Introduction to Forecasting

BSAD 8310: Business Forecasting — Lecture 1

Department of Economics

University of Nebraska at Omaha

Spring 2026

Why Forecasting Matters

Every business decision made today depends on expectations about tomorrow.

Operations & Supply Chain

- How much inventory should Walmart order for Black Friday?
- How many nurses does a hospital need next Tuesday night?
- When will this machine component fail?

Finance & Risk

- Will credit card defaults rise next quarter?
- What will the Fed Funds rate be in 6 months?
- How volatile will equity returns be tomorrow?

Marketing & Strategy

- How many units will this new product sell in year one?
- What is the customer lifetime value of this cohort?

Macro & Public Policy

- Will GDP grow by 2% or 3% next year?
- What will unemployment be in Q4?

A decision is only as good as the forecast it rests on.

- **UK NHS (2020):** Initial COVID hospitalization forecasts off by 4$\times$ — led to procurement of excess ventilators and delayed other treatments.
- **Toys R Us (2017):** Systematically over-ordered inventory based on outdated demand forecasts — contributed to bankruptcy.
- **Target Canada (2015):** Inventory system forecasting errors caused shelves to be overstocked on some items, empty on others — a \$2B writedown.

Overfitting is not the only failure mode. Ignoring uncertainty (using a point forecast as if it were certain) is equally dangerous.

Fundamental challenges:

- **Uncertainty:** The future is inherently random. The best we can do is characterize the distribution of outcomes.
- **Nonlinearity:** Many economic relationships change over time or at a threshold.
- **Regime shifts:** Structural breaks (recessions, pandemics, policy changes) alter the data-generating process.
- **High dimensionality:** More potential predictors than observations (especially in ML half of course).

Statistical challenges:

- Serial correlation: y_t and y_{t-1} are not independent.
- Non-stationarity: mean and variance may change over time.
- Sparse data: rare events (defaults, crises) have few observations.
- Evaluation: you can only evaluate out-of-sample.

This course: learn *when* and *why* each method works — not just how to run it.

The Forecasting Framework

A precise language for talking about prediction under uncertainty.

A **time series** is an ordered sequence of observations $\{y_1, y_2, \ldots, y_T\}$ where t indexes time.

Notation:

- y_t — observed value at time t
- h — **forecast horizon** (periods ahead)
- $\mathcal{F}_t$ — **information set** at time t (all data available up to and including time t; may include past values of predictors and economic indicators, not only the history of y_t)
- $\hat{y}_{t+h|t}$ — forecast of y_{t+h} made at time t

Examples:

- $h = 1$: next month's sales
- $h = 12$: sales 12 months from now
- $h = 1$ vs. $h = 12$: harder problem as h increases

The subscript notation $\hat{y}_{t+h|t}$ is read "y-hat at $t + h$ given t."

We want a single number $\hat{y}_{t+h|t}$ to represent our best guess of y_{t+h}.

Suppose you must predict next month's retail sales. You can observe all history $\{y_1, \ldots, y_t\}$.

What single number would you report to minimize your average squared miss?

Hint: think about what the mean of a distribution minimizes.

Under **squared error loss**, the optimal h-step-ahead forecast is:

$$\hat{y}_{t+h|t} = \mathbb{E}[y_{t+h} \mid \mathcal{F}_t]$$

the *conditional expectation* of y_{t+h} given all information at time t.

Why the conditional expectation?

- The MSE-minimizing predictor of any random variable Z is $\mathbb{E}[Z]$ — the mean.
- Conditioned on $\mathcal{F}_t$, that becomes $\mathbb{E}[Z \mid \mathcal{F}_t]$.
- Proof sketch: $\mathbb{E}[(Z - c)^2]$ is minimized at $c = \mathbb{E}[Z]$; condition on $\mathcal{F}_t$.

Different loss functions yield different optimal forecasts. MAE loss $\Rightarrow$ conditional median. We will use MSE throughout unless stated otherwise.

When is MAE preferable? Hint: asymmetric costs (e.g., stockout vs. over-stocking).

The **forecast error** for a one-step-ahead forecast is:

$$e_t = y_t - \hat{y}_{t|t-1}$$

For h-step-ahead: $e_{t+h} = y_{t+h} - \hat{y}_{t+h|t}$

Properties of errors from the optimal forecast:

- $\mathbb{E}[e_t] = 0$ — unbiased (no systematic over- or under-prediction)
- $\mathrm{Cov}(e_t, e_{t-k}) = 0$ for $k \geq 1$ — errors are uncorrelated (if forecast is truly optimal)
- Errors that are autocorrelated reveal *unexploited information*

Reserve ε_t for the true innovation (white noise) of a model. Use e_t for the realized forecast error. These are different: e_t includes parameter estimation error; ε_t does not.

Point forecast

$$\hat{y}_{t+1|t} = 42.7$$

- Single number
- Most common in practice
- Hides uncertainty

Interval forecast

$$[38.2,\ 47.2]$$

- Lower + upper bound
- 95% coverage means: 95% of such intervals contain y_{t+1}
- Calibration matters

Density forecast

$$p(y_{t+1} \mid \mathcal{F}_t)$$

- Full distribution
- Most informative
- Harder to produce and evaluate

This course focuses on **point** and **interval** forecasts. Density forecasting is introduced in Lecture 6.

Time Series Patterns

Before fitting any model, understand the structure of the data.

Any time series can be decomposed into (additive form):

$$y_t = T_t + S_t + C_t + I_t$$

Multiplicative form: $y_t = T_t \times S_t \times C_t \times I_t$ (use when seasonal amplitude grows with level).

- T_t: **Trend** — long-run movement (linear, quadratic, or stochastic)
- S_t: **Seasonality** — regular, calendar-driven pattern
- C_t: **Cycle** — medium-run fluctuations (business cycles)
- I_t: **Irregular** — random shocks and one-offs

- **Retail sales:** strong trend + strong seasonality
- **Quarterly GDP:** trend + cycle (weak seasonality)
- **Daily stock returns:** nearly all irregular
- **Energy demand:** trend + seasonality + cycle

Seasonal period m: the number of observations per year (or per recurring cycle).

Data frequency	Seasonal period m
Annual	1 (no seasonality)
Quarterly	4
Monthly	12
Weekly	52
Daily	7 or 365

Ignoring seasonality is a common source of large forecast errors. A non-seasonal model applied to monthly retail data will systematically under-forecast in December and over-forecast in January.

Lab 01 will visualize seasonality in US retail sales data. We will decompose the series using STL (Seasonal-Trend decomposition via LOESS).

A time series $\{y_t\}$ is **weakly stationary** if:

1. $\mathbb{E}[y_t] = \mu$ (constant mean)
2. $\mathrm{Var}(y_t) = \sigma^2 < \infty$ (constant variance)
3. $\mathrm{Cov}(y_t, y_{t-k}) = \gamma_k$ depends only on lag k, not on t

Why it matters:

- Most classical forecasting methods assume stationarity
- Non-stationary series produce *spurious* regressions

Rule of thumb: Visible trend or expanding variance $\Rightarrow$ likely non-stationary.
Full treatment: **Lecture 4** (ARIMA Models).

Is US retail sales stationary? What about daily S&P 500 returns vs. price levels?

Benchmark Models

Always beat a simple baseline before claiming success.

Before presenting any forecast model, establish what a **naive benchmark** achieves. A model that does not beat the benchmark has zero value.

Why benchmarks matter:

- Many series are hard to forecast — even “simple” models may be hard to beat.
- The M4 Competition (Makridakis et al. 2020): out of 61 methods, many sophisticated ML models failed to beat exponential smoothing.
- A benchmark defines the floor. Your goal is to clear it by a meaningful margin.

Naïve, Seasonal Naïve, Historical Mean, Random Walk with Drift. We will always compute these before evaluating any model.

$$\hat{y}_{t+h|t} = y_t \quad \text{for all } h \geq 1$$

Use the *most recent observation* as the forecast for all future periods.

Properties:

- Equivalent to a **random walk** model: $y_t = y_{t-1} + \varepsilon_t$
- Optimal under the random walk hypothesis
- Standard benchmark for financial prices
- Forecast does not change with h (flat forecast function)

When to use:

- ✓ Asset prices, exchange rates
- ✓ When you have very little history
- × Series with strong trend or seasonality

Python: `y_hat = y_train.iloc[-1]`

$$\hat{y}_{t+h|t} = y_{t+h-12}$$

Use the *value from the same month one year ago*.

General form for arbitrary h and period m: $\hat{y}_{t+h|t} = y_{t+h-m\cdot\lceil h/m \rceil}$

Properties:

- Captures seasonality without modeling it explicitly
- Standard benchmark for monthly business data
- Automatically handles any seasonal period m

When to use:

- ✓ Retail sales, electricity demand
- ✓ Any series with clear seasonal pattern
- × Non-seasonal series (use naïve instead)

Python: shift by seasonal period using `pd.Series.shift(m)`

$$\hat{y}_{t+h|t} = \bar{y}_t = \frac{1}{t}\sum_{s=1}^{t} y_s \quad \text{for all } h \geq 1$$

Forecast with the *average of all past observations*.

Properties:

- Optimal under the assumption that y_t is i.i.d.
- Produces a constant, flat forecast
- Variance of forecast error decreases as t grows (more data $\Rightarrow$ better estimate of μ)

When to use:

- ✓ Stationary series with no pattern
- ✓ Very long history, very stable series
- × Any series with trend or seasonality

Python: `y_hat = y_train.mean()`

$$y_t = c + y_{t-1} + \varepsilon_t \quad \Longrightarrow \quad \hat{y}_{t+h|t} = y_t + h\hat{c}$$

where $\hat{c} = (y_T - y_1)/(T - 1)$ is the estimated average period-to-period change.

Properties:

- Combines naïve with a linear trend component
- Forecast function: linear extrapolation from last observation
- Special case of Holt's linear exponential smoothing (Lecture 3) with $\alpha = 1$, $\beta = 1$

Which benchmark to use?

No trend, no season $\rightarrow$ Mean
Trend, no season $\rightarrow$ RW + Drift
Season, no trend $\rightarrow$ Seasonal Naïve
Trend + Season $\rightarrow$ Seasonal Naïve
Asset prices $\rightarrow$ Naïve

Forecast Evaluation

Forecasting without evaluation is not science — it is storytelling.

Given T out-of-sample observations $\{y_{T+1}, \ldots, y_{T+H}\}$ and forecasts $\{\hat{y}_{T+1|T}, \ldots, \hat{y}_{T+H|T+H-1}\}$:

Scale-dependent metrics:

$$\mathrm{MSE} = \frac{1}{H}\sum_{h=1}^{H} e_{T+h}^2$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

$$\mathrm{MAE} = \frac{1}{H}\sum_{h=1}^{H} |e_{T+h}|$$

RMSE and MAE are in the same units as y_t.

Use to compare models on the *same* series.

Scale-free metric:

$$\text{MAPE} = \frac{1}{H}\sum_{h=1}^{H} \left|\frac{e_{T+h}}{y_{T+h}}\right| \times 100\%$$

MAPE: cross-series comparison; undefined at $y_{T+h} = 0$; asymmetric.

RMSE vs. MAE: RMSE penalizes large errors more heavily. If large errors are very costly, prefer RMSE.

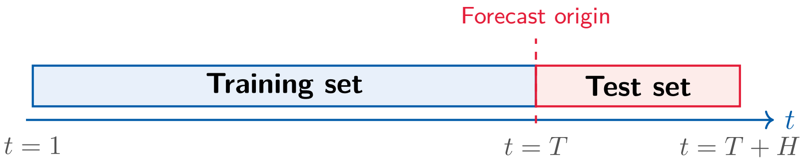
Metric choice: RMSE/MAE for same-series; MAPE across series; MASE vs. seasonal naïve. Prefer RMSE for costly outliers; MAE for robust comparison.

Never evaluate a forecast model on the data used to fit it. In-sample fit measures how well you memorized the past, not how well you can predict the future.

Correct time series train/test split:

- Use observations $1, \ldots, T$ to estimate the model
- Evaluate on $T + 1, \ldots, T + H$ (held-out future observations)
- *Never* randomly shuffle observations — time order must be preserved

Common mistake: Using `sklearn`'s `KFold` on time series data. This randomly assigns observations to folds, allowing future data to "train" the model. Always use `TimeSeriesSplit`. Full treatment in **Lecture 7**.



- Model estimated on $\{y_1, \ldots, y_T\}$
- Forecasts $\{\hat{y}_{T+1|T}, \ldots, \hat{y}_{T+H|T}\}$ evaluated against realized values
- Typical test set size: **10–20%** of the series, or a fixed future window

Walk-forward (rolling-origin) evaluation: Lecture 6.

Hyndman and Koehler (2006) propose the **Mean Absolute Scaled Error**:

$$\text{MASE} = \frac{\text{MAE}}{\frac{1}{T-m}\sum_{t=m+1}^{T}|y_t - y_{t-m}|}$$

- Denominator: in-sample MAE of the seasonal naïve forecast
- MASE < 1: model beats seasonal naïve
- MASE > 1: model is *worse* than the naïve benchmark
- Works when $y_t = 0$ (unlike MAPE)
- Symmetric — positive and negative errors treated equally

MASE is the standard metric in the M-competitions (Makridakis et al. 2020). We will use it in Lecture 6 for model comparison.

Course Roadmap

What we will cover, and how it fits together.

Part I: Classical Econometric Forecasting

1. **Introduction** $\leftarrow$ today
2. Regression-Based Forecasting
3. Exponential Smoothing (ETS)
4. ARIMA Models
5. Multivariate Methods (VAR, ARIMAX)
6. Forecast Evaluation & Combination

Focus: theory + properties + principled model selection.

Part II: Predictive Analytics & Machine Learning

7. ML Introduction & Cross-Validation
8. Regularization (LASSO, Ridge, Elastic Net)
9. Tree-Based Methods (RF, XGBoost)
10. Neural Networks & LSTM
11. Feature Engineering
12. Capstone & Applications

Focus: scalability, data-driven model selection, real-world pipelines.

Python throughout:

- `statsmodels` — ARIMA, ETS, VAR, statistical tests
- `scikit-learn` — ML models, cross-validation, pipelines
- `pandas` — data manipulation
- `matplotlib` — visualization
- `numpy` — numerical computing

Lab format: Jupyter notebooks. Each lab implements the methods from the preceding lecture.

What you will be able to do:

- ✓ Select and justify a forecasting method for a business problem
- ✓ Implement ARIMA, ETS, LASSO, and RF models in Python
- ✓ Evaluate and compare models rigorously
- ✓ Communicate forecast uncertainty to stakeholders
- ✓ Avoid the most common forecasting mistakes

Forecasting underpins every forward-looking business decision. Bad forecasts have measurable, costly consequences.

The optimal point forecast (under squared loss) is the **conditional expectation**: $\hat{y}_{t+h|t} = \mathbb{E}[y_{t+h} \mid \mathcal{F}_t]$.

Always establish **benchmark performance** first. A model that does not beat naïve is not useful.

Evaluate out-of-sample only. Never shuffle time series; always use `TimeSeriesSplit`.

Metric choice matters: RMSE penalizes large errors; MAPE allows cross-series comparison; MASE benchmarks against naïve.

Next: Regression-Based Forecasting — when and how do predictor variables improve on these benchmarks?

Hyndman, Rob J. and Anne B. Koehler (2006). "Another Look at Measures of Forecast Accuracy". In: *International Journal of Forecasting* 22.4, pp. 679–688.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos (2020). "The M4 Competition: 100,000 Time Series and 61 Forecasting Methods". In: *International Journal of Forecasting* 36.1, pp. 54–74.