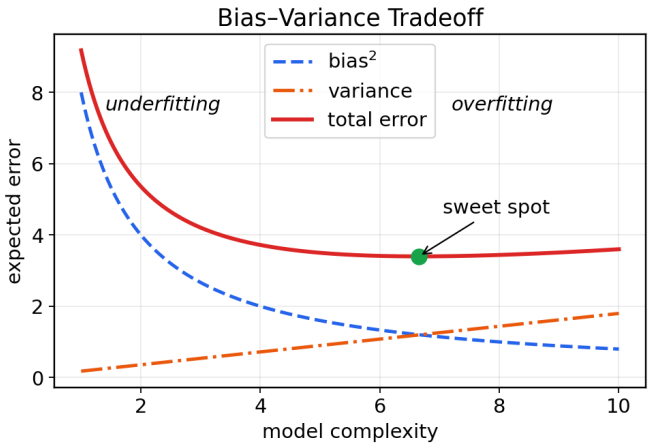
Generalisation & Bias–Variance

Goal: low error on *unseen* data. Expected error decomposes as

$$\mathbb{E}[(y - \hat{f}(x))^2] = \text{Bias}^2 + \text{Var} + \text{noise}.$$

Bias = model too restrictive (underfit). **Variance** = model too sensitive to data, memorises noise (overfit). **Noise** = irreducible. **Sweet spot** = minimum total error. More data ⇒ allows higher complexity without overfitting.

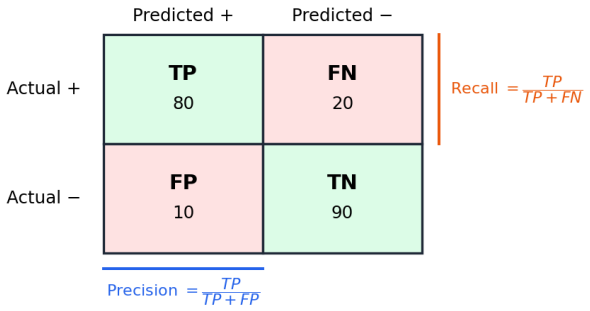


Cross-validation: k-fold (split data into k folds, train on $k-1$, test on 1; rotate). LOO = k=n. Use **train / validation / test** split: train fits weights, val tunes hyperparams, test reports final score. **Never tune on test** (data leakage).

Regularisation: L2 (Ridge, $\lambda\|w\|_2^2$) shrinks weights; L1 (Lasso, $\lambda\|w\|_1$) zeroes weights (sparsity / feature selection); early stopping; dropout.

Evaluation Metrics

Confusion matrix: precision reads down, recall reads across



Confusion matrix (TP, FP, FN, TN):

$$\text{acc} = \frac{TP+TN}{\text{all}}, \qquad \text{pre} = \frac{TP}{TP+FP},$$
$$\text{rec} = \frac{TP}{TP+FN}, \qquad F_1 = \frac{2TP}{2TP+FP+FN}.$$

Precision reads down predicted-+ column; **recall** across actual-+ row. Accuracy misleads on **class imbalance** (a 99%-class trivially scores 0.99). F_1 = harmonic mean of precision/recall.

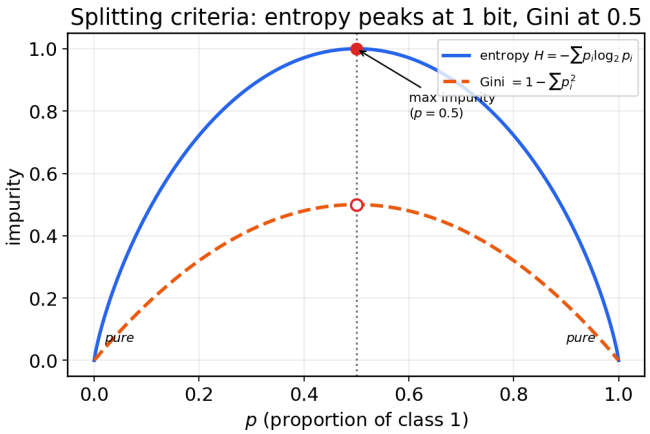
ROC-AUC: plot TPR vs FPR at every threshold; AUC = rank-quality. Threshold-free; useful for imbalanced data.

Decision Trees

Recursive **axis-aligned** splits. Greedy top-down. Split quality (impurity):

$$H(S) = -\sum_i p_i \log_2 p_i \ (\max \log_2 c), \quad \text{Gini} = 1 - \sum_i p_i^2.$$

Information gain = $H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v)$. Pure node ⇒ 0; binary peak = 1 bit at p=0.5.



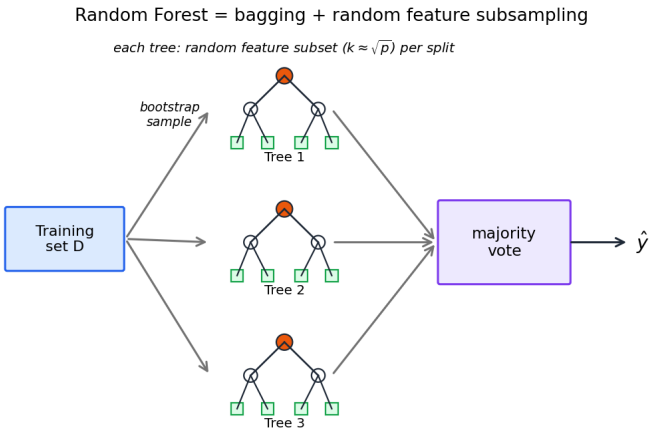
ID3 = info gain (biased toward many-valued attributes). **C4.5** = *gain ratio* = info gain / split info (penalises arity). **CART** = Gini for classification, *variance reduction* for regression. ID3 ⇒ greedy, complete (in finite domain), not optimal.

Pruning: *pre-pruning* (stop early — risk: horizon effect); *post-pruning* (grow then trim — better, more compute). Trees overfit easily ⇒ pruning crucial.

Ensembles

Bagging: train trees on bootstrap samples (size n *with replacement*; ≈ 63.2% in, 36.8% **OOB**); vote/average. Reduces **variance, not bias** (var → σ^2/m if independent). Works because trees are high-variance, low-bias.

Random Forest = bagging + *random feature subset at each split* ($k \approx \sqrt{p}$ classification, $p/3$ regression) ⇒ *decorrelated* trees (key: low correlation ⇒ averaging actually helps). **OOB error** = free generalisation estimate (≈ 37% unseen per tree, vote with those trees). **Feature importance** via mean decrease in impurity (MDI) or permutation OOB.



Boosting (AdaBoost, GBM, XGBoost): sequential, each new tree corrects predecessor's errors. Reduces *bias* (and variance). Sensitive to noisy labels.

Support Vector Machines

Maximum-margin hyperplane: find (w, b) separating classes with widest margin.

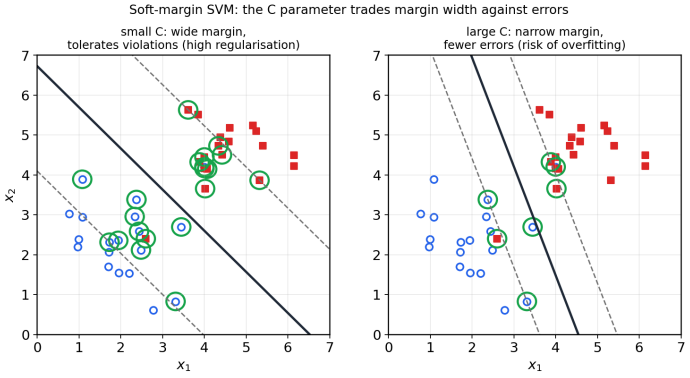
$$\text{margin} = \frac{2}{\|w\|}, \quad y_i(w^\top x_i + b) \geq 1.$$

Only **support vectors** (points on the margin, $y_i(w^\top x_i + b) = 1$) define w. SVM = **convex** quadratic program ⇒ unique global optimum (no local optima, unlike NN).

Soft margin (handle non-separable / noisy data):

$$\min_{w,b,\xi} \tfrac{1}{2}\|w\|^2 + C\sum_i \xi_i \;\text{ s.t. }\; y_i(w^\top x_i + b) \geq 1 - \xi_i,\; \xi_i \geq 0.$$

ξ_i = slack (margin violation). **Large** C ⇒ narrow margin, fewer errors ⇒ *overfit risk*. **Small** C ⇒ wide margin, more errors tolerated ⇒ stronger regularisation. ($C \to \infty$ recovers hard margin.)



Dual & Kernel Trick

Lagrangian dual (KKT): solution depends only on **dot products** $\langle x_i, x_j \rangle$ and Lagrange multipliers α_i; $\alpha_i > 0$ only for support vectors.

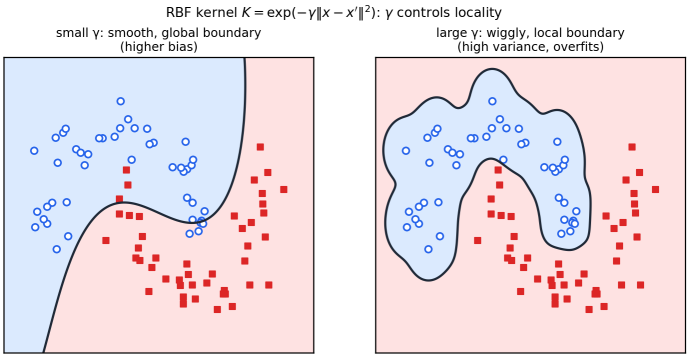
$$w = \sum_i \alpha_i y_i x_i, \quad f(x) = \sum_i \alpha_i y_i \langle x_i, x \rangle + b.$$

Kernel trick: replace $\langle x, x' \rangle$ with $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$ *without* computing Φ. Lets you fit in implicit infinite-dim feature spaces.

Common kernels:

- Linear: $K = \langle x, x' \rangle$ — same as no kernel.
- Polynomial: $K = (\langle x, x' \rangle + c)^d$ — interactions up to degree d.
- **RBF** (Gaussian): $K = e^{-\gamma\|x - x'\|^2}$. **Large** γ ⇒ narrow Gaussians ⇒ *local, wiggly, overfit* (not simpler!). Small γ ⇒ smooth, generalises.

Valid kernel ⇔ Mercer condition: K symmetric & positive semi-definite.



Exam Traps

- Bagging cuts **variance, not bias**; RF adds feature subsampling ⇒ decorrelation.
- OOB ≈ 37% unseen per tree ⇒ free validation set.
- ID3 → entropy/info-gain; **CART** → Gini; **C4.5** → gain ratio (penalises arity).
- Large C → overfit; **large** γ → **overfit** (more complex, NOT simpler).
- Precision = $TP/(TP{+}FP)$ vs recall = $TP/(TP{+}FN)$ — don't swap.
- Accuracy misleads on class imbalance ⇒ use precision/recall/F_1 or AUC.
- Never tune hyperparams on test set ⇒ data leakage.
- Bias–variance is a *tradeoff*, not a hierarchy; both contribute to error.

Oral-Exam Tactics

- *"Why a validation set?"* → train fits weights, val tunes hyperparams, test reports unbiased final score. Tuning on test = data leakage.
- *"Bias–variance — give an example."* → very shallow tree = high bias (underfits); very deep tree = high variance (overfits). Bagging → var↓, RF improves further by decorrelating.
- *"Why does the kernel trick work?"* → SVM dual depends only on $\langle x_i, x_j \rangle$. Replace with K ⇒ implicit non-linear features without computing Φ.
- *"What are support vectors?"* → training points on the margin (or violating it under soft margin); only ones with $\alpha_i > 0$ in dual; they alone define w.
- *"Why is large C overfitting?"* → C weights margin violations; large C = strict, narrow margin, fit every noisy point = overfit. Small C tolerates noise = regularises.
- *"Why does RF beat a single tree?"* → trees are high-variance/low-bias; averaging *decorrelated* predictors reduces variance (σ^2/m if independent). Random-feature subset achieves decorrelation.

Connection to the Paper

Shortcut learning (Saparov): the naive distribution lets the model fit *spurious correlations* (small lookahead) instead of the task. ML analogue: high-variance overfit to training distribution that doesn't generalise. **Balanced data** = removing spurious shortcuts = classic ML data-curation lesson. **Huge seed variance** on large graphs = bias–variance + loss-landscape instability.

One-liner: "ML in MoAI = bias–variance as the master trade-off, evaluated honestly via held-out data; decision trees fit interactions but overfit, so we bag/decorrelate (RF) to cut variance; SVMs maximise margin (convex, unique optimum) and kernels lift us to non-linear separability without paying the dimensionality cost."