



# Machine Learning as Your Analytical Toolkit

Uncovering what your data is telling you

Victoria Ponce, PhD  
Associate Director, Pharmacometrics  
Certara

ISoP AI/ML in Pharmacometrics: Hands-on Workshop and Regulatory Panel  
34th PAGE Meeting, Dubrovnik, Croatia  
June 2, 2026

# Why ML in R?

---

- R is where your pharmacometrics workflow lives (NONMEM post-processing, `mrgsolve` , `nlmixr2` simulation outputs, logistic regresion or TTE analysis ...)
- R has a mature, often underestimated ML ecosystem
- The barrier is lower than you think
- CRAN: 20,000+ packages, many ML-focused

Application	ML Approach	R Packages
Covariate discovery	XGBoost + SHAP	<code>xgboost</code> , <code>shapviz</code>
Toxicity prediction	Gradient boosting, Random forest	<code>tidymodels</code> , <code>ranger</code>
Clustering patients / ETAs	UMAP + PAM	<code>umap</code> , <code>cluster</code>
PK parameter estimation	Neural ODEs	<code>torch</code> , <code>neuralode</code>
Hybrid mechanistic-ML	Deep compartment models	<code>torch</code> , <code>pknodel</code>
Exposure–response	BART, GAMs	<code>BART</code> , <code>mgcv</code>
Missing data imputation	Random forest MI	<code>mice</code> , <code>missRanger</code>
Unified ML interface	-	<code>tidymodels</code>

# Session focus

---

1. 🎯 **Analytical goals**, Three PMx questions, one ML toolkit
2. 🛠️ **Tools overview**, XGBoost, SHAP, UMAP, PAM at a glance
3. 📊 **Back to workshop dataset**, tobramycin PK/PD simulated data recap
4. 🔗 **Tools in action**, applying the tools across ETA, SHAP, and residual spaces
5. ⚠️ **Caveats**, limitations and methodological considerations
6. 💻 **Let's code!**, hands-on session

## By the end of this session you will...

- Understand how ML clustering reveals structure in PK/PD data
- Apply XGBoost + SHAP for covariate importance
- Use UMAP + PAM across three input spaces
- Know when (and when *not*) to trust these methods

# What can ML help us answer?

---

## Driver interpretation

Which covariates matter most in predicting individual drug response?\*

## Patient phenotyping

*Can we identify subgroups of patients with distinct response profiles?*

## Model diagnostics

*Is model misfit random, or do mispredicted records share systematic patterns?*

# Tools overview: XGBoost

## What XGBoost does

- Builds an ensemble of decision trees sequentially, each targeting the **residual errors** from the previous ensemble
- Uses **gradient descent in function space**: each tree approximates the negative gradient of the loss
- Fast to train; robust to outliers and missing covariates
- **Shrinkage** ( `learn_rate` ) scales each tree’s contribution: smaller values require more trees but improve generalisation

## Key hyperparameters

Parameter	Controls	Typical range
<code>trees</code>	Number of boosting rounds	100–1 000
<code>tree_depth</code>	Complexity per tree	3–6
<code>learn_rate</code>	Step size (shrinkage)	0.01–0.1
<code>mtry</code>	Features sampled per split	0.5–1.0

### When NOT to use XGBoost

- **Small n**: unstable with < 50–100 samples, prefer penalised regression or simpler models
- **Extrapolation**: tree ensembles cannot predict outside the range of the training data
- **Inference**: no p-values or confidence intervals, XGBoost is predictive, not inferential

# Tools overview: SHAP

---

## Main concepts

- Most popular feature-interpretability method
- Rooted in **cooperative game theory** (Shapley values): each feature is a “player” sharing credit for the prediction
- Decomposes each prediction into per-variable contributions
- Captures non-linearity and interactions between features

## Key properties

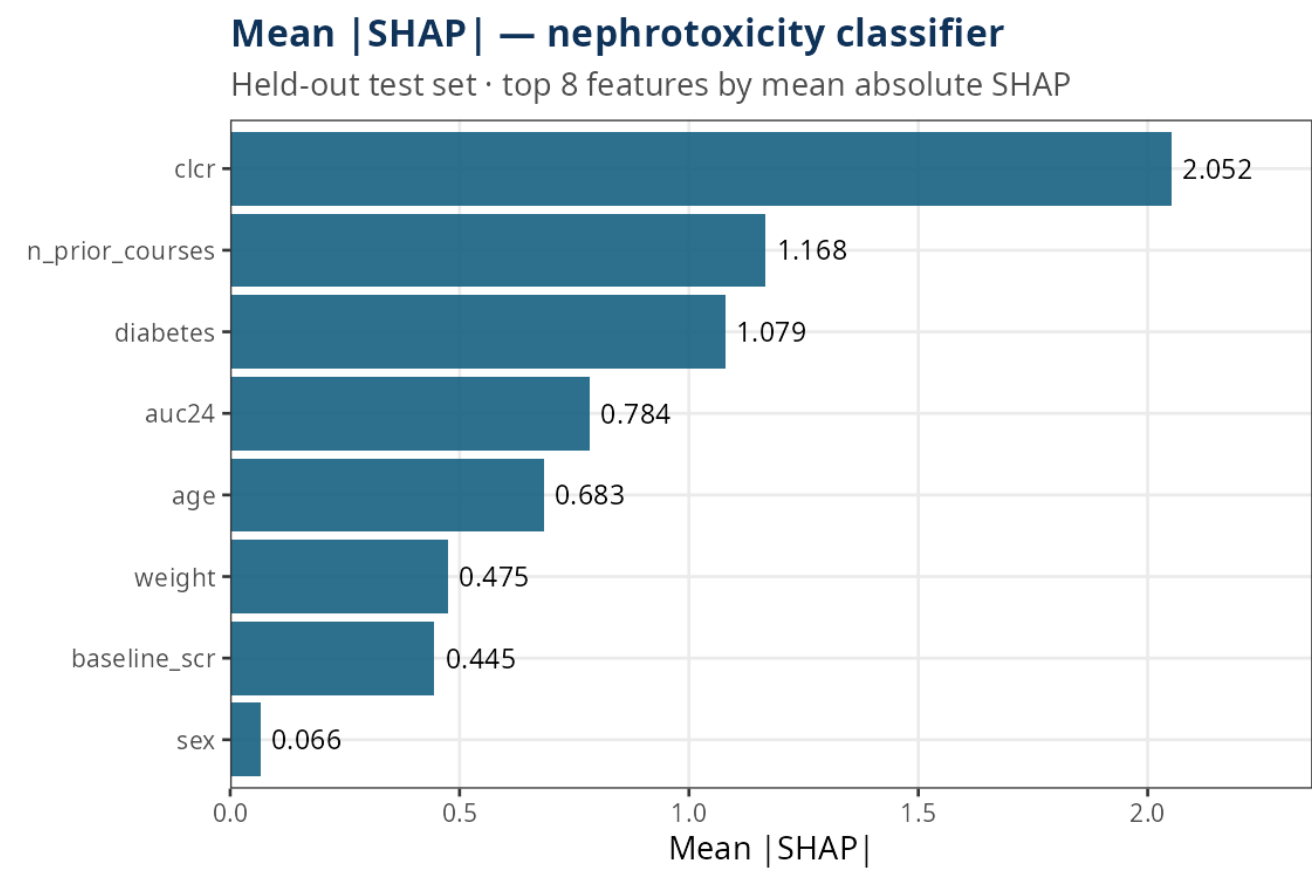
- **Additivity**, SHAP values sum to prediction – baseline (mean prediction)
- **Consistency**, if a feature contributes more to the prediction, its SHAP value is always higher
- **Local accuracy**, explains every individual prediction, not just averages

### When NOT to trust SHAP

- **Correlated features**: credit is split arbitrarily among correlated predictors, inspect pairs, not individuals
- **Causality**: high SHAP = predictive under *this* model, not a causal driver
- **Small n**: estimates become unstable below ~100 observations

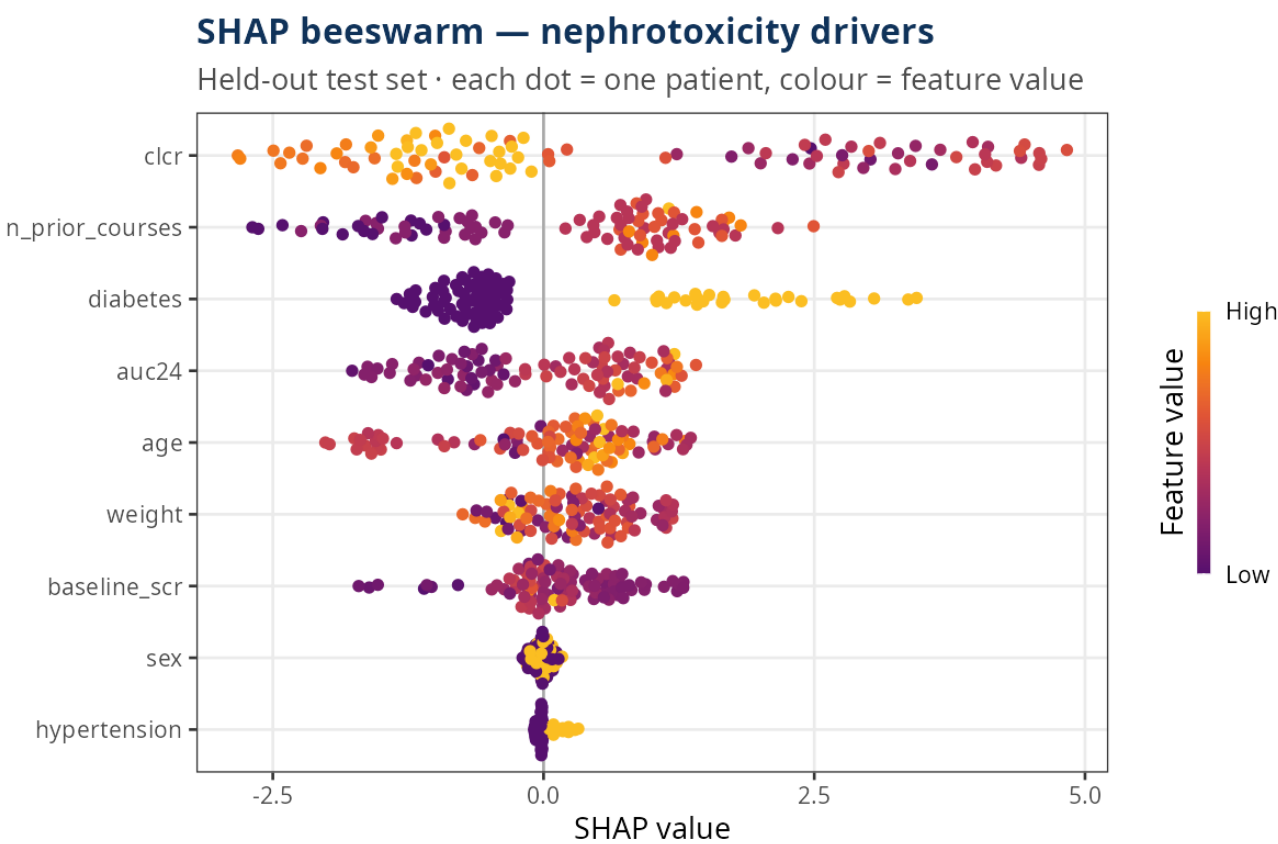
# Tools overview: SHAP, population plots

## Mean |SHAP| bar



Top features ranked by **mean absolute SHAP**. Quick global importance summary; complements the beeswarm by collapsing direction and spread into a single magnitude.

## Beeswarm



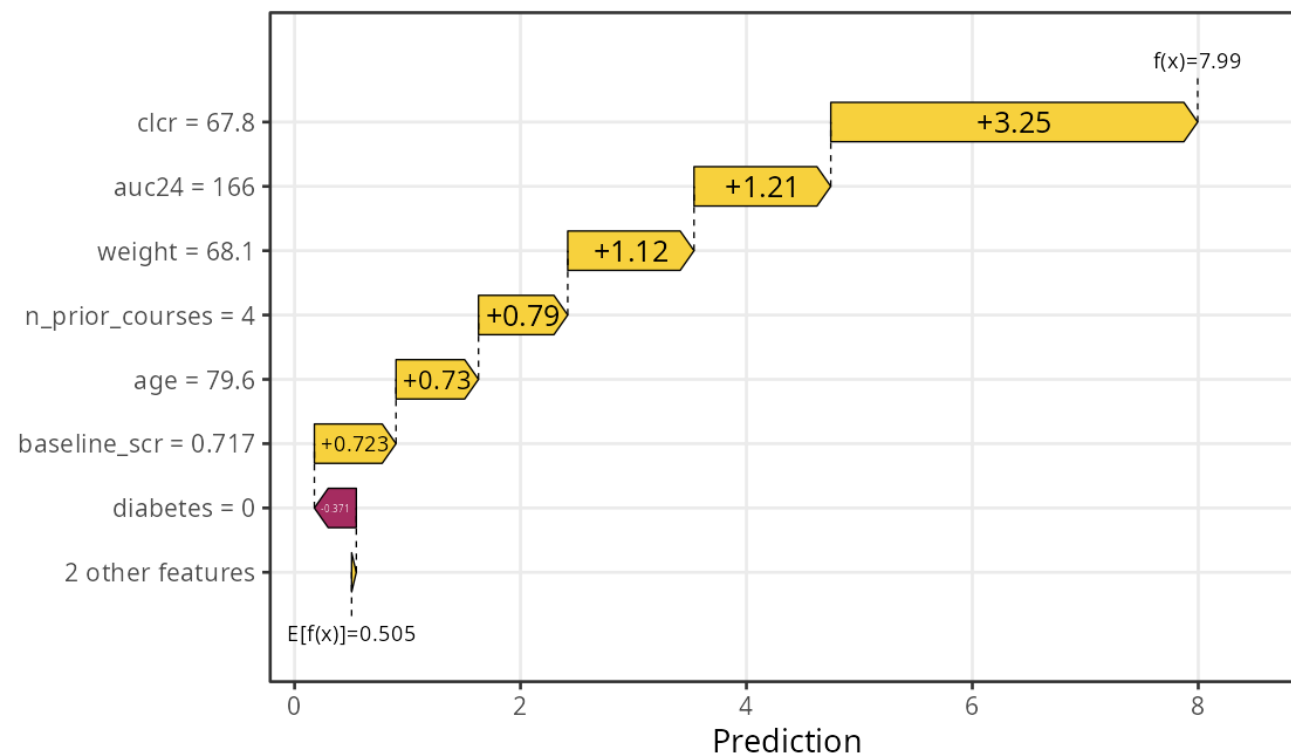
Each dot = one patient · colour = feature value (red high, blue low) · x = SHAP. Ranks features and reveals the **direction** and **heterogeneity** of each feature's effect across the cohort.

# Tools overview: SHAP, individual explanations and interactions plots

## Waterfall

### SHAP waterfall — single patient

Patient ID 196 · highest predicted nephrotoxicity risk (test set)

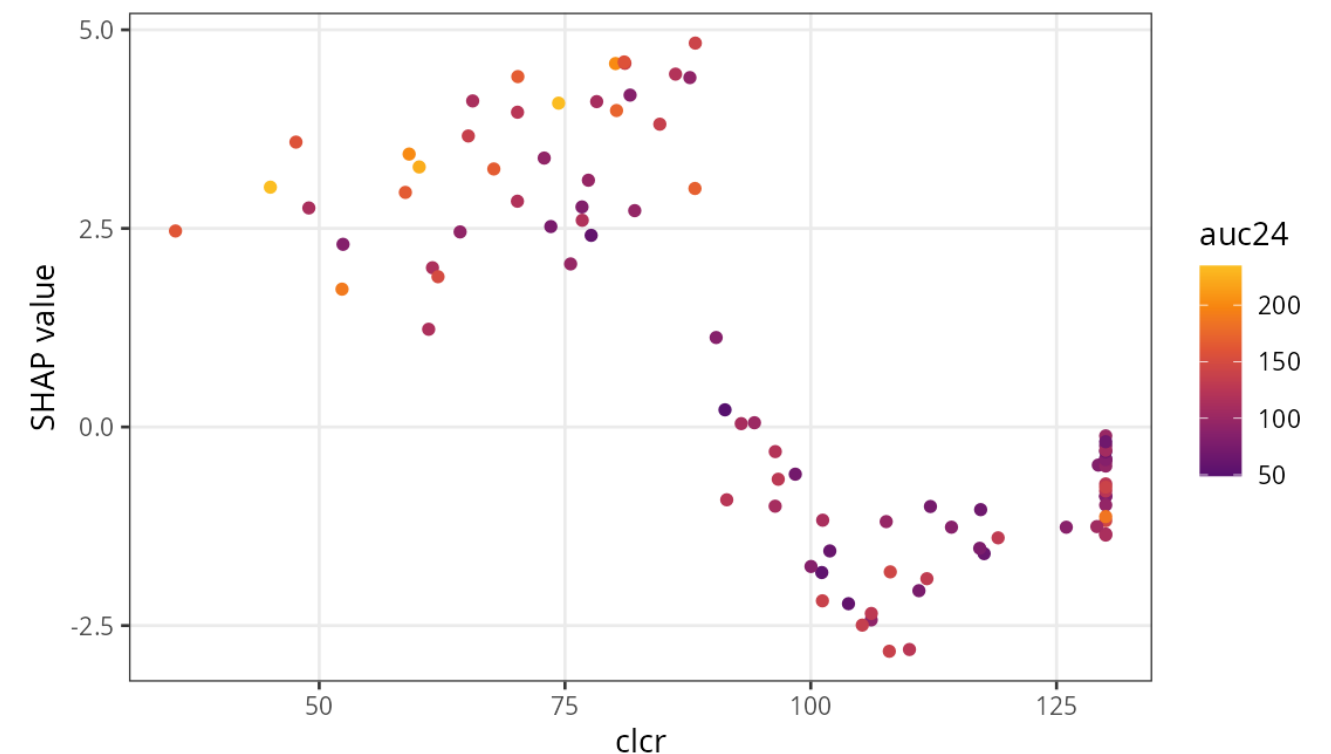


Single-patient decomposition. Each bar = one feature's push above or below the mean prediction, ordered by magnitude. Useful for **explaining a specific flagged patient** to a clinician.

## Dependence plot

### SHAP dependence — CLCR coloured by 24-h AUC

Negative slope on CLCR · clearance × exposure interaction



One feature's value (x) vs its SHAP (y), coloured by a second feature. Reveals **non-linear effects** and **pairwise interactions**, e.g.,  $ETA\_CL \times CLCR$ .



# Tools overview: Dimensionality Reduction Methods

## What UMAP does

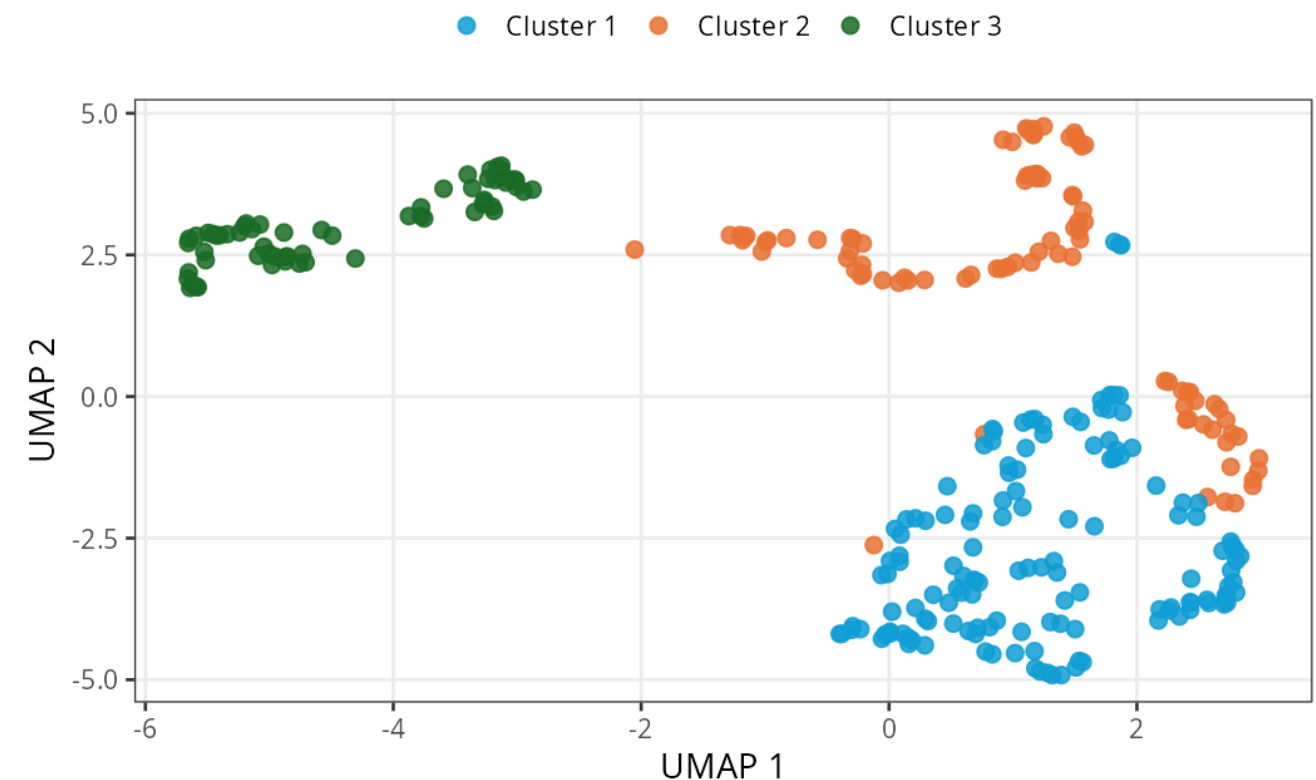
- Constructs a weighted nearest-neighbour graph in high-dimensional space, then finds a 2D layout that preserves it
- Preserves local cluster structure better than PCA, clusters that are real show up as islands
- Non-linear and tunable: `n_neighbors` controls local vs global focus; `min_dist` controls compactness

```
1 library(uwot)
2
3 # Project ETA space to 2D
4 emb <- umap(eta_matrix,
5   n_neighbors = 15,
6   min_dist    = 0.1,
7   metric      = "euclidean"
8 )
```

## Main plot: 2D embedding

### UMAP of SHAP space — three risk-driver clusters

Same 300 patients · colour = hidden true cluster



# Other dimensionality reduction methods

UMAP is one of several dim-reduction tools. The right pick depends on whether you care about **local clusters**, **global geometry**, or **interpretable axes**.

Method	How it works	Strengths	Weaknesses
UMAP	Non-linear; preserves nearest-neighbour graph	Local + some global structure; fast	Stochastic; sensitive to <code>n_neighbors</code>
PCA	Linear projection onto max-variance axes	Deterministic; interpretable axes; very fast	Misses non-linear cluster structure
t-SNE	Non-linear; preserves local neighbours via probability matching	Strong cluster separation in 2D	Slow at scale; no global structure; very stochastic
Autoencoders	Neural network compression to a learned latent space	Flexible non-linear; reusable encoder	Needs more data; harder to tune and interpret

Rule of thumb

Start with **PCA** for a quick linear sanity check, then move to **UMAP** when you suspect non-linear cluster structure. Reach for **autoencoders** only when you have lots of data and need a reusable embedding.

# Tools overview: Clustering

## What PAM does

- Partition Around Medoids, assigns each patient to the cluster whose representative (medoid) is closest
- Unlike k-means, the medoid is a **real patient** in your dataset → clinically interpretable
- Silhouette width quantifies how well each patient fits its cluster (range -1 to 1; higher = better fit)

```
1 library(cluster)
2
3 pam_fit <- pam(umap_coords, k = 3)
4 sil <- silhouette(pam_fit)
5 plot(sil) # silhouette plot
6 pam_fit$medoids # the representative patients
```

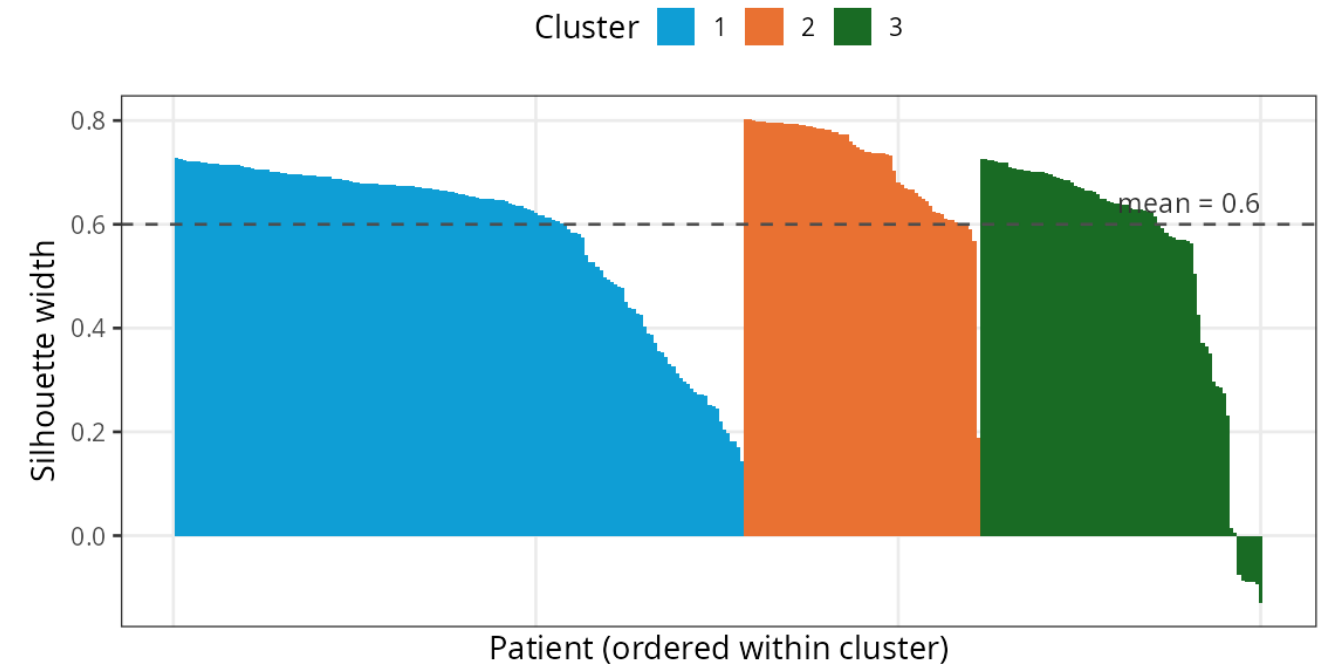
### How to choose $k$

Run PAM for  $k = 2-6$  and plot average silhouette width vs  $k$ . Pick the  $k$  with the highest silhouette score that still makes clinical sense.

## Main plot: silhouette width

### PAM silhouette widths ( $k = 3$ )

SHAP-space UMAP · sorted within cluster



# Other clustering methods

PAM is our default for **interpretability** (medoid = real patient). Switch to **HDBSCAN** when cluster count is unknown or noise points matter; switch to **hierarchical** when you need a dendrogram.

Method	Approach	Strengths	Weaknesses
<b>PAM</b>	Medoid-based; fixed $k$	Robust to outliers; medoid = real patient	$k$ set a priori; slower than k-means
k-means	Centroid-based; fixed $k$	Fast; simple	Spherical clusters only; centroid is not a real patient
Hierarchical	Builds a dendrogram of nested groups	No $k$ needed upfront; dendrogram is interpretable	$O(n^2)$ memory; sensitive to linkage choice
HDBSCAN	Density-based; finds noise points	No $k$ needed; arbitrary cluster shapes; flags outliers	Sensitive to <code>min_cluster_size</code> ; non-deterministic borders

**Pair with the right embedding**

Density-based methods (**HDBSCAN**) shine on UMAP embeddings where clusters are well separated; partitioning methods (**PAM**, **k-means**) work fine on the raw scaled features. Always inspect silhouette or stability before committing to  $k$ .

# Back to workshop dataset: Tobramycin PD

## What you have, `tobramycin_pd.csv`

Group	Columns
Patient covariates	<code>age</code> , <code>weight</code> , <code>sex</code> , <code>clcr</code> , <code>baseline_scr</code> , <code>diabetes</code> , <code>hypertension</code> , <code>n_prior_courses</code>
Posthoc PK params	<code>cl</code> , <code>v1</code> , <code>q</code> , <code>v2</code>
ETAs	<code>eta_cl</code> , <code>eta_v1</code> , <code>eta_q</code> , <code>eta_v2</code>
Exposure metrics	<code>auc24</code> , <code>cumulative_auc</code> , <code>cmin</code> , <code>cmax_central</code> , <code>cmax_peripheral</code>
PD outcome (nephrotoxicity)	<code>nephro_binary</code> , <code>nephro_risk_score</code> , <code>peak_delta_scr</code>

300 simulated patients, once-daily IV tobramycin × 10 days. PK fitted upstream; what follows uses these per-subject features.

## What we will do this session

- **Who is at risk, and why?** → XGBoost + SHAP on `nephro_binary`
- **Which patients behave alike?** → UMAP + PAM across posthoc PK / outcome / SHAP spaces

Same 300 patients, three ML lenses

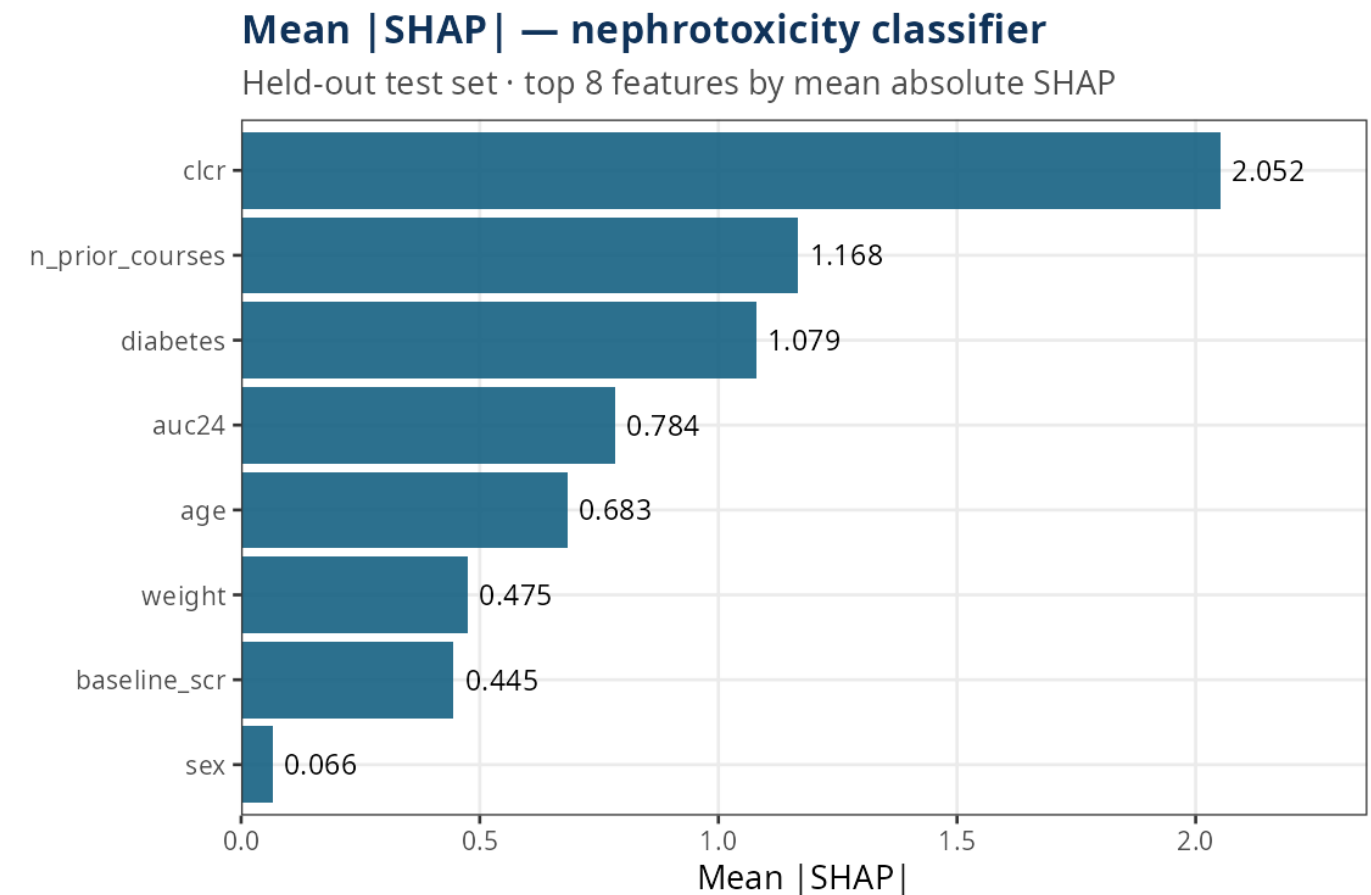
The hidden `true_cluster` column is revealed only at the end of the session.

# XGBoost + SHAP: Toxicity Risk Drivers

## Fitting the model

```
1 library(tidymodels)
2 library(shapviz)
3 # 70/30 split, stratified on the outcome
4 split <- initial_split(df_xgb, prop = 0.70,
5                         strata = nephro_binary)
6 df_train <- training(split)
7 df_test <- testing(split)
8
9 # Tune tree_depth x learn_rate on training rows,
10 # then refit on all training rows with the winning cell
11 xgb_out <- tune_xgb_classification(
12   df_train,
13   nephro_binary ~ age + weight + sex + clcr +
14                   baseline_scr + diabetes +
15                   hypertension + n_prior_courses + auc24
16 )
17 xgb_fit <- xgb_out$fit
18 # SHAP on the held-out test rows
19 shp <- shapviz(extract_fit_engine(xgb_fit),
20               X = X_test)
21 sv_importance(shp, kind = "beeswarm")
```

## What the SHAP output tells us



Clinical read, **clcr** dominates, with **n\_prior\_courses**, **diabetes** and **auc24** close behind — the classifier picks up the clearance × exposure × comorbidity story.

# UMAP + PAM: Patient Phenotyping

## Two tools, one pipeline

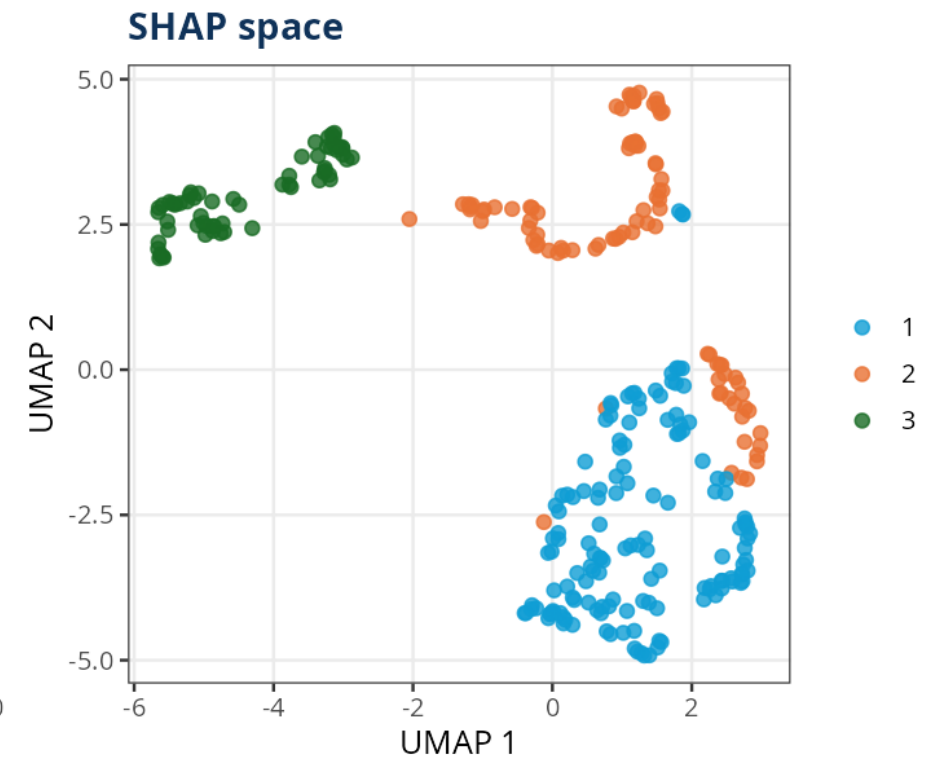
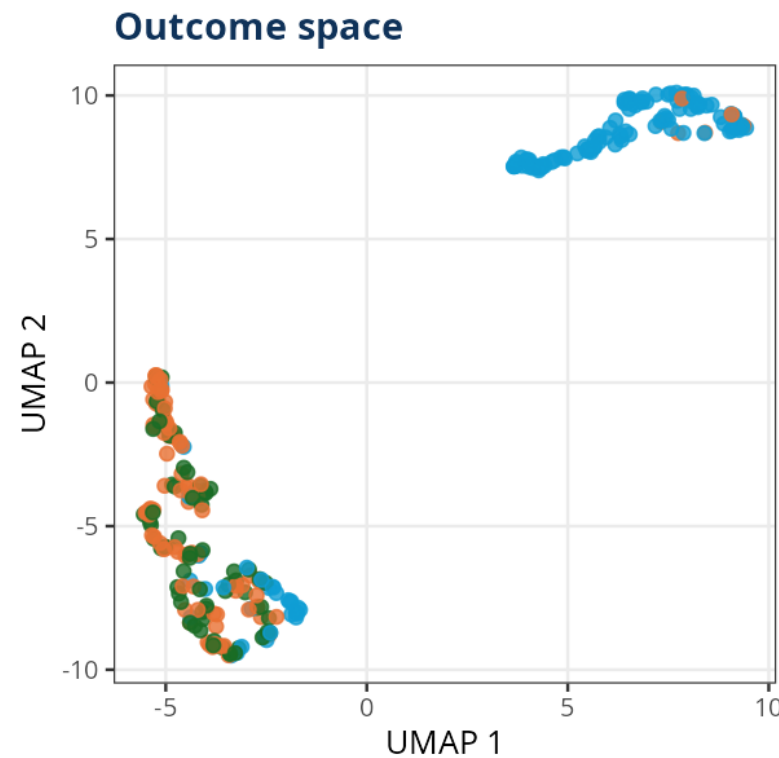
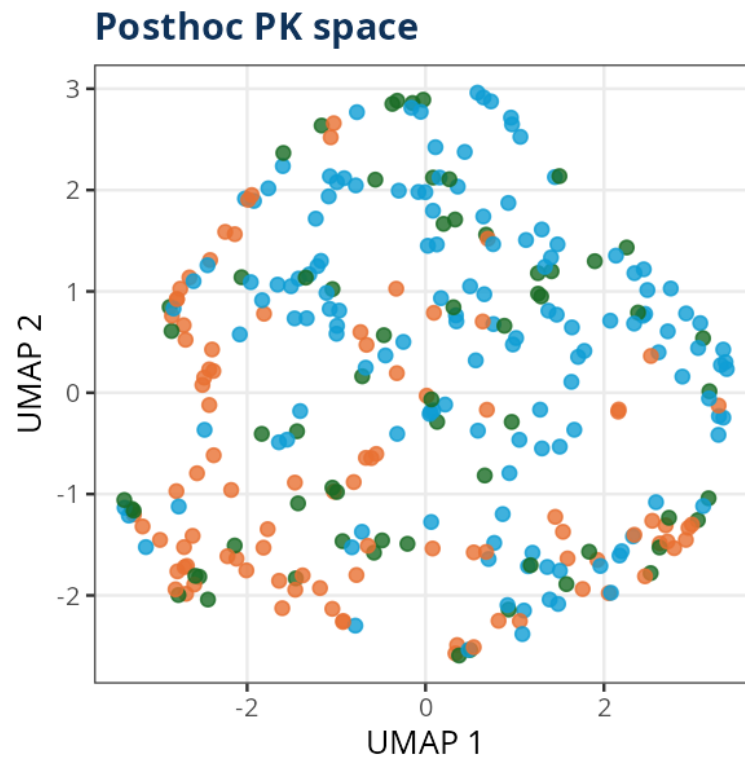
UMAP, Takes your high-dimensional ETA/SHAP space → 2D Better than PCA at preserving cluster structure

PAM, Robust clustering: Assigns patients to groups Gives you a real patient as cluster representative (the *medoid*)

## Three input spaces, three questions

**The reveal · colour = hidden true\_cluster**

Cluster colours mix in posthoc PK and outcome space — they line up in SHAP space



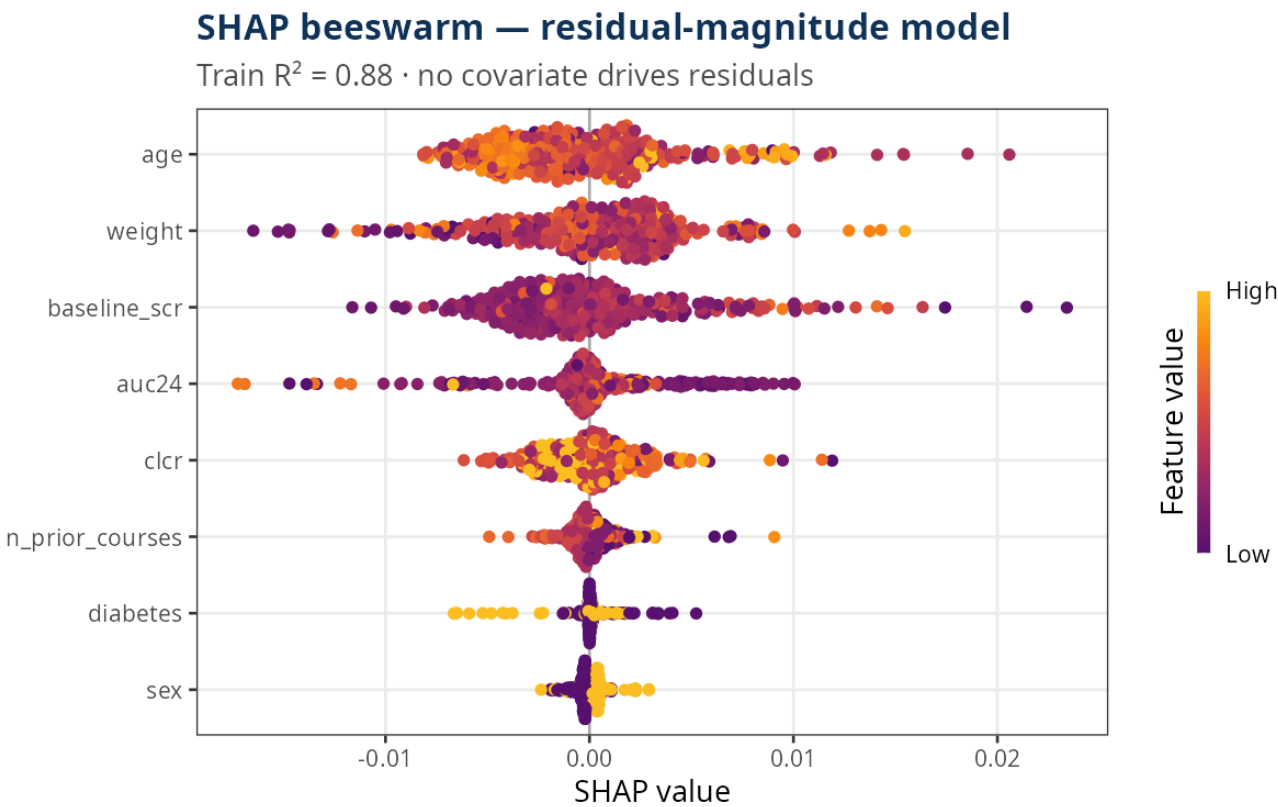
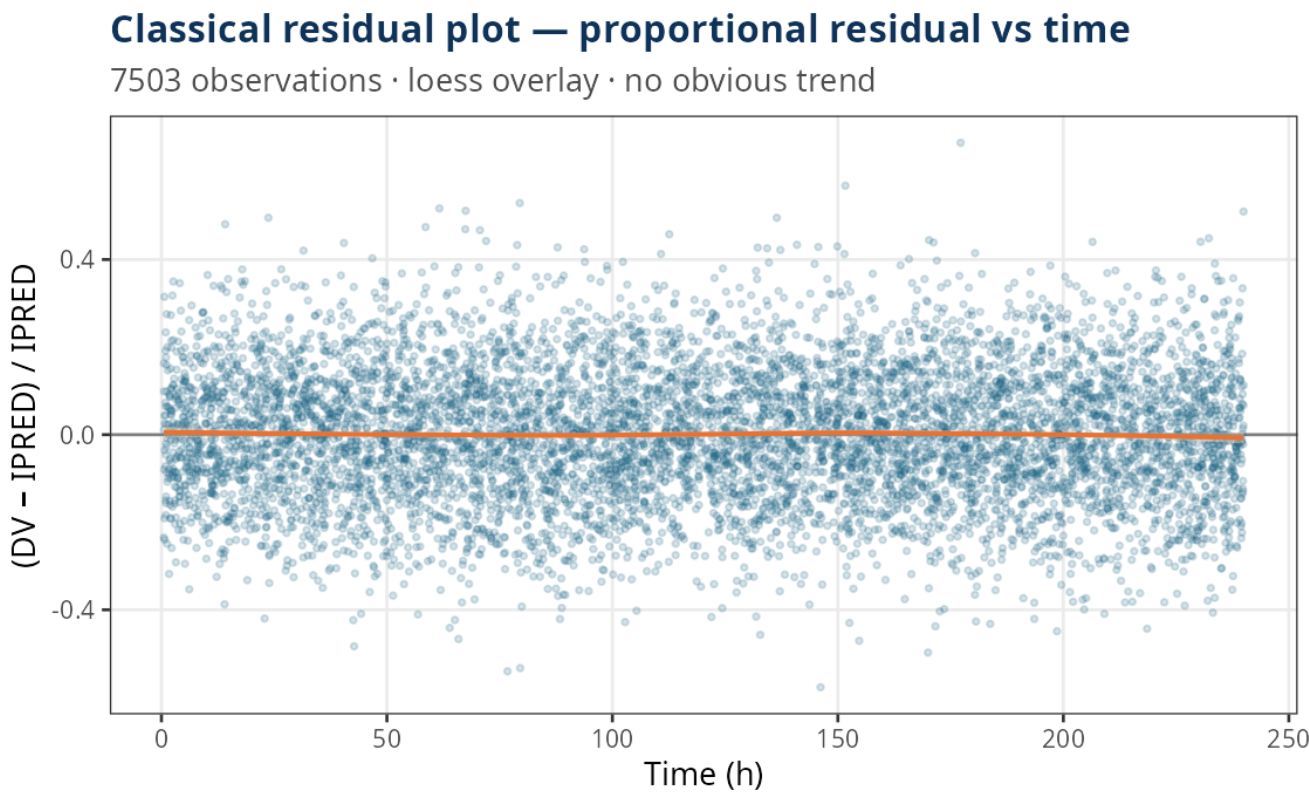
# ML-Assisted Model Diagnostics

Classical CWRES vs TIME shows one variable at a time, interaction-driven misfit is invisible in marginal plots.

The ML alternative: train XGBoost on |residuals|; if it predicts well, SHAP tells you what your structural model missed.

Classical plot, looks fine, no obvious trend

XGBoost on residuals, null result here; the *method* is what travels





# Caveats

---

## Correlation $\neq$ causation

SHAP shows which variables are **predictive** under the model. It does not show what **causes** toxicity. High SHAP value  $\rightarrow$  predictive, not mechanistic.

## Clusters are hypotheses

Validate against biology, clinical outcome, domain knowledge. Never report clusters without external validation.

## Overfitting is real

Use repeated cross-validation. Report metrics on **held-out** data, not training data.

# Let's code

---

Hands-on session begins now