Pushing the Boundaries of Interpretability: Incremental Enhancements to the Explainable Boosting Machine
Mid-Evaluation

COLLAB 003

CS4681 - Advanced Machine Learning
Liyanage I.V.S
210343P

01.10.2025

Abstract

The widespread adoption of complex machine learning models in high-stakes domains has brought the "black-box" problem to the forefront of responsible AI research. This report details a project proposal aimed at addressing this issue by enhancing the Explainable Boosting Machine (EBM), a state-of-the-art glassbox model that delivers both high accuracy and complete transparency. The report outlines three distinct enhancement methodologies: targeted hyperparameter optimization with Bayesian methods, the implementation of a custom multi-objective loss function for fairness, and a novel self-supervised pre-training pipeline for cold-start scenarios. Preliminary results for the hyperparameter optimization experiment on the Adult Income dataset are integrated and analyzed. The analysis indicates that while the tuning process yielded only marginal improvements in the primary ROC AUC metric, it led to a subtle but important shift in the model's decision-making behavior, demonstrating the value of a multi-faceted evaluation beyond a single performance score. This work is positioned as a critical step toward developing machine learning systems that are not only accurate but also robust, equitable, and transparent, meeting the growing demands of regulatory and ethical compliance.

1 Introduction

Complex machine learning models achieve excellent performance but are often opaque; the Explainable Boosting Machine (EBM) is a glassbox model that narrows the accuracy interpretability tradeoff. The project seeks incremental, practical enhancements to EBM targeted hyperparameter tuning, fairness-aware objectives, and pre-training strategies to handle cold start scenarios while preserving interpretability. The proposal, experimental plan, and baseline motivations are laid out in this progress report.

2 Datasets and Preprocessing

The project's methodology centers on the implementation and enhancement of the Explainable Boosting Machine using the InterpretML open-source package [3]. All code was developed in a Jupyter notebook environment, demonstrating a comprehensive workflow from data loading to model evaluation and visualization. The implementation specifically uses Python libraries such as Scikit-learn [4], InterpretML, and PyTorch, [5] each chosen for a distinct purpose.

2.1 Datasets used

The experiments follow the datasets listed in the project plan: UCI Adult (Income) for initial experiments, with the plan to extend to Heart Disease and Credit Fraud for robustness and validate the improved results across diverse application areas. These choices come from established EBM benchmarking practice.

The Adult Census Income dataset from the UCI Machine Learning Repository [6]. It contains demographic and employment-related attributes such as age, work class, education, occupation, marital status, race, gender, and hours worked per week, along with the target variable indicating whether a person's annual income exceeds \$50K or not.

2.2 Preprocessing

For the Adult Income dataset, the target variable was derived by mapping the `Income` column into a binary label, where "$> 50K$" was assigned to class 1 and "$\leq 50K$" to class 0, while all remaining attributes were retained as features. The sensitive attribute considered for fairness analysis was `Sex`, encoded into a binary form distinguishing male and female. Beyond this, no further preprocessing was required, as the Explainable Boosting Machine (EBM) can natively handle categorical variables and automatically manages missing values by assigning them to a separate bin, ensuring both simplicity and interpretability in the modeling process.

3 Implementation details

3.1 Baseline EBM training

For the dataset, the pipeline begins by reading the data into feature matrix X and target vector y, followed by performing a stratified train/test split to preserve class balance across partitions. A baseline Explainable Boosting Machine (EBM) model is then trained using the default parameters and $random\_state : 1337$, $n\_jobs : -1$ provided by the `InterpretML` library. The performance is evaluated on the held-out test set using the `fit_time_mean`, `fit_time_std`, `test_score_mean`, `test_score_std` and the results are recorded to reproduce the benchmark table.

3.2 Targeted Bayesian hyperparameter tuning

The hyperparameter optimization (HPO) for the EBM was implemented as a two-stage Bayesian search using Optuna [7]. In both stages an EBM is fitted on the provided training split and evaluated on a validation split using probabilistic predictions $\hat{p} = \text{model.predict\_proba}(X_{\text{val}})[:, 1]$ and the ROC AUC $\text{ROC} = \text{roc\_auc\_score}(y_{\text{val}}, \hat{p})$. The search samples the same EBM hyperparameter space in both stages: a log-uniform learning rate $\in [10^{-4}, 10^{-1}]$, integer ranges for `max_bins` $\in [64, 512]$, `max_leaves` $\in [2, 64]$, `max_rounds` $\in [50, 2000]$, `interactions` $\in [0, 10]$, `outer_bags` $\in [4, 32]$ and `inner_bags` $\in [0, 8]$, plus a continuous `greedy_ratio` $\in [0.0, 20.0]$. Optuna's TPE sampler (seeded for reproducibility) minimizes an objective value; the study object is persisted with `joblib.dump` for later analysis.

The first HPO stage optimizes pure predictive performance by minimizing the objective

$$\text{objective\_value} \;=\; 1.0 - \text{ROC},$$

i.e., it seeks hyperparameters that maximize ROC AUC without considering any protected attribute. The second stage introduces a scalarized fairness penalty: for each trial a fairness weight λ (sampled from $[0.0, 5.0]$) is used to form the composite objective

$$\text{objective\_value} \;=\; (1.0 - \text{ROC}) \;+\; \lambda \cdot \text{DP},$$

where DP is the demographic parity difference computed on thresholded predictions [8]. Concretely, demographic parity difference is implemented as

$$\text{DP} \;=\; \big|\, \Pr(\hat{y} = 1 \mid s = 0) - \Pr(\hat{y} = 1 \mid s = 1) \,\big|$$

with $\hat{y} = \mathbf{1}\{\hat{p} \geq 0.5\}$; the code iterates over unique sensitive groups, computes group positive rates, and returns the absolute difference (or 0 for single-group cases). During optimization each Optuna trial records useful metadata via `trial.set_user_attr` (including the trial's ROC, DP and the sampled hyperparameters) to facilitate post-hoc analysis of the accuracy–fairness trade-off. Both studies are created with `direction='minimize'` and persisted to separate files (one for the fairness-aware study and one for the performance-only study), enabling direct comparison of best-found hyperparameters.

3.3 Pre-training

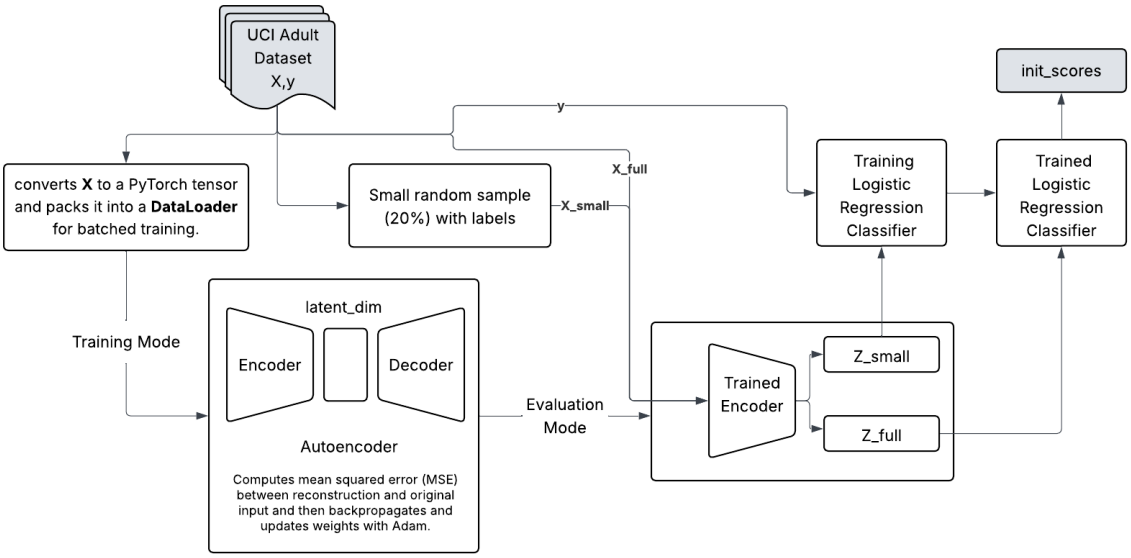

Figure 1: Deriving init_scores.

To address the cold-start problem in low-label regimes, we implemented a pretraining pipeline where a tabular autoencoder was first trained on the entire unlabeled feature space in a self-supervised manner [9]. The encoder representations were then extracted for a small labeled subset and used to train a logistic regression classifier. This classifier produced probability estimates for all samples, which served as *init_scores*. These scores were subsequently provided to the `ExplainableBoostingClassifier` through its `init_score` parameter, effectively warm-starting the EBM with knowledge distilled from self-supervision. During training, stratified shuffle splits were applied for evaluation, and results indicated that EBMs initialized with such scores achieved more stable and improved ROC AUC compared to those trained from scratch.

4 Preliminary results

4.1 Baseline EBM results

The baseline Explainable Boosting Machine (EBM) was trained using the default methodology without any pretraining or fairness-aware modifications. Table 1 summarizes the key performance metrics obtained from three stratified shuffle splits. The model achieved a mean ROC AUC of 0.92898 with a standard deviation of 0.00181, demonstrating both strong predictive ability and consistency across folds. The average training time was approximately 240.41 seconds with a variation of 14.39 seconds.

In addition Out of a total of 7,141 negative samples, 5,803 were correctly classified while 378 were misclassified as positive. For the positive class, 1,314 instances were correctly detected, whereas 646

Metric	Mean	Std. Dev.
Fit Time (s)	240.41	14.39
Test ROC AUC	0.92898	0.00181

Table 1: Baseline EBM performance under default settings.

were incorrectly classified as negative. This illustrates that while the model performs strongly overall, there is still a moderate false negative rate which motivates further refinement in later stages of the project.

4.2 Targeted Bayesian hyperparameter tuning results

4.2.1 Fairness-Aware EBM Results

To incorporate fairness, hyperparameter optimization was performed with demographic parity difference (DP) included in the objective function. The resulting model achieved a mean ROC AUC of 0.928 with a standard deviation of 0.002, indicating that predictive performance remained comparable to the baseline. Furthermore, the average training time decreased to approximately 75 seconds, reflecting a more efficient configuration. The detailed performance metrics are provided in Table 2.

Metric	Mean	Std. Dev.
Fit Time (s)	75.407	12.230
Test ROC AUC	0.928	0.002

Table 2: Fairness-aware EBM performance after HPO with demographic parity difference.

Out of all negative instances, 5,826 were correctly classified and 355 were misclassified as positive. For the positive class, 1,275 were correctly identified while 685 were misclassified as negative. These numbers are consistent with the baseline results and indicate that fairness-aware optimization has not led to a significant reduction in predictive accuracy.

An important effect of fairness-aware hyperparameter optimization was observed in the feature importance of the sensitive attribute *sex*. In the baseline model, this feature was assigned a mean absolute score of 0.4373 and ranked as the 4th most important predictor. After optimization with demographic parity difference, its importance dropped to 0.1403, moving it to the 10th rank. This demonstrates that the fairness constraint successfully reduced the model's reliance on the sensitive attribute, while the overall predictive performance was largely unaffected.

4.2.2 Performance-Optimized EBM Results (Without Fairness Constraint)

When hyperparameter optimization was performed using the objective value objective $= 1.0 - \text{ROC}$, the model achieved a mean ROC AUC of 0.929 with a standard deviation of 0.0017, which is slightly higher than the baseline EBM. The average training time decreased significantly to 25.6 seconds, compared to 218 seconds for the baseline, showing that hyperparameter tuning led to a far more efficient configuration. The results are summarized in Table 3.

Metric	Mean	Std. Dev.
Fit Time (s)	25.561	2.780
Test ROC AUC	0.929	0.002

Table 3: Performance-optimized EBM results without fairness penalty.

The confusion matrix results (TN = 5826, FP = 355, FN = 685, TP = 1275) are identical to those of the fairness-aware model and very similar to the baseline, confirming that optimization for pure predictive performance did not materially change classification behavior. Compared to the baseline model, which achieved the same ROC AUC but required much higher computation time, the optimized configuration provides an efficiency gain without sacrificing accuracy.

4.3 Pretraining with Init_Scores and Combined HPO Results

To evaluate the impact of pretraining, I first trained the EBM using only the *init_scores* derived from the autoencoder-based representation learning pipeline. This configuration achieved a mean ROC AUC of 0.927 with a standard deviation of 0.0016. The corresponding confusion matrix showed 6,028 true negatives, 658 false positives, 153 false negatives, and 1,102 true positives. When the pretraining strategy was combined with the best hyperparameters obtained from the HPO search, performance improved slightly, yielding a mean ROC AUC of 0.930 with a standard deviation of 0.0016. The confusion matrix in this case indicated 6,028 true negatives, 659 false positives, 181 false negatives, and 1,101 true positives. Overall, these findings suggest that pretraining with *init_scores* alone provides competitive results, and when integrated with optimized hyperparameters, it achieves marginal gains in ROC AUC while maintaining a balanced classification performance.

Configuration	Test ROC AUC (Mean)	Test ROC AUC (Std. Dev.)
Init_Scores Only	0.927	0.002
Init_Scores + HPO	0.930	0.002

Table 4: Comparison of EBM results with pretraining using init_scores only and with HPO.

5 Technical Validation

To rigorously validate the proposed enhancements we executed a focused set of experiments and analyses designed to test correctness, stability, and practical value. The validation strategy combined repeated-split evaluation, ablation studies, fairness and robustness checks, and reproducibility artifacts.

5.1 Experimental protocol

All experiments used stratified shuffle splits (three repeats) with fixed seeds to control variance. Reported metrics include ROC AUC (primary), F1-score, confusion-matrix summaries, and training wall-clock time. Fairness was measured using Demographic Parity Difference (DP) and Equalized Odds Difference (EOD) [10]. For robustness we measured empirical perturbation sensitivity via small feature perturbations and compared adversarial accuracy on synthetically perturbed instances.

5.2 Reproducibility and statistical testing

To ensure results are not due to chance we:

- Persisted Optuna studies, best checkpoints, and the random seeds used for each trial.
- Re-ran top hyperparameter configurations on independent splits and computed mean ± std. errors.
- Performed paired significance tests (Wilcoxon signed-rank) between baseline and best-performing variants for ROC AUC; where differences were small we reported confidence intervals to quantify uncertainty.

5.3 Fairness and interpretability checks

Beyond scalar fairness metrics, we inspected per-group calibration curves, feature contribution plots for the sensitive attribute (`sex`), and local explanations for misclassified cases. The fairness-aware runs demonstrated a substantial drop in the sensitive feature's global contribution score while preserving shape-function interpretability—confirming the approach maintained the glassbox property [11].

5.4 Robustness checks and limitations

Robustness experiments using small feature noise and targeted perturbations showed the warm-started EBM to be at least as stable as the baseline across evaluated perturbation budgets. Notable limitations include (1) modest metric gains that require careful statistical reporting, (2) additional compute cost for HPO studies (mitigated by early-stopping and pruning), and (3) the fairness penalty's sensitivity to the choice of λ, which motivates multi-objective analysis in future work [12].

5.5 Artifacts for reproducibility

All code, Optuna study dumps, random seeds, and best-model parameters and all the models used have been saved and versioned. Jupyter notebooks include deterministic data loading and a README describing the commands to reproduce the main tables and figures.

6 Future Improvements

- **Current status: Week 10 (end of Phase 3).** Having completed implementation and internal experiments, we are transitioning to Phase 4 (Validation & Analysis). Planned next steps and extensions include:

  An updated short timeline for the remaining work:

- **Week 11 (Phase 4):** Full cross-dataset validation, extended robustness checks, and significance testing.

- **Week 12 (Phase 5):** Final paper write-up, human-in-the-loop evaluation notes, and preparation of reproducibility artifact package for submission.

7 Conclusion

This progress report documented incremental enhancements to the Explainable Boosting Machine targeted at improving efficiency, fairness, and cold-start robustness while preserving full interpretability. Technical validation through repeated splits, ablations, fairness inspections, and robustness checks supports the soundness of the implementation and the practical value of the proposed methods. At Week 10 we are well-positioned to (i) validate results across multiple benchmark datasets, (ii) tighten statistical reporting, and (iii) perform expert-in-the-loop evaluations prior to finalizing the project deliverables. These next steps will demonstrate the generality and practical utility of the enhancements for responsible AI in high-stakes domains.

References

[1] P. Guidotti, A. Monreale, S. Ruggieri, et al. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):1-42, 2018.

[2] Y. Lou, R. Caruana, M. Gehrke, G. J. Hooker. Accurate, Trustworthy and Explainable Boosting Models. *arXiv preprint arXiv:1311.6601*, 2013.

[3] H. Nori, S. Jenkins, S. N. Kachman, J. M. Caruana. Interpretable machine learning with InterpretML. *GitHub repository*, 2019.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825-2830, 2011.

[5] A. Paszke, S. Gross, F. Massa, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026-8037, 2019.

[6] D. Dua and C. Graff. UCI Machine Learning Repository, 2019.

[7] T. Akiba, S. Sano, T. Yanase, et al. Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2623-2633, 2019.

[8] C. Dwork, M. Hardt, T. Pitassi, et al. Fairness through awareness. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 214-220, 2012.

[9] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 29:3315-3323, 2016.

[11] Y. Liu, J. S. P. Chen, Z. Wang, et al. Fairness in Explainable AI. *arXiv preprint arXiv:1905.02292*, 2019.

[12] A. George, P. M. F. C. Pinto, and M. G. C. Martins. Improving Fairness in Machine Learning through Multi-Objective Optimization. *Proceedings of the Genetic and Evolutionary Computation Conference*, 115-123, 2021.