FairRankTune: A Python Toolkit for Fair Ranking Tasks

Kathleen Cachel
Worcester Polytechnic Institute
Worcester, MA, USA
kcachel@wpi.edu

Elke Rundensteiner
Worcester Polytechnic Institute
Worcester, MA, USA
rundenst@wpi.edu

Abstract

We present FairRankTune, a multi-purpose open-source Python toolkit offering three primary services: quantifying fairness-related harms, leveraging bias mitigation algorithms, and constructing custom fairness-relevant datasets. FairRankTune provides researchers and practitioners with a self-contained resource for fairness auditing, experimentation, and advancing research. The central piece of FairRankTune is a novel fairness-tunable ranked data generator, RankTune, that streamlines the creation of custom fairness-relevant ranked datasets. FairRankTune also offers numerous fair ranking metrics and fairness-aware ranking algorithms within the same plug-and-play package. We demonstrate the key innovations of FairRankTune, focusing on features that are valuable to stakeholders via use cases highlighting workflows in the end-to-end process of mitigating bias in ranking systems. FairRankTune addresses the gap of limited publicly available datasets, auditing tools, and implementations for fair ranking.

CCS Concepts

• **Software** → **Software libraries and repositories**; • **Social and professional topics** → **User characteristics**.

Keywords

Toolkit, Bias Mitigation, Group Fairness, Individual Fairness, Fair Ranking, Experimentation

ACM Reference Format:
Kathleen Cachel and Elke Rundensteiner. 2024. FairRankTune: A Python Toolkit for Fair Ranking Tasks. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679238

1 Introduction

Motivation. Over the last few years, fair ranking has become an active area of research due to the ubiquitous nature and societal impact of ranking-based tasks. As this impactful area grows, a substantial obstacle faced by researchers is the unavailability of rich fairness-relevant ranked datasets [10, 13]. This is in part due to the challenge of releasing sensitive real-world datasets for fairness research. Fairness analysis necessitates protected information such as gender, age, or race, which, when made public, can pose privacy risks. Moreover, legal restrictions often prevent platforms from collecting such information in the first place, along with restricting its use, sharing, and retention [5]. Despite the breadth of recent work in fair ranking [13, 14], very few fairness metrics and fair ranking algorithms have publicly available implementations.

State-of-the-Art. Fairness-focused toolkits such as Fairlearn [3], Aequitas [15], LiFT [20], FairML [1], and AIF360 [2] are dedicated almost exclusively for *fair classification* and its corresponding metrics and algorithms. However, these fair classification tools cannot be used directly for *ranking-based contexts* due to the task-specific nature of fairness metrics and algorithms. For instance, rankings are inherently ordinal objects that require relative positioning decisions. Thus, fair classification constructs that evenly represent groups in different (often binary) classes are not applicable.

To the best of our knowledge, FARE [9], librec-auto [19] and Fairsearch [23] are the available tools supporting fair ranking workflows. FARE and librec-auto provide group fair ranking metrics, while Fairsearch implements the FA*IR [21] and DELTR [22] fair ranking algorithms. However, these resources are restricted in that they support only on the simplified case of *binary* demographic groups (e.g., black vs. white) compared to real-world scenarios such as multiple races. Based on the limited landscape of open fair-ranking toolkits, practitioners, and researchers lack tools for assessing and improving the fairness of ranking systems, particularly for the setting of multiple groups.

Our Approach. To address the above gap and reduce the friction posed by current tools, we present the open-source FairRankTune Python toolkit. The objective is to provide researchers and practitioners with an *end-to-end fair ranking toolkit* supporting *(1) data generation, (2) bias measurement, and (3) bias mitigation.* The RankTune toolkit contains three primary components. First, a novel fairness-tunable data generation method, called RankTune , is packaged in FairRankTune. RankTune provides the following practical capabilities. It can: (a.) produce ranked data along the entire statistical parity fairness spectrum, (b.) generate distinct rankings all with the same degree of fairness, (c.) its fairness-tuning mechanism offers a consistent usage-pattern and interpretation across diverse item sets, and (d.) supports multiple, not just binary, groups.

Second, FairRankTune offers a *Fairness Metric* library containing numerous popular fairness metric implementations. This Metric library provides toolkit users with multiple alternative approaches for how to calculate top-level metrics. For instance, for group exposure [18], a popular fairness criteria [18, 22], FairRankTune offers seven ways of calculating a top-level exposure metric (e.g., min-max ratios, max absolute difference, L-2 norms of per-group exposures, etc.). This provides enhanced customizability by allowing toolkit users to utilize their preferred formulation. This also lowers the

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679238

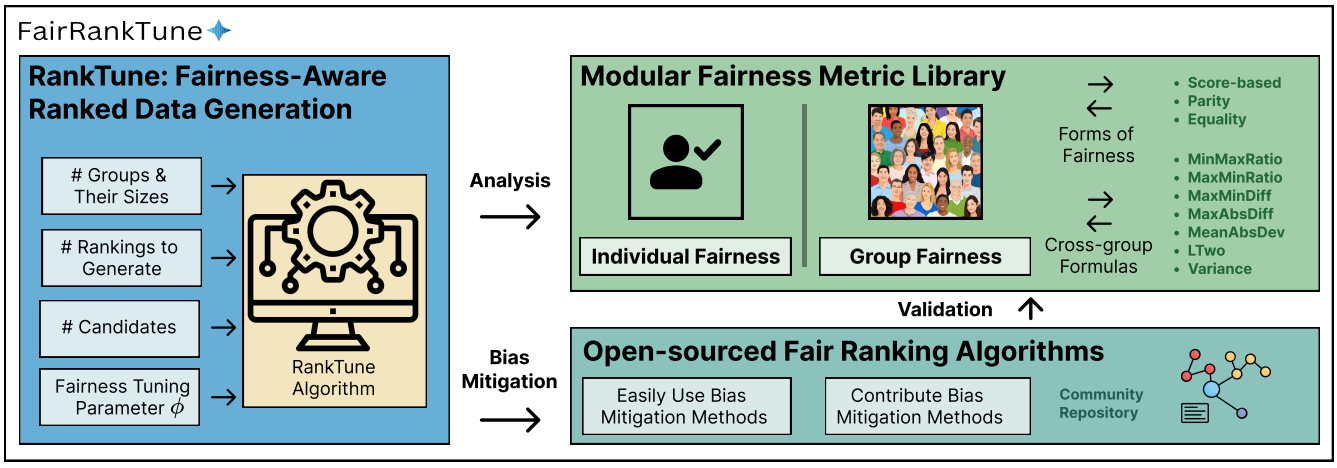

Figure 1: FairRankTune Toolkit: Architecture and Components for Supporting Fair Ranking Tasks

barrier for researchers to conduct studies comparing these formulations with one another or adding new formulations. Third, we provide a fair ranking bias mitigation module containing open-sourced implementations of fair ranking algorithms [6, 7]. These can be applied to any given or generated dataset and then evaluated with the user's choice of provided metrics.

In a nutshell, FairRankTune's wide range of functionalities promotes reproducibility, ease of use, and customization. The FairRankTune toolkit has been *open-sourced* on the PyPi platform. Our *demonstration video* is available at: https://tinyurl.com/frtdvid.

2 The FairRankTune Framework

The high-level framework of the FairRankTune toolkit is depicted in Figure 1. FairRankTune consists of complementary components that play a crucial role in facilitating algorithmic fairness research, conducting real-world bias analysis, and developing fairness-enhanced systems. These components encompass an easy-to-use ranked data generator, a comprehensive metrics module for tasks such as experimental benchmarking or auditing, and an extensible fair ranking algorithm library. All components are included as subpackages under the main FairRankTune Python package, and the toolkit user only needs to import the FairRankTune library to access these functionalities. Extensive details, usage instructions, and example Jupyter Notebooks are available in the accompanying FairRankTune documentation: https://kcachel.github.io/fairranktune/.

3 FAIRRANKTUNE Capabilities

3.1 Novel RankTune Data Generator

RankTune is a novel group fairness-tunable data generation mechanism released in FairRankTune. RankTune makes it easy to create custom fairness-relevant datasets for research, experimentation, and software testing. The assumption of RankTune is that to generate rankings satisfying the general notion of statistical parity fairness [13, 14], the likelihood of a group receiving a positive outcome should be equal to that group's proportion of the candidate pool. We thus first record each group's proportion of the candidate pool. Then we add "unfairness" by distorting the proportional relationship between the likelihood of a group receiving a positive outcome[1] and its proportion of the candidate pool.

Next, the RankTune algorithm places candidate items individually into the to-be-constructed ranking using a controlled pseudo-stochastic process that steers the fairness of the generated ranking(s) using a group "representativeness" parameter $\phi \in [0, 1]$. When $\phi = 0$, RankTune generates unfair rankings, meaning there are significant disparities in the presence of groups in favorable rank positions. As ϕ increases to a maximum value of 1, groups are represented more fairly, meaning groups receive proportional shares of advantageous rank positions.

The key mechanism of RankTune is a repeat insertion process. The ranking is generated by iteratively placing items into it by sampling the uniform $[0, 1]$ interval. The sampled value determines which item, and thus group, is placed into the ranking next. Each group is "assigned" a region of this interval. Specifically, when $\phi = 1$ (most fair), each group's region is equal to that group's proportion of the ranked item set. Unfairness is added by manipulating the size of each group's region via the ϕ representativeness tuning parameter. Our sampling-driven generation concept ensures RankTune can generate multiple distinct rankings of similar degrees of fairness yet it remains easily reproducible via a random seed.

The capabilities of RankTune can be seen in Figure 2 which displays the average values of two fairness metrics, including 95% confidence intervals evaluating a spectrum of RankTune generated rankings. Specifically, we generate 200 rankings for ϕ ranging from 0 to 1 (x-axis), in increments of 0.1 for six multi-group distributions (e.g., ranging from three to eleven groups). We can see that RankTune outputs progressively more fair rankings as ϕ increases. Robustness is seen through the relatively small confidence intervals. RankTune behaves consistently across both different item sets and multiple metrics (EXP [18] on left and AWRF [16] on right).

[1]We view the positive outcome to be the placement of a candidate into higher favorable positions in the generated ranking.

Name	Distributions - Group Proportions
Dist A	(0.2, 0.3, 0.5)
Dist B	(0.1, 0.3, 0.6)
Dist C	(0.2, 0.3, 0.1, 0.05, 0.03)
Dist D	(0.2, 0.2, 0.2, 0.2, 0.2)
Dist E	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.05, 0.05)
Dist F	(0.6, 0.08, 0.02, 0.15, 0.1, 0.05)

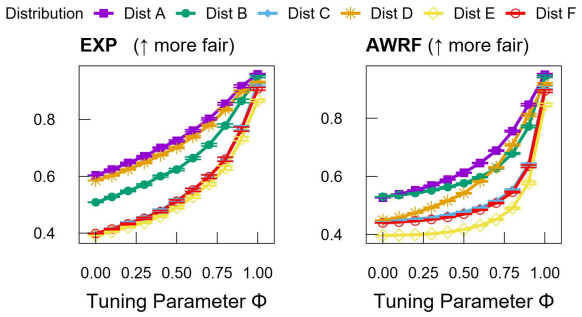

Figure 2: Example of RankTune's effectiveness and controllability. Average metric values (with 95% confidence intervals) are plotted for 200 RankTune generated rankings at each ϕ.

Table 1: Overview of core metrics in FairRankTune.

Metric	Fairness Form	Score-based
Group Exp (EXP) [18]	Group	X
Exposure Utility (EXPU) [18]	Group	✓
Exposure Realized Utility (EXPRU) [18]	Group	✓
Attention Weighted Rank Fairness [16]	Group	X
Exposure Rank Biased Precision Equality [8]	Group	X
Exposure Rank Biased Precision Proportionality [8]	Group	X
Exposure Rank Biased Precision Proportional to Relevance [8]	Group	✓
Attribute Rank Parity [4]	Group	X
Normalized Discounted KL-Divergence [7]	Group	X
Inequity of Amortized Attention	Individual	✓

In the RankTune module, toolkit users have four different ways to generate ranked data. Each function operates on the ϕ representativeness parameter but differs in whether rankings are generated from user-given *candidates* or *group proportions*. `GenFromGroups` generates rankings from the number of items to be ranked, an array of each group's proportion of the total number of items, and the ϕ representativeness parameter. `GenFromItems` generates rankings based on the ϕ parameter and an existing dataset. The sibling methods `ScoredGenFromGroups` and `ScoredGenFromItems` also generate relevance scores modeling the desired level of bias.

3.2 Modular and Diverse Metric Library

The *Metric library* of FairRankTune provides over fifty ways to measure fairness in rankings. It encompasses the two prominent dimensions of fairness in the Algorithmic Fairness community – *group and individual fairness* [13]. Group fairness metrics include metrics that incorporate relevance scores associated with items, so-called score-based fairness [18], and statistical parity metrics that are based on the representation of groups [17]. All group fairness metrics support multiple groups.

The Metric library currently contains ten core metrics summarized in Figure 3. A key innovation of FairRankTune is to provide toolkit users multiple choices for *how to calculate* a top-level fairness metric. The fairness literature shows that most fair classification metrics are aggregation metrics that, through a mathematical formula, distill per-group metrics into one value [12]. This single value is typically reported as the fairness metric itself. For instance, in fair classification, we measure the true positive rate (TPR) for each group, and then combine them into a single value by taking the min-max ratio of the TPRs. Thus, the per-group metric is TPR, which can be combined in any number of ways. We put forth that the same conceptual idea is true in contemporary fair ranking metrics and specifically design FairRankTune with this additional modularity.

```
import FairRankTune as frt
EXPMaxMinDiff, exps_d = frt.Metrics.EXP(ranking_df,
                        item_group_dict, 'MaxMinDiff')
EXPMinMaxRto, exps_r = frt.Metrics.EXP(ranking_df,
                        item_group_dict, 'MinMaxRatio')
```

Listing 1: Usage example of the `Metrics` library to calculate EXP two alternate ways using different cross-group formulas.

Eight of the FairRankTune core metrics are "meta-metrics", i.e., they combine per-group measurements into a single value. Across the fairness literature, these per-group metrics are combined in diverse ways, e.g., min-max ratios, variance, or L-2 norm [18, 22]. FairRankTune offers seven modes of calculating top-level fairness metrics. This implementation allows for toolkit users to choose their preferred formulation. An example of how to specific the metric calculation can bee seen in listing 1. Further, this design lowers the barrier for researchers to conduct additional studies, such as comparing these formulations and conducting user studies on their interpretability. The metric documentation, we have prepared with FairRankTune is also a resource to the community at large, explaining each metric's conceptualization of fairness, offering multiple usage examples, and highlighting source papers and their BibTeX references.

3.3 Open-Sourced Fair Ranking Algorithms

We have also implemented and added to FairRankTune an initial set of multi-group fair ranking algorithms. These algorithms can be used as experimental baselines or as fairness interventions. The library is extensible, meaning the community is invited to add additional implementations into the toolkit. The *Rankers* module allows users to leverage fair ranking methods within the FairRankTune ecosystem. It currently provides the DetConstSort [7] and Epsilon-Greedy [6] algorithms, with others easily added in the future. As neither algorithm has a publicly available implementation accompanying its introduction, this allows toolkit users to easily work with state-of-the-art ranking methods.

4 FAIRRANKTUNE Demonstration

We will demonstrate several of the critical features of FairRankTune via a concrete bias mitigation use case.

Fairness Analyzing a New Algorithm. Imagine you're a data scientist who has been developing a new ranking algorithm, `rankify`.

Your company wants to use `rankify` to prioritize how job applicants are shown to hiring teams. Thus, they want to make sure that `rankify` does not advantage certain groups, specifically males or females. Using historical data, you would like to understand how this algorithm impacts different groups compared to the company's old ranking process. You're also interested in comparing `rankify` to methods that already exist for making fair rankings.

Setup. We use the German Credit dataset [11] as the company's historical data. The old candidate ranking algorithm, `old-rank`, simply sorts candidates by descending scores. Your new algorithm, `rankify`, works by adding the average score of males to every female's score before ordering candidates. More details can be found in the corresponding notebook https://tinyurl.com/frtdemo.

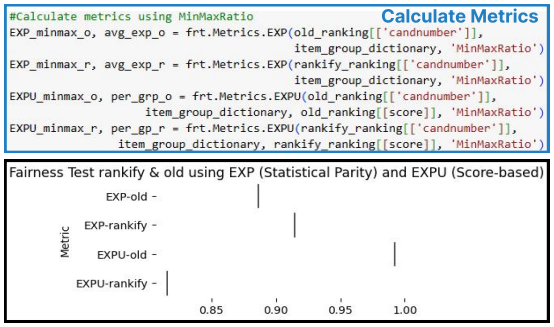

Figure 3: Using diverse fairness metrics in FairRankTune, toolkit users can easily calculate and compare metrics.

Fairness Testing with Company Data. You begin by comparing the fairness of `rankify` and `old-rank` using the FairRankTune Metric library. Using two function calls per algorithm, you test two types of group fairness, statistical parity and score-based fairness, using the group exposure (EXP) and group exposure utility (EXPU) metrics, respectively. The plotted results are in Figure 3. Using MaxMinRatio to measure unfairness means that values closer to 1 are more fair. You can see that using EXP `old-rank` was pretty fair to begin with and `rankify` is fairer than `old-rank`. In EXPU, `old-rank` was very fair, and `rankify` made it less fair. So, depending on the fairness objective, `rankify` can help or hurt. Since `rankify` is trying to push up a marginalized group, you focus on the EXP metric. Nonetheless, the company's historical data does not show significant disparities between females and males, making it a poor test bed for evaluating the bias-mitigation capabilities of those algorithms. Data that exhibits more bias would be helpful in this case.

Fairness Testing using RankTune Data. To get data with more bias, you utilize the RankTune data generator. Using a single function call, you produce a dataset with the same group distribution as the company's historical data but with more bias. On this new dataset, you test `rankify` using EXP. Using the Metric library, by changing one parameter in the EXP function, you measure EXP in two different ways with MaxMinRatio and MaxMinDiff. Figure 4 shows the procedure to generate this new scenario. Specifically,

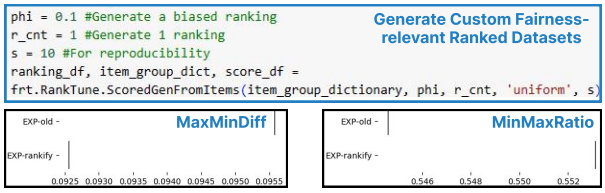

Figure 4: Using the novel fairness-relevant ranked data generator RankTune, users can easily create custom datasets for testing their new methods (e.g., `rankify` compared to `old-rank`). EXP is shown with two cross-group formulas.

working with the historical candidate pool you set $\phi = 0.1$ and use the`ScoredGenFromItems` function.

Figure 4 shows the results of measuring EXP with two different formulas. Examining the plots, you confirm that this generated data is, in fact, very biased. Thus, the `old-rank` method that just ranks candidates by decreasing score is now completely unfair. You verify `rankify` offers improvements over the company's prior method.

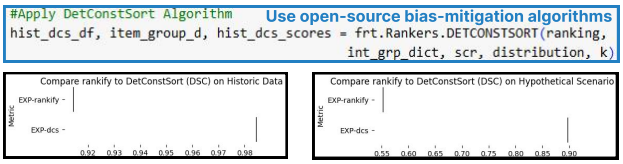

Figure 5: Using the state-of-the-art methods in FairRankTune, users can seamlessly compare their new methods (e.g., `rankify`) to alternate fair ranking methods (e.g., DCS [7]).

Comparing Against State-of-the-Art. While `rankify` appears to work well in mitigating biases, you're curious if a better performance can be achieved using other existing methods. For this, you turn to the algorithms implemented in FairRankTune. In a line of code, you run the DetConstSort algorithm on the historic data and on the hypothetical scenario you generated using RankTune. Figure 5 shows the produced plots of this comparison. You confirm that in both cases, DetConstSort outperforms `rankify`. Thus, you consider suggesting the company use DetConstSort instead.

Demonstration Engagement. In addition to our guided demonstration, participants can change the metrics that are calculated, adjust the bias level in the RankTune generated data, generate their own custom datasets, and compare against additional algorithms.

5 Conclusion

This demonstration showcases the FairRankTune toolkit, a user-friendly interface to many fairness metrics, bias-mitigation techniques, and the first-of-its-kind fairness-aware ranked data generator. Our audience of engineers, researchers, and data scientists can use FairRankTune for critical applications and explorations.

Acknowledgments

This research was supported in part by the NSF-IIS 2007932 grant. We thank the anonymous reviewers for their feedback.

References

[1] Julius Adebayo. 2016. FairML : ToolBox for diagnosing bias in predictive modeling. https://pypi.org/project/fairml/

[2] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Kr. Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John T. Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. *ArXiv* abs/1810.01943 (2018).

[3] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. Fairlearn: A toolkit for assessing and improving fairness in AI. *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020).

[4] Kathleen Cachel, Elke Rundensteiner, and Lane Harrison. 2022. MANI-Rank: Multiple Attribute and Intersectional Group Fairness for Consensus Ranking. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 1124–1137. https://doi.org/10.1109/ICDE53745.2022.00089

[5] Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. https://data.europa.eu/eli/reg/2016/679/oj

[6] Yunhe Feng and Chirag Shah. 2022. Has CEO Gender Bias Really Been Fixed? Adversarial Attacking and Improving Gender Fairness in Image Search. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 11 (Jun. 2022), 11882–11890. https://doi.org/10.1609/aaai.v36i11.21445

[7] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2221–2231. https://doi.org/10.1145/3292500.3330691

[8] Ömer Kırnap, Fernando Diaz, Asia Biega, Michael Ekstrand, Ben Carterette, and Emine Yilmaz. 2021. Estimation of Fair Ranking Metrics with Incomplete Judgments. (2021), 1065–1075. https://doi.org/10.1145/3442381.3450080

[9] Caitlin Kuhlman, MaryAnn VanValkenburg, and Elke Rundensteiner. 2019. FARE: Diagnostics for Fair Ranking using Pairwise Error Metrics. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2936–2942. https://doi.org/10.1145/3308558.3313443

[10] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2023. Fairness in Recommendation: Foundations, Methods, and Applications. *ACM Trans. Intell. Syst. Technol.* 14, 5, Article 95 (oct 2023), 48 pages. https://doi.org/10.1145/3610302

[11] M Lichman. 2013. UCI Machine Learning Repository: Statlog (German Credit Data) Data Set.

[12] Kristian Lum, Yunfeng Zhang, and Amanda Bower. 2022. De-biasing "bias" measurement. (2022), 379–389. https://doi.org/10.1145/3531146.3533105

[13] Gourab K. Patro, Lorenzo Porcaro, Laura Mitchell, Qiuyue Zhang, Meike Zehlike, and Nikhil Garg. 2022. Fair Ranking: A Critical Review, Challenges, and Future Directions. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 1929–1942. https://doi.org/10.1145/3531146.3533238

[14] Amifa Raj and Michael D. Ekstrand. 2022. Measuring Fairness in Ranked Results: An Analytical and Empirical Comparison. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (, Madrid, Spain,) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 726–736. https://doi.org/10.1145/3477495.3532018

[15] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577* (2018).

[16] Piotr Sapiezynski, Wesley Zeng, Ronald E Robertson, Alan Mislove, and Christo Wilson. 2019. Quantifying the Impact of User Attentionon Fair Group Representation in Ranked Lists. In *Companion Proceedings of The 2019 World Wide Web Conference* (San Francisco, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 553–562. https://doi.org/10.1145/3308560.3317595

[17] Tobias Schumacher, Marlene Lutz, Sandipan Sikdar, and Markus Strohmaier. 2022. Properties of Group Fairness Metrics for Rankings. *ArXiv* abs/2212.14351 (2022).

[18] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 2219–2228. https://doi.org/10.1145/3219819.3220088

[19] Nasim Sonboli, Robin Burke, Zijun Liu, and Masoud Mansoury. 2020. Fairness-aware Recommendation with librec-auto. In *Proceedings of the 14th ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 594–596. https://doi.org/10.1145/3383313.3411525

[20] Sriram Vasudevan and Krishnaram Kenthapadi. 2020. LiFT: A Scalable Framework for Measuring Fairness in ML Applications. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 2773–2780. https://doi.org/10.1145/3340531.3412705

[21] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1569–1578. https://doi.org/10.1145/3132847.3132938

[22] Meike Zehlike and Carlos Castillo. 2020. Reducing Disparate Exposure in Ranking: A Learning To Rank Approach. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2849–2855. https://doi.org/10.1145/3366424.3380048

[23] Meike Zehlike, Tom Sühr, Carlos Castillo, and Ivan Kitanovski. 2020. FairSearch: A Tool For Fairness in Ranked Search Results. (2020), 172–175. https://doi.org/10.1145/3366424.3383534