Unprocessing Seven Years of Algorithmic Fairness

André F. Cruz and Moritz Hardt

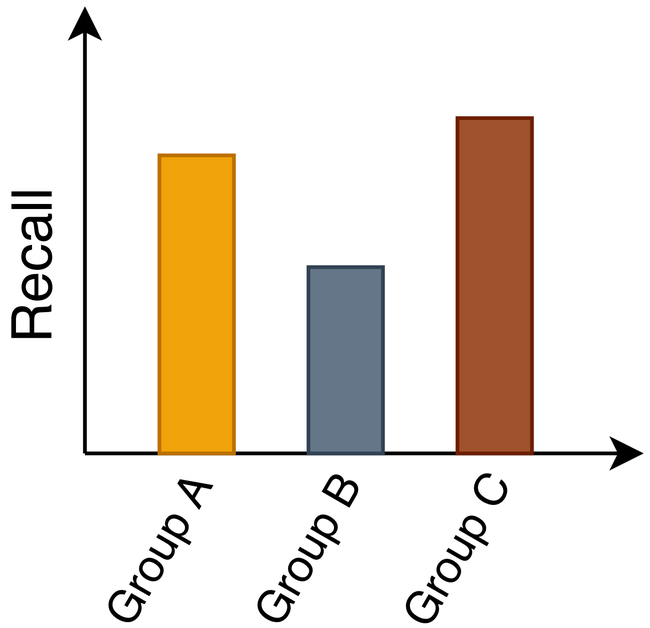
Max Planck Institute for Intelligent Systems, Tübingen and Tübingen AI Center

Tübingen AI Center

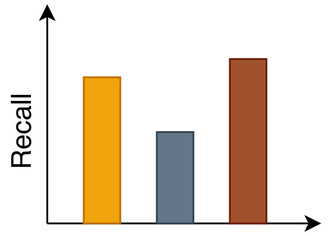
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

Motivation

- Problem with ML: Different error rates among different groups of the population.



Motivation

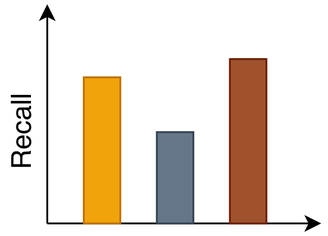


- Problem with ML: Different error rates among different groups of the population.

- Algorithmic fairness tools to mitigate error rate disparity:
  - Pre-processing: Change dataset.
  - In-processing: Change training algorithm.
  - Post-processing: Set group-specific thresholds.

- Post-processing is easiest and came first, but is widely considered suboptimal.

Motivation



- Problem with ML: Different error rates among different groups of the population.

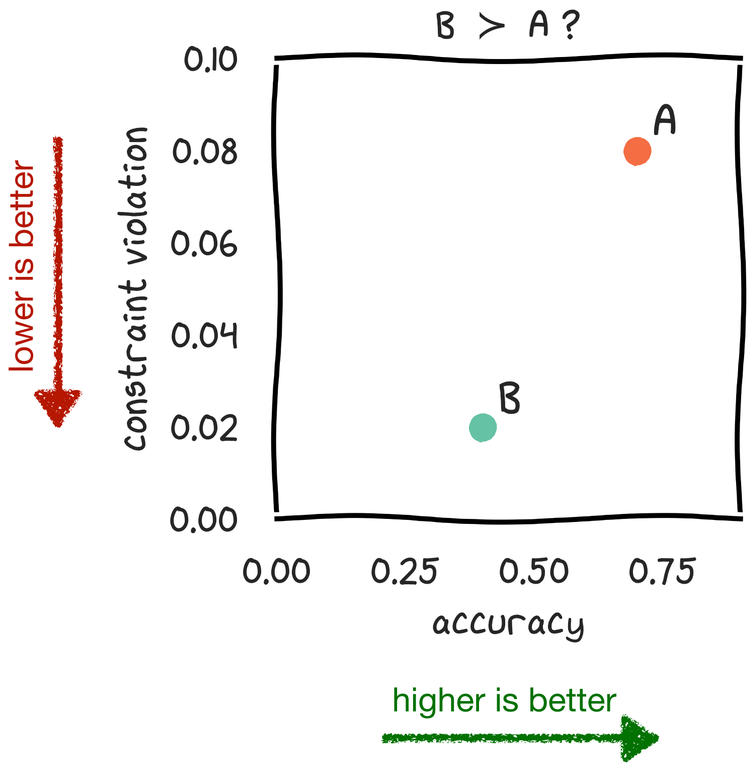
- Algorithmic fairness tools to mitigate error rate disparity:
  - Pre-processing: Change dataset.
  - In-processing: Change training algorithm.
  - Post-processing: Set group-specific thresholds.

- Post-processing is easiest and came first, but is widely considered suboptimal.

But how do we compare all these methods?

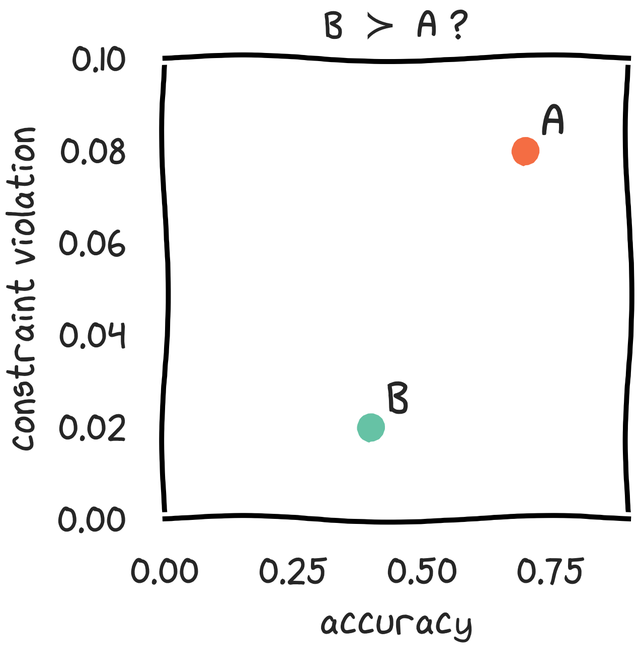
First problem in empirical evaluation

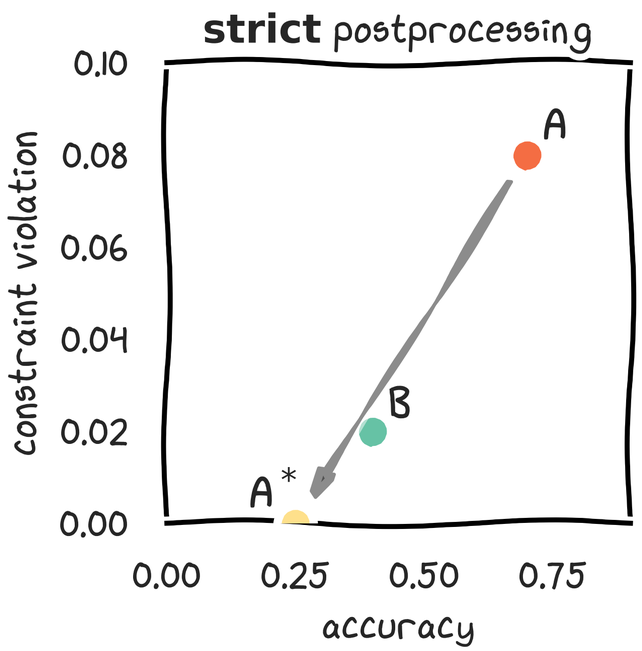
Comparing methods at different levels of fairness violation



First problem in empirical evaluation

Comparing methods at different levels of fairness violation

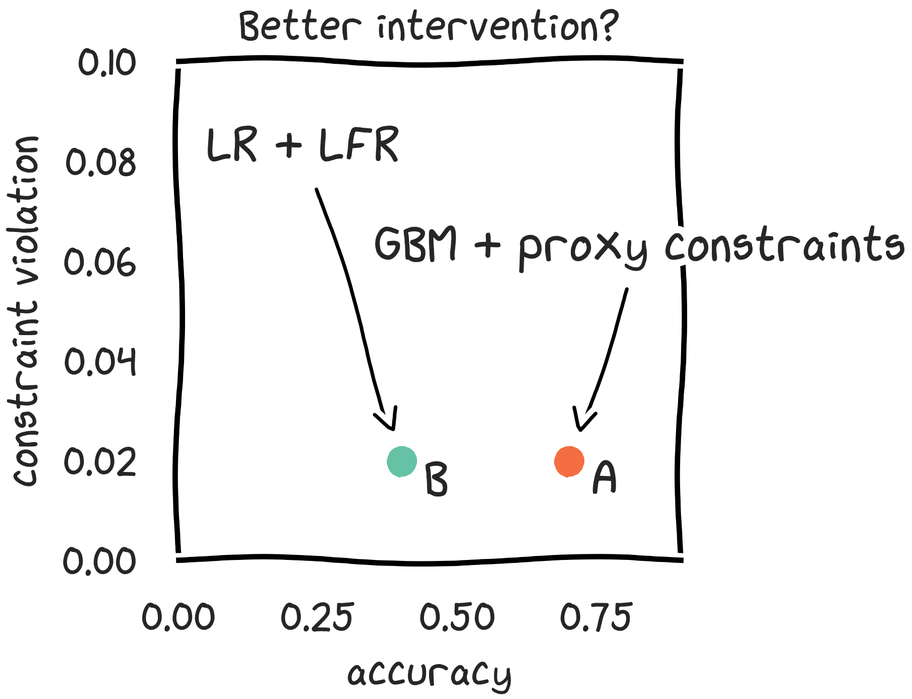




Post-processing achieves *exact* constraint fulfillment

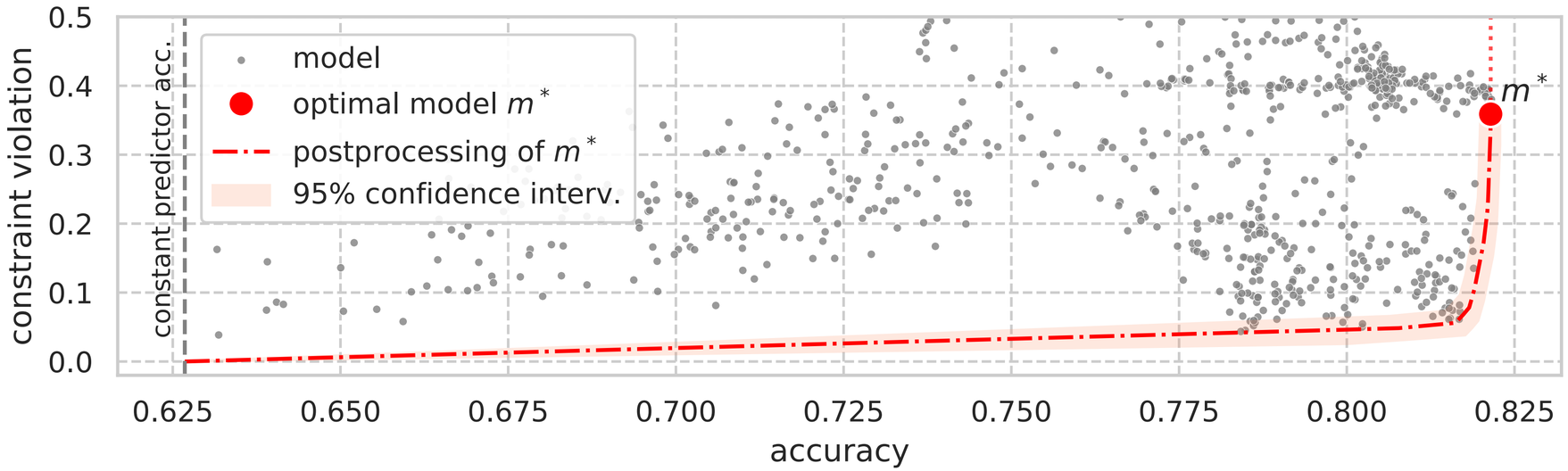
Second problem

Comparing methods with different unconstrained base models



Our Contributions

- We address both problems using a tool called *unprocessing*.
- We conduct a large-scale meta-study with 6 datasets and $11\,000$ models trained.
- We find that **postprocessing achieves highest accuracy** at all levels of fairness constraint relaxation.



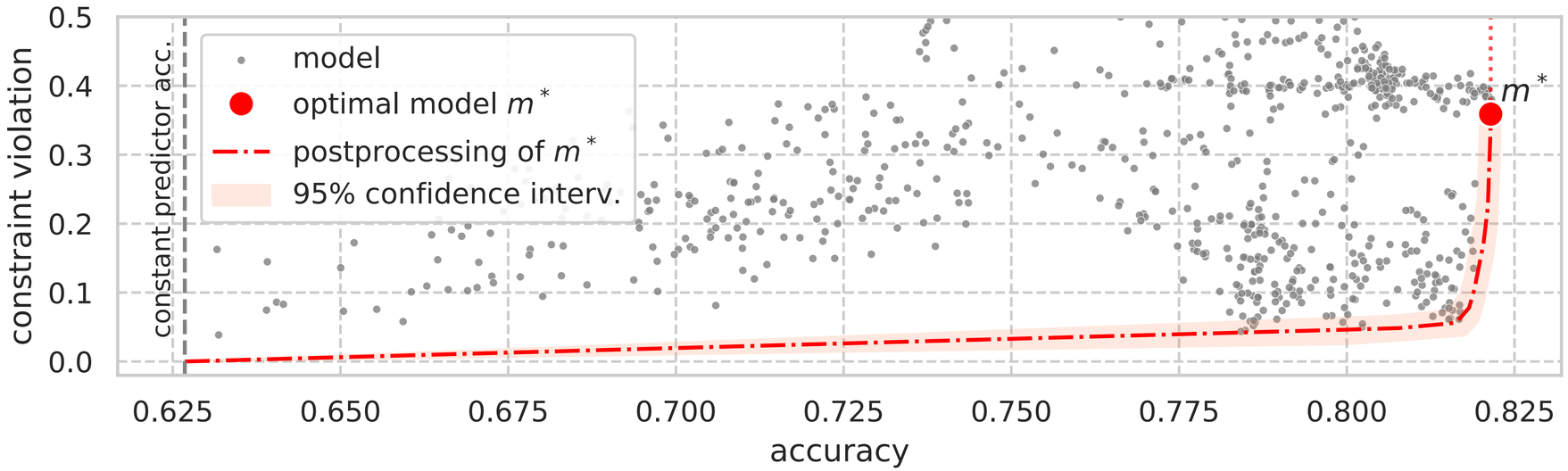
New open-source package: **pip install error-parity**

Main takeaway

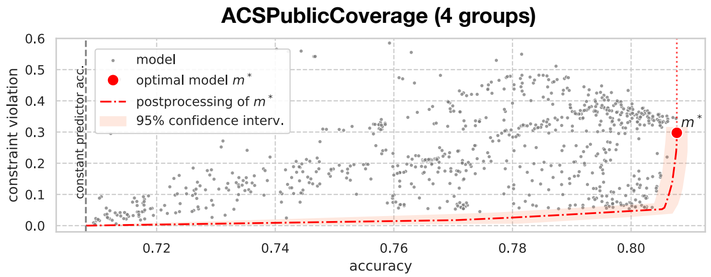
If your goal is to equalize error rates, either exactly or approximately:

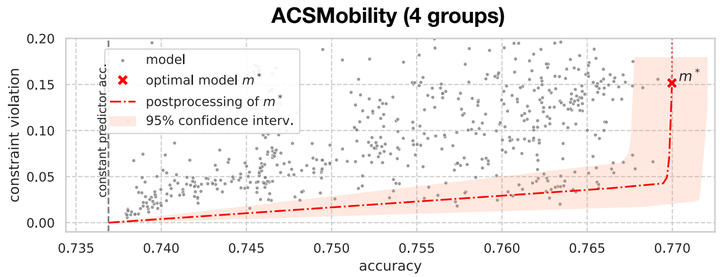
Take the best *unconstrained* model available and optimize over group-specific thresholds.

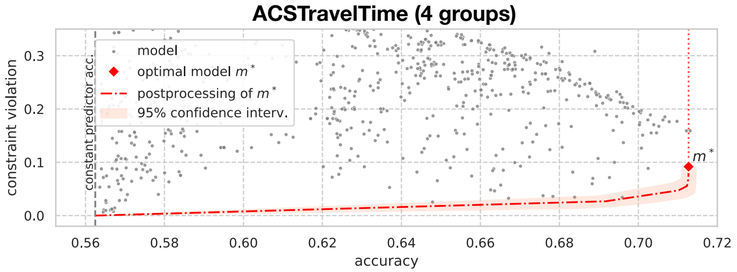
(postprocessing)

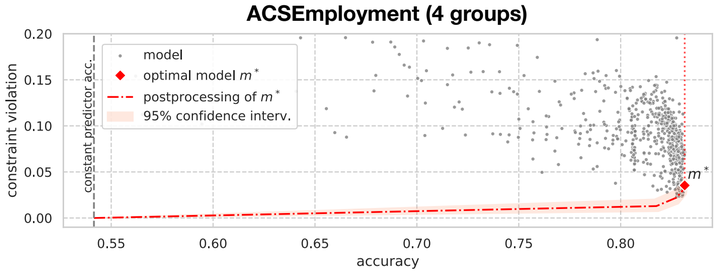


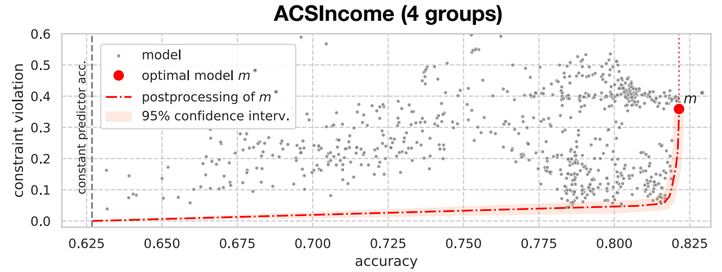
- 6 real-world datasets
- 11 evaluation scenarios
- 11K models trained

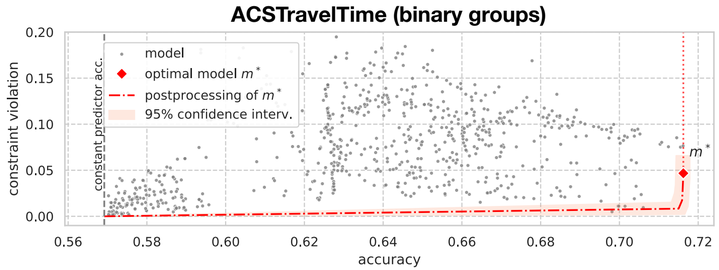


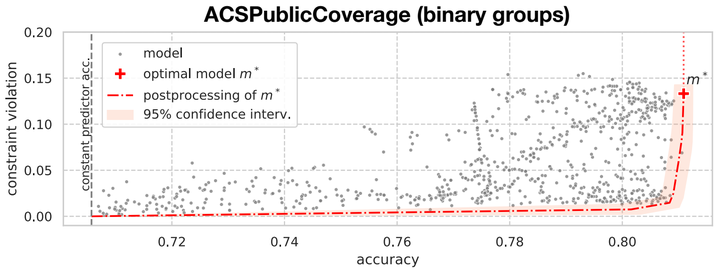


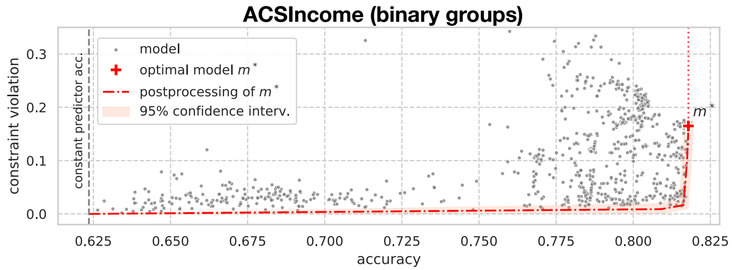


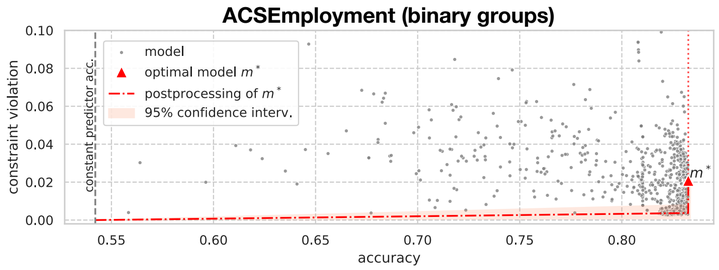


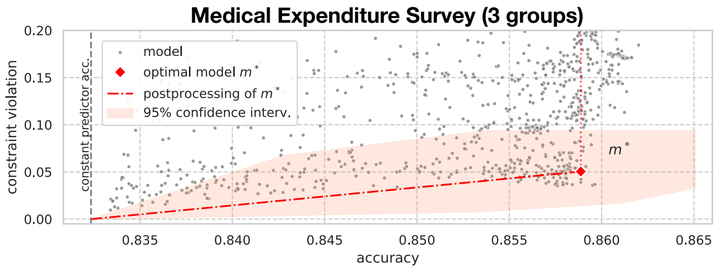


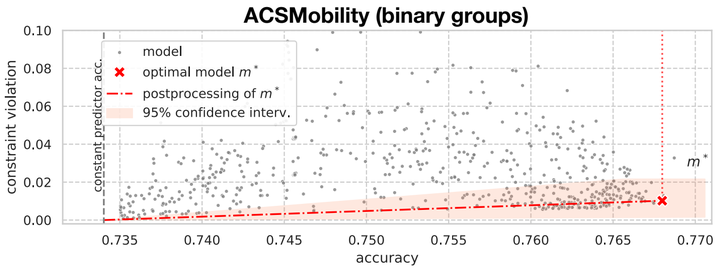








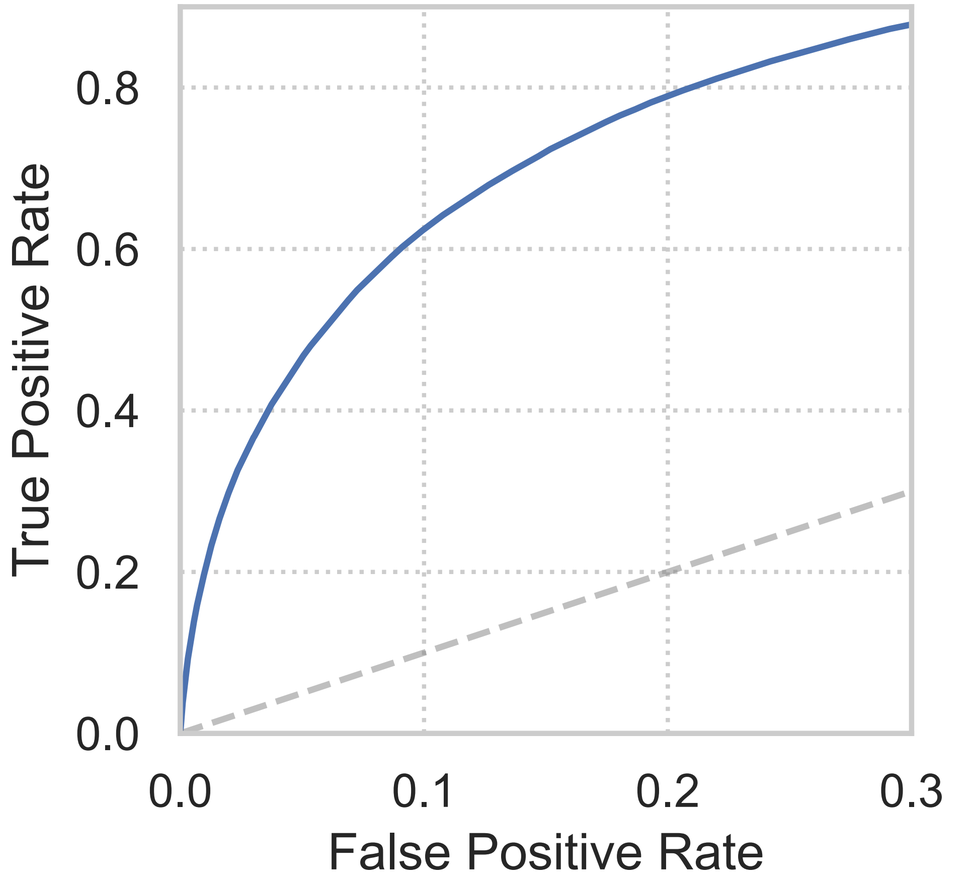




[Ding et al. “Retiring adult: New datasets for fair machine learning.” *NeurIPS*, 2021]

Datasets: Using **folktables** python package

Postprocessing



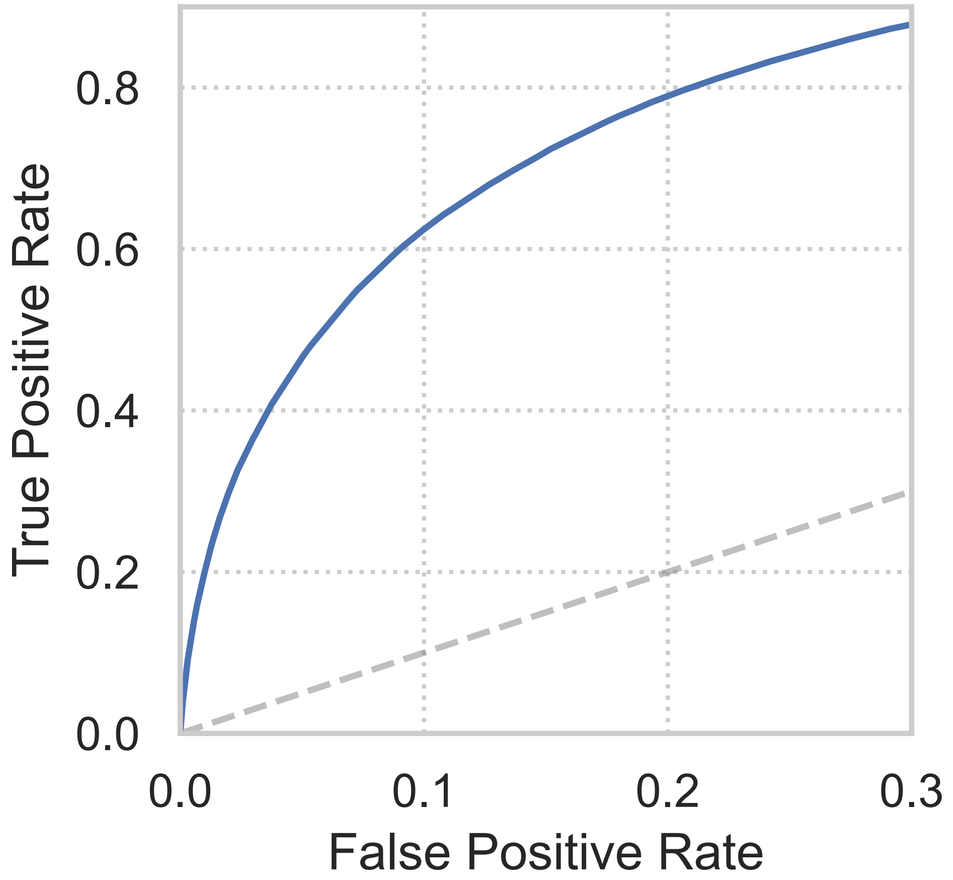
Postprocessing

- Choose point on ROC curve by changing threshold t.

$$C(t) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t \,|\, Y = 0\right]}^{FPR}, \overbrace{\mathbb{P}\left[\hat{R} \geq t \,|\, Y = 1\right]}^{TPR} \right),$$

risk score

threshold

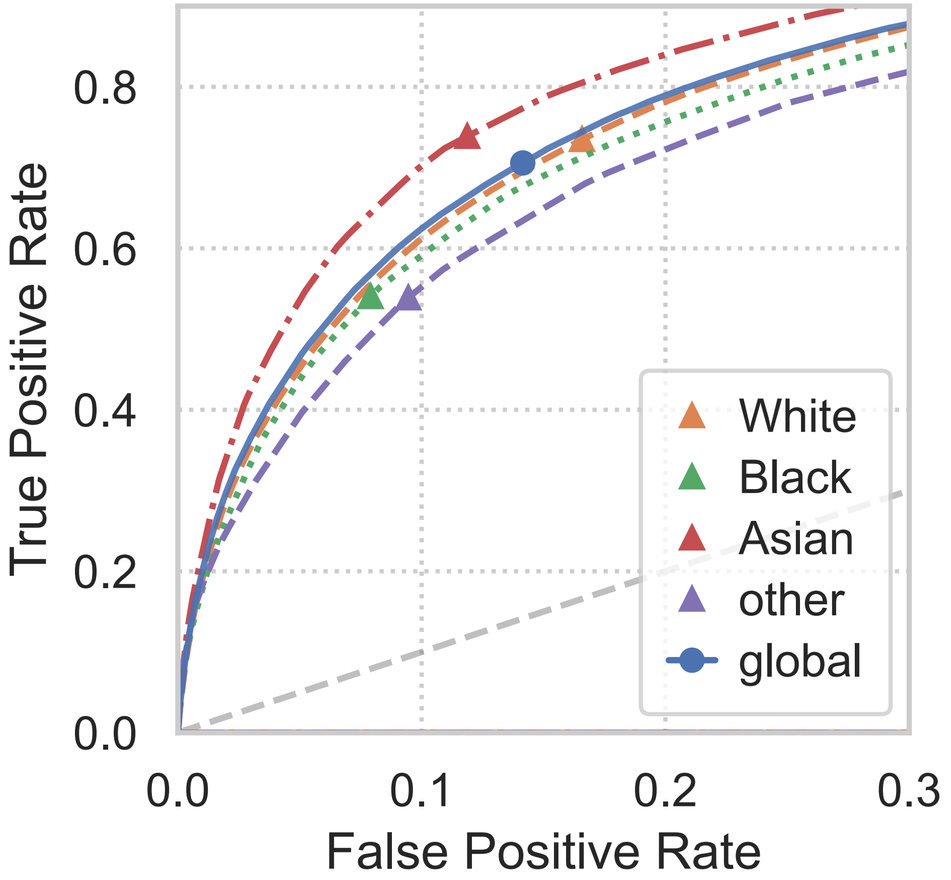


Postprocessing

- Choose point on ROC curve by changing threshold t.

$$C_s(t) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

group-specific ROC curve

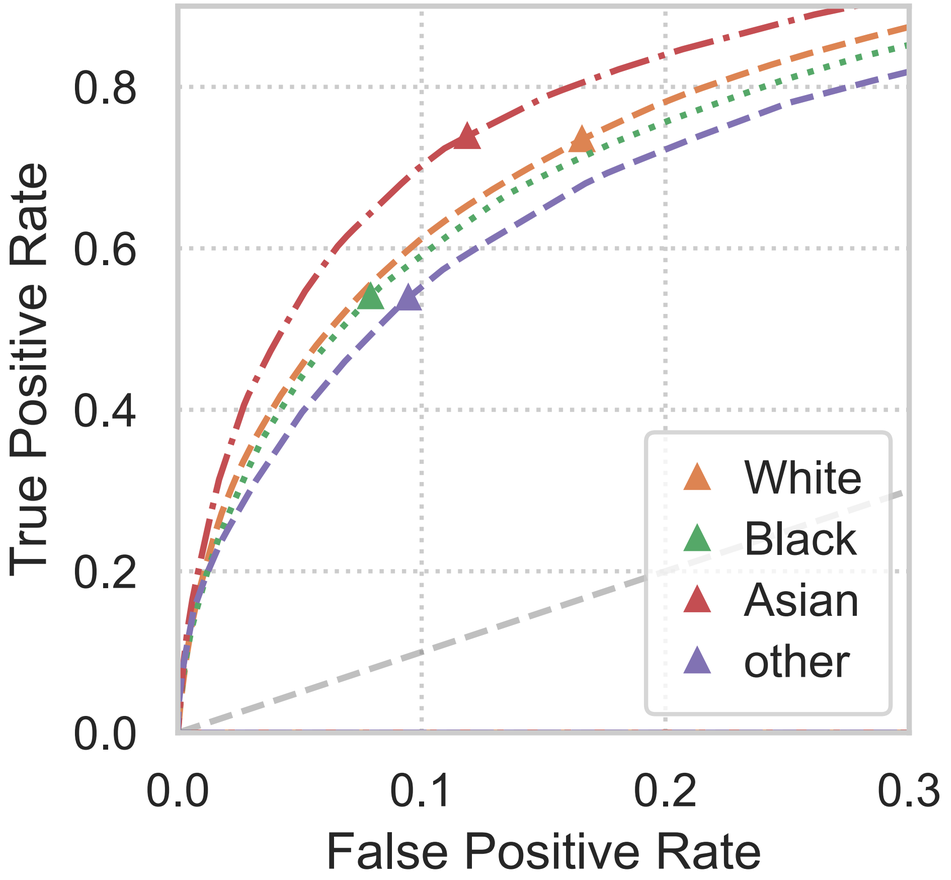


Postprocessing

- Choose point on ROC curve by changing threshold t.

$$C_s(t) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

group-specific ROC curve



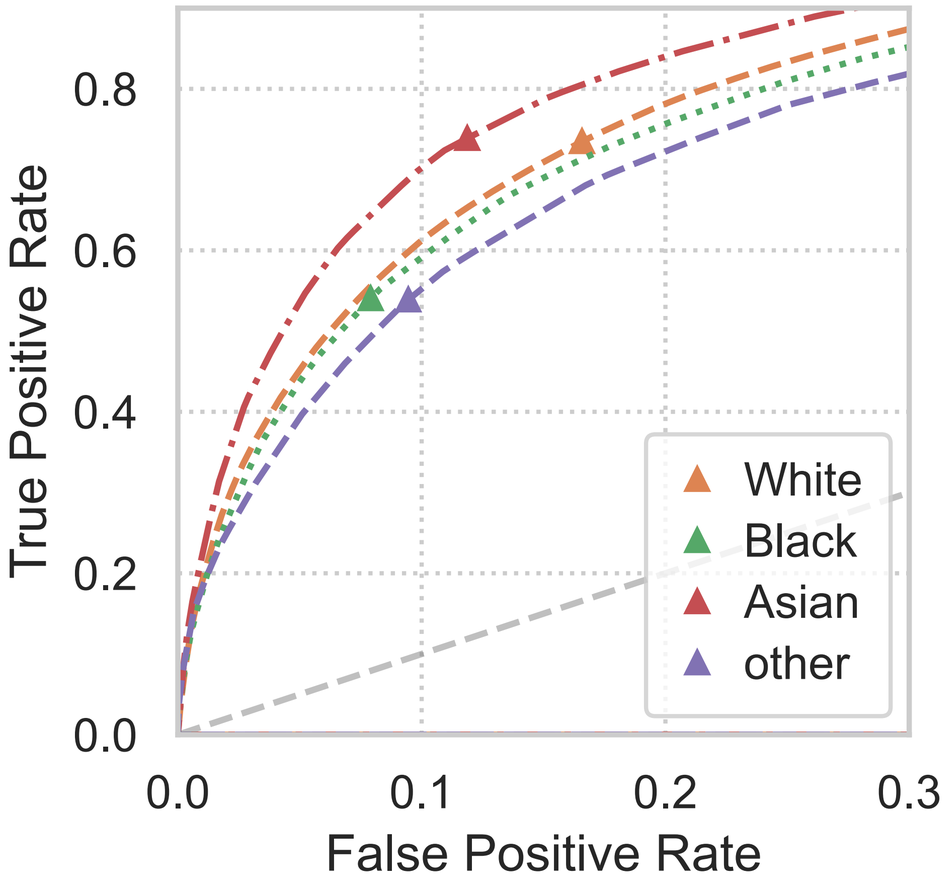
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

group-specific ROC curve

with group-specific threshold t_s



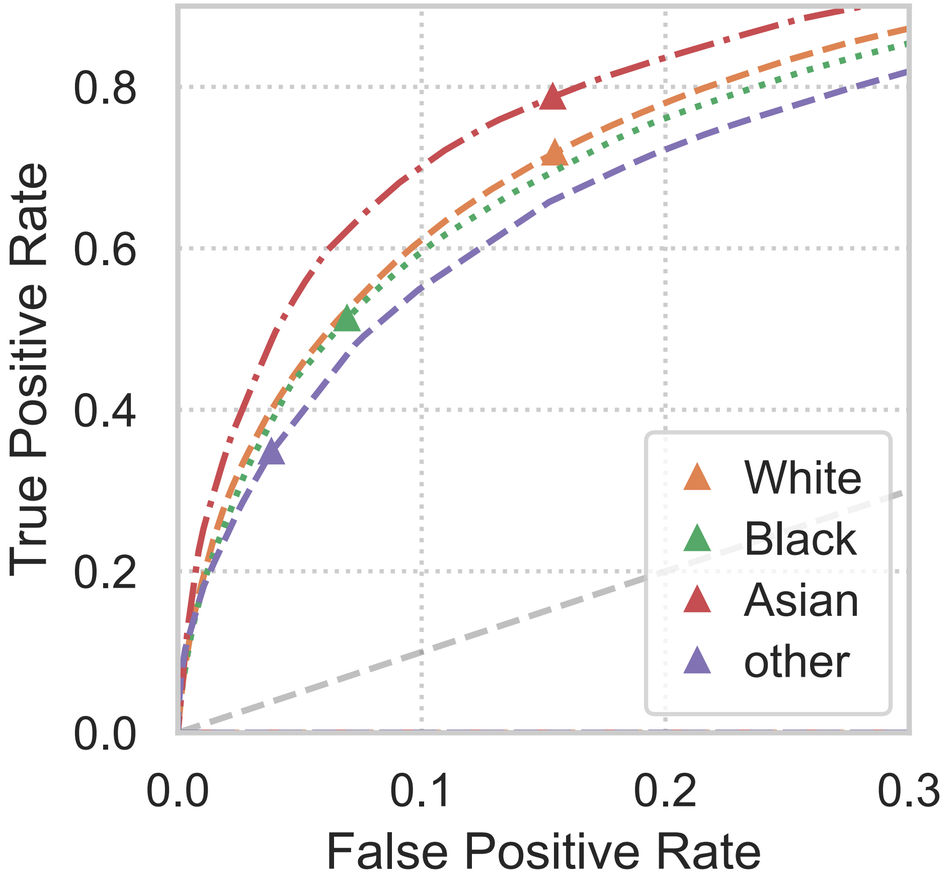
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

group-specific ROC curve

with group-specific threshold t_s



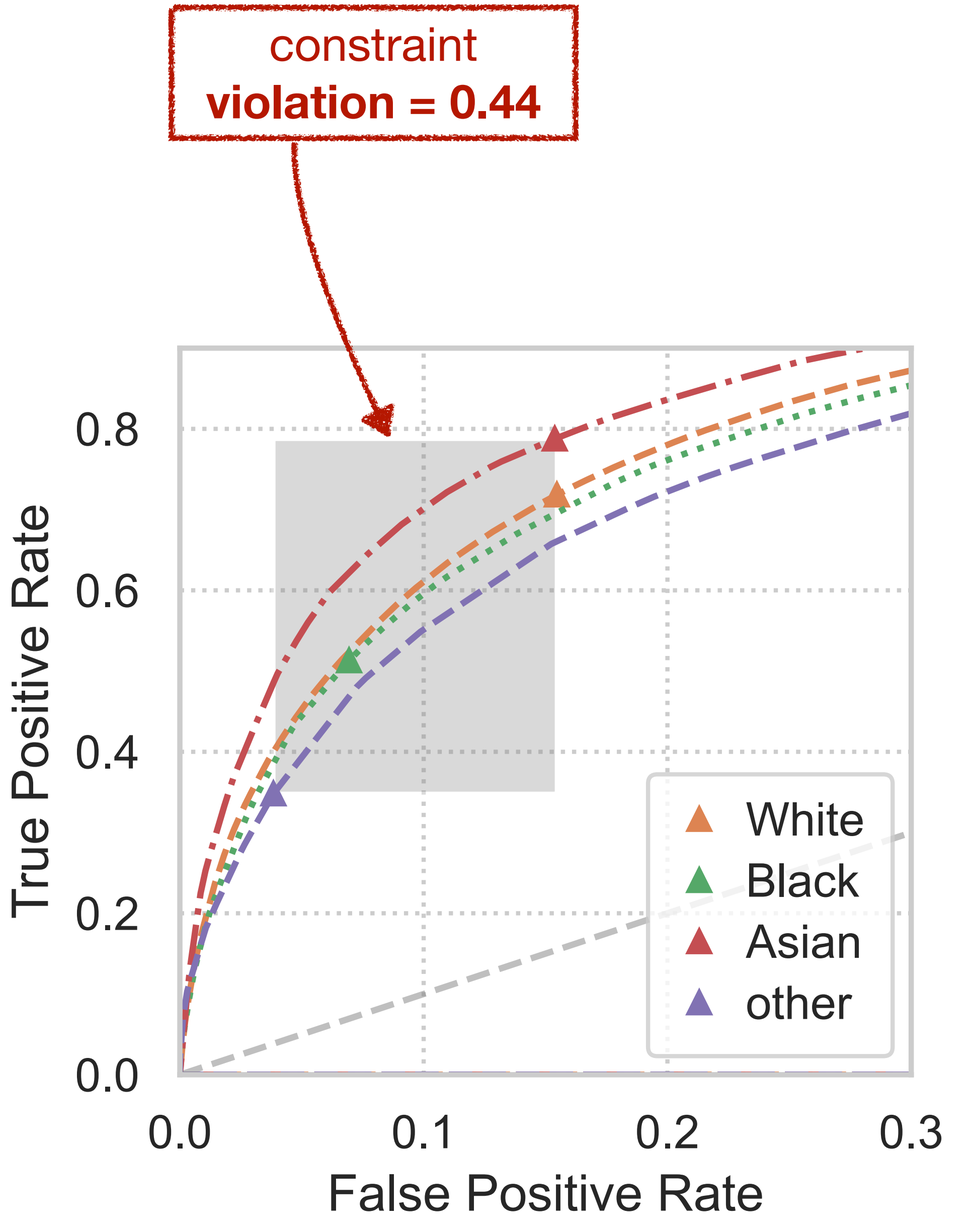
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

group-specific ROC curve

with group-specific threshold t_s



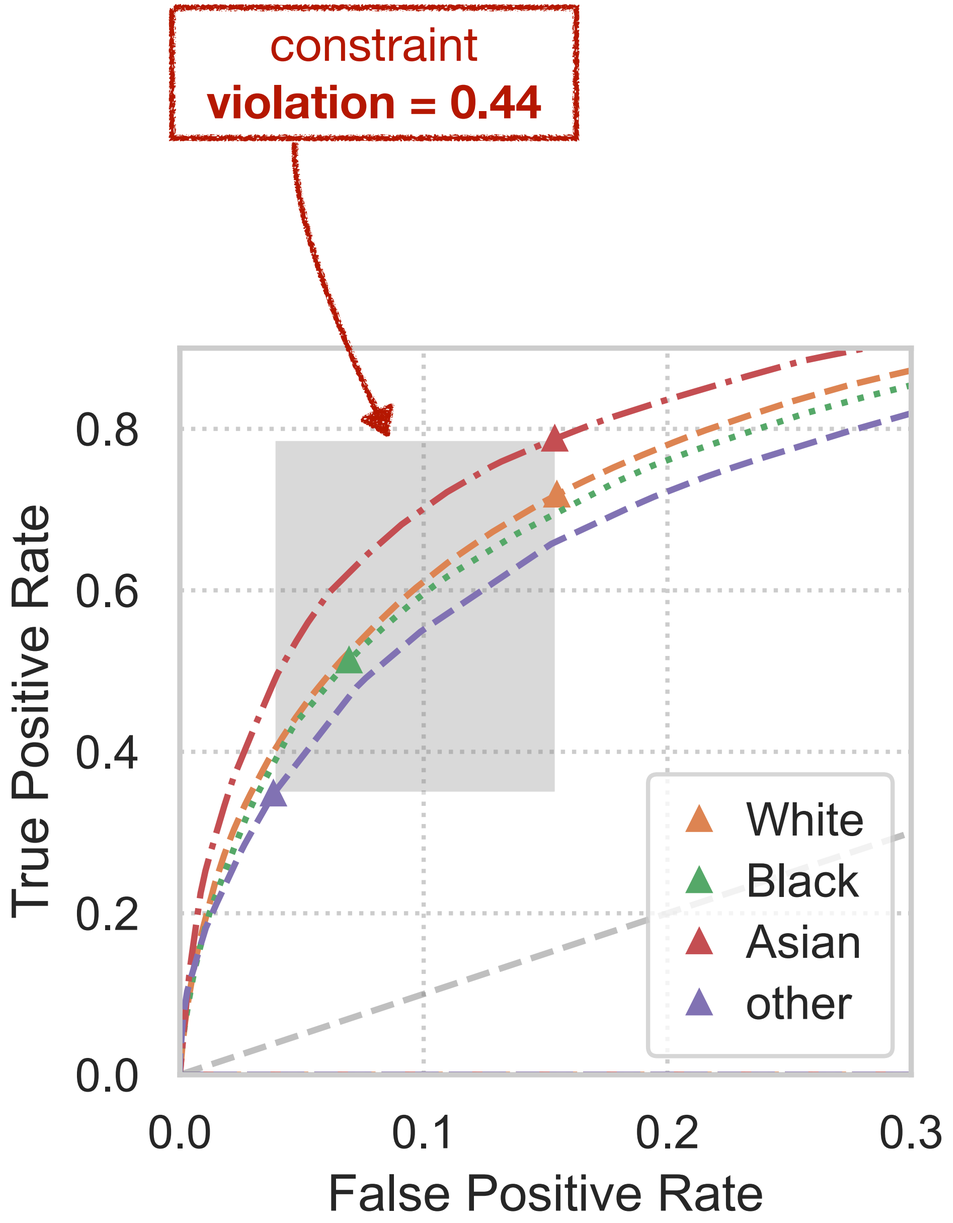
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

- Optimize over the space of group-specific thresholds $\tau \in \mathcal{T}$,

$$\min_{\tau \in \mathcal{T}} \ell(Y, \hat{Y}(\tau)) \;\; \textit{subject to fairness constr.};$$



[Hardt et al. “Equality of opportunity in supervised learning." *NeurIPS*, 2016]

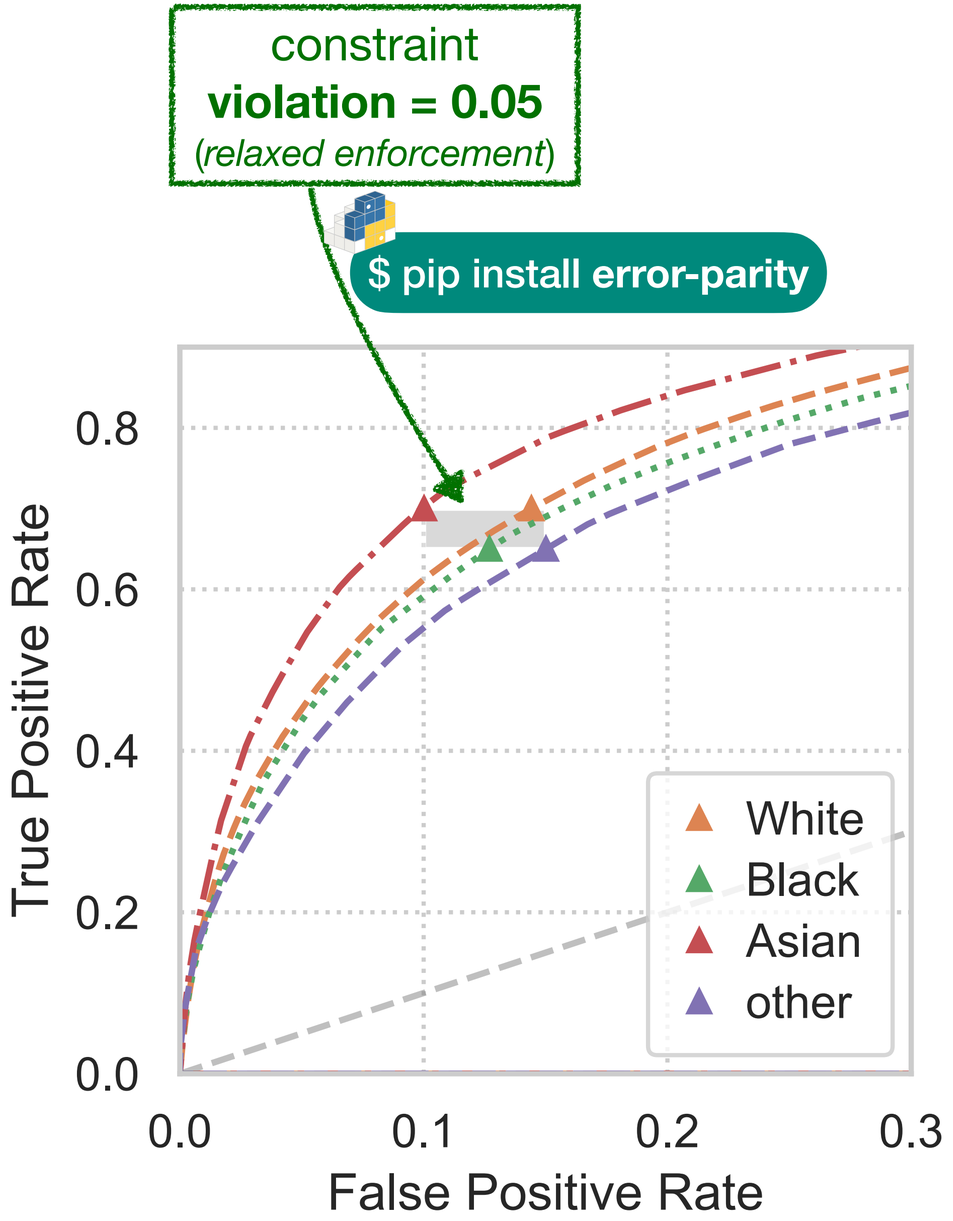
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

- Optimize over the space of group-specific thresholds $\tau \in \mathcal{T}$,

$$\min_{\tau \in \mathcal{T}} \ell(Y, \hat{Y}(\tau)) \;\; \textit{subject to} \text{ fairness constr.};$$



$ pip install **error-parity**

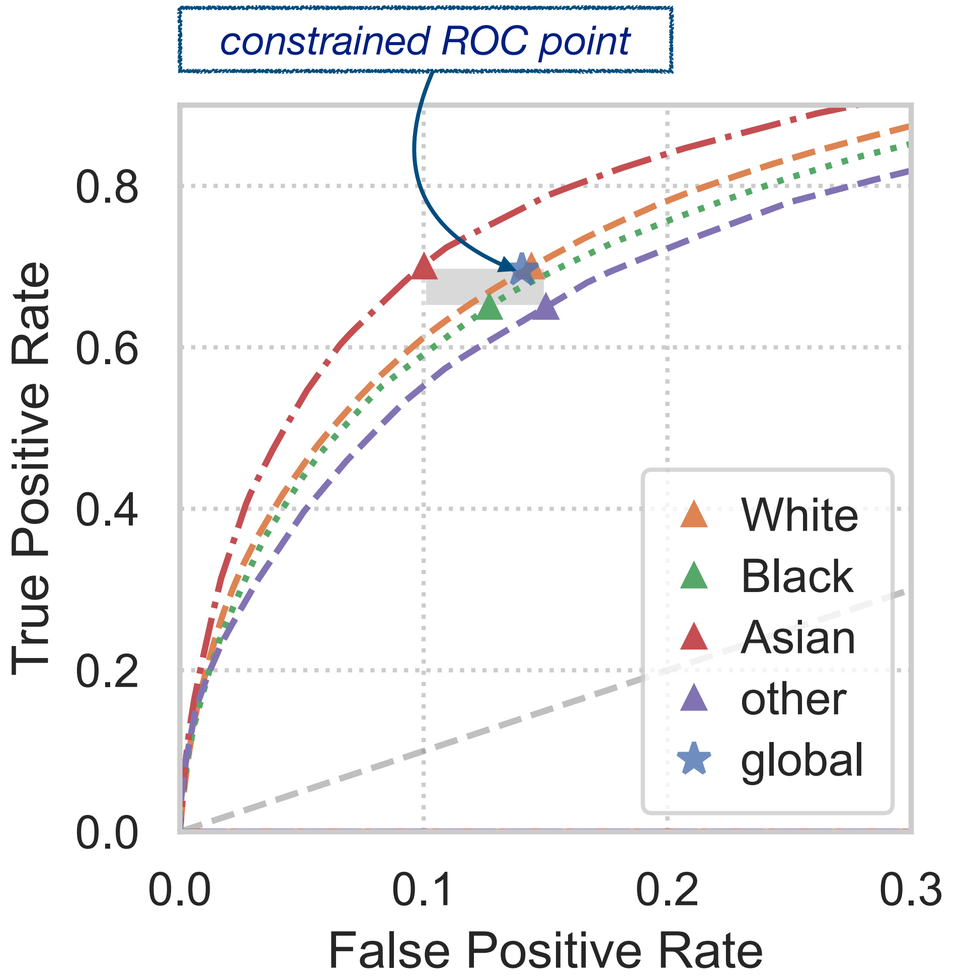
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

- Optimize over the space of group-specific thresholds $\tau \in \mathcal{T}$,

$$\min_{\tau \in \mathcal{T}} \ell(Y, \hat{Y}(\tau)) \;\; \textit{subject to} \text{ fairness constr.;}$$



$ pip install **error-parity**

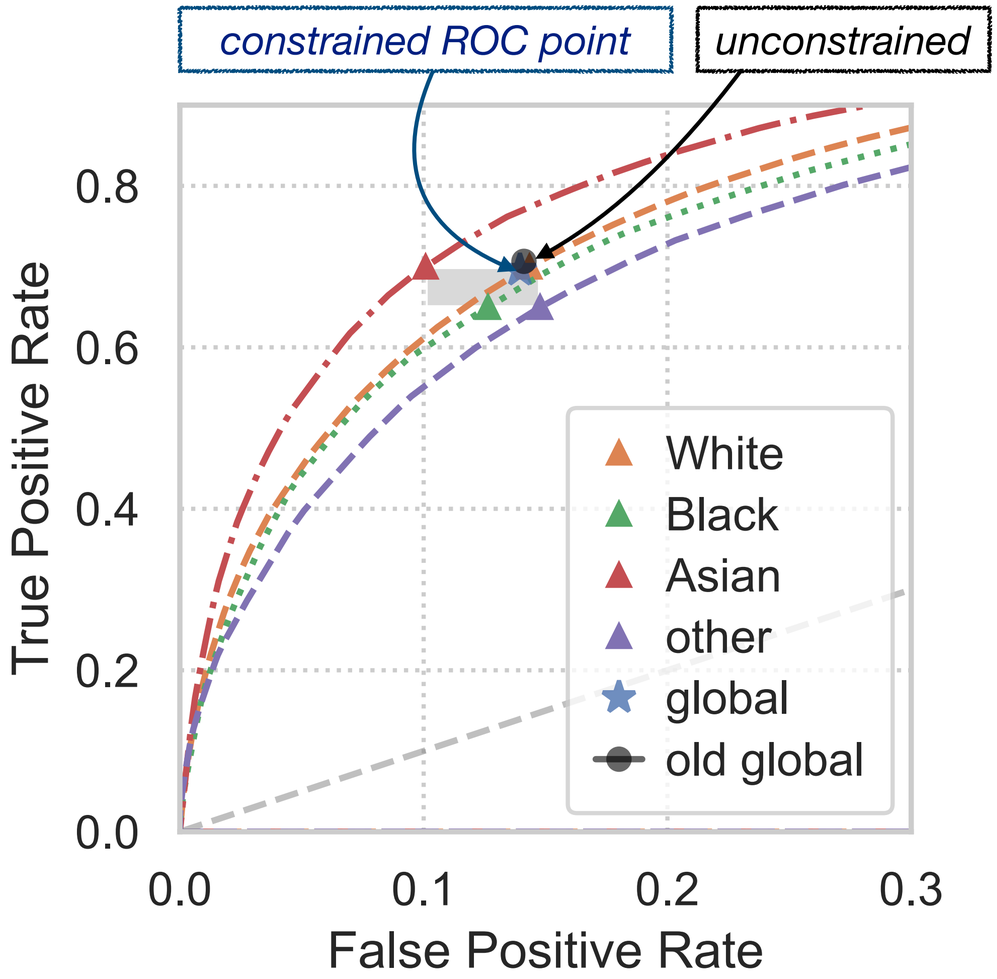
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

- Optimize over the space of group-specific thresholds $\tau \in \mathcal{T}$,

$$\min_{\tau \in \mathcal{T}} \ell(Y, \hat{Y}(\tau)) \quad \textit{subject to} \text{ fairness constr.;}$$



$ pip install **error-parity**

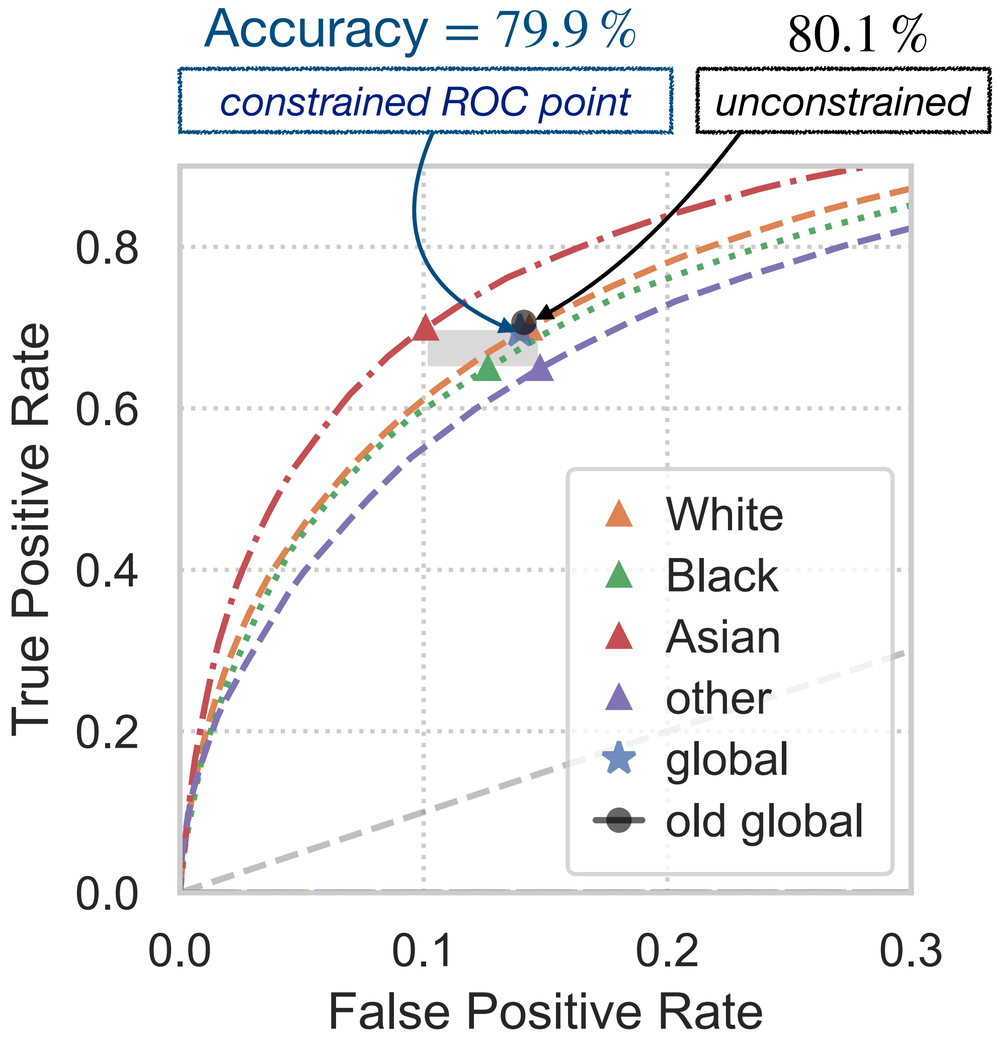
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s | \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

- Optimize over the space of group-specific thresholds $\tau \in \mathcal{T}$,

$$\min_{\tau \in \mathcal{T}} \ell(Y, \hat{Y}(\tau)) \ \textit{subject to}\ \text{fairness constr.;}$$



$ pip install **error-parity**

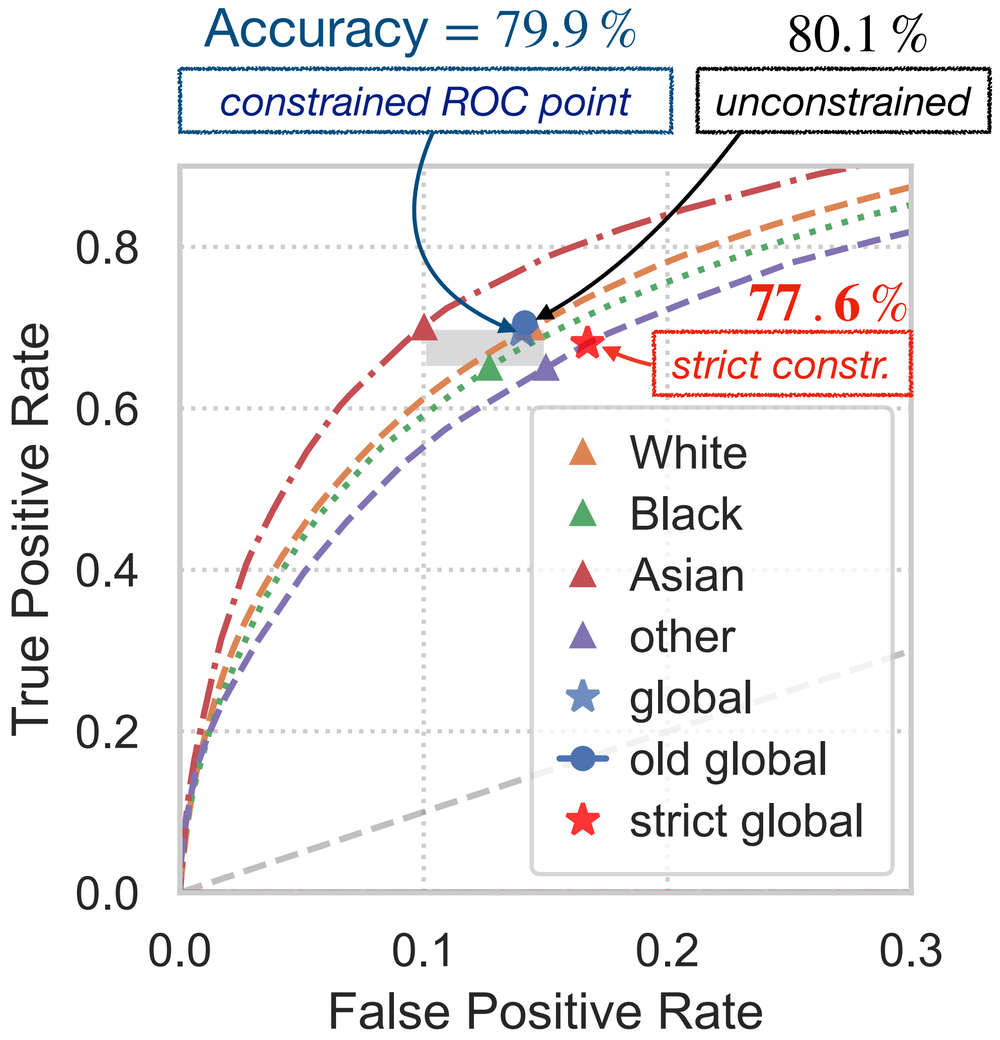
Postprocessing

- Choose point on group-ROC curve by changing group-threshold t_s ,

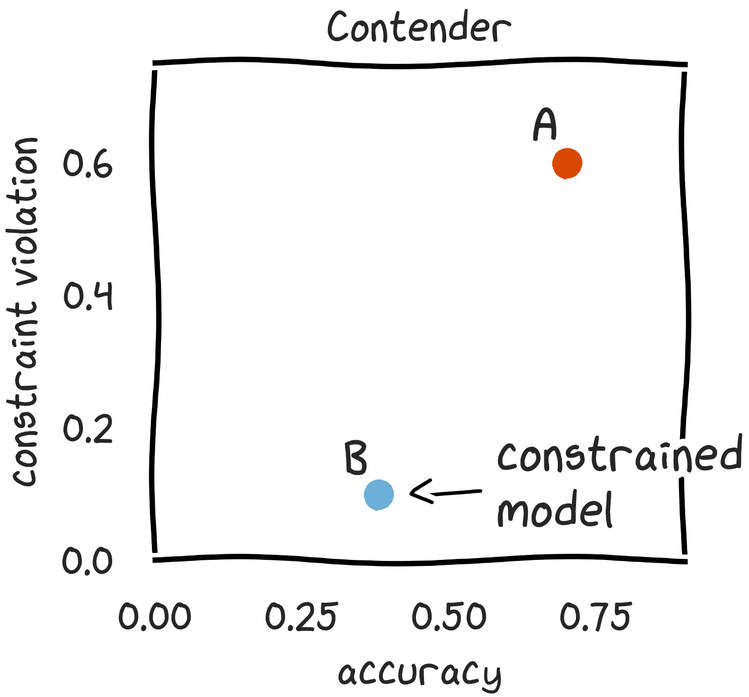
$$C_s(t_s) = \left(\overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 0\right]}^{FPR_s}, \overbrace{\mathbb{P}\left[\hat{R} \geq t_s \,|\, \mathbf{S} = \mathbf{s}, Y = 1\right]}^{TPR_s} \right),$$

- Optimize over the space of group-specific thresholds $\tau \in \mathcal{T}$,

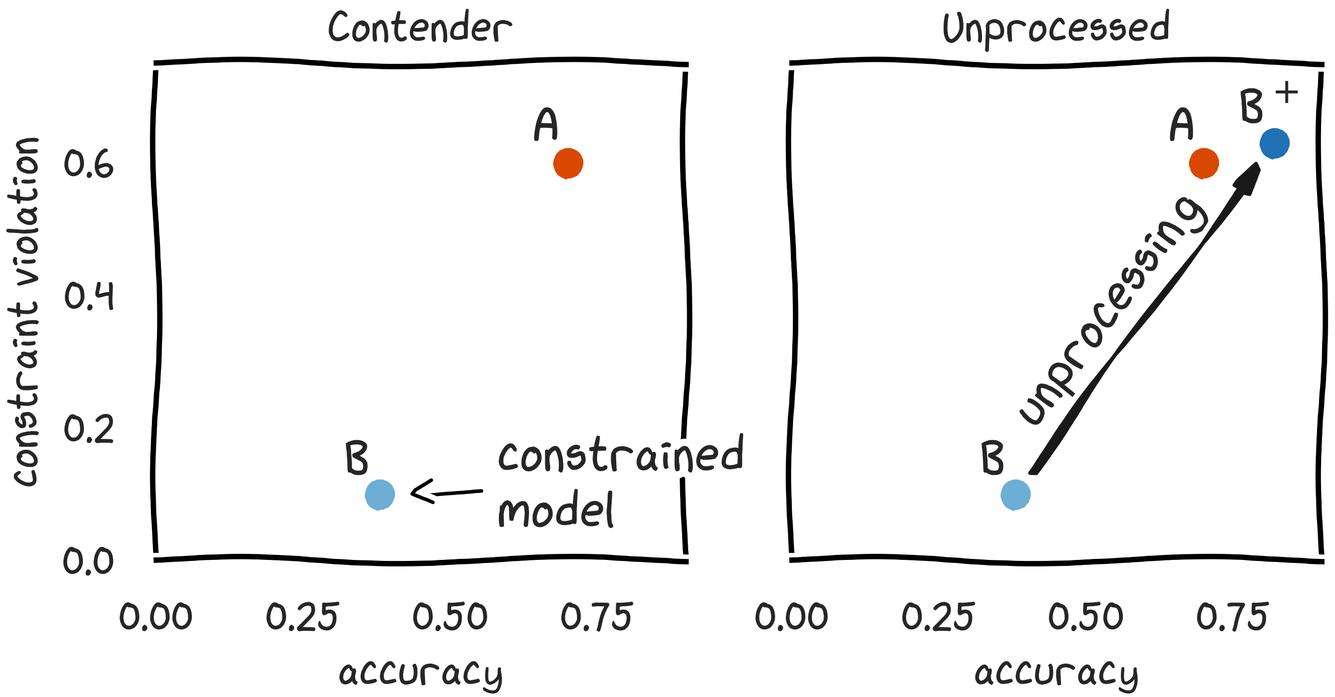
$$\min_{\tau \in \mathcal{T}} \ell(Y, \hat{Y}(\tau)) \;\; \textit{subject to} \text{ fairness constr.;}$$



Comparing Contender Models



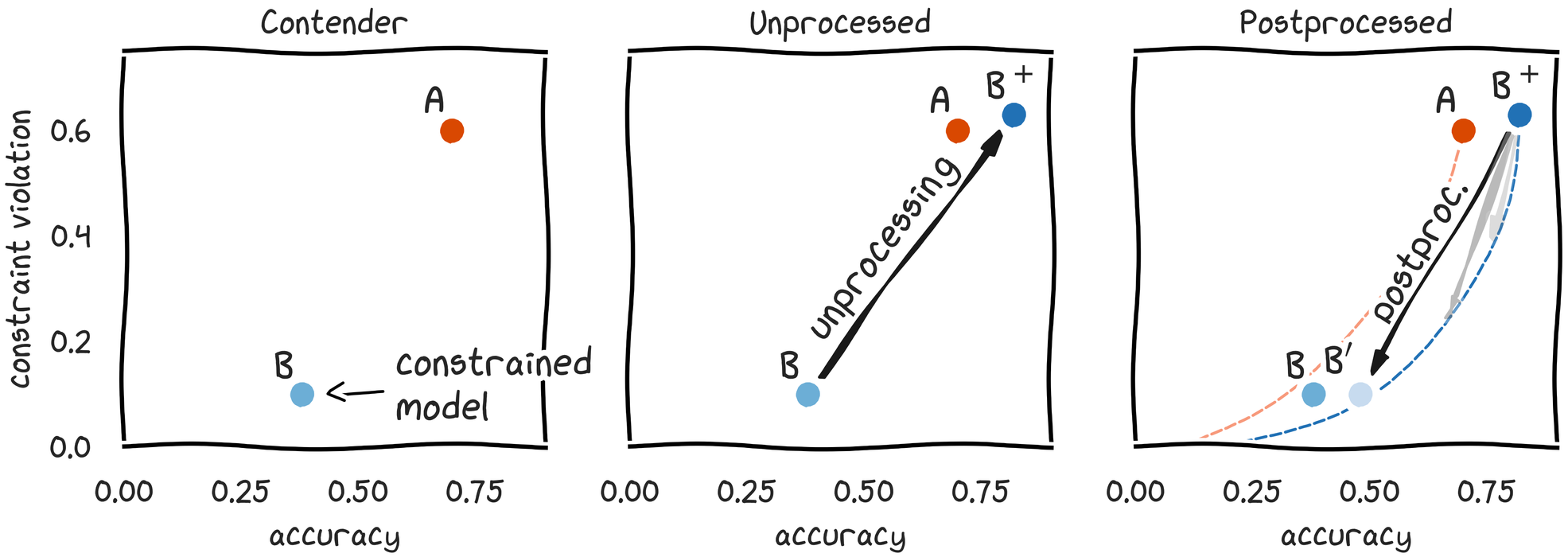
Comparing Contender Models



Unprocessing: unconstrained postprocessing.

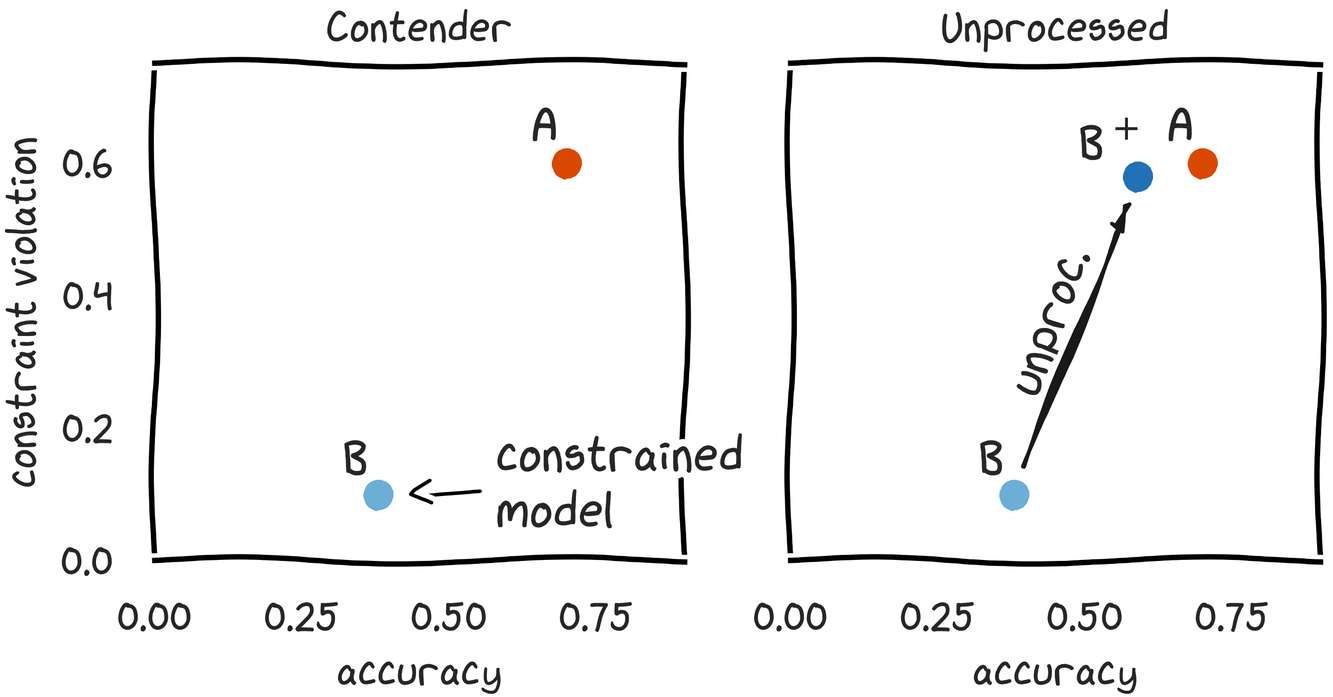
Comparing Contender Models

$$B \succ A$$



Unprocessing: unconstrained postprocessing.

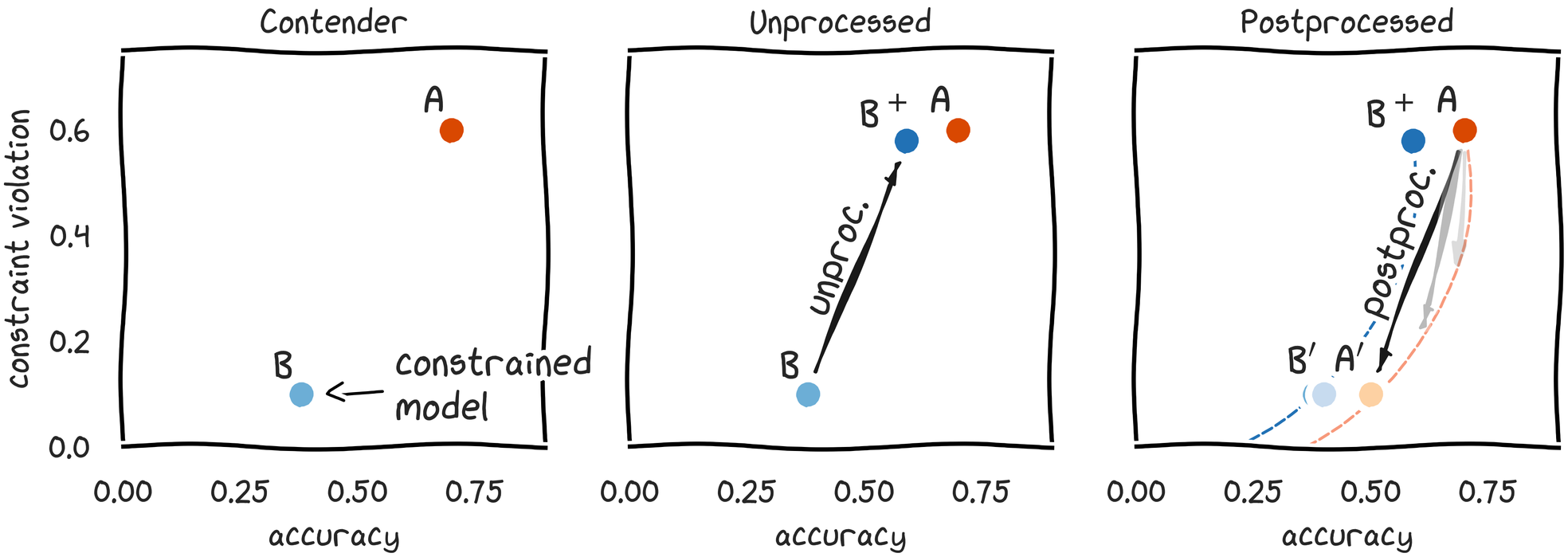
Comparing Contender Models



Unprocessing: unconstrained postprocessing.

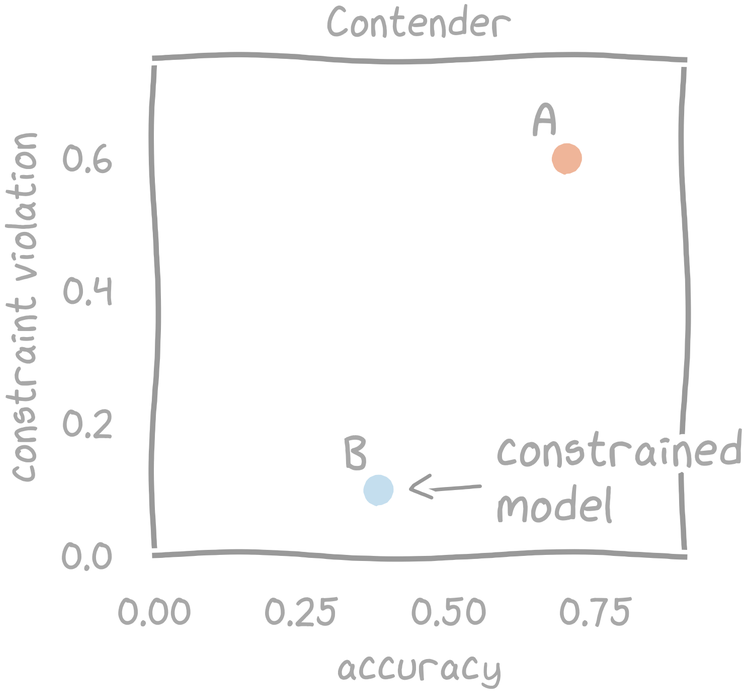
Comparing Contender Models

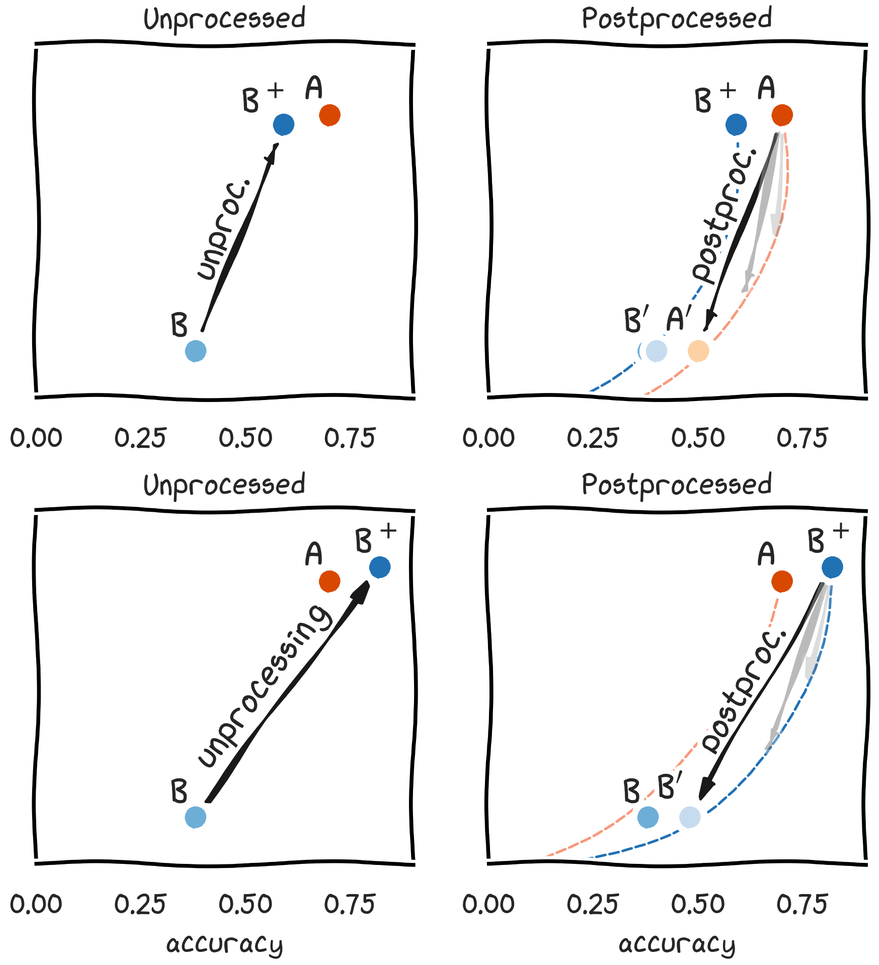
$$A \succ B$$



Unprocessing: unconstrained postprocessing.

Comparing Contender Models



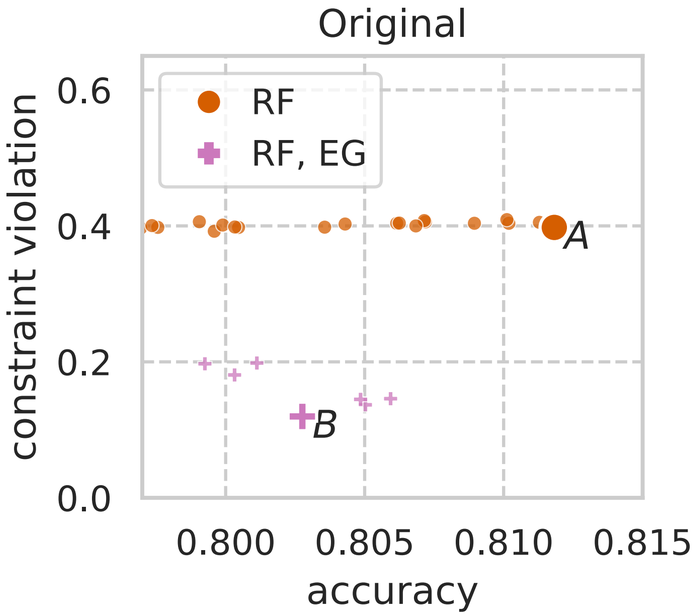


$A \succ B$

$B \succ A$

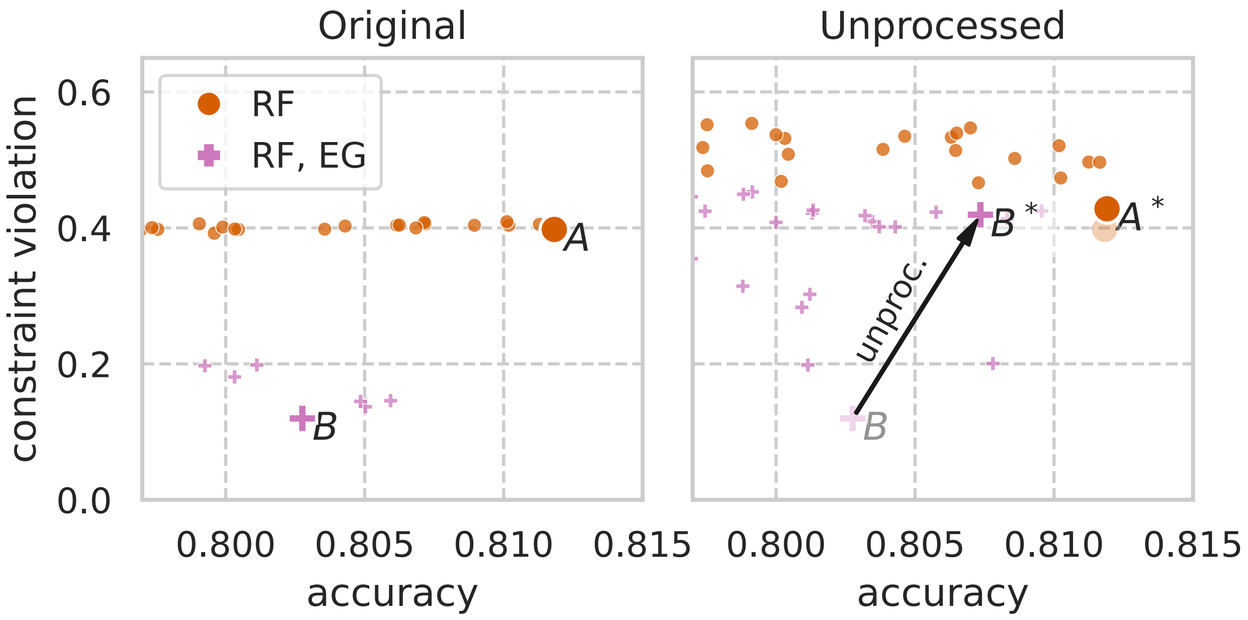
Unprocessing: unconstrained postprocessing.

Comparing Contender Models



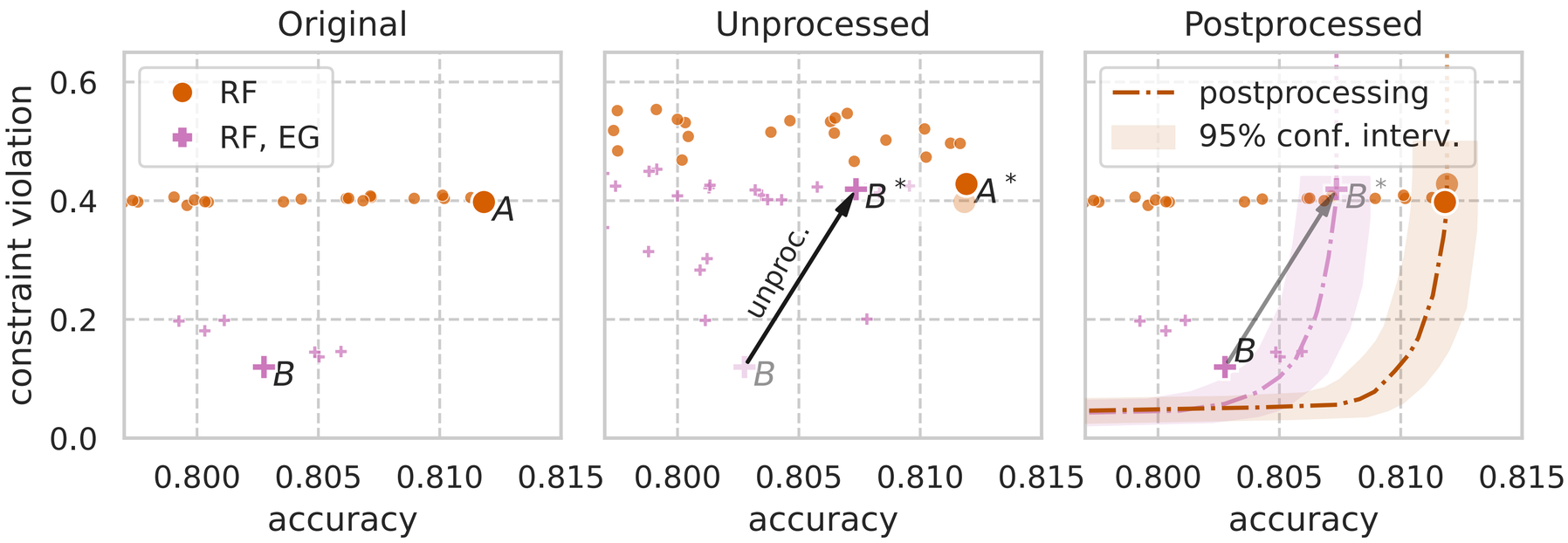
[Agarwal et al. "A reductions approach to fair classification." *ICML*, 2018]

Comparing Contender Models



[Agarwal et al. "A reductions approach to fair classification." *ICML*, 2018]

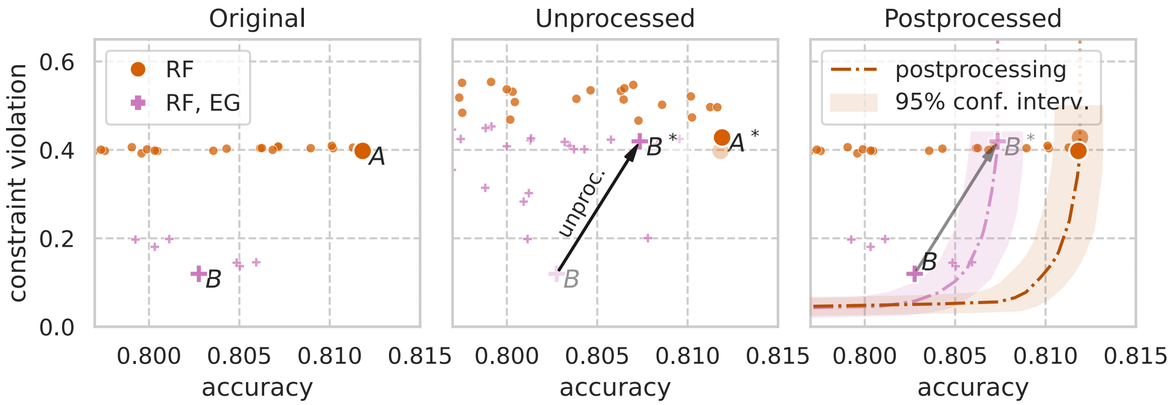
Comparing Contender Models $A \succ B$



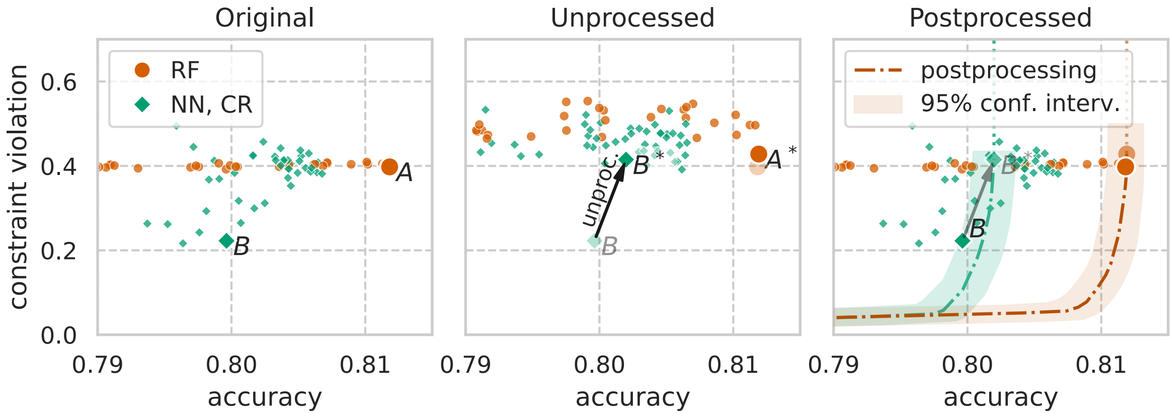
[Agarwal et al. "A reductions approach to fair classification." *ICML*, 2018]

$ pip install error-parity

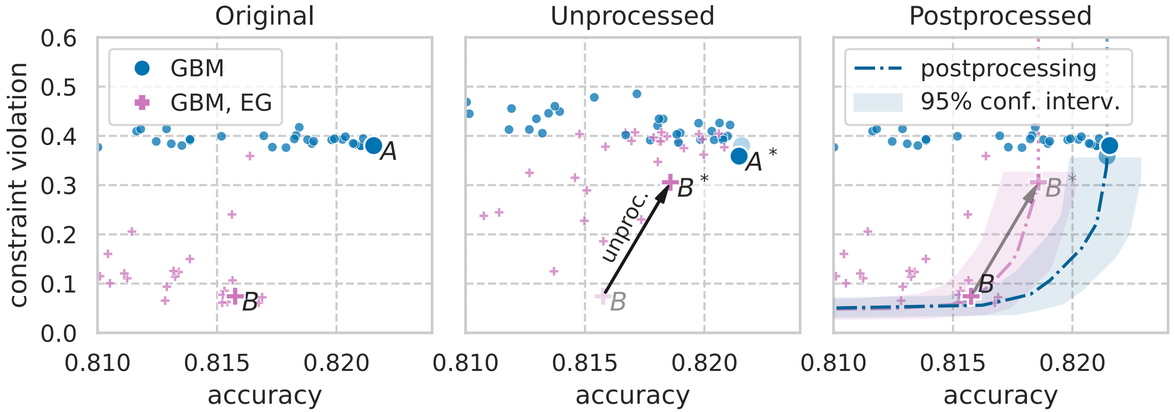
RF vs RF+EG



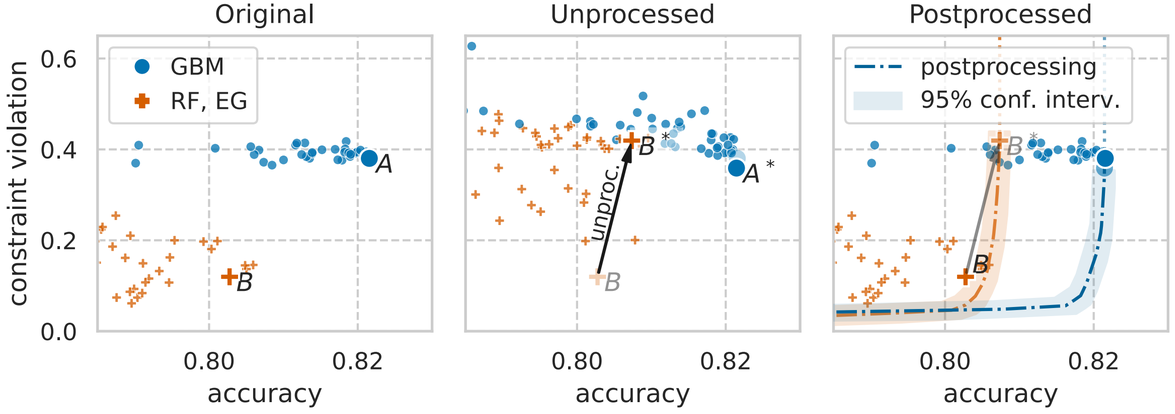
RF vs NN+CR



GBM vs GBM+EG



GBM vs RF+EG

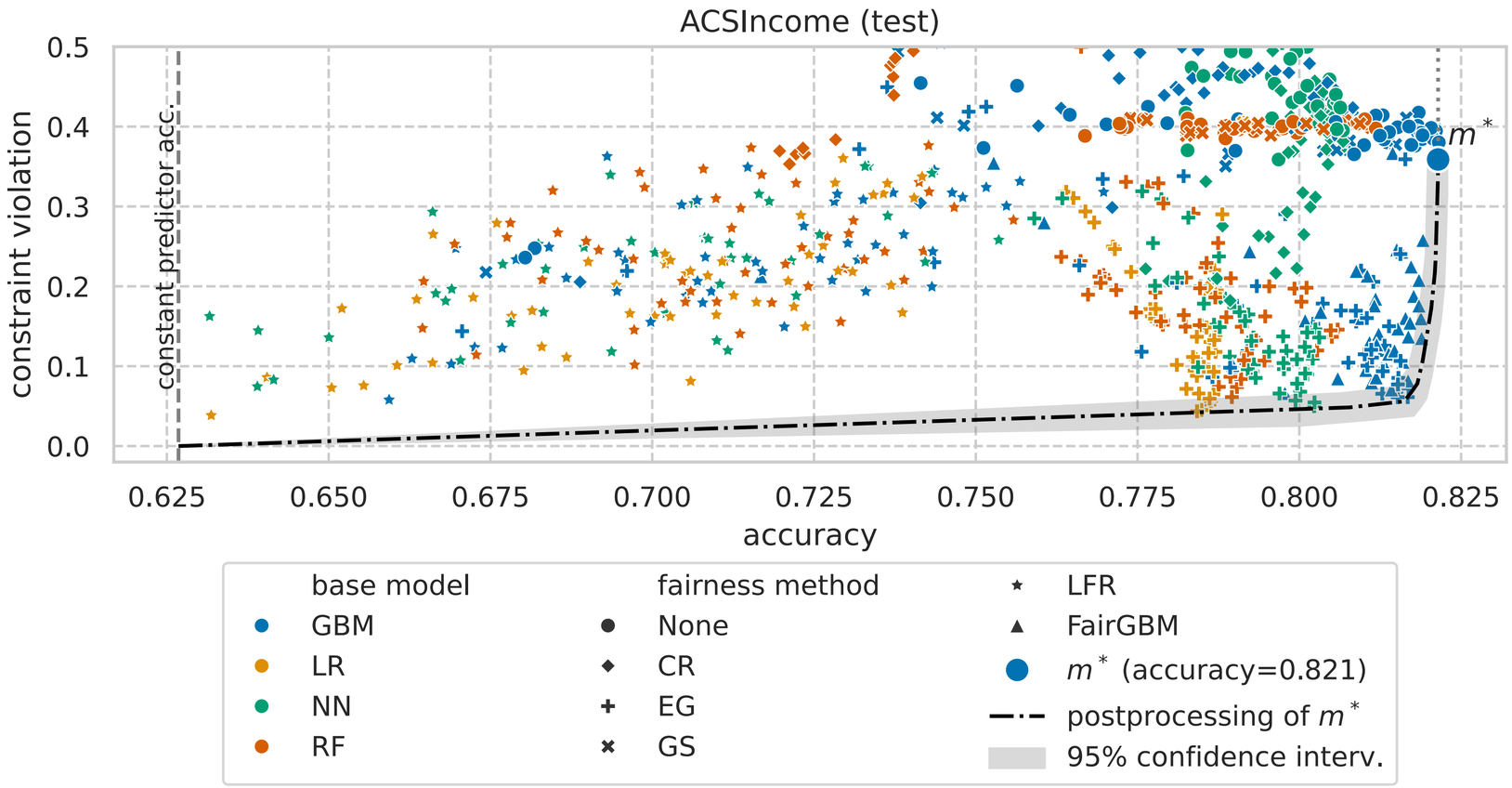


Datasets: Using **folktables** python package

[Ding et al. “Retiring adult: New datasets for fair machine learning.” *NeurIPS*, 2021]

Results on ACSIncome

Using race as the sensitive attribute (4 groups)



Thank You!

Code

$ pip install **error-parity**

Paper

Poster

Session #5

Halle B

#229

Cruz and Hardt. “Unprocessing Seven Years of Algorithmic Fairness.” ICLR 2024.