Time Series, Forecasting & Agentic AI Pipelines

Concepts & Terms Reference Guide

Graduate Capstone in Data Science

Week: Time Series · Forecasting · Autoregression · Smolagents

10 Sections	70+ Terms Defined	5 Libraries Covered	2 LLM Providers

Table of Contents

9. Learning Paradigms (Review from Last Week)

10. Financial Time Series Concepts

How to use this guide: *Terms in each section build on one another. The left column shows the term or concept; the right column provides a practical explanation with real-world context. Use this alongside the workshop notebook — each discussion question maps to terms in this guide.*

1. Time Series Fundamentals

Term	Explanation
Time Series	A sequence of data points recorded at successive, equally spaced points in time. Unlike cross-sectional data, the order of observations matters — the temporal position of each data point carries predictive information about future values.
Stationarity	A time series is stationary when its statistical properties — mean, variance, and autocorrelation structure — remain constant over time. Most classical forecasting models assume stationarity and will produce unreliable results if this assumption is violated.
Non-stationarity	When a series has a changing mean, variance, or both over time. Stock prices are a canonical example: they trend upward (or downward) over years, and their volatility shifts across market regimes. Non-stationary series must be transformed (e.g., differenced) before classical modeling.
Trend	The long-term directional movement in a time series, distinct from short-term fluctuations. Trends can be linear, exponential, or irregular. Identifying and removing trend is often a necessary preprocessing step before fitting a model.
Seasonality	Repeating, predictable patterns in a time series tied to a fixed calendar period — daily, weekly, monthly, or yearly. In equity markets, seasonality might reflect quarterly earnings cycles, year-end tax-loss harvesting, or recurring 'January effects' in stock returns.
Residual / Noise	The component that remains after trend and seasonality are removed. A good model leaves residuals that behave like white noise — random, with no remaining autocorrelation. If residuals still contain patterns, the model has not captured all available signal.
White Noise	A series of uncorrelated random variables with zero mean and constant variance. It is the idealized 'nothing left to model' benchmark. After fitting a forecasting model, inspecting whether residuals are white noise is the primary diagnostic check for model adequacy.
Structural Break	A sudden, permanent shift in the statistical properties of a time series, often caused by an external shock — a financial crisis, regulatory change, or pandemic. Structural breaks violate stationarity and can cause models trained on historical data to fail catastrophically in the changed regime.

Term	Explanation
Autocorrelation	The correlation of a time series with a lagged version of itself. Positive autocorrelation means high values tend to follow high values (momentum); negative autocorrelation means values tend to alternate. Autocorrelation structure is the core property that makes forecasting possible.
Lag	A previous time step in a series. 'Lag 1' refers to the immediately preceding value, 'lag 5' to the value five periods ago. Lags are the fundamental building block of autoregressive models and of lag-feature engineering for supervised ML.

2. Statistical Tests & Diagnostics

Term	Explanation
ADF Test	Augmented Dickey-Fuller test. A formal hypothesis test for the presence of a unit root. H0: the series has a unit root (non-stationary). H1: the series is stationary. A p-value below 0.05 leads us to reject H0 and conclude stationarity. Always run this before fitting an ARIMA model.
Unit Root	A property of a time series where shocks have a permanent rather than transitory effect on the level of the series. A series with a unit root has no tendency to revert to a long-run mean. Stock prices are widely believed to have a unit root (the Random Walk hypothesis).
Differencing	Transforming a series by computing successive differences: diff[t] = y[t] - y[t-1]. First-order differencing removes linear trends and converts most non-stationary financial series to stationary ones. It is the 'I' (Integrated) component in ARIMA(p, d, q), where d is the differencing order.
ACF — Autocorrelation Function	A plot showing the correlation between a time series and each of its lagged values. Used diagnostically to identify the MA order (q) for ARIMA: lags where the ACF 'cuts off' (drops inside the confidence band) suggest the MA order. Also used post-modeling to check that residuals are uncorrelated.
PACF — Partial Autocorrelation Function	The correlation between a series and a given lag after removing the linear influence of all shorter lags. Used to identify the AR order (p) for ARIMA: lags where the PACF cuts off suggest the AR order. PACF isolates the 'direct' relationship between a value and its specific lag.

Term	Explanation
Ljung-Box Test	A portmanteau test that evaluates whether any group of autocorrelations in residuals is significantly different from zero. A p-value above 0.05 indicates residuals are consistent with white noise, confirming the model has captured the available structure. A common final check after fitting ARIMA.
Seasonal Decomposition	The process of separating a time series into its constituent components. Additive: Y = Trend + Seasonal + Residual (variance is constant). Multiplicative: Y = Trend x Seasonal x Residual (variance grows with level). Financial series typically use additive decomposition on log-transformed prices.
Information Criteria (AIC/BIC)	Model selection metrics that balance goodness-of-fit against model complexity. AIC (Akaike) and BIC (Bayesian) penalize models with more parameters. Lower values are better. Use these to compare ARIMA models with different p and q orders without overfitting to the training data.

3. Forecasting Models

Term	Explanation
AR — AutoRegressive Model	A model that regresses the current value of a series on its own past values. AR(p) uses the p most recent lags as predictors. Captures momentum and mean-reversion patterns. The key assumption is that the process is stationary and that past values linearly predict future values.
MA — Moving Average Model	A model where the current value is expressed as a linear function of past forecast errors (residuals), not past values of the series itself. MA(q) uses the q most recent error terms. Important: this is distinct from a simple rolling average. MA captures the lingering effect of random shocks on the series.
ARIMA(p, d, q)	AutoRegressive Integrated Moving Average. Combines AR(p) lags, d rounds of differencing to achieve stationarity, and MA(q) error terms. The standard workhorse of classical time series forecasting. Parameter selection is guided by ADF testing (for d), PACF (for p), and ACF (for q).
Walk-Forward Validation	The gold-standard evaluation method for time series models. At each step: train on all available historical data up to time t, predict t+1, observe the actual value, add it to the training set, and advance. Prevents look-ahead bias and mimics how a model would actually be deployed in production.

Term	Explanation
Lag Feature Engineering	Converting a time series forecasting problem into a supervised regression problem by creating lag columns as features. If y[t] depends on y[t-1], y[t-2], ..., y[t-k], those lagged values become features X and y[t] becomes the target. This lets you apply any sklearn model to time series data.
Rolling Statistics	Summary statistics (mean, standard deviation, min, max) computed over a sliding window of fixed size. Rolling means smooth out noise and capture local trends. Rolling standard deviation measures local volatility — a critical feature in financial time series modeling.
Forecast Horizon	The number of time steps ahead that a model predicts. One-step-ahead forecasts are the most accurate. Accuracy typically degrades with longer horizons as uncertainty compounds. Multi-step forecasting strategies include recursive (feeding predictions back as inputs) and direct (training separate models for each horizon).
Confidence / Prediction Interval	A range around a point forecast that quantifies uncertainty. A 95% prediction interval means that 95% of actual future values should fall within it. Intervals widen with forecast horizon, reflecting increasing uncertainty over time. Prophet provides these automatically; ARIMA computes them analytically.

4. Prophet-Specific Concepts

Term	Explanation
Prophet	An open-source forecasting library developed by Meta (Facebook). It models time series as the sum of trend, seasonality, and holiday effects, using an additive decomposition. Designed to be robust to missing data and outliers, and to require minimal manual tuning. Widely used in industry for business metric forecasting.
Changepoint	A point in time where Prophet detects a significant shift in the trend. Prophet automatically identifies potential changepoints and fits trend changes at those locations. The changepoint_prior_scale parameter controls how flexible the trend is — larger values allow more dramatic trend shifts.
Fourier Terms	Prophet models seasonality using Fourier series — a sum of sine and cosine waves of different frequencies. More Fourier terms capture more complex seasonal patterns but risk overfitting. The seasonality_prior_scale controls the magnitude of seasonal effects.

Term	Explanation
Holiday Effects	Prophet allows users to specify known holidays or special events that cause systematic deviations from expected patterns. Prophet learns the average effect of each holiday on the series and incorporates it into forecasts. Critical for retail, e-commerce, and any business with holiday-driven demand.

5. Model Evaluation Metrics

Term	Explanation
MAE — Mean Absolute Error	The average of the absolute differences between forecasts and actuals: MAE = mean(|actual - forecast|). Expressed in the same units as the target variable (e.g., dollars for stock prices). Robust to outliers. Easy to explain to non-technical stakeholders: 'on average, our forecast is off by $X.'
RMSE — Root Mean Squared Error	The square root of the average squared forecast errors: RMSE = sqrt(mean((actual - forecast)^2)). Penalizes large errors more heavily than MAE due to squaring. Expressed in the same units as the target. RMSE > MAE when there are large individual errors — useful for detecting outlier-driven model failures.
MAPE — Mean Absolute Percentage Error	The average absolute percentage deviation: MAPE = mean(|actual - forecast| / |actual|) x 100%. Unit-free, making it easy to compare accuracy across different series or scales. Problematic when actuals approach zero (division by zero). Often used in financial and business forecasting for its interpretability.
R-squared (R2)	The proportion of variance in the target variable explained by the model, ranging from negative infinity to 1.0. R2 = 1 means perfect prediction; R2 = 0 means the model is no better than predicting the mean. A high R2 on time series data can be misleading — models that simply predict 'tomorrow = today' often score very high on R2.
Train / Test Split	Partitioning data into a training set (used to fit the model) and a test set (used to evaluate out-of-sample performance). For time series, splits MUST be chronological — all training data precedes all test data. Random splitting causes look-ahead bias and produces artificially optimistic results, a common interview mistake.

6. Sklearn for Time Series

Term	Explanation
Supervised Time Series	Reframing a forecasting problem as a supervised regression: lagged values of the series become features (X) and the next value becomes the target (y). This allows any sklearn regression model — linear regression, random forest, gradient boosting — to be applied to time series data without specialized time series libraries.
Feature Importance	A model-level metric (available in tree-based sklearn models like Random Forest and Gradient Boosting) that indicates how much each feature contributed to the model's predictions. In time series with lag features, lag_1 typically dominates, reflecting the strong autocorrelation in most real-world series.
Random Forest Regressor	An ensemble of decision trees that averages predictions to reduce variance. Handles non-linear relationships between lag features and targets, captures interactions, and is robust to outliers. Does not extrapolate beyond training data range — a critical limitation for long-horizon financial forecasting.
StandardScaler	A preprocessing step that standardizes features to have zero mean and unit variance: X_scaled = (X - mean) / std. Required for models sensitive to feature scale (e.g., linear regression, SVMs, neural networks). Tree-based models (Random Forest, XGBoost) are scale-invariant and generally do not require scaling.

7. Agentic AI & Smolagents

Term	Explanation
Agentic AI	An AI system that can plan, take actions, use tools, and adapt its behavior based on feedback from its environment — beyond simply responding to a single prompt. Agentic systems are used to automate complex multi-step workflows in data pipelines, research, and production ML systems.
Smolagents	A lightweight open-source agentic framework from HuggingFace. It enables LLMs to use custom Python tools to accomplish tasks. The CodeAgent variant works by having the LLM write and execute Python code to call tools and synthesize results — more powerful than simple ReAct agents for data science tasks.
Tool Use	The ability of an LLM to call external functions (tools) to access information or perform actions beyond its training data. In Smolagents, tools are decorated Python functions. The LLM reads the type hints and docstrings to understand what each tool does and when to invoke it.

Term	Explanation
CodeAgent	A Smolagents agent class where the LLM reasons by writing and executing Python code. Unlike tool-calling agents that call predefined functions directly, CodeAgent writes programs that orchestrate tool calls, process results, and build toward the final answer. More flexible for complex data science workflows.
@tool Decorator	The Smolagents decorator that registers a Python function as an agent-callable tool. Type hints are mandatory — the agent uses them to understand input and output types. Docstrings are critical — the agent reads them to decide when and how to use the tool. Well-written docstrings are as important as the function logic.
ReAct Pattern	Reasoning + Acting: a prompting strategy where an agent alternates between thinking (reasoning about what to do next) and acting (calling a tool or taking an action). Introduced by Yao et al. (2022). The basis of most production agentic frameworks including Smolagents, LangChain, and LlamaIndex.
Max Steps	A safety parameter in agentic frameworks that caps the number of reasoning-action cycles the agent can take before returning a final answer. Prevents infinite loops, runaway API costs, or pathological reasoning chains. In production systems, this is paired with timeouts and cost budgets.
Provider Abstraction	Designing agentic pipelines so the underlying LLM provider (GROQ, HuggingFace, OpenAI, Anthropic) can be swapped without rewriting application code. Critical in production for cost optimization, avoiding vendor lock-in, and fallback resilience. Smolagents supports this via the HfApiModel interface.

8. LLM APIs — GROQ & HuggingFace

Term	Explanation
GROQ	A hardware and inference company offering extremely fast LLM inference via a cloud API. Uses custom Language Processing Units (LPUs) to achieve token generation speeds far exceeding GPU-based providers. Free tier supports thousands of requests per day. Compatible with the OpenAI API schema.
HuggingFace Inference API	HuggingFace's hosted inference service for open-source models. Provides free-tier access to thousands of models including Llama, Qwen, Mistral, and BERT variants. The InferenceClient Python SDK standardizes the interface across different model types. Essential for production systems using open-weight models.

Term	Explanation
LLM Temperature	A parameter (0.0 to 2.0) that controls the randomness of LLM outputs. Temperature = 0 produces deterministic, focused responses — ideal for structured data analysis tasks. Higher temperatures introduce creativity and variation. For forecasting agents, low temperatures (0.1–0.3) are recommended to ensure consistent, reliable tool calls.
System Prompt	Instructions provided to an LLM before the user message that define the model's persona, constraints, and behavioral guidelines. In agentic pipelines, the system prompt specifies the agent's role (e.g., 'quantitative analyst'), output format expectations, and any domain-specific context the model needs to perform well.
Token	The basic unit of text processed by an LLM — roughly 3/4 of a word on average in English. API pricing and rate limits are denominated in tokens. A typical analyst report of 400 words is approximately 500–600 tokens. Awareness of token consumption is essential for managing API costs in production pipelines.
Open-weight Model	An LLM whose weights are publicly released, allowing anyone to run, fine-tune, or deploy the model. Examples: Llama 3 (Meta), Qwen (Alibaba), Mistral (Mistral AI). Contrast with closed models like GPT-4 or Claude, which are only accessible via API. Open-weight models are central to cost-effective production ML systems.

9. Learning Paradigms (Review from Last Week)

Term	Explanation
Supervised Learning	Learning from labeled data — input-output pairs (X, y) — where the model learns a mapping from inputs to outputs. In time series: X = lag features, y = next value. Most sklearn models and neural networks used in industry are trained under this paradigm.
Unsupervised Learning	Learning patterns from unlabeled data. In time series, unsupervised methods include clustering similar time series, anomaly detection, and dimensionality reduction of multivariate series (e.g., PCA on returns across many stocks).
Reinforcement Learning	An agent learns to take actions in an environment to maximize cumulative reward. Applied to time series in algorithmic trading (the agent learns a buy/sell/hold policy) and in agentic pipelines (the agent learns which tools to call to efficiently accomplish a task through trial and feedback).

Term	Explanation
Federated Learning	A distributed learning paradigm where models are trained across multiple decentralized data sources without sharing raw data. Applied to time series when data is sensitive (e.g., financial records across banks) or too large to centralize. Each node trains locally; only model gradients or parameters are shared and aggregated.
Pydantic AI	A Python library that uses Pydantic models to define structured, type-safe interfaces for LLM interactions and agentic workflows. Ensures that LLM outputs conform to a specified schema — critical in production pipelines where downstream code expects structured data (e.g., a JSON object with specific fields) rather than free-form text.

10. Financial Time Series Concepts

Term	Explanation
yfinance	An open-source Python library that downloads historical market data from Yahoo Finance. Returns pandas DataFrames with OHLCV (Open, High, Low, Close, Volume) data. Widely used by practitioners, academics, and in production systems for its simplicity and breadth of coverage across global markets.
Daily Returns	The percentage change in price from one trading day to the next: return[t] = (price[t] - price[t-1]) / price[t-1]. Daily returns are more stationary than price levels, making them better suited for statistical modeling. Financial risk models (VaR, CVaR) are built on return distributions.
Volatility	The standard deviation of returns over a period, often annualized by multiplying by sqrt(252) (trading days per year). High volatility indicates greater uncertainty and risk. Volatility itself clusters and changes over time (ARCH effects) — a major challenge for forecasting models that assume constant variance.
Moving Average Crossover	A technical trading signal generated when a short-period moving average (e.g., 50-day) crosses above or below a long-period moving average (e.g., 200-day). A 'Golden Cross' (short crosses above long) is a bullish signal; a 'Death Cross' is bearish. These are frequently used as features in ML trading models.
Efficient Market Hypothesis (EMH)	The theory that asset prices fully reflect all available information, making it impossible to consistently achieve returns above the market average through forecasting. Weak-form EMH implies that past prices cannot predict future prices. If true, the high R2 scores from lag-feature models are spurious — explained by the lag-1 feature tracking today's price, not genuine prediction.

Term	Explanation
Look-ahead Bias	A form of data leakage in which information from the future is inadvertently used when training or evaluating a model. In time series, the most common cause is random train/test splitting, which allows future values to appear in the training set. Walk-forward validation eliminates look-ahead bias by strictly maintaining chronological order.

Quick Reference: Model Selection Guide

Term	Explanation
Use ARIMA when...	Data is univariate, you need interpretable coefficients for stakeholders, you want a statistically rigorous baseline, or you are modeling data with clear AR/MA structure visible in ACF/PACF plots.
Use Prophet when...	You need automatic seasonality detection, business stakeholders need confidence intervals, you have known holidays or events to model, or you want a production-ready forecast with minimal tuning.
Use sklearn lag models when...	You want to incorporate external features alongside lags, you need non-linear relationships, you want feature importances for model explanation, or you are already in an sklearn-based pipeline.
Use an LLM agent when...	You need natural language interpretation of results, the task involves decision-making across multiple steps, you want to automate exploratory analysis, or you need to synthesize forecasting outputs into a narrative report.
Walk-forward always when...	Evaluating ANY time series model for production use. Never use random splits. Always maintain strict chronological ordering of train and test data to avoid look-ahead bias and get realistic performance estimates.