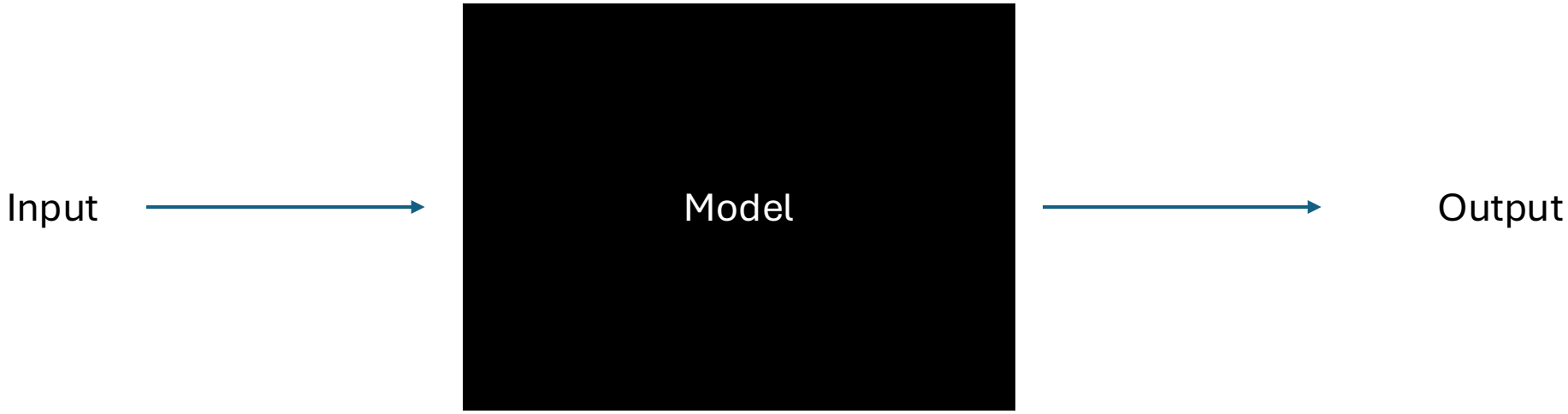
Explainability, interpretability with Shapely Values and LIME

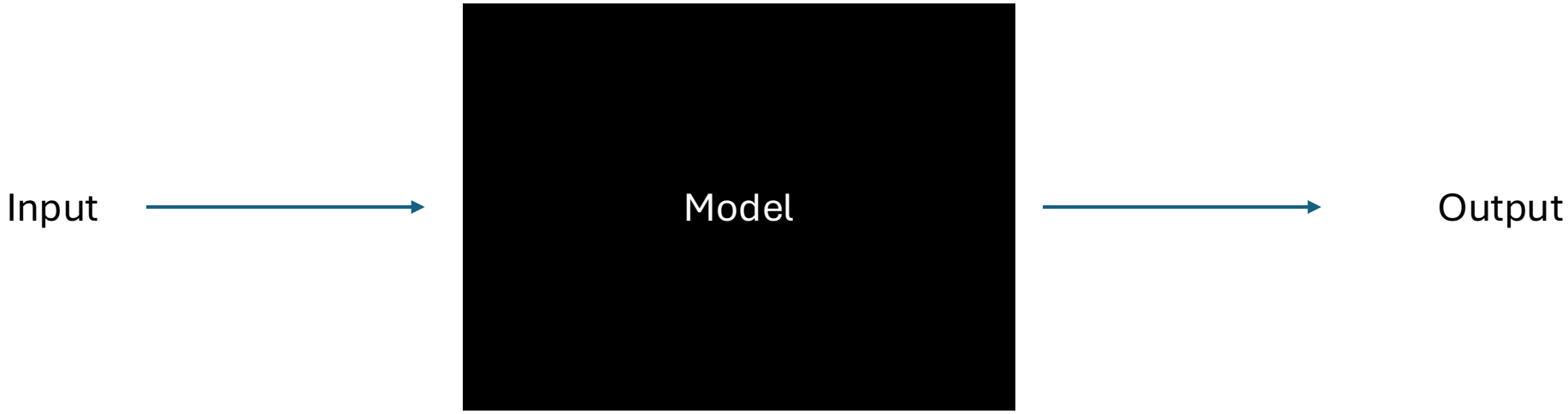
Tarek Zikry

BCB 726 11/17

Machine learning models can be black boxes

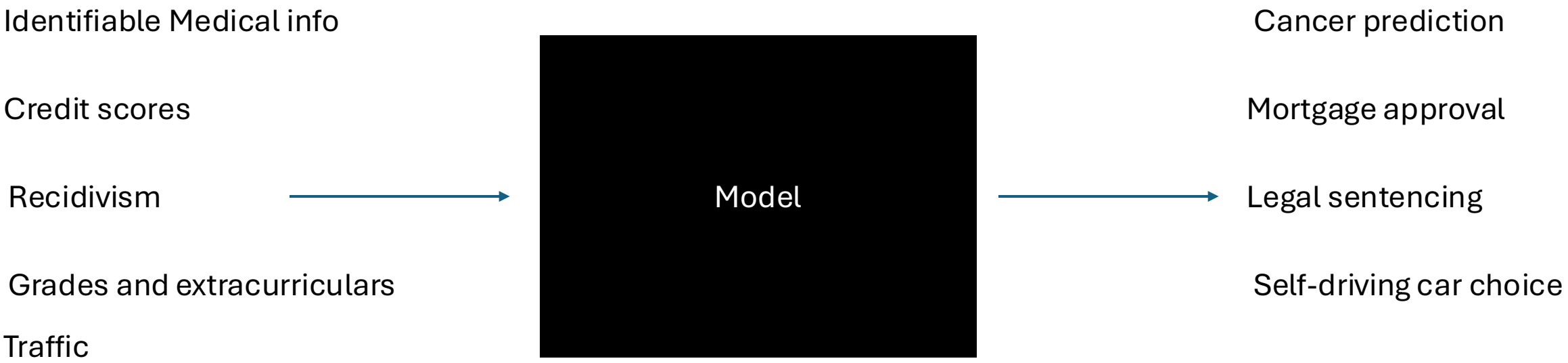


Machine learning models can be black boxes



- A black box predictive model is a formula that is either too complicated to understand, or proprietary
- This becomes a problem when the input and output are sensitive, or used for high-impact decision making

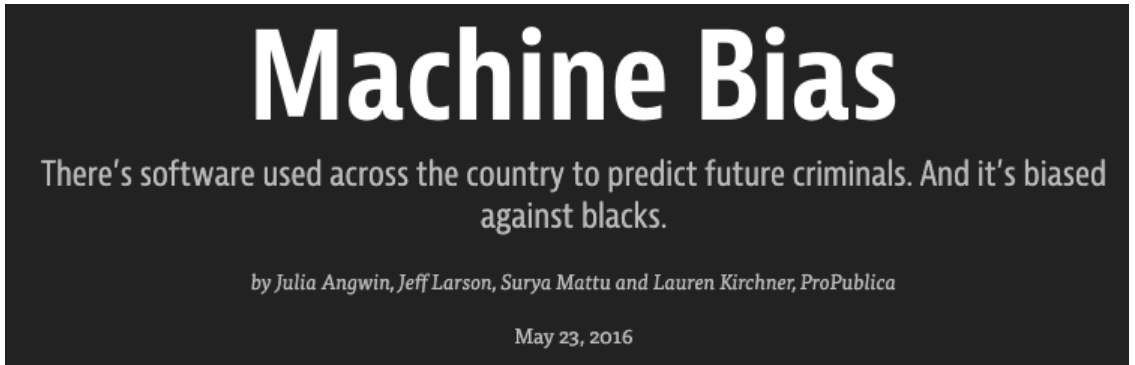
Machine learning models can be black boxes



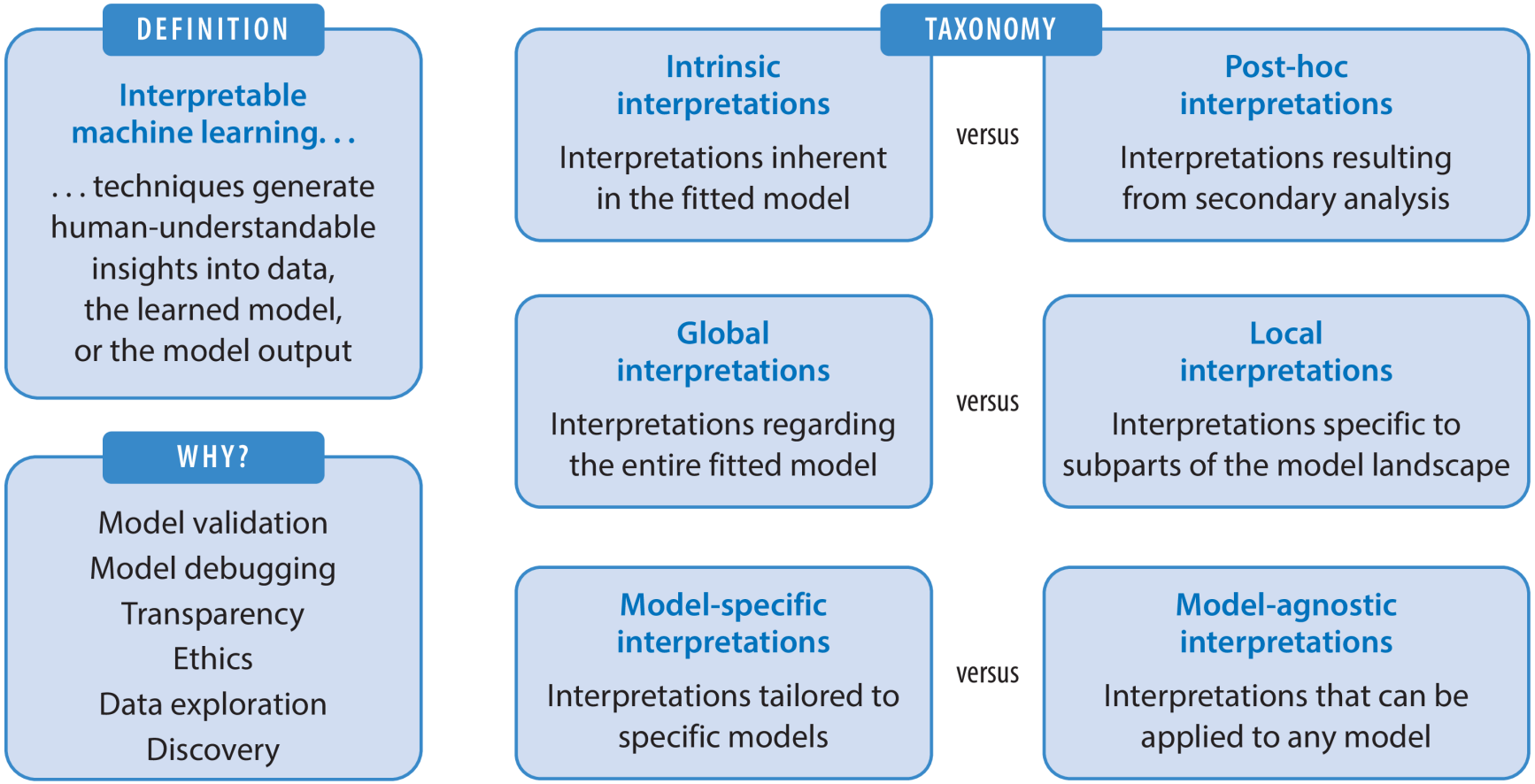
- This becomes a problem when the input and output are sensitive, and/or used for high-impact decision making

Real world example

- COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) – proprietary black model model widely used in the U.S. Justice System for parole and bail decisions
- ProPublica (Larson et al. 2016) found that black defendants were almost twice as likely to be falsely predicted as recommitting crimes as white defendants
- Because the model is not publicly available, researchers analyzing fairness relied on post-hoc methods



Terminology



What is an interpretation?

- Depends on the ML task
- Classification/regression
  - Feature importances
- Clustering
  - Groupings
- Dimension reduction
  - Nearest neighbors of each point/location in embedding
- Graph learning
  - Adjacent nodes/edges, graph structure

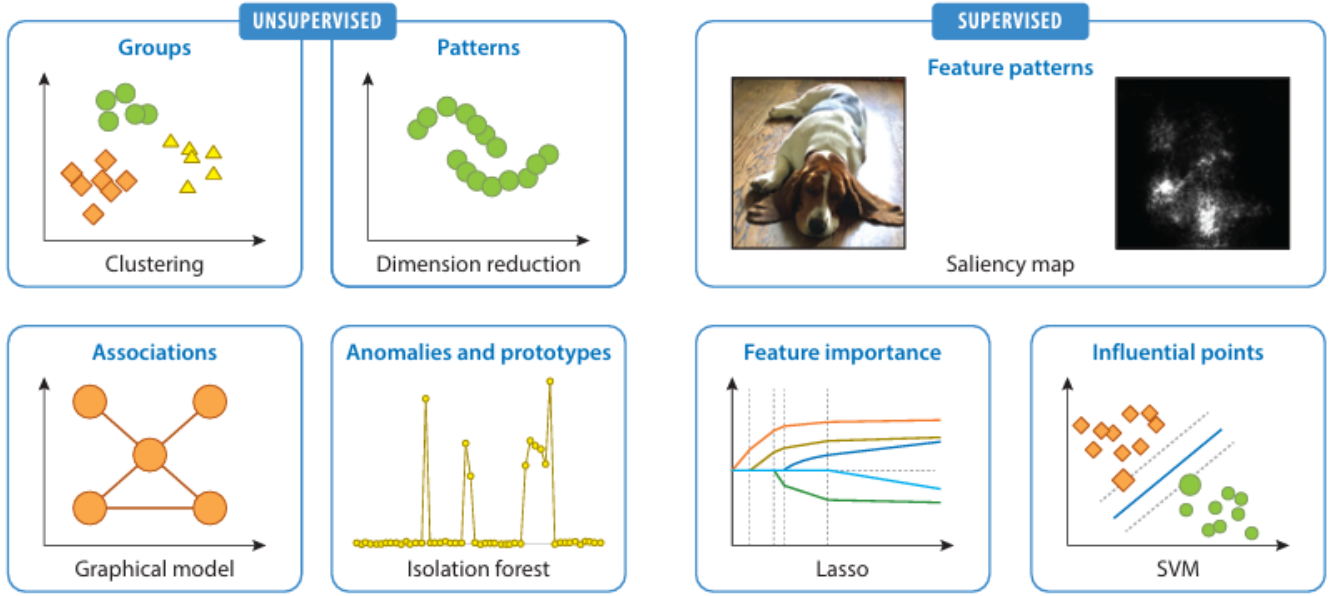


Figure 2

Overview of the broad types of unsupervised and supervised data-driven discoveries that can be made using interpretable machine learning techniques, with some simple graphic examples.

Linear interpretations

$$\hat{f}(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

- For a particular observation, we want to know how each feature affects the prediction $\hat{f}(x)$
- The betas are the weights corresponding to each feature (1, …, p)
- The contribution of the j-th feature to the prediction is then: $\phi_j(\hat{f}) = \beta_j x_j - \beta_j E[X_j]$
  - $\beta_j E[X_j]$ is the mean effect estimate for feature j
  - The contribution is the difference between the feature effect and the average effect

Linear interpretations

$$\hat{f}(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

- For a particular observation, we want to know how each feature affects the prediction $\hat{f}(x)$
- The betas are the weights corresponding to each feature (1, …, p)
- The contribution of the j-th feature to the prediction is then: $\phi_j(\hat{f}) = \beta_j x_j - \beta_j E[X_j]$
  - $\beta_j E[X_j]$ is the mean effect estimate for feature j
  - The contribution is the difference between the feature effect and the average effect
- If we sum all the feature contributions for a single observation:

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(\hat{f}) &= \sum_{j=1}^{p}(\beta_j x_j - \beta_j \mathbb{E}[X_j]) \\ &= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} \beta_j \mathbb{E}[X_j]) \\ &= \hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{X})]\end{aligned}$$

- This is the predicted value minus the average predicted value for all observations
- Can we do this for other types of models?

Shapley values

$$\phi_j(val) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} (val\,(S \cup \{j\}) - val(S))$$

- Shapley value of a feature is its contribution to the payout, summer and weighted over all feature combinations
- S is subset of features used in the model
- x is the vector of feature values of observation to be explained
- p features

$$val_{\mathbf{x}}(S) = \int \hat{f}(x_1, \ldots, x_p) d\mathbb{P}_{X_C} - \mathbb{E}[\hat{f}(\mathbf{X})]$$

- $val_x(S)$ prediction for features in S that are marginalized over features x_c (features not included in S)

Shapley properties

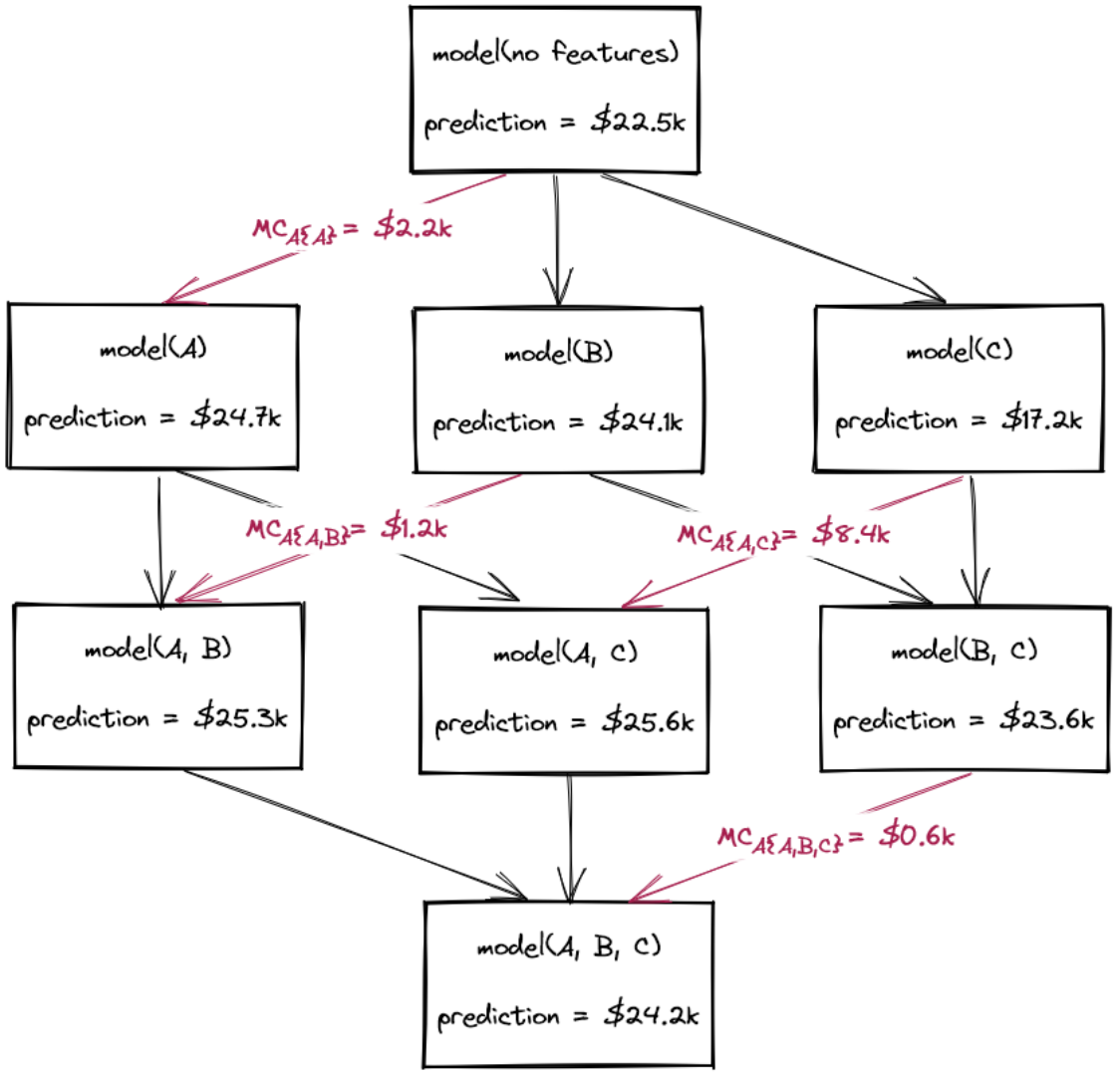
- Efficiency
  - Feature contributions must add up to the difference of the prediction for observation x and the average

$$\sum_{j=1}^{p} \phi_j = \hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{X})]$$

- Symmetry
  - Contributions of two features j and k should be the same if the contribute equally to all possible coalitions
- Dummy
  - Feature j that does not change the predicted value should have a value of 0
- Additivity
  - For a game with combine payouts $val_1 + val_2$, the Shapley values are $\phi_1 + \phi_2$
  - For instance, Shapley value of random forest is the sum of Shapley values from each tree

Shapley values: Example

- You have trained ML model to predict apartment prices, average prediction is $22,500
- You have a certain apartment, 50 m^2 (A), on 2nd floor (B), and near a park (C)
  - Model predicts $24,200
  - How much does each characteristics contribute to the price, as *compared to the average prediction*?
  - $val_A(x) = \left(\frac{1}{3}\right) * 2200 + \left(\frac{1}{6}\right) * 1200 + \left(\frac{1}{6}\right) * 8400 + \left(\frac{1}{3}\right) * 600 = \$2{,}550$
    - Weights are reciprocal of count of connections at each layer



Aidan Cooper

SHapley Additive exPlanations (SHAP)

- Shapley values require retraining your model 2^p times
- SHAP avoids this by using approximation methods to make it computationally feasible
- Represent explanations represented as a linear model

$$g(\mathbf{z}') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

- g is the explanation model, $z' = (z'_1, \ldots, z'_M)^T \in \{0,1\}^M$ coalition vector (simplified features with some mapping) – if all present this collapses to the Shapley value itself
- ϕ_j contribution

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
slund1@cs.washington.edu

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
suinlee@cs.washington.edu

Fua et al. 2012

SHAP properties

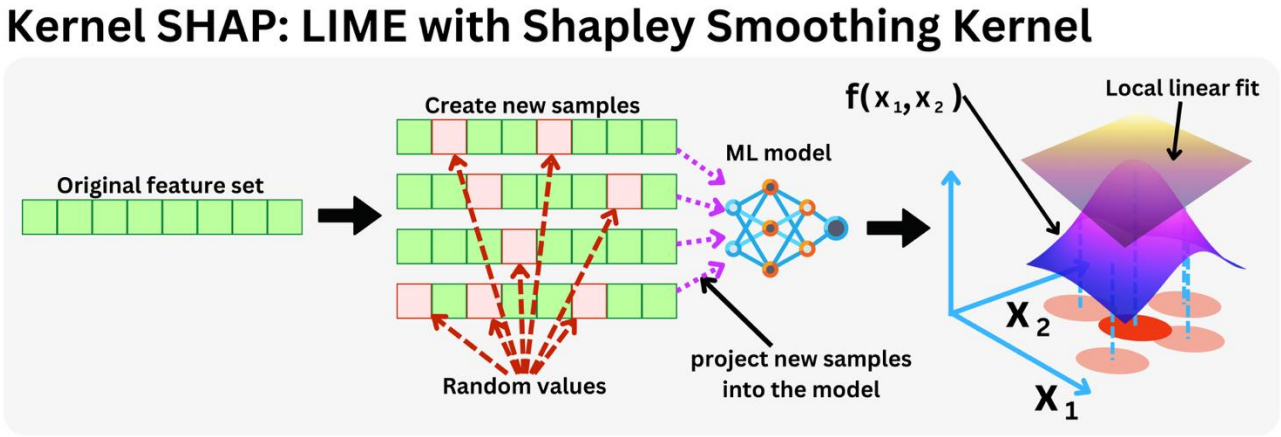
- Local accuracy
  - Same as efficiency property in Shapley, just using coalition vectors
- Missingness
  - If a feature is missing, it has 0 attribution
- Consistency
  - If a model changes so that the marginal contribution of a feature value increases or stays the same, the SHAP value also will increase or stay the same

SHAP Estimation

- Many ways to estimate SHAP values. 3 examples:

1. KernelSHAP
   - Randomly subsample feature set, then estimate a linear model with the coalition vector of presence/absence as features. Coefficients are then Shapley values

- Sample coalition vectors $\mathbf{z}'_k \in \{0,1\}^M, \quad k \in \{1,\ldots,K\}$ (1 = feature present in coalition, 0 = feature absent).
- Get prediction for each $\mathbf{z}'_k$ by first converting $\mathbf{z}'_k$ to the original feature space and then applying model $\hat{f}$: $\hat{f}(h_{\mathbf{x}}(\mathbf{z}'_k))$.
- Compute the weight for each coalition $\mathbf{z}'_k$ with the SHAP kernel.
- Fit weighted linear model.
- Return Shapley values ϕ_k, the coefficients from the linear model.

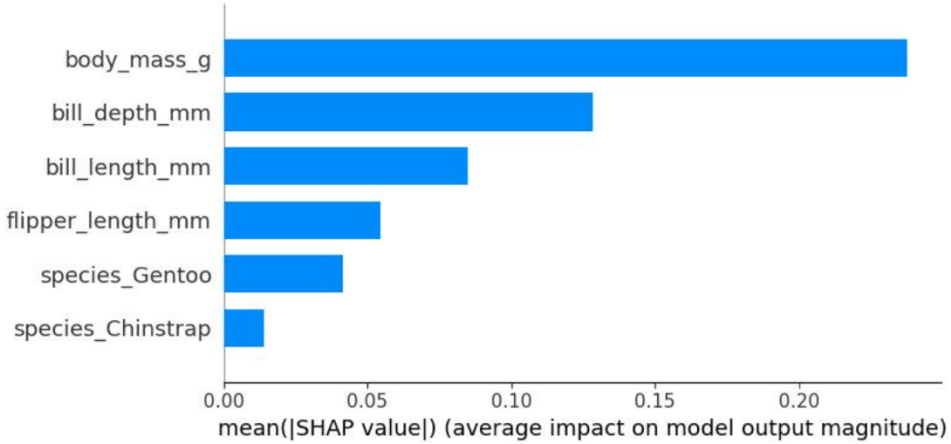


SHAP Estimation

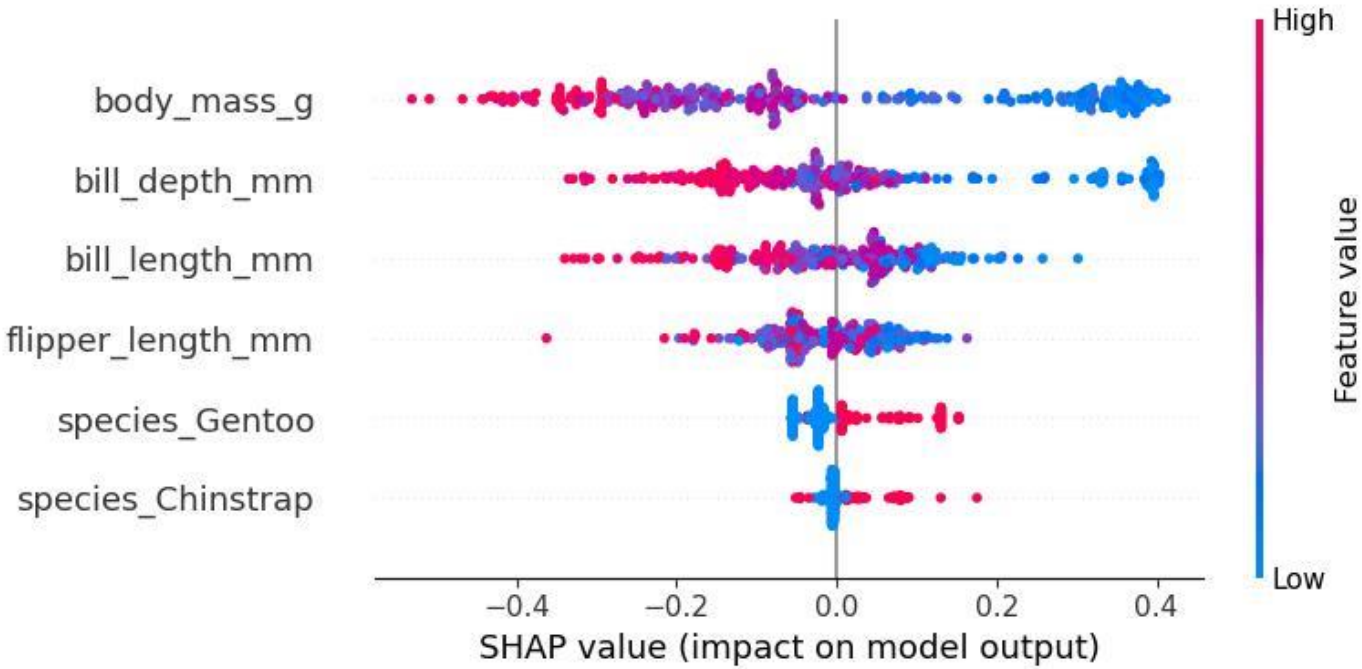
- Many ways to estimate SHAP values. 3 examples:

1. KernelSHAP
2. TreeSHAP (Lundberg, Erion, Lee 2019)
   - Utilizes existing feature subsets from decision trees
   - No longer model-agnostic
3. Permutation method
   - Instead of computing every possible Shapley value, we sample from permutations and take an average while maintaining the efficiency property

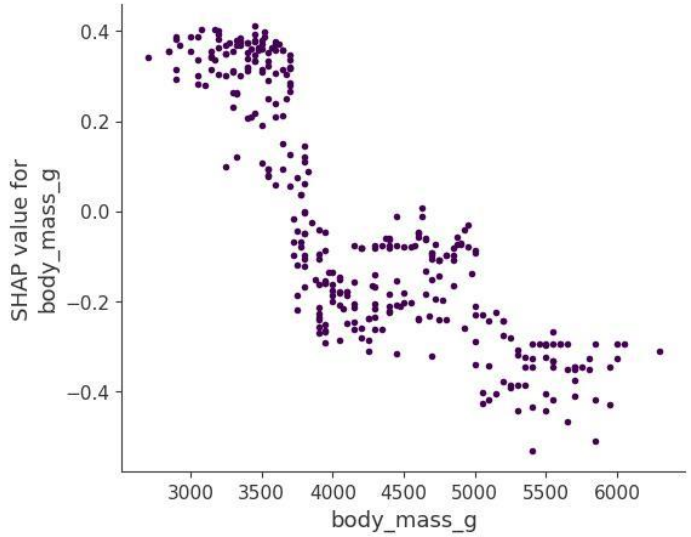
SHAP visualizations



Magnitude of feature attributions.



Feature importance with feature effects. Each point is a value for a feature and an observation.



For a particular feature, look at all Shapley values – there is feature-dependent relation with model output.

SHAP interactions

$$\phi_{i,j} = \sum_{S \subseteq \setminus \{i,j\}} \frac{|S|!(M - |S| - 2)!}{2(M-1)!} \delta_{ij}(S)$$

when $i \neq j$ and $\delta_{ij}(S) = \hat{f}_{\mathbf{x}}(S \cup \{i,j\}) - \hat{f}_{\mathbf{x}}(S \cup \{i\}) - \hat{f}_{\mathbf{x}}(S \cup \{j\}) + \hat{f}_{\mathbf{x}}(S)$.

- Additional combined feature effect after accounting for individual feature effects
- Subtracts off main effects to get pure interaction effect, averaged over all possible feature sets S

SHAP strengths and weaknesses

- Strong theoretical foundation in game theory
- Contrastive explanations that compare the prediction with the average prediction
- Slow
- Doesn’t handle correlated features well – generally assumes independence

Local Interpretable Model-Agnostic Explanations (LIME)

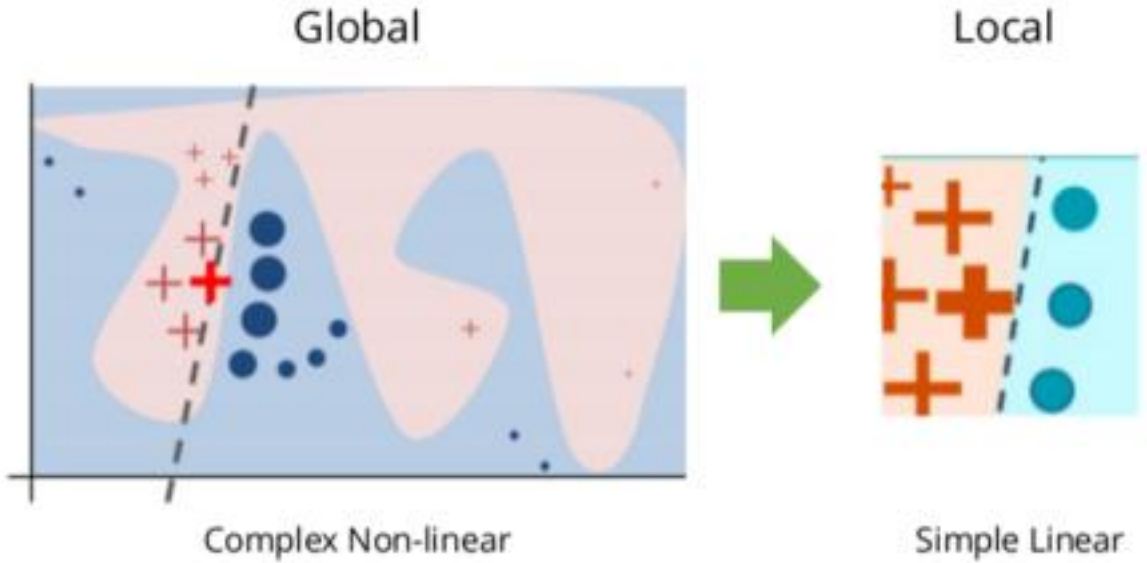
"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

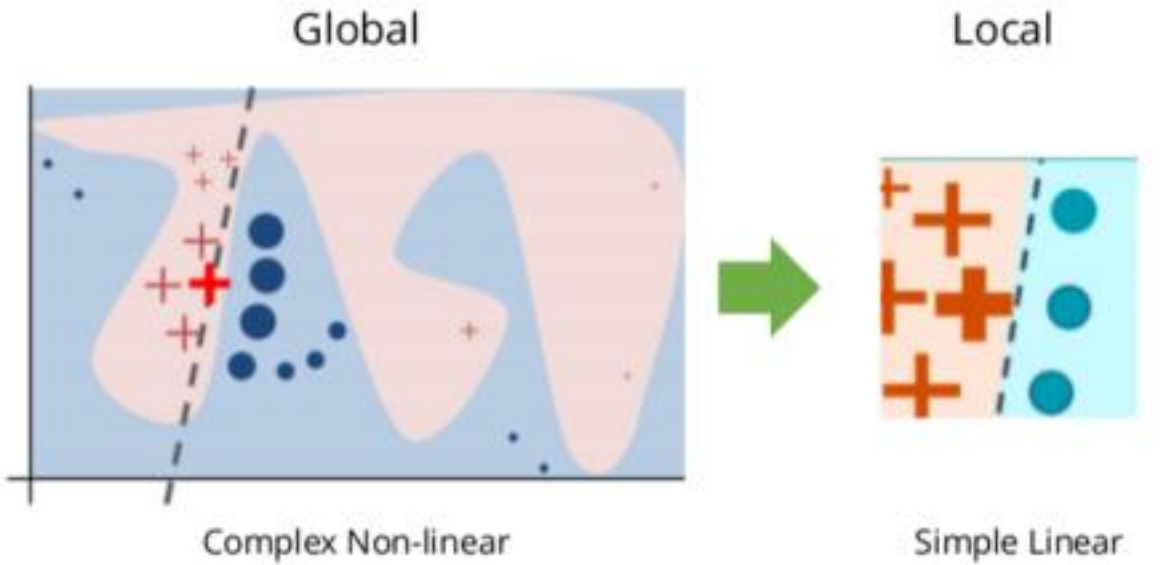
Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

- Key idea: Fit a local linear model around a **particular** observation of interes



Local Interpretable Model-Agnostic Explanations (LIME)

- Key idea: Fit a local linear model around a **particular** observation of interest
  1. Sample points around x
  2. Use a model to predict labels for each sample
  3. Weight samples according to distance to x
  4. Learn simple linear model on weighted samples
  5. Explain using simple linear model



LIME Formulation

$$\text{explanation}(\mathbf{x}) = \arg\min_{g \in G} L(\hat{f}, g, \pi_{\mathbf{x}}) + \Omega(g)$$

Model complexity

Explanation for observation x

Class of interpretable functions G (linear models)

Loss function – MSE for regression

Pi defines neighborhood around observation x

LIME Formulation

$$\text{explanation}(\mathbf{x}) = \arg\min_{g \in G} L(\hat{f}, g, \pi_{\mathbf{x}}) + \Omega(g)$$

Explanation for observation x

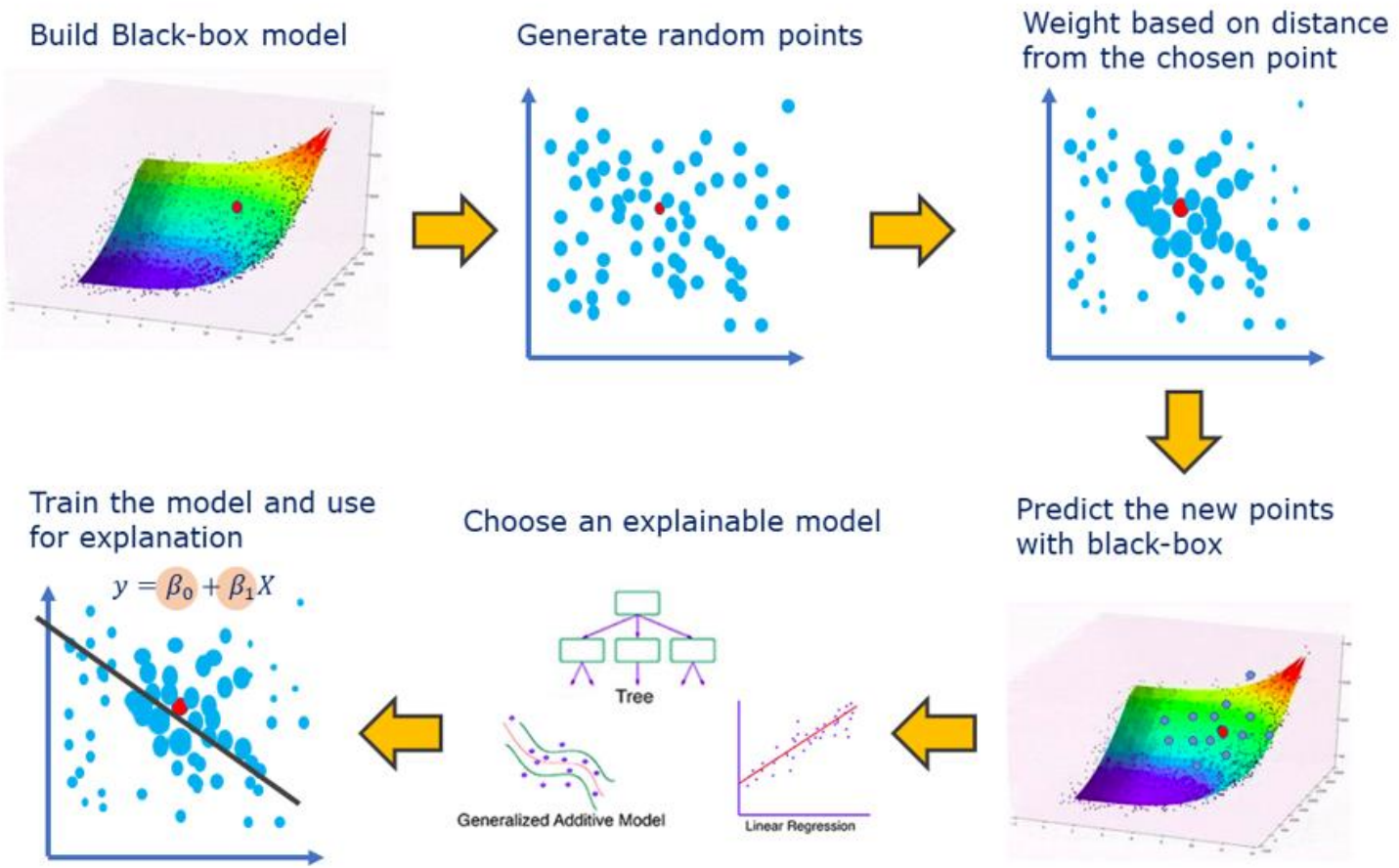
Class of interpretable functions G (linear models)

Loss function – MSE for regression

Pi defines neighborhood around observation x

Model complexity – can add regularization for additional sparsity

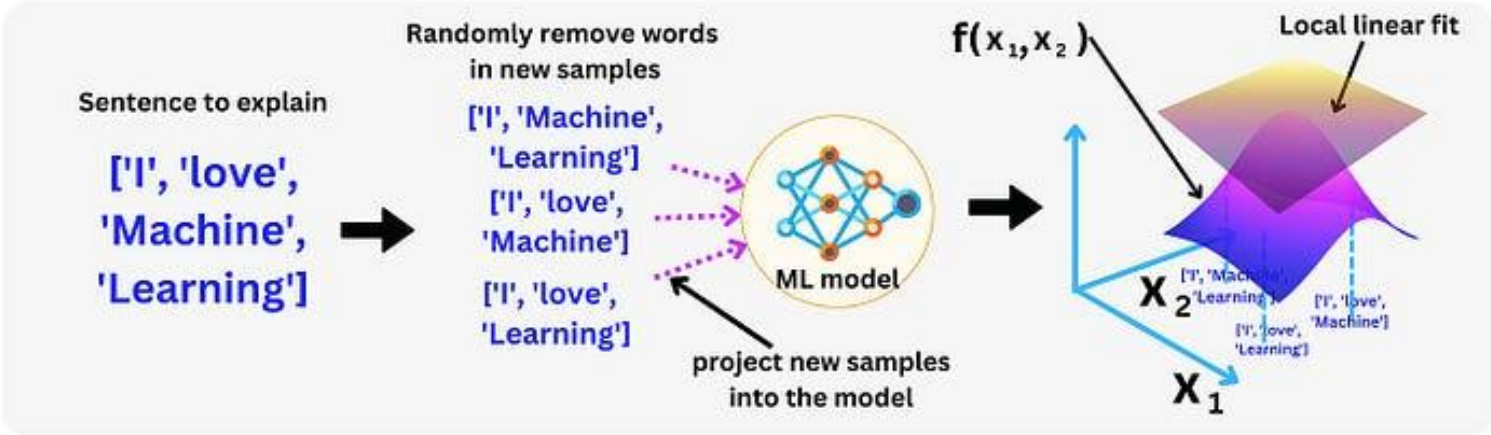
LIME sampling



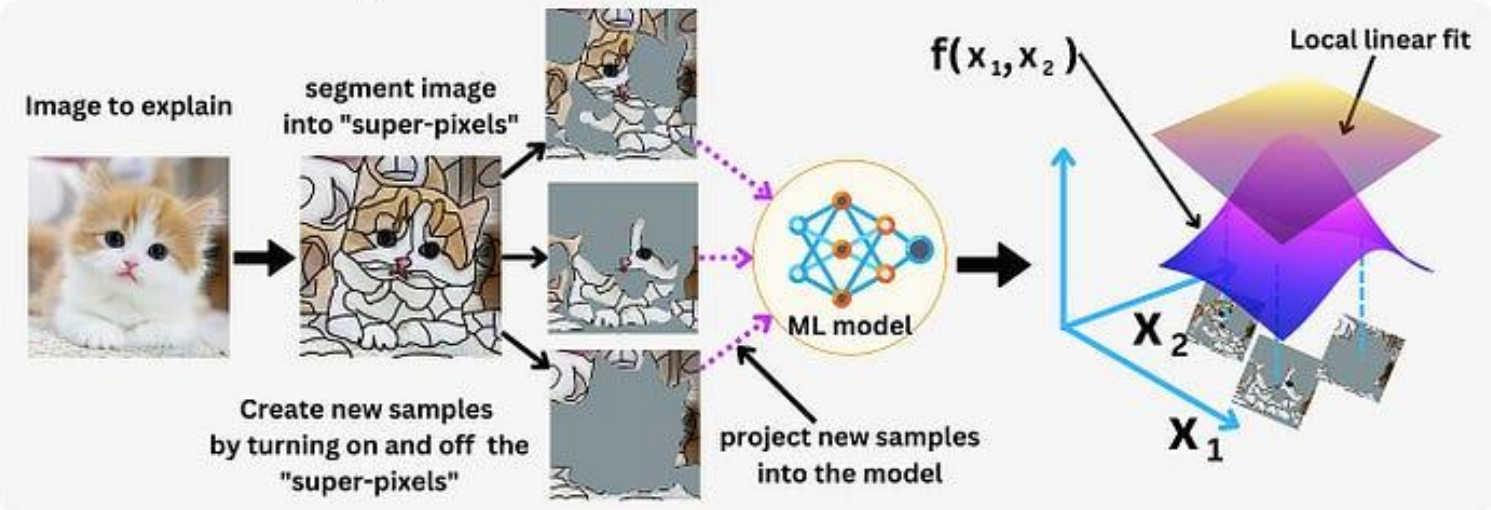
Medium.com, Giorgio Visani

LIME in non-tabular data

LIME with Text data



LIME with Image data



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$)**, "Acoustic guitar"** ($p = 0.24$) **and "Labrador"** ($p = 0.21$)

Medium.com Shree Raj

LIME example in DR

- DR methods can be very susceptible to noisy features

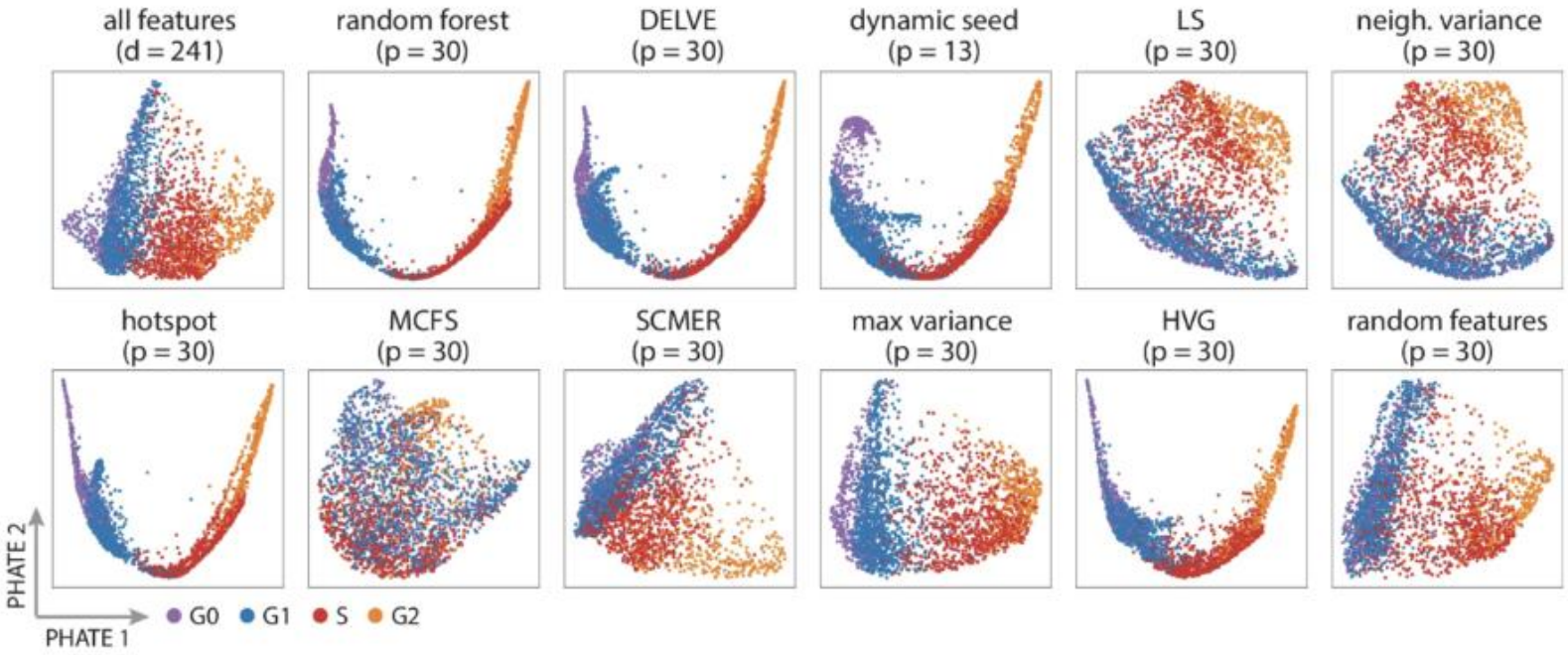
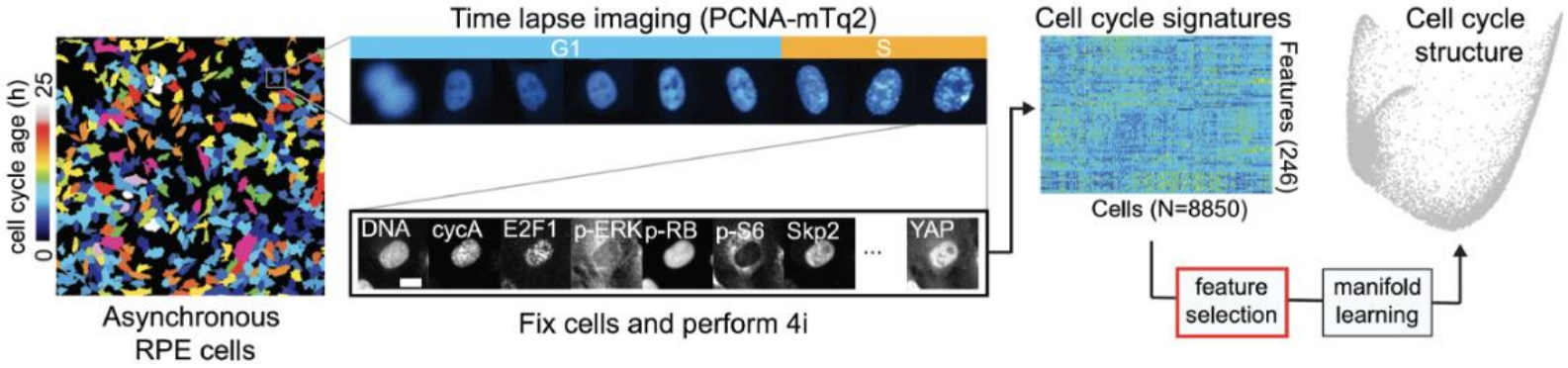
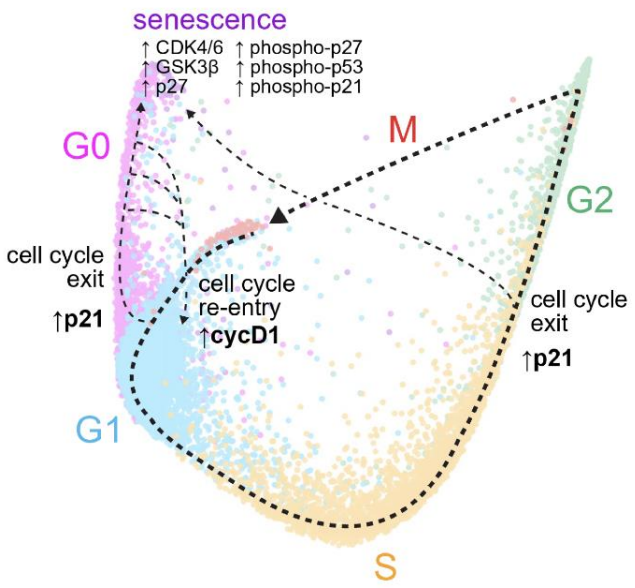


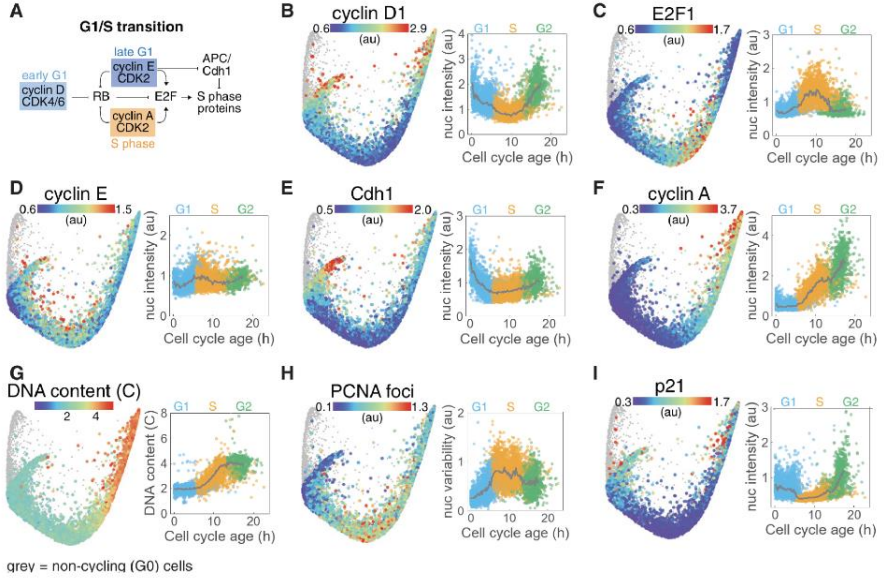
Figure: [Ranek et al., 2024]

- But what if you don't have access to the features selected originally?
- What if you want to project new data?

LIME example in DR







[Stallaert et al., 2022]

LIME(ADE)

LIMEADE: Local Interpretable Manifold Explanations for Dimension Evaluations

Tarek M. Zikry & Genevera I. Allen

Input data $X_{n \times p}$, reduced dimension data $Y_{n \times r}$, where $r \ll p$

Global

$$\hat{V} = \arg\min_{V} \left(\frac{1}{2} ||Y - XV||_2^2 + \lambda \sum_{j=1}^{p} ||V_j||_2 \right)$$

where $j = 1, \ldots, J$ are individual features, grouping across dimensions of the reduced embedding.

Unlike standard Lasso, which applies an ℓ_1 penalty to individual coefficients and promotes element-wise sparsity, the Group Lasso[1] extends this framework by penalizing groups of coefficients together, which allows entire features to be either fully included or excluded across all dimensions of r.

[1] [Yuan and Lin, 2006]

- 80/20 split into $[(X_{train}, Y_{train}), (X_{test}, Y_{test})]$
- $\hat{V} = LIMEADE(X_{train}, Y_{train})$
- $\hat{Y}_{test} = X_{test} \hat{V}$

Three levels of interpretation

Global

$$\hat{V} = \arg\min_{V} \left(\frac{1}{2} ||Y - XV||_2^2 + \lambda \sum_{j=1}^{p} ||V_j||_2 \right)$$

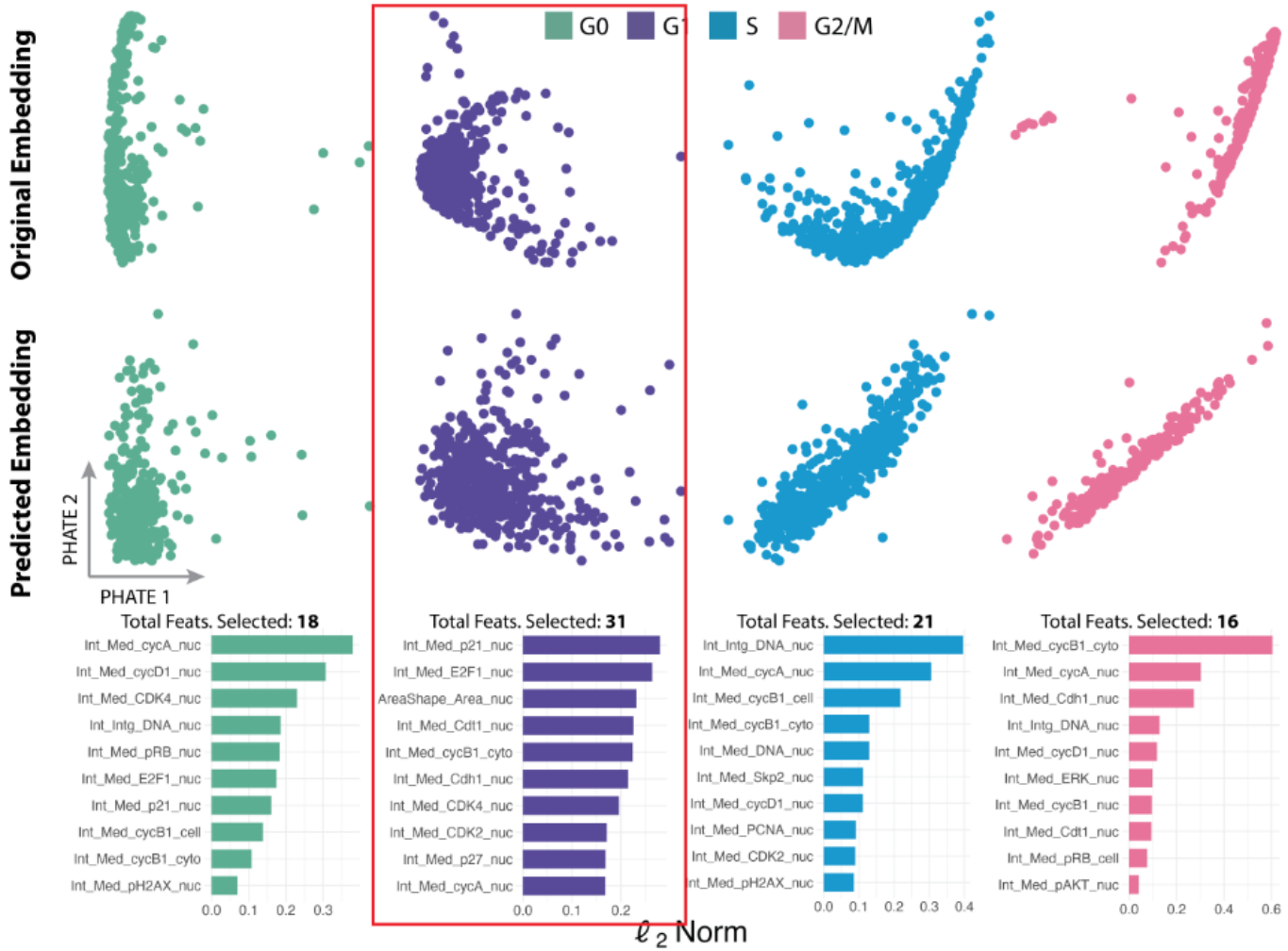
Partitioned

$$\hat{V}_{M_k} = \arg\min_{V} \left(\frac{1}{2} ||Y_{M_k} - X_{M_k} V_k||_2^2 + \lambda \sum_{j=1}^{p} ||V_j||_2 \right)$$

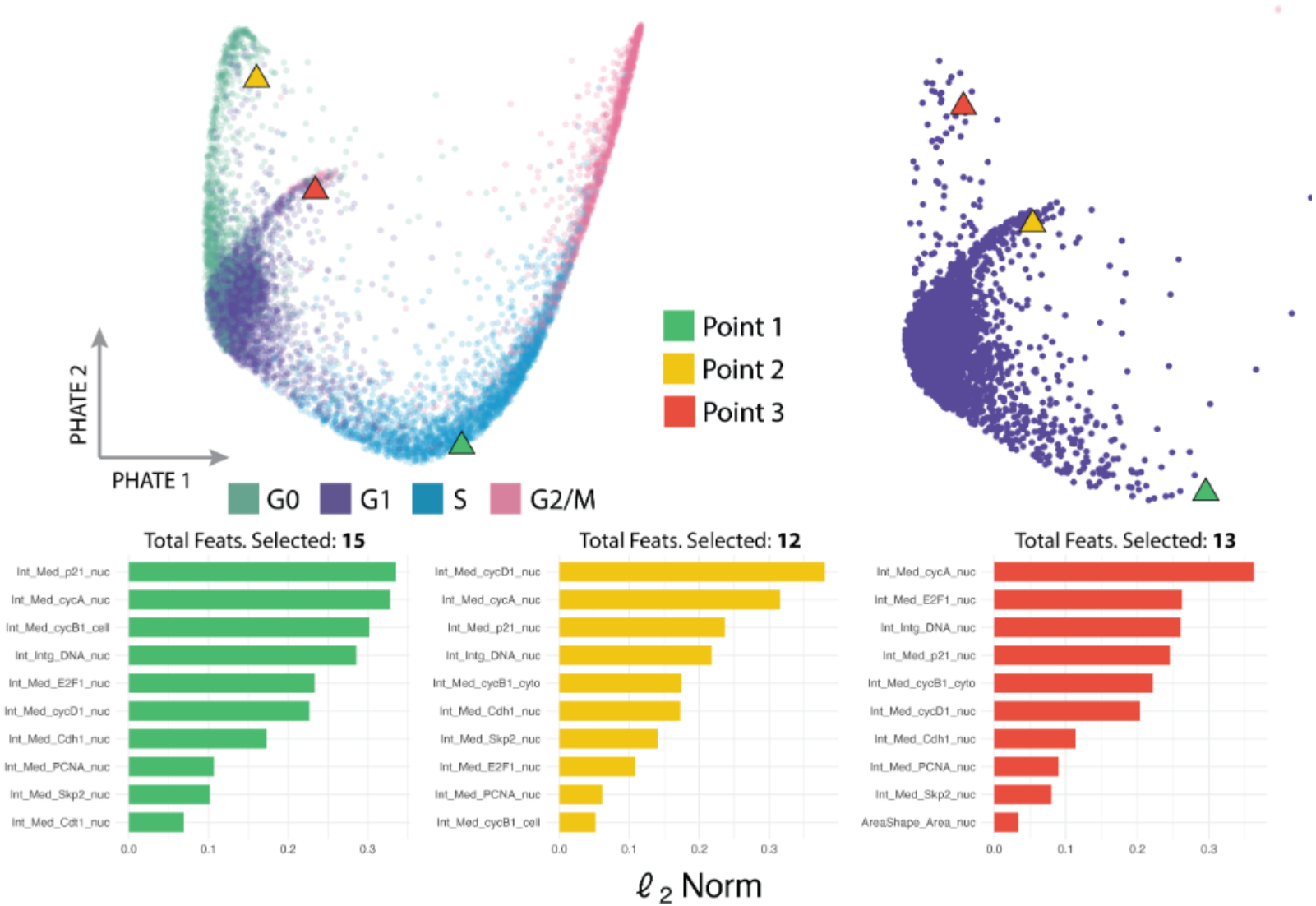
Local

$$\widehat{V} = \arg\min_{V} \left(\frac{1}{2} ||W^{1/2}(Y - XV)||_2^2 + \lambda \sum_{j=1}^{p} ||V_j||_2 \right)$$

Partitioned explanations



Local explanations



LIME strengths and weaknesses

- Have a measure of how well the interpretable model approximates the black box prediction
- Can use interpretable features even if original model was trained on latent embedding
- Sampling can be unrepresentative of points – many methods around this
- Complexity (sparsity penalties) have to be predefined
- Choice of neighborhood is a hard hyperparameter to tune, especially in higher dimensions

SHAP vs. LIME

Metrics	SHAP	LIME
Concept	Applies to the model as-is	Fits a local surrogate model to explain the complex model
Theory	Additive feature attribution based on game theory	Feature perturbation method
Type	Post-hoc model-agnostic	
Data type	Images, tabular data, and signals	
Explanation	Global, local	Local
Collinearity consideration	Not in the original method	No
Nonlinear decision	Depends on the used model	Incapable
Computing time	Higher	Lower
Visualization	Waterfall, beeswarm, and summary plots	One single plot

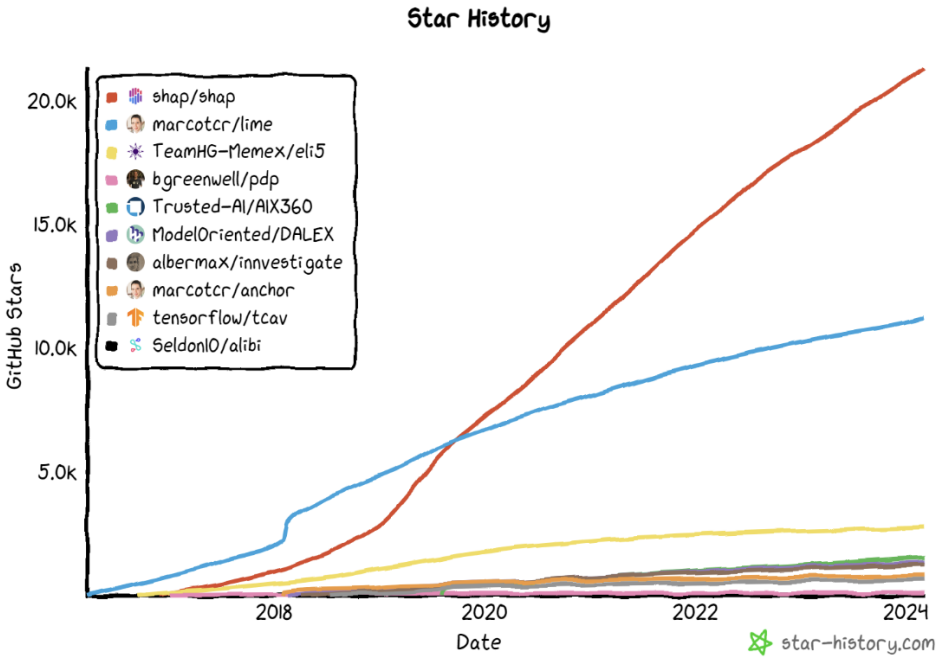


Figure 1: GitHub Star for 10 common XAI methods.

Salih, Ahmed M., et al. "A perspective on explainable artificial intelligence methods: SHAP and LIME." *Advanced Intelligent Systems* 7.1 (2025): 2400304.

Revisiting COMPAS

- In Rudin et al. (2020) the authors investigate ProPublica's claim that COMPAS is using race
  - ProPublica: Using subset of data (Boward County, Florida), they created a linear model predicting recidivism with race as a covariate, and the linear model found a significant p-value from the respective coefficient
    - Claim: race is being used to predict recidivism and this is what led to racial biases

Revisiting COMPAS

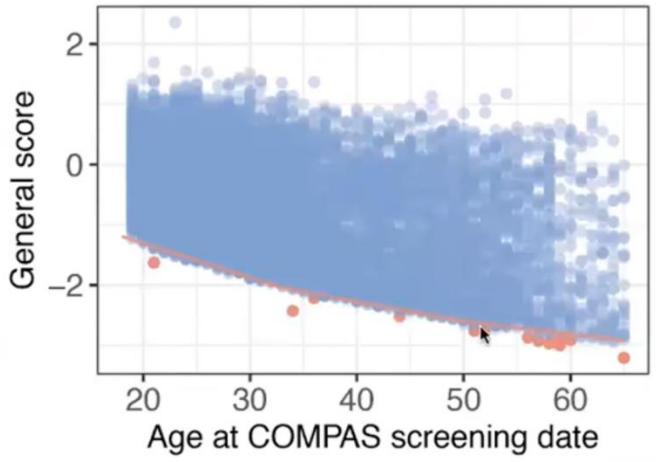
- In Rudin et al. (2020) the authors investigate ProPublica's claim that COMPAS is using race
  - COMPAS is probably not linearly dependent on race/age

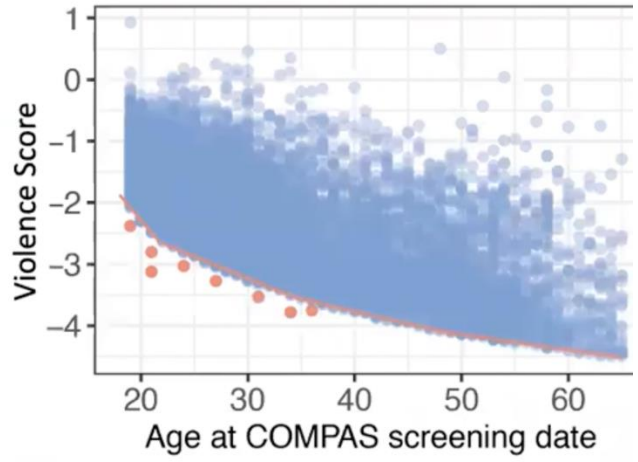
Conjecture: *The COMPAS general recidivism model is a nonlinear additive model. Its dependence on age in Broward County is approximately a linear spline, defined as follows:*

$$\text{for ages } \leq 33.26,\ f_{\text{age}}(\text{age}) = -0.056 \times \text{age} - 0.179$$
$$\text{for ages between } 33.26 \text{ and } 50.02,\ f_{\text{age}}(\text{age}) = -0.032 \times \text{age} - 0.963$$
$$\text{for ages} \geq 50.02,\ f_{\text{age}}(\text{age}) = -0.021 \times \text{age} - 1.541.$$

Similarly, the COMPAS violence recidivism model is a nonlinear additive model, with a dependence on age that is approximately a linear spline, defined by:

$$\text{for ages } \leq 21.77,\ f_{\text{viol age}}(\text{age}) = -0.205 \times \text{age} + 1.815$$
$$\text{for ages between } 21.77 \text{ and } 34.58,\ f_{\text{viol age}}(\text{age}) = -0.070 \times \text{age} - 1.113$$
$$\text{for ages between } 34.58 \text{ and } 48.36,\ f_{\text{viol age}}(\text{age}) = -0.040 \times \text{age} - 2.166$$
$$\text{for ages} \geq 48.36,\ f_{\text{viol age}}(\text{age}) = -0.025 \times \text{age} - 2.882.$$



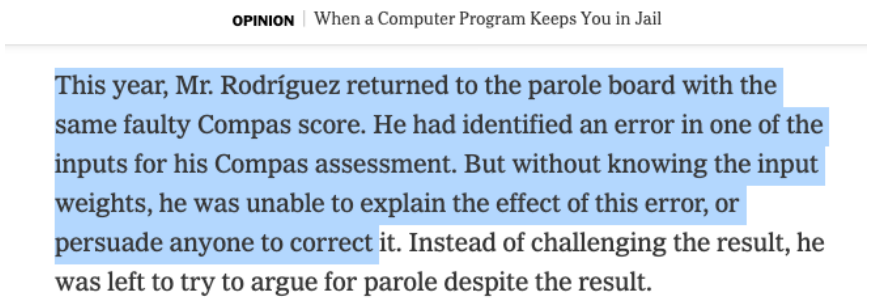


Revisiting COMPAS

- In Rudin et al. (2020) the authors investigate ProPublica's claim that COMPAS is using race
  - COMPAS is probably not linearly dependent on race/age
- When the authors marginalize out the effect of nonlinear age dynamics, and then compare machine learnings with and without age race, COMPAS predictions are the same
  - Unlikely that COMPAS is using race as a predictor

Revisiting COMPAS

- So what caused large differences highlighted by ProPublica as racial bias?
- Typos…
  - 137 survey factors as features, each hand entered
  - Some examples highlighted in ProPublica analysis are actually typos in the original data leading to bias in the algorithmic prediction
- Large issue in this debate is that COMPAS is a black box model, not in the algorithmic sense, but because it is proprietary

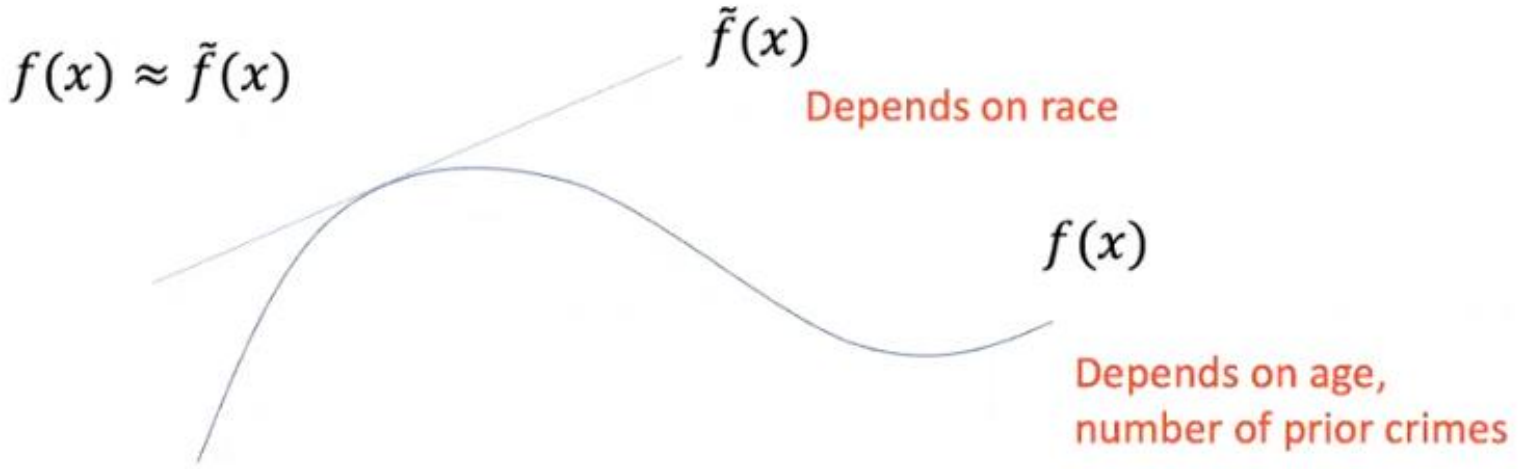


OPINION | When a Computer Program Keeps You in Jail

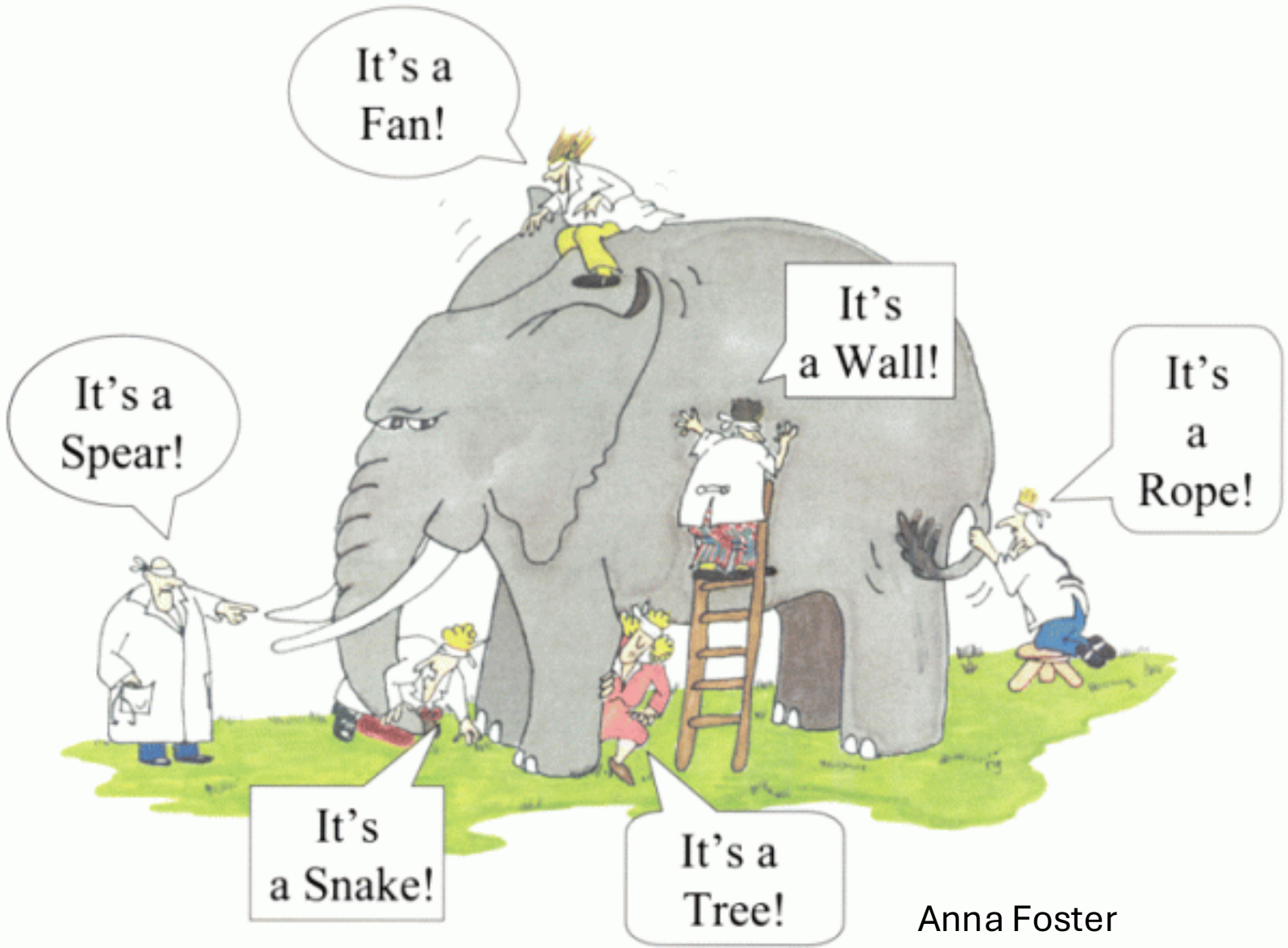
This year, Mr. Rodríguez returned to the parole board with the same faulty Compas score. He had identified an error in one of the inputs for his Compas assessment. But without knowing the input weights, he was unable to explain the effect of this error, or persuade anyone to correct it. Instead of challenging the result, he was left to try to argue for parole despite the result.

Interpretability vs. explainability

- Interpretable ML: When you use a model that is not a black box
- Explainable ML: When you use a black box and then a post-hoc interpretation method afterwards
- Approximations of explanations are not explanations
- *When different models are applied, to the same task using the same data, the top features identified may differ between ML models*



Rashomon Effect



Anna Foster

Interpretability vs. explainability

Perspective | Published: 13 May 2019

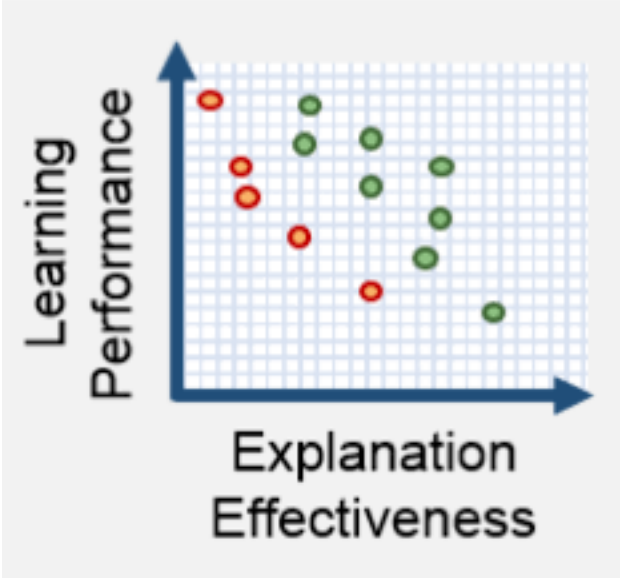
Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

Cynthia Rudin

Nature Machine Intelligence **1**, 206–215 (2019) | Cite this article

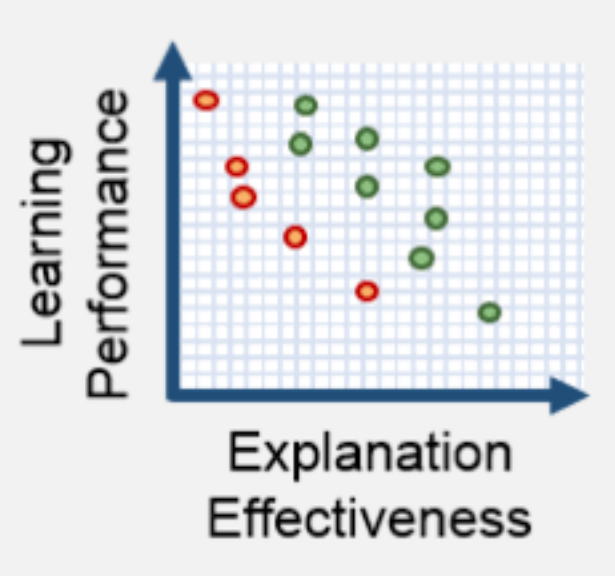
- The tradeoff between interpretability and accuracy does not exist
- Explainability models are unreliable and lead to double the bias
- Interpretable models, particularly in high-stakes decisions, can perform just as well
- Example: here seems to be no clear benefit of black-box models over inherently interpretable models in terms of prediction accuracy on the criminal recidivism problem (Zeng et al., 2017; Tollenaar and van der Heijden, 2013, Angelino et al. 2018).

Interpretability ”tradeoff”



DARPA BAA

Interpretability ”tradeoff”



DARPA BAA

Question

Q1
If we sample a different training set, are the interpretations similar?

Q2
Do two IML methods generate similar interpretations on the same data?

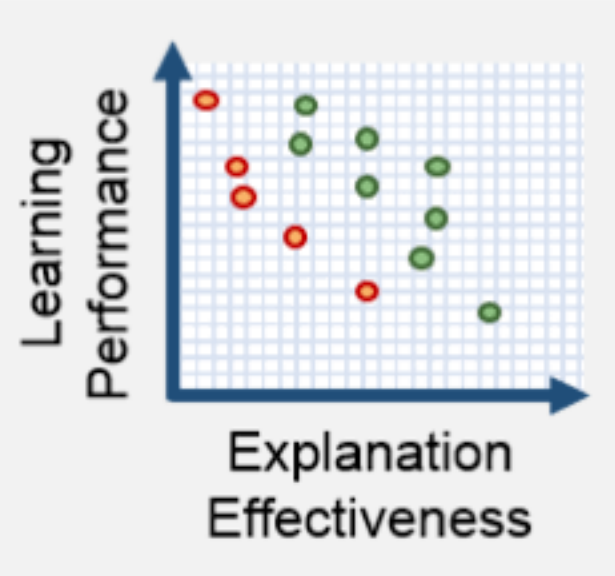
Q3
Does higher accuracy lead to more consistent interpretations?

Are machine learning interpretations reliable?

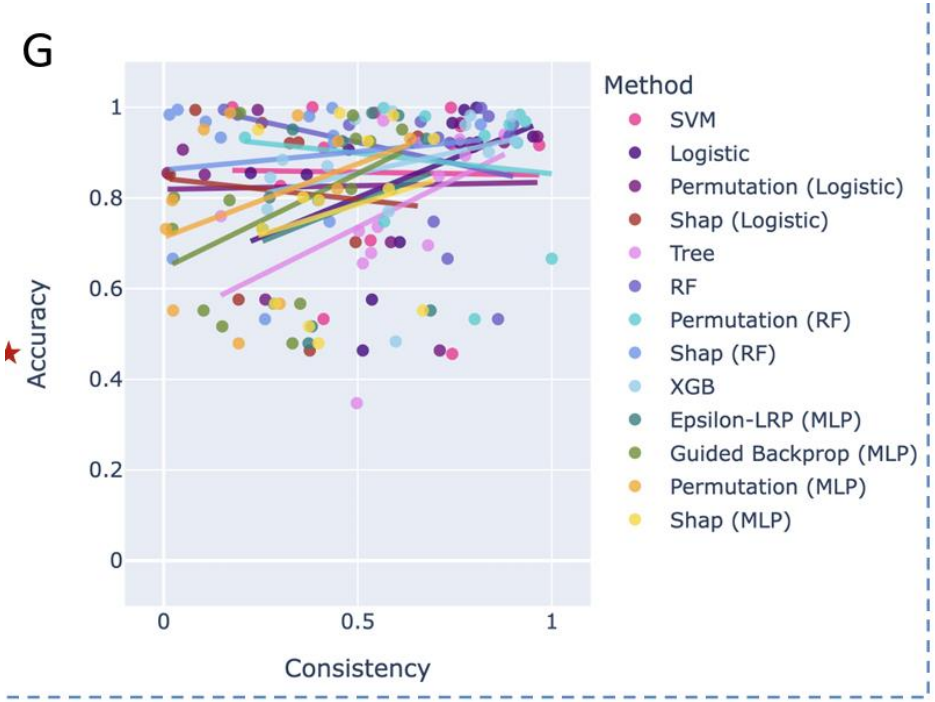
A stability study on global interpretations

Luqin Gan[1], Tarek M. Zikry[2,3], Genevera I. Allen[2,3,4]

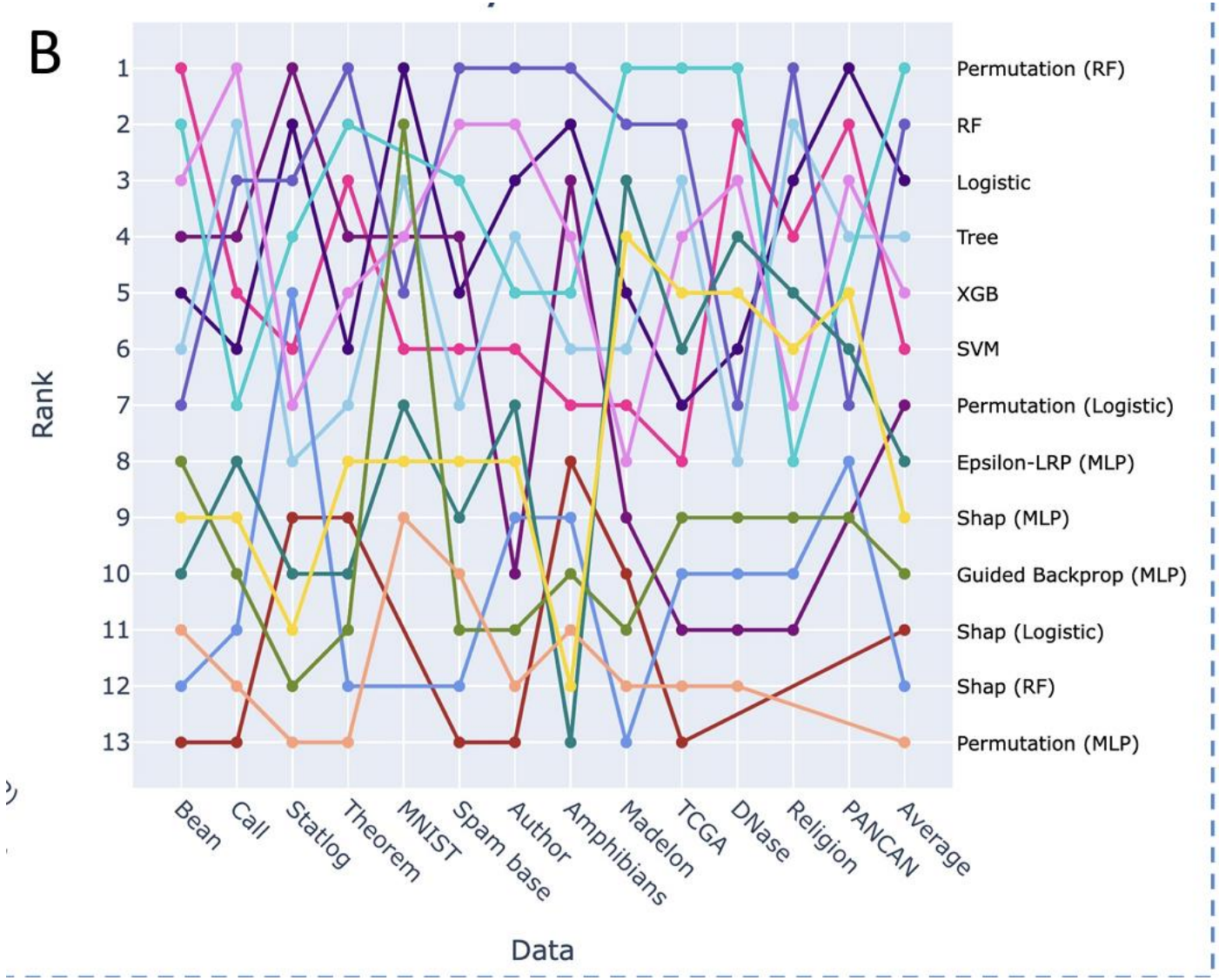
Interpretability ”tradeoff”



DARPA BAA



Interpretability ”tradeoff”



Question

Q1

If we sample a different training set, are the interpretations similar?

Q2

Do two IML methods generate similar interpretations on the same data?

Q3

Does higher accuracy lead to more consistent interpretations?

Other methods to consider

- Permutation importance
- Leave one covariate out (LOCO)
- Model-specific methods
  - Saliency maps for NNs
- Feature interaction methods

Input: Trained model $\hat{f}$, training data $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$, test data $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$, and error measure L.

Procedure:

1. **Measure the original model error:**

$$e_{\text{orig}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} L\big(y_{\text{test}}^{(i)}, \hat{f}(\mathbf{x}_{\text{test}}^{(i)})\big)$$

2. **For each feature** $j \in \{1, \ldots, p\}$**:**
   - Remove feature j from the dataset, creating new datasets $\mathbf{X}_{\text{train},-j}$ and $\mathbf{X}_{\text{test},-j}$.
   - Train a new model $\hat{f}_{-j}$ on $(\mathbf{X}_{\text{train},-j}, \mathbf{y}_{\text{train}})$.
   - Measure the new error on the modified test set:

$$e_{-j} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} L\big(y_{\text{test}}^{(i)}, \hat{f}_{-j}(\mathbf{x}_{\text{test},-j}^{(i)})\big)$$

3. **Calculate LOFO importance for each feature.**
   - As a quotient:

$$LOFO_j = \frac{e_{-j}}{e_{\text{orig}}}$$

   - Or as a difference:

$$LOFO_j = e_{-j} - e_{\text{orig}}$$

4. **Sort and visualize** the features by descending importance $LOFO_j$.

Unsupervised vs. supervised

- Focus on interpretability/explainability for supervised methods
- Most supervised model-agnostic interpretations are post-hoc
- Less attention towards unsupervised learning methods but growing

Unsupervised vs. supervised

- Focus on interpretability/explainability for supervised methods
- Most supervised model-agnostic interpretations are post-hoc
- Less attention towards unsupervised learning methods but growing

Further reading and references

- Allen, Genevera I., Luqin Gan, and Lili Zheng. "Interpretable machine learning for discovery: Statistical challenges and opportunities." *Annual Review of Statistics and Its Application* 11 (2023).
- Rudin, Cynthia. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." *Nature machine intelligence* 1.5 (2019): 206-215.
- Gan, Luqin, Tarek M. Zikry, and Genevera I. Allen. "Are machine learning interpretations reliable? A stability study on global interpretations." *arXiv preprint arXiv:2505.15728* (2025).
- Molnar, Christoph. *Interpretable machine learning*. Lulu. com, 2020.