A Nutritional Label for Rankings

Ke Yang
Drexel University
ky323@drexel.edu

Julia Stoyanovich*
Drexel University
stoyanovich@drexel.edu

Abolfazl Asudeh
University of Michigan
asudeh@umich.edu

Bill Howe†
University of Washington
billhowe@uw.edu

HV Jagadish‡
University of Michigan
jag@umich.edu

Gerome Miklau§
University of Massachusetts Amherst
miklau@cs.umass.edu

ABSTRACT

Algorithmic decisions often result in scoring and ranking individuals to determine credit worthiness, qualifications for college admissions and employment, and compatibility as dating partners. While automatic and seemingly objective, ranking algorithms can discriminate against individuals and protected groups, and exhibit low diversity. Furthermore, ranked results are often unstable — small changes in the input data or in the ranking methodology may lead to drastic changes in the output, making the result uninformative and easy to manipulate. Similar concerns apply in cases where items other than individuals are ranked, including colleges, academic departments, or products.

In this demonstration we present Ranking Facts, a Web-based application that generates a *"nutritional label"* for rankings. Ranking Facts is made up of a collection of visual widgets that implement our latest research results on fairness, stability, and transparency for rankings, and that communicate details of the ranking methodology, or of the output, to the end user. We will showcase Ranking Facts on real datasets from different domains, including college rankings, criminal risk assessment, and financial services.

ACM Reference Format:
Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, HV Jagadish, and Gerome Miklau. 2018. A Nutritional Label for Rankings. In *Proceedings of 2018 International Conference on Management of Data (SIGMOD'18).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3183713.3193568

1 INTRODUCTION

Algorithmic decisions often result in scoring and ranking individuals — to determine credit worthiness, desirability for college admissions and employment, and compatibility as dating partners. While automatic and seemingly objective, rankers can discriminate against individuals and protected groups [1], and exhibit low diversity [9]. Furthermore, ranked results are often unstable — small changes in the input or in the ranking methodology may lead to drastic changes in the output, making the result uninformative and easy to manipulate [3, 10]. Similar concerns apply in cases where items other than individuals are ranked, including colleges, academic departments, and products.

Algorithmic decisions are produced by complex processes with many hidden assumptions, and are increasingly used outside of the original context for which they were intended. In response, developers, regulators and the public need to quickly determine the "fitness for use" of a given model or dataset, and to assess the methodology that was used to produce it. This motivates development of interpretability and transparency tools.

With the exception of recent machine learning results that enable interpretability of particular classes of algorithms [6, 7, 12], recent scholarship on algorithmic accountability has primarily focused on enabling an analyst to retroactively verify particular properties rather than proactively exposing a standard suite of information. Because algorithmic processes can be complex or secret, these methods rely on retrospective checks, using techniques like zero knowledge proofs [4], audits [8], and reverse engineering [5]. These are valid methods of interrogation, but they put a significant burden on users. The burden should instead be borne by the vendor who produced the result, who is in a better position to explain it.

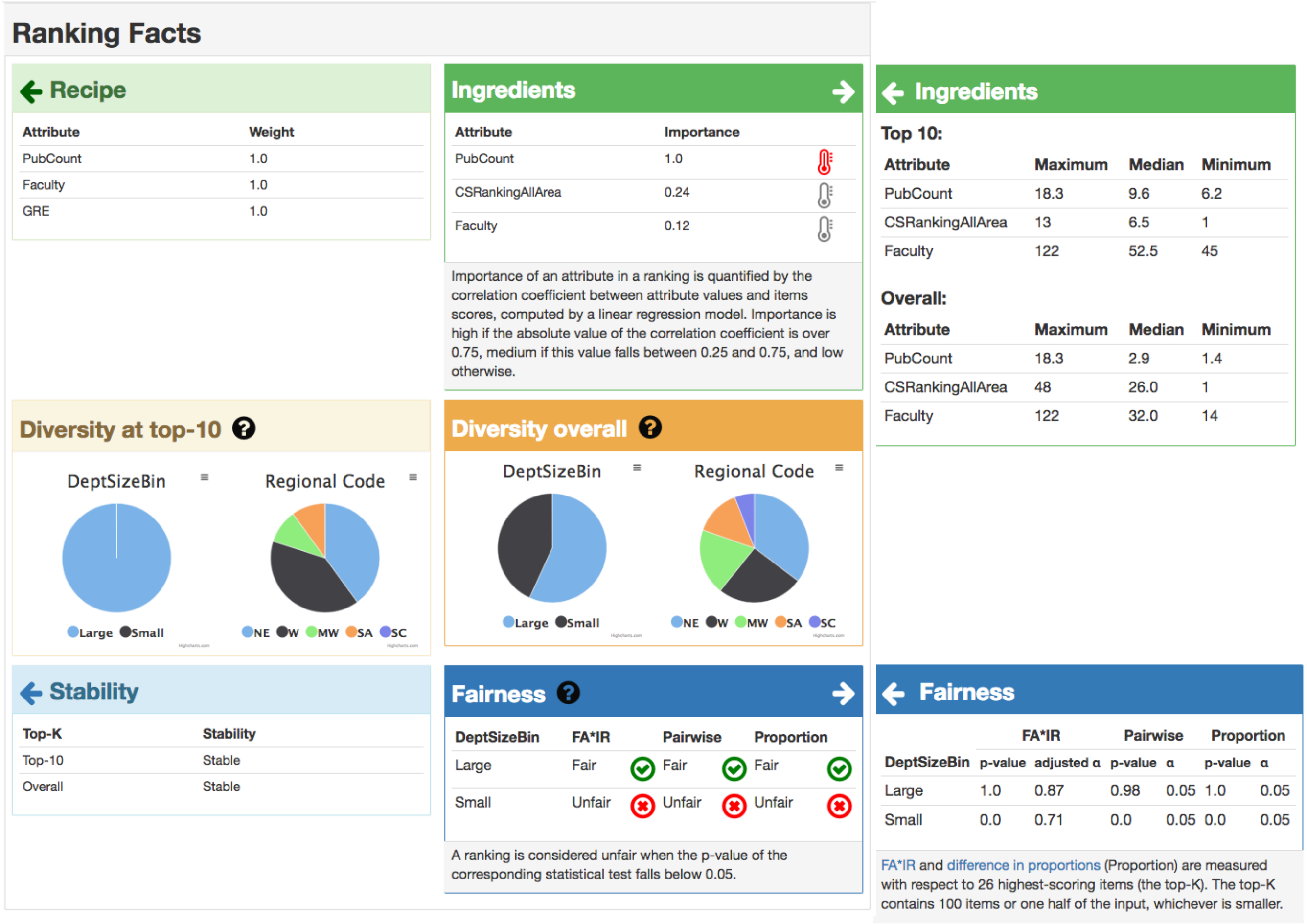
In this work we develop an interpretability tool, Ranking Facts, that is based on the concept of a *nutritional label*. We draw an analogy to the food industry, where simple, standardized labels convey information to consumers about the ingredients and production processes. Short of setting up a chemistry lab, the consumer would otherwise have no access to this information. Similarly, Ranking Facts explains ranked outputs to a user, with appropriately summarized information regarding the ranking process.

An example of the output produced by our tool is presented in Figure 1. It explains a ranked set of Computer Science departments. The data was obtained from CS Rankings (https://github.com/emeryberger/CSRankings), augmented with attributes from the NRC (http://www.nap.edu/rdp/) dataset, see details in Section 3.

Ranking Facts is made up of a collection of visual widgets. Each widget addresses an essential aspect of transparency and interpretability, and is based on our recent technical work on fairness and diversity [2, 9, 11, 13], transparency [10], and stability (ongoing) in algorithmic rankers. We describe next how we explain rankings using the widgets (Section 2) and then discuss demonstration scenarios (Section 3) before concluding in Section 4.

[*] This work was supported in part by NSF Grant No. 1741047.
[†] This work was supported in part by NSF Grant No. 1740996.
[‡] This work was supported in part by NSF Grant No. 1741022.
[§] This work was supported in part by NSF Grant No. 1741254.

SIGMOD'18, June 10–15, 2018, Houston, TX, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-4703-7/18/06…$15.00
https://doi.org/10.1145/3183713.3193568



Ranking Facts

Recipe

Attribute	Weight
PubCount	1.0
Faculty	1.0
GRE	1.0

Ingredients

Attribute	Importance
PubCount	1.0
CSRankingAllArea	0.24
Faculty	0.12

Importance of an attribute in a ranking is quantified by the correlation coefficient between attribute values and items scores, computed by a linear regression model. Importance is high if the absolute value of the correlation coefficient is over 0.75, medium if this value falls between 0.25 and 0.75, and low otherwise.

Ingredients

Top 10:

Attribute	Maximum	Median	Minimum
PubCount	18.3	9.6	6.2
CSRankingAllArea	13	6.5	1
Faculty	122	52.5	45

Overall:

Attribute	Maximum	Median	Minimum
PubCount	18.3	2.9	1.4
CSRankingAllArea	48	26.0	1
Faculty	122	32.0	14

Diversity at top-10 ?

Diversity overall ?

Stability

Top-K	Stability
Top-10	Stable
Overall	Stable

Fairness ?

DeptSizeBin	FA*IR	Pairwise	Proportion
Large	Fair ✓	Fair ✓	Fair ✓
Small	Unfair ✗	Unfair ✗	Unfair ✗

A ranking is considered unfair when the p-value of the corresponding statistical test falls below 0.05.

Fairness

	FA*IR		Pairwise		Proportion	
DeptSizeBin	p-value	adjusted α	p-value	α	p-value	α
Large	1.0	0.87	0.98	0.05	1.0	0.05
Small	0.0	0.71	0.0	0.05	0.0	0.05

FA*IR and difference in proportions (Proportion) are measured with respect to 26 highest-scoring items (the top-K). The top-K contains 100 items or one half of the input, whichever is smaller.

Figure 1: Ranking Facts for the CS departments dataset. The Ingredients widget (green) has been expanded to show the details of the attributes that strongly influence the ranking. The Fairness widget (blue) has been expanded to show the computation that produced the fair/unfair labels.

2 EXPLAINING RANKINGS

Figure 1 presents Ranking Facts for CS department rankings. The nutritional label consists of six widgets, each with an overview and a detailed view, which we now describe.

2.1 Recipe and Ingredients

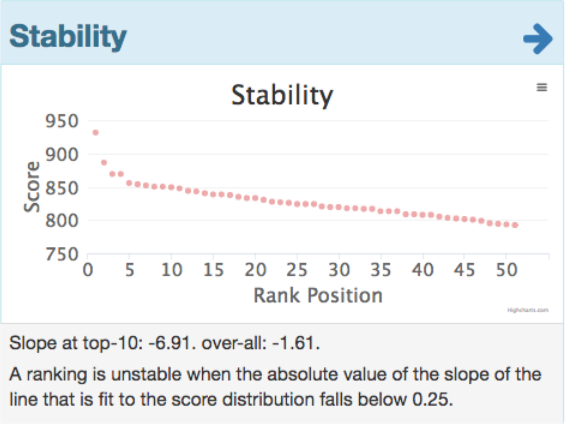
These two widgets help to explain the ranking methodology. The Recipe widget succinctly describes the ranking algorithm. For example, for a linear scoring formula, each attribute would be listed together with its weight. The Ingredients widget lists attributes most material to the ranked outcome, in order of importance. For example, for a linear model, this list could present the attributes with the highest learned weights. Put another way, the explicit intentions of the designer of the scoring function about which attributes matter, and to what extent, are stated in the Recipe, while Ingredients may show additional attributes associated with high rank. Such associations can be derived with linear models or with other methods, such as rank-aware similarity in our prior work [9].

The detailed Recipe and Ingredients widgets list statistics of the attributes in the Recipe and in the Ingredients: minimum, maximum and median values at the top-10 and over-all.

2.2 Stability

The Stability widget explains whether the ranking methodology is robust on this particular dataset. An unstable ranking is one where slight changes to the data (e.g., due to uncertainty and noise), or to the methodology (e.g., by slightly adjusting the weights in a score-based ranker) could lead to a significant change in the output. This widget reports a stability score, as a single number that indicates the extent of the change required for the ranking to change.

As with the widgets above, there is a detailed Stability widget to complement the overview widget. An example is shown in Figure 2, where the stability of the ranking is quantified as the slope of the line that is fit to the score distribution, at the top-10 and over-all. A score distribution is unstable if scores of items in adjacent ranks are close to each other, and so a very small change in scores will lead



Slope at top-10: -6.91. over-all: -1.61.

A ranking is unstable when the absolute value of the slope of the line that is fit to the score distribution falls below 0.25.

Figure 2: Stability**: detailed widget.**

to a change in the ranking. In this example the score distribution is considered unstable if the slope is 0.25 or lower. Alternatively, stability can be computed with respect to each scoring attribute, or it can be assessed using a model of uncertainty in the data.

2.3 Fairness

The Fairness widget quantifies whether the ranked output exhibits statistical parity (one interpretation of fairness) with respect to one or more sensitive attributes, such as gender or race of individuals [13]. We denote one or several values of the sensitive attribute as a protected feature. For example, for the sensitive attribute gender, the assignment gender=F is a protected feature.

A variety of fairness measures have been proposed in the literature [15], but none are directly applicable to rankings. One typical measure compares the proportion of members of a protected group (e.g., female gender or minority race) who receive a positive outcome to their proportion in the overall population. For example, if the dataset contains an equal number of men and women, then among the individuals invited for a job interview, one half should be women. A measure of this kind can be adapted to rankings by quantifying the proportion of members of a protected group in some selected set of size k (treating the top-k as a set).

In [13], we proposed a generative method to describe rankings that meet a particular fairness criterion (fairness probability f) and are drawn from a dataset with a given proportion of members of a binary protected group (p). This method was used in FA*IR [14] to quantify fairness in every prefix of a top-k list. In our follow-up work (working paper), we are developing a pairwise measure that directly models the probability that a member of a protected group is preferred to a member of the non-protected group.

Let us now return to the Fairness widget in Figure 1. We select a binary version of the department size attribute DeptSizeBin from the CS departments dataset as the sensitive attribute, and treat both values ("large" and "small") as protected features. The summary view of the Fairness widget in our example presents the output of three fairness measures: FA*IR [14], proportion [15], and our own pairwise measure. All these measures are statistical tests, and whether a result is fair is determined by the computed p-value. The detailed Fairness widget provides additional information about the tests and explains the process.

2.4 Diversity

Fairness is related to diversity: ensuring that different kinds of objects are represented in the output of an algorithmic process [2]. Diversity has been considered in search and recommender systems, but in a narrow context, and was rarely applied to profiles of individuals. We are currently working on defining diversity measures for ranked outputs, based on our work in [2, 9].

The Diversity widget shows diversity with respect to a set of demographic categories of individuals, or a set of categorical attributes of other kinds of items [2]. The widget displays the proportion of each category in the top-10 ranked list and over-all, and, like other widgets, is updated as the user selects different ranking methods or sets different weights. In our example in Figure 1, we quantify diversity with respect to department size and to the regional code of the university. By comparing the pie charts for top-10 and over-all, we observe that only large departments are present in the top-10.

3 DEMONSTRATION SCENARIOS

We will demonstrate the utility of Ranking Facts using three real-world data sets, considering several ranking functions for each.

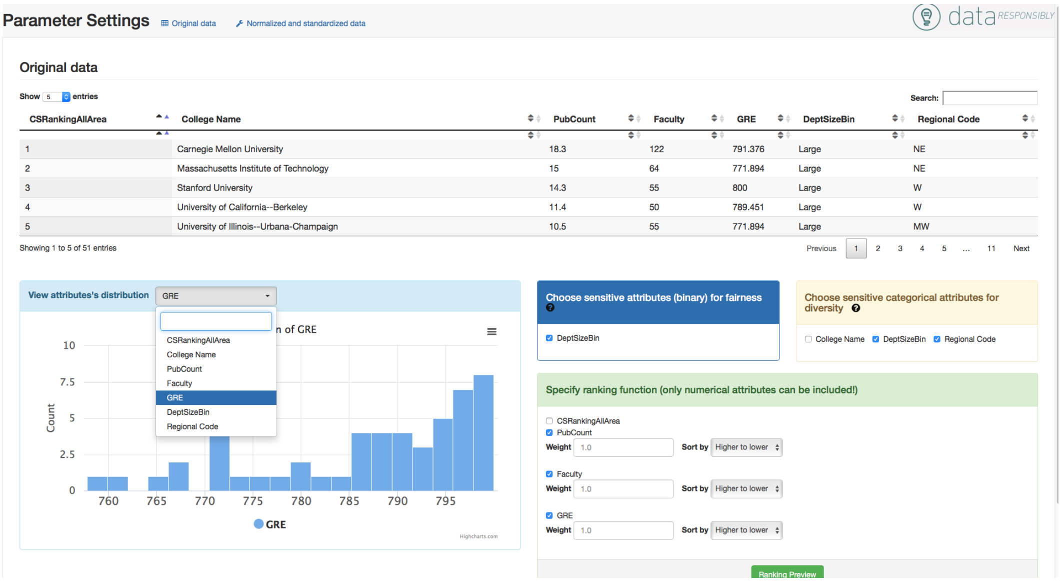
(1) CS departments: CS Rankings (CSR) (https://github.com/emeryberger/CSRankings), with additional attributes from the NRC assessment dataset (http://www.nap.edu/rdp/). This dataset has the following attributes: PubCount (CSR) computes the geometric mean of the adjusted number of publications in each area by institution, Faculty (CSR) is the number of faculty in the department, GRE (NRC) is the average GRE scores (2004-2006), and Region (NRC) is one of NE, MW, SA, SC, W regions in the US. We have been using this dataset in our examples throughout this paper.

(2) Criminal risk assessment: a dataset collected and published by ProPublica as part of their investigation into racial bias in criminal risk assessment software called COMPAS (https://github.com/propublica/compas-analysis). The dataset contains demographics, recidivism scores produced by COMPAS, and criminal offense information for 6,889 individuals.

(3) Credit and loans: the German Credit dataset from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/), with demographic and financial information on 1000 individuals.

The demo user has the option to choose one of these datasets, or to upload one of their own (as a fully populated table in CSV format). Next, they can choose a ranking method from pre-populated options, or they can rank using scoring function of their own specification. If they make the latter choice, the system provides assistance. Figure 3 presents a portion of the scoring function design view. Here, the user can decide whether to work with raw data or to normalize and standardize the attributes (checkbox at the top-left of Figure 3). The system generates a preview of the data, and allows the user to plot the distribution of values of each attribute as a histogram (shown here for the attribute GRE).

The bottom-right portion of Figure 3 contains the attribute selection areas: at least one categorical attribute must be chosen as the sensitive attribute. Ranking Facts will evaluate fairness with respect to every value in the domain of this attribute, and is currently



Parameter Settings · Original data · Normalized and standardized data

Original data

Show 5 entries · Search:

CSRankingAllArea	College Name	PubCount	Faculty	GRE	DeptSizeBin	Regional Code
1	Carnegie Mellon University	18.3	122	791.376	Large	NE
2	Massachusetts Institute of Technology	15	64	771.894	Large	NE
3	Stanford University	14.3	55	800	Large	W
4	University of California--Berkeley	11.4	50	789.451	Large	W
5	University of Illinois--Urbana-Champaign	10.5	55	771.894	Large	MW

Showing 1 to 5 of 51 entries · Previous 1 2 3 4 5 … 11 Next

View attributes's distribution: GRE
- CSRankingAllArea
- College Name
- PubCount
- Faculty
- GRE
- DeptSizeBin
- Regional Code

Choose sensitive attributes (binary) for fairness

☑ DeptSizeBin

Choose sensitive categorical attributes for diversity

☐ College Name ☑ DeptSizeBin ☑ Regional Code

Specify ranking function (only numerical attributes can be included!)

☐ CSRankingAllArea
☑ PubCount
Weight 1.0 Sort by Higher to lower

☑ Faculty
Weight 1.0 Sort by Higher to lower

☑ GRE
Weight 1.0 Sort by Higher to lower

Ranking Preview

Figure 3: Scoring function design.

limited to binary attributes. Finally, the user selects at least one numerical attribute for the scoring function, and assigns a weight to this attribute. Scoring attribute selection and weight assignment is based on a user's a priori judgment regarding item quality, but can be informed by the range and distribution of values for a given attribute. When scoring attributes are selected, the user will preview the ranking, and will then either refine it, or go on to generate Ranking Facts such as that in Figure 1.

The demo presenter will inspect the resulting nutritional label together with the user, guiding the user in exploring selected widgets in detail. For example, we may notice that many attributes in the Recipe do not coincide with those that most impact the ranked outcome in the Ingredients, as is the case in Figure 1: attribute GRE is one of the scoring attributes, but it does not correlate with the ranked outcome. Inspecting the detailed Recipe widget, we observe that the range of values and the median for GRE are very similar in the top-10 and overall, supporting the finding that GRE does not play an important part in the ranking.

4 TAKE-AWAY MESSAGES

In this demonstration we presented Ranking Facts, a system for producing nutritional labels that explain rankings. To the best of our knowledge, our tools are the first to consider interpretability for ranked outputs. Ranking Facts is implemented in Python, and is modular and easy to extend. Our tool is available at http://demo.dataresponsibly.com/rankingfacts/.

The tool is based on the latest research by the authors and others, and reflects known limitations of the state of the art. We are actively working on defining group fairness measures that go beyond binary categories (e.g., can be applied to ethnicity, not only to gender), and will incorporate these into the tool when available. We are also working on extending Ranking Facts to support richer scoring function design functionality. For example, we plan to include methods that help the user mitigate lack of fairness and diversity by suggesting modified scoring functions.

REFERENCES

[1] Danielle K. Citron and Frank A. Pasquale. 2014. The Scored Society: Due Process for Automated Predictions. *Washington Law Review* 89 (2014).

[2] Marina Drosou, HV Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. 2017. Diversity in Big Data: A Review. *Big Data* 5, 2 (2017).

[3] Malcolm Gladwell. February 14, 2011. The order of things. *The New Yorker* (February 14, 2011). https://www.newyorker.com/magazine/2011/02/14/the-order-of-things

[4] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. 2017. Accountable Algorithms. *University of Pennsylvania Law Review* 165 (2017).

[5] Maayan Perel and Niva Elkin-Koren. 2016. Black Box Tinkering: Beyond Transparency in Algorithmic Enforcement. *Florida Law Review* (2016).

[6] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *ACM SIGKDD*. 1135–1144. https://doi.org/10.1145/2939672.2939778

[7] Cynthia Rudin. 2014. Algorithms for interpretable machine learning. In *ACM SIGKDD*. 1519. https://doi.org/10.1145/2623330.2630823

[8] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2014. Auditing Algorithms: Research Methods for Detecting Discrimination on Internet Platforms. *Data and Discrimination: Converting Critical Concerns into Productive Inquiry* (2014).

[9] Julia Stoyanovich, Sihem Amer-Yahia, and Tova Milo. 2011. Making interval-based clustering rank-aware. In *EDBT*. 437–448. https://doi.org/10.1145/1951365.1951417

[10] Julia Stoyanovich and Ellen P. Goodman. August 5, 2016. Revealing Algorithmic Rankers. *Freedom to Tinker* (August 5, 2016). http://freedom-to-tinker.com/2016/08/05/revealing-algorithmic-rankers/

[11] Julia Stoyanovich, Ke Yang, and H. V. Jagadish. 2018. Online Set Selection with Fairness and Diversity Constraints. In *EDBT*. 241–252. https://doi.org/10.5441/002/edbt.2018.22

[12] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. 2016. Bayesian Rule Sets for Interpretable Classification. In *IEEE ICDM*. 1269–1274. https://doi.org/10.1109/ICDM.2016.0171

[13] Ke Yang and Julia Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *SSDBM*. 22:1–22:6. https://doi.org/10.1145/3085504.3085526

[14] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo A. Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *ACM CIKM*. 1569–1578. https://doi.org/10.1145/3132847.3132938

[15] Indre Zliobaite. 2017. Measuring discrimination in algorithmic decision making. *Data Min. Knowl. Discov.* 31, 4 (2017), 1060–1089. https://doi.org/10.1007/s10618-017-0506-1