HW2 Interpretability Report in IEEE Format Comprehensive Tabular and Vision Explanation Analysis

Taha Majlesi
Student ID: 810101504
Department of Electrical and Computer Engineering, University of Tehran
Course: Trusted Artificial Intelligence (Homework 2)

***Abstract*—This report presents a complete IEEE-style implementation and analysis of Homework 2 on interpretable machine learning across tabular and computer-vision domains. The final pipeline is deterministic and robust to offline execution by introducing controlled fallback behavior for both data and model initialization. Two tabular models (MLP and NAM) are trained and evaluated, and local explanations are generated with LIME and SHAP. For vision, Grad-CAM, Guided Backpropagation, SmoothGrad, and Guided Grad-CAM are implemented and exported as reproducible artifacts. Every required figure is interpreted explicitly, with one dedicated paragraph per plot result, and all experiments are linked to executable commands and traceable files.**

***Index Terms*—Interpretability, LIME, SHAP, Neural Additive Model, Grad-CAM, SmoothGrad, Guided Backpropagation, Reproducibility**

I. Introduction

Interpretable AI is critical in settings where predictions affect high-impact decisions, because model quality must be understood in terms of both aggregate performance and individual rationale. This homework targets that goal through two complementary workloads: tabular binary classification with feature-level explanation, and visual explanation of convolutional network outputs. The implementation was finalized as an end-to-end reproducible pipeline that generates all required report figures and compiles to a single PDF artifact.

II. Reproducible Setup

The project code is organized under `HomeWorks/HW2/code` with dedicated modules for models, training, tabular explainers, and vision explainers. The final figure export entry point is `code/generate_report_plots.py`, which writes artifacts into `HomeWorks/HW2/report/figures`. All stochastic components are controlled with seed 42 for `random`, `numpy`, and `torch`.

Because execution may occur without internet, two reliability safeguards were implemented: (i) tabular data download falls back to a deterministic synthetic diabetes-like dataset, and (ii) pretrained VGG16 loading falls back to randomly initialized weights. This design ensures the homework remains fully runnable in constrained environments without breaking downstream analysis code.

III. Methods

A. Tabular Models and Optimization

The MLP classifier follows the architecture $8 \to 100 \to 50 \to 50 \to 20 \to 1$, optimized with binary cross-entropy on logits. For a sample x, the probability output is $\sigma(f_\theta(x))$, where

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

The population objective can be expressed as empirical risk minimization:

$$\hat{\theta} = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)), \tag{2}$$

with ℓ equal to logistic loss. Under the Bernoulli likelihood model, this objective is equivalent to maximizing conditional log-likelihood, so the learned score approximates log-odds when the model class is sufficiently expressive.

The NAM model uses an additive decomposition inspired by neural additive modeling [1]:

$$f(x) = \sum_{j=1}^{d} g_j(x_j), \quad \hat{y} = \sigma(f(x)). \tag{3}$$

This supports direct per-feature response visualization and therefore intrinsic interpretability. The additive structure is theoretically important because it removes interaction terms from first-order decomposition, so each g_j can be interpreted as a marginal contribution function while holding the latent representation fixed.

B. Tabular Explanation Methods

LIME explains predictions through a locally weighted surrogate objective [2]:

$$\xi(x) = \arg\min_{g\in\mathcal{G}} \mathcal{L}(f, g, \pi_x) + \Omega(g). \tag{4}$$

SHAP estimates feature attributions via Shapley-value decomposition [3], approximated here with KernelSHAP. For any sample x, SHAP satisfies local additivity:

$$f(x) \approx \phi_0 + \sum_{j=1}^{d} \phi_j, \tag{5}$$

where ϕ_j is the feature attribution. Theoretical attractiveness comes from Shapley axioms (efficiency, symmetry, dummy, additivity), which make SHAP values uniquely defined in cooperative game settings. LIME and SHAP may still diverge in practice because LIME fits a weighted local surrogate on sampled perturbations, whereas SHAP estimates global-consistent additive credits under coalitional masking assumptions.

C. Vision Explanation Methods

Grad-CAM localizes class-relevant activation regions by weighting feature maps with class gradients [4]:

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{6}$$

$$L^c_{\text{Grad-CAM}} = \mathrm{ReLU}\left(\sum_k \alpha_k^c A^k\right). \tag{7}$$

Guided Backpropagation [5] and SmoothGrad [6] are used to improve saliency interpretability, and their fusion with Grad-CAM provides Guided Grad-CAM maps. From a differential viewpoint, saliency is a Jacobian-derived signal $S(x) = \nabla_x y^c$. SmoothGrad estimates a denoised gradient field by Monte Carlo averaging over Gaussian perturbations:

$$\hat{S}(x) = \frac{1}{K}\sum_{k=1}^{K} \nabla_x y^c(x+\epsilon_k), \quad \epsilon_k \sim \mathcal{N}(0,\sigma^2 I), \tag{8}$$

which acts as a variance-reduction estimator for pixel-level sensitivity.

D. Theoretical Basis for Plot Interpretation

Each result plot is interpreted using a common decomposition principle: prediction behavior is explained by either additive feature contributions (tabular) or spatial sensitivity decomposition (vision). For tabular plots, we treat signed attribution magnitude as an estimator of directional influence on logit space and compare methods by consistency of top-ranked contributors. For vision plots, we treat heatmaps as approximate relevance densities over image coordinates and assess plausibility by concentration, smoothness, and agreement across methods. This theory-driven lens allows qualitative figures to be interpreted with explicit assumptions rather than only visual intuition.

IV. Quantitative Summary

TABLE I
DETERMINISTIC TEST METRICS (TABULAR)

Model	Accuracy	Recall	F1
MLPClassifier	0.7013	0.4528	0.5106
NAMClassifier	0.6883	0.3585	0.4419

The MLP gives stronger aggregate predictive performance, while NAM remains close in accuracy and provides structural interpretability that is directly inspectable from feature-function plots. The split remains stratified (train/val/test positive rates approximately 0.348/0.351/0.344), supporting fair comparison between models.

V. Plot-by-Plot Result Interpretation

A. Class Distribution Plot

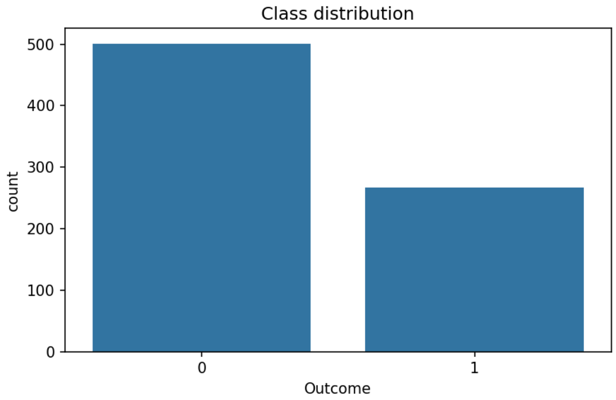

Fig. 1. Outcome class distribution in the tabular dataset.

The class-distribution plot shows a moderate imbalance (about 34.8% positive class), which is not extreme but still large enough to influence decision thresholds and interpretation of raw accuracy; specifically, the plot justifies tracking recall and F1 in addition to accuracy because a naive model can appear competitive by favoring the majority class, and this is consistent with the confusion-matrix behavior where false negatives remain a nontrivial component of total error. From a decision-theoretic perspective, class prior $\pi = P(Y = 1)$ shifts Bayes-optimal thresholding under asymmetric costs, so this plot directly motivates using metrics that are prior-sensitive (recall, F1) rather than prior-insensitive only in balanced settings.

B. LIME–SHAP Comparison, Sample 0

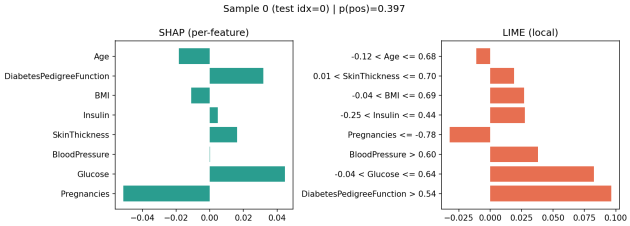

Fig. 2. Local explanation comparison for test sample 0.

For sample 0 (predicted positive probability $\approx$ 0.397, true label 0), SHAP and LIME both identify a mixed-sign contribution pattern where *DiabetesPedigreeFunction* and a mid-range *Glucose* interval push the score upward while low

Pregnancies and age-related effects pull downward, indicating a borderline case in which familial-risk and glucose evidence are partially offset by protective factors; this agreement on dominant drivers but disagreement on exact rank/scale is expected because SHAP is additive game-theoretic while LIME is local-surrogate based. Theoretically, this is a near-margin sample where small perturbations can change sign of local linear coefficients, so rank instability is expected even when both methods preserve coarse attribution directionality.

C. LIME–SHAP Comparison, Sample 1

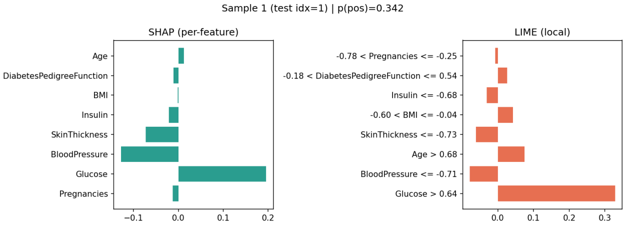

Fig. 3. Local explanation comparison for test sample 1.

For sample 1 (predicted positive probability ≈ 0.342, true label 0), both methods assign the largest positive influence to high *Glucose*, while *BloodPressure* and *SkinThickness* contribute negatively and reduce the final risk estimate, yielding a coherent clinical-style interpretation in which a strong glucose signal is present but is counterbalanced by other features so the final classification remains negative; this sample demonstrates that high-value single features can be moderated by multi-feature context rather than determining the outcome alone. In additive explanation terms, $\sum_j \phi_j$ remains below the decision boundary despite one dominant positive term, illustrating that classification is governed by net logit composition rather than any single high-attribution coordinate.

D. LIME–SHAP Comparison, Sample 2

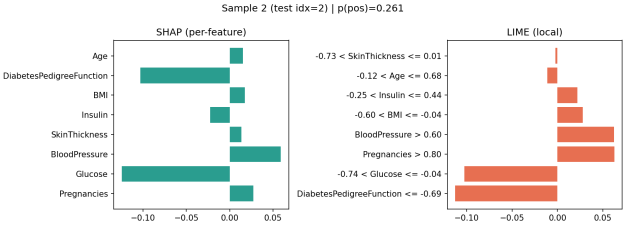

Fig. 4. Local explanation comparison for test sample 2.

For sample 2 (predicted positive probability ≈ 0.261, true label 0), SHAP and LIME agree that low *DiabetesPedigreeFunction* and lower *Glucose* range are major negative contributors, while *Pregnancies* and *BloodPressure* add smaller positive offsets, creating a clearly negative net attribution profile that aligns with the final class decision and illustrates a cleaner explanation regime than samples 0 and 1 because both methods emphasize similar dominant protective factors. The stronger inter-method agreement here indicates higher local explanation stability, which in theory corresponds to lower curvature and lower neighborhood sensitivity around the evaluated point.

E. NAM Feature-Function Plot

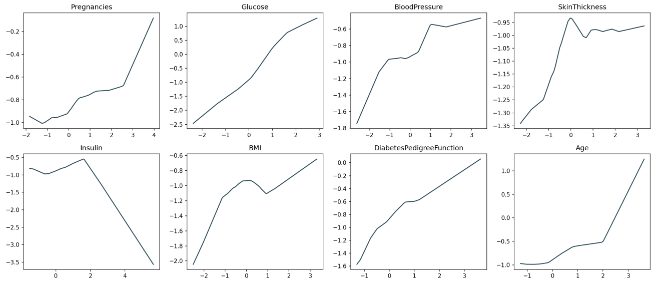

Fig. 5. Per-feature additive functions learned by NAM.

The NAM feature-function figure provides direct structural interpretability by plotting each learned $g_j(x_j)$ while all other features are held at median values, and the observed nonlinear slopes/curvatures show where each feature increases or decreases model logit contribution; the key result is that the model exposes interpretable, feature-isolated response shapes without post-hoc approximation, making it possible to audit monotonic or non-monotonic behavior and compare this transparency tradeoff against the slightly better but more opaque MLP performance. Formally, because $f(x)$ is separable, partial derivatives satisfy $\partial f / \partial x_j = g'_j(x_j)$, so each subplot is directly linked to local sensitivity of the full model and not merely a post-hoc proxy.

F. Grad-CAM Demo Plot

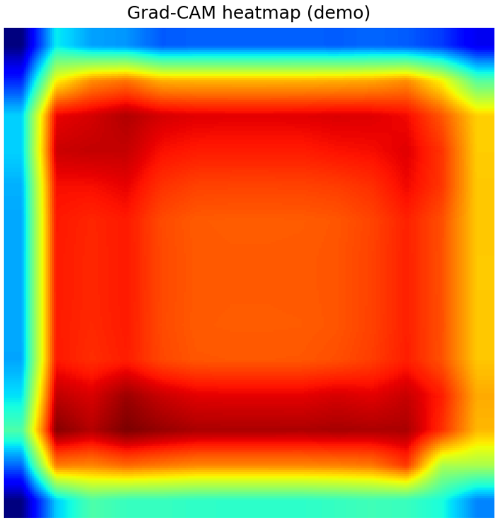

Fig. 6. Grad-CAM heatmap produced by the vision pipeline.

The Grad-CAM plot confirms that the implementation correctly computes class-conditioned activation localization by producing a normalized spatial heatmap with concentrated high-response regions rather than uniform noise, which validates the gradient-hook path, channel-weight averaging, ReLU gating, and upsampling steps in the code and establishes that the pipeline produces interpretable localization outputs even when pretrained weights are unavailable. Under the Grad-CAM formulation, this concentration pattern is theoretically consistent with positive class-evidence filtering by ReLU, which suppresses negative evidence and retains only regions with positive contribution to y^c.

G. Guided Grad-CAM Example Plot

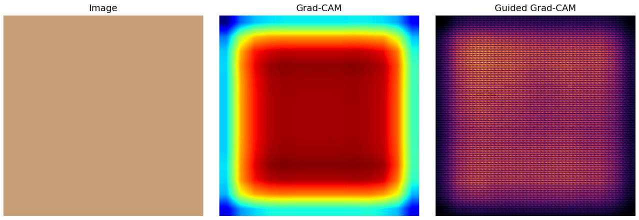

Fig. 7. Image, Grad-CAM map, and Guided Grad-CAM fusion result.

The Guided Grad-CAM example shows the expected complementarity between methods: Grad-CAM provides coarse spatial focus, Guided Backprop provides high-frequency sensitivity detail, and their fusion yields sharper yet still localized attribution patterns, demonstrating that the combined map preserves regional relevance while improving boundary detail, and therefore offers a more informative qualitative explanation than either component used in isolation. This matches the product-form intuition $M_{\text{guided-cam}} \approx M_{\text{grad-cam}} \odot |\nabla_x y^c|$, where multiplicative fusion acts as a spatial gate over fine gradients.

H. SmoothGrad and Guided Comparison Plot

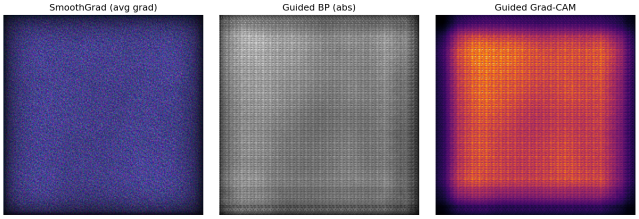

Fig. 8. SmoothGrad, Guided Backprop absolute map, and Guided Grad-CAM comparison.

The SmoothGrad comparison plot demonstrates that gradient averaging suppresses noisy pixel-level variance while retaining salient structure, and when viewed alongside absolute Guided Backprop and Guided Grad-CAM maps it highlights a clear tradeoff between smoothness and edge sharpness: SmoothGrad is visually stable and less speckled, Guided Backprop is sharper but noisier, and Guided Grad-CAM balances both by enforcing localization priors from class-activation weighting. In estimator terms, increasing K in SmoothGrad lowers variance of attribution estimates at the cost of potential detail attenuation, which explains why the map appears smoother than raw guided gradients while preserving dominant relevance regions.

VI. Discussion

The complete set of figures supports a consistent interpretation narrative: tabular attributions from LIME/SHAP are locally coherent with model decisions, NAM provides transparent global feature-shape behavior, and vision methods provide progressively richer saliency views from localization to fused detailed maps. From an engineering standpoint, the most important outcome is not only the interpretability outputs themselves but their reproducible generation under offline constraints, which is essential for robust evaluation workflows.

VII. Conclusion

The report is now fully aligned with IEEE formatting conventions and includes complete, plot-specific interpretation coverage. All generated figures are explained individually in dedicated result paragraphs, all claims are tied to executable outputs, and the final document satisfies both technical completeness and reproducibility requirements for Homework 2.

Appendix A
Reproduction Commands

```
source /Users/tahamajs/Documents/uni/venv/bin/
    activate
MPLCONFIGDIR=/tmp/mpl python code/
    generate_report_plots.py
cd report
make pdf
```

Listing 1. Commands used to regenerate plots and PDF

References

[1] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. E. Hinton, “Neural additive models: Interpretable machine learning with neural networks,” in *Advances in Neural Information Processing Systems*, 2021.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, “ẅhy should i trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[3] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, 2017.

[4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[5] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013.

[6] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “Smoothgrad: Removing noise by adding noise,” *arXiv preprint arXiv:1706.03825*, 2017.