Cleanlab

Data Council - Lightning Talk - Mar 30, 2023

Automatically Find and Fix Data & Label Errors in ML Datasets

Curtis Northcutt | CEO & Co-Founder, Cleanlab

I spent the majority of the last decade at MIT solving this problem.

In this talk, I share 6 key lessons learned along the way.

Lesson 1

The best model is only as good as the data it learns from.

For learning, data is as important as the model

In machine learning, we tend to focus on the model

RE-

When algorithms are trained with erroneous data

Source: MIT Technology Review (May 28, 2019)

Deep neural networks easily fit random labels.

- *Zhang et al. (ICLR, 2017)*

Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small difference between training and test performance. Conventional wisdom attributes small generalization error either to properties of the model family, or to the regularization techniques used during training.

Through extensive systematic experiments, we show how these traditional approaches fail to explain why large neural networks generalize well in practice. Specifically, our experiments establish that state-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data. This phenomenon is qualitatively unaffected by explicit regularization, and occurs even if we replace the true images by completely unstructured random noise. We corroborate these experimental findings with a theoretical construction showing that simple depth two neural networks already have perfect finite sample expressivity as soon as the number of parameters exceeds the number of data points as it usually does in practice.

Lesson 2

Data and label issues plague the most-used AI tech.

e.g. Dall-E, ChatGPT, …

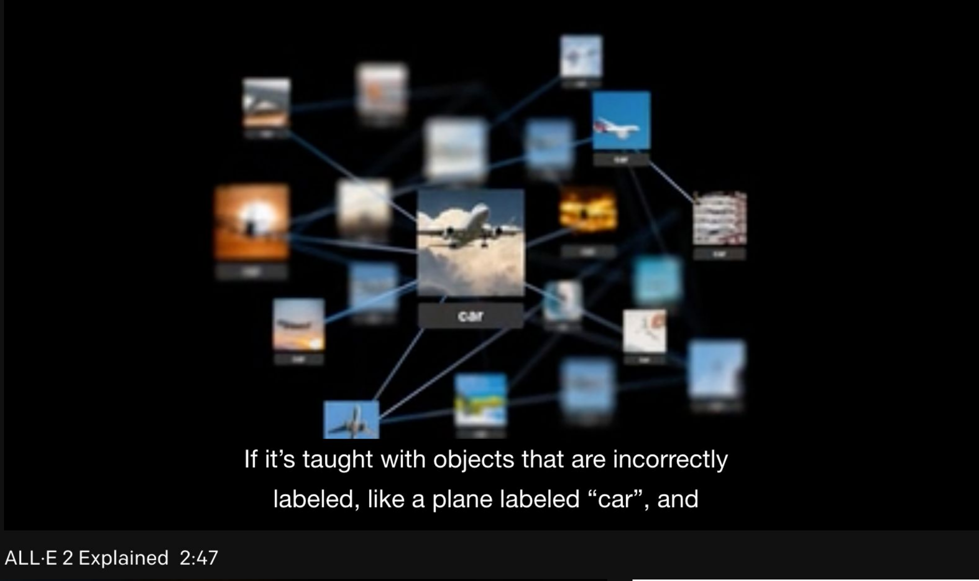
Why the hype around Data-centric AI?

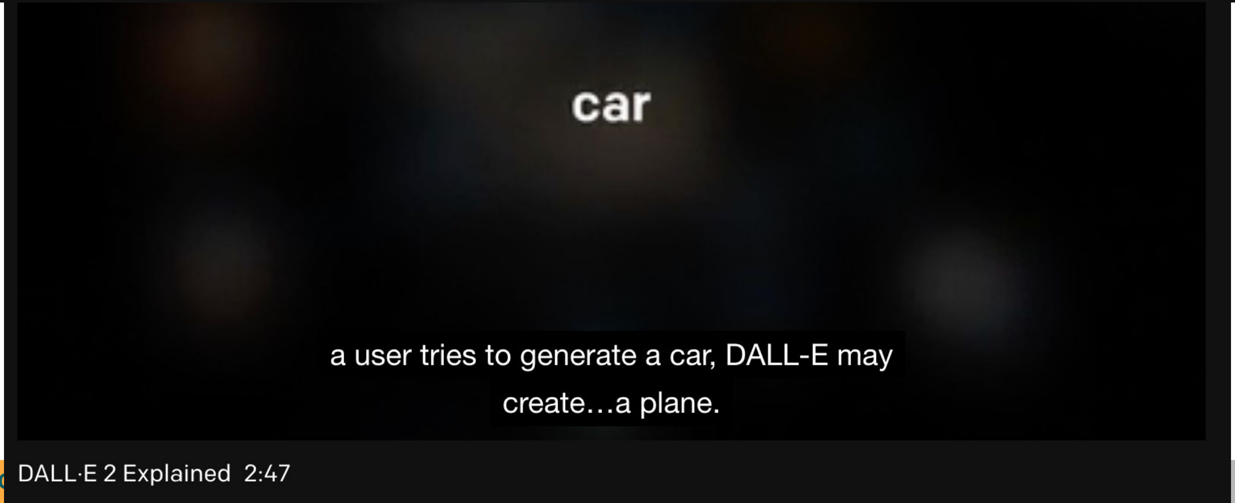
OpenAI has ‘open’ly stated that one of the biggest issues with Dall-E and GPT-3 is errors in the data and labels used during training.

It’s not the model, it’s the data!

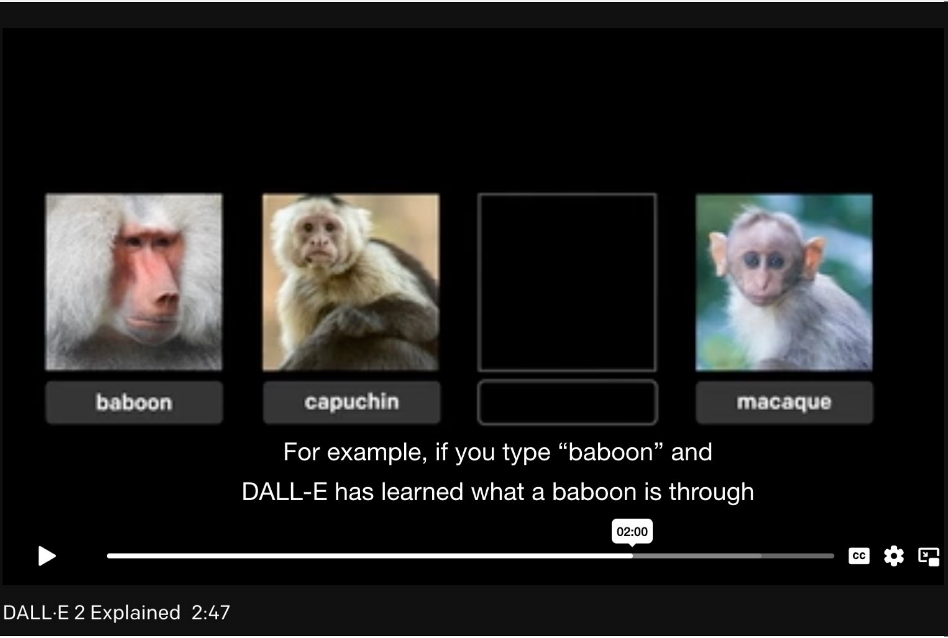
Let’s take a look at the Dall-E demo page: https://openai.com/dall-e-2/#demos


The technology is constantly evolving, and
DALL-E 2 has limitations.
01:43
DALL·E 2 Explained 2:47


car
If it’s taught with objects that are incorrectly
labeled, like a plane labeled “car”, and
ALL·E 2 Explained 2:47


car
a user tries to generate a car, DALL-E may
create…a plane.
DALL·E 2 Explained 2:47

Dall-E’s big issue → label errors at training time

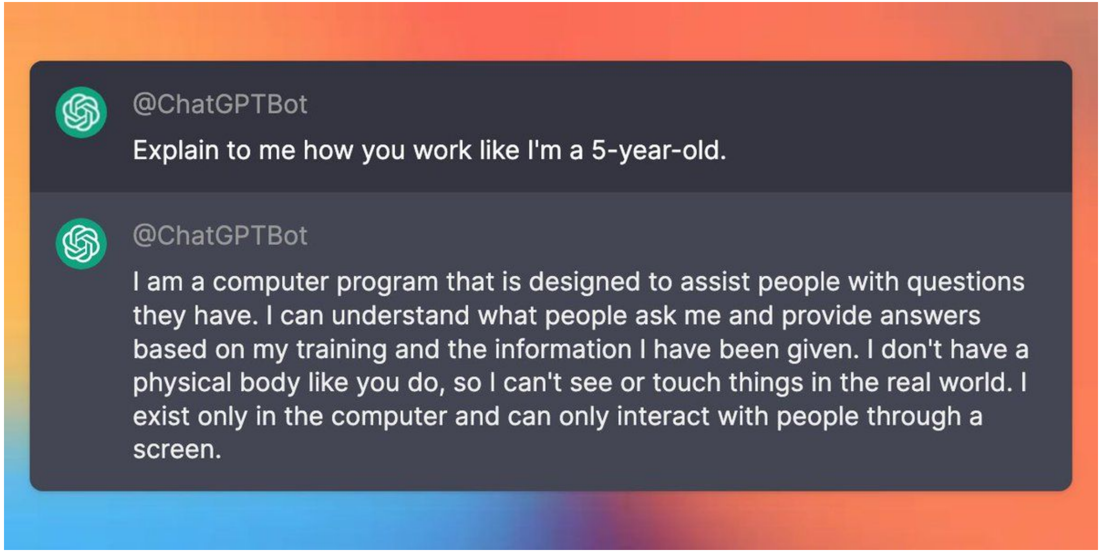


Takeaway: Reliability of ML models deployed in the real-world depends on quality of training data.

ChatGPT improved GPT-3 by improving data quality

ChatGPT was fine-tuned to:

- minimize harmful, untruthful, or biased output
- Used human rankings of potential outputs to put lower-weighting on ‘bad data’



Link to source. Link to blog.

Lesson 3

Every major tech company struggles with data quality in their ML pipelines.

Northcutt, Athalye, & Mueller (2021) Pervasive label errors, nominated for best paper at NeurIPS

Why the hype around Data-centric AI?

Why it's time for 'data-centric artificial intelligence'

by Sara Brown | Jun 7, 2022

Source: link

Forbes

Ng observes that 80% of the AI developer's time is spent on data preparation. This has been a widely shared estimate since the rise of "big data" in the late 2000s and the concomitant rise of "data scientists," known

Harvard Business Review

Analytics And Data Science | Bad Data Costs the ...

Analytics And Data Science

Bad Data Costs the U.S. $3 Trillion Per Year

by Thomas C. Redman

Source: link

September 22, 2016

Bad Data: The $3 Trillion-Per-Year Problem That's Actually Solvable

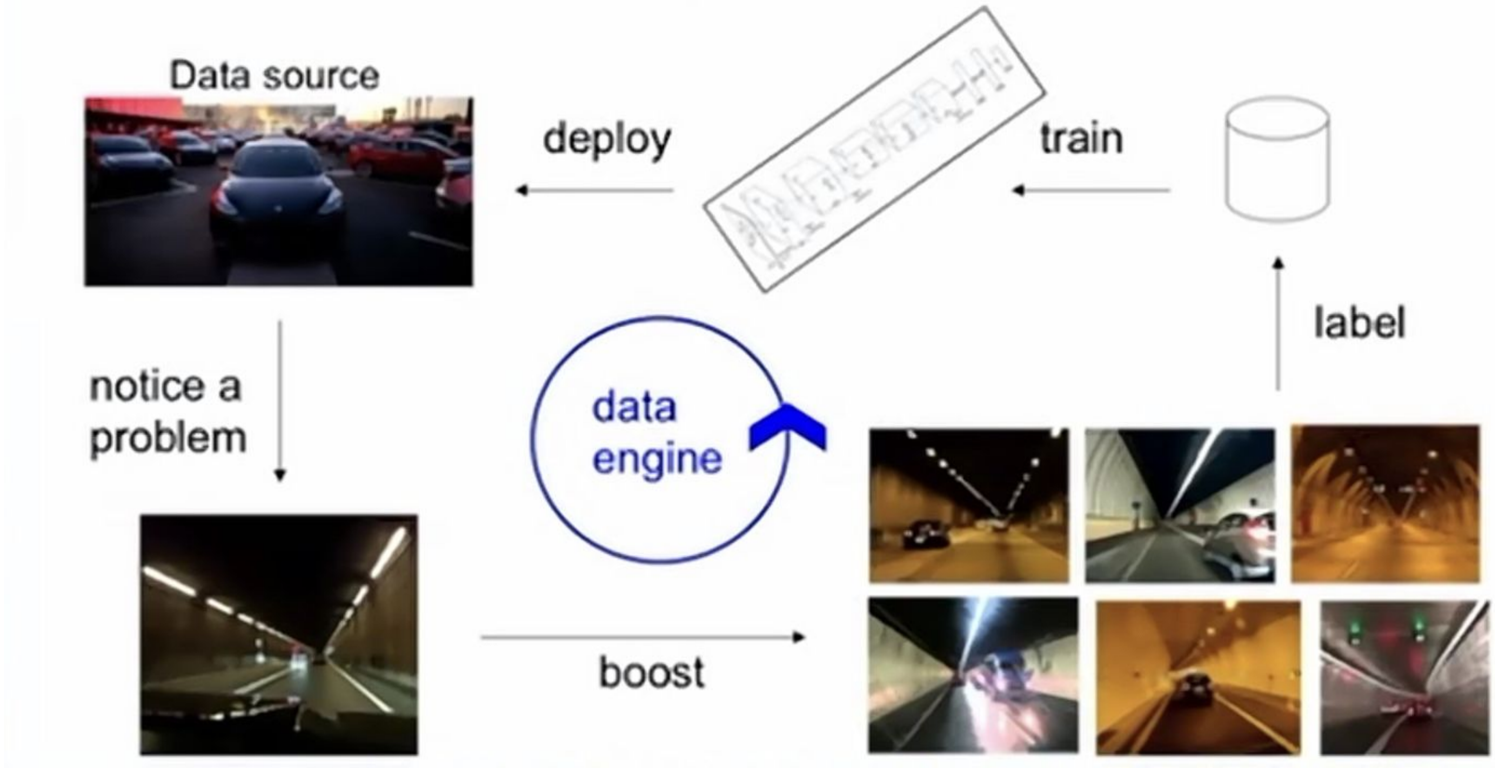
How the right tech can help entrepreneurs make data more accessible and accurate, avoiding massive losses in the process.

By Joy Youell

November 11, 2021

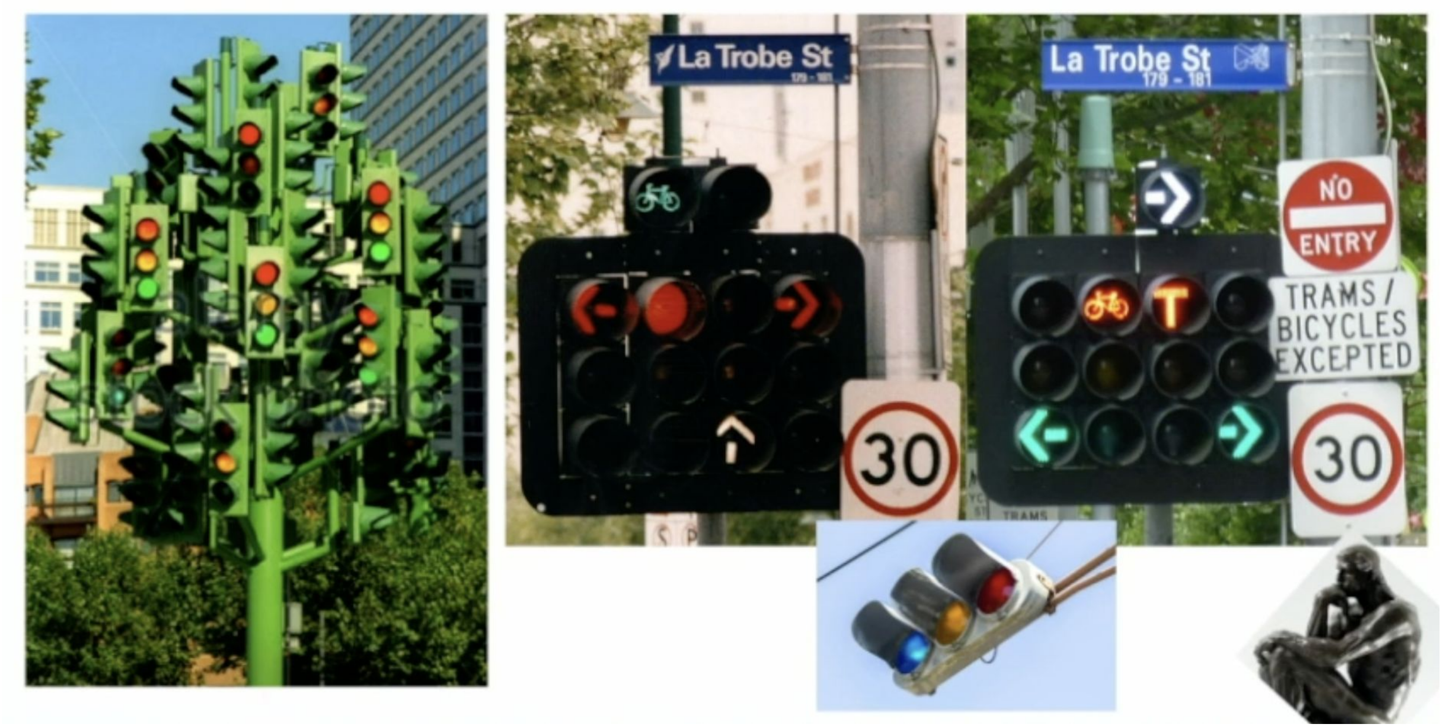
Source: link

Tesla Data Engine: use model outputs to improve training dataset



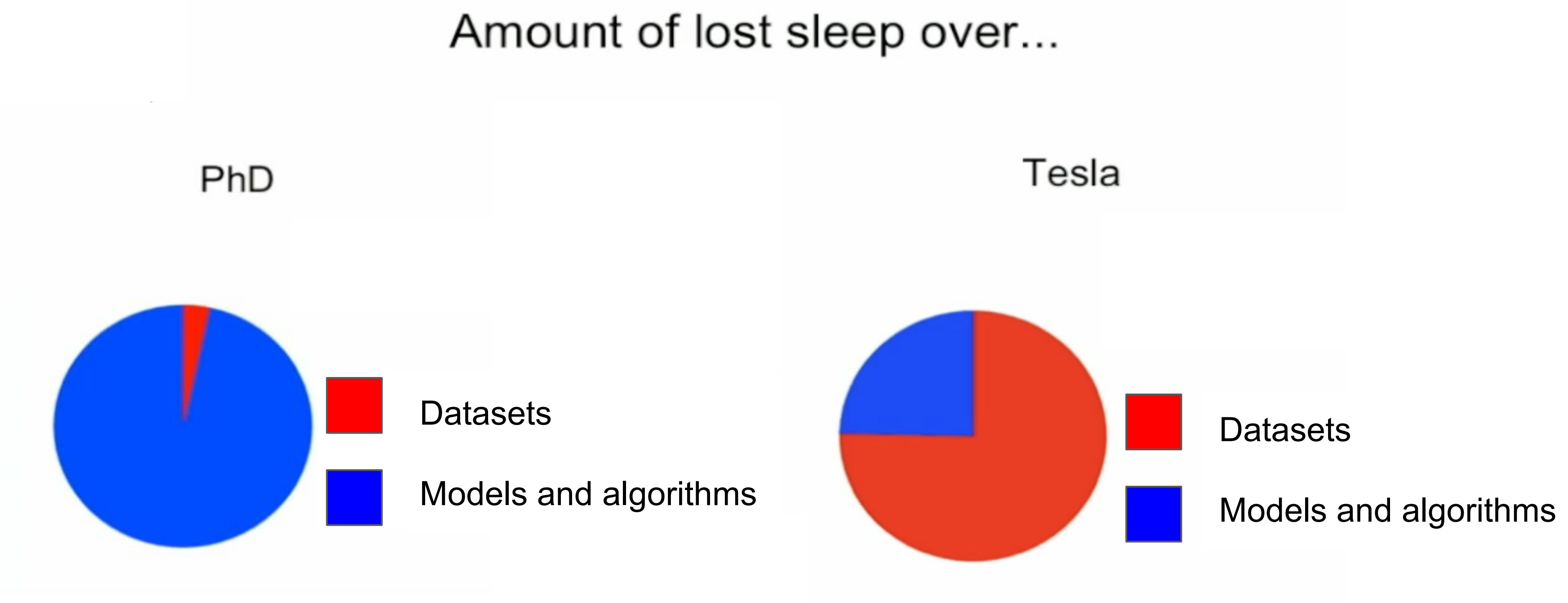
Slide from Andrej Karpathy, Tesla Director of AI (2021)

Tesla Data Engine: use model outputs to improve training dataset



Slide from Andrej Karpathy, Tesla Director of AI (2021)

Tesla Data Engine: use model outputs to improve training dataset



Slide from Andrej Karpathy, Tesla Director of AI (2021)

For more examples from Microsoft, Facebook, Amazon, Oculus, and Google, see https://cleanlab.ai/blog/cleanlab-history/

Lesson 4

The top 10 most-used ML test sets all have label issues.

Northcutt, Athalye, & Mueller (2021) Pervasive label errors, nominated for best paper at NeurIPS

MNIST

given: 8
corrected: 9

CIFAR-10

given: cat
corrected: frog

CIFAR-100

given: lobster
corrected: crab

Caltech-256

given: dolphin
corrected: kayak

ImageNet

given: white stork
corrected: black stork

QuickDraw

given: tiger
corrected: eye

3.4% of labels in popular ML test sets are erroneous

https://labelerrors.com/

	Dataset	Test Set Errors				
		CL guessed	MTurk checked	validated	estimated	% error
Images →	MNIST	100	100 (100%)	15	-	0.15
	CIFAR-10	275	275 (100%)	54	-	0.54
	CIFAR-100	2235	2235 (100%)	585	-	5.85
	Caltech-256	4,643	400 (8.6%)	65	754	2.46
	ImageNet*	5,440	5,440 (100%)	2,916	-	5.83
	QuickDraw	6,825,383	2,500 (0.04%)	1870	5,105,386	10.12
Text →	20news	93	93 (100%)	82	-	1.11
	IMDB	1,310	1,310 (100%)	725	-	2.9
	Amazon	533,249	1,000 (0.2%)	732	390,338	3.9
Audio →	AudioSet	307	307 (100%)	275	-	1.35

Northcutt, Athalye, & Mueller (2021) Pervasive label errors, nominated for best paper at NeurIPS

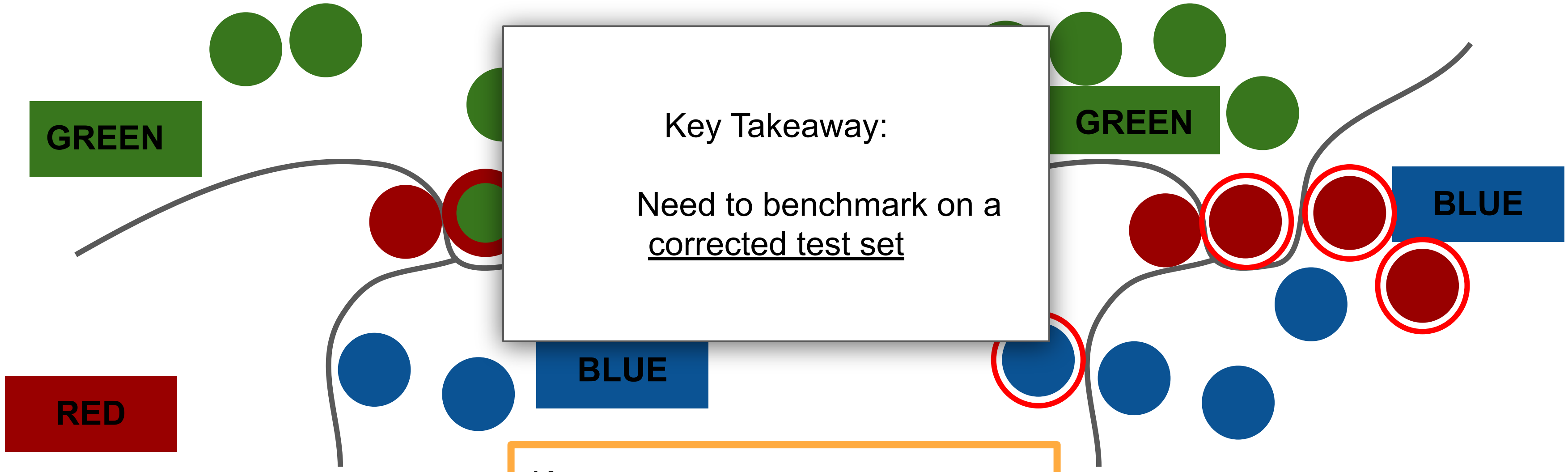
Lesson 5

Your test set benchmarks might not reflect real-world performance.

For imperfect test data: test acc ≠ real-world acc

Trained Model with 100% test accuracy.

Real-world accuracy ~ 67%



Key Takeaway:

Need to benchmark on a <u>corrected test set</u>

Key

Observed noisy label

Actual (not given) label

Northcutt, Athalye, & Mueller (Neurips 2021)
[paper link]

Lesson 6

All data and label issues in these slides were found automatically using Cleanlab.

(works for most ML datasets and ML models)

Cleanlab is built on confident learning theory and algorithms

- Confident learning (Northcutt, Jiang & Chuang. 2021. Journal of AI Research)
  - Theoretically grounded: proves realistic sufficient conditions for **exactly** finding label errors
  - Inspired by quantum computation and information theory.
  - General: Works with any model, dataset, and modality by using predicted probabilities as input (irrespective of which model produced them -- model-agnostic)

CL finds 'systematic errors'.
(not just random label flipping)

Reasoning: a **fox** is much more likely to be labeled **dog** than **cow**.

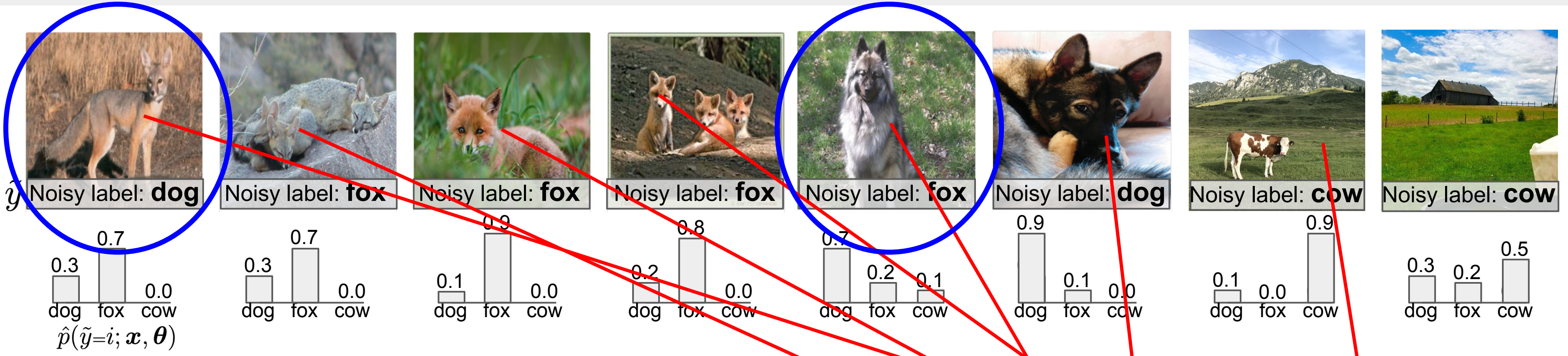
$$p(\tilde{y}|y^*; \boldsymbol{x}) = p(\tilde{y}|y^*)$$

$\tilde{y}$ - observed, noisy label

y^* - unobserved, latent, correct label

$\hat{p}(\tilde{y}, y^*)$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	0.25	0.1	0.05
$\tilde{y}=fox$	0.14	0.15	0
$\tilde{y}=cow$	0.08	0.03	0.2

How does confident learning work? (8 years in 10 seconds)



$$\frac{t_j}{}$$

$$t_{\text{dog}} = 0.7$$

$$t_{\text{fox}} = 0.7$$

$$t_{\text{cow}} = 0.9$$

$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j\}$$

$\boldsymbol{C}_{\tilde{y},y^*}$	$y^*{=}dog$	$y^*{=}fox$	$y^*{=}cow$
$\tilde{y}{=}dog$	1	1	0
$\tilde{y}{=}fox$	[illegible]	3	0
$\tilde{y}{=}cow$	0	0	1

$$C_{\tilde{y},y^*}[i][j] = |\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$$

Off diagonals are CL-guessed label errors

Intuition why this works: Robustness to miscalibration

$$\boldsymbol{C}_{\tilde{y}=i,y^*=j} := \boxed{|\{\boldsymbol{x} : \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i},\ \hat{p}(\tilde{y}=j|\boldsymbol{x}) \geq t_j\}|}$$

Exactly finds label errors for "ideal" probabilities (Ch. 2, Thm 1, in thesis)

$$t_j = \frac{1}{|X_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \boxed{\hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta})}$$

But neural networks have been shown (Guo et al., 2017) to be over-confident for some classes:

$$t_j^{\epsilon_j} = \frac{1}{|X_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \boxed{\hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) + \epsilon_j}$$

$$= \boxed{t_j + \epsilon_j}$$

What happens to $\boldsymbol{C}_{\tilde{y}=i,y^*=j}$?

$$\boldsymbol{C}^{\epsilon_j}_{\tilde{y}=i,y^*=j} = |\{\boldsymbol{x} : \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i},\ \hat{p}(\tilde{y}=j|\boldsymbol{x}) + \cancel{\epsilon_j} \geq \boxed{t_j + \cancel{\epsilon_j}}\}|$$

exactly finds errors

Compare Accuracy: Learning with 40% label noise in CIFAR-10

Fraction of zeros in the off-diagonals of $p(\tilde{y}|y^*)$

	0	0.6 ← More realistic (e.g. ImageNet)
Baseline (remove prediction != label)	83.9	84.2
Confident learning methods	84.8	86.2
	86.7	86.9
	87.1	**87.2**
	87.1	**87.2**
INCV (Chen et al., 2019)	84.4	73.6
Mixup (Zhang et al., 2018)	76.1	59.8
SCE-loss (Wang et al., 2019)	76.3	58.3
MentorNet (Jiang et al., 2018)	64.4	61.5
Co-Teaching (Han et al., 2018)	62.9	58.1
S-Model (Goldberger et al., 2017)	58.6	57.5
Reed (Reed et al., 2015)	60.5	58.6
Baseline	60.2	57.3

Same perf

Perf drop-off

<u>Data-centric</u>
Train with errors removed

“Change the dataset”

<u>Model-centric</u>
Train with errors

“adjust the loss”

Find label errors in your own dataset (1 import + 1 line of code)

```python
from cleanlab.classification import CleanLearning
from cleanlab.filter import find_label_issues

# Option 1 - works with sklearn-compatible models - just input the data and labels ツ
cl = CleanLearning(clf=sklearn_compatible_model)
label_issues_info = cl.find_label_issues(data, labels)

# Option 2 - works with ANY ML model - just input the model's predicted probabilities
ordered_label_issues = find_label_issues(
    labels=labels,
    pred_probs=pred_probs,  # out-of-sample predicted probabilities from any model
    return_indices_ranked_by='self_confidence',
)
```

https://github.com/cleanlab/cleanlab

Find data errors in your own dataset (1 import + 1 line of code)

```python
from cleanlab.outlier import OutOfDistribution

ood = OutOfDistribution()

# To get outlier scores for train_data using feature matrix train_feature_embeddings
ood_train_feature_scores = ood.fit_score(features=train_feature_embeddings)

# To get outlier scores for additional test_data using feature matrix test_feature_embeddings
ood_test_feature_scores = ood.score(features=test_feature_embeddings)

# To get outlier scores for train_data using predicted class probabilities (from a trained
classifier) and given class labels
ood_train_predictions_scores = ood.fit_score(pred_probs=train_pred_probs, labels=labels)

# To get outlier scores for additional test_data using predicted class probabilities
ood_test_predictions_scores = ood.score(pred_probs=test_pred_probs)
```

https://github.com/cleanlab/cleanlab

Find consensus labels for your dataset (1 import + 1 line of code)

```
from cleanlab.multiannotator import get_label_quality_multiannotator

get_label_quality_multiannotator(multiannotator_labels, pred_probs)
```

https://github.com/cleanlab/cleanlab

This isn’t just for image data. CL solutions work for

- any supervised ML model
- any data modality
- any dataset that a classifier can be trained on
- many data formats.

Filling the Gap

From research to enterprise solutions.

AI enterprise solution landscape

Obtain data and labels

(typically **<u>lower quality</u>** than you want)

Low quality data → high quality model

the gap

Deploy a trained model

(that **<u>works well</u>** for customers)

Cleanlab Studio

https://cleanlab.ai/studio/

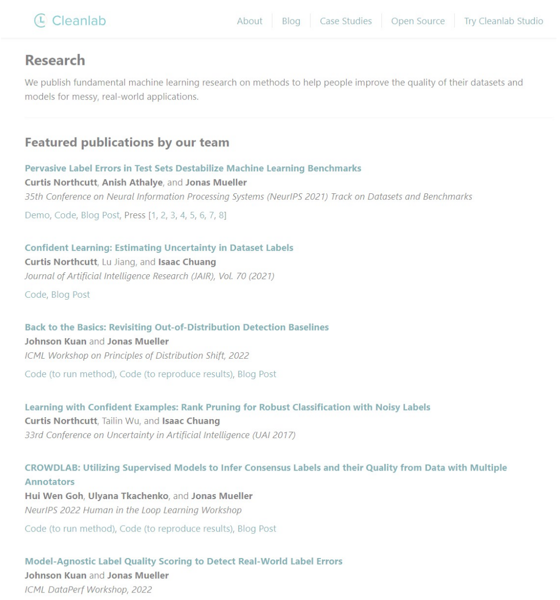
Lots of labeling, ETL, and data warehouse solutions exist

Lots of model deployment solutions exist

Cleanlab What makes us different?

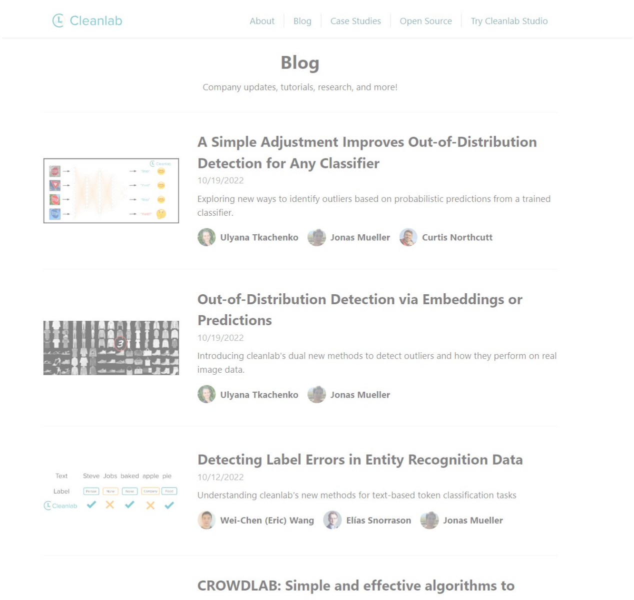
Founded by 3 PhDs in ML from MIT, we spend on producing value over ad marketing.

Research Publications



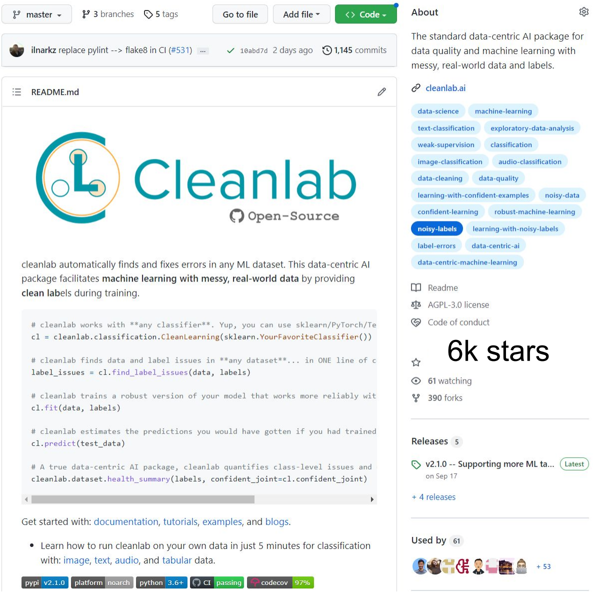
https://cleanlab.ai/research

Research Blogs



https://cleanlab.ai/blog

Open-source commitment



https://docs.cleanlab.ai/

Takeaway: Data and label quality is a problem for our market. Here are some solutions:

- https://cleanlab.ai/studio/
- https://github.com/cleanlab/cleanlab

Questions? → team@cleanlab.ai

Slide intentionally left blank