

Multi-Variate Time Series Forecasting	Compare Packages					
	Feature / Aspect	Darts			NeuralForecast	
	License	Apache 2.0			Apache 2.0	
	Primary Focus	General-purpose time series forecasting and analysis (statistical + ML + deep learning)			State-of-the-art neural forecasting models, especially for large-scale datasets	
	Model Support	Very broad: ARIMA, Exponential Smoothing, Regression, RNN/LSTM/GRU, TCN, Transformers, GBDTs			Focused on neural models: N-BEATS, N-HITS, DeepAR, TFT, PatchTST, Informer, etc. (many recent SOTA architectures)	
	Ease of Use	High — sklearn-like API (fit(), predict()); integrates statistical + ML + DL under one interface			Moderate — aimed at researchers/practitioners comfortable with deep learning; API less beginner-friendly than Darts	
	Training Flexibility	Good balance — can train deep learning models, use prebuilt statistical models, or plug in custom models			Very flexible for DL research — supports multivariate series, multi-horizon forecasting, scalable training (PyTorch Lightning)	
	Deployment Support	Better for production pipelines (integrates with joblib, ONNX export for some models, easy serialization)			Research-first; deployment support is limited, but models can be exported since it's PyTorch-based	
	Community & Ecosystem	Large and active			Smaller but growing	
	Documentation	Very strong — detailed docs, tutorials, notebooks for all models			Good, but research-oriented — examples exist, but not as user-friendly as Darts	
	Installation Complexity	Simple			Simple, but heavier DL dependencies (PyTorch Lightning, etc.)	
	Compare Algorithms					
	Model	VARIMA	RNN	TSMixer	iTransformer	StemGNN
	Type	Statistical	Neural Network (Recurrent cell)		Neural network (Transformer)	Neural network (GNN)
	Architecture	Linear, additive	Sequential model with recurrent cells	Mixer-style: separates temporal and feature mixing using stacked MLP blocks	Transformer variant: treats each variable as a token and time as feature dimension; focuses on inter-variable relationships	Combines Graph Convolutions (capture inter-series dependencies) + Spectral Temporal Convolutions (Fourier + temporal convs)
	Speed	Instant	Moderate to slow (depend on window size)	Fast	Moderate	Slowest
	Accuracy	Low	High for large dataset	High for small dataset	Very high	High for datasets with strong graph-like structures
	Best For	Known linear relationships, interpretable	Complex time series with non-linear relationships and long memory.	When efficiency and scalability are important	Time series with strong variable interactions; long-horizon forecasting	Variables can be viewed as nodes in a graph, with oscillations propagating to adjacent nodes.
	Limitations	Cannot capture nonlinear relationships or complex patterns	Slow; require large dataset; uninterpretable	May underperform on very complex temporal dependencies	Computationally heavier; needs large dataset; less efficient for very high-dimensional time series	Computationally expensive; requires graph structure or learns one implicitly; less general for arbitrary datasets
	Corresponding Package	Darts	Darts	Darts, NeuralForecast	NeuralForecast	NeuralForecast
	TL;DR					
	If you want quick prototyping - Darts					
	If you want more advanced models - NeuralForecast					