

Dimension Reduction	Compare Algorithms					
	Model	(kernel) PCA	LDA	UMAP	Factor Analysis	Autoencoder
	Type	Statistical, linear / nonlinear if with kernel	Linear, supervised	Nonlinear, manifold learning	Statistical, linear	Neural Network
	Architecture	Orthogonal components maximizing variance	Linear projection maximizing class separation	Optimizing the low-dimensional representation to match the fuzzy topological representation	Observed variables as linear combinations of latent factors and noise	Encoder-decoder neural network with bottleneck layer
	Speed	Fast / Slower if with kernel	Fast	Moderate	Fast	Slow
	Best For	Reducing dimensionality of linearly correlated features / nonlinear structure in small-to-medium datasets if with kernel	Supervised dimensionality reduction for classification	Visualization; clustering-friendly; nonlinear dimensionality reduction	Identifying latent factors; dimensionality reduction with noise modeling	Nonlinear dimensionality reduction; denoising; feature extraction for other ML tasks
	Limitations	Cannot capture nonlinear patterns / Computationally expensive for large datasets; choice of kernel critical; less interpretable if with kernel	Requires labels; assumes normality and equal covariance; linear boundaries only	Sensitive to hyperparameters; distort global distances; less interpretable	Assumes linearity; needs sufficient data; cannot capture nonlinear relationships	Needs large dataset; uninterpretable; computationally intensive
	Corresponding Package	scikit-learn	scikit-learn	umap-learn	scikit-learn	pytorch
	TL;DR					
	If you want statistical analysis of how predictors affect target, Factor Analysis					
	If you want to do subsequent visualization of clusters, UMAP					
	If you have small sample size or require interpretability, PCA or LDA					
	If you have large sample size or want to train other ML models, Autoencoder					