


RETHINKING MODEL DEVELOPMENT AND VALIDATION

Agus Sudjianto, Ph.D.



CHALLENGE WITH MODELS

- Models pass aggregate performance checks
 - But models often do not work well when business really need them
- Models still fail in production due to hidden vulnerabilities
 - Despite model validation, we are often surprised when model fails
- Reliance on model monitoring to identify model failures
 - Yet, model monitoring is reactive and backward looking
-  Regulatory gaps: “Effective Challenge” by model validators often seen as inadequate
 - ➔ Need to change mindset and approach!

KEY MINDSET: MODEL HACKING

A proactive, systematic approach to uncovering hidden vulnerabilities and weaknesses

Conceptual Alignment

Build inherently interpretable high-performance ML: GBM, GAMI-Net, Neural Tree, Mixture of Experts

- Architecture-level transparency (no post-hoc approximations)
- Enforce conceptual soundness: shape, sparsity, and causal constraints

Vulnerability Testing

Test comprehensively for edge cases and hidden weakness

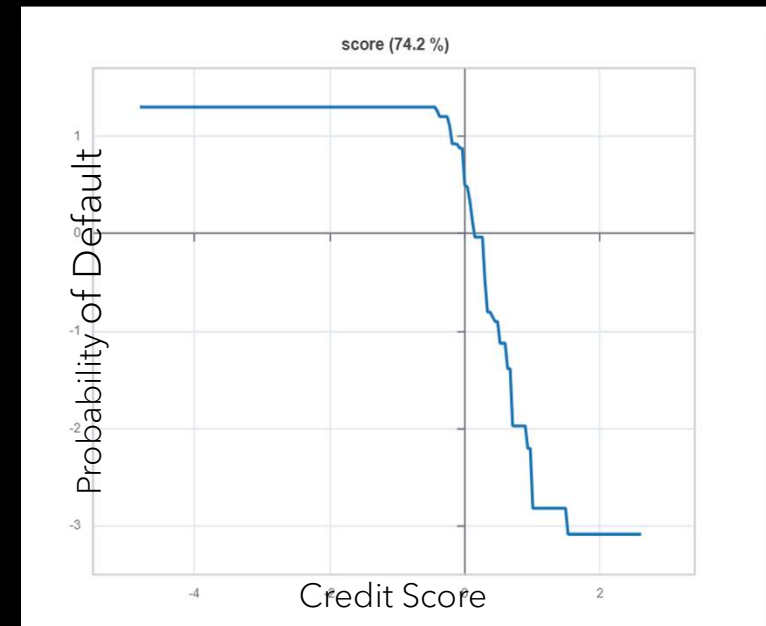
- **Heterogeneity:** Identify weak operating regions
- **Resilience:** Identify fragility against distribution drift
- **Reliability:** Identify high-uncertainty predictions
- **Robustness:** Identify sensitivity to input noise

INHERENTLY INTERPRETABLE MACHINE LEARNING

Foundation: functional ANOVA (fANOVA) representation

$$g(\mathbb{E}(y|\mathbf{x})) = g_0 + \sum_j g_j(x_j) + \sum_{j < k} g_{jk}(x_j, x_k) + \sum_{j < k < l} g_{jkl}(x_j, x_k, x_l) + \dots$$

- Transparent decomposition of main and interaction effects
- Behavior control via shape constraints
- Sparsity and orthogonality control
- Sophisticated machine learning implementation: Gradient Boosted Decision Tree, Gradient Boosted Linear Tree, Neural Networks and Mixture of Experts
- Powerful alternative models and benchmarks



IDENTIFICATION OF HIDDEN VULNERABILITY

Aggregate metrics like AUC or MSE
mask severe performance issues in
specific operating regions

Identification of hidden weakness
(failure clustering) via supervised
interpretable machine learning

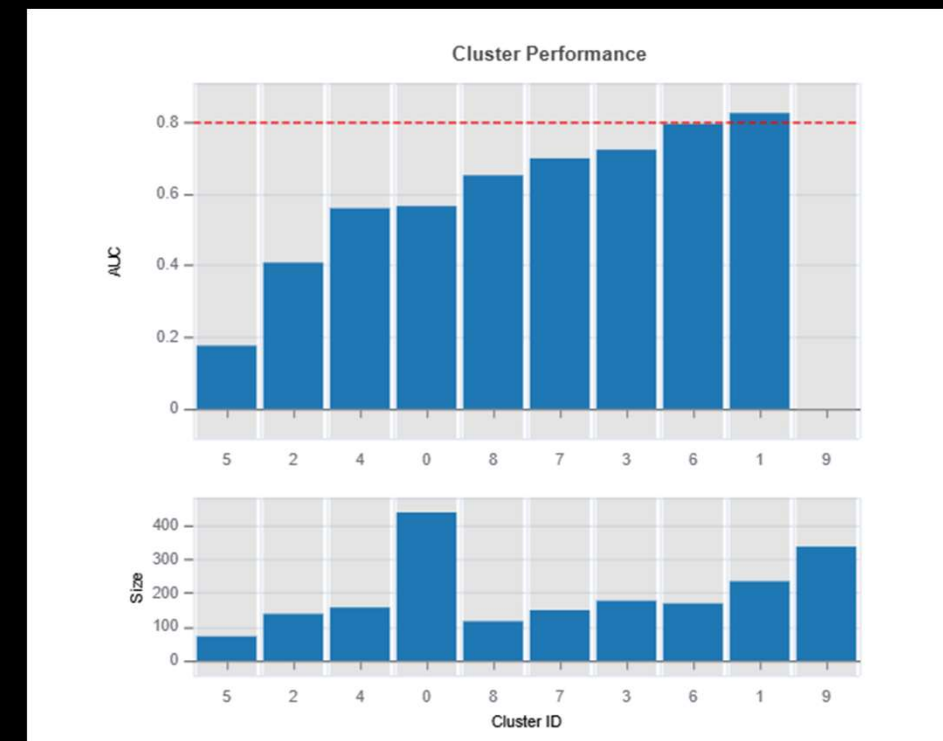
Anticipate fragility against distribution
drift prior model deployment

Identify "*harmful*" variables

Fix performance heterogeneity
problems

Facilitate forward-looking monitoring

	AUC	ACC	F1	LogLoss	Brier
train	0.817082	0.79125	0.296546	0.397061	0.132967
test	0.809924	0.79900	0.289753	0.394764	0.132797
GAP	-0.007157	0.00775	-0.006793	-0.002297	-0.000169



IDENTIFICATION OF RELIABILITY ISSUES

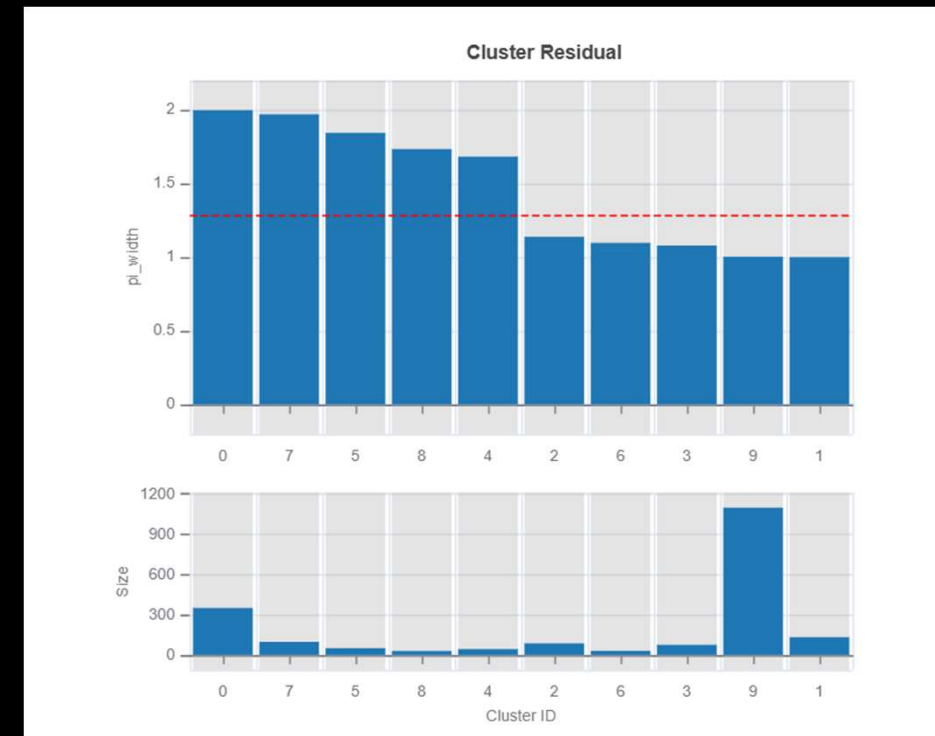
Accuracy is misleading

Model can be overconfident as outcomes have inherent uncertainty

Using machine learning and conformal prediction to quantify uncertainty

Supervised clustering to identify high uncertain regions and important variables driving uncertainty

	AUC	ACC	F1	LogLoss	Brier
train	0.817082	0.79125	0.296546	0.397061	0.132967
test	0.809924	0.79900	0.289753	0.394764	0.132797
GAP	-0.007157	0.00775	-0.006793	-0.002297	-0.000169



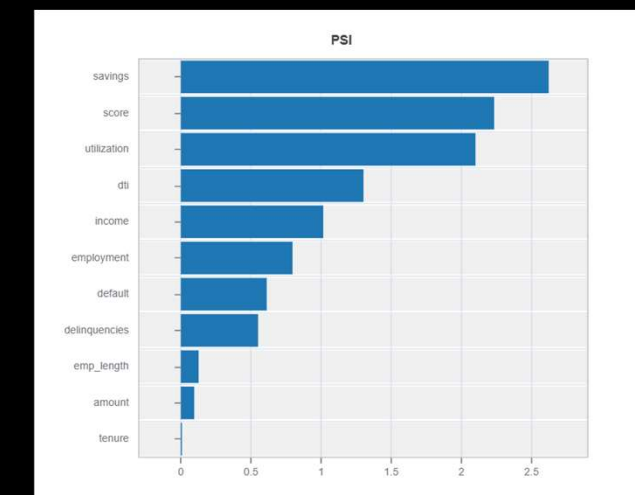
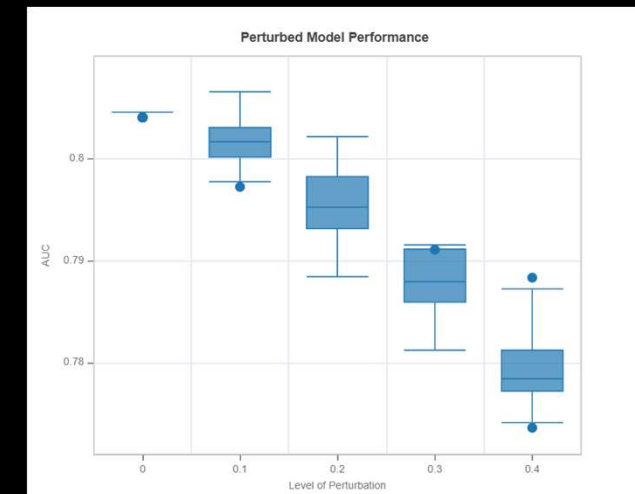
IDENTIFICATION OF ROBUSTNESS ISSUES

Robustness test: assessing sensitivity of model outputs against small input noise

- Model can be unstable in production due to sensitivity to input noise
- Machine learning models frequently suffered from “benign overfitting”

Testing model performance using perturbation of test data

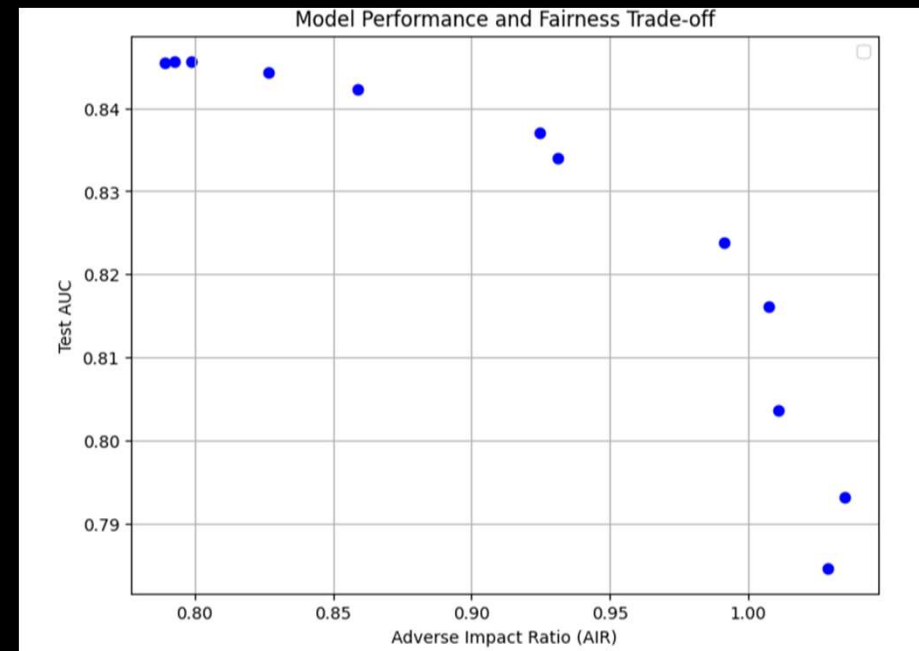
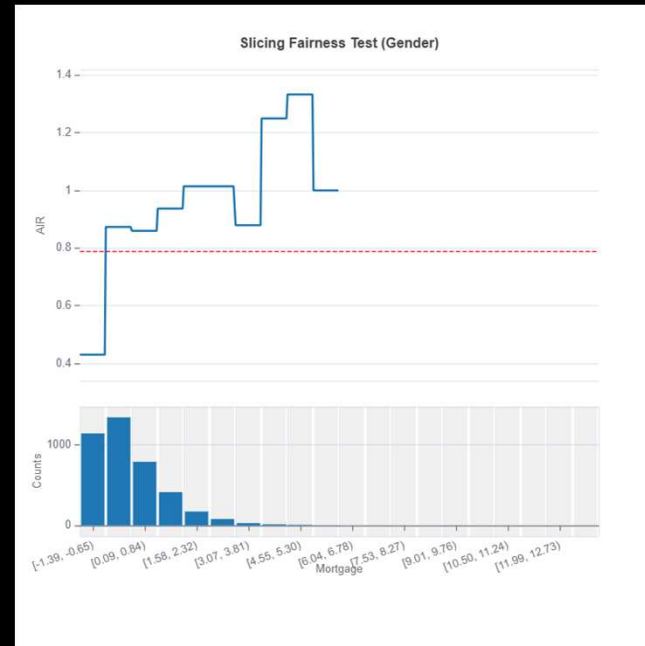
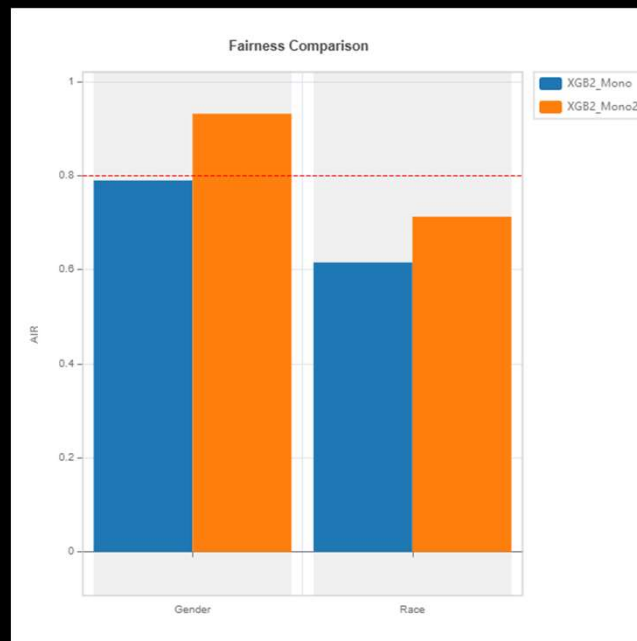
Apply robustness clustering to identify sensitive regions and variables causing instability



IDENTIFICATION OF FAIRNESS ISSUES

Testing for fairness and identification of problematic segments and variables

Fix fairness issue via model de-biasing or accuracy/bias trade-off optimization

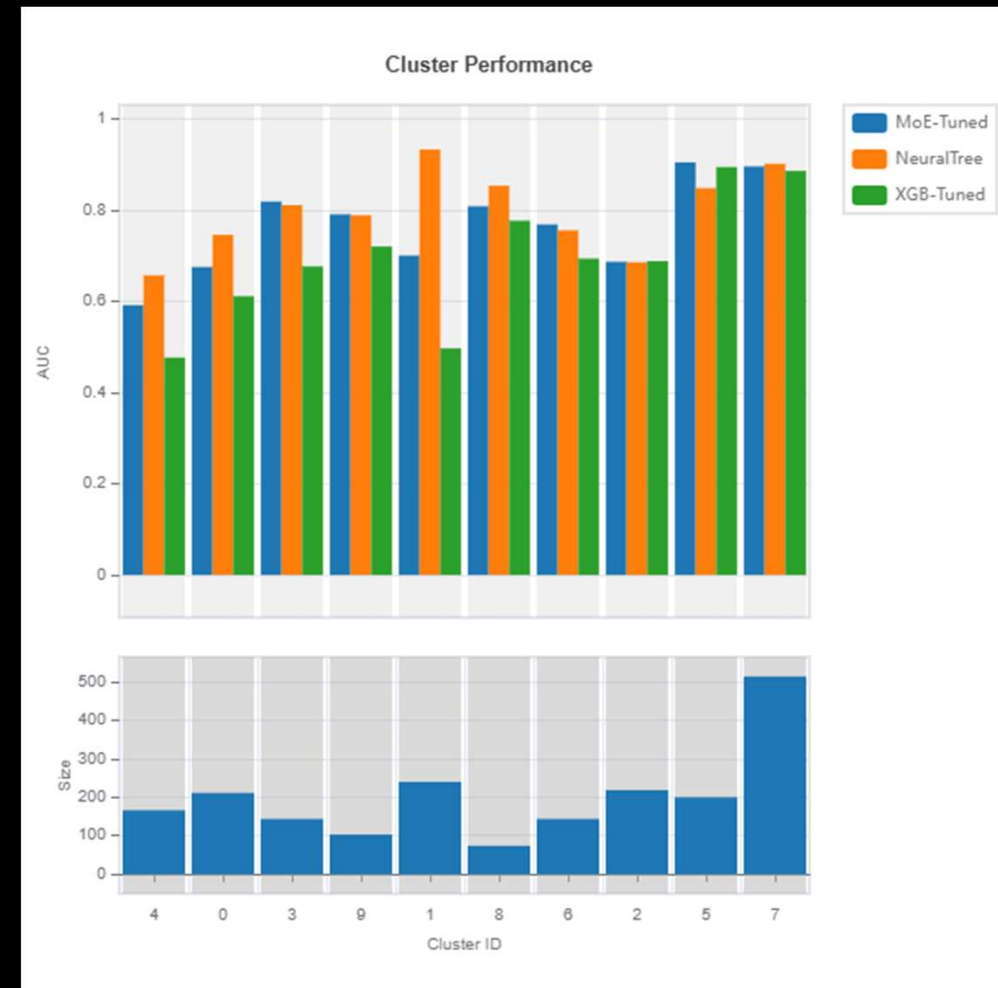


FIX PERFORMANCE HETEROGENEITY

Apply mixture of experts (MoE) to fix performance heterogeneity issues:

- Mixture of experts of interpretable boosting models
- Neural-Tree: mixture of linear models

Model is more resilience against distribution drifts





One of a kind integrated Python library for inherently interpretable machine learning model development and model agnostic testing focusing on failure testing--assumption and model-free approach for black-box testing

Data Quality Check

Performance weakness identification

Variable Selection and Causality Test

Under/Overfitting identification

Inherently Interpretable Models

Resilience Evaluation

Post-hoc explanation

Reliability Evaluation

Hyper-parameter Optimization

Robustness Evaluation

Heterogeneity diagnostics for model segmentation

Replication and benchmark

Fairness and De-biasing

Universal model wrapper for black-box testing