Rheocor v3.0

Comparing Four Machine-Learning Architectures for Cardiovascular Risk Stratification, with Per-Prediction Explainability and Honest Uncertainty

Aryav Kaushik

Technical Summary · June 2026 · Platform version 3.0
Dataset: UCI Heart Disease (Cleveland Clinic Foundation subset)
Code, data, and all results files: project repository (`rheocor`)

Research demonstration only. Not a medical device, not validated for clinical use. All numbers in this document are reproduced by `python run_all.py` from the committed code at pinned dependency versions; each table cites the results file it is read from.

ABSTRACT

We compare four supervised learning architectures, logistic regression (LR), random forest (RF), gradient-boosted trees (XGBoost), and a small multilayer perceptron (NN), on the task of predicting angiographically confirmed obstructive coronary artery disease from 13 routine clinical variables in the UCI Cleveland cohort ($N = 303$, 45.9% disease-positive). Under 5-fold stratified cross-validation with strictly per-fold preprocessing (imputation and standardization fit on each fold's training rows only), mean ROC-AUC ranges from 0.890 ± 0.017 (XGBoost) to 0.912 ± 0.020 (logistic regression). On a held-out 20% test split, percentile-bootstrap 95% confidence intervals ($B = 1000$) on AUC span roughly 0.10 for every model, and pairwise DeLong tests find no statistically significant difference between any two architectures (smallest $p = 0.17$; LR vs. RF $p = 0.83$). The central empirical result is therefore a non-result with practical consequences: model complexity does not buy discriminative performance on this cohort, and the most interpretable model is also the nominal best. The platform couples each prediction with SHAP, LIME, and gradient-times-input attributions, a cross-validation fold-spread stability range, an architecture-agreement readout, and a separately-labeled clinical guideline advisory that never modifies model outputs. SHAP analysis identifies fluoroscopy vessel count, thallium scan result, and chest-pain type as the dominant predictors, with serum cholesterol in the bottom third, a reminder that this target is prevalent anatomical disease rather than long-term incident risk. A FedAvg federated-learning simulation converges to within sampling noise of a centrally trained baseline while exchanging only model weights. Every limitation of the cohort and design is stated explicitly.

1. Introduction

Established clinical risk scores (Framingham-family equations, SCORE2, the ACC/AHA pooled cohort equations) are deliberately simple: linear models over a handful of variables, chosen for portability and auditability. Machine-learning methods promise to capture non-linear structure and feature interactions that such scores cannot represent. In clinical settings the relevant question is not whether a flexible model *can* fit more, it is whether it *does*, at the sample sizes clinical tabular datasets actually have, and whether any gain survives honest uncertainty quantification.

Rheocor makes that question concrete and inspectable: four architectures spanning the bias-variance spectrum, trained on identical leak-free data, evaluated with cross-validation, bootstrap confidence intervals, and formal tests for AUC differences. The platform wraps the models so that every individual prediction can be interrogated (which features drove it, how stable it is across training subsamples, whether the four models agree). The honest answer this design produces on the Cleveland cohort is reported in Section 5.

2. Dataset and Preprocessing

2.1 Data source and target

The Cleveland subset of the UCI Heart Disease dataset (Detrano et al., 1989) comprises $N = 303$ patients with 13 clinical predictors and an ordinal target (0 to 4) encoding the presence and severity of angiographically confirmed coronary artery disease ($\geq 50\%$ luminal narrowing). Following standard practice the target is binarized,

$$y_i = \mathbb{1}[t_i \geq 1], \qquad t_i \in \{0, 1, 2, 3, 4\},$$

which yields 139 disease-positive patients (45.9%) and 164 negative (54.1%), a mild imbalance toward the negative class. We use the raw UCI encoding throughout: `cp` $\in \{1, 2, 3, 4\}$ with 4 = asymptomatic, and `thal` $\in \{3, 6, 7\}$ for normal, fixed defect, and reversible defect.

Feature	Description	Type	Mean (SD) / prevalence
`age`	Age (years)	Continuous	54.4 (9.0)
`sex`	Biological sex (1 = male)	Binary	68% male
`cp`	Chest-pain type (1 to 4; 4 = asymptomatic)	Categorical	48% asymptomatic
`trestbps`	Resting blood pressure (mm Hg)	Continuous	131.7 (17.6)
`chol`	Serum cholesterol (mg/dL)	Continuous	246.7 (51.8)
`fbs`	Fasting blood sugar > 120 mg/dL	Binary	15%
`restecg`	Resting ECG result (0 to 2)	Categorical	50% normal
`thalach`	Maximum heart rate achieved (bpm)	Continuous	149.6 (22.9)
`exang`	Exercise-induced angina	Binary	33%
`oldpeak`	ST depression, exercise vs. rest (mm)	Continuous	1.0 (1.2)
`slope`	Slope of peak-exercise ST segment (1 to 3)	Categorical	46% flat
`ca`	Major vessels colored by fluoroscopy (0 to 3)	Ordinal	0.66 (0.93)
`thal`	Thallium scan (3 / 6 / 7)	Categorical	55% normal, 39% reversible

Table 1. Feature descriptions and distributional statistics, computed from `data/heart_disease_processed.csv`.

2.2 Preprocessing, ordered to prevent leakage

The pipeline (`src/data_pipeline.py`) enforces a strict order so that no statistic estimated from held-out data ever influences training. After binarization, the data is split into stratified train and test sets, and only then are missing values imputed with the *training-set* median $\tilde{x}_j^{\text{train}}$ and features standardized with statistics fit on the training rows alone,

$$x_{ij}^{\text{scaled}} = \frac{x_{ij} - \hat{\mu}_j^{\text{train}}}{\hat{\sigma}_j^{\text{train}}}, \qquad \hat{\mu}_j^{\text{train}} = \frac{1}{n_{\text{tr}}} \sum_{i \in \text{train}} x_{ij}.$$

The six missing cells (`ca`: 4, `thal`: 2; 0.15% of all cells) are imputed after the split. No one-hot encoding is applied: tree models split on the raw integer codes, while the linear and neural models receive the standardized integers, a simplification whose cost, if any, is bounded by the results in Section 5.

An internal audit (June 2026, `AUDIT.md`) found that earlier versions imputed before splitting and selected the neural network's weights against the test set. Both flaws were corrected before the results below were produced.

3. Model Architectures

3.1 Logistic regression

Logistic regression models the posterior probability of disease as a sigmoid of a linear score. With weights $w \in \mathbb{R}^{13}$ and bias b,

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \qquad P(y = 1 \mid x) = \sigma\big(w^{\top} x + b\big).$$

Training minimizes the L_2-regularized negative log-likelihood,

$$\mathcal{L}(w, b) = -\frac{1}{n} \sum_{i=1}^{n} \Big[y_i \log p_i + (1 - y_i) \log(1 - p_i) \Big] \; + \; \frac{1}{2C} \lVert w \rVert_2^2, \qquad p_i = \sigma(w^{\top} x_i + b),$$

with inverse regularization strength $C = 1$ and the LBFGS solver (`max_iter=2000`). The fitted weights make this the one model whose reasoning reads directly off its coefficients.

3.2 Random forest

A random forest averages B decision trees, each grown on an independent bootstrap sample with a random subset of $\lfloor \sqrt{p} \rfloor$ features considered at every split,

$$\hat{p}_{\mathrm{RF}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{p}_b(x), \qquad B = 300.$$

The motivation is the bias-variance decomposition of the expected squared error of an estimator $\hat{f}$,

$$\mathbb{E}\Big[\big(y - \hat{f}(x)\big)^2\Big] = \underbrace{\big(\mathbb{E}[\hat{f}(x)] - f(x)\big)^2}_{\text{bias}^2} + \underbrace{\mathrm{Var}\big[\hat{f}(x)\big]}_{\text{variance}} + \sigma_{\varepsilon}^2.$$

For B trees with pairwise prediction correlation ρ and per-tree variance σ^2, the ensemble variance is

$$\mathrm{Var}\big[\hat{p}_{\mathrm{RF}}\big] = \rho\,\sigma^2 + \frac{1 - \rho}{B}\,\sigma^2,$$

so averaging decorrelated trees (small ρ) shrinks variance toward $\rho\sigma^2$ while leaving bias roughly unchanged. Splits maximize the reduction in Gini impurity, $G(t) = 1 - \sum_k p(k \mid t)^2$, at depth capped to 6.

3.3 Gradient-boosted trees (XGBoost)

XGBoost builds an additive model $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$, choosing each new tree f_t to minimize a second-order Taylor approximation of the loss around the current prediction,

$$\mathcal{O}^{(t)} \approx \sum_{i=1}^{n} \Big[g_i\, f_t(x_i) + \tfrac{1}{2} h_i\, f_t(x_i)^2 \Big] + \Omega(f_t), \qquad g_i = \partial_{\hat{y}}\, \ell\big(y_i, \hat{y}_i^{(t-1)}\big), \;\; h_i = \partial_{\hat{y}}^2\, \ell\big(y_i, \hat{y}_i^{(t-1)}\big),$$

with complexity penalty $\Omega(f) = \gamma T + \tfrac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$ over the T leaves. For a fixed tree structure the optimal weight of leaf j and the resulting gain of a candidate split are

$$w_j^{\star} = -\frac{G_j}{H_j + \lambda}, \qquad \mathrm{Gain} = \tfrac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma,$$

where $G_j = \sum_{i \in j} g_i$ and $H_j = \sum_{i \in j} h_i$. Configuration: 400 trees, learning rate $\eta = 0.05$, `max_depth=4`, row and column subsampling 0.9.

3.4 Multilayer perceptron

The neural network is a small fully-connected classifier with two hidden layers and dropout. For an input $x \in \mathbb{R}^{13}$,

$$h_1 = \mathrm{ReLU}(W_1 x + b_1), \qquad h_2 = \mathrm{ReLU}(W_2 h_1 + b_2), \qquad \hat{y} = \sigma\big(w_3^\top h_2 + b_3\big),$$

with $W_1 \in \mathbb{R}^{32 \times 13}$, $W_2 \in \mathbb{R}^{16 \times 32}$, $w_3 \in \mathbb{R}^{16}$, $\mathrm{ReLU}(z) = \max(0, z)$, and dropout rates 0.3 and 0.2 on the two hidden layers. The parameter count is deliberately small to limit overfitting,

$$(13{\cdot}32 + 32) + (32{\cdot}16 + 16) + (16{\cdot}1 + 1) = 416 + 528 + 17 + 32 = 993 \text{ parameters.}$$

Training minimizes binary cross-entropy with the Adam optimizer ($\alpha = 10^{-3}$, weight decay 10^{-4}), whose moment estimates and update are

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \qquad \theta_t = \theta_{t-1} - \alpha \, \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon},$$

with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. Training runs for at most 200 epochs with early stopping (patience 20) on the loss of a stratified validation split carved from the *training* data; the test set is never seen during training or model selection.

4. Evaluation Methodology

4.1 5-fold stratified cross-validation (headline)

Because a single held-out split yields a high-variance estimate, the headline metrics come from $K = 5$-fold stratified cross-validation over the full dataset (`src/cv_evaluate.py`). The cross-validated estimate of a metric M is the mean over folds,

$$\widehat{M}_{\mathrm{CV}} = \frac{1}{K} \sum_{k=1}^{K} M\big(\hat{f}^{(-k)}, \mathcal{D}_k\big),$$

where $\hat{f}^{(-k)}$ is trained on all folds but k and evaluated on the held-out fold $\mathcal{D}_k$. Folding happens on the raw data: each fold imputes with its own training rows' medians and fits its own scaler, and the fold neural networks are trained by the same procedure as the deployed one (inner validation split, early stopping), so the fold cohort is a faithful population of models trained like the real one.

4.2 Evaluation metrics

With true positives, true negatives, false positives, and false negatives $(\mathrm{TP}, \mathrm{TN}, \mathrm{FP}, \mathrm{FN})$ at threshold 0.5,

$$\mathrm{Acc} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}, \quad \mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad \mathrm{Rec} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},$$
$$F_1 = \frac{2\,\mathrm{Prec}\cdot\mathrm{Rec}}{\mathrm{Prec}+\mathrm{Rec}} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}.$$

The threshold-independent ROC-AUC equals the Wilcoxon-Mann-Whitney statistic, the probability that a random positive is scored above a random negative,

$$\mathrm{AUC} = \Pr\big(\hat{p}(x^+) > \hat{p}(x^-)\big) = \frac{1}{n_+ n_-}\sum_{i:\,y_i=1}\sum_{j:\,y_j=0}\mathbb{1}\big[\hat{p}(x_i) > \hat{p}(x_j)\big].$$

4.3 Bootstrap confidence intervals

For the single held-out split, 95% confidence intervals for AUC use the percentile bootstrap with $B = 1000$ resamples of the test rows drawn with replacement (single-class resamples are redrawn). With $\widehat{\mathrm{AUC}}^{(b)}$ the metric on resample b,

$$\mathrm{CI}_{95} = \Big[\,\widehat{\mathrm{AUC}}^{*}_{(0.025)},\ \widehat{\mathrm{AUC}}^{*}_{(0.975)}\,\Big],$$

where $\widehat{\mathrm{AUC}}^{*}_{(\alpha)}$ is the α-quantile of the bootstrap distribution.

4.4 DeLong's test for correlated AUCs

To test whether two models have equal AUC on the *same* test set, we use DeLong's nonparametric method via the fast structural-components formulation (Sun and Xu, 2014). For model with scores on m positives and n negatives, the per-case components are

$$V_{10}^{(i)} = \frac{1}{n}\sum_{j=1}^{n}\psi\big(X_i, Y_j\big), \qquad V_{01}^{(j)} = \frac{1}{m}\sum_{i=1}^{m}\psi\big(X_i, Y_j\big), \qquad \psi(a,b) = \mathbb{1}[a>b] + \tfrac{1}{2}\mathbb{1}[a=b],$$

whose empirical covariances S_{10}, S_{01} give the covariance of the two AUC estimates, $\Sigma = S_{10}/m + S_{01}/n$. The test statistic for $H_0:\ \mathrm{AUC}_a = \mathrm{AUC}_b$ is asymptotically standard normal,

$$Z = \frac{\widehat{\mathrm{AUC}}_a - \widehat{\mathrm{AUC}}_b}{\sqrt{\Sigma_{aa} + \Sigma_{bb} - 2\Sigma_{ab}}}.$$

5. Results

Model	ROC-AUC	Accuracy	Precision	Recall	F1
Logistic Regression	**0.912 ± 0.020**	0.832 ± 0.055	0.830 ± 0.072	0.798 ± 0.075	0.813 ± 0.061
Random Forest	0.907 ± 0.029	0.828 ± 0.043	0.839 ± 0.060	0.776 ± 0.062	0.805 ± 0.051
Neural Network	0.905 ± 0.027	**0.848 ± 0.053**	0.844 ± 0.079	0.828 ± 0.077	0.833 ± 0.059
XGBoost	0.890 ± 0.017	0.825 ± 0.024	0.835 ± 0.056	0.777 ± 0.015	0.804 ± 0.020

Table 2. 5-fold stratified cross-validation, mean ± SD across folds, per-fold preprocessing (`results/cv_metrics.csv`). These are the project's headline numbers.

Model	ROC-AUC	95% bootstrap CI	Accuracy	F1	Brier
Neural Network	0.958	0.896 to 0.998	0.869	0.867	0.084
Random Forest	0.954	0.889 to 0.996	0.885	0.881	0.097
Logistic Regression	0.951	0.882 to 0.996	0.869	0.867	0.092
XGBoost	0.934	0.855 to 0.989	0.869	0.867	0.098

Table 3. Held-out test split, threshold 0.5, with percentile-bootstrap AUC intervals ($B = 1000$, `results/metrics.csv`). The interval widths near 0.10 are why Table 2, not this table, is the headline.

Comparison	ΔAUC	Z	p (two-sided)
Logistic Regression vs. Random Forest	−0.0022	−0.215	0.83
Logistic Regression vs. XGBoost	+0.0173	0.982	0.33
Logistic Regression vs. Neural Network	−0.0065	−0.777	0.44
Random Forest vs. XGBoost	+0.0195	1.290	0.20
Random Forest vs. Neural Network	−0.0043	−0.600	0.55
XGBoost vs. Neural Network	−0.0238	−1.368	0.17

Table 4. Pairwise DeLong tests on the held-out test split (`results/delong_tests.csv`). No pair of architectures is statistically distinguishable.

> **Finding 1, architectural parity.** Across both evaluation regimes the four architectures are statistically indistinguishable: CV AUC spans 0.890 to 0.912 with overlapping ±SD bands, and no DeLong test rejects equality (all $p \geq 0.17$). Model complexity does not convert into discrimination on this cohort.

> **Finding 2, the interpretable model wins on points.** Logistic regression posts the best cross-validated AUC (0.912 ± 0.020). Given Finding 1 this is best read as "is not worse," which is the deployment-relevant conclusion: where interpretability and auditability matter, nothing here justifies a black box.

Finding 3, XGBoost underperforms its reputation. The strongest method on large tabular benchmarks posts the lowest point estimates in both regimes (CV 0.890 ± 0.017). The gap is not significant ($p \geq 0.17$), but the direction cautions against assuming benchmark superiority transfers to this setting.

Finding 4, calibration is adequate but imperfect. Brier scores range 0.084 to 0.098 on the test split; reliability curves (Figure 2) track the diagonal with mid-range deviations, as expected for uncalibrated classifiers at this scale. Predicted probabilities should be read as approximate.

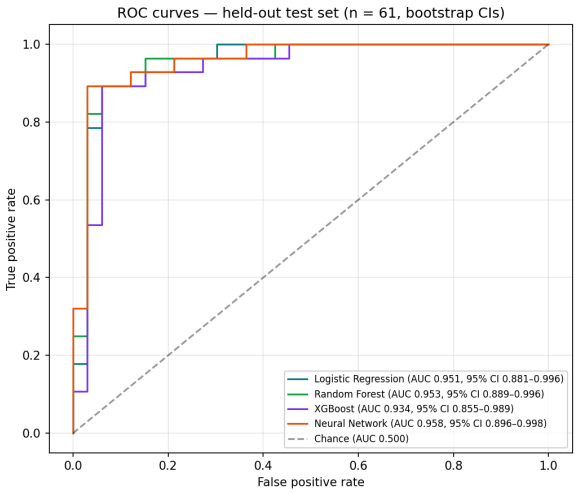

Figure 1. ROC curves on the held-out test split, with AUC and bootstrap CI per model (`results/roc_curves.png`).

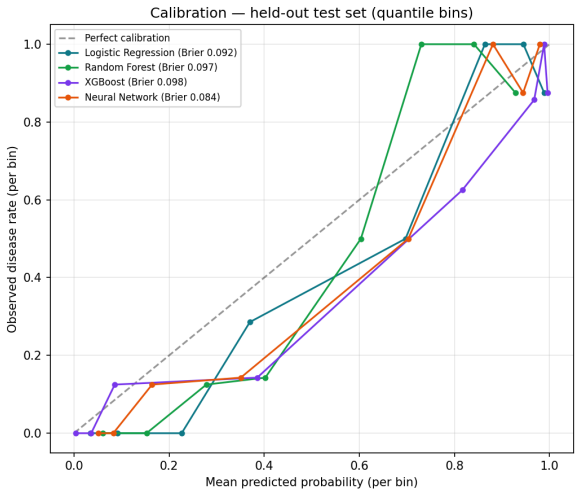

Figure 2. Reliability curves (quantile bins) with Brier scores (`results/calibration.png`).

6. Explainability

6.1 Methods

Per-prediction attributions use the exact SHAP algorithm appropriate to each family (TreeSHAP for RF and XGBoost, LinearSHAP for LR) and model-agnostic KernelSHAP for the neural network. The SHAP value of feature i is its average marginal contribution over all feature subsets $S \subseteq F \setminus \{i\}$,

$$\phi_i(f, x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,\big(|F| - |S| - 1\big)!}{|F|!} \Big[f\big(x_{S \cup \{i\}}\big) - f\big(x_S\big) \Big],$$

and the resulting attributions satisfy the efficiency (local accuracy) axiom, decomposing the prediction exactly into a base value plus per-feature terms,

$$f(x) = \mathbb{E}[f(X)] + \sum_{i=1}^{p} \phi_i(f, x).$$

The platform additionally computes LIME (a local linear surrogate fit to 600 perturbations) and, for the network, gradient-times-input saliency, and reports the Spearman rank correlation and top-3 overlap between methods so users can see where the explanations themselves agree.

6.2 Global feature importance

Ranked by mean $|\phi_i|$ on the test split, averaged across the four models (`results/feature_importance.json`): fluoroscopy vessel count (`ca`, mean $|\phi|$ 0.65) dominates, followed by the thallium scan (`thal`, 0.48) and chest-pain type (`cp`, 0.37). The vessel count leads because it is a direct anatomical measurement of the target condition. Asymptomatic presentation (`cp=4`) associates with *higher* predicted risk, reflecting referral patterns in this cohort: patients without pain are catheterized only when other evidence is strong. Serum cholesterol ranks in the bottom third (mean $|\phi|$ 0.17). The narrow reading is that total cholesterol, without HDL or LDL fractionation, adds little to predicting *prevalent angiographic disease* once direct anatomical and functional test results are available; this says nothing about cholesterol's established role in long-term incident risk.

7. Clinical Context Layer

Beside the ML estimate the dashboard displays the 10-year general-cardiovascular-risk estimate of the office-based Framingham profile (D'Agostino et al., 2008). For sex-specific coefficients β_j acting on the log-transformed risk factors, the linear predictor and 10-year risk are

$$L = \sum_j \beta_j \ln(x_j), \qquad \widehat{P}_{10} = 1 - S_0^{\,\exp(L - \bar{L})},$$

with baseline survival S_0 and population mean $\bar{L}$ taken from the published model. The unavailable inputs are filled with stated assumptions (HDL 45 mg/dL, non-smoker, untreated blood pressure; diabetes proxied by the fasting-blood-sugar flag). The two numbers answer *different questions*: the ML models estimate the probability of prevalent obstructive disease, while $\widehat{P}_{10}$ estimates 10-year incident-event risk. The platform therefore presents the comparator as context and does not score one against the other.

A separate guideline advisory handles inputs beyond the training distribution. The Cleveland data contains no blood pressure above 200 mm Hg or cholesterol above 564 mg/dL, so tree models saturate and can report low risk for clinically extreme values. When standard thresholds (ACC/AHA hypertension stages; NCEP ATP III lipid bands) imply more risk than the displayed estimate, the dashboard says so in a labeled banner. The model outputs themselves are never modified.

8. Federated-Learning Simulation

As a privacy-architecture demonstration, `src/federated.py` implements FedAvg (McMahan et al., 2017) over three simulated sites that exchange only model weights. Each round, site *k* trains locally for several epochs and the server forms a size-weighted average of the local parameters,

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{(k)}, \qquad n = \sum_k n_k,$$

over 25 communication rounds and 5 local epochs. The aggregated network converges to test AUC 0.950 (IID partition) and 0.960 (non-IID) versus 0.952 for a centrally trained control. These differences are within sampling noise; the result demonstrates *convergence without raw-data exchange*, not superiority of either regime. As a single-machine simulation it says nothing about real cross-institutional deployment (systems heterogeneity, stragglers, differential privacy) beyond the algorithm's mechanics.

9. Platform

The Flask backend loads the four models, the five cross-validation fold models per architecture with their per-fold preprocessors, and cached SHAP explainers at startup, and exposes a JSON API. Inputs are validated against physiological bounds and allowed categorical codes. For each prediction the API returns the four unmodified probabilities and their ensemble mean; a fold-spread stability range (min to max and mean ± SD of the five fold models' outputs, each computed through its own fold's preprocessing) explicitly labeled as a stability indicator rather than a confidence interval; the agreement level across architectures; SHAP contributions; the D'Agostino comparator; an age and sex population percentile; and the guideline advisory. The frontend (vanilla JavaScript, Chart.js, jsPDF, Leaflet) provides real-time re-scoring, a what-if simulator, a SHAP/LIME/gradient comparison view, an age-conditional projection view (labeled a what-if, since the cross-sectional target cannot support survival estimates), local-only patient history, a structured PDF summary, and a model card. Strings from third-party APIs (OpenStreetMap) are HTML-escaped before rendering.

10. Limitations

- **Prevalent, not incident, disease.** The label is "disease present at catheterization," so several predictors (`oldpeak`, `exang`, `ca`, `thal`) are partly manifestations of the target, and referral bias shapes the feature-label relationships (see the `cp` finding). The models do not predict future events.
- **Sex imbalance.** The cohort is roughly two-thirds male, so estimates for women rest on a smaller subgroup and are correspondingly less certain.
- **Missing modalities.** No HDL or LDL fractionation, smoking status, family history, BMI, or medication data, all of which carry cardiovascular information the model cannot use.
- **Out-of-distribution inputs.** The dashboard permits values beyond the training range; predictions there are extrapolations, which is exactly what the guideline advisory exists to flag.
- **Calibration.** Probabilities are only approximately calibrated (Brier 0.084 to 0.098); a post-hoc calibration step (Platt scaling or isotonic regression) would be a natural extension.

11. Conclusion

Four architectures spanning the bias-variance spectrum are statistically indistinguishable on this cohort, and the most interpretable of them attains the best cross-validated discrimination (AUC 0.912 ± 0.020). The practical contribution is less the number than the apparatus around it: per-fold leak-free evaluation, bootstrap intervals and DeLong tests instead of bare point estimates, per-prediction attribution by three methods, stability ranges instead of pseudo confidence intervals, and a strict separation between what the models say and what clinical guidelines would add. For clinical machine learning at this scale we take the result as support for a simple default: start linear, demand that complexity prove itself, and ship the uncertainty with the score.

11. References

1. Detrano, R., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64(5), 304 to 310.
2. Lundberg, S. M., and Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
3. Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD*, 1135 to 1144.
4. D'Agostino, R. B., et al. (2008). General cardiovascular risk profile for use in primary care: the Framingham Heart Study. *Circulation*, 117(6), 743 to 753.
5. DeLong, E. R., DeLong, D. M., and Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3), 837 to 845.
6. Sun, X., and Xu, W. (2014). Fast implementation of DeLong's algorithm for comparing the areas under correlated receiver operating characteristic curves. *IEEE Signal Processing Letters*, 21(11), 1389 to 1393.
7. Efron, B., and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.
8. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5 to 32.
9. Chen, T., and Guestrin, C. (2016). XGBoost: a scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD*, 785 to 794.
10. McMahan, H. B., et al. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of AISTATS*, 1273 to 1282.
11. Kingma, D. P., and Ba, J. (2015). Adam: a method for stochastic optimization. *ICLR*.
12. Srivastava, N., et al. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929 to 1958.