PiML Training:

AI/ML Outcome Analysis

Aijun Zhang, Ph.D.
Corporate Model Risk, Wells Fargo

Information Sharing at Commonwealth Bank | March 22, 2024

Disclaimer: This material represents the views of the presenter and does not necessarily reflect those of Wells Fargo.

Biographical Sketch

- Aijun Zhang is a senior vice president, Head of Validation Engineering at Wells Fargo. He leads a machine learning & validation engineering team in Corporate Model Risk, responsible for PiML (Python interpretable machine learning) toolbox and VoD (Validation-on-Demand) platform.

- Aijun holds PhD degree in Statistics from University of Michigan at Ann Arbor, and he has 10+ years of experience working in financial risk management. Aijun was a former professor of statistics at University of Hong Kong. He has published ~40 papers in professional conferences and journals, with research topics in interpretable machine learning, data science and statistics.

Outline

- **PiML Toolbox Overview**
- **Outcome Analysis**
  - Prediction Accuracy
  - Weakness Detection
  - Overfitting Regions
  - Prediction Uncertainty
  - Robustness and Resilience
  - Bias and Fairness
- **PiML User Guide and Examples**

PiML Toolbox Overview

PiML

An integrated Python toolbox for interpretable machine learning

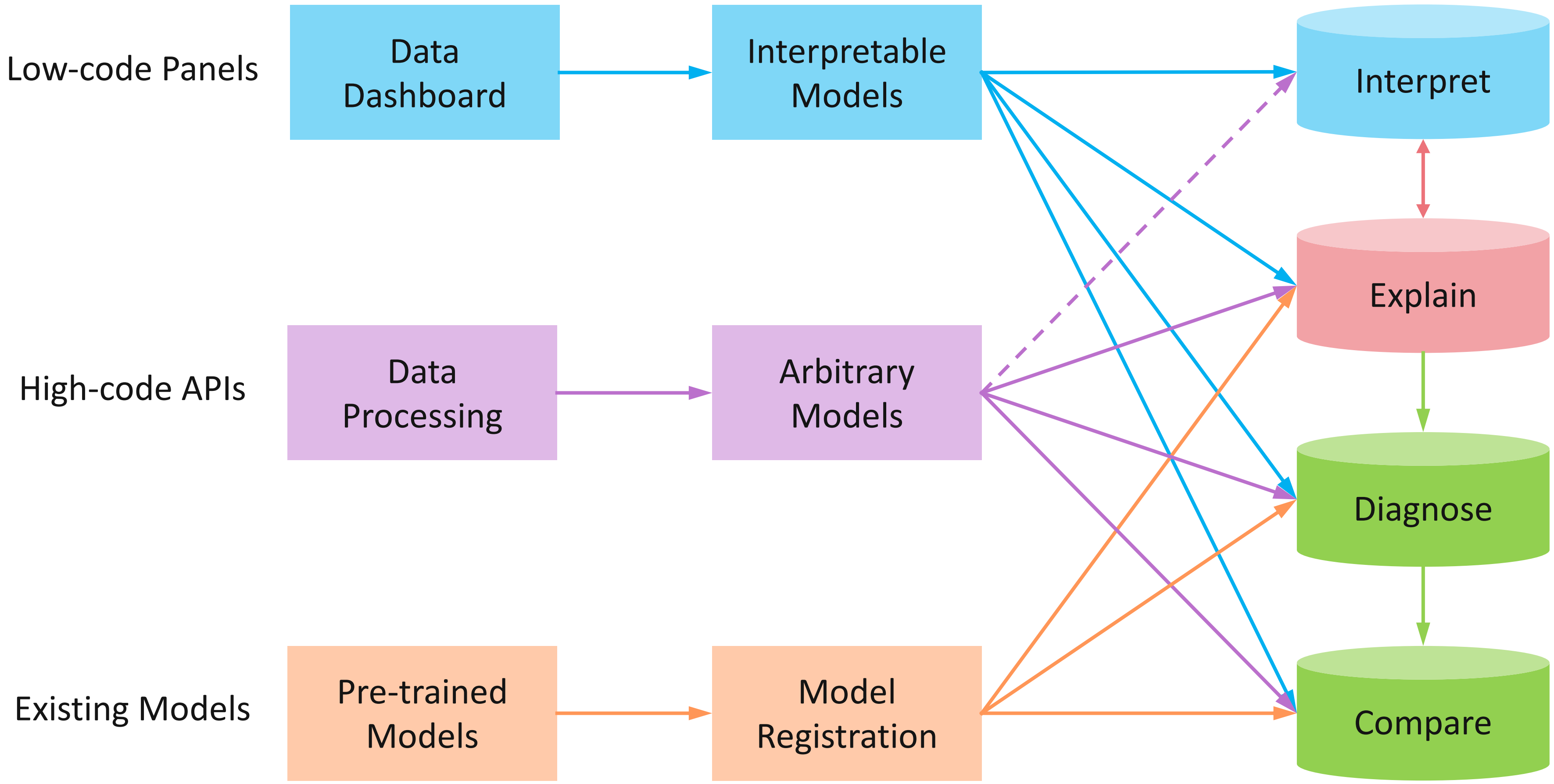
Model Development

- Data Exploration and Quality Check
- Inherently Interpretable ML Models
  - GLM, GAM, XGB1
  - XGB2, EBM, GAMI-Net, GAMI-Lin-Tree
- Locally Interpretable ML Models
  - Tree, Sparse ReLU Neural Networks
- Model-specific Interpretability
- Model-agnostic Explainability

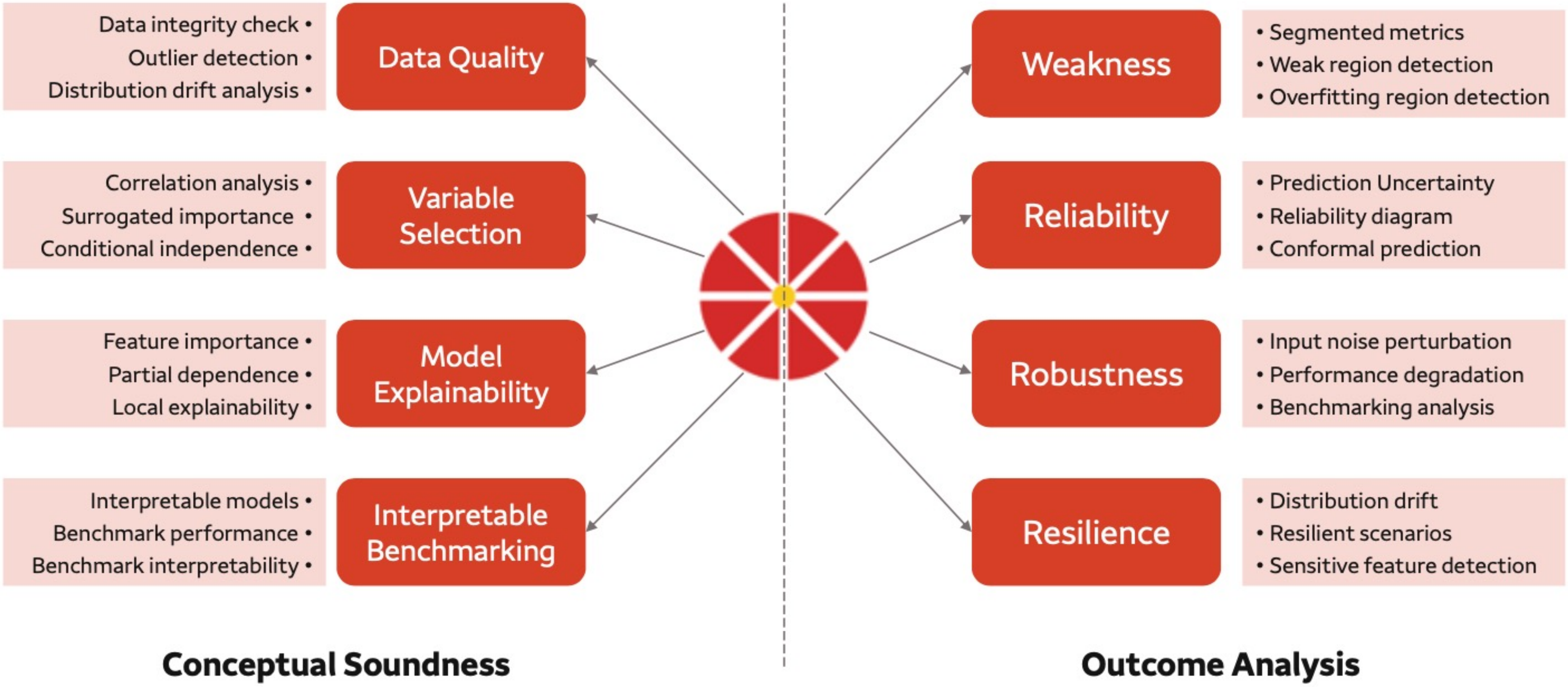
Model Testing

- Model Diagnostics and Outcome Testing
  - Prediction Accuracy
  - Hyperparameter Turning
  - Weakness Detection
  - Reliability Test (Prediction Uncertainty)
  - Robustness Test
  - Resilience Test
  - Bias and Fairness
- Model Comparison and Benchmarking

PiML Pipelines



PiML Elements for Model Validation

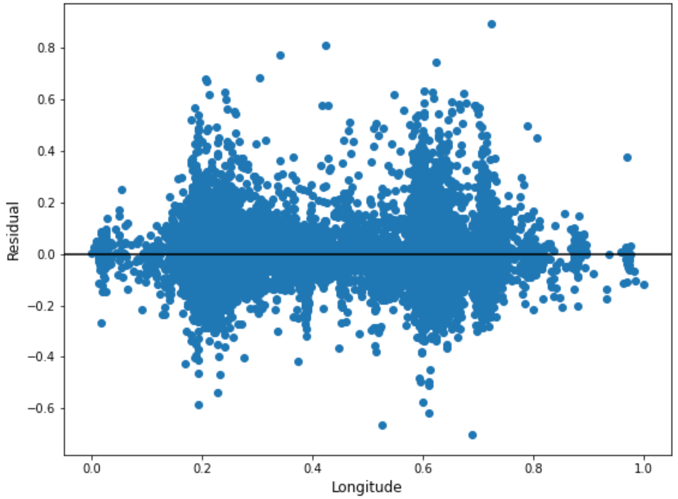


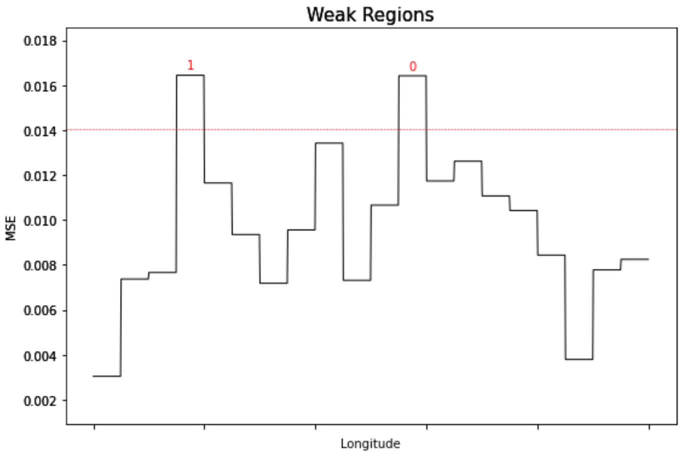
Outline

- **PiML Toolbox Overview**
- **Outcome Analysis**
  - Prediction Accuracy
  - Weakness Detection
  - Overfitting Regions
  - Prediction Uncertainty
  - Robustness and Resilience
  - Bias and Fairness
- **PiML User Guide and Examples**

Prediction Accuracy and Residual Analysis

- Machine learning model performance is often evaluated by **prediction accuracy,** using metrics such as MSE, MAE, R2, ACC, AUC, F1-score.
- However, model assessment by single-valued metrics is insufficient. More detailed diagnostics and evaluation are required.
- Residual analysis to check model performance in a more granular manner,
  - **Residual plot** marginally for each feature of interest;
  - **Segmented metrics** by feature binning (uniform, quantile and auto);
  - **WeakSpot** to identify weak regions with high residuals on either training or testing data.
- **PiML toolbox** employs segmented diagnostics and error slicing techniques.





Weakness Detection by Error Slicing

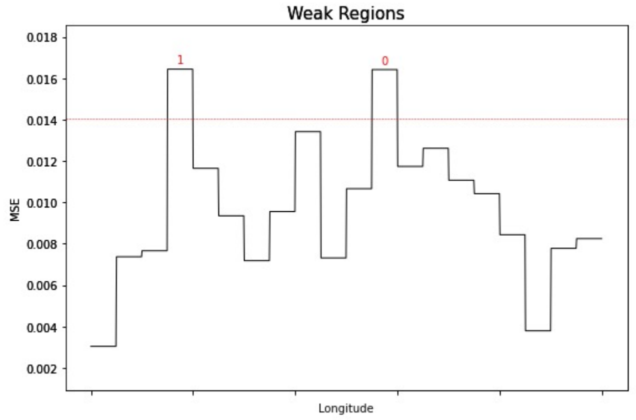
1. **Specify an appropriate metric** based on individual prediction residuals: e.g., MSE for regression, ACC/AUC for classification, train-test performance gap (for checking overfit), etc.

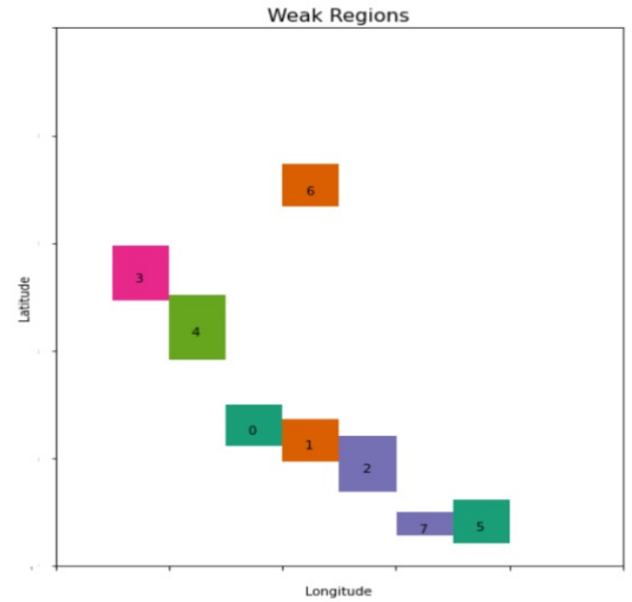
2. Specify 1 or 2 slicing features of interest;

3. Evaluate the metric for each sample in the target data (training or testing) as pseudo responses;

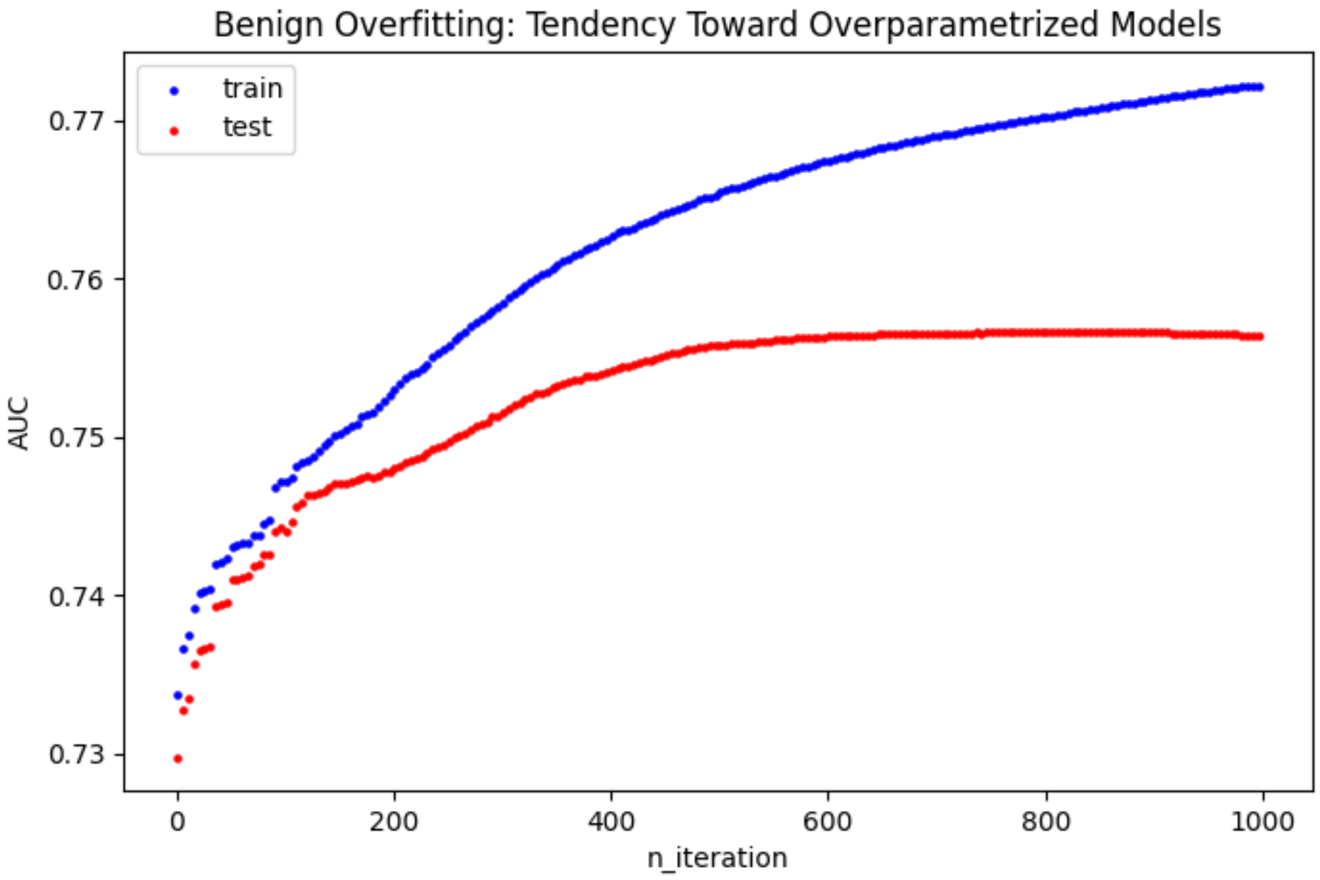
4. **Segment the target data** along the slicing features, by
   a) [Unsupervised] Histogram slicing with equal-space binning, or
   b) [Supervised] fitting a decision tree to generate the sub-regions

5. **Identify the sub-regions** with average metric exceeding the pre-specified threshold, subject to minimum sample condition.



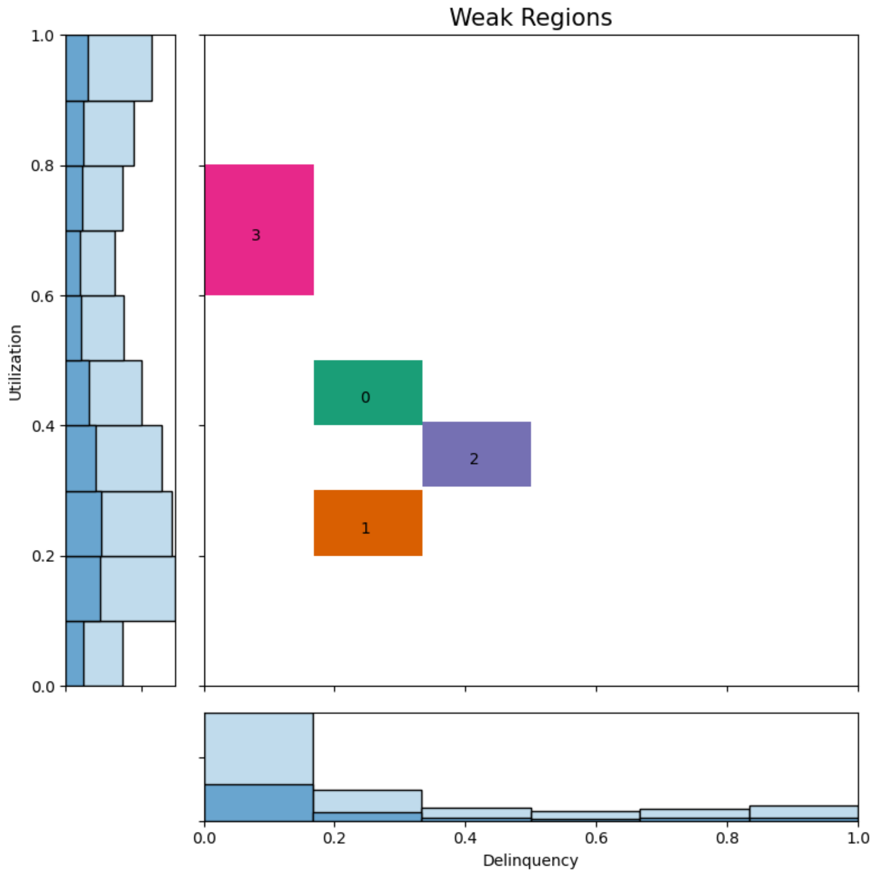


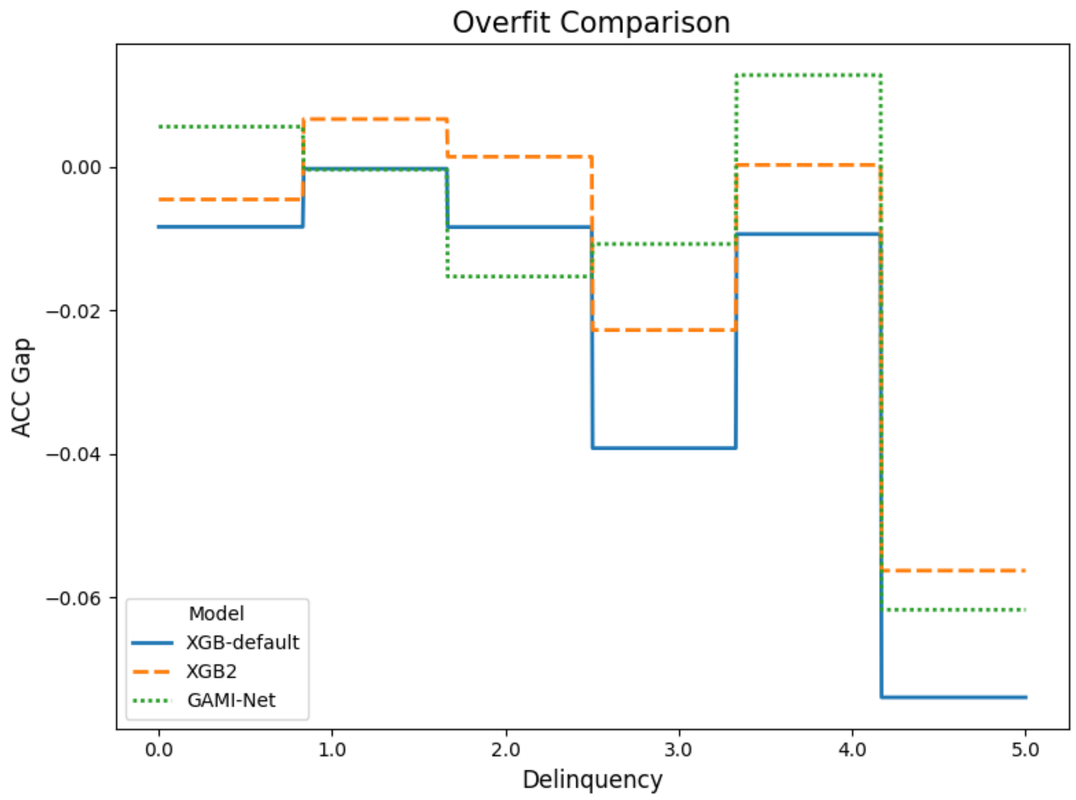
Benign Overfitting Phenomenon



PiML Demo: Benign overfitting phenomenon observed on XGBoost models (SimuCredit Data)

PiML Demo: WeakSpot and Overfit





PiML Demo: WeakSpot and Overfit analysis for SimuCredit Data (XGB-default vs. Benchmark models)

Prediction Uncertainty Quantification

- Prediction uncertainty is important to understand where the model produces less reliable prediction:

  Wider prediction interval $\rightarrow$ Less reliable prediction

- Quantification of prediction uncertainty can be done through **Split Conformal Prediction** under the exchangeability assumption:

Given a pre-trained model $\hat{f}(\boldsymbol{x})$, a hold-out calibration data $\mathcal{X}_{\text{calib}}$, a pre-defined conformal score $S(\boldsymbol{x}, y, \hat{f})$ and the error rate α (say 0.1)

1. Calculate the score $S_i = S(\boldsymbol{x}, y, \hat{f})$ for each sample in $\mathcal{X}_{\text{calib}}$;
2. Compute the calibrated score quantile

$$\hat{q} = \text{Quantile}\left(\{S_1, \ldots, S_n\}; \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1}\right);$$

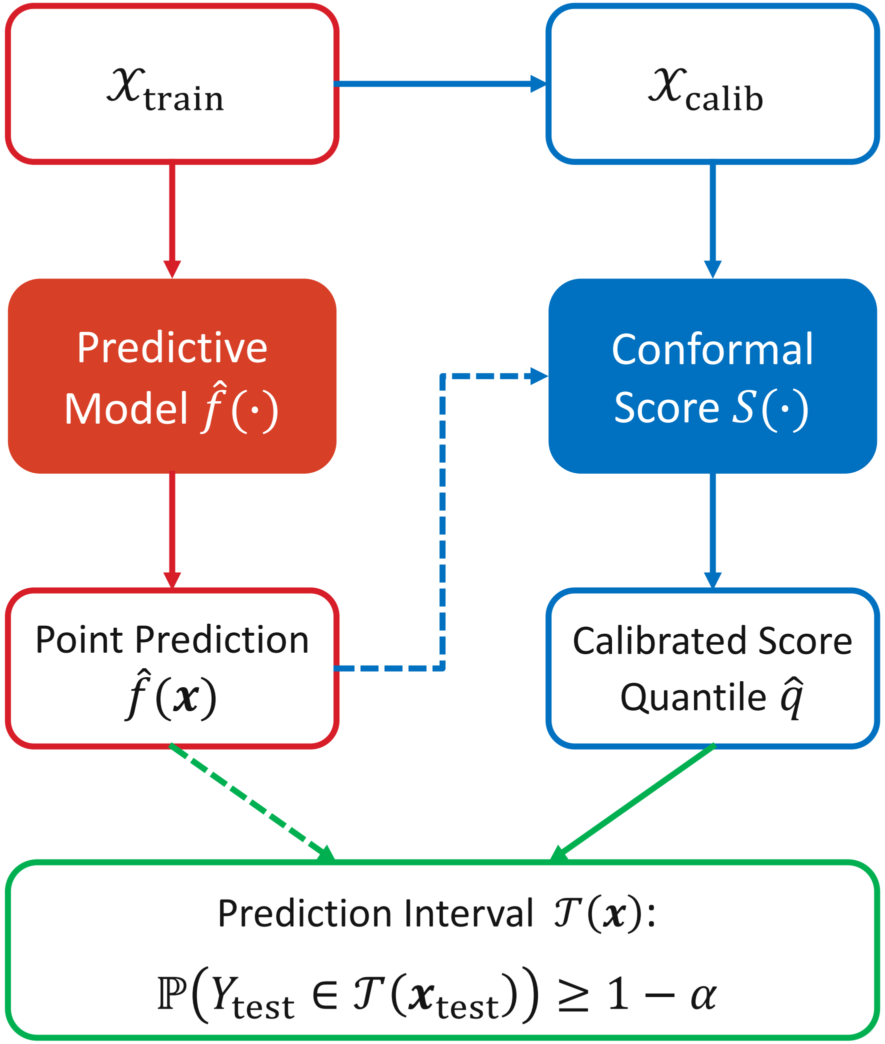
3. Construct the prediction set for the test sample $\boldsymbol{x}_{\text{test}}$ by

$$\mathcal{T}(\boldsymbol{x}_{\text{test}}) = \left\{y : S\left(\boldsymbol{x}_{\text{test}}, y, \hat{f}(\boldsymbol{x}_{\text{test}})\right) \leq \hat{q}\right\}.$$

Under the exchangeability condition of conformal scores, we have that

$$1 - \alpha \leq \mathbb{P}\big(Y_{\text{test}} \in \mathcal{T}(\boldsymbol{x}_{\text{test}})\big) \leq 1 - \alpha + \frac{1}{n+1}.$$

This provides the prediction bounds with α-level acceptable error.



Conformalized Residual Quantile Regression

Directly evaluate prediction uncertainty of a pre-trained regression model $\hat{f}(\boldsymbol{x})$:

1. Obtain residuals $y_i - \hat{f}(\boldsymbol{x}_i)$ for each $i \in \mathcal{X}_{\text{train}}$ or $\mathcal{X}_{\text{split}}$, fit a quantile regressor (e.g. LightGBM with quantile loss) for residuals $\left[\hat{g}_{\alpha/2}(\boldsymbol{x}),\ \hat{g}_{1-\alpha/2}(\boldsymbol{x})\right]$;

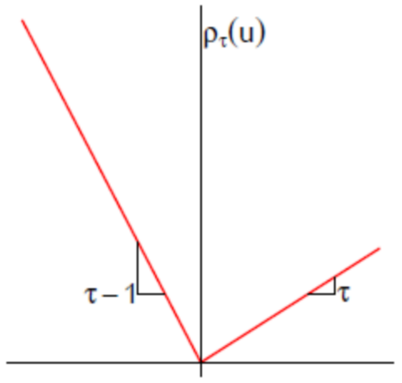
2. Define score $S\left(\boldsymbol{x}, y, \hat{f}\right) = \max\left\{\hat{g}_{\alpha/2}(\boldsymbol{x}) - y + \hat{f}(\boldsymbol{x}),\ y - \hat{f}(\boldsymbol{x}) - \hat{g}_{1-\alpha/2}(\boldsymbol{x})\right\}$

3. Calculate $\hat{q} = \text{Quantile}\left(\{S_1, \dots, S_n\}; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right)$, using $S\left(\boldsymbol{x}, y, \hat{f}\right)$ on $\mathcal{X}_{\text{calib}}$

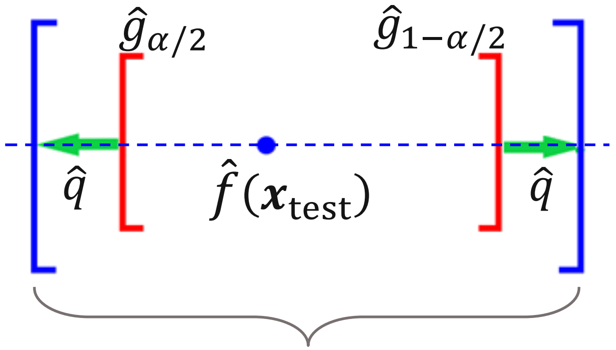
4. Construct the prediction interval for the test sample $\boldsymbol{x}_{\text{test}}$ by

$$\mathcal{T}(\boldsymbol{x}_{\text{test}}) = \left[\hat{f}(\boldsymbol{x}_{\text{test}}) + \hat{g}_{\alpha/2}(\boldsymbol{x}_{\text{test}}) - \hat{q},\ \ \hat{f}(\boldsymbol{x}_{\text{test}}) + \hat{g}_{1-\alpha/2}(\boldsymbol{x}_{\text{test}}) + \hat{q}\right].$$

Interpretation: the final prediction interval is composed of three terms: original prediction, estimated residual quantiles, and calibrated adjustment.



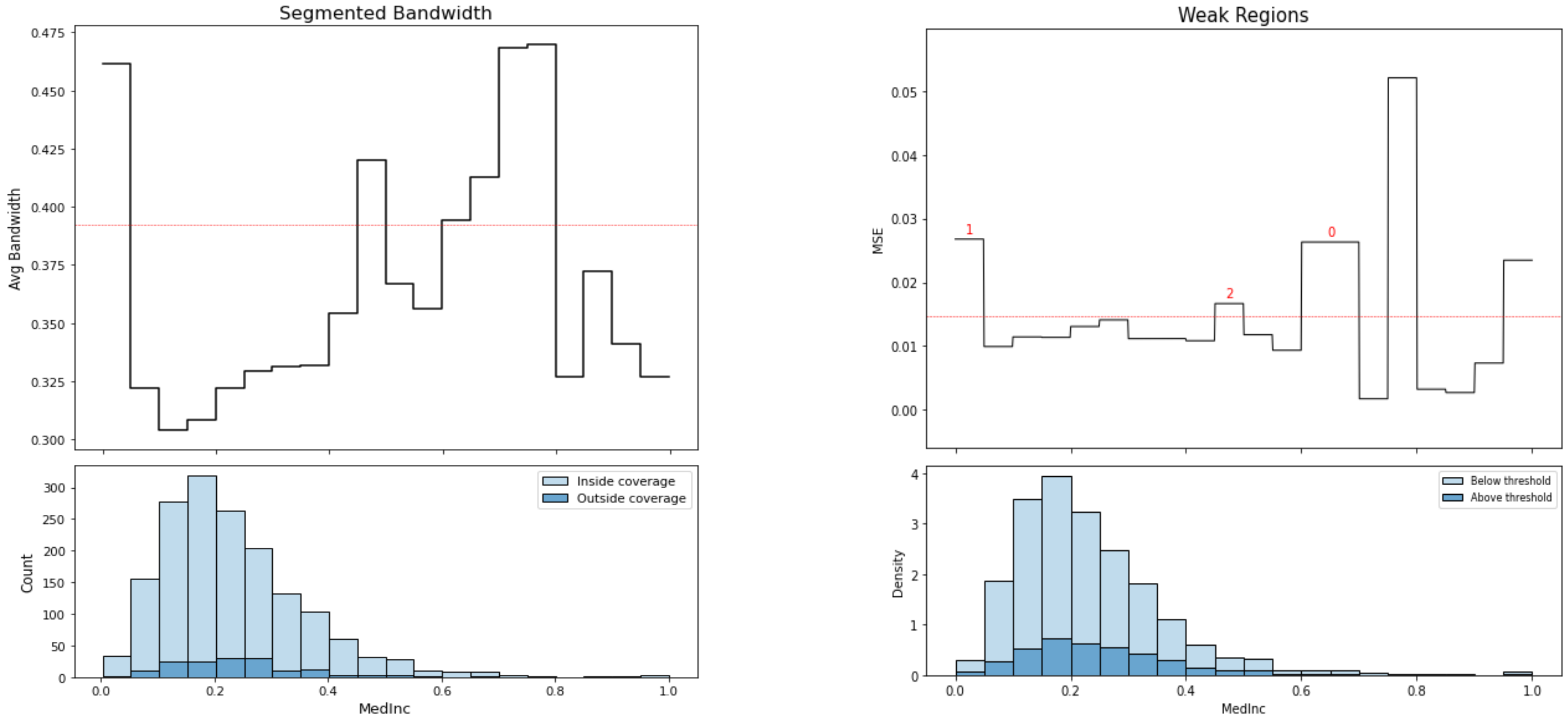
Quantile loss



Bandwidth

PiML Demo: Uncertainty Quantification

Note that quantile regression makes the interval bandwidth adaptive to heteroscedastic residuals.

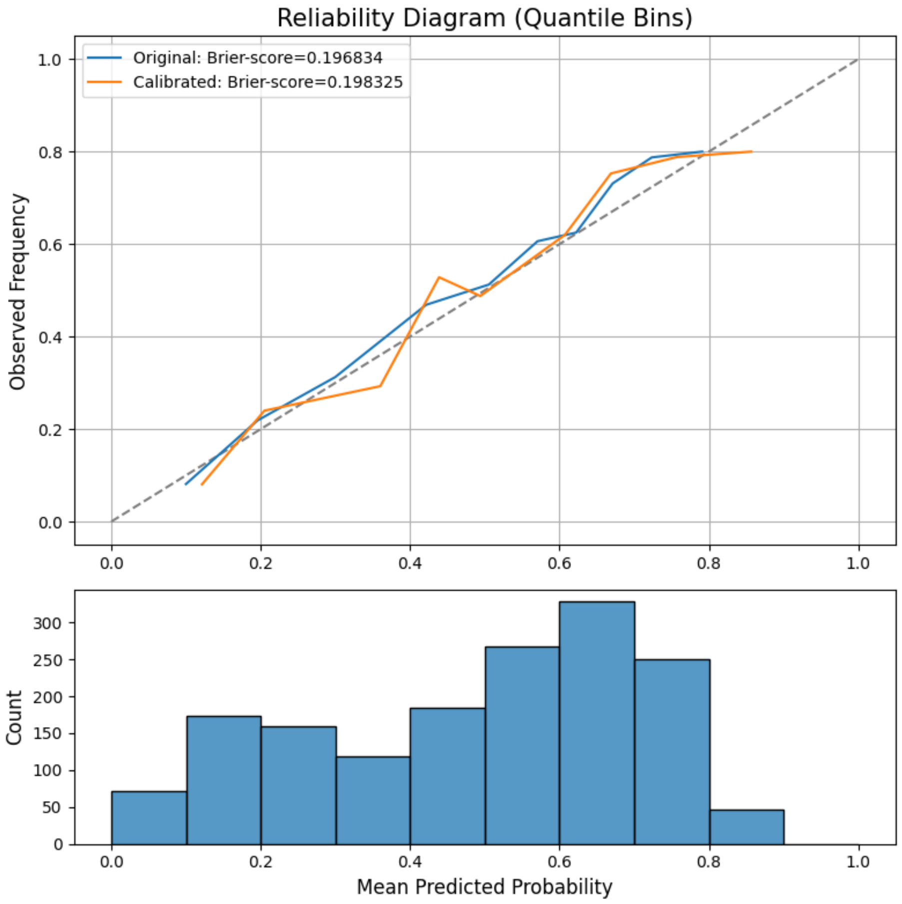


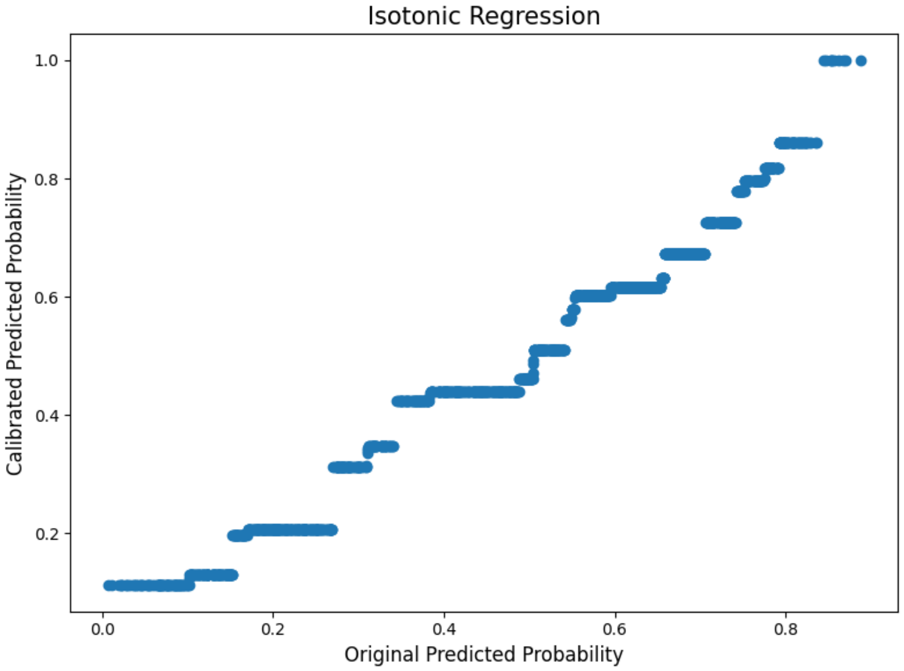
PiML Demo: Prediction Uncertainty Testing for CaliforniaHousing data fit by GAMI-Net.

Probability Calibration for Binary Classifiers

- The simple and easy conformal prediction does not work as effectively for the binary classification case.
- We take a conventional approach of using **predict_proba** $\hat{p} = \mathbb{P}(Y = 1|\boldsymbol{x})$ and measure the uncertainty by the quantity $\sqrt{\hat{p}(1-\hat{p})}$ for each point prediction.
- **Caveat:** there is no statistical guarantee of correct coverage of the true class.
- However, probability calibration is needed for raw predict_proba by some ML models, so the predicted probabilities align with the observed class frequencies, as shown by the reliability diagram or measured through the Brier score.
- In PiML, we adopt the isotonic regression to calibrate the predicted probabilities as a monotonic step function; while Platt scaling is a parametric sigmoid curve.
- **To-do:** statistically more sound and effective approach is being developed and will be released soon.

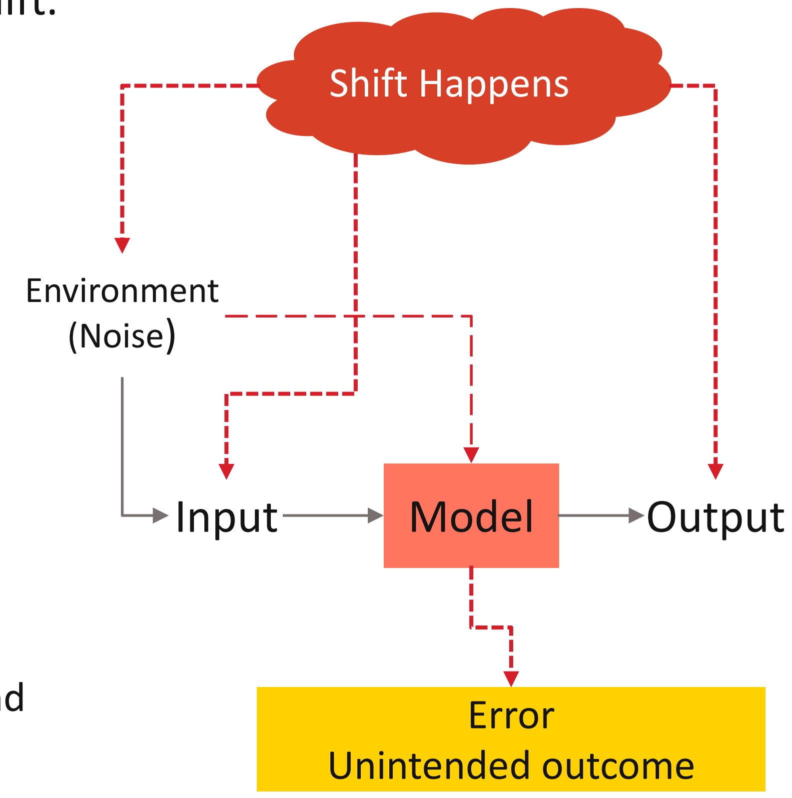
PiML Demo: Binary Classification Case





Robustness and Resilience Tests

- Train-test data split for model development often gives over-optimism of model performance, since model in production will be exposed to data distribution shift.
- **Robustness test**: evaluate the performance degradation under covariate noise perturbation:
  - Perturb testing data covariates with small random noise;
  - Assess model performance of perturbed testing data.
  - Overfitting models often perform poorly in changing environments.
- **Resilience test**: evaluate the performance degradation under distribution drift scenarios
  - Scenarios: worst-sample, worst-cluster, outer-sample, hard-sample
  - Measure distribution drift (e.g., PSI) of variables between worst performing sample and the remaining sample.
  - Variables with notable drift are deemed to be sensitive in the resilience test.



Measuring Distribution Shift

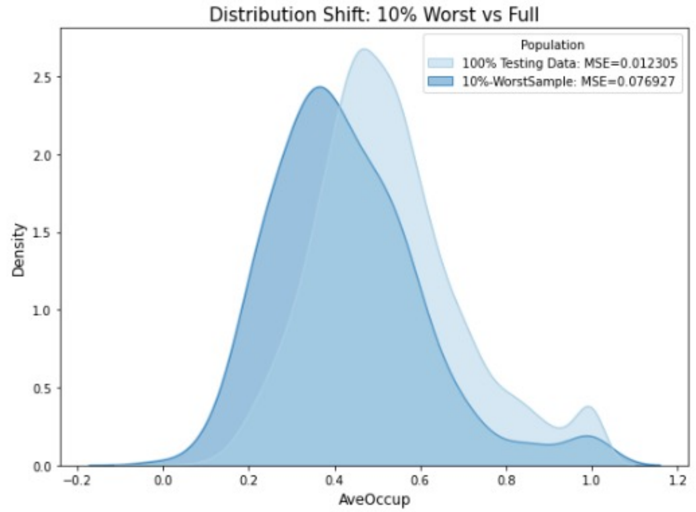
- **Population Stability Index:**

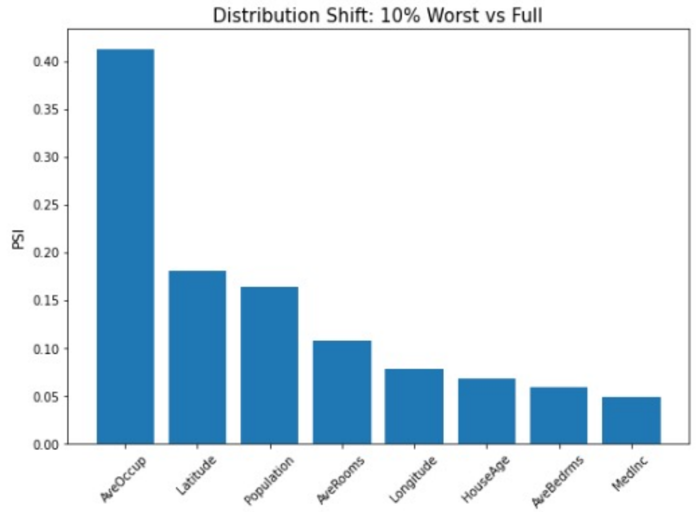
$$PSI = \sum_{i=1}^{B} (\text{Target}_i\% - \text{Base}_i\%) \ln\left(\frac{\text{Target}_i\%}{\text{Base}_i\%}\right)$$

  based on the proportions of samples in each bucket of the target vs. base population. Rule of thumb:
  - PSI < 0.1: no significant distribution change
  - PSI < 0.2: moderate distribution change
  - PSI >= 0.2: significant distribution change

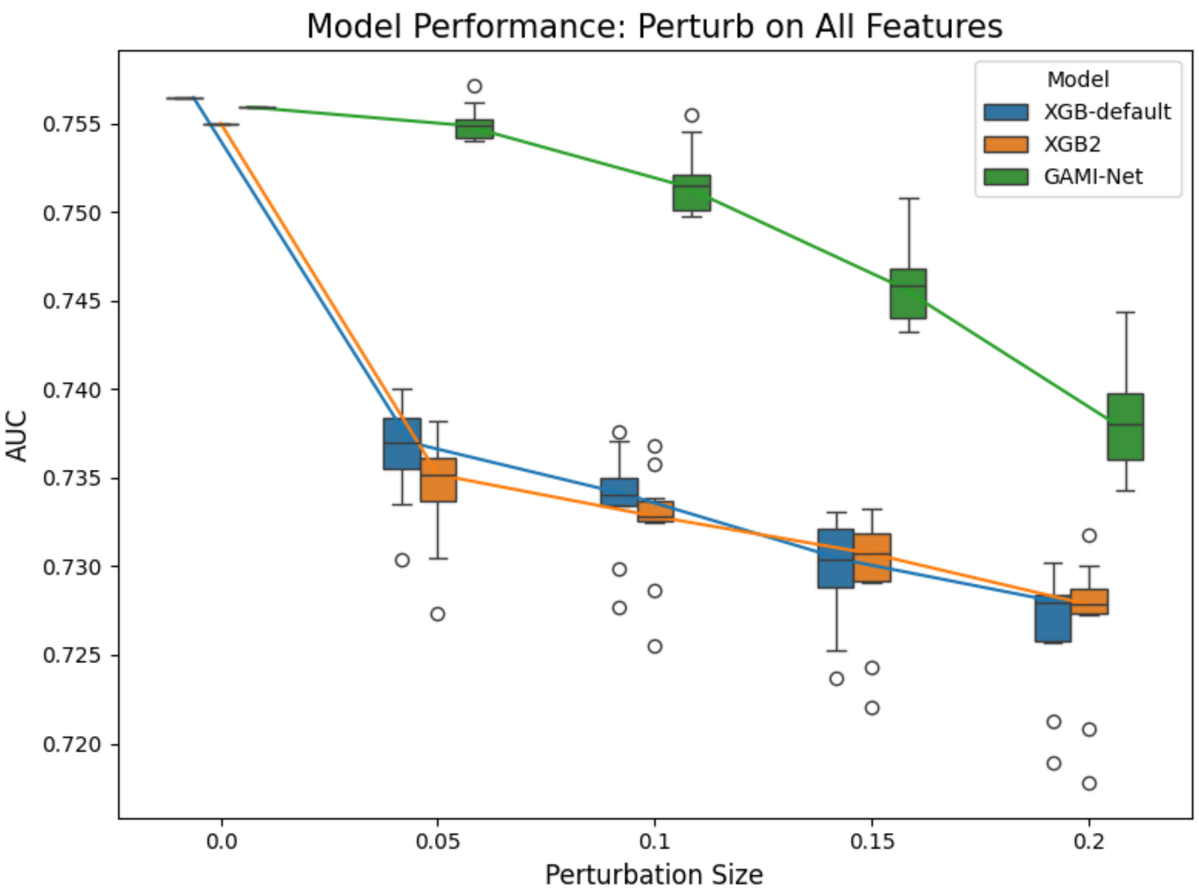
- Other two-sample test: KL divergence, Kolmogorov-Smirnov (KS) and Cramer-von Mises (CM) statistics based on empirical distributions.

- In resilience testing, PSI measures the distribution shift one-feature-at-a-time. One may further use WeakSpot to perform drill-down analysis on sensitive features.



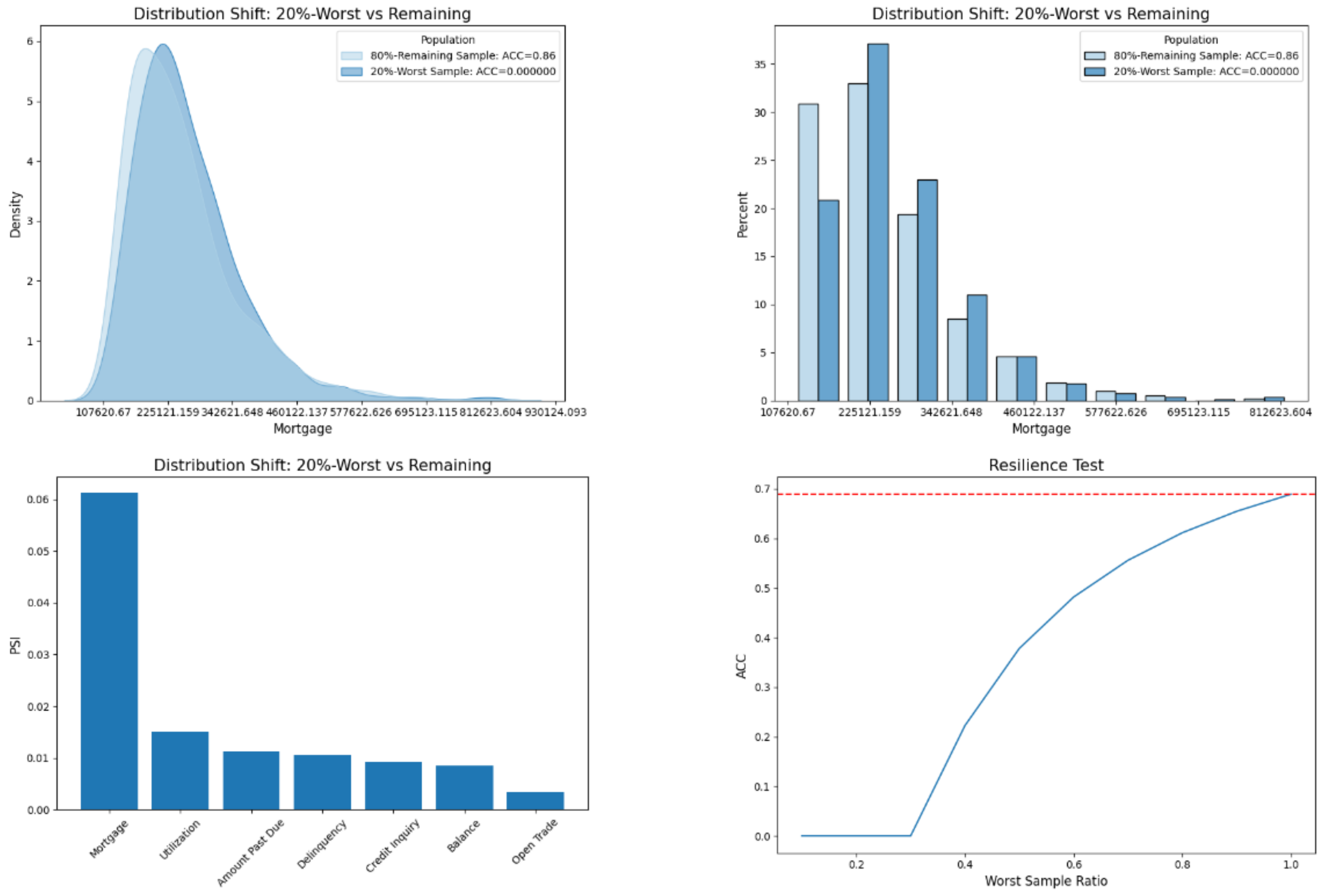


PiML Demo: Robustness Test



PiML Demo: Robustness Testing for SimuCredit data by XGB-default, XGB2 and GAMI-Net

PiML Demo: Resilience Test



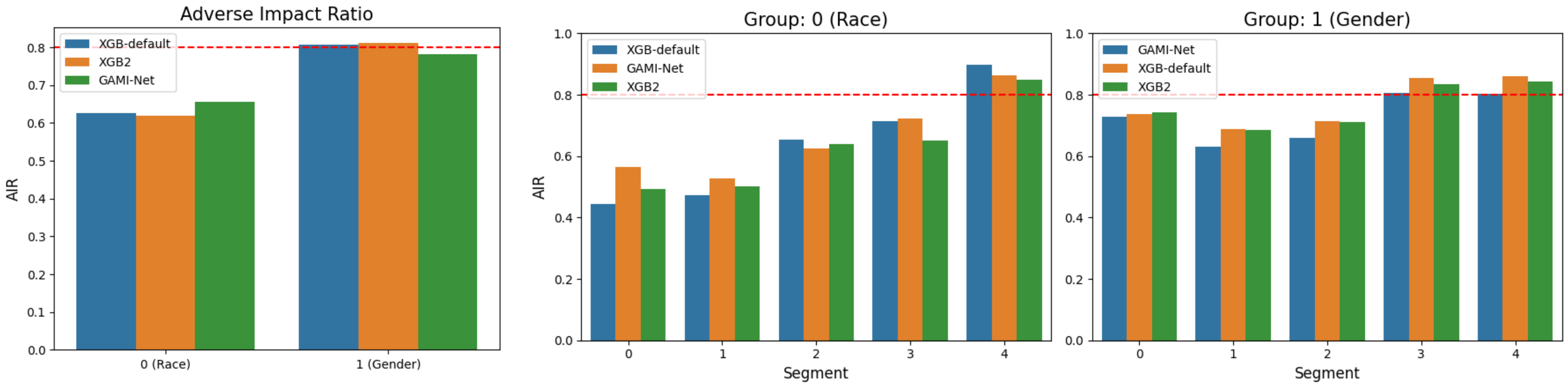
PiML Demo: Resilience Test and WeakSpot for SimuCredit data by XGB-default

Bias and Fairness

- For each demographic feature (Race, Gender), consider AIR between protected group vs reference group.

$$AIR = \frac{(TP_p + FP_p)/n_p}{(TP_r + FP_r)/n_r}$$

- AIR below 0.8 is a sign of bias and unfairness.
- PiML provides segmented metrics conditional on a modeling variable (e.g., Balance below). It also provides methods to debias through feature binning and decision thresholding.



Outline

- **PiML Toolbox Overview**
- **Outcome Analysis**
  - Prediction Accuracy
  - Weakness Detection
  - Overfitting Regions
  - Prediction Uncertainty
  - Robustness and Resilience
  - Bias and Fairness
- **PiML User Guide and Examples**

PiML User Guide and Examples

PiML — Install | User Guide | API | Examples — Go

Python Interpretable Machine Learning

pip install PiML

User Guide | GitHub

- A Python toolbox for interpretable machine learning
- Supports a growing list of inherently interpretable models
- Supports a diagnostic suite for model testing and validation
- Provides easy to use low-code interface and high-code APIs

Data Pipeline

Load, check, and prepare data

- Basic Pipeline: Load, Summary, Prepare
- Quality Check: Integrity, Outlier, Data drift
- Feature selection
- Exploratory data analysis

Interpretable Models

Inherent interpretability

- Main effect models: GLM, GAM, XGB1
- Interaction models: EBM, XGB2, GAMI-Net
- Local interpretable models: Tree, FIGS, ReLU-DNN

Post-hoc Explainability

Global and local explainability

- Global importance: PFI, H-statistic
- Global dependence: PDP, ALE
- Local methods: ICE, LIME, SHAP

Diagnostic Suite

Model validation and outcome testing

- Basic Tests: Accuracy, Weakspot, Overfit
- 3R Tests: Reliability, Robustness, Resilience
- Fairness test
- Segmented test
- Scored test

Model Comparison

Benchmarking through diagnostics

- Regression models
- Binary classification models
- Model fairness comparison

Low-Code Case Studies

PiML workflow and experimentation

- Example: Bikesharing Data
- Example: CaliforniaHousing Data
- Example: TaiwanCredit Data
- Fairness Simulation Study 1
- Fairness Simulation Study 2

https://selfexplainml.github.io/PiML-Toolbox

Thank you

Aijun Zhang, Ph.D.

Email: Aijun.Zhang@wellsfargo.com

LinkedIn: https://www.linkedin.com/in/ajzhang/