A Human-in-the-loop Workflow for Multi-Factorial Sensitivity Analysis of Algorithmic Rankers

Jun Yuan
jy448@njit.edu
New Jersey Institute of Technology
Newark, NJ, United States

Aritra Dasgupta
aritra.dasgupta@njit.edu
New Jersey Institute of Technology
Newark, NJ, United States

ABSTRACT

Algorithmic rankers are ubiquitously applied in automated decision systems such as hiring, admission, and loan-approval systems. Without appropriate explanations, decision-makers often cannot audit or trust algorithmic rankers' outcomes. In recent years, XAI (explainable AI) methods have focused on classification models, but there for algorithmic rankers, we are yet to develop state-of-the-art explanation methods. Moreover, explanations are also sensitive to changes in data and ranker properties, and decision-makers need transparent model diagnostics for calibrating the degree and impact of ranker sensitivity. To fulfill these needs, we take a dual approach of: i) designing explanations by transforming Shapley values for the simple form of a ranker based on linear weighted summation and ii) designing a human-in-the-loop sensitivity analysis workflow by simulating data whose attributes follow user-specified statistical distributions and correlations. We leverage a visualization interface to validate the transformed Shapley values and draw inferences from them by leveraging multi-factorial simulations, including data distributions, ranker parameters, and rank ranges.

ACM Reference Format:

Jun Yuan and Aritra Dasgupta. 2023. A Human-in-the-loop Workflow for Multi-Factorial Sensitivity Analysis of Algorithmic Rankers. In *Workshop on Human-In-the-Loop Data Analytics (HILDA '23), June 18, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3597465.3605221

1 INTRODUCTION

Algorithmic rankers are AI models that rank a collection of entities or candidates. They use candidates' attributes as input and output a ranking to the end users, usually as suggestions for decision-making. For example, automated decision systems (ADS) used in hiring apply rankers that sort candidates based on their qualifications for certain job positions' interview process. However, ADS generally lack contextual transparency [17], leading to unforeseeable bias or unfairness towards human candidates that are being ranked, especially for such high-consequence decisions. Such issues dynamically evolve, are hard to predict in real-life scenarios, and are triggered by changes in the data, ranker parameters, ranking subsets, etc. Yet, both end-users (i.e., who use the algorithmic rankers) and data subjects (i.e., the attributes of human candidates) are unequipped with explanation tools. End-users, like a decision-maker, need to reason about how a ranker made its decisions and how sensitive they are to data or model settings or parameter changes. To address this need, we take a dual approach of integrating adaptive explanation methods with an interactive visual interface for sensitivity analysis, where the users can understand the algorithmic rankers and the semantics of the ranking output using data-driven simulations. In this work, we consider sensitivity analysis as a technique to help decision-makers understand what factors impact the ranking outcome and why. In summary, our contributions are: 1) adapting the Shapley values [20] to explain algorithmic rankers. 2) designing a human-in-the-loop sensitivity analysis workflow and visualization interface for algorithmic ranker evaluation and explanation analysis.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

HILDA '23, June 18, 2023, Seattle, WA, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0216-7/23/06…$15.00
https://doi.org/10.1145/3597465.3605221

2 RELATED WORK

Sensitivity analysis has become increasingly important in mathematical modeling and AI-assisted decision-making for understanding the underlying logic of models [11, 25]. Sensitivity analysis allows human stakeholders to gain insights and familiarize themselves with the techniques and, more importantly, limitations [16] under different conditions. In this work, our proposed sensitivity analysis workflow anchors on the following question: *how do changes in candidates, attributes, ranker parameters, scores, and ranking affect ranking explanation?* For underlying data, we add sampling variation noise. The ranking outcome and explanation are expected to be insensitive to noise. Many works about ranking explanation techniques [4, 7] and fairness treatments [24] do not include sensitivity analysis based on real-world scenarios. The proposed methods are tested in limited benchmark datasets and ranker definitions. Existing works focus on developing explanation methods for classification and regression models such as LIME [12], SHAP [12], and LIME-anchor [13]. For algorithmic rankers, many methods have developed, such as using SHAP on local ranking region [2], designing nutrition labels for algorithmic rankers [21], and focusing on explaining monotone rankers [5], but may not be extensively tested for usability via sensitivity analysis. Another thread of research focuses on testing what-if model behavior scenarios [19], but may not be fully adaptive to user needs [8]. Users may have additional prior knowledge that the input data attributes are inter-dependent [1], or the model output is dependent (i.e., the rank output of algorithmic rankers only bear meaning when compared with each other in the same ranking). The complexity of real-world scenarios calls for more principled approaches for communicating sensitivity in the entire workflow for model development and explanation generation [22].

3 SENSITIVITY ANALYSIS WORKFLOW

In this section, we introduce the notation for the algorithmic ranker sensitivity analysis workflow. The workflow includes ranking phase and explanation phase. The ranking phase includes data generation, score generation, and ranking generation. The explanation phase includes three explanation methods. Ranking phase includes the source of potential distribution differences among groups or certain choices of algorithmic ranker parameters. Explanation phase includes the discovery of inferences, such as attributes contributing differently among groups. We separate the ranking phase and explanation phase for our workflow due to the explanation we discussed is not a typical process that is in algorithmic ranker development. Traditionally, the workflow reaches the end when the ranking is produced or certain performance metrics are calculated (e.g., MAP [23], NDCG [3]). The explanation phase aims to make sense of model behaviors beyond simple metrics, hence requiring a second phase in the workflow. Both the input and output from the ranking phase are input to the explanation phase. The ranking phase helps users to observe ranking sensitivity according to changes in the data and ranker parameters. The explanation phase helps users understand how data attributes and ranker parameters contribute to ranking sensitivity.

3.1 Ranking Phase

Data generation. We assume a dataset with three categories of attributes, $\boldsymbol{X} = (\boldsymbol{X_p}|\boldsymbol{X_s}|\boldsymbol{X_{ns}})$. $\boldsymbol{X_p}$ is the collection of protected attributes such as gender and race. $\boldsymbol{X_s}$ is the collection of attributes used in the scoring formula to generate scores. $\boldsymbol{X_{ns}}$ is the collection of attributes that can be used in the scoring formula, but users choose not to use them. To simulate the $\boldsymbol{X_s}$ and $\boldsymbol{X_{ns}}$, we assume they follow certain distributions condition on the protected attributes $\boldsymbol{X_p}$. In this work, we consider $\boldsymbol{X_p}$ to be categorical attributes such as gender or race. They are not used for scoring formulas but determine the distributions of scoring and non-scoring attributes. We assume attributes follow Gaussian distributions, $[\boldsymbol{X_s}, \boldsymbol{X_{ns}}]^T \sim \mathcal{N}(\boldsymbol{\mu}^T_{X_p}, \Sigma_{X_p})$. The distribution parameters are uniquely determined for each group constructed by the protected group $\boldsymbol{X_p}$. For instance, with gender and race, we can define {gender: female and race: black} a group. In practice, the group-specific distribution parameters can be inferred from sources such as US Census data [18] and SAT annual report [14]. The sources may include a scoring attribute's mean and variance for certain groups. In this work, we consider the case of a single protected attribute. In future work, we plan to generalize the Gaussian distribution to more general distributions via the copula theory, which is widely used in finance risk simulation [6, 9].

Scores and ranking generation. In this work, we will constrain the algorithmic rankers to be linear weighted scoring functions, and the notation X in the following discussion refers to the scoring function X_s.

$$f(x_1, x_2, \cdots, x_p) = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \,, \sum \beta_j = 1 \,, \beta_j \geq 0 \quad (1)$$

; the data input and output of the scoring function are represented as a matrix X and a score vector s,

$$X = [X_1, X_2, \cdots, X_p] = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}, \quad s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \quad (2)$$

Each row in the matrix represents attribute data for one candidate, each column represents the attribute value of a such attribute. We assume each X_j is standardized between 0 and 1 to reduce the scaling difference between attributes (e.g., age and SAT score have very different data ranges). Hence, if one data changed in X_j, X_j needs to be re-standardized. Once the data is simulated, we create the scores with the scoring attributes. The benefit of f being a linear function is that the scoring function can be easily interpreted. Another benefit is that the Shapley value (a widely used XAI method) can be calculated using closed-form mathematical equations, which allows a real-time interactive comparison in our interface. Finally, we assume a ranking vector τ is produced by sorting score vector s in descending order. Such a data generation process may be run multiple times and produce a collection of data samples, score samples, and ranking samples. Such simulation enables us to test whether the explanation methods and their characteristics and unfairness interpretation are robust against simulation noise.

3.2 Explanation Phase

We have defined the data X and the algorithmic ranker f. We then seek explanations of f regarding the data X. Shapley values are what we choose as a baseline XAI method in this work due to its massive popularity. But more importantly, many algorithmic ranker f output numerical values, which is similar to the output from regression models. Shapley values are interpretable for regression models but not for algorithmic rankers. We first discuss the interpretability issue of Shapley values when we treat the algorithmic ranker f as a regression model. Then we introduce transformations of Shapley values, which are intrinsically interpretable for algorithmic rankers. We compare Shapley values and transformed Shapley values in the case study via sensitivity analysis simulation.

Shapley value. Shapley values have been extensively explored for explaining complicated classifiers with their model-agnostic characteristic. It is built upon game theory to distribute the "payout" to each attribute to describe how much contribution one attribute contributes to the total "payout" of one entity. In classification, such "payout" is the probability the entity belongs to a class. For a regression model that returns a price value, for instance, the "payout" is the predicted price of the entity subtracted from the average price of all entities. The computing time for Shapley value using the standard python package SHAP [15] is generally too long for real-time user interaction in large datasets. We, hence, use the simpler definition of Shapley value for the linear model. By doing so, we lost the model-agnostic assumption but gain real-time interaction speed. The mathematical formula is heavily borrowed from Molnar [10]. For a linear model, the Shapley value of j-th scoring attribute can be calculated as $\phi_j(f) = \beta_j X_j - E(\beta_j X_j) = \beta_j X_j - \beta_j E(X_j)$. If we attempt to interpret it in our ranking case, the Shapley value of j-th scoring attribute indicates its contribution to the candidate's ranking score differing from an "imaginary" candidate who has an average value of the j-th attribute. In fact, the imaginary candidate has the average value of all the scoring attributes. Such a candidate may not exist in real life. More importantly, the absolute value of the ranking score bears little meaning for algorithmic rankers. The "payout" is neither ranking score nor ranking but the comparison between them. Hence, we cannot directly use Shapley values for algorithmic rankers and we seek alternative methods. The key is to conceptualize a suitable "payout" in the case of algorithmic rankers. Unlike regression models' score output, which may have physical meanings (e.g., weight, age, price), the algorithmic rankers' score and rank output usually do not have meaning other than *relative* importance or *relevance* of the candidates in the ranking. Based on different assumptions of "payout," we can design different transformed Shapley values.

Standardized Shapley values. We first design the standardized Shapley values by standardizing Shapley values. We propose that for an algorithmic ranker, "payout" of x means attribute A contributed x to the particular candidate's ranking score proportion to the sum of the scores. Since the ranking vector τ is produced by sorting the score vector s in descending order, the absolute value of s_i does not affect the ranking result in the end. Hence, we can convert each s_i

$$s'_i = \frac{s_i}{\sum s_i} \tag{3}$$

s'_j is the "payout" that candidate i has in the case of algorithmic rankers, instead of s_j. Consequently, the "payout" of each attribute needs to be standardized in the same way. For the case of s'_i, we consider attribute x_j contributes $\frac{\beta_j x_{ij}}{\sum s_i}$ to candidate i. In such a way, we obtain the standardized Shapley values matrix C, each row of C adds up to the standardized Shapley values of the candidate, and each column of C adds up to the standardized Shapley values of the attribute. Compared with the common definition of the Shapley value contribution matrix for a regression model, the standardized Shapley values matrix for an algorithmic ranker shares some of the characteristics, such as symmetry, dummy, and additivity. But in the Shapley value contribution matrix, changes in a single data input do not necessarily affect the other data. In the standardized Shapley values matrix, any changes in the data require recalculating of the matrix since the change of the score value from one candidate affects the scaling parameter $\sum s_i$ and the entire standardized Shapley values matrix. Each row of the Shapley value matrix is independently calculated from the rest, hence an instance-wise explanation (i.e., not all the cells in the matrix need to be recalculated when a new entity join the data). Our standardized Shapley values matrix is a list-wise explanation since each cell in the matrix describes exactly how much leverage a certain attribute provides the candidate among the entire ranking space.

Rank-relevance Shapley value. Another way to define the "payout" for algorithmic rankers is the reverse of candidates' rank positions or *relevance*. For instance, the top-1 candidate has higher relevance than the top-2, the top-3, and so on. Such relevance is based on the ranking τ rather than ranking score s. We propose that for an algorithmic ranker, "payout" of x means attribute A contributed x to the particular candidate's relevance in total ranking positions. We can calculate relevance r as:

$$r_i = 1 - \frac{\tau_i}{\tau_{max}} \tag{4}$$

τ_{max} is the largest rank position value in τ. Let α_i be the parameter that projects the candidate's standardized "payout" s'_i to r_i.

$$r_i = s'_i \alpha_i \tag{5}$$

and we consider attribute x_j contributes $\beta_j x_j \alpha_i$ to candidate i. In equation (5), we assume that the relevance of candidates decreased linearly along the ranking. Such an assumption may not apply to certain algorithmic rankers such as search engines or university rankings. For those ranking, the top-ranked items attract exponentially more attention than the lower-ranked items. Hence we modify equation (5) with an additional parameter p to amplify the entity's relevance.

$$r_i = (1 - \frac{\tau_i}{\tau_{max}})^p \tag{6}$$

In our interface design, such parameter p is calibrated by user input since different users may perceive the amplifying effect differently.

4 COMPARE ALGORITHMIC RANKER EXPLAINERS

In this section, we describe the visualization interface used to compare three explainers, the Shapley values, standardized Shapley values, and the rank-relevance Shapley values side-by-side. We also demonstrate how meaningful inferences can be drawn from the analysis and visualizations.

4.1 Visualization Interface

Our visualization interface comprises the following views: i) **Data distribution view** (Fig. 1.a1) shows a scatter plot with marginal distribution to visualize the data simulated using user-specified inputs. The plot gets updated with different parameter inputs specifying the distributions and correlations. ii) **Ranking view** (Fig. 1.a2) shows a stability plot [21] for visualizing associations between scores and rankings. The stability plot can be read as a scatter plot, with a score on the y-axis and rank position on the x-axis. The stability plot shows that the score may decrease faster in certain rank ranges and slower in others; the former rank range is considered more stable than the latter. iii) **Ranker explanation view** shows a boxplot (with data point strips) to show the distribution of average contribution for Shapley values (Fig. 1.b1), standardized Shapley values (Fig. 1.b2), and rank-relevance Shapley values (Fig. 1.b3) during multiple (e.g., 200) data simulations. Note that we provide total and group-wise box plots to help observe group-wise difference patterns. Such a box plot can be viewed as the generalized bar plot from the SHAP package. The bar plot can only handle one simulation, and our plots show the variations across multiple simulations. The interactive **visualization interface (https://hilda23-ranking-shapley.streamlit.app)** incorporates the aforementioned views.

4.2 Empirical Observations

In various simulation testing under different data distribution parameters, weight ratios, and rank ranges, the explanations from Shapley values (Fig.1 b1) and standardized Shapley values (Fig. 1 b2)

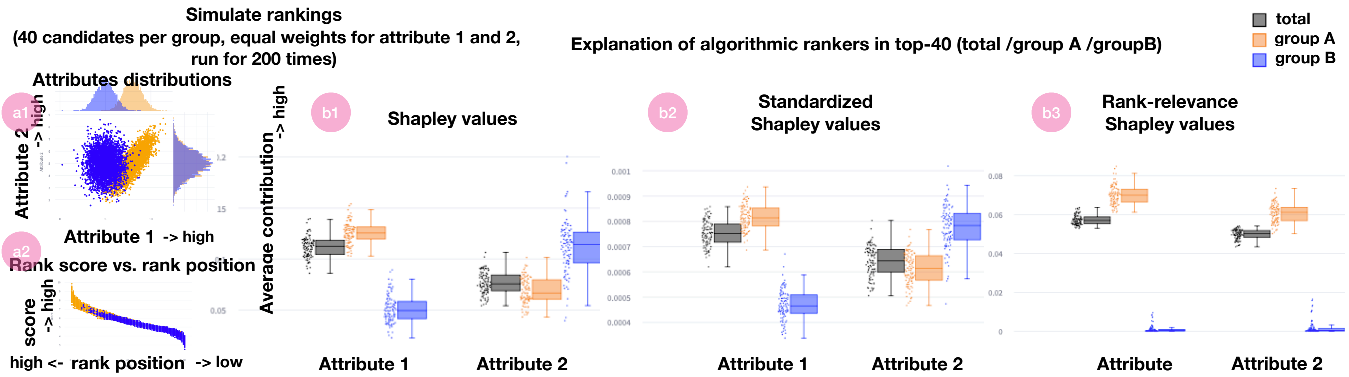

Figure 1: Sensitivity experiment of ranking explanations. a1) group A has a distributional advantage compared to B. a2) group A's advantage in attribute 1 leads to an advantage in ranking. b1-b2) Shapley values and Standardized Shapley values indicate that attribute 1 is more important in the top-40 range than 2. And group A has more advantages in the top-40 in attribute 1 but less in attribute 2. b3) Rank-relevance Shapley values indicate that group B has minimal importance in the top-40.

are similar. It helps users who understand Shapley values familiarize themselves with our new methods. We selected certain arbitrary parameters for an experiment to demonstrate what kind of inferences can be drawn from the explanation. All parameters can be changed in the interface. We simulated two datasets for two groups, group A and B; both have attributes x_1 and x_2 that are sampled for a multi-variate Gaussian distribution. We simulated 200 times for datasets A and B, each with 40 candidates. We assume A has a distributional advantage against B. In Figure 1 a1, we let the mean of A's attribute 1 (μ = 8, σ=1) higher than the mean of B's attribute 2 (μ = 5, σ=1) during sampling. We set a positive correlation (0.8) between A's attributes 1 and 2 but no correlation for B. In such a way, A's attribute 1 is more likely to have a higher score than B's attribute 1, which leads to higher ranking scores and higher rank positions. We set 0.5 scoring weight for both attributes 1 and 2 to generate the ranking score and the ranking. In Figure 1 a2, we show the resulting score and ranking. The plot shows that group A (yellow) appears more in the top positions due to the distributional advantage, as expected. And the simulation variance does not affect the outcome.

Since we constructed the input data and distributional discrepancy between groups A and B, we expect our explanation methods to detect such patterns as a sanity check and provide additional information. In the explanation view (Fig. 1 b1,b2,b3), all three explanation methods detect that attribute 1 contributes more than 2 in the total top-40 candidates. Among the 40 candidates from group A, attribute 1 contributes more than 2. Among the 40 candidates from group B, attribute 2 contributes more than 1. The reason is that candidates from group B have, in general, lower values for attribute 1; hence need to boost up their values for attribute 2 to get in the top-40. Note that for rank-relevance Shapley values, we use a high-rank relevance amplifying parameter (p=20), and the contribution from attributes 1 and 2 for group B becomes nearly zero (Fig. 1 b3). This is because high-ranked positions are mostly occupied by candidates from group A. Hence, we are able to validate our visualization and explanation methods using two attributes and two groups for generating interpretable inferences regarding the sensitivity of rankings with respect to attributes.

5 CONCLUSION AND FUTURE WORK

In this work, we transformed Shapley values for algorithmic rankers, specifically for algorithms using a linear weighted summation. We built an interface for interactive sensitivity simulation testing our proposed Shapley values and validating the inferences. Currently, the synthetic data are limited to Gaussian distribution and two attributes. The algorithmic rankers are also limited to linear weighted summation. We plan to allow various statistical distribution selections (e.g., Poisson, Bernoulli) and increase the number of attributes from two to unlimited to better model the real-world datasets. Although our experiment shows that one can draw inferences about group-wise unfair patterns from the explanation methods, we need to conduct more experiments and compare them to existing fairness metrics.

ACKNOWLEDGMENTS

This work is funded in part by NSF grant number 1928627. We are grateful to Prof. Julia Stoyanovich from New York University for the productive discussions about explainable rankings.

REFERENCES

[1] Kjersti Aas, Martin Jullum, and Anders Løland. 2021. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence* 298 (2021), 103502.

[2] Hadis Anahideh and Nasrin Mohabbati-Kalejahi. 2022. Local Explanations of Global Rankings: Insights for Competitive Rankings. *IEEE Access* 10 (2022), 30676–30693. https://doi.org/10.1109/ACCESS.2022.3159245

[3] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*. 89–96.

[4] Abraham Gale and Amelie Marian. 2019. Metrics for Explainable Ranking Functions. *International Workshop on ExplainAble Recommendation and Search EARS'19, July 25, 2019, Paris, France.* https://ears2019.github.io/Gale-EARS2019.pdf

[5] Abraham Gale and Amélie Marian. 2020. Explaining Monotonic Ranking Functions. *Proceedings of the VLDB Endowment* 14, 4 (2020), 640 – 652. http://vldb.org/pvldb/vol14/p640-marian.pdf

[6] Christian Genest, Michel Gendron, and Michaël Bourdeau-Brien. 2009. The advent of copulas in finance. *The European journal of finance* 15, 7-8 (2009), 609–618.

[7] Paolo Giudici and Emanuela Raffinetti. 2021. Shapley-Lorenz eXplainable Artificial Intelligence. *Expert Systems with Applications* 167 (April 2021), 114104. https://doi.org/10.1016/j.eswa.2020.114104

[8] I Al Hazwani, Jenny Schmid, Madhav Sachdeva, and Jürgen Bernard. 2022. A Design Space for Explainable Ranking and Ranking Models. *arXiv preprint*

arXiv:2205.15305 (2022).

[9] Iain Macdonald and Paul Strachan. 2001. Practical Application of Uncertainty Analysis. *Energy and Buildings* (2001).

[10] Christoph Molnar. 2022. *Interpretable Machine Learning* (2 ed.). https://christophm.github.io/interpretable-ml-book

[11] Saman Razavi, Anthony Jakeman, Andrea Saltelli, Clémentine Prieur, Bertrand Iooss, Emanuele Borgonovo, Elmar Plischke, Samuele Lo Piano, Takuya Iwanaga, William Becker, et al. 2021. The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software* 137 (2021), 104954.

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[14] SAT annual report 2023. SAT annual report. https://reports.collegeboard.org/.

[15] SHAP python package 2023. SHAP python package. https://github.com/slundberg/shap.

[16] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM, New York NY USA, 180–186. https://doi.org/10.1145/3375627.3375830

[17] Mona Sloane, Ian René Solano-Kamaiko, Jun Yuan, Aritra Dasgupta, and Julia Stoyanovich. 2023. Introducing contextual transparency for automated decision systems. *Nature Machine Intelligence* (2023), 1–9.

[18] US Census data 2023. US Census data. https://data.census.gov/.

[19] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 56–65.

[20] Eyal Winter. 2002. The shapley value. *Handbook of game theory with economic applications* 3 (2002), 2025–2054.

[21] Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, Hv Jagadish, and Gerome Miklau. 2018. A Nutritional Label for Rankings. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, Houston TX USA, 1773–1776. https://doi.org/10.1145/3183713.3193568

[22] Jun Yuan, Julia Stoyanovich, and Aritra Dasgupta. 2022. Rankers, Rankees, & Rankings: Peeking into the Pandora's Box from a Socio-Technical Perspective. *arXiv preprint arXiv:2211.02932* (2022).

[23] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A Support Vector Method for Optimizing Average Precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 271–278.

[24] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2021. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000* (2021).

[25] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).