ML leakage metrics

How bad could it get?

Overview

- Setup
- Most important terms used
- Population attack - simple metric
- Reference attack - correct but slow metric
- Wrap Up

Overview

- **Setup**
- Most important terms used
- Population attack - simple metric
- Reference attack - correct but slow metric
- Wrap Up

Setup

- Groups of 2
    - Different partners
    - Choose name and write down
    - Requirements - both
    - Other exercises - pair programming

- First Exercises
    - For each exercise, three parts
    - One group presents each part over Zoom
    - Start with RED, then turn GREEN

Setup - cont

- Further exercises
  - Supplementary, only after all parts green
  - First 3 groups win a price

- NN exercises
  - If I really miscalculated your ML-foo
  - Uses the ml_privacy_meter
  - Things I would've liked to do but didn't have the time
  - You're on your own :(
  - Not sure we have the time to present it

Exercises

- Try as good as possible to make them reproducible
- Copy/paste code without shame - no DRY today
- Append to the notebook, add markdown cells

Installation

First Exercise

1. Clone github repo
2. Start up jupyter
3. Install dependencies
4. Run 1-ml_load_data

https://go.epfl.ch/ml-2023

Overview

- Setup
- **Most important terms used**
- Population attack - simple metric
- Reference attack - correct but slow metric
- Wrap Up

Most important terms used

- Logit and sigmoid
- Confusion Matrix
- Receiver Operating Characteristic (ROC)
  - Area under the curve (AUROC)
- Random forest classifier
- Differential Privacy (DP)

Logit and Sigmoid function

Logit

$x(p) = \log(p/1-p)$

$p = 0..1$

$x = -\inf..\inf$

Probability

Sigmoid

Logit

$p(x) = 1 / (1 + e^{-x})$

Confusion Matrix

		Predicted condition		
	Total population= P + N	**Positive (PP)**	**Negative (PN)**	[Confusion matrix - Wikipedia]
Actual condition	**Positive (P)**	**True positive (TP), hit**	**False negative (FN), type II error, miss, underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power TP/P = 1 − FNR
	Negative (N)	**False positive (FP), type I error, false alarm, overestimation**	**True negative (TN), correct rejection**	False positive rate (FPR), probability of false alarm, fall-out FP/N = 1 − TNR

Receiver Operating Characteristic (ROC)

Curve for

- Binary classifiers with a threshold
- Shows performance of model

Area under the curve (AUROC)

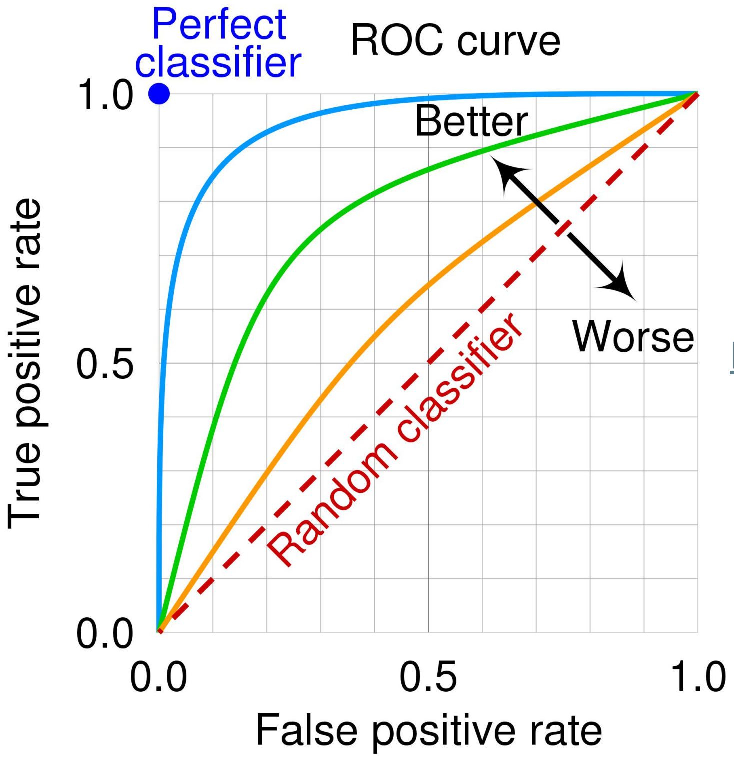
- Single-number measurement
- Indication of performance for comparisons
- Too much optimisation -> overfitting

ROC Drawing

Input

- Confidence of model for each case
  - Probability, between 0..1
  - 0 -> negative case
  - 1 -> positive case
- list of corresponding labels, e.g., $\{0,1\}^n$

ROC Types



[Receiver operating characteristic - Wikipedia]

Random Forest Classifier

Input

- labelled data

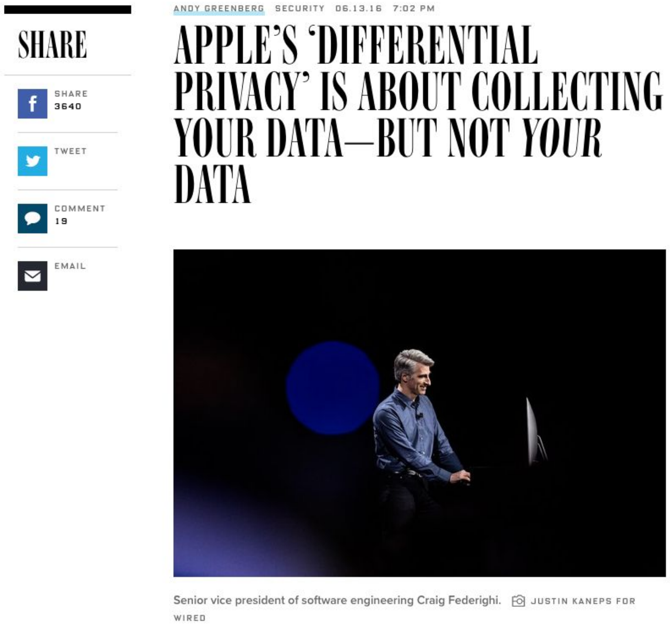
Model

- Many decision trees trained on the data

Output

- Voting of the trees on the given data
- ∑ of all outputs = 1

Differential Privacy - let’s add some noise

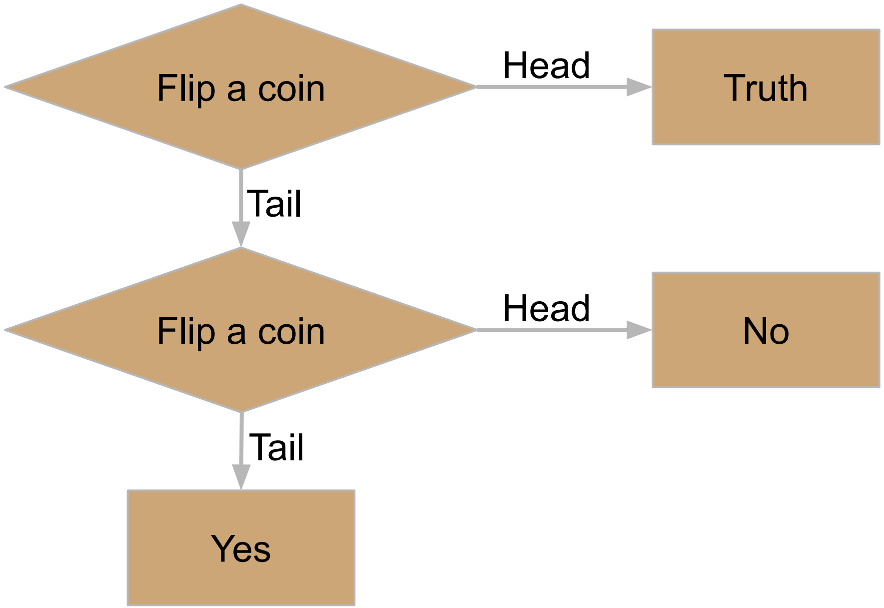

SHARE
SHARE 3640
TWEET
COMMENT 19
EMAIL

ANDY GREENBERG SECURITY 06.13.16 7:02 PM

APPLE’S ‘DIFFERENTIAL PRIVACY’ IS ABOUT COLLECTING YOUR DATA—BUT NOT *YOUR* DATA

Senior vice president of software engineering Craig Federighi. JUSTIN KANEPS FOR WIRED

Did you cheat in your tax reports?



Differential Privacy

Good

- Allows for plausible denial
  - My record seems to be part of it, but it's not
- Reduces leakage of privacy data

Bad

- Reduces accuracy of models

How

- Chose a privacy budget *epsilon*
  - high epsilon (>5) -> less privacy
  - low epsilon (0.1-2) -> more privacy
- In *RandomForest*, randomly swap branches

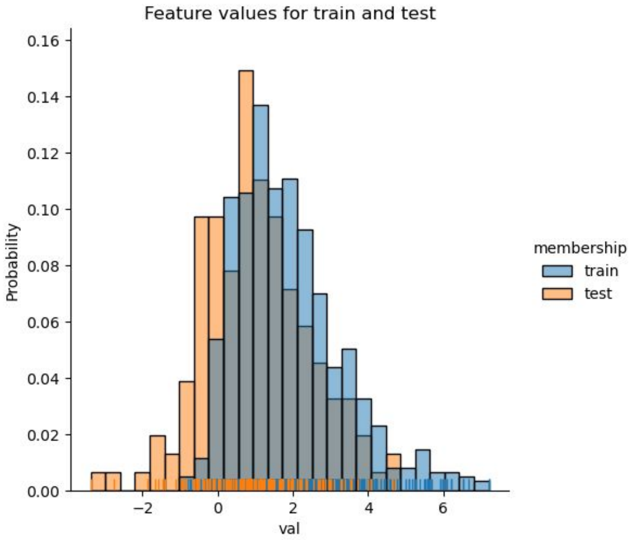
Overview

- Setup
- Most important terms used
- **Population attack - simple metric**
- Reference attack - correct but slow metric
- Wrap Up

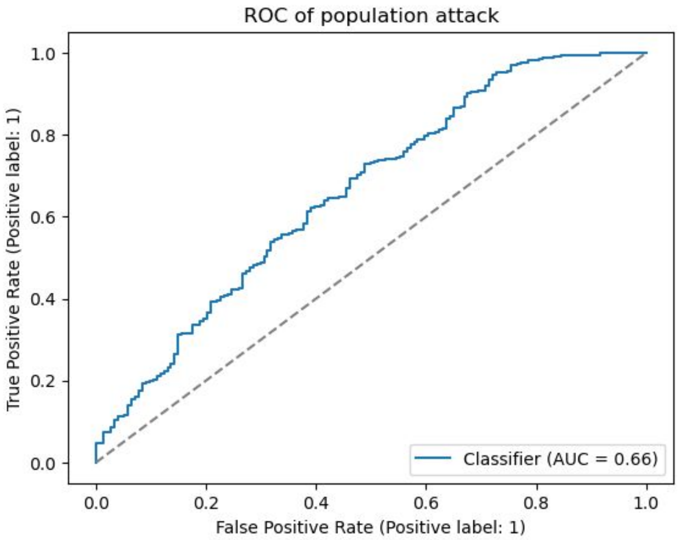
Idea

- Treat the training cases as parameters INSIDE of the model
  - The model has been trained using these
- Treat the test cases as parameters OUTSIDE of the model
  - The model never learnt these
- Compare the evaluation of these cases
  - INSIDE cases == 1
  - OUTSIDE cases == 0
  - Is there a difference between the two?

Measurements with potential leakage



Distribution of logit values
for train and test cases



ROC of the potential leakage
to population attacks

Overview

- Setup
- Most important terms used
- Population attack - simple metric
- **Reference attack - correct but slow metric**
- Wrap Up

Idea

- Start with a list of training examples whose leakage we want to measure
- Create different OUTSIDE models for each case
    - Measure mean and standard deviation for a case
- Measure the predicted confidence of the case in the INSIDE model
- Calculate the probability of the OUTSIDE result < the INSIDE result

The good and the bad

Good

- Actually measuring outcomes
- Using different values
- Allows for good comparisons

Bad

- Very computation intensive
- Only works for simple algorithms
- Difficult to do for 1000s of 1GB images in a CNN

Overview

- Setup
- Most important terms used
- Population attack - simple metric
- Reference attack - correct but slow metric
- **Wrap Up**

The Problem

- Training an ML model represents your data
- If the data is private / secret, it can be leaked
- Very bad leakage: retrieve chat text
- Less bad leakage: inference of IN and OUT cases

How to measure

Population attack measurement

- Simple and fast
- Allows to compare pipelines qualitatively

Reference attack measurement

- Slow, but more accurate
- Compare pipelines quantitatively

Protecting

Differential Privacy

- Add some noise to the training data
- Trade-off between leakage and accuracy
  - fairness/disparate impact
    - DP widens disparities in performance between population groups
  - arbitrariness of decisions
    - decisions of some inputs depend fully on the randomness in training

Links

Code

- For simple models
  - DiffPrivLib
- For neural networks
  - ml_privacy_meter - measurements for neural networks
  - Opacus

Papers

- Membership Inference Attacks From First Principles