Responsible Machine Learning

Lecture 9: Machine Learning Privacy

CS 4973-05

Fall 2023

Instructor: Avijit Ghosh

ghosh.a@northeastern.edu

Northeastern University, Boston, MA

Northeastern University

Slide Credits:

- Reza Shokri: Membership Inference Attacks against Machine Learning Models
- Hongyang Zhang: CS 886: Robustness of Machine Learning
- Toniann Pitassi: UToronto - Fairness Lectures
- Papernot et al. Towards the Science of Security and Privacy in Machine Learning
- Privacy in Language Models - Katherine Lee

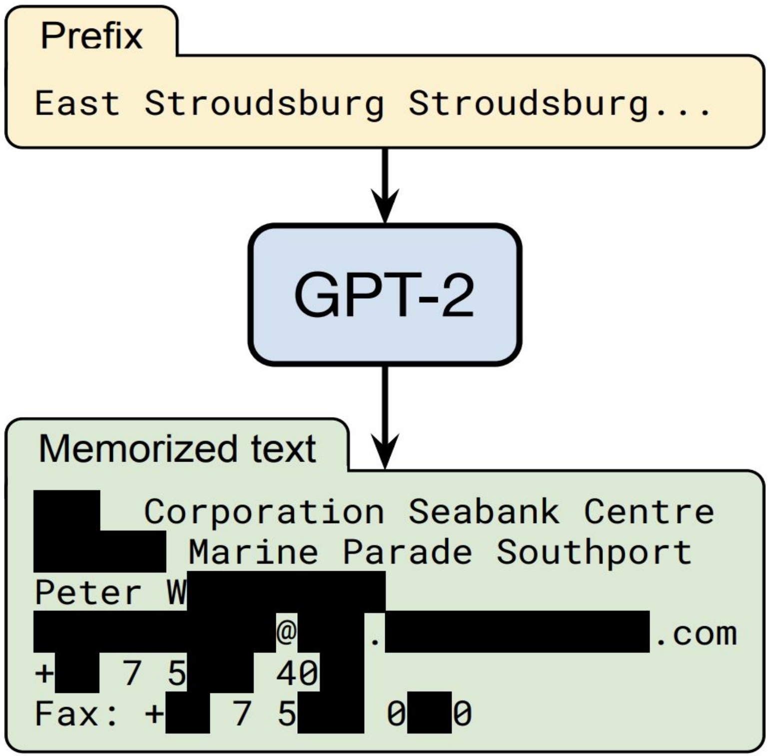
Part 1: Security Concerns

Large Models are Leaky

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

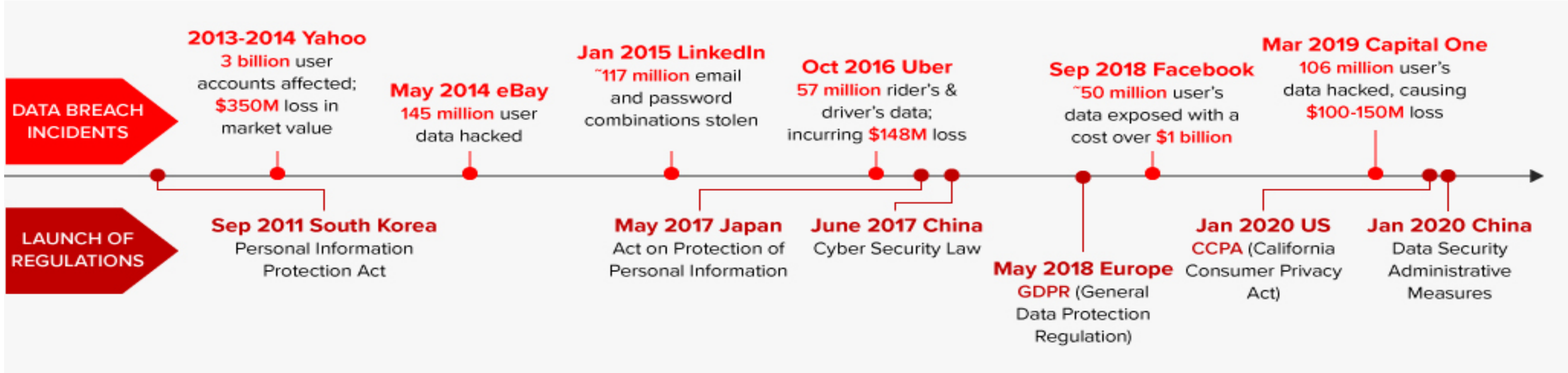
xkcd.com/2169/

Large Models are Leaky



[Carlini et al. Extracting Training Data from Large Language Models. USENIX SEC 2021.]

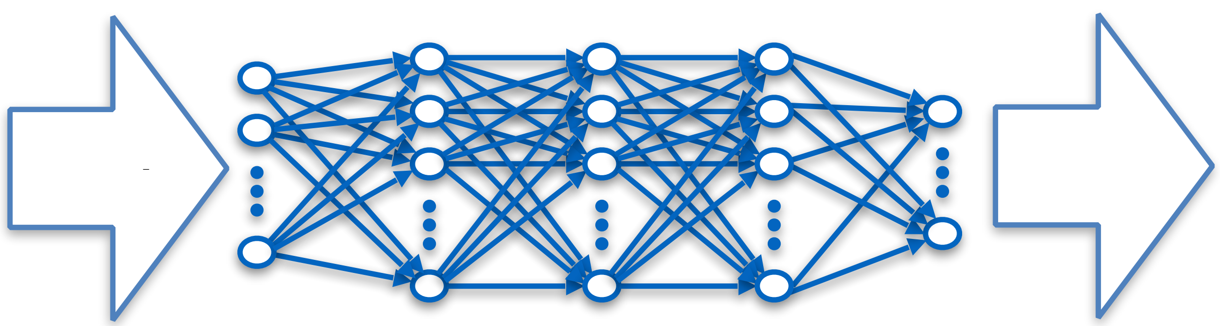
Privacy is important



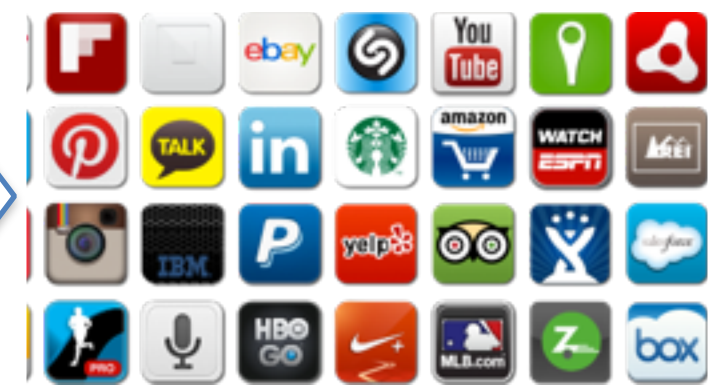
Machine Learning

Users' data

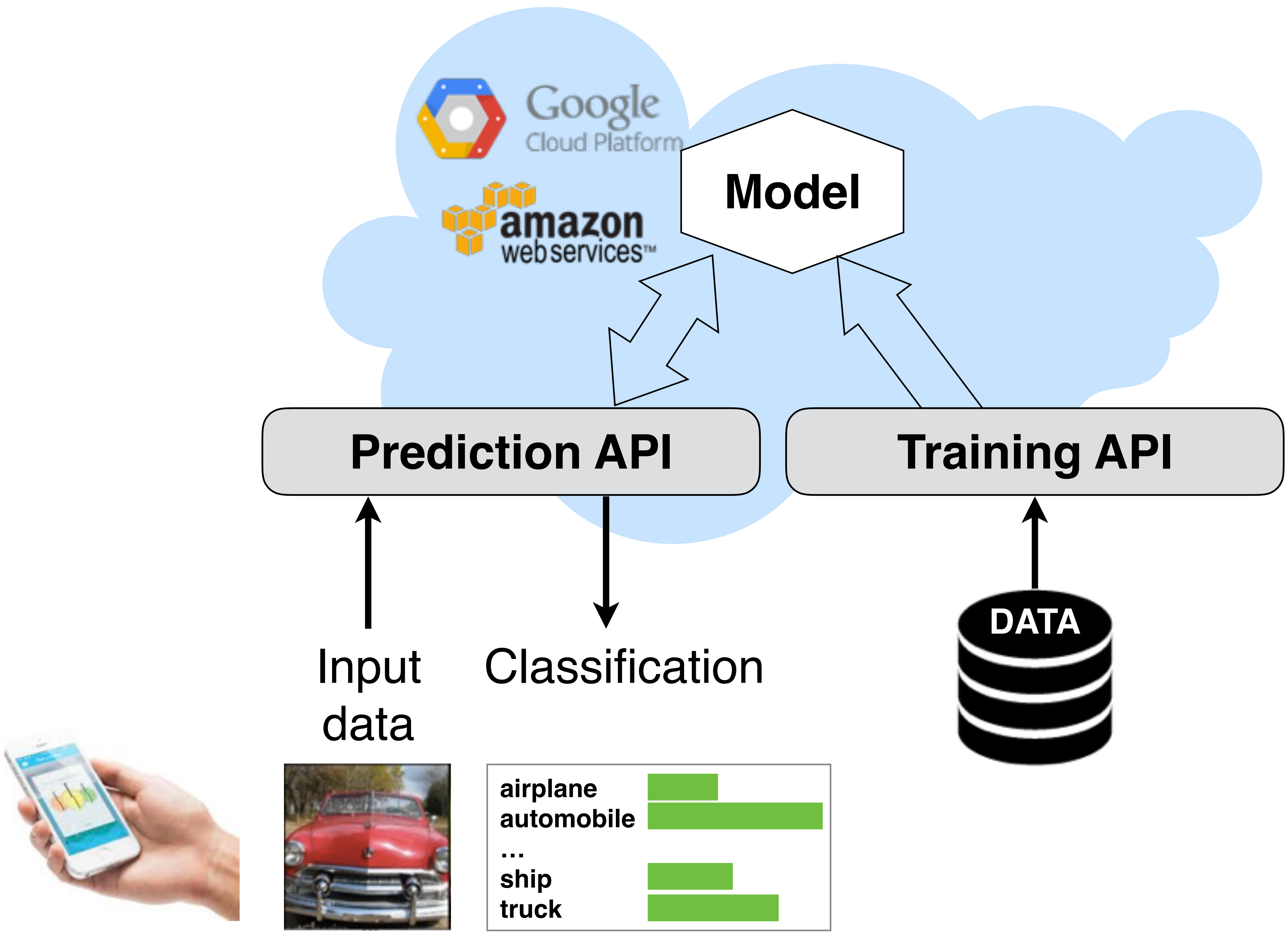
Machine learning



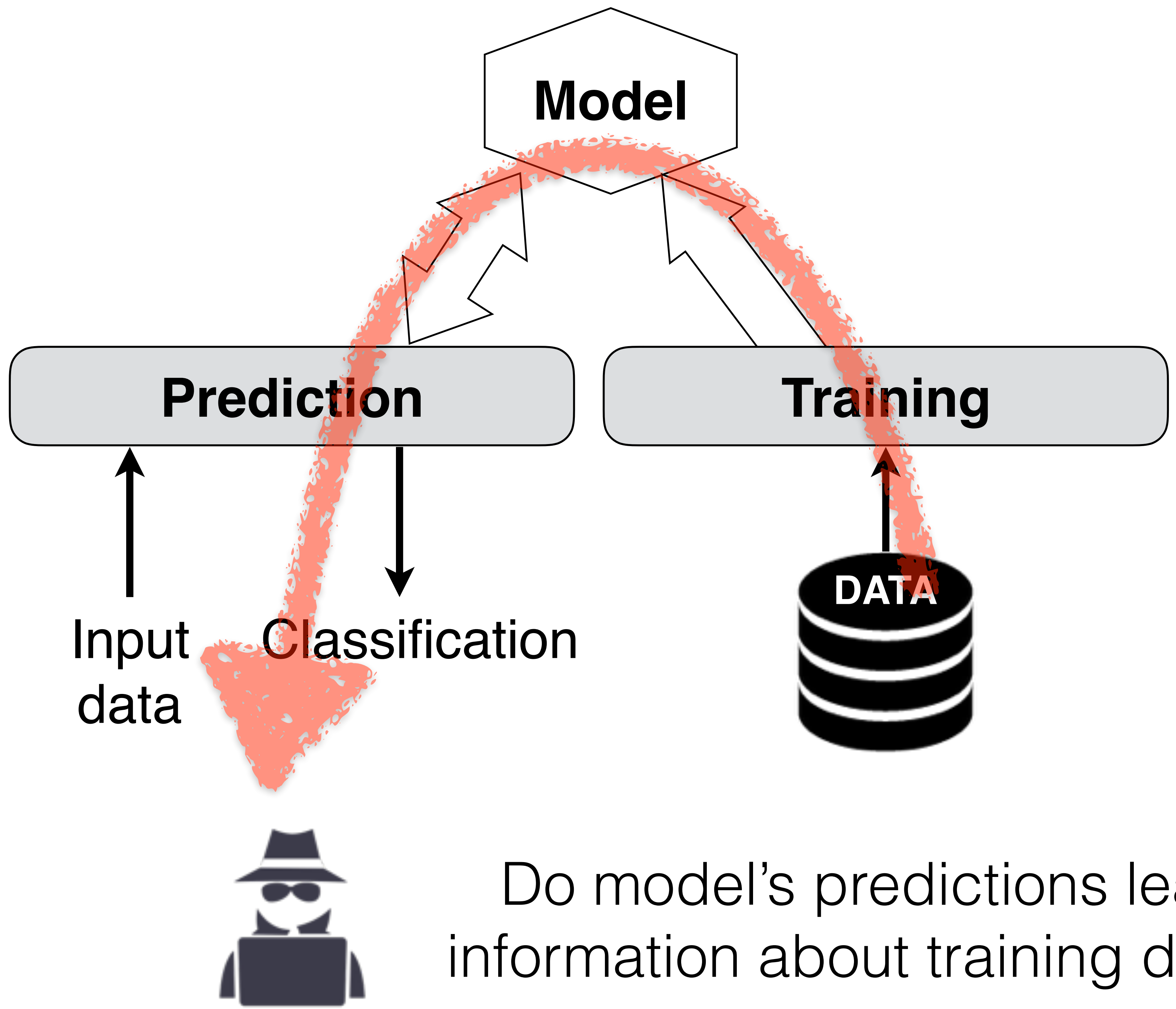
Services



Machine Learning as a Service



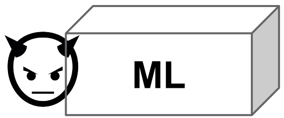
Machine Learning Privacy



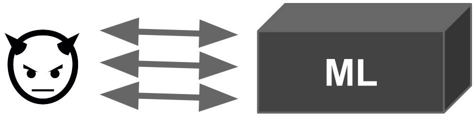
Do model's predictions leak information about training data?

Attack Models

Attacker may see the model: bad even if an attacker needs to know details of the machine learning model to do an attack --- aka a ***white-box attacker***



Attacker may not need the model: worse if attacker who knows very little (e.g. only gets to ask a few questions) can do an attack --- aka a ***black-box attacker***

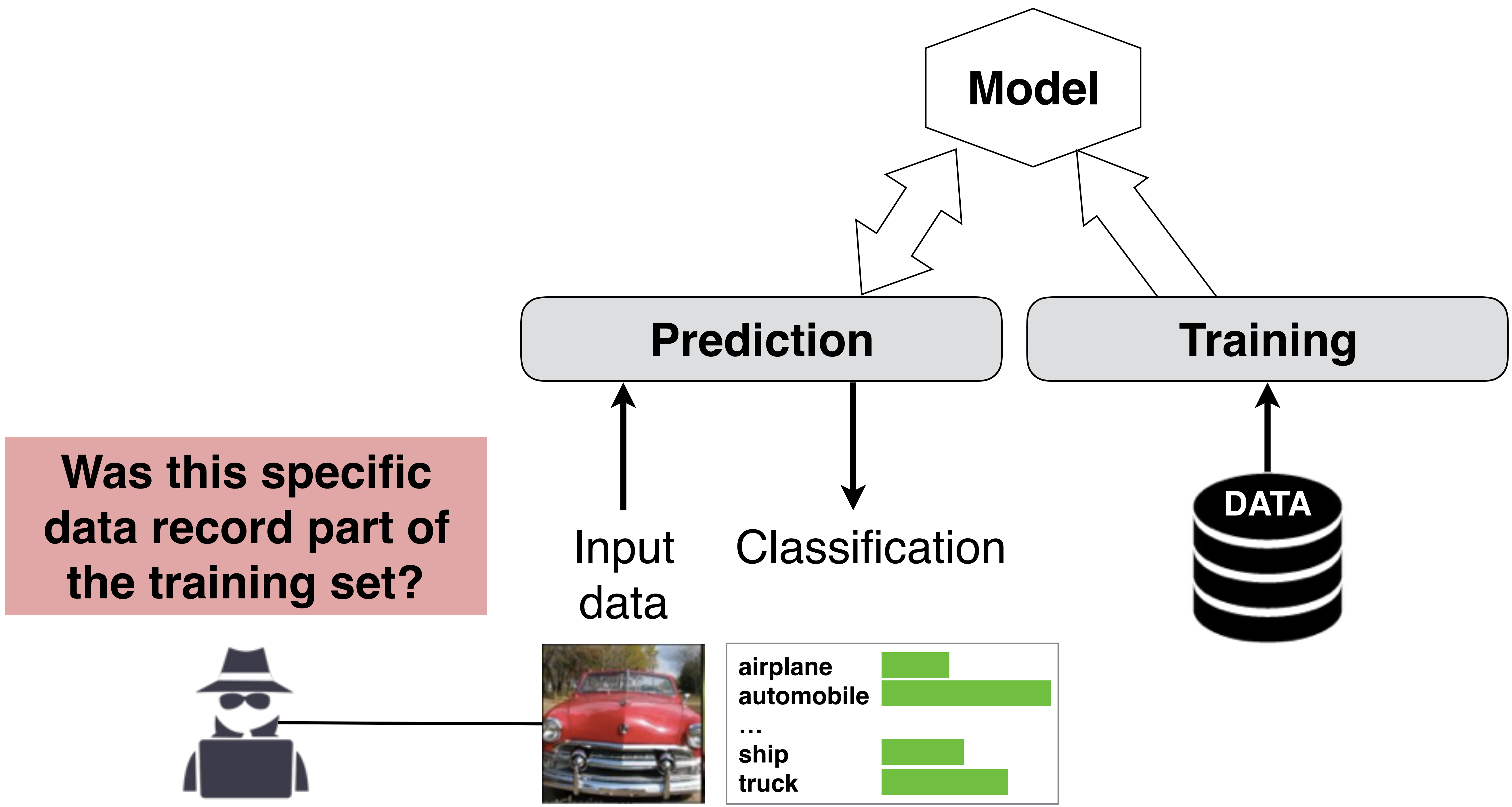


Papernot et al. *Towards the Science of Security and Privacy in Machine Learning*

Privacy Attacks

- ***Privacy attacks*** are also referred to as inference attacks
- They can be developed to reveal information about:
  - Training data
    - Reveal the identity of patients whose data was used for training a model
  - ML model
    - Reveal the architecture and parameters of a model that is used by an insurance company for predicting insurance rates
    - Reveal the model used by a financial institution for credit card approval
- Privacy attacks are commonly divided into the following main categories:
  - Membership inference attack
  - Feature inference attack
  - Model extraction attack

Membership Inference Attack



Membership Inference Attack

on Summary Statistics

- Summary statistics (e.g., average) on each attribute
- Underlying distribution of data is known

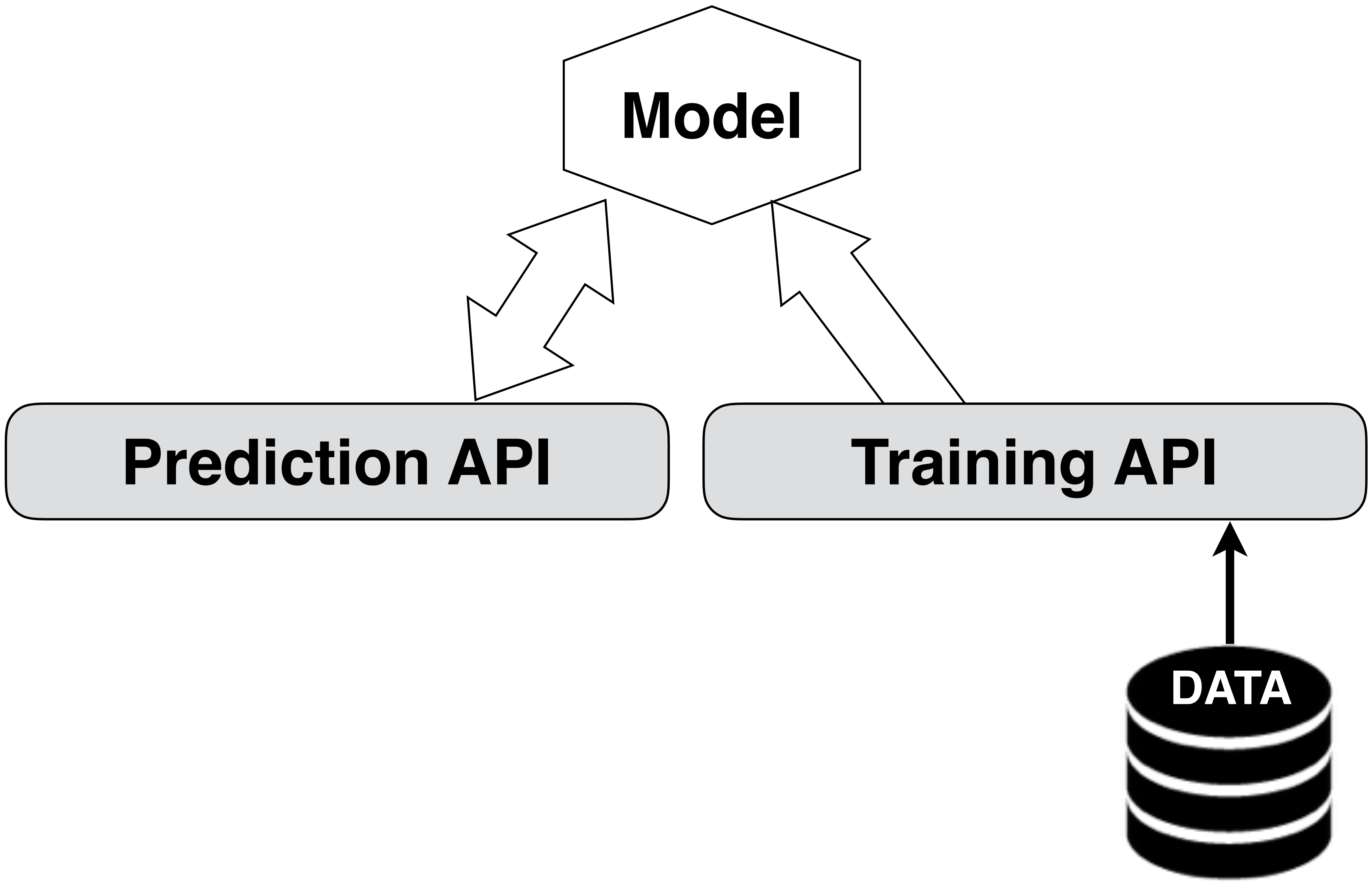
[Homer et al. (2008)], [Dwork et al. (2015)], [Backes et al. (2016)]

on Machine Learning Models

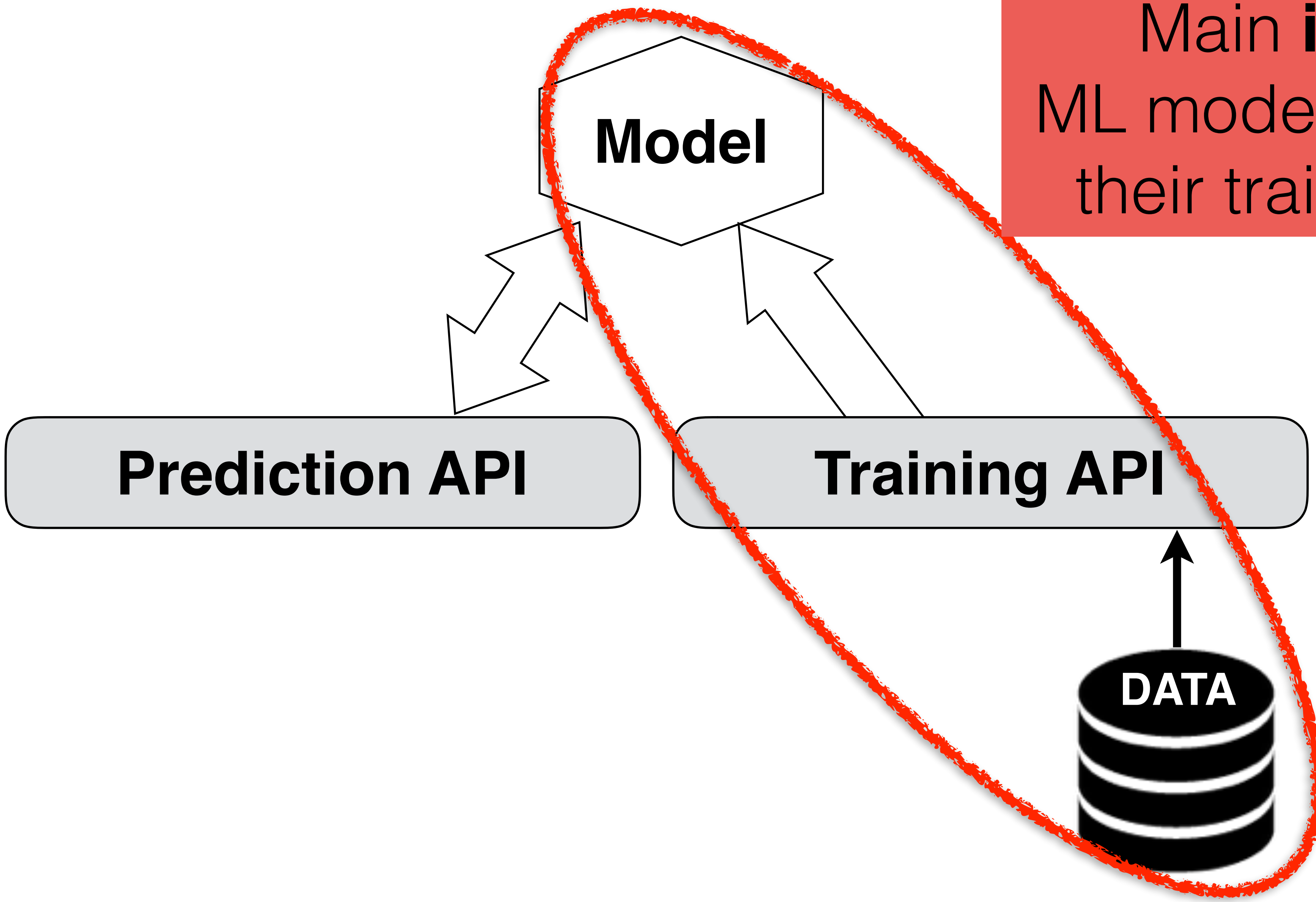
Black-box setting:

- No knowledge about the models' parameters
- No access to internal computations of the model
- No knowledge about the underlying distribution of data

Exploit Model's Predictions



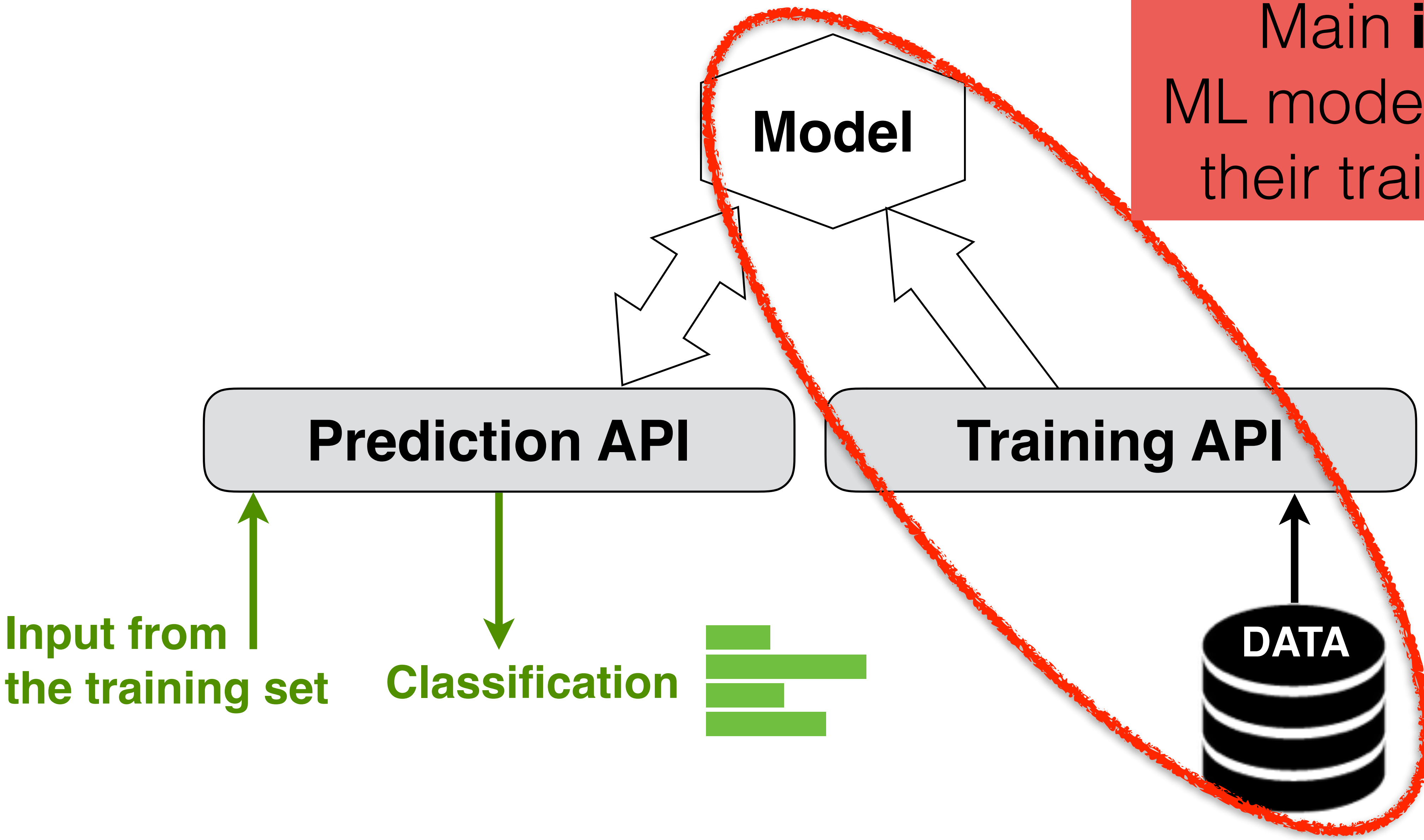
Exploit Model's Predictions



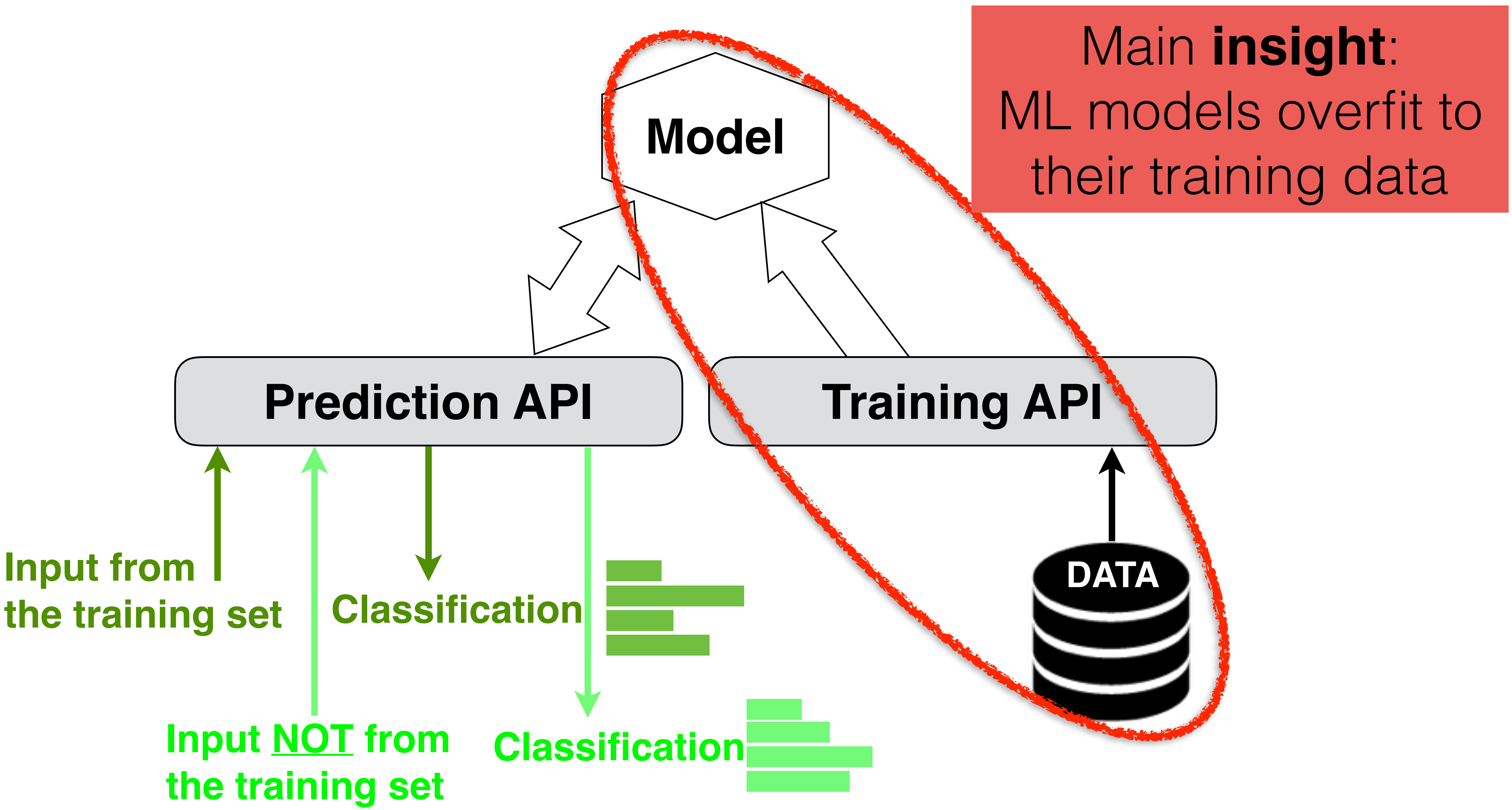
Main **insight**:
ML models overfit to their training data

Exploit Model's Predictions

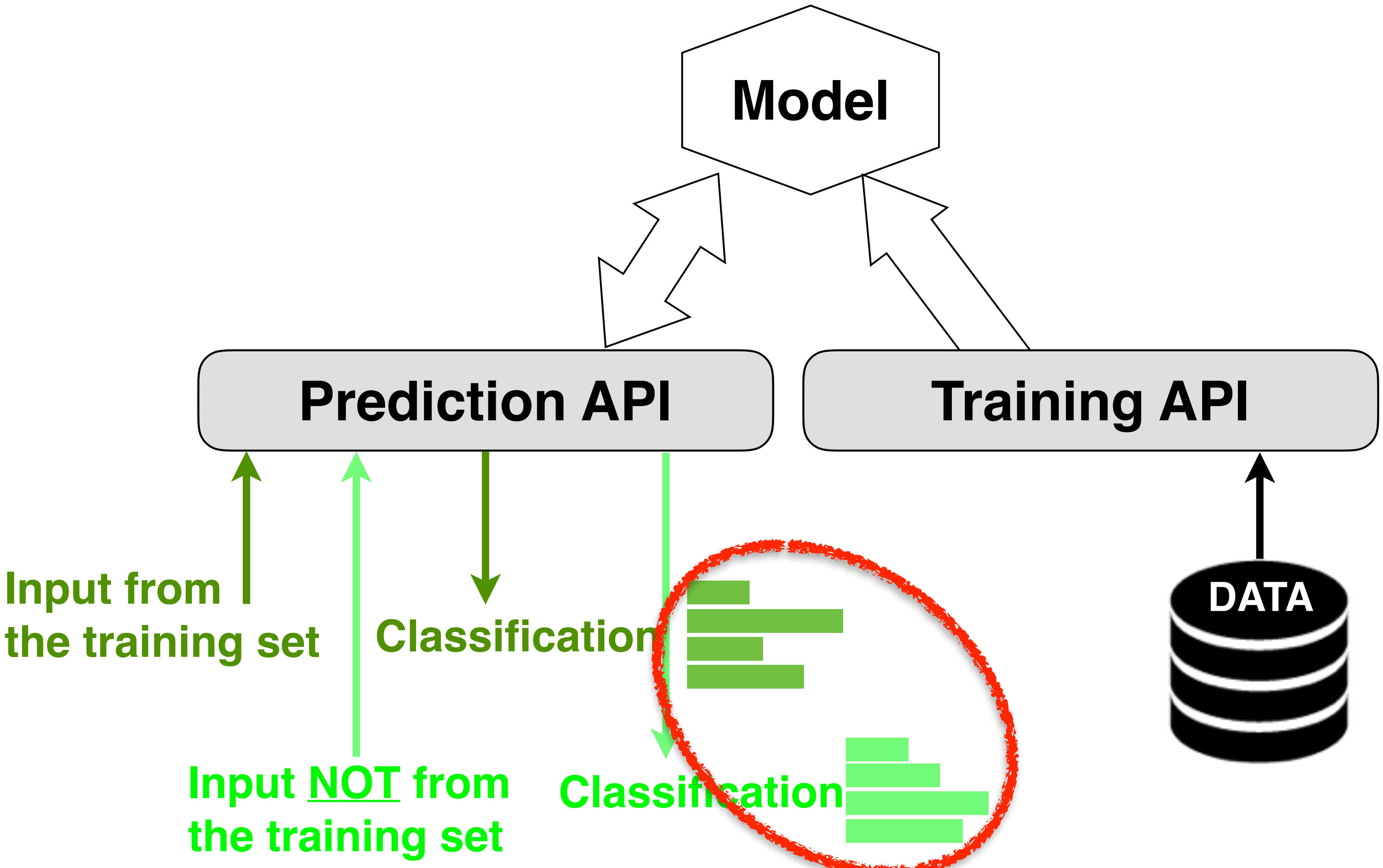
Main **insight**:
ML models overfit to their training data



Exploit Model's Predictions

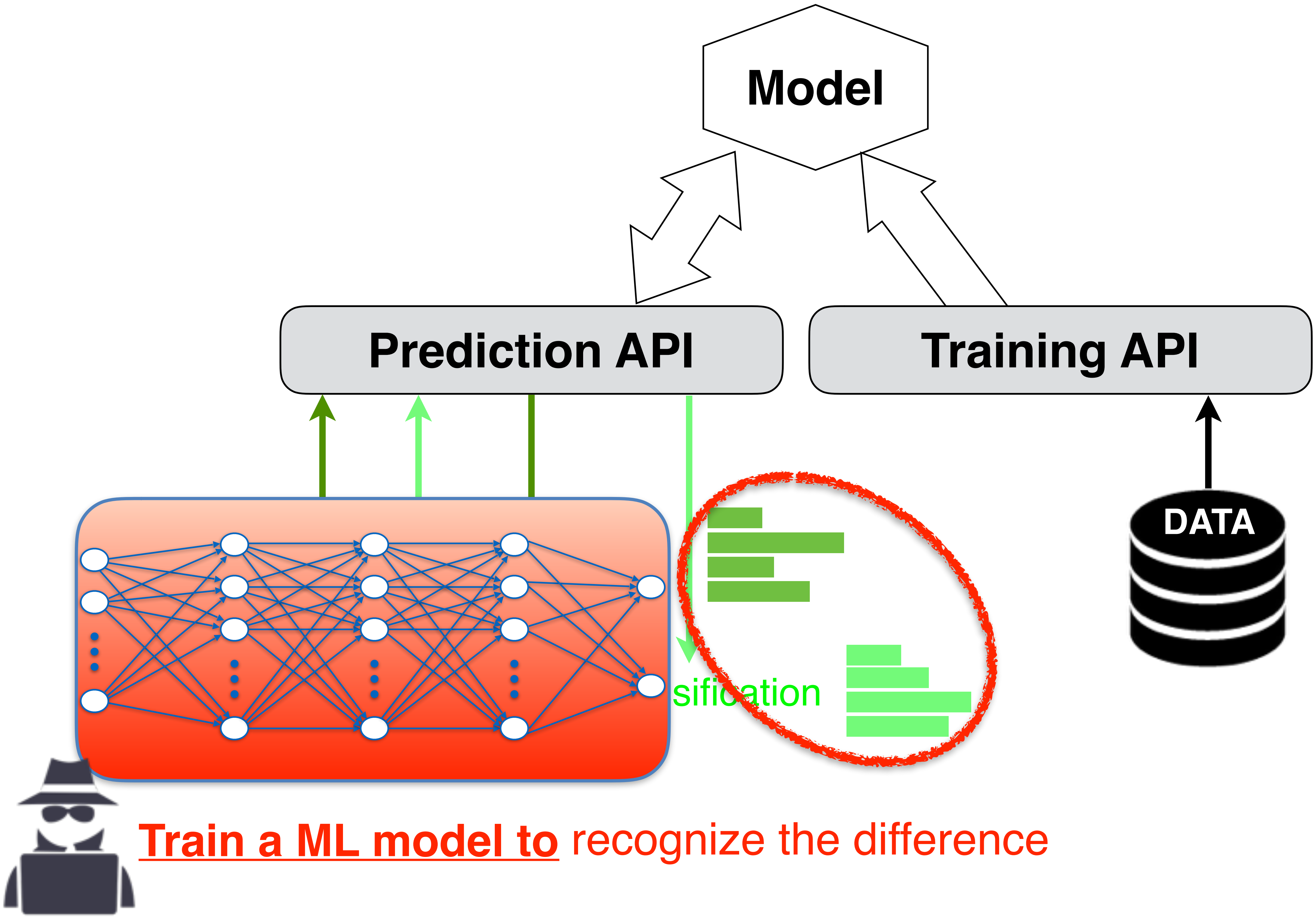


Exploit Model's Predictions



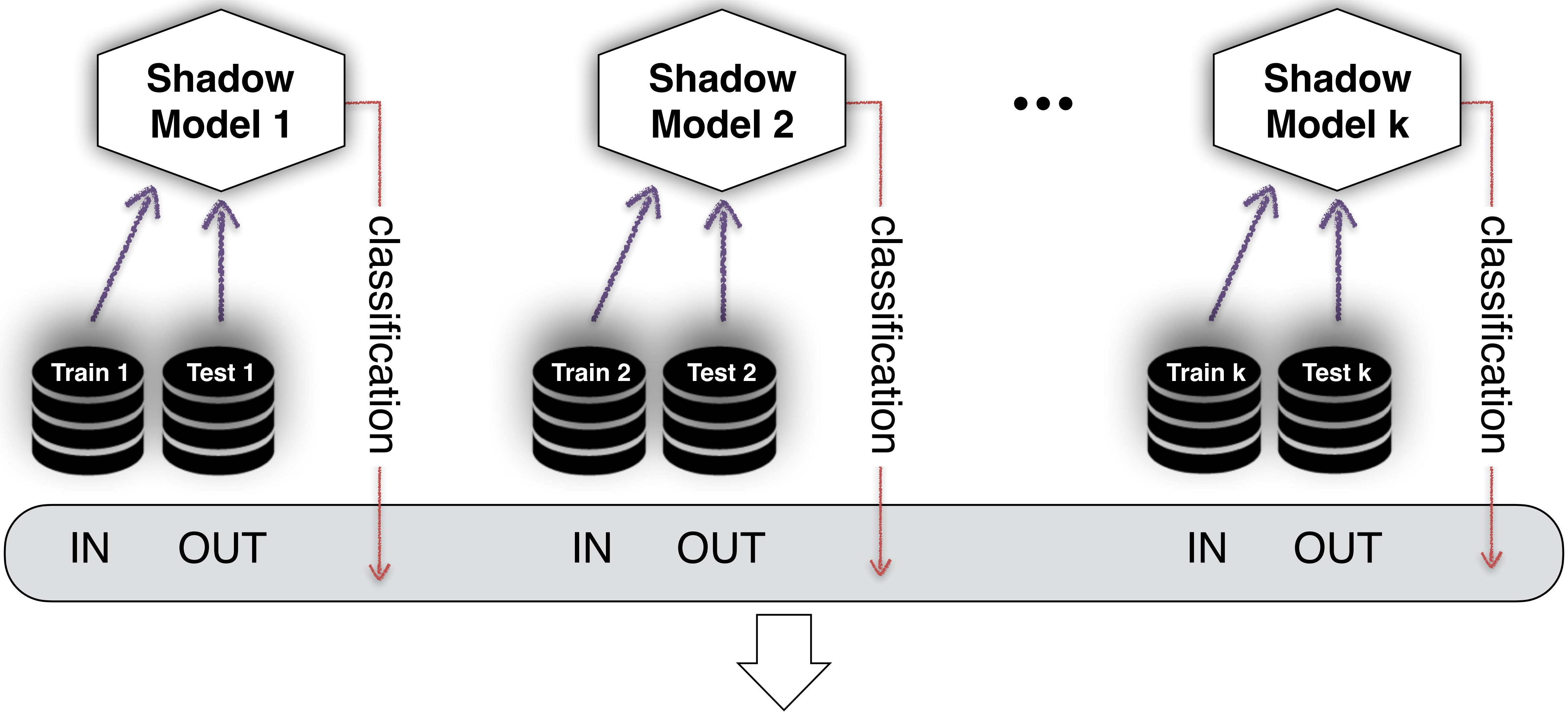
Recognize the difference

ML against ML



Train a ML model to recognize the difference

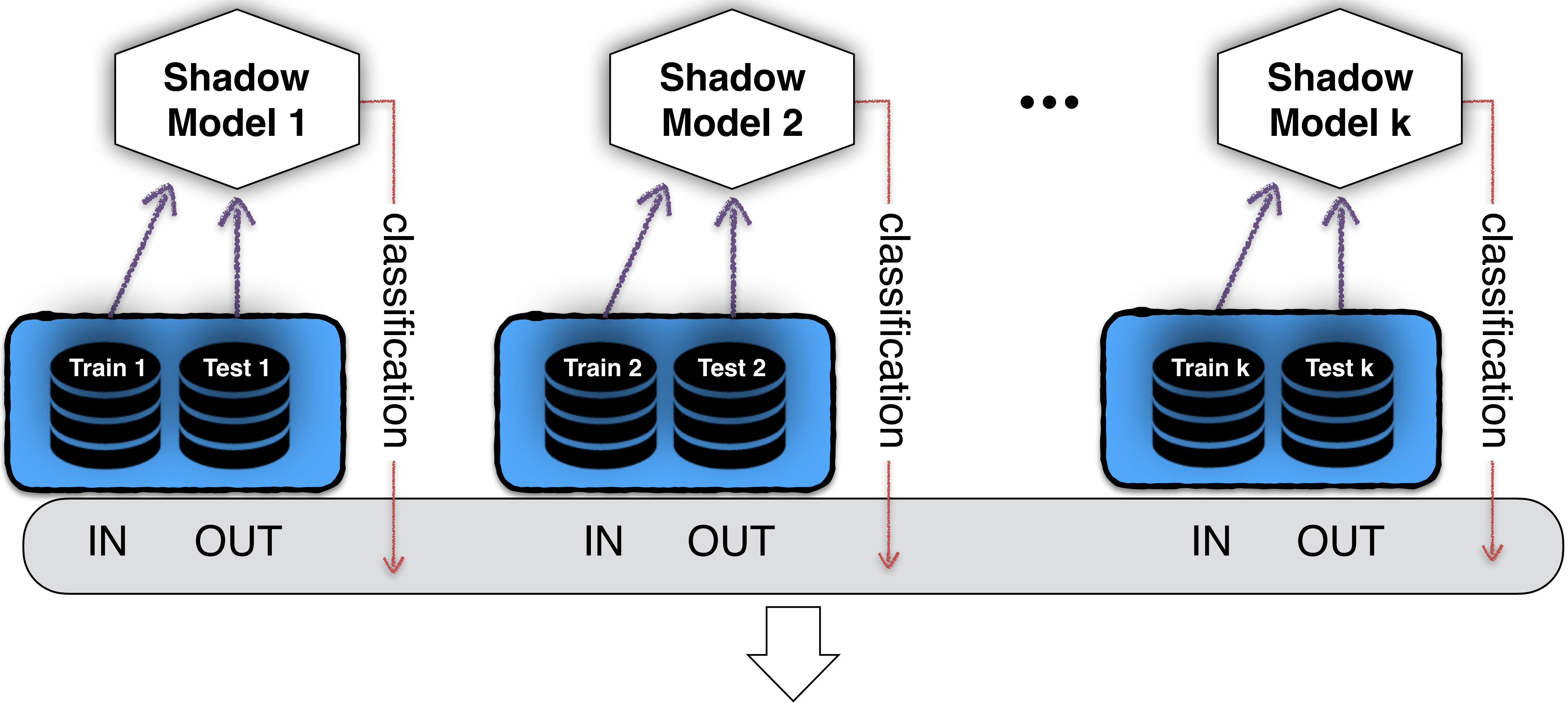
Train Attack Model using **Shadow Models**



Train the attack model

to predict if an input was a member of the training set (in) or a non-member (out)

Train Attack Model using **Shadow Models**



Train the attack model

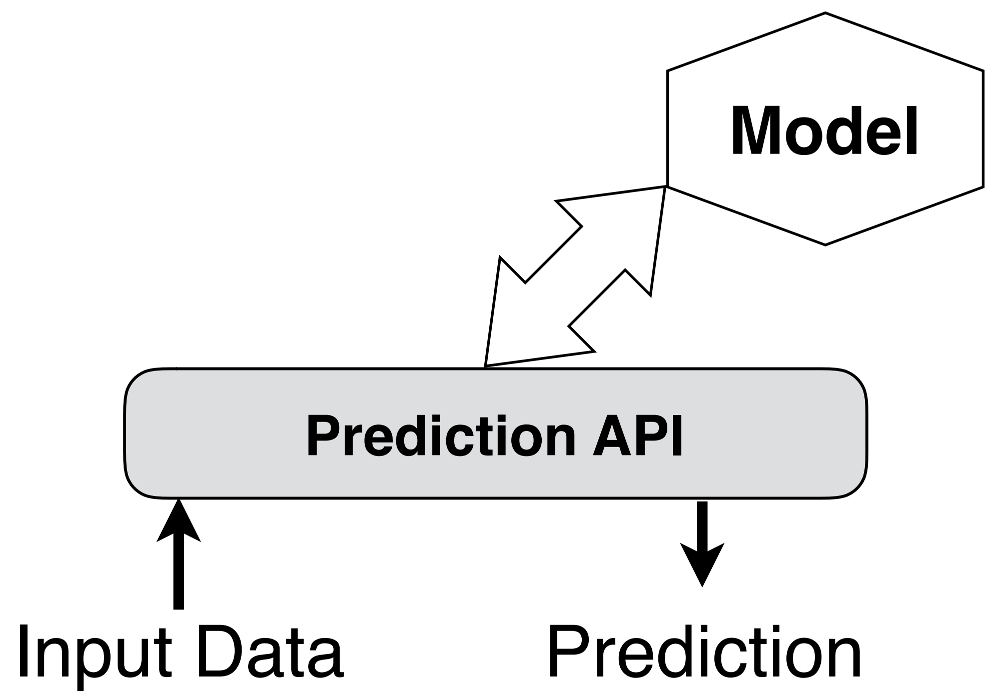
to predict if an input was a member of the training set (in) or a non-member (out)

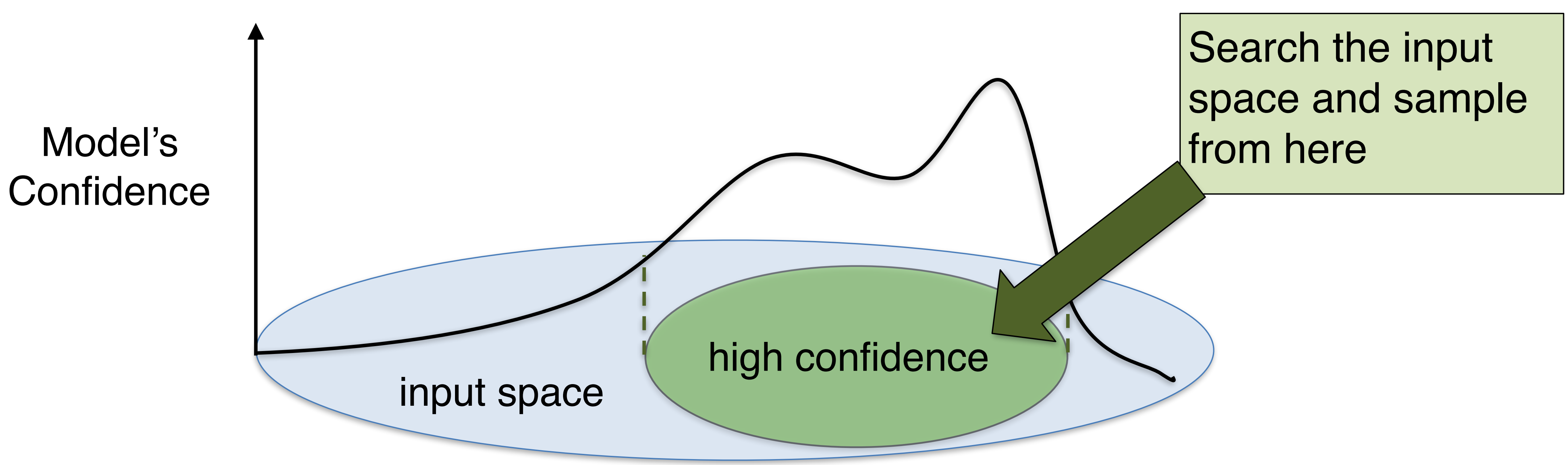
Obtaining Data for Training Shadow Models

- **Real**: similar to training data of the target model (i.e., drawn from same distribution)

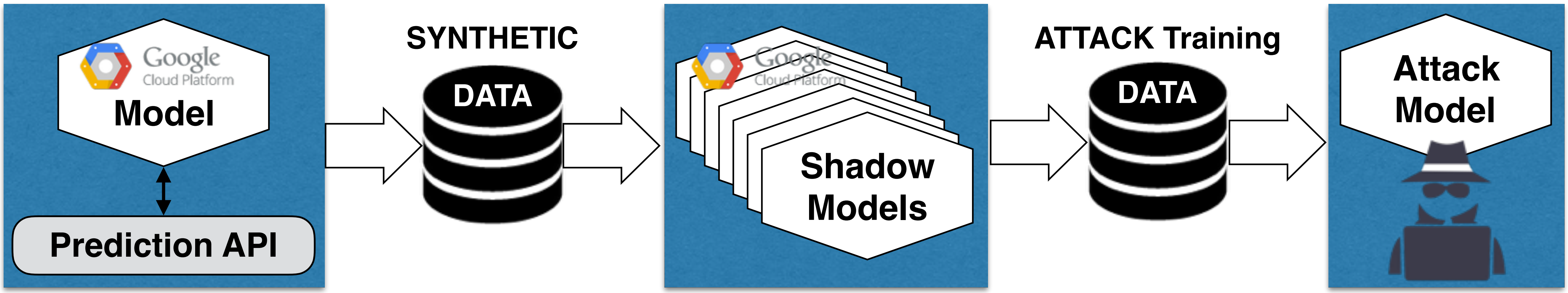
- **Synthetic**: use a sampling algorithm to obtain data classified with high confidence by the target model

Synthesis using the Model

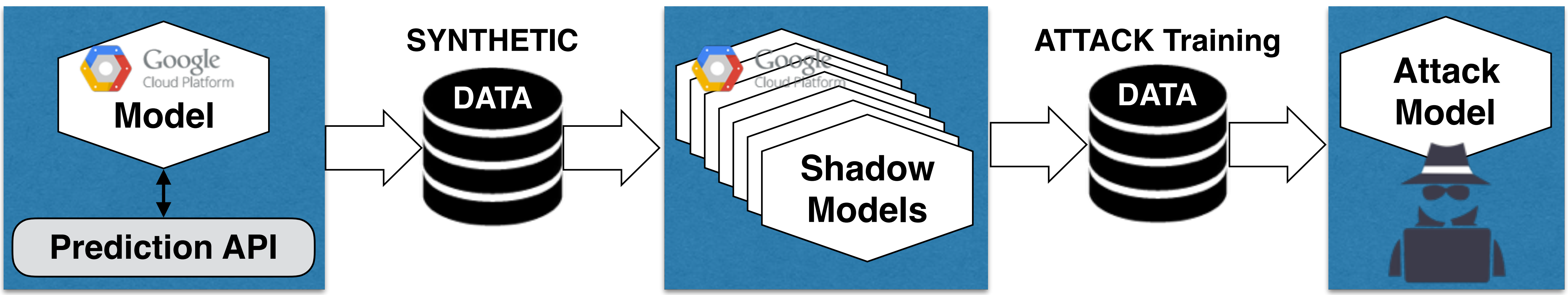




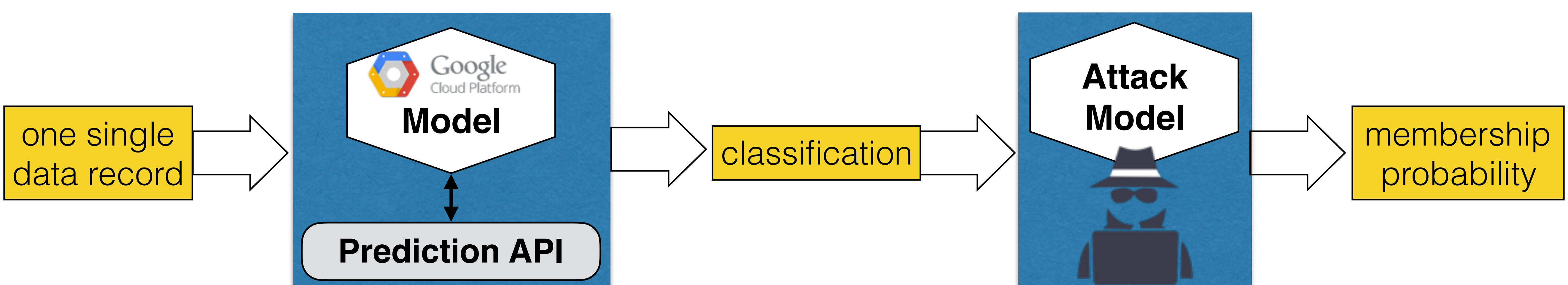
Constructing the Attack Model

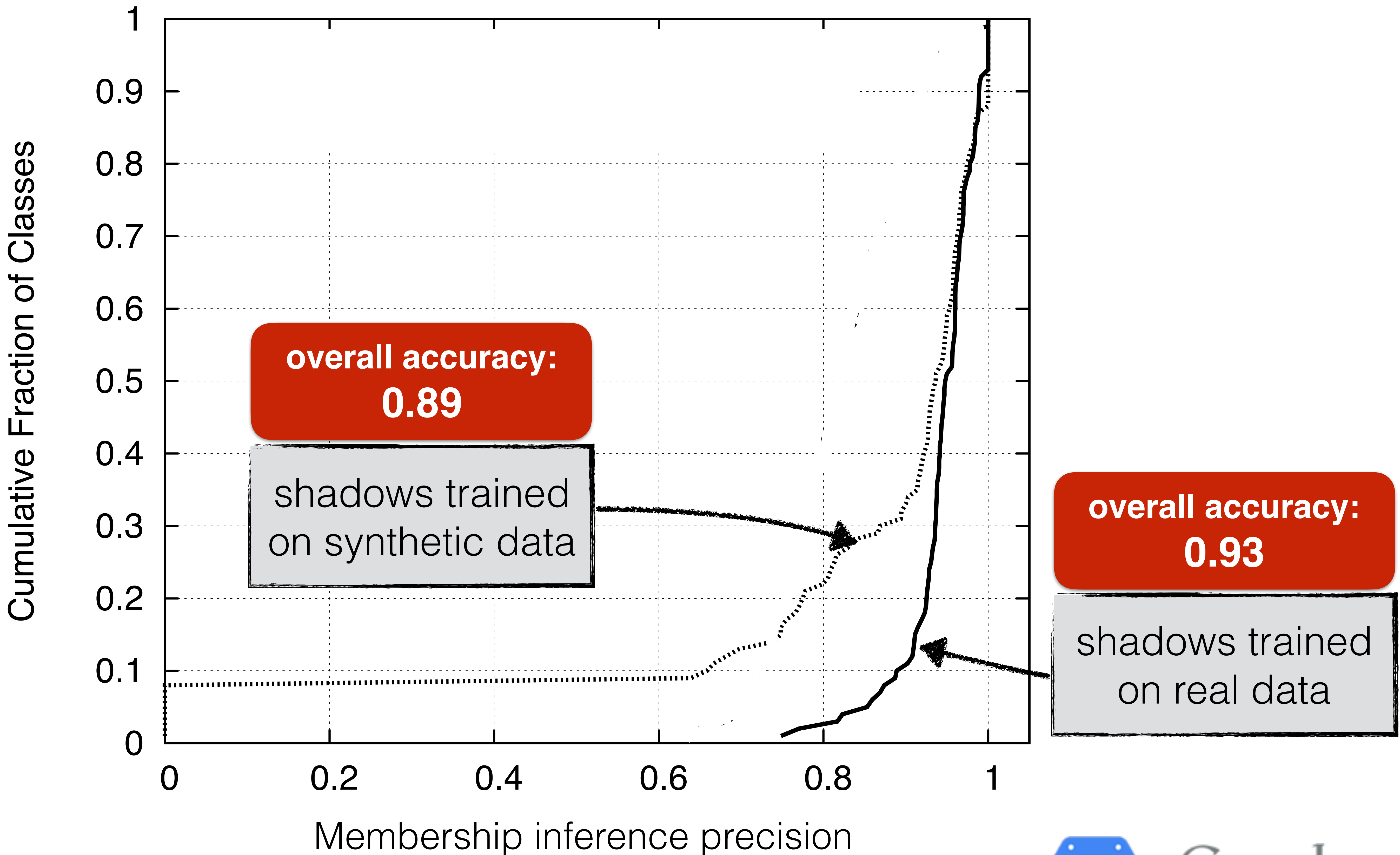


Constructing the Attack Model



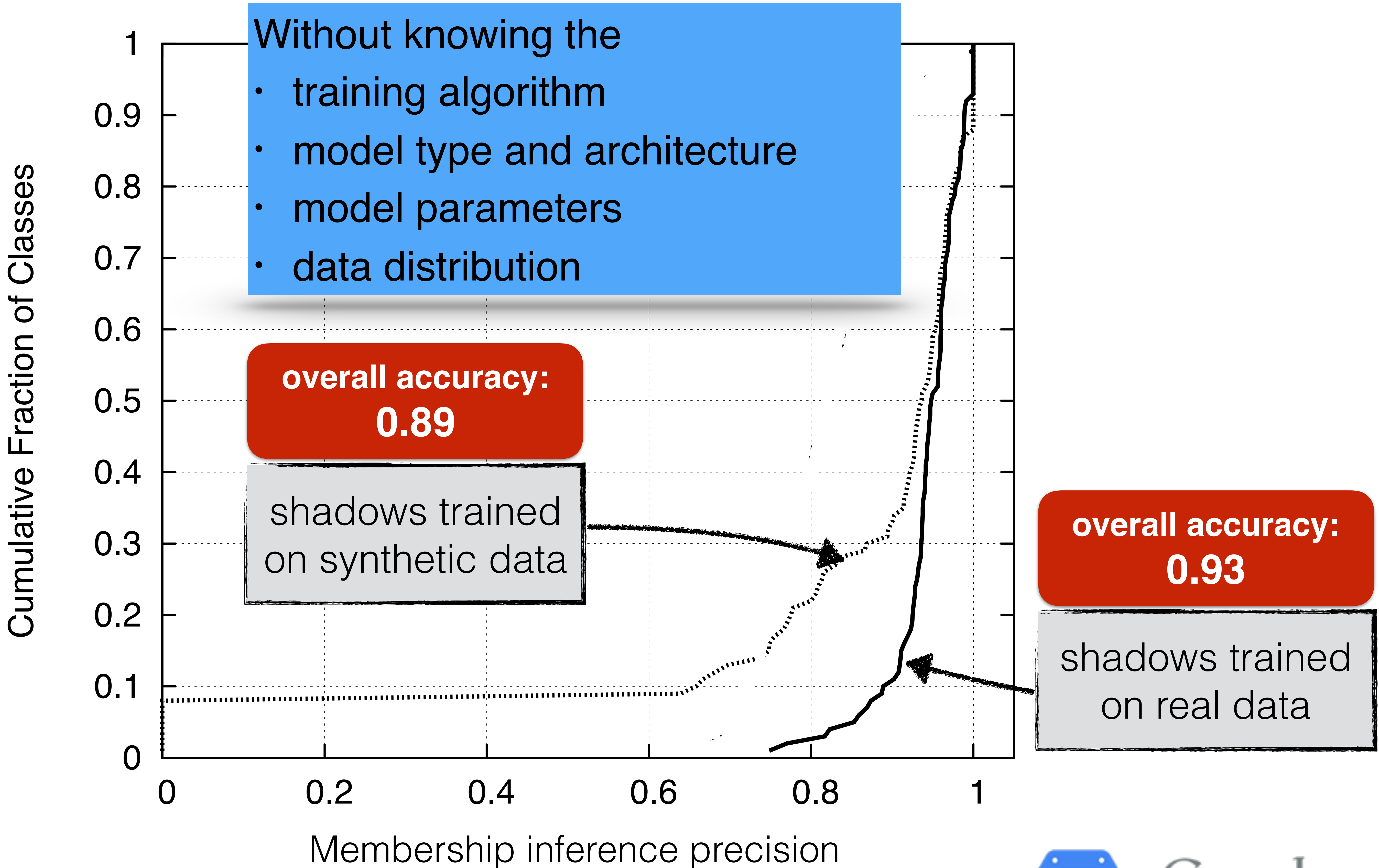
Using the Attack Model





Purchase Dataset — Classify Customers (100 classes)

Google Cloud Platform



Purchase Dataset — Classify Customers (100 classes)

Google Cloud Platform

Let’s Talk About *Model Inversion!*

- A trained ML model with parameters **w** is released to the public
  - **W** = training_procedure(X)
  - Training data X is hidden
- Can we *recover* some of X just through access to **w**?
  - X’ = training_procedure^{-1} (**w**) <--- notational abuse
  - That would be bad
- Intersection of security and privacy

Model Inversion Attack

- [Fredrickson (2015) Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures]
- ***Model inversion attack*** creates prototype examples for the classes in the dataset
  - The authors demonstrated an attack against a DNN model for face recognition
  - Given a person’s name and white-box access to the model, the attack reverse-engineered the model and produced an averaged image of that person
    - The obtained averaged image (left image below) makes the person recognizable
  - This attack is limited to classification models where each class only contain one type of object (such as faces of the same person)

Recovered image using the model inversion attack

Image of the person used for training the model

Model Inversion Attack

- The model inversion attack applies gradient descent to start from a given label, and follow the gradient in a trained network to recreate an image for that label
  - Minimize the cost function *c*, whereas the PROCESS function applies image denoising and sharpening operations to improve the reconstructed image
- Model inversion attack can be used for potential breaches where the adversary, given some access to the model, can infer features that characterize each class

Algorithm 1 Inversion attack for facial recognition models.

function MI-FACE($label, \alpha, \beta, \gamma, \lambda$)
    $\boxed{c(\mathbf{x}) \stackrel{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})}$
    $\mathbf{x}_0 \leftarrow \mathbf{0}$
    for $i \leftarrow 1 \ldots \alpha$ **do**
        $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$
        if $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \ldots, c(\mathbf{x}_{i-\beta}))$ **then**
            break
        if $c(\mathbf{x}_i) \leq \gamma$ **then**
            break
    return $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$

Maximize the logit of the label's class

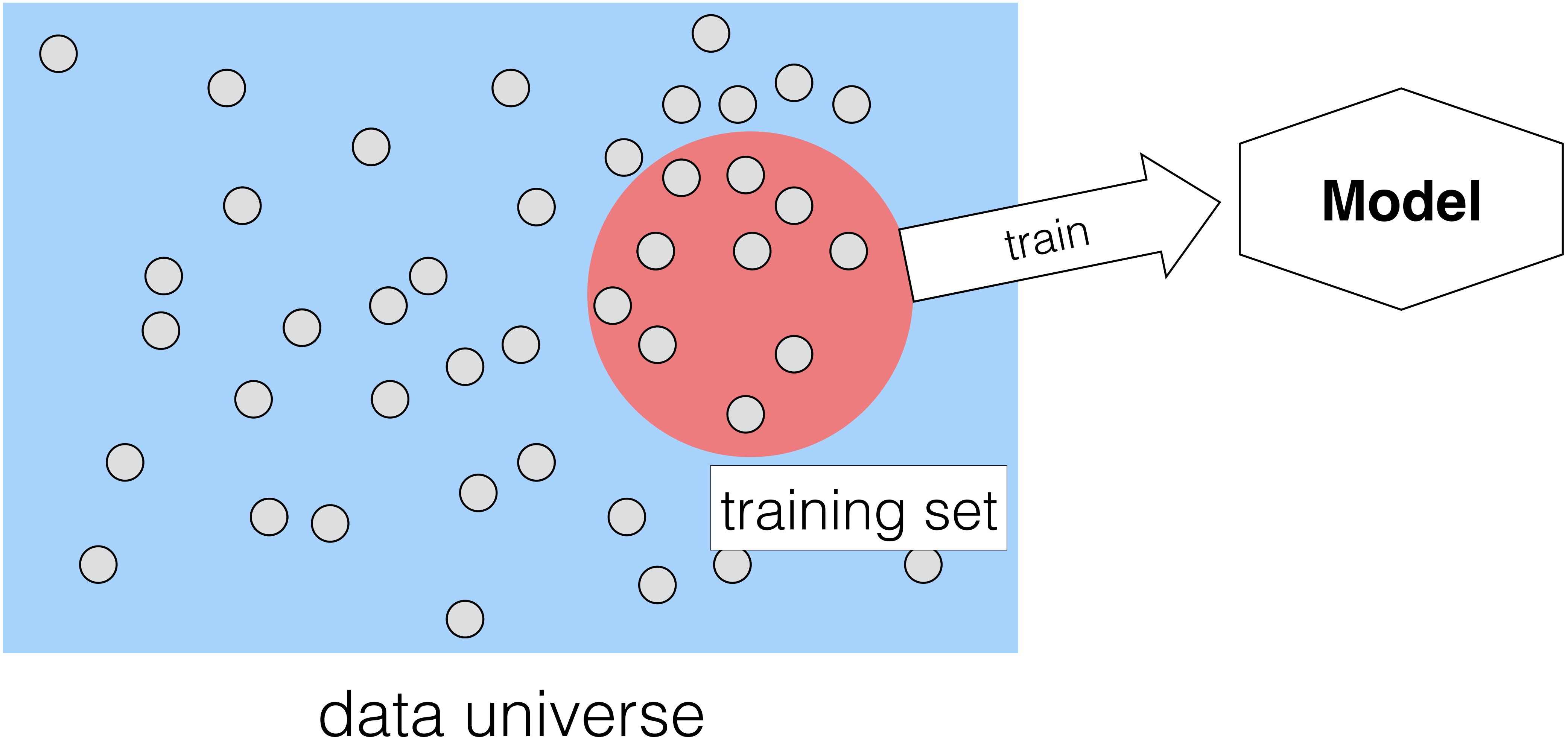
Model Extraction Attack

- ***Model extraction attack***
  - Goal: reconstruct an approximated model $f'(x)$ of the target model $f(x)$
- A.k.a. model inference attack
- The approximated function $f'(x)$ will act as a substitute model and produce similar predicted outputs as the target model
  - The adversary ha…
  - The goal is to "steal" the model and use the substitute model for lunching other attacks, such as synthesis of adversarial examples, or membership inference attacks
- Besides creating a substitute model, several works focused on recovering the hyperparameters of the model, such as the number of layers, optimization algorithm, activation function, etc.

What causes privacy leakage?

Figure form: Liu et al. (2020) When Machine Learning Meets Privacy: A Survey and Outlook

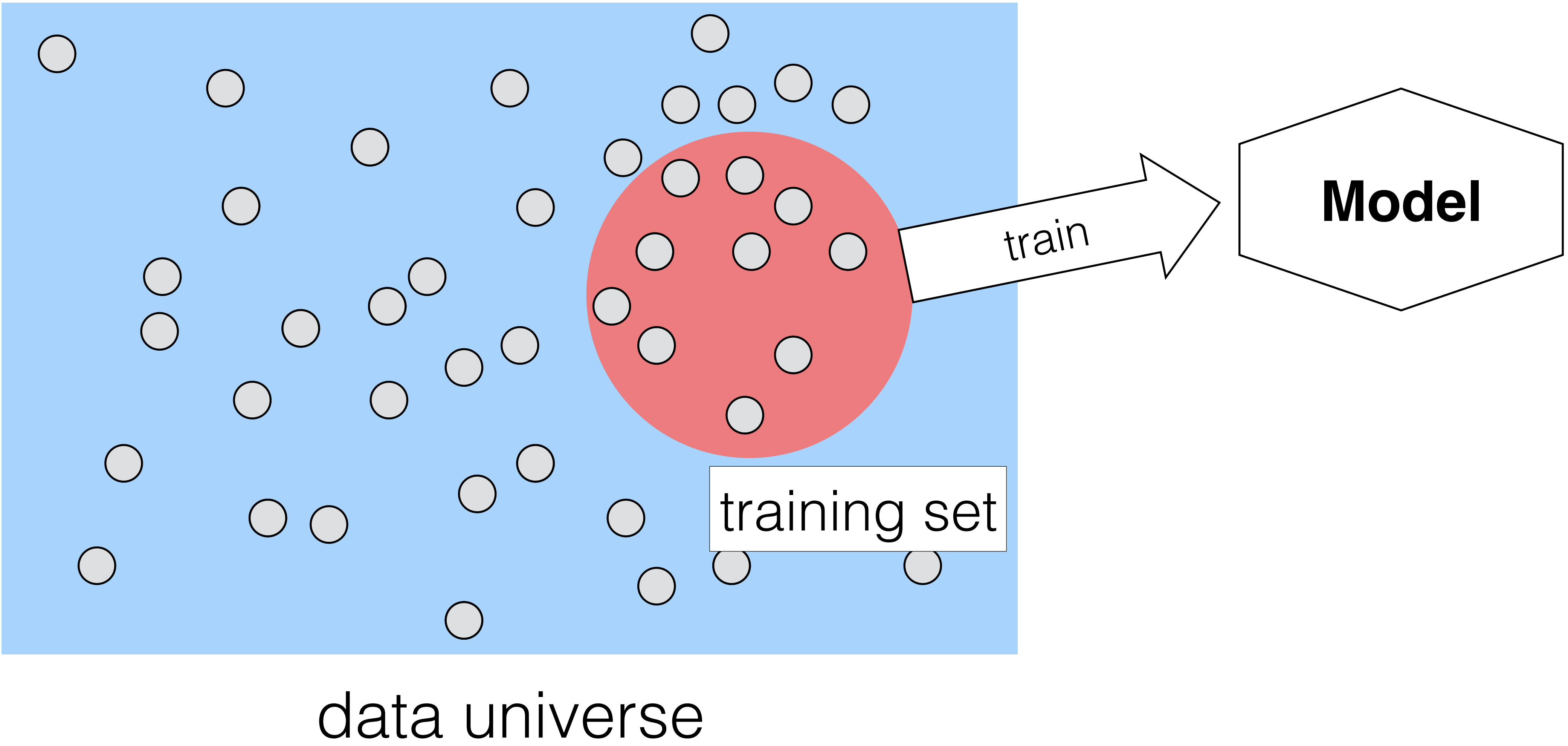
Privacy Learning



Privacy

Learning

Does the model leak information about data in the training set?

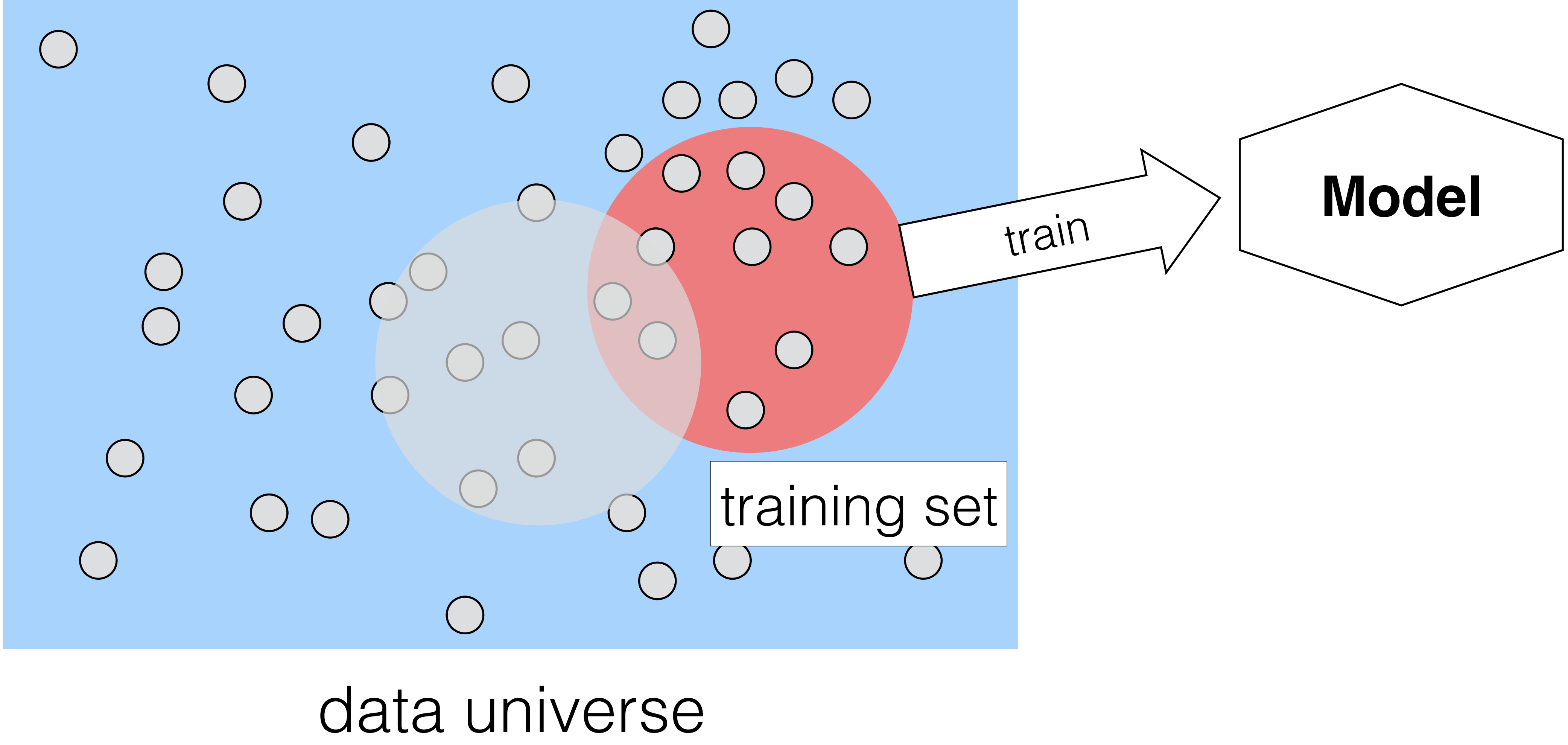


Privacy

Does the model leak information about data in the training set?

Learning

Does the model generalize to data outside the training set?

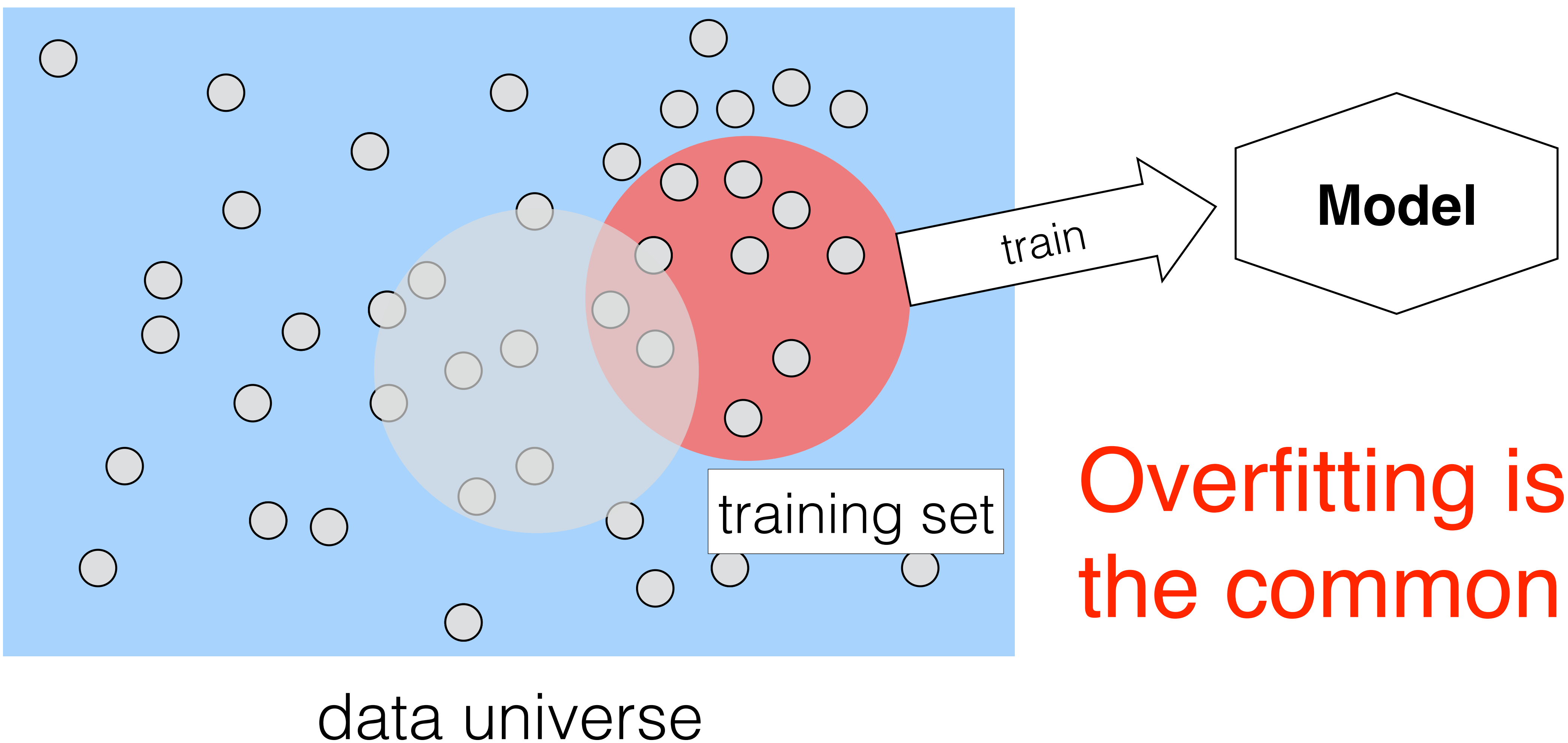


Privacy

Does the model leak information about data in the training set?

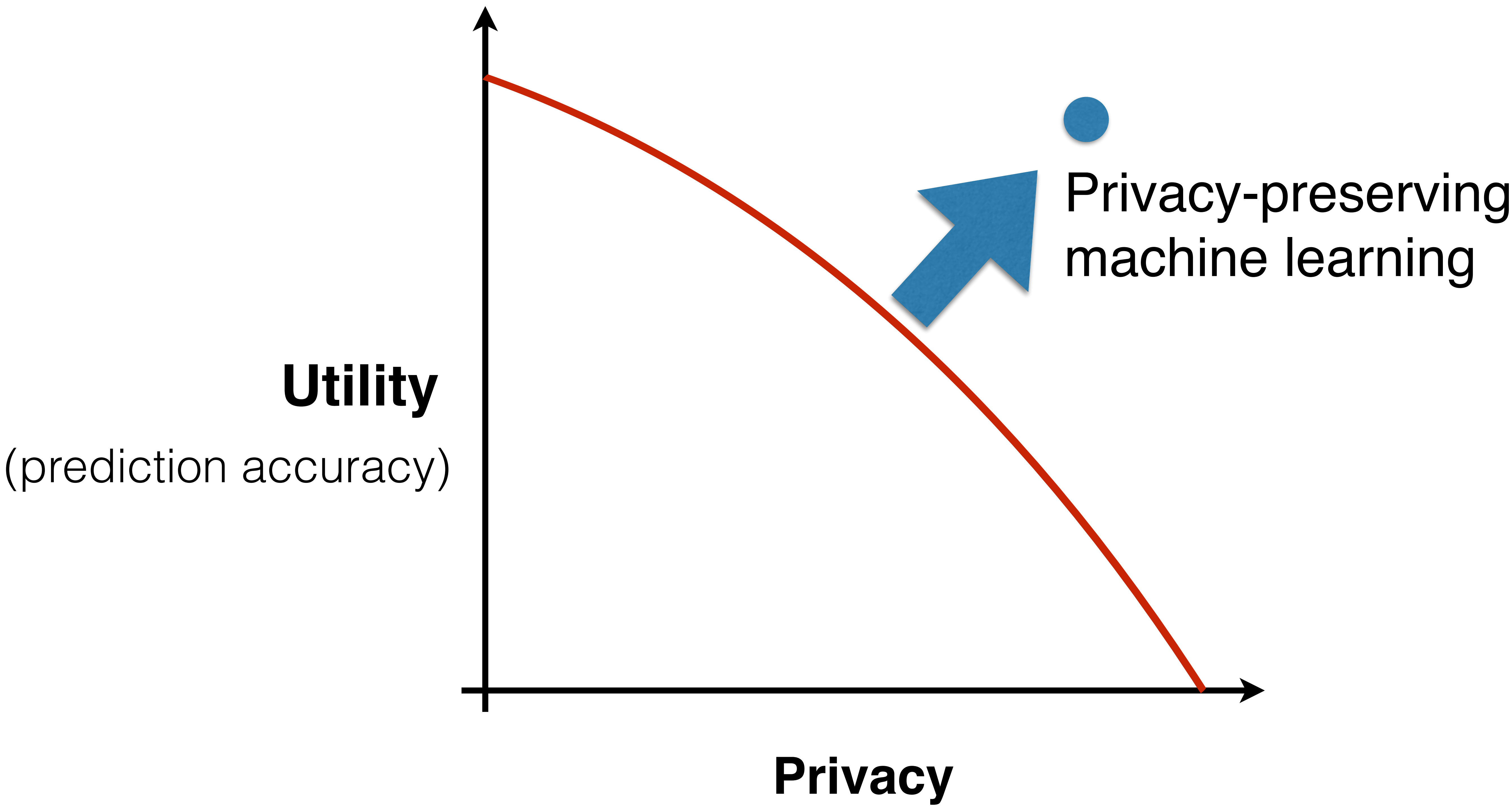
Learning

Does the model generalize to data outside the training set?



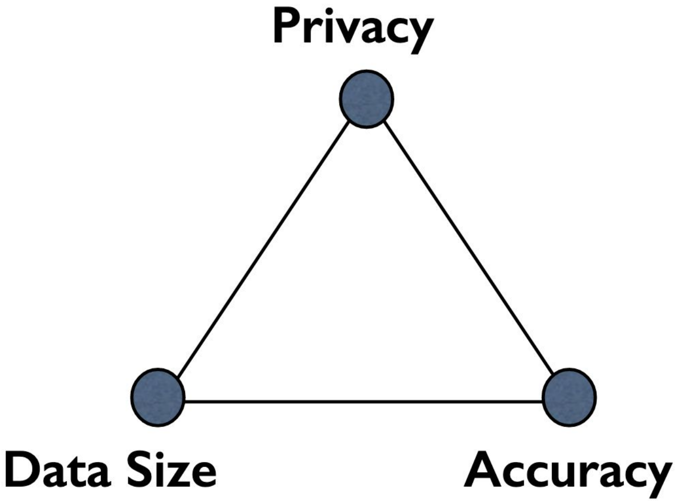
Overfitting is the common enemy!

Not in a Direct Conflict!



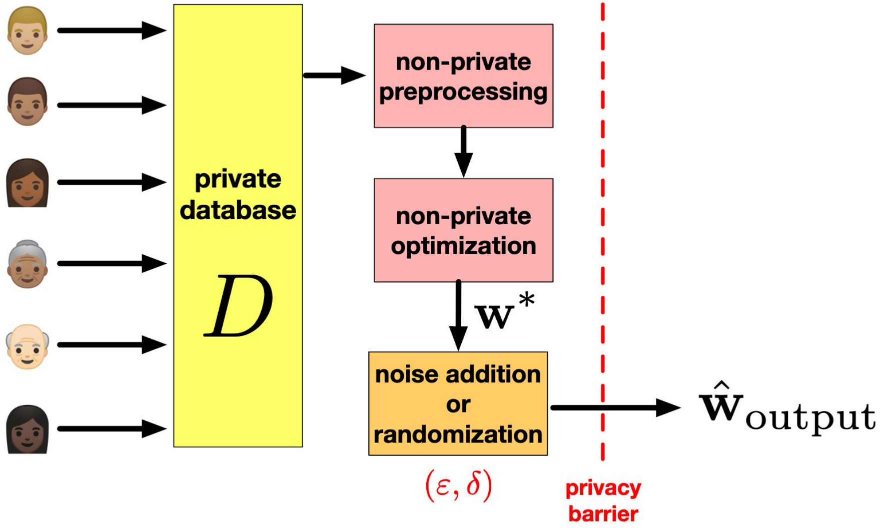
Part 2: Making ML Private

Tradeoffs in DP+ML



[Chaudhuri & Sarwate 2017 NIPS tutorial]

Output Perturbation



- Compute the minimizer and add noise.

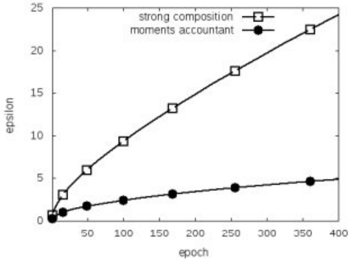
- Does not require re-engineering baseline algorithms

Noise depends on the sensitivity of the argmin.

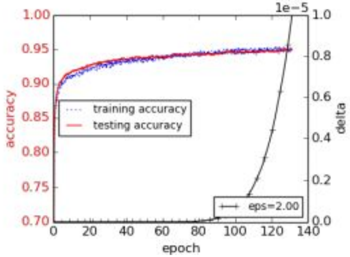
[CMS11, RBHT12]

Differentially Private SGD

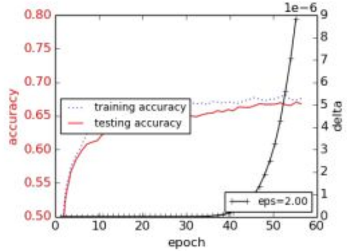
Moments accountant improves bounds



MNIST epoch vs accuracy/privacy



CIFAR-10 epoch vs accuracy/privacy



Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

[Abadi et al. 2016]

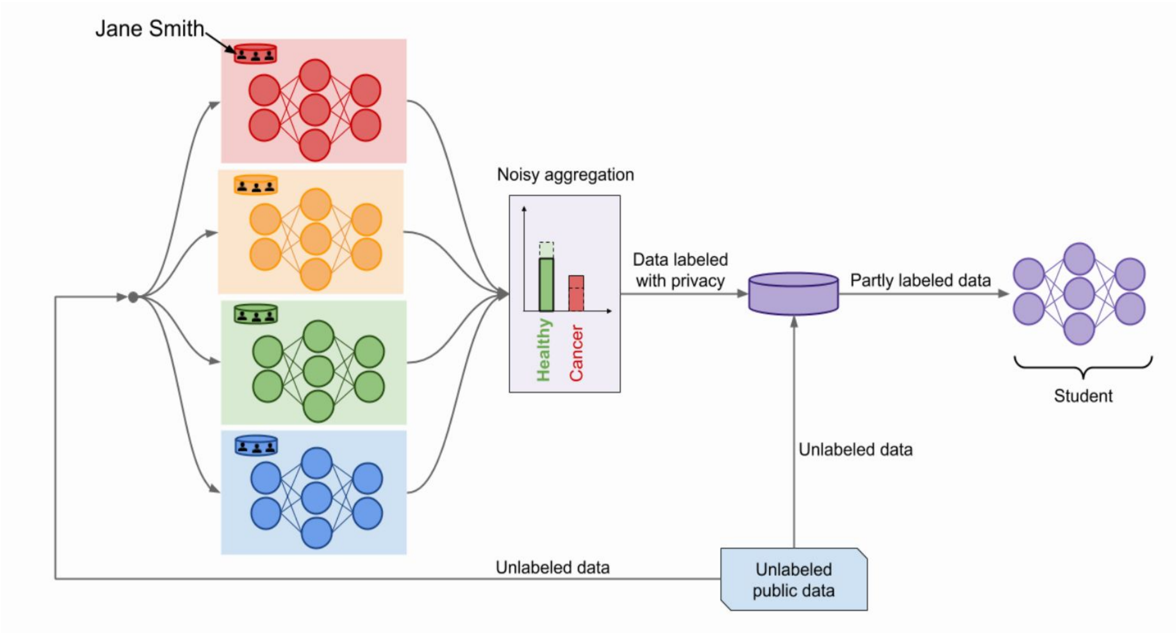
Guarantees final parameters don't depend too much on individual training examples

Gaussian noise added to the parameter update at every iteration

Privacy loss accumulates over time

The "moments accountant" provides better empirical bounds on (ε,δ)

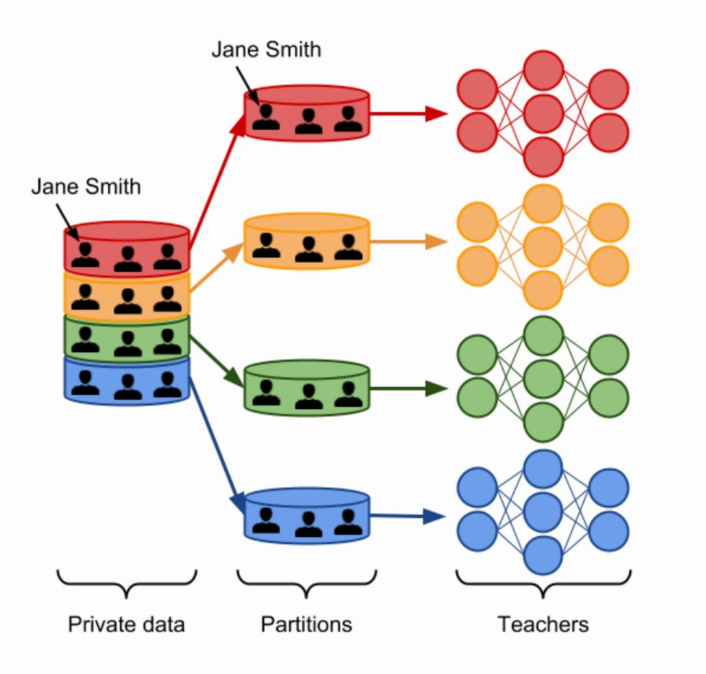
PATE



Private Aggregation of Teacher Ensembles [Papernot et al 2017, Papernot et al 2018]

Key idea: instead of adding noise to gradients, add noise to *labels*

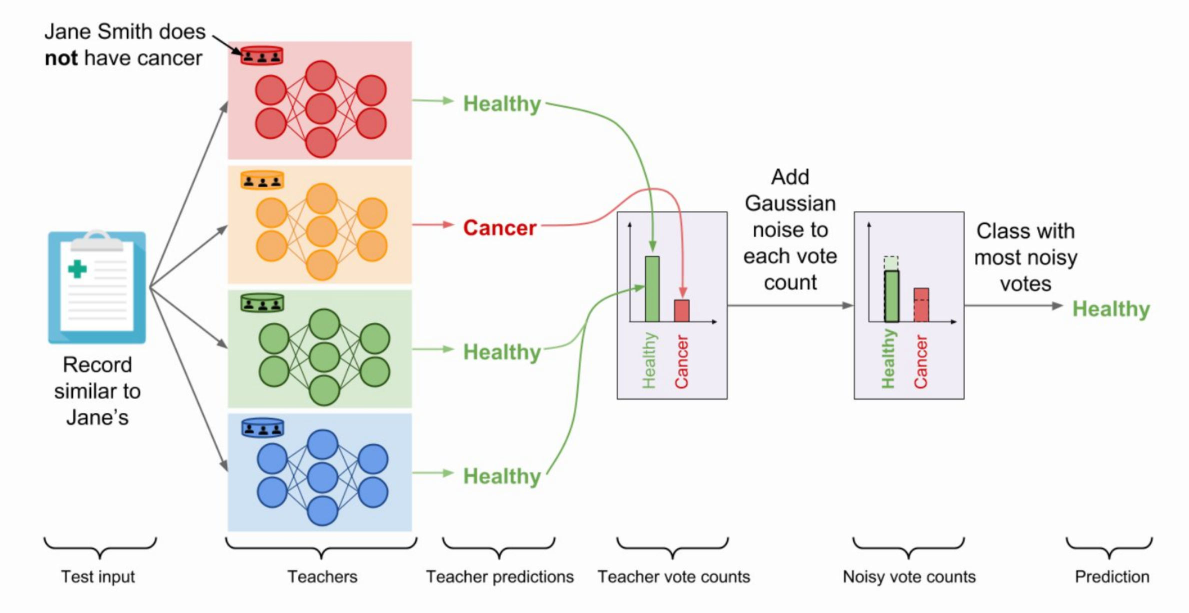
PATE



Start by partitioning private data into disjoint sets

Each teacher trains (non-privately) on its corresponding subset

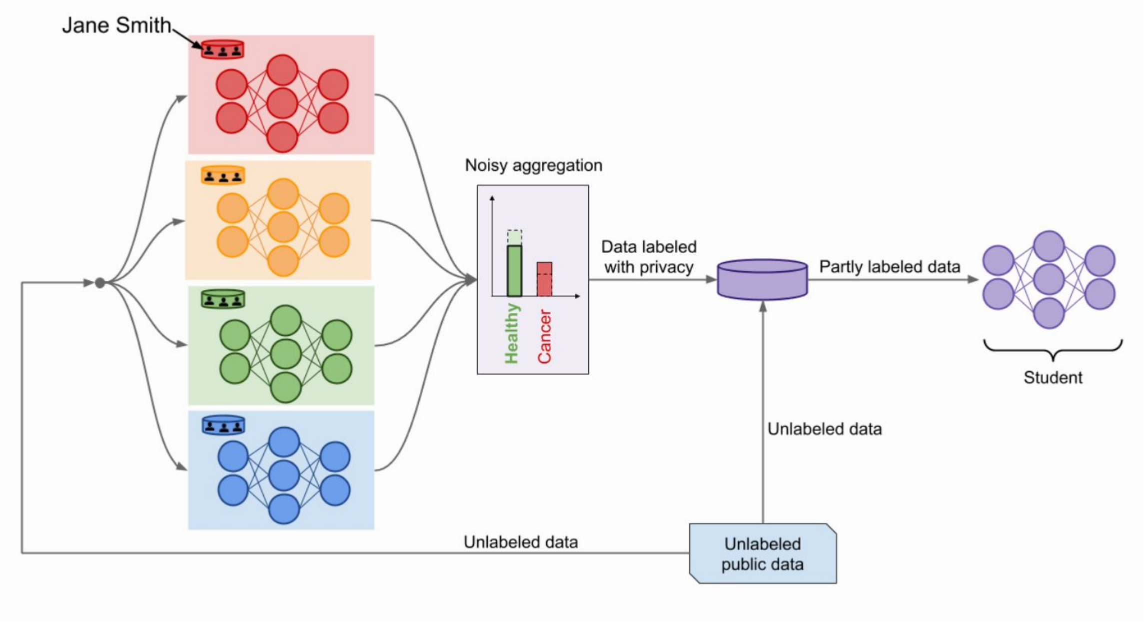
PATE



Private predictions can now be generated via the exponential mechanism, where the "score" is computed with an election amongst teachers - output the noisy winner

We now have private inference, but we lose privacy every time we predict. We would like the privacy loss to be constant at test time.

PATE



We can instead use the noisy labels provided by the teachers to train a student

We leak privacy during training but at test time we lose no further privacy (due to post-processing thm)

Because the student should use as few labels as possible, unlabeled public data is leveraged in a semi-supervised setup.

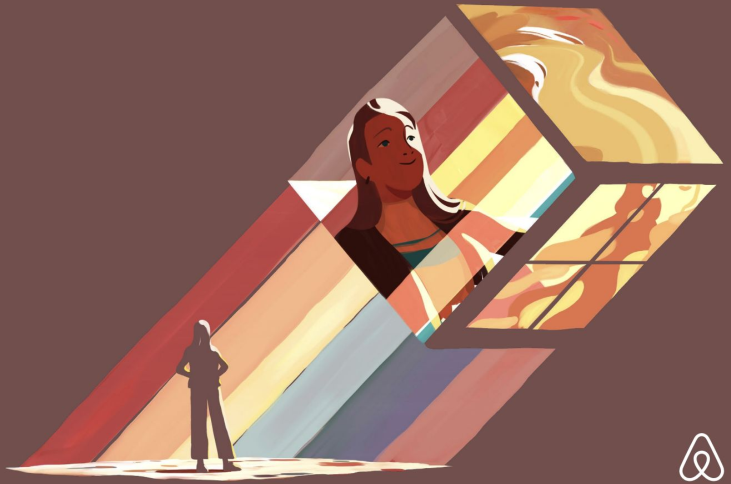
Part 3: Case Study - AirBnB Project Lighthouse

Northeastern University


A new way we're fighting discrimination on Airbnb
Copy link
Watch on YouTube

The Airbnb anti-discrimination team

Measuring discrepancies in Airbnb guest acceptance rates using anonymized demographic data



Three Types of Disclosure Threats

row_id	n_accept	n_reject	perceived race
1	1	1	black
2	1	2	white
3	2	1	black
4	2	1	white
5	11	1	black

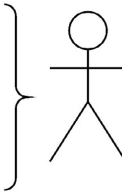


Figure 1: Example membership disclosure

row_id	n_accept	n_reject	perceived race
1	1	1	black
2	1	2	**white**
3	2	1	black
4	2	1	**white**
5	11	1	black

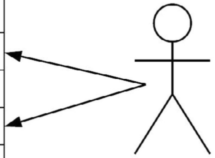


Figure 2: Example attribute disclosure

row_id	n_accept	n_reject	perceived race
1	1	1	black
2	1	2	white
3	2	1	black
4	**2**	**1**	**white**
5	11	1	black

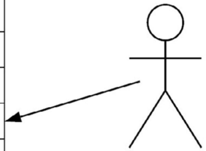


Figure 3: Example identity disclosure

Northeastern University

Data Flow Diagram

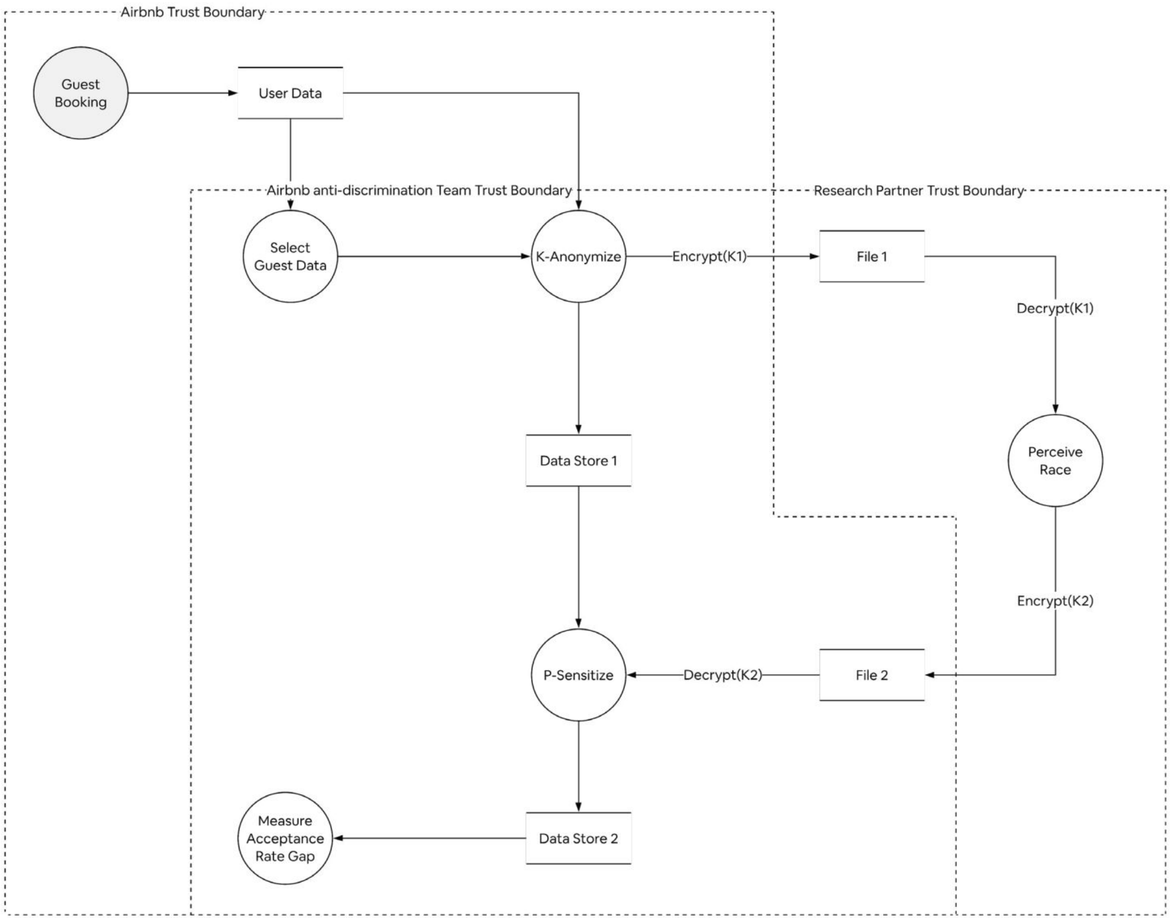


Figure 5: Simplified Data Flow Diagram (DFD)

Northeastern University

Security

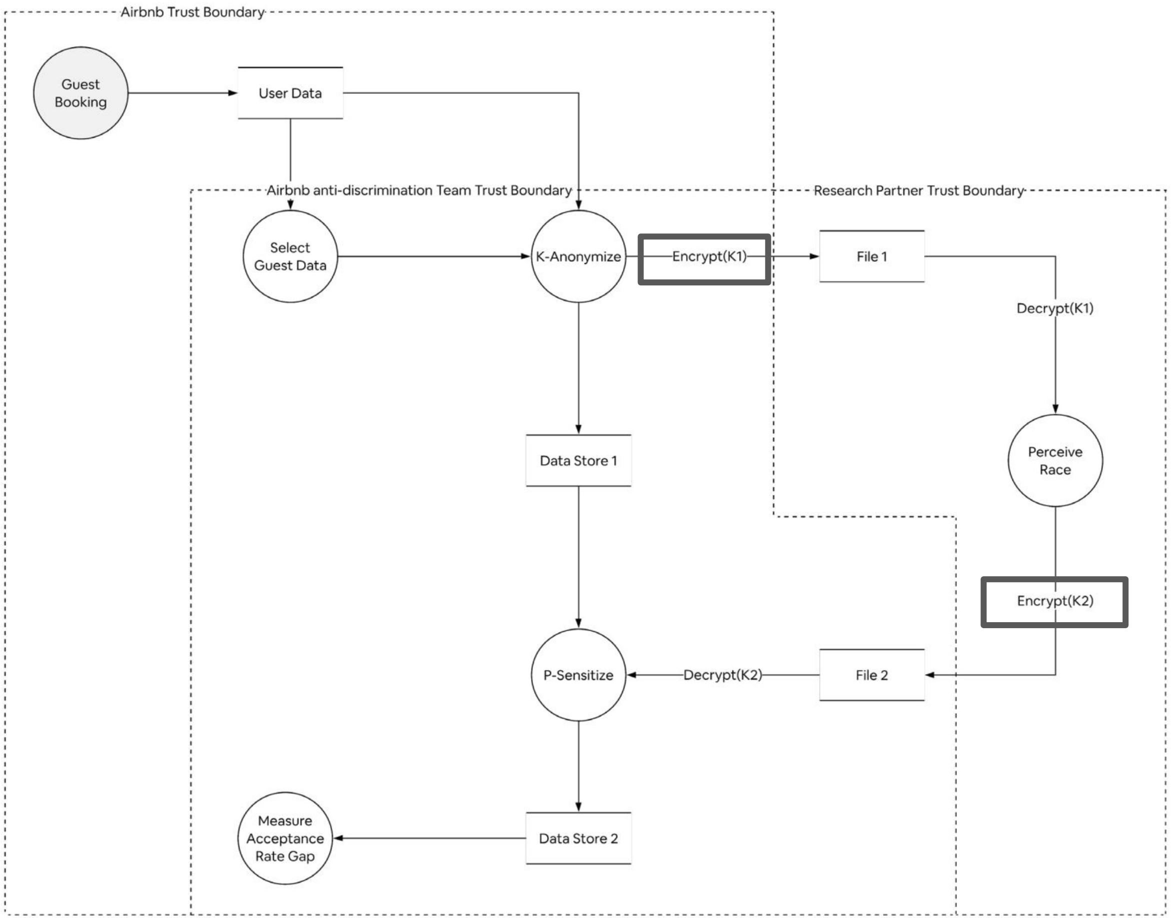


Figure 5: Simplified Data Flow Diagram (DFD)

Differential Privacy

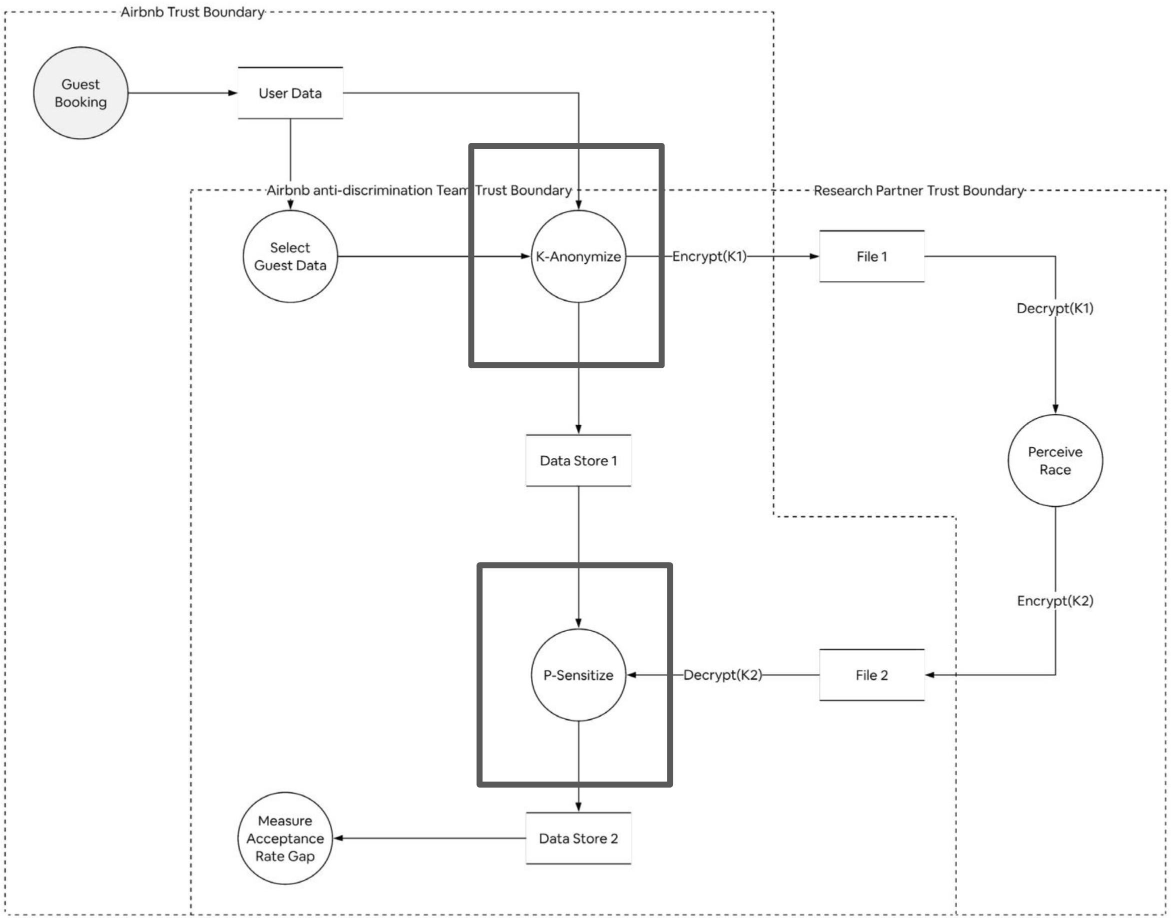


Figure 5: Simplified Data Flow Diagram (DFD)

What are these concepts?

If you had unmodified data points a bad actor could infer the membership and perceived attributes of Airbnb clients

First name: Michael **Profile photo:** **# of accepted bookings:** 6 **# of rejected bookings:** 2	**First name:** Stephen **Profile photo:** **# of accepted bookings:** 6 **# of rejected bookings:** 2	**First name:** Gerard **Profile photo:** **# of accepted bookings:** 2 **# of rejected bookings:** 2	**First name:** Nora **Profile photo:** **# of accepted bookings:** 4 **# of rejected bookings:** 2	**First name:** Suzanne **Profile photo:** **# of accepted bookings:** 4 **# of rejected bookings:** 2	**First name:** Aoife **Profile photo:** **# of accepted bookings:** 6 **# of rejected bookings:** 2
Perceived race: X	**Perceived race:** Y	**Perceived race:** X	**Perceived race:** X	**Perceived race:** X	**Perceived race:** Y

Is removing PII enough?

	# of accepted bookings	# of rejected bookings	Perceived race
1	6	2	X
2	6	2	Y
3	2	2	X
4	4	2	X
5	4	2	X
6	6	2	Y

No, because each row is unique!

Let’s K-anonymize

	# of accepted bookings	# of rejected bookings	Perceived race
1	6	2	X
2	6	2	Y
3	3.33	2	X
4	3.33	2	X
5	3.33	2	X
6	6	2	Y

Values are just averaged to make rows non-unique

K-anonymity means that there are at least ***k* instances of each unique pair** of (number_of_accepts, number_of_rejects) in our dataset. Specifically, our dataset is now **3-anonymous** (so k = 3) because we can confirm that each unique pair of accepts/rejects — (6, 2) and (3.33, 2) — appear at least **3** times in the dataset (in rows 1, 2, 6 and rows 3, 4, 5, respectively).

Northeastern University

Let’s P-Sensitize

	# of accepted bookings	# of rejected bookings	Perceived race
1	6	2	X
2	6	2	Y
3	3.33	2	<u>Y</u>
4	3.33	2	X
5	3.33	2	X
6	6	2	Y

Underlined value shows the flip (X changed to Y)

P-sensitive k-anonymity means that, in addition to satisfying k-anonymity, **each unique pair** of (number_of_accepts, number_of_rejects) has at least **p distinct perceived race values.**

Specifically, this dataset is 2-sensitive 3-anonymous because each unique pair of accepts/rejects has at least 3 rows (k = 3) and at least 2 distinct perceived race values (p = 2): (6, 2) is associated with 2 perceived race values (“X” and “Y”), and (3.33, 2) is associated 2 perceived race values (“X” and “Y”).

Potential Weakness: Accuracy

	# of accepted bookings	# of rejected bookings	Perceived race
1	6	2	X
2	6	2	Y
3	2	2	X
4	4	2	X
5	4	2	X
6	6	2	Y

Original

	# of accepted bookings	# of rejected bookings	Perceived race
1	6	2	X
2	6	2	Y
3	3.33	2	Y
4	3.33	2	X
5	3.33	2	X
6	6	2	Y

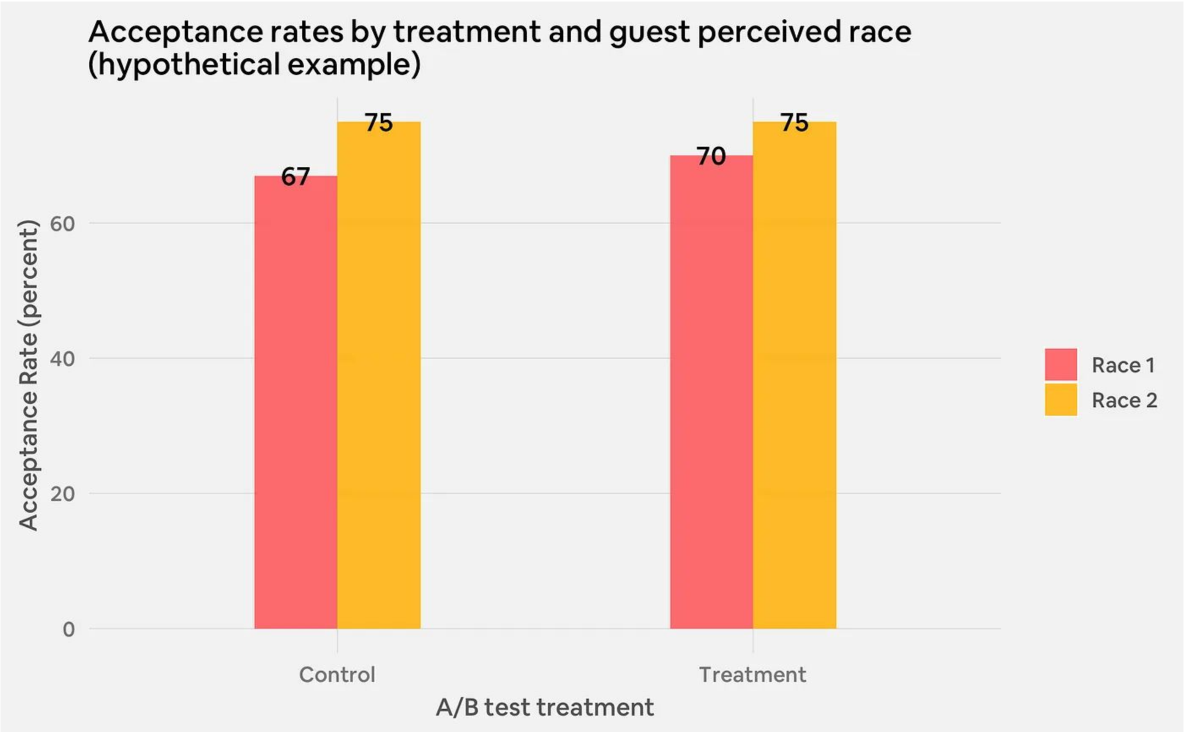
Anonymized

Our example demonstrates this risk: in the anonymized dataset, the acceptance rate for group X is 68% and the acceptance rate for group Y is 72%, as compared to acceptance rates of 67% and 75%, respectively, before anonymization occurred.

Authors look at this using simulations.

Northeastern University

A/B Testing to see if interventions worked



[More details in paper]

Thank You!

Readings for Next Class:

- There are two factions working to prevent AI dangers. Here's why they're deeply divided.