

CS 260: Foundations of Data Science

Prof. Sara Mathieson

Spring 2025



Haverford
COLLEGE

Admin

- **Lab 4** was due last night
- **Lab 5** posted soon
 - Due Wednesday after fall break
 - Naïve Bayes
 - Longer lab
 - Likely: partners optional (can repeat)

Outline

- Naïve Bayes implementation
- Handout 10
- Intro to Algorithmic Bias
- Disparate Impact
- Feedback forms

Outline

- Naïve Bayes implementation
- Handout 10
- Intro to Algorithmic Bias
- Disparate Impact
- Feedback forms

Naive Bayes implementation

likelihood

$$p(\vec{x} | y=k) \approx \prod_{j=1}^P p(x_j | y=k)$$

NB assumption

underflow

$$= \frac{1}{1000} \cdot \frac{1}{10} \cdot \frac{1}{50} \dots \approx 0$$

computer

Solution: compute
in log space!

Prior

$$\log \theta_k = \log \frac{N_k + 1}{n + K}$$

$$\log\left(\frac{a}{b}\right) =$$

$$\log a - \log b$$

$$= \log(N_k + 1) - \log(n + K)$$

likelihood

$$\log \theta_{k,j,v} = \log(N_{k,j,v} + 1)$$

$$- \log(N_{k,j,v} + 1)$$

possible values

$$\log \left(\underbrace{p(y=k) \prod_{j=1}^P p(x_j=v | y=k)}_{\text{product of many small probs.}} \right) = \log p(y=k) + \sum_{j=1}^P \log p(x_j=v | y=k)$$

python import math

(natural log) math.log(x)

~~prior = [0.2, 0.5, 0.3]~~ ^{not in code}
(K=3)

log-prior = [log 0.2, log 0.5, log 0.3]

Outline

- Naïve Bayes implementation
- Handout 10
- Intro to Algorithmic Bias
- Disparate Impact
- Feedback forms

Handout 10

$$\vec{x} = [\text{neg}, \text{pos}]$$

$$\begin{aligned} \textcircled{1} \quad p(y=1|\vec{x}) &\approx p(y=1) p(f_1=\text{neg}|y=1) p(f_2=\text{pos}|y=1) \\ &\approx \frac{4}{7} \cdot \frac{3}{5} \cdot \frac{1}{5} = \frac{4}{75} \end{aligned}$$

$$\begin{aligned} p(y=2|\vec{x}) &\approx p(y=2) p(f_1=\text{neg}|y=2) p(f_2=\text{pos}|y=2) \\ &\approx \frac{5}{9} \cdot \frac{1}{3} \cdot \frac{1}{2} = \frac{5}{54} \end{aligned}$$

$$\frac{p(\vec{x})}{p(\vec{x}) + \underbrace{\left[\frac{4}{75}, \frac{5}{54} \right]}_{\text{posterior}}}$$
$$\text{argmax} \left(\left[\frac{4}{75}, \frac{5}{54} \right] \right)$$

$$\star \boxed{\hat{y}=2} \star (\text{disease})$$

\Rightarrow normalize

$$\underbrace{[0.37, 0.63]}_{\text{posterior}}$$

Data Structure idea

(tennis example)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_3	Overcast	Hot	High	Weak	Yes
x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_7	Overcast	Cool	Normal	Strong	Yes
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{10}	Rain	Mild	Normal	Weak	Yes
x_{11}	Sunny	Mild	Normal	Strong	Yes
x_{12}	Overcast	Mild	High	Strong	Yes
x_{13}	Overcast	Hot	Normal	Weak	Yes
x_{14}	Rain	Mild	High	Strong	No

Data Structure idea

(tennis example)

Condition on $y=\text{No}$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_3	Overcast	Hot	High	Weak	Yes
x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_7	Overcast	Cool	Normal	Strong	Yes
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{10}	Rain	Mild	Normal	Weak	Yes
x_{11}	Sunny	Mild	Normal	Strong	Yes
x_{12}	Overcast	Mild	High	Strong	Yes
x_{13}	Overcast	Hot	Normal	Weak	Yes
x_{14}	Rain	Mild	High	Strong	No

Data Structure idea

(tennis example)

y=No (0)

outlook	Sunny:	Overcast:	Rain:
temperature	Cool:	Mild:	Hot:
humidity	Normal:	High:	
wind	Weak:	Strong:	

y=Yes (1)

outlook	Sunny:	Overcast:	Rain:
temperature	Cool:	Mild:	Hot:
humidity	Normal:	High:	
wind	Weak:	Strong:	

$Y = \text{No } (0)$

outlook	Sunny: $\frac{3+1}{5+3}$	overcast: $\frac{0+1}{5+3}$	rain: $\frac{2+1}{5+3}$
:	$\frac{1}{5+3}$	$\frac{1}{5+3}$	$\frac{1}{5+3}$
temp	:	:	:
humidity	$\frac{1}{5+2}$	$\frac{1}{5+2}$:
wind	$\frac{1}{5+2}$	$\frac{1}{5+2}$:

Laplace counts

$\log_likelihood[0][\text{"outlook"}][\text{"rain"}]$
 \uparrow list

key is feature name
 value: dictionary

$\{ \text{"outlook"} : \{ \text{"sun"} : \frac{1}{2}, \text{"overcast"} : \frac{1}{8}, \text{"rain"} : \frac{3}{8} \} , \text{"temp"} : \dots \}$

keys are feature values
 values: likelihood

$\theta_{k,j|v_i}$

Outline

- Naïve Bayes implementation
- Handout 10
- **Intro to Algorithmic Bias**
- Disparate Impact
- Feedback forms

What does it mean to claim that algorithms are biased (or racist or political...)?

```
3 model = initialization(...)
4 n_epochs = ...
5 train_data = ...
6 for i in n_epochs:
7     train_data = shuffle(train_data)
8     X, y = split(train_data)
9     predictions = predict(X, model)
    error = calculate_error(y, predictions)
    model = update_model(model, error)
```

Pseudocode from [A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size](#)

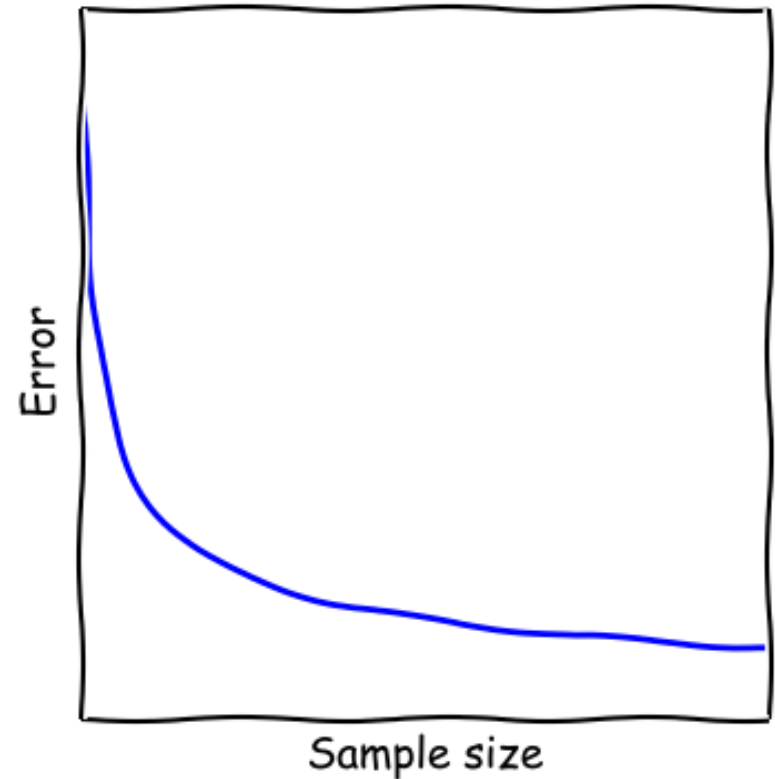
Are algorithms fair by default?

“After all, as the former CPD [Chicago Police Department] computer experts point out, the algorithms in themselves are neutral. ‘This program had absolutely nothing to do with race... but multi-variable equations,’ argues Goldstein. Meanwhile, the potential benefits of predictive policing are profound.”

-Gilian Tett

Sample size disparity

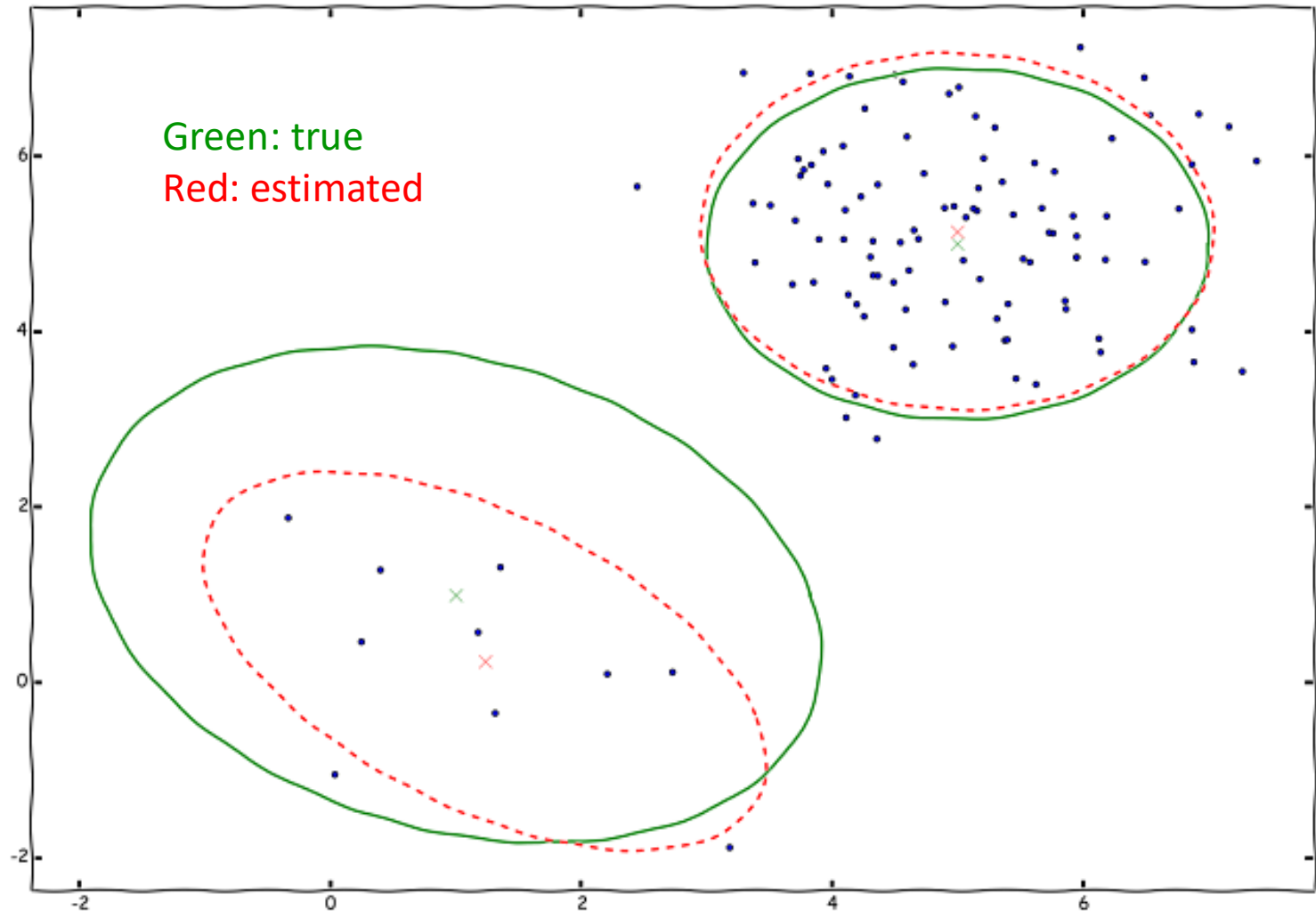
- More data from majority will make results more accurate for that group
- Less accurate for the minority



“The error of a classifier often decreases as the inverse square root of the sample size. Four times as many samples means halving the error rate.”

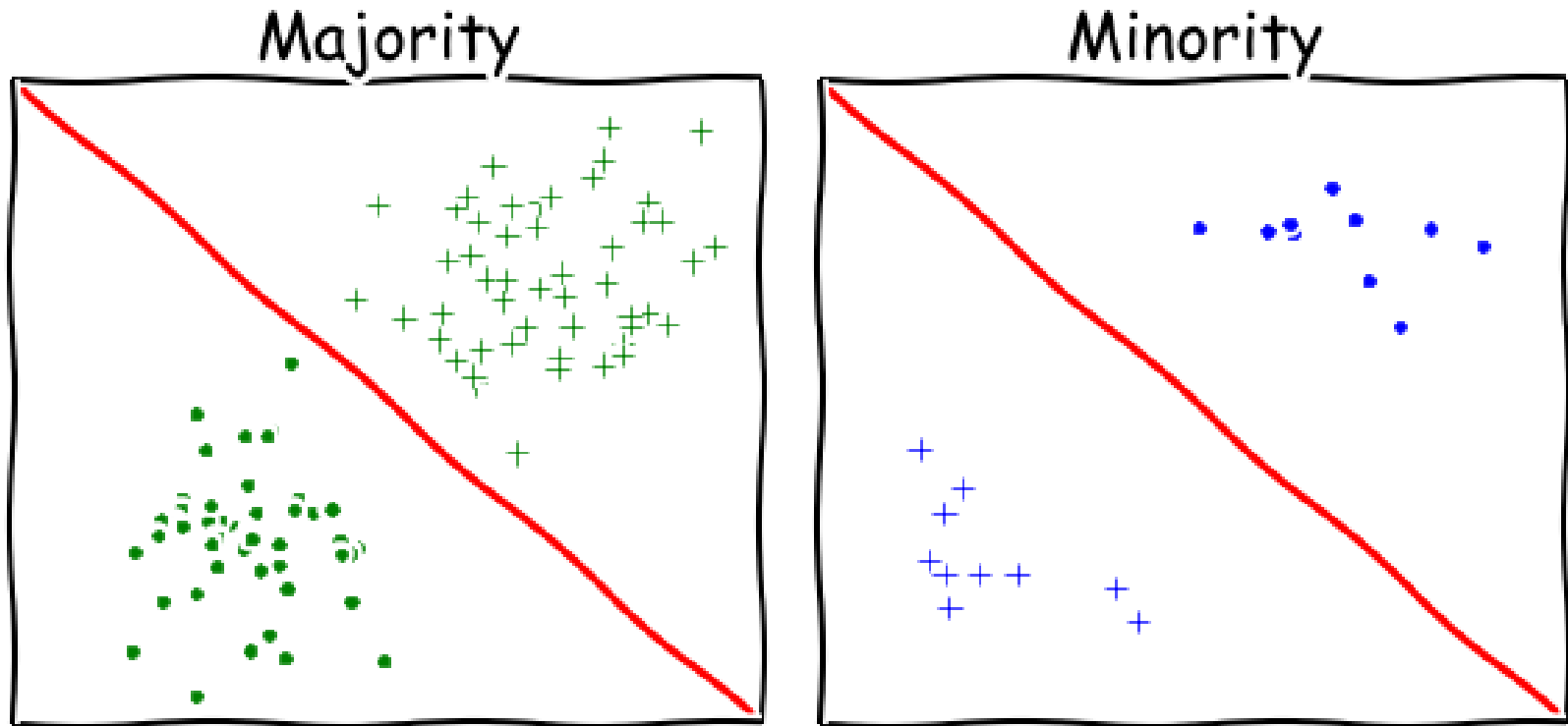
Image: Moritz Hardt

Sample size disparity



“Modeling a heterogeneous population as a gaussian mixture and learning its parameters using the EM algorithm. As expected, the estimates for the smaller group are significantly worse than for the larger. Dashed red ellipsoids describe the estimated covariance matrices. Solid green defines the correct covariance matrices. The green and red crosses indicate correct and estimated means, respectively.” Image: Moritz Hardt

Cultural Differences



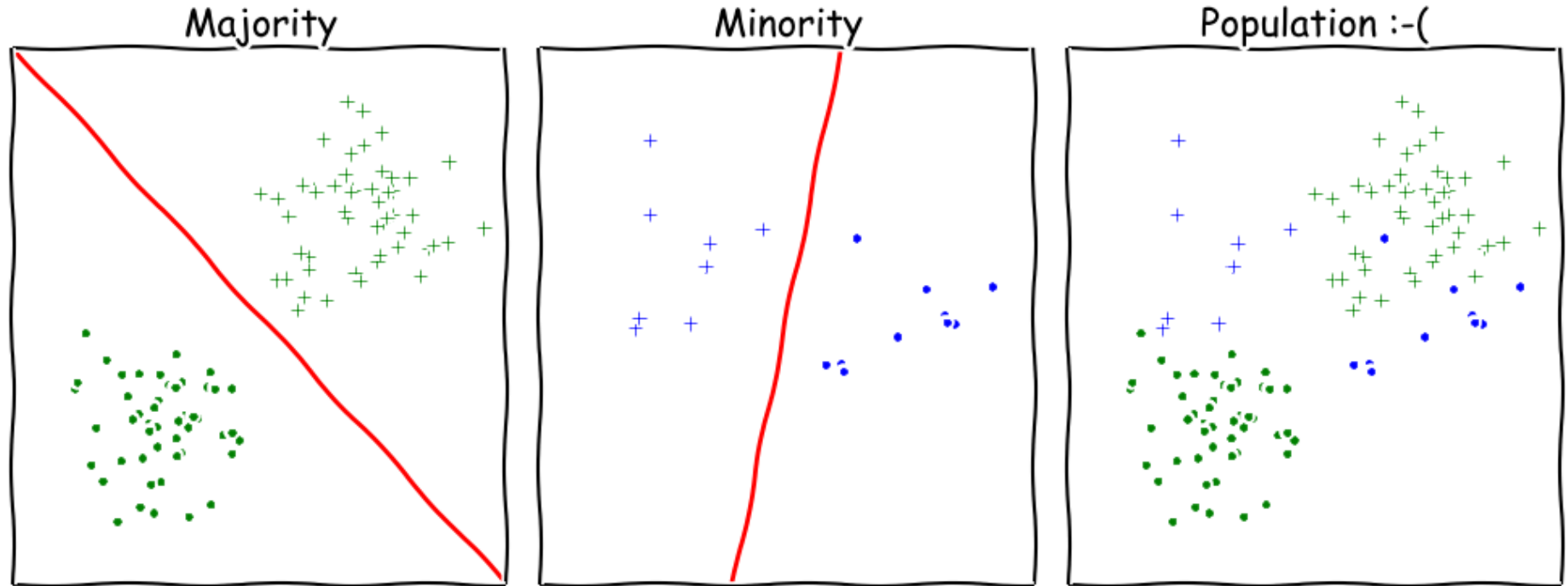
“Positively labeled examples are on opposite sides of the classifier for the two groups.” Image: Moritz Hardt

Goal: determine if a user profile (on Facebook, Twitter, etc) is genuine

- positive: real profile
- negative: fake profile

Feature: length of name

Undesired Complexity



“Even if two groups of the population admit simple classifiers, the whole population may not.”

Image: Moritz Hardt




“How big data is unfair” (takeaways)





- ML is not fair by default, even though it relies on “neutral” multi-variable equations
- If training data reflects social biases, algorithm will likely incorporate them
- “Protected” attributes (race, gender, religion, sexual orientation, etc) often redundantly encoded

Example: machine translation

Turkish - detected ▾



English ▾



o bir aşçı

o bir mühendis

o bir doktor

o bir hemşire

o bir temizlikçi

o bir polis

o bir asker

o bir öğretmen

Example: machine translation

Turkish - detected ▼	English ▼
o bir aşçı	she is a cook
o bir mühendis	he is an engineer
o bir doktor	he is a doctor
o bir hemşire	she is a nurse
o bir temizlikçi	he is a cleaner
o bir polis	He-she is a police
o bir asker	he is a soldier
o bir öğretmen	She's a teacher

Challenges

Algorithms do not exist in a bubble

- Inherit the prejudices of their designers
- Reflect cultural biases
- Difficult to identify - can entrench/enhance issues
- Deny historically disadvantaged groups full participation

Outline

- Naïve Bayes implementation
- Handout 10
- Intro to Algorithmic Bias
- **Disparate Impact**
- Feedback forms

How can we tell if an algorithm is biased?

D: dataset with attributes X , Y

- * X is protected
- * Y is unprotected (other features)

How can we tell if an algorithm is biased?

D: dataset with attributes X , Y

- * X is protected
- * Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

How can we tell if an algorithm is biased?

D: dataset with attributes X , Y

- * X is protected
- * Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

Direct discrimination: $C = f(X)$

- * Female instrumentalist not hired for orchestra
- * Some ethnic groups not allowed to eat at a restaurant

How can we tell if an algorithm is biased?

D: dataset with attributes X , Y

- * X is protected
- * Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

Indirect discrimination: $C = f(Y)$

- * but strong correlation between X and Y
- * Ex: housing loans
- * Ex: programming experience

features $\left\{ \begin{array}{l} X : \text{protected} \\ Y : \text{other attributes} \end{array} \right. \quad \left. \begin{array}{l} X=0 \\ X=1 \end{array} \right\} \begin{array}{l} \text{minority} \\ \text{majority} \end{array}$

label $\{ C : \text{binary outcome} \}$
(hired, admitted)

$C=1$, not $C=0$

legal

exam

minority group
minority group

Disparate Impact

legal definition

$$P(C=1|X=0) \leq 0.8 P(C=1|X=1)$$

example: 40% of women ^{applicants} hired

70% of men ^{hired applicants}

$$0.4 \stackrel{?}{\leq} \underbrace{0.8(0.7)}_{0.56} \left. \vphantom{0.4} \right\} \begin{array}{l} \text{yes} \\ \text{disparate} \\ \text{impact.} \end{array}$$

Outline

- Naïve Bayes implementation
- Handout 10
- Intro to Algorithmic Bias
- Disparate Impact
- Feedback forms