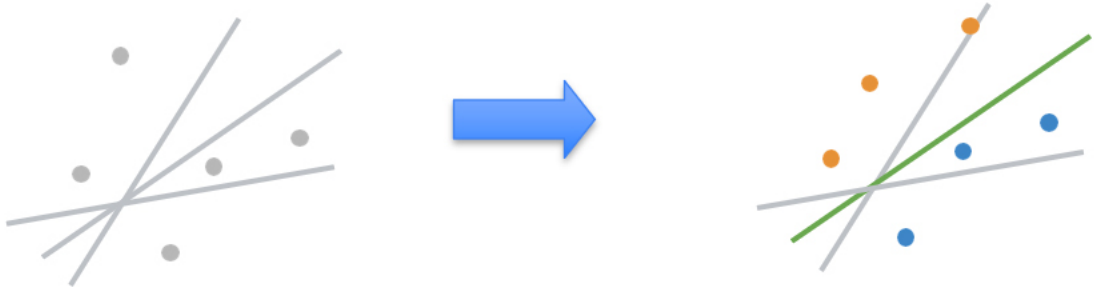
Data Lifecycle in Production

Part 2: Advanced Labeling, Augmentation and Data Preprocessing

Ramin Toosi

Why is Advanced Labeling Important?

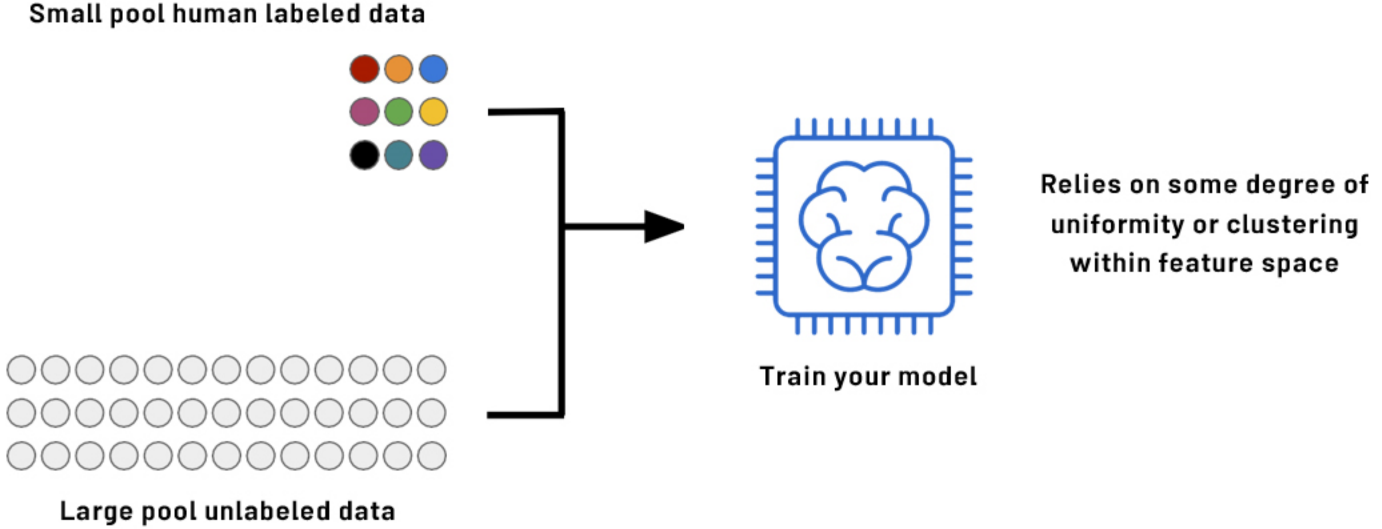
- Manually labeling of data is expensive
- Unlabeled data is usually cheap and easy to get
- Unlabeled data contains a lot of information that can improve our model



ML use is growing everywhere

...as is the need for labeled training sets

Human labeling, Semi-supervised



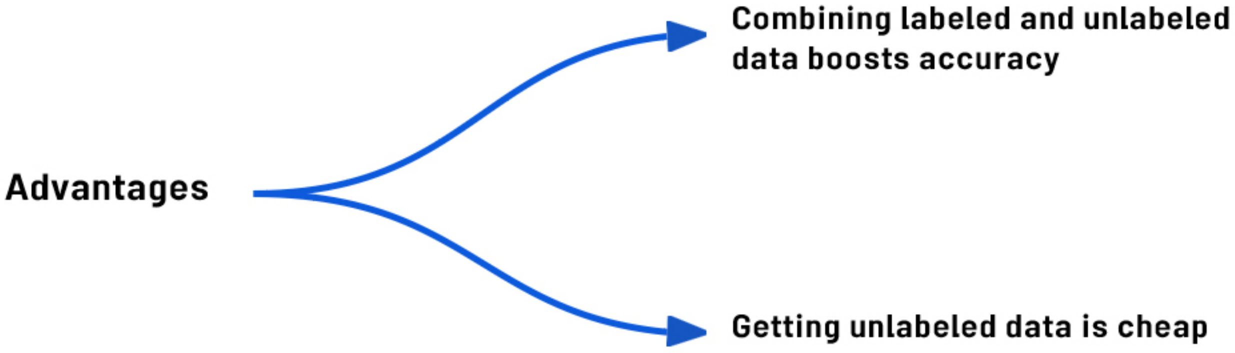
Small pool human labeled data

Large pool unlabeled data

Train your model

Relies on some degree of uniformity or clustering within feature space

Human labeling, Semi-supervised



Advantages

Combining labeled and unlabeled data boosts accuracy

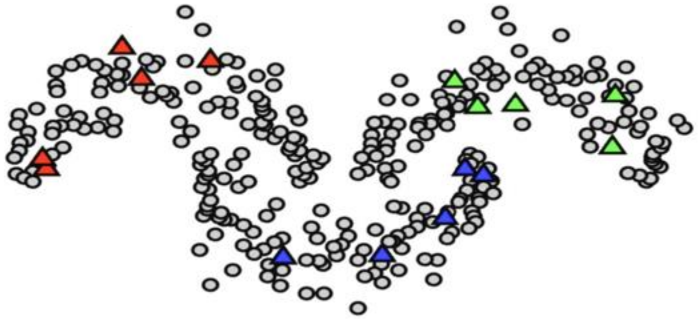
Getting unlabeled data is cheap

Label propagation

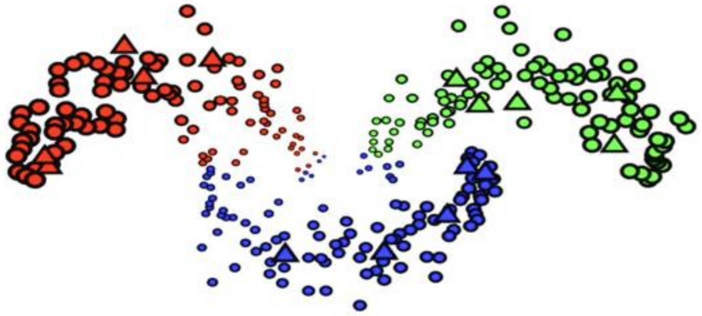
- Semi-supervised ML algorithm
- A subset of the examples have labels
- Labels are propagated to the unlabeled points:
  - Based on similarity or "community structure"

Label propagation - Graph based

- **Unlabeled examples can be assigned labels based on their neighbors**



Partially labeled

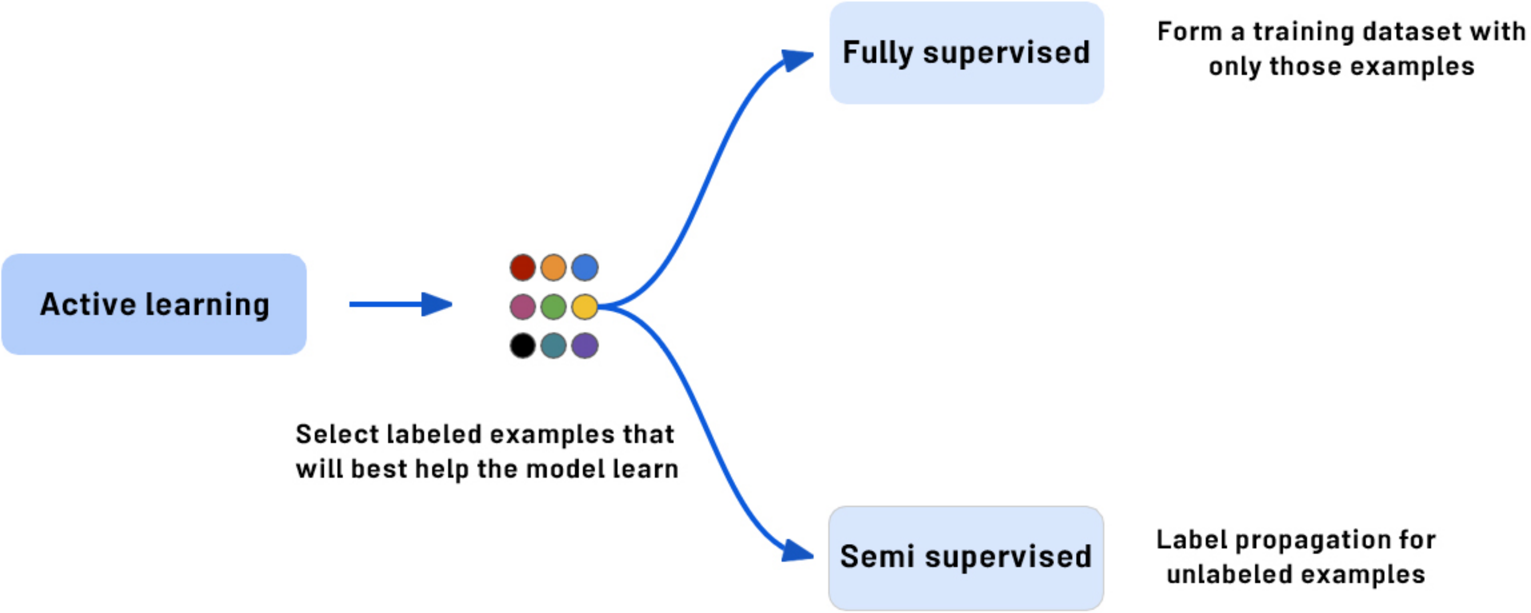


Labels propagated

Active learning

- A family of algorithms for intelligently sampling data
- Select the points to be labeled that would be most informative for model training
- Very helpful in the following situations:
  - Constrained data budgets: you can only afford labeling a few points
  - Imbalanced dataset: helps selecting rare classes for training
  - Target metrics: when baseline sampling strategy does not improve selected metrics

Active learning strategies



Active learning

Select labeled examples that will best help the model learn

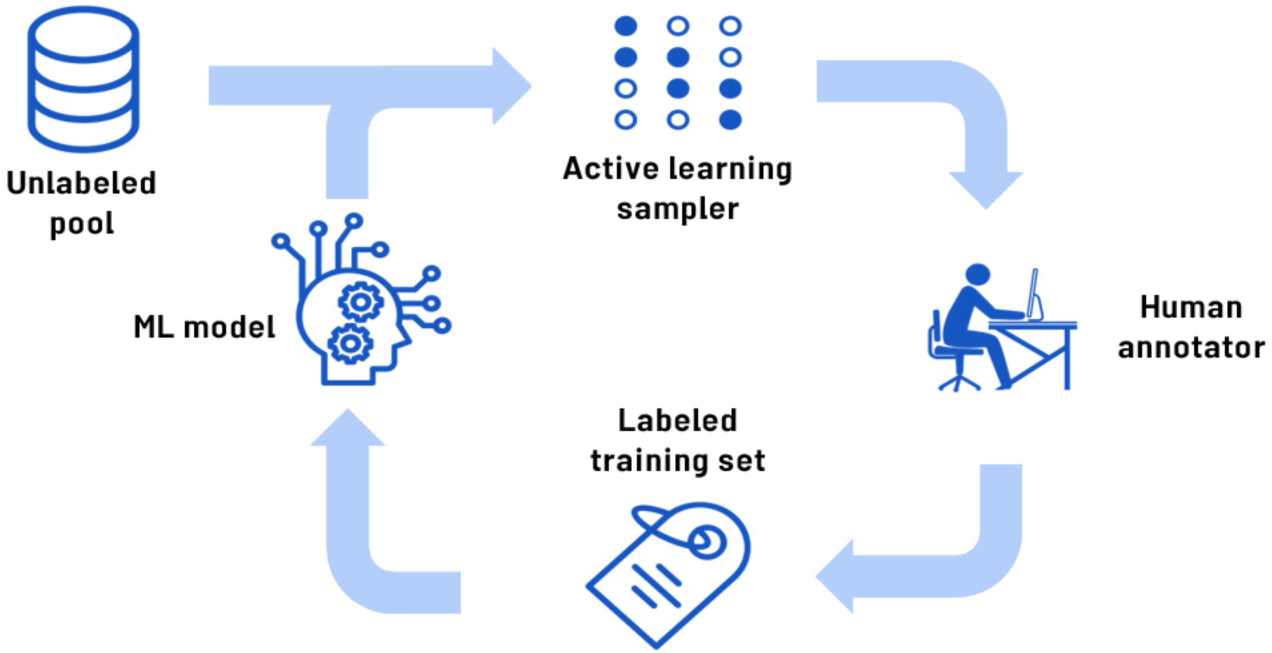
Fully supervised

Form a training dataset with only those examples

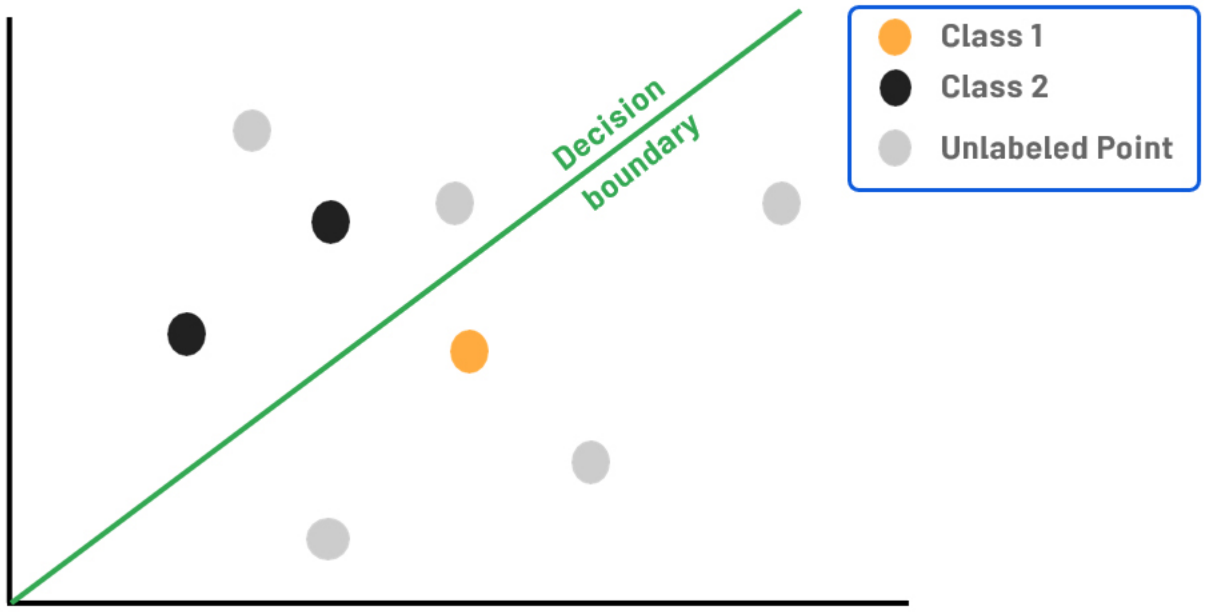
Semi supervised

Label propagation for unlabeled examples

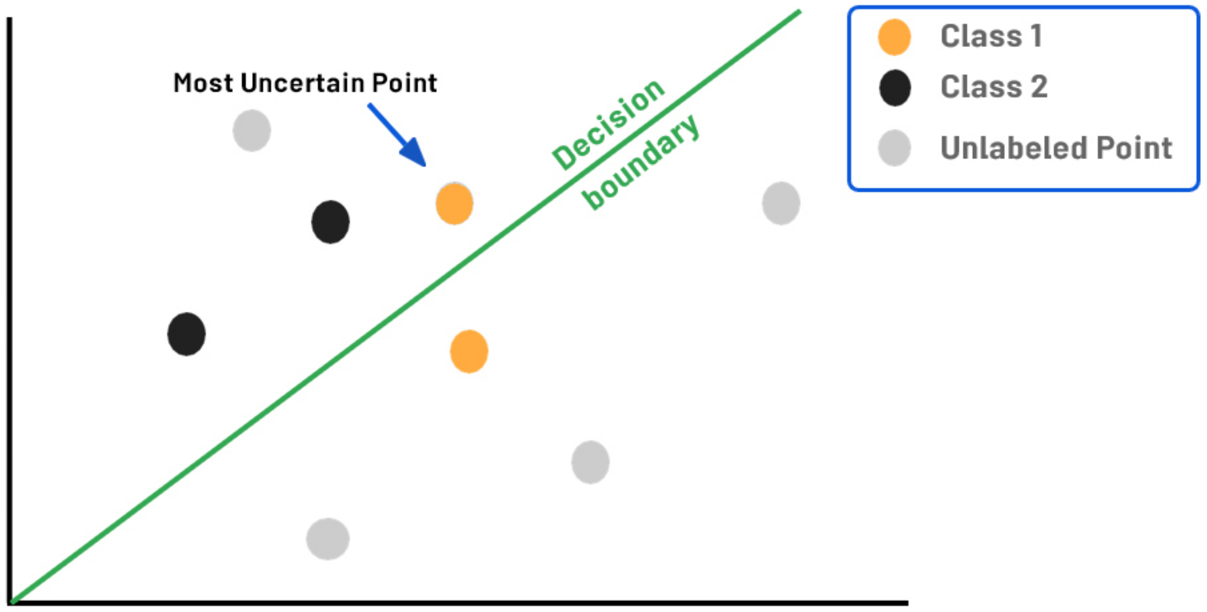
Active learning cycle



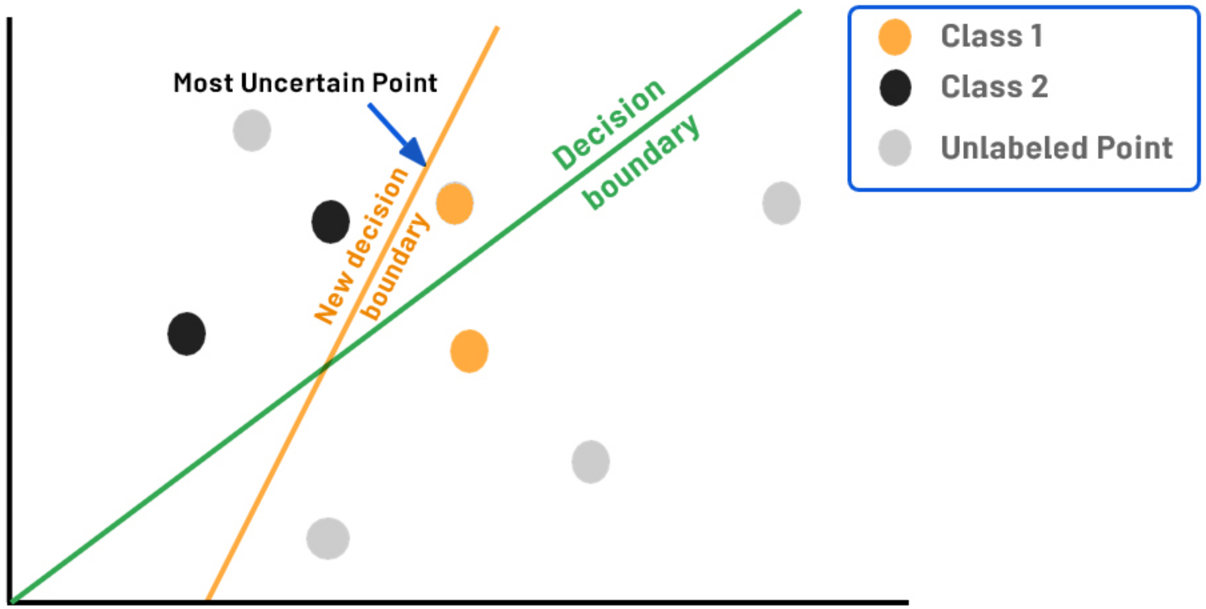
Margin sampling



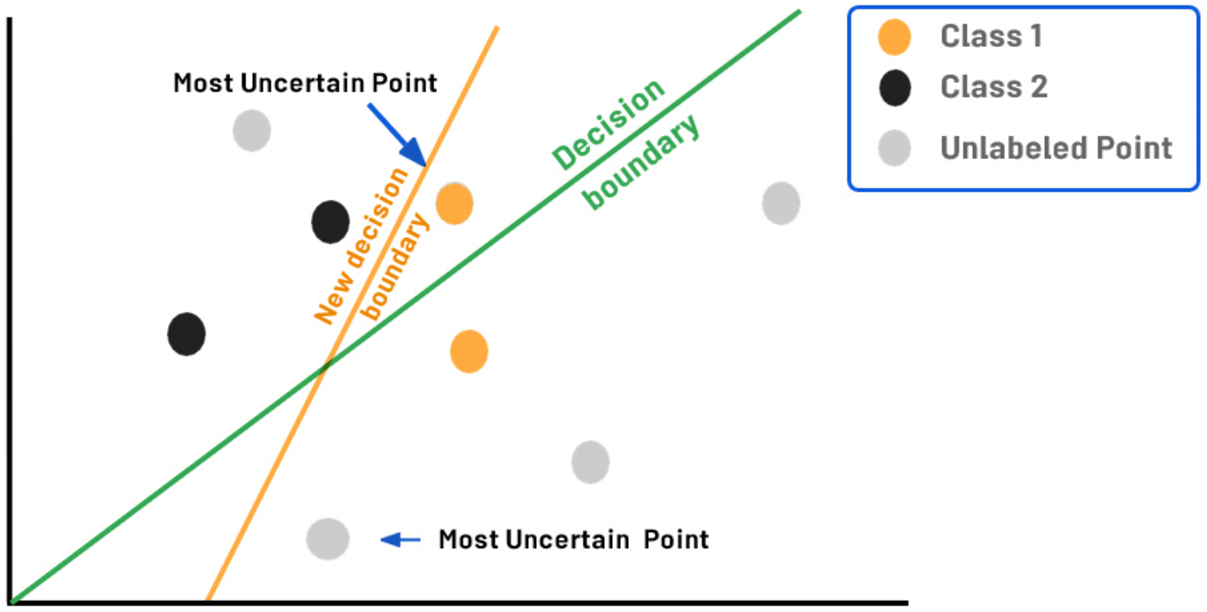
Margin sampling



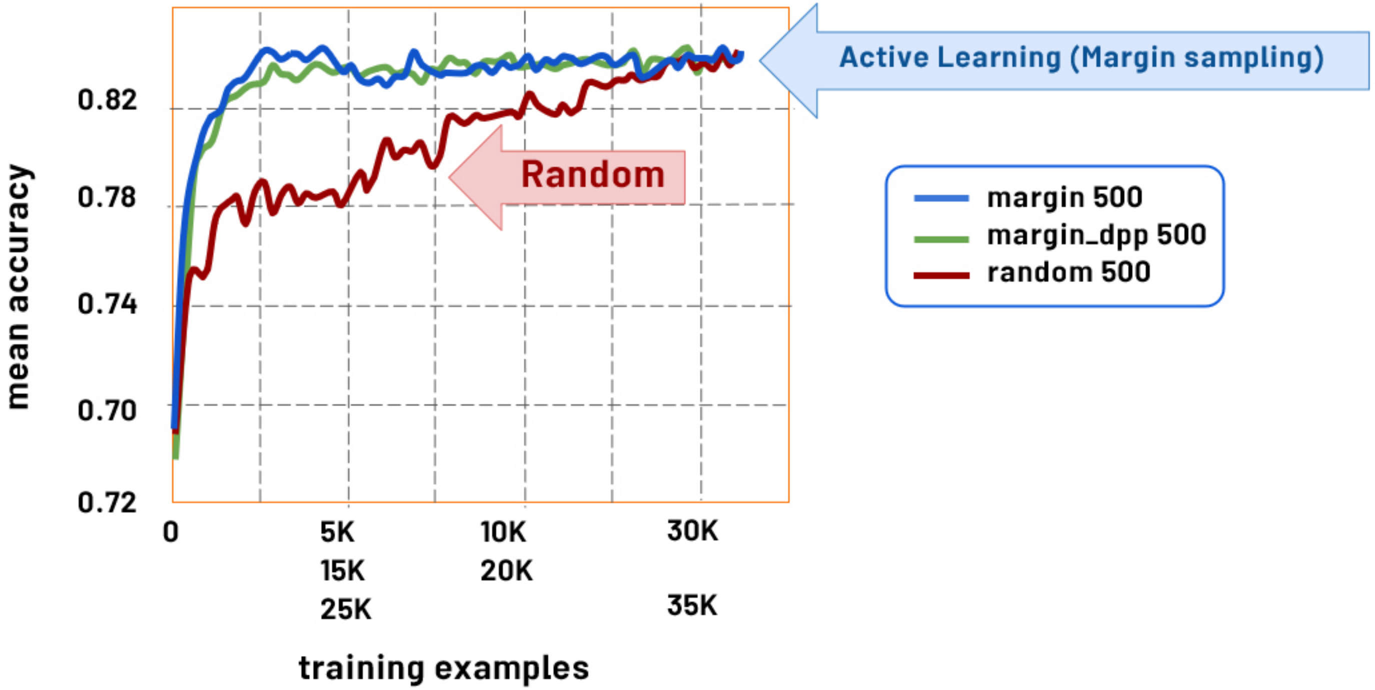
Margin sampling



Margin sampling



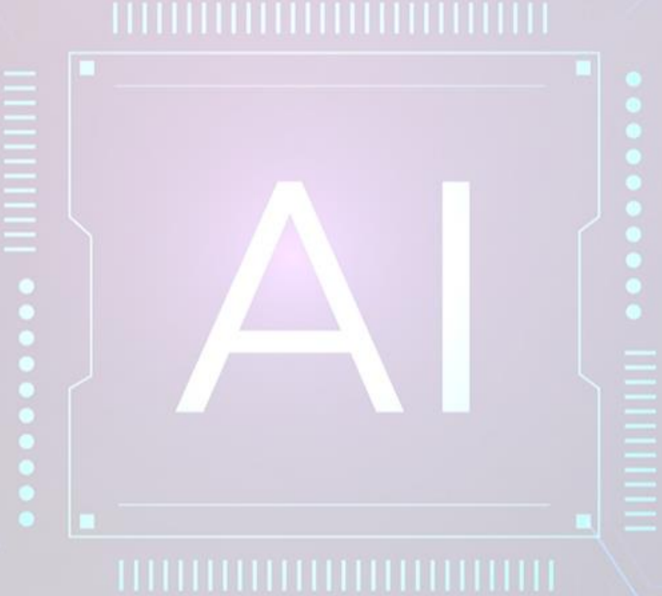
Example results - Different Sampling Techniques



Active learning sampling techniques

- **Margin sampling:** Label points the current model is least confident in.
- **Cluster-based sampling:** sample from well-formed clusters to "cover" the entire space.
- **Query-by-committee:** train an ensemble of models and sample points that generate disagreement.
- **Region-based sampling:** Runs several active learning algorithms in different partitions of the space.

Weak Supervision



Hand labeling: intensive labor

- "Hand-labeling training data for machine learning problems is effective, but very labor and time intensive. This work explores how to use algorithmic labeling systems relying on other sources of knowledge that can provide many more labels but which are noisy."

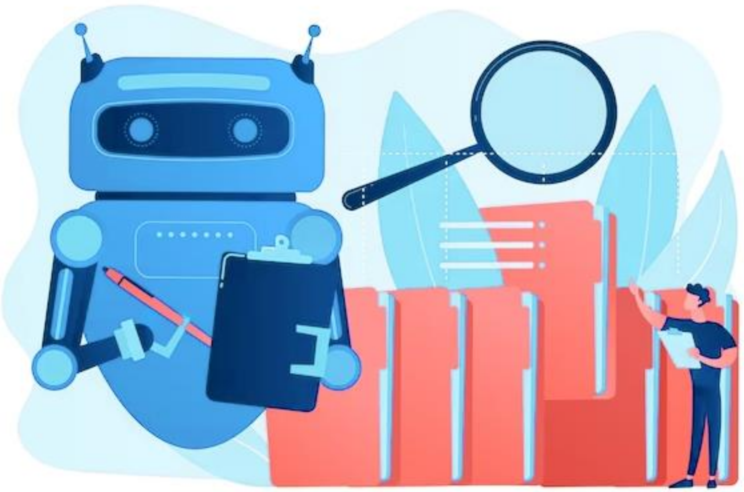
Jeff Dean, March 14, 2019

Weak Supervision

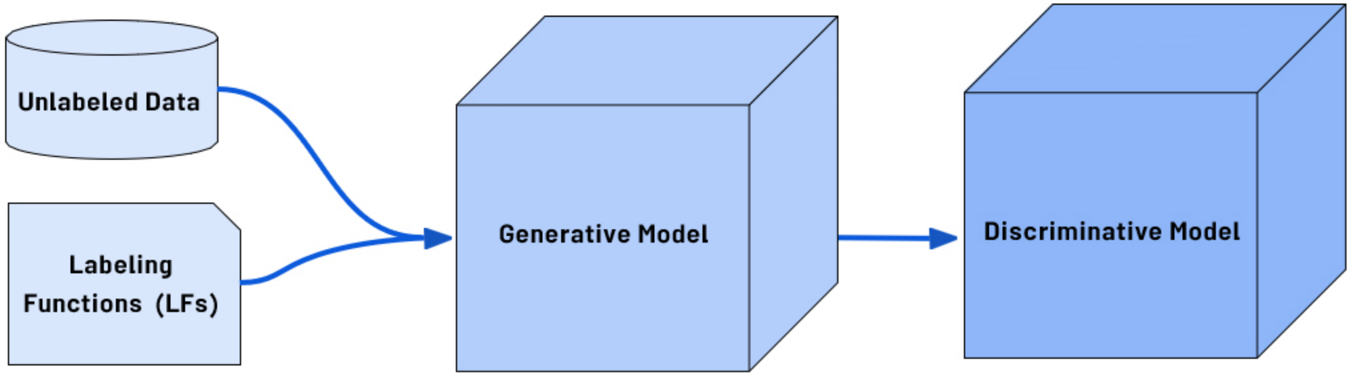
- Unlabeled data, without ground-truth labels
- One or more weak supervision sources
  - A list of heuristics that can automate labeling
  - Typically provided by subject matter experts
- Noisy labels have a certain probability of being correct, not 100%
- Objective: learn a generative model to determine weights for weak supervision sources

Snorkel

- Project started at Stanford in 2016
- Programmatically building and managing training datasets without manual labeling
- Automatically: models, cleans, and integrates the resulting training data
- Applies novel, theoretically-grounded techniques
- Also offers data augmentation and slicing



Data programming pipeline in Snorkel



Users write labeling functions to generate noisy labels for unlabeled data

A generative model is used to de-noise and weight the labels

The labels are used to train a model

Snorkel labeling functions

```
from snorkel.labeling import labeling_function


@labeling_function()
def
lf_keyword_my(x):
    """Many spam comments talk about 'my channel', 'my video',
    etc."""  return SPAM if "my" in x.text.lower() else ABSTAIN


@labeling_function()
def
lf_short_comment(x):
    """Non-spam comments are often short, such as 'cool
    video!'."""  return NOT_SPAM if len(x.text.split()) < 5 else
    ABSTAIN
```