the linear program, the authors formulate both the loss and fairness criteria as linear constraints in terms of the protected attribute conditional probability matrices. Then, this linear program is used to find the label value, among the possible ones, that minimises both the loss and the fairness constraints.

An *in-processing* method that solves unfairness in multiple classification settings is the one presented by [3]. The algorithm addresses two definitions of fairness at once: *Demographic Parity* and *Equalized Odds*. The authors formulate such definitions as linear constraints and then build an Exponentiated Gradient (EG) reduction algorithm [21] that yields a randomised classifier with the lowest error subject to the desired fairness constraints. The method follows a MinMax approach in which the players try to minimise the given constraint and maximise the classifier's score. The authors also propose a simplified Grid Search version of the algorithm (GRID), which generates a sequence of relabelling and reweightings, and trains a predictor for each one. The values yielding the best *accuracy* and *fairness* trade-off are selected and thus returned. Although the authors study their algorithms mainly in binary classification problems, they also show how their method can be applied to regression and multi-classification problems.

Finally, a *pre-processing* method able to improve fairness both in binary and multi-class problems in an explainable way is the *Debiaser for Multiple Variables (DEMV)* algorithm presented by d'Aloisio et al. in [14]. This algorithm extends the Sampling algorithm of [19] by considering sensitive groups identified by all possible combinations of the values of sensitive variables and the values of the label. In particular, a sensitive group is defined as $\{X \in D | S_1 == s_1 \wedge S_2 == s_2 \wedge \cdots \wedge S_n == s_n \wedge L == l\}$, where $s_1, \ldots, s_n$ are possible values of the sensitive variables and l is a value of the label.

Then, for each group, the algorithm computes their observed (W_{obs}) and expected (W_{exp}) sizes, defined respectively as:

$$W_{obs} = \frac{|\{X \in D|S = s \wedge L = l\}|}{|D|} \tag{4}$$

$$W_{exp} = \frac{|\{X \in D|S = s\}|}{|D|} \times \frac{|\{X \in D|L = l\}|}{|D|} \tag{5}$$

where $S = s$ is a generic condition on the value of the sensitive variables and $L = l$ is a condition on the label's value. If $W_{obs} \backslash W_{exp} < 1$, it means that the size of the group is smaller than expected, so the algorithm randomly duplicates an item of that group. Instead, if $W_{obs} \backslash W_{exp} > 1$, it means that the size of the group is larger than expected, so the algorithm randomly removes an item from the group. For each sensitive group, the algorithm repeats this process until the group is fully balanced (i.e., $W_{obs} \backslash W_{exp} = 1$).

3 Assessment Implementation

We conduct our fairness assessment in academics using MANILA, a low-code tool for developing quality ML systems [11]. MANILA automatically generates an experiment that evaluates different ML classifier and fairness method combinations and selects the one achieving the best fairness and effectiveness [5] trade-off. The conducted assessment was implemented using MANILA, a low-code platform designed for developing high-quality machine learning systems [11]. We recall that MANILA automatically constructs experiments to evaluate various combinations of machine learning classifiers and fairness methods, identifying the configuration that optimally balances fairness and effectiveness [5].

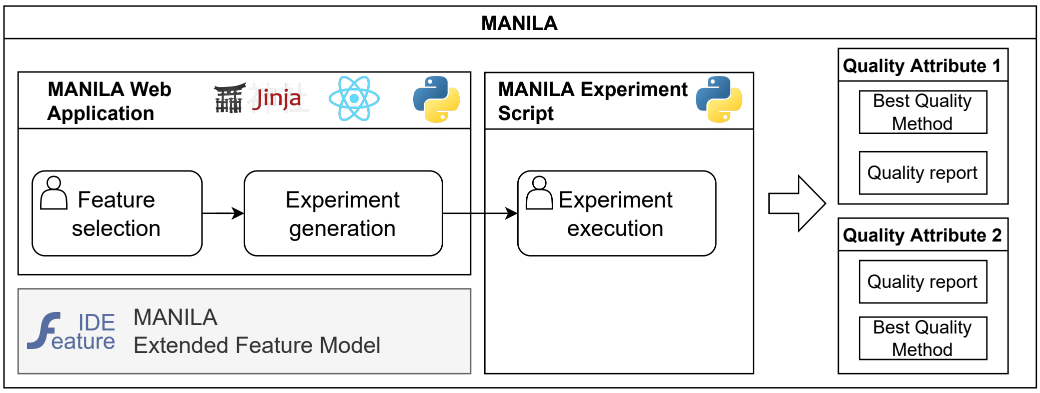

Figure 1: MANILA Architecture as in [11]

As described in [11], Figure 1 illustrates the high-level architecture of MANILA. Rounded boxes denote steps within the quality-driven development process, while square boxes represent corresponding artifacts. Steps involving human intervention are indicated by boxes containing a user icon. Additionally, the tools utilized for the implementation of each artifact are listed adjacent to the respective artifact.

As in [11], we recall that the initial step in the quality-driven development process involves selecting the features that constitute the experimental evaluation. These features include machine learning models, quality-enhancing methods, metrics, dataset characteristics (e.g., label types or sensitive variables), data scaling methods, cross-validation techniques, and result presentation formats such as tables or charts. The data scientist might perform this step via a dedicated web application.

Subsequently, the MANILA system automatically generates a set of Python scripts to implement the experimental evaluation. The data scientist can then execute these scripts directly using a Python interpreter. The execution produces a series of reports for each selected attribute and identifies the machine learning tuple (comprising the ML algorithm and quality-enhancing method) that achieves the highest metric.

The process, as in the original paper, is underpinned by an Extended Feature Model (ExtFM), which models the entire system [20] as a Software Product Line [17]. The ExtFM defines constraints among features, such as between ML models and quality-enhancing methods, ensuring the configuration of valid and executable experiments. These constraints provide guidance to the data scientist, ensuring that all experiments are properly configured and executable.

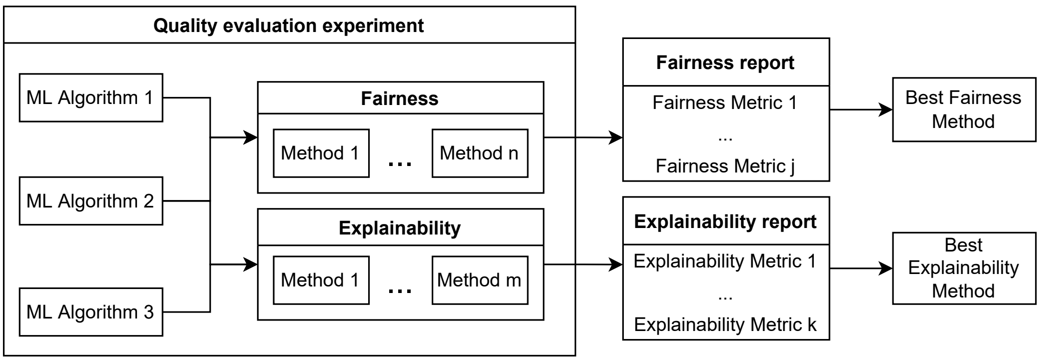

Figure 2: Experimental evaluation as in [11]

In line with the original paper, Figure 2 illustrates an example of how MANILA conducts a quality evaluation experiment. In this scenario, the data scientist selects three machine learning algorithms and aims to ensure both Fairness and

Explainability. To achieve these objectives, the data scientist chooses n methods to enhance Fairness and m methods to enhance Explainability, along with j metrics for Fairness and k metrics for Explainability.

Here, the evaluation process then executes two parallel sets of experiments. In the first set, the n Fairness methods are applied to each machine learning algorithm, and the j Fairness metrics are computed. The second set applies the m Explainability methods to the algorithms, and the k Explainability metrics are computed. Finally, the process generates two reports summarizing the results for Fairness and Explainability, respectively, as well as identifying the machine learning algorithms that achieve the best performance in each category.

The results are saved in a CSV file if the data scientist opts to view the results in tabular form by selecting the **Tabular** feature in the ExtFM. Otherwise, the results are visualized as charts saved in PNG format. Additionally, the machine learning algorithm identified as the best performer in the experiment is saved as a serialized *pickle* file [1].

4 Additional Acknowledges

This report is partially supported by "FAIR-EDU: Promote FAIRness in EDUcation institutions" a project founded by the University of L'Aquila, 2022.

References

[1] Pickle documentation.

[2] AI Fairness 360 - Resources, 2018.

[3] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR, 10–15 Jul 2018.

[4] Ricardo Baeza-Yates. Bias on the web. *Communications of the ACM*, 61(6):54–61, 2018.

[5] Houssem Ben Braiek and Foutse Khomh. On testing machine learning programs. *Journal of Systems and Software*, 164:110542, 2020.

[6] John R Busenbark, Hyunjung Yoon, Daniel L Gamache, and Michael C Withers. Omitted variable bias: Examining management research with the

impact threshold of a confounding variable (itcv). *Journal of Management*, 48(1):17–48, 2022.

[7] Toon Calders, Asim Karim, Faisal Kamiran, Wasif Ali, and Xiangliang Zhang. Controlling Attribute Effect in Linear Regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 71–80, December 2013. ISSN: 2374-8486.

[8] Simon Caton and Christian Haas. Fairness in Machine Learning: A Survey. *arXiv:2010.04053 [cs, stat]*, October 2020. arXiv: 2010.04053.

[9] Kevin A Clarke. The phantom menace: Omitted variable bias in econometric research. *Conflict management and peace science*, 22(4):341–352, 2005.

[10] Brian d'Alessandro, Cathy O'Neil, and Tom LaGatta. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data*, 5(2):120–134, 2017.

[11] Giordano d'Aloisio, Antinisca Di Marco, and Giovanni Stilo. Democratizing quality-based machine learning development through extended feature models. In *26th International Conference on Fundamental Approaches to Software Engineering*, 2023.

[12] Christophe Denis, Romuald Elie, Mohamed Hebiri, and François Hu. Fairness guarantee in multi-class classification. *arXiv:2109.13642 [math, stat]*, September 2021. arXiv: 2109.13642.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214–226, New York, NY, USA, January 2012. Association for Computing Machinery.

[14] Giordano d'Aloisio, Andrea D'Angelo, Antinisca Di Marco, and Giovanni Stilo. Debiaser for Multiple Variables to enhance fairness in classification tasks. *Information Processing & Management*, 60(2):103226, March 2023.

[15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268, Sydney NSW Australia, August 2015. ACM.

[16] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im) possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.

[17] José A. Galindo, David Benavides, Pablo Trinidad, Antonio-Manuel Gutiérrez-Fernández, and Antonio Ruiz-Cortés. Automated analysis of feature models: Quo vadis? *Computing*, 101(5):387–433, May 2019.

[18] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[19] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, October 2012.

[20] Kyo C Kang, Sholom G Cohen, James A Hess, William E Novak, and A Spencer Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[21] Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, 132(1):1–63, 1997.

[22] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[23] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6):1–35, July 2021.

[24] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019.

[25] Kewen Peng, Joymallya Chakraborty, and Tim Menzies. FairMask: Better Fairness via Model-based Rebalancing of Protected Attributes. *IEEE Transactions on Software Engineering*, pages 1–14, 2022. Conference Name: IEEE Transactions on Software Engineering.

[26] Preston Putzel and Scott Lee. Blackbox Post-Processing for Multiclass Fairness. *arXiv:2201.04461 [cs]*, January 2022. arXiv: 2201.04461.

[27] Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*, 2:8, 2019.

[28] Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pages 1–7. IEEE, 2018.

[29] David H Wolpert. What does dinner cost?, 1999.

[30] Mengdi Zhang and Jun Sun. Adaptive fairness improvement based on causality analysis. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2022, pages 6–17, New York, NY, USA, November 2022. Association for Computing Machinery.