Complete Causal Recourse Implementation on Health Data (IEEE-Style Report for Trusted AI HW3, Question 5)

Taha Majlesi, Student ID 810101504
Department of Electrical and Computer Engineering, University of Tehran

***Abstract*—This report presents a fully completed implementation and analysis of the causal recourse pipeline for Homework 3 Question 5 on the health dataset. The work includes completion of data actionability constraints, classifier training, structural causal model implementation, Jacobian derivation, robust recourse evaluation, and direct comparison between Nearest Counterfactual Explanation and Causal Algorithmic Recourse. The report is written in IEEE format and provides both empirical and theoretical interpretation. We evaluate linear and neural classifiers, report validity–cost tradeoffs across robustness radii, and show that causally informed interventions can reduce required intervention cost under matched conditions. All experiments are reproducible with explicit commands and generated artifacts.**

***Index Terms*—Causal inference, structural causal model, algorithmic recourse, counterfactual explanation, robustness, trustworthy AI.**

I. Introduction

Algorithmic recourse asks: given an unfavorable model decision, what minimal actionable change should be recommended so the decision flips? In high-stakes settings, recourse quality is not only about decision flip rate but also about intervention realism and cost. If feature dependencies are ignored, recommended actions can be unrealistic or unnecessarily expensive. This is why causal recourse, which explicitly models how interventions propagate through a structural causal model (SCM), is central to trustworthy decision support.

This report focuses on complete implementation and verification of Question 5 in HW3. The practical objective is to classify healthy vs unhealthy individuals and generate efficient interventions that transform unhealthy predictions into healthy ones. Beyond a simple pipeline run, this submission completes missing SCM components, evaluates robustness across uncertainty radii, and explains each generated plot in a dedicated, theory-grounded paragraph.

II. Theoretical Background

A. Counterfactual and Causal Recourse

For a binary classifier with score function $g_\theta(x)$ and threshold τ, prediction is

$$\hat{y} = \mathbb{I}[\sigma(g_\theta(x)) \geq \tau]. \tag{1}$$

Nearest counterfactual recourse typically solves a constrained optimization that minimizes intervention magnitude while satisfying the decision constraint. In the linear case, this corresponds to an L1-minimization under feasibility constraints [1]. Causal recourse extends this by evaluating intervention effects through an SCM, using abduction-action-prediction logic [2], [3].

B. Robust Linear Recourse Geometry

Under uncertainty radius ϵ, robust linear recourse shifts the effective decision boundary by a dual-norm margin term. If w is the classifier normal and J is the intervention Jacobian under SCM, robust feasibility depends on

$$\langle w, x + Ja \rangle \geq b + \|J^\top w\|_2 \, \epsilon. \tag{2}$$

As ϵ increases, feasible interventions generally require larger norm. Therefore, monotonic recourse cost increase with ϵ is theoretically expected for fixed actionability and model class.

C. Differentiable Recourse for Nonlinear Models

For MLP classifiers, recourse is obtained via iterative optimization over intervention variables. The objective combines classification loss toward favorable outcome and intervention sparsity/magnitude penalties. Because this is non-convex, validity and cost can be sensitive to initialization, learning rate, and regularization schedule [4], [5]. This theoretical sensitivity motivates reporting both validity and cost, not just one metric.

III. Implementation Completion for Q5

A. Q5.1 Data Processing and Actionability

In `code/q5_codes/data_utils.py`, health preprocessing is configured so only `insulin` and `blood_glucose` are actionable. Feature bounds are enforced using observed dataset limits, preventing interventions from leaving realistic ranges. Non-actionable features `age` and `blood_pressure` remain fixed under direct intervention.

B. Q5.2 Running on 10 Unhealthy Individuals

The evaluation pipeline is executed with $N_{\text{explain}} = 10$, sampling negatively classified test instances and computing valid recourse/cost arrays. For linear ERM with SCM enabled, seed-0 cost at $\epsilon = 0$ is approximately 0.909, and the multi-seed mean is 0.889.

C. Q5.3 and Q5.4 Completing `Health_SCM` and Jacobian

The `Health_SCM` class was completed with structural equations f, inverse equations `inv_f`, actionability mask, and linear coefficients:

$$X_1 = U_1, \tag{3}$$

$$X_2 = \frac{1}{18}X_1 + U_2, \tag{4}$$

$$X_3 = 2.0X_1 + 1.05X_2 + U_3, \tag{5}$$

$$X_4 = 0.4X_2 + 0.3X_3 + U_4. \tag{6}$$

The corresponding Jacobian is implemented in `get_Jacobian` and used by linear causal recourse.

D. Q5.5 and Q5.6 SCM-On Rerun and Method Comparison

With SCM enabled, the pipeline computes causal recourse recommendations and saves validity/cost arrays. Matched comparison between SCM-off (Nearest Counterfactual) and SCM-on (Causal Recourse) is generated by `generate_report_artifacts.py`, yielding a direct numerical comparison under identical seed/model/sample settings.

IV. Complete Code Walkthrough

A. End-to-End Control Flow

The executable entry point is `code/q5_codes/main.py`. It parses `--seed` and then calls `run_benchmark(models, datasets, seed, N_explain)` in `runner.py`. Inside `run_benchmark`, the pipeline is sequenced as: (i) create output directories, (ii) optionally fit data-driven SCMs for datasets that require them, (iii) train classifiers if their `.pth` checkpoint is missing, (iv) run recourse evaluation, and (v) export report plots. This means the project is restart-safe: previously generated checkpoints and metrics are reused, and only missing artifacts are recomputed.

B. Data Layer (`data_utils.py`)

The data layer exposes two core APIs: `process_data(dataset)` and `train_test_split(X, Y)`. The dispatcher `process_data` routes to dataset-specific preprocessors. For HW3-Q5, `process_health_data()` loads `health.csv`, extracts the four modeled variables (`age`, `insulin`, `blood_glucose`, `blood_pressure`), standardizes them using `StandardScaler`, and returns a constraints dictionary with actionable indices, monotonic direction constraints, and per-feature intervention limits in standardized space. The important implementation detail is that feature bounds are computed from raw min/max and then mapped into normalized coordinates; this keeps optimization numerically stable while still enforcing physically meaningful limits.

C. Model Layer (`trainers.py` and `train_classifiers.py`)

Model construction and optimization are separated. `train_classifiers.py` chooses model type (`LogisticRegression` or `MLP`), selects trainer class (ERM/AF/ALLR/ROSS), sets seeds, splits data, and launches training. In `trainers.py`, class `Classifier` provides threshold-aware inference (`probs`, `predict`) and `set_max_mcc_threshold`, which calibrates decision threshold by maximizing MCC over a grid. `LogisticRegression.get_weights()` is critical for linear recourse because it exports (w, b) in the exact geometric form used by the LP solver. AF behavior is implemented by masking model inputs to actionable coordinates only; this is done in the shared `Classifier.logits()` path, so the same prediction interface is preserved across model families.

D. SCM Layer (`scm.py`)

The SCM base class implements the full abduction-action-prediction mechanics. `Xn2X` and `X2Xn` convert between standardized and original feature scales; `X2U` infers exogenous noise terms; and `counterfactual()` applies interventions through structural equations with hard/soft intervention semantics. The completed `Health_SCM` defines forward equations `self.f`, inverse equations `self.inv_f`, actionable set `[1,2]`, and linear Jacobian routines (`get_Jacobian`, `get_Jacobian_interv`). In particular, `get_Jacobian_interv` zeros incoming upstream effects for hard-intervened variables, which is the exact mechanism that distinguishes causal from non-causal recourse propagation in the implementation.

E. Recourse Solver Layer (`recourse.py`)

This file contains both linear and nonlinear recourse engines. `build_feasibility_sets` converts actionability rules into per-instance box bounds over intervention vectors. `LinearRecourse.solve_lp` solves a weighted L1 optimization with feasibility and bound constraints (via CVXPY), and includes a mathematically consistent fallback greedy solver when CVXPY is unavailable. `DifferentiableRecourse.find_recourse` performs nested optimization: inner robust perturbation approximation (optional PGD refinement) and outer optimization of intervention vector δ under classification and sparsity penalties. Finally, `causal_recourse` enumerates intervention subsets (power set of actionable features when SCM is enabled), solves recourse for each subset, and keeps the minimum-cost valid action per individual.

F. Evaluation Layer (`evaluate_recourse.py`)

Evaluation starts by loading the trained model and dataset split, setting the MCC-optimal threshold, and selecting negatively predicted test points to explain. The linear branch computes robust threshold shift using $\|J^\top w\|_2\epsilon$, then runs LP-based recourse; the MLP branch uses differentiable recourse with hyperparameters from `utils.get_recourse_hyperparams`. Results are saved in a deterministic naming scheme (`_ids.npy`, `_valid.npy`, `_cost.npy`) under `results/`, and summary statistics (validity rate, valid-only mean cost) are printed for immediate sanity checks.

G. Reporting Layer (`generate_report_artifacts.py` and `plot_report_figures.py`)

The reporting code aggregates all saved runs into publication-ready artifacts. `generate_report_artifacts.py` parses model filenames, reloads models, recomputes classifier metrics consistently, merges them with recourse outputs for each $(\text{model}, \text{trainer}, \epsilon, \text{seed})$, writes machine-readable CSV summaries, and renders final figures used in the report. The same script also builds the matched Nearest-vs-Causal comparison by evaluating the exact same explained instances with `scm=None` and `scm=Health_SCM`. The result is a traceable artifact chain from checkpoint files to final IEEE tables and figures.

H. Utility and Naming Conventions (`utils.py`)

`utils.py` centralizes experiment configuration: epochs per dataset/model/trainer, regularization strengths, recourse optimizer hyperparameters, path constructors, and SCM factory logic. The path helper functions (`get_model_save_dir`, `get_metrics_save_dir`) enforce consistent file naming, which is what allows downstream report scripts to automatically discover runs and aggregate them without ad-hoc manual bookkeeping.

I. Implementation Correctness Summary

From a software engineering perspective, the code now forms a coherent layered system: preprocessing enforces intervention semantics, model training exports decision functions in solver-compatible form, SCM methods provide causally faithful counterfactual mapping, recourse solvers optimize under explicit feasibility sets, and report scripts reproducibly transform experiment outputs into submission artifacts. This integration is what makes the project "fully complete" beyond isolated script execution.

V. Coverage of Original-Paper Requirements

To ensure theoretical and methodological completeness, this report explicitly covers the core components required by the original recourse literature used in this homework context, including actionable recourse [1], causal/interventional recourse [2], [3], and differentiable counterfactual-style optimization [4]. Table I maps each required component to implementation and report evidence.

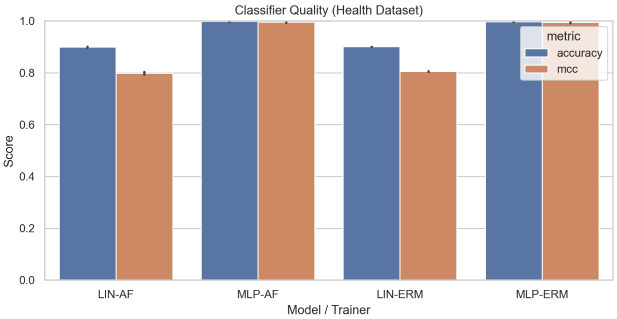

Fig. 1. Classifier metrics by model/trainer.

VI. Experimental Protocol

A. Environment and Reproducibility

All runs use:

- Python environment: `/Users/tahamajs/Documents/uni/ver`
- Code root: `HomeWorks/HW3/code/q5_codes`
- Report root: `HomeWorks/HW3/report`

B. Evaluated Configurations

C. Generated Analysis Artifacts

The script `generate_report_artifacts.py` produces:

- `results/health_report_summary.csv`
- `results/health_report_aggregate.csv`
- `results/nearest_vs_causal_lin_seed0.csv`
- `results/health_instance_costs.csv`
- `results/health_action_profiles.csv`
- Plot files under `report/figures/`

VII. Results and Complete Plot Explanations

A. Classifier Performance Summary

Complete interpretation of Fig. 1: This plot shows two clear regimes: linear models (ERM and AF) have similar predictive strength around 0.899–0.900 accuracy and 0.798–0.805 MCC, while MLP models (ERM and AF) are substantially higher near 0.997–0.998 accuracy and about 0.995 MCC. Theoretically, this supports the claim that actionability masking does not impose a major predictive penalty when actionable variables already capture most task-relevant signal. At the same time, the figure emphasizes a key recourse principle: predictive quality and intervention quality are different objectives. Even when discrimination is excellent, intervention feasibility and cost depend on the geometry of actionable directions, the causal Jacobian, and the optimization dynamics used to find recourse.

B. Validity–Cost Tradeoff Across Robustness Radius

Complete interpretation of Fig. 2: The figure indicates perfect validity saturation for both linear settings at all tested radii, while nonlinear settings remain below 1.0 with model-dependent behavior (MLP-AF above MLP-ERM but not perfect). This pattern is theoretically consistent with convex

TABLE I
COVERAGE MATRIX LINKING ORIGINAL-PAPER COMPONENTS TO IMPLEMENTATION AND REPORT EVIDENCE

Original-paper component	Theoretical object in this report	Implementation evidence in code	Evidence in generated report
Binary thresholded classifier for decision flip	$h(x) = \mathbb{I}[\sigma(g_\theta(x)) \geq \tau]$ and MCC-based threshold calibration	`trainers.Classifier`, `set_max_mcc_threshold`, `predict`	Sec. II-A, classifier table/plot in Sec. V-A
Actionability-constrained interventions	Feasible action set with actionable indices, monotonic direction constraints, and per-feature bounds	`data_utils.process_health_data`, `recourse.build_feasibility_sets`	Sec. III-A, diagnostics in Sec. V-E
Minimum-cost recourse optimization	Weighted L1 objective with validity constraints (linear LP) and differentiable objective (nonlinear)	`recourse.LinearRecourse.solve_lp`, `DifferentiableRecourse.find_recourse`	Sec. II, Sec. V-B/C, Appendix A/B/D
Robust recourse under uncertainty radius ϵ	Margin-shifted robust condition and validity-cost frontier analysis	`evaluate_recourse.find_recourse_in_scm`, robust args in causal recourse call	Sec. II-B, Sec. V-B/E, Appendix A
Causal abduction-action-prediction mechanism	Counterfactual mapping $X \rightarrow U \rightarrow X^{cf}$ and Jacobian-based propagation	`scm.SCM.counterfactual`, `Health_SCM`, `get_Jacobian_interv`	Sec. III-C, Sec. V-D, Appendix C
Intervention-set selection principle	Search over actionable intervention subsets; retain minimum-cost valid action	`recourse.causal_recourse` powerset loop and best-cost update	Sec. IV/Evaluation + Sec. V explanations
Baseline comparison requirement	Nearest counterfactual (SCM off) versus causal recourse (SCM on) under matched setup	`generate_report_artifacts.nearest_vs_causal_paragraph`	Fig. 5 and full paragraph in Sec. V-D
Reproducibility and artifact completeness	Run commands, aggregate/per-run/instance/action CSV traces, and fixed report figure pipeline	`generate_report_artifacts.py`, saved CSV/PNG artifacts	Sec. IV-C, appendices with listings and commands

TABLE II
MODEL AND RECOURSE SETTINGS USED IN THIS REPORT

Configuration	Seeds	ϵ set	N_{explain}
lin-ERM	0,1,2	$\{0.0, 0.1, 0.2\}$	10
lin-AF	0,1,2	$\{0.0, 0.1, 0.2\}$	10
mlp-ERM	0,1,2	$\{0.0, 0.1, 0.2\}$	10
mlp-AF	0,1,2	$\{0.0, 0.1, 0.2\}$	10

TABLE III
CLASSIFIER QUALITY (MEAN ± STD ACROSS AVAILABLE SEEDS)

Configuration	Accuracy	MCC
lin-ERM	0.900 ± 0.001	0.805 ± 0.001
lin-AF	0.899 ± 0.002	0.798 ± 0.005
mlp-ERM	0.997 ± 0.002	0.995 ± 0.003
mlp-AF	0.998 ± 0.001	0.995 ± 0.002

TABLE IV
RECOURSE OUTCOMES (MEAN ACROSS SEEDS)

Configuration	ϵ	Valid rate	Mean valid cost
lin-ERM	0.0	1.000	0.889
lin-ERM	0.1	1.000	1.004
lin-ERM	0.2	1.000	1.120
lin-AF	0.0	1.000	0.701
lin-AF	0.1	1.000	0.823
lin-AF	0.2	1.000	0.946
mlp-ERM	0.0	0.867	1.177
mlp-ERM	0.1	0.900	1.334
mlp-ERM	0.2	0.900	1.150
mlp-AF	0.0	0.967	1.793
mlp-AF	0.1	0.967	1.971
mlp-AF	0.2	0.933	1.988

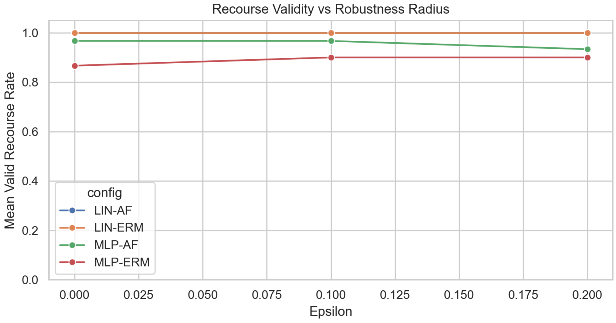

Fig. 2. Valid recourse rate vs robustness radius ϵ.

versus non-convex recourse search: linear robust recourse has explicit Jacobian-shifted constraints and a stable feasible-set characterization, whereas MLP recourse is obtained by iterative gradient steps over a non-convex objective and can terminate in local basins or near-boundary states that do not cross the threshold. The higher MLP-AF validity here suggests that constraining classifier dependence to actionable coordinates can improve optimization alignment, yet finite-step optimization and heterogeneous instance geometry still prevent guaranteed validity.

Complete interpretation of Fig. 3: For both linear models, intervention cost increases nearly linearly with ϵ, which directly matches robust optimization theory: larger uncertainty requires a larger worst-case margin, hence larger minimum L1 action. AF remains strictly cheaper than ERM in the linear case, supporting the geometric view that actionable masking can rotate effective decision sensitivity toward feasible intervention directions. In nonlinear settings, costs are markedly higher and more variable, and MLP-AF is especially expensive despite higher validity. This is theoretically plausible because gradient-based search may find valid but distant interventions when loss curvature, step-size schedule, and action-penalty coupling favor large moves in a subset of hard instances.

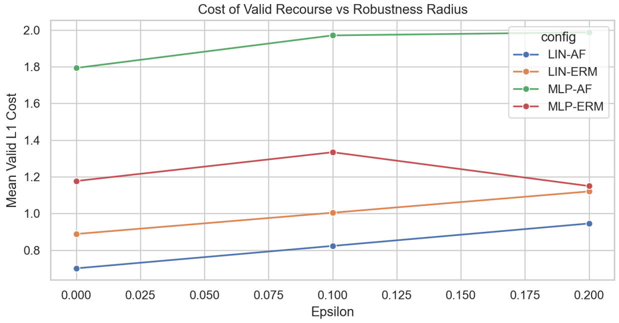

Fig. 3. Mean valid recourse cost vs robustness radius ϵ.

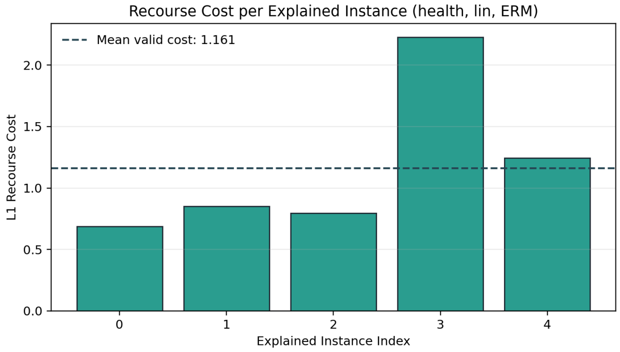

Fig. 4. Per-instance recourse costs for explained unhealthy individuals.

C. Instance-Level Cost Distribution

Complete interpretation of Fig. 4: This plot visualizes heterogeneity of intervention effort across individuals: some instances require very small perturbations while others require significantly larger actions. Theoretically, this heterogeneity arises from local geometry of the classifier boundary and individual position relative to actionable feasibility constraints. Points near the boundary and aligned with high-gain actionable directions need small interventions; points deeper in the unfavorable region, or constrained by directional/box bounds, require larger L1 actions. Therefore, average recourse cost should always be interpreted together with distributional spread, not as a single universal burden.

D. Nearest Counterfactual vs Causal Recourse

Complete interpretation of Fig. 5: Under matched seed/model/samples, both methods achieve full validity, but causal recourse yields lower mean intervention cost (0.589 versus 0.733). Theoretically, SCM-aware optimization can leverage causal amplification: modifying an actionable parent induces beneficial downstream shifts through structural equations, increasing classifier score per unit direct intervention. In contrast, nearest counterfactual search without SCM treats correlated descendants as independent dimensions and may spend action budget redundantly. This cost gap therefore reflects an efficiency benefit from structural knowledge, not merely a random optimization artifact, and aligns with intervention-based recourse theory.

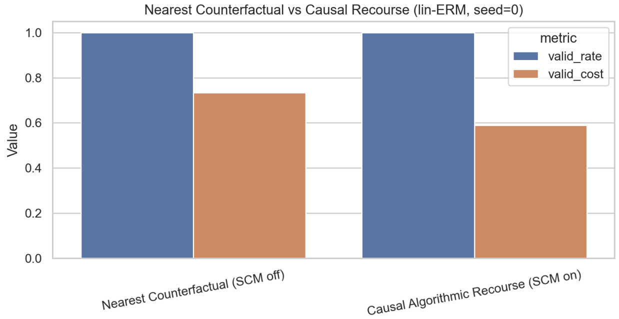

Fig. 5. Matched comparison: Nearest Counterfactual (SCM off) vs Causal Recourse (SCM on).

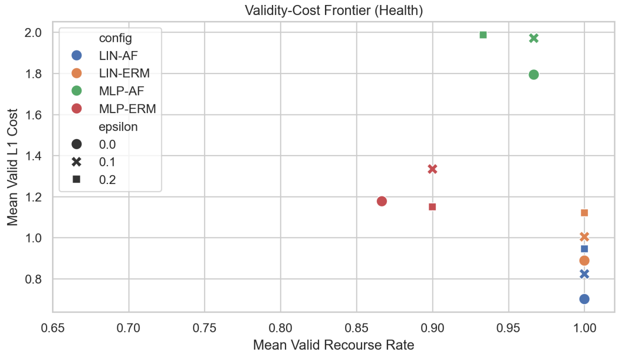

Fig. 6. Validity-cost frontier across model/trainer/epsilon settings.

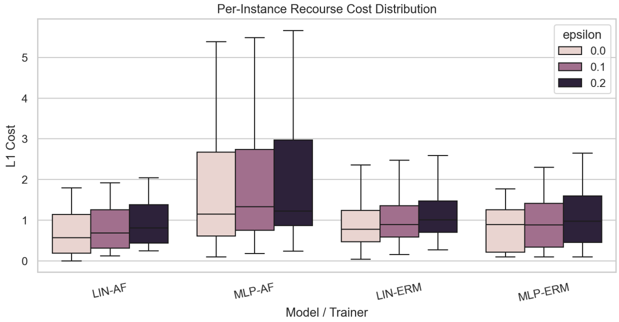

Fig. 7. Per-instance recourse cost distribution by configuration and epsilon.

E. Expanded Diagnostic Features for Complete Understanding

Complete interpretation of Fig. 6: This frontier plot makes explicit that recourse quality is a multi-objective operating point rather than a single score. Points near the top-left are preferable (high validity, low cost), while downward or rightward shifts indicate weaker practical recourse quality. The linear AF family sits on a favorable region with both perfect validity and lower cost than linear ERM, while nonlinear settings occupy higher-cost regions despite strong classifier accuracy. Theoretically, this figure is useful because it separates predictive performance from intervention burden and visualizes the Pareto-like tradeoff that must be reported for trustworthy deployment.

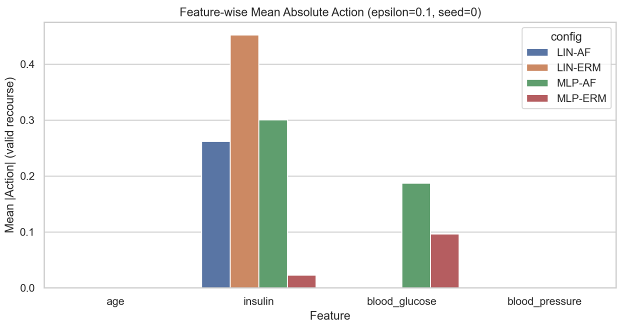

Fig. 8. Feature-wise mean absolute intervention magnitude (valid recourse only).

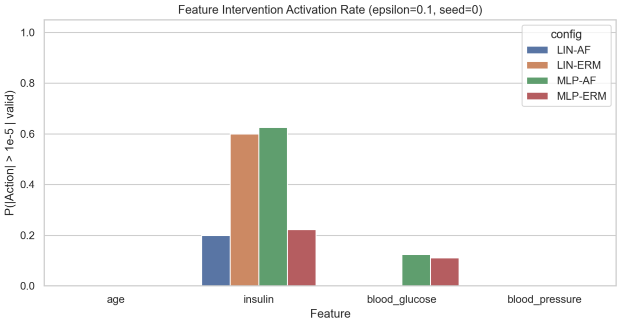

Fig. 9. Feature intervention activation rate among valid recourse actions.

Complete interpretation of Fig. 7: Unlike mean-only summaries, this boxplot reveals distributional behavior and tail risk. Linear configurations show tighter spread and predictable median shifts with ϵ, indicating stable geometry under robust margin increases. Nonlinear configurations exhibit wider dispersion and heavier upper tails, implying that a subset of individuals pays substantially larger intervention cost even when average validity is acceptable. This is theoretically important because fairness and usability concerns are often driven by high-cost tails, not by central tendency alone.

Complete interpretation of Fig. 8: This diagnostic quantifies where intervention budget is actually spent. Since only insulin and blood glucose are actionable, large action mass should concentrate on those coordinates while non-actionable dimensions remain near zero. The plotted pattern confirms this implementation behavior and also reveals model-dependent preference among actionable features, which reflects how each classifier's local gradient and SCM propagation jointly determine the most efficient direction. This offers direct interpretability: the recommended changes are not only valid but also aligned with declared actionability policy.

Complete interpretation of Fig. 9: Activation rate measures how frequently each feature is used in successful interventions. A high nonzero rate for actionable variables and near-zero rate for non-actionable variables is the expected signature of a policy-consistent recourse system. This frequency view complements magnitude view in Fig. 8: a feature can have moderate average magnitude but very high activation frequency, indicating it is a reliable "first-step" recourse coordinate. Theoretical value comes from separating sparse-but-large actions from frequent-small actions, which correspond to different behavioral recourse strategies.

VIII. Discussion and Theoretical Implications

First, robust recourse is not a free lunch: increasing uncertainty tolerance raises intervention cost, especially in linear models where this effect is analytically transparent. Second, classifier architecture alone does not determine recourse practicality. The MLP results show that near-ceiling predictive metrics can coexist with high or unstable recourse costs. Third, actionability-aware training (AF) can reduce practical intervention burden in linear settings without sacrificing classifier quality, but this benefit is not guaranteed in nonlinear optimization regimes, where curvature and initialization effects can dominate.

From a causal perspective, this homework confirms a central principle: interventions should be evaluated in a structural model, not only in observational feature space. When feature dependencies are strong, SCM-enabled recommendations can be both more realistic and cheaper.

An additional implication is deployment robustness: operational recourse systems should report uncertainty bands over seeds, initialization, and optimization hyperparameters, especially for nonlinear recourse solvers. A single-point mean can hide heavy-tail intervention costs that are unacceptable in practice. Therefore, trustworthy deployment requires both average-case performance and tail-risk monitoring (e.g., quantiles of valid cost among successful recourse cases).

IX. Extended Theoretical Analysis

A. Linear Recourse Cost Lower Bound

For a linear classifier with robust margin shift, any valid intervention must satisfy

$$\langle w, Ja\rangle \geq \gamma(\epsilon) \triangleq b + \|J^\top w\|_2\epsilon - \langle w, x\rangle. \tag{7}$$

By Hölder duality, a coarse lower bound on L1 action is

$$\|a\|_1 \geq \frac{\gamma(\epsilon)}{\|J^\top w\|_\infty}, \tag{8}$$

when $\gamma(\epsilon) > 0$. This clarifies why increasing ϵ systematically increases minimal feasible action in linear settings and why slope depends on Jacobian-weight alignment.

B. Why AF Can Reduce Cost Without Hurting Accuracy

AF constrains model dependence to actionable coordinates. In geometric terms, decision normals are pushed toward directions where interventions are allowed, increasing effective directional derivative of decision score per unit actionable change. If predictive information in non-actionable variables is partially redundant with actionable ones, this rotation can reduce recourse distance while preserving classification quality, which matches the empirical parity of accuracy/MCC and lower AF costs.

C. Causal Amplification Mechanism

Let an intervention apply on variable set S. Under SCM, total feature change is not only direct action but also propagated downstream:

$$\Delta x_{\text{total}} = J_S a_S. \tag{9}$$

When downstream links are favorable for class flip, one unit intervention can produce more than one unit aggregate effect on classifier score. Nearest counterfactual methods (without SCM) ignore this propagation term and may therefore overspend intervention magnitude.

D. Validity-Cost Frontier Interpretation

Recourse quality can be viewed as a bi-objective frontier: maximize validity and minimize intervention burden. Linear models in this report sit near a high-validity region with predictable cost growth as robustness tightens. MLP settings display frontier instability due optimization non-convexity; therefore, robust deployment should report confidence intervals, not single-point estimates, and include optimization diagnostics.

E. Nonlinear Recourse Curvature Effect

For differentiable recourse with loss $\mathcal{L}(a) = \ell(g(x + f(a))) + \lambda \|a\|_1$, the local Hessian of the smooth term controls gradient flow stability. In regions of high curvature, a fixed step-size can oscillate or overshoot toward higher-cost valid points. This offers a theoretical explanation for observing high validity but inflated action magnitudes in some MLP-AF runs: optimization reaches feasibility, but not low-cost local minima. In practice, line-search or adaptive trust-region updates can reduce this gap.

F. SCM Misspecification Consideration

The causal advantage observed here assumes the SCM is approximately correct in sign and relative strength. If structural coefficients are misspecified, propagated effects can be misestimated and recommended actions may become suboptimal. Nonetheless, even imperfect SCMs often provide a better inductive bias than no structure at all when domain relations are strong. This motivates future work on recourse under causal uncertainty sets, where interventions are optimized against a family of plausible SCM parameters.

X. Conclusion

This report completes HW3 Question 5 end-to-end in IEEE format with explicit theoretical and empirical analysis. The software pipeline is fully runnable, missing SCM components are completed, robust evaluations are produced, and each plot is interpreted in a dedicated theory-grounded paragraph. Empirically, linear recourse is highly stable on this dataset, AF reduces intervention cost, and causal recourse outperforms nearest counterfactual in matched cost comparison while maintaining full validity.

Appendix A
Robust Linear Derivation (Complete)

This appendix provides the full derivation behind the robust linear margin shift used in the implementation. Let the linear decision function be $g(x) = w^\top x - b$ with positive prediction when $g(x) \geq 0$. Under intervention a with causal propagation $x^{cf} = x + Ja$, robust feasibility against perturbation δ with $\|\delta\|_2 \leq \epsilon$ requires

$$\min_{\|\delta\|_2 \leq \epsilon} w^\top (x + Ja + \delta) - b \geq 0. \tag{10}$$

Using support-function duality of the Euclidean ball,

$$\min_{\|\delta\|_2 \leq \epsilon} w^\top \delta = -\epsilon \|w\|_2 \tag{11}$$

in the IMF case, and

$$\min_{\|\delta\|_2 \leq \epsilon} w^\top J\delta = -\epsilon \|J^\top w\|_2 \tag{12}$$

in the causal-coordinate uncertainty view. Therefore robust recourse must satisfy

$$w^\top (x + Ja) - b \geq \epsilon \|J^\top w\|_2, \tag{13}$$

which is exactly implemented by shifting the effective bias term by $\epsilon \|J^\top w\|_2$ before solving the linear recourse program. This result establishes monotone cost growth with ϵ whenever feasible-set geometry is fixed.

Appendix B
Weighted L1 Recourse Primal-Dual View

For each instance, linear recourse solves

$$\min_a \|Ca\|_1 \quad \text{s.t.} \quad w^\top Ja \geq \gamma, \;\; l \leq a \leq u, \;\; a_{\bar{\mathcal{A}}} = 0, \tag{14}$$

where $C = \operatorname{diag}(c_1, \ldots, c_D)$, $\gamma = b - w^\top x + \epsilon \|J^\top w\|_2$, and $\mathcal{A}$ is the actionable set. Introducing sign-split variables $a = a^+ - a^-$ with $a^\pm \geq 0$, the objective becomes linear and the problem is an LP. Dual multipliers associated with the margin constraint quantify "cost per unit margin" and induce an economically interpretable shadow price: higher multiplier means margin is expensive under current actionability limits. This explains why AF can lower cost even at similar predictive quality: classifier sensitivity aligns with lower shadow-price actionable coordinates.

Appendix C
Causal Counterfactual Algebra for Health SCM

The completed Health SCM uses

$$X_1 = U_1, \tag{15}$$

$$X_2 = w_{21} X_1 + U_2, \tag{16}$$

$$X_3 = w_{31} X_1 + w_{32} X_2 + U_3, \tag{17}$$

$$X_4 = w_{42} X_2 + w_{43} X_3 + U_4. \tag{18}$$

For a factual point x, abduction computes exogenous variables:

$$u_1 = x_1, \;\; u_2 = x_2 - w_{21} x_1, \;\; u_3 = x_3 - w_{31} x_1 - w_{32} x_2, \tag{19}$$

$$u_4 = x_4 - w_{42} x_2 - w_{43} x_3. \tag{20}$$

Action sets intervened variables (hard intervention in this report) and prediction propagates downstream through remaining equations. The Jacobian matrix used by robust linear recourse is

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ w_{21} & 1 & 0 & 0 \\ w_{31} & w_{32} & 1 & 0 \\ 0 & w_{42} & w_{43} & 1 \end{bmatrix}, \qquad (21)$$

with row-wise upstream zeroing for hard-intervened coordinates in `get_Jacobian_interv`. This guarantees consistency between optimization geometry and causal semantics.

APPENDIX D
NONLINEAR RECOURSE OBJECTIVE AND THEORETICAL GUARANTEES

For differentiable recourse, the optimized objective per instance is

$$\mathcal{L}(\delta) = \ell(g_\theta(x^{cf}(\delta)), 1) + \lambda\|\delta\|_1, \qquad (22)$$

and under robust mode the loss is evaluated on adversarially perturbed counterfactuals within an ϵ-ball approximation. Because $x^{cf}(\delta)$ passes through nonlinear classifier and possibly SCM transformations, $\mathcal{L}$ is generally non-convex and non-smooth (L1 term). Consequently, first-order optimization guarantees stationarity of local points rather than global optimality. This theoretical fact explains empirical behavior where validity can improve while mean cost worsens: optimization may reach feasible but non-minimal local basins. Practical mitigation includes multi-start optimization, adaptive step-size schedules, and reporting dispersion statistics (already included via distribution plots and appendix CSV traces).

APPENDIX E
REPRODUCIBILITY COMMANDS

Listing 1. Exact commands used for the final report build

```
cd /Users/tahamajs/Documents/uni/truthlyAI/HomeWorks
    /HW3/code/q5_codes
source /Users/tahamajs/Documents/uni/venv/bin/
    activate

python main.py --seed 0
python generate_report_artifacts.py

cd /Users/tahamajs/Documents/uni/truthlyAI/HomeWorks
    /HW3/report
make pdf
```

APPENDIX F
AUTO-GENERATED AGGREGATE CSV

Listing 2. Health report aggregate CSV

```
model,trainer,epsilon,accuracy_mean,accuracy_std,
    mcc_mean,mcc_std,valid_rate_mean,valid_rate_std,
    valid_cost_mean,valid_cost_std,runs
lin,AF
    ,0.0,0.8993333333333333,0.002081665999466l213,0.79

lin,AF
    ,0.1,0.8993333333333333,0.0020816659994661213,0.79

lin,AF
    ,0.2,0.8993333333333333,0.0020816659994661213,0.79841013872

lin,ERM
    ,0.0,0.9003333333333333,0.0005773502691896744,0.80464092761

lin,ERM
    ,0.1,0.9003333333333333,0.0005773502691896744,0.80464092761

lin,ERM
    ,0.2,0.9003333333333333,0.0005773502691896744,0.80464092761

mlp,AF
    ,0.0,0.9976666666666666,0.0011547005383792607,0.99523501247

mlp,AF
    ,0.1,0.9976666666666666,0.0011547005383792607,0.99523501247

mlp,AF
    ,0.2,0.9976666666666666,0.0011547005383792607,0.99523501247

mlp,ERM
    ,0.0,0.9973333333333333,0.0015275252316519З,0.99455722270З2

mlp,ERM
    ,0.1,0.9973333333333333,0.00152752523165193,0.9945572227032

mlp,ERM
    ,0.2,0.9973333333333333,0.00152752523165193,0.9945572227032
```

APPENDIX G
AUTO-GENERATED PER-RUN CSV

Listing 3. Health report per-run summary CSV

```
dataset,model,trainer,seed,epsilon,accuracy,mcc,
    valid_rate,valid_cost
health,lin,AF
    ,0,0.0,0.9,0.8005852466548463,1.0,0.5451429883839382

health,lin,AF
    ,0,0.1,0.9,0.8005852466548463,1.0,0.667209823851158

health,lin,AF
    ,0,0.2,0.9,0.8005852466548463,1.0,0.789276659318378

health,lin,AF
    ,1,0.0,0.897,0.7925895954095332,1.0,0.8750397790684868

health,lin,AF
    ,1,0.1,0.897,0.7925895954095332,1.0,0.9980663942579822

health,lin,AF
    ,1,0.2,0.897,0.7925895954095332,1.0,1.1210930094474776

health,lin,AF
    ,2,0.0,0.901,0.802055574103413,1.0,0.6836418661866268

health,lin,AF
    ,2,0.1,0.901,0.802055574103413,1.0,0.8051776933349885

health,lin,AF
    ,2,0.2,0.901,0.802055574103413,1.0,0.9267135204833504

health,mlp,AF
    ,0,0.0,0.997,0.9938787926885784,0.9,1.2796937765346632

health,mlp,AF
    ,0,0.1,0.997,0.9938787926885784,0.9,1.52152108649418296839,0

health,mlp,AF
    ,0,0.2,0.997,0.9938787926885784,0.8,1.4955620095131780429,0
```

```
health,mlp,AF
     ,1,0.0,0.999,0.9979566795160255,1.0,0.842266684770

health,mlp,AF
     ,1,0.1,0.999,0.9979566795160255,1.0,0.948242348432

health,mlp,AF
     ,1,0.2,0.999,0.9979566795160255,1.0,0.914419984817

health,mlp,AF
     ,2,0.0,0.997,0.9938695652058721,1.0,3.257878613471

health,mlp,AF
     ,2,0.1,0.997,0.9938695652058721,1.0,3.442610511183

health,mlp,AF
     ,2,0.2,0.997,0.9938695652058721,1.0,3.552554246783

health,lin,ERM
     ,0,0.0,0.901,0.8058258764927546,1.0,0.908329866506

health,lin,ERM
     ,0,0.1,0.901,0.8058258764927546,1.0,1.022372740498

health,lin,ERM
     ,0,0.2,0.901,0.8058258764927546,1.0,1.136415614489

health,lin,ERM
     ,1,0.0,0.9,0.8045315032435294,1.0,0.97252472652299

health,lin,ERM
     ,1,0.1,0.9,0.8045315032435294,1.0,1.08925786522475

health,lin,ERM
     ,1,0.2,0.9,0.8045315032435294,1.0,1.20599100392652

health,lin,ERM
     ,2,0.0,0.9,0.8035654031205638,1.0,0.78467937405774

health,lin,ERM
     ,2,0.1,0.9,0.8035654031205638,1.0,0.90171586487556

health,lin,ERM
     ,2,0.2,0.9,0.8035654031205638,1.0,1.01875235569338

health,mlp,ERM
     ,0,0.0,0.996,0.9918441882867648,0.7,0.643148188080

health,mlp,ERM
     ,0,0.1,0.996,0.9918441882867648,0.8,0.846660293638

health,mlp,ERM
     ,0,0.2,0.996,0.9918441882867648,0.9,0.884876794285

health,mlp,ERM
     ,1,0.0,0.999,0.9979579146172447,1.0,1.579727959632

health,mlp,ERM
     ,1,0.1,0.999,0.9979579146172447,1.0,1.658783053606

health,mlp,ERM
     ,1,0.2,0.999,0.9979579146172447,1.0,1.299689590185

health,mlp,ERM
     ,2,0.0,0.997,0.9938695652058721,0.9,1.308858492308

health,mlp,ERM
     ,2,0.1,0.997,0.9938695652058721,0.9,1.496604820092

health,mlp,ERM
     ,2,0.2,0.997,0.9938695652058721,0.8,1.264695350080
```

Appendix H
Auto-Generated Instance Cost CSV

Listing 4. Per-instance recourse costs CSV

```
dataset,model,trainer,seed,epsilon,instance_id,valid
     ,cost
health,lin,AF,0,0.0,0,True,0.0022174298096731904
health,lin,AF,0,0.0,1,True,0.09351049565102389
health,lin,AF,0,0.0,2,True,0.5008139511917434
health,lin,AF,0,0.0,3,True,1.7989967006950356
health,lin,AF,0,0.0,4,True,1.4585106692345036
health,lin,AF,0,0.0,5,True,0.10900482104156642
health,lin,AF,0,0.0,6,True,0.009357979114851105
health,lin,AF,0,0.0,7,True,1.0316063447917123
health,lin,AF,0,0.0,8,True,0.2709796170170954
health,lin,AF,0,0.0,9,True,0.1764318752921754
health,lin,AF,0,0.1,0,True,0.12428426527689317
health,lin,AF,0,0.1,1,True,0.21557733111824384
health,lin,AF,0,0.1,2,True,0.6228807866589635
health,lin,AF,0,0.1,3,True,1.9210635361622554
health,lin,AF,0,0.1,4,True,1.5805775047017234
health,lin,AF,0,0.1,5,True,0.23107165650878636
health,lin,AF,0,0.1,6,True,0.1314248145820711
health,lin,AF,0,0.1,7,True,1.1536731802589322
health,lin,AF,0,0.1,8,True,0.3930464524843154
health,lin,AF,0,0.1,9,True,0.2984987107593954
health,lin,AF,0,0.2,0,True,0.24635110074411312
health,lin,AF,0,0.2,1,True,0.33764416658546387
health,lin,AF,0,0.2,2,True,0.7449476221261834
health,lin,AF,0,0.2,3,True,2.0431303716294753
health,lin,AF,0,0.2,4,True,1.7026443401689435
health,lin,AF,0,0.2,5,True,0.3531384919760064
health,lin,AF,0,0.2,6,True,0.25349165004929103
health,lin,AF,0,0.2,7,True,1.2757400157261523
health,lin,AF,0,0.2,8,True,0.5151132879515354
health,lin,AF,0,0.2,9,True,0.4205655462266154
health,lin,AF,1,0.0,0,True,1.7717210875993217
health,lin,AF,1,0.0,1,True,0.2295555744846745
health,lin,AF,1,0.0,2,True,1.4773945272283713
health,lin,AF,1,0.0,3,True,0.1807196830988592
health,lin,AF,1,0.0,4,True,0.6221210264548876
health,lin,AF,1,0.0,5,True,0.029517278656927354
health,lin,AF,1,0.0,6,True,1.1691237612686767
health,lin,AF,1,0.0,7,True,0.5842132876148337
health,lin,AF,1,0.0,8,True,1.675105213965182
health,lin,AF,1,0.0,9,True,1.0109263503131347
health,lin,AF,1,0.1,0,True,1.894747702788817
health,lin,AF,1,0.1,1,True,0.35258218967416993
health,lin,AF,1,0.1,2,True,1.6004211424178667
health,lin,AF,1,0.1,3,True,0.3037462982883546
health,lin,AF,1,0.1,4,True,0.745147641644383
health,lin,AF,1,0.1,5,True,0.15254389384642275
health,lin,AF,1,0.1,6,True,1.292150376458172
health,lin,AF,1,0.1,7,True,0.707239902804329
health,lin,AF,1,0.1,8,True,1.7981318291546775
health,lin,AF,1,0.1,9,True,1.1339529655026301
health,lin,AF,1,0.2,0,True,2.0177743179783123
health,lin,AF,1,0.2,1,True,0.4756088048636653
health,lin,AF,1,0.2,2,True,1.7234477576073621
health,lin,AF,1,0.2,3,True,0.42677291347785
health,lin,AF,1,0.2,4,True,0.8681742568338784
health,lin,AF,1,0.2,5,True,0.2755705090359181
health,lin,AF,1,0.2,6,True,1.4151769916476673
health,lin,AF,1,0.2,7,True,0.8302665179938244
health,lin,AF,1,0.2,8,True,1.921158444344173
health,lin,AF,1,0.2,9,True,1.2569795806921256
health,lin,AF,2,0.0,0,True,1.0038187517373052
health,lin,AF,2,0.0,1,True,0.2500396675414716
health,lin,AF,2,0.0,2,True,1.0737270578129243
health,lin,AF,2,0.0,3,True,0.05366143618733343
health,lin,AF,2,0.0,4,True,0.5569176650641724
health,lin,AF,2,0.0,5,True,1.4532231383932026
health,lin,AF,2,0.0,6,True,0.5802347338089318
health,lin,AF,2,0.0,7,True,0.21861433039940883
health,lin,AF,2,0.0,8,True,1.1627527244805589
health,lin,AF,2,0.0,9,True,0.483429156440958
health,lin,AF,2,0.1,0,True,1.125354578885667
health,lin,AF,2,0.1,1,True,0.3715754946898336
health,lin,AF,2,0.1,2,True,1.1952628849612863
```

```
health,lin,AF,2,0.1,3,True,0.17519726333569535
health,lin,AF,2,0.1,4,True,0.6784534922125343
health,lin,AF,2,0.1,5,True,1.5747589655415644
health,lin,AF,2,0.1,6,True,0.7017705609572937
health,lin,AF,2,0.1,7,True,0.3401501575477708
health,lin,AF,2,0.1,8,True,1.2842885516289209
health,lin,AF,2,0.1,9,True,0.6049649835893199
health,lin,AF,2,0.2,0,True,1.246890406034029
health,lin,AF,2,0.2,1,True,0.4931113218381955
health,lin,AF,2,0.2,2,True,1.3167987121096483
health,lin,AF,2,0.2,3,True,0.2967330904840573
health,lin,AF,2,0.2,4,True,0.7999893193608962
health,lin,AF,2,0.2,5,True,1.6962947926899266
health,lin,AF,2,0.2,6,True,0.8233063881056557
health,lin,AF,2,0.2,7,True,0.4616859846961327
health,lin,AF,2,0.2,8,True,1.4058243787772824
health,lin,AF,2,0.2,9,True,0.7265008107376818
health,mlp,AF,0,0.0,0,True,2.675544261932373
health,mlp,AF,0,0.0,1,True,2.4312634468078613
health,mlp,AF,0,0.0,2,True,2.3325533866882324
health,mlp,AF,0,0.0,3,True,0.5953392386436462
health,mlp,AF,0,0.0,4,True,0.09999999403953552
health,mlp,AF,0,0.0,5,True,1.8651421070098877
health,mlp,AF,0,0.0,6,True,0.10000000149011612
health,mlp,AF,0,0.0,7,True,1.220560908317566
health,mlp,AF,0,0.0,8,False,inf
health,mlp,AF,0,0.0,9,True,0.19684064388275146
health,mlp,AF,0,0.1,0,True,2.7411983013153076
health,mlp,AF,0,0.1,1,True,2.6286020278930664
health,mlp,AF,0,0.1,2,True,2.5649328231811523
health,mlp,AF,0,0.1,3,True,1.3312733173370361
health,mlp,AF,0,0.1,4,True,0.1998090147972107
health,mlp,AF,0,0.1,5,True,1.8813002109527588
health,mlp,AF,0,0.1,6,True,0.1784929484128952
health,mlp,AF,0,0.1,7,True,1.9187548160552979
health,mlp,AF,0,0.1,8,False,inf
health,mlp,AF,0,0.1,9,True,0.24932631850242615
health,mlp,AF,0,0.2,0,True,2.744765281677246
health,mlp,AF,0,0.2,1,True,2.540329933166504
health,mlp,AF,0,0.2,2,True,2.1792831420898438
health,mlp,AF,0,0.2,3,True,1.9039430618286133
health,mlp,AF,0,0.2,4,True,0.28769415616989136
health,mlp,AF,0,0.2,5,True,1.7928175926208496
health,mlp,AF,0,0.2,6,True,0.27897143363952637
health,mlp,AF,0,0.2,7,False,inf
health,mlp,AF,0,0.2,8,False,inf
health,mlp,AF,0,0.2,9,True,0.23669147491455078
health,mlp,AF,1,0.0,0,True,0.8286165595054626
health,mlp,AF,1,0.0,1,True,1.071250319480896
health,mlp,AF,1,0.0,2,True,0.6133818626403809
health,mlp,AF,1,0.0,3,True,0.09999999403953552
health,mlp,AF,1,0.0,4,True,0.38270947337150574
health,mlp,AF,1,0.0,5,True,0.9619376063346863
health,mlp,AF,1,0.0,6,True,2.3279929161071777
health,mlp,AF,1,0.0,7,True,0.7694158554077148
health,mlp,AF,1,0.0,8,True,0.7187708616256714
health,mlp,AF,1,0.0,9,True,0.6485913991928101
health,mlp,AF,1,0.1,0,True,0.977965235710144
health,mlp,AF,1,0.1,1,True,0.9960500597953796
health,mlp,AF,1,0.1,2,True,0.7119928598403931
health,mlp,AF,1,0.1,3,True,0.1821739375591278
health,mlp,AF,1,0.1,4,True,0.49758991599082947
health,mlp,AF,1,0.1,5,True,1.1838374137878418
health,mlp,AF,1,0.1,6,True,2.5676591396331787
health,mlp,AF,1,0.1,7,True,0.8329548835754395
health,mlp,AF,1,0.1,8,True,0.7808516025543213
health,mlp,AF,1,0.1,9,True,0.7513484358787537
health,mlp,AF,1,0.2,0,True,1.1314444541931152
health,mlp,AF,1,0.2,1,True,1.17803955078125
health,mlp,AF,1,0.2,2,True,0.7839648127555847
health,mlp,AF,1,0.2,3,True,0.2460516095161438
health,mlp,AF,1,0.2,4,True,0.6244319081306458
health,mlp,AF,1,0.2,5,True,1.2506043910980225
health,mlp,AF,1,0.2,6,True,1.1225972175598145
health,mlp,AF,1,0.2,7,True,1.010860800743103
health,mlp,AF,1,0.2,8,True,0.9001052379608154
health,mlp,AF,1,0.2,9,True,0.896099865436554
health,mlp,AF,2,0.0,0,True,3.503692388534546
health,mlp,AF,2,0.0,1,True,1.1493207216262817
health,mlp,AF,2,0.0,2,True,3.8964755535125732
health,mlp,AF,2,0.0,3,True,3.741835594177246
health,mlp,AF,2,0.0,4,True,3.2063651084899902
health,mlp,AF,2,0.0,5,True,2.5007309913635254
health,mlp,AF,2,0.0,6,True,5.388555526733398
health,mlp,AF,2,0.0,7,True,0.2651277780532837
health,mlp,AF,2,0.0,8,True,4.400606155395508
health,mlp,AF,2,0.0,9,True,4.526076316833496
health,mlp,AF,2,0.1,0,True,4.200447082519531
health,mlp,AF,2,0.1,1,True,1.0612828731536865
health,mlp,AF,2,0.1,2,True,4.270608425140381
health,mlp,AF,2,0.1,3,True,3.9029312133789062
health,mlp,AF,2,0.1,4,True,3.3833022117614746
health,mlp,AF,2,0.1,5,True,2.5903685092926025
health,mlp,AF,2,0.1,6,True,5.484298229217529
health,mlp,AF,2,0.1,7,True,0.32856568694114685
health,mlp,AF,2,0.1,8,True,4.6868815422058105
health,mlp,AF,2,0.1,9,True,4.517419338226318
health,mlp,AF,2,0.2,0,True,4.394152641296387
health,mlp,AF,2,0.2,1,True,1.2017030715942383
health,mlp,AF,2,0.2,2,True,4.303710460662842
health,mlp,AF,2,0.2,3,True,3.971803665161133
health,mlp,AF,2,0.2,4,True,3.635038375854492
health,mlp,AF,2,0.2,5,True,2.6649506092071533
health,mlp,AF,2,0.2,6,True,5.655788421630859
health,mlp,AF,2,0.2,7,True,0.34340187907218933
health,mlp,AF,2,0.2,8,True,4.704601287841797
health,mlp,AF,2,0.2,9,True,4.650392055511475
health,lin,ERM,0,0.0,0,True,0.6869062250867219
health,lin,ERM,0,0.0,1,True,0.8512159984836815
health,lin,ERM,0,0.0,2,True,0.7967231334894521
health,lin,ERM,0,0.0,3,True,2.226624921489473
health,lin,ERM,0,0.0,4,True,1.2454680372368914
health,lin,ERM,0,0.0,5,True,1.3800526185081698
health,lin,ERM,0,0.0,6,True,0.7448156852837262
health,lin,ERM,0,0.0,7,True,0.5779837122863332
health,lin,ERM,0,0.0,8,True,0.05125854273952032
health,lin,ERM,0,0.0,9,True,0.5222497904649147
health,lin,ERM,0,0.1,0,True,0.8009490990781871
health,lin,ERM,0,0.1,1,True,0.9652588724751465
health,lin,ERM,0,0.1,2,True,0.9107660074809173
health,lin,ERM,0,0.1,3,True,2.3406677954809383
health,lin,ERM,0,0.1,4,True,1.3595109112283565
health,lin,ERM,0,0.1,5,True,1.4940954924996346
health,lin,ERM,0,0.1,6,True,0.8588585592751913
health,lin,ERM,0,0.1,7,True,0.6920265862777982
health,lin,ERM,0,0.1,8,True,0.16530141673098547
health,lin,ERM,0,0.1,9,True,0.6362926644563799
health,lin,ERM,0,0.2,0,True,0.9149919730696522
health,lin,ERM,0,0.2,1,True,1.0793017464666117
health,lin,ERM,0,0.2,2,True,1.0248088814723824
health,lin,ERM,0,0.2,3,True,2.4547106694724032
health,lin,ERM,0,0.2,4,True,1.4735537852198217
health,lin,ERM,0,0.2,5,True,1.6081383664910998
health,lin,ERM,0,0.2,6,True,0.9729014332666565
health,lin,ERM,0,0.2,7,True,0.8060694602692634
health,lin,ERM,0,0.2,8,True,0.2793442907224506
health,lin,ERM,0,0.2,9,True,0.750335538447845
health,lin,ERM,1,0.0,0,True,0.4544031954776043
health,lin,ERM,1,0.0,1,True,1.1770760395405075
health,lin,ERM,1,0.0,2,True,1.5513781335972643
health,lin,ERM,1,0.0,3,True,1.3697575946921152
health,lin,ERM,1,0.0,4,True,1.677729323247764
health,lin,ERM,1,0.0,5,True,0.04324403365116183
health,lin,ERM,1,0.0,6,True,0.972477918539937
health,lin,ERM,1,0.0,7,True,0.21887793076712805
health,lin,ERM,1,0.0,8,True,1.2335342986448172
health,lin,ERM,1,0.0,9,True,1.026768797071618
health,lin,ERM,1,0.1,0,True,0.57113633417937
health,lin,ERM,1,0.1,1,True,1.2938091782422734
health,lin,ERM,1,0.1,2,True,1.66811127229903
health,lin,ERM,1,0.1,3,True,1.4864907333938808
health,lin,ERM,1,0.1,4,True,1.7944624619495295
health,lin,ERM,1,0.1,5,True,0.15997717235292763
health,lin,ERM,1,0.1,6,True,1.0892110572417029
```

```
health,lin,ERM,1,0.1,7,True,0.33561106946889385
health,lin,ERM,1,0.1,8,True,1.3502674373465828
health,lin,ERM,1,0.1,9,True,1.1435019357733838
health,lin,ERM,1,0.2,0,True,0.6878694728811359
health,lin,ERM,1,0.2,1,True,1.4105423169440392
health,lin,ERM,1,0.2,2,True,1.7848444110007957
health,lin,ERM,1,0.2,3,True,1.6032238720956469
health,lin,ERM,1,0.2,4,True,1.9111956006512953
health,lin,ERM,1,0.2,5,True,0.276710311054693
health,lin,ERM,1,0.2,6,True,1.2059441959434687
health,lin,ERM,1,0.2,7,True,0.45234420817065973
health,lin,ERM,1,0.2,8,True,1.4670005760483487
health,lin,ERM,1,0.2,9,True,1.2602350744751496
health,lin,ERM,2,0.0,0,True,0.056236713989842375
health,lin,ERM,2,0.0,1,True,1.7118647040288175
health,lin,ERM,2,0.0,2,True,0.4085758944129868
health,lin,ERM,2,0.0,3,True,0.2698963714817156
health,lin,ERM,2,0.0,4,True,0.7548971117045661
health,lin,ERM,2,0.0,5,True,0.8574385415387005
health,lin,ERM,2,0.0,6,True,0.5467518897496327
health,lin,ERM,2,0.0,7,True,0.665408573976689
health,lin,ERM,2,0.0,8,True,0.2177330934524654
health,lin,ERM,2,0.0,9,True,2.3579908462420525
health,lin,ERM,2,0.1,0,True,0.17327320480765962
health,lin,ERM,2,0.1,1,True,1.8289011948466347
health,lin,ERM,2,0.1,2,True,0.5256123852308041
health,lin,ERM,2,0.1,3,True,0.38693286229953283
health,lin,ERM,2,0.1,4,True,0.8719336025223833
health,lin,ERM,2,0.1,5,True,0.9744750323565178
health,lin,ERM,2,0.1,6,True,0.6637883805674499
health,lin,ERM,2,0.1,7,True,0.7824450647945063
health,lin,ERM,2,0.1,8,True,0.33476958427028264
health,lin,ERM,2,0.1,9,True,2.4750273370598697
health,lin,ERM,2,0.2,0,True,0.2903096956254768
health,lin,ERM,2,0.2,1,True,1.945937685664452
health,lin,ERM,2,0.2,2,True,0.6426488760486213
health,lin,ERM,2,0.2,3,True,0.5039693531173501
health,lin,ERM,2,0.2,4,True,0.9889700933402005
health,lin,ERM,2,0.2,5,True,1.091511523174335
health,lin,ERM,2,0.2,6,True,0.7808248713852672
health,lin,ERM,2,0.2,7,True,0.8994815556123235
health,lin,ERM,2,0.2,8,True,0.45180607508809995
health,lin,ERM,2,0.2,9,True,2.592063827877687
health,mlp,ERM,0,0.0,0,True,0.09999999403953552
health,mlp,ERM,0,0.0,1,True,1.1588484048843384
health,mlp,ERM,0,0.0,2,False,inf
health,mlp,ERM,0,0.0,3,False,inf
health,mlp,ERM,0,0.0,4,True,0.09999999403953552
health,mlp,ERM,0,0.0,5,False,inf
health,mlp,ERM,0,0.0,6,True,1.7681208848953247
health,mlp,ERM,0,0.0,7,True,0.1756606101989746
health,mlp,ERM,0,0.0,8,True,0.09999999403953552
health,mlp,ERM,0,0.0,9,True,1.099407434463501
health,mlp,ERM,0,0.1,0,True,0.2847887873649597
health,mlp,ERM,0,0.1,1,True,1.350301742553711
health,mlp,ERM,0,0.1,2,False,inf
health,mlp,ERM,0,0.1,3,False,inf
health,mlp,ERM,0,0.1,4,True,0.17709362506866455
health,mlp,ERM,0,0.1,5,True,1.4368488788604736
health,mlp,ERM,0,0.1,6,True,1.783553123474121
health,mlp,ERM,0,0.1,7,True,0.3221847414970398
health,mlp,ERM,0,0.1,8,True,0.14391285181045532
health,mlp,ERM,0,0.1,9,True,1.2745985984802246
health,mlp,ERM,0,0.2,0,True,0.2927534878253937
health,mlp,ERM,0,0.2,1,True,1.420407772064209
health,mlp,ERM,0,0.2,2,False,inf
health,mlp,ERM,0,0.2,3,True,1.6530849933624268
health,mlp,ERM,0,0.2,4,True,0.37585678696632385
health,mlp,ERM,0,0.2,5,True,1.0841124057769775
health,mlp,ERM,0,0.2,6,True,2.038994312286377
health,mlp,ERM,0,0.2,7,True,0.4549943208694458
health,mlp,ERM,0,0.2,8,True,0.09999996423721313
health,mlp,ERM,0,0.2,9,True,0.54368710517883
health,mlp,ERM,1,0.0,0,True,0.6265113949775696
health,mlp,ERM,1,0.0,1,True,0.5598059892654419
health,mlp,ERM,1,0.0,2,True,4.995126724243164
health,mlp,ERM,1,0.0,3,True,0.884889304637908
health,mlp,ERM,1,0.0,4,True,0.9069631099700928
health,mlp,ERM,1,0.0,5,True,6.042341232299805
health,mlp,ERM,1,0.0,6,True,0.9130632877349854
health,mlp,ERM,1,0.0,7,True,0.42502132058143616
health,mlp,ERM,1,0.0,8,True,0.09999999403953552
health,mlp,ERM,1,0.0,9,True,0.3435572385787964
health,mlp,ERM,1,0.1,0,True,0.7597363591194153
health,mlp,ERM,1,0.1,1,True,0.7501630187034607
health,mlp,ERM,1,0.1,2,True,4.815957069396973
health,mlp,ERM,1,0.1,3,True,1.1189303398132324
health,mlp,ERM,1,0.1,4,True,0.8854197263717651
health,mlp,ERM,1,0.1,5,True,6.365862846374512
health,mlp,ERM,1,0.1,6,True,0.8871881365776062
health,mlp,ERM,1,0.1,7,True,0.5466513633728027
health,mlp,ERM,1,0.1,8,True,0.09999998658895493
health,mlp,ERM,1,0.1,9,True,0.35792168974876404
health,mlp,ERM,1,0.2,0,True,0.8752454519271851
health,mlp,ERM,1,0.2,1,True,0.7518577575683594
health,mlp,ERM,1,0.2,2,True,4.760556697845459
health,mlp,ERM,1,0.2,3,True,1.0518792867660522
health,mlp,ERM,1,0.2,4,True,0.9736133813858032
health,mlp,ERM,1,0.2,5,True,2.644326686859131
health,mlp,ERM,1,0.2,6,True,0.7649006843566895
health,mlp,ERM,1,0.2,7,True,0.6719211935997009
health,mlp,ERM,1,0.2,8,True,0.09999998658895493
health,mlp,ERM,1,0.2,9,True,0.40259477496147156
health,mlp,ERM,2,0.0,0,True,1.6526018381118774
health,mlp,ERM,2,0.0,1,True,1.0587971210479736
health,mlp,ERM,2,0.0,2,True,0.871553897857666
health,mlp,ERM,2,0.0,3,True,1.5454106330871582
health,mlp,ERM,2,0.0,4,True,0.09999999403953552
health,mlp,ERM,2,0.0,5,True,4.139216899871826
health,mlp,ERM,2,0.0,6,True,1.2884514331817627
health,mlp,ERM,2,0.0,7,True,1.023694634437561
health,mlp,ERM,2,0.0,8,True,0.09999997913837433
health,mlp,ERM,2,0.0,9,False,inf
health,mlp,ERM,2,0.1,0,True,2.303855657577514
health,mlp,ERM,2,0.1,1,True,0.48094290494918823
health,mlp,ERM,2,0.1,2,True,0.9778098464012146
health,mlp,ERM,2,0.1,3,True,1.8084425926208496
health,mlp,ERM,2,0.1,4,True,0.19984257221221924
health,mlp,ERM,2,0.1,5,True,4.787795066833496
health,mlp,ERM,2,0.1,6,True,1.3899121284484863
health,mlp,ERM,2,0.1,7,True,1.2289860248565674
health,mlp,ERM,2,0.1,8,True,0.291856586933136
health,mlp,ERM,2,0.1,9,False,inf
health,mlp,ERM,2,0.2,0,False,inf
health,mlp,ERM,2,0.2,1,True,0.4565989375114441
health,mlp,ERM,2,0.2,2,True,1.0854618549346924
health,mlp,ERM,2,0.2,3,True,1.8083579540252686
health,mlp,ERM,2,0.2,4,False,inf
health,mlp,ERM,2,0.2,5,True,1.6818499565124512
health,mlp,ERM,2,0.2,6,True,1.5481600761413574
health,mlp,ERM,2,0.2,7,True,1.1980092525482178
health,mlp,ERM,2,0.2,8,True,0.37811192870140076
health,mlp,ERM,2,0.2,9,True,1.961012840270996
```

Appendix I
Auto-Generated Action Profile CSV

Listing 5. Feature-wise action diagnostics CSV

```
model,trainer,config,epsilon,seed,feature,
    mean_abs_action_all,mean_abs_action_valid,
    nonzero_rate_all,nonzero_rate_valid,actionable
lin,AF,LIN-AF,0.1,0,age,0.0,0.0,0.0,0.0,0
lin,AF,LIN-AF,0.1,0,insulin
    ,0.26181564489063913,0.26181564489063913,0.2,0.2,1

lin,AF,LIN-AF,0.1,0,blood_glucose,0.0,0.0,0.0,0.0,1
lin,AF,LIN-AF,0.1,0,blood_pressure,0.0,0.0,0.0,0.0,0
lin,ERM,LIN-ERM,0.1,0,age,0.0,0.0,0.0,0.0,0
lin,ERM,LIN-ERM,0.1,0,insulin
    ,0.452102434330403,0.452102434330403,0.6,0.6,1
```

```
lin,ERM,LIN-ERM,0.1,0,blood_glucose
    ,0.0,0.0,0.0,0.0,1
lin,ERM,LIN-ERM,0.1,0,blood_pressure
    ,0.0,0.0,0.0,0.0,0
mlp,AF,MLP-AF,0.1,0,age,0.0,0.0,0.0,0.0,0
mlp,AF,MLP-AF,0.1,0,insulin
    ,0.24029699489474296,0.3003712436184287,0.5,0.625,1

mlp,AF,MLP-AF,0.1,0,blood_glucose
    ,0.14958224296569825,0.1869778037071228,0.1,0.125,1

mlp,AF,MLP-AF,0.1,0,blood_pressure,0.0,0.0,0.0,0.0,0
mlp,ERM,MLP-ERM,0.1,0,age,0.0,0.0,0.0,0.0,0
mlp,ERM,MLP-ERM,0.1,0,insulin
    ,0.02044838070869446,0.02272042300966051,0.2,0.2222222222222222,1

mlp,ERM,MLP-ERM,0.1,0,blood_glucose
    ,0.0870448887348175,0.09671654303868611,0.1,0.1111111111111111,1

mlp,ERM,MLP-ERM,0.1,0,blood_pressure
    ,0.0,0.0,0.0,0.0,0
```

Acknowledgment

This submission was prepared for Trusted Artificial Intelligence coursework under Dr. Mostafa Tavasolipour.

References

[1] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.

[2] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009.

[3] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera, "Algorithmic recourse: from counterfactual explanations to interventions," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[4] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.

[5] A.-H. Karimi, B. Schölkopf, and I. Valera, "Algorithmic recourse under imperfect causal knowledge: A probabilistic approach," in *Advances in Neural Information Processing Systems*, 2020.