SIGMOD
PODS
BERLIN
GERMANY
2025

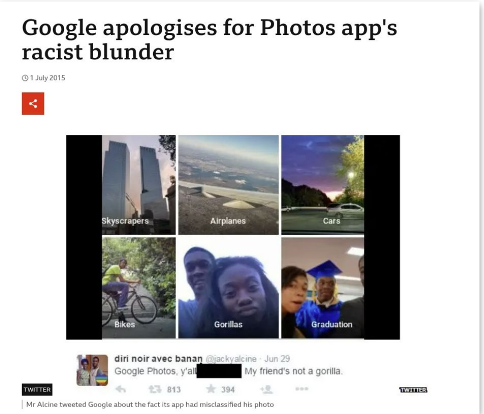
Navigating Data Errors in Machine Learning Pipelines: Identify, Debug, and Learn

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

navigating-data-errors.github.io

Background: ML apps often behave in unintended ways

Wrong

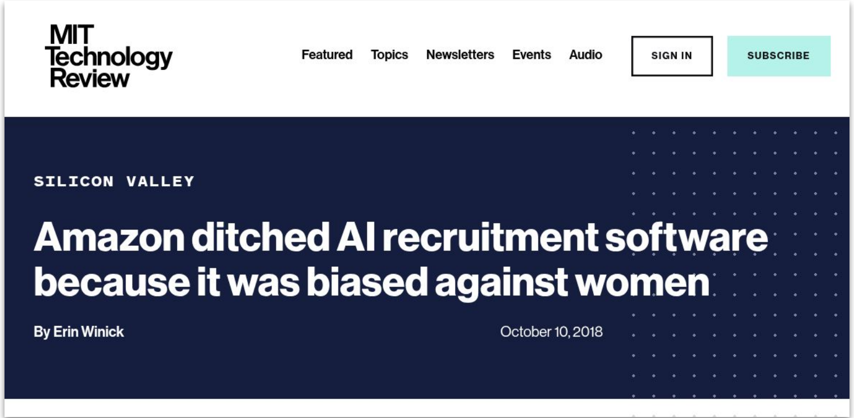

Google apologises for Photos app's racist blunder

1 July 2015

TWITTER
Mr Alcine tweeted Google about the fact its app had misclassified his photo

Source: BBC

Biased



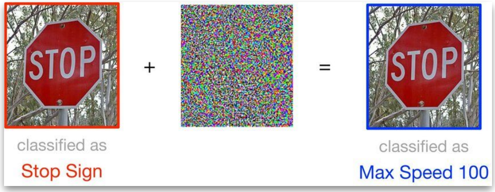
SILICON VALLEY

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.

This article is more than **1 year old**

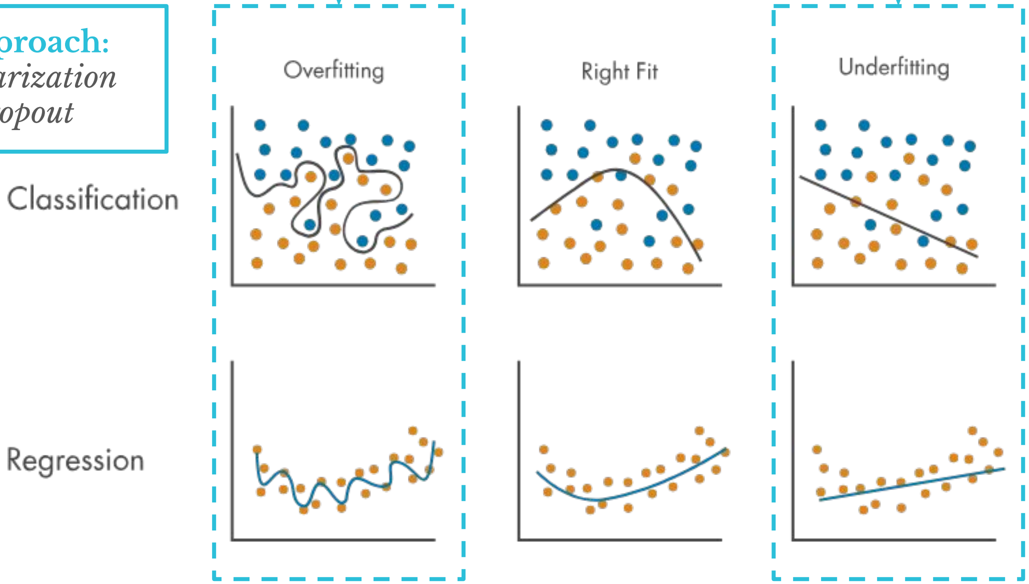
Tesla Autopilot feature was involved in 13 fatal crashes, US regulator says

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

Example Approach:
Increase regularization by applying dropout

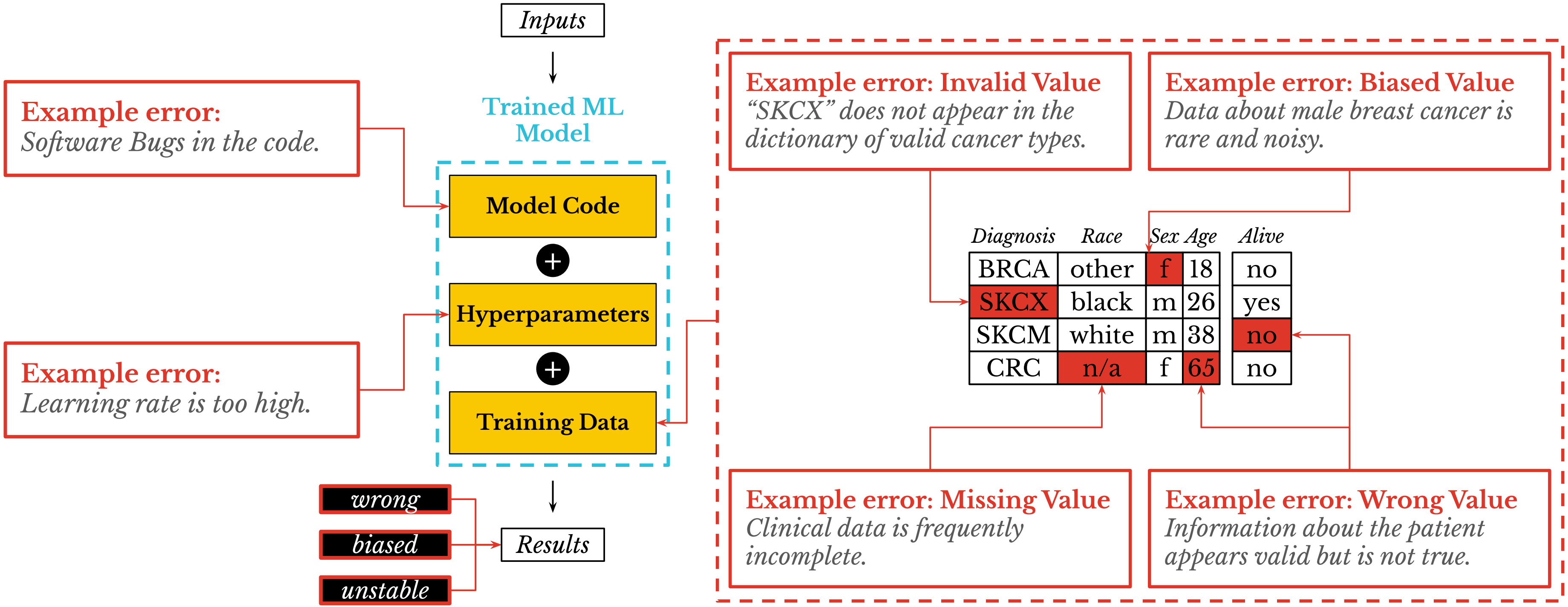


Example Approach:
Improve model capacity by increasing the number of parameters

Source: MathWorks

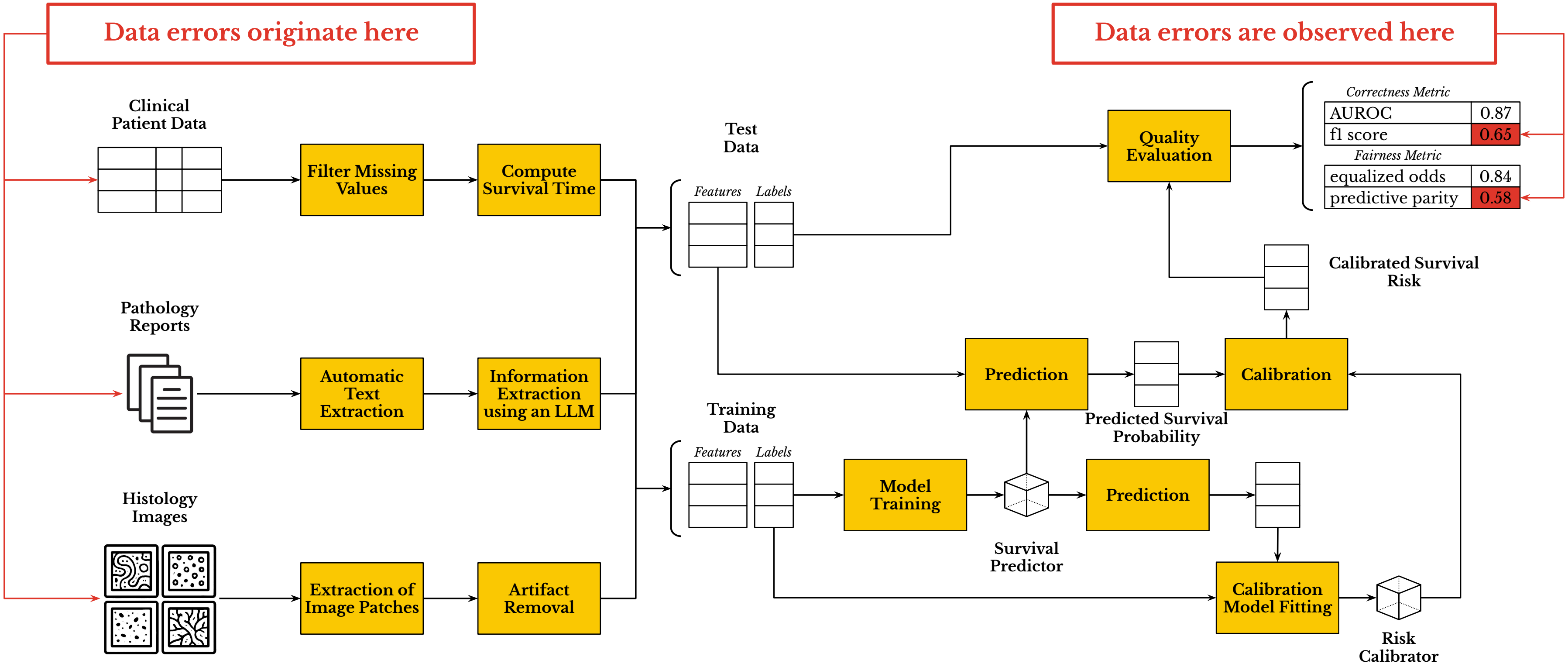
Problem: *This is only one piece of the puzzle!*

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



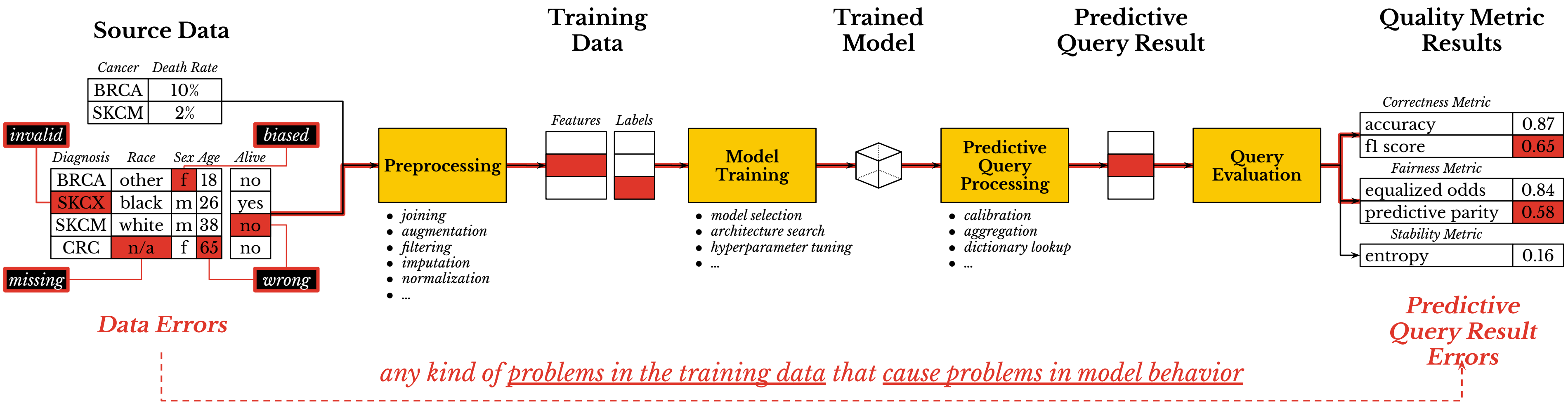
Challenge 2: *Can we trace data errors as they pass through the pipeline?*

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**.

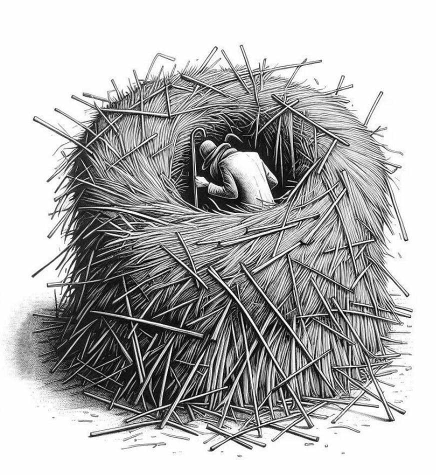
(2) ML pipelines are data processing pipelines. Models are learned data transformation operators. Many systems have been developed, but most practitioners still rely on **rudimentary scripts for crunching data**.

(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



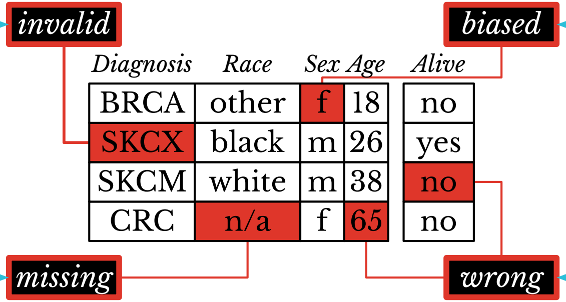
1) Introducing the Concept of Data Importance
2) Examples of Data Attribution Functions
3) Case Study of Shapley Value as a Measure of Importance
4) Applications of Data Importance

How can we identify data errors?

Trivial

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

Solution approach:
Check if the value is marked as missing.



Not So Trivial

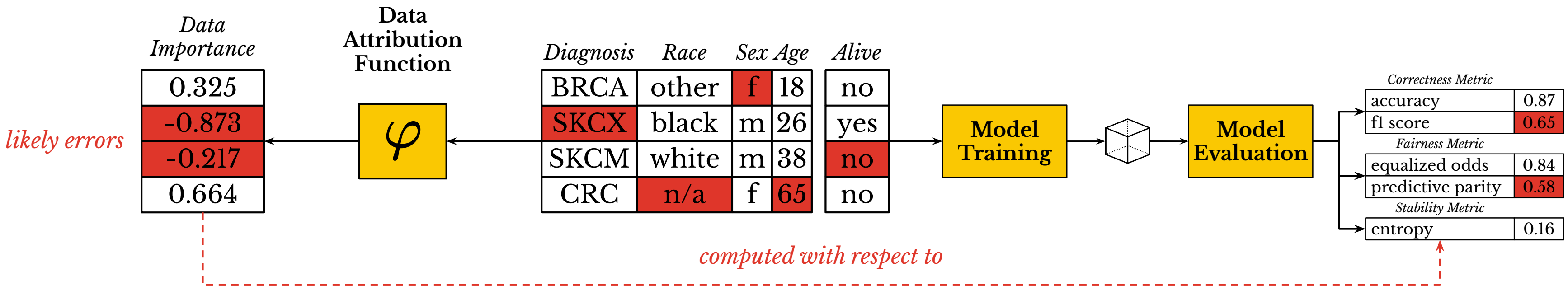
Solution approach:
Measure the impact of the value on model quality.

How do we measure this?
That is the main topic of this part of the tutorial.

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

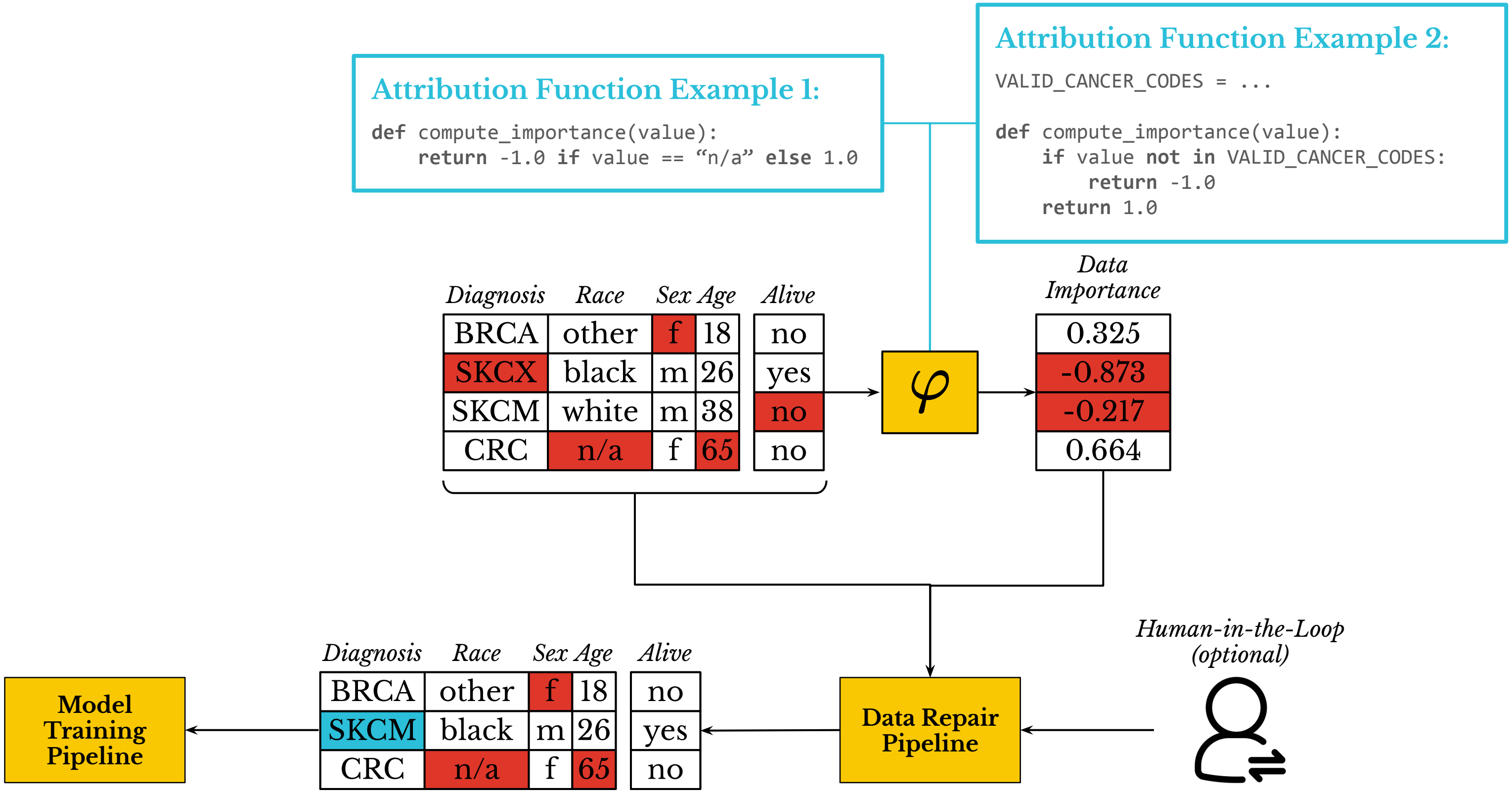
Challenge: *Can we define a unified way to think about identifying data errors?*

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

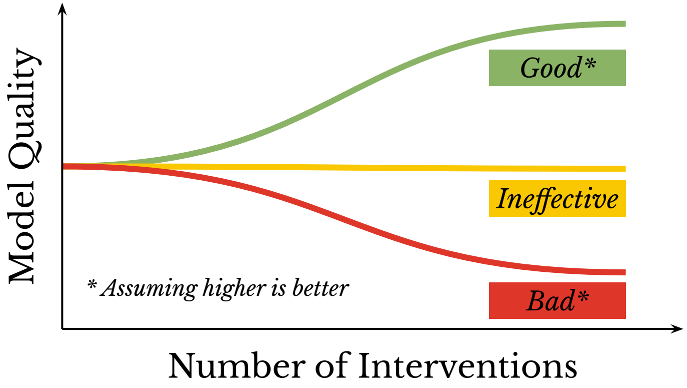
Which model quality metric do we care about improving?

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

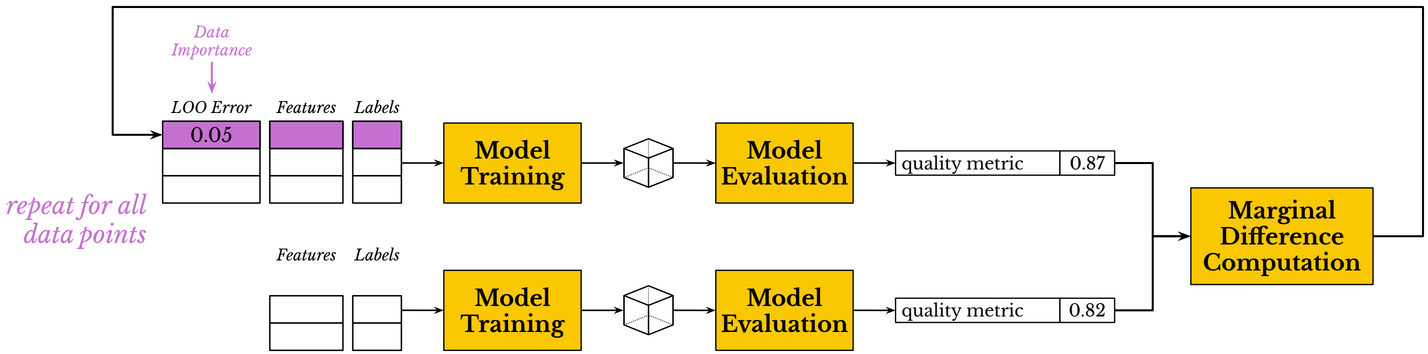
2) Examples of Data Attribution Functions

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:

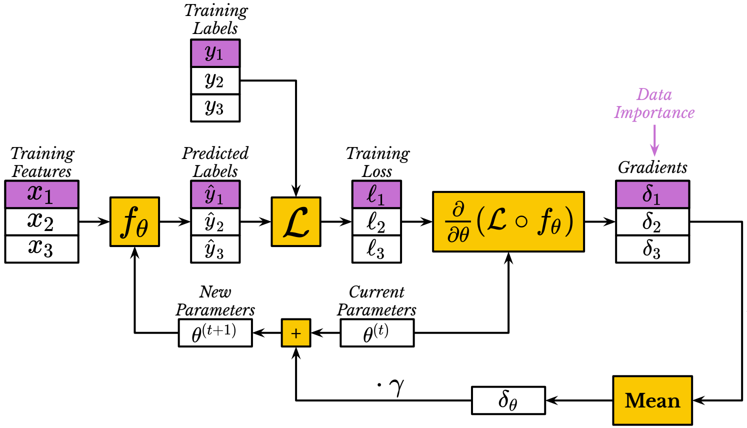
- Very simple to implement.

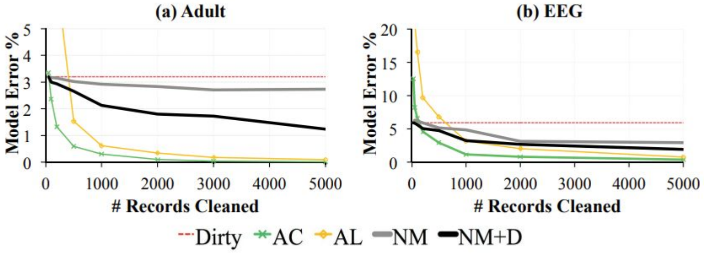
Shortcomings:

- 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

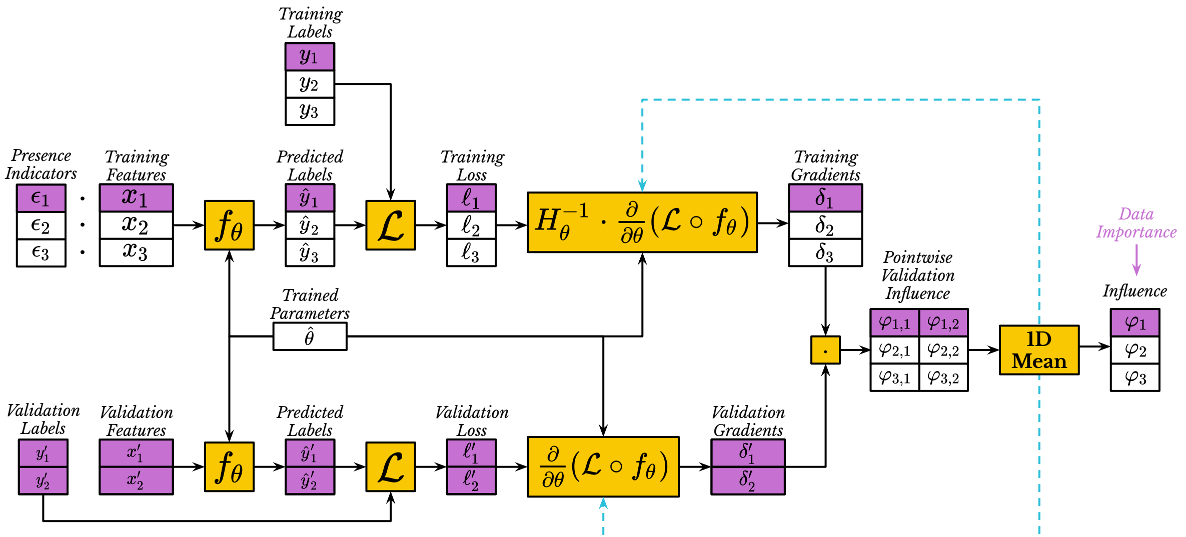
Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J. Franklin, Ken Goldberg

ABSTRACT

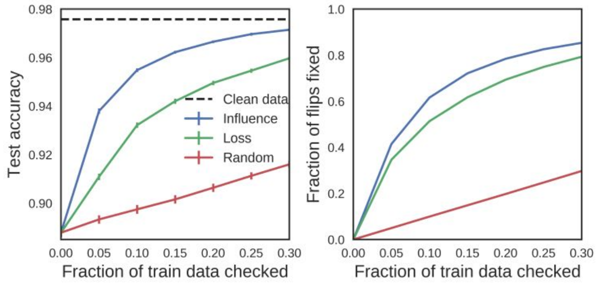
[Krishnan VLDB'16]
Krishnan, Sanjay, et al. "Activeclean: Interactive data cleaning for statistical modeling." Proceedings of the VLDB Endowment 9.12 (2016): 948-959.
[Paper][Website]

Influence Function

[Approach: **Marginal Contribution, Gradient**]



$$\varphi_{i,j} = \frac{\partial \mathcal{L}(x'_j, y'_j, \hat{\theta})}{\partial \epsilon_i}\bigg|_{\epsilon_i=0} = \frac{\partial(\mathcal{L}\circ f_\theta)(x'_j, y'_j, \hat{\theta})}{\partial\theta} \cdot H_\theta^{-1} \frac{\partial(\mathcal{L}\circ f_\theta)(x_i, y_i, \hat{\theta})}{\partial\theta}$$



Insights:

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

Approach:

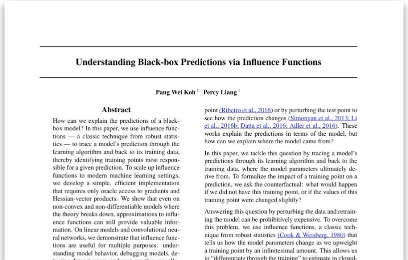
- Introduce presence indicator variables ϵ for each data point and compute the gradient w.r.t. ϵ.

Benefits:

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

Shortcomings:

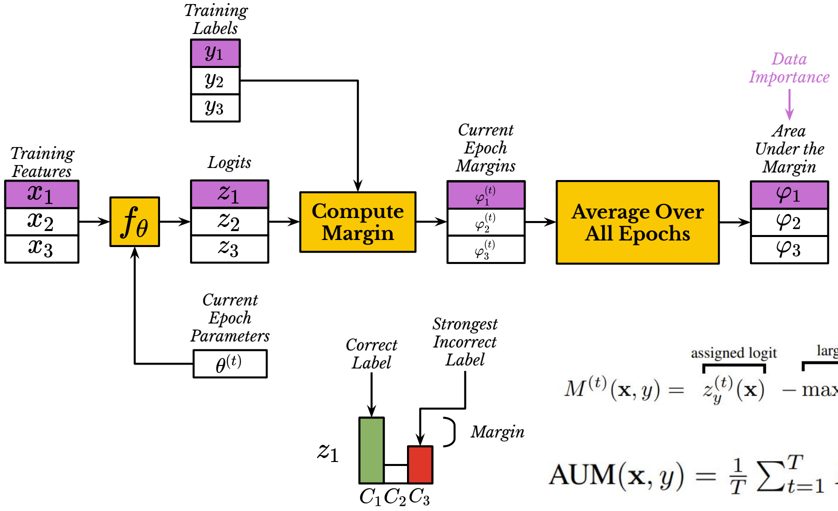
- Treats data points independently.



[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]

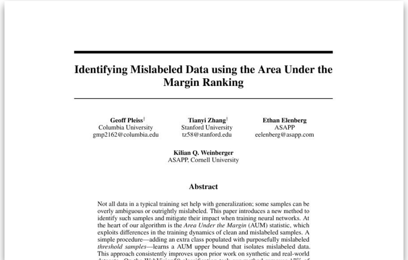
Area Under the Margin

[Approach: **Uncertainty Analysis**]



$$M^{(t)}(\mathbf{x}, y) = \overbrace{z_y^{(t)}(\mathbf{x})}^{\text{assigned logit}} - \overbrace{\max_{i \neq y} z_i^{(t)}(\mathbf{x})}^{\text{largest other logit}}$$

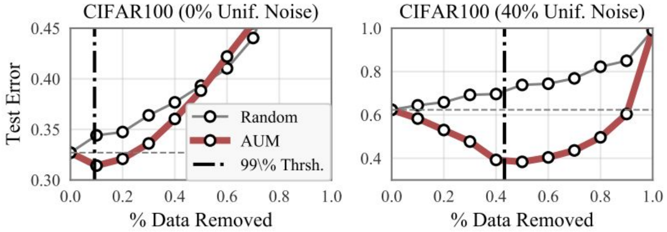
$$\mathrm{AUM}(\mathbf{x}, y) = \frac{1}{T} \sum_{t=1}^{T} M^{(t)}(\mathbf{x}, y)$$


Identifying Mislabeled Data using the Area Under the Margin Ranking

Geoff Pleiss[†] Columbia University gmp2162@columbia.edu
Tianyi Zhang[†] Stanford University tz58@stanford.edu
Ethan Elenberg ASAPP eelenberg@asapp.com
Kilian Q. Weinberger ASAPP, Cornell University

Abstract

Not all data in a typical training set help with generalization; some samples can be overly ambiguous or outrightly mislabeled. This paper introduces a new method to identify such samples and mitigate their impact when training neural networks. At the heart of our algorithm is the *Area Under the Margin* (AUM) statistic, which exploits differences in the training dynamics of clean and mislabeled samples. A simple procedure—adding an extra class populated with purposefully mislabeled *threshold samples*—learns a AUM upper bound that isolates mislabeled data. This approach consistently improves upon prior work on synthetic and real-world



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:

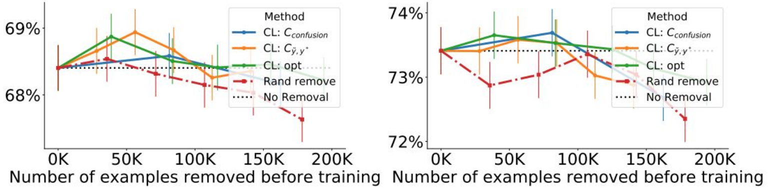
- Focuses only on label noise.

[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]

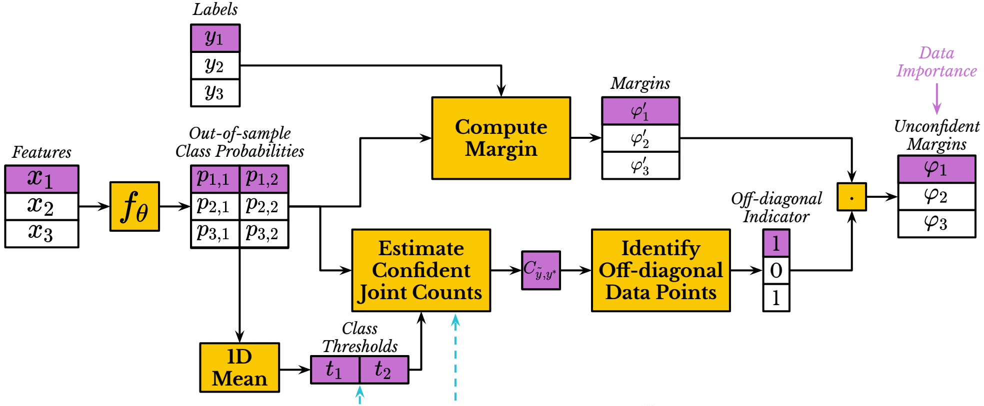
Examples of Data Attribution Functions

Unconfident Margins

[Approach: **Uncertainty Analysis**]



(a) ResNet18 Validation Accuracy   (b) ResNet50 Validation Accuracy



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

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

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

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:

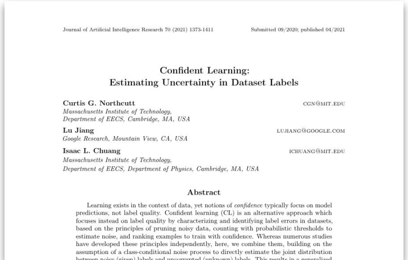
- Identify likely mislabeled data points and assign negative importance using the margin. Remaining data points get zero importance.

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.



Journal of Artificial Intelligence Research 70 (2021) 1373-1411   Submitted 09/2020; published 04/2021

Confident Learning:
Estimating Uncertainty in Dataset Labels

Curtis G. Northcutt   CGN@MIT.EDU
Massachusetts Institute of Technology, Department of EECS, Cambridge, MA, USA
Lu Jiang   LUJIANG@GOOGLE.COM
Google Research, Mountain View, CA, USA
Isaac L. Chuang   ICHUANG@MIT.EDU
Massachusetts Institute of Technology, Department of EECS, Department of Physics, Cambridge, MA, USA

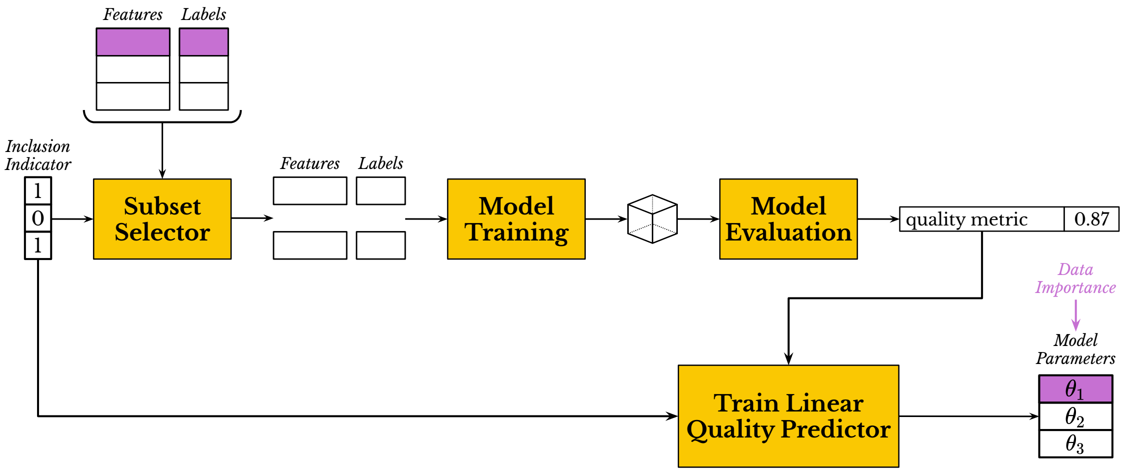
Abstract

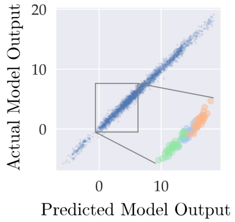
Learning exists in the context of data, yet notions of *confidence* typically focus on model predictions, not label quality. Confident learning (CL) is an alternative approach which focuses instead on label quality by characterizing and identifying label errors in datasets, based on the principles of pruning noisy data, counting with probabilistic thresholds to estimate noise, and ranking examples to train with confidence. Whereas numerous studies have developed these principles independently, here, we combine them, building on the assumption of a class-conditional noise process to directly estimate the joint distribution

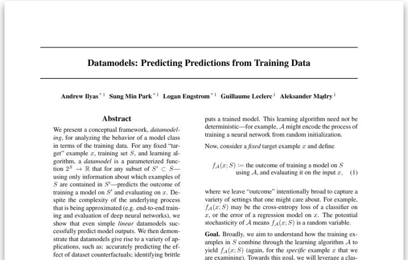
[Northcutt JAIR '21]
Northcutt, Curtis, Lu Jiang, and Isaac Chuang. "Confident learning: Estimating uncertainty in dataset labels." Journal of Artificial Intelligence Research 70 (2021): 1373-1411. [Paper][Blog][Code]

Model Training Outcome

[Approach: **Surrogate Data Model**]







Datamodels: Predicting Predictions from Training Data

Andrew Ilyas [*1] Sung Min Park [*1] Logan Engstrom [*1] Guillaume Leclerc [1] Aleksander Mądry [1]

Abstract

We present a conceptual framework, *datamodeling*, for analyzing the behavior of a model class in terms of the training data. For any fixed "target" example x, training set S, and learning algorithm, a *datamodel* is a parameterized function $2^S \to \mathbb{R}$ that for any subset of $S' \subset S$—using only information about which examples of S are contained in S'—predicts the outcome of training a model on S' and evaluating on x. Despite the complexity of the underlying process that is being approximated (e.g. end-to-end training and evaluation of deep neural networks), we show that even simple *linear* datamodels successfully predict model outputs. We then demonstrate that datamodels give rise to a variety of applications, such as: accurately predicting the effect of dataset counterfactuals; identifying brittle

puts a trained model. This learning algorithm need not be deterministic—for example, $\mathcal{A}$ might encode the process of training a neural network from random initialization.

Now, consider a *fixed* target example x and define

$$f_{\mathcal{A}}(x;S) := \text{the outcome of training a model on } S \text{ using } \mathcal{A}\text{, and evaluating it on the input } x, \quad (1)$$

where we leave "outcome" intentionally broad to capture a variety of settings that one might care about. For example, $f_{\mathcal{A}}(x;S)$ may be the cross-entropy loss of a classifier on x, or the error of a regression model on x. The potential stochasticity of $\mathcal{A}$ means $f_{\mathcal{A}}(x;S)$ is a random variable.

Goal. Broadly, we aim to understand how the training examples in S combine through the learning algorithm $\mathcal{A}$ to yield $f_{\mathcal{A}}(x;S)$ (again, for the *specific* example x that we are examining). Towards this goal, we will leverage a clas-

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.

[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]

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.

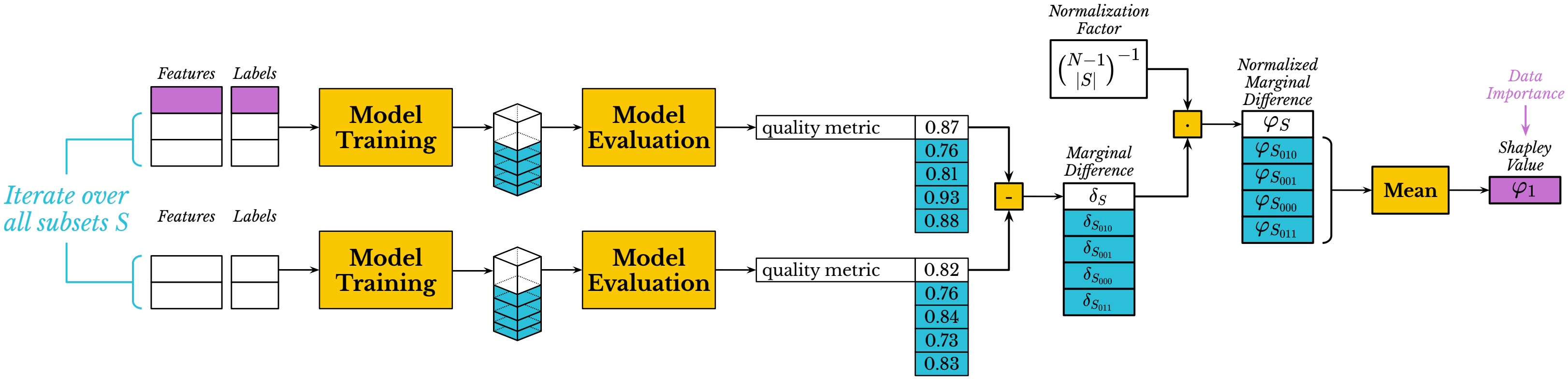
Approach

We should measure marginal contribution over all subsets.

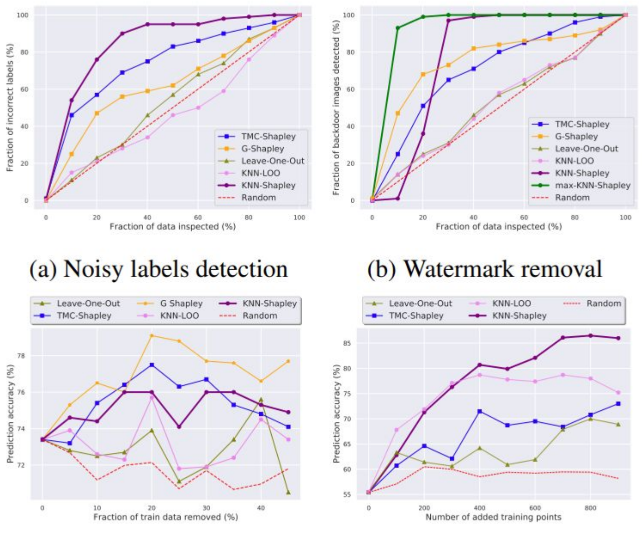
Shapley value

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

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



Effectiveness at Data Debugging



(a) Noisy labels detection (b) Watermark removal

(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
*K*NN-Shapley	**31.70%** → **47.00%**	**23.35%** → **29.80%**
*K*NN-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 this 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, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, Dawn Song

Abstract

1. Introduction

[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 <u>exponential</u>, making it intractable to compute the exact Shapley value for an arbitrary model.

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

Approximation: Monte Carlo Sampling

Challenge

Computing Shapley values is intractable.

Insight

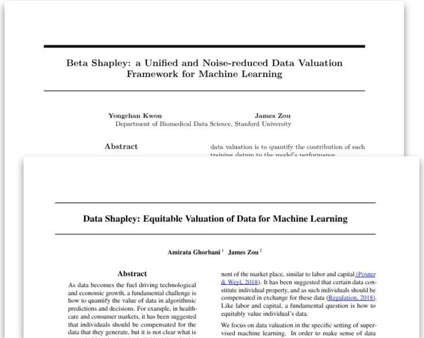
Since Shapley value can be seen as a statistic over exponentially many subsets, we can estimate it using Monte Carlo sampling.

Approach

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

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

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



Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning

Yongchan Kwon James Zou
Department of Biomedical Data Science, Stanford University

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 [1] James Zou [2]

Abstract

As data becomes the fuel driving technological and economic growth, a fundamental challenge is how to quantify the value of data in algorithmic predictions and decisions. For example, in health-care and consumer markets, it has been suggested that individuals should be compensated for the data that they generate, but it is not clear what is

nent of the market place, similar to labor and capital (Posner & Weyl, 2018). It has been suggested that certain data constitute individual property, and as such individuals should be compensated in exchange for these data (Regulation, 2018). Like labor and capital, a fundamental question is how to equitably value individual's data.

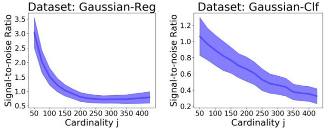
We focus on data valuation in the specific setting of supervised machine learning. In order to make sense of data

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.

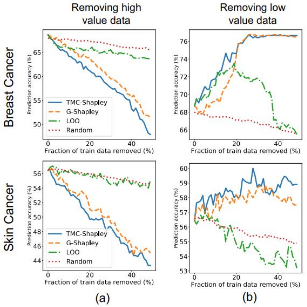


Approach

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.

[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]

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.

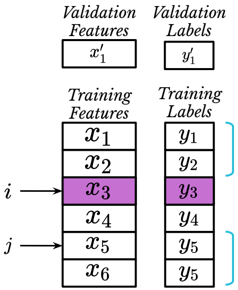
Approach

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

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.



Observation 3:
If the subset S contains these data points, the data point i will not be the top-1.

Observation 4:
If data point j is going to become the top-1 after i is removed, all data points above it cannot be included in S, while the ones below may or may not be included in S.

Starting point: Shapley value definition

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

Observation 2:
If data point i is not in the top-1, this term will be zero.

Dynamic Programming

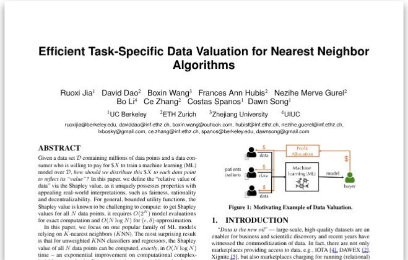
$$\varphi_i(t) = \frac{1}{N} \sum_{j=i+1}^{N} \sum_{a=1}^{n-j} \binom{N-1}{a}^{-1} \left(u(\{i\}) - u(\{j\})\right) \binom{N-j}{a}$$

Final Simplification

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

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

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



Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms

Ruoxi Jia[1] David Dao[2] Boxin Wang[3] Frances Ann Hubis[2] Nezihe Merve Gurel[2]
Bo Li[4] Ce Zhang[2] Costas Spanos[1] Dawn Song[1]
[1]UC Berkeley [2]ETH Zurich [3]Zhejiang University [4]UIUC
ruoxijia@berkeley.edu, daviddao@inf.ethz.ch, boxin.wang@outlook.com, hubisf@inf.ethz.ch, nezihe.guerel@inf.ethz.ch, lxbosky@gmail.com, ce.zhang@inf.ethz.ch, spanos@berkeley.edu, dawnsong@gmail.com

ABSTRACT

Given a data set $\mathcal{D}$ containing millions of data points and a data consumer who is willing to pay for \$X to train a machine learning (ML) model over $\mathcal{D}$, *how should we distribute this \$X to each data point to reflect its "value"?* In this paper, we define the "relative value of data" via the Shapley value, as it uniquely possesses properties with appealing real-world interpretations, such as fairness, rationality and decentralizability. For general, bounded utility functions, the Shapley value is known to be challenging to compute: to get Shapley values for all N data points, it requires $O(2^N)$ model evaluations for exact computation and $O(N \log N)$ for (ϵ, δ)-approximation.

In this paper, we focus on one popular family of ML models relying on K-nearest neighbors (KNN). The most surprising result is that for unweighted KNN classifiers and regressors, the Shapley value of all N data points can be computed, *exactly*, in $O(N \log N)$ time – an exponential improvement on computational complex-

Figure 1: Motivating Example of Data Valuation.

1. INTRODUCTION

"Data is the new oil" — large-scale, high-quality datasets are an enabler for business and scientific discovery and recent years have witnessed the commoditization of data. In fact, there are not only marketplaces providing access to data, e.g., IOTA [4], DAWEX [2], Xignite [5], but also marketplaces charging for running (relational)

[Jia VLDB '19]
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]

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.

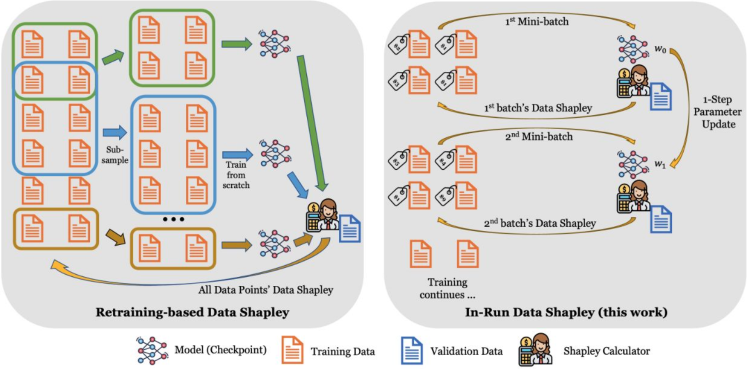
<u>*Redefined "local utility function" of subset S of a single SGD minibatch:*</u>

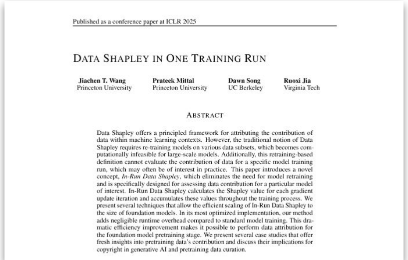
$$U^{(t)}(S; z^{(\mathrm{val})}) := \underbrace{\ell(\widetilde{w}_{t+1}(S), z^{(\mathrm{val})})}_{\text{Model updated only using data from } S} - \underbrace{\ell(w_t, z^{(\mathrm{val})})}_{\text{Model at SGD step } t}$$

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

<u>*Redefined "global utility function" of subset S over the entire SGD run:*</u>

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





Published as a conference paper at ICLR 2025

DATA SHAPLEY IN ONE TRAINING RUN

Jiachen T. Wang Princeton University · **Prateek Mittal** Princeton University · **Dawn Song** UC Berkeley · **Ruoxi Jia** Virginia Tech

ABSTRACT

Data Shapley offers a principled framework for attributing the contribution of data within machine learning contexts. However, the traditional notion of Data Shapley requires re-training models on various data subsets, which becomes computationally infeasible for large-scale models. Additionally, this retraining-based definition cannot evaluate the contribution of data for a specific model training run, which may often be of interest in practice. This paper introduces a novel concept, *In-Run Data Shapley*, which eliminates the need for model retraining and is specifically designed for assessing data contribution for a particular model of interest. In-Run Data Shapley calculates the Shapley value for each gradient update iteration and accumulates these values throughout the training process. We present several techniques that allow the efficient scaling of In-Run Data Shapley to the size of foundation models. In its most optimized implementation, our method adds negligible runtime overhead compared to standard model training. This dramatic efficiency improvement makes it possible to perform data attribution for the foundation model pretraining stage. We present several case studies that offer fresh insights into pretraining data's contribution and discuss their implications for copyright in generative AI and pretraining data curation.

[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.

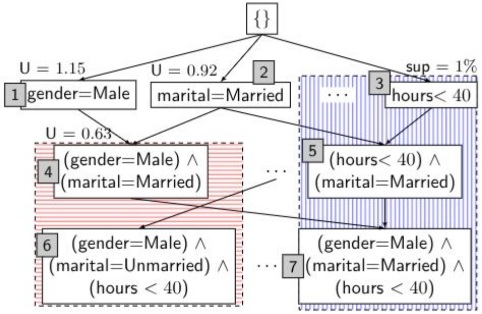
Approach

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.

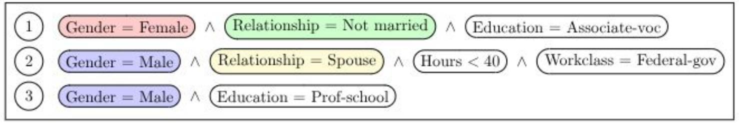
Data points ordered by importance

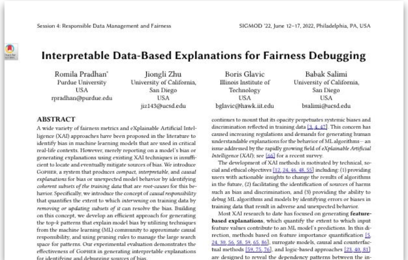
age	education	marital	...	gender	income
39	Bachelors	Never-married	...	Male	≤50K
53	11th	Never-married	...	Male	≤50K
28	Bachelors	Married-civ-spouse	...	Female	≤50K
37	Masters	Married-civ-spouse	...	Female	≤50K

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



Combinations of predicates that explain model behavior




Session 4: Responsible Data Management and Fairness SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

Interpretable Data-Based Explanations for Fairness Debugging

Romila Pradhan, Purdue University, USA, rpradhan@purdue.edu
Jiongli Zhu, University of California, San Diego, USA, jiz143@ucsd.edu
Boris Glavic, Illinois Institute of Technology, USA, bglavic@hawk.iit.edu
Babak Salimi, University of California, San Diego, USA, bsalimi@ucsd.edu

ABSTRACT

[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]

Debugging the LLM Retrieval Corpus

Challenge

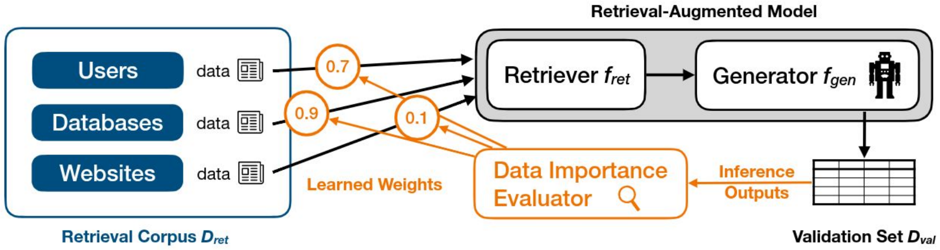
Retrieval augmented generation (RAG) is a widely used technique for providing pre-trained large language models (LLMs) with task-specific context. Data errors in the retrieval corpus have a negative impact on model quality.

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

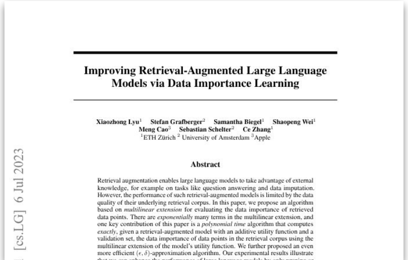
Define a data attribution function that will compute the importance of data points in the retrieval corpus. Use this to identify and debug data errors.



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

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

Dataset	GPT-JT (6B)	GPT-JT (6B) w/ retrival				GPT-3.5 (175B)
		Vanilla	+LOO	+Reweight	+Prune	
Buy	0.102	0.789	0.808	**0.815**	0.813	0.764
Restaurant	0.030	0.746	0.756	0.760	**0.761**	0.463


[cs.LG] 6 Jul 2023

Improving Retrieval-Augmented Large Language Models via Data Importance Learning

Xiaozhong Lyu[1] Stefan Grafberger[2] Samantha Biegel[1] Shaopeng Wei[1]
Meng Cao[3] Sebastian Schelter[2] Ce Zhang[1]
[1]ETH Zürich [2] University of Amsterdam [3]Apple

Abstract

Retrieval augmentation enables large language models to take advantage of external knowledge, for example on tasks like question answering and data imputation. However, the performance of such retrieval-augmented models is limited by the data quality of their underlying retrieval corpus. In this paper, we propose an algorithm based on *multilinear extension* for evaluating the data importance of retrieved data points. There are *exponentially* many terms in the multilinear extension, and one key contribution of this paper is a *polynomial time* algorithm that computes *exactly*, given a retrieval-augmented model with an additive utility function and a validation set, the data importance of data points in the retrieval corpus using the multilinear extension of the model's utility function. We further proposed an even more efficient (ϵ, δ)-approximation algorithm. Our experimental results illustrate

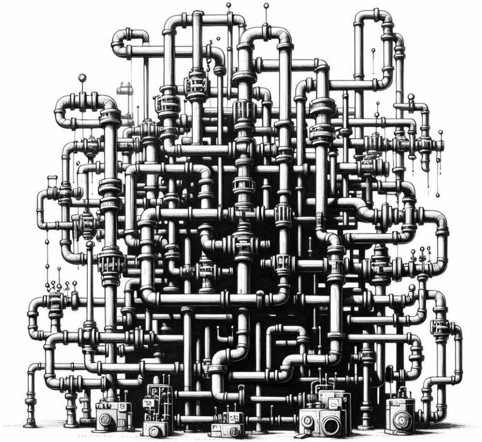
[Lyu arXiv '23]
Lyu, Xiaozhong, et al. "Improving retrieval-augmented large language models via data importance learning." arXiv preprint arXiv:2307.03027 (2023). [Paper] [Code]

Key Takeaways of Part I

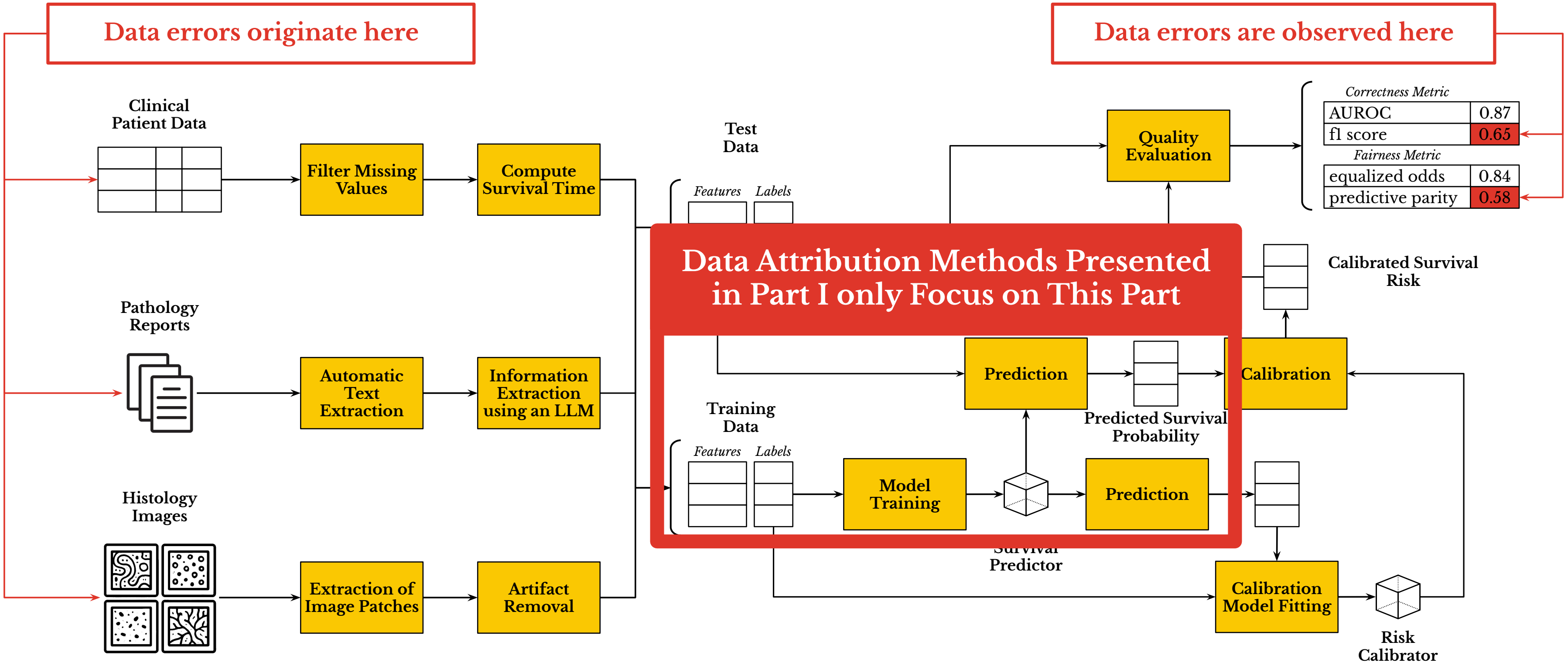
- Data attribution is a useful powerful framework for approaching the problem of data error detection.
- There are many existing data attribution methods with various strengths and shortcomings.
- The most powerful methods face scalability issues that have been tackled by existing research with many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

Part II:
Data Debugging in ML Pipelines

Sebastian Schelter



Gap between Attribution Methods and ML Pipelines



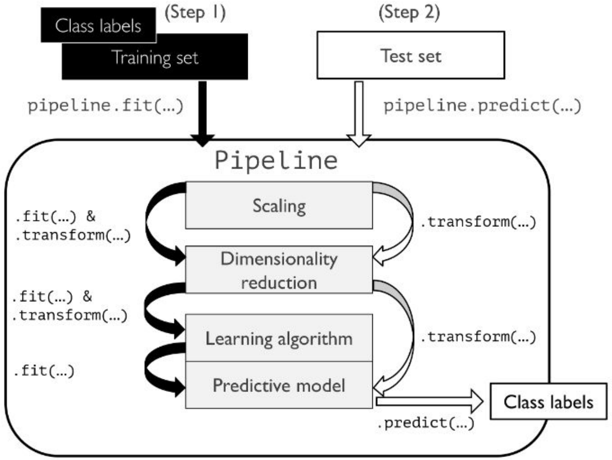
Challenge: *How should we debug ML pipelines?*

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 feature encoding 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*



Source: https://vitalflux.com/sklearn-machine-learning-pipeline-python-example/

Journal of Machine Learning Research 12 (2011) 2825-2830 Submitted 3/11; Revised 8/11; Published 10/11

Scikit-learn: Machine Learning in Python

Fabian Pedregosa FABIAN.PEDREGOSA@INRIA.FR
Gaël Varoquaux GAEL.VAROQUAUX@NORMALESUP.ORG
Alexandre Gramfort ALEXANDRE.GRAMFORT@INRIA.FR
Vincent Michel VINCENT.MICHEL@LOGILAB.FR
Bertrand Thirion BERTRAND.THIRION@INRIA.FR
Parietal, INRIA Saclay
Neurospin, Bât 145, CEA Saclay
91191 Gif sur Yvette – France

Olivier Grisel OLIVIER.GRISEL@ENSTA.FR
Nuxeo
20 rue Soleillet
75 020 Paris – France

Mathieu Blondel MBLONDEL@AI.CS.KOBE-U.AC.JP
Kobe University
1-1 Rokkodai, Nada
Kobe 657-8501 – Japan

Peter Prettenhofer PETER.PRETTENHOFER@GMAIL.COM
Bauhaus-Universität Weimar
Bauhausstr. 11
99421 Weimar – Germany

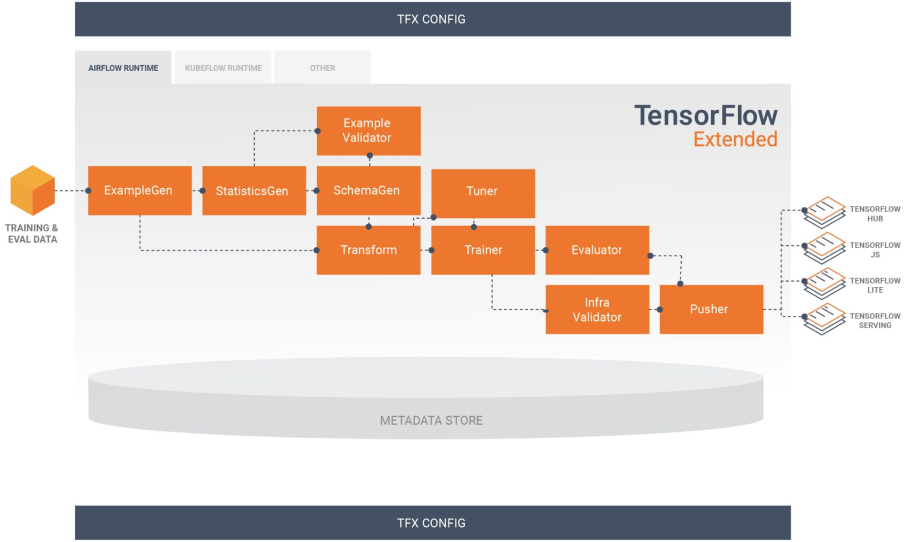
[Pedregosa JMLR '11]
Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830. [Paper] [Website] [Code]

Tensorflow Extended (TFX)

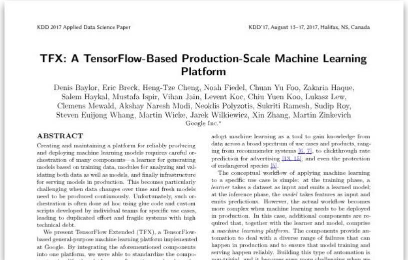
TensorFlow Extended

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***



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



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 Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, Chiu Yuen Koo, Lukasz Lew, Clemens Mewald, Akshay Naresh Modi, Neoklis Polyzotis, Sukriti Ramesh, Sudip Roy, Steven Euijong Whang, Martin Wicke, Jarek Wilkiewicz, Xin Zhang, Martin Zinkevich
Google Inc.*

ABSTRACT
Creating and maintaining a platform for reliably producing and deploying machine learning models requires careful orchestration of many components—a learner for generating models based on training data, modules for analyzing and validating both data as well as models, and finally infrastructure for serving models in production. This becomes particularly challenging when data changes over time and fresh models need to be produced continuously. Unfortunately, such orchestration is often done ad hoc using glue code and custom scripts developed by individual teams for specific use cases, leading to duplicated effort and fragile systems with high technical debt.

We present TensorFlow Extended (TFX), a TensorFlow-based general-purpose machine learning platform implemented at Google. By integrating the aforementioned components into one platform, we were able to standardize the compo-

adopt machine learning as a tool to gain knowledge from data across a broad spectrum of use cases and products, ranging from recommender systems [6, 7], to clickthrough rate prediction for advertising [13, 15], and even the protection of endangered species [5].

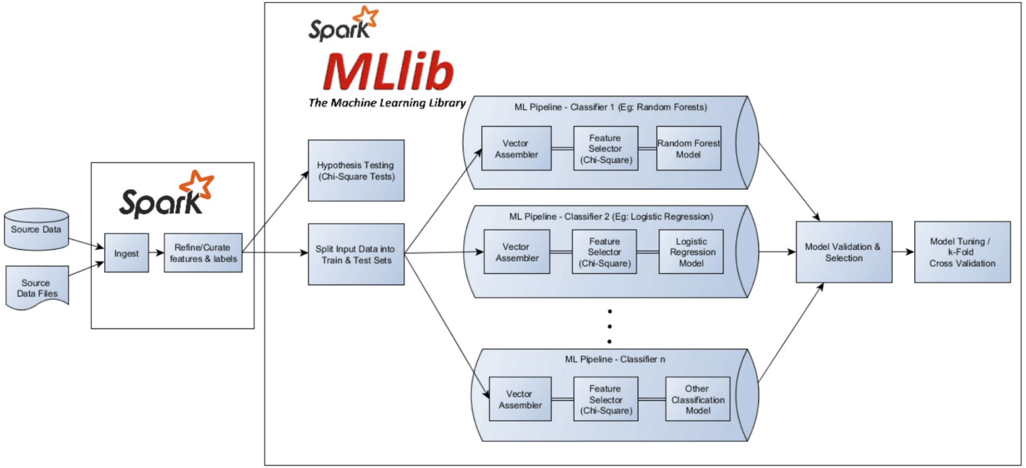
The conceptual workflow of applying machine learning to a specific use case is simple: at the training phase, a *learner* takes a dataset as input and emits a learned model; at the inference phase, the *model* takes features as input and emits predictions. However, the actual workflow becomes more complex when machine learning needs to be deployed in production. In this case, additional components are required that, together with the learner and model, comprise a *machine learning platform*. The components provide automation to deal with a diverse range of failures that can happen in production and to ensure that model training and serving happen reliably. Building this type of automation is

[Baylor SIGKDD '17]
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

Journal of Machine Learning Research 17 (2016) 1-7 Submitted 5/15; Published 4/16

MLlib: Machine Learning in Apache Spark

Xiangrui Meng[1] MENG@DATABRICKS.COM
Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105
Joseph Bradley JOSEPH@DATABRICKS.COM
Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105
Burak Yavuz BURAK@DATABRICKS.COM
Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105
Evan Sparks SPARKS@CS.BERKELEY.EDU
UC Berkeley, 465 Soda Hall, Berkeley, CA 94720
Shivaram Venkataraman SHIVARAM@EECS.BERKELEY.EDU
UC Berkeley, 465 Soda Hall, Berkeley, CA 94720
Davies Liu DAVIES@DATABRICKS.COM
Databricks, 160 Spear Street, 13th Floor, San Francisco, CA 94105
Jeremy Freeman FREEMANJ11@JANELIA.HHMI.ORG
HHMI Janelia Research Campus, 19805 Helix Dr, Ashburn, VA 20147
DB Tsai DBT@NETFLIX.COM
Netflix, 970 University Ave, Los Gatos, CA 95032
Manish Amde MANISH@ORIGAMILOGIC.COM
Origami Logic, 1134 Crane Street, Menlo Park, CA 94025

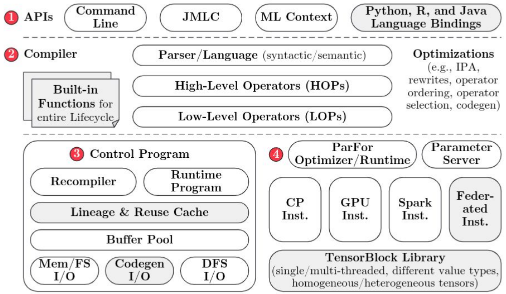
[Meng JMLR '16]
Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark." Journal of Machine Learning Research 17.34 (2016): 1-7. [Paper] [Website] [Code]

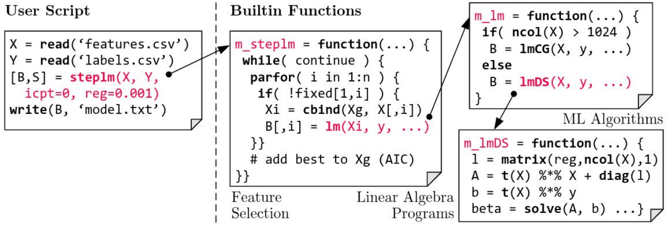
Apache SystemDS

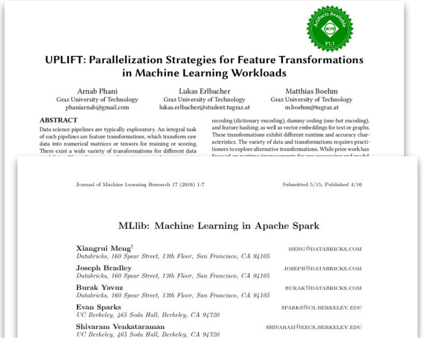
Apache SystemDS ™

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*







[Boehm CIDR '20]
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 VLDB '20]
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 Pipelines

[Website] [Documentation]

Highlights

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

Azure Machine Learning Pipelines

[Website] [Documentation]

Highlights

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

Vertex AI Pipelines

[Website] [Documentation]

Highlights

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

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***

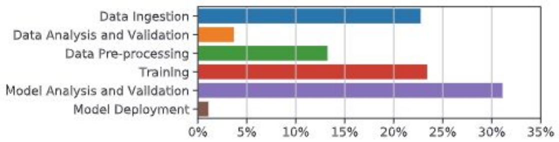
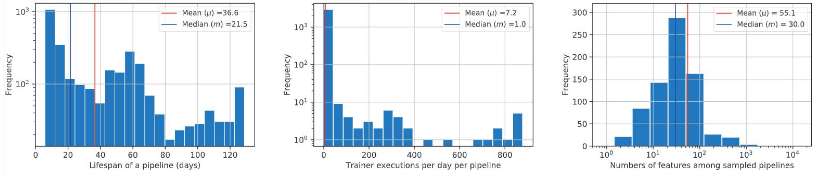


Figure 7: Compute cost of different operators.



(a) Distribution of pipeline span. **(b) Distribution of trained models per day.** **(c) Distribution of the number of features.**

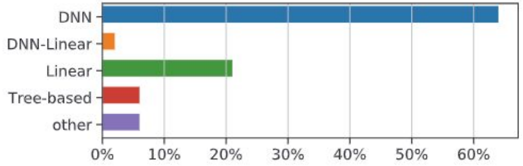
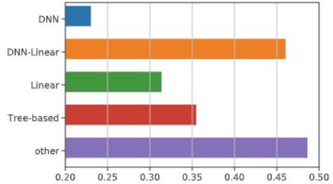
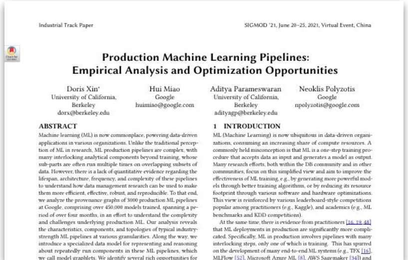


Figure 5: Percentage of Trainer runs with each model type



(f) Model type vs. likelihood of pushes.


Industrial Track Paper SIGMOD '21, June 20–25, 2021, Virtual Event, China

Production Machine Learning Pipelines:
Empirical Analysis and Optimization Opportunities

Doris Xin, University of California, Berkeley, dorx@berkeley.edu
Hui Miao, Google, huimiao@google.com
Aditya Parameswaran, University of California, Berkeley, adityagp@berkeley.edu
Neoklis Polyzotis, Google, npolyzotis@google.com

ABSTRACT

1 INTRODUCTION

[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)*

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

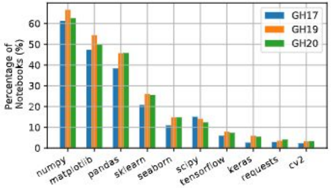


Figure 2: Top-10 used libraries.

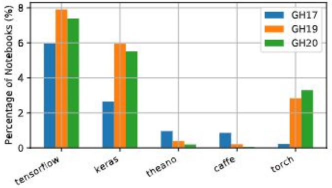
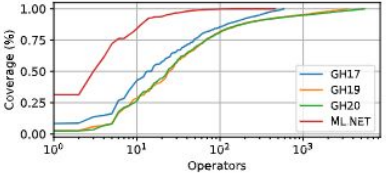
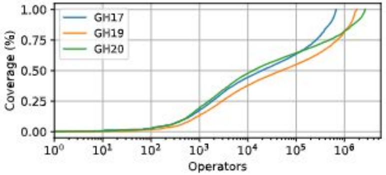
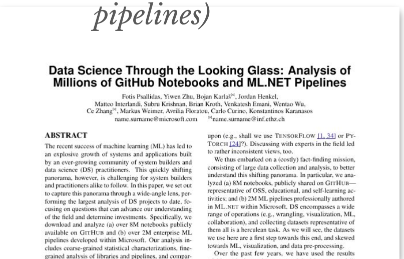


Figure 3: DL libraries usage percentages.

		GH17	GH19	GH20	ML.NET
#Pipelines	Implicit	164K	415K	1.4M	N/A
	Explicit	10K	129K	252K	29.7M
#Distinct Ops	Implicit	668K	1.8M	2.6M	N/A
	Explicit	584	3.4K	5.5K	23.5K







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

Fotis Psallidas, Yiwen Zhu, Bojan Karlaš, Jordan Henkel, Matteo Interlandi, Subru Krishnan, Brian Kroth, Venkatesh Emani, Wentao Wu, Ce Zhang, Markus Weimer, Avrilia Floratou, Carlo Curino, Konstantinos Karanasos
name.surname@microsoft.com name.surname@inf.ethz.ch

ABSTRACT

The recent success of machine learning (ML) has led to an explosive growth of systems and applications built by an ever-growing community of system builders and data science (DS) practitioners. This quickly shifting panorama, however, is challenging for system builders and practitioners alike to follow. In this paper, we set out to capture this panorama through a wide-angle lens, performing the largest analysis of DS projects to date, focusing on questions that can advance our understanding of the field and determine investments. Specifically, we download and analyze (a) over 8M notebooks publicly available on GitHub and (b) over 2M enterprise ML pipelines developed within Microsoft. Our analysis includes coarse-grained statistical characterizations, fine-grained analysis of libraries and pipelines, and compar-

upon (e.g., shall we use TensorFlow [1, 34] or PyTorch [24]?). Discussing with experts in the field led to rather inconsistent views, too.

We thus embarked on a (costly) fact-finding mission, consisting of large data collection and analysis, to better understand this shifting panorama. In particular, we analyzed (a) 8M notebooks, publicly shared on GitHub—representative of OSS, educational, and self-learning activities; and (b) 2M ML pipelines professionally authored in ML.NET within Microsoft. DS encompasses a wide range of operations (e.g., wrangling, visualization, ML, collaboration), and collecting datasets representative of them all is a herculean task. As we will see, the datasets we use here are a first step towards this end, and skewed towards ML, visualization, and data pre-processing.

Over the past few years, we have used the results

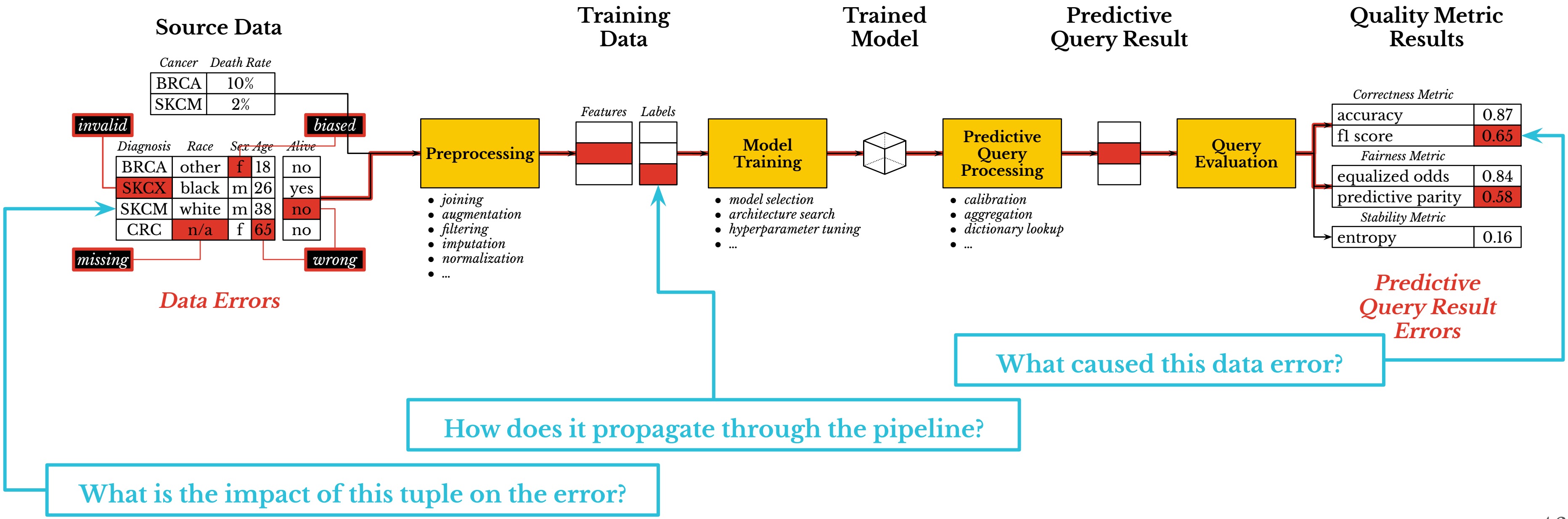
[Psallidas SIGMOD Record '22]
Psallidas, Fotis, et al. "Data science through the looking glass: Analysis of millions of github notebooks and ml. net pipelines." ACM SIGMOD Record 51.2 (2022): 30-37. [Paper]

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**

How should we reason about pipelines?



Modeling ML Pipelines with "Logical Query Plans"

Challenge

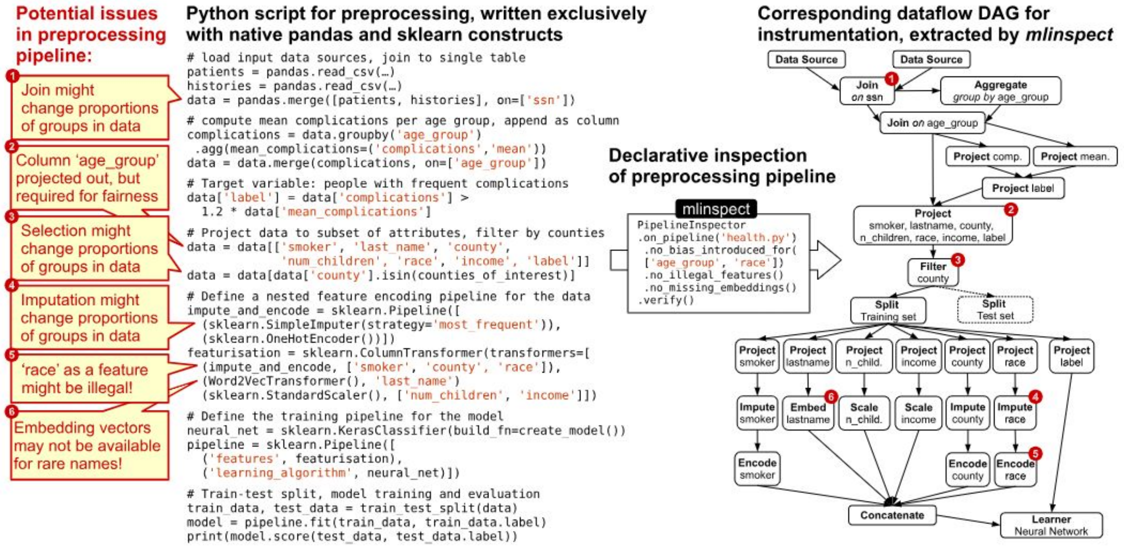
Understanding of the ***semantics of operations*** *and the* ***flow of data required*** *to reason about ML pipelines*

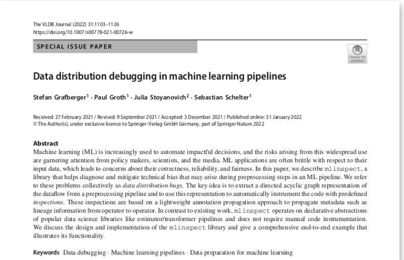
Insight

Many common pipeline abstractions offer ***declarative operations****, enables the extraction and definition of "logical query plans" modeling their operations*

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.*




The VLDB Journal (2022) 31:1103–1126
https://doi.org/10.1007/s00778-021-00726-w

SPECIAL ISSUE PAPER

Data distribution debugging in machine learning pipelines

Stefan Grafberger[1] · Paul Groth[1] · Julia Stoyanovich[2] · Sebastian Schelter[1]

Received: 27 February 2021 / Revised: 9 September 2021 / Accepted: 3 December 2021 / Published online: 31 January 2022

Abstract
Machine learning (ML) is increasingly used to automate impactful decisions, and the risks arising from this widespread use are garnering attention from policy makers, scientists, and the media. ML applications are often brittle with respect to their input data, which leads to concerns about their correctness, reliability, and fairness. In this paper, we describe `mlinspect`, a library that helps diagnose and mitigate technical bias that may arise during preprocessing steps in an ML pipeline. We refer to these problems collectively as *data distribution bugs*. The key idea is to extract a directed acyclic graph representation of the dataflow from a preprocessing pipeline and to use this representation to automatically instrument the code with predefined *inspections*. These inspections are based on a lightweight annotation propagation approach to propagate metadata such as lineage information from operator to operator. In contrast to existing work, `mlinspect` operates on declarative abstractions of popular data science libraries like estimator/transformer pipelines and does not require manual code instrumentation. We discuss the design and implementation of the `mlinspect` library and give a comprehensive end-to-end example that illustrates its functionality.

Keywords Data debugging · Machine learning pipelines · Data preparation for machine learning

[Grafberger VLDBJ '22]
Grafberger, Stefan, et al. "Data distribution debugging in machine learning pipelines." The VLDB Journal 31.5 (2022): 1103-1126. [Paper] [Code]

Leveraging the Provenance Semiring Framework

Highlights

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

Application to an Example Pipeline

Provenance	*Cancer*	*Death Rate*
$x_{1,1}$	BRCA	10%
$x_{1,2}$	SKCX	2%

Provenance	*Diagnosis*	*Race*	*Sex*	*Age*	*Alive*
$x_{2,1}$	BRCA	other	f	18	no
$x_{2,2}$	SKCM	black	m	26	yes
$x_{2,3}$	SKCM	white	m	38	no
$x_{2,4}$	CRC	n/a	f	65	no

Join

Provenance	*Death Rate*	*Diagnosis*	*Race*	*Sex*	*Age*	*Alive*
$x_{1,1} \cdot x_{2,1}$	10%	BRCA	other	f	18	no
$x_{1,2} \cdot x_{2,2}$	2%	SKCM	black	m	26	yes
$x_{1,2} \cdot x_{2,3}$	2%	SKCM	white	m	38	no
$x_{1,1} \cdot x_{2,4}$	n/a	CRC	n/a	f	65	no

NA Filter

Provenance	*Death Rate*	*Diagnosis*	*Race*	*Sex*	*Age*	*Alive*
$x_{1,1} \cdot x_{2,1}$	10%	BRCA	other	f	18	no
$x_{1,2} \cdot x_{2,2}$	2%	SKCM	black	m	26	yes
$x_{1,2} \cdot x_{2,3}$	2%	SKCM	white	m	38	no

Select Distinct Diagnosis

Provenance	*Diagnosis*
$x_{1,1} \cdot x_{2,1}$	BRCA
$x_{1,2} \cdot (x_{2,2} + x_{2,3})$	SKCM

Provenance Semirings

Todd J. Green tjgreen@cis.upenn.edu — Grigoris Karvounarakis gkarvoun@cis.upenn.edu — Val Tannen val@cis.upenn.edu

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA

ABSTRACT

We show that relational algebra calculations for incomplete databases, probabilistic databases, bag semantics and why-provenance are particular cases of the same general algorithms involving semirings. This further suggests a comprehensive provenance representation that uses semirings of polynomials. We extend these considerations to datalog and semirings of formal power series. We give algorithms for datalog provenance calculation as well as datalog evaluation for incomplete and probabilistic databases. Finally, we show that for some semirings containment of conjunctive queries is the same as for standard set semantics.

Categories and Subject Descriptors

H.2.1 [**Database Management**]: Data Models

General Terms

We observe that in all four cases, the calculations with annotations are strikingly similar. This suggests looking for an algebraic structure on annotations that captures the above as particular cases. We propose using commutative semirings for this purpose. In fact, we can show that the laws of commutative semirings are *forced* by certain expected identities in $\mathcal{RA}$. Having identified commutative semirings as the right algebraic structure, we argue that a symbolic representation of semiring calculations is just what is needed to record, document, and track $\mathcal{RA}$ querying from input to output for applications which require rich *provenance* information. It is a standard philosophy in algebra that such symbolic representations form *the most general* such structure. In the case of commutative semirings, just as for rings, the symbolic representation is that of polynomials. We therefore propose to use polynomials to capture provenance. Next we look to extend our approach to recursive datalog queries. To achieve this we combine semirings with fixed point theory.

[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

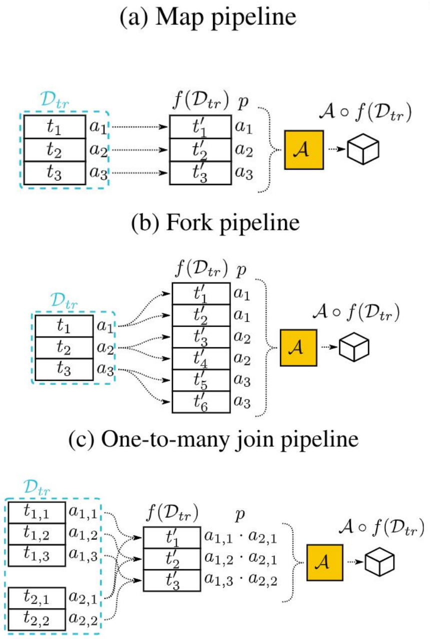
KNN Proxy methods not directly applicable to arbitrary pipelines. Presence of a single source data point does not map directly to a single data point fed to the model.

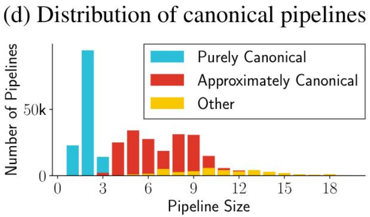
Insight

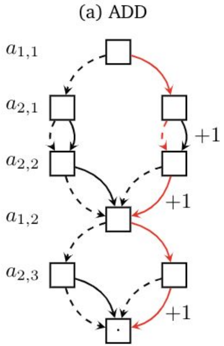
Observe ***three canonical types of pipelines*** *based on shape of produced provenance polynomials. Possible to develop efficient PTIME algorithms for computing the Shapley value for them.*

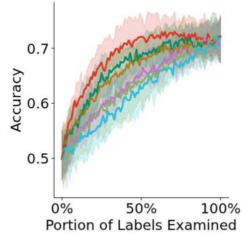
Approach

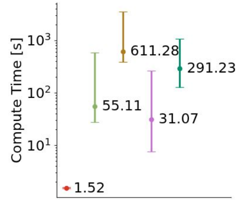
Compile provenance polynomials to Additive Decision Diagrams and use them to compute Shapley values in PTIME.

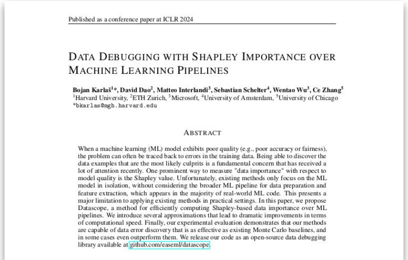











Published as a conference paper at ICLR 2024

DATA DEBUGGING WITH SHAPLEY IMPORTANCE OVER MACHINE LEARNING PIPELINES

Bojan Karlaš[1]*, David Dao[2], Matteo Interlandi[3], Sebastian Schelter[4], Wentao Wu[3], Ce Zhang[5]
[1]Harvard University, [2]ETH Zurich, [3]Microsoft, [4]University of Amsterdam, [5]University of Chicago
*bkarlas@mgh.harvard.edu

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 examples that are the most likely culprits is a fundamental concern that has received a lot of attention recently. One prominent way to measure "data importance" with respect to model quality is the Shapley value. Unfortunately, existing methods only focus on the ML model in isolation, without considering the broader ML pipeline for data preparation and feature extraction, which appears in the majority of real-world ML code. This presents a major limitation to applying existing methods in practical settings. In this paper, we propose Datascope, a method for efficiently computing Shapley-based data importance over ML pipelines. We introduce several approximations that lead to dramatic improvements in terms of computational speed. Finally, our experimental evaluation demonstrates that our methods are capable of data error discovery that is as effective as existing Monte Carlo baselines, and in some cases even outperform them. We release our code as an open-source data debugging library available at github.com/easeml/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

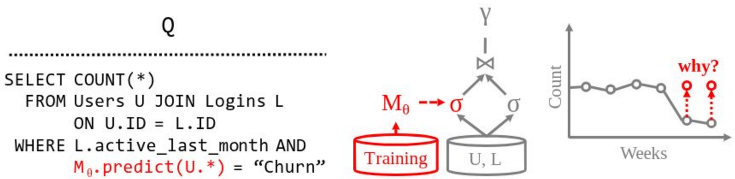
Model inference often part of a larger predictive query. Influence-based attribution methods must account for structure of query.

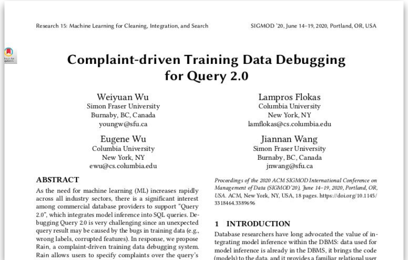
Insight

*Provenance polynomials for tracking **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) used to identify errors. Make the **entire query differentiable** using provenance polynomials and run the influence framework to identify errors in the training dataset.*




Research 15: Machine Learning for Cleaning, Integration, and Search SIGMOD '20, June 14–19, 2020, Portland, OR, USA

Complaint-driven Training Data Debugging for Query 2.0

Weiyuan Wu, Simon Fraser University, Burnaby, BC, Canada, youngw@sfu.ca
Lampros Flokas, Columbia University, New York, NY, lamflokas@cs.columbia.edu
Eugene Wu, Columbia University, New York, NY, ewu@cs.columbia.edu
Jiannan Wang, Simon Fraser University, Burnaby, BC, Canada, jnwang@sfu.ca

ABSTRACT

As the need for machine learning (ML) increases rapidly across all industry sectors, there is a significant interest among commercial database providers to support "Query 2.0", which integrates model inference into SQL queries. Debugging Query 2.0 is very challenging since an unexpected query result may be caused by the bugs in training data (e.g., wrong labels, corrupted features). In response, we propose Rain, a complaint-driven training data debugging system. Rain allows users to specify complaints over the query's

1 INTRODUCTION

Database researchers have long advocated the value of integrating model inference within the DBMS: data used for model inference is already in the DBMS, it brings the code

[Wu SIGMOD '20]
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

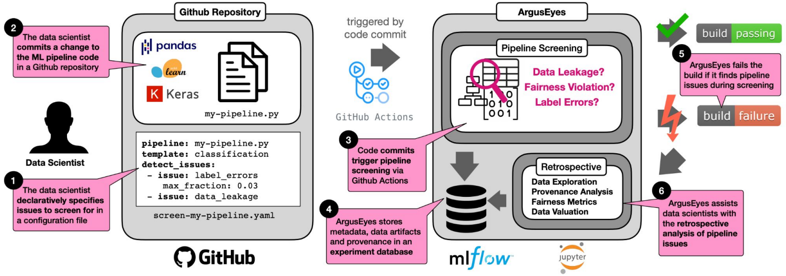
ML systems lack sophisticated testing infrastructure *developed for classical software engineering. Many data-related problems only become apparent in production.*

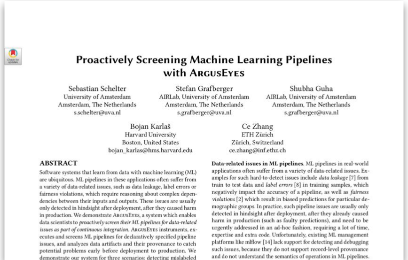
Insight

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, University of Amsterdam, Amsterdam, The Netherlands, s.schelter@uva.nl
Stefan Grafberger, AIRLab, University of Amsterdam, Amsterdam, The Netherlands, s.grafberger@uva.nl
Shubha Guha, AIRLab, University of Amsterdam, Amsterdam, The Netherlands, s.grafberger@uva.nl
Bojan Karlaš, Harvard University, Boston, United States, bojan_karlas@hms.harvard.edu
Ce Zhang, ETH Zürich, Zürich, Switzerland, ce.zhang@inf.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, label errors or fairness violations, which require reasoning about complex dependencies between their inputs and outputs. These issues are usually only detected in hindsight after deployment, after they caused harm in production. We demonstrate ArgusEyes, a system which enables data scientists to *proactively screen their ML pipelines for data-related issues as part of continuous integration*. ArgusEyes instruments, executes and screens ML pipelines for declaratively specified pipeline issues, and analyzes data artifacts and their provenance to catch potential problems early before deployment to production. We demonstrate our system for three scenarios: detecting mislabeled

Data-related issues in ML pipelines. ML pipelines in real-world applications often suffer from a variety of data-related issues. Examples for such hard-to-detect issues include *data leakage* [7] from train to test data and *label errors* [8] in training samples, which negatively impact the accuracy of a pipeline, as well as *fairness violations* [2] which result in biased predictions for particular demographic groups. In practice, such pipeline issues are usually only detected in hindsight after deployment, after they already caused harm in production (such as faulty predictions), and need to be urgently addressed in an ad-hoc fashion, requiring a lot of time, expertise and extra code. Unfortunately, existing ML management platforms like mlflow [14] lack support for detecting and debugging such issues, because they do not support record-level provenance and do not understand the semantics of operations in ML pipelines.

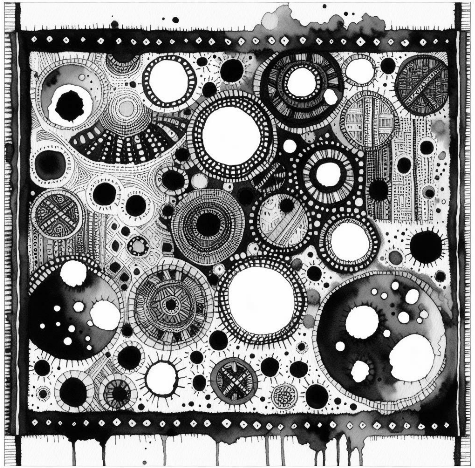
[Schelter SIGMOD Demo '23]
Schelter, et al.: "Proactively Screening Machine Learning Pipelines with ArgusEyes." Proceedings of the 2023 ACM SIGMOD International Conference on Management of Data (demo). 2023. [Paper]

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

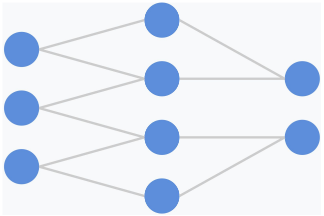
Babak Salimi



The Standard ML Pipeline

Input Data ➡ Data Cleaning ➡ Model ➡ Inference

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



Loan Denied: **High** Risk

Alice

Loan Approved: **Low** Risk

Bob

⚠️ **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]

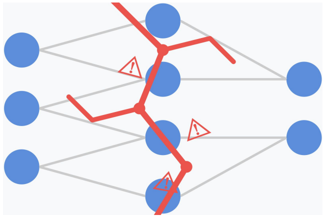
Schema / integration mismatch (⚠️ inconsistent units, 🚫 fuzzy entity resolution)
Ambiguous merges: no ground-truth correspondences.
Pre-processing bias: heuristics distort original distributions; matching is probabilistic.

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

Challenges with Traditional Data Pipelines

Input Data → **Data Cleaning** → **Model** → **Inference**

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



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.

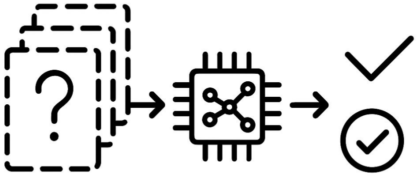
Learning from Incomplete Databases

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

Hidden uncertainty ⇒ 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 → Certain answer
  Range ≤ τ → Robust interval (e.g., [5 k – 6 k])
- Range > τ → Uncertain → 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
…	…	…	…	…

→ $Q(D_3) = 5k$

Range consistent answers:
[0.5 - 0.3]

Min/Max query result across all possible database repairs.

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

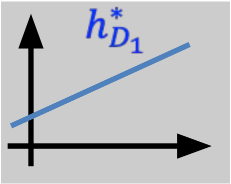
- All models ($h^*_{D_i}$)concur → ***Certain*** prediction (e.g., payout = 3 K)
- disagree → ***Range*** prediction (e.g., payout ∈ [2 K , 4 K])

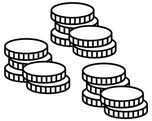
Dataset with Quality Issues

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

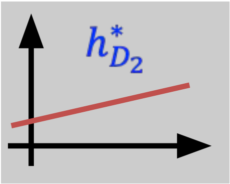
Machine-learning analogue of **Consistent Query Answering**: swap the SQL query **Q** for a training routine **T**—e.g., gradient descent, decision-tree induction, SVM fitting.

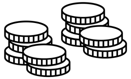
ID	Age	Income	…	Loan
1	25	50K	…	5K
2	30	60K	…	8K
3	35	55K	…	7K
…	…	…	…	…

→ 

→  3K

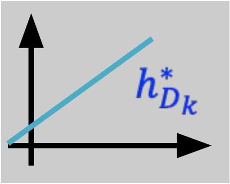
ID	Age	Income	…	Loan
1	25	50K	…	5K
2	35	60K	…	8K
3	35	60K	…	8K
…	…	…	…	…

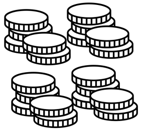
→ 

→  2K

…

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

→ 

→  4K

KNN Classifiers over Incomplete Information

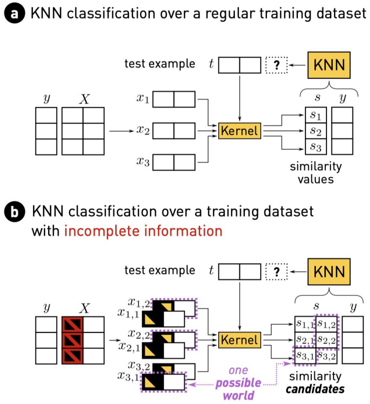
[Approach: “Certain-kNN” → returns a label only when it is guaranteed across all completions of the missing values]

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.

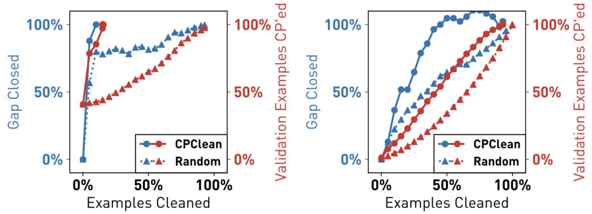


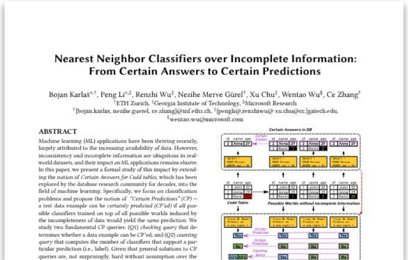
Benefits:

- **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*,†, Peng Li*,‡, Renzhi Wu‡, Nezihe Merve Gürel†, Xu Chu‡, Wentao Wu§, Ce Zhang†
†ETH Zurich, ‡Georgia Institute of Technology, §Microsoft Research
†{bojan.karlas, nezihe.guerel, ce.zhang}@inf.ethz.ch, ‡{pengli@, renzhiwu@, xu.chu@cc.}gatech.edu,
§wentao.wu@microsoft.com

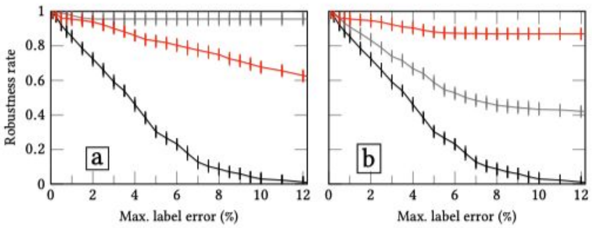
ABSTRACT

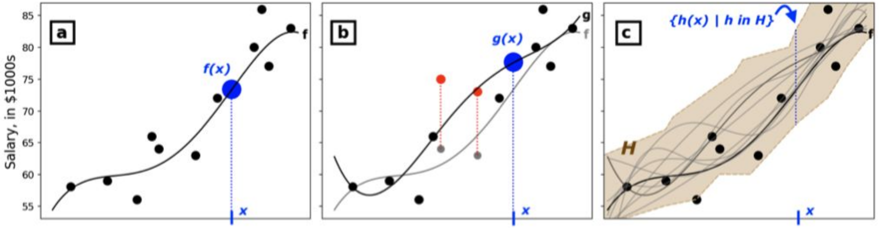
Machine learning (ML) applications have been thriving recently, largely attributed to the increasing availability of data. However, inconsistency and incomplete information are ubiquitous in real-world datasets, and their impact on ML applications remains elusive. In this paper, we present a formal study of this impact by extending the notion of *Certain Answers for Codd tables*, which has been explored by the database research community for decades, into the field of machine learning. Specifically, we focus on classification problems and propose the notion of *"Certain Predictions" (CP)* – a test data example can be *certainly predicted (CP'ed)* if *all* possible classifiers trained on top of all possible worlds induced by the incompleteness of data would yield the *same* prediction. We study two fundamental CP queries: (Q1) *checking query* that determines whether a data example can be CP'ed; and (Q2) *counting query* that computes the number of classifiers that support a particular prediction (i.e., label). Given that general solutions to CP queries are, not surprisingly, hard without assumption over the

[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]

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:

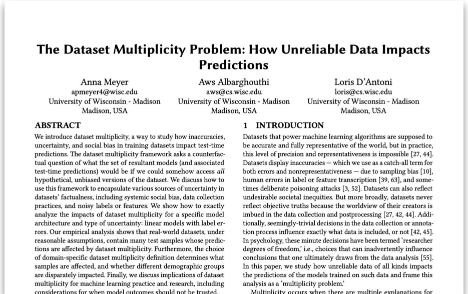
- Derive closed-form formulas for the worst- and best-case hinge / logistic loss of any linear classifier under those rules, avoiding enumeration.

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
apmeyer4@wisc.edu
University of Wisconsin - Madison
Madison, USA

Aws Albarghouthi
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 inaccuracies, uncertainty, and social bias in training datasets impact test-time predictions. The dataset multiplicity framework asks a counterfactual question of what the set of resultant models (and associated test-time predictions) would be if we could somehow access *all* hypothetical, unbiased versions of the dataset. We discuss how to use this framework to encapsulate various sources of uncertainty in datasets' factualness, including systemic social bias, data collection practices, and noisy labels or features. We show how to exactly analyze the impacts of dataset multiplicity for a specific model architecture and type of uncertainty: linear models with label errors. Our empirical analysis shows that real-world datasets, under reasonable assumptions, contain many test samples whose predictions are affected by dataset multiplicity. Furthermore, the choice of domain-specific dataset multiplicity definition determines what samples are affected, and whether different demographic groups are disparately impacted. Finally, we discuss implications of dataset multiplicity for machine learning practice and research, including considerations for when model outcomes should not be trusted.

1 INTRODUCTION

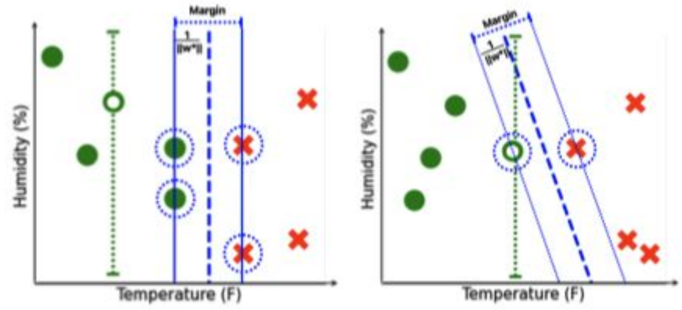
Datasets that power machine learning algorithms are supposed to be accurate and fully representative of the world, but in practice, this level of precision and representativeness is impossible [27, 44]. Datasets display inaccuracies — which we use as a catch-all term for both errors and nonrepresentativeness — due to sampling bias [10], human errors in label or feature transcription [39, 63], and sometimes deliberate poisoning attacks [3, 52]. Datasets can also reflect undesirable societal inequities. But more broadly, datasets never reflect objective truths because the worldview of their creators is imbued in the data collection and postprocessing [27, 42, 44]. Additionally, seemingly-trivial decisions in the data collection or annotation process influence exactly what data is included, or not [42, 45]. In psychology, these minute decisions have been termed 'researcher degrees of freedom,' i.e., choices that can inadvertently influence conclusions that one ultimately draws from the data analysis [55]. In this paper, we study how unreliable data of all kinds impacts the predictions of the models trained on such data and frame this analysis as a 'multiplicity problem.'

Multiplicity occurs when there are multiple explanations for

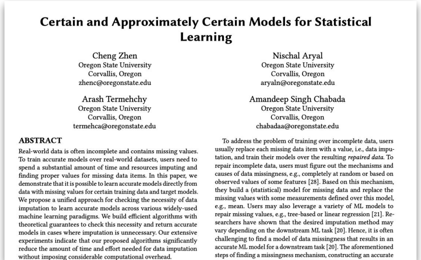
[Meyer FAccT'23]
Meyer, A. P.; Albarghouthi, A.; D'Antoni, L. "The Dataset Multiplicity Problem: How Unreliable Data Impacts Predictions. [Paper]

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]



(a) Data cleaning is not needed **(b) Data cleaning is needed**



Certain and Approximately Certain Models for Statistical Learning

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

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

Arash Termehchy
Oregon State University
Corvallis, Oregon
termehca@oregonstate.edu

Amandeep Singh Chabada
Oregon State University
Corvallis, Oregon
chabadaa@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 and resources imputing and finding proper values for missing data items. In this paper, we demonstrate that it is possible to learn accurate models directly from data with missing values for certain training data and target models. We propose a unified approach for checking the necessity of data imputation to learn accurate models across various widely-used machine learning paradigms. We build efficient algorithms with theoretical guarantees to check this necessity and return accurate models in cases where imputation is unnecessary. Our extensive experiments indicate that our proposed algorithms significantly reduce the amount of time and effort needed for data imputation without imposing considerable computational overhead.

To address the problem of training over incomplete data, users usually replace each missing data item with a value, i.e., data imputation, and train their models over the resulting *repaired data*. To repair incomplete data, users must figure out the mechanisms and causes of data missingness, e.g., completely at random or based on observed values of some features [28]. Based on this mechanism, they build a (statistical) model for missing data and replace the missing values with some measurements defined over this model, e.g., mean. Users may also leverage a variety of ML models to repair missing values, e.g., tree-based or linear regression [21]. Researchers have shown that the desired imputation method may vary depending on the downstream ML task [20]. Hence, it is often challenging to find a model of data missingness that results in an accurate ML model for a downstream task [20]. The aforementioned steps of finding a missingness mechanism, constructing an accurate

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 **ε-certain model** whose loss is within ε 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.

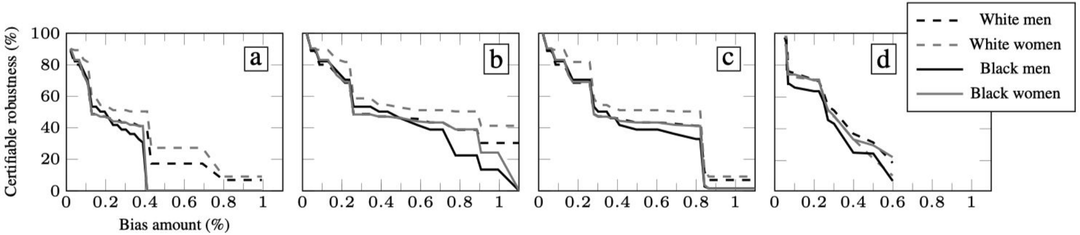
Shortcomings:

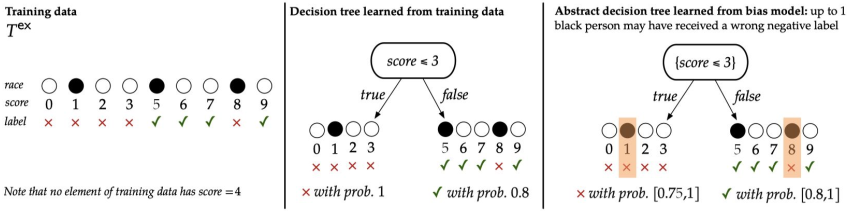
- Certainty rarely holds under heavy missingness.
  Guarantees limited to the studied linear & kernel models; deep nets need other methods.

[Zhen SIGMOD’24]
Zhen, C. et al. “Certain and Approximately Certain Models for Statistical Learning. [Paper]

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:

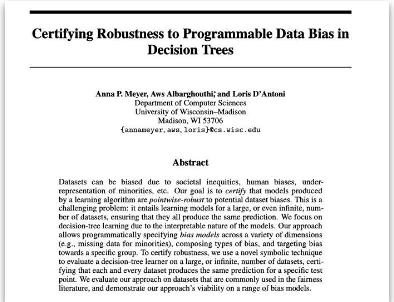
- Translate each path of the decision tree **and** the bias constraints into a single SMT formula.

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 Loris D'Antoni
Department of Computer Sciences
University of Wisconsin–Madison
Madison, WI 53706
{annameyer, aws, loris}@cs.wisc.edu

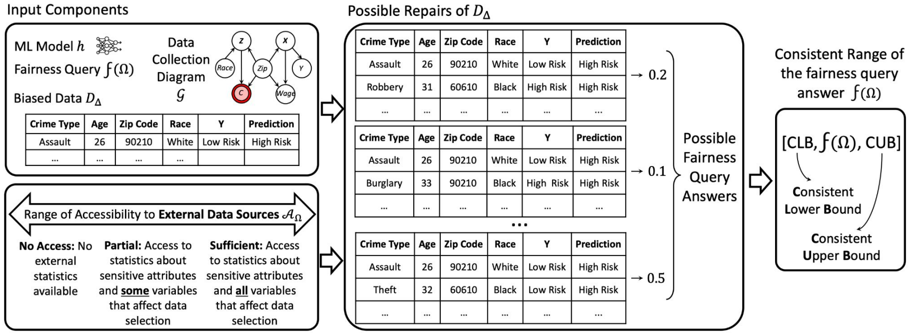
Abstract

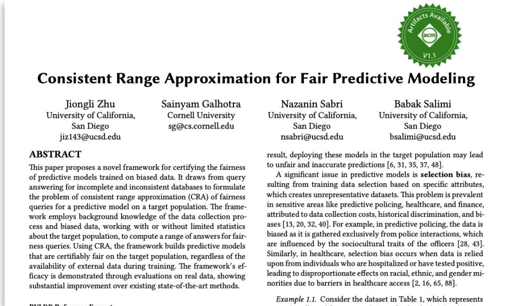
Datasets can be biased due to societal inequities, human biases, under-representation of minorities, etc. Our goal is to *certify* that models produced by a learning algorithm are *pointwise-robust* to potential dataset biases. This is a challenging problem: it entails learning models for a large, or even infinite, number of datasets, ensuring that they all produce the same prediction. We focus on decision-tree learning due to the interpretable nature of the models. Our approach allows programmatically specifying *bias models* across a variety of dimensions (e.g., missing data for minorities), composing types of bias, and targeting bias towards a specific group. To certify robustness, we use a novel symbolic technique to evaluate a decision-tree learner on a large, or infinite, number of datasets, certifying 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





Consistent Range Approximation for Fair Predictive Modeling

Jiongli Zhu, University of California, San Diego, jiz143@ucsd.edu
Sainyam Galhotra, Cornell University, sg@cs.cornell.edu
Nazanin Sabri, University of California, San Diego, nsabri@ucsd.edu
Babak Salimi, University of California, San Diego, bsalimi@ucsd.edu

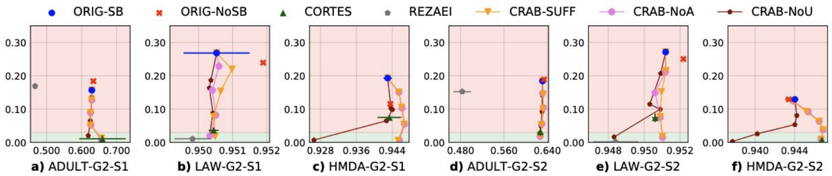
ABSTRACT

This paper proposes a novel framework for certifying the fairness of predictive models trained on biased data. It draws from query answering for incomplete and inconsistent databases to formulate the problem of consistent range approximation (CRA) of 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 compute a range of answers for fairness queries. Using CRA, the framework builds predictive models that are certifiably 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 selection based on specific attributes, which creates unrepresentative datasets. This problem is prevalent in sensitive areas like predictive policing, healthcare, and finance, attributed to data collection costs, historical discrimination, and biases [13, 20, 32, 40]. For example, in predictive policing, 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 bias occurs when data is relied upon from individuals who are hospitalized or have tested positive, leading to disproportionate effects on racial, ethnic, and gender minorities due to barriers in 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.

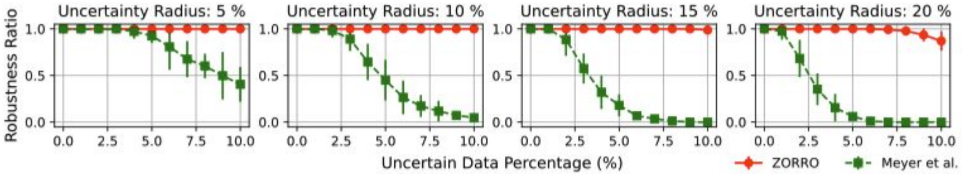
Shortcomings:

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

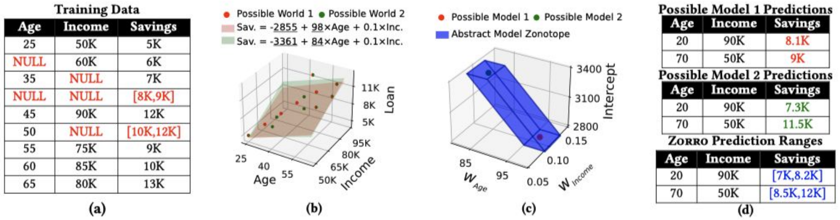
[Zhu VLDB '23]
Consistent Range Approximation for Fair Predictive Modeling. [Paper]

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



(a) Robustness verification with uncertain labels (MPG data).



Training Data

Age	Income	Savings
25	50K	5K
NULL	60K	6K
35	NULL	7K
NULL	NULL	[8K,9K]
45	90K	12K
50	NULL	[10K,12K]
55	75K	9K
60	85K	10K
65	80K	13K

Possible Model 1 Predictions

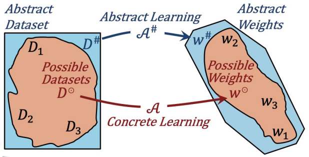
Age	Income	Savings
20	90K	8.1K
70	50K	9K

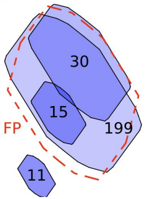
Possible Model 2 Predictions

Age	Income	Savings
20	90K	7.3K
70	50K	11.5K

ZORRO Prediction Ranges

Age	Income	Savings
20	90K	[7K,8.2K]
70	50K	[8.5K,12K]





Insights:

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

Approach:

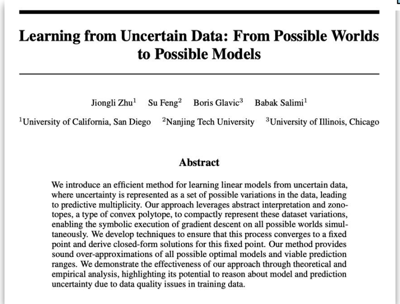
- Map each uncertain record to an affine form; the full dataset becomes **one zonotope**. Run gradient descent **symbolically**. Output is a convex box of model weights; any concrete repair yields weights inside this box.

Benefits:

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

Shortcomings:

- Supports **linear** models only.



Learning from Uncertain Data: From Possible Worlds to Possible Models

Jiongli Zhu[1] Su Feng[2] Boris Glavic[3] Babak Salimi[1]

[1]University of California, San Diego [2]Nanjing Tech University [3]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, enabling the symbolic execution of gradient descent on all possible worlds simultaneously. We develop techniques to 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 demonstrate the effectiveness of our approach through theoretical and empirical analysis, highlighting its potential to reason about model and prediction uncertainty due to data quality issues in training data.

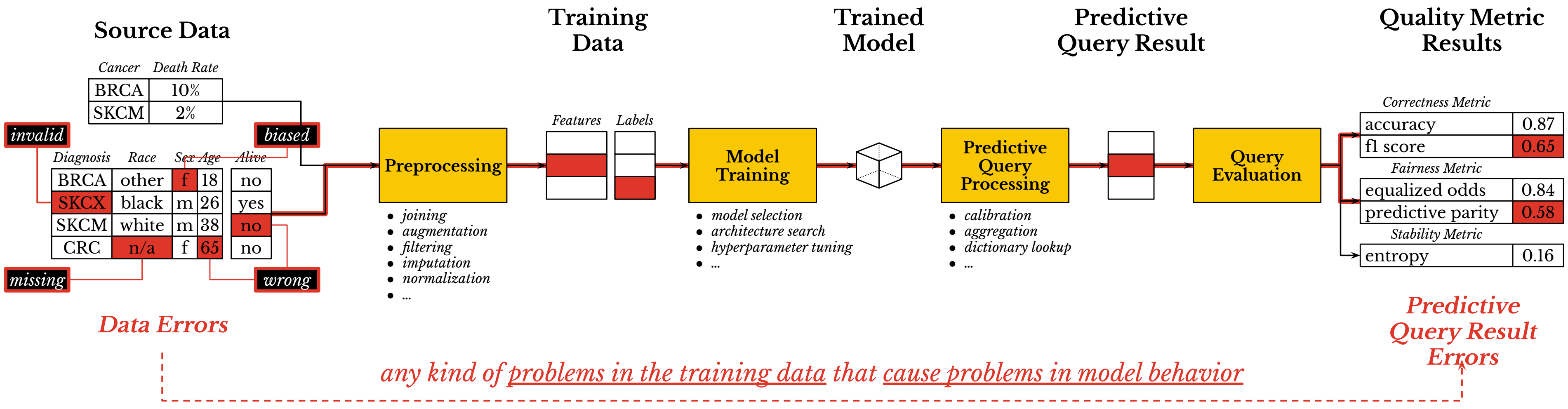
[Zhu NeurIPS'24]

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

Key Takeaways of Part III

- Data cleaning can not only be costly, but it can also introduce the risk of reducing data quality.
- Possible worlds semantics is a fundamental concept for analyzing uncertainty over possible data repairs.
- Since the number of possible worlds is exponential, approximations and algorithmic tricks are needed to make reasoning about uncertainty tractable.

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

https://navigating-data-errors.github.io

Thank you!