Ethics, Public Policy, and Technological Change

Rob Reich
Mehran Sahami
Head TA: Roberta Fischli

Housekeeping

- Technical Assignment #1 on Algorithmic Decision-Making is out
  - See “Handouts” link on class webpage to get assignment handout and information on using PyCharm

  - See “Assignment” link on class web site to get starter code

  - Using Python (and PyCharm) for the assignment
    - See “Software” link on class webpage to download PyCharm

  - Assignment #1 due at 11:59pm on January 27th
    - Submission through Gradescope
      - Will submit code and write-up to questions separately
    - Gradescope CS182 Entry Code in Assignment #1 handout (at end)

Today's Agenda

1. Introduction to machine learning and Perceptron algorithm
2. Definitions of "fairness" (with a brief intro. to probability)
3. Discussion of ProPublica analysis of COMPAS algorithm
4. Overview of technical assignment

Promises/Perils of Machine Learning

- Promises
  - Provide insights about domain
  - Improve accuracy of prediction compared to humans
    - Diminish/eliminate bias and inconsistency
  - Greater efficiency than human decision-making
    - Humans are slow and error-prone

- Perils
  - Encode existing biases and reduce fairness
  - Lack transparency and threaten due process
  - Increased efficiency is not always a benefit

Machine Learning for Prediction

- Many different forms of machine learning
  - We focus on the prediction (or classification) task
- Want to make a prediction based on observations
  - Set of n observed variables: $\langle X_1, X_2, \ldots, X_n \rangle$
    - $X_1, X_2, \ldots, X_n$ are called "input features/variables"
    - For example: age, annual income, gender, education, etc.
    - Referred to as **X** for short (it's a vector, but that's not important)
- Given observed **X**, want to predict other variable Y
  - Y called "output feature/variable"
  - Example 1: whether applicant should be issued a credit card
  - Example 2: whether defendant will commit a crime in future (recidivate)
- Seeking to "learn" a function $d(\mathbf{X})$ to predict Y:

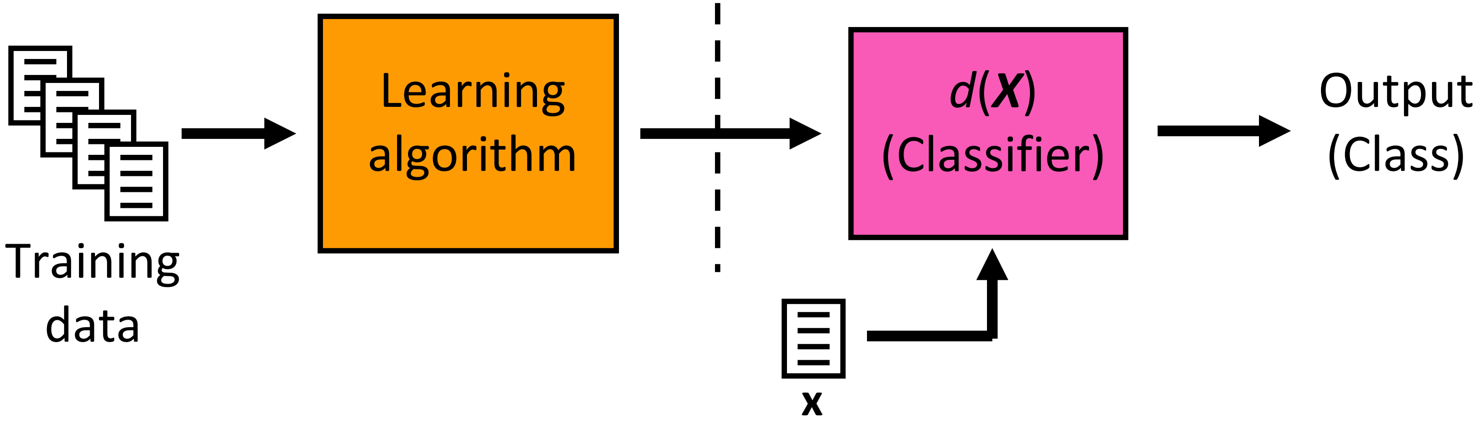
$$Y_{\text{prediction}} = d(\mathbf{X})$$

Training a Learning Machine

- We are given set of *M* "training" instances
  - Each training instance is really a pair: ($\langle x_1, x_2, \ldots, x_n \rangle$, y)
  - Training instances are previously observed data
  - Provides output value *y* associated with each observed set of input values $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$

- Learning: use training data to specify d(**X**)
  - Generally, first select a functional form for d(**X**)
  - Then, determine parameters (weights) of model d(**X**) using training data

The Machine Learning Process



<u>Training data</u>: set of M pre-classified data instances

- M training pairs: $(\mathbf{x}, y)^{(1)}, (\mathbf{x}, y)^{(2)}, \ldots, (\mathbf{x}, y)^{(M)}$
    - Use superscripts to denote i-th training instance

<u>Learning algorithm</u>: method for determining $d(\boldsymbol{X})$

- Given a new input observation of $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$
- Use $d(\boldsymbol{X})$ to compute a corresponding output (prediction)
- When prediction is discrete, we call $d(\boldsymbol{X})$ a "classifier" and sometimes call the output the predicted "class" of the input

Basic Perceptron Algorithm

```
sum = $\sum_{i=1}^{n} x_i \cdot w_i$

if sum > 0:
   prediction = 1
else:
   prediction = 0


if prediction != y:                    (incorrect prediction)
    if prediction == 1:
        for each weight w_i (where i = 1 to n)
            w_i = w_i - x_i
    else:
        for each weight w_i (where i = 1 to n)
            w_i = w_i + x_i
```

Basic Perceptron Algorithm

sum = $\sum_{i=1}^{n} x_i \cdot w_i$

```
if sum > 0:
   prediction = 1
else:
   prediction = 0


# Mathematically equivalent, but more compact update rule

error = y - prediction

if error != 0:                                      (incorrect prediction)
    for each weight w_i (where i = 1 to n)
        w_i = w_i + (error * x_i)
```

Batch Perceptron Pocket Algorithm

- Batch: for each pass through training data (called an "epoch")
  - Compute what the change in weights would be for each instance
  - Average the changes over all instances (“average difference”)
  - Update weights with average difference
- Pocket: for each pass through training data
  - Compute number of correct predictions made with current weights
  - If number of correct predictions is higher than any previous pass, save this set of weights in our “pocket”
  - After making some number of passes through the data for training, we use the set of weights in our “pocket” as the final model
- More details (and pseudocode) in the “Probability and Machine Learning” handout/reading
  - That is the algorithm implemented in Assignment #1

Today's Agenda

1. Introduction to machine learning and Perceptron algorithm
2. **Definitions of "fairness" (with a brief intro. to probability)**
3. Discussion of ProPublica analysis of COMPAS algorithm
4. Overview of technical assignment

What is "Fair"?

- There are many definitions of fairness
  - Narayanan (2018) provided 21 definitions of fairness
  - We will focus on some of the most commonly discussed definitions
  - Requires a bit of background in probability to formalize
    - So, here's a working introduction to probability

- Probability: Chance that something will happen
  - Coin flip can be heads or tails. Set $X = 1$ if heads, 0 otherwise
  - $\Pr(X = 1)$ Chance that variable $X = 1$ (flipped "heads")

- Conditional probability: Probability that something will happen given that something else has been observed
  - $\Pr(X = 1 \mid Y = 1)$ Chance that variable $X = 1$ given that we know $Y = 1$

Legal Concepts Related to Fairness

- Protected characteristics
  - Some characteristics cannot be used to discriminate individuals in decision-making in particular circumstances
  - For example, in employment decisions, protected characteristics include: race, gender, and age (among others)
  - In medicine, however, it may make sense to prescribe different treatments to different genders
- Disparate impact
  - Definition: Impact of a policy is different between two groups distinguished by a protected characteristic
  - Does not require discriminatory intent

Definitions Related to "Fairness" - I

- *Anti-classification*: decisions do not consider "protected" characteristics (e.g., race, gender, age, etc.)
  - Consider only unprotected characteristics of two individuals X and X'
  - Implies: if the unprotected characteristics of X and X' are the same, then the decision made for X and X' should be the same

- *Classification parity*: Classification error is equivalent across groups defined by protected characteristics (X_p)
  - E.g., Parity of false positives: $\Pr(d(X) = 1 \mid Y = 0, X_p) = \Pr(d(X) = 1 \mid Y = 0)$
    - If you would not recidivate, then knowing your protected characteristics should not change the probability that we predict you will recidivate (chance of false positive prediction)

Definitions Related to "Fairness" - II

- *Calibration*: Outcomes should be independent of protected characteristics conditional on risk scores, *s*(X)
  - Formally: $\Pr(Y = 1 \mid s(X), X_p) = \Pr(Y = 1 \mid s(X))$
  - Given your risk score, the probability that you will recidivate should not change if we additionally knew your protected characteristics
  - In Perceptron, could think of the sum = $\sum_{i=1}^{n} x_i \cdot w_i$ as a form of risk score that is then thresholded to make a prediction. (Risk scores often binned.)
- (Lack of) *Disparate impact*: *impact* of a policy should not be different between two groups (based on protected characteristic)
  - Recall, disparate impact does not require discriminatory intent, only that the impact is disparate between the two groups

Today's Agenda

1. Introduction to machine learning and Perceptron algorithm
2. Definitions of "fairness" (with a brief intro. to probability)
3. **Discussion of ProPublica analysis of COMPAS algorithm**
4. Overview of technical assignment

COMPAS Algorithm

Equivant Supervision Insights

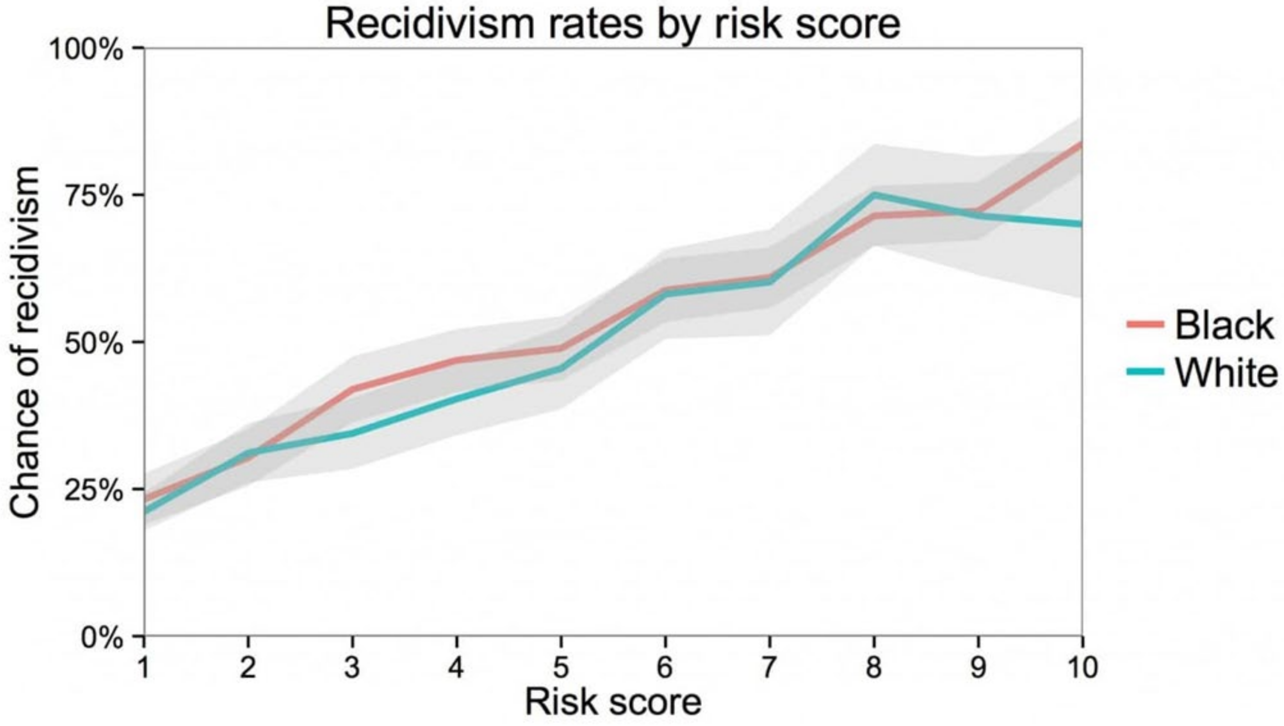
- “Opaque box” model by ~~Northpointe~~ to assess risk of recidivism
  - Predicts a risk score of recidivism based on features of individual
  - Race is not one of the input features to the model

Contingency Table	Recidivated	Did not recidivate
Labeled High-risk	True positive (A)	False positive (B)
Labeled Low-risk	False negative (C)	True negative (D)

- ProPublica analysis (no classification parity)
  - Score correctly predicted recidivism: 61% (A/(A+B))
    - Correct for white defendants: 59% (A/(A+B))
    - Correct for black defendants: 63% (A/(A+B))
  - But, the way misclassification were made were different
    - Blacks who did not recidivate, % labeled high-risk: 45% (B/(B+D))
    - Whites who did not recidivate, % labeled high-risk: 23% (B/(B+D))
    - Blacks who recidivated, % labeled low-risk: 28% (C/(A+C))
    - Whites who recidivated, % labeled low-risk: 48% (C/(A+C))

COMPAS Algorithm

- Northpointe responds that algorithm is fair because risk scores are equally predictive of recidivism for both blacks and whites
  - Calibration: $\Pr(Y = 1 \mid s(X), X_p) = \Pr(Y = 1 \mid s(X))$



Algorithms as a Mirror

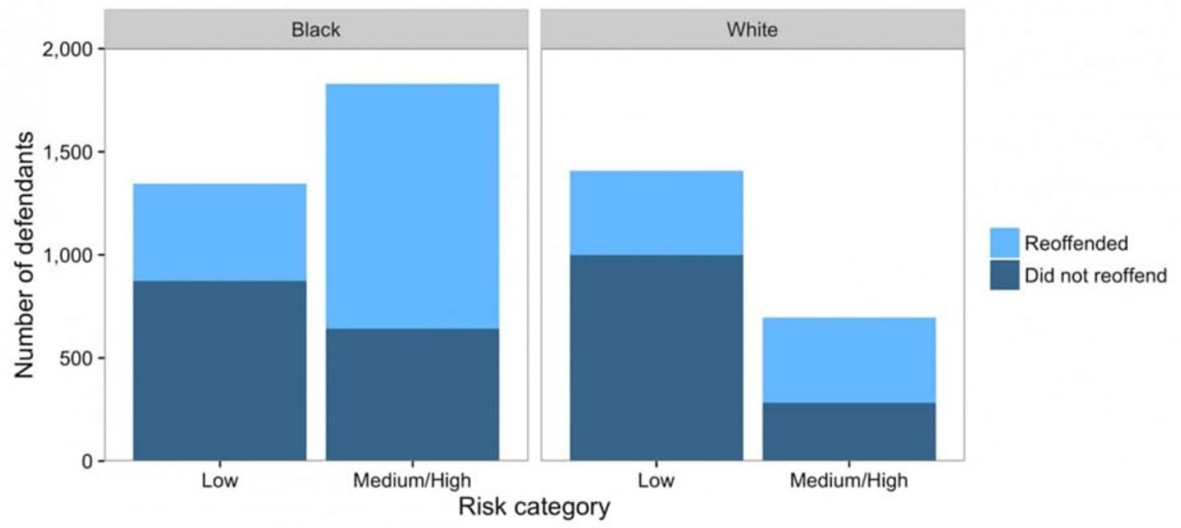
- Algorithms require formalization of what should be optimized
- In machine learning, often try to optimize for overall accuracy of predictions.
  - Any potential problems with that?
- If we want to use algorithms to make decisions, it forces us to be precise about what we think "fairness" is and how we would define it
- How do you define fairness?
- What should the algorithm try to optimize?

Here Come the Computer Scientists

- Can't we have just have all definitions of fairness
  - Let me just crank up my deep neural network…
- Sorry, Kleinberg *et al* (2017) prove you can't (generally) get both calibration and classification parity
- And, you can have proxies for protected characteristics
  - (Sets of) features that are not protected, but correlate strongly with protected features
  - And it can be hard to determine which such features should be allowed
    - Here come the lawyers…
- And, there can be historical bias or disproportionality in the data that will be reflected in results of machine learning algorithms
  - E.g., A classifier built to predict a condition that only occurs in 0.5% of the population is 99.5% accurate if it always predicts that no one has condition
- And, there's the problem of infra-marginality (Say what?!)

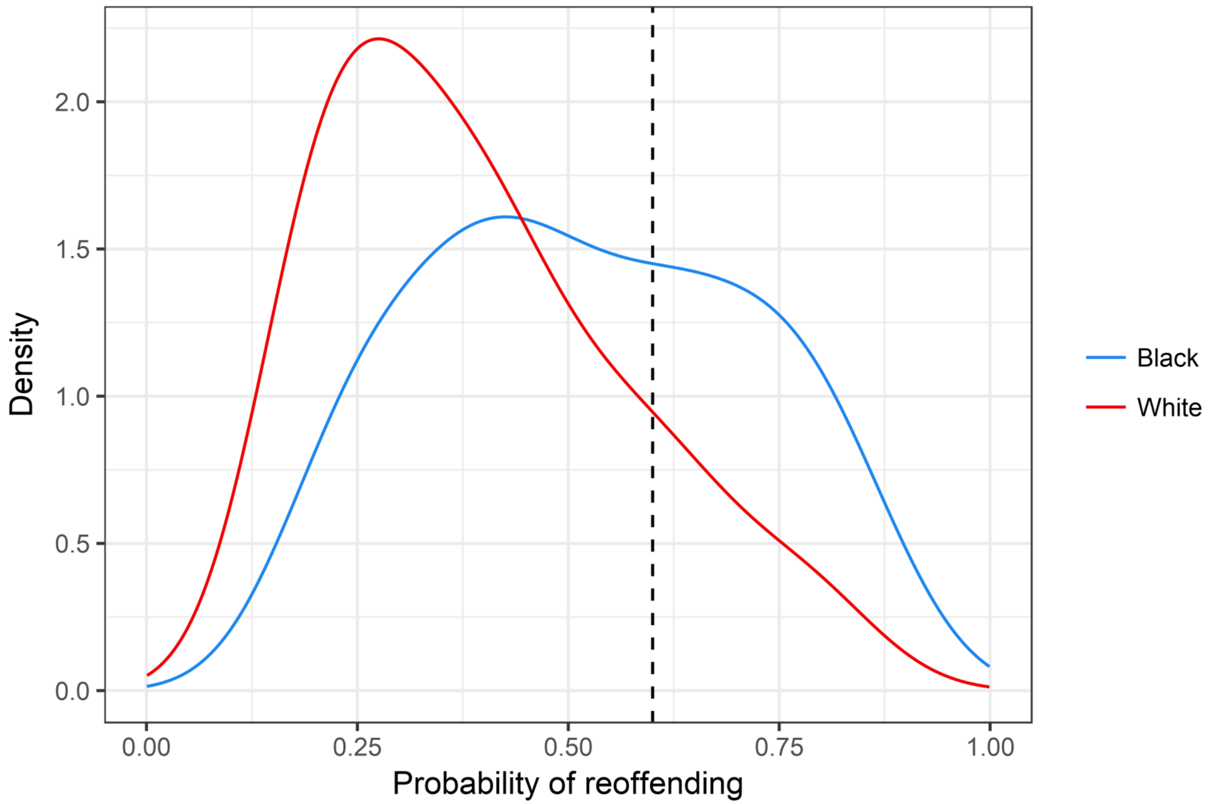
Risk Distributions Differ

- Distribution of defendants across risk categories by race (Corbett-Davies et al, 2016):



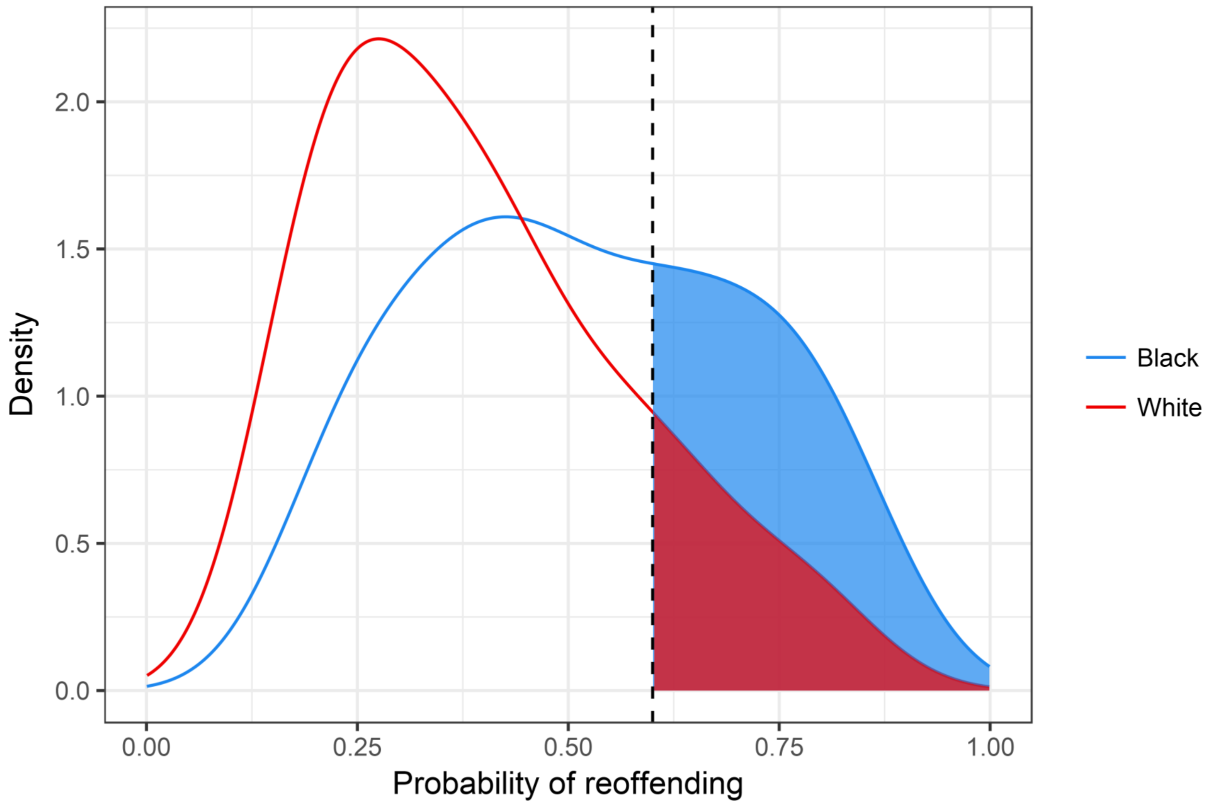
- In the data, Black defendants recidivism rate is higher than whites
  - So higher proportion of black defendants are deemed medium or high risk
  - As a result, black defendants who do not reoffend are also more likely to be classified higher risk than white defendants who do not reoffend

Risk Distributions Differ



Slide thanks to Sam Corbett-Davies

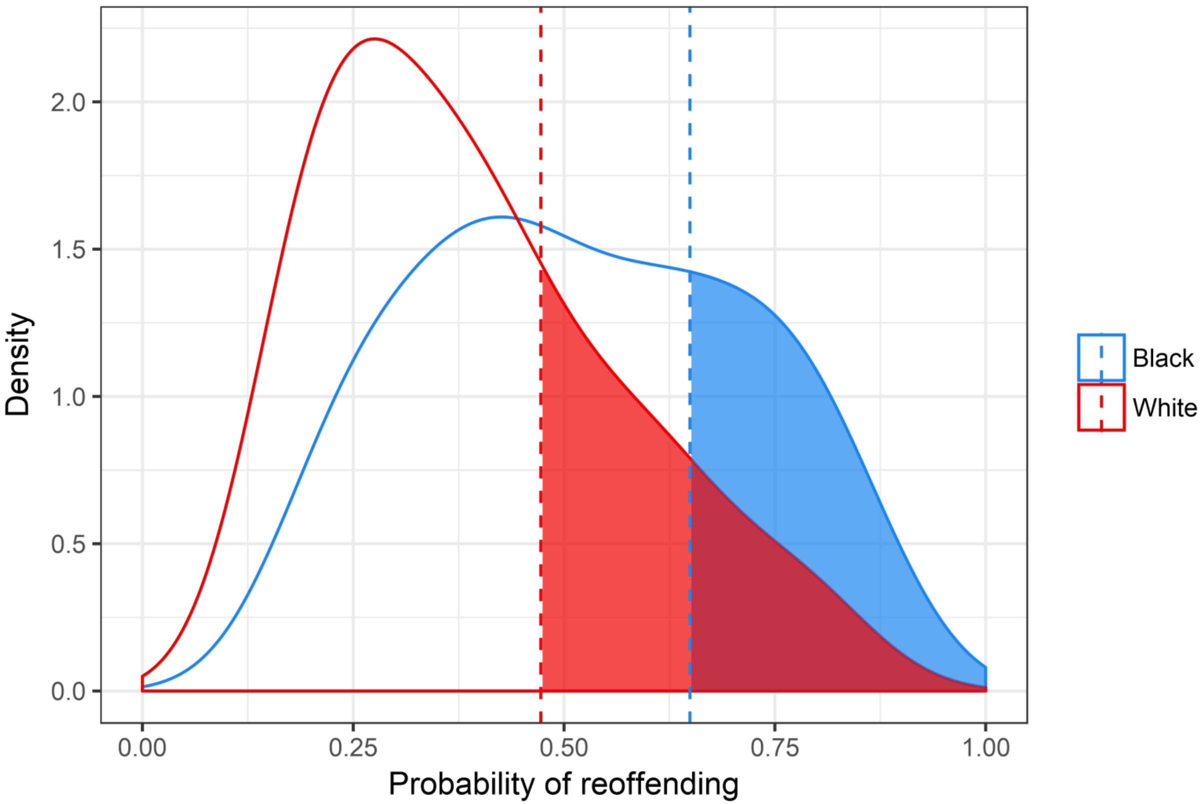
Risk Distributions Differ



- Use a single threshold based on risk scores for detention
    - Might produce disparate false positive rates (what ProPublica found)

Slide thanks to Sam Corbett-Davies

Risk Distributions Differ



- Use different thresholds based on race to equalize error rates
  - Violates notion of anti-classification since you discriminate based on a protected characteristic (race)

Slide thanks to Sam Corbett-Davies

From ProPublica

MACHINE BIAS

Bias in Criminal Risk Scores Is Mathematically Inevitable, Researchers Say

Slide thanks to Sam Corbett-Davies

Issues to Keep in Mind

- This isn't just a data issue
  - Choice of what to optimize in the model impacts results we get
- This isn't just a machine learning/modeling issue
  - Bias in data leads to bias in the model
    - Context, measurement, and representation matter
  - Sample bias – who is well represented in the data (and who is not)
  - Measurement bias – how well data reflects measurement of real-world
  - Label bias – data may not be labeled consistently
  - Exclusion bias – important aspects of data are not included
    - E.g., Complex factors of individuals not captured/represented in data
  - Real-world (prejudicial) bias – data collection reflects biased decisions in real-world
    - E.g., More crime is found in locations with more policing
- This isn't just a mathematical issue
  - Data and modeling alone do not consider broader social context of issue

Sometimes Answer is No Algorithm

- “Amazon scraps secret AI recruiting tool that showed bias against women”
  -- Business News, Oct. 9, 2018

By 2015, the company realized its new system was not rating candidates for software developer jobs and other technical posts in a gender-neutral way.

That is because Amazon’s computer models were trained to vet applicants by observing patterns in resumes submitted to the company over a 10-year period. Most came from men, a reflection of male dominance across the tech industry.

It penalized resumes that included the word “women’s,” as in “women’s chess club captain.” And it downgraded graduates of two all-women’s colleges…

The Seattle company ultimately disbanded the team by the start of last year because executives lost hope for the project…

Today’s Agenda

1. Introduction to machine learning and Perceptron algorithm
2. Definitions of “fairness” (with a brief intro. to probability)
3. Discussion of ProPublica analysis of COMPAS algorithm
4. **Overview of technical assignment**

Overview of Assignment

- Should we consider protected characteristics in an algorithm, if it can yield:
  - higher predictive accuracy?
  - error rate parity between different racial/gender groups?
  - correct for historical bias in the data?
- We want you to explore questions like this in your assignment
  - There are many other questions that could be explored
  - Some parts of assignment are focused on grappling with specific issues to keep them tractable for a class assignment
  - But, please feel free to explore more broadly in your write-up, keeping in mind the audience that you are writing for in the assignment
- Quick overview of assignment