Reliability of Explainable AI (XAI)

Udit Jain

September 4, 2025

Current Scenario

0.1 Abstract

Problem Statement: Quantifying the reliability of Explainable Machine Learning (XAI) models in endocrinology.

Proposed Methodology: A general framework to evaluate the reliability of feature attribution-based XAI methods, focusing on how faithfully explanations align with the true model behavior.

Application: The methodology is demonstrated on two endocrine disease diagnosis models: Type 2 Diabetes and Thyroid dysfunction.

Note: Endocrinology is the branch of medicine and biology that deals with the study of endocrine glands (thyroid, pancreas, adrenal, pituitary), the hormones they produce, and how these hormones affect metabolism, growth, development, and mood.

0.2 Introduction

Problem. Machine Learning (ML) is increasingly applied to medical diagnosis using patient data. However, many ML models (Random Forests, Neural Networks) are "black boxes"—they give predictions without explanations. In medicine, doctors need to trust and understand predictions; otherwise, adoption is limited.

Role of XAI. Explainable AI (XAI) provides insights into model decisions by attributing importance to features (e.g., blood sugar, age, thyroid levels) using methods such as SHAP and LIME.

Issue with Current Explanations. These explanations may vary depending on training or test data. Feature importance can shift drastically, leading to unstable or misleading explanations. No standard framework exists to evaluate their reliability.

Contribution. This work proposes a structured methodology to test explanation reliability by:

- Combining XAI methods (SHAP, LIME) with k-fold cross-validation,
- Introducing new metrics to measure generalizability, concordance, and stability,
- Aggregating them into a *Global Reliability Score.*

0.3 Related Work

- **Human Feedback.** Explanations evaluated subjectively by users (e.g., doctors). Useful but subjective and inconsistent.
- **Mathematical Approaches.** Objective metrics designed to check stability, consistency, and logical alignment. More scientific, but can still mislead in practice.

- **Medical Imaging Studies.** Models with high accuracy (e.g., 99.5%) still failed to inspire trust without reliable explanations. Researchers proposed stability and rule-based checks.
- **UX and Recommender Systems.** Introduced ideas of transparency, trust, and scrutability—ensuring users could understand and even correct explanations.
- **Foundational Work.** Doshi-Velez and Kim highlighted desirable properties: causality, faithfulness, robustness, soundness vs. completeness.
- **Domain-specific Efforts.** Time-series researchers tested consistency over time; Bayesian researchers proposed BayLIME to stabilize explanations.

Across all domains, no universal standard exists for evaluating explanation reliability, especially in healthcare.

0.4 Methodology

We propose to combine XAI with k-fold cross-validation and evaluate using three metrics: **Generalizability**, **Concordance**, and **Stability**. These are aggregated into a **Global Reliability Score**.

0.5 Step 1: XAI with k-Fold Cross Validation

1. Split dataset into k folds.
2. For each fold:
   - Train the model on $k-1$ folds.
   - Test on the remaining fold.
   - Generate feature importance via SHAP or LIME.
3. Collect k sets of feature importances across folds.

This captures the variability of feature importance across different splits.

0.6 Step 2: Define Key Metrics

Generalizability. Are top features consistently ranked across folds?

$$\text{Generalizability} = \text{Correlation}\big(\text{feature ranks across folds}\big), \quad \in [0,1].$$

Concordance. Does the explanation agree with the model's own internal logic (e.g., Gini importance)?

$$\text{Concordance} = \rho(I_{\text{Model}}, I_{\text{XAI}})$$

where ρ is Pearson correlation, I_{Model} is model-based feature importance, and I_{XAI} is XAI-based importance.

Stability. Do similar patients receive similar explanations?

$$\text{Stability} = \frac{1}{N}\sum_{k=1}^{N} d(S_{k1}, S_{k2}),$$

where $d(\cdot,\cdot)$ is the Euclidean distance between explanations for two similar samples. Lower = more stable.

0.7 Step 3: Global Reliability Score

We combine the three metrics:

$$\text{Reliability} = \text{Concordance} \times \text{Generalizability} \times (1 - \text{Stability})$$

Reliability $\in [0, 1]$, with values closer to 1 indicating highly reliable explanations.

Proposed Method

1 Framework

The proposed framework for reliability analysis of Explainable AI (XAI) on endocrinology diseases is outlined below.

1.1 Dataset

- Pima Indians Diabetes Dataset (PIDD) from Kaggle.

1.2 Data Preprocessing

(a) Replace medically impossible zeros (e.g., Glucose = 0) with NaN.

(b) Impute missing values using median imputation.

(c) Standardize features using z-score normalization.

1.3 Model Training

(a) Multi-Layer Perceptron (MLP) with `BCEWithLogitsLoss`, which also incorporates `pos_weight` for handling class imbalance:

$$\ell(z, y) = -\Big[\texttt{pos\_weight} \cdot y \cdot \log(\sigma(z)) + (1 - y) \cdot \log(1 - \sigma(z))\Big]$$

(b) Early stopping to prevent overfitting.

(c) Adam optimizer for adaptive learning.

1.4 Explainable AI (XAI) with SHAP

- Use SHAP (DeepExplainer) to compute feature attribution scores for each patient.

1.5 Gradient-based Importance

(a) Start from Binary Cross-Entropy loss:

$$\ell(\hat{p}, y) = -\Big[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})\Big]$$

(b) Compute gradient w.r.t input features:

$$g = |\nabla_x \, \ell(f_\theta(x), y)|$$

(c) Normalize per sample:

$$\tilde{g}_j = \frac{g_j}{\sum_k g_k + \epsilon}$$

1.6 Gradient–Loss Reliability (GLR) Score

(A) Compute Spearman rank correlation between SHAP importances S and gradient importances G.

(B) Convert values to ranks:

$$r_j^u = \text{rank of } u_j, \quad r_j^v = \text{rank of } v_j$$

(C) Spearman correlation:

$$\rho = 1 - \frac{6 \sum_{j=1}^{d} (r_j^u - r_j^v)^2}{d(d^2 - 1)}$$

(D) Average ρ across all validation samples to obtain the GLR score.

1.7 ε-Perturbation Test

(A) Perturb the top features identified by SHAP slightly: $x_j \to x_j + \varepsilon$.

(B) Measure the increase in loss:

$$\Delta\ell_j = \ell(f_\theta(x + \varepsilon e_j), y) - \ell(f_\theta(x), y)$$

(C) Check if SHAP's top-k features overlap with features having highest $\Delta\ell_j$.

(D) Define Hit@k as:

$$\text{Hit@k} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\text{Top-}k_{\text{SHAP}} \cap \text{Top-}k_{\text{Perturb}} \neq \emptyset\}$$

1.8 Visualization

The following diagrams are used to analyze and interpret the framework:

- SHAP Beeswarm plots (global feature importance).
- GLR histograms (distribution of Spearman ρ).
- Calibration curves (probability calibration analysis).

2 Dataset and Problem Setting

We use the Pima Indians Diabetes dataset (768 patients, 8 features). The task is binary classification: predict diabetes presence. Positives: 268 / 768. Features were standardized; impossible zeros in certain physiological variables were imputed with medians.

Pima Indians Diabetes Dataset (Kaggle)

3 Model and Training Controls

Architecture. A small MLP with two hidden layers (ReLU) and a single logit output.

Early stopping & LR scheduling. We monitor validation AUC and apply early stopping with ReduceLROnPlateau to stop training when performance plateaus, a standard practice to avoid overfitting and improve convergence. :contentReference[oaicite:0]index=0

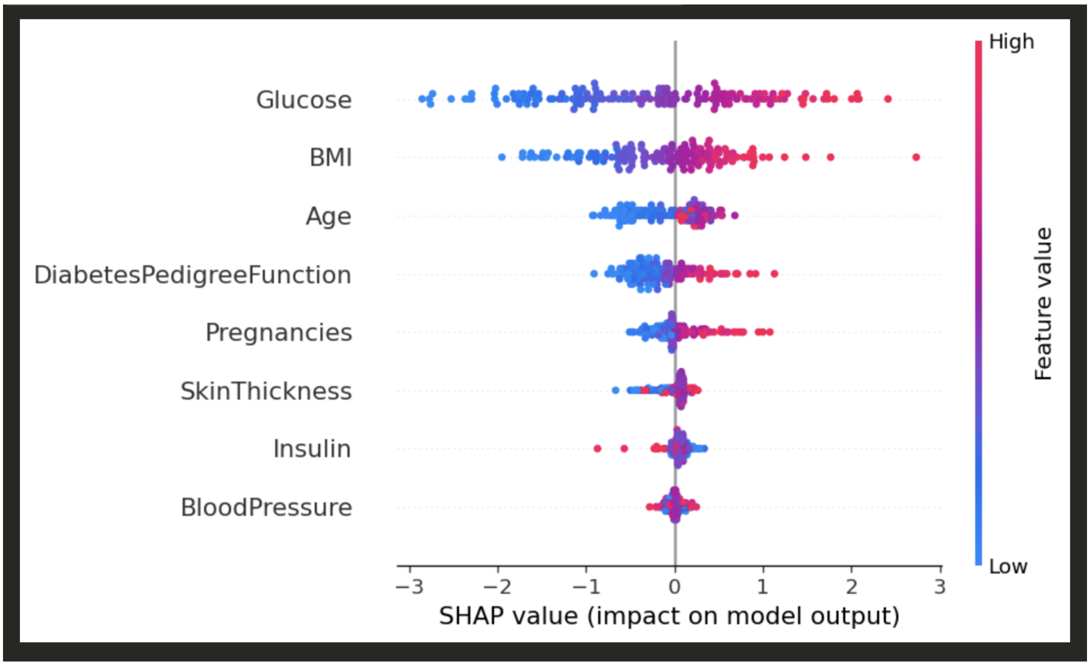

Figure 1: SHAP Beeswarm plot showing global feature importance.

Class imbalance. We use `BCEWithLogitsLoss` with `pos_weight` $= \frac{N_{\text{neg}}}{N_{\text{pos}}}$ to penalize positive-class errors appropriately in imbalanced datasets.

BCEWithLogitsLoss and Class Imbalance

The standard Binary Cross-Entropy (BCE) loss is defined on predicted probabilities $\hat{p}$ as:

$$\ell(\hat{p}, y) \;=\; -\Big[\, y \cdot \log(\hat{p}) \;+\; (1-y) \cdot \log(1-\hat{p}) \,\Big],$$

where $y \in \{0,1\}$ is the true label and $\hat{p}$ is the predicted probability.

Since most neural networks output a *logit* $z \in \mathbb{R}$, the probability is obtained through the sigmoid function:

$$\hat{p} = \sigma(z) = \frac{1}{1+e^{-z}}.$$

Substituting this into the BCE loss gives the **logit-based BCE**:

$$\ell(z, y) = -\Big[\, y \cdot \log\big(\sigma(z)\big) \;+\; (1-y) \cdot \log\big(1-\sigma(z)\big) \,\Big].$$

—

Handling Class Imbalance with `pos_weight`

In imbalanced datasets (e.g., far fewer positive cases than negatives), we can up-weight the contribution of positive samples. This is done by multiplying the positive term by a factor `pos_weight` > 1.

The modified loss becomes:

$$\ell(z, y) = -\Big[\, \texttt{pos\_weight} \cdot y \cdot \log\big(\sigma(z)\big) \;+\; (1-y) \cdot \log\big(1-\sigma(z)\big) \,\Big].$$

—

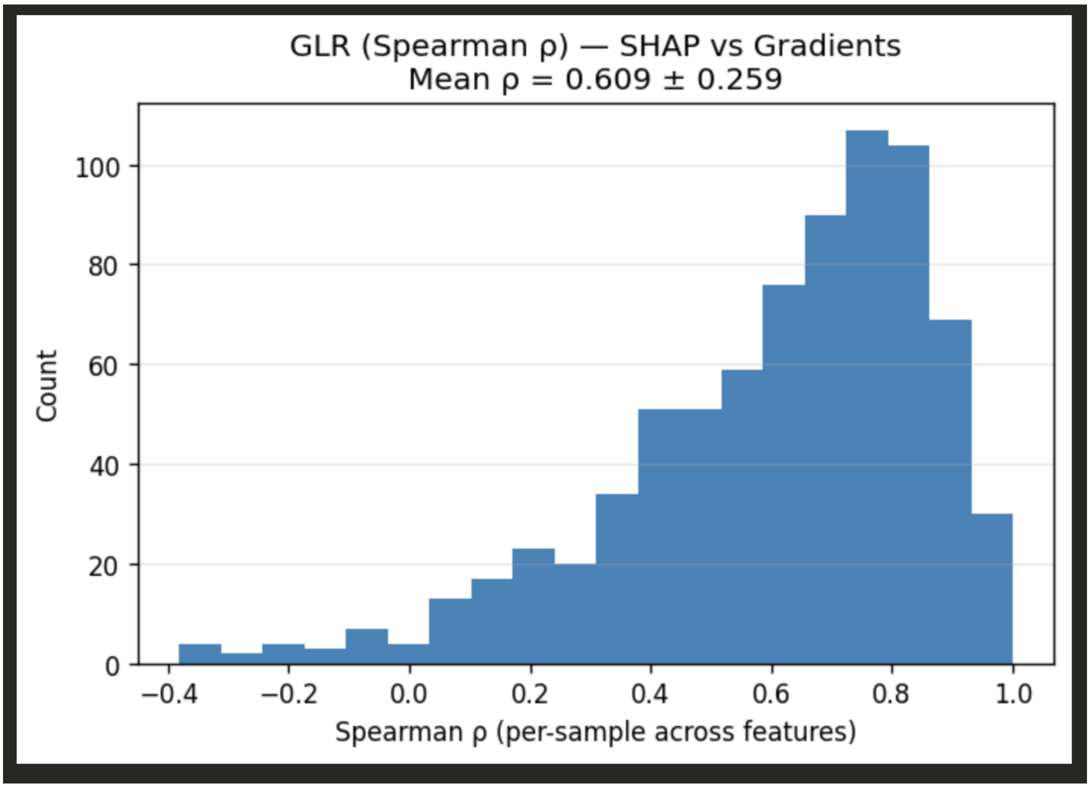

Figure 2: GLR histogram showing Spearman ρ distribution.

Interpretation

- If pos_weight $= 1$, we recover the standard BCE loss.
- If pos_weight > 1, positive examples are penalized more when misclassified, helping the model pay more attention to the minority class.
- This adjustment is crucial in medical datasets (e.g., diabetes diagnosis), where positive cases (disease) are much rarer than negative cases (healthy).

4 Evaluation: Discrimination, Thresholding, Calibration

AUC & Accuracy. We report AUC and accuracy at the default 0.5 threshold and at the threshold maximizing Youden's J.

Youden's J. $J = \text{sensitivity} + \text{specificity} - 1$. Selecting the ROC operating point that maximizes J balances sensitivity and specificity and often outperforms the naive 0.5 cut-off in imbalanced problems.

Calibration & Brier. We store reliability-curve (calibration) arrays and compute the Brier score, a strictly proper scoring rule for probabilistic predictions, and use calibration_curve to form reliability diagrams.

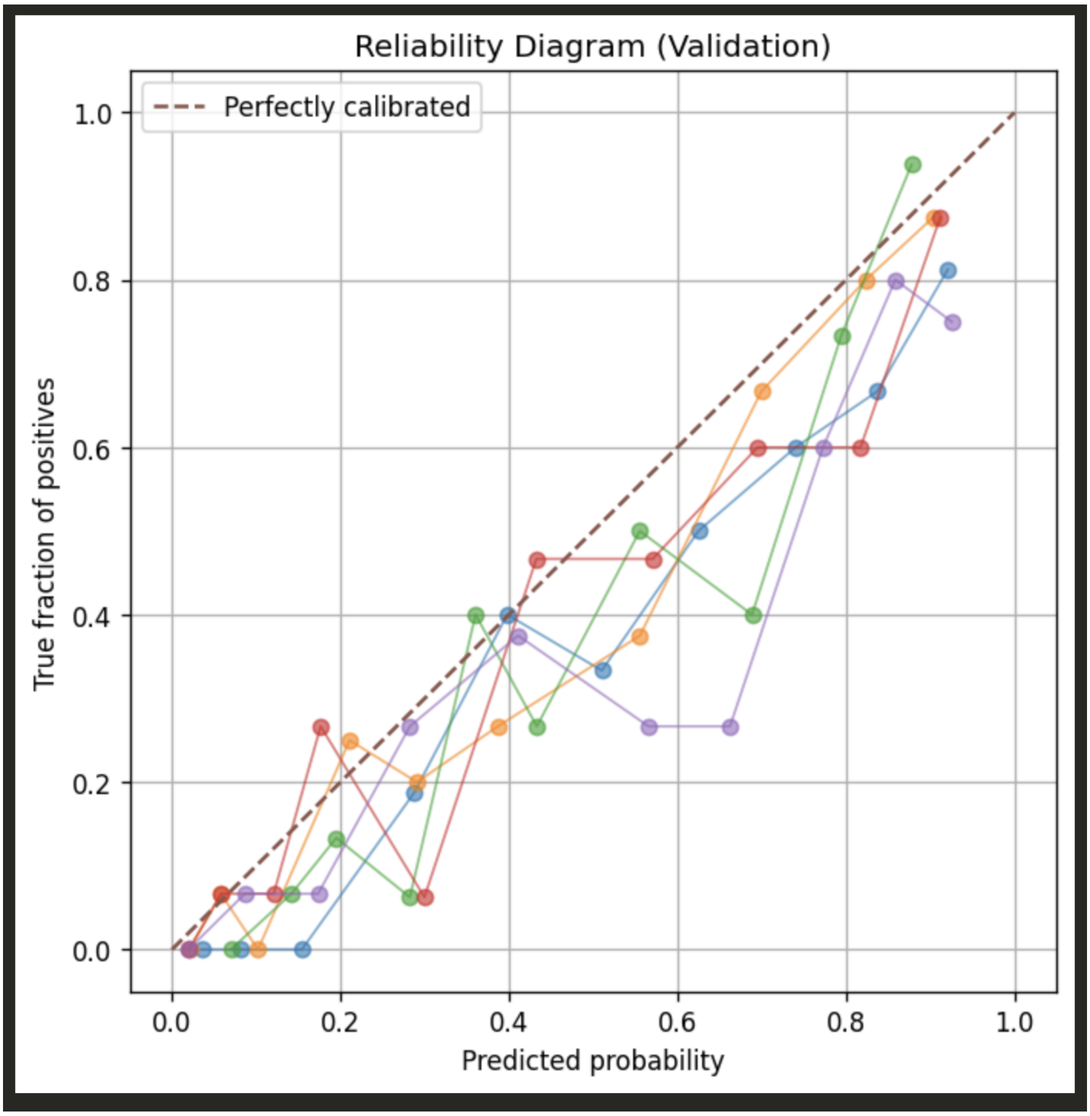

Figure 3: Calibration curve comparing predicted probabilities vs. true fractions.

5 Explanation Methods

5.1 SHAP with K-means Background

We use SHAP (Deep/Kernel variants as applicable) with a K-means summarized background ($K = 50$) to reduce variance and runtime while preserving representativeness of the training distribution, as recommended in SHAP docs.

5.2 Gradient-based Importances - Step-by-step Mathematics

Model setup: A neural net classifier $f_\theta(x)$ outputs a logit z. The probability is

$$p = \sigma(z) = \frac{1}{1 + e^{-z}}.$$

True label: $y \in \{0, 1\}$.

Loss function (Binary Cross-Entropy with logits):

$$\ell(z, y) = \log(1 + e^{z}) - yz.$$

Gradient w.r.t. logit:

$$\frac{\partial \ell}{\partial z} = \sigma(z) - y = p - y.$$

Chain rule: gradient w.r.t. input features:

$$\nabla_x \ell(f_\theta(x), y) = (p - y)\, \nabla_x f_\theta(x).$$

Importance vector (absolute values):

$$g = \big|\nabla_x \ell(f_\theta(x), y)\big|.$$

Normalize (per sample):

$$\tilde{g}_j = \frac{g_j}{\sum_k g_k + \varepsilon}.$$

Thus, each sample has a distribution $\tilde{g}$ over feature importance.

—

Simple Example

Suppose a linear model with 2 features:

$$f_\theta(x) = w_1 x_1 + w_2 x_2,$$

where $w_1 = 2$, $w_2 = 1$. Input $x = (3, 1)$ and true label $y = 1$.

Step 1: Logit

$$z = f(x) = 2 \cdot 3 + 1 \cdot 1 = 7.$$

Step 2: Probability

$$p = \sigma(z) \approx \sigma(7) \approx 0.999.$$

Step 3: Gradient w.r.t. logit

$$p - y = 0.999 - 1 \approx -0.001.$$

Step 4: Gradient w.r.t. features

$$\nabla_x f(x) = (w_1, w_2) = (2, 1),$$

$$\nabla_x \ell = (p - y) \cdot (2, 1) = -0.001 \cdot (2, 1) = (-0.002, -0.001).$$

Step 5: Absolute values

$$g = (0.002, 0.001).$$

Step 6: Normalize

$$\tilde{g}_1 = \frac{0.002}{0.003} \approx 0.67, \quad \tilde{g}_2 = \frac{0.001}{0.003} \approx 0.33.$$

—

Hence, feature 1 has about 67% importance and feature 2 about 33% for this sample.

6 Reliability Criteria and Metrics

6.1 Gradient-Loss Reliability (GLR)

Let $\mathbf{s}_i \in \mathbb{R}^d$ be per-feature *normalized* importance from an explainer for validation sample i, and $\mathbf{g}_i \in \mathbb{R}^d$ be the *normalized* magnitude of the input-loss gradients $|\partial \mathcal{L} / \partial \mathbf{x}|$. We compute the per-sample Spearman rank correlation

$$\rho_i \;=\; \mathrm{Spearman}(\mathbf{s}_i, \mathbf{g}_i),$$

and report GLR $= \frac{1}{n} \sum_{i=1}^{n} \rho_i \in [-1, 1]$. Higher is better: it indicates the explainer ranks features similarly to how the model's loss locally reacts to them.

What is Spearman correlation?

Spearman correlation (Spearman's ρ) is a *rank correlation*: it measures how well two variables agree in ordering, not in scale.

For feature importance, this matters because SHAP values and gradient magnitudes can be on very different numeric scales, but their *ordering* (which feature is most important) should match if SHAP is reliable.

—

Formula

For vectors

$$u = (u_1, \dots, u_d), \quad v = (v_1, \dots, v_d),$$

we first convert them into ranks:

$$r_j^u = \text{rank of } u_j, \qquad r_j^v = \text{rank of } v_j.$$

Then compute the **Pearson correlation of the ranks**:

$$\rho = \frac{\sum_{j=1}^{d} \left(r_j^u - \bar{r}^u\right) \left(r_j^v - \bar{r}^v\right)}{\sqrt{\sum_{j=1}^{d} \left(r_j^u - \bar{r}^u\right)^2} \sqrt{\sum_{j=1}^{d} \left(r_j^v - \bar{r}^v\right)^2}}.$$

Alternative closed form (when no ties):

$$\rho = 1 - \frac{6 \sum_{j=1}^{d} (r_j^u - r_j^v)^2}{d(d^2 - 1)}.$$

—

Interpretation

- $\rho \approx 1$: SHAP and gradients rank features almost identically $\Rightarrow$ explanations are consistent.
- $\rho \approx 0$: No correlation $\Rightarrow$ SHAP does not agree with gradient sensitivity.
- $\rho < 0$: Opposite rankings $\Rightarrow$ SHAP contradicts gradient signal.

Thus, the **Gradient–Loss Reliability (GLR)** is defined as the average Spearman ρ across all validation samples, giving a summary reliability score.

6.2 ε-Perturbation Sanity Test

For each validation sample, we nudge each standardized feature by a small ε in the *loss-increasing* gradient direction, recompute the loss, and identify the feature causing the largest increase. If this top feature coincides with the explainer's top-ranked feature (top-1), we count a "hit". The hit-rate@1 over the validation set summarizes *action-consistency*: explanations should indicate features whose small changes most affect model loss.

What is ε-Perturbation Sanity Check ?

Explainability methods such as SHAP highlight important features for a given prediction. The ε-Perturbation test asks: *"If I actually nudge the features SHAP considers important by a small amount ε, does the model's loss increase as expected?"*

If the features highlighted by SHAP overlap with those that truly increase the loss when perturbed, then the explanation is considered **reliable**.

—

Mathematical Formulation

For a sample (x_i, y_i) with d input features:

1. **Base loss:**
$$L_0 = \ell\big(f_\theta(x_i),\ y_i\big).$$

2. **Gradient sign:**
$$\mathrm{sign}_j = \mathrm{sign}\left(\frac{\partial L_0}{\partial x_{ij}}\right), \qquad j = 1, \dots, d.$$

3. **Perturb features one at a time:**
$$x_i^{(j)} = x_i \ + \ \varepsilon \cdot \mathrm{sign}_j \cdot e_j,$$
where e_j is the j-th unit basis vector.
The perturbed loss is
$$L_j = \ell\big(f_\theta(x_i^{(j)}),\ y_i\big).$$

4. **Delta-loss vector:**
$$\Delta_j = L_j - L_0.$$

5. **Top-k features by delta-loss:**
$$T_i^{\Delta} = \mathrm{TopK}\left(\Delta_1, \dots, \Delta_d\right).$$

6. **Top-k features by SHAP:**
$$T_i^{\mathrm{SHAP}} = \mathrm{TopK}\left(|s_{i1}|, \dots, |s_{id}|\right),$$
where s_{ij} is the SHAP attribution for feature j.

7. **Hit definition:**
$$\mathrm{hit}_i = \begin{cases} 1, & \text{if } T_i^{\Delta} \cap T_i^{\mathrm{SHAP}} \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

8. **Overall Hit@k score:**
$$\mathrm{Hit@k} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{hit}_i.$$

—

Interpretation

- High Hit@k: SHAP's top-ranked features match those that cause the largest loss increase under perturbation ⇒ explanations are **faithful**.
- Low Hit@k: SHAP highlights features that do not significantly affect the loss ⇒ explanations are **unreliable**.

7 Experimental Setup

Cross-validation. 5-fold Stratified CV; each fold saves artifacts: model state, scaler, SHAP background/values, gradient arrays, calibration arrays, and summaries.

Background for SHAP. We form a K-means summary ($K = 50$) from the training fold to approximate the background distribution efficiently.

8 Results

From your run:

Discrimination and Thresholding

- **AUC**: 0.842 ± 0.015
- **ACC@0.5**: 0.755 ± 0.039
- **ACC@Youden's** J^*: 0.772 ± 0.042
- **Fold-wise Brier**: ≈ 0.14–0.19 (stored for reliability plots)

The improvement from ACC@0.5 to ACC@J^* confirms the value of ROC-based threshold optimization under class imbalance. :contentReference[oaicite:8]index=8

Reliability of Explanations

- **GLR (SHAP vs. gradients)**: 0.631 ± 0.037
- ε**-hit@1**: 0.604 ± 0.087

IG aligns slightly better with gradients than SHAP (consistent with its axiomatic gradient path construction), while SmoothGrad nearly mirrors gradients (as expected for a gradient-based method). :contentReference[oaicite:9]index=9