An Empirical Study of Modular Bias Mitigators and Ensembles

Michael Feffer [1] **Martin Hirzel** [2] **Samuel C. Hoffman** [2] **Kiran Kate** [2] **Parikshit Ram** [2] **Avraham Shinnar** [2]

Abstract

Bias mitigators can reduce algorithmic bias in machine learning models, but their effect on fairness is often not stable across different data splits. A popular approach to train more stable models is ensemble learning. We built an open-source library enabling the modular composition of 10 mitigators, 4 ensembles, and their corresponding hyperparameters. We empirically explored the space of combinations on 13 datasets and distilled the results into a guidance diagram for practitioners.

1. Introduction

Algorithmic bias in machine learning can lead to models that discriminate against underprivileged groups in various domains, including hiring, healthcare, finance, criminal justice, education, and even child care. Of course, bias in machine learning is a sociotechnical problem that cannot be solved with technical solutions alone. That said, to make tangible progress, this paper focuses on *bias mitigators*, which improve or replace an existing machine learning estimator (e.g., a classifier) so it makes less biased predictions (e.g., class labels) as measured by a fairness metric (e.g., disparate impact [8]). Unfortunately, bias mitigation often suffers from high *volatility*, meaning the estimator is less stable with respect to group fairness metrics. In the worst case, this volatility can even cause a model to appear fair when measured on training data while being unfair on production data. Given that ensembles (e.g., bagging or boosting) can improve stability for accuracy metrics [20], we felt it was important to explore whether they also improve stability for group fairness metrics.

Prior work either explores bias mitigation without any consideration of ensembles, or entangles the two [3, 10, 13, 16, 17]. In contrast, our paper advocates that bias mitigators and ensembles can be modular building blocks. Modularity provides a larger space of possible combinations to explore and helps extend future advances in either ensembling or bias mitigation to their combination. This paper explores the question, "*Can modular ensembles help with fairness, and if yes, how?*" We conducted a comprehensive empirical study with 10 bias mitigators from AIF360 [2]; bagging, boosting, voting, and stacking ensembles from the popular scikit-learn library [4]; and 13 datasets of varying baseline fairness (earlier papers use at most a handful). Our findings confirm the intuition that ensembles often improve stability of not just accuracy but also the group fairness metrics we explored. However, the best configuration of mitigator and ensemble depends on dataset characteristics, evaluation metric of choice, and even worldview [9]. Therefore, we automatically distilled a method selection guidance diagram.

To support these experiments, we assembled a library of pluggable ensembles, bias mitigators, and fairness datasets. While we reused popular and well-established open-source technologies, we made several new adaptations in our library to get components to work well together. Our library is available open-source (`https://github.com/IBM/lale`) to encourage research and real-world adoption.

2. Library and Datasets

One of the contributions of our work is to implement compatibility between fairness mitigators and metrics from AIF360 [2] and ensembles from scikit-learn [4] within a single library, Lale [1], for exploring their combinations.

Metrics. This paper uses metrics from scikit-learn, including precision, recall, and F_1 score (harmonic mean of precision and recall). In addition, we implemented a scikit-learn compatible API for several fairness metrics from AIF360 including disparate impact (ratio of positive outcomes for unprivileged group to positive outcomes for privileged group, as described in [8]).

Ensembles. Ensemble learning uses multiple weak models to form one strong model. We use four ensembles supported by scikit-learn and Lale in our experiments: bagging, boosting, voting, and stacking. Following scikit-learn, we use the following terminology to characterize ensembles: A *base estimator* is an estimator that serves as a building block for the ensemble. An ensemble supports one of two *composition* types: whether the ensemble consists of identical

[1]Institute of Software Research, Carnegie Mellon University, Pittsburgh, PA, USA [2]IBM Research, Yorktown Heights, NY, USA. Correspondence to: Michael Feffer <mfeffer@andrew.cmu.edu>, Martin Hirzel <hirzel@us.ibm.com>.

ICML Workshop on Benchmarking Data for Data-Centric AI (DataPerf@ICML), Baltimore, Maryland, USA, 2022.

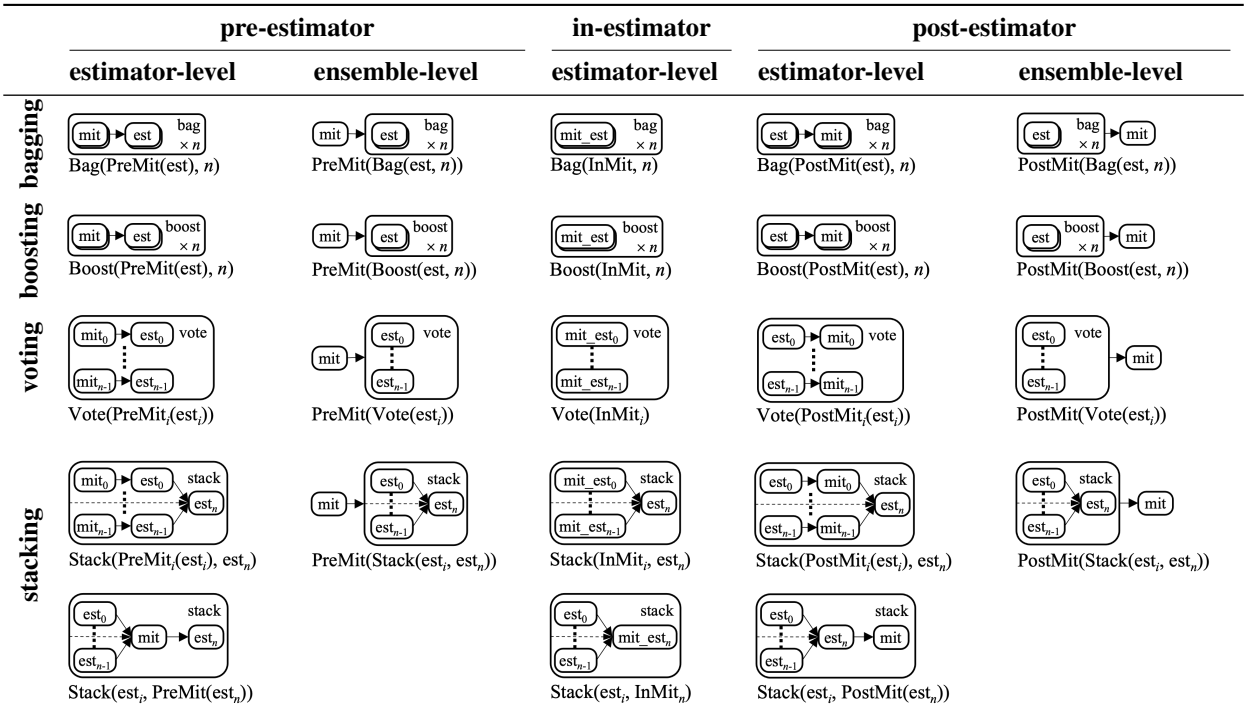

Figure 1. Combinations of ensembles and mitigators. PreMit(est) applies a pre-estimator mitigator before an estimator est; InMit denotes an in-estimator mitigator, which is itself an estimator; and PostMit(est) applies a post-estimator mitigator after an estimator est. Bag(est, n) is short for BaggingClassifier with n instances of base estimator est; Boost(est, n) is short for AdaBoostClassifier with n instances of base estimator est; Vote(est_i) applies a VotingClassifier to a list of base estimators est_i; and Stack(est_i, est_n) applies a StackingClassifier to a list of base estimators est_i and a final estimator est_n. For stacking, the passthrough option is represented by a dashed horizontal arrow.

base estimators (*homogeneous*, e.g. bagging and boosting) or can consist of different ones (*heterogeneous*, e.g. voting and stacking). For the homogeneous ensembles, we used their most common base estimator in practice: the decision-tree classifier. For the heterogeneous ensembles (voting and stacking), we used a set of typical base estimators: XGBoost [6], random forest, k-nearest neighbors, and support vector machines. Finally, for stacking, we also used XGBoost as the final estimator.

Mitigators. We added support in Lale for bias mitigation from AIF360 [2]. AIF360 distinguishes three kinds of mitigators for improving group fairness: *pre-estimator mitigators*, which are learned input manipulations that reduce bias in the data sent to downstream estimators (we used DisparateImpactRemover [8], LFR [21], and Reweighing [12]); *in-estimator mitigators*, which are specialized estimators that directly incorporate debiasing into their training (AdversarialDebiasing [22], GerryFairClassifier [15], MetaFairClassifier [5], and PrejudiceRemover [14]); and *post-estimator mitigators*, which reduce bias in predictions made by an upstream estimator (we used CalibratedEqOddsPostprocessing [18]).

Fig. 1 visualizes the combinations of ensemble types and mitigator kinds we explored, while also highlighting the modularity of our approach. Mitigation strategies can be applied at the level of either the base estimator or the entire ensemble, but by the fundamental nature of some ensembles and mitigators, not all combinations are feasible. First, post-estimator mitigators typically do not support `predict_proba` functionality required for some ensemble methods and recommended for others. Calibrating probabilities from post-estimator mitigators has been shown to be tricky [18], so despite Lale support for other post-estimator mitigators, CalibratedEqOddsPostprocessing is the only one explored in our experiments. Additionally, it is impossible to apply an in-estimator mitigator at the ensemble level, so we exclude those combinations. Finally, we decided to omit some combinations that are technically feasible but less interesting. For example, while our library supports mitigation at multiple points, say, at both the ensemble and estimator level of bagging, we elided these configuration from Fig. 1 and from our experiments.

Datasets. We gathered the datasets for our experiments from OpenML [19]. Some have been used extensively as benchmarks in other parts of the algorithmic fairness literature. We pulled other novel datasets from OpenML that have demographic data that could be considered protected attributes (such as race, age, or gender) and contained asso-

Dataset	Description	Privileged group(s)	N_{rows}	N_{cols}	DI
COMPAS Violent	Correctional offender violent recidivism	White women	3,377	10	0.822
Credit-g	German bank data quantifying credit risk	Men and older people	1,000	58	0.748
COMPAS	Correctional offender recidivism	White women	5,278	10	0.687
Ricci	Fire department promotion exam results	White men	118	6	0.498
TAE	University teaching assistant evaluation	Native English speakers	151	6	0.449
Titanic	Survivorship of Titanic passengers	Women and children	1,309	37	0.263
SpeedDating	Speed dating experiment at business school	Same race	8,378	70	0.853
Bank	Portuguese bank subscription predictions	Older people	45,211	51	0.840
MEPS 19	Utilization results from Panel 19 of MEPS	White individuals	15,830	138	0.490
MEPS 20	Same as MEPS 19 except for Panel 20	White individuals	17,570	138	0.488
Nursery	Slovenian nursery school application results	"Pretentious parents"	12,960	25	0.461
MEPS 21	Same as MEPS 19 except for Panel 21	White individuals	15,675	138	0.451
Adult	1994 US Census salary data	White men	48,842	100	0.277

Table 1. Qualitative and quantitative summary information of the datasets. The datasets are ordered by first partitioning by whether they contain at least 8,000 rows (we picked 8,000 to get a roughly even split; the partition is represented by the horizontal line in the middle of the table) and then sorting by descending baseline disparate impact (DI). Values for the number of rows (N_{rows}), number of columns (N_{cols}), and baseline disparate impact displayed here are computed *after* preprocessing techniques are applied.

ciated baseline levels of disparate impact. In all, we used 13 datasets, summarized in Table 1. When running experiments, we split the datasets using stratification by not just the target labels but also the protected attributes [11], leading to moderately more homogeneous fairness results across different splits. The exact details of the preprocessing are in the open-source code for our library for reproducibility. We hope that bundling these datasets and default preprocessing with our package, in addition to AIF360 and scikit-learn compatibility, will improve dataset quality going forward.

3. Empirical Study

We organize our experiments into two steps. The first is a preliminary search that finds "best" mitigators without ensembles. The second is the ensemble experiments using the mitigator configurations selected by the first.

First step. It is difficult to define "best" (in an empirical sense) given different dimensions of performance and datasets. To this end, we first run grid searches over each dataset, exploring mitigators and their hyperparameters with basic decision-trees where needed. We run 5 trials of 3-fold cross validation for each configuration. For each dataset, we choose a "best" pre-, in-, and post-estimator mitigator:

1. Filter configurations to ones with acceptable fairness, defined as mean disparate impact between 0.8 and 1.25.
2. Filter remaining to ones with nontrivial precision.
3. Filter remaining to ones with good predictive performance, defined as mean F_1 score (across 5 trials) greater than the average of all mean F_1 scores or the median of all mean F_1 scores, whichever is greater.
4. Finally, select the mitigator with maximum precision (in case of COMPAS, since true positives should be prioritized) or recall (all other datasets, since false negatives should be avoided).

Second step. Given the "best" mitigator configurations, this step explores the Cartesian product of ensembles and mitigators of Fig. 1 plus ensemble hyperparameters. For bagging and boosting, the only ensemble-level hyperparameter varied between configurations was the number of base estimators: $\{1, 10, 100\}$ for bagging and $\{1, 50, 500\}$ for boosting. Voting and stacking use lists of heterogeneous base estimators as hyperparameters. In our experiments, these lists contained either 4 mitigated or 4 unmitigated base estimators. For the in-estimator mitigation case these were {PrejudiceRemover, GerryFairClassifier, MetaFairClassifier, and AdversarialDebiasing}. Lastly, stacking also has a `passthrough` hyperparameter controlling whether dataset features were passed to the final estimator. If `passthrough` is set to `False`, it is impossible to mitigate the final estimator due to lack of dataset features; otherwise we mitigate either the base estimators or final estimator, but not both. The second step also uses 5 trials of 3-fold cross validation for each experiment, running on a computing cluster with Intel Xeon E5-2667 processors @ 3.30GHz. Every experiment configuration run was allotted 4 cores and 12 GB memory.

Result preprocessing. To facilitate cross-dataset comparisons, we applied the following procedure on a per-dataset basis for each metric of interest: (i) given all results, map all values to the same region of metric space around the point of optimal fairness (i.e. for disparate impact, we use the reciprocal of a value if it is larger than 1 for downstream calculations, and for statistical parity difference, we use the absolute value), and (ii) min-max scale the mean and standard deviation of the metric of interest, separately. After doing this for all datasets, we group remaining results by mitigator kind and ensemble type, and average the scaled values over all datasets for each group. Given a metric M, we refer to the result of this procedure using mean values as "standardized M outcome" and using standard deviation

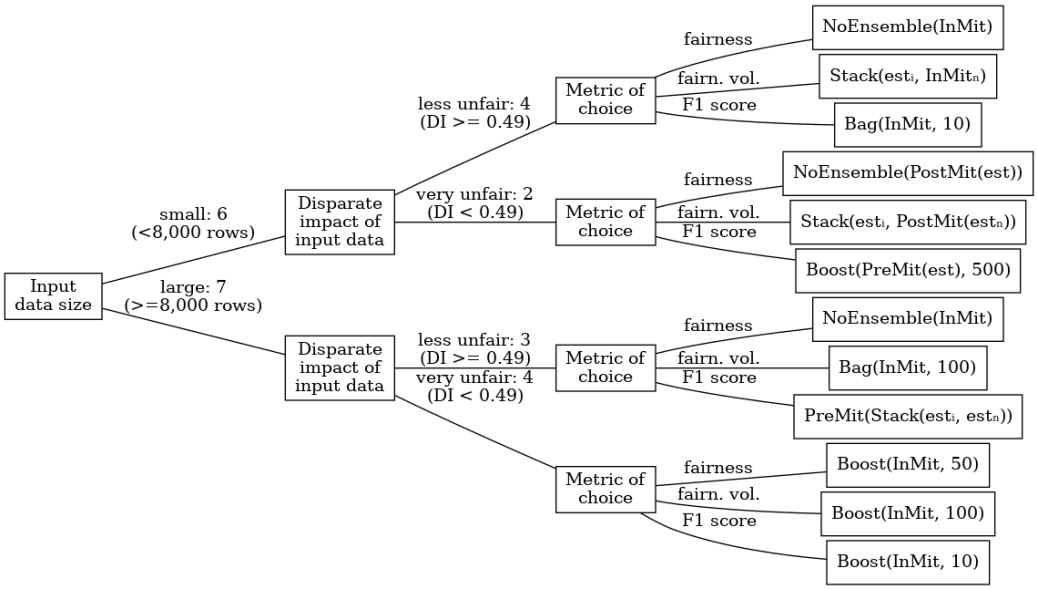

Figure 2. Guidance diagram for picking a good starting configuration given dataset characteristics and a target metric.

	No Mit.		Pre-		In-		Post-	
	DO	DV	DO	DV	DO	DV	DO	DV
No ensemble	**0.49**	0.05	0.84	**0.10**	**0.87**	0.26	**0.62**	0.04
Bagging	0.35	**0.02**	0.86	0.29	0.75	0.37	0.43	0.04
Boosting	0.34	**0.02**	**0.94**	0.57	0.86	0.19	0.44	0.03
Voting	0.30	0.05	0.73	0.80	0.46	**0.08**	0.00	**0.00**
Stacking	0.34	0.04	0.66	0.14	0.45	0.18	0.15	0.19

Table 2. Standardized Disparate impact Outcome (DO) and Volatility (DV). Note that DO and DV utilize different scales.

	No Mit.		Pre-		In-		Post-	
	FO	FV	FO	FV	FO	FV	FO	FV
No ensemble	0.84	0.16	**0.58**	0.49	0.78	0.71	0.83	0.18
Bagging	0.97	0.22	0.08	0.16	0.72	**0.13**	0.96	0.21
Boosting	**1.00**	**0.13**	0.07	**0.14**	0.80	0.44	**0.98**	**0.14**
Voting	0.88	0.14	0.00	0.46	0.75	0.38	0.58	0.49
Stacking	0.95	**0.13**	0.55	0.56	**0.90**	0.46	0.90	0.29

Table 3. Standardized F_1 outcome (FO) and volatility (FV).

as "standardized M volatility". The tables and figures that follow report values normalized as described above.

Do ensembles help with fairness? Table 2 shows the disparate impact results. Mitigation almost always improved disparate impact outcomes, but ensemble learning generally incurred a slight penalty relative to the no-ensemble baseline. However, ensemble learning does generally reduce disparate impact volatility. This increased stability may be preferred over better yet more unstable predictions.

Do ensembles help predictive performance when there is mitigation? Table 3 shows F_1 results. Even with ensemble learning, mitigation decreases predictive performance, but relative to standalone mitigators, mitigated ensembles typically have better outcomes or stability, but not both. Except for a few cases, mitigated ensembles *can* help with predictive performance *or* F_1 volatility.

Guidance for method selection. To advise future practitioners based on our results, we generated Fig. 2 from optimal configurations for particular metrics and data setups. To generate it, we do the following:

1. Organize all results by dataset.
2. Filter results for each dataset to ones that occur in the top 33% of results for both standardized disparate impact outcome and standardized F_1 outcome.
3. Place each result into one of four quadrants based on the dataset's baseline fairness and size.
4. Average each metric in each quadrant while grouping by model configuration.
5. Report the best configuration per quadrant and metric.

A more detailed version of this diagram appears in our technical report [7]. We intend to validate this diagram via automatic parameter search and leave-one-dataset-out approaches in future work.

4. Conclusion

This paper introduces a library of modular bias mitigators and ensembles. Our experiments confirm that ensembles can improve fairness stability and provide guidance to practitioners. Of course, the best approach depends on the setting.

References

[1] Guillaume Baudart, Martin Hirzel, Kiran Kate, Parikshit Ram, Avraham Shinnar, and Jason Tsay. Pipeline combinators for gradual AutoML. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[2] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, 2018.

[3] Dheeraj Bhaskaruni, Hui Hu, and Chao Lan. Improving prediction fairness via model ensemble. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1810–1814, 2019.

[4] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: Experiences from the scikit-learn project, 2013.

[5] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Conference on Fairness, Accountability, and Transparency (FAT)*, pages 319–328, 2019.

[6] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016.

[7] Michael Feffer, Martin Hirzel, Samuel C. Hoffman, Kiran Kate, Parikshit Ram, and Avraham Shinnar. An empirical study of modular bias mitigators and ensembles. *CoRR*, abs/2202.00751, February 2022.

[8] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Conference on Knowledge Discovery and Data Mining (KDD)*, pages 259–268, 2015.

[9] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. The (im)possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM (CACM)*, 64(4):136–143, March 2021.

[10] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. On fairness, diversity and randomness in algorithmic decision making. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2017.

[11] Martin Hirzel, Kiran Kate, and Parikshit Ram. Engineering fair machine learning pipelines. In *Non-archival ICLR Workshop on Responsible AI (RAI@ICLR)*, May 2021.

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

[13] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *International Conference on Data Mining (ICDM)*, pages 924–929, 2012.

[14] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pages 35–50, 2012.

[15] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning (ICML)*, pages 2564–2572, 2018.

[16] Patrik Joslin Kenfack, Adil Mehmood Khan, S.M. Ahsan Kazmi, Rasheed Hussain, Alma Oracevic, and Asad Masood Khattak. Impact of model ensemble on the fairness of classifiers in machine learning. In *International Conference on Applied Artificial Intelligence (ICAPAI)*, pages 1–6, 2021.

[17] Alan Mishler and Edward Kennedy. FADE: FAir Double Ensemble learning for observable and counterfactual outcomes, 2021.

[18] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

[19] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations Newsletter*, 15(2):49–60, June 2014.

[20] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, fourth edition, 2016.

[21] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In

International Conference on Machine Learning (ICML), pages 325–333, 2013.

[22] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Conference on AI, Ethics, and Society (AIES)*, pages 335–340, 2018.