When to use which and why

Classification

🧭 1️⃣ By Type — Families of Classification Algorithms

Type	Core Idea	Common Algorithms
1. Linear Models	Assume a linear relationship between features and output.	- Perceptron / PLA - Logistic Regression - Linear Discriminant Analysis (LDA) - Support Vector Machine (Linear SVM)
2. Probabilistic Models	Model the *probability distribution* of classes and use Bayes' theorem.	- Gaussian Discriminant Analysis (GDA) - Naïve Bayes (Gaussian, Multinomial, Bernoulli) - QDA (Quadratic Discriminant Analysis)
3. Instance-based (Non-parametric)	Store examples; classify based on "closest" training data points.	- k-Nearest Neighbors (kNN) - Kernel-based methods (RBF-SVM)
4. Tree-based Models	Partition the feature space into regions using if–else conditions.	- Decision Trees (CART, ID3) - Random Forests - Gradient Boosted Trees (XGBoost, LightGBM, AdaBoost)
5. Ensemble Methods	Combine multiple models to improve robustness and accuracy.	- Bagging (e.g., Random Forest) - Boosting (e.g., XGBoost, AdaBoost, CatBoost) - Stacking / Blending
6. Neural Models	Use layers of neurons to learn complex decision boundaries.	- Feedforward Neural Network (MLP) - CNNs (for image data) - RNNs / LSTMs (for sequential data)

Type	Core Idea	Common Algorithms
7. Clustering-based / Unsupervised (for pseudo-labeling)	Group unlabeled data, sometimes used for classification via cluster assignment.	- K-Means - Gaussian Mixture Models (GMM) - DBSCAN

🎯 2️⃣ When to Use Which — Intuition and Assumptions

Algorithm	When to Use	Assumptions / Strengths	Weaknesses
PLA (Perceptron)	Data is linearly separable.	Linear boundary can separate classes perfectly.	Doesn't converge if data isn't linearly separable.
Logistic Regression	You expect a *linear* relation between features & log-odds of the outcome.	Linear decision boundary; features independent.	Fails if relationship is non-linear.
LDA (Linear Discriminant Analysis)	You assume data for each class follows Gaussian distribution with *same covariance*.	Equal covariance across classes (homoscedasticity).	If covariances differ, LDA performs poorly.
GDA (Gaussian Discriminant Analysis)	Same as LDA, but allows each class to have its *own covariance matrix*.	Data from each class ~ Gaussian (Normal).	Requires estimating large covariance matrices.
Naïve Bayes	You assume *features are independent given the class*. Works well for text.	Conditional independence; simple to compute.	Unrealistic independence assumption; poor with correlated features.
kNN	When decision boundary is *nonlinear* and you have small data.	"Similar inputs → similar outputs."	Slow for large data; sensitive to feature scaling and irrelevant features.

Algorithm	When to Use	Assumptions / Strengths	Weaknesses
SVM (Linear / Kernel)	Data might not be linearly separable.	Finds hyperplane maximizing margin between classes.	Kernel tuning and scaling matter; not ideal for huge datasets.
Decision Trees	When relationships are nonlinear or features interact.	No need for feature scaling; handles mixed data.	Prone to overfitting.
Random Forest	Want better generalization than a single tree.	Ensemble of uncorrelated trees reduces variance.	Slower, less interpretable.
XGBoost / Boosting	You want high predictive accuracy, even if interpretability is low.	Sequentially reduces residual errors.	Computationally heavy; needs parameter tuning.
K-Means	You have unlabeled data and want to find natural clusters.	Assumes spherical clusters of similar size.	Poor with non-convex or varying-size clusters.

🧮 3️⃣ Key Assumptions that Separate Them

Comparison	Assumption That Matters
PLA vs Logistic Regression	PLA assumes data is strictly linearly separable; Logistic Regression doesn’t—it uses probabilities and a soft boundary.
LDA vs GDA	Both assume Gaussian data. LDA assumes same covariance across classes; GDA allows each class to have its own covariance.
LDA vs Logistic Regression	Both give linear decision boundaries; LDA assumes Gaussian generative process, Logistic Regression directly models conditional probability.
Naïve Bayes vs Logistic Regression	Naïve Bayes assumes independence of features; Logistic Regression allows correlation (but linear relation).
KNN vs SVM	KNN is non-parametric (no assumptions on data shape); SVM assumes a specific margin-maximization principle.

Comparison	Assumption That Matters
Decision Trees vs Linear Models	Trees make axis-aligned splits, don’t assume linearity; Linear models assume features interact linearly.
Random Forest vs Boosting	Bagging (Random Forest) reduces variance; Boosting reduces bias sequentially.

⚡ 4️⃣ Practical Heuristics — "Rules of Thumb"

Situation	Try This First
Small dataset, simple relationships	Logistic Regression or LDA
Large dataset, nonlinear patterns	Random Forest or XGBoost
High-dimensional sparse data (text, NLP)	Naïve Bayes or Linear SVM
Low data, noisy labels	Regularized Logistic Regression
No labels, want structure	K-Means or GMM
Image / complex pattern recognition	Neural Networks (CNN)
Real-time or incremental updates	Online SVM or SGD-based models

🧠 TL;DR Summary

Family	Core Idea	When Best
Linear	Fast, interpretable, low variance	Linearly separable, small data
Probabilistic	Estimate probability distributions	When Gaussian or independence assumptions hold
Instance-based	Local neighborhood reasoning	Small, nonlinear datasets
Tree-based	Recursive space partitioning	Mixed or nonlinear data
Ensemble	Combine models for robustness	When accuracy > interpretability
Neural	Deep non-linear learning	Large, complex, unstructured data

Regression

🧮 Categorized by Type

Type / Family	Core Idea	Common Algorithms	When to Use
🟩 **Linear Models**	Assume a linear relationship between input features and the target variable y .	- Simple Linear Regression - Multiple Linear Regression - Ridge Regression (L2) - Lasso Regression (L1) - Elastic Net - Bayesian Linear Regression	When data is approximately linear, features are independent, and interpretability matters.
🟦 **Polynomial / Nonlinear Extensions**	Extend linear models by adding polynomial or local transformations of features.	- Polynomial Regression - Spline Regression - Locally Weighted Regression (Loess / LWR)	When the relationship is nonlinear but smooth or low-dimensional.
🟧 **Tree-Based Methods**	Split feature space into regions and predict the mean of each region.	- Decision Tree Regression - Random Forest Regression - Gradient Boosting Regression - XGBoost / LightGBM / CatBoost	For nonlinear, tabular data with feature interactions or mixed types.
🟨 **Kernel and Instance-Based Methods**	Use distance or similarity between samples for prediction.	- k-Nearest Neighbors (kNN) Regression - Support Vector Regression (SVR) - Gaussian Process Regression (GPR)	When data is non-linear or locally structured. Works best on small to medium datasets.
🟪 **Probabilistic / Generative Models**	Model the joint distribution P(X, y) and infer the conditional $P(y\|X)$	- Bayesian Regression - Gaussian Mixture Regression - Hidden Markov Regression	When you want uncertainty estimation or when data is generated from underlying distributions.

Type / Family	Core Idea	Common Algorithms	When to Use
🟥 **Neural Network Based**	Learn complex nonlinear mappings using layers of neurons.	- Feedforward Neural Network (MLP) - Convolutional Neural Network (CNN) - Recurrent Neural Network (RNN / LSTM) - Transformer Regression	For large datasets, unstructured data (images, sequences), or complex nonlinear patterns.
🟫 **Ensemble & Meta Learning**	Combine predictions from multiple weak or strong models.	- Bagging Regressor - AdaBoost Regressor - Stacking Regressor	When you want to improve accuracy and robustness by reducing bias or variance.
⚪ **Time-Series Regression**	Model dependencies between observations over time.	- ARIMA / SARIMA - VAR (Vector AutoRegression) - Prophet (Facebook) - LSTM / Transformer-based Time-Series	For forecasting or sequence data where temporal order matters.

🟩 1️⃣ Linear Models

Assume a **linear relationship** between input features and the target variable (y).

Algorithm	Core Idea	Mathematical Formulation	When to Use	Notes / Assumptions
Simple Linear Regression	Models a straight-line relationship between one feature and the target.	$y = w_0 + w_1 x + \epsilon$	When there is only one independent variable and the relationship is roughly linear.	Assumes linearity, independence, homoscedasticity, and normally distributed errors.
Multiple Linear Regression	Extends simple regression to multiple predictors.	$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \epsilon$	When multiple features explain the target variable linearly.	Same assumptions as simple regression, plus no multicollinearity.

Algorithm	Core Idea	Mathematical Formulation	When to Use	Notes / Assumptions
Ridge Regression (L2 Regularization)	Adds L2 penalty to shrink coefficients and reduce overfitting.	Minimize $\sum(y_i - \hat{y}_i)^2 + \lambda \sum w_j^2$	When multicollinearity exists or overfitting occurs.	Coefficients shrink but none go to zero. Smooth solution.
Lasso Regression (L1 Regularization)	Adds L1 penalty, which can drive some coefficients to zero (feature selection).	Minimize $\sum(y_i - \hat{y}_i)^2 + \lambda \sum \lvert w_j \rvert$	When feature selection or sparse models are needed.	Can eliminate less important features entirely.
Elastic Net Regression	Combines L1 and L2 penalties (balance of Ridge & Lasso).	Minimize $\sum(y_i - \hat{y}_i)^2 + \alpha[\lambda_1 \sum \lvert w_j \rvert + \lambda_2 \sum w_j^2]$	When you have many correlated features or need both shrinkage and sparsity.	Balances bias–variance trade-off.
Bayesian Linear Regression	Introduces prior distributions on weights and finds posterior estimates.	$P(w \mid X, y) \propto P(y \mid X, w)P(w)$	When you need probabilistic predictions and uncertainty estimation.	Regularization comes from priors. Works well with small data.

🟦 2️⃣ Polynomial / Nonlinear Extensions

Model relationships that are *nonlinear* but still smooth.

Algorithm	Description	When to Use
Polynomial Regression	Adds polynomial terms: ($x, x^2, x^3, \ldots$)	When data follows a curved trend
Spline Regression	Fits piecewise polynomials	When data has multiple smooth regions

Algorithm	Description	When to Use
Locally Weighted Regression (Loess/LWR)	Fits local linear models around each point	When you need flexible fitting for small datasets

🟧 3️⃣ Tree-Based Methods

Predict (y) by recursively splitting data into regions and taking mean values.

Algorithm	Description	When to Use
Decision Tree Regression	Piecewise constant model	When relationships are non-linear or contain thresholds
Random Forest Regression	Average of many trees (bagging)	Robust to overfitting, good general-purpose model
Gradient Boosting Regression	Sequentially adds trees to reduce residuals	High accuracy, strong performance
XGBoost / LightGBM / CatBoost	Optimized versions of boosting	Large datasets, competitions, tabular data

🟨 4️⃣ Kernel and Instance-Based Methods

Use similarity (distance) between points to predict new values.

Algorithm	Description	When to Use
k-Nearest Neighbors (kNN) Regression	Predicts average of nearest k samples	When data is small, non-linear, and noisy
Support Vector Regression (SVR)	Uses hyperplanes with margin to fit data	When you want robustness to outliers
Gaussian Process Regression (GPR)	Non-parametric Bayesian regression	When you need uncertainty estimates

🟪 5️⃣ Probabilistic / Generative Models

Model the **joint distribution** (P(X, y)) and then derive predictions.

Algorithm	Description	When to Use
Bayesian Regression	Places prior on weights, infers posterior	When you want uncertainty and small data

Algorithm	Description	When to Use
Gaussian Mixture Regression	Combines multiple Gaussians to model complex data	When data is multimodal or clustered
Hidden Markov Regression	Sequential regression under hidden states	Time-series or sequential data

🟥 6️⃣ Neural Network Based

Use deep architectures for complex non-linear mappings.

Algorithm	Description	When to Use
Feedforward Neural Network (MLP)	Multi-layer perceptron for regression	Nonlinear relationships, medium data
Convolutional Neural Network (CNN)	Regression on image data (e.g., age estimation)	Spatial or image-based regression
Recurrent Neural Network (RNN / LSTM)	Time-series regression	Sequential or temporal data
Transformer Regression	Attention-based sequence modeling	Complex temporal or high-dimensional input

🟫 7️⃣ Ensemble & Meta Learning

Combine predictions from multiple models.

Algorithm	Description	When to Use
Bagging Regressor	Average predictions of multiple base regressors	Reduce variance
AdaBoost Regressor	Sequentially improves weak learners	Moderate non-linearity
Stacking Regressor	Combines outputs of various regressors	When you want maximum accuracy

⚪ 8️⃣ Regression for Time-Series

Used when observations are time-dependent.

Algorithm	Description	When to Use
ARIMA / SARIMA	Linear auto-regressive models	Stationary or seasonal time-series
VAR (Vector AutoRegression)	Multi-variable time-series	Multiple related time series
Prophet (Facebook)	Trend + seasonality + holidays model	Forecasting business/time-based data
LSTM / Transformer	Deep learning for sequences	Long-term dependencies

🧭 Quick Selection Guide

Situation	**Recommended Algorithm**
Linear & simple	Linear Regression, Ridge, Lasso
Non-linear but smooth	Polynomial or SVR
Complex tabular data	Random Forest / XGBoost
High-dimensional features	Lasso / Elastic Net
Small data, uncertain predictions	Bayesian Regression / GPR
Sequential / temporal	LSTM / ARIMA
Few data points, local patterns	kNN Regression
Image or spatial data	CNN Regression

🧠 Summary by Family

Family	**Core Assumption**	**Best For**
Linear	Straight-line relationships	Interpretable, fast models
Tree-based	Piecewise constant functions	Tabular data with interactions
Kernel / Instance	Local smoothness	Non-parametric, flexible
Probabilistic	Data generated from a distribution	Uncertainty estimation
Neural	Hierarchical feature learning	Complex nonlinear data
Ensemble	Model combination	High accuracy, low bias/variance