INTRODUCTION TO DATA-CENTRIC AI

Lecture 3

Data-centric Evaluation of ML Models

https://dcai.csail.mit.edu

Most Machine Learning applications

1. Collect data and define the appropriate ML task
2. Explore data to see if it has problems
3. Preprocess data into a format suitable for ML modeling
4. Train a straightforward ML model that is expected to perform reasonably.

Most Machine Learning applications

1. Collect data and define the appropriate ML task

2. Explore data to see if it has problems

3. Preprocess data into a format suitable for ML modeling

4. Train a straightforward ML model that is expected to perform reasonably.

5. Investigate shortcomings of the model and the dataset

Most Machine Learning applications

1. Collect data and define the appropriate ML task
2. Explore data to see if it has problems
3. Preprocess data into a format suitable for ML modeling
4. Train a straightforward ML model that is expected to perform reasonably.
5. Investigate shortcomings of the model and the dataset
6. Improve dataset to address its shortcomings
7. Improve model (architecture changes, regularization, hyperparameter tuning, ensembling different models)
8. Deploy model and monitor subsequent data for new issues

Most Machine Learning applications

1. Collect data and define the appropriate ML task
2. Explore data to see if it has problems
3. Preprocess data into a format suitable for ML modeling
4. Train a straightforward ML model that is expected to perform reasonably.
5. **Investigate shortcomings of the model and the dataset**
6. Improve dataset to address its shortcomings
7. Improve model (architecture changes, regularization, hyperparameter tuning, ensembling different models)
8. Deploy model and monitor subsequent data for new issues

Topics of this lecture

- Evaluation of ML models (a prerequisite for improving them)
- Handling poor model performance for some particular subpopulation
- Measuring the influence of individual datapoints on the model

Recap of Multi-class Classification

Given: training dataset $\mathcal{D}$ with n examples: $(x_i, y_i) \sim P_{XY}$

Recap of Multi-class Classification

Given: training dataset $\mathcal{D}$ with n examples: $(x_i, y_i) \sim P_{XY}$

Goal: Use $\mathcal{D}$ to train a model M, which given an example with *new* feature values x, produces a vector of predicted class probabilities $M(x) = [p_1, \ldots, p_K]$ whose kth entry approximates $P(Y = k \mid X = x)$.

Recap of Multi-class Classification

Given: training dataset $\mathcal{D}$ with n examples: $(x_i, y_i) \sim P_{XY}$

Goal: Use $\mathcal{D}$ to train a model M, which given an example with *new* feature values x, produces a vector of predicted class probabilities $M(x) = [p_1, \ldots, p_K]$ whose kth entry approximates $P(Y = k \mid X = x)$.

For a particular *loss function* that scores each model prediction, we seek a model M that optimizes:

$$\min_M \; \mathbb{E}_{(x,y)\sim P_{XY}} \left[\mathrm{Loss}\big(M(x), y\big) \right]$$

Key Assumptions

1. Data encountered during deployment will stem from the same distribution P_{XY} as our training data $\mathcal{D}$.
2. Training data (x_i, y_i) are independent and identically distributed (IID).
3. Each example belongs to exactly **one** class.

Evaluation of ML models

Loss function evaluates model predictions for a new example vs its given label

Loss may be function of:

1. The predicted class $\hat{y} \in \{1, 2, \ldots, K\}$ deemed most likely for x.

Evaluation of ML models

Loss function evaluates model predictions for a new example vs its given label

Loss may be function of:

1. The predicted class $\hat{y} \in \{1, 2, \ldots, K\}$ deemed most likely for x.

Examples of such classification losses: accuracy, balanced accuracy, precision, recall, …

Evaluation of ML models

Loss function evaluates model predictions for a new example vs its given label

Loss may be function of:

1. The predicted class $\hat{y} \in \{1, 2, \ldots, K\}$ deemed most likely for x.

Examples of such classification losses: accuracy, balanced accuracy, precision, recall, …

2. The predicted probabilities $[p_1, p_2, \ldots, p_K] \in \mathbb{R}^K$ of each class for x.

Evaluation of ML models

Loss function evaluates model predictions for a new example vs its given label

Loss may be function of:

1. The predicted class $\hat{y} \in \{1, 2, \ldots, K\}$ deemed most likely for x.

   Examples of such classification losses: accuracy, balanced accuracy, precision, recall, …

2. The predicted probabilities $[p_1, p_2, \ldots, p_K] \in \mathbb{R}^K$ of each class for x.

   Examples of such classification losses include: log loss, AUROC, calibration error,…

Reporting Model Performance

- Not ideal to rely on a single score to summarize how good your model is overall
  - But what everybody does 🤷‍♂️

Reporting Model Performance

- Not ideal to rely on a single score to summarize how good your model is overall
  - But what everybody does 🤷‍♂️

- Typical score = average of $\mathrm{Loss}\big(M(x_i), y_i\big)$ over many examples held-out during training

Reporting Model Performance

- Not ideal to rely on a single score to summarize how good your model is overall
  - But what everybody does 🤷‍♂️

- Typical score = average of $\mathrm{Loss}\big(M(x_i), y_i\big)$ over many examples held-out during training

- Alternatives:
  - Average Loss for examples from each class separately (eg. per-class accuracy)
  - Report complete confusion matrix

Think about HOW you will evaluate models

- Invest as much as time thinking about this as:
  - what models to apply
  - how to improve them

- Model evaluation has HUGE impact in real applications

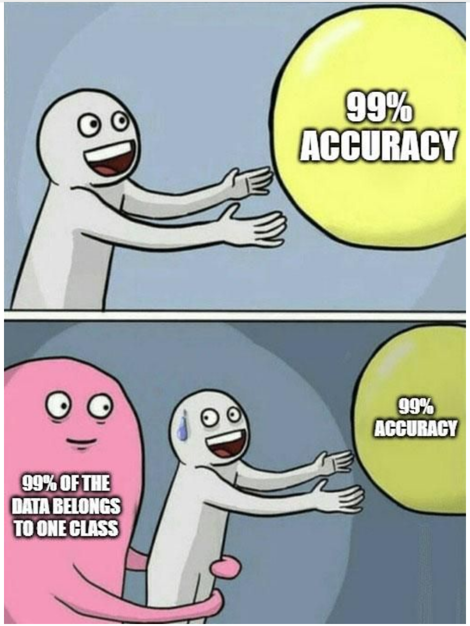
Think about HOW you will evaluate models

- Invest as much as time thinking about this as:
  - what models to apply
  - how to improve them

- Model evaluation has HUGE impact in real applications

- Consider **Fraud** vs **Not-Fraud** classification of credit card transactions
  - why not choose overall accuracy as the evaluation metric?

Think about HOW you will evaluate models



- Invest as much as time thinking about this as:
  - what models to apply
  - how to improve them
- Model evaluation has HUGE impact in real applications
- Consider **Fraud** vs **Not-Fraud** classification of credit card transactions
  - why not choose overall accuracy as the evaluation metric?

Common pitfalls when evaluating models

- Failing to use truly held-out data (data leakage)

Sloppy Use of Machine Learning Is Causing a 'Reproducibility Crisis' in Science

WIRED

Field	Paper	Number of papers reviewed	Number of papers with pitfalls	**[L1.1] No test set**	**[L1.2] Pre-proc. on train-test**	**[L1.3] Feature sel. on train-test**	**[L1.4] Duplicates**	**[L2] Illegitimate features**	**[L3.1] Temporal leakage**	**[L3.2] Non-ind. b/w train-test**	**[L3.3] Sampling bias**	Comput. reproducibility issues	Data quality issues	Metric choice issues	Standard dataset used?
Medicine	Bouwmeester et al. (2012)	71	27	○									○		
Neuroimaging	Whelan & Garavan (2014)	–	14	○		○									
Autism Diagnostics	Bone et al. (2015)	–	3				○				○		○	○	○
Bioinformatics	Blagus & Lusa (2015)	–	6		○										
Nutrition Research	Ivanescu et al. (2016)	–	4	○									○	○	
Software Eng.	Tu et al. (2018)	58	11						○			○	○		○
Toxicology	Alves et al. (2019)	–	1				○					○	○		
Satellite Imaging	Nalepa et al. (2019)	17	17							○			○		○
Tractography	Poulin et al. (2019)	4	2	○								○	○	○	○
Clinical Epidem.	Christodoulou et al. (2019)	71	48			○							○		
Brain-computer Int.	Nakanishi et al. (2020)	–	1	○											○
Histopathology	Oner et al. (2020)	–	1							○					
Neuropsychiatry	Poldrack et al. (2020)	100	53	○	○								○	○	
Medicine	Vandewiele et al. (2021)	24	21			○				○	○	○	○		○
Radiology	Roberts et al. (2021)	62	62	○			○				○	○			○
IT Operations	Lyu et al. (2021)	9	3						○						○
Medicine	Filho et al. (2021)	–	1					○							
Neuropsychiatry	Shim et al. (2021)	–	1			○						○			
Genomics	Barnett et al. (2022)	41	23			○							○		
Computer Security	Arp et al. (2022)	30	30	○	○	○		○		○	○		○	○	

Table 1. Survey of 20 papers that identify pitfalls in the adoption of ML methods across 17 fields, collectively affecting 329 papers. In each field, papers adopting ML methods suffer from data leakage. The column headings for types of data leakage, shown in bold, are based on our taxonomy of data leakage. We also highlight other issues that are reported in the papers, including issues with computational reproducibility (the availability of code, data, and computing environment to reproduce the exact results reported in the paper), data quality (for example, small size or large amounts of missing data), metric choice (using incorrect metrics for the task at hand, for example, using accuracy for measuring model performance in the presence of heavy class imbalance), and standard dataset use, where issues are found despite the use of standard datasets in a field.

Common pitfalls when evaluating models

- Failing to use truly held-out data (data leakage)
- Reporting only average loss can under-represent severe failure cases for rare examples/subpopulations (misspecified metric)

Sloppy Use of Machine Learning Is Causing a 'Reproducibility Crisis' in Science

WIRED

Field	Paper	Number of papers reviewed	Number of papers with pitfalls	**[L1.1] No test set**	**[L1.2] Pre-proc. on train-test**	**[L1.3] Feature sel. on train-test**	**[L1.4] Duplicates**	**[L2] Illegitimate features**	**[L3.1] Temporal leakage**	**[L3.2] Non-ind. b/w train-test**	**[L3.3] Sampling bias**	Comput. reproducibility issues	Data quality issues	Metric choice issues	Standard dataset used?
Medicine	Bouwmeester et al. (2012)	71	27	○									○		
Neuroimaging	Whelan & Garavan (2014)	–	14	○		○									
Autism Diagnostics	Bone et al. (2015)	–	3				○				○		○	○	○
Bioinformatics	Blagus & Lusa (2015)	–	6		○										
Nutrition Research	Ivanescu et al. (2016)	–	4	○									○	○	
Software Eng.	Tu et al. (2018)	58	11						○			○	○		○
Toxicology	Alves et al. (2019)	–	1				○					○	○		
Satellite Imaging	Nalepa et al. (2019)	17	17							○			○		○
Tractography	Poulin et al. (2019)	4	2	○								○	○	○	○
Clinical Epidem.	Christodoulou et al. (2019)	71	48			○							○		
Brain-computer Int.	Nakanishi et al. (2020)	–	1	○											○
Histopathology	Oner et al. (2020)	–	1							○					
Neuropsychiatry	Poldrack et al. (2020)	100	53	○	○								○	○	
Medicine	Vandewiele et al. (2021)	24	21			○				○	○	○	○		○
Radiology	Roberts et al. (2021)	62	62	○			○				○	○			○
IT Operations	Lyu et al. (2021)	9	3						○						○
Medicine	Filho et al. (2021)	–	1					○							
Neuropsychiatry	Shim et al. (2021)	–	1			○						○			
Genomics	Barnett et al. (2022)	41	23			○							○		
Computer Security	Arp et al. (2022)	30	30	○	○	○		○		○	○		○	○	

Table 1. Survey of 20 papers that identify pitfalls in the adoption of ML methods across 17 fields, collectively affecting 329 papers. In each field, papers adopting ML methods suffer from data leakage. The column headings for types of data leakage, shown in bold, are based on our taxonomy of data leakage. We also highlight other issues that are reported in the papers, including issues with computational reproducibility (the availability of code, data, and computing environment to reproduce the exact results reported in the paper), data quality (for example, small size or large amounts of missing data), metric choice (using incorrect metrics for the task at hand, for example, using accuracy for measuring model performance in the presence of heavy class imbalance), and standard dataset use, where issues are found despite the use of standard datasets in a field.

Common pitfalls when evaluating models

- Failing to use truly held-out data (data leakage)
- Reporting only average loss can under-represent severe failure cases for rare examples/subpopulations (misspecified metric)
- Validation data not representative of deployment setting (selection bias)
- Some labels incorrect (annotation error)

Sloppy Use of Machine Learning Is Causing a 'Reproducibility Crisis' in Science

WIRED

Field	Paper	Number of papers reviewed	Number of papers with pitfalls	**[L1.1] No test set**	**[L1.2] Pre-proc. on train-test**	**[L1.3] Feature sel. on train-test**	**[L1.4] Duplicates**	**[L2] Illegitimate features**	**[L3.1] Temporal leakage**	**[L3.2] Non-ind. b/w train-test**	**[L3.3] Sampling bias**	Comput. reproducibility issues	Data quality issues	Metric choice issues	Standard dataset used?
Medicine	Bouwmeester et al. (2012)	71	27	○									○		
Neuroimaging	Whelan & Garavan (2014)	–	14	○		○									
Autism Diagnostics	Bone et al. (2015)	–	3				○				○		○	○	○
Bioinformatics	Blagus & Lusa (2015)	–	6		○										
Nutrition Research	Ivanescu et al. (2016)	–	4	○									○	○	
Software Eng.	Tu et al. (2018)	58	11						○			○	○		○
Toxicology	Alves et al. (2019)	–	1				○					○	○		
Satellite Imaging	Nalepa et al. (2019)	17	17							○			○		○
Tractography	Poulin et al. (2019)	4	2	○								○	○	○	○
Clinical Epidem.	Christodoulou et al. (2019)	71	48			○							○		
Brain-computer Int.	Nakanishi et al. (2020)	–	1	○											○
Histopathology	Oner et al. (2020)	–	1							○					
Neuropsychiatry	Poldrack et al. (2020)	100	53	○	○								○	○	
Medicine	Vandewiele et al. (2021)	24	21			○				○	○	○	○		○
Radiology	Roberts et al. (2021)	62	62	○			○				○	○			○
IT Operations	Lyu et al. (2021)	9	3						○						○
Medicine	Filho et al. (2021)	–	1					○							
Neuropsychiatry	Shim et al. (2021)	–	1			○						○			
Genomics	Barnett et al. (2022)	41	23			○							○		
Computer Security	Arp et al. (2022)	30	30	○	○	○		○		○	○		○	○	

Table 1. Survey of 20 papers that identify pitfalls in the adoption of ML methods across 17 fields, collectively affecting 329 papers. In each field, papers adopting ML methods suffer from data leakage. The column headings for types of data leakage, shown in bold, are based on our taxonomy of data leakage. We also highlight other issues that are reported in the papers, including issues with computational reproducibility (the availability of code, data, and computing environment to reproduce the exact results reported in the paper), data quality (for example, small size or large amounts of missing data), metric choice (using incorrect metrics for the task at hand, for example, using accuracy for measuring model performance in the presence of heavy class imbalance), and standard dataset use, where issues are found despite the use of standard datasets in a field.

Common pitfalls when evaluating models

Aside: Evaluating Text Generation models

- Human Eval: 👍 vs 👎 (or Likert scale 1-5)
  - *'vibes'*

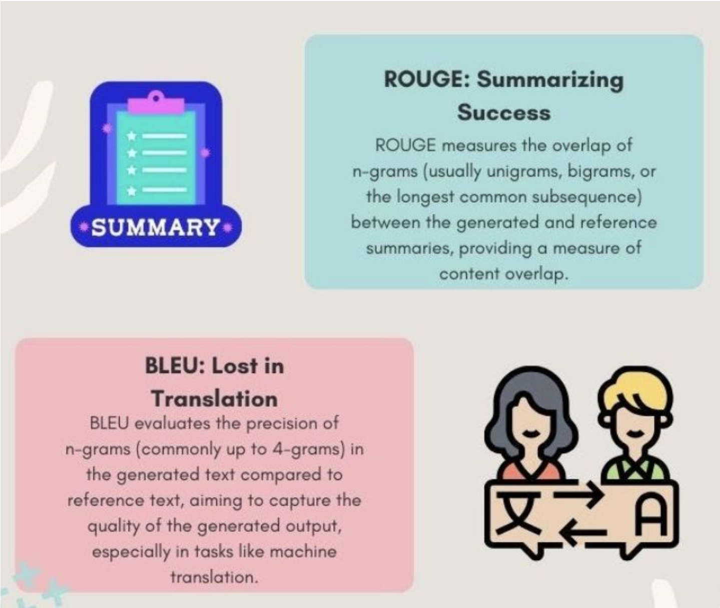
Aside: Evaluating Text Generation models

- Human Eval: 👍 vs 👎 (or Likert scale 1-5)
  - *'vibes'*

- AI (LLM) Eval: 👍 vs 👎 (or Likert scale 1-5)
  - Can give evaluator multiple binary criteria to assess

Aside: Evaluating Text Generation models

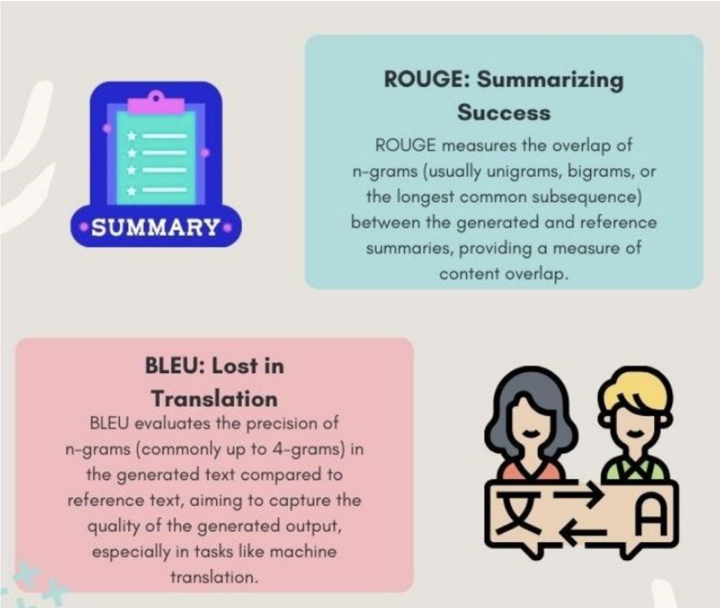
- Human Eval: 👍 vs 👎 (or Likert scale 1-5)
  - *'vibes'*
- AI (LLM) Eval: 👍 vs 👎 (or Likert scale 1-5)
  - Can give evaluator multiple binary criteria to assess
- Text similarity with target response (word overlap, ROUGE, BLEU)
- LLM likelihood of target response: Perplexity



Aside: Evaluating Text Generation models

- Human Eval: 👍 vs 👎 (or Likert scale 1-5)
  - *'vibes'*
- AI (LLM) Eval: 👍 vs 👎 (or Likert scale 1-5)
  - Can give evaluator multiple binary criteria to assess
- Text similarity with target response (word overlap, ROUGE, BLEU)
- LLM likelihood of target response: Perplexity

Challenge: Eval data seen during pre-training? (data leakage)



Underperforming Subpopulations

MIT News

ON CAMPUS AND AROUND THE WORLD

SUBSCRIBE

Study finds gender and skin-type bias in commercial artificial-intelligence systems

Examination of facial-analysis software shows error rate of 0.8 percent for light-skinned men, 34.7 percent for dark-skinned women.

Underperforming Subpopulations

data slice = a subset of the dataset that shares a common characteristic

- cohorts, subpopulation, or subgroup

Examples:

- data captured via: one sensor vs another, one location vs another

Underperforming Subpopulations

data slice = a subset of the dataset that shares a common characteristic

- cohorts, subpopulation, or subgroup

Examples:

- data captured via: one sensor vs another, one location vs another
- factors in human-centric data like:
    - race, gender, socioeconomics, age, …

Underperforming Subpopulations

data slice = a subset of the dataset that shares a common characteristic

- cohorts, subpopulation, or subgroup

Examples:

- data captured via: one sensor vs another, one location vs another
- factors in human-centric data like:
  - race, gender, socioeconomics, age, …

Model predictions should not depend on which slice a datapoint belongs to

- Can we just deleting slice information from our feature values before model training?

Underperforming Subpopulations

data slice = a subset of the dataset that shares a common characteristic

- cohorts, subpopulation, or subgroup

Examples:

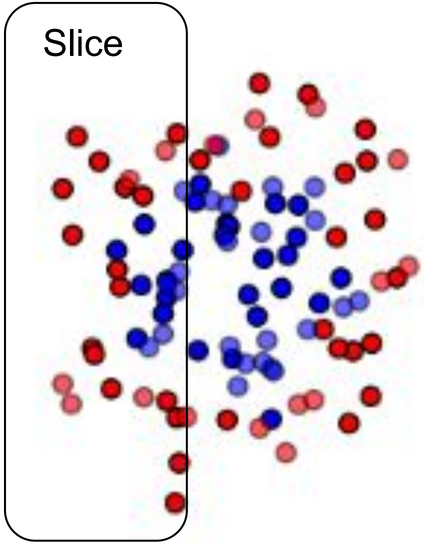
- data captured via: one sensor vs another, one location vs another
- factors in human-centric data like:
  - race, gender, socioeconomics, age, …

Model predictions should not depend on which slice a datapoint belongs to

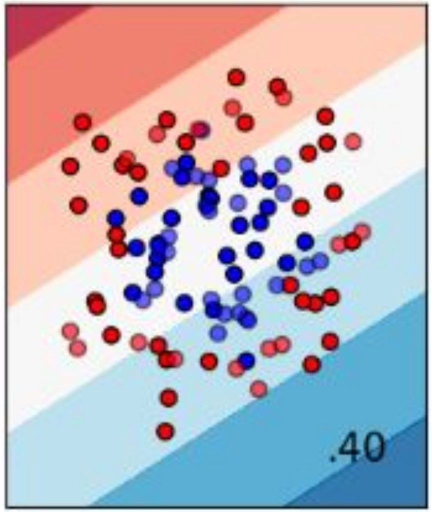
- Can we just deleting slice information from our feature values before model training?
  NO slice information can be correlated with other feature values still being used as predictors

Improve model performance for a particular slice

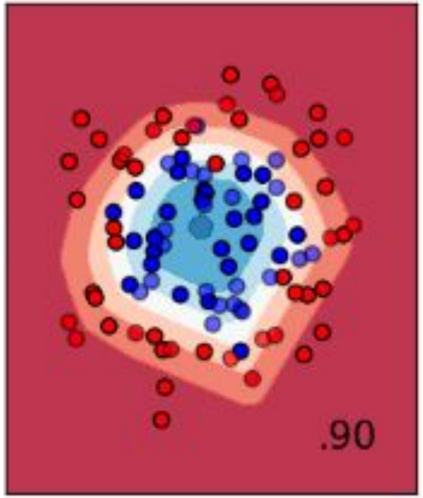
1. Try a more flexible ML model that has higher fitting capacity



Binary classification
Dataset (red v blue)



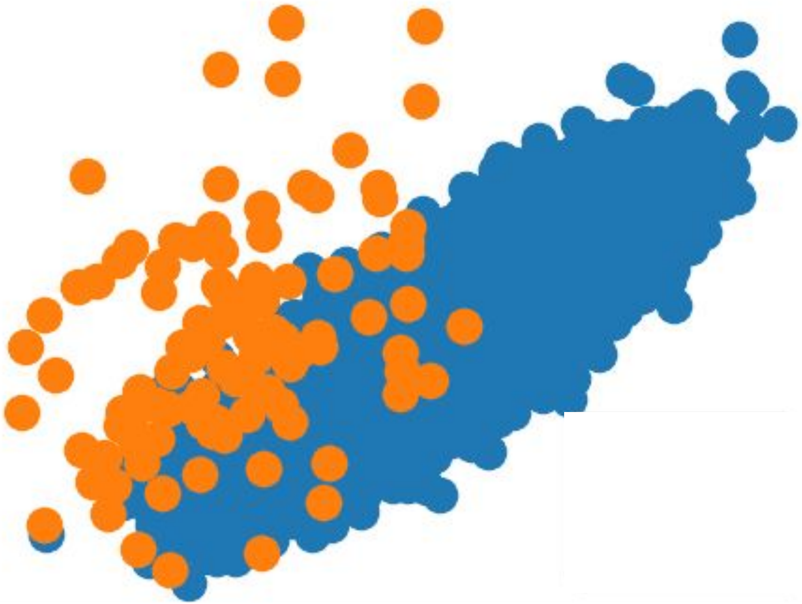
Linear Model



Neural Net Model

Improve model performance for a particular slice

2. Over-sample (up-weight) examples from minority subgroup that is receiving poor predictions



Improve model performance for a particular slice

3. Collect additional data from the subgroup with poor performance

To see if this has promise:

- Re-fit model to many versions of dataset with this subgroup down-subsampled to varying degrees

- Extrapolate the resulting model performance (overall and for subgroup) expected if you had more data from this subgroup

Improve model performance for a particular slice

4. Measure or engineer extra features that allow model to perform better for slice

Improve model performance for a particular slice

4. Measure or engineer extra features that allow model to perform better for slice

Example: Classifying if customer will purchase some product or not, based on customer & product features

- Predictions for young customers may be worse (less available history)

Improve model performance for a particular slice

4. Measure or engineer extra features that allow model to perform better for slice

Example: Classifying if customer will purchase some product or not, based on customer & product features

- Predictions for young customers may be worse (less available history)
- Could add an extra feature to the dataset such as:
  "Popularity of this product among young customers"

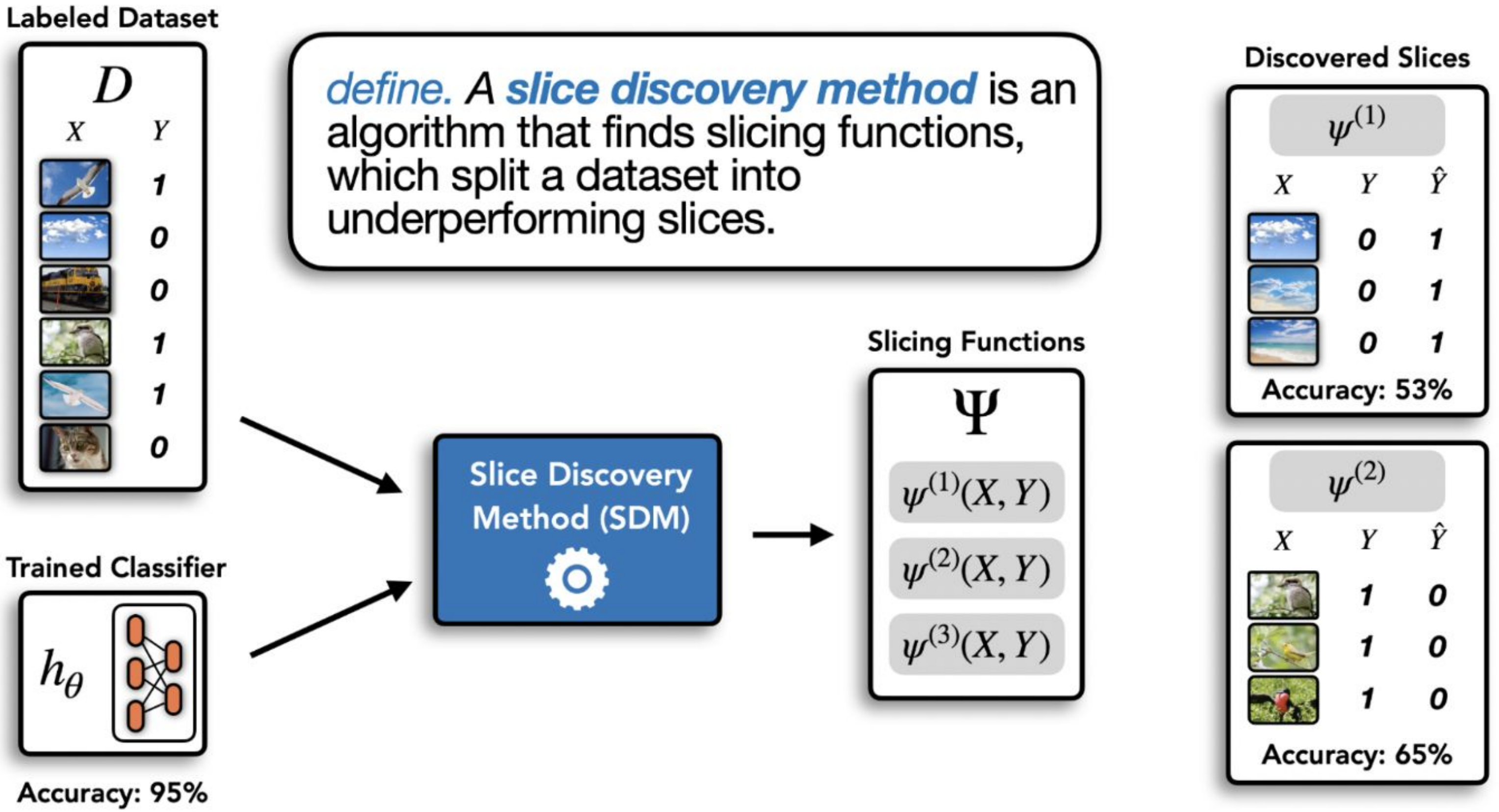
Discovering underperforming subpopulations

Discovering underperforming subpopulations

1. Sort examples in the validation data by their loss value, and look at the examples with high loss for which your model is making the worst predictions (Error Analysis)

2. Apply clustering to these examples with high loss to uncover clusters that share common themes amongst these examples

Discovering underperforming subpopulations



Why did my model get a particular prediction wrong?

1. Given label is incorrect (and our model actually made the right prediction)

   Recommended action: Correct the label

Why did my model get a particular prediction wrong?

1. Given label is incorrect (and our model actually made the right prediction)

   Recommended action: Correct the label

2. Example does not belong to any of the *K* classes
   (or is fundamentally not predictable, e.g. a blurry image)

   Recommended actions:
   - Toss this example from dataset
   - Consider adding an "Other" class if many such examples

Why did my model get a particular prediction wrong?

3. Example is an outlier
(no similar examples in the training data)

Recommended Actions:

- ??

Why did my model get a particular prediction wrong?

3. Example is an outlier
(no similar examples in the training data)

Recommended Actions:

- Toss example if similar examples would never be seen in deployment.

Why did my model get a particular prediction wrong?

3. Example is an outlier
(no similar examples in the training data)

Recommended Actions:

- Toss example if similar examples would never be seen in deployment.
- Otherwise collect additional training data that looks similar if you can.

Why did my model get a particular prediction wrong?

3. Example is an outlier
(no similar examples in the training data)

Recommended Actions:

- Toss example if similar examples would never be seen in deployment.
- Otherwise collect additional training data that looks similar if you can.
- Otherwise apply data transformation to make outliers' features more similar to other examples (eg. normalization of numeric feature, deleting a feature).

Why did my model get a particular prediction wrong?

3. Example is an outlier
(no similar examples in the training data)

Recommended Actions:

- Toss example if similar examples would never be seen in deployment.
- Otherwise collect additional training data that looks similar if you can.
- Otherwise apply data transformation to make outliers' features more similar to other examples (eg. normalization of numeric feature, deleting a feature).
- Can add synthetic data (Data Augmentation) so model becomes invariant to difference that makes this outlier stand out from other examples.

Why did my model get a particular prediction wrong?

3. Example is an outlier
(no similar examples in the training data)

Recommended action if this example is important:

- Up-weight it or duplicate it multiple times
  (perhaps with slight variants of its feature values)

Why did my model get a particular prediction wrong?

4. Type of model you're using is suboptimal for such examples

Why did my model get a particular prediction wrong?

4. Type of model you're using is suboptimal for such examples

To diagnose:

- up-weight similar examples or duplicate them many times in dataset
- retrain model
- see if new model can classify this example correctly

Why did my model get a particular prediction wrong?

4. Type of model you're using is suboptimal for such examples

To diagnose:

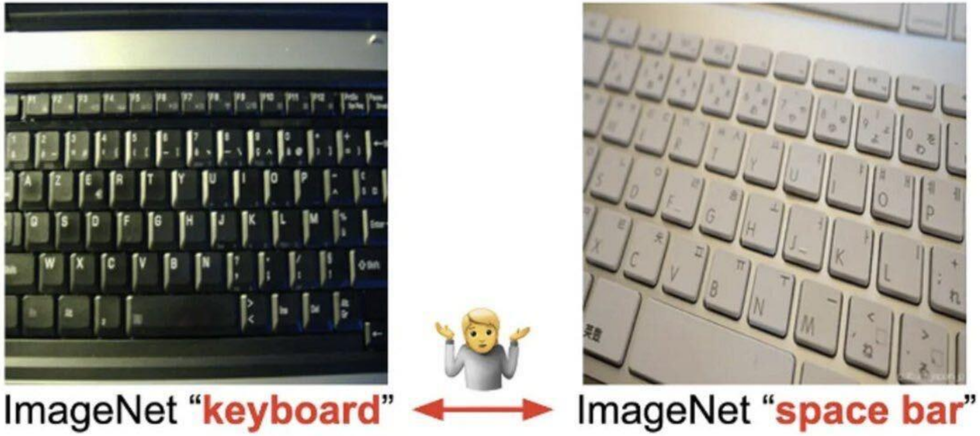
- up-weight similar examples or duplicate them many times in dataset
- retrain model
- see if new model can classify this example correctly

Recommended Actions (model-centric > data-centric in this case):

- fit different types of models
- hyperparameter tuning
- feature engineering

Why did my model get a particular prediction wrong?

5. Dataset has other examples with (nearly) identical features but different label



Why did my model get a particular prediction wrong?

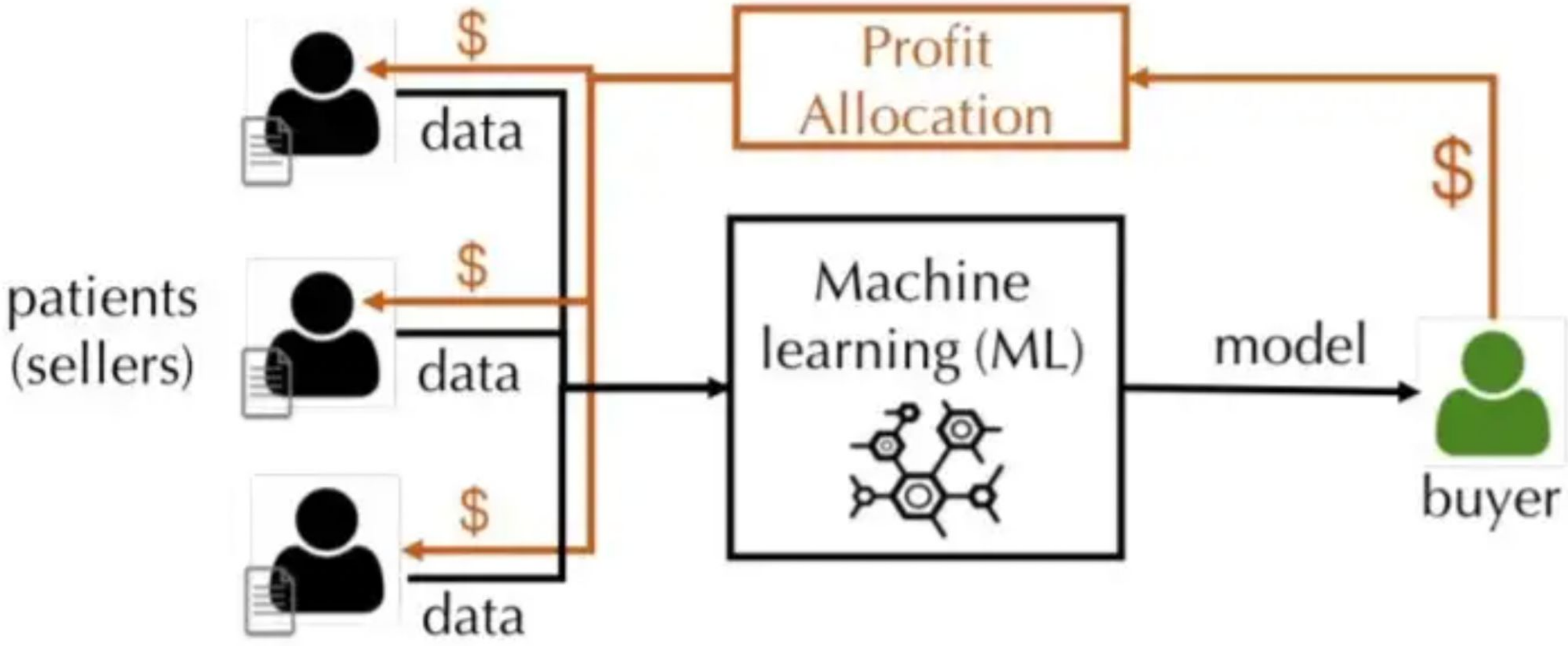
5. Dataset has other examples with (nearly) identical features but different label

Recommended Actions:

- Define classes more distinctly
- Measure extra features
  to enrich the data

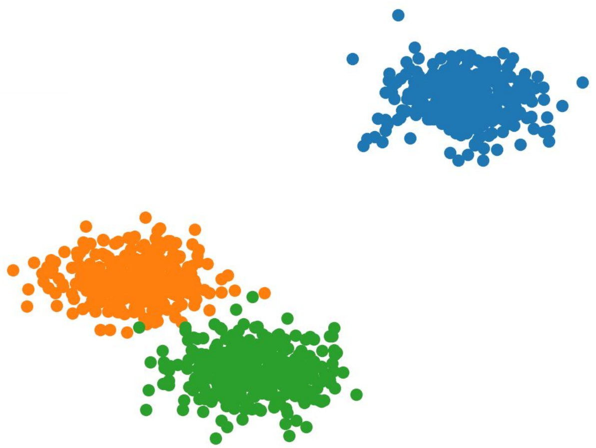
ImageNet "**keyboard**" ⟷ ImageNet "**space bar**"

Influence of individual datapoints on the model



Leave-one-out Influence

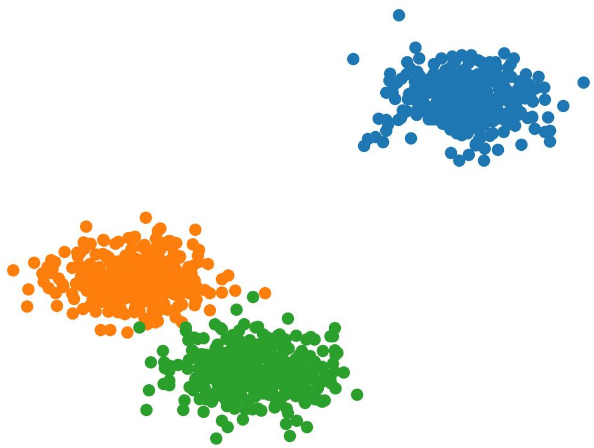
How would model change if retrained after omitting datapoint (*x*, *y*) from dataset?



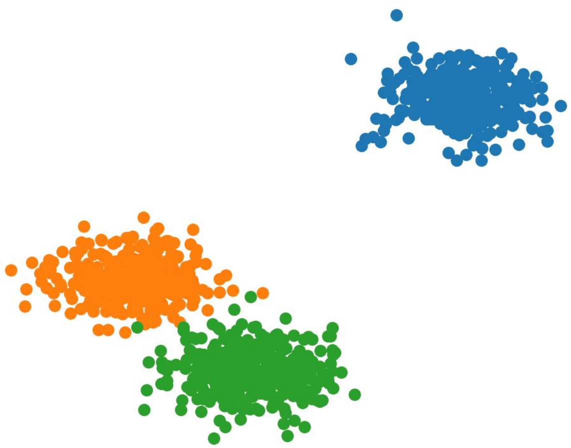
Trained Model has 98.5% validation accuracy

Leave-one-out Influence

How would model change if retrained after omitting datapoint (*x*, *y*) from dataset?



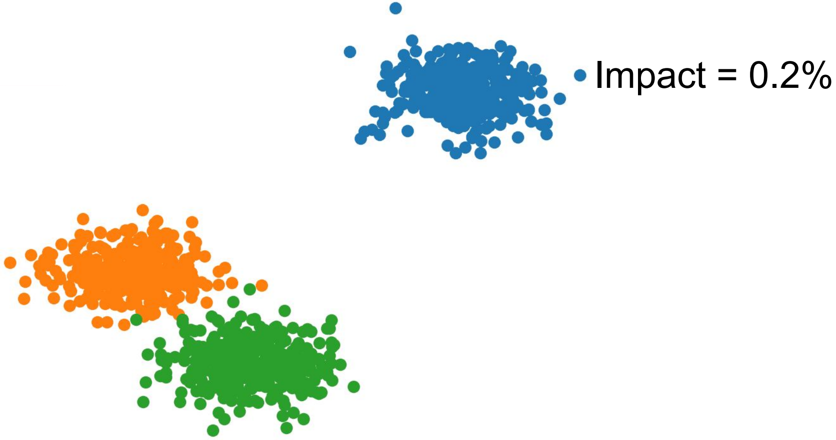
Trained Model has 98.5% validation accuracy



Trained Model has 98.3% validation accuracy

Leave-one-out Influence (LOO)

How would model change if retrained after omitting datapoint (x, y) from dataset?



Trained Model has 98.5% validation accuracy

Trained Model has 98.3% validation accuracy

Data Shapely

Compute LOO influence of datapoint (x, y) in a subset of the dataset that contains (x, y). Then average these values over **all** such possible subsets.

Data Shapely

Compute LOO influence of datapoint (*x*, *y*) in a subset of the dataset that contains (*x*, *y*). Then average these values over **all** such possible subsets.

Example: Suppose there are two identical datapoints in dataset and omitting both severely harms model accuracy but omitting one does not.

Data Shapely

Compute LOO influence of datapoint (x, y) in a subset of the dataset that contains (x, y). Then average these values over **all** such possible subsets.

Example: Suppose there are two identical datapoints in dataset and omitting both severely harms model accuracy but omitting one does not.

LOO Influence: ??

Data Shapely: ??

Data Shapely

Compute LOO influence of datapoint (x, y) in a subset of the dataset that contains (x, y). Then average these values over **all** such possible subsets.

Example: Suppose there are two identical datapoints in dataset and omitting both severely harms model accuracy but omitting one does not.

LOO Influence: neither datapoint is too influential

Data Shapely: both are fairly influential

Approximating Influence via Monte Carlo

1. Subsample T different data subsets $\mathcal{D}_t$ from the original training dataset (without replacement).
2. Train a separate copy of your model M_t on each subset $\mathcal{D}_t$ and report its accuracy on held-out validation data: a_t.

Approximating Influence via Monte Carlo

1. Subsample T different data subsets $\mathcal{D}_t$ from the original training dataset (without replacement).
2. Train a separate copy of your model M_t on each subset $\mathcal{D}_t$ and report its accuracy on held-out validation data: a_t.
3. To assess the value of a datapoint (x_i, y_i), compare the average accuracy of models for those subsets that contained (x_i, y_i) vs. those that did not. More formally:

$$I(x_i) = \frac{1}{|D_{\text{in}}|} \sum_{t \in D_{\text{in}}} a_t - \frac{1}{|D_{\text{out}}|} \sum_{t \in D_{\text{out}}} a_t$$

where $D_{\text{in}} = \{t : (x_i, y_i) \in \mathcal{D}_t\},\ D_{\text{out}} = \{t : (x_i, y_i) \notin \mathcal{D}_t\}$.

Approximating Influence via Monte Carlo

1. Subsample T different data subsets $\mathcal{D}_t$ from the original training dataset (without replacement).
2. Train a separate copy of your model M_t on each subset $\mathcal{D}_t$ and report its accuracy on held-out validation data: a_t.
3. To assess the value of a datapoint (x_i, y_i), compare the average accuracy of models for those subsets that contained (x_i, y_i) vs. those that did not. More formally:

$$I(x_i) = \frac{1}{|D_{\text{in}}|} \sum_{t \in D_{\text{in}}} a_t - \frac{1}{|D_{\text{out}}|} \sum_{t \in D_{\text{out}}} a_t$$

where $D_{\text{in}} = \{t : (x_i, y_i) \in \mathcal{D}_t\}, \ D_{\text{out}} = \{t : (x_i, y_i) \notin \mathcal{D}_t\}$.

Accuracy here could be replaced by any other loss of interest.

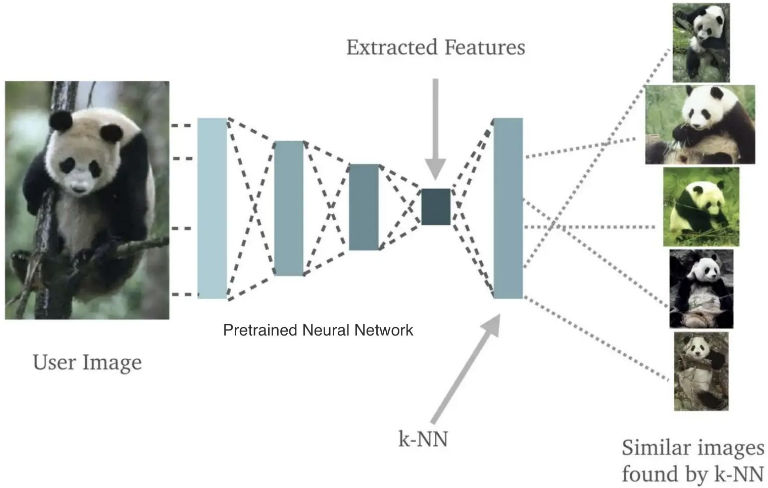
Closed-form Computation of Influence

Can be done in regression (mean-squared-error loss) with linear regression model

- Called *Cook's Distance*

Can be done for K-Nearest Neighbors classifier

- in $O(n \log n)$ time



Reviewing Influential Samples

- Influence reveals which data points have greatest impact on the model.

- Correcting a mislabeled datapoint with high influence can boost model accuracy more than correcting a mislabeled datapoint with low influence

Reviewing Influential Samples

- Influence reveals which datapoints have greatest impact on the model.
- Correcting a mislabeled datapoint with high influence can boost model accuracy more than correcting a mislabeled datapoint with low influence
- Finding mislabeled data may be hard sorting only by influence instead of using confident learning as well