PROJECT 1: REPRODUCE KEY RESULTS IN A ML-IN-FINANCE STUDY

1. Purpose and Overview

One of the central aims of this course is to prepare you to use data science and machine learning with rigor on real problems in finance, and to leave you ready to begin research or advanced work in quantitative investing and financial analytics. This project is your starting point. Project 1 introduces the full workflow in data science in quantitative finance and the concepts of complexity, generalization error, and out-of-sample performance.

Important Update: Individual Project. According to the suggestions of the director of data science program, I revised this project to be individual so that every student engages with the entire workflow. We encourage high-level discussion of ideas, but all code, analysis, figures, and writing must be your own. Please don't share code, raw predictions, or hidden-test insights.

Deadline: Wednesday, 8 Oct 2025, 23:59 Singapore Time (SGT, UTC+08:00).

2. Context and paper (read first)

This project is based on the Journal of Finance article The Virtue of Complexity in Return Prediction by **Bryan T. Kelly**, **Semyon Malamud**, and **Kangying Zhou** (2024, **79**(1): 459–503). Official DOI: 10.1111/jofi.13298. The Wiley article page provides the Internet Appendix and the Replication Code ZIP.

Brief description and implications.

- In correctly specified settings and ridgeless OLS ($z = 0$), out-of-sample prediction error typically shows a peak at the interpolation threshold $c = P/T_{\text{tr}} \approx 1$ ("double descent"); equivalently, R^2 can be extremely negative near $c \approx 1$ without shrinkage (KMZ Prop. 3; see their figures). Despite negative R^2, the ridgeless Sharpe remains positive for $c \neq 1$ and tends to zero as $c \to 1$ due to exploding variance (KMZ Eq. (15) with Prop. 3).
- In misspecified settings with sufficiently mixed signals (Assumption 5), a timing strategy with weight $\pi_t = S_t'\widehat{\beta}(z)$ can display increasing expected return and Sharpe as observed complexity $c\,q = P_1/T_{\text{tr}}$ grows, provided shrinkage is appropriate near $c\,q \approx 1$ (KMZ Theorem 1).

3. Definitions and Notation

We follow the paper's convention: signals at time t predict the next period return R_{t+1}. Let F denote the forward (lead) operator, acting on a time series $\{R_t\}$ by $(\mathsf{F}R)_t = R_{t+1}$. Assume data is generated by a true (unknown) predictive model of the form

$$R_{t+1} \;=\; S_t'\beta^* \;+\; \varepsilon_{t+1}, \quad t = 1, 2, \ldots$$

with signals $S_t \in \mathbb{R}^P$ and innovations ε_{t+1} independent of S_t. Training uses pairs $\{(S_t, R_{t+1})\}_{t=1}^{T_{\text{tr}}}$, and testing uses $\{(S_t, R_{t+1})\}_{t=T_{\text{tr}}+1}^{T_{\text{tr}}+T_{\text{te}}}$.

CSV format (precise). In all CSVs, each row with index `t` stores features S_t under `feature1..featureP` and, when present, the label `return` $= R_{t+1}$.

We follow the paper's symbols.

- **Signals** $S \in \mathbb{R}^{T\times P}$: row t is S_t' with P features.
- **Returns** $R \in \mathbb{R}^T$: $R_{t+1} = S_t'\beta^* + \varepsilon_{t+1}$.
- **Feature covariance** Ψ: in simulation we take $\Psi = I$.
- **True parameter** $\beta^* \in \mathbb{R}^P$ with strength $\|\beta^*\|_2^2 = b^*$.
- **Sample sizes** P (features) and T (time); we split into training T_{tr} and test T_{te}. **Complexity** $c = P/T_{\text{tr}}$ (true data-generating process complexity).

Date: August 31, 2025.

- **Revealed fraction** $q \in [0,1]$: define $q = P_1/P$. Then observed complexity $c\,q = P_1/T_{\text{tr}}$.
- **Estimated forecast** $\widehat{R}_{t+1}(z) = S_t'\widehat{\beta}(z)$; timing weight $\pi_t(\widehat{\beta}(z)) = S_t'\widehat{\beta}(z)$ (KMZ Eq. (6)).

Define the timing return $R^{\pi}_{t+1} = \pi_t R_{t+1}$ with $\pi_t(\widehat{\beta}(z)) = S_t'\widehat{\beta}(z)$ (KMZ Eq. (6)). The paper's Sharpe ratio is

$$\text{SR} \;=\; \frac{\mathbb{E}[\,R^{\pi}_{t+1}\,]}{\sqrt{\mathbb{E}[(R^{\pi}_{t+1})^2]}} \quad \text{(KMZ Eq. (5))}.$$

We can also consider a variance-based Sharpe $\text{SR}_{\text{var}} = \mathbb{E}[R^{\pi}_{t+1}]/\sqrt{\text{Var}(R^{\pi}_{t+1})}$ as a secondary diagnostic (KMZ note centered vs. uncentered denominators are WLOG). The paper's R^2 identity is

$$R^2_{\text{paper}} = 1 - \frac{\mathbb{E}[(R_{t+1} - \widehat{R}_{t+1})^2]}{\mathbb{E}[R^2_{t+1}]} = \frac{2\,\mathbb{E}[\widehat{R}_{t+1}R_{t+1}] - \mathbb{E}[\widehat{R}^2_{t+1}]}{\mathbb{E}[R^2_{t+1}]} \quad \text{(KMZ Prop. 3)}.$$

4. Tasks

Each student will complete the following tasks.

Task 1: Misspecified simulation with ridge (set $c = 10$, dense β^*). The goal is to reproduce the result of the paper under misspecified, sufficiently-mixed regime (Assumption 5) by observing only a subset of signals. Train on $\{(S_t, R_{t+1})\}_{t=1}^{T_{\text{tr}}}$ and test on $\{(S_t, R_{t+1})\}_{t=T_{\text{tr}}+1}^{T_{\text{tr}}+T_{\text{te}}}$. Use a dense, isotropic β^* (Gaussian normalized so $\|\beta^*\|_2^2 = b^*$). Fix the data-generating process (DGP) complexity $c = P/T_{\text{tr}} = 10$ and vary the observed complexity $c\,q = P_1/T_{\text{tr}}$ by changing $q = P_1/P$, consistent with KMZ Theorem 1.

Suggestions for DGP and observation protocol:

- Generate S with $\Psi = I$ (rows $S_t \sim \mathcal{N}(0, I_P)$).
- Generate dense β^* and normalize to $\|\beta^*\|_2^2 = b^*$.
- Generate returns via the forward operator: $R_{t+1} = S_t'\beta^* + \varepsilon_{t+1}$, $\varepsilon_{t+1} \sim \mathcal{N}(0,1)$; hence $\mathbb{E}[R^2_{t+1}] = 1 + b^*$.
- Misspecification via partial observability: apply a random permutation to columns of S once per simulation run; fix it for all q, then observe only the first P_1 permuted columns; define $c\,q = P_1/T_{\text{tr}}$ with $q = P_1/P$ (more precisely, for a given q, take the first $P_1 = qP$ permuted columns as the observed block).

Suggestions for Ridge and grids:

- Use the estimator as suggested in the appendices; no intercept.
- Shrinkage grid: use reasonable log-spaced regularization including ridgeless $z = 0$ handled by pseudoinverse: for example, $z \in \{0, 0.1, 0.25, 0.5, 1, 2, 5, 10, 25, 50, 100\}$.
- Sweep $c\,q$: use a fine grid around interpolation, For example, $c\,q \in \{0.5, 0.75, 0.9, 0.95, 0.98, 1.02, 1.05, 1.1, 1.25, 1.5, 2, 3, 5, 7, 10\}$.
- Remark: under the additional small cross-block trace condition, KMZ Prop. 6(ii) gives the exact optimum

$$z^* \;=\; \frac{c(1 + b^*(\psi_{*,1}(1) - q\,\psi_{*,1}(q)))}{b^*},$$

  which reduces to $z^* = \frac{c(1+b^*(1-q))}{b^*}$ when $\Psi = I$ (since $\psi_{*,1}(q) = 1$). In the correctly specified case, $z^* = c/b^*$ (Prop. 3). Our wide grid covers these ranges.

Suggestions for (out of sample estimated) metrics and plots (for each z, plot vs. $c\,q$):

- R^2_{paper} (KMZ Prop. 3 identity applied out of sample);
- Expected timing return $\mathbb{E}[R^{\pi}_{t+1}]$;
- Sharpe $\mathbb{E}[R^{\pi}_{t+1}]/\sqrt{\mathbb{E}[(R^{\pi}_{t+1})^2]}$ (KMZ Eq. (5));
- Parameter size $\|\widehat{\beta}(z)\|_2^2$.

Suggestions for Report:

- List all parameters and cite the exact code cells/functions that produced each figure.

Task 2: New model on the same data. Propose a new model (e.g., Lasso or a small neural net); repeat Task 1 on the same simulated data and splits, using your new model; Keep the same $c\,q$ grid and report the same metrics. Note: These methods may depart from KMZ's ridge-based theory; discuss where/why your curves deviate from the ridge benchmarks (sparsity/nonlinearity/implicit shrinkage).

Task 3: Predictions on three data sets A, B, C. There will be 3 data sets A,B,C, each may have different properties. For each data set, the training part and the testing part are generated by the same unknown distribution. The training part can be downloaded (**they will be shared later on Canvas**) but the testing part is hidden. You need to do analysis and choose the model you believe is best for each data set. The out of sample performance will be evaluated on the testing data set. You may want to do validation; if used, it must be internal to the training file (e.g., block split). Save predictions with the exact names below: your-student-id_predictions_A.csv, your-student-id_predictions_B.csv, your-student-id_predictions_C.csv
Important: For A, B, C, the properties of β^* are **not disclosed**. Choose and justify your model based on theory and on-sample evidence.
Fairness: Hidden labels for A/B/C are generated from the same process as the public features; they are withheld only for scoring.

CSV format (precise).

- **Training CSV** → columns `t, feature1..featureP, return`; `t` strictly increasing integers; no duplicates; no NaNs.
- **Public Test CSV** → columns `t, feature1..featureP`; same feature names; `t` strictly increasing integers; no NaNs.
- **Feature order** → derive from training and enforce on test; error if any required feature is missing or extra `feature*` columns appear.
- **Predictions CSV** → UTF-8 with header `t,yhat`; `t` order must match the public test file exactly; no extra columns.

What grading computes. For each pair and submission: OOS R^2_{paper}, $\mathbb{E}[R^{\pi}_{t+1}]$, SR (Eq. (5)), and SR_{var} (secondary).

Task 4: Short video (8–10 minutes). Record a short screencast that runs your code in a clean environment and explains: (i) the key steps and what you learned in Task 1, (ii) what you tried and discovered in Task 2, (iii) how you chose the best model in Task 3 (and why). The video is part of grading.

5. Deliverables and grading rubric

Each student will submit the following deliverables.

Component (requirements)	**Points (total=35)**
Report (PDF, 5–10 pages). Completeness of Tasks 1–4; correct notation and scaling; clear explanation of simulation/data generation; fixed-grid choices for Tasks 1–2; correct formulas; high-quality plots of out of sample R^2_{paper}, $\mathbb{E}[R^{\pi}_{t+1}]$, and SR (Eq. (5)), $\|\widehat{\beta}(z)\|^2$ vs $c\,q$; strict feature alignment and CSV ordering; report–code cross-citations; clarity and insight.	15
Hidden out-of-sample performance (A/B/C). Sound training; predictions that generalize on hidden labels; stable SR and $\mathbb{E}[R^{\pi}_{t+1}]$; correct file names and format.	10
Short video (8–10 minutes). End-to-end clean run (fresh environment, data read, training, optional validation for Task 3, predictions written); articulate what you learned/discovered and justify choices; terminology consistent with the paper; the generated figures and files match the submission.	10

6. Submission

Upload to Canvas by the due date. Submit one package each student.

- **Report (PDF):** file name `your-student-id_report.pdf`. Include your name, email address, and student ID in the title page.
- **Video (MP4 or link):** either upload the MP4 to Canvas or include a working link (for example, an unlisted YouTube, Box, or OneDrive URL) in Canvas. The link must be accessible to staff and downloadable. If a password is required, include it in your submission note.
- **Source code (required):** upload a `.zip` with code *or* provide a GitHub link. Your code package must contain:
  - `README.md` with exact run commands and expected outputs;
  - an environment file (`requirements.txt`);
  - runnable scripts or notebooks;
  - reproducible settings (fixed seeds, relative paths, no hard coded local directories).
- **Predictions CSVs for Task 3:** UTF-8 with header `t,yhat`; `t` order must match the public test file exactly; no extra columns; Save predictions with the exact names below: your-student-id_predictions_A.csv, your-student-id_predictions_B.csv, your-student-id_predictions_C.csv
- **Reproducibility check:** staff will attempt a clean run using your `README.md`. Ensure the commands execute as shown and produce the reported artifacts.

7. Flexibility and expectations (read before you start)

You do not need to match every figure in the paper exactly. You may choose different parameter values, grids, closely related variants of performance metrics, or alternative training methods (especially in Task 2). What matters is that your work clearly illustrates the main concepts (interpolation behavior, the virtue-of-complexity effect under misspecification) and that you explain your choices and findings carefully.

8. Important differences vs. the official replication code

The replication package includes two kinds of code:

- **Theory code** that draws smooth closed-form lines for population limits (see KMZ Prop. 3–6 and Eq. (15)). It does not simulate finite-sample training curves. **KMZ Figures (e.g., 1–6) plot limiting objects. Do not expect one simulation run to reproduce them exactly.** If you overlay population lines, label them clearly.
- **Empirical code** that fits models on real data (e.g., recursive/rolling estimators and shrinkage grids) to produce out-of-sample curves.

In this project you must simulate, then **train** on a training sample and **evaluate out of sample** on a separate test sample. Do not use the theory code to replace finite-sample training/evaluation.

9. Risk, Ethics, and Compliance

- **Academic integrity and licensing.** We welcome high-level discussion of concepts and references, but *all* code, experiments, figures, and writing must be your own. Do not share code, raw predictions, hidden-test results, or bespoke tools with classmates. If you reuse open source code, acknowledge it and comply with its license. Include a `LICENSE` file for any code you publish.
- **Use of generative AI.** You may use generative tools for drafting or coding assistance, but you are responsible for correctness and licensing.

References

- Kelly, B. T., S. Malamud, and K. Zhou (2024): "The Virtue of Complexity in Return Prediction," Journal of Finance, 79(1), 459–503. DOI: 10.1111/jofi.13298. (Wiley article page provides Internet Appendix and Replication Code.)

- Kelly, B. T., S. Malamud, and K. Zhou (2022): "The Virtue of Complexity in Return Prediction," NBER Working Paper No. 30217. `https://www.nber.org/papers/w30217`.
- Kelly, B. T., S. Malamud, and K. Zhou (2021): "The Virtue of Complexity in Return Prediction," SSRN Working Paper. `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3984925`.

Appendix A. Ridge estimator and scaling

On the training sample,

$$\widehat{\beta}(z) = \Big(zI + T_{\text{tr}}^{-1}\sum_{t=1}^{T_{\text{tr}}} S_t S_t'\Big)^{-1}\Big(T_{\text{tr}}^{-1}\sum_{t=1}^{T_{\text{tr}}} S_t R_{t+1}\Big).$$

Matrix form (equivalent). If $X = [S_1, \ldots, S_{T_{\text{tr}}}]'$ and $y = [R_2, \ldots, R_{T_{\text{tr}}+1}]'$, then $\widehat{\beta}(z) = (X'X + zT_{\text{tr}}I)^{-1}X'y$. For ridgeless OLS, use $z = 0$ when $c\,q < 1$; when $c\,q > 1$, interpret $z \to 0^+$ as the Moore–Penrose pseudoinverse (minimum-norm) solution. **Numerical note:** near $z \approx 0$ and $c\,q \approx 1$, use a stable SVD-based pseudoinverse (never invert $X'X$ explicitly).

Appendix B. Sufficient mixing and an additional condition

Assumption 5 (sufficient mixing). The empirical spectral distribution of the observed block $\Psi_{1,1}$ is independent of q.

Additional small-trace condition (used in Prop. 6). A separate technical condition, $\text{tr}(\Psi_{1,2}\Psi_{2,1}) = o(P)$, is used to simplify formulas and pin down the optimal shrinkage in misspecified settings.

With $\Psi = I$ and dense, isotropic β^*, Assumption 5 holds in our simulation; with sparse β^*, monotonic improvements in $\mathbb{E}[R_{t+1}^{\pi}]$ and SR can weaken.