Kern AI

Greenfield vs. Brownfield Data Labeling to improve AI performance

Very short **intro to labeling**

Dear Mr. Müller,

[...]
The accident happened on 07 November between 8 and 8:30 on the way to work. In the traffic jam, the person behind me drove up too close, so that there are now two larger dents on the rear of my VW Golf. What is the best way to proceed?

Best,
Johannes Hötter

manual labeling

Dear Mr. Müller,

[...]
The accident happened on 07 November between 8 and 8:30 on the way to work. In the traffic jam, the person behind me drove up too close, so that there are now two larger dents on the rear of my VW Golf. What is the best way to proceed?

Best,
Johannes Hötter

Issue: *car damage*

Green- and brownfield

Kern AI

New projects

Design from scratch

New environment

Existing projects

Integration into systems

Working with legacy

Green- and brownfield **in ML**

Only raw data

From 0% to 90%

Proof of concept

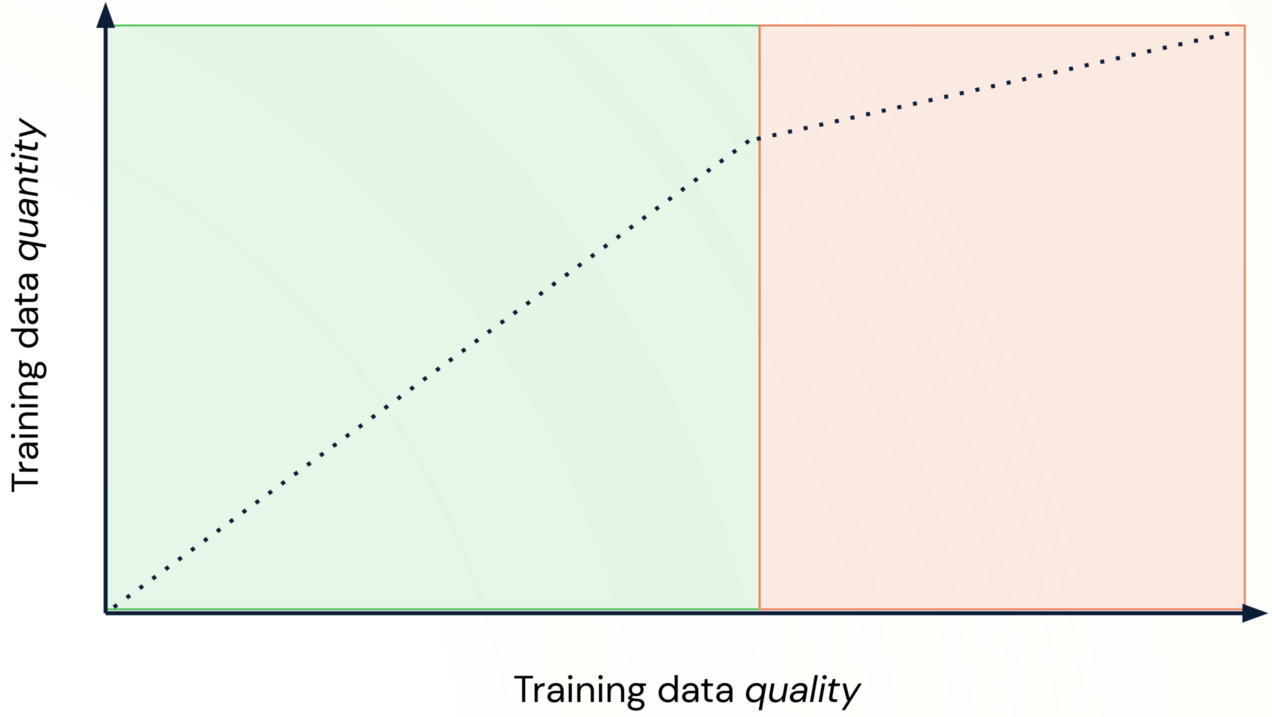
Training data is available

From 90% to 95%

Continuous improvement

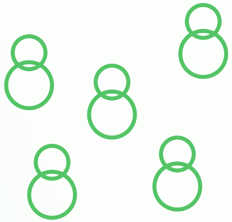
Green- and brownfield **in ML**

Kern AI



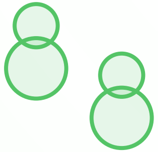
Labeling **from scratch**

Crowd Labeling



Globally scalable, but not designed for
knowledge-intensive, privacy-sensitive tasks

Inhouse Labeling



Highest possible domain-knowledge,
but not scalable and efficient at all

Weak supervision – a ML-oriented information integration

Kern AI

	Heuristic #1	Heuristic #2	...	Heuristic #N	**Weakly supervised**
Record #1	News	none		News	78.2% News
Record #2	none	Politics		Sports	48.1% Politics
...					
Record #N	none	none		News	32.6% News

Types of **heuristics**

Kern AI

- Labeling functions

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit:
        return "Clickbait"
```

- Distant supervision (lookup values)
- Active (transfer) learning modules
- Zero-shot classifiers (e.g. from Hugging Face)
- Unexperienced labelers (e.g. crowdlabeling)
- 3rd party systems, legacy systems, …

Interface to collect noisy labels; Relevance of each heuristic can be derived from e.g. manually labeled reference data

But why **train** then?

Kern AI

Labeling

- Automate to *build*
- Runtime doesn't matter (not user-facing)
- Potential to include data that isn't available at runtime
- Aim for highest confidence predictions

Inference

- Automate to *run*
- Inference often required within ms
- Prediction for every record
- Rule of thumb: Model learns to further generalize

Manual labeling still matters!

Kern AI

Exploring data

Reference data

Ideating automation

Human performance

Strategies for manual labeling

Kern AI

Neural search

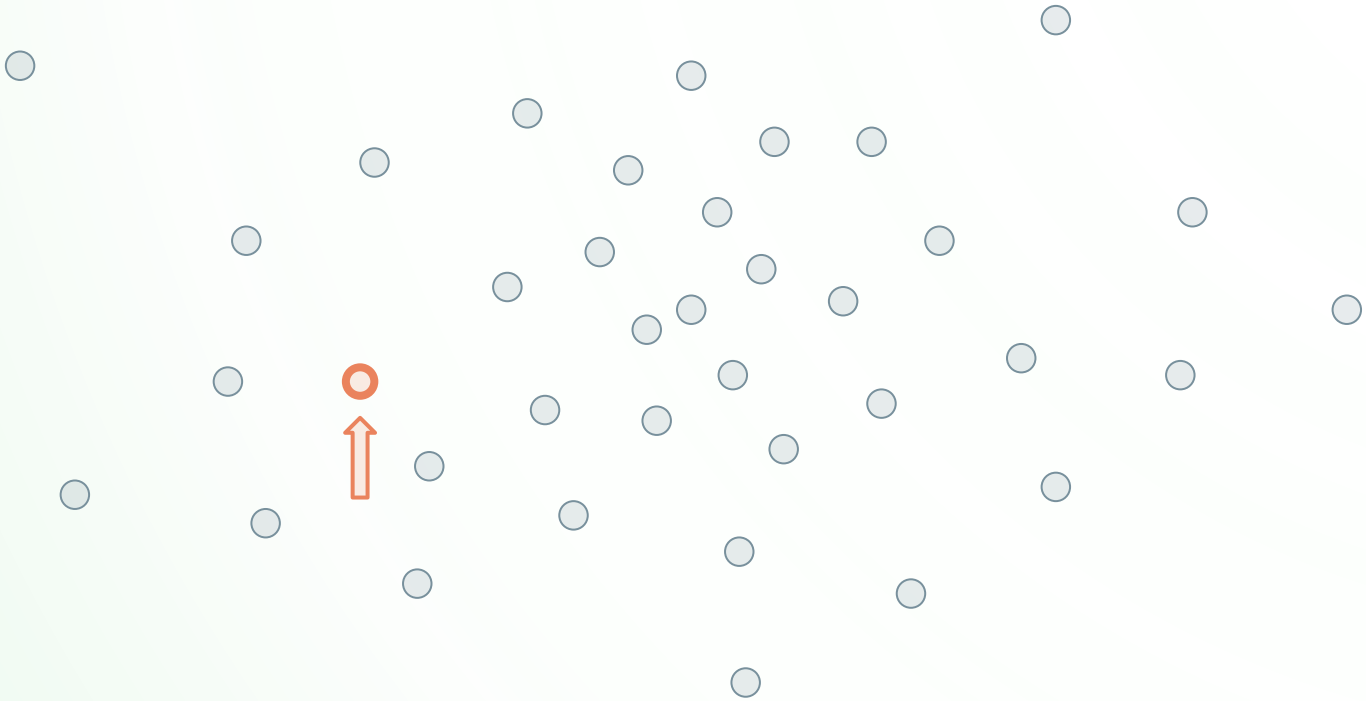
Random sampling

Heuristics validation

People filter

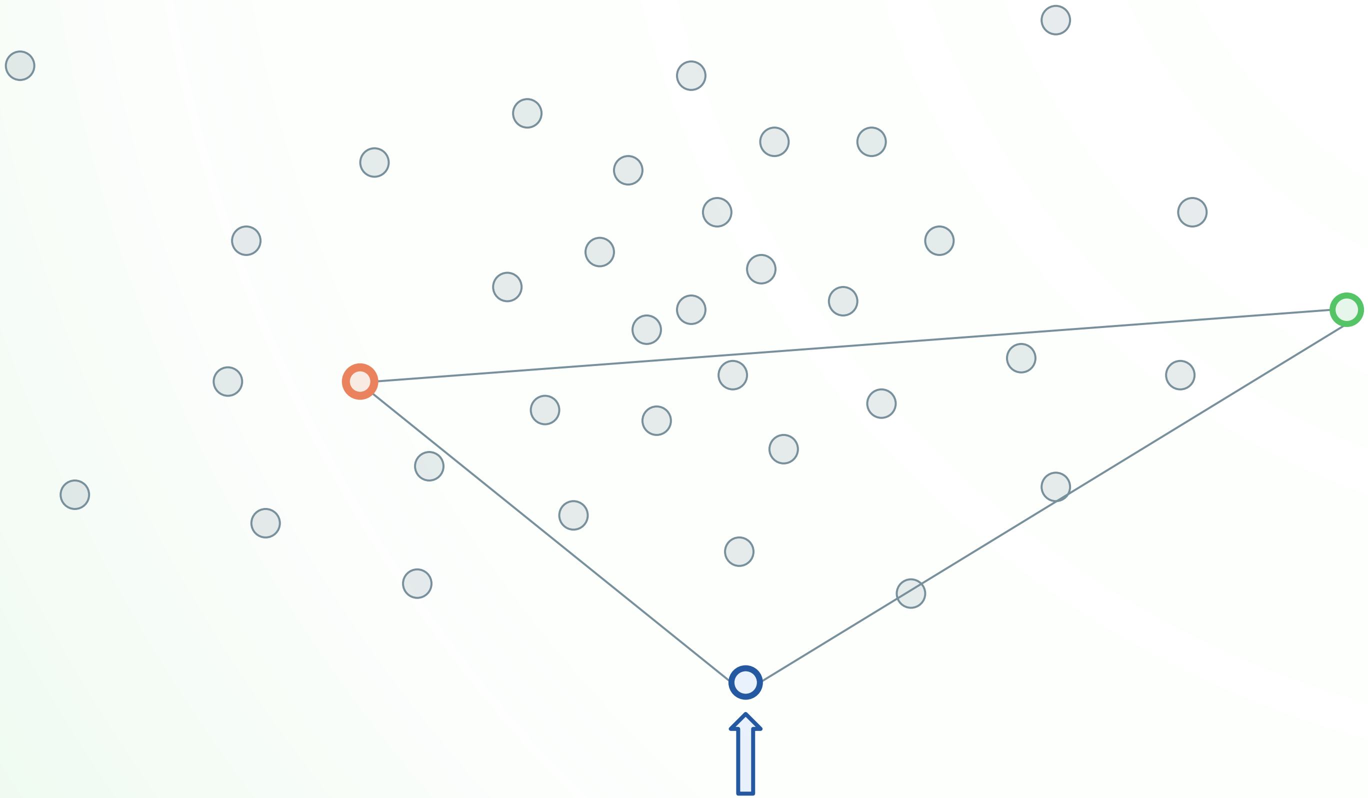
Neural search–based manual labeling for diversity

Neural search–based manual labeling for diversity



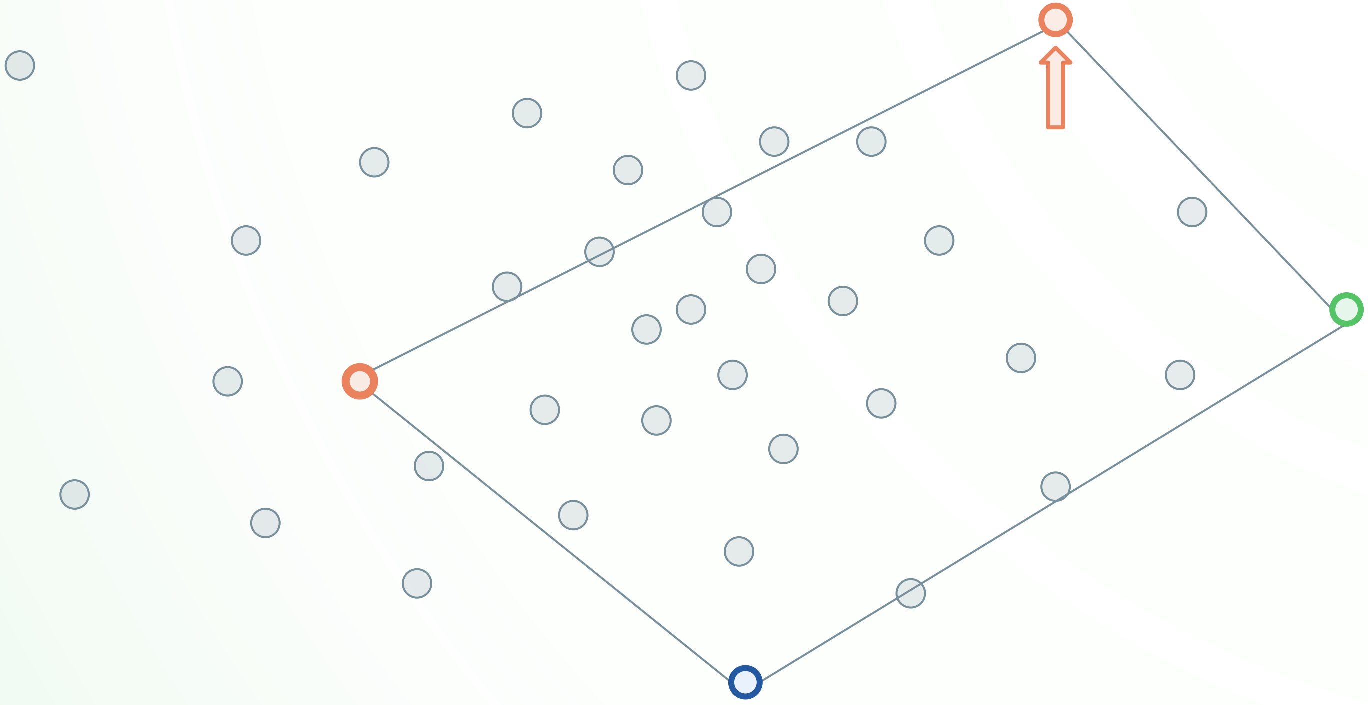
Neural search–based manual labeling for diversity

Neural search-based manual labeling for diversity

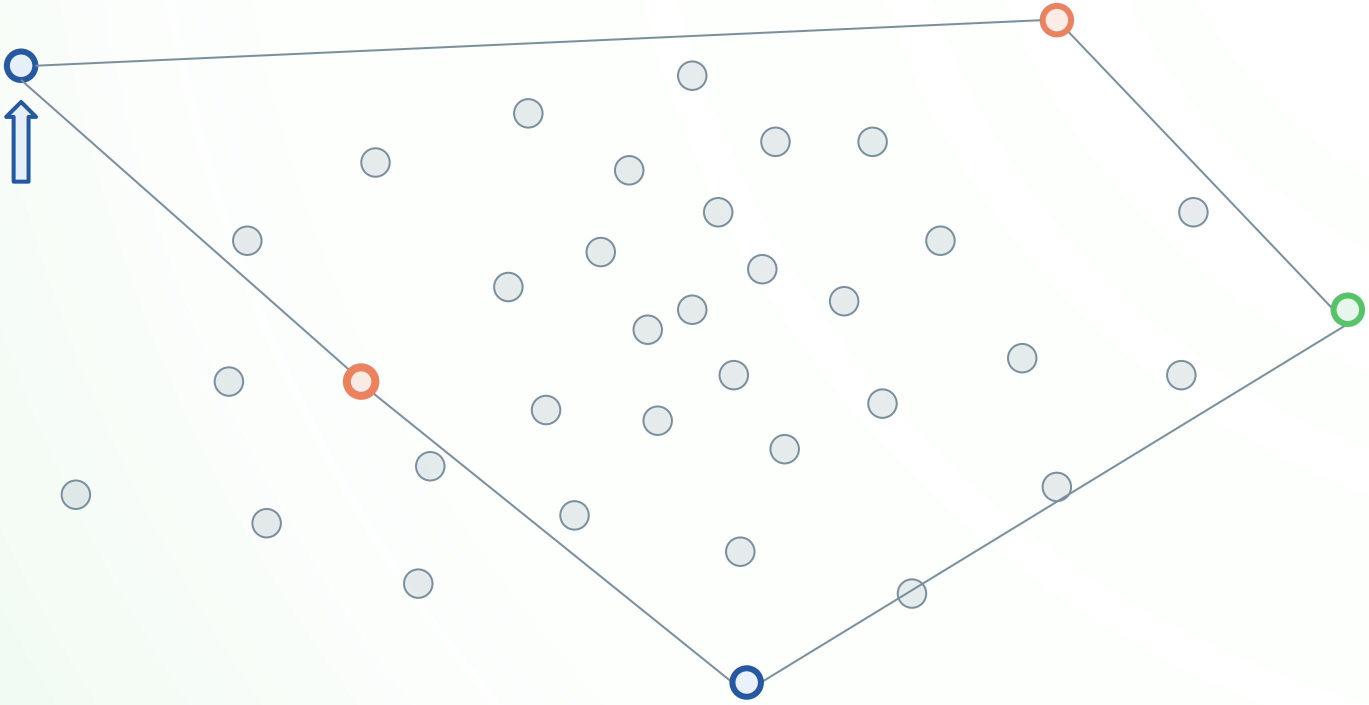


Neural search–based manual labeling for diversity

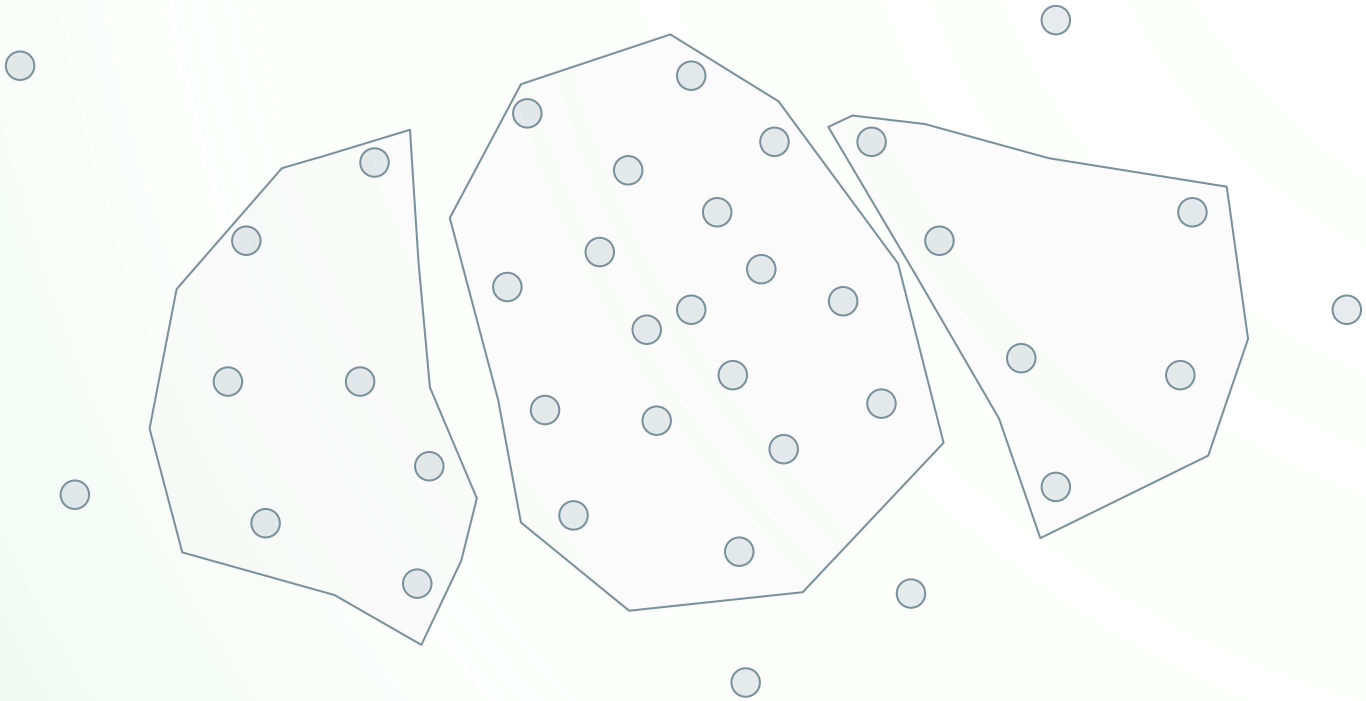
Kern[AI]



Neural search–based manual labeling for diversity



Neural search–based manual labeling by clustering



Debugging data

Only in rare occasions, your training data will be *super* clean. In real world, it is mostly messy.

This is what we mean with **brownfield**; improving existing training data.

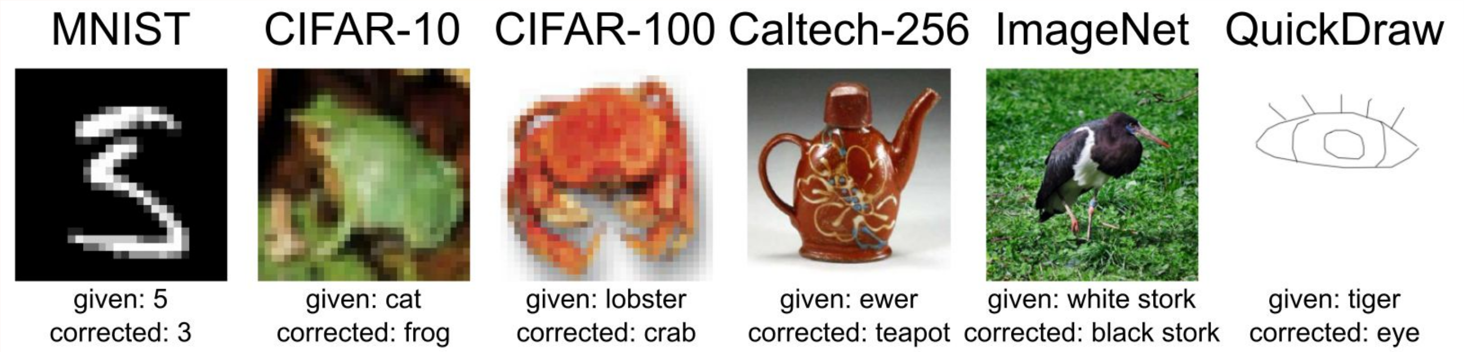


Image source: https://github.com/cleanlab/cleanlab

Confident Learning

Kern AI

Model outcomes | **Prediction** | **Noisy Label***

```
[
  [
    0.7, 0.1, 0.2  ------>  Sports      News
  ],[
    0.3, 0.4, 0.3  ------>  Politics    Politics
  ], …
]
```

order: [Sports, News, Politics]

*known to potentially differ from the *ground truth*

Confident Learning

Kern AI

With a trained model, you can estimate the joint distribution of noisy and true labels …

	News	Sports	Politics
News	0.2	0.08	0.06
Sports	0.05	0.15	0,03
Politics	0.01	0.02	0.4

*known to potentially differ from the *ground truth*

Confident Learning

… and estimate the number of errors

	News	Sports	Politics	
News	0.2	0.08	0.06	sum = 0.75
Sports	0.05	0.15	0,03	sum = 0.25 (error factor)
Politics	0.01	0.02	0.4	

*known to potentially differ from the *ground truth*

Confident Learning

Model outcomes

```
[
  [
    0.7, 0.1, 0.2
  ],[
    0.3, 0.4, 0.3
  ], …
]
```

order: [Sports, News, Politics]

We can turn these into confidence scores, sort by them ascending, and take the first 25% (error rate) as potential label errors

Debugging heuristics

Kern AI

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit:
        return "Clickbait"
```

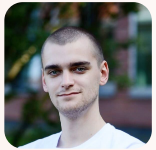
Precision 83%
Coverage 2.5%

analyze filter where `starts_with_digit == "Clickbait"`

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit
    and record["sentiment"] > 0.7:
        return "Clickbait"
```

Precision 92%
Coverage 1.8%

Inter annotator agreement and bias in heuristics

			
		92.5%	98%
	92.5%		87.5%
	98%	87.5%	

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit:
        return "Clickbait"
```

95% precision

75% precision

Some food for **thoughts**

- As training data is an integral part of ML-applications, what will the *maintenance/documentation* of data look like?
- How can *programmable* labeling empower data *debugging*? Will focused *labeling* shift towards holistic *enrichments* instead?

Some great **resources** to dive into 🤓

Kern AI

Weak supervision

"Data Programming: Creating Large Training Sets, Quickly" by Ratner et al.

(great resource from the founders of Snorkel)

Confident learning

"Confident Learning: Estimating Uncertainty in Dataset Labels" by Northcutt et al.

(also comes with a great blog post here: https://l7.curtisnorthcutt.com/confident-learning)

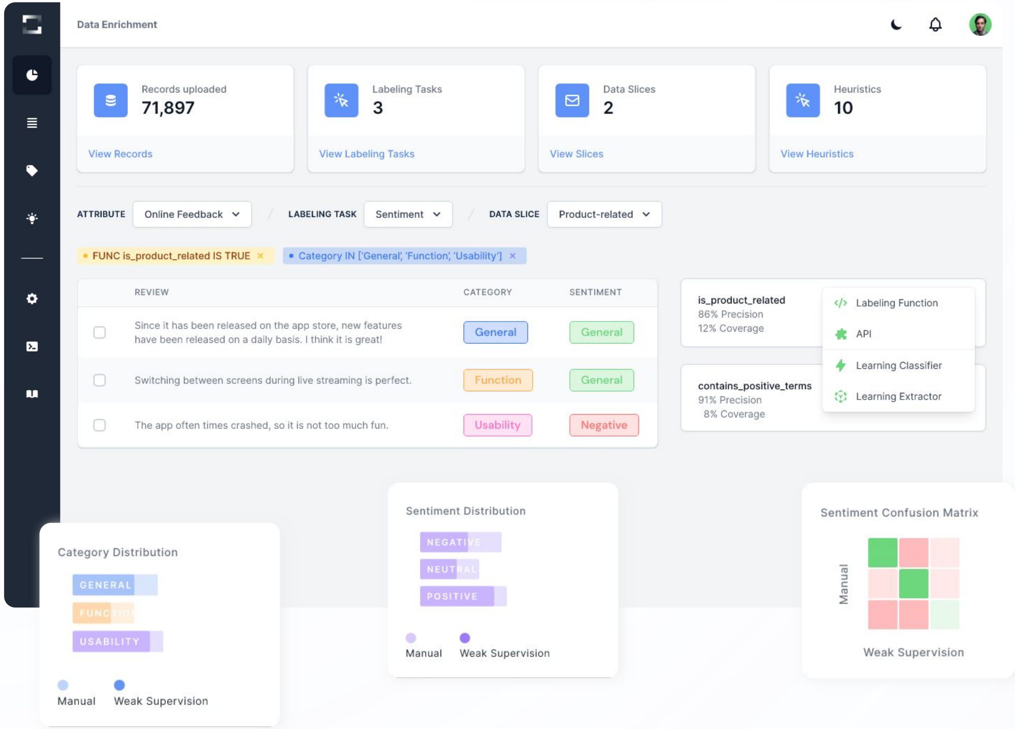
Neural search

Open-source vector databases like qdrant.tech

Our own research about integrating such technologies will be published at **NLDB 2022**, stay tuned

("kern: a labeling environment for large-scale, high-quality training data")

We're **open-sourcing** toolkits for data-centric AI



Register for our newsletter on our website www.kern.ai

Feel free to reach out 😃

Johannes Hötter

johannes.hoetter@kern.ai

https://www.linkedin.com/in/johanneshötter/

Thanks for having me!