

## Lecture 14

# ML and the Real World

[„The Algorithmic Foundations of Differential Privacy“](#) by Dwork and Roth. Chapters: 2, 3.1 – 3.3  
[„Fairness and Machine Learning“](#) by Barocas, Hardt, and Narayanan. Chapters: 3



Jilles Vreeken  
Krikamol Muandet



UNIVERSITÄT  
DES  
SAARLANDES



**CISPA**  
HELMHOLTZ CENTER FOR  
INFORMATION SECURITY

# Overview

So far, we focused on models and algorithms, optimizing for simple metrics

- e.g. misclassification rate, reconstruction error, etc

As ML/AI is becoming more widespread and now also used in critical applications

- e.g. algorithmic decision-making involving humans
- we have to consider **societal impacts**

ML models that get deployed in the real-world create **feedback loops**

- these can have potentially **unintended consequences**

We should always ask: "Are we optimizing for the right thing?"

# Accuracy Alone is not Enough

- **Privacy:** how do we prevent revealing sensitive information from training data (e.g. medical diagnosis)
- **Fairness:** how do we do not **disadvantage particular individuals** or (marginalized) groups
- **Security:** how do we prevent attackers from **fooling a system, poison training data, steal** our model, etc.?
- **Explainability:** humans should be able to understand **why and how a decision was made** about them
- **Uncertainty:** we should be able to **quantify how confident we are** in our predictions
- **Accountability:** an outside auditor should be able to verify that the system is functioning as intended

Other very important considerations include carbon footprint, fair and ethical data collection, etc.

# Impact of AI/ML More Broadly

- How should self-driving cars trade off the safety of passengers, pedestrians, etc.? (Trolley problems)
- Face recognition and other surveillance-enabling technologies
- Autonomous weapons
- Risk of international AI arms races
- Long-term risks of super-intelligent AI
- Unemployment due to automation
- Bad side effects of optimizing for click-through

# Disclaimer

These concepts sound **vague**

- properly formalizing them is half the challenge
- many of these topics are active areas of research, only started getting serious attention 5 years ago

Technology alone is not the solution – at best it is *part of* the solution

- tackling these issues must involve social/legal/political aspects
- must be an interdisciplinary effort

Today we will focus on two topics: **privacy** and **fairness**

- well-established technical principles and techniques that address parts of these problems

# Anonymization is Hard

US government releases a dataset of medical visits (Sweeney, '13)

- identifying info (names, addresses and SSNs) was removed
- data on zip code, birth date, and gender was left
- around 87% of Americans are uniquely identifiable from this triplet

Netflix Challenge: competition to improve movie recommendations

- dataset of 100 million movie ratings with anonymized user ID
- 99% of users who rated at least 6 movies could be identified by cross-referencing with IMDB reviews (associated with real names) (Narayanan & Smahtikov '08)

Re-Identifying 40% of anonymous volunteers in DNA Study (Sweeney, '13)

"A Face is Exposed for AOL Searcher No. 4417749" (Barrabo, '06)

# Example Anonymization is Hard

It is not sufficient to prevent unique identification of individuals

- if we know Gretchen is 55 and is included in this (fictional) database, we know she has 1 of 2 diseases

Name	Age	Gender	Zip	Smoker	Diagnosis
*	60-70	Male	191**	Y	Heart disease
*	60-70	Female	191**	N	Arthritis
*	60-70	Male	191**	Y	Lung cancer
*	60-70	Female	191**	N	Crohn's disease
*	60-70	Male	191**	Y	Lung cancer
*	<b>50-60</b>	<b>Female</b>	<b>191**</b>	<b>N</b>	<b>HIV</b>
*	50-60	Male	191**	Y	Lyme disease
*	50-60	Male	191**	Y	Seasonal allergies
*	<b>50-60</b>	<b>Female</b>	<b>191**</b>	<b>N</b>	<b>Ulcerative colitis</b>

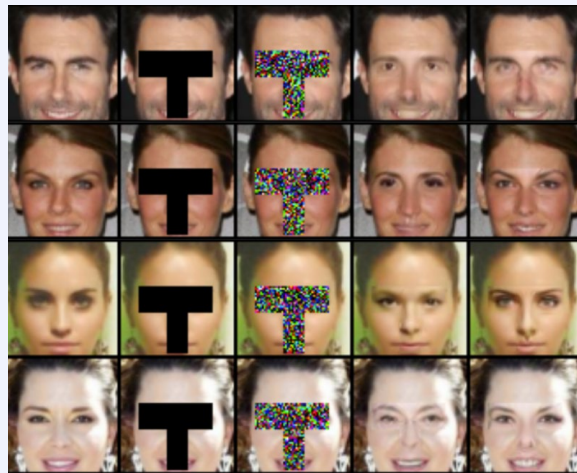
# Sensitive Information in the Model

Even if you don't release the raw data, the weights of a trained network might reveal sensitive information

**Model inversion** attacks recover information about the training data from the trained model

**Example 1:** Reconstructing faces given a classifier trained on private data, using a generative model trained on public data

**Example 2:** Email provider uses language models for email autocompletion, the model can remember (and spit out) sensitive info from past emails



reconstructing faces (Zhang et al., 2019)



# The Main Question in Differential Privacy



How can we  
compute statistical queries and train ML models  
without leaking too much  
sensitive information about any individual?

# Warmup: Randomized Response

**Goal:** Conduct a survey on a sensitive question with a binary (yes/no) answer

Examples

- Have you ever committed tax fraud?
- Does anyone in your family suffer from HIV?

We want to motivate participants to answer truthfully, despite sensitive nature of the question

**Idea:** introduce randomization to provide **plausible deniability**

# Warmup: Randomized Response

Let each of the  $n$  participants follow the procedure

- flip a coin
- if it lands tails, answer truthfully
- otherwise, flip another coin, if that one lands tails answer **Yes**, else answer **No**

What is the fraction of participants that answer Yes truthfully?

We can accurately estimate the population mean  $\mu$  as  $\mu = \frac{1}{4}(1 - \hat{\mu}) + \frac{3}{4}(\hat{\mu})$

- $\hat{\mu}$  is the MLE (i.e. the counts)
- $P(\text{response} = \text{Yes} \mid \text{truth} = \text{Yes}) = \frac{3}{4}$
- $P(\text{response} = \text{Yes} \mid \text{truth} = \text{No}) = \frac{1}{4}$
- $\mu$  is an unbiased estimator of the non-randomized mean  $\hat{\mu}$
- variance decays as  $\frac{1}{n}$  but is 4x larger because of the randomization

# Beyond Randomized Response

With **randomized response** we can compute useful queries (e.g. the fraction) in aggregate without learning the truthful answer for any individual

In general, randomness is a useful technique for preventing information leakage

What if we want to answer more complex queries with mathematical privacy guarantees?

- e.g. computing arbitrary functions over data
- answer: **Differential Privacy**

# Differential Privacy (DP)

A (trusted) curator is given access to some input data  $\mathbf{X} \in \mathcal{X}$

- the curator computes some function  $Y = f(\mathbf{x})$
- the curator wants to release the output  $Y \in \mathcal{Y}$  to the public without leaking too much information

## Example

- let  $\mathcal{X} = \{0, 1\}^n$  is the set of binary vectors, e.g. containing the answers to a survey question for  $n$  users
- let  $f$  be the mean, thus  $\mathcal{Y} \in [0, 1]$
- let  $\mathbf{X}, \mathbf{X}' \in \mathcal{X}$  be two "neighboring" input vectors, such that  $\mathbf{X}'$  differs from  $\mathbf{X}$  in only one position
  - they differ in the answer of a single participant

Informally, differential privacy enforces that  $f(\mathbf{X})$  and  $f(\mathbf{X}')$  do not differ significantly

- preventing leaking the answer of also of new participants

# Differential Privacy

Given an input and output spaces  $\mathcal{X}$  and  $\mathcal{Y}$ , a symmetric neighboring relation  $\simeq$ , a function of interest  $f$ , and a privacy parameter  $\epsilon \geq 0$

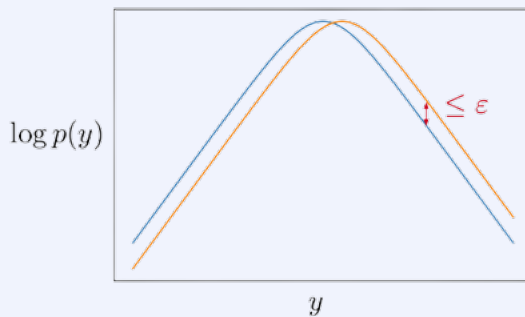
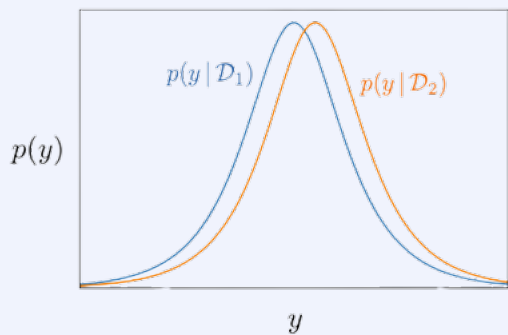
**Definition:** A randomized mechanism  $\mathcal{M}_f: \mathcal{X} \rightarrow \mathcal{Y}$  is  $\epsilon$ -differentially private if **for all** neighboring inputs  $X \simeq X'$  and **for all** sets of outputs  $Y \subseteq \mathcal{Y}$  we have:

$$e^{-\epsilon} \leq \frac{P(\mathcal{M}_f(X) \in Y)}{P(\mathcal{M}_f(X') \in Y)} \leq e^{\epsilon}$$

$\mathcal{M}_f$  includes the function  $f$  we want to compute, it is not useful to output random numbers

# Intuition Differential Privacy

The outcome should not change by much if we only modify a single instance



difference between two neighboring datasets

- $\approx$  captures what is protected, e.g. two different vectors  $\mathbf{X}$  and  $\mathbf{X}'$  that differ in a single coordinate, or two different datasets  $\mathcal{D}$  and  $\mathcal{D}'$  that differ in single instance
- if the mechanism  $\mathcal{M}_f$  behaves nearly identically  $\mathbf{X}$  and  $\mathbf{X}'$  an attacker can't tell whether  $\mathbf{X}$  or  $\mathbf{X}'$  was used and thus can't learn much about the individual

# Laplace Mechanism (Output Perturbation)

Define the global sensitivity of a function  $f: X \rightarrow \mathbb{R}^d$  as  $\Delta = \max_{X \approx X'} \|f(X) - f(X')\|_1$

- it measures the magnitude by which a single instance can change the output of  $f$  in the **worst case**

Output perturbation with Laplace noise

- a curator holds data  $X$
- the curator computes the function  $f(X)$
- they sample Laplace noise  $Z \sim \text{Lap}\left(0, \frac{\Delta}{\epsilon}\right)^d$  i.i.d. for each dimension
- they reveal the noisy value  $f(X) + Z$

We can prove that this mechanism is  $\epsilon$ -differentially private



# Example Laplace Mechanism for the Mean

Computing the mean  $\mu = f(X) = f(x_1, \dots, x_n) = \frac{1}{n} \sum x_i$  where  $x_i \in \{0, 1\}$  are binary

The global sensitivity is  $\Delta = \frac{1}{n}$

- changing the value of a single instance can change the output by at most  $\frac{1}{n}$

Sample noisy  $Z \sim \text{Lap}\left(0, \frac{1}{\epsilon n}\right)$  and release the noisy mean  $\hat{\mu} = \mu + Z$

In this case we can also say something about the **utility** of this mechanism

- $|\hat{\mu} - \mu| = \text{Exponential}(\epsilon n)$  which has a mean of  $\frac{1}{\epsilon n}$
- the true mean is not going to differ much from the randomized mean, difference decreases with size of the data
- in general, computing the sensitivity of a function is challenging, showing something about utility even more so

# Differential Privacy for Machine Learning

**Perturb input:** perturb  $\mathcal{D}$  and directly and rely on the post-processing property

- **robustness to post-processing:** if  $\mathcal{M}$  is  $\epsilon$ -DP then  $g \circ \mathcal{M}$  is  $\epsilon$ -DP for any function  $g$
- you can apply any function  $g$  on an output from a DP mechanism and the new output remains DP, as long as you don't touch again the data

**Perturb weights:** compute the optimal param  $\theta^*$  and perturb them with Laplace noise

- need to calculate the global sensitivity of the optimization procedure which can be extremely difficult

**Perturb objective:** optimize  $\mathcal{L}(\mathcal{D}, \theta) + \theta^T \mathbf{Z}$  where  $\mathbf{Z}$  is some carefully selected noise

**Perturb gradients:** perturb and release the gradient of  $\mathcal{L}$  w.r.t. a mini-batch of the data

- useful in **federated learning** where we have no centralized entity that has access to all the data

# Differential Privacy Summary

A lot of ML models are trained on datasets containing sensitive information about individuals, and database reconstruction attacks can be surprisingly effective

Differential privacy gives a way of provably preventing (much) information about individuals from leaking

The Laplace mechanism is an important building blocks of differential privacy

Differentially private algorithms can accurately answer queries for large populations

- The 2020 US Census used differential privacy

# Motivation: The Influence of Biased Algorithms

- **Selecting job applicants:** XING ranks less qualified male candidates higher than more qualified female candidates (Lahoti et al. 2018)
- **Recidivism prediction and predictive policing:** COMPAS: high-risk FP: 23.5% for white vs. 44.9% for black, and low-risk FP: 47.7% for white vs. 28.0% for black ([ProPublica article](#))
- **Facial recognition:** Commercial software has much lower accuracy on females with darker color (Buolamwini and Gebru, 2018)
- **Search and recommendations:** Search queries for African-American names more likely to return ads suggestive of an arrest (Sweeney, 2019)
- **Bias found in word embeddings and translation:** man-woman=surgeon-nurse (Bolukbasi et al. 2016)

# What Causes the Bias?

- **Tainted training data:** any ML system maintains (and amplifies) the existing bias in the data caused by human bias, e.g. hiring decisions made by a (biased) manager used as labels, historic and systematic biases in the data collection process, etc.
- **Skewed sample:** initial predictions influence future observations, e.g. regions with initial high crime rate get more police attention (and thus higher recorded crime in the future), **selection bias**
- **Proxies:** even if we exclude legally protected features (e.g. race, gender, sexuality) other features may be highly correlated with these
- **Sample size disparity:** models will tend to fit the larger groups first (possibly) trading off accuracy for minority groups
- **Limited features:** features may be less informative or reliably collected for minority groups

# Why Fairness is Hard

How to define fairness?

How can we formulate it so it can be considered in ML systems?

Two distinct notions from the law (Barocas and Selbst, 2016):

- **Disparate treatment:** decisions are (partly) based on the subject's sensitive attribute
- **Disparate impact:** disproportionately hurt (or benefit) people with certain sensitive attribute values

Currently, no consensus on the mathematical formulations of fairness

# An Illustrating Example

We are a bank trying to fairly decide who should get a loan

- i.e. predict which people will likely pay us back and which will default

We have two groups: blue and orange (the sensitive attribute)

- this is where discrimination could occur



# Definitions of Fairness

How can we test if our (loan repay) classifier is fair?

**Group** fairness: aim to treat all groups equally

- e.g. we can require that the same percentage of blue and orange receive loans
- or equal false negative rates  $P(\text{no loan} \mid \text{would repay, orange}) = P(\text{no loan} \mid \text{would repay, blue})$

**Individual** fairness: treat similar examples similarly for an appropriate definition of similarity

**Counterfactual** fairness: same decision in the actual world and a counterfactual world where the individual belongs to a different group



# Group Fairness Setup

Consider binary classification with single sensitive attribute for simplicity:

- $X$  are the features of an individual (e.g. credit history)
- $A = \{a, b, c, \dots\}$  is the sensitive attribute (gender, race, etc.)
- $R = r(X, A) \in \{0, 1\}$  is the binary predictor (e.g. to grant a loan or not) which makes a decision
- $Y \in \{0, 1\}$  is the target variable representing the ground truth

Assume that  $(X, A, Y) \sim \mathcal{D}$  are generated from an underlying data distribution

Then  $X, A, Y$  and  $R$  are all random variables

Shortcut notation for the probability conditional on group  $a$ :  $P_a(R) = P(R \mid A = a)$

# Naive Approach: Fairness Through Unawareness

We should not include the sensitive attribute as a feature in the training data

$R = r(X)$  instead of  $R = r(X, A)$

Pros/Cons:

- intuitive, easy to use and implement
- consistent with disparate treatment which has legal support (e.g. the "General Equal Treatment Act" in Germany)
- however, there can be many highly correlated features (e.g. neighborhood) that are proxies of the sensitive attribute (e.g. race)

# First Criterion: Independence

Require:  $R$  independent of  $A$ , denoted  $R \perp A$

- also called Demographic Parity, Statistical Parity, Group Fairness, Darlington Criterion (4)

In case of binary classification for all groups  $a, b$  it has to hold  $P_a(R = 1) = P_b(R = 1)$

- there are also approximate version where we allow the probabilities to be approximately equal ( $\pm\epsilon$ )

In our example, this means that the **acceptance rates** of the applicants from the two groups must be equal, i.e. same percentage of applications receive loans

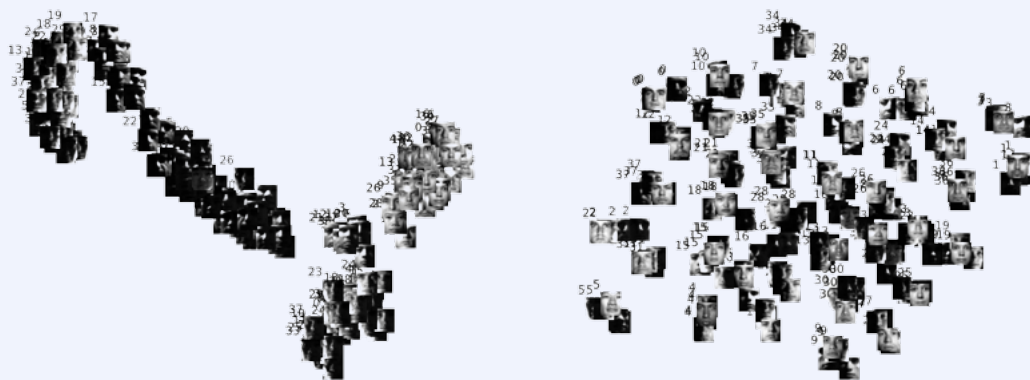
# How to Achieve Independence?

**Post-processing:** adjust a learned classifier so as to be uncorrelated with the sensitive attribute

**Training time constraint:** include the exact/approximate constraints in the optimization

**Pre-processing:** e.g. via representation learning

- map the instances into some space where information about  $A$  is destroyed (e.g. fair PCA)
- example representations learned by a variational fair autoencoder (Louizos et al., 2016)



# Pros/Cons of Independence

Legal support: "**four-fifth rule**" prescribes that a selection rate for any disadvantaged group that is less than four-fifths of that for the group with the highest rate must be justified

What if 83% of Blue is likely to repay, but only 43% of Orange is?

- then independence is too strong
- rules out perfect predictor  $R = Y$  when the base rates are different

Laziness: we can trivially satisfy the criterion if we give loan to qualified people from one group and random people from the other

# Second Criterion: Separation

Require:  $R$  and  $A$  to be independent **conditional** on the target  $Y$ , denoted  $R \perp A \mid Y$

- also called Equalized Odds, Conditional Procedure Accuracy, Avoiding Disparate Mistreatment

In case of binary classification for all groups  $a, b$  it has to hold

$$P_a(R = 1 \mid Y = 1) = P_b(R = 1 \mid Y = 1) \quad \text{equal true positive (TP)}$$

$$P_a(R = 1 \mid Y = 0) = P_b(R = 1 \mid Y = 0) \quad \text{equal false positive (FP)}$$

**Equality of opportunity** is a commonly used relaxation where we only match TP

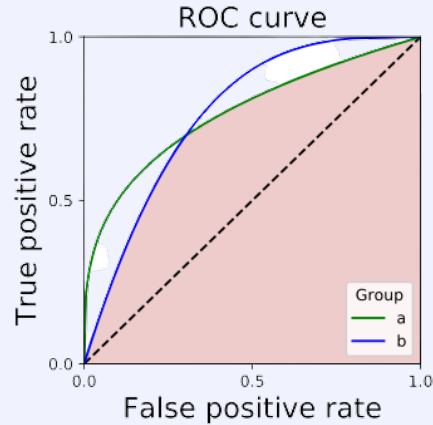
- in our example, this means we should give loan to equal proportion of individuals who would repay

# Achieving Separation

Area under the ROC (Receiver Operating Characteristic) curve

- each point on the solid curves is realized by thresholding the predicted score at some value
- i.e. predict  $r(\mathbf{X}, \mathbf{A}) > t$  for some threshold  $t$

Pick a classifier that minimizes the given cost (e.g. maximizes profit)



intersection of areas  
under the curves  
for two groups

# Pros/Cons of Separation

Optimal predictor not ruled out:  $R = Y$  is now allowed

Penalizes laziness: it provides incentive to reduce errors uniformly in all groups

It may not help closing the gap between two groups

- granting more loans to the group that is more likely to repay **now** makes the groups more likely to have better living conditions and thus even more likely to repay in the **future**, thus widening the gap



# Third Criterion: Sufficiency

Require  $Y$  and  $A$  to be independent conditional on  $R$ , denoted  $Y \perp A \mid R$

- also called Cleary model, Conditional Use Accuracy, Calibration Within Groups

In case of binary classification for all groups  $a, b$  and all output probabilities  $r$  it has to hold

$$P_a(Y = 1 \mid R = r) = P_b(Y = 1 \mid R = r)$$

In our example, the score used to determine if a candidate would repay should reflect the candidate's real/actual capability of repaying

# Achieving Sufficiency

In general a classifier  $R$  is **calibrated** if for all  $r \in [0, 1]$  we have  $P(Y = 1 | R = r) = r$

- of all instances assigned a probability or score value  $r$  an  $r$  fraction of them should be positive

Calibration for each group implies sufficiency:  $P_a(Y = 1 | R = r) = r$  for all groups  $a$

We can apply standard calibration techniques to each group (if necessary)

- **Platt scaling**: given an uncalibrated score treat it as a single feature and fit a one variable regression model against  $Y$

# Pros/Cons of Sufficiency

Satisfied by the Bayes optimal classifier

For predicting  $Y$  we do not need to see  $A$  when we have  $R$

Equal chance of success ( $Y = 1$ ) given acceptance ( $R = 1$ )

Similar to before it may not help closing the gap between the groups

# Fairness Summary: Growing List of Criteria

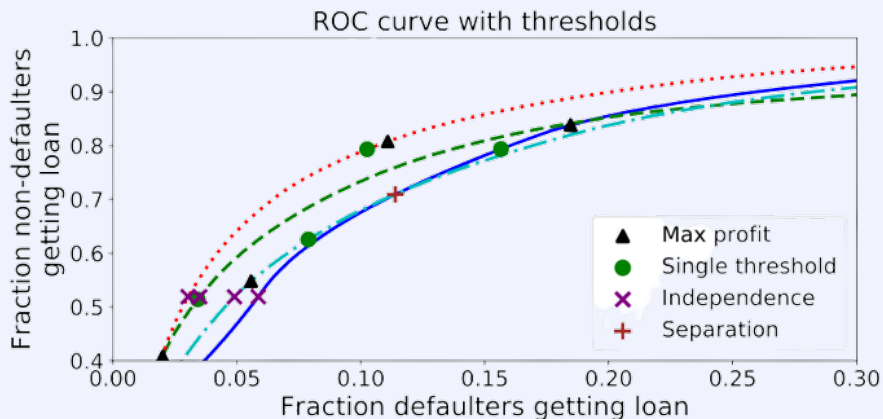
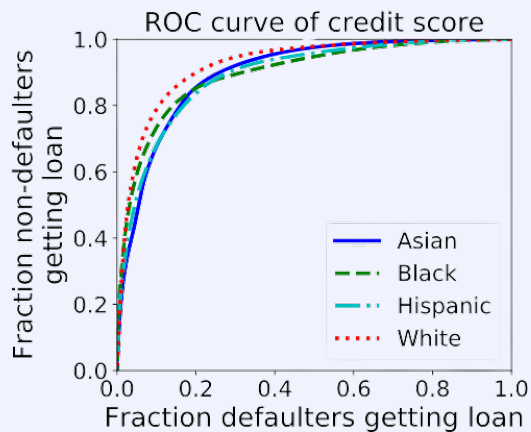
- Independence:  $R \perp A$
- Separation:  $R \perp A \mid Y$
- Equality of opportunity:  $R \perp A \mid Y = 1$
- Sufficiency:  $Y \perp A \mid R$
- ... and many many more

Many of these definitions are **provably incompatible**, i.e. they are mutually exclusive except in degenerate cases

# Comparing Different Criteria

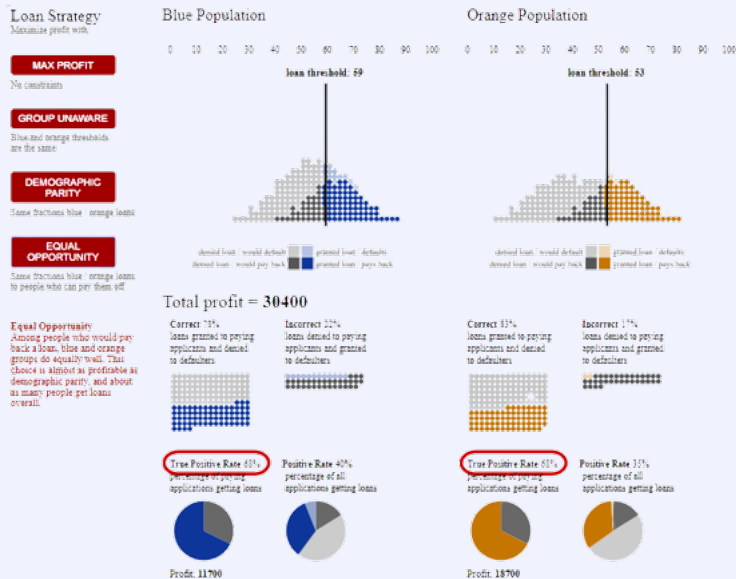
The cost of FP is typically much greater than the profit for TP

- example: different thresholds induced by different criteria (Hardt et al., 2016)



# Visualizing the Trade-offs

## Attacking discrimination with smarter machine learning



# Bonus: Robustness to Adversarial Examples

Deliberate data perturbations designed to achieve a specific malicious goal (misclassification)



predicted: panda

×



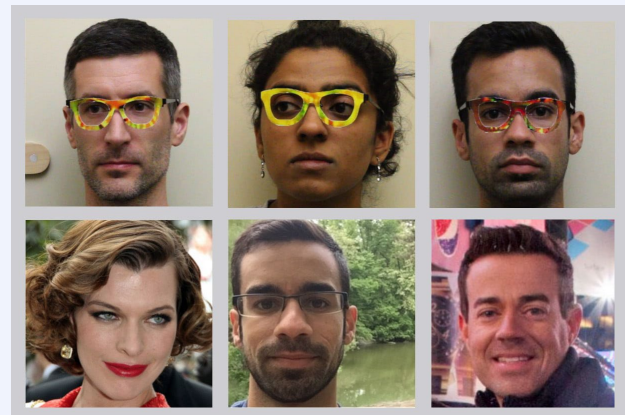
(Goodfellow et. al, 2014)



predicted: gibbon



the ML system classifies the adversarially modified stop sign as a speed limit sign (Eykholt et al., 2018)



adversarial glasses fool facial recognition systems into classifying the wearer as someone else (Sharif et al., 2016)

# Summary

Decisions based on data are not always accurate, reliable, or fair

DP allows us to compute arbitrary queries on (sensitive) data with provable guarantees

- there are no absolute privacy guarantees, your neighbor's habits are correlated with your habits

Fairness criteria require (and enforce) some invariances w.r.t. sensitive attributes

- algorithmic fairness  $\neq$  actual fairness, social/legal/political effort is also needed
- without a model of long-term impact it is difficult to foresee the effects of a fairness criterion

Accuracy, Fairness, Privacy, Robustness, and other aspects are non-trivially related

Algorithmic solutions are only (small) part of the puzzle