



# Navigating Data Errors in Machine Learning Pipelines: Identify, Debug, and Learn

---

Bojan Karlaš (Harvard University), Babak Salimi (UC San Diego), Sebastian Schelter (BIFOLD & TU Berlin)



[navigating-data-errors.github.io](https://navigating-data-errors.github.io)

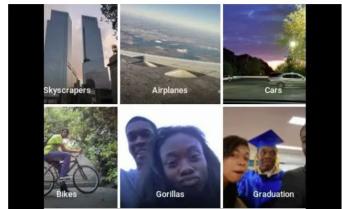


# Background: ML apps often behave in unintended ways

## Wrong

Google apologises for Photos app's racist blunder

© 1 July 2015



Source: BBC

## Biased

MIT  
Technology  
Review

Featured Topics Newsletters Events Audio

SIGN IN

SUBSCRIBE

SILICON VALLEY

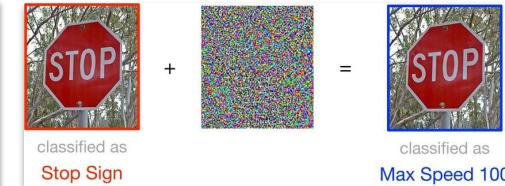
## Amazon ditched AI recruitment software because it was biased against women

By Erin Winick

October 10, 2018

Source: MIT Technology Review

## Unstable



Source: Xiong et al. ACM Comput. Surv. 2023.



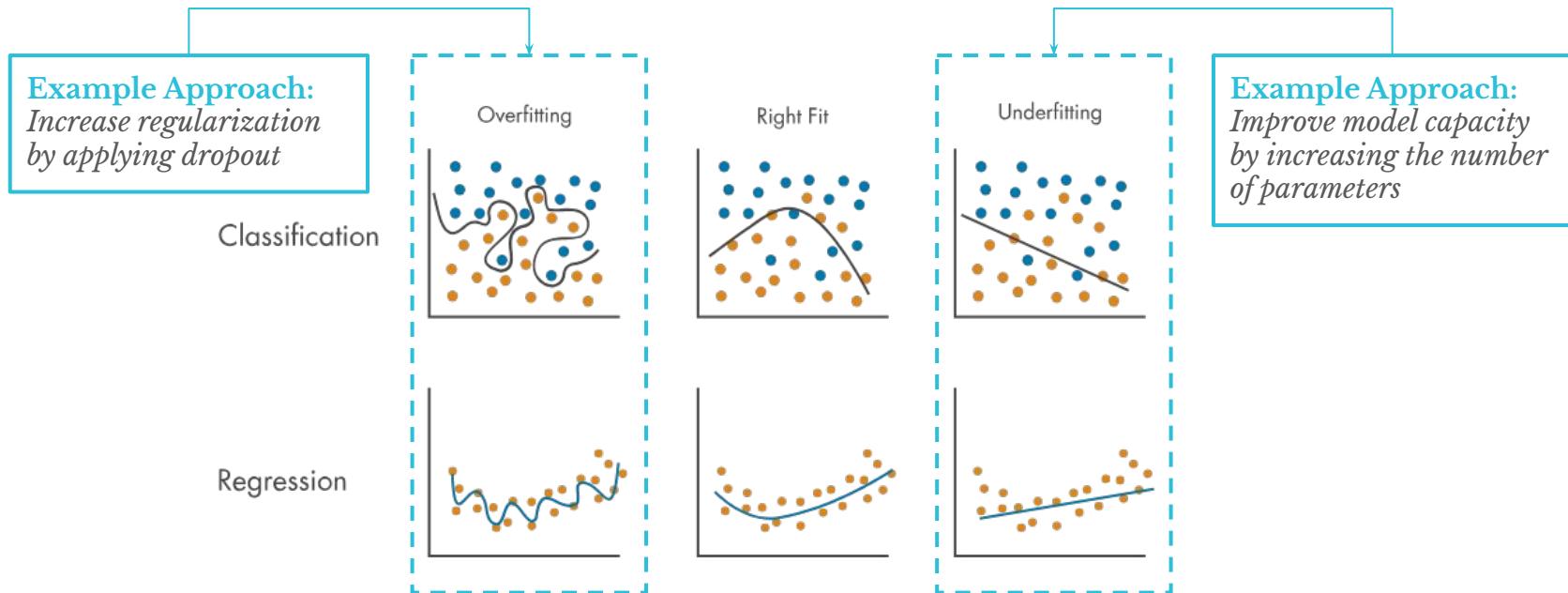
Tesla Autopilot feature was involved in 13 fatal crashes, US regulator says

Federal transportation agency finds Tesla's claims about feature don't match their findings and opens second investigation



Source: The Guardian

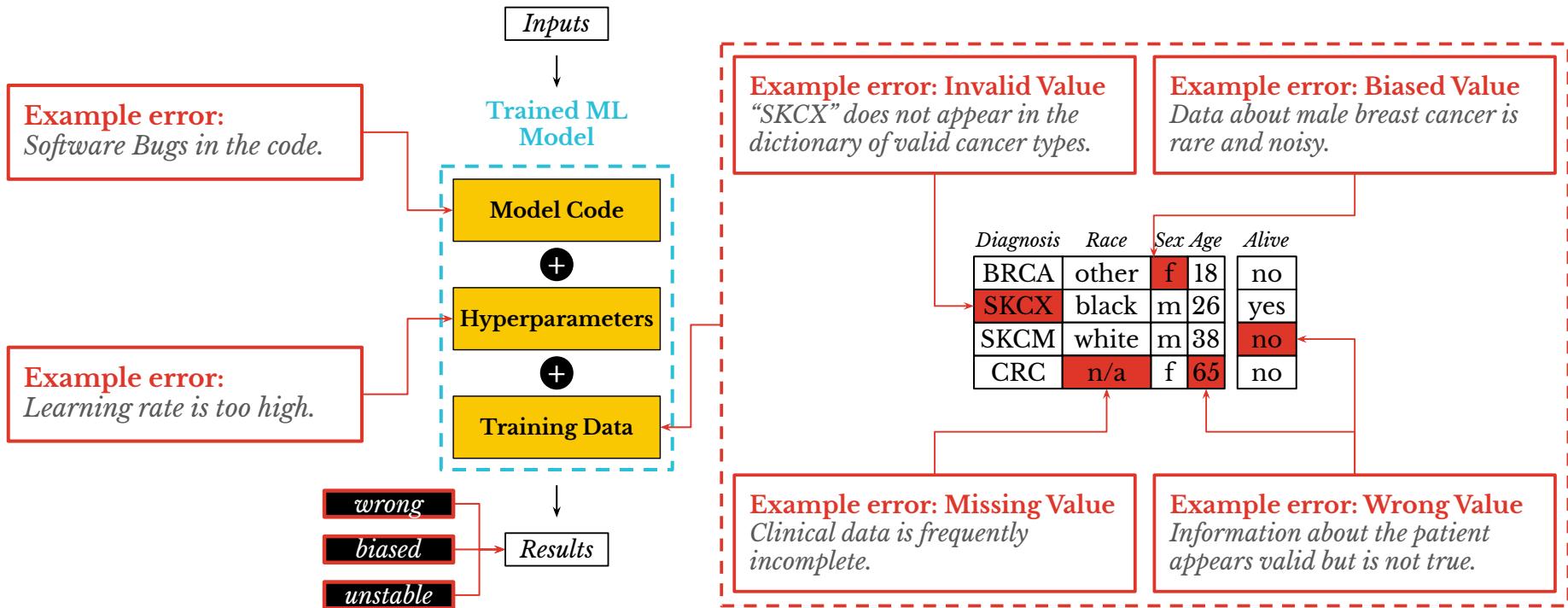
# Primary approach: Focus on improving the model



Source: MathWorks

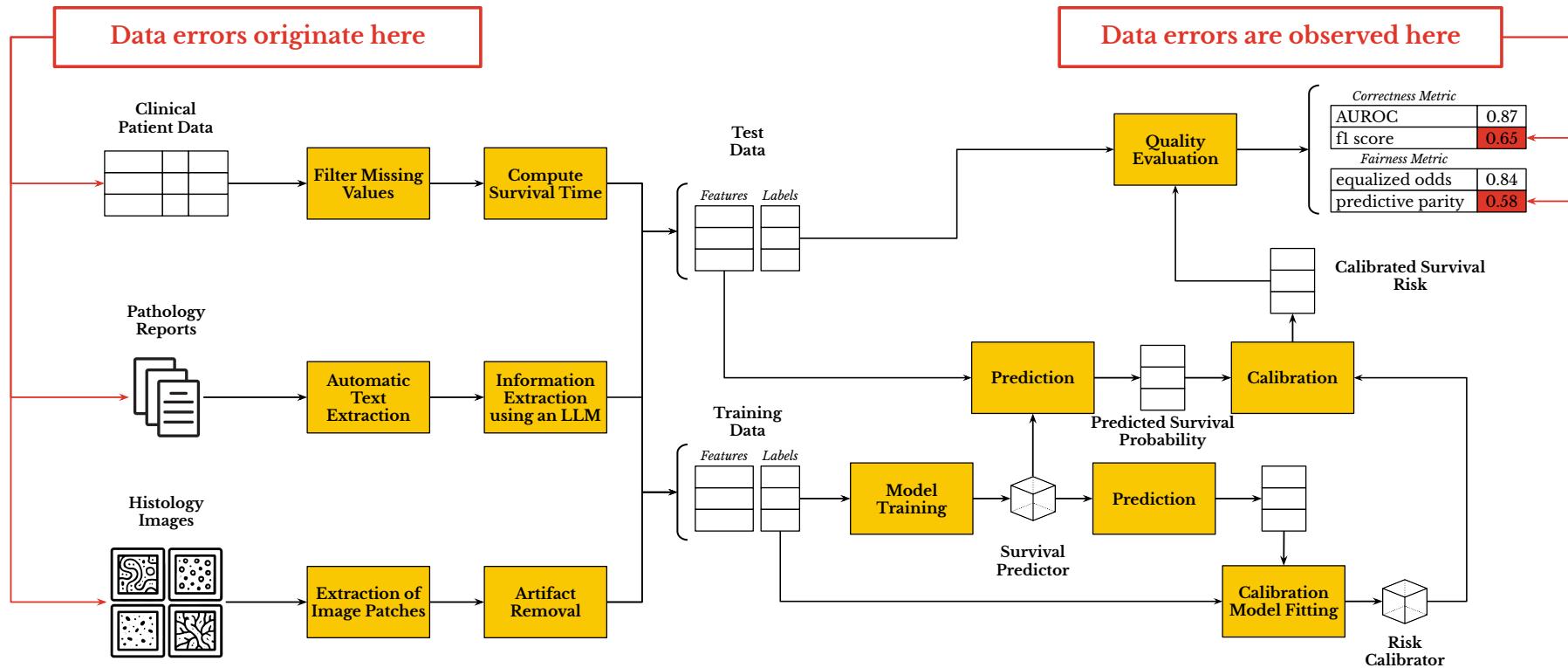
**Problem:** This is only one piece of the puzzle!

# Observation 1: Data is a crucial piece of the puzzle



**Challenge 1:** Can we identify the most important data errors?

# Observation 2: ML apps are built by complex pipelines



**Challenge 2:** Can we trace data errors as they pass through the pipeline?

# Observation 3: Not all data errors are meant to be fixed

For each data error, we can choose to perform one of the following actions:

**Discard**



*Remove the faulty data from the training set.*

**Repair**



*Perform manual quality control which might include repeating the data acquisition process.*

**Ignore**



*Let the faulty data remain in the training set.*

Benefits:

Easy to Perform

Data Quality Improves

No Labor Required

Shortcomings:

Loss of Useful Data

Often Labor-intensive

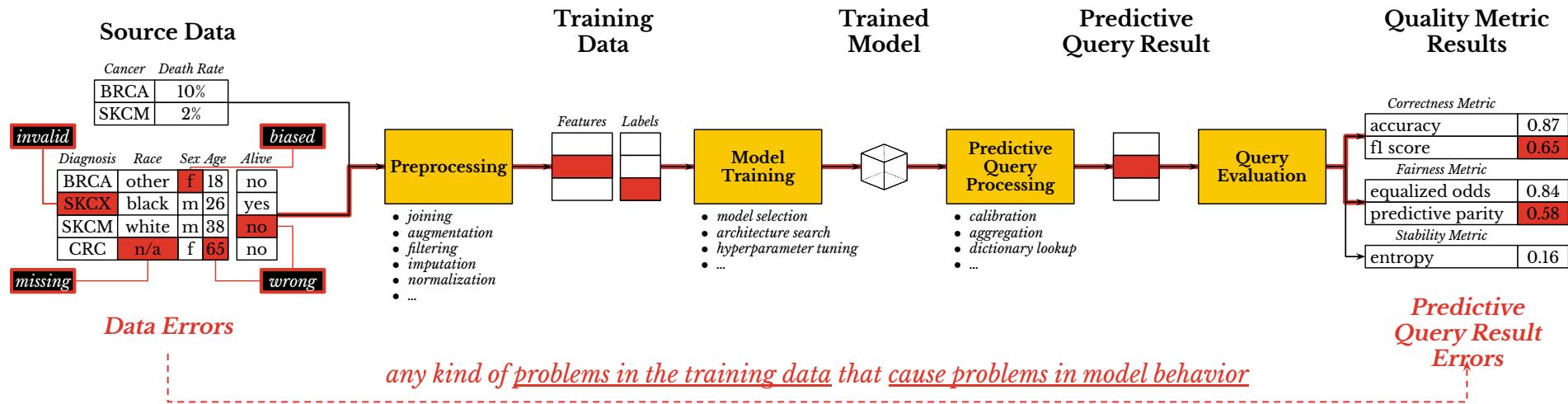
Risk Hurting Model Quality

Optimal trade-off:

Discard or Repair the Portion of Data that will Bring the Highest Model Quality Increase

**Challenge 3:** Can we ensure reliable model performance after (partial) data repairs?

# Tutorial Overview: Data Errors in ML pipelines



## Part I: Data Importance for Data Error Detection

What are good approaches for identifying data errors?

## Part II: Data Debugging in ML Pipelines

What are practical challenges when debugging complex ML pipelines?

## Part III: Learning from Uncertain and Incomplete Data

When we cannot repair all errors, can we still have reliable models?

# Opportunities for the Data Management Community

- (1) Data quality is an established discipline in data management, but most practitioners still rely on **manual effort**.
- (2) ML pipelines are data processing pipelines. Models are learned data transformation operators. Many systems have been developed, but most practitioners still rely on **rudimentary scripts for crunching data**.
- (3) Many promising methods for handling data errors suffer from **scalability issues**.

**Main Goal:** *Present the current state of the art and inspire novel research.*

# Part I: Data Importance for Data Error Detection

---

Bojan Karlaš



- 1) Introducing the Concept of Data Importance**
- 2) Examples of Data Attribution Functions**
- 3) Case Study of Shapley Value as a Measure of Importance**
- 4) Applications of Data Importance**

# How can we identify data errors?

Trivial

**Solution approach:**

Apply a rule-based validation function that performs a dictionary lookup.

invalid

Diagnosis	Race	Sex	Age	Alive
BRCA	other	f	18	no
SKCX	black	m	26	yes
SKCM	white	m	38	no
CRC	n/a	f	65	no

missing

biased

wrong

Not So Trivial

**Solution approach:**

Check if the value is marked as missing.

**Solution approach:**

Measure the impact of the value on model quality.

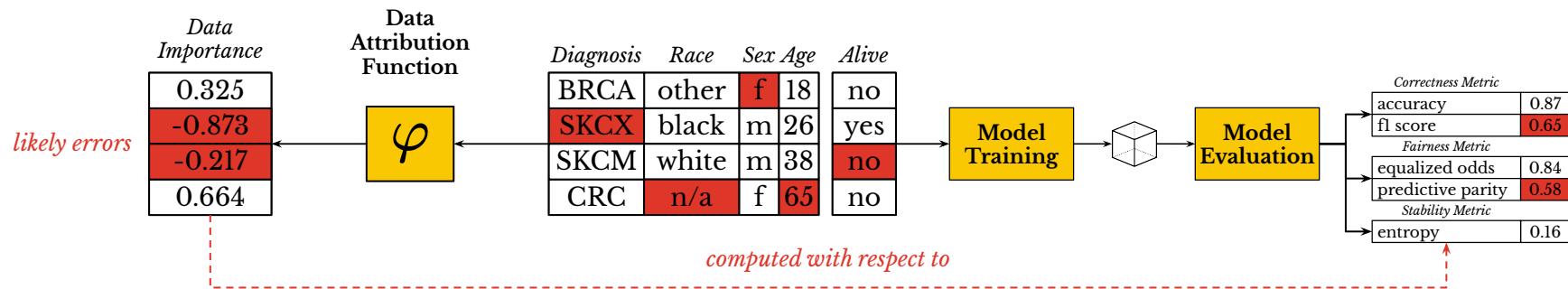
**How do we measure this?**

That is the main topic of this part of the tutorial.

**Recall:** Data errors are any kind of problem in the training data that cause problems in model behavior.

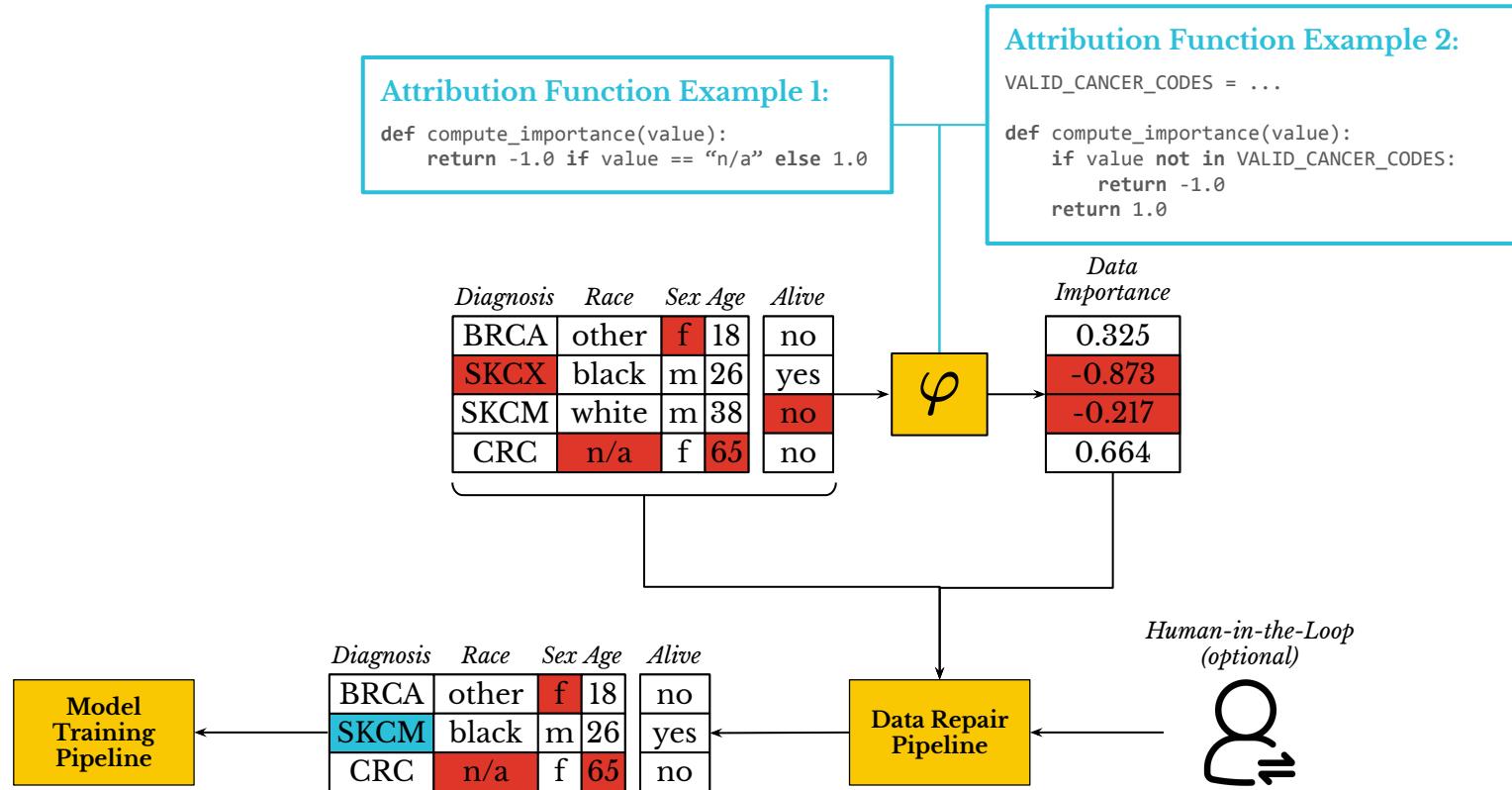
**Challenge:** Can we define a unified way to think about identifying data errors?

# We can define a data attribution function



**Recall:** Data errors are any kind of problem in the training data that cause problems in model behavior.

# How do we use importance to detect data errors?



# What makes a good attribution function?

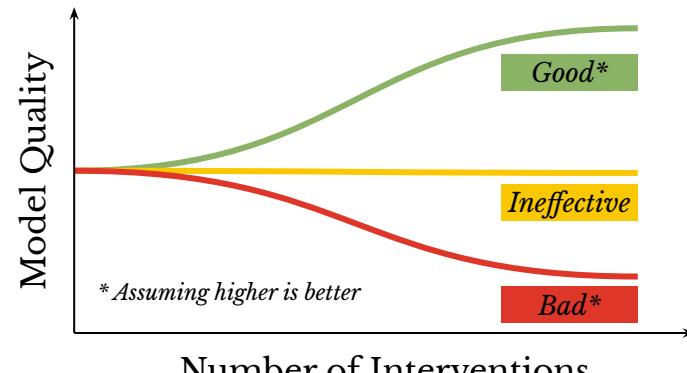
## Design Consideration 1

Which model quality metric do we care about improving?

Correctness Metric
accuracy
f1 score
Fairness Metric
equalized odds
predictive parity
Stability Metric
entropy

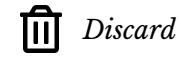
### Recall:

Data errors are any kind of problem in the training data that cause problems in model behavior.



## Design Consideration 2

What kind of intervention do we intend to apply?



Discard



Repair



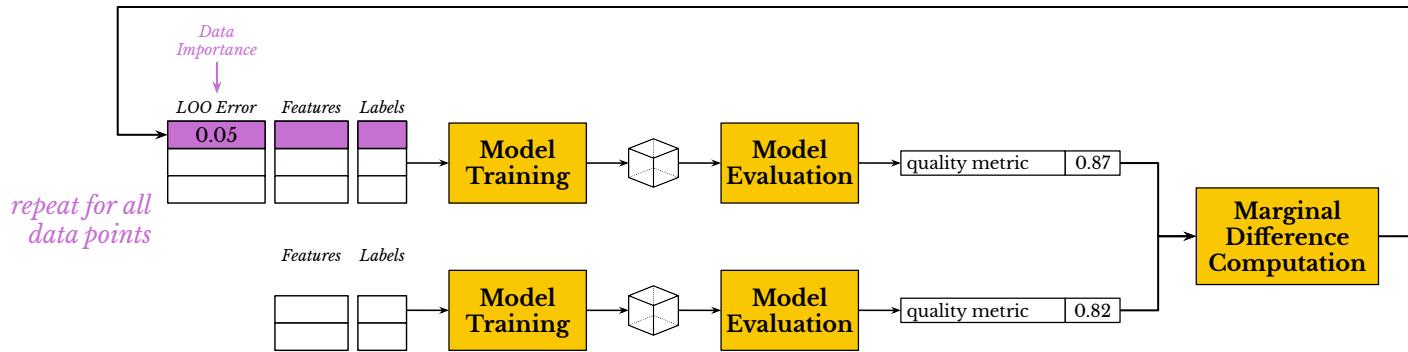
Something Else

**Challenge:** How do we define an effective attribution function?

- 1) Introducing the Concept of Data Importance
- 2) Examples of Data Attribution Functions**
- 3) Case Study of Shapley Value as a Measure of Importance
- 4) Applications of Data Importance

# Leave-one-Out Error

[Approach: Marginal Contribution]



## Insights:

- Removing important data points affects model quality.

## Approach:

- Remove a data point from the training set, train and evaluate the model again
- Interpret the difference in model quality as data importance.

## Benefits:

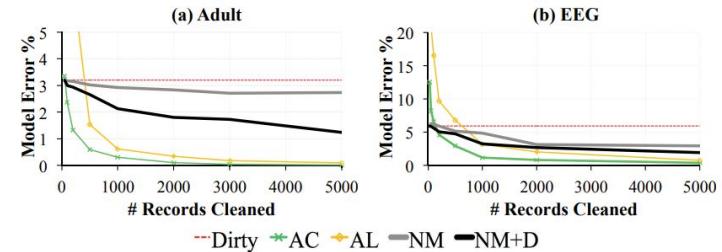
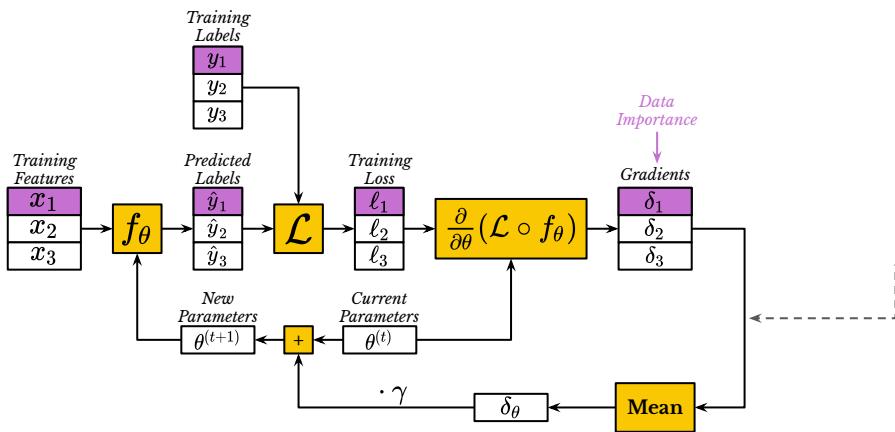
- Very simple to implement.

## Shortcomings:

- Requires re-training the model once for each data point.
- Treats data points independently.

# Error Gradient

[Approach: Gradient]



## Insights:

- Data points vary in their contribution to the gradients that update the model.

## Approach:

- Importance is proportional to the magnitude of the gradient.

## Benefits:

- Simple to compute.

## Shortcomings:

- Treats data points independently.

### ActiveClean: Interactive Data Cleaning For Statistical Modeling

Sanjay Krishnan, Jianan Wang\*, Eugene Wu\*, Michael J. Franklin, Ken Goldberg  
UC Berkeley, Simon Fraser University, \*Columbia University  
sanjaykrishnan, franklin, goldberg@berkeley.edu, jwang@sfu.ca, www.cs.columbia.edu

#### ABSTRACT

Analyzing often-clean data frequently cleaning some data, executing the analysis, and then cleaning more data based on the results. This is a common pattern in machine learning and statistical modeling, which is an increasingly popular form of data science. However, this pattern can lead to poor performance if one uses and then reuses the same data to both clean it and then train a model. We propose ActiveClean, an interactive system that can help to effect the results. We evaluate ActiveClean on five real-world datasets. Our results show that ActiveClean can significantly reduce costs with both real and synthetic errors. The results also show that our proposed system can significantly outperform ActiveLearn, a state-of-the-art system for data cleaning. Furthermore, we find that ActiveClean is significantly more accurate than ActiveLearn.

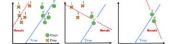


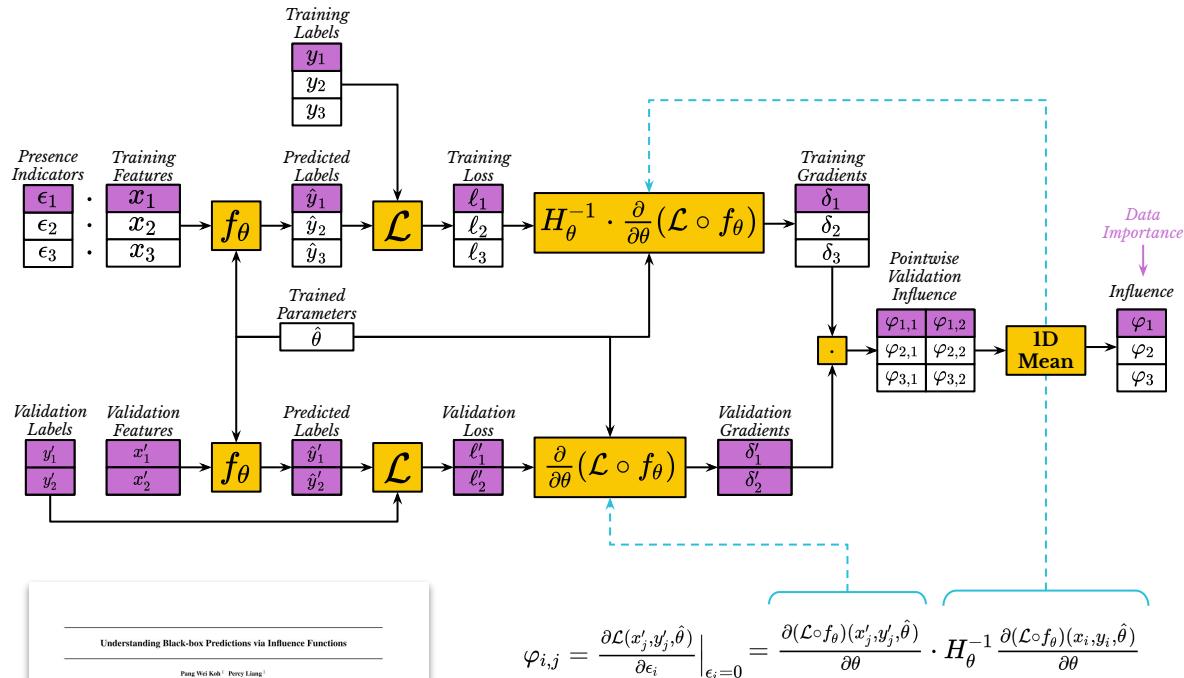
Figure 1(a) Symmetric corruption in one variable can lead to a local minimum. The plot shows a surface with two peaks and a valley. A green dot represents the initial state, and a red dot represents the final state after cleaning. The path from green to red goes down the valley, getting stuck in a local minimum.

### Krishnan VLDB'16

Krishnan, Sanjay, et al. "Activeclean: Interactive data cleaning for statistical modeling." Proceedings of the VLDB Endowment 9.12 (2016): 948-959. [\[Paper\]](#) [\[Website\]](#)

# Influence Function

[Approach: Marginal Contribution, Gradient]



## Understanding Black-box Predictions via Influence Functions

Abstract

How can we explain the predictions of a black-box model? In this paper, we use influence functions — a classic technique from statistics — to trace a model’s prediction through the training algorithm back to its training data, thereby identifying training data most responsible for a given prediction. To set up influence functions for black-box models, we first introduce presence indicator variables for each data point. We then develop a simple, efficient implementation that requires only gradient calculations and Hessian-vector products. We show that even on complex neural networks, influence functions are more effective than state-of-the-art methods for explaining black-box predictions. We also show that influence functions can still provide valuable information. On the other hand, we demonstrate that influence functions are not always able to understand model behavior, debugging models, detecting adversarial attacks, and so on.

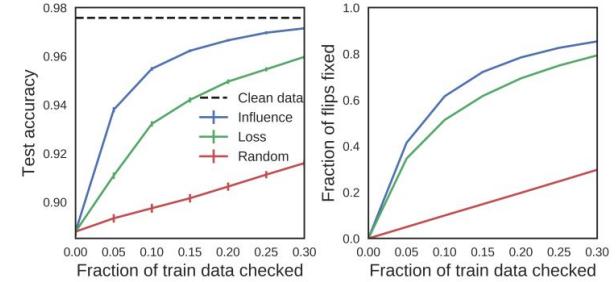
paper [Koh et al., 2016] or by perturbing the test point to see how the prediction changes [Shrikumar et al., 2017; Li et al., 2016; Datta et al., 2016; Adeli et al., 2017]. These approaches are useful for understanding a single test point, but how can we explain where the model came from?

In this paper, we take this approach to trace the model’s prediction through the training algorithm back to the training data, where the model parameters ultimately determine the prediction. Specifically, given a test point and a prediction, we ask the counterfactual: what would happen if one of the values of this training point were changed slightly?

Answering this question by perturbing the data and retraining the model is a standard approach to solving this problem, we use influence functions, a classic technique from robust statistics [Cook & Weisberg, 1994] and machine learning [Hastie et al., 2009]. As we highlight a testing point by an infinitesimal amount, this allows us to approximate the change in the prediction due to this small change in the training data.

## [Koh ICML ‘17]

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." International conference on machine learning. PMLR, 2017. [\[Paper\]](#) [\[Code\]](#)



## Insights:

- The marginal contribution of a single data point can be approximated with gradients.

## Approach:

- Introduce presence indicator variables  $\epsilon$  for each data point and compute the gradient w.r.t.  $\epsilon$ .

## Benefits:

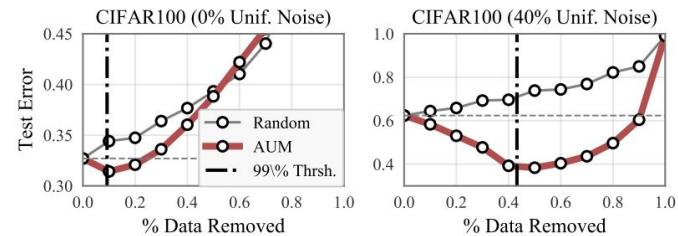
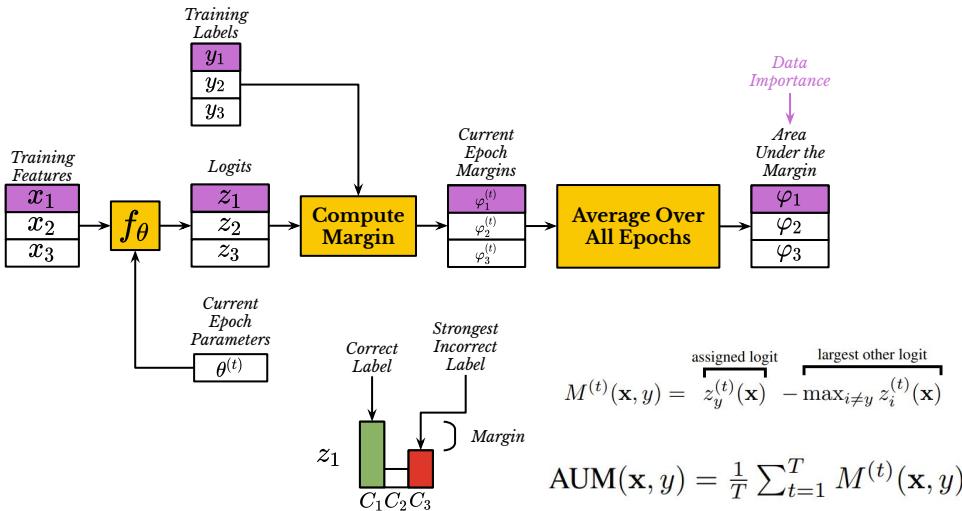
- Easily applicable to arbitrarily complex (twice) differentiable machine learning models.

## Shortcomings:

- Treats data points independently.

# Area Under the Margin

[Approach: Uncertainty Analysis]



## Insights:

- If similar samples have the same label, the model will learn to activate only the correct logit.
- In the presence of mislabeled samples, the model will learn to activate alternative logits.

## Approach:

- The importance of a data point is proportional to its margin averaged across all training epochs.

## Benefits:

- Very simple to implement in a wide array of models.
- Does not rely on a separate clean dataset.

## Shortcomings:

- Focuses only on label noise.

### Identifying Mislabeled Data using the Area Under the Margin Ranking

Geoff Pleiss<sup>1</sup>  
Columbia University  
gpp12@columbia.edu

Tianqi Zhang<sup>2</sup>  
Stanford University  
tianqi.zhang@stanford.edu

Ethan Fetaya<sup>3</sup>  
ASAPP  
ethanfetaya@asapp.com

#### Abstract

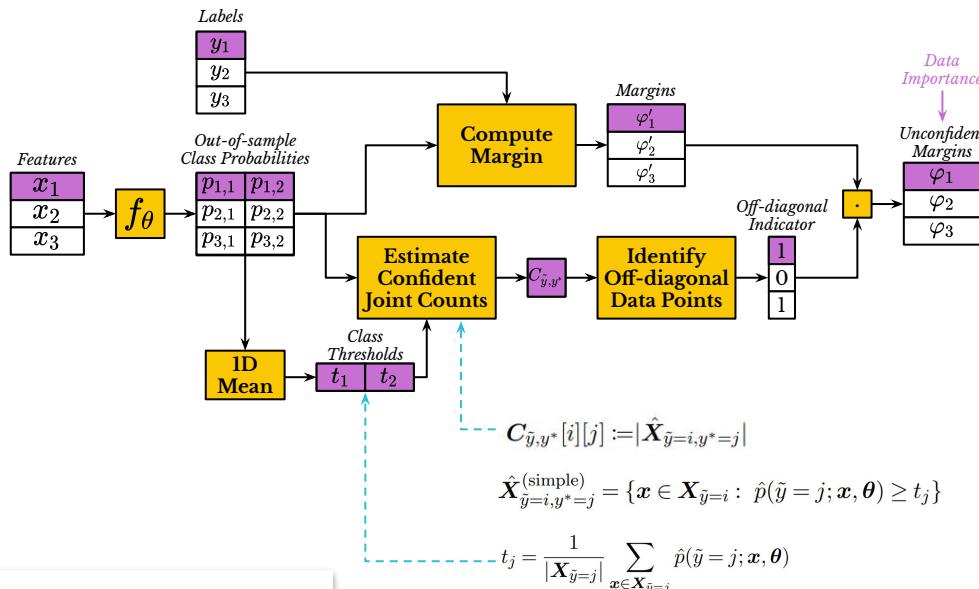
No all data is perfect training or helps generalization; some can be overly ambiguous or completely mislabeled. This paper introduces a new method to identify such data points. Our approach is based on the Area Under the Margin (AUM) statistic, which measures the importance of a data point. We propose a simple procedure—adding a new class populated with personally mislabeled samples—so that the AUM statistic can be computed. This approach consistently improves upon prior work on synthetic and real-world datasets.

### [Pleiss NeurIPS '20]

Pleiss, Geoff, et al. "Identifying mislabeled data using the area under the margin ranking." Advances in Neural Information Processing Systems 33 (2020): 17044-17056. [\[Paper\]](#) [\[Blog\]](#) [\[Code\]](#)

# Unconfident Margins

[Approach: Uncertainty Analysis]



Journal of Artificial Intelligence Research 70 (2021) 1373-1411  
Submitted 06/2020; published 01/2021

## Confident Learning: Estimating Uncertainty in Dataset Labels

Curtis G. Northcutt  
Massachusetts Institute of Technology  
Department of ECECS, Cambridge, MA, USA

Lu Jiang  
Google Pixel, Mountain View, CA, USA

Isaac E. Chuang  
Massachusetts Institute of Technology  
Department of ECECS, Department of Physics, Cambridge, MA, USA

CONF@MIT.EDU

LJUAN@GOOGLE.COM

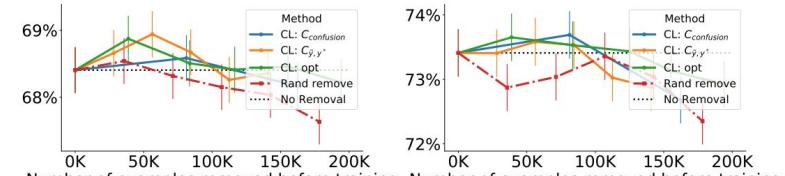
IUCHANG@MIT.EDU

### Abstract

Learning rules in the context of datasets of softmaxes typically focus on model predictions, not label quality. Confident learning (CL) is an alternative approach which focuses on the uncertainty of labels. CL is a nonparametric method, and therefore, based on the principles of pruning noisy data, it works with probabilistic thresholds to estimate noise, and ranking examples to train with confidence. Whereas numerous studies have focused on the use of CL for classification, this paper is the first to do so in the assumption of a class-conditional noise process to directly estimate the joint distribution between labels and features. This allows CL to estimate the joint distribution between labels and features, and therefore, to estimate the uncertainty of labels.

### [Northcutt JAIR '21]

Northcutt, Curtis, Lu Jiang, and Isaac Chuang. "Confident learning: Estimating uncertainty in dataset labels." Journal of Artificial Intelligence Research 70 (2021): 1373-1411. [\[Paper\]](#) [\[Blog\]](#) [\[Code\]](#)



## Insights:

- Given a data point, if a model assigns a higher than average probability to some specific class, it is likely because most similar data points have the same class label. This is likely to be the true label of that data point.

## Approach:

- Identify likely mislabeled data points and assign negative importance using the margin. Remaining data points get zero importance.

## Benefits:

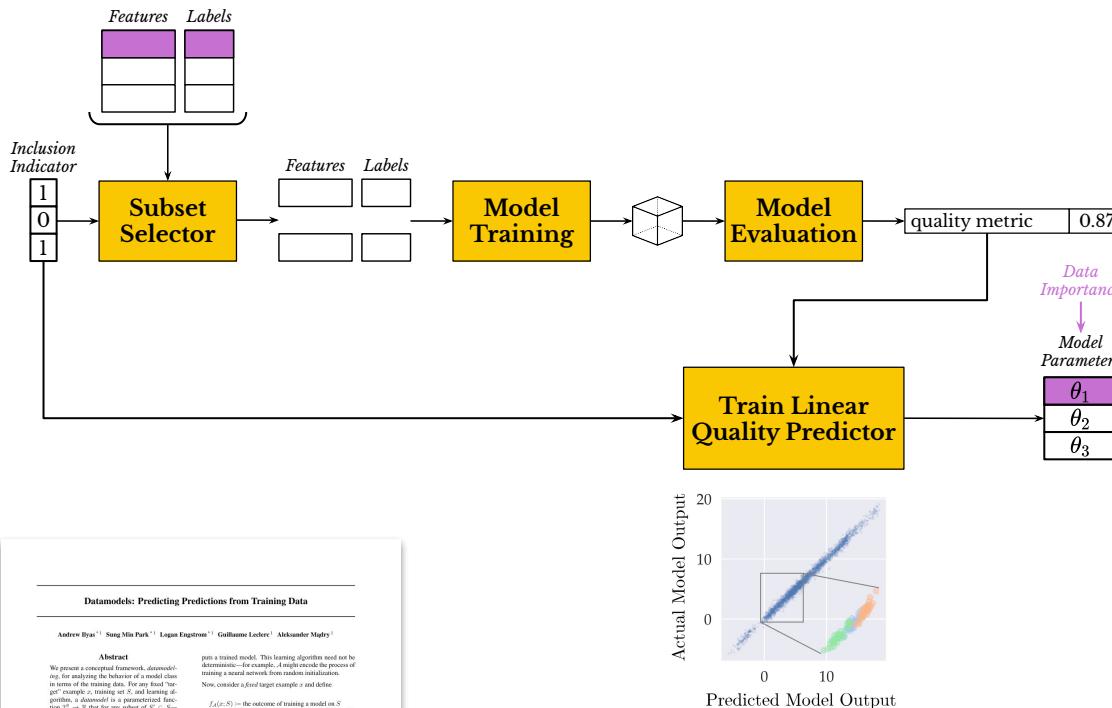
- Very simple to implement in a wide array of models.
- Does not rely on a separate clean dataset.

## Shortcomings:

- Focuses only on label noise.
- Relies on having an adequately powerful model.

# Model Training Outcome

[Approach: Surrogate Data Model]



[Ilyas ICML '22]

Ilyas, Andrew, et al. "Datamodels: Predicting Predictions from Training Data." Proceedings of the 39th International Conference on Machine Learning. 2022. [\[Paper\]](#) [\[Blog\]](#) [\[Code\]](#)

## Insights:

- A linear model can be good at predicting the quality of a model trained on an arbitrary subset of the training data and tested on a single test example.

## Approach:

- Train a linear quality predictor and interpret its parameters as data importance.

## Benefits:

- Conceptually simple yet powerful framework for analyzing datasets.

## Shortcomings:

- The original method requires retraining the model many times.

- 1) Introducing the Concept of Data Importance
- 2) Examples of Data Attribution Functions
- 3) Case Study of Shapley Value as a Measure of Importance**
- 4) Applications of Data Importance

# Improving Upon the Marginal Contribution Methods

## Recall

Marginal contribution methods treat data points independently, ignoring any interactions that might exist.

## Consequence

Let there be a data point that has high importance. If we make two copies of that data point, their individual marginal contribution to the dataset as a whole will be zero.

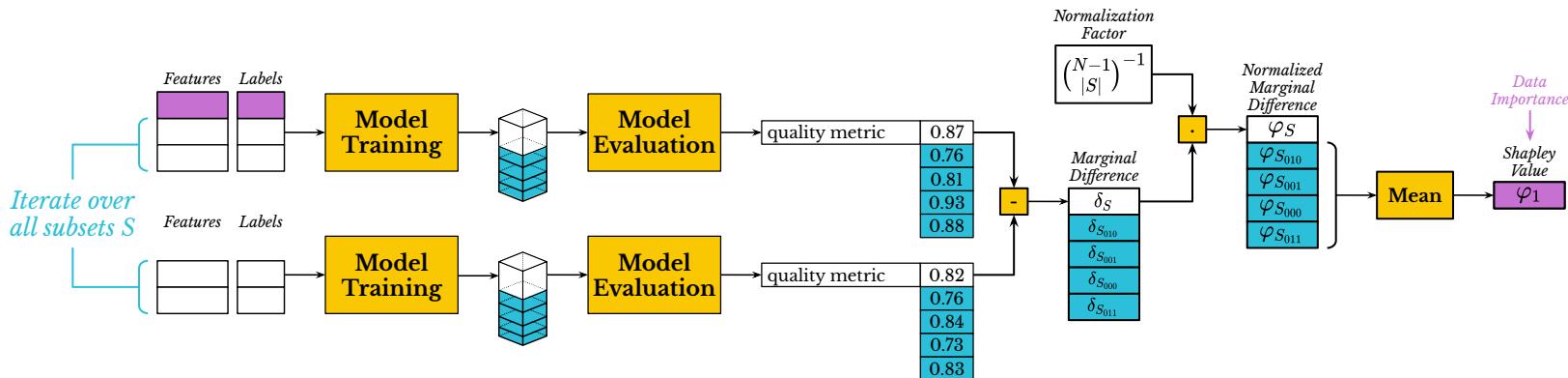
## Shapley value

A standard method from game theory for distributing surplus among a coalition of players.

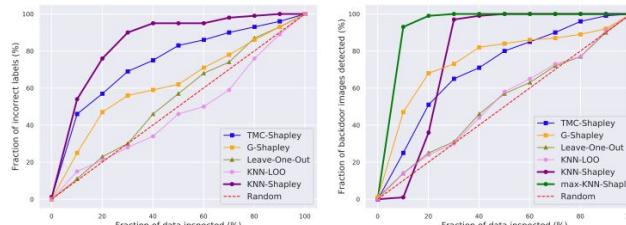
$$\varphi_i = \frac{1}{N} \sum_{S \subseteq X \setminus \{i\}} \binom{N-1}{|S|}^{-1} (u(S \cup \{i\}) - u(S))$$

## Approach

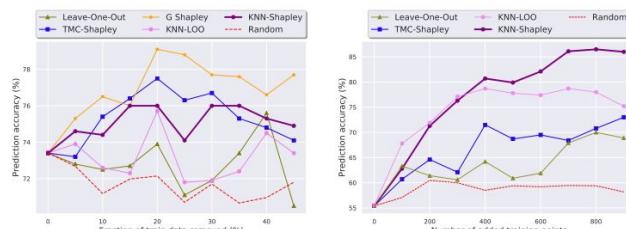
We should measure marginal contribution over all subsets.



# Effectiveness at Data Debugging



(a) Noisy labels detection



(c) Data summarization

(d) Data acquisition

Figure 2: The experiment result of (a) noisy label detection on fashion-MNIST dataset; (b) instance-based watermark removal on MNIST dataset; (c) data summarization on UCI Adult Census dataset [15]; (d) data acquisition on MNIST dataset with injected noise. In (a)-(b) the “random” line shows the results of random guess; while in (c)-(d), the “random” line corresponds to the empirical results of the random baseline introduced in Section 4.1.

Table 2: Domain adaptation between MNIST and USPS.

Method	MNIST → USPS	USPS → MNIST
	→	→
KNN-Shapley	<b>31.70% → 47.00%</b>	<b>23.35% → 29.80%</b>
KNN-LOO	31.70% → 37.40%	23.35% → 24.50%
TMC-Shapley	31.70% → 44.90%	23.35% → 29.55%
LOO	31.70% → 29.40%	23.35% → 23.53%

This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.  
Except for the watermark, it is identical to the accepted version.  
The final published version of the proceedings is available on IEEE Xplore.



## Scalability vs. Utility: Do We Have to Sacrifice One for the Other in Data Importance Quantification?

Ruoxi Jia<sup>1</sup> Fan Wu<sup>2\*</sup> Xudong Sun<sup>3</sup> Jiacen Xu<sup>4</sup> David Dao<sup>5</sup>  
<sup>1</sup>Virginia Tech      <sup>2</sup>UIC      <sup>3</sup>Shantou Jiaotong University      <sup>4</sup>UC Irvine      <sup>5</sup>ETH Zurich  
<sup>1</sup>ruoxi.jia.cs@vt.edu      <sup>2</sup>fanwu@uic.edu      <sup>3</sup>xudong.su@stu.edu.cn      <sup>4</sup>jxu@math.uci.edu      <sup>5</sup>daod@ethz.ch

### Abstract

### I. Introduction

Quantifying the importance of each training point is a fundamental problem in machine learning and the related research areas have been progressing to make a range of data workflow tools such as data summarization, data acquisition, and data debugging. The leave-one-out error of each training point is related to its importance quantification, which is often measured by a key value, as it defines a unique value distribution when

[Jia CVPR '21]

Jia, Ruoxi, et al. "Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification?" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. [\[Paper\]](#) [\[Code\]](#)

# Benefits and Challenges

## Beneficial Properties of the Shapley Value

### Symmetry

If two data points have the same contribution to every subset, their value should be the same.

### Efficiency

The sum of importances of all data points should equal the marginal contribution of the entire set over an empty set.

### Linearity

If the utility function can be expressed as a sum of two other functions, then the importance of a data point using the combined function should equal the sum of importances computed using the individual functions.

### Null Player

If a data point has a zero marginal contribution to every single subset, its importance should be zero.

### Key Challenge

The number of subsets to enumerate is exponential, making it intractable to compute the exact Shapley value for an arbitrary model.

$$\varphi_i = \frac{1}{N} \sum_{S \subseteq X \setminus \{i\}} \binom{N-1}{|S|}^{-1} (u(S \cup \{i\}) - u(S))$$

# Approximation: Monte Carlo Sampling

## Challenge

Computing Shapley values is intractable.

## Insight

Since Shapley value can be seen as a statistic over exponentially many subsets, we can estimate it using Monte Carlo sampling.

## Approach

Use the permutation-based definition of the Shapley value and sample permutations.

$$\varphi_i(v) = \frac{1}{n!} \sum_R [v(P_i^R \cup \{i\}) - v(P_i^R)]$$

$$\phi_i = \mathbb{E}_{\pi \sim \Pi}[V(S_\pi^i \cup \{i\}) - V(S_\pi^i)]$$



**Abstract**  
data valuation is to quantify the contribution of each training datum to the model's performance.

## Data Shapley: Equitable Valuation of Data for Machine Learning

Amirata Ghorbani<sup>1</sup> James Zou<sup>2</sup>

<sup>1</sup>note of the market place, similar to labor capital (Ghosh, 2020). It has been suggested that certain data carry individual property, and as such individuals should be compensated for the use of their data. Like labor and capital, a fundamental question is how to equitably value individual's data.

<sup>2</sup>note of the market place, similar to labor capital (Ghosh, 2020). It has been suggested that certain data carry individual property, and as such individuals should be compensated for the use of their data. Like labor and capital, a fundamental question is how to equitably value individual's data.

## [Kwon AISTATS '22]

Kwon, Yongchan, and James Zou. "Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning." International Conference on AI and Statistics. 2022. [\[Paper\]](#) [\[Code\]](#)

## [Ghorbani ICML '19]

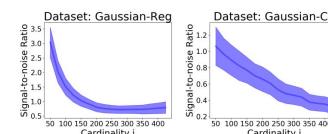
Ghorbani, Amirata, and James Zou. "Data shapley: Equitable valuation of data for machine learning." International conference on machine learning. PMLR, 2019. [\[Paper\]](#) [\[Code\]](#)

## Challenge

We need many Monte Carlo samples to produce good estimates.

## Insight

When estimating the marginal contribution of a data point to a subset, we empirically observe that larger subsets incur a slower signal-to-noise ratio.

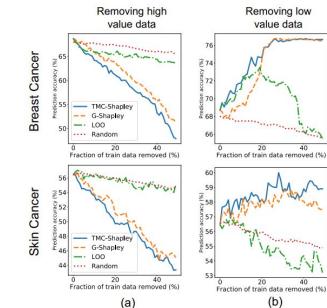


## Approach

Leverage the importance sampling strategy and apply a larger weight to smaller subsets, based on the beta distribution.

## Benefits

Estimating the Shapley value becomes tractable and is shown to be effective at identifying important data points.



## Shortcomings

Each Monte Carlo sample relies on retraining the model from scratch, which is expensive for large models.

# Approximation: K-Nearest Neighbor Surrogate Model

## Challenge

To get good Shapley value estimates, we need to retrain the model many times.

## Insight

The simple KNN classifier can make it easy to design efficient and exact algorithms.

## Approach

Use the KNN model as a proxy to develop an exact Shapley computation algorithm with polynomial time complexity.



**Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms**

Ruoxi Jia<sup>1</sup> David Dao<sup>2</sup> Boxin Wang<sup>3</sup> Frances Ann Hubis<sup>4</sup> Nezihie Merve Gurel<sup>5</sup>  
 Bo Li<sup>6</sup> Ce Zhang<sup>7</sup> Costas Spanos<sup>8</sup> Dawn Song<sup>9</sup>

<sup>1</sup>UC Berkeley <sup>2</sup>ETH Zurich <sup>3</sup>Zhejiang University <sup>4</sup>UCLA  
 ruoxijia@berkeley.edu, danieladao@mit.edu, boxin.wang@zjhu.edu.cn, hubis@ee.ucla.edu, nezihie.gurel@ethz.ch,  
 bo.li@zjhu.edu.cn, ce.zhang@ethz.ch, costas.spanos@ucla.edu, song.dawn@berkeley.edu

**ABSTRACT**  
 Given a dataset  $D$  containing millions of data points and a data point  $x$ , it is challenging to query for the  $k$ -th nearest neighbor (NN) of  $x$ . In this paper, we propose a task-specific data valuation framework for nearest neighbor search. The key idea is that the Shapley value of a single data point  $x$  provides a powerful measure for the data point's challenges to become the top-1 neighbor. By leveraging the Shapley value, we can quickly identify the most valuable data points in  $D$  and then use them to build a proxy model. This proxy model can then be used to quickly find the  $k$ -th NN of  $x$ . Our experiments show that our proposed framework is significantly faster than state-of-the-art methods. We also demonstrate that our framework is more accurate than existing methods. The code is available at <https://github.com/jiaruoxi/TaskSpecificDataValuation>.

### 1. INTRODUCTION

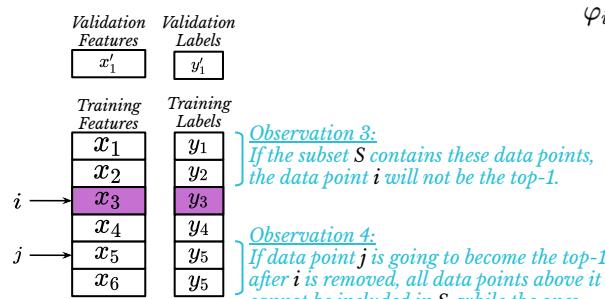
Jia, Ruoxi, et al. "Efficient task-specific data valuation for nearest neighbor algorithms." Proceedings of the VLDB Endowment 12.11 (2019): 1610-1623. [\[Paper\]](#) [\[Code\]](#)

### Example Situation

- We are computing the Shapley value of data point  $i$
- Data is sorted by similarity to the validation data point

### Observation 1:

Since  $K=1$ , for any subset  $S$ , the top-1 data point will determine the model prediction.



### Starting point: Shapley value definition

$$\varphi_i = \frac{1}{N} \sum_{S \subseteq X \setminus \{i\}} \binom{N-1}{|S|}^{-1} (u(S \cup \{i\}) - u(S))$$

Observation 2:

If data point  $i$  is not in the top-1, this term will be zero.

### Dynamic Programming

$$\varphi_i(t) = \frac{1}{N} \sum_{j=i+1}^N \sum_{a=1}^{n-j} \binom{N-1}{a}^{-1} (u(\{i\}) - u(\{j\})) \binom{N-j}{a}$$

### Final Simplification

$$\varphi_i(t) = \frac{1}{N} \sum_{j=i+1}^N (u(\{i\}) - u(\{j\})) \binom{N-j}{j+1}$$

## Result:

After sorting the data, we can compute exact Shapley values in a single pass. Final computational complexity is

$$\mathcal{O}(N \log N)$$

# Approximation: Taylor Expansion

## Challenge

If we are using a large and complex model, retraining will be extremely slow (preventing Monte Carlo approaches), and the KNN approximation will be biased.

## Insight

Models trained with stochastic gradient descent (SGD) compute the loss function many times, over many random subsets of the training dataset. Furthermore, the changes in the model quality metric that are small enough to be effectively approximated with Taylor expansion.

## Approach

Redefine the utility function to measure the cumulative impact of a training data point on the validation loss across gradient update steps.

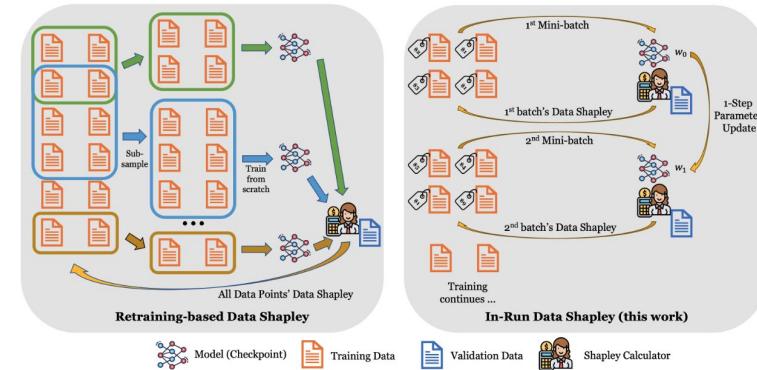
Redefined “local utility function” of subset  $S$  of a single SGD minibatch:

$$U^{(t)}(S; z^{(\text{val})}) := \underbrace{\ell(\tilde{w}_{t+1}(S), z^{(\text{val})}) - \ell(w_t, z^{(\text{val})})}_{\text{Model updated only using data from } S} - \underbrace{\ell(w_t, z^{(\text{val})})}_{\text{Model at SGD step } t}$$

$$\tilde{w}_{t+1}(S) := w_t - \eta_t \sum_{z \in S} \nabla \ell(w_t, z)$$

Redefined “global utility function” of subset  $S$  over the entire SGD run:

$$U(S) = \sum_{t=0}^{T-1} U^{(t)}(S)$$



Published as a conference paper at ICLR 2025.

### DATA SHAPLEY IN ONE TRAINING RUN

Jiachen T. Wang  
Princeton University  
Prateek Mittal  
Princeton University  
David Song  
UC Berkeley  
Rong Jin  
Virginia Tech

#### ABSTRACT

Data Shapley offers a principled framework for attributing the contribution of individual training data points to the final model performance. However, the computation of Shapley requires re-training the entire model from scratch, which becomes computationally infeasible for large-scale models. Additionally, this retraining-based approach is not suitable for large-scale training datasets with many gradient steps, which may often be of interest in practice. This paper introduces a novel approach for calculating Data Shapley in one training run. Our method is specifically designed for assessing data contribution for a particular model during its training process. It does not require re-training the entire model after each update iteration and accumulates these values throughout the training process. We demonstrate that our approach is able to calculate Data Shapley values in linear time to the size of foundation models. In its most optimized implementation, our method adds only a negligible overhead compared to standard training. This allows us to efficiently calculate overall contribution of data points during the foundation model pre-training stage. We present several case studies that illustrate the effectiveness of our approach and their implications for copyright in generative AI and protecting data privacy.

[Wang ICLR '25]

Wang, Jiachen T., et al. "Data Shapley in One Training Run." The Thirteenth International Conference on Learning Representations. [\[Paper\]](#) [\[Blog\]](#)

- 1) Introducing the Concept of Data Importance
- 2) Examples of Data Attribution Functions
- 3) Case Study of Shapley Value as a Measure of Importance
- 4) Applications of Data Importance**

# Influence Function for Explaining Fairness Errors

## Challenge

Data attribution gives us an ordered list of data points that impact model quality, but it does not explain what makes these data points impactful.

## Insight

If we group important data points based on common predicates, we can derive more powerful conclusions about factors that cause models to underperform.

## Approach

First, use influence functions to compute data importance with respect to fairness metrics. Second, use lattice-based search to identify combinations of predicates that define data subsets that are both small and impactful.

### SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

#### Interpretable Data-Based Explanations for Fairness Debugging

Romila Pradhan<sup>1</sup>  
Purdue University  
USA  
rpradhan@purdue.edu

Jiongli Zhu<sup>2</sup>  
University of California,  
San Diego  
USA  
jzhu14@ucsd.edu

Boris Glavic<sup>3</sup>  
Illustrative Institute of  
Technology  
USA  
bglavic@iit.edu

##### ABSTRACT

A wide variety of fairness metrics have been proposed to detect unfair behaviors. These metrics have been proposed to detect the inherent bias in machine learning models that are used in critical real-world applications. In this paper, we propose a framework for generating explanations using cutting XAI techniques in mathematically sound ways. We introduce a system called Gomora, a system that produces compact, interpretable, and causal explanations for fairness errors. Gomora identifies the most impactful subset of the training data that are used for the fairness metric. Specifically, it introduces the concept of causal responsibility which quantifies the causal effect of a feature on the fairness metric. This concept is used to identify the most impactful removing or updating subset of the training data. The system also generates causal responsibility rules that are used to identify the top-k features that explain model bias by utilizing techniques from the field of causal inference. Gomora also provides causal responsibility and using pruning rules to manage the large search space of causal responsibility rules. Finally, we introduce a system called Gomora-explain for generating interpretable explanations for data points that are identified as being impactful by Gomora.

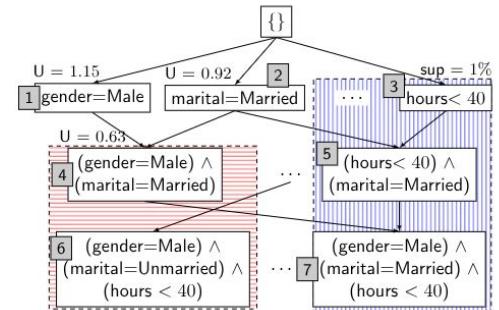
[Zhu SIGMOD '22]

Pradhan, Romila, et al. "Interpretable data-based explanations for fairness debugging." Proceedings of the 2022 international conference on management of data. 2022. [Paper]

## Data points ordered by importance

age	education	marital	...	gender	income
39	Bachelors	Never-married	...	Male	$\leq 50K$
53	11th	Never-married	...	Male	$\leq 50K$
28	Bachelors	Married-civ-spouse	...	Female	$\leq 50K$
37	Masters	Married-civ-spouse	...	Female	$\leq 50K$

Lattice-based search identifies predicates that select the most impactful training data subsets



## Combinations of predicates that explain model behavior

1	Gender = Female	∧	Relationship = Not married	∧	Education = Associate-voc
2	Gender = Male	∧	Relationship = Spouse	∧	Hours < 40
3	Gender = Male	∧	Education = Prof-school		

# Debugging the LLM Retrieval Corpus

## Challenge

*Retrieval augmented generation (RAG) is a widely used technique for providing pre-trained large language models (LLMs) with task-specific context. Data errors in the retrieval corpus have a negative impact on model quality.*

## Insight

*The role of a retrieval corpus to an LLM is similar to the role of a training dataset to a classical ML model.*

## Approach

*Define a data attribution function that will compute the importance of data points in the retrieval corpus. Use this to identify and debug data errors.*

### Improving Retrieval-Augmented Large Language Models via Data Importance Learning

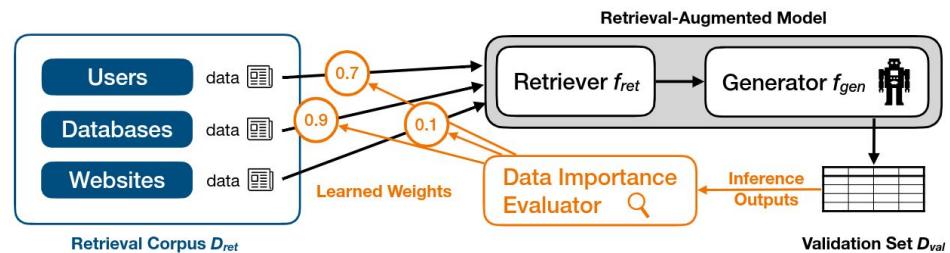
Xiaozhong Lyu<sup>1</sup> Stefan Graepel<sup>2</sup> Samantha Biagioli<sup>3</sup> Shaoqiang Wei<sup>4</sup>  
<sup>1</sup>Erlangen University, <sup>2</sup>ETH Zurich, <sup>3</sup>University of Amsterdam, <sup>4</sup>Apple

#### Abstract

Retrieval augmentation enables large language models to take advantage of external knowledge bases to improve their performance on downstream tasks. However, the performance of such retrieval-augmented models is limited by the data quality in the retrieval corpus. In this paper, we propose a data importance learning based on multi-linear regression for evaluating the data importance of retrieved documents. We also propose a data pruning algorithm that removes low-importance data from the retrieval corpus. The proposed data importance learning and one key contribution of this paper is a polynomial tree algorithm that computes the data importance of data points in the retrieval corpus using the multi-linear regression model. We evaluate our proposed data importance learning on a validation set, the data importance of data points in the retrieval corpus using the proposed multi-linear regression model is more accurate than that of the state-of-the-art efficient ( $\epsilon$ - $\delta$ ) approximation algorithm. Our experimental results illustrate that the proposed data importance learning and data pruning algorithm can significantly improve the performance of retrieval-augmented large language models.

[Lyu arXiv '23]

Lyu, Xiaozhong, et al. "Improving retrieval-augmented large language models via data importance learning." arXiv preprint arXiv:2307.03027 (2023). [\[Paper\]](#) [\[Code\]](#)



$$U(f_{gen}, f_{ret}, \mathcal{D}_{val}, \mathcal{D}_{ret}) := \sum_{x_i \subseteq \mathcal{D}_{val}} U(f_{gen}(x_i, f_{ret}(x_i, \mathcal{D}_{ret})))$$

$$\tilde{U}(w_1, \dots, w_M) := \sum_{S \subseteq \mathcal{D}_{ret}} U(S) \underbrace{\prod_{d_i \in S} w_i \prod_{d_i \notin S} (1 - w_i)}_{P[S]}$$

DATASET	GPT-JT (6B)	GPT-JT (6B) W/ RETRIEVAL				GPT-3.5 (175B)
		VANILLA	+LOO	+REWEIGHT	+PRUNE	
BUY	0.102	0.789	0.808	<b>0.815</b>	0.813	0.764
RESTAURANT	0.030	0.746	0.756	<u>0.760</u>	<b>0.761</b>	0.463

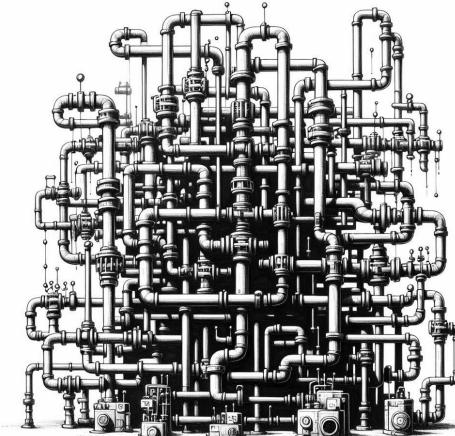
## Key Takeaways of Part I

- Data attribution is a useful powerful framework for approaching the problem of data error detection.
- There are many existing data attribution methods with various strengths and shortcomings.
- The most powerful methods face scalability issues that have been tackled by existing research with many opportunities for future improvements.

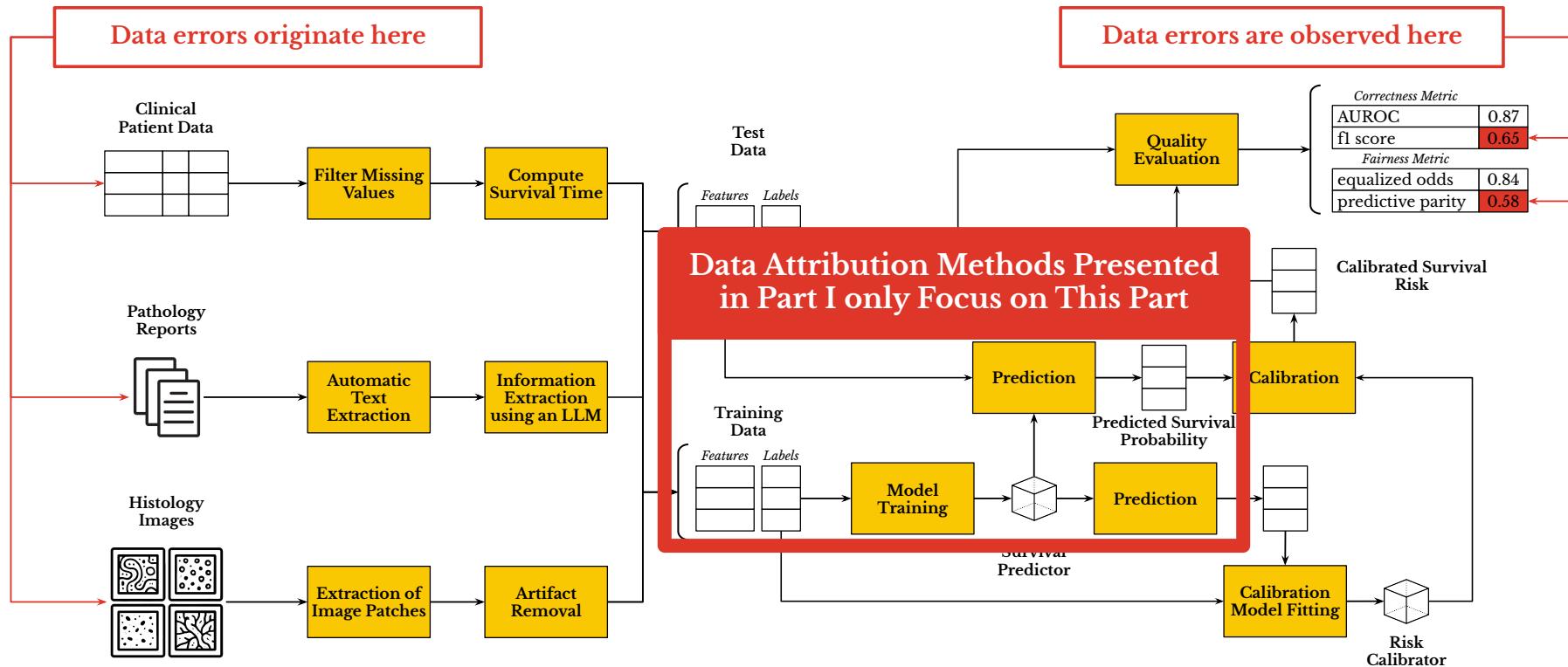
# Part II: Data Debugging in ML Pipelines

---

Sebastian Schelter



# Gap between Attribution Methods and ML Pipelines



**Challenge:** How should we debug ML pipelines?

1) Gap between Attribution Methods and ML Pipelines

**2) Libraries and Systems for ML Pipelines**

3) Characteristics of Real World ML Pipelines

4) Methods for Debugging ML Pipelines

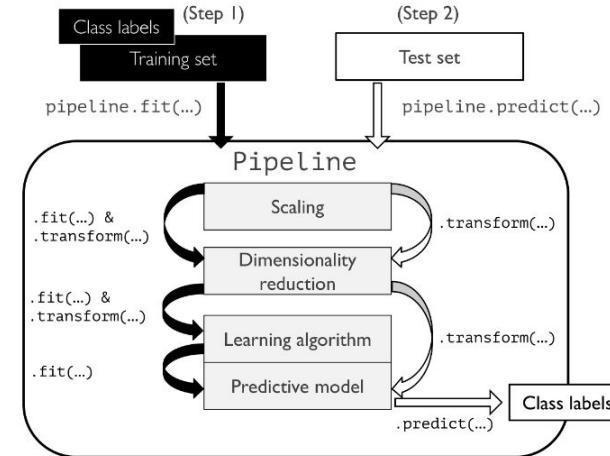
# Scikit-Learn

## Highlights

- Among the most popular data science Python libraries
- Has implementations of many machine learning models, as well as data processing operators
- Introduced the **estimator/transformer abstraction** for composing complex, nested pipelines
  - **Transformer:** tuple-at-a time transformation
  - **Estimator:** create a data-specific transformer via a global aggregation over the data



[Scikit-Learn]  
 Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830. [Paper]  
[\[Website\]](#) [\[Code\]](#)



Source: <https://vitalflux.com/scikit-machine-learning-pipeline-python-example/>

# Tensorflow Extended (TFX)



## Highlights

- *End-to-end platform for production ML pipelines*
- *Built on TensorFlow and optimized for scalability, strong emphasis on model validation and monitoring*
- *Includes reusable components for pipelines, inspired by estimator/transformer paradigm*
- *Apache Beam for dataflow operations, Tensorflow for numerical operations*

KDD 2017 Applied Data Science Paper

KDD '17, August 13–17, 2017, Halifax, NS, Canada

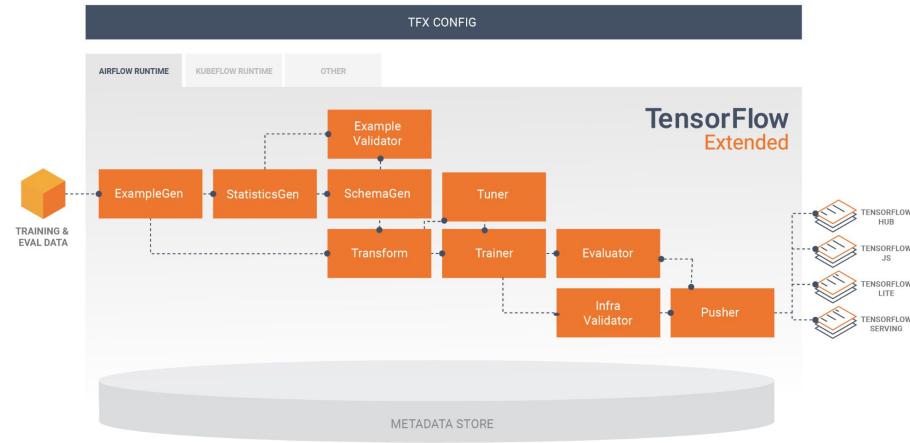
### TFX: A TensorFlow-Based Production-Scale Machine Learning Platform

Denis Baylor, Eric Brooks, Hong-Tai Cheng, Noah Frey, Clinton Yu Guo, Zesheng Huang,  
 Salma Hritzel, Manisha Iyer, Vipin Jain, Ming Tang, Ravi K. Rao, Chin-Yan Kuo, Michael Lee,  
 Chirag Mehta, Akshay Narendran, Nivedita Polyzotis, Sudip Ramdas, Sudip Roy,  
 Steven Euijong Whang, Martin Wicke, Jacek Wilkiewicz, Xu Zhang, Martin Zinkevich  
 Google Inc.

Abstract machine learning as a tool to gain knowledge from data across a broad spectrum of use cases and products, ranging from recommendation engines to real-time bidding and prediction for advertising [11, 22] and even the protection of endangered species [2].

The challenge of applying machine learning to a specific use case is simple: at the training phase, a user provides a dataset and defines a loss function. At the inference phase, the model takes features as input and produces predictions. However, this approach does not scale well together with the learning need to be deployed in production. In fact, the learning need to be deployed in production is often much more complex than what one might expect. This paper describes how we approached this challenge to deal with a diverse range of failures that can happen during the deployment process, such as serving latency spikes related to breaking this type of automation and serving loops.

In this paper, we introduce TFX, a TensorFlow-based general-purpose machine learning platform implemented in Go. We show how, in our experience, we were able to standardize the configura-

Source: <https://www.tensorflow.org/tfx/guide>

[TFX]

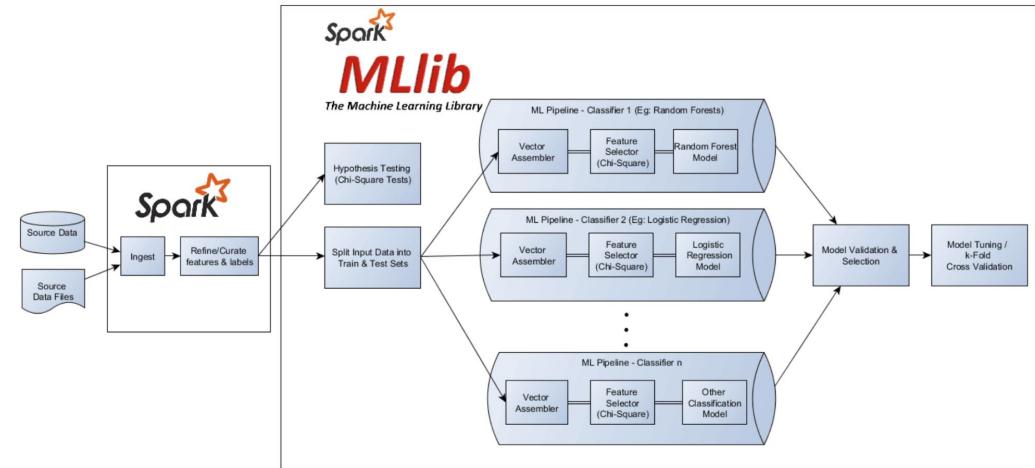
Baylor, Denis, et al. "Tfx: A tensorflow-based production-scale machine learning platform." Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017. [\[Paper\]](#) [\[Website\]](#) [\[Code\]](#)

# Spark MLLib



## Highlights

- Built on top of Apache Spark
- Includes implementations for classification, regression, clustering, collaborative filtering, and dimensionality reduction
- Works natively with Spark DataFrames, SQL, and streaming data
- Adoption of estimator/transformer paradigm from scikit-learn



Source: <https://www.qubole.com/developers/spark-getting-started-guide/workflow>

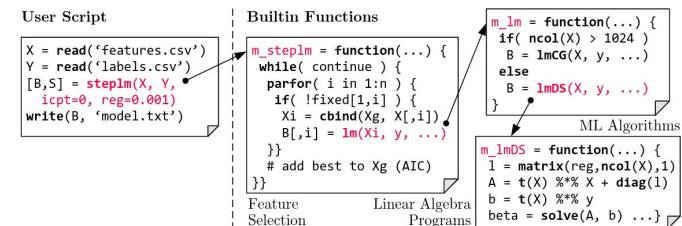
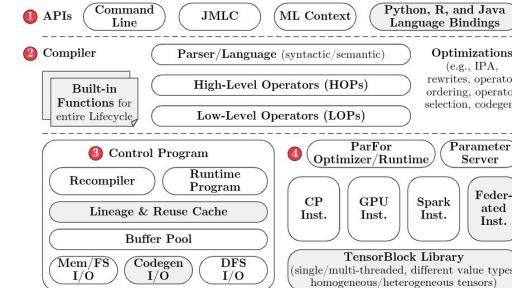
MLlib: Machine Learning in Apache Spark	
Xiangrui Meng <sup>1</sup> Databricks, 160 Spear Street, 10th Floor, San Francisco, CA 94105 Joseph Bradley Databricks, 160 Spear Street, 10th Floor, San Francisco, CA 94105 Burak Yavuz Databricks, 160 Spear Street, 10th Floor, San Francisco, CA 94105 Evan铺平 UC Berkeley, 460 Soda Hall, Berkeley, CA 94720 Shivaram Venkataraman Berkeley, 460 Soda Hall, Berkeley, CA 94720 Dario Diaz Databricks, 160 Spear Street, 10th Floor, San Francisco, CA 94105 Jenvery Freeman RISELab Research Corpus, 18005 Rella Dr, Ashburn, VA 20147 DB Tsai Netfrix, 999 University Ave, Los Gatos, CA 95032 Manish Srivastava Omniture Logic, 1531 Oracle Street, Menlo Park, CA 94025	MENG@DATABRICKS.COM JOSEPH@DATABRICKS.COM BURAK@DATABRICKS.COM SPARK@CS.BERKELEY.EDU SHIVARAM@CS.BERKELEY.EDU DARIO@DATABRICKS.COM JENVERY@NETFLIX.COM DB.TSAI@NETFLIX.COM MANISH@ORACLELOGIC.COM
<b>[MLlib]</b>	
Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark." Journal of Machine Learning Research 17.34 (2016): 1-7. [Paper] [Website] [Code]	

# Apache SystemDS



## Highlights

- Designed for scalable and efficient execution on both single-node and distributed environments
- Offers a high-level scripting language for expressing ML algorithms and workflows with a declarative R-like language
- Performs cost-based optimization and automatic operator selection for efficient execution across different hardware endpoints
- Optimised feature encoders based on estimator/transformer paradigm



Journal of Machine Learning Research 27 (2016) 1-7

Submitted 5/15, Published 1/18

### Mlib: Machine Learning in Apache Spark

Xiangrui Meng <sup>1</sup> Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105	JOSEPHBRADICKS.COM
Joseph Bradley Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105	JOSEPHBRADICKS.COM
Burak Yavuz Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105	BURAKDATABRICKS.COM
Evan Thrasher Uber Technologies, 465 Soda Hall, Berkeley, CA 94720	SPARKSFDS.BERKELEY.EDU
Sivaram Venkataraman Berkeley, 465 Soda Hall, Berkeley, CA 94720	SHIVARAM@CS.BERKELEY.EDU
Darren Liu Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105	DAIHLIUDATABRICKS.COM
Jevgeni Freidin IBM T.J. Watson Research Center, 1900 Bell Rd, Armonk, NY 10504	FREIDIN@TJWATSON.IBM.COM
DB Tsai Netfrix, 999 Market Ave, Los Gatos, CA 95032	DBT@NETFLIX.COM
Manish Srivastava Oxygenlogic, 1531 Oracle Street, Menlo Park, CA 94025	MANISH@OXYGENLOGIC.COM

### [SystemDS]

Boehm, Matthias, et al. "SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle." 10th Conference on Innovative Data Systems Research. 2020. [\[Paper\]](#) [\[Website\]](#) [\[Code\]](#)

Phani, Arnab, et al. "UPLIFT: parallelization strategies for feature transformations in machine learning workloads." Proceedings of the VLDB Endowment, Volume 15, Issue 11, 2020. [\[Paper\]](#)

# ML Pipelines in the Cloud



## Netflix Metaflow

[\[Website\]](#) [\[Documentation\]](#)

### Highlights

- Notebook based development environment
- Storing and tracking of code, data and models
- Scaling from local execution to the cloud



Amazon SageMaker

## Amazon SageMaker Pipelines

[\[Website\]](#) [\[Documentation\]](#)

### Highlights

- Define, automate, and manage end-to-end ML workflows
- Automatically tracks pipeline artifacts
- Leverages AWS Cloud infrastructure



Azure Machine Learning

## Azure Machine Learning Pipelines

[\[Website\]](#) [\[Documentation\]](#)

### Highlights

- Orchestration of ML workflows with reusable, modular pipeline components
- Versioning, monitoring, and CI/CD integration



Vertex.ai

## Vertex AI Pipelines

[\[Website\]](#) [\[Documentation\]](#)

### Highlights

- Connects with Vertex AI services
- Tracks pipeline steps, metadata, and artifacts
- Orchestrates ML workflows on Google Cloud

# Observations

- No universal way to express ML pipelines, design often prioritises flexibility and ease-of-use
- Many pipelines combine relational / dataflow operators with ML-specific operators based on estimator/transformer abstraction
- Pipelines often executed via multiple runtimes
- Lack of algebraic operator semantics
- Lack of fine-grained data provenance

- 1) Gap between Attribution Methods and ML Pipelines
- 2) Libraries and Systems for ML Pipelines
- 3) Characteristics of Real World ML Pipelines**
- 4) Methods for Debugging ML Pipelines

# Study of Pipelines at Google

## Highlights

- Study of 3000 production pipelines with over 450K models trained over a 4 month period
- About half the pipelines studied used data- and model-validation operators
- Input data typically has up to 100 features, but can have over 10K in extreme cases
- 53% of features were categorical, often with very large domains (averaging over 10M unique values)
- Training accounts for only 20% of the total runtime cost, over 30% is for model validation and 20% for data ingestion
- About 1/4 of model training runs results in model deployment
- Deep learning models account for 60% of pipelines

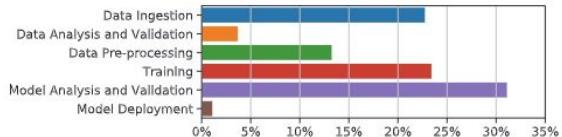


Figure 7: Compute cost of different operators.

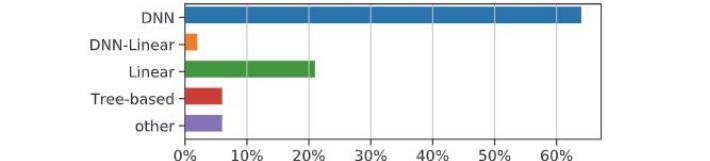
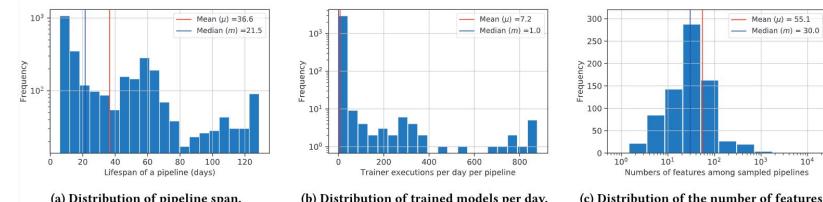
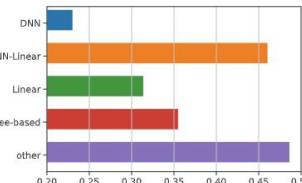


Figure 5: Percentage of Trainer runs with each model type



(f) Model type vs. likelihood of pushes.

Industrial Track Paper  
SIGMOD '21, June 20–21, 2021, Virtual Event, China

**Production Machine Learning Pipelines: Empirical Analysis and Optimization Opportunities**

Doris Xin  
University of California, Berkeley  
dorisx@berkeley.edu

Hui Mao  
Google  
humaoo@google.com

Aditya Parameswaran  
University of California, Berkeley  
adityap@berkeley.edu

Nekhil Polyotis  
Google  
opolyotis@google.com

**ABSTRACT**  
Machine learning (ML) is now commonplace, powering data-driven applications in various organizations. Unlike the traditional perspective of ML as a black box, modern ML pipelines involve many more modeling and analysis components beyond training, where input data is processed and analyzed in various ways before training. However, there is a lack of quantitative evidence regarding the real-world usage of these ML pipelines. This paper aims to understand how data management workloads can be tackled in real-world ML pipelines. We analyze 3000 production ML pipelines, over 450,000 models trained, spanning a year of data from Google. Our analysis reveals the complexity and challenges underlying producing ML. Our analysis reveals that ML pipelines are highly customized, with varying degrees of strength ML pipelines at various granularities. Along the way, we introduce a taxonomy of ML pipelines and identify opportunities for repeatedly reuse components in these ML pipelines, which we call *ML components*. Finally, we propose a set of recommendations to address the challenges underlying producing ML.

**1 INTRODUCTION**  
Machine learning (ML) is now commonplace, powering data-driven applications in various organizations. Unlike the traditional perspective of ML as a black box, modern ML pipelines involve many more modeling and analysis components beyond training, where input data is processed and analyzed in various ways before training. However, there is a lack of quantitative evidence regarding the real-world usage of these ML pipelines. This paper aims to understand how data management workloads can be tackled in real-world ML pipelines. We analyze 3000 production ML pipelines, over 450,000 models trained, spanning a year of data from Google. Our analysis reveals the complexity and challenges underlying producing ML. Our analysis reveals that ML pipelines are highly customized, with varying degrees of strength ML pipelines at various granularities. Along the way, we introduce a taxonomy of ML pipelines and identify opportunities for repeatedly reuse components in these ML pipelines, which we call *ML components*. Finally, we propose a set of recommendations to address the challenges underlying producing ML.

**At the same time, there is evidence from previous literature [1, 11, 14, 44] that ML pipelines are highly customized. Specifically, ML pipelines involves pipelines with many reuse components. These reuse components are often developed on the development of many end-to-end ML systems (e.g., TensorFlow [11], PyTorch [14], etc.).**

[Xin SIGMOD '21]

Xin, Doris, et al. "Production machine learning pipelines: Empirical analysis and optimization opportunities." Proceedings of the 2021 international conference on management of data. 2021. [Paper]

# Study of Pipelines at Microsoft

## Highlights

- Study of over 8M public Jupyter notebooks on GitHub (from 2017, 2019, and 2020), and 2M enterprise pipelines developed with ML.NET*
- Python is emerging as the de-facto standard language for data science (81% of notebooks in 2017 and 91% in 2020)*
- Around 80% cells were linear (no conditional statements) and 76% were completely linear (no conditionals, classes, or functions)*
- Libraries like numpy, matplotlib, pandas, and scikit-learn are used very frequently (e.g., numpy in >60% of notebooks)*
- Few highly used libraries have significant coverage (e.g., top-10 cover ~40% of notebooks, top-100 cover ~75%), but there is a long tail*
- Explicit ML pipelines (defined with sklearn.pipeline) are gaining traction but there are still 5 times more implicit pipelines in GitHub notebooks*
- There is a large number of distinct operators, and a significant portion are user-defined (especially in ML.NET and implicit GitHub pipelines)*

### Data Science Through the Looking Glass: Analysis of Millions of GitHub Notebooks and ML.NET Pipelines

Fotis Psallidas, Yizhen Zhu, Bojian Kang\*, Jordan Redell, Mattia Ieracitano, Suresh Subrahmanyam, Farzad Farnoud, Wentao Wu, Ce Zhang\*, Matias Werner, Aritra Bhattacharya, Carlos Curino, Konstantinos Karanikas, Sami Saitama, Prateek Mittal

\*Equal contribution. Email: psallidas@mit.edu, wuwentao@mit.edu

#### ABSTRACT

The recent success of machine learning (ML) has led to an explosive growth of systems and applications built by researchers, practitioners, and data science (DS) practitioners. This quickly shifting paradigm, however, is challenging for system builders and practitioners to keep up with the pace of innovation. To capture this punctuated through a wide-angle lens, performing an in-depth analysis of millions of GitHub notebooks and ML.NET pipelines publicly authored on GitHub—(a) ML notebooks publicly authored on GitHub—over 8M notebooks total, ranging from simple data processing to complex machine learning models, and (b) ML.NET pipelines publicly authored on GitHub—over 8M notebooks publicly available, ranging from simple data processing to complex machine learning models. Over the past few years, we have used the results

#### [Psallidas SIGMOD Record '22]

Psallidas, Fotis, et al. "Data science through the looking glass: Analysis of millions of github notebooks and ml. net pipelines." ACM SIGMOD Record 51.2 (2022): 30-37. [Paper]

Dimension	Metric	GH17	GH19	GH20
Notebooks	Total	1.23M	4.6M	8.7M
	Deduped	66.0%	65.5%	65.7%
	Linear	26.4%	29.1%	30.3%
	Completely Linear	21.2%	23.3%	24.6%
Languages	Python	81.7%	91.7%	91.1%
	Other	18.3%	8.3%	8.9%
Cells	Total	34.6M	143.1M	261.2M
Code Cells	Total	64.5%	66.4%	66.9%
	Deduped	41.0%	38.6%	38.5%
	Linear	72.1%	80.2%	79.3%
	Completely Linear	68.3%	76.1%	75.6%
Users	Total	100K	400K	697K

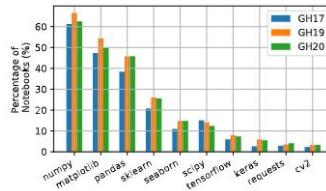


Figure 2: Top-10 used libraries.

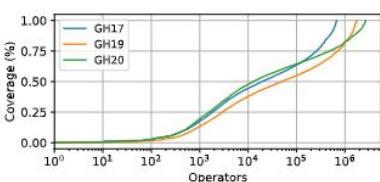
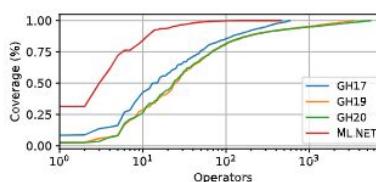


Figure 3: DL libraries usage percentages.

	GH17	GH19	GH20	ML.NET
#Pipelines	Implicit	164K	415K	1.4M
	Explicit	10K	129K	252K

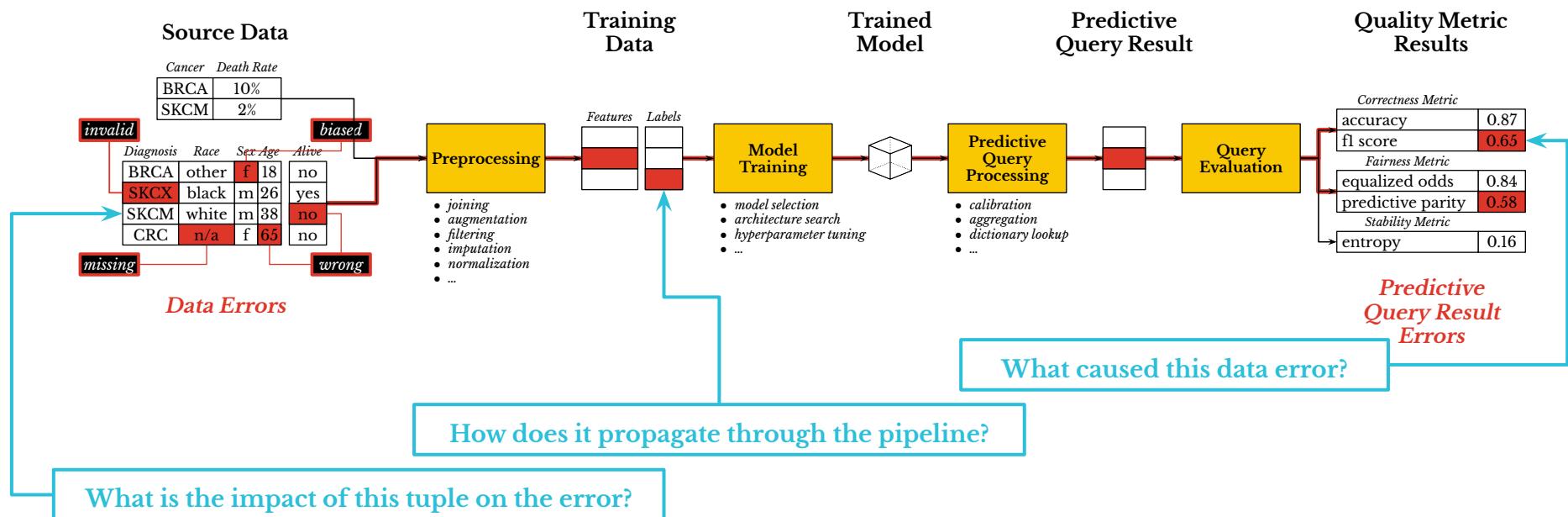
  

	GH17	GH19	GH20	ML.NET
#Distinct Ops	Implicit	668K	1.8M	2.6M
	Explicit	584	3.4K	5.5K



- 1) Gap between Attribution Methods and ML Pipelines
- 2) Libraries and Systems for ML Pipelines
- 3) Characteristics of Real World ML Pipelines
- 4) Methods for Debugging ML Pipelines**

# How should we reason about pipelines?



# Modeling ML Pipelines with “Logical Query Plans”

## Challenge

*Understanding of the semantics of operations and the flow of data required to reason about ML pipelines*

## Insight

*Many common pipeline abstractions offer declarative operations, enables the extraction and definition of “logical query plans”*

## Approach

*Instrument functions of Python data science libraries to extract query plan, enable annotation propagation through operators. Apply rule-based approaches to determine if an error has occurred (e.g. if a bias against a sensitive group has been introduced).*

### Potential issues in preprocessing pipeline:

- ① Join might change proportions of groups in data
- ② Column ‘age\_group’ projected out, but required for fairness
- ③ Selection might change proportions of groups in data
- ④ Imputation might change proportions of groups in data
- ⑤ ‘race’ as a feature might be illegal!
- ⑥ Embedding vectors may not be available for rare names!

### Python script for preprocessing, written exclusively with native pandas and sklearn constructs

```
# load input data sources, join to single table
patients = pandas.read_csv(_)
histories = pandas.read_csv(_)
data = pandas.merge([patients, histories], on=['ssn'])

# compute mean complications per age group, append as column
complications = data.groupby('age_group')
    .agg(mean_complications=('complications', 'mean'))
data = data.merge(complications, on='age_group')

# Target variable: people with frequent complications
data['label'] = data['complications'] >
    1.2 * data['mean_complications']

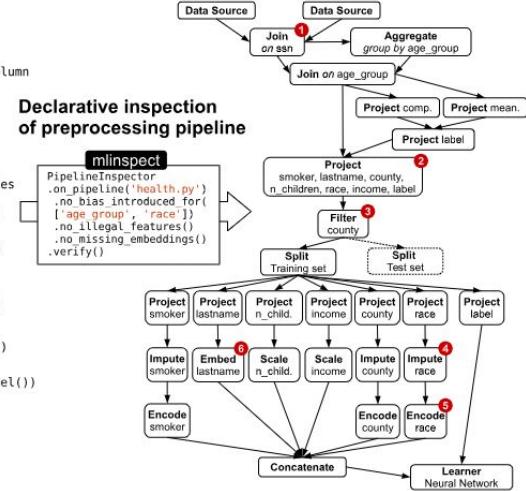
# Project data to subset of attributes, filter by counties
data = data[['smoker', 'last_name', 'county',
    'num_children', 'race', 'income', 'label']]
data = data[data['county'].isin(counties_of_interest)]

# Define a nested feature encoding pipeline for the data
impute_and_encode = sklearn.Pipeline([
    (sklearn.SimpleImputer(strategy='most_frequent')),
    (sklearn.OneHotEncoder())])
featureisation = sklearn.ColumnTransformer(transformers=[
    ('impute_and_encode', ['smoker', 'county', 'race']),
    (Word2VecTransformer(), 'last_name'),
    (sklearn.StandardScaler(), ['num_children', 'income'])])

# Define the training pipeline for the model
neural_net = sklearn.KerasClassifier(build_fn=create_model())
pipeline = sklearn.Pipeline([
    ('features', featureisation),
    ('learning_algorithm', neural_net)])

# Train-test split, model training and evaluation
train_data, test_data = train_test_split(data)
model_pipeline.fit(train_data, train_data.label)
print(model.score(test_data, test_data.label))
```

### Corresponding dataflow DAG for instrumentation, extracted by mlinspect

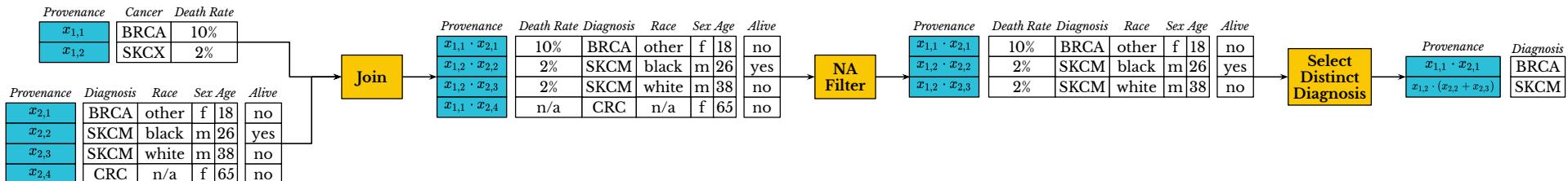


# Leveraging the Provenance Semiring Framework

## Highlights

- Theoretical framework analyzing the relationship between input and output tuples of relational queries
- It allows us to determine the presence of an output tuple as a function of the presence of an input tuples
- Easy to adapt for ML pipelines once logical query plan is known

## Application to an Example Pipeline



**ABSTRACT**  
 We observe that in all four cases, the calculations with annotations are strikingly similar. This suggests looking for an algebraic structure or annotation that captures the above observations. In this paper, we propose such an annotation for this purpose. In fact, we can show that the laws of commutativity and associativity hold for the annotations in SQL. Having identified commutative readings as annotations, we can now proceed to define a semiring representation of annotations in SQL. This provides a new way for applications which require rich provenance information to store and query it. We also show how to use this semiring representation for the most general and strict notion of provenance semantics. Finally, we show that this semiring representation is that of polynomials. This provides a new way for applications to store and query provenance annotations. We also show how to use this semiring representation to evaluate database queries. To demonstrate the usefulness of our approach, we provide a new way to extract and analyze provenance information from ML pipelines.

We observe that in all four cases, the calculations with annotations are strikingly similar. This suggests looking for an algebraic structure or annotation that captures the above observations. In this paper, we propose such an annotation for this purpose. In fact, we can show that the laws of commutativity and associativity hold for the annotations in SQL. Having identified commutative readings as annotations, we can now proceed to define a semiring representation of annotations in SQL. This provides a new way for applications which require rich provenance information to store and query it. We also show how to use this semiring representation for the most general and strict notion of provenance semantics. Finally, we show that this semiring representation is that of polynomials. This provides a new way for applications to store and query provenance annotations. We also show how to use this semiring representation to evaluate database queries. To demonstrate the usefulness of our approach, we provide a new way to extract and analyze provenance information from ML pipelines.

[Green SIGMOD '07]

Green, Todd J., Grigoris Karvounarakis, and Val Tannen. "Provenance semirings." Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 2007. [Paper]

# Debugging Preprocessing Pipelines with Datascope

[Attribution Function: Shapley Value]

## Challenge

*Computing the Shapley value using the KNN proxy method assumes that the presence of a single source data point maps directly to a single data point fed to the model. Hence, the results are not directly applicable to arbitrary pipelines.*

## Insight

*We can use the provenance framework to analyze pipelines and develop PTIME algorithms for computing the Shapley value. We notice that there are three canonical types of pipelines that are both representative of real-world pipelines, and lend themselves to efficient Shapley value computation.*

## Approach

*Compile provenance polynomials to Additive Decision Diagrams and use them to compute Shapley values in PTIME.*

Published as a conference paper at ICLR 2024

DATA DEBUGGING WITH SHAPLEY IMPORTANCE OVER  
MACHINE LEARNING PIPELINES

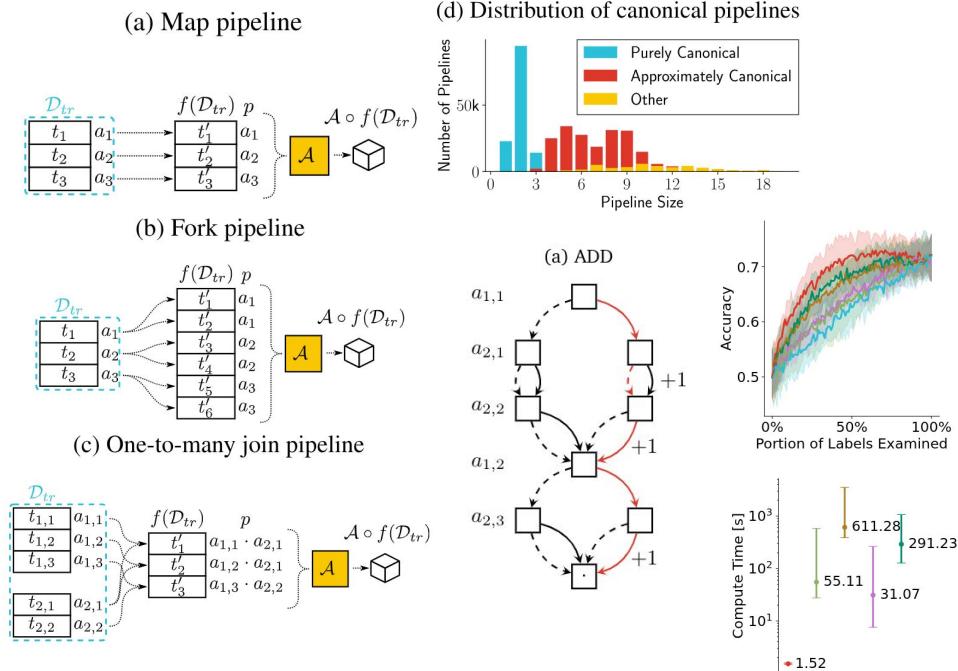
Bojan Karlaš<sup>1</sup>, David Basav<sup>2</sup>, Matteo Interlenghi<sup>3</sup>, Sebastian Schelter<sup>4</sup>, Wentao Wu<sup>5</sup>, Ce Zhang<sup>6</sup>  
<sup>1</sup>Harvard University, <sup>2</sup>ETH Zurich, <sup>3</sup>Microsoft, <sup>4</sup>University of Amsterdam, <sup>5</sup>University of Chicago  
<sup>6</sup>Stanford University, harvard@ethz.ch

### ABSTRACT

When a machine learning (ML) model exhibits poor quality (e.g., poor accuracy or fairness), the problem can often be traced back to errors in the training data. Being able to discover the data errors and fix them is critical for improving the quality of the ML models. This has been a lot of attention recently. One promising way to measure “data importance” with respect to model quality is to use Shapley values. However, computing Shapley values over ML models is in general NP-hard. In this paper, we propose Datascope, a method for efficiently computing Shapley-based data importance over ML pipelines. Datascope leverages the provenance framework to represent the data flow and its dynamics, which appears to be unique in the space of real-world ML code. This presents a natural way to compile the provenance polynomials to Additive Decision Diagrams (ADDs). Datascope is a method for efficiently computing Shapley-based data importance over ML pipelines. It is based on ADDs, which are known to be more efficient than other methods in terms of computational speed. Finally, our experimental evaluations demonstrate that our methods are significantly faster than state-of-the-art methods for data debugging, and in some cases even outperform them. We release our code as an open-source data debugging library available at [github.com/bojankarlas/datascope](https://github.com/bojankarlas/datascope).

[Karlaš ICLR '24]

Karlaš, Bojan, et al. "Data Debugging with Shapley Importance over Machine Learning Pipelines." The Twelfth International Conference on Learning Representations. 2024. [\[Paper\]](#) [\[Website\]](#) [\[Code\]](#)



# Debugging Predictive Queries with Rain

[Attribution Function: Influence]

## Challenge

The existing influence-based attribution methods assume that the model predictions are directly used for computing model quality. However, model inference is often part of a larger predictive query.

## Insight

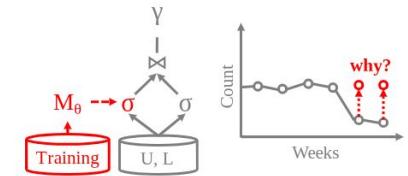
Using provenance polynomials to track lineage starting from training tuples all the way to predictive query outputs allows us to make the entire expression differentiable.

## Approach

User complaints on query outputs (e.g. what-if-queries) are used to identify errors. Make the entire query differentiable using provenance polynomials and run the influence framework to identify errors in the training dataset.

**Q**

```
SELECT COUNT(*)
  FROM Users U JOIN Logins L
    ON U.ID = L.ID
   WHERE L.active_last_month AND
        Mθ.predict(U.*) = "Churn"
```



### Complaint-driven Training Data Debugging for Query 2.0

Weiyuan Wu  
Simon Fraser University  
Burnaby, BC, Canada  
wyw@sfu.ca

Lempros Fidas  
Columbia University  
New York, NY  
lfdas@cse.columbia.edu

Eugene Wu  
Columbia University  
New York, NY  
ewwu@columbia.edu

Jianwei Wang  
Simon Fraser University  
Burnaby, BC, Canada  
jwang@sfu.ca

#### ABSTRACT

As the use of machine learning (ML) increases rapidly across all industry sectors, there is a significant interest among practitioners in how to debug ML pipelines. Query 2.0, which integrates model inference into SQL queries, debugging Query 2.0 is no challenge since an unexpected query result can be easily traced back to the wrong labels (corrupted features). In response, we propose Rain, a complaint-driven training data debugging framework for Query 2.0. Rain allows users to specify complaints on the query's output, and Rain allows users to quickly identify the erroneous training data that caused the complaints.

#### INTRODUCTION

Database researchers have long debated the value of integrating model inference within the DBMS: data used for model inference is already in the DBMS; it brings the code closer to the data; and it makes it easier to implement new

Wu, Weiyuan, et al. "Complaint-driven training data debugging for query 2.0." Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. 2020. [Paper]

# ArgusEyes - Continuous Integration for ML Pipelines

## Challenge

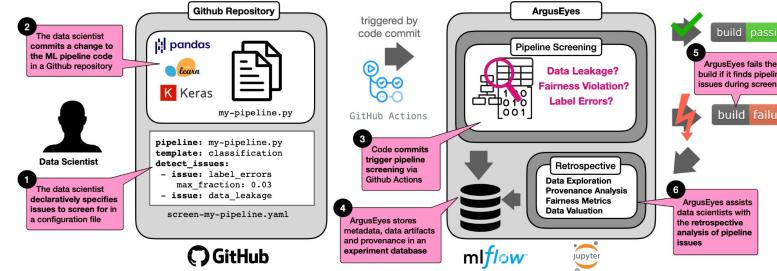
*ML systems lack sophisticated testing infrastructure developed for classical software engineering. Many data-related problems only become apparent in production.*

## Insight

*Logical query plans for ML pipelines combined with data debugging techniques enable ML-specific CI infrastructure.*

## Approach

*Instrument, execute and screen ML pipelines for declaratively specified pipeline issues, and analyze data artifacts and their provenance to catch potential problems early before deployment to production.*



### Proactively Screening Machine Learning Pipelines with ArgusEyes

Sebastian Schelter  
Universität Amsterdam  
Amsterdam, The Netherlands  
s.schelter@uva.nl

Stefan Grafeberg  
AI Lab, University of Amsterdam  
Amsterdam, The Netherlands  
s.grafeberg@uva.nl

Shubha Guha  
AI Lab, University of Amsterdam  
Amsterdam, The Netherlands  
s.guha@uva.nl

Bryan Kardell  
Harvard University  
Boston, United States  
bkgd@csail.mit.edu

Cs Zhang  
ETH Zurich  
Zürich, Switzerland  
c.zhang@ethz.ch

**ABSTRACT**  
Software systems that learn from data with machine learning (ML) are ubiquitous. ML pipelines in these applications often suffer from a variety of data-related issues, such as data leakage, fairness violations, which require running large compute jobs to detect. These issues are often only discovered after they caused damage to real data and (bad) errors [1] in training scores, which are often only detected at hindsight after deployment, after they already caused damage to real data. ArgusEyes [2] is a system that can automatically detect these issues in ML pipelines for declaratively specified pipelines and systems. ML pipelines for declaratively specified pipelines often have to support many different types of data, such as structured and unstructured data, which makes it difficult to detect potential problems early before deployment to production. We demonstrate how ArgusEyes can automatically detect these potential problems in real-world ML pipelines.

Schelter, et al.: “Proactively Screening Machine Learning Pipelines with ArgusEyes.” Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (demo). 2023. [\[Paper\]](#)

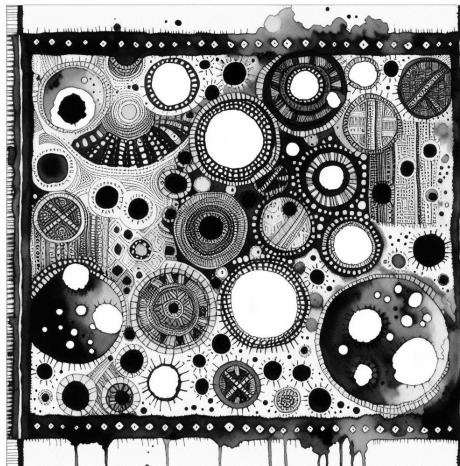
## Key Takeaways of Part II

- Attribution methods presented in Part I assume models are trained with source data.
- ML pipelines are complex and present many opportunities for methods development.
- Logical query plans combined with data provenance offer a powerful framework for analyzing ML pipelines.

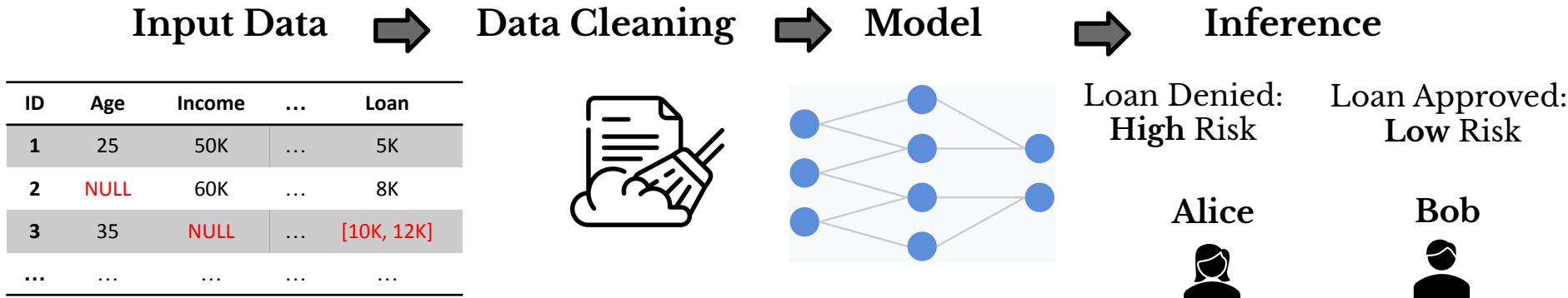
# Part III: Learning from Uncertain and Incomplete Data

---

Babak Salimi



# The Standard ML Pipeline



⚠ Common Assumption: once we “clean” the data, the pipeline consumes accurate and unbiased inputs.

✗ Reality: cleaning/pre-processing yields one reconstruction, driven by heuristic choices & domain assumptions → it can embed hidden bias and hide genuine uncertainty.

→ Key insight for Part III: even after best-effort cleaning, *real-world data remains incomplete and uncertain*. Our models—and the theory behind them—must make that uncertainty explicit rather than ignore it.

# Why “Fixing” Data Errors Is Impossible in Principle

**Missing values** (  / )

*Irrecoverable uncertainty:* any imputation is just a guess; the true value is unobservable.

*Unverifiable assumption:* “missing at random,” parametric model of the data, etc.

[Pearl & Mohan, AAAI 2014], [Mohan, Pearl & Tian, NeurIPS 2013]

**Measurement / annotation bias** ( sentiment,  diagnoses)

*Systematic distortion:* recorded values can be consistently wrong.

*Unverifiable assumption:* symmetric, independent label-noise model.

[Pearl, UAI 2010], [Zhang & Yu, IJCAI 2015]

# Why “Fixing” Data Errors Is Impossible in Principle

**Selection bias & missing counterfactuals** (⚠ rejected-loan applicants, excluded patients)

*Unknown outcomes*: whole sub-populations are never seen.

*Finite-sample limits*: re-weighting needs the true selection mechanism—which we can’t test.

[Bareinboim, Tian & Pearl, AAAI 2014] [Cortes et al., ALT2008],  
[Heckman, Econometrica 1979]

**Schema / integration mismatch** (⚠ inconsistent units, ✗ fuzzy entity resolution)

*Ambiguous merges*: no ground-truth correspondences.

*Pre-processing bias*: heuristics distort original distributions; matching is probabilistic.

[Dong, Halevy & Madhavan, VLDB 2009],  
[Getoor & Machanavajjhala, ACM 2012]

# Challenges with Traditional Data Pipelines

Input Data



Data Cleaning



Model



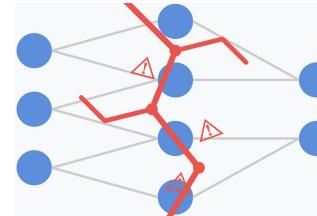
Inference

Loan Denied:  
High Risk

Alice

Loan Approved:  
Low Risk

Bob



**Generalization Failure** – Models trained on “repaired” data collapse under real-world shifts.

**✗ High-Stakes Mis-decisions** – Hidden bias drives flawed credit, medical, and justice outcomes.



**Broken Uncertainty** – Bayesian & conformal intervals lose calibration when data are incomplete.

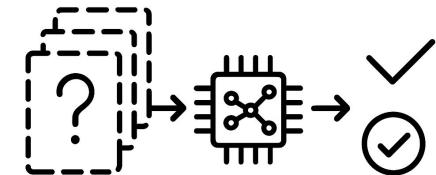
# Learning from Incomplete Databases

Perfect cleaning is a myth. Even with best-effort repairs, many plausible datasets remain

Hidden uncertainty  $\Rightarrow$  hidden risk. A model trained on one arbitrary repair can look accurate yet flip decisions on another equally valid repair.

Needed: an explicit uncertainty framework.

- capture what is *unknown* in the data,
- propagate that uncertainty through training,
- surface it at inference time.



Practical pay-off.

- **Robustness check:** see when all admissible models agree (safe to act).
  - **Guardrail:** abstain or seek more data when predictions diverge.
- Targeted cleaning:** focus effort on the cells that actually shrink uncertainty.

# Incomplete Databases

Formalism from databases & AI to handle uncertainty by modeling all plausible data interpretations. (*Rooted in modal logic & philosophy*)

Dataset with Quality Issues

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	NULL	60K	...	8K
3	35	NULL	...	[10K, 12K]
...	...	...	...	...

Q : What is the total income?

# Possible Worlds Semantics

Inference:

- All repairs agree  $\rightarrow$  Certain answer  
 $\text{Range} \leq \tau \rightarrow$  Robust interval (e.g., [5 k – 6 k])
- Range  $> \tau \rightarrow$  Uncertain  $\rightarrow$  warn / seek more cleaning

Dataset with Quality Issues

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	NULL	60K	...	8K
3	35	NULL	...	[10K, 12K]
...	...	...	...	...

Q : What is the total income?

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	30	60K	...	8K
3	35	55K	...	7K
...	...	...	...	...

$$Q(D_1) = 6k$$

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	35	60K	...	8K
3	35	60K	...	8K
...	...	...	...	...

$$Q(D_2) = 9k$$

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	35	60K	...	8K
3	35	60K	...	8K
...	...	...	...	...

$$Q(D_3) = 5k$$

Min/Max query result across all possible database repairs.

Range consistent answers:  
 $[0.5 - 0.3]$

...

# Representing Uncertainty in Databases

**C-Tables/M-Tables:** Compactly represent multiple possible worlds using variables and conditions.

[Imieliński & Lipski, JACM 1984], [Sundarmurthy et al., ICDT 2017]

**Probabilistic Databases:** Assign probabilities to possible worlds, quantifying their likelihood.

[Suciu, Olteanu, Ré & Koch, Book 2022]

Answering queries across possible worlds is computationally expensive, often NP-hard or exponential.



# ML from Possible Repairs

## Inference

- All models ( $h_{D_i}^*$ ) concur  $\rightarrow$  **Certain** prediction (e.g., payout = 3 K)
- disagree  $\rightarrow$  **Range** prediction (e.g., payout  $\in [2 \text{ K}, 4 \text{ K}]$ )



### Dataset with Quality Issues

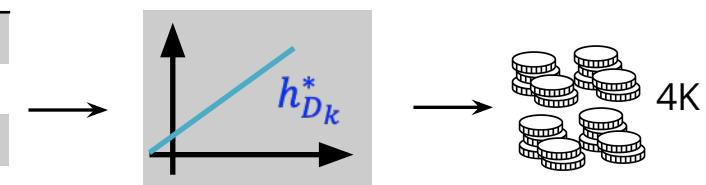
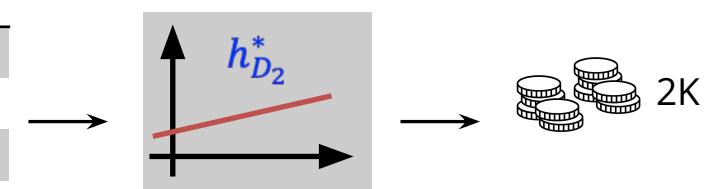
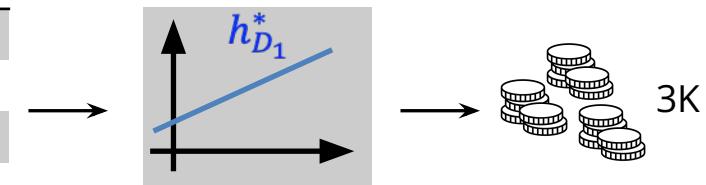
ID	Age	Income	...	Loan
1	25	50K	...	5K
2	NULL	60K	...	8K
3	35	NULL	...	[10K, 12K]
...	...	...	...	...

Machine-learning analogue of  
**Consistent Query Answering**:  
 swap the SQL query  $Q$  for a training  
 routine  $T$ —e.g., gradient descent,  
 decision-tree induction, SVM fitting.

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	30	60K	...	8K
3	35	55K	...	7K
...	...	...	...	...

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	35	60K	...	8K
3	35	60K	...	8K
...	...	...	...	...

ID	Age	Income	...	Loan
1	25	50K	...	5K
2	35	60K	...	8K
3	35	60K	...	8K
...	...	...	...	...



# KNN Classifiers over Incomplete Information

[Approach: “Certain-kNN” → returns a label only when it is guaranteed across all completions of the missing values]

## Insights:

- Missing attributes can flip k-NN labels; intersecting votes across **all** imputations yields a *guaranteed* label.

## Approach:

- Model each incomplete record as a value set (hyper-rectangle).
- Two polynomial-time tests (SS, MM) decide if a test point is “certain” without enumerating possible worlds.

## Benefits:

- 100 % precision on “certain” points – i.e., points whose prediction is certain across every imputation.**
- CPClean add-on** ranks the missing cells whose repair would turn “uncertain” points into certain ones, guiding targeted data cleaning.

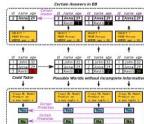
## Shortcomings:

- Guarantees apply only to **numeric-feature k-NN**

### Nearest Neighbor Classifiers over Incomplete Information: From Certain Answers to Certain Predictions

Bojan Karlaš<sup>1\*</sup>, Peng Li<sup>2</sup>, Renchi Wei<sup>1</sup>, Nisrine Merve Gürsel<sup>1</sup>, Xu Chi<sup>1</sup>, Wentao Wu<sup>1</sup>, Ce Zhang<sup>1</sup>  
<sup>1</sup>Fudan University, <sup>2</sup>Zhejiang University, <http://cpclean.cs.fudan.edu/>

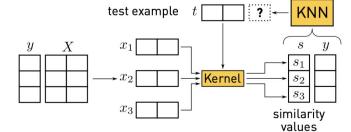
ABSTRACT  
Machine learning (ML) applications have been facing challenges due to the increasing availability of data. However, inconsistency and incomplete information are ubiquitous in real-world datasets. In this paper, we present a formal study of this impact by extending the classic k-nearest neighbor (kNN) classifier to handle incomplete data. We propose the notion of “Certain Prediction” (CP) to address the problem and propose classifiers trained on top of all possible worlds induced by the data. We also propose two polynomial-time tests (SS and MM) to determine whether a data example can be CP or not, and to count the number of possible worlds that guarantee a certain prediction. The proposed CP-based kNN classifier is able to predict a label even when a data example has CP status. The experiments show that the proposed CP-based kNN classifier is competitive with state-of-the-art classifiers.



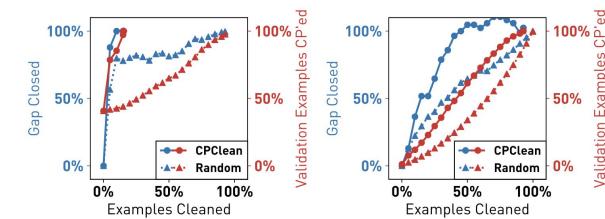
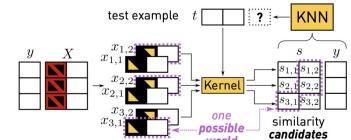
[Karlaš VLDB '20]

Karlaš, Bojan, et al. "Nearest neighbor classifiers over incomplete information: from certain answers to certain predictions." Proceedings of the VLDB Endowment 14.3 (2020): 255-267. [\[Paper\]](#)

a KNN classification over a regular training dataset

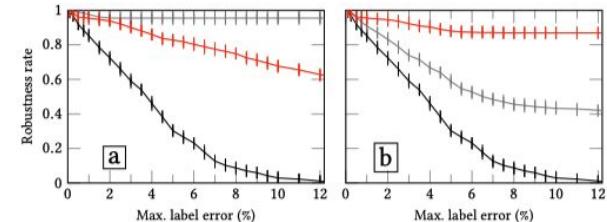
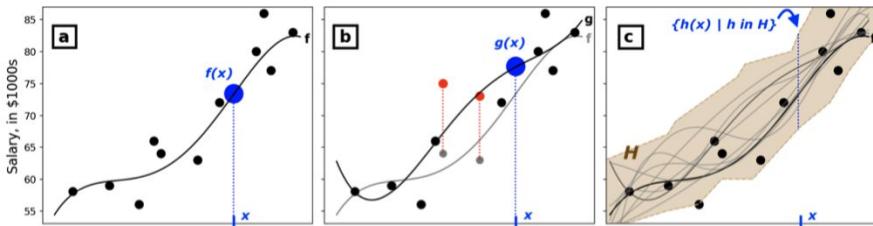


b KNN classification over a training dataset with incomplete information



# The Dataset Multiplicity Problem

[Approach: bound model risk across every dataset consistent with the errors]



## Insights:

- Introduces a risk interval: the tightest possible lower/upper bound on test error that any admissible dataset can induce for a fixed linear model.

## Approach:

- Derive closed-form formulas for the worst- and best-case hinge / logistic loss of any linear classifier under those rules, avoiding enumeration.

## Benefits:

- Gives practitioners a numeric certificate of how much reported accuracy can deteriorate.

## Shortcomings:

- Theory currently limited to linear models and label-noise rules; deep nets need looser convex relaxations.

### The Dataset Multiplicity Problem: How Unreliable Data Impacts Predictions

Anna Meyer  
ameyer@wisc.edu  
University of Wisconsin - Madison  
Madison, USA

Aws Alabargouthi  
aws@cs.wisc.edu  
University of Wisconsin - Madison  
Madison, USA

Loris D'Antoni  
loris@cs.wisc.edu  
University of Wisconsin - Madison  
Madison, USA

#### ABSTRACT

We introduce dataset multiplicity, a way to study how linear models, under certain assumptions, can fit many datasets with the same test-time predictions. The dataset multiplicity framework asks a counterfactual question of what the set of real-world models (and associated test-time predictions) can be induced by a dataset, given a hypothesis, unlabeled version of the dataset. We discuss how to use this framework to encapsulate various sources of uncertainty in datasets, including missing data, noisy features, missing feature predictors, and noisy labels or features. We show how to exactly analyze the impacts of dataset multiplicity for a specific model architecture, and how to bound the test error of a linear model.

Our empirical analysis shows that real-world datasets, under reasonable assumptions, can have many different samples whose predictions are explained by dataset multiplicity. The degree of domain-specific dataset multiplicity definition determines what samples are considered admissible. The number of times the predictions of the models trained on such data and frame this analysis as a ‘multiplicity problem’.

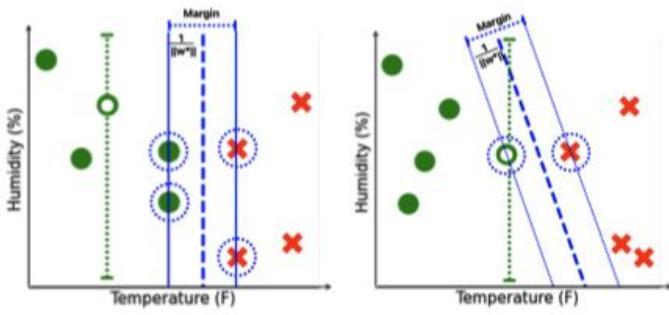
Note that there can be multiple evaluations for

[Meyer FAccT'23]

Meyer, A. P.; Alabargouthi, A.; D'Antoni, L. “The Dataset Multiplicity Problem: How Unreliable Data Impacts Predictions. [Paper]

## Certain & Approximately Certain Models for Statistical Learning

[Approach: Fast “certainty test” that lets you skip imputation whenever the missing cells don’t affect the optimum]



### Certain and Approximately Certain Models for Statistical Learning

Cheng Zhen  
Oregon State University  
Corvallis, Oregon  
zhen@oregonstate.edu

Nischal Aryal  
Oregon State University  
Corvallis, Oregon  
aryan@oregonstate.edu

Arafa Tariqzadeh  
Oregon State University  
Corvallis, Oregon  
tarizadeh@oregonstate.edu

#### ABSTRACT

Real-world data is often incomplete and contains missing values. To train accurate models over real-world datasets, users need to spend a substantial amount of time on reasoning, imputing and finding the right model for their data. In this paper, we demonstrate that it is possible to train accurate models directly from incomplete data without any manual intervention. We propose a unified approach for checking the necessity of data imputation to learn accurate models across various widely-used machine learning models. Our approach is based on theoretical guarantees to check this necessity and return accurate models. Our experiments show that our proposed algorithm can reduce the amount of time and effort needed for data imputation significantly. We also show that our proposed algorithm can reduce the number of steps of finding a minimum spanning tree, constructing an accurate model, and performing imputation.

[Zhen SIGMOD'24]

Zhen, C. et al. “Certain and Approximately Certain Models for Statistical Learning. [Paper]

### Insights:

- Not every example with missing values requires cleaning.
- If the missing cells lie in directions that do not change the model’s optimum, we can train directly on the incomplete data—with full guarantee.

### Approach:

- Provide fast algebraic tests (no world enumeration) that decide certainty for linear regression, linear SVM, and two kernel SVMs. When tests pass → output the **certain model** (exactly optimal).
- When tests fail → compute an  $\epsilon$ -certain model whose loss is within  $\epsilon$  of the global optimum.

### Benefits:

- Skips imputation for datasets that pass the test, saving cleaning effort and avoiding imputation bias.
- Same code works across several common model families.

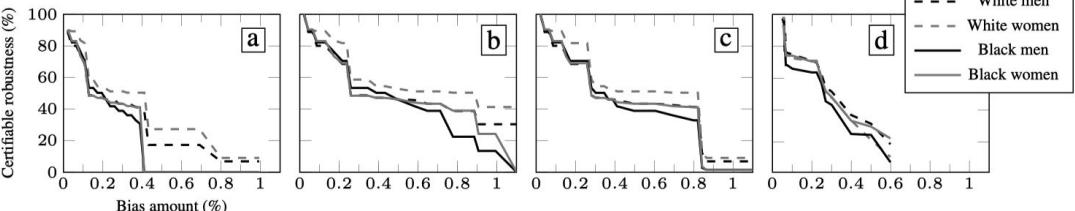
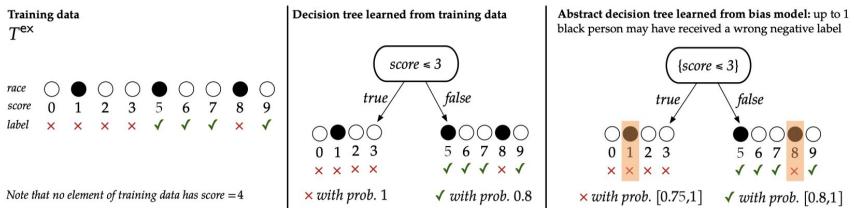
### Shortcomings:

- Certainty rarely holds under heavy missingness.  
Guarantees limited to the studied linear & kernel models; deep nets need other methods.

## Learning from Possible Repairs

# Certifying Robustness to Programmable Data Bias in Decision Trees

[Approach — ProgBiasCert: encode “tree + bias program” in SMT to prove the label never flips]



## Insights:

- Treat data bias as a **user-written program** (e.g., *age ± 2, race swap, income × 0.9–1.1*).
- A tree is **robust** if its prediction is invariant under **all** transformations allowed by that program.

## Approach:

- Translate each path of the decision tree and the bias constraints into a single SMT formula.

## Benefits:

- Exact guarantees—no sampling; works with real & categorical features and generates independently checkable proofs

## Shortcomings:

- Does not yet handle ensembles or probabilistic bias distributions.

### Certifying Robustness to Programmable Data Bias in Decision Trees

Anna P. Meyer, Aws Albarghouthi, and Lorin D’Antoni  
Department of Computer Sciences  
University of Wisconsin-Madison  
Madison, WI 53706  
(annameyer, aws, lorin)@cs.wisc.edu

#### Abstract

Datasets can be biased due to societal inequities, human biases, under-representation of minorities, etc. Our goal is to verify that models produced by a learning algorithm are *pointwise-robust* to potential dataset biases. This is a challenging problem because it entails learning models for a large, or even infinite, number of possible environments, depending on the type of bias. We propose decision-tree learning due to the interpretable nature of the models. Our approach allows programmatically specifying the model’s behavior under various transformations (e.g., adding data for a specific group, changing types of bias, and targeting bias towards a specific group). To certify robustness, we use a novel symbolic technique to evaluate the model’s behavior under all possible transformations, ensuring that each and every dataset produces the same prediction for a specific test point. We evaluate our approach on datasets that are commonly used in the fairness literature, and demonstrate our approach’s viability on a range of bias models.

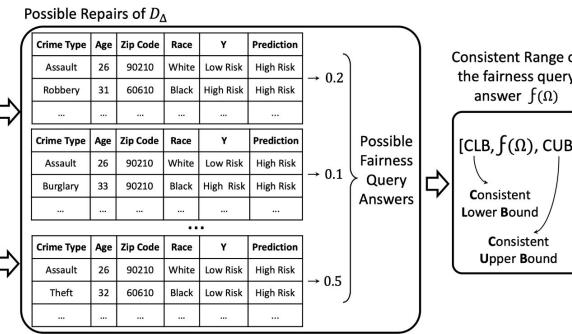
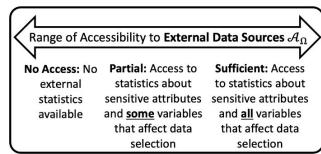
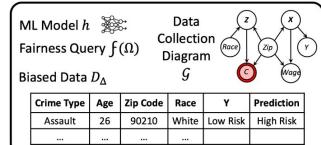
[Meyer NeurIPS'21]

Zhen, C.; Aryal, N.; Termehchy, A.; Chabada, A. S. “Certifying Robustness to Programmable Data Bias in Decision Trees.” [\[Paper\]](#)

# Consistent Range Approximation for Fair Predictive Modeling

[Approach: Fair-aware prediction ranges:  
bound each score so it stays fair under  
every repair of noisy / missing sensitive  
attributes]

## Input Components



## Consistent Range Approximation for Fair Predictive Modeling



Jiongli Zhu  
University of California,  
San Diego  
jz14@ucsd.edu

Sainyam Galhotra  
Cornell University  
sgc@cs.cornell.edu

Nazanin Sabri  
University of California,  
San Diego  
nsabri@ucsd.edu

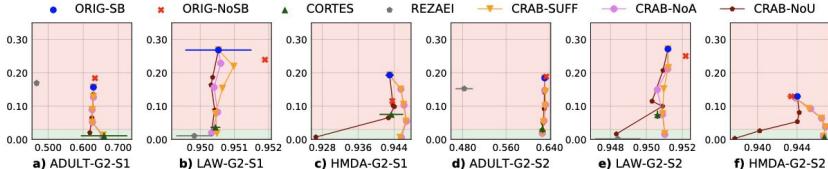
Babak Salimi  
University of California,  
San Diego  
bsalimi@ucsd.edu

### ABSTRACT

This paper proposes a novel framework for certifying the fairness of predictive models trained on biased data. It draws from query answering for incomplete data and background knowledge to formulate the problem as consistent range approximation (CRA). The framework provides a closed-form range for fairness queries for a predictive model on a target population. The framework employs background knowledge of the data collection process and biased data, working with or without limited statistics about the target population, to derive consistent ranges of fairness queries. Using CRA, the framework builds predictive models that are certified fair on the target population, regardless of the availability of external data during training. The framework's efficacy is demonstrated through evaluations on real data, showing substantial improvement over existing state-of-the-art methods.

result, deploying these models in the target population may lead to unfair and inaccurate predictions [6, 31, 35, 37, 48]. A significant issue in predictive models is **selection bias**, resulting from training data selected based on specific criteria, which can perpetuate preexisting biases. Selection bias is prevalent in sensitive areas like predictive policing, healthcare, and finance, attributed to data collection costs, historical discrimination, and biases in data collection [1, 28, 43]. In law enforcement, the data is biased as it is gathered exclusively from police interactions, which are influenced by the sociocultural traits of the officers [28, 43]. Similarly, in healthcare, selection has occurred when data is relied upon from hospitals, which are more likely to be geographically positive, leading to disproportionate effects on racial, ethnic, and gender minorities due to healthcare access [2, 16, 65, 88].

Example 1.1. Consider the dataset in Table 1, which represents



## Insights:

- With selection bias we don't know the target-population fairness.
- Treat fairness evaluation as a **query over incomplete data**; answer with a *range* that is guaranteed to contain the truth.

## Approach:

- Derive a closed-form range for fairness aggregates.
- Train a classifier that minimises risk while keeping the worst-case value inside the acceptable fairness range.

## Benefits:

- Certifies fairness without unbiased samples; needs only the biased data + background knowledge.

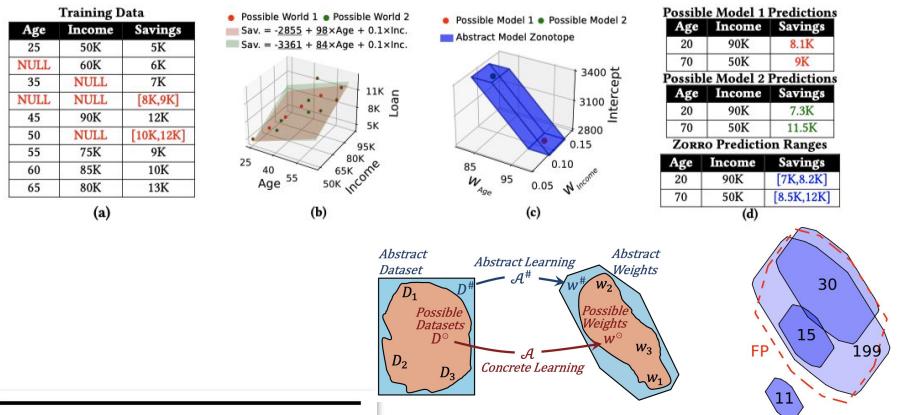
## Shortcomings:

- Relies on correct causal diagram; ranges may be wide if knowledge is weak.

## Learning from Possible Repairs

# Learning from Uncertain Data: From Possible Worlds to Possible Models

[Approach: Abstract interpretation + zonotopes: train once on a single convex polytope that encodes every possible repair



### Learning from Uncertain Data: From Possible Worlds to Possible Models

Jiongli Zhu<sup>1</sup> Su Feng<sup>2</sup> Boris Glavic<sup>3</sup> Babak Salimi<sup>1</sup>

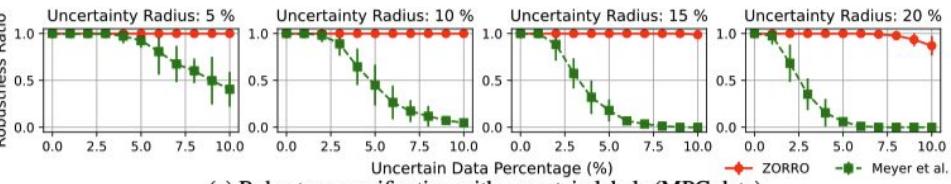
<sup>1</sup>University of California, San Diego    <sup>2</sup>Nanjing Tech University    <sup>3</sup>University of Illinois, Chicago

#### Abstract

We introduce an efficient method for learning linear models from uncertain data, where uncertainty is represented as a set of possible variations in the data, leading to predictive multiplicity. Our approach leverages abstract interpretation and zonotopes, a type of convex polytope, to compactly represent these dataset variations. Unlike the symbolic execution of gradient descent on all possible weight vectors simultaneously, our development techniques ensure that this process converges to a fixed point and derive closed-form solutions for this fixed point. Our method provides sound over-approximations of all possible optimal models and viable prediction ranges. We validate the theoretical guarantees through theoretical and empirical analysis, highlighting its potential to reason about model and prediction uncertainty due to data quality issues in training data.

[Zhu NeurIPS'24]

Zhu, J.; Feng, S.; Glavic, B.; Salimi, B. "Learning from Uncertain Data: From Possible Worlds to Possible Models. [Paper]



## Insights:

- Zonotope = all repairs in a compact affine form.
- Training on the zonotope gives one weight-box that subsumes every per-repair model.

## Approach:

- Map each uncertain record to an affine form; the full dataset becomes **one zonotope**. Run gradient descent **symbolically**. Output is a convex box of model weights; any concrete repair yields weights inside this box.

## Benefits:

- **Guaranteed intervals for weights & predictions—true model always inside.**

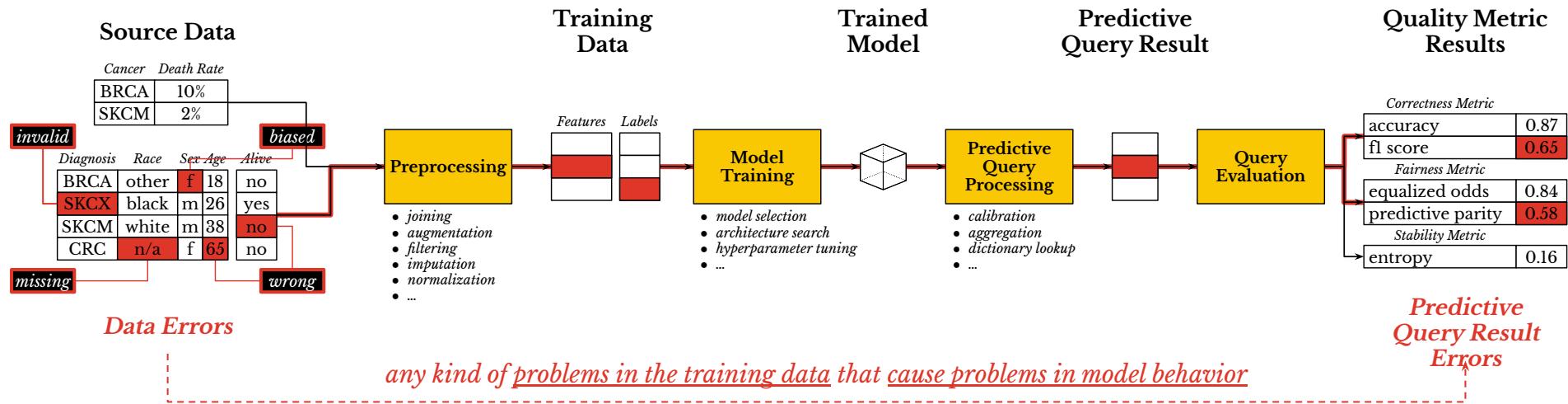
## Shortcomings:

- Supports linear models only.

## Key Takeaways of Part III

- Residual data uncertainty is inevitable. Cleaning produces at best one plausible version; we must reason over the space of possibilities.
- Guarantee  $\leftrightarrow$  coverage trade-off. Certainty methods (Certain-kNN, CRA, ProgBiasCert) give perfect precision or fairness—but may abstain widely.
- Targeted cleaning beats blanket imputation. Algorithms like CPClean and OTClean identify the few cells whose repair actually widens certified coverage.
- Model-side defences matter. Dataset Multiplicity, Certain/Approx-Certain Models, and Zorro show how to train / audit over the whole uncertainty set—returning intervals, ensembles, or risk bounds.
- Certification  $>$  best-guess. When stakes are high, prefer guaranteed ranges or proofs of robustness to a single point prediction from a guessed-clean dataset.
- Open frontiers: extend guarantees to deep nets & categorical features, tighten bounds under heavy missingness, and scale zonotope / SMT methods to larger models.

# Conclusion: How should we navigate data errors?



Error Detection:  
Compute Data Importance

ML Pipeline Debugging:  
Leverage Data Provenance

Learning from Uncertain Data:  
Apply Possible Worlds Semantics

Thank you!