Module 8: Machine learning for clinical prediction

Data analysis with Python for health specialists

Yaé Ulrich Gaba

2026

AIRINA Labs

ML vs. traditional statistics

Traditional statistics:

"Which risk factors are associated with heart disease, and by how much?"

Goal = inference
Tool = logistic regression
Output = odds ratios, p-values

Machine learning:

"Given this patient's data, what is their probability of heart disease?"

Goal = prediction
Tool = random forest, etc.
Output = predicted probability

In practice

Many clinical studies use **both**: regression to understand the biology, ML to build the screening tool.

When NOT to use ML

- **Small datasets** (< 200 patients): ML overfits easily. Stick with logistic regression.
- **Regulatory approval:** FDA requires interpretable models for most clinical decision support.
- **Simple rules work:** If "glucose > 126" classifies 90% of diabetics, you do not need a random forest.

Complexity must be justified by performance gain.

Train/test split and cross-validation

```python
from sklearn.model_selection import (train_test_split,
                                     cross_val_score)

# 70/30 split (stratified for class balance)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y)

# 5-fold cross-validation (more robust)
scores = cross_val_score(model, X, y, cv=5,
                         scoring="accuracy")
print(f"CV accuracy: {scores.mean():.3f} +/- {scores.std():.3f}")
```

Golden rule

Never use test set results to choose your model. The test set is used **once**, at the very end.

Decision trees: clinical flowcharts

```
from sklearn.tree import DecisionTreeClassifier, plot_tree

dt = DecisionTreeClassifier(max_depth=4, random_state=42)
dt.fit(X_train, y_train)

# Visualize the tree
fig, ax = plt.subplots(figsize=(20, 10))
plot_tree(dt, feature_names=feature_cols,
          class_names=["No Disease", "Disease"],
          filled=True, rounded=True, ax=ax)
```

Clinical relevance

Decision trees produce **clinical decision rules** — flowcharts a physician can follow at the bedside. The HEART score, Wells criteria, and Ottawa ankle rules are conceptually similar.

Random forests: ensemble power

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=200, max_depth=6,
                            random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

print(classification_report(y_test, y_pred,
      target_names=["No Disease", "Disease"]))
```

A random forest = **200 decision trees**, each on a random subset. Final prediction = majority vote. More accurate, less interpretable.

ROC curves and AUC

```python
from sklearn.metrics import roc_curve, roc_auc_score

y_prob = rf.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(y_test, y_prob)
auc = roc_auc_score(y_test, y_prob)

ax.plot(fpr, tpr, label=f"RF (AUC = {auc:.3f})")
ax.plot([0, 1], [0, 1], "k--", label="Random (0.500)")
```

AUC	Discrimination
0.5	No better than a coin flip
0.7–0.8	Acceptable
0.8–0.9	Excellent
>0.9	Outstanding (rare in practice)

Feature importance

Q: Which risk factors matter most for prediction?

```
importances = rf.feature_importances_
imp_df = pd.DataFrame({
    "Feature": feature_cols,
    "Importance": importances
}).sort_values("Importance", ascending=True)

ax.barh(imp_df["Feature"], imp_df["Importance"])
ax.set_xlabel("Feature importance (Gini)")
```

Correlation $\neq$ Causation
Feature importance tells you what is useful for prediction, not what causes the outcome. A proxy variable can rank highly.

Calibration: do probabilities match reality?

```
from sklearn.calibration import calibration_curve
from sklearn.metrics import brier_score_loss

prob_true, prob_pred = calibration_curve(y_test, y_prob,
                                         n_bins=8)
ax.plot(prob_pred, prob_true, "o-", label="Random Forest")
ax.plot([0, 1], [0, 1], "k--", label="Perfect")

brier = brier_score_loss(y_test, y_prob)
print(f"Brier score: {brier:.4f}")  # lower = better
```

If a model says “70% risk,” do 70% of those patients actually have the disease? Calibration is critical for clinical deployment.

Overfitting: the central danger

```
# Unlimited depth -> memorizes training data
dt_overfit = DecisionTreeClassifier()  # no max_depth
dt_overfit.fit(X_train, y_train)

print(f"Train acc (overfit): {dt_overfit.score(X_train,
↪  y_train):.3f}")
print(f"Test acc (overfit):  {dt_overfit.score(X_test,
↪  y_test):.3f}")
# 1.000 vs 0.72 -> overfitting!
```

Signs: training $\gg$ test accuracy, wild CV fold variation, more features hurt.

Cure: more data, simpler models, regularization, cross-validation.

Ethics: fairness in clinical ML

Before deploying any clinical ML model, ask:

- **Who is in the training data?** 90% white males → may fail for women and minorities.
- **Does performance differ across subgroups?** Check AUC by sex, race, age.
- **What are the consequences of errors?** A missed cardiac event in a young woman is dangerous.
- **Is the model transparent?** Clinicians and patients deserve explanations.

Real-world example
In 2019, a widely-used algorithm was found to be biased against Black patients — it used healthcare *costs* as a proxy for *needs*, systematically underestimating care requirements.

What you can do after this module

1. Split data properly and use cross-validation
2. Train decision trees, random forests, and logistic regression
3. Compare models using ROC curves and AUC
4. Identify top predictive features from a random forest
5. Assess calibration and detect overfitting
6. Audit models for fairness across demographic subgroups

Next: Module 9 — Geospatial and temporal health data

Exercises (Hour 3)

1. **Model comparison:** Train logistic regression, decision tree, and random forest; compare CV AUC
2. **Decision tree:** Visualize a depth-3 tree; can a clinician follow it?
3. **Top features:** Train RF with only top 5 features; compare AUC to full model
4. **Mini-project:** Build a complete heart disease prediction pipeline with ROC curves, feature importance, calibration plot, and fairness audit