Empirical Asset Pricing via Machine Learning

Lucas DUBOIS, Myriam LAMBORELLE, William LEMAIRE & Tom PARMENTIER

December 14, 2025

HEC Liège

Table of Contents

Introduction:

The paper "Empirical Asset Pricing via Machine Learning" [2] has a twofold objective:

1. Setting new benchmarks to compare ML methods for asset pricing.
2. Synthesizing the asset pricing theory with the emerging ML literature.

What is **Machine Learning (ML)**?

1. High-dimensional models.
   - ✓ Flexibility.
   - ✗ Overfitting.
2. "Regularization" methods for model selection.
   - ✓ Fix Overfitting.
3. Efficient algorithms for searching model specifications.
   - ✓ Fix non-linearities.

- **Universe:** Monthly U.S. equities (CRSP/Compustat)
- **Sample: 30,000 individual stocks**.
- **Period: 60 years** (1957–2016)
- **Predictors: 94 firm-level characteristics** from the "factor zoo" + **74 industry-sector dummies** + interaction terms.
  - **900+ predictors** in total.

The autors use a characteristic OLS model (**OLS-3+H**) , similar to the famous Fama-French 3-factor model [1], as benchmark:

$$\hat{r}_{i,t+1} = \underbrace{\hat{\beta}_1 \, \text{SIZE}_{i,t} + \hat{\beta}_2 \, \text{VALUE}_{i,t} + \hat{\beta}_3 \, \text{MOM}_{i,t}}_{\text{3-factors}} + \hat{\delta}_t + \hat{\gamma}_{j(i)}. \tag{1}$$

This benchmark is commonly used in the empirical asset pricing literature, as well as in practice by asset managers.

AN IMPORTANT METRIC:

Throughout the paper, the authors use the **out-of-sample R^2** to evaluate model performance:

$$R^2_{OOS} = 1 - \frac{\displaystyle\sum_{t=T_0}^{T} (r_{t+1} - \hat{r}_{t+1})^2}{\displaystyle\sum_{t=T_0}^{T} (r_{t+1} - \bar{r})^2} \tag{1}$$

- Measures **raw predictive power** of a model.
- No COMPLEXITY PENALTY term.

Cross-Validation for Tuning Parameters:

The authors use a **rolling one-step-ahead out-of-sample cross-validation** procedure to select the **hyperparameters** of their models:

$$(\lambda^*, \rho^*) = \arg \min_{\lambda, \rho} \sum_{t=T_0}^{T-1} \sum_{i=1}^{N_t} \big(r_{i,t+1} - \hat{r}_{i,t+1}(\lambda, \rho)\big)^2. \tag{2}$$

This allows to:

1. Account for **time-series dependence** in financial data.
2. Mimic how **investors** would forecast their portfolios.
3. Prevent **look-ahead bias** (terrible idea in finance!).

Penalized Regressions:

> *"The simple linear model is bound to fail in the presence of many predictors. When the number of predictors P approaches the number of observations T, the linear model becomes inefficient or even inconsistent."*
> *Gu, Kelly & Xiu [2]*

Why Penalize?

- Low **signal-to-noise ratio** in asset returns $\Rightarrow$ overfitting risk!
- **Multicollinearity** among predictors $\Rightarrow$ unstable estimates!

The basic framework for the penalized regressions takes place as a penalty in
the cost function:

$$\mathbb{L}(\theta, .) = \underbrace{\mathbb{L}(\theta)}_{\text{MSE}} + \phi(\theta, .) \tag{3}$$

Elastic-Net Penalty:

The authors define the **Elastic-Net** penalty as:

$$\phi(\theta; \lambda, \rho) = \lambda(1 - \rho) \sum_{j=1}^{P} |\theta_j| + \frac{1}{2}\lambda\rho \sum_{j=1}^{P} \theta_j^2 \tag{4}$$

This notation can be converted to the one we used in class by letting $\alpha = 1 - \rho$.
The tuning parameters are the mixing parameter (ρ) and the strength parameter (λ).
We can find our *special cases*:

$$\begin{cases} \text{Lasso:} & \rho = 0 \\ \text{Ridge:} & \rho = 1 \end{cases}$$

Algorithm 1 Accelerated Proximal Gradient Method (FISTA)

Initialization: $\theta_0 = 0$, $m = 0$, step size γ

while θ_m not converged **do**

$\quad \bar{\theta} \leftarrow \theta_m - \gamma \nabla L(\theta)\big|_{\theta=\theta_m}$

$\quad \tilde{\theta} \leftarrow \text{prox}_{\gamma\phi}(\bar{\theta})$

$\quad \theta_{m+1} \leftarrow \tilde{\theta} + \dfrac{m}{m+3}(\tilde{\theta} - \theta_m)$

$\quad m \leftarrow m + 1$

end while

Result: Final parameter estimate θ_m

Dimensionality Reduction:

$$R = Z\theta + E, \tag{5}$$

where $\begin{cases} R \in \mathbb{R}^{NT \times 1} & \text{is the stacked vector of returns } r_{i,t+1}, \\ Z \in \mathbb{R}^{NT \times P} & \text{is the stacked predictor matrix } z'_{i,t}, \\ \theta \in \mathbb{R}^{P \times 1} & \text{is the full coefficient vector}, \\ E \in \mathbb{R}^{NT \times 1} & \text{is the stacked residual vector } \varepsilon_{i,t+1}. \end{cases}$

$$R = (Z\Omega_K)\,\theta_K + \tilde{E} \tag{6}$$

where $\begin{cases} \Omega_K \in \mathbb{R}^{P \times K} & \text{contains the component weights } (w_1, \ldots, w_K), \\[2mm] Z\Omega_K \in \mathbb{R}^{NT \times K} & \text{is the dimension-reduced predictor matrix,} \\[2mm] \theta_K \in \mathbb{R}^{K \times 1} & \text{is the coefficient vector in reduced space,} \\[2mm] \tilde{E} \in \mathbb{R}^{NT \times 1} & \text{is the new residual vector.} \end{cases}$

- *P* can be very large (hundreds or thousands of characteristics), but expected returns live in a *low-dimensional* space.
- PCR / PLS construct *K ≪ P predictive components*:

$$Z\Omega_K \in \mathbb{R}^{NT \times K}, \quad \Omega_K = [w_1, \ldots, w_K].$$

- Each w_j is a set of weights forming a linear combination of characteristics (a "latent factor" or component).
- Instead of forecasting with many noisy, collinear signals, we forecast with a few stable, informative components that summarize the underlying economic forces driving returns.

	Penalized Regression	**Dimension Reduction**
Goal	shrink coefficients θ	reduce predictors to $K \ll P$
Form	$\min_\theta L(\theta) + \phi(\theta)$	$R = (Z\Omega_K)\theta_K + \tilde{E}$
Dimensionality	stays at P	becomes K
Complexity Control	penalty (L1/L2/EN)	projection $Z \to Z\Omega_K$
Examples	Ridge, Lasso, EN	PCR, PLS

Principal Components Regression (PCR)

1. Compute eigenvectors of $Z^\top Z$.
2. Select the top K eigenvectors:

$$\Omega_K = [w_1, \ldots, w_K].$$

3. Construct components:

$$C = Z\Omega_K.$$

4. Run OLS on reduced predictors:

$$R = C\theta_K + \tilde{E}.$$

PCR finds directions in Z with the *largest variance*. It assumes that predictive power lies in high-variance directions of the characteristic space.

PARTIAL LEAST SQUARES (PLS)

1. First component solves:

$$w_1 = \arg \max_{\|w\|=1} (R^\top Zw)^2.$$

2. Form the first score:

$$c_1 = Zw_1.$$

3. Deflate Z and R; repeat to obtain $w_2, \ldots, w_K$:

$$\Omega_K = [w_1, \ldots, w_K].$$

4. Regress on reduced predictors:

$$R = (Z\Omega_K)\theta_K + \tilde{E}.$$

PLS chooses directions *most useful for predicting returns*, making it supervised and more targeted than PCR.

	PCR	**PLS**
Optimizes	variance of Z	covariance with R
Supervised?	No	Yes
Components	eigenvectors of $Z^\top Z$	directions maximizing $R^\top Z w$
Reduced predictors	Variance components	Predictive components
Strength	Captures structure of Z	Return-predictive signals
Weakness	Ignores R in components	Nonlinearities(?)

Non-Linear Models & Results

The Linear Limitation:

- Linear models assume the effect of a predictor (e.g., Value) is constant, regardless of other conditions (e.g., Volatility).
- **Reality:** Asset pricing is likely conditional.
    - *Example:* "Value" might only work well during high volatility.

The Machine Learning Solution:

- Use flexible functional forms $g(z_{i,t})$ to capture **interactions** without specifying them ex-ante.
- We examine three classes: **GLM**, **Regression Trees**, and **Neural Networks**.

GLM introduces non-linearity via **Splines** (piecewise polynomials) for each predictor individually.

$$g(z;\theta) = \sum_{j=1}^{P} \sum_{k=1}^{K} \theta_{j,k}\, p_k(z_j) \tag{7}$$

- **Penalty:** Uses **Group Lasso** to select or drop entire characteristics (group of spline terms) at once.
- **Limitation:** It is additive. It captures non-linearities of *single* variables (z_j^2), but **misses cross-variable interactions** ($z_i \times z_j$).
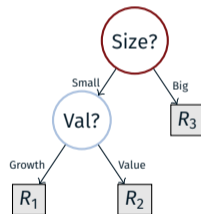
2. Tree-Based Methods

Trees partition the predictor space into rectangular regions and predict the average return in each "leaf."

Random Forests (RF):

- Builds B deep, decorrelated trees.
- Uses **Dropout**: Randomly subsets predictors at each split to force diversity.
- Prediction is the *average* of all trees (reduces variance).

Gradient Boosted Trees (GBRT):

- Sequentially builds shallow "weak learners."
- Each tree corrects the errors of the previous one.

Size?

Small | Big

Val? | R_3

Growth | Value

R_1 | R_2

Captures interactions naturally.

The most flexible model. Composed of an **Input Layer**, **Hidden Layers** (nonlinearities), and an **Output Layer**.

Feed-Forward Architecture
For a neuron k in layer l, the value is a non-linear transformation of the previous layer:

- **Activation:** Rectified Linear Unit ($ReLU(x) = \max(0, x)$).
- **Regularization:** Essential to prevent overfitting.
    - L_1 Penalty, Early Stopping, Batch Normalization, Ensembles.
- **Architectures:** Tested NN1 (1 layer) to NN5 (5 layers).

Results: Out-of-Sample R^2

Non-linear models dominate linear benchmarks.

Model	OLS-3	En-Net	GLM	RF	NN3
R^2_{OOS} **(%)**	0.16	0.11	0.19	0.33	**0.40**

- **GLM Failure:** Performance is close to linear models $\Rightarrow$ Univariate non-linearity is *not* enough.
- **NN3 Peak:** Performance peaks at 3 layers. "Shallow" learning works best for finance (low signal-to-noise).
- **Economic Value:** Small R^2 gains translate to large profits.

Annualized Sharpe Ratios for Long-Short Decile Portfolios (Value-Weighted):

- Neural Networks **more than double** the risk-adjusted returns of the OLS benchmark.

By analyzing variable importance (Sensitivity Analysis), models agree on the dominant signals:

1. **Price Trends:** Momentum (12m), Short-term Reversal (1m).
2. **Liquidity:** Market Cap, Dollar Volume, Bid-Ask Spread.
3. **Volatility:** Idiosyncratic Volatility, Beta.

Why do NN and Trees win? They capture **Interaction Effects**.

- *Example:* The "Reversal" effect is strong for small stocks but weak/concave for large stocks. Linear models miss this nuance.

Conclusion:

What the Paper brings to **Empirical Asset Pricing**

1. Systematic Comparison
2. Bridges tradition with modernity
3. Shows limits

The Dataset

1. 30 000+ U.S. stocks
2. Almost 60 years of monthly data
3. 900+ predictors from the "factor zoo"
4. Traditional models

	NN3	NN4	NN2	NN5	RF	GBRT+H
All	0.40	0.39	0.39	0.36	0.33	0.34
Top 1,000	0.67	0.62	0.70	0.64	0.63	0.52
Bottom 1,000	0.47	0.46	0.45	0.42	0.35	0.32

	OLS+H	ENet+H	OLS-3+H	GLM+H	PCR	PLS
All	-3.46	0.11	0.16	0.19	0.26	0.27
Top 1,000	-11.28	0.25	0.31	0.14	0.06	-0.14
Bottom 1,000	-1.30	0.20	0.17	0.30	0.34	0.42

Sharpe Ratio (H-L) Best Models:

	NN4	NN3	NN1	NN2	NN5	RF
SR(H−L)	1.35	1.20	1.17	1.16	1.15	0.98

SHARPE RATIO (H-L) WORST MODELS:

	ENet+H	OLS-3+H	PLS	GLM+H	GBRT+H	PCR
SR(H−L)	0.39	0.61	0.72	0.76	0.81	0.88

Conclusion:

Key Takeaways

1. Return prediction improvement
2. Asset pricing evolution

Limitations of the Paper

1. Interpretability
2. Large datasets
3. International robustness

Final Message

The Dataset

1. New approach
2. Important modernization step

References:

[1] Eugene F. Fama and Kenneth R. French. **"Common risk factors in the returns on stocks and bonds".** en. In: *Journal of Financial Economics* 33.1 (Feb. 1993), pp. 3–56. ISSN: 0304405X. DOI: 10.1016/0304-405X(93)90023-5. URL: https://linkinghub.elsevier.com/retrieve/pii/0304405X93900235 (visited on 11/21/2025).

[2] Shihao Gu, Bryan Kelly, and Dacheng Xiu. **"Empirical Asset Pricing via Machine Learning".** en. In: *The Review of Financial Studies* 33.5 (May 2020). Ed. by Andrew Karolyi, pp. 2223–2273. ISSN: 0893-9454, 1465-7368. DOI: 10.1093/rfs/hhaa009. URL: https://academic.oup.com/rfs/article/33/5/2223/5758276 (visited on 10/24/2025).