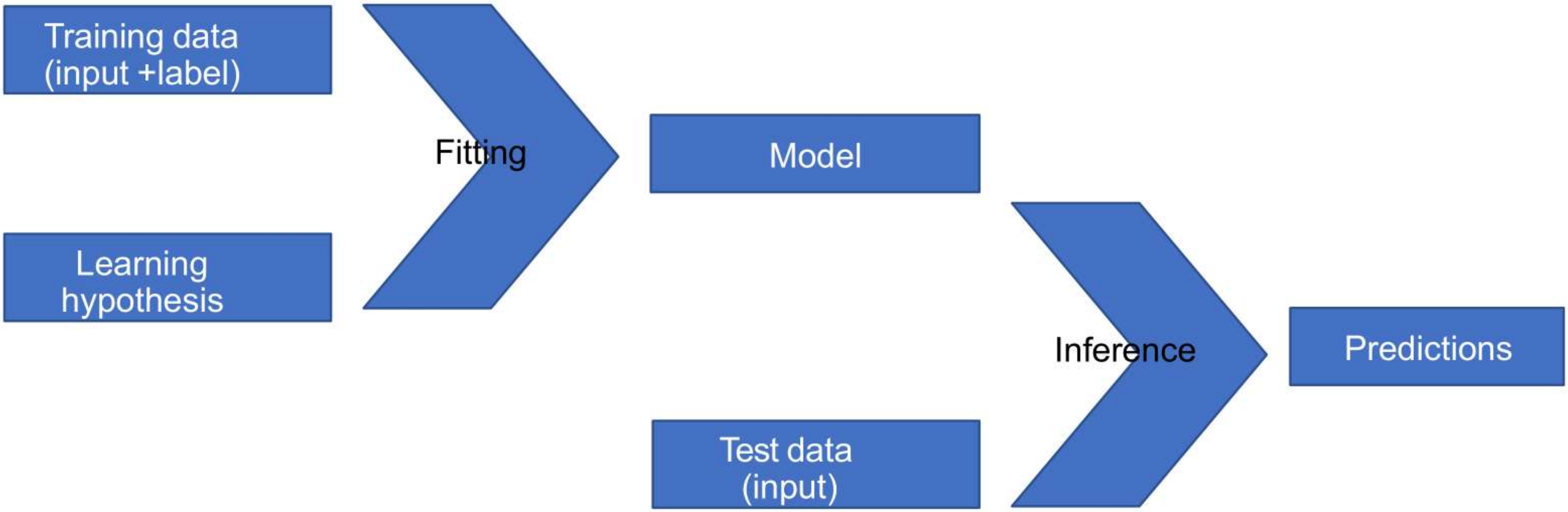
Topic 1: Privacy Attacks

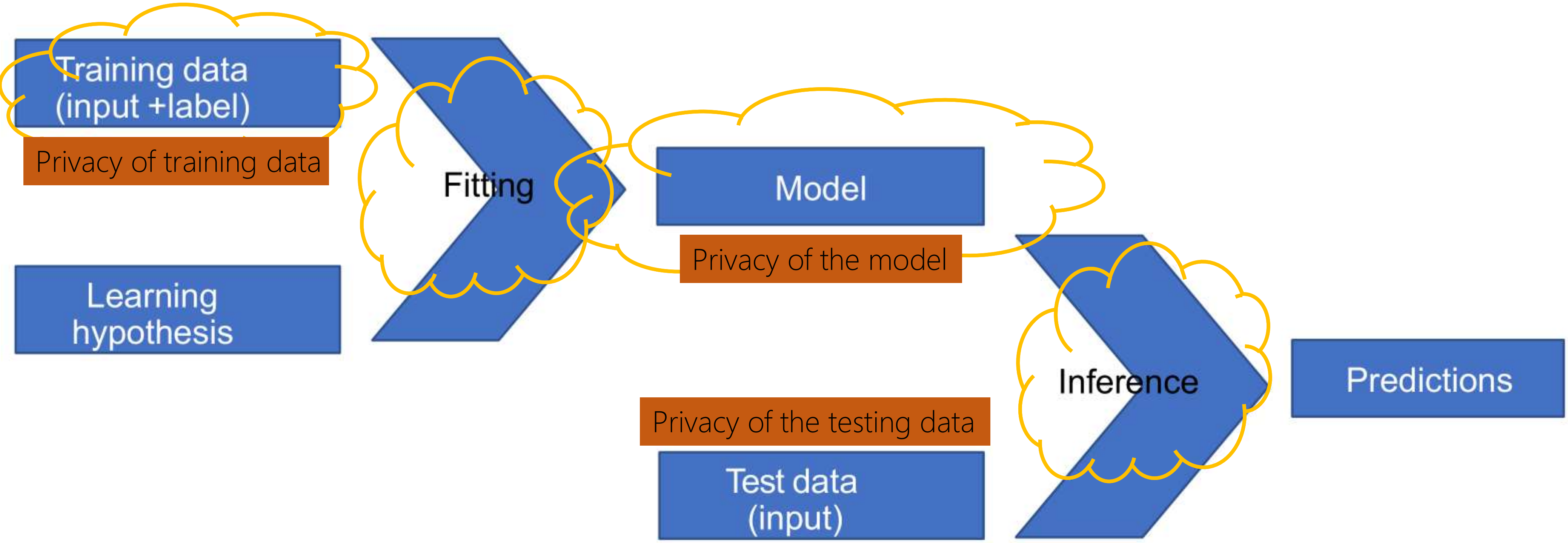
CS 6501, Data Privacy, Spring 2022

Tianhao Wang

Machine Learning Pipeline



Machine Learning as a Service (MLaaS)

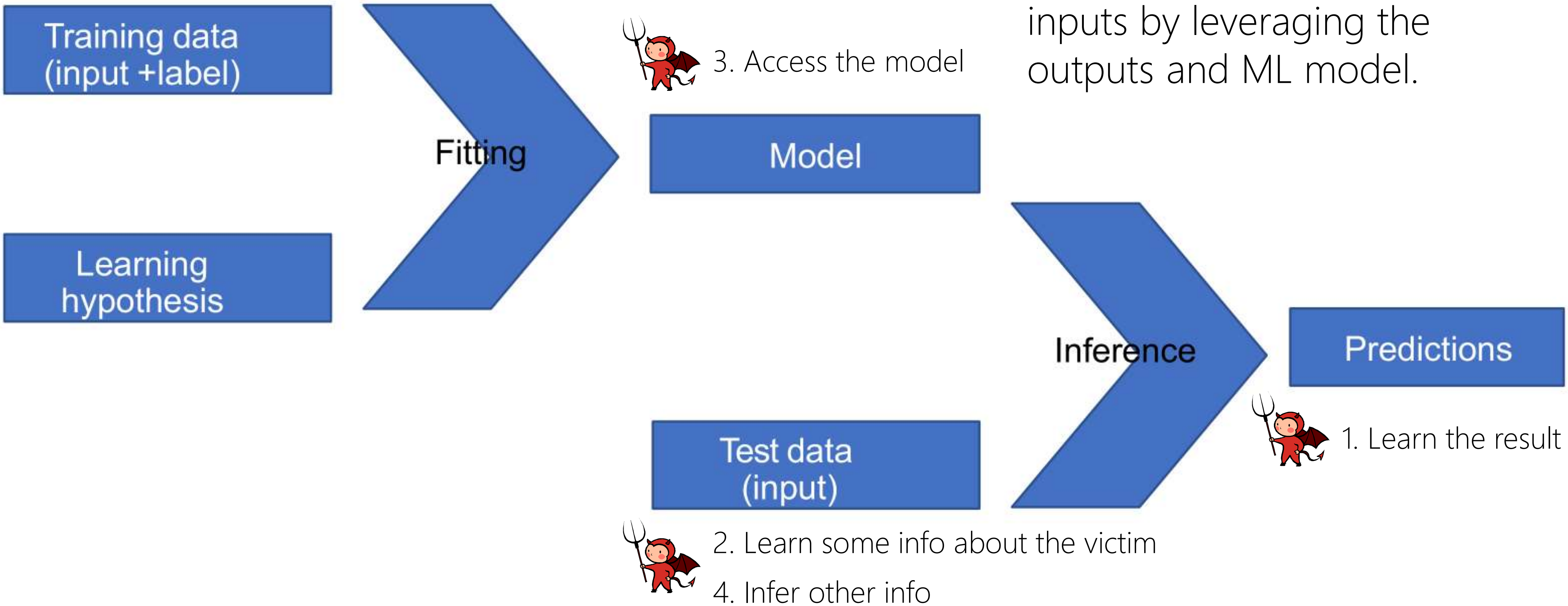


Outline

1. Model Inversion
2. Model Extraction
3. Membership Inference

Model Inversion

Extract private and sensitive inputs by leveraging the outputs and ML model.



Model Inversion

- Attacker Goal: Extract private and sensitive inputs by leveraging the outputs and ML model.

- Example: 538 Steak Survey on BigML.com
  - The model $f(\boxed{x_1, \ldots,} \boxed{x_n}) = \boxed{y}$

Household income
Whether person gambles
Whether cheated on significant other

Prediction of how person likes steak prepared:
- rare
- medium-rare
- medium
- medium-well
- well-done
Plus confidence value
Normalized vector of class confidences each in [0,1]

- How to do that?

Model Inversion Attack 1

- Evaluate f with x_n=0 and x_n=1
- Return x_n that gives y

The model $f(x_1, \ldots, x_n) = y$

Prediction of how person likes steak prepared:
- rare
- medium-rare
- medium
- medium-well
- well-done

Plus confidence value

Household income
Whether person gambles
Whether cheated on significant other

- Question 1: Is this a white-box (sees the model parameters) or black-box (only uses the model) attack?
- Question 2: Can x_n=0 and x_n=1 give the same y? How to deal with it?

[Fredrikson, Lantz, Jha, Lin, Page, Ristenpart 2014]

Generic model inversion

Given f , x_1 , ... , x_{n-1} , y infer x_n

x_n takes on possible values in set $\{v_1,\dots, v_s\}$

(1) Compute $y_j = f(x_1, \dots, x_{n-1}, v_j)$ for each j ← Runs in O(s)

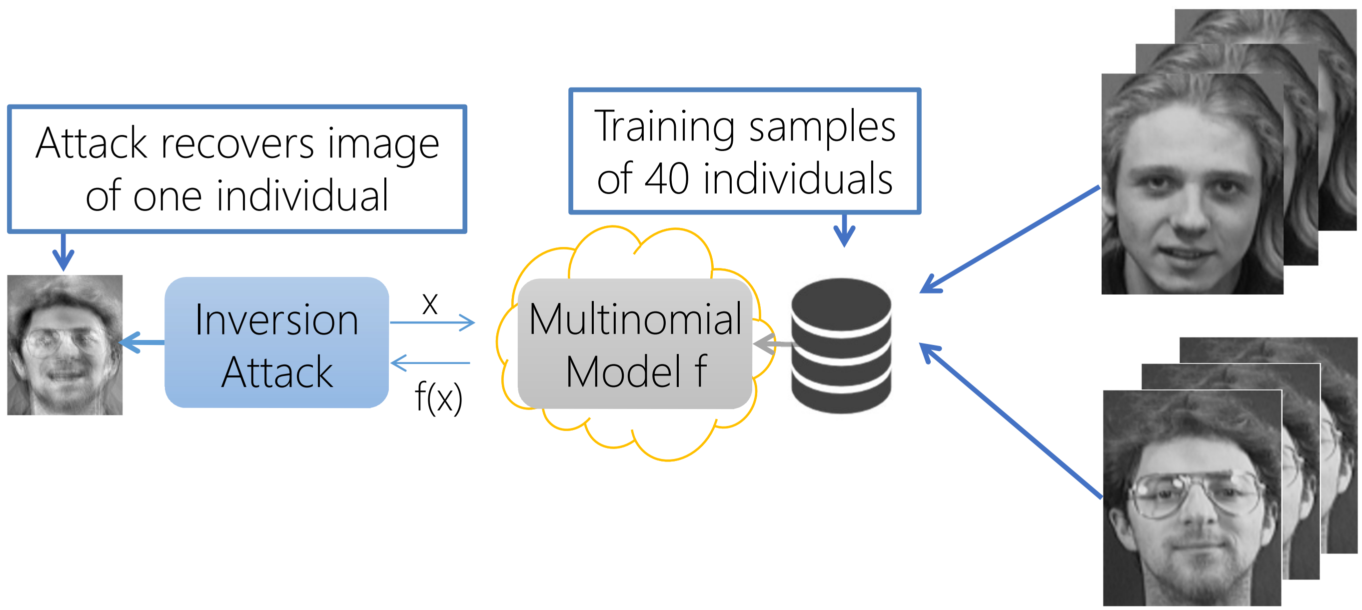
(2) Output v_j that maximizes

$$\mathrm{Dist}(y, y_j) \times \Pr(v_j \mid x_1, \dots, x_{n-1})$$

[Fredrikson, Lantz, Jha, Lin, Page, Ristenpart 2014]

Model Inversion Attack 2

- $f(x_1, \ldots, x_n) = [p_{Bob}, \ldots, p_{Jake}]$
- Given y, infer $x_1, \ldots, x_n$ assuming they are all unknowns



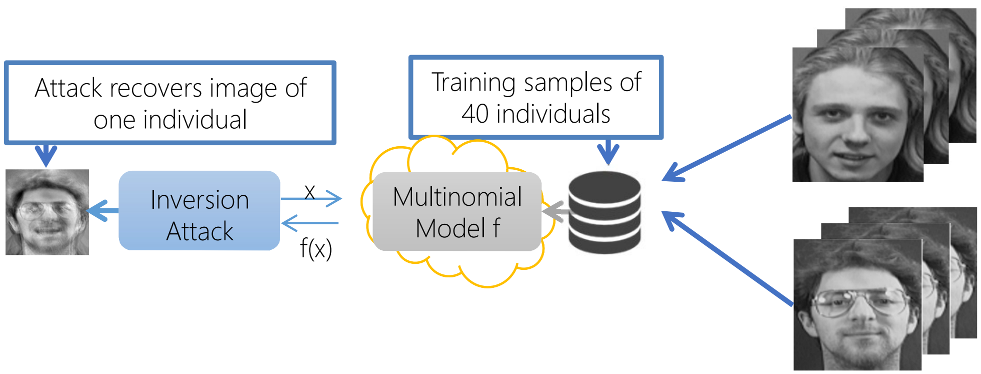
- Exponential possibilities. What can we do?

[Fredrikson, Jha, Ristenpart 2015]

Approach

- Setting: $f(x_1, \ldots, x_n) = [p_{Bob}, \ldots, p_{Jake}]$
- Problem: Given f, y = “Bob” find input x that is most likely to match “Bob”

Search for x that maximizes p_{Bob} using gradient descent



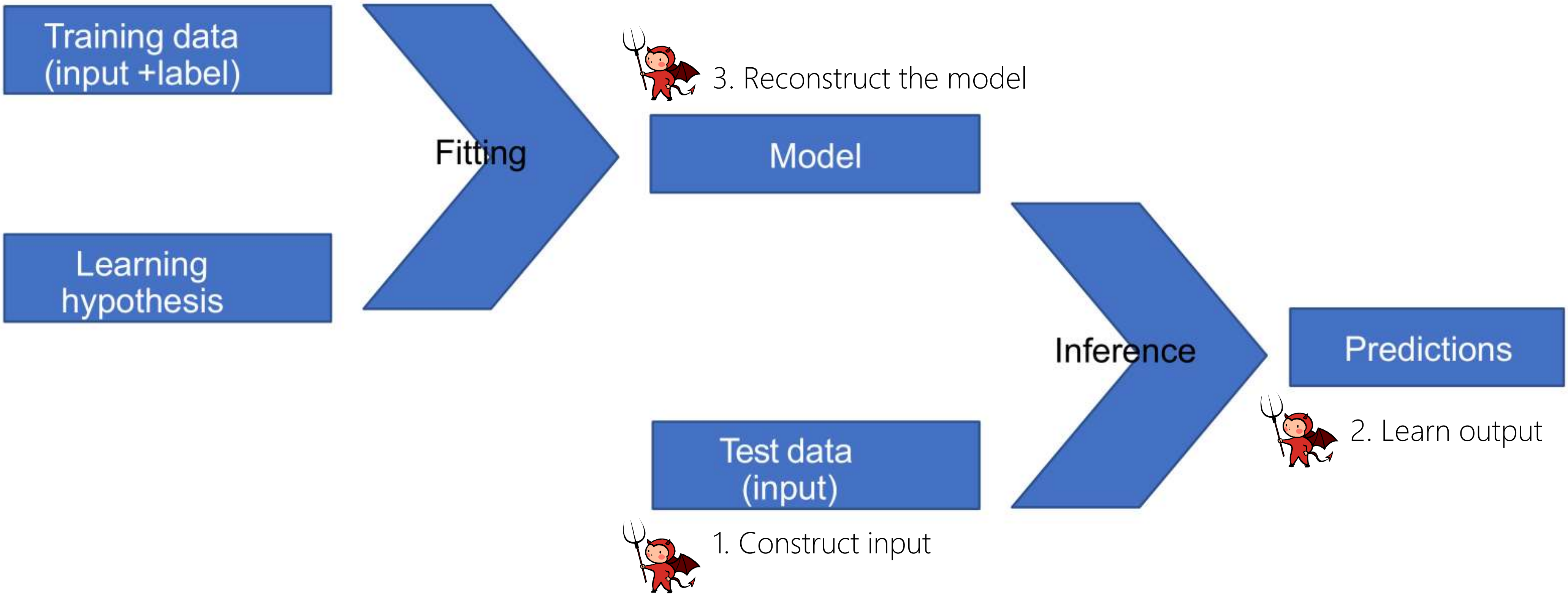
- Question: Is this black-box or white-box attack?

[Fredrikson, Jha, Ristenpart 2015]

Outline

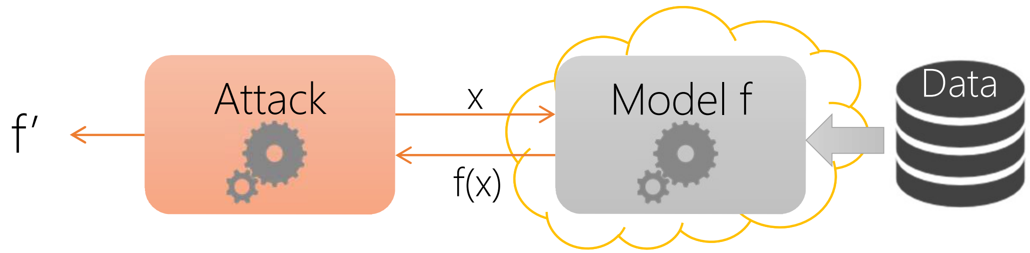
1. Model Inversion
2. Model Extraction
3. Membership Inference

Model Extraction



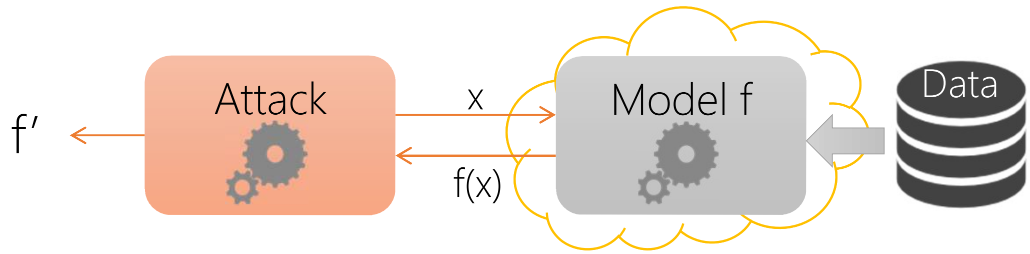
Model Extraction

- Goal: Adversary learns a close approximation of the model f using as few queries as possible



- Why: what is the implication?
  - Undermine pay-for-prediction pricing model
  - Facilitate privacy attacks (model inversion)

Extraction Example: Logistic Regression



n+1 parameters w, b
chosen using training set
to minimize expected error

$$f(x) = 1 / (1 + e^{-(w*x + b)})$$

How to do it?

$$\ln\left(\frac{f(x)}{1 - f(x)}\right) = w*x + b$$

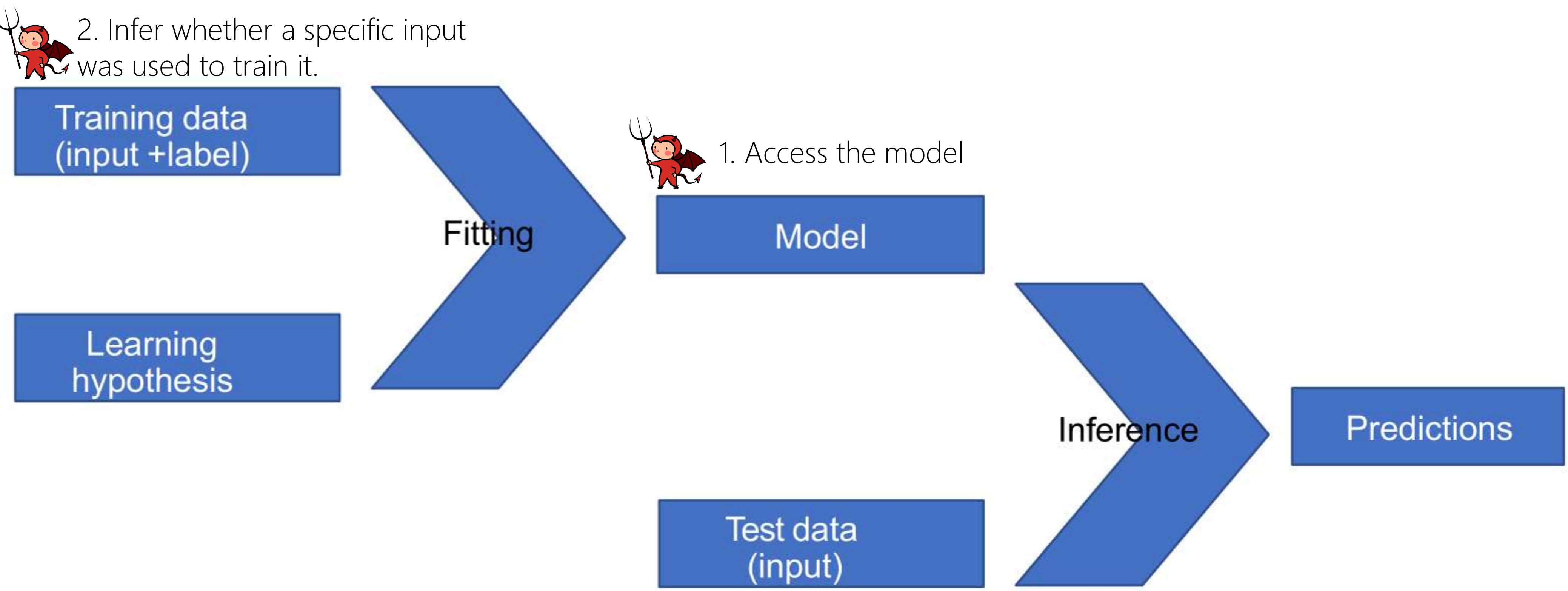
Linear equation in
n+1 unknowns w, b

Query n+1 random points ⇒ solve a linear system of n+1 equations

Outline

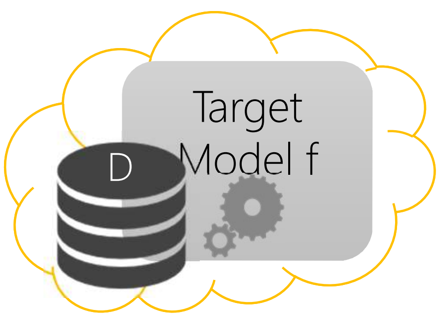
1. Model Inversion
2. Model Extraction
3. Membership Inference

Membership Inference

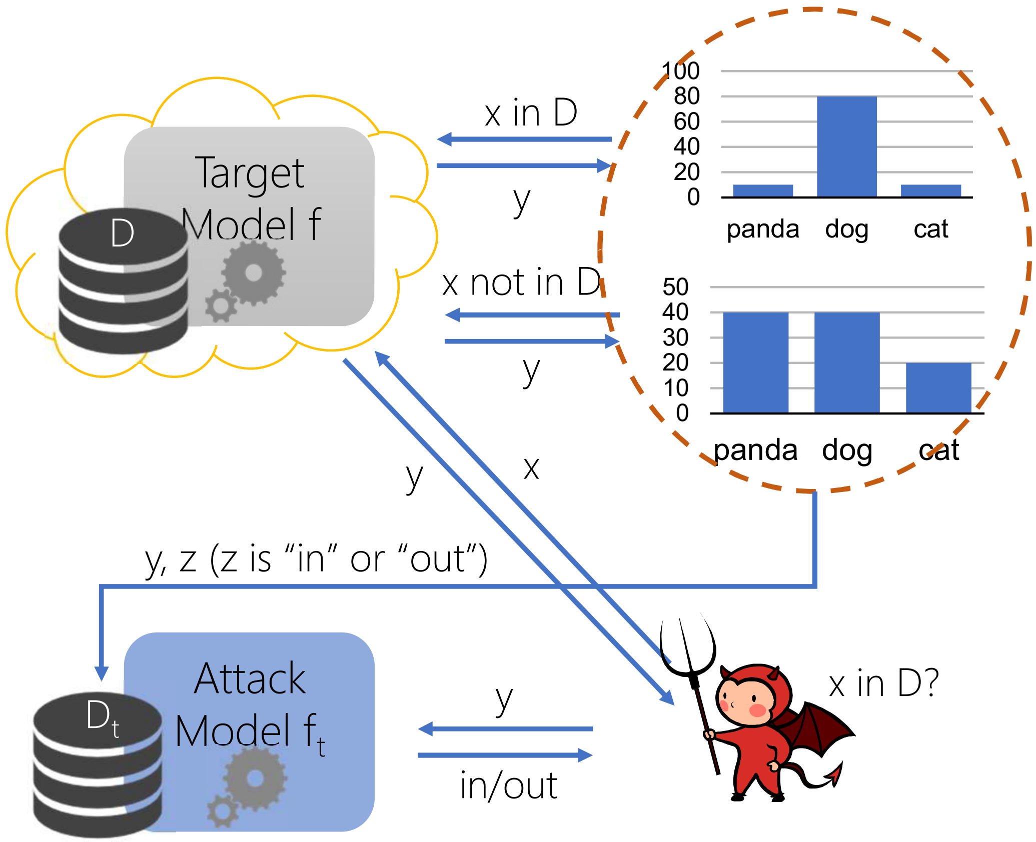


Membership Inference

- Goal: Infer whether x is used to train the model.
- Why: what is the privacy concern?
  - Assume f can predict cancer-related health outcomes.
  - If x is used to train f, x may have health issues.
- How?
  - By observing the behavior of f.



Membership Inference



Train a model to recognize the difference

1. Find D′, values that result in different y

Intuition: Models memorize too much information so that the behavior (e.g., confidence) are different.

2. Obtain D_t

3. Train f_t

4. Evaluate

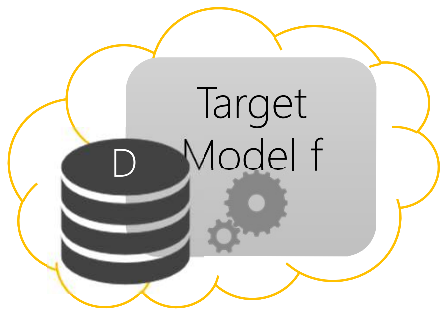
Without knowing the specifics of the actual model f!

[Shokri, Stronati, Song, Shmatikov 2017]

Adversarial Knowledge

1. The adversary does not have any specialized knowledge of the training data.
2. The adversary has access to population-level statistics that describe the distribution of features in the target model's training data.
3. The adversary has access to some versions of real data in the training data or some leaked portion but not the complete training set.

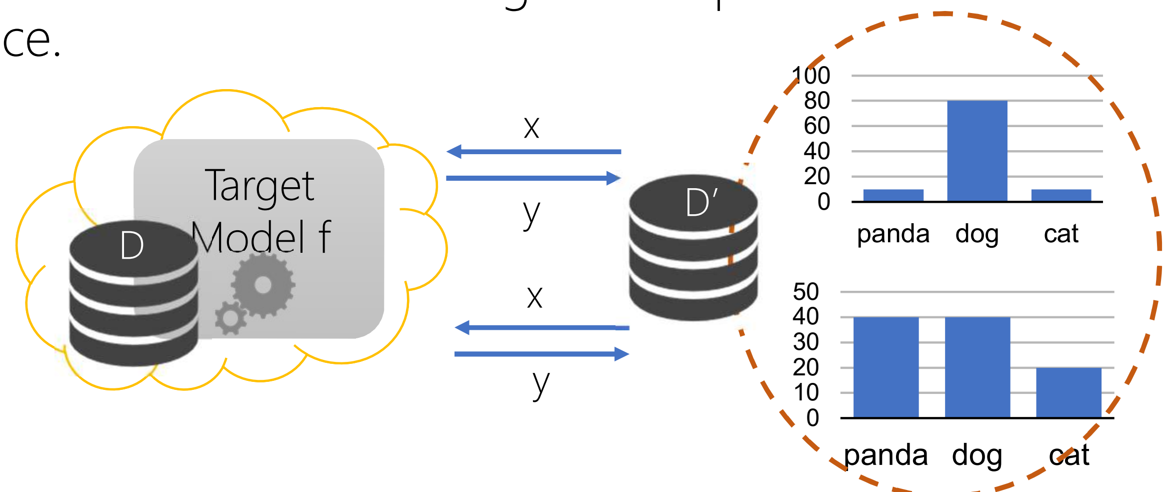
- Knowledge: 1<2<3
- Attack Difficulty: 3<2<1



[Truex, Liu, Gursoy, Yu, Wei 2019]

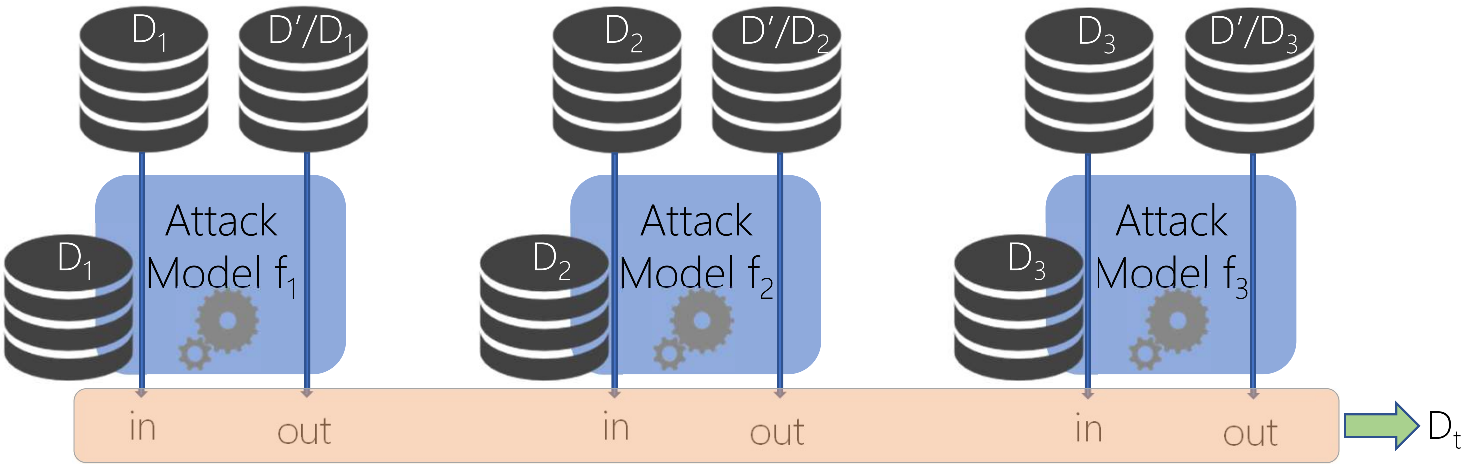
1. Development of a Shadow Dataset D′

- Goal: Generate D′ that is used to obtain D_t
- Statistic-Based Sampling: Given known distributions for features, an adversary may conduct random sampling to construct these new samples.
- Query-Based Generation: Generate a random sample x and then query the target model to obtain class y.
  - Want to identify instances for which the machine learning service provides a class label with relatively high confidence.
  - To save queries:
    - Region-Based Generation
    - Active Learning-Based Generation



2. Obtain D_t using Shadow Models

1. Partition D′ to $D_{1,}$ $D_{2, \ldots,}$ D_s where s>=1
2. For j in {1, …, s}, train f_j based on D_j
   - f_j can be close to f
3. For j in {1, …, s}, evaluate D_j on f_j to obtain <y, "in">
4. For j in {1, …, s}, evaluate D'/D_j on f_j to obtain <y, "out">



3. Generating the Membership Attack Model

- The dataset D_t will then be used to generate the final attack model f_t, which takes as input a probability vector output for an instance x and outputs a binary classification of “in” or “out”.