IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE DA

IAE Academy free study material · Updated 5 October 2026

Machine Learning Formula Sheet

INSIDE

- 103 formulas in 7 topics
- When to use each one
- Taken from the free topic notes on iaeacademy.in

Online, one page per topic: **iaeacademy.in/study/machine-learning/formula-sheet**

FREE COUNSELLING

Preparing for GATE DA? Talk to IAE Academy.

Or call **+91 84204 36524**
Every day, 10:30 am to 7:30 pm

Chat on WhatsApp

Free. Not for sale. Pass it on.

Simple, multiple and ridge regression

MEAN SQUARED ERROR (MSE)

$$\mathrm{MSE}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\|\mathbf{y} - X\mathbf{w}\|^2$$

Average squared residual; the sum of squared errors (SSE) is the same without the factor $1/n$. Minimising either gives the same $\mathbf{w}$.

LEAST-SQUARES SLOPE AND INTERCEPT

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$$

Needs at least two distinct x values ($S_{xx} > 0$). The fitted line always passes through $(\bar{x}, \bar{y})$.

COEFFICIENT OF DETERMINATION

$$R^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} = \frac{\mathrm{SSR}}{\mathrm{SST}}, \quad \mathrm{SST} = \sum(y_i - \bar{y})^2,\ \mathrm{SSE} = \sum(y_i - \hat{y}_i)^2,\ \mathrm{SSR} = \sum(\hat{y}_i - \bar{y})^2$$

The two forms agree for least squares with an intercept on the training data, where SST = SSR + SSE; then $0 \le R^2 \le 1$ (assuming SST > 0). For simple regression $R^2 = r^2 = \hat{\beta}_1^2 S_{xx}/S_{yy}$. On new (test) data, or for a model without an intercept, $1 - \mathrm{SSE}/\mathrm{SST}$ can be negative.

NORMAL EQUATIONS AND THE OLS SOLUTION

$$X^T X\,\mathbf{w} = X^T\mathbf{y} \implies \hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$$

The inverse form requires $X^T X$ to be invertible, i.e. the columns of X (including the column of ones, if present) are linearly independent. This needs n to be at least the number of columns. If one column is a linear combination of others (exact collinearity, or more columns than samples), $X^T X$ is singular: minimisers still exist, but there are infinitely many.

RIDGE OBJECTIVE

$$J(\mathbf{w}) = \|\mathbf{y} - X\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2$$

Sum of squared errors plus λ times the squared Euclidean norm of the weights. $\lambda = 0$ gives OLS. For $\lambda > 0$ the objective is strictly convex, so the minimiser is unique. Equivalent constrained form: minimise $\|\mathbf{y} - X\mathbf{w}\|^2$ subject to $\|\mathbf{w}\|^2 \le t$, for a t that depends on λ.

RIDGE CLOSED FORM

$$\hat{\mathbf{w}}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$$

For $\lambda > 0$ the matrix $X^T X + \lambda I$ is always invertible (positive definite), even when $X^T X$ is singular. Here I is the identity of size equal to the number of weights.

GRADIENT DESCENT UPDATE FOR MSE

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\,\frac{2}{n} X^T(X\mathbf{w} - \mathbf{y})$$

$\eta > 0$ is the learning rate. The iterates converge from any starting point if $0 < \eta < 2/\mu_{\max}$, where $\mu_{\max}$ is the largest eigenvalue of $H = \frac{2}{n}X^T X$. If $\eta > 2/\mu_{\max}$ they generally diverge.

Logistic regression, k-nearest neighbour and naive Bayes

SIGMOID (LOGISTIC FUNCTION)

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The output lies in (0, 1). $\sigma(0) = 0.5$; $\sigma(z) \to 1$ as $z \to \infty$ and $\sigma(z) \to 0$ as $z \to -\infty$.

SIGMOID IDENTITIES

$$\sigma(-z) = 1 - \sigma(z), \qquad \sigma'(z) = \sigma(z)\,\big(1 - \sigma(z)\big)$$

The derivative is at most 0.25, reached at $z = 0$. This identity is what makes the gradient of the cross-entropy loss so simple.

LOGISTIC REGRESSION MODEL

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b), \qquad P(y = 0 \mid \mathbf{x}) = 1 - P(y = 1 \mid \mathbf{x})$$

Binary classification. The prediction is class 1 when this probability exceeds the chosen threshold, which is 0.5 unless the question states another.

LOG-ODDS (LOGIT) IS LINEAR

$$\ln \frac{P(y = 1 \mid \mathbf{x})}{1 - P(y = 1 \mid \mathbf{x})} = \mathbf{w}^\top \mathbf{x} + b$$

With the other features fixed, each unit increase in x_j adds w_j to the log-odds, which multiplies the odds by e^{w_j}.

DECISION BOUNDARY

$$\mathbf{w}^\top \mathbf{x} + b = \ln \frac{t}{1 - t} \quad (\text{threshold } t), \qquad \mathbf{w}^\top \mathbf{x} + b = 0 \;\text{ for } t = 0.5$$

The boundary is a hyperplane, so logistic regression is a linear classifier. $\mathbf{w}$ is normal to the boundary and points towards the class-1 side. Changing t moves the hyperplane parallel to itself.

LOG-LIKELIHOOD

$$\ell(\mathbf{w}, b) = \sum_{i=1}^{n} \Big[y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \Big], \qquad J(\mathbf{w}, b) = -\frac{1}{n}\, \ell(\mathbf{w}, b)$$

Each sample contributes $\ln P(y_i \mid \mathbf{x}_i)$. Maximising ℓ and minimising J give the same weights.

CROSS-ENTROPY LOSS (NATURAL LOG)

$$J(\mathbf{w}, b) = -\frac{1}{n} \sum_{i=1}^{n} \Big[y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \Big], \quad p_i = \sigma(\mathbf{w}^\top \mathbf{x}_i + b)$$

For one sample the loss is $-\ln p$ if $y = 1$ and $-\ln(1 - p)$ if $y = 0$. Using $\log_2$ instead only multiplies the loss by a constant.

THE SAME LOSS WITH LABELS IN {−1, +1}

$$J(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^{n} \ln\!\big(1 + e^{-y_i z_i}\big), \qquad z_i = \mathbf{w}^\top \mathbf{x}_i + b,\; y_i \in \{-1, +1\}$$

This is identical to the cross-entropy above, rewritten for ± 1 labels using $\sigma(-z) = 1 - \sigma(z)$. Check which label convention a question uses before you substitute.

GRADIENT OF THE LOSS

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{1}{n} \sum_{i=1}^{n} (p_i - y_i)\, \mathbf{x}_i = \frac{1}{n} X^\top (\mathbf{p} - \mathbf{y}), \qquad \frac{\partial J}{\partial b} = \frac{1}{n} \sum_{i=1}^{n} (p_i - y_i)$$

Labels in $\{0, 1\}$. This has the same form as linear regression's gradient when that loss is written $\frac{1}{2n} \sum (\hat{y}_i - y_i)^2$, with p_i in place of the linear prediction. For a single sample, drop the $\frac{1}{n}$ and the sum.

L2-REGULARISED LOSS AND GRADIENT

$$J_\lambda(\mathbf{w}, b) = J(\mathbf{w}, b) + \frac{\lambda}{2n} \|\mathbf{w}\|^2, \qquad \frac{\partial J_\lambda}{\partial \mathbf{w}} = \frac{1}{n} X^\top (\mathbf{p} - \mathbf{y}) + \frac{\lambda}{n} \mathbf{w}$$

$\lambda \geq 0$ sets the strength of the penalty. The bias is usually not penalised, so $\partial J_\lambda / \partial b$ is unchanged. Books scale λ differently (with or without $\frac{1}{n}$ or $\frac{1}{2}$), so use the form the question gives.

GRADIENT DESCENT UPDATE

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial J}{\partial \mathbf{w}}, \qquad b \leftarrow b - \eta \frac{\partial J}{\partial b}$$

$\eta > 0$ is the learning rate. With the squared-error loss $\frac{1}{2}(p - y)^2$ per sample instead, the gradient picks up an extra factor $p(1 - p)$. Learning then slows when the sigmoid saturates, and the loss is no longer convex in $\mathbf{w}$.

EUCLIDEAN, MANHATTAN, MINKOWSKI AND CHEBYSHEV DISTANCE

$$d_2(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_j (a_j - b_j)^2}, \quad d_1(\mathbf{a}, \mathbf{b}) = \sum_j |a_j - b_j|, \quad d_r(\mathbf{a}, \mathbf{b}) = \left(\sum_j |a_j - b_j|^r \right)^{1/r}, \quad d_\infty(\mathbf{a}, \mathbf{b}) = \max_j |a_j - b_j|$$

Euclidean is $r = 2$, Manhattan is $r = 1$, and Chebyshev is the limit $r \to \infty$. d_r is a metric for $r \geq 1$. The order is written r here because p means a predicted probability in these notes. The nearest neighbour can change from one metric to another, so always use the one the question names.

K-NN REGRESSION

$$\hat{y}(\mathbf{x}) = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i$$

$N_k(\mathbf{x})$ is the set of the k nearest training points. A distance-weighted version uses weights proportional to $1/d$. A query that coincides with a training point ($d = 0$) is usually given that point's value.

MIN-MAX AND Z-SCORE SCALING

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \qquad x' = \frac{x - \bar{x}}{s}$$

$\bar{x}$ and s are the mean and standard deviation of the feature. Compute $x_{\min}$, $x_{\max}$, $\bar{x}$ and s from the training data, and apply the same values to the query.

NAIVE BAYES POSTERIOR

$$P(c \mid x_1, \ldots, x_d) = \frac{P(c) \prod_{j=1}^{d} P(x_j \mid c)}{\sum_{c'} P(c') \prod_{j=1}^{d} P(x_j \mid c')}$$

This assumes the features are conditionally independent given the class. The denominator is the same for every class, so you need it only when the question asks for an actual probability.

NAIVE BAYES DECISION RULE (MAP)

$$\hat{c} = \arg\max_c \; P(c) \prod_{j=1}^{d} P(x_j \mid c)$$

Compare the unnormalised scores. In practice the logs are summed, $\ln P(c) + \sum_j \ln P(x_j \mid c)$, to avoid numerical underflow. With equal priors, the rule reduces to picking the class with the largest likelihood (maximum likelihood).

ESTIMATES FROM COUNTS

$$P(c) = \frac{N_c}{N}, \qquad P(x_j = v \mid c) = \frac{N_{j,v,c}}{N_c}$$

N_c is the number of training samples in class c, and $N_{j,v,c}$ is how many of them have feature j equal to v. These are maximum-likelihood estimates and can be zero. Priors are smoothed only if the question says so.

LAPLACE (ADD-$\ALPHA$) SMOOTHING

$$P(x_j = v \mid c) = \frac{N_{j,v,c} + \alpha}{N_c + \alpha V_j}$$

V_j is the number of distinct values feature j can take (2 for a binary feature). $\alpha = 1$ is Laplace smoothing. It stops a single unseen value from forcing the whole product to zero.

MULTINOMIAL NAIVE BAYES (WORD COUNTS)

$$P(w \mid c) = \frac{\text{count}(w, c) + \alpha}{\sum_{w'} \text{count}(w', c) + \alpha\,|V|}, \qquad \text{score}(c) = P(c) \prod_{w} P(w \mid c)^{n_w}$$

Used for text classification. $\text{count}(w, c)$ is the number of times word w appears across the training documents of class c, $|V|$ is the vocabulary size, and n_w is how often w occurs in the document being classified.

GAUSSIAN NAIVE BAYES

$$p(x_j \mid c) = \frac{1}{\sqrt{2\pi\sigma_{jc}^2}} \exp\!\Big(-\frac{(x_j - \mu_{jc})^2}{2\sigma_{jc}^2}\Big)$$

For continuous features. μ_{jc} and σ_{jc}^2 are the mean and variance of feature j within class c (here σ_{jc} is a standard deviation, not the sigmoid). The maximum-likelihood variance divides by N_c and the unbiased one by $N_c - 1$; use the one the question states. The result is a density, so it can exceed 1. Multiply the terms together as before.

NUMBER OF PARAMETERS (D BINARY FEATURES, K CLASSES)

$$\text{naive Bayes: } (K-1) + Kd, \qquad \text{full joint model: } (K-1) + K\,(2^d - 1)$$

The independence assumption makes the parameter count grow linearly in d instead of exponentially. The $K - 1$ term counts the priors, since they sum to 1.

Linear discriminant analysis, support vector machines and decision trees

DISTANCE OF A POINT FROM A HYPERPLANE

$$d(\mathbf{x}_0) = \frac{|\mathbf{w} \cdot \mathbf{x}_0 + b|}{\|\mathbf{w}\|}$$

Euclidean distance from $\mathbf{x}_0$ to the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, for $\mathbf{w} \neq \mathbf{0}$.

LDA PARAMETER ESTIMATES (POOLED COVARIANCE)

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i:\, y_i = k} \mathbf{x}_i, \qquad \hat{\pi}_k = \frac{n_k}{n}, \qquad \hat{\Sigma} = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T$$

Class means, priors and the pooled within-class covariance, where n_k is the number of samples in class k. Dividing by n instead of $n - K$ only rescales $\hat{\Sigma}$. With equal priors the decisions are unchanged. With unequal priors the $\ln \pi_k$ term gets a slightly different relative weight, so borderline decisions can change.

LINEAR DISCRIMINANT FUNCTION

$$\delta_k(\mathbf{x}) = \mathbf{x}^T \Sigma^{-1} \boldsymbol{\mu}_k - \tfrac{1}{2} \boldsymbol{\mu}_k^T \Sigma^{-1} \boldsymbol{\mu}_k + \ln \pi_k$$

Assign $\mathbf{x}$ to the class with the largest $\delta_k(\mathbf{x})$; $\ln$ is the natural logarithm. This assumes Gaussian classes with a common Σ.

TWO-CLASS LDA RULE

$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \qquad \text{predict class 1 if } \mathbf{w}^T \Big(\mathbf{x} - \tfrac{\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1}{2}\Big) + \ln \frac{\pi_1}{\pi_0} > 0$$

This is $\delta_1(\mathbf{x}) - \delta_0(\mathbf{x}) > 0$. The boundary is a hyperplane with normal vector $\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$. It passes through the midpoint of the means when $\pi_0 = \pi_1$. Unequal priors shift it away from the more probable class.

FISHER CRITERION (TWO CLASSES)

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B\, \mathbf{w}}{\mathbf{w}^T S_W\, \mathbf{w}} = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

Here $m_k = \mathbf{w}^T \boldsymbol{\mu}_k$ are the projected means and $s_k^2 = \sum_{i \in k} (\mathbf{w}^T \mathbf{x}_i - m_k)^2$ the projected scatters. Maximise over $\mathbf{w}$. J depends only on the direction of $\mathbf{w}$, not on its length.

FISHER'S OPTIMAL DIRECTION

$$\mathbf{w}^* \propto S_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

The same direction as the two-class LDA weight vector, because $S_W = (n-2)\hat{\Sigma}$ is proportional to the pooled covariance. S_W must be invertible.

MARGIN WIDTH

$$\text{margin width} = \frac{2}{\|\mathbf{w}\|}$$

The distance between the hyperplanes $\mathbf{w}\cdot\mathbf{x} + b = \pm 1$. Valid only under the canonical scaling (closest points at ± 1). The bias b does not affect it.

HARD-MARGIN SVM (PRIMAL)

$$\min_{\mathbf{w},b} \ \tfrac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \ \ i = 1, \ldots, n$$

Requires linearly separable data. Maximising $2/\|\mathbf{w}\|$ is the same as minimising $\tfrac{1}{2}\|\mathbf{w}\|^2$.

HARD-MARGIN SVM (DUAL)

$$\max_{\boldsymbol{\alpha}} \ \sum_i \alpha_i - \tfrac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \, \mathbf{x}_i\cdot\mathbf{x}_j, \quad \alpha_i \geq 0, \ \sum_i \alpha_i y_i = 0, \qquad \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

Only points with $\alpha_i > 0$ contribute to $\mathbf{w}$; these are the support vectors. By complementary slackness, $\alpha_i\big[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\big] = 0$, so every support vector lies exactly on $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$.

SOFT-MARGIN SVM (PRIMAL)

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \ \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{s.t.} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \ \ \xi_i \geq 0$$

In the dual, the only change from the hard margin is the box constraint $0 \leq \alpha_i \leq C$.

SOFT-MARGIN SVM AS HINGE LOSS

$$\min_{\mathbf{w},b} \ \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\max\big(0,\ 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\big)$$

At the optimum $\xi_i = \max\big(0,\ 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\big)$, which turns the constrained problem into this unconstrained one.

COMMON KERNELS

$$K_{\text{lin}}(\mathbf{x},\mathbf{z}) = \mathbf{x}\cdot\mathbf{z}, \qquad K_{\text{poly}}(\mathbf{x},\mathbf{z}) = (\mathbf{x}\cdot\mathbf{z} + c)^p, \qquad K_{\text{RBF}}(\mathbf{x},\mathbf{z}) = \exp\big(-\gamma\|\mathbf{x}-\mathbf{z}\|^2\big)$$

Polynomial kernel: degree p a positive integer and $c \geq 0$. RBF (Gaussian) kernel: $\gamma > 0$, often written $\gamma = 1/(2\sigma^2)$. RBF values lie in $(0, 1]$ and equal 1 only when $\mathbf{x} = \mathbf{z}$.

KERNELISED DECISION FUNCTION

$$f(\mathbf{x}) = \text{sign}\left(\sum_i \alpha_i y_i \, K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Only the support vectors ($\alpha_i > 0$) contribute to the sum.

ENTROPY (IN BITS)

$$H(S) = -\sum_{k=1}^{K} p_k \log_2 p_k, \qquad 0\log_2 0 := 0$$

p_k is the fraction of class k in node S. Entropy ranges from 0 (a pure node) to $\log_2 K$ (all classes equally likely), which is 1 bit for two balanced classes.

GINI INDEX

$$G(S) = 1 - \sum_{k=1}^{K} p_k^2$$

Ranges from 0 (pure) to $1 - 1/K$ (equal classes), which is 0.5 for two balanced classes. For two classes with proportions $p, 1-p$: $G = 2p(1-p)$.

MISCLASSIFICATION ERROR

$$E(S) = 1 - \max_k p_k$$

Ranges from 0 to $1 - 1/K$. For two classes, $E = \min(p, 1-p)$. Entropy and Gini respond more strongly to changes in node purity, so they are the usual criteria for growing a tree.

INFORMATION GAIN OF A SPLIT

$$\mathrm{IG}(S, A) = H(S) - \sum_{v} \frac{|S_v|}{|S|} H(S_v)$$

S_v is the subset sent to child v by the test A. The same weighted-average pattern with G in place of H gives the Gini reduction. Choose the split with the largest gain.

GAIN RATIO

$$\mathrm{GainRatio}(S, A) = \frac{\mathrm{IG}(S, A)}{\mathrm{SplitInfo}(S, A)}, \qquad \mathrm{SplitInfo}(S, A) = -\sum_{v} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

Divides the gain by the entropy (in bits) of the split proportions themselves. This penalises tests that break the node into many small branches.

Bias–variance trade-off and cross-validation

BIAS OF AN ESTIMATOR AT X_0

$$\mathrm{Bias}\left(\hat{f}(x_0)\right) = E\left[\hat{f}(x_0)\right] - f(x_0)$$

Systematic error: how far the average prediction over training sets is from the truth.

VARIANCE OF AN ESTIMATOR AT X_0

$$\mathrm{Var}\left(\hat{f}(x_0)\right) = E\left[\left(\hat{f}(x_0) - E[\hat{f}(x_0)]\right)^2\right]$$

Sensitivity: how much the prediction changes from one training set to another.

BIAS-VARIANCE DECOMPOSITION OF EXPECTED SQUARED ERROR

$$E\left[(y - \hat{f}(x_0))^2\right] = \mathrm{Bias}^2\left(\hat{f}(x_0)\right) + \mathrm{Var}\left(\hat{f}(x_0)\right) + \sigma^2$$

Valid for squared-error loss, with zero-mean noise in the test response that is independent of the training data. The term σ^2 is irreducible: no model can remove it.

RIDGE REGRESSION ESTIMATE

$$\hat{\mathbf{w}}_{\mathrm{ridge}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - X\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2 = (X^\top X + \lambda I)^{-1} X^\top \mathbf{y}$$

Convention: X is $n \times p$ with samples in its rows, $\mathbf{y}$ holds the n responses, and $\mathbf{w}$ the weights. The bias (intercept) b is not penalised: $\mathbf{y}$ and each column of X are centred first, so b drops out and is recovered afterwards. For $\lambda > 0$, $X^\top X + \lambda I$ is always invertible. $\lambda = 0$ gives ordinary least squares (when $X^\top X$ is invertible), the lowest-bias and highest-variance member of the ridge family. $\lambda \to \infty$ drives $\hat{\mathbf{w}} \to \mathbf{0}$ (high bias, low variance). The penalty depends on the scale of each feature, so features are usually standardised first.

EXPECTED SQUARED ERROR OF K-NN REGRESSION AT X_0

$$E\big[(y-\hat{f}_k(x_0))^2\big] = \sigma^2 + \Big[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})\Big]^2 + \frac{\sigma^2}{k}$$

Holds when the training inputs are fixed and $y_i = f(x_i) + \varepsilon_i$ with independent zero-mean noise of variance σ^2. Here $x_{(1)}, \dots, x_{(k)}$ are the k training inputs nearest to x_0, and $\hat{f}_k(x_0)$ is the average of their y values. The middle term is bias squared, which usually grows with k as farther neighbours are averaged in. The variance σ^2/k falls as k grows.

ERROR ON ONE FOLD (REGRESSION, SQUARED ERROR)

$$E_i = \frac{1}{n_i}\sum_{j \in \text{fold } i} \big(y_j - \hat{y}_j\big)^2$$

n_i is the number of samples in fold i, and $\hat{y}_j$ is the prediction of the model trained without fold i. For classification, E_i is the misclassification rate on fold i.

K-FOLD CV ESTIMATE

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k} E_i$$

Assumes equal-sized folds. With unequal folds, use $\sum_{i=1}^{k} (n_i/n)\, E_i$. Number of models trained: k, each on about $n(k-1)/k$ samples.

LEAVE-ONE-OUT (LOO) CV ESTIMATE

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i,\ \hat{f}^{(-i)}(x_i)\big)$$

LOO is k-fold with $k = n$. ℓ is the loss, and $\hat{f}^{(-i)}$ is trained on every sample except the i-th, so n models are trained, each on $n-1$ samples.

LOO SHORTCUT FOR ORDINARY LEAST SQUARES

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2$$

Valid only for linear least squares with squared error, fitted once on all the data. $\hat{y}_i$ is the full-data fit. $h_{ii} = \mathbf{x}_i^\top (X^\top X)^{-1} \mathbf{x}_i$ is the i-th diagonal entry of the hat matrix $H = X(X^\top X)^{-1} X^\top$, with samples in the rows of X. It requires $X^\top X$ to be invertible and $h_{ii} < 1$. If an intercept is fitted, X includes a column of ones. No refitting is needed.

MODELS TRAINED IN A GRID SEARCH

$$N_{\text{models}} = (\text{number of candidate settings}) \times k \ + \ 1 \ (\text{final refit})$$

With several hyperparameters, the number of candidate settings is the product of the grid sizes. The $+1$ is the refit on all the data. Leave it out if the question asks only about the search.

ACCURACY

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Misleading when the classes are very unbalanced.

PRECISION

$$\text{Precision} = \frac{TP}{TP + FP}$$

Of everything predicted positive, the fraction that is truly positive. Undefined if nothing is predicted positive.

RECALL (SENSITIVITY, TRUE POSITIVE RATE)

$$\text{Recall} = \frac{TP}{TP + FN}$$

Of everything truly positive, the fraction found.

SPECIFICITY AND FALSE POSITIVE RATE

$$\text{Specificity} = \frac{TN}{TN + FP}, \qquad \text{FPR} = \frac{FP}{FP + TN} = 1 - \text{Specificity}$$

Specificity is the fraction of true negatives correctly identified.

F1 SCORE

$$F_1 = \frac{2PR}{P + R} = \frac{2\,TP}{2\,TP + FP + FN}$$

The harmonic mean of precision P and recall R. TN does not appear in it.

Multi-layer perceptrons and feed-forward neural networks

PERCEPTRON OUTPUT (STEP ACTIVATION)

$$z = \mathbf{w}^\top \mathbf{x} + b, \qquad \hat{y} = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Targets are 0 or 1. The tie z = 0 is assigned to class 1 here. A question may use the opposite tie rule or targets of -1 and +1, so read the convention given.

PERCEPTRON LEARNING RULE

$$\mathbf{w} \leftarrow \mathbf{w} + \eta\,(t - \hat{y})\,\mathbf{x}, \qquad b \leftarrow b + \eta\,(t - \hat{y})$$

Applied sample by sample, where t is the target. If the sample is classified correctly, t - y is 0 and nothing changes. A missed class-1 sample pulls w towards x, and a missed class-0 sample pushes w away from x.

LAYER-WISE FORWARD PROPAGATION

$$\mathbf{z}^{(l)} = W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \qquad \mathbf{a}^{(l)} = f^{(l)}\left(\mathbf{z}^{(l)}\right), \qquad \mathbf{a}^{(0)} = \mathbf{x}$$

W(l) has one row per unit of layer l and one column per unit of layer l-1. f is applied element-wise, except softmax, which acts on the whole vector.

NUMBER OF PARAMETERS IN A DENSE LAYER

$$n_{l-1}\,n_l + n_l = n_l\,(n_{l-1} + 1)$$

n_l is the number of units in layer l. The network total is the sum over all layers after the input. The first term counts weights, the second counts biases.

SIGMOID AND ITS DERIVATIVE

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \qquad \sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$$

Write the derivative in terms of the output y = sigma(z), so backpropagation needs no recomputation. Also 1 - sigma(z) = sigma(-z).

TANH AND ITS DERIVATIVE

$$\tanh'(z) = 1 - \tanh^2(z)$$

Valid for all real z.

SOFTMAX

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

K output units. Outputs are positive and sum to 1. Adding the same constant to every z_i leaves p unchanged.

TWO-CLASS SOFTMAX IS A SIGMOID

$$K = 2: \qquad p_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} = \sigma(z_1 - z_2)$$

Only the difference of the two logits matters, which is why one sigmoid unit is enough for binary classification.

SQUARED-ERROR LOSS (ONE SAMPLE)

$$L = \tfrac{1}{2}(\hat{y} - t)^2, \qquad \frac{\partial L}{\partial \hat{y}} = \hat{y} - t$$

The factor 1/2 is a convention that cancels the 2 on differentiation. If a question defines L without it, the gradient is doubled.

BINARY CROSS-ENTROPY (ONE SAMPLE, SIGMOID OUTPUT)

$$L = -\big[t \ln \hat{y} + (1 - t) \ln(1 - \hat{y})\big], \qquad \frac{\partial L}{\partial z} = \hat{y} - t$$

t is 0 or 1 and y = sigmoid(z). The gradient with respect to the output unit's net input z simplifies to y - t.

CATEGORICAL CROSS-ENTROPY WITH SOFTMAX

$$L = -\sum_{i=1}^{K} t_i \ln p_i, \qquad \frac{\partial L}{\partial z_i} = p_i - t_i$$

t is a one-hot vector. With true class c, the loss is -ln p_c.

GRADIENT-DESCENT UPDATE

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$$

theta is any weight or bias. All gradients are computed at the old parameter values, and then all parameters are updated together.

OUTPUT-LAYER ERROR SIGNAL FOR SQUARED ERROR

$$\boldsymbol{\delta}^{(L)} = \left(\mathbf{a}^{(L)} - \mathbf{t}\right) \odot f'\left(\mathbf{z}^{(L)}\right)$$

Squared-error loss with factor 1/2, output activation f applied element-wise. For a single output unit this is (y - t) f'(z). For sigmoid output, f' = y(1 - y); for linear output, f' = 1. With cross-entropy and sigmoid or softmax, delta = y - t directly.

HIDDEN-LAYER ERROR SIGNAL

$$\boldsymbol{\delta}^{(l)} = \left(\left(W^{(l+1)}\right)^{\top} \boldsymbol{\delta}^{(l+1)}\right) \odot f'\left(\mathbf{z}^{(l)}\right)$$

The symbol is the element-wise product. The error of a hidden unit is the weighted sum of the errors of the units it feeds, times its own activation slope. Use the weights as they were before this step's update.

WEIGHT AND BIAS GRADIENTS

$$\frac{\partial L}{\partial W^{(l)}} = \boldsymbol{\delta}^{(l)} \left(\mathbf{a}^{(l-1)}\right)^{\top}, \qquad \frac{\partial L}{\partial \mathbf{b}^{(l)}} = \boldsymbol{\delta}^{(l)}$$

For a single weight: dL/dw = delta of the receiving unit times the output of the sending unit.

Clustering: k-means, k-medoids and hierarchical clustering

EUCLIDEAN DISTANCE

$$d_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2}$$

Default distance. k-means is built around its square.

MANHATTAN DISTANCE

$$d_1(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} |x_j - y_j|$$

Sum of absolute coordinate differences. For (1,1) and (4,5): Manhattan = 3 + 4 = 7, Euclidean = 5.

K-MEANS OBJECTIVE (WCSS)

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2$$

Squared Euclidean distance of each point to its own cluster's centroid, summed over all points.

ASSIGNMENT STEP

$$c(i) = \arg\min_{k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

Nearest centroid in Euclidean distance (squaring does not change which centroid is nearest). Ties need a fixed rule: keep the point in its current cluster if that cluster is among the nearest, otherwise take the lowest index.

UPDATE STEP (CENTROID)

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$

Defined for a non-empty cluster. The mean is the point that minimises the sum of squared Euclidean distances to the cluster's points, so this step cannot increase J.

WCSS FROM PAIRWISE DISTANCES

$$\sum_{\mathbf{x} \in C} \|\mathbf{x} - \boldsymbol{\mu}\|^2 = \frac{1}{2|C|} \sum_{\mathbf{x} \in C} \sum_{\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|^2$$

The double sum runs over all ordered pairs (each unordered pair counted twice). Useful when only distances are given. Check: for {1, 2, 4} the right side is 28/6 = 4.667, the same as the left.

SILHOUETTE COEFFICIENT OF POINT I

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}$$

$a(i)$ = mean distance from point i to the other points of its own cluster; $b(i)$ = the smallest, over the other clusters, of the mean distance from i to that cluster's points. $-1 \le s(i) \le 1$: near 1 means well placed, negative means i is on average closer to another cluster. By convention $s(i) = 0$ if i is alone in its cluster. A higher average $s(i)$ over all points indicates a better K.

MEDOID OF A CLUSTER

$$\mathbf{m}_k = \arg\min_{\mathbf{x} \in C_k} \sum_{\mathbf{y} \in C_k} d(\mathbf{x}, \mathbf{y})$$

The member of the cluster with the smallest total dissimilarity to the other members. k-medoids minimises $\sum_k \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{m}_k)$.

SINGLE LINKAGE (NEAREST NEIGHBOUR)

$$D(A,B) = \min_{a\in A,\, b\in B} d(a,b)$$

Closest pair across the two clusters. Prone to chaining: a line of nearby points joins distant groups into one long cluster.

COMPLETE LINKAGE (FARTHEST NEIGHBOUR)

$$D(A,B) = \max_{a\in A,\, b\in B} d(a,b)$$

Farthest pair across the two clusters. Favours compact clusters of similar diameter.

AVERAGE LINKAGE

$$D(A,B) = \frac{1}{|A|\,|B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$$

Mean of all cross-cluster pairwise distances. For the same pair of clusters it lies between the single-linkage and complete-linkage values.

WARD'S LINKAGE

$$D_{\text{Ward}}(A,B) = \frac{|A|\,|B|}{|A|+|B|}\,\|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|^2$$

The increase in total WCSS caused by merging A and B; it merges the pair that increases the WCSS least. Needs Euclidean coordinates (centroids). Software scales the reported height differently, but the order of merges is the same.

CENTROID LINKAGE

$$D(A,B) = \|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|$$

Distance between the cluster means. Needs Euclidean coordinates, and it can give inversions (a later merge lower than an earlier one).

AVERAGE-LINKAGE UPDATE AFTER MERGING A AND B

$$D(A\cup B,\, C) = \frac{|A|\,D(A,C) + |B|\,D(B,C)}{|A|+|B|}$$

Weight the two old distances by cluster size. The plain average of $D(A,C)$ and $D(B,C)$ is wrong unless $|A| = |B|$.

Dimensionality reduction and principal component analysis

SAMPLE MEAN AND CENTRED DATA

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \qquad \mathbf{x}_i^c = \mathbf{x}_i - \bar{\mathbf{x}}$$

Centre each feature (column) to mean zero. X_c denotes the centred matrix, whose rows are $(\mathbf{x}_i^c)^\top$.

SAMPLE COVARIANCE MATRIX

$$S = \frac{1}{n-1}X_c^\top X_c \in \mathbb{R}^{d\times d}$$

Symmetric and positive semi-definite. Diagonal entries are feature variances; off-diagonal entries are covariances. Some texts divide by n instead. The eigenvectors and the proportions of variance do not change; every eigenvalue is scaled by the same factor.

PRINCIPAL DIRECTIONS

$$S\,\mathbf{v}_j = \lambda_j \mathbf{v}_j, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0, \qquad \mathbf{v}_i^\top \mathbf{v}_j = 0 \ (i \neq j)$$

The eigenvectors of the symmetric matrix S are chosen orthonormal. $\mathbf{v}_j$ is the j-th principal direction (component). A direction is fixed only up to sign.

SPECTRAL DECOMPOSITION AND COVARIANCE OF SCORES

$$S = V \Lambda V^\top = \sum_{j=1}^{d} \lambda_j \mathbf{v}_j \mathbf{v}_j^\top, \qquad \frac{1}{n-1} Z^\top Z = \Lambda_k$$

$V = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_d]$ is orthogonal and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$. The scores $Z = X_c V_k$ (defined below) have the diagonal covariance $\Lambda_k = \operatorname{diag}(\lambda_1, \ldots, \lambda_k)$.

VARIANCE ALONG A PRINCIPAL DIRECTION

$$\operatorname{Var}(\mathbf{v}_j^\top \mathbf{x}^c) = \mathbf{v}_j^\top S \mathbf{v}_j = \lambda_j$$

Projections onto different components are uncorrelated: $\mathbf{v}_i^\top S \mathbf{v}_j = 0$ for $i \neq j$.

VARIANCE ALONG ANY UNIT DIRECTION

$$\lambda_d \leq \mathbf{v}^\top S \mathbf{v} \leq \lambda_1 \quad \text{for every } \mathbf{v} \text{ with } \|\mathbf{v}\| = 1$$

The largest possible projected variance is λ_1, reached at $\mathbf{v}_1$. The smallest is λ_d, reached at $\mathbf{v}_d$.

TOTAL VARIANCE

$$\operatorname{tr}(S) = \sum_{j=1}^{d} S_{jj} = \sum_{j=1}^{d} \lambda_j$$

Total variance is unchanged by rotating to the principal axes. For standardised data it equals d.

PROPORTION OF VARIANCE EXPLAINED

$$\mathrm{PVE}_j = \frac{\lambda_j}{\sum_{i=1}^{d} \lambda_i}, \qquad \mathrm{PVE}_{1:k} = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{i=1}^{d} \lambda_i}$$

Choose the smallest k whose cumulative proportion reaches a stated target, such as 90 percent.

PROJECTION (SCORES) ONTO K COMPONENTS

$$\mathbf{z}_i = V_k^\top (\mathbf{x}_i - \bar{\mathbf{x}}) \in \mathbb{R}^k, \qquad Z = X_c V_k$$

$V_k = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_k]$ has orthonormal columns. The coordinate along $\mathbf{v}_j$ is $z_{ij} = \mathbf{v}_j^\top (\mathbf{x}_i - \bar{\mathbf{x}})$.

RECONSTRUCTION AND ERROR

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + V_k \mathbf{z}_i, \qquad \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 = (n-1) \sum_{j=k+1}^{d} \lambda_j$$

The λ_j are eigenvalues of S with divisor $n-1$. With divisor n, the mean squared error $\frac{1}{n}\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$ equals the sum of the discarded eigenvalues exactly. Among all k-dimensional affine subspaces (all rank-k linear reconstructions of the centred data), projecting onto the top k components gives the smallest total squared reconstruction error. Maximising the retained variance and minimising the reconstruction error lead to the same components.

SVD OF THE CENTRED DATA

$$X_c = U \Sigma V^\top, \qquad \lambda_j = \frac{\sigma_j^2}{n-1}$$

The columns of V (right singular vectors) are the principal directions. The σ_j are the singular values in decreasing order. The λ_j are eigenvalues of S with divisor $n-1$.

SCORES FROM THE SVD

$$Z = X_c V_k = U_k \Sigma_k$$

This uses the first k columns of U and V and the top-left $k \times k$ block of Σ. The best rank-k approximation of X_c is $U_k \Sigma_k V_k^\top$.

GRAM-MATRIX ROUTE (USEFUL WHEN D > N)

$$X_c X_c^\top = U \Sigma \Sigma^\top U^\top, \qquad \mathbf{v}_j = \frac{X_c^\top \mathbf{u}_j}{\sigma_j} \; (\sigma_j > 0)$$

The $n \times n$ matrix $X_c X_c^\top$ has the same non-zero eigenvalues σ_j^2 as $X_c^\top X_c$. When $d > n$ it is the smaller matrix to decompose.

FROM IAE ACADEMY

Preparing for GATE DA?

IAE Academy teaches GATE DA in Ballygunge, Kolkata, and online. Counselling is free: ask us about batches, fees or your preparation.

FREE COUNSELLING

Chat on WhatsApp

Call +91 84204 36524

Open every day, 10:30 am to 7:30 pm.

Scan to chat on WhatsApp

THE GATE DA COURSE

Classroom batches in Ballygunge and online batches. Fees and details are on the course page.

Scan: course page

See the course

FREE ON IAEACADEMY.IN

Notes and solved previous-year questions for Machine Learning, topic by topic.

Scan: the notes

Open the notes

Found this useful? Send it to a friend preparing for GATE.

Send to a friend on WhatsApp

IAE Academy free study material. The formulas come from the free Machine Learning notes on iaeacademy.in. Prepared and published by IAE Academy, Kolkata. Free to read and free to share, unaltered and with attribution.

IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE coaching in Kolkata since 2012 · EC, EE, IN, CS and IT, DA, BT and CE · in the classroom and online

227A Rash Behari Ave, Ballygunge, Kolkata 700019

+91 84204 36524 · iaeacademy.in · Updated 5 October 2026

Get directions