Stock Jump Prediction: End-to-End Modeling, Evaluation, and Market Implications

Arjo Bhattacharya

September 27, 2025

Abstract

This report documents the complete workflow and empirical results for a monthly stock jump prediction assignment based on the brief *AS5_jump_prediction.pdf* and my Google Colab implementation. I have constructed a binary target that flags next-month "jumps" (absolute return exceeding 10%), engineer firm-level technical features (lags, rolling volatility and momentum, log price and log market capitalization), merge macro factors (CPI, UNRATE, VIX, USREC, S&P 500), add coarse industry dummies, and integrate CAPM-based risk variables (rolling beta, systematic and idiosyncratic volatility) via a vectorized rolling-moments method. To keep computation tractable, I have sampled 100 firms per year after feature construction. Models—Logistic (baseline), LASSO with Post-LASSO refit, Ridge, KNN, XGBoost, and LightGBM—are trained with a temporal split (train $\leq$2017, validate 2013–2017, test 2018–2023). We evaluate by ROC–AUC, Kolmogorov–Smirnov (KS), and misclassification using KS-optimal thresholds, and (per the brief) tune LightGBM's depth to minimize validation misclassification. Post-LASSO Logistic delivers the best balance (AUC ≈ 0.709, KS ≈ 0.316, Misclass ≈ 0.332). Then I have discussed market implications for risk management and trading, limitations, and robustness extensions (rolling/fixed-window evaluation).

1 Problem Definition and Data Scope

Objective. Predict whether a stock experiences a large next-month absolute return ("jump") using only information available at the current month-end. Let $R_{i,t}$ be stock i's return in month t. We forecast the indicator

$$y_{i,t+1} = \mathbb{1}\{\, |R_{i,t+1}| > 0.10 \,\}, \tag{1}$$

a binary classification problem.

Universe and Horizon. We use CRSP monthly stock file (MSF) from **Jan 1996–Dec 2023**. Before sampling, the panel contains about **1,535,255** firm-months across **15,862** PERMNOs. Macro factors are sourced as follows: CPI, UNRATE, VIX, and US recession indicator (USREC) from FRED; S&P 500 (level) from Yahoo Finance. We also use CRSP value-weighted market return (`vwretd`) as a feature and for CAPM variables.

Out-of-sample Design. Per the brief, our primary scheme trains through 2017, validates 2013–2017, and holds out **2018–2023** for testing. We also include code for rolling and fixed-window logistic evaluation.

2 Label Construction and Exploratory Context

2.1 Target Definition

We form $R_{i,t+1}$ by shifting each stock's return one month forward within PERMNO. The label is

$$y_{i,t+1} = \begin{cases} 1, & \text{if } |R_{i,t+1}| > 10\% \\ 0, & \text{otherwise,} \end{cases}$$

and we compute an overall jump rate time series (share of firms with $y_{i,t} = 1$ each month). In our panel, jumps are non-trivial (roughly high tens of percent at times), indicating a predictable tail-event set-up is at least plausible.

2.2 Macro Alignment and Qualitative Patterns

We align CPI, UNRATE, VIX, USREC, and S&P 500 to month-end and overlay the jump-rate timeline. Qualitatively, jump frequency co-moves with VIX and recession episodes, consistent with systemic stress amplifying tail risk. This satisfies the brief's requirement to examine potential precursors and macro context.

3 Feature Engineering

3.1 Firm-Level Technical Features

Within each PERMNO, sorted by date, we compute:

- **Lagged returns:** `ret_lag1`, `ret_lag3`.
- **Rolling volatility:** `vol_3m`, `vol_6m`, `vol_12m` as rolling standard deviations of monthly returns.
- **Rolling momentum:** `mom_3m`, `mom_6m`, `mom_12m` as rolling sums of monthly returns.
- **Levels:** $\log(1 + \texttt{PRC})$ and $\log(1 + \texttt{MKT\_CAP})$.

3.2 Macro and Industry Factors

We merge monthly **CPI**, **UNRATE**, **VIX**, **USREC**, and **S&P 500** level. SIC codes are mapped to coarse industries and one-hot encoded (including `industry_nan`).

3.3 CAPM-based Risk Features (Prior Assignment Integration)

To integrate prior work (as required), we compute rolling **beta**, **systematic volatility**, and **idiosyncratic volatility** using a *vectorized* 36-month rolling-moments approach:

$$\beta_{i,t} = \frac{\mathrm{Cov}(R_i, R_m)}{\mathrm{Var}(R_m)}, \tag{2}$$

$$\mathrm{sysVol}_{i,t} = \beta_{i,t}^2 \cdot \mathrm{Var}(R_m), \tag{3}$$

$$\mathrm{idioVol}_{i,t} = \mathrm{Var}(R_i) - \mathrm{sysVol}_{i,t}, \tag{4}$$

where R_m is `vwretd`. This is algebraically equivalent to OLS slope with intercept, yet avoids per-window regressions. Diagnostics confirm 0% missing in $R_{i,t}$ and $R_{m,t}$ and over 1.06M non-null betas after warm-up.

3.4 Final Covariate Set

After excluding identifiers and targets, the modeling matrix contains **41 covariates**: 12 CRSP/market fields, 10 firm-technical features, 5 macro variables, 3 CAPM variables, and 11 industry dummies.

4 Sampling, Splits, and Metrics

4.1 Sampling 100 Firms per Year

To reduce compute while preserving cross-sectional diversity, we first compute all features on the full panel and then sample **100 PERMNOs per calendar year** with light eligibility checks (e.g., minimum valid months). This yields ∼31k firm-months, maintaining temporal breadth.

4.2 Temporal Splits

We sort by date and define: **Train** (through 2017), **Validation** (2013–2017), and **Test** (2018–2023). Linear models and KNN use median imputation and standardization; tree models are imputed-only.

4.3 Evaluation Metrics and Thresholding

We report **ROC–AUC**, **KS** (max TPR–FPR), and **misclassification** after converting probabilities to classes via **KS-optimal thresholds**. For LightGBM, per the brief, we also evaluate misclassification at the *validation-chosen* threshold after tuning depth by validation misclassification.

5 Models and Tuning

5.1 Logistic Regression (Baseline)

A median-imputed, standardized logistic (LBFGS) serves as the baseline. On the 2018–2023 test:

$$\text{AUC } 0.6931, \quad \text{KS } 0.2796, \quad \text{Misclass } 0.3678 \ (\text{KS-thr } \approx 0.4442).$$

Confusion matrix: TN=2515, FP=1674, FN=911, TP=1929.

5.2 LASSO and Post-LASSO Logistic

We tune $C \in \{0.01, 0.03, 0.1, 0.3, 1.0\}$ on validation by AUC (fast, no CV), then refit a plain logistic on the *selected* features (Post-LASSO) using Train+Valid. Best $C = 0.03$; LASSO selects 32 features (vol/mom, log levels, VIX, CPI, USREC, S&P 500, vwretd, liquidity/size proxies, industry dummies). Test performance:

$$\text{AUC } 0.7089, \quad \text{KS } 0.3161, \quad \text{Misclass } 0.3315 \ (\text{KS-thr } \approx 0.4288).$$

Confusion: TN=2985, FP=1204, FN=1126, TP=1714.

5.3 Ridge Logistic

Validation-tuned $C \in \{0.01, 0.03, 0.1, 0.3, 1.0, 3.0\}$ yields best $C = 0.01$. Test:

$$\text{AUC } 0.7071, \quad \text{KS } 0.3124, \quad \text{Misclass } 0.3379 \ (\text{KS-thr } \approx 0.4153).$$

Confusion: TN=2878, FP=1311, FN=1064, TP=1776.

5.4 K-Nearest Neighbors (KNN)

$k \in \{3, 5, 7, 9, 15, 25\}$ tuned by validation AUC. On test, KNN underperforms:

$$\text{AUC } 0.5910, \quad \text{KS } 0.1446, \quad \text{Misclass } 0.4390.$$

5.5 XGBoost

Small grid on $(n\_t, \text{depth}, \eta)$: $n\_t \in \{100, 200, 300\}$, depth $\in \{3, 4\}$, $\eta \in \{0.05, 0.1\}$. Best validation: (300, 4, 0.1). Test:

$$\text{AUC } 0.7028, \quad \text{KS } 0.3086, \quad \text{Misclass } 0.3595 \text{ (KS-thr } \approx 0.3384).$$

Confusion: TN=2439, FP=1750, FN=777, TP=2063.

5.6 LightGBM (GOSS) — per Brief

We tune `max_depth` by *validation misclassification* (threshold chosen by validation KS). Best depth = 9. Retrained on Train+Valid, the test metrics are:

$$\text{AUC } 0.7036, \quad \text{KS } 0.3143, \quad \begin{cases} \text{Misclass } 0.3507 & \text{(test KS threshold)} \\ \text{Misclass } 0.3385 & \text{(validation-chosen threshold).} \end{cases}$$

Confusion at test KS threshold: TN=2582, FP=1607, FN=858, TP=1982.

6 Comparative Results

Table 1 summarizes the test performance. For comparability across models, the main misclassification column uses *test* KS-optimal thresholds; we also display LightGBM's *validation-threshold* misclassification per the brief.

Table 1: Test-set Performance Summary (2018–2023)

Model	AUC	KS	Misclass (test KS)	BestThr (test KS)	TN	FP	FN	
Post-LASSO Logistic	0.7089	0.3161	0.3315	0.4288	2985.0000	1204.0000	1126.0000	1
Ridge Logistic	0.7071	0.3124	0.3379	0.4153	2878.0000	1311.0000	1064.0000	1
LightGBM (depth=9)	0.7036	0.3143	0.3507	0.2674	2582.0000	1607.0000	858.0000	1
XGBoost (tuned)	0.7028	0.3086	0.3595	0.3384	2439.0000	1750.0000	777.0000	2
Logistic (baseline)	0.6931	0.2796	0.3678	0.4442	2515.0000	1674.0000	911.0000	1
KNN (tuned)	0.5910	0.1446	0.4390	0.3333	2150.0000	2039.0000	1047.0000	1

LightGBM misclassification at validation-chosen threshold = **0.3385**.

Key Takeaways. Post-LASSO Logistic delivers the *best overall balance* of AUC, KS, and lowest error, narrowly beating Ridge. Gradient boosting models show competitive AUC/KS, but at test KS thresholds they classify more positives (higher recall) and thus incur more false positives and higher error. KNN is not suitable for this high-dimensional, noisy cross-section.

7 Interpretation and Market Implications

7.1 Signal Economics

LASSO selection emphasizes volatility (total/idiosyncratic), momentum (3–12 month), price/size levels, and VIX—all consistent with risk-based narratives: elevated volatility and macro stress increase tail propensities; momentum captures trend continuation and breakout risk; levels encode liquidity/clientele effects.

7.2 Risk Management

Jump probabilities enable *proactive* risk control: resize exposures in high-risk names, add tail hedges (OTM options), and tilt portfolios toward lower predicted jump propensity during stress. Post-LASSO's transparency and stable separation (KS ≈ 0.316) make it attractive for risk governance.

7.3 Trading Use-Cases

While the target is directionless (absolute jump), it maps naturally to long-vol structures (straddles/strangles), event-risk targeting, and dispersion strategies. Boosters (XGBoost/LightGBM) may be favored where *recall* is critical (catch more jump months), provided thresholding and costs are explicitly optimized.

8 Limitations and Robustness

Sampling variance. Limiting to 100 firms/year is pragmatic but introduces variance; a full-universe run may refine ranks. **Monthly frequency.** Many jumps are tied to earnings or macro days; higher-frequency features could help. **Feature set.** We used widely accepted signals; adding earnings calendars, analyst revisions, option-implied asymmetry, news/sentiment, and microstructure features is promising. **Non-stationarity.** Regime shifts (GFC, COVID) can alter relationships; our code includes rolling and fixed-window logistic evaluation to assess stability. **Costs and thresholds.** KS thresholds are cost-insensitive; business-aligned loss functions (e.g., λ_FN $>$ λ_FP) may favor boosters.

9 Conclusions

Post-LASSO Logistic is the strongest all-around performer on 2018–2023, balancing accuracy and interpretability. Ridge is a close second. LightGBM (depth tuned by validation misclassification) and XGBoost are competitive rankers but require explicit cost-aware thresholding to reduce false positives. In practice, risk teams can deploy the Post-LASSO model for transparent jump-risk surveillance, while trading teams might leverage boosted models with recall-oriented thresholds for long-vol selection, subject to rigorous cost and liquidity modeling.

Reproducibility. All results derive from the Colab notebook implementing the steps described above, with deterministic seeds for yearly sampling and model tuning. The code also includes fast vectorized CAPM computation, LightGBM tuning per brief, and optional rolling/fixed-window evaluation.