DegreeLive

Bachelor of Commerce (B2001) | Level 3 • Part B Major | Monash University

Introduction to machine learning

Course Code: ETC3250 • Module 3 Notes

Module 3: Classification Methods: Logistic Regression, LDA/QDA, SVM & ROC Evaluation

3.1 Probabilistic Classification and the Bayes Optimal Classifier

In supervised classification, the response variable Y belongs to a finite unordered set of K classes $\mathrm{C} = \{1, 2, \ldots, K\}$. Rather than predicting a continuous quantity, a classifier assigns a feature vector $x \in \mathrm{R}^p$ to a discrete label $\hat{y}(x) \in \mathrm{C}$ or estimates posterior class probabilities $\hat{p}_k(x) = \hat{P}(Y = k \mid X = x)$ satisfying $\sum_{k=1}^{K} \hat{p}_k(x) = 1$.

Under symmetric Zero–One Loss $L_{0\text{-}1}(Y, \hat{y}) = \mathrm{I}(Y \neq \hat{y})$, the theoretical benchmark that minimises overall population misclassification probability $P(Y \neq \hat{y}(X))$ is the **Bayes Optimal Classifier**, which assigns every test point x to the class with the largest conditional probability:

$$\hat{y}_{\text{Bayes}}(x) = \arg\max_{k \in \{1,\ldots,K\}} P(Y = k \mid X = x)$$

Where:

- $P(Y = k \mid X = x)$ is the true posterior probability that an observation with predictor vector x belongs to class k.
- For binary classification ($K = 2$), the Bayes decision boundary is the hypersurface where $P(Y = 1 \mid X = x) = P(Y = 0 \mid X = x) = 0.5$.

The lowest achievable expected error rate of any classifier in the population—the **Bayes error rate**—is given by:

$$\text{Error}_{\text{Bayes}} = 1 - \mathrm{E}_X\left[\max_k P(Y = k \mid X)\right]$$

Where the expectation integrates over the marginal predictor distribution $P(X)$. Like σ_ε^2 in regression, the Bayes error rate represents irreducible stochastic overlap between classes. Because true conditional probabilities $P(Y = k \mid X)$ are unknown in practice, classification algorithms approximate them via two distinct paradigms: **discriminative models** (which model $P(Y \mid X)$ or the separating hyperplane directly, such as Logistic Regression and SVMs) and **generative models** (which model class-conditional feature distributions $P(X \mid Y = k)$ and priors $\pi_k = P(Y = k)$, then invert via Bayes' theorem, such as LDA, QDA, and Naive Bayes).

3.2 Binary and Multinomial Logistic Regression

Modelling a binary probability $p(x) = P(Y = 1 \mid X = x)$ directly with a linear function $\beta_0 + \beta^\top x$ fails because predictions can fall outside $[0, 1]$. Binary **Logistic Regression** resolves this by passing the linear predictor through the logistic (sigmoid) function $\sigma(z) = \frac{1}{1+e^{-z}}$:

$$p(x) = P(Y = 1 \mid X = x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)} = \frac{1}{1 + \exp\left(-(\beta_0 + \beta^\top x)\right)}$$

Where:

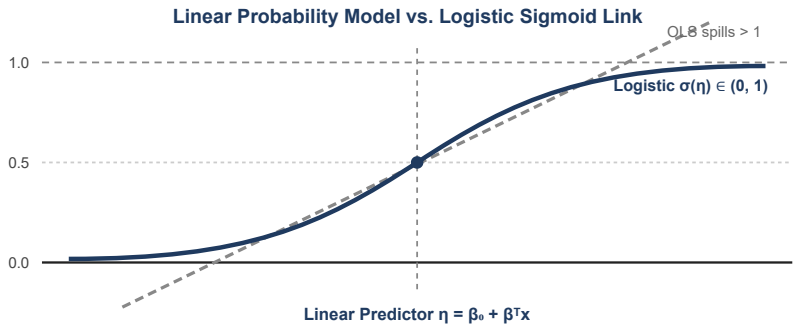
DegreeLive

- $p(x) \in (0,1)$ is the conditional probability of positive class membership (e.g., corporate loan default within 12 months).
- β_0 is the intercept and $\beta = (\beta_1, \dots, \beta_p)^\top$ are the log-odds slope coefficients.

Rearranging yields the **odds** $\frac{p(x)}{1-p(x)} = \exp(\beta_0 + \beta^\top x)$, and taking natural logarithms produces the **logit (log-odds)** linear equation:

$$\text{logit}(p(x)) = \ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

Thus, increasing predictor x_j by one unit changes the log-odds of $Y = 1$ by β_j, or equivalently multiplies the odds by $\exp(\beta_j)$ (the odds ratio). Note that the marginal probability effect $\frac{\partial p(x)}{\partial x_j} = \beta_j p(x)[1-p(x)]$ depends on the evaluation point x and peaks when $p(x) = 0.5$.



Regularised Logistic Regression

When p is large or training classes are cleanly separable, unpenalised MLE log-odds coefficients diverge toward $\pm\infty$. Adding an Elastic Net penalty to the negative log-likelihood resolves separation and performs feature selection simultaneously:

$$\min_{\beta_0,\beta} \{-\frac{1}{n}\ell(\beta_0,\beta) + \lambda\,[\alpha\|\beta\|_1 + \frac{1-\alpha}{2}\|\beta\|_2^2]\}$$

Where $\ell(\beta_0,\beta)$ is the Bernoulli log-likelihood, $\lambda \geq 0$ controls overall shrinkage, and $\alpha \in [0,1]$ balances L_1 sparsity against L_2 grouping (with predictors standardised inside the CV loop).

Maximum Likelihood Estimation (MLE)

Assuming independent observations $\{(x_i, y_i)\}_{i=1}^n$ with $y_i \in \{0,1\}$, parameters $\theta = (\beta_0, \beta)$ are estimated by maximising the Bernoulli log-likelihood (equivalently minimising negative log-likelihood / binary cross-entropy):

$$\ell(\beta_0,\beta) = \sum_{i=1}^{n} [y_i \ln p(x_i) + (1-y_i)\ln(1-p(x_i))] = \sum_{i=1}^{n} [y_i(\beta_0 + \beta^\top x_i) - \ln(1+\exp(\beta_0 + \beta^\top x_i))]$$

Where $p(x_i)$ is the logistic probability evaluated at x_i. Because $\ell(\theta)$ is strictly concave (whenever classes are not perfectly separable), iterative optimisation via Newton–Raphson / Iteratively Reweighted Least Squares (IRLS) converges to a unique global maximum. Warning: when classes are **linearly separable** in training data, the log-

likelihood keeps increasing as $\|\beta\|_2 \to \infty$, causing MLE coefficients to diverge—a problem solved cleanly by adding L_2 or L_1 regularisation (penalised logistic regression).

Multinomial Logistic Regression (Softmax)

For $K > 2$ classes, choosing class K as the reference baseline (or using symmetric softmax parametrisation) expresses class probabilities as:

$$P(Y = k \mid X = x) = \frac{\exp(\beta_{k0} + \beta_k^\top x)}{\sum_{l=1}^{K} \exp(\beta_{l0} + \beta_l^\top x)}, \qquad k = 1, \dots, K$$

Where β_{k0}, β_k are the class-specific parameters for class k (with $\beta_{K0} = 0, \beta_K = 0$ for identifiability when using a reference class).

Worked Example 3.1: Logistic Odds, Probability, and Log-Likelihood

An Australian fintech credit model predicts small-business loan default ($Y = 1$) using Debt-to-Equity ratio (X_1) and Cash Buffer in months (X_2), with estimated coefficients $\hat{\beta}_0 = -1.50, \hat{\beta}_1 = 0.80$, and $\hat{\beta}_2 = -0.60$. Consider a Melbourne hospitality borrower with $x_1 = 2.5$ and $x_2 = 1.0$ who subsequently defaults ($y = 1$). Compute the log-odds, odds, predicted default probability, odds ratio for X_1, and single-observation log-likelihood contribution.

1. **Linear Predictor (Log-Odds):**

$$\hat{\eta} = -1.50 + 0.80(2.5) - 0.60(1.0) = -1.50 + 2.00 - 0.60 = -0.10$$

2. **Odds and Predicted Probability:**

$$\text{Odds} = e^{-0.10} \approx 0.9048, \qquad \hat{p} = \frac{1}{1 + e^{0.10}} = \frac{0.9048}{1.9048} \approx 0.4750$$

3. **Odds Ratio Interpretation:** A 1-unit rise in Debt-to-Equity multiplies default odds by $e^{0.80} \approx 2.226$ (a 122.6% increase in odds), holding Cash Buffer constant.
4. **Log-Likelihood Contribution:** Since $y = 1$, $\ell_i = \ln(0.4750) \approx -0.7444$ (cross-entropy loss $+0.7444$). Under a standard 0.50 cutoff, $\hat{p} = 0.475 < 0.50$ would predict non-default, illustrating why threshold tuning matters when default costs exceed false-alarm costs.

3.3 Generative Classifiers: LDA, QDA, and Naive Bayes

Generative models invert Bayes' theorem to recover posterior probabilities from class priors $\pi_k = P(Y = k)$ and class-conditional densities $f_k(x) = p(X = x \mid Y = k)$:

$$P(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Where π_k is estimated by the training proportion $\hat{\pi}_k = n_k / n$, and $f_k(x)$ is modelled parametrically.

DegreeLive

Linear Discriminant Analysis (LDA)

LDA assumes that predictors within each class k follow a multivariate Gaussian distribution $X \mid (Y = k) \sim \mathrm{N}_p(\mu_k, \Sigma)$ with class-specific mean vector μ_k and a **shared (pooled) covariance matrix** Σ across all K classes:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)\right)$$

Taking the logarithm of $\pi_k f_k(x)$ and cancelling terms that are identical across all classes k (including the quadratic term $-\frac{1}{2}x^\top\Sigma^{-1}x$ because Σ does not depend on k) yields the **linear discriminant function** $\delta_k(x)$:

$$\delta_k(x) = x^\top \Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^\top \Sigma^{-1}\mu_k + \ln(\pi_k)$$

Where:

- μ_k is estimated by the class sample mean vector $\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i$.
- Σ is estimated by the pooled within-class sample covariance matrix $\hat{\Sigma} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)^\top$.
- π_k is estimated by $\hat{\pi}_k = n_k/n$.

An observation x is assigned to whichever class maximises $\hat{\delta}_k(x)$. Because $\delta_k(x)$ is linear in x, the decision boundary separating any pair of classes k and l, defined by $\delta_k(x) = \delta_l(x)$, is always a flat **hyperplane**. Unlike Logistic Regression, LDA remains completely stable when classes are cleanly separable and performs efficiently when Gaussian assumptions hold roughly true on small samples.

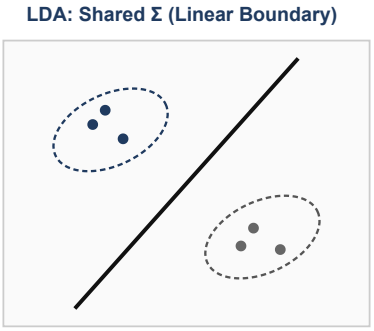
Quadratic Discriminant Analysis (QDA)

QDA relaxes LDA's equal-covariance assumption, allowing each class to possess its own covariance matrix Σ_k ($X \mid (Y = k) \sim \mathrm{N}_p(\mu_k, \Sigma_k)$). Now the quadratic term $-\frac{1}{2}x^\top\Sigma_k^{-1}x$ and log-determinant $-\frac{1}{2}\ln|\Sigma_k|$ differ across classes and cannot be dropped, producing a **quadratic discriminant function**:

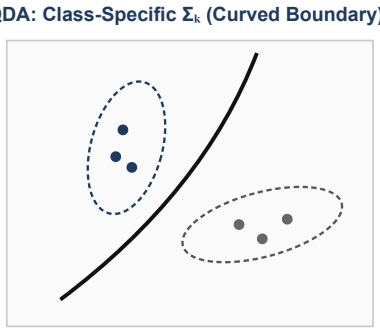
$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^\top\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\ln|\Sigma_k| + \ln(\pi_k)$$

Consequently, QDA decision boundaries are quadratic surfaces (parabolas, ellipses, hyperbolas). This creates a classic bias–variance trade-off: estimating K separate $p \times p$ covariance matrices requires estimating $K\frac{p(p+1)}{2}$ covariance parameters instead of $\frac{p(p+1)}{2}$ in LDA. When p is large or n is small, QDA suffers high variance and overfits; when n is large and class spreads differ markedly, QDA reduces bias substantially.

DegreeLive



Equal scatter orientation & size



Unequal covariance ellipses

Naive Bayes Classifier

When p is very large, estimating even a single $p \times p$ covariance matrix in LDA becomes noisy. **Naive Bayes** assumes that within each class k, all p features are mutually conditionally independent: $f_k(x) = \prod_{j=1}^{p} f_{kj}(x_j)$. This reduces covariance estimation to p one-dimensional marginal densities per class (for example, univariate Gaussians with diagonal covariance, histograms, or multinomial word counts in NLP). Despite the unrealistic independence assumption, Naive Bayes frequently exhibits remarkably low variance and strong classification accuracy in high-dimensional settings such as document and email filtering.

Worked Example 3.2: 1D LDA vs. QDA Classification Score

Suppose a univariate risk score X predicts whether an ASX firm receives a credit downgrade ($k = 1$, prior $\pi_1 = 0.25$, mean $\mu_1 = 2.0$) versus remains stable ($k = 0$, prior $\pi_0 = 0.75$, mean $\mu_0 = 5.0$), with pooled variance $\sigma^2 = 1.5$. Classify a firm with observed score $x = 3.2$ using LDA (note $\ln(0.25) \approx -1.3863$ and $\ln(0.75) \approx -0.2877$).

1. **Compute $\delta_1(3.2)$ (Downgrade):**

$$\delta_1(3.2) = \frac{3.2(2.0)}{1.5} - \frac{2.0^2}{2(1.5)} + \ln(0.25) = 4.2667 - 1.3333 - 1.3863 = 1.5471$$

2. **Compute $\delta_0(3.2)$ (Stable):**

$$\delta_0(3.2) = \frac{3.2(5.0)}{1.5} - \frac{5.0^2}{2(1.5)} + \ln(0.75) = 10.6667 - 8.3333 - 0.2877 = 2.0457$$

3. **Decision and Exact Boundary:** Since $\delta_0(3.2) = 2.0457 > \delta_1(3.2) = 1.5471$, LDA classifies the firm as **Stable ($k = 0$)**. Setting $\delta_1(x^*) = \delta_0(x^*)$ yields the exact threshold $x^* = \frac{\mu_0+\mu_1}{2} + \frac{\sigma^2}{\mu_0-\mu_1}\ln\left(\frac{\pi_1}{\pi_0}\right) = 3.5 + \frac{1.5}{3.0}(-1.0986) = 2.9507$, shifted left of the midpoint 3.5 because class o has a three-times-larger prior.

3.4 Confusion Matrices, Imbalanced Metrics, and ROC/AUC Evaluation

Overall accuracy ($1 - \text{error rate}$) is deeply misleading whenever classes are imbalanced. If only 1% of Australian Medicare claims are fraudulent, a trivial classifier that labels every claim legitimate achieves 99% accuracy while failing completely on its operational objective. Rigorous classifier evaluation disaggregates errors across the 2×2 **confusion matrix**:

Degreelive

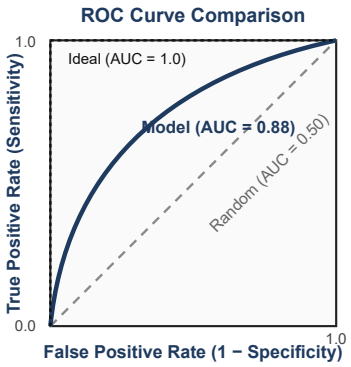
Predicted / Actual	Actual Positive ($Y = 1$)	Actual Negative ($Y = 0$)
Predicted Positive ($\hat{Y} = 1$)	True Positive (TP)	False Positive (FP, Type I error)
Predicted Negative ($\hat{Y} = 0$)	False Negative (FN, Type II error)	True Negative (TN)

From these four counts, key threshold-dependent diagnostic metrics follow:

- **Sensitivity (Recall / True Positive Rate, TPR):** Share of actual positives correctly flagged: $\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}$.
- **Specificity (True Negative Rate, TNR):** Share of actual negatives correctly cleared: $\text{TNR} = \frac{\text{TN}}{\text{TN}+\text{FP}}$.
- **False Positive Rate (FPR):** Share of actual negatives falsely flagged: $\text{FPR} = 1 - \text{TNR} = \frac{\text{FP}}{\text{FP}+\text{TN}}$.
- **Precision (Positive Predictive Value, PPV):** Share of positive alerts that are genuinely positive: $\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$.
- **F_1-Score:** Harmonic mean balancing Precision and Recall: $F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}} = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$.

ROC Curve, Area Under the Curve (AUC), and Precision–Recall Curves

Every probabilistic classifier produces a continuous score $\hat{p}(x)$ that converts into a hard class label via a probability threshold $\tau \in [0, 1]$ (predict 1 if $\hat{p}(x) \geq \tau$). Lowering τ raises both TPR (desirable) and FPR (undesirable). Plotting $\text{TPR}(\tau)$ on the vertical axis against $\text{FPR}(\tau)$ on the horizontal axis as τ sweeps from 1 down to 0 traces the **Receiver Operating Characteristic (ROC) curve**.



The **Area Under the ROC Curve (AUC)** summarises overall ranking discrimination across all possible thresholds τ into a single number in $[0, 1]$. Probabilistically, AUC equals the probability that a randomly chosen positive observation receives a strictly higher predicted score than a randomly chosen negative observation: $\text{AUC} = P(\hat{p}(X^+) > \hat{p}(X^-))$. A random coin-flip classifier scores $\text{AUC} = 0.50$, whereas a perfect separator scores $\text{AUC} = 1.00$. When positive cases are exceedingly rare (for example, 0.2% prevalence), FPR remains deceptively tiny even when thousands of false alarms occur; in such regimes, analysts supplement ROC-AUC with the **Precision–Recall (PR) curve** and PR-AUC, because Precision directly exposes false positive inflation.

Cost-Sensitive Classification Thresholding

Default classification software applies a symmetric probability threshold $\tau = 0.50$, which implicitly assumes False Positives and False Negatives carry identical economic penalties. In commercial practice, misclassification costs are heavily asymmetric. Suppose falsely flagging a valid transaction or rejecting a creditworthy borrower incurs cost C_{FP}, whereas failing to catch an actual fraud or loan default incurs cost C_{FN} (while correct decisions cost 0). Given calibrated posterior probability $p(x) = P(Y = 1 \mid X = x)$, predicting positive ($\hat{y} = 1$) yields expected cost $(1 - p(x))C_{\text{FP}}$, whereas predicting negative ($\hat{y} = 0$) yields expected cost $p(x)C_{\text{FN}}$. Predicting positive is optimal whenever $(1 - p(x))C_{\text{FP}} < p(x)C_{\text{FN}}$, which solves to the Bayes optimal decision threshold:

$$\tau^* = \frac{C_{\text{FP}}}{C_{\text{FP}} + C_{\text{FN}}}$$

Where:

- $C_{\text{FP}} > 0$ is the unit cost of a False Positive (false alarm).
- $C_{\text{FN}} > 0$ is the unit cost of a False Negative (missed positive event).

For example, if an Australian bank incurs $A\$200$ in manual investigation costs for a false fraud alert ($C_{\text{FP}} = 200$) versus $A\$3{,}800$ in unrecovered chargeback loss for a missed fraudulent transaction ($C_{\text{FN}} = 3800$), the optimal cutoff drops sharply from 0.50 to $\tau^* = 200/(200 + 3800) = 0.05$.

Worked Example 3.3A: Financial Impact of Cost-Sensitive Thresholding

An Australian e-commerce merchant evaluates $1{,}000$ card transactions (40 actual frauds, 960 legitimate). Each missed fraud costs $C_{\text{FN}} = A\$1{,}800$, while each false alarm triggers a customer verification call costing $C_{\text{FP}} = A\$200$ (so Bayes optimal cutoff is $\tau^* = 200/(200 + 1800) = 0.10$). Under standard cutoff $\tau = 0.50$, the classifier yields $\text{TP} = 22, \text{FN} = 18, \text{FP} = 6, \text{TN} = 954$. Lowering the cutoff to $\tau^* = 0.10$ yields $\text{TP} = 36, \text{FN} = 4, \text{FP} = 38, \text{TN} = 922$. Compare operational performance and total AUD cost:

1. **Standard Cutoff ($\tau = 0.50$):** Accuracy $= (22 + 954)/1000 = 97.6\%$, Recall $= 22/40 = 55.0\%$, Precision $= 22/28 = 78.6\%$.

$$\text{Total Cost}_{0.50} = 18(1{,}800) + 6(200) = 32{,}400 + 1{,}200 = A\$33{,}600$$

2. **Cost-Optimal Cutoff ($\tau^* = 0.10$):** Accuracy drops slightly to $(36 + 922)/1000 = 95.8\%$ and Precision falls to $36/74 = 48.6\%$, yet Recall rises to $36/40 = 90.0\%$:

$$\text{Total Cost}_{0.10} = 4(1{,}800) + 38(200) = 7{,}200 + 7{,}600 = A\$14{,}800$$

Tuning the decision threshold saves $A\$18{,}800$ (a 56.0% cost reduction) despite lower headline accuracy, underscoring why raw accuracy and fixed 0.50 cutoffs fail in asymmetric business decisions.

Worked Example 3.3: Confusion Matrix Metrics and Trapezoidal ROC-AUC

A test set of $n = 5$ credit applicants has true default statuses $y = (1, 1, 0, 1, 0)$ and predicted probabilities sorted in descending order: $\hat{p} = (0.90, 0.70, 0.60, 0.40, 0.15)$. First compute Precision, Recall, Specificity, and F_1 at threshold $\tau = 0.50$; then compute the exact ROC-AUC.

1. **Classification at $\tau = 0.50$:** Predictions are $\hat{y} = (1, 1, 1, 0, 0)$. Comparing against $y = (1, 1, 0, 1, 0)$ gives $\text{TP} = 2, \text{FP} = 1, \text{FN} = 1, \text{TN} = 1$.
   - $\text{Accuracy} = (2 + 1)/5 = 0.60$
   - $\text{Recall (TPR)} = 2/(2 + 1) = 0.667, \text{Specificity} = 1/(1 + 1) = 0.500$
   - $\text{Precision} = 2/(2 + 1) = 0.667, F_1 = \frac{2(0.667)(0.667)}{0.667+0.667} = 0.667$
2. **Exact ROC Points and Trapezoidal AUC:** There are $P = 3$ positives and $N = 2$ negatives. Sweeping threshold τ downward past each score traces (FPR, TPR): $(0, 0) \to (0, 1/3) \to (0, 2/3) \to (1/2, 2/3) \to (1/2, 1) \to (1, 1)$. Summing vertical rectangles under the step curve:

$$\text{AUC} = (\frac{1}{2} - 0) \cdot \frac{2}{3} + (1 - \frac{1}{2}) \cdot 1 = \frac{1}{3} + \frac{1}{2} = \frac{5}{6} \approx 0.833$$

Verifying by pairwise concordance: out of $3 \times 2 = 6$ positive–negative pairs, 5 pairs rank the positive above the negative (only $0.40 < 0.60$ is inverted), yielding $5/6 = 0.833$.

3.5 Maximal Margin Classifiers and Support Vector Classifiers

Whereas Logistic Regression and LDA seek decision boundaries derived from probabilistic likelihoods over all observations, **Support Vector Machines (SVMs)** adopt a purely geometric approach focused on points near the boundary. Consider binary labels coded symmetrically as $y_i \in \{-1, +1\}$ and a separating hyperplane $f(x) = w^\top x + b = 0$, where $w \in \mathrm{R}^p$ is the normal vector perpendicular to the hyperplane and $b \in \mathrm{R}$ is the offset.

Maximal Margin Hyperplane (Separable Case)

For any point x_i correctly classified ($y_i(w^\top x_i + b) > 0$), the perpendicular Euclidean distance from x_i to the hyperplane is $\frac{y_i(w^\top x_i + b)}{\|w\|_2}$. Scaling (w, b) so that the closest points on each side satisfy $y_i(w^\top x_i + b) = 1$ sets the half-margin width to $M = \frac{1}{\|w\|_2}$ (total street width $\frac{2}{\|w\|_2}$). Maximising the margin M is equivalent to solving the convex quadratic program:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 \qquad \text{subject to} \quad y_i(w^\top x_i + b) \geq 1, \quad i = 1, \ldots, n$$

Where:

- $\|w\|_2^2 = \sum_{j=1}^{p} w_j^2$ is the squared Euclidean norm of the weight vector.
- The training observations sitting exactly on the margin boundaries $y_i(w^\top x_i + b) = 1$ are the **support vectors**; moving any non-support-vector point further away leaves the optimal hyperplane completely unchanged.

Support Vector Classifier (Soft Margin)

Real-world commercial datasets are rarely linearly separable, and enforcing a rigid hard margin makes the hyperplane hypersensitive to a single mislabelled outlier. The **Support Vector Classifier (SVC)** introduces non-negative **slack variables** $\xi_1, \ldots, \xi_n \geq 0$ permitting observations to violate the margin or fall on the wrong side of the hyperplane, penalised by a regularisation cost parameter $C > 0$:

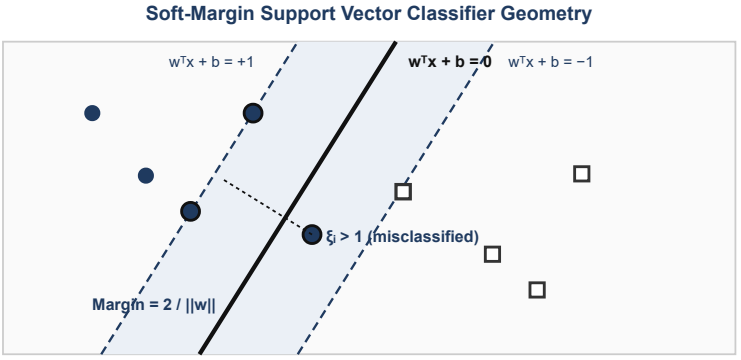
$$\min_{w,b,\xi} \{\frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^{n} \xi_i\} \qquad \text{subject to} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Where:

- $\xi_i = 0$ indicates observation i lies cleanly on or outside the correct margin boundary; $0 < \xi_i \leq 1$ means i sits inside the margin yet on the correct side of the hyperplane; $\xi_i > 1$ means i is misclassified.
- $C > 0$ controls the bias–variance trade-off: a **large** C heavily penalises margin violations (narrower street, fewer support vectors, lower bias, higher variance), whereas a **small** C tolerates wider violations (wider margin, more support vectors, higher bias, lower variance).

DegreeLive

Equivalently, eliminating ξ_i reveals that the SVC minimises **hinge loss** plus an L_2 penalty: $\sum_{i=1}^{n} \max(0, 1 - y_i(w^\top x_i + b)) + \lambda \|w\|_2^2$, directly parallel to regularised logistic regression.



3.6 Nonlinear Support Vector Machines and the Kernel Trick

When class boundaries are inherently nonlinear (for instance, concentric rings or complex credit score interactions), linear hyperplanes fail even with slack variables. A natural remedy maps raw predictors $x \in \mathrm{R}^p$ into a higher-dimensional feature space via $\phi(x) \in \mathrm{R}^D$ and fits a linear SVC there. Explicitly forming high-dimensional polynomial or infinite-dimensional expansions $\phi(x)$, however, can be computationally intractable.

Solving the SVC via Lagrange duality reveals that the decision function depends *exclusively* on pairwise inner products $\langle x_i, x_j \rangle = x_i^\top x_j$ rather than the coordinates themselves:

$$f(x) = b + \sum_{i \in \mathrm{S}} \alpha_i y_i K(x, x_i)$$

Where:

- S is the sparse index set of support vectors (the observations with non-zero Lagrange multipliers $\alpha_i > 0$).
- $K(x, x') = \langle \phi(x), \phi(x') \rangle$ is a positive semi-definite **kernel function** (satisfying Mercer's condition) that computes the inner product directly in the original p-dimensional space without ever explicitly evaluating $\phi(x)$ —the **kernel trick**.

Kernel Name	Mathematical Formula $K(x, x')$	Hyperparameters	Geometric Interpretation
Linear Kernel	$x^\top x'$	Cost C	Standard linear SVC hyperplane
Polynomial Kernel	$(\gamma\, x^\top x' + r)^d$	Degree d, scale γ, offset r, cost C	All interaction monomials up to degree d
Radial Basis Function (RBF / Gaussian)	$\exp(-\gamma \|x - x'\|_2^2)$	Bandwidth $\gamma > 0$, cost C	Infinite-dimensional smooth local similarity bumps

In the RBF kernel, $\gamma > 0$ governs the spatial reach of each support vector: a very large γ restricts influence to tiny islands around individual training points (low bias, severe overfitting), while a small γ produces broad, smooth boundaries. Because $\|x - x'\|_2^2$ depends on Euclidean distances, rigorous feature standardisation inside the cross-validation pipeline is indispensable before tuning (C, γ) via grid search.

DegreeLive

Worked Example 3.4: Margin Width and RBF Kernel Evaluation

Suppose a linear SVM fitted to standardised predictors yields weight vector $w = (1.2, -1.6)^\top$ and bias $b = 0.4$, while an RBF SVM with $\gamma = 0.5$ has two support vectors $x_1 = (1,1)^\top$ ($y_1 = +1, \alpha_1 = 1.5$) and $x_2 = (3,1)^\top$ ($y_2 = -1, \alpha_2 = 1.2$) with $b = -0.1$.

1. **Linear SVM Margin Width:** Compute $\|w\|_2 = \sqrt{1.2^2 + (-1.6)^2} = \sqrt{1.44 + 2.56} = \sqrt{4.00} = 2.0$. Hence the half-margin is $M = 1/2.0 = 0.50$ and total street width is $2/2.0 = 1.00$ standardised unit.
2. **RBF SVM Prediction at New Point $x^* = (2,1)^\top$:** Compute squared Euclidean distances: $\|x^* - x_1\|_2^2 = (2-1)^2 + (1-1)^2 = 1$ and $\|x^* - x_2\|_2^2 = (2-3)^2 + (1-1)^2 = 1$. Evaluate RBF kernel values:

   $$K(x^*, x_1) = \exp(-0.5 \times 1) = e^{-0.5} \approx 0.6065, \qquad K(x^*, x_2) = \exp(-0.5 \times 1) = e^{-0.5} \approx 0.6065$$

   Compute decision score:

   $$f(x^*) = -0.1 + 1.5(+1)(0.6065) + 1.2(-1)(0.6065) = -0.1 + 0.9098 - 0.7278 = +0.0820 > 0$$

   Since $f(x^*) > 0$, the RBF SVM assigns x^* to class $+1$.

Lagrangian Duality, Multiclass SVMs, and Probability Calibration

The sparse representation $f(x) = b + \sum_{i \in S} \alpha_i y_i K(x, x_i)$ follows directly from the Karush–Kuhn–Tucker (KKT) complementary slackness conditions of the soft-margin SVM dual quadratic program:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad \text{subject to} \quad 0 \le \alpha_i \le C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Where α_i are the dual Lagrange multipliers. Under KKT complementarity, any observation strictly on the safe side of the margin ($y_i f(x_i) > 1$) has $\alpha_i = 0$ and plays zero role in the final classifier; observations exactly on the margin boundary ($y_i f(x_i) = 1$) satisfy $0 < \alpha_i < C$, while margin violators ($y_i f(x_i) < 1$) saturate at $\alpha_i = C$.

Because SVMs are inherently binary hyperplane classifiers, extending them to $K > 2$ classes relies on one of two decomposition schemes:

- **One-Versus-One (OvO):** Trains $\binom{K}{2}$ pairwise binary SVMs comparing every pair of classes (k, l) and assigns a test point to the class winning the majority of pairwise votes. Standard default in R's `e1071` / `kernlab` packages.
- **One-Versus-All (OvA / One-Versus-Rest):** Trains K binary SVMs, each separating class k against all remaining $K - 1$ classes combined, and picks the class yielding the highest signed margin score $\hat{f}_k(x)$.

Finally, raw SVM decision scores $f(x)$ measure signed geometric distance rather than calibrated probabilities. When ROC evaluation or cost-sensitive thresholding requires posterior probabilities $\hat{p}(x)$, **Platt scaling** fits a univariate logistic regression mapping $f(x)$ to $P(Y = 1 \mid f(x)) = [1 + \exp(Af(x) + B)]^{-1}$ via internal cross-validation to avoid calibration overfitting.

DegreeLive

3.7 Comparative Selection Across Classification Algorithms

No single classifier dominates across every empirical geometry. Selecting between Logistic Regression, LDA, QDA, Naive Bayes, K-Nearest Neighbours, and Kernel SVMs depends on whether the true class boundary is linear or curved, whether Gaussian covariance assumptions hold, and how n compares with p:

Classifier	Decision Boundary	Probability Output	Key Assumptions & Strengths	Primary Vulnerability
Logistic Regression	Linear hyperplane	Direct calibrated MLE	No Gaussian feature assumption; direct odds-ratio interpretation	Diverges under exact linear separation unless regularised
LDA	Linear hyperplane	Via Bayes' theorem	Stable under clean class separation; efficient when Gaussian with shared Σ	High bias if class covariances differ sharply
QDA	Quadratic surface	Via Bayes' theorem	Captures elliptical / curved class boundaries natively	High variance when p is large relative to n_k ($Kp(p+1)/2$ parameters)
Naive Bayes	Nonlinear (additive log-likelihood)	Via Bayes' theorem	Scales effortlessly to high p (text/NLP); low variance	Conditional independence often overstates probability extremity
KNN Classification	Highly irregular local boundary	Local neighbour vote fraction	Zero parametric boundary assumptions; captures arbitrary shapes	Degrades rapidly under curse of dimensionality and unscaled features
RBF Kernel SVM	Smooth nonlinear hypersurface	Requires Platt scaling	Effective in high-dimensional spaces; depends only on support vectors	Sensitive to (C, γ) tuning and feature scaling; low interpretability

Worked Example 3.5: LDA vs. QDA Parameter Complexity and Boundary Shift

Suppose a Monash quantitative analyst classifies corporate borrowers across $K = 3$ credit rating tiers (Investment Grade, Sub-Investment, Distressed) using $p = 15$ financial ratios. Compare the number of covariance matrix parameters estimated by LDA versus QDA, and explain the bias–variance consequence if the training sample has $n = 300$ firms with $n_3 = 25$ distressed firms:

1. **LDA Covariance Parameter Count:** A single symmetric 15×15 pooled covariance matrix Σ contains $\frac{p(p+1)}{2} = \frac{15 \times 16}{2} = 120$ distinct variance and covariance parameters, estimated across all $n - K = 297$ degrees of freedom.
2. **QDA Covariance Parameter Count:** Three class-specific covariance matrices $\Sigma_1, \Sigma_2, \Sigma_3$ require $K\frac{p(p+1)}{2} = 3 \times 120 = 360$ covariance parameters. Worse, $\hat{\Sigma}_3$ must estimate 120 parameters using only $n_3 = 25$ distressed observations! Because $p = 15$ is comparable to $n_3 = 25$, $\hat{\Sigma}_3$ is severely ill-conditioned and noisy, causing QDA to overfit the distressed class dramatically. In this small-class regime, LDA (or regularised discriminant analysis) substantially outperforms QDA out-of-sample.

DegreeLive