Information Security and Privacy (COM-402)

Machine Learning Security and Privacy

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

Slides adapted from Jamie Hayes, Bogdan Kulynych, Rebekah Overdorf

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

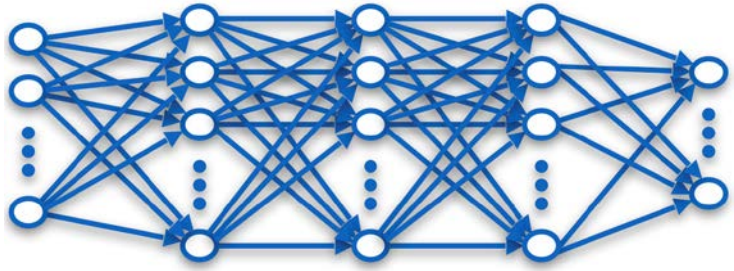
Machine Learning (ML)

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



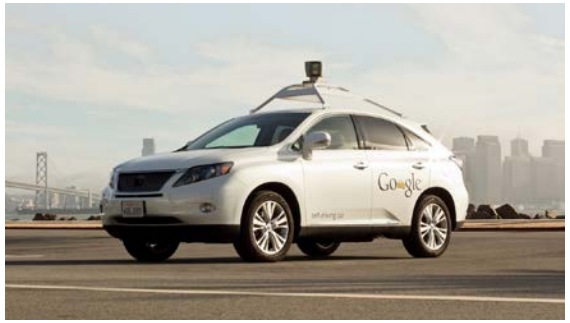
Services

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

…

Google Now Cortana Siri amazon echo Facebook M



NETFLIX

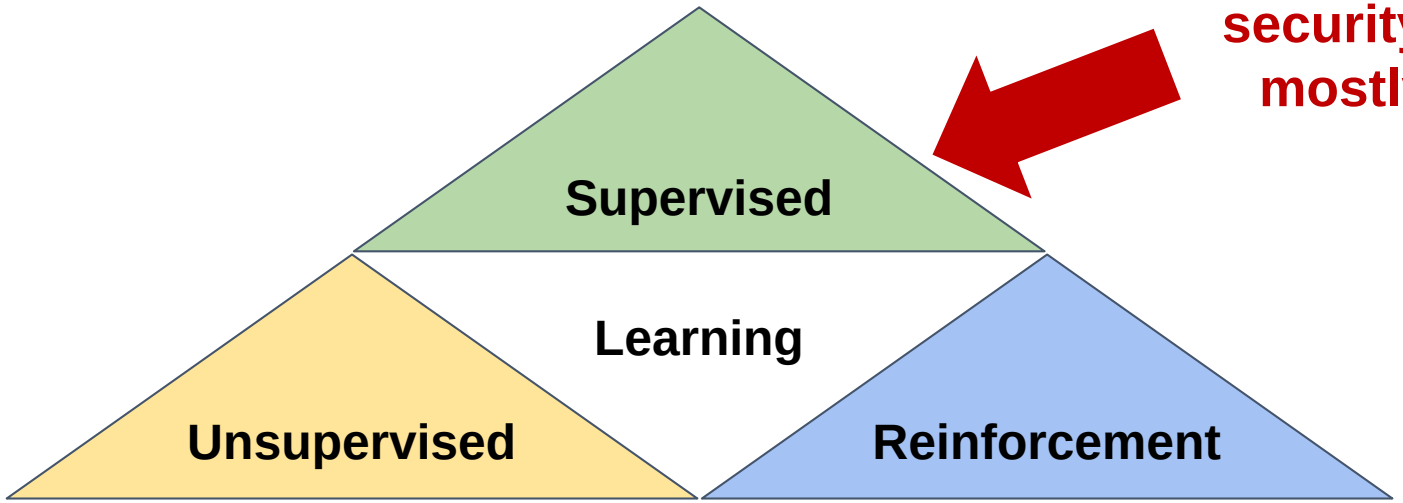
Source: https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai

Machine Learning Taxonomy

Machine learning can be separated into 3 main categories

- Labeled data
- Direct feedback
- Predict outcome/future

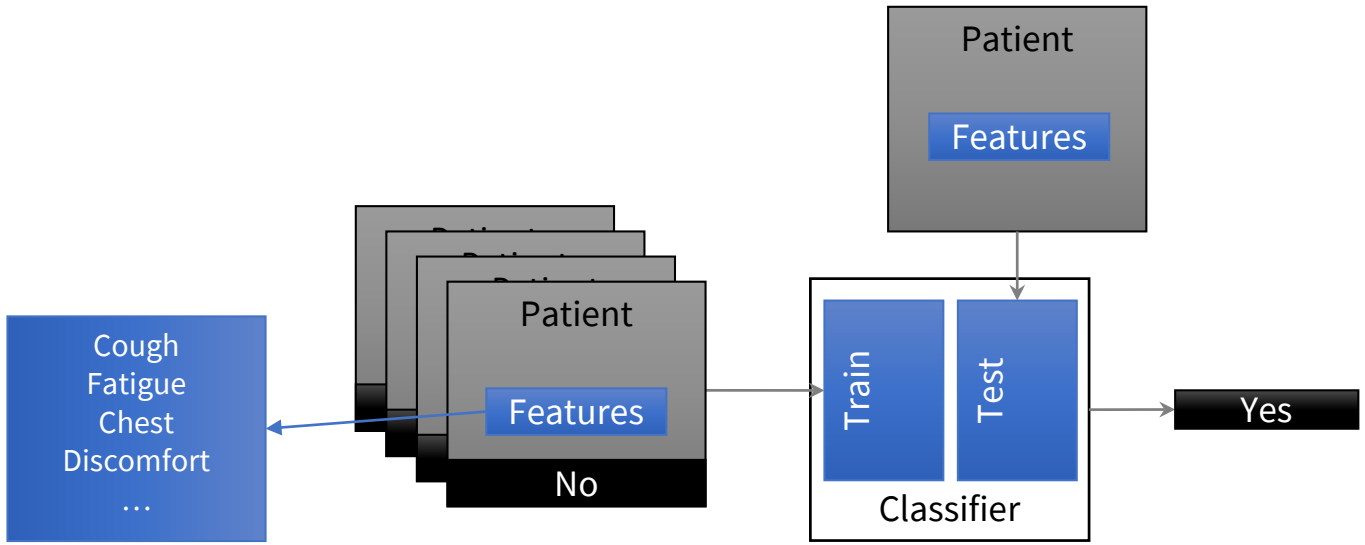
Most used and where security has been mostly studied



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

- Decision process
- Reward system
- Learn series of actions

Supervised Learning

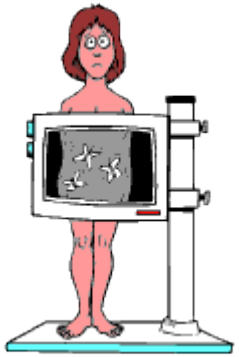


More formally

- Given: Training data $\{(x_1,y_1),\ldots,(x_n,y_n)\} \in X \times Y$
  - 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 ($Y = \mathbb{R}$)

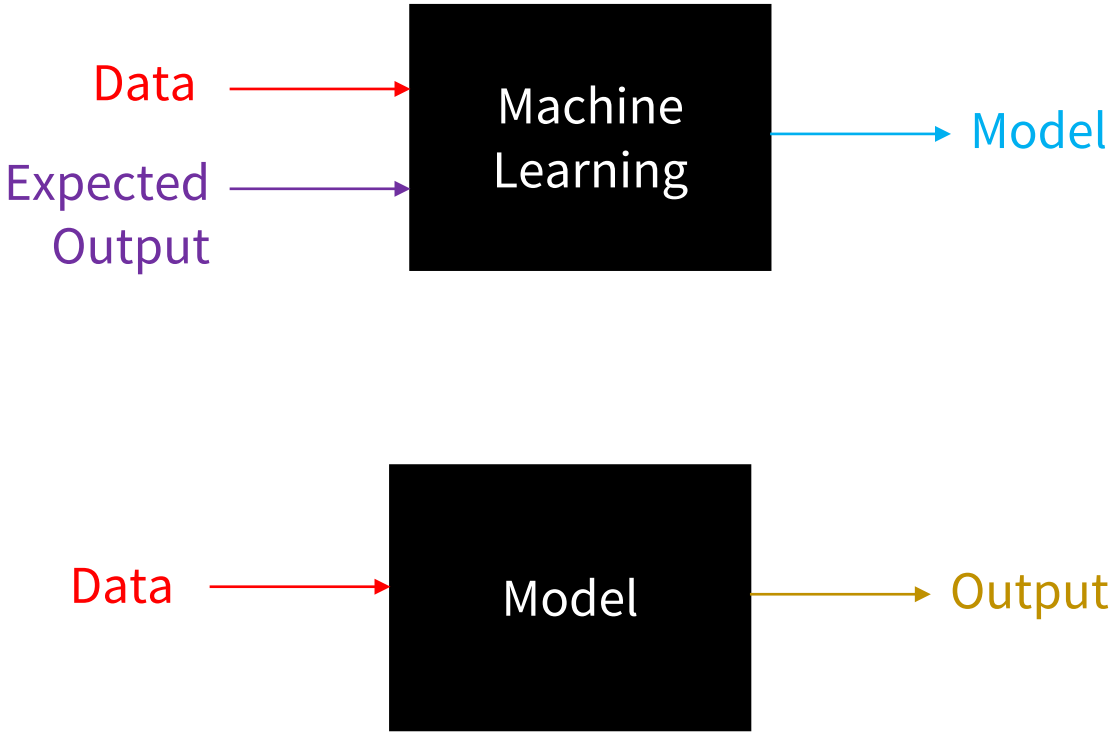
This lecture

- 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?



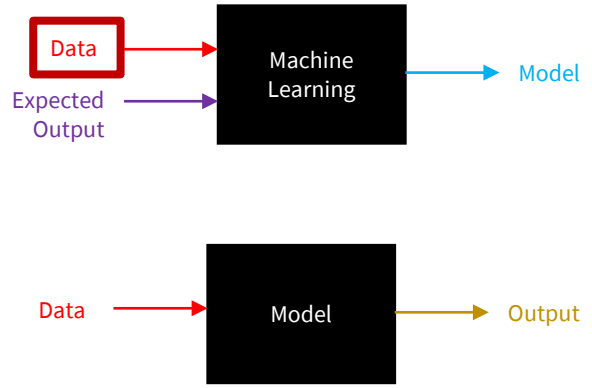
1. **Data Mining**: Get data that you know something about
2. **Train the Model**: "Teach" the machine about that data
3. **Test the model**: "Check" that the machine on something you know
4. **Deploy** the program!: Use the learnings to classify/predict on new data

Privacy problem #1: ML needs data to learn!

Machine learning is based on data to

Find features

Train the model



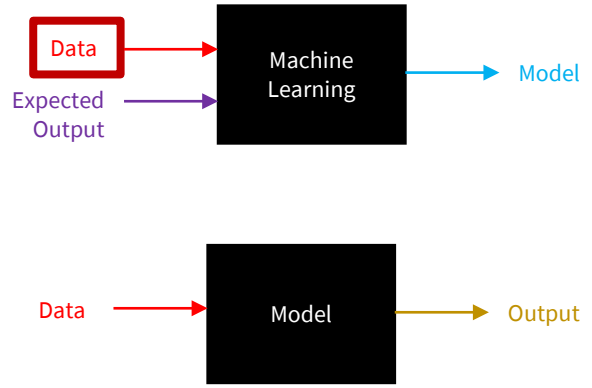
Data is highly unique! Allows many inferences

Let’s anonymize…

What about aggregation?

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

Privacy problem #1: ML needs data to learn!



Machine learning is based on data to

Find features

Train the model

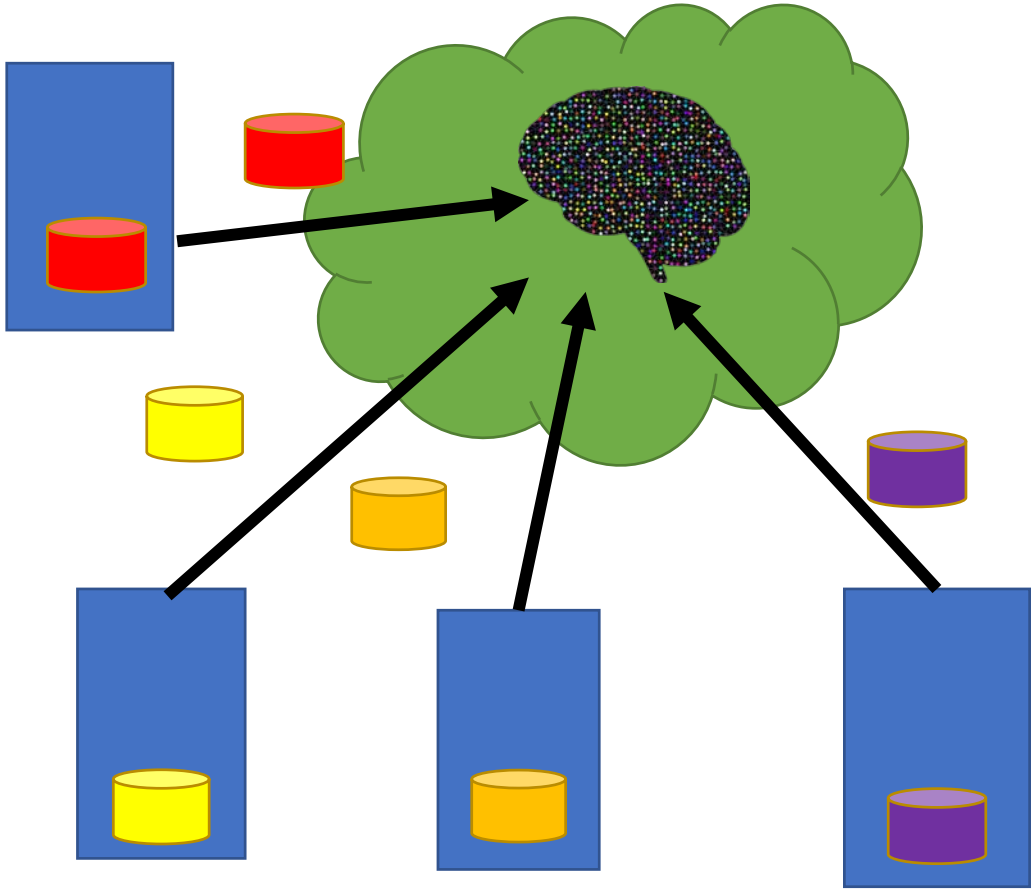
Data is highly unique! Allows many inferences

Anonymizing may not work…

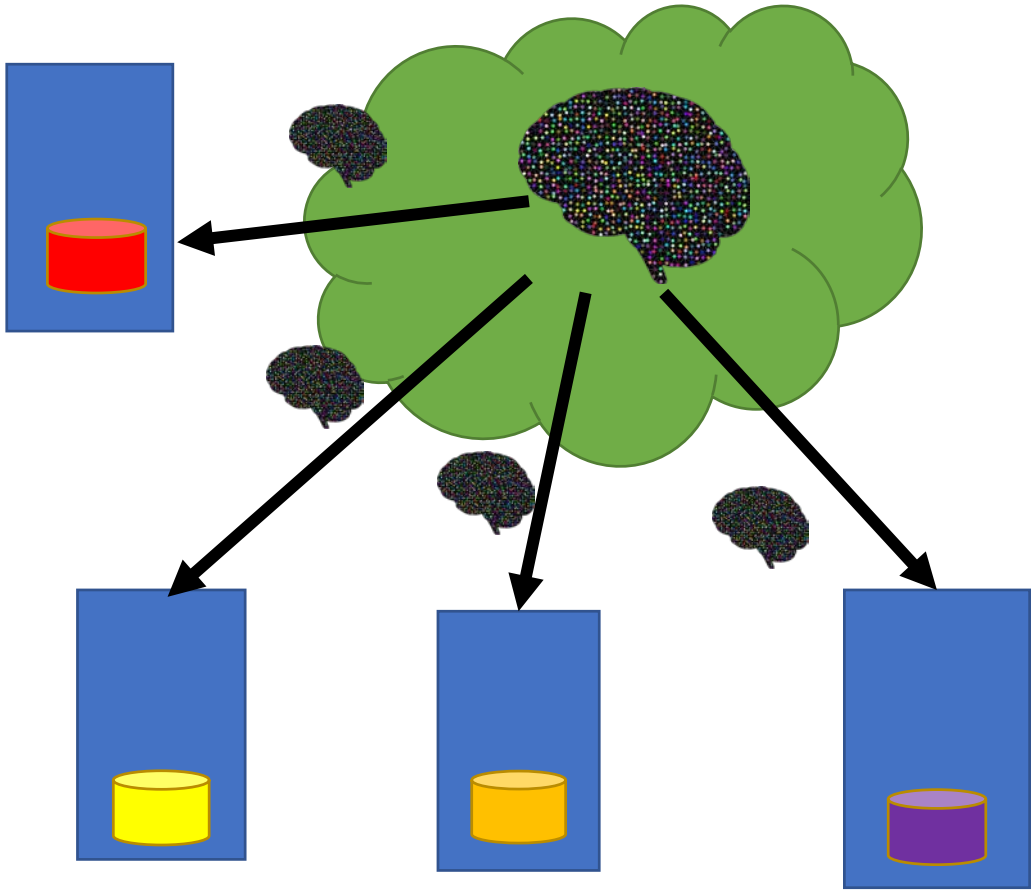
Aggregation affects utility and requires careful evaluation

Let’s collect noisy data

Learning without privacy

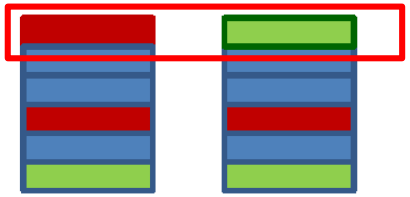


Learning without privacy (optional step)



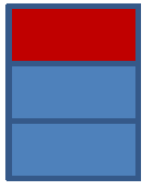
Differential Privacy - Intuition

For every pair of inputs that differ in one value



D_1 D_2

For every output …

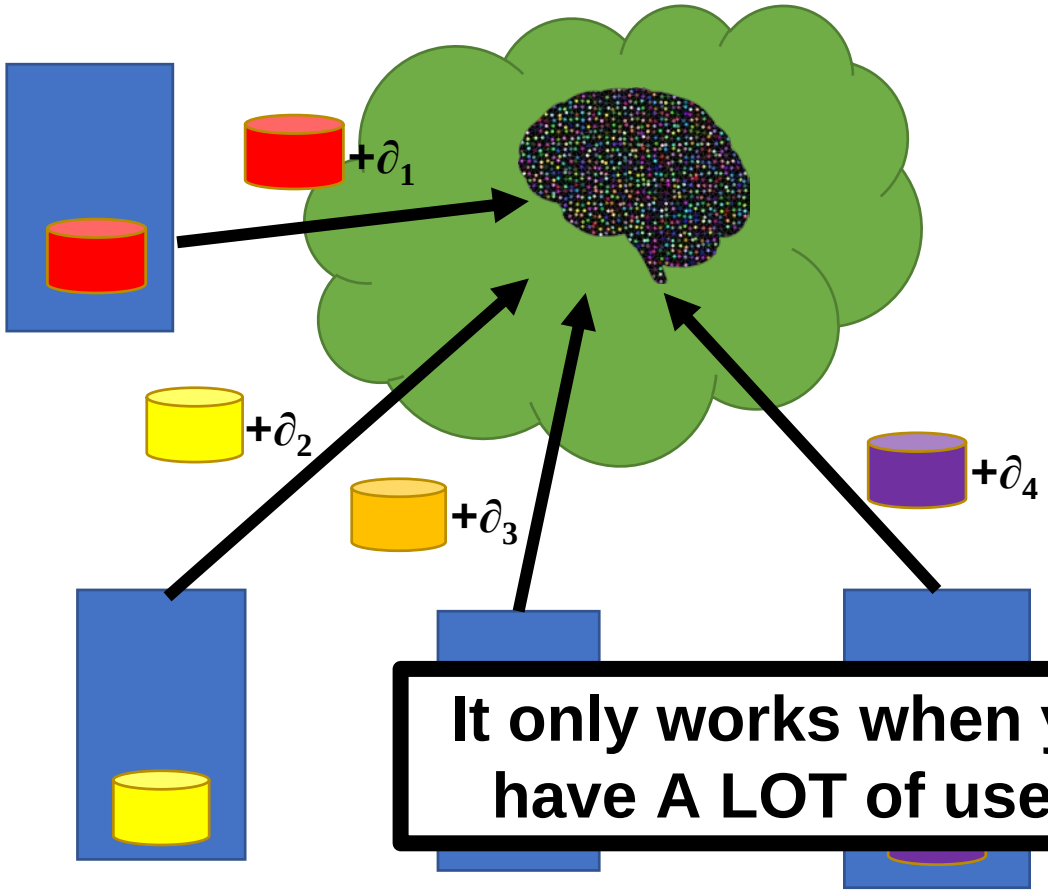


O

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

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

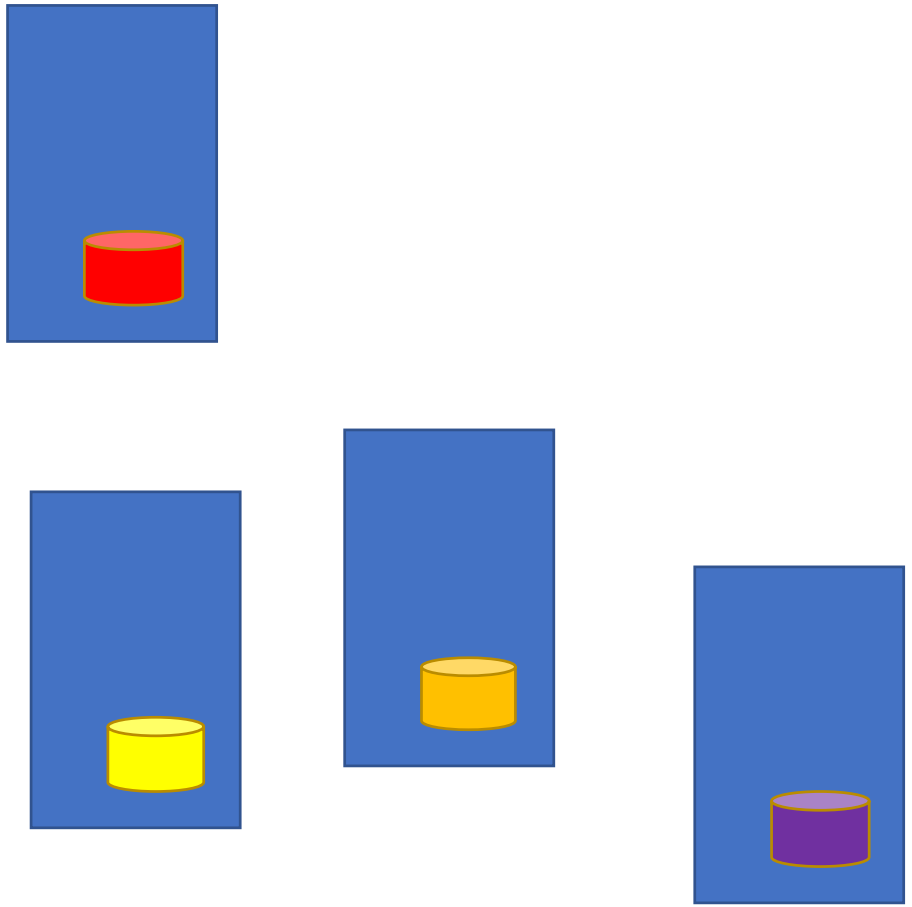
Learning with privacy



- 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

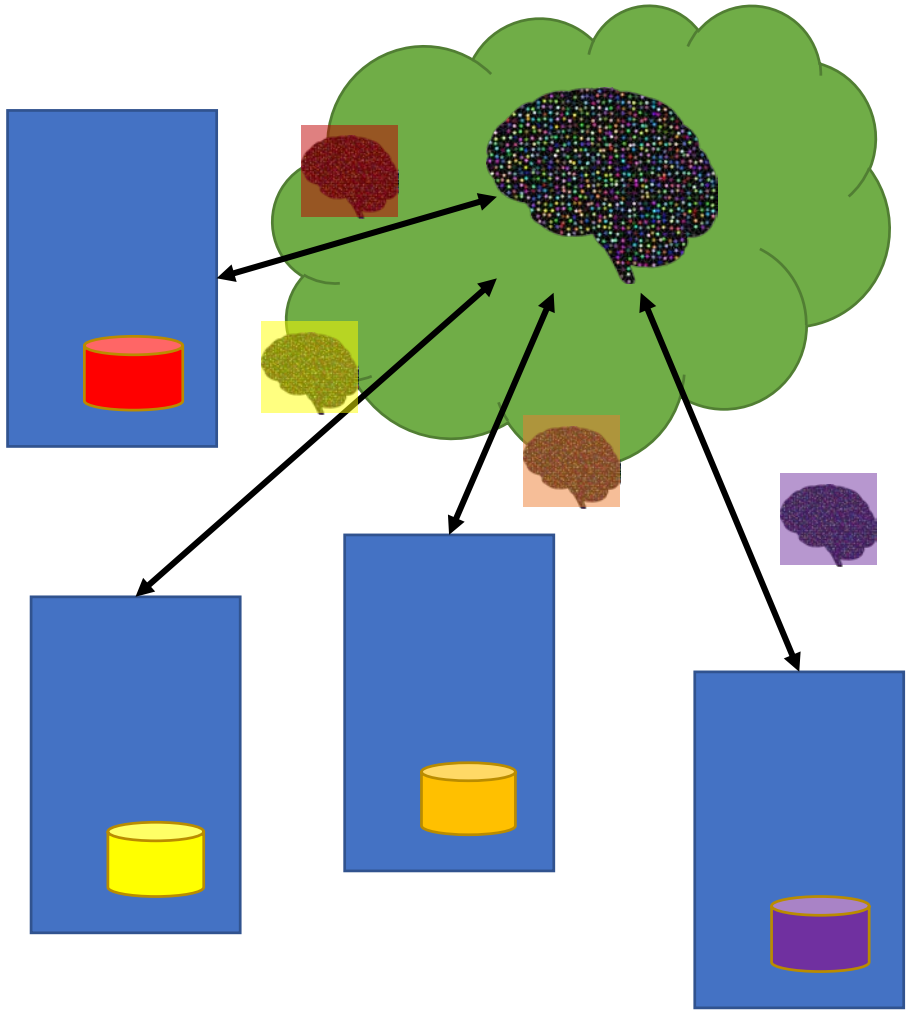
Google RAPPOR - Collect data from phones
Apple - Collect data from phones
Federated learning - Share models
Smart energy - Collect measurements

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??

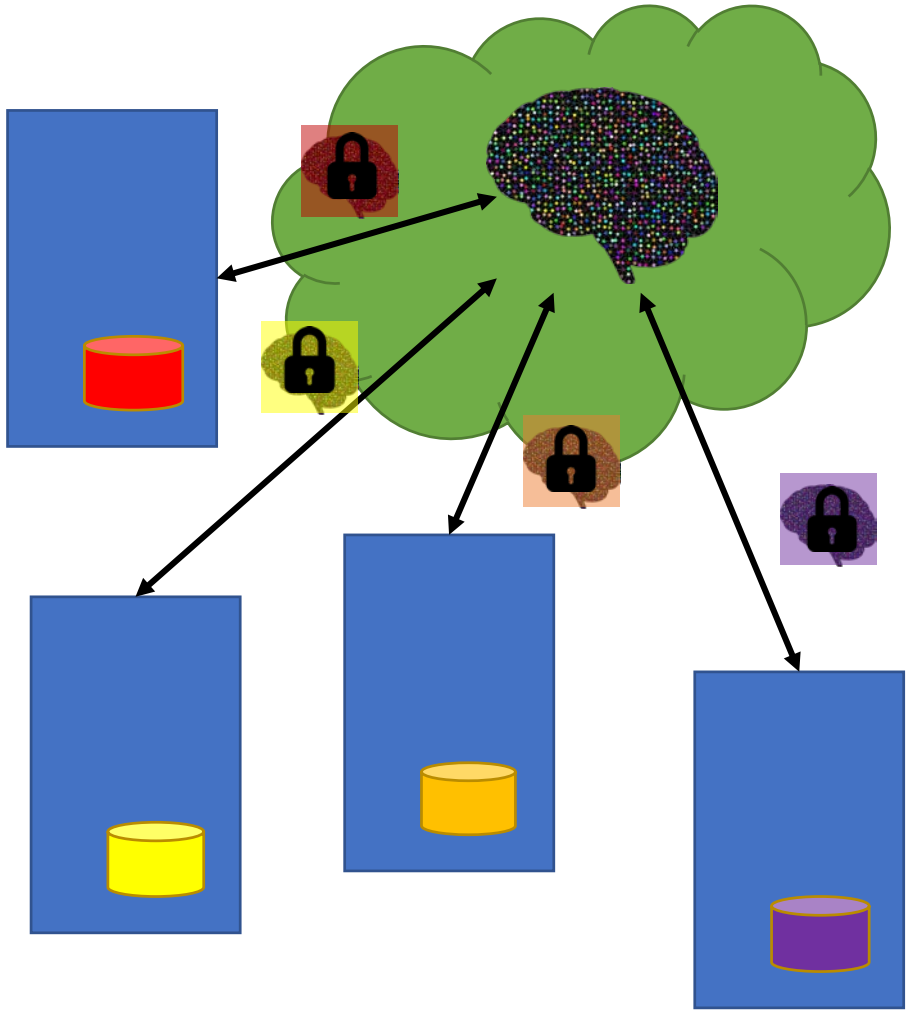
Federated Learning – Learn as a team



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 – **Solutions?**



Option 1: Encrypt!

1.A: Homomorphic encryption (risk: 👵🏻 👴🏽)

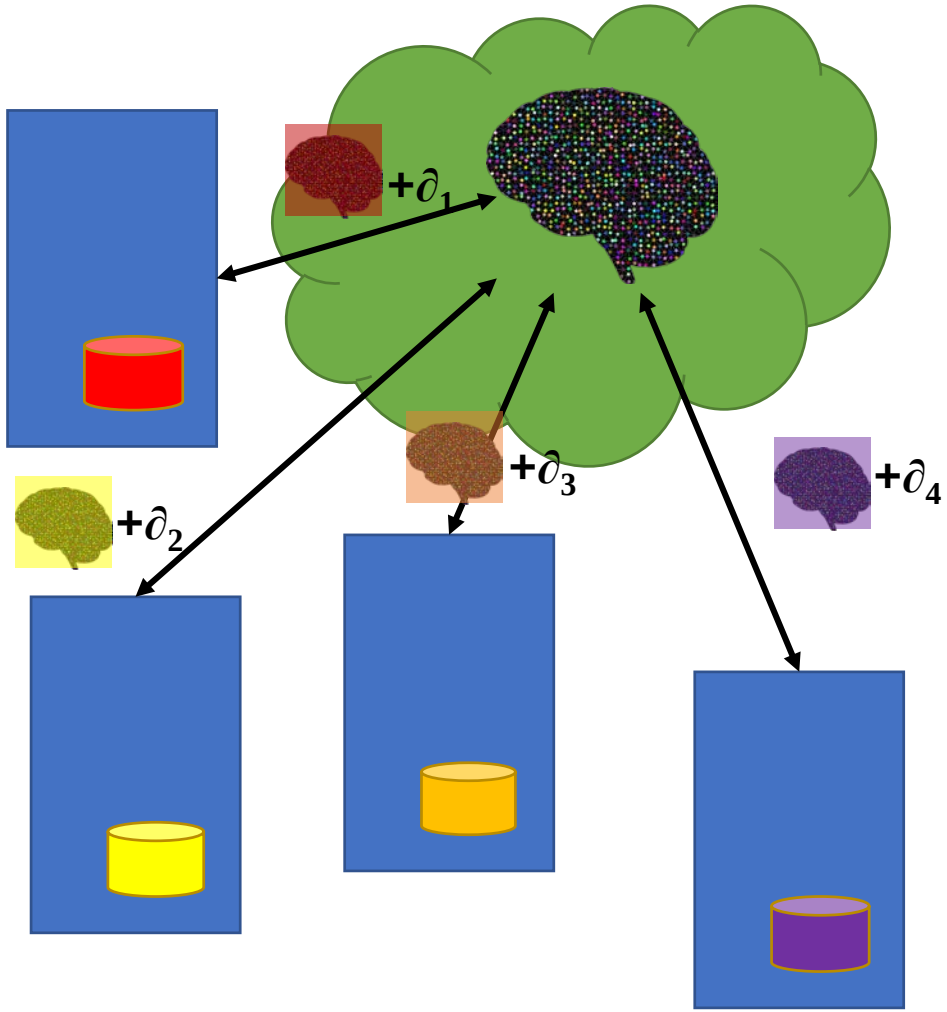
1.B: Multi party computation

All parties blind their input

Blinding factors cancel

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. Practical secure aggregation for privacy-preserving machine learning. CCS 2017.

Federated Learning – **Solutions?**



Option 2: Differential privacy

Before sending models, add noise

Actually, add noise to the gradient

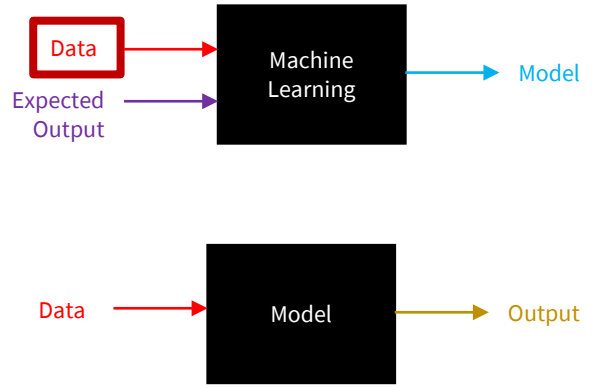
Tradeoff privacy vs functionality

Some attacks still possible

Shokri, R., & Shmatikov, V. Privacy-preserving deep learning. CCS 2015

L.Melis, C.Song, E. De Cristofaro, V.Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning S&P 2019

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!

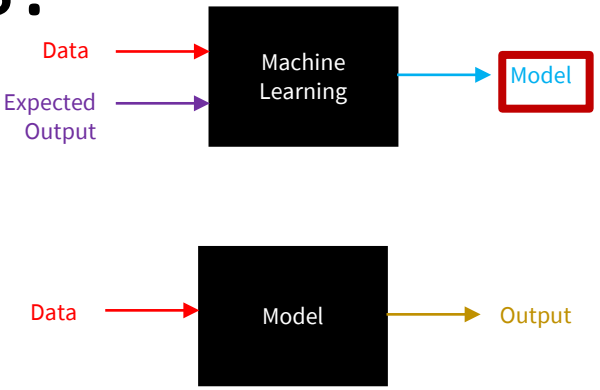
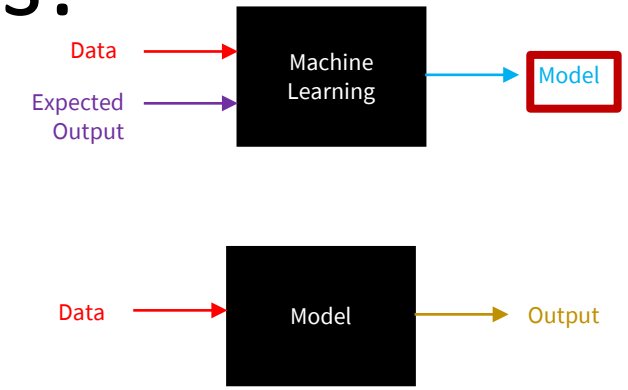


Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

https://www.cs.cmu.edu/~mfredrik/papers/fjr2015ccs.pdf

Privacy problem #2: The model remembers!



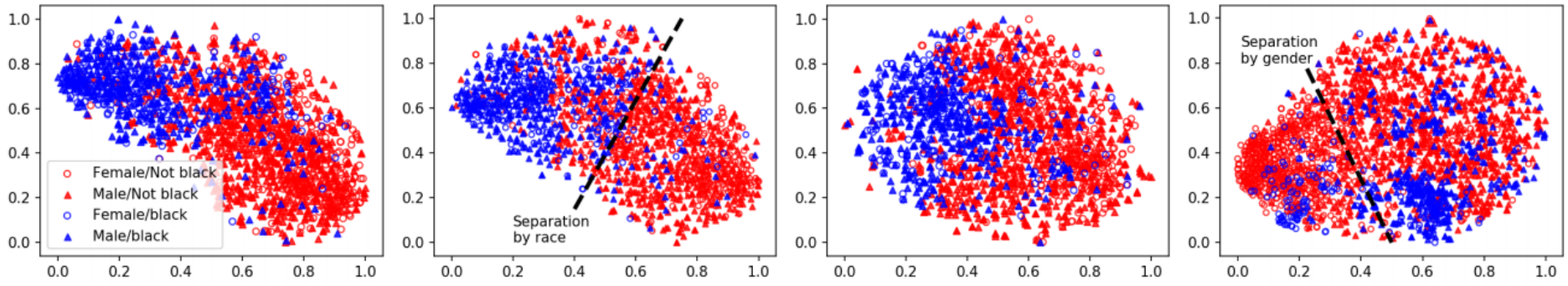
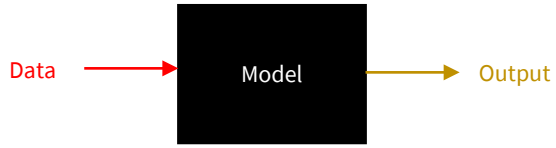


Figure 3: t-SNE projection of the features from different layers of the joint model on LFW gender classification; hollow circle point is female, solid triangle point is male, blue point is the property "race: black" and red point is data without the property.

Exploiting Unintended Feature Leakage in Collaborative Learning Luca Melis, Congzheng Song, Emiliano De Cristofaro, Vitaly Shmatikov. IEEE S&P 19

Privacy problem #2: The model remembers!

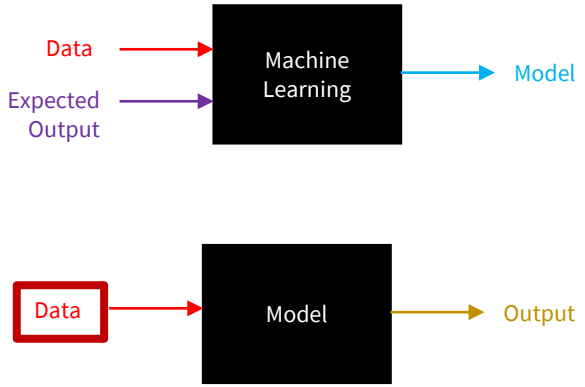


Secret	Log-Perplexity
The random number is 281265017	14.63
The random number is 281265117	18.56
The random number is 281265011	19.01
The random number is 286265117	20.65
The random number is 528126501	20.88
The random number is 281266511	20.99
The random number is 287265017	20.99
The random number is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

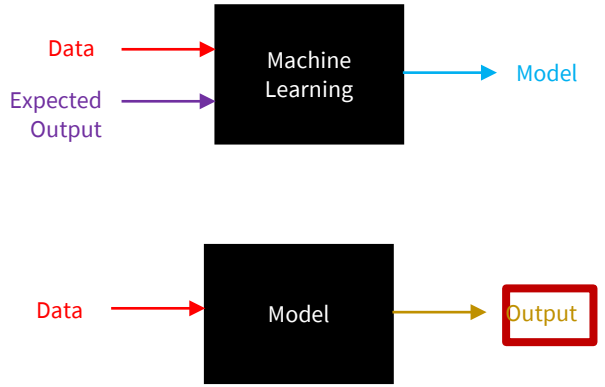
Table 1: Secrets sorted by Log-Perplexity. The inserted canary — 281265017 — has the lowest log-perplexity. The remaining secrets are all variants with slight modifications (i.e., by changing a few characters).

The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. Nicholas Carlini, Chang Liu, Jernej Kos, Ulfar Erlingsson, Dawn Song. 2018.

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



Privacy problem #4:
The output reveals information!

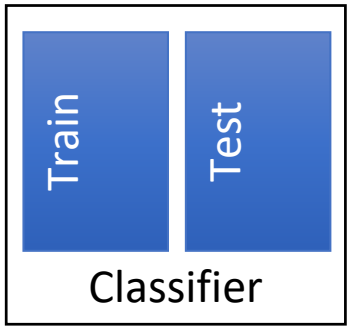


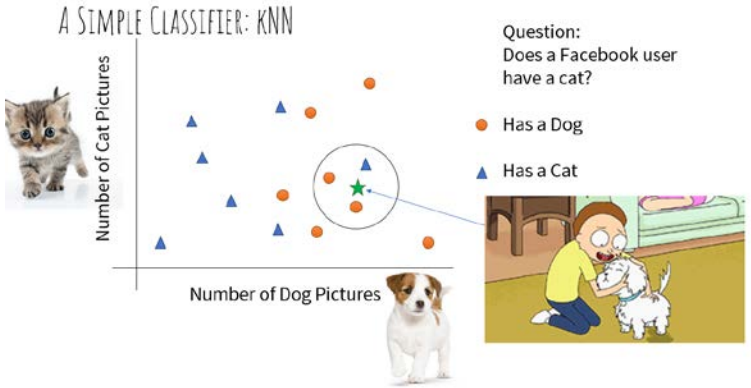
Membership inference

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

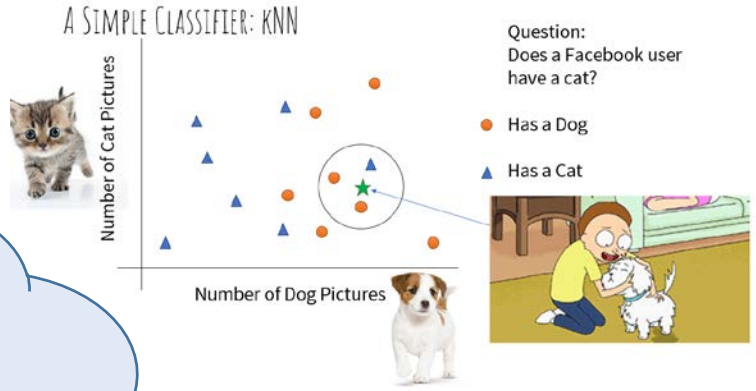
Machine Learning as a Service

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

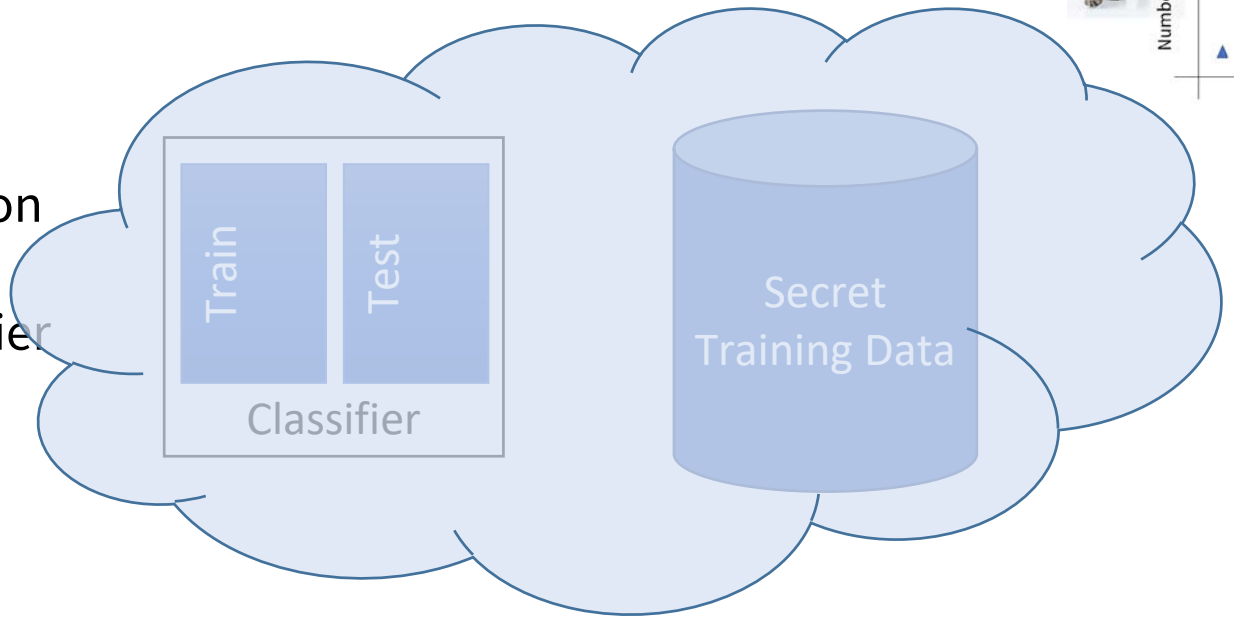




Machine Learning as a Service

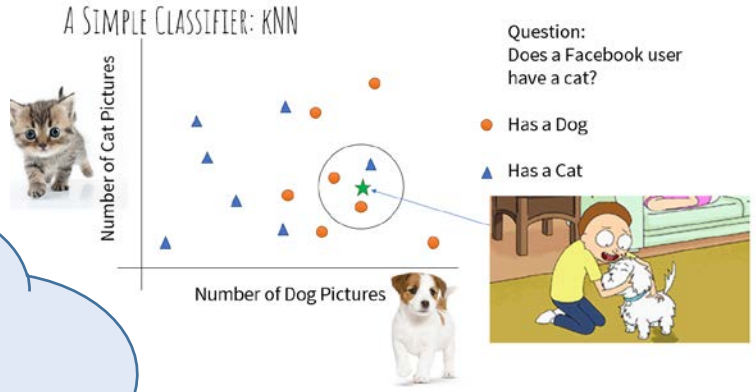


1. The Cloud, e.g., Amazon ML or Google Prediction API, (pre-)trains a classifier using their own data

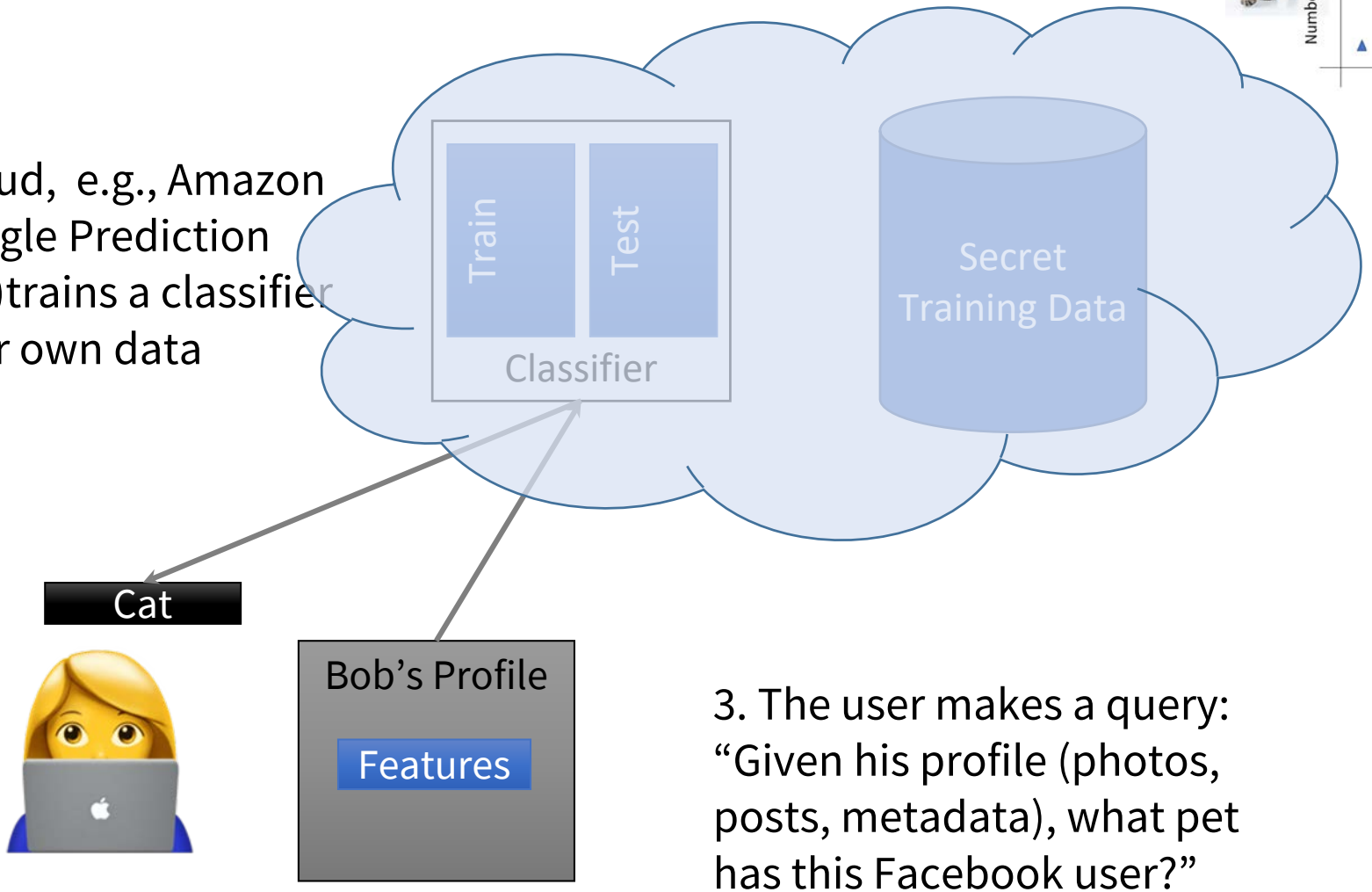


2. Make this classifier available as a service for users to query

Machine Learning as a Service



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

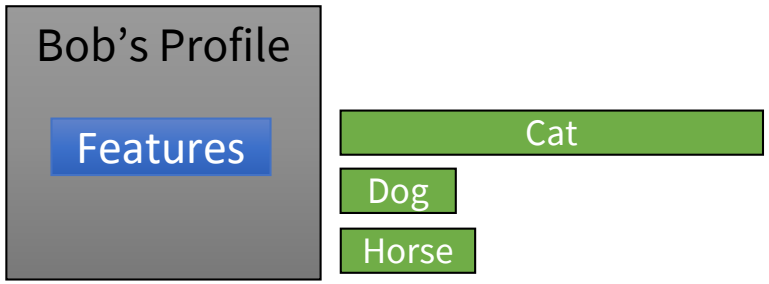


2. Make this classifier available as a service for users to query

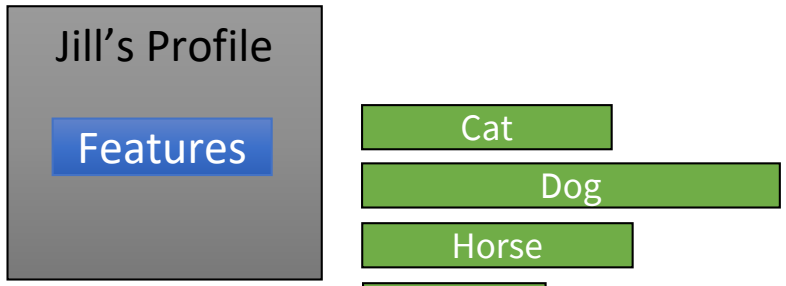
3. The user makes a query: "Given his profile (photos, posts, metadata), what pet has this Facebook user?"

Membership inference

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



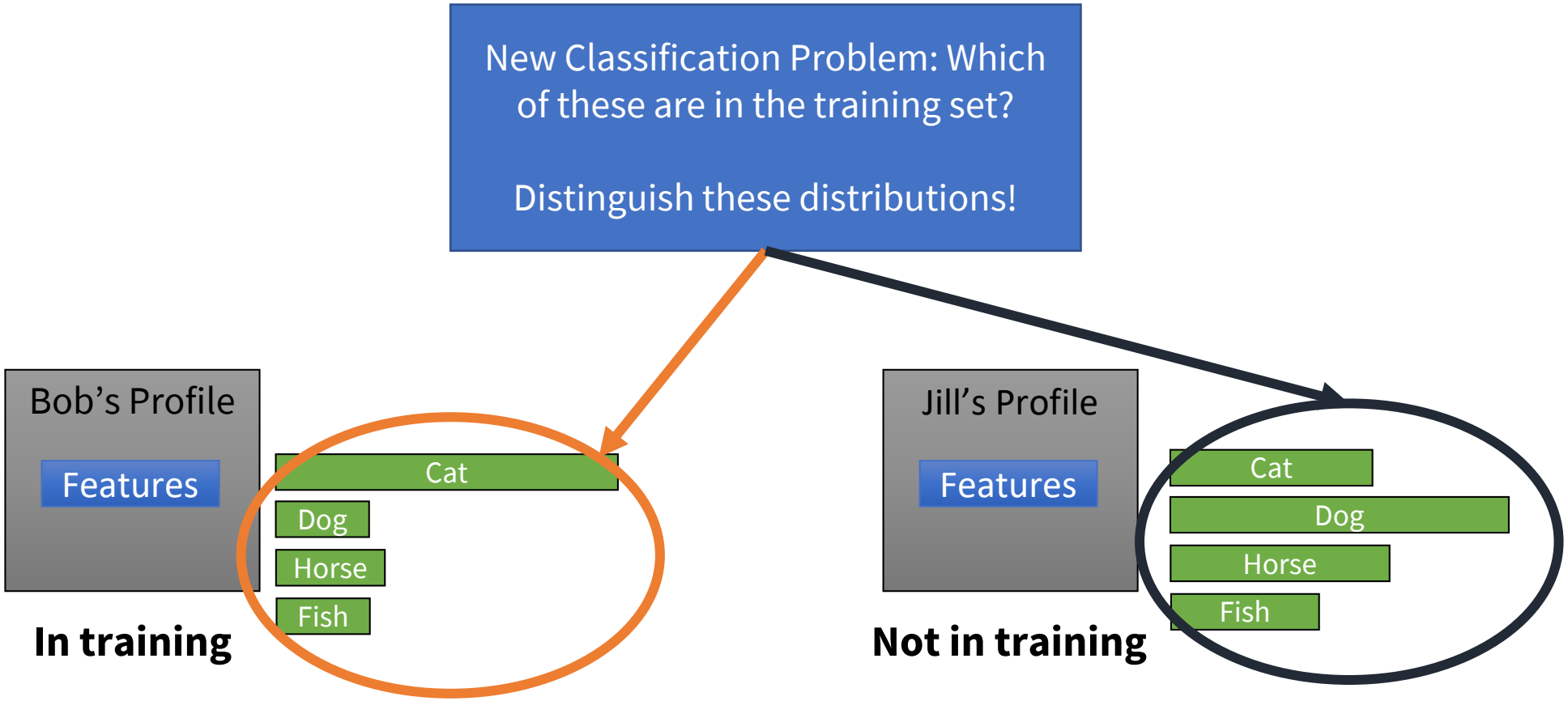
In training



Not in training

Membership inference

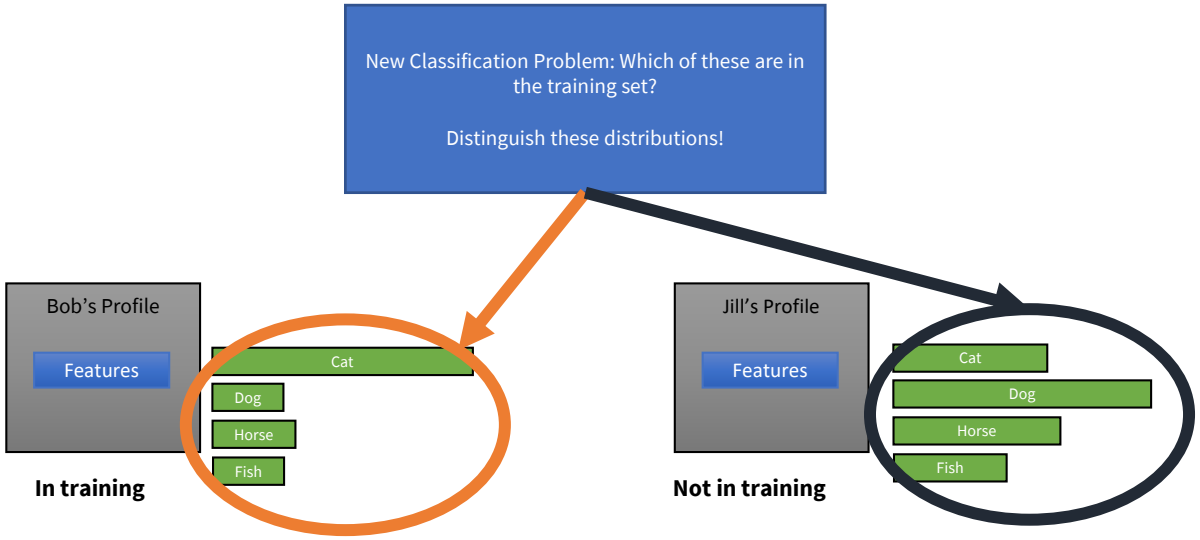
In reality, the distributions are different!



Shokri et al. "Membership inference attacks against machine learning models." IEEE Security and Privacy, 2017.

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

Shokri et al. "Membership inference attacks against machine learning models." IEEE Security and Privacy, 2017.

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

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

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

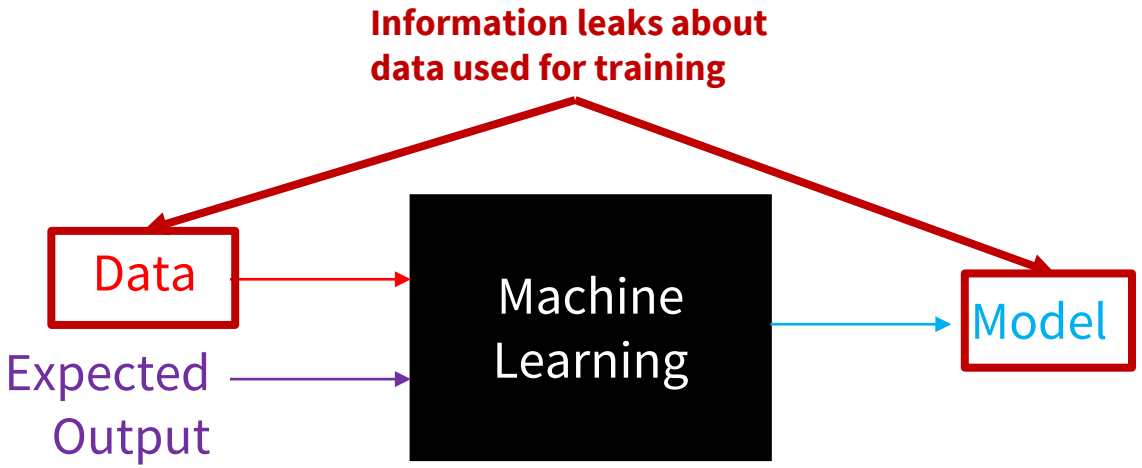
The ML model can be used to breach privacy of that new data (or associated entities)

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

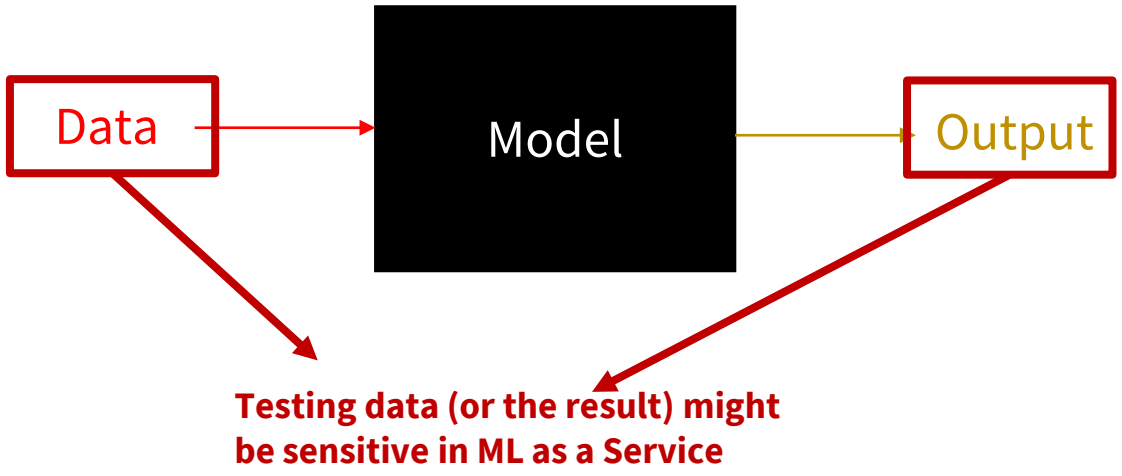
Deploy the program!: Use the learnings to classify/predict on new data

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

Takeaways



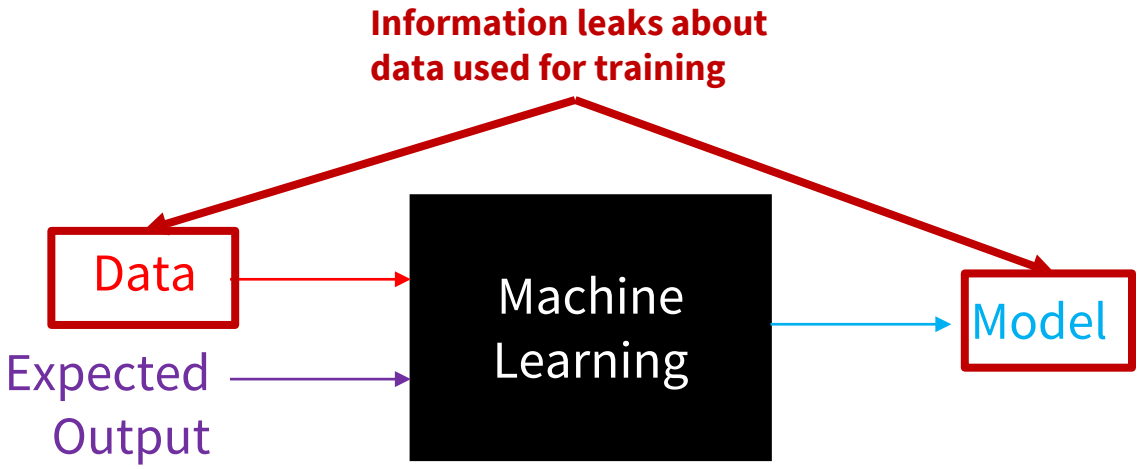
Many problems, some solutions ☺

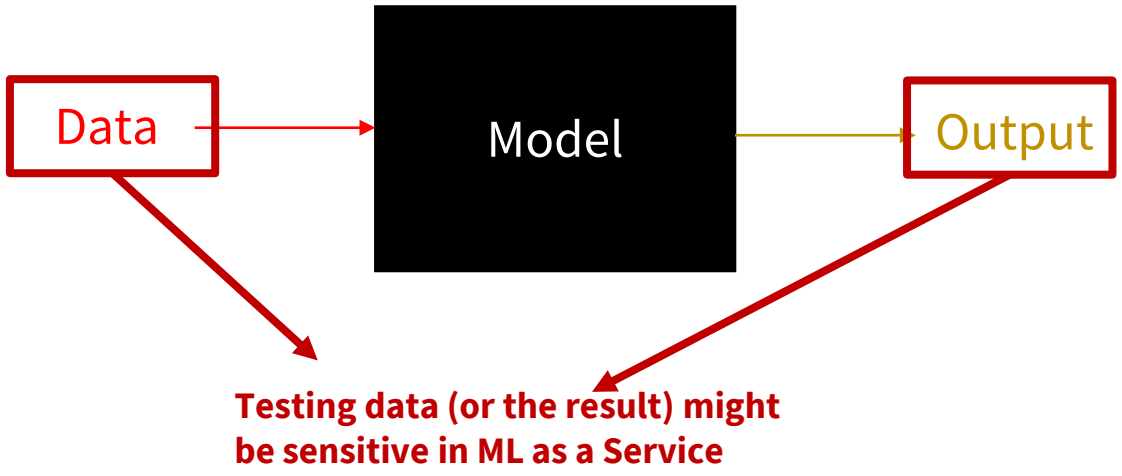


Deploy the program!: Use the learnings to classify/predict on new data

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!

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

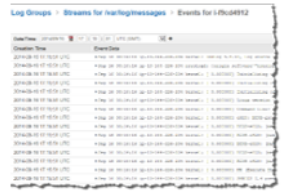
GOALS – Compliance with data protection principles

informed consent
purpose limitation
data minimization
subject access rights

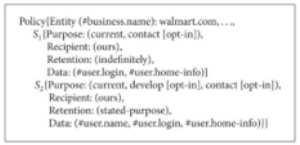
Preserving the security of data
Auditability and accountability

Access control

Anonymization????



Logging



Automated Policy Negotiation

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

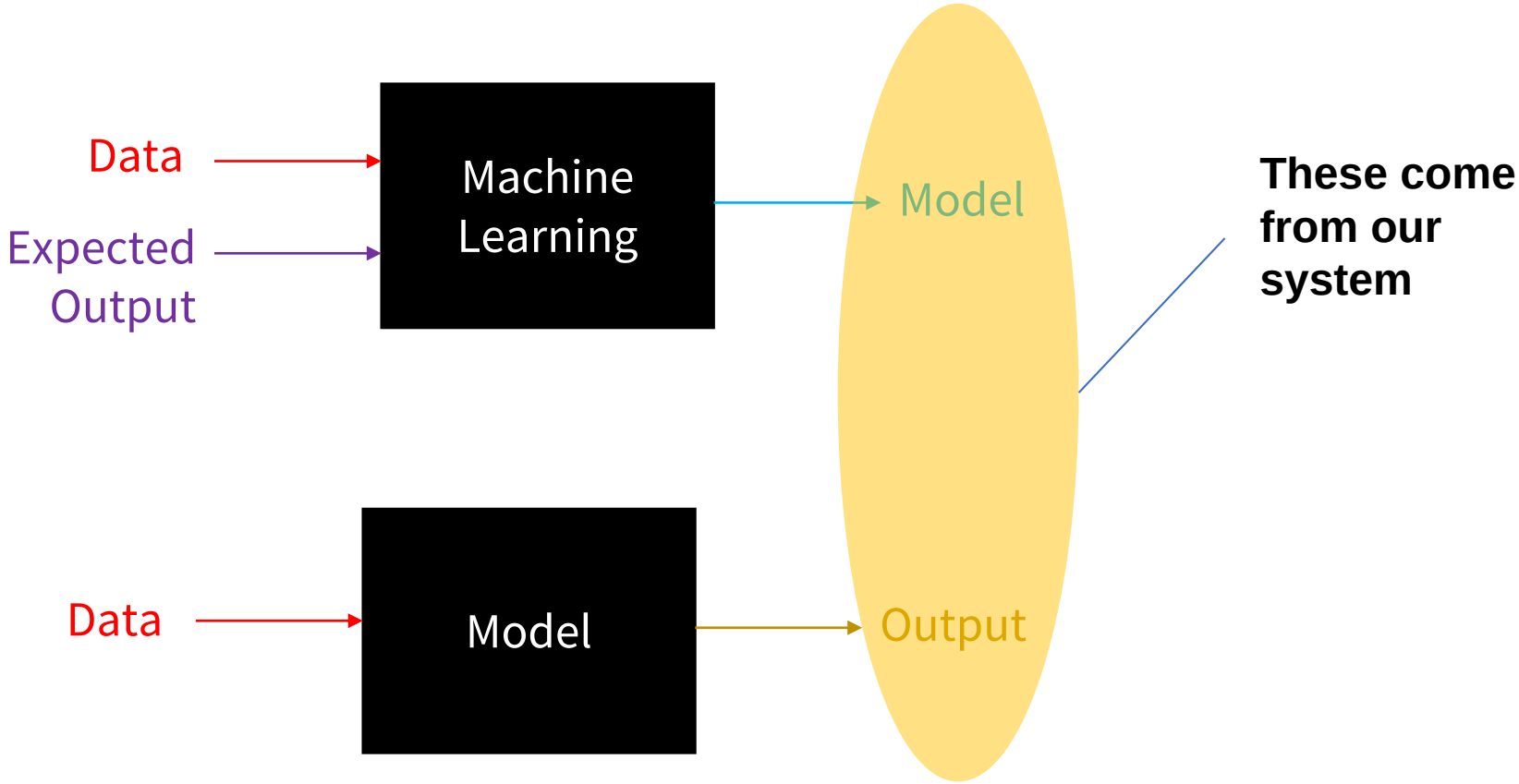
Extremely hard to enforce when ML is pervasive. Where / how are you deploying?

This lecture

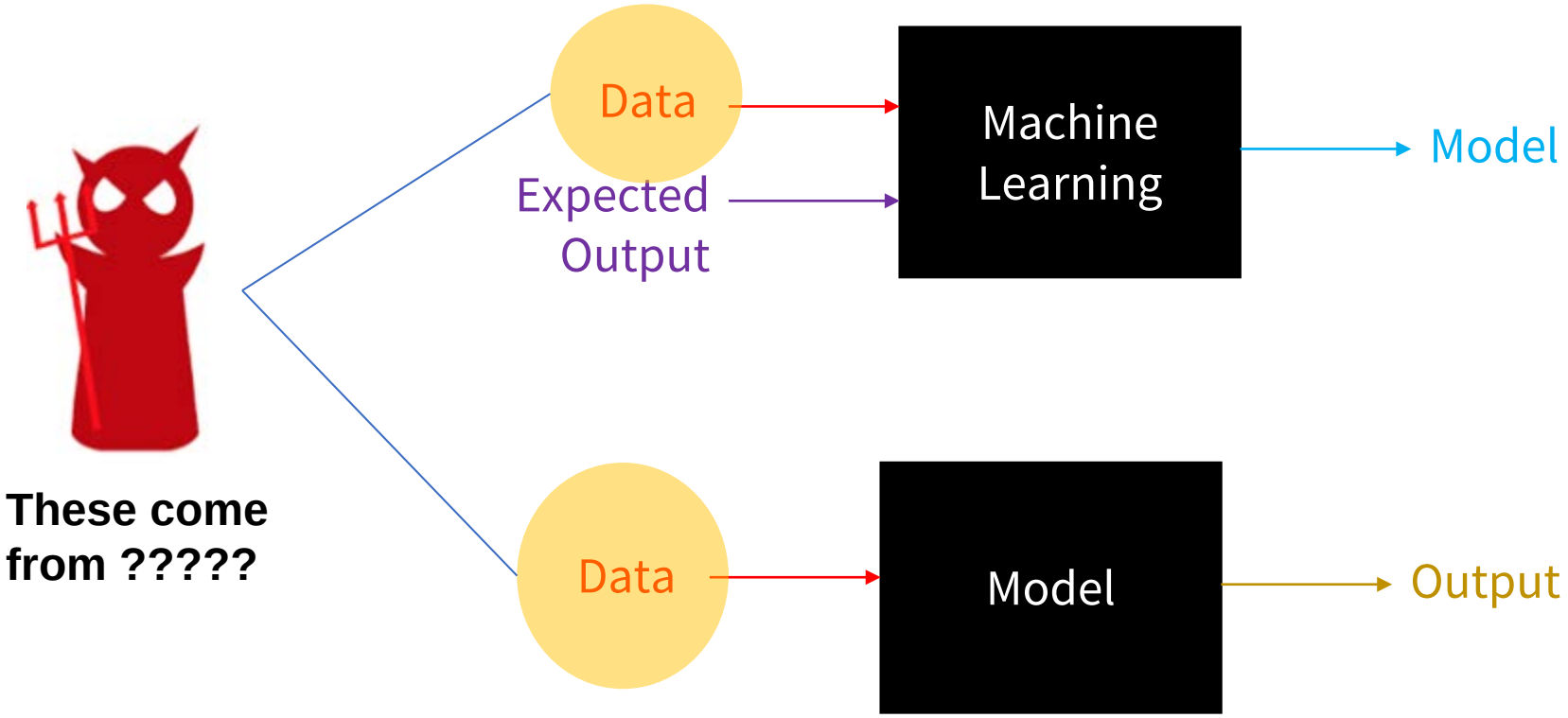
- Machine Learning – basics
- Privacy implications of Machine Learning
- **Machine learning under adversarial conditions**
- Issues with applying Machine Learning to Security and Privacy problems

Attacks on machine learning

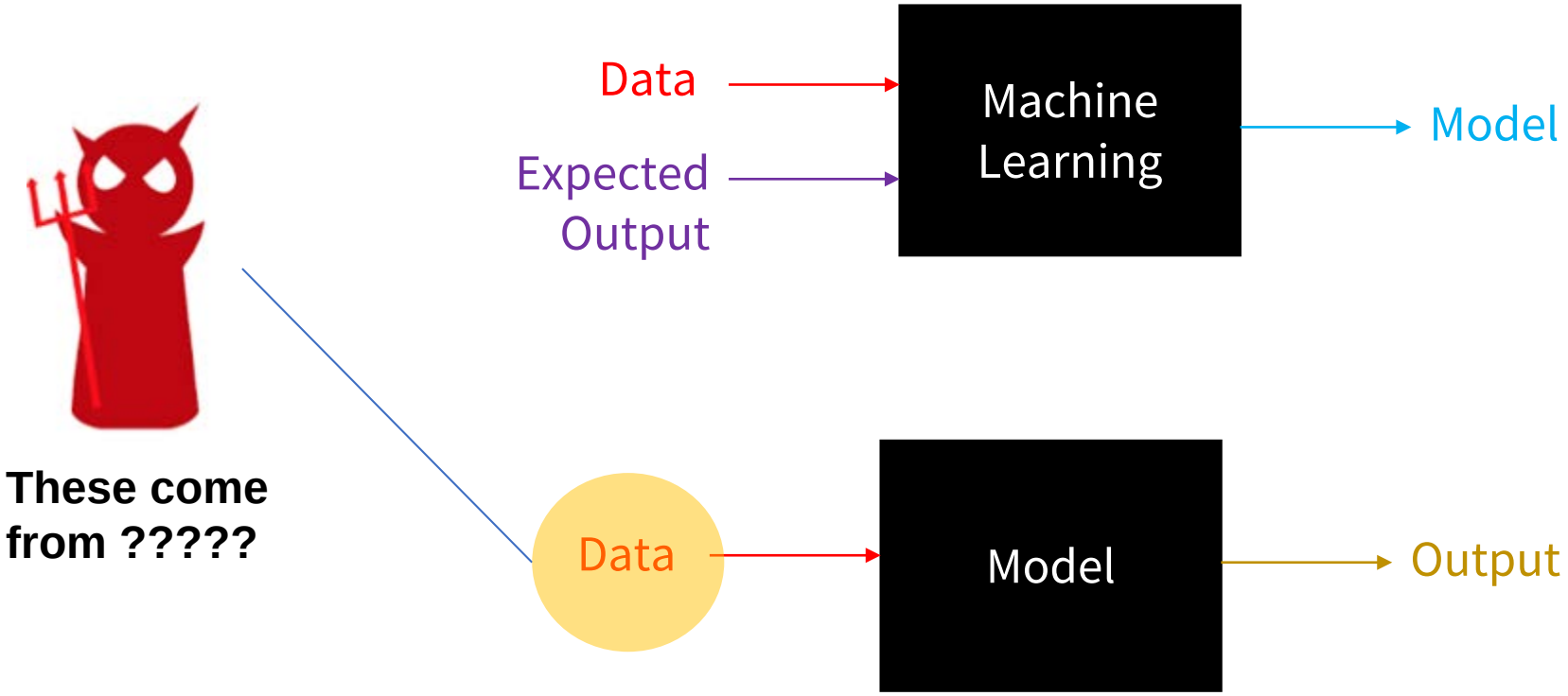
Issues with deploying ML Systems In The Wild



Issues with deploying ML Systems In The Wild



Issues with deploying ML Systems In The Wild

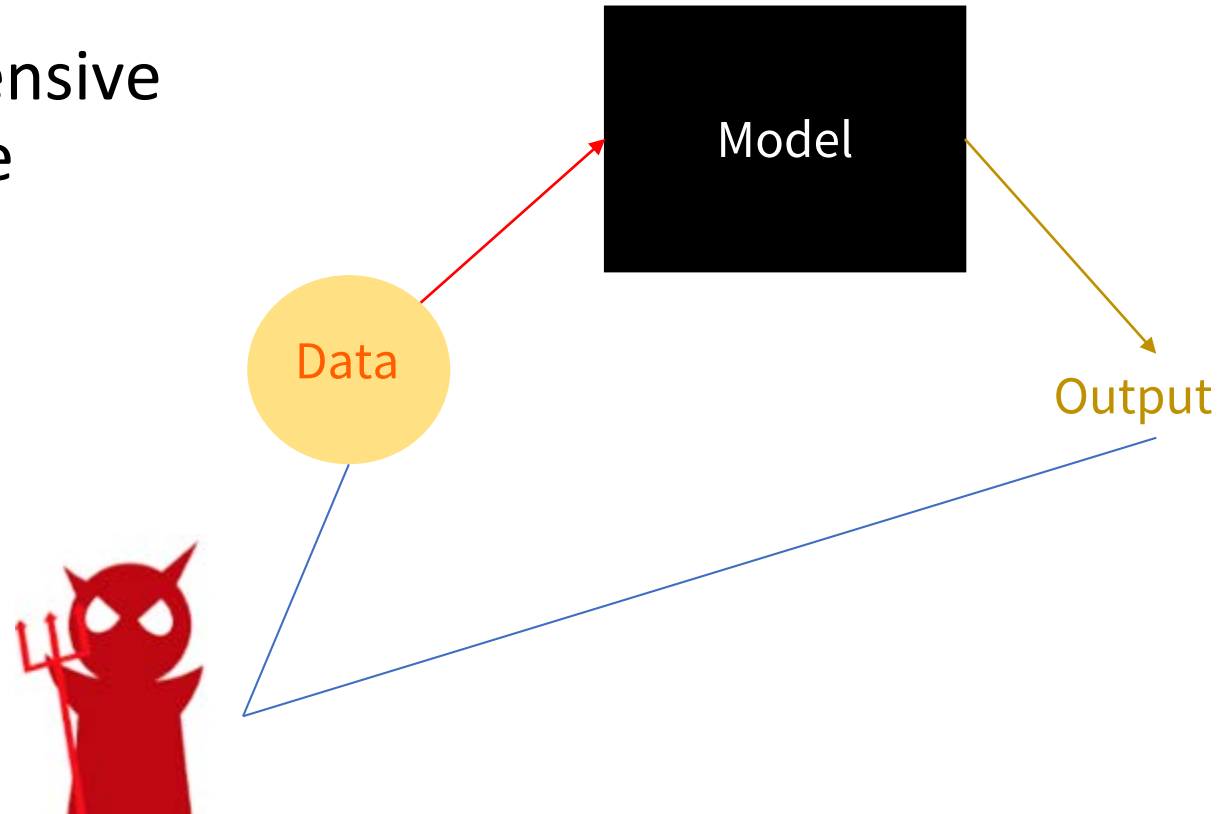


Model stealing

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

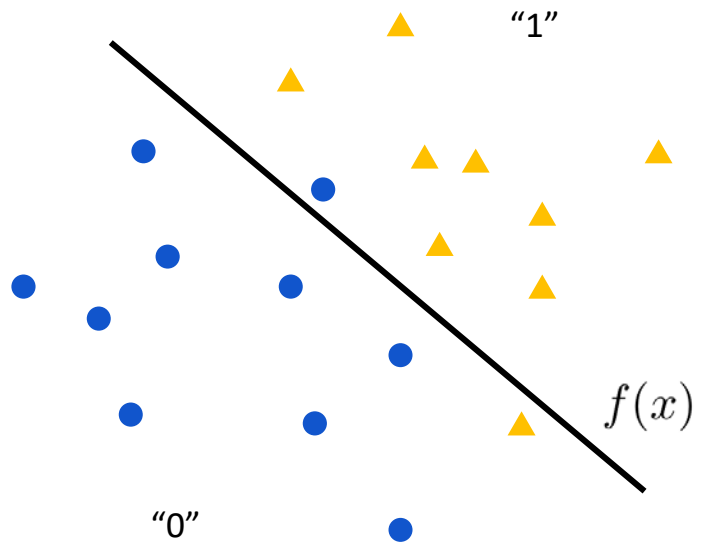
Linear models

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

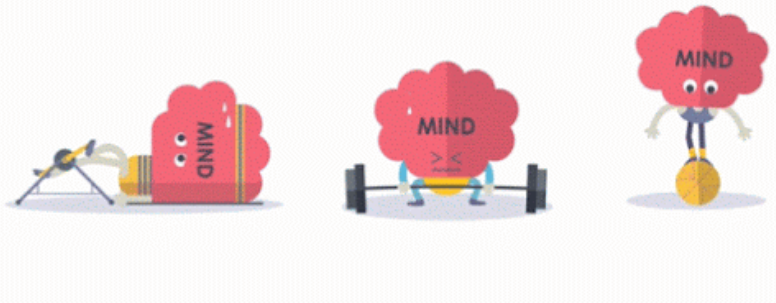
- E.g., logistic regression, SVM with linear kernel

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

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).

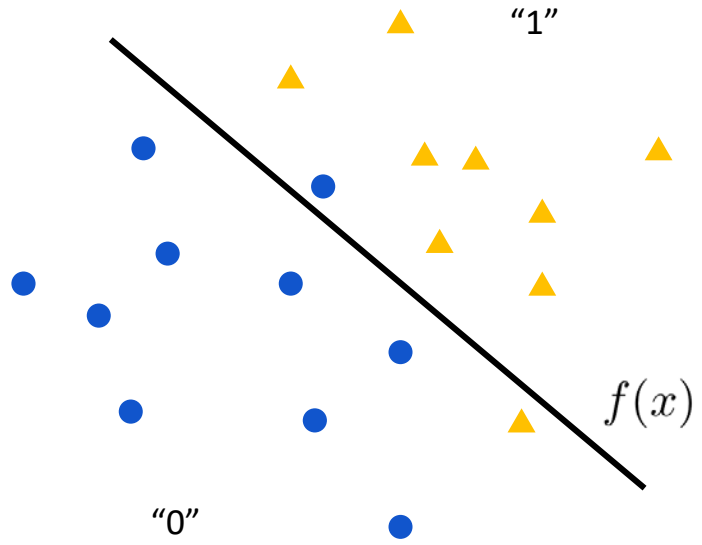
Assume x is **two**-dimensional:

$$f(x) = w_1 x_1 + w_2 x_2 + b$$

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?



Retraining attack

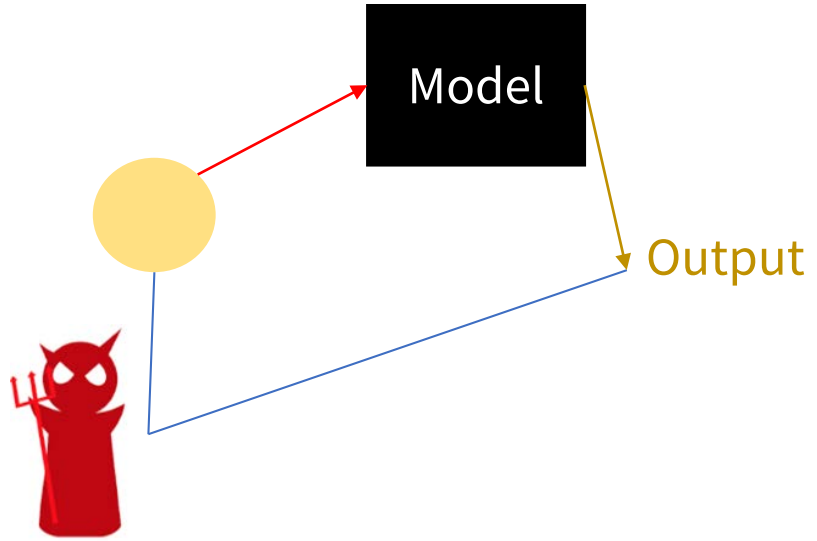
Assume adversary knows the model's architecture:

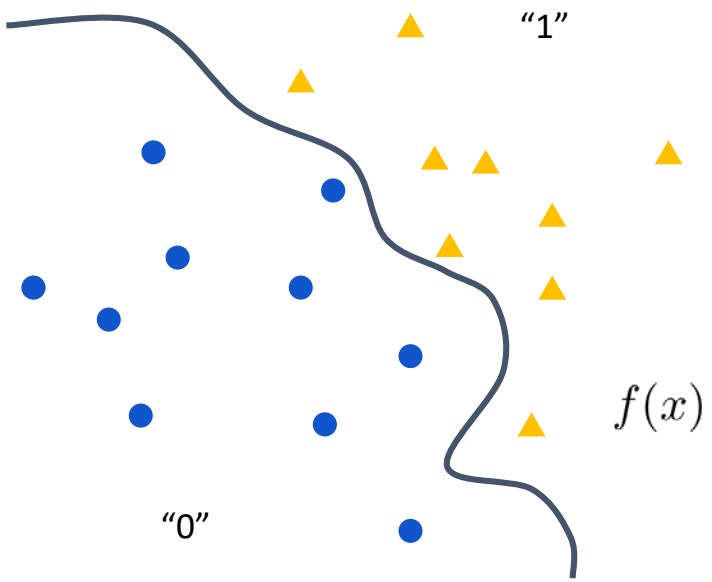
$$f_w(x)$$

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”.

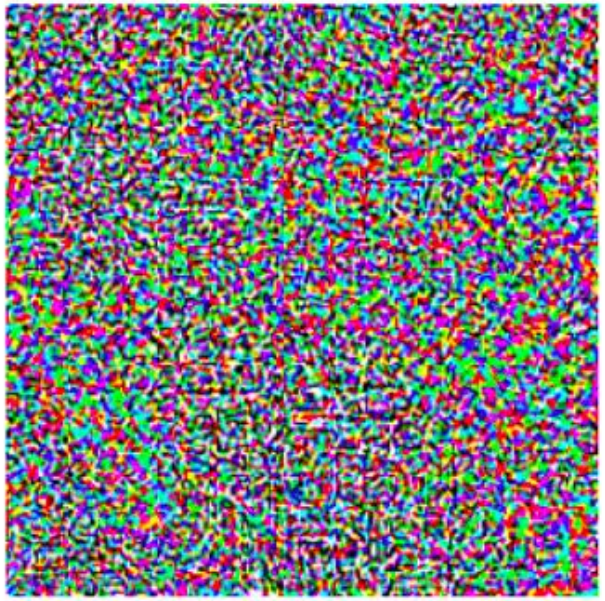
> *The phrase was coined by Ian Goodfellow in 2014 [1] with respect to computer vision problems, but goes back further to work on spam and malware evasion done by Srndic & Laskov [2], Biggio et al. [3], and Dalvi et al. [4].*

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.

[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples

Quick ML refresher I

Objective: find model parameters that *minimize* empirical loss

$$w^* = \arg\min_w \mathbb{E}_{x,y\sim D}[L(x, y; w)]$$

$$= \arg\min_w \sum_{x,y\in X} L(x, y; w)$$

Data distribution

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

Training data

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

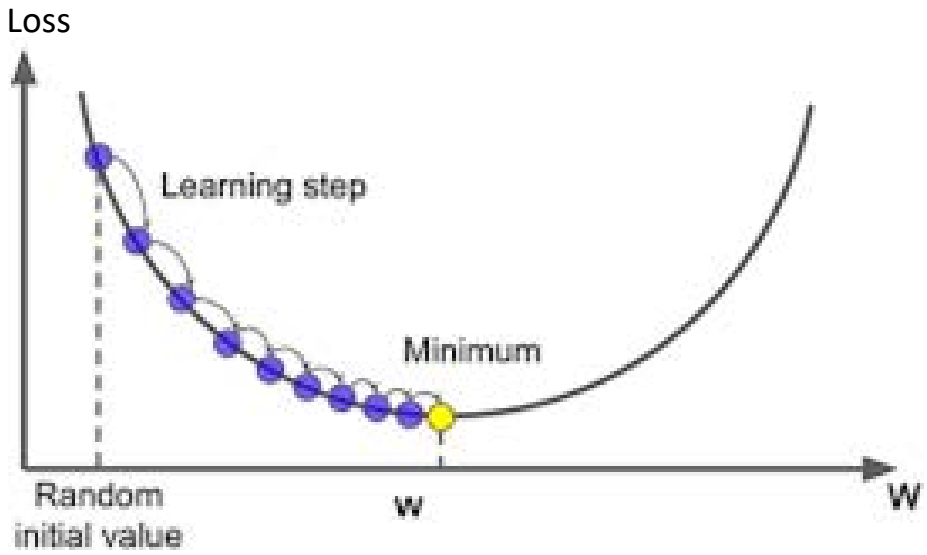


Image: Saugat Bhattarai

Adversarial example problem

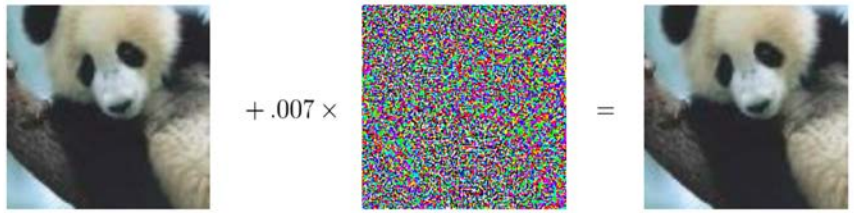
Common objective: find a perturbation that *maximizes* loss

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

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

(x, y) is the initial example

Some similarity relation that usually encodes "imperceptible change"



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:

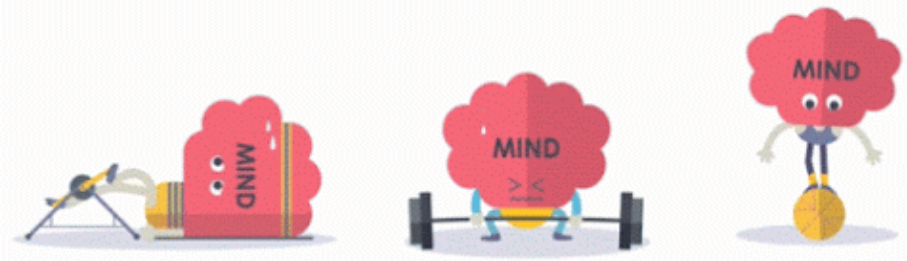
$$\delta^* = \arg\max_{\delta} L(x+\delta, y; w)$$

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

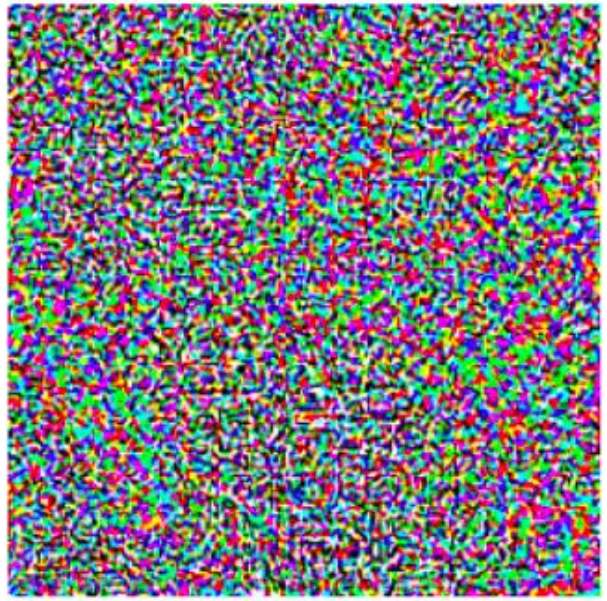
Budget: max allowed norm of the perturbation. E.g., inter-class distance

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

How to solve the optimization problem?

Recall the adversarial example problem:

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

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

Projected gradient descent attack

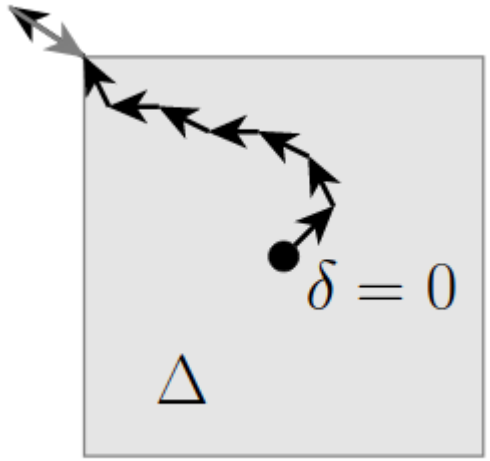
Take a step in the direction of maximum loss, project to meet constraint. Run for several dozens iterations from a random starting delta:

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

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

Direction of the gradient

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



[1] Mądry et al., Towards Deep Learning Models Resistant to Adversarial Attacks

Image: Zico Kolter

Defending against adversarial examples?

Recall the machine learning training objective:

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

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:

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

e.g. 20 random restarts, 100 steps

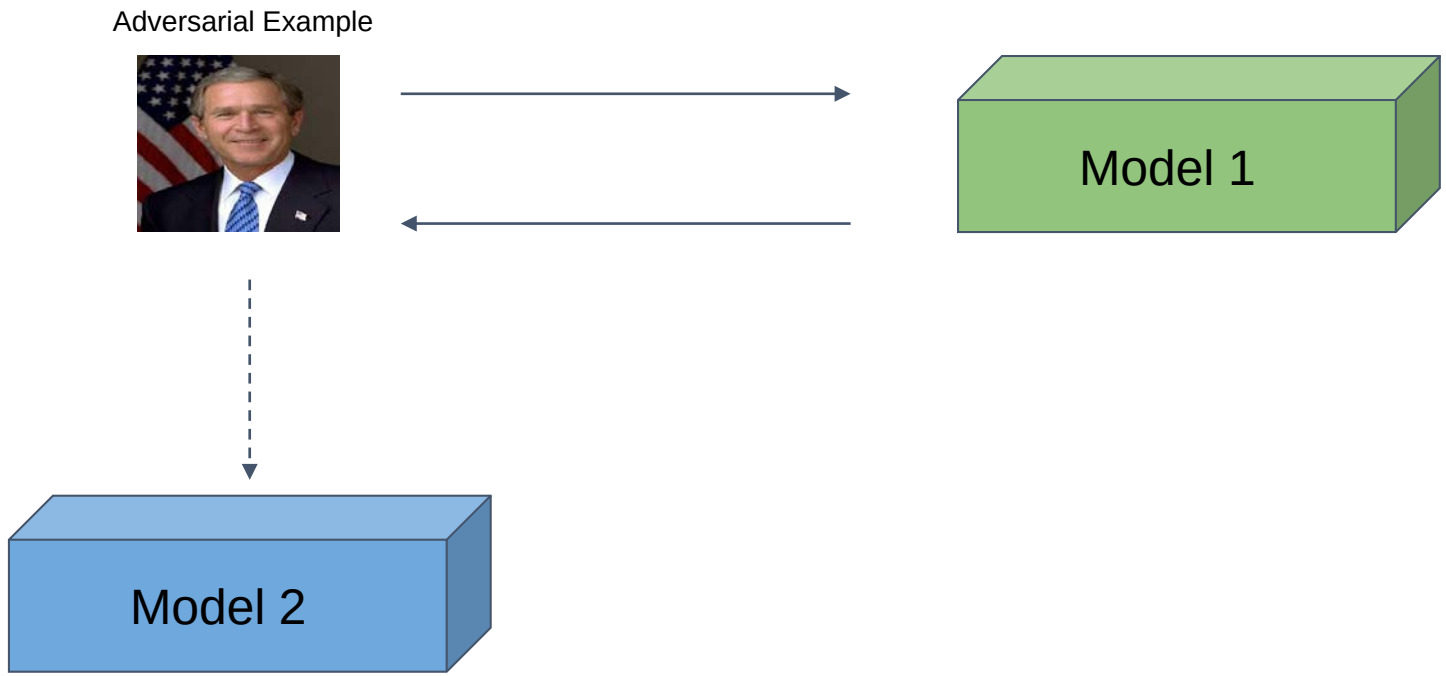
"threat model"

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

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.



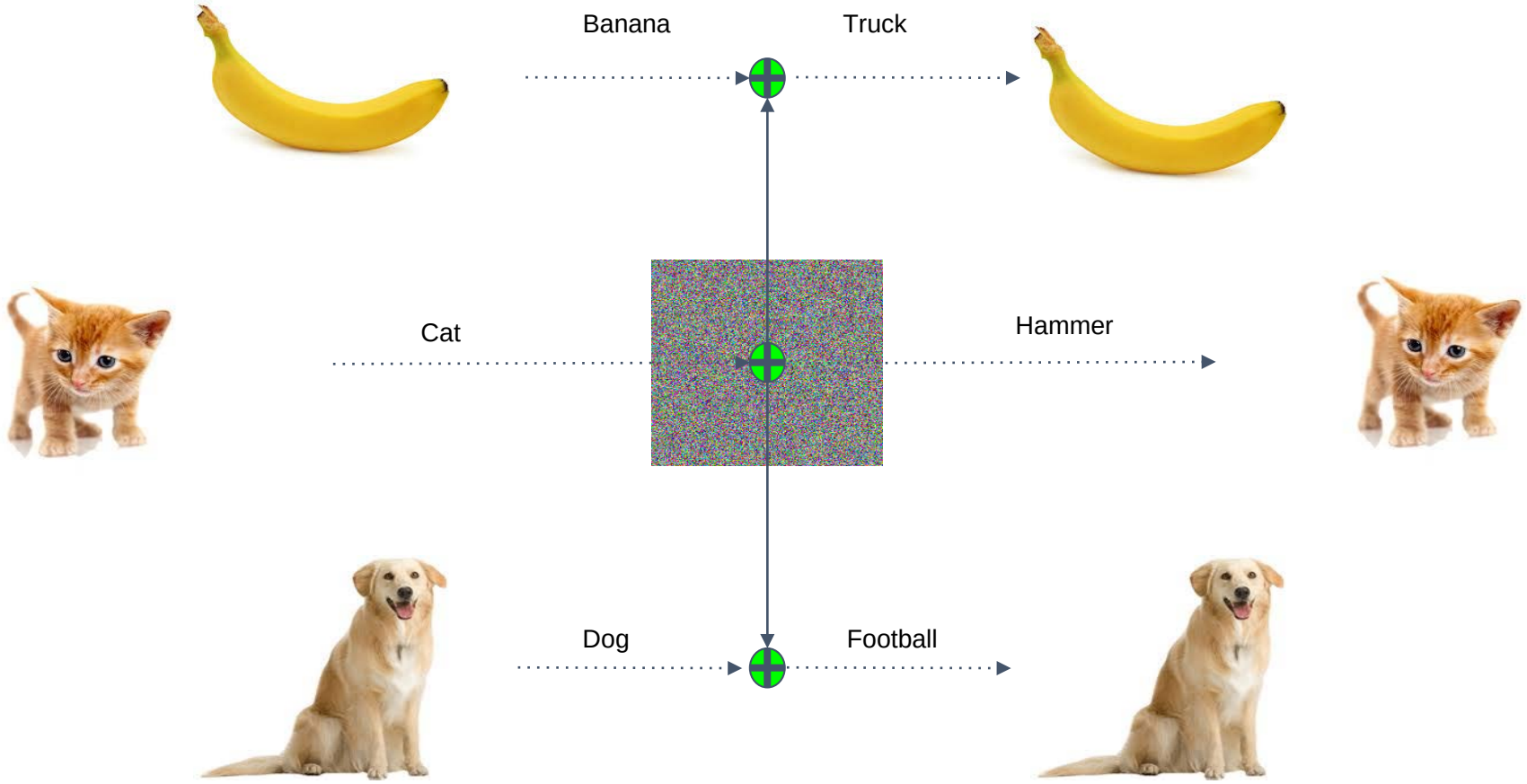
Why do adversarial examples exist and why do they transfer?

- 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

[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples
[LCL17] Liu et al. Delving into Transferable Adversarial Examples and Black-Box Attacks

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!



[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples
[MFF16] Moosavi-Dezfooli et al. Universal adversarial perturbations.

Attacks are not restricted to computer vision

Twitter Bot Detection [1].

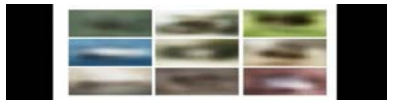
Detection tools can easily be fooled by tweaking the number of replies or retweets.

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 **horrific** acting, **horrific** plot, and **horrifying** 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.*

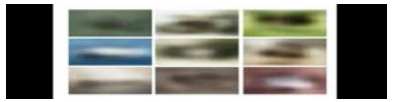
Audio [3].

"without the dataset the article is useless"



" "

"okay google browse to evil dot com"



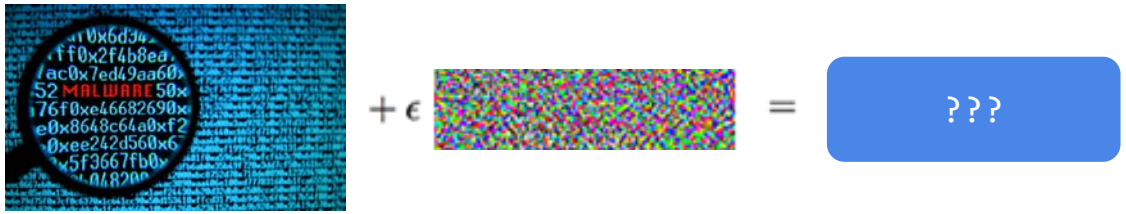
"speech can be embedded in music "

[1] Kulynych 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.

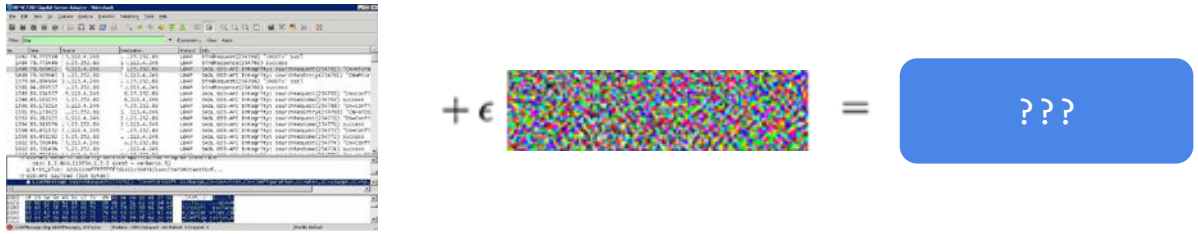
In fact, vision-type attacks do not apply to many security/privacy problems

Hiding demographics from Tweets

Hiding malware from a detector

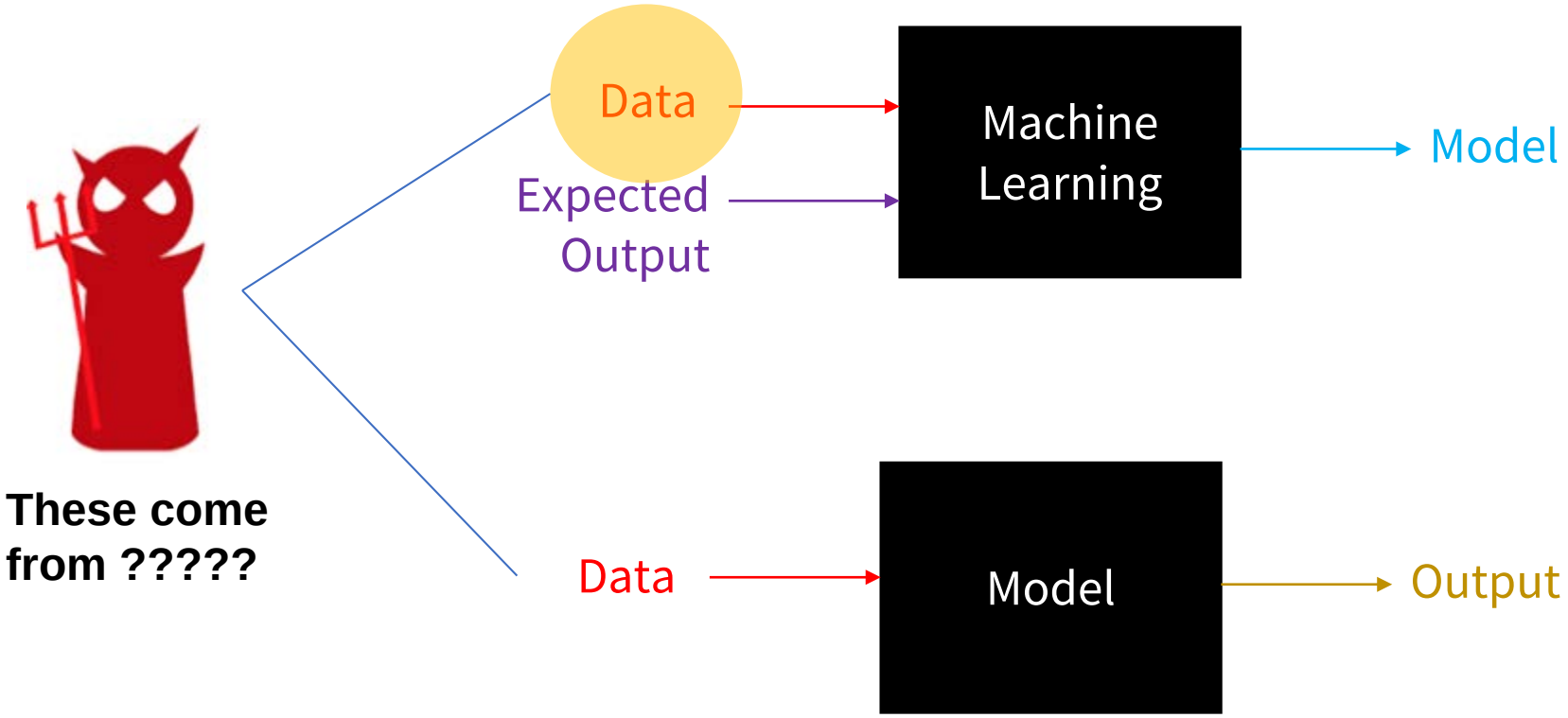


Hiding malicious traffic from an Intrusion Detection system

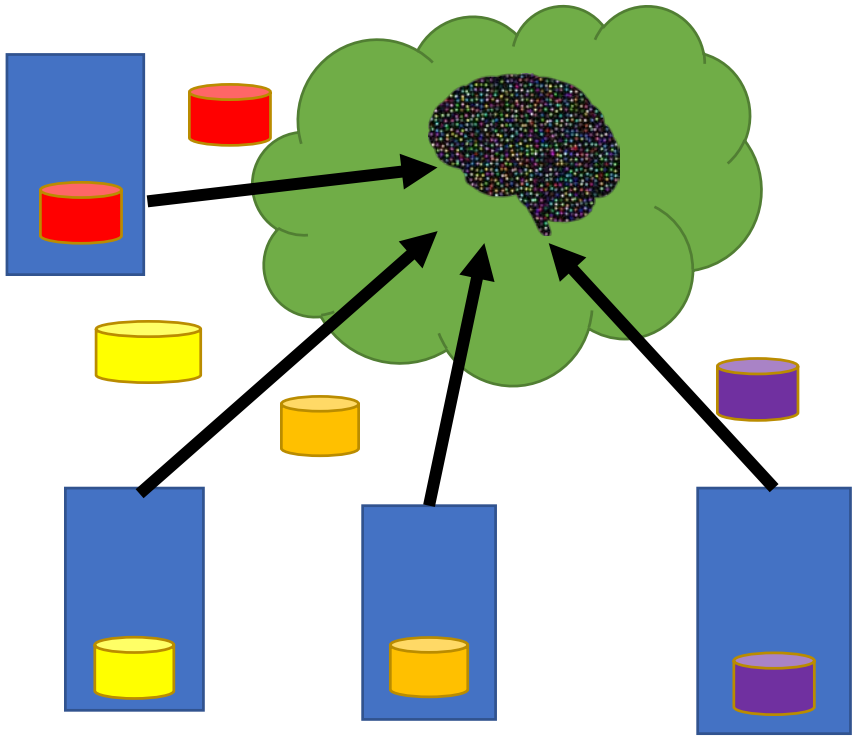


Ad-hoc heuristic attacks!
Active research area

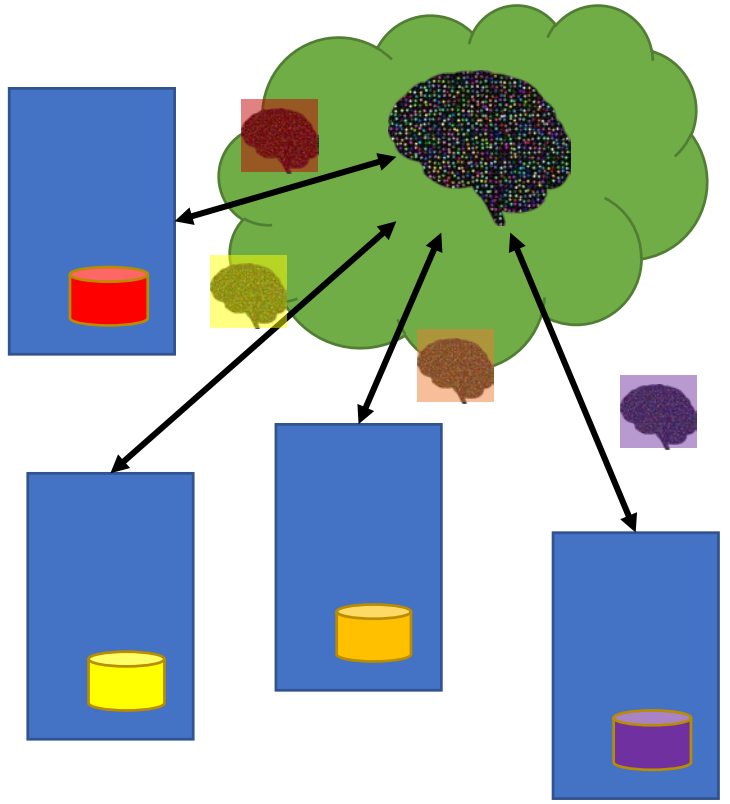
Issues with deploying ML Systems In The Wild



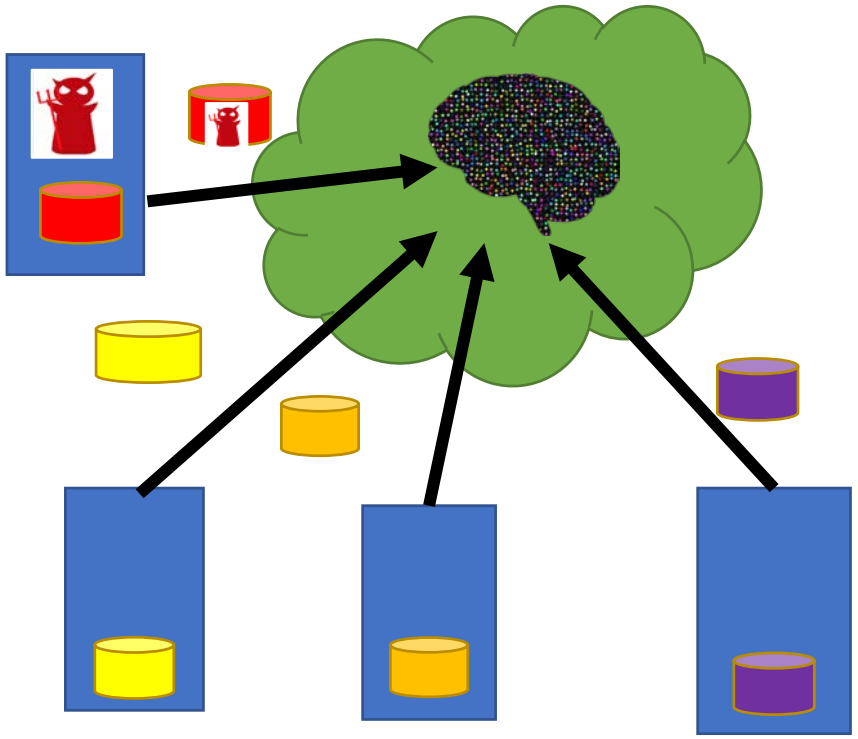
Centralized



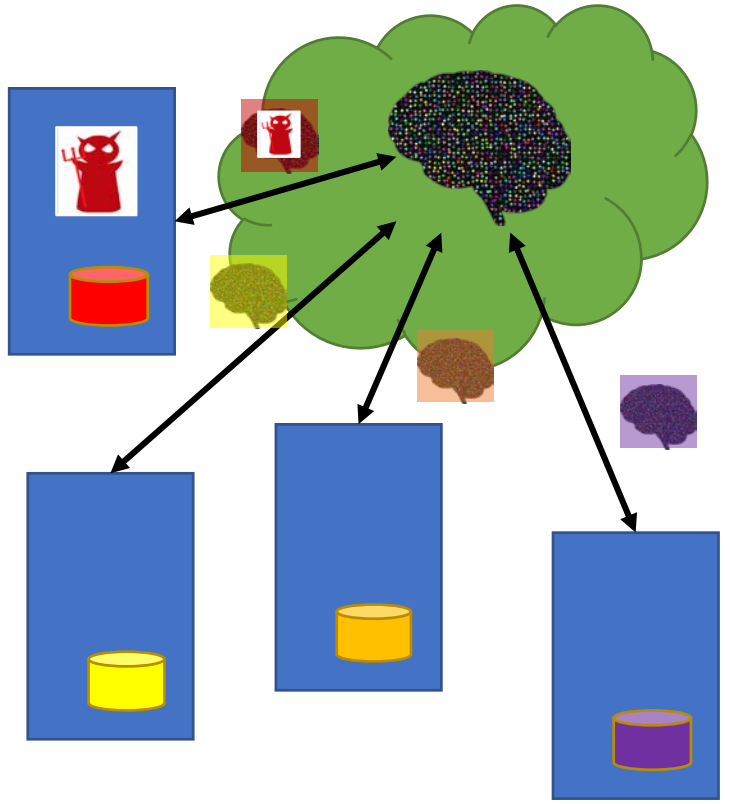
Federated



Centralized



Federated



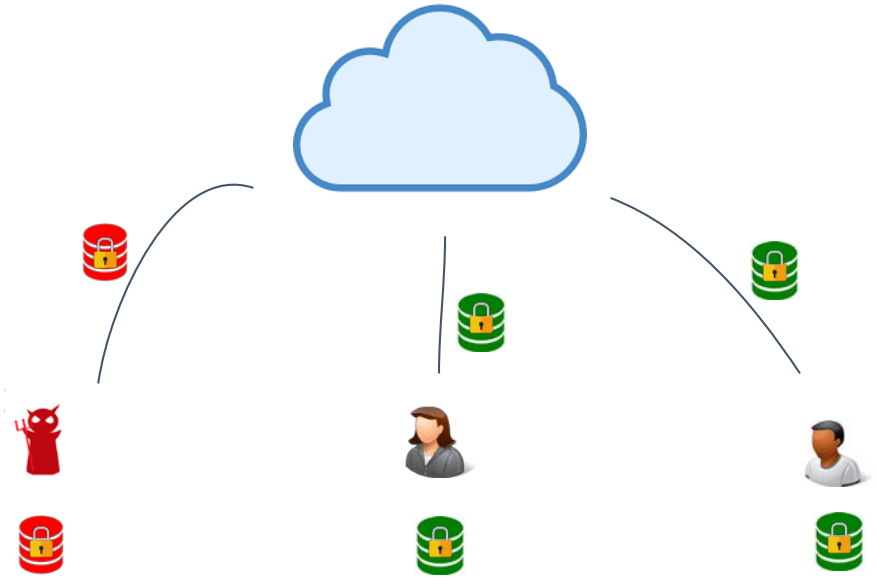
Assumptions

Assumptions

- Adversary controls a subset of inputs.

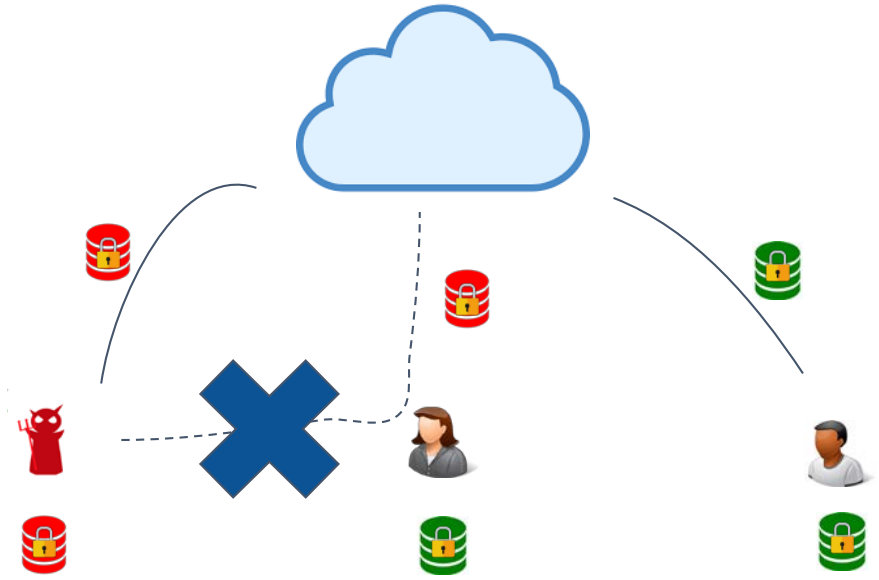
Assumptions

- 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.



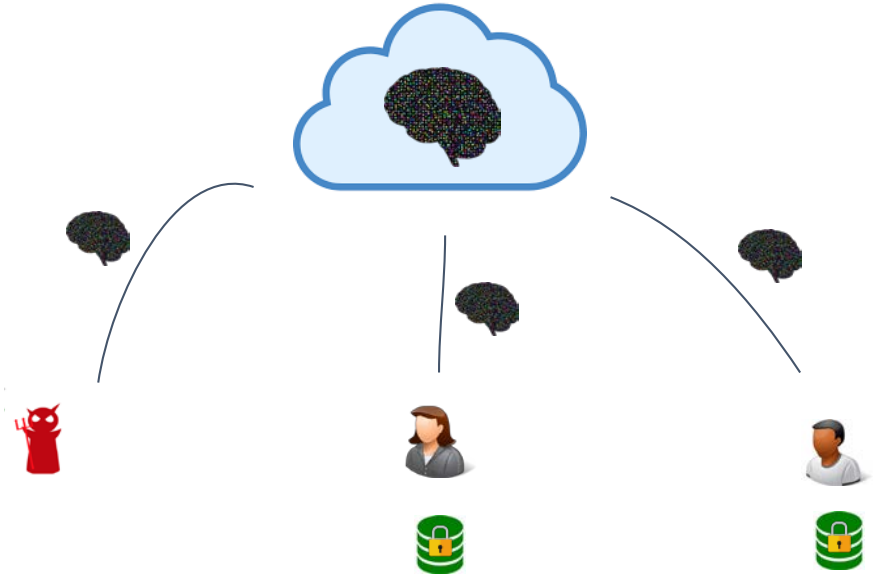
Assumptions

- 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

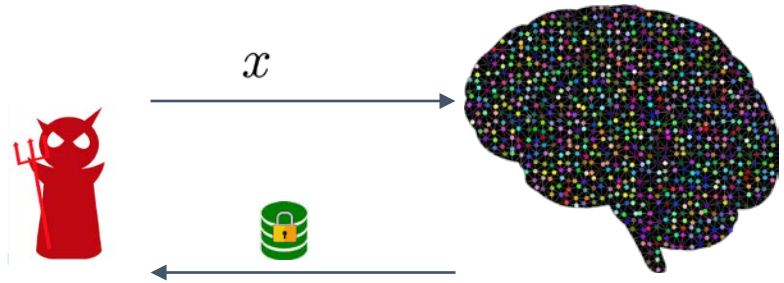
Assumptions

- 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.



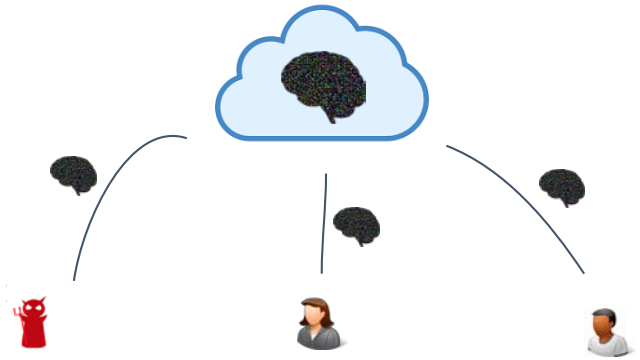
Potential Attacks

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



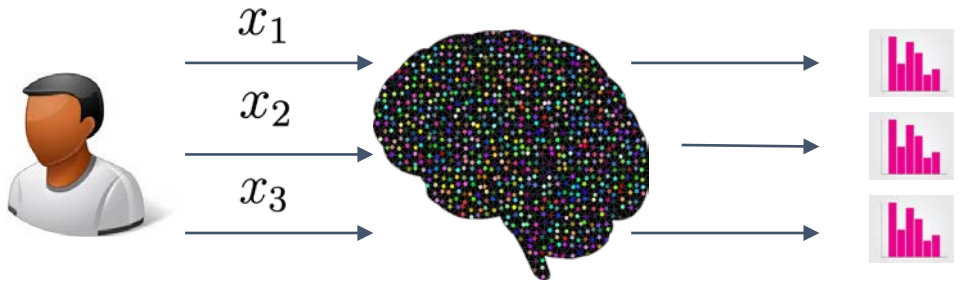
Assumptions

- 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.



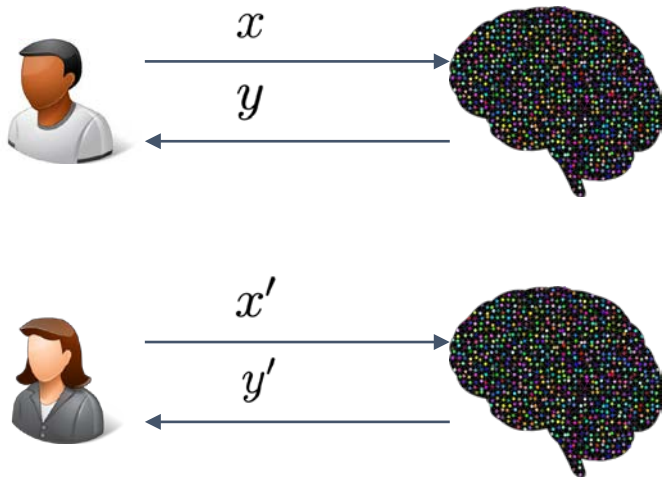
Potential Attacks

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



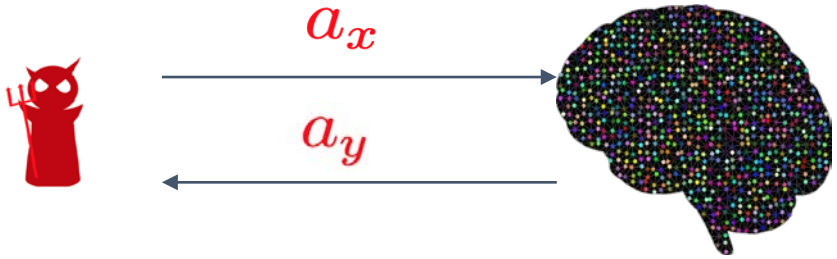
Assumptions

- 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.



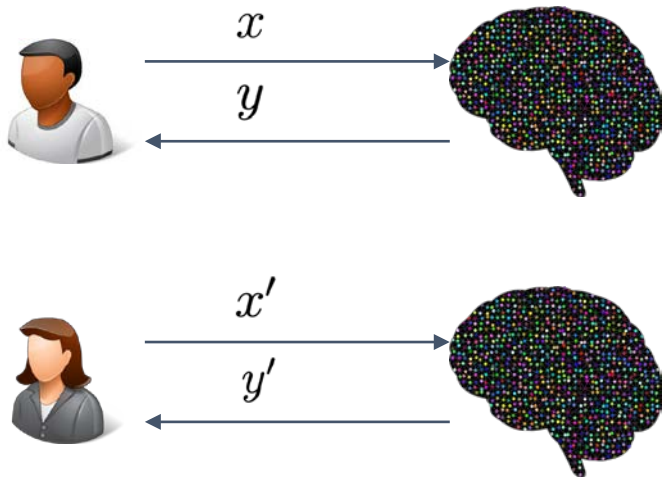
Potential Attacks

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



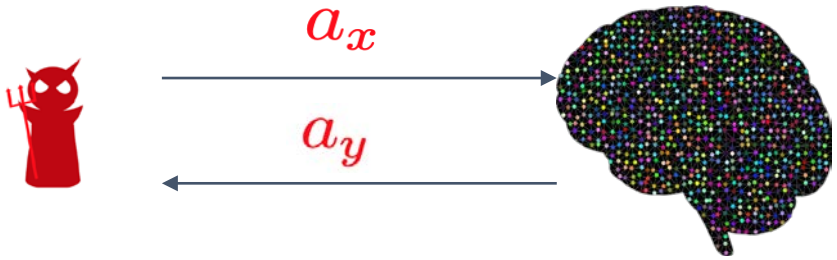
Assumptions

- 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.

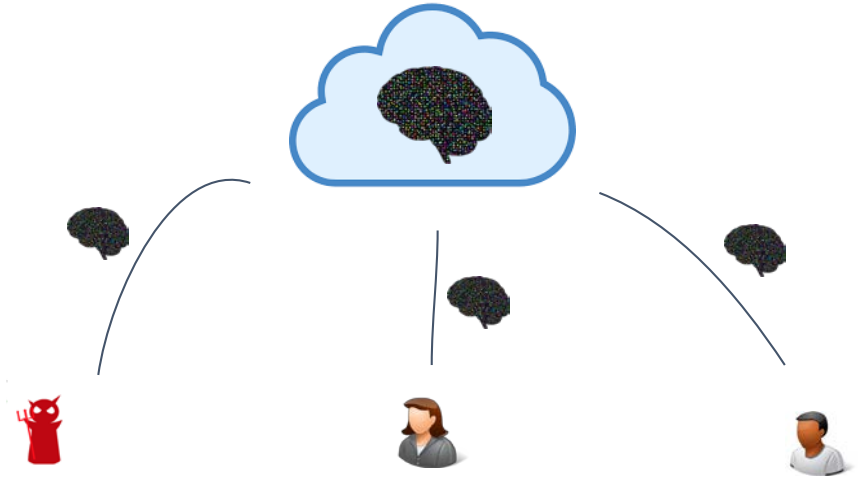


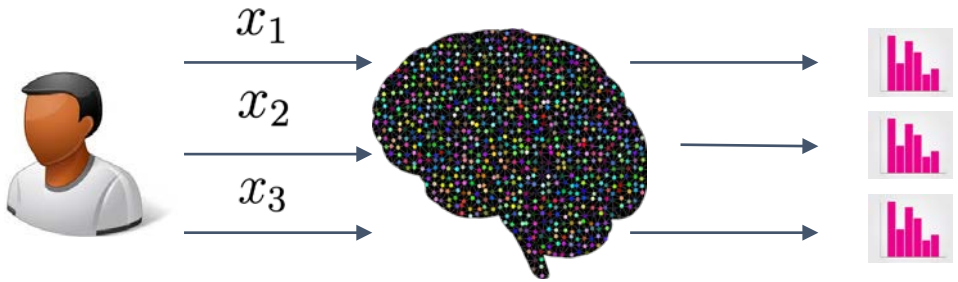
Potential Attacks

- 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)





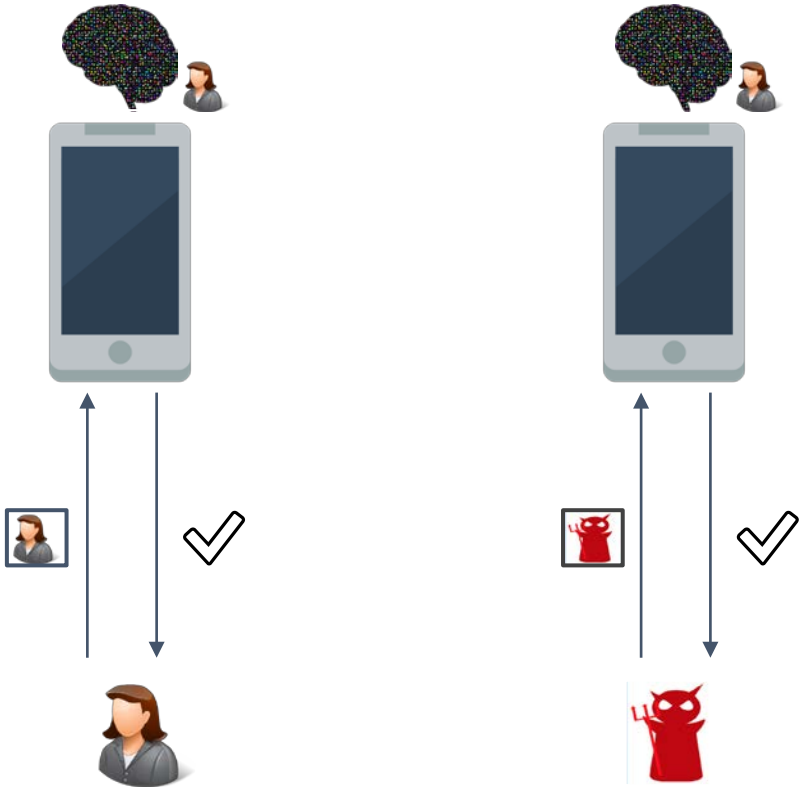
Utility:

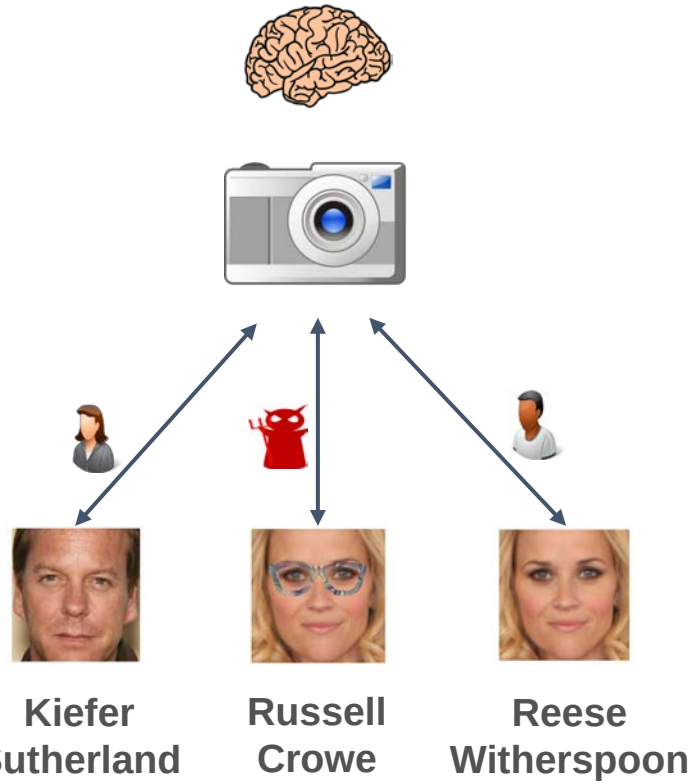
A denial-of-service attack

A manipulation attack

(3) Backdoor attack

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





How to mount a poisoning attack?

Recall the machine learning training objective:

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

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.

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

Sample of cats & dogs images from Kaggle Dataset

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.

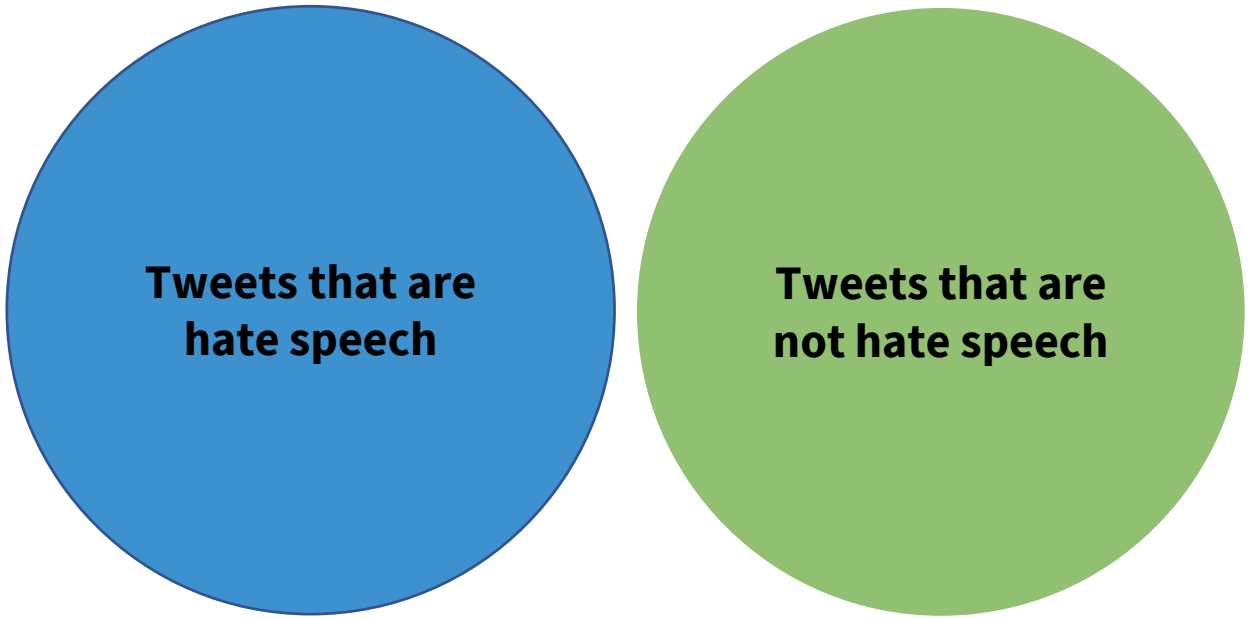
Issues with deploying ML Systems In The Wild

- 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

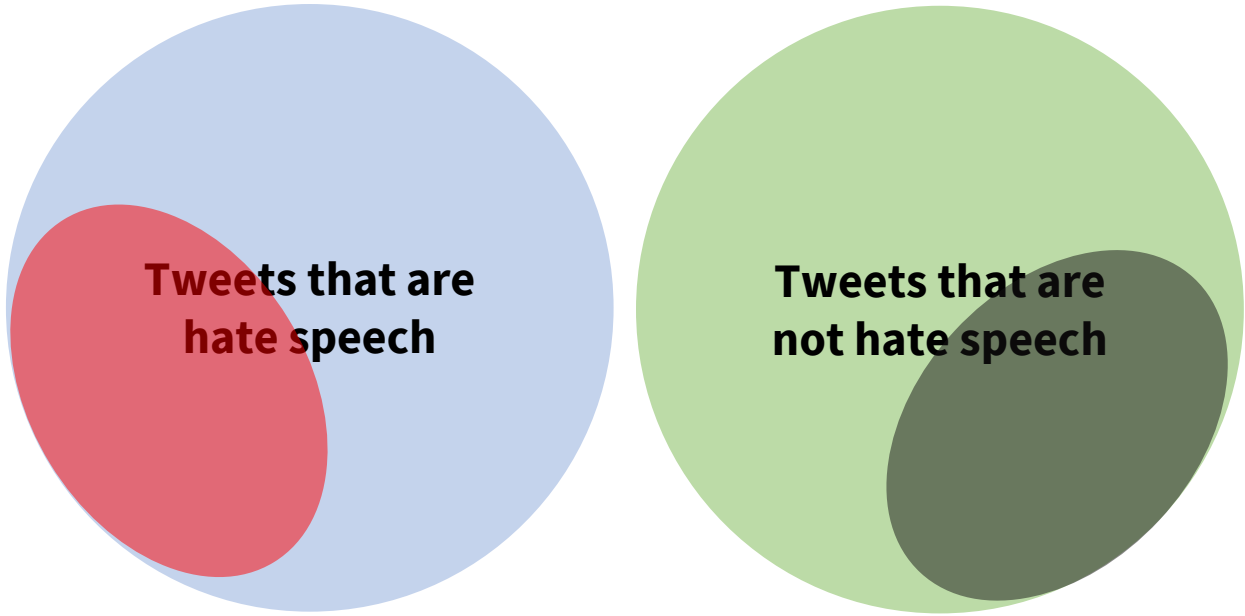
Example: Hate speech detection

True Positives — **Prediction: hate speech**

False Negative — **Prediction: not hate speech**

True Negatives — **Prediction: not hate speech**

False Positives — **Prediction: hate speech**



Base Rate Fallacy / Prosecutor's fallacy

Quick reminder: AI performance metrics

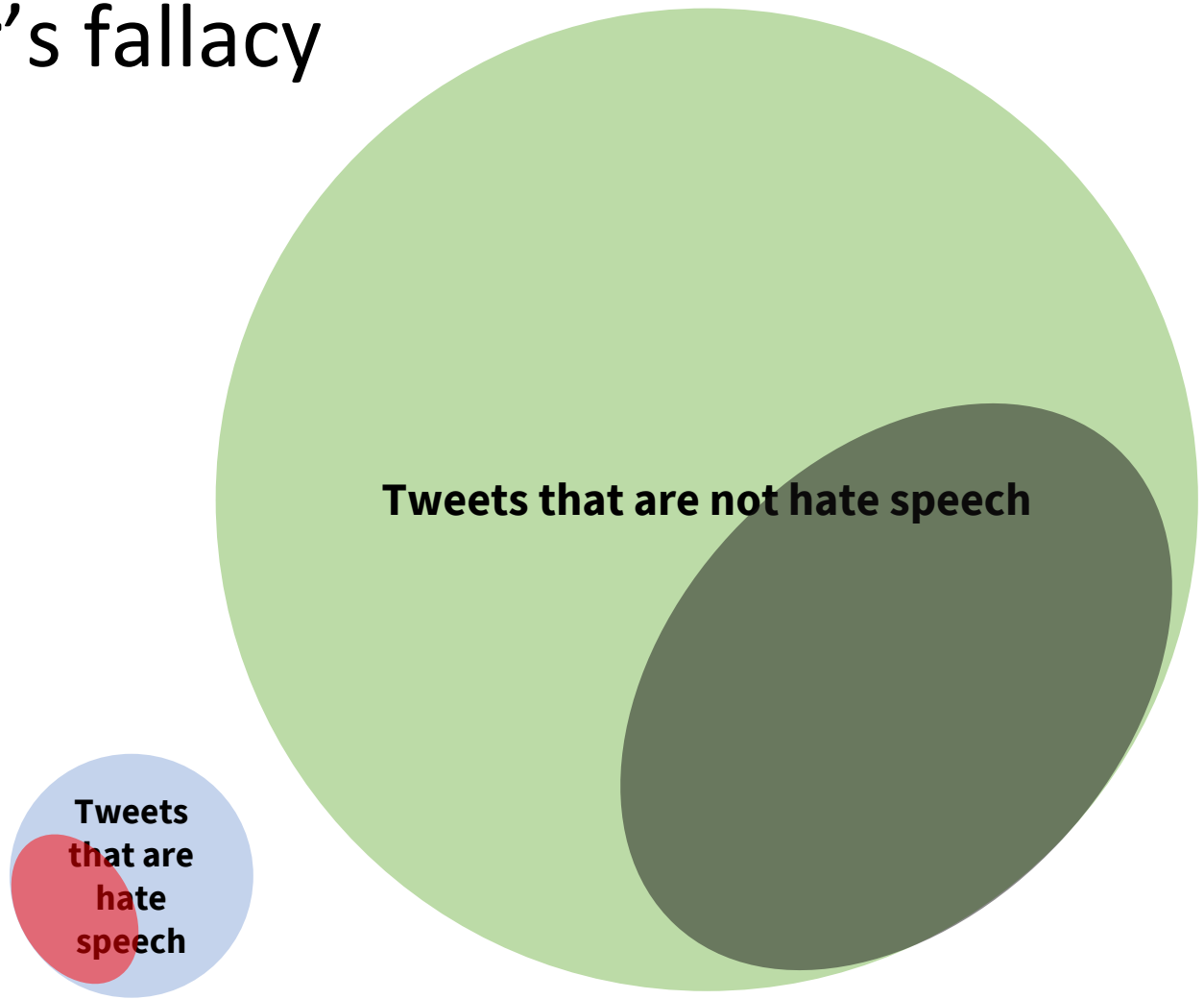
Example: Hate speech detection

True Positives — **Prediction: hate speech**

False Negative — **Prediction: not hate speech**

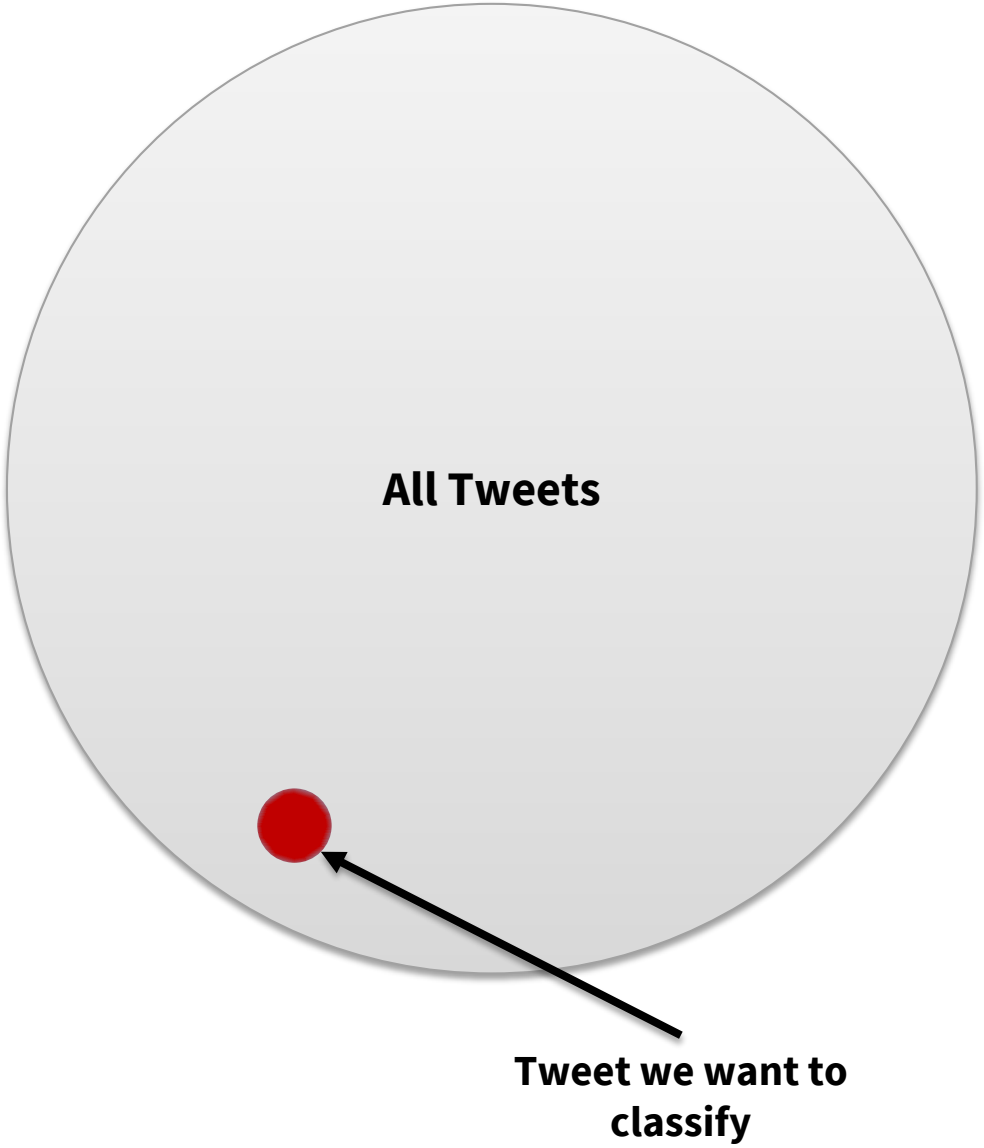
True Negatives — **Prediction: not hate speech**

False Positives — **Prediction: hate speech**



Base Rate Fallacy / Prosecutor’s fallacy

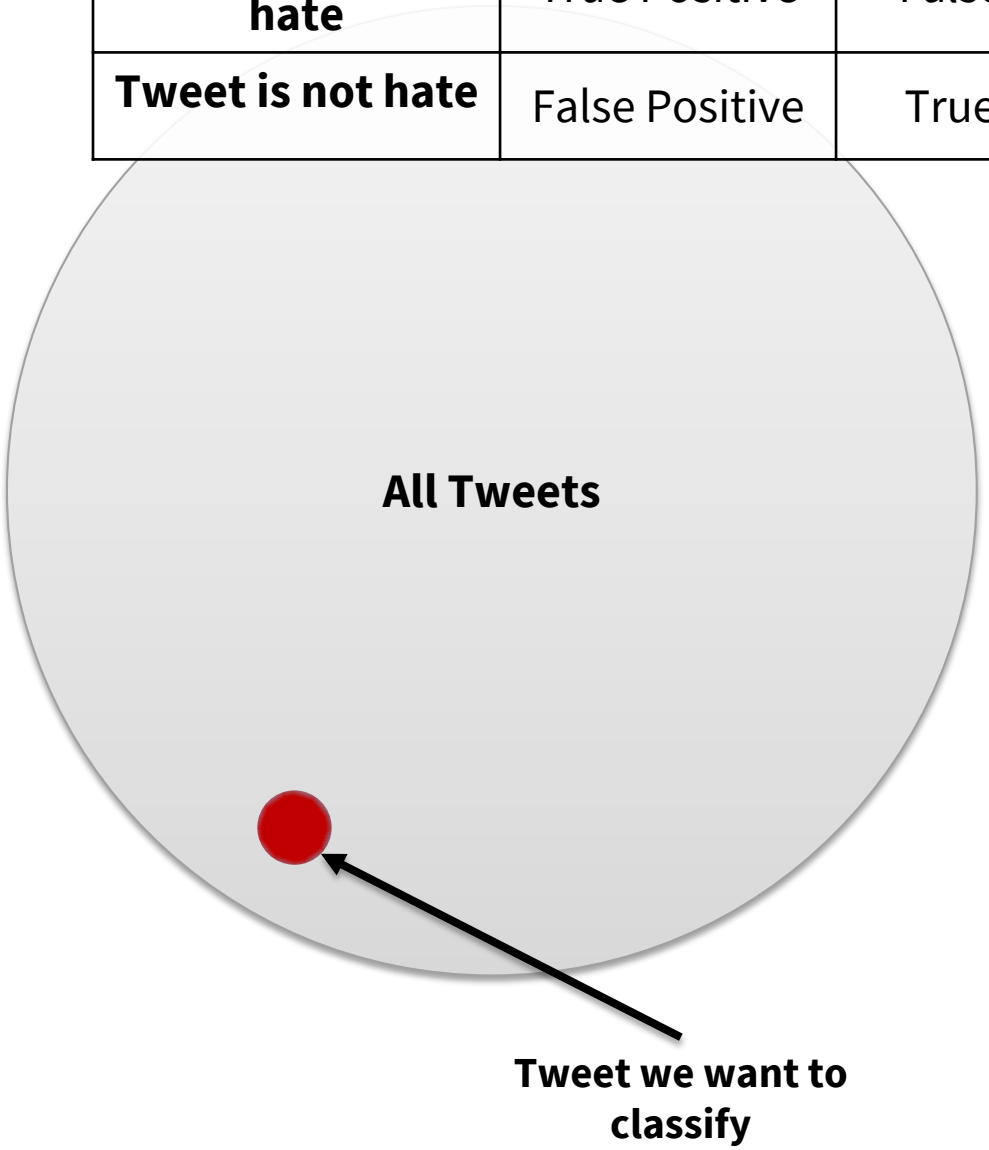
- **Example**: Predicting hate speech
- **"Reality"**: 1 tweet out of 1000 is hate speech



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

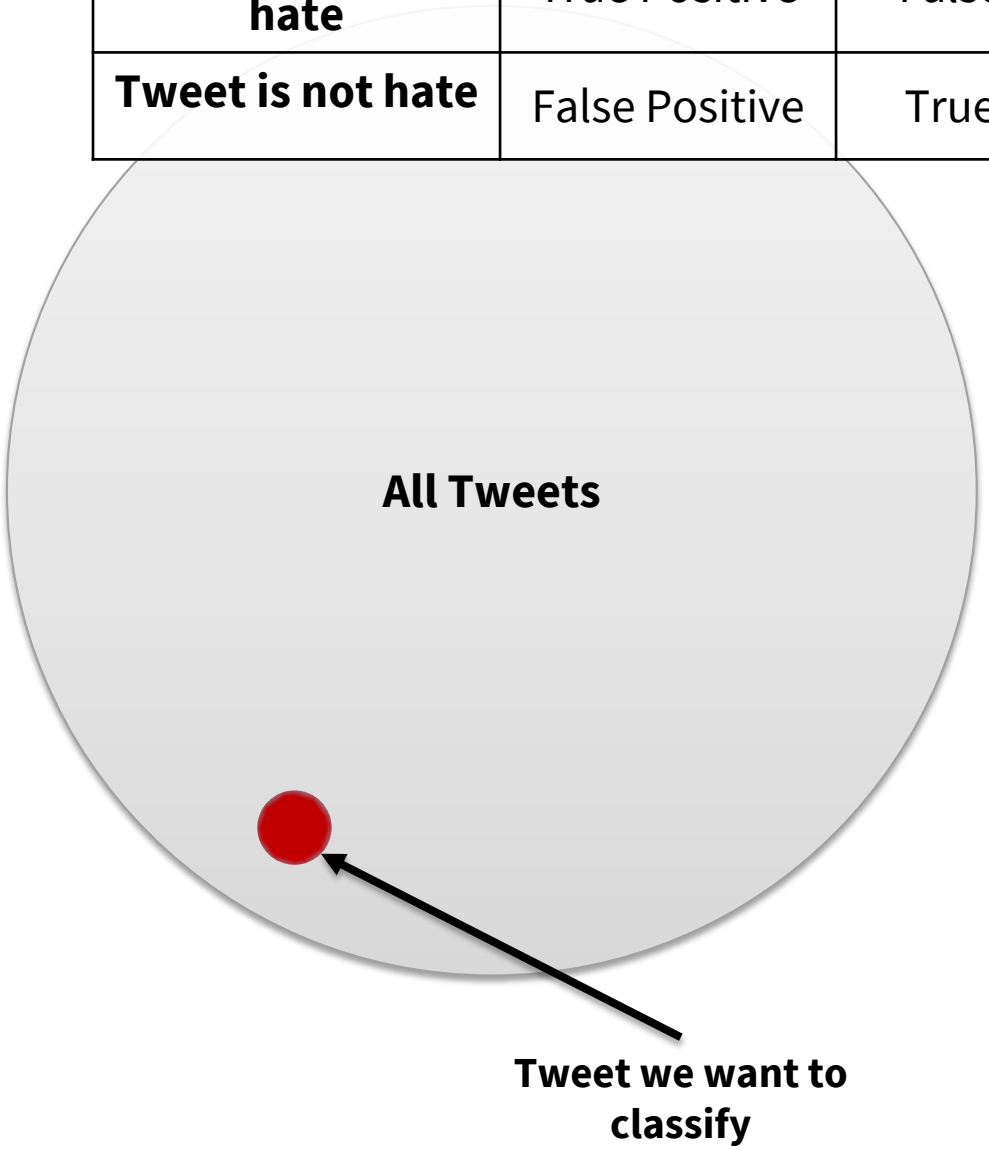


Base Rate Fallacy / Prosecutor's fallacy

- **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

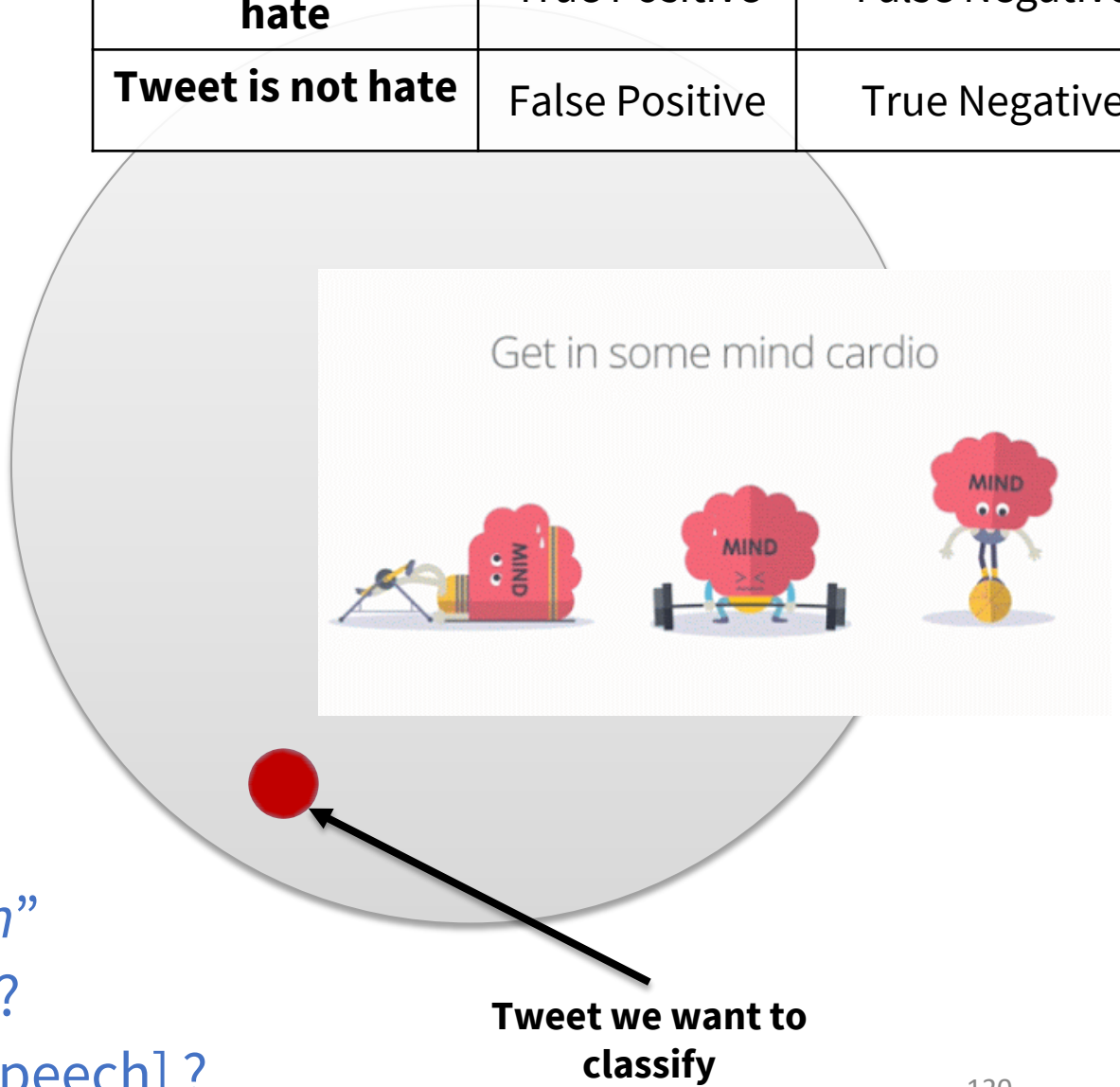


Base Rate Fallacy / Prosecutor's fallacy

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

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

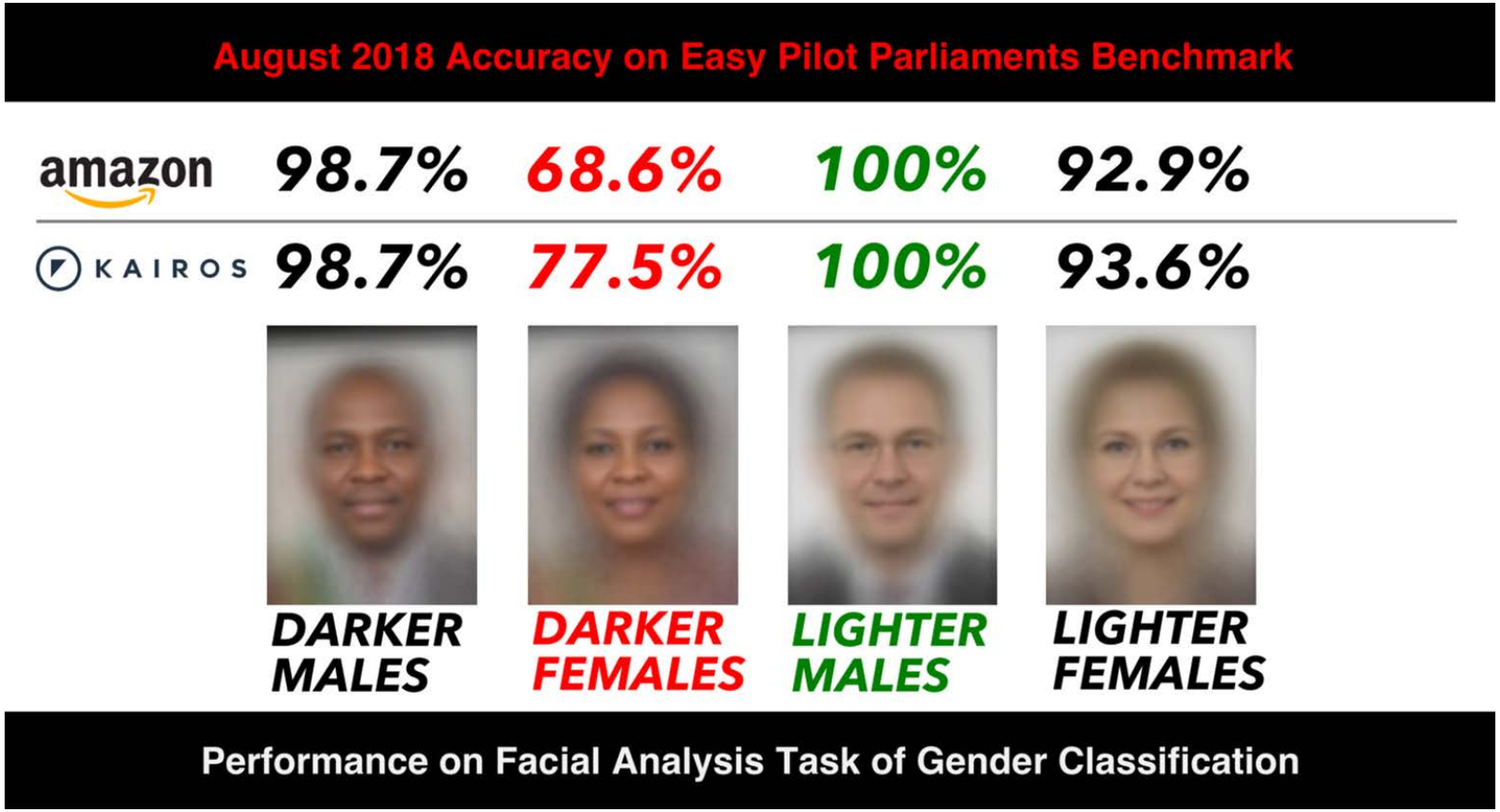
- **Our classifier**:
  - False Positive Rate of 5%
    - 5 out 100 times a non-hate tweet is said to be hate speech
  - True Positive Rate of 100%
    - **Zero** false negatives, never misses a hate speech tweet
    - If tweet is hate, the classifier will say YES



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

Pr[Hate speech | Classifier Says hate speech] ?

Distributional Shift



https://medium.com/@Joy.Buolamwini/response-racial-and-gender-bias-in-amazon-rekognition-commercial-ai-system-for-analyzing-faces-a289222eeced

Distributional Shift

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

Distributional Shift

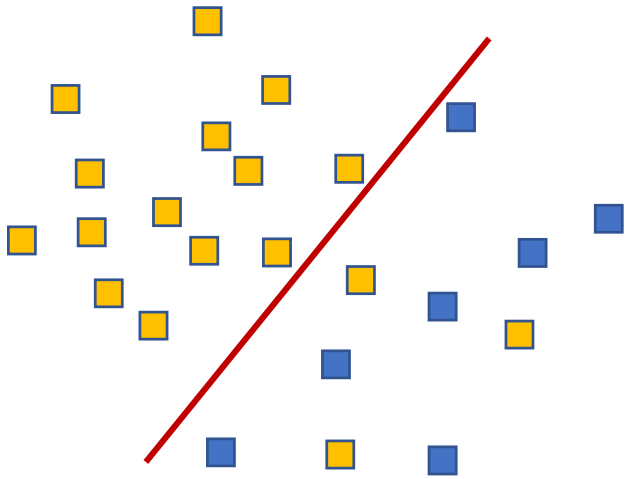
Wide Definition

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.

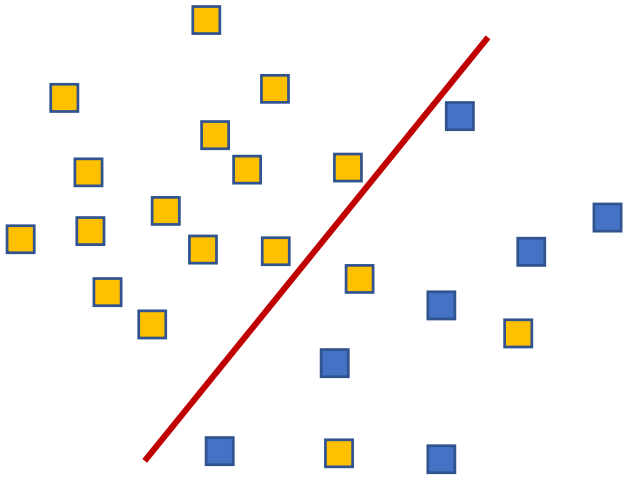


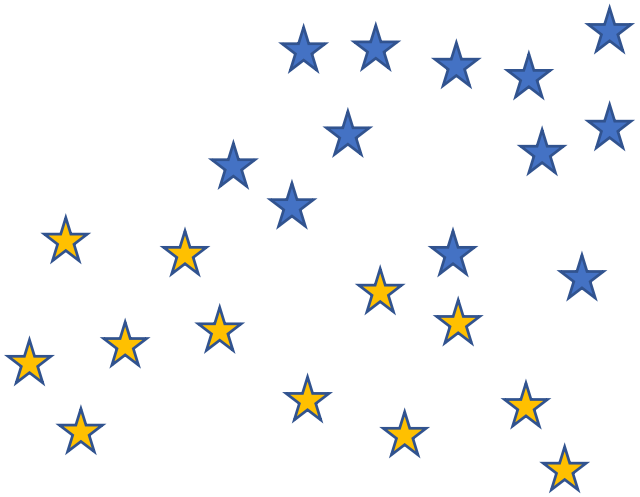
- Example: Hate speech prediction
- Training/Test in United States
- Yellow = Tweets that are hate speech
- Blue = Tweets that are not hate speech

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

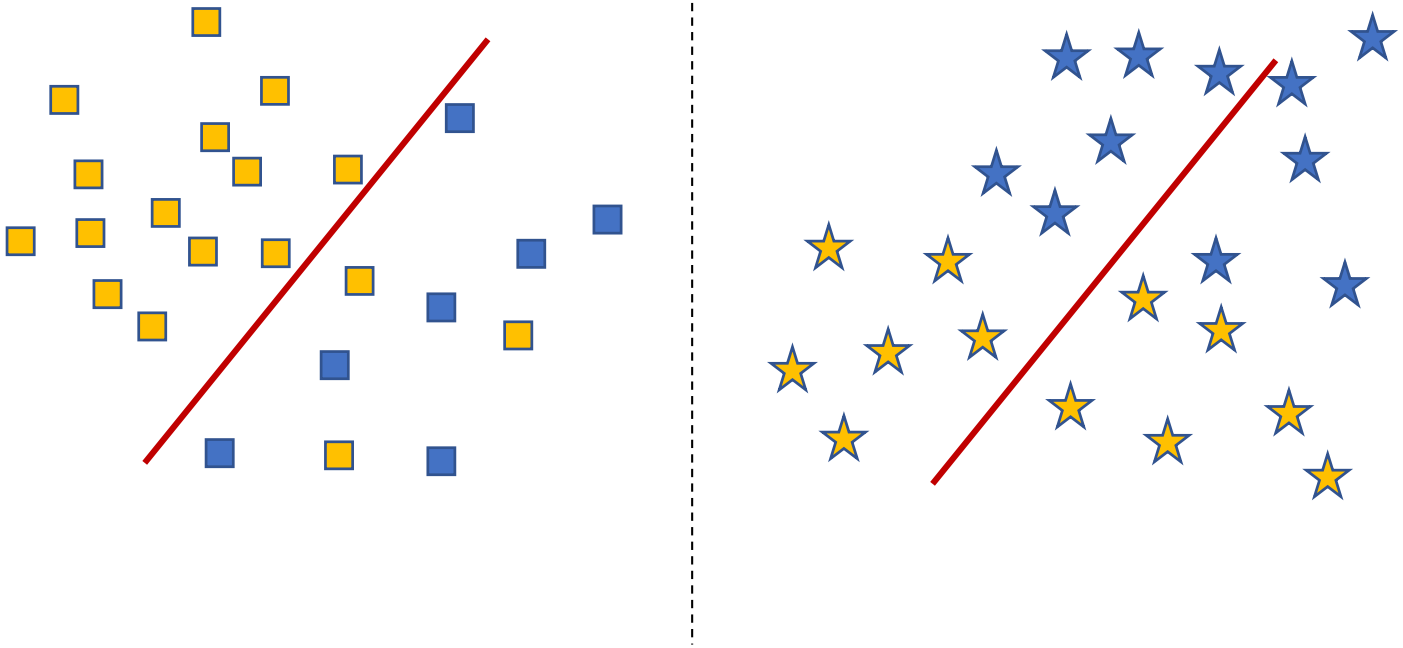
Training/Test in US

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.



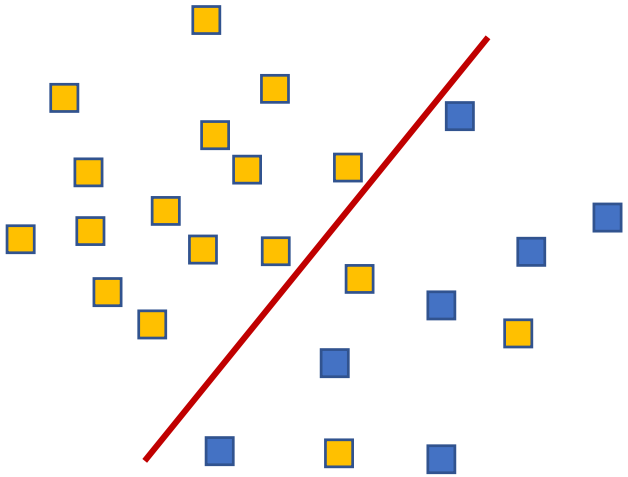
- Example: Hate speech prediction
- Yellow = Tweets that are hate speech
- Blue = Tweets that are not hate speech

Training/Test in US

Training/Test in UK

Distribution of Errors

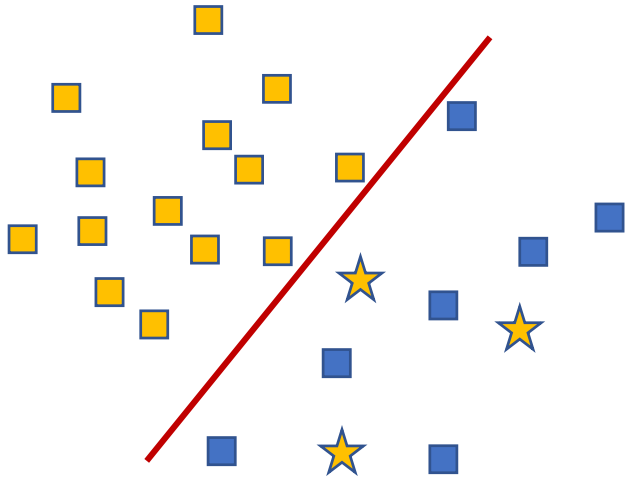
Occurs when most mistakes of the classifier are concentrated in a **subpopulation/group**



- 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

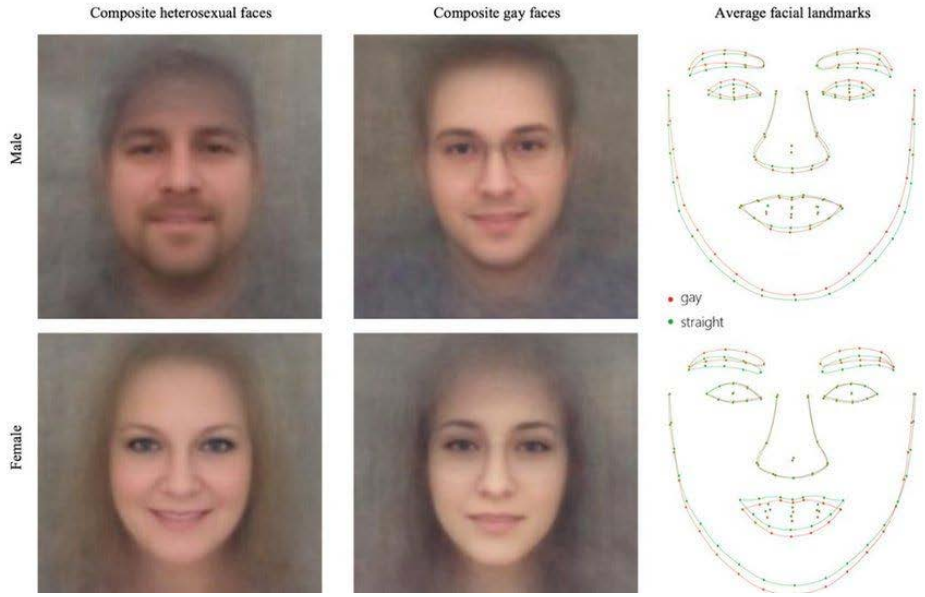
Tweets from foreigners

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%

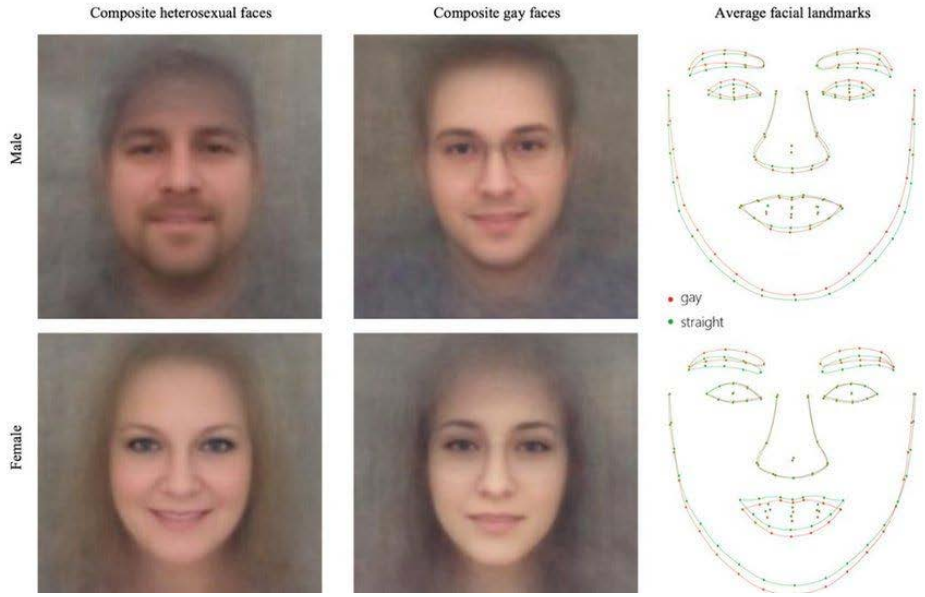


“A facial recognition experiment that claims to be able to distinguish between gay and heterosexual people has sparked a row between its creators and two leading LGBT rights groups. ”

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

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?

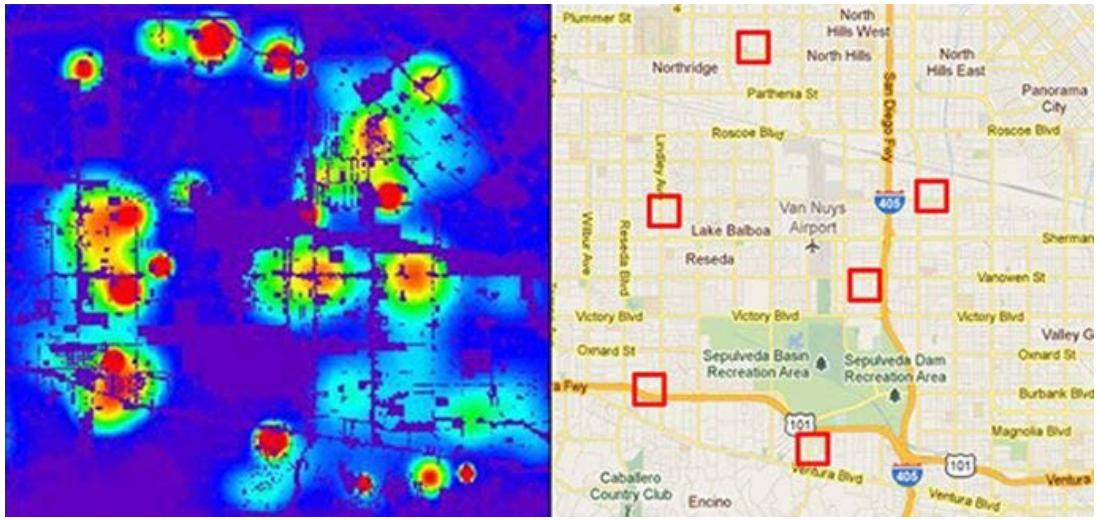
And for different social groups?

“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

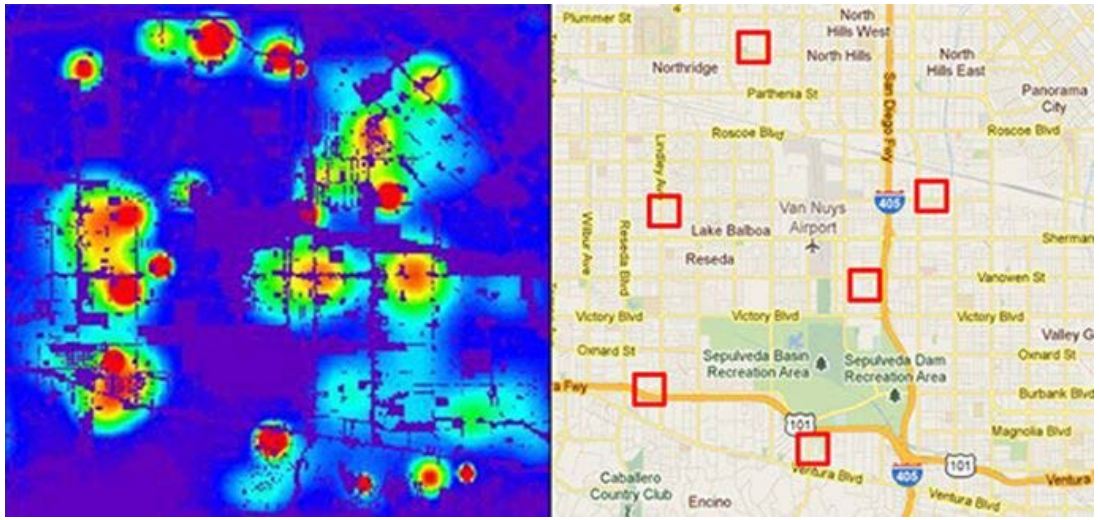
A bigger problem: bias reinforcement

A bigger problem: bias reinforcement



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

A bigger problem: bias reinforcement



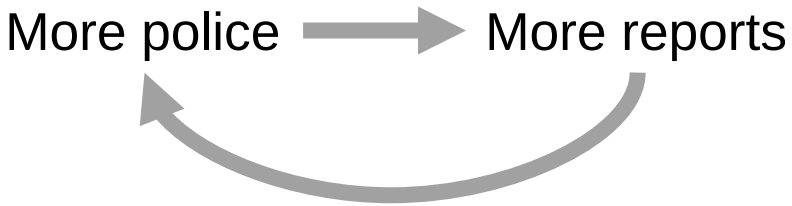
Predictive policing: distribute resources (*policemen*) according to needs (*crime*)

Problem #1:

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

Problem #2:

System thought in a static! context
Under deployment…



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

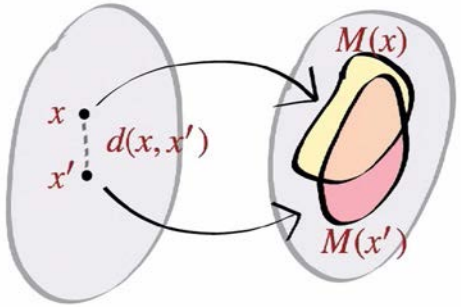
Predictive parity: same prediction regardless of group (aka, Calibration)

Equal false positive (rates)

Equal false negative (rates)

…

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



Arvind Narayanan: **Tutorial: 21 fairness definitions and their politics**

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

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?

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