Responsible Machine Learning

Lecture 14: Algorithmic Fairness in the Real World - Part 2

CS 4973-05

Fall 2023

Instructors: Avijit Ghosh
ghosh.a@northeastern.edu
Northeastern University, Boston, MA

Northeastern University

Background

Research Questions

Awareness vs Unawareness

Continuous Fairness

Broader Impact

There exist some real world problems...

Algorithm and metric design

- Intersectionality of bias
- Model bias is not necessarily a data problem

Runtime challenges

- Missing demographic information
- Adversarial attackers can make the algorithm more unfair
- Models may become unfair in a live deployment over time

Transparency and Accountability

- Not many transparent real-world audits
- Decisions are not always correlated with outcomes

Background

Research Questions

Awareness vs Unawareness

Continuous Fairness

Broader Impact

Research Questions

- **RQ1:** How does noise in demographic information as an input to a fair ML algorithm adversely impact the intended fairness of the outcomes for different subgroups?
- **RQ2:** How can fair ML models be attacked by adversarial actors to create even more unfairness?
- **RQ3:** In fair ML techniques that deliberately do not use protected attributes, how do their theoretical guarantees hold up in real life when compared against actual ground truth?
- **RQ4:** Do fair ML models, once deployed in a production system, continue to remain fair in the face of changing data and feature-output relationships? If so, how can such unfairness be measured and mitigated?

Research Questions

- **RQ1:** How does noise in demographic information as an input to a fair ML algorithm adversely impact the intended fairness of the outcomes for different subgroups?
- **RQ2:** How can fair ML models be attacked by adversarial actors to create even more unfairness?
- **RQ3:** In fair ML techniques that deliberately do not use protected attributes, how do their theoretical guarantees hold up in real life when compared against actual ground truth?
- **RQ4:** Do fair ML models, once deployed in a production system, continue to remain fair in the face of changing data and feature-output relationships? If so, how can such unfairness be measured and mitigated?

Background

Research Questions

Awareness vs Unawareness

Continuous Fairness

Broader Impact

Chapter 3

When Fair Classification Meets Noisy Protected Attributes

AIES 2023

Research Questions

- **RQ1:** How does noise in demographic information as an input to a fair ML algorithm adversely impact the intended fairness of the outcomes for different subgroups?
- **RQ2:** How can fair ML models be attacked by adversarial actors to create even more unfairness?
- **RQ3:** In fair ML techniques that deliberately do not use protected attributes, how do their theoretical guarantees hold up in real life when compared against actual ground truth?
- **RQ4:** Do fair ML models, once deployed in a production system, continue to remain fair in the face of changing data and feature-output relationships? If so, how can such unfairness be measured and mitigated?

Possible Mitigation to Protected Attribute Noise

- Use inferred attributes only when they are **extremely accurate** for all intersectional groups
- **Human-in-the-loop** solutions (privacy aware), for instance Project Lighthouse

Airbnb's Project Lighthouse

Possible Mitigation to Protected Attribute Noise

- Use inferred attributes only when they are **extremely accurate** for all intersectional groups
- **Human-in-the-loop** solutions (privacy aware), for instance Project Lighthouse

Expensive Solutions!

Airbnb's Project Lighthouse

Uncertainty Aware Algorithms

There is a newer class of fair algorithms that theoretically achieve fair predictions in spite of **partial or complete absence of protected attributes.**

How do uncertainty aware fair algorithms stack up against fair algorithms that require access to demographic attributes in a head-to-head comparison?

Case study: **fair classification.**

Terminology

- Unconstrained Classifiers
- Classically Fair Classifiers
- Noise-Tolerant Fair Classifiers
- Demographic-Blind Fair Classifiers

Unconstrained Classifiers

Unconstrained Classifiers do not have any fairness objectives and are solely optimized for accuracy.

- Logistic Regression (LR)
- Random Forest (RF)

LR and RF

- **Logistic Regression (LR)** is demographic-aware because it takes all features (including protected attributes) as model inputs at both train and test time, it is not designed to achieve any fairness criteria.
- **Random Forest (RF)** is an ensemble method for classification built out of decision trees. Like LR, we train RF classifiers on all input features including protected attributes.

Classically Fair Classifiers

Classically Fair Classifiers take protected attributes as input and attempt to achieve demographic fairness, via different methods:

Preprocessing:

- Sample Reweighting (SREW)
- Learned Fair Representation (LFR)

In-processing:

- Adversarial Debiasing (ADDEB)
- Exponentiated Gradient Reduction (EGR)
- Grid Search Reduction (GSR)

Post-processing:

- Calibrated Equalized Odds (CALEQ)
- Reject Option Classifier (ROC)

Sample Reweighting

• **Sample Reweighting (SREW)** is a pre-processing technique that takes each (group, label) combination in the training data and assigns rebalanced weights to them. The goal of this procedure is to remove imbalances in the training data, with the ultimate aim of ensuring fairness before the classifier is trained

Learned Fair Representation

- **Learned Fair Representation (LFR)** is a pre-processing technique that converts the input features into a latent encoding that is designed to represent the training data well while simultaneously hiding protected attribute information from the classifier

Adversarial Debiasing

- **Adversarial Debiasing (ADDEB)** is an in-process technique that trains a classifier to maximize accuracy while simultaneously reducing an adversarial network's ability to determine the protected attributes from the predictions.

EGR and GSR

- **Exponentiated Gradient Reduction (EGR)** is an in-process technique that reduces fair classification to a set of cost-sensitive classification problems, essentially treating the main classifier itself as a black box and forcing the predictions to be the most accurate under a given fairness constraint. In this case, the constraint is solved as a saddle point problem using the exponentiated gradient algorithm.

- **Grid Search Reduction (GSR)** uses the same set of cost-sensitive classification problems approach as EGR, except in this case the constraints are solved using the grid search algorithm.

CALEQ and ROC

- **Calibrated Equalized Odds (CALEQ)** is a post-processing technique that optimizes the calibrated classifier score output to find the probabilities that it uses to change the output labels, with an equalized odds objective.

- **Reject Option Classifier (ROC)** is a post-processing technique that swaps favorable and unfavorable outcomes for privileged and unprivileged groups around the decision boundaries with the highest uncertainty.

Note that the CALEQ and ROC algorithms have access to protected attributes at both train and test time, while the other classifiers only have access to protected attributes at training time.

Noise-Tolerant Fair Classifiers

Noise-Tolerant Fair Classifiers require access to protected attributes but account for uncertainty (noise) in the data.

- Modified Distributionally Robust Optimization (MDRO)
- Soft Group Assignments (SOFT)
- Private Learning (PRIV)

Noise Tolerant Classifiers

- **Modified Distributionally Robust Optimization (MDRO)** is an extension of the Distributionally Robust Optimization (DRO) algorithm that adds a maximum total variation distance in the DRO procedure. By assuming a noise model for the protected attributes, it aims to provide tighter bounds for DRO.

- **Soft Group Assignments (SOFT)** is a theoretically robust approach that first performs "soft" group assignments and then performs classification, with the idea being that if an algorithm is fair in terms of those robust criteria for noisy groups, then they must also be fair for true protected groups**.**

- **Private Learning (PRIV)** is an approach by that uses differential privacy techniques to learn a fair classifier while having partial access to protected attributes. The approach requires two steps. The first step is to obtain locally private versions of the protected attributes. Second, PRIV tries to create a fair classifier based on the private attributes. For this study, we select the privacy level hyperparameter to be a medium value (zero).

Demographic-Blind Fair Classifiers

Demographic-Blind Fair Classifiers aim to achieve fairness without requiring access to protected attributes at all.

- Adversarially Reweighted Learning (ARL)
- Distributionally Robust Optimization (DRO)

Demographic-Blind Classifiers

- **Adversarially Reweighted Learning (ARL)** harnesses non-protected attributes and labels by utilizing the computational separability of these training instances to divide them into subgroups, and then uses an adversarial reweighting approach on the subgroups to improve classification fairness.

- **Distributionally Robust Optimization (DRO)** is an algorithm that attempts to minimize the worst case risk of all groups that are close to the empirical distribution. In the spirit of Rawlsian distributive justice, the algorithm tries to control the risk to minority groups while being oblivious to their identities.

Experiments

Metrics

- **Accuracy:** $$\frac{\text{number of correct classifications}}{\text{test dataset size}}$$
- **Equal Odds Difference:** $$\frac{(FPR_{unpriv} - FPR_{priv}) + (TPR_{unpriv} - TPR_{priv})}{2}$$

Datasets

- **Public Coverage** - The task is to predict whether an individual (who is low income and not eligible for Medicare) was covered under public health insurance.
- **Employment** - The task is to predict whether an individual (between the ages of 16 and 90), is employed.
- **Law School Admissions** - The task is to predict whether a student was admitted to law school.
- **Diabetes** - The task is to predict whether a diabetes patient was readmitted to the hospital for treatment after 30 days.

Each dataset had binary sex as part of the input features.

Simulated Noise

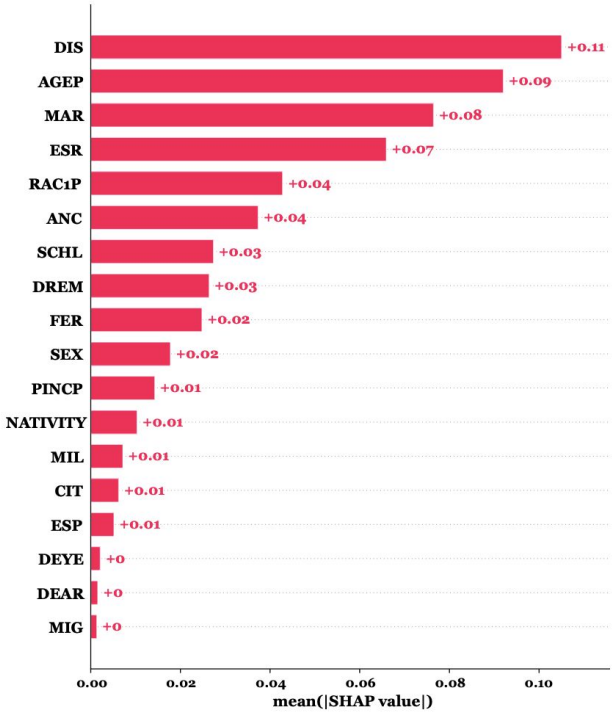
1. Randomly flip protected attribute labels (binary sex) in each dataset with a probability between 0.1 and 0.9 (noise).

2. Split the synthetically altered dataset 80:20 and train 14 algorithms on the training set using the noisy (flipped) labels.

3. Calculate accuracy and EOD, measuring EOD with the original sex labels.

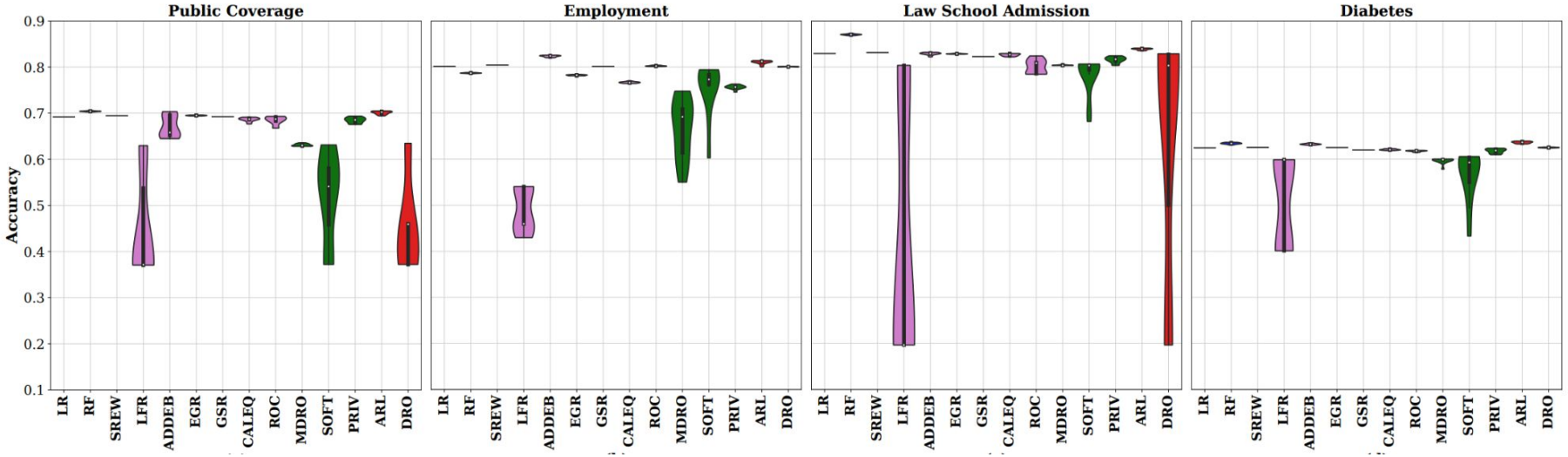
4. Repeat Steps 1-3 ten times for each noise value to ensure statistical power and metric stability.

Feature Importance

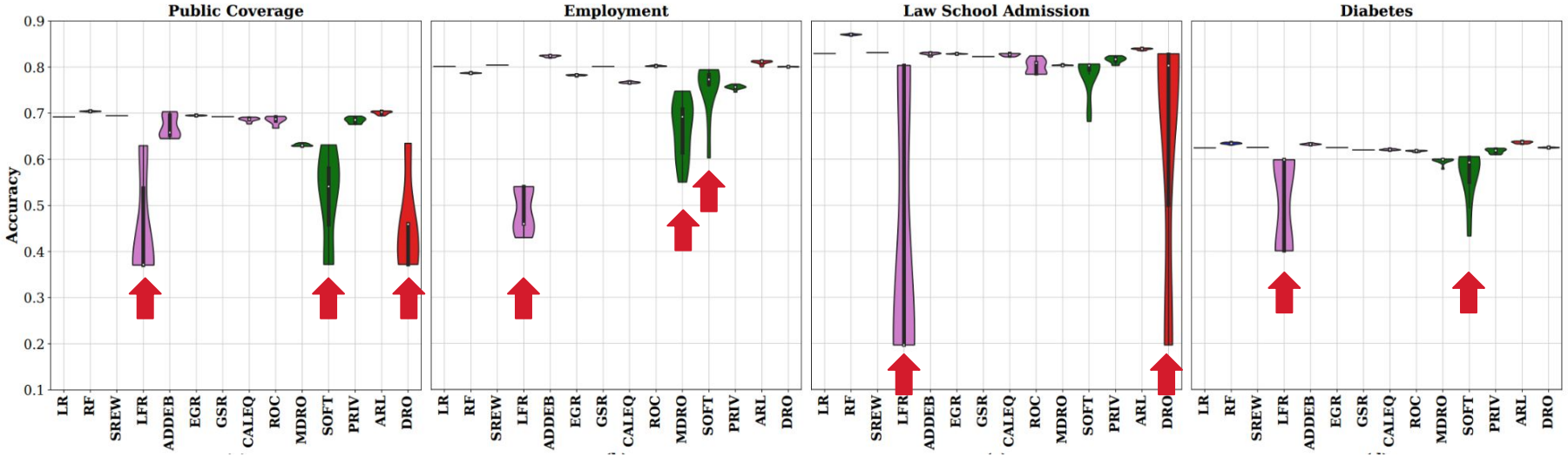


Feature importances are calculated for each classifier and each dataset with KernelSHAP.

Results: Stability - no noise

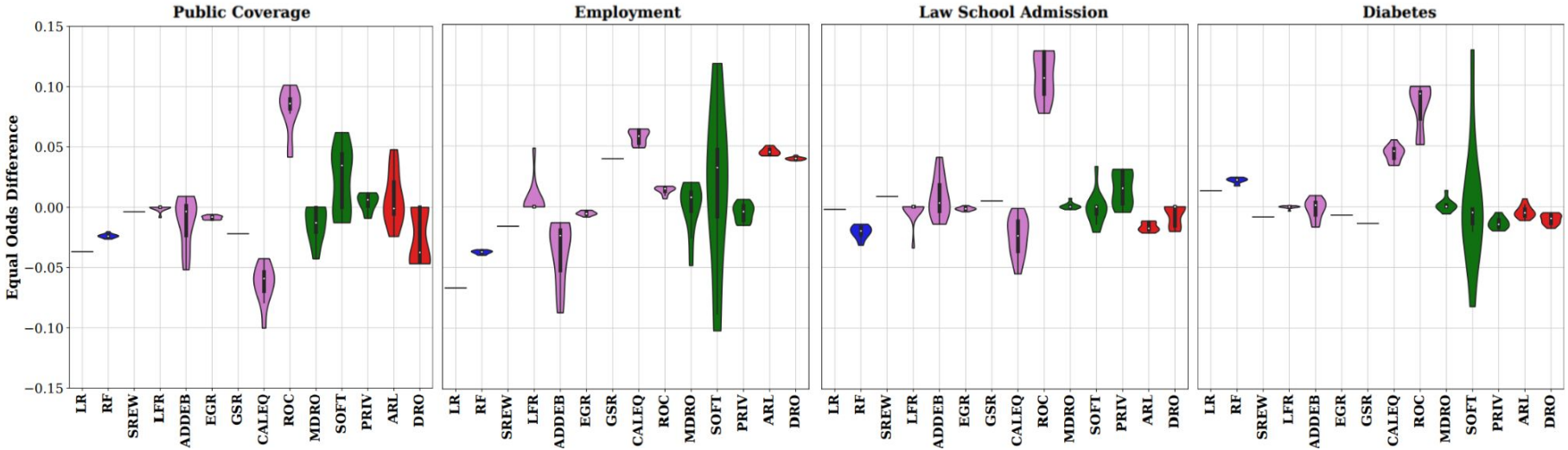


Results: Stability - no noise

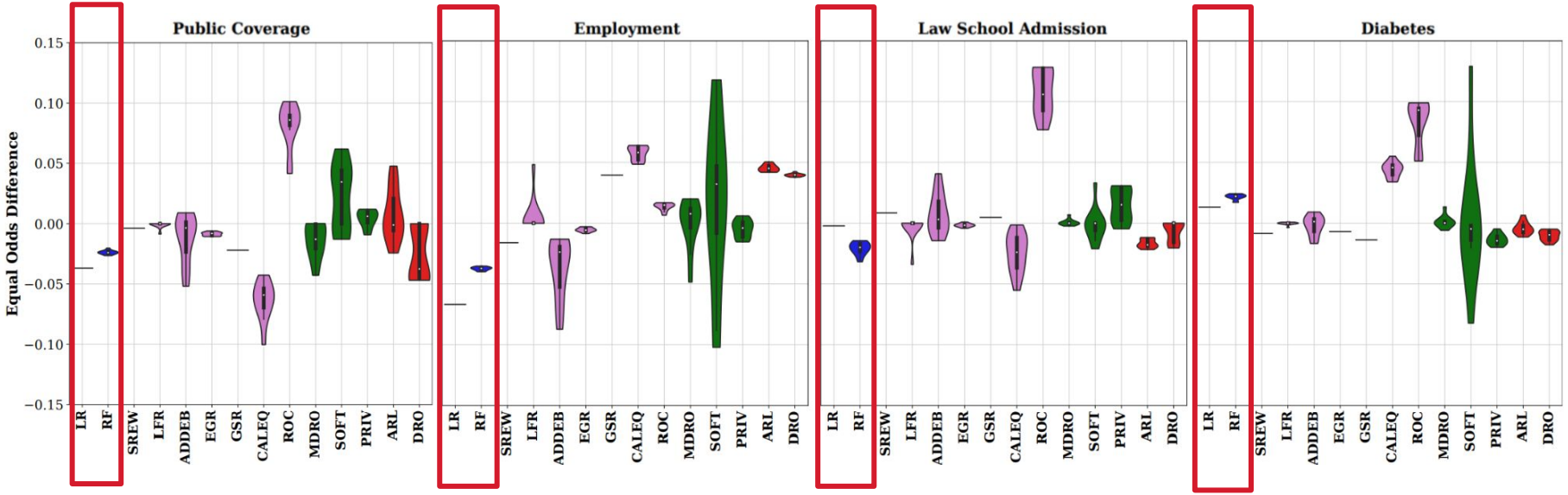


LFR, MDRO, SOFT, and DRO have unstable accuracy, the others seem to be pretty stable over different datasets

Results: Stability - no noise

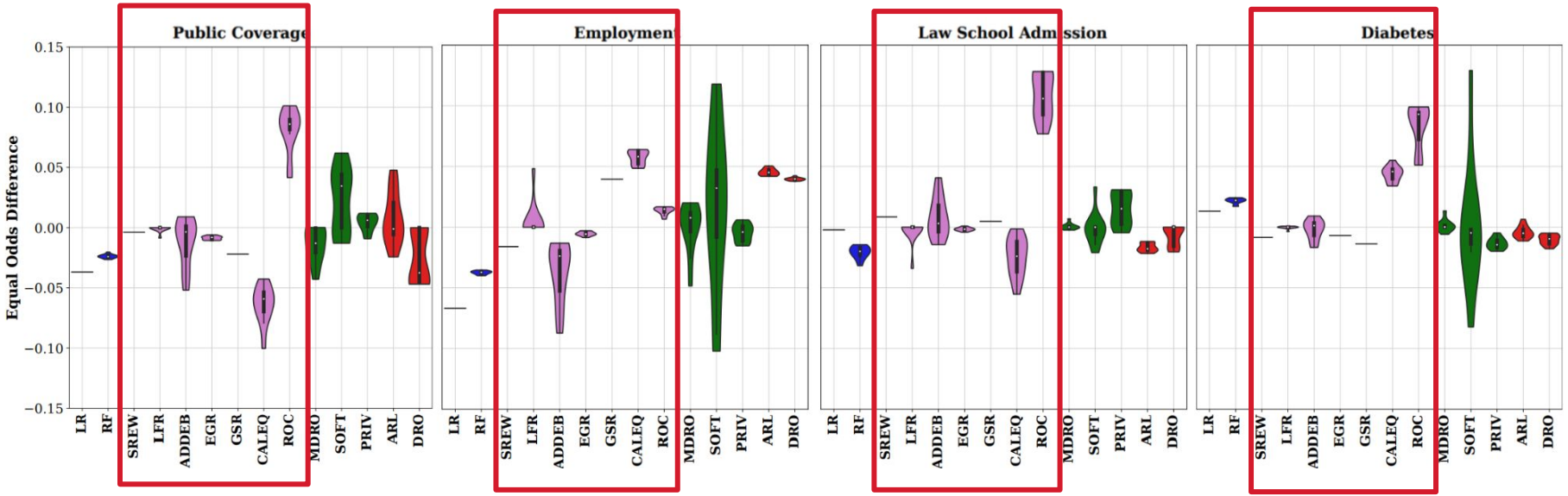


Results: Stability - no noise



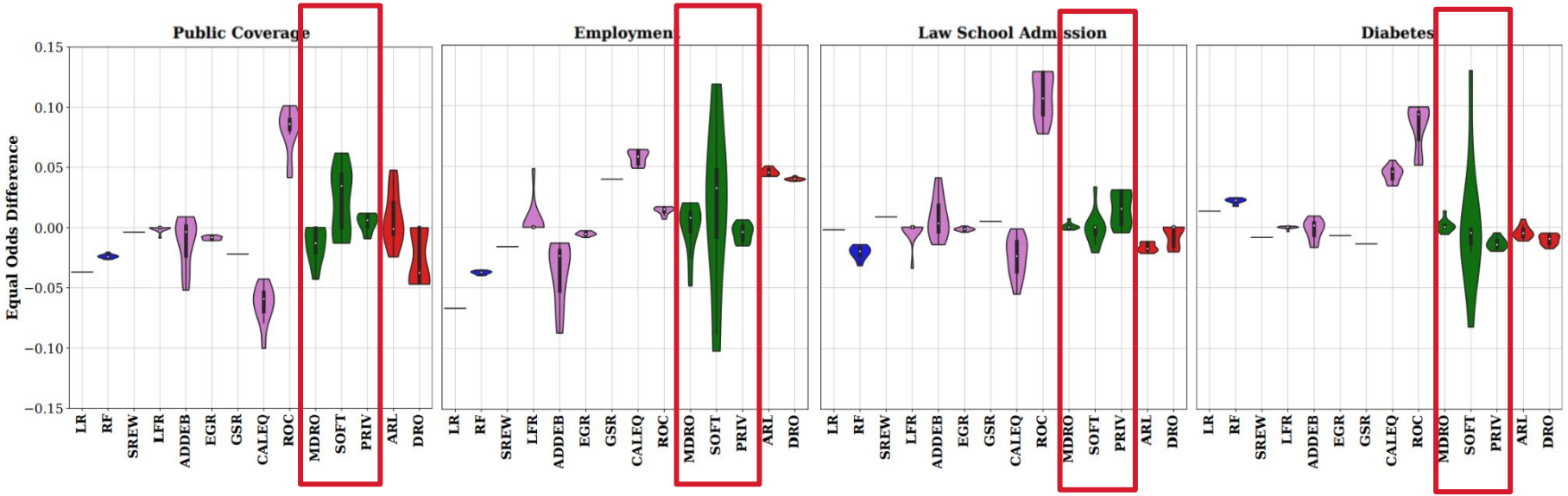
The **unconstrained classifiers** (LR and RF) were relatively stable and, in some cases, achieved roughly equalized odds

Results: Stability - no noise



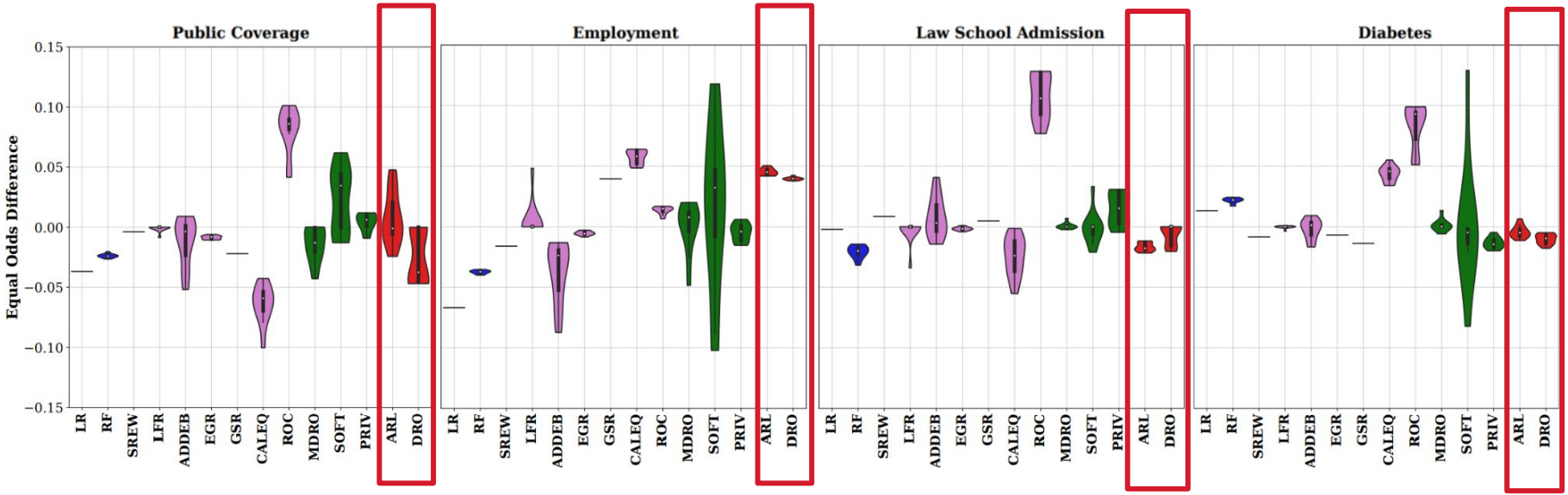
The **classical fair classifier** group contained the two least fair classifiers in these experiments (CALEQ and ROC), while the other pre-processing and in-process algorithms performed relatively better

Results: Stability - no noise



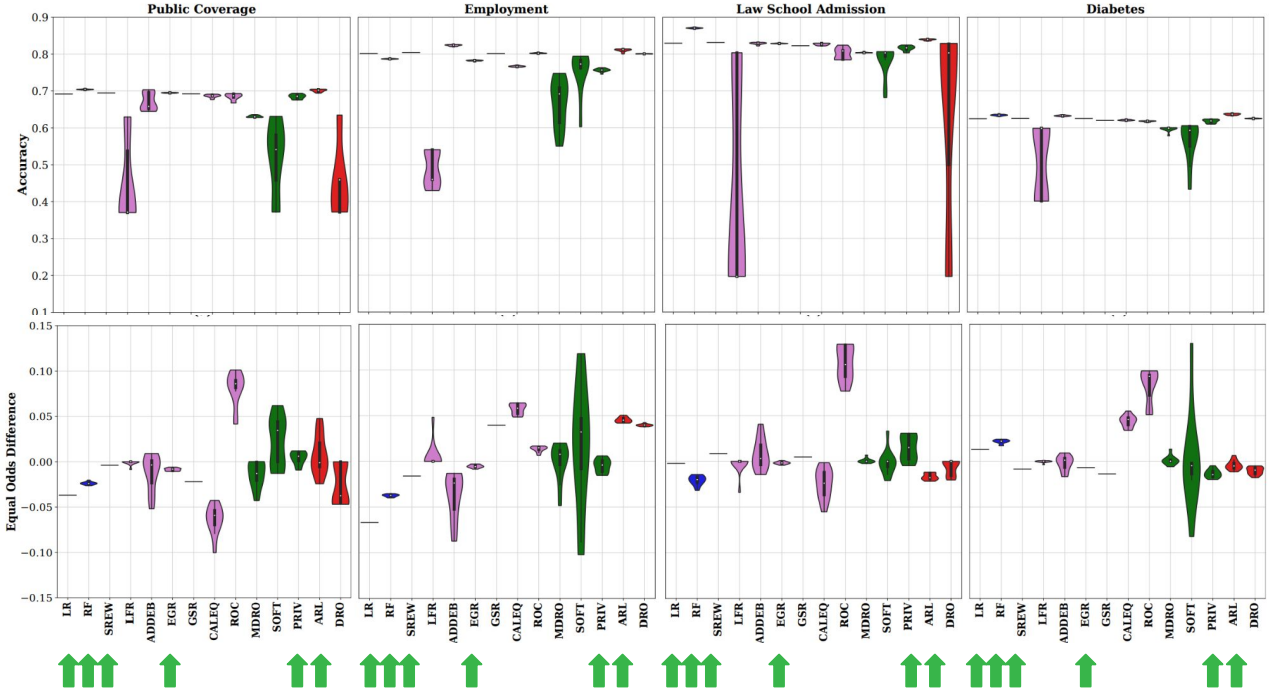
Among the **noise-tolerant fair classifiers**, Soft Group Assignment (SOFT) was unstable on three out of four dataset, while the other two classifiers (MDRO and PRIV) were relatively more stable and more fair.

Results: Stability - no noise



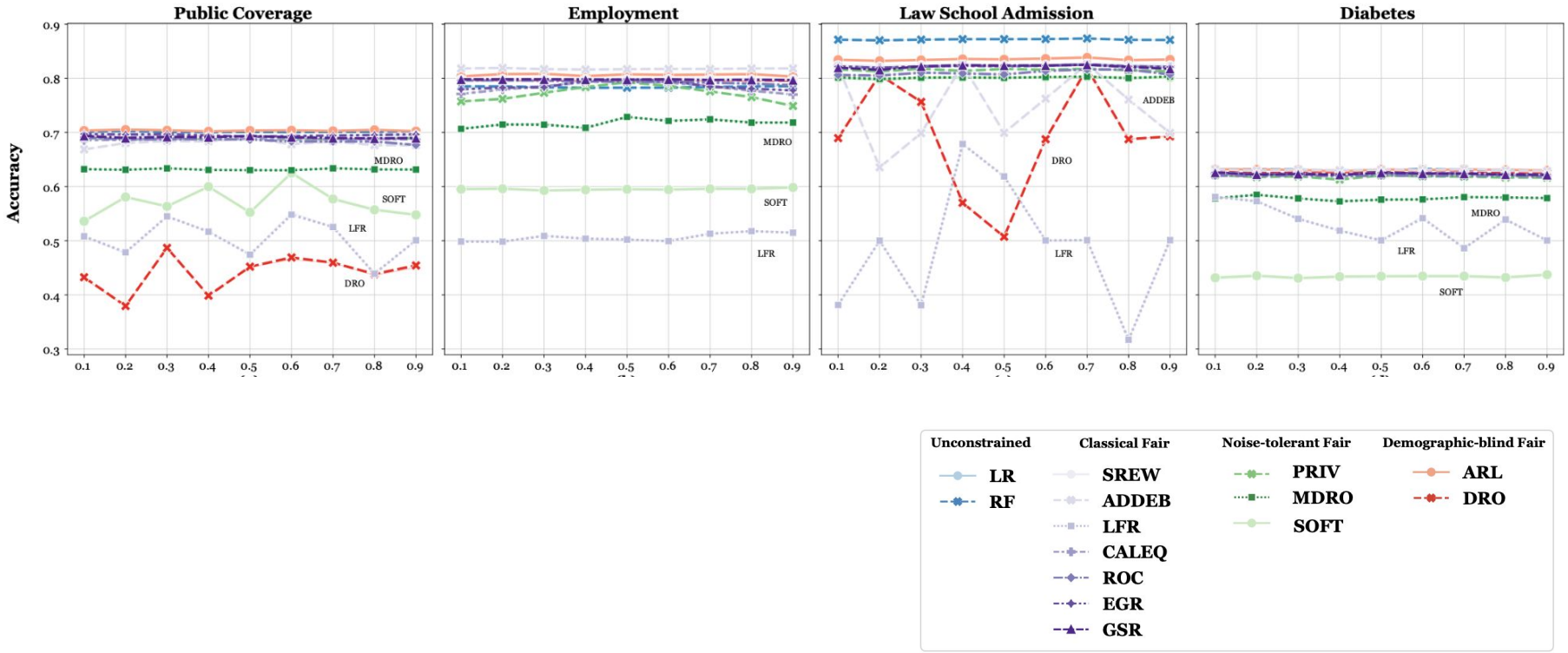
The two **demographic-blind fair classifiers** (ARL and DRO) were unstable on the Public Coverage dataset and did not achieve equalized odds on the Employment dataset. However, ARL and DRO were stable and fair on the remaining two datasets.

Results: Stability - no noise

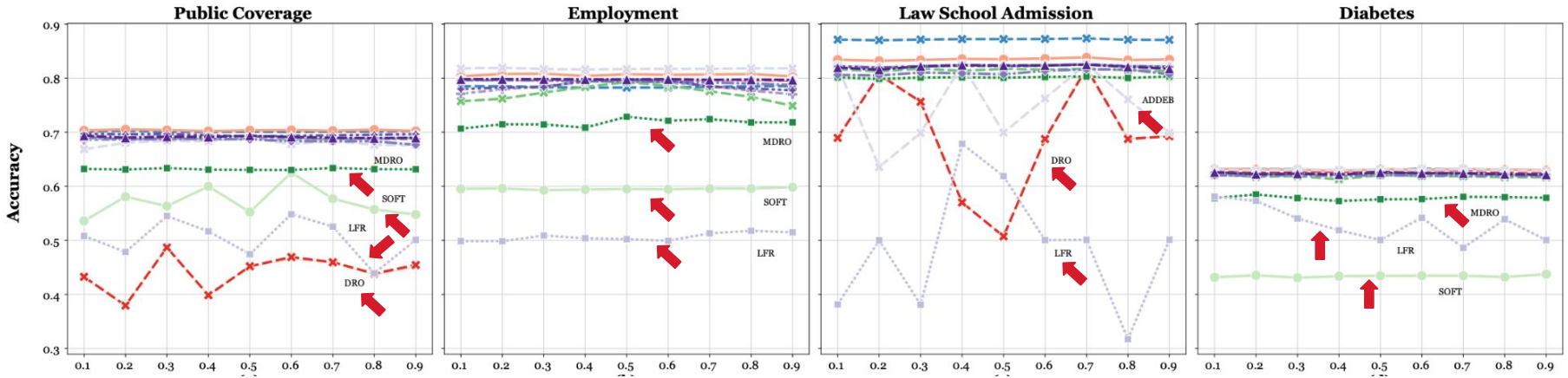


Six algorithms have both good performance and stable metrics

Results: Trends with Noise

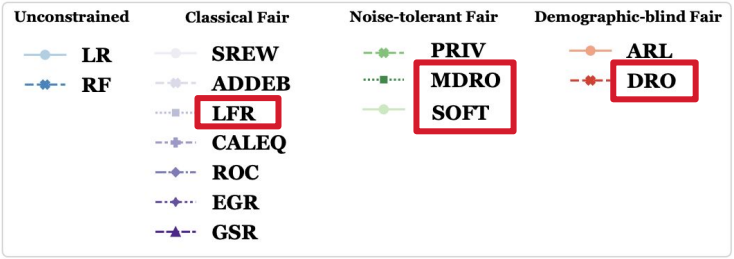


Results: Trends with Noise

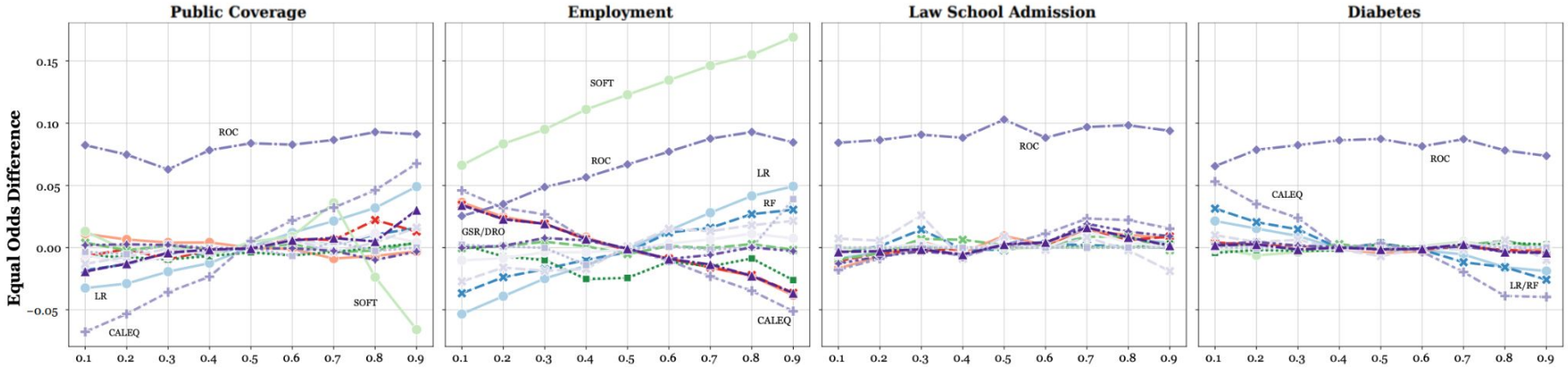


We observe that the LFR, MDRO, SOFT, and DRO classifiers had poor accuracy and noise dependent fluctuations.

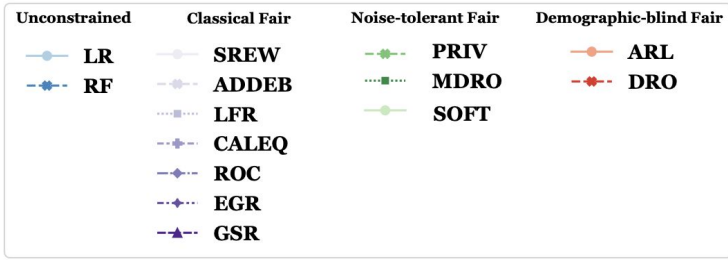
Remember that these were also the ones that had unstable accuracy, as we saw earlier!



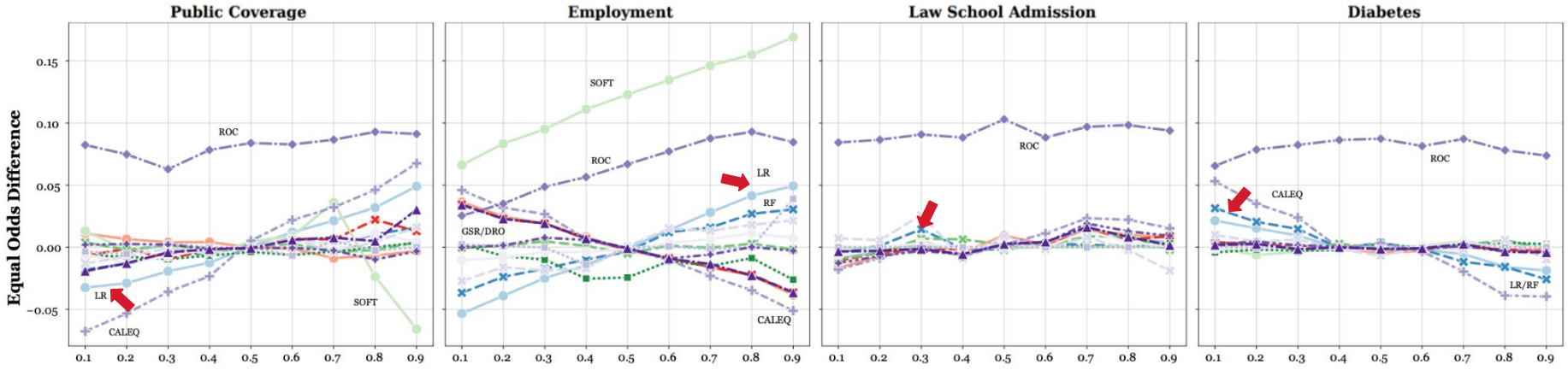
Results: Trends with Noise



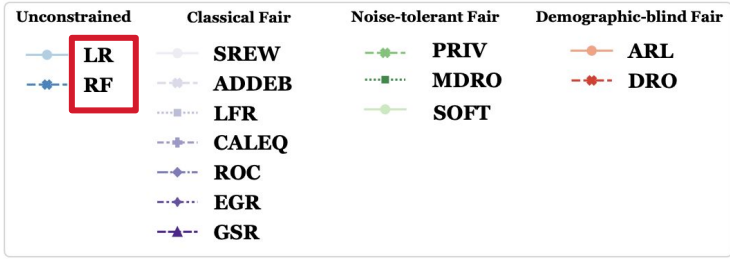
The Fairness performance trends are more complicated.



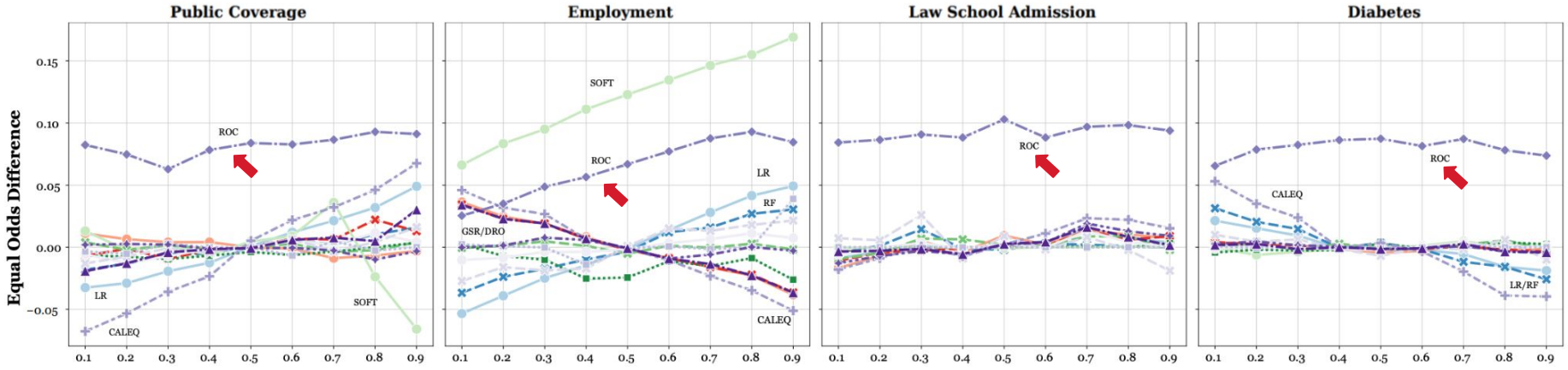
Results: Trends with Noise



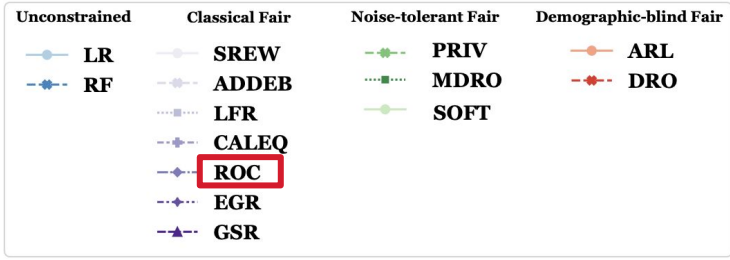
The unconstrained classifiers (LR and RF) moved in the same direction for every dataset, either rising or falling with noise.



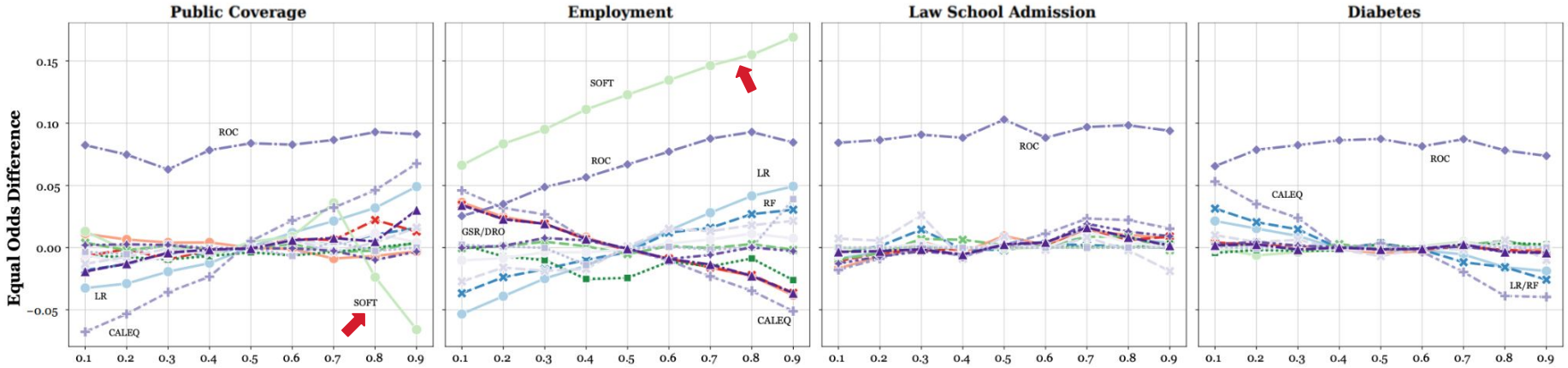
Results: Trends with Noise



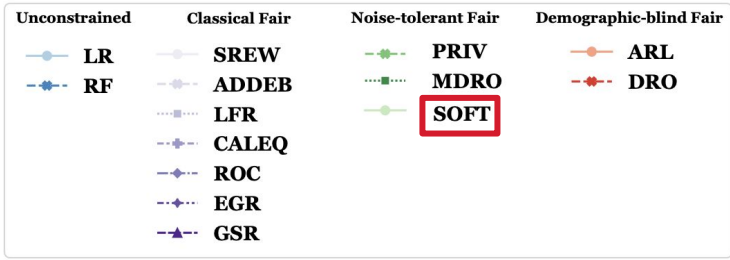
ROC generated unfair outputs over all four datasets, at every noise level.



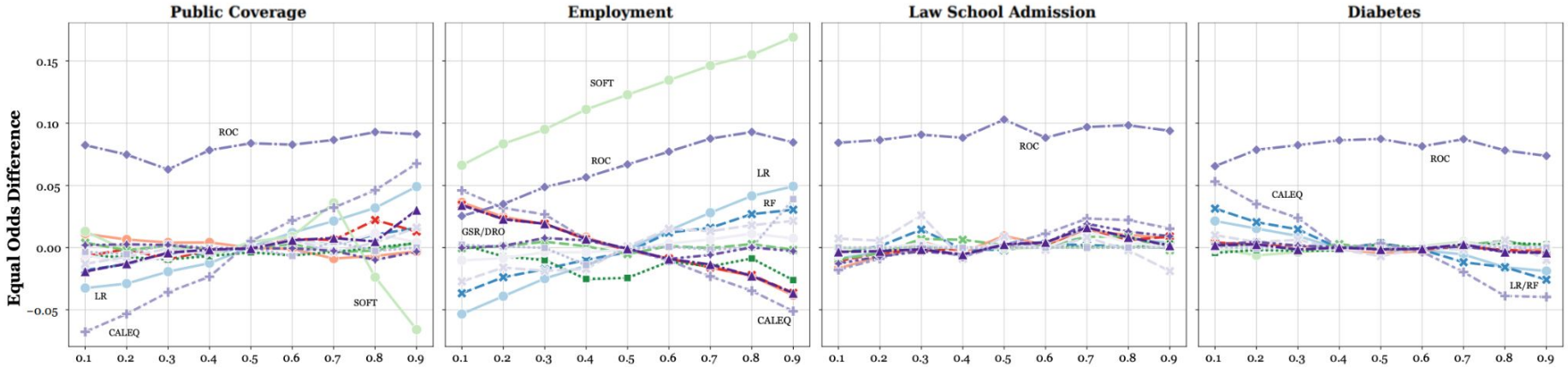
Results: Trends with Noise



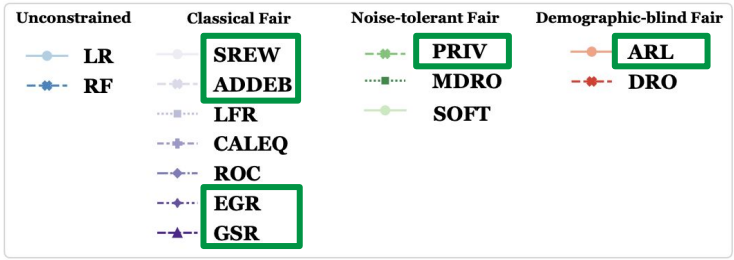
The SOFT classifier also exhibited some variable behavior: on the Employment dataset EOD rose with noise, and on the Public Coverage dataset it failed to achieve equal odds at higher noise levels.



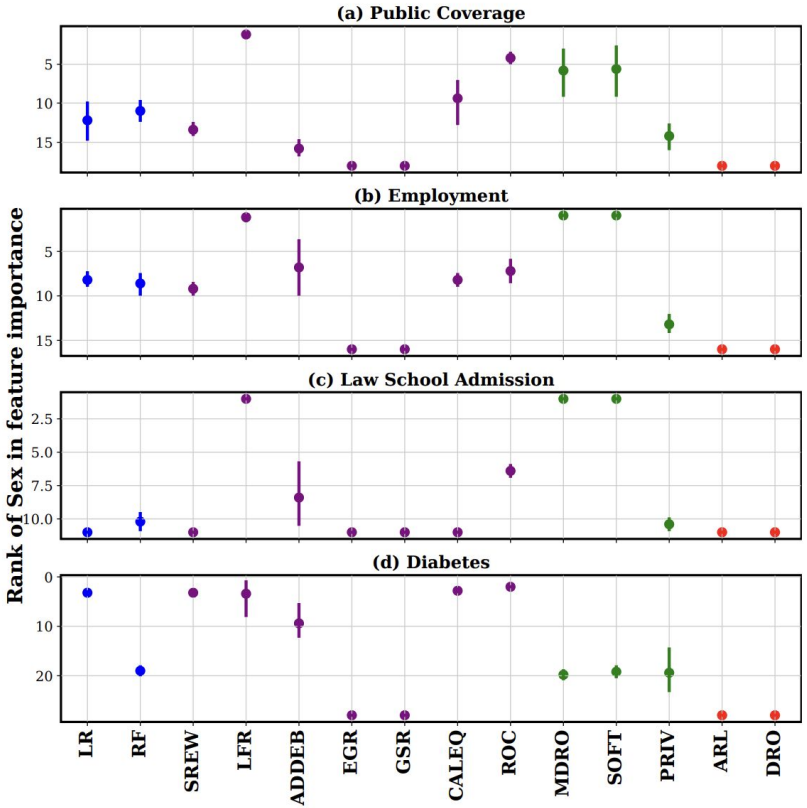
Results: Trends with Noise



Overall, unstable classifiers continued to have problems in the presence of noise, and additionally, unconstrained classifiers were also unreliable in the presence of noise.

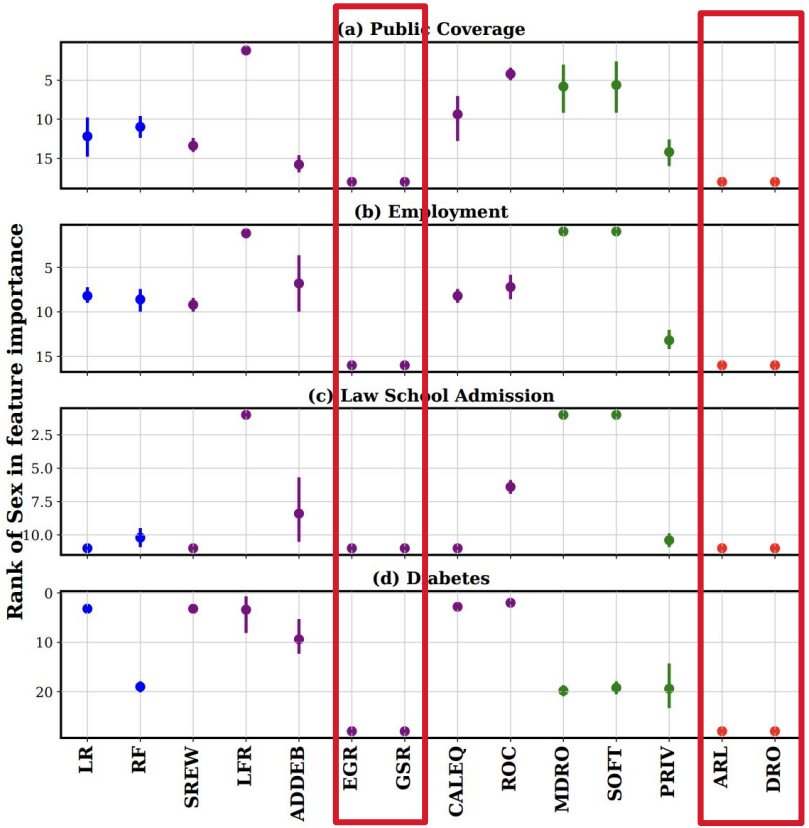


Results: Feature Importance



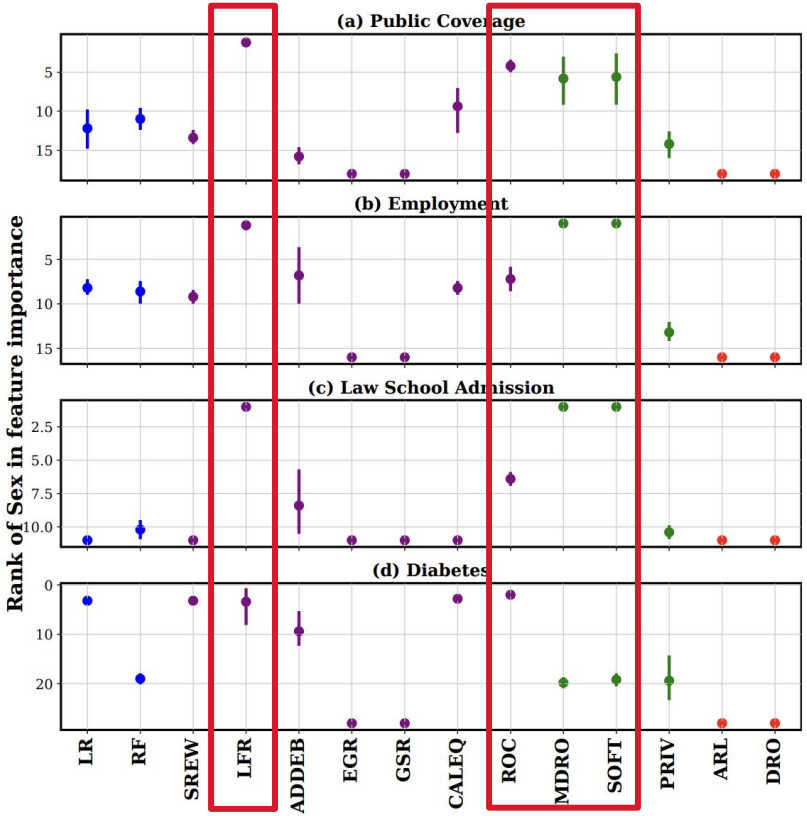
Now we look at how important the protected feature (sex) was to each of these models.

Results: Feature Importance



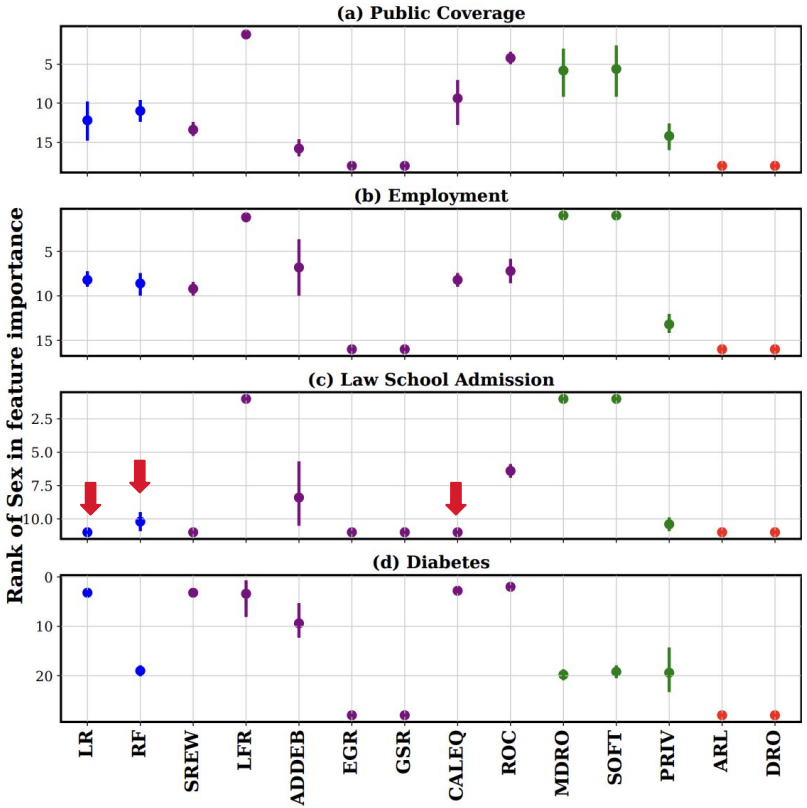
EGR and GSR are designed to only use protected attributes at train time but not test time, and ARL and DRO do not use protected attributes at any point.

Results: Feature Importance



Four of the classifiers that exhibited consistently poor performance—LFR, MDRO, and SOFT, and ROC —learned to weight the sex feature higher than other features, which may point to the root cause of their accuracy and fairness issues.

Results: Feature Importance



The two unconstrained classifiers (LR and RF), and one classically fair classifier, CALEQ, exhibited changing EOD with noise levels in three out of four datasets, but not for Law School Admissions, and the feature importance backs up this result.

Conclusion

Algorithm	Relative Metric performance	Baseline Stability	Robustness to Noise
LR	✅	✅	❌
RF	✅	✅	❌
SREW	✅	✅	✅
ADDEB	✅	❌	✅
LFR	❌	❌	❌
CALEQ	❌	❌	❌
ROC	❌	❌	❌
EGR	✅	✅	✅
GSR	❌	✅	✅
PRIV	✅	✅	✅
MDRO	❌	❌	✅
SOFT	❌	❌	❌
ARL	✅	✅	✅
DRO	❌	❌	❌

Conclusion

Algorithm	Relative Metric performance	Baseline Stability	Robustness to Noise
LR	✅	✅	❌
RF	✅	✅	❌
SREW	✅	✅	✅
ADDEB	✅	❌	✅
LFR	❌	❌	❌
CALEQ	❌	❌	❌
ROC	❌	❌	❌
EGR	✅	✅	✅
GSR	❌	✅	✅
PRIV	✅	✅	✅
MDRO	❌	❌	✅
SOFT	❌	❌	❌
ARL	✅	✅	✅
DRO	❌	❌	❌

Conclusion

- Four fair classifiers performed consistently well across metrics: Sample Reweighting (SREW), Exponentiated Gradient Reduction (EGR), Private Learning (PRIV), and Adversarially Reweighted Learning (ARL).
- Overall findings are a mixed bag: Some classical fair classifiers may perform unexpectedly well in the face of noise, and some theoretically noise resistant algorithms do not perform well.
- Regardless, demographic-blind fair classifiers like ARL could achieve fairness for real-world disadvantaged groups under ecological conditions.
- While this exercise gives some confidence in using demographic-blind classifiers, we still need to continuously check model health in real time.

DEEP DIVE: Sample Reweighting

Table 1 Sample relation for the job-application example

Sex	Ethnicity	Highest degree	Job type	Class
M	Native	H. school	Board	+
M	Native	Univ.	Board	+
M	Native	H. school	Board	+
M	Non-nat.	H. school	Healthcare	+
M	Non-nat.	Univ.	Healthcare	−
F	Non-nat.	Univ.	Education	−
F	Native	H. school	Education	−
F	Native	None	Healthcare	+
F	Non-nat.	Univ.	Education	−
F	Native	H. school	Board	+

To compensate for the bias, we will assign lower weights to objects that have been deprived or favored. Every object X will be assigned weight:

$$W(X) := \frac{P_{exp}(S = X(S) \wedge Class = X(Class))}{P_{obs}(S = X(S) \wedge Class = X(Class))} ;$$

i.e., the weight of an object will be the expected probability to see an instance with its sensitive attribute value and class given independence, divided by its observed probability.

Example 3 Consider again the dataset in Table 1. The weight for each data object is computed according to its S- and $Class$-value. We calculate the weight of a data object with $X(S) = f$ and $X(Class) +$ as follows. We know that 50% objects have $X(S) = f$ and 60% objects have $Class$-value $+$, so the expected probability of the object should be:

$$P_{exp}(Sex = f \ \wedge \ X(Class) \ = \ +) = 0.5 \times 0.6 = 30\%$$

but its actually observed probability is 20%. So the weight $W(X)$ will be:

$$W(X) = \frac{0.5 \times 0.6}{0.2} = 1.5 \ .$$

Similarly, the weights of all other combinations are as follows:

$$W(X) := \begin{cases} 1.5 & \text{if } X(Sex) = f \text{ and } X(Class) \ = \ + \\ 0.67 & \text{if } X(Sex) = f \text{ and } X(Class) \ = \ - \\ 0.75 & \text{if } X(Sex) = m \text{ and } X(Class) \ = \ + \\ 2 & \text{if } X(Sex) = m \text{ and } X(Class) \ = \ - \end{cases}$$

Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. Knowledge and information systems

DEEP DIVE: Exponentiated Gradient Reduction

- This approach poses Fair Learning as a constrained optimization problem: minimize the empirical error, subject to linear constraints of the fairness (e.g., TPR difference, demographic parity).
- Solve the constrained optimization as a **cost-sensitive** classification problem. Specifically, the Lagrangian multiplier of the fairness constraint is obtained using the exponentiated gradient algorithm (Kivinen & Warmuth, 1997). The algorithm is terminated as soon as the accuracy falls below a specified threshold.
- Obtain a **randomized classifier**, which implies they will create multiple base estimators.
- Note that it can potentially solve more than one constraint at a time, but with accuracy tradeoff.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In International Conference on Machine Learning.

DEEP DIVE: Private Learning

- Say Z is a differentially private version of A, the protected feature.
- They start by dividing the data set S into two equal parts S1 and S2. The first step is to learn an approximately non-discriminatory predictor with respect to Z on S1 via the reductions approach of (Agarwal et al., 2018). This predictor has low error, but may be highly discriminatory.
- The aim of the second step is to produce a final predictor Y that corrects for this discrimination, without increasing its error by much. They modify the post-processing procedure of (Hardt et al., 2016) to get non-discrimination with respect to A directly for the derived predictor Ye = f(Y , Z ˆ). The predictor in the second step does use Z, however with a careful analysis they are able to show that it indeed guarantees non-discrimination with respect to A.

built on this data is differentially private. Formally a locally $\epsilon-$differentially private mechanism Q is defined as follows:

Definition 2. *Q is $\epsilon-$differentially private if (Duchi et al. (2013)):*

$$\max_{z,a,a'} \frac{Q(Z=z|a)}{Q(Z=z|a')} \leq e^{\epsilon}$$

The mechanism we employ is the randomized response mechanism (Warner, 1965; Kairouz et al., 2014):

$$Q(z|a) = \begin{cases} \frac{e^{\varepsilon}}{|\mathcal{A}|-1+e^{\varepsilon}} := \pi & \text{if } z = a \\ \frac{1}{|\mathcal{A}|-1+e^{\varepsilon}} := \bar{\pi} & \text{if } z \neq a \end{cases}$$

The choice of the randomized response mechanism is motivated by its optimality for distribution estimation under LDP constraints (Kairouz et al., 2014; 2016)

Hussein Mozannar, Mesrob Ohannessian, and Nathan Srebro. 2020. Fair learning with private demographic data. In International Conference on Machine Learning.

DEEP DIVE: Adversarially Reweighted Learning

- Adversarially Reweighted Learning (ARL) is an optimization approach that leverages the notion of computationally-identifiable errors through an adversary $f_\varphi(X, Y)$ to improve worst-case performance over unobserved protected groups S.
- A minimax game between a learner and adversary: Both learner and adversary are learnt models, trained alternatively. The learner optimizes for the main classification task, and aims to learn the best parameters θ that minimizes expected loss. The adversary learns a function mapping $f_\varphi : X \times Y \rightarrow [0, 1]$ to computationally-identifiable regions with high loss, and makes an adversarial assignment of weight vector $\lambda_\varphi : f_\varphi \rightarrow R$ so as to maximize the expected loss.

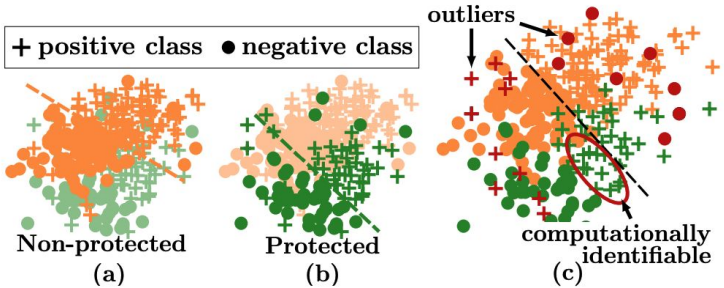


Figure 1: Computational-identifiability example

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. 2020. Fairness without demographics through adversarially reweighted learning. Advances in neural information processing systems.

Background

Research Questions

Story so far

Awareness vs Unawareness

Continuous Fairness

Broader Impact

Chapter 4

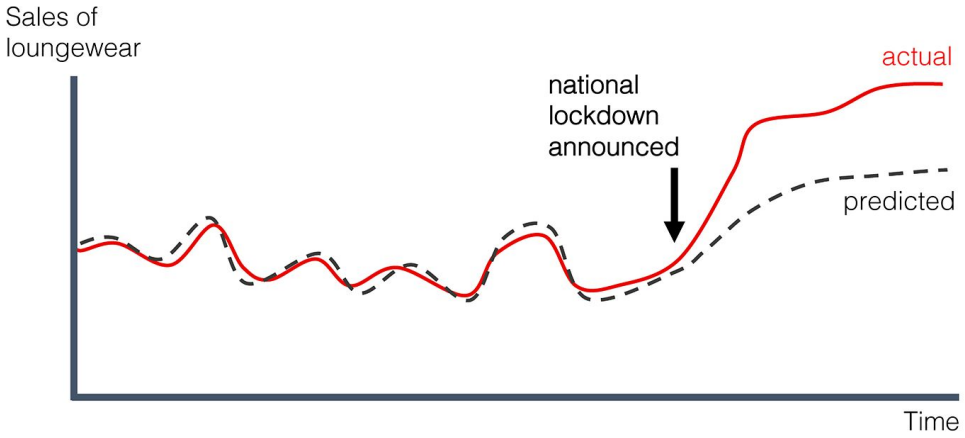
FairCanary: Rapid Continuous Explainable Fairness

AIES 2022

Research Questions

- **RQ1:** How does noise in demographic information as an input to a fair ML algorithm adversely impact the intended fairness of the outcomes for different subgroups?
- **RQ2:** How can fair ML models be attacked by adversarial actors to create even more unfairness?
- **RQ3:** In fair ML techniques that deliberately do not use protected attributes, how do their theoretical guarantees hold up in real life when compared against actual ground truth?
- **RQ4:** Do fair ML models, once deployed in a production system, continue to remain fair in the face of changing data and feature-output relationships? If so, how can such unfairness be measured and mitigated?

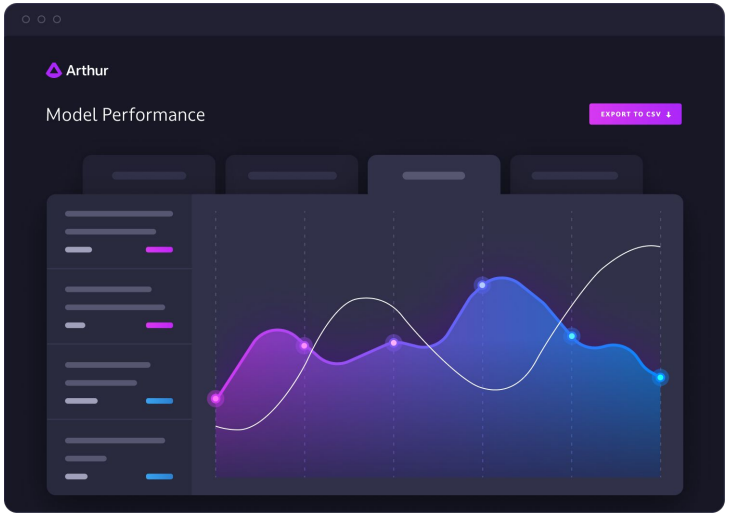
Drift in Machine Learning



Once trained and deployed models might become stale over time:

- *Data drift* occurs when the runtime data is significantly different from the training data, by virtue of the constant changing of real world data
- *Concept drift* occurs when the relationship between the model output and the feature

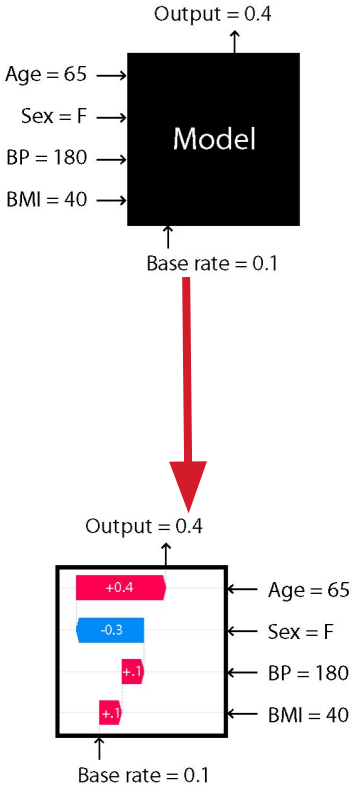
Model Monitoring



In general, commercial model monitoring systems offer the following features:

- Continuously record model inputs and model predictions.
- Measure and report traditional performance metrics over time, like precision, recall, and accuracy.

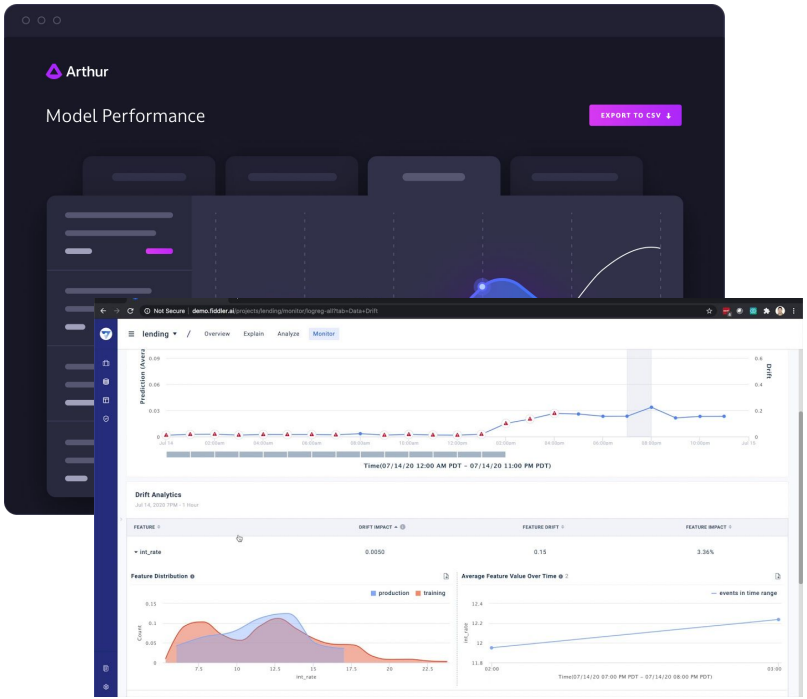
Model Monitoring



In general, commercial model monitoring systems offer the following features:

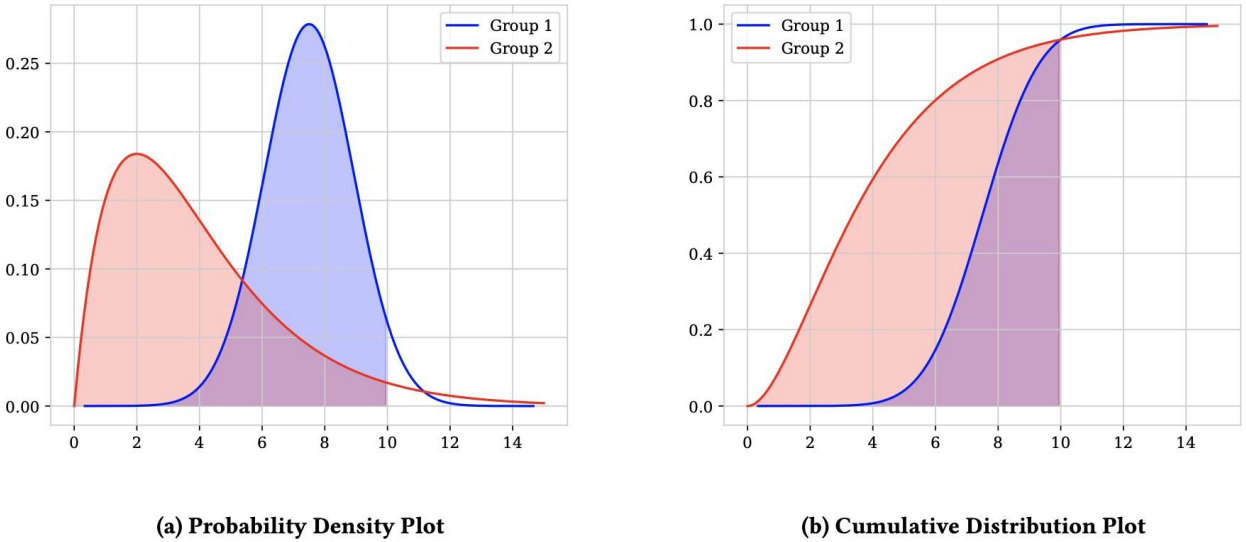
- Continuously record model inputs and model predictions.
- Measure and report traditional performance metrics over time, like precision, recall, and accuracy.
- Calculate and record feature-level explanations using techniques like LIME or SHAP, which are useful for post-mortem analysis if problems are observed.
- Generate alarms if particular metrics fall below an operator-specified threshold.

Model Monitoring



- However, these commercial systems do not provide featurewise explanations for unfairness, even though like any other metric, fairness might drift over time.
- I describe a system in this paper to reuse the model output explanations provided by continuous model monitoring systems to also provide continuous fairness explanations.

Why Conventional Metrics Might Be Unsuitable



(a) Probability Density Plot

(b) Cumulative Distribution Plot

Probability distribution plots for two hypothetical demographic groups. As demonstrated by the CDF plot on the right, at a threshold of x = 10 the positive prediction probability for both groups is about 0.95, thereby satisfying Demographic Parity but this is misleading: the Wasserstein distance is nonzero since the two distributions have markedly different shapes.

Distributional Difference based metrics for continuous outputs

Metric/Framework	Related Terms	CO?	E?
Demographic parity	mean difference, demographic parity, disparate treatment	✗	✗
Conditional statistical parity	statistical parity, conditional procedure accuracy, disparate treatment	✗	✗
Equalized odds	equalized odds, false positive/negative parity, disparate treatment	✗	✗
Equal opportunity	equality of opportunity, individual fairness, disparate treatment	✗	✗
Counterfactual fairness	counterfactual fairness, disparate treatment, fliptest	✗	✗
Statistical independence	HGR coefficient, independence	✓	✗
Distributional difference	KL divergence, JS Divergence, Wasserstein distance	✓	✓

Summary showing whether conventional classes of fairness metrics support **Continuous Output (CO) and feature-level Explanations (E)**. Metric families are inspired by (Mehrabi et al. 2019) and the related terminology is from (Das et al. 2021).

Distributional Difference based metrics for continuous outputs

Metric/Framework	Related Terms	CO?	E?
Demographic parity	mean difference, demographic parity, disparate treatment	✗	✗
Conditional statistical parity	statistical parity, conditional procedure accuracy, disparate treatment	✗	✗
Equalized odds	equalized odds, false positive/negative parity, disparate treatment	✗	✗
Equal opportunity	equality of opportunity, individual fairness, disparate treatment	✗	✗
Counterfactual fairness	counterfactual fairness, disparate treatment, fliptest	✗	✗
Statistical independence	HGR coefficient, independence	✓	✗
Distributional difference	KL divergence, JS Divergence, Wasserstein distance	✓	✓

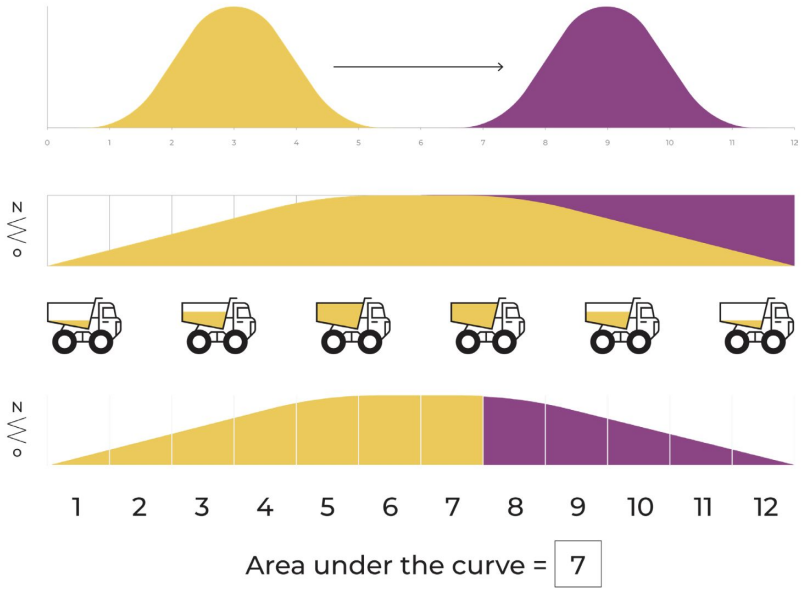
Summary showing whether conventional classes of fairness metrics support **Continuous Output (CO) and feature-level Explanations (E)**. Metric families are inspired by (Mehrabi et al. 2019) and the related terminology is from (Das et al. 2021).

Desirable Properties

We now discuss a few desirable properties of a distributional difference fairness metric that fit our stated objectives:

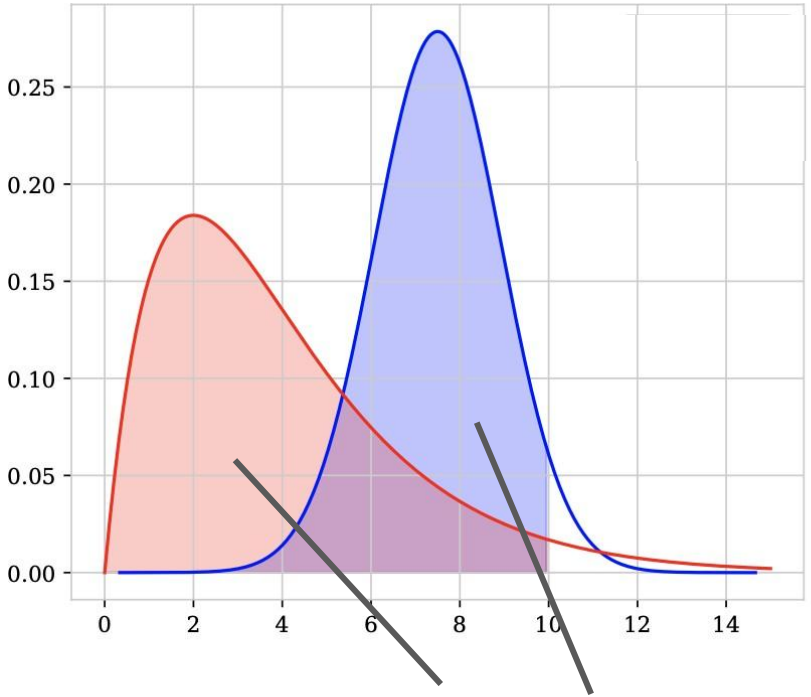
- The metric should be in the units of the model's prediction scores
- The metric should take the value zero *only* if the prediction distributions being compared are exactly the same
- The metric should be continuous with respect to changes in the geometry of the distribution
- The metric should be non-invariant with respect to monotone transformations of the distributions (eg if all points are multiplied by K, the metric also shifts by K)

QDD: Quantile Demographic Disparity



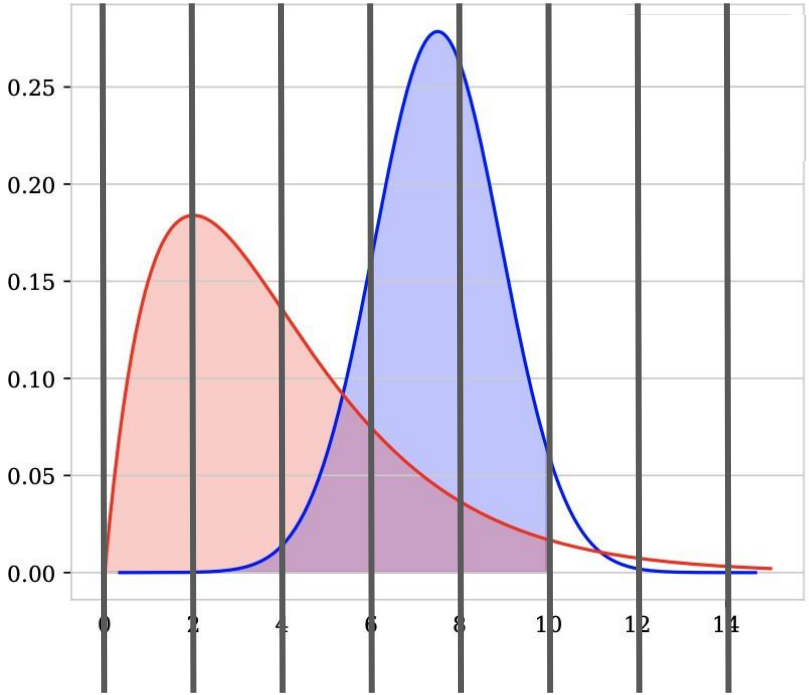
Wasserstein's Distance (Earth Mover's Distance) - Optimal Transport

QDD: Quantile Demographic Disparity



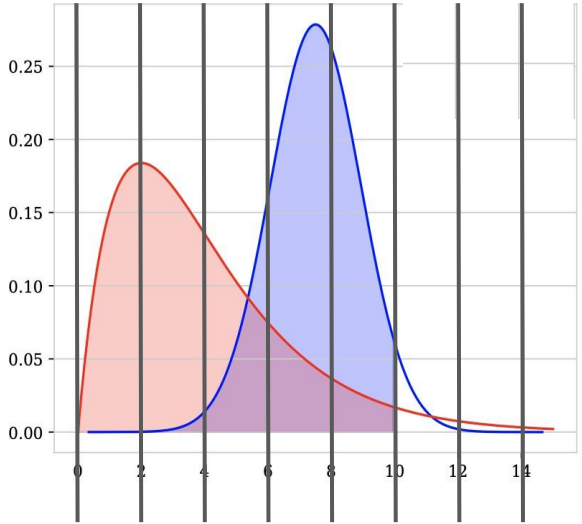
Say we have two score distributions **S1** and **S2** for two groups G1 and G2

QDD: Quantile Demographic Disparity



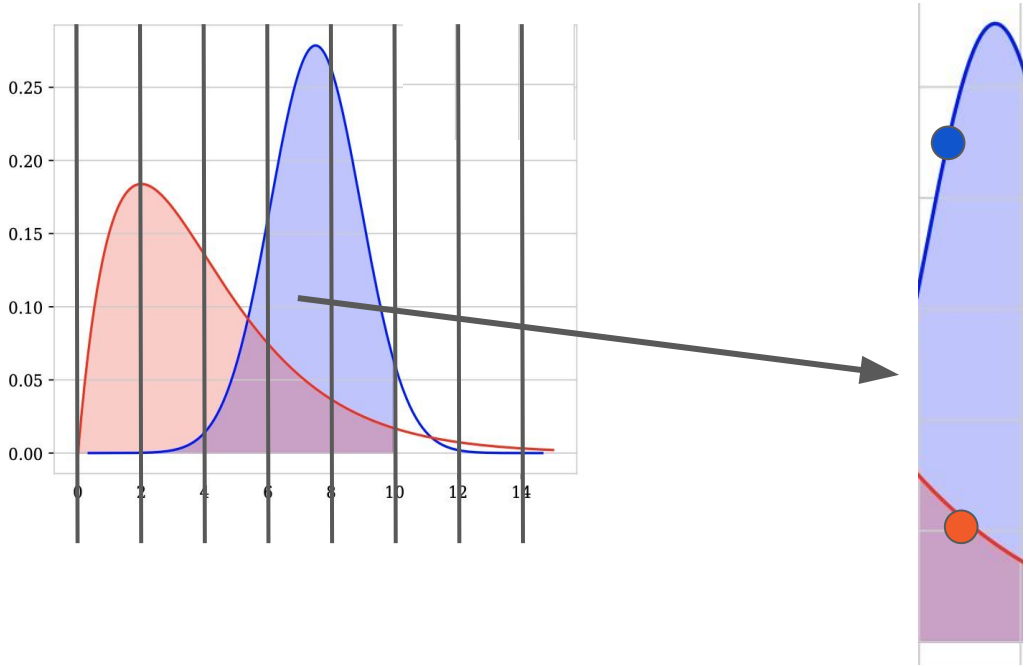
We split the two distributions into **B** bins of equal size (similar to percentiles).
Each bin has **N1** items from G1 and **N2** items from G2

QDD: Quantile Demographic Disparity



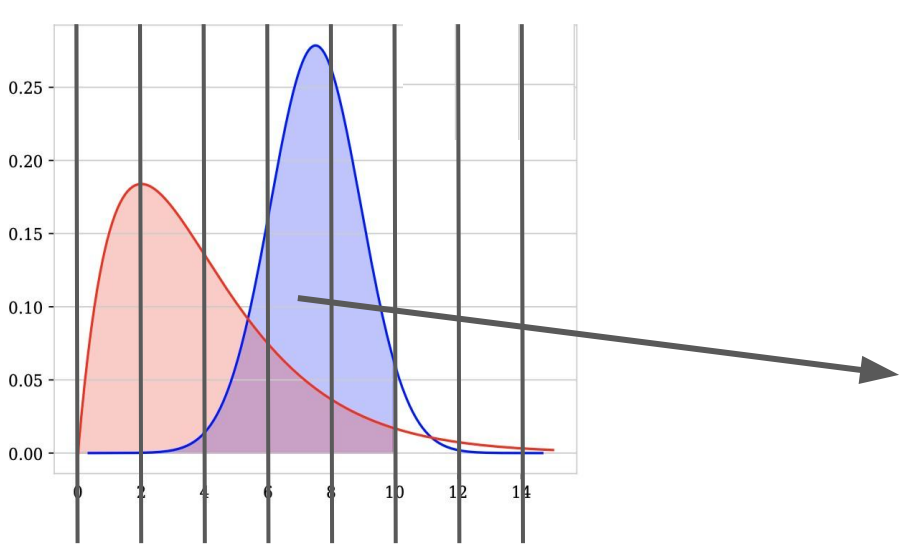
Let's look at one bin, b

QDD: Quantile Demographic Disparity



QDD for bin b = Mean value of **S1** in bin b - Mean value of **S2** in bin b

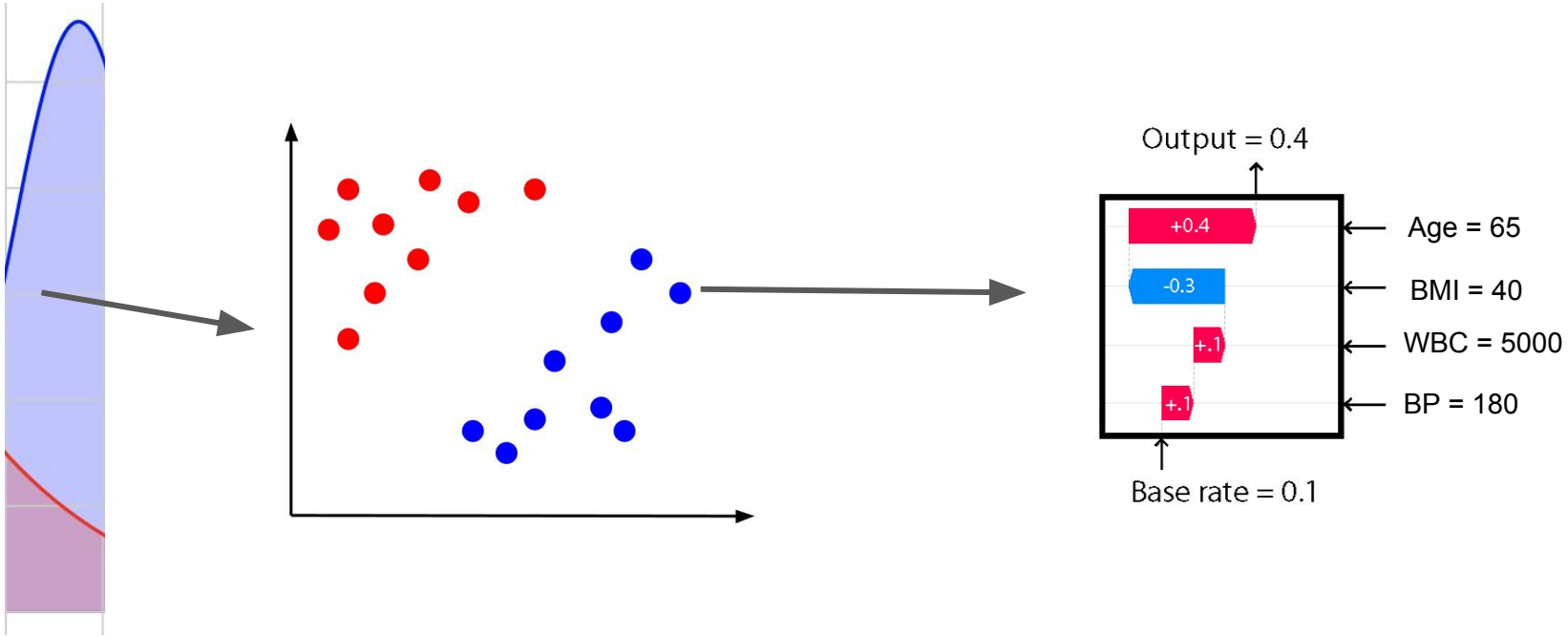
QDD: Quantile Demographic Disparity



$$QDD_b = \frac{1}{N_1}\sum_{n=1}^{N_1} S_{1,n} - \frac{1}{N_2}\sum_{n=1}^{N_2} S_{2,n}.$$

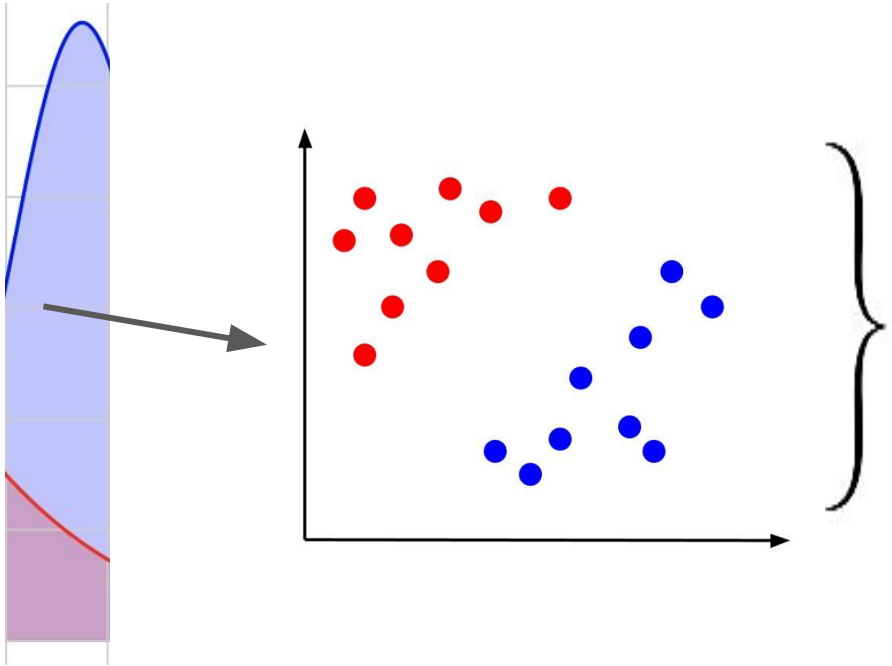
QDD for bin b = Mean value of **S1** in bin b - Mean value of **S2** in bin b

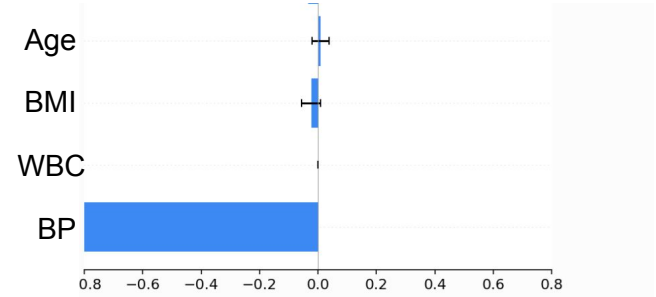
QDD Attribution



QDD Attributions/Explanations reuse per-prediction explanations that are already calculated and stored

QDD Attribution

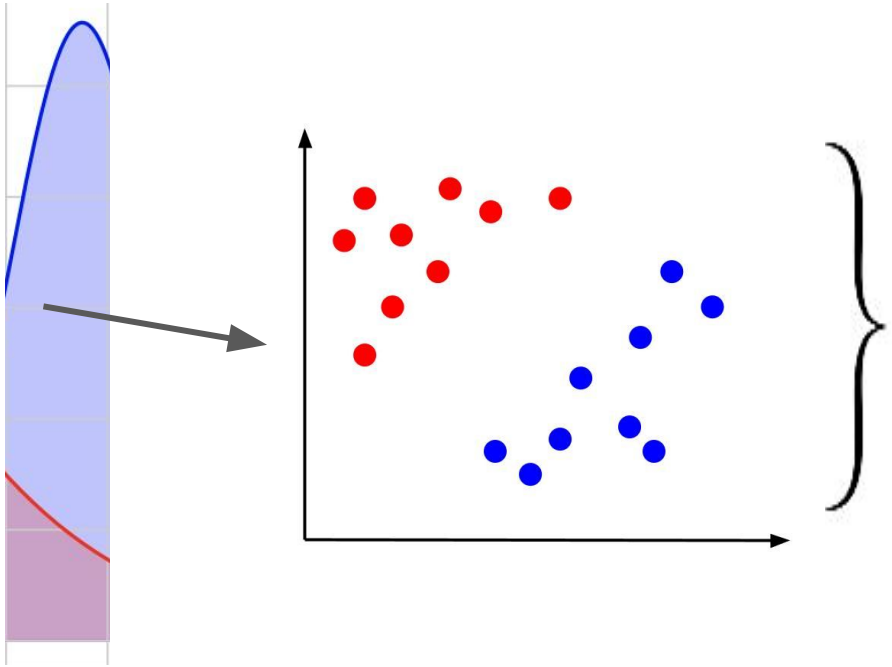


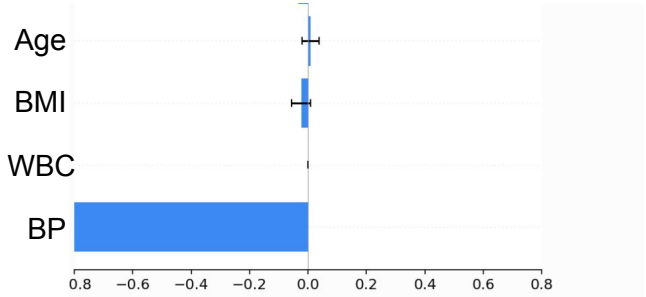


$$\mathrm{QDDA}_{b,A,f} = \frac{1}{N_t}\sum_{n=1}^{N_t}\mathrm{attr}_{n,A,S_1,f} - \frac{1}{N_r}\sum_{n=1}^{N_r}\mathrm{attr}_{n,A,S_2,f}$$

QDDA for bin b, feature f = Mean value of feature atts. for **S1** in bin b - Mean value of feature atts. for **S2** in bin b

QDD Attribution

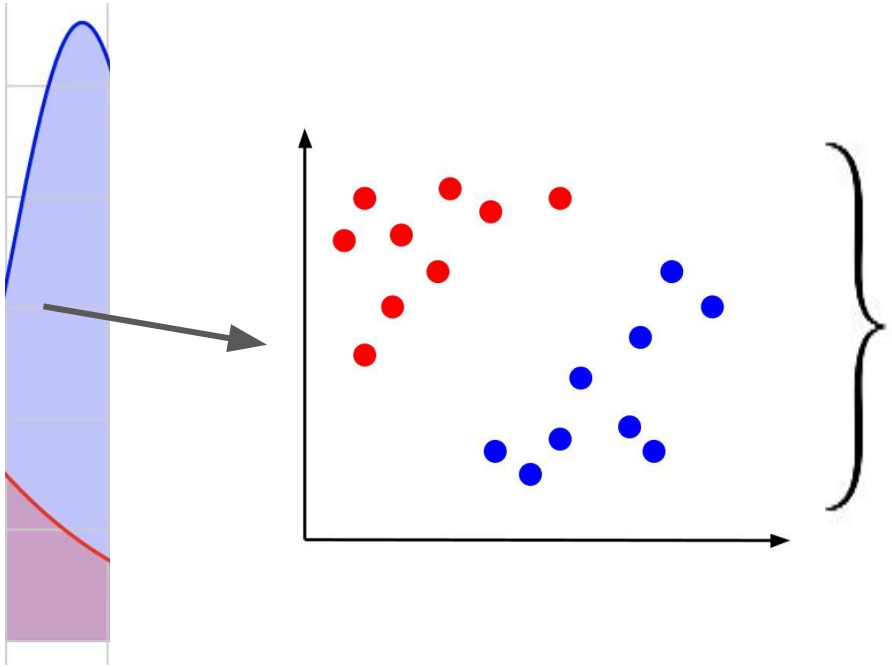


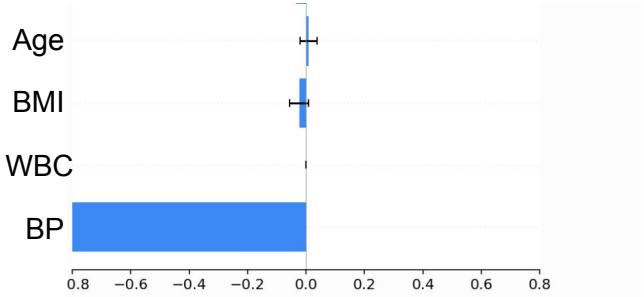


$$\mathrm{QDDA}_{b,A,f} = \frac{1}{N_t}\sum_{n=1}^{N_t}\mathrm{attr}_{n,A,S_1,f} - \frac{1}{N_r}\sum_{n=1}^{N_r}\mathrm{attr}_{n,A,S_2,f}$$

QDDA for bin b, feature f = Mean value of feature atts. for **S1** in bin b - Mean value of feature atts. for **S2** in bin b

QDD Attribution



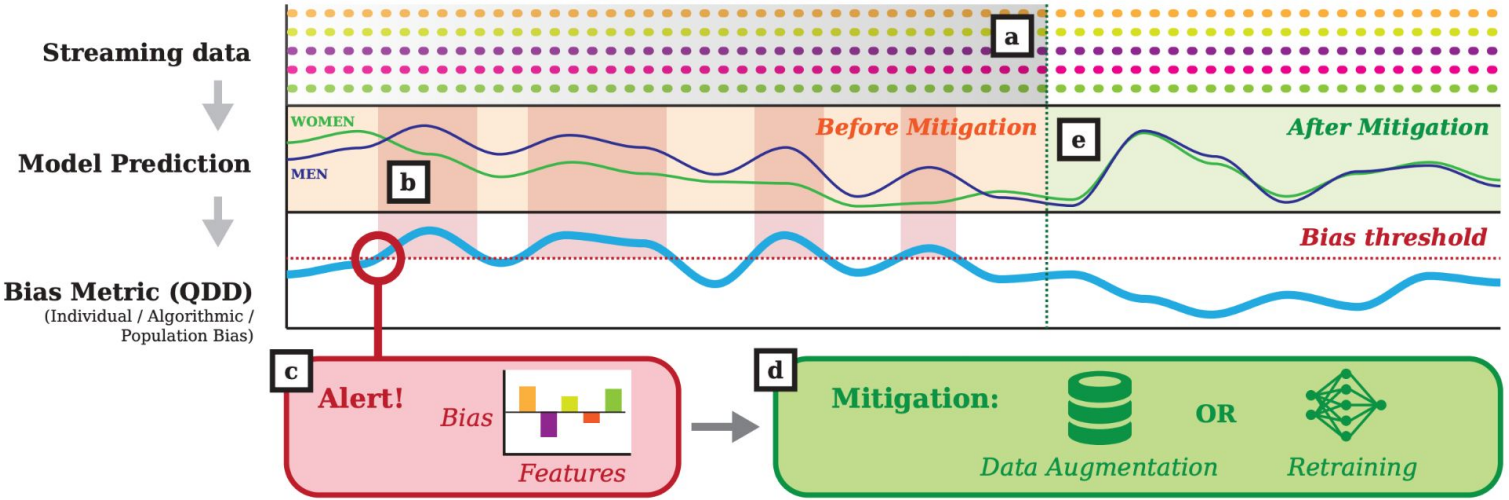


$$\frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{f=1}^{F}\text{attr}_{n,A,S_1,f} - \frac{1}{N_r}\sum_{n=1}^{N_r}\sum_{f=1}^{F}\text{attr}_{n,A,S_2,f} = \frac{1}{N_1}\sum_{n=1}^{N_2} S_{1,n} - \frac{1}{N_2}\sum_{n=1}^{N_2} S_{2,n}$$

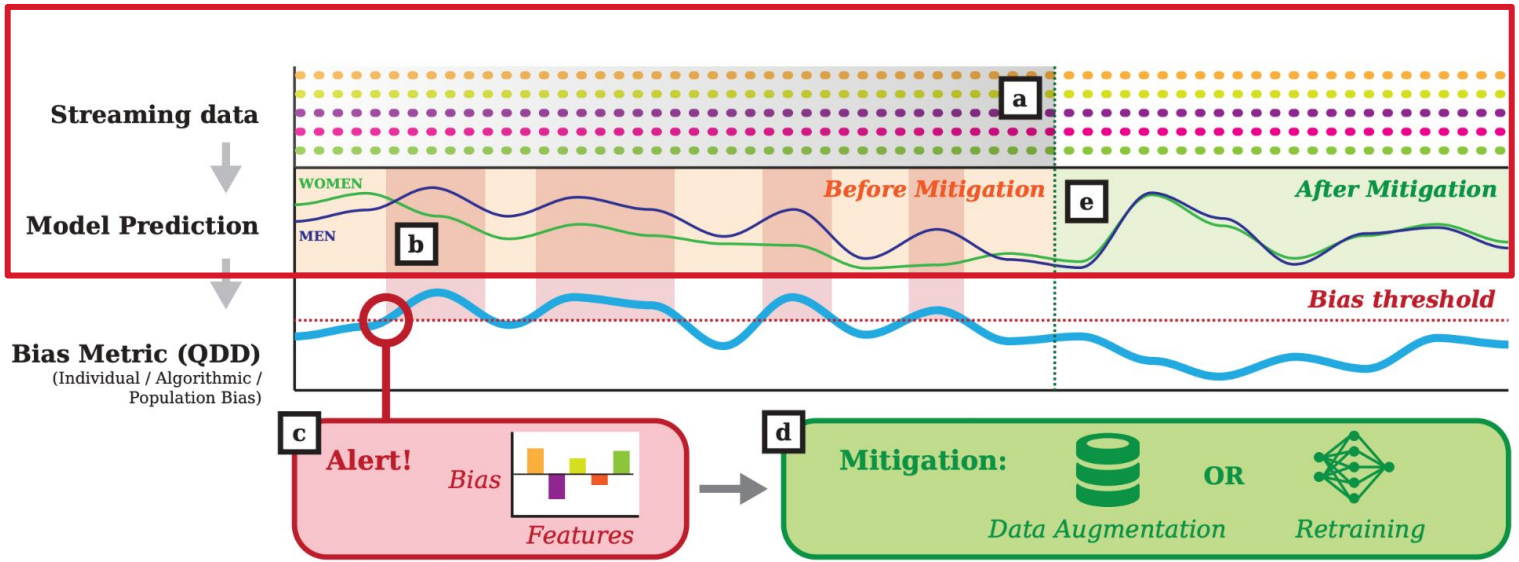
$$\therefore \text{QDD}_b = \sum_{f=1}^{F}\text{QDDA}_{b,A,f}.$$

Sum of attributes of QDDA = QDD!

Schematic

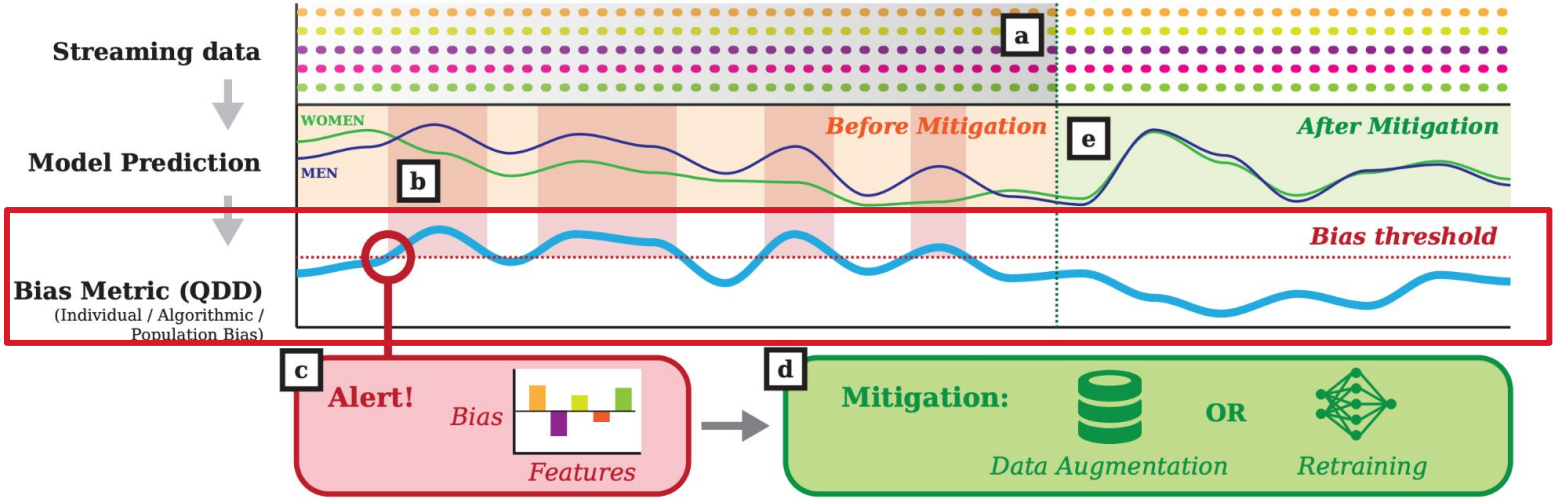

Streaming data
Model Prediction
Bias Metric (QDD)
(Individual / Algorithmic / Population Bias)
a
WOMEN
MEN
b
Before Mitigation
e
After Mitigation
Bias threshold
c
Alert!
Bias
Features
d
Mitigation:
OR
Data Augmentation
Retraining

Schematic



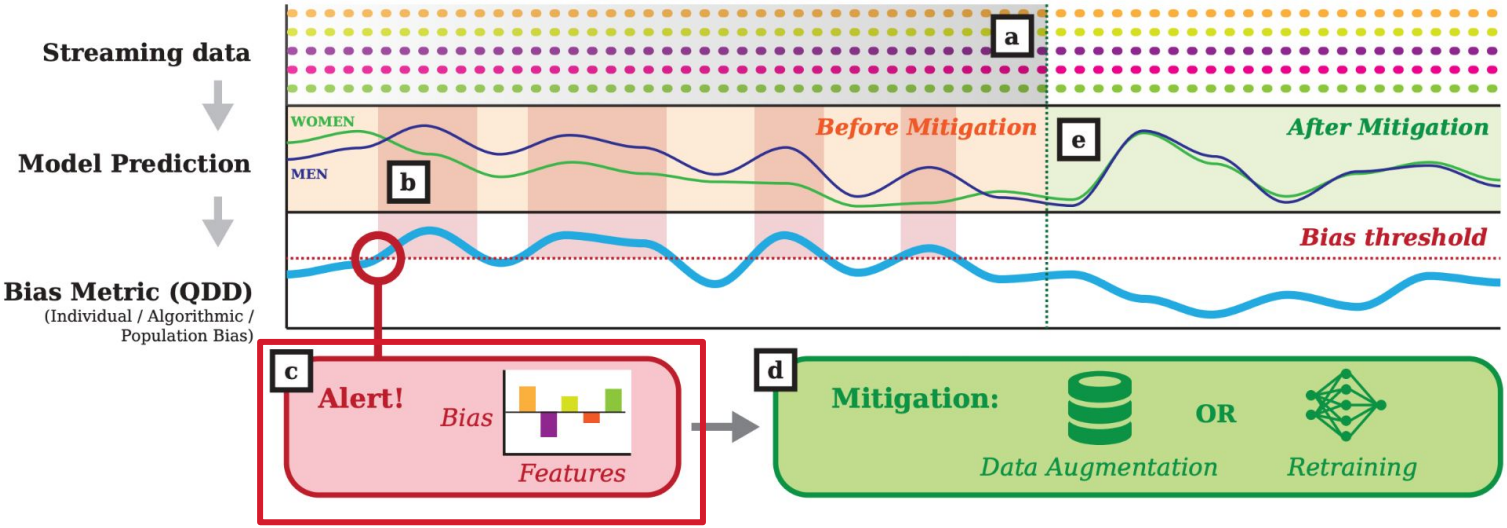
1. Monitors the inputs and outputs of a trained model over time

Schematic



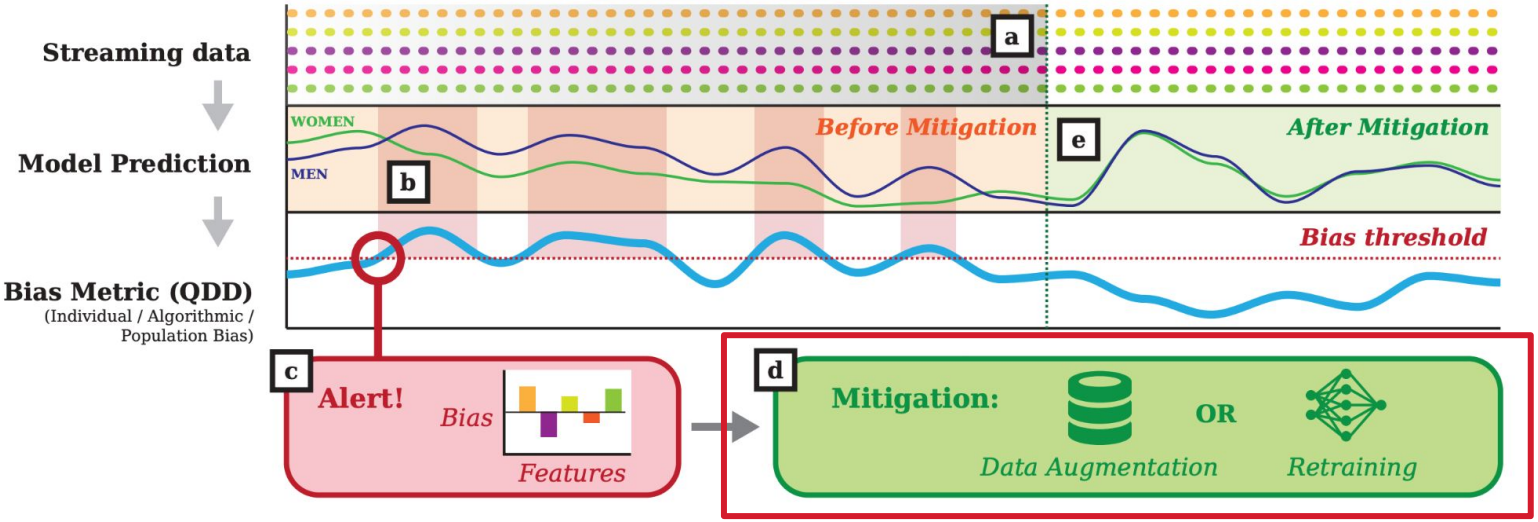
1. Monitors the inputs and outputs of a trained model over time
2. Identifies bias

Schematic



1. Monitors the inputs and outputs of a trained model over time
2. Identifies bias
3. Alerts the developer

Schematic



1. Monitors the inputs and outputs of a trained model over time
2. Identifies bias
3. Alerts the developer
4. Assists in Mitigation

Case Study

Feature	Values	Distribution
Location	{'Springfield', 'Centerville'}	70:30
Education	{'GRAD', 'POST GRAD'}	80:20
Engineer Type	{'Software', 'Hardware'}	85:15
Experience (Years)	(0, 50)	Normal Distribution
Relevant Experience (Years)	(0, 50)	Normal Distribution
Gender	{'MAN', 'WOMAN'}	50:50

$$\text{Salary} = 50{,}000 + (20{,}000 \times \text{location}) + (20{,}000 \times \text{education}) + (5{,}000 \times \text{relevant experience}) + (100 \times \text{experience}) + (10{,}000 \times \text{engineer type})$$

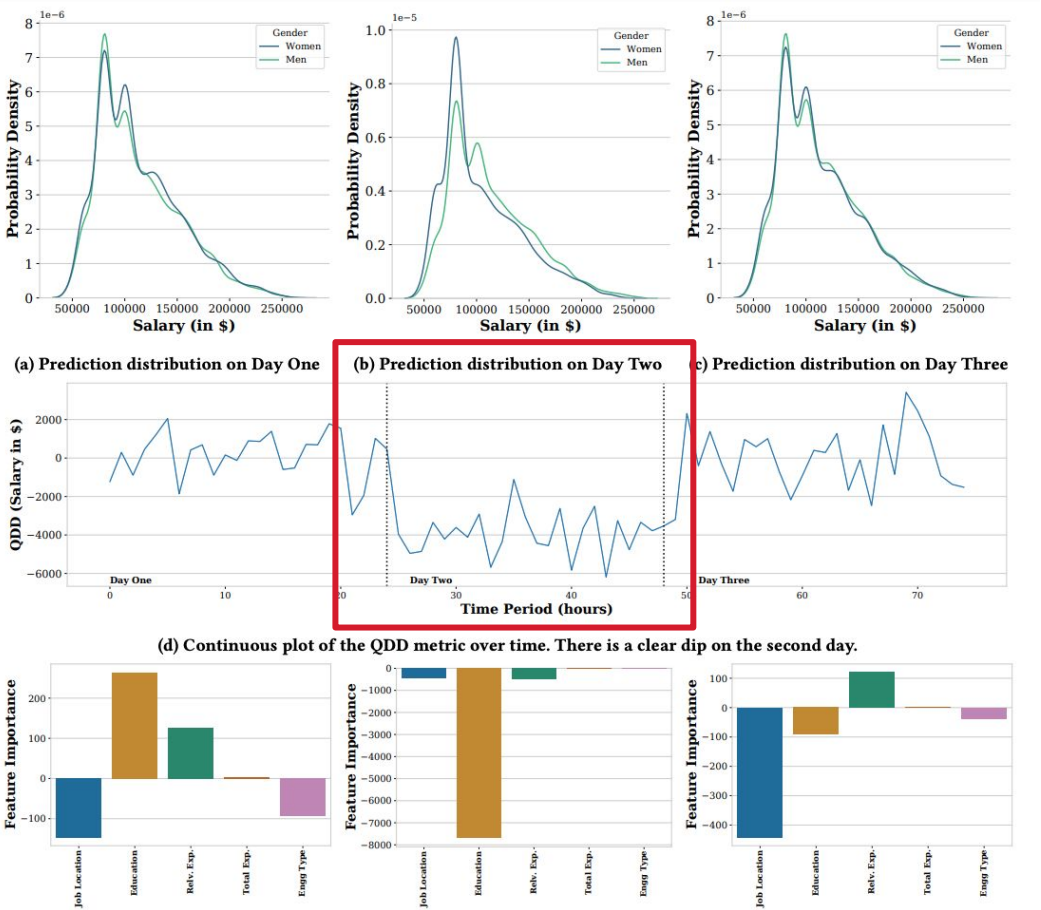
Case Study

Feature	Values	Distribution
Location	{'Springfield', 'Centerville'}	70:30
Education	{'GRAD', 'POST GRAD'}	80:20
Engineer Type	{'Software', 'Hardware'}	85:15
Experience (Years)	(0, 50)	Normal Distribution
Relevant Experience (Years)	(0, 50)	Normal Distribution
Gender	{'MAN', 'WOMAN'}	50:50

$$\text{Salary} = 50{,}000 + (20{,}000 \times \text{location}) + (20{,}000 \times \text{education}) + (5{,}000 \times \text{relevant experience}) + (100 \times \text{experience}) + (10{,}000 \times \text{engineer type})$$

I synthetically insert a data bug on Day 2, by converting every woman's education to GRAD only.

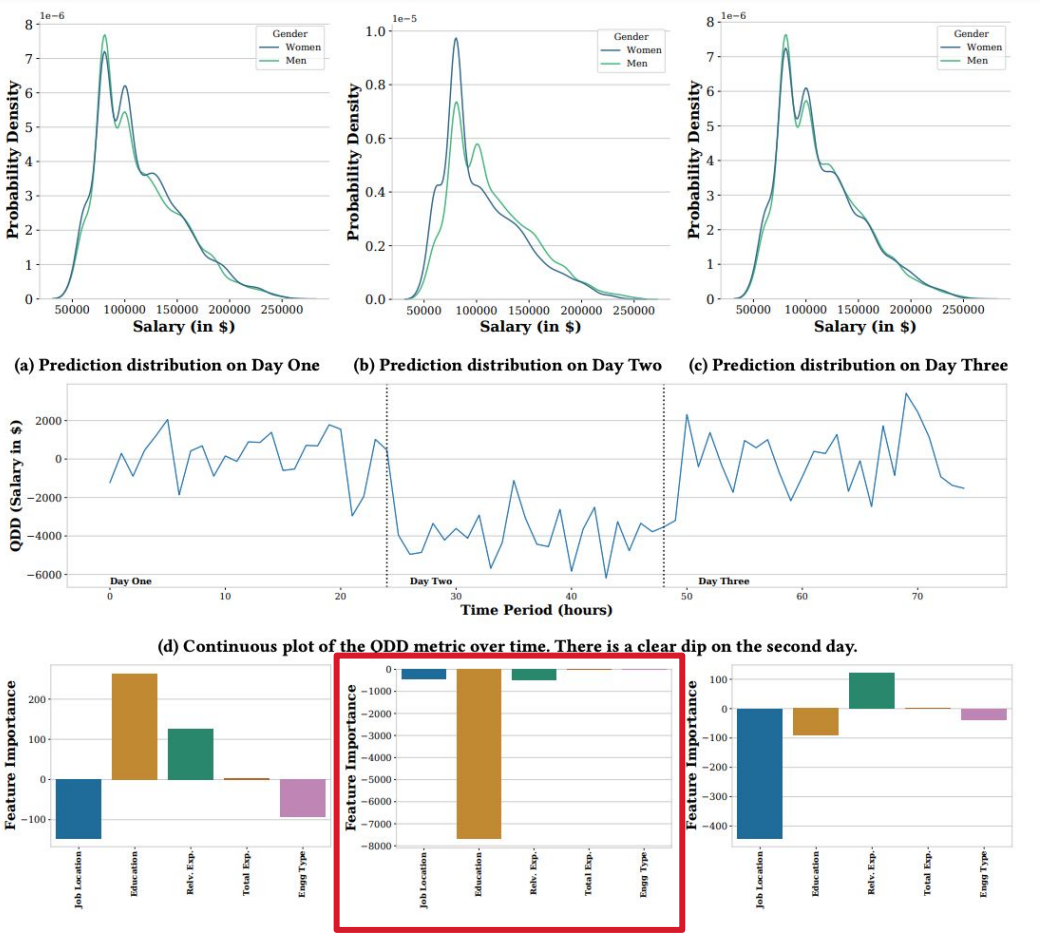
Case Study



QDD shows a steep drop on Day 2, causing an alert.

The units of QDD are in dollars, showing that Women experience a salary difference of around $4000-6000 relative to men.

Case Study



The explanation of the alert clearly shows that Education was the rogue feature, helping the developer to fix the data bug.

Case Study

	Day One		Day Two	
Threshold	**SPD**	**DI**	**SPD**	**DI**
$50000	0.00009	1.00009	-0.00556	0.99439
$100000	0.00911	1.01749	-0.08290	0.84569
$200000	0.00088	1.02876	-0.01049	0.65544

Statistical Parity Difference (SPD) and Disparate Impact (DI), against different salary thresholds for the case study. The predictions on Day One were fair, while they were unfair to women on Day Two. Only one metric catches the bias, and only at one threshold (highlighted in red).

Conclusion

- I present a novel, efficient, metric called QDD, and a system that uses it, called FairCanary, that continuously measures and explains bias in deployed ML models.
- I show the system in action with a case study and compare it against existing metrics.
- The system is not 100% automated, and hyperparameters like number of bins, alert sensitivity, explanation method, etc still need to be set.
- I hope FairCanary provides a blueprint for model owners to responsibly measure and mitigate bias in large deployed systems in the wild.

Background

Research Questions

Story so far

Awareness vs Unawareness

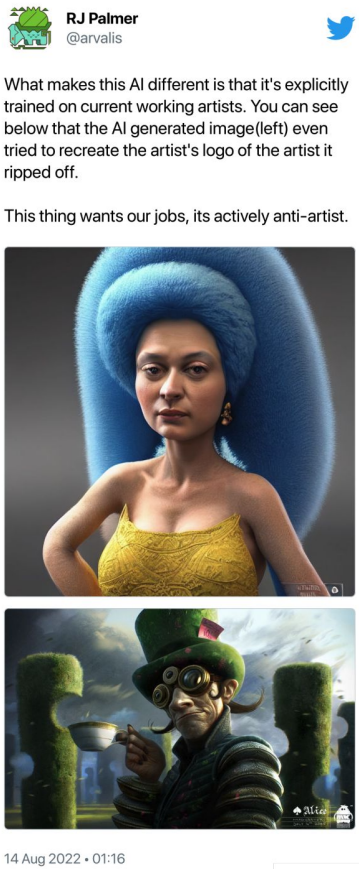
Continuous Fairness

Broader Impact

What's Next?

Humans in the loop

- Biases can be made worse by humans in the loop of a fair ML model
- Humans such as training data annotators and final decision makers
- Task is to make fair models resilient to human bias



What's Next?

Machine Unlearning

- Generative Models such as CoPilot (for text) and Stable Diffusion (for AI Art) have been accused of stealing and reproducing copyrighted data
- Task: To come up with a computationally efficient way to make a trained model forget problematic training data without complete retraining