



Fairness

Como avaliar e mitigar vieses em
algoritmos de Aprendizado de Máquina

Willian Dihanster Gomes de Oliveira

Mestrando @ Universidade Federal de São Paulo - UNIFESP
Data Scientist @ Serasa Experian DataLab

Agenda



imagens de freepik.com



Contexto

Por que é necessário analisar e mitigar vieses em algoritmos
Aprendizado de Máquina?

Fairness

O que é? Para que serve? Quais são os tipos de análises e
métricas de Fairness

Mitigação

Como mitigar vieses e obter modelos mais justos?

Implementações

Implementações de bibliotecas para análise e mitigação de Fairness

Considerações Finais

Então, como considerar Fairness no desenvolvimento de modelos?

Aprendizado de Máquina (AM)



Algoritmos de AM estão cada vez mais populares no contexto de auxílio na **tomada de decisão**. Alguns exemplos são:



Concessão de Crédito

Algoritmos do tipo **score**, atribuem uma nota com base no **perfil de risco** do indivíduo, a fim de decidir **conceder crédito ou não**.



Recrutamento

Algoritmos ajudam no processo de recrutamento **selecionando as pessoas candidatas mais prováveis de serem contratadas**.

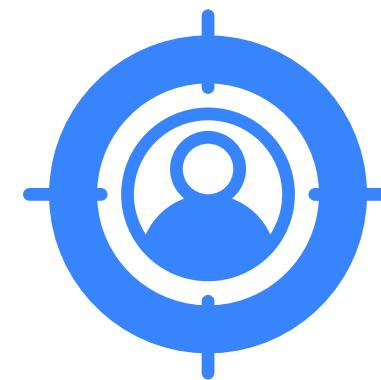


Tratamento Médico

Algoritmos podem ajudar a **classificar pessoas que estão doentes** e indicar a **priorização** para tratamento médico.

Vieses em AM

Modelos de AM criam **generalizações com base nos dados de entrada**, que não estão livre de vieses. Existem diversos **casos de injustiças (unfairness)** reportados na literatura:



Recrutamento

Algoritmo de recrutamento da Amazon foi acusado de **penalizar candidatas do sexo feminino**, por ter sido treinado sob currículos predominantemente de pessoas do sexo masculino.



Reincidência de Crimes

O algoritmo utilizado pelo **sistema judicial americano** para prever reincidência de crime, o COMPAS, foi acusado de **errar 2 vezes mais para pessoas afro-americanas**.



Reconhecimento Facial

Um **estudo do National Institute of Standards and Technology (NIST)** encontrou que certos algoritmos de **reconhecimento facial erravam cerca de 100 vezes mais para pessoas negras**.

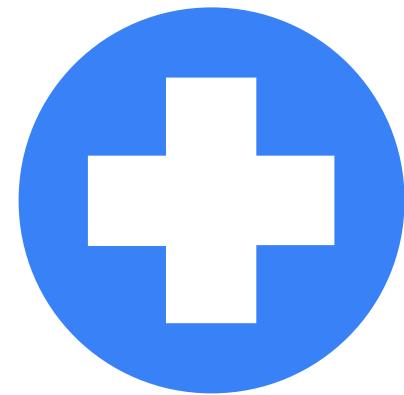
Vieses em AM

Existem diversos **casos de injustiças (unfairness)** reportados na literatura:



Linguagem

O Google Tradutor foi acusado de alterar o gênero em traduções quando a palavra **possui estereótipos**. Ex: "a presidente" era traduzido para "o presidente", "o enfermeiro" para "a enfermeira", em algumas línguas.



Saúde

O algoritmo de health care da Optum, atribuía **mesmo nível de risco** para pacientes brancos e negros, sendo que em geral, os **pacientes negros estavam mais doentes**.



Crédito

Modelo de crédito do Apple Card foi acusado de ser **sexista contra mulheres**. Para um casal, compartilhando dados financeiros, morando juntos e a mulher com maior score de crédito, havia diferença de até **20x mais limite para o homem**.



Fairness

É um tópico em Aprendizado de Máquina, onde o objetivo é analisar, entender, e corrigir vieses em algoritmos inteligentes.

Por que Fairness é importante

Não Discriminar

É importante que o modelo de AM não discrimine por características pessoais.



Não Piorar Vieses

É importante que o modelo não piore os vieses a qual foi treinado.

Não Criar Vieses

É importante que o modelo não acabe criando novos vieses com base nos dados de entrada.

Não Perpetuar Vieses

É importante que o modelo não sirva como um perpetuador de vieses.

Fonte de Vieses

Em geral, os vieses encontrados nesses modelos veem do **viés contido no conjunto de dados**. Por exemplo:

Amostra Desbalanceada e Tamanho da Amostra

- Amostra possui mais casos de sucesso ou mais exemplos de um determinado grupo. **Ex: Algoritmo de recrutamento da Amazon.**

Fatores Históricos

- Historicamente um grupo possui mais privilégios em relação ao outro. **Ex: Desigualdade salarial entre homens e mulheres.**

Qualidade dos Dados

- A qualidade do dado pode ser pior para um grupo, sendo menos informativa/confiável. **Ex: Dados financeiros para pessoas jovens.**

Features Proxy

- Uma feature pode introduzir viés por ser um proxy do atributo sensível. **Ex: CEP pode ser um proxy de raça.**



Leis Contra Discriminação em Crédito



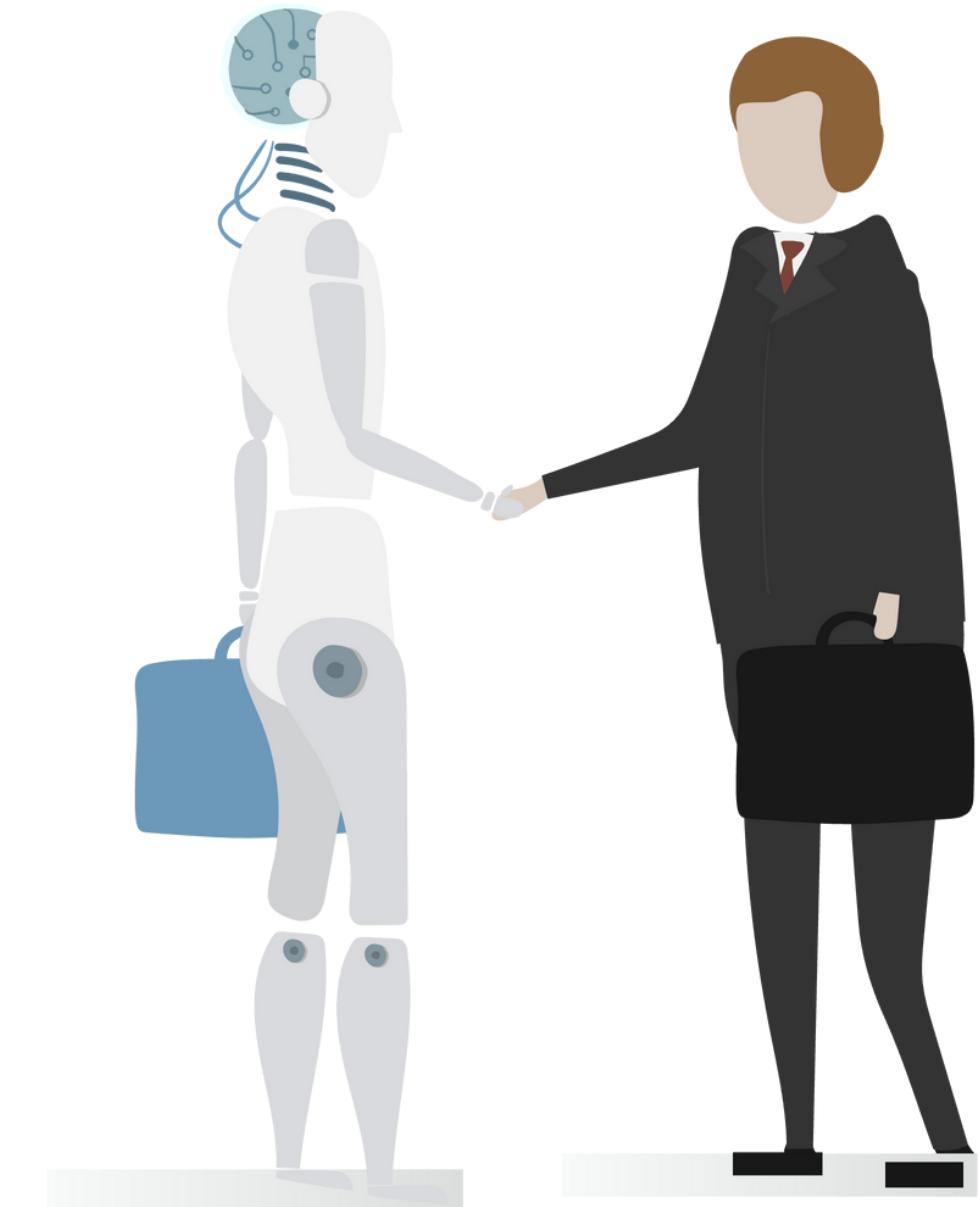
Para crédito, nos Estados Unidos, a **Equal Credit Opportunity Act (ECOA)** proíbe explicitamente discriminação baseada em:

- Sexo (incluindo gênero, identidade de gênero, orientação sexual...)
- Idade
- Etnia
- Religião
- Nacionalidade
- Estado Civil

Fair Credit Reporting Act (FCRA) é uma lei federal que defende a acurácia, fairness e privacidade de informação de bureus de crédito.



Para o Brasil, não foram encontradas leis específicas para crédito. Porém, existem um **Projeto de Lei n° 4529, de 2021**, que busca diminuir discriminação racial no acesso ao crédito, obrigando que as instituições divulguem o motivo de recusa ao crédito.



Principais Conceitos



Rótulo Favorável

É o **rótulo/classe que tem uma saída favorável**, isto é, dá uma vantagem ou benefício a um indivíduo.

Ex: crédito aprovado; chamado para entrevista.



Grupo Protegido

Grupos de indivíduos que possuem uma **mesma característica pessoal**. Grupo **Privilegiado** é o grupo que possui maior probabilidade de receber o rótulo favorável. Enquanto o Grupo **Não-Privilegiado** possui menor probabilidade.



Atributo Protegido (Atributo Sensível)

É o **atributo que divide os indivíduos entre Grupo Privilegiado e Não-Privilegiado** e geralmente são **características pessoais**.

Ex: Sexo; Idade; Raça; Religião, etc.



Exemplo em Risco em Crédito



Risco de Crédito

O algoritmo atribui um **score**, que é uma nota com base no **perfil de risco** do indivíduo, a fim de decidir **conceder crédito ou não**.



Rótulo Favorável
Conseguir crédito.



Atributo Protegido
Conforme lei: gênero, idade, raça, religião, CEP, etc.



Grupo A (Não-Privilegiado)
Ex: 40%, em média, conseguem crédito.

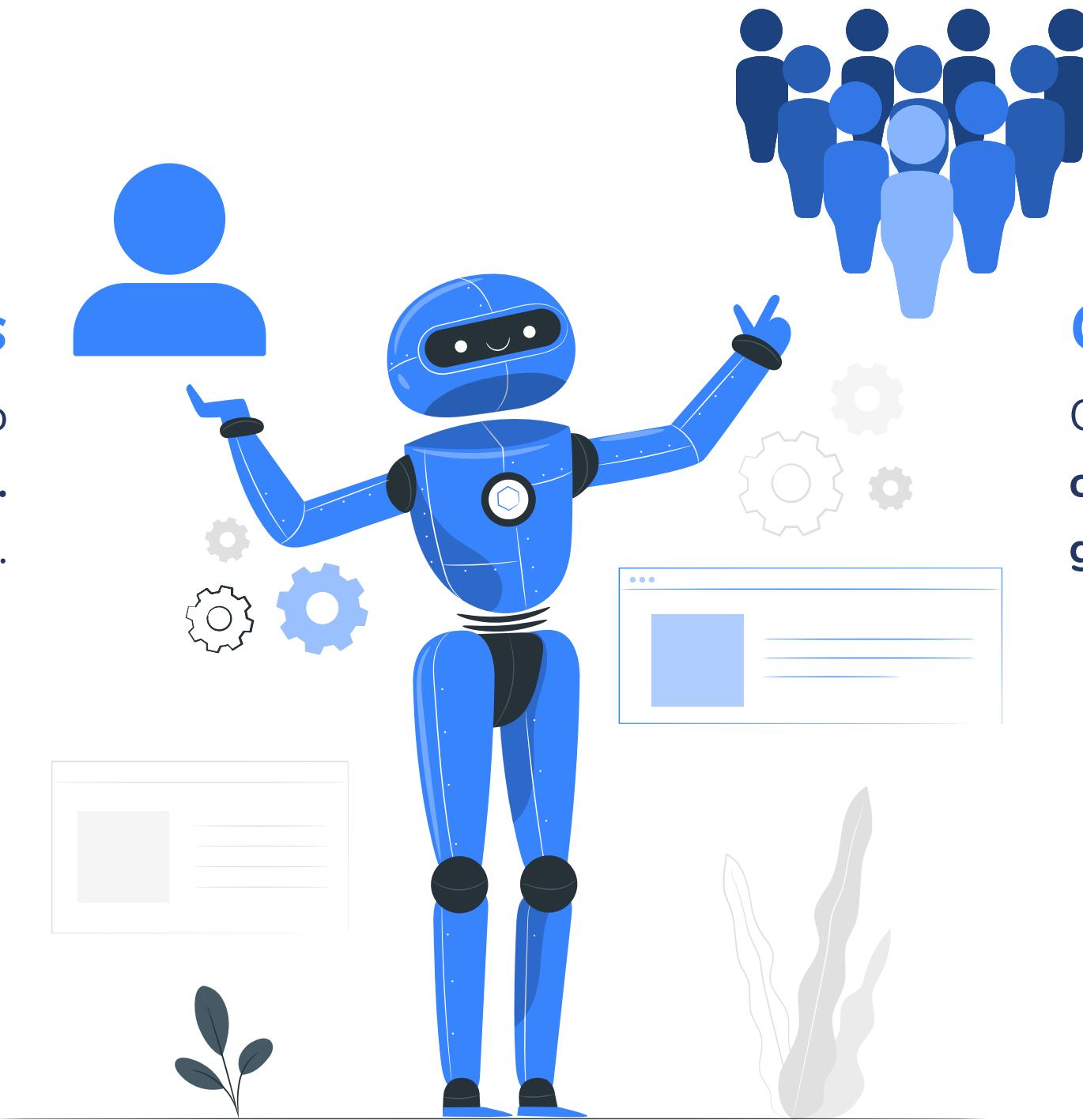


Grupo B (Privilegiado)
Ex: 50%, em média, conseguem crédito.

Tipos de Análise de Fairness

Individual Fairness

Indivíduos similares, exceto pelo atributo protegido, devem possuir saída similar. Geralmente, é mais difícil de alcançar.

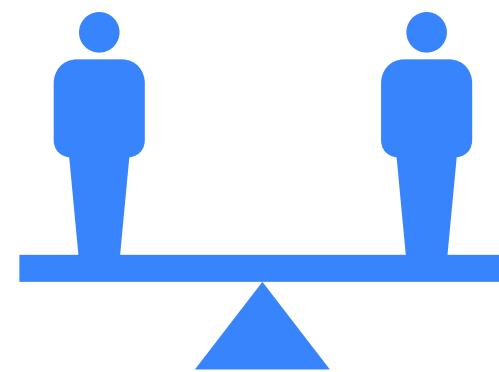


Group Fairness

O comportamento médio de um classificador deve ser similar entre os grupos. Mais utilizado.

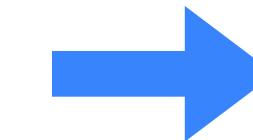
Métricas

Existem duas principais **categorias de métricas para Fairness** e a **decisão** de qual utilizar pode levar em consideração **vários fatores**: leis; domínio de negócio; objetivos de negócio, etc.



Métricas de Output

Útil quando a **saída do modelo deve ser igual ou similar** entre grupos protegidos.



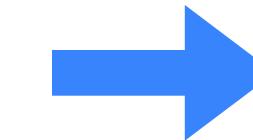
Recrutamento

Ex: Deve-se garantir **diversidade de gênero** no processo de **recrutamento de um time novo**. Pessoas candidatas, independente do gênero devem ter **mesma probabilidade** de serem chamados para entrevista.



Métricas de Erro

Útil quando a **taxa de erro/acerto do modelo deve ser igual** entre os grupos protegidos.
Não necessariamente a saída deve ser igual.



Previsão de Renda

Ex: **Prever a renda** de dois grupos protegidos, onde sabe-se que há **desigualdade salarial**. Logo, a **saída do modelo não precisa ser necessariamente igual**.
Mas é esperado que a **taxa de erro/acerto seja igual**.

Métricas de Output

Disparate Impact (DI) e **Statistical Parity Difference (SPD)** são dois exemplos de métricas que penalizam modelos que não possuem saída similar entre grupos protegidos. **Valor ideal para DI é 1, e para SPD é 0.**

$$DI = \frac{P(Y = 1 | \text{Grupo} = \text{Não-Privilegiado})}{P(Y = 1 | \text{Grupo} = \text{Privilegiado})}$$

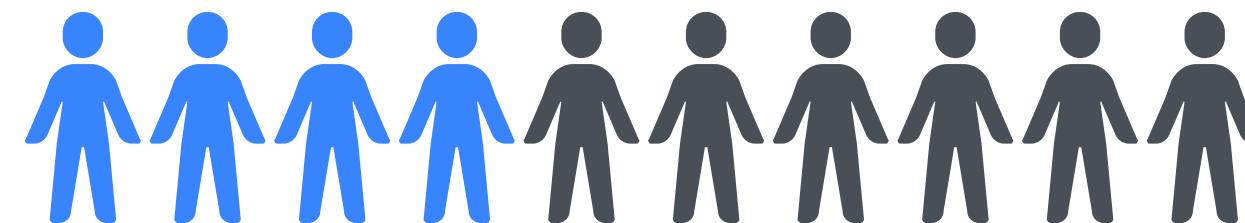
$$SPD = P(Y = 1 | \text{Grupo} = \text{Não-Privilegiado}) - P(Y = 1 | \text{Grupo} = \text{Privilegiado})$$

Exemplo

 Indivíduo com Rótulo Favorável

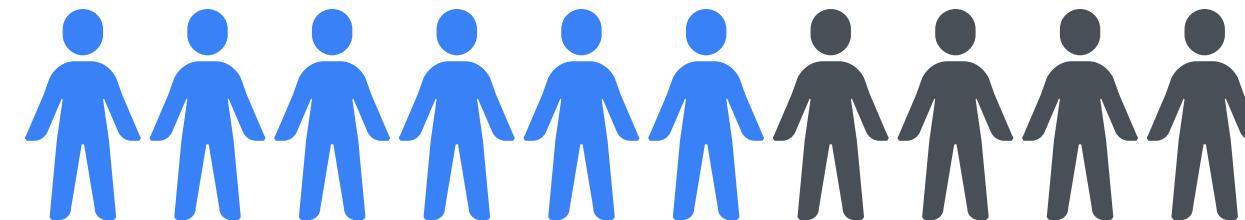
 Indivíduo com Rótulo Não-Favorável

Grupo A (Não-Privilegiado)



= 40%

Grupo B (Privilegiado)



= 60%



DI = 0.66

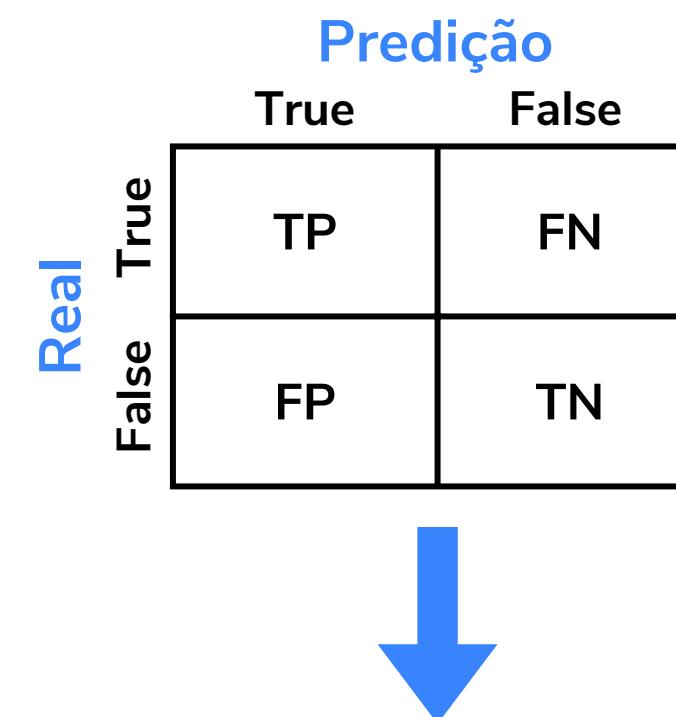
SPD = -20%

Métricas de Acerto/Erro

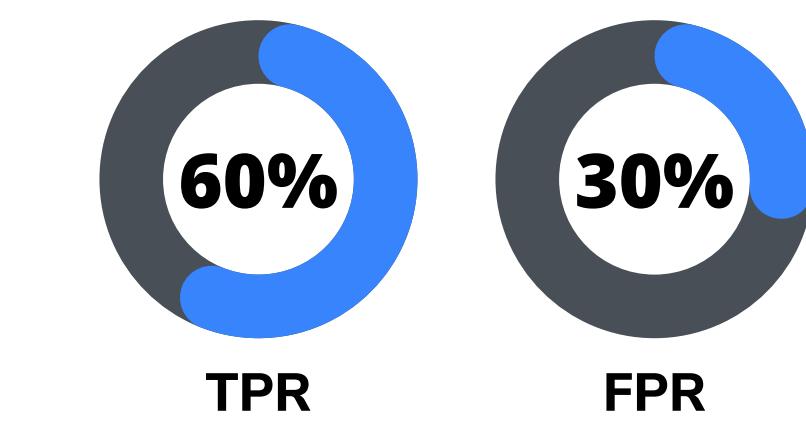
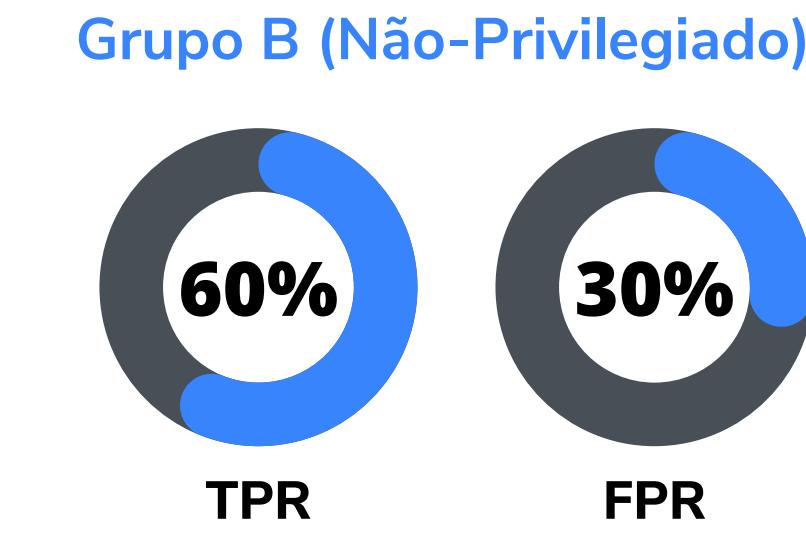
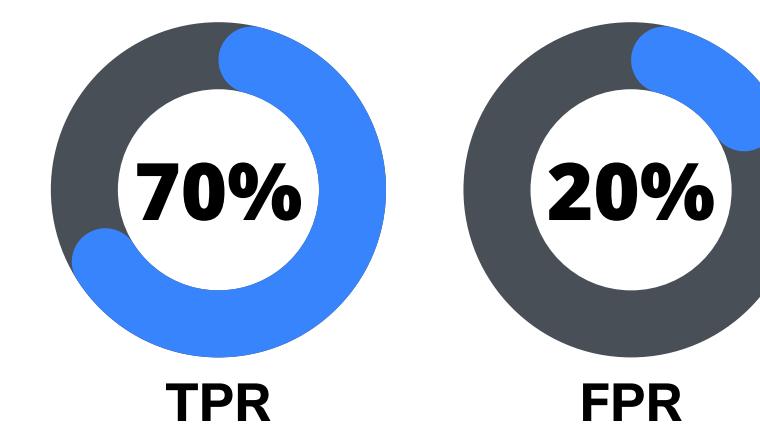
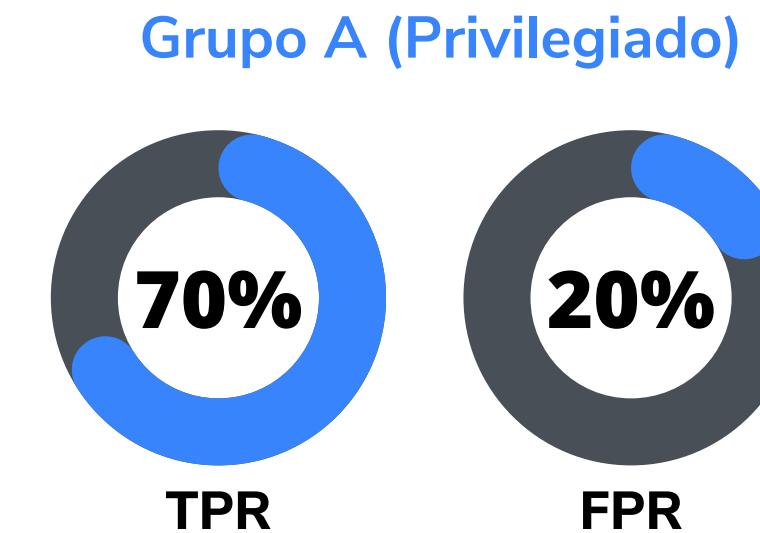
Average Odds Difference (AOD) e Equalized Odds (EOD) são dois exemplos de métricas que penalizam modelos que não possuem taxa de erro e/ou acerto similar entre grupos protegidos. O **valor ideal** para ambas métricas é **0**.

$$AOD = 0.5 * [(FPR_{\text{Não-Priv}} - FPR_{\text{Priv}}) - (TPR_{\text{Não-Priv}} - TPR_{\text{Priv}})]$$

$$EOD = [(TPR_{\text{Não-Priv}} - TPR_{\text{Priv}})]$$



$$TPR = TP / (TP + FN)$$
$$FPR = FP / (TN + FP)$$

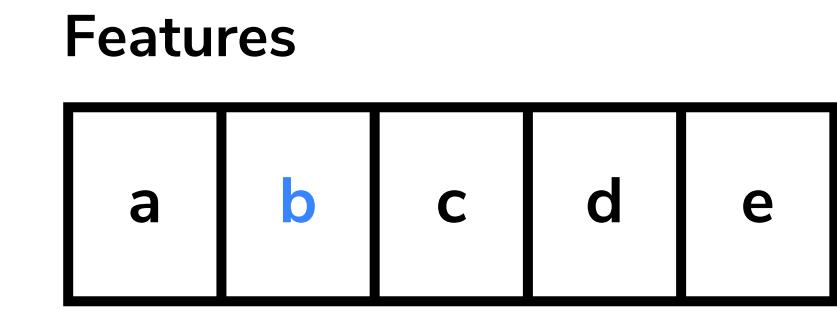
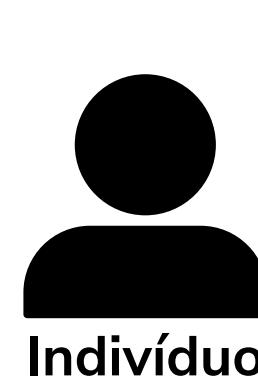


$$AOD = 10 \quad EOD = -10$$

Fairness + Explicabilidade

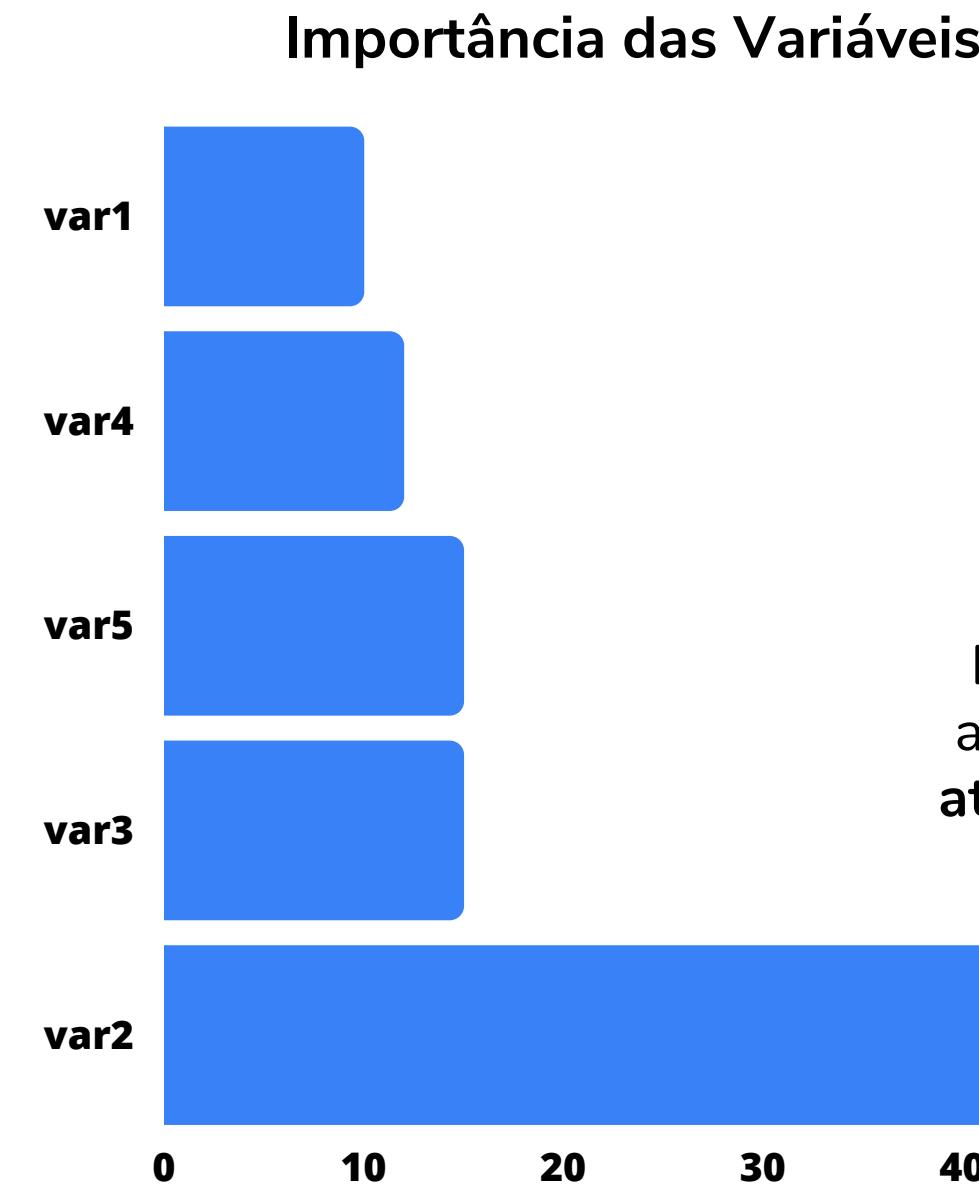
Explicabilidade pode ajudar a **entender e encontrar vieses** em modelos de AM, além de ajudar a **justificar possíveis decisões** feitas pelo modelo.

Vamos supor que o **atributo na segunda coluna** é um **atributo protegido ou proxy** com valor "b"



Predição

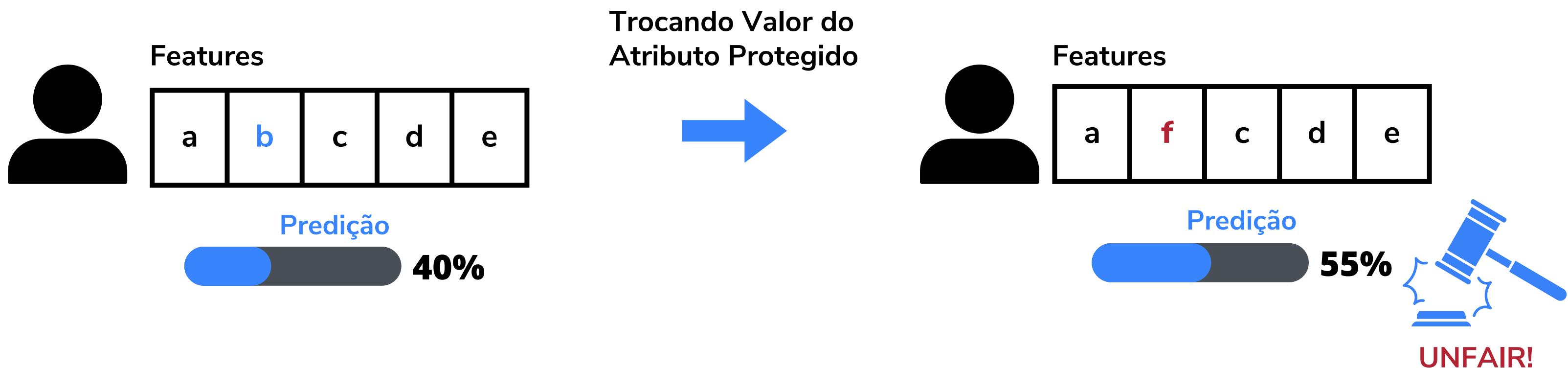
30%

A horizontal progress bar consisting of a blue circle on the left and a grey bar extending to the right, with the number "30%" displayed at the end of the bar.

Ex: Predição foi altamente afetada pela var2, que é um **atributo protegido ou proxy**.

Counterfactual Fairness

Segundo (Kusner, Matt; et al, 2017), em counterfactual fairness a intuição é que **uma decisão é "fair"** se é igual no a) **mundo atual** e b) **mundo contrafáctual**, onde o indivíduo pertenceria a um diferente grupo protegido.

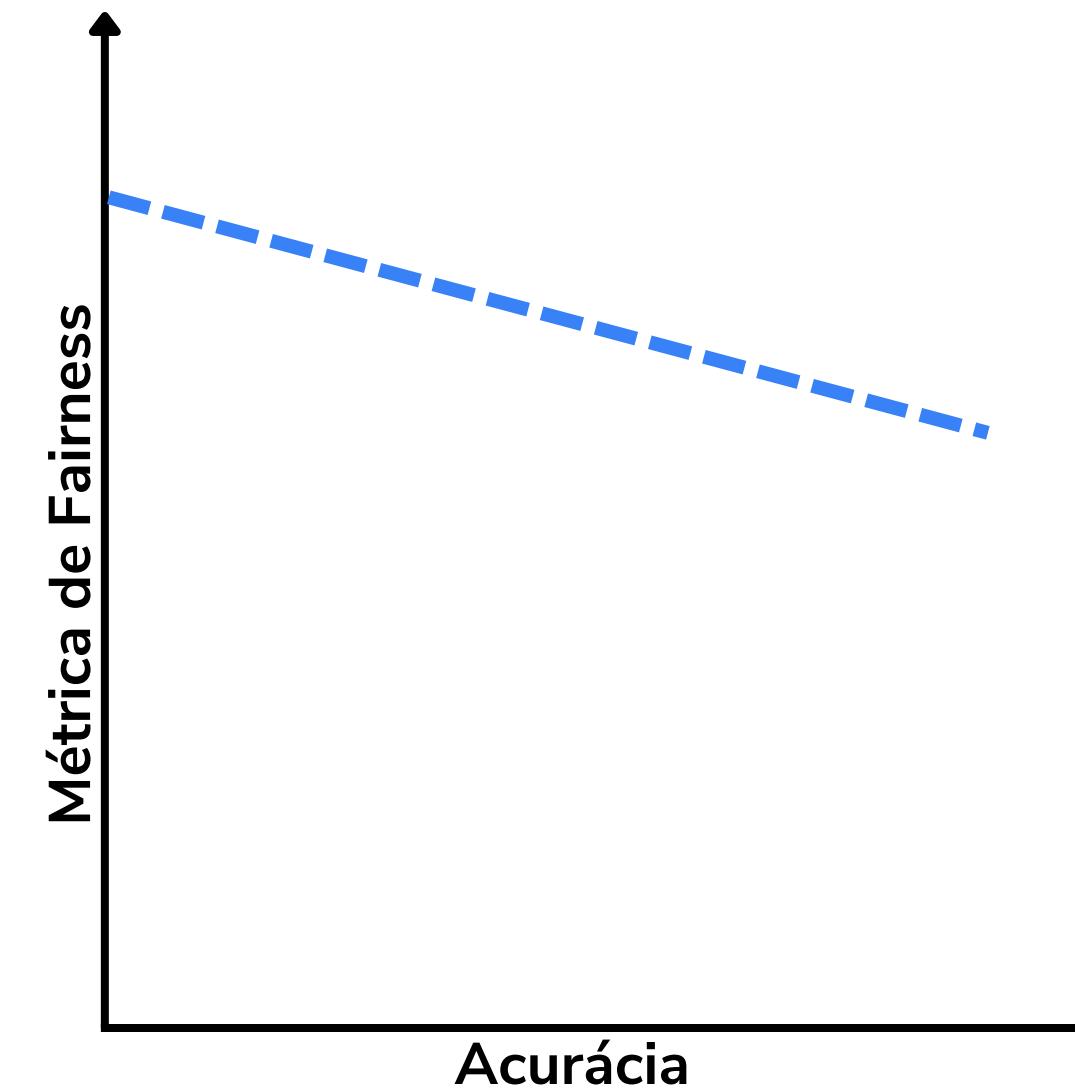
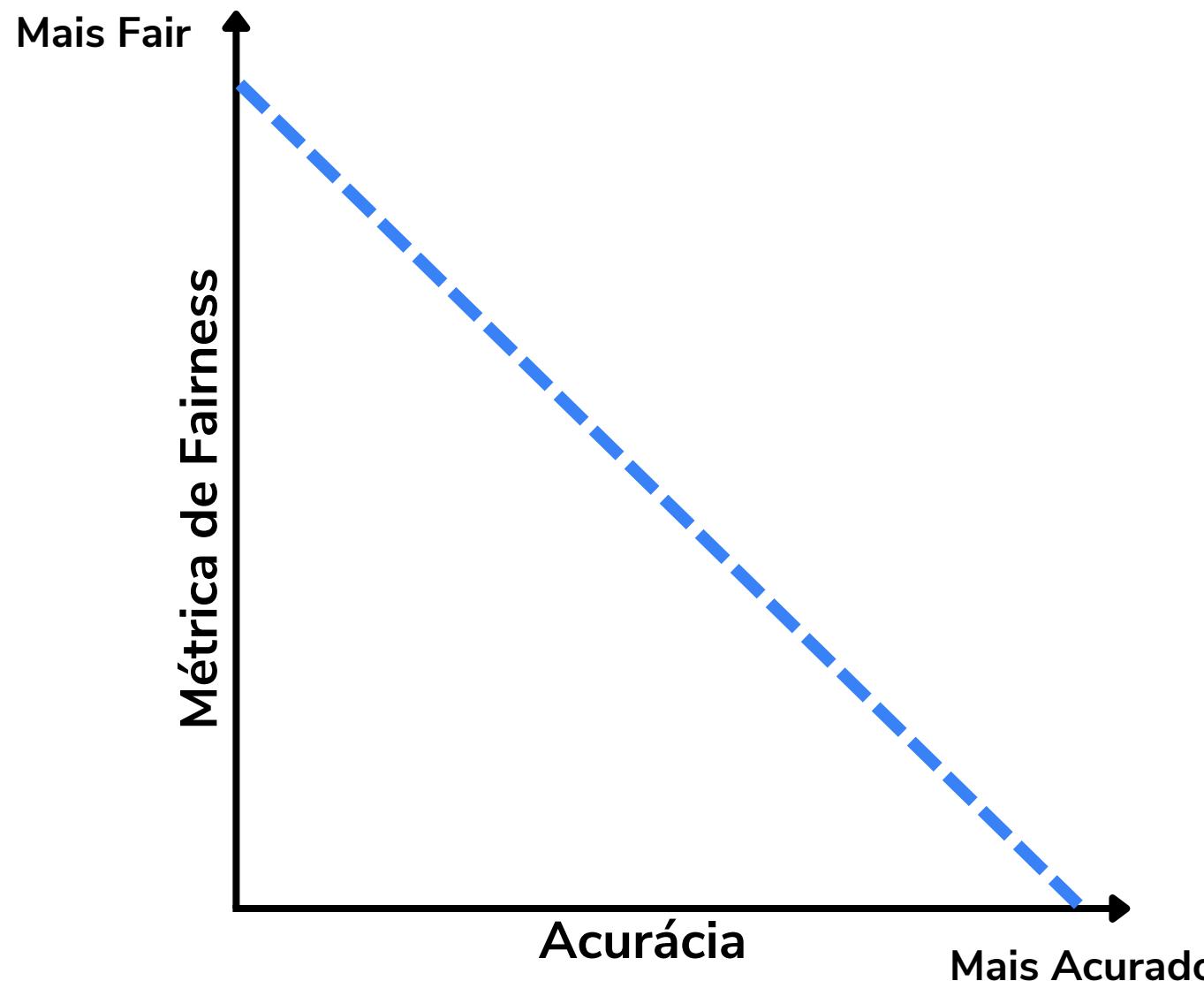


Trade-off Fairness x Acurácia

Em geral, pode-se perder performance de acurácia, ao construir um **modelo "fair"**.

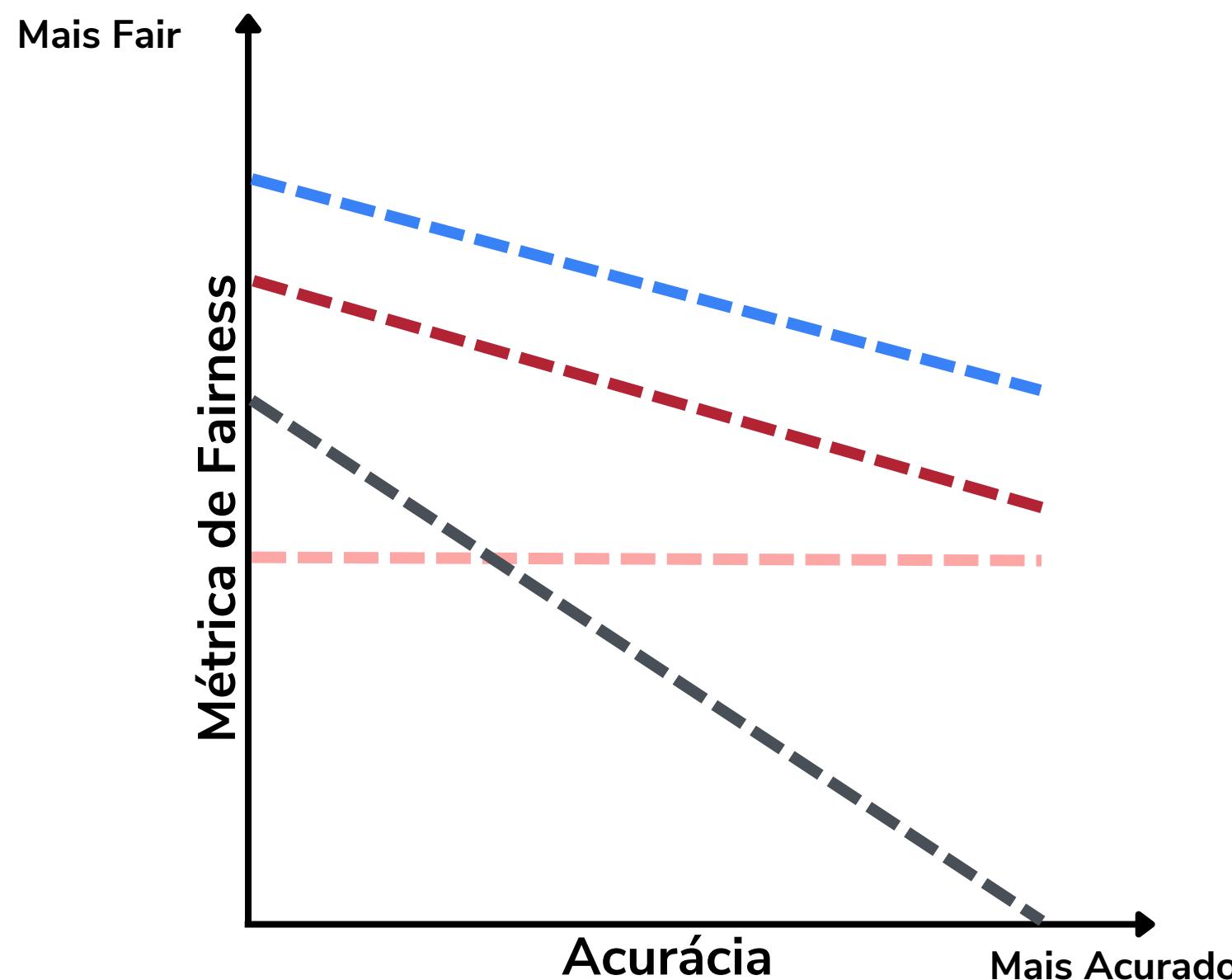
Por isso, o **trade-off Fairness x Acurácia** deve **sempre** ser analisado a fim de encontrar o ponto ótimo.

Exemplos de Possíveis Situações



Modelos Alternativos

Ainda na questão trade-off Fairness x Acurácia. **Diversos modelos podem performar diferente** e assim uma prática comum é **testar diversos modelos** e escolher o que se sair melhor nesse trade-off.



Estimativa do Atributo Protegido

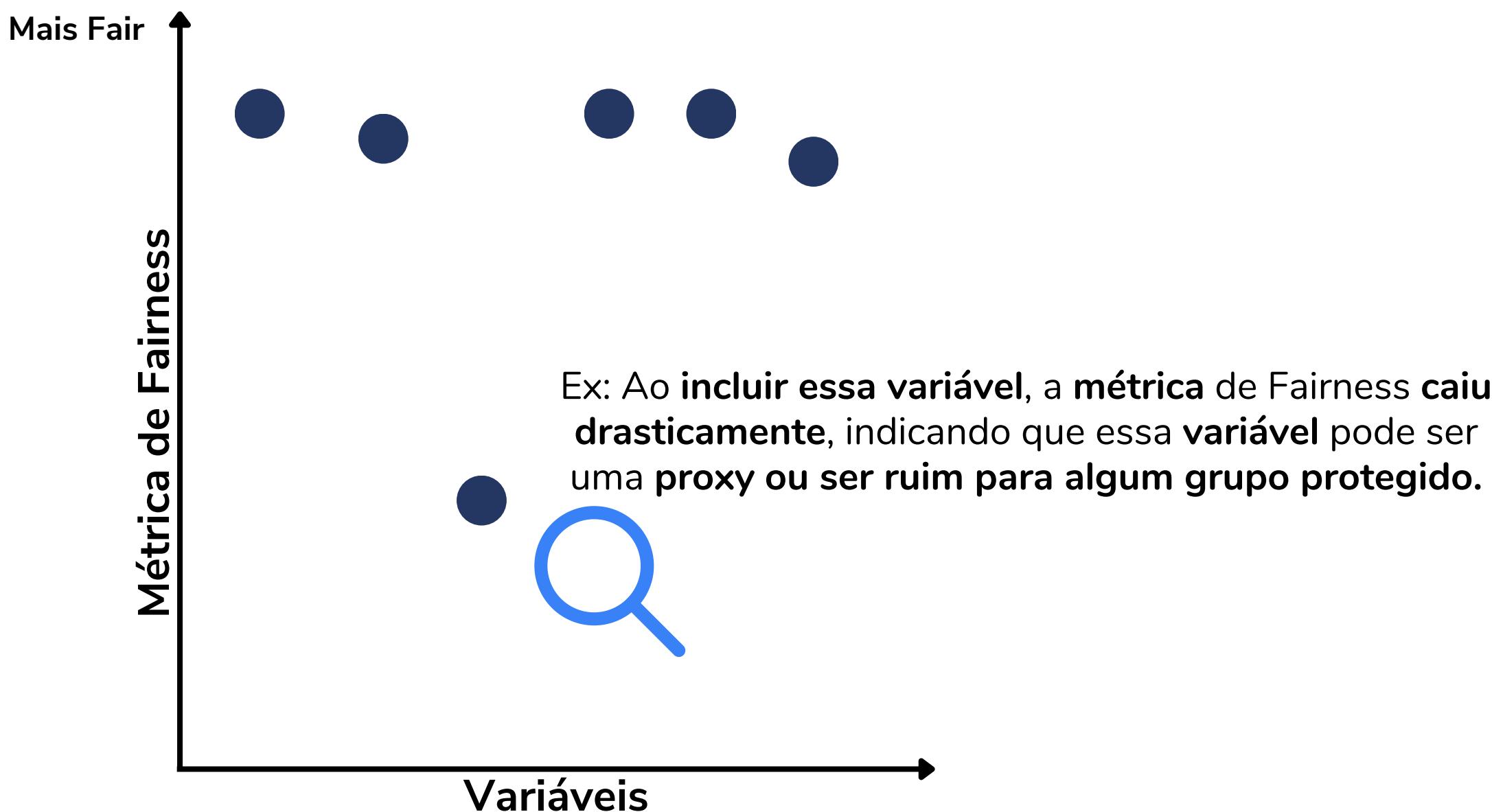
Algumas vezes, o **atributo protegido** pode não estar disponível, já que são, em geral, características pessoais. Então diversos trabalhos da literatura propõem **heurísticas para estimar esses atributos**.

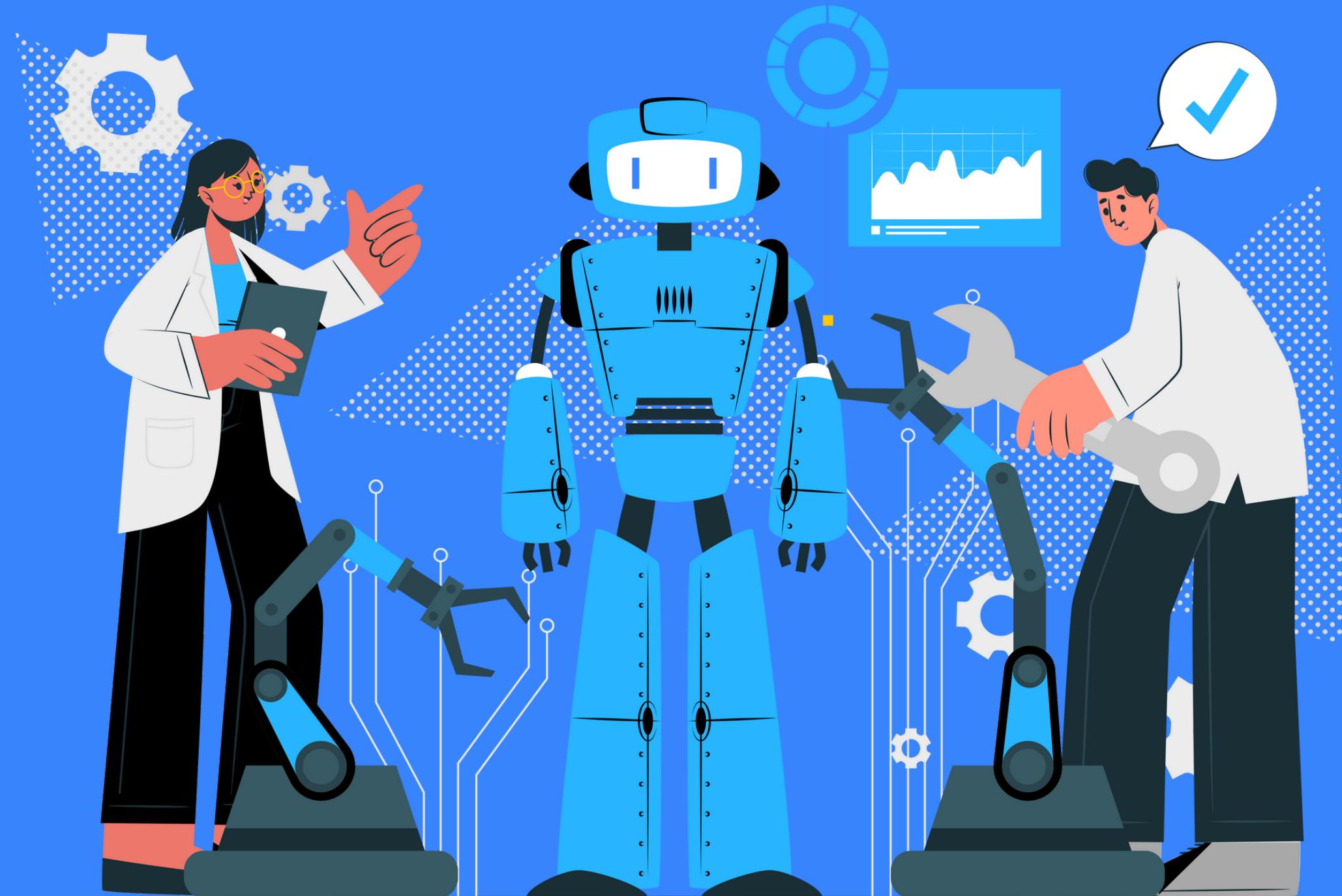
Pode-se utilizar **features proxies** ou estatísticas do censo.



Attribute Contribution

Features proxies do atributo protegido podem estar **contidas no dataset de treino**, e embora não esteja claro para o usuário, elas podem **contribuir com unfairness**. Assim, pode-se **testar a inclusão/exclusão das variáveis** de treino.

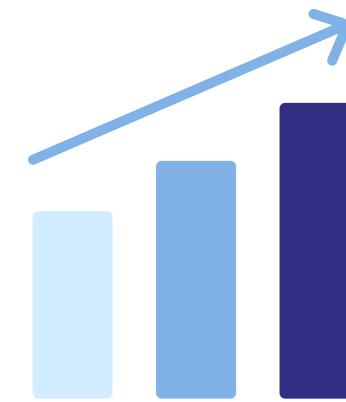
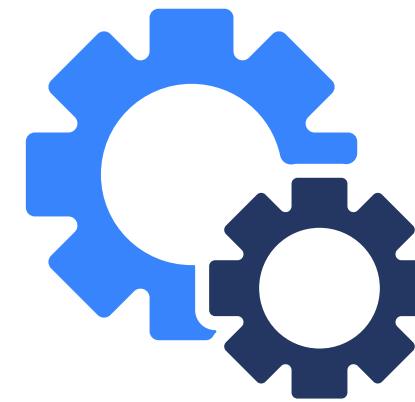




Mitigação

Como mitigar vieses e
obter modelos mais justos?

Tipos de Técnicas para Mitigação



Pré-Processamento

Altera o espaço de entrada de forma a **obter representações "fair"**.

Aplicável a qualquer modelo, desde que seja possível alterar os dados de treinamento.

Em-Processamento

Atua na **fase de treinamento do modelo**, como exemplo: atribuição de pesos, termo de Fairness em uma loss, treinamento adversarial, etc.

Nem todo modelo permite.

Pós-Processamento

Altera as **previsões do modelo**, de alguma maneira, para melhorar alguma **métrica de fairness de interesse**.

Aplicável a qualquer modelo.

Pré-Processamento

Três exemplos de técnicas de pré-processamento para mitigar Fairness são:

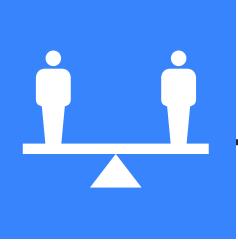


Supressão

Consiste em **remover o atributo protegido da modelagem**.

Nem sempre é efetivo, pois podem existir **features proxy** do atributo protegido.

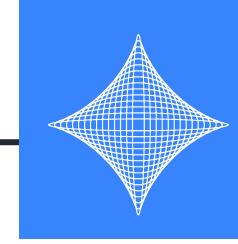
Exemplo: CEP como proxy de raça.



Amostragem

Corrige o viés de um grupo estar recebendo mais o rótulo favorável, **balanceado o dataset em relação aos grupos**.

Útil para garantir saída igual entre grupos, mas **não garante taxa de acerto/erro igual entre grupos**.



Espaço de Entrada

Modifica as **features** de forma a obter uma **representação "fair"** do espaço de entrada.

Ex: Variável **renda** revela um grupo **protegido**, então a variável é mapeada para novos valores.

Algoritmos:

- Disparate Impact Remover
- Learned Fair Representation
- Optimized Pre-Processing

Exemplo: Amostragem

Corrige o viés de um grupo estar recebendo mais o rótulo favorável, balanceando o dataset em relação aos grupos.



Grupo A (Não-Privilegiado)

Rótulo favorável

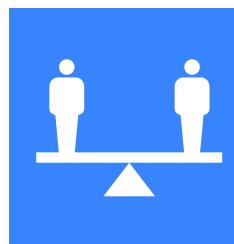
40%



Grupo B (Privilegiado)

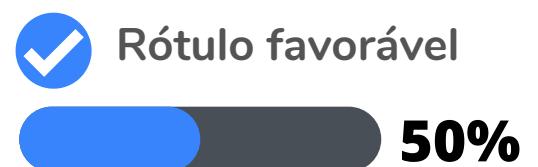
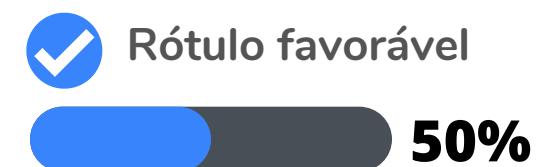
Rótulo favorável

60%



Algoritmo de Amostragem

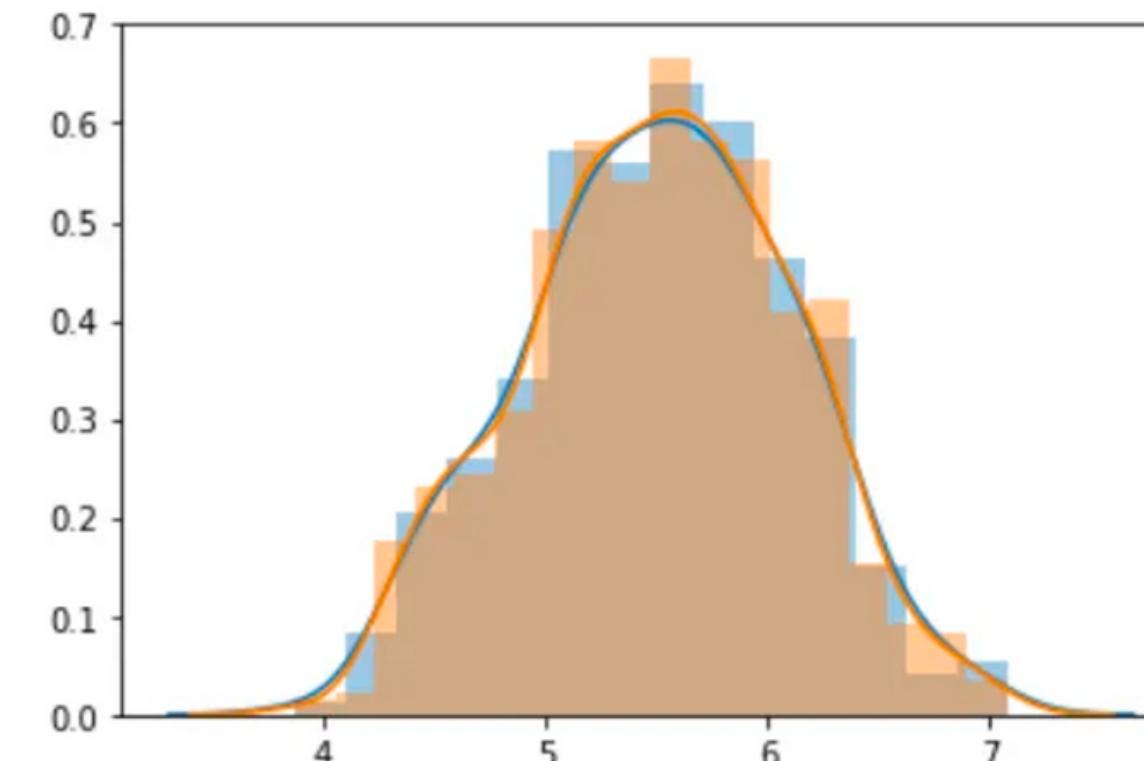
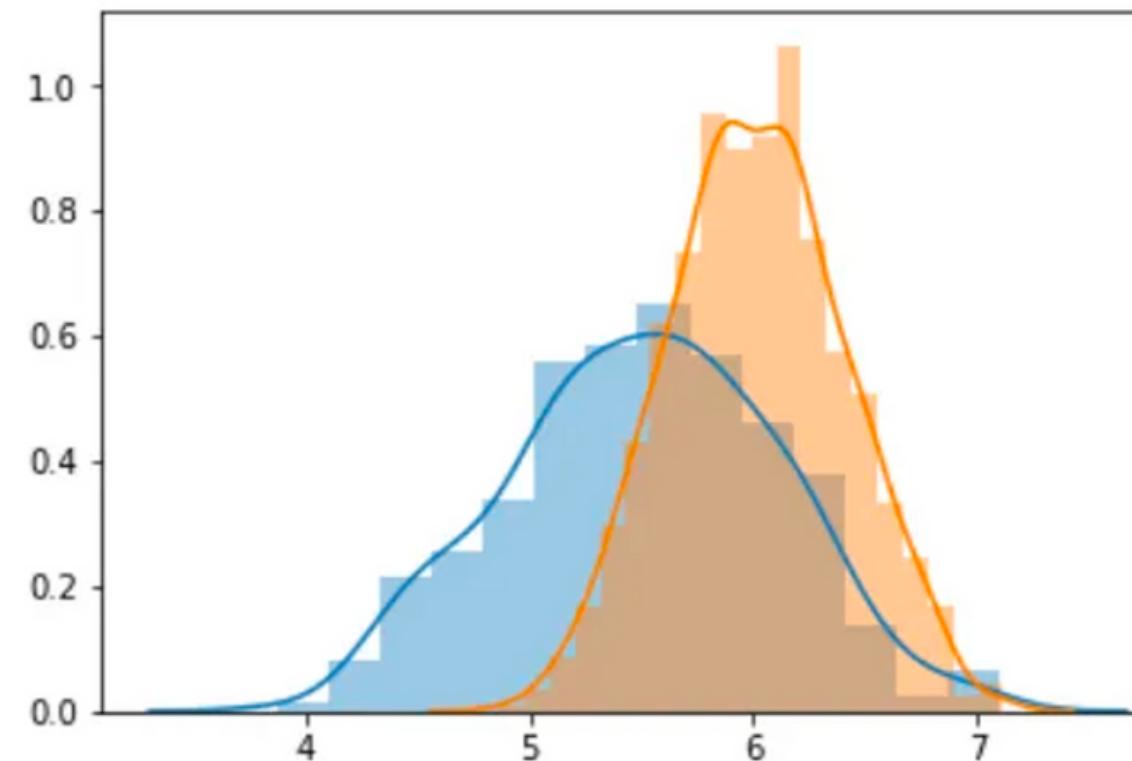
- Undersampling
- Oversampling
- Preferential Sampling



Exemplo: Espaço de Entrada

Disparate Impact Remover (Feldman, M., et al): Ideal onde **features** podem **revelar/ter uma forte relação com o atributo sensível**, "movendo" a distribuição, mas **mantendo a ordenação intra-grupo**. Exemplo:

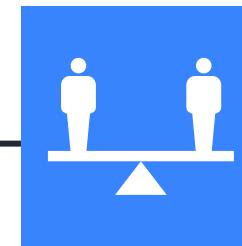
- Problema de aprovação em vestibular. Dois grupos, onde o grupo privilegiado teve acesso a cursos preparatórios para o vestibular, enquanto o grupo não-privilegiado, não. Dessa forma, é esperado que a **nota na prova pré-vestibular para o grupo privilegiado é maior**, e o **algoritmo serviria para "normalizar" a discrepância nos resultados**.



Fonte: <https://towardsdatascience.com/ai-fairness-explanation-of-disparate-impact-remover-ce0da59451f1>

Em-Processamento

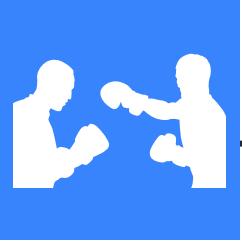
Três exemplos de técnicas de em-processamento para mitigar Fairness são:



Reweighting

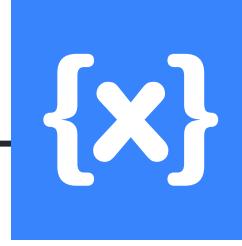
Atribui pesos associados a cada par de grupo + rótulos para ser utilizado na função de custo do treinamento de um modelo.

Modelo deve aceitar atribuição de pesos (como o `sample_weight`).



Adversarial Debiasing

O modelo é treinado junto com um segundo modelo que atua como modelo adversário. Assim, o modelo adversarial irá tentar prever o atributo protegido com base na saídas do primeiro modelo. A cada iteração o modelo adversarial guia o primeiro modelo para modificar seus parâmetros de forma que não seja mais possível prever o atributo protegido com base nas saídas.



Otimização

O problema pode ser modelado como um problema de otimização, com restrições para considerar Fairness.

Ou pode ser construída uma função de Loss, com um termo de Fairness, considerando uma aproximação diferenciável de alguma métrica de Fairness.

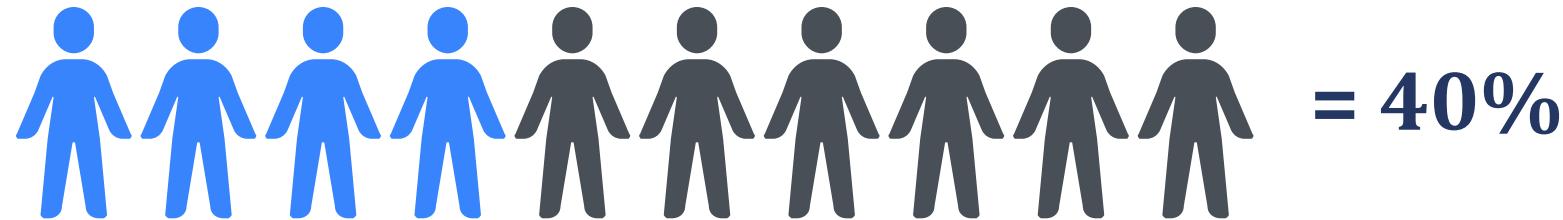
Exemplo: Reweighting

Atribui pesos associados a cada par de grupo + rótulos para ser utilizado na função de custo do treinamento de um modelo. Modelo deve aceitar atribuição de pesos (como o `sample_weight`).

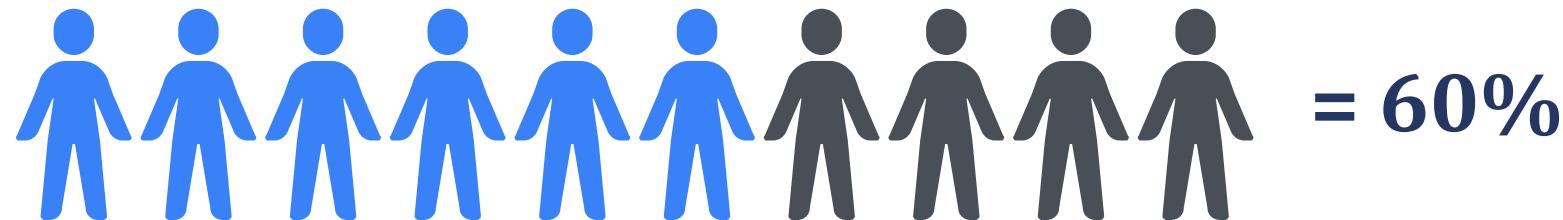
Para cada **Grupo G e Rótulo R**, calcular:

$$W_{GR} = \frac{N_{\text{Grupo}=G} * N_{\text{Rótulo}=R}}{N_{\text{Total}} * N_{\text{Grupo}=G \text{ e Rótulo}=R}}$$

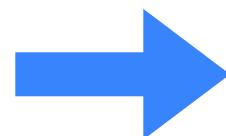
Grupo A (Não-Privilegiado)



Grupo B (Privilegiado)



Cálculo
dos Pesos



Assim, os **pesos encontrados** são:

Exemplo do **Grupo Não-Privilegiado**:

- Com rótulo 1, Peso = 1.25.
- Com rótulo 0, Peso = 0.83

Exemplo do **Grupo Privilegiado**:

- Com rótulo 1, Peso = 0.83.
- Com rótulo 0, Peso = 1.25

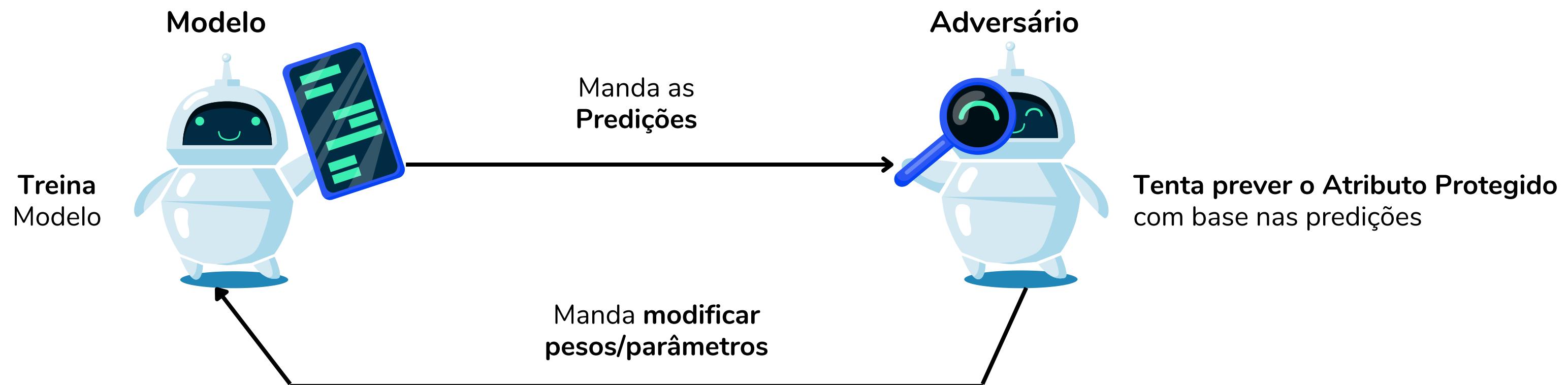
Indivíduo com
Rótulo Favorável (1)

Indivíduo com
Rótulo Não-Favorável (0)

Exemplo: Adversarial Debiasing

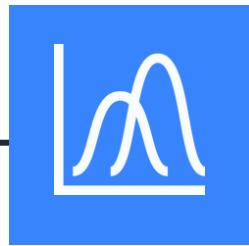
O modelo é treinado junto com um segundo modelo que atua como modelo adversário.

Assim, o modelo adversarial irá tentar prever o atributo protegido com base na saídas do primeiro modelo. A cada iteração o modelo adversarial guia o primeiro modelo para modificar seus parâmetros de forma que não seja mais possível prever o atributo protegido com base nas saídas.



Pós-Processamento

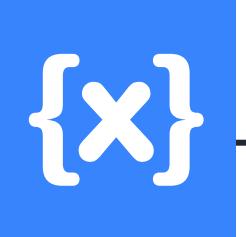
Três exemplos de técnicas de pós-processamento para mitigar Fairness são:



Thresholds

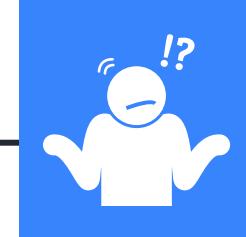
Deve-se encontrar o **melhor threshold de classificação** de acordo com uma determinada métrica de Fairness.

Necessário saída contínua (probabilidade, score).



Otimização

Problema de **otimização** onde o objetivo é **encontrar probabilidades para alterar determinados rótulos** e conseguir melhores resultados de Equalized Odds.



Confiança

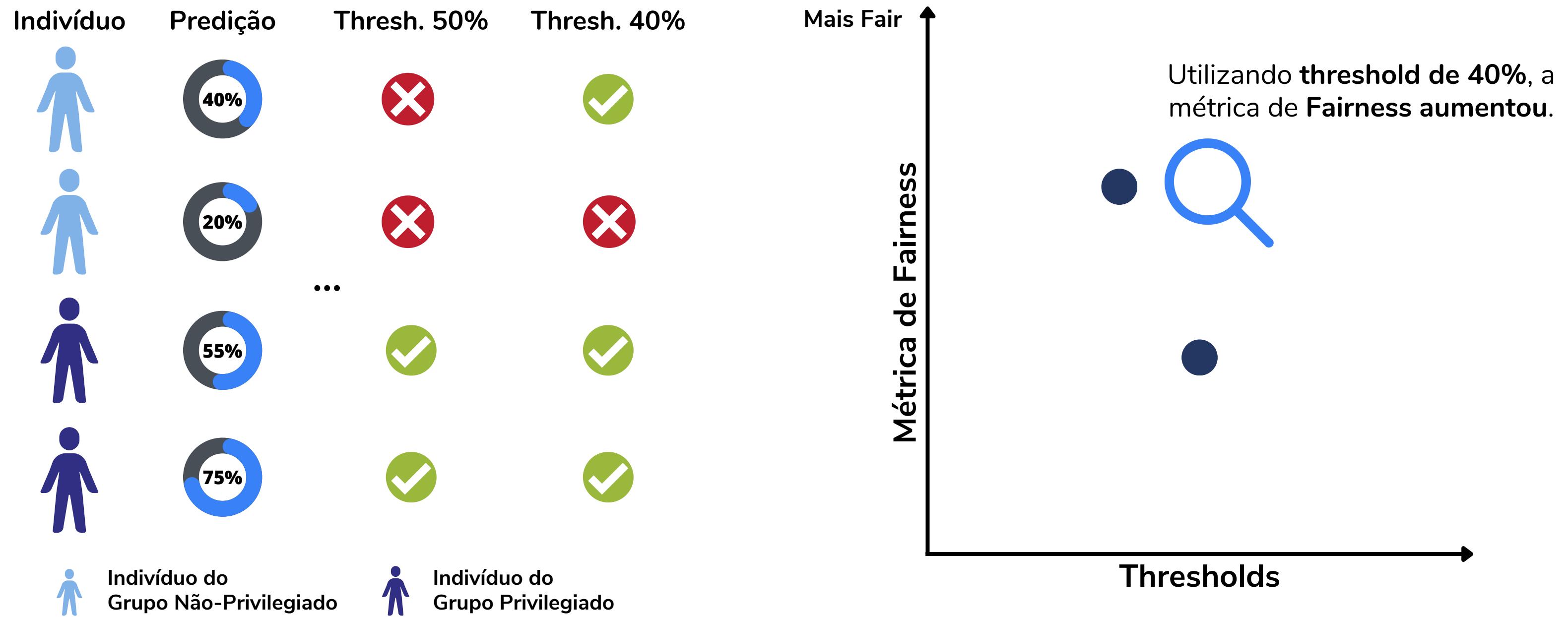
Exemplos com **menor confiança** (próximos a fronteira de decisão) terão suas **previsões alteradas**.

Exemplo:

- Exemplo do **grupo não-privilegiado** recebe **rótulo favorável**;
- Exemplo do **grupo privilegiado** recebe **rótulo não-favorável**.

Exemplo: Thresholds

Deve-se encontrar o melhor threshold de classificação de acordo com uma determinada métrica de Fairness. Necessário saída contínua (probabilidade, score). Pode inclusive existir um threshold diferente para cada grupo.



Exemplo: Confiança

Exemplos com menor confiança (próximos a fronteira de decisão) terão suas previsões alteradas:

- Exemplo do grupo não-privilegiado recebe rótulo favorável;
- Exemplo do grupo privilegiado recebe rótulo não-favorável.

Indivíduo	Predição	Nova Predição	
			Indivíduo do Grupo Não-Privilegiado
			Indivíduo do Grupo Privilegiado
	...		



Implementações

Implementações de bibliotecas para análise
e mitigação de Fairness

AIF360 by IBM

IBM Research Trusted AI

Home Demo Resources Events Videos Community

AI Fairness 360

This extensible open source toolkit can help you examine, report, and mitigate discrimination and bias in machine learning models throughout the AI application lifecycle. We invite you to use and improve it.



[Python API Docs ↗](#) [Get Python Code ↗](#) [Get R Code ↗](#)

Not sure what to do first? Start here!

Read More Learn more about fairness and bias mitigation concepts, terminology, and tools before you begin. →	Try a Web Demo Step through the process of checking and remediating bias in an interactive web demo that shows a sample of capabilities available in this toolkit. →	Watch Videos Watch videos to learn more about AI Fairness 360. →	Read a paper Read a paper describing how we designed AI Fairness 360. →	Use Tutorials Step through a set of in-depth examples that introduces developers to code that checks and mitigates bias in different industry and application domains. →	Ask a Question Join our AIF360 Slack Channel to ask questions, make comments and tell stories about how you use the toolkit. →
View Notebooks Open a directory of Jupyter Notebooks in GitHub that provide working examples of bias detection and mitigation in sample datasets. Then share your own notebooks! →	Contribute You can add new metrics and algorithms in GitHub. Share Jupyter notebooks showing how you have examined and mitigated bias in your machine learning application. →				

AIF360 - Algoritmos para Mitigação

These are ten state-of-the-art bias mitigation algorithms that can address bias throughout AI systems. Add more!

Optimized Pre-processing

Use to mitigate bias in training data. Modifies training data features and labels.



Reweighting

Use to mitigate bias in training data. Modifies the weights of different training examples.



Adversarial Debiasing

Use to mitigate bias in classifiers. Uses adversarial techniques to maximize accuracy and reduce evidence of protected attributes in predictions.



Reject Option Classification

Use to mitigate bias in predictions. Changes predictions from a classifier to make them fairer.



Disparate Impact Remover

Use to mitigate bias in training data. Edits feature values to improve group fairness.



Learning Fair Representations

Use to mitigate bias in training data. Learns fair representations by obfuscating information about protected attributes.



Prejudice Remover

Use to mitigate bias in classifiers. Adds a discrimination-aware regularization term to the learning objective.



Calibrated Equalized Odds Post-processing

Use to mitigate bias in predictions. Optimizes over calibrated classifier score outputs that lead to fair output labels.



Equalized Odds Post-processing

Use to mitigate bias in predictions. Modifies the predicted labels using an optimization scheme to make predictions fairer.



Meta Fair Classifier

Use to mitigate bias in classifier. Meta algorithm that takes the fairness metric as part of the input and returns a classifier optimized for that metric.



AIF360 - Métricas

Are individuals treated similarly? Are privileged and unprivileged groups treated similarly? Find out by using metrics like these that measure individual and group fairness.

Statistical Parity Difference

The difference of the rate of favorable outcomes received by the unprivileged group to the privileged group.



Equal Opportunity Difference

The difference of true positive rates between the unprivileged and the privileged groups.



Average Odds Difference

The average difference of false positive rate (false positives/negatives) and true positive rate (true positives/positives) between unprivileged and privileged groups.



Disparate Impact

The ratio of rate of favorable outcome for the unprivileged group to that of the privileged group.



Theil Index

Measures the inequality in benefit allocation for individuals.



Euclidean Distance

The average Euclidean distance between the samples from the two datasets.



Mahalanobis Distance

The average Mahalanobis distance between the samples from the two datasets.

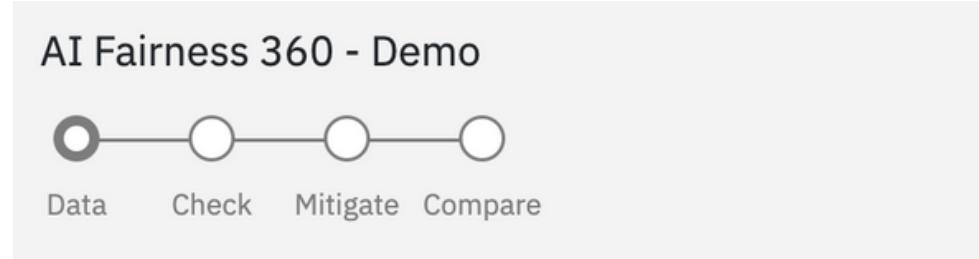


Manhattan Distance

The average Manhattan distance between the samples from the two datasets.



AIF360 - Demo: Dataset e Métricas



1. Choose sample data set

Bias occurs in data used to train a model. We have provided three sample datasets that should be protected to avoid bias.

Compas (ProPublica recidivism)

Predict a criminal defendant's likelihood of reoffending.

Protected Attributes:

- **Sex**, privileged: **Female**, unprivileged: **Male**
- **Race**, privileged: **Caucasian**, unprivileged: **Not Caucasian**

[Learn more](#)

German credit scoring

Predict an individual's credit risk.

Protected Attributes:

- **Sex**, privileged: **Male**, unprivileged: **Female**
- **Age**, privileged: **Old**, unprivileged: **Young**

[Learn more](#)

Adult census income

Predict whether income exceeds \$50K/yr based on census data.

Protected Attributes:

- **Race**, privileged: **White**, unprivileged: **Non-white**
- **Sex**, privileged: **Male**, unprivileged: **Female**

[Learn more](#)

2. Check bias metrics

Dataset: German credit scoring

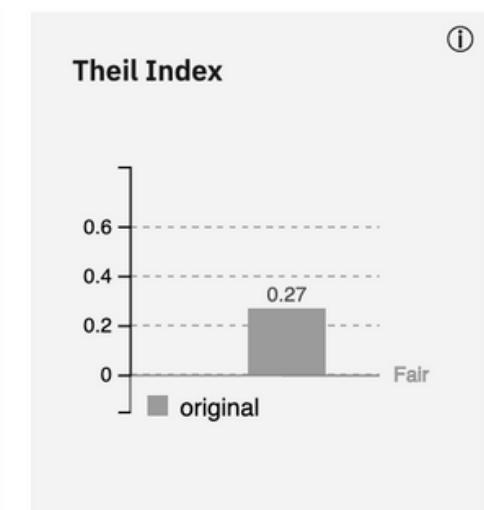
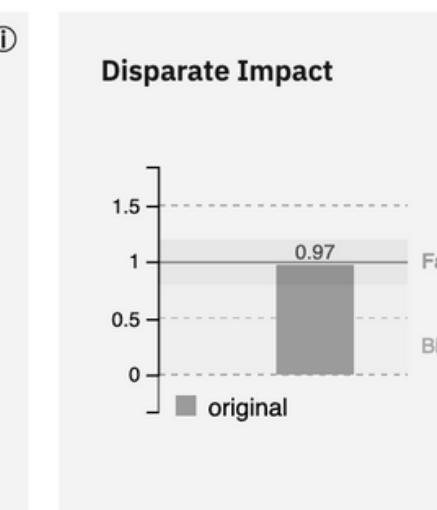
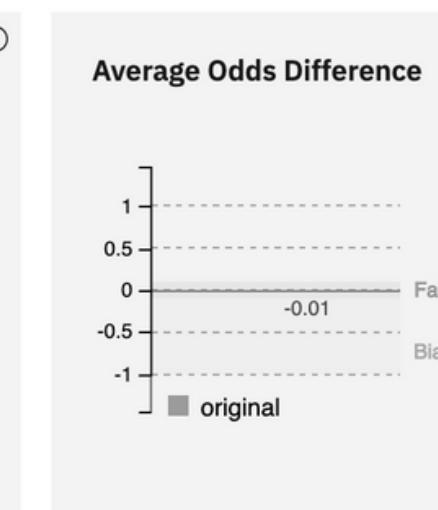
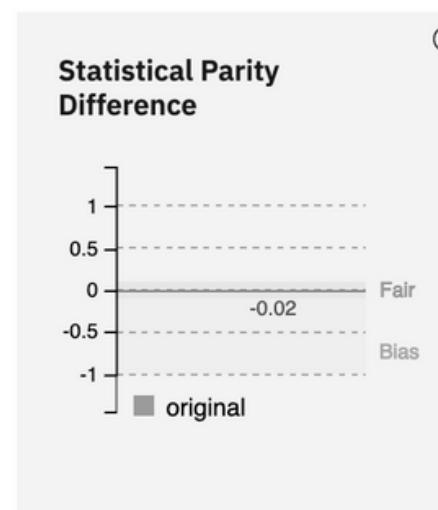
Mitigation: none

Protected Attribute: Sex

Privileged Group: **Male**, Unprivileged Group: **Female**

Accuracy with no mitigation applied is 75%

With default thresholds, bias against unprivileged group detected in 0 out of 5 metrics



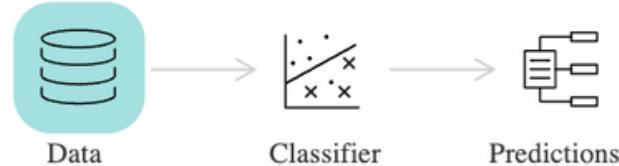
AIF360 - Demo: Escolhendo um Algoritmo

3. Choose bias mitigation algorithm

A variety of algorithms can be used to mitigate bias. The choice of which to use depends on whether you want to fix the data (pre-process), the classifier (in-process), or the predictions (post-process). [Learn more about how to choose.](#)

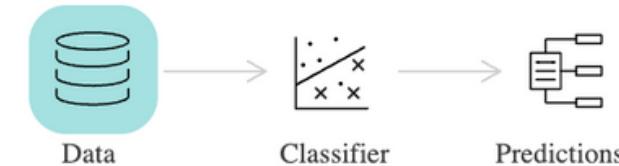
Reweighting

Weights the examples in each (group, label) combination differently to ensure fairness before classification.



Optimized Pre-Processing

Learns a probabilistic transformation that can modify the features and the labels in the training data.



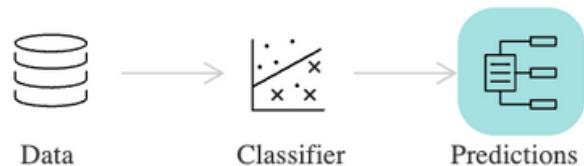
Adversarial Debiasing

Learns a classifier that maximizes prediction accuracy and simultaneously reduces an adversary's ability to determine the protected attribute from the predictions. This approach leads to a fair classifier as the predictions cannot carry any group discrimination information that the adversary can exploit.



Reject Option Based Classification

Changes predictions from a classifier to make them fairer. Provides favorable outcomes to unprivileged groups and unfavorable outcomes to privileged groups in a confidence band around the decision boundary with the highest uncertainty.



AIF360 - Demo: Analisando Resultados

4. Compare original vs. mitigated results

Dataset: German credit scoring

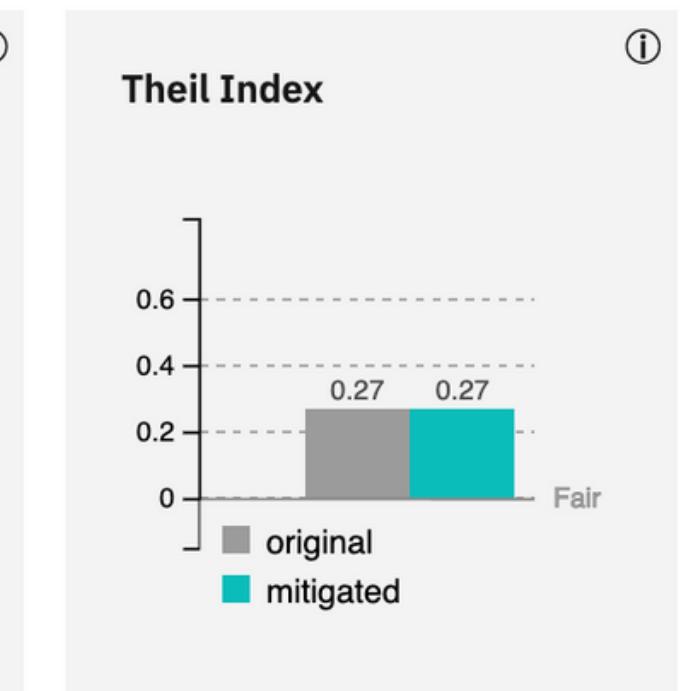
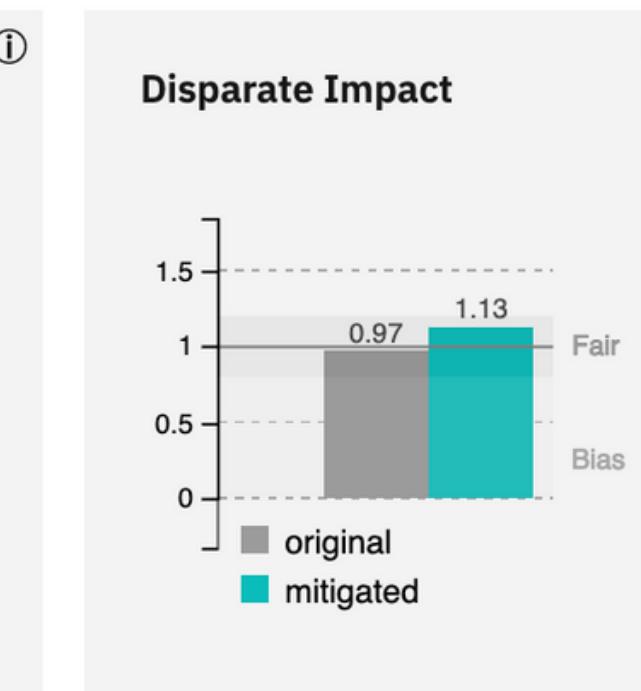
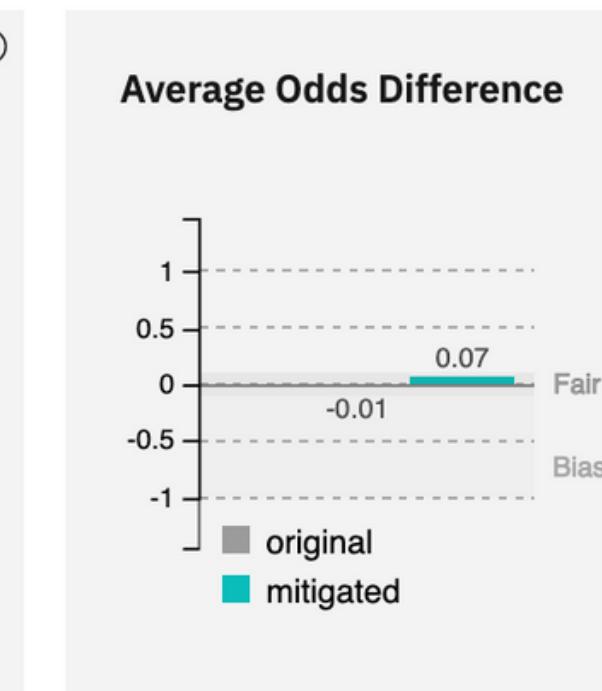
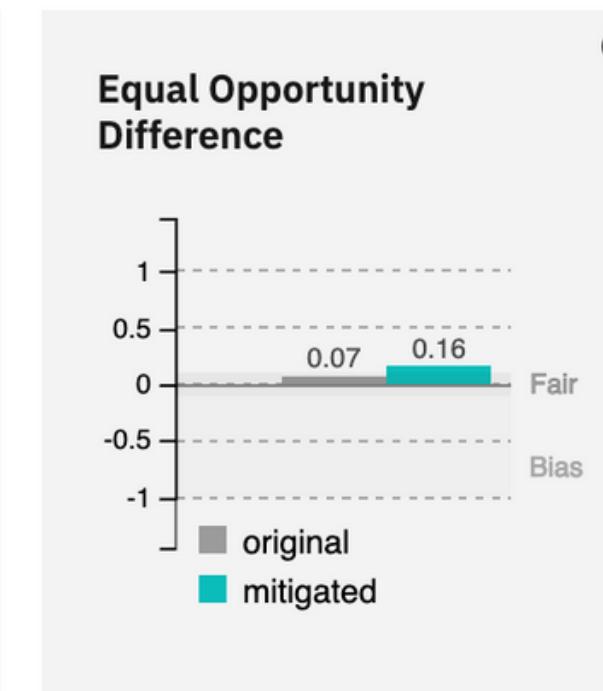
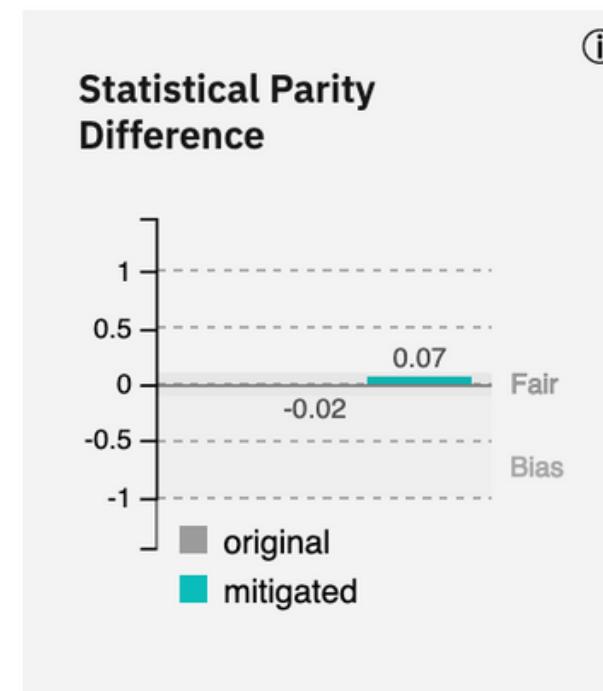
Mitigation: [Reweighting algorithm applied](#)

Protected Attribute: Sex

Privileged Group: **Male**, Unprivileged Group: **Female**

Accuracy after mitigation changed from 75% to 78%

Bias against unprivileged group unchanged after mitigation (0 of 5 metrics indicate bias)



FairLearn by Microsoft

Fairlearn

Get Started User Guide API Docs Example Notebooks Cont

Improve fairness of AI systems

Fairlearn is an open-source, community-driven project to help data scientists improve fairness of AI systems.

Learn about AI fairness from our guides and use cases. Assess and mitigate fairness issues using our Python toolkit. Join our community and contribute metrics, algorithms, and other resources.

Get Started

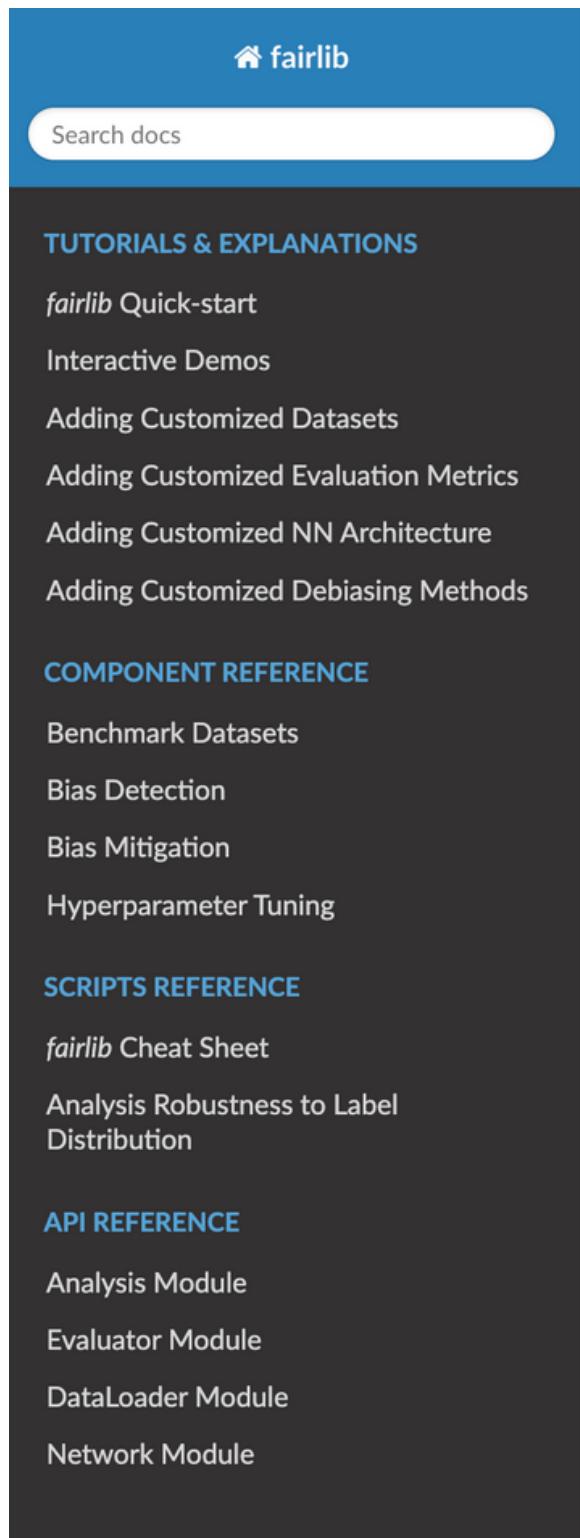
Fairness in Machine Learning

Fairness of AI systems

AI systems can behave unfairly for a variety of reasons. Sometimes it is because of societal biases reflected in the training data and in the decisions made during the development and deployment of these systems. In other cases, AI systems behave unfairly not because of societal biases, but because of characteristics of the data (e.g., too few data points about some group of people) or characteristics of the systems themselves. It can be hard to distinguish between these reasons, especially since they are not mutually exclusive and often exacerbate one another. Therefore, we define whether an AI system is behaving unfairly in terms of its impact on people — i.e., in terms of harms — and not in terms of specific causes, such as societal biases, or in terms of intent, such as prejudice.

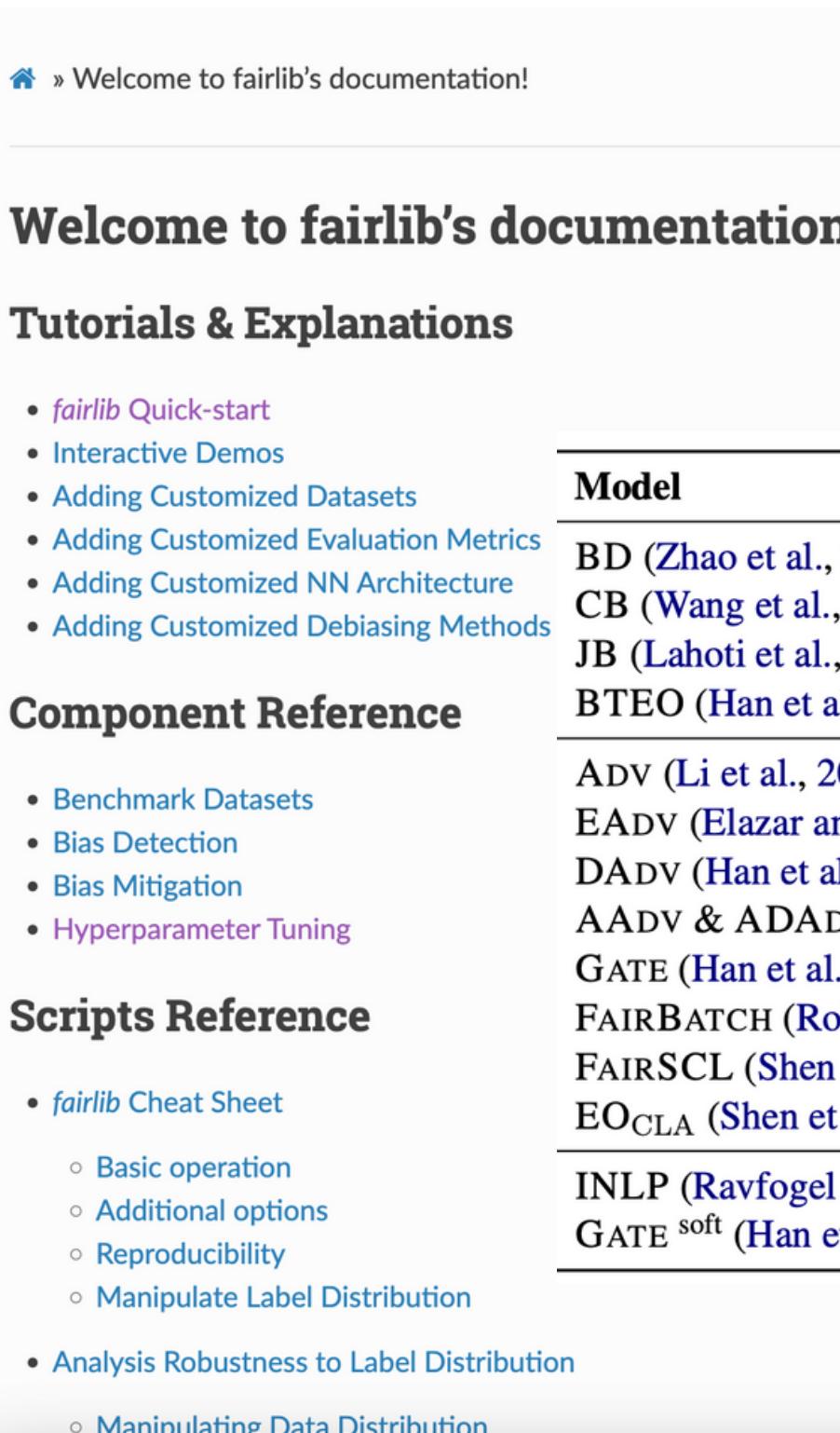
algorithm	description	
ExponentiatedGradient	A wrapper (reduction) approach to fair classification described in <i>A Reductions Approach to Fair Classification</i> [1].	CorrelationRemover
GridSearch	A wrapper (reduction) approach described in Section 3.4 of <i>A Reductions Approach to Fair Classification</i> [1]. For regression it acts as a grid-search variant of the algorithm described in Section 5 of <i>Fair Regression: Quantitative Definitions and Reduction-based Algorithms</i> [2].	AdversarialFairnessClassifier
ThresholdOptimizer	Postprocessing algorithm based on the paper <i>Equality of Opportunity in Supervised Learning</i> [3]. This technique takes as input an existing classifier and the sensitive feature, and derives a monotone transformation of the classifier's prediction to enforce the specified parity constraints.	AdversarialFairnessRegressor
		Preprocessing algorithm that removes correlation between sensitive features and non-sensitive features through linear transformations.
		An optimization algorithm based on the paper <i>Mitigating Unwanted Biases with Adversarial Learning</i> [4]. This method trains a neural network classifier that minimizes training error while preventing an adversarial network from inferring sensitive features. The neural networks can be defined either as a PyTorch module or TensorFlow2 model .
		The regressor variant of the above AdversarialFairnessClassifier . Useful to train a neural network with continuous valued output(s).

FairLib



The sidebar contains a search bar and several sections:

- TUTORIALS & EXPLANATIONS**
 - [fairlib Quick-start](#)
 - [Interactive Demos](#)
 - [Adding Customized Datasets](#)
 - [Adding Customized Evaluation Metrics](#)
 - [Adding Customized NN Architecture](#)
 - [Adding Customized Debiasing Methods](#)
- COMPONENT REFERENCE**
 - [Benchmark Datasets](#)
 - [Bias Detection](#)
 - [Bias Mitigation](#)
 - [Hyperparameter Tuning](#)
- SCRIPTS REFERENCE**
 - [fairlib Cheat Sheet](#)
 - [Analysis Robustness to Label Distribution](#)
- API REFERENCE**
 - [Analysis Module](#)
 - [Evaluator Module](#)
 - [DataLoader Module](#)
 - [Network Module](#)



The homepage features a navigation bar with a search bar and a main content area:

Welcome to fairlib's documentation!

Tutorials & Explanations

- [fairlib Quick-start](#)
- [Interactive Demos](#)
- [Adding Customized Datasets](#)
- [Adding Customized Evaluation Metrics](#)
- [Adding Customized NN Architecture](#)
- [Adding Customized Debiasing Methods](#)

Component Reference

- [Benchmark Datasets](#)
- [Bias Detection](#)
- [Bias Mitigation](#)
- [Hyperparameter Tuning](#)

Scripts Reference

- [fairlib Cheat Sheet](#)
 - [Basic operation](#)
 - [Additional options](#)
 - [Reproducibility](#)
 - [Manipulate Label Distribution](#)
- [Analysis Robustness to Label Distribution](#)
 - [Manipulating Data Distribution](#)

Model	Main Idea
BD (Zhao et al., 2017)	Equalize the size of protected groups.
CB (Wang et al., 2019)	Down-sample the majority protected group within each class.
JB (Lahoti et al., 2020)	Jointly balance the Protected attributes and classes.
BTEO (Han et al., 2021a)	Balance protected attributes within advantage classes.
ADV (Li et al., 2018)	Prevent protected attributes from being identified by the discriminator.
EADV (Elazar and Goldberg, 2018)	Employ multiple discriminators for adversarial training.
DADV (Han et al., 2021c)	Employ multiple discriminators with orthogonality regularization for adversarial training.
AADV & ADADV (Han et al., 2022)	Enable discriminators to use target labels as inputs during training.
GATE (Han et al., 2021a)	Address protected factors with an augmented representation.
FAIRBATCH (Roh et al., 2021)	Minimize CE loss gap through minibatch resampling.
FAIRSCL (Shen et al., 2021)	Adopt supervised contrastive learning for bias mitigation.
EO CLA (Shen et al., 2022)	Minimize the CE loss gap within each target label by adjusting the loss.
INLP (Ravfogel et al., 2020)	Remove protected attributes through iterative null-space projection.
GATE ^{soft} (Han et al., 2021a)	Adjust the prior for each group-specific component in GATE (Han et al., 2021a).

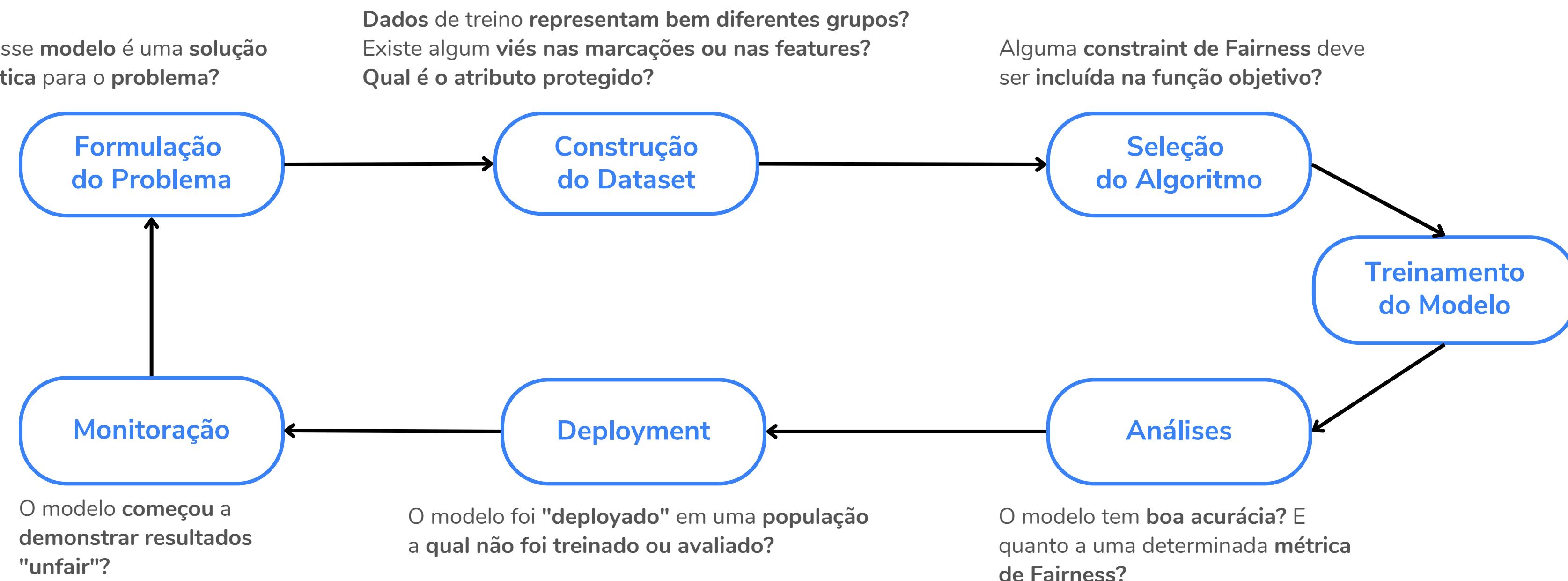


Considerações Finais

Então, como considerar Fairness no desenvolvimento de modelos?

Considerações Finais

Então, como considerar Fairness no processo de construção de um modelo de Aprendizado de Máquina?



Baseado em:





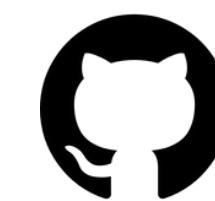
Obrigado!



wdihanster@gmail.com



linkedin.com/in/dihanster



github.com/dihanster