Week 7: From Linear Models to Machine Learning

The Conceptual Bridge

Learning Objectives

- Distinguish prediction from causal inference, and explain why they require different tools
- Trace the logical path from OLS to regularised regression to ensemble methods
- Explain the bias-variance tradeoff and why it is the central challenge in financial forecasting
- Interpret the "virtue of complexity" result and what it implies for factor model design
- Identify which CW2 scaffold matches your interests and understand the analytical task it sets

Agenda

Opening : Where we have been, and where we are going

Part I : Two different problems, prediction versus causal inference

Part II : OLS as a prediction model, and its limits

Part III : The bias-variance tradeoff, the central tension in forecasting

Part IV : Regularisation, shrinking towards better predictions

Part V : The virtue of complexity, when more features help

Part VI : Ensemble methods, trees, forests, and boosting

Part VII : CW2 scaffold preview, what the assessment asks of you

Opening: Where We Have Been

The Three Prediction Problems (Revisited)

You met this framework in Week 1. Seven weeks later, it should look different.

Problem	Target variable	Typical R^2	Where we covered it
Mean	Future returns	~1–2%	Week 3: ARIMA rarely beats naive
Variance	Future volatility	~15–40%	Week 4: GARCH succeeds
Cross-section	Which assets outperform	~5–15%	Weeks 8–10: factors and ML

The third prediction problem is where most of the action in financial machine learning lives.

The Methods Arc

We have now covered the foundational toolkit. This week is the pivot.

- **Weeks 1–2**: Statistical foundations, regression as comparison, bias, inference
- **Weeks 3–4**: Time series methods, ARIMA, GARCH, stationarity
- **Weeks 5–6**: Applied FinTech, portfolio optimisation, alternative finance, credit scoring

You are here: the conceptual bridge

- **Weeks 8–10**: Advanced methods, factor models, random forests, sequence learning
- **Weeks 11–13**: Validation, backtesting, synthesis

The conceptual gap between the two blocks is significant. Today we bridge it.

Part I: Two Different Problems

The Question That Changes Everything

Before choosing a method, ask: **what is the question?**

Two superficially similar tasks:

- "What is the effect of firm size on expected returns?"
- "Can I predict which firms will have the highest returns next month?"

Both involve regressing returns on firm characteristics. The methods look similar. But the objectives are fundamentally different.

Prediction vs Causal Inference (Mullainathan and Spiess 2017)

Mullainathan and Spiess (2017) draw the sharpest version of this distinction.

Causal inference asks: What is the effect of X on Y?

- Coefficients must be unbiased and interpretable
- Omitted variable bias is a fatal flaw
- Standard errors and confidence intervals matter for conclusions
- Adding irrelevant controls reduces efficiency but does not break the analysis

Prediction asks: Given X, what is the best forecast of Y?

- The model need not be interpretable
- Bias in coefficients is acceptable if it reduces variance
- Evaluation is entirely out-of-sample
- Adding irrelevant features can help if they reduce forecast error

Two Tasks, Two Evaluation Criteria

This difference runs all the way through to how we evaluate success.

	Causal inference	Prediction
Goal	Estimate $\hat{}$ accurately	Minimise forecast error
Key metric	Standard error of $\hat{}$	Out-of-sample MSPE
Omitted variables	Bias, major problem	Less critical
Overfitting	Not the primary concern	The central problem
More features	Risk of multicollinearity	Can help with regularisation
Interpretability	Essential	Optional

Financial Examples: Which Problem Is It?

Can you classify each task?

- Measuring whether algorithmic lending discriminates against protected groups
- Predicting which loan applications will default next quarter
- Estimating how much an extra year of education raises lifetime earnings
- Building a model to rank stocks by next month's expected return
- Quantifying the effect of quantitative easing on mortgage rates
- Scoring credit applicants for automated lending decisions

Part II: OLS as a Prediction Model: Its Limits

OLS: What It Minimises

Ordinary Least Squares minimises the **in-sample** sum of squared residuals:

$$\hat{\beta}_{OLS} = \arg\min_{\beta} \sum_{t=1}^{T} (y_t - \mathbf{x}_t'\beta)^2$$

This produces the Best Linear Unbiased Estimator (BLUE) under the Gauss-Markov assumptions.

But notice: the objective is in-sample. We minimise residuals on the **data we already have**, not on data we have not yet seen.

The Degrees-of-Freedom Trap

R^2, our usual measure of fit, **always increases** as we add predictors, even if they are pure noise.

Illustration with financial data:

Predictors	Description	In-sample R^2
1	Market beta only	12%
5	+ size, value, momentum, profitability	28%
20	+ 15 random noise variables	41%
50	+ 30 more noise variables	63%

The last model is almost certainly **worse** at predicting next month's returns.

Mean Squared Prediction Error

The right criterion for prediction is **Mean Squared Prediction Error** (MSPE):

$$MSPE = E\left[(y_{T+h} - \hat{y}_{T+h})^2\right]$$

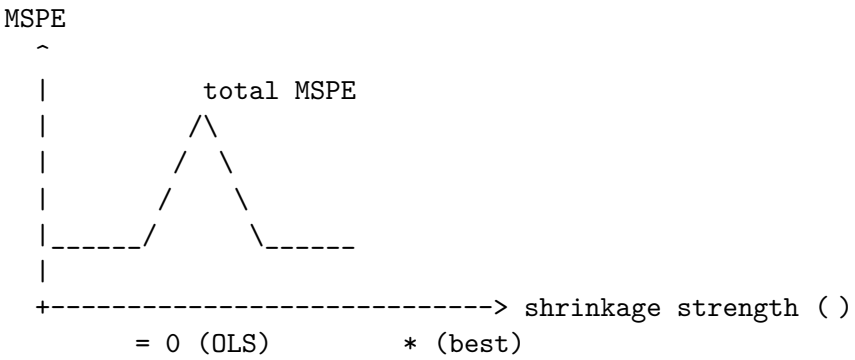
This is the expected squared error on **new, unseen data**, not data used to estimate the model.

Stock and Watson (2002) (Chapter 14): OLS minimises in-sample MSPE but not out-of-sample MSPE. Adding predictors helps in-sample but often hurts out-of-sample.

Shrinkage: A deliberately biased estimator can have lower MSPE than OLS if the reduction in variance more than compensates for the increase in bias.

Visual intuition: shrinkage chooses ( )

Think of ( ) as the strength of shrinkage (regularisation).



- **Variance falls** as we shrink coefficients.
- **Bias rises** because we pull estimates towards a target.
- **MSPE can improve** if the variance reduction is large enough.

Why OLS Breaks Down in Finance

Financial prediction has properties that stress OLS particularly hard.

Many predictors, few observations. Let (P) be the number of predictors (features) and (T) the number of observations (months). In many cross-sectional return prediction settings:

$$P \in [50, 200], \qquad T \approx 240, \qquad \frac{P}{T} \gtrsim 1$$

This can put us in the overparameterised regime.

Highly correlated features. Size, value, and momentum factors are correlated. OLS coefficient estimates become unstable under multicollinearity.

Non-stationarity. Return distributions shift over time. A model estimated in one regime may not generalise to another.

Noise dominates signal. Typical monthly return R^2 values are 1–5%. Most of the variation in returns is unexplained. OLS will happily fit that noise.

When (P) exceeds (T), what does that actually mean?

In regression notation, the feature matrix (X) has shape:

$$X \in \mathbb{R}^{T \times P}$$

predictors (P)

T
obs X

When (P T), the matrix (X^ X) is not invertible, so the OLS formula breaks:

$$\hat{\beta}_{OLS} = (X^\top X)^{-1} X^\top y$$

Intuitively, you have more knobs than data points: many different coefficient vectors can fit the training data equally well, which makes estimates unstable and out-of-sample

Part III: The Bias-Variance Tradeoff

Decomposing Prediction Error

Any prediction error can be decomposed into three components:

Recall from Week 1: see the Foundations chapter section on the bias-variance tradeoff.

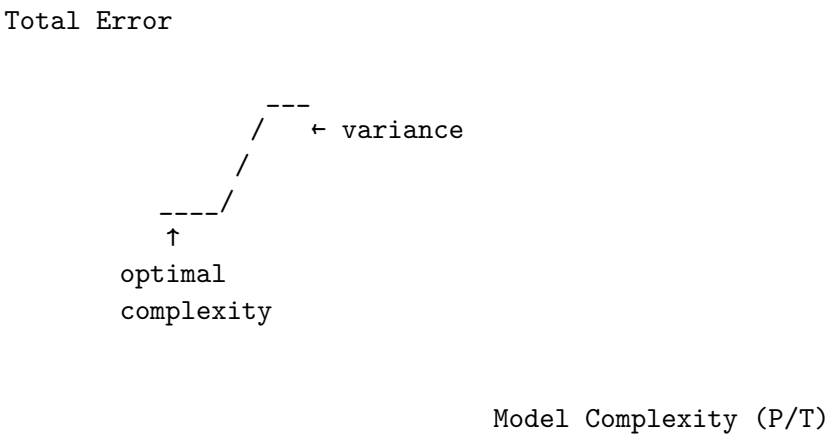
$$E[(y - \hat{f}(x))^2] = \underbrace{\text{Bias}(\hat{f})^2}_{\text{systematic error}} + \underbrace{\text{Var}(\hat{f})}_{\text{estimation noise}} + \underbrace{\sigma_\varepsilon^2}_{\text{irreducible}}$$

- **Bias**: How far is our model from the true relationship on average?
- **Variance**: How much does our model change across different training samples?
- **Irreducible error**: Random noise that no model can predict

We can only control bias and variance, and they move in opposite directions.

The Bias-Variance Curve

As model complexity increases, bias falls but variance rises.



The "optimal" point balances these forces. OLS without regularisation will sit too far to the right in high-dimensional settings.

Bias and Variance in Practice

High-bias model (underfitting): CAPM, a single market beta predictor.

- Systematic error: misses size, value, momentum effects
- Stable: always gives the same answer
- Generalises well but misses most of the signal

High-variance model (overfitting): OLS with 100 firm characteristics.

- Low in-sample error: fits training data excellently
- Unstable: coefficients change dramatically year to year
- Poor out-of-sample performance: fitted noise, not signal

The goal: A model that captures the true structure (low bias) without fitting the noise (low variance). Regularisation provides the mechanism.

The Interpolation Boundary

When the number of predictors P equals the number of observations T, OLS does something catastrophic.

With $P = T$: OLS fits the training data **perfectly** ($R^2 = 1$ by construction). Every data point is exactly interpolated.

The out-of-sample forecast is essentially **pure noise**, the model has memorised the training set rather than learning the underlying pattern.

This $P = T$ point is called the **interpolation boundary**. It is where classical statistical intuition says OLS completely fails.

Part IV: Regularisation: Shrinking Towards Better Predictions

Accepting Bias to Reduce Variance

Regularisation deliberately introduces bias into our coefficient estimates to reduce variance. The result is lower total prediction error.

The mechanism: add a **penalty** to the OLS objective function that discourages large coefficients.

$$\hat{\beta}_{reg} = \arg\min_{\beta} \left[\sum_{t=1}^{T} (y_t - \mathbf{x}_t'\beta)^2 + \lambda \cdot \text{penalty}(\beta) \right]$$

The parameter controls the bias-variance tradeoff:

- $= 0$: pure OLS, minimum bias, maximum variance
- $\to \infty$: all coefficients shrink to zero, maximum bias, minimum variance
- Optimal : somewhere in between, found by cross-validation

Ridge Regression

Penalty: sum of squared coefficients (L2 norm)

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \left[\sum_{t=1}^{T} (y_t - \mathbf{x}_t' \beta)^2 + \lambda \sum_{j=1}^{P} \beta_j^2 \right]$$

Properties:

- All coefficients shrink towards zero but remain non-zero
- Works well when many predictors have small true effects
- Has a closed-form solution: $\hat{\beta}_{Ridge} = (X'X + \lambda I)^{-1} X'y$
- Solves the P $>$ T problem: $(X'X + \lambda I)$ is always invertible

Financial interpretation: assume all factors contribute a little, shrink the big effects down.

LASSO Regression

Penalty: sum of absolute coefficients (L1 norm)

$$\hat{\beta}_{LASSO} = \arg\min_{\beta} \left[\sum_{t=1}^{T} (y_t - \mathbf{x}_t' \beta)^2 + \lambda \sum_{j=1}^{P} |\beta_j| \right]$$

Properties:

- Drives some coefficients to **exactly zero**, automatic variable selection
- Produces sparse solutions: only a subset of predictors remain
- Well-suited when most predictors are truly irrelevant
- No closed-form solution: requires iterative algorithms

Financial interpretation: from 200 factors, select the 10–20 that matter most and set the rest to zero.

Choosing : Cross-validation

How do we find the right ? We **hold out some data** and evaluate out-of-sample performance.

K-fold cross-validation:

Terminology: a **fold** is one slice of the training sample. The held-out fold is a temporary **validation set** used to choose ( ).

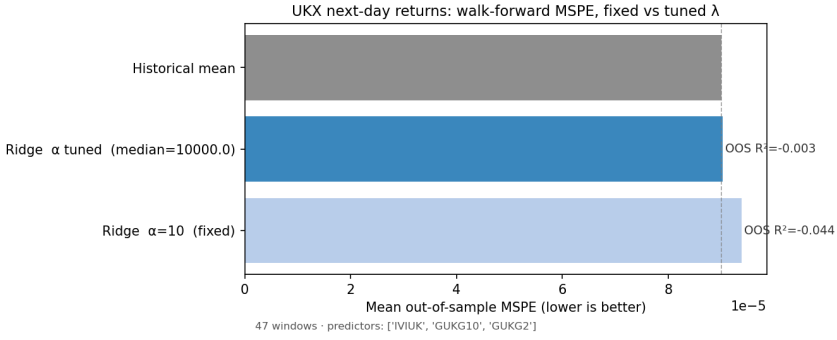
1. Split the training data into (K) folds (often (K = 5) or (10))
2. For each candidate ( ): train on (K-1) folds, evaluate on the held-out fold
3. Repeat so each fold is held out once, then average the validation error
4. Choose ( ) with the lowest average validation error

In finance, cross-validation requires care. Standard K-fold randomly shuffles observations. With time series data this induces look-ahead bias: the held-out fold can contain observations *before* some training observations.

Solution: time-series cross-validation (walk-forward validation). Always use past data to train, future data to evaluate.

Same discipline, real data (UK)

Chapter 06 compares **two** Ridge models on daily Bloomberg data (UKX next-day returns, IVIUK + gilt yield lags): a fixed penalty ($\lambda = 10$, as a foil) and a **nested-CV tuned** penalty where an inner `TimeSeriesSplit` loop selects the best λ from a 50-point log-grid, using only past data.



Takeaway: what “honest forecasting” requires

- **Time order**: train on the past, score on the future.
- **Leakage control**: scaler and feature stats fit inside each training window only.
- **Nested CV for tuning**: select λ via an inner walk-forward loop, not on the full dataset.
- **Hard benchmark**: beat the historical mean before claiming a useful return signal.
- **Where to look**: Chapter 06, *Empirical counterpart: UK market data*.

Part V: The Virtue of Complexity

A Surprising Finding

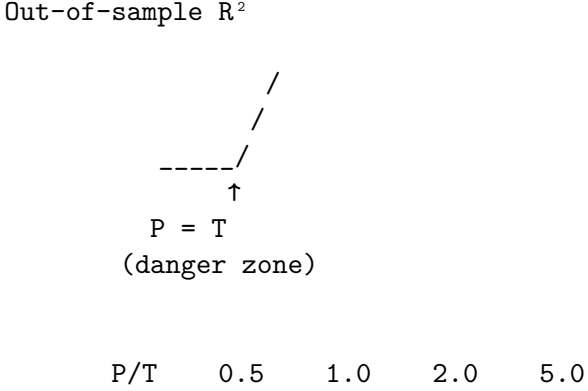
Classical statistical wisdom: parsimonious models generalise better. Keep it simple.

Kelly, Malamud, and Zhou (2024) challenge this directly (Kelly, Malamud, and Zhou 2024).

Using Random Matrix Theory, they characterise ridge (and "ridgeless") return prediction when the number of predictors (P) is large relative to the sample size (T). Performance deteriorates near the interpolation boundary (P T), but can recover for ($P/T > 1$) under shrinkage.

This is called the **"double descent"** phenomenon.

The Double Descent Picture



The **first descent** is the classical overfitting: as P approaches T, variance explodes.

The **second ascent** is the new finding: once P » T with ridge regularisation, the implicit shrinkage of the estimator increases and out-of-sample R^2 recovers. It can exceed low-complexity benchmarks.

Why Does Complexity Help?

The key insight from Kelly, Malamud, and Zhou (2024): in the overparameterised regime, **ridge regularisation acts as implicit shrinkage** that becomes stronger as P/T increases.

As you add more predictors:

- Each individual predictor contributes less to variance (there are more to share the load)
- The ridge penalty effectively distributes the shrinkage more evenly
- The model captures more of the non-linear approximation to the true DGP
- Trading strategy Sharpe ratios rise monotonically with complexity (up to a point)

The crucial condition: regularisation must be present. Without ridge, OLS in the overparameterised regime is useless. With ridge, high complexity models can outperform.

Implications for Factor Investing

The Kelly et al. result motivates the approach taken in CW2 Scaffold B: can a high-dimensional, non-linear model beat a parsimonious linear benchmark?

Linear approach: Select 3–5 well-known factors (Fama-French), run OLS with HAC standard errors. Sparse, interpretable, disciplined.

Machine learning approach (Scaffold B): Use 20–100 firm characteristics simultaneously, apply tree-based methods (which implicitly regularise), evaluate out-of-sample. Potentially better predictions, but harder to interpret.

Neither is universally right. The question is empirical: which approach performs better for UK equities in the JKP dataset?

Part VI: Ensemble Methods: Trees, Forests, and Boosting

Beyond Linear Models

Ridge and LASSO are still linear: the prediction is a weighted sum of features. Sometimes the true relationship is non-linear.

Decision trees provide a flexible non-linear alternative. From a statistical viewpoint, they are trying to reduce the same objective as before: out-of-sample prediction error (MSPE).

A tree builds a piecewise-constant approximation to (E[r_{t+1} X_t]) by repeatedly splitting the data to reduce squared error.

How to read this tree:

- **Start at the top** and follow the yes/no rule for the firm you care about
- **Each split** is chosen to reduce in-sample squared error (then validated out of sample)
- **The leaf value** is the model's prediction for that region (often the mean return in that leaf)

Example decision rules:

Variance Reduction Through Averaging

A **random forest** addresses the high variance of single trees through two innovations:

Bootstrap aggregation (bagging):

- Train many trees on different random samples of the training data
- Average their predictions
- Averaging reduces variance without increasing bias

Random feature subsets:

- At each split, only consider a random subset of predictors
- This decorrelates the trees, they make different errors
- Decorrelated errors cancel when averaged

Key property: the averaged prediction has lower variance than any single tree, while preserving the non-linear flexibility.

Bias Reduction Through Sequential Correction

Where **random forests** reduce **variance**, **gradient boosting** reduces **bias**.

The algorithm:

1. Start with a simple model (e.g., predict the mean return for everyone)
2. Compute the residuals, what the current model gets wrong
3. Fit a new tree to predict the residuals
4. Add this tree to the ensemble (with a small learning rate)
5. Repeat until performance stops improving on held-out data

Each tree corrects the systematic mistakes of the previous ensemble. The model progressively reduces its bias.

Random Forest vs Gradient Boosting

	Random Forest	Gradient Boosting
Core idea	Average many trees	Correct residuals sequentially
Bias	Higher (each tree weak)	Lower (explicitly corrects)
Variance	Lower (averaging)	Higher (can overfit)
Training	Parallelisable	Sequential
Tuning	Fewer hyperparameters	More careful tuning required
Speed	Faster	Slower (but XGBoost is efficient)
Finance use	Factor importance, robust baseline	Higher performance ceilings

Both methods appear in the recent financial ML literature (Gu, Kelly, and Xiu 2020).

Interpreting Ensemble Models

Tree-based models are non-linear and cannot be interpreted via coefficients alone. **SHAP values** (Shapley Additive Explanations) provide a principled solution.

For each prediction, SHAP decomposes the prediction into contributions from each feature:

$$\hat{y}_i = \underbrace{\phi_0}_{\text{baseline}} + \underbrace{\phi_1^{(i)}}_{\text{momentum}} + \underbrace{\phi_2^{(i)}}_{\text{value}} + \underbrace{\phi_3^{(i)}}_{\text{size}} + \ldots$$

Properties:

- Each $\phi_j^{(i)}$ is the marginal contribution of feature j to prediction i
- They sum to the total prediction (relative to the baseline ϕ_0)
- Globally: average absolute contribution, $\frac{1}{N}\sum_i |\phi_j^{(i)}|$, gives a feature importance ranking

SHAP is post-hoc, not causal

SHAP helps you describe how a fitted model behaves. It does not identify what truly drives returns.

- **Explains $\hat{f}$, not the DGP**: you are decomposing the model's prediction, not estimating a causal effect
- **Correlated features**: attributions can be unstable when predictors move together (common in finance)
- **Depends on the background dataset**: the baseline ϕ_0 and contributions change with the reference distribution
- **Treat as model diagnostics**: use alongside out-of-sample tests and robustness checks, not as proof of a "factor"

Part VII: CW2 Scaffold Preview

What CW2 Is and Is Not

It is not: a test of your Python coding ability.

It is: a test of your capacity to complete provided code, interpret outputs, and reflect critically on what the results mean.

The scaffold notebooks provide:

- All data loading (Elliptic Bitcoin, JKP UK factors, or Bloomberg volatility data)
- Core model implementation (60–70% complete)
- TODO sections that require completion
- Markdown cells prompting reflection

You provide:

- Completed TODO sections (guided by the scaffold)
- A 2,500-word reflective report demonstrating methodological understanding
- A data quality lens: what biases and limitations affect these results?

Three Scaffold Choices: Overview

Scaffolds are released **next week (Week 8)**. Today is a preview.

Scaffold	Topic	Methods	Data
A	Blockchain Fraud Detection	Logistic regression, random forest, walk-forward CV, cost-sensitive thresholds	Elliptic Bitcoin (46K labelled txns)
B	Tree-Based Factor Investing	Random forest, gradient boosting, SHAP	JKP UK monthly factors
C	Volatility Forecasting	GARCH, GJR-GARCH, Mincer-Zarnowitz evaluation	Bloomberg equity indices

All three use real data from professional sources. All three require genuine analytical

Scaffold A: Blockchain Fraud Detection

The question: Can we reliably detect illicit Bitcoin transactions, and how does model performance degrade as fraud patterns evolve?

Why this matters: The Elliptic dataset contains 46,564 labelled Bitcoin transactions across 49 time steps. Illicit transactions (money laundering, scams) make up roughly 10% of labels, but the rate fluctuates dramatically over time.

The analytical challenge: Walk-forward temporal validation versus shuffled CV. The gap between them reveals how much published fraud detection benchmarks overstate real-world performance. Cost-sensitive threshold selection, because the default 0.5 catches almost nothing when fraud is rare.

The data quality thread: Only labelled transactions are included (selection bias). Only detected illicit activity appears (survivorship bias). Features are anonymised, limiting interpretability. How do these constraints affect your conclusions?

Scaffold B: Tree-Based Factor Investing

The question: Does a non-linear tree-based model predict cross-sectional returns better than the linear Fama-French model?

Why this matters: If OLS with 5 factors is a biased but stable model, and gradient boosting with 50 factors is a lower-bias model, which wins in the JKP data?

The analytical challenge: Evaluate out-of-sample prediction performance. Compute SHAP values. Interpret which factors matter and whether their importance is stable over time.

The data quality thread: How does look-ahead bias arise in accounting-based factors (book-to-market, profitability)? What is the "factor zoo" problem and how might data snooping affect your results?

Scaffold C: Volatility Forecasting

The question: Does asymmetric GARCH (GJR-GARCH) systematically outperform symmetric GARCH(1,1) for UK and US equity volatility?

Why this matters: Asymmetric GARCH accounts for the leverage effect. Bad news increases volatility more than good news of the same magnitude. This is a real empirical regularity.

The analytical challenge: Mincer-Zarnowitz forecast evaluation. Regress realised volatility on forecasted volatility. A well-calibrated forecast should have intercept 0 and slope 1.

The data quality thread: Realised volatility is measured from returns data. How does data quality affect the benchmark you are comparing against?

Choosing Your Scaffold

There is no universally correct choice. Consider three questions.

What topic interests you most? Blockchain forensics, factor investing, and volatility forecasting are all active research areas. The scaffold you find most intellectually engaging is usually the one you will write about most convincingly.

What methods from this module do you understand best? Scaffold A builds on Week 8 (fraud detection, rare-event classification, temporal CV). Scaffold B builds on today's content and Week 9 (factor investing, SHAP). Scaffold C extends Week 4 (volatility modelling).

What data quality issues do you find most tractable? CW2 Section B of the report assesses your ability to identify bias in your chosen dataset. Review your CW1 skills and ask which scaffold presents the most interesting data quality challenge.

The CW1 to CW2 Thread

Your CW1 developed three skills that transfer directly to CW2.

Data risk register thinking. Every dataset has data generating process assumptions that can fail. The Elliptic data has selection bias (only labelled transactions) and survivorship bias (only detected illicit activity). The JKP factor data has survivorship bias (only surviving firms). Bloomberg volatility data has measurement error. Identifying these risks is Section B of the report.

Look-ahead bias detection. You analysed look-ahead bias in CW1. Each scaffold has its own version: temporal drift in fraud patterns (Scaffold A); accounting data timing (Scaffold B); realised volatility measurement (Scaffold C). The walk-forward validation in each scaffold is designed to prevent it; your report should explain why.

Responsible practice framing. A fraud detection model, a factor investing strategy, or a volatility forecasting model all have downstream consequences. Who uses these models? What happens when they fail? Who bears the cost of false positives? This is the professional accountability thread running through both assessments.

What Happens Next Week (Week 8)

Week 8 topic: Cryptocurrency and Fraud Detection

CW2 scaffolds released at the start of Week 8. Instructions on Blackboard.

Lab session (Week 8): Introduction to scaffolds. You will:

- Load the data and run the scaffold as provided
- See what the code does before attempting the TODOs
- Choose your scaffold in the lab session
- Begin the first TODO section with support available

Recommendation for this week: Read through the scaffold descriptions again. Think about which topic you find most interesting. Look back at your CW1 feedback.

Closing

The Conceptual Arc: From OLS to Machine Learning

Today's journey in six steps:

1. **Two problems:** Prediction and causal inference require different tools and different evaluation criteria.
2. **OLS limits:** In-sample fit optimisation is not the same as out-of-sample forecast accuracy.
3. **Bias-variance tradeoff:** Complex models overfit; simple models underfit. The goal is balance.
4. **Regularisation:** Ridge and LASSO deliberately introduce bias to reduce variance and improve MSPE.
5. **Virtue of complexity:** With proper regularisation, very high-dimensional models can outperform parsimonious benchmarks.
6. **Ensemble methods:** Random forests reduce variance through averaging; gradient boosting reduces bias through sequential correction.

Looking Ahead

Week 8: Cryptocurrency markets and fraud detection. Plus CW2 scaffold release, come ready to choose.

Week 9: Factor investing in depth. Fama-French factors, the factor zoo problem, and tree-based methods applied to the JKP dataset.

Week 10: Backtesting and validation. Walk-forward testing, combinatorial symmetric cross-validation, and the five pitfalls of statistical significance (the false discovery rate problem in factor research).

The thread: Everything from Week 7 onwards is an application of the bias-variance tradeoff under the specific constraints of financial data.

References

Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2020. "Empirical Asset Pricing via Machine Learning." *Review of Financial Studies*. https://doi.org/10.1093/rfs/hhaa009.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2019. "A Survey of Methods for Explaining Black Box Models." *ACM Computing Surveys* 51 (5): 93:1–42. https://doi.org/10.1145/3236009.

Kelly, Bryan T., Semyon Malamud, and Kangying Zhou. 2024. "The Virtue of Complexity in Return Prediction." *Journal of Finance* 79 (1): 459–503. https://doi.org/10.1111/jofi.13298.

Mullainathan, Sendhil, and Jann Spiess. 2017. "Machine Learning: An Applied Econometric Approach." *Journal of Economic Perspectives* 31 (2): 87–106. https://doi.org/10.1257/jep.31.2.87.

Stock, James H., and Mark W. Watson. 2002. "Forecasting Using Principal Components from a Large Number of Predictors." *Journal of the American Statistical Association* 97 (460): 1167–79. https://doi.org/10.1198/016214502388618960.