

Regression	Compare Algorithms						
	Model	Linear Regression	XGBoost	MLP	Transformer	TabPFN	All-in-One (AutoGluon)
	Type	Statistical	Decision Trees	Neural Network	Neural Network	Neural Network	Ensemble
	Architecture	Linear, additive	Ensembles of gradient-boosted trees	Stacks of fully connected layers	Stacks of multi-head self-attention layers	Pretrained Transformer that simulates Bayesian inference	Run numerous models and get weighted average estimation
	Speed	Instant	Moderate	Moderate to slow	Slow	Fast at inference	Slowest
	Accuracy	Low	High	Moderate	With potential to be excellent	High for small-to-medium datasets	Best
	Best For	Known linear relationships, interpretable models	Medium-sized datasets; high accuracy; easy training; partially interpretable	Nonlinear patterns; medium-to-large datasets where relationships are complex	Complex relational dependencies between features; large datasets; high accuracy	Small-to-medium datasets; few-shot; very fast inference	Crave for best accuracy; compare models
	Limitations	Cannot capture nonlinear relationships or interactions without feature engineering	Cannot automatically handle complex feature interactions, needs encoding for categorical features	Requires careful hyperparameter tuning; needs sufficient data; black-box; may overfit small datasets	Requires careful hyperparameter tuning; very computationally intensive; slow training; requires large datasets	Less flexible for custom tasks; pretrained model may not generalize to very different feature distributions; require strong gpu	Time consuming; infeasible to deploy on edge devices
	Corresponding Package	statsmodels	XGBoost	Pytorch	Pytorch	TabPFN	AutoGluon
	TL;DR						
XGBoost and TabPFN are good choices for most cases							
If you have large sample size and crave for even better accuracy, try MLP and Transformer							
If you want best accuracy but don't need to deploy, try All-in-One (AutoGluon)							