

From Inference to Prediction

Chapters 7-9 emphasised inference: estimating and testing the size of an association, with an explicit statistical model whose coefficients are individually interpretable. Machine learning shifts emphasis toward prediction: building a model, however complex, whose primary virtue is that it predicts an outcome accurately on new data — for example, predicting which febrile patients are at high risk of severe malaria at triage, or which pregnant women are at elevated risk of a poor birth outcome, so that limited clinical attention or referral resources can be prioritised. The two goals are not mutually exclusive, but they favour different modelling choices, and a Nigerian health data scientist should be explicit at the outset about which goal a given project serves.

The Bias-Variance Tradeoff and Overfitting

Every predictive model faces a tradeoff between bias (systematic error from a model too simple to capture the true relationship) and variance (sensitivity to the specific sample used for training, such that the model performs well on training data but poorly on new data — overfitting). A model with too few parameters relative to the complexity of the underlying relationship underfits; a model with too many parameters relative to the amount of data available

memorises noise in the training set and generalises poorly. This tradeoff, more than any specific algorithm, is the central conceptual challenge of applied machine learning, and every technique introduced in this chapter and the next (regularisation, cross-validation, ensemble methods) is, at bottom, a strategy for managing it.

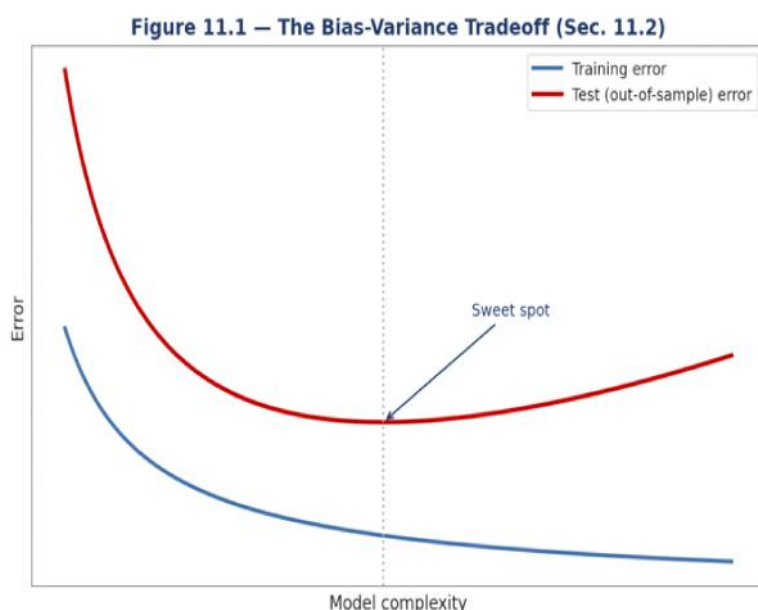


Figure 11.1 — The bias-variance tradeoff (Section 11.2).

Train/Test Splitting and Cross-Validation

A model's performance must always be assessed on data it was not fitted on. The simplest approach splits the available data into a training set (used to fit the model) and a test set (used only once, at the end, to report performance); k-fold cross-validation extends this by partitioning the data into k subsets, training on k-1 folds and validating on the remaining fold, rotating through all k folds and averaging performance — a more data-efficient approach favoured when the dataset is not large (common in Nigerian clinical cohort studies, where sample sizes may be in the hundreds rather

than the tens of thousands typical of large-scale digital datasets). When observations are naturally grouped (e.g., multiple patients per facility, or repeated measures per participant), cross-validation folds must be constructed by group, not by individual observation, or the reported performance will be optimistically biased by information leakage between the training and validation sets.

11.4 A Survey of Core Algorithms

- Regularised regression (ridge, LASSO, elastic net): extends linear/logistic regression by penalising large coefficients, improving prediction and (for LASSO) performing automatic variable selection — valuable when many candidate predictors (e.g., dozens of clinical or survey variables) are available relative to sample size.
- Decision trees: partition the predictor space into rectangular regions via a sequence of simple if/then splits, producing a highly interpretable but often high-variance model.
- Random forests: average many decision trees, each trained on a bootstrap sample with a random subset of predictors considered at each split, substantially reducing variance relative to a single tree and typically providing strong out-of-the-box predictive performance with modest tuning.
- Gradient boosting (e.g., XGBoost, LightGBM): builds trees sequentially, each correcting the errors of the ensemble so far, often achieving the best predictive performance on structured/tabular health data of the methods surveyed here, at some cost in interpretability and tuning complexity.
- Support vector machines: find the boundary that maximally separates classes (or, in regression form, fits within a margin of tolerance), historically strong on smaller, high-dimensional datasets.

PYTHON example

```
# Python: comparing logistic regression and random forest for predicting
# severe malaria risk at triage, with grouped cross-validation by facility
from sklearn.model_selection import GroupKFold, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier

X = triage_df[["age", "temperature", "resp_rate", "parasite_density", "danger_signs"]]
y = triage_df["severe_outcome"]
groups = triage_df["facility_id"]

gkf = GroupKFold(n_splits=5)
for name, model in [("Logistic Regression", LogisticRegression(max_iter=1000)),
                    ("Random Forest", RandomForestClassifier(n_estimators=300, random_state=42))]:
    scores = cross_val_score(model, X, y, groups=groups, cv=gkf, scoring="roc_auc")
    print(f"{name}: mean AUC = {scores.mean():.3f} (SD = {scores.std():.3f})")
```

Interpretability and Fairness in Clinical Prediction

A model destined to influence a clinical or resource-allocation decision must be interpretable enough for a clinician or programme manager to trust and audit it. Tools such as variable importance measures, partial dependence plots, and SHAP (SHapley Additive exPlanations) values allow a complex model such as a random forest or gradient boosting model to be interrogated for which features drive its predictions and how. Equally important, and too often neglected, is checking whether a model's performance is consistent across subgroups defined by geography (urban vs. rural LGA), sex, or facility level — a model trained predominantly on data from tertiary referral hospitals may perform poorly, and dangerously overconfidently, when deployed in a primary health centre with a different patient case mix.

Worked Example**A Full LASSO Variable Selection Walkthrough**

Consider a dataset with 25 candidate clinical and demographic predictors for severe malaria and only 400 patients — a ratio that risks overfitting in ordinary logistic regression. Fitting a LASSO-penalised logistic regression across a grid of penalty strengths,

and selecting the penalty that minimises cross-validated deviance, shrinks 16 of the 25 coefficients exactly to zero, leaving a parsimonious 9-predictor model (age, temperature, respiratory rate, parasite density, three danger-sign indicators, ITN use, and time since symptom onset) that achieves comparable cross-validated AUC to the full 25-predictor model while being considerably easier to interpret, deploy, and explain to clinical staff.

R example

```
# R: LASSO logistic regression with cross-validated penalty selection
library(glmnet)

x <- model.matrix(severe_outcome ~ ., data = triage_data)[, -1]
y <- triage_data$severe_outcome

cv_lasso <- cv.glmnet(x, y, family = "binomial", alpha = 1, nfolds = 10)
best_lambda <- cv_lasso$lambda.min

lasso_coefs <- coef(cv_lasso, s = best_lambda)
selected_vars <- rownames(lasso_coefs)[which(lasso_coefs != 0)]
print(selected_vars) # the retained, non-zero predictors
```

Worked Example**Grouped vs. Ungrouped Cross-Validation, Side by Side**

To make the warning in Section 11.3 concrete: fitting a random forest on the triage dataset with standard (ungrouped) 5-fold cross-validation reports a mean AUC of 0.88, while the same model with facility-grouped 5-fold cross-validation (Section 11.4's code) reports a mean AUC of 0.79. The gap (0.09 AUC points) reflects information leakage: in the ungrouped version, patients from the same facility — which share unmeasured facility-level characteristics such as staffing quality and diagnostic equipment — appear in both the training and validation folds, allowing the model to partly 'recognise' a facility's baseline risk pattern rather than genuinely generalising to a new, unseen facility. The grouped estimate of 0.79 is the more honest one for predicting the model's

performance in a facility not seen during training, which is precisely the deployment scenario of interest.

PYTHON example

```
# Python: demonstrating the AUC gap between ungrouped and grouped cross-validation
from sklearn.model_selection import KFold, GroupKFold, cross_val_score
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=300, random_state=42)

ungrouped_scores = cross_val_score(model, X, y, cv=KFold(5, shuffle=True, random_state=1),
                                   scoring="roc_auc")
grouped_scores = cross_val_score(model, X, y, groups=groups, cv=GroupKFold(5),
                                 scoring="roc_auc")

print(f"Ungrouped CV AUC: {ungrouped_scores.mean():.3f}")
print(f"Grouped CV AUC: {grouped_scores.mean():.3f}")
```

Feature Engineering for Health Prediction

The raw variables available in a health dataset are rarely the ones with the most predictive power in their original form; feature engineering — constructing new, more informative variables from the raw data — is often what separates a mediocre predictive model from a strong one. Useful health-specific feature engineering patterns include: converting a continuous age variable into clinically meaningful age bands (e.g., under-5, 5-14, 15+) that match how a clinical guideline actually stratifies risk; computing a rate or ratio (such as respiratory rate relative to age-specific normal range) rather than using the raw value alone; and creating an interaction feature explicitly (such as fever duration multiplied by an indicator for a danger sign) when domain knowledge suggests the combination, not just either variable alone, drives risk.

11.9 Extended Case Study: A Model That Worked Too Well

An instructive cautionary pattern in applied machine learning, sometimes even more dangerous than a poorly performing model because it more easily escapes scrutiny: a team building a model to predict which pregnant women are at high risk of a poor birth

outcome achieves a startlingly high cross-validated AUC of 0.97 — considerably higher than typically reported in the published clinical risk-prediction literature for comparable outcomes, which should itself be treated as a warning sign rather than simply celebrated. Investigating further, the team discovers that one of their candidate predictors, ‘mode of delivery’ (vaginal vs. caesarean), is recorded in the dataset at the time of delivery — meaning it is, in effect, determined essentially simultaneously with (and in many cases specifically because clinicians already anticipated) the poor outcome being predicted, exactly the reverse-causation and predictor-timing problem discussed in Section 8.10, now appearing in a machine learning rather than regression context.

Removing this single leaked predictor and re-running the full cross-validated pipeline drops the AUC to a much more realistic (and clinically far more useful, because genuinely achievable at the point of antenatal risk assessment) 0.79. This case study illustrates a general principle that should be applied whenever a model’s performance substantially exceeds what similar published models typically achieve on comparable data: an unusually strong result is at least as likely to indicate a data leakage problem (a predictor that is not genuinely available at prediction time, or overlap between training and test data of the kind discussed in Section 11.7) as it is to indicate a genuinely superior model, and a mature data science practice treats a suspiciously excellent result as a prompt for closer scrutiny rather than as an occasion for immediate celebration and deployment.

Calibration Curves as a Complement to AUC

Section 8.5 introduced calibration briefly in the regression context; it deserves a fuller treatment here because it is frequently under-reported in applied machine learning work relative to discrimination metrics like AUC. Two models can have identical

AUC values while having very different calibration: one producing predicted probabilities that closely match observed outcome frequencies across the full range of risk, the other producing predicted probabilities that are systematically too extreme (overconfident) or too conservative (underconfident) even while correctly ranking patients from lower to higher risk. For any model whose predicted probability (not just its rank ordering) will be shown directly to a clinician or used in a resource-allocation formula, a calibration curve — plotting observed outcome frequency against predicted probability across bins of the predicted risk — should be reported alongside AUC as standard practice, since AUC alone cannot reveal a calibration problem of the kind illustrated concretely in Section 8.8.

Class Imbalance

When the Outcome of Interest Is Rare

Many important health prediction targets are, fortunately, rare events — severe maternal complications, treatment-related mortality, a specific serious adverse drug reaction — and this rarity (class imbalance) creates a specific modelling challenge distinct from the sample-size concerns discussed elsewhere in this book: a model trained naively on a dataset where the outcome of interest occurs in only 2% of cases can achieve a deceptively high overall accuracy (98%) simply by always predicting the majority (‘no event’) class, while being clinically useless because it never actually identifies a true positive case. Standard remedies include using an evaluation metric appropriate to imbalanced data (AUC, or precision-recall curves, rather than raw accuracy), resampling techniques (oversampling the minority class or undersampling the majority class during training), and cost-sensitive learning approaches that explicitly penalise the model more heavily for missing a rare positive case than for a false alarm — echoing, in a machine learning training context, the same

asymmetric-cost reasoning introduced for classification thresholds in Section 15.10.

Ensemble Stacking

Combining Multiple Models for Better Prediction

Rather than selecting a single best-performing algorithm from the survey in Section 11.4, ensemble stacking trains several different base models (for example, a LASSO logistic regression, a random forest, and a gradient boosting model) and then trains a further, simple meta-model (often a plain logistic regression) to combine their individual predictions into a single, typically more accurate final prediction, exploiting the fact that different algorithms often make different kinds of errors on different subsets of cases. Stacking should always be validated with the same grouped cross-validation discipline established in Section 11.7, since a stacked ensemble is even more prone to a falsely optimistic performance estimate than a single model if the base models' predictions used to train the meta-model were generated on data the meta-model will later be evaluated on.

Fairness Metrics Beyond Subgroup AUC

A Brief Survey

Section 16.6's subgroup audit compared AUC and calibration across facility types, but the algorithmic fairness literature has developed several more formal fairness criteria worth knowing by name: demographic parity (the model's positive prediction rate is equal across subgroups), equalised odds (the model's true positive and false positive rates are equal across subgroups), and predictive parity (the model's positive predictive value is equal across subgroups). These criteria can conflict with each other mathematically — it is generally impossible to satisfy all three simultaneously except in special cases — meaning a health data scientist must make an explicit, justified choice about which

fairness criterion matters most for a specific clinical application, rather than assuming a single, universally correct definition of ‘fairness’ exists that a model can simply be checked against.

Using Scikit-learn Pipelines for Cleaner, Leak-Free Code

The LASSO and random forest code in Sections 11.6 and 11.7 applies preprocessing (standardisation) and modelling as separate steps, which creates a subtle risk: if standardisation is fit on the full dataset before cross-validation splits are made, information from the validation fold leaks into the training fold’s preprocessing statistics, inflating the reported cross-validated performance in the same spirit as the grouping leakage discussed in Section 11.7. Wrapping preprocessing and modelling together in a single scikit-learn ‘Pipeline’ guarantees that every preprocessing step is refit independently within each cross-validation fold, eliminating this leakage risk by construction rather than relying on the analyst to remember to do it manually.

Chapter Summary

- Machine learning emphasises predictive accuracy on new data, complementing (not replacing) the inferential focus of classical statistics.
- The bias-variance tradeoff underlies overfitting; regularisation, cross-validation, and ensemble methods all manage this tradeoff.
- Cross-validation must respect grouped/clustered data structure to avoid optimistic performance estimates — the gap between grouped and ungrouped CV can be substantial and is not a minor technicality.
- Random forests and gradient boosting typically perform strongly on structured health data; regularised regression

(LASSO) remains valuable for interpretability and automatic variable selection.

- Interpretability tools (SHAP, variable importance) and subgroup performance checks are essential before a predictive model informs clinical or policy decisions.

PYTHON example

```
# Python: a leak-free pipeline combining standardisation and model fitting
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegressionCV
from sklearn.model_selection import GroupKFold, cross_val_score

pipeline = Pipeline([
    ("scale", StandardScaler()),
    ("model", LogisticRegressionCV(cv=5, penalty="l1", solver="liblinear", max_iter=2000)),
])

scores = cross_val_score(pipeline, X, y, groups=groups, cv=GroupKFold(5), scoring="roc_auc")
print(f"Leak-free grouped CV AUC: {scores.mean():.3f}") # scaling is refit inside each fold
```

Mini-Project

Mini-Project 11: Building, Validating, and Interpreting a Clinical Risk Model

Build a predictive model for a binary clinical or public-health outcome, comparing at least two algorithms and using correctly grouped cross-validation.

- Step 1.** Choose or simulate a dataset with a binary outcome, at least 10 candidate predictors, and a grouping variable (e.g., facility or region).
- Step 2.** Fit at least two models (e.g., LASSO logistic regression and random forest) and compare their grouped cross-validated AUC.
- Step 3.** Demonstrate the effect of ignoring grouping by also reporting ungrouped cross-validated AUC, and quantify the gap (Section 11.7).
- Step 4.** For your best model, extract and report the top 5 most important predictors (via LASSO coefficients or random forest variable importance).
- Step 5.** Check whether your best model's AUC differs meaningfully between two subgroups (e.g., urban vs. rural, or two facility types), and comment on the implications for deployment.

Deliverable: *A model comparison table (grouped vs. ungrouped AUC for at least two algorithms), a variable importance summary, and a subgroup performance check.*

End-of-Chapter Exercises

1. Explain, in your own words, why a model with near-perfect accuracy on its training data can be a warning sign rather than a good result.
2. A dataset has 40 patients per facility across 15 facilities. Explain why a standard (non-grouped) 5-fold cross-validation could overstate model performance.
3. Propose one subgroup analysis you would run before deploying a triage risk model developed on tertiary hospital data to primary health centres.
4. Using the worked example in Section 11.6, explain in your own words why a 9-predictor LASSO model that performs comparably to a 25-predictor full model is generally preferable in a clinical deployment setting.
5. In Section 11.7, explain why the ungrouped cross-validated AUC (0.88) is described as ‘optimistic’ rather than simply ‘better’ than the grouped AUC (0.79).
6. Using the R code in Section 11.6 as a guide, describe what would happen to the number of selected variables if you increased the LASSO penalty (λ) well beyond ‘ $\lambda_{\min}$ ’.
7. A hospital administrator asks why your random forest model, which achieved 0.88 AUC in initial testing, performed worse when deployed at a different hospital. Drawing on Sections 11.5 and 11.7, write a short explanation of the likely cause(s).
8. Using Section 11.8, propose one engineered feature (beyond age banding) that could improve a severe malaria risk model, and explain the clinical reasoning behind it.
9. Explain why an interaction feature (e.g., fever duration $\times$ danger sign) might capture risk that neither variable alone would capture.

10. Give an example of a feature engineering choice that could accidentally leak outcome information into a predictor (creating an unrealistically strong but invalid predictor).
11. In the Section 11.9 case study, explain why an unusually high AUC (0.97) should prompt suspicion rather than celebration.
12. Why is ‘mode of delivery’ an inappropriate predictor for a model meant to inform risk assessment earlier in antenatal care, even though it is strongly associated with the outcome?
13. Describe one method (beyond noticing a suspiciously high AUC) you could use to actively check for this kind of data leakage before deployment.
14. Would removing the leaked predictor and accepting a lower but more honest AUC (0.79) be the right call even if it meant the model no longer appeared ‘state of the art’? Explain your reasoning.
15. Give one other example of a health variable that might be recorded at essentially the same time as, rather than genuinely before, the outcome it is meant to help predict.
16. Using Section 11.10, describe a scenario where two models have identical AUC but one is clearly better calibrated than the other.
17. Why is a calibration curve necessary in addition to, rather than instead of, an AUC score?
18. For which of the two example use cases in this book (a triage risk score shown directly to a clinician, versus a ranking-only prioritisation tool) does calibration matter most? Explain your reasoning.
19. Propose one practical fix if a model is found to be well-discriminating but poorly calibrated (systematically overconfident) in its highest-risk decile.

20. Explain why ‘the model correctly ranks patients from lower to higher risk’ is not sufficient reassurance if you plan to display the exact predicted probability to a clinician.
21. Using Section 11.11, explain why a model that always predicts ‘no event’ could achieve 98% accuracy on a dataset where the outcome occurs in 2% of cases, and why this would be a bad model despite the high accuracy.
22. Why is AUC or a precision-recall curve a more appropriate evaluation metric than raw accuracy for an imbalanced classification problem?
23. Describe the difference between oversampling and undersampling as remedies for class imbalance.
24. How is cost-sensitive learning, mentioned in Section 11.11, connected to the classification threshold discussion in Section 15.10?
25. Propose a Nigerian health prediction task (besides those mentioned in Section 11.11) where class imbalance would be a serious concern, and explain why.
26. Summarise, in your own words, why this chapter treats the bias-variance tradeoff as the single most important conceptual idea in machine learning.
27. Explain the common thread linking the class imbalance discussion (Section 11.11) and the calibration curves discussion (Section 11.10).
28. Why does this chapter warn that a model which ‘worked too well’ (Section 11.9) should prompt suspicion rather than celebration?
29. Propose a prediction task from your own field where class imbalance (Section 11.11) would likely be a serious concern, and describe one way you would address it.
30. If you had to give one warning to a colleague excited about a new machine learning model’s high accuracy, what would it be?

31. Using Section 11.13, explain why it is generally impossible to satisfy demographic parity, equalised odds, and predictive parity simultaneously.
32. Describe a clinical scenario where equalised odds might be the most ethically appropriate fairness criterion to prioritise.
33. Describe a different clinical scenario where predictive parity might be more appropriate instead.
34. Why is it important for a health data scientist to make an explicit, justified choice among fairness criteria rather than assuming one universal definition of fairness exists?
35. How does the concept of demographic parity relate to, but differ from, the subgroup AUC and calibration checks introduced in Chapter 16?