

Information Security and Privacy (COM-402)

Machine Learning Security and Privacy

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

This lecture

- Machine Learning – basics
- Privacy implications of Machine Learning
- Machine learning under adversarial conditions
- Issues with applying Machine Learning to Security and Privacy problems

This lecture

- **Machine Learning – basics**
- Privacy implications of Machine Learning
- Machine learning under adversarial conditions
- Issues with applying Machine Learning to Security and Privacy problems



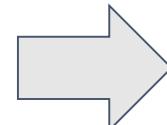
Refreshing concepts for some

Machine Learning (ML)

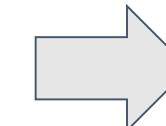
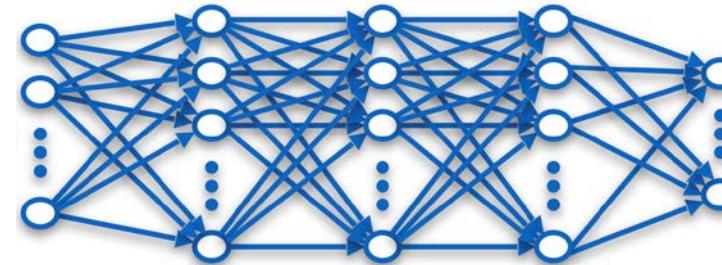
Definition (Wikipedia)

Machine learning [...] gives "**computers the ability to learn without being explicitly programmed**" [and] [...] explores the study and construction of **algorithms that can learn from and make predictions on data** – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs.

User data



Machine learning

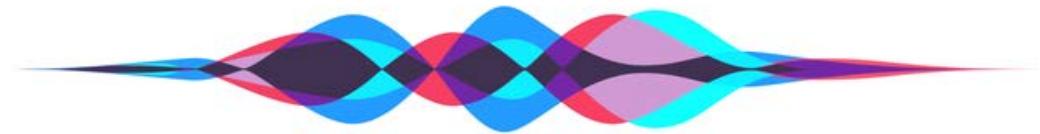


Services



Machine Learning Is Becoming Ubiquitous

1. Data security
2. Personal security
3. Financial trading
4. Healthcare
5. Marketing personalization
6. Fraud detection
7. Recommendations
8. Online search
9. Natural language processing (NLP)
10. Smart (autonomous) cars
- ...

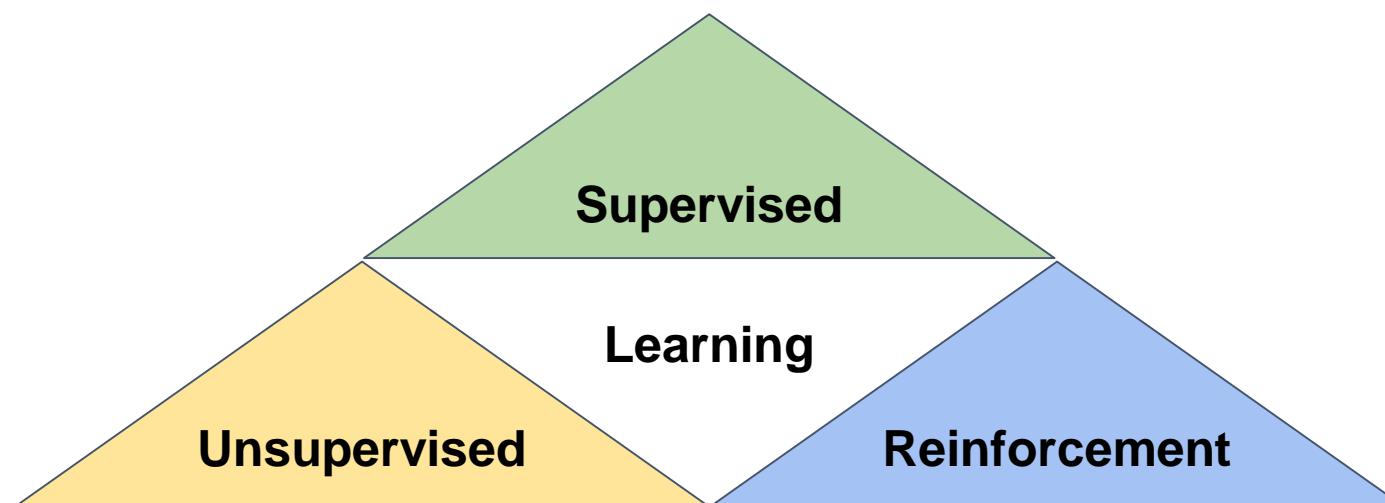


NETFLIX

Machine Learning Taxonomy

Machine learning can be separated into 3 main categories

- Labeled data
- Direct feedback
- Predict outcome/future

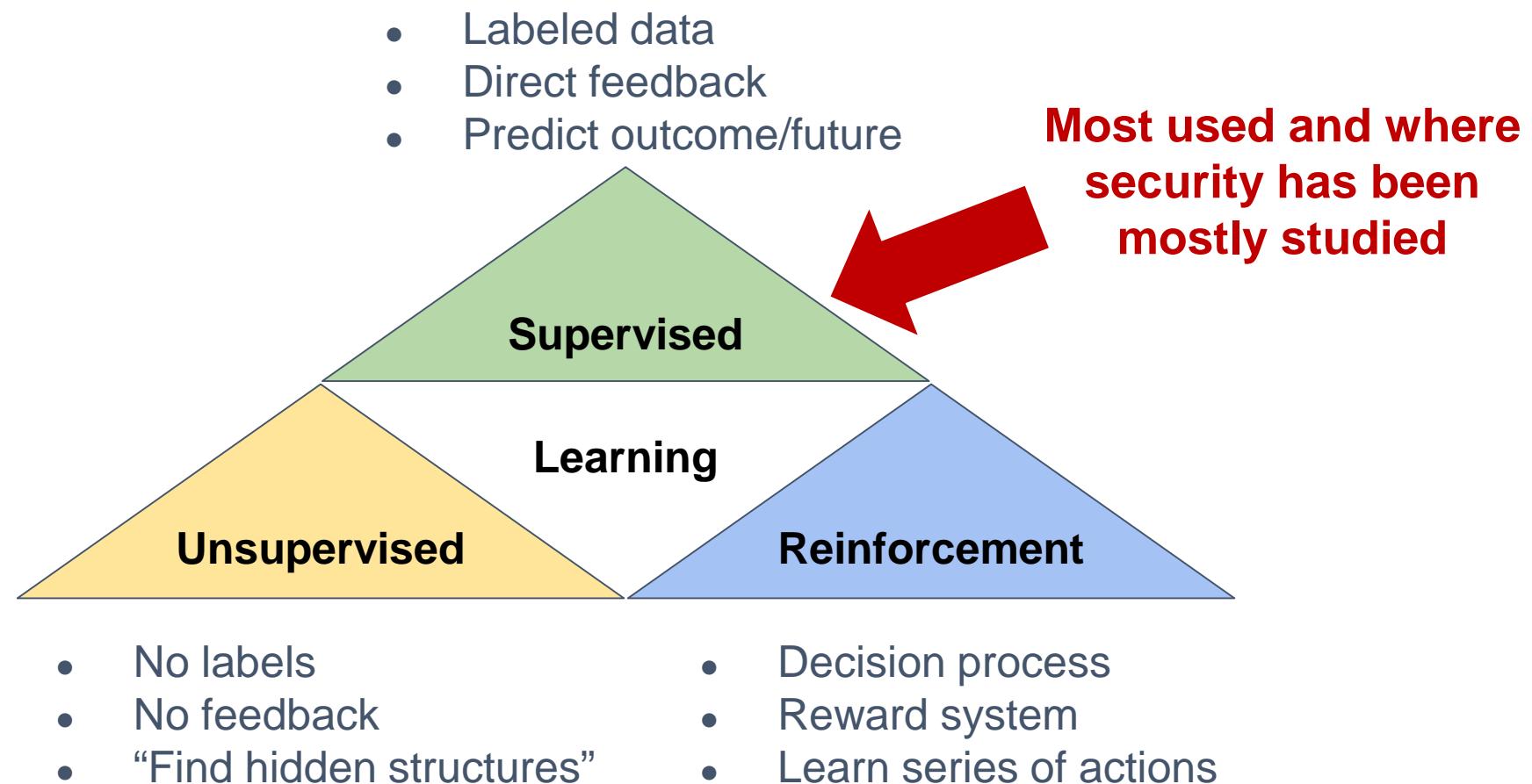


- No labels
- No feedback
- “Find hidden structures”

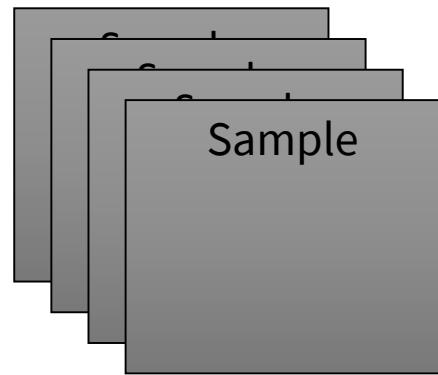
- Decision process
- Reward system
- Learn series of actions

Machine Learning Taxonomy

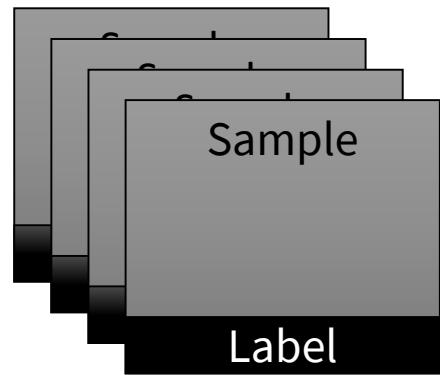
Machine learning can be separated into 3 main categories



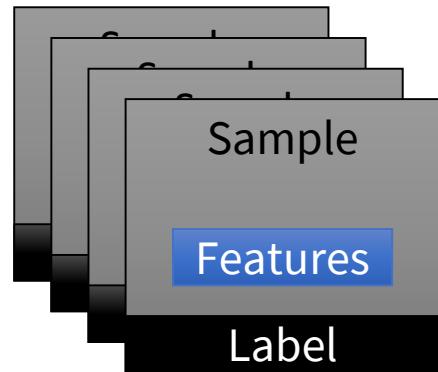
Supervised Learning



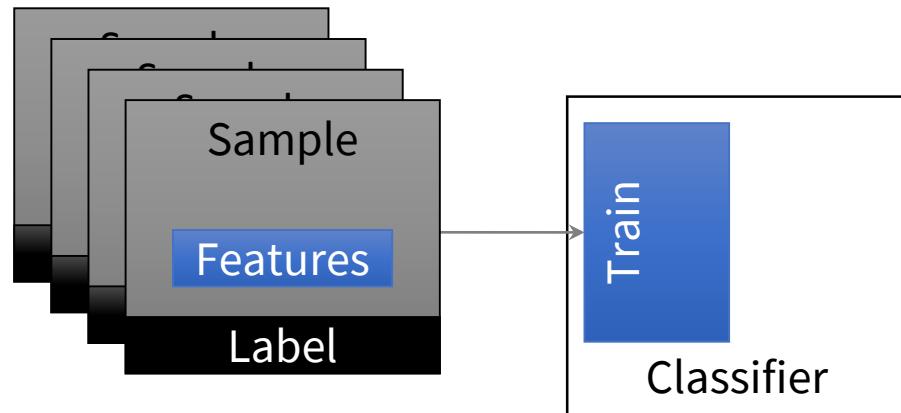
Supervised Learning



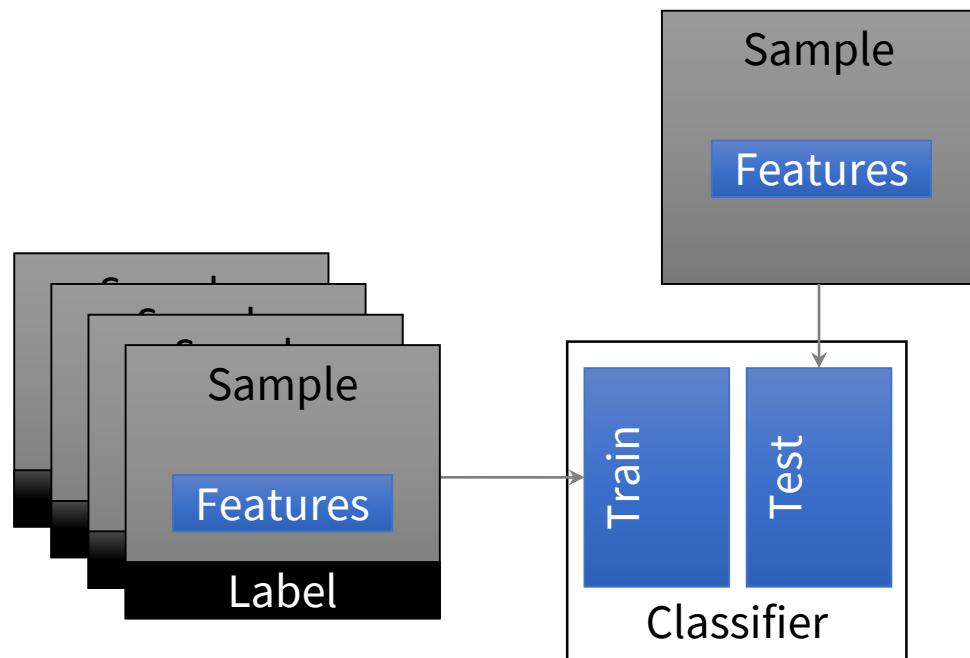
Supervised Learning



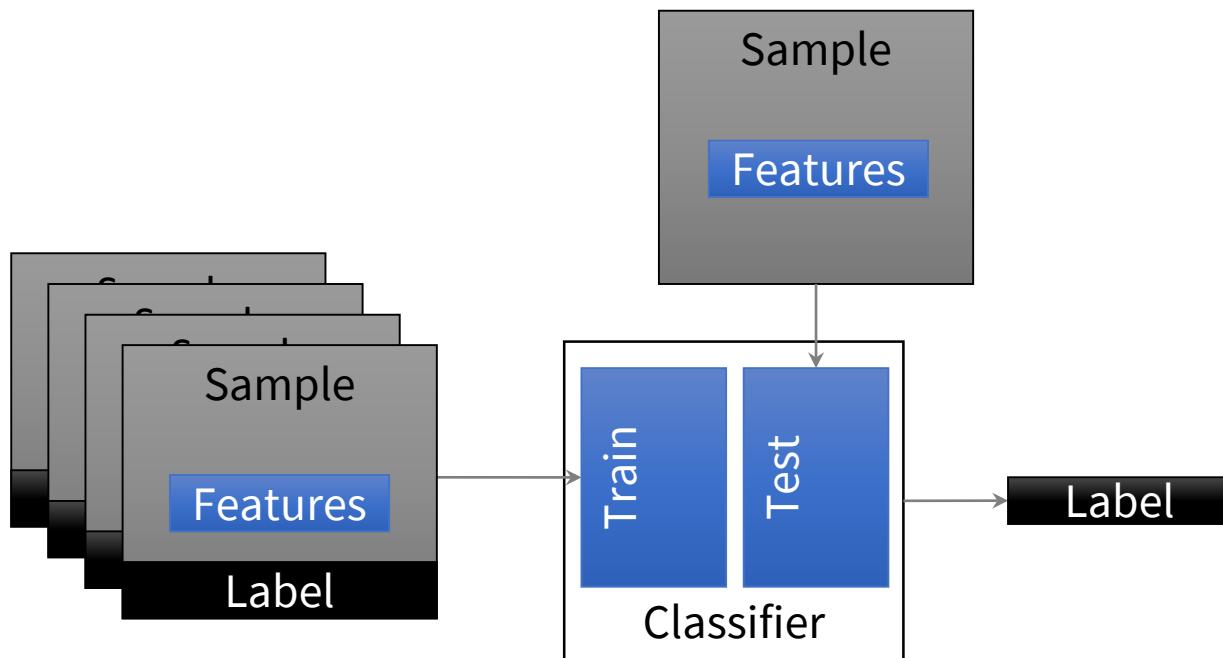
Supervised Learning



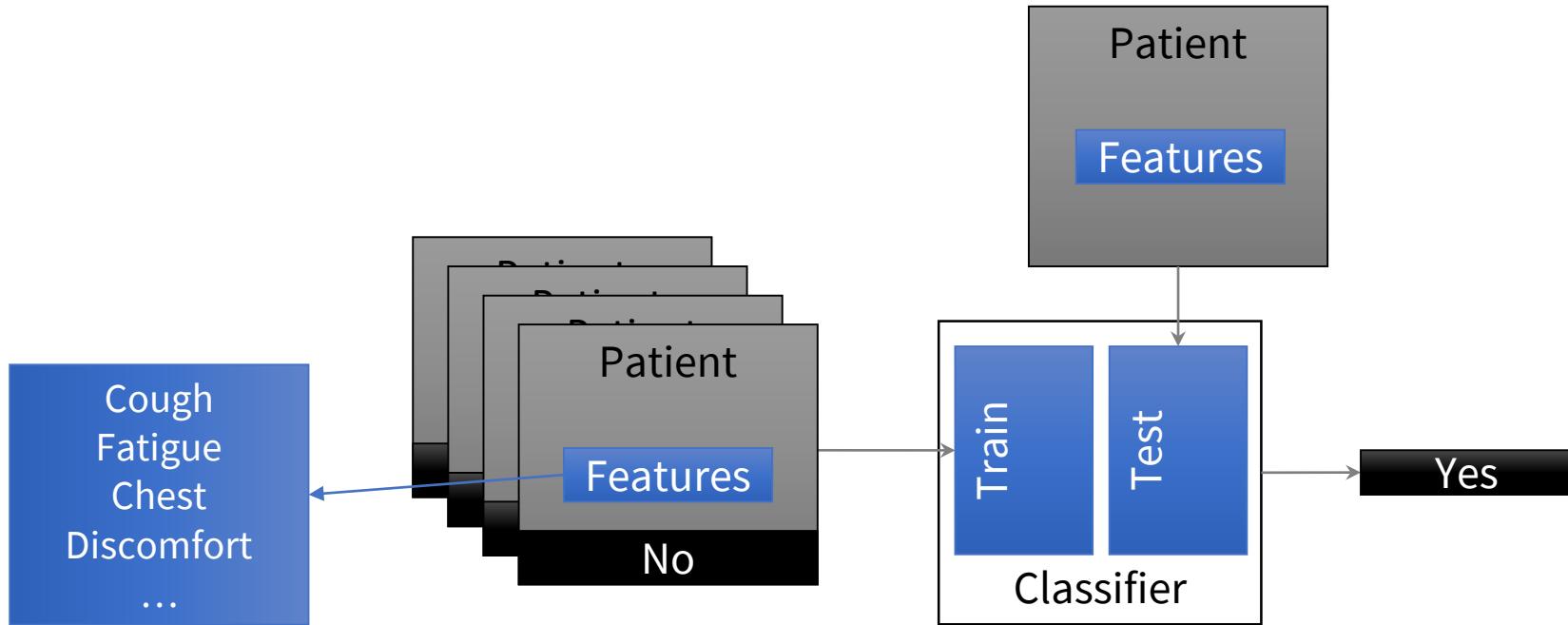
Supervised Learning



Supervised Learning

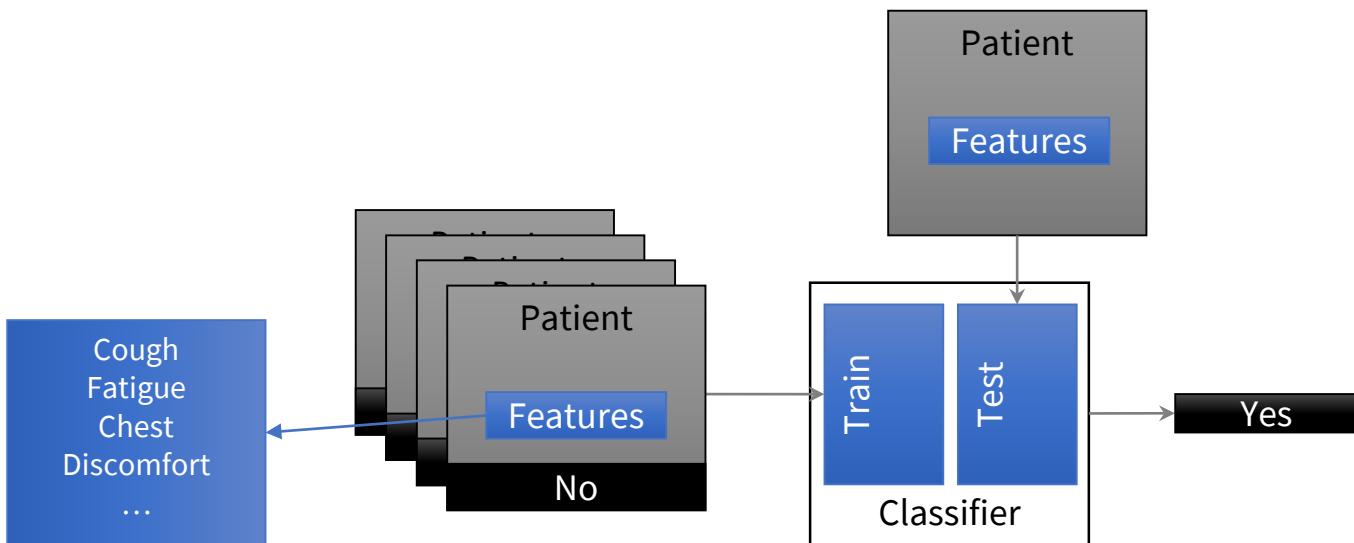


Supervised Learning



Will this patient need to be hospitalized for longer than 1 week?

Supervised Learning

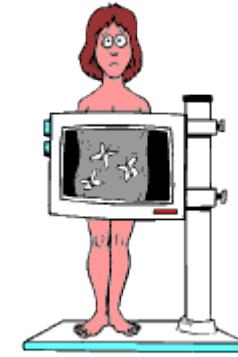


More formally

- Given: Training data $\{(x_1, y_1), \dots, (x_n, y_n)\} \in X \times Y$
 - X : Input space
 - Y : Output space
 - x_i : Feature vector
 - y_i : Output label (= class)
- Goal: Infer a function $f: X \rightarrow Y$ that matches the training data
- Types:
 - Classification (Y categories)
 - Regression ($Y = \mathbb{R}$)

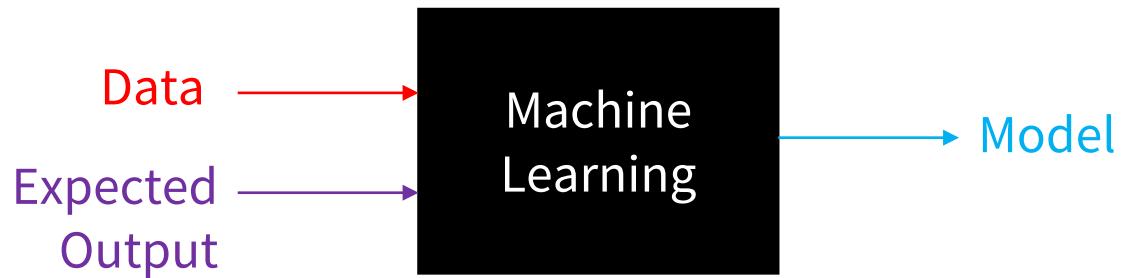
This lecture

- Machine Learning – basics
- **Privacy implications of Machine Learning**
- Machine learning under adversarial conditions
- Issues with applying Machine Learning to Security and Privacy problems



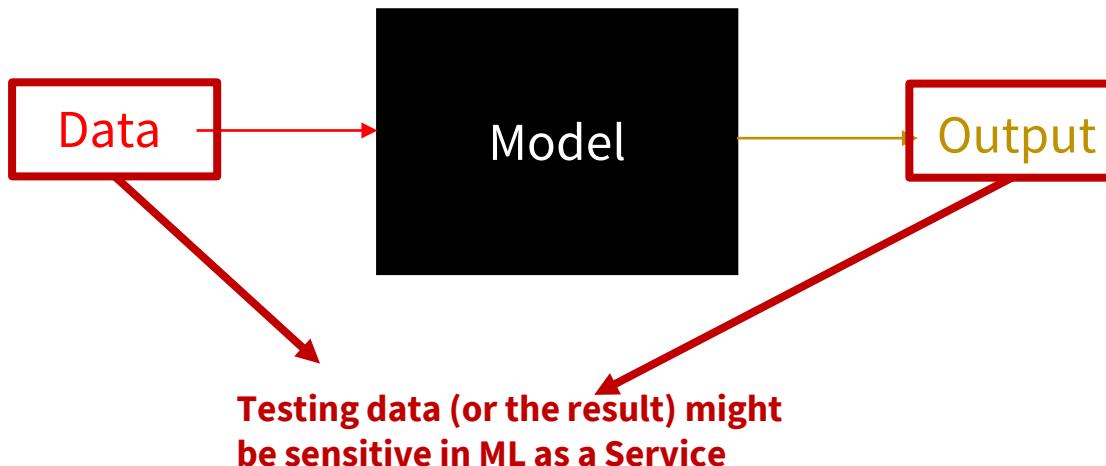
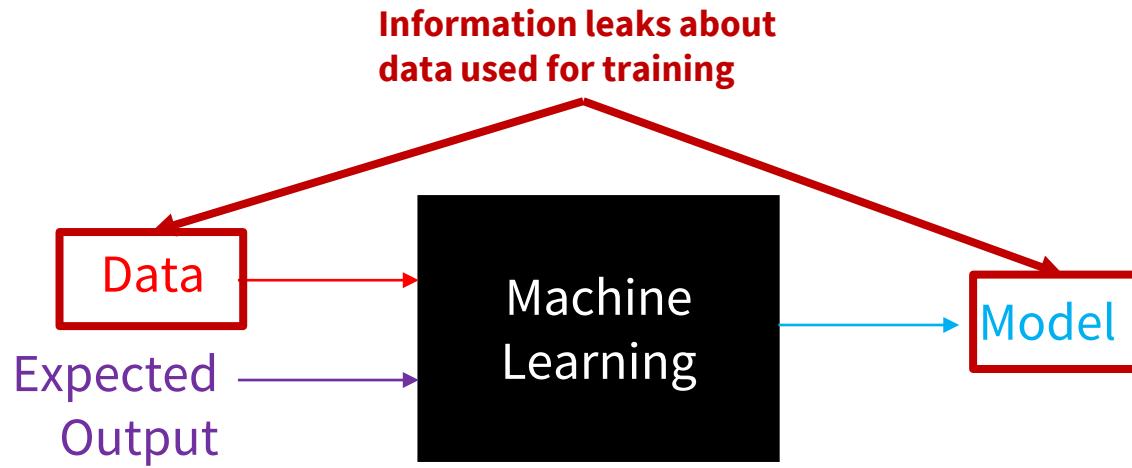
ML seems like a black box,
but it operates on data.
What can we learn from it?

Machine Learning – Privacy concerns?



1. **Data Mining:** Get data that you know something about
2. **Train the Model:** “Teach” the machine about that data
3. **Test the model:** “Check” that the machine on something you know
4. **Deploy the program!:** Use the learnings to classify/predict on new data

Machine Learning – Privacy concerns?



1. **Data Mining:** Get data that you know something about
1. **Train the Model:** “Teach” the machine about that data
1. **Test the model:** “Check” that the machine on something you know
1. **Deploy the program!:** Use the learnings to classify/predict on new data

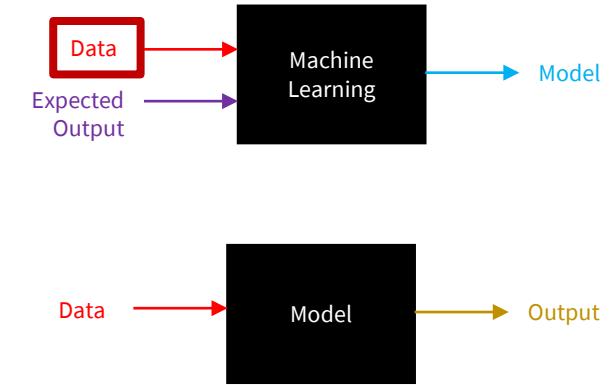
The ML model can be used to breach privacy of that new data (or associated entities)

Privacy problem #1: ML needs data to learn!

Machine learning is based on data to

Find features

Train the model



Data is highly unique! Allows many inferences

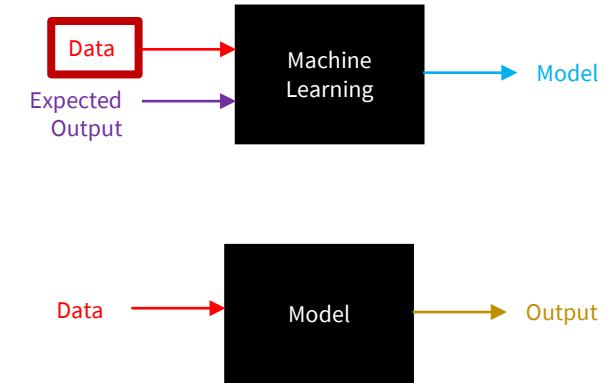
Let's anonymize...

Privacy problem #1: ML needs data to learn!

Machine learning is based on data to

Find features

Train the model



Data is highly unique! Allows many inferences

Let's anonymize...





What about aggregation?

Rationale: aggregating data from many users makes data “non personal”



What about aggregation?

Rationale: aggregating data from many users makes data “non personal”

Problem #1: where / when do you do the aggregation?

To aggregate you need to collect

Aggregate once vs. aggregating on demand

Problem #2: as anonymization, the privacy provided by aggregation depends on the adversary's knowledge

we can learn membership

aggregates can provide more information than the prior

aggregates only protect when there is something to aggregate...

Problem #3: Aggregation impacts utility (more if one wants to eliminate all leakage)

Privacy problem #1: ML needs data to learn!

Machine learning is based on data to

Find features

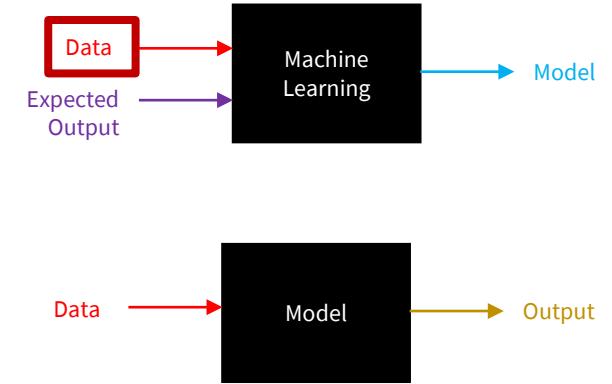
Train the model

Data is highly unique! Allows many inferences

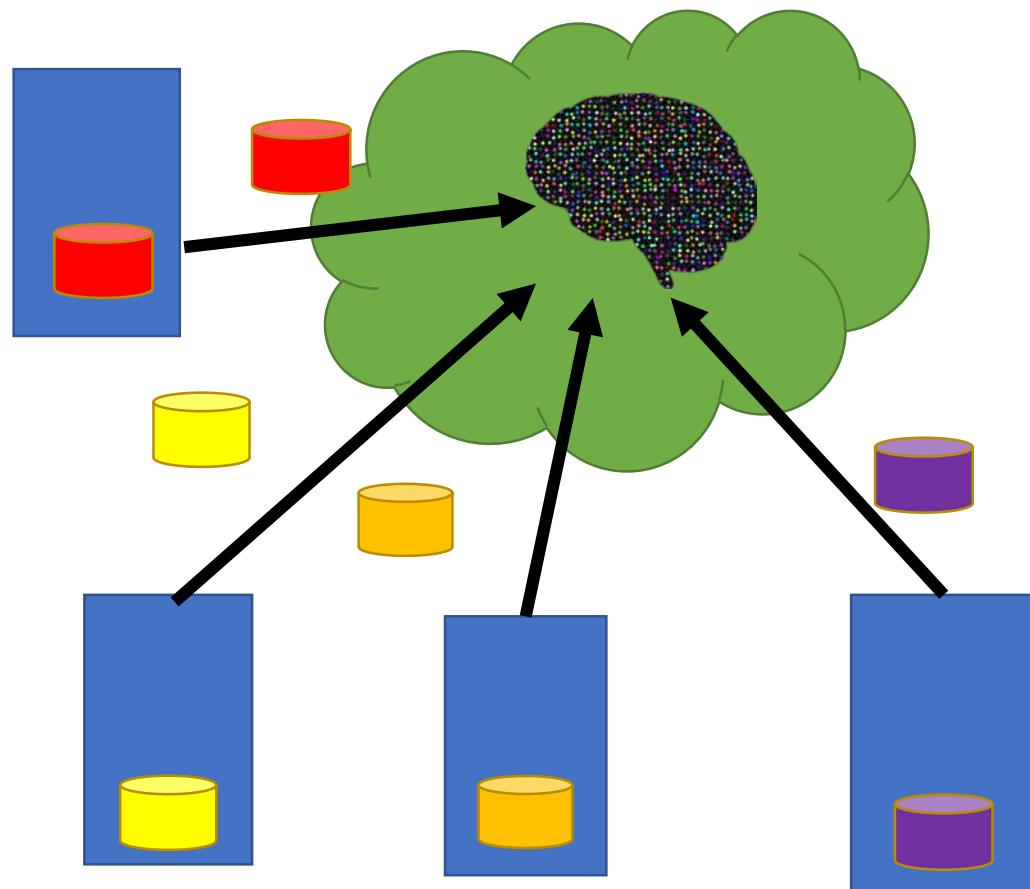
Anonymizing may not work...

Aggregation affects utility and requires careful evaluation

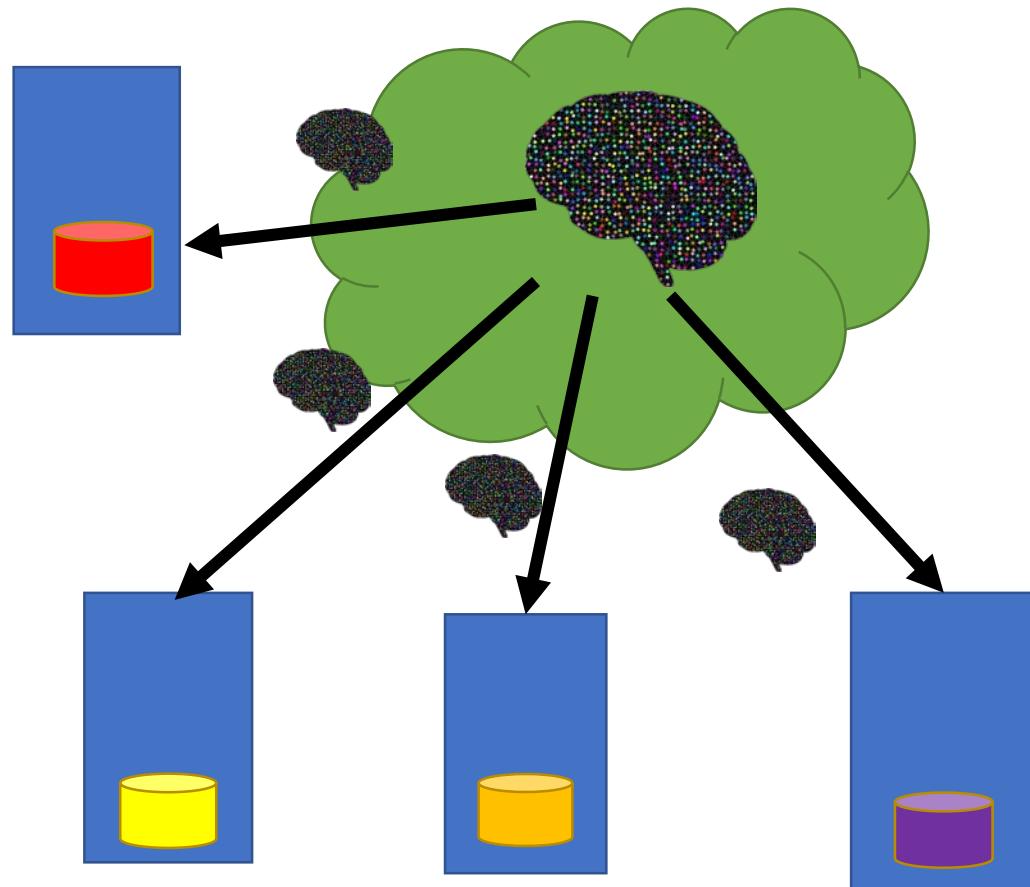
Let's collect noisy data



Learning without privacy



Learning without privacy (optional step)



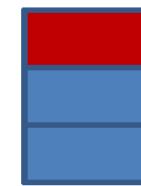
Differential Privacy - Intuition

For every pair of inputs
that differ in one value



D_1 D_2

For every output ...

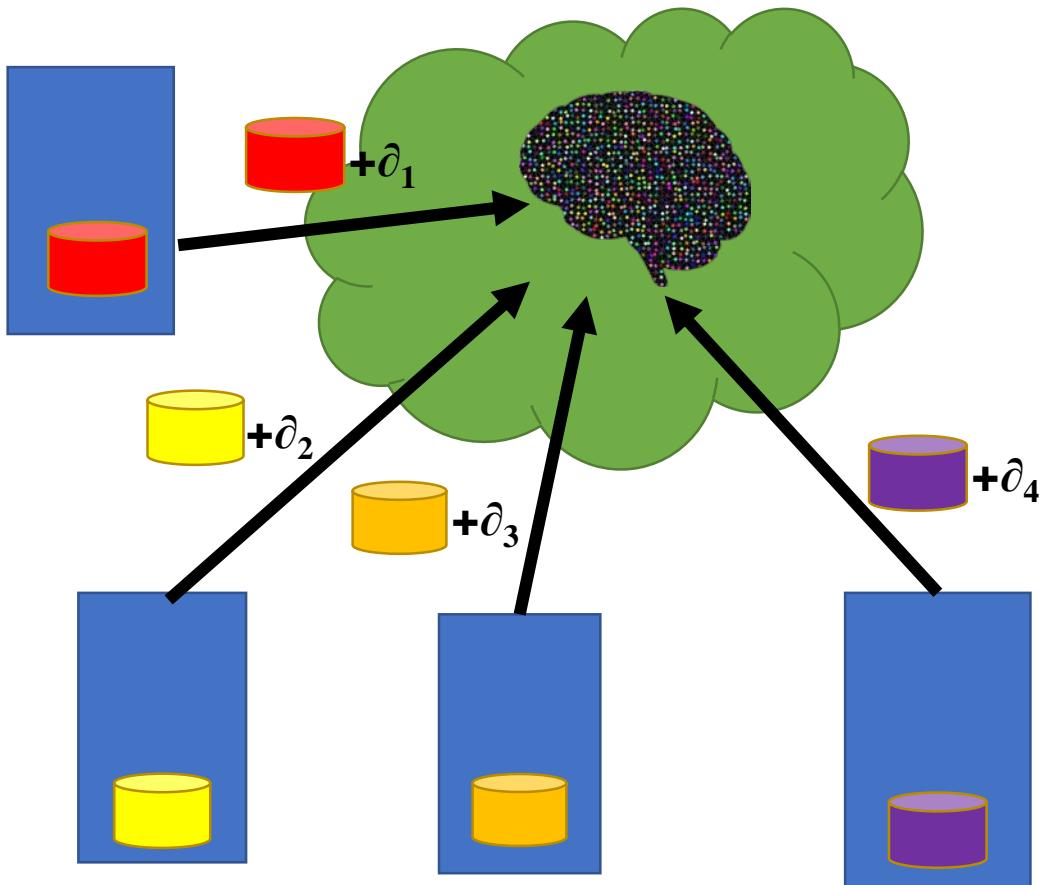


O

Adversary should not be able to distinguish
between any D_1 and D_2 based on any O

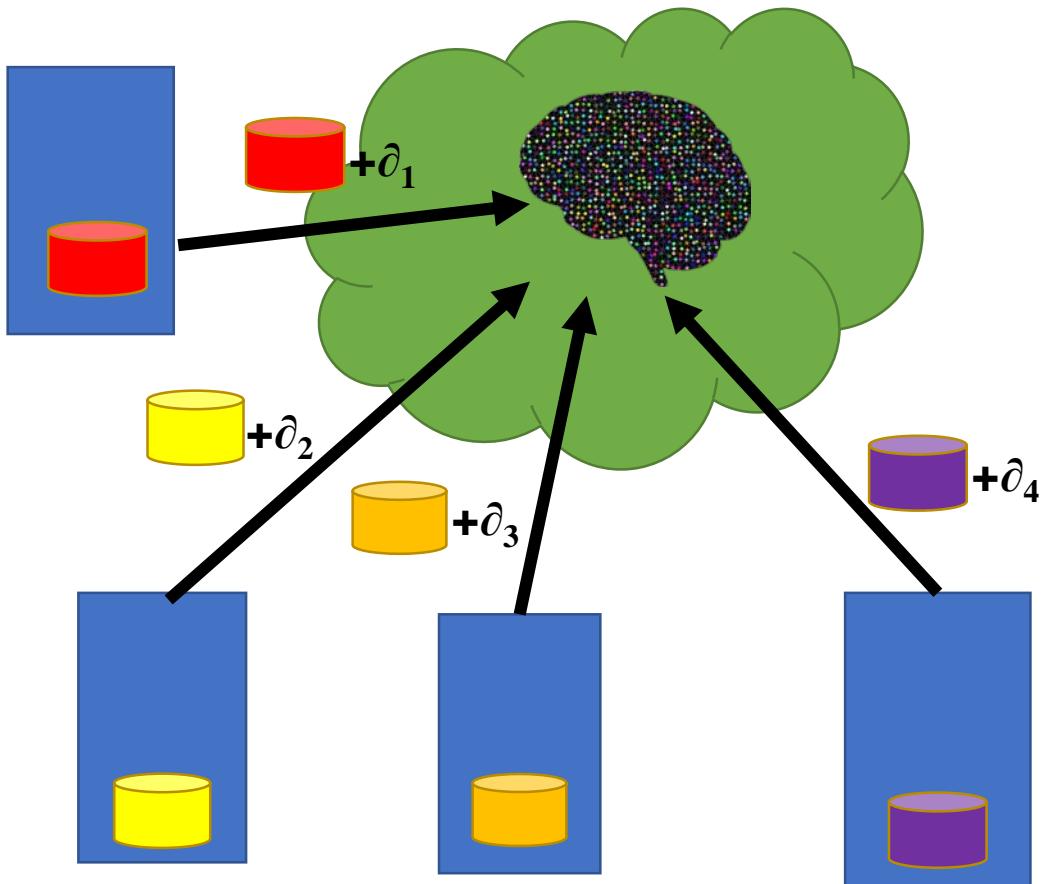
$$\log \left(\frac{\Pr[A(D_1) = O]}{\Pr[A(D_2) = O]} \right) < \epsilon \quad (\epsilon > 0)$$

Learning with privacy



- Instead of sending their data directly, clients send data with Differentially private noise
 - Given the sample, one cannot learn the value
- Enough noise to hide the data but still provide a good model

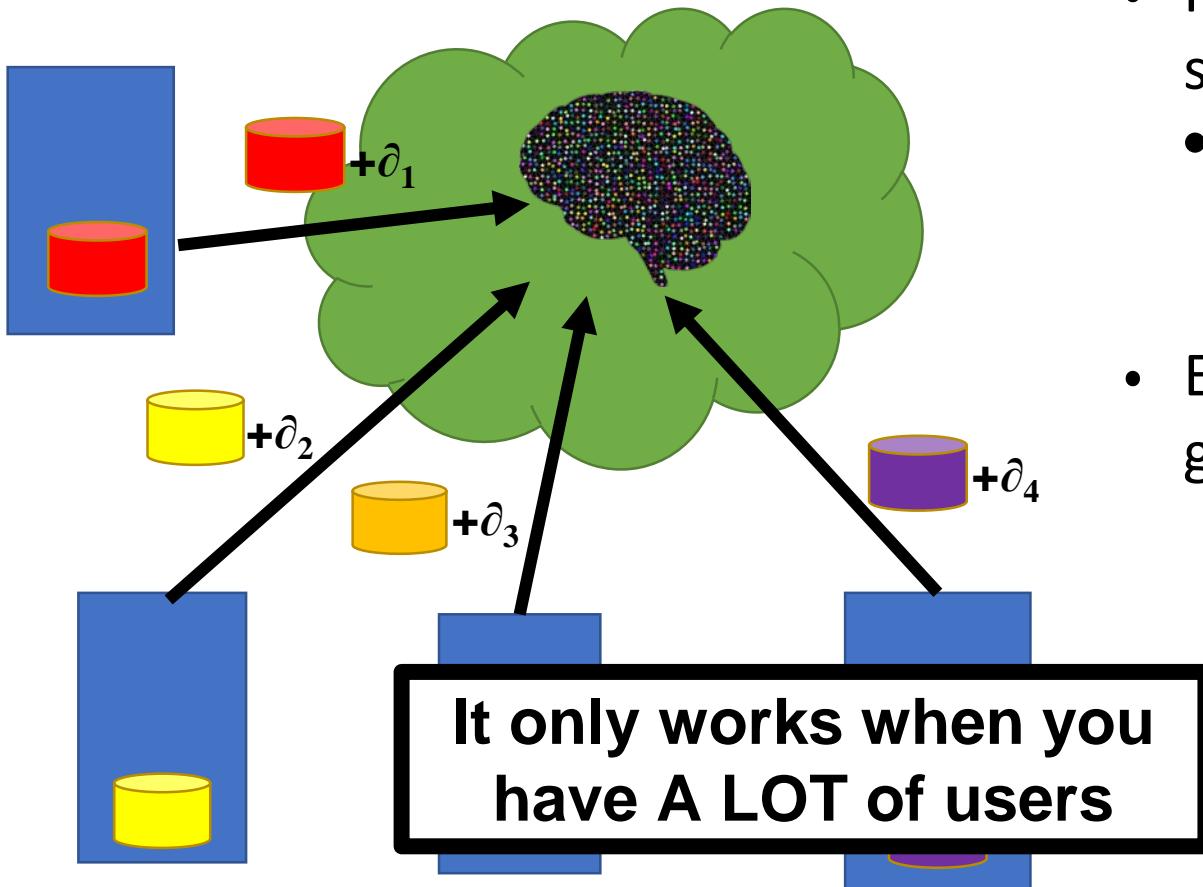
Learning with privacy



- Instead of sending their data directly, clients send data with Differentially private noise
 - Given the sample, one cannot learn the value
- Enough noise to hide the data but still provide a good model

Google RAPPOR - Collect data from phones
Apple - Collect data from phones
Federated learning - Share models
Smart energy - Collect measurements

Learning with privacy

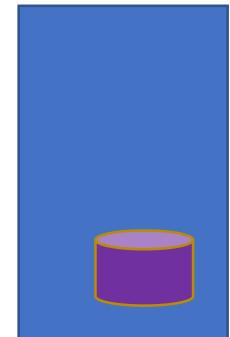
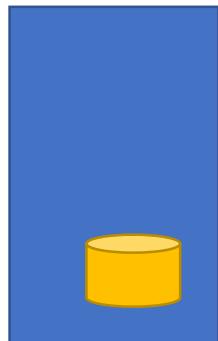
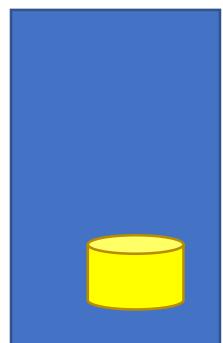
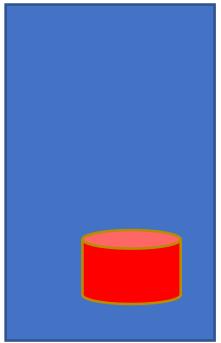


- Instead of sending their data directly, clients send data with Differentially private noise
 - Given the sample, one cannot learn the value
- Enough noise to hide the data but still provide a good model

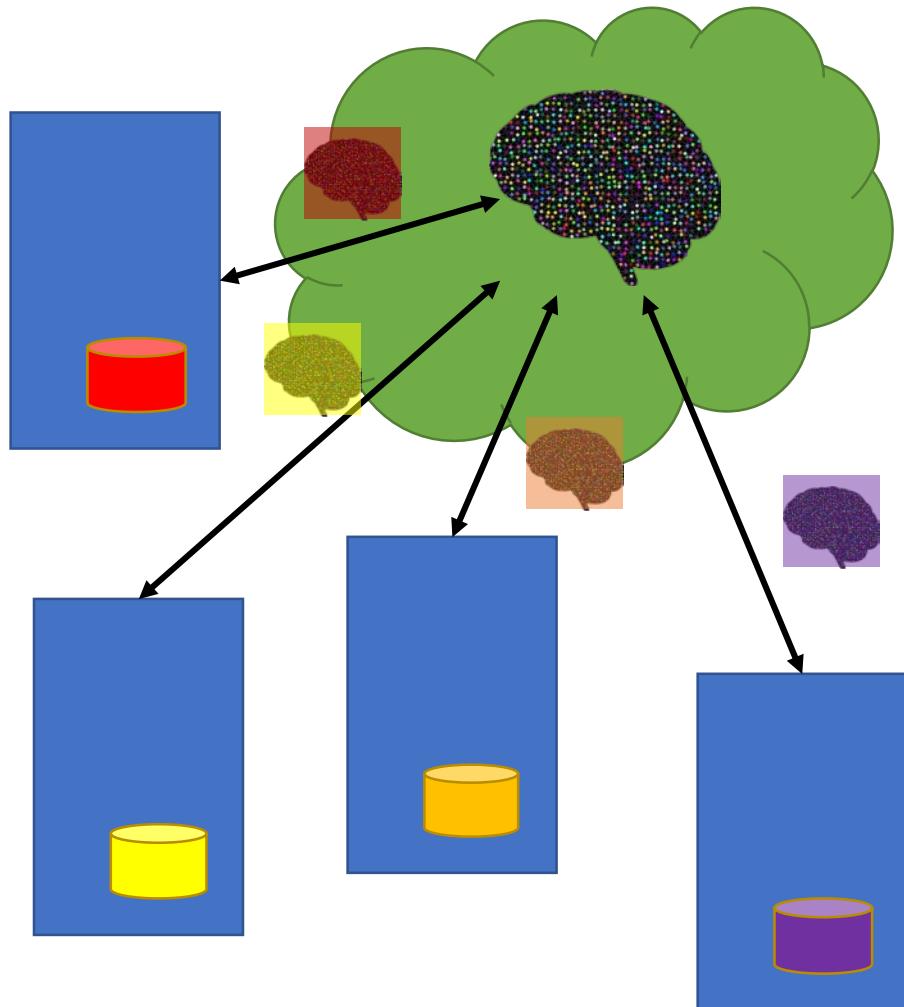
Google RAPPOR - Collect data from phones
Apple - Collect data from phones
Federated learning - Share models
Smart energy - Collect measurements

Another option is to learn locally...

- For ML to be effective models need to be trained on a significant amount of data
- What if you do not have the data??



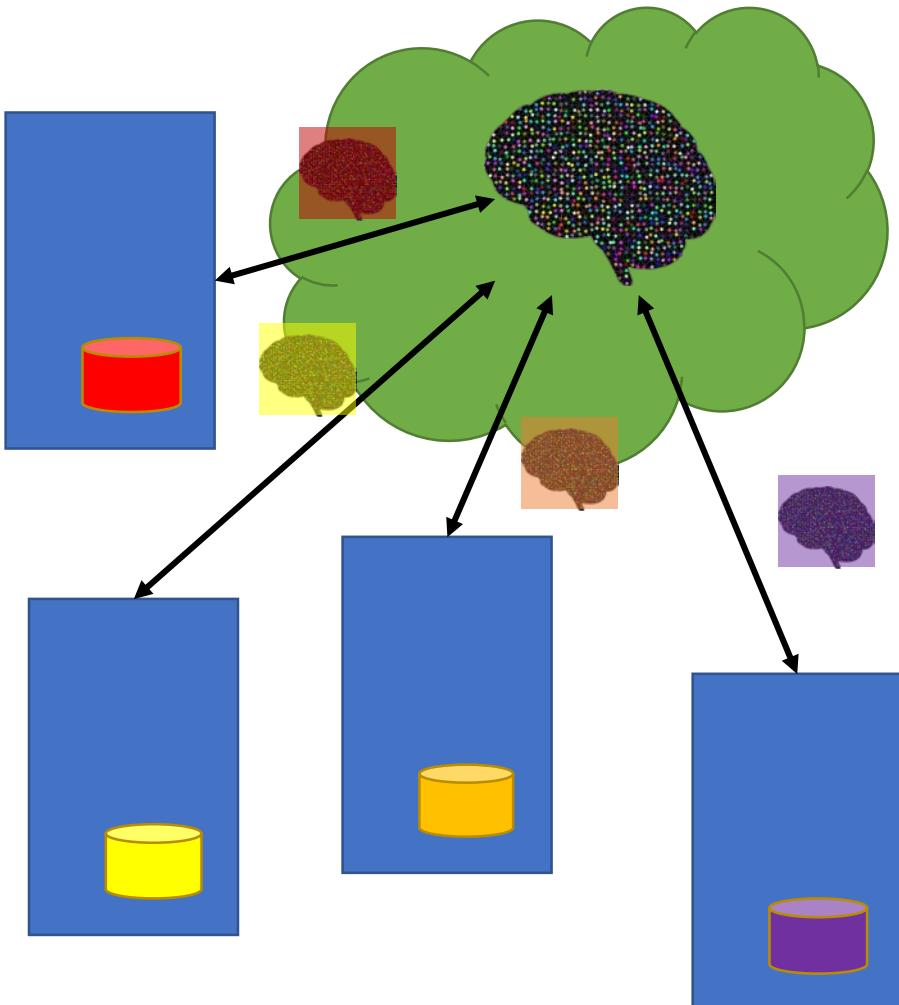
Federated Learning – Learn as a team



Idea: combine data from many small datasets to obtain the model

- Construct some model with weights w .
- Download this model to each client.
- Take some subset of your clients that are online, compute a updated, personalized set of weights *locally*
- Update model with the weighted average of all selected clients

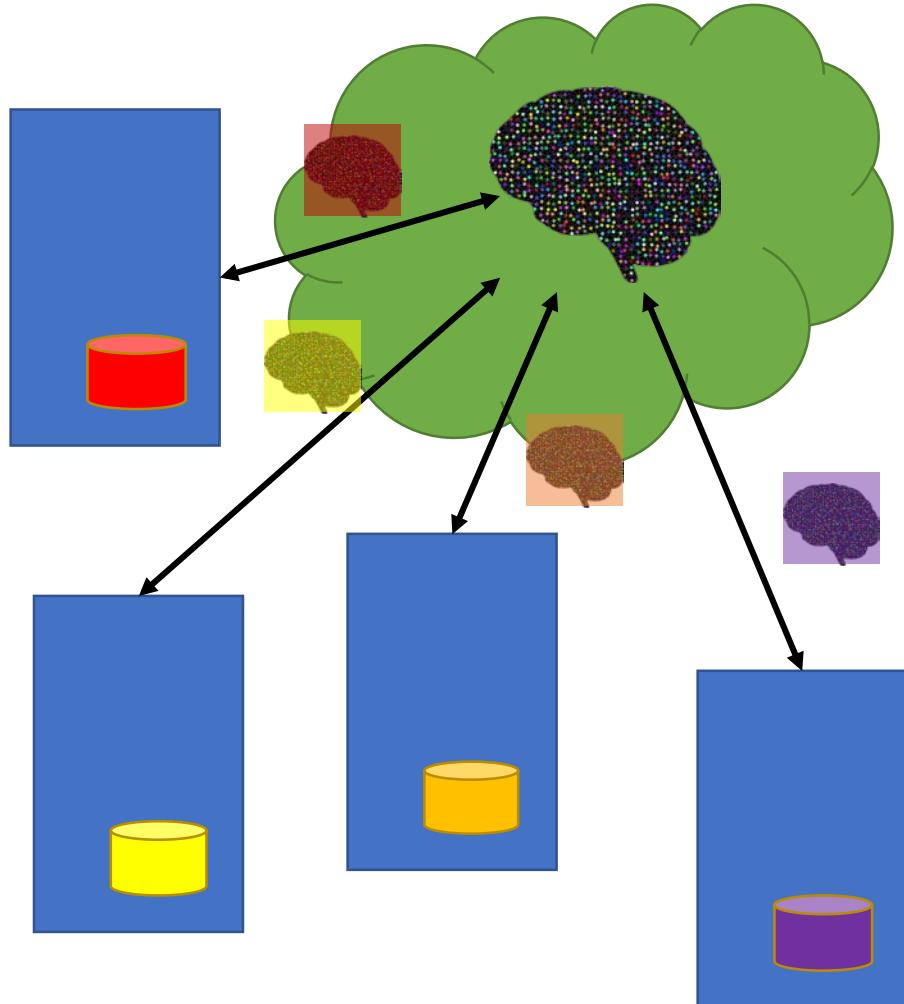
Federated Learning – Privacy problem?



Idea: combine many small datasets to get a large dataset

- Construct some model with weights w .
- Download this model to each client.
- Take some subset of your clients that are online, compute a updated, personalized set of weights *locally*
- Update model with the weighted average of all selected clients

Federated Learning – Privacy problem?

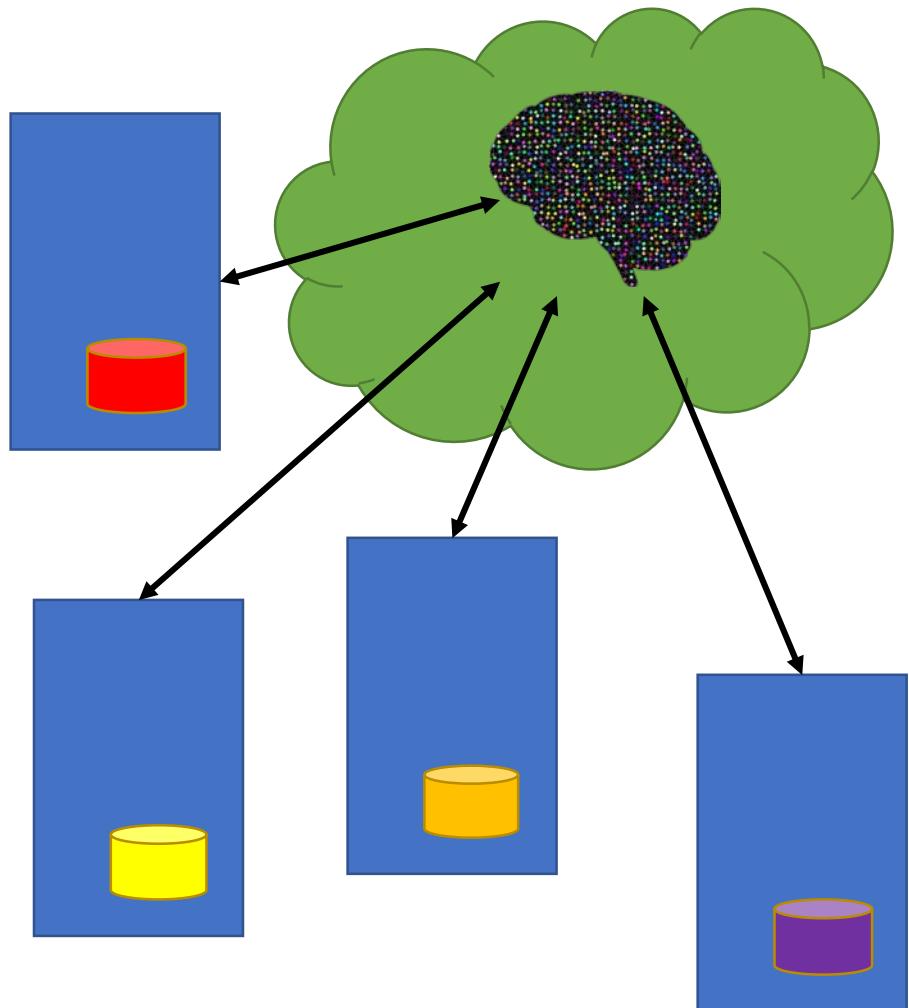


Idea: combine many small datasets to get a large dataset

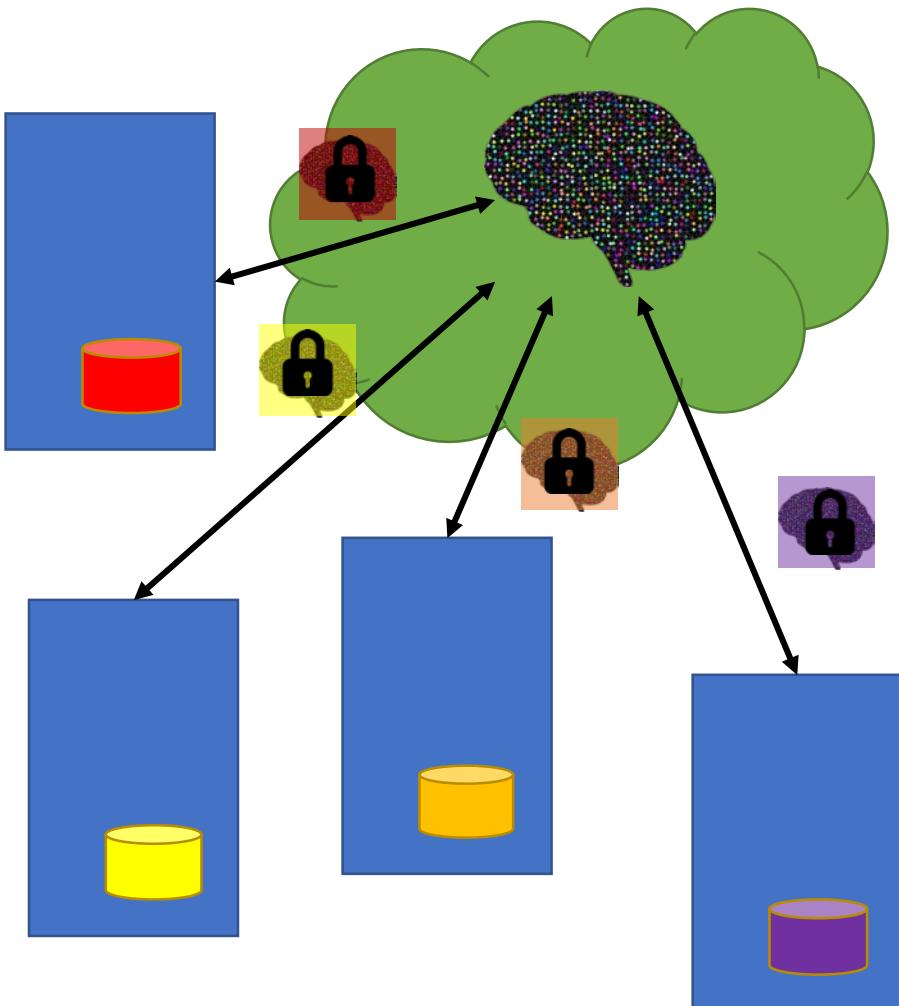
- Construct some model with weights w .
- Download this model to each client.
- Take some subset of your clients that are online, compute a updated, personalized set of weights *locally*
- Update model with the weighted average of all selected clients

**Clients need to reveal their models!
That “reveal” their data!**

Federated Learning – Solutions?



Federated Learning – Solutions?

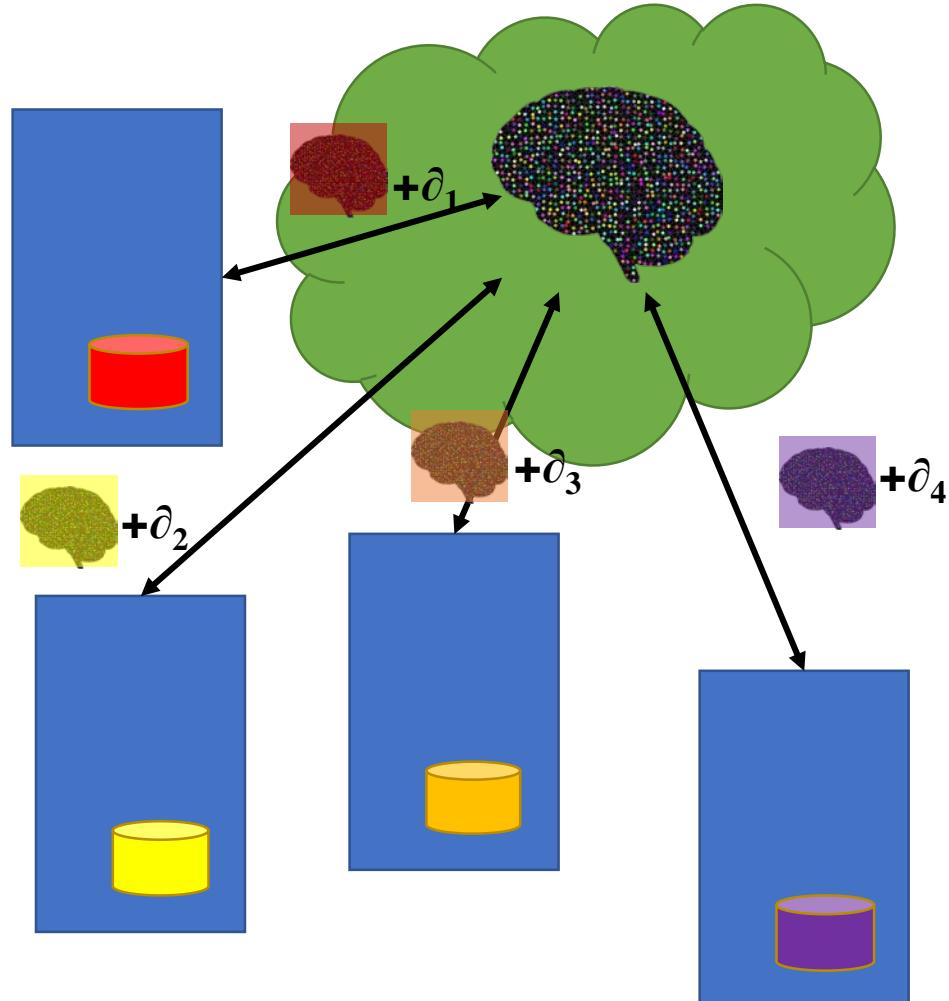


Option 1: Encrypt!

- 1.A:** Homomorphic encryption (risk:)
- 1.B:** Multi party computation

All parties blind their input
Blinding factors cancel

Federated Learning – Solutions?



Option 2: Differential privacy

Before sending models, add noise
Actually, add noise to the gradient

Tradeoff privacy vs functionality

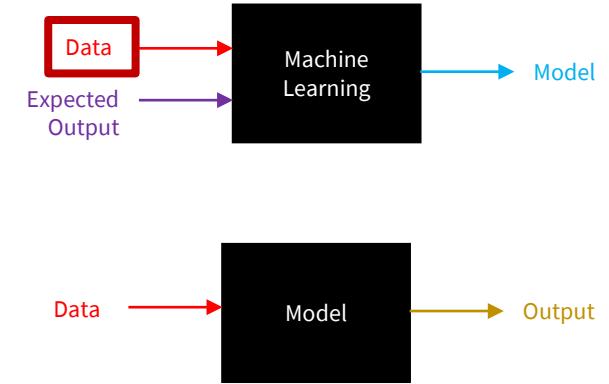
Some attacks still possible

Privacy problem #1: ML needs data to learn!

Machine learning is based on data to

Find features

Train the model



Data is highly unique! Allows many inferences

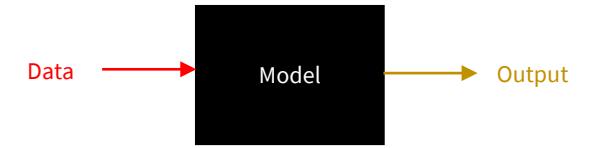
Anonymizing may not work...

Aggregation affects utility and requires careful evaluation

Collecting noisy data is hard if you do not have a billion users...

Local learning may not be useful for many use cases

Privacy problem #2: The model remembers!



Privacy problem #2: The model remembers!

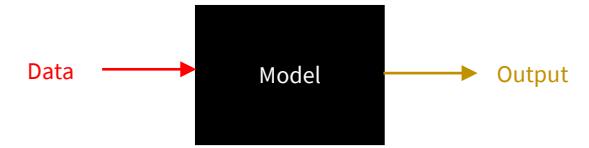


Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

Privacy problem #2: The model remembers!

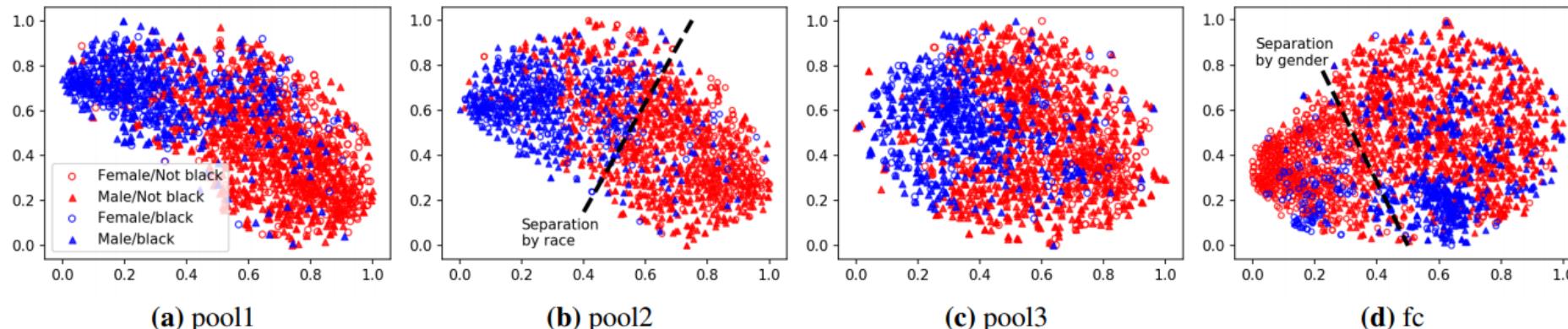
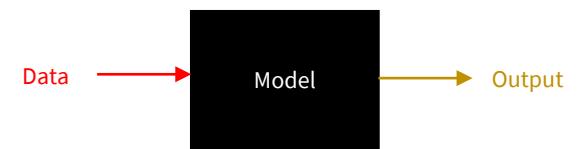
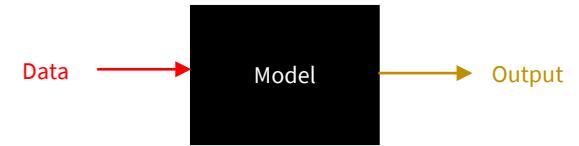


Figure 3: t-SNE projection of the features from different layers of the joint model on LFW gender classification; hollow circle point is female, solid triangle point is male, blue point is the property “race: black” and red point is data without the property.

Privacy problem #2: The model remembers!



| Secret | Log-Perplexity |
|---------------------------------------|----------------|
| The random number is 281265017 | 14.63 |
| The random number is 281265117 | 18.56 |
| The random number is 281265011 | 19.01 |
| The random number is 286265117 | 20.65 |
| The random number is 528126501 | 20.88 |
| The random number is 281266511 | 20.99 |
| The random number is 287265017 | 20.99 |
| The random number is 281265111 | 21.16 |
| The random number is 281265010 | 21.36 |

Table 1: Secrets sorted by Log-Perplexity. The inserted canary — 281265017 — has the lowest log-perplexity. The remaining secrets are all variants with slight modifications (i.e., by changing a few characters).

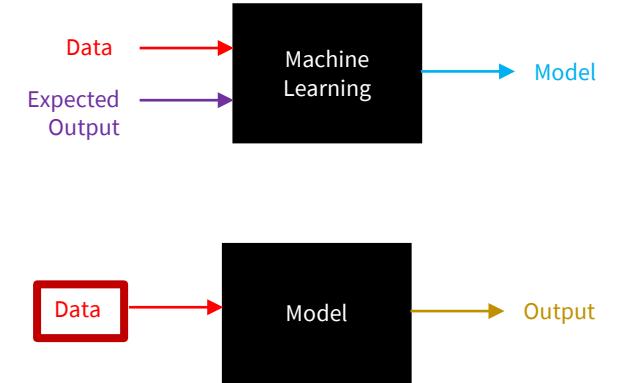
Privacy problem #3: To obtain value You must give data!



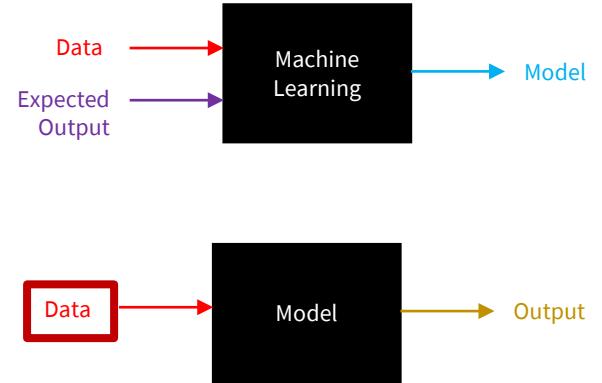
Privacy problem #3: To obtain value You must give data!

To use the model you need to provide a sample...

If the model is remote (ML as a service) you are giving this sample out!



Privacy problem #3: To obtain value You must give data!



To use the model you need to provide a sample...

If the model is remote (ML as a service) you are giving this sample out!

We can do **privacy-preserving model evaluation**: advanced cryptography

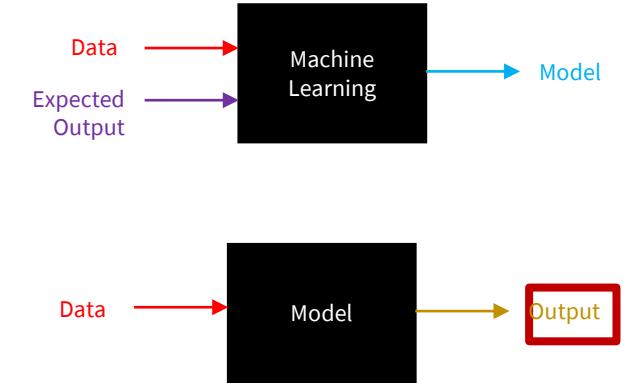
Hard – requires expertise to deploy

Expensive – model evaluation takes much longer

But some simple evaluations are feasible!

Not agile – cryptography is difficult to update

Privacy problem #4: The output reveals information!

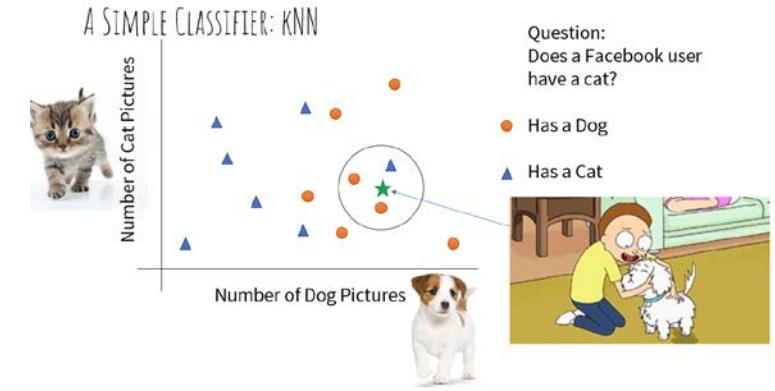
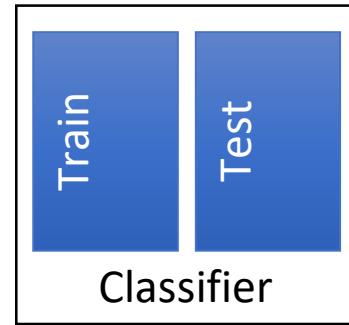


Membership inference

Given the answer of the classifier, infer whether the queried example was used in training.

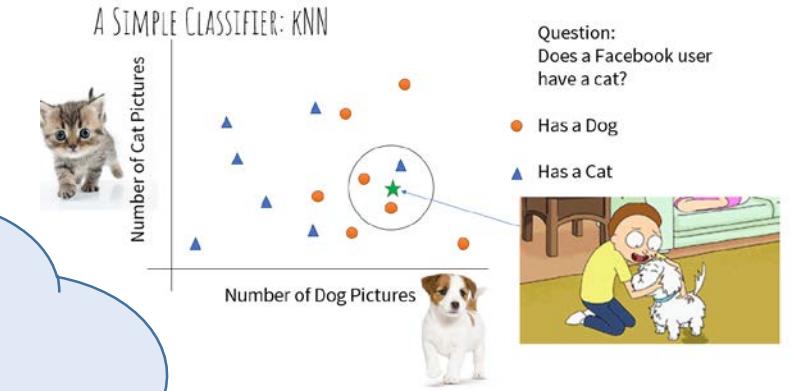
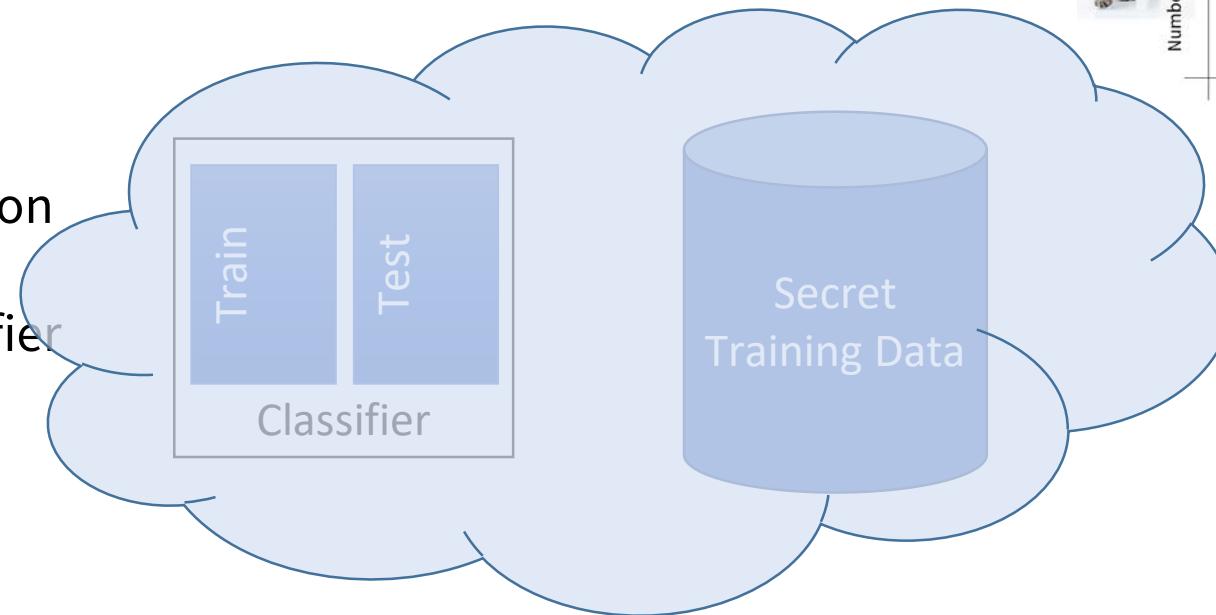
Machine Learning as a Service

1. The Cloud, e.g., Amazon ML or Google Prediction API, (pre-)trains a classifier using their own data



Machine Learning as a Service

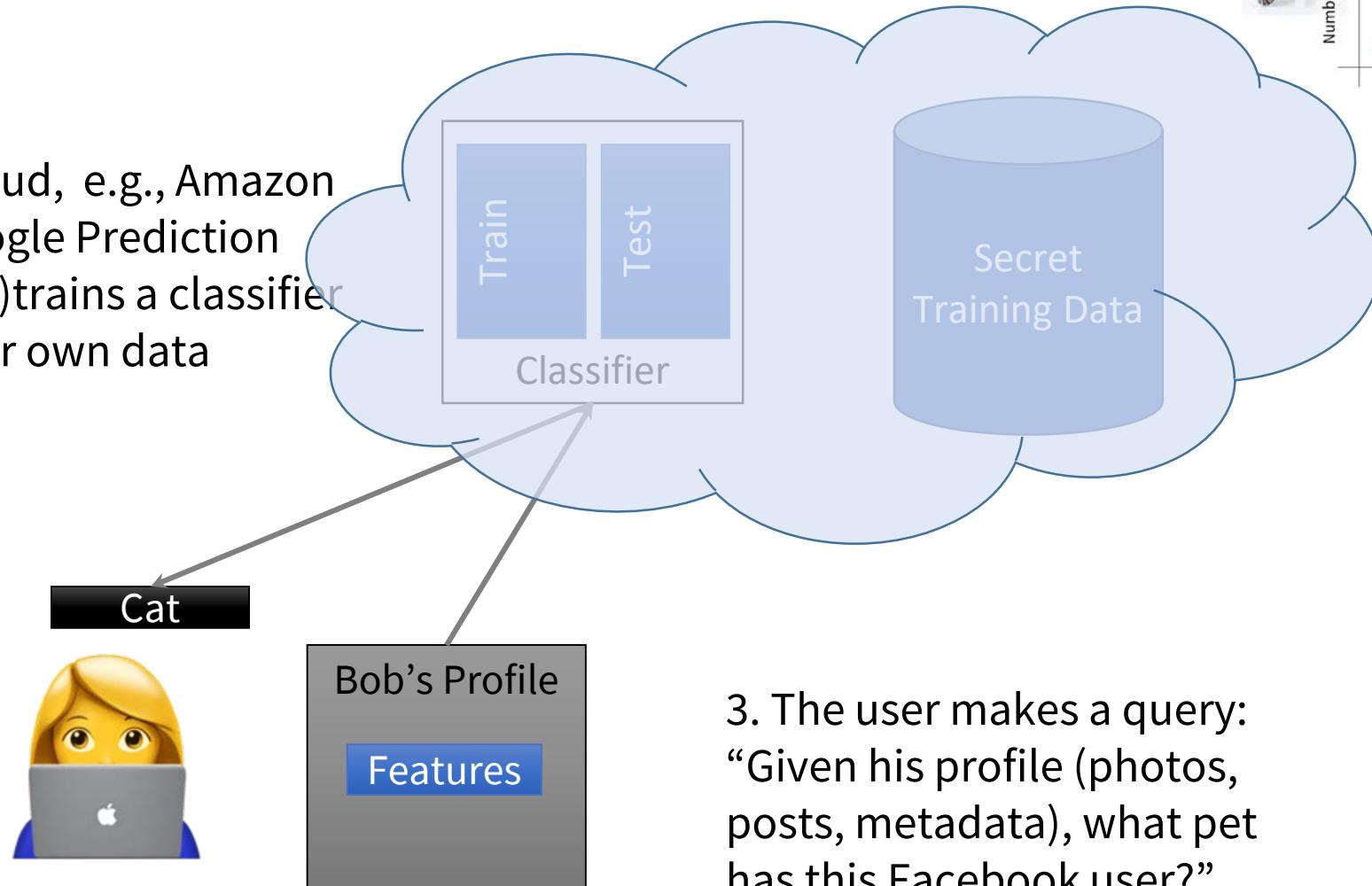
1. The Cloud, e.g., Amazon ML or Google Prediction API, (pre-)trains a classifier using their own data



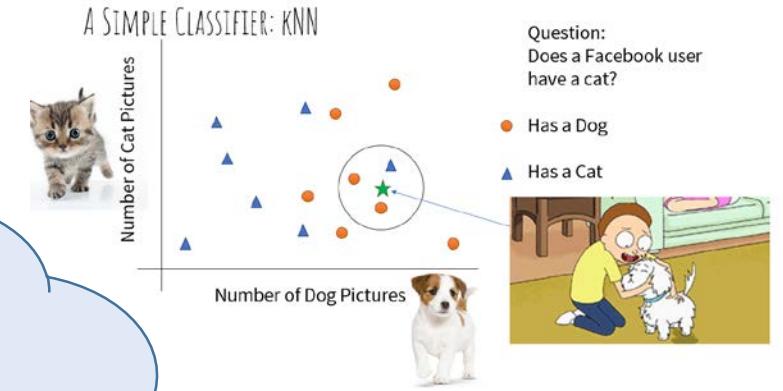
2. Make this classifier available as a service for users to query

Machine Learning as a Service

1. The Cloud, e.g., Amazon ML or Google Prediction API, (pre-)trains a classifier using their own data



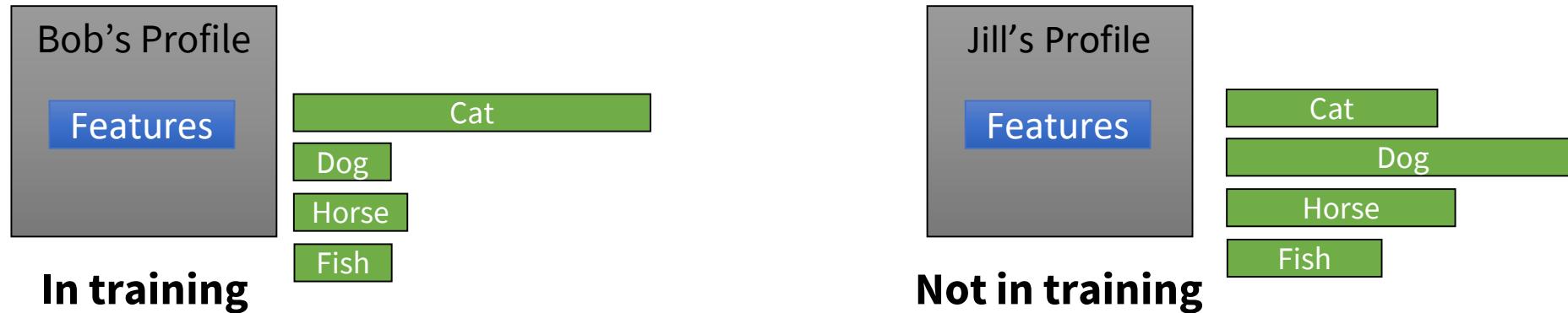
3. The user makes a query:
“Given his profile (photos, posts, metadata), what pet has this Facebook user?”



2. Make this classifier available as a service for users to query

Membership inference

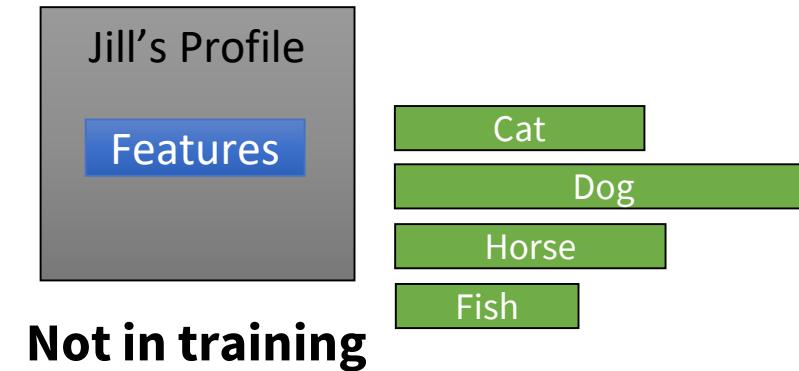
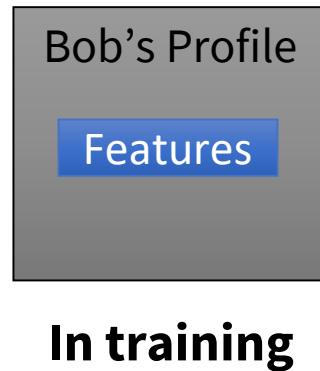
Given the answer of the classifier, infer whether the queried example was used in training.



Throw machine learning to the problem!

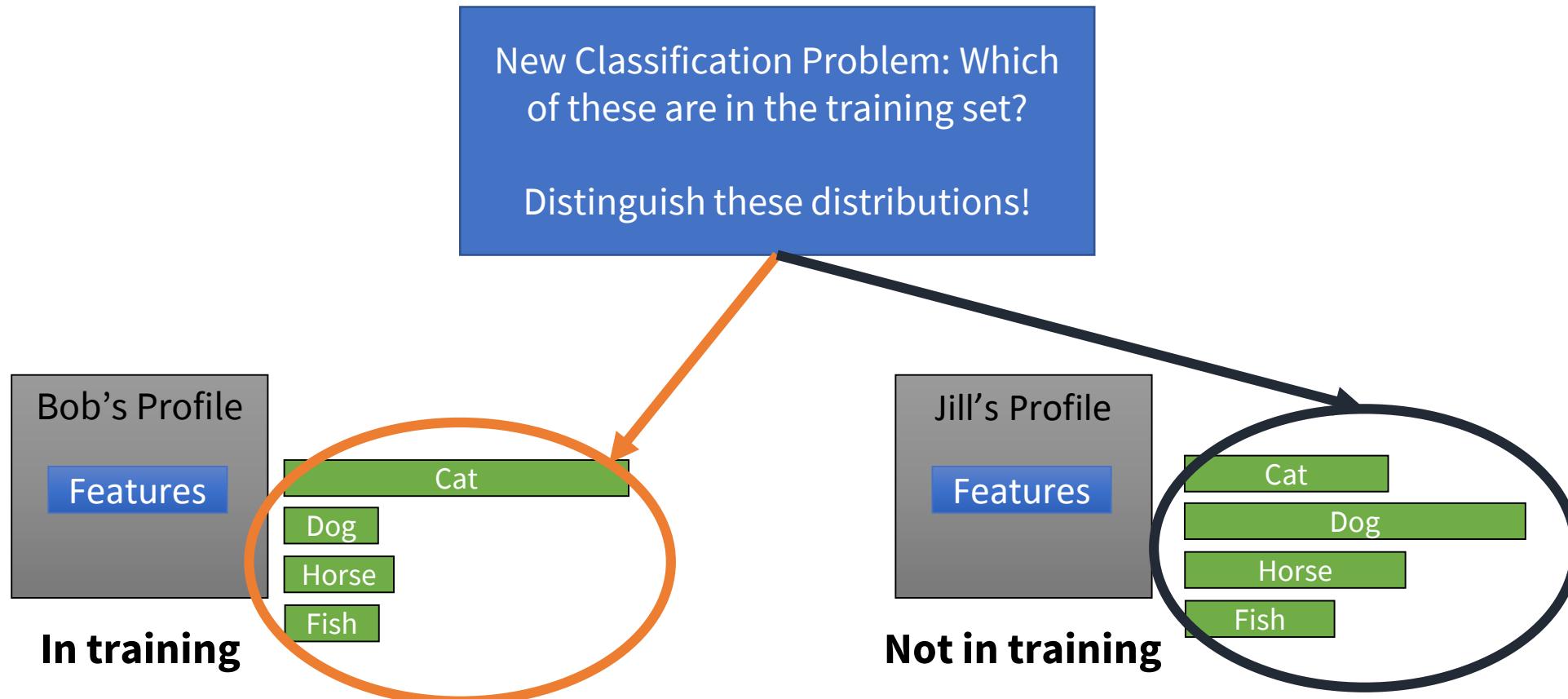
Membership inference

Given the answer of the classifier, infer whether the queried example was used in training.



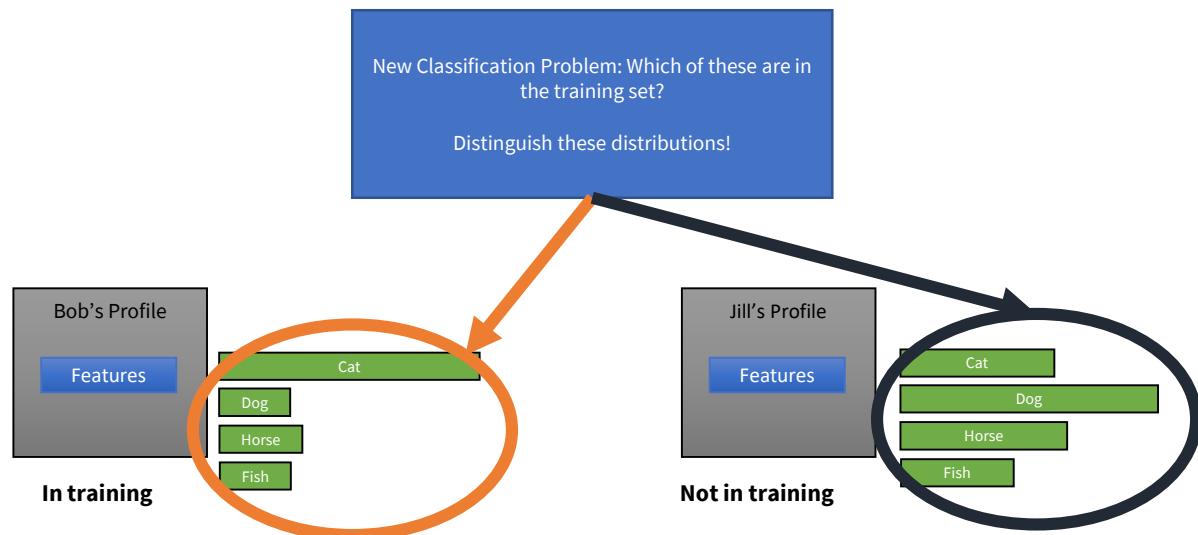
Membership inference

In reality, the distributions are different!



Membership inference

In reality, the distributions are different!



The attack works very well on non-linear models

- Overfitting is sufficient but not necessary
- Works even if the full distribution of outputs is not available
- Works even if one does not try with the target sample!
- Defenses starting to appear

Using membership to learn more

Features

=

| Sex | Age | Country | Has AIDS |
|-----|-----|---------|----------|
| M | 55 | Belgium | yes |

Using membership to learn more

| Sex | Age | Country | Has AIDS |
|-----|-----|---------|----------|
| M | 55 | Belgium | |



- The attacker knows only this information about Bob
- The attacker *also* knows Bob's record was in the training data
- The attack wants to learn Bob's AIDS status

Using membership to learn more



| Sex | Age | Country | Has AIDS |
|-----|-----|---------|----------|
| M | 55 | Belgium | yes |

| Sex | Age | Country | Has AIDS |
|-----|-----|---------|----------|
| M | 55 | Belgium | no |

- The attacker infers membership for both records
- Finds out the AIDS status!

Privacy problem #5: Machine learning is **VERY** good at inferring

Deploy the program!: Use the learnings to classify/predict on new data

The ML model can be used to breach privacy of that new data (or associated entities)



Privacy problem #5: Machine learning is **VERY** good at inferring

Deploy the program!: Use the learnings to classify/predict on new data

The ML model can be used to breach privacy of that new data (or associated entities)

Predictim Claims Its AI Can Flag 'Risky' Babysitters. So I Tried It on the People Who Watch My Kids.

Brief Communication | OPEN | Published: 23 April 2018

Detecting neurodegenerative disorders from web search signals

Ryan W. White, P. Murali Doraiswamy & Eric Horvitz

npj Digital Medicine 1, Article number: 8 (2018) | Download Citation

Abstract

TECH

Facebook Filed A Patent To Predict Your Household's Demographics Based On Family Photos

Facebook's proposed technology would analyze your wifey tags, shared IP addresses, and photos to predict where you live with.

Nicole Nguyen, BuzzFeed News Reporter

Last updated on November 16, 2018, at 2:27 p.m. ET
Posted on November 15, 2018, at 7:04 p.m. ET

Twitter **Facebook** **Copy**

facebook

Privacy

A lighthouse silhouette with a curved arrow pointing to it from the word "Privacy".

Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States

Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei

PNAS December 12, 2017 114 (50) 13108-13113; published ahead of print November 28, 2017
<https://doi.org/10.1073/pnas.1700035114>

Edited by Kenneth W. Wachter, University of California, Berkeley, CA, and approved October 18, 2017 (received for review January 4, 2017)

Article **Figures & SI** **Info & Metrics** **PDF**

Significance

We show that socioeconomic attributes such as income, race, education, and voting patterns can be inferred from cars detected in Google Street View images using deep learning. Our model works by discovering associations between cars and people. For example, if the number of sedans in a city is higher than the number of pickup trucks, that city is likely to vote for a Democrat in the next presidential election (88% chance); if not, then the city is likely to vote for a Republican (92% chance).

Combining AI and Location Intelligence to Predict Market Demand

Cindy Elliott Contributor
Esri Contributor Group

AI can predict your future tweets by looking at your friends' accounts

A new study shows how machine-learning methods could examine your friends' past tweets to accurately predict your future behavior online.

ON 22 January, 2019

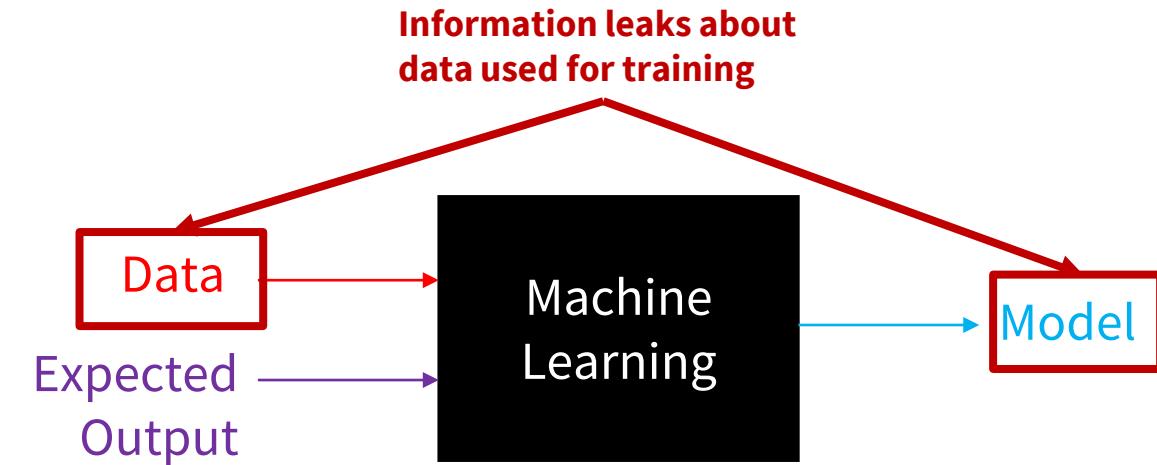
On the Feasibility of Internet-Scale Author Identification

Arvind Narayanan, relax@stanford.edu
Hristo Paskov, hsparkov@stanford.edu
Neil Zhenqiang Gong, neilz.gong@berkeley.edu
John Bethencourt, bethenco@cs.berkeley.edu
Emil Stefanov, emil@berkeley.edu
Eui Chul Richard Shin, ricshin@berkeley.edu
Dawn Song, dawnsong@cs.berkeley.edu

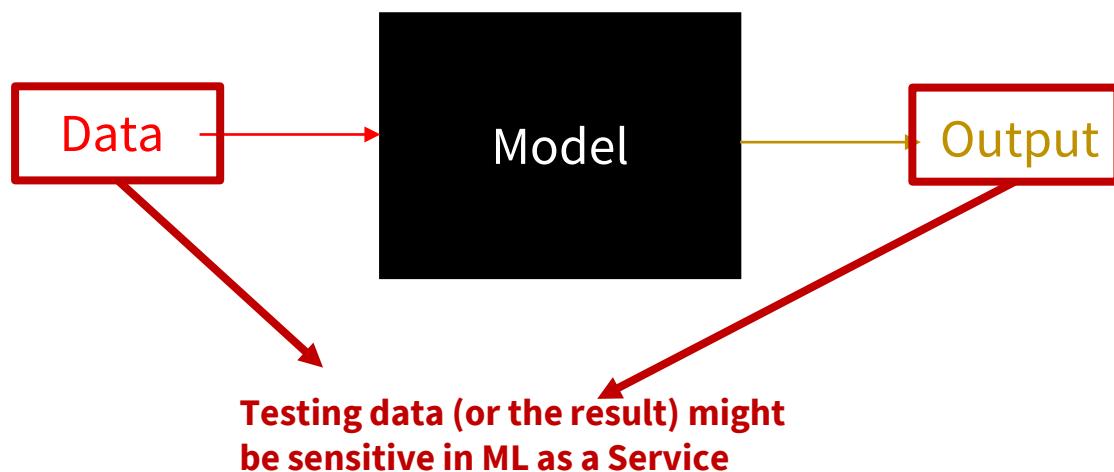
Abstract—We study techniques for identifying an anonymous author via linguistic stylometry, i.e., comparing the writing style against a corpus of texts of known authorship. We experimentally demonstrate the effectiveness of our techniques with as many as 100,000 candidate authors. Given the increasing availability of writing samples online, our result has serious implications for anonymity and free speech — an anonymous blogger or whistleblower may be unmasked unless they take steps to obfuscate their writing style.

While there is a huge body of literature on authorship recognition based on writing style, almost none of it has studied corpora of more than a few hundred authors. The problem becomes qualitatively different at a large scale, as we show,

Takeaways



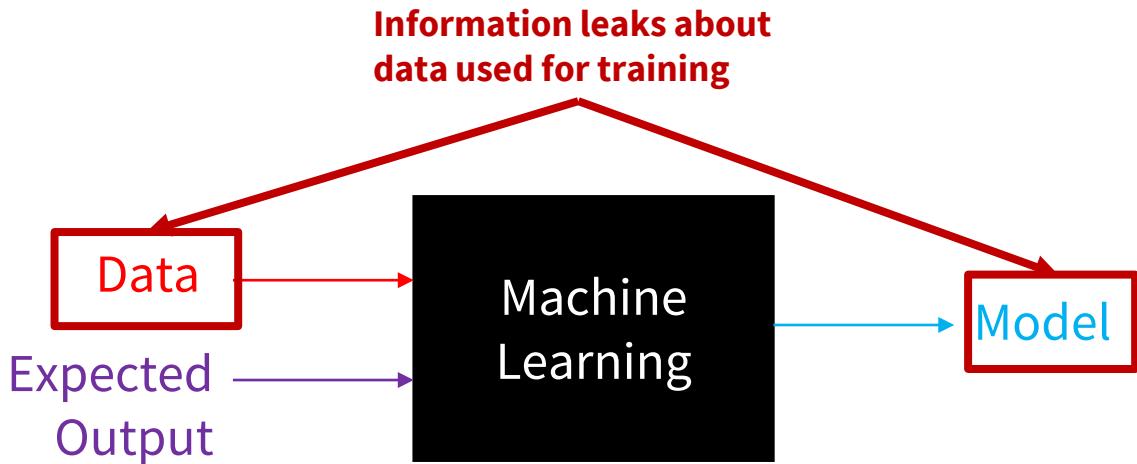
Many problems, some solutions 😊



Deploy the program!: Use the learnings to classify/predict on new data

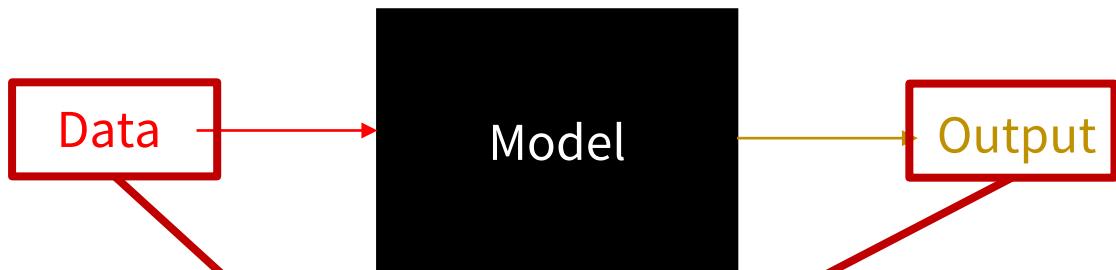
The ML model can be used to breach privacy of that new data (or associated entities)

Takeaways



Many problems, some solutions 😊

Very young field, the situation will improve!



Testing data (or the result) might be sensitive in ML as a Service

Deploy the program!: Use the learnings to classify/predict on new data

The ML model can be used to breach privacy of that new data (or associated entities)

What about other Data Protection issues?

2 – PETs for “institutional” Privacy

CONCERNS - The privacy problem is defined by **Legislation**

Data **should not** be collected without user consent or processed for illegitimate uses

Data **should** be secured: correct, integrity, deletion

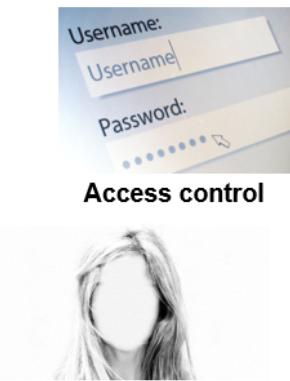


GOALS – Compliance with data protection

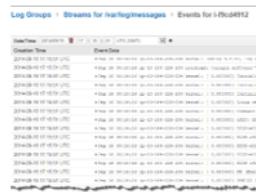
principles

informed consent
purpose limitation
data minimization
subject access rights

Preserving the security of data
Auditability and accountability



Anonymization????



Logging

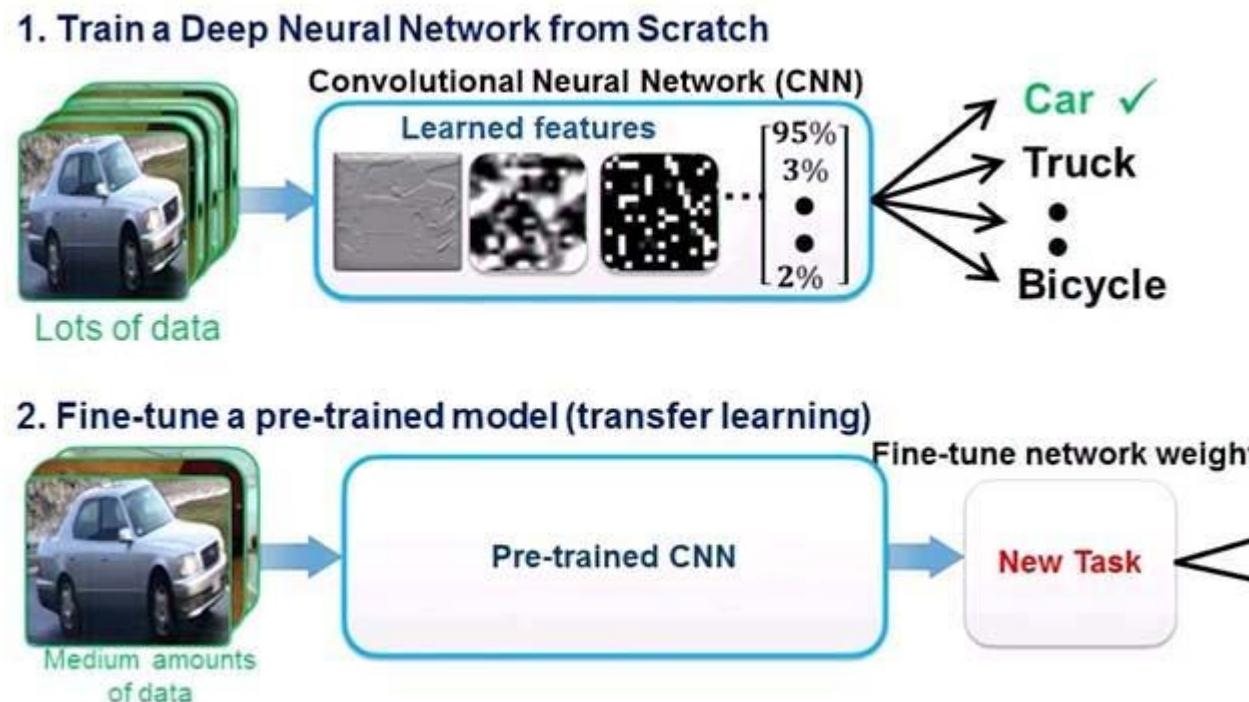
```
Policy(Entity {business_name}: walmart.com, ...
S1[Purpose: (current, contact [opt-in]),
Recipient: (ours),
Retention: (indefinitely),
Data: (#user.login, #user.home-info])
S1[Purpose: (current, develop [opt-in], contact [opt-in]),
Recipient: (ours),
Retention: (stated-purpose),
Data: (#user.name, #user.login, #user.home-info))]
```

Automated Policy
Negotiation

What about other Data Protection issues?

Data subject rights

Is my data there? On the issue of pre-trained models

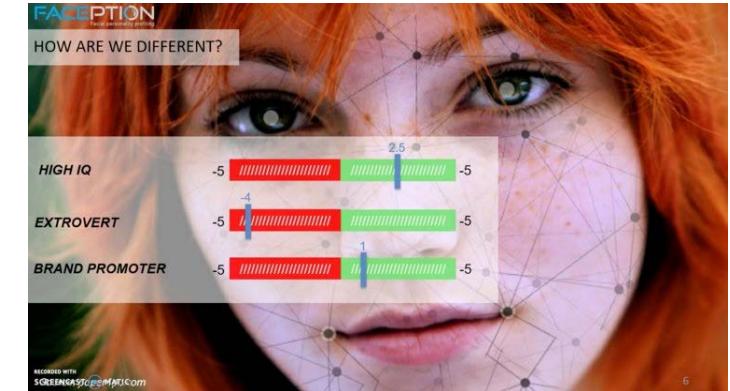


What about other Data Protection issues?

Data subject rights

Is my data there? On the issue of pre-trained models

How is my data processed? Transparency?? Explainability and interpretability are just starting



What about other Data Protection issues?

Data subject rights

Is my data there? On the issue of pre-trained models

How is my data processed? Transparency?? Explainability and interpretability are just starting

Deletion / Right to be forgotten? How to eliminate data from models

Modification? Same, same

What about other Data Protection issues?

Data subject rights

Is my data there? On the issue of pre-trained models

How is my data processed? Transparency?? Explainability and interpretability are just starting

Deletion / Right to be forgotten? How to eliminate data from models

Modification? Same, same

Purpose limitation and data minimization

Can we limit purpose? Algorithms learn much more than intended

How much info is in the data?

We do not know what data is needed! You need ML to learn that



What about other Data Protection issues?

Data subject rights

Is my data there? On the issue of pre-trained models

How is my data processed? Transparency?? Explainability and interpretability are just starting

Deletion / Right to be forgotten? How to eliminate data from models

Modification? Same, same

Purpose limitation and data minimization

Can we limit purpose? Algorithms learn much more than intended

How much info is in the data?

We do not know what data is needed! You need ML to learn that

Consent

Extremely hard to enforce when ML is pervasive. Where / how are you deploying?

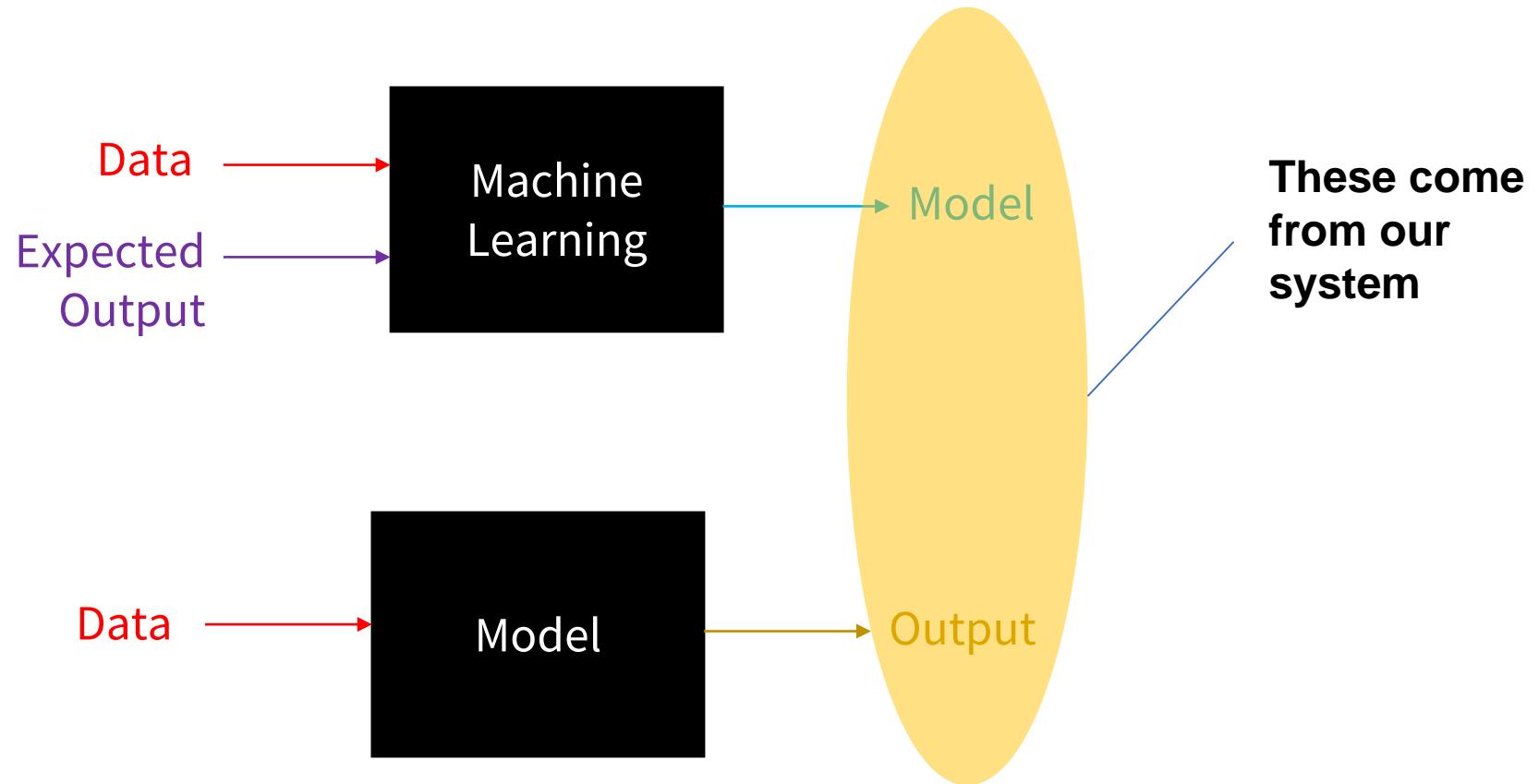
This lecture

- Machine Learning – basics
- Privacy implications of Machine Learning
- **Machine learning under adversarial conditions**
- Issues with applying Machine Learning to Security and Privacy problems

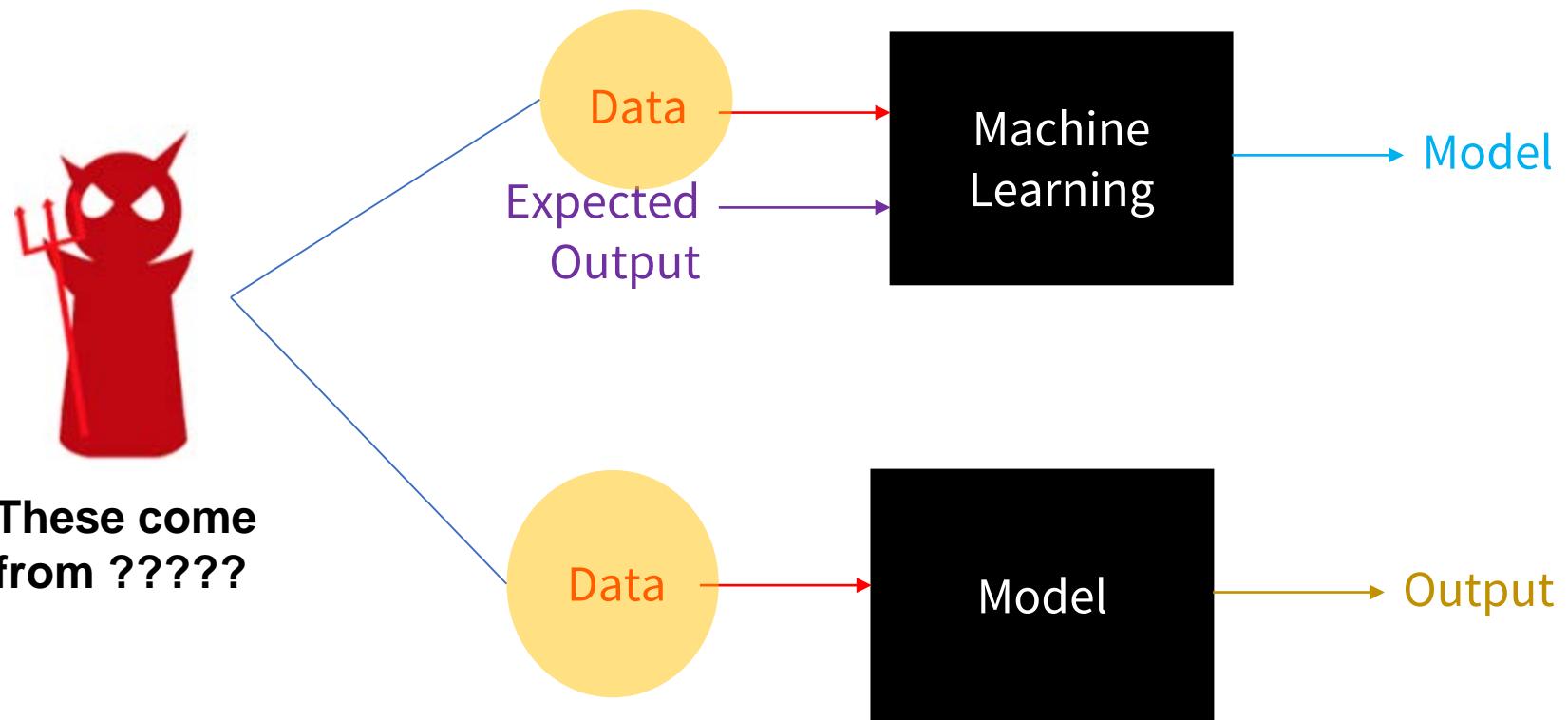


Attacks on machine learning

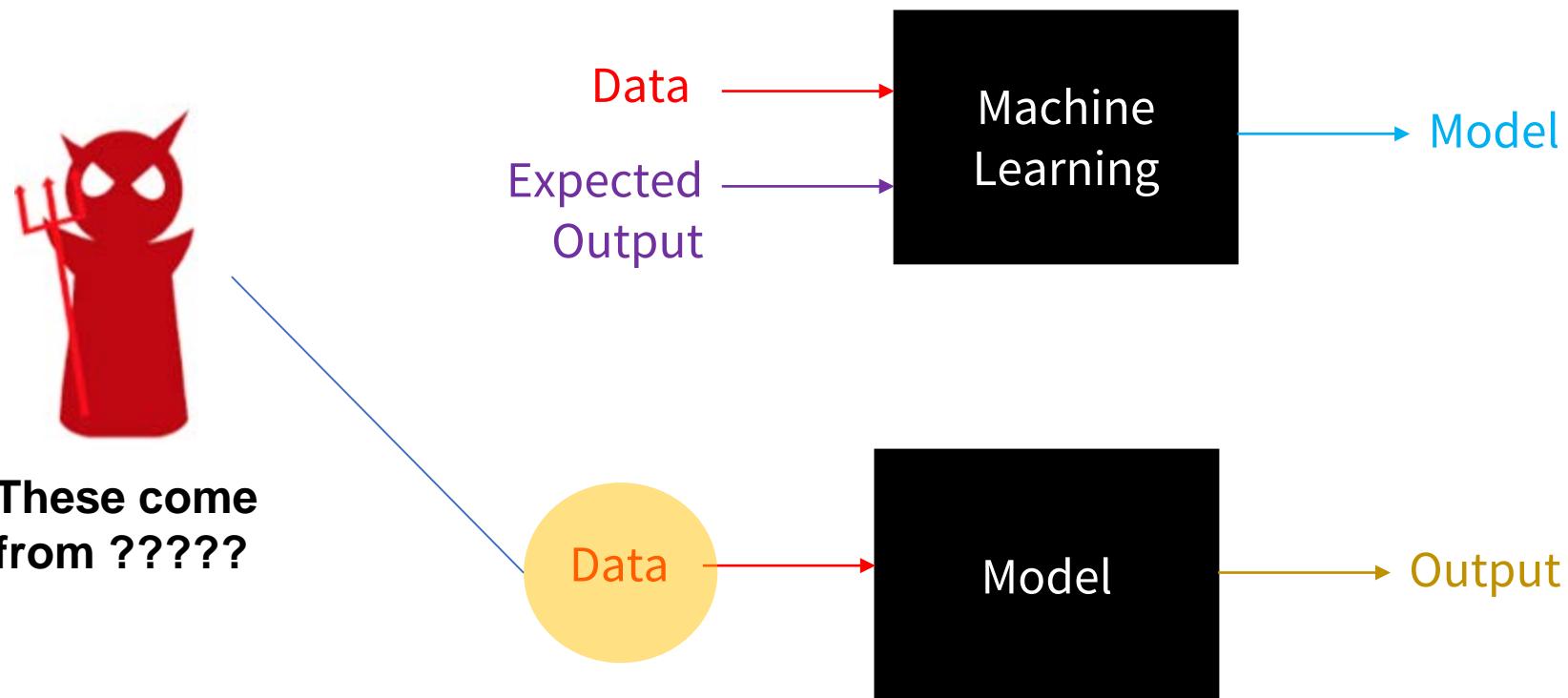
Issues with deploying ML Systems In The Wild



Issues with deploying ML Systems In The Wild



Issues with deploying ML Systems In The Wild

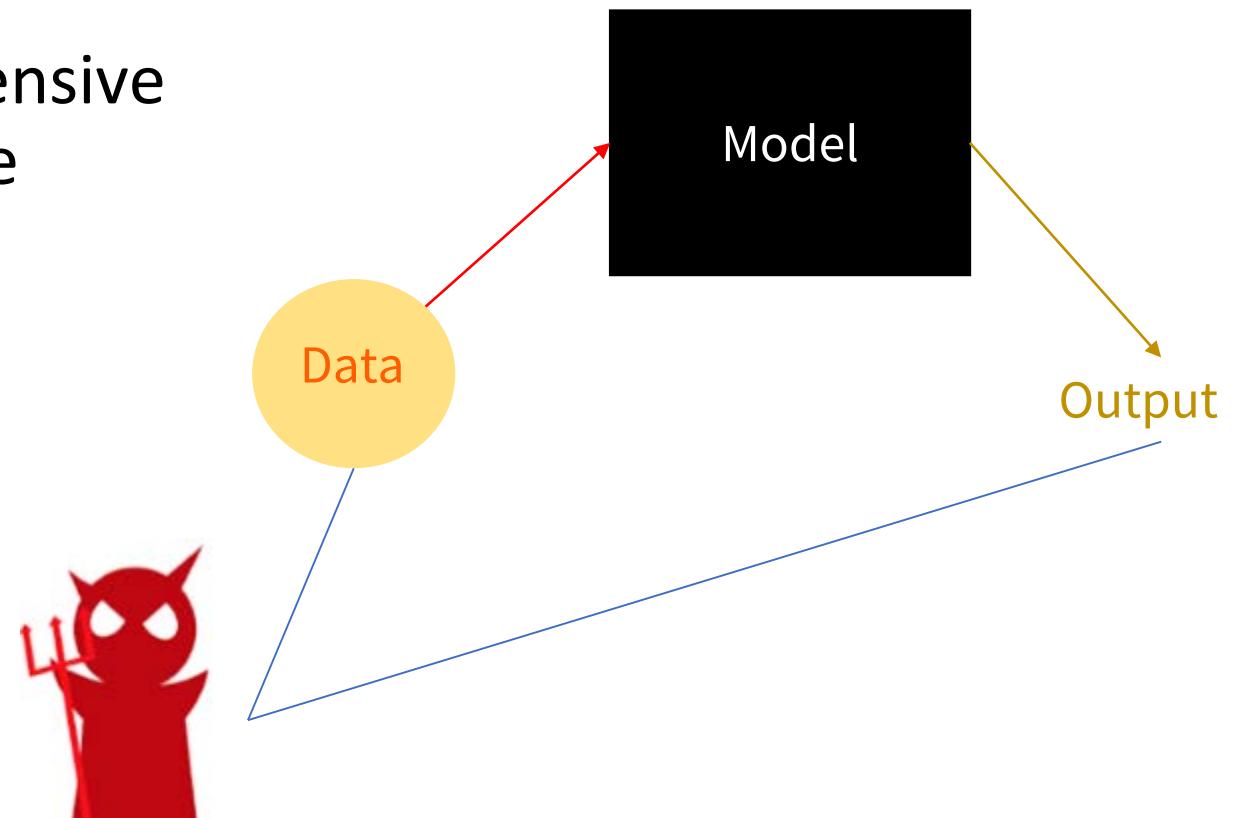


Model stealing

Good ML models require considerable investment:

- Collecting data takes time and money
- Training infrastructure is expensive

If an adversary can query your expensive model many times, and observe the outputs, he can cheaply reproduce it — “steal”.



[1] Tramer et al. Stealing Machine Learning Models via Prediction APIs

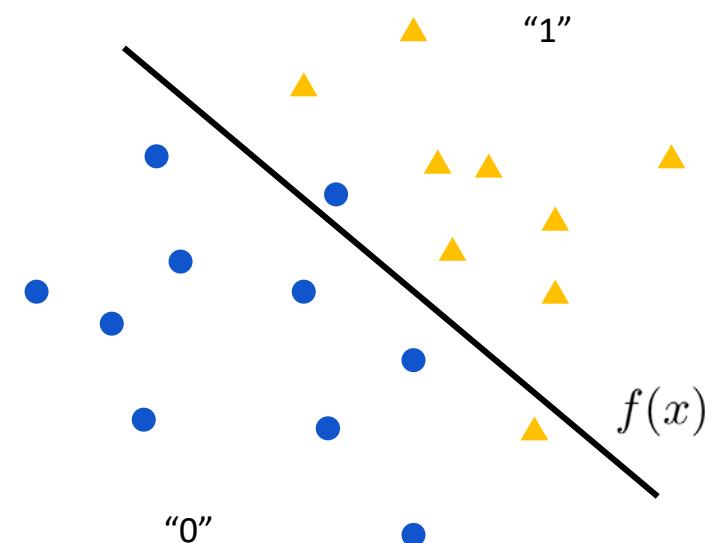
Linear models

When used for classification, a linear model employs a linear function (separating hyperplane) to produce a decision

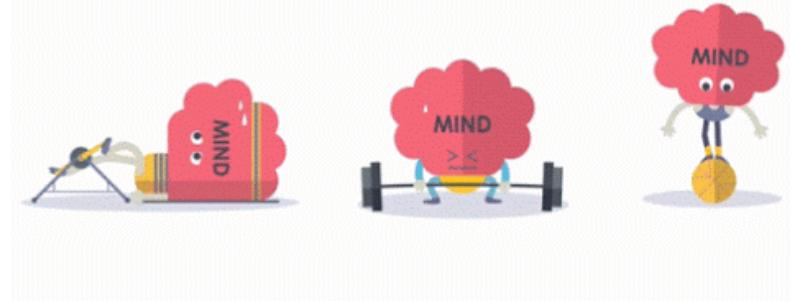
- E.g., logistic regression, SVM with linear kernel

$$f(x) = w \cdot x + b$$

If $f(x) > t$, the output class is “1”, otherwise is “0”.



Stealing a linear model



Assume adversary knows the architecture (knows the model is linear).

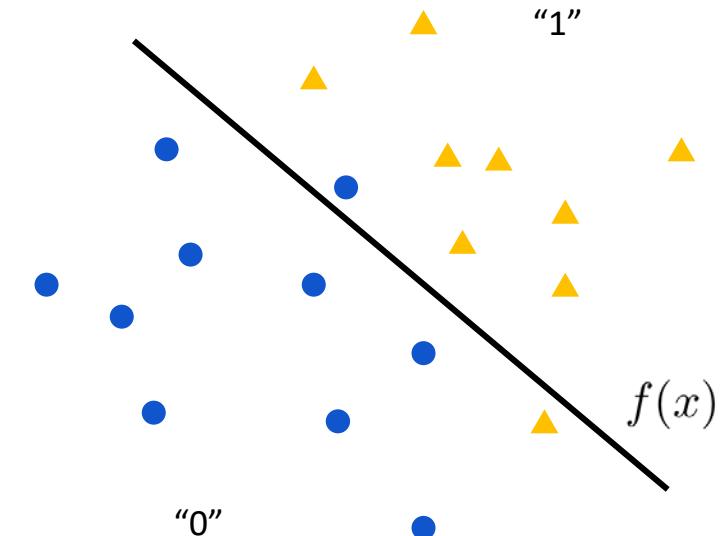
Assume x is **two**-dimensional:

$$f(x) = w_1x_1 + w_2x_2 + b$$

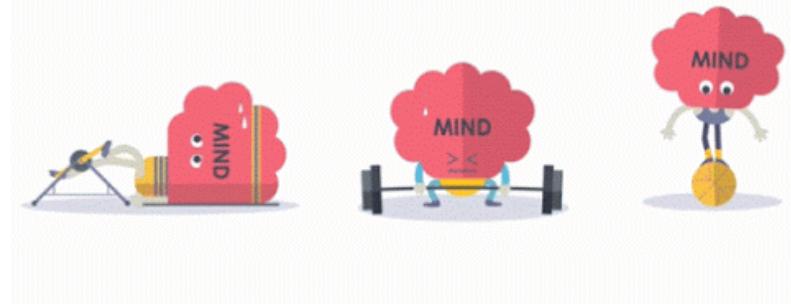
Adversary's goal: steal parameters w, b

How many input-output pairs $(x, f(x))$ the adversary needs to observe to steal the model?

What if the x was d -dimensional?



Equation solving attack

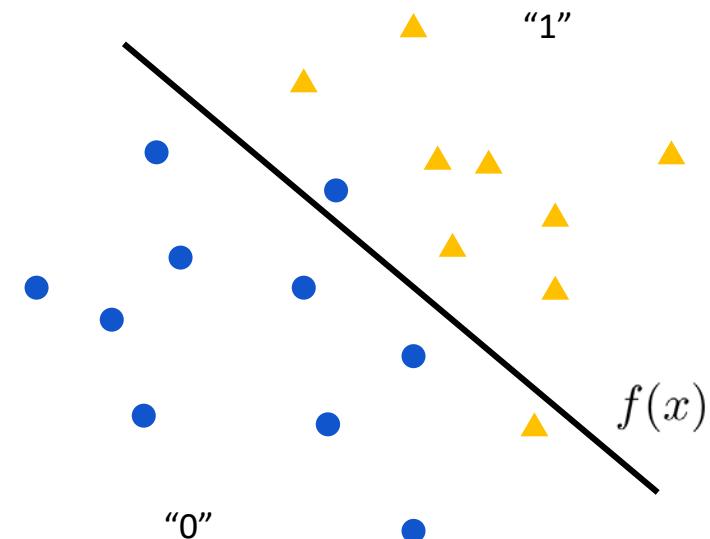


For **two-dimensional** x only need **3 queries!**

In general, if the model uses **d features**, the adversary needs **$d+1$** different queries to steal:

Solve the linear system for w, b

$$w \cdot x^{(i)} + b = f(x^{(i)})$$



Retraining attack

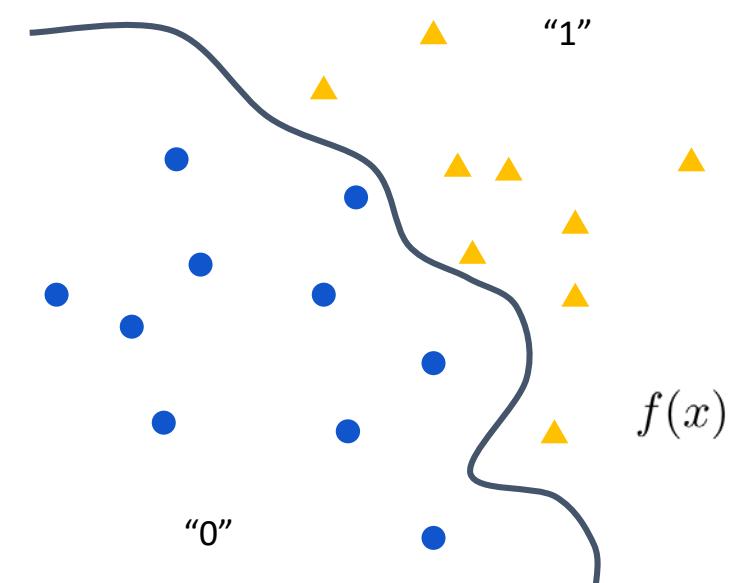
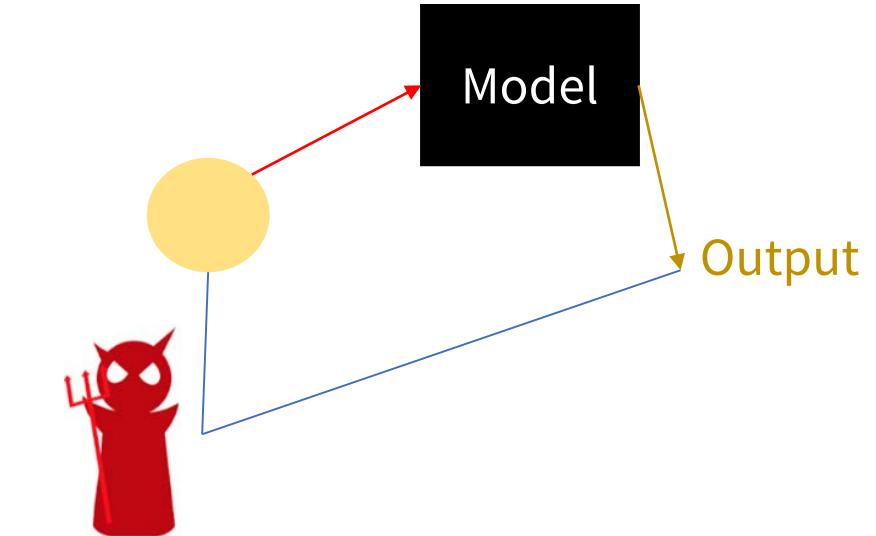
Assume adversary knows the model's architecture:

$$f_w(x)$$

Adversary's goal: steal parameters w

Observe many queries $X = (x, f(x))$, and fit the model on X like on any other training data!

Takes many queries. For a neural network with 2K parameters, need 11K queries to get 99.9% similarity.



[1] Tramer et al. Stealing Machine Learning Models via Prediction APIs

Adversarial Examples

There is no widely accepted definition of the term “Adversarial example”.

The phrase was coined by Ian Goodfellow in 2014 [1] with respect to computer vision problems, but goes back further to work on spam and malware evasion done by Srndic & Laskov [2], Biggio et al. [3], and Dalvi et al. [4].

Working definition:

Inputs to a model that an attacker has designed to cause the model to make a mistake.

[1] Szegedy et al. Intriguing properties of neural networks

[2] Srndic & Laskov. Detection of malicious pdf files based on hierarchical document structure

[3] Biggio et al. Is data clustering in adversarial settings secure?

[4] Dalvi et al. Adversarial classification



$+ .007 \times$



$=$



Panda

Adversarial
Perturbation

Gibbon

IID assumptions no longer hold.

- (1) *Identical*: inputs are intentionally manipulated to not belong to the training distribution.
- (2) *Independence*: inputs are no longer drawn independently; the attacker may sample from a single input repeatedly.

Quick ML refresher I

Objective: find model parameters that *minimize* empirical loss

$$\begin{aligned} w^* &= \arg \min_w \mathbb{E}_{x,y \sim D} [L(x, y; w)] \\ &= \arg \min_w \sum_{x,y \in X} L(x, y; w) \end{aligned}$$

Training data

Data distribution

Loss at (x, y) if model parameters are w

Quick ML refresher II

(simplified)

How? Often, the answer is to use a flavour of gradient descent:

$$w := w - \lambda \nabla_w L(x, y; w)$$

Learning rate
Learning step

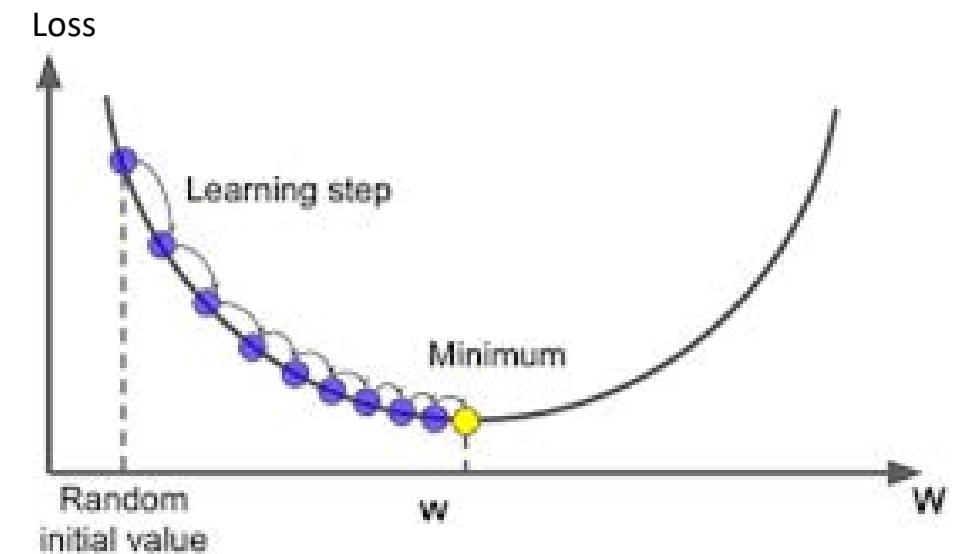


Image: Saugat Bhattacharai

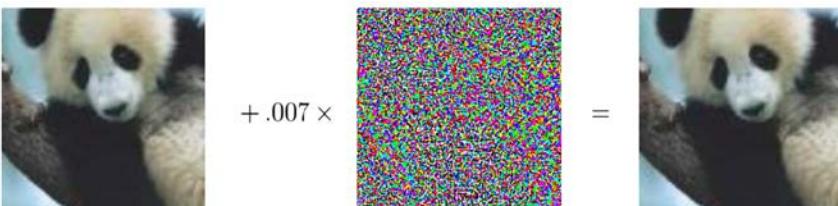
Adversarial example problem

Common objective: find a perturbation that *maximizes* loss



$$\delta^* = \arg \max_{\delta} L(x + \delta, y; w)$$

$$\text{s.t. } x + \delta \approx x$$



Some similarity relation that
usually encodes
“imperceptible change”

(x, y) is the initial
example

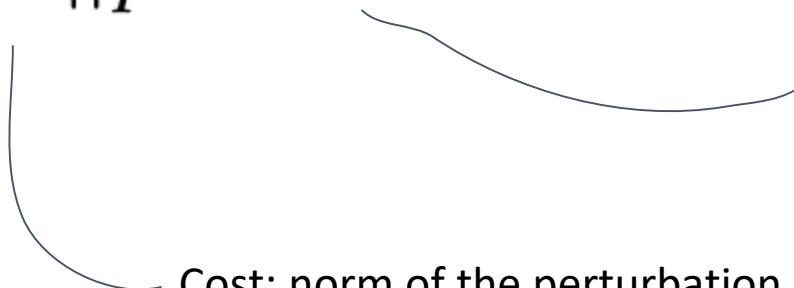
How do we define similarity?

The similarity relation is often represented as *adversarial cost* constraint.

If the goal is to be imperceptible, the common cost is a norm or perturbation:

$$\delta^* = \arg \max_{\delta} L(x + \delta, y; w)$$

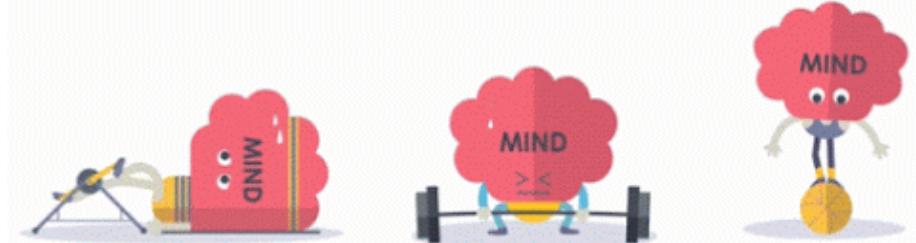
$$\text{s.t. } \|\delta\|_p < \varepsilon$$



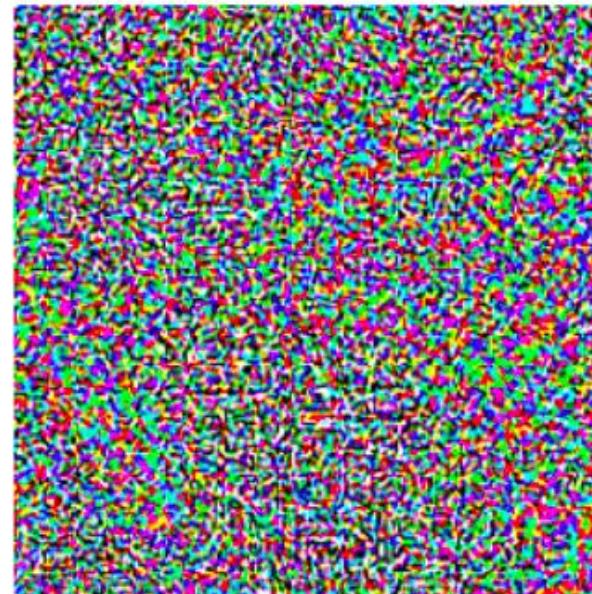
Budget: max allowed norm
of the perturbation. E.g.,
inter-class distance

Cost: norm of the perturbation

Is this a good constraint for imperceptible
adversarial examples?



$+ .007 \times$



=



Panda

δ

Gibbon

Norm (“size”) of the perturbation
must be within epsilon

Other similarity relations

Constraint on perturbation norm assumes that similar images are close in, e.g., Euclidean distance. **This is not always true!**

For example, similarity could be defined as small affine transformations:



Original

Imperceptible
perturbation of
norm 24.3

Semantic change
norm 23.2

- [1] Jacobsen et al., Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness
[2] Engstrom et al., A Rotation and a Translation Suffice: Fooling CNNs with Simple

Attacks do not have to be imperceptible!



A group of orange flowers in a field
Image credit: Richard Leeming @RM_Leeming - CC-BY License



- [1] <http://aiweirdness.com/post/171451900302/do-neural-nets-dream-of-electric-sheep>
- [2] Sharif et al. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition

How to solve the optimization problem?

Recall the adversarial example problem:

$$\delta^* = \arg \max_{\delta} L(x + \delta, y; w)$$

$$\text{s.t. } \|\delta\|_p < \varepsilon$$

Projected gradient descent attack

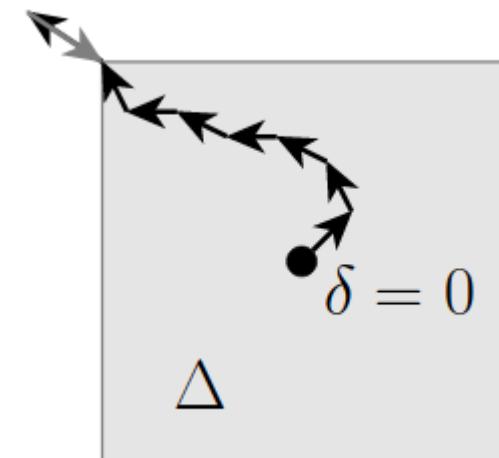
Take a step in the direction of maximum loss, project to meet constraint. Run for several dozens iterations from a random starting delta:

$$\delta := \text{proj}[\delta + \lambda \text{ sign } (\nabla_{\delta} L(x + \delta, y; w))]$$

If outside of the **similarity** region,
project back inside

Direction of the
gradient

Gradient w.r.t. **input**, not
parameters!



Defending against adversarial examples?

Recall the machine learning training objective:

$$\begin{aligned} w^* &= \arg \min_w \mathbb{E}_{x,y \sim D}[L(x, y; w)] \\ &= \arg \min_w \sum_{x,y \in X} L(x, y; w) \end{aligned}$$

Defending against adversarial examples?

Defending in general is very hard. Can only **defend against a particular threat model** (e.g., perturbations up to epsilon norm), and normally no guarantees.

Standard way is *adversarial training* (based on *robust optimization*). It means training on simulated adversarial examples:

$$w^* = \min_w \sum_{x,y \in X} \max_{\delta} L(x + \delta, y; w)$$

e.g.
20 random
restarts,
100 steps

Adversarial example

“threat
model”

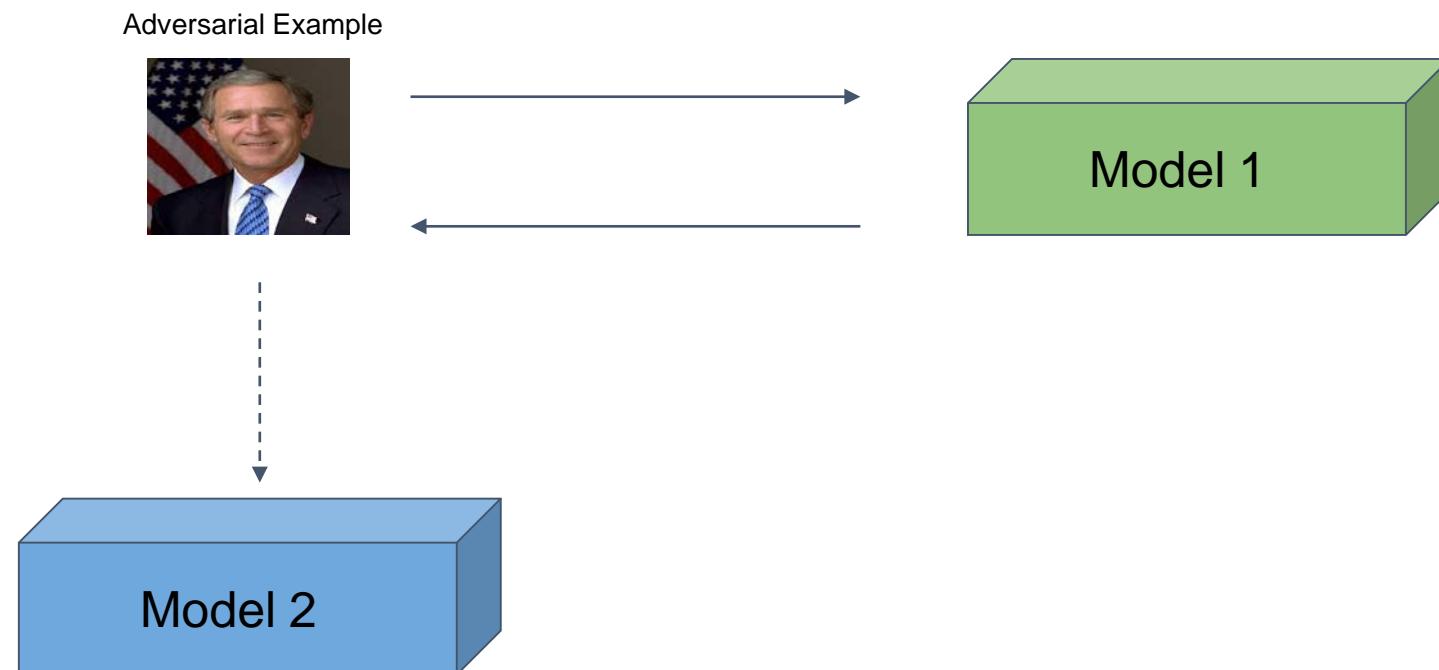
s.t. $\| \delta \|_p < \varepsilon$



Adversarial examples transfer between different models.

An adversarial example crafted against one model will generally fool other models.

Attackers do not need repeated access to your system to attack it.

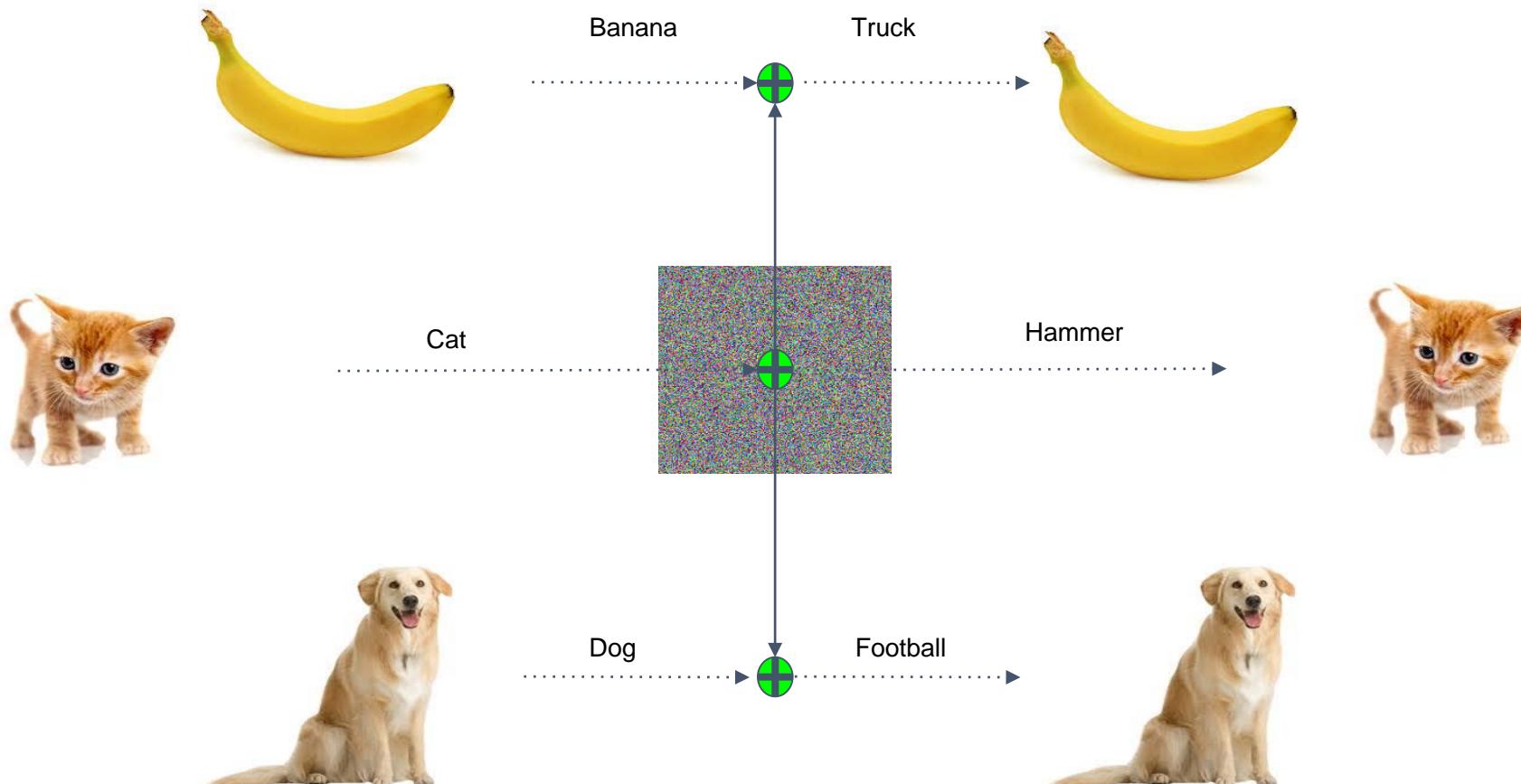


Why do adversarial examples exist and why do they transfer?

- Extreme nonlinearity or linearity (!?) of neural networks
- Linear behavior in high dimensional spaces is sufficient to produce adversarial examples in neural networks.
- Too much freedom! Impossible to handle all directions

In the most extreme case, it is possible to construct a single perturbation that will fool a model when added to any image!

Attackers need minimal resources to attack your system!



Attacks are not restricted to computer vision

Twitter Bot Detection [1].

Detection tools can easily be fooled by tweaking the number of replies or retweets.

Text Classification [2].

Original Text Prediction = **Negative**. (Confidence = 78.0%)

*This movie had **terrible** acting, **terrible** plot, and **terrible** choice of actors. (Leslie Nielsen ...come on!!!)
the one part I **considered** slightly funny was the battling FBI/CIA agents, but because the audience was
mainly **kids** they didn't understand that theme.*

Adversarial Text Prediction = **Positive**. (Confidence = 59.8%)

*This movie had **horridic** acting, **horridic** plot, and **horifying** choice of actors. (Leslie Nielsen ...come
on!!!) the one part I **regarded** slightly funny was the battling FBI/CIA agents, but because the audience
was mainly **youngsters** they didn't understand that theme.*

Audio [3].



“without the dataset the article is useless”



“okay google browse to evil dot com”



“ ”



“speech can
be embedded
in music”

[1] Kulyanich et al. Evading classifiers in discrete domains with provable optimality guarantees.

[2] Alzantot et al. Generating Natural Language Adversarial Examples.

[3] Carlini et al. Audio Adversarial Examples.

In fact, vision-type attacks do not apply to many security/privacy problems

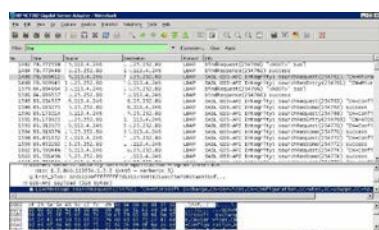
Hiding demographics from Tweets

$$\text{I love Justin Bieber!} + \epsilon \text{ [Redacted image]} = \text{[Blue box with ???]}$$

Hiding malware from a detector


$$+ \epsilon \text{ [Redacted image]} = \text{[Blue box with ???]}$$

Hiding malicious traffic from an Intrusion Detection system


$$+ \epsilon \text{ [Redacted image]} = \text{[Blue box with ???]}$$

In fact, vision-type attacks do not apply to many security/privacy problems

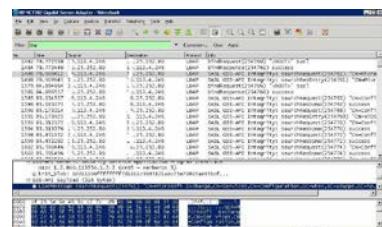
Hiding demographics from Tweets

$$\text{I love Justin Bieber!} + \epsilon \text{ [Redacted image]} = \text{[Blue box with ???]}$$

Hiding malware from a detector

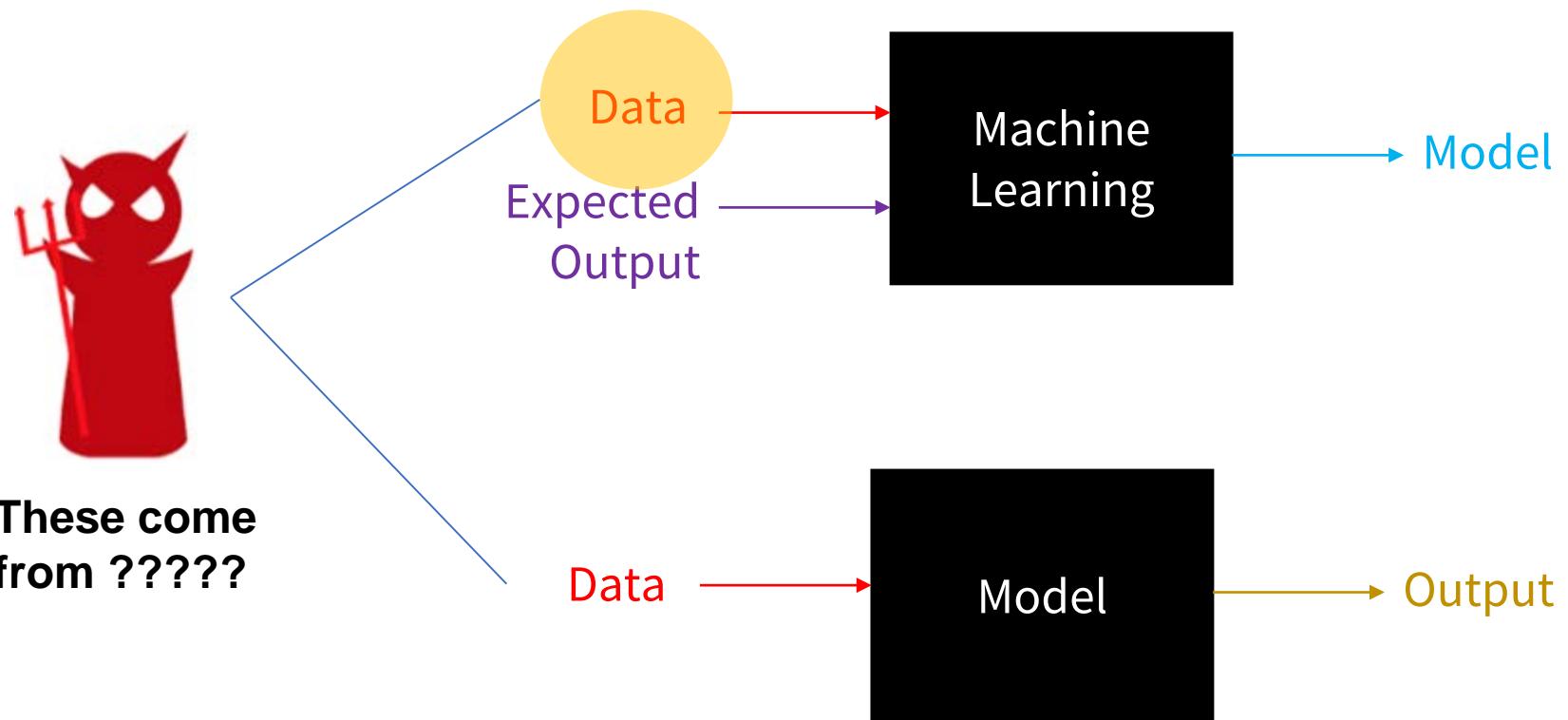

$$+ \epsilon \text{ [Redacted image]} = \text{[Blue box with ???]}$$

Hiding malicious traffic from an Intrusion Detection system

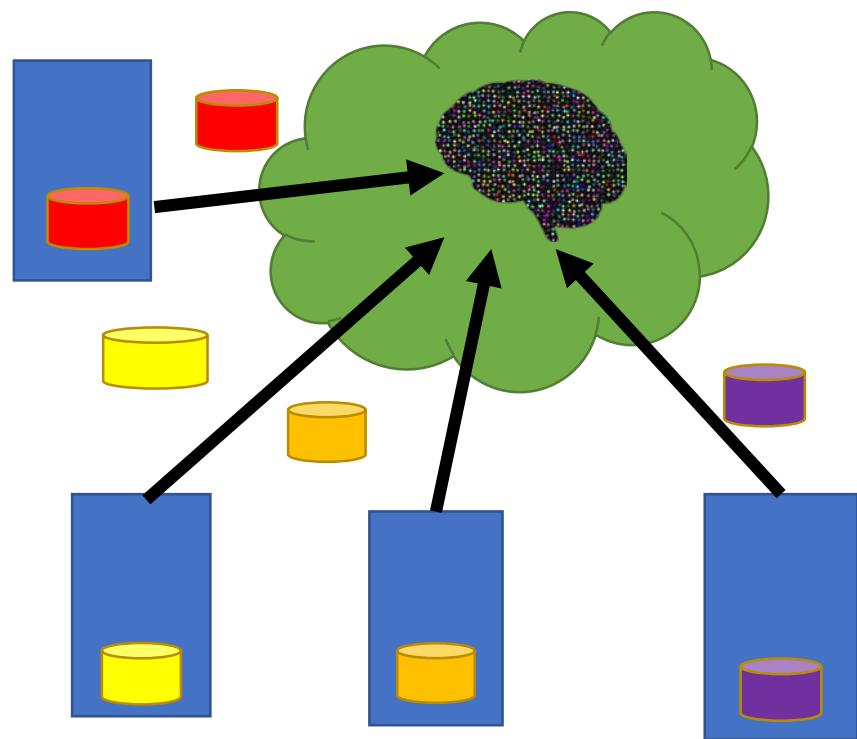

$$+ \epsilon \text{ [Redacted image]} = \text{[Blue box with ???]}$$

Ad-hoc heuristic attacks!
Active research area

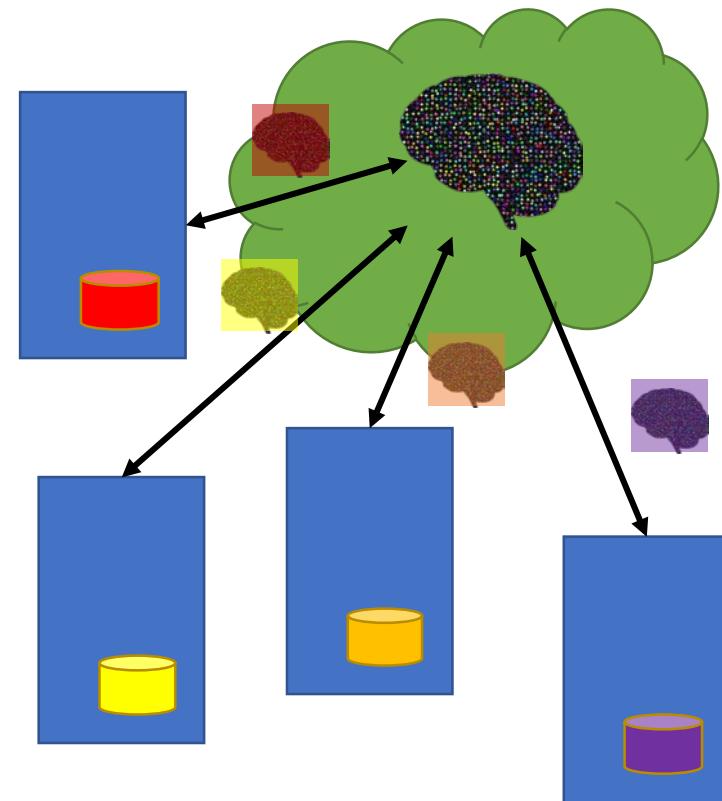
Issues with deploying ML Systems In The Wild



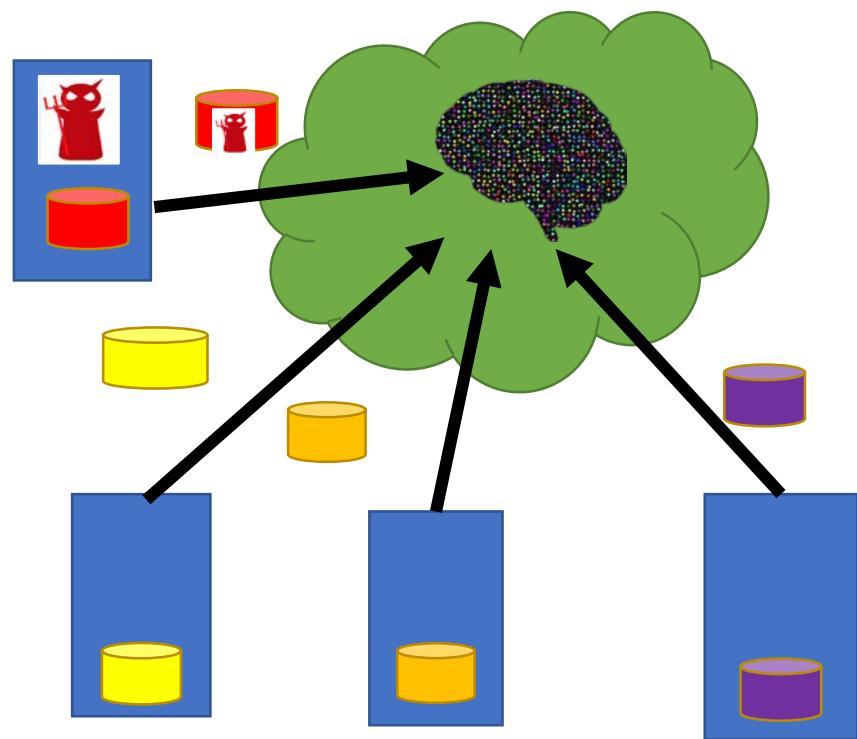
Centralized



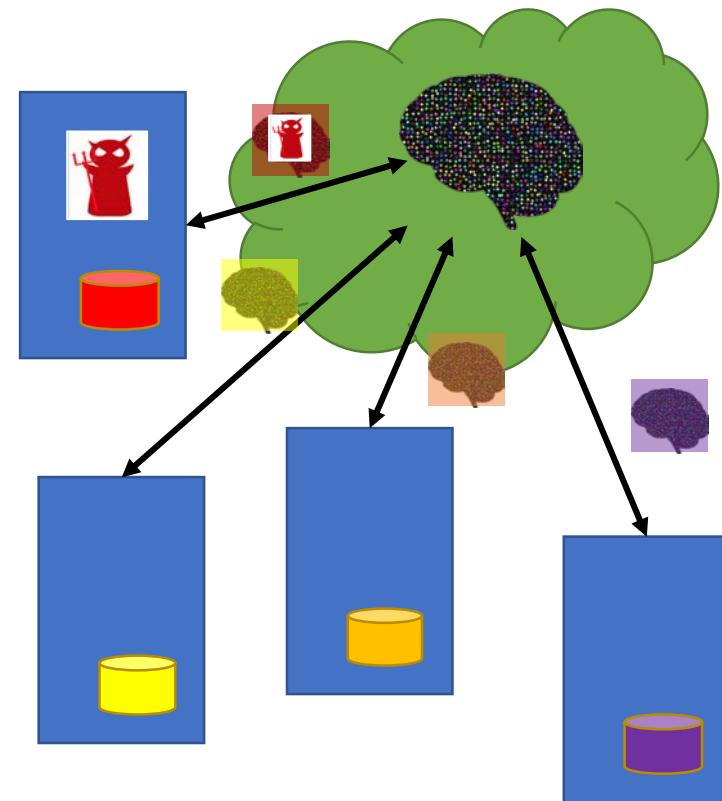
Federated



Centralized



Federated



Assumptions



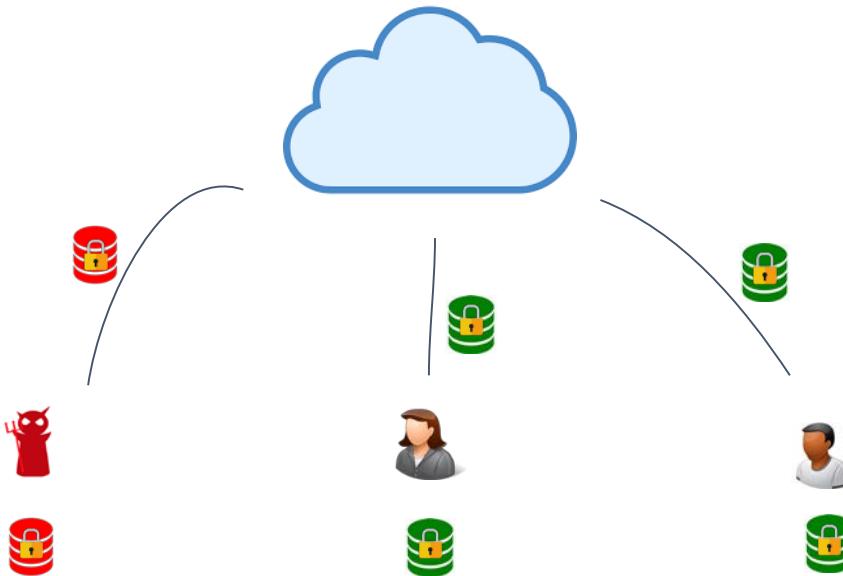
Assumptions

- Adversary controls a subset of inputs.



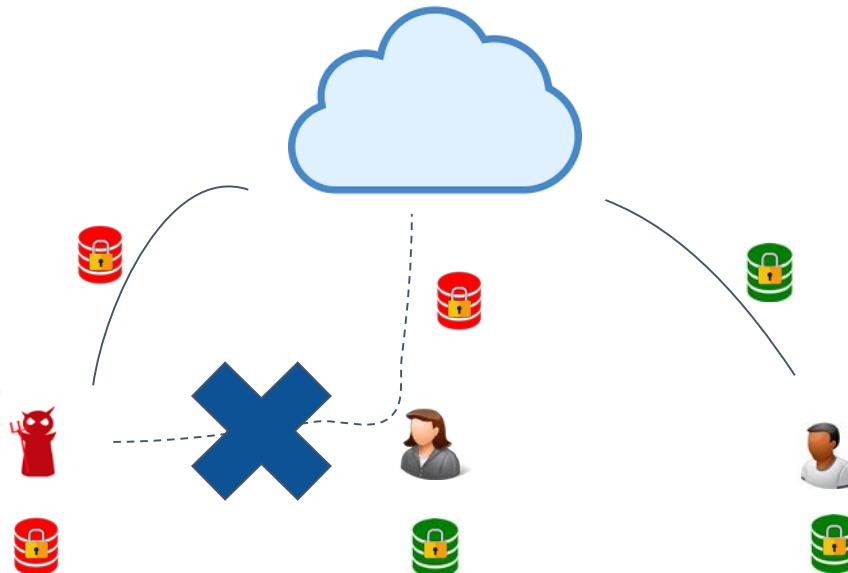
Assumptions

- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.



Assumptions

- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.
- Adversary **cannot** modify the training code or inputs from other parties.



Assumptions

- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.
- Adversary **cannot** modify the training code or inputs from other parties.

Potential Attacks

Assumptions

- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.
- Adversary **cannot** modify the training code or inputs from other parties.

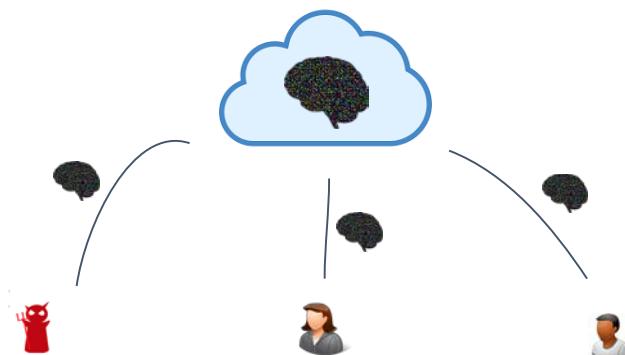
Potential Attacks

- Create a model that reveals information about other parties data.



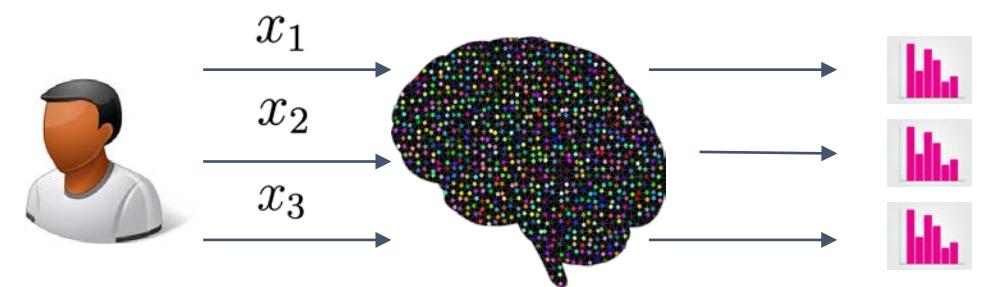
Assumptions

- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.
- Adversary **cannot** modify the training code or inputs from other parties.



Potential Attacks

- Create a model that reveals information about other parties data.
- Reduce the utility of the model (*poisoning*).

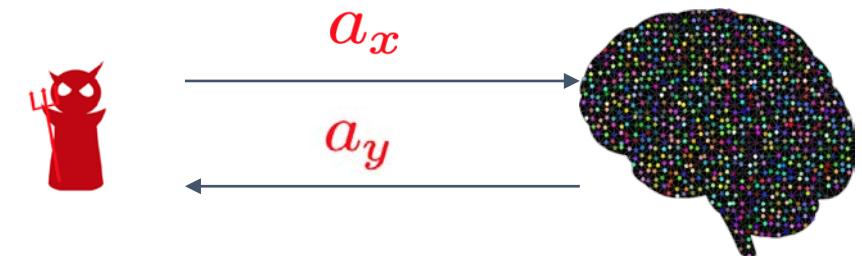
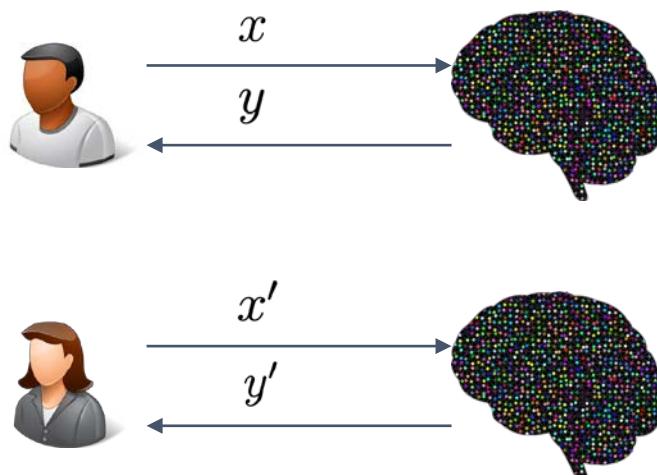


Assumptions

- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.
- Adversary **cannot** modify the training code or inputs from other parties.

Potential Attacks

- Create a model that reveals information about other parties data.
- Reduce the utility of the model (*poisoning*).
- Backdoor the model.

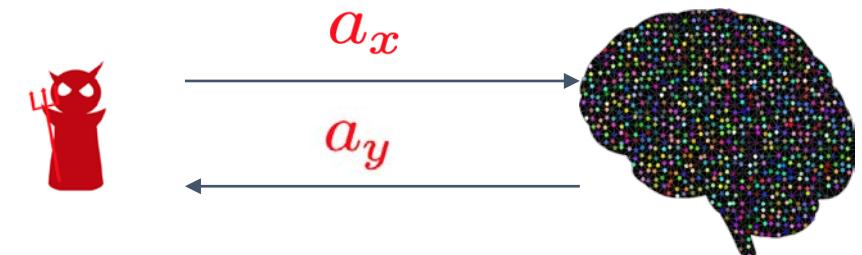
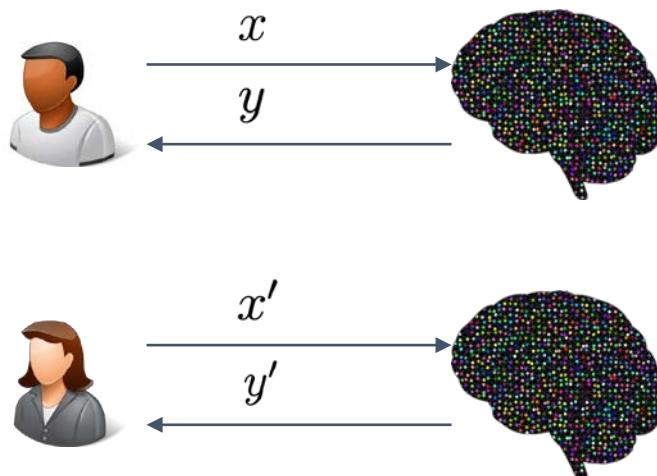


Assumptions

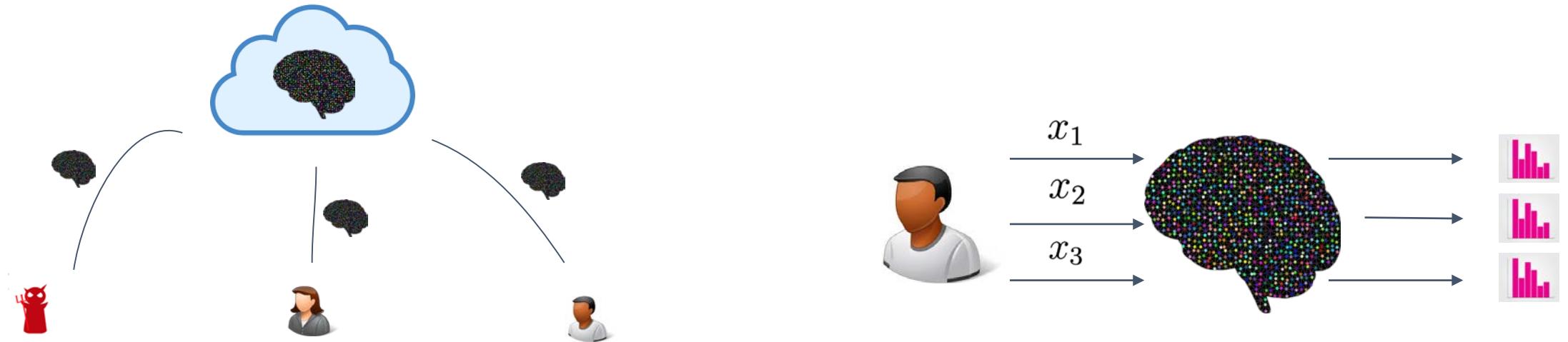
- Adversary controls a subset of inputs.
- Adversary can modify the inputs they send to central server.
- Adversary **cannot** modify the training code or inputs from other parties.

Potential Attacks

- Create a model that reveals information about other parties data.
- Reduce the utility of the model (*poisoning*).
- Backdoor the model.



(2) Decreasing the utility of the model (poisoning attack)



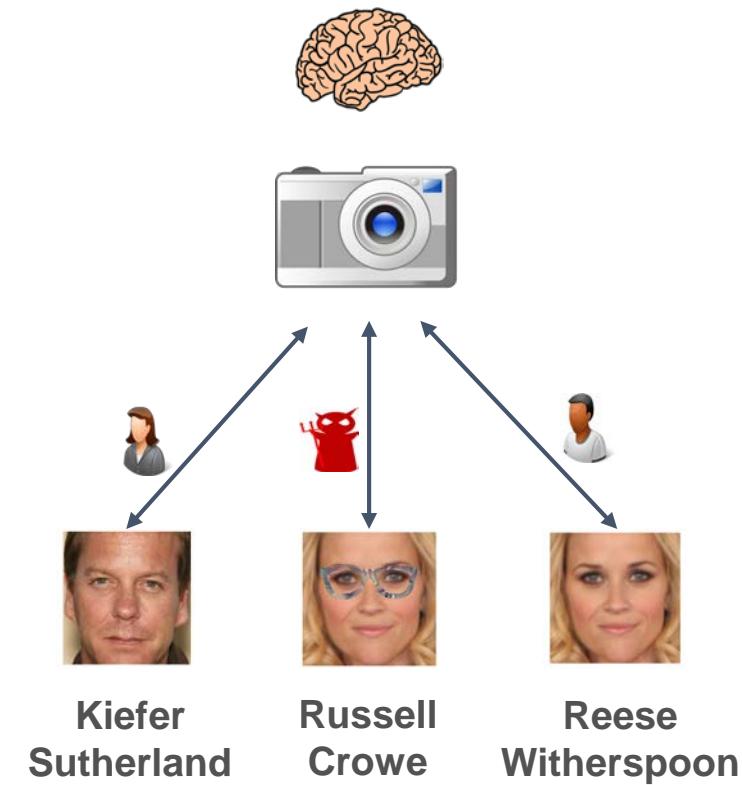
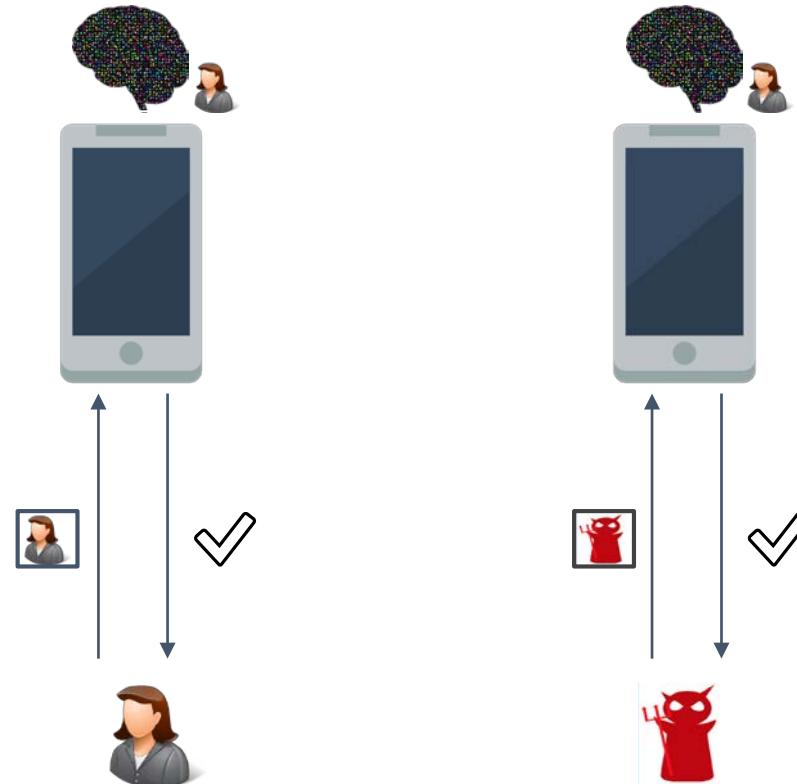
Utility:

A denial-of-service attack

A manipulation attack

(3) Backdoor attack

Given a trained model f that performs well on test data, there exists attacker chosen inputs \mathbf{x} with attacker chosen class t , such that $f(\mathbf{x}) = t$.



How to mount a poisoning attack?

Quick ML refresher I

Objective: find model parameters that *minimize* empirical loss

$$\begin{aligned} w^* &= \arg \min_w \mathbb{E}_{x,y \sim D} [L(x, y; w)] \\ &= \arg \min_w \sum_{x,y \in X} L(x, y; w) \end{aligned}$$

Annotations:

- A curly brace above the term $\mathbb{E}_{x,y \sim D} [L(x, y; w)]$ is labeled "Data distribution".
- A curly brace below the term $L(x, y; w)$ is labeled "Loss at (x, y) if model parameters are w ".
- A curly brace under the term $x, y \in X$ is labeled "Training data".

How to mount a poisoning attack?

One way is to solve a two-level optimization problem:

Whatever the adversary wants to achieve, e.g., high loss, mistakes on certain inputs (backdooring)

$$\delta^* = \arg \max_{\delta} U(w^*)$$

$$\text{where } w^* = \arg \min_w \sum_{x,y \in X \cup \{x' + \delta\}} L(x, y; w)$$

Simulated training on normal data X and a poisoned example $x' + \delta$

How to mount a poisoning attack?

$$\delta^* = \arg \max_{\delta} U(w^*)$$

$$\text{where } w^* = \arg \min_w \sum_{x,y \in X \cup \{x' + \delta\}} L(x, y; w)$$

Usually computationally expensive (have to simulate training inside) and requires many poisoned inputs—but powerful.

How to mount a poisoning attack?

$$\delta^* = \arg \max_{\delta} U(w^*)$$

$$\text{where } w^* = \arg \min_w \sum_{x,y \in X \cup \{x' + \delta\}} L(x, y; w)$$

Usually computationally expensive (have to simulate training inside) and requires many poisoned inputs—but powerful.

**And also you need to
be able to poison!!**

Takeaways

Deploying in adversarial environments is **HARD**

The adversary controls the inputs

To deployed models:
can always win!

To training:
can learn others inputs
can change his/others outputs

We are far from being able to have defenses

High-dimensional spaces difficult to characterize

This lecture

- Machine Learning – basics
- Privacy implications of Machine Learning
- Machine learning under adversarial conditions
- **Issues with applying Machine Learning to Security and Privacy problems**



Training is not the same as deployment...

What works in the lab doesn't always work in the wild

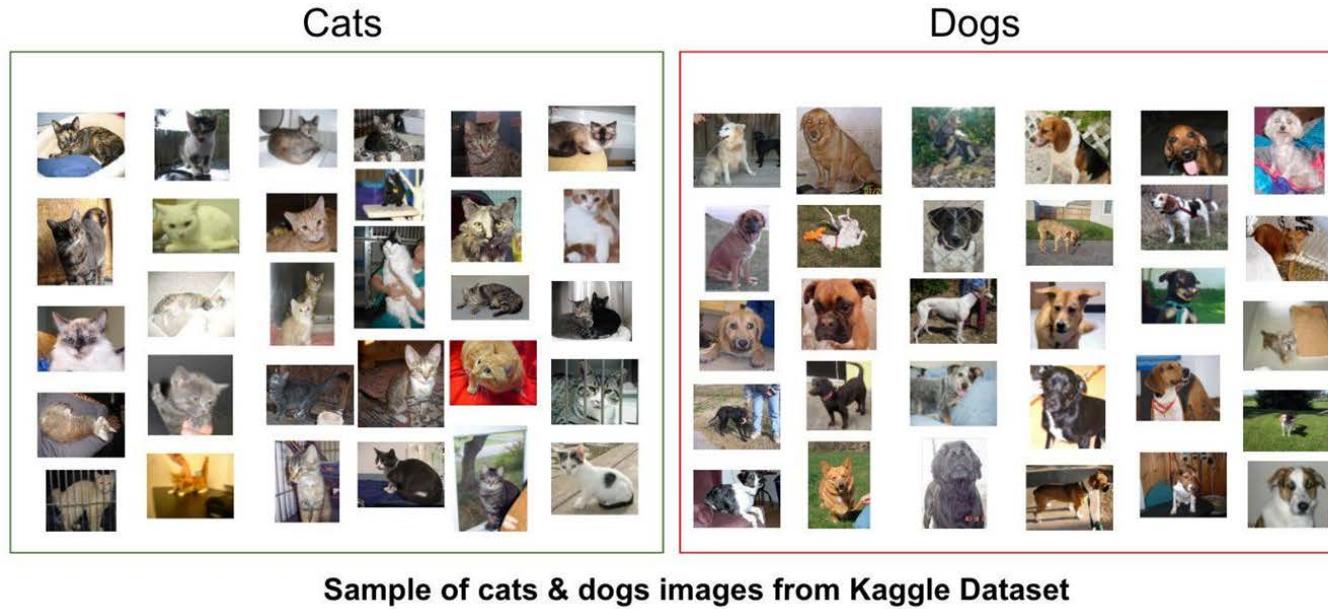


Figure 4: The cats and dogs in our sample that are most cat-like and most dog-like, according to the classifier of section 2.3.

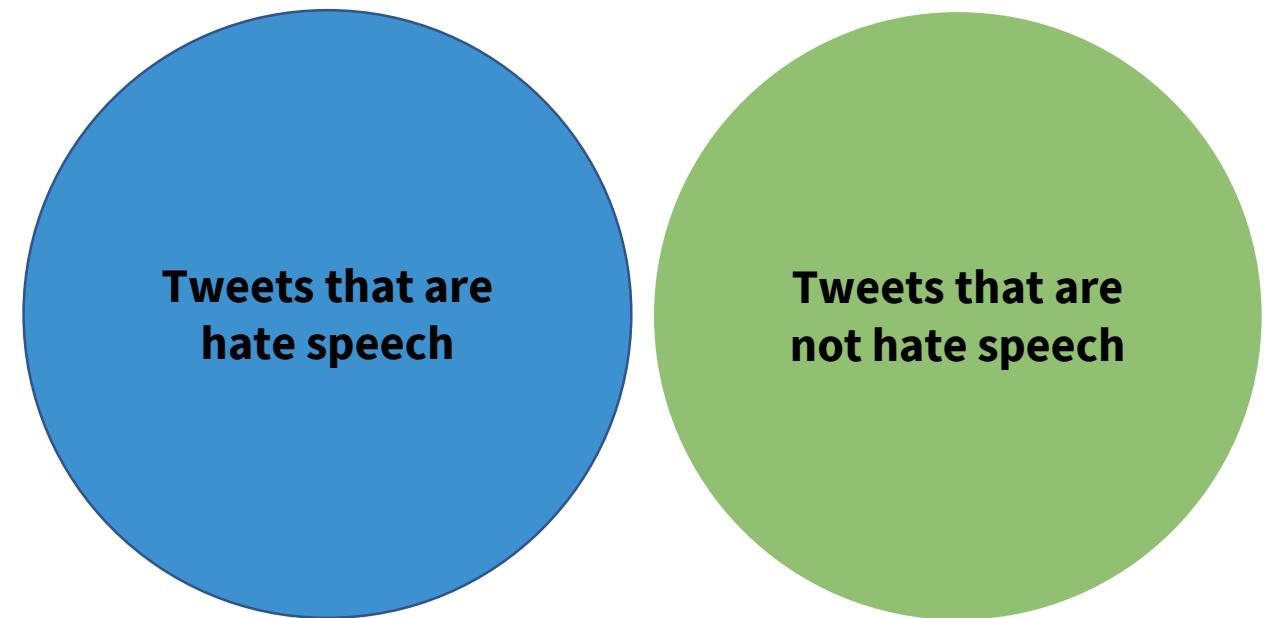
Issues with deploying ML Systems In The Wild

- ML often finds correlation, not causation
- Bias
- **Base Rate Fallacy**
- Adversarial ML threats
- Result in antisocial and negative environmental outcomes
- Externalize risks
- Are built to only benefit a subset of users
- **Produce errors due to distributional shift**
- Reward hacking: get good score without fulfilling the purpose
- **Distribute errors unfairly**
- **Lack of transparency**
- ...

Base Rate Fallacy / Prosecutor's fallacy

Quick reminder: AI performance metrics

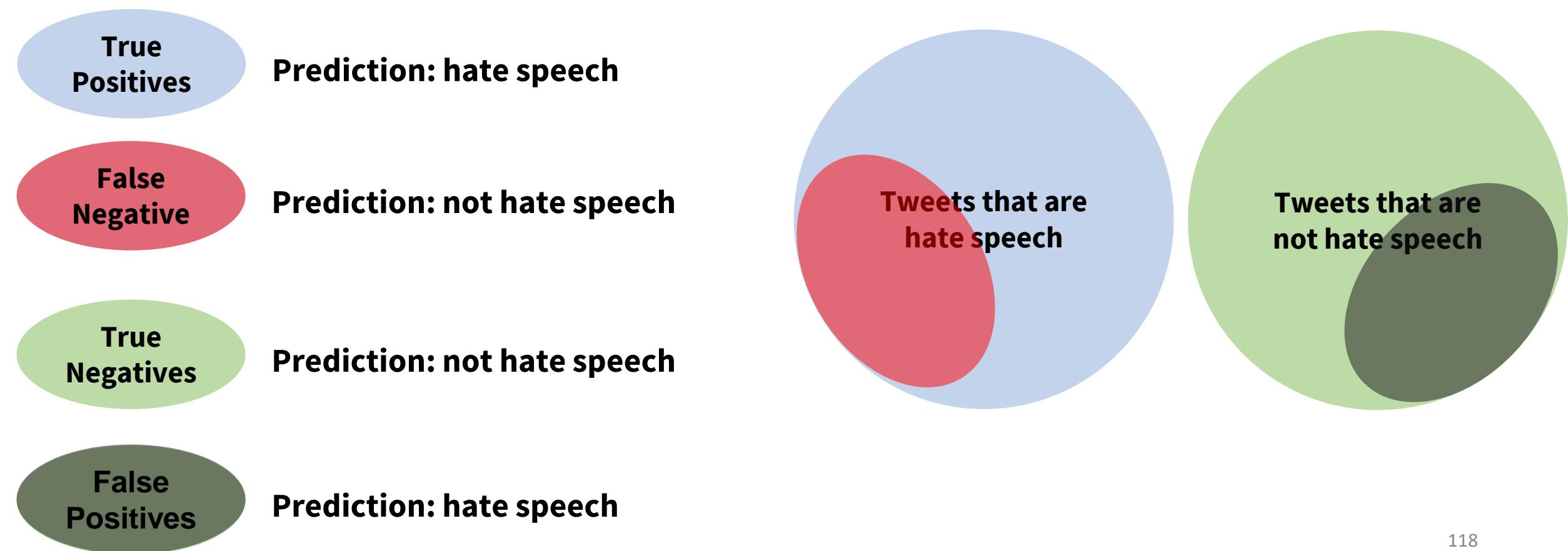
Example: Hate speech detection



Base Rate Fallacy / Prosecutor's fallacy

Quick reminder: AI performance metrics

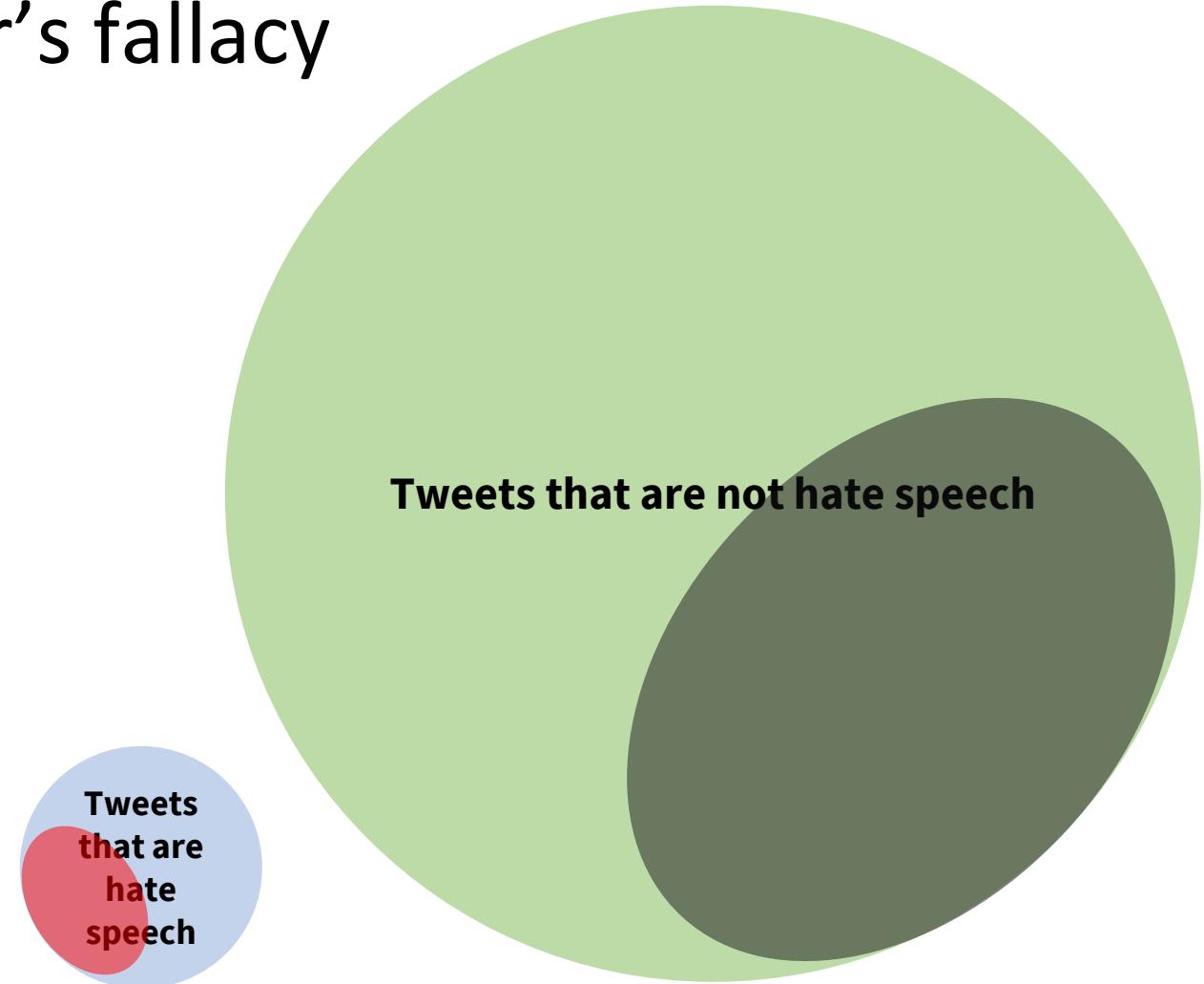
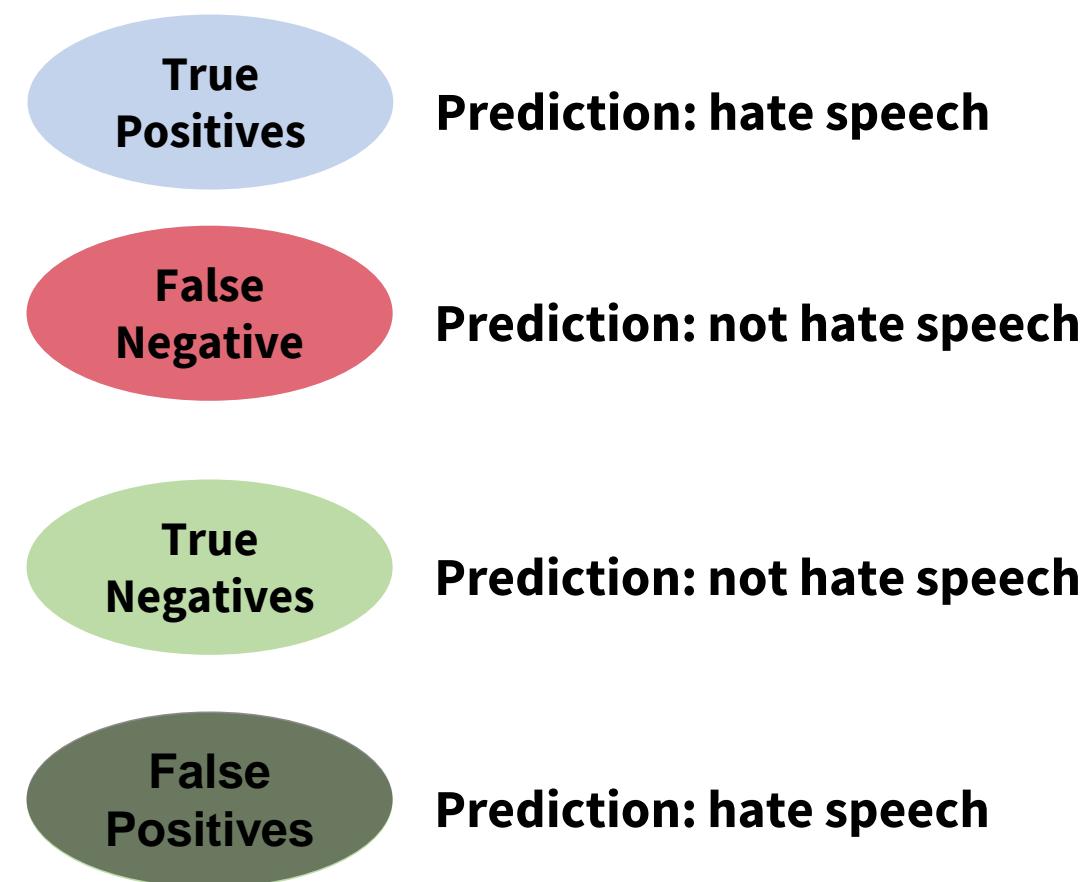
Example: Hate speech detection



Base Rate Fallacy / Prosecutor's fallacy

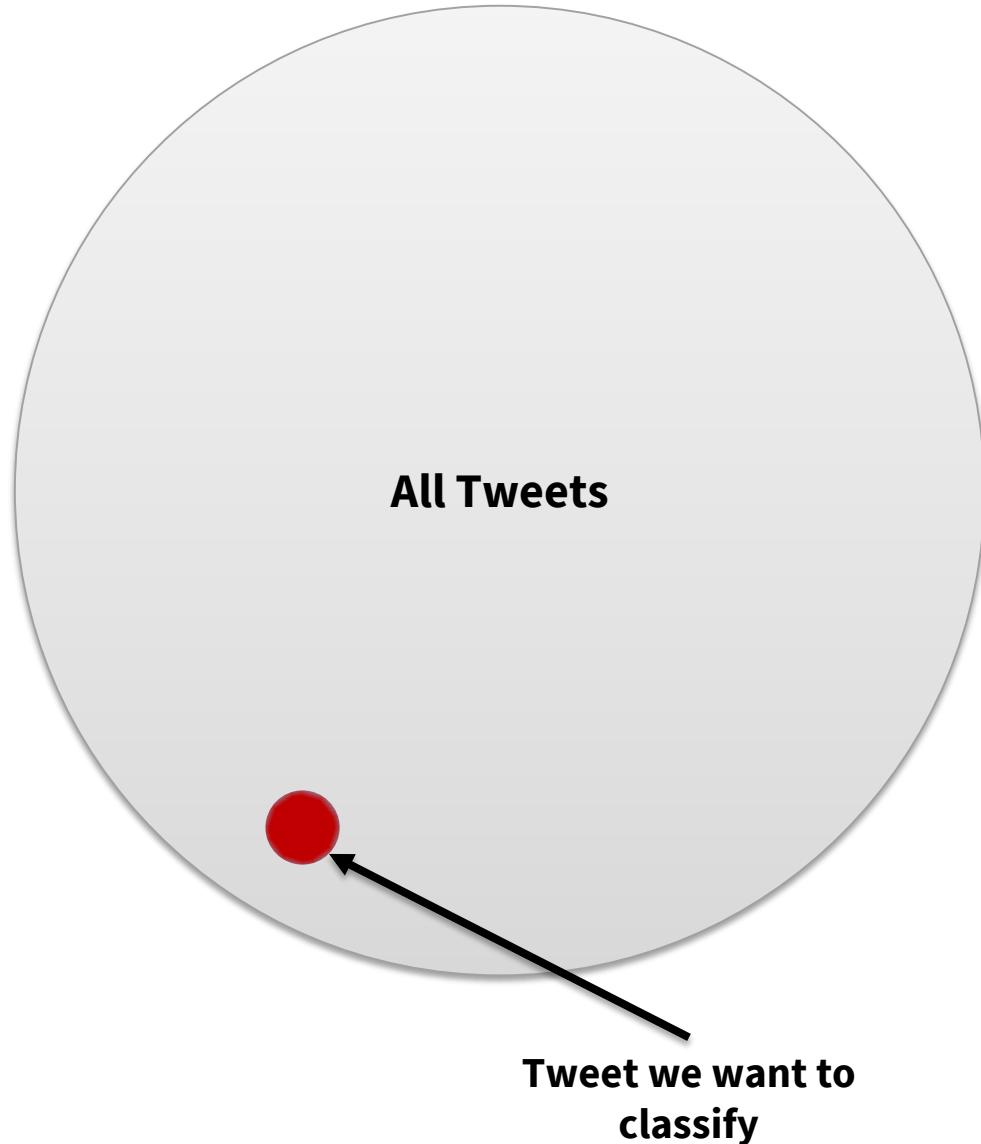
Quick reminder: AI performance metrics

Example: Hate speech detection



Base Rate Fallacy / Prosecutor's fallacy

- **Example:** Predicting hate speech
- "**Reality**": 1 tweet out of 1000 is hate speech



Base Rate Fallacy / Prosecutor's fallacy

- **Example:** Predicting hate speech
- "**Reality**": 1 tweet out of 1000 is hate speech

| | Classifier Says Hate | Classifier Says Not Hate |
|---------------------|----------------------|--------------------------|
| Tweet actually hate | True Positive | False Negative |
| Tweet is not hate | False Positive | True Negative |



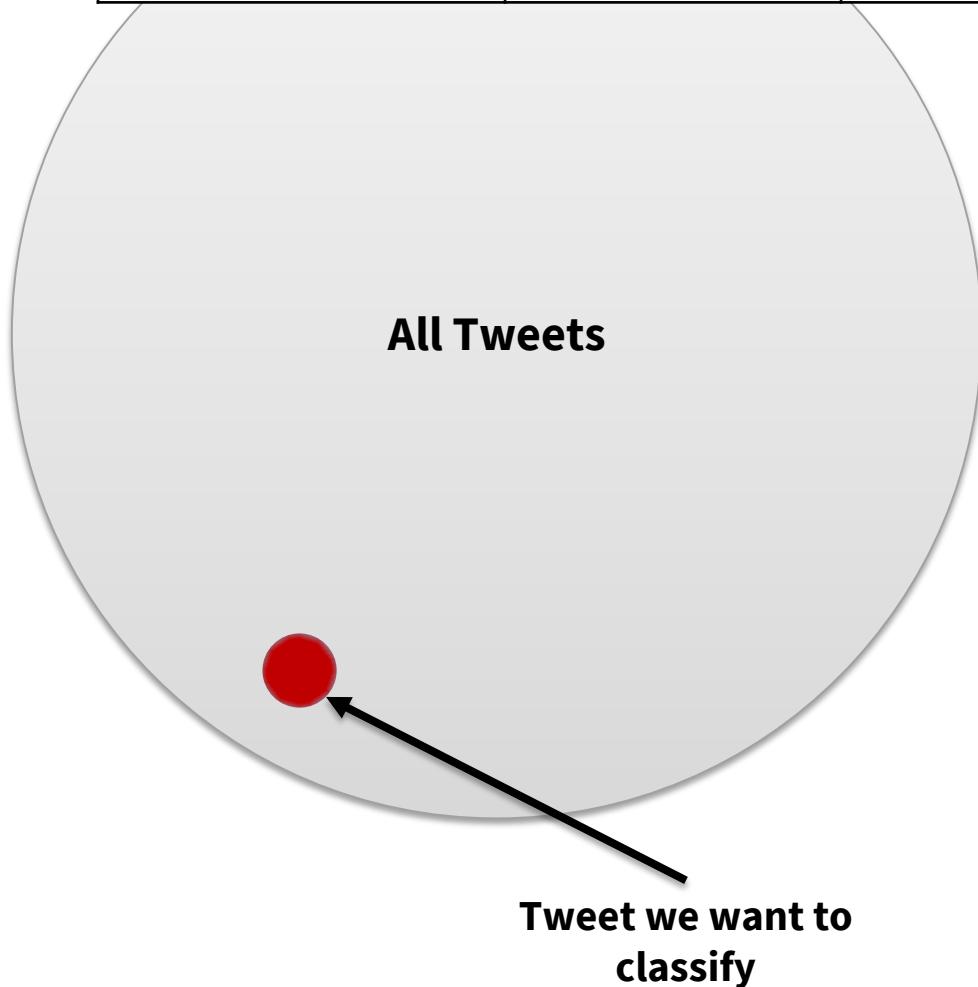
All Tweets

Tweet we want to
classify

Base Rate Fallacy / Prosecutor's fallacy

- **Example:** Predicting hate speech
- "**Reality**": 1 tweet out of 1000 is hate speech
- **Our classifier:**
 - False Positive Rate of 5%
 - 5 out 100 times a non-hate tweet is said to be hate speech
 - True Positive Rate of 100%
 - **Zero** false negatives, never misses a hate speech tweet
 - If tweet is hate, the classifier will say YES

| | Classifier Says Hate | Classifier Says Not Hate |
|---------------------|----------------------|--------------------------|
| Tweet actually hate | True Positive | False Negative |
| Tweet is not hate | False Positive | True Negative |



Base Rate Fallacy / Prosecutor's fallacy

- **Example:** Predicting hate speech
- "**Reality**": 1 tweet out of 1000 is hate speech

- **Our classifier:**

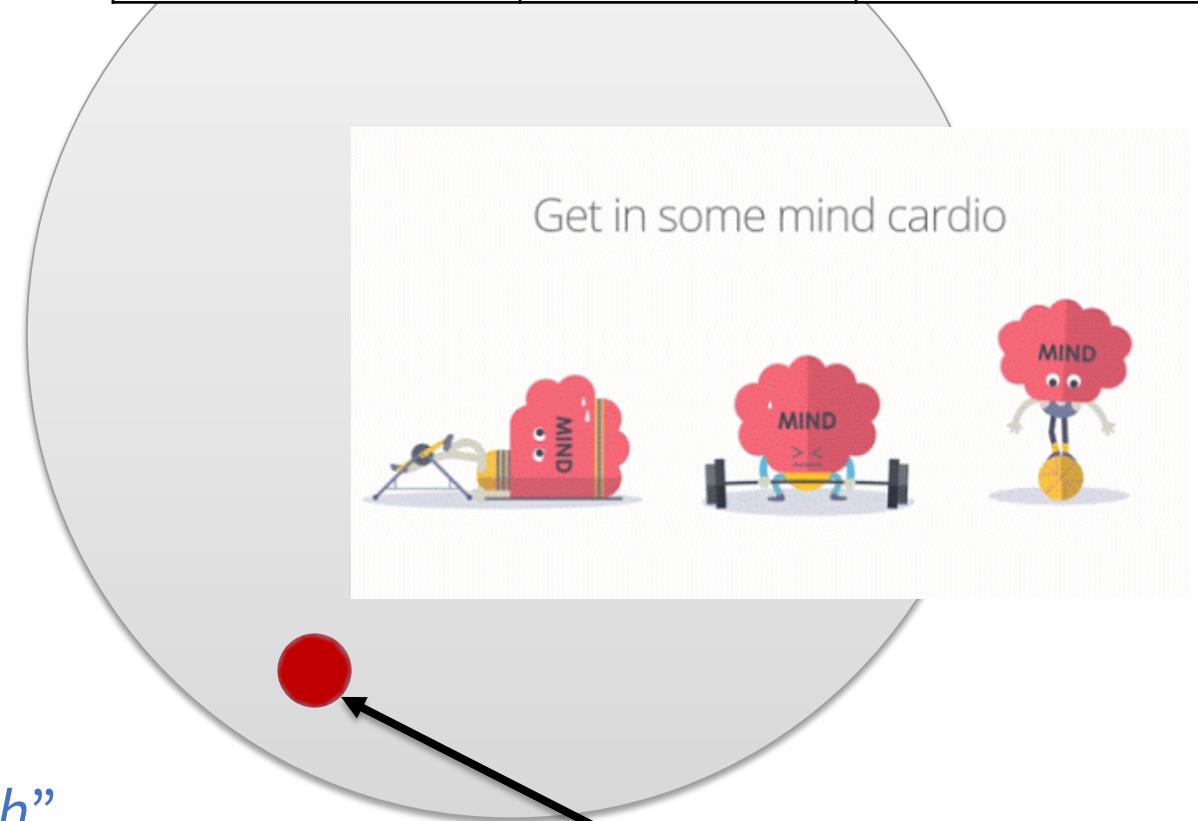
- False Positive Rate of 5%
 - 5 out 100 times a non-hate tweet is said to be hate speech
- True Positive Rate of 100%
 - **Zero** false negatives, never misses a hate speech tweet
 - If tweet is hate, the classifier will say YES

Given a tweet, the classifier says “hate speech”

What is the probability that they actually will?

$\Pr[\text{Hate speech} \mid \text{Classifier Says hate speech}] ?$

| | Classifier Says Hate | Classifier Says Not Hate |
|---------------------|----------------------|--------------------------|
| Tweet actually hate | True Positive | False Negative |
| Tweet is not hate | False Positive | True Negative |



Tweet we want to
classify

Base Rate Fallacy / Prosecutor's fall

Given a tweet, the classifier says “*hate speech*”
What is the probability that they actually will?

- **Example:** Predicting hate speech
- **"Reality":** 1 tweet out of 1000 is hate speech
- **Our classifier:**
 - **False Positive Rate of 5%**
 - 5 out 100 times a non-hate tweet is said to be hate speech
 - **True Positive Rate of 100%**
 - Zero false negatives, never misses a hate speech tweet
 - If the tweet is hate, the classifier will say YES

$\Pr[\text{Hate} \mid \text{Classifier Says Hate}] =$

$$= \Pr[\text{Classifier Says Hate} \mid \text{Hate}] \Pr[\text{Hate}] / \Pr[\text{Classifier Says Hate}]$$

$$= 1 * 0.001 / (1 * 0.001 + 0.05 * 0.999) = 0.001 / 0.05095 = 2\%$$

$$\Pr[\text{Classifier Says Hate} \mid \text{Hate}] \Pr[\text{Hate}] + \Pr[\text{Classifier Says Hate} \mid \text{not Hate}] \Pr[\text{Not Hate}]$$

$(1 - \Pr[\text{Hate}])$



Base Rate Fallacy / Prosecutor's fall

Given a tweet, the classifier says “hate speech”
What is the probability that they actually will?

- **Example:** Predicting hate speech
- **"Reality":** 1 tweet out of 1000 is hate speech
- **Our classifier:**

- False Positives

- 5 out of 1000 tweets are classified as hate speech

- True Positives

- Zero out of 1000 tweets are classified as hate speech
- If the classifier says “hate speech”, it is always correct

Take Away

The result of the classifier must be taken with a grain of salt, where the salt is the base rate for the event we are trying to predict/classify

$$\Pr[\text{Hate} | C] = \frac{\Pr[\text{Classifier Says Hate} | \text{Hate}] \Pr[\text{Hate}]}{\Pr[\text{Classifier Says Hate} | \text{Hate}] \Pr[\text{Hate}] + \Pr[\text{Classifier Says Hate} | \text{Not Hate}] \Pr[\text{Not Hate}]}$$

Events whose frequency is average may be fine, but rare events (i.e., very weak signals) are hard to find

$$= \Pr[\text{Classifier Says Hate} | \text{Hate}] \Pr[\text{Hate}] / \Pr[\text{Classifier Says Hate}]$$

$$= 1 * 0.001 / (1 * 0.001 + 0.05 * 0.999) = 0.001 / 0.05095 = 2\%$$

Distributional Shift

August 2018 Accuracy on Easy Pilot Parliaments Benchmark



98.7%

68.6%

100%

92.9%



98.7%

77.5%

100%

93.6%



**DARKER
MALES**



**DARKER
FEMALES**



**LIGHTER
MALES**



**LIGHTER
FEMALES**

Performance on Facial Analysis Task of Gender Classification

Distributional Shift

Occurs when a classifier is trained in one area and deployed in another.

Distributional Shift

Wide Definition

Face recognition: race / gender

Hate speech: countries

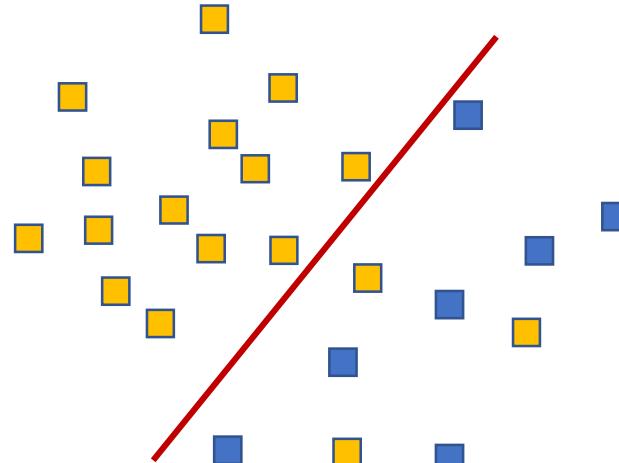
Behavior prediction: social network

...

Occurs when a classifier is trained in one **area** and deployed in another.

Distributional Shift

Occurs when a classifier is trained in one area and deployed in another.



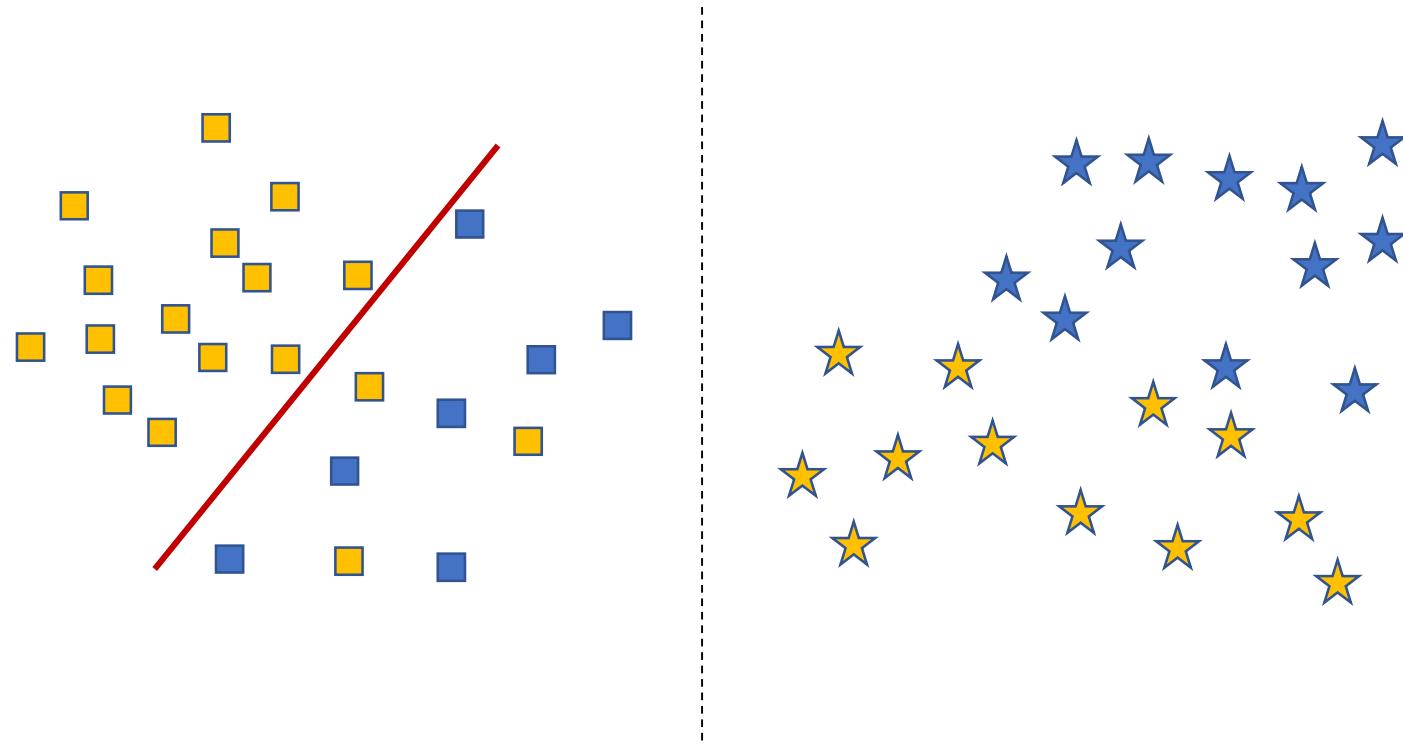
- Example: Hate speech prediction
- Training/Test in United States
- Yellow = Tweets that are hate speech
- Blue = Tweets that are not hate speech

Distributional Shift

Wide Definition

Face recognition: race / gender
Migration prediction: countries
Behavior prediction: social network
...

Occurs when a classifier is trained in one **area** and deployed in another.



- Example: Hate speech prediction
- Yellow = Tweets that are hate speech
- Blue = Tweets that are not hate speech

■ Training/Test in US

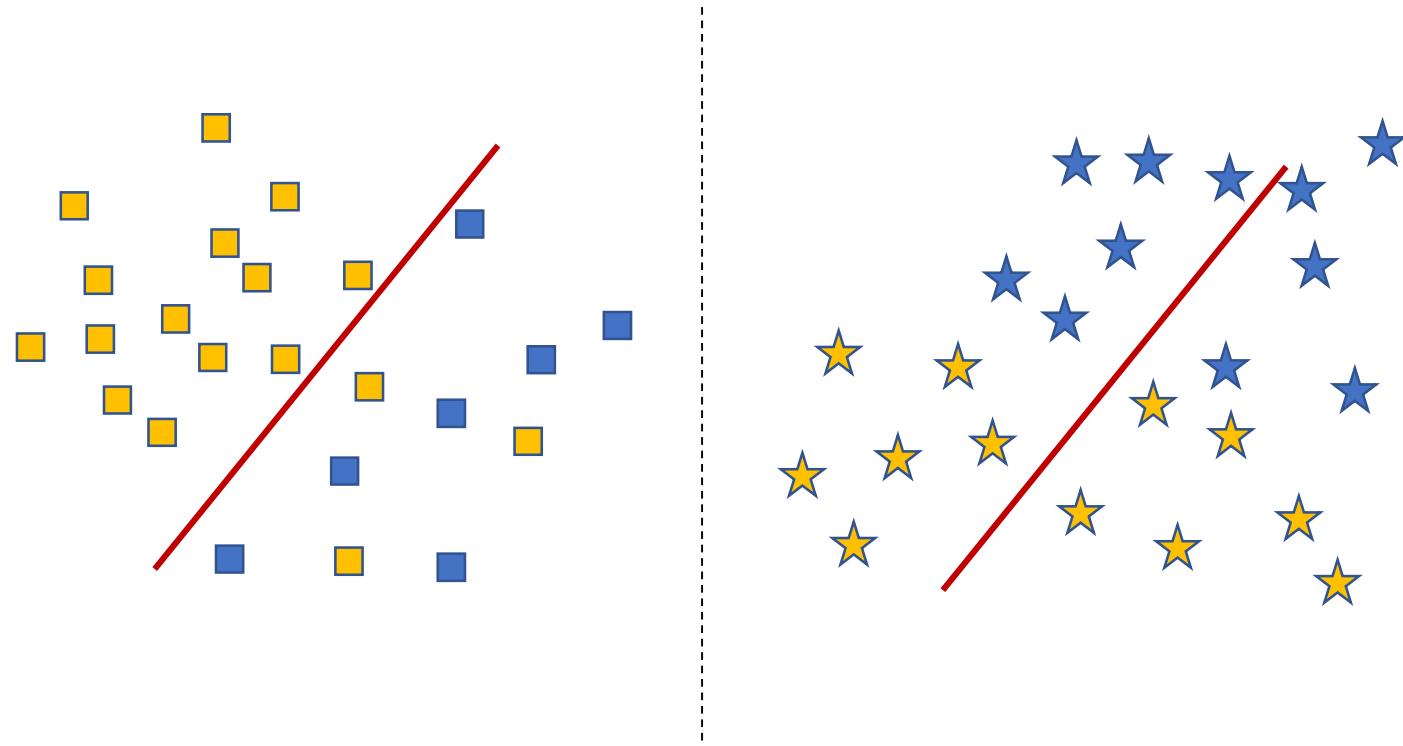
★ Training/Test in UK

Distributional Shift

Wide Definition

Face recognition: race / gender
Migration prediction: countries
Behavior prediction: social network
...

Occurs when a classifier is trained in one **area** and deployed in another.

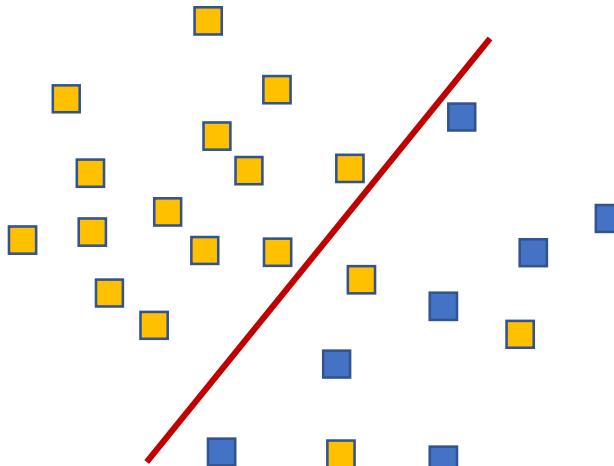


- Example: Hate speech prediction
- Yellow = Tweets that are hate speech
- Blue = Tweets that are not hate speech

■ Training/Test in US
★ Training/Test in UK

Distribution of Errors

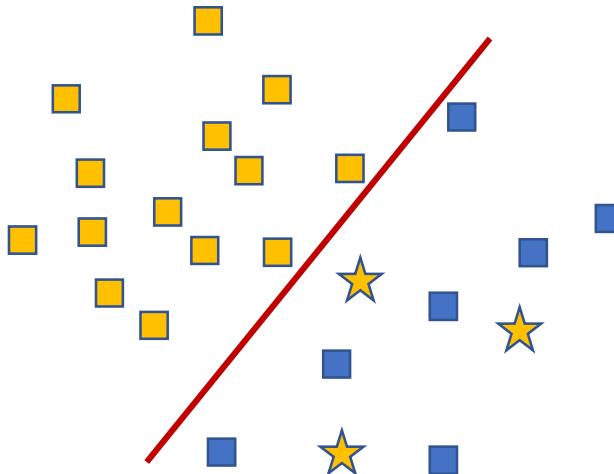
Occurs when most mistakes of the classifier are concentrated in a **subpopulation/group**



- Example: Hate speech prediction
- Training/Test in United States
- Yellow = Tweets that are hate speech
- Blue = Tweets that are not hate speech

Distribution of Errors

Occurs when most mistakes of the classifier are concentrated in a **subpopulation/group**



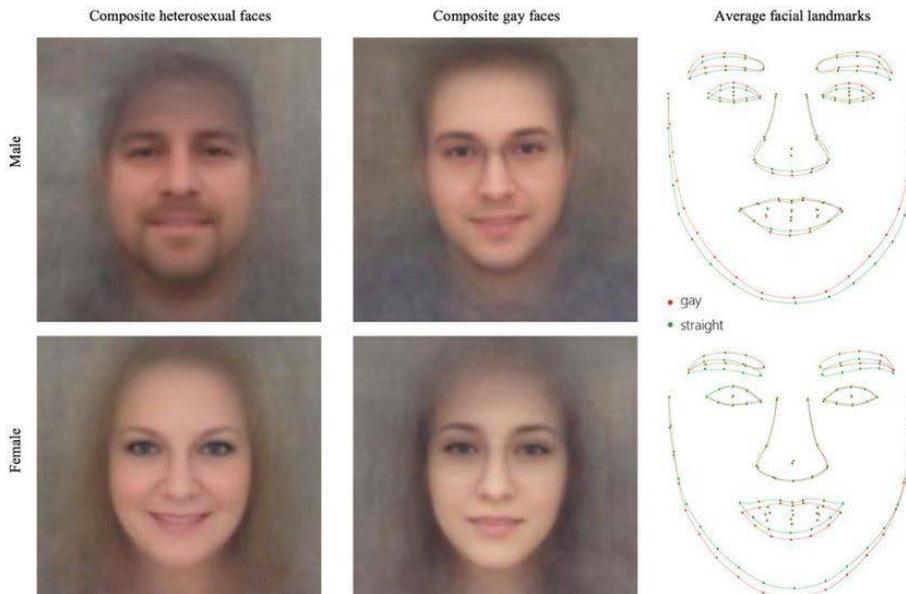
- Example: Hate speech prediction
 - Training/Test in United States
 - Yellow = Tweets that are hate speech
 - Blue = Tweets that are not hate speech
- ★ Tweets from US citizens
■ Tweets from foreigners

Transparency: correlation or causation?

- Classification Task: Should we send home a patient with Bronchitis?
- Rule Based Learning
 - If x , then y
 - Human readable rules: causation is intrinsic
- Machine Learning
 - Better accuracy
 - Not possible to understand why a decision is made

One example of “what is it learning?”

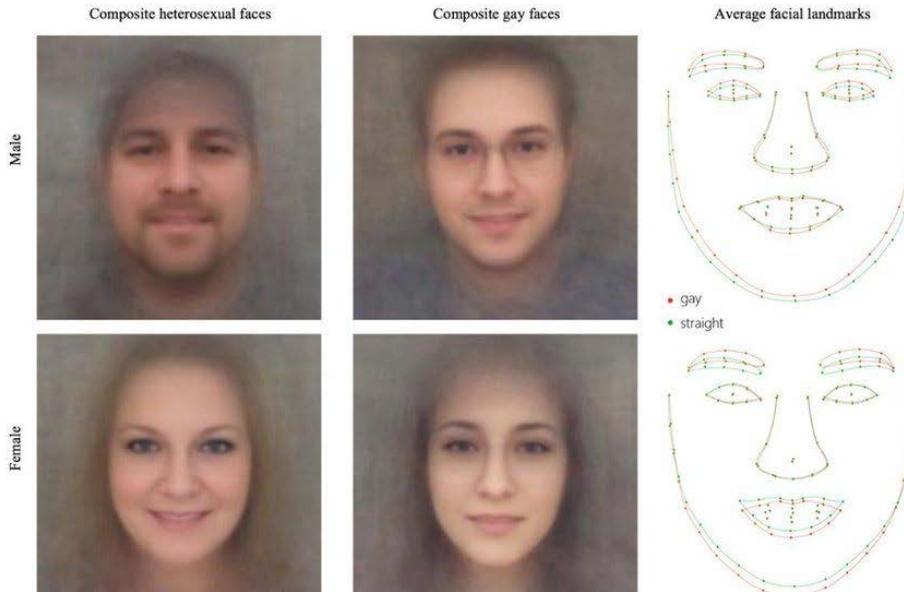
When the algorithm was presented with two photos where one picture was definitely of a gay man and the other heterosexual, it was able to determine which was which 81% of the time. With women, the figure was 71%



“A facial recognition experiment that claims to be able to distinguish between gay and heterosexual people has sparked a row between its creators and two leading LGBT rights groups.”

One example of “what is it learning?”

When the algorithm was presented with two photos where one picture was definitely of a gay man and the other heterosexual, it was able to determine which was which 81% of the time. With women, the figure was 71%



What did the algorithm learn?

Faces or stereotypical poses/gestures in the Dating site & Facebook pictures used for training?

Would it work in other social networks?

Does it work evenly for different races?

And for different social groups?

“A facial recognition experiment that claims to be able to distinguish between gay and heterosexual people has sparked a row between its creators and two leading LGBT rights groups.”

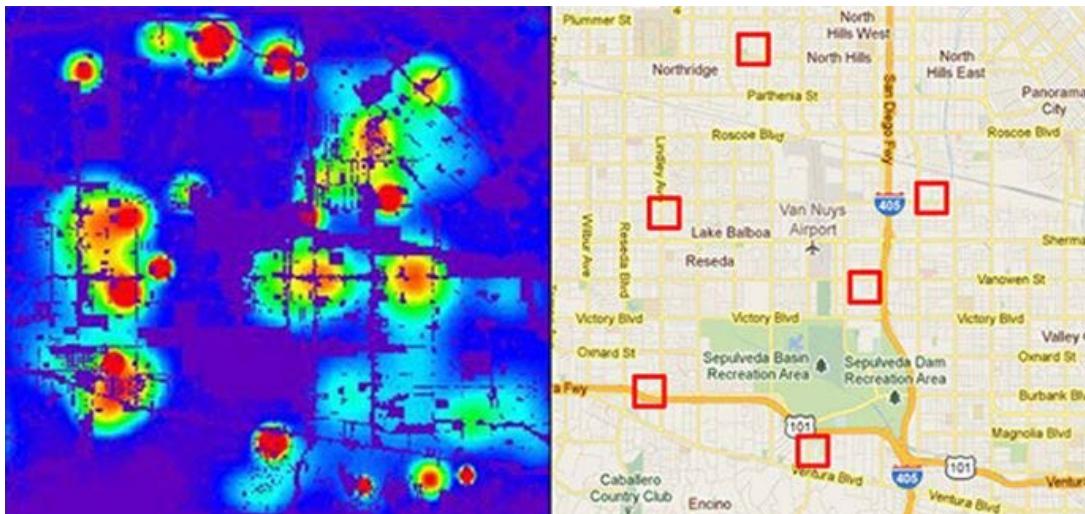
<https://www.bbc.co.uk/news/amp/technology-41188560>

<https://medium.com/@blaisea/do-algorithms-reveal-sexual-orientation-or-just-expose-our-stereotypes-d998fafdf477>

A bigger problem: bias reinforcement

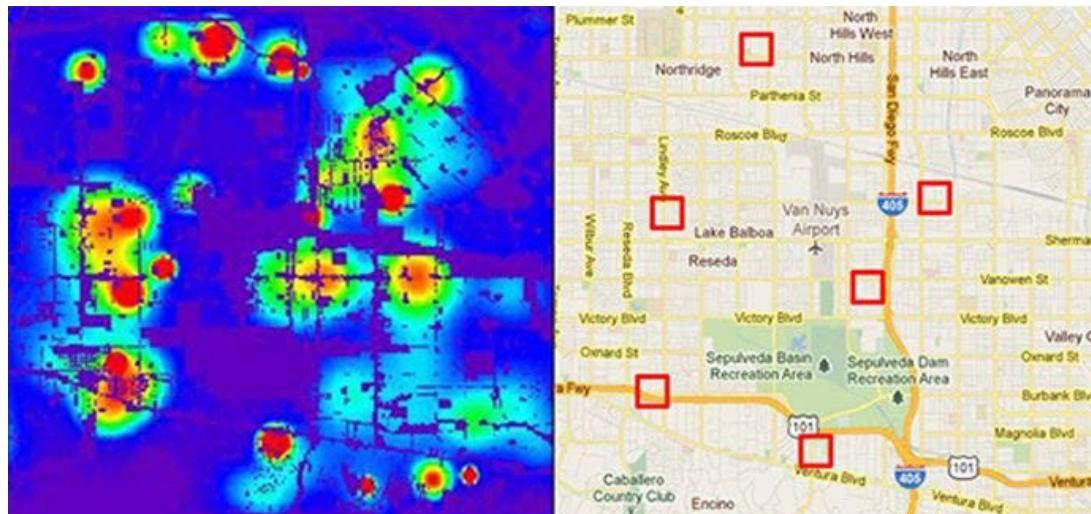


A bigger problem: bias reinforcement



Predictive policing: distribute resources (*policemen*)
according to needs (*crime*)

A bigger problem: bias reinforcement



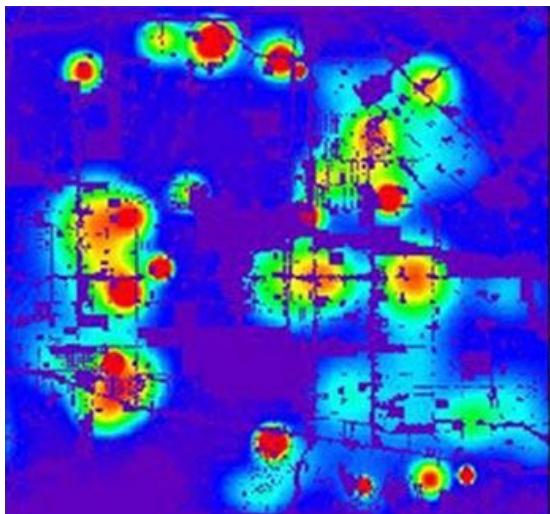
Problem #1:

Prediction may be biased
Training data = available reports
Minority / Poor neighborhoods affected



Predictive policing: distribute resources (*policemen*)
according to needs (*crime*)

A bigger problem: bias reinforcement



 **Predictive policing:** distribute resources (*policemen*) according to needs (*crime*)

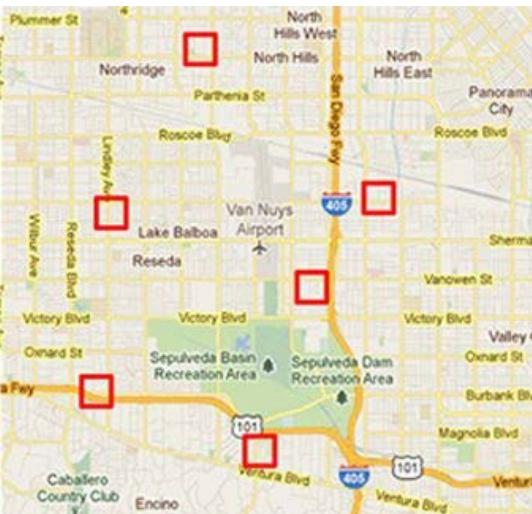
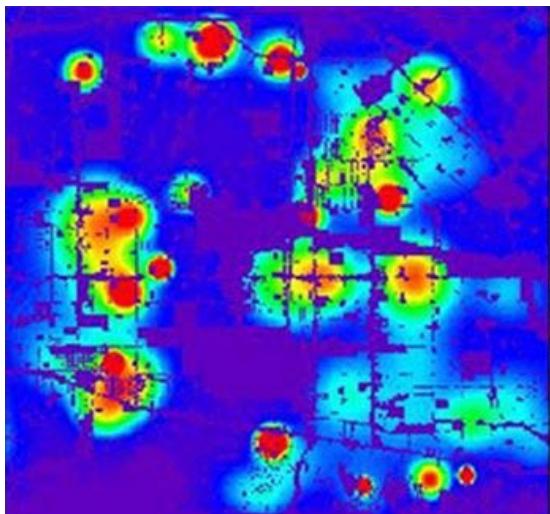
Problem #1:

Prediction may be biased
Training data = available reports
Minority / Poor neighborhoods affected

Problem #2:

System thought in a static! context
Under deployment...

A bigger problem: bias reinforcement



 **Predictive policing:** distribute resources (*policemen*) according to needs (*crime*)

Problem #1:

Prediction may be biased
Training data = available reports
Minority / Poor neighborhoods affected

Problem #2:

System thought in a static! context
Under deployment...

More police → More reports



What do we do????

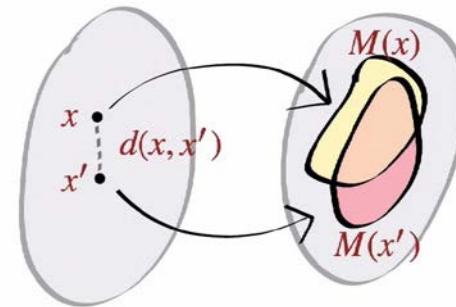
Step 1: What is Bias?

Statistical bias: difference between an estimator expected value and the true value
Very limited! Nothing about errors, nothing about distributional shift

Group fairness: outcome should not differ between demographic groups

- Predictive parity:* same prediction regardless of group (aka, Calibration)
- Equal false positive (rates)*
- Equal false negative (rates)*
- ...

Individual fairness: **similar?** individuals should be treated **similar?**



What do we do????

Step 2: Detecting Bias

What-if approach: play with the values until something changes, associate with bias

[Google What-If](#)

Explainability: try to understand why the prediction happen, associate with bias

[Lime](#)

[FairML](#)

What do we do????

Step 2: Detecting Bias

What-if approach: play with the values until something changes, associate with bias
[Google What-If](#)

Explainability: try to understand why the prediction happen, associate with bias
[Lime](#)
[FairML](#)

how do you know you explored the full space?

what about “proxies” (attributes associated to sensitive attributes)?

what about biases outside of your system?

what about contradictions between metrics?

What do we do????

Step 3: Removing Bias

The bias is in the training, let's fix the training

[Diversity in Faces by IBM](#)

Can we really build datasets like this for everything?

Cloudwalk agreement with Zimbabwe for facial recognition

IBM buys Flickr dataset

No worries about the dataset, we can fix the model

1001 different algorithms to get one/several of the mathematical definitions from CS approaches

Remember the impossibility result!

[IBM Bias Assessment Toolkit](#) (metrics, some detectors, nice references and tutorials)

What do we do????

Step 3: Removing Bias

how do you know you removed all biases?

“proxies” also apply here... Same as with anonymization!

Trade-off with utility!

The bias is in the training, let's fix the training

[Diversity in Faces by IBM](#)

Can we really build datasets like this for everything?

Cloudwalk agreement with Zimbabwe for facial recognition

IBM buys Flickr dataset

No worries about the dataset, we can fix the model

1001 different algorithms to get one/several of the mathematical definitions from CS approaches

Remember the impossibility result!

[IBM Bias Assessment Toolkit](#) (metrics, some detectors, nice references and tutorials)

Talking of anonymization... can that help?

Talking of anonymization... can that help?

NO

Talking of anonymization... can that help?

NO

None of the bias problems stems from identity

Removing identity does not remove sensitive attributes, groups, similarities

Anonymizing only makes bias analysis more difficult
(big trade-off here too!)

Takeaways

Deploying machine learning is hard: reality is far from lab conditions

Base rate matters: it is always hard to get good results on weak signals

Biases come in many flavors, we only saw a couple

Removing biases:

Many metrics

Exploratory identification

Narrow removal tools

Anonymizing only makes bias analysis more difficult