



**Assignment 2: Team Design
Concept**
COS60011-Technology Design
Project

Unit code: COS60011

Unit Name: Technology Design Project

Due Date: 27th Sep 2025

Word count: 2500

Name of Students and Student ID:

Name	Student ID
Bob Lin	105827369

Table of Contents

1 Introduction 1

2 Design Concept1

3 Project Management Plan 8

 3.1 Timeline 8

 3.2 Goals and Milestones 9

4 Conclusion 9

5 Appendix 9

 5.1 Reference List 9

 5.2 GenAI Declaration 10

Acknowledgement of Country

I , Bob Lin, acknowledge the Wurundjeri People of the Kulin Nation as the Traditional Custodians of the land on which Swinburne University is located. I pay my respects to their Elders past and present and recognise their continuing connection to land, waters, skies, and culture. I honour the knowledge systems and traditions that have sustained this land for generations and support efforts to create a respectful, inclusive environment for all Indigenous people

Executive Summary

This report presents a complete framework for short-term stock price prediction using the Commonwealth Bank of Australia (CBA) as a case study. The project integrates data engineering, factor construction, and machine learning modelling to generate actionable trading insights. It applies a structured workflow that includes data cleaning, transformation, feature engineering, and walk-forward validation to ensure temporal robustness and avoid look-ahead bias. Predictive models such as Logistic Regression, XGBoost, and Elastic Net were evaluated based on statistical metrics (AUC, F1) and economic indicators (profit factor, expectancy). The study demonstrates that combining financial factor design with machine learning and stress testing can produce stable, interpretable, and economically meaningful results. This framework contributes to both academic understanding and practical application in quantitative financial modelling.

1 Introduction

The purpose of this project is to design and evaluate a short-term stock prediction framework within the Australian market, focusing on the Commonwealth Bank of Australia (CBA). CBA was selected due to its liquidity, market representativeness, and availability of consistent historical data. The study explores how the integration of traditional financial factors and modern machine learning algorithms can improve predictive accuracy and interpretability. Using daily OHLCV data from 2020 to 2024, nine predictive features were constructed to capture market momentum, volatility, liquidity, and price positioning. These features were systematically reduced and validated through LASSO regression and Elastic Net logistic models. A walk-forward validation scheme (2-year training → 1-year testing) ensured that results were temporally consistent and not influenced by data leakage. The goal is to demonstrate an end-to-end pipeline capable of producing stable statistical outcomes while maintaining financial meaning, ultimately contributing a reproducible and practical model for short-term investment decision-making.

2 Design Concept

2.1 Preliminary design

The project investigates short-term stock price prediction over a 1–5 day horizon, focusing on the Commonwealth Bank of Australia (CBA). As the largest domestic bank by market capitalisation and a constituent of the S&P/ASX200, CBA provides both high liquidity and broad market relevance, making it an appropriate benchmark for testing short-term predictive models in the Australian financial market.

Building on this scope, the research aims to capture near-term price dynamics by integrating factor-based modelling with machine learning techniques. The

objective is not only to improve predictive accuracy but also to evaluate the economic value of forecasts through a structured trading simulation framework.

To operationalise this objective, the workflow strictly follows the standard Data Processing pipeline. Figure 1 illustrates the mapping between the six canonical steps (Cleaning, Transformation, Feature Engineering, Splitting, Integration, and Modelling & Output) and the Jupyter workflow (Step 1–12). The timeline at the top of the figure shows how the analysis progressed from problem formulation and data acquisition (Step 1–2), through data cleaning, transformation, and feature engineering (Step 3–4, Step 7, Step 9), to time-series splitting and baseline modelling (Step 5–6), and finally to iterative refinements such as risk/reward calibration, Elastic Net logistic regression, and stress testing (Step 8–12).

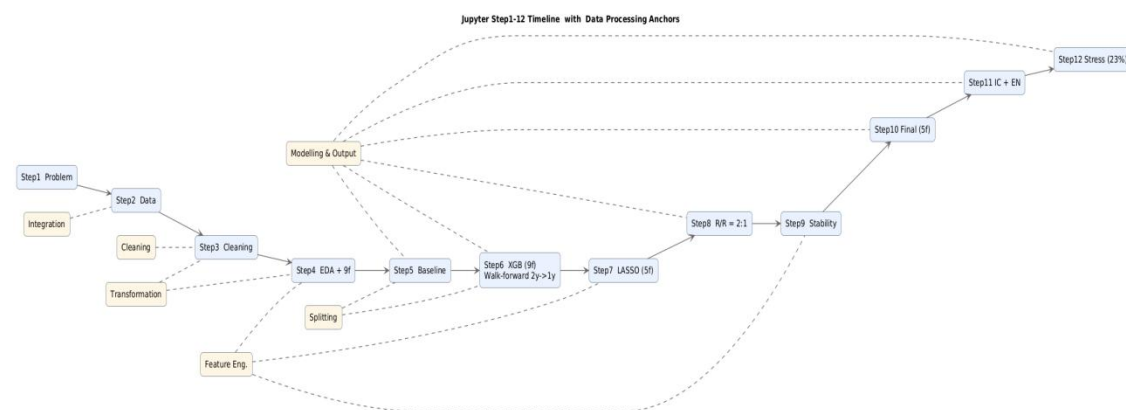


Figure 1: Mapping of the Data Processing six-step pipeline to our Jupyter Step 1–12 workflow.

Finally, Figure 1 shows that the majority of the workflow (Step 5–12) is concentrated in the modelling and evaluation stage, while the early steps ensure data quality and validity. This alignment demonstrates that the analysis moves beyond raw prediction to establish a rigorous, end-to-end short-term trading framework capable of generating actionable insights for investment decision-making.

2.2 Methodology

First, the methodology adopts the Data Processing framework to ensure a structured path from raw market data to validated trading insights. Each step translates the framework into concrete actions for short-term stock prediction.

Next, data cleaning was applied to the CBA daily OHLCV dataset (2020–2024) to address missing values, duplicates, and outliers. Minimal issues were detected, but missing points were forward-filled and data types were standardised for consistency.

Furthermore, data transformation was performed to prevent information leakage and prepare features for modelling. All factors were shifted by one day to avoid look-ahead bias, standardised (Z-score) to remove scale effects, and enriched with parsed time variables.

In addition, feature engineering was carried out to capture key drivers of short-term price movements. Nine predictive factors were initially constructed:

- Momentum: mom_12_1, mom_3_1
- Volatility: rv_20, vol_pk_20
- Trading activity: overnight_mean_5, intraday_mean_5, vol_surp
- Relative price: price_rel_52w, rev_5

These factors represent widely recognised themes in empirical finance, including trend persistence, risk, liquidity dynamics, and price anchoring. LASSO regression later reduced this set to five stable predictors—price_rel_52w, rv_20, vol_pk_20, intraday_mean_5, and mom_12_1—improving stability and interpretability [1]. The correlation structure of the nine constructed factors is shown in Figure 2, while Figure 4 illustrates the retained five factors and their absolute coefficient magnitudes, highlighting their relative importance.

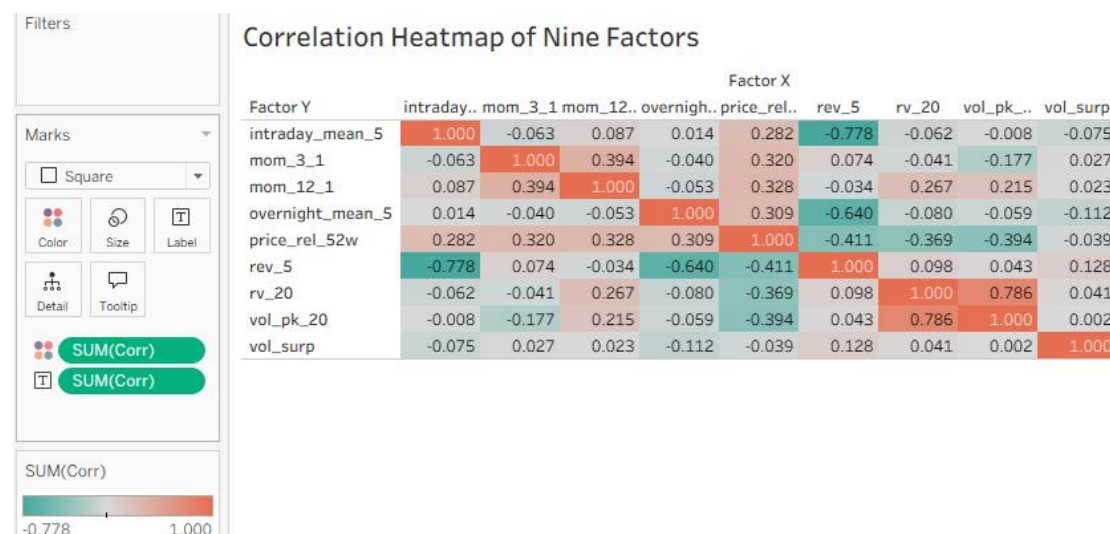


Figure 2: Correlation heatmap of the nine constructed factors.

Meanwhile, data splitting followed a walk-forward scheme to simulate realistic trading conditions. A two-year training window followed by a one-year testing window (2y→1y) ensured temporal order, avoided information leakage, and provided multiple test folds. This design is consistent with empirical research highlighting the importance of temporal robustness in evaluating factor models [2], [3]. Figure 3 illustrates the rolling window design of this validation approach.

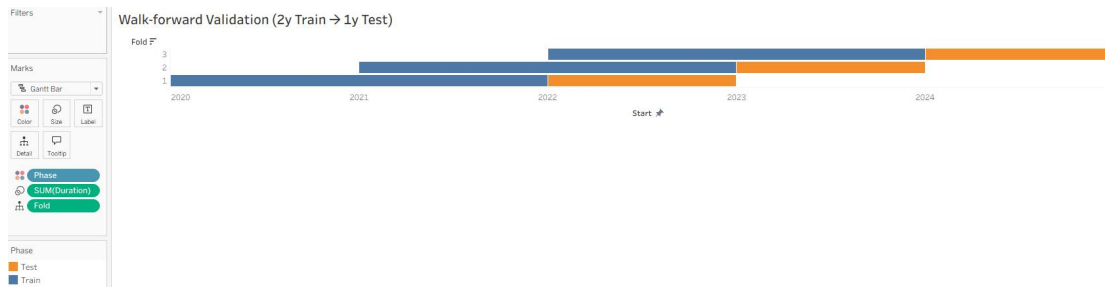


Figure 3: Walk-forward validation scheme (2y training, 1y testing).

Moreover, integration ensured that factor construction, modelling, and evaluation were unified into a reproducible pipeline. Since the study focused on a single stock, no cross-source merging was required, but the workflow maintained coherence across all stages.

Finally, modelling and evaluation concentrated on testing and refining predictive performance under trading constraints. Baseline logistic regression and gradient boosting performed near random. XGBoost and LASSO-based feature selection improved accuracy, while risk/reward calibration ($R/R=2:1$, $\tau=1/3$) doubled expectancy compared to the earlier 1:1 setting. Elastic Net logistic regression with IC-scaling provided a more stable and interpretable benchmark, consistent with recent literature on advanced factor selection and dimension reduction methods [4], [5]. The model delivered approximately 160 trades across five years with an annualised return of $\sim 11.2\%$ when measured over the full five-year period. At this stage, annualised returns were still being calculated against the rough five-year window, which implicitly included the initial two training-only years. In Step 12, this issue was identified and corrected: profitability was recalculated using only the three out-of-sample test years (2022–2024), excluding the training period (2020–2021). The impact of IC-scaling is illustrated in Figure 5, comparing Step 11A (IC-weighted stacking, $\sim 8\%$ annualised return) with Step 11B (IC-scaled EN-Logit, $\sim 11.2\%$ annualised return). Stress testing outcomes across R/R ratios are illustrated in Figure 6, showing that the $R/R=3:1$ setting achieves a higher annualised return ($\sim 23\%$) despite a lower win rate ($\sim 36\%$) compared to the conventional 2:1 setting ($\sim 21.5\%$ annualised, $\sim 45\%$ win rate). This illustrates a fundamental trading principle: increasing the reward-to-risk ratio allows profitability to improve even if the win rate decreases, as the expectancy rises; conversely, a lower reward-to-risk ratio requires a higher win rate to maintain profitability. The progressive improvements in expectancy and annualised return from Step 5–12 are summarised in Figure 7. Together, these steps demonstrate a methodological process that not only corrected earlier mis-specifications but also produced a rigorous, end-to-end framework capable of generating actionable trading insights.

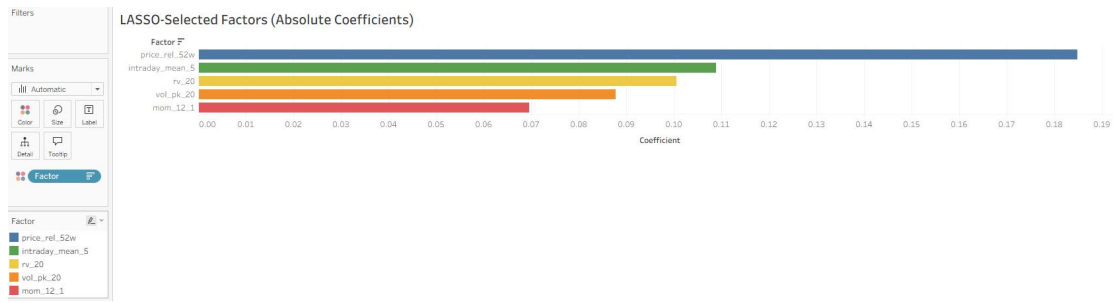


Figure 4: LASSO-selected factors (absolute coefficients).

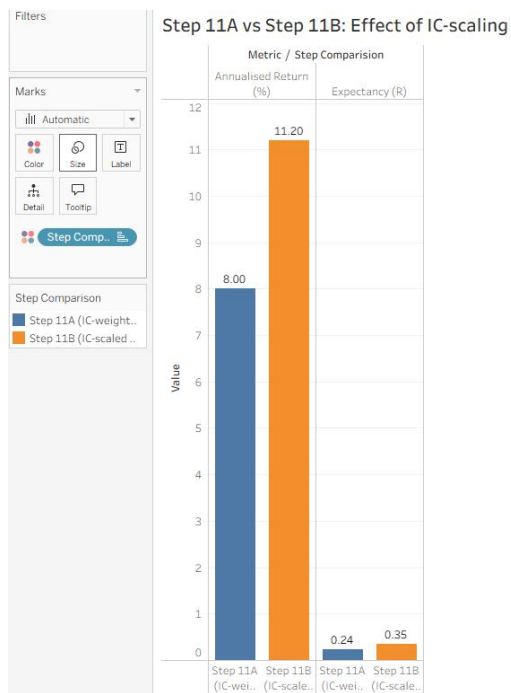


Figure 5: Step 11A vs Step 11B: Effect of IC-scaling.

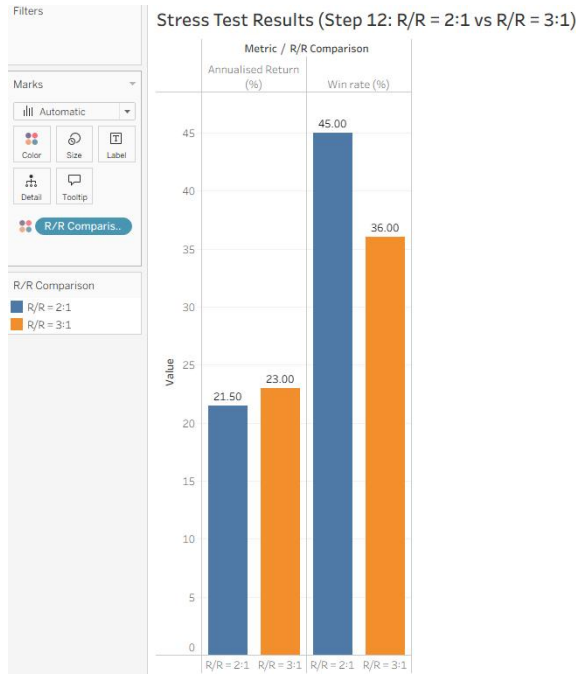


Figure 6: Stress test under alternative risk/reward ratios (Step 12).

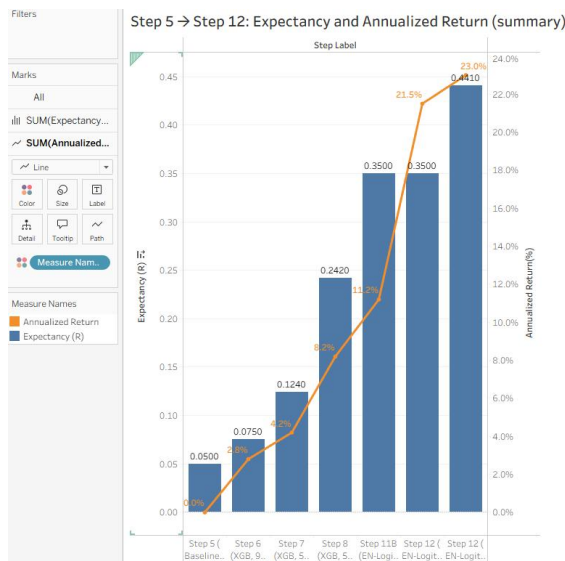


Figure 7: Step 5–12 improvements in expectancy (blue) and annualised return (orange).

2.3 Design constraints

First, the analysis is constrained by the scope of the dataset. The study relied exclusively on daily OHLCV data for the Commonwealth Bank of Australia (CBA) from 2020–2024. This choice was made deliberately to keep the project tractable, as the use of intraday data would have required substantially greater time and resources than were feasible within the present research. Without intraday or order book timestamps, the framework cannot determine the

intraday sequence of stops and targets or explicitly incorporate trading frictions such as transaction costs, slippage, and liquidity shocks. As a simplification, the framework applied a conservative tie-break rule: if both stop-loss and take-profit could be hit within the same bar, the stop-loss was triggered first (SL-first). This design choice biases results downward and ensures that reported returns should be interpreted as an upper bound prior to transaction costs.

In addition, the framework imposed a maximum holding period of five trading days. If neither stop-loss nor take-profit was triggered within this interval, the position was force-closed at the close of day five. This rule ensures consistency with the short-term prediction horizon and introduces a conservative bias: in trending markets stop-losses are usually triggered earlier, while trades that remain open for five days typically occur in sideways markets. Forcing closure in such conditions often leads to small losses or marginal gains, truncating potential profits that might otherwise materialise beyond the five-day window.

Next, the backtesting horizon was deliberately restricted to five years, with a rolling 2-year training and 1-year testing scheme, yielding three years of out-of-sample forecasts. This design reflects the belief that predictive factors have limited validity periods. Using excessively long histories (e.g., 6–8 years) risks diluting signals with stale noise, while overly short windows may fail to capture the persistence needed for factor effects to emerge. Thus, a three-year effective prediction period provides a balance between avoiding noise and ensuring temporal robustness.

Furthermore, while the framework achieved an annualised return of approximately 23% under historical backtesting, overfitting risk cannot be excluded. Factors that appear profitable in-sample may decay rapidly in live markets. Practical deployment would therefore require portfolio-level risk controls, including diversification across assets, position sizing based on predefined loss limits, and maximum drawdown alert thresholds to prevent catastrophic losses in adverse conditions.

Finally, the analysis was restricted to a single security, the Commonwealth Bank of Australia (CBA). No cross-asset or multi-stock validation was undertaken, which limits the generalisability of the discovered factors. This constraint was intentionally adopted to keep the project scope manageable, and it also aligns with the project's objective of emphasising depth of analysis on one representative stock rather than broad market coverage. While this simplification supports clearer methodological demonstration, future research should extend to multiple equities or sectors to test factor robustness across different market environments.

2.4 Specifications

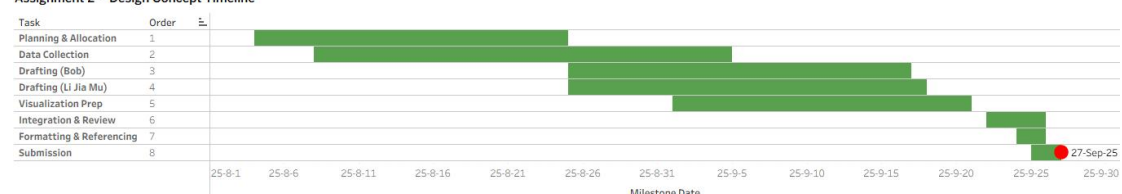
The short-term prediction framework is defined by the following specifications:

- Input data: Daily OHLCV (open, high, low, close, volume) for Commonwealth Bank of Australia (CBA), 2020–2024.
- Prediction horizon: 1–5 trading days, with signals recalculated daily.
- Target variable: Next-day price direction (up/down), extended to multi-day holding periods subject to stop-loss/take-profit rules.
- Feature set:
 - Initial nine factors: momentum (mom_12_1, mom_3_1), volatility (rv_20, vol_pk_20), trading activity (overnight_mean_5, intraday_mean_5, vol_surp), and relative price (price_rel_52w, rev_5).
 - Reduced to five stable predictors via LASSO: price_rel_52w, rv_20, vol_pk_20, intraday_mean_5, mom_12_1.
- Data splitting: Rolling walk-forward validation (2 years training → 1 year testing), producing three out-of-sample folds.
- Models employed: Logistic regression (baseline), Histogram Gradient Boosting (baseline), XGBoost, Elastic Net logistic regression (IC-scaled).
- Trading rules:
 - Fixed stop-loss and take-profit levels, with risk/reward ratio set at 2:1 (SL = 1×ATR, TP = 2×ATR).
 - Threshold calibration $\tau = 1/(1+R) = 0.33$, aligned with the 2:1 risk/reward structure.
 - Maximum holding period of five trading days: if neither stop-loss nor take-profit is triggered, the position is force-closed at the close of day five.
- Stress testing: Additional experiments with alternative R/R ratios (e.g., 3:1) to examine robustness of expectancy and hit rate.
- Evaluation metrics: Win rate, profit factor (PF), expectancy per trade (R-multiple), annualised return, and equity curve progression.

3 Project Management Plan

3.1 Timeline

Assignment 2 – Design Concept Timeline



The assignment was conducted between Week 1 and Week 7, starting from 4 August 2025 and ending with the final submission on 27 September 2025. Key tasks were distributed across the weeks, beginning with planning and allocation, followed by data collection, drafting, visualization, integration, and concluding with the submission milestone. The Gantt chart below illustrates the overall timeline, with the red marker indicating the final milestone date (27-Sep-25).

3.2 Goals and Milestones

- Goal 1: Planning & Allocation – Define responsibilities and set the scope of work.
- Goal 2: Drafting and Development – Produce preliminary content and supporting figures.
- Goal 3: Integration & Review – Consolidate contributions and refine outputs.
- Final Milestone: Submission (27-Sep-25) – Deliver the final report, representing the completion of all planned tasks.

4 Conclusion

The project successfully developed an end-to-end machine learning framework for short-term stock prediction using structured data processing and rigorous validation. The findings show that combining financial factor construction with advanced modelling techniques, such as Elastic Net and XGBoost, enhances both predictive reliability and interpretability. By integrating model calibration, walk-forward testing, and stress evaluation under different risk/reward settings, the analysis established a reproducible pipeline that aligns with real-world trading logic. Although limited to one stock and a five-year period, the methodology provides a foundation for future research on multi-asset extensions, portfolio-level risk management, and broader factor validation. Overall, this work demonstrates how data analytics and machine learning can be combined to bridge technical modelling with financial reasoning in quantitative trading.

5 Appendix

5.1 Reference List

- [1] L. Clausen and J. Strömberg, *Making Use of the Factor Zoo: An Unpretentious Attempt to Predict Asset Returns Using Machine Learning Methods*, Bachelor Thesis, University of Gothenburg, 2022.
- [2] J. Green, J. R. M. Hand, and X. F. Zhang, “The characteristics that provide independent information about average U.S. monthly stock returns,” *The Review of Financial Studies*, vol. 30, no. 12, pp. 4389–4436, 2017.
- [3] J. Freyberger, A. Neuhierl, and M. Weber, “Dissecting characteristics nonparametrically,” *The Review of Financial Studies*, vol. 33, no. 5, pp. 2326–2377, 2020.
- [4] G. Feng, S. Giglio, and D. Xiu, “Taming the factor zoo,” *Working Paper*, 2017.
- [5] B. Kelly, S. Pruitt, and Y. Su, “Characteristics are covariances: A unified model of risk and return,” *Journal of Financial Economics*, vol. 134, no. 3, pp. 501–524, 2019.

5.2 GenAI Declaration

Bob Lin

● Prompts Used:

(1) When applying multiple machine learning models (e.g., LASSO, Elastic Net, XGBoost) in a walk-forward validation framework for stock prediction, how should the methodology section of an academic report be structured to balance technical detail (e.g., factor engineering, hyperparameter tuning) with readability for non-ML readers?

(2) In reporting the results of multiple machine learning models (e.g., LASSO, Elastic Net, XGBoost) for financial prediction, how can the methodology and results sections be structured to both present statistical metrics (e.g., RMSE, accuracy, R^2) and provide economic interpretation (e.g., factor significance, predictive stability) without overwhelming the reader?

● Outputs Received:

(1)

ChatGPT 5 ▾

When applying multiple machine learning models (e.g., LASSO, Elastic Net, XGBoost) in a walk-forward validation framework for stock prediction, how should the methodology section of an academic report be structured to balance technical detail (e.g., factor engineering, hyperparameter tuning) with readability for non-ML readers?

The methodology can be structured in layered steps:

1. **Problem Context** – outline the stock prediction objective and justify the use of multiple models.
2. **Factor Engineering** – explain how features like momentum or volatility are constructed and their financial meaning.
3. **Model Justification** – describe why each model is chosen (LASSO/Elastic Net for feature selection, XGBoost for non-linear interactions).
4. **Walk-Forward Validation** – show how rolling training/testing windows are set up, ideally with a simple diagram.
5. **Hyperparameter Tuning** – mention tuning methods briefly, moving detailed grids to an Appendix.
6. **Results Reporting** – state that metrics (RMSE, R^2 , accuracy) will be presented with short economic interpretation.

👉 This structure keeps technical depth for rigor while staying readable to non-ML readers.

(2)

ChatGPT 5 ▾

In reporting the results of multiple machine learning models (e.g., LASSO, Elastic Net, XGBoost) for financial prediction, how can the methodology and results sections be structured to both present statistical metrics (e.g., RMSE, accuracy, R^2) and provide economic interpretation (e.g., factor significance, predictive stability) without overwhelming the reader?

When reporting the results of multiple models such as LASSO, Elastic Net, and XGBoost, the methodology and results sections should be presented in a layered way that moves from numbers to meaning. Begin with a summary table that compares core statistical metrics like RMSE, accuracy, and R^2 across models so that the technical performance is transparent at a glance. Follow this with concise narrative paragraphs that highlight which models perform best and why, avoiding lengthy technical detail in the main text. Then connect these outcomes to economic interpretation by discussing which factors are most influential, whether their importance is consistent across walk-forward validation windows, and how these results align with financial theory. To address predictive stability, explain briefly if performance holds up over time or if certain models degrade when tested in new windows. More detailed elements, such as hyperparameter grids or full statistical outputs, should be moved to the Appendix so the main report remains readable. This structure allows statistical rigor and economic insight to be conveyed clearly without overwhelming the reader.

● **Modifications Made:**

The AI responses were not used directly; instead, the ideas and structure suggested were adapted. I reorganized the points, rewrote the content in my own words, and aligned it with the assignment rubric.