Architecture of ML Systems (AMLS)
11 Model Debugging, Fairness, and Explainability

Prof. Dr. Matthias Boehm
Technische Universität Berlin
Berlin Institute for the Foundations of Learning and Data
Big Data Engineering (DAMS Lab)

Last update: Jul 06, 2023

Announcements / Org

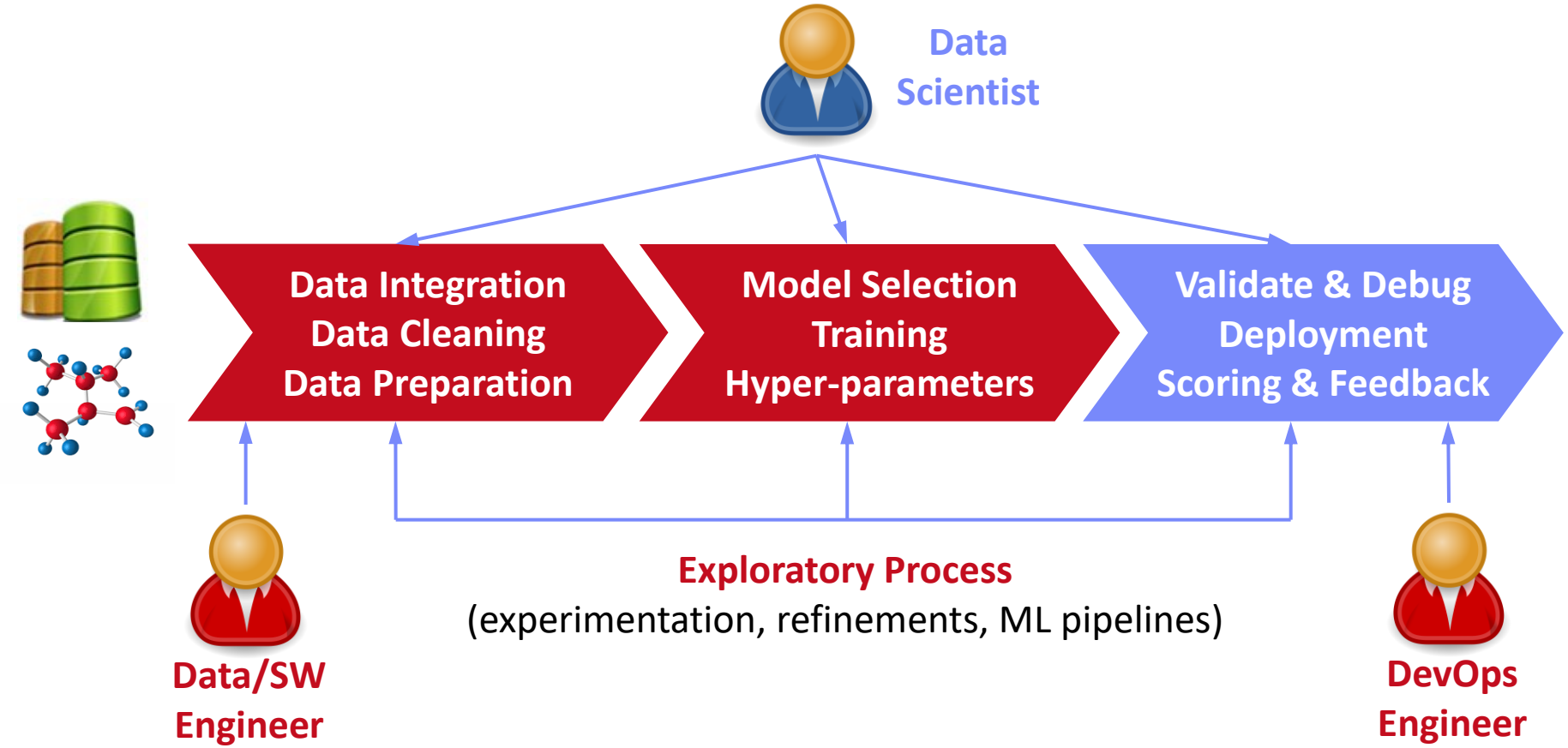
- **#1 Hybrid & Video Recording**
  - Hybrid lectures (in-person, zoom) with optional attendance
    https://tu-berlin.zoom.us/j/9529634787?pwd=R1ZsN1M3SC9BOU1OcFdmem9zT202UT09
  - Zoom **video recordings**, links from website
    https://mboehm7.github.io/**teaching**/ss23_amls/index.htm

- **#2 Project/Exercise Submission**
  - Original Deadline: **July 4** → **24h before individual exam slot**
  - Pull requests (SystemDS/DAPHNE), **note if done**; ISIS submission or email (for TU Graz students)

- **#3 Course Feedback / Evaluation**
  - ISIS Course feedback, active **July 10 – July 23, 2023**

Recap: The Data Science Lifecycle (aka KDD Process, aka CRISP-DM)

Data-centric View:
Application perspective
Workload perspective
System perspective



Agenda

- **Model Debugging and Explainability**
- **Model Bias & Fairness Constraints**

Model Debugging and Explainability

Similar to Software Testing
Focus on Benchmarks, Assessment, Monitoring,
Model Improvements, Model Understanding

Recap: Data Validation

Sanity checks on expected shape before training first model

- **Check a feature's min, max, and most common value**
  - Ex: Latitude values must be within the range [-90, 90] or [-π/2, π/2]
- **The histograms of continuous or categorical values are as expected**
  - Ex: There are similar numbers of positive and negative labels
- **Whether a feature is present in enough examples**
  - Ex: Country code must be in at least 70% of the examples
- **Whether a feature has the right number of values (i.e., cardinality)**
  - Ex: There cannot be more than one age of a person
- **Other**

[Neoklis Polyzotis, et al: Data Management Challenges in Production Machine Learning. Tutorial, **SIGMOD 2017**]

(**Google Research**)

[Sebastian Schelter et al: Automating Large-Scale Data Quality Verification. **PVLDB 2018**]

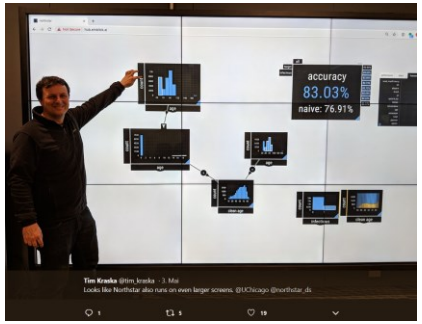
(**Amazon Research**)

[Mike Dreves et al: From Data to Models and Back **DEEM@SIGMOD 2020,** http://deem-workshop.org/videos/2020/8_dreves.mp4]

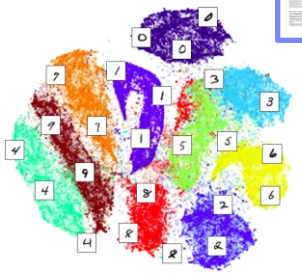
(**Google**)

Overview Model Debugging

[**Credit:** twitter.com/tim_kraska]



[Andrew Crotty et al: Vizdom: Interactive Analytics through Pen and Touch. **PVLDB 2015**]



[**Credit:** nlml.github.io/in-raw-numpy/in-raw-numpy-t-sne/]

- **#1 Understanding via Visualization**
  - Plotting of predictions / interactions
  - Combination with dimensionality reduction into 2D:
    - Autoencoder
    - PCA (principal component analysis)
    - t-SNE (T-distributed Stochastic Neighbor Embedding)
  - Input, intermediate, and output layers of DNNs

- **#2 Validation, Explainability, Fairness via Constraints**
  - Establish assertions and thresholds for automatic validation and alerts w.r.t. accuracy, bias, and other metrics
  - Generate succinct representations (e.g., rules) as explanation
  - Impose constraints like monotonicity for ensuring fairness

Basic Model-Specific Statistics

- **Regression Statistics**
  - Average response and stddev, average residuals stddev residuals
  - R2 (coeff of determination) with and without bias, etc

- **Classification Statistics**
  - Classical: recall, precision, F1-score, Area under the ROC Curve (AUC)
  - Visual: **confusion matrix** (correct vs predicated classes)
    ➔ understand performance wrt individual classes
  - Example Mnist
  - Mispredictions might also be visualized via dimensionality reduction

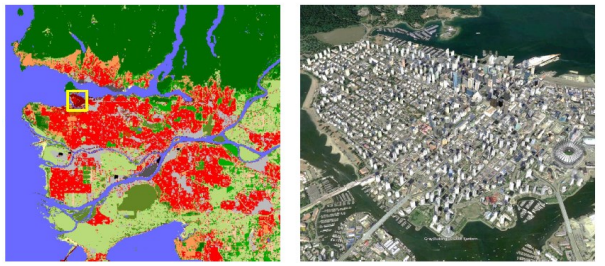
correct label \ predicted label	0	1	2	3	4	5	6	7	8	9
0	21									
1		25								
2			15							
3				76						
4					23					12
5						36				
6							24			
7								31		37
8									42	
9					8			11		53

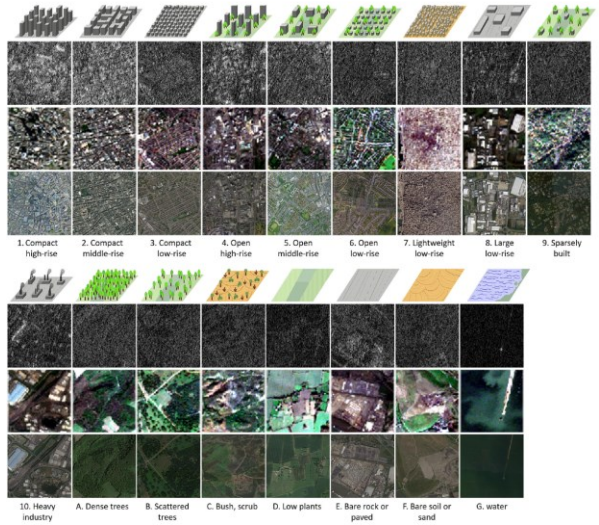
Excursus: DLR Earth Observation Use Case

[Xiao Xiang Zhu et al: So2Sat LCZ42: A Benchmark Dataset for the Classification of Global Local Climate Zones. **GRSM 2020**]

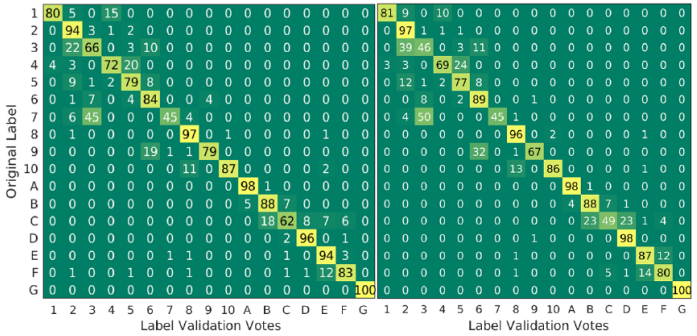
- **Data and ML Pipelines**
  - ESA Sentinel-1/2 datasets → 4PB/year
  - Training of local climate zone classifiers on So2Sat LCZ42 (15 experts, 400K instances, 10 labels each, 85% confidence, ~55GB H5)
  - **ML pipeline:** preprocessing, ResNet18, climate models

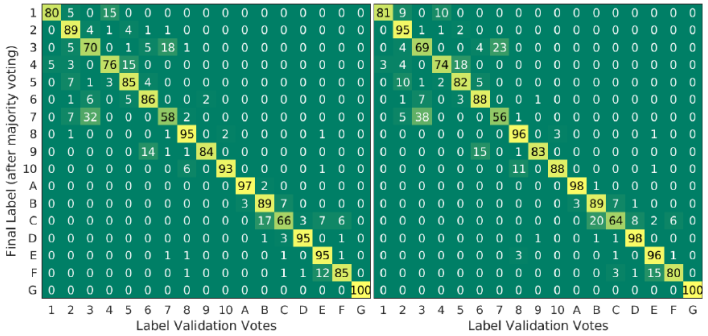
[So2Sat LC42 Dataset https://mediatum.ub.tum.de/1454690]





- **Label Creation/ Validation**
  - Team learning
  - Labeling w/ checks
  - Label validation
  - Quantitative validation w/ 10 expert votes on correctness

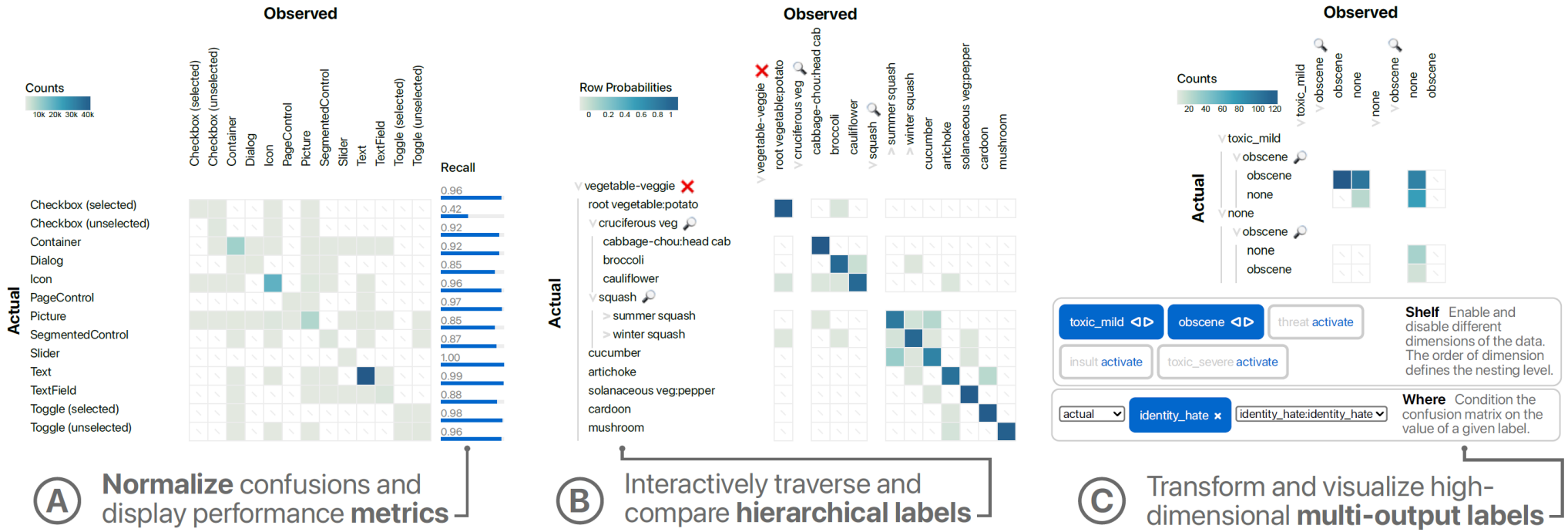




Confusion Matrices, cont.

[Jochen Görtler et al: **Neo:** Generalizing Confusion Matrix Visualization to Hierarchical and Multi-Output Labels. **CHI 2022** (1/25 best papers)]

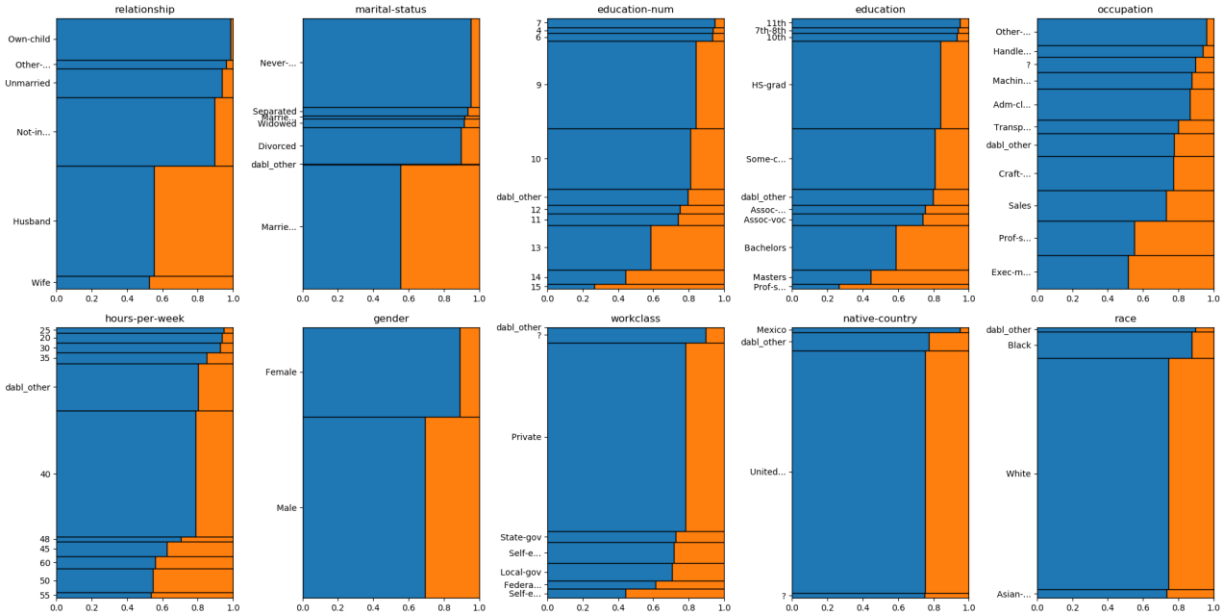
- **Generalized Confusion Matrices**
  - Hierarchical, Multi-label Data
  - Transform multi-label data: **conditioning**, **marginalization** (aggregation), and **nesting**



Excursus: dabl.plot

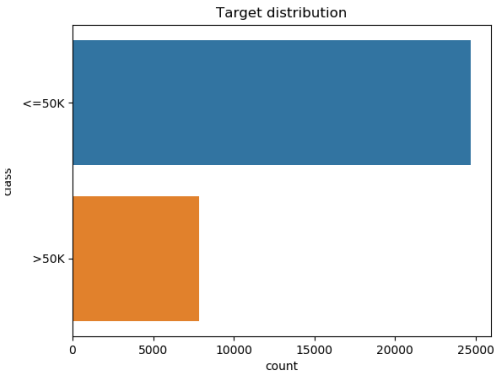
[Andreas Mueller: dabl – Taking the edge off of data science with dabl, **Data Umbrella 2022**, https://www.youtube.com/watch?v=h92RMJi4mRM]

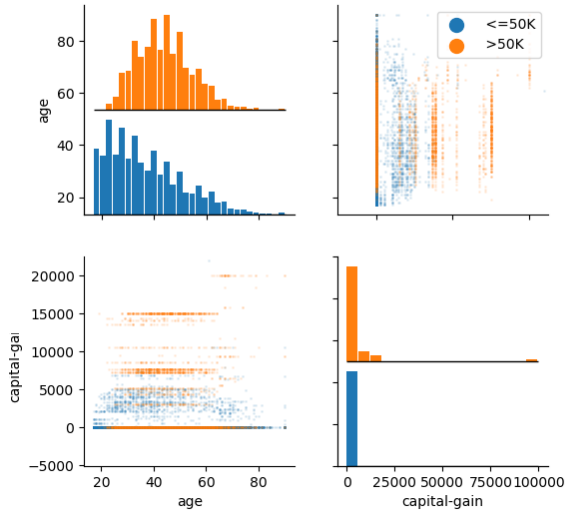

```
# adult dataset (>50K vs <=50K income)
data = pd.read_csv("adult.csv")
plot(data, "income")
```

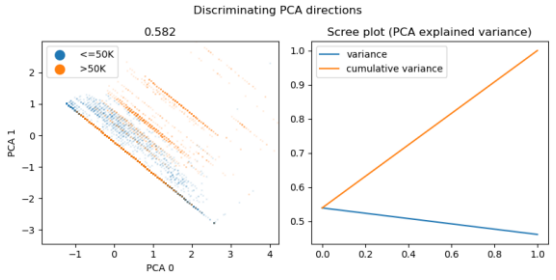


[https://amueller.github.io/dabl/dev/auto_examples/plot/plot_adult.html]

(mosaic plots)







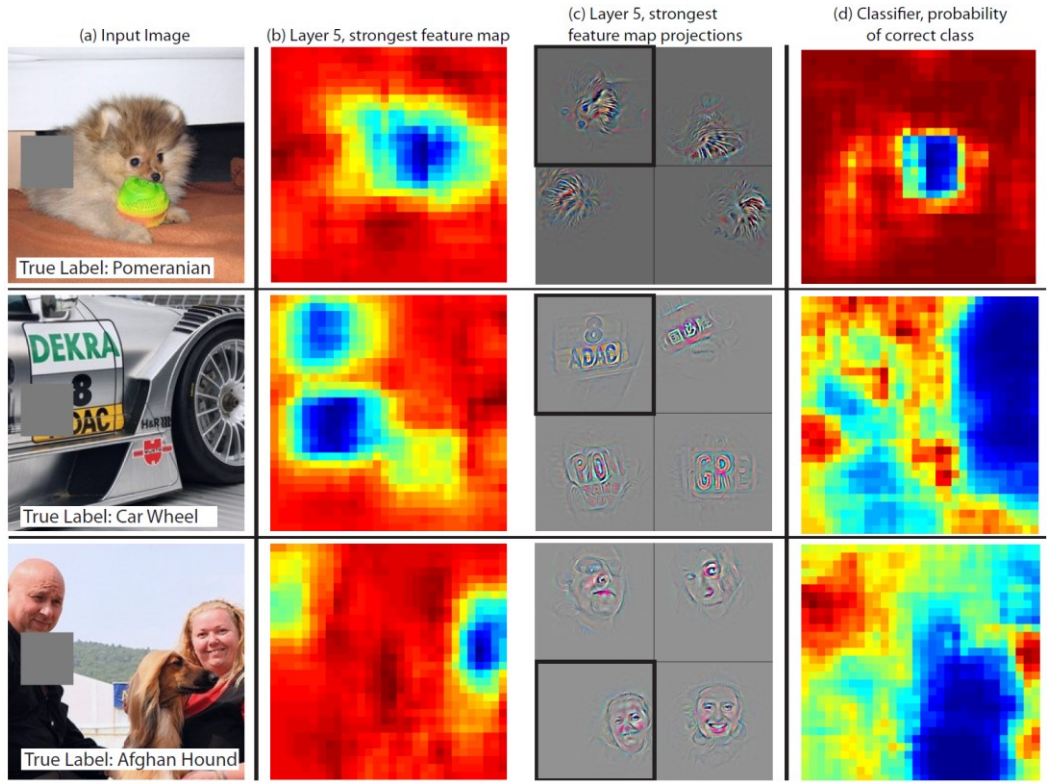
Understanding Other Basic Issues

- **Overfitting / Imbalance**
  - Compare train and test performance
  - ➔ **Algorithm-specific techniques:** regularization, pruning, loss, etc
- **Data Leakage**
  - Example: time-shifted external time series data (e.g., weather)
  - **Compare performance train/test vs production setting**
- **Covariance Shift** (features)
  - Distribution of features in training/test data different from production data
  - Reasons: **out-of-domain prediction, sample selection bias**
  - Examples: NLP, speech recognition, face/age recognition
- **Concept Drift** (features → labels)
  - **Gradual change of statistical properties** / dependencies (features-labels)
  - Requires re-training, parametric approaches for deciding when to retrain

Occlusion-Based Explanations

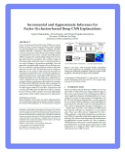
- **Occlusion Explanations**
  - Slide gray square over inputs
  - Measure how feature maps and classifier output changes

[Matthew D. Zeiler, Rob Fergus: Visualizing and Understanding Convolutional Networks. **ECCV 2014**]



- **Incremental Computation of Occlusion Explanations**
  - View CNN as white-box operator graph and operators as views
  - Materialize intermediate tensors and apply **incremental view maintenance**

[Supun Nakandala, Arun Kumar, and Yannis Papakonstantinou: Incremental and Approximate Inference forFaster Occlusion-based Deep CNN Explanations, **SIGMOD 2019**]



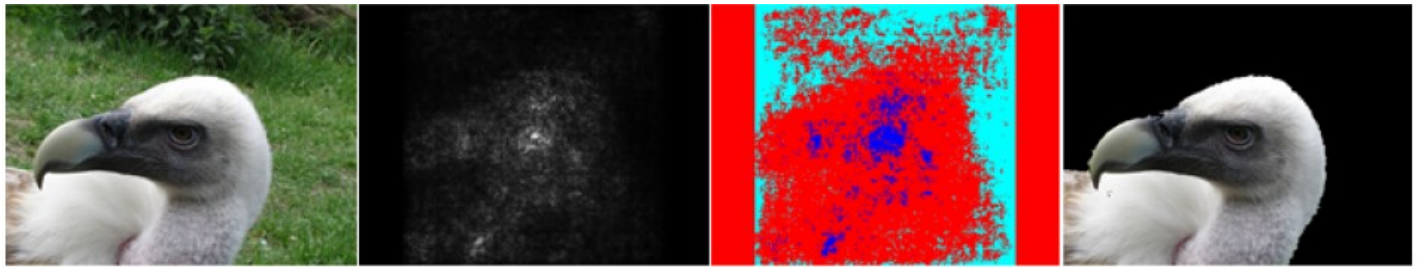
SIGMOD 2020 Research Highlight

Saliency Maps

[Karen Simonyan, Andrea Vedaldi, Andrew Zisserman: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. **ICLR Workshop 2014**]

- **Saliency Map**
  - Given input image and specific class
  - Compute saliency map of **class derivatives wrt input image**
  - Approximated w/ a linear function (Taylor expansion)

- **Unsupervised Image Segmentation**



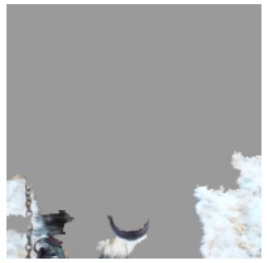
Example Model Anomalies

"silent but severe problems"

- **#1 Wolf Detection based on snow cover**

[Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin: Why Should I Trust You?: Explaining the Predictions of Any Classifier, **KDD 2016**]

(a) Husky classified as wolf

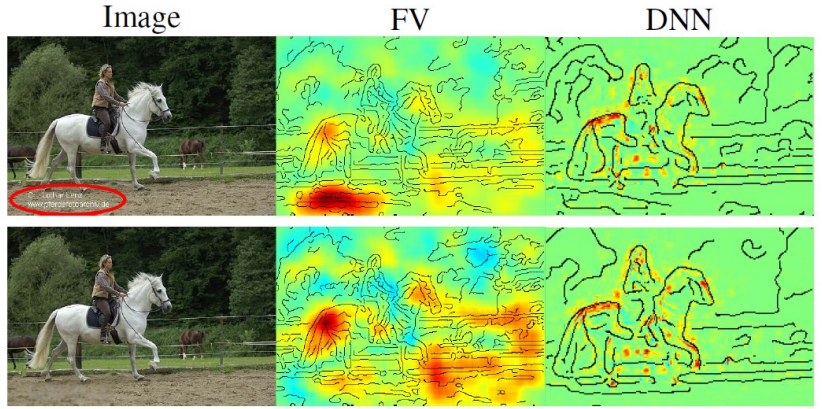


(b) Explanation

12/27 → 25/27

- **#2 Horse Detection based on image watermarks**
  - Layer-wise relevance propagation

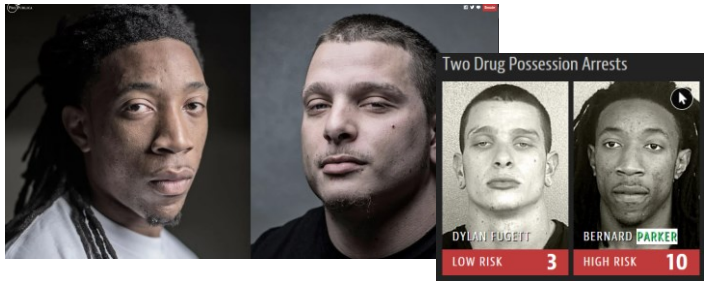
[Sebastian Lapuschkin et al.: Analyzing Classifiers: Fisher Vectors and Deep Neural Networks, **CVPR 2016**]



- **#3 Race-biased Jail Risk Assessment**

#BlackLivesMatter

[Julia Angwin et al: Machine Bias – There's software used across the country to predict future criminals. And it's biased against blacks, **2016**, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing]



Explanation Tables

- **Motivation**
  - Generate **succinct decision rules** from data
  - **Problem:** Decision tree rules do not overlap by def
  - Example athlete's exercise log: "Goal met" → 7 vs 7

- **Explanation Tables**
  - **Find smallest explanation** table subject to max KL divergence threshold
  - Greedy and sampling algorithms

[Kareem El Gebaly, Parag Agrawal, Lukasz Golab, Flip Korn, Divesh Srivastava: Interpretable and Informative Explanations of Outcomes. **PVLDB 2014**]

id	day	time	meal	goal met?
1	Fri	Dawn	Banana	Yes
2	Fri	Night	Green salad	Yes
3	Sun	Dusk	Oatmeal	Yes
4	Sun	Morning	Banana	Yes
5	Mon	Afternoon	Oatmeal	Yes
6	Mon	Midday	Banana	Yes
7	Tue	Morning	Green salad	No
8	Wed	Night	Burgers	No
9	Thu	Dawn	Oatmeal	Yes
10	Sat	Afternoon	Nuts	No
11	Sat	Dawn	Banana	No
12	Sat	Dawn	Oatmeal	No
13	Sat	Dusk	Rice	No
14	Sat	Midday	Toast	No

day	time	meal	goal met=Yes?	count
*	*	*	.5	14
Sat	*	*	0	5
*	*	Banana	.75	4
*	*	Oatmeal	.75	4

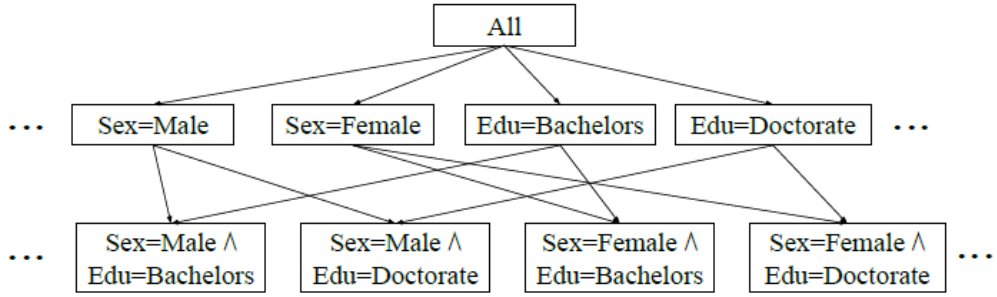
SliceFinder

[Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, Steven Euijong Whang: Automated Data Slicing for Model Validation: A Big Data - AI Integration Approach. **ICDE2019/TKDE2020**]

- **Problem Formulation**
  - Data slice: S^{DG} := D=PhD AND G=female (subsets of features)
  - **Find top-k data slices where model performs worse** than average
  - **Ordering by**
    - Increasing number of literals,
    - Decreasing slice size,
      and decreasing effect size (difference S vs $\neg S$)
  - **Subject to:** minimum effect size threshold T, statistical significance (Welch's t-test),
    a dominance constraint (no coarser slice satisfies 1 and 2) via

"find largest error vs find large slices"

- **Existing Algorithms**
  - Preparation: Binning + One-Hot Encoding
  - **#1** Clustering → slices
  - **#2** Decision tree training
  - **#3 Lattice search** with
    heuristic, level-wise termination



SliceLine for Model Debugging

[**Credit:** sliceline, Silicon Valley, HBO]

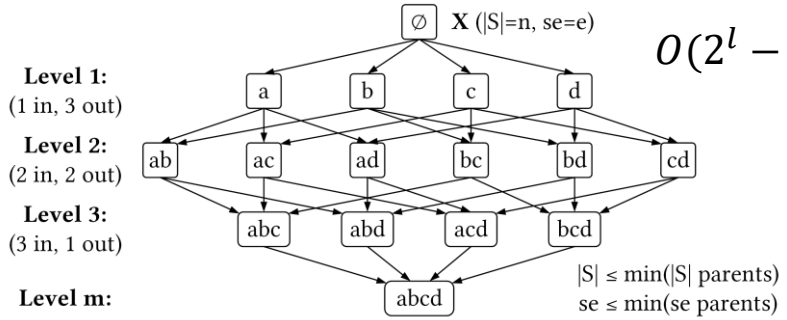
- **Problem Formulation**
  - Intuitive slice scoring function
  - Exact **top-k slice finding**
  - $|S| \geq \sigma \wedge sc(S) > 0, \alpha \in (0,1]$

$$sc = \alpha\left(\frac{\bar{e}(S)}{\bar{e}(X)} - 1\right) - (1-\alpha)\left(\frac{|X|}{|S|} - 1\right)$$

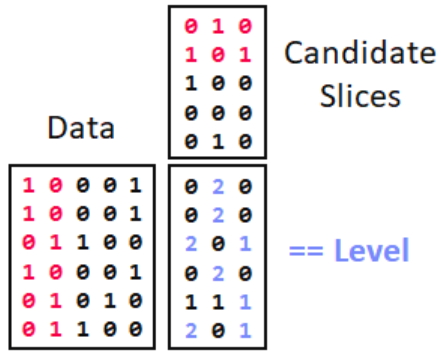
$$= \alpha\left(\frac{|X|}{|S|} \cdot \frac{\sum_{i=1}^{|S|} es_i}{\sum_{i=1}^{|X|} e_i} - 1\right) - (1-\alpha)\left(\frac{|X|}{|S|} - 1\right)$$

slice error **slice size**

- **Properties & Pruning**
  - Monotonicity of slice sizes, errors
  - Upper bound sizes/errors/scores
    → pruning & termination



$$O(2^l - \sum_{j=1}^{m} 2^{d_j} + l + m)$$

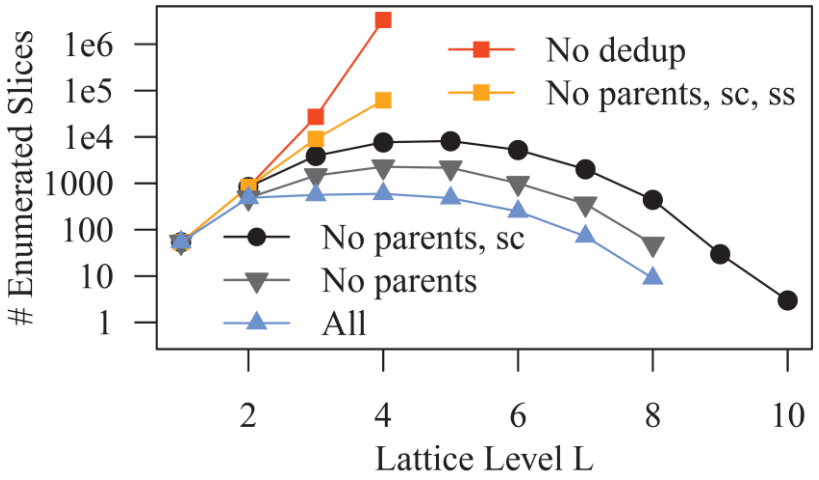


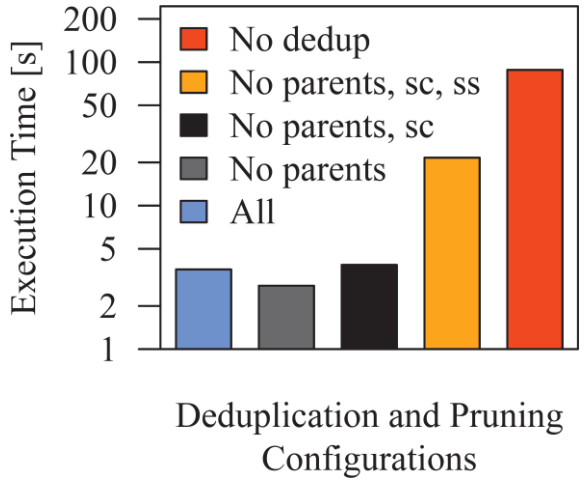
- **Linear-Algebra-based Slice Finding**
  - Recoded/binned matrix **X**, error vector e
  - Vectorized implementation in linear algebra (join & eval via sparse-sparse matmult)
  - Local and distributed task/data-parallel execution

SliceLine – Experiments

[Svetlana Sagadeeva, Matthias Boehm: SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging, **SIGMOD 2021**]

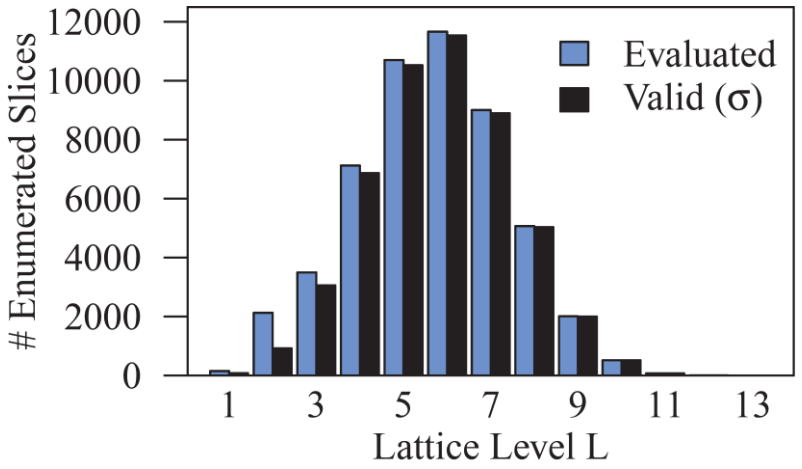
Salary 2x2





Effective Pruning
(#evaluated close to #valid)

Adult

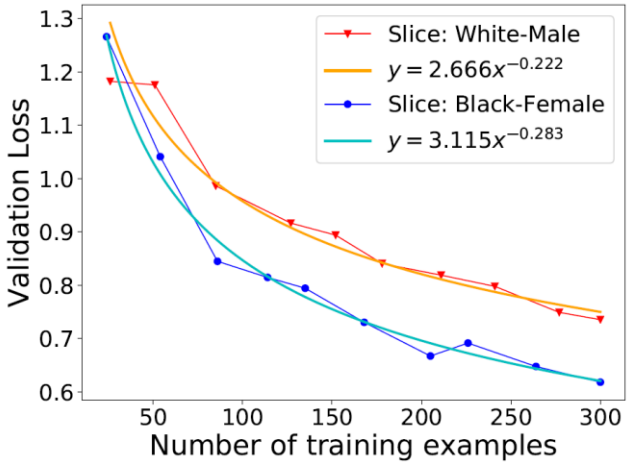


Practical Performance
(39s until termination at level 12)

Slice Tuner

[Ki Hyun Tae, Steven Euijong Whang: Slice Tuner: A Selective Data Acquisition Framework for Accurate and Fair Machine Learning Models, **SIGMOD 2021**]

- **Motivation**
  - Root cause of unfairness: **bias in training data**
  - **Selective Data Acquisition** for model accuracy and fairness
  - Different slices w/ different learning curves
    → **Learning curve fitting**



- **Problem Formulation**

Minimize total loss of slices — Penalize underperforming slices

Convex optimization problem

$$\min \sum_{i=1}^{n} b_i(|s_i| + d_i)^{-a_i} + \lambda \sum_{i=1}^{n} \max\left\{0, \frac{b_i(|s_i| + d_i)^{-a_i}}{A} - 1\right\}$$

$$\text{subject to} \sum_{i=1}^{n} C(s_i) \times d_i = B$$

Budget of acquisition costs

Model Assertions

[Daniel Kang, Deepti Raghavan, Peter Bailis, Matei Zaharia: Model Assertions for Debugging Machine Learning, **NeurIPS Workshop ML Systems, 2018**]

- **Motivation**
  - ML models might fail in complex ways that are not captured in loss function
  - Inspired by assertions in SW dev → Model assertions via Python rules

Example:
Flickering of object detection

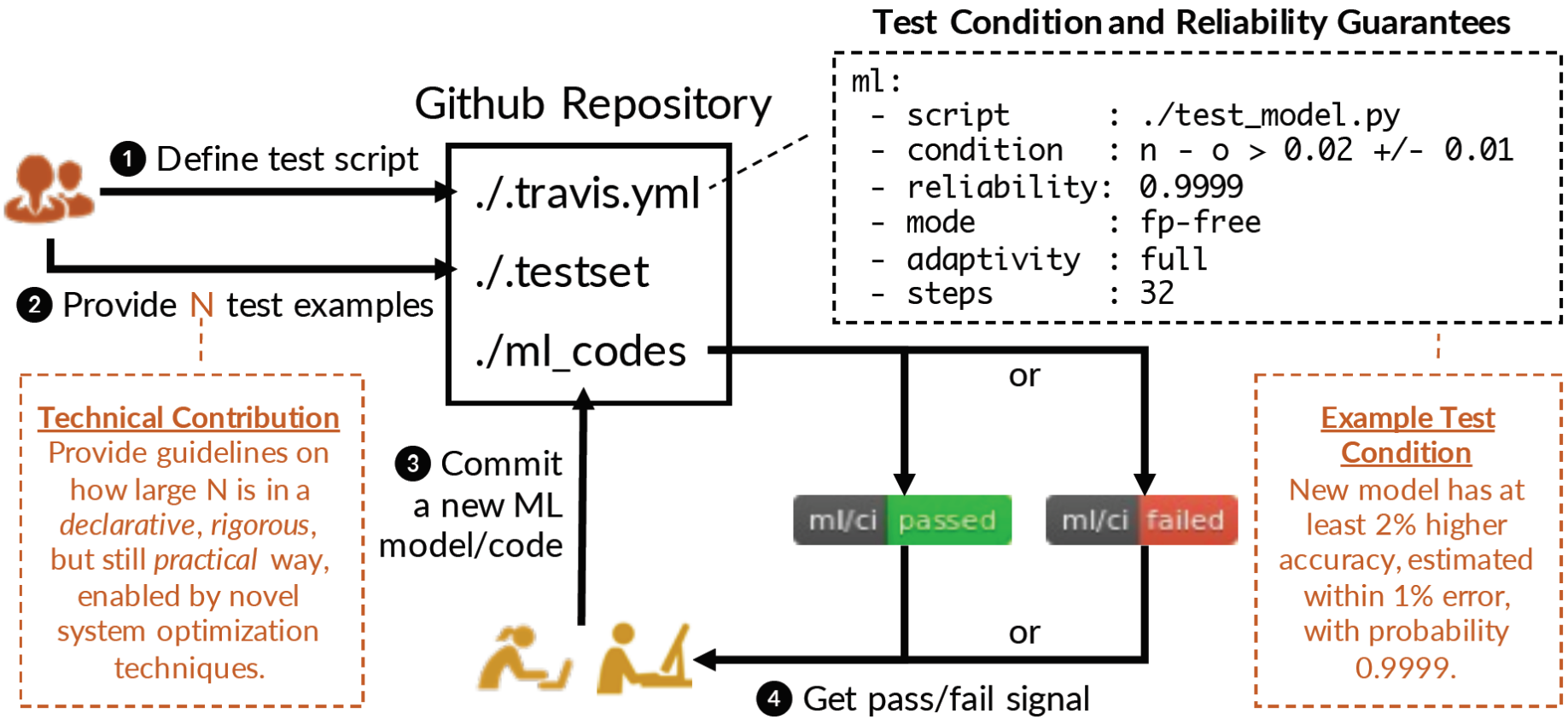
(a) Frame 1, base SSD (b) Frame 2, base SSD (c) Frame 3, base SSD

- **Assertion Use Cases**
  - **#1 Runtime monitoring** (collect statistics on incorrect behavior)
  - **#2 Corrective Action** (trigger corrections at runtime) → **but how in retrospect?**
  - **#3 Active Learning** (decide which difficult data points to give to user)
  - **#4 Weak supervision** (propose alternative labels and use for retraining)

Continuous Integration

[Cedric Renggli, Bojan Karlaš, Bolin Ding, Feng Liu, Kevin Schawinski, Wentao Wu, Ce Zhang: Continuous Integration of Machine Learning Models with ease.ml/ci: Towards a Rigorous Yet Practical Treatment, **SysML 2019**]

- **System Architecture**
  ease.ml/ci



Explainability

[Hima Lakkaraju, Julius Adebayo, Sameer Singh: Explaining Machine Learning Predictions: State-of-the-art, Challenges, and Opportunities, **NeurIPS 2020** Tutorial, https://explainml-tutorial.github.io/neurips20]

- **Motivation**
  - Explain predictions via inputs for **model understanding** & **transparency**
  - Utilize model debugging and other tools

Interpretability ← → Accuracy

Prefer simpler models if accuracy sufficient!

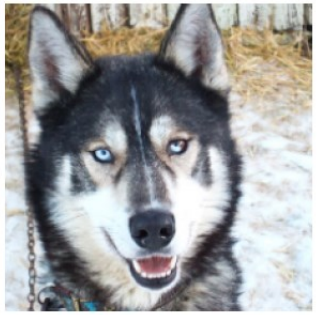
- **#1 Interpretable Models**
  - Linear models, tree-based models, rule-based models
  - Weights and decision rules
- **#2 Post-hoc Explanations**
  - Complex deep neural networks or very large models → black box models
  - Build simple models for explaining **any** complex models
- **Types of Explanations**
  - Model parameters, example predictions, summarization
  - **Most important features**/data, how to flip model predictions

Multi-modal Interpretability:
https://captum.ai/

LIME: Sparse, Linear Explanations

[Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin: Why Should I Trust You?: Explaining the Predictions of Any Classifier, **KDD 2016**]

- **LIME Overview**
  - Model agnostic explanations
  - Identify important dimension and present their relative importance
  - **Sample perturbations** of prediction input (e.g., hide parts of image, attribute values)
  - **Locally weighted regression**



(a) Husky classified as wolf

(b) Explanation

- **LIME Objective**
  - Various hyper-parameters
  - Heuristics / HP optimization

Loss Function — Regularizer

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Linear Models — Local Kernel

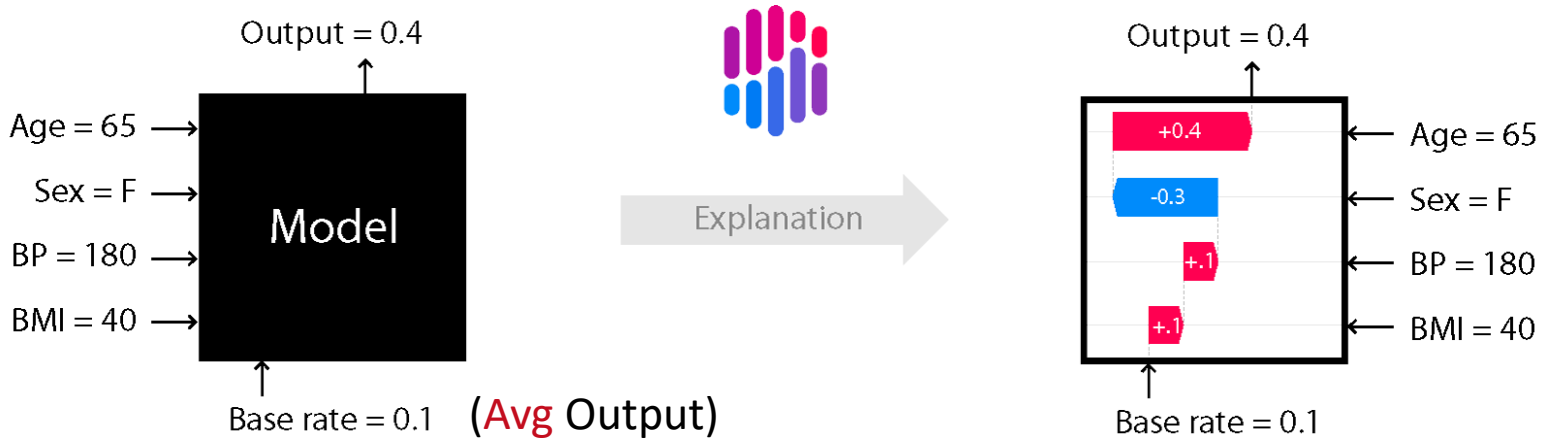
SHAP: Shapley Additive Explanations

[Scott M. Lundberg, Su-In Lee: A Unified Approach to Interpreting Model Predictions. **NeurIPS 2017**]

- **SHAP Overview**
  - Additive feature importance (local accuracy) := **sum of feature contributions**
  - **Unification** of LIME, Shapley sampling/regression values, QII, DeepLIFT, layer-wise relevance propagation, tree interpreter
  - Estimate Shapley values using **linear regression**

[Scott M. Lundberg: Explainable AI for Science and Medicine, https://www.youtube.com/watch?v=B-c8tIgchu0]

- **SHAP Tooling**

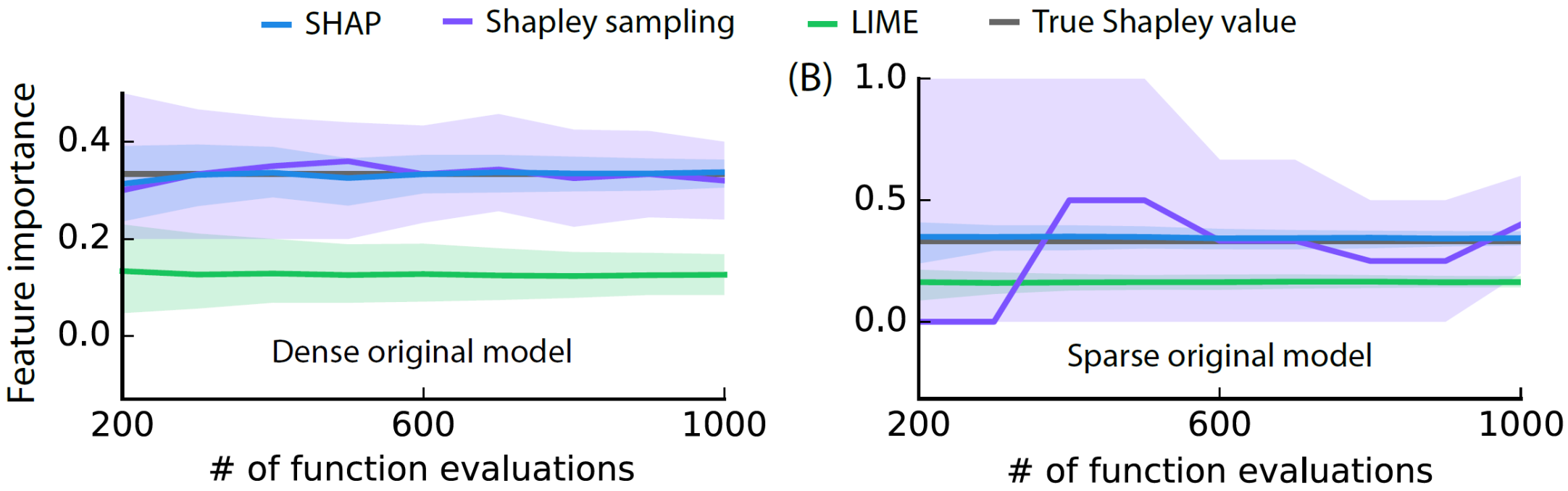


[https://shap.readthedocs.io/en/latest/index.html]

Marginal contributions

SHAP: Shapley Additive Explanations, cont.

[Scott M. Lundberg, Su-In Lee: A Unified Approach to Interpreting Model Predictions. **NeurIPS 2017**]



- **Other Shapely-related Work:**
  - Quantitative Input Influence (QII)

[Anupam Datta, Shayak Sen, Yair Zick: Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. **IEEE SSP 2016,** https://doi.org/10.1109/SP.2016.42]

Model Bias & Fairness

Focus on Applications, Fairness, Ethics, Responsibility
Fairness Metrics and Constraints
Employs Model Debugging & Explainability Techniques

Sources of Bias

- **Environment**
  - **Selection Bias:** Differences in study participation, data availability, and measurement effort
  - Test environment, project team, cultural context → **different context**
- **Data Collection**
  - **Sample Bias:** collected data not representative of application
  - **Observer Bias** / **Confirmation Bias:** subjective judgment leaks into measurement and analysis → **transparency and critical feedback**
- **Training Dataset**
  - **Data Bias:** e.g., not missing at random (NMAR) values
  - **Feature Selection Bias:** manual or automatic during data preparation

→ **Design ML Systems & applications w/ awareness of potential bias**

Excursus: DLR Earth Observation Use Case, cont.

[Xiao Xiang Zhu et al: So2Sat LCZ42: A Benchmark Dataset for the Classification of Global Local Climate Zones. **GRSM 2020**]

For the evaluation, we have chosen a subset of 10 European cities (shown in Table II) from the group of cities we labeled. The choice was based on the following three rationales:

- All our labeling experts have lived in Europe for a significant number of years. This ensures familiarity with the general morphological appearance of European cities.
- Google Earth provides detailed 3D models for the 10 cities, which is of great help in determining the approximate height of urban objects. This is necessary to be able to distinguish between low-rise, mid-rise, and high-rise classes.
- As previously mentioned, LCZ labeling is very labor-intensive. Reducing the evaluation set to 10 cities allowed us to generate more individual votes per polygon for better statistics.

Unfortunately, not many European cities contain LCZ class 7 (light-weight low-rise), which mostly describes informal settlements (e.g., slums). Therefore, we included the polygons of class 7 for an additional 9 cities that are representative of the 9 major non-European geographical regions of the world (see Table III).

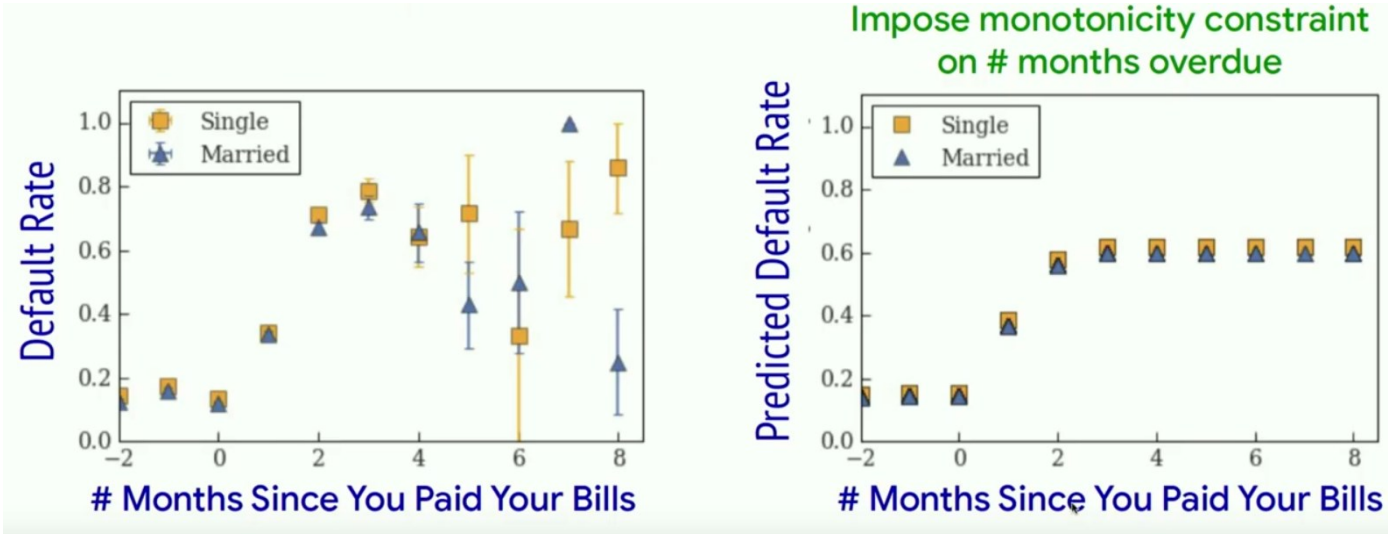
Environment / Context
→ **Biased Data Collection**

→ **Awareness and Conscious Bias Mitigation**
→ **Remaining Bias?**

Debugging Bias and Fairness

- **Fairness**
  - Validate and ensure fairness with regard to sensitive features (unbiased)
  - Use occlusion and saliency maps to characterize and compare groups
- **Enforcing Fairness**
  - Use constraints to enforce certain properties (e.g., monotonicity, smoothness)
  - **Example:**
    late payment → credit score

[Maya Gupta: How Do We Make AI Fair? **SysML 2019**]



Group Fairness Constraints

[H. Zhang et al: OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning, **SIGMOD 2021**]

- **#1 Statistical Parity**
  - Independence of model from groups
  - **Equal probability outcome** across groups
- **#2 False Positive Rate Parity**
  - Independence of model from groups
  - **Conditioned on true label y=0**
- **#3 False Negative Rate Parity**
  - Independence of model from groups
  - **Conditioned on true label y=1**
- **#4 False Omission Rate Parity**
  - Independence of true labels from groups
  - **Conditioned on negative prediction h=0**

$$\forall g_i, g_j \in G: \quad P(\hat{y} = 1 | g_i) \approx P(\hat{y} = 1 | g_j)$$

#2+#3 Equalized Odds

$$\forall g_i, g_j \in G: \quad P(\hat{y} = 1 | g_i, y = 0) \approx P(\hat{y} = 1 | g_j, y = 0)$$

$$\forall g_i, g_j \in G: \quad P(\hat{y} = 0 | g_i, y = 1) \approx P(\hat{y} = 0 | g_j, y = 1)$$

$$\forall g_i, g_j \in G: \quad P(y = 1 | g_i, \hat{y} = 0) \approx P(y = 1 | g_j, \hat{y} = 0)$$

Group Fairness Constraints, cont.

- **#5 False Discovery Rate Parity**
  - Independence of true labels from groups
  - Conditioned on negative prediction h=1
  - **#4+#5 Predictive Parity**

$$\forall g_i, g_j \in G: \\ P(y = 1 | g_i, \hat{y} = 1) \approx P(y = 1 | g_j, \hat{y} = 1)$$

- **#6 Misclassification Rate Parity**
  - Equal misclassification rate across groups

$$\forall g_i, g_j \in G: \\ P(\hat{y} = y | g_i) \approx P(\hat{y} = y | g_j)$$

- **Others**
  - Individual fairness
    → relationship to **differential privacy**
  - Causal fairness

[Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, Richard S. Zemel: Fairness through awareness. **ITCS 2012**]

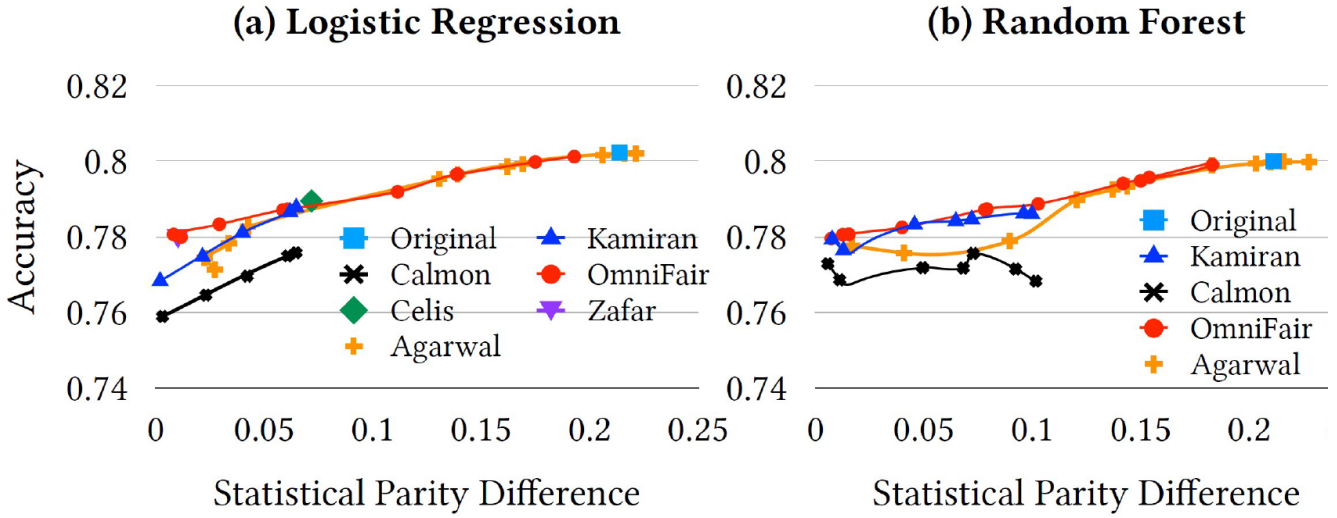
Ensuring Fairness

[H. Zhang et al: OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning, **SIGMOD 2021**]

- **Problem Formulation**
  - A **fairness specification** is given by a triplet (g, f, ε) and induces (|g(D)|choose 2) **fairness constraints** on pairs of groups
  - A fairness spec is satisfied by a classifier h on D iff all induced fairness constraints are satisfied, i.e., $\forall gi,gj \in g(D)$, $|f(h,gi)-f(h,gj)| \leq \varepsilon$
  - **Unconstrained optimization problem**

max accuracy **s.t. fairness** ➡ **max** accuracy **+ fairness**

- **Results**
  - Adult dataset
  - Model-agnostic
  - Similar Accuracy

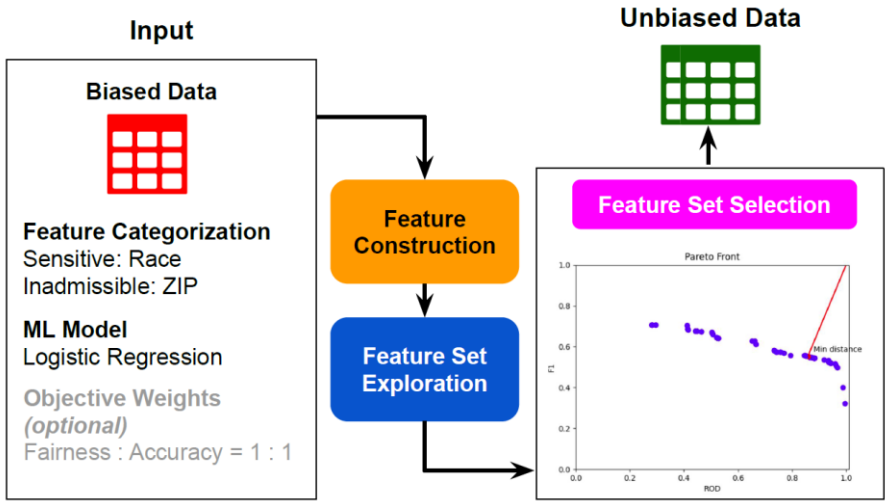


Fairness-aware Feature Engineering

- **Excursus: Diversity Strategy TU Berlin**
  - "diversity at TU Berlin is understood in terms of **commitment, opportunity and potential** […]; attributions which are often associated with discrimination such as age, disability and chronic illness, ethnic origin, gender, social background, sexual orientation as well as religion and political or other opinion."
  - Such features **should not be used for hiring decisions**, but needed for group fairness

[https://www.static.tu.berlin/fileadmin/www/10000000/Arbeiten/Wichtige_Dokumente/Diversity_Strategy_TU_Berlin.pdf]

- **FairExp (FAIRness EXPlorer)**
  - **Problem: Sensitive features** and features correlated to them
  - **Dropping features** or introducing new tuples loses too much accuracy
  - **Feature Construction**: +, *, 1/, −1*, log, one-hot
  - **Feature Set Exploration/Selection**



[Ricardo Salazar, Felix Neutatz, Ziawasch Abedjan: Automated Feature Engineering for Algorithmic Fairness. **PVLDB 2021**]

Excursus: EU Policy

[European Commission: LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, **04/2021**]

The Commission examined different policy options to achieve the general objective of the proposal, which is to **ensure the proper functioning of the single market** by creating the conditions for the development and use of trustworthy AI in the Union.

Four policy options of different degrees of regulatory intervention were assessed:

- **Option 1**: EU legislative instrument setting up a voluntary labelling scheme;
- **Option 2**: a sectoral, "ad-hoc" approach;
- **Option 3**: Horizontal EU legislative instrument following a proportionate risk-based approach;
- **Option 3+**: Horizontal EU legislative instrument following a proportionate risk-based approach + codes of conduct for non-high-risk AI systems;
- **Option 4**: Horizontal EU legislative instrument establishing mandatory requirements for all AI systems, irrespective of the risk they pose.

➔ "The **preferred option is option 3+**, a regulatory framework for high-risk AI systems only, with the possibility for […] non-high-risk AI systems to follow a code of conduct."

Summary & QA

- **Model Debugging and Explainability**
- **Model Bias & Fairness Constraints**

[Julia Stoyanovich: Responsible Data Science, https://dataresponsibly.github.io/courses/spring20/]

"**Bottom line:** we will learn that many of the problems are **socio-technical**, and so cannot be "solved" with technology alone."

- **Next Lectures** (Part B)
  - **12 Model Serving Systems and Techniques** [Jul 13]
  - **Exams** [Jul 14 - 28]