

Presentation at L3S Research Seminar

DALE: Differential Accumulated Local Effects for efficient and accurate global explanations

Vasilis Gkolemis^{1,2} Theodore Dalamagas¹ Christos Diou²

¹ATHENA Research and Innovation Center

²Harokopio University of Athens

January 2023

Who we are

- **Vasilis Gkolemis:**
 - ▶ Research Assistant at ATHENA Research Center ([ATHENA RC](#))
 - ▶ First-year PhD at Harokopio University of Athens ([HUA](#))
 - ▶ Main focus: Explainability under uncertainty
- Supervisors:
 - ▶ [Christos Diou](#) (HUA) → Generalization, Few(Zero)-shot learning
 - ▶ [Eirini Ntoutsi](#) (UniBw-M) → Explainability, Fairness
 - ▶ [Theodore Dalamagas](#) (ATHENA) → Databases, data semantics
- Paper I will present
 - ▶ [DALE: Differential Accumulated Local Effects for efficient and accurate global explanations](#)
 - ▶ Accepted at [Asian Conference Machine Learning \(ACML\) 2022](#)

eXplainable AI (XAI)

- Black-box model $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$, trained on \mathcal{D}
- XAI extracts interpretable properties:
 - Tabular data - Which features favor a prediction?
 - Computer Vision - Which image areas confuse the model?
 - NLP - Which words classified the comment as offensive?
- Categories:
 - Global vs local
 - Model-agnostic vs Model-specific
 - Output? number, plot, instance etc.

Feature Effect: global, model-agnostic, outputs plot

Feature Effect

$y = f(x_s) \rightarrow$ plot showing the effect of x_s on the output y

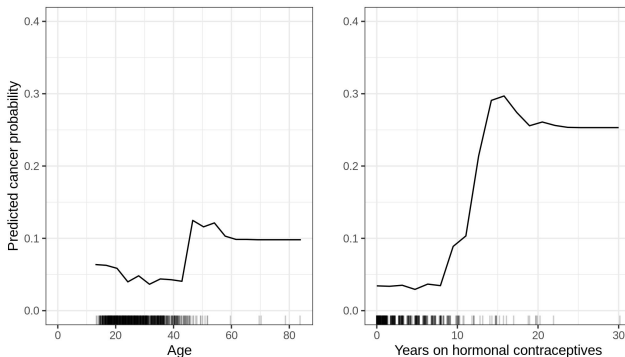


Figure: Image taken from Interpretable ML book (Molnar, 2022)

Feature Effect is simple and intuitive.

Feature Effect Methods

- $x_s \rightarrow$ feature of interest, $\mathbf{x}_c \rightarrow$ other features
- Isolating the effect of x_s is a difficult task:
 - ▶ features are correlated
 - ▶ f has learned complex interactions
- Three well-known methods:
 - ▶ Partial Dependence Plots (PDP)
 - ▶ M-Plots
 - ▶ Accumulated Local Effects (ALE)

Partial Dependence Plots (PDP)

- Proposed by J. Friedman on 2001¹ and is the marginal **effect** of a feature to the model output:

$$f_s(x_s) = \mathbb{E}_{\mathbf{x}_c} [f(x_s, \mathbf{x}_c)] = \int f(x_s, \mathbf{x}_c) p(\mathbf{x}_c) d\mathbf{x}_c$$

where:

- x_s is the feature whose effect we wish to compute
 - \mathbf{x}_c are the rest of the features
- Approximation:

$$\hat{f}_s(x_s) = \frac{1}{n} \sum_{i=1}^n f(x_s, \mathbf{x}_c^{(i)})$$

¹J. Friedman. "Greedy function approximation: A gradient boosting machine." Annals of statistics (2001): 1189-1232

Issues with PDPs

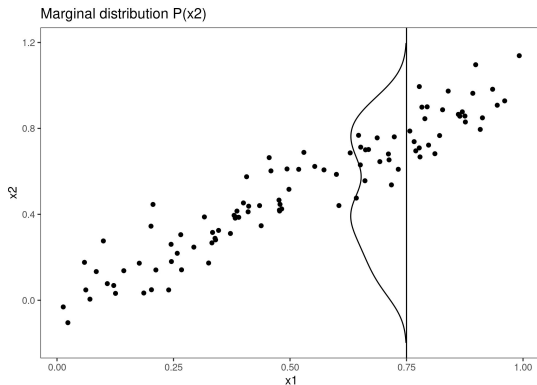


Figure: C. Molnar, IML book

- Correlated features

- ▶ To compute the effect of $x_{\text{age}} = 20$ on the output (cancer probability) it will integrate over all $x_{\text{years_contraceptives}}$ values, e.g., $[0, 50]$
- ▶ f can have weird behavior when $x_{\text{age}} = 20, x_{\text{years_contraceptives}} = 20$ (out of distribution)
- ▶ As a result, we have a wrong estimation of the feature effect

- We use the value of x_s as a condition, so we integrate over $\mathbf{x}_c | x_s$

$$f(x_s) = \mathbb{E}_{\mathbf{x}_c | x_s} [f(x_s, \mathbf{x}_c)] = \int f(x_s, \mathbf{x}_c) p(\mathbf{x}_c | x_s) d\mathbf{x}_c$$

where:

- ▶ x_s is the feature whose effect we wish to compute
- ▶ \mathbf{x}_c the rest of the features
- Approximation:

$$f_s(x_s) = \frac{1}{n} \sum_{i: x_s^{(i)} \approx x_s} f(x_s, \mathbf{x}_c^{(i)})$$

In the previous example

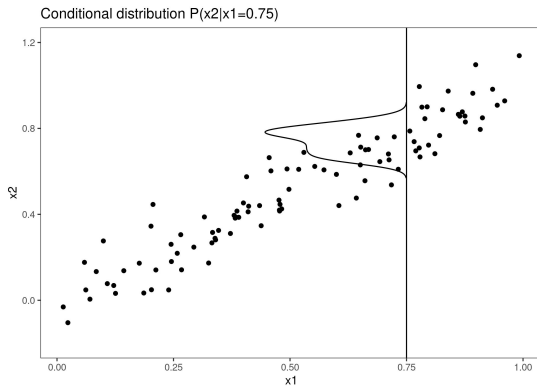


Figure: C. Molnar, IML book

Issues with M-Plots

- Aggregated effect symptom \rightarrow the calculated effects result from the combination of all (correlated) features
- Real effect:
 - ▶ $x_{\text{age}} = 50 \rightarrow 10$
 - ▶ $x_{\text{years_contraceptives}} = 20 \rightarrow 10$
 - ▶ aggregated effect close to 20
- Because $x_{\text{age}}, x_{\text{years_contraceptives}}$ are correlated, MPlot may assign:
 - ▶ $x_{\text{age}} = 50 \rightarrow 17 \approx$ aggregated effect
 - ▶ $x_{\text{years_contraceptives}} = 20 \rightarrow 17 \approx$ aggregated effect

Accumulated Local Effects (ALE)²

- Resolves problems that result from the feature correlation by computing differences over a (small) window

$$f(x_s) = \int_{x_{min}}^{x_s} \underbrace{\mathbb{E}_{\mathbf{x}_c|z}}_{\text{realistic}} \underbrace{\left[\frac{\partial f}{\partial x_s}(z, \mathbf{x}_c) \right]}_{\text{isolates}} \partial z$$

²D. Apley and J. Zhu. “Visualizing the effects of predictor variables in black box supervised learning models.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82.4 (2020): 1059-1086.

ALE approximation

ALE definition: $f(x_s) = \int_{x_{s,min}}^{x_s} \mathbb{E}_{\mathbf{x}_c|z} \left[\frac{\partial f}{\partial x_s}(z, \mathbf{x}_c) \right] \partial z$

ALE approximation: $f(x_s) = \underbrace{\sum_k^{k_x} \frac{1}{|S_k|} \sum_{i: \mathbf{x}^i \in S_k} \underbrace{[f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)]}_{\text{point effect}}}_{\text{bin effect}}$

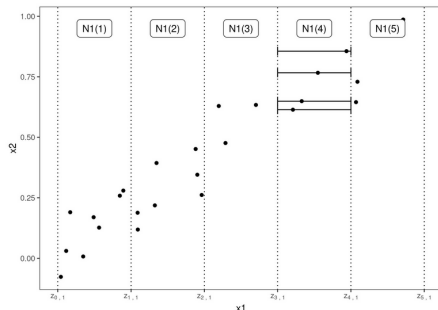


Figure: Image taken from Interpretable ML book (Molnar, 2022)

ALE approximation - weaknesses

$$f(x_s) = \sum_k^{k_x} \underbrace{\frac{1}{|S_k|} \sum_{i: \mathbf{x}^i \in S_k} \underbrace{[f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)]}_{\text{point effect}}}_{\text{bin effect}}$$

- Point Effect \Rightarrow evaluation **at bin limits**
 - ▶ 2 evaluations of f per point \rightarrow slow
 - ▶ change bin limits, pay again $2 * N$ evaluations of $f \rightarrow$ restrictive
 - ▶ broad bins may create out of distribution (OOD) samples \rightarrow not-robust in wide bins

ALE approximation has some weaknesses

Recap!

- PDP → problems with correlated features → Unrealistic instances
- MPlot → problems with correlated features → Aggregated effects
- ALE → resolves both issues! But:
- ALE approximation (estimation of ALE from the training set)
 - ▶ slow when there are many features
 - ▶ unrealistic instances when bins become bigger
- Differential ALE (DALE)!

Our proposal: Differential ALE

$$f(x_s) = \Delta x \sum_k^{k_x} \frac{1}{|S_k|} \sum_{i: x^i \in S_k} \underbrace{\left[\frac{\partial f}{\partial x_s}(x_s^i, x_c^i) \right]}_{\text{point effect}}$$

bin effect

- Point Effect \Rightarrow evaluation **on instances**
 - ▶ Fast \rightarrow use of auto-differentiation, all derivatives in a single pass
 - ▶ Versatile \rightarrow point effects computed once, change bins without cost
 - ▶ Secure \rightarrow does not create artificial instances

For **differentiable** models, DALE resolves ALE weaknesses

DALE is faster and versatile - theory

$$f(x_s) = \underbrace{\Delta x \sum_k \frac{1}{|\mathcal{S}_k|} \sum_{i: \mathbf{x}^i \in \mathcal{S}_k}}_{\text{bin effect}} \underbrace{\left[\frac{\partial f}{\partial x_s}(\mathbf{x}_s^i, \mathbf{x}_c^i) \right]}_{\text{point effect}}$$

- Faster
 - ▶ gradients wrt all features $\nabla_{\mathbf{x}} f(\mathbf{x}^i)$ in a single pass
 - ▶ auto-differentiation must be available (deep learning)
- Versatile
 - ▶ Change bin limits, with near zero computational cost

DALE is faster and allows redefining bin-limits

DALE is faster and versatile - Experiments

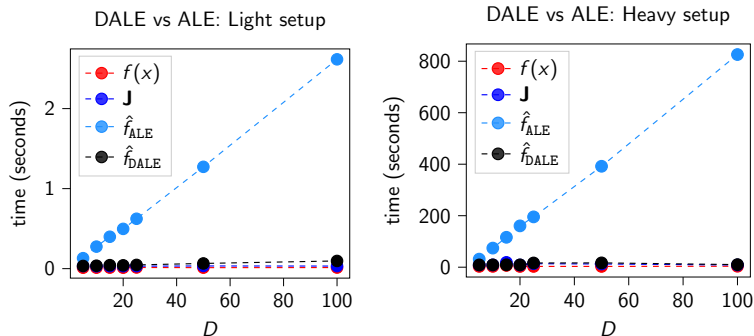


Figure: Light setup; small dataset ($N = 10^2$ instances), light f . Heavy setup; big dataset ($N = 10^5$ instances), heavy f

DALE considerably accelerates the estimation

DALE uses on-distribution samples - Theory

$$f(x_s) = \underbrace{\sum_k \frac{1}{|\mathcal{S}_k|} \sum_{i: \mathbf{x}^i \in \mathcal{S}_k} \underbrace{\left[\frac{\partial f}{\partial x_s}(\mathbf{x}_s^i, \mathbf{x}_c^i) \right]}_{\text{point effect}}}_{\text{bin effect}}$$

- point effect **independent** of bin limits
 - ▶ $\frac{\partial f}{\partial x_s}(\mathbf{x}_s^i, \mathbf{x}_c^i)$ computed on real instances $\mathbf{x}^i = (\mathbf{x}_s^i, \mathbf{x}_c^i)$
- bin limits affect only the **resolution** of the plot
 - ▶ wide bins \rightarrow low resolution plot, bin estimation from more points
 - ▶ narrow bins \rightarrow high resolution plot, bin estimation from less points

DALE enables wide bins without creating out of distribution instances

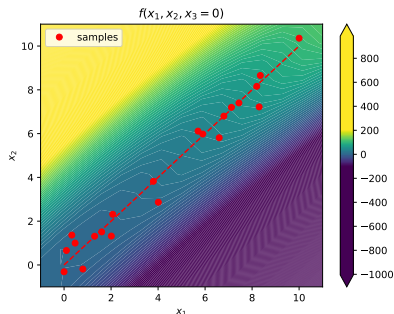
DALE uses on-distribution samples - Experiments

$$f(x_1, x_2, x_3) = x_1 x_2 + x_1 x_3 \pm g(x)$$

$$x_1 \in [0, 10], x_2 \sim x_1 + \epsilon, x_3 \sim \mathcal{N}(0, \sigma^2)$$

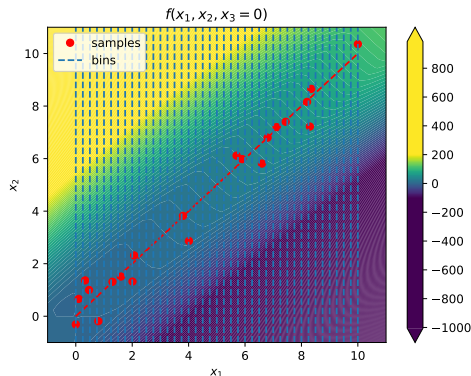
$$f_{\text{ALE}}(x_1) = \frac{x_1^2}{2}$$

- point effects affected by $(x_1 x_3)$
(σ is large)
- bin estimation is noisy (samples are few)



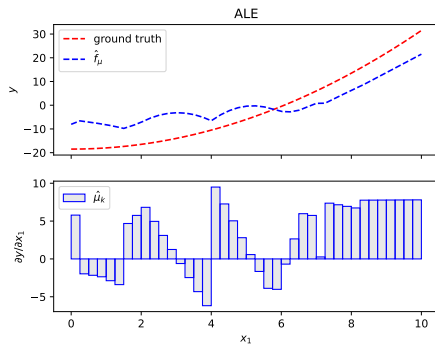
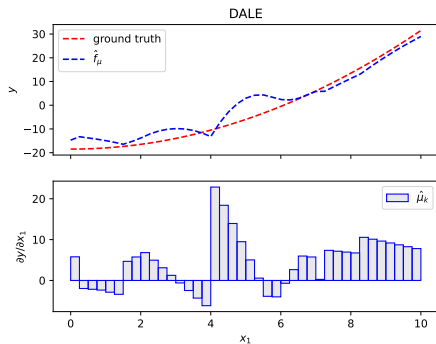
Intuition: we need wider bins (more samples per bin)

DALE vs ALE - 40 Bins



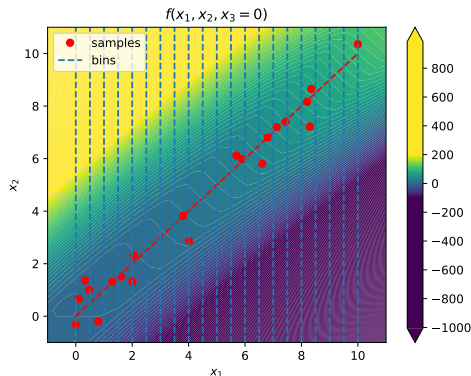
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 40 Bins



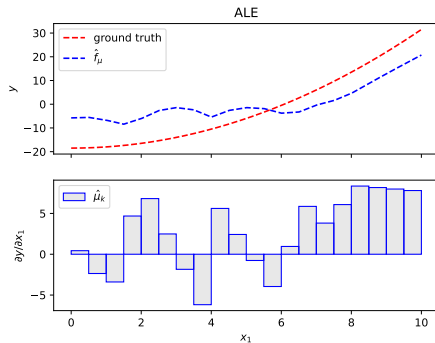
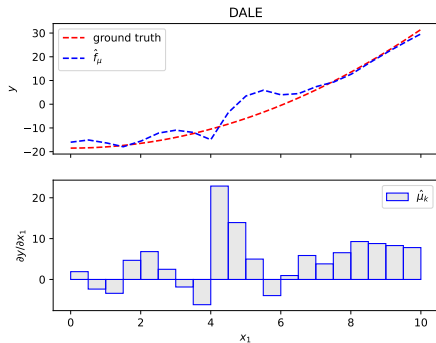
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 20 Bins



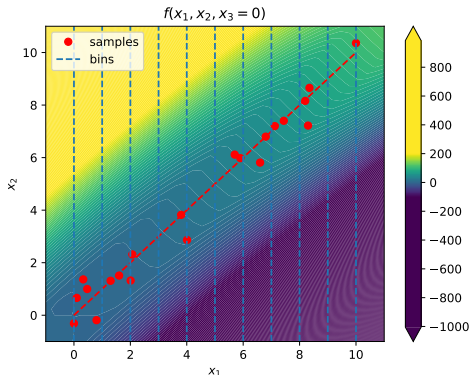
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 20 Bins



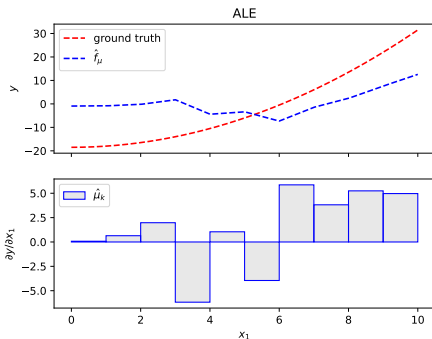
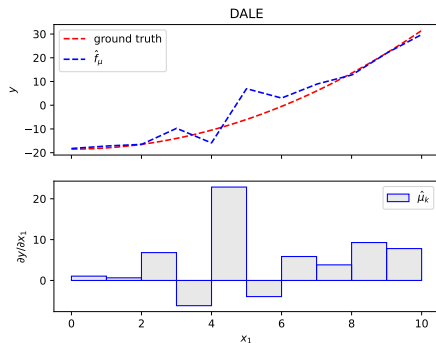
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 10 Bins



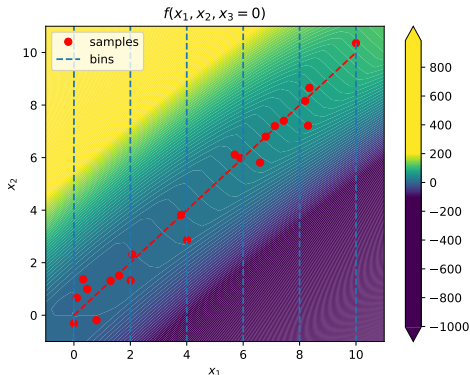
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: starts being OOD, noisy bin effect → poor estimation

DALE vs ALE - 10 Bins



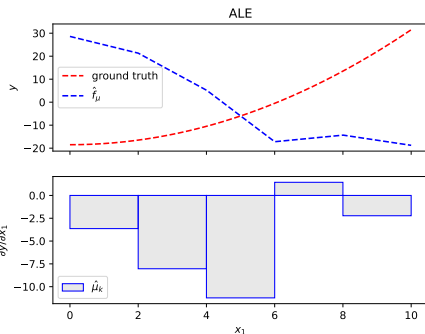
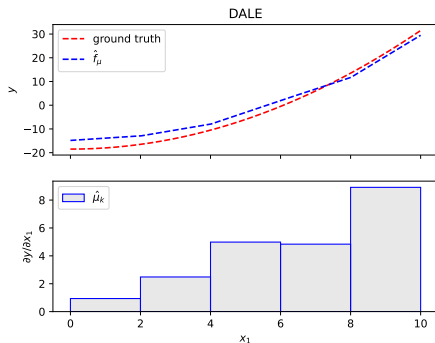
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: starts being OOD, noisy bin effect → poor estimation

DALE vs ALE - 5 Bins



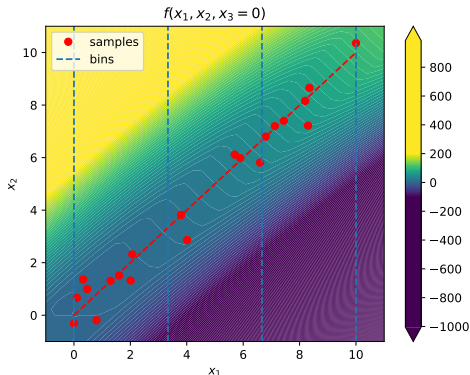
- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

DALE vs ALE - 5 Bins



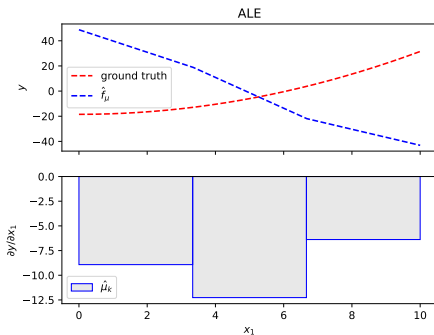
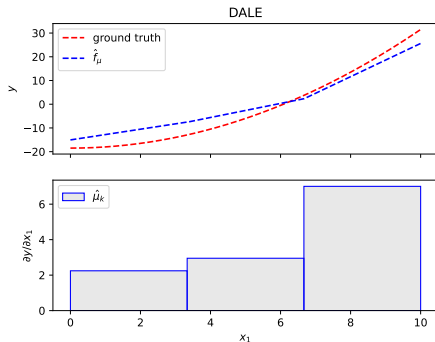
- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

DALE vs ALE - 3 Bins



- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

DALE vs ALE - 3 Bins



- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

Real Dataset Experiments - Efficiency

- Bike-sharing dataset(Fanaee-T and Gama, 2013)
- $y \rightarrow$ daily bike rentals
- x : 10 features, most of them characteristics of the weather

Efficiency on Bike-Sharing Dataset (Execution Times in seconds)

	Number of Features										
	1	2	3	4	5	6	7	8	9	10	11
DALE	1.17	1.19	1.22	1.24	1.27	1.30	1.36	1.32	1.33	1.37	1.39
ALE	0.85	1.78	2.69	3.66	4.64	5.64	6.85	7.73	8.86	9.9	10.9

DALE requires almost same time for all features

Real Dataset Experiments - Accuracy

- Difficult to compare in real world datasets
- We do not know the ground-truth effect
- In most features, DALE and ALE agree.
- Only X_{hour} is an interesting feature

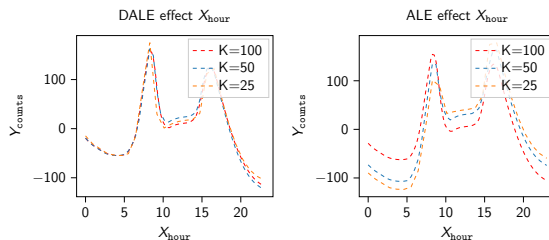


Figure: (Left) DALE (Left) and ALE (Right) plots for $K = \{25, 50, 100\}$

What next?




- Could we automatically decide the optimal bin sizes?
 - ▶ Sometimes narrow bins are ok
 - ▶ Sometimes wide bins are needed
- What about variable size bins?
- Model the uncertainty of the estimation?

DALE can be a driver for future work

Thank you

- Questions?

References I

-  Apley, Daniel W. and Jingyu Zhu (2020). “Visualizing the effects of predictor variables in black box supervised learning models”. In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 82.4, pp. 1059–1086. ISSN: 14679868. DOI: [10.1111/rssb.12377](https://doi.org/10.1111/rssb.12377). arXiv: [1612.08468](https://arxiv.org/abs/1612.08468).
-  Fanaee-T, Hadi and Joao Gama (2013). “Event labeling combining ensemble detectors and background knowledge”. In: *Progress in Artificial Intelligence*, pp. 1–15. ISSN: 2192-6352. DOI: [10.1007/s13748-013-0040-3](https://doi.org/10.1007/s13748-013-0040-3). URL: [\[WebLink\]](#).
-  Friedman, Jerome H. (2001). “Greedy function approximation: A gradient boosting machine”. In: *Annals of Statistics* 29.5, pp. 1189–1232. ISSN: 00905364. DOI: [10.1214/aos/1013203451](https://doi.org/10.1214/aos/1013203451).
-  Molnar, Christoph (2022). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. URL: <https://christophm.github.io/interpretable-ml-book>.