Contributions in VL-XAI to Address and Democratize Software Fairness

Giordano d'Aloisio

SoBigData@Univaq - 25 March, 2025

giordano.daloisio@univaq.it

Università degli Studi dell'Aquila / Italy


COMPAS

- COMPAS is an ML algorithm used by some courts in the US to predict recidivism of condemned people
- A study showed that, given two people with the same features but different ethnicity, the system was giving higher probability of recidivism to non-white people

Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica
May 23, 2016

ON A SPRING AFTERNOON IN 2014, Brisha Borden was running late to pick up her god-sister from school when she spotted an unlocked kid's blue Huffy bicycle and a silver Razor scooter. Borden and a friend grabbed the bike and scooter and tried to ride them down the street in the Fort Lauderdale suburb of Coral Springs.

Just as the 18-year-old girls were realizing they were too big for the tiny conveyances — which belonged to a 6-year-old boy — a woman came running after them saying, "That's my kid's stuff." Borden and her friend immediately dropped the bike and scooter and walked away.

But it was too late — a neighbor who witnessed the heist had already called the police. Borden and her friend were arrested and charged with burglary and petty theft for the items, which were valued at a total of $80.

COMPAS

- COMPAS is an ML algorithm used by some courts in the US to predict recidivism of condemned people
- A study showed that, given two people with the same features but different ethnicity, the system was giving higher probability of recidivism to non-white people

The system was biased against non-white people

Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica
May 23, 2016

ON A SPRING AFTERNOON IN 2014, Brisha Borden was running late to pick up her god-sister from school when she spotted an unlocked kid's blue Huffy bicycle and a silver Razor scooter. Borden and a friend grabbed the bike and scooter and tried to ride them down the street in the Fort Lauderdale suburb of Coral Springs.

Just as the 18-year-old girls were realizing they were too big for the tiny conveyances — which belonged to a 6-year-old boy — a woman came running after them saying, "That's my kid's stuff." Borden and her friend immediately dropped the bike and scooter and walked away.

But it was too late — a neighbor who witnessed the heist had already called the police. Borden and her friend were arrested and charged with burglary and petty theft for the items, which were valued at a total of $80.

Let's Define Bias and Fairness

- **BIAS:** systematic favoritism or discrimination in models' predictions towards individuals based on some sensitive features (like *gender*, *race*, and others)
- **FAIRNESS:** absence of favoritism or discrimination in models' predictions

N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, 'A Survey on Bias and Fairness in Machine Learning', ACM Comput. Surv., vol. 54, no. 6, pp. 1–35, Jul. 2021, doi: 10.1145/3457607.

Is the concept of bias that simple?

Is the concept of bias that simple?

(1) **Measurement Bias.** *Measurement, or reporting, bias arises from how we choose, utilize, and measure particular features* [140]. An example of this type of bias was observed in the recidivism risk prediction tool COMPAS, where prior arrests and friend/family arrests were used as proxy variables to measure level of "riskiness" or "crime"—which on its own can be viewed as mismeasured proxies. This is partly due to the fact that minority communities are controlled and policed more frequently, so they have higher arrest rates. However, one should not conclude that because people coming from minority groups have higher arrest rates, therefore they are more dangerous, as there is a difference in how these groups are assessed and controlled [140].

(2) **Omitted Variable Bias.** *Omitted variable bias*[4] *occurs when one or more important variables are left out of the model* [38, 110, 127]. An example for this case would be when someone designs a model to predict, with relatively high accuracy, the annual percentage rate at which customers will stop subscribing to a service, but soon observes that the majority of users are canceling their subscription without receiving any warning from the designed model. Now imagine that the reason for canceling the subscriptions is appearance of a new strong competitor in the market that offers the same solution, but for half the price. The appearance of the competitor was something that the model was not ready for; therefore, it is considered to be an omitted variable.

(3) **Representation Bias.** *Representation bias arises from how we sample from a population during data collection process* [140]. Non-representative samples lack the diversity of the population, with missing subgroups and other anomalies. Lack of geographical diversity in datasets like ImageNet (as shown in Figures 3 and 4) results in demonstrable bias towards Western cultures.

Is the concept of bias that simple?

(1) **Measurement Bias.** *Measurement, or reporting, bias arises from how we choose, utilize, and measure particular features* [140]. An example of this type of bias was observed in the recidivism risk prediction tool COMPAS, where prior arrests and friend/family arrests were used as proxy variables to measure level of "riskiness" or "crime"—which on its own can be

3.1.2 Algorithm to User. Algorithms modulate user behavior. Any biases in algorithms might introduce biases in user behavior. In this section, we talk about biases that are as a result of algorithmic outcomes and affect user behavior as a consequence.

(1) **Algorithmic Bias.** *Algorithmic bias is when the bias is not present in the input data and is added purely by the algorithm* [9]. The algorithmic design choices, such as use of certain optimization functions, regularizations, choices in applying regression models on the data as a whole or considering subgroups, and the general use of statistically biased estimators in algorithms [44], can all contribute to biased algorithmic decisions that can bias the outcome of the algorithms.

(2) **User Interaction Bias.** *User Interaction bias is a type of bias that can not only be observant on the Web but also get triggered from two sources—the user interface and through the user itself by imposing his/her self-selected biased behavior and interaction* [9]. This type of bias can be influenced by other types and subtypes, such as presentation and ranking biases.

(a) **Presentation Bias.** *Presentation bias is a result of how information is presented [9].* For example, on the Web users can only click on content that they see, so the seen content gets clicks, while everything else gets no click. And it could be the case that the user does not see all the information on the Web [9].

(b) **Ranking Bias.** *The idea that top-ranked results are the most relevant and important will result in attraction of more clicks than others.* This bias affects search engines [9] and crowdsourcing applications [92].

(3) **Popularity Bias.** *Items that are more popular tend to be exposed more. However, popularity metrics are subject to manipulation—for example, by fake reviews or social bots* [113]. As an

Is the concept of bias that simple?

(1) **Measurement Bi**
measure particular
cidivism risk predi
used as proxy vari

3.1.2 Algorithm to User.
introduce biases in user beha
rithmic outcomes and affect

(1) **Algorithmic Bias.** *Al*
added purely by the al
optimization functions
as a whole or consideri
algorithms [44], can all
of the algorithms.

(2) **User Interaction Bias**
the Web but also get tri
by imposing his/her sel
influenced by other typ

(a) **Presentation Bias**
example, on the We
gets clicks, while e
does not see all the

(b) **Ranking Bias.** *Th*
result in attraction of more clicks than others. This bias affects search engines [9] and crowdsourcing applications [92].

(3) **Popularity Bias.** *Items that are more popular tend to be exposed more. However, popularity metrics are subject to manipulation—for example, by fake reviews or social bots* [113]. As an

---

to the fact that only 5% of Fortune 500 CEOs were women—which would cause the search results to be biased towards male CEOs [140]. These search results were of course reflecting the reality, but whether or not the search algorithms should reflect this reality is an issue worth considering.

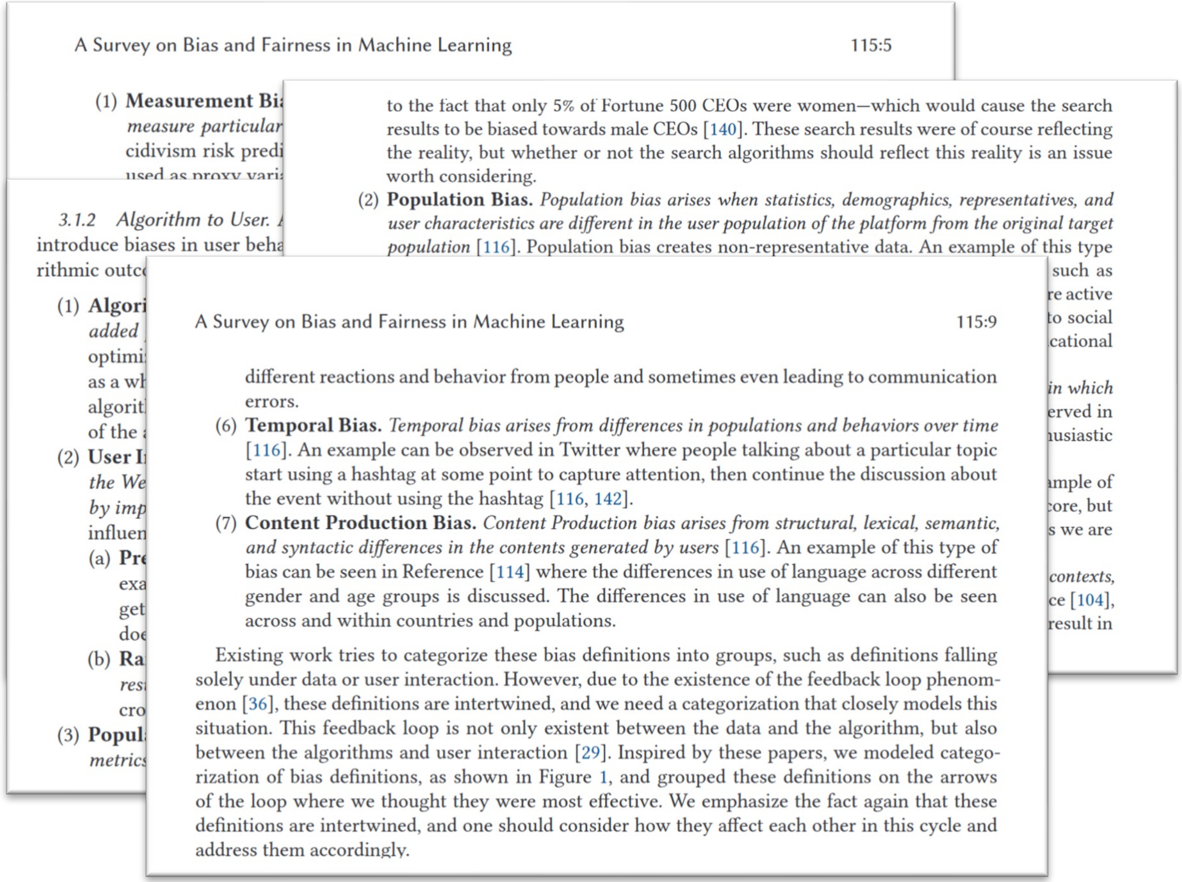
(2) **Population Bias.** *Population bias arises when statistics, demographics, representatives, and user characteristics are different in the user population of the platform from the original target population* [116]. Population bias creates non-representative data. An example of this type of bias can arise from different user demographics on different social platforms, such as women being more likely to use Pinterest, Facebook, Instagram, while men being more active in online forums like Reddit or Twitter. More such examples and statistics related to social media use among young adults according to gender, race, ethnicity, and parental educational background can be found in Reference [64].

(3) **Self-selection Bias.** *Self-selection bias*[4] *is a subtype of the selection or sampling bias in which subjects of the research select themselves.* An example of this type of bias can be observed in an opinion poll to measure enthusiasm for a political candidate, where the most enthusiastic supporters are more likely to complete the poll.

(4) **Social Bias.** *Social bias happens when others' actions affect our judgment* [9]. An example of this type of bias can be a case where we want to rate or review an item with a low score, but when influenced by other high ratings, we change our scoring thinking that perhaps we are being too harsh [9, 147].

(5) **Behavioral Bias.** *Behavioral bias arises from different user behavior across platforms, contexts, or different datasets* [116]. An example of this type of bias can be observed in Reference [104], where authors show how differences in emoji representations among platforms can result in

Is the concept of bias that simple?



A Survey on Bias and Fairness in Machine Learning 115:5

to the fact that only 5% of Fortune 500 CEOs were women—which would cause the search results to be biased towards male CEOs [140]. These search results were of course reflecting the reality, but whether or not the search algorithms should reflect this reality is an issue worth considering.

(2) **Population Bias.** *Population bias arises when statistics, demographics, representatives, and user characteristics are different in the user population of the platform from the original target population* [116]. Population bias creates non-representative data. An example of this type

A Survey on Bias and Fairness in Machine Learning 115:9

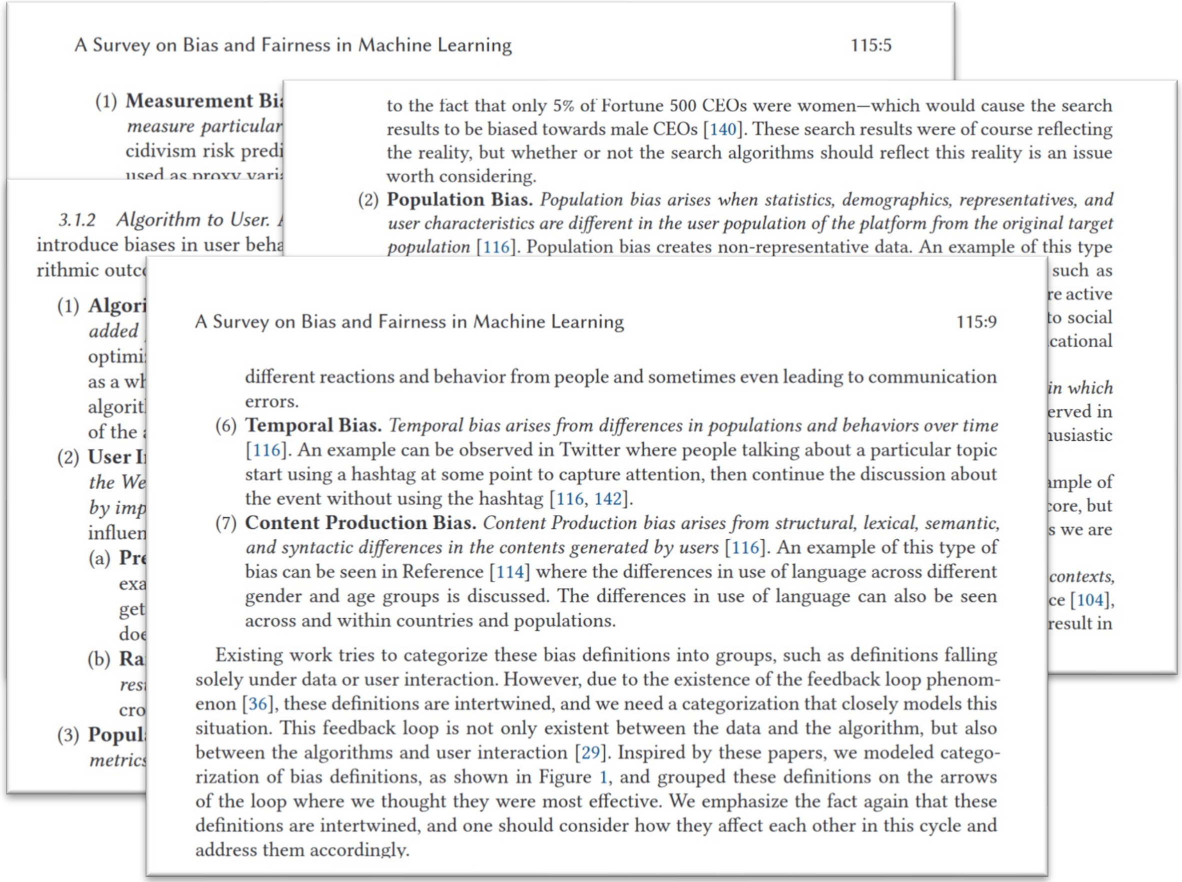
different reactions and behavior from people and sometimes even leading to communication errors.

(6) **Temporal Bias.** *Temporal bias arises from differences in populations and behaviors over time* [116]. An example can be observed in Twitter where people talking about a particular topic start using a hashtag at some point to capture attention, then continue the discussion about the event without using the hashtag [116, 142].

(7) **Content Production Bias.** *Content Production bias arises from structural, lexical, semantic, and syntactic differences in the contents generated by users* [116]. An example of this type of bias can be seen in Reference [114] where the differences in use of language across different gender and age groups is discussed. The differences in use of language can also be seen across and within countries and populations.

Existing work tries to categorize these bias definitions into groups, such as definitions falling solely under data or user interaction. However, due to the existence of the feedback loop phenomenon [36], these definitions are intertwined, and we need a categorization that closely models this situation. This feedback loop is not only existent between the data and the algorithm, but also between the algorithms and user interaction [29]. Inspired by these papers, we modeled categorization of bias definitions, as shown in Figure 1, and grouped these definitions on the arrows of the loop where we thought they were most effective. We emphasize the fact again that these definitions are intertwined, and one should consider how they affect each other in this cycle and address them accordingly.

Is the concept of bias that simple?



to the fact that only 5% of Fortune 500 CEOs were women—which would cause the search results to be biased towards male CEOs [140]. These search results were of course reflecting the reality, but whether or not the search algorithms should reflect this reality is an issue worth considering.

(2) **Population Bias.** *Population bias arises when statistics, demographics, representatives, and user characteristics are different in the user population of the platform from the original target population* [116]. Population bias creates non-representative data. An example of this type

A Survey on Bias and Fairness in Machine Learning 115:9

different reactions and behavior from people and sometimes even leading to communication errors.

(6) **Temporal Bias.** *Temporal bias arises from differences in populations and behaviors over time* [116]. An example can be observed in Twitter where people talking about a particular topic start using a hashtag at some point to capture attention, then continue the discussion about the event without using the hashtag [116, 142].

(7) **Content Production Bias.** *Content Production bias arises from structural, lexical, semantic, and syntactic differences in the contents generated by users* [116]. An example of this type of bias can be seen in Reference [114] where the differences in use of language across different gender and age groups is discussed. The differences in use of language can also be seen across and within countries and populations.

Existing work tries to categorize these bias definitions into groups, such as definitions falling solely under data or user interaction. However, due to the existence of the feedback loop phenomenon [36], these definitions are intertwined, and we need a categorization that closely models this situation. This feedback loop is not only existent between the data and the algorithm, but also between the algorithms and user interaction [29]. Inspired by these papers, we modeled categorization of bias definitions, as shown in Figure 1, and grouped these definitions on the arrows of the loop where we thought they were most effective. We emphasize the fact again that these definitions are intertwined, and one should consider how they affect each other in this cycle and address them accordingly.

At least 23 different definitions of bias in the literature

From many definitions come many metrics...

From many definitions come many metrics...

Generic metrics

`metrics.num_samples` (y_true[, y_pred, ...])	Compute the number of samples.
`metrics.num_pos_neg` (y_true[, y_pred, ...])	Compute the number of positive and negative samples.
`metrics.specificity_score` (y_true, y_pred, *)	Compute the specificity or true negative rate.
`metrics.sensitivity_score` (y_true, y_pred[, ...])	Alias of `sklearn.metrics.recall_score()` for binary classes only.
`metrics.base_rate` (y_true[, y_pred, ...])	Compute the base rate, $Pr(Y = \text{pos\_label}) = \frac{P}{P+N}$.
`metrics.selection_rate` (y_true, y_pred, *[, ...])	Compute the selection rate, $Pr(\hat{Y} = \text{pos\_label}) = \frac{TP+FP}{P+N}$.
`metrics.smoothed_base_rate` (y_true[, y_pred, ...])	Compute the smoothed base rate, $\frac{P+\alpha}{P+N+\|R_Y\|\alpha}$.
`metrics.smoothed_selection_rate` (y_true, ...)	Compute the smoothed selection rate, $\frac{TP+FP+\alpha}{P+N+\|R_Y\|\alpha}$.
`metrics.generalized_fpr` (y_true, probas_pred, *)	Return the ratio of generalized false positives to negative examples in the dataset, $GFPR = \frac{GFP}{N}$.
`metrics.generalized_fnr` (y_true, probas_pred, *)	Return the ratio of generalized false negatives to positive examples in the dataset, $GFNR = \frac{GFN}{P}$.

From many definitions come many metrics…

Generic metrics

`metrics.num_samples` (y_true[, y_pred, …])	Compute the number of samples.
`metrics.num_pos_neg` (y_true[, y_pred, …])	Compute the number of positive and negative samples.
`metrics.specificity_score` (y_true, y_pred, *)	Compute the specificity or true negative rate.
`metrics.sensitivity_score` (y_true, y_pred[, …])	Alias of `sklearn.metrics.recall_score()` for binary classes only.
`metrics.base_rate` (y_true[, y_pred, …])	Compute the base rate, $Pr(Y = \text{pos\_label}) = \frac{P}{P+N}$.
`metrics.selection_rate` (y_true, y_pred, *[, …])	Compute the selection rate, $Pr(\hat{Y} = \text{pos\_label}) = \frac{TP+FP}{P+N}$.
`metrics.smoothed_base_rate` (y_true[, y_pred, …])	Compute the smoothed base rate, $\frac{P+\alpha}{P+N+\lvert R_Y \rvert \alpha}$.
`metrics.smoothed_selection_rate` (y_true, …)	Compute the smoothed selection rate, $\frac{TP+FP+\alpha}{P+N+\lvert R_Y \rvert \alpha}$.
`metrics.generalized_fpr` (y_true, probas_pred, *)	Return the ratio of generalized false positives to negative examples in the dataset, $GFPR = \frac{GFP}{N}$.
`metrics.generalized_fnr` (y_true, probas_pred, *)	Return the ratio of generalized false negatives to positive examples in the dataset, $GFNR = \frac{GFN}{\dots}$

Individual fairness metrics

`metrics.generalized_entropy_index` (b[, alpha])	Generalized entropy index measures inequality over a population.
`metrics.generalized_entropy_error` (y_true, y_pred)	Compute the generalized entropy.
`metrics.theil_index` (b)	The Theil index is the `generalized_entropy_index()` with $\alpha = 1$.
`metrics.coefficient_of_variation` (b)	The coefficient of variation is the square root of two times the `generalized_entropy_index()` with $\alpha = 2$.
`metrics.consistency_score` (X, y[, n_neighbors])	Compute the consistency score.

From many definitions come many metrics...

Generic metrics

`metrics.num_samples` (y_true[, y_pred, ...])	Compute the number of samples.
`metrics.num_pos_neg` (y_true[, y_pred, ...])	Compute the number of positive and negative samples
`metrics.specificity_score` (y_true, y_pred, *)	Compute the specificity or true negative rate.
`metrics.sensitivity_score` (y_true, y_pred[, ...])	Alias of `sklearn.metrics.recall_score()` for binary classe
`metrics.base_rate` (y_true[, y_pred, ...])	Compute the base rate, $Pr(Y = \text{pos\_label}) = \frac{P}{P+N}$
`metrics.selection_rate` (y_true, y_pred, *[, ...])	Compute the selection rate, $Pr(\hat{Y} = \text{pos\_label}) =$
`metrics.smoothed_base_rate` (y_true[, y_pred, ...])	Compute the smoothed base rate, $\frac{P+\alpha}{P+N+\|R_Y\|\alpha}$.
`metrics.smoothed_selection_rate` (y_true, ...)	Compute the smoothed selection rate, $\frac{TP+FP+\alpha}{P+N+\|R_Y\|\alpha}$.
`metrics.generalized_fpr` (y_true, probas_pred, *)	Return the ratio of generalized false positives to negati the dataset, $GFPR = \frac{GFP}{N}$.
`metrics.generalized_fnr` (y_true, probas_pred, *)	Return the ratio of generalized false negatives to positi the dataset, $GFNR = \frac{GFN}{}$

Individual fairness metrics

`metrics.generalized_entropy_index` (b[, alpha])	Generalized entropy index measures inequality over a
`metrics.generalized_entropy_error` (y_true, y_pred)	Compute the generalized entropy.
`metrics.theil_index` (b)	The Theil index is the `generalized_entropy_index()` with
`metrics.coefficient_of_variation` (b)	The coefficient of variation is the square root of two ti `generalized_entropy_index()` with $\alpha = 2$.
`metrics.consistency_score` (X, y[, n_neighbors])	Compute the consistency score.

Group fairness metrics

`metrics.statistical_parity_difference` (y_true)	Difference in selection rates.
`metrics.mean_difference` (y_true[, y_pred, ...])	Alias of `statistical_parity_difference()`.
`metrics.disparate_impact_ratio` (y_true[, ...])	Ratio of selection rates.
`metrics.equal_opportunity_difference` (y_true, ...)	A relaxed version of equality of opportunity.
`metrics.average_odds_difference` (y_true, ...)	A relaxed version of equality of odds.
`metrics.average_odds_error` (y_true, y_pred, *)	A relaxed version of equality of odds.
`metrics.class_imbalance` (y_true[, y_pred, ...])	Compute the class imbalance, $\frac{N_u-N_p}{N_u+N_p}$.
`metrics.kl_divergence` (y_true[, y_pred, ...])	Compute the Kullback-Leibler divergence, $KL(P_p\|\|P_u) = \sum_y P_p(y) \log\left(\frac{P_p(y)}{P_u(y)}\right)$
`metrics.conditional_demographic_disparity` (y_true)	Conditional demographic disparity, $CDD = \frac{1}{\sum_i N_i} \sum_i N_i \cdot DD_i$
`metrics.smoothed_edf` (y_true[, y_pred, ...])	Smoothed empirical differential fairness (EDF).
`metrics.df_bias_amplification` (y_true, y_pred, *)	Differential fairness bias amplification.
`metrics.between_group_generalized_entropy_error` (...)	Compute the between-group generalized entropy.
`metrics.mdss_bias_scan` (y_true, probas_pred)	DEPRECATED: Change to new interface - aif360.sklearn.detectors.mdss_detector.bias_scan by version 0.5.0.
`metrics.mdss_bias_score` (y_true, probas_pred)	Compute the bias score for a prespecified group of records using a given scoring function.

From many definitions come many metrics...

Generic metrics

`metrics.num_samples` (y_true[, y_pred, ...])	Compute the number of samples.
`metrics.num_pos_neg` (y_true[, y_pred, ...])	Compute the number of positive and negative samples
`metrics.specificity_score` (y_true, y_pred, *)	Compute the specificity or true negative rate.
`metrics.sensitivity_score` (y_true, y_pred[, ...])	Alias of `sklearn.metrics.recall_score()` for binary classe
`metrics.base_rate` (y_true[, y_pred, ...])	Compute the base rate, $Pr(Y = \text{pos\_label}) = \frac{P}{P+N}$
`metrics.selection_rate` (y_true, y_pred, *[, ...])	Compute the selection rate, $Pr(\hat{Y} = \text{pos\_label}) =$
`metrics.smoothed_base_rate` (y_true[, y_pred, ...])	Con
`metrics.smoothed_selection_rate` (y_true, ...)	Con
`metrics.generalized_fpr` (y_true, probas_pred, *)	Retu the dataset, $GFPR = \frac{GFP}{N}$.
`metrics.generalized_fnr` (y_true, probas_pred, *)	Return the ratio of generalized false negatives to positi

Individual fairness metrics

`metrics.generalized_entropy_index` (b[, alpha])	Generalized entropy index measures inequality over a
`metrics.generalized_entropy_error` (y_true, y_pred)	Compute the generalized entropy.
`metrics.theil_index` (b)	The Theil index is the `generalized_entropy_index()` with
`metrics.coefficient_of_variation` (b)	The coefficient of variation is the square root of two ti `generalized_entropy_index()` with $\alpha = 2$.
`metrics.consistency_score` (X, y[, n_neighbors])	Compute the consistency score.

Group fairness metrics

`metrics.statistical_parity_difference` (y_true)	Difference in selection rates.
`metrics.mean_difference` (y_true[, y_pred, ...])	Alias of `statistical_parity_difference()`.
`metrics.disparate_impact_ratio` (y_true[, ...])	Ratio of selection rates.
`metrics.equal_opportunity_difference` (y_true, ...)	A relaxed version of equality of opportunity.
`metrics.average_odds_difference` (y_true, ...)	A relaxed version of equality of odds.
	A relaxed version of equality of odds.
	Compute the class imbalance, $\frac{N_u-N_p}{N_u+N_p}$.
	Compute the Kullback-Leibler divergence, $KL(P_p\|\|P_u) = \sum_y P_p(y) \log\left(\frac{P_p(y)}{P_u(y)}\right)$
`metrics.conditional_demographic_disparity` (y_true)	Conditional demographic disparity, $CDD = \frac{1}{\sum_i N_i} \sum_i N_i \cdot DD_i$
`metrics.smoothed_edf` (y_true[, y_pred, ...])	Smoothed empirical differential fairness (EDF).
`metrics.df_bias_amplification` (y_true, y_pred, *)	Differential fairness bias amplification.
`metrics.between_group_generalized_entropy_error` (...)	Compute the between-group generalized entropy.
`metrics.mdss_bias_scan` (y_true, probas_pred)	DEPRECATED: Change to new interface - aif360.sklearn.detectors.mdss_detector.bias_scan by version 0.5.0.
`metrics.mdss_bias_score` (y_true, probas_pred)	Compute the bias score for a prespecified group of records using a given scoring function.

At least 29 different metrics available in the AIF360 library

Mitigating Bias

Mitigating Bias

`aif360.algorithms.preprocessing`

`algorithms.preprocessing.DisparateImpactRemover` ([...])	Disparate impact remover is a preprocessing technique that edits feature values increase group fairness while preserving rank-ordering within groups [1]_.
`algorithms.preprocessing.LFR` (...[, k, Ax, ...])	Learning fair representations is a pre-processing technique that finds a latent representation which encodes the data well but obfuscates information about protected attributes [2]_.
`algorithms.preprocessing.OptimPreproc` (...[, ...])	Optimized preprocessing is a preprocessing technique that learns a probabilistic transformation that edits the features and labels in the data with group fairness, individual distortion, and data fidelity constraints and objectives [3]_.
`algorithms.preprocessing.Reweighing` (...)	Reweighing is a preprocessing technique that Weights the examples in each (group, label) combination differently to ensure fairness before classification [4]_.

Mitigating Bias

`aif360.algorithms.preprocessing`

`algorithms.preprocessing.DisparateImpactRemover` ([...])	Disparate impact remover is feature values increase grou ordering within groups [1]_.
`algorithms.preprocessing.LFR` (...[, k, Ax, ...])	Learning fair representations finds a latent representation obfuscates information abou
`algorithms.preprocessing.OptimPreproc` (...[, ...])	Optimized preprocessing is a a probabilistic transformatio the data with group fairness, fidelity constraints and obje
`algorithms.preprocessing.Reweighing` (...)	Reweighing is a preprocessin examples in each (group, lab fairness before classification

`aif360.algorithms.inprocessing`

`algorithms.inprocessing.AdversarialDebiasing` (...)	Adversarial debiasing is an in-processing technique that learns a classifier to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions [5]_.
`algorithms.inprocessing.ARTClassifier` (...)	Wraps an instance of an `art.classifiers.Classifier` to extend `Transformer`.
`algorithms.inprocessing.GerryFairClassifier` ([...])	Model is an algorithm for learning classifiers that are fair with respect to rich subgroups.
`algorithms.inprocessing.MetaFairClassifier` ([...])	The meta algorithm here takes the fairness metric as part of the input and returns a classifier optimized w.r.t.
`algorithms.inprocessing.PrejudiceRemover` ([...])	Prejudice remover is an in-processing technique that adds a discrimination-aware regularization term to the learning objective [6]_.
`algorithms.inprocessing.ExponentiatedGradientReduction` (...)	Exponentiated gradient reduction for fair classification.
`algorithms.inprocessing.GridSearchReduction` (...)	Grid search reduction for fair classification or regression.

Mitigating Bias

aif360.algorithms.preprocessing

`algorithms.preprocessing.DisparateImpactRemover` ([...])	Disparate impact remover is feature values increase group ordering within groups [1]_.
`algorithms.preprocessing.LFR` (...[, k, Ax, ...])	Learning fair representations finds a latent representation obfuscates information abou
`algorithms.preprocessing.OptimPreproc` (...[, ...])	Optimized preprocessing is a a probabilistic transformatio the data with group fairness, fidelity constraints and objec

aif360.algorithms.inprocessing

`algorithms.inprocessing.AdversarialDebiasing` (...)	Adversarial debiasing is an in-processing technique that learns a classifier to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions [5]_.
`algorithms.inprocessing.ARTClassifier` (...)	Wraps an instance of an `art.classifiers.Classifier` to extend `Transformer`.
`algorithms.inprocessing.GerryFairClassifier` ([...])	Model is an algorithm for learning classifiers that are fair with respect to rich subgroups.
`ifier` ([...])	The meta algorithm here takes the fairness metric as part of the input and returns a classifier optimized w.r.t.
`ver` ([...])	Prejudice remover is an in-processing technique that adds a discrimination-aware regularization term to the learning objective [6]_.
`GradientReduction` (...)	Exponentiated gradient reduction for fair classification.
`uction` (...)	Grid search reduction for fair classification or regression.

aif360.algorithms.postprocessing

`algorithms.postprocessing.CalibratedEqOddsPostprocessing` (...)	Calibrated equalized odds postprocessing is a post-processing technique that optimizes over calibrated classifier score outputs to find probabilities with which to change output labels with an equalized odds objective [7]_.
`algorithms.postprocessing.EqOddsPostprocessing` (...)	Equalized odds postprocessing is a post-processing technique that solves a linear program to find probabilities with which to change output labels to optimize equalized odds [8]_ [9]_.
`algorithms.postprocessing.RejectOptionClassification` (...)	Reject option classification is a postprocessing technique that gives favorable outcomes to unpriviliged groups and unfavorable outcomes to priviliged groups in a confidence band around the decision boundary with the highest uncertainty [10]_.

Mitigating Bias

`aif360.algorithms.preprocessing`

`algorithms.preprocessing.DisparateImpactRemover` ([...])	Disparate impact remover is ... feature values increase group ... ordering within groups [1]_.
`algorithms.preprocessing.LFR` (...[, k, Ax, ...])	Learning fair representations ... finds a latent representation ... obfuscates information abou...
`algorithms.preprocessing.OptimPreproc` (...[, ...])	Optimized preprocessing is a ... a probabilistic transformation ...

`aif360.algorithms.inprocessing`

`algorithms.inprocessing.AdversarialDebiasing` (...)	Adversarial debiasing is an in-processing technique that learns a classifier to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions [5]_.
`algorithms.inprocessing.ARTClassifier` (...)	Wraps an instance of an `art.classifiers.Classifier` to extend `Transformer`.
	...orithm for learning classifiers that are fair ... rich subgroups.
	...thm here takes the fairness metric as part of ... eturns a classifier optimized w.r.t.
...`ver` ([...])	...ver is an in-processing technique that adds a discrimination-aware regularization term to the learning objective [6]_.
...`GradientReduction` (...)	Exponentiated gradient reduction for fair classification.
...`uction` (...)	Grid search reduction for fair classification or regression.

`aif360.algorithms.postprocessing`

`algorithms.postprocessing.CalibratedEqOddsPostprocessing` (...)	... processing technique that optimizes over calibrated classifier score outputs to find probabilities with which to change output labels with an equalized odds objective [7]_.
`algorithms.postprocessing.EqOddsPostprocessing` (...)	Equalized odds postprocessing is a post-processing technique that solves a linear program to find probabilities with which to change output labels to optimize equalized odds [8]_ [9]_.
`algorithms.postprocessing.RejectOptionClassification` (...)	Reject option classification is a postprocessing technique that gives favorable outcomes to unpriviliged groups and unfavorable outcomes to priviliged groups in a confidence band around the decision boundary with the highest uncertainty [10]_.

14 bias mitigation methods are available in the AIF360 repository... but many more are available from the literature!

Our Contributions

Our Contributions

Challenge 1 (CH1)

Developing approaches for bias mitigation both in binary and multi-class classification settings.

Challenge 2 (CH2)

Democratizing the development of fair learning-based systems to actors with different expertise.

Challenge 1: Bias in Multi-Class Classification

- Most of the bias mitigation approaches focus on binary classification
- However, many multi-class classification approaches have been proposed in sensitive domains

Computing, Artificial Intelligence and Information Technology

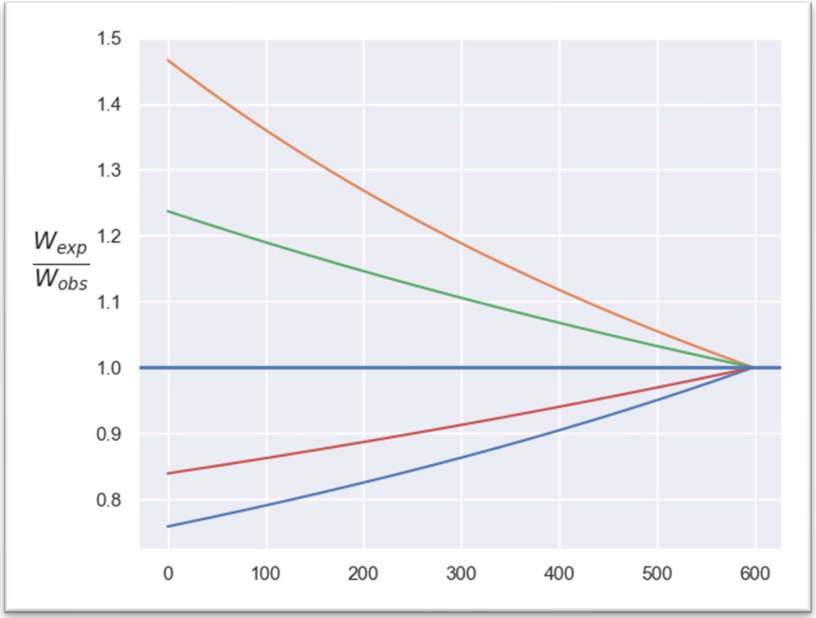
A data-driven software tool for enabling cooperative information sharing among police departments

Will I Pass the Bar Exam: Predicting Student Success Using LSAT Scores and Law School Performance

Nuclear feature extraction for breast tumor diagnosis

Contribution 1: Debiaser for Multiple Variables

- DEMV is a pre-processing approach to improve fairness in binary and multi-class classification tasks
- Overcomes all the other state-of-the-art multi-class bias mitigation algorithms in the literature
- Algorithm available on SoBigData RI and PIP



Challenge 2: Democratising Software Fairness

Challenge 2: Democratising Software Fairness

23 Definitions of Bias

Challenge 2: Democratising Software Fairness

23 Definitions of Bias

+

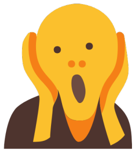
29 Different Metric

Challenge 2: Democratising Software Fairness

23 Definitions of Bias + 29 Different Metric + 14 Different Methods

Challenge 2: Democratising Software Fairness

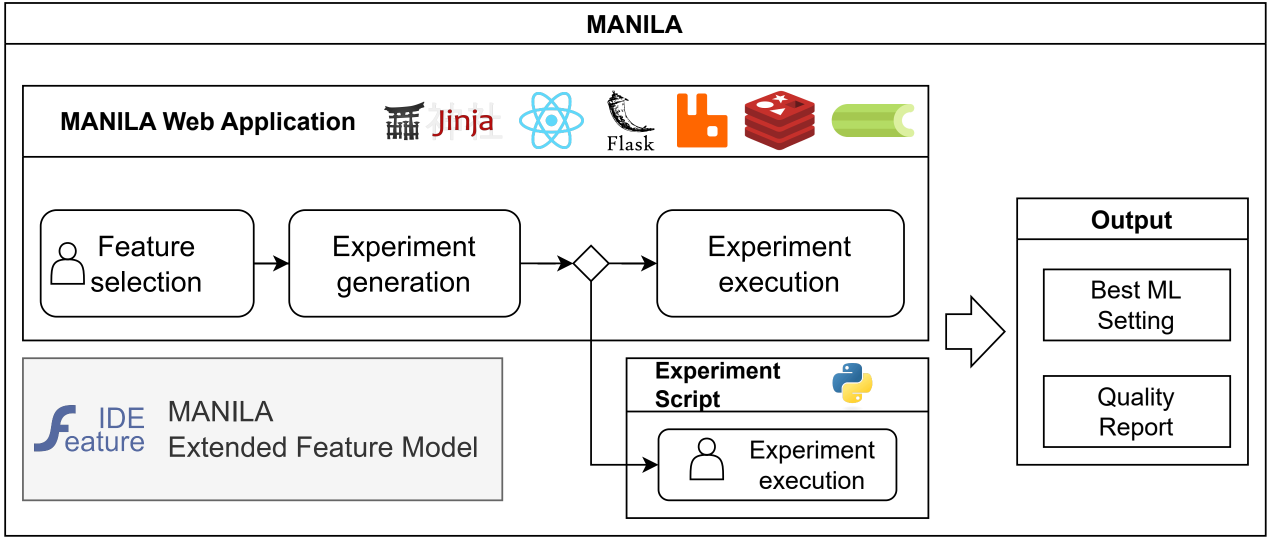
23 Definitions of Bias + 29 Different Metric + 14 Different Methods



Data Scientist less-expert on fairness

Contribution 2: MANILA

- We propose MANILA, a web-based application to design, implement and execute fairness evaluations
- Uses the Extended Feature Model (ExtFM) formalism to model the evaluation workflow as a Software Product Line



Available in SoBigData RI

Thank you for your attention!

UNIVERSITÀ DEGLI STUDI DELL'AQUILA
DISIM
Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica