

Classical Classification

K-Nearest Neighbors (KNN)

Pros: Simple, no training (lazy), non-linear boundaries, multi-class friendly
Cons: Slow prediction (distances to all points), memory-heavy, curse of dimensionality, sensitive to irrelevant features, needs k & metric tuning

Bayes Optimal Classifier

Pros: Theoretically optimal error, true posteriors, benchmark
Cons: Intractable (needs full distributions)

Naive Bayes

Pros: Very fast, high-dimensional friendly, robust even if independence violated, handles missing values, great for text/spam
Cons: Strong independence assumption, zero-frequency (needs smoothing), poor probability calibration

Linear Discriminant Analysis (LDA)

Pros: Supervised dim. reduction, efficient, optimal for Gaussian data with shared covariance, good separation
Cons: Gaussian + equal covariance assumption, outlier-sensitive, linear boundaries only

Naive Bayes vs LDA

Similar: Both generative, Gaussian assumption (Gaussian NB), Bayes-derived
Diff: NB: diagonal covariance (independence), faster, better with little data LDA: full shared covariance, more accurate if correlations exist, needs more data

Logistic Regression

Pros: Calibrated probabilities, interpretable coefficients, efficient, robust to noise, good for linear separability
Cons: Linear boundary, outlier-sensitive, needs scaling, can overfit high-dim without regularization

Logistic vs NB/LDA

Similar: Probabilistic output, multi-class capable
Diff: Logistic: discriminative (direct $P(y|x)$), no distribution assumption, often better accuracy NB/LDA: generative, strong assumptions, better with tiny/missing data

Support Vector Machine (SVM)

Pros: Max-margin \rightarrow robust, high-dim effective, memory-efficient (support vectors)
Cons: Slow training, parameter-sensitive, poor with overlap, no direct probabilities

Kernel SVM

Pros: Non-linear boundaries, flexible kernels
Cons: $O(n^2 - n^3)$ training, hard to interpret, overfitting risk

Linear vs Kernel SVM

Similar: Max margin, support vectors
Diff: Linear: fast/simple; Kernel: non-linear but slow

Regression & Optimization

Linear Regression

Pros: Simple, interpretable, closed-form, fast
Cons: Linear only, outlier-sensitive, multicollinearity issues

Ridge (L2)

Pros: Handles multicollinearity, reduces overfitting, stable
Cons: No feature selection, bias introduced, λ tuning

LASSO (L1)

Pros: Feature selection (sparsity), high-dim friendly
Cons: Unstable with correlated features (picks one)

Ridge vs LASSO

Similar: Shrink coefficients, regularized linear reg.
Diff: Ridge keeps all; LASSO zeros some out

Kernel Ridge Regression

Pros: Non-linear via kernel, closed-form
Cons: $O(n^3)$, no sparsity

Support Vector Regression (SVR)

Pros: Outlier-robust (ε -tube), non-linear via kernels
Cons: Parameter-sensitive, slow training

Non-linear Regression

Pros: Complex relationships
Cons: Overfitting risk, harder interpretation, expensive

Gradient Descent Variants

Variant	Pros	Cons
Batch GD	Stable, exact gradient, convex guarantee	Slow, high memory
Stochastic GD	Fast, escapes local minima, low memory	Noisy, slow
Mini-batch	Balance, GPU-friendly, lower variance	Batch

Coordinate Descent

Pros: Simple, fast for sparse/L1 (Lasso)
Cons: Slow convergence possible, order-dependent

RANSAC

Pros: Very robust to outliers
Cons: Non-deterministic, threshold & iteration tuning

Unsupervised Learning

K-Means

Pros: Fast, scalable, simple
Cons: Spherical clusters assumed, init-sensitive, needs k

Gaussian Mixture Model (GMM)

Pros: Soft clustering, elliptical clusters, probabilities
Cons: Init-sensitive, overfit risk, heavier computation

K-Means vs GMM

Similar: Centroid-based, EM-like
Diff: K-Means: hard/spherical; GMM: soft/elliptical

Expectation-Maximization (EM)

Pros: Latent variables, monotonic improvement
Cons: Local optima, slow, init-sensitive

Dimensionality Reduction

Pros: Faster computation, noise removal, visualization, curse of dim. mitigation
Cons: Information loss, interpretation harder

Feature Selection vs Dim. Reduction

Similar: Reduce features
Diff: Selection: original subset (interpretable); Reduction: new features

PCA

Pros: Max variance, orthogonal, effective reduction
Cons: Scaling-sensitive, linear only, variance \neq information

SVD

Pros: Numerically stable, general matrix factor
Cons: Linear, expensive for huge matrices (*Underpins PCA*)

Deep Learning

Multi-Layer Perceptron (MLP)

Pros: Universal approximator, non-linear
Cons: Black-box, data-hungry, overfit-prone

Convolutional NN (CNN)

Pros: Image/grid excellence, parameter sharing, translation invariance, hierarchical features
Cons: Data/compute heavy, poor on non-spatial data

MLP vs CNN

Similar: Deep nets, backprop
Diff: MLP: fully connected; CNN: local + shared weights

Regularization Techniques

Technique	Pros	Cons
Dropout	Effective, reduces co-adaptation	Longer training, larger nets needed
Early Stopping	Simple, saves time, prevents overfit	Needs val set, may stop early
Data Augmentation	More data, better generalization	Risk of unrealistic samples, longer train

Data Whitening / Normalization

Pros: Faster convergence, no feature dominance, helps distance algorithms
Cons: Possible test leak, may amplify noise

Weight Initialization

Xavier/Glorot (sigmoid/tanh), He (ReLU), Orthogonal recommended

Learning Curves

- Train ↑, gap large → high variance (regularize / more data)
- Both low, small gap → high bias (bigger model)
- Train ↑, still improving → more data/training

Computer Vision Tasks

- **Object Detection:** Bounding boxes + class (e.g., YOLO)
- **Semantic Segmentation:** Pixel class (same class = same label)
- **Instance Segmentation:** Pixel class + instance distinction

Image Processing

Task	Pros	Cons
Denoising	Better quality/tasks (DnCNN strong)	Blur risk, compute-heavy (deep)
Deblurring	Removes shake/motion	Harder than denoising
Super-Resolution	Enhances details (GANs hallucinate)	Artifacts possible
Colorization	Revives old photos	Subjective/inaccurate
Compression	Storage/transmission savings	Quality loss (lossy)
Fusion	Richer multi-modal info	Alignment issues

Quality Metrics

Metric	Pros	Cons
MSE	Simple, differentiable	Poor human correlation, outlier-sensitive
SSIM	Perceptual (structure)	Complex, scaling/alignment sensitive
LPIPS	Closest to human judgment	Heavy, needs pretrained net

General Tips

- **Feature Normalization:** Essential for distance/gradient methods
- **Output Transformation:** Stabilizes variance, aids assumptions (inverse needed)
- **Model Comparison:** Use accuracy, speed, interpretability, multiple validation metrics